ROBOTICS
AND
MECHATRONICS

HARDWARE FOR DIY AT HOME LAB FOR TELEMANIPULATION CONTROL

F. (Frank) Bosman

BSC ASSIGNMENT

Committee:
dr. ir. D. Dresscher
dr. ir. E. Dertien

July, 2023

041RaM2023
Robotics and Mechatronics
EEMathCS
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

UNIVERSITY OF TWENTE. | TECHMED CENTRE
UNIVERSITY OF TWENTE. | DIGITAL SOCIETY INSTITUTE

Contents

1 Introduction

1.1 Context

Remote control has been around for some time, but it is making a re-surge in the form of remotely controlled robots. With faster connections worldwide, controlling systems from large distances is becoming possible. This enables telerobotic systems to be used in surgical applications, search and rescue, and much more. These systems have the ability to open many doors and shorten the distance between people all over the world.

An important part of this technology is telemanipulation. Telemanipulation is the ability to carry out operations in a remote place with the use of a robotic system (Melchiorri, 2003). This technology is expanding every day with new discoveries, faster remote connections, and better hardware. Spreading awareness and knowledge about it will spark new innovations (Orta Martinez et al., 2020). Avatars.report has made steps to improve and expand the available learning materials (Dresscher, 2023). They have created an online teaching platform where they host a lineup of pen casts to teach about different telemanipulation systems. Currently, they are looking into expanding it with an educational build-at-home telemanipulation setup to learn its users about kinesthetic telemanipulation. There are different ways to feel feedback in telemanipulation, but an intuitive option is kinesthetic because it lets the user feel the forces directly.

1.2 Problem statement

A build-at-home (DIY) telemanipulation setup has been chosen because it has a number of advantages, mainly improved learning and cost. As stated by Han and Black (2011), a physical system that allows users to: implement, experiment with, and feel the actual forces at play, will enhance the learning of users about the system and concepts. The users will gain a more concrete understanding of the system because they will be able to create a more well-rounded mental model. But currently, there does not exist many affordable educational kinesthetic telemanipulation devices that a consumer can buy. This is why Avatars.Report wants to provide a list of materials and a manual on how to assemble them allowing users to cheaply build a system like this themselves. Letting users assemble the system themselves offers multiple advantages, it does not need to be assembled in-house, and no distribution is needed because users buy the parts themselves. These advantages help to keep the cost of the setup as low as possible. Furthermore, the user also gains valuable insight into how the system works mechanically by assembling it themselves. And finally, because the users build the system themselves, they will be more attached to it which encourages them to continue learning. Offering a DIY setup about kinesthetic telemanipulation will help students when learning about this field at an affordable price.

1.3 Contribution

In collaboration with Avatars.Report, Famke and I aim to build an affordable and intuitive educational setup that will teach and encourage users about kinesthetic telemanipulation. This project will be split up into two parts, Famke will design and implement the software and I will focus on the hardware. The software part will consist of writing the code to run the telemanipulation architectures on a microprocessor and creating a user interface for computers where our users can experiment with the system to test their knowledge. The hardware part will include creating an easy-to-assemble hardware kit, which has the needed parts and the 3D print models to combine the parts. Furthermore, the hardware is also responsible for interfacing the hardware on the microprocessor with easy-to-use function calls which abstracts any needed technology such as feedback cycles or low-level hardware management.

1.4 Approach

To develop the educational DIY kit one main research question has been constructed:

> "What are the most important requirements when designing a DIY kinesthetic telemanipulation kit for education?"

This research question is divided into multiple sub-questions:
- "What are the most important concerns when designing a hardware kit for students?"
- "How are already existing systems designed?"

The design of the DIY kit is structured following the guidelines of the Creative Technology design process. There are three phases in this process, the ideation where many feasible ideas are generated, those ideas are then narrowed down in the specification phase. And after a suitable design has been chosen it will be developed into a prototype in the realization phase Mader and Eggink (2014). This process is extended with an analysis before the ideation phase to gain a concrete understanding of the current concerns and state of the art of the technology field, in the analysis the requirements will also be determined. Furthermore, the ideation phase will be incorporated into the analysis as the system design. Because what normally would take place in the ideation for developing such a kit is already determined in the requirements and system design chapters.

2 Background

2.1 Telemanipulation

Simply put telemanipulation is manipulation with feedback (Dresscher, 2023). There are multiple ways to give this feedback to the user, visual, tactile, and kinesthetic. Visual systems are currently the state of the art in telemanipulation, it uses visual feedback to inform the user about the environment on the robot side (Dresscher, 2023). The feedback in advance systems is recorded with stereo cameras and is shown to the user using a VR headset.
Tactile feedback is used to convey feedback through the skin of the user and communicates touch or pressure. The pressure is measured at the robot side, often using pressure sensors, and is presented to the user via hydraulic pressure pads or vibrotactile actuators (vibrations).
Kinesthetic feedback is a more natural way of giving feedback. The user controls the input side of the system and the output side interacts with its environment. Kinesthetic feedback would make it feel "as if human and robot interaction ports are directly connected" (Dresscher, 2023). To achieve this an energetic connection is made between the in and output ports via software. The software uses mathematical models to approximate the energetic connection, and how close it is to a direct connection is also known as the transparency of the system.

2.2 Control architectures

Avatars.Report explains four different control architectures for kinesthetic telemanipulation, the DIY kit should be able to support all of these architectures. A general setup will be given after which the four architectures will be briefly explained. The information about the architectures is provided by the lectures on Avatars.Report (Dresscher, 2023).

2.2.1 General setup

The following architectures are explained using the following general abstract system. It consists of an input for the human and an output which is the robot arm that interacts with the environment. In this setup, they are both one-degree-of-freedom (1DOF) paddles to simplify the example. To control both the in and output a controller is used, the controller can exchange the force and velocity from the human to the robot and back. See Figure 2.1 for a visual representation of the general setup.

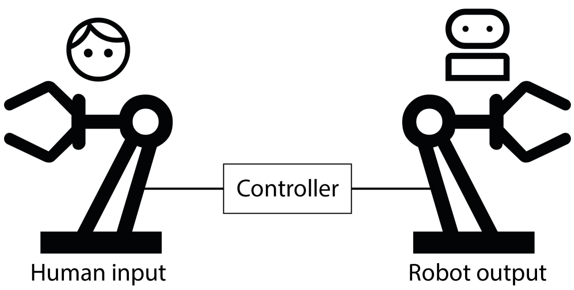


Figure 2.1: Abstract general setup of the system

Both the in and output have 3 power ports: the interaction port, the dynamics port which is determined by the physical system, and the controller port. Each power port has a velocity and a force. In this report, these variables will be denoted with first a capital 'F' or 'V' for force or velocity respectively. And secondly with a pair of subscripts with the first character being a 'h' for human or 'r' for robot. The second subscript character denotes the power port: 'i' for interaction, 'd' for dynamics, and 'c' for the controller.

2.2.2 Position-Computed Force Architecture

The position computed force is the simplest architecture. It models the controller as a spring. To accomplish this it calculates the differences in the positions between the input and output. Then the difference is multiplied by a constant, the spring constant. And afterward is split and returned to both the input and output as a force. The constant determines how stiff the spring is, so higher values mean that it will take more force to move the string whilst lower values mean less force is needed. For telemanipulation, we want a high constant so the spring feels like a more ridged connection. This in turn makes it feel like the input and output are directly and solidly connected, increasing transparency. But there is a limit to how much we can raise the constant because higher values can make the system chaotic.

2.2.3 Position-Measured Force Architecture

The position-measured force architecture is very similar to the position-computed force. The beginning part is the same, it calculates the difference in the position and multiplies it by a constant. But instead of returning the result to the input, it sends the force to the output. Afterward, it measures the force at the output and sends it to the input. This has a major advantage, it removes the influence of the force dynamics on the robot side, F_{rd}. This happens because the force is added to the robot and then measured, this means that all the robot dynamics have been calculated through it.

Another advantage of this system is that the modeled spring has now been moved to the robot side. This means that is no longer part of the communication which in turn makes sure that the spring will not increase in energy due to a time delay. This allows us to increase the spring constant to improve transparency.

2.2.4 Position-Position Architecture

The position-position architecture is based on a feedback cycle. It has a feedback cycle on both the input and output which allows it to set the position to a set point on both sides. The goal of this architecture is to make sure that the set points are the same on both sides. To accomplish this the set points are bound with a model spring to the position of the other side. To gain the optimal connection and better transparency the spring should be rigid which means the constants should be infinite.

2.2.5 4-Channel Architecture

The 4-channel architecture works a bit differently from the other architectures because it is designed on feed-forward control. It sets the new position with the use of the forces on both sides and the position of the other side, but not its own position. It is called the four-channel architecture because of the four channels that cross from one side to the other. This architecture can reach very high transparency, but the dynamics of the systems have to be dialed in well.

2.2.6 Motor control

There are multiple ways to control a motor. The two basic options are velocity control by changing the voltage to the motor, or torque (force) control by changing the current. To reliably change the current a linear current amplifier could be used, to control the voltage a simple H bridge or voltage divider can be used. But Only controlling it in such a manner is not very precise and you can only control the velocity and current as denoted above. To solve this feedback loops can be used, feedback loops are meant to make sure that once you set a certain set point the system will move to that set point and stay there. Position can also be controlled using a feedback loop, but for a feedback loop, we always have to be able to measure the variable we want to control. For position, this would be the angular position which can be measured using an encoder. To create a feedback loop you first define a set point, this will be the position we want to reach. Then the system checks the difference between the set point and the current position and multiples it by a certain constant before instructing the motor to move using the result. This is the simplest feedback loop which is modeled like a spring. To make more complicated feedback loops a mass spring damper could also be used, a feedback loop like this is called a PID controller.

A feedback system can also be used to set the force using the voltage instead of the current. This is accomplished by measuring the force and setting the voltage (velocity) based on it as described above.

3 Analysis

In this section, I will discuss the results of the analysis. In the analysis, I looked at existing literature and did a stakeholder analysis to determine the requirements that a kinesthetic telemanipulation system should have. Afterward, I also explored the existing design space to discover what has already been created so the system design can be built up from there.

3.1 Requirements

To develop a complete product a comprehensive list of requirements should be created. In order to accomplish this a few strategies will be combined. To gain a rough requirement list a stakeholder analysis will be performed. Afterward, these requirements will be refined when determining the scope of the project. Then educational and DIY concerns will be evaluated. Finally, mechanical requirements will be created from the other requirements. At the end, all the requirements will be summarized in an overview. These requirements will form the basis of this project and will be used in the system design and specification to determine how the project will look like.

3.1.1 Stakeholder Analysis

To make sure that the project meets the need of all parties involved, a stakeholder analysis is carried out. First, all stakeholders will be identified, after which they will be prioritized in the power-interest matrix developed by Mendelow (1981). In the end, a list of requirements of each stakeholder will be summarized.

Stakeholder Identification

There are four stakeholders who are interested in this project, they are the client Avatars.Report, the end-users, the University of Twente, and my collaborator.

To start off, the client of my project is Avatars.Report. They are, in their own words “A trend-watching and knowledge authority on robotic avatars”. This means that they are at the forefront of robot avatars and are knowledgeable about the encompassing technology. They share this knowledge through video lectures, pen casts, on their online learning platform. They have given us the assignment to create an educational DIY kit to expand the learning materials on their platform. They would like the new kit to be accessible and affordable.

Following up, the second stakeholder is the end user of the product. Currently, the end users are only robotics and interaction technology students at the University of Twente. But in the future, it will be extended to everyone who is interested in learning about kinesthetic telemanipulation in robots and has some basic technical knowledge. This means that the kit should be accessible and relatively easy to construct.

Thirdly, Another stakeholder is the University of Twente. The graduation project is done at the University for my bachelor Creative Technology, which means that they are responsible for grading the results. Furthermore, this also means that the project development has to adhere to the guidelines of the Creative Technology graduation program.

Finally, the last stakeholder is my collaborator for this project, Famke van den Boom. As discussed in the introduction, Famke focuses on the software to make this kit work and educational, whilst the scope of this project is to focus on the accompanying hardware. This means that the hardware has to be able to support the software with the correct measurement and actuation required.

Power-Interest Matrix

After determining the stakeholders it is insight full to prioritize them using a power-interest matrix. Doing so will allow us to gain more insight into how we should handle the stakeholder. The matrix will sort the stakeholders on two axes, the level of power they hold over this project and their interest in it. The completed power-interest matrix can be found in figure 3.1.

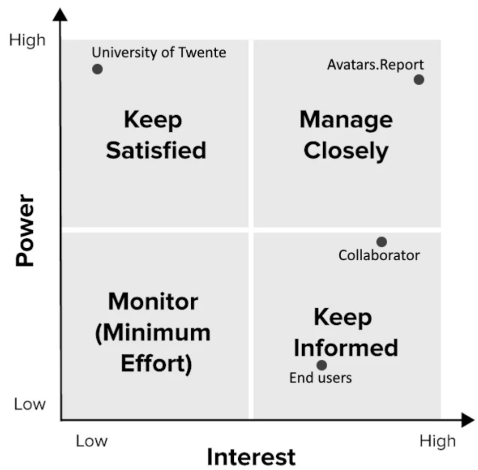


Figure 3.1: Power-Interest Matrix of the stakeholders (Mind Tools Content Team, 2023)

The filled power-interest matrix shows how to handle the stakeholders based on the four quadrants. Starting in the top left, the University of Twente has a lot of power over this project because it will determine if it is good enough for a graduation project. However, it is not very interested in the specifics of this project. This means that should be kept in the loop to successfully complete this project. Secondly, the top right section is the most important section, it houses stakeholders that are very interested and have much power. In this project, only Avatars.Report is in this section. This means that they should be managed closely because they have a high impact on decision-making and are interested in the development of the project. The final filled section is the 'keep informed' section. The stakeholders in this section do not have a lot of power in the outcome of the project but are very interested in it. This means that they can be invaluable when making decisions or thinking of new concepts. This means that they should be kept in the loop. The end users and the collaborator are in this section. The collaborator is a bit different from the end user because she does have quite a bit of power in the decision-making. This is because the hardware should be able to support the software of the system.

Stakeholder Requirements

From the stakeholder analysis, multiple requirements can be determined.

Avatars.Report and end users
The end users and the client can be combined because Avatars.Report want to target the end users. This makes sure that they have the same requirements.

Requirement 1. The DIY kit must be affordable.
Requirement 2. The DIY kit must be easy to assemble.
Requirement 3. The DIY kit must be intuitive to use.
Requirement 4. The DIY kit must be an educational learning tool.
Requirement 5. The DIY kit should be robust and have measurements in place to stop it from completely breaking itself if the user makes a mistake in the user interface.

Collaberator, Famke

Requirement 6. The hardware has to be able to run the software.
Requirement 7. The hardware has to support the kinesthetic architectures developed on the software, which will be elaborated on in Section 3.1.2.

University of Twente

Requirement 8. The development of this project has to comply with the guidelines of the Creative Technology graduation program.
Requirement 9. The project has to be completed in the given time.

3.1.2 Scope Requirements

As stated in the requirements of the collaborator, "the kit should support the kinesthetic architectures developed on the software". However, there exist many different architectures with some needing different in and output controls than others. To limit the scope and connect better to the learning material of avatar.report, the architectures will be limited to the four that are explained in their video lectures. These are position-computed force, position-measured force, position-position, and 4-channel architectures. This extends requirement 7 to "The hardware must support the position-computed force, position-measured force, position-position, and 4-channel architectures". From these architectures, multiple inputs and outputs of the system can be determined. These will be used in the mechanical requirements, Section 3.1.4, to determine which kind of sensors and actuator requirements are needed.

Architecture	**Input**	**Output**
Position-Computed Force	Position of human controller Position of robot controller	Force of human controller Force of robot controller
Position-Measured Force	Position of human controller Position of robot controller Force of the robot interaction	Force of human controller Force of robot controller
Position-Position	Force of human interaction Force of robot interaction	Position of the human interaction Position of the robot interaction
4-Channel	Force of human interaction Force of robot interaction	Velocity of the human interaction Velocity of the root interaction

Table 3.1: An overview of the needed input and output of each control architecture

In Table 3.1 the needed inputs and outputs are given for each architecture. These input and output parameters can be converted into sub-requirements of requirement 7. The new requirement is as follows:

Requirement 7. The hardware must support the position-computed force, position-measured force, position-position, and 4-channel architectures.
7.1. The hardware must be able to set the force, position, and velocity of the DIY kit
7.2. The hardware must be able to measure the position and force of the DIY kit

3.1.3 Educational and Use Requirements

The product will be used by a wide target audience, everyone who is interested in learning about kinesthetic telemanipulation and has basic technical knowledge. This means that the product will be used in different ways, it will be used as a personal project by individuals, in courses together with the lectures of Avatars.Report, or in a completely separate manner. This adds many requirements to the product to ensure that the product can be used by our entire target audience.

Use Considerations

To start off, the users have to order and assemble the parts themselves, this means that the parts need to be accessible for the consumer. Furthermore, these parts should be easy to assemble to make sure that little knowledge about electrical components is needed. This can be accomplished with the use of jumper wires and 3D prints. This combination allows the users to click into place the components whilst being guided by the 3D printed holder. However, care should be taken when designing the 3D prints to ensure that the prints are easy to print and assemble without the user having to shift through many small parts.
The system should also be robust and make sure that the user can not accidentally break the setup when assembling it. This can be achieved with a clear set of step-by-step instructions combined with a sturdy 3D print that will not easily break. Furthermore, the system should also have limiters or breakaway pieces that make sure that the user can not accidentally destroy it when experimenting with it. Limiters in the code could be used to tackle this, but small and easily repairable breakaway parts should also be included in the design. These breakaway parts make sure that even if the user codes the setup themselves the system will not destroy itself.
Finally, making the product affordable has many benefits. It allows them to be used in tutorials with many students (Gillespie et al., 2003). It also encourages people learning the subject to buy one themselves to experiment with it (Orta Martinez et al., 2020; Martinez et al., 2017).

Educational Considerations

As stated above, the users will build the system themselves. This has another benefit beyond cost savings, it will give the users a more in-depth understanding of the mechanical systems in the DIY kit (Yigit Sizlayan and Ankarali, 2019). Furthermore, this more concrete understanding will also help the user in creating a better mental model of the device which will enhance the learning process as well (Han and Black, 2011). To take advantage of this, the system should be easy to code with a fast turnaround time between coding and feeling the system work. Because of this, a microcontroller should be used.

Sumerized Requirements

Summarizing the requirement above creates four new requirements.

Requirement 10. Parts should be accessible to consumers.

Requirement 11. The assembly of the system should be straightforward and should be possible with basic technical knowledge.

Requirement 12. The system should be robust in assembly and experimentation.

Requirement 13. Easy to program with a fast turnaround between changing and feeling the effects.

3.1.4 Mechanical Requirements

The sections above discussed global requirements for the system but this section will explain the specific requirements that the system and its sensor and actuators have to meet. To accomplish this a literature review has been done. However, not much literature exists about affordable telemanipulation systems for educational purposes. Because of this, the literature review looked at haptic systems instead. Haptic systems have similar hardware as telemanipulation systems because they both give feedback to the user on the actions they performed. The search was further limited to only look at haptic DIY systems that give kinesthetic-like feedback using a motor. From this literature review, multiple requirements were discovered.

First high fidelity and low latency are important for a feedback system. A goal often mentioned in robotic teleoperation systems is a 1ms round trip latency (Kundu et al., 2021). This is needed because latency adds a time delay between the human interface and the robot output. Time delay lowers the transparency of the system and can introduce problems for the mathematical models which can lead to an increase in energy in the system. Furthermore, high fidelity is also very important as it refers to low measurement noise, actuation noise, and uncertainty (Yigit Sizlayan and Ankarali, 2019). It is often hard to include low latency and high fidelity in an affordable device. This is why it will be the main challenge of making an educational telemanipulation device.

Riddle Riddle et al. (2020) states that for daily tasks humans exert 1.4 ± 0.6 N to 34.8 ± 1.6 N of force. However, without the thumb, the maximum force used is around 15 newtons for a healthy person. Humans are able to exert more force, but it is not needed for daily activities. The force that will be used on our device should be around the same magnitude. Following this, the force sensor should be able to measure around 20N or 2kg, a bit higher than the maximum force daily used by your fingers.

The motor requirements are more complicated because they are dependent on the design of the system. Other existing systems could be investigated to calculate an indication value for the torque needed. The Hapkit (Orta Martinez et al., 2020) has a lever from the human interface to the transmission. The motor is connected to the transmission. The torque needed at the motor can be calculated using the following formula, Equation 3.1. The 'F' is the force in N at the tip of the lever, 'l' is the length of the lever in meters, and 'g' is the gear ratio.

$$\frac{F \cdot l}{g} \tag{3.1}$$

Taking a force of 15N as discussed with the force sensor, a lever length of 10 cm and a gear ratio of 8 gives us 0.1875 Nm or 18.75 Ncm of torque. The needed torque depends on the final design, but the motor should have around 18.75Ncm of torque. However, this is a lot of torque in a motor, and it will be hard to achieve two of these motors in an affordable kit. Because of this, the motor can be compromised a bit so the requirement halved to 9.375Ncm. This is possible because 15N was the maximum daily use and the average is around 7N.

The angular position sensor should be very precise because it will be used in feedback loops for the architectures. Because of this, a precision of a least 0.1 degrees will be needed. The needed precision can be achieved affordably using a magnetic encoder.

3.1.5 Requirements Overview

The previous sections determined multiple requirements which are needed for this project. Below is a combined list of them.

- Requirement 1. The DIY kit must be affordable.
- Requirement 2. The DIY kit must be easy to assemble.
- Requirement 3. The DIY kit must be intuitive to use.
- Requirement 4. The DIY kit must be an educational learning tool.
- Requirement 5. The DIY kit should be robust and have measurements in place to stop it from completely breaking itself if the user makes a mistake in the user interface.
- Requirement 6. The hardware has to be able to run the software.
- Requirement 7. The hardware must support the position-computed force, position-measured force, position-position, and 4-channel architectures.
  - 7.1. The hardware must be able to set the force, position, and velocity of the DIY kit
  - 7.2. The hardware must be able to measure the position and force of the DIY kit
- Requirement 8. The development of this project has to comply with the guidelines of the Creative Technology graduation program.

Requirement 9. The project has to be completed in the given time.
Requirement 10. Parts should be accessible to consumers.
Requirement 11. The assembly of the system should be straightforward and should be possible with basic technical knowledge.
Requirement 12. The system should be robust in assembly and experimentation.
Requirement 13. Easy to program with a fast turnaround between changing and feeling the effects.
Requirement 14. The latency of the controller should be around 2ms for a round trip.
Requirement 15. The force sensor has to be able to measure at least 15N.
Requirement 16. The motor must have a torque of around 9.375Ncm.
Requirement 17. The angular position sensor must have a precision of at least 0.1 degrees.

3.2 Existing solutions

The systems that will be looked at for this product will all be one-degree-of-freedom machines (1DOF), this means that the input can only move in 1 axis. This restriction is made to lower the price and complexity of the device whilst still being useful for education (Orta Martinez et al., 2020). If needed two 1DOF devices could be chained together to form a singular 2DOF device as (Wong and Okamura, 2005; Martinez et al., 2017).

The systems that will be discussed include the Hapkit and its different versions (Orta Martinez et al., 2020; Martinez et al., 2016); the haptic paddle (Okamura et al., 2002); the iTouch (Gillespie et al., 2003); the ETHZ haptic paddle (Gassert et al., 2013) and the haptic interface (Yigit Sizlayan and Ankarali, 2019). Most of these devices have the same structural outline. they are made up of a 3D-printed body and handle with a motor at the bottom directly driving the input. The handle hinges on the top of the body where an angular position sensor is located to measure the angular position of the handle, Figure 3.2a shows the design of Hapkit 1. The iTouch has a little bit of a different design, it has a horizontal input with the sensor and motor both at the rotation point, Figure 3.2b. It is possible to do this because they use self-made motors with a limited rotation angle and increased torque (Gillespie et al., 2003).

(a) The Hapkit 1.0 (Orta Martinez et al., 2020) **(b)** The iTouch (Gillespie et al., 2003)

Figure 3.2: Design of the hapkit version 1 and the iTouch

3.2.1 Position sensor

Two different kinds of position sensors are used. Most of the mentioned devices use a Hall effect sensor with two permanent magnets attached to the bearing axis. The downside of using a Hall effect sensor this way is that it has to be calibrated every time the product is assembled, but after it has been calibrated it is very reliable (Okamura et al., 2002). The haptic interface uses an optical encoder to measure the position, but because very precise encoders are quite expensive they opted for a moderate 500CPR encoder, (Avago Systems, HEDS-5540) (Yigit Sizlayan and Ankarali, 2019). To compensate for the low resolution they use an electronic gyroscope to measure the angular velocity. Electrical gyroscopes do suffer from noisy output this is why they use an I-State Kalman estimator to estimate the angular position of the handle. The simplest way to get rid of the noise for the angular velocity is to use IIR or FIR filters, but this will introduce delay which makes it unsuitable for this application (Yigit Sizlayan and Ankarali, 2019).

3.2.2 Motor drivers

Just like the sensors, the existing systems use two different kinds of motor drivers. The Hapkit iterations use a dual full-bridge as a motor driver, similar to the off-the-shelve L298 circuit (Orta Martinez et al., 2020). The L298 circuit is also available as an Arduino shield called Arduino motor shield rev3. This shield can control the speed and direction of two dc motors at a time. Using an Arduino shield will make assembly easier because the user does not have to solder anything and only has to plug in the shield. The motors can be connected with wires to the screw terminals. This system does have a downside, it can only directly control the speed of the motor by changing the voltage, but it can not change the current to control the torque directly. It would be possible to create a current control loop by measuring the current and changing the speed accordingly It is able to measure the current through the motor and change the PWM accordingly (Yigit Sizlayan and Ankarali, 2019). A force feedback loop with the force sensor can also be created to tackle this challenge. Other systems solve this problem by using a custom-designed linear current amplifier. Using such an amplifier the torque of the motor can be controlled directly at the cost of a more complex hardware assembly.

3.2.3 Force sensor

The haptic systems that were discussed in the previous chapters do not have a force sensor. But a force sensor is needed for a telemanipulation system as examined in the requirements. The most reliable and affordable force sensors are load cells, they can be purchased cheaply in many different sizes.

3.2.4 Computational platform and interfacing

The discussed systems use different computational platforms for data collection and control. Most systems use microcontrollers to lower the price and complexity of the products (Orta Martinez et al., 2020; Yigit Sizlayan and Ankarali, 2019). Not all systems took this approach, the iTouch uses an analog computer connected to an amplifier to control the system. This approach was chosen because the differential equation that guides the analog computer also guides the simulated environment (Gillespie et al., 2003). Apart from the iTouch, the discussed systems use microcontrollers to control the system.

The systems allowed some degree of interfacing between it and the student. All the discussed systems, apart from the iTouch, collect data about the sensor and motor for the student (Gassert et al., 2013; Yigit Sizlayan and Ankarali, 2019; Orta Martinez et al., 2020; Martinez et al., 2016). This data can be used to learn how the system behaves and if build by hand it can be used to debug problems. The Hapkit goes one step further, it allows students to reprogram the system if they want to. It uses a modified Arduino as its microprocessor, which makes reprogramming easier because it enables students to use the comprehensive Arduino language to write their code (Orta Martinez et al., 2020).

3.3 System Design

In the review of the existing systems, two main structures for the system were found. A vertical design like the Hapkit, see Figure 3.3a and a horizontal design, Figure 3.3b. These designs have been tested in their courses and minor changes have been made, to the material and the transmission, between devices. So a design like this would work well, and the vertical one seemed easier to assemble than the horizontal one.

(a) Vertical design of Hapkit 1.0 (Orta Martinez et al., 2020)

(b) Horizontal design of iTouch (Gillespie et al., 2003)

Figure 3.3: Different orientation of structural designs of existing systems

3.3.1 System structure

The system's structure can be manufactured in multiple ways but the most common and robust DIY methods for low-scale production are laser-cutting and 3D printing. The original Hapkit was made from laser-cut acrylic and high-quality 3D prints. Whilst laser-cutting is a good option for educational courses and distribution, it is not accessible to individuals or high schools. A laser cutter is a big and expensive machinery, to which most people do not have access. Therefore it is better to 3D-print as much as possible. The entire body, paddle, magnet holder, and transmission driver should be printed. When designing these models a cheaper and less accurate printer should be considered. This means fewer overhangs, no sharp corners, and more significant margins. The entire stand with the motor bracket and position sensor holder should be printed in one go, decreasing the complexity of assembly. Furthermore, the paddle with the transmission and a bracket for the bearing and force sensor should also be printed as one design. Some parts like bolts might be needed for mounting the devices and the transmission as well. But apart from these, the design should consist of as few separate parts as possible.

3.3.2 Transmission

Multiple transmissions are used in the different systems, a direct drive Figure 3.4a, a friction/gear drive Figure 3.4b, and a capstan drive Figure 3.4c. These systems have their own advantages and disadvantages. Direct drive is very easy to assemble, but the motor needs a lot of torque and a high resolution to be able to move the paddle to the right spot. A friction or gear drive is also easy to assemble, using either gear teeth in the 3D model or rubber at the bottom of the paddle. But the downside of a friction drive is that it always has to press up against both sides. Cheap 3D printers might not be able to print this precisely. To solve this the height should be adjustable. However, then the system has to be calibrated correctly, which could be a challenge as too little and too much friction will both decrease performance. Capstan drive offers increased performance without the need for calibration but at the cost of a more difficult assembly, the wire has to be tensioned around the motor shaft and to the bottom of the handle. This process can be simplified with the use of a tensioner, by first mounting the cable to one side, the other side can be tensioned with a slider on the side of the handle, like the design of the Hapkit 3. The downside of a capstan drive is that it is not as robust and can not act as a breakaway in case of a problem.

(a) Direct drive **(b)** Friction drive **(c)** Capstan drive

Figure 3.4: Different transmissions of existing systems (Orta Martinez et al., 2020; Gillespie et al., 2003)

4 Specification

The new system can be divided into six parts: the digital interface, the motor driver, the motor, the position sensor, the transmission, and the force sensor. In the analysis, we discussed existing systems and how each of them implemented these parts. But the discussed systems were haptic feedback systems and not telemanipulation systems. As stated in the analysis the hardware for these systems looks very similar, but an additional force sensor is needed to improve transparency. The parts of the system will be discussed separately, but an overview of the needed components can be seen in Figure 4.1.

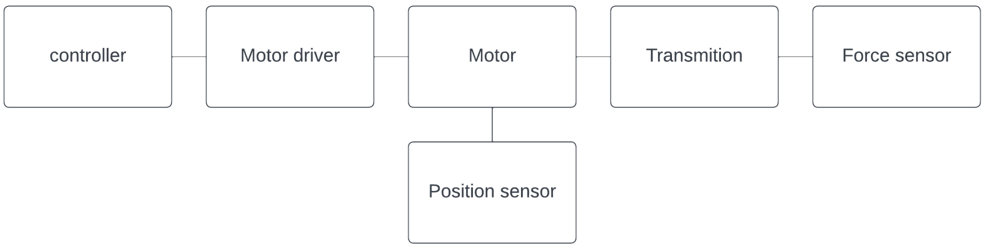


Figure 4.1: Block diagram with the components of the system

4.1 Controller/Computational platform

Starting with the computational platform, it is responsible for controlling the system, processing the inputs, calculating the outputs, and handling the outputs. But as stated in the requirements, it should also be easy to program as well. The Arduino Uno is an excellent candidate because it is familiar to many students and hobbyists. It also uses the comprehensive Arduino language which makes it easy to program with a lot of community libraries and documentation which will improve the development. Furthermore, electrical components can easily be connected because it has 5V header pins. An ESP32 can be used as a backup because of its dual-core and high clock cycle, it can also be programmed with the Arduino language just like the Arduino itself and it is affordable as well. Another option would be to control the system from an external computer or laptop, this would improve the calculation speed and interface for the user as well. But the communication speed between the microcontroller and the computer would be the bottleneck. This bottleneck would mean that we will not meet our requirement of around 1ms for a round trip. One Arduino will be used to control both parts of the systems and it will do the math directly. If it is not fast enough an ESP32 will be used.

4.2 Motor Driver

The second component is the motor driver, the primary choice for the motor driver is a full-H bridge Arduino shield, like the Arduino Motor Shield Rev3. This shield can control two motors independently, it can change the speed and measure the current over the motor. The biggest advance of using this shield is its ease of assembly, the user can just plug the shield on top of the Arduino and screw in the cables to the motor. If an Arduino is not used as the microcontroller a loose L298N could also be used, but it lowers the ease of assembly.

The downside of using a full-H bridge is that it can not control the torque directly, which is needed for a telemanipulation system like this one. To fix this a feedback system can be made in the software that measures the current over the motor or the force on the paddle to adjust the velocity accordingly. A feedback loop will also be made to set the position reliably using the angular position sensor. The Hapkit lineup also uses a technique that resembles this one (Orta Martinez et al., 2020). But if it is found to be unfeasible a linear current amplifier could be used instead. A linear current amplifier does offer direct control of the torque of the motor by modulating the current. But an amplifier like this one is harder to assemble. For this reason, an experiment will be done first to find the feasibility of a full-H bridge, and if it determines that it is unfeasible a linear current amplifier will be used.

4.3 Motor

The motor will be quite simple, all the existing affordable educational haptic systems use a simple brushed dc motor. The specific motor varies a bit, from a Maxon A-Max motor with no cogging torque to a Mabuchi RF-370CA with significant cogging torque. Low cogging torque is preferred to decrease time latency (Orta Martinez et al., 2020). However, the Mabuchi RF-370CA is cheap, readily available, and has worked very well for many systems (Orta Martinez et al., 2020). The motors used by the discussed system do not offer the torque requirement for our project. The RS-775 DC MOTOR does come closer with a torque of 6.3765 Ncm. This motor is more expensive, but still relatively affordable.

4.4 Position Sensor

For the position sensor, a magnetic encoder will be used. magnetic encoders are very reliable, affordable, and easy to use. There exist multiple affordable pre-soldered modules with connector pins to create a simple connection to the Arduino. Most Hall-effect magnetic encoders only have a resolution of 12 per rotation, which is not much if directly attached to the handle. To improve this, magnet position sensors using a bimagnetic magnet can be used like the AS5048 chip. These sensors measure absolute rotation in 360 degrees with high precision, exceeding our requirements.

4.5 Transmission

As discussed in the system design, Section 3.3, there are multiple options for transmission. But the direct drive is not possible with the selected sensor and motor. Furthermore, the friction drive using gears is very suitable for this project. Most of the problems with 3D printing are not as big a problem as they were a few years ago because the printers have improved significantly. Furthermore, its ease of assembly, design, robustness, and possibility for a breakaway gear makes it a suitable option. To maximize the efficiency of the gears, double helical gears are chosen. These types of gears can transmit the force more smoothly because both of the gears have constant contact with each other.

4.6 Force Sensor

The force sensor will be a load cell attached to the handle of the product. This will sense the force on the handle which is needed for most of the telemanipulation architectures. A 5V load cell will be used to work together with the Arduino. And long wires will be used to handle the rotation of the handle. As discussed in the analysis a 2KG load cell should be enough for this project.

4.7 Summary

To summarize the following hardware components will be used, see Table 4.1.

Table 4.1: Table of the hardware components

Component	Part Name	Amount
Controller	Arduino	1
Motor driver	Arduino motor shield rev3	1
Motor	RS-775DC Motor	2
Position senor	AS5048B adapterboard	2
Force sensor	2KG loadcell	2
Loadcell amplifier	HX711 Amplifier	2

5 Realization

The realization will look at the implementation of the system and the design choices that were made. It will go over the design choices of the 3D models and how they work together to form a combined setup. The implementation and calibration of the position and force sensors and finally the position controller will be discussed . In this chapter the implementation of the PID controller explained, after which the tuning will be discussed. But due to time constraints is has not been possible to design and implement a well designed force feedback-controller.

5.1 3D design

The 3D designs will form the entire setup to which the components can be connected. Furthermore, it will form an guide on how to connect the components by only allowing the components to be slotted the correct way. The 3D design is separated into multiple parts: the position sensor holder, the motor holder, the paddle holder, the top paddle, and the bottom paddle.
The paddle holder is combined together with the motor and position sensor holders using a base plate. This is done to make the 3D model easier to print and to make sure the separate designs have a fixed distance from each other. Furthermore, the paddle holder and motor holder are both split into two parts which will be explained in Section 5.1.2 and Section 5.1.3. There will also be a separate gear which is attached to the motor to drive the transmission on the paddle and there will be a small magnet holder on the back of the motor. This means that the entire system consists of seven parts. All of the models will have smoother edges, fillets, to make them easier to print. The subsections below will describe the separate systems in more detail and will show renders of the parts because they are more visible. The printed setup without components can be seen in Figure 5.1.

Figure 5.1: Printed setup, without components

5.1.1 Position sensor holder

The position sensor holder will keep the sensor directly over the center of the motor axis. The sensor will be attached using m2.5 bolts and nuts, preferably nylon or any nonmagnetic bolts and nuts as stated on the data sheet. A magnet holder will be used to keep the magnet at the right spot and will be friction fit on the back axis of the motor. A render without the position sensor can be seen in Figure 5.2. There are small walls inside the hole of the holder to ensure that the user can not accidentally push the magnet into the position sensor chip.

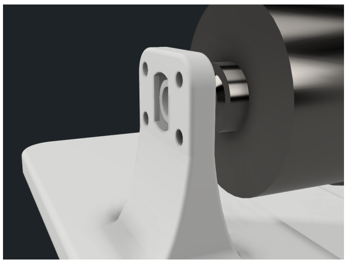


Figure 5.2: A render of the Position sensor- and magnet holder (white) with the motor (black)

5.1.2 Motor holder

The motor holder secures the motor in place. A top and bottom part are clamped together using two M3 bolts, see Figure 5.3. The nuts are inset in the 3D print which allows the users to more easily screw in the bolts without having to hold the nuts. Furthermore, the motor holder also has a small brim in the holder which matches with the metal casing of the motor. This ensures that the motor is positioned at the right distance from the position sensor.

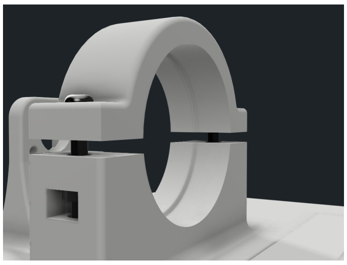


Figure 5.3: A render of the motor holder without the motor

5.1.3 Paddle holder

The paddle holder is the biggest part of the design and it is responsible for ensuring that the paddle is kept in the right position, see Figure 5.4a. It also supports the motor as well. The user should slide the motor through this hole and then fasten it using the motor holder. The height of the paddle holder is adjustable to ensure that the gears of the transmission are kept at the right distance, twice the pitch radius. The top of the paddle holder can be screwed into place using two M3 bolts. It uses the same mechanism to keep the nuts in place as the motor holder, see Figure 5.4b.

(a) Render of the paddle holder **(b)** Closeup render of the top part

5.1.4 Paddle

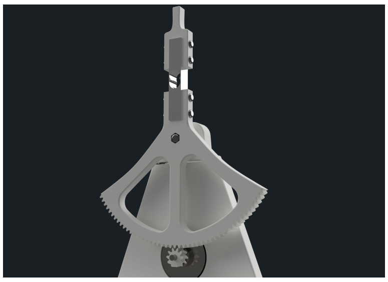
The paddle is the most complicated part as it is the main input of the user, houses the force sensor, and has the transmission to the motor. It consists of three 3D models: the top of the paddle, the main paddle, and a small gear that is connected to the motor shaft, see Figure 5.5a.

The top part of the paddle, Figure 5.5b, is the input of the user. It is attached to the top of the force sensor and the bottom of the force sensor is connected to the main body of the paddle. The user can easily connect the load cell, and force sensor to both sides of the paddle using bolts.

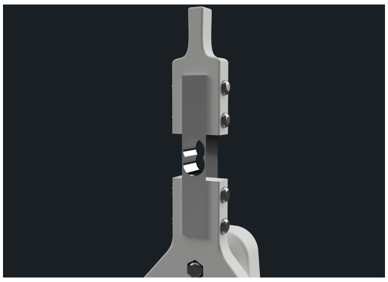
The paddle is connected to the holder at the center hub. Here is a hexagonal cutout in which an M5 nut can be glued. In the paddle holder, there is a hole in the top in which a bearing can be glued. The bearing in the holder and the nut in the paddle can be connected using an m5 bolt. multiple screws around the bearing can be used to fasten it in the correct place.
The first three designs of the paddle and holder used a single bearing in the holder and in the paddle. But this allowed the paddle to move a bit because the bolt didn't completely fill the bearing. To solve this two bearings are used in the paddle holder and a screw is used in the paddle. This also improves the lifetime of the ball bearing, because the ball bearing can not support much twisting force.

The last part of the paddle is the transmission. As discussed in the specification, Section 4.5, double helical gears are used for the transmission, see Figure 5.5c. Multiple gear ratios have been tested, 8, 10, and 13.6. 13.6 was the latest tested gear ratio and is around the maximum that can be easily printed with a 3D printer using PLA. A bigger gear ratio has been chosen because it allows us to get closer to the torque specified in the requirements using the chosen motor.

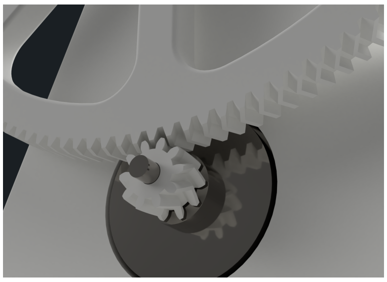
To reduce costs and to keep printing time down a thinner design has been chosen of 10mm. Furthermore, the paddle has a hollow design with only three rounds supporting the gear. These support rods are strong enough to support the paddle and the transmission.



(a) Render of the entire paddle



(b) Closeup render of the force sensor



(c) Closeup render of transmission

5.1.5 Print settings

To ensure that the 3D models are accessible they were tested and printed using Cura (Ulti-Maker, 2023). Cura is a free slicing software for 3D printing designed by Ultimaker. The print settings that have been used are the default for tough PLA, but with support. Furthermore, the engineering preset with a .15mm layer height has been chosen. The filaments used are tough PLA for the final design with breakaway as the support. Normal PLA has been tested for some iterations as well, but normal PLA was harder to work with and printed not as cleanly as tough PLA. The system did work with normal PLA, but tough PLA is advised as it is easier to use.

5.2 Sensors

Two sensor are used in this setup, a position and a force sensor. The previous section showed how these sensors are mounted, this section will discuss how they are calibrated and read. As discussed in the specification, Chapter 4, the Arduino language has been chosen. It is an variant of c++ with a wide range of libraries written by its community (Arduino, 2023).

5.2.1 Position sensor

The position sensor is an AS5048B adapterboard. It used the AS5048 chip to sense the rotation of a bimagnatic magnet. The B means that is used I2C interface instead of the SPI used by the A variant. The reason the B variant was chosen is because it was more available in Europe.

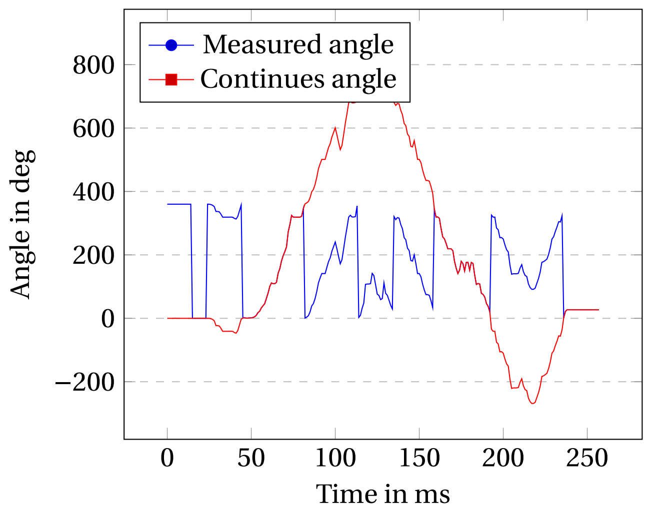
Instead of using interrupts over PWM I2C has been chosen. Because the Arduino Uno only has two external interrupt pin and two encoders have to be connected. Furthermore, the interrupt blocks the code which could mean that serial data, which the user send from the interface, got lost. Finally not using interrupt also makes the code more readable.

The sensor is interfaced using the ams_as5048b library developed and maintained by sosandroid (Sosandroid, 2021). This library handles the I2C communication with the sensor, and when a new measurement is requested it will forward that request to the sensor. The AS5048 sensor is an absolute angle sensor when used in I2C mode. This means that it does not send values to the controller if it didn't request it. Being an absolute angle sensor also means that the angle ranges from 0-360 which means that it has to be transformed because the motor can rotate more than one rotation because of the transmission. To accomplish this the rotations are also tracked. When the measured angle jumps a large amount it means that the angle overflowed. The pseudo code is the following:

```
Retrieve a new angle from the sensor
Compare new angle to the previous angle
if the difference is bigger than 180deg increment or decrease the rotations
     by 1
set new continues angle to the new angle plus rotation multiplied by 360
set the previous angle to the new angle
```

This pseudo-code creates the following behavior, see Graph 5.1. Here you can clearly see that the measured angle, —●—, is confined in the 0-360 range whilst the continues angle, —■— shows the correct angle.

Graph 5.1, Angle measurement and calculated continues angle



5.2.2 Force sensor

The force sensor consists of two parts, the load cell, and the amplifier. The load cell is connected in the paddle as shown in the *3D design* section. The amplifier is connected to the load cell on one side and on the other side is connected to the Arduino Uno, see Figure 5.8.

To interface with the amplifier the HX711 Arduino library by Rob has been chosen (Tillaart, 2023). This library is a superset of another available HX711 library, but this library gives more control with a smaller footprint. The load cell does need to be calibrated to scale the incoming signal to a meaningful value. To accomplish this the zero point is first defined. Because there is a bit of noise in the signal the zero point is measured 20 times and then averaged together to gain value which we will subtract from the load cell output. The second step in the calibration is to determine the scaling factor. This is done by measuring a known force, 15N, and then scaling the average values to match that force. To measure and calibrate it the sensor was wedged in a vise and then a spring scale was attached to the top. the other side of the spring scale was connected to weight at the correct height, see Figure 5.6.

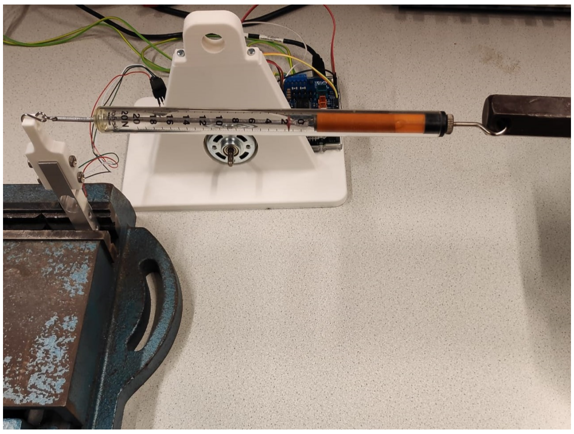


Figure 5.6: Setup to measure and calibrate the force sensor

5.3 Position Controller

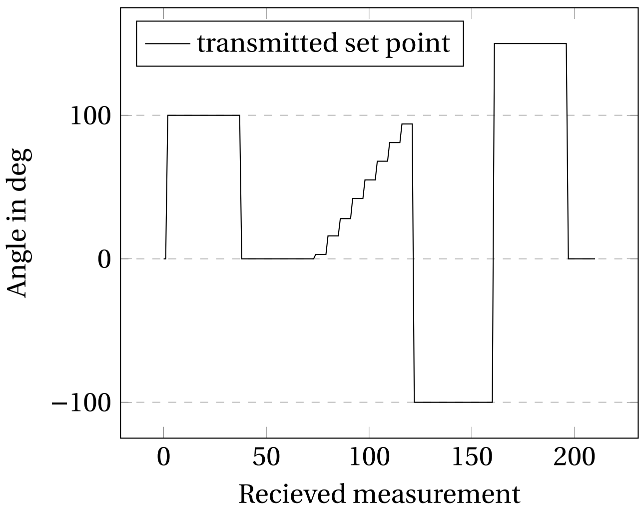
To control the motor a PID controller with position feedback is used. This allows us to set a position and then the system will move toward the new position. In order to accomplish this, it uses feedback from the position sensor. The PID controller uses the sum of three errors to move toward the setpoint. The first one is the proportional component, which multiplies the direct error between the setpoint and the current position with a constant K_p, the proportional gain. The proportional component allows the controller to act on the direct difference. The second component is the integral component, it multiplies the time integral error with another constant K_i, the integral gain. The integral component helps the controller with accumulated past errors, this minimizes the tracking and steady-state errors. Finally, the last component is the derivative component. It uses the time derivative of the error and multiplies with another constant K_d, the derivative gain. This component lowers the velocity changes at the output which can reduce overshoots. (P, 2022).

This controller was written in C++ as an Arduino library, this way it can easily be added to the program without it becoming cluttered. Furthermore, the motor is turned off if it reached a certain error from the setpoint. This helps the motor because it has a harder time rotating small distances because it does not have much torque when running at low voltages.

5.3.1 Calibration

There are multiple ways to calibrate a PID controller. An important part of calibrating it is gaining an inside into how the system works. Because of this, a helper program has been written in processing, java. The helper program shows the transmitted angle, set point, and motor control in one graph and allows the user to change the parameters directly with a slider. Furthermore, it has a test action that generates a test plot as shown in Graph 5.2. The slope is stepped because a new set point is transmitted every 250ms, this has been done because the serial connection should email available to receive sensor data.

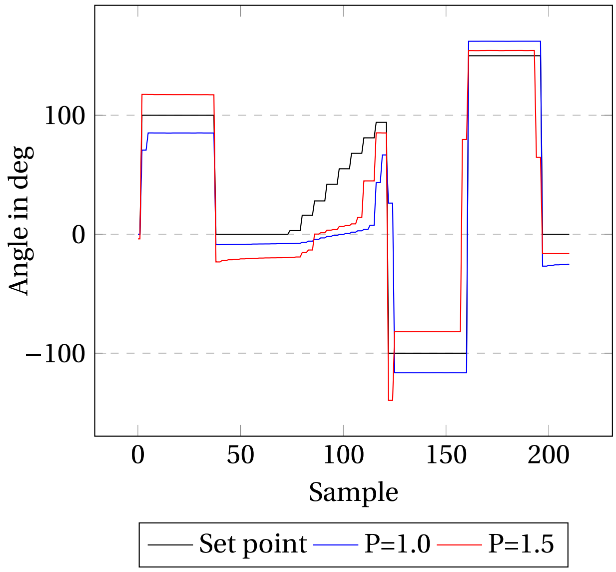
Graph 5.2, generated setpoint over time



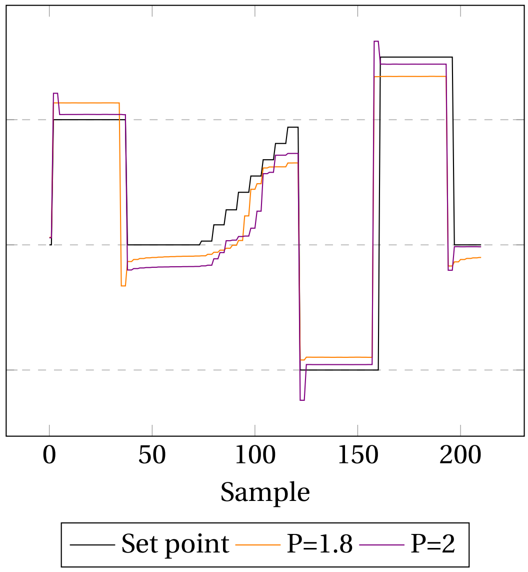
Because of the system dynamics manual tuning has been chosen over the Ziegler-Nichols method. Because the Ziegler-Nichols method needs the system to oscillate which is hard for our setup because of the friction and limited degrees of freedom (P, 2022). The manual tuning involves multiple iterative steps to arrive at good gain values. In the beginning, all gains are set to zero, giving us a known starting point. The first value to tune will be the proportional gain, it should be increased until it reaches the set point with a small overshoot. After which the derivative gain is increased until the overshoot disappears. When the derivative gain is increased too much the system will become sluggish and the signal will become noisy, which should be avoided (P, 2022). After the derivative gain has been tuned the signal should follow the set point well, but there might be a tracking error where it does not reach the set point accurately. To tackle this the integral gain can be increased. It should be increased until the actuation follows the set point well, if it is increased too much it will overshoot when a big change in the set point is introduced. The integral gain is very helpful when dealing with ramps where the set point slowly changes over time (P, 2022).

The same approach has been followed to calibrate the system. First, multiple values for P have been tested and plotted, see Graphs 5.3A and 5.3B. The plots have been separated for clarity. As shown in the first plot, a gain of 1, ——, is not enough as it lacks behind the set point quite a lot and often can not reach it. The gain of 2, ——, is too much as it overshot the set point. A proportional gain of 1.8 has been chosen to continue with the testing.

Graph 5.3A, P gains 1 and 1.5

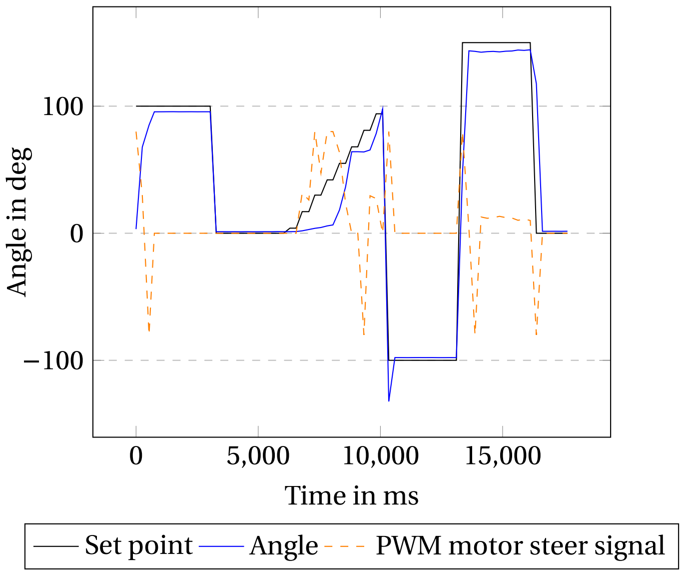


Graph 5.3B, P gains 1.8 and 2

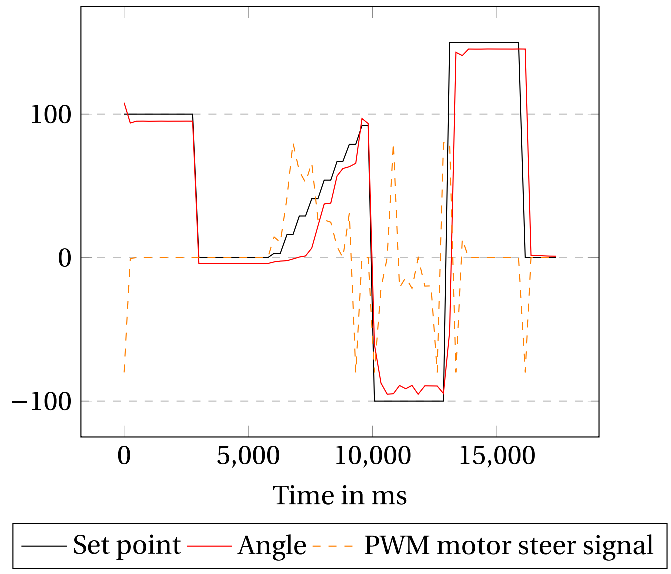


The second step is to tune the D to decrease the overshoot, as you can see in Graph 5.4A, a D of 0.25 decreases the overshoot, but some overshoot still remains. A D of 0.3 removes nearly all overshoots. But the signal does get a bit more chaotic, using a higher value for D increases the chaoticness of the signal even more. The signal already follows the set point roughly but there is a small offset in the measured angle. To improve this an integral gain can also be added as shown in Graphs 5.5A and 5.5B. This significantly improves the signal, but it does in some cases add more overshoots to the system, for instance when using a gain of .18.

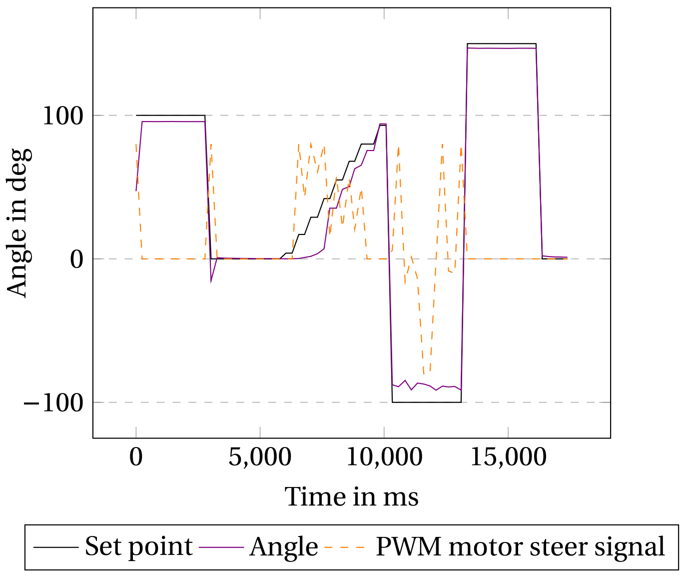
Graph 5.4A, P=1.8 I=0 D=0.25



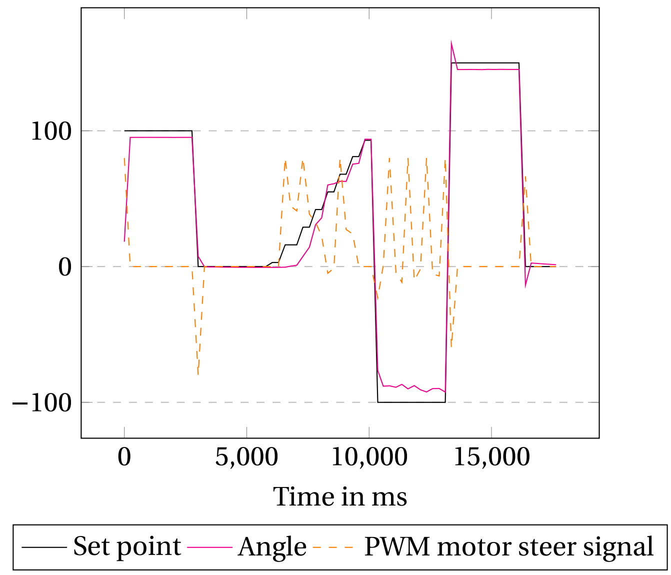
Graph 5.4B, P=1.8 I=0 D=0.3



Graph 5.5A, P=1.8 I=0.15 D=0.25

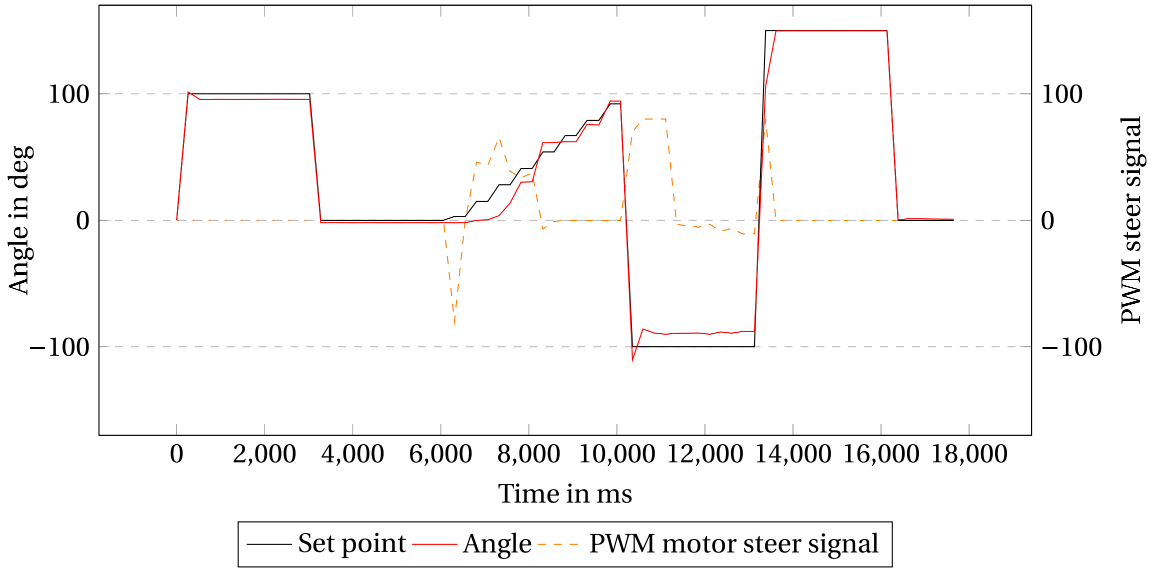


Graph 5.5B, P=1.8 I=0.18 D=0.3



The controller is further tuned using trial and error by slightly tweaking and testing the outcome. The final result is a proportional gain of 1.88, an integral gain of 0.18, and a derivative gain of 0.23. The final result can be seen in Graph 5.6. This controller is not perfect, but it follows the set point nicely in this situation and even follows the ramp quite well.

Graph 5.6, P=1.88 I=0.18 D=0.23



5.4 Wiring

The setup can be assembled as described in Section 5.1, 3D design. And the completed setup can be seen in Figure 5.7. The wiring of the system is straightforward and can mostly be done with jumper cables. But not all components come with header pins attached. It is possible to order them pre-soldered, but that will increase the price. Only the motor needs to be attached with separate cables that have to be soldered on the motor pins and screwed into the terminals at the Arduino shield. The complete circuit diagram can be seen in Figure 5.8. In the complete circuit, the Arduino is left out and only the shield is shown. This has been done to simplify the circuit diagram because the Arduino is connected to the shield on almost all pins. There are a few connections that need extra attention. First, the Vin Connect jumper on the back of the motor driver shield needs to be removed. Because this jumper connects the external power supply to the Arduino. Arduino (2023) recommends removing this jumper when using an external power supply higher than 9V, our setup uses 12V. Furthermore, the position sensor is used in 3.3 Volt mode, which means that the 3.3V and 5V pins need to be connected on the position sensor.

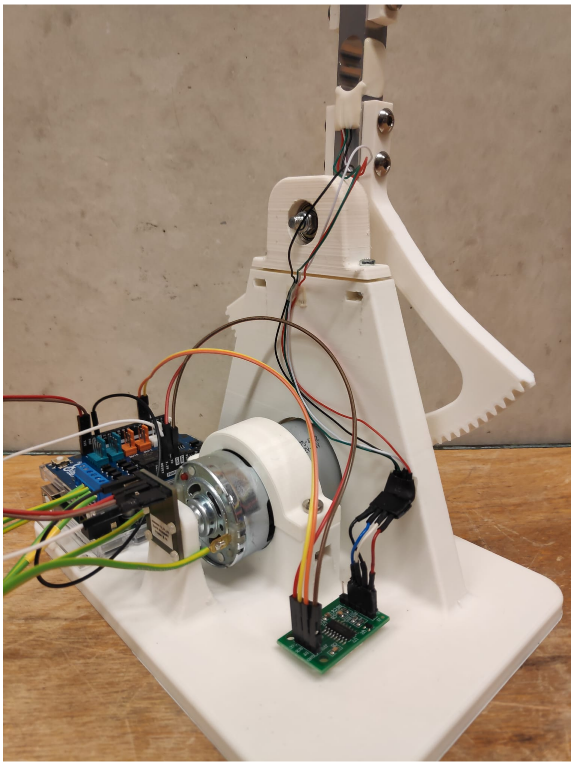


Figure 5.7: Completed setup

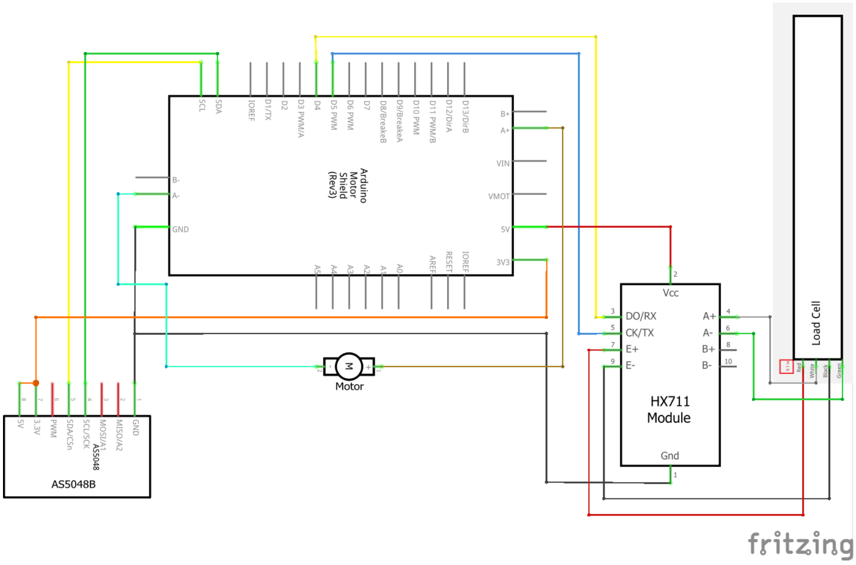


Figure 5.8: Wiring of the system (without Arduino)

6 Evaluation

An system evaluation has been carried out to test and evaluate the final setup. This evaluation focuses on the all the separate components of the system, the physical setup, the computational platform, the position sensor, the force sensor and the PID position controller.

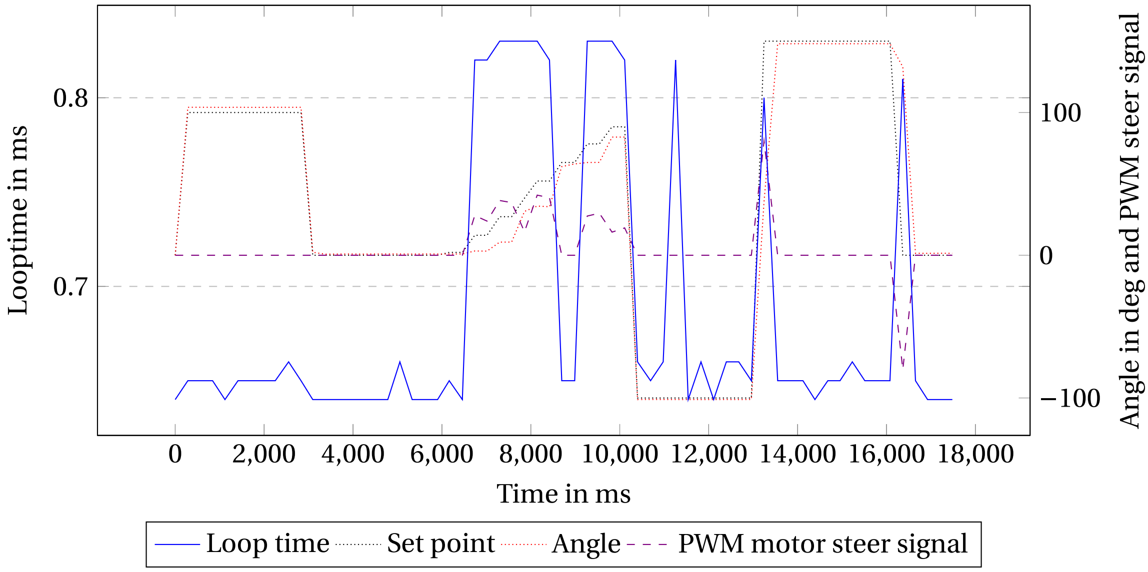
6.1 Physical setup

There are multiple requirements for the physical setup. But because no user evaluation is not part of a system evaluation requirements 2,3 can not be evaluated. These requirements state that the system should be easy to assemble and intuitive to use. The other requirements can be evaluated.
The physical setup has been built for €158.40. This could be cheaper if the user already has an Arduino or motor shield they can use because these are familiar parts. Furthermore, all these parts can be ordered by consumers from well-known websites. The setup can be built with basic technical knowledge. Because the component can easily be assembled using 3D-printed models and some bolts. They can also easily be wired using jumper cables and two wires soldered to the motor pins and screwed into the screw terminal on the motor shield.
The system's transmission also adds to its protection. The gears will detach and the motor can spin freely if the user modifies the code to make the motor spin too fast or raise the set point too high. This will help protect the system when the user experiments with it.

6.2 Computational platform

For the controller, the requirement was formulated that it should be able to complete a round trip under 2 milliseconds. Because the controller only controls the hardware control we can not test the actual loop time it will have with the control architecture. But we can test the loop time it has now. From measurements, the loop time without reading the serial is 0.6ms. In Graph 6.1, the controller follows the test set point path that has also been used to calibrate the controller. Here it receives a new set point every 500ms and transmits the motor control, angle, current set point, and previous loop time every 250ms. In this condition, it has an average loop time of 0.6867ms. Furthermore, because an Arduino is used the system has a fast turnaround time, costing a maximum of half a minute to send a new program to the Arduino. Finally, it reacts to a Serial message with one character per loop cycle and a message is often six characters long, $'S15.20'$. This means that it takes on average $0.6867 \cdot 6 = 4.1202$ milliseconds to react to a command over serial.

Graph 6.1, Loop time of the system whilst following the test movement

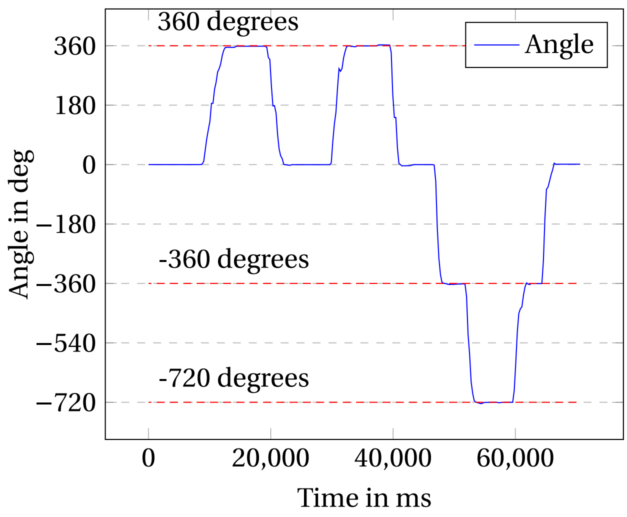


6.3 Input-Output

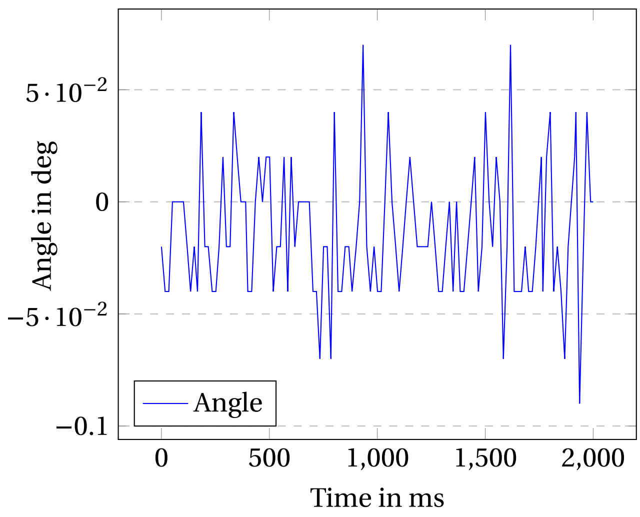
6.3.1 Position sensor

To evaluate the position sensor, it is rotated a complete circle multiple times. The data is then recorded and added to a graph. In graph 6.2A the small gear has been clockwise twice both times returning to zero. and after that, it is rotated counterclockwise twice without returning to zero. The zero position was marked with a marker on the gear and motor and then it has been rotated by hand. When rotated clockwise it reaches 358.51 degrees, and on the second time, it got to 362.72 degrees. the lowest value reached was -723.96, but it averaged at -720.3deg which is close to two negative rotations, -720 degrees. This is probably because it was very hard to align the gear in the same position each time. In Graph 6.2B the position is recorded whilst it stays at zero degrees. The maximum deviation from zero is 0.09 degrees and the precision is 0.023 degrees. This means it satisfies the requirement of at least a precision of 0.1 degrees. The accuracy of the sensor in this setup is -0.012 degrees, measured over a period of 10 seconds.

Graph 6.2A, Measured angle



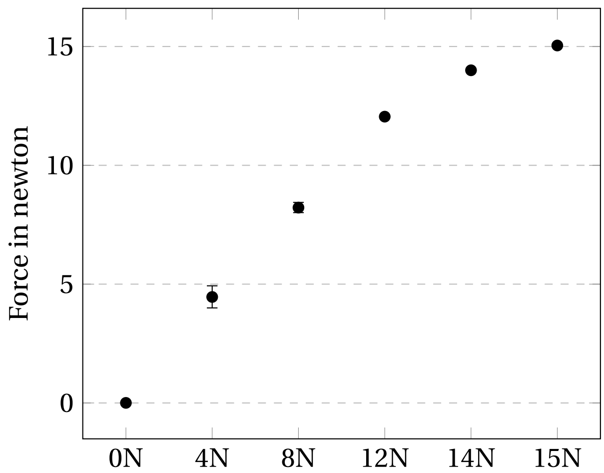
Graph 6.2B, measured angle at 0 deg



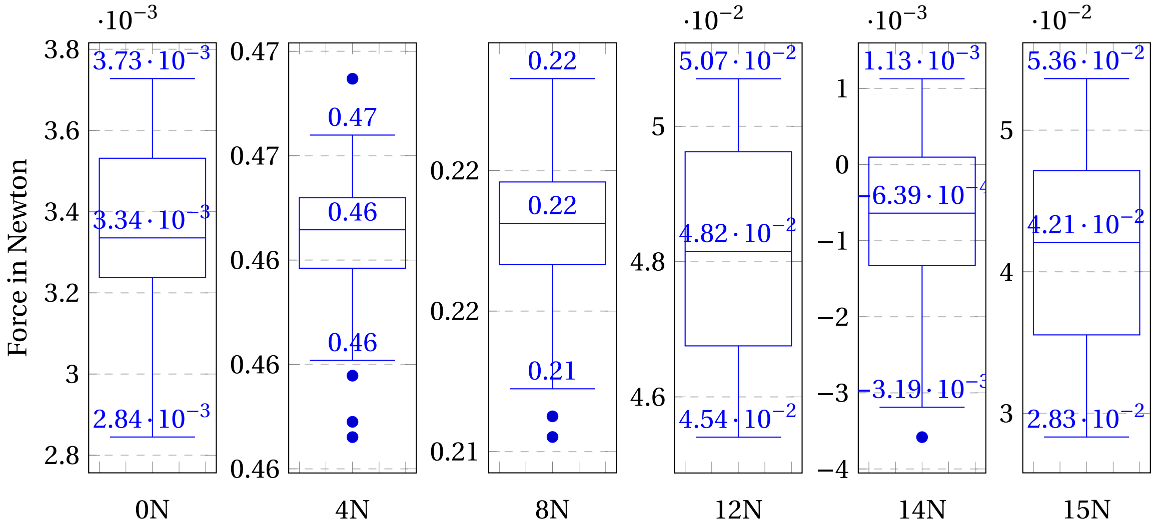
6.3.2 Force sensor

The force sensor has been measured at certain newton of forces using the calibration setup discussed in the realization, see Figure 5.6. The sensor already satisfies the requirement of being able to measure 15N, because that was also the requirement when ordering the parts. But the force sensor does not have a linear error as can be seen in Graph 6.3. The graph shows the measured data points and the measured force there, it also shows error bars to visual the maximum and minimum value measured at that point. But the error bars are only visible on 4N and 8N, which means that the error is different for the amount of force that is measured. The calibration has improved the error on zero newtons and around 15 newtons.

Graph 6.3 Force sensor, different newton measurements with error bars



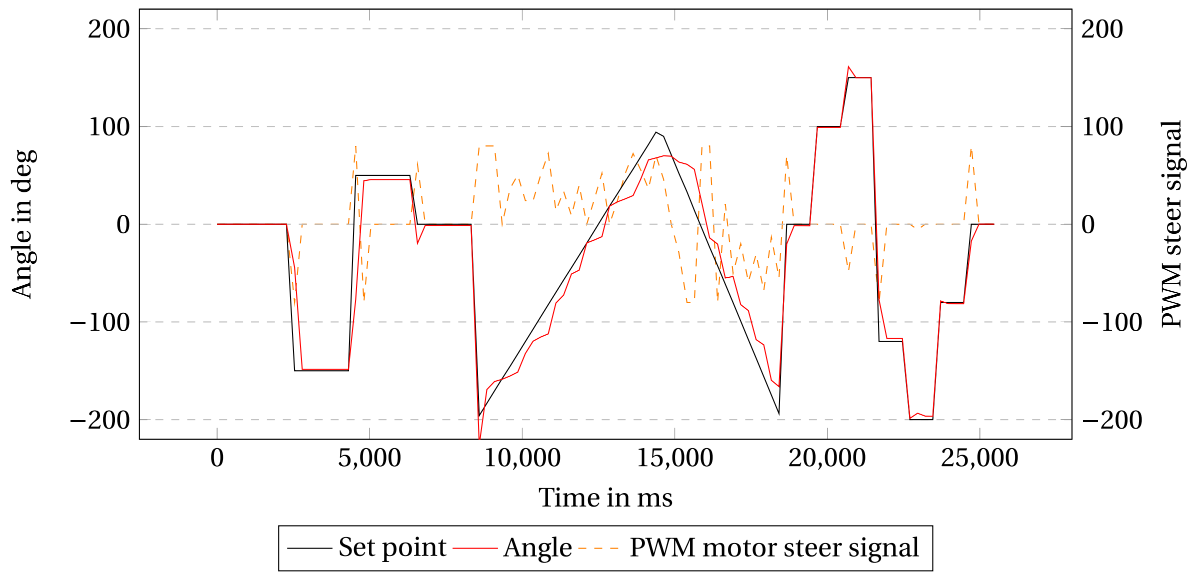
In Graphs 6.4 the errors from the wanted measurement point are plotted to gain more insight into the sensor profile at the measured newtons. As shown in these graphs, 4N and 8N not only have an lower accuracy, they also have an higher spread of the error. This means that the precision is also lower at those measurement points.



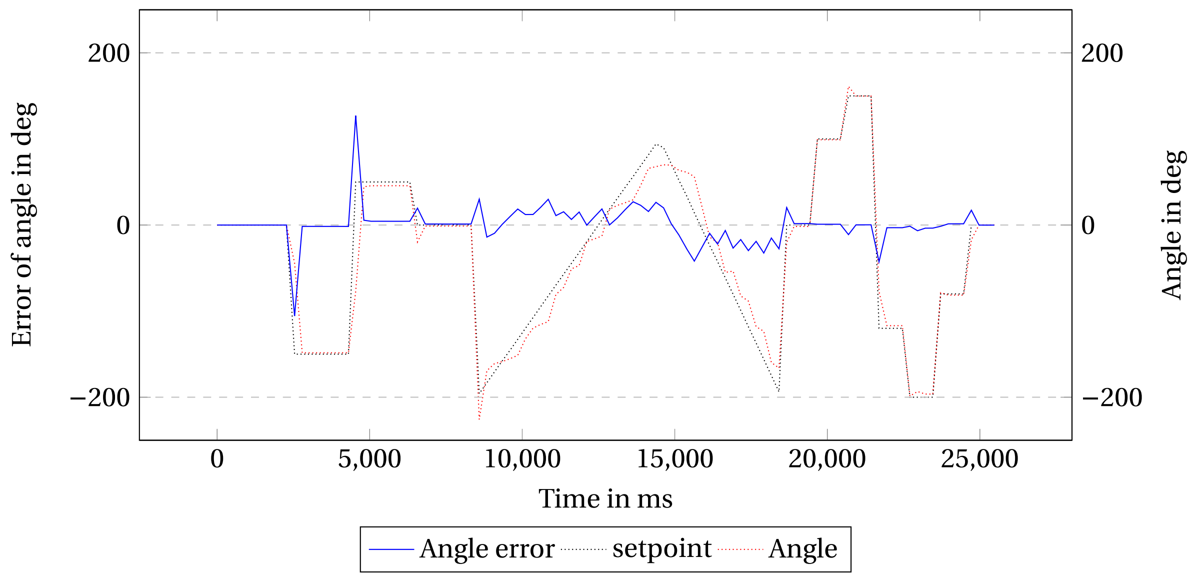
6.3.3 Position actuation

To evaluate the actuation of the position a set point path has been created. The interference of serial communication has been lowered by designing the set point path directly in the Arduino. This allows the set point path to be a lot smoother. When the sequence of set points has been triggered the system will transmit the data to the connected laptop to record the data. In Graph 6.5 the results are presented of this test. The set point path, ——, is different and more complex than the calibration version. This has been chosen to test the system in various situations. It can be seen that the system often finds the right set point, but it can not always get to the perfect angle. And it struggles quite a bit with slopes and it seems like it uses steps to go up or down on them. The controller seems to handle the faster changes in the third part quite well, as shown in Graph 6.6, the error is not that big there.

Graph 6.5 Position actuation test



Graph 6.6 Angle error in position actuation test



7 Discussion

In Chapter 6 Evaluation, a system evaluation has been carried out. The results that are presented there will be discussed in this chapter. It is divided into the same parts, starting with the physical setup, after which the computational platform will be discussed, followed by the sensor and finally, the position PID controller will be discussed.

7.1 Physical setup

A user evaluation has not been carried out because of time constraints and because it does not add a lot of value to this graduation project. Almost the entire system can be evaluated using the system evaluation. Requirements 2 and 3, which state that the system should be easy to construct and intuitive, can not be assessed without user testing. The other requirements were assessed, requirement 1 is met because the client said that €158.40 is an affordable price for the entire setup. Secondly, requirement 5 has been tacked in the 3D design, the paddle can detach from the gear driven by the motor when the motor spins too much. A stop has been added in the software which also turns off the motor when it happens. Thirdly requirements 10 and 11 have also been met, because as stated in the evaluation, all parts are ordered at well-known consumer web shops and the setup remained intact even after intensive testing in the position actuation testing. Finally, requirement 11, which states that the system should be straightforward to assemble with only basic technical knowledge, has been partly passed. The original goal was to achieve a setup that does not need any soldering. But it quickly became apparent that many parts will be more expensive or not available with pre-soldered headers. It is possible to order most parts with headers for a more expensive price, but the motor needs to be soldered. On the other side, the solders in this setup are all relatively easy and should be possible for people with little soldering experience. Furthermore, our target group generally does have soldering experience.

7.2 Computational platform

As discussed in the evaluation, Section 6.2 and shown in Graph 6.1, the loop time varies between 0.6ms at rest and 0.83ms at peak. This is a lot lower than the requirement 14, 2ms. But this is without the control architectures. According to my collaborator Famke van den Boom, her system with the control architectures, hardware, and serial communication to interface runs at 1ms. From this, it could be expected that this setup with the control architecture can also run under 2ms. But this will have to be tested in the future to be sure. Furthermore, in Graph 6.1 it seems that the loop time increases every time the motor is controlled with a PWM signal which is not zero. After some testing, the time it takes to write PWM signals was roughly determined and it is not enough to influence the loop time enough for it to be visible in the graph. It takes about 12 microseconds for an Arduino Uno loop that writes 0 or 255 to a PWM pin, but it takes between 8 and 20 microseconds to write any value between 0-255 to a PWM pin. There is a bit of optimization in the controller code that I wrote which turns the controller off if the angle is close to the set point. This allows the controller to skip the math and control statements, which influences the loop time.

7.3 Position sensor

The Position sensor has been evaluated in Section 6.3.1. The sensor is very accurate and precise when turning the motor shaft. The small differences from 360 degrees could be because it was turned by hand and it is very hard to align the gear with the line. Furthermore, it also has a good precision of 0.023 degrees which far exceeds the requirements. This high precision and accuracy is needed for the feedback loop used to set a position with the motor. The position sensor does not have any calibration but will be reset to zero every time the Arduino is powered on. This has been chosen to allow the setup to be reset when it spins off the paddle when testing. This does mean that it is expected that the user has the paddle straight upwards when they power it on.

7.4 Force sensor

The force sensor has a nonlinear error, this means that the accuracy and precision of the sensor are not proportional nor the same at a different force. The sensor has been calibrated at zero and at 15N which allows the sensor to be accurate at 0N and around 15N. But the sensor is not as accurate away from these values, between 4 and 8 Newton. When the sensor was calibrated at 6 Newton the higher forces were further off, having a spread of 0.2 Newton (at 14N), which is more than 4N has now. Furthermore, the recession and accuracy are quite good at 0 and 12 to 16 Newton. It could be investigated in the future if a different calibration method that could find the scalar of multiple force values gives more accurate values. But because of time constraints, this has been tested yet. The measured forces could also be off because the setup to measure the forces was all mechanical relying on human eyes to read off the results. Furthermore, it could have been possible the setup moved a tiny bit when measuring the forces because one side of the force measurement wasn't rigidly attached. Instead, it relied on friction from the weights to keep them in place.

7.5 Position actuation

Graph 6.5 in Section 6.3.3 shows the capabilities of the PID controller well. and Graph 6.6 shows the error between the measured angle and the set point. The goal is to keep the error as small as possible. But at the start, there are two spikes in the error which are caused because the controller is lacking behind the set point. This together with the slower approaches of the set point later on in the graph could mean that the integral gain is a bit too high. Furthermore, the controller looks good at following the quick changing set points at the final part of the test which will be useful in the final setup. But the most important part will be slopes because the actual set points will most likely also form slopes. Currently, the system is not that good at slopes because it goes up the slope in steps. When the slope is steeper, the controller can more easily blend these steps to follow it easier as shown with the second slope. On the other hand, the controller overshoots the change in the slopes quite a bit which could have multiple reasons. It is not can not or is not allowed to change the velocity of the motor quickly with a bit too low proportional gain or a bit too high derivative gain. It could also have a bit too high integral gain which wants the motor to first correct the steady state error it had going up the slope. The stepping behavior and the steady state offset at some angles have to do with the cogging of the motor. Within the motor, cogging torque is produced when the sides of the rotor teeth line up with the sides of the stator teeth (Burroughs, 2022). This torque causes the motor to have preferred angles it would like to stop in and it costs constant force to stop at another position. Motor cogging can be solved by swapping out the motor for a cordless one (Burroughs, 2022). But cogging could also be tackled using the controller. The cogging torque can be seen as friction which changes with the angle. To minimize it one can map the cogging torque at different angles and apply the corresponding counter torque.

8 Conclusion and Future Work

8.1 Conclusion

The goal of this graduation project has been to design and implement the hardware for an affordable, intuitive educational kit that teaches its users about telemanipulation. Such a kit is needed because currently there are not many affordable educational telemanipulation setups. And a physical educational kit has the potential to improve the learning of its users by allowing them to experiment with it (Han and Black, 2011).

In order to design the DIY kit two sub-research question has been constructed. The first sub-research question inquires about the most important concerns when designing the kit. The most important concerns are that the system should be affordable, robust, and able to measure and actuate accurately. These concerns have been transformed into requirements in Section 3.1. The second sub-research question is about the existing systems. These existing systems formed the basis of the design of the setup. The 3D designs have been inspired b the existing systems combined with the requirements which created a complete DIY kit. Creating a setup using 3D prints has allowed us to make it more intuitive as it guides the user through the design.

In the end, a nearly completed prototype has been developed that can sense the position and force and can actuate the position as well. Due to time constraints, the force controller has not been implemented. There are also a few more areas of the system that could use some improvement, this will be laid out in the next section, Future Work.

8.2 Future Work

As stated in the conclusion and discussion multiple areas of the system could use improvement, this section will go over them and give an inside on how they can be tackled.

The force actuation control has not been implemented yet. But it is needed in order to use half of the architecture, it is needed for the position-computed force and position-measured force architectures. For this reason, it is important to implement a force controller which can actuate the force accurately. A P controller has been tested, but this kind of controller was not found to be very accurate because of multiple reasons. It suffered from poor force measurement precision, friction in the transmission, and cogging torque of the motor.

This brings us to the next recommendation. The cogging torque is having an influence on the position actuation and, as discussed before, also on the force actuation. It would be useful to investigate the cogging torque and how to minimize the effect of it. One possible way to do it would be to map out the cogging torque at different angles and then apply a counter torque at the corresponding angle.

Thirdly a user evaluation should be carried out to validate the ease of assembly and robustness of the system. The user evaluation would let multiple users assemble the setup and would let them interact with the setup in order to evaluate if the requirements have been met.

Finally, it would be beneficial if the force sensor would be more accurate and precise. In order to achieve this it could be experimented with different calibration scales. Currently, the calibration scale has been calculated from 15N. Because of this, the sensor is very accurate around 15N. It should be evaluated if calibration at different forces would give better results. Furthermore, different ways of calibration could also be explored. For instance, a more global calibration that would use multiple data points at different newtons could also be explored.

Bibliography

Arduino (2023), Arduino.
https://www.arduino.cc

Burroughs, M. (2022), Cogging torque in permanent magnet motors.
https://www.precisionmicrodrives.com/cogging-torque-in-permanent-magnet-motors#:~:text=Introduction,stator%20teeth%20within%20the%20motor.

Dresscher, D. (2023), Avatars.report.
https://avatars.report/

Gassert, R., J.-C. Metzger, K. Leuenberger, W. L. Popp, M. R. Tucker, B. Vigaru, R. Zimmermann and O. Lambercy (2013), Physical Student–Robot Interaction With the ETHZ Haptic Paddle, **vol. 56**, no.1, pp. 9–17, ISSN 0018-9359, doi:10.1109/TE.2012.2219310.

Gillespie, R., M. Hoffinan and J. Freudenberg (2003), Haptic interface for hands-on instruction in system dynamics and embedded control, in *11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2003. HAPTICS 2003. Proceedings.*, IEEE Comput. Soc, pp. 410–415, ISBN 0-7695-1890-7, doi:10.1109/HAPTIC.2003.1191328.

Han, I. and J. B. Black (2011), Incorporating haptic feedback in simulation for learning physics, **vol. 57**, no.4, pp. 2281–2290, ISSN 0360-1315, doi:https://doi.org/10.1016/j.compedu.2011.06.012.

Kundu, S., B. O'Flynn, J. T. Sanchez and M. Walsh (2021), Low Latency Haptic Feedback for Battery Powered HCI for the Tactile Internet, in *2021 Smart Systems Integration (SSI)*, IEEE, pp. 1–4, ISBN 978-1-6654-4092-9, doi:10.1109/SSI52265.2021.9466961.

Mader, A. and W. Eggink (2014), A DESIGN PROCESS FOR CREATIVE TECHNOLOGY.

Martinez, M. O., J. Campion, T. Gholami, M. K. Rittikaidachar, A. C. Barron and A. M. Okamura (2017), Open source, modular, customizable, 3-D printed kinesthetic haptic devices, in *2017 IEEE World Haptics Conference (WHC)*, IEEE, pp. 142–147, ISBN 978-1-5090-1425-5, doi:10.1109/WHC.2017.7989891.

Martinez, M. O., T. K. Morimoto, A. T. Taylor, A. C. Barron, J. D. A. Pultorak, J. Wang, A. Calasanz-Kaiser, R. L. Davis, P. Blikstein and A. M. Okamura (2016), 3-D printed haptic devices for educational applications, in *2016 IEEE Haptics Symposium (HAPTICS)*, IEEE, pp. 126–133, ISBN 978-1-5090-0903-9, doi:10.1109/HAPTICS.2016.7463166.

Melchiorri, C. (2003), Robotic telemanipulation systems: an overview on control aspects, **vol. 36**, no.17, pp. 21–30, ISSN 14746670, doi:10.1016/S1474-6670(17)33365-7.

Mendelow, A. L. (1981), Environmental Scanning-The Impact of the Stakeholder Concept, in *International Conference on Interaction Sciences*.

Mind Tools Content Team (2023), Stakeholder Analysis.

Okamura, A. M., C. Richard and M. R. Cutkosky (2002), Feeling is Believing: Using a Force-Feedback Joystick to Teach Dynamic Systems, **vol. 91**, no.3, pp. 345–349, ISSN 10694730, doi:10.1002/j.2168-9830.2002.tb00713.x.

Orta Martinez, M., C. M. Nunez, T. Liao, T. K. Morimoto and A. M. Okamura (2020), Evolution and Analysis of Hapkit: An Open-Source Haptic Device for Educational Applications, **vol. 13**, no.2, pp. 354–367, ISSN 1939-1412, doi:10.1109/TOH.2019.2948609.

P, P. (2022), PID Controllers.
https://tttapa.github.io/Pages/Arduino/Control-Theory/Motor-Fader/PID-Controllers.html

Riddle, M., J. MacDermid, S. Robinson, M. Szekeres, L. Ferreira and E. Lalone (2020), Evaluation of individual finger forces during activities of daily living in healthy individuals and those with hand arthritis, **vol. 33**, no.2, pp. 188–197, ISSN 0894-1130, doi:https://doi.org/10.1016/j.jht.2020.04.002.
https://www.sciencedirect.com/science/article/pii/S089411302030079X

Sosandroid (2021), Arduino Library for AMS AS5048B I2C - 14-bit magnetic rotary position sensor.
https://github.com/sosandroid/AMS_AS5048B

Tillaart, R. (2023), GitHub - RobTillaart/HX711: Arduino library for HX711 24 bit ADC used for load cells and scales.
https://github.com/RobTillaart/HX711

UltiMaker (2023), UltiMaker Cura.
https://ultimaker.com/software/ultimaker-cura/

Wong, C. and A. Okamura (2005), The Snaptic Paddle: A Modular Haptic Device, in *First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, IEEE, pp. 537–538, ISBN 0-7695-2310-2, doi:10.1109/WHC.2005.135.

Yigit Sizlayan, S. and M. M. Ankarali (2019), Design of Programmable, High-Fidelity Haptic Paddle, in *2019 International Aegean Conference on Electrical Machines and Power Electronics (ACEMP) & 2019 International Conference on Optimization of Electrical and Electronic Equipment (OPTIM)*, IEEE, pp. 540–546, ISBN 978-1-5386-7687-5, doi:10.1109/ACEMP-OPTIM44294.2019.9007129.