

Anders Akeberg  
Benjamin Bradaric  
Håkon Einarsen  
Thanish Gnanasegaram

# Design and Construction of an Automated Calibration System Utilizing Electric Motors for Underwater Cameras

Bachelor's thesis in Mechanical Engineering  
Supervisor: Håkon J. D. Johnsen  
May 2024





Anders Akeberg  
Benjamin Bradaric  
Håkon Einarsen  
Thanish Gnanasegaram

# **Design and Construction of an Automated Calibration System Utilizing Electric Motors for Underwater Cameras**

Bachelor's thesis in Mechanical Engineering  
Supervisor: Håkon J. D. Johnsen  
May 2024

Norwegian University of Science and Technology  
Faculty of Engineering  
Department of Mechanical and Industrial Engineering





REPORT BACHELOR THESIS	
<b>Title</b> - Design and Construction of an Automated Calibration System Utilizing Electric Motors for Underwater Cameras - Design og konstruksjon av et automatisert kalibreringssystem som utnytter elektriske motorer for undervannskameraer	
<b>Project number</b> MTP-K-2024-13	
<b>Authors</b> - Anders Akeberg, Benjamin Bradaric, Håkon Einarsen, Thanish Gnanasegaram	
<b>Company (external)</b> - Optoscale	
<b>Supervisor NTNU</b> - Håkon J. D. Johnsen	
Report is OPEN	
<b>Date</b> - 21.05.2024	
<b>Short abstract both in English and Norwegian:</b>  - To provide fish monitoring, Optoscale utilizes underwater cameras. These cameras require precise calibration within a pool involving the movement of a calibration plate with three degrees of freedom. To achieve precise movement with customizable speed and range, a comprehensive system is required, integrating electric motors, mechanical structures, programming and a user-friendly interface. The purpose of this thesis is to design and construct a reliable automated solution for plate movement, enhancing efficiency and precision in the calibration process.  - For å tilby informasjon om fisk, benytter Optoscale undervannskameraer. Disse kameraene krever nøyaktig kalibrering innenfor et basseng som involverer bevegelsen av en kalibreringsplate med tre grader av frihet. For å oppnå presis bevegelse med tilpassbar hastighet og rekkevidde, kreves et omfattende system som integrerer elektriske motorer, mekaniske strukturer, programmering og et brukervennlig grensesnitt. Formålet med denne oppgaven er å designe og konstruere en pålitelig automatisert løsning for platebevegelse som øker effektiviteten og presisjonen i kalibreringsprosessen.	
<b>Keywords:</b> - Arduino Programming - Camera Calibration - Construction - Electric motors - Fluid Mechanics - Mechatronics	<b>Stikkord:</b> - Arduino Programmering - Kamera kalibrering - Konstruksjon - Elektriske motorer - Fluidmekanikk - Mekatronikk

---

## Preface

This bachelor thesis was completed by four mechanical engineering students, and was made possible by Optoscale. Optoscale is a company in Trondheim that specializes in aquaculture, specifically the surveillance of fish welfare through underwater bioscopes. The entrusted task given to us by Optoscale was to automatize their in-place calibration system for the cameras in their bioscopes. They have been meaning to develop it themselves but it was not prioritized. This was a perfect opportunity for us as mechatronics is a common interest. The finished automation system was to be completed by mid-May of 2024.

The scope of the work included programming, selection of components, mechanical calculations and solutions, as well as electrical setup and implementation at the Optoscale laboratory. The implementation included drilling, creating screw threads, soldering wires, and general installation of components under professional supervision by the personnel at the laboratory.

We would like to send our gratitude to our professor and supervisor throughout the project, Håkon J.D Johnsen, as well as professors Anna Olsen for mechanical expertise. We would also like to extend our gratitude to Optoscale for allowing us to work on this project, aiding resources, giving us access to their workshop and providing general assistance throughout the project.

Date 21.05.2024



---

Anders Akeberg



---

Benjamin Bradaric



---

Håkon Einarsen



---

Thanish Gnanasegaram

---

## Abstract

To provide fish monitoring, the company Optoscale utilizes underwater cameras which are an essential part of their bioscope. These cameras require precise calibration within a pool involving the movement of a calibration plate with three degrees of freedom. To achieve precise movement in a predefined path within the camera's field of view, and customizable speed and range, a comprehensive system is required.

The calibration process was prior to this project done manually. A worker pushed the carriages on the rails and its connecting plate along the width and length of the pool. This process was time consuming and frustrating for the workers at Optoscale. The purpose of this thesis is to design and construct a reliable automated solution for the plate movement enhancing efficiency and precision in the calibration process, meeting challenges specified by Optoscale.

The concept's development aimed for seamless integration with the existing rail system, drawing inspiration from 3D-printer gantry systems. Mechanical solutions, including gears, pulleys, and belts, were employed as transmission for movements. Cable chains managed cables, while bolts, nuts, and brackets supported installations, enhancing stability in construction.

Electrical components, such as a microcontroller, stepper motors, drivers, a servomotor, and transformers was implemented. An electric setup were housed in an electronic control panel. System limits were determined through simplified mechanical and fluid analyses, considering factors like motor torque and plate integrity. Programming involved writing code in Arduino C/C++, featuring a state machine architecture for sequential motor control and efficient management of user input signals. A user interface, was designed using 3D modeling, and securely installed onto the side of the pool. Effective communication between different components and thorough testing was crucial for the system's success.

The final result was a fully functioning and automated calibration system implemented on the in-place railing system at Optoscale's laboratory. The system included a user interface with a start-and stop button and a digital joystick that moves the carriage in all directions. It was capable of calibrating cameras with the click of a button. Optoscale expressed great satisfaction with the outcome of the project. In conclusion, this thesis successfully completed the task. However, there are opportunities for future enhancements.

---

## Sammendrag

For å tilby informasjon om fisk, benytter Optoscale undervannskameraer. Disse kameraene er en viktig del av bioskopet, og krever nøyaktig kalibrering innenfor et basseng som involverer bevegelsen av en kalibreringsplate med tre frihetsgrader. For å oppnå presis bevegelse med tilpassbar hastighet og rekkevidde, kreves et omfattende system.

Kalibreringsprosessen ble tidligere utført manuelt. En arbeider dyttet vognene på skinnene og dens tilknyttede plate langs bassengets bredde og lengde. Denne prosessen var tidkrevende og frustrerende for arbeiderne hos Optoscale. Formålet med denne oppgaven er å designe og konstruere en pålitelig automatisert løsning for platebevegelse som øker effektiviteten og presisjonen i kalibreringsprosessen.

Konseptets utvikling hadde som mål å integreres sømløst med det eksisterende skinnesystemet, med inspirasjon hentet fra store 3D-printer systemer. Mekaniske løsninger, inkludert tannhjul og belter for overføring, ble brukt til bevegelser. Kabelkjeder håndterte kabler, mens bolter, muttere og braketter støttet installasjoner, og økte stabilitet.

Elektriske komponenter, som en mikrokontroller, steppermotorer med drivere, servomotor og transformatorer ble implementert. Elektrisk oppsett ble plassert i et elektronisk kontrollpanel beskyttet mot vann. Systemgrensene ble fastsatt gjennom forenkede mekaniske og fluidanalyser, med hensyn til faktorer som motorenes dreiemoment og plateintegritet. Programmering involverte skriving av kode i Arduino C/C++, med en tilstandsmaskinarkitektur for sekvensiell motorstyring og effektiv håndtering av brukerinngangssignaler. Et brukergrensesnitt, designet med 3D-modellering, ble installert sikkert på siden av bassenget. Effektiv kommunikasjon mellom ulike komponenter og grundig testing var avgjørende for systemets suksess.

Det endelige resultatet var et fullt fungerende og automatisert kalibreringssystem implementert på det eksisterende skinnesystemet i Optoscales laboratorium. Systemet inkluderte et brukergrensesnitt med start- og stoppknapp og en digital styrespak som beveger vognen i alle retninger. Det kunne kalibrere kameraer med et tastetrykk. Optoscale uttrykte stor tilfredshet med resultatet av prosjektet. Konklusjonen er at dette prosjektet vellykket fullførte oppgaven, men det er muligheter for fremtidige forbedringer.

---

## Notations and Abbreviations

**AC** Alternating current

**CAD** Computer aided design

**CM** Center of mass

**CFP** Critical function prototype

**DC** Direct current

**DIR** Direction

**MCU** Microcontroller Unit

**PID** Proportional-integral-derivative

**PWM** Pulse width modulation

**PUL** Pulse

---

## List of Symbols

- $A$  - Area/Ampere
- $a$  - Length of moment arm
- $C_d$  - Drag coefficient
- $d$  - Distance
- $F_d$  - Drag force
- $f$  - Frequency
- $i$  - Gear ratio
- $L$  - Characteristic Length of an Object
- $M$  - Moment
- $N$  - Amount of steps in stepper motor
- $Re$  - Reynolds number
- $T$  - Torsion
- $t$  - Duration
- $V$  - Volume/Voltage
- $W_y$  - Section modulus
- $z$  - Number of teeth on a gear
- $\alpha$  - Angular acceleration
- $\eta$  - Dynamic viscosity
- $\kappa$  - Servo delay
- $\rho$  - Density
- $\tau$  - Torque
- $\omega$  - Angular velocity



---

# Table of Contents

<b>Preface</b>	<b>ii</b>
<b>Abstract</b>	<b>iii</b>
<b>Sammendrag</b>	<b>iv</b>
<b>Notations and Abbreviations</b>	<b>v</b>
<b>List of Figures</b>	<b>xii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Current Calibration Solution . . . . .	2
1.3 Objective . . . . .	4
1.4 Performance Goals . . . . .	4
1.5 Impact Goals . . . . .	5
1.6 Scope . . . . .	5
1.7 Limitations . . . . .	6
1.8 Organization . . . . .	6
<b>2 Theory</b>	<b>7</b>
2.1 Gantry System . . . . .	7
2.1.1 Basic Components . . . . .	7
2.2 Stepper Motors . . . . .	8
2.3 Servo Motors . . . . .	8
2.4 Motor Drivers . . . . .	9
2.5 Microcontrollers . . . . .	9
2.6 Law of Gearing . . . . .	10
2.7 Fluid mechanics . . . . .	11

---

2.7.1	Turbulent and Laminar Flow . . . . .	11
2.7.2	Reynolds Number . . . . .	11
2.7.3	Drag Force . . . . .	12
2.7.4	Drag Coefficient . . . . .	13
2.7.5	Added Mass . . . . .	14
2.7.6	Turbines . . . . .	15
2.8	Mechanics of Materials . . . . .	16
2.8.1	Mechanical Stress . . . . .	16
2.8.2	Combined Stresses . . . . .	17
2.8.3	Beam Theory . . . . .	17
2.8.4	Shear Force and Bending Moment Diagrams . . . . .	18
2.9	3D Modelling and Simulations . . . . .	19
2.9.1	Computer-Aided Design software . . . . .	19
2.9.2	Finite Element Analysis . . . . .	19
2.9.3	Software and Hardware Simulation . . . . .	19
<b>3</b>	<b>Method</b>	<b>20</b>
3.1	Product Development . . . . .	20
3.2	Electronics . . . . .	20
3.3	Analysis . . . . .	20
3.4	Drag Coefficients . . . . .	21
3.4.1	Driving Force Assumptions . . . . .	22
3.5	Rotational Torque Assumptions . . . . .	23
3.5.1	Worst-Case Scenario Assumptions . . . . .	23
3.5.2	Euler Turbine Approach Assumptions . . . . .	24
3.6	Limiting Factor . . . . .	25
3.6.1	Linear Movement Assumptions . . . . .	25
3.6.2	Rotational Movement Assumptions . . . . .	26

---

---

3.6.3	Material Strength . . . . .	26
3.7	Software Development . . . . .	27
3.8	Full-Scale Model Implementation . . . . .	28
<b>4</b>	<b>Concept Development</b>	<b>29</b>
4.1	Foundation . . . . .	29
4.2	Linear Motion . . . . .	30
4.2.1	Movement Imbalance . . . . .	31
4.2.2	Two-Pulley Solution . . . . .	32
4.3	Rotary motion . . . . .	35
4.3.1	Rotary Motion Solution . . . . .	35
4.4	Cable Chains . . . . .	36
4.5	Interface Concept . . . . .	36
4.6	Finalized Concept . . . . .	37
<b>5</b>	<b>Components and Implementation</b>	<b>38</b>
5.1	Mechanical Components . . . . .	38
5.2	X- and Y-Axis Motors . . . . .	39
5.2.1	X- and Y-axis Torque Requirements . . . . .	39
5.2.2	X- and Y-Motor Selection . . . . .	41
5.2.3	Implementation of X- and Y-Motors . . . . .	42
5.3	Rotational Motor . . . . .	43
5.3.1	Rotational Torque Requirements . . . . .	43
5.3.2	Rotational Motor Selection . . . . .	45
5.3.3	Implementation of Rotary Solution . . . . .	46
5.4	Cable Management . . . . .	47
5.5	Microcontroller . . . . .	48
5.6	Power Supply . . . . .	49

---

5.6.1	AC/DC 48V Transformer . . . . .	49
5.6.2	AC/DC 5V Transformer . . . . .	49
5.6.3	DC/DC 48V Input 6.5V Output . . . . .	49
5.6.4	Fuse Switch . . . . .	50
5.7	Implemented User Interface . . . . .	50
5.8	Electrical Schematic Diagram . . . . .	51
5.9	Implemented Electronic Control Panel . . . . .	52
<b>6</b>	<b>Limiting Factors of the System</b>	<b>53</b>
6.1	Linear Motion Parameters . . . . .	53
6.1.1	Motor Limits . . . . .	53
6.1.2	Limits of the Pole . . . . .	54
6.1.3	Limits of the Plate . . . . .	55
6.1.4	Abaqus Simulation . . . . .	55
6.2	Rotational Limits . . . . .	57
<b>7</b>	<b>Software</b>	<b>58</b>
7.1	Arduino . . . . .	58
7.2	Code Criteria . . . . .	60
7.3	Programming Thought Process . . . . .	65
7.4	Code Description . . . . .	66
<b>8</b>	<b>Results</b>	<b>67</b>
8.1	The Finished Implemented System . . . . .	67
8.2	System Modifications and Tuning . . . . .	68
<b>9</b>	<b>Discussion</b>	<b>69</b>
9.1	Calculations . . . . .	69
9.2	Electrical Components . . . . .	69

---

---

9.3	Sustainability . . . . .	70
9.4	Evaluation of Goals and Results . . . . .	70
9.5	Future Development . . . . .	71
9.5.1	User Interface . . . . .	71
9.5.2	Feedback . . . . .	71
9.5.3	Code Improvement . . . . .	72
<b>10</b>	<b>Conclusion</b>	<b>73</b>
	<b>References</b>	<b>74</b>
	<b>Appendix</b>	<b>80</b>

---

## List of Figures

1	The Bioscope . . . . .	1
2	Calibration Plate Front View . . . . .	2
3	Calibration Plate Rear View and Camera Position . . . . .	2
4	Calibration Path . . . . .	3
5	Cartesian Gantry System . . . . .	7
6	Stepper Motor Steps . . . . .	8
7	Pulse Width Control . . . . .	8
8	Simplified H-Bridge . . . . .	9
9	Kinematic Representation of Connected Gears . . . . .	10
10	Laminar and Turbulent Flow . . . . .	11
11	Streamlined vs Non-streamlined Body in a Current . . . . .	12
12	Drag Coefficients of a Cylinder Perpendicular Flow . . . . .	13
13	Flow Over a Parallel Plate . . . . .	14
14	Centrifugal Pump . . . . .	15
15	Centrifugal Pump Control Volume . . . . .	15
16	Types of Stress . . . . .	16
17	Combined Stresses . . . . .	17
18	Beam Coordinates and Their Simplification in 2D . . . . .	17
19	Beam with an Imaginary Cut . . . . .	18
20	Forces and Moments on the System . . . . .	22
21	Tilt . . . . .	22
22	Equivalent Scenario . . . . .	23
23	Control Volume Assumption . . . . .	24
24	Cantilever Beam . . . . .	25
25	Representation of the Plate Viewed from Above . . . . .	25

---

26	Wokwi Simulation Model . . . . .	27
27	Assembled Fusion Prototype . . . . .	28
28	Prototype . . . . .	28
29	Rail System with Updated Terminology . . . . .	29
30	Timing Pulley Concept . . . . .	30
31	Carriage Imbalance . . . . .	31
32	Large Axle Concept . . . . .	32
33	Dual Motor Solution . . . . .	33
34	Rotational Motion Concept . . . . .	35
35	Limiting Space . . . . .	36
36	Final Concept Sketch . . . . .	37
43	Nema23 Stepper Motor . . . . .	41
44	CL57 Stepper Driver . . . . .	41
45	Implemented Y-Axis Motor . . . . .	42
46	Implemented X-Axis Motor . . . . .	42
47	GoBilda Servo Motor . . . . .	45
48	Implemented Rotary Solution . . . . .	46
49	Large Cable Chain . . . . .	47
50	Small Cable Chain . . . . .	47
51	Arduino Mega . . . . .	48
52	Tech Specs . . . . .	48
53	Connecting Wires . . . . .	50
54	Soldering Wires . . . . .	50
55	User Interface . . . . .	50
56	Schematic Diagram . . . . .	51
57	Control Panel . . . . .	52
58	Drag Force of the Plate as a Function of Velocity . . . . .	53

---

59	Moment and Shear Diagram for the Pole . . . . .	54
60	Moments and Shear Force Diagram for the Plate . . . . .	55
61	Simulated Stress Distribution . . . . .	56
62	Flow During Deceleration . . . . .	57
63	Video of the Final Product . . . . .	67



---

# 1 Introduction

Based on (National Oceanic and Atmospheric Administration, 2023), aquaculture is the breeding, rearing, and harvesting of marine organisms in all types of water environments. According to (National Geographic, 2023) aquaculture "is the fastest growing industry for producing protein, one of the building blocks of our diet.". Fish farming, a branch of aquaculture, is a very important industry for the future that contributes to achieving the fourteenth sustainability goal of the united nations which aims to "Conserve and sustainably use the oceans, seas and marine environment for sustainable development."

## 1.1 Background

Optoscale AS, a technology company in Trondheim specializes in developing advanced monitoring technology to track the growth and health of fish in aquaculture with minimal manual intervention. By deploying advanced cameras within fish pens, they deliver precise real-time measurements of weight, lice and overall health. This information contributes to ensuring optimal fish welfare while simultaneously driving cost reduction and optimizing efficiency.

Optoscale's newly developed bioscopes are tested in their laboratory. The bioscope used for testing can be seen in Figure 1 (The Bioscope, 2023). Testing of the bioscopes entails submerging them into a pool of water, followed by a calibration process. The calibration process involves moving a plate submerged in water to predetermined locations. The cameras are then calibrated by tracking the plate's movement. Additionally, during the calibration, the plate is rotated around its symmetry axis in front of the camera. The calibration process is completed in their laboratory in a small pool of water. A railing system has been installed onto the pool to aid in moving the plate.

This calibration process was prior to this project, done by hand where a worker pushed the rail system manually to move the plate around in the pool. The plate is required to move very slowly within the camera frame for the calibration to be successful, and this was time consuming.



Figure 1: The Bioscope

---

## 1.2 Current Calibration Solution

The rail system used for camera calibration consists of two parallel carriages traversing the pool's length of three meters, with both carriages being connected via an additional rail. The rail connecting the two carriages is being traversed by a third carriage moving along the pool's width of two meters. A metallic pole attached to the plate passes through the third carriage via a pillow bearing. The plate has an area of  $0.49 \text{ m}^2$  and thickness of  $7 \text{ mm}$ . The plate was manually pushed along the width and length of the pool, and rotated with a mounted bar. The calibration rig used for camera testing can be seen in Figures 2 and 3.

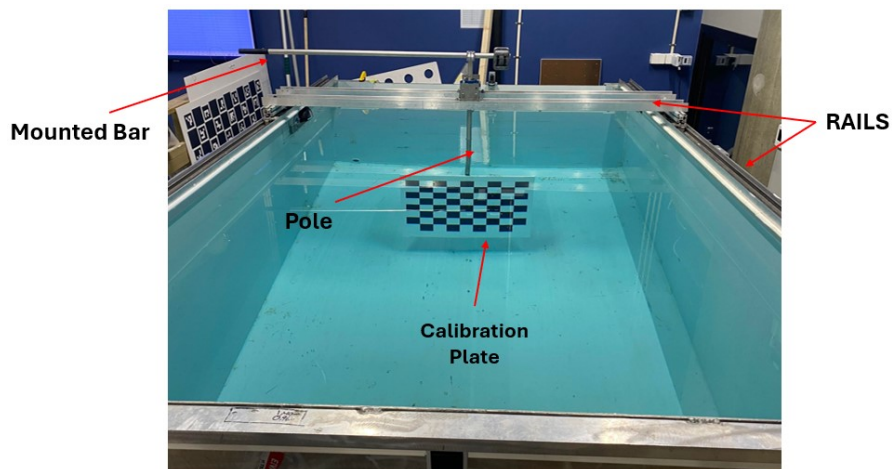


Figure 2: Calibration Plate Front View

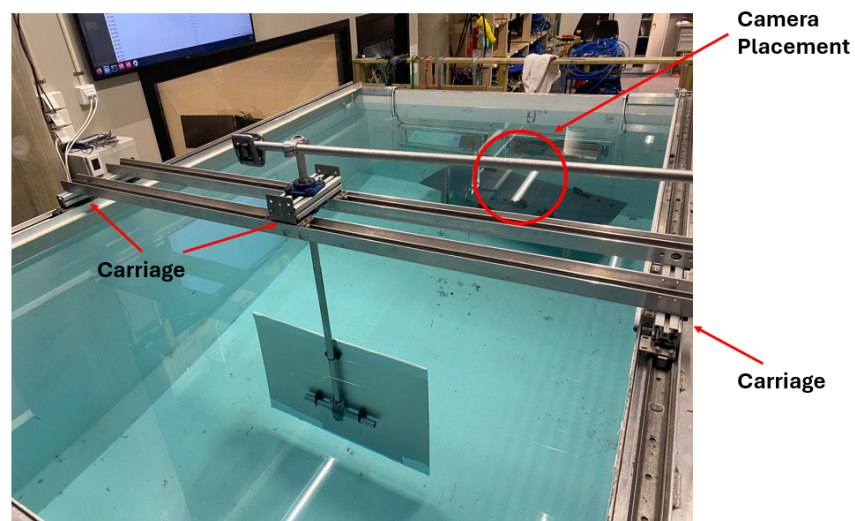


Figure 3: Calibration Plate Rear View and Camera Position

Figure 4 illustrates an approximate path in which the plate moves to calibrate the camera. The red point situated in the middle of the pool serves as the starting point of the plate during calibration. Furthermore, moving along the defined path illustrated by the arrows before returning to the starting point. When the plate returns to its initial position it will perform a rotation both ways around the longitudinal axis of the pole, afterwards the calibration process is finished.

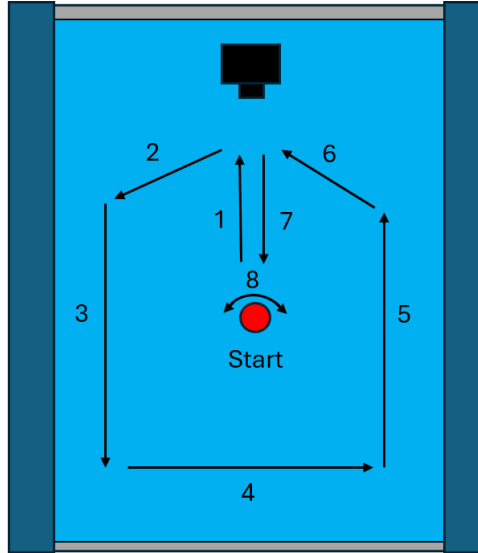


Figure 4: Calibration Path

The information given in Table 1 entails specifications surrounding the system in place at Optoscale's laboratory. For optimal camera calibration an estimated speed of  $20\text{cm/s}$  was given for linear motion along the width and length of the pool. The optimal speed when rotating the plate is approximately  $10\text{ degrees/s}$ . Additionally, the plate should approximately rotate 30 degrees to both sides.

Pool's length	3 m
Pool's width	2 m
Pool's average depth	1.5 m
Plate length	0.7 m
Plate width	0.7 m
Plate thickness	7 mm
Pole outer diameter	25 mm
Pole inner diameter	21.5 mm
Pole length	1.3 m

Table 1: Current system measurements

---

### 1.3 Objective

*This project is tasked with developing and constructing an automated system for maneuvering the calibration plate within the pool. The solution should allow the plate to be moved along the predefined path automatically. Additionally, the system's movement should have the option to be controlled by a user interface, allowing a worker to do so utilizing buttons and joysticks.*

### 1.4 Performance Goals

The project's main objective is to develop an automated calibration system for the company Optoscale. Following this main objective a set of performance goals can be drawn as foundational building blocks to achieve project success.

- **Developing a Mechanical Solution**

The system does not contain the necessary components for an automatic solution. Consequently, mechanical concepts need to be evaluated and constructed onto the in-place system.

- **Evaluation of Electrical Components**

Electrical components necessary in the automation process must be evaluated, ordered and integrated onto the mechanical solutions.

- **Mechanical Analysis of the System**

Forces affecting the plate must be considered and calculated to grasp a better understanding of the systems limits.

- **Development of the Script**

A program must be written to control the electronics of the system where the necessary specifications and plate positioning are defined.

- **Implementation on the Full-Scale Model**

Mechanical solutions and respective electrical components must be integrated onto the pool, furthermore the automation program needs to be tested on the finished system and tuned accordingly to fit within the camera view.

---

## 1.5 Impact Goals

By the end of the semester the team wanted to have acquired valuable experiences and skills in areas such as mechatronics, automation, fluid mechanics, project managing and programming as well as more experience in practical aspects like, using tools, and cooperation with an outside partner. The expectation was that the team find innovative and effective solutions that also complement the green shift through the United Nations sustainability goals; specifically, Goal 14: "Life below water".

The goal is to meet the expectations and standards of Optoscale, create flexibility and allow them to allocate extra personnel in any way they deem useful. In addition it will also introduce them to a new modifiable and reliable system that can be adapted to similar laboratories.

## 1.6 Scope

The calibration process was previously performed manually, so the provided values for approximate velocities and the plate's rotation angle for optimal calibration might be insufficient. Consequently, the created automation system must be tuned and tested with a bioscope to ensure that the values are appropriate. If the values are not precise for optimal calibration, consequently the code will be altered.

The project entails selecting suitable components such as electrical motors and driving belts. Furthermore, finding mechanical solutions to integrate the components into a cohesive system. The project entails development of a software to make components cooperate. The project also includes discovering the limitations of the system through mechanical calculations and the use of failure theory.

A construction budget of 30,000 Norwegian kroner was allocated for the procurement of necessary electronic and mechanical components. While the project does not emphasize the economic aspects of construction, the cost of components and adherence to the budget were considered. The primary objective was to develop a robust and long-lasting system, with Optoscale prioritizing functionality and durability.

---

## 1.7 Limitations

The project had boundaries that specified the working conditions. As the pool is in use during the working hours of the company and well integrated in the work flow, the team was told not to make dramatical changes to the existing rail system. Consequently, little freedom was given to conceptualize functioning solutions for the movements.

Mentioned previously, a budget was set to limit the spending, but still allowing freedom of choice to realize an efficient movement without additional constraints. The partnering company also requested that the user interface was to be fastened to the edge of the pool, and not on a remote controller.

## 1.8 Organization

This report is structured so that the relevant theory, methods and assumptions are presented before discussing the development of the concept as well as workshop labor and installation of components. Subsequently, it contains the approach to establish the operational limits of the system, and the programming thought process and execution, ahead of the presentation and evaluation of the final result.

---

## 2 Theory

The theory included in this section lays the foundation to every aspect in this project, such as programming, designing, mechanics, automation etc. The theory will also provide a basis for the choices to be made for the duration of the project.

### 2.1 Gantry System

Gantry systems, widely used in various industries, are integral to industrial automation. Also known as Cartesian robots, they are mechanical structures designed for multi-axis operations. Typically, a gantry system comprises two linear rails and an overhead bridge, allowing movement along the x, y, and z axes, as shown in Figure 5 (GANTRY ROBOTS , n.d.). By combining different axes and adding rotation, these systems can perform specific movement patterns. Ideal for automated movement, gantry systems offer benefits such as precision, speed, efficiency, productivity, and versatility in numerous applications (Rao, 2023).



Figure 5: Cartesian Gantry System

#### 2.1.1 Basic Components

Linear rails provide the foundation for sliding motion, varying by load, bearing type, size, speed, and precision requirements. Sliders, fitted to these rails, compensate for friction and facilitate smooth sliding using wheels with spheres. Key to any bearing is its long lifespan, often enduring thousands of daily revolutions with proper maintenance. Motors drive the linear motion by converting electrical energy into mechanical motion, crucial for speed, accuracy, and efficiency in gantry systems. Common choices are stepper and servo motors, each with distinct advantages depending on requirements. Control systems and interfaces translate commands into signals, managing motors, drivers, and sensors. They ensure high precision, speed, and efficiency in the gantry system.

---

## 2.2 Stepper Motors

Stepper motors are widely used in engineering for their precise positioning. They consist of multiple "steps" or magnets around a rotor that can be switched on and off by electrical fields as shown in Figure 6 gathered from (Fiore, n.d.). They have high torque at low speeds, long lifespans due to no contact between the rotor and stator, and excellent response times for stopping and rotating in any direction. This makes them ideal for tasks requiring precise movement, including automated patterns, as they can count steps to remember positions (Motioncontrolproducts, n.d). However, if power is lost, an encoder is needed to determine the exact position. An encoder, which can be external or built-in, checks the motor's position (Nanotec, n.d).

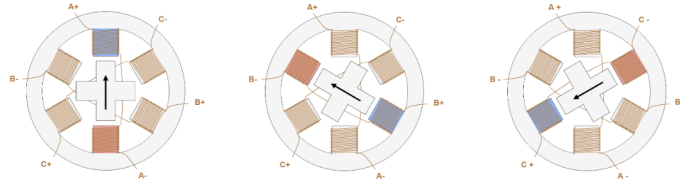


Figure 6: Stepper Motor Steps

The precision of each step can be calculated using the following formula, that shows the relationship between the amount of total steps and the change of angle between each one.

$$Step\ angle = \frac{360^\circ}{N} \quad (1)$$

## 2.3 Servo Motors

Servomotors are electrical motors used in machinery requiring high angular precision. They consist of an AC or DC motor controlled via PWM, a position sensor, and a controller with a closed-loop feedback system (Electrical4U, 2023). The controller uses feedback from the sensor and an external input (e.g., a pulse width baseline of  $1500\mu s$  as shown in Figure 7) to generate an error signal, causing the motor to move until the error signal is zero or within tolerance. PID control algorithms can enhance precision.

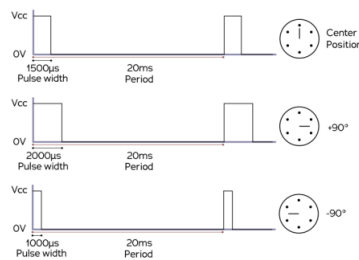


Figure 7: Pulse Width Control



---

## 2.4 Motor Drivers

Motors need a driver circuit to be able to use the input from an MCU as these values are too low to create motion. A motor driver is an integrated circuit that receives signals from the MCU before interpreting and amplifying them, and as a result controls the direction and speed of a motor (Robocraze, n.d). *PUL* and *DIR* signals control the movement and direction of the stepper motor. By varying the timing and sequence of these signals, you can achieve precise control over the stepper motor's rotation. Drivers for DC motors also need to include H-bridges that allow the motor to rotate both ways shown in Figure 8. These are typically included. (Dahl, 2018).

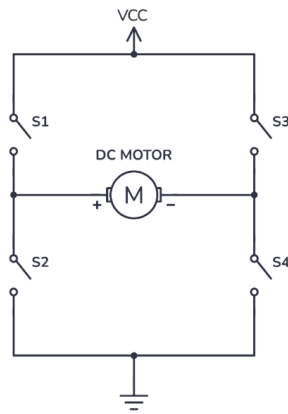


Figure 8: Simplified H-Bridge

## 2.5 Microcontrollers

A microcontroller unit (MCU) serves as the brain of a system and controls its components. When selecting an MCU, it is important to consider capabilities and compatibility. An MCU integrates a processor core, memory, and programmable input/output ports. Its purpose is to execute specific tasks and control connected devices based on stored instructions.

Critical to the MCU is its operational voltage, the voltage where it is designed to work and function optimally. In addition to voltage considerations, features like digital input/output pins, flash memory for program storage, and SRAM (Static Random-Access Memory) for temporary data retention significantly influence the MCU's functionality. Clock speed further dictates the pace at which instructions are processed, directly impacting overall performance. Thus, a comprehensive understanding of these factors is essential when evaluating MCU options for a given application. (Johnsen, 2024)

---

## 2.6 Law of Gearing

Gears are mechanical components with teeth, designed to transmit rotational motion and torque between each other. By employing gears of varying sizes, it is possible to modify the torque or angular velocity according to specific requirements.

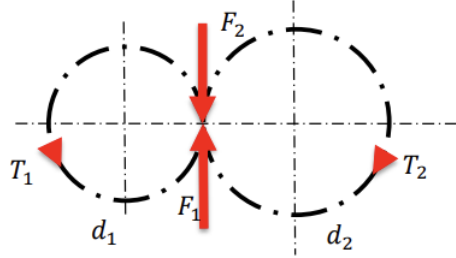


Figure 9: Kinematic Representation of Connected Gears

For a pair of connected gears, the forces that act on each gear are identical, as illustrated in Figure 9, and the relationship between their rotational velocities and the consequent torque transmission can be described by the gear ratio, as shown in Equation 2(Olsen, 2022).

$$i = \frac{\omega_1}{\omega_2} = \frac{z_2}{z_1} = \frac{d_2}{d_1} = \frac{\tau_2}{\tau_1} \quad (2)$$

---

## 2.7 Fluid mechanics

Due to the calibrations system moving a plate submerged in water, certain fluid mechanical concepts will be discussed throughout the report. The following section entails fluid mechanical theory which is applied during development.

### 2.7.1 Turbulent and Laminar Flow

Laminar and turbulent flows are terms used to describe the state of a dynamic fluid. Laminar flow describes the movement of the fluid as structured and smooth streamlines of motion. In contrast, a turbulent flow describes the fluid as erratic and with fluctuating velocities. A third state of flow is transitional flow which describes the fluid in a state where it transitions between laminar and turbulent. The difference between laminar and turbulent flow is illustrated in Figure 10. Determining the state of a fluid as it moves around an object is mainly based on the value of the Reynolds number. However, other factors such as geometry, surface roughness, flow velocity and surface temperature can influence the state of the fluid (Cengel and Cimbala, 2018, p.354)

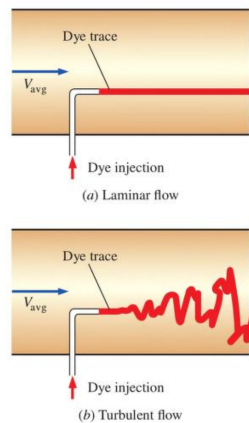


Figure 10: Laminar and Turbulent Flow

### 2.7.2 Reynolds Number

To perform calculations on objects moving through a fluid it is important to consider the state of flow. An indication of the flow state is expressed Equation 3 for Reynolds number, that utilizes the ratio between the inertial force of an object and the viscous force of the fluid. In this equation,  $L$  is the characteristic length of the object, or in simple terms the dimension that affects the flow, and  $\rho$  and  $\eta$  are the density and dynamic viscosity of the fluid respectively.

$$Re = \frac{\rho v L}{\eta} \quad (3)$$

---

The value of the Reynolds number will approximate the amount of turbulence in a current. The higher the Reynolds number, the greater degree of turbulence is present. The transition point at which flow shifts from laminar to turbulent is influenced by several factors, including whether the flow is external or internal, as well as the geometry of the object involved(Reynolds number, n.d).

### 2.7.3 Drag Force

For any object moving through a fluid it is important to consider the fluid's influence on the movement, specifically the drag force experienced on the object from the fluid. The drag force depends on the geometry and orientation of a body, where the skin friction drag comes from the shear wall forces that are typically present in streamlined bodies shown to the left in Figure 11 from (Cengel and Cimbala,2018), where the flow is parallel with the body, and the pressure drag present in non-streamlined bodies shown to the right where the flow is perpendicular to the body.

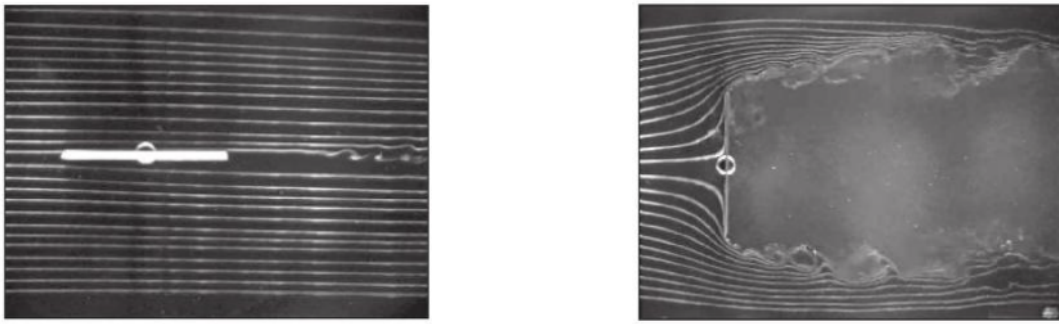


Figure 11: Streamlined vs Non-streamlined Body in a Current

For a flow moving towards an object, the force from the fluid is proportional to the velocity of the flow squared, shown in Equation 4, and can be used to calculate the drag force acting on the object.

$$F_d = \frac{1}{2} \rho v^2 C_d A \quad (4)$$

Where  $\rho$  serves as the density of the fluid,  $v$  velocity of the object and  $C_d$  is the object's drag coefficient. The area  $A$  of the object will depend based on the type of drag force. For streamlined objects with low pressure drag, the relevant area corresponds to the surface area subject to friction. Conversely, for non-streamlined bodies with high pressure drag, the critical area is the one perpendicular to the flow direction.

### 2.7.4 Drag Coefficient

According to (Cengel and Cimbala, 2018, p.616), "The drag and lift forces depend on the density  $\rho$  of the fluid, the upstream velocity  $V$ , and the size, shape, and orientation of the body, among other things, and it is not practical to list these forces for a variety of situations.", and therefore the drag coefficient becomes useful as a dimensionless number that describes the characteristics of the body, that can be determined experimentally. To calculate the drag coefficient, Equation 4 can be rewritten to solve for  $C_d$  and perform tests at various velocities, following the assumption that  $F_d \propto v^2$ . For a Reynolds number above  $10^4$  the drag coefficient will in most cases remain constant, and the values for common geometries can typically be found in tables.

if  $Re > 10^4$  then the drag coefficient for thin plates with cross sectional area vertical to the flow can be found using Equation 5 based on Figure C1 from Appendix C. Where L and D represent the plates length and height, respectively. Additionally the formula only applies when  $\frac{1}{30} < \frac{L}{D} < 30$  (Cengel and Cimbala, 2018, p.624).

$$C_D = 1.10 + 0.02 \cdot (L/D + D/L) \quad (5)$$

Determining the drag coefficients of a pole vertical to the flow can vary based on differing diameters and lengths of the pole. Additionally, its drag coefficients are only constant at laminar flows where the  $Re \lesssim 2 \times 10^5$ . Figure 12 gathered from (Cengel and Cimbala page, 2018, p.624) illustrates the way to calculate the drag coefficients on a pole perpendicular to the flow.

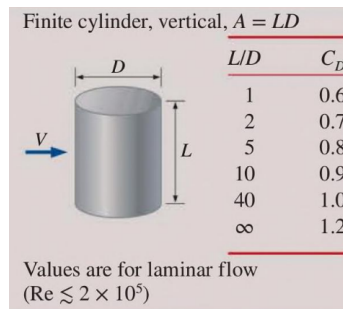


Figure 12: Drag Coefficients of a Cylinder Perpendicular Flow

When determining the drag coefficient for a plate when its parallel to the flow, the plate will experience various regions with laminar, transitional and turbulent flow shown in Figure 13 from (Cengel and Cimbala, 2018, p.629). Due to its streamline shape the drag coefficient will mostly be equal to the friction coefficient as there is very little pressure drag.

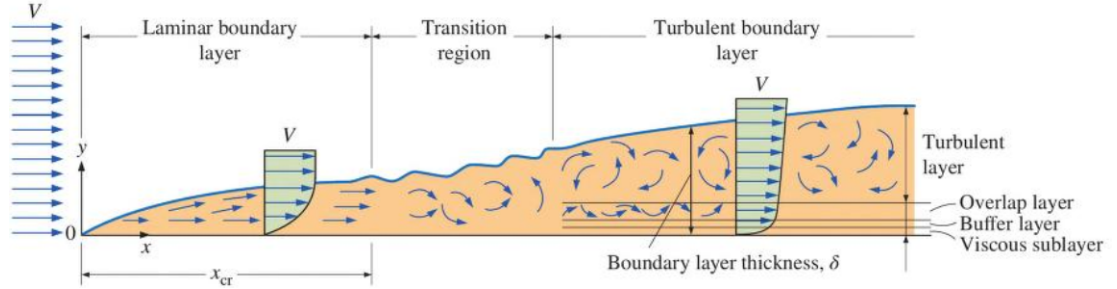


Figure 13: Flow Over a Parallel Plate

Due to the fluid over a parallel plate having different flow states it will experience differing Reynolds values. To determine the average friction coefficient over the entire plate, Equation 6 can be applied for laminar flow (Cengel and Cimbala, 2018, p.631):

$$\text{Laminar: } C_f = \frac{1.33}{Re^{1/2}} \quad Re_L \lesssim 5 \times 10^5 \quad (6)$$

$$\text{Turbulent: } C_f = \frac{0.074}{Re^{1/2}} \quad 5 \times 10^5 \lesssim Re_L \lesssim 10^7$$

### 2.7.5 Added Mass

Added mass refers to the additional mass an object appears to have due to the inertial forces in a fluid system. These inertial forces arise because the object must accelerate the surrounding fluid, which cannot occupy the same physical space as the object. In simple terms, added mass results from the mass of the fluid that needs to be displaced for the object to move. The degree of added mass depends on the type of fluid, as well as its characteristics and state (Added mass, nd) (Wallis, 2011).

### 2.7.6 Turbines

A centrifugal pump is a pump that uses a shaft with blades to accelerate water from the center of rotation out of an impeller to turn mechanical rotational energy into kinetic energy for the fluid (Hydraulic Institute, 2018), as shown in Figure 14 gathered from (Cengel and Cimbala page, 2018, p.276).

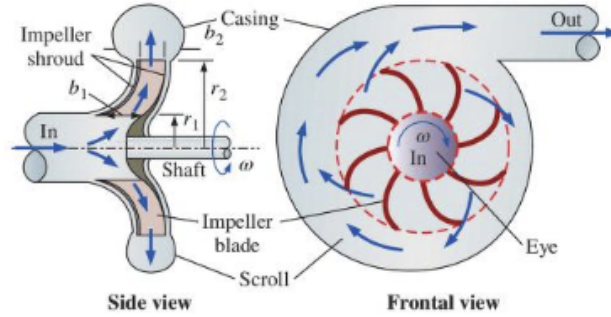


Figure 14: Centrifugal Pump

A centrifugal pump can be analyzed by choosing the swept volume of the impeller as the control volume. Then it is possible to calculate the needed shaft torque for a given mass flow and angular velocity as well as the radii of the swept area and the shaft. This is applicable for an ideal case where the velocity of the fluid is equal to the velocity of the impeller blades. The ideal case is shown in Figure 15 gathered from (Cengel and Cimbala, 2018, p.276).

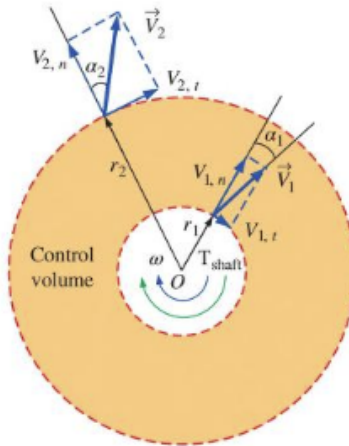


Figure 15: Centrifugal Pump Control Volume

Based on the previous assessments, the ideal version of Euler's turbine equation, Equation 7, can be used to describe the turbine.

$$T_{shaft,ideal} = \dot{m}\omega(r_2^2 - r_1^2) \quad (7)$$

## 2.8 Mechanics of Materials

This section focuses on the study of strength and mechanics of materials, which is a fundamental pillar in the engineering and architectural disciplines, encompassing the behavior of materials under various loading conditions.

### 2.8.1 Mechanical Stress

Stress is defined as force per unit area, and a material will experience stress and strain responding to the type of load acting upon it. The types of stress are shown in Tables 2, 3 and Figure 16 taken from (Strength of Materials, n.d.).

Normal stress	
Axial stress	$\sigma = \frac{F}{A}$
Bending stress	$\sigma_b = \frac{M_y}{I} \cdot z$

Table 2: Normal stresses

Shear stress	
Transverse stress	$\tau = \frac{F}{A}$
Torsional stress	$\tau = \frac{T \cdot r}{I_p}$

Table 3: Shear stresses

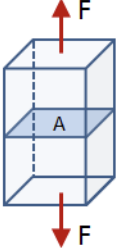
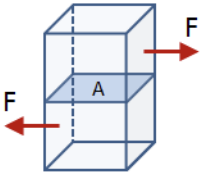
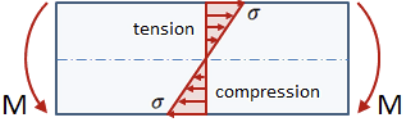
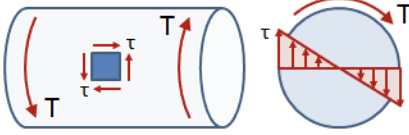
Loading Type	Stress Type	Illustration
Axial Force	<ul style="list-style-type: none"> <li>Axial Stress (general case)</li> <li>Tensile Stress (if force is tensile)</li> <li>Compressive Stress (if force is compressive)</li> </ul>	
Shear Force	Transverse Shear Stress	
Bending Moment	Bending Stress	
Torsion	Torsional Stress	

Figure 16: Types of Stress



### 2.8.2 Combined Stresses

The combined stresses on an element is an amalgamation of normal and shear stresses as a result of an external load. The stress state at any point in the element can be decomposed into a combination of the stresses shown in Figure 17, in both two and three dimensions.

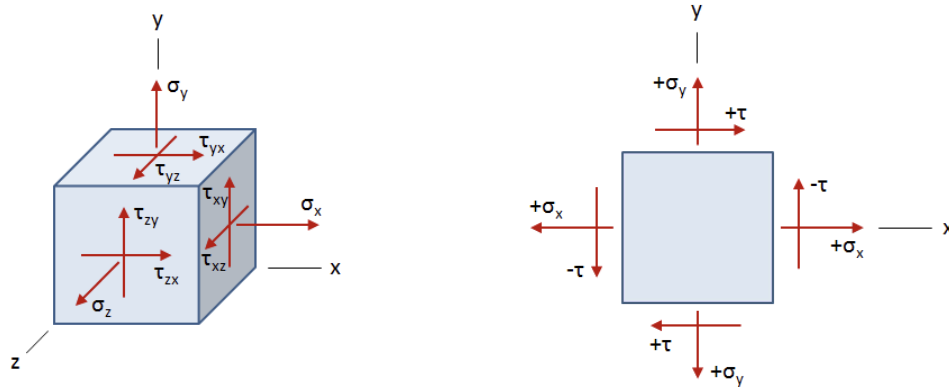


Figure 17: Combined Stresses

From the cooperation of stresses, it is possible to predict the behavior of the material under certain loads. The combined stresses all join to formulate the equation for von Mises stress in Equation 8 from (Hurlston, 2021, Section "von Mises Stress" ). The value from this equation is then compared to the yield stress of the material to determine if the material can withstand the specific loading conditions.

$$\sigma_{VM} = \sqrt{\frac{1}{2}[(\sigma_{xx} - \sigma_{yy})^2 + (\sigma_{yy} - \sigma_{zz})^2 + (\sigma_{zz} - \sigma_{xx})^2] + 3(\tau_{xy} + \tau_{yz} + \tau_{xz})^2} \quad (8)$$

### 2.8.3 Beam Theory

Beams are common elements in all types of applications, and it is important to understand their uses and how they respond under different conditions. The orientation of beams are arbitrary, but the most common orientation of beams and their simplified version in two dimensions are illustrated in Figure 18, as well as the three most common bearings being the roller support, hinge support and fixed support.

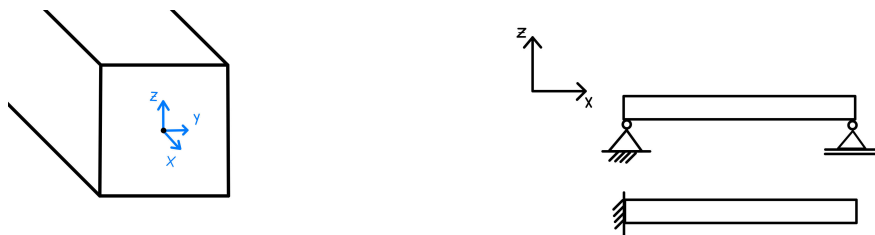


Figure 18: Beam Coordinates and Their Simplification in 2D

---

### 2.8.4 Shear Force and Bending Moment Diagrams

Shear force and bending moment diagram are used to analyze beams under different loading conditions to locate where failure is most likely to occur due to stress concentrations. The diagrams for common loading conditions are predefined and can be found from different sources. However, if one is unable to find a diagram to a specific loading condition, the diagrams must be created by hand or through other resources. To create these diagrams, an imaginary cut is made in the beam an unknown length  $x$  away from a support followed by formulating functions of the length of the beam expressed as  $x$ , based on the reaction forces in the supports of the beam.

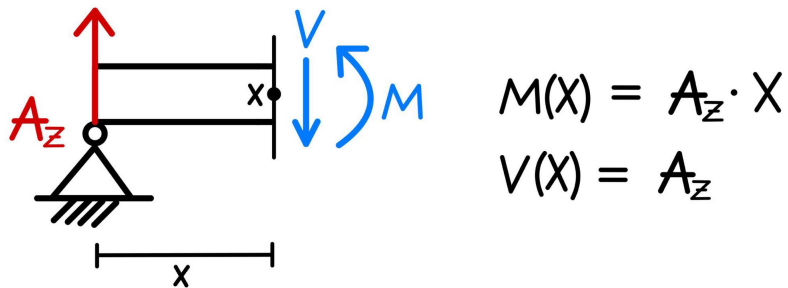


Figure 19: Beam with an Imaginary Cut

By moving along the length of the beam and creating a cut before each applied load, a new function of  $x$  for each length unit of the beam can be constructed as shown in Figure 19, and used to locate the point of interest where the stress concentrations are the greatest.

---

## 2.9 3D Modelling and Simulations

In modern engineering, the use of advanced software tools is beneficial for creating, analyzing, and testing complex designs and systems. The following subsections provide an overview of some of the key software tools utilized in these processes.

### 2.9.1 Computer-Aided Design software

CAD, short for Computer-Aided Design, replaced the traditional manual design and drafting processes with a computer-aided 3D modeling software solution. This software allows designers to witness their own design in real time, allowing for a better and more reliable development, modification, and optimization process. By providing instant visual feedback, designers can easily craft more accurate designs and enhancing overall design quality. CAD software elevates designers efficiency and creativity. CAD software excels not only in designing but also in animation and visually representing movement of connected parts.

### 2.9.2 Finite Element Analysis

Finite Element Analysis involves simulating various conditions applied to a design to assess its response. This is achieved through computer-based simulations that solve complex mathematical equations. The simulation process divides the entire model into smaller elements or nodes, forming a mesh. Based on defined conditions, diverse interactions between the model and its structural behavior can be analyzed. By carefully selecting the parameters for the model, it is possible to predict the structural behavior of the real model with a high degree of accuracy. FEA software also excels in visualizing structural behavior with color grading of critical areas.

### 2.9.3 Software and Hardware Simulation

Wokwi is an online platform that provides a simulated environment for prototyping, and testing electronics projects. Wokwi provides seamless integration with the Arduino platform, allowing users to write and upload Arduino code. It offers a range of features and tools aiding with developing and experimenting with hardware projects without the need for physical components or equipment.(Wokwi, 2023)

---

## 3 Method

The following section outlines the methodologies, thought processes, and assumptions applied throughout this project, spanning from product development to the simplifications made in mechanical analysis. It discusses how the system's limiting factors were identified, the role of prototyping, the development of software, and the implementation of the full-scale model.

### 3.1 Product Development

Designing the solutions for the full-scale model was done on the basis to build upon the in-place rail system rather than incorporating a new foundational movement setting for the mechanism. Mechanical solutions were developed with easy implementation of automatic motion controls in mind.

### 3.2 Electronics

Electrical components were selected to meet the system's solutions and requirements. The finalized calibration system incorporated various electrical components, including actuators, motor drivers, transformers and a microcontroller. Each of these components was carefully evaluated based on the value they added to the overall system.

### 3.3 Analysis

Implementation of the system's actuators required extensive analysis of reaction forces affecting the plate's movements. A force analysis was performed to determine the system's torque requirements at operating speeds. Torque requirements lay the foundation for the specified motors used in the mechanism. The calculations regarding the forces acting upon the system have excluded contributions of added mass and vortex shedding. These phenomenon introduce significant complexity to the analysis, which falls outside the scope of this project. The focus of the calculations were to support the primary objective of creating an automated system. However, to address the added mass to some extent, all accelerations were slightly reduced.

---

### 3.4 Drag Coefficients

Given that the mechanism is designed to operate at slow velocities, it was assumed that all motion through water could be considered equivalent to the plate remaining stationary while the water flows towards it. The flow of water is external and its Reynolds value will differ based on the plate's orientation to the flow. Therefore, the Reynolds value given for linear motion of the plate with a position perpendicular to the flow of water was set to account for all motions of the plate, excluding the pole. The characteristic length used in the Reynolds value is given as the square root of cross-sectional area and the water density  $\rho$  is set to  $998.21 \text{ kg/m}^3$  as the temperature of the water is approximately  $20^\circ\text{C}$ . Consequently the dynamic viscosity  $\eta$  equates to  $0.0010016 \text{ PaS}$  at  $20^\circ\text{C}$  (The Engineering Toolbox, 2004).

$$R_e = \frac{998.21 \text{ kg/m}^3 \cdot 0.20 \text{ m/s} \cdot \sqrt{0.7 \text{ m} \cdot 0.7 \text{ m}}}{0.0010016 \text{ PaS}} = 139\,526.1581$$

Applying the given Reynolds value, the drag coefficients for the system movements can be determined. The given coefficients are applied when calculating the driving torque in Section 5.

#### **Drag Coefficient of the Plate Perpendicular to the Flow**

Since the Reynolds number is greater than  $10^4$ , the drag equation applies to the linear movement and can be calculated based on the assumption that  $F \propto v^2$ . Using Equation 5, the drag coefficient of the plate when its perpendicular to the direction of flow can be set to  $C_{Dy} = 1.14$ .

#### **Drag Coefficient of the Plate Parallel to the Flow**

Looking at the plate when it moves parallel to the flow of water, its surface area is greatly reduced and its shape is more streamlined. With the assumption that the entire plate experiences a laminar flow excluding the different flow regions formed at differing lengths, the Reynolds value will account for all areas of the plate. Due to  $R_e < 5 \times 10^5$ , the drag coefficient can be found using Equation 6 where the  $C_{Dx} = 3.56 \times 10^{-3}$

#### **Drag Coefficient of the pole**

Using the pole's diameter as the characteristic length gives a Reynolds number smaller than the criteria of  $2 \times 10^5$ . The drag coefficient can be found using Figure 12 where the length of the pole is defined as the distance from the water surface to the top of the plate, measuring 0.89 m. Using Table 1, the diameter is set to 0.025 m making  $\frac{L}{D} = 35.6$ . To round up, the drag coefficient of the pole can be set to  $C_{Dp} = 1$ .

---

### 3.4.1 Driving Force Assumptions

Specific motors utilized by the calibration system were selected based on motor capabilities, specifically the output torque. The necessary torque requirements for the motion of the plate were determined based on the drag force acting against the movement direction. Assessment of the forces acting upon the plate and pipe have neglected the force contribution imposed when the plate moves closer to the walls.

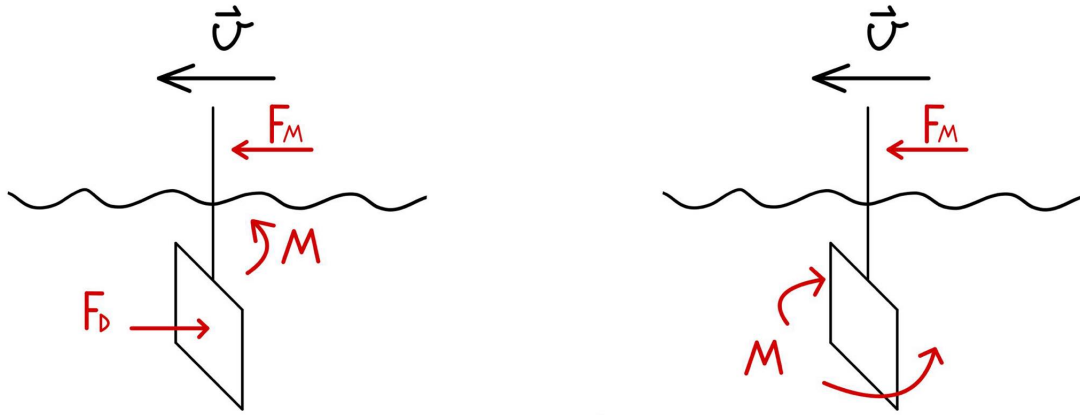


Figure 20: Forces and Moments on the System

When the carriage moves the plate linearly, it will experience a drag force opposite to its direction and proportional to its velocity shown in Figure 20. Due to the previously established assumptions in which the plate and pole are viewed as stationary while the flow is an approaching upstream velocity, the drag force equation 4 can be applied. To calculate the necessary driving force it was assumed that the drag force was equal to the force driving the carriage at all times. This meant that any tilt, shown in Figure 21, of the plate was disregarded. Additionally, any friction forces between the carriages and rails were disregarded.

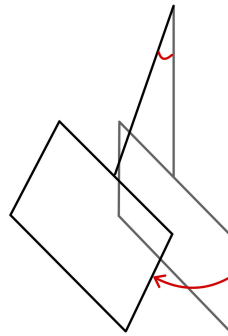


Figure 21: Tilt

---

### 3.5 Rotational Torque Assumptions

The rotational movement could not be calculated as simply as the linear movement, and due to the uncertainty when dealing with fluid mechanics, two approaches were applied to determine the rotational torque requirement. The two results were then compared to establish a greater approximation. The two approaches have been named *Worst-Case approach* and *Euler turbine approach* and their assumptions are laid out in the following section and applied in Section 5.

#### 3.5.1 Worst-Case Scenario Assumptions

As the plate rotates around its symmetry axis, it encounters a drag force from the pressure differential on both sides resisting the motor's torque. The pressure intensifies toward the ends of the plate., shown as the red area in the following figure, where the pressure drag at the center is zero. This is due to the increase in linear velocity toward the ends of the plate.

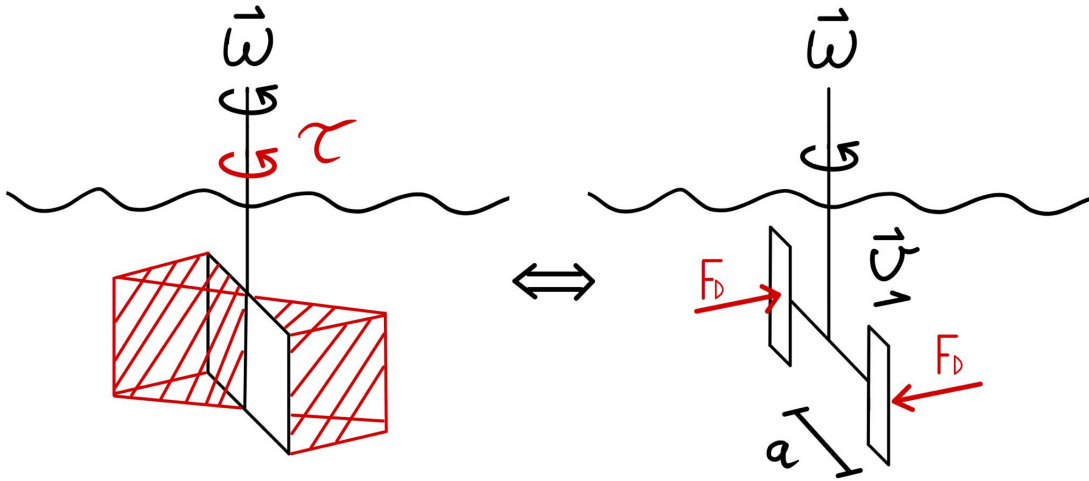


Figure 22: Equivalent Scenario

To manage the complex fluid dynamics, the rotation was simplified by assuming a worst-case scenario illustrated to the right in Figure 22. The idea of the worst case scenario was to create an equivalent situation to approximate the drag force of the rotation. Firstly, the plate was split into two hypothetical plates, each with half the area of the actual plate. The pressure drag on the two plates was considered uniformly distributed and identical to that of linear movement at the corresponding velocity at a distance, denoted as  $a$ , from the center of rotation. This will create an exaggerated torque due to the concentration of the forces towards the ends of plate, but will also be slightly reduced since the water is no longer restricted to only flow around one side of the plate as opposed to the actual scenario. The drag coefficient in this scenario can be calculated from Equation 5 where  $L = 0.35\text{ m}$  and  $D = 0.7\text{ m}$ , giving a drag coefficient  $C_{C,R}$  of 1.15.

---

### 3.5.2 Euler Turbine Approach Assumptions

The second approach to determining rotational torque involves assuming an ideal Euler turbine equation where the tangential fluid velocity is equal to the angular velocity of the turbine blades or in this case the rotating plate. All fluid is dispelled tangentially to the radius created by the plate's rotation.

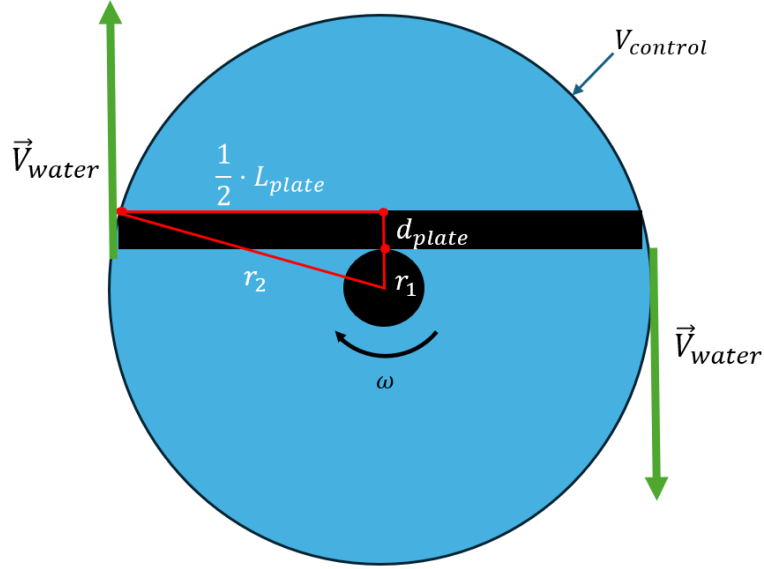


Figure 23: Control Volume Assumption

Figure 23 illustrates the thought process of the plate's rotation seen from above. The  $V_{control}$  is given by finding the circular area from the radius  $r_2$  minus the pole's radius  $r_1$ , and multiplying it with the height of the plate  $h_{plate}$ , shown in Equation 9.

$$V_{control} = h_{plate} \cdot \pi \cdot h_{plate} \cdot (r_2^2 - r_1^2) \quad (9)$$

To determine the mass flow  $\dot{m}$  it was assumed that all the water within the control volume is dispelled when the plate has rotated 180 degrees or  $\pi$ . By dividing the angular velocity  $\omega$  with the rotation  $\pi$ , and then multiplying it with the water density  $\rho$  and the control volume  $V_{control}$ , the outgoing mass flow can be determined.

$$\dot{m} = V_{control} \cdot \rho \cdot \frac{\omega}{\pi}$$

Applying this mass flow equation into the ideal Euler Equation 7 gives the following combined Equation 10.

$$\tau_{shaft} = V_{control} \cdot \rho \cdot \frac{\omega^2}{\pi} \cdot (r_2^2 - r_1^2) \quad (10)$$



---

### 3.6 Limiting Factor

It is also important to consider the limits of the system. The velocity will be a significant factor where small velocity changes can result in damages to the plate or its attached metallic pole due to the forces generated during movement. Based on the kinematic representation in Figure 20, it is possible to make mechanical simplifications to test its limits.

#### 3.6.1 Linear Movement Assumptions

The transversal motion was not focused on when ascertaining the limit of the system, as the greatest reaction forces are generated when the plate moves perpendicular to the flow, as shown in Figure 20. To find out how much stress is needed for the pole to yield, a few assumptions had to be made. Firstly, the pole was considered a cantilever beam as shown in Figure 24. This is because the pole is only fastened at one point that counteracts the moment from external loads, and because it can rotate around its longitudinal axis.

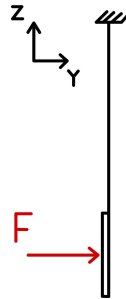


Figure 24: Cantilever Beam

The second assumption is that the force from the pressure drag can be exchanged for two forces that generate an equivalent moment around the plate's vertical axis. To make calculations simpler, the horizontal supports at the center of the plate shown in Appendix A, Figure A3 can be disregarded, and the plate can be considered a simple support beam with one point of support along the centerline of the plate parallel to the pole. Since the horizontal supports resist the moment created by the drag force, this will result in an exaggerated moment at the center of the plate shown in Figure 25. The resulting calculations utilized theoretical calculations and simplifications based on basic fluid mechanics and failure theory. Abaqus modelling software was used to solidify the results.

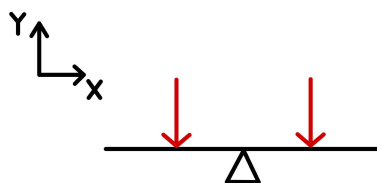


Figure 25: Representation of the Plate Viewed from Above

---

### 3.6.2 Rotational Movement Assumptions

During rotation, the plate will cause torsion in the pole, but due to the roller bearing attachment at the top, the stresses in the pole are reduced significantly during continuous rotation; however, stresses may occur during starting due to the added mass previously mentioned in Subsection 2.7.5, and during stopping due to existing currents around the plate. Both of these cases will generate torsion in the pole, but the effect of added mass at startup is difficult to account for accurately, and was therefore assumed to have similar stresses as when rotation stops. Regarding the stopping, the motor will effectively work as a fixed support, resisting rotation from the water currents, where the method used to determine the necessary rotational torque can also be utilized to find the maximum torsion. It was also assumed a state of pure torsion.

### 3.6.3 Material Strength

Optoscale did not provide the material strength specifications for the pole and the plate. It was presumed that the tree-plate was made of oak, which has a typical yield stress of 70 MPa, as cited in (Material properties, n.d.). For the pole, the selected material was NS14350, a stainless steel variant commonly utilized in the manufacture of plates and pipes with a yield stress of 210 MPa, as detailed in (TEKNISKE TABELLER, 2002).

To solidify the results of the manual calculations, it was decided to use Abaqus to verify the results. The maximum force from the manual calculations was used as input to compare the maximum stresses of the material.

---

### 3.7 Software Development

To control and define the system, a code was written in the Arduino programming language. To aid in visualization of electrical components, the Wokwi webpage was employed. Using Wokwi, a simplified hardware of the final construction could be simulated, which greatly aided in testing and developing the code. The simulated hardware can be seen in the following Figure 26 gathered from (Wokwi, 2024).

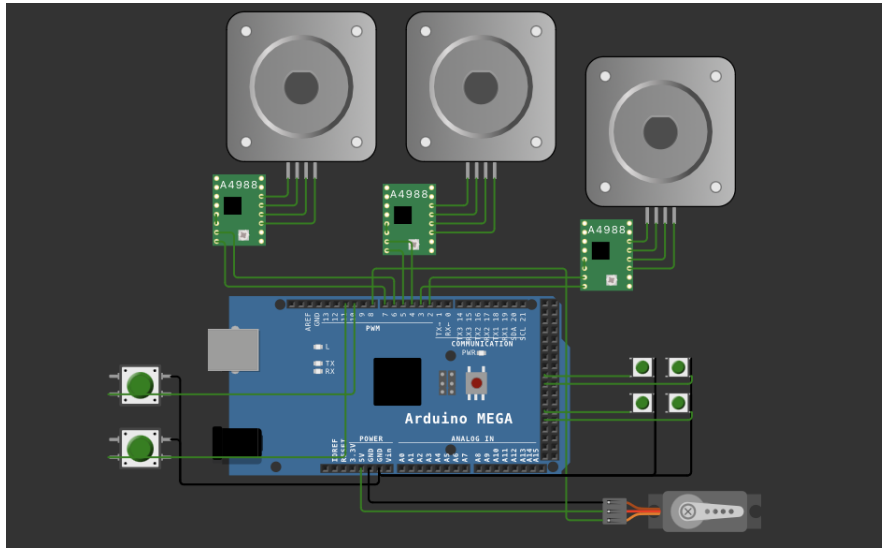


Figure 26: Wokwi Simulation Model

The system consisted of many working parts and due to its complexity, it required a way of testing before employment. For this reason, a critical functional prototype (CFP) was constructed to validate the developed concepts and to simulate the path of the system on a smaller scale. "CFP is useful when developing product service systems. Focusing on testing the critical component of the system first, as opposed to the whole system" (Freyer et al.,(2022), p.1). The CFP was created using 3D-printers and employed similar but smaller scale electrical components used in the final model shown in Figure 28. The CFP testing allowed for similar testing as the Wokwi webpage, however it allowed for greater visual understanding compared to that of wokwi. Figure 27 shows an assembled fusion model of the CFP. More about how and why the prototype was made can be found under Appendix F.

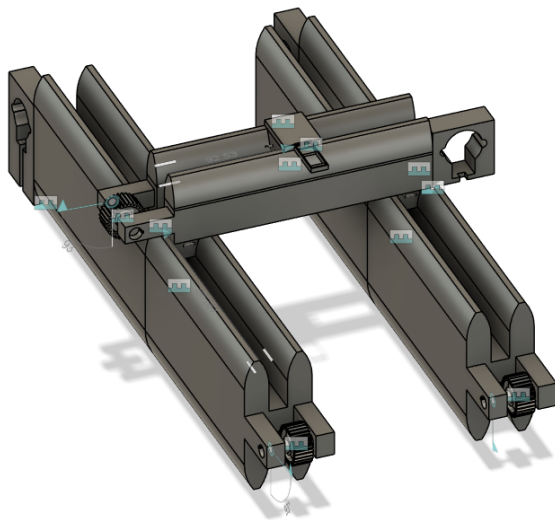


Figure 27: Assembled Fusion Prototype

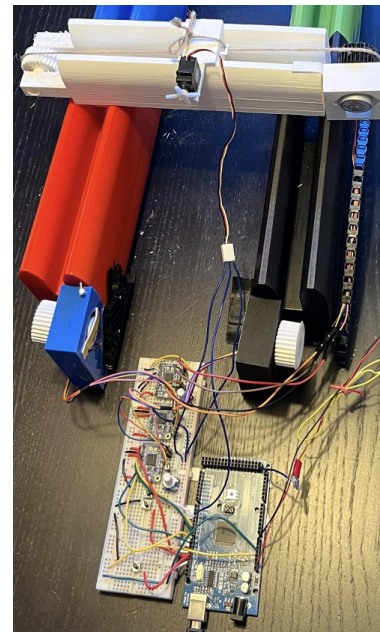


Figure 28: Prototype

The path for the plate's movement was predetermined by the company prior to project start. Detailed calculations, such as optimal operational speeds and motor specifications, were key factors incorporated into the software architecture. Furthermore, criteria such as the option to control the system with a manual control and activation buttons via a dedicated control panel, were seamlessly integrated into the software framework.

Initially, operational speeds of approximately 20cm/s and 10 degrees/s were given. These values served as estimations and would later be adjusted through tuning and testing of the mechanism using a full-scale model with a bioscope. The software code was designed with these approximate values in mind.

### 3.8 Full-Scale Model Implementation

To ensure seamless implementation of electrical and mechanical components, safety protocols was important to consider, especially considering that power tools and a soldering iron was utilized. These essential tools were accessible within Optoscale's laboratory. The system utilized voltages up to 230V. Extensive wiring was performed, and due to the high-voltage nature of the work, an electrician from Optoscale supervised the process. Additionally, a fuse switch was added as a safety precaution.

The implementation of the system entailed careful soldering of cables, drilling, assembling and wiring. Tasks undertaken by the project team under the guidance of experienced technicians from Optoscale. This collaborative effort ensured quality work, but also the safety of the system's electrical and mechanical infrastructure.

---

## 4 Concept Development

This section will be discussing the concept for the creation of the mechanism. The chapter will discuss conceptual solutions for linear and rotational motion of the plate where differing solutions will be evaluated. Additionally, adequate components for the solutions will be mentioned.

### 4.1 Foundation

To understand future discussions and calculations, the existing rail system will be reintroduced using new terminology. The company's rail system consists of three carriages traversing the pool's length and width, designated as the y-axis and x-axis, respectively. The two carriages moving along the y-axis are referred to as the y-axis carriages, while the carriage traversing the pool's width will be referred to as the x-axis carriage. The following Figure 29, illustrates the existing rail system using the updated terms.

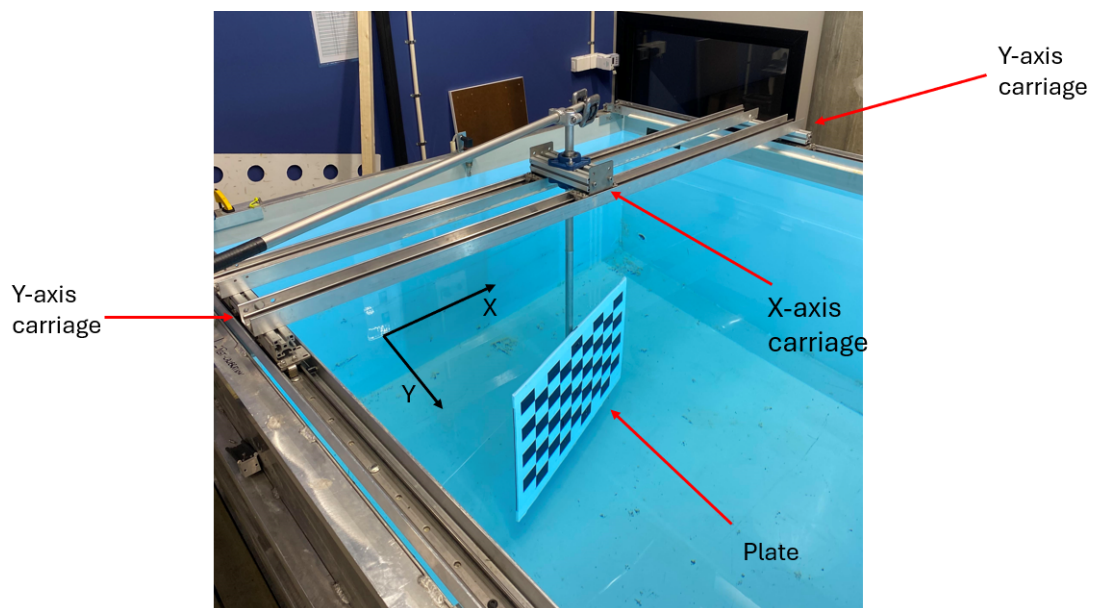


Figure 29: Rail System with Updated Terminology

---

## 4.2 Linear Motion

The in-place rail system used roller bearings and allowed for consistent motion along the x- and y-directions. Due to the company constantly needing to use the rail system a replacement of the rail system to a linear guide rail or a roller screw was not an option. Additionally, due to the rail system being reliable and well-integrated onto the pool, the only option was to build upon the system rather than incorporating a new foundation. To build upon the rail system in such a way that would allow automatic motion would require motors. Before the selection of motors, it is crucial to understand how they should be integrated to move the system.

When researching similar constructions it was apparent that mechanism used for such operations are considered gantry mechanisms. To construct a gantry mechanism without altering the in-place rail system, a timing belt solution was drawn. The solution involves connecting motors to timing pulleys, which facilitate the movement of belts attached to the system's carriages, thereby driving the carriages. This solution would not require changes to the rails, however, it would require adding timing pulleys at the far ends of the pool and furthermore connecting pulleys and carriages via belts. This solution would allow a motor to be connected to, and rotate a pulley via an axle which then would induce movement in the system.

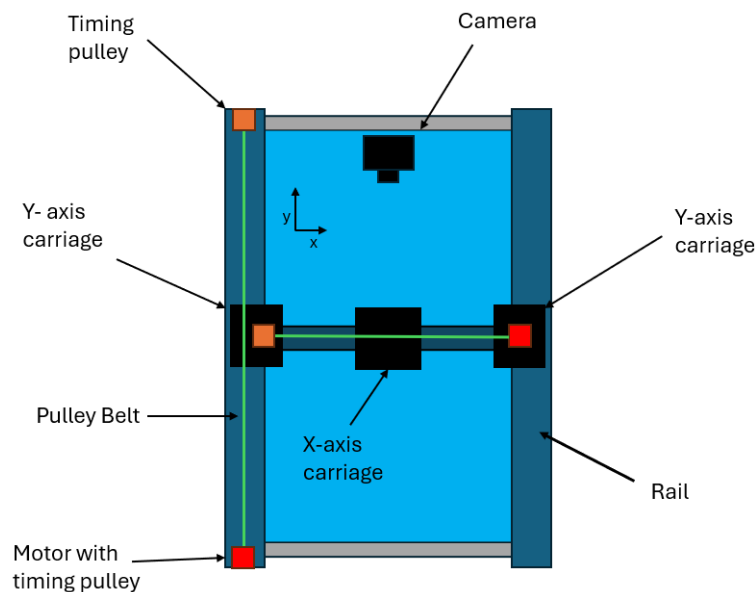


Figure 30: Timing Pulley Concept

Figure 30 illustrates the timing pulley concept for the linear motion of the calibration system. As shown in the image, timing belts in green can be placed along both rail directions, with motors integrated along the far ends away from the camera. Additionally, a motor would be integrated onto one of the y-axis carriages. Opposite of the motors are their respective timing pulleys rotating around an axle without a motor.

---

Utilizing pulleys for the linear motion is an option for easy integration of motors and other electrical components along the pools length and the rails connecting the y-axis carriages. This solution has drawn inspiration from 3D printers, which are a form of gantry mechanism that utilize motorized pulley solutions for the movement. The pulley solution is great on small scale 3D printers, however on the larger calibration system a few challenges must be addressed.

#### 4.2.1 Movement Imbalance

When moving the system along its y-axis it is crucial that both carriages move cohesively. If both carriages move with differing speeds to one another, an imbalance will occur. An imbalance between the carriages can lead to damages to the rails or carriages and inconsistent motion, and consequently lead to more forces required to move the system. If there only is one motor applying force on one carriage, said carriage will accelerate before the other. The imbalance between the driven and held-back end can cause vibrations or "humping" motions.

During discussions with Optoscale's technicians, it was recommended to induce motion along the y-axis by applying equal force to both of y-axis carriages. Furthermore, the group tested the system for carriage imbalance by applying force on one side to study the motion of the system. It was apparent that the motion of the system experienced unbalanced movement. Figure 31 illustrates the discussed carriage imbalance. In this figure the pulley system for the x-axis has been removed to emphasize the y-axis imbalance.

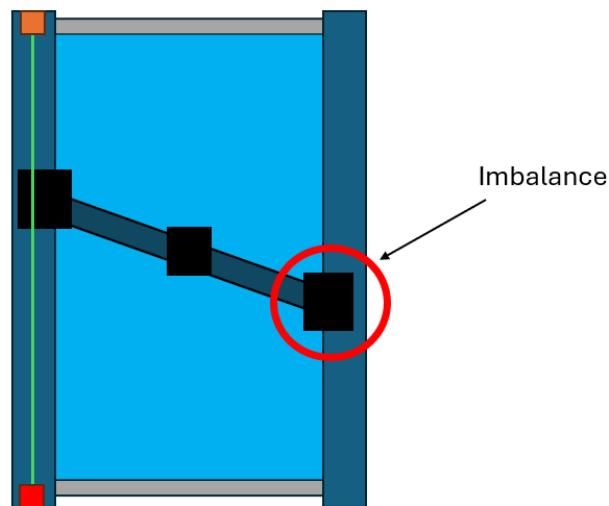


Figure 31: Carriage Imbalance

The movement of the x-axis utilizes one carriage contrasting the set of two on the y-axis; however, the x-axis carriage as shown in Figure 29 moves on two sets of rails. Experiencing an imbalance when moving along the x-axis is unlikely as the length between the rails are insignificant. An evaluation was conducted by pushing the x-axis carriage by only applying force on one side of the carriage, similar to the y-axis carriage test. From this test no apparent imbalance was experienced.

---

### 4.2.2 Two-Pulley Solution

To address the imbalance issue along the y-axis motion, a set of timing pulley pairs must be integrated along both lengths of the pool. Each timing pulley pair will be connected to the carriages via separate belts. These pulleys must be rotated simultaneously to ensure cohesive movement of the carriages.

### Large Axle Solution

The project team concluded with two options in which a timing pulley pair could be rotated simultaneously. The first considered solution was attaching an axle that traverses the entire width of the pool. Said axle would then be driven by a single powerful motor and two pulleys could be attached to it. By utilizing this solution, one axle would rotate both pulleys which in return would move both attaching belts simultaneously. Furthermore, this solution would only require one long axle on the motorized side as the other side could use pulleys on separate axles.

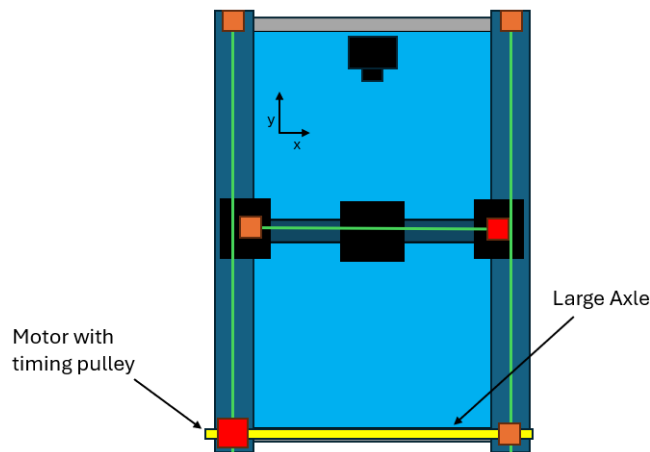


Figure 32: Large Axle Concept

Illustrated in Figure 32, the large axle solution, which can be applied to solving the imbalance problem, would require an axle of about two meters. This case would require a unique axle which would need to be tailored specifically to this solution due to its size.



---

## Dual Motor Solution

The second solution in solving the imbalance problem would be to attach two motorized timing pulleys at opposing corners along the width of the pool as shown in Figure 33. In this manner, each y-axis carriage would be pulled by a separate motor. The motors would need to be synchronized to ensure cohesive output. Synchronizing the motors would result in both carriages being moved simultaneously and subsequently removing imbalance between the carriages.

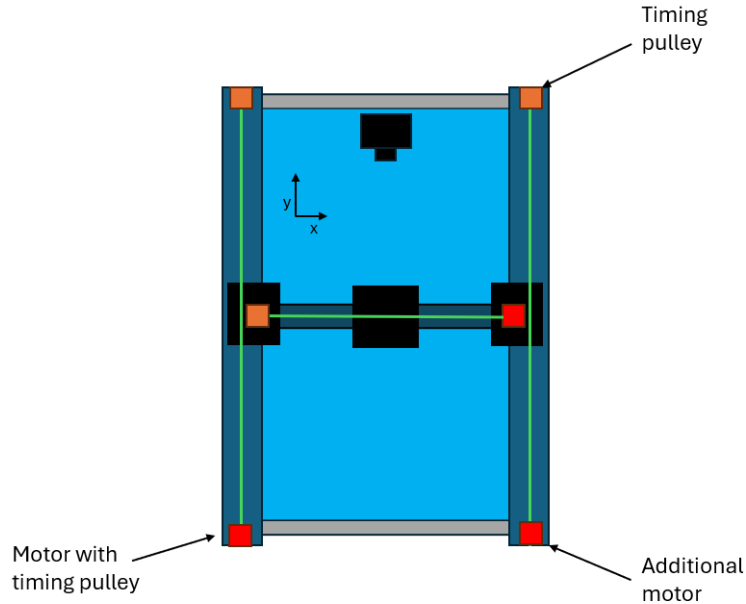


Figure 33: Dual Motor Solution

The dual motor concept is illustrated in Figure 33, which is similar to Figure 30, but includes an additional motorized pulley on the right. This solution can pose challenging as it heavily relies on the synchronization of the two y-axis motors. Uncoordinated movements between the pairs would result in an imbalance. Lastly, as the x-axis motion also requires a motor, the system will therefore require to synchronize three motors for the linear motion. However, the use of two motors along the y-axis could distribute the necessary torque among the two motors, resulting in less strain on each motor.

---

To determine which of the two solutions would be implemented, it was decided to base the evaluation on available resources. Firstly, when analyzing the large axle concept, the determining factor was the large axle itself. A specific axle such as the one mentioned was not easily available for purchasing. As mentioned previously, it would require customizing an axle of about two meters and a diameter fitting both pulleys and a motor. This would be time consuming compared to purchasing pre-made components and possibly more expensive. The purchasing of finalized components would not be a problem as long as it is within the project budget. Secondly, when analyzing the dual motor solution, synchronizing multiple motors is a necessary task that might appear challenging. However, this complexity is mitigated by the inherent requirement to synchronize the x, y, and rotational motors.

Consequently, the addition of an extra motor along the y-axis was not deemed problematic. Moreover, all necessary components for the dual motor solutions were less challenging to acquire compared to the long axle solution. The project team concluded that the dual motor solution would be the best foundation for movement along the y-axis.

---

### 4.3 Rotary motion

Described in Section 1.2, rotation of the plate is achieved by rotating the connecting pole passing through the x-axis carriage. The pole and plate are connected, therefore a rotation of the pole subsequently rotates the plate. As the pole is more accessible to work on rather than the plate itself, a solution would aim to rotate the pole.

The solution for the rotary motion should consider the system's rotational criterion, which is an approximate rotation angle of 30 degrees to both sides. However, since this rotation angle is approximate and can only be verified through testing, the implemented solution should be capable of performing a full rotation.

#### 4.3.1 Rotary Motion Solution

To achieve the rotary motion, a gear mechanism was implemented. This approach involved using a pair of gears: one smaller gear and one larger gear. The smaller gear is attached to a motor, which drives the larger gear to rotate the pole. This solution offers the advantage of increased torque by applying the principles outlined in Section 2.6.

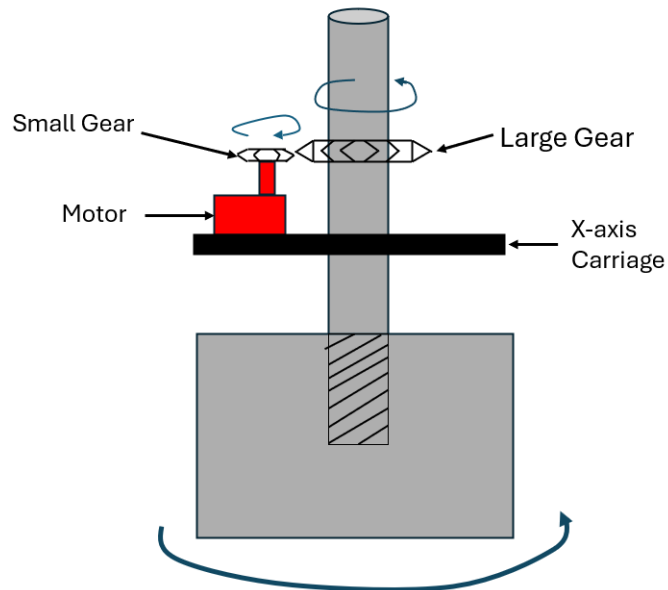


Figure 34: Rotational Motion Concept

---

Figure 34 illustrates the gear solution, where one can see a motor placed onto the x-axis carriage with a small gear head attached to it. The motor in return rotates the large gear head and subsequently the pole. Attaching a gear onto the pole would require a clamping hub, as shown in Appendix A, to be mounted and furthermore attach the large gear onto the hub. Additionally, the placement of the motor is challenging as there is no clear option to integrate it onto the x-axis carriage as displayed in Figure 35. Therefore, it is essential to design an enclosure for the motor and other electrical components discussed in Section 5, that also ensures precise alignment of the gears. The enclosure should be precisely secured to the carriage and fully cover all motor cables to protect them from potential water exposure.

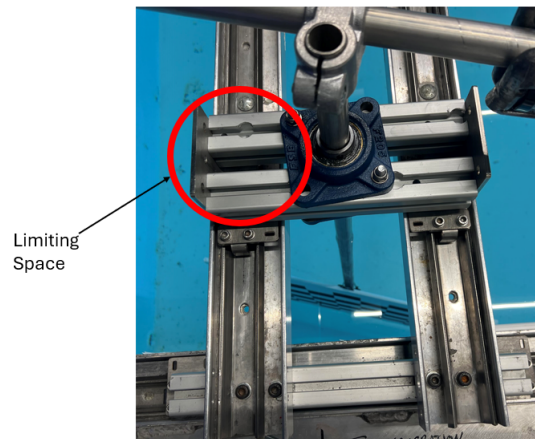


Figure 35: Limiting Space

#### 4.4 Cable Chains

One important thing to note about the discussed solution is the cable management due to the x-axis motor and rotational motor being attached to two of the system's carriages. Moving the carriages during automatic operations would consequently result in the two motors being moved along with the carriages. For this reason, two sets of cable chains need to be implemented. Firstly, one larger cable chain is to be attached along length of the pool to supply cables to the x-axis motor and rotational motor. Secondly, a cable chain must be placed along the x-axis rails to supply cables only to the rotational motor.

#### 4.5 Interface Concept

The calibration system must have an interface that can control its movements and automated process. This was specified as an important criterion of the project. For this reason the interface could have one joystick, a start- and a stop button. The interface is to be fixed onto the side of the pool. Cables from the interface connecting to the system's microcontroller would be attached to the outside of the pool.

---

## 4.6 Finalized Concept

To conclude this Section, the calibration system will feature two motorized pulleys driving each of the carriages along the y-axis, facilitating cohesive motion between them. Traversing the pool's x-axis will be achieved using a motorized pulley solution implemented on the y-axis carriages. The plate's rotation will be enabled by a gear system driven by a motor housed within a secured protective enclosure. To supply power to the motors as they move with the carriages, two sets of cable chains will be utilized. Finally, the system's interface will include a simple display integrated onto the side of the pool. The illustration given in Figure 36 showcases the finalized concept for the calibration system.

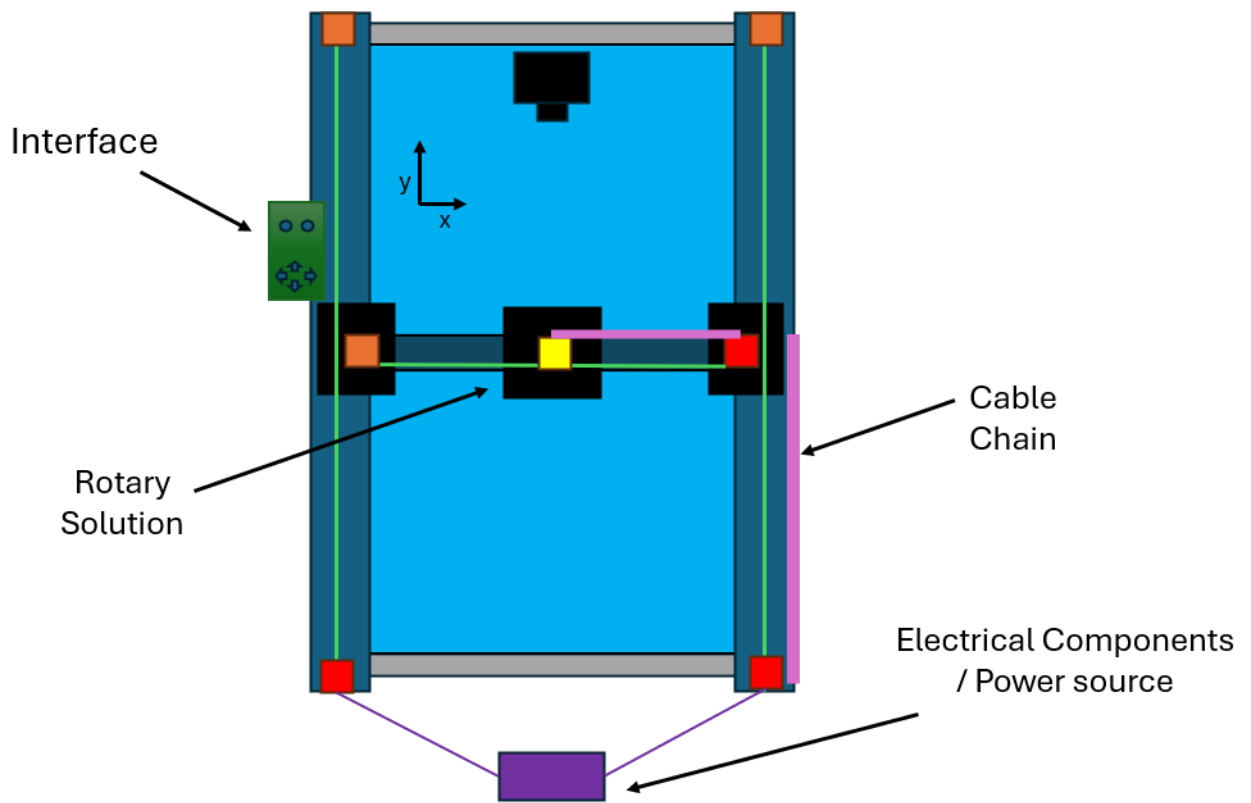


Figure 36: Final Concept Sketch

---

## 5 Components and Implementation

The following section will discuss the electrical and mechanical components and their implementation process. Previously discussed concepts and calculation assumptions will be brought up in this section to solidify mechanical- and electrical component selection.

### 5.1 Mechanical Components

The mechanical solutions which the automated system is built upon were previously established to be a Pulley solution and a Gear solution. The six following components shown in Figures 37 through 42 are the main components utilized in constructing the solutions. The pulley solution applied 16 meters of 3mm pitch timing belts, Three pillow bearings and 6 timing pulleys with 3mm pitch. The following images have excluded the motor enclosure as its more thoroughly discussed in Section 5.3.3.

#### Main Components in Pulley Solution



Figure 37: Pulley Belt

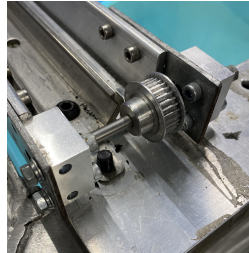


Figure 38: Pillow Bearing

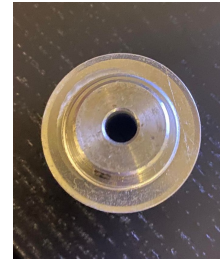


Figure 39: Timing Pulley

#### Main Components in Gear Solution



Figure 40: 20 Teeth Gear



Figure 41: 100 Teeth Gear



Figure 42: Clamping Hub

---

## 5.2 X- and Y-Axis Motors

When the plate is moved along the x- and y-axes, certain motor criteria are needed. As the plate is to be moved to predefined positions, the motors must be able to pull the system horizontally and diagonally. Additionally, this movement needs to be precise to ensure an efficient camera calibration. Moreover, it is crucial for the system to understand its positioning. Understanding the positioning of the equipment is important for safety, as incorrect positioning can result in inadequate camera calibration or the plate colliding with the walls of the pool. Before determining the motor-type, an evaluation of the system's torque is required.

### 5.2.1 X- and Y-axis Torque Requirements

When the system operates, the larger surface area of the plate will always face the camera, except when the system rotates. Moving along the y-axis will result in a considerably greater drag force contrasting that of the x-axis.

A velocity approximately equal to  $20\text{cm/s}$  was given for the plate. The necessary driving force based on this velocity was calculated by setting up Newtons second law of motion for the plate where the drag force from the plate and the pole is equal to the driving force of the belt, and the torque generated from the drag force on the plate is equal to the torque from the motor driving the belt shown in Equation 11.

$$\sum F = F_{Driving} - F_{Drag} = 0 \Leftrightarrow F_{Driving} = F_{d,y} = F_{d,plate} + F_{d,pole} \quad (11)$$

To find the torque requirement along the y-axis one can expand Equation 11 and implement the previously established drag coefficients  $C_{Dy}$  and  $C_{Dp}$  given in Section 3.4. Moreover, using the frontal area of the pole  $A_{cylinder}$  and the frontal area of the plate  $A_{plate}$ , the drag force along the y-axis can be calculated as follows in Equation 12.

$$F_{d,y} = \frac{1}{2} \cdot \rho \cdot v^2 \cdot C_{Dy} \cdot A_{plate} + \frac{1}{2} \cdot \rho \cdot v^2 \cdot C_{Dp} \cdot A_{cylinder} \quad (12)$$

$$F_{d,y} = \frac{1}{2} \cdot 998.21 \text{ kg/m}^3 \cdot (0.2 \text{ m/s})^2 \cdot (1.14 \cdot 0.7 \text{ m} \cdot 0.7 + 1 \cdot 0.89 \text{ m} \cdot 0.025 \text{ m}) = 11.60 \text{ N}$$

---

The minimum motor torque is equal to the torque created by the drag force. To express the resistance from the water as torque, the arm of the pulley that is to be rotated by the motor must be considered. The timing pulleys used in the system have a radius of  $1.5\text{cm}$ . Additionally, the torque is distributed among two motors. Hence, the minimum torque for each motor along the y-axis can be expressed as follows:

$$\tau_y = \frac{1}{2} \cdot F_{d,y} \cdot r_{pulley} \quad (13)$$

$$\tau_y = \frac{1}{2} \cdot 11.60 \text{ N} \cdot 0.015 \text{ m} = 0.087 \text{ Nm}$$

To determine the necessary torque requirement along the x-axis the  $F_{dy}$  in Equation 12 can be replaced with the drag force from the plate when it moves parallel to the flow  $F_{dx}$ . This scenario uses the previously established drag coefficient  $C_{Dx}$  and one single motor is utilized during this movement. This motion experiences little to no pressure drag and all of the drag force will be caused by the friction drag, therefore the plate's surface area used in the equation will consist of both larger surface areas which are subject to the friction force.

$$F_{d,x} = \frac{1}{2} \cdot 998.21 \text{ kg/m}^3 \cdot (0.2 \text{ m/s})^2 \cdot (3.56 \cdot 10^{-3} \cdot 2 \cdot 0.7\text{m} \cdot 0.7 \text{ m} + 1 \cdot 0.89\text{m} \cdot 0.025 \text{ m})$$

$$\tau_x = 0.514 \text{ N} \cdot 0.015 \text{ m} = 0.008 \text{ Nm}$$

Evaluating both  $\tau_y$  and  $\tau_x$  one can notice the required torque being substantially small. However, the two torque calculations are simplified calculations neglecting other fluid dynamical properties. Larger torque will be required when accelerating and at a higher operational speed. It must be noted that the speed given as  $20 \text{ cm/s}$  is approximate, and only through testing the system with a bioscope can it be confirmed. Nevertheless, the optimal calibrating speed might be higher or lower. To adjust for the speed being approximate and other fluid dynamical forces, a greater required torque along both axes is expected.



### 5.2.2 X- and Y-Motor Selection

To achieve the motion criteria previously mentioned, the chosen motors for the x- and y-movements of the mechanism were stepper motors. This is largely due to their ability of precise positioning, low speeds and high torque. Other gantry mechanisms such as 3D-printers also use stepper motors.

Stepper motors “count” steps to understand their positioning. This method can in some cases fail and miscounting can occur. To add an additional safety measure to the motor positioning, a closed loop stepper motor was chosen. The closed loop function offers an additional safety layer by gathering positional feedback from the motor as it drives. Furthermore, the closed loop function allows for a greater efficiency in which it reduces the energy consumption when the system is not operating, using the positional feedback to check whether or not the motor is moving. The driver activates the motors which turns on the detent torque only when it is needed. The closed loop ensures that steps are not missed, resulting in identical movements, which fixes the imbalance problem discussed in Subsection 4.2.1.

To determine the specific type of closed loop stepper motor the previously established x- and y-torque requirements were of importance. The necessary motors needed greater torque than the estimations. Therefore, the chosen motor for all x- and y-movements was the Closed loop NEMA23 57EH76A4001 stepper motor. Additionally, each motor used a HanBuild Closed loop stepper driver 57 (CL57). This stepper motor offers a rated torque of 1.8 Nm significantly greater than the minimum requirements. The selected motors enhance the system’s capability to operate at higher velocities and with greater efficiency. An example of a motor and driver pair is illustrated in Figures 43 and 44.



Figure 43: Nema23 Stepper Motor



Figure 44: CL57 Stepper Driver

---

### 5.2.3 Implementation of X- and Y-Motors

The implementation process of the stepper motors involved placing timing pulleys with 3mm pitch onto the axles of the motors and furthermore attaching the motors to the pools frame using motor brackets. The motor brackets for the y-axis motors were L-shaped and were secured to the edges of the pool, as shown in Figure 45. The placement process involved using power tools to drill holes into the material of the pool, to further allow nuts and bolts to fasten the brackets. The following image shows one of the two implemented y-axis motors with the timing pulley.

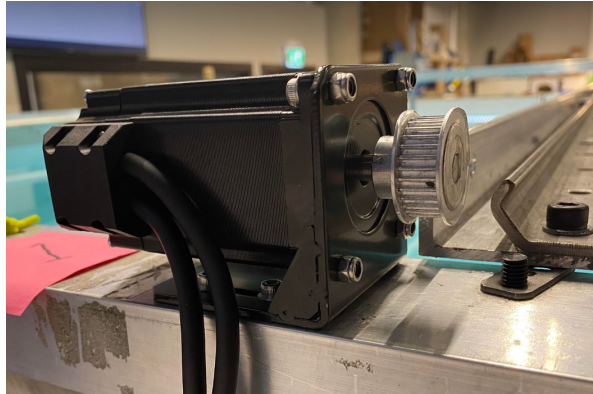


Figure 45: Implemented Y-Axis Motor

Due to the limited available area for the placement of the x-axis motor, a different type of motor bracket was applied. This motor bracket was flat, and mounted directly to the rail traversing the pool's width. The rail traversing the pool needed to be disassembled, and a pillar drilling machine was required to create the adequate bores required for the bracket placement. The motor was then tightly and carefully fastened onto the rail. Figure 46 shows the x-axis motor placement.

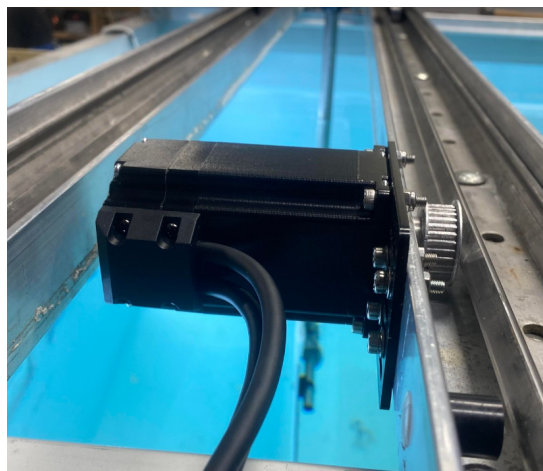


Figure 46: Implemented X-Axis Motor

---

## 5.3 Rotational Motor

The rotation of the plate also requires a motor, however with differing criteria from the x- and y-axis motors. Previously mentioned in Section 4.3 the plate should be able to make full rotations in either direction and its rotational speed should be around 10 degrees per second. Lastly, as the motor is to be placed onto the x-axis carriage its size must be considered due to the limiting space of the carriage itself. Additionally, the wires supplying power to the motor should be in cables placed in a cable chain fitting beside the x-axis rails.

The concept behind the rotational solution utilized a small gear with 20 teeth connected to the motor and a larger gear with 100 teeth fastened to the pole, increasing the torque output of the motor due to the gear ratio of 1:5. With this gear ratio, a full rotation of the smaller gear head rotates the large gear and consequently the plate by 72 degrees. The selected gears were picked due to their size fitting the limited area of the x-axis carriage.

### 5.3.1 Rotational Torque Requirements

Discussed in Section 3 two approaches were applied to calculate rotational torque requirement. The two approaches are referred to as the *Worst Case Approach* and *Euler's Turbine Approach*, where  $\tau_1$  and  $\tau_2$  represent their respective torques.

#### Worst-Case Approach to Rotational Torque

From the assumptions laid out in Subsection 3.3 it is possible to simulate a linear movement at the ends of the plate and solve the equation similar to a plate moving in a straight line as shown previously in Figure 22, and then calculating the torque it creates. By applying Newton's second law of motion for rotation, the following equation describes the torque of the plate assuming constant rotational velocity.

$$\tau_1 = 2 \cdot F_d \cdot a$$

Applying the established  $C_{D,R}$  and using Equation 4, a moment arm equal to half the length of the plate and the relationship between angular and linear velocity in rotating bodies,  $v = \omega r$ , the equation is as follows:

$$\tau_1 = 2 \cdot 0.35 \text{ m} \cdot \frac{1}{2} \cdot 1.15 \cdot 998.21 \text{ kg/m}^3 \cdot 0.70 \text{ m} \cdot 0.35 \text{ m} \cdot \left(\frac{\pi}{18} \text{ s}^{-1} \cdot 0.35 \text{ m}\right)^2 \cdot \frac{1}{5}$$

$$\tau_1 = 0.0735 \text{ Nm}$$

---

## Eulers Approach to Rotational Torque

Applying the assumptions established within Section 3.4.1, the control radius can be found using the Pythagorean theorem on the triangle drawn in Figure 23, and using the construction measurements given in Table 1.

$$r_2^2 = (r_1 + d_{plate})^2 + \left(\frac{1}{2} \cdot L_{plate}\right)^2$$

$$r_2 = \sqrt{(0.012 \text{ m} + 0.007 \text{ m})^2 + \left(\frac{1}{2} \cdot 0.7 \text{ m}\right)^2} = 0.351 \text{ m}$$

Using the established  $r_2$  and  $r_1$  and implementing them into equation 9, the control volume can be determined.

$$V_{control} = 0.7 \text{ m} \cdot \pi \cdot ((0.351 \text{ m})^2 - (0.012 \text{ m})^2) = 0.27 \text{ m}^3$$

Lastly, as the operational angular speed  $\omega$  being 10 degrees/s or in radians/s  $\frac{1}{18}\pi$ , the required rotational torque can be calculated using equation 10 with the addition of the gear ratio  $\frac{1}{5}$ .

$$\tau_2 = 0.27 \text{ m}^3 \cdot 998.21 \text{ kg/m}^3 \cdot \frac{\left(\frac{1}{18} \cdot \pi\right)^2}{\pi} \cdot ((0.351 \text{ m})^2 - (0.0125 \text{ m})^2) \cdot \frac{1}{5}$$

$$\tau_2 = 0.0644 \text{ Nm}$$

After analysing both methods for calculating the required rotational torque needed for an operating velocity of 10 degrees/s, it can be noted that the methods give similar values. In conclusion the minimum required torque was approximately between 0.0644 Nm and 0.0735 Nm.

---

### 5.3.2 Rotational Motor Selection

For rotating the plate, a stepper motor like the one used for the x- and y-motion was considered. A stepper motor would be a sufficient option for the rotation as it offers good precision and would handle the torque requirements; however, a stepper motor would need an additional driver and its size could be problematic to integrate onto the x-axis carriage.

The motor chosen for the rotational motion was a servo motor. Servo motors offer considerable amounts of torque for its small size and usually have integrated drivers, making implementation easier. Servo motors are often regarded as “sweeping motors” and the rotation of the plate can be viewed as a sweep.

The specific servo motor chosen was a GoBilda 2000 Series 5-Turn, Dual Mode Servo (25-2, Torque) shown in Figure 47. This servo motor satisfies the rotational motor torque discussed in Subsection 5.3.1 as the GoBilda servo motor has a maximum stall torque of  $25.2 \text{ kg/cm}$  or  $2.47 \text{ Nm}$ . Similar to the stepper motors, the torque is considerably larger than the required torque. This will allow for a potential increase in rotational speed if needed. Lastly, the servo motor’s attributes covers all motion criteria mentioned in the start of Section 5.3 for the rotation.



Figure 47: GoBilda Servo Motor



---

### 5.3.3 Implementation of Rotary Solution

To implement this motor onto the x-axis carriage a motor enclosure had to be modelled to fit the motor and electronics. This enclosure was designed using Fusion360, and 3D-printed using PLA material. The infill of the construction was 50%. The infill percentage was chosen based on previous experiences with 3D-printed components. More specifics surrounding the enclosure's dimension can be found in Appendix E. When creating the enclosure, it is important that it fits perfectly and is well secured onto the carriage, as it is moved around by both the x-axis and y-axis stepper motors it will experience movement in all directions. Consequently, the enclosure was modeled such that it could be attached to the carriage via bolts and a press fit.

The enclosure was placed onto the carriage and the servo motor was secured inside the enclosure using screws. Furthermore, a clamping hub was placed onto the metallic pole with the large gear consisting of 100 teeth, which was secured to the clamping hub. This was done as the bore diameter of the large gear was wider than the pole's outer diameter. The clamping hub consequently allowed for implantation of the larger gear. The smaller gear head consisted of 20 teeth and was attached to the servomotor. The fully implemented solution for the rotation is shown in Figure 48.

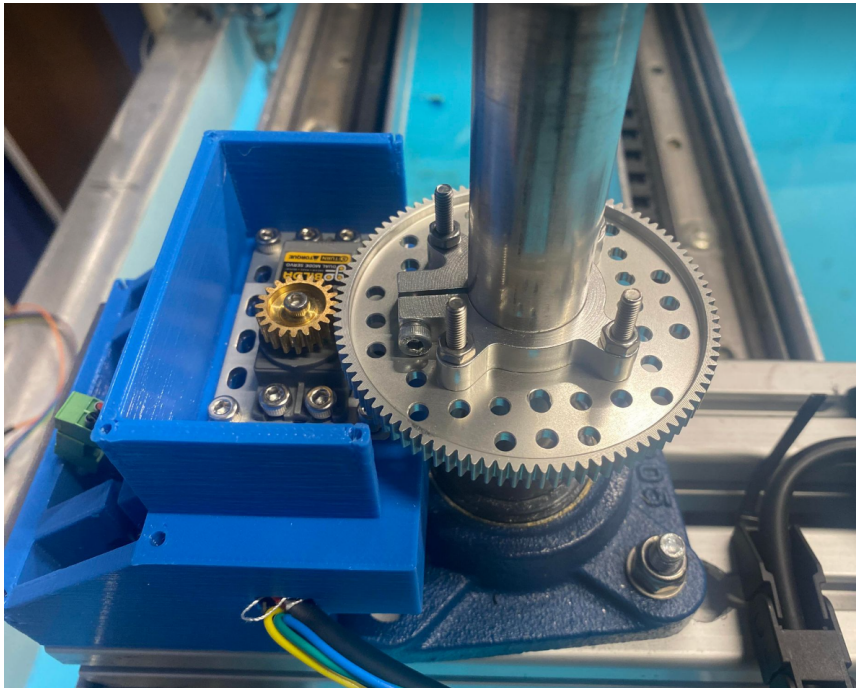


Figure 48: Implemented Rotary Solution

---

## 5.4 Cable Management

The two stepper motors fastened along the y-axis are stationary and their cables could as a result simply be fastened along the pool toward, a electronic panel. The stepper motor mounted on the y-axis carriage and the servomotor fastened to the x-axis carriage needs longer cables that can move along the x- and y-axis of the pool, previously mentioned in Section 4.4. Firstly a larger cable chain was mounted along the length of the pool. This larger cable chain held cables for both the stepper and the servo motor. The cables for the servo motor leave the larger cable chain and enter the smaller cable chain. The cable chains in the finalized model follow the finalized concept sketch in Figure 36. Both cable chains can be seen in Figures 49 and 50.

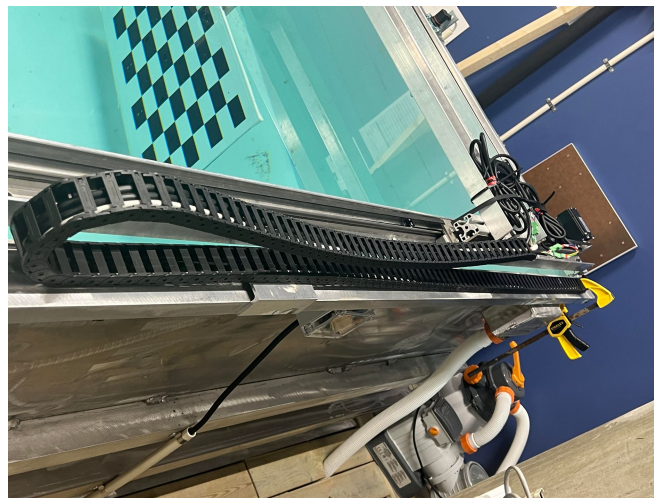


Figure 49: Large Cable Chain

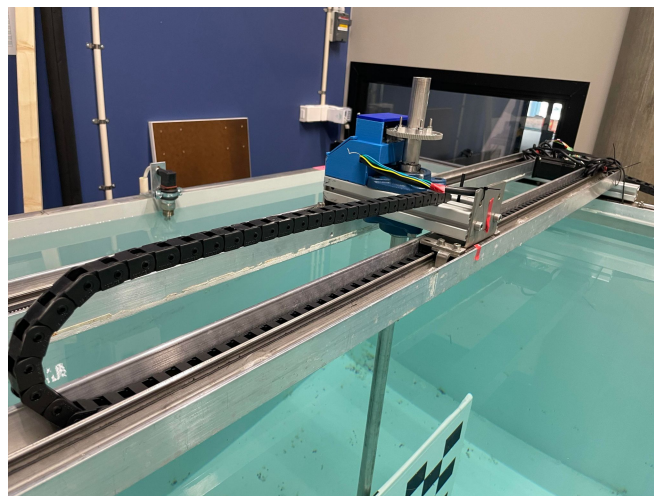


Figure 50: Small Cable Chain

---

## 5.5 Microcontroller

The Arduino Mega was chosen due to its reliability and many I/O(Input/Output) pins providing resources to connect all the motors and buttons, and still have available pins, allowing the possibility to implement more components if needed. Furthermore, it fulfils the need to store code in a nonvolatile memory. On the Mega, the program sketch uses 10890 bytes (4%) of program storage space out of a maximum of 253952 bytes. Global variables use 423 bytes (5%) of dynamic memory, leaving 7550 bytes for local variables. Maximum is 8192 bytes. The technical specifications is shown in Figure 52.

The Mega provides sufficient processing power, but the compatibility characteristics and superior support is why it was chosen over the competition. Being part of the Arduino ecosystem, it is compatible with a lot of equipment and libraries that makes the programming much simpler. There is also much support available on the internet due to its popularity. Previous experience with Arduino programming and Arduino microcontrollers also improved the feasibility of the project.

Other microcontrollers like the Raspberry Pi and ESP32 were considered. These can offer more computational power and different connectivity options, but this was not needed, and so the choice of the Arduino Mega was agreed upon with Optoscale. In the following Figures 51 and 52, the Arduino Mega and its technical specifications is shown, respectively.

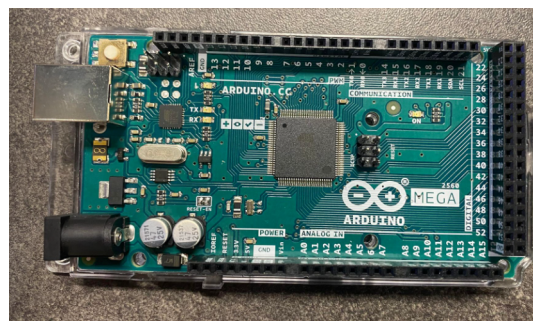


Figure 51: Arduino Mega

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

Figure 52: Tech Specs



---

## 5.6 Power Supply

Critical to the functionality of the system is its power supply, which provides the necessary energy to drive its components. The entire system is powered by a single plug connected to a 230 volt AC power outlet. The components employed, and the overall strategy implemented to ensure reliable and efficient power delivery is outlined in the following sections.

### 5.6.1 AC/DC 48V Transformer

To transform the AC from the power outlet into the DC needed for the stepper motors in the system, an AC/DC transformer, the TIB 480-148, was employed. This transformer features a rated input voltage range of 100-240 VAC and operates at an input frequency of 50/60Hz. It delivers an output voltage of 48V and a current of 10A, providing a maximum output power of 480W. According to (Traco Power, 2022), it integrates an efficient circuit topology with an optimized cost/performance ratio for industrial environments and electrical control cabinets. It boasts a high efficiency of up to 95%. Additionally, it can operate with a boost power of 150% for a minimum of 4 seconds, which facilitates activating stepper motors.

### 5.6.2 AC/DC 5V Transformer

An AC/DC 5V power supply converts the 100-240 VAC input from the power outlet into a stable 5V DC output, suitable for sensitive electronic components. The power supply ensures reliable operation by providing consistent voltage and current, which is essential for the proper functioning of components. This AC/DC power supply is a critical part of the system, delivering necessary power with minimal energy loss.

### 5.6.3 DC/DC 48V Input 6.5V Output

The TSR 1WI, a DC/DC 6.5V converter, is a non-isolated point-of-load (POL) converter with an ultra-wide 8:1 input voltage range, capable of delivering a maximum output current of 1A. Originally, it was intended to power the servomotor by connecting a cable from the 48V transformer to the converter. However, the converter was damaged during installation due to rough handling and its delicate pins. Prior to installation, it was tested and functioned as expected, but its fragility became apparent during the installation process, rendering it unsuitable for the application. As a result, an alternative solution was adopted to power the servomotor using the same 5V supply as the microcontroller unit (MCU). This revised approach is reflected in the schematics and the electrical control panel diagrams provided later. The new setup ensures a more robust and reliable power supply for the servomotor.

---

#### 5.6.4 Fuse Switch

A fuse switch was strategically installed within the electrical cabinet to ensure safe operation of the system and easy wiring access. This fuse switch serves as a protective measure to mitigate the risk of overheating and potential damage to components. By incorporating this safety mechanism, the overall reliability and longevity of the system was enhanced, providing peace of mind to operators and ensuring uninterrupted performance in industrial environments. Additionally, monitoring and maintenance of the fuse switch are recommended to uphold its safeguard and protect the system against unforeseen electrical faults.

### 5.7 Implemented User Interface

Figure 53 shows how the wires from the user interface was connected to the microcontroller. There were four digital pins connected from the joystick and one from each button. They all share a common ground wire, resulting in a total of seven wires. As these wires transmit signals at very low currents they can be thin, optimizing space and minimizing material usage without compromising performance. The distance between the Arduino and the user interface was approximately five meters. Due to this long length and thin wiring, an Ethernet cable, which can house up to eight wires, was stripped and utilized. This choice ensured water resistance and provided a visually appealing, easy-to-install solution. At the MCU end of the wires, male header pins were soldered on, allowing for easy connection to the MCU ports.

Figure 54 shows how the wires were connected to the user interface. On the joystick, two cables required soldering for each direction. One wire to ground and one to NO (Normally Open). NO means that the default position of the joystick has an open circuit, preventing any current from flowing through it. When the joystick is moved, it closes a circuit so that the current can flow through it, and a signal is read. The buttons also have the need for two wires to be soldered onto them, which also are ground and NO.

Figure 55 shows the installed user interface mounted on the side of the pool. It was securely fastened to the panel with bolts and nuts, ensuring a stable placement that allows workers to use it comfortably.

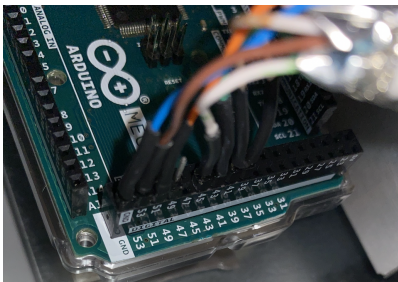


Figure 53: Connecting Wires

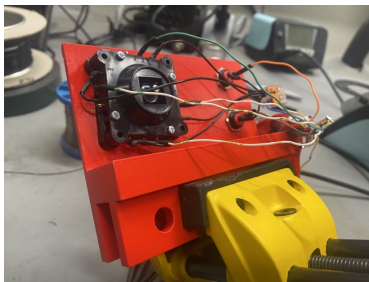


Figure 54: Soldering Wires



Figure 55: User Interface

---

## 5.8 Electrical Schematic Diagram

An electrical schematic diagram is essential for understanding and designing electronic control panels. It provides a visual, which is crucial for troubleshooting, maintenance, and communication. In the electrical schematic diagram shown in the following Figure 56, key elements like the components, connections, voltage levels and power sources are shown.

The whole system gets its power from one plug into the power outlet. It passes through the 6A fuse switch and then to both the 48V and 5V transformers. All the stepper drivers are powered by the 48V transformer, which then distributes the power to the stepper motors in a controlled manner. The servomotor and the MCU are both powered by the 5V transformer. A common ground between the components is established.

The stepper drivers share a 5V supply from the MCU, which is connected to all the *PUL+* and *DIR+* terminals. The *PUL-* and *DIR-* terminals are connected to digital pins on the MCU. There are six signal wires and four power wires running from the driver to the motor, transmitted through an encoder cable and a power cable, respectively. There are three drivers and stepper motors in the system, all wired the same except for different digital pins. Only one pair is shown in Figure 56.

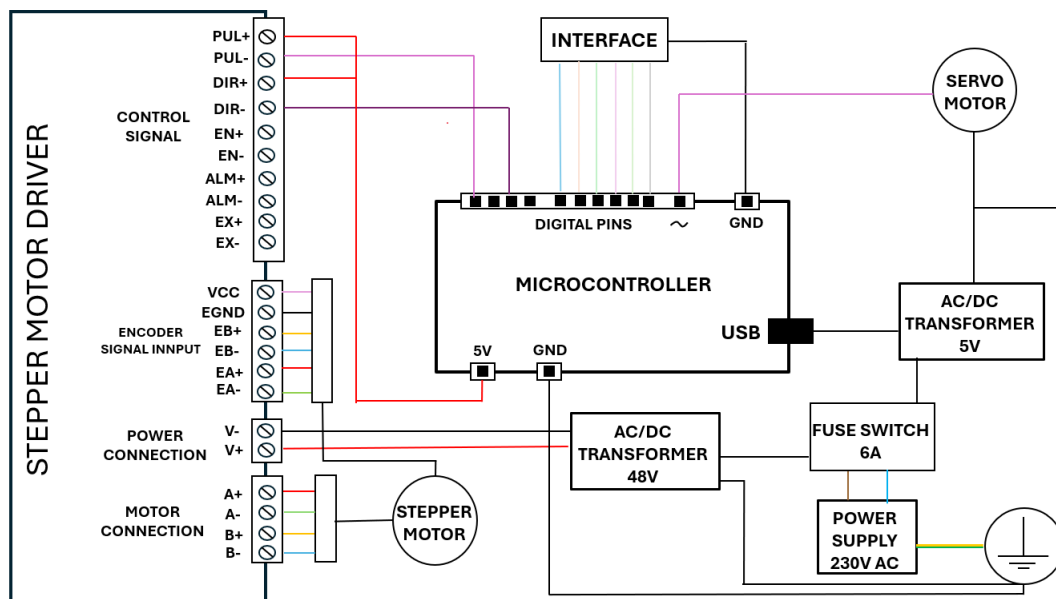


Figure 56: Schematic Diagram

---

## 5.9 Implemented Electronic Control Panel

In Figure 57 a picture of the electrical control panel is shown with all components mounted within a cabinet. To ensure that the cabinet can be closed and remain waterproof, holes were strategically made on the sides to accommodate the entry of wires from the motors and the user interface. The common ground was established at the screw hole connected to the metallic back plate in the cabinet.

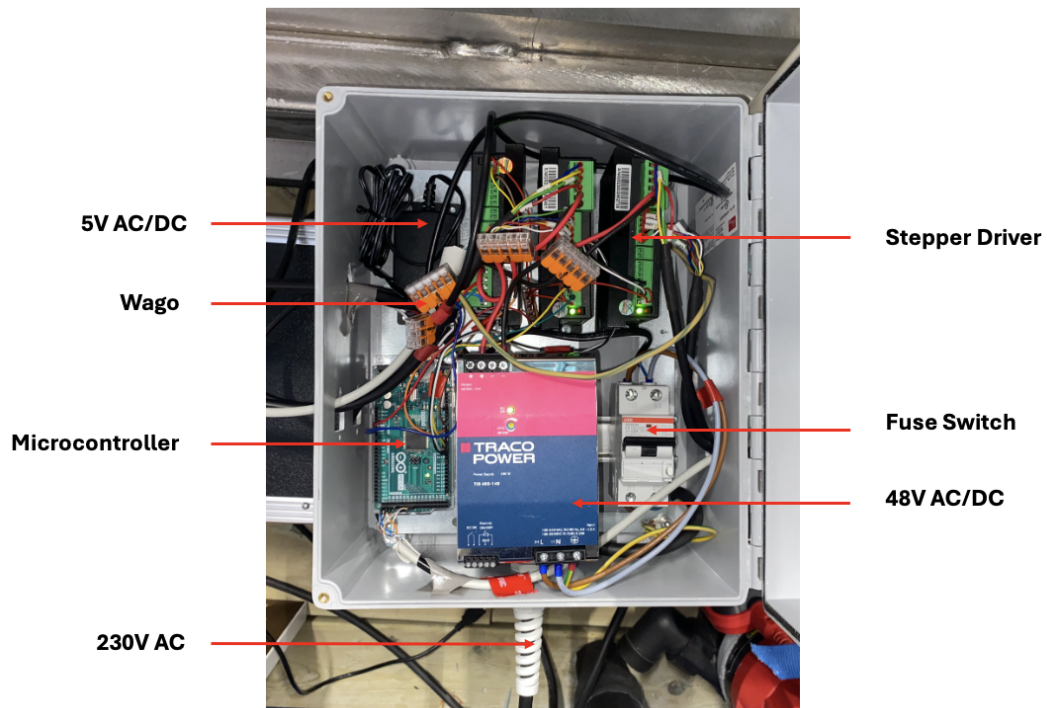


Figure 57: Control Panel

---

## 6 Limiting Factors of the System

Building on the methods laid out in Section 3 it is possible to define the operational limits of the system based on failure theory from Subsection 2.8. The following calculations consistently use the coordinate-system defined in Figure 29 where the rails move in the xy-plane with the z-axis parallel to the pole.

### 6.1 Linear Motion Parameters

The limiting factor in this system is either the motor driving the movement, the strength of the pole or the strength of the plate itself. The forces acting on the plate during linear movement is illustrated in Figure 20. During the linear movement, the pressure on and around the plate will create a force opposite of the direction of the plate, as well as trying to bend the plate around the connection point on the pole. These moments are considered when establishing the limits of the system, as well as the shear stresses induced by the drag force.

#### 6.1.1 Motor Limits

The movement along the y-axis has two stepper motors driving the movement. The stepper motors have a maximum continuous torque of 1.8 Nm each, and the maximum force output from the rated torque can be calculated by dividing the torque by the radius of the pulleys.

$$F_{max} = \frac{2 \cdot \tau_{rated}}{r_{pulley}} = 240N \quad (14)$$

The drag force from the water can be plotted as a function of the the velocity of the plate to find the maximum operational speed based on the rated torque from Equation 14, which can be seen as 0.93 m/s from Figure 58.

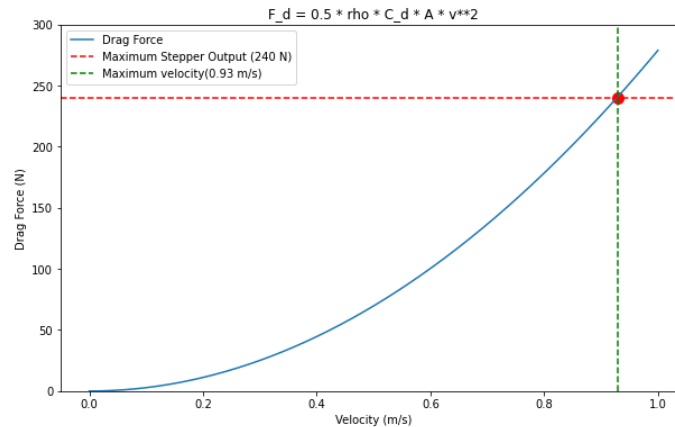


Figure 58: Drag Force of the Plate as a Function of Velocity

### 6.1.2 Limits of the Pole

Based on the assumptions laid out in Subsection 3.6 it is possible to create a shear force and moment diagram shown in Figure 59 that illustrates the distribution of stresses in the pole. These diagrams are based on the assumption from Subsection 3.6 that the pole can be considered a cantilever beam, and the origin of the diagrams represent the fixed support between the pole and the x-axis carriage.

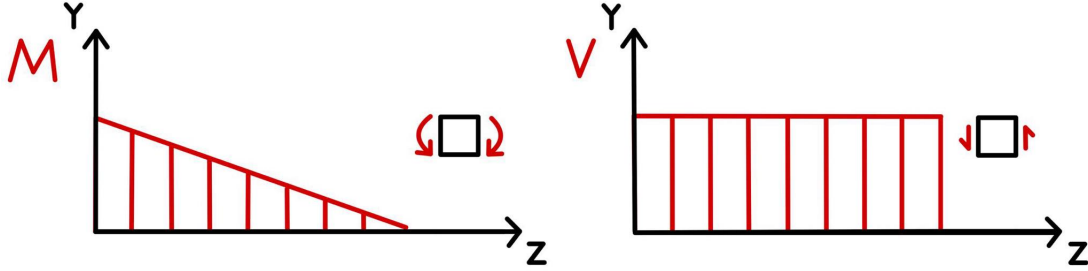


Figure 59: Moment and Shear Diagram for the Pole

From the diagrams it is clear that toward the point of contact will be the first point to yield as this is where the stresses are concentrated, assuming a uniformly distributed shear force on the cross section. The combined stresses are a result of the bending and shear forces working in the longitudinal direction of the pole with the shear force having an additional component in the radial direction. The stress state of the pole can be seen in Figure B4 in Appendix B. These stresses can be calculated from the equations in Table 2 and 3, and are shown in the following equations.

$$\sigma_b = \frac{M_{Max}}{W_y} = \frac{F_d \cdot a}{\frac{\pi}{32} \cdot \frac{D_o^4 - D_i^4}{D_o}} \quad \tau_{avg} = \frac{V}{A} = \frac{F_d}{\pi(r_o^2 - r_i^2)}$$

The load is parallel to the longitudinal axis of the pool (y-axis), where the moment bends the pole around the transversal axis (x-axis) and the shear stresses act in the yz-plane. Under these specific loading conditions, the resultant combined stresses can be evaluated against the yield stress of the material through Equation 8:

$$\sigma_{VM} = \sqrt{\frac{1}{2}[(\sigma_{xx})^2 + \frac{1}{2}(-\sigma_{xx})^2] + 3(\tau_{yz})^2} \Leftrightarrow \sigma_{yield} = \sqrt{\left(\frac{F_d \cdot a}{W_y}\right)^2 + 3 \cdot \left(\frac{F_d}{A}\right)^2}$$

The drag force and the velocity can be isolated from Equation 4 and 8 to find their maximum value before bending. After inserting the values as seen in Appendix B Listing 11, the maximum force and corresponding velocity are can be seen in Equations 15 and 16 respectively.

$$F_{d,max} = \sqrt{\frac{\sigma_{yield}^2}{\frac{a^2}{W_y^2} + \frac{3}{A^2}}} = 136 \text{ N} \quad (15) \quad v_{max} = \sqrt{\frac{2F_{d,max}}{\rho C_d A}} = 0.698 \text{ m/s} \quad (16)$$

---

### 6.1.3 Limits of the Plate

The greatest stresses of the plate will occur when the flow of water is perpendicular to it. This is when the drag force is greatest and the pressure drag creates a bending effect around the center of the plate as shown in Figure 20. From the previously established assumptions in Subsection 3.6.1, it is possible to create a shear force and moment diagram to find the point of failure, shown in Figure 60. These diagrams are based on the assumption that the plate can be considered a simple support beam, and the center along the x-axis of the diagrams correspond to the center of the plate viewed from above.

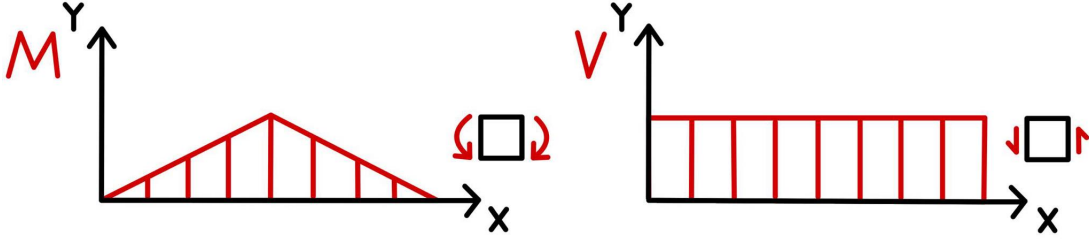


Figure 60: Moments and Shear Force Diagram for the Plate

The maximum stress will be a result of cooperation of shear and bending stresses identical to the previous subsection, and the concentration of stresses are located towards centerline of the plate from Figure 60. The following equations show the bending and shear stress expressed by the drag force.

$$\sigma_b = \frac{M_{max}}{W_y} = \frac{2 \cdot F_d}{\frac{1}{6}bh^2} \qquad \tau_{avg} = \frac{V}{A} = \frac{2 \cdot F_d}{bh}$$

The equation for the maximum drag force and velocity can be derived similarly to the previous subsection, but with differing section moduli and cross sections. The code used for calculation can be seen in Appendix B Listing 12, and the maximum force and corresponding velocity are shown in Equations 17 and 18:

$$F_{d,max} = \sqrt{\frac{\sigma_{yield}^2}{\frac{a^2}{W_y^2} + \frac{3}{4A^2}}} = 1.143 \text{ kN} \qquad (17) \qquad v_{max} = \sqrt{\frac{2F_{d,max}}{\rho C_d A}} = 2.025 \text{ m/s} \qquad (18)$$

### 6.1.4 Abaqus Simulation

Based on the results of the manual computation it is clear that the safety of the system relies on the integrity of the pole during the linear movement. Considering the simplifications of the calculation process, it was decided to use Abaqus to verify the result. Since the pole would yield at a much lower speed than the plate and that the plate calculations were more conservative, it was decided that it would be sufficient to solely simulate the pole. In Abaqus, the pole was defined as a beam fixed at the x,y and z axes and with a length of 107.5 mm, with additional info in Table 4.

<b>Material Properties</b>	
Young's Modulus	200 000 Pa
Poission's Ratio	0.3
<b>Element</b>	
Element type	(Family) Beam
Seed type	number of elements
Section type	Pipe
<b>Load Case</b>	
Boundary Condition	U.1=0, U.2=0, U.3=0
	Displacement and rotation
Load	136 N

Table 4: Abaqus conditions

In Abaqus, the beam profile was chosen as a thin-walled pipe with with an outer radius of 12,5 mm and 2 mm thickness shown in Appendix B, Figure B2. Then the maximum force of 136 N from Equation 15 was applied to the end of the beam.

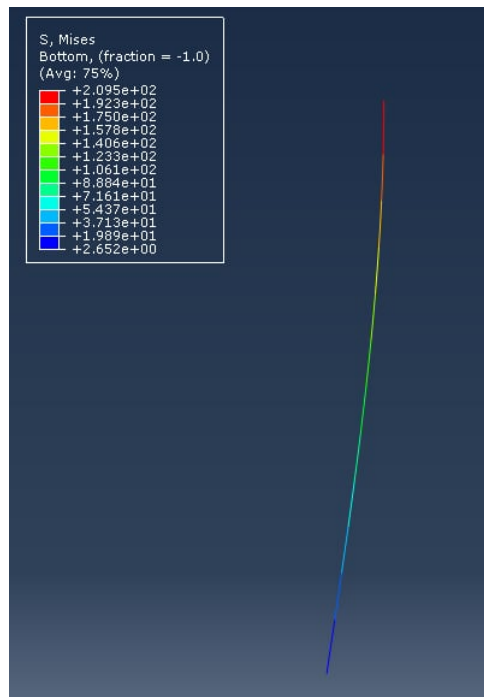


Figure 61: Simulated Stress Distribution

The stress in the beam is shown in the table on the image to the right in Figure 61, and as millimeters was used as unit, the stress is given directly as MPa, as shown in Appendix B, Figure B1. The table shows the stress in the pole at various intervals highlighted with corresponding colors. From this table it is clear that the plate is close to failure at the connection point where the stress is equal to 209,5 MPa, which is 0.24% away from the yield stress of 210 MPa.



## 6.2 Rotational Limits

As mentioned in Subsection 3.6.2, the pole will not experience great stresses during continuous rotation, but this can occur during starting or stopping. The following calculation is therefore based on the stop scenario, further founded on details the previous subsection. When the rotation stops, the previously generated current will continue to rotate the plate in the same direction due to inertia, as illustrated in Figure 62 where  $\omega$  is the angular velocity, and  $\alpha$  is the deceleration. As a result, the pole is twisted around its longitudinal axis, causing shear stresses and can potentially reach failure at high velocities or accelerations.

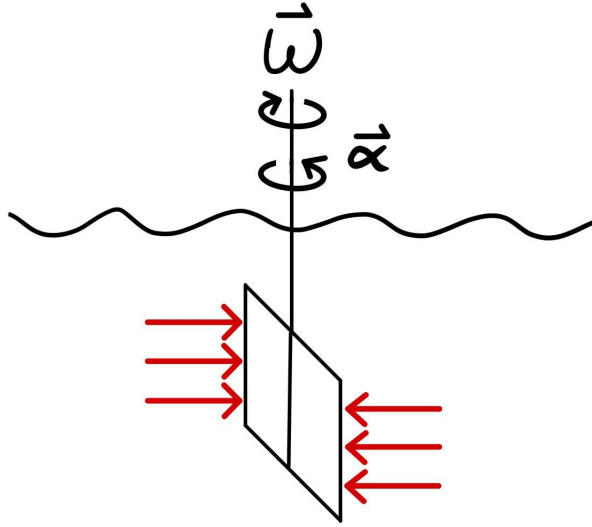


Figure 62: Flow During Deceleration

By applying the *Worst-case scenario* method to the rotation, the stresses from the torsion of the pole can be analyzed. The loading conditions are in the horizontal plane (xy-plane), where shear stresses occur due to the torsion around the vertical axis. Failure will occur towards the outer circumference of the pole where the shear stress is the greatest, identical to the diagram at the bottom of Figure 16, where no other stress components are present. The resulting von Mises and shear stress is therefore:

$$\sigma_{max} = \sqrt{3\tau_{xy}^2} \qquad \tau_{xy} = \frac{T_{max}}{W_p} = \frac{2 \cdot F_d \cdot a}{\frac{\pi}{16} \frac{D^4 - d^4}{D}}$$

By using the torque generated from the torsion and the polar section modulus for a circular cross section, the equation for torsional stress from shear stress table in Section 2.8 can be used to isolate the drag force and corresponding angular velocity. The following equations show the maximum force and angular velocity before failure.

$$F_{d,max} = \frac{\sigma_{Max} W_p}{2\sqrt{3}a} = 240.71 \text{ N} \qquad \omega_{max} = \sqrt{\frac{F_{d,max}}{\rho C_d A}} r^{-1} = 2.643 \text{ s}^{-1}$$

---

## 7 Software

Software is the backbone of the calibration system, transforming user commands into precise mechanical actions and ensuring the system operates smoothly and consistently. The following section will discuss different software, parameter calculations for motors and the code written to control the system. This section will focus on specific areas of the code, elaborating on the programming choices, highlighting the critical programming decisions that ensures the system's functionality.

### 7.1 Arduino

The automated calibration system is controlled with a code, and Arduino is a well established software option for managing the system's control. The Arduino platform offers several advantages that make it well-suited for this project. Arduino provides a user-friendly development environment with a simple yet powerful programming language based on C/C++. The Arduino IDE offers many libraries of pre-written code examples and extensive documentation, facilitating rapid prototyping and development.

For controlling the motor's movements, alternatives such as Arduino libraries, Gcode, CNC software, 3D printing software, Marlin and GRBL were considered. Each of these alternatives offers features and functionalities that would make them suitable for this system, but because of previous experiences with coding motor movements using libraries in Arduino, this was chosen. The utilized libraries were AccelStepper and Servo, offering everything needed for this project. Their inclusion is shown in the following code snippet, Listing 1.

```
1 #include <AccelStepper.h>
   #include <Servo.h>
```

Listing 1: Libraries

For coding, Micropython offers ease of use and flexibility, but with previous experiences using Arduino C/C++. and the extensive software support, the choice was clear. Arduino's simplicity, ease of use, reliability, versatility, and real-time capabilities made it a good choice for controlling the system involving multiple motors and input devices.

---

With the stepper- and servo libraries, motor control can be integrated, where step frequency and amount of rotations are used as parameters. The speed of the stepper motors has the unit steps per second, based on the step angle of the motor from Equation 1. The amount of steps per revolution is set through dip switches on the driver to 800. Determining the step frequency depends on the desired speed of the plate and the corresponding angular velocity of the pulley driving the belt shown in Equation 19.

$$f_{Step} = \frac{v_{belt}}{r_{pulley}} \cdot \frac{800 \text{ steps}}{2\pi \text{ rad}} = 1697,65 \text{ steps/s} \approx 2 \text{ rotations/s} \quad (19)$$

The amount of rotations to move the plate a certain distance given by the calibration path, can be expressed by dividing said distance by the periphery of the pulleys. For the x- and y-axis motors, with the distances of 1 meter and 1.5 meters respectively, the amount of rotations needed to move the mentioned distances can be found in the following Equation 20.

$$n_{stepper} = \frac{d}{2\pi \cdot r_{pulley}} \quad (20)$$

From servomotor specifications shown in Appendix D, Figure D3 it can be read that the servo motor has a set rotational travel of  $1.11\mu s/^\circ$  or  $0.9^\circ/\mu s$ . The average speed can be modified by increasing the delay between each microsecond. The input used to specify the angular velocity in the Arduino code is defined as milliseconds of delay between every microsecond of rotation, where one microsecond equals 0.9 degrees. A factor  $\kappa$  can be used to specify the exact servo delay to achieve a desired average angular velocity. An equation can be set up so that  $\kappa$  times the rotation speed of the servo is equal to the required angular velocity to rotate the plate 10 degrees per second.  $\kappa$  can be isolated by expressing the angular velocity as seconds per radian rather than radians per second and dividing it by the rotation speed of the servo, resulting in the servo delay with a unit of  $ms/\mu s$  shown in Equation 21.

$$\kappa_{Servo} = \frac{1}{0.9\mu s/^\circ} \cdot \frac{1000ms/s}{\omega_{plate} \cdot i} = 22.22ms/\mu s \quad (21)$$

To tell the script how far to rotate the servo, the Arduino library needs the duration of the rotation in microseconds, denoted as  $t$ , required to reach this angle. For a given rotation of the plate equal to 30 degrees, the duration  $t$  can be calculated by dividing 30 degrees by the product of the gear ratio and the rotation speed, which is 0.9 degrees per microsecond shown in Equation 22.

$$t = \frac{30^\circ}{0.9^\circ/\mu s \cdot i} = 166.67\mu s \quad (22)$$

---

## 7.2 Code Criteria

Several criteria are essential for the code's functionality. Some of them are: Continuous Button Monitoring, Sequential Execution of Motor Functions, Structured State-Based Communication, Joystick Handling and User-Friendly Configuration.

In order to achieve this, the code is structured with a main loop that continuously monitors button presses and updates the current state accordingly. When the button that initiates the process is pressed, the boolean variable "START" is set to true, triggering the execution of the code within the corresponding "if" statement. This changes the state to "STEPP1", starting the sequential executions. The following code snippet, Listing 2 demonstrates the states and the method for reading button inputs.

```
enum State {  
2  STEPP1,  
  STEPP2,  
4  STEPP3,  
  STEPP4,  
6  STEPP5,  
  STEPP6,  
8  STEPP7,  
  STEPP8,  
10 SERVO,  
  IDLE  
12 };  
void loop() {  
14  bool START = debounceButton(START_PIN);  
  bool STOP = debounceButton(STOP_PIN);  
16  
  if (START){  
18    stepp1();  
    currentState = STEPP1;  
20  }  
}
```

Listing 2: State Structure and Button Input

---

Because the press of a button can make multiple rapid electrical contacts, which can lead to false positives, debouncing is essential to ensure accurate and reliable detection of button presses. With debouncing, the readings are not done all the time. After testing, it was found that a reading interval of 200ms was optimal. The function "millis()" is used instead of "delay()", as it allows the program to continue executing other tasks during the waiting period. Listing 3 demonstrates how debouncing is implemented and how button readings are handled.

```
1 const unsigned long debounceDelay = 200; // Adjust debounce delay as needed
   unsigned long lastDebounceTime = 0; // Variable to track the last debounce time
3 bool lastButtonState = HIGH; // Variable to track the last button state

5 bool debounceButton(int buttonPin) {
   unsigned long currentTime = millis();
7   bool currentState = digitalRead(buttonPin);

9   // Check if there's a transition from high to low
   if (currentState == LOW && lastButtonState == HIGH) {
11     if (currentTime - lastDebounceTime > debounceDelay){
       lastDebounceTime = currentTime; // Update last debounce time
13     lastButtonState = LOW; // Update last button state
       return true; // Button press detected
15     }
   } else if (currentState == HIGH && lastButtonState == LOW) {
17     lastButtonState = HIGH; // Update last button state
   }
19   return false; // No stable button press detected
}
```

Listing 3: Debounce

Motor functions are defined separately for each of the eight movements, with parameters adjustable for speed and distance. The max speed and max acceleration is defined in the "setup()". Because the motor movements are split into multiple functions, the speed and acceleration along with distance can be individually set for each function, or default to max. The parameters for all the motors were tuned after implementing the full-scale model and receiving feedback from the bioscope's calibration. Taking into consideration calculations such as limiting speed, the speed given in steps per second is calculated in Equation 19. The number of rotations the stepper motors should perform in each direction was calculated using Equation 20. There are eight stepper motor functions corresponding to the eight direction changes in the movement pattern. These functions, along with the setup for speed and acceleration, are shown in the following code snippet, Listing 4.

```

const int stepsPerRevolution = 800; // Set by dipswitches on the driver
2 const int stepper12speed = 400;    // Adjust after desired speed
const int stepper12acc = 800;       // Adjust after and acceleration
4 const int stepper3speed = 400;     // Steps per second, 800=1rpm
const int stepper3acc = 800;       // Steps per second^2
6 void setup() {
    stepper1.setMaxSpeed(stepper12speed); // speed applied with "move"
8    stepper1.setAcceleration(stepper12acc); // acceleration applied with "move"
    stepper2.setMaxSpeed(stepper12speed);
10   stepper2.setAcceleration(stepper12acc);
    stepper3.setMaxSpeed(stepper3speed);
12   stepper3.setAcceleration(stepper3acc);
}
14 void stepp1() {
    float stepper12Rotations = 15; //number of rotations
16   stepper1.move(- stepper12Rotations * stepsPerRevolution); // Opposite directions
    stepper2.move( stepper12Rotations * stepsPerRevolution); }// Opposite directions
18 void stepp2() {
    float stepper123Rotations = 5;
20   stepper1.move( stepper123Rotations * stepsPerRevolution);
    stepper2.move(- stepper123Rotations * stepsPerRevolution);
22   stepper3.move(- stepper123Rotations * stepsPerRevolution); }
void stepp3(){
24   float stepper12Rotations = 10;
    stepper1.move( stepper12Rotations * stepsPerRevolution);
26   stepper2.move(- stepper12Rotations * stepsPerRevolution); }
void stepp4(){
28   float stepper3Rotations = 10;
    stepper3.move( stepper3Rotations * stepsPerRevolution); }
30 void stepp5(){
    float stepper12Rotations = 10;
32   stepper1.move(- stepper12Rotations * stepsPerRevolution);
    stepper2.move( stepper12Rotations * stepsPerRevolution); }
34 void stepp6(){
    float stepper123Rotations = 5;
36   stepper1.move(- stepper123Rotations * stepsPerRevolution);
    stepper2.move( stepper123Rotations * stepsPerRevolution);
38   stepper3.move(- stepper123Rotations * stepsPerRevolution); }
void stepp7(){
40   float stepper12Rotations = 7;
    stepper1.move( stepper12Rotations * stepsPerRevolution);
42   stepper2.move(- stepper12Rotations * stepsPerRevolution); }
void stepp8(){
44   float stepper12AntallRunder = 8;
    stepper1.move( stepper12Rotations * stepsPerRevolution);
46   stepper2.move(- stepper12Rotations * stepsPerRevolution); }

```

Listing 4: Stepper Motor Functions

---

There is a ninth motor function which is for the servo. The servo moves 30 degrees to both sides and is the last step in a successful calibration. The "interval servo" variable controls the speed of the motor. This is the pause the motor takes between every iteration. The speed is shown in Equation 21. The different target degrees are stored in "targetPositions", with the home position at 1500. The motor has specifications of 0.9degrees/millisecond, and determining how far from the home position the target positions are shown in Equation 22. The following code snippet, Listing 5 shows the servomotors function.

```
const int targetPositions[] = {1333, 1667, 1500}; //0.9degree/microsec
2 int interval servo = 22; // Adjust interval after desired speed millisec/degree
int pos=1500; // Home position
4
void serv1() {
6 unsigned long previousMillis = millis();
8 for (int i = 0; i < 3; i++) { // Loop through the array of target positions
int targetPos = targetPositions[i];
10 while (pos != targetPos) { // Move the servo to the target position
if (pos < targetPos) {
12 pos++;
} else {
14 pos--;
}
16 servo.writeMicroseconds(pos);
while (millis() - previousMillis < interval servo) {
18 // Wait for the specified interval
}
20 previousMillis = millis();
}
22 }
}
```

Listing 5: Servo Function

The code iterates through motor functions using states with specific conditions that allow the iteration to proceed. The following code snippet, Listing 6 checks whether or not all the stepper motors have finished their movement. If they have, the next function and state will be active.

```
1 case STEPP1:
if (stepper1.distanceToGo() == 0 && stepper2.distanceToGo() == 0 && stepper3.
distanceToGo() == 0) {
3 stepp2();
currentState = STEPP2; }
5 break;
```

Listing 6: State Condition

---

The system enables manual control through joystick movements, offering Optoscale the flexibility to move motors beyond the automated calibration. The joystick is digital and acts like four buttons, triggering stepper motor movement upon activation. For instance, upward joystick movement prompts two stepper motors to adjust the plate toward the camera. To prevent constant motor engagement, a boolean variable "motorsStopped" is implemented. Speed settings are configured at twice the calibration speed, a value below the maximum operational speed for the pole from Equation 16. The joystick is connected using Normally Open(NO), therefore the if-statements check for "LOW" readings, indicative of false boolean states. The corresponding code used, only displaying one direction, is shown in the following Listing 7.

```
1   bool JOYSTICK_X_1 = digitalRead(JOYSTICK_X_1_PIN);
2   bool JOYSTICK_X_2 = digitalRead(JOYSTICK_X_2_PIN);
3   bool JOYSTICK_Y_1 = digitalRead(JOYSTICK_Y_1_PIN);
4   bool JOYSTICK_Y_2 = digitalRead(JOYSTICK_Y_2_PIN);
5   if (JOYSTICK_X_1==LOW){
6       stepper1.setSpeed(-stepper12speed*2); // twice the speed
7       stepper2.setSpeed(stepper12speed*2);  // twice the speed
8       stepper1.runSpeed();
9       stepper2.runSpeed();
10      motorsStopped = false;
11  }
12  else if(JOYSTICK_X_1 && JOYSTICK_X_2 && !motorsStopped){
13      stepper1.setSpeed(0);
14      stepper2.setSpeed(0);
15      motorsStopped=true;
16  }
```

Listing 7: Reading Joystick

It is important to have a user-friendly configuration that contributes to a code that is easy to understand, modify and maintain, promoting efficient development and troubleshooting processes. Because Optoscale might change the movement pattern in the future, it is important to be able to make changes with minimal effort. To achieve this, descriptive naming, comments, a structured state machine and a readable code layout were implemented. Variable- and function-names are descriptive and provide clarity regarding their purpose. For example, the defined pins regarding drivers and user interface are clearly indicated, which is shown in the following Listing 8.

```
1 #define STEPPER1_STEP_PIN 6 // pulse
2 #define STEPPER1_DIR_PIN 7 // direction
3 #define SERVO_PIN 13 // servo signal
4 #define START_PIN 53 // start button
5 #define STOP_PIN 51 // stop button
6 #define JOYSTICK_X_1_PIN 41 // joystick x1 direction
7 #define JOYSTICK_Y_1_PIN 45 // joystick y1 direction
```

Listing 8: Descriptive Naming



---

The code includes comments that explain the purpose of certain variables, functions, and sections of code. These comments make it easier to understand and modify the program. The code implements a state machine to manage the sequence of operations. Each state represents a specific stage of the process, making it easier to track and understand the program flow. The code is formatted in a readable manner with consistent indentation and spacing, which enhances clarity and makes it easier to follow the logical structure. The states correspond to the numbered movements in Figure 4, with two additional states named "IDLE" and "STEPP8". The "IDLE" state occurs when the system is inactive, awaiting signals from the user interface. The "STEPP8" state represents a movement executed after calibration to ensure that the starting position matches the end position. This is advantageous because it standardizes the starting position, eliminating the need for manual adjustments between calibrations.

### 7.3 Programming Thought Process

Programming the system was time consuming. Due to the inherent need for frequent modifications to the code, it helped to have the webpage Wokwi, which served as digital counterpart that emulates functionality and responses. This allowed rapidly testing, validation of code and experiments with different parameters, and troubleshooting potential issues before deploying the software to the actual hardware.

During the process of developing the code it is essential to map out the required functionalities of the system. This involved identifying the types of movements, control mechanisms, and any additional features needed. Choosing the appropriate libraries significantly impacts the system's performance. A lot of time was used in the decision making between using "GRBL" script or the "AccelStepper" library to control the motor movements. It depended on factors like ease of use, compatibility with hardware and specific requirements such as acceleration and movements.

It was important to organize the code into modular components that enhanced readability and facilitates future modifications. Different coding methods like iterating through a "for-loop" was thought of, but the choice of using states and functions that represents a distinct actions was chosen. Implementing a state machine architecture enables sequential execution of motor movements and other tasks. States correspond to different stages of operation, allowing orderly progression through the system's workflow, while efficiently managing input signals, such as joystick movements and button presses, ensuring responsive system behavior.

The closed-loop motors enhanced system reliability by providing feedback to adjust performance in real-time. Initially, a response code utilizing PID (Proportional-Integral-Derivative) control was considered to manage this feedback. However, the motor drivers already integrated the necessary control software, rendering the additional PID code unnecessary. This integration simplified the system and ensured efficient and accurate motor control without the need for further programming.

---

During testing and development, serial communication in the Arduino IDE proved helpful. The "Serial.print()" function was utilized to monitor states, track movements, and determine the optimal debounce interval. The following code snippet illustrates its use in identifying the appropriate debounce interval. It prints a message when the start button is pressed, along with a counter index to clearly display the readings. Various intervals were tested iteratively until the button presses were optimized.

```
1 int count = 0; // remove after test
2 void loop() {
3   bool START = debounceButton(START_PIN);
4   if (START){
5     Serial.print("START BUTTON PRESSED: ");
6     count = count + 1;
7     Serial.println(count);
8   }
9 }
```

Listing 9: Testing

## 7.4 Code Description

Shortly explained the code has a state machine architecture that enables sequential execution of motor movements. States correspond to different stages of the operation, allowing orderly progression through the system's workflow, while efficiently managing input signals, such as joystick movements and button presses.

---

## 8 Results

This section will present the finished product and discuss the results gathered from testing the calibration system with a bioscope.

### 8.1 The Finished Implemented System

In Figure 63, the final full-scale model result is linked as a video. Click the following image to watch it in a web browser. The complete code and a simulation along with a user manual is available at the webpage Wokwi, linked here: [\[Open Wokwi\]](#).

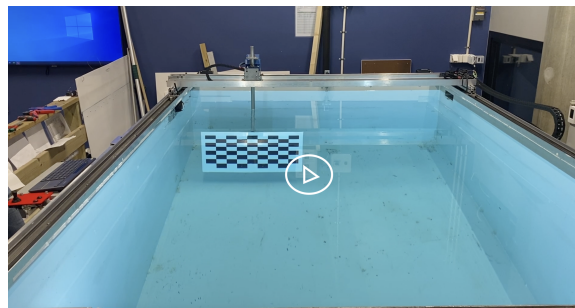


Figure 63: Video of the Final Product

The final result was a fully functioning and automated calibration system implemented on the in-place railing system at Optoscale's laboratory. The automated system incorporated all criteria such as a user interface with a start/stop button and a digital joystick that moves the carriage in all directions: forward, backward, sideways and diagonally. The calibration system effectively completed the automatic calibration process for the cameras in the bioscope. The automated movements were accomplished precisely as defined in the code and all components communicated. The system was completed within the allocated budget and time frame. Optoscale expressed great satisfaction with the outcome of the project.

#### Optoscale Comment:

*"The students have delivered a project that transformed a simple problem description from us into a finished product that will directly impact the efficiency of our work. We are very pleased and wish the students the best of luck in their future careers!"*



- *Sven J. Kolstø* CEO OPTOSCALE

---

## 8.2 System Modifications and Tuning

The code and force analysis were based on the approximate values given by the company during commencement of the project. The company estimated a driving speed of 20 *cm/s*, rotational speed of 10 *degrees/s* and a rotation of 30 *degrees*. Initial test runs of the system applied the approximate values and resulted in a non-sufficient calibration. After iterative testing of the system, new values for system operations were determined. The finalized system employed a 10 *cm/s* driving speed and a rotational speed of 10 *degrees/s* with a 40 degree rotation of the plate. Applying these values into the code gave a fully optimal calibration system.

---

## 9 Discussion

This discussion serves as the team's reflection of the choices and changes made throughout the project. It will also evaluate the results based on the assignment's objectives and goals and discuss further potential improvements upon the system.

### 9.1 Calculations

Calculating the drag force was central to this project, and therefore it was taken many steps to obtain an adequate estimation. Considering that the calculation and Abaqus simulation were relatively similar for the linear movement, this approximation was deemed sufficient. Regarding the rotation, both simulation and calculation would be difficult due to the uncommon nature of the case.

A potential solution to analyze the rotation was to calculate the drag coefficient of the plate experimentally with the use of Equation 4 to create a function for the drag force through regression analysis. This would then question the accuracy of the force measurements. This was then dismissed in favor of mechanical simplifications. The simplifications were based on conservative assumptions to obtain a minimum torque required to drive the system, and due to the values coinciding to an adequate degree and being rather low, it was concluded that the results were sufficient.

The theoretical mechanical limits ascertained previously were not verifiable, and therefore the accuracy of the estimations are under question; however, the exactness is not crucial to the goal of creating an automated calibration system. The limits were used as threshold values to prevent potential mechanical failure due to high velocities, and it was therefore concluded that the estimations were acceptable in accordance with the purpose of the project.

### 9.2 Electrical Components

To power the servomotor, initially a dc-dc converter placed close to the servomotor was planned, which would avoid voltage drops that occur with long cable runs, thin cables, and high current draw. It would convert 48V to 6.5V, but it broke due to the fragility of the pins. An easier solution was thought of after brainstorming other solutions. It was realized that the 5V AC-DC already in the cabinet could power both the Arduino and the servomotor. Verification involved measuring the amperes drawn by the servomotor and monitoring voltage levels near the motor during rotation. Following successful testing, the new solution improved the overall system.

---

### 9.3 Sustainability

”We believe that OptoScale’s technology will enable us to produce salmon with a lower environmental footprint, enhance fish welfare, and improve quality in the years to come.” (Knut Utheim, n.d. cited from Optoscale, 2023). Optoscales cameras represent a significant advancement towards a less invasive method of researching marine life, allowing for detailed observations without disturbing their natural habitat. Incorporating the automated calibration, frees up manpower to focus on enhancing camera software and capabilities to further improve sustainable practices in fish inspections.

In accordance with the United Nations Sustainable Development Goals, Optoscale’s camera technology will contribute to goals: 3 *good health and well-being*, 12 *responsible consumption and production* and 14 *preserving life below water*. Fish are among the most sustainable protein sources and will play a crucial role in prompting general good health in the future. Optoscales camera technology will ensure good fish welfare. In combination with better charting and optimized camera inspections waste in fish farming can be reduced, and allow for a more efficient and sustainable food production which upholds set quotas. Real time measurements using camera technology will minimize disruption in aquatic ecosystems and ensures sustainable use of marine resources, crucial for preserving life below water. This technology can also help, advance research about marine biodiversity and protection of fragile marine habitats.

### 9.4 Evaluation of Goals and Results

The objective of the assignment was to create an automated calibration system with a user interface to control it manually and activate the automation. This result was achieved while also adhering to the performance goals and budget established in the introduction; however after testing, it was realized that the initial estimations were imprecise. The inaccuracy of the estimations however did not result in any inconveniences to the system, but did exaggerate the scale of the forces resulting in much stronger motors than necessary. The strength of the motors was already considerably greater than the system requirements, and this was intentional. Given the team’s inexperience, it was important to ensure the system would operate effectively even at slightly higher speeds, and based on the function of the drag from Figure 58, the motors would not be able to drive the plate much faster. On the premise of the budget not being a limiting factor and the recommendation from Optoscale’s technician to buy a reliable motor, it was decided to go for a motor with high torque and built-in feedback to prevent positional errors. On top of that, extra powerful motors could allow for future upgrades and ensure a long-lasting system.

---

## 9.5 Future Development

The currently implemented system is mainly limited to fundamental components. This decision comes from Optoscale's emphasis on a simple and reliable system. Therefore the team prioritized completing the initial build rather than overcomplicating it with unspecified features. In the future however, there are potential improvements that can be applied upon the system to streamline it towards a fully automated system.

### 9.5.1 User Interface

The current method of turning the system on and off involves plugging it in and out of the power outlet. However, there is room for improvement to reduce power losses and enhance user-friendliness. Several options could be explored: 1. Installing a relay capable of toggling power from the power outlet. 2. Wiring the enable-pins on the driver to deactivate motor coil currents. 3. Incorporating a relay near the MCU toggling power to the 48V transformer. Toggling this power off, would only supply power to the MCU and the Servomotor. Notably, the servomotor consumes negligible power unless prompted by the MCU, and the MCU itself draws minimal power in the IDLE state. These options can all be managed through the user interface by incorporating an additional component or leveraging the start and stop buttons. By modifying the software, a double press could for instance serve as a power switch.

The user interface functioned according to expectations. For a more enduring and versatile solution, transitioning to a wireless setup could be advantageous. Incorporating a wireless communication module, such as a Raspberry Pi or an ESP32 would enable remote execution of code changes and actions from anywhere. This update could not only ensure greater flexibility but also align with sustainability goals by reducing reliance on physical connections.

### 9.5.2 Feedback

The feedback present in the linear motion system is the encoders included in the motors, providing insight into distance traveled. It functions as a step position verification which can calculate errors in the event of a stall and use that to correct the step count. On top of that, the stepper drivers have a ceiling value for current consumption, where the motors are shut off if exceeded. The code for the system runs a predefined path from a fixed starting position to an acceptable degree; however deviating from the set start position from unaware or unintended plate movement could lead to inaccuracies in calibration or shutdown due to increased current consumption from collisions with the pool wall. Considering the goal for the finished rig was to be fully automated with as minimal manual intervention as possible, more sensors could be implemented to expand feedback capabilities and increase safety.

---

To address the issue of starting position deviation, two limit switches can be installed along the x and y-axis paths. A limit switch is a sensor that will signal when the top railway bridge has reached a given position on the path. Not only is the sensor useful for defining the whole pathway length, but they also provide feedback on the number of steps required to reach the end of the pathway and can be used to calibrate the motors, similar to how 3D-printers operate and reduce positional shifting over time. With the incorporation of limit switches, the plate can be reset to its start position within the pool regardless of its current location. This reset functionality can be initiated either as an automatic step following the activation of the start button or as a separate action through a dedicated reset button.

### 9.5.3 Code Improvement

By incorporating camera calibration results and live-feedback from camera software, calibrations can be further improved. Success rates from calibrations can be incorporated in the code to improve the calibration route speed and make them as efficient as possible. The complete code was delivered, where adjustments to the parameters can be made to fine-tune the movement of the plate and optimize the performance.



---

## 10 Conclusion

In conclusion, this thesis successfully completed the task of creating an automated calibration system which can be controlled through an interface. The project met its objective through developed concepts, force analysis, construction and programming. The final system operated as expected and met all the company requirements. Furthermore, the finalized construction was built significantly stronger than the requirements, which allows it to operate at higher velocities if needed.

The selected components and concepts were successfully integrated and operated as a cohesive unit, incorporating three closed loop stepper motors for linear motion of the plate, and one servo motor for rotational motion. The code written for controlling the system was sufficient, but required slight adjustments, due to approximate values for optimal velocity and rotation needed to be reduced to half of the original values.

---

## References

<https://www.ljmu.ac.uk/-/media/library/library-skills/harvardreferencingimages.pdf>

[1] *Added mass*(n.d)

Available at:

[https://www.chemeurope.com/en/encyclopedia/Added\\_mass.html](https://www.chemeurope.com/en/encyclopedia/Added_mass.html)

(Accessed 18 May 2024)

[2] *Arduino Mega*(n.d.)[Photograph]

Available at:

<https://store.arduino.cc/products/arduino-mega-2560-rev3>

(Accessed: 17 April 2024)

[3] Cengel, Y.A, Cimbala, J.M (2018) *An annular control volume that encloses the impeller section of a centrifugal pump*. [Digital Image]

Available at:

[https://archive.org/details/ed4\\_20201119/page/275/mode/2up](https://archive.org/details/ed4_20201119/page/275/mode/2up)

(Accessed: 10 April 2024)

[4] Cengel, Y.A, Cimbala, J.M. (2018) *Fluid Mechanics Fundamentals and Application*. 4th ed. McGraw-Hill Higher Education

[5] Cengel, Y.A, Cimbala, J.M(2018) *Representative drag coefficients for various three dimensional bodies* [Digital illustration]

Available at:

[https://archive.org/details/ed4\\_20201119/page/623/mode/2up](https://archive.org/details/ed4_20201119/page/623/mode/2up)

[6] Cengel, Y.A, Cimbala, J.M (2018) *Side and frontal views of a typical centrifugal pump*. [Digital Image]

Available at:

[https://archive.org/details/ed4\\_20201119/page/275/mode/2up](https://archive.org/details/ed4_20201119/page/275/mode/2up)

(Accessed: 10 April 2024)

[7] Cengel, Y.A, Cimbala, J.M (2018) *Streamlined vs Non-streamlined Body in a Current* [Image]

Available at:

[https://archive.org/details/ed4\\_20201119/page/617/mode/2up](https://archive.org/details/ed4_20201119/page/617/mode/2up)

(Accessed: 17 April 2024)

- 
- [8] Cengel, Y.A, Cimbala, J.M. (2018) *The behaviour of colored fluid injected into the flow in (a) laminar and (b) turbulent flow in a pipe* [ Digital illustration]  
Available at:  
[https://archive.org/details/ed4\\_20201119/page/353/mode/2up](https://archive.org/details/ed4_20201119/page/353/mode/2up)  
(Accessed: 9 May 2024)
- [9] Cengel, Y.A, Cimbala, J.M. (2018) *The development of the boundary layer for flow over a flat plate, and the different flow regimes.* [Digital illustration]  
Available at:  
[https://archive.org/details/ed4\\_20201119/page/629/mode/2up](https://archive.org/details/ed4_20201119/page/629/mode/2up) (Accessed: 1 May 2024)
- [10] Cengel, Y.A, Cimbala, J.M. (2018) [Digital illustration]  
Available at:  
[https://archive.org/details/ed4\\_20201119/page/629/mode/2up](https://archive.org/details/ed4_20201119/page/629/mode/2up) (Accessed: 1 May 2024)  
item *Closed loop NEMA23 stepper motor*(n.d.)[Digital illustration]  
Available at:  
<https://www.zeptobit.com/index.php?product=249>  
(Accessed: 05 April 2024)
- [11] *Combined stresses*(n.d.)[Digital illustration]  
Available at:  
<https://mechanicalc.com/reference/strength-of-materials>  
(Accessed 4 April 2024)
- [12] *Controlling a servo motor through PWM*(2024) [Digital illustration]  
Available at:  
[https://ntnu.blackboard.com/ultra/courses/\\_43188\\_1/cl/outline](https://ntnu.blackboard.com/ultra/courses/_43188_1/cl/outline)  
(Accessed: 24 January 2024)
- [13] Dahl, Ø.N. (2018) *What is an H-bridge?*  
Available at:  
<https://www.build-electronic-circuits.com/h-bridge/>  
(Accessed: 17 January 2024)
- [14] Fiore, C.(n.d.)  
Available at:  
<https://www.monolithicpower.com/en/stepper-motors-basics-types-uses>  
(Accessed: 17 January 2024)
-

- 
- [15] Fiore.C (n.d) *Stepper Motor Steps* [Digital illustration].  
Available at:  
<https://www.monolithicpower.com/en/stepper-motors-basics-types-uses>  
(Accessed: 17 Januar 2024)
- [16] Freyer et al.(2022) Critical Function Prototyping of an Automated Cleaning System for Photovoltaic Panels, *NordDesign 2022*. Copenhagen 16th - 19th August 2022. Copenhagen: Design Society, p.1.  
Available at:  
<https://doi.org/10.35199/NORDDDESIGN2022>
- [17] *[GANTRY ROBOT]* [online image] (n.d.)  
Available at:  
<https://www.mwes.com/types-of-industrial-robots/cartesian-gantry-rectilinear-robots/>  
(Accessed: 5 March 2024)
- [18] Hydraulic Institute(2018)*How Centrifugal Pumps Work*  
Available at:  
<https://www.pumpsandsystems.com/how-centrifugal-pumps-work>  
(Accessed 9 May 2024)
- [19] Hurlston, R.(2021) *What Is von Mises Stress*  
Available at:  
<https://www.fidelisfea.com/post/what-is-von-mises-stress>  
(Accessed 17 March 2024)
- [20] Johannessen, J. (2002) *TEKNISKE TABELLER*. 2nd ed. Norway: J. W. Cappelens Forlag A/S
- [21] Johnsen, H.J.D, 2024. Regulering og Mikrokontrollere. *TPK4125*. Mechatronics.  
Available at:  
Blackboard-TPK4125  
(Accessed: 22 March 2024)
- [22] Material properties (n.d) *Strength of materials - Tensile - Yield*  
Available at:  
<https://material-properties.org/strength-of-materials-tensile-yield/>  
(Accessed at 10 April 2024)
-

- 
- [23] *Motioncontrolproducts* (n.d.)  
Available at:  
<https://motioncontrolproducts.com/resources/stepper-motor-selection-guide>  
(Accessed: 17 January 2024)
- [24] Nanotec(n.d) *CLOSED-LOOP TECHNOLOGY*  
Available at:  
<https://en.nanotec.com/knowledge-base-article/closed-loop-technology/>  
(Accessed: 2 April 2024)
- [25] National Geographic (2023) *What is aquaculture? It may be the solution to overfishing.*  
Available at:  
<https://www.nationalgeographic.com/environment/article/aquaculture-explainer-seaweed-fish-benefits-challenges>  
(Accessed: 10 April 2024)
- [26] National Oceanic Atmospheric Administration (2023) *What is aquaculture?*  
Available at:  
<https://oceanservice.noaa.gov/facts/aquaculture.html>  
(Accessed: 10 April 2024)
- [27] Olsen, A. (2022). Tannhjul. MAST2002. Mechanisms and dynamics.  
Available at: [https://ntnu.blackboard.com/ultra/courses/\\_34512\\_1/cl/outline](https://ntnu.blackboard.com/ultra/courses/_34512_1/cl/outline)  
(Accessed: 18 March 2024)
- [28] Olsen, A. (2022) *Kinematikk Representation of Connected Gears.*[Digital Image]  
Available at:  
[https://ntnu.blackboard.com/ultra/courses/\\_34512\\_1/cl/outline](https://ntnu.blackboard.com/ultra/courses/_34512_1/cl/outline)  
(Accessed: 18 March 2024)
- [29] Knut Utheim, (n.d.), Optoscale 2023 *Accurate data leads to profitable decisions*  
Available at:  
<https://optoscale.no/?lang=en>  
(Accessed: 13 May 2024)
- [30] *Pinout Diagram*(n.d.)[Digital Image]  
Available at:  
<https://store.arduino.cc/products/arduino-mega-2560-rev3>  
(Accessed: 17 April 2024)
-

- 
- [31] Rao, R (2023) *Gantry Systems: A Comprehensive Guide to Understanding and Implementing Gantry Technology*.  
Available at:  
<https://www.wevolver.com/article/gantry-systems-a-comprehensive-guide-to-understanding-and-implementing-gantry-technology>  
(Accessed: 1 March 2024)
- [32] *Reynolds number* (2003)  
Available at:  
[https://www.engineeringtoolbox.com/reynolds-number-d\\_237.html](https://www.engineeringtoolbox.com/reynolds-number-d_237.html)  
(Accessed: 8 February 2024)
- [33] Robocraze(n.d) *What is a Motor Driver?*  
Available at:  
<https://robocraze.com/blogs/post/what-is-motor-driver>  
(Accessed: 18 January 2024)
- [34] Electrical4U(2023) *Servo Motor: Definition, Working Principle, and Applications*  
Available at:  
<https://www.electrical4u.com/what-is-servo-motor/>  
(Accessed: 18 January 2024)
- [35] *Simplified H-bridge*(2018) [Digital illustration] Available at:  
<https://www.build-electronic-circuits.com/h-bridge/>  
(Accessed: 25 January 2024)
- [36] *Specs* (n.d.) [Digital illustration].  
Available at:  
<https://www.gobilda.com/2000-series-5-turn-dual-mode-servo-25-2-torque/>  
(Accessed: 15 April 2024)
- [37] *Strength of Materials* (n.d)  
Available at: <https://mechanicalc.com/reference/strength-of-materials>  
(Accessed 4 April 2024)
- [38] *The Bioscope*(2023)[Digital image]  
Available at:  
<https://optoscale.no/?lang=en>  
(Accessed: 16 May 2024)
-

- 
- [39] *The Engineering Toolbox* (2004)  
Available at:  
[https://www.engineeringtoolbox.com/water-dynamic-kinematic-viscosity-d\\_596.html](https://www.engineeringtoolbox.com/water-dynamic-kinematic-viscosity-d_596.html)  
(Accessed: 8 February 2024)
- [40] *The Engineering Toolbox* (2003)  
Available at:  
[https://www.engineeringtoolbox.com/water-density-specific-weight-d\\_595.html](https://www.engineeringtoolbox.com/water-density-specific-weight-d_595.html)  
(Accessed: 8 February 2024)
- [41] *Types of stress*(n.d.)[Digital illustration]  
Available at:  
<https://mechanicalc.com/reference/strength-of-materials>  
(Accessed: 4 April)
- [42] *Traco Power* (2022) *tib-480-148*  
Available at:  
<https://www.tracopower.com/model/tib-480-148>  
(Accessed: 20 March 2024)
- [43] Wallis, G.B (2011) *Added mass*  
Available at:  
<https://www.thermopedia.com/content/289/>  
(Accessed 18 May 2024)
- [44] *Wokwi*(2023) *Welcome to Wokwi*  
Available at:  
[https://docs.wokwi.com/?utm\\_source=wokwi](https://docs.wokwi.com/?utm_source=wokwi)  
(Accessed: 20 April 2024)
- [45] *Wokwi*(2024)[Digital illustration]  
Available at:  
<https://wokwi.com/projects/395493553900803073>  
(Accessed: 23 April 2024)

---

## Appendices

**Appendix[A] - Mechanical components**

**Appendix[B] - Mechanics and Physics**

**Appendix[C] - Table with Drag Coefficients**

**Appendix[D] - Motor specification**

**Appendix[E] - Fusion 360 sketches**

**Appendix[F] - Prototyping**

**Appendix[G] - Electrical**

**Appendix[H] - CFD Analysis**

**Appendix[I] - Budget**

**Appendix[J] - Risk Assessment**

**Appendix[K] - AI-declaration**



---

## Appendix[A] - Mechanical components

### The in place railway system at the start of the project

The system is built of a C-profile within a U-profile, with aluminium profile strut atop of the slider with three rollers attached to a plate which can slide along the C-profile like shown in figure A1 and A2.

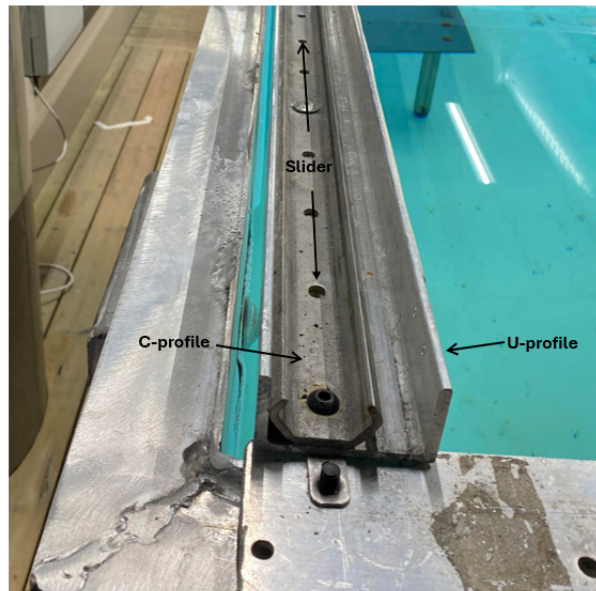


Figure A1: Profiles on The Railway

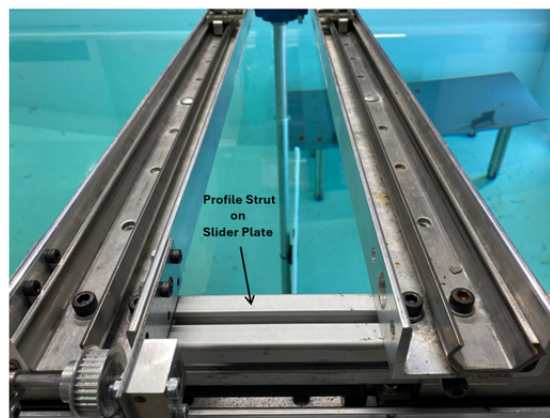


Figure A2: Aluminum Profile Strut atop of Slider Plate

The exact profiles and sliders:

Aluminium U-profil 6082-T6 ([norskstaal.no](http://norskstaal.no))

C-rail ([rollco.no](http://rollco.no))

Slider Plate ([rollco.no](http://rollco.no))

Aluminium Profile Strut ([no.rs-online.com](http://no.rs-online.com))

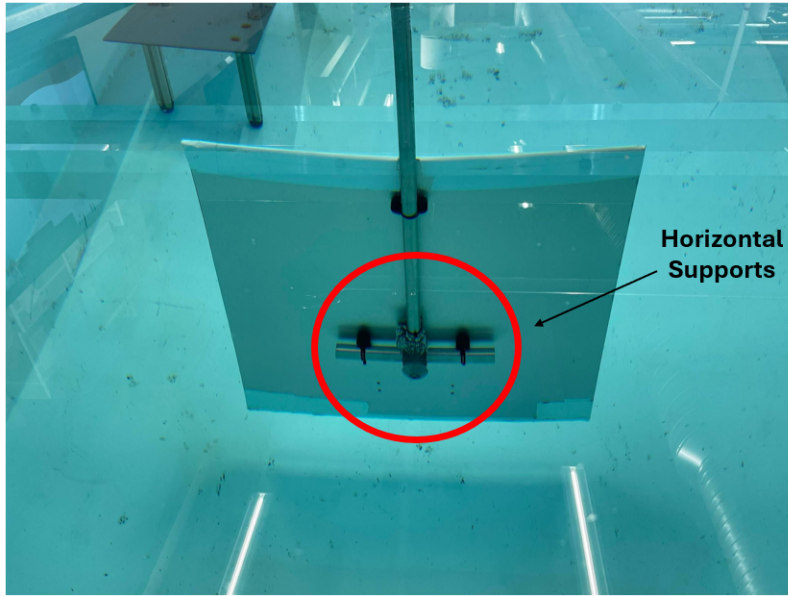


Figure A3: Horizontal Supports

Appendix[B] - Mechanics and Physics

**Table 1. Consistent units.**

Quantity	SI	SI (mm)	US Unit (ft)	US Unit (inch)
Length	m	mm	ft	in
Force	N	N	lbf	lbf
Mass	kg	tonne ( $10^3$ kg)	slug	$\text{lbf s}^2/\text{in}$
Time	s	s	s	s
Stress	Pa ( $\text{N}/\text{m}^2$ )	MPa ( $\text{N}/\text{mm}^2$ )	$\text{lbf}/\text{ft}^2$	psi ( $\text{lbf}/\text{in}^2$ )
Energy	J	mJ ( $10^{-3}$ J)	ft lbf	in lbf
Density	$\text{kg}/\text{m}^3$	$\text{tonne}/\text{mm}^3$	$\text{slug}/\text{ft}^3$	$\text{lbf s}^2/\text{in}^4$

Figure B1: Abaqus Units

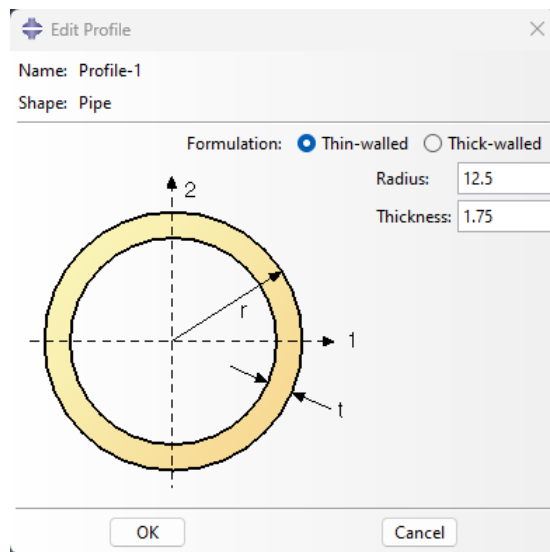


Figure B2: Abaqus Pipe Dimension

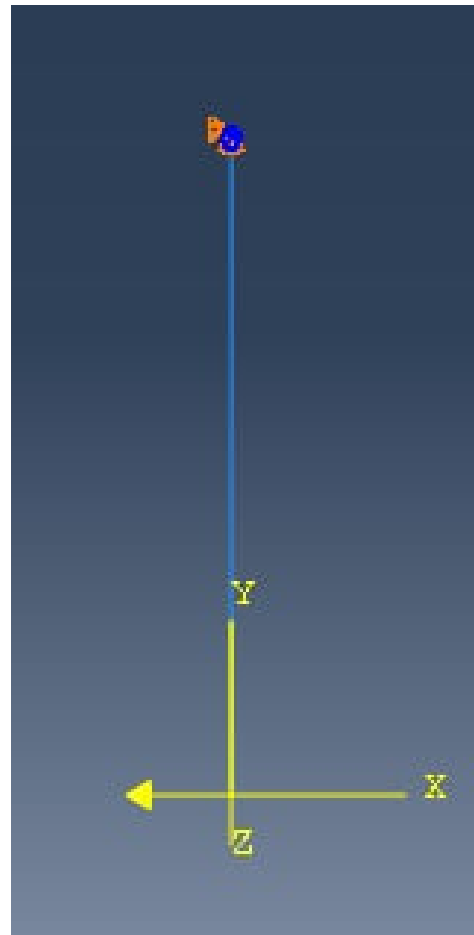


Figure B3: Boundary Conditions for the Beam in Abaqus

```

1 #Drag Force Plot
3
4 import numpy as np
5 import matplotlib.pyplot as plt
7 rho = 998.21 # Density of water      [kg/m^3]
8 C_d = 1.14   # Drag coefficient
9 A = 0.7**2   # Cross-sectional area  [m^2]
11 v = np.linspace(0, 1, 500)
13 F_d = 0.5 * rho * C_d * A * v**2
15 # Maximum stepper output in Newtons
16 max_stepper_output = 240
17
18 # Crossing point
19 crossing_velocity = 0.93
20 crossing_force = 240
21
22 # Create the plot
23 plt.figure(figsize=(10, 6))
24 plt.plot(v, F_d, label='Drag Force')
25 plt.axhline(y=max_stepper_output, color='r', linestyle='--', label='Maximum Stepper
    Output (240 N)')
26 plt.axvline(x=crossing_velocity, color='g', linestyle='--', label=f'Maximum
    velocity(0.93 m/s)')
27 plt.scatter(crossing_velocity, crossing_force, color='red', s=100) # Draw the
    crossing point in red and larger
28 plt.xlabel('Velocity (m/s)')
29 plt.ylabel('Drag Force (N)')
30 plt.title('F_d = 0.5 * rho * C_d * A * v**2')
31 plt.legend()
32 plt.ylim(0, 300)
33 plt.show()

```

Listing 10: Python Code for the Drag Force Plot

```

1 #Pole limitation
3 import numpy as np
5 sigma_y = 210e6 #Yield stress of the
   material [MPa]
a = 1.075 #Length of momentarm [m]
7 A = 0.7**2 #Area of the plate [m^2]
r_o = 12.5e-3 #Outer radius of the pole [
   m]
9 r_i = 10.75e-3 #Inner raduis of the pole [
   m]
A_pole = np.pi*(r_o**2 - r_i**2) #Area of the cross section
   of the pole [m^2]
11 Wy = (np.pi/32)*(((2*r_o)**4 - (2*r_i)**4)/(2*r_o)) #Section modulus for a
   hollow circular cross section [m^3]
rho = 998.21 #Density of water at 20
   degrees C [kg/m^3]
13 Cd = 1.14 #Drag coefferient of the
   plate
15
Fd = round(np.sqrt((sigma_y**2)/((a**2)/(Wy**2) + (16)/(3*(A_pole)**2) ))*10**(-3)
   ,3)
17
v = round(np.sqrt((2*Fd*10**3)/(rho*Cd*A)),3)
19
print(f"The pole can withstand up to {Fd}kN")
21
print(f"The corresponds to a velocity of {v}m/s")

```

Listing 11: Python Code for Calculating Maximum Stress and Velocity for the Pole

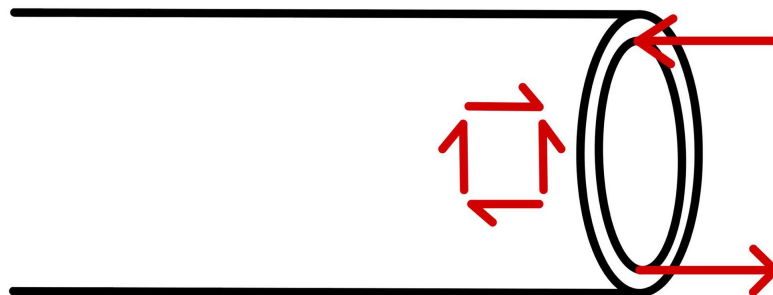


Figure B4: Stress State - Pole

```

#Plate limitation
2
import numpy as np
4
sigma_y = 35e6           #Yield stress of the material [MPa]
6 a = 0.175              #Length of moment arm [m]
b = 0.7                  #Width of the cross section [m]
8 h = 7e-3              #Height of the cross section [m]
A = 0.7**2              #Area of the plate [m^2]
10 Wy = (1/6)*b*h**2     #Sections modulus for a rectangular cross section [m^3]
rho = 998.21            #Density of water at 20 degrees C [kg/m^3]
12 Cd = 1.14            #Drag coeffericient of the plate

14
Fd = round(np.sqrt(((sigma_y**2)/((a**2)/(Wy**2) + (3)/(4*(b*h)**2) ))*10**(-3) ),3)
16
v = round(np.sqrt((2*Fd*10**3)/(rho*Cd*A)),3)
18
print(f"The plate can withstand up to {Fd}kN")
20
print(f"The corresponds to a velocity of {v}m/s")

```

Listing 12: Python Code for Calculating the Maximum Stress and Corresponding Velocity of the Plate

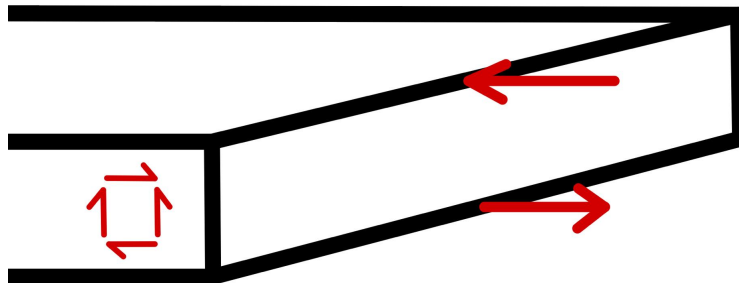


Figure B5: Stress State - Plate

```

2 #Torsional limitation

4 import numpy as np

6 sigma_yield = 210e6           #Yield stress of NS14350 [MPa]
  D_o = 25e-3                   #Outer diameter of the pole [m]
8  D_i = 21.5e-3                #Inner diameter of the pole [m]
  L = 0.35                      #Length of the moment arm [m]
10 A = 0.7**2                   #Area of the plate [m^2]
  Wp = np.pi/16 * ((D_o**4 - D_i**4)/D_o) #Polar section modulus of a pipe [m
    ^3]
12 rho = 998.21                 #Density of water at 20 degrees C [
    kg/m^3]
  Cd = 1.15                     #Drag coefficient of the rotation

14

16 Fd = round((sigma_yield * Wp)/(2*np.sqrt(3)*L),3)

18 v = round(np.sqrt((2*Fd)/(rho*Cd*A)),3)

20 omega = round(v/L,3)

22 print(f"The maximum drag force of the rotation is equal to {Fd}N")

24 print(f"The corresponding maximum velocity at the end of the plate is equal to {v}m
    /s")

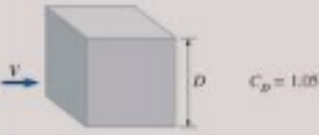
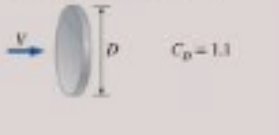

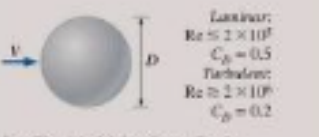
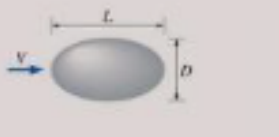
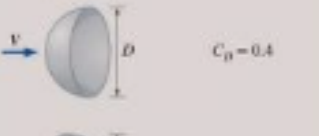
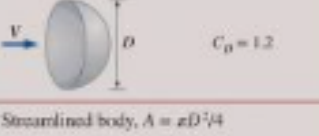
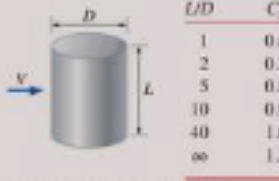
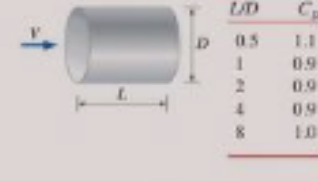


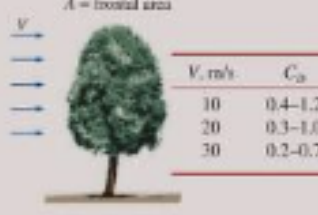

26 print(f"The corresponding maximum angular velocity is equal to {omega}/s")

```

Listing 13: Python Code for Calculating the Maximum Torsional Stress of the Pole

Appendix [C] - Fluid mechanics

**TABLE 11-2**  
 Representative drag coefficients  $C_D$  for various three-dimensional bodies based on the frontal area for  $Re > 10^5$  unless stated otherwise (for use in the drag force relation  $F_D = C_D A \rho V^2 / 2$  where  $V$  is the upstream velocity)

<p>Cube, <math>A = D^2</math></p>  <p><math>C_D = 1.05</math></p>	<p>Thin circular disk, <math>A = \pi D^2/4</math></p>  <p><math>C_D = 1.1</math></p>	<p>Cone (for <math>\theta = 30^\circ</math>), <math>A = \pi D^2/4</math></p>  <p><math>C_D = 0.5</math></p>																										
<p>Sphere, <math>A = \pi D^2/4</math></p>  <p>Laminar:  <math>Re \leq 2 \times 10^5</math>  <math>C_D = 0.5</math>                      Turbulent:  <math>Re \geq 2 \times 10^5</math>  <math>C_D = 0.2</math></p> <p>See Fig. 11-36 for <math>C_D</math> vs. <math>Re</math> for smooth and rough spheres.</p>	<p>Ellipsoid, <math>A = \pi D^2/4</math></p>  <table border="1"> <thead> <tr> <th rowspan="2"><math>LD</math></th> <th colspan="2"><math>C_D</math></th> </tr> <tr> <th>Laminar <math>Re \leq 2 \times 10^5</math></th> <th>Turbulent <math>Re \geq 2 \times 10^5</math></th> </tr> </thead> <tbody> <tr> <td>0.75</td> <td>0.5</td> <td>0.2</td> </tr> <tr> <td>1</td> <td>0.5</td> <td>0.2</td> </tr> <tr> <td>2</td> <td>0.3</td> <td>0.1</td> </tr> <tr> <td>4</td> <td>0.3</td> <td>0.1</td> </tr> <tr> <td>8</td> <td>0.2</td> <td>0.1</td> </tr> </tbody> </table>	$LD$	$C_D$		Laminar $Re \leq 2 \times 10^5$	Turbulent $Re \geq 2 \times 10^5$	0.75	0.5	0.2	1	0.5	0.2	2	0.3	0.1	4	0.3	0.1	8	0.2	0.1							
$LD$	$C_D$																											
	Laminar $Re \leq 2 \times 10^5$	Turbulent $Re \geq 2 \times 10^5$																										
0.75	0.5	0.2																										
1	0.5	0.2																										
2	0.3	0.1																										
4	0.3	0.1																										
8	0.2	0.1																										
<p>Hemisphere, <math>A = \pi D^2/4</math></p>  <p><math>C_D = 0.4</math></p>  <p><math>C_D = 1.2</math></p>	<p>Finite cylinder, vertical, <math>A = LD</math></p>  <table border="1"> <thead> <tr> <th><math>LD</math></th> <th><math>C_D</math></th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0.6</td> </tr> <tr> <td>2</td> <td>0.7</td> </tr> <tr> <td>5</td> <td>0.8</td> </tr> <tr> <td>10</td> <td>0.9</td> </tr> <tr> <td>40</td> <td>1.0</td> </tr> <tr> <td><math>\infty</math></td> <td>1.2</td> </tr> </tbody> </table> <p>Values are for laminar flow (<math>Re \leq 2 \times 10^5</math>)</p>	$LD$	$C_D$	1	0.6	2	0.7	5	0.8	10	0.9	40	1.0	$\infty$	1.2	<p>Finite cylinder, horizontal, <math>A = \pi D^2/4</math></p>  <table border="1"> <thead> <tr> <th><math>LD</math></th> <th><math>C_D</math></th> </tr> </thead> <tbody> <tr> <td>0.5</td> <td>1.1</td> </tr> <tr> <td>1</td> <td>0.9</td> </tr> <tr> <td>2</td> <td>0.9</td> </tr> <tr> <td>4</td> <td>0.9</td> </tr> <tr> <td>8</td> <td>1.0</td> </tr> </tbody> </table>	$LD$	$C_D$	0.5	1.1	1	0.9	2	0.9	4	0.9	8	1.0
$LD$	$C_D$																											
1	0.6																											
2	0.7																											
5	0.8																											
10	0.9																											
40	1.0																											
$\infty$	1.2																											
$LD$	$C_D$																											
0.5	1.1																											
1	0.9																											
2	0.9																											
4	0.9																											
8	1.0																											
<p>Streamlined body, <math>A = \pi D^2/4</math></p>  <p><math>C_D = 0.04</math></p>	<p>Parachute, <math>A = \pi D^2/4</math></p>  <p><math>C_D = 1.3</math></p>	<p>Tree, <math>A = \text{frontal area}</math></p>  <table border="1"> <thead> <tr> <th><math>V, \text{ m/s}</math></th> <th><math>C_D</math></th> </tr> </thead> <tbody> <tr> <td>10</td> <td>0.4–1.2</td> </tr> <tr> <td>20</td> <td>0.3–1.0</td> </tr> <tr> <td>30</td> <td>0.2–0.7</td> </tr> </tbody> </table>	$V, \text{ m/s}$	$C_D$	10	0.4–1.2	20	0.3–1.0	30	0.2–0.7																		
$V, \text{ m/s}$	$C_D$																											
10	0.4–1.2																											
20	0.3–1.0																											
30	0.2–0.7																											
<p>Rectangular plate, <math>A = LD</math></p>  <p><math>C_D = 1.10 + 0.02(L/D + D/L)</math>                      for <math>1/30 &lt; (L/D) &lt; 30</math></p>																												

(continues)

Figure C1: Drag Coefficient Table for Common Geometries (Cengel and Cimbala, 2018, p.624)



## Appendix[D] - Motor Specification

### ◆ DIP setting

Number of subdivisions pulse/rev	2	4	8	16	32	64	128	256	5	10	20	25	40	50	100	200
Sw1	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF	ON	OFF
Sw2	ON	ON	OFF	OFF	ON	ON	OFF	OFF	ON	ON	OFF	OFF	ON	ON	OFF	OFF
Sw3	ON	ON	ON	ON	OFF	OFF	OFF	OFF	ON	ON	ON	ON	OFF	OFF	OFF	OFF
Sw4	ON	ON	ON	ON	ON	ON	ON	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF


Sw5 : motor rotation direction setting      OFF=clockwise      ON=counterclockwise

Sw6 : single and double pulse setting      OFF=pulse+direction      ON=double pulse

Sw7, Sw8 pulse delay setting

pulse delay(ms)	0	4	20	40
Sw7	ON	OFF	ON	OFF
Sw8	ON	ON	OFF	OFF

### ◆ Indicator light and interface description

Sign	Function	Annotation
PWR/FLT	Voltage and fault indicators	Voltage and fault indicator green light flashes: the drive is normal, no pulse signal is received; the green light is always on: the pulse signal is received, the motor is rotating; one red and one green: overcurrent or interphase short circuit fault; two red and one green: no motor detected Or motor wiring error; three red and one green: overvoltage fault; four red and one green: undervoltage fault; five red and one green: tracking error out of tolerance fault.
PUL+	Pulse input signal photoelectric isolation positive terminal	Pulse signal voltage 3.3~24V can be driven
PUL-	Pulse input signal photoelectric isolation negative terminal	The falling edge is valid, and the motor takes one step every time the pulse changes from high to low. Pulse width greater than 2.5 microseconds
DIR+	Direction input signal photoelectric isolation positive terminal	Pulse signal voltage 3.3~24V can be driven
DIR-	Direction input signal photoelectric isolation negative terminal	Used to change the direction of the motor. The falling edge is valid, and the motor takes one step every time the pulse changes from high to low. Pulse width greater than 2.5 microseconds
EN+	Enable input signal photoelectric isolation positive terminal	Enable signal voltage 3.3~24V can be driven
EN-	Enable input signal photoelectric isolation negative terminal	When it is valid (low level), the motor coil current is turned off, the motor is in a free state and the alarm signal is cleared.
ALM+	Alarm signal output positive terminal	When the red light flashes, the alarm signal is valid (the output optocoupler is turned on). ALM+ is connected to the pull-up resistor to the positive pole of the output power supply, ALM- is connected to the negative pole of the output power supply, and the maximum driving current is 10mA.
ALM-	Alarm signal output negative terminal	
EX+	In-position signal output positive terminal	When the driver completes the given pulse, the in-position signal is valid (the output optocoupler is turned on). EX+ is connected to the pull-up resistor to the positive pole of the output power supply, EX- is connected to the negative pole of the output power supply, the maximum driving current is 10mA.
EX-	In-position signal output negative terminal	
VCC	Encoder power supply is positive	Encoder 5V power supply positive terminal
EGND	Encoder power ground	Encoder power ground
EB+	Encoder phase B input positive terminal	Connect to the positive input of the B channel of the encoder
EB-	Encoder phase B input negative terminal	Connect to the negative input of the B channel of the encoder
EA+	Encoder A phase input positive terminal	Connect to the positive input of the A channel of the encoder
EA-	Encoder A phase input negative terminal	Connect to the negative input of channel A of the encoder
V-	Drive power supply negative	EA drive power supply negative
V+	Drive power positive	Drive power positive, 18~90VDC
A+	Motor wiring A+	
A-	Motor wiring A-	
B+	Motor wiring B+	
B-	Motor wiring B-	

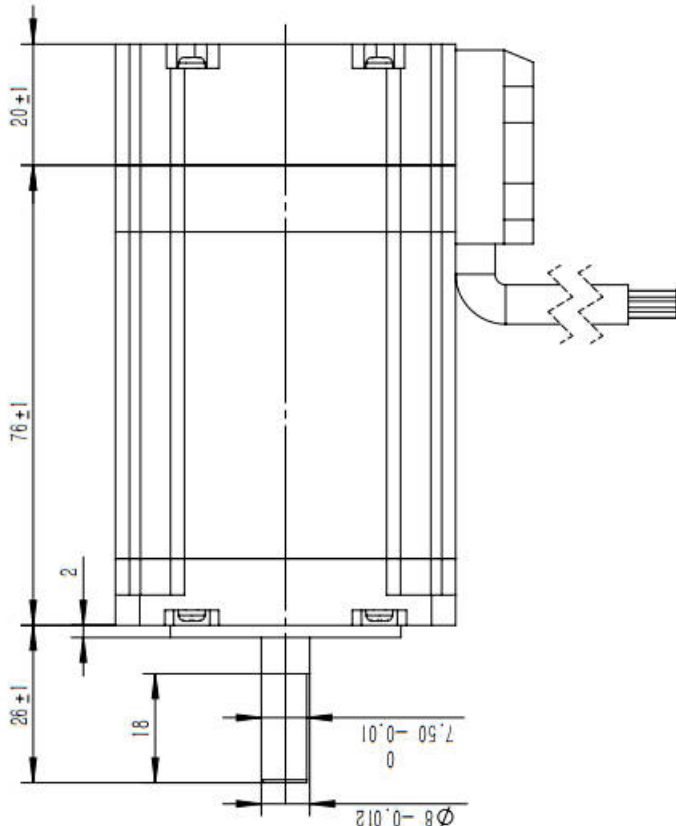
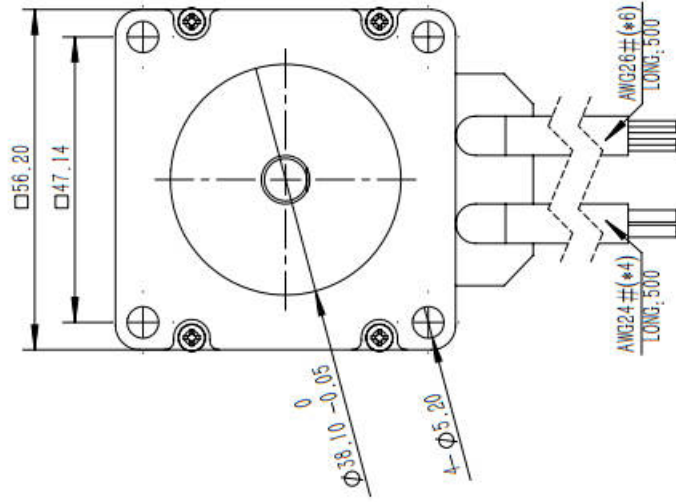


Notice:

1. Please ensure that the wiring of the motor and the encoder is correct, otherwise the motor will alarm when it is powered on and receives pulses.
2. When the motor is installed, it is strictly forbidden to knock the back cover of the motor to avoid damage to the encoder.

Figure D2: Nema23 Driver linked on webpage (Closed loop NEMA23, n.d.)

REV. 版本号	REVISIONS 修订栏	DATE日期
	DESCRIPTION 描述	2021. 3. 27
	首次发布	



Technical parameters 技术参数	
Phases 电机相数	2
Step angle 步距角	Step angle error 步距角误差 0.09mm
Rated voltage 额定电压	1.8° Radial runout max 径向跳动最大值 0.025mm
Rated current 额定相电流	2.4V Axial runout max 轴向跳动最大值 0.075mm
Rated resistance 额定电阻	4.0A Insulation class 绝缘等级 B
Rated inductance 额定电感	0.6Ω ±10% Insulation resistance 绝缘电阻 100MΩ @500VDC
Holding torque 保持力矩	3mH ±20% Operating temperature 工作环境温度 -20℃~+50℃
	Heat rise 温升 80K
DESIGNED BY 设计: 陈榆涛	
DRAWN BY 制图: 57BYGH76	

编码控制 (1000线)

1	白	+5VCC
2	黑	EGED
3	绿	EA-
4	红	EA+
5	蓝	EB-
6	黄	EB+

励磁顺序

STEP	A+	B+	A-	B-
1	+	+	-	-
2	-	+	+	-
3	-	-	+	+
4	+	-	-	+
5	+	+	-	-

CCW ←

→ CW

Figure D1: Nema23 Datasheet

---

## SPECS

Weight	64g
Gear Ratio	300:1
Output Shaft Style	25 Tooth Spline
Voltage Range	4.8V ~ 7.4V
No-Load Speed (4.8V)	0.25sec/60° (40RPM)
No-Load Speed (6.0V)	0.20sec/60° (50RPM)
No-Load Speed (7.4V)	0.17sec/60° (60RPM)
Stall Torque (4.8V)	17.2 kg.cm (240 oz-in)
Stall Torque (6.0V)	21.6 kg.cm (300 oz-in)
Stall Torque (7.4V)	25.2 kg.cm (350 oz-in)
Max PWM Range	500 - 2500 $\mu$ sec
Max PWM Range (Continuous)	900 - 2100 $\mu$ sec
Travel per $\mu$ sec (out of box)	0.90°/ $\mu$ sec
Pulse Amplitude	3-5V
Direction	Clockwise w/ Increasing PWM Signal
Deadband Width	4 $\mu$ sec
Motor Type	Brushed DC
Feedback Style	5K $\Omega$ Potentiometer
Output Shaft Support	Dual Ball Bearing
Gear Material	Steel
Wire Length	300mm
Wire Gauge	22 AWG
Connector Type	3-Pos TJC8 Servo Connector [MH-FC]

Figure D3: Specifications - GoBuilda Servo Motor (Specs, n.d.)

---

## Appendix[E] - Fusion 360 Sketches

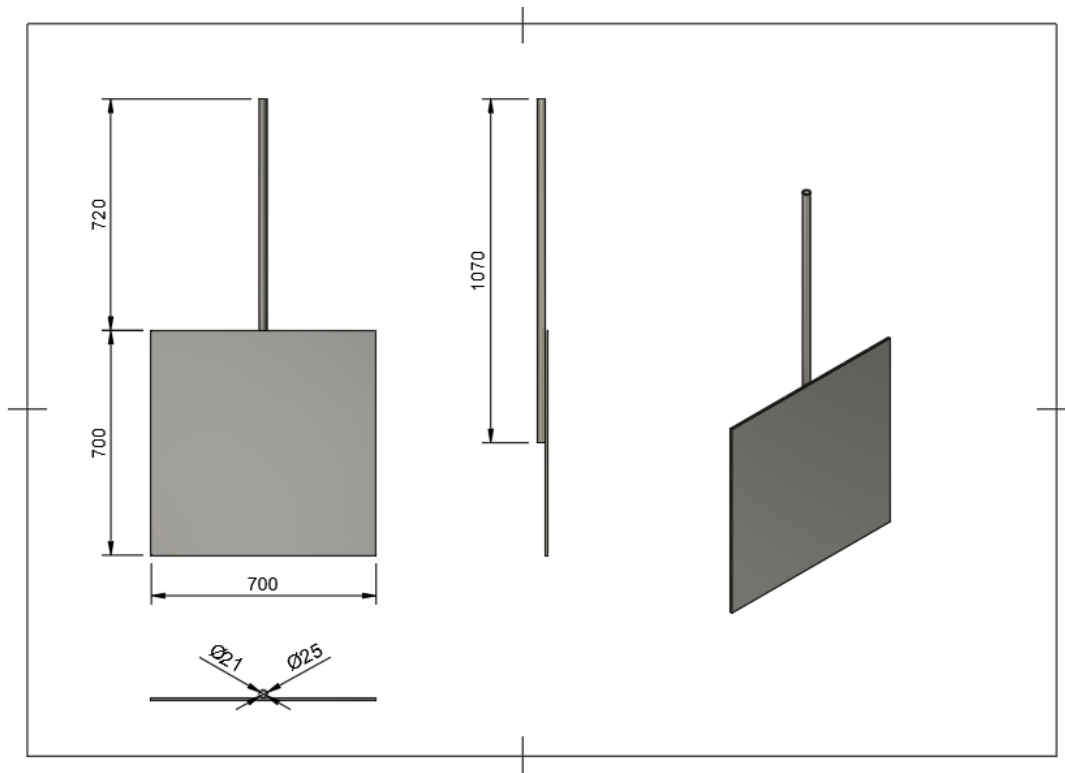


Figure E1: Plate and Pole Dimensions

The servo enclosure was crafted with a series of manual measurements and iterative testing to achieve the perfect fit. The design aimed to accommodate several key features: a secure attachment point for the servo motor, a bottom part capable of being firmly pressed into a steel profile for stability and rigidity, an accessible cable opening, and a cable pocket to shield connections from water splashes. Additionally, protective roof was made to cover the cable pocket and the servo motor. Numerous threaded screw holes are visible on the model to secure mounting for the servo motor, bracket, and roof. These threads were a reliable fastening mechanism, eliminating the risk of screws involuntarily coming loose. A fusion sketch can be seen in figure E2.

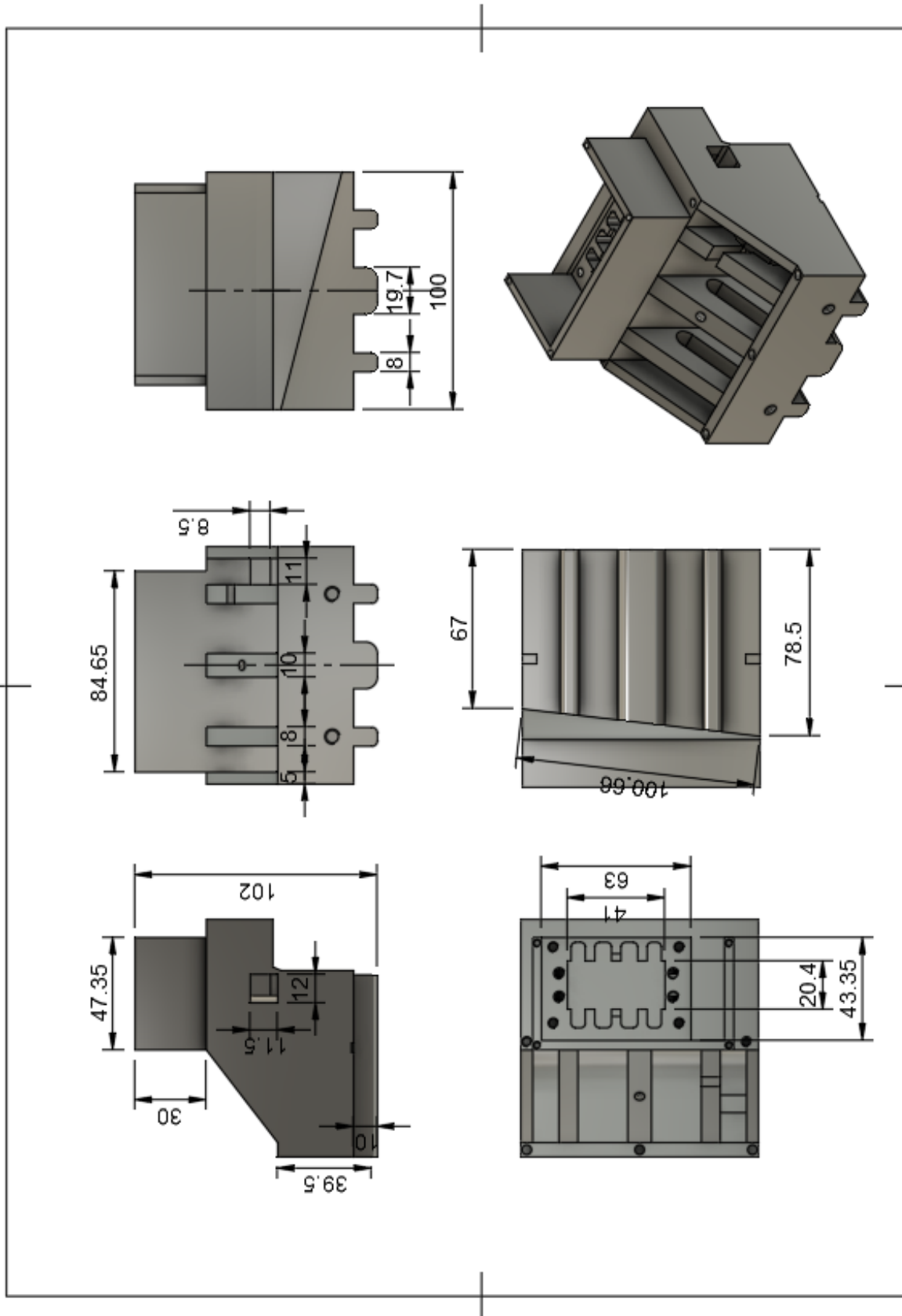


Figure E2: Servo enclosure

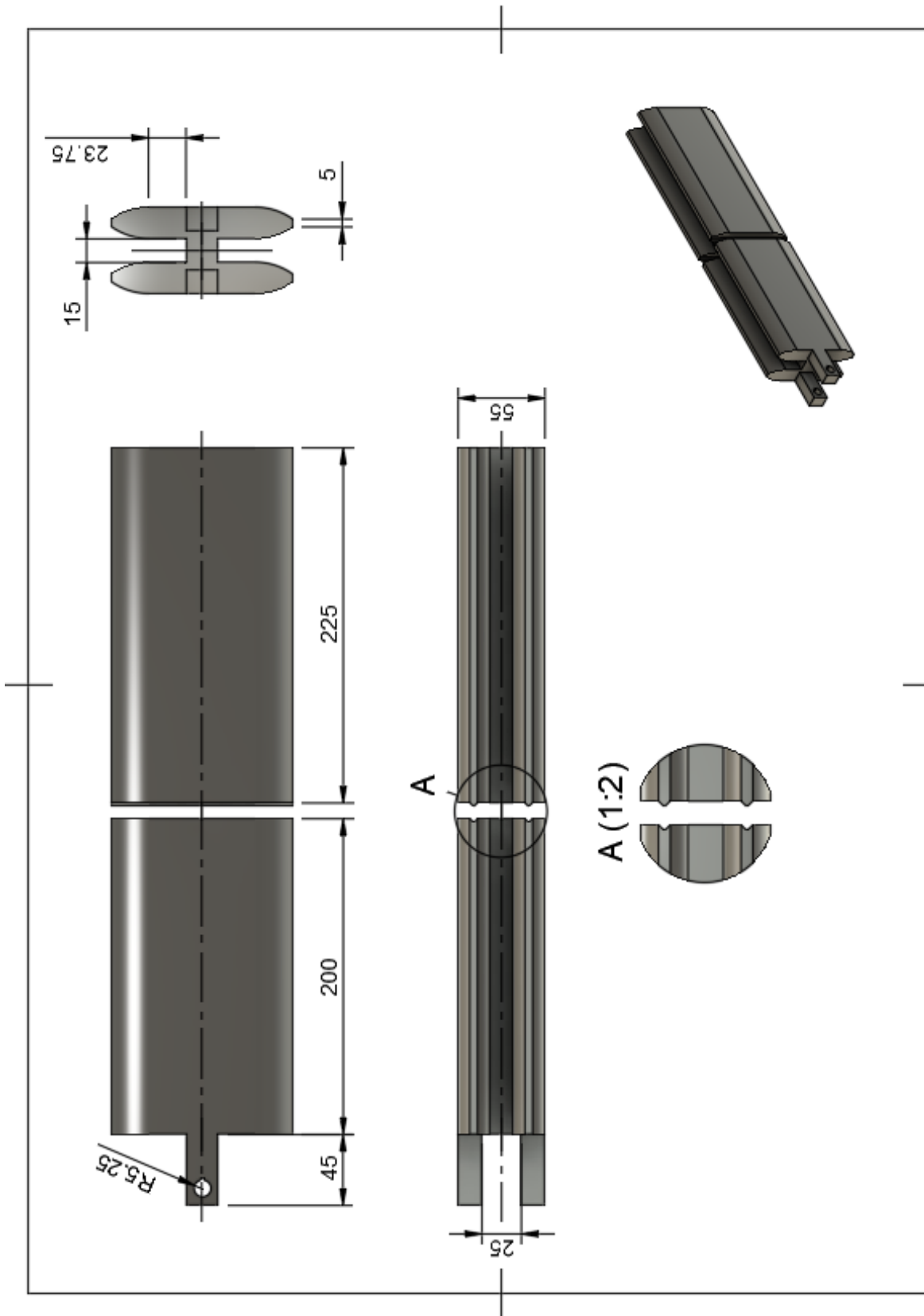


Figure E3: Parallel Railway Set

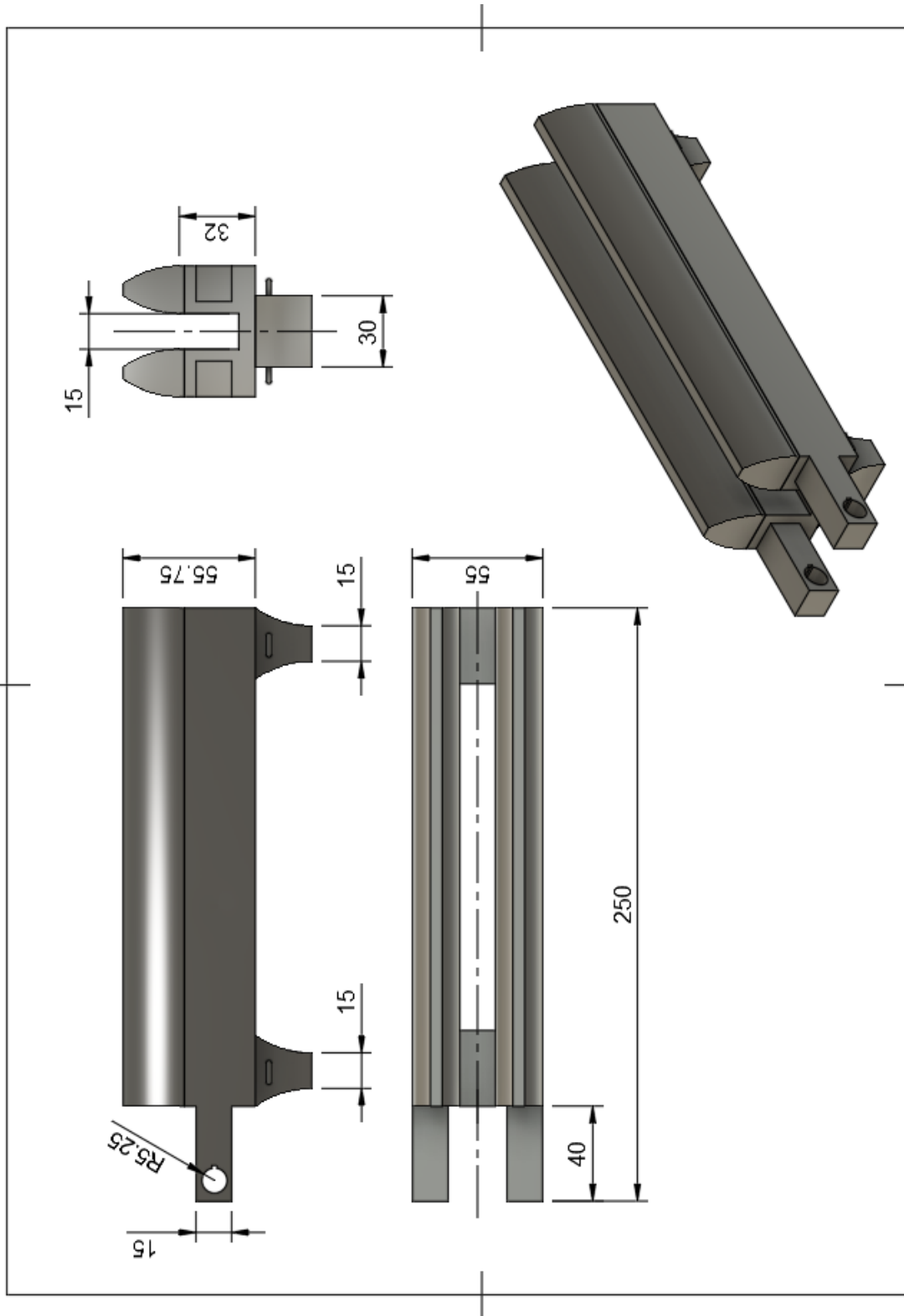


Figure E4: Transverse Railway

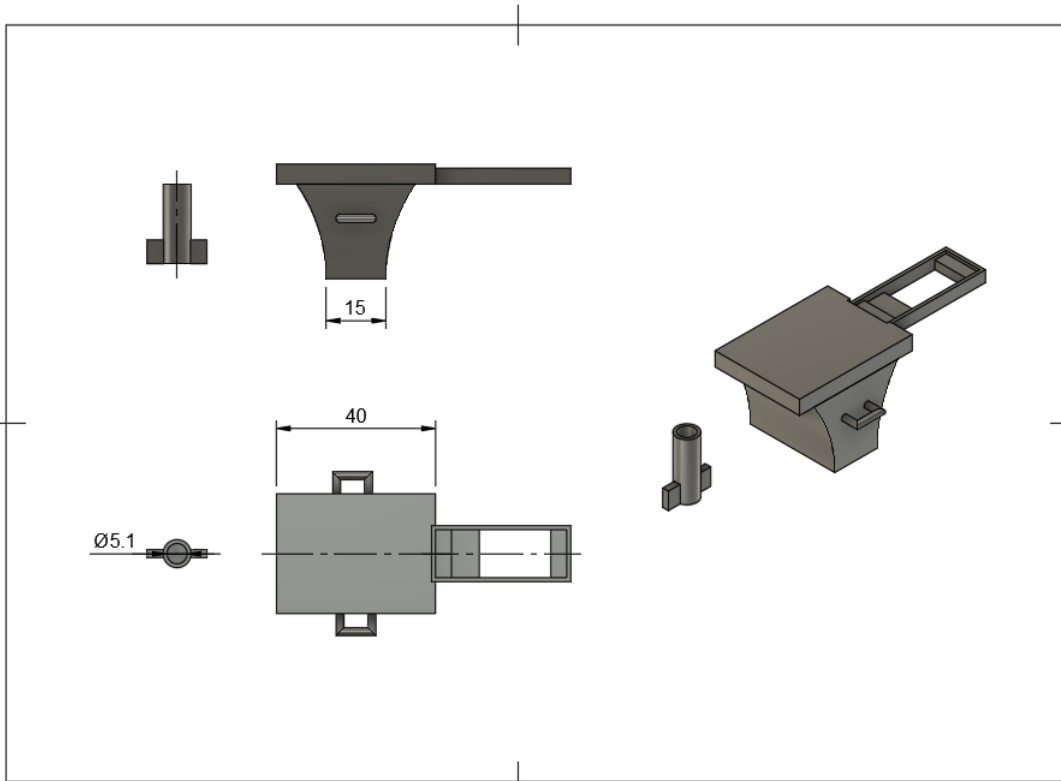


Figure E5: Rotary Solution

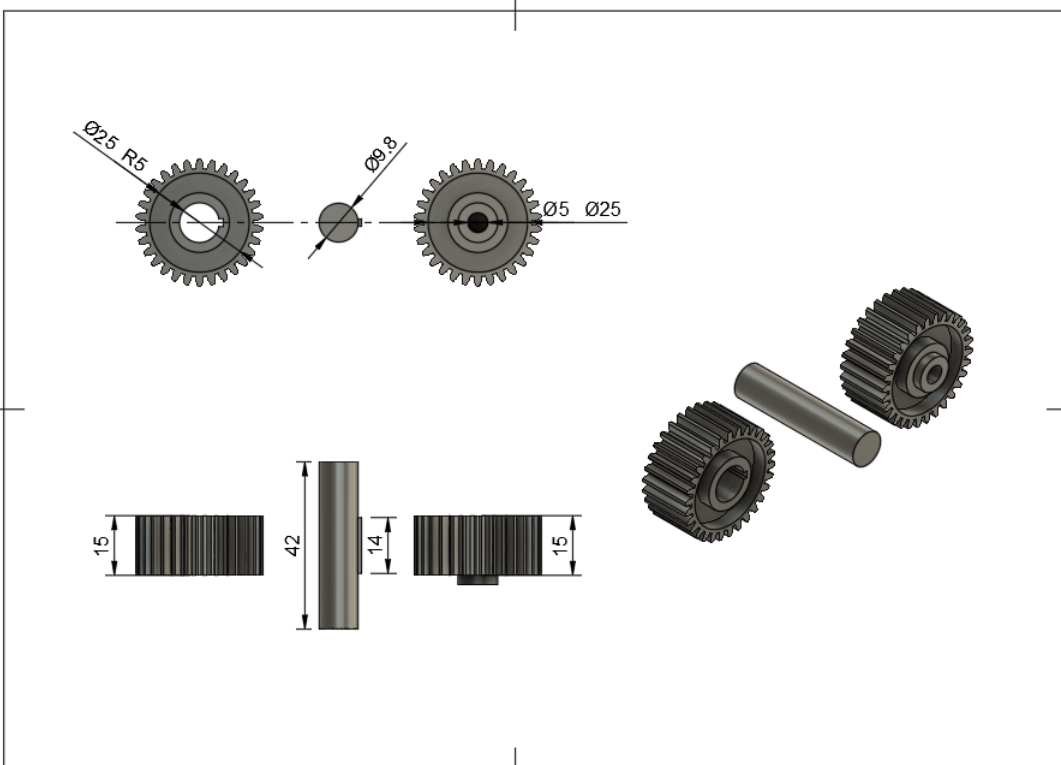


Figure E6: Parallel Railway Set



---

## Appendix[F] - Prototyping

Given the impracticality of testing on the actual large-scale gantry system due to its size and daily usage, a decision was made to construct a Critical Functional Prototype (CFP) for motion and code testing purposes. Specifically, a CFP is a prototype which focuses on the most critical components and functions to mitigate risks involved. This allows flaws in the system to be detected and improved before full-scale implementation. Below some advantages from prototyping can be seen.

- Get started right away
- Reduces risk
- Visualization
- Early error detection
- Saves time and money

### Design principles

The prototype in question was to be built of the same key components as the real system, just on a smaller scale and without the need for robust electrical components. The prototype was to function as a blueprint for the full-scale model. For sketching and design process, the CAD software Fusion 360. In addition, three small stepper motors and a small servo motor were used to drive motion.

To ensure an accurate functional model that represented the actual model, the design was to be based on the motion principles that the system undergoes. In total there are three motion principles, firstly linear motion along the length of the pool driven by two parallel railways, secondly the transverse motion which is driven by a railway bridge connected to the parallel railways and lastly the rotational movement of a plate connected to the transverse motion. An illustration of the movement principles can be seen in figure F1 .

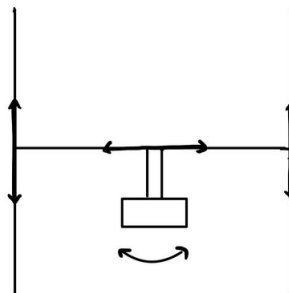


Figure F1: Movement Principles

---

## Design Process

The prototype designs were made on the basis of the motion principles stated above. For longitudinal motion, establishing a stable foundation with solid footing was important, given its role in supporting the entire model. Drawing inspiration from the letter H, the railway shapes was designed. To accommodate the requirements of code testing, it became necessary for the railway to extend beyond the length of the 3D printing plate. Consequently, for each parallel railway set, two railway segments were printed separately and later joined by gluing. For each set, one of the railways printed had an additional axle attachment to accommodate the pulley mechanism at one end. A sketch of the design can be seen in figure F2.

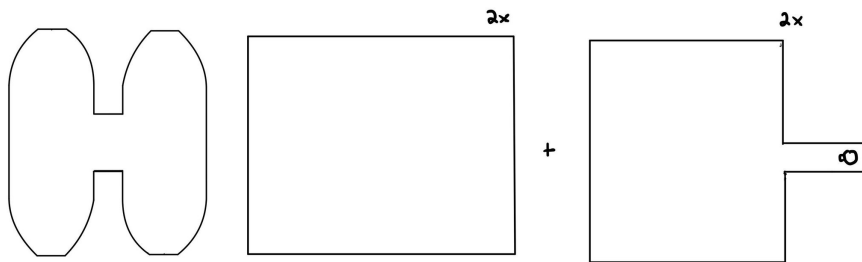


Figure F2: Railways Along the Length

For the transverse motion the shape of the railway was halved and equipped with two sliders fitted for the parallel railways in the longitudinal direction. The railway had likewise additional attachment to accommodate the pulley mechanism at one end. A sketch of the design can be seen in figure F3.

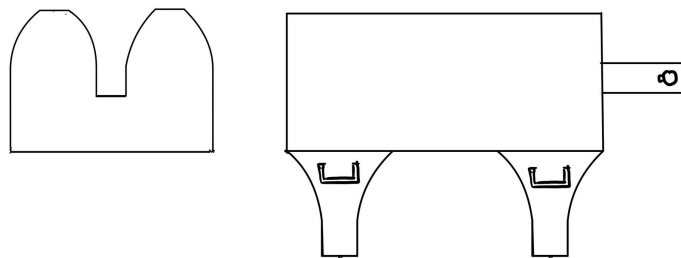


Figure F3: Bridge Railway

The rotary motion was designed by making a slider fit for the transverse railway with an additional servo mount, providing a stable platform for the servo motor. The servo would drive the rotational movement attached to a sign connected to the servo axle. A sketch of the design can be seen in figure F4.

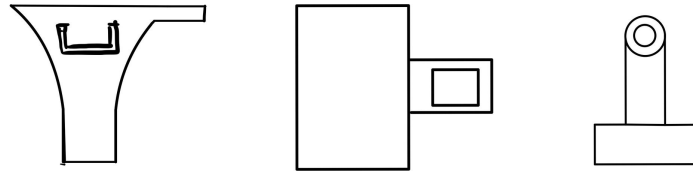


Figure F4: Rotary Motion

To drive the automated motion for the system a belt driven pulley mechanism was used. This setup utilized a pair of gears: one gear was affixed to an axle on the railway attachment at one end, while the other gear was mounted to a stepper motor at the opposite end of the railway. These were then driven by a belt passing over them. A sketch of the design can be seen in figure F5.

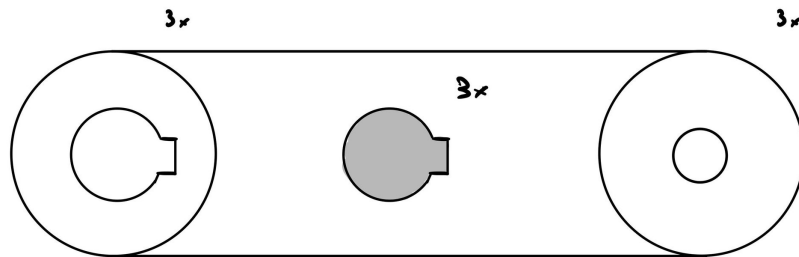


Figure F5: Belt-driven Pulley System

In figure 27 a fully assembled fusion model of the system based on the sketches above can be seen. Additionally the model has motion links were slider motion can be simulated. By putting in angel inputs for the gears and motion links a functioning motion analysis can be simulated.

# Appendix[G] - Electrical

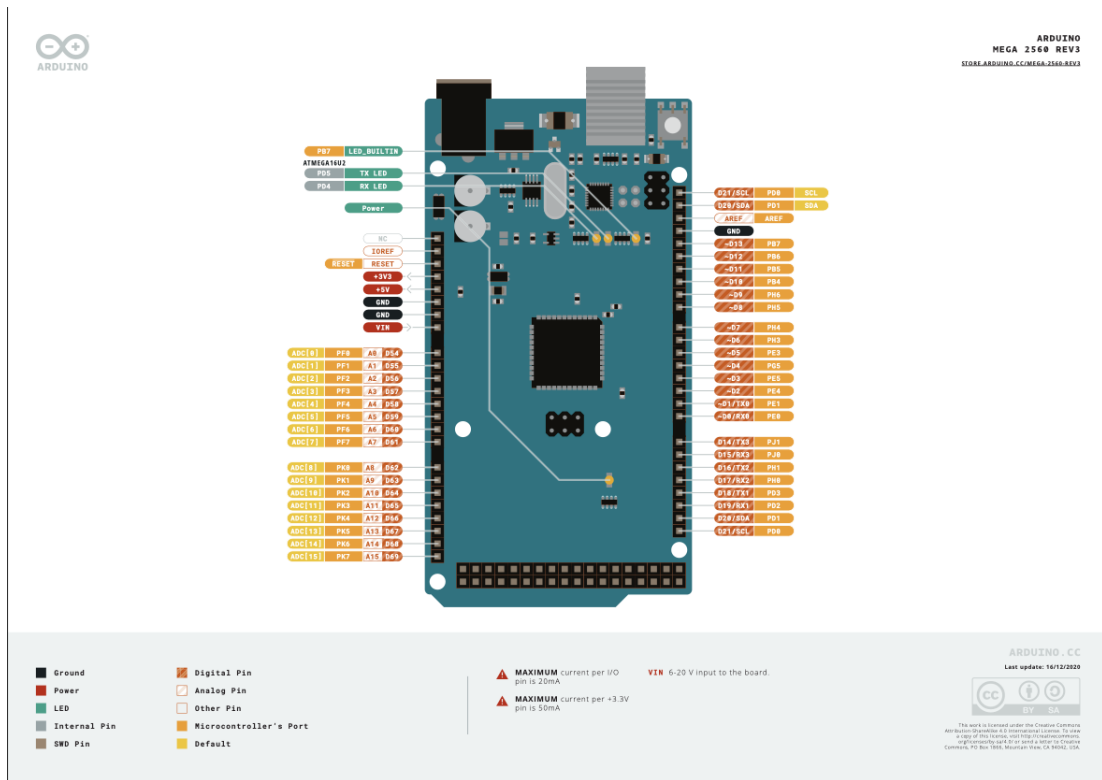


Figure G1: Arduino Mega Pins (Pinout Diagram, n.d.)

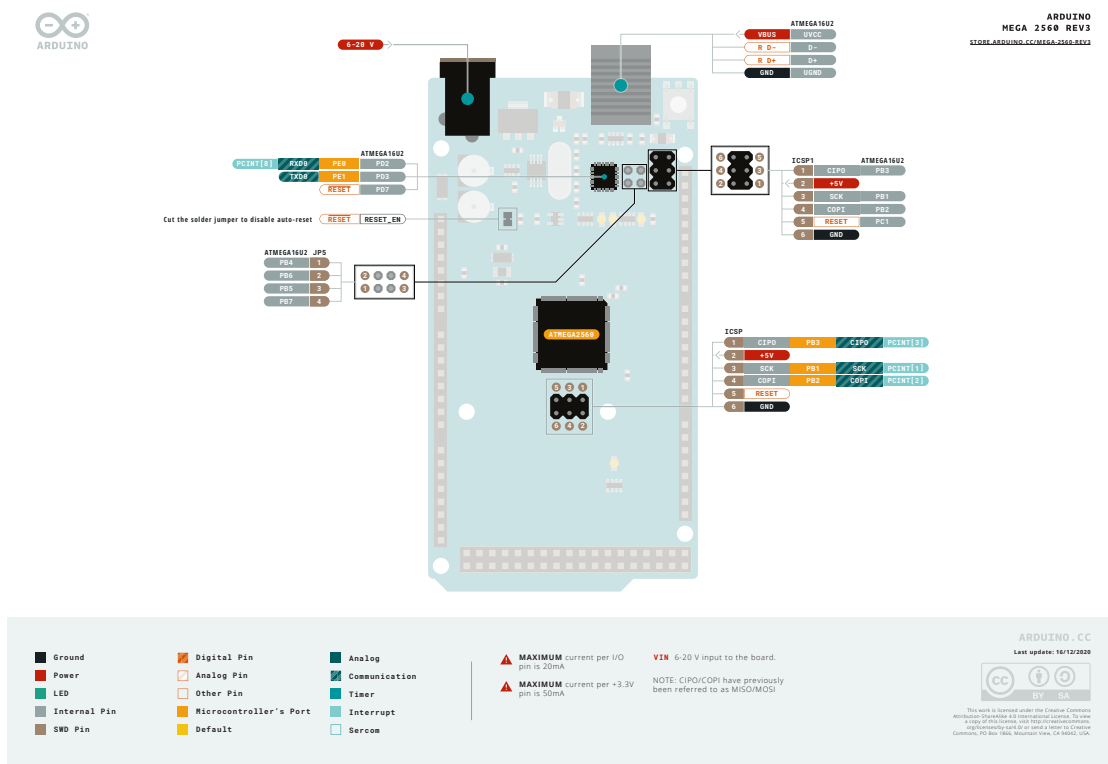


Figure G2: Arduino Mega Pins (Pinout Diagram, n.d.)

---

## Appendix[H] - CFD Analysis

During the project it was attempted to utilize CFD programs to more accurately conclude the drag forces enacting on the plate. The group hoped that CFD analysis would enable the use of more complex fluid mechanical phenomenon such as vortex shedding. The use of CFD analysis was however not pursued due to the inexperience and high level understanding needed to accurately simulate results.

The following simulation was created by implementing a CAD of the plate and pole into Autodesk CFD program. A large geometry was defined to encapsulate the flow around the object. Boundary conditions of the simulations entailed setting the walls to a slip/symmetry while an inflow was set to 77 cm/s and outflow set to 0 static gauge pressure. Figures H1 and H2 showcase the attempted simulation. The simulation used 500 iterations and K-epsilon turbulence model.

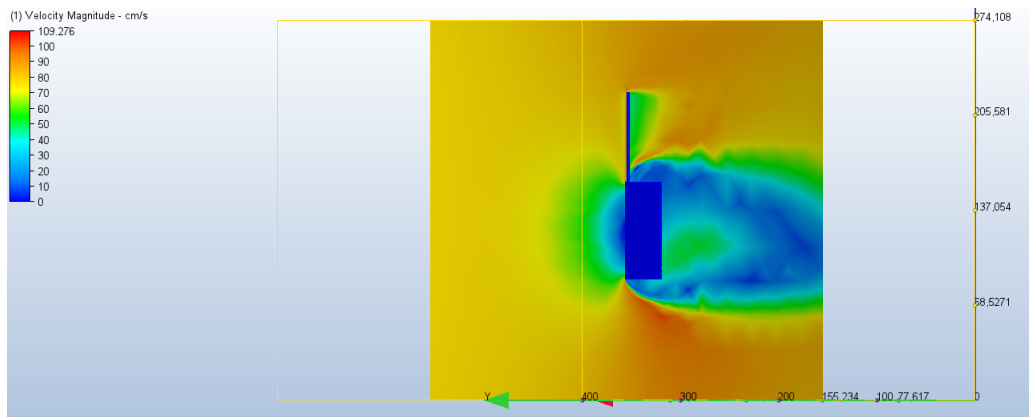


Figure H1: CFD Analysis

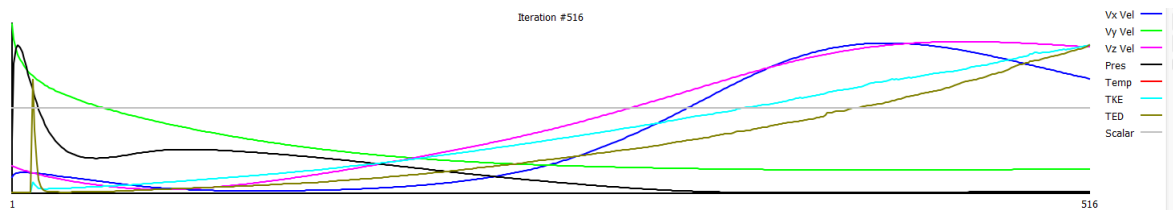


Figure H2: Convergence Plot

## Appendix[I] - Budget

The given budget for the calibration system was 30000 *kr*. Figure I1 illustrates the cost sheet for the project. Components mentioned in the cost sheet were bought from various online stores. Other components such as timing pulleys and belts were excluded from the sheet due to them being available at the company laboratory. The resulting cost of crating the calibration system is far below the given maximum budget.

ITEM	COST	UNITS	TOTAL COST
NEMA23 STEPPER MOTOR AND CL57 MOTOR DRIVER	kr 2 165,00	3	kr 6 495,00
ARDUINO MEGA	kr 467,00	1	kr 467,00
GOBILDA STEPPER MOTOR	kr 700,00	1	kr 700,00
CLAMPING HUB	kr 86,00	1	kr 86,00
100 THEETH GEAR	kr 160,00	1	kr 160,00
20 THEET GEAR	kr 107,00	1	kr 107,00
AC/DC 48V TRANSFORMER	kr 3 663,00	1	kr 3 663,00
DC/DC TRANSFORMER	kr 392,00	1	kr 392,00
SERVO PLATE	kr 54,00	1	kr 54,00
JOYSTIC	kr 321,00	1	kr 321,00
BUTTON	kr 47,00	2	kr 94,00
CABLE CHAIN 10X20mm	kr 90,00	3	kr 150,00
CABLE CHAIN 10X10mm	kr 75,00	2	kr 150,00
L-SHAPE MOTOR BRACKETS	kr 60,00	2	kr 120,00
FLAT MOTOR BRACKETS	kr 119,00	1	kr 119,00
SMALER COMPONENTS	kr 500,00	1	kr 500,00
<b>SUM OF TOTAL COST</b>		<b>kr</b>	<b>13 578,00</b>

Figure I1: Cost Sheet

---

## Appendix[J] - Risk Assessment

This project is completed by four mechanical engineering students who are very very familiar with each other from previous group assignments in various courses including MAST2002, ISTT1002, MAST1002 and TPK5100. The collective experience of the team has been positive. The goals and ambitions of the members are similar and the group dynamic will not be the cause of any issues throughout the project. The expected risks are separated into 3-4 categories which are evaluated in the following paragraphs.

### Components

The project entails working with both mechanical and electrical components. This includes potential risks such as finding sufficient parts to integrate into the system. It is paramount that the components are able to perform their designated tasks, and if the right precautions are not taken there is a small risk that this can happen. If this were to happen, it would be necessary to find new and improved components to order before the deadline, which becomes a greater issue further into development. This risk however, can be reduced through mechanical calculations and seeking advice from someone with experience. After the components are integrated, they can potentially become damaged due to various issues such as great forces, high currents or coming in contact with water, and these situations could be detrimental to the final result considering the time frame and the given budget. It is therefore significant to take extra care during implementation.

### Assembly and Installation

The assembly of the system will take place in Optoscale's workstead with access to various types of machinery. Here precautions such as hearing aid and safety glasses must be worn to insure safety, and can have very serious consequences should they not be upheld. This however, is not considered a likely scenario as the project will be supervised, and the group will have receive aid in use. Another important aspect of implementation is electrical work. This includes wiring, soldering and coupling. The consequences of faulty electrical work can vary from mild consequences such as insufficient voltage supply, or more concerning outcomes where components are damaged. Optoscale has electricians available to guide and oversee this process, as well as intervening during setup which will reduce these risks, but most of the work will be completed by the team members and some complications can be expected.

---

## Technical Aspects and Outside Factors

The scope of the work is extensive and mistakes can be made outside of the physical labor. Many factors can therefore contribute to end up behind schedule and risk not completing the project inside the time frame. The programming aspect of the work is very technical and it can be very exhausting locating errors in the code, and creating the optimal script. The chances of finding errors in the programming is expected, but through the use of a prototype or other tools, it is likely that this will be solved before applying it the finished model. Additionally it is possible that manual calculations do not take everything into account and can cause unexpected problems, but this can be greatly reduced by including safety factors before placing orders. Table 5 shows the risks and consequences of the work involved with the project.

<b>P r o b a b i l i t y</b>	5. Very likely					
	4. Likely	Programming difficulties	Faulty wiring	Difficulties implementing components	Electrical difficulties	Insufficient Soldering
	3. Moderate	Unaccurate calculations		Components in contact with water	Behind schedule	
	2. Unlikely	Misalignment of rails	Inadequate parts	Late arrival of parts	Requireing new components	Damaging components
	1. Very unlikely	Internal conflict			Budget issues	Hearing damage
		1. Very small	2. Small	3. Moderate	4. Serious	5. Very serious
<b>Consequence</b>						

Table 5: Risk Assessment Diagram



## Declaration of AI aids and -tools

Have any AI-based aids or tools been used in the creation of this report?

No

Yes

If yes: please specify the aid/tool and area of use below.

### Text

**Spell checking.** Are parts of the text checked by:  
*Grammarly, Ginger, Grammarbot, LanguageTool, ProWritingAid, Sapling, Trinkai.ai* or similar tools?

**Text generation.** Are parts of the text generated by:  
*ChatGPT, GrammarlyGO, Copy.AI, WordAi, WriteSonic, Jasper, Simplified, Rytr* or similar tools?

**Writing assistance.** Are one or more of the reports ideas or approach suggested by:  
*ChatGPT, Google Bard, Bing chat, YouChat* or similar tools?

If yes, use of text aids/tools apply to this report - please specify usage here:

AI was used to modify LaTeX layout, including figures and tables. AI was also used for general inspiration for thesis layout.

### Codes and algorithms

**Programming assistance.** Are parts of the codes/algorithms that i) appear directly in the report or ii) have been used to produce results such as figures, tables or numerical values been generated by: *GitHub Copilot, CodeGPT, Google Codey/Studio Bot, Replit Ghostwriter, Amazon CodeWhisperer, GPT Engineer, ChatGPT, Google Bard* or similar tools?

If yes, use of programming assistance aid/tools apply to this report - please specify usage here:

At the initial start of the coding phase, ChatGPT served as a valuable tool for establishing an initial framework and ideas for the codebase.

### Images and figures

**Image generation.** Are one or more of the reports images/figures generated by:  
*Midjourney, Jasper, WriteSonic, Stability AI, Dall-E* or similar tools?

If yes, use of image generator aids/tools apply to this report - please specify usage here:

**Other AI aids or tools.** Have you used other types of AI aids or -tools in the creation of this report?  
If yes, please specify usage here:

I am familiar with NTNU's regulations: *Submitting a report generated with the assistance of AI tools and claiming the work to be partially or fully my own, is not permitted. I therefore declare that any use of AI aids or tools are explicitly stated i) directly in the report or ii) in this declaration form.*

Thanish Gnanasegaram / 19.05.24/Tondheim Benjamin Brađavic 14.05.2024 Trondheim

Andreas Helberg / 14.05.24/Tondheim Håkon Eimarsen 14.05.2024 Trondheim

Signature / Date / Place

Figure K1: Signed AI-declaration

