University of Twente

Faculty of Engineering Technology

The Design Process of a Biorobotic arm for DMD patients

Group 2

Authors
Felix Tieleman S2144182
Jan Haido S2174723
Joep Smits S2107910
Henk Jekel S1802178
Daniel Ding S2117045

Student Assistant
Jorn Jansen

June 27, 2024

Contents

Chapter 1

Abstract

This report describes the design and realization of a robot arm with EMG based control. The robot arm, also referred to as the plant, is a serial robot, powered by two DC motors. The program of the robot has four states: The first two states move each link of the robot arm to the homed configuration, the third state calibrates the EMG signals and in the fourth state the mouse is ready to use. The EMG signals serve as a control interface between the user and the robot. The main system is able to move in any direction in a two dimensional plane on a desk. This is accomplished by a combination of the Jacobian transpose method, effectively a cat-and-mouse game between the setpoint and the end effector (mouse) and a function in which the flexure-effort is translated in a faster or slower moving reference point. In this text, reference point and setpoint are used interchangeably. With the implementation of the Jacobian transpose method, the end effector can move (in a straight line) left, right, up and down. The flexure-effort function makes sure that any combination of these movements with any desired ratio is possible. The setpoint space was limited to be a subspace of the (physical) work space of the robot to reduce the risk of the robot sweeping through its work space limits and possibly damaging the robot or hurting the user.

Looking at the characteristics of the plant, it is not an LTI (Linear Time Invariant) system, since there is static friction between the mouse holder and the desk and in the joints of the robot. The integrator component of the PID controller compensates for this, by increasing the motor torques until the end effector starts to move, but this will still result in an initial shocking movement of the robot, since static friction is always higher then kinetic friction (at low speeds).

In hindsight, the robot turned out bigger then desired, such that it does not fit on a regular desk. Also, more time should have been spend in SolidWorks to redesign the counterweight mechanism. If this was done adequately, there would not be a need for a support under the second hinge. Besides this, the pulleys should have been made bigger (larger diameter), because now and again the belt skips a tooth.

Also, the robot itself is somewhat redundant, since control of the cursor can also be accomplished by directly connecting the EMGs to the computer.

Chapter 2

Introduction

2.1 Casus

People with Duchenne Muscular Dystrophy (DMD) have a genetic disorder that makes them suffer from progressive muscle degeneration and weakness due to the alterations of a protein called Dystrophin that helps keep muscle cells intact. These people are very bad at performing complex physical tasks.

With ever-improving technology, life expectancy and quality of life has dramatically increased for patients. Nowadays people with Duchenne usually reach their 30s and some even their 40s and 50s. To help people with DMD to have similar life paths to people without DMD there is a big need for independence. This independence can be achieved by using technological solutions for tasks people with DMD cannot perform themselves. In this way, help from caretakers can be minimized.

2.2 Design Target

The sole purpose of this project was to increase the quality of life of people with DMD. Arguably some of the most important standards of quality of life are education, recreation, leisure time and social belonging. With improving technology, there is one simple task that can help people with DMD improve in all these aspects of life, controlling a computer. This will help them with accessing educational platforms, recreational platforms and social media platforms. A computer also gives endless possibilities to spend ones leisure time.

The goal of this project is to design, simulate and realize a robot that can control a computer mouse. It must be able to move, click and drag, left click, right click and double click. It should also be safe to use and safe for its direct environment. Controlling the mouse using the robot arm should be as quick as the average healthy person controlling the mouse without the use of the robot arm.

Chapter 3

Requirements

3.1 Stakeholder Analysis

The stakeholders involved in this project include the patient with DMD, his/her family and friends, the caretakers and the students and/or engineers realizing the project. Obviously the person with DMD is the main stakeholder since the robot will improve their quality of life. This means that on a power-interest grid, this individual would list under high power and interest and its stake should be considered at all times. With the person with DMD becoming more independent, the caretakers, family and friends will have to help less, reducing their workload, making them stakeholders as well.

In this case the project is done for educational purposes. If this robot were to be designed as an actual consumer project, the concerned company would also have a great stake in this project concerning mainly cost and maintenance. However, this stakeholder and its stake will not be considered in this project.

3.2 User Requirements

User Requirements	**Requirement**
Control	An arm to carry and control a mouse for use on a computer.
	The possibility to use the left and right buttons on the mouse.
	The possibility to use the scroll-wheel of the mouse on webpages.
	The possibility to click and drag.
Comfort	The whole design should fit on a regular sized desk.
	The movement of the arm should not interfere with other devices on the desk.
	It should be easy to mount or remove the mouse from the arm.

Table 3.1: User requirements

The user might have different requirements than an engineer might have. The following requirements are ones that could be rooted for from a user perspective. Since there was no interview conducted with a person with DMD, the user requirements were set up using common sense. These requirements are listed in table 3.1.

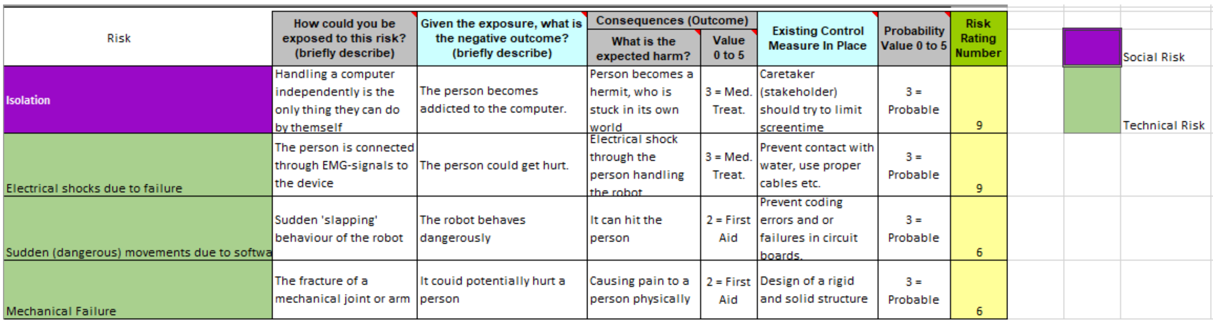
The user requirements given in table 3.1 will be considered during the design and concept phases for design process that follows.

3.3 System specifications

For the engineering side of the perspective, there are requirements regarding the robot and its functioning. These requirements are split into functional and non-functional requirements. The system is a robot that can control a computer mouse. The inputs of the system are the EMG signals of the user, the output of the system is the movement of the cursor on the screen of the monitor. To use the EMG signals as the input for the robot they need to be filtered. EMG signals contain noise which can be removed using filters. After filtering the system can recognize the difference between a contracted muscle and a relaxed muscle. If the system recognizes an activated muscle the system will move the setpoint in a certain direction.

3.4 Social Impact

While designing a product, one should consider the environmental-, economic- and social impact of such product. Innovators and companies have the responsibility to release ethically acceptable innovations. The robot arm that handles a mouse will purely be an asset that replaces the human arm. It replaces an everyday handling. The biggest social impact factor will probably be the safety (like electrical shocks or sudden 'slapping' moves) of the potential robot. The product gives the ability to a disabled person to browse the internet, while the patients slowly deteriorate over time. In addition to this, giving them the ability to perform a task (mostly) independently alleviates a significant amount of work-pressure on the caretakers, as they have to support them 24/7. The thought of being able to do something independently (for a DMD-patient) will bring great joy to both the stakeholder (caretaker) and the patient itself mentally.



Risk	How could you be exposed to this risk? (briefly describe)	Given the exposure, what is the negative outcome? (briefly describe)	Consequences (Outcome)		Existing Control Measure In Place	Probability Value 0 to 5	Risk Rating Number
			What is the expected harm?	Value 0 to 5			
Isolation	Handling a computer independently is the only thing they can do by themself	The person becomes addicted to the computer.	Person becomes a hermit, who is stuck in its own world	3 = Med. Treat.	Caretaker (stakeholder) should try to limit screentime	3 = Probable	9
Electrical shocks due to failure	The person is connected through EMG-signals to the device	The person could get hurt.	Electrical shock through the person handling the robot	3 = Med. Treat.	Prevent contact with water, use proper cables etc.	3 = Probable	9
Sudden (dangerous) movements due to softwa	Sudden 'slapping' behaviour of the robot	The robot behaves dangerously	It can hit the person	2 = First Aid	Prevent coding errors and or failures in circuit boards.	3 = Probable	6
Mechanical Failure	The fracture of a mechanical joint or arm	It couid potentially hurt a person	Causing pain to a person physically	2 = First Aid	Design of a rigid and solid structure	3 = Probable	6

🟪 Social Risk
🟩 Technical Risk

Figure 3.1: A table which describes all risks socially, ethically and technologically

Chapter 4

Concepts

Multiple concepts were created to solve the problem at hand. These concepts should fit the requirements stated in section 4.1. These concepts are then assessed based on how well they fulfil the requirements, after which a final design is chosen.

4.1 Function Definitions

Besides the user requirements in section 3.2, some non-functional and functional, S.M.A.R.T. requirements are also considered in order to eventually test our system with exact answers. This means that these requirements are testable on the final design. These requirements are given in table 4.2 and in table 4.1. These requirements sets the minimum/maximum of the robot of how it should perform ultimately.

Functional Requirements
Be able to move the mouse to any point on the monitor within the given dimensions.
Be able to click the mouse button.
Be able to double click the mouse button such that it can perform more actions.
Be able to click and drag using the mouse, such that it can also perform more actions.
Be able to keep the mouse fixed to the end-effector, such that it won't rotate in order to prevent undesired behaviour.
Be able to remove/replace the mouse ->Detachable.

Table 4.1: Functional requirements

Non-functional requirements
Perform each task(double click, left click, right-click, click and drag) within 5 seconds, such that the delay is not too significant, but still realizable.
Perform each task with 3 mm radius accuracy on screen, with a sensitivity of ∼1:1. The lower the sensitivity, the more accurate movement on the screen.
Move the mouse within a range of a square with dimensions of 300x300 mm, such that the cursor can cover the entire screen.
The robot should weigh 8 kg maximally, to prevent excessive weight.
The robot should carry a weight of 4 kg maximally (including arm), such that the motor can still rotate/translate the system.
The maximum packed size of the robot should not exceed 56x39x42 cm. This is the size of the given box, where the robot should fit in.

Table 4.2: Non-Functional requirements

4.2 Concept Generation

In order to generate concepts, a morphological diagram is made. With this diagram, the best solutions are chosen and combined to produce a whole concept, which should fit within the given requirements.

Motor location	At base	At joint	On arm			
Type of joint	Hinge	Universal	Revolute	Prismatic	Helical	Spherical
Transmission	Gears	Pulley-belt	Toothed belt			
Robot type	Serial	Parallel	Hybrid			
Type of Beam	I-beam	H-beam	Fixed end	Cantilever	T-beam	Square beam

Table 4.3: Morphological diagram

Concept 1: Slotted arm (Vertical)

This concept is focused on simplicity. It contains of 2 arms, a linear guide and a rotational component. The 2 arms, which are in series, can move the cursor up and down, which is represented by the mouse moving forward and backwards. To move the mouse from left to right, the full arm is rotated at the base as point of rotation. Note that this means that the robot rotates as a whole. This system is shown in 4.1.

As **advantages**: this concept is really simple. It consists of little moving parts and few connections. It is low profile and weighs very little.
As **disadvantages**: high pressure and friction will be present on the linear guide due to the force components of the arm not being parallel to the slot. Furthermore, the movement from left to right, which is done by rotation, rotates the physical mouse. This could cause the cursor on the screen to move unpredictable.

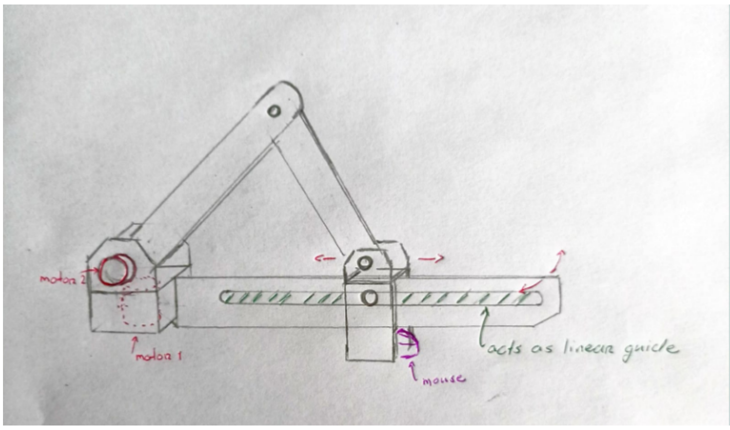


Figure 4.1: Concept one: Vertical slotted arm

Concept 2: Double-Parallelogram model

This concept is based on parallelograms. The first parallelogram is connected to the base of the robot. Due to the geometry of a parallelogram, the end of the arm remains parallel with respect to the base. On this end a triangular support is connected. This support will therefor also be parallel with the base at all times. The second parallelogram is mounted on this triangular support. Using the principles described before, the end of this second parallelogram will remain parallel with the triangular support. These characteristic properties of a parallelogram are used to eliminate the problem describe of concept 1; the mouse rotating. This concept was created to solve that issue. This system in shown in 4.2.

As **advantages**: this concept solves the issue of the mouse rotating with respect to its original position. Furthermore, the concept is sturdy and strong. It also remains level with the table i.e it does not move up or down. This means that, comparing it with concept 1, it has a low profile in that direction, which could be beneficial.
As **disadvantages**: the design in quite intricate and complex: it contains a lot of elements, moving parts and connections. This leads to a high profile design with a high mass.

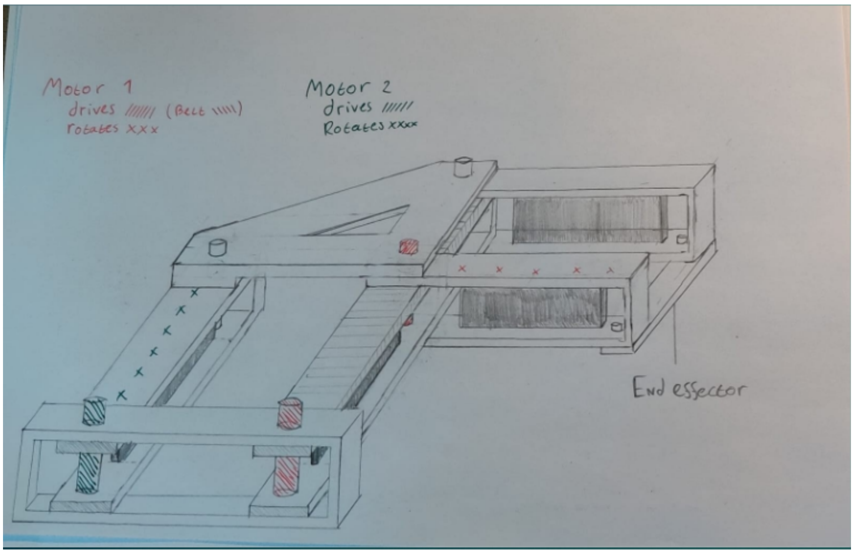


Figure 4.2: Concept two: Double parallelogram

Concept 3: Single-Parallelogram model

This concept is based on a parallelogram as well. It contains a single parallelogram. The configuration of this parallelogram is controlled by the two motors being connected on 2 hinges near the base of the design. By changing the shape of this parallelogram, the end effector, mounted at the very tip of the long arm, is moved. Note that if the mouse is mounted rigidly to this endpoint, the problem of the mouse rotating with respect to its original position will still be present. To solve this, a belt system was designed to counter the rotation of the mouse by rotating it the opposite direction at the same speed. Due to the geometry of this design, this system is passive, meaning that no extra motor is necessary to drive this belt system. This system is shown in 4.3.

As **advantages**: the robot itself is simple and has little elements. There are very little connections and its mass is quite low.
As **disadvantages**: The belt system it requires is a complex system and is probably less reliable in real world. The connections and pulleys would have to be connected very rigidly to avoid play in the system. The design also requires a floating motor, meaning that this robot is to be carried by the arm and can not be mounted to the fixed world. This adds complexity and weight to the design

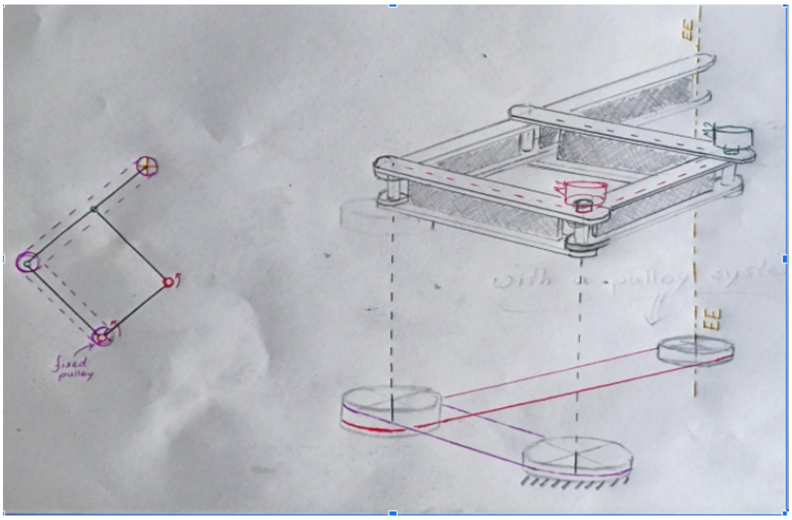


Figure 4.3: Concept three: Single parallelogram

4.3 Concept evaluations

After comparing the three concepts, it has been decided to choose the first concept. The decision was made after the peer review session which made it clear that this concept is the most doable among the different concepts and it provides all the needed functions without any complications. Even though it is quite large in size, the solution is simple. it is the most suitable solution to prevent the rotation of the end-effector and it is within the limits of the user requirements and the functional requirements. There are however a few things we need to address in this design.
It might happen that the first arm starts to twist since, if the mouse does not serve as a support, this creates a torque on the first arm. However, if the mouse holder does support the structure this torque is not there. This does mean that most friction between the mouse holder and the ground needs to be removed by, for example, a piece of fabric on the underside of the holder.
To reflect on the requirements in table 4.1 and 4.2:

- Concept 1 is a serial robot, which comes with the benefit of a large workspace compared to its footprint.
- There are no impeding factors in the design that could result in a non-operational mouse holder.
- Regarding accuracy, a serial robot is not that stiff and errors accumulate over the joints. This means that extra attention needs to be paid to making it a stiff and reliable structure and that a proper controller is applied to the arm.
- The weight of the arm cannot exceed 8 kilograms. This, for a serial robot, especially one that needs to be extra stiff sounds like quite a daunting task. But not one that cannot be overcome.
- Also extra attention should be paid to the axles and the fact that if they bend and are put under stress the motors might not be able to turn.
- requirements regarding the mouse-actions are incorporated in the mouse holder section (section 4.4).

Also, a look at the morphological diagram tells a lot about what kind of robot this is:

- Motors at the base, to reduce the amount of mass in the arms. This also reduces the inertia in the arms.
- Regular hinges, which means that only 1 rotation is allowed. This is done with shafts and bearings.
- Pulley-belt transmission. Because the motors are at the base a transmission system is needed between a motor and the second link. This is done with a timing-belt, because a regular belt can slip, which would result in a second link that cannot be precisely controlled.
- The robot is serial, which means that the second link is attached to the first link.

This does mean however that errors in the motors accumulate.

- I-beams. This is done to stiffen the arm to reduce bending due to gravity, but also keep the mass of the arms to a minimum.

4.4 Mouse holder

As an extra's to the robot-arm, a mouse holder was made in order to hold and control the mouse (Figure 4.4). A servo is installed that communicates with an Arduino in the base as a secondary system, since this system does not necessarily have to connect with the NUCLEO.
The angle of the servo motor is controlled with 2 buttons connected to the Arduino, which means that clicking either button results in a mouse-click. Holding either button makes the servo turn a given angle to click the button and remain at this angle until the button is released. This makes it possible to hold a mouse button to click-and-drag.
To reflect on the requirements in table 4.1 and 4.2:

- The mouse can be clicked, click-and-dragged and double-clicked with the help of a servo motor that is mounted on this effector.
- There is a spring mechanism on this effector that holds the mouse in place and can easily be lifted in order to remove and replace the mouse.
- A downside is that this mouse-holder is only made for a cheap Microsoft mouse. Would the arm ever be released to the market, customers either have to buy a cheap Microsoft mouse with it, or different holders should be made.
- The holder has 2 holes to allow shafts to pass through that are connected to the arm. This allows for assembly onto the arm.
- As mentioned in section 4.3, this holder supports the second link of the arm. This means that the mouse holder touches the ground and creates friction. This friction can be reduced by adding felt to the underside. Also, what can be expected from this is that a PID controller is necessary in order to get the final error to 0, since there is friction involved in a system which is position controlled.

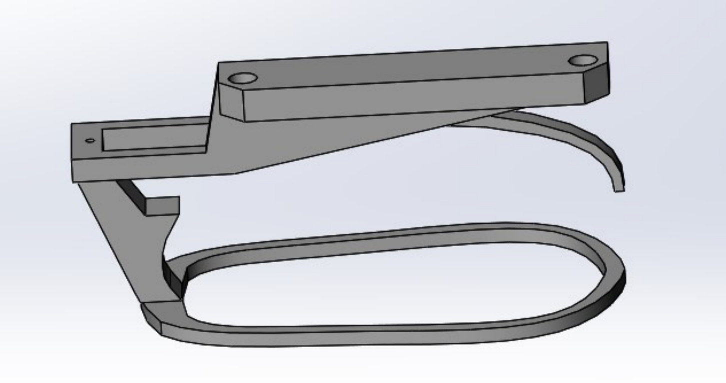


Figure 4.4: The End-Effector that holds the mouse and a servo to control the buttons

Chapter 5

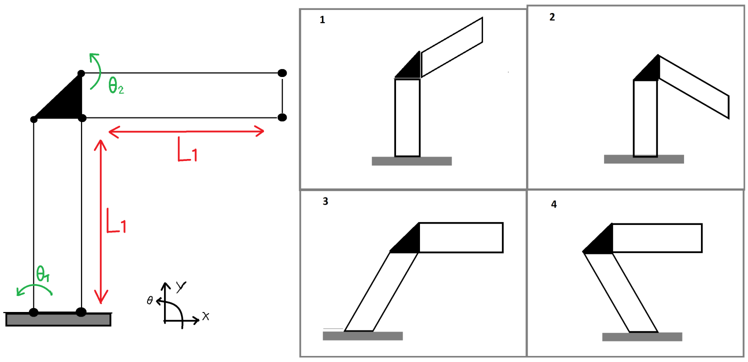
Design

After the creation of concepts, a design decision was made. This design was the one to be worked out fully and built. The design choices and modelling are to be explained to reason why the optimal result was achieved.

5.1 Design Modeling

5.1.1 Kinematic Model

A kinematic model is used to show the basic concept schematically. A coordinate system is shown as well as some dimensions. This model is a simplification of the final design and is used to describe different kinetics and kinematic relations later. The kinematic model is shown in Figure 5.1.



]

Figure 5.1: Kinematic model and possible configurations.

Apart from the kinematic model, the degrees of freedom and possible configurations are also shown in Figure 5.1. This model shows that there are 2 degrees of freedom present in the design.

The amount of degrees of freedom can also be found using the Chebychev–Grübler–Kutzbach's criterion. This criterion finds the amount of degrees of freedom from the amount of links and constraints present in the design.

$$M = 3(N - 1) - C \tag{5.1}$$

In equation 5.1, M represents the mobility, thus the amount of DoF, N represents the amount of bodies and C represents the constraints.

From the kinematic model the following parameters can be found:

$$N = 7 \tag{5.2}$$

$$C = 8 \cdot 2 = 16 \tag{5.3}$$

Evaluating formula 5.1 using these parameters, give a mobility of 2, which suggests that there are 2 degreee of freedom present in the design, which satisfies our expectations.

5.1.2 Dynamic Model and Controller

Each time the robot is switched on, the robot arm needs to be homed. Homing makes sure that the program knows what position the robot is in. A schematic of the homed position is displayed in Figure 5.1 (left). The reference frame is attached to the first motor, but does not rotate with the axle.
The potentiometer on the biorobotics shield is used to bring the robot arm to the homed position. Here, the angle range of each joint is mapped to the range of the potentiometer. In this way a certain angle of the potentiometer results in a certain angle of the joint of the robot arm. Each joint has a PID controller, that makes sure that the setpoint set by the potentiometer is reached fast, without error and with little overshoot and vibration. Respectively P, I and D are responsible. These controllers were designed by empirical testing. Where a step input was given to the motors and the resulting motion was analysed.

The robot arm is able to move in straight lines using EMG signals. There are 4 EMG signals, one to move to the left, one to move to the right, one to go up and one to go down. The control of the robot arm is designed such, that by contracting a muscle, the EMG signal can reach a value above the given threshold. This value is then multiplied by a certain gain such that the speed of the setpoint is dependent on how strongly the patient contracts his/her muscles. One can view this as a cat-and-mouse game, where the cat is the end effector and the mouse the setpoint. The end effector will always try to be where the setpoint is, but will always lag behind. And when contracting different muscles with different intensities simultaneously, any line can be drawn by the end effector. This ensures full capability to operate a computer with the mouse.

To achieve this, inverse kinematics needs to be used. Inverse kinematics is the problem of finding, for a desired end effector position, the corresponding joint angles that will result in the desired configuration. Calculating the inverse kinematics can be problematic due to the following:

- There might be a whole region in the end effector space that cannot be reached since the robot is not able to extend/retract to all points in 2D space. Because it is outside of the work space of the robot arm.
- For a 2 DOF (Degree of Freedom) robot arm, the joint space is 2 dimensional while the end effector space is 3 dimensional. For this reason, the joint space only maps to a certain subset of the end effector space.

- The third problem is that one point in the end effector space can be reached by two points in the joint space, two different joint configurations.

Because of this, generally, there is no closed form solution for inverse kinematics. A clever design for which you are always able to calculate the inverse kinematics, would be one in which the setpoint is always within the workspace of the end effector, in which the orientation of the end effector can be any arbitrary one and where every given setpoint can only be reached with one set of joint coordinates.

For this design, the orientation of the end effector frame can be any arbitrary one, since the design is such that the mouse itself will not rotate due to the double parallelogram. Also, the setpoint space was limited to be a subspace of the workspace of the robot. This was done to reduce the risk of the robot sweeping through its workspace limits and possibly damaging the robot or hurting the user. Now also every point in the setpoint space can be reached by the end effector of the robot. However, since the design is a serial robot with two rotation joints, there will always be two different joint configurations to get to a certain setpoint, unless the setpoint is on the circle with radius equal to the sum of the lengths of the two links of the robot arm.

To solve the problem of inverse kinematics, the Jacobian transpose method is used, also called the force based method. It simulates a force on the end effector towards the setpoint that goes to zero as the setpoint is reached. This can be simulated as a spring that pulls the end effector towards the setpoint, see Equation 5.4.

$$F_s = k\left({}^{0}p_{ref} - {}^{0}p_{ee} \right) \tag{5.4}$$

Here, the force of the spring can be calculated using some stiffness k multiplied by the error between the positions of the setpoint and end effector. Brocketts formula, Equation 5.5 and 5.6, describes the forward kinematics to go from a set of joint coordinates to the end-effector coordinates expressed in the reference frame. Knowing the position of the end effector, the error can be calculated and used to determine the spring force.

$$H^0_{ee}(q) = e^{\,{}^{0}\tilde{\hat{T}}^{0}_{1} q1} e^{\,{}^{0}\tilde{\hat{T}}^{1}_{ee} q2} H^0_{ee}(q=0) \tag{5.5}$$

$$H^0_{ee}(q) = \begin{bmatrix} \cos(q_1) & -\sin(q_1) & 0 \\ \sin(q_1) & \cos(q_1) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(q_2) & -\sin(q_2) & 0.63\sin(q_2) \\ \sin(q_2) & \cos(q_2) & 0.63 - 0.63\cos(q_2) \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0.63 \\ 0 & 1 & 0.63 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.6}$$

$$q_1 = \text{motorangle1} \tag{5.7}$$

$$q_2 = \text{motorangle2} - \text{motorangle1} \tag{5.8}$$

Looking at this second equation one can see that q2 depends on motorangle 1 and motorangle 2. This is because motor 2 is positioned in the base. The spring force can be split into an x and y component expressed in the reference frame. Equation 5.9 and Equation 5.10.

$$F_x = k({}^0ref_x - {}^0x_{ee}) \tag{5.9}$$

$$F_y = ({}^0ref_y - {}^0y_{ee}) \tag{5.10}$$

This force can be expressed as a wrench acting on the end effector expressed in the reference frame, Equation 5.11.

$$ {}^0W_{ee} = [\, {}^0x_{ee}F_y - {}^0y_{ee}F_x,\ F_x,\ F_y] \tag{5.11}$$

Equation 5.12 uses this wrench together with the Jacobian of the current situation to find the joint torques that result from the force acting on the end effector. This equation follows from the Twist-Wrench duality. The Jacobian is a map from joint space to end effector space for velocities, which means that it is also a map to go from end effector space to joint space for forces.

$$ {}^0T^0_{ee} = J\dot{q} <=> \tau^T = J^T\, {}^0W^T_{ee} \tag{5.12}$$

To find the joint velocities the joint torque need to be divided by some fiction coefficient b, Equation 5.13.

$$\dot{q}_r = \frac{\tau^T}{b} \tag{5.13}$$

$$\dot{q}_r = \frac{J^T\, {}^0W^T_{ee}}{b} \tag{5.14}$$

Substituting Equation 5.12 into Equation 5.13 results into Equation 5.14. Using Equation 5.14 and given the Jacobian (Equation 5.15), wrench 5.16 and friction parameter b, the joint velocities can be calculated directly. These joint velocities will make the end effector follow the setpoint.

$$J = \begin{bmatrix} 1 & 1 \\ 0 & 0.63\cos(q_1) \\ 0 & 0.63\sin(q_1) \end{bmatrix} \tag{5.15}$$

$$ {}^0W_{ee} = \begin{bmatrix} {}^0x_{ee}F_y - {}^0y_{ee}F_x \\ F_x \\ F_y \end{bmatrix}^T \tag{5.16}$$

Since these joint velocities can not be implemented directly, a fixed step Euler integration was integrated in the code of the robot, Equation 5.17.

$$q_{i+1} = q_i + \dot{q}_r \Delta t \tag{5.17}$$

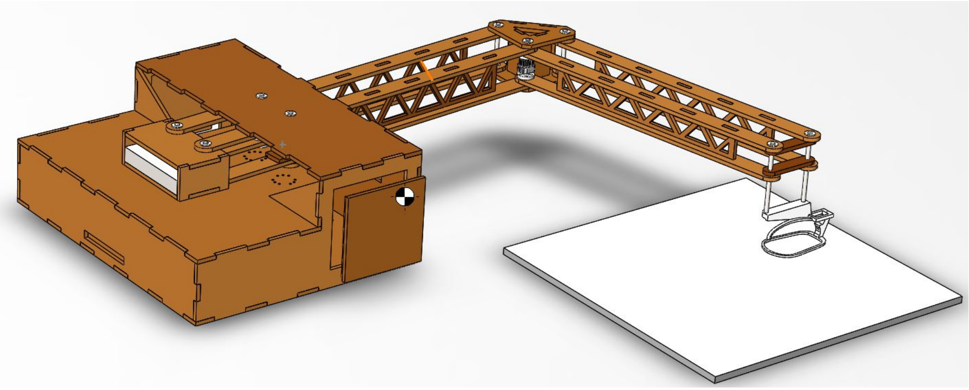
Connecting this to control theory, this is a proportional control architecture where the stiffness k is the proportional gain P. Since the setpoint is to be reached without a final error and with minimal overshoot and vibrations, respectively integral and differential control is added to this proportional control, forming a PID controller.

5.2 Design Details

5.2.1 Mechanical Design

As discussed in section 4.3, the first concept was chosen due to its advantages over the other concepts. This chosen concept was worked out further in 3D CAD using SolidWorks.

The full model is shown in Figure 5.2. This is the detailed design based on the first concept. A base is implemented for stability and contains all electrical components. This base will be mounted on the desk on which the mouse is controlled. This base itself acts as counterweight and no extra weight is to be added to this base to prevent it from tipping. Even though Figure 5.2. shows the center of mass not being under the base, which would cause tipping, the design will not tip over; the motors in the base have not been given any mass in the design as well as the circuit boards not being in the mass calculations. When these are added in the design, this will shift the Center of Mass below the support, since the motors are quite a big portion of the full mass: The whole design (excluding the electrical components) has a mass of 7.1 kilograms. Two doors are added in the side of this base, as well as a slot in the back for wiring and accessibility.



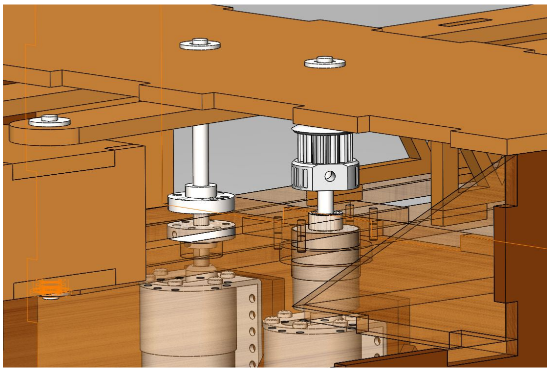
]

Figure 5.2: Full Design

The first set of arms, which are connected to the base and motors, resemble a parallelogram. The explanation on why this is done can be found in section 4.3. A box, which can be filled with counter weights is added on the back of this first set of arms to prevent a moment on the motor shaft. The left most arm is directly driven by the first motor, which results in driving the first set of arms. The right most arm is, due to its geometry also indirectly driven by the first motor. At the very end of this first parallelogram, a triangular support is added. Note that, due to the geometric properties of a parallelogram, this support will never rotate with respect to the base.

A second motor is mounted under the right most arm to drive a pulley with a belt. Note that this second motor is connected to the right most arm with bearings, which means that this arm is in no way affected by the second motor and its rotation. This motor configuration is shown in Figure 5.3.

The second arm is driven by a belt which is connected to a pulley on the base side of the



] Figure 5.3: Motor configuration

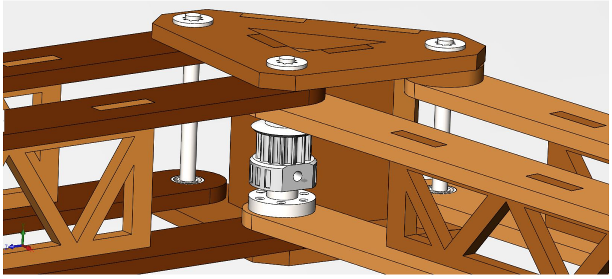


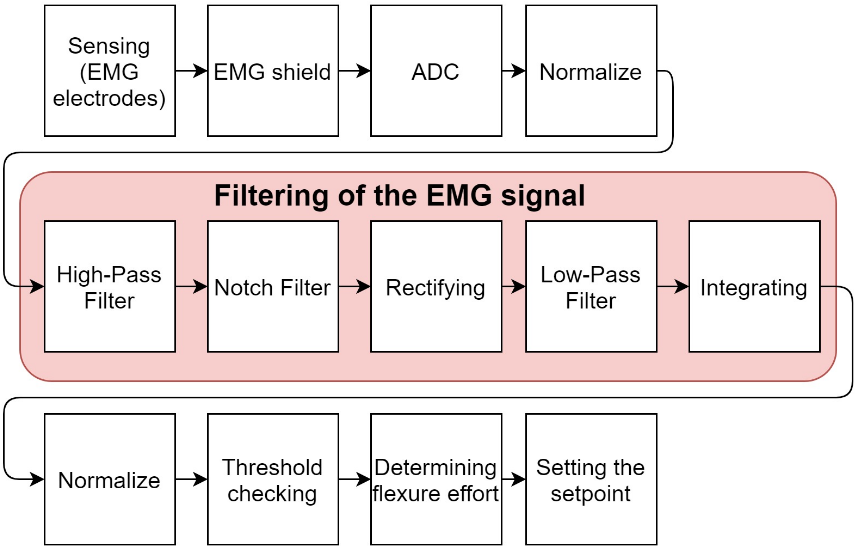
Figure 5.4: Pulley Configuration

robot. A belt connection was chosen to prevent having to mount a heavy motor on the arm itself. Since the belt has a relatively high stiffness, accuracy will be sufficient for our purposes. The configuration of this second arm and pulley system is shown in detail in Figure 5.4. This second set of arms, which also resemble a parallelogram, is connected to this triangular support, which, as explained, never rotates with respect to the base. since this second set of arms resemble a parallelogram, connected to a non rotating body, the very end of this arm set also never rotates with respect to the base. This is very beneficial for the purposes of this robot.

Finally, at the end point the mouse holder is added, to guide the mouse over the chosen surface within the determined range. The mouse will be placed in a 3D printed holder which fits our chosen mouse. If different mouses would be used, a different mount could be attached to support this specific type.
However this possibility is not considered in this project. A servo motor is added on the end effector to left and right click on the mouse. This control loop is set up on a separate board and works independently. How this is set up can be seen in Figure 4.4. To save weight, some cutouts were made in the arms. These cutouts are dimensioned such that the material is still sufficiently strong to hold the loads it is subjected to. These cutouts can be seen in Figure 5.2.

5.2.2 EMG Processing Chain

The robot arm is controlled using EMG signals. Since EMG signals are noisy signals, they first need to be processed in order to use it to control the robot[1]. A general overview of this process is given in Figure 5.5. Here, the processing is describing where the raw EMG signals are translated to booleans to be used in the software. The goal is to agree upon a certain threshold to which the signals envelope is compared, resulting in booleans that are used as input for the controller. These booleans are meant to activate a second function, in which the flexure-effort is translated in a faster or slower moving reference point. However, the noise in raw EMG signals can exceed the threshold and produce booleans that make the motor move. Since noise is unintended and unpredictable, using a raw EMG signal, Figure 5.6A, will lead to an uncontrollable robot arm. For this reason the EMG signal needs to be filtered before it can be used as an input for the motor control. To ensure stability of the processed signal, the signal is filtered using only second order stable filters. Hence, only bi-quadratic filters (see reference [2]) are used. Its coefficients are calculated using Matlab (see reference [3]). The filtering process is displayed starting with an unfiltered raw digitized EMG signal and then adding a filter action in each subsequent figure. The analog to digital conversion was performed using 2^{16} bits.



]

Figure 5.5: Overview of the EMG processing chain

First, the signal is high-passed and a notch filter is applied to it, to remove the 50 Hz electrical noise. This is displayed in Figure 5.6B and 5.6C respectively. The high pass filtering action only removes the 0Hz component of the signal, which is also known as the DC gain. Translated to the time domain, this high pass filtering action removes the constant contribution.

Second, the signal had to be rectified, this is shown in Figure 5.6D. The absolute value of the signal had to be determined such that, in the end, it can be compared with the positive valued threshold. Rectifying is also an essential step before applying a low-pass filter.

Thirdly, a second order low pass filter with a cutoff frequency of 80 Hz is used to filter the signal. A pure EMG signal typically does not have frequencies above 80 Hz. When choosing this cutoff frequency, one needs to keep in mind that the continuous EMG signal is discretized. This discretization happens with a certain sampling rate, that has to be chosen such, that the frequency of 80 Hz (main frequency of an EMG) can still be included in the signal, according to the Nyquist-Shannon sampling theorem. Our sampling rate is equal to the refresh-rate of the micro-controller (500 Hz), this means that signals with a frequency up to 250 Hz can still be reconstructed.

The second order low pass filter enables envelope detection. Methods to achieve this include the moving average method and the root mean square method. However since these methods are computationally very heavy, the low pass filter was designed to fulfil

]

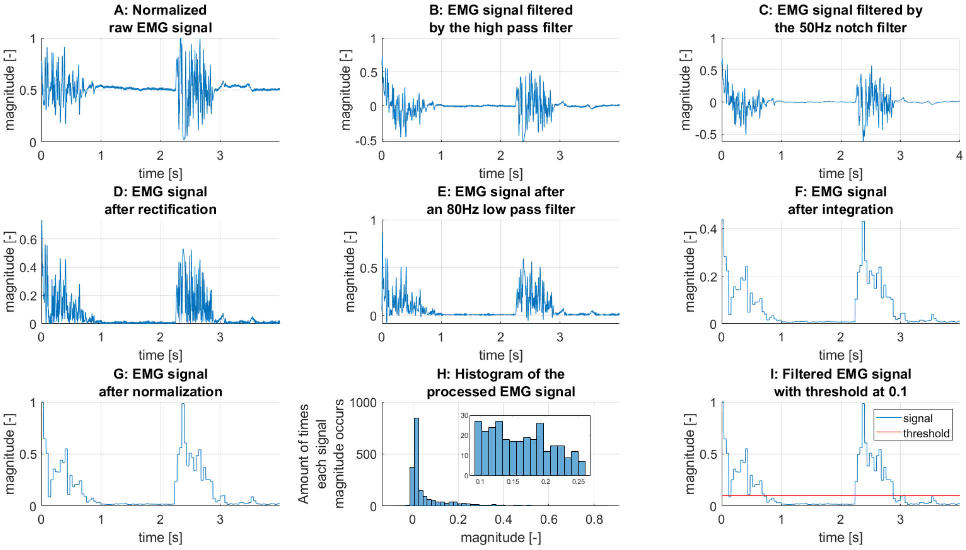


Figure 5.6: EMG processing chain

this task. Filtering out the higher frequency noise makes sure that the signal does not exceed the threshold when the patient is relaxed. Similarly it also makes sure that the signal does not drop below the threshold when the patient is contracting his/her muscles. The result of the second order low pass filter on the EMG signal is displayed in Figure 5.6E.

Fourthly, the signal was integrated. Just like differentiation of the signal amplifies its noise, the integration of the signal damps the (higher frequency) noise. This is displayed in Figure 5.6F.

Before the threshold is applied to the resulting filtered EMG signal, it needs to be normalized, see Figure 5.6G. Normalizing the signal can be done using the acceptable maximum effort concept. In this concept the patient needs to attempt a maximum effort during calibration. The resulting maximum value of the signal will then be stored as the maximum value. Similarly, during calibration, the patient needs to be in a resting position, to obtain the minimum value. Both values are positive.

To determine the threshold of the normalized signal it needs to be determined at which values the EMG signal spends the most time. Looking at Figure 5.6G, this is hard to tell. However, using the histogram of the EMG signal, Figure 5.6H, determining this becomes rather easy. The histogram shows two peaks. The first peak, at 0.001, results from the non-zero rest value. As stated before, applying the high pass filter resulted in a subtraction of the mean value of the signal to obtain a zero rest value. Apparently this rest value is not completely zero. The first peak is very high compared to its width, although it decays much slower to the right then it does to the left. The reason for this is probably hardware related. Each time the patient has contracted his/her muscles, the signal does not stay at a zero value immediately, as can be seen in Figure 5.6G. It shows

Function	Pin
EMG (1 to 4)	A0 up and including A3
Potentiometer	A5
Statebutton	C13
Motor 1 (PWM - Direction)	D5 - D4
Motor 2 (PWM - Direction)	D6 - D7
Encoder 1	D0 - D1 (A and B)
Encoder 2	D11 - D12 (A and B)

Table 5.1: Pin connections on the NUCLEO controller

some kind of resting peak, right after contraction. Keeping this in mind, one can imagine, looking at Figure 5.6H, that without this resting peak, there would be two separate peaks. Here the second peak, would be a perfect example of a normal distribution. This second peak displays the distribution of the amplitudes of the peaks when the patient contracts his/her muscles. The threshold has been determined such that the robot will not react to any remaining noise but also does not delete any low intensity peaks. The red line in the filtered EMG signal of Figure 5.6I shows the threshold of 0.1.

5.2.3 Hardware I/O

This section describes all the hardware needed to build the robot. The hardware included four EMG shield (with a set of electrodes each), a Biorobotics shield, a NUCLEO board (as master) and a motor shield (to control the two motors). The robot has a second, Arduino powered system, working in tandem, but more on this later.

The EMG shields are connected from pin A0 to A3 on the nucleo board. The potentiometer, used in the calibration states, is connected to pin A5.
Table 5.1 shows to which pins the remaining hardware components are connected. Each motor is connected to two digital pins, a pin for the PWM signal and a pin to switch direction.
Encoders are connected to the Nucleo board in order to provide position feedback (each encoder has an A and B encoder).
Each EMG shield comes with a set of three electrodes, of which one is the ground electrode. Since only one ground is needed for the whole robot, the jumper settings on the EMG shields are adjusted such that only EMG 1 is grounded.
The tandem system consists of an Arduino master with 2 buttons and a servo connected to it. The buttons are pulled-down (meaning that in untouched situation the value is 0) and are connected to D1 and D2 (since pushing a button results in a digital input). The servo however has a PWM input, which means that the Arduino pin needs a digital PWM pin. For this purpose D3 is used.

5.2.4 Software Design

State Diagram

In this section an elaboration of the state diagram is presented. The state diagram is displayed in the appendix in Figure A.1. The calibration state is entered by pressing the power button. In this state, first motor 1 is turned on, the EMG control for this motor is turned off and it indicates that its position (1) is not homed. This ”not-homed” offset position is returned. Using this information the first link is calibrated to its initial, desired position. Similarly, link 2 of motor 2 is calibrated. These offsets are saved and used in the run phase, since the encoders are absolute and since they are not resettable when active. When this process is finished, it enters its test state. In this homed position, the motors go off and the EMG on. During this state, the minimum and maximum EMG values are returned. This is also used in the run phase, since the EMG-values will be mapped between 0 and 100, with 0 being the lowest value the EMG gave in the test state, and 100 the highest (maximum effort).
Lastly, when the run function is called, the control and the motors go on again. With control, the action between the EMGs and the motors is meant. During this process the robot is controlled until the power button is physically pressed or when there are no counts with a PWM duty cycle of bigger than 0. This is to ensure that the motor powers down whenever it needs to move somewhere, but cannot reach that position (and might destroy the wooden structure).

Flow Diagram

The flow diagram is added to the appendix in Figure A.2. In this design, there will be 4 different EMG signals. Each signal is generated by a different muscle, creating a movement in x and y direction, either positive or negative. Each sample is stored (along with 2 previous samples) and passed through a 2nd order bi-quad filter, since 2 delays are stored. This filtered output indicates whether the setpoint moves in x or y direction. This passes through the inverse Jacobian to get the reference angles velocities. These values are multiplied by the time step to acquire the reference angles 1 and 2. These angles form the inputs for the system, which passes through two PID controllers (one for each motor) to adjust the stability, response and steady state error to desired values. This then passes through motor 1 and 2, which results in encoder counts, which is then unwrapped to know in an absolute sense, ’where you are’ in space. Here, also the offsets (that are acquired in the calibration phases are used), to know the true position of the robot arm. Ultimately, the correct angles are found such that the end-effector can move towards its desired position.
In addition to this, the true angle is looped back to compute the error relative to the input angle from the sample. Eventually also a PID controller was added between the end-effector position and the reference position, this to ensure that the final position is actually reached. Also an Arduino is added to control the mouse clicks, this is a separate system since it does not need any input from states. This is programmed using C++. The buttons connected to the Arduino are used to read a High/Low (1 or 0) value in order to determine in which direction the servo has to move. This servo is controlled with

a PWM signal.

Timer/Channel definition

In table 5.2, the timers and channels are defined for the encoders and the motors on the NUCLEO board. No further tickers are used, this also means that reading the EMG's is not happening on a separate ticker. However, the test and calibration state are fully finished when it is called again by the state machine. This means that the EMG's do run on a fixed interval, namely 500 Hz. So a separate ticker is not necessary in this case (this also holds for the PID controllers and other fixed time interval functions).

]

Function	**Timer / Channel**
Motor 1	Timer 1 PWM-channel 2
Motor 2	Timer 1 PWM-channel 1
Encoder 1	Timer 4
Encoder 2	Timer 3
The robot/state machine	Timer 2 (500 Hz)

Table 5.2: Objects using a timer

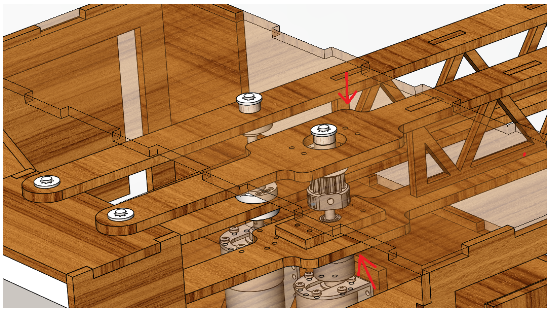
5.3 Design Realization

During the assembly phase of the design, many problems occurred:

- Initially when the components were given, the wood was bend. This probably happened due to the thermal deformation during the laser cut process. In addition to this, some wooden plates were initially engraved instead of laser cut due to a processing error. This could have also caused the deformation.
- Secondly, the hole sizes of certain sections were wrong and had to be drilled again. This was expected, due to the fact that the thickness of the laser itself is not taken into account.
- The ratio of the width of the first arm width relative to the hole diameter (see Figure 5.7) is insufficient. Such that the wood breaks too easily. Therefore the arm is widened in order to strengthen the arm.
- The purchased belt, which drives the hub was too long and thus caused slipping. This is solved by cutting it by half and connected by means of tie rips.
- The hub of the connecting arm section (see the triangular section in Figure 5.4) was initially not driving the second arm. This was solved by adding an additional hub.
- The mouse-holder, see Figure(5.8), holes were too small and had to be drilled additionally. This is due to a small error in the 3D-printing process.
- The counterweight of the arms were redundant, as the wood is too flexible to suffice its function as a counterweight.

- The robot in general is way too large and could potentially move in a way larger end-effector space than the given required space. Therefore the operating conditions were limited.
- After assembly, it was discovered that the moment on the entire arm relative to its base was way too large. Therefore an additional slider support has been added under the triangular section to alleviate this moment.
- It was also discovered that there is a lot of friction, especially between the mouse holder and the desk. To reduce this friction, the underside of the mouse holder was sanded down, and a smooth desk was chosen.

These problems are ultimately solved accordingly, such that the assembly is functioning properly and safely.



] Figure 5.7: The widened arm

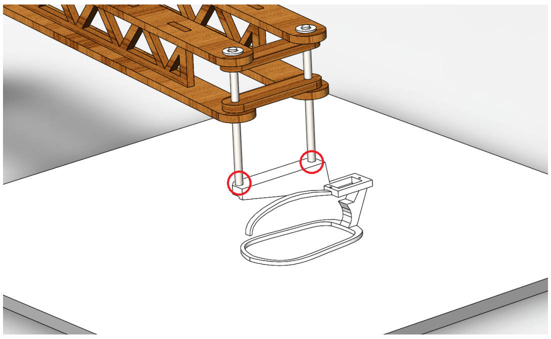


Figure 5.8: Location of drilling holes

5.4 Specific Risk Analysis of Final Realization

The potential risks involved in the final design are mostly evaded by means of taking certain safety measures. For example, the hardware and its motors are all packed inside a box, such that no electrical parts are laid bare. This is done to prevent any potential damage to the hardware and to protect the user from electrical shocks. Secondly, there are no sharp parts or splinters such that it could harm the user. In addition to this, the design movement itself is harmless, as it moves at a rather slow velocity. Also, the movement space for the reference point is limited, which means that, when the machine is situated correctly, it cannot move outside a specified range and potentially harm the user. Also, an anti-windup is added to the controller (limiting the integrating action) so the robot will not show unstable behaviour.

Chapter 6

Evaluation

6.1 Requirement Evaluation

Ultimately, the final robot should perform according to the given requirements during the conceptual phase (see section 4.1). Initially, the robot should behave according to the user requirements (see table 3.1). After physically testing the robot, it was concluded that it satisfies almost *all* the given requirements. The unit moves and clicks properly and does not have any dangerous parts/movements. However, the robot is rather large for the given work space. So it does only not satisfy the mentioned size limitation.
Then, the design satisfies all the functional requirements: It is able to move, click an drag as well as double clicking. Looking at Figure 4.4 the mouse is detachable and keeps it fixed to the end-effector such that it won't rotate.
Lastly, The non-functional requirements are also satisfied. It performs its task within the given time window, within the given work space, and weighs under the given limit (7.3 kg). The robot also fits perfectly within the given dimensions when it is disassembled. The robot performs its task within its given accuracy radius for its given sensitivity. However, the accuracy of this process is dependant on the design of the controller and the sensitivity of the mouse. These factors tune the error margin as well as the response speed of the movement.

6.2 Safety and Ethical Reflection

During the design, it had to be ensured that the robot does not contain any dangerous parts, such as open electrical circuits or sharp parts which could potentially harm the user. In addition loose elements or moving segments had to be designed in a safe manner. As for these potential problems, they are mostly solved. The robot itself is also standing on a platform (e.g. a desk), to ensure extra safety. This to avoid the risk of anyone tripping over the robot.
From a societal perspective, this device allows the patient to perform a task individually. It allows them to enjoy modern technology using technology. The person can then learn more about the world than physically possible. Thus, such ”small” technological advancement can help them greatly mentally as they are allowed to perform a task individually without the constant need of others, which makes them morally aware that they are not always dependent on their caretaker. This develops ethical maturity to the user, as they become aware of their concern for others. So this simplistic product can develop the ethical awareness of the patient and help them to develop as a person.

Chapter 7

Conclusions

7.1 Suitability for Target

The goal of this project was to design a robot for people suffering from a muscle disease (DMD). This robot should allow those patients to perform a task individually. In this case, the patient should be able to move a computer mouse by means of muscle contraction and be able to click on the mouse. Even though the robot is able to perform its tasks within its requirements, the relevance for this type of robot is questionable. Why would one design a robot to merely move a mouse? In addition, it would actually be better to design a robot directly to his or her arm such that they are allowed to do any movement (within limits of the capabilities of the robot), instead of merely clicking a mouse and moving it. So in conclusion, the suitability for the design target of this robot, is relatively speaking, excessive.

7.2 Recommendations

After the completion of the project, there are still some problems left with the product. For example, during the preparation of the live demonstration everything worked, however, as soon as the robot was moved to the presentation room, the EMG's started bugging and did not read anything useful anymore. This is a bug that needs to be removed in order to have a proper product. For now, no culprit was discovered and for further development, time needs to be spend removing this bug.

Besides that, the friction needs to be reduced, the counterweight should be adjusted. Friction within the robot could be reduced with the help of nylon washers (with and without shoulders) between wood on wood connections. An interface between the mouse holder and desk, for example a mouse pad, would also reduce the friction.

As a last recommendation, it might be a better solution to use an extra arm to move the second linkage, instead of a belt, since it was discovered that a belt introduces a lot of play in the system.

Bibliography

[1] R. Shiavi, *Introduction to applied statistical signal analysis: Guide to biomedical and electrical engineering applications*. Elsevier, 2010.

[2] P. Lutus. (). "Biquaddesigner," [Online]. Available: `https://arachnoid.com/BiQuadDesigner/`. accessed: 15.11.2020.

[3] N. Redmon. (). "Biquad filter," [Online]. Available: `https://www.earlevel.com/main/2003/02/28/biquads/`. accessed: 15.10.2020.

Appendix A

Appendix

A.1 State diagram

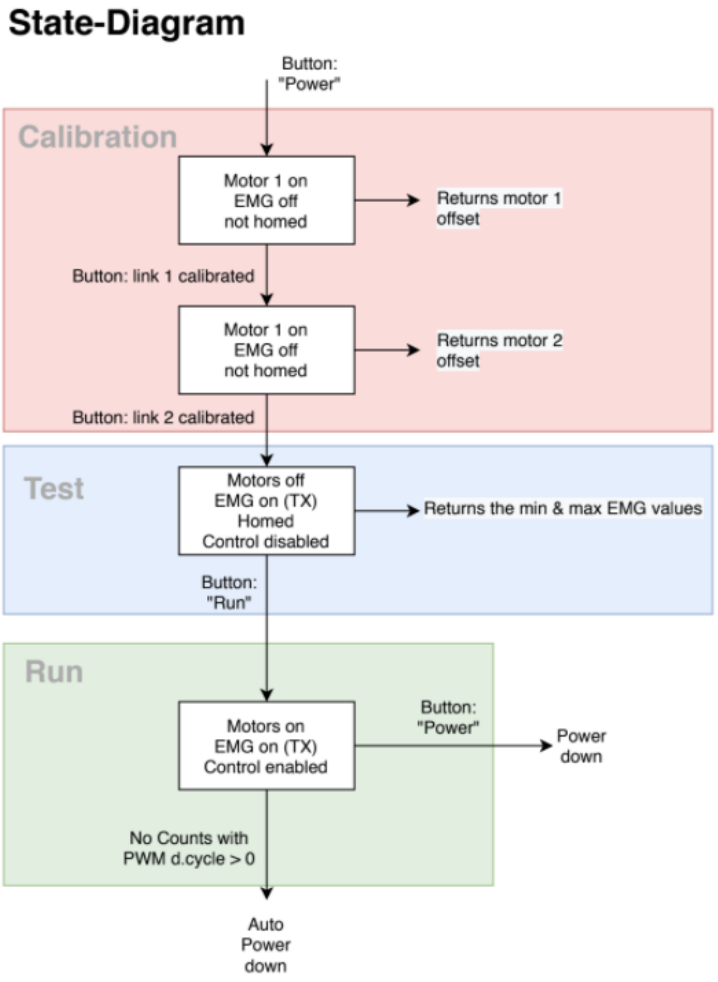


Figure A.1: Overview of the states of the system

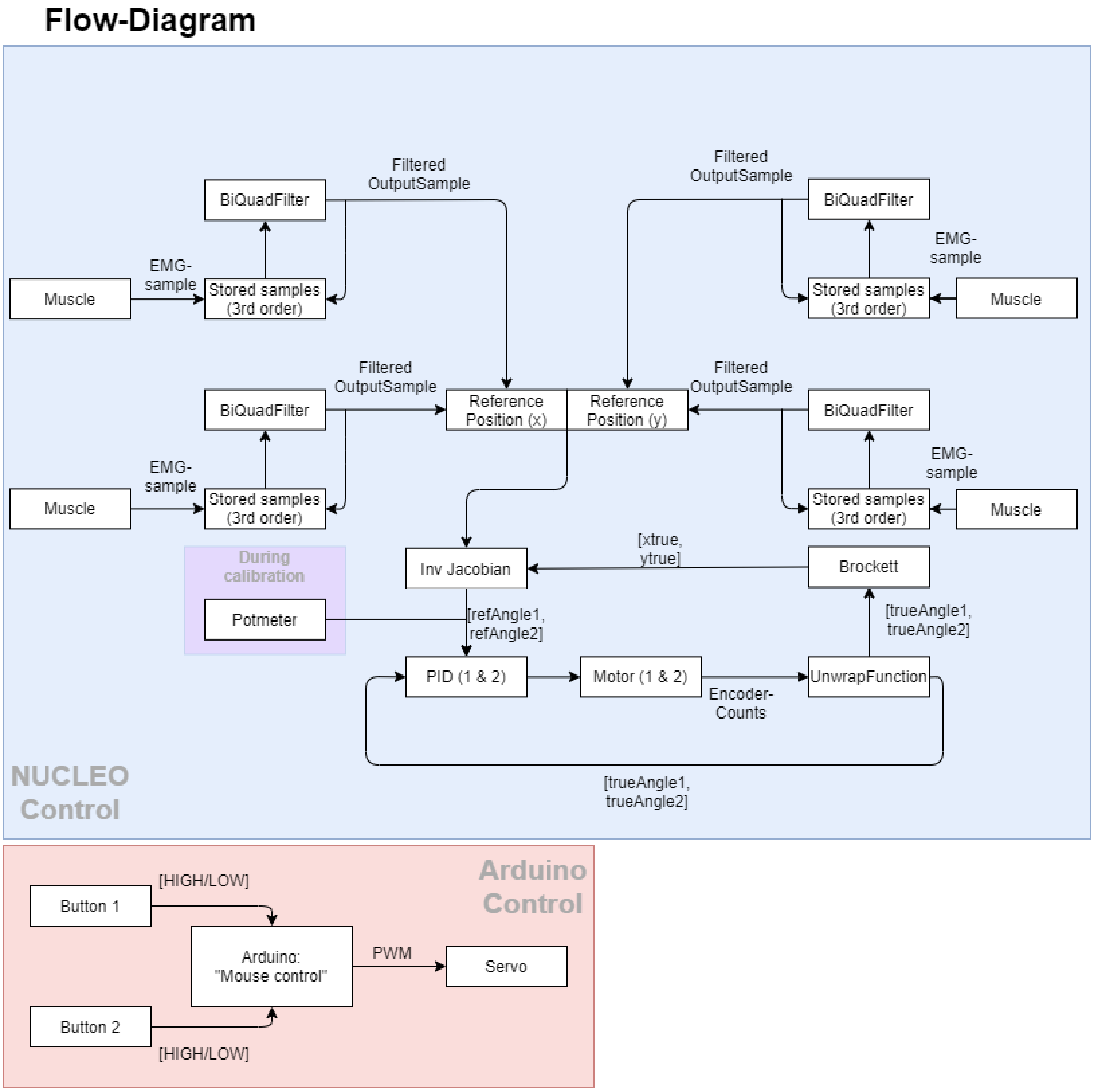


Figure A.2: Overview of the NUCLEO control and the Arduino control