Chapter 1

Introduction

The Loomo from Segway Robotics is a quite unique platform, combining both a robot and a transportation device. It covers most wishes one can have from a development platform within the robotics field of engineering. This Bachelor's thesis will delve into the development of manipulators compatible of being installed on the Loomo, without interfering with the transporting capability that makes the Loomo so unique. The manipulator was designed in such a way as to give extra reach to the small-scale robot-development platform. Addressing the fundamental design considerations, component selection, control-methods necessary to develop a functional, responsive, and capable robotic system was done in the product development process.

1.1 Task objective

The following Bachelor's thesis main focus and task case:

Design, produce, and equip the robotic manipulators to the Loomo while allowing further development
This includes working on the following: Designing, constructing, and evaluating the integration of manipulators onto the Loomo robot platform to expand its functionality and enable object manipulation.

1.2 Focus and methodology

In this project report, critical design considerations for building a manipulator, including payload capacity, range, accuracy, and workspace will be elaborated upon.
Basic components of a robotic arm, such as actuators, joints, linkages, end effectors, sensors, and power supplies, and their selection criteria will then be delved deeper into.
Furthermore, the choice of appropriate actuators and mechanical design in detail will be discussed.

By following this project report, readers will gain an understanding of the design consideration and choices that was done developing a manipulator to be equipped on the Loomo Segway, opening it up for new possibilities. Such as pick- and place-operations and pushing elevator buttons.

Chapter 2

Concept and Concept evaluation

This section will describe design choices and evaluations for the different concepts relating to the design of the manipulator.

2.1 Design specifications

To make the best possible robot manipulators, many choices must be made. The design specifications define what to base the evaluation on. Together with cost, time, and personal preference it defines the deciding factors and choices that need to be made.

Design specifications:

- The manipulator must be able to reach buttons in elevators and on doors
- The manipulator with end effector (e.e.) should be able to lift a load of 1 kg
- All original features should be preserved
- The responsiveness and speed of the manipulator must be reasonable
- It must be safe to use, meaning it introduces negligible extra harm
  - It should be possible to stand and drive the robot with extra equipment mounted on
  - No cameras, sensors, etc should be blocked by the manipulators
  - No other features should be blocked or ruined
- It must be able to drive (and therefore balance) on its own with all extra equipment mounted on the robot
- No irreversible modifications should be done to the robot, i.e. no destroying/cutting or otherwise permanently changing the original parts
- The manipulators should be easy to develop further and therefore be of use to future UiA students and staff
- The weight of the extra equipment must not limit who can use the robot to a large degree, max total carrying capacity of the Loomo is 100 kg if the extra equipment would weigh 20 kg, there are a lot of people that could not use it anymore [11]
- It should be easy to fix and replace components

2.2 Length of the manipulators

At the very beginning of the project, one decision seemed to be key to how the rest of the project would lay out;

Should the manipulator without any extensions be long enough to hit the elevator buttons one meter up on the wall?

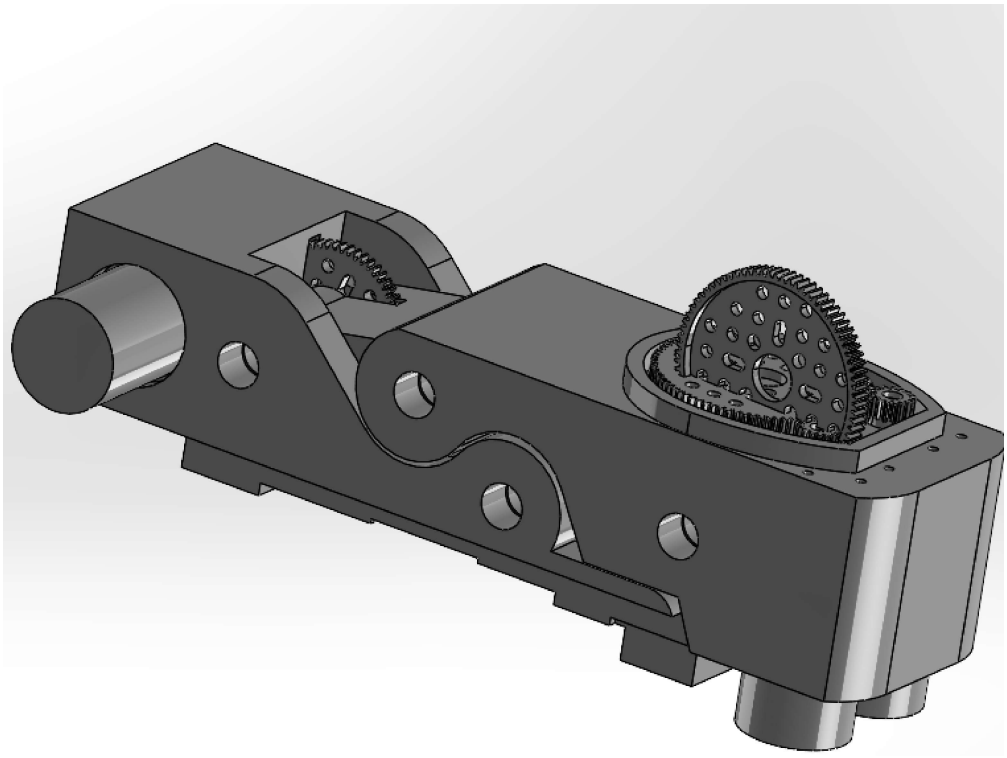
If so, the manipulator would have to be long. About the length of an adult human's arms, or about 75 cm, on a robot which would be positioned about 40 cm off the ground. This would look very disproportionate, and the manipulator itself would look very slender if thickness would not be scaled to fit the length. Another issue is that motors, joints, and links need to be strong enough to support the weight of the manipulator in addition to the specified max load and safety.

On the other hand, a short manipulator would in many ways defeat its purpose. The whole point of installing manipulators would be to make the Loomo able to traverse into different floors, and back, without external help. Short manipulators will not be able to do this. If the manipulators could be short most of the time, and extend only when necessary, short manipulators would be a viable option. In other words, introducing an extending portion of the manipulators. The main concern about this is that the mechanism to do this could be bulky and make the manipulator heavy. Potentially requiring the same torque as the long manipulator version. It also introduces complexity since there will be an extra degree of freedom, only so that it can be short some of the time.
The long manipulator was the best alternative considering the uncertainties with the short manipulator design. This makes designing easier since there is more room, and the manipulators are useful as long as they are strong enough to move. Now the main concern would be whether or not it will be possible to stand on the robot when the arms are mounted. The key to his is the shoulder joint.

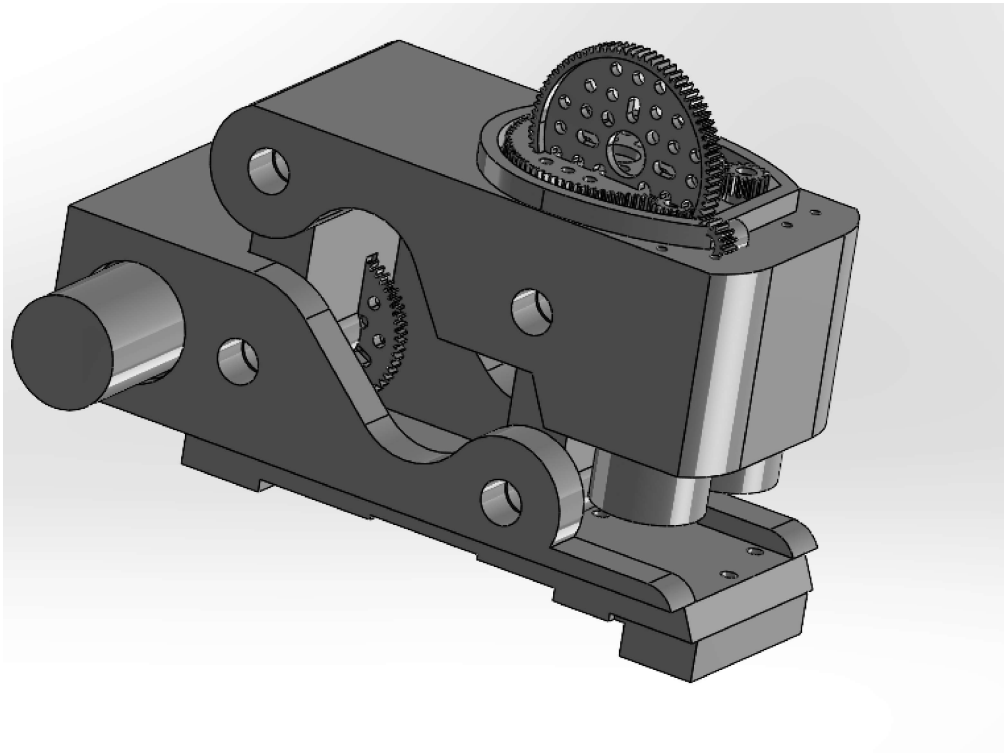
2.3 Shoulder mechanism and location

Figure 2.1: Little space to mount the manipulator

Since the robot is able to transform from SBV mode to robot, it should be able to do this with the manipulators mounted. Then the shoulder must be in a place where the driver does not stand when using the Loomo. This placement proved difficult with only 5 cm clearance between the robot and the legs of the driver, see Fig. 2.1.



(a) Collarbone retracted



(b) Collarbone mechanism pulled out

Figure 2.2: Collarbone Mechanism

Making a mechanism that withdraws the shoulder into a docked position is a valid option. There is not enough space to do a normal rotation to retract the shoulder, but it is possible to make a mechanism that moves the mounting plate for the manipulator always parallel to the robot's side. The design can be seen in Fig. 2.2. The design brings better comfort, a more original look, and a flexible solution for the space problem. The block itself in the model is 5.5 cm thick when folded. With this design, gears are the only viable option for power transmission since there is space for nothing else. The gears would have to be bought from a factory and be made of metal to with-

stand the forces. This would be expensive. In its retracted state the manipulator would be moved very far forward to make room for the two motors that will do the lifting and rotating of the said manipulator. This is bad for balance since more weight is moved forward. There are sensors in front that would be blocked by the motors, meaning they would be rendered useless as long as the manipulators are folded together. All of these things make this extraordinary solution unwanted since it also adds a lot of complexity.

Choosing gears as the power transmitting method in this project could be a good idea, but only if the gears are guaranteed to stay aligned. This usually means having a gearbox to house the gears. Designing one would take too much time, and buying one for each specific application was out of budget. The gearing, therefore, has to be part of the gearbox on each actuator respectively, and other alternatives are chosen for other gearing.

A stationary shoulder is the other possibility. If there would be a place where the axle could be fastened, the actuator could transmit power and not block the person standing on the Loomo, this could be possible. As long as the shoulder is placed in front of the ankles of the driver, it will not interfere, and the axle could be fastened with a machined bracket. Power transmission is by far the most challenging. This needs to be the most powerful joint, and it is also important that the shoulder is not poking out too far to the point where it is impractical and will look weird. Then how will the shoulder raise the manipulator? The actuator needs to be put in a place where it, along with power transmitting parts, pokes less than 5 cm out from the Loomo on both sides. There are two different possible ways to do this. Either mounting the actuator pointing into the side of the shoulder, using bevel gears to transmit power, or choosing belt drive where the actuator is placed behind the Loomo. Fitting the motor, bevel gears, a bracket to hold the axle, and a cover in only 50 mm was not possible. These two bevel gears would cost more than double the price of the motor.

To make the manipulator able to lift the desired amount, gearing down the shoulder actuator in a 2:1 configuration means that it spins at half speed, but will lift double. Since the same actuator is placed further out, at about halfway, both will have about the same capacity. The fact that the manipulator is slower at the shoulder rotation means only better balance.

Taking this into account, gearing with a belt is easier considering the spacial limitations of the Loomo, and cheaper since 3D printing the pulley wheels is an option. The small gear will however require some form of fastening that is adequate for the amount of torque. The only downside of this solution is that a tightening mechanism needs to be implemented.

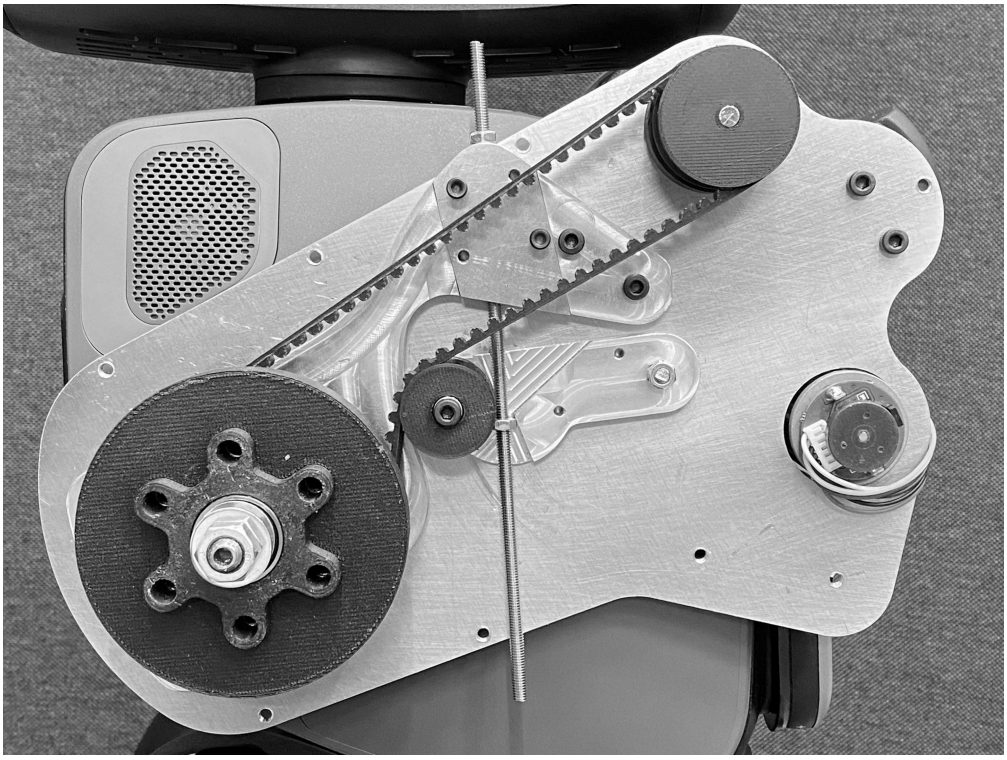


Figure 2.3: Belt pulley with tightening mechanism in upper configuration

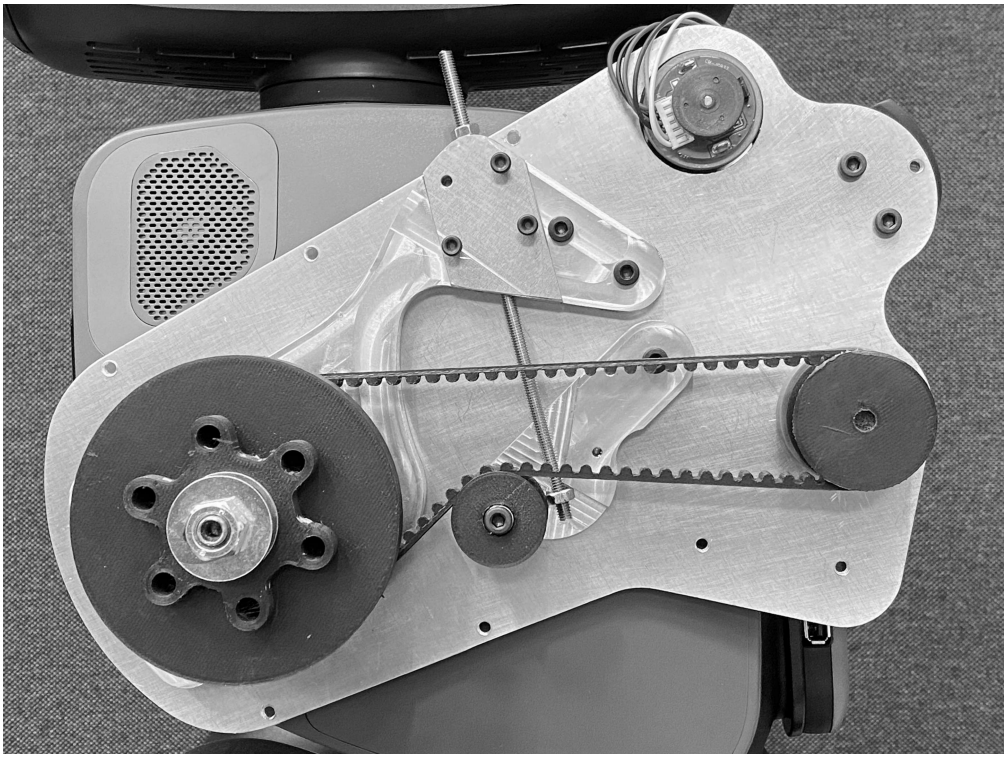


Figure 2.4: Belt pulley with tightening mechanism in lower configuration

The belt mechanism positions the actuators at the back of the Loomo. This introduces a new problem; if the sides are to be as symmetrical or equal as possible, the actuators occupy the same space from the left and right actuators. It is wanted to have the idler and tightening side on the bottom part of the belt, since in most cases, and for the highest loads, all the force from the arm will transfer along the top part of the belt. If one is placed above and the other below, the tightening would be different for each side. The only exception being the tightening mechanism is flexible enough to tighten the belt for both positions for the actuator. A solution like this means it would

be entirely up to the installer which side should have its actuator installed where, and more parts are interchangeable. Also, the distance between the actuator and the shoulder must be the same to use the same belt.

2.4 General Aesthetics and Safety

When it comes to how the manipulator will look, there are mostly two categories it could fall into. It is either designed as an industrial manipulator or as a human-like arm.

The human-like manipulator would be round, have realistic and limited joint range, and be rounded off. Even though it is hard plastic, it would be fine to hit it in the event that the knees are bent more than usual, and safer to fall over. In the event of the driver falling over or otherwise onto the manipulator, there would be no sharp edges, and therefore a lesser chance to get hurt. While interacting with the robot it is also likely to have a reduced pinching point hazard because of the circular geometry at the joints.

Industrial robots are often made to be efficient and versatile. They usually operate alone, having security measures to either keep humans away or stopping when they come too close. Having a cage to work in or motion sensors to sense when humans get too close. Because of this, range of motion, strength, and weight can be the entire focus. In this application, this does not apply. It has to be able to be around humans and operate near them. Making this type of manipulator would be easier and may be the only option to fit the necessary components in the manipulator. It could even end up lighter with the same payload potential. However, safety and comfort must be weighed heavily.

Since there is a driver that will stand on the Loomo with the manipulator mounted on it, is it important that it is safe and comfortable. It is also a priority to have a good payload potential, but this is not the main goal of the manipulator. High acceleration and speeds would likely make the Loomo struggle to balance, and affect the final product negatively if exaggerated too much. The human-like manipulator is therefore chosen, because of its safety characteristics, and therefore being a better fit for this application.

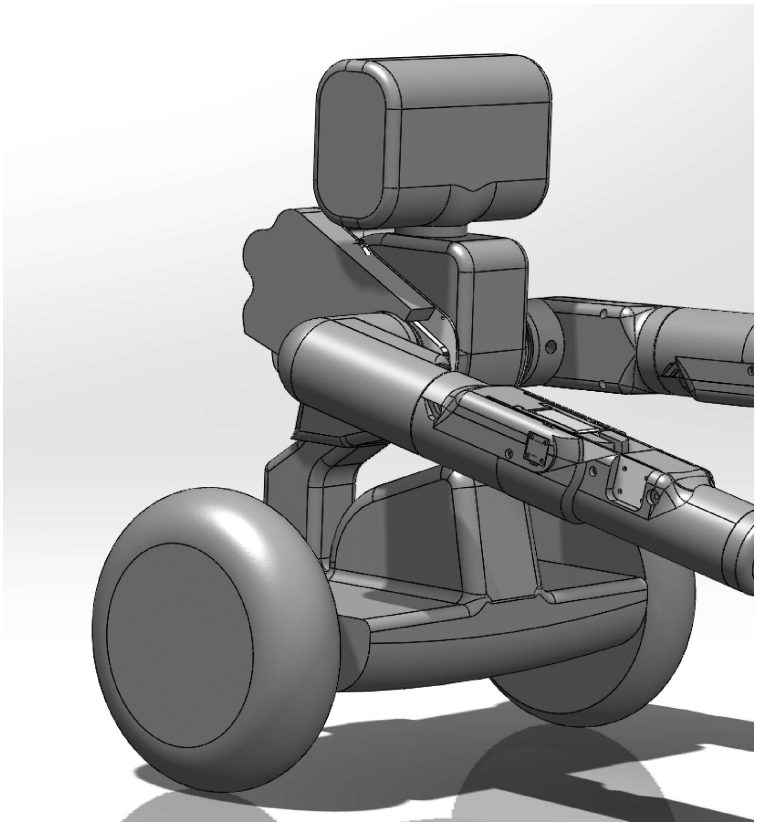


Figure 2.5: Rounded covers makes the ride comfortable and safe

2.5 Components for the manipulators

There are some components that have to be in the manipulators, these include angle sensors, IMUs, and motors. There are some components that reduce cables massively. BUS components are an example, functioning like hubs, having many cables going in and only a few cables going out. And there are the rest of the components; Arduinos, motor controllers, voltage converters, and current sensors. All of these combined take up much space.

There is much space behind the robot, the only features occupying this space are the Lifting handle and the extension bay. It is possible to move the Lifting handle around a bit, and the extension bay is seen as part of the design here. Any additional extensions need to be connected to the same system anyways. Putting most of the components here means that the manipulator is connected heavily to the robot when mounted, meaning it will be harder to remove just the manipulators. Special connectors could make this simpler. Since the manipulators will be long, some weight compensation is needed. Putting as much weight as possible on the robot's back helps.

It could be possible to fit them all into the manipulator but this requires the positioning of components and geometry of everything around it to be very precise. Components would most likely need to be made specifically for this platform, and custom PCBs would be necessary to connect it all together. It would make for a very easy manipulator to take on and off, and arguably a better product as a whole. It would also be more expensive. If the only way to ride the robot were to remove the manipulators, focusing on this being easy would be important. It is on the other hand important for the product to follow and fit the design specifications, so the old functionality should not be blocked by these new manipulators.

2.6 Degrees of freedom

The manipulator needs to be able to reach high up and fold together nicely. To do this it needs some degrees of freedom. Each of these comes with geometry, bearings, motor, sensors, and other components, so limiting the amount will make the manipulator more viable as an extension of the existing robot. Having too few means a very rigid and useless manipulator.
There are some degrees of freedom that are absolutely necessary for the basic functions of the manipulator. Firstly the shoulder joint, which makes it possible to raise the arm forward and up defining most of the workspace. The workspace with only this shoulder joint is that of a circle, meaning it can reach only points exactly the manipulator's length away parallel to the plane it was mounted on. To be able to reach points closer than this it needs another joint. It was placed midway and resembles an elbow-joint. It was placed off center to be able to fold as far as possible. These two, combined with the Loomo's ability to reposition by driving around, makes it possible to reach most points in space below a height-threshold.

In and of itself the manipulator is only two dimensional, mounted on a moving robot, so doing small adjustments means driving around to reposition. Having the ability to reach points in 3D would mean way easier and faster actions. A torsion of the upper part of the manipulator means that the workspace increases with a bent elbow joint, and when the manipulator is stretched out, the ability to rotate the end effector to hit buttons at a certain angle is possible. Adding this degree of freedom adds so many abilities, and is a good addition to the robot.

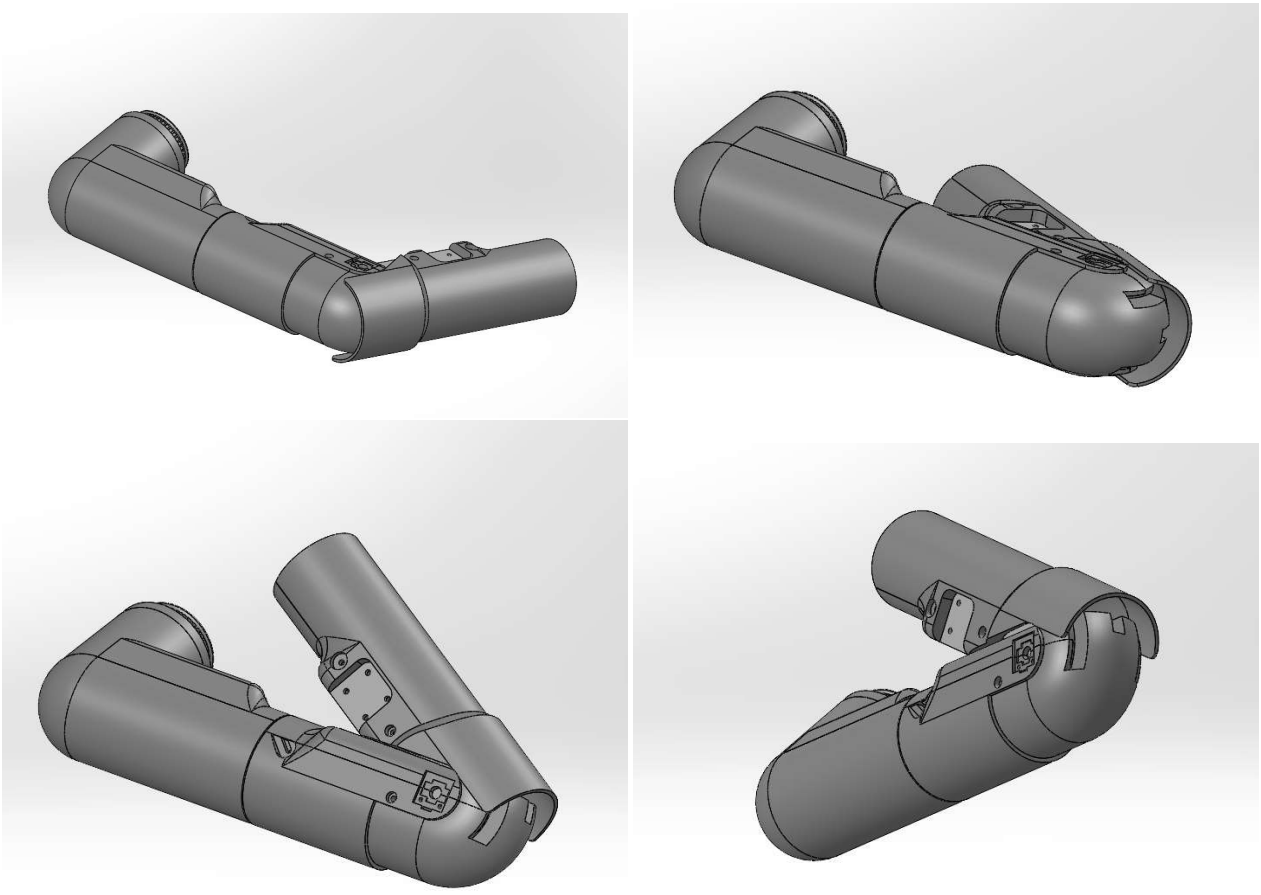
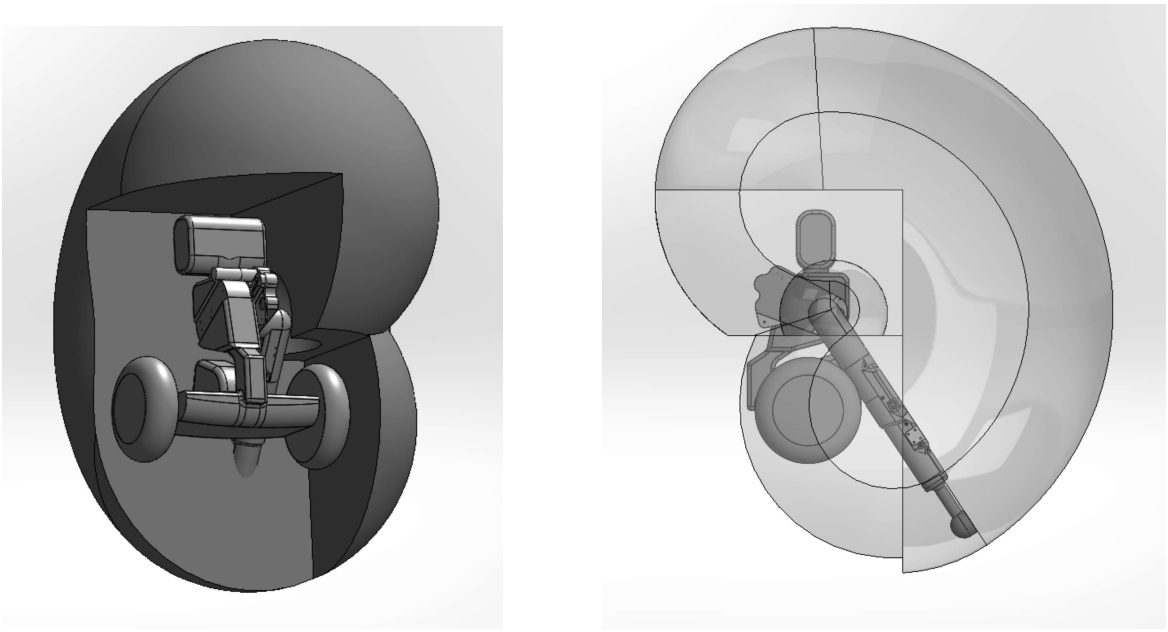


Figure 2.6: A showcase of the Degrees of Freedom in the manipulator

Lateral movement of the shoulder would make the workspace much larger. Adding mostly space beside the robot, one can question whether or not it is necessary. In the context of it being human-like, this added flexibility makes sense. However, adding it makes the shoulder joint too complex

taking into account the limited space on the robot.

Torsion of the forearm is a degree of freedom that makes manipulating objects much easier. It is necessary for any manipulator with a grabber or anything similar, but since the main focus of this manipulator is to hit buttons, it was not included. However, the manipulator itself was shortened to account for further development and the addition of an end effector where this degree of freedom is necessary. To hit buttons it is therefore necessary to install a pointer of about 25 cm.



(a) View from the back

(b) View from the side

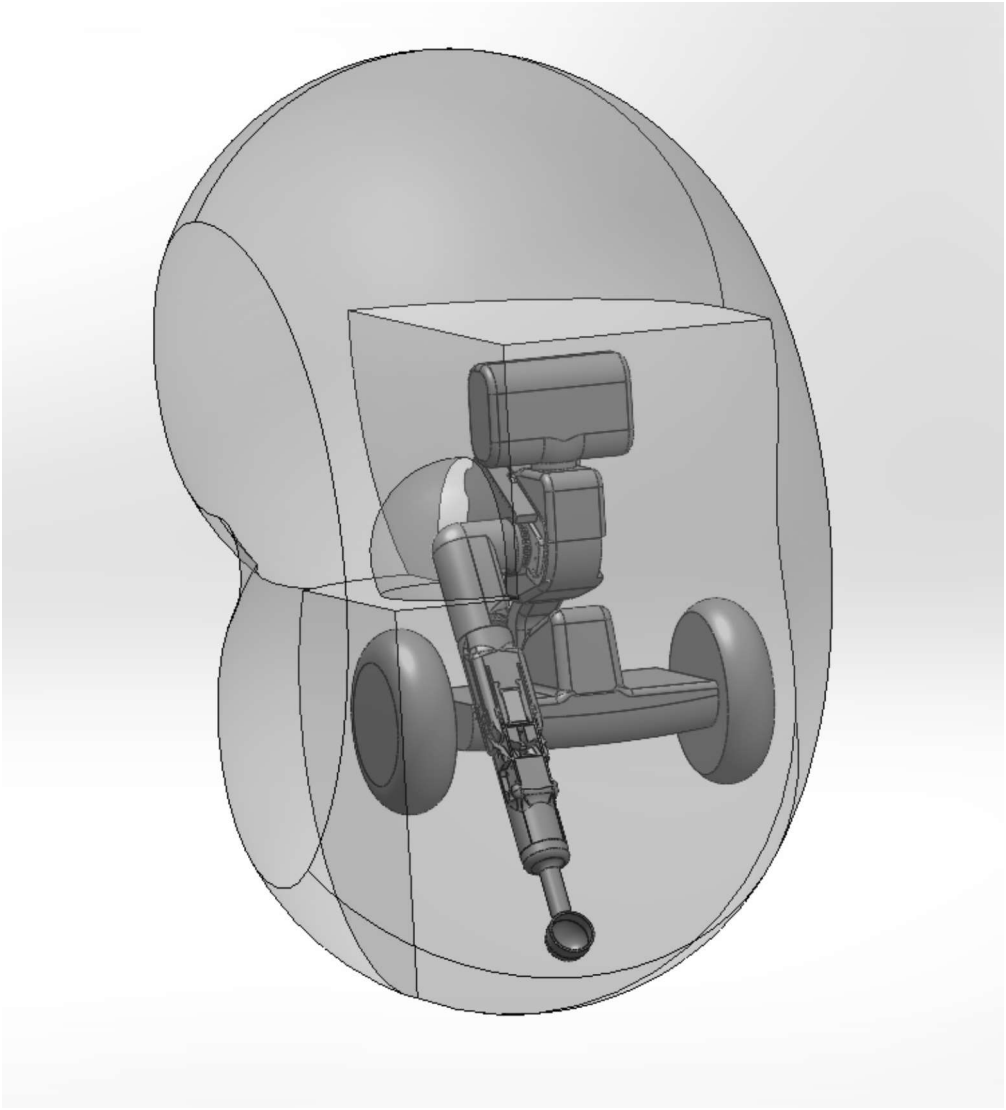


Figure 2.7: The workspace of the right manipulator

2.7 Smaller Decisions

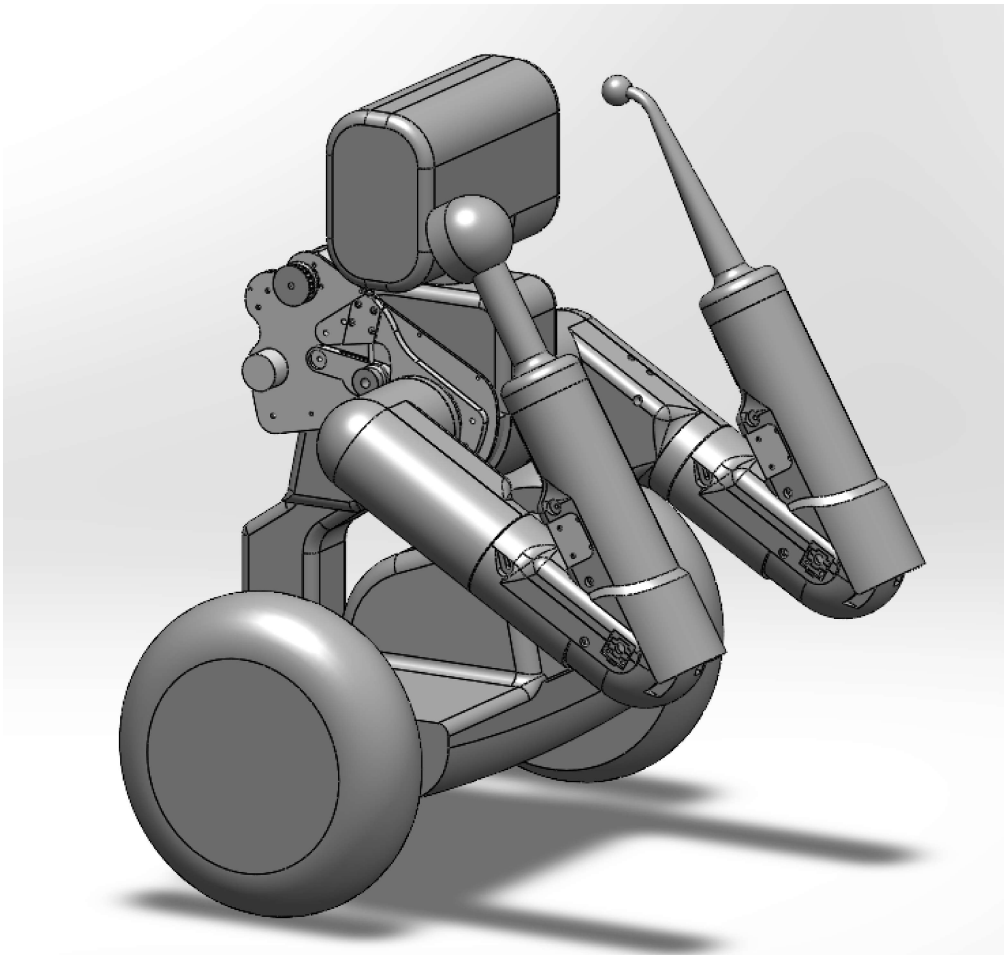


Figure 2.8: The Home Position of the robotic arms on Loomo

In the beginning, the diameter was set as 70 mm. A happy medium to have room for what was needed in the form of components, and not too big in the sense that it fits on the robot. But when the installation of sensors came into play, the magnetic sensor for the elbow required more room to be flush with the rest. This combined with it looking slender in its full length, it was decided to increase thickness to 80 mm. It is however reduced to 70 mm still right after the elbow joint.

The inevitable choice of a home position for the robot was one that was taken into consideration throughout the entire process. It had to first of all be practical, but also look natural, human-like. Also, be easy to translate to, stable, and require less strain on the motor and its gearboxes than in a normal position. Combining all of these, while not blocking the driver's feet or sticking out to the side, proved difficult, but the best solution seemed to be the position shown in Fig. 2.8

When it comes to maintenance and accessibility, being able to open the manipulator up and dismantle certain components was important. Also being able to easily pull cables through the design and install new components, equipment or end effectors with both was important. Therefore the design makes it possible to remove covers in any order. Some things like swapping motors and structural parts still require dismantling.

2.8 Final manipulator

Figure 2.9: The Final Product, with a human model[17] for scale

The manipulator is versatile because of its workspace, strength, and the fact that it is still possible to stand on the Loomo with all the gear mounted on it. This was also tested by printing out parts and mounting them on the Loomo. The manipulators are also made identical so both can be mounted on both sides.

2.9 Comparison table

Key variants have been assembled in Tab. 2.2. Based on relevant qualities they have been compared against each other. The color of the tables cell, and the text that it contains describe if the variant performed well. The scale is represented in Tab. 2.1. The variants/Degrees of Freedom that was chosen as part of the manipulators, are colored cyan.

Table 2.1: Color coding for Comparison table

Futile	Bad ——————————>	Good	Vital

Chosen In solution

Table 2.2: Major Product-Variants Comparison

___ Quality ___ Variant	**Strength**	**Balance**	**Flexibility**	**Aesthetics**
Long manipulator	Weaker since the manipulators are longer and therefore requires more torque	Harder to balance with same load and movements	Large work area makes the robot a more flexible tool	A longer manipulator will look weird on such a small robot
Short manipulator	Stronger since the manipulators are shorter	Easier to balance making the robot more stable	Less work area introduces a need to extend the manipulator to reach the buttons	Fits with proportions of the rest of the robot
___ Quality ___ Variant	**Balance**	**Stiffness**	**Flexibility**	**Aesthetics**
Collarbone mechanism	Bad balance especially when folded into resting position since all the weight lies in front of center of mass	Lower stiffness, and most likely more backlash	The extra degree of freedom makes it easy to stand on it and just fold the manipulator back when it is needed	Better comfort, may look more original if a good idle position is found
Locked shoulder	Better balance when folded, worse when in use	A stiff solution	Goes clear of the head but cannot move the manipulator out of the way	Easier to find a good idle position since the manipulator is positioned further back
___ Quality ___ Variant	**Durability**	**Space and turnover ratio**	**Maintenance, gripping teeth**	**Precision**
Gear drive	Shorter lifespan since teeth will see a lot of stress and can break with monetary big loads	More compact and can achieve high turnover ratio in the form of a gearbox	Must potentially swap out gears over time, few teeth gripping	Some backlash and good precision as long as the gears stay close and does not slip
Belt drive	Very durable solution	Takes up more space and require tightening mechanism	Low maintenance, many teeth in grip	Negligible backlash, high precision if the belt is tightened enough to not slip

Table 2.3: Continued: Major Product-Variants Comparison

___ Quality ___ Variant	**Comfort and safety**	**Ease of maintenance**	**Size and strength**	**Aesthetics**
Human-like manipulator	Comfortable to stand beside, and since most components are covered less danger for user	Harder because of hidden components, but they are more protected and might have less maintenance	Covers and human like joint structure leads to a bigger manipulators. Motors and their brackets will be smaller and weaker	Will look very good since the round and human look is part of the robots image already
Industrial	If manipulators are installed there is less comfort and more risk of injury if user falls forward	Easier to balance making the robot more stable	Smaller or able to fit more components, can build the manipulator to be as strong as possible	Will not look as good, aesthetics are not in focus in industrial applications
___ Quality ___ Variant	**Cable amount**	**Weight and size**	**Cost and development**	**Ease of maintenance**
All components in the manipulators	Going in to the manipulator there will be very few cables resulting in few connectors	Manipulator is heavier and bigger making it able to lift less and is harder to balance	The development and component costs will increase with tighter design specifications	Easy in the back, harder in the manipulators. More cables going through joints
Most in the "backpack"	Many cables into the manipulators, but less chaos in the manipulators	Lighter and thinner manipulator more component clutter in backpack	Easier, cheaper and faster development	Easier in the manipulators, harder in the back
___ DoF ___ Choice	**Shoulder and elbow**	**Torsion upper manipulator**	Lateral movement shoulder	Torsion forearm
Adding the degree of freedom	Necessary for the basic functions of the manipulators	The robot will not need to drive around to position correctly and make small adjustments to sideways positioning	Makes workspace much larger	Mostly affects e.e. movement and makes it possible to rotate it
Not adding the degree of freedom		It is possible to drive around to move sideways	Not strictly speaking necessary	This degree of freedom is not necessary to push buttons

Chapter 3

Theory

3.1 Mechanical formulas

Static analysis studies the behavior of structures and mechanisms under static, or stationary, conditions. In such cases, the sum of forces and moments acting on the structure must be equal to zero, ensuring that the structure is in equilibrium. The equations for such a case in 2 dimensions are shown in Eq. (3.1).

$$\begin{aligned} \sum F_x = 0 \rightarrow Fx_1 - (Ax_1 + Ax_2 + ... + Ax_n) = 0 \\ \sum F_y = 0 \rightarrow Fy_1 - (Ay_1 + Ay_2 + ... + Ay_n) = 0 \\ \sum M = 0 \rightarrow M_1 - (Ay_1 \cdot x_1 + Ay_2 \cdot x_2 + ... + Ay_n \cdot x_n) \cdot g = 0 \end{aligned} \tag{3.1}$$

By dividing the beam up in pieces when loads are changin, the axial force, shear force and bending moment diagram (ASB) can be found. Analyzing a beam's forces and moment will result in the location of the highest stress concentration. Using these equation (3.2, 3.3 and 3.4). This will give the stress in a specific crosssection of a beam.

$$\sigma_x = \pm \frac{32|M|d_{\text{out}}}{\pi \left(d_{\text{out}}^4 - d_{\text{in}}^4\right)} \tag{3.2}$$

$$\tau_{\max} = \frac{4}{3}\frac{|V|}{A}\left(\frac{r_2^2 + r_1 r_2 + r_1^2}{r_2^2 + r_1^2}\right) \tag{3.3}$$

$$\sigma' = \sqrt{\sigma^2 + 3\tau^2} \tag{3.4}$$

Dynamic analysis studies the behavior of a structure in motion. The sum of forces and moment are shown in Eq. (3.5). Mass moment of inertia as shown in Eq. (3.6) describes a bodies resistance in change to rotation. The kinematic equation shown in Eq. (3.7) describes the motion without reference to the forces and torques applied that cause motion.

$$\begin{aligned} \sum F_x = m \cdot a_{gy} \\ \sum F_y = m \cdot a_{gy} \\ \sum M_G = I \cdot \alpha \end{aligned} \tag{3.5}$$

$$\sum_i I = m_i r_i^2 \tag{3.6}$$

$$\vec{A}_A = \vec{A}_O + \alpha_1 \breve{\vec{R}}_{A/O} - \omega_1^2 \vec{R}_{A/O} \tag{3.7}$$

The gear ratio n is given from Eq. (3.8) which determines the relationship between the input and output speed/torque. The gear ratio is the ratio of the number of teeth or the diameter of the larger pulley and the smaller pulley. The gear ratio is used to calculate the output torque on the

large pulley with Eq. (3.9). This torque is used to find the max payload for the manipulator at the maximum length of the manipulator l_M, Eq. (3.10).

For any m'th derivative of the angle θ, the relationship remains the same, i.e. angular- position, speed, acceleration, jerk, and so on, Eq. (3.11).

$$n = \frac{D_L}{D_S} \tag{3.8}$$

$$M_L = M_S \cdot n \tag{3.9}$$

$$F_P = \frac{M_L}{l_m} \tag{3.10}$$

$$\theta_L^{(m)} = \frac{\theta_S^{(m)}}{n} \tag{3.11}$$

3.2 DH-table

The Denavit-Hartenberg (DH) table is used to relate the different links and their local rotation as coordinate systems or reference frames. These are changed depending on how the robot's links connect to each other and how long they are. The only two axis included in the table is the z and x axis. This is to simplify and reduce the size, seeing as it is only required to have two axis to translate anywhere with any rotation.

Table 3.1: Example of a DH table with one row

RotZ	TransZ	TransX	RotX
α	L_z	L_x	β

The DH table is used to find the matrix for forward kinematics, which is described below.

3.3 Forward kinematics

Forward kinematics is the process of determining the position and orientation of a robotic manipulator's end effector based on the given joint angles, utilizing the Denavit-Hartenberg (DH) table for calculations. This concept is essential in robotics, as it allows us to determine the end effector's location using a set of joint angles, which is valuable for tasks such as path planning and obstacle avoidance. For example a robotic arm with three joints. By using the DH table, the end effector's position and orientation based on the joint angles can be calculated.

$$\left[{}^{i-1}T_i\right] = \begin{bmatrix} \left[{}^{i-1}R_i\right] & \left\{{}^{i-1}P_i\right\} \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix}, \quad \text{4 x 4 matrix} \tag{3.12}$$

3.4 Inverse kinematics

Inverse kinematics is the process of calculating the joint angles required to achieve a desired position and orientation of the end effector of a robotic manipulator. The inverse kinematics problem is important because it gives the possibility to control the motion of the robot by specifying the position and orientation of the end effector. This is useful for tasks such as robot control and programming. For example, to move the end effector of a manipulator to a specific location in space, If the orientation also is important, pipers solution can be used to calculate the joint angles required to achieve that position and orientation, but then the robot must have many degrees of freedom.

3.5 Jacobian

The Jacobian Matrix is a matrix consisting of all the first-degree derivatives of a vector-valued function. In robotics, the Jacobian is a matrix that relates the velocities of the robot joints to the velocities of the end-effector or the tool attached to the robot manipulator. It tells how fast the velocity of the end effector changes with respect to the joint velocity. The Jacobian depends on the robot's geometry, kinematics, and configuration. The Jacobian matrix relates the linear and angular velocities of the end-effector of a robot manipulator to the joint velocities of the robot, making it useful for designing feedback controllers and trajectory planners [9].

3.6 Motion control

Eq. (3.13) is the academic formula for a PID controller. To solve the equation in a microcontroller which works sequentially the equation have to be computed numerically as shown in Eq. (3.18). The error of the system is given with Eq. 3.14 wich is the difference between the setpoint and process value. Together with Eq. 3.15 and Eq. 3.16 which sums up the error each iteration can be calculated with. This is used to calculate the amount of control signal the integral part gives to the system. For the derivative part the rate of change of the error have to be calculated with Eq. 3.17. The gain values for the proportional (K_p), integral (K_i) and derivative (K_d) can be found with trial and error or using some tuning method such as Ziegler-Nichols.

$$PID = K_p \cdot e + K_i \int_{t=0}^{t=\infty} e\,\mathrm{dt} \; + K_d \cdot \frac{\mathrm{de}}{\mathrm{dt}} \tag{3.13}$$

$$e = \text{reference} - \text{sensorvalue} \tag{3.14}$$

$$\mathrm{dt} = \text{time} - \text{lastTime} \tag{3.15}$$

$$e_{int} = e_{int} + e \cdot \mathrm{dt} \tag{3.16}$$

$$\dot{e} = \frac{e - e_{prev}}{\mathrm{dt}} \tag{3.17}$$

$$u = K_p \cdot e + K_i \cdot e_{int} + K_d \cdot \dot{e} \tag{3.18}$$

3.7 Complementary filter

A complementary filter is a sensor fusion technique that consists of a low pass and high pass filter, which combines data from multiple sources (e.g accelerator and gyroscope) to produce a more accurate and reliable output. Where each of them has their own strength and weakness.

A system that measures the orientation of an object such as an MPU with an accelerometer and a gyroscope. The accelerometer is good at measuring the static orientation since gravity is always pointing in the same direction, but it is sensitive to external noises. The gyroscope is good at measuring dynamic changes in orientation but drifts over time due to small errors accumulating [2].

The complementary filter combines sensor values from both sensors by using a weighted average. It takes advantage of that the accelerometer is more reliable at low frequencies and the gyroscope is more reliable at high frequencies. By weighing the gyroscope higher than the accelerometer as shown in Eq. (3.19). Where $(1 - \alpha)$ is the high pass constant and is α the low pass constant. The previously filtered angle from the gyroscope is given from $\theta_G(n-1)$, angular velocity is $\omega_G(n)$ from the gyroscope, and dt is the sampling time. Lastly, $\theta_A(n)$ is the angle from the accelerometer. The result of this will give a more accurate and stable estimate of the object's orientation than the sensors could provide individually [2].

$$\theta_G(n) = (1 - \alpha)\left[\theta_G(n-1) + \omega_G(n) \cdot dt\right] + \alpha\theta_A(n) \tag{3.19}$$

3.8 Battery calculations

To make a battery pack for meeting a specific design requirement some calculations have to be done, these are shown in Eq. (3.20) and (3.21). To reach a specific capacity in Ah or Wh the cells are connected in parallel. Also to increase the voltage of the pack multiple cells are connected in series. When a battery pack is finalized it is in a configuration as e.g 6S4P, which means 6 cells in series and 4 cells in parallel this results in a 24voltage and 12Ah battery pack.

$$\text{Number of cells in parallel} = \frac{\text{desired capacity}}{\text{capacity of one cell}} \tag{3.20}$$

$$\text{Number of cells in series} = \frac{\text{desired voltage}}{\text{voltage of one cell}} \tag{3.21}$$

Chapter 4

Mechanical design

When building a mechatronic system, such as a robot manipulator the mechanical design criteria are the first ones to consider. Most applications have some physical requirements or qualities that the system is designed around. The design criteria are based on ensuring the ease of manufacturing, assembling, maintaining the manipulator, and also ensuring its functionality. In this case, the design specifications are specified in Section 2.1.

Choosing the right type of bearing is important in ensuring the system's functionality and longevity. The bearings must be able to hold both axial and radial loads while maintaining a smooth operation.

Selecting the appropriate gear ratio is crucial in ensuring the proper output torque and speed of the system. The gear ratio is to be chosen carefully to ensure that the output torque is sufficient for the application, without sacrificing too much speed on the output shaft.

Choosing the right type of gear drive system is also essential in ensuring the system's efficiency and functionality. There are various gear drive systems available, such as spur gears, helical gears, and bevel gears. The choice of gear drive system depends on the specific requirements of the mechanical system being designed.

When designing a part for CNC milling, it is essential to consider factors that can affect the machining process. One aspect is the complexity of the geometry. Not all designs that work on paper or that are 3D printed may be suitable for CNC milling, especially when using a 3-axis machine with limited degrees of freedom. It is important to design parts with accessible geometries that can be machined from a single orientation, as undercuts or hidden features may not be achievable in a single setup.

It is also important to specify accurate dimensions and tolerances for ensuring the correct fit and function of the machined part. This includes being precise with hole sizes and specifications, whether they are threaded or non-threaded holes.

Hand calculations are crucial in ensuring the structural integrity of a mechanical system. The calculations help to determine the maximum payload that the system can handle without deforming under load. They also help in choosing the appropriate motor with sufficient torque to drive the manipulator and payload.

To verify the hand calculations and get a more detailed view of the entirety of the manipulator, there will be done Finite Element Analysis in the next chapter, and a divergence study to strengthen the reliability of the results.

All the mechanical drawings for the machined parts is shown in appendix E.

4.1 Bearing

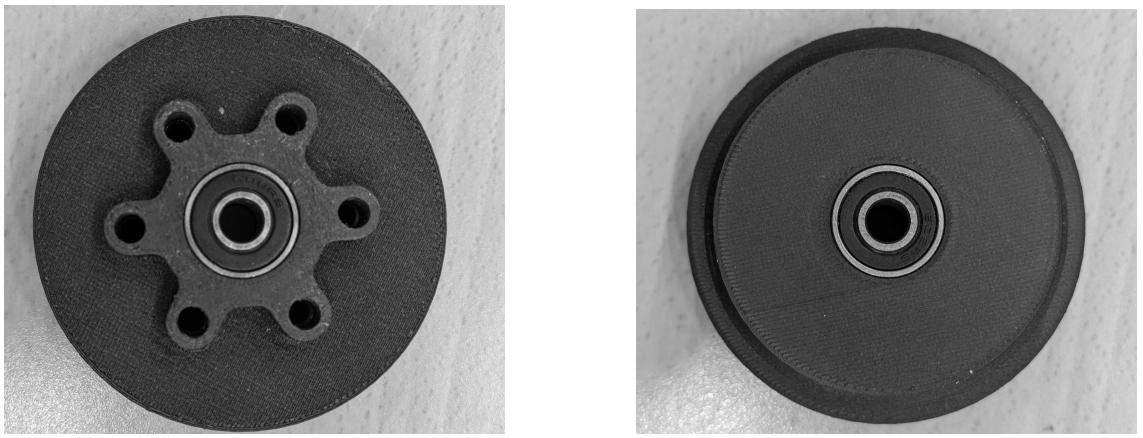
A bearing is a component that supports and allows rotation or movement of one part relative to another with minimal friction and wear. Bearings work by creating smooth, low friction between two moving parts. This is achieved by using a rolling or sliding mechanism, with a lubricant present between the two parts to reduce friction and wear.

In this project different kinds of bearings were used such as;

- Single row ball bearing, which is a specific type of deep groove ball bearing that consists of a single row of balls held in place by a retainer. Single-row deep ball bearings can primarily handle radial but also a small amount of axial loads. They have a deep groove design that helps distribute the load evenly across the balls and races [20].

- Roller bearings, a type of rolling element bearing that uses cylindrical rollers that rotates between two raceways. The inner race is mounted on the rotating shaft, while the outer race is mounted in the housing that supports the shaft. The rollers are held in place by a retainer, which maintains their position and prevents them from coming into contact with each other. Needle roller bearings have a more compact design compared to ball bearings, which makes them useful where space is a huge consideration [1].

- Thrust needle roller bearings which use cylindrical rollers arranged in a retainer. This type of bearing is designed to handle high axial loads and both high- and low-speed applications. The raceways have a low profile making them useful for application where there is little space to mount it. The disadvantage of this bearing is that it can not handle large radial loads [14].



(a) Bearing placement for large pulley in front (b) Bearing placement for large pulley in back

Figure 4.1: Bearing placement for large pulley

It was used a single row ball bearing to handle the radial loads on the large pulley, where there is mounted one on each side of the pulley to ensure a smooth rotation with little friction as shown in figure 4.1a and 4.1b. This bearing is also used for the connection between the shoulder and the torsion of the elbow to handle the radial loads as shown in figure 4.2.

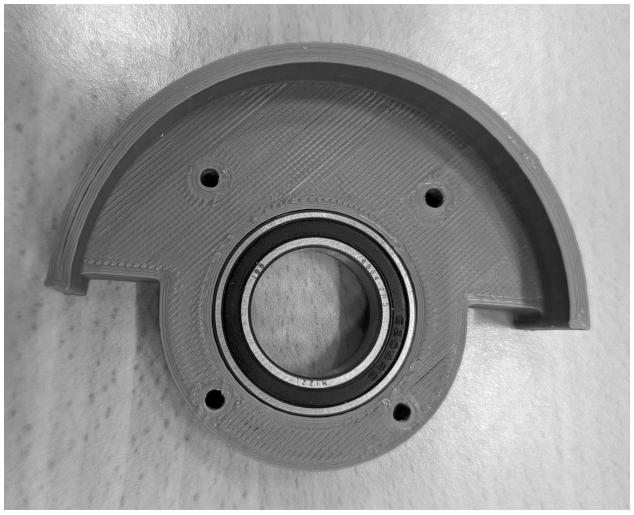
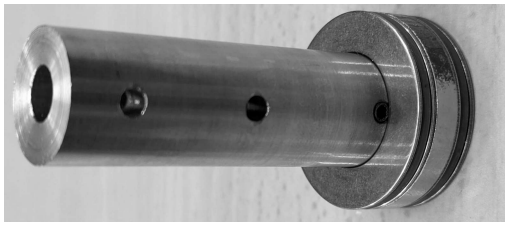
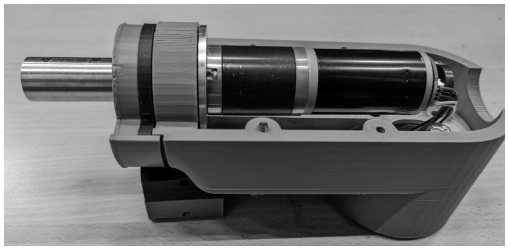


Figure 4.2: Bearing placement of the axial rotation for the upperarm

The joint that experiences the most axial load which is the torsion of the elbow, especially when the manipulator is pointing upward or downward. Therefore it is used two thrust bearings one on each side of the axle and the single row ball bearing as shown in figure 4.3a. This will result by combining both ball and thrust bearing a smooth rotation and high resistance against radial and axial loads. The complete assembly of the bearings for the torsion is shown in figure 4.3b.



(a) Bearing placement for torsion axle



(b) Complete torsion axle bearing setup

Figure 4.3: Bearing placement for Shoulder link

When the manipulator was tested with a large load there was discovered that the gearbox inside the motor that controls the small pulley made grinding noises, signalizing excessive wear. It was therefore decided to use an extra ball bearing to support the axle on the motor to relieve the gearbox as shown in Fig. 4.4. This results in a longer lifespan of the motor and most likely a tighter belt meaning a stiffer system.

Figure 4.4: Bearing for supporting motor axle

To add extra support and smoother rotation in the idler the roller bearing shown in figure 4.5 was added. This is because the idler is relatively small, so this was the only option. Roller bearings have a more compact design than ball bearings, especially when they can roll directly on the bolt or rod.

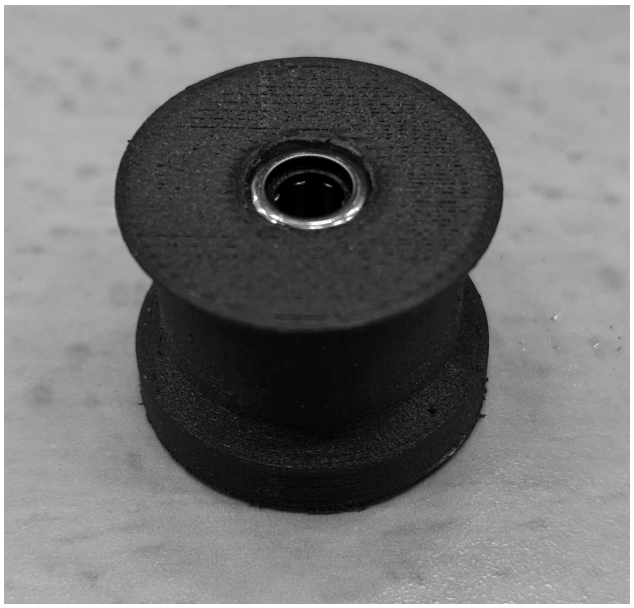


Figure 4.5: Bearing for idler

4.2 Belt drive

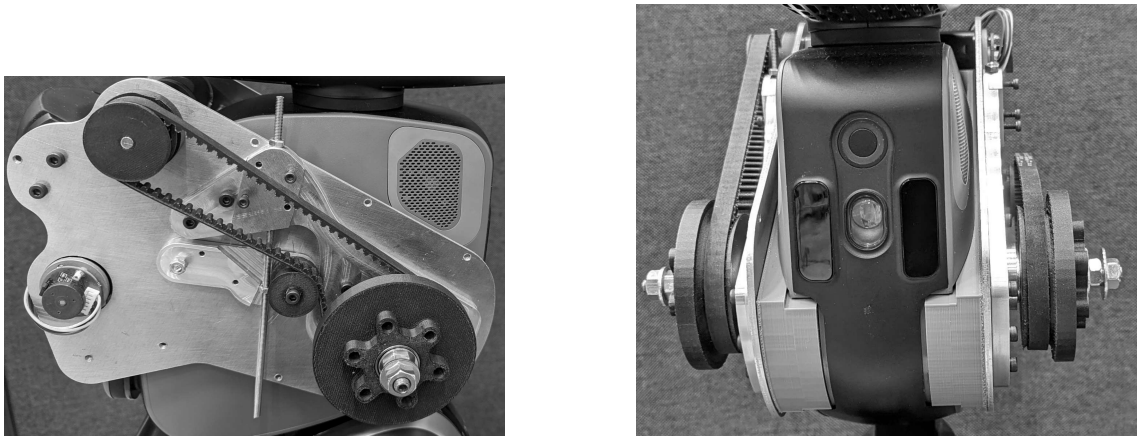
A belt drive is a mechanical system used to transmit power or motion between two or more rotating shafts, typically utilizing pulleys and a continuous belt. The purpose of a belt drive is to transfer rotational force from one shaft to another while allowing for variable speed and torque. Timing belts are characterized by teeth on the inside that fits with corresponding grooves on the pulleys.

This ensures smooth and efficient power transfer to maintain a constant speed ratio between the two pulleys. This makes the belt drive system ideal for applications requiring precise timing and speed control, such as controlling a robot manipulator.

Table 4.1: Pulley size and the resulted gear ratio

Large pulley $[D_L]$	65.52mm
Small Pulley $[D_S]$	30.69mm
Gear ratio	2.04

Seen in table 4.1 is the chosen size of the pulleys and the resulting gear ratio from the belt drive system, calculated using eq. 3.8. This means that the smaller pulley rotates approximately two times the speed of the larger pulley but with only half of the torque. The length of the belt is connected so that there was a specific distance where the motor with the small pulley could be mounted. There is added a tightening mechanism with an idler as shown in figure 4.6a. Since there was a specific requirement for still being able to stand on the Loomo, the width of the belt had to be taken into account as shown in figure 4.6b. The distance in the belt drive system on both sides is equal, and using as little space as possible resulted in choosing a timing belt with a length of 475mm and 9mm width.



(a) Belt drive

(b) Space for the belt drive

Figure 4.6: Belt drive system

Adjusting the size of the large pulley makes it possible to change the gear ratio to achieve a faster or slower manipulator depending on the application of the manipulator. For instance, a higher gear ratio can provide more torque, making it suitable for heavy loads, while a lower gear ratio can offer higher speeds, suitable for rapid acceleration or high-speed applications.

4.3 Free body diagram

The free body diagram (FBD) is used for calculating which motor to be chosen. When the manipulator is extended to its longest form, the moment is the highest. The movement then behaves as a 2D system and can be calculated as one, Eq. (3.1). To find the requirements for the motors, there were done calculations by assuming that center of mass was in the middle of the manipulator. Also, the weight of the end effector and payload was placed at the end, this is shown in figure 4.7. The assumed weight of the different parts can be seen in table 4.2.

Table 4.2: Assumed weight for the manipulator and end effector

3D-print	0.4 kg
Motor	0.5 kg
Aluminum	0.8 kg
e.e + Payload	0.5 kg + L

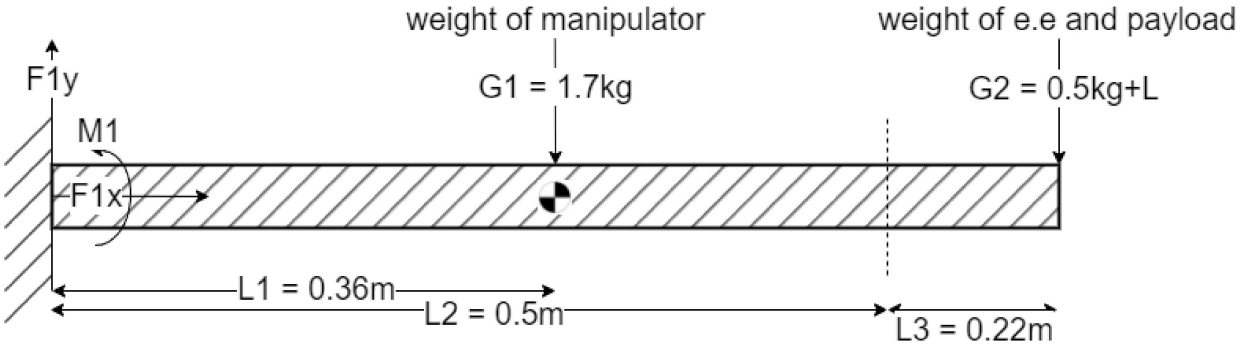


Figure 4.7: ASSUMED static properties of the manipulator

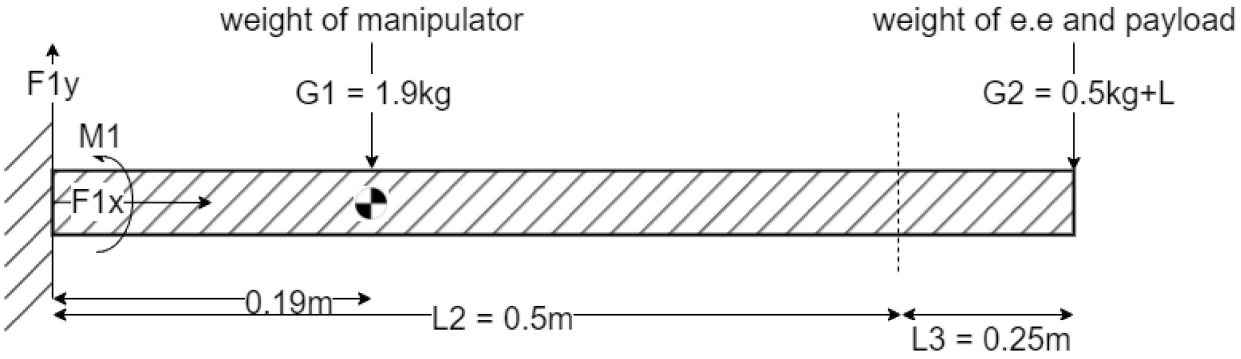


Figure 4.8: ACTUAL static properties of the entire manipulator

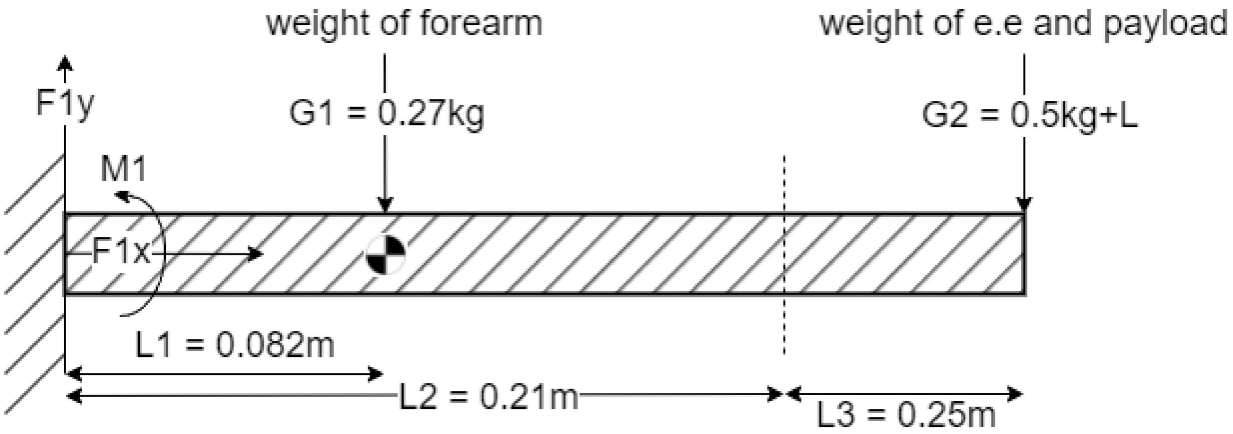


Figure 4.9: Actual static properties of outer half of manipulator

By using a trial and error method using different motors' rated torque to get a relatively large payload without sacrificing too much speed. The different motor types that come from the calculations can be seen in table 4.3. Calculations were done using equation 3.9 for the motor controlling shoulder rotation that gave the rated and stall torque on the load side. The results of the calculation from the chosen motors can be seen in table 4.4 and 4.5.

Table 4.3: The DC motors in the manipulator

Motor	Voltage	Torque	RPM	Gear ratio	Qty
31mm DC planetary gear	24V	85kg.cm	10	1:596	4
Dual shaft DC worm gear	24V	70kg.cm	9	1:500	2

Table 4.4: Shoulder motor rated and stall torque

Rated Motor torque	8.34 Nm
Stall Motor torque	27.66 Nm
Output rated torque	17 Nm
Output stall torque	56.44 Nm

Table 4.5: Elbow motor rated and stall torque

Rated Motor torque	6.87 Nm
Stall Motor torque	6.87 Nm

To get the payload, equation 3.1 and 3.10 was used since the motor moments are known. The final results from static analysis for both motors rated and stall condition are shown in table 4.6, 4.7 and 4.8. These values give the rated and absolute maximum payload the motors can handle without breaking the gearboxes mounted on them.

Table 4.6: Shoulder to e.e forces from FBD

Force	Assumed	Actual
F_x	0 N	0 N
F_y	32.1 N	37.28 N
Payload	1.05 kgf	1.4 kgf

Table 4.7: Elbow to e.e forces from FBD

Force	Actual
Fx	0
Fy	18.84 N
Payload	1.15 kgf

Table 4.8: Shoulder to e.e stall from FBD

Force	Assumed	Actual
F_x	0 N	0 N
F_y	86,82 N	90.34 N
Payload	6.63 kgf	7.00 kgf

For the finished prototype, the manipulator was assembled and the final weight was measured. The rotation point was found which resulted in where the center of mass is located. The FBD for the actual manipulator can be found in figure 4.8 and 4.9.

A static analysis for the elbow link was done and is shown in figure 4.9. This is to get the payload for the elbow motor which will give the maximum payload the entire manipulator can handle. By comparing the result of the assumed vs actual weight of the manipulator the motor controlling the shoulder rotation can have a larger rated and stall payload as seen in table 4.6 and 4.8. But since the elbow motor has a limit of a smaller payload, the maximum payload at the end effector will be as shown in 4.5.

4.4 Axial force, shear force, and bending (ASB) diagram

The ASB diagram gives a representation of the position where a structure experiences the most stress. By dividing the beam into regions before a change in loading. This can be seen in figure 4.10 where the beam will be cut into two regions. Firstly all the reaction forces and moments are found through statics equilibrium and the results are shown in table 4.9.

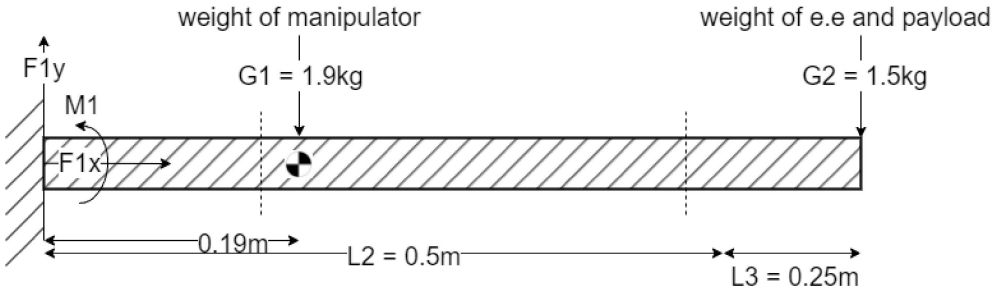


Figure 4.10: Axial force, shear force and bending moment diagram

Table 4.9: Reaction forces and moments ASB diagram

F1	0 N
F2	33.35 N
M1	14.6 Nm

Dividing the beam into regions is shown in figure 4.11 and 4.12. Now the reaction forces and moments are calculated with rotation around points A and B as shown in the figures. Table 4.10 shows the axial, shear, and bending as functions of "x".

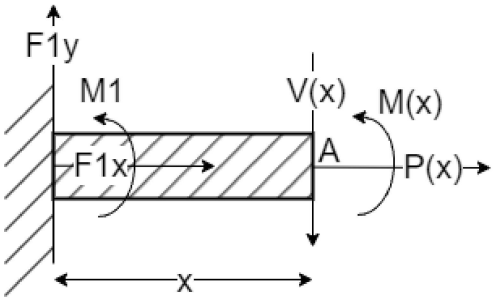


Figure 4.11: First cut on ASB diagram

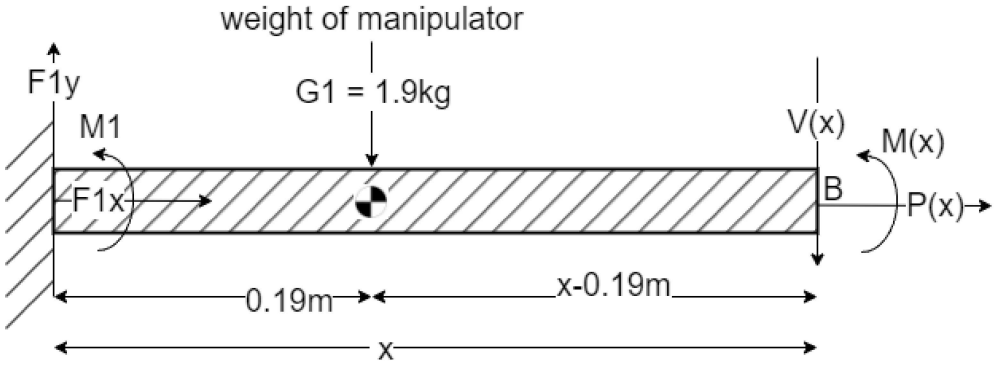


Figure 4.12: Second cut on ASB

Table 4.10: Regions from ASB diagram

	Region 1: $0 < x < 0.19$	Region 2: $0.19 < x < 0.75$
P(x)	0	0
V(x)	33.35	14.71
M(x)	33.35x - 14.6	33.35x - 18.64(x-0.19)-14.6

The results from table 4.10 are plotted in MATLAB to visualize the axial forces, shear forces, and bending moment in the beam. This is shown in figure 4.13. This graph represents the forces and moments as a function of the length of the manipulator.

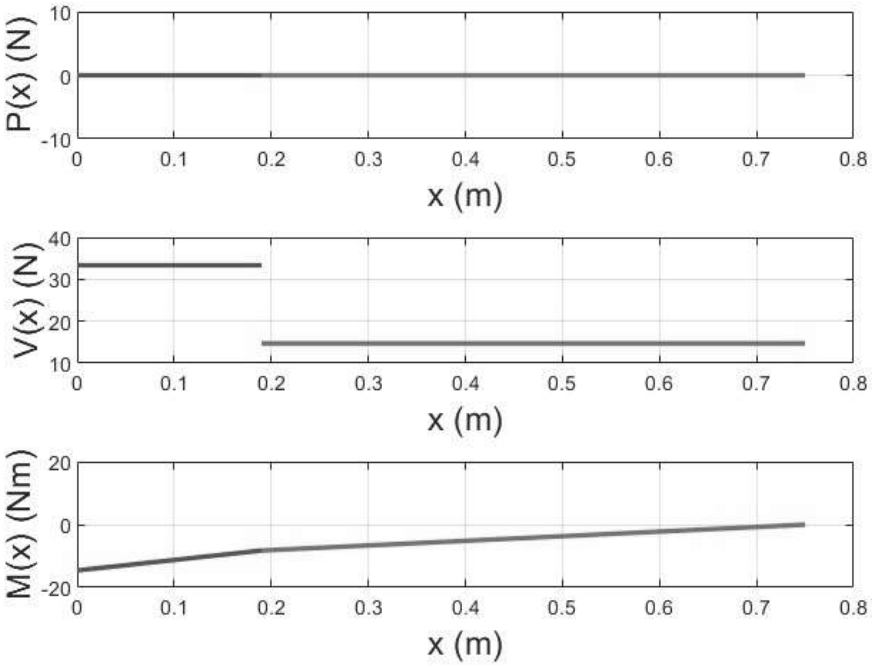


Figure 4.13: Axial force, shear force and bending moment graph

Table 4.11: von Mises Stress

Worst case	x = 0.28m
M(0.28)	-6.94 Nm
σ_{bend}	9.07 MPa
τ_{shear}	0.1 MPa
von Mises Stress	9.07 Mpa

The point where the manipulator's structure will experience the most stress is where the structure change geometrically. This is located at the torsion axle, this is due to the fact that it experiences the most stress from the load. Since it is a structural part that connects the shoulder link to the elbow and all the stress must go through this part. The result is shown in table 4.11 for the von Mises stress at this cross-section.

Chapter 5

Finite Element Analysis

FEM, or Finite Element Method, is a mathematical technique used to simulate and solve deformation and stress in a complex structure by dividing geometry into a system of elements. This system can then be solved to find the behavior of the structure under various conditions, such as force, temperature, or pressure changes. The FEM is then analyzed to find stress in the body and in this case, test the geometry of the manipulator. Note that this is a static analysis, and will therefore not take into account the extra stress from moving the manipulator. The gravitational forces are placed as point forces at the centers of mass to simplify the amount of measuring seeing as there are many parts.

The result from FEA makes it easier to determine where there are stress concentrations on the structure and where failures could occur. With the visualization of the structure it is easy to spot weak spots in the design, and with the new data, redesign parts to make it stronger against deformation.

5.1 System description

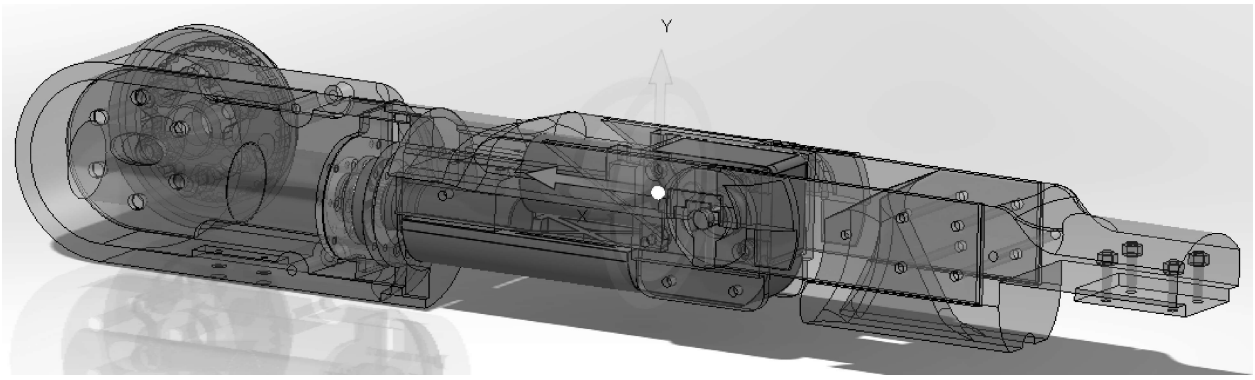


Figure 5.1: System for analysis, focused components in blue

Running the FEM for the manipulator meant doing so on a complex assembly with all the parts of the manipulator, not including bolts and screws. To simplify the analysis FEM was therefore run with bonded surface-surface component interaction. To get a holistic picture of whether or not the model actually will support the weight, bolts, and screws would have to be included. This means that stress concentration around holes and in bolts is not calculated. Also, because of the bonded component interaction, bearings and surrounding supporting geometry will have inaccurate stress and deformation profiles.

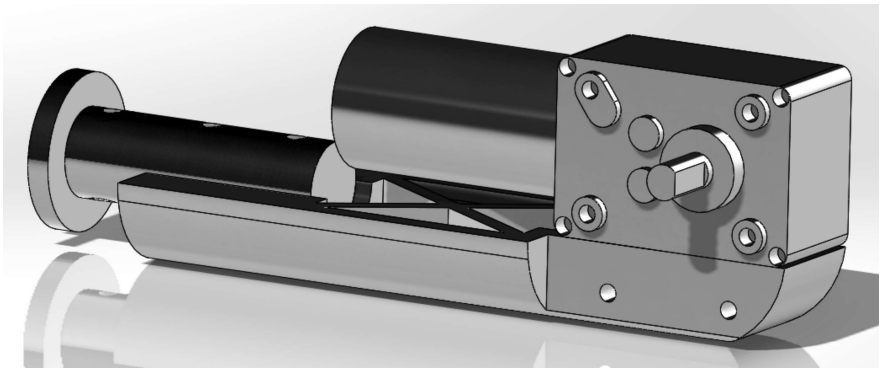


Figure 5.2: These are the parts in focus

The main parts that are in focus are the axle in the torsion joint, the worm-gear motor, and the bracket that is in between. This is because they have axles or other geometry with a low profile. These parts are therefore likely to experience the most stress, most coming from the bending moment. The meshing of this part will therefore vary more throughout the analysis, and generally be finer than the rest of the model.

5.2 Approximation tool

The FEM analysis is here used as a tool to first and foremost evaluate geometry by finding high-stress locations. Since many parts are 3D printed, the deformation and strength of the model is not necessarily an accurate representation of the real manipulator. There are however parts in metal and aluminum that are realistic. These are the parts that also carry most of the forces. These are made from aluminum, kevlar-reinforced onyx material from markforged, and S355J2 steel. Datasheets are in the appendix. A. Since the manipulator has varying geometry, the analysis will be run for two different configurations where the manipulator experiences max load. One of these being with the inside of the elbow joint inward to the side, and one upwards.

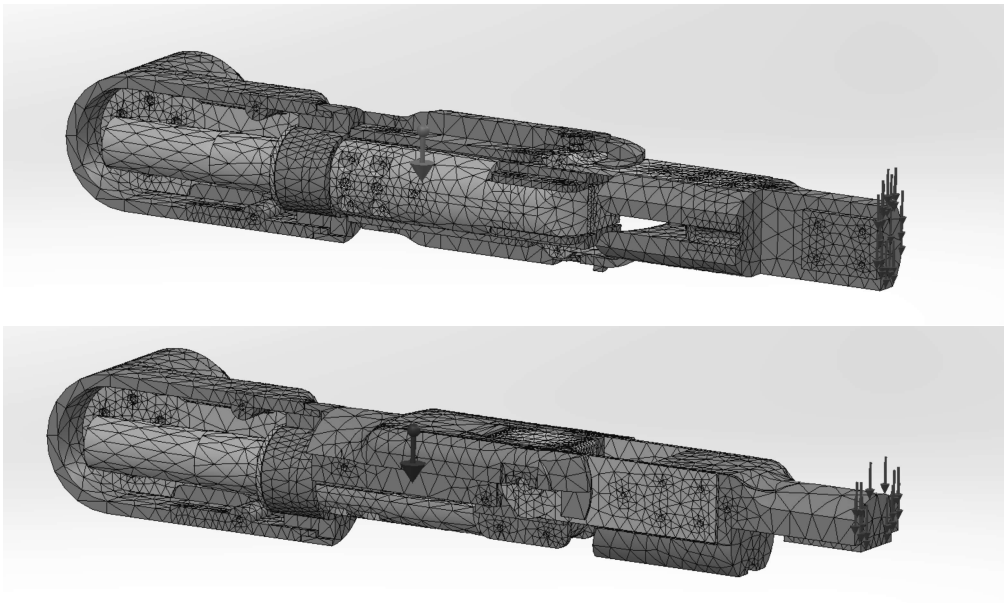


Figure 5.3: Manipulator mesh in both configurations

In Fig. 5.3 both configurations for the analysis are shown with mesh. The mesh consists of second-order elements, to accurately map the 3D objects and curve around the advanced geometry.

5.3 Design loads

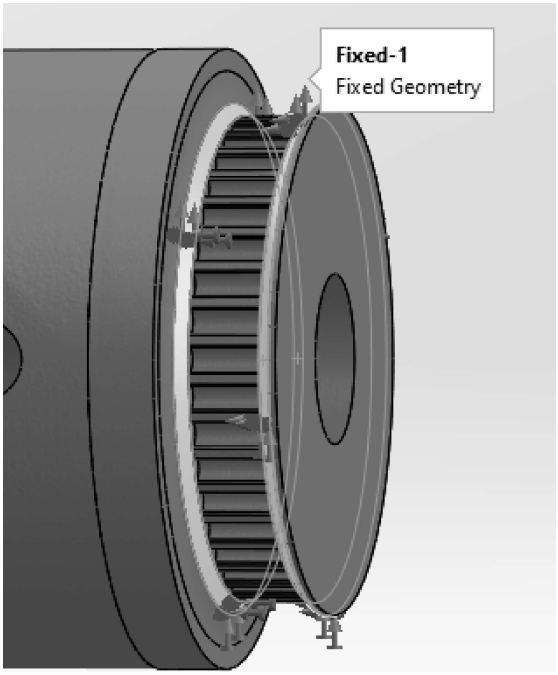


Figure 5.4: The fixed geometry for analysis

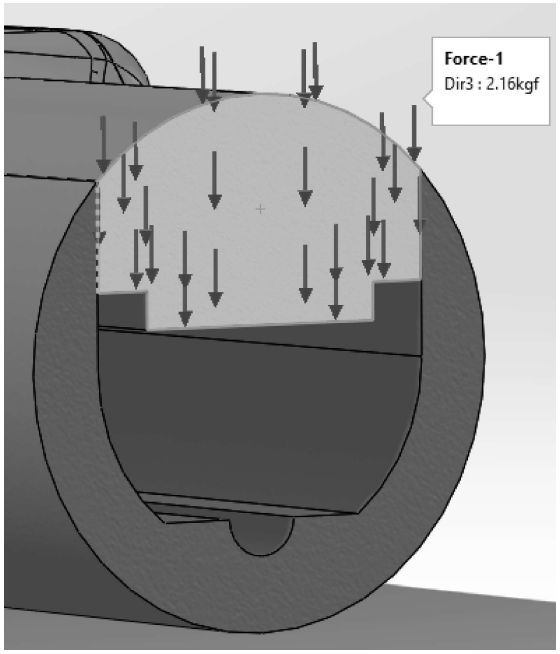


Figure 5.5: The external force of 2.16 kgf

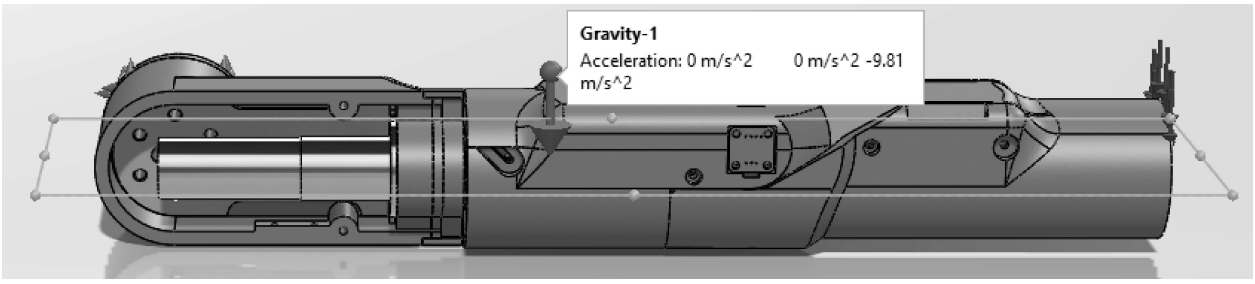


Figure 5.6: Gravitational force affecting the manipulator

The fixed geometry for the FEM analysis was selected to be the sides nearest to where the pulley would be fastened, see Fig. 5.4. The main load affecting the system is the payload force. It is applied to the model as a force affecting the end of the manipulator without the end effector, Fig. 5.5. This force is that which gives a rated max payload at the end of the end effector (25 cm further out, including the weight of the end effector) of 1.5 kg or 14.7 N. The load was decided to be given in kgf because it is then easy to measure and know if something is overweight. Also affecting the model is the gravitational force shown in Fig. 5.6. It affects the model as a weight for each individual part but is represented as a force applied to the center of gravity.

5.4 High Stress locations

The figures shown here are results from FEM analyses and are labeled accordingly. There are three tests in total. In the first one, the mesh of the whole part is coarse. In the second the whole mesh is fine, and in the third and last one, mesh-control was applied to the three focused parts. Probing happened at the exact same places every time.

5.5 First FEA

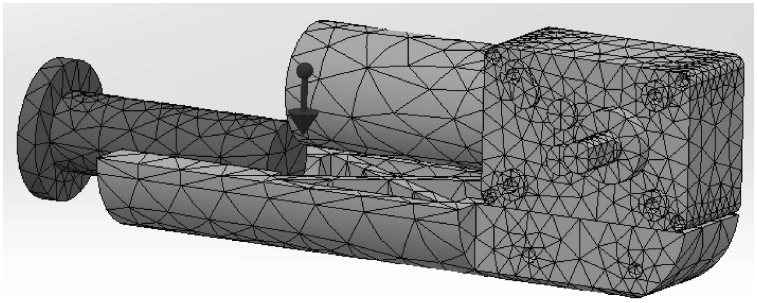


Figure 5.7: First FEA: Mesh-size of focused parts in an upward position, equal size for Sideways

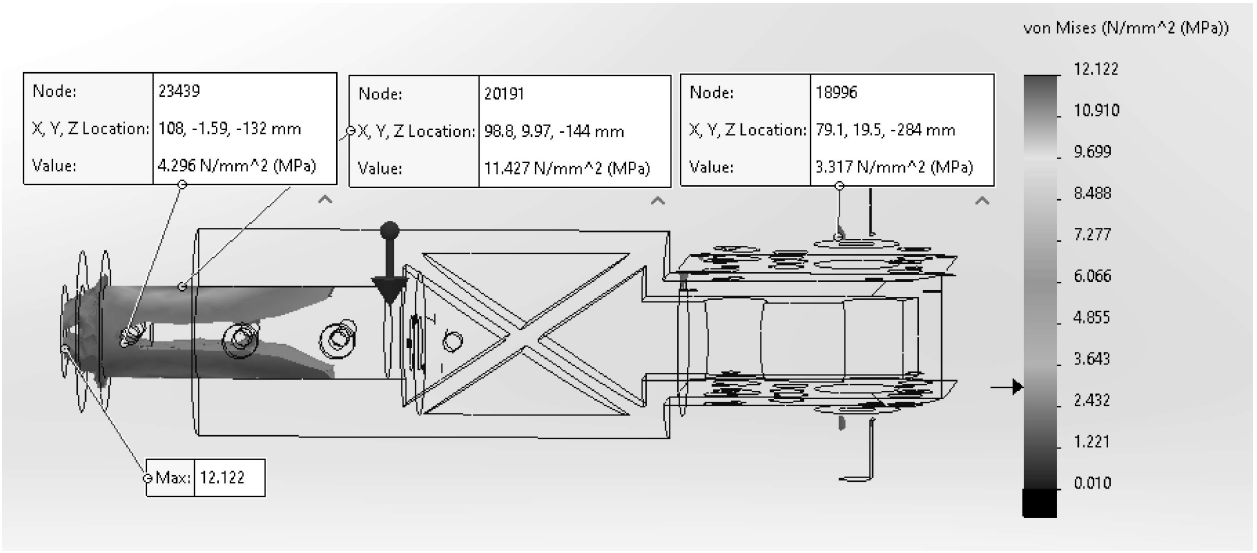


Figure 5.8: First FEA: Probing of test points in sideways position

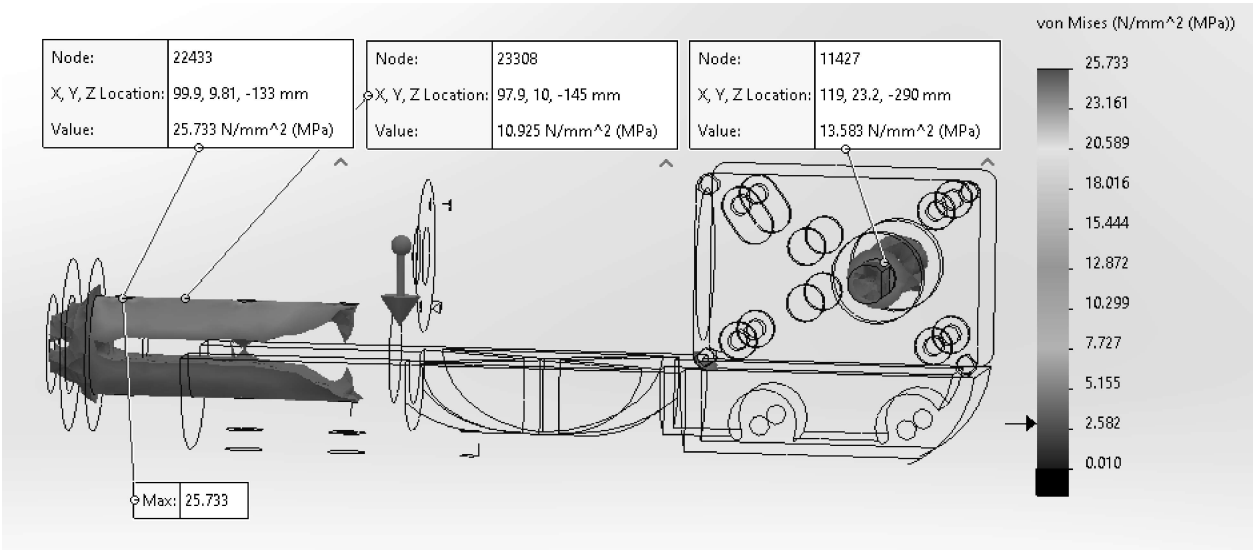


Figure 5.9: First FEA: Probing of test points in upwards position

5.6 Second FEA

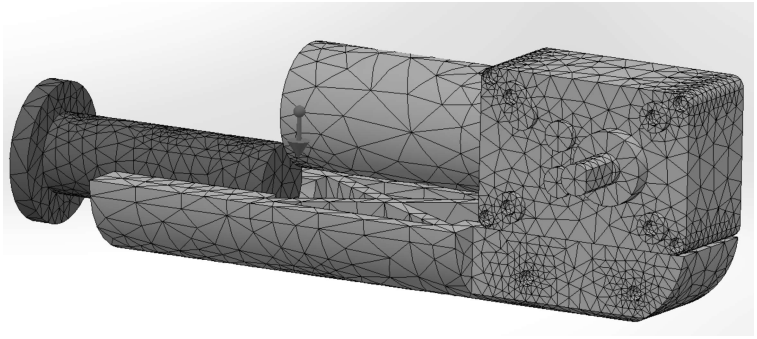


Figure 5.10: Second FEA: Mesh-size of focused parts in an upward position, equal size for Sideways

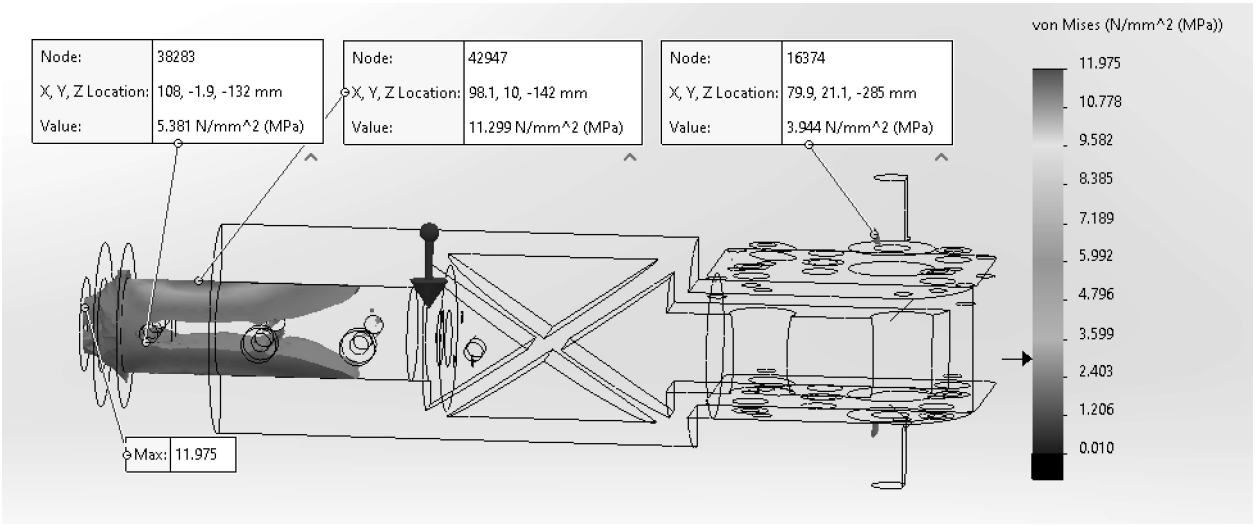


Figure 5.11: Second FEA: Probing of test points in sideways position

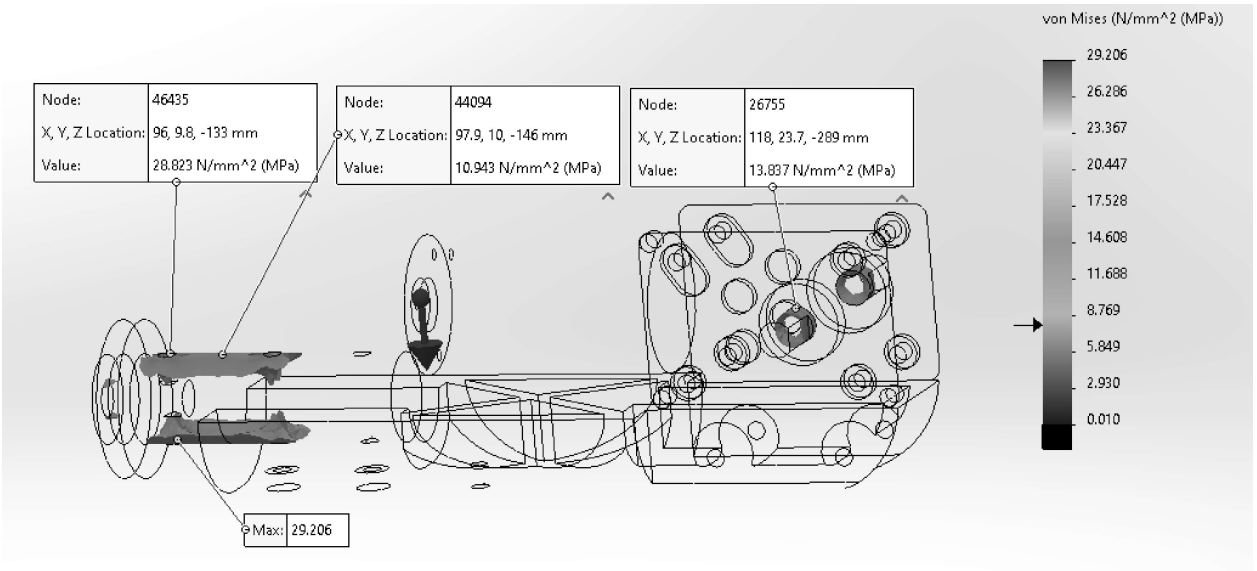


Figure 5.12: Second FEA: Probing of test points in upwards position

5.7 Third FEA

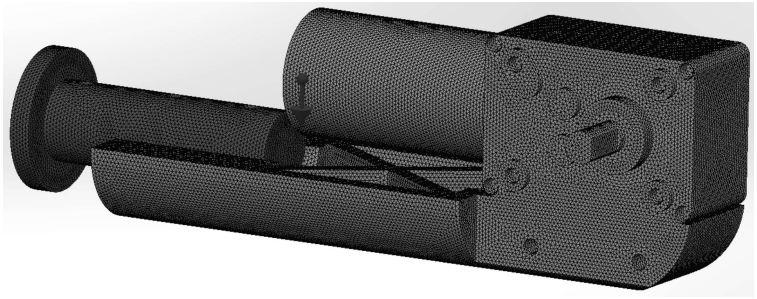


Figure 5.13: Third FEA: Mesh-size of focused parts in an upward position, equal size for Sideways

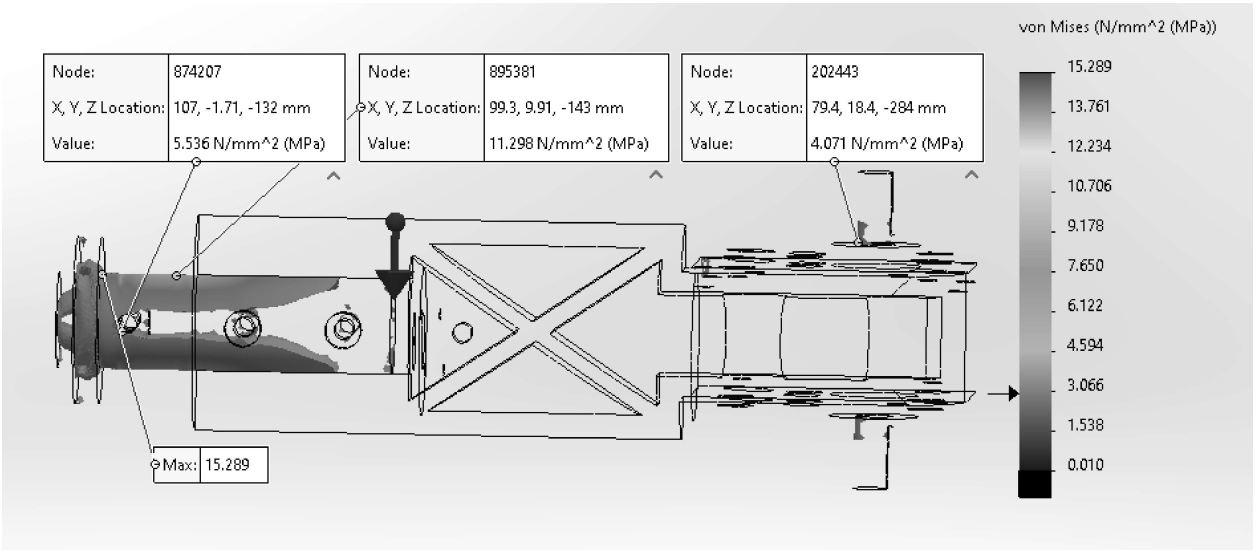


Figure 5.14: Third FEA: Probing of test points in sideways position

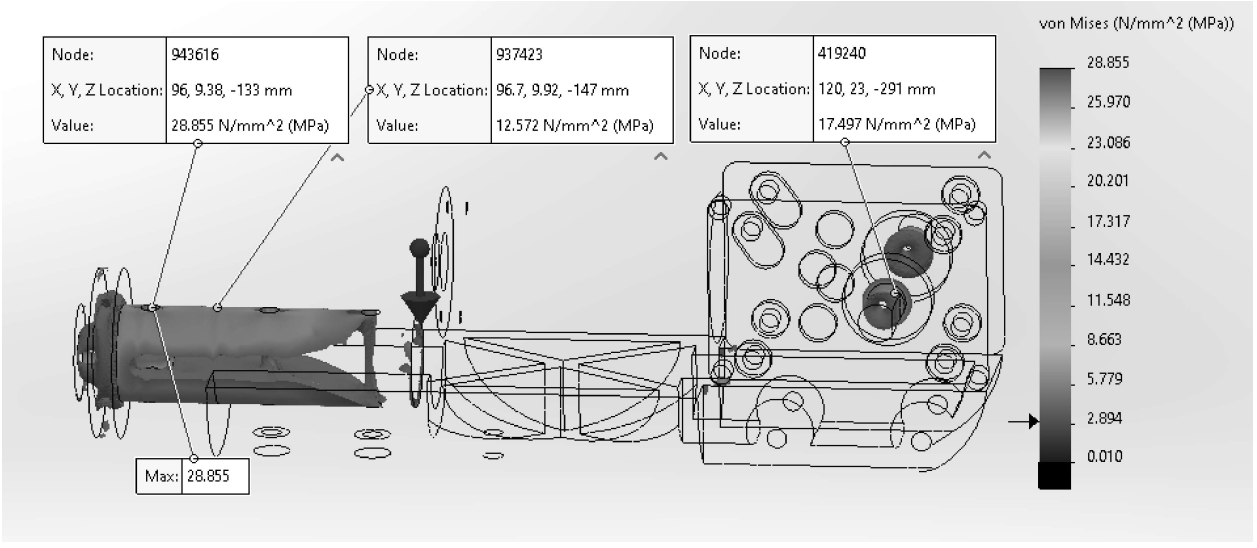


Figure 5.15: Third FEA: Probing of test points in upwards position

After these first three FEM analyses, the sideways testing seemed to all be of converging points, as seen in Fig.5.16. However the study of the manipulator in the upwards position had way higher stresses, and two of the points had not settled it seemed. Therefore three more FEM analyses were run, the probing results for these are in the appendix in Appx. C.

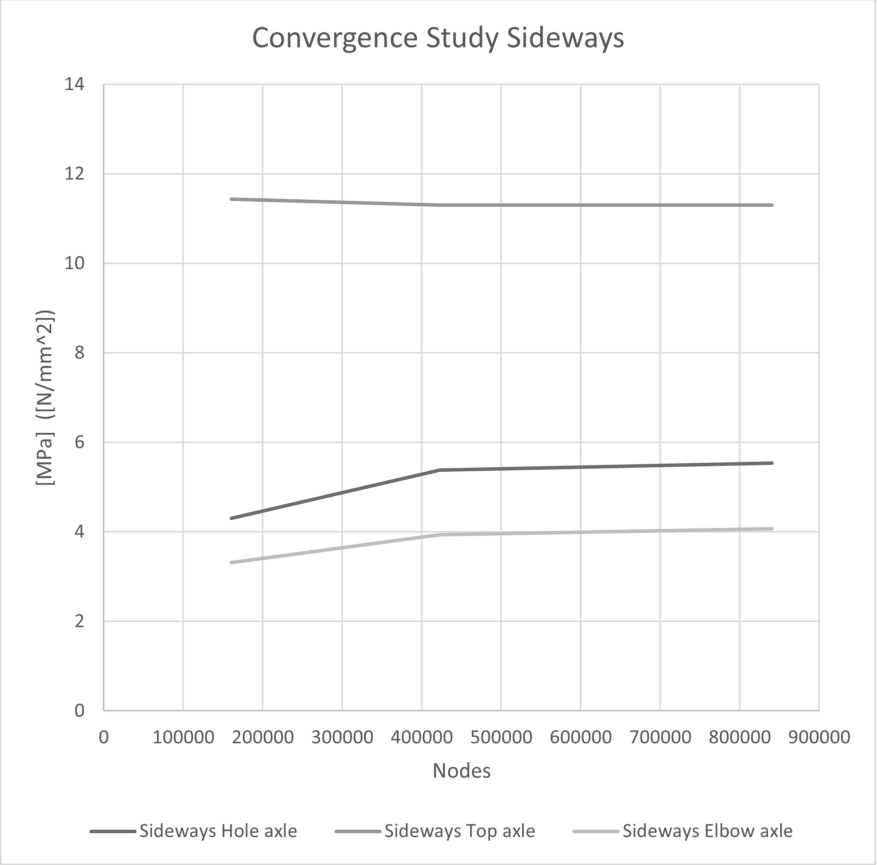


Figure 5.16: Convergence study of manipulator in sideways position

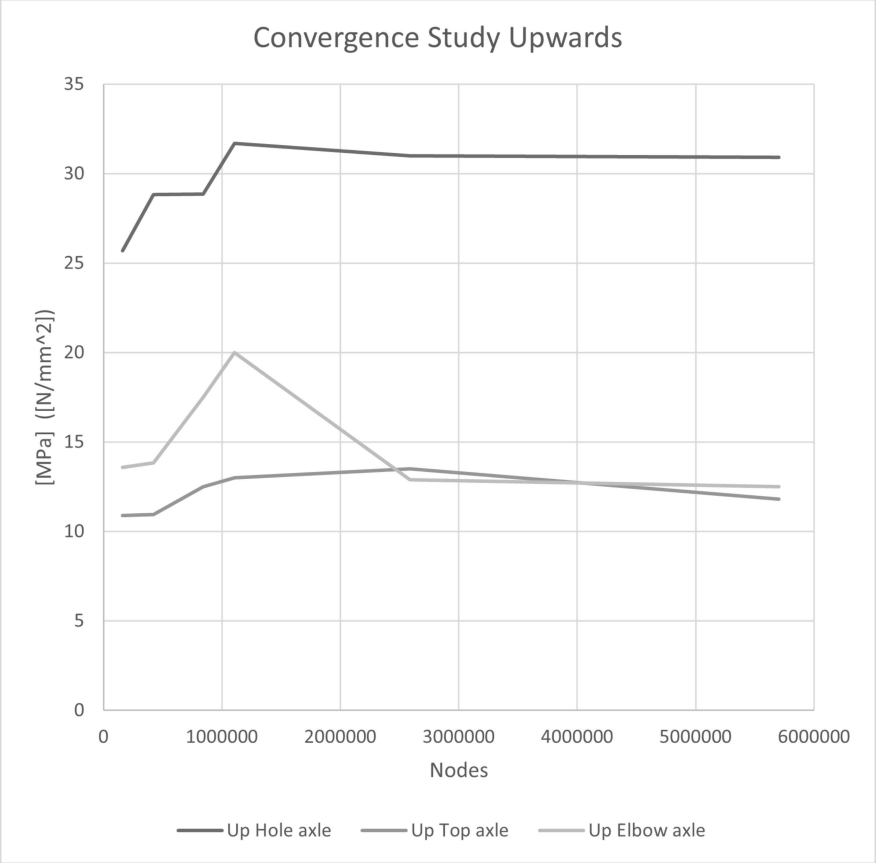


Figure 5.17: Convergence study of manipulator in upwards position

This study also converges as shown in Fig. 5.17. There are therefore no points that were probed that have a false stress concentration. Since the values for all points in the last two analyses increased very little or nothing at all, all points are therefore seen as converging points.

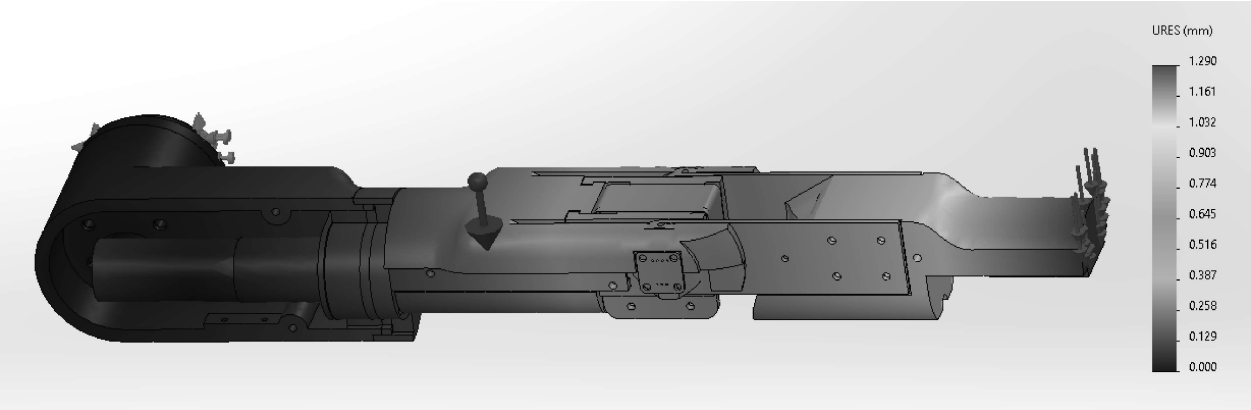


Figure 5.18: Displacement estimate from FEA

The FEM analysis showed a 1.3 mm displacement in the manipulator Fig. 5.18, however, this is not necessarily an accurate value. Because of the bonded component interactions, there could be extra flex in joints or individual parts. In addition, the materials do not give an accurate idea of how the model would behave since many of them are 3D printed.

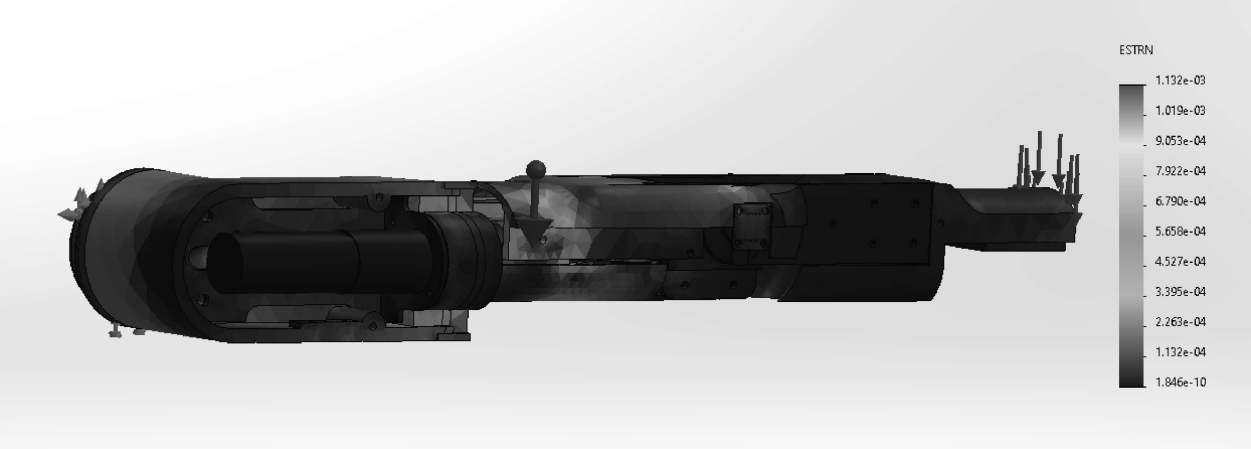


Figure 5.19: Estimated strain in parts

The strain shown in Fig. 5.19, is a way to see what parts experience the most elongation, contributing to the displacement in Fig. 5.18. However, due to the material being inaccurate and the bonding global interaction, this must also be seen as a rough estimate.

Chapter 6

Electrical design

When it comes to electrical parts for the manipulator, there are different criteria to consider. Such as what type of sensor is needed to get the appropriate feedback from the manipulator orientation and overall system stability. Also what types of communication channels they do use and if there is any need for extra components and provide them with the appropriate supply voltage. Proper choice of motor controllers ensuring that it can supply the actuator with enough power. The use of a microcontroller that have the appropriate amount of I/O that is needed and communication between the different devices. Since this is a SBV with an attached manipulator it is important to consider the use of an extra dedicated power supply that can supply the components and actuators. All this will be further discussed in the upcoming sections.

6.1 Micro controller

A microcontroller is a device that combines a processor, memory, and input/output (I/O) peripherals within a single integrated circuit. With the use of microcontrollers, these can easily be implemented as efficient, reliable, and cost-effective solutions for a wide range of applications. Programming a microcontroller can be done in Integrated Development Environments (IDEs) such as the Arduino IDE or Visual Studio [24].

In this project, two Arduino Mega microcontrollers are used to control a pair of robotic manipulators. Each Arduino Mega serves a specific function: the primary Arduino is responsible for controlling one manipulator and getting messages from the Loomo, while the secondary Arduino takes charge of the other arm getting messages from the primary Arduino. The decision to use two separate Arduino Megas comes from the fact that a single Arduino does not possess an adequate number of PWM pins to manage both manipulators. This is due to the used motor controller's requirement for two PWM signals to control the motor's movement in both directions.

6.2 UART

Serial communication with UART (Universal Asynchronous Receiver/Transmitter) is a method used for transferring data between two devices, one bit at a time, over a communication channel. It is serial because the packages are sent in series, and "asynchronous" because the transmitter and receiver use their own internal clocks to send and receive data packages, without relying on a shared clock signal. For reliable communication, both the transmitting and receiving devices must have the same baud rate. Errors can occur if the transmitter and receiver have a different baud rate or noise on the communication channel [16].

For the communication between the two Arduino Mega's, a UART serial communication protocol is utilized. This enables the microcontrollers to exchange data seamlessly and efficiently, ensuring smooth operation of the manipulators. Additionally, the primary Arduino is connected to the Loomo's USB port, which also uses UART for communication purposes. This connection allows

the primary Arduino to interface with the Loomo platform effectively, ensuring the system's overall functionality and performance.

6.3 Brushed DC-motor

A brushed DC motor is a type of electric motor that runs on direct current (DC) power and uses mechanical brushes to transfer electrical current to the rotating parts. It consists of a stator and a rotor. The stator consists of permanent magnets, while the rotor is a coil of wire wound around an iron core.

When a voltage is applied to the motor's terminals, current flows through the brushes and into the rotor. The current in the rotor generates a magnetic field that interacts with the magnetic field from the permanent magnets in the stator, this causes the rotor to rotate. The commutator makes it possible for the rotor to rotate continuously by reversing the current direction e.g. each half revolution [19].

These types of motors are simple in design, easy to control, and low-cost. However, they have some disadvantages, such as limited lifespan due to brush wear, increased maintenance requirements, and lower efficiency compared to brushless DC motors.

Since there was a relatively small budget for this project and brushed DC motors are cheaper than brushless and they are easier to control. Therefore it was natural to choose a brushed DC motor over a brushless one. There were chosen different types of geared motors as shown in the table. 4.3. This is because there was different requirement such as torque and application on the different joint of the manipulator. Also, different voltage levels were considered, both 12V and 24V. There was selected a motor running on 24V because it needs less current compared to 12V, also the Loomo can supply 24V. Resulting in an easy way of designing a battery pack to drive the motors and recharge it from the Loomo's output port if that is needed.

Choosing the right motor for each of the joints is based on the calculation of the static equilibrium of the manipulator. Since the moment arm is longest from the shoulder to the e.e the planetary geared 31mm DC motor was chosen in collaboration with a belt drive. This is because of the strength and the easiness to mount the motor at the back of the Loomo. The same motor was used for the torsion of the elbow but with direct drive.

For the elbow joint, there was used a dual shaft worm geared motor. This is done since it had a design that made it easy to mount it in the link. With the dual shaft, it was used to connect the elbow link easily directly to the motor.

Lastly, there was ordered a 20mm planetary geared motor to be used on the e.e. It was supposed to be used for rotating the e.e and closing a gripper.

6.4 CJMCU-7960 Motor controller

The motor controller consists of an H-bridge is an electronic circuit that enables the control of a motor's direction and speed by reversing the voltage and current applied to the motor. Essentially, it consists of four MOSFETs which function as switches to direct the flow of current through the motor.

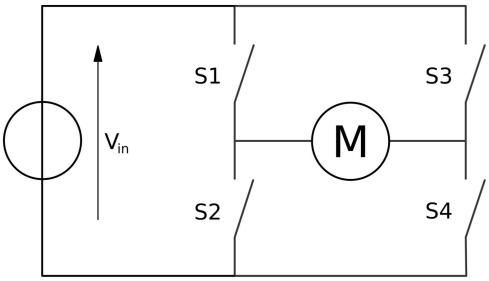


Figure 6.1: H-bridge [25]

The four MOSFETs are arranged in a square formation, with the motor connected in the middle so the circuit looks like an "H" as shown in figure 6.1. When a specific pair of diagonal switches is activated (S1 and S4), current flows in one direction through the motor, causing it to rotate in a particular direction. To reverse the motor's direction, the other pair of diagonal switches is activated, causing the current to flow in the opposite direction [8].

A motor controller can control the motor's speed. This is accomplished by using pulse width modulation (PWM). PWM varies the duty cycle of a periodic signal, essentially controlling the proportion of time the signal is on and off. By modulating the voltage applied to the motor using PWM, the speed of the motor can be accurately controlled.

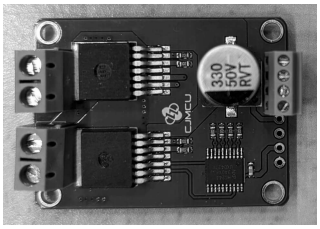


Figure 6.2: CJMCU-7960 BTS7960B motor controller

The motor controller of the type CJMCU-7960 BTS7960B was used in this project because these were available at the university. They also met the requirements for supplying the different motors with enough voltage and current. These motor controller are shown in figure 6.2

6.5 ACS712 Current sensor

A current sensor that uses the Hall effect to measure current is a type of device that detects the magnetic field generated by the flow of electrical current in a conductor. A Hall effect sensor produces a voltage depending on the magnetic field that is created when current is flowing through a conductor. When an electrical current flows through a conductor, it generates a magnetic field around it. The strength of this magnetic field is proportional to the output voltage [7].

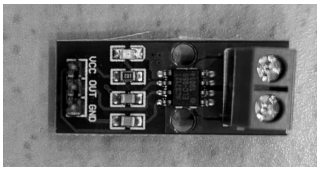


Figure 6.3: ACS712 current sensor

An ACS712 current sensor is used as shown in the figure. This sensor is connected to each of the different motors to be able to measure current to monitor the power usage on each of the motors to get information if the motor operates over the rated and/or near stall torque.

6.6 Absolute and incremental encoder

Absolute and incremental encoders are devices used for measuring position, rotation, or movement. They convert the position or rotation of an object into a digital or analog signal that can be read by a control system.

An absolute encoder provides a unique code (output) for each position or angles it measures. This means that even after a power loss or system restart, the encoder can determine its exact position without the need for a reference (home) position. Absolute encoders are typically more complex and expensive than incremental encoders but offer an absolute position of an object.

An incremental encoder generates output signals relative to the movement from its previous position. It does not store or provide absolute position information. To obtain position information, the control system must count the number of pulses and keep track of the direction of movement. Incremental encoders are generally simpler and less expensive than absolute encoders, but they require a reference (home) position after a power loss or system restart to re-establish their position.

The main difference between absolute and incremental encoders lies in the way they provide position information. Absolute encoders give unique codes for each position, allowing them to remember their exact position even after a power loss. Incremental encoders provide relative position information and require a reference position to determine their exact location [3].

6.7 AS5600 Magnet encoder

A magnetic encoder like the AS5600 operates by detecting the rotation of an object through the use of magnetic fields. A permanent magnet is fixed to the rotating object, such as a motor shaft. A magnetic sensor that uses a Hall element is positioned nearby the rotating magnet and mounted in a fixed location. As the object rotates, the magnetic field produced by the magnet also rotates. The Hall element senses the changes in the direction and strength of the magnetic field as the magnet rotates. The sensor converts these changes into electrical signals that represent the object's rotational position [4].

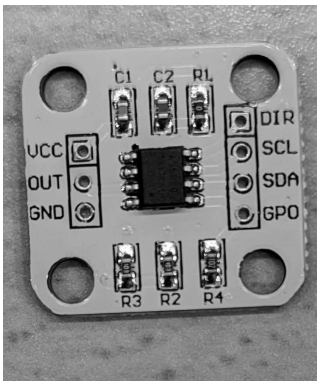


Figure 6.4: AS5600 magnetic encoder

The AS5600 as shown in figure 6.4 magnetic encoder operates by sensing the magnetic field of a diametrically magnetized magnet placed on the rotating part of the joint. This encoder translates the magnetic field changes into digital output signals that correspond to the angular position of the joint. This encoder was chosen due to its ability to be mounted on a fixed structure, making it suitable for these particular joints. This results in an easy way of measuring the angle of the shoulder and elbow joints.

6.8 Motion Processing Unit

A Motion Processing Unit (MPU) is a microelectromechanical system (MEMS) that combines several sensors to measure motion and orientation. The primary components of an MPU are an accelerometer and a gyroscope. These sensors provide information on linear acceleration and angular velocity [13].

The MPU processes data from these sensors to calculate an object's movement and orientation in three-dimensional space. By merging data from multiple sensors, the MPU can deliver more precise and dependable measurements than when using individual sensors separately.

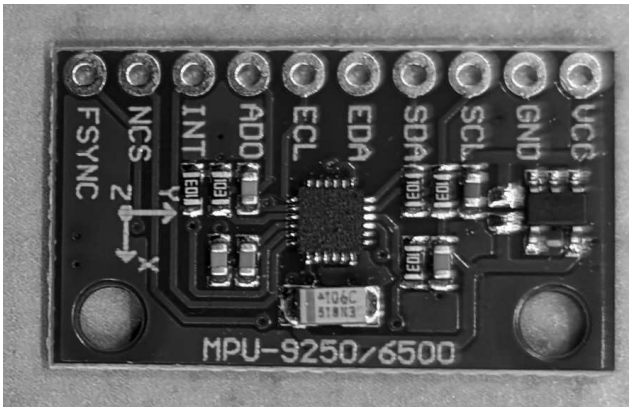


Figure 6.5: MPU9250 sensor

Since it was not possible to mount the magnetic sensor directly on the joint responsible for rotating the elbow, known as the torsion joint. To overcome this challenge, a different approach was taken by implementing two MPU-9250 sensors. This sensor is shown in figure 6.5. These sensors were placed on the shoulder link and elbow joint to enable the comparison of angles between them. By comparing the data from both sensors, it becomes possible to calculate the relative angle between the shoulder link and elbow joint. This angle provides an indirect measurement of the torsion joint's orientation for the elbow.

6.9 I2C

I2C is a communication protocol used to transfer data between a controller and a device. I2C, short for Inter-Integrated Circuit, is often called a two-wire interface because it only requires two wires for data transfer: SDA (data line) and SCL (clock line). The data is transferred serially on the SDA line. The SCL line is the clock signal used to synchronize the processes between the controller and the device. A drawback of I2C is that it is half-duplex, meaning it cannot send and receive data simultaneously due to having only one data transfer line [6].

I2C devices that work with different voltage levels (e.g., one at 5V and another at 3.3V). Connecting these devices can cause damage or improper functionality due to voltage incompatibility. An I2C level converter is an electronic device used to connect two I2C devices operating at different voltage levels by converting the voltage levels of the I2C signals between them. An I2C level converter ensures safe and reliable communication between I2C devices that operates with different voltages. To allow data to be transmitted in both directions while maintaining correct voltage levels for each device the level converter consists of a bidirectional buffer [23].

An I2C multiplexer allows multiple I2C devices to be connected to a single I2C bus. It simplifies communication between a microcontroller and multiple I2C devices, by providing multiple separate communication channels on a single bus. The microcontroller can select which device it wants to communicate with at any time by toggling the corresponding channel on the multiplexer, so this

means that an I2C multiplexer works like a switch. This allows for easy management of multiple I2C devices, even if they have the same address to prevent address conflict [22].

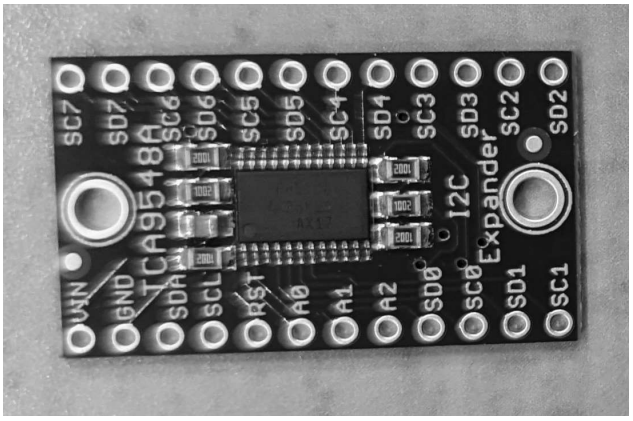


Figure 6.6: TCA9548A multiplexer

The motion and orientation sensors rely on a sensor that utilizes I2C serial bus communication to transmit data. In this project, several identical sensors are used, each possessing the same address. Consequently, it becomes crucial to implement a multiplexer in order to accurately obtain sensor readings from the appropriate device. The specific multiplexer chosen is shown in figure 6.6

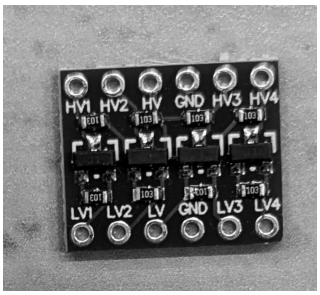


Figure 6.7: I2C level converter

The MPU-9250 operates on a 5V power supply and communicates with the Arduino through a 5V I2C interface. On the other hand, the AS5600 sensor requires a 3.3V power supply and necessitates a 3.3V I2C communication with the Arduino. To address the difference in voltage requirements, an I2C level converter is implemented. This level converter is shown in figure 6.7

This converter is connected between the Arduino and the AS5600 sensor, transforming the signal from 5V to 3.3V and vice versa, without ruining the data transmitted across the bus. This solution enables the sensors to function with each other, despite their difference in voltage requirements, and ensures accurate and reliable motion and orientation detection.

6.10 LM2596S Buck converter

A DC-DC buck converter takes an input voltage and converts it into a lower output DC voltage while increasing the current. The buck converter achieves this by quickly switching the current flow on and off, which adjusts the voltage and current levels as needed. This process makes it more energy-efficient compared to linear regulators, which dissipate energy as heat [10].

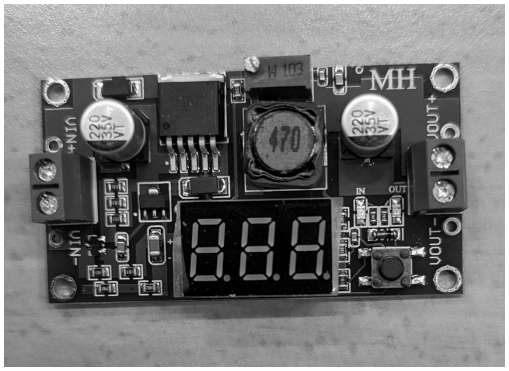


Figure 6.8: LM2596S Buck converter

To supply the components used in this project there was chosen to implement a buck converter. As shown in figure 6.8 is the chosen buck converter. This converter could deliver 2 A and a maximum of 3 A, since the controllers use 1 A in total and the rest of the sensor power consumption is low these could be neglected. All the sensors require 5 V except the magnetic encoder, there is only needed one buck converter to supply the components. The magnetic encoder is supplied from the Arduino 3.3V pinout.

6.11 Battery pack

The Loomo can supply 24V/1A output from its hardware kit, which makes it necessary to implement a battery pack to provide enough supply for the various components. The voltage capacity was chosen based on the selected motor type, which required 24V, while the capacity in ampere-hours (Ah) was determined by considering the duration for which the manipulator could be operated and the need for a counterweight to better be able to balance Loomo.

Table 6.1: Power supply requirements

Component	Rated current	Voltage	Quantity
NFP-GA32Y-31ZY-EN	1.0A	24V	4
NFP-5840-31ZYS-D	1.2A	24V	2
NFP-GA20Y-180	0.21A	24V	4
Arduino Mega 2580	0.5A	5V	2

Table 6.2: Battery pack results

Nominal voltage for each cell	3.6V
Current capacity for each cell	3000mAh
Desired voltage	24V
Desired capacity	12000mAh
Number of cells in series	6
Number of cells in parallel	4
Battery pack size	6S4P

The requirement for the system to run is shown in table 6.1. Table 6.2 shows the desired voltage and capacity and what size of the battery these values resulted in.

Figure 6.9: Battery pack for Loomo manipulators

It was decided, since the cells were on hand, to oversize the battery pack capacity. Longer battery life is always good, but in this case, it could function as a counterweight as well, compensating for the manipulator's weight to easier balance Loomo. As shown in figure 6.9 is the result of the assembled pack and the chosen shape. The pack was assembled using a spot welder instead of soldering the nickel strips because heating the cells too much can damage them. Since the manipulator's pack have better capacity than the Loomo's there was decided not to recharge it from the hardware bay. A DC-DC Buck converter was implemented to reduce the 24V input to 5V, thereby powering the Arduino and the sensors. The AS5600 magnetic encoder is a 3.3V sensor, this is powered directly from the Arduino's 3.3V voltage pin.

When designing a battery pack there is important to have a Battery Management System (BMS) which is a component that monitors, control, and protect the performance and safety of rechargeable battery packs. The main task of a BMS is to extend the life, maintain optimal performance, and ensure the safe operation of a battery pack [21].

When charging and discharging the battery pack, there was implemented a BMS which is responsible for ensuring that all cells within the pack maintain a similar charge level. This process, known as cell balancing, is essential for preventing overcharging, over-discharging, or uneven distribution of charge among the cells, which could lead to decreased battery life or failure.

Chapter 7

Wiring diagrams

Here the connection of different components will be described. All sensors will be placed inside the manipulator except the current sensor. The power supply and the control units will be placed in a "backpack" on the back side of Loomo.

7.1 Power supply and serial communication wiring diagram

Figure 7.2 is a representation of the connection for supplying both Arduino and the serial communication between them. The 5V terminal block connected with the buck converter will supply the various components, such as sensors and Arduino that operate on 5V. There will be made a simple PCB board for the 5V input and the output will have multiple terminal blocks connected in parallel to supply the various components. Also, a circuit board that can provide 24 voltages to supply the motor controllers.

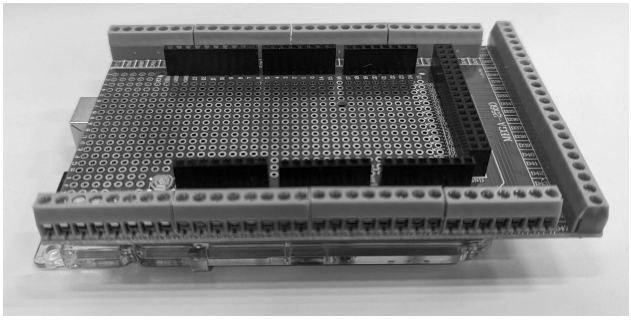


Figure 7.1: Arduino terminal block shield

To make the cable connection more solid and ease of rewiring there is mounted a terminal block shield on top of both the Arduino, as shown in figure 7.1. Here will all the connections to the different motor controllers and sensors be connected.

By connecting the TX from primary to RX on the secondary Arduino and the opposite, will open up serial communication between them. Also connecting a USB cable between Loomo and the Primary Arduino to send commands.

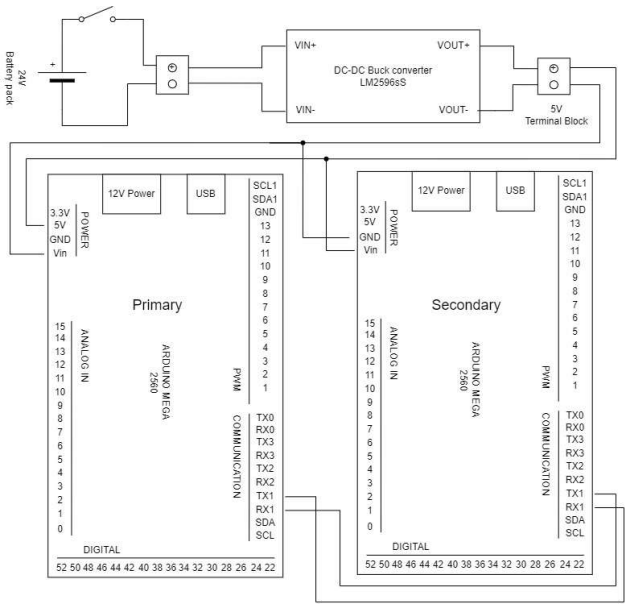


Figure 7.2: Power supply circuit diagram

7.2 Motor controller wiring diagram

As shown in figure 7.3 is the wiring diagram for the motor controller. The 5V terminal block will supply the enable pins on each of the motor controllers, to make it easier and there is no need of using any digital output pins on the Arduino. This motor controller needs two PWM pins to control a motor in CW and CCW directions. This resulted in the need for two Arduinos. In this configuration, three motors per manipulator are used, but it is possible to add three more for each manipulator. So it is possible to have at least a total of 12 motors.

To be able to control the torque provided by each motor there is implemented a current sensor between the motor controller and the motor. This is done to get feedback from each motor to know if one of the motors is starting to stall. This means that the max payload is reached.

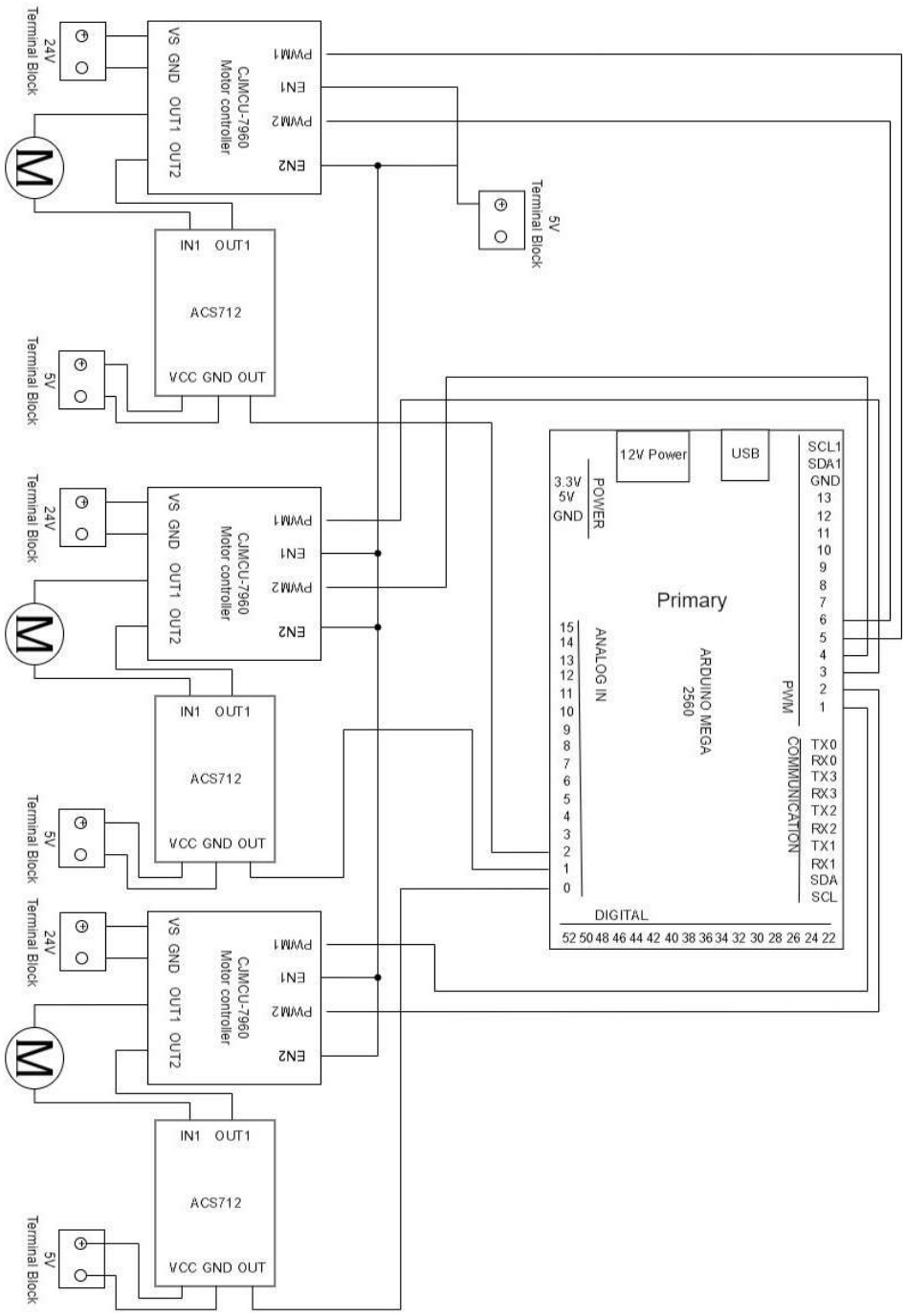


Figure 7.3: Wiring of the actuation system

7.3 Wiring diagram for position and orientation sensors

Here is an overview of the wiring diagram shown in figure 7.4 and 7.5.
A total of four AS5600 magnetic encoders are used, two on each manipulator. This encoder is the only component that is in need of a 3.3V power supply. Since the Arduino has a 3.3 output pin this could be utilized to supply the encoder instead of adding another buck-converter. The direction pin (DIR) can either be connected to Vcc or Gnd, depending on the application. If it is connected to ground values increase when it rotates clockwise and when connected to Vcc value increase with counterclockwise rotation.
An MPU9250 is utilized to get the motion and orientation of the manipulators. In the designed manipulator there is used a total of five MPUs, two on each manipulator, and one is supposed to be in the backpack. For an end effector there is planned to be connected two extra MPUs on each manipulator, this is for the rotation of the e.e.

A multiplexer is connected between the AS5600 and the MPU. This is due to that the I2C address is equal for the same type of sensor. There is used two multiplexers one for each of the manipulators. There is possible to connect a total of four more sensors, two at each manipulator.

Since the AS5600 communicates over I2C with 3.3V and Arduino 5V, there is in need for an I2C-level converter. This is connected between the multiplexer and the magnet encoder. Multiplexer is connected to the high voltage side (HV) and the encoder is connected to the low voltage side (LV). The level converter is supplied with both 3.3V and 5V.

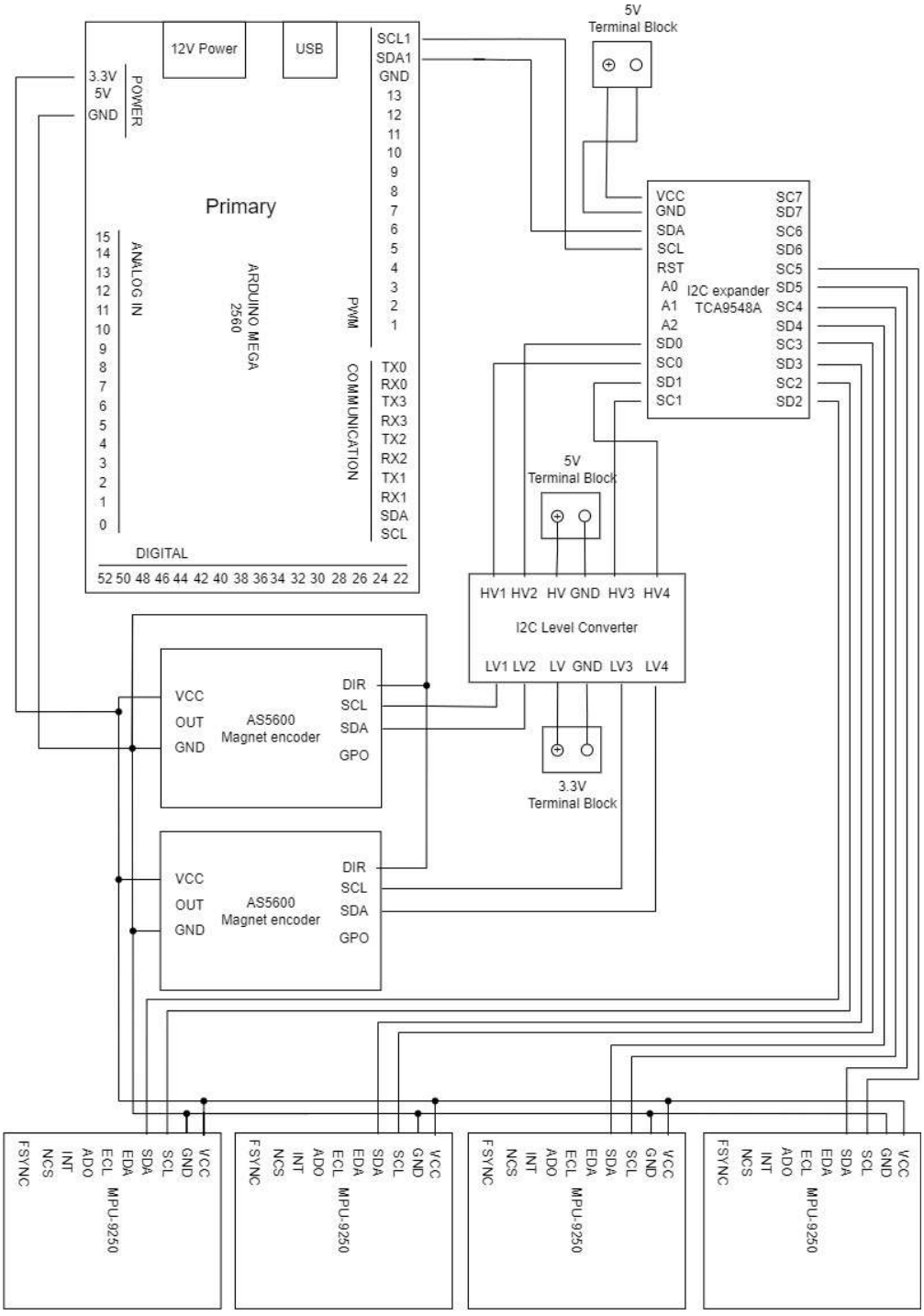


Figure 7.4: Connection diagram for the angular sensors

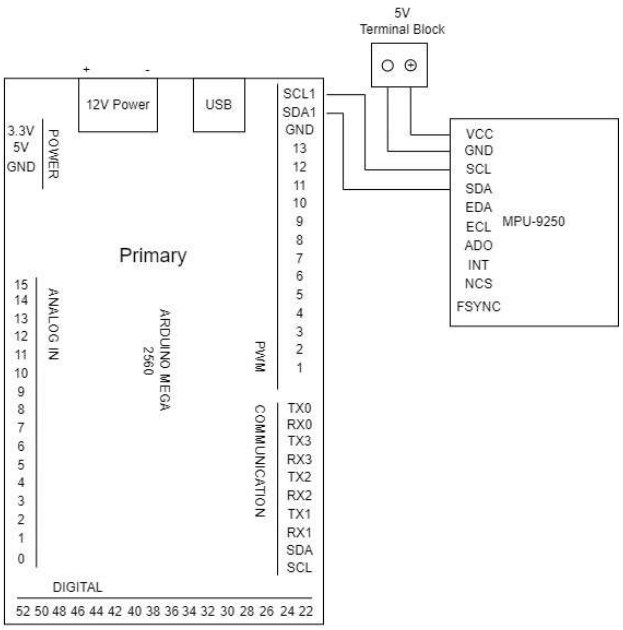


Figure 7.5: MPU for the backpack on Loomo wiring diagram

Chapter 8

Control system architecture

This part of the project is paper only since there was no implementation of the final program. All the parts functioned with components separately, so the remaining job is to put them together.

8.1 Android Studio 3.5.1

Android Studio is an Integrated Development Environment (IDE) designed for building Android applications. It is developed and maintained by Google and is the official IDE for Android app development. Android Studio provides everything needed for to develop Android apps from writing, testing, and debugging. This editor makes the app development process easier for the Android platform.

Android Studio provides a code editor that supports various programming languages like Java, Kotlin, and XML. This IDE offers features like code auto-completion, syntax highlighting, and error detection. This editor provides an easy way to design user interfaces (UI) by dragging and dropping components like buttons, text views, and images onto the screen. Android Studio uses Gradle as its build system, this manages the dependencies, compiling code, and generating the final APK (Android Package) file. The APK file is installed on the device for users to access the app. The editor also provides a debugging tool to make it easy to identify and fix issues in the code. Version Control is integrated into Android Studio and supports different such as GitHub. This makes it easy for developers to collaborate, see changes and updates that are done, and manage code versions throughout the development process.

8.2 Coding in C++

C++ is the coding language used for Arduino and many microcontrollers. It is a very basic language that is closer to machine code compared to many widely used languages like Python. This makes the programs written in C++ lightweight and perfect for applications like microcontrollers that focus on being small and inexpensive. C++ is an object-oriented coding language meaning that functionality is usually assigned to an object. That object can in turn be used as many times as wanted throughout the code, making big projects scale very well.
The native coding environment, IDE, for Arduino, is the Arduino IDE. This is a great program for smaller projects because of its simplistic UI and few options. However, the need for better debugging and easier file management is obvious. By using a program called "Visual Micro", programming can be done in Visual Studio while Arduino is in the background. This simplifies the process immensely and is recommended.
The libraries "Wire" and "invensense imu" for MPU9250 are both necessary for the program to work. The "wire" library makes it possible to communicate over I2C with the sensors that are wanted, with fewer wires. The library for MPU9250 helps simplify the process of retrieving data from the MPU9250 IMU.

```
#include "invensense_imu.h"
#include "Wire.h"
#include "arduino.h"
#include "motor.h"
#include "serial.h"
#include "magI2C.h"
#include "mpu9250.h"

```

Figure 8.1: C++, Arduino libraries

8.3 Communication between Arduino and Loomo

Different libraries are added in the Android project as shown in 8.2. This makes it possible to set up communication with Arduino using a USB connection. Also, be able to implement buttons and textview for the UI.

```
import android.hardware.usb.UsbDevice;
import android.os.Handler;
import android.app.Activity;
import android.os.Bundle;
import android.text.method.ScrollingMovementMethod;
import android.widget.TextView;
import android.view.View;
import android.widget.Button;
import me.aflak.arduino.Arduino;
import me.aflak.arduino.ArduinoListener;
```

Figure 8.2: Java, Android studios libraries

```
    private View.OnClickListener homeListener = new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            arduino.send("H".getBytes());
        }
    };

    private View.OnClickListener waveListener = new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            arduino.send("W".getBytes());
        }
    };

    private View.OnClickListener stopListener = new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            arduino.send("S".getBytes());
        }
    };
}
```

Figure 8.3: Java, Simple code for sending commands to Arduino

The Java code as shown in figure 8.3 is part of a simple example application on how to communicate with an Arduino device via USB using the 'me.aflak.arduino' library. The application contains a simple user interface with buttons such as "Home", "Wave" and "Stop" as seen in 8.4. The onClickListeners are defined for each button that is connected to the UI. When a button is pressed Loomo sends a command to Arduino which are the following 'H', 'W', and 'S'. This program only sends commands and does not get any feedback from Arduino on how the manipulator behaves. All the calculations and controls are supposed to be on the Arduino. The following GitHub projects are used to implement this communication (1 [15] and 2) [5]

Figure 8.4: User interface on Loomo

```
1  void loop() {
2  if (Serial.available() > 0) { /* Check if there's data available on
3  the serial port */
4      char command = Serial.read(); // Read the incoming byte
5      switch (command) {
6            case 'W': // Start Wave function
7              // Wave
8              break;
9            case 'H': // Go to home position
10             // Home
11             break;
12           case 'S': // Stop wave function
13             // Stop
14             break;
15         }
16   }
17 }
```

Figure 8.5: C++, Arduino skeleton code for retrieving commands from Loomo

The Arduino code is shown in figure 8.5 is a simple program that checks if there is data available on the serial port each scan cycle. When a command is sent from Loomo there will be data on the serial port that is read. This command will then be stored in the "command" variable. Depending on the incoming command from Loomo either of the cases is run and the manipulator can e.x start a wave function, go to the home position or stop.

8.4 Actuator control

Actuator control needs to happen on the Arduino since it is connected to all the actuators and sensors. The Arduino will have many pins connected since there are many actuators. Therefore the program for control was written with the use of classes. This makes the code very scalable, easy to implement new features with, and easy to use. There is one class called "Motor" Fig. 8.6, which is used to set up and control the actuators through a built-in PID controller member function.

```
1   #pragma once
2   class Motor
3   {
4   public:
5       Motor(int cwNew, int ccwNew, float KpNew, float KiNew, float KdNew);
6       ~Motor() = default;
7
8       int assignedPins(int cwNew, int ccwNew);
9       int PIDvalues(float KpNew, float KiNew, float KdNew);
10
11      int PID(double ang, double desAng, long int time);
12
13      int drive(bool doDrive);
14
15      int cw;
16      int ccw;
17      float Kp;
18      float Ki;
19      float Kd;
20      int PWM;
21      bool dir;
22      float summedError;
23      long int lastTime;
24      float lastAng;
25  };
```

Figure 8.6: The declaration of the Motor class in the header file

The code to read angles off of the magnetic encoder is strongly inspired by "Curious Scientist" [18] code but made into a class with member functions, local variables and changed slightly to output angle, and not angle compared to starting position.

```
2   class MagI2CUnit {
3   public:
4       MagI2CUnit(int ID, int &deviceCounter);
5       void I2CComm();
6       int getID(bool printMe);
7       int I2CScan();
8
9       float readAngle();
10
11      void checkMagnetPresence();
12      bool checkMagnetPresence(bool onlyOnce);
13      float trueAngle;
14      int magnetStatus;
15      int slaveAdress;
16  };
17
18  int I2CScanGeneral();
```

Figure 8.7: The declaration of the MagI2CUnit class in the header file

The "compFilter" class, Fig. 8.8, is one that handles reading values from the IMUs with a complimentary filter built-in member function. The code for this class was made mostly by chatGPT, but it was not given in the form of a class. Making it a class makes the use of the code easier and helps

possible future projects.

```
2   class MagI2CUnit {
3   public:
4       MagI2CUnit(int ID, int &deviceCounter);
5       void I2CComm();
6       int getID(bool printMe);
7       int I2CScan();
8
9       float readAngle();
10
11      void checkMagnetPresence();
12      bool checkMagnetPresence(bool onlyOnce);
13      float trueAngle;
14      int magnetStatus;
15      int slaveAdress;
16  };
17
18   int I2CScanGeneral();
```

Figure 8.8: The declaration of the compFilter class in the header file

The class variables are public purely for testing purposes. It should not be necessary to have these public in the final script since member functions can limit accessibility in a much better way if variabes are not directly accessible.

8.5 Kinematics

Controlling the manipulator. When moving a manipulator to a position it is normal to know the position of the desired point, and not the angles required to position the end effector there. Therefore, calculating the angle values from the position of the manipulator's end effector position is required. This is called inverse kinematics and is an advanced form of robotics.

The solution will vary based on how the joins are oriented relative to each other and vary greatly in complexity. Many times the Inverse kinematics result in many solutions, or manipulator configurations, that gives the required position of the end effector. However some of these might be invalid since the manipulator collides with itself, or another restriction is applied. Usually, each angle has a range that the joint allows movement within. Also, the geometry of the manipulator itself introduces restrictions for certain movements. For manipulators mounted on a moving robot like this, not only must it take into consideration the angle restrictions and the Loomo itself, but also surrounding objects and people. When all of these invalid solutions are filtered out, the best one needs to be selected. This problem of having too many valid solutions is more common in higher DOF robots, but will still be present with this 3 DOF manipulator.
While the Inverse kinematics give angles based on position, the angular speeds based on cartesian velocities are calculated from the inverse Jacobian matrix. With both of these, the current to the motors can be controlled by a cascade control from angle and angular speed values. The set values would come from a path-planning algorithm.

Chapter 9

Results

9.1 Final prototype

Figure 9.1: The final prototype of the manipulator

The design of the prototype was supposed to look like a human arm. Also, it had to keep the Loomo's original functionality and the possibility of still being able to ride it. Mainly the structural parts are made of aluminum, steel, and 3D-printed parts with reinforcing filaments such as kevlar. The manipulator has 3 DOF ensuring it has a relatively large workspace. From testing with the final prototype, it is still possible to ride Loomo when the manipulators was mounted. Figure 9.1 shows the final prototype.

The YouTube link to the project: Loomo Manipulators

9.2 Achieved qualities

The manipulator is not finished, but the prototype still has achieved many of the design specifications:

- **Reaching the buttons, at 110 cm up on the wall without any difficulty.** If there is any need to have the manipulator extend even further, this is possible by just making a longer pointer or a translational mechanism
- **It can lift 1kg at the maximal extended length of 75cm, but not at the rated current** because of friction in the system. In its current state, it can lift 750g from a standstill at a horizontal position while at rated current
- **All original features have been preserved.** The lifting handle is the only one that has been changed a little, meaning it has been positioned a little higher than original. It is still possible to lift the Loomo's entire weight from it and the button functions as normal. Installing/uninstalling the manipulators and shoulders does not require unplugging the cable
- **It is still possible to ride the Loomo with the manipulators installed** because of the thin belt solution.
- **It can balance on its own,** but if there is weight at the end of the manipulator, Loomo's balancing ability is very poor.
- Since all original parts have been preserved, **Loomo can be put back to its original state without any mark of ever having the manipulators installed**.
- With access to all source files and the end effector being as flexible as it is, **developing further should be no issue**
- The manipulator resembles a human-like arm, and fits the Loomo's colors and design

9.3 Stress comparison FEA and hand calculations

The results from calculating the von Mises stress by hand are shown in table 4.11. These are compared with the von Mises stress from the FEA. The convergence study as shown in figure 5.17 proves that the value converges towards a specific value. The von Mises stress from FEA is 11.7 MPa as shown in figure C.3. Differences between FEA and calculation can be caused by that in calculation there is assumed that the manipulator is a slender rod while in FEA it takes the whole geometry for the computations.

Table 9.1: Comparison between von Mises from Hand calculated and FEA

	FEA	Hand calculations
$\sigma_{vonMises}$	11.7 MPa	9.07 MPa

Table 9.2: Yield Strength and safety factor for torsion axle S235 steel

Yield Strength	Safety factor
235 MPa	20

With the biggest stress, the von Mises from FEA Tab. 9.1, the safety factor on the steel rod is about 20, since S235 is a ductile material Yield strength is used to calculate the safety factor, Tab. 9.2.

9.4 Dynamic analysis

With the manipulator mounted on, the dynamic behavior could be tested, resulting in the

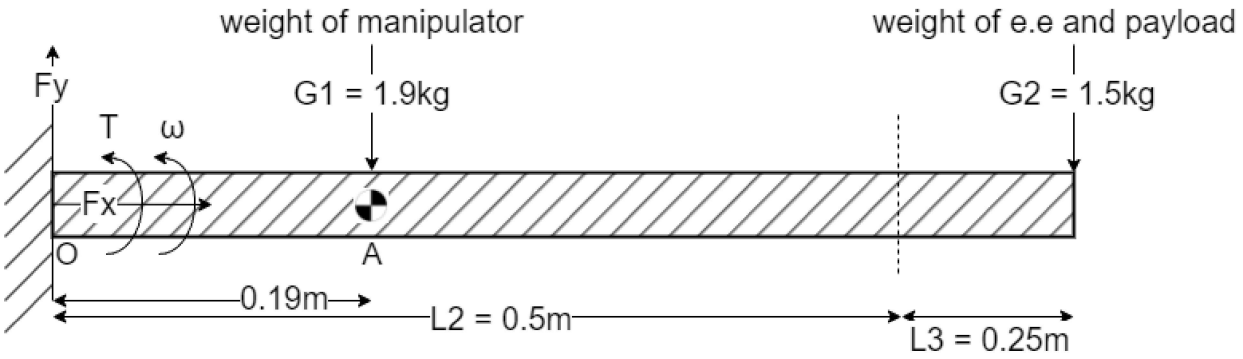


Figure 9.2: Dynamic load analysis

When testing the final prototype, a payload of 1 kg was used to see if the manipulator could lift it. The manipulator was able to lift the payload past the point where the manipulator is the longest. Since this is a dynamic problem there is done a dynamic analysis of the system is shown in Fig. 9.2.

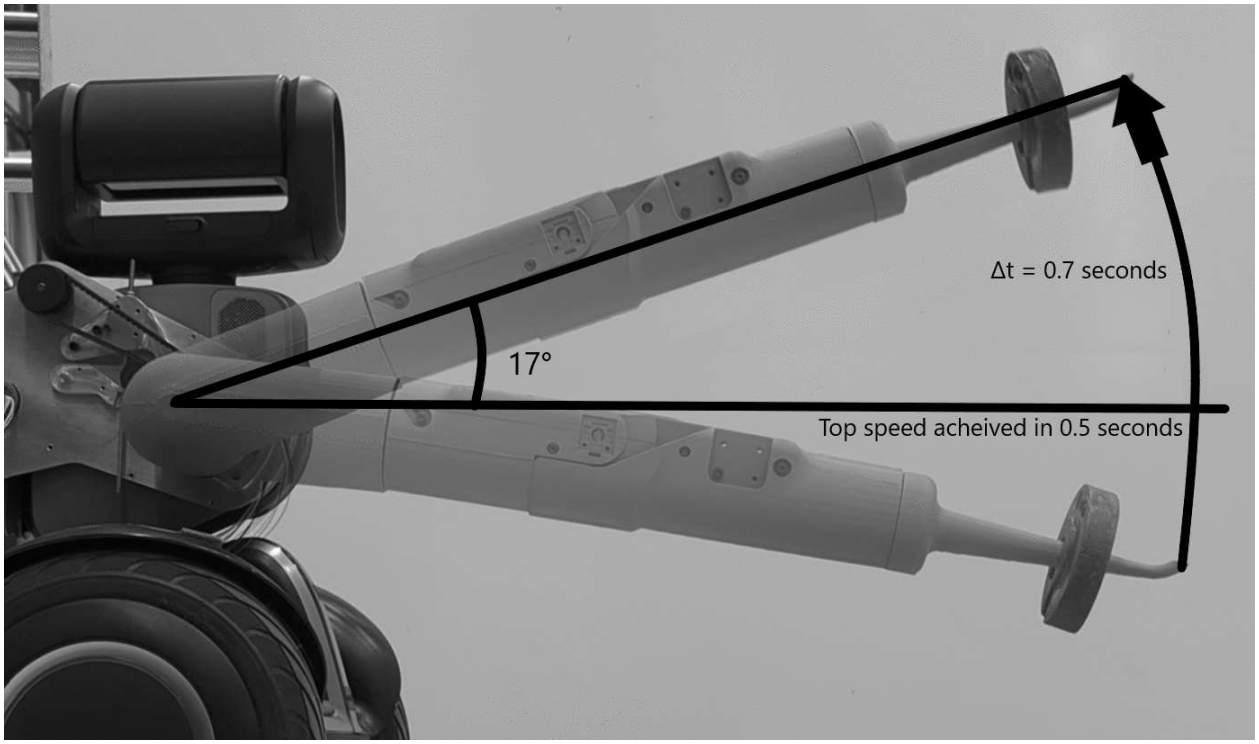


Figure 9.3: Manipulator when lifting 1kg, speed and acceleration approximation

When the supply was set to 1.8 A and 24 V, the angular velocity and acceleration were approximated by studying the video provided in this thesis, see Fig. 9.3. The result from this can be seen in table 9.3. By utilizing the equations for dynamics Eq. (3.5) and kinematics Eq. (3.7) the approximate torque the motor needs to lift this payload is shown in table 9.4.

Table 9.3: Speed and acceleration results

time in motion	0.5 s
acceleration time	0.7 s
angle	17°
ω_{max}	0.42 $\frac{rad}{s^2}$
α_{max}	0.84 $\frac{rad}{s^2}$
$\bar{I}$	0.86 $kg.m^2$

Table 9.4: Results from dynamic calculation

a_{Gx}	-0.034 $\frac{m}{s^2}$
a_{Gy}	0.16 $\frac{m}{s^2}$
$F1$	0.065 N
$F2$	33.65 N
T_D	18.12 Nm
Payload$_D$	0.95 kgf

Payload$_D$ from Tab. 9.4 means that with the rated current of 1A, assuming a friction-less system, the manipulator could handle the 1.45kg dynamically from a straight arm position. In reality, to lift 1 kg the power supply had to deliver 1.8 A which is over the rated current for the specific motor, but still less than the stall current which is 3 A. This means there is significant friction in the system. Most likely in the belt mechanism, since some displacement was observed here.

9.5 Robotics

Table 9.5: DH table for Left manipulator

RotZ	TransZ	TransX	RotX
$-q_1 - 45$	L_1	0	-90
$-q_2$	$L_2 + L_3$	0	90
$-q_3 + 30$	0	0	90
$(-q_4)$	$L_4(+L_5)$	0	0

Table 9.6: DH table for Right manipulator

RotZ	TransZ	TransX	RotX
$q_1 + 45$	L_1	0	90
q_2	$L_2 + L_3$	0	-90
$q_3 - 30$	0	0	-90
(q_4)	$L_4(+L_5)$	0	0

Table 9.7: Variables in DH table

Angle	q_1	q_2	q_3	q_4
Description	Shoulder rotation	Upperarm torsion	Elbow	Forearm torsion
Range	$-15° \sim 165°$	$(-135°) \sim 135°$	$(-150°) \sim 0°$	$(-\infty°) \sim \infty°$

Table 9.8: Manipulator lengths for DH table

Length	L_1	L_2	L_3	L_4	L_5
Description	Length from pulley wheel to shoulder bend	Length from shoulder to upperarm split	Length from upperarm split to elbow	Length from elbow to mount for end effector	Length of end effector to TCP
Value	75 mm	150 mm	140 mm	210 mm	$\approx$ 200 mm

There was made DH-table when the manipulator is placed in the home position from the final prototype. The DH-table for both manipulators is shown in table 9.5 and 9.6. These can further be utilized to find the Forward kinematics, Jacobian and Inverse kinematics. Forward kinematic show the position and orientation based on the joint angle. While Inverse kinematics gives the joint angles based on a specified position and orientation of e.e. The IK and Jacobian are necessary to be able to control the manipulator.

Chapter 10

Discussions

The Segway Loomo is a robot designed to be a helper and a companion, at home, at work, and always with humans. It has a round design to look friendly and even has the capability to be a form of transportation. All of these things had to be taken into account when deciding for the design of the manipulator. Along with its purpose, to be able to hit elevator buttons, the manipulators had to be safe to stand near and strong enough to be developed further. Possibly lifting small objects and having mounted end effectors to do various things, and of course not break if something minor happened. The first parts of the manipulator had to be made to fit the existing holes in the Loomo. They were placed with uneven spacing which affected the design in the beginning. Both the parts and drawings that connect to the Loomo, therefore, have what looks to be highly illogical hole spacing. With the original drawings for the Loomo, it could be possible to refer in a better way since they probably have a pattern.

Throughout the remaining parts, measurements have rounder values to ease readability and recreation. Something problematic about the parts is the way they are made dimensionless to a large degree. They are referenced heavily and hard to make readable drawings of. In other words, they are easy to change values of, and the geometry remains. So if an adjustment needs to be made on some parts, this is easier. Unfortunately, the final drawings suffer from this. Parts that need to do many things at once; holding sensors, wrapping other parts, and making room for cables, have a geometry that is impossible to make meaningful drawings of. Some of these had to be machined, and they should have been designed so it was easier to recreate. All the other parts however are parts that are designed as covers and since they will never have any structural purpose, will never be machined. So whether or not this is a problem in practice remains a question.

Spending much time on designing and geometry means a better product, and there are many things that show the positives of this: It is possible to take off all covers without needing to disassemble the arm because the design process was as long as it was. There is a cable gate that goes through the entire arm, which makes it possible to wire and unwire components with the manipulator mounted on the robot. There are no hooks or loops that must be looped through to get cables out, only one to get it from the shoulder to the actual manipulator. The limited space made it hard to have this modularity with the structural components as well, but half of the arm is dismountable without interfering with any other part. The same would be encouraged with the design of the end effector and further design as a whole.

10.1 Electrical and wiring

When choosing how to connect all components together, a BUS seemed logical since it meant fewer cables. As many components as possible were placed in the back where it would be possible to focus on organizing components, cables, and systems. Expanding the capabilities of the manipulator is therefore also rather simple. Adding more actuators, beyond the capability to have two more than installed already, is as simple as adding cables, and placing the components in the much roomier

back. If the manipulator would be placed remotely, off the Loomo, redesigning the casing for the shoulder to hold the components on there instead is a possibility. Although because of the design, the manipulator should always be mounted on a vertical surface, and never move too far behind the shoulder joint without redesigning the belt-tightening mechanism. It is not designed to have a full range of motion.
In the manipulator, there is no part that is designed to fail first in case of overload. This would require excessive testing and is outside of the focus of this study.

10.2 Machining

There has been a steep learning curve about how to best design good and machinable parts. If the project were to be done again, and for all future projects, drawing simplicity would be a much higher priority. More right angles, more hubs to transfer force through, and better clearance. The parts that were chosen to be in aluminum needed the stiffness of metal, and the axle extension in the torsion of the upper arm had to be steel because there was no aluminum rod thick enough, otherwise it could have been made in aluminium.
The machined parts of the manipulator, meaning brackets, axles and such, was necessary for the design to work.

10.3 Use of the robot with manipulators

With the manipulators mounted on the robot, before getting on, one must always put the arms in the home position and check the surroundings. There is extra risk in driving the Loomo, especially if driving fast, with the manipulators mounted on it. It has not yet been tested if the balancing, speed limit, and other driving assistance are fully functional.
If the Loomo is tilting forwards or backward when in robot mode, and the driver wants to get on, one must let the Loomo set itself straight before leaning weight on it. This is since when in SBV mode, the balancing point is unaffected by external loads.

10.4 Solidworks

Parts that are not structurally important did not get it's own drawing. The logic being that to construct the arm, only machining of structural components will ever be needed. The rest was 3D printed.
3D printed parts was printed in a plastic called PLA, although there exists information about this material, it is not accurate for 3D printed parts. This is since the plastic have imperfections, in its layers and behave like wood in many ways. Strongest with its fibers, or in the case of 3D print, with its layers. Not as strong transverse of them. In the FEM analysis a custom defined material was defined to roughly resemble the strength, weight and other characteristics of the plastic.

10.5 Displacement due to payload

When loading the manipulator with the rated max payload, the arm experiences a displacement. However due to having IMUs in every link, this can easily be compensated for.

10.6 Dynamic problem actual payload

A robot manipulator is a dynamic system. Due to this, the motor goes over the rated current to lift a payload of 1kg from the position where it requires the highest moment to lift this payload. This is when the manipulator is pointing straight out. To overcome this the motor has to draw a current of 1.8A for a small amount of time and then it goes back to nominal conditions. Even though it can lift 1kg it could result in wear of the motors. This result in that if the motor shall continuously run on rated values, the payload have to be less.

10.7 Modifying motor axele

There was drilled a hole in each of the different motor axles as can be seen in figure 10.1. This was done to get less backlash on the parts connected to the motor axles. Also, the smaller pulley d-bore started to slip too early.

Figure 10.1: Modified motor axle

10.8 What we would do differently

- Go for a naming convention from the start to get a good system for all parts
- Made simpler geometry, at least for parts that were machined
- Used more standard axles
- Modify the design to fit a metal pulley wheel for the motor side since this is a weak spot in the design

Chapter 11

Future work

Seeing as one of the goals of this bachelor's thesis was to make manipulators that could be developed further, this chapter represents what we think is a natural direction to continue development.

11.1 Mechanical design changes and enhancements

From this prototype, there is a place to mount magnetic angular sensors and some IMUs on the manipulator. There is no place to mount some of the components in the manipulator such as the I2C level converter, the multiplexer, and the supply voltage terminal blocks. So for these components, there have to be added mounting possibilities inside the manipulator.

Cable managing could be modeled as part of the 3D printable covers to more easily control the cables going through the manipulator.

11.2 Limit sensors

As there is now it is not implemented any sensors for detecting limit positions in the different joints. This is a crucial component to have on a manipulator for detecting end-positions as a backup if one of the angular sensors fails. This is to get a redundancy so that the manipulator will not crash into itself. There have been considered using either a mechanical limit switch or a proximity sensor. Since a mechanical switch can wear over time, there is more reasonable to use a proximity sensor. This type of sensor is non-contact which means that it relies on a magnet that produces a magnetic field and a sensing device that detects this field. This sensor can be mounted on a fixed location and the magnet is to be mounted on the rotating part.

11.3 Make a backpack for the battery pack, controller, and components

To mount the battery pack and components a backpack has to be developed and mounted on the back of Loomo. This is to have a storage place for the battery pack, Arduino's, motor controllers, current sensor and one of the MPU9250 to be mounted in. The use of a backpack will keep the various components safe from the surroundings, also the Loomo will have a more natural design by hiding the electrical components. Also, a cover for the belt mechanism is important to have made.

11.4 Electrical wiring

The components are not yet wired. To be able to control the actuation of the manipulator all the components have to be connected in a specific way. The connections are shown in Appx. D. This shows which pinout on the Arduino and which supply terminal that is utilized. This solution is proposed to use a voltage branch for 5V and 3.3V both in the backpack and the manipulator. This means there will only be one of each voltage cable running up into the arm.

11.5 Calibrate magnet encoders

When mounting the manipulator for the first time, or taking off the magnet and installing it again, the direction of the magnet is unknown. Therefore a calibrating sequence or separate script to find the values for the home position is a crucial part of making these encoders work.

11.6 Assemble the battery pack

To utilize the battery pack, a BMS must be added. This is to safely supply components and recharge the pack. Also adding a battery level indicator to get an indication of the available battery capacity would lift the user experience.

11.7 Inverse kinematics and control system

To be able to control the position of the TCP from coordinates, Inverse kinematics must be found to get angle values. The DH tables that form a basis for this are given in results, Fig. 9.5, 9.6. With this the arduino can PID-regulate the angle values.

The Jacobian matrix of the system, and then the Inverse Jacobian matrix make it possible to get angular speeds from cartesian velocities in the TCP. With these two one can cascade-regulate the current to each motor and do path planning. Then all that is needed from the user is to input the wished movement or position and the Loomo with its arduinos will calculate it. Optionally one could use its cameras, or install another one, to get a point in space to keep the TCP at. There is an infinite number of possibilities.

Chapter 12

Conclusions

The design and implementation of a manipulator for Loomo open up new possibilities and applications it can be used for, such as picking- and placing of objects and pushing elevator buttons. The main goal is to make Loomo able to push buttons. This is to make it usable for guiding people around the university campus. This is done by keeping all the functionality without blocking sensors, the lifting handle at the back, and still being able to ride Loomo.

The final result of the prototype where Loomo now got manipulators, and is still possible to ride on is something that was not known if was possible at the start. In addition to this, it can also lift a payload of 0.75 kg on an arm that is long enough to reach up to the buttons in the elevator it will use in the future, which was the original goal. The mechanical design of the manipulator is strong enough for most of the applications the Loomo could ever be set to perform while maintaining a balanced robot in the process. Also, aesthetically, the design fits well with the overall design of the Loomo.

The implementation of a manipulator has transformed Loomo into a more versatile SBV that can be used in more applications. With the ability to lift 0.75 kg and reach elevator buttons, with equipped manipulator Loomo can perform a variety of tasks that were previously impossible.

The successful implementation of the robot manipulator illustrates the potential for further enhancements. It has proven that the integration of complex design requirements for a manipulator into the Loomo base structure is feasible. This opens up possibilities for future upgrades and additions to increase functionality.