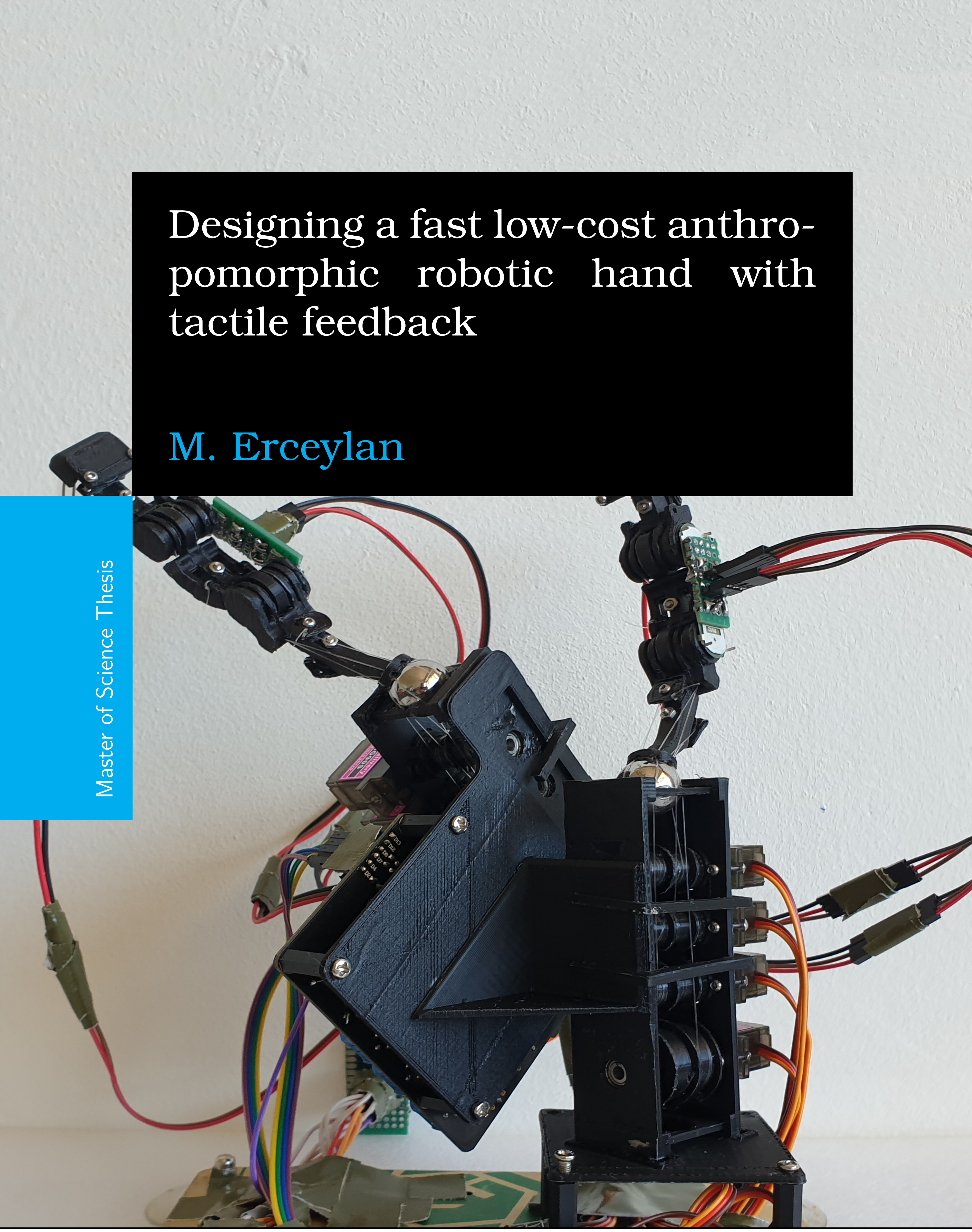


Designing a fast low-cost anthropomorphic robotic hand with tactile feedback

M. Erceylan

Master of Science Thesis

TU Delft
Delft University of Technology

Designing a fast low-cost anthropomorphic robotic hand with tactile feedback

Master of Science Thesis

For the degree of Master of Science in Embedded Systems at Delft University of Technology

M. Erceylan

July 15, 2022

Faculty of Electrical Engineering, Mathematics and Computer Science (EEMCS) · Delft University of Technology

Abstract

In this Thesis a fast, low-cost, anthropomorphic robotic hand with tactile feedback is designed. The hand consists of an index finger and a thumb, both of which have four degrees of freedom. All eight degrees of freedom are fully actuated using eight servomotors whose forces are transferred using tendons. The design introduces tendon routing to minimize the change in tendon tension as the joints rotate. To accomplish this at the 2 degree of freedom joints of thumb and index finger, a novel design for the ball and socket joint is created. The hand can be controlled using a tactile glove called the SenseGlove which also enables the user to receive force feedback. Testing reveals that the robot can open and close its fingers within 275 ms, and is able to grab a variety of objects.

Table of Contents

1 Introduction **1**

1-1 Tactile internet and its applications . . . 2

1-2 Challenges in Tactile Internet . . . 3

1-3 This project . . . 3

2 Other robotic hands **5**

2-1 Robotic hand designs . . . 5

2-1-1 Actuation of robotic joints . . . 6

2-1-2 Other tendons based robotic hand designs . . . 6

3 Towards designing an actuation and transmission system **11**

3-1 Anatomy of human hand . . . 11

3-1-1 Joints of the fingers . . . 12

3-1-2 Movements of fingers . . . 13

3-2 Actuation methods . . . 15

3-3 Kinematic model of the robotic hand . . . 16

3-4 A single 1 DOF joint . . . 17

3-4-1 First look at a naive 1 DOF joint implementation . . . 17

3-4-2 Guiding along pulleys . . . 20

3-5 A second 1 DOF joint . . . 21

3-5-1 Required number of tendons and servomotors . . . 21

3-5-2 First implementation of the second 1 DOF joint . . . 24

3-5-3 Crossing tendons . . . 25

3-5-4 Routing the flexor next to the extensor . . . 26

3-5-5 Joint angles affecting others . . . 27

3-6 The 2 DOF joint . . . 30

3-6-1 Using two 1 DOF joints to emulate a 2 DOF joint . . . 30

3-6-2 A ball and socket joint as a 2 DOF joint . . . 30

4 Creating the parts and assembling the robotic hand **35**
- 4-1 The finger and its first two joints 35
  - 4-1-1 Core of the finger 36
  - 4-1-2 Guiding the tendons 36
  - 4-1-3 Combining the fingers layers 38
- 4-2 Creating the ball and socket joint 39
- 4-3 The base of the finger 40
  - 4-3-1 Designing bobbins 41
  - 4-3-2 Combining two fingers 42
- 4-4 Making the hand tactile 44
- 4-5 Electrical wiring of the strain gauges and the servomotors 45
- 4-6 Costs of the robotic hand 47

5 Implementing the robotic hands movement **49**
- 5-1 Moving the servomotors using Arduino 50
- 5-2 Communications between Arduino and Unity 51
- 5-3 Using Unity to control the Robotic hand 52
  - 5-3-1 Changing joint angles manually in Unity 52
  - 5-3-2 Changing joint angles using SenseGlove inputs 52
  - 5-3-3 Implementing force feedback 56

6 Evaluating the robotic hands performance **59**
- 6-1 Experiments using manual control 60
  - 6-1-1 Opening and closing at maximal speed 60
  - 6-1-2 Opening and closing the finger periodically 60
- 6-2 Experiments using the SenseGlove 62
  - 6-2-1 Copying hand posture using SenseGlove 62
  - 6-2-2 Grabbing objects using the SenseGlove 64

7 Conclusion **71**
- 7-1 Future work and additional insights 72

Bibliography **75**

Chapter 1

Introduction

During the COVID-19 pandemic people have realised more and more the need to see, feel, and talk to others. The limits that were put on interaction with other humans have left many in a decreased mental state. While talking and seeing others can comfort people in desperate times, it is no substitute for a hug or simple touch on the shoulder and during the pandemic these interactions that require the sense of touch, are the ones people were limited in the most. Our technology has come far enough for us see and speak to loved ones, however far apart. This communication was in the form of for example phone calls. Later video communications were added to these interactions. Humans could now see each other in seemingly real time and the combination of both auditory and visual information made the illusion of being present somewhere else increasingly convincing. Now with ever increasing internet speeds and decreasing latencies the possibility to add tactile communications to the mix is becoming increasingly realistic.

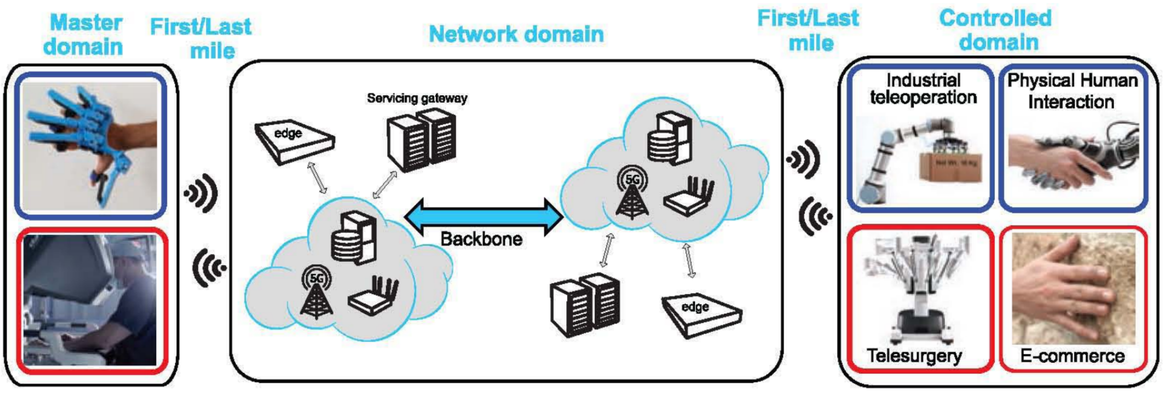


Figure 1: A schematic representation of the various parts of tactile internet. The master domain often contains a human user that interacts with a different environment (controlled domain) through the use of a master device. (from [14])

1-1 Tactile internet and its applications

Transmitting the sense of touch from one place to another is a form of tactile communications. Figure 1 depicts a schematic representation of the various parts that comprise it. In the master domain a human user usually interacts with a distant environment through the use of master device. This master device can for example be in the form of a wearable tactile glove. In this case the master device could send sensory data regarding the users hand position and posture through the network domain to the controlled domain. The controlled domain could for example contain a robotic hand. This robotic hand could be made to replicate the humans movements and any attached sensors could register data such as forces applied to the robotic hand. In turn the robotic hand would transmit this sensory data through the network domain back to the human user. The master device can then provide the user with feedback derived from the robotic hands sensory data. This feedback can be in the form of blocking the human hands movement. This would complete the loop of tactile communications. Figure 2 shows an example of such a human wearable tactile glove that is controlling a well known robotic hand, the Shadow Hand [6]. The user can be seen wearing haptic gloves that not only record the users movements, but also apply haptic feedback to the users fingers. This haptic feedback comes from the sensors of robotic hand and creates the illusion of being directly able to touch, feel, and handle objects held by the robotic hand. This type of interaction with distant environments has all kinds of interesting applications. It opens up possibilities to implement remote surgery, move objects in industrial applications, or handle objects in otherwise dangerous or remote environments such as space. On top of this, being able to handle objects in real physical environments is only one of the implementations of haptic technologies. Haptic device users can also interact with virtual environments. This allows among others to provide medical training without the risks, users to feel and handle items before they buy them in a shop, and countless of application in the gaming and entertainment industries [13].

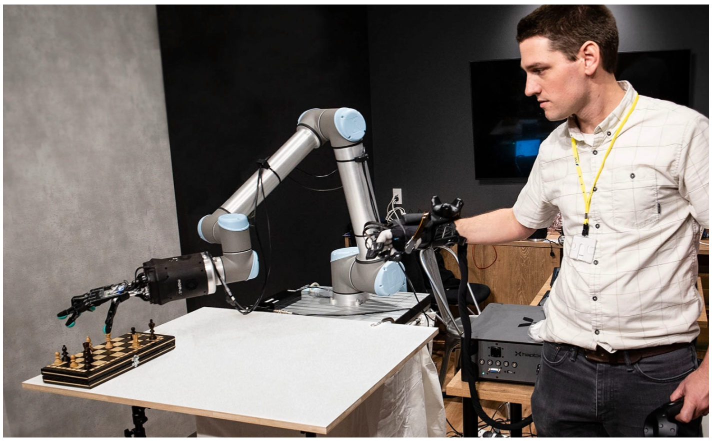


Figure 2: The Shadow Hand [6] (left) is controlled by human user who is wearing tactile gloves (right). Tactile sensors present in the robotic hand provide feedback to the user. The worn gloves apply this feedback to the users hands, which makes the user feel as though he is handling the objects in his own hands.

1-2 Challenges in Tactile Internet

One of the reason tactile applications have not taken over the world is stringent latency requirements. Auditory and visual data can, depending on application, trick the human mind into being perceived as happening in real-time in the 100 ms latency range. For tactile applications where a human is directly in control of a haptics device this latency needs to be brought down to around 1 to 10 ms depending on the application [8] [15] [24].

Currently changes in the communications infrastructure such as the fifth generation of mobile communications (5G) have specific goals to meet these ultra low latency requirements [8]. Such upgrades can be a stepping stone towards haptic communications, however there are other obstacles as well. Most devices that are used in haptic experiments are not as fast nor accurate as the human hand for example. While for many entertainment applications this would be mere inconvenience, for critical applications such a remote surgery this would be a matter of life and death. Commercial options that are available now such as the Shadow Hand [6] are very expensive. This can be huge deterrent to additional research into haptics and to making it commercially successful. While there have been hands such as the Open-source, anthropomorphic, underactuated (OSAU) robot hand [17], that attempt to reduce costs significantly. These types of robotic hands are often not suitable for use in haptic applications, due to not having enough control over the hand movements. Lately designing underactuated hands has become more popular. These type of hands are not able to independently control every movement of the hand. This often leads them to cost less money to fabricate due to a decreased amount of actuators, but also causes them to lack the dexterity required to match the human hands movements. A third variable that has often been overlook is the human hands speed. The category of hands designs that specifically aim to provide the necessary dexterity, ignore the fact that many movements of the hand require a certain amount of speed. Think about flicking a coin or typing on a keyboard. These reasons let us to design our own anthropomorphic, dexterous robotic hand where the main focus is placed on novel improvements to the hands movement design to increase speed and accuracy, while keeping the costs comparatively low.

1-3 This project

This project aims to create low-cost, anthropomorphic robot hand with novel joint design that increases speed and accuracy over other designs. Using cheap readily available parts and 3D printing costs can be reduced to $<$ 200 USD for a robot hand, which is orders of magnitude smaller than commercial robotic hands such as the Shadow Hand, which costs $>$ 60 000 USD [25]. The design of this robotic hand will revolve around improving the actuation and transmission system of mechanical forces to increase the robotic hands speed and accuracy, this will be extensively discussed in Chapter 3. In the process a novel design for a free moving ball and socket joint is developed. Chapter 4 deals with the design of the actual parts that form the hand and how these are combined to form the robotic hand. In Chapter 5 the code and software used to move the hand is discussed using a connected computer and using a tactile glove named the SenseGlove [4]. Finally in Chapter 6 experiments will be conducted to test the robotic hands capabilities.

Chapter 2

Other robotic hands

The human hand is one of the most complex parts of the human body. Their dexterity is what sets us apart from other animals by enabling us to interact with our environment in complex ways and to create and use complex tools. Because of the great variety of ways that we use our hands many others have tried to replicate it. Use cases include industrial applications, prostheses or in our case for use in tactile applications.

This chapter deals with other designs and research into robotic hands. Their strength and weaknesses are discussed and how their movement is realized.

2-1 Robotic hand designs

Because of the many purposes human hands serve, many attempts have been made to replicate them. Depending on the use different factors of the hand become important. In this case the robotic hand will be used for tele-operation and tactile purposes, this leads to the following points being important:

Rotational speeds of the joints

The robotic hands joints should ideally be able to move as fast as the human users hand is capable of. This is important for the quality of experience for the user. Certain human movements also require fast movements to be able to achieve at all. Think about flipping a coin or throwing something with your fingers.

Accuracy of joint angles

To convincingly interact with a distant environment using a robotic hand, the human hands position should be accurately copied. This means accurate angle of joints. Even if this robotic hand joint angles are off by a few degrees, some tasks that require precise hand coordination become impossible.

Similarity of shape

This is related to accuracy. A hand that is not similar in size to the human user will most likely feel very unnatural to a human user, as they have been living their entire lives with a single pair of hands of particular size. If size of parts of the fingers do not match up, accurately copying a human users joint angles does not translate to the same fingertip position. If instead just the end position of the fingertip is to be followed, the other phalanges of the finger will be in different positions when compared to the human user. In a lot of cases these other parts of the finger are also involved in handling objects. Any mismatch between parts of the robotic hand and the user hands (such a phalangeal length) could lead to the user having to grab an object another way than that user is used to and on top of this the perceived shape of the object might be entirely different for a user when force feedback is used.

Space for sensors

The point of a haptics device is interact with its environment and provide feedback in the form of forces. To measure these forces applied to the robotic hand requires sensors. These in turn requires space. To achieve the needed space all the parts that move the hand should just be designed sufficiently small.

2-1-1 Actuation of robotic joints

Many of the aforementioned points apply generally to anthropomorphic robotic hands. One of the first design choices that has a large impact on these point is the used actuation and transmission system for the robotic hand. Lately the use of servomotors has been favored by most robotic hand designs for their ease use, low cost, and relatively small size and weight. To transfer the forces from these actuators to their respective joints, tendons (cables/strings) are often used similarly to how this happens in the human hand. Compared to other transmission systems such as gears, tendons are very lightweight, require very little amount of space, and are very efficient in transferring forces.

Another option is to omit the transmission system and to place the used actuators directly at the joints location. This option is highly dependent on the used actuators and often limits the choice to electrical motors. While these are relatively small, having to place them directly at the joints location still requires a lot of space and also add more weight to the fingers. This could lead to the fingers moving slower or the need for higher power motors.

Using another transmission system while using tendons is sometimes also done. A well known example of this is the Shadow Hand [6]. The Shadow Hand uses air muscles which are similarly to human muscles only able to contract. This means to fully actuate a 1 DOF joints, two air muscles are needed. On top of this they are expensive when compared to the other options.

2-1-2 Other tendons based robotic hand designs

The human hand is most often modelled as having three joints per finger. With the four basic fingers having two 1 DOF joints and a single 2 DOF joint. The thumb on the other hand has two 2 DOF and one 1 DOF joint in this model. This is shown visually in Figure 3.

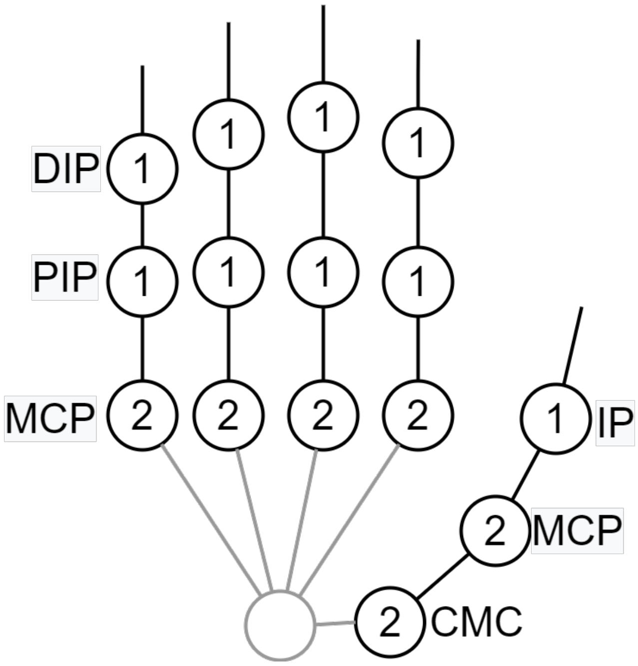


Figure 3: Kinematic model of the human hand. The numbers represent the degrees of freedom associated each joint.

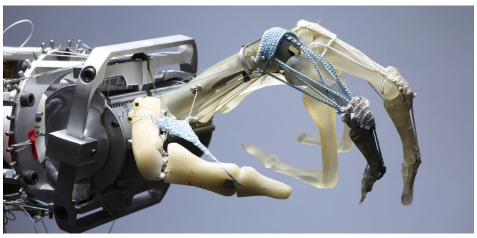


Figure 4: the Anatomically Correct Testbed hand

One robotic hand that tries to accurately mimic this kinematic model is the Anatomically Correct Testbed (ACT) hand [12] [9]. The aim of this hand design is to mimic the inner workings of the human hand as closely as possible and derive insight from this into human hand anatomy. Every tendon and muscles that is present in the human hand is present in this design. This causes the ACT hand to be very complex, relatively slow, and contain a lot of actuators to control the hand. Because of this the robotic hand is very impractical as a tactile device and also contains no sensors in its current form. Figure 4 depicts the hand.

Having two 2 DOF joints put so closely together in the thumb makes its design significantly more complex while space requirements are tight. This leads many hand designs to utilize underactuation to reduce the space requirements. In these types of hands not every DOF of each joint is independently controllable. An extreme example of this is the Open-source, anthropomorphic, underactuated (OSAU) robot hand [17]. This hand has four fingers with each have three 1 DOF joints and a thumb with two 1 DOF joints. All these joints are controlled using only a single servomotor. The user can control which joints are affected by the motors movement by an internal locking mechanism. Underactuated hands such as the OSAU

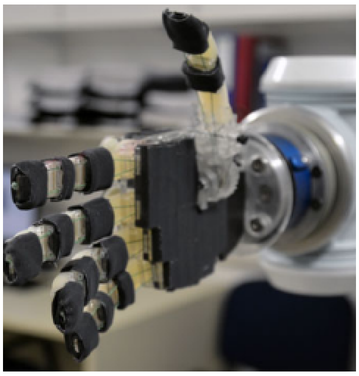


Figure 5: the Open-source, anthropomorphic, underactuated robot hand

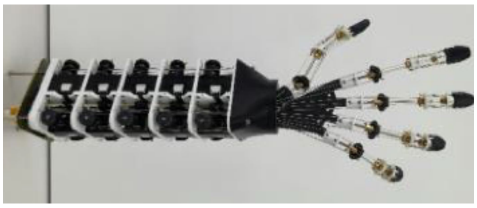


Figure 6: the Cable-driven Anthropomorphic robot hand

hand are often aimed at making prostheses more affordable. Using fewer actuators reduces cost and also reduces weight, which are two of the most important factors for prostheses designs. Because of its extreme form of underactuation this hand misses the dexterity required for tactile hand. Figure 5 depicts the hand.

A less drastic example of an underactuated hand is the Cable-driven Anthropomorphic (CDA) robot hand [21]. This hand tries to keep the kinematic model relatively true to the human hand. The only exception is the MCP joint of the thumb. The thumbs second 2 DOF is in this case replaced by a 1 DOF joint. The underactuation in this hand can be found in the two outer 1 DOF joints of each finger (including the thumb). Each fingers two outer joints are controlled by a single servomotor which makes them not independently controllable. While generally underactuated hands should be avoided as tactile devices, the mechanic that is used to control the two outer joints using a single servomotor could be acceptable is some tactile applications. Figure 6 depicts the hand.

The highly biomimetic anthropomorphic (HBA) robotic hand is another type of underactuated hand. It has 4 DOF for each fingers but actuates this with only 10 servomotors. Only two servomotors are used to control the little finger and the ring finger. Both the index finger and the middle finger have two servomotors that control flexion/extension, while a third provides coupled movement at the 2 DOF joint. The last three servomotors are used for the thumb. The hand is shown in Figure 7. While this type of hand is shown to be able to grab large variety of objects, it is quite limited in the way it is able to grab them. The great amount of underactuation would in this case not make it a robotic hand suited as a tactile device.

Replacing the thumbs 2 DOF joints by a 1 DOF is not exclusive to underactuated hand design. This simplification of the human hands kinematic model is common with a lot of recent robotic

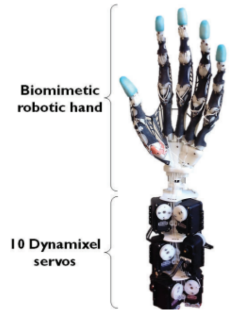


Figure 7: The highly biomimetic anthropomorphic robotic hand

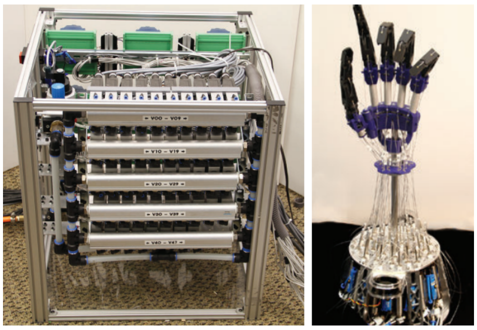


Figure 8: The Low-cost and Modular, 20-DOF Anthropomorphic Robotic Hand

hand designs. It can often be done without drastically impacting the performance of the hand because the thumbs MCP and CMC joints often move together. One such an example is the Five-fingered (5F) robot hand [27]. This designs thumbs similarly to the CDA hand has a 1 DOF MCP joint at the thumb, making the joints that make up each finger similar. Each of the 4 DOF of each finger is fully actuated by total of 20 ultrasonic motors.

The Low-cost and Modular, 20-DOF Anthropomorphic (LMA) Robotic Hand [25] is fully actuated robotic. It as its name suggest has 20 DOF with each of its five fingers having 4 DOF. This design is when compared to other devices very low cost. The cost of materials are approximately 100 USD, however this does not include the most costly parts such as the actuation system not the tactile sensing system it includes. This brings the cost to around 500 USD. From a tactile device point of view this device is relatively fast as it able to open and close fingers at around 3 Hz, fully actuated, and relatively low cost. The only thing that is unclear is whether it is dexterous enough to grab various objects. Another disadvantage of this system is that it is relatively large as it uses pneumatics to actuate the joints. Because of this the hand cannot easily be attached to an arm. Figure 8 shows the robotic hand and its actuation system.

Another example of a fully actuated robot hands is the Dexterous Anthropomorphic Robotic Typing (DART) hand [23], a robot hand designed to type on keyboard. This hand while not being designed as tactile device moves relatively fast and accurate. However it lacks the sideways movement of each of the basic fingers and does not contain any tactile sensors. Not having sufficient dexterity and mostly being designed for a single purpose makes this hand in its current state not suited for tactile purposes.

The KI-TECH Hand [18], a four fingered hand. Which each finger having 4 DOF. While its

Robotic hands	Actuation/ Transmission	Cost	Grasp Speed	actuated joints
Shadow hand [6]	air muscles, tendons, springs	>$ 100 000	< 1 second	underactuated: 20/24
ACT hand [9]	motors, tendons	no data	> seconds	Anatomically correct
OSAU Hand [17]	motors, tendons, springs	$ 200	no data	underactuated: 1/13
HBA hand [26]	motors, tendons	no data	no data	underactuated: 10/20
CDA hand [21]	motors, tendons	> $1500	> seconds	underactuated: 15/20
LMA hand [25]	pneumatics, tendons	> $400 USD +actuation	no data	fully actuated: 20/20
5F Hand [27]	motors, tendons, springs	no data	200 ms	fully actuated: 20/20
DART Hand [23]	motors, tendons	no data	350 ms	fully actuated: 16/16
KI-TECH Hand [18]	motors, tendons	> $2000	no data	fully actuated: 16/16
DLR-Hand [11]	motors, gears	no data	no data	underactuated: 13/16
Gifu III [22]	motors, gears	no data	no data	underactuated: 16/20
Robonaut hand [20]	motors, flex shaft	no data	no data	underactuated: 11/20

Table 2-1: A comparison of some other anthropomorphic robotic hand designs

speed is unknown it is shown to have great dexterity while grabbing a variety of objects. It is however quite an expensive hand with each of its actuators costing over 100 USD. This leads a hand that costs more than 2000 USD.

Table 2-1 summarizes the discussed robotic hands and lists some more that will not be discussed in detail. In general it is concluded that most anthropomorphic hand designs tend to be underactuated which is undesirable for tactile devices as often lack the dexterity to move like a human hand. Most fully actuated hands that are suitable as tactile devices are often very expensive and often lack data concerning their speed.

Chapter 3

Towards designing an actuation and transmission system

At the core of designing a robotic hand lies the implementation of the movement of its joints. In human hands the forces that move the joints are actuated by muscles. These forces are then transmitted through tendons to their respective joints. This chapter will first deal with some anatomy of the human hand and more importantly the movements it is capable off and then go into detail how the movement is achieved in this robotic hand design, or more specifically in a single finger. This designed finger can later be combined with a second finger and a hand palm to form a two fingered hand. The made design choices that ensure all required joint angles can be reached will be extensively discussed here.

The discussion of the robotic hand design will start off at the tip of the finger, from where it will built up joint by joint. This single finger will be modelled after the basic finger, having three joints. The outer two joints will be 1 DOF joints. The inner joint that connects to the base of the hand, will be a 2 DOF joint. The joints of this finger (starting at the tip) will be referred to as DIP, PIP, and MCP joints, respectively. While this type of finger resembles a basic finger, such as the index finger, the same model will be used for the thumb as well. By increasing the ranges of motion, specifically at the MCP joint (which is connected to the handpalm), the dexterity required for various tasks can still be realized. This is quite similar to other hand designs as discussed in Section 2-1-2.

3-1 Anatomy of human hand

The human hand consists of 27 bones which form the hand and its fingers, as well as the wrist (Figure 9). This relatively high number of separate parts make the hand complex a highly versatile and complicated human body part. The main focus of this project is the movement of the hands fingers of which there are five. The four basic finger consisting of the index finger, middle finger, ring finger and the little finger share a lot of similarity in structure as well as movements. The fifth finger, the thumb, has a significantly different structure than the other four, which allows it more movement. This additional movement is vital to the unmatched dexterity of the human hand, allowing humans to grab and handle objects of all kinds of shapes in a precise manner.

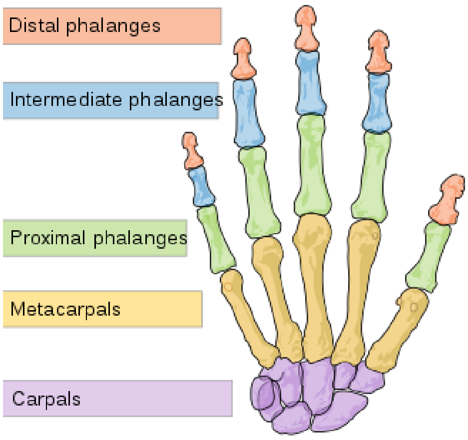


Figure 9: The Bone structure of the human hand. For the basic fingers the distal, intermediate, and proximal phalanges, are the main moving parts. For the thumb that lacks an intermediate phalanx, the metacarpal is also able to move.

3-1-1 Joints of the fingers

At the core of the human hand lay the bones. These rigid parts offer stability and support for the other parts and function as anchor points for tendons and muscles. The attached tendons and muscles provide the actual moving forces of the human hand, but the structure of the bones and more precisely its joints (connection between different bones) dictate the allowed movement planes. Two types of joints that are commonly found in the human body are the hinge joint, and the ball and socket joint. Figure 10 illustrates these two types of joints. Because of the two axes of rotation allowed by the ball and socket joint, the joint is identified as a 2 Degree of Freedom joint (2 DOF). The hinge joint is an example of 1 Degree of Freedom joint (1 DOF), since it allows rotation only along a single plane.

The five fingers that make up the hand can be categorized into two groups: First the four basic fingers, containing the index finger, the middle finger, the ring finger, and the little finger, and the second group, the thumb. The four basic fingers share a similar structure with similar joints. Each of them consists of three joints. Starting where the finger is connected to the hand palm the first joint is the Metacarpophalangeal joint (MCP joint). This joint is a 2 DOF type joint, allowing rotation in two directions. The second joint is Proximal interphalangeal joint (PIP joint). This joint exhibits only a single DOF. The basic fingers third joint is the Distal interphalangeal joint (DIP joint). This joint is very similar to the PIP joint and also is a 1 DOF joint. Figure 11 shows the location of these joints.

When looking at Figure 11, the thumb seems to have a joint fewer than the basic fingers. It has only a single IP (interphalangeal) joint instead of two. To make up for this the thumbs movement is highly dependent on a third joint: the Carpometacarpal joint (CMC joint). While this joint is also present in the four basic fingers, the small amount of movement that

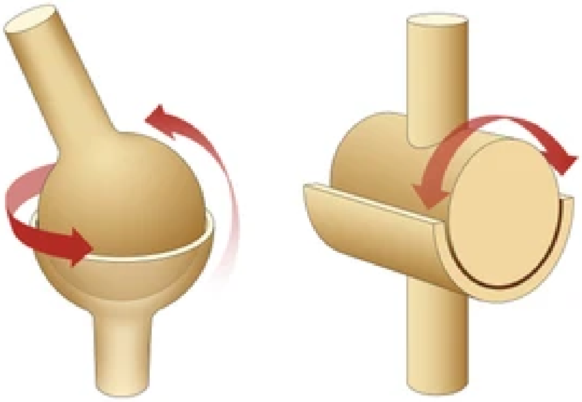


Figure 10: The Ball and Socket Joint shown on the left allows rotation in two direction, whereas the hinge joint (right) allows rotation along a single axis.

it allows plays a negligible role in the basic fingers movement. In the thumb however this is the joint that connects the thumb to the handpalm. The thumbs CMC joint is most often identified as a 2 DOF joint, allowing rotation in two directions. The joint however also allows some axial rotation which some consider to be third DOF. This third rotation is dependent on the rotation of the two other rotations and the thumbs CMC joint is thus strictly speaking not a 3 DOF joint [19]. The thumbs second joint is the Metacarpophalangeal joint (MCP). This joint similarly to the basic fingers MCP joints is a 2 DOF joint. Lastly the thumb has a single IP joint. Like the basic fingers IP joints the thumbs also exhibits a single DOF.

3-1-2 Movements of fingers

While the bone structure and the formed joints provide the support and allowed movements of the fingers, it is the attached muscles and tendons that actuate these movements. Muscles, which are the producers of forces on the joints, are only able to exert a pulling force in single direction. They do this by contracting. When the muscle relaxes it exerts no forces upon the joints it is connected to. Thus to be able to rotate a 1 DOF joint back and forth requires at least two muscles. One of the muscles task is bend the joint (flex), while the other is needed to straighten the joint again (extend). Each of these muscles that provide opposite forces on a joint is called an antagonistic muscle pair. Muscles that make up an antagonistic muscle pair are often connected to the same bone on opposing sides. Exactly where the muscle is connected to the bone differs between individuals and has significant effect on the Range of Motion (ROM) and how effective the transmitted force is in rotating the particular joint.

In most cases muscles are connected to bone through bands of dense fibrous connective tissue with high tensile strength, called tendons. In some cases this allows the relatively large muscles to be located elsewhere than directly next to the connected joint. For the muscles that actuate the hands movements a significant number of so called extrinsic muscles are located in the forearm instead of the hand directly. This allows the muscles that control the hand to be larger have have more power than if they were to fit inside the hand. While having a general idea of how the hands movements are realized is important, discussing the exact muscles and tendons involved in moving the hand is outside the scope of this project. The

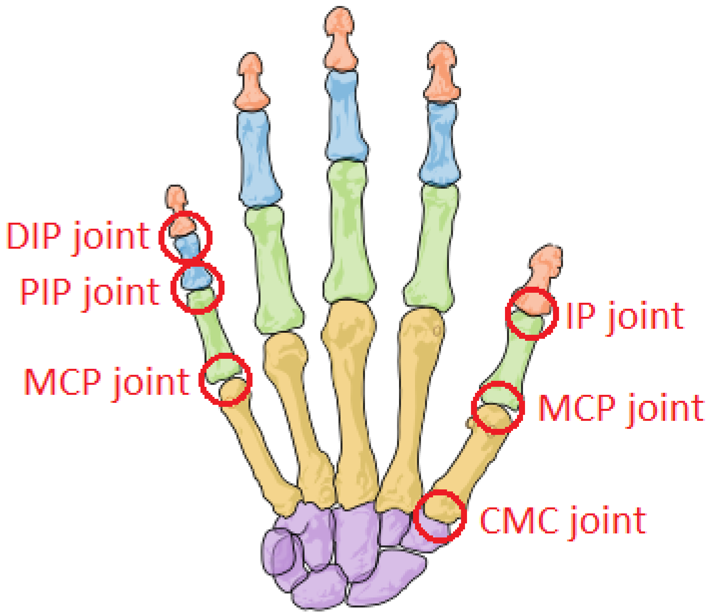


Figure 11: The joints of the hands fingers.

main aim is replicating the resulting movement, how this movement is achieved is of little importance.

To replicate the human fingers movement analysing each joints Range of Motion (ROM) is most important. A joints ROM (Range of motion) is measured in degrees and captures how far the joint is able to rotate in each direction. In 1 DOF joints rotation we consider two types of movement: flexion and extension. Flexion is the movement that bends or flexes the joint. Extension is the movement that straightens or extends the joint. For 2 DOF joints two additional movements are identified: adduction and abduction. Adducation is usually defined as movement towards the body or another body part, abduction instead as moving away from that body part. For example, spreading your basic fingers apart would be considered abduction, bringing them together, adduction.

Range of motion of fingers

As mentioned the basic fingers structure is very similar. Each has three joints of which two are 1 DOF joints and one is 2 DOF. The joints ROM when comparing separate fingers varies slightly, with the index finger having the smallest ROM which increases with each finger as we move to the little finger [19]. Starting with flexion/extension at the MCP joint the average ROM for the four basic fingers is 109°, 108° for the PIP joint, and 90° [10]. The exact ROM of fingers can be found in Figure 12. For abduction/adduction at the fingers MCP joint ROM was found to be around 20°. It should be noted that abduction/adducation ROM at the MCP joint of the basic fingers is largest when the fingers are extended and restricted when the fingers are flexed [19].

The thumbs first 2 DOF joint, the CMC joint, allows for the greatest ROM: 75° of flexion/extension and 72° of abduction/adduction. The MCP joint exhibits 65° of flexion/extension

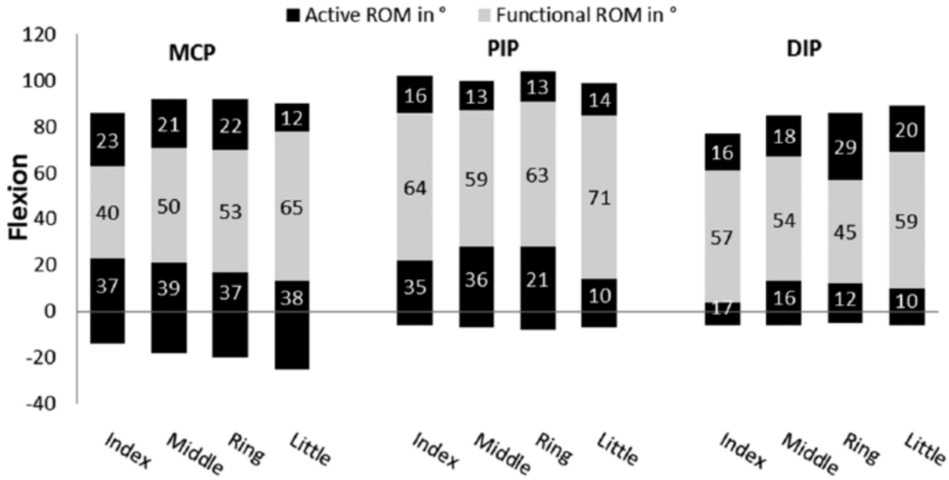


Figure 12: ROM of the fingers, taken from [10]

and 52° abduction/adduction. The thumbs single IP joint allows for approximately 135° of extension/extension. This large ROM of this IP joint is mostly due to great amount of hyperextension that the thumbs IP joint allows [16]. The results are summarized in Figure 13.

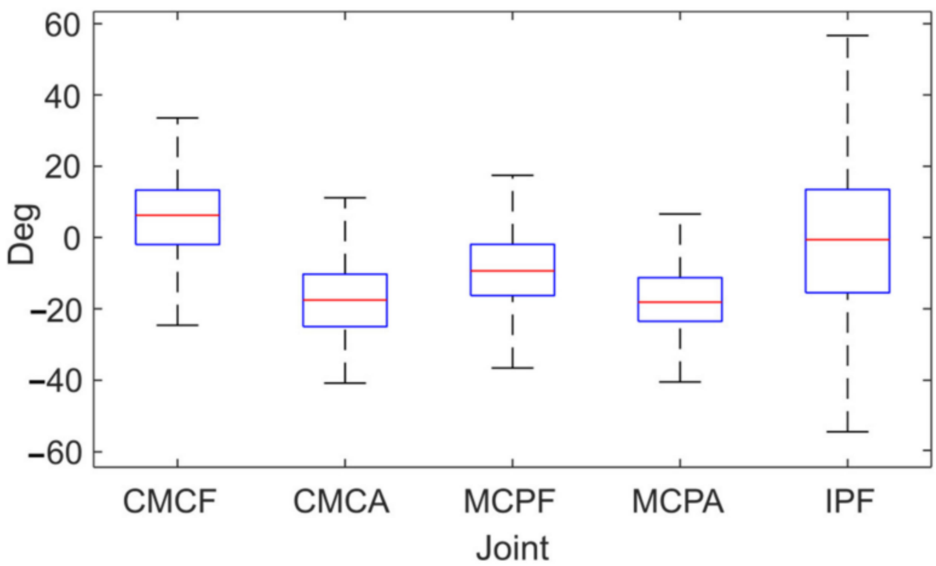


Figure 13: ROM of the thumb, taken from [16]

3-2 Actuation methods

As discussed in the previous chapter 2, there are various ways to actually move the fingers joints. The most common of which is to use tendons to transmit the forces caused by servomotors to the relevant joints. The reason this approach is so common, is because the relatively cheap and easy to operate servomotors do require some space. Putting them inside the actual joints thus becomes a major hassle when the fingers are designed. On top of this, if one of the servomotors breaks, replacing it often means the whole finger needs to be disassembled. By placing the motors either in the hand palm or the forearm and transferring their forces to the relevant joints using tendons, valuable space is saved in fingers. In the design of this robotic hand tendons are opted for as well. The chosen motors are placed in the hand palm of this design because designing a wrist or forearm is not part of this hand. This also enables the option to attach the hand to another robotic arm.

3-3 Kinematic model of the robotic hand

The most commonly accepted kinematic model of the human hand is one where each of the four basic fingers is represented as having three joints. The two outers joints have a single DOF, while the inner one is a 2 DOF joint allowing side wards movement as well. The thumb also has three joints but only the outer one is a 1 DOF joint. The two inner joints have 2 DOF. In Section 2-1-2 is shown that many other anthropomorphic hand designs simplify one of the thumbs 2 DOF joints to a 1 DOF joint. These robotic hand designs have shown that this can be done with negligible loss of range of motion. The complexity of designing a thumb with two 2 DOF joints right next to each other lead us to omit one of its DOF as well. This means the basic fingers and thumb now have the same kinematic model and their design can thus be similar. Figure 14 shows the kinematic model our robotic hand. The middle, ring, and little finger are shown in grey because in this project they will not be part of the final design. This is only a first version of the hand and the similarity of the index finger to the other basic fingers makes reusing the basic fingers design relatively easy. To enable the addition of more fingers in the future fingers will be designed in a way that allows them to easily be placed next to each other.

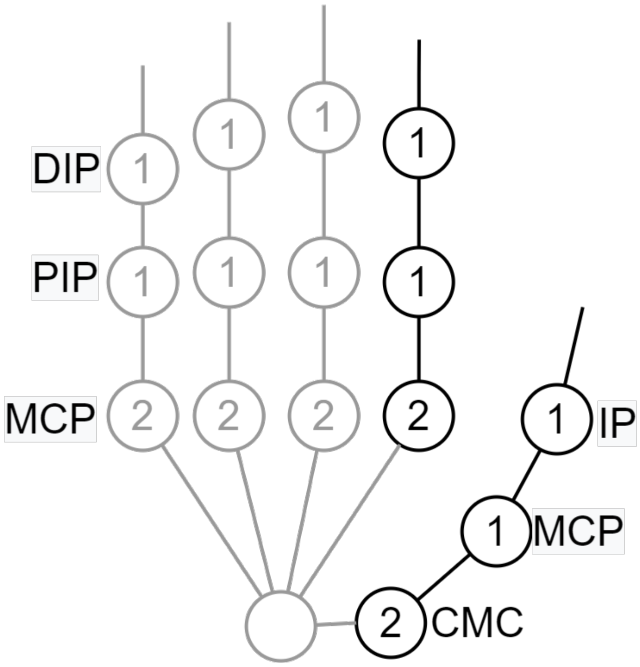


Figure 14: The kinematic model of our proposed robotic hand. Compared to the generally accepted anatomical model of the human hand, the thumbs MCP joint has only 1 DOF instead of 2.

3-4 A single 1 DOF joint

We start off the design of a single finger at the tip. The first joint we encounter here as we travel towards the hand palm is an IP joint. This is a joint that has one degree of freedom, allowing flexion and extension.

As discussed previously, we use tendons to rotate joints. These tendons similarly to the human body can only exert forces upon joints by pulling them. Thus to be able to rotate a joint in both directions requires at least two tendons. We name a pair of tendons that controls opposing motions of a joint an antagonistic tendon pair similarly to its name in human anatomy. For the first IP joint this means two tendons need to be connected, one that when pulled causes the joint to flex (bend) and one that when pulled extends the joint. It is important to note that when either tendon is pulled that antagonistic tendon need be released.

3-4-1 First look at a naive 1 DOF joint implementation

Limiting the design to just what happens at this joint, we create out first naive implementation of flexion and extension at the IP joint. This is shown in figure 15. It shows two tendons: the red flexor provides flexion when pulled, the blue extensor provides extension when pulled. Each tendon is fixed to the outer phalanx of the finger. This is marked with a dot. At the bottom the dot marks a hole through which the tendon is routed and is allowed to slide through back and forth.

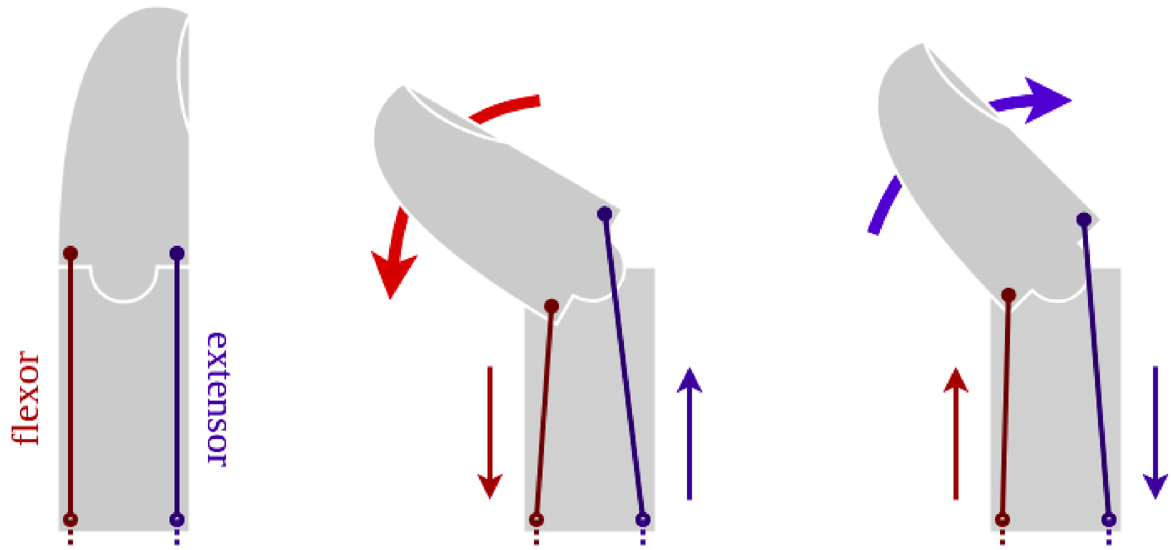
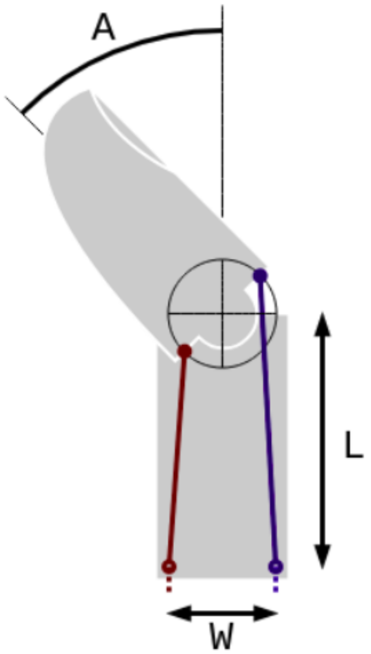


Figure 15: A basic implementation of the movement of a single DOF IP joint using two tendons: Two tendons are fastened at the upper phalanx of the finger and at the bottom of the finger the tendons can be pulled and released. To flex at the joint, the flexor (red) should be pulled and the extensor (blue) released. For extension, the extensor should be pulled and the flexor should be released.

As expected an antagonistic tendon pair is able to provide the 1 DOF motion at the IP joint. When we pay closer attention to the tendons when the IP joint is rotated, we see that the length of tendon from the previous joint (at the bottom) to the fixed end point (upper dot) changes as the joint rotates. When the IP joint is flexed, the flexor length is shortened, while the extensors is increased. When the joint is extended, the length of flexor from one point to the other increases and the extensors decreases.

The lengths of tendons changing is obvious and does not by its self cause a problem. However when we actually delve deeper and calculate by how much each of the tendon lengths change when the joint is rotated, we notice that both tendons shorten and lengthen by different amount. Figures 16 and 17 illustrate this effect. Equations 3-1 and 3-2 present in Figure 16 relate the lengths of the flexor and extensor tendons to the joint angle for angles between 0° and 90° and Figure 17 shows by how much the lengths of tendons change as the joint rotates.

At the extended position where the joints angle is rotated 0°, the total length of tendon from one end to the other will be shortest. When the joints angle is at 90°, the tendons combined length will be the largest. While this change of total tendon length is relatively small it can still have a negative effect on the joints rotational speed and accuracy. Consider that the tendons that control a joint have to be tensioned. If a tendon is not under tension it can not immediately transfer its pulling force to the joint. The slack must first be pulled out of it, which decreases the rotational speeds whenever the joint is rotated in another direction. Having tendons be slightly loose also means the joints angle is more prone to external forces. Any disturbance can slightly move the joint until one of tendons tightens and prevents moving further. This decreases the accuracy of the joint in question. Thirdly there also is the chance that the joint angle may overshoot when the joint should be moved to a specific angle. When this happens repeatedly oscillations may occur causing the joint angle



- W, distance between tendons
- L, length of tendons when fully extended
- A, angle from fully extended
- L_F, flexor length
- L_E, extensor length

For $0 < A < 90$:

$$L_F(A) = \sqrt{(L - \frac{W}{2}\sin A)^2 + (\frac{W}{2} - \frac{W}{2}\cos A)^2} \quad (3\text{-}1)$$

$$L_E(A) = \sqrt{(L + \frac{W}{2}\sin A)^2 + (\frac{W}{2} - \frac{W}{2}\cos A)^2} \quad (3\text{-}2)$$

Figure 16: Equations 3-1 and 3-2 relate the lengths of the flexor and extensor tendons to the angle of the joint between $0°$ and $90°$.

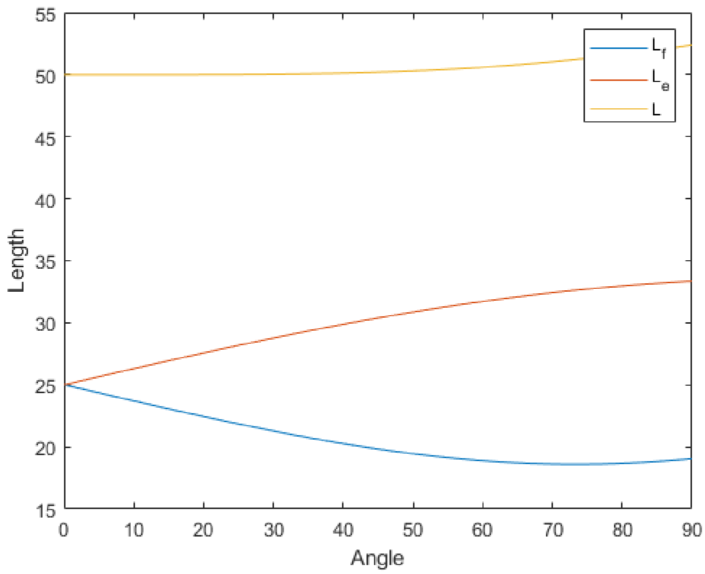


Figure 17: Using Equations 3-1 and 3-2 and values of 25 mm and 15 for L and W respectively, the total tendon length increases from 50 to 52.39 mm as the joint rotates 90 degrees. This change in length translates to loose tendons. Loose tendons cause slower joint movement and less accuracy.

to never become stationary.

Thus in order to prevent these cases tendons have to been under tension under all circumstances. This is not the case with the design of Figure 15. There are ways to tension the tendons even when we cannot guarantee the length of the tendons stay constant. This involves using such as using springs or having separate servomotors to control each joint. However both involve using extra parts in an already restricted space and springs will still cause tension change on the tendons. Having too much tension on a tendon can increase the friction between part of the finger and the tendon, thus slowing down the speed of the joints. Ideally a way should be found to keep the combined tendon length irrespective of joint angle.

Another disadvantage of routing the tendons in this way is that the amount of force transferred to rotate the joint will also be decreased as the joint rotates. As we can see in figure 16, the tendons will only be tangential to the rotation when the joint is fully extended at 0°. However when it rotates the tendons will no longer pull at the right angle and the pulling force of the tendons will not be efficiently converted into rotation. This also will cause slower rotation of the joint.

3-4-2 Guiding along pulleys

To circumvent the problems discussed in subsection 3-4-1, we need to make sure the combined tendon length of the flexor and extensor are constant and thus do not depend on the joint angle. On top of this we desire the angle at which the tendons pull the joint to always be tangential to the joint. This ensures the pulling force is efficiently converted into rotation at the joint.

The easiest solution that solves both issues, is to guide the tendons along pulleys. Instead of simply fastening the end point of a tendon to the outer phalanx, the tendon is first (partially) wound around this pulley. Figure 18 illustrates this. As we can see, when the finger is fully extended, the extensor is completely straight, but the flexor is partially wound around a pulley. When the joint is flexed, we see the extensor now starts winding around the pulley, while the flexor is being unwound. By making sure the tendon is tangential to the pulley for all allowed joint angles, we have made the combined tendon lengths independent of joint angle for these allowed joint angles. In figure 18 the flexor is wound approximately 90° around the pulley when the joint is extended. That means from this position the joint is allowed to flex up until 90° since for these angels the tendons will be tangential to the pulley. If a greater range of motion at this joint is desired, the tendon should be wound around the pulley longer. This is exactly what was done in the final design of the outer joint which is discussed in more depth in Chapter 4.

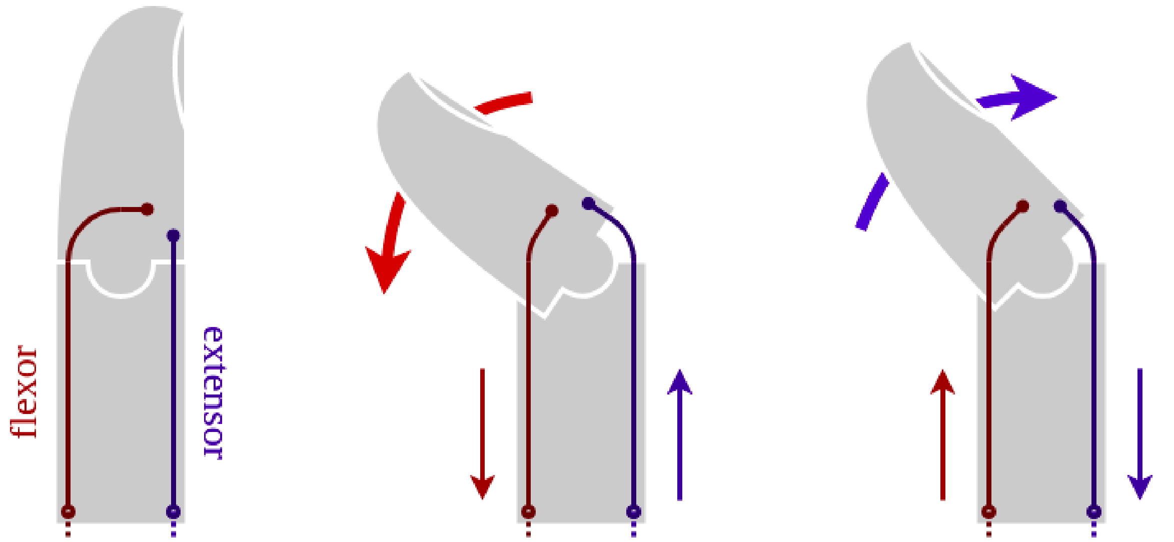


Figure 18: In this implementation of a single DOF joint tendons are guided partially along a pulley located at the joint. By doing so the combined lengths of the flexor and extensor tendons does not change during rotation of the joint as long as the tendons remain tangential to the pulley. This design makes sure the tension on the tendons does not change during rotation.

3-5 A second 1 DOF joint

After having implemented the first IP joint and discussed its working, we move to the second joint. This again is a 1 degree of freedom joint, allowing flexion and extension.

Adding this joint to our scope, complicates the design significantly. The first thing we have to consider is that these additional tendons require space. Note that the tendons that control this second IP joint, are not the only ones that have to cross this joint. The tendon pair that was discussed in the previous section (3-4) needs also be routed along. This extra space is accounted for by routing the new tendons through another layer than the one the previous tendon pair was routed through. Doing this guarantees the tendon pairs do not cross, avoiding unnecessary friction. Each tendon having a different route to its joint also makes assembling the actual parts and routing the tendons easier. But before we delve into the details of how the added tendons are routed through the hand, first is discussed how many tendons are actually needed to to add independent control of second joint.

3-5-1 Required number of tendons and servomotors

In general when designing a tendon actuated robot, the number of tendons should be at least 1 more than the number of degrees of freedom in the system. If the number of tendons is less than this, the system is considered to be underactuated. This means not every degree of freedom can be controlled independently. Since independent control over each of the fingers four degrees of freedom is desired, at least five tendons are required for each finger. The minimum of required tendons for each DOF is only discussed now because in a single DOF joint it is easily seen that at least two are needed to realize rotation in both directions. Now, with the addition of a second joint comes the possibility to opt for adding only a single tendon.

Having the fewest amount of tendons inside the finger may at first sight seem like the most desirable option, however as always this comes at the cost of other things. In this case using three tendons to actuate two joints requires us to control each of the tendons individually. Consider Figure 19. To move the outer joint flexor 1 and the extensor need to both move. To move the inner joint flexor 2 and the extensor have to move. In this setup the extensors length influences two separate joints.

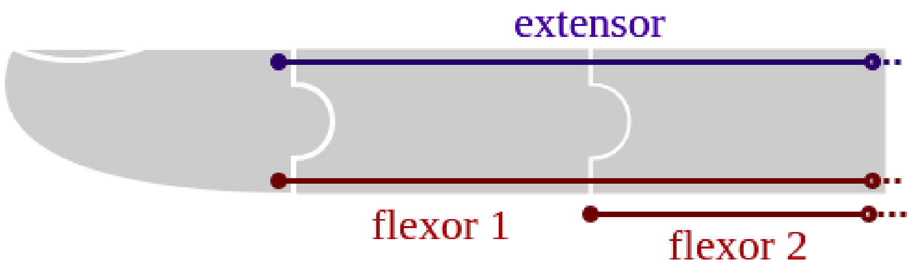


Figure 19: Using only three tendons to control two joints: flexor 1s length affects the outer joint, flexor 2s length affects the inner joint, and the extensor affects both joints

It is at this point that what actually moves the joint should be considered. These are the servomotors. If having the least amount of tendons is desired separate control of each of tendons is needed. The easiest solution to achieve this is to use three servomotors: one for each tendon. This is quite similar to what happens in the human hand, where a muscle is only connected to single tendon and is only able to exert force in a single direction. This is shown in figure 20. Now when the outer joint needs to flex, flexor 1 can be pulled, and the extensor released.

And afterwards the inner joint can flexed by pulling flexor 2 while releasing the extensor even further. When only looking at the tendons this is a really elegant solution, however is it worth adding another servomotor and more complexity to the system? Simply having four tendons: two controlling each joint, means a single servomotor for each tendon pair. Note that servomotors are most often the most expensive parts of the such robot.

However there is another solution that limits the amount of servomotors required to two, whilst also enabling the use of only three tendons. Figure 21 shows how this can be achieved. Instead of each servomotor affecting only a single tendon, each of the two servomotors now affects the length of their flexor tendon, but also the extensors. The extensor tendon is now routed through the middle of servomotors rotating part and attached somewhere to the base of the finger. Rotating each servomotor causes their own flexor to be affected, but also the common extensor. If the extensor can slide through the middle part easily, such a system consisting of three tendons and two motors will work. It should be observed however that in this design rotating a servomotor affects the tendons by different amounts. When either servo rotates the extensors affected length will be twice the amount of the flexors. This is the case because the extensor in Figure 21, is wound or unwound at both the top and the bottom, while the flexor only at the bottom. This effect can be compensated for at the joints location. By routing the extensor along a pulley that is twice as large as the flexors pulley, the extensors affected length being twice as much as the flexors is negated.

The design was in fact tested, and its major flaw was immediately found: this design adds too much friction to the tendons. Figure 21 illustrates clearly how the extensor is routed

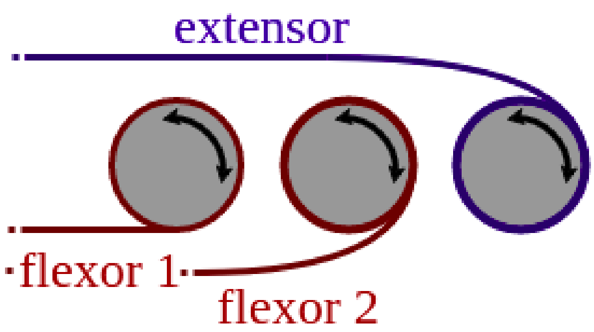


Figure 20: 2 joints are controlled by 3 servomotors (shown as grey circles from the top). Each servomotor has a single tendons wound around it. Flexor 1 affects the angle of the first joint. Flexor 2 affects the angle of the second joint. The single extensor tendon controls the combined angles of the two joints. Pulling and releasing tendons is achieved rotating the servomotor one way or another to wind them around or unwind them.

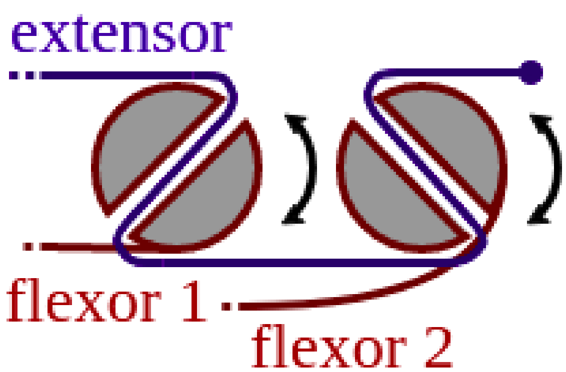


Figure 21: Two servomotors. Each one is connected to a single flexor similarly to Figure 20. However a single extensor tendon is routed through the servomotors rotating piece where it is allowed to move freely within. This ensures both servomotors are able to pull and lengthen the extensor.

through the rotating parts of the servomotors at sharp angles. This angle, specifically where a tendon enters or leaves a servomotor head, causes a lot of friction. This friction causes the whole tendon to move slower, or worse may cause the tendon to snap.

Because of these reasons these more complicated solutions where abandoned. Instead the simplest design was chosen: each joint controlled by a tendon pair, and each tendon pair being controlled by a single servomotor. This greatly simplifies the system design as now each actuator controls only a tendon pair that is connected to a single joint. And there is no complex relationship or control necessary between the different actuators and joints.

3-5-2 First implementation of the second 1 DOF joint

In the discussion of the implementation of the first IP joint (Section 3-4), we concluded that tendons need to be routed along pulleys. Doing so ensured the total length of the tendon pair did not change when a joint is rotated.

Logically the same ideas are applied at this joint. The second joint is added and each of the tendon pairs is routed along its own pulley. Figure 22 shows how the 2 tendon pairs are routed, when both joints are completely extended.

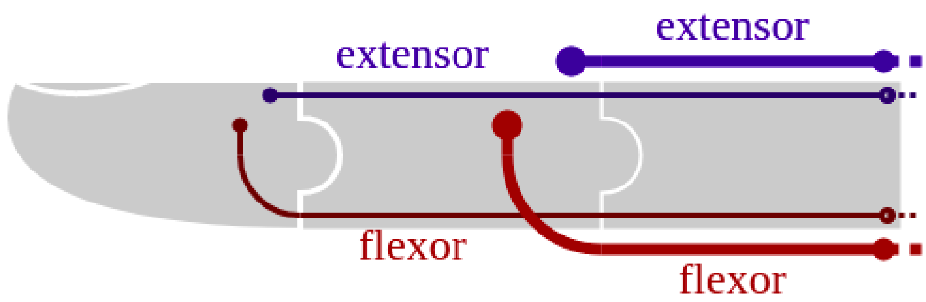


Figure 22: Both tendon pairs are routed along a second pulley. The thinner tendon pair controls the outer joint. The thicker tendons control the inner one. (Note that tendons are not actually different size nor are they routed on the outside of the finger, this representation is just for clarity.)

As before when the finger is fully extended (both joints are straight), errors are not easily spotted. When the newly added joint is moved however, an issue can be spotted. The issue is not with the newly added tendon pair, but arises because of how the already present outer joint tendons interact with rotation at an earlier joint.

When routing tendons as shown in Figure 22, there are two possibilities. The left side of figure 23 shows what happens when there is nothing to keep the outer joint flexor in place. The right side shows the case where there is something that keeps the flexor from exiting the finger. Both cases however result in the total length of the tendon pair changing when the joints rotates.

Like before the only way the total length of a tendon pair can be kept independent of joint angles is by ensuring the tendons are routed along pulleys for all allowed joint angles.

Our presented solution to this problem at the previous joint (Figure 18) relied on the fact that the flexor and extensor tendons are wound around the pulley even when the finger is rotated.

In this case the solution however is not as simple because the tendon actually needs to be routed from one pulley to the next, whereas at the first joint it did not matter where the other tendons went as long as it was attached somewhere to the phalanx. To solve this problem the flexor tendon needs to be partially wound around the pulley when the finger is fully extended. In the next subsection some solutions are discussed.

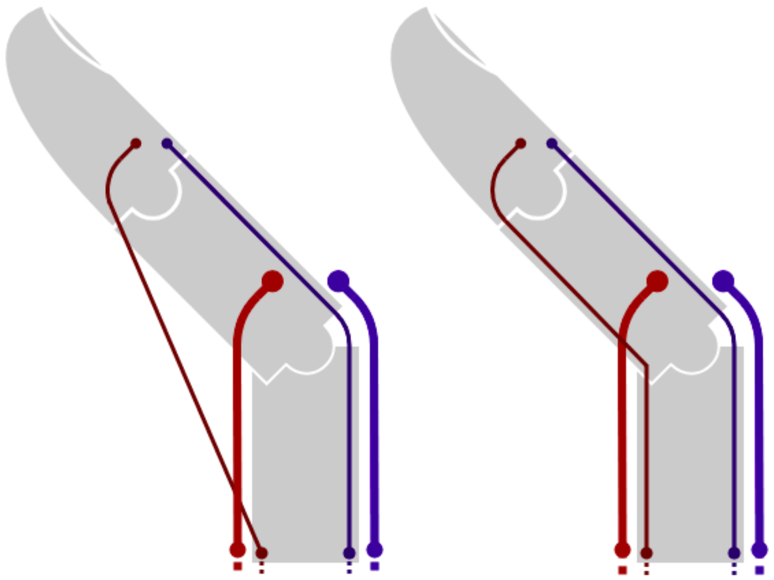


Figure 23: Using the plain pulley approach of section 3-4 for a second joint. On the left the red flexor is assumed to not be locked into place. The tendon thus exits the finger entirely and the combined length of tendon pair will be affected by joint rotation. On the right the red flexor is held in to place, however rotation at the inner joint still causes the total length tendon to be affected. The length increase at the extensor scales linearly with the rotation, whilst the flexors does not.

3-5-3 Crossing tendons

The first solution that was considered, was to cross tendons between each joint. This is illustrated in Figure 24. The left side shows the tendon routing in the fully extended position and the right side when the second joint is rotated 45°. As long as both tendons remain in contact with the pulley tangentially, tendon pair length will be unaffected by joint rotation.

However this solution does not guarantee the joint is able to rotate the required range of motion. For the crossing tendon solution be able to rotate 90° for example, certain geometric conditions need to be met. In this case the allowed rotation at the joint without causing change in total tendon pair length is dependent on the pulley diameter, W, and the length between the pulleys, L. The relationship between the geometry and maximum angle, a_{max}, is given by equation 3-3. So to be able to reach an angle of 90°, $W \geq \frac{L}{2}$. This means the pulleys have to literally be touching for this work. Even if the condition of Equation 3-3 can be met, for example if the flexing angle is significantly smaller than 90°, there is another problem with this solution.

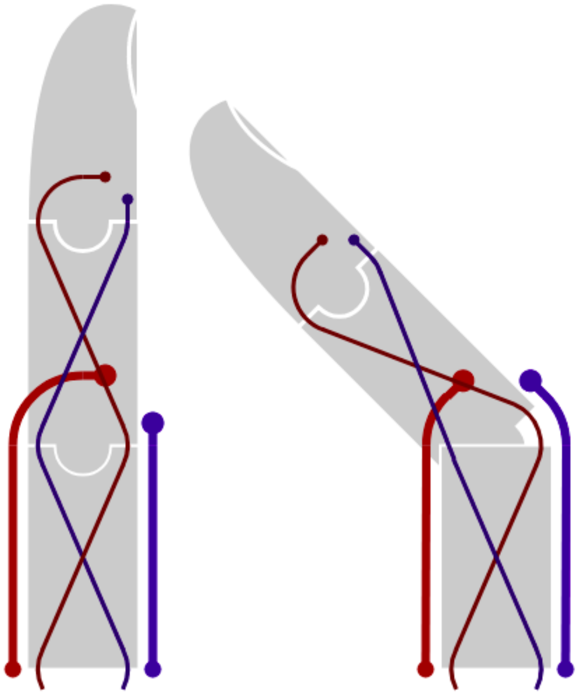


Figure 24: Tendons are crossed between each joint.

$$a_{max} = 2\arcsin\frac{2W}{L} \tag{3-3}$$

The actual issue of this approach is really the fundamental part of the solution itself: the tendons have to cross each other. If the tendons cross and they touch each other, the tendons will cause additional friction on each other, which is undesirable, and even worse they may even cause them to tangle. Of course the easy solution to this problem is to have the tendons not be in the same layer. While this may seem as a reasonable solution, keep keep in mind that in that case an additional layer is needed for each tendon. which in the case of two IP joints already means four layers. This is undesirable, as it would increase the size of the finger needlessly.

3-5-4 Routing the flexor next to the extensor

The second solution to the issue of tendons leaving the finger, is to move the entire flexor closer to the extensor. By routing the tendon in this way the tendon will always be partially wound around a pulley. This is shown in figure 25. Only the tendon pair that controls the outer joint is shown here. As can be seen the flexor is guided sufficiently close to the other side of the finger so that the tendon is wound around the pulley. As discussed before, if the joints need to be able to rotate 90°, the tendon should be wound around that joints pulley at least 90°. Only this ensures the combined tendon length of a pair remains the same throughout various joint rotations. This condition is satisfied here.

Figure 26 shows the two 1 DOF joints of this design, where both tendon pairs are shown in their own layer. In this figure the fingers joints have been rotated by various amounts and it

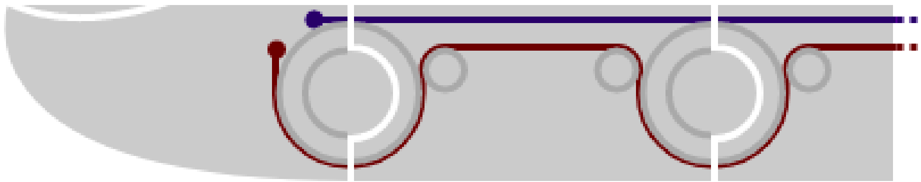


Figure 25: The flexor is guided closer to the other side of the finger. This ensures the flexor is always wound around the pulleys, even as the joint rotates. Only a single tendon pair controlling the outer joint is shown.

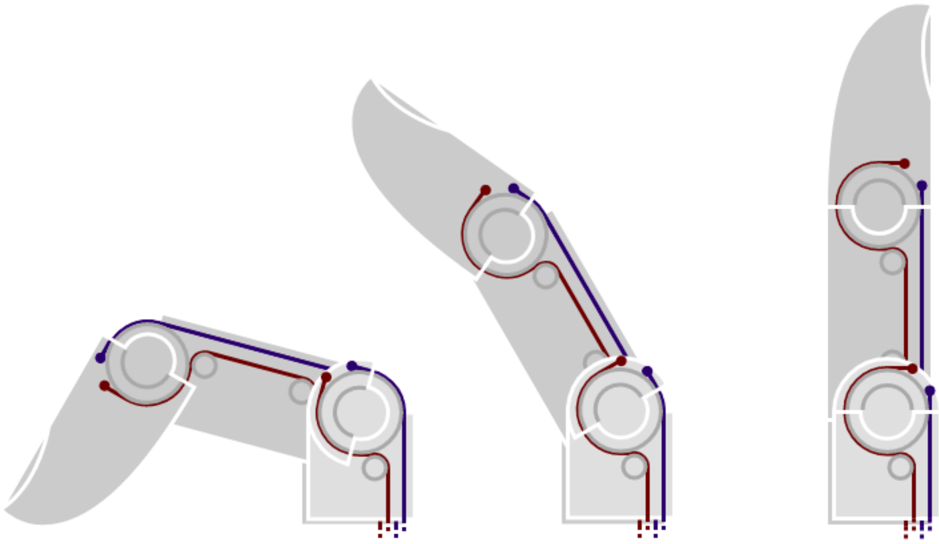


Figure 26: The finger consisting of two joints, is shown in various positions. The tendon pair in the back layer (darker colour) controls the outer joint, the tendon pair in the front layer (lighter colour) controls the inner joint.

is clear from the picture that for both joints, both tendon pair are always wound around the pulleys.

The only drawback of this design, is that the particular routing of the tendons causes more friction in some angles. From the extended position (right side in Figure 26), the red flexor is wound further around the pulleys than in the flexed position. This will likely mean it will be affected by more friction in this position than the other. This may cause a slight difference in rotational speed depending on the fingers starting position.

3-5-5 Joint angles affecting others

After concluding the routing of tendons for the two outer joints, how the servomotors affect the joint angles needs to be discussed in more depth.

As discussed in subsection 3-5-1 each tendon pair is connected to a single servo motor that controls the angle of the connected joint. One might expect a single servomotor to thus only affect the rotation of a single joint. This however is not the case. As can be seen in for example Figure 26, the angular position of the inner joints affects the routed path of the tendon pair

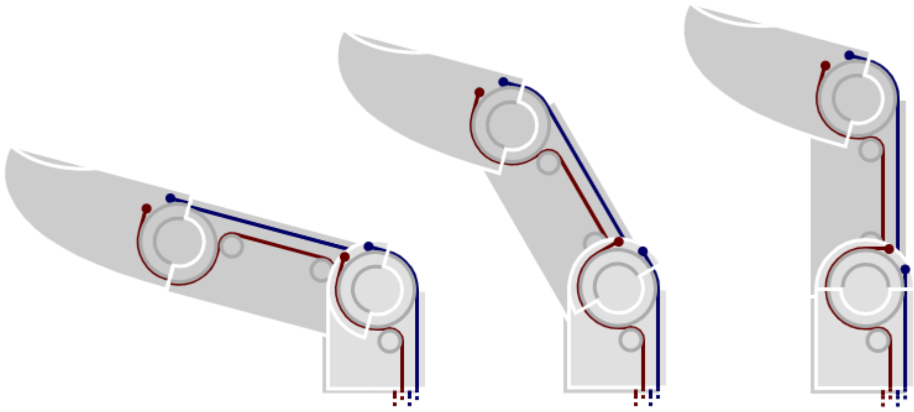


Figure 27: While the outer joint start off in the extended position, rotation of only the inner joint servomotor causes the outer joint angle to change as well. When the servomotor that controls the outer joint remains stationary, the angle between the outer phalanx and the palm of the hand remains unchanged.

that controls the outer joint. When the inner joint is flexed, the length of the outer joint flexor that is guided along the inner joint pulley, will be shorter than in the extended position. The opposite will be true for the outer joint extensor.

What one might have expected in this case is that the servomotors control the joint angle directly. By joint angle is meant the angle between the previous finger segment (phalanx) and the next one. However due to that servomotors being located inside the palm of the hand, a servomotor will control the angle between the connected phalanx and the hand palm instead. This effect is shown in Figure 27. Here only the servomotor that controls the inner joint is rotated. As the connected joints angle changes, so does the outer joints. This is because since the outer joints servomotor remains stationary, the angle between the outer phalanx and the base of the finger (where the servomotors are located) remains constant.

So with this setup if only a single joint needs to rotate while the other remain stationary, multiple servomotors need to be rotated. This may at first glance not appear to be a problem, however when the angular range that each joint is required to have is considered an issue becomes apparent. Let us assume each joint should be able to flex 90° from the extended position. Three joints make up each finger (the third will be discussed in depth later). Rotating the first inner joint by 90° is not a problem; all three servomotors are rotated 90°. Afterwards rotating the middle joint by 90° requires two servos to rotate another 90°. Lastly rotating the outer joint by 90° requires the connected servomotor to rotate another 90°. Thus the fully flex the finger from the extended position requires the outer joint to rotate a total of $3 \times 90° = 270°$ relative to the hand palm. The issue with this is that the connected servomotors have only a rotational range of about 180°. While the ratio between the pulley radius and servomotor radius can be changed to compensate for this, this also decreases the precision or resolution of the joints angles.

To combat this issue changing how the servomotors influence the tendons was considered first. This is very similar to the option that was discussed in Section 3-5-1 and shown in Figure 21. In this option the extensor tendons would be guided through multiple servomotors. This would result in multiple servomotors influencing the length of a single tendon. The same

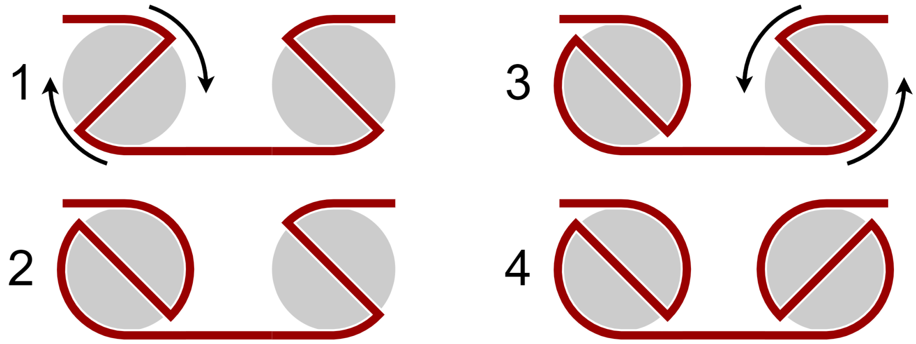


Figure 28: Two servomotors are shown in grey from the top. A single tendon is routed through the rotating piece of both servomotors. Both servomotors are able to pull and release the tendon:

1. The left servo rotates, causing the tendon to be pulled
2. The two servos after the rotation
3. The right servo rotates, causing the tendon to be pulled again
4. The two servos after both have rotated

could be done here to make sure no servomotor needs to rotate more than 180°. This is shown in Figure 28. Similarly to Section 3-5-1 however, this causes a lot of additional friction in the tendons which slows them down significantly and causes wear on the tendons.

The second option that was opted for in the end is to modify the servomotors themselves. The core of the problem is that the servomotors are only able to rotate 180°. The reason for this is that the angular sensor that is used to measure the angle of the output shaft of servomotor is not able to fully rotate. However if the angular sensor is removed, the servomotor is able to continuously rotate. As long as the angular sensor has not reached the desired angle, the motor will keep rotating in this case. An interesting option arises from these observations: the angular sensor can be removed from the servomotor and instead a similar sensor can be placed at the relevant joint instead. By doing so the servomotor now receives direct feedback from the joint angle instead of the servomotor output shaft. This means the motor will rotate as long as the desired angle is not reached at the particular joint. The 180° limit that was present earlier at the motor is now present at the joint. However the required range of motion of each joint is less than 180°, so this is no issue.

Having the angular sensors located at the joint locations directly has other significant benefits as well. By closing the feedback loop between the joint angle and the motor, the angular control will be more precise and faster. The drawbacks that this type of design has is that the servomotors controller may not be optimised for the additional delay the tendons may cause. This may result in the angle oscillating in some cases. This controller nor its parameters can not be easily changed, the only thing that can avoid this is by tightening the tendons more. Another minor disadvantage is that now electrical wiring is needed to the joints. Nevertheless the benefits this system has still overshadows its drawbacks.

3-6 The 2 DOF joint

The last joint of the finger is a 2 DOF joint allowing rotation along 2 axes. Because of this additional axis a single pulley used in the previously discussed 1 DOF joints will not work in this case. Getting this joint to work properly is not only hard because of this additional degree of freedom but also because it is the first joint of the finger. For each of the fingers degrees of freedom a tendon pair must be routed across this joint. This means a total of eight tendons must traverse this joint while also ensuring each pairs length is unaffected by rotation along either axis. There are basically two option to realize this joint. The first is the separate the single 2 DOF into two 1 DOF, the second is to find away to overcome its complexity and design the single 2 DOF joint.

3-6-1 Using two 1 DOF joints to emulate a 2 DOF joint

Some other robotic hands such as the ACT hand [9] opt to divide the single 2 DOF joint into two 1 DOF joints. This simplifies part of the design, and argued is that dividing this joint into two more accurately describes the anatomy of this joint. However these hand designs do not try to keep the total tendon length of a pair unaffected by angle and often use multiple actuators for each joint. In this specific case such a design may indeed by simpler. However consider what the joint must do if two 1 DOF are placed closely together. Figure 29 shows this. Eight tendons are seen to emerge from the right side (the hand palm). These eight tendons then need to traverse these two joints while being routed along two close pulleys that are rotated 90 degrees relative to each other. Between these two joints all eight tendons need to be routed very specifically. Not only does this require really precise and small parts, it also requires the tendons to be routed at very specific and abrupt angles. This is very likely to cause lot of additional friction. On top of this the eight tendons will likely rub against each other since four antagonistic pairs need to be rotated and routed to next pulley in a really right space. This causes significant friction and may even cause tendons getting tangles which would truly be disastrous.

3-6-2 A ball and socket joint as a 2 DOF joint

The second option is to design a single 2 DOF and route the tendons in a particular way to ensure a pairs tendons length is unaffected by joint angle. While discussing the previous two 1 DOF joints (Sections 3-4 and 3-5) concluded was that this could be ensured by routing along pulleys, there was in fact a second solution. The second solution was to route the tendons through the center of joints rotation. This idea can be extended to a 2 DOF joint. In this case all the tendons would have to routed through a common point. This idea has been executed before in other robotic hands such as the DART hand [23] (shown in Figure 30).

While this design works in theory, in practise there are some complications. The most obvious is that routing multiple tendons through a common point is impossible because tendons do have certain width. While the error this introduces is rather small, crossing tendons and having them touch and move along each other introduces more friction and wear on the tendons. Ideally this is avoided.

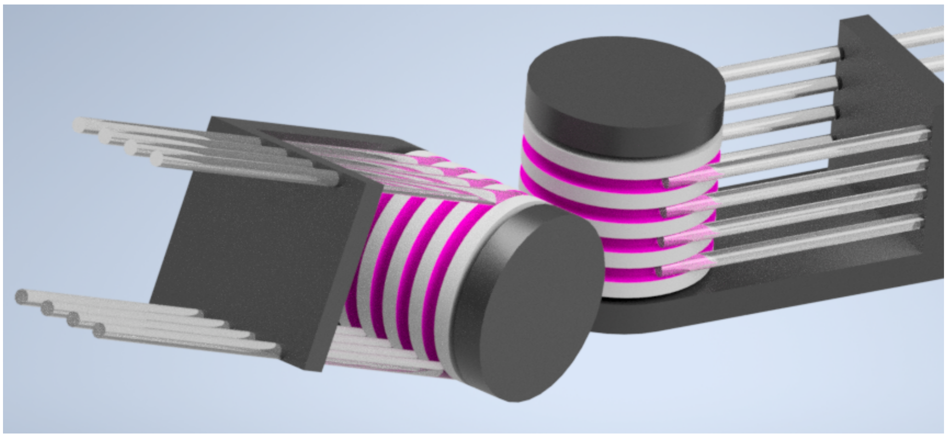


Figure 29: Two 1 DOF joints put closely together can emulate a 2 DOF joint. In magenta the grooves of pulleys are shown along which the transparent tendons. The routing between the two 1 DOF joints is not shown. The tendons from one side of the pulley would need to be rotated to the next pulley while also making sure the required ROM is guaranteed while the tendons remain in contact with the pulleys. Routing between two closely placed 1 DOF joints and guaranteeing length of a tendon pair is unaffected by either rotation is almost impossible.

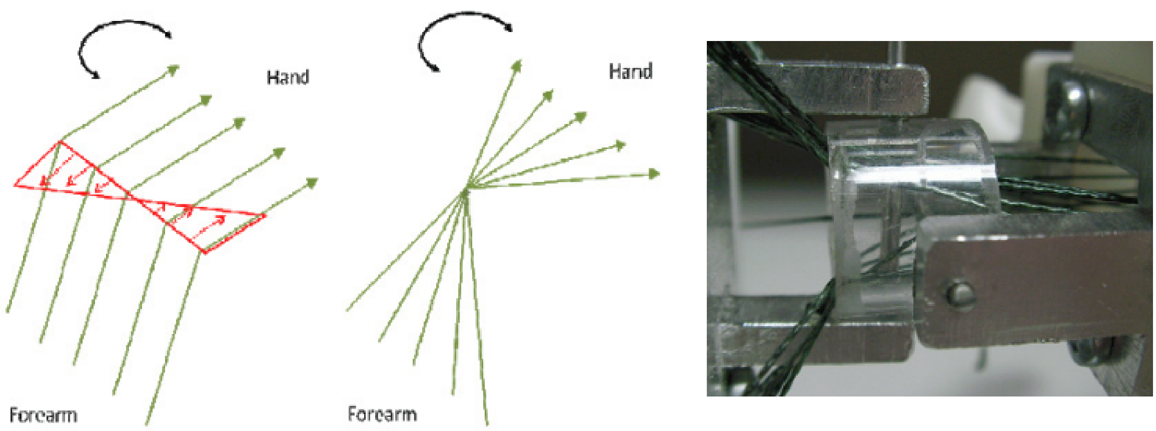


Figure 30: The left side illustrates how rotating through a common point ensures joint rotation does not affect the lengths of tendons. On the right side the implementation can be seen as done in the DART hand [23].

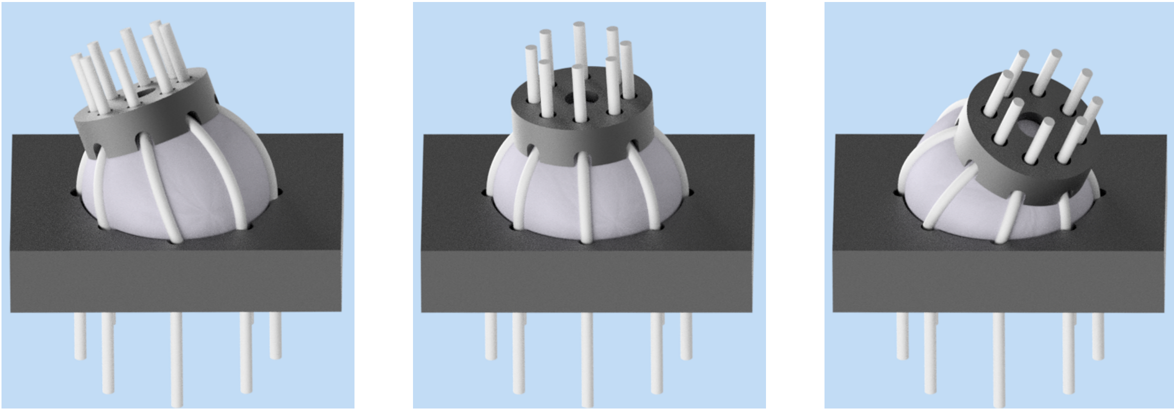


Figure 31: A 2-DoF joint with 8 tendons routed through it in three different angles. Tendons forming a tendon pair are routed on opposite side of the ball joint. The ball joint consists of a half sphere along which the tendons are routed. This ensures the total tendon length of a pair remains unaffected by the angle of the joint.

The final implementation that was opted for in the end was the results of expanding the 1 DOF pulley solution to three dimensions. Now instead of a circle a sphere is used to guide a set a tendons along. Figure 31 shows how this was achieved. The hand palm is located at the bottom and the finger at the top. A sphere is halfway put into another part creatively called the sphere holder. The sphere holder has a set of eight holes equally spaced along the edge of where the sphere resides. Eight tendons emerge from the bottom. Tendons that emerge on opposing sides of the sphere constitute a pair. The four opposing pairs are routed along the sphere to another part called the socket. The socket is part of the finger and contains eight holes for the tendons to routed through as well. The eight tendons can then routed to the other joints in the finger. Any rotation in either direction of this 2 DOF joint does not affect the total length of tendon pair at all.

To better illustrate why this design ensures that a pairs length is unaffted by angle consider Figure 39. This image shows a cross section of the ball and socket joint where a single tendon pair is shown. Notice how the sideways movement works similarly to the 1 DOF pulley joint. When rotation in the other direction happens (in the paper and away from the paper), the tendons will still be halfway routed along the sphere. This is the case because the point where the emerge from the sphere holder always remains the same.

This is the design that was used in the end. What is interesting from this design as well is that the finger and the base of the finger or not directly connected. The finger is free to slide along the ball joint. The only thing keeping it in place is the tension of the tendons. This ensures no ball bearings or other parts are needed to smoothen the movement.

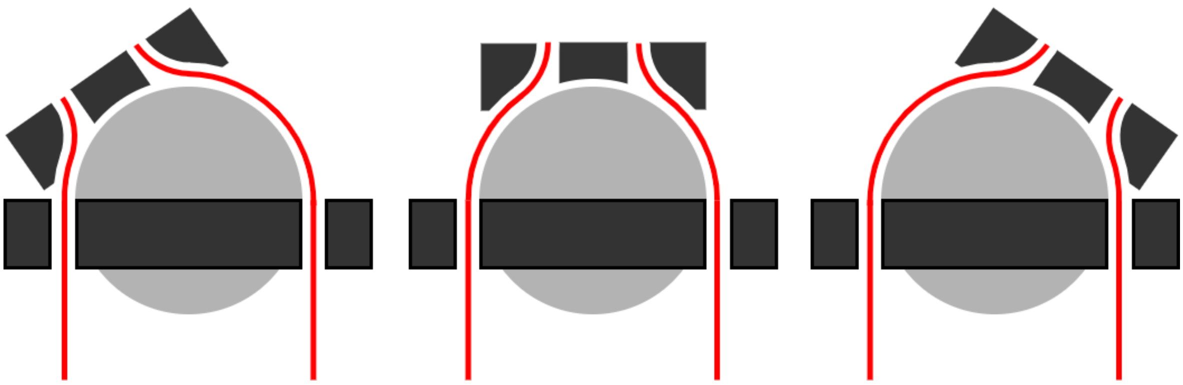


Figure 32: Cross section of the ball and socket joint and a single tendon pair being routed along the sphere. The bottom part in which the sphere is half way submerged is the sphere holder. The rest of the handpalm is connected to this. The top part forming the socket, is attached to the finger. The ball and socket joint is only held together by the 8 tendons, allowing free movement along the sphere.

Chapter 4

Creating the parts and assembling the robotic hand

To realize the finger using the design choices discussed in Chapter 3, parts where designed using 3D CAD software. Because of the small 3D parts required for the finger the design process and the printing of the parts using a 3D printer took a lot of trial and error. Because of the many revisions and version of the finger that were the designed not all version will be discussed in this chapter. Instead the focus will be on the final version that was used in the experiments, and why the design process lead to this final design with its choices of its parts.

This chapter will begin with the design of the finger and all the parts the comprise it, then this will be extended to the base of each fingers that holds the servomotors. Then two these fingers will combined to form a thumb and an index finger while also allowing space to potentially expand the hand to a five fingered hand in the future. And lastly the parts that enable it to be an actual tactile hand by introducing the sensors.

4-1 The finger and its first two joints

The design of the fingers first two joints is split into multiple parts according to the multiple layers of the finger design. This multi-layered design is used mostly because of 3D printer limitations that need to be considered. Because the designed parts are rather small relatively high accuracy and precision is needed for the printed parts. To achieve this 'bridging' in the 3D printing should be avoided. Bridging occurs a printed part consists of parts that are raised without underlying support. This is shown in Figure 33. When parts are needed that require bridging, the part is divided into multiple sub parts. These parts are later glued together. This process produces much more accurate parts than if the combined sub parts are printed as a single part.

The following sections describe the various layers that make up the finger. First the core layer that allows smooth movement of the two joints using bar bearings is discussed. Then

Figure 33: When parts are printed without underlying support, parts tend to be less accurate. To increase accuracy 'briding' is avoided as much as possible and often parts where bridging would be required are divided into multiple sub parts that are later glued together.

the guiding layers through which tendons are routed are discussed. Finally we conclude with how all these layers are put together.

4-1-1 Core of the finger

The first layer that is discussed makes up the core of the finger. It contains the parts in which ball bearings are placed to allow smooth movements. On top of these parts the other layers will be built. Figure 34 shows the parts that are part of this layer. For clarity each part that is part of another phalanx is presented in a different colour. The red middle part is the part that contains the ball bearings. The ball bearings are placed in the large holes present at the sides of this part. The green parts form the outer phalanx of the finger. It consists of a core and several parts that ensure this outer phalanx is well connected to the ball bearing present in the red middle phalanx. The blue parts make up the proximal phalanx which will eventually connect to the ball and socket joint. Apart from the length of its core it is very similar to the green outer phalanx. The loose parts needed to secure this phalanx to the ball bearing will be connected to the core in a different layer.

4-1-2 Guiding the tendons

The movement of the joints is actuated by the servomotors and their forces transmitted through tendons. This is extensively discussed in Chapter 3. The chapters conclusions show that routing the tendons in specific ways is required to increase the speed and precision of the movement. For the two 1 DOF joints guiding along pulleys ensures this.

To accomplish this in the actual 3D design of parts of finger various guiding pins were used to route the tendons through the finger. A cross section of one the fingers layers that contains a single extensor and a single flexor is shown in Figure 35. This particular layer is a guiding layer, whose function is guiding the two tendons. It consists of three 3D printed parts, one for each of the phalanges.

On the right side of the image, 2 tendons emerge from the 2 DOF joint. The first phalanx encountered contains a set of guiding pins on the right. These pins guide the tendons to right

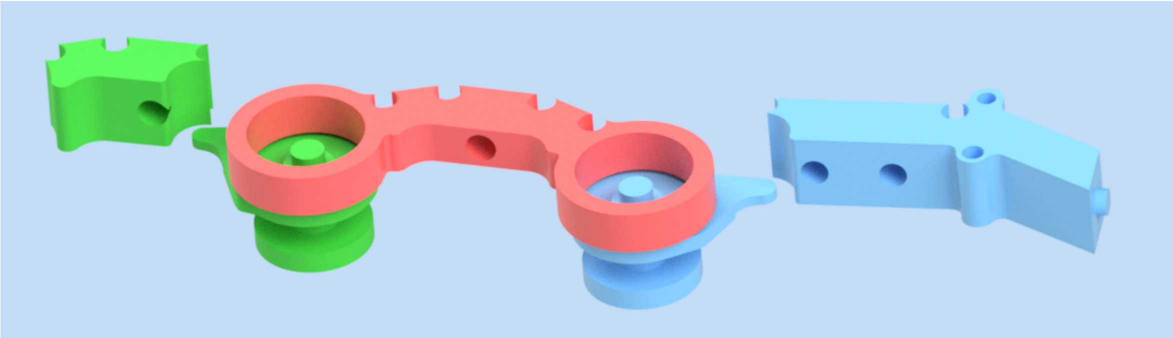


Figure 34: The 3D printed parts that form the solid base of the finger and contain the ball bearings to allow smooth rotation. The red center part that is part of the intermediate phalanx contains the two ball bearings. The ball bearings will be placed in the holes on the sides of the part. The green parts are part of the distal phalanx and is connected through the ball bearings centre to the intermediate phalanx. The blue parts are part of the proximal phalanx and are connected the red part in the same was as the distal phalanx.

height in the finger and make sure the tendons do not cross. The next set of pins is found right before the circular pulley. These pins guide the tendons in such a way that the tendons are partially wound around the pulley for all allowed angles, while minimizing sharp angles in the tendons to minimize friction. Lastly the tendons are guided along the pulley.

The second phalanx contains no pulleys, only guiding pins. These pins ensure the tendons are always partially wound around the pulleys that are present in the other two phalanges for the allowed joint angles.

The third and final phalanx of the finger contains the final pulley and the guiding pins which serve similar function as the previous ones. The tendons depicted in the Figure 35 end abruptly in this last phalanx. This however is not the case in the actual design. One of the tendons that makes up tendon pair exits the last phalanx from below where it is fastened using the small hole found below one of the guiding pins. The upper tendon of the tendon pair exits the phalanx at the top where it is secured to a rotatable piece that allows the tension on the tendon to be changed. This part will be discussed in Section 4-1-3.

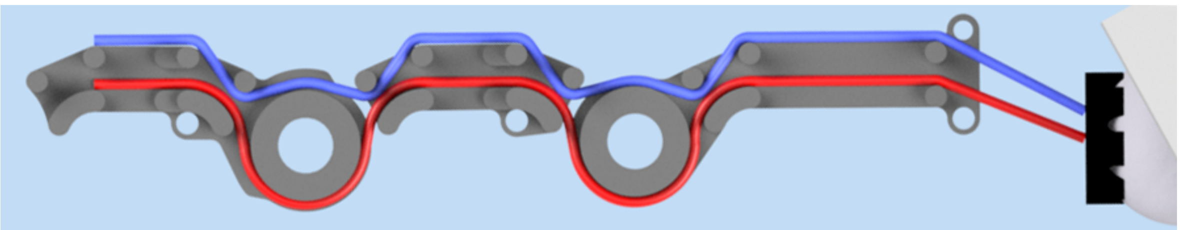


Figure 35: Cross section of the robotic finger design showing the routing of the extensor (blue) and flexor (red) tendon. The tendons exit from the ball and socket joint on the right and are fastened at the left side in finger. As long as the tendon remains tangential to the joint-pulleys, the tension of the tendons is unaffected by the angles.

While Figure 35 gives a good explanation of how the tendons are routed in the finger, it shows only a single layer. It also gives the illusion that each tendon pair is routed through the same layer. This however is not the case. For both 1 DOF joints of the finger each of the antagonistic tendons is routed in a different layer. This is depicted in Figure 36. The figure shows the two guiding layers that are present in the finger. The layers are mirror images of each other and similar to Figure 35. This view however shows 4 tendons, forming two tendon pairs, emerging from the 2 DOF joint: the green tendon pair controls the outer joint, the magenta tendon pair controls the inner joint. As can be seen the extensor and flexor of each tendon pair is routed in different layers on opposing sides. Routing tendons of a tendon pair on opposing sides of the finger allows for more stability in the finger. To illustrate this, imagine what would happen if the green pair, responsible for the outer joint, was routed in a single layer. The tension of the tendon pair would cause the joint to bend to the left or the right when viewed from the top. Another reason that necessitates routing the tendons on opposing sides is due to the design of the ball and socket joint. This will be discussed in Section 4-2.

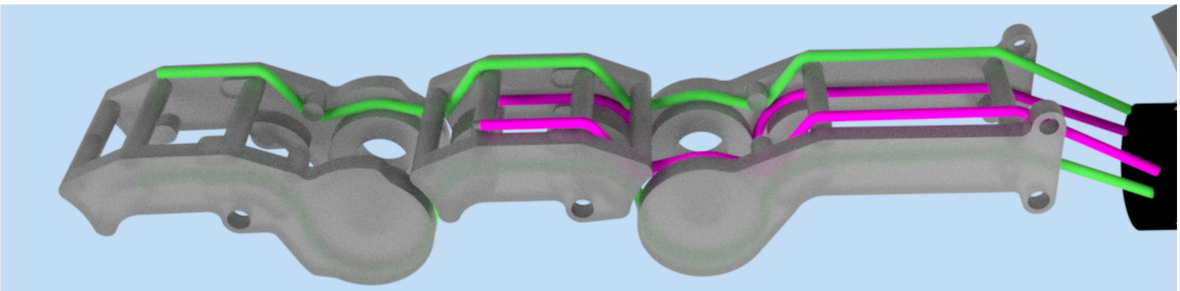


Figure 36: Two pairs of tendons that provide the rotation of the two 1-DoF joints. The green pair rotates the outer 1-DoF joint. The pink pair the inner 1-DoF joint. As the two tendons forming a pair exit the ball and socket joint on opposing sides, the pair is split between two layers.

The discussed guiding layer design shown in Figures 35 and 36, allows the two IP joints about 20 degrees of hyperextension from the fully extended position shown in both Figures, and about 160 degrees of flexion. These actual angles for which the tendons remain tangential to the pulleys is affected by the thickness of the chosen tendon. In this case the used tendons are made of fishing line with a thickness of 0.25 mm.

4-1-3 Combining the fingers layers

The final part of the finger designs is the put the layers together. The solid layer has holes that the pins of the guiding layers fit neatly, while allowing space for the tendons to move. A single routing layer (blue) and part of the solid base layer (green) is shown in Figure 37. As can be seen the parts that where first not joined together in Figure 34 are joined when putting them together with the other layers.

Other than the layers discussed, the finger contains two more parts: The angular sensors that are used for the two outer joints, and a part called the 'tightener' that is used to change the tension on the tendons. The rotatable part of the angular sensors are easily glued to the guiding layer of the proximal and distal phalanges in Figure 37. More specifically to the green part that is partially shown at the bottom below the guiding layer. The other part of the angular sensor is secured to the guiding layer of the intermediate phalanx. This ensures the angular sensors measures the angle between distal phalanx and the intermediate phalanx, and the proximal phalanx and the intermediate phalanx.

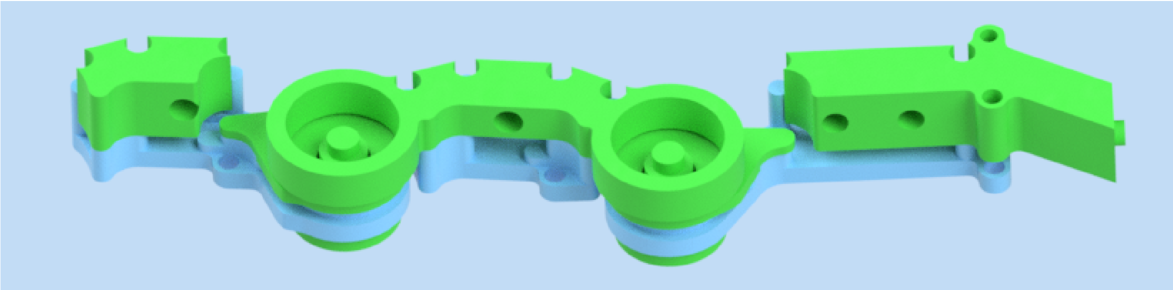


Figure 37: A single routing layer is shown in blue below. This layer contains the tendons that provide the forces that move the finger. The solid base layer shown in green contains the ball bearing that reduces friction. The second routing layer that would be placed on top is not shown here.

The final parts that are placed at each finger are four tighteners. These are rotatable printed parts that are secured to the finger by using bolts. Figure 38 shows these four tighteners on the finger. A single tendon of each tendon pair is secured to each tightener. By securing a single tendon of each pair, the entire tendons pairs tension can be controlled because the their tension depends on each other. By rotating the tightener the connected tendon is either tightened or released. This allows precise control of tension on each of tendons.

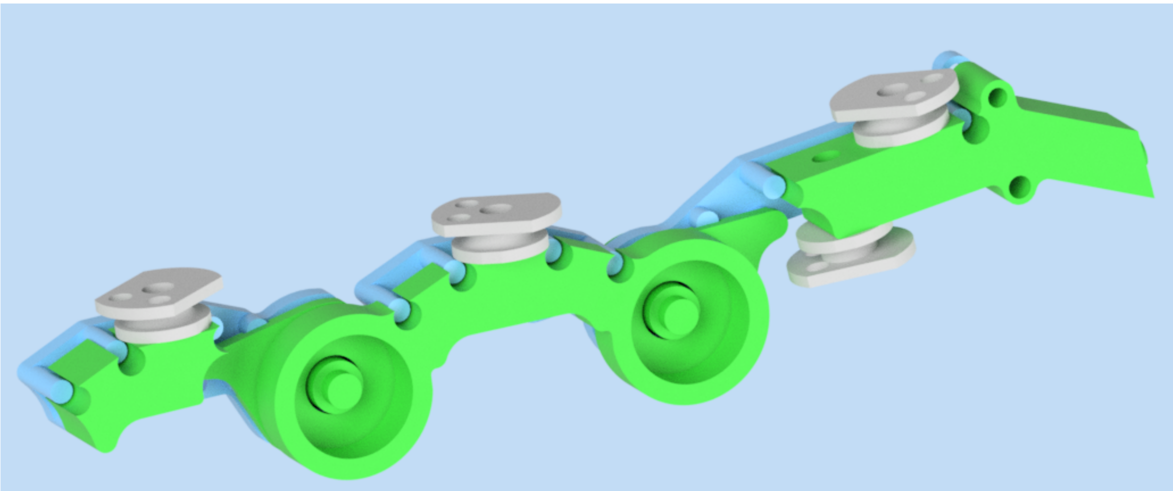


Figure 38: Four tighteners are shown in grey on top of a single guiding layer and the solid base layer. The tendons that would be connected to the tighteners are not shown. When a tightener is rotated one way the tension on the secured tendon is increased because it is wound more around the tightener. If it is rotated the other the tension decreases. This allows the tension of each tendon to be controlled precisely. The tension of each tendon has effect on the speed and accuracy of the movement.

4-2 Creating the ball and socket joint

The 2-DOF joint as discussed in Chapter 3 was implemented using a ball and socket joint. A smooth sphere was placed halfway inside a 3D printed 'sphere holder' and secured to it.

The combined part of the sphere holder and the smooth sphere is part of the rigid hand palm which contains the servomotors. The sphere holder has eight holes equally spaced around the edge of where the sphere resides to provide the guides for the eight tendons. The tendons that emerge from the sphere holder curve along the sphere until they reach the socket of the finger. This 3D printed part is directly connected to the finger shown in Figure 36 and is the black part the tendons emerge from in this image. This socket similarly to the sphere holder has eight holes placed close to the edge of the socket through which the tendons are routed to the rest of the finger. By keeping the holes of the sphere holder and the socket relatively small, the tendons are constrained and it can be guaranteed that the tendons are routed closely along the curvature of the sphere. The eight holes are needed for the eight tendons that form four pairs. Each of these pairs consists of two opposing tendons that need to have a combined length that is irrespective of the joints angle. This can only be achieved by routing the two tendons of each pair on opposing sides of the sphere, forming a construction that is very similar the pulley design but also works in multiple dimensions. A schematic cross section of this ball and socket joint showing a single tendons pair is shown in Figure 39.

The range of motion the ball and socket has in its two directions is influenced by the dimensions of its part. In Figure 39 the ROM is limited by the socket colliding with sphere holder. It is clear that a smaller socket and larger sphere enable a larger ROM. In this case a sphere with a diameter of 16 mm was used, and a socket of 12 mm width. This made sure a ROM of 90° could be achieved in both directions.

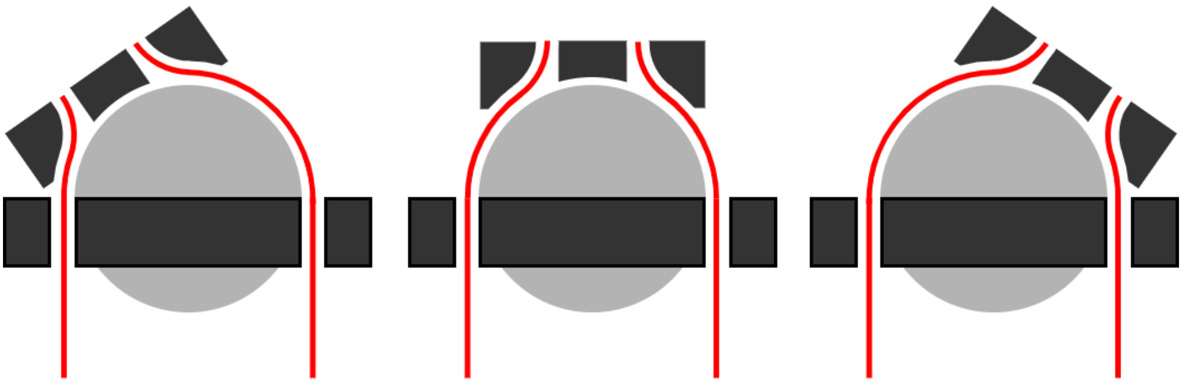


Figure 39: Cross section of the ball and socket joint and a single tendon pair being routed along the sphere. The bottom part in which the sphere is half way submerged is the sphere holder. The rest of the hand palm is connected to this. The top part forming the socket, is attached to the finger. The ball and socket joint is only held together by the eight tendons, allowing free movement along the sphere.

4-3 The base of the finger

The base of each finger contains the servomotors. To control each of the four DOF of the three joints of each finger, four servomotors are used per finger. Each of the servomotors is connected to a pair of tendons. These tendons as previously discussed end in the finger. One of each pair is tied to a tightener the other is secured to hole of the appropriate phalanx. In the base of the finger the tendons are wound around a so called bobbin. The bobbins are connected the servomotors. Holding these bobbins and the servomotors in place is the task of the finger base.

Figure 40 shows the base of single finger and two bobbins (yellow). The blue top part holds the servomotors. The spacing between the holes that hold the servomotors is dependent on the size of the bobbins. It is important the bobbins are not placed too closely as them touching would increase friction in the finger. The grey bottom part holds 4 ball bearings. These are used to secure the bobbins at the bottom while reducing friction. In earlier designs of the base where bobbins were only secured at one side to the servomotors. The tension of the tendons on the the bobbins caused significant tilting in the bobbins. Securing the bobbins at the top and the bottom makes sure they remain upright. The red part is the sphere holder that was discussed earlier in Section 4-2. Other than holding the sphere used for the ball and socket joint, it adds space and support between the servo holder and the ball bearing holder. The last green part is used to add additional support to the base construction.

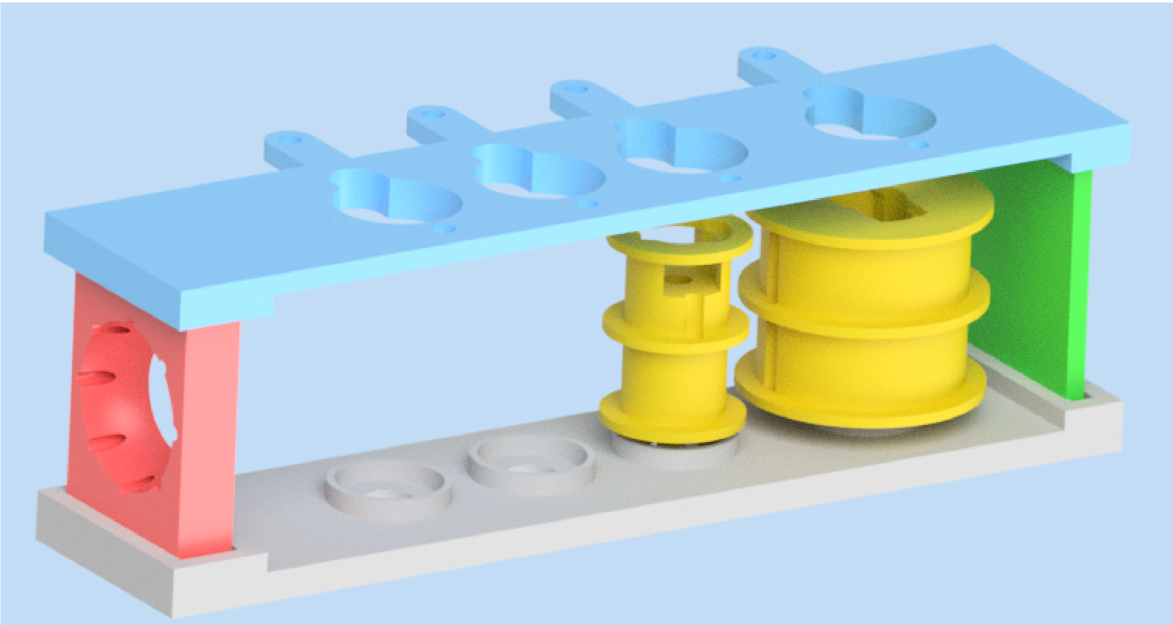


Figure 40: The base of a single finger consists of 4 parts: The socket in which the 16 mm sphere is placed that forms the ball and socket joint (red), The blue part that holds the servomotors. A green part that add stability to the construction, and the grey parts that holds 4 ball bearings used to secure the bobbins. The bobbins (yellow) are connected to the servomotors at the top and secured to the ball bearings at the bottom.

4-3-1 Designing bobbins

The bobbins are the parts that are directly connected to the servos. Around these the tendons are wound. Two bobbins of two widths are shown in Figure 41. To increase the accuracy of the 3D prints, they are printed in four parts, as indicated by the four colours. The bobbin is placed directly inside the base. The top part (red) contains an extrusion which is placed inside a ball bearing. If this top part is not secured in some way the tension in the connected tendons cause the bobbins to tilt in the direction of the tendons. Around the yellow part one of the tendons of a tendon pair is wound. Around the green part another tendon of the pair is wound. This time in the other direction. This makes sure rotating the bobbin one way reels in one tendon and releases the other. The wider part of the green part also contains a hole. Through this hole the two tendons that make up a pair are connected. So basically each pair is actually a single piece of tendon. This is done so that no tendons have to be secured to any part here. By winding the tendons around the bobbins multiple times the tendon will be secure and will not slip from the bobbin.

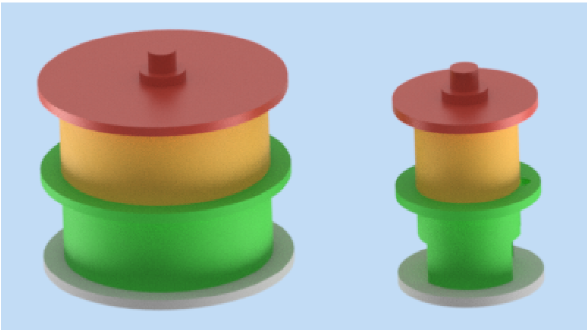


Figure 41: Two sizes of bobbins are printed in four parts to avoid bridging inaccuracies during 3D printing. The red top part is secured to a ball bearing present in the base. Around the yellow and green parts tendons are wound. The wider green part contains a hole through which the tendons are connected (forming a single tendon). The bottom grey part is secured to a servomotor.

As Figures 40 and 41 suggest, two size of bobbins are used. For the bobbins connected to tendons that control the 2 DOF ball and socket joint and the inner 1 DOF joint the thinner variation of bobbin is used. Ideally this size bobbin would be used for the final outer 1 DOF joint aswell, since this would reduce the space requirements for the base. In earlier designs having all bobbins be the same size was tested. However when the speed of the joints was tested the outer joint was significantly slower than the others. Due to how the tendons are routed, the servomotor that controls the outer 1 DOF joint has to rotate as well when the other joints rotate. When closing the finger from an open position, each joint moves approximately 90°. This means the servomotor of the outer joint has to rotate a total of $3 \times 90° = 270°$. While the rotation limit was solved by removing the angular sensor from the servomotor output shaft, the rotation speed was not increased. To compensate for this the size of the bobbin was increased, effectively changing the gear ratio between the bobbin and the joint: a small rotation of the servomotor would cause a larger rotation at the outer 1 DOF joint. While this decreases the resolution somewhat, speed increase was deemed more important.

4-3-2 Combining two fingers

As discussed previously, the design of the thumb is similar to the index finger. Using the previously discussed design for the creation of a finger, two fingers are printed. The last step is to combine these two fingers. The most important thing of how to combine these two fingers representing a thumb and an index finger, is the angle and distance between them. In particular between the 2 DOF ball and socket joint of the two fingers. Using two 3D printed parts, together called the combiner, the accurate angle and distance at which the two fingers should be separated was realized. This is illustrated in Figure 42. This is done in such a way that additional fingers can be added with only minor alterations. An example of what this could look like is shown in Figure 43.

Figure 42: The combiner made of 2 green parts connects the index finger (yellow) to the thumb (red). Olive green part is secured to the thumb. The Bright green part is fixed to the index finger. Holes present allow screws to be used to secure the two finger together. The multiple rows of holes present in both green parts allow the distance from the fingers to be adjusted.

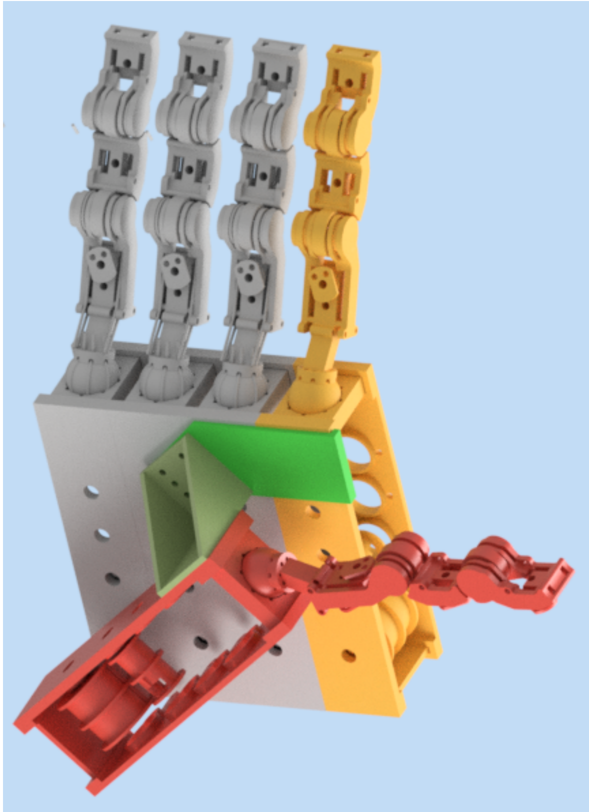


Figure 43: The index finger (yellow) and thumb (red) are shown on the right. The other three fingers (grey) can later be added.

4-4 Making the hand tactile

The last parts to be designed are parts that are required to make the hand actually tactile. This is done by using strain gauges to detect pressure at the tip of the fingers. Initially the strain gauges were just attached to last phalanx of both fingers using tape. In this version no additional parts were 3D printed. While this did work to detect pressure at the tip of the fingers, it was not very sensitive and required a very specific angle at which would detected anything. Because of this additional parts were created to house the strain gauges and direct forces that are applied to the tip of the finger towards the strain gauge. Figure 44 shows the three parts that make up the finger tip that hold the strain gauge. The blue and red parts are together glued to the green core of the outer phalanx. These two parts contain the strain gauge. The yellow part is only loosely connected to the red part using bolts. This allows the yellow part to be pressed which directs the forces onto the strain gauge. Figure 45 show the yellow part in more detail. The circular extrusion present in this part presses against the strain gauge when the yellow part is pressed. These parts provide much greater accuracy and sensitivity to the force detection compared to simply attaching the strain gauges to the end of the finger.

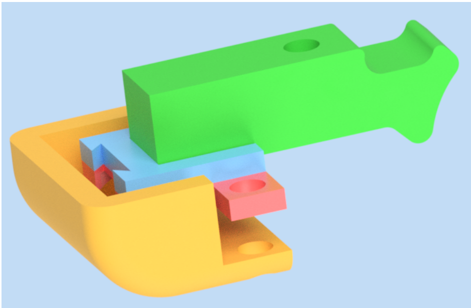


Figure 44: The finger tip in which the strain gauge is placed. The core of the outer phalanx is shown in green. The blue and red parts are glued to the green core and contain the strain gauge. The yellow part is loosely connected to the red part using bolts. By pressing the yellow part the forces are directed towards the strain gauge

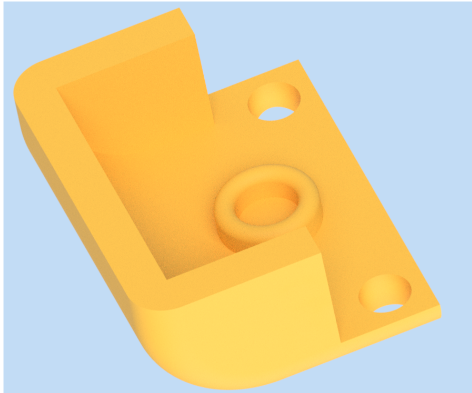


Figure 45: The yellow part directs the pressing force towards the strain gauge using the circular extrusion. It is loosely connected to the other part of the finger tip using bolts.

4-5 Electrical wiring of the strain gauges and the servomotors

This last section of the hardware design of the hand deals with the electrical wiring. The two fingers are each controlled by four MG90S servomotors. Each servomotor has three electrical inputs as shown in Figure 46. These are V_{cc} signal, the ground signal, and the PWM signal. The first two provide the power to the servomotor. Their voltage should be in the range of 4.8 to 6 Volts. The last input requires a Pulse Width Modulated (PWM) signal whose duty cycle dictates the servomotors desired angle. To control the servomotors an Arduino Uno is used. Initially the Arduino was used to control as well as provide power to the servomotors. The power Arduino could provide was not sufficient when using eight servomotors so an external power supply was instead used. Figure 47 shows the electrical wiring of the subsystems. On the left the the fingers with their four servomotors are shown. The red V_{cc} signals and the black ground signal all go to the external power supply. The orange PWM signals of each servomotor go to separate ports that each generate their own PWM signal. It should be noted that the ground of the external power supply is connected the ground of the Arduino.

To detect pressure in the strain gauges a simple resistor divider circuit is used. This is also shown in Figure 47. Each of the strain gauges is directly connected to a 5V output of the Arduino. The other output of the strain gauge is each connected to its own resistor which is in turn connected to the ground of Arduino. Pressing the strain gauge lowers its resistance value and the IO port that one of the strain gauges inputs is connected to reads the voltage at this terminal, which enables us to detect pressure.

The Arduino is through a serial communication connected to a PC which is in turn connected to an external glove, the SenseGlove. This will be discussed more in depth in Chapter 5.

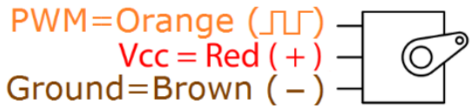


Figure 46: The MG90S has three inputs: The red and brown signals are the power input. Their voltage should be in the range of 4.8 to 6 V. The orange signal should contain a PWM signal. The duty cycle of this PWM signal dictates the angle the servomotors strives for.

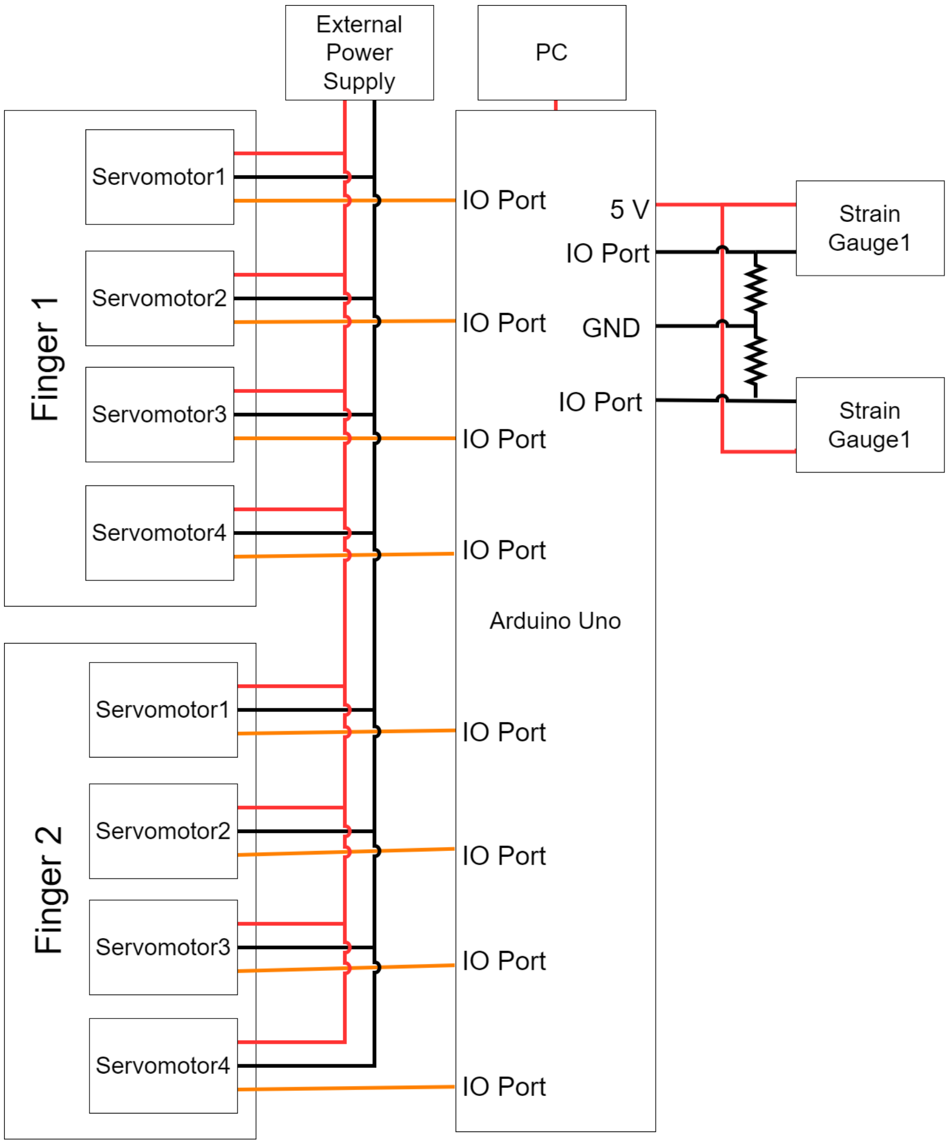


Figure 47: The electrical diagrams of the systems. On the left side, a total of eight servomotors which are divided between two fingers are powered by an external power supply. The ground of this power supply is also connected to that of Arduino. Each servos PWM signal goes to a separate PWM generating port at the Arduino. On the right, a resistor divider circuit is used the detect pressure at the strain gauges.

4-6 Costs of the robotic hand

The created robotic hands consists of eight MG90S servomotors, twelve ball bearings, two metal sphere, an Arduino Uno, two strain gauges, an external power supply, and some fishing line, bolts and nuts, and 3D printed parts. The servomotors cost around €20, the ball bearings €15, the power adapter €15, the two metal spheres €2, the Arduino €30, the strain gauges €5, and the other parts less than €20. This adds up to less then €110, which is extremely cheap for a robotic hand design that has eight fully actuated degrees of freedom.

Chapter 5

Implementing the robotic hands movement

To actually move the hand an Arduino was used to control the servomotors that control the fingers. The Arduino is connected through a serial communications to a computer. The computer runs the game engine Unity [7] on which code from SenseGlove-Unity [5], which is expanded upon, is used to communicate with the Arduino and a SenseGlove [4]. The SenseGlove is user wearable tactile glove that records the user hands position and can provide force feedback by preventing the users fingers from closing. Figure 48 shows a user wearing the SenseGlove. All the code used in the Thesis can be found at github [3].

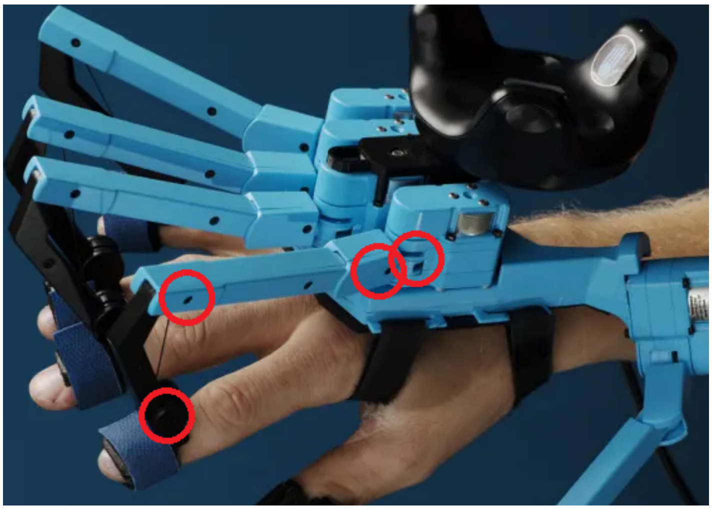


Figure 48: A human user wearing the SenseGlove. The 4 marked points contain the locations of the angular sensors that are used to determine the fingers position. The exoskeletons kinematics is the same for each finger including the thumb. Between the two left red circles a black string can be seen. This string provides the force feedback by halting the user from closing their finger.

This chapter will focus on the code that is used to let a user wearing a SenseGlove control the robotic hand whilst also being able to receive force feedback. Starting at the robotic hand side, first the Arduino code that is used to control the servomotors is discussed. Then, the communication between Arduino and Unity is discussed. After which the Unity side will be discussed. This consists of how Unity controls the robotic hand, retrieves the correct joint angles, and generates the force feedback for the SenseGlove. Figure 49 shows the system end to end.

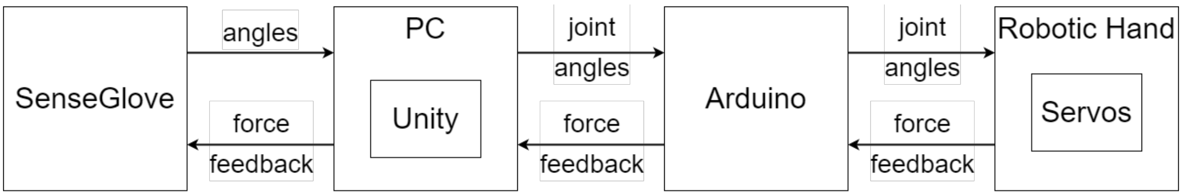


Figure 49: The end the end system. On the left SenseGlove receives force feedback from Unity and provides Unity with its angular sensor readings. The PC calculates the correct joint angles for the robotic hand and sends these to Arduino. Arduino reads the robotic hands force feedback data from the strain gauges and delivers these to Unity. It also applies an offset to the received joint angles before it sends these to the robotic hand.

5-1 Moving the servomotors using Arduino

To move the robotic hand eight servomotors are used. These servomotors are the cheap MG90S. As discussed in Section 4-5 these servomotors have three inputs, two are from power and one is used to control the servomotors position. This is achieved by generating a Pulse Width Modulated (PWM) signal to the servomotor. The PWM signals duty cycle dictates the servomotors angular position. Figure 50 illustrates this.

Using the Servo-library [2], these signals are easily be generated using an Arduino. Simply assigning one of Arduino's the dedicated pins as a servo-pin allows one to set the servo angles in degrees and the library will generate the correct PWM signal.

The angles that the servomotors should be set to can be controlled in two ways. The first is by manually turning potmeters on the Arduino, the second is by receiving angular commands from Unity. The first of three potmeters controls whether the angular data from Unity (if it is present) should be used to control the robotic hands servomotors or the whether manual control is used. The last two potmeters are used to control flexion/extension, and abduction/adduction of the robotic hand by moving the servomotors. Manual control was mostly implemented for debugging purposes and to be used for calibration purposes.

Applying an offset to every servomotor is required. This is the case for the ones controlling the outer joints because an angular sensor is manually placed at this joint, so it can easily be off by a few degrees. For the servos that control the 2 DOF joint the original angular sensors are still present in the servomotors, however depending on the initial joint angle and the bobbin angle when the robotic hand was assembled the angle may be off as well for these two servomotors. For each of the joints recorded is what sent angle value results in actual joint angles of 90° and 180°. From this a linear relationship is created to correct for the aforementioned errors.

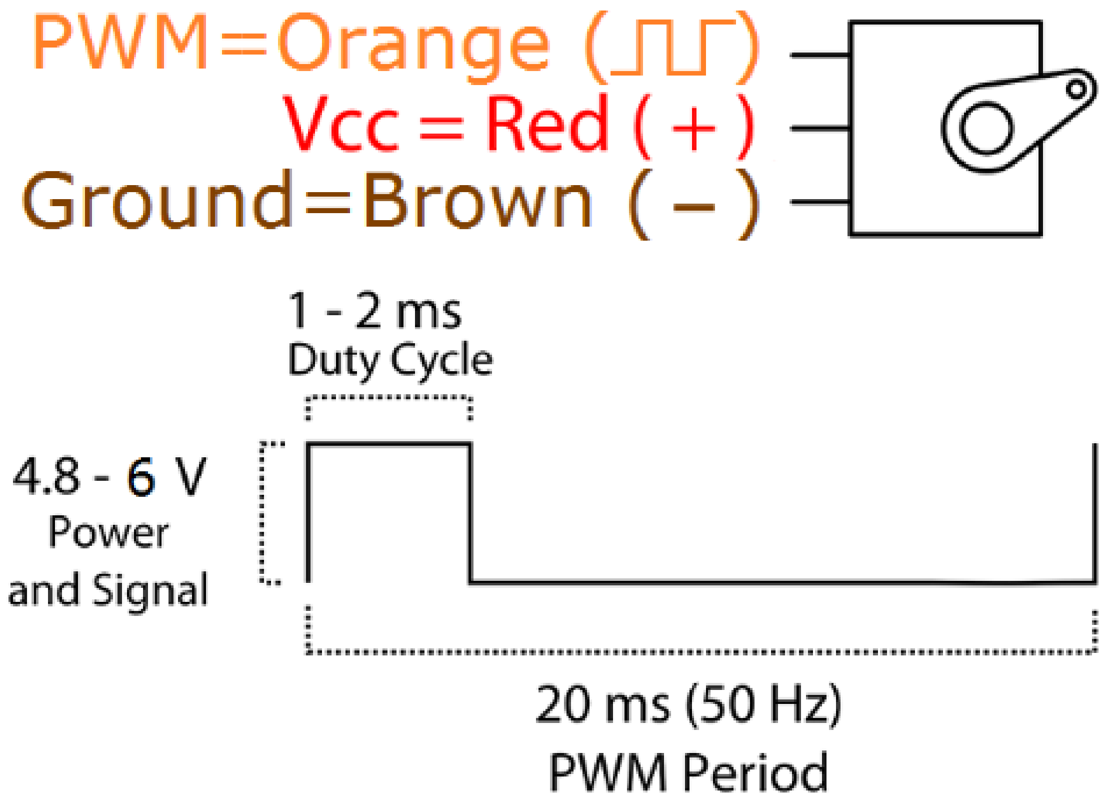


Figure 50: PWM signal of MG90s. A 1 ms pulse rotates the servmotor to position -90 degrees. A 2 ms pulse rotates the servo to a position of +90 degrees.

5-2 Communications between Arduino and Unity

The robotic hand consists of two fingers both which are controlled by four servomotors. Thus Unity needs to send a set of eight angles. The servomotors have a range of 0° to 180° To be able to fully utilize this a singly Byte consisting of 8 bits is transmitted. Because only 180 of the 256 possible combinations of 8 bits are used to transmit angles, other combinations can be used signify the start of sequence or other control signals. In this case a byte value of 255 is used to mark the start of a sequence of 8 angles. Figure 51 shows the sequence of nine bytes, starting with the Start byte, followed by 4 bytes representing angles of the index finger, and ending with the 4 bytes representing the joint angles of the thumb. If no starting byte is utilized, there would be no way for the Arduino to know which received angle corresponds to which joint. The sequence of these nine bytes is sent at 60 Hz from Unity to Arduino. As will be discussed in Section 5-3, the Unity code will also allow to sent a value of 250 as an angle. When this value is received the angle of the servomotor will not be changed. This is useful for debugging purposes when the SenseGlove is used.

The other way Arduino sends force feedback to Arduino, the interval at which this happens is initially set at 10 Hz. If the force feedback data would be sent every Arduino loop Data would be sent too fast for Unity to handle. In this case only two values which range from 0 to 255 are sent, one for each finger. The voltage values that are read from the strain gauges

Start (255)	Index 1	Index 2	Index 3	Index 4	Thumb 1	Thumb 2	Thumb 3	Thumb 4

Figure 51: serial sequency

by the Arduino originally range from 0 to 1023, however because this does not fit inside a single byte these values are scaled down. The resolution lost in this way is not of importance because the SenseGlove is not able the handle a force feedback range larger than 0 to 100. Similarly to sending the angle data, a starting byte is used to differentiate between the two fingers.

5-3 Using Unity to control the Robotic hand

The main tasks of Unity is to send joint angle data to the Arduino. Arduino then adjusts these joint angles to compensate for offset due to inaccuracies of the angular sensor placement and then sends these angles using PWM signals to the servmotors. Two options are implemented to control the joint angles. They can be generated in Unity through user input or angles be derived from sensor data from the SenseGlove.

5-3-1 Changing joint angles manually in Unity

For the first case there are multiple ways implemented for the Unity user to change the transmitted joint angles. The transmitted joint angles are displayed in 2 ways. Firstly a slider is used that ranges from 0° to 180°. This slider displays the currently transmitted joint angle for each of the 8 joints. It can be be moved directly by the user to change the transmitted angle. Secondly a text field next to the slider displays the exact value of the transmitted joint value for each joint. Here the user can also directly input a value, that is then transmitted to the Arduino. A third way of changing the transmitted joint angles in Unity is implemented by sinusoidally varying the transmitted joint angles. This is implemented using two input text field to control the frequency and its offset. By enabling this mode for each joint the angle is automatically changed according to the frequency and offset given by the user. This mode is especially useful to test the limitations the robotic hands in terms of speed and movement delay. The entire interface is shown in Figure 52.

5-3-2 Changing joint angles using SenseGlove inputs

The raw joint angles that come from the SenseGlove contain the angular position of four joints for each finger. Figure 53 shows the location of these sensors. These sensors do not directly translate to a hand position as only the tip of the finger and the base of the hand are directly attached to the SenseGlove. Thus an algorithm is needed to derive the human fingers positions from this data. SenseGlove-Unity [5] provides a demo code for the SenseGlove for Unity. This provides an algorithm to directly calculate the users hand posture. This way was initially used, however some imperfections with this algorithm were found. This is shown in Figures 54 and **??**. In Figure 54 the SenseGlove user bends their finger at approximately 90° at their middle joint and in Figure 55 90° at their 2 DOF joint. However both cases lead

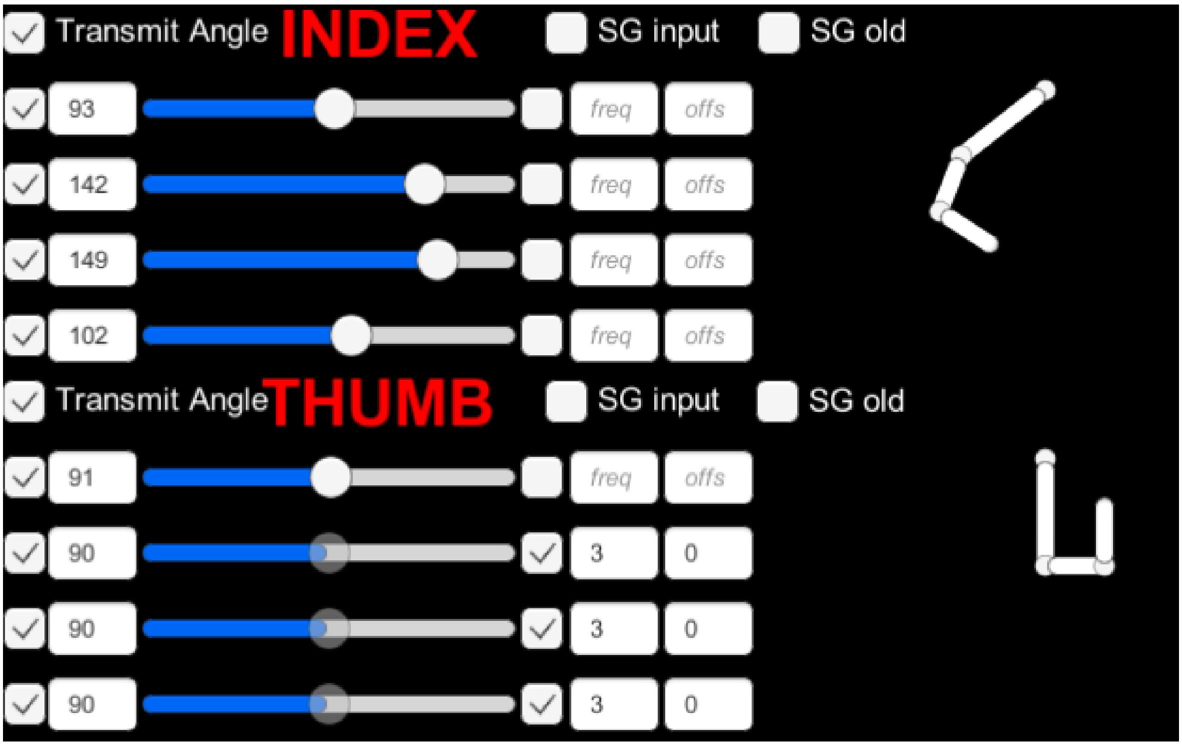


Figure 52: The Interface through which the user can change the transmitted joint angles. The top half shows the interface for the Index Finger, The bottom half for the Thumb. The sliders represent the transmitted joints angles ranging from $0°$ to $180°$. To the left of each slider the exact transmitted joint angle is shown. To the right of the slider the frequency control is shown. By inputting a frequency and an offset and toggling the its box, frequency control is enabled. On the right a graphical side view of the joint angles is shown. Each of the two fingers tip slider is for the sideways movement at the 2 DOF joint, the other three sliders are the inner joints, middle joints, and the outer joints flexion angle. The two togglable boxes in the top right of each fingers control interface override the manual input and uses the SenseGloves angles instead. In this case the index finger has been manually set to a stationary position and the thumbs three joints are currently flexing and extending periodically at a frequency of 0.3 Hz.

to almost the same incorrect posture. This is especially problematic if grasping objects is desired.

Algorithm the derive the hands joint angles from SenseGlove

Because of the aforementioned issues with the standard code as presented in SenseGlove Unity, an algorithm is written which derives the users joint angles from the incoming SenseGlove data. For the algorithm it is assumed that the user always wears the SenseGlove in the same way. This means that the location of the users 2 DOF joint relative to the SenseGlove is always the same. From measuring the part lengths of the SenseGlove and the phalanx length of the users fingers the human users outer joint position and finger tip can be derived. From the outer joint and the 2 DOF joint locations the intersection point of two circles can found to retrieve the location of the middle joint, using a circle intersection algorithm [1]. From the joint positions and the finger tip position their joint angles are calculated. Figure 56 illustrates

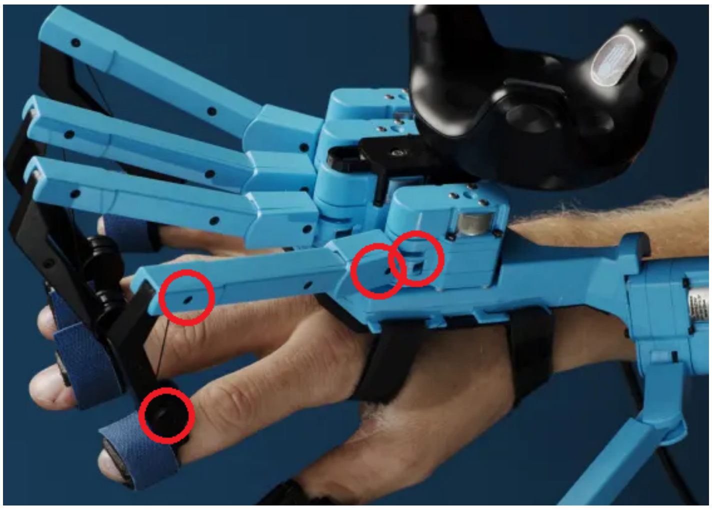


Figure 53: A human user wearing the SenseGlove. The 4 marked points contain the locations of the angular sensors that are used to determine the fingers position. The exoskeletons is the same for each finger including the thumb.

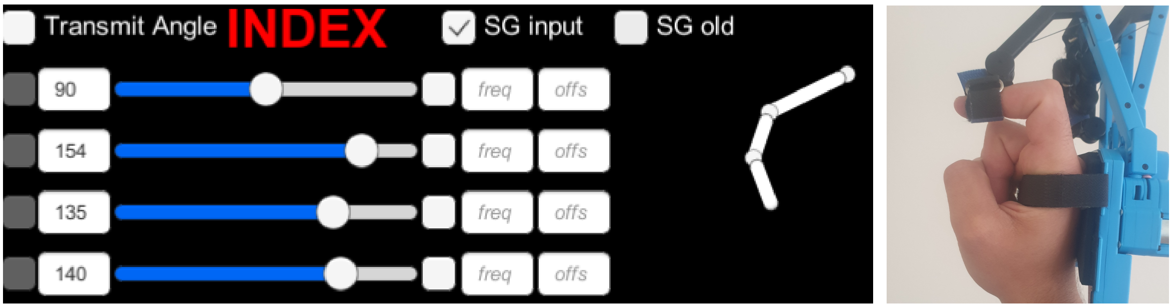


Figure 54: The users index finger is bent $90°$ at the second PIP joint, this does not translate well using the original algorithm.

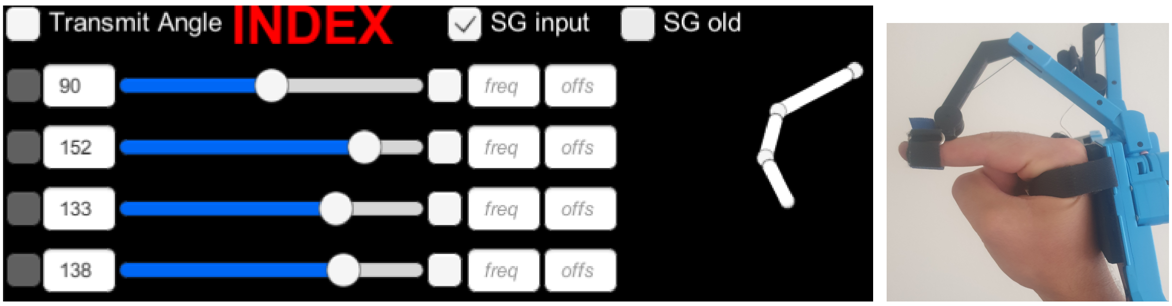


Figure 55: The users index finger is bent $90°$ at the first MCP joint, this does not translate well using the original algorithm.

the algorithm. The sideways movement of the users inner 2 DOF joint is not derived using this algorithm, instead the direct angular data from the SenseGlove can be used because this is the only joint that allows for sideways movement.

For most hand postures this algorithm performs better than the original. This is shown in

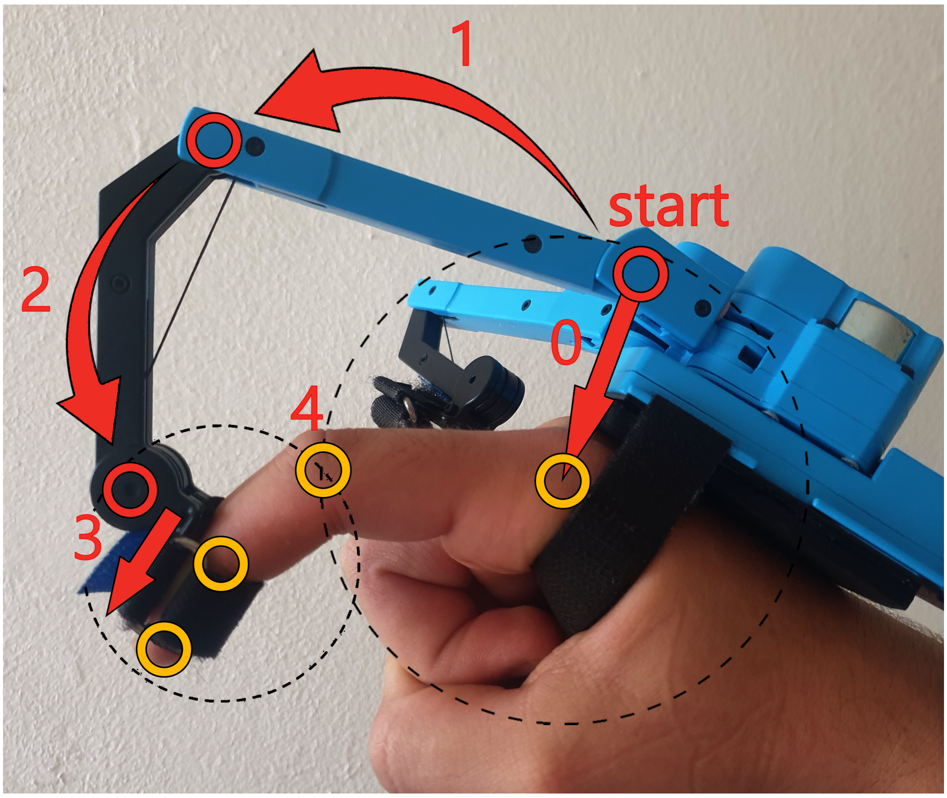


Figure 56: Algorithm to derive the users joint angles:

0. The first joint of the SenseGlove marked with start is the initial position. From here the users 2 DOF joint is always assumed to have the same position relative to start.
1. The known lengths of the SenseGloves parts allow the position of the next joint of the SenseGlove to be calculated.
2. The location of the SenseGloves last joint relative to start is calculated.
3. From this SenseGloves last joint angle and location, the position and angle of the users outer phalanx can be derived.
4. From the users known outer joint and inner joint positions and the users phalanx length (which were measured), two circles can be drawn. The intersection location gives the middle joints position. Now that all of the users joint and finger tip locations relative to start are known, each joints angles can be calculated.

Figure 57: The users index finger is bent $90°$ at the second PIP joint, using the newly written algorithm the derived posture has increased significantly.

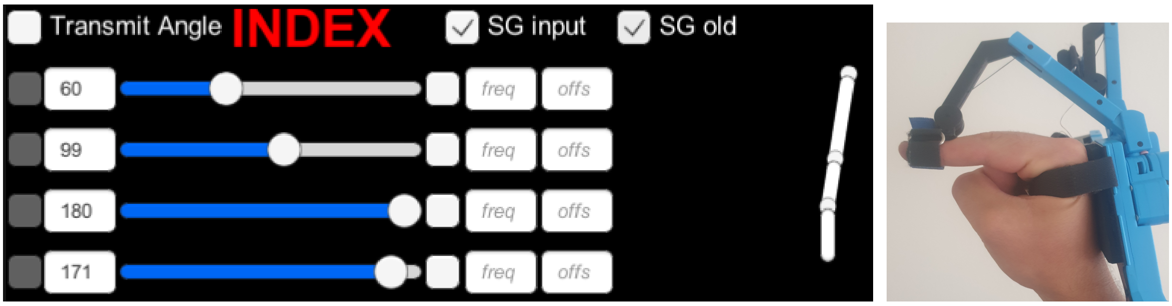


Figure 58: The users index finger is bent $90°$ at the first MCP joint, using the newly written algorithm the derived posture has increased significantly.

Figures 57 and 58. This algorithm performs better at distinguishing between the different joint. The original tended to treat each joint as bending equally as the hand closed. It should be noted that the new algorithm depends more on the dimensions of the users hand since the length of their phalanges are directly used in the algorithm. This means for other users tweaking of these parameters may be required to perform well. Another disadvantage of this algorithm is that the movement is sometimes more jittery and sometimes has trouble noticing a completely extended finger.

5-3-3 Implementing force feedback

The final step of realizing the tactile glove is to implement the force feedback. This is easily implemented in two ways. The first allows manual control via two sliders. The sliders range from 0 to 100, where 0 results in no force being applied at the finger and 100 is maximum force. Figure 59 shows the interface and the user wearing a SenseGlove. In this case the users index finger is prevented from moving further. When not using the sliders, the received force feedback data from the robotic hand can be used to apply force to the user.

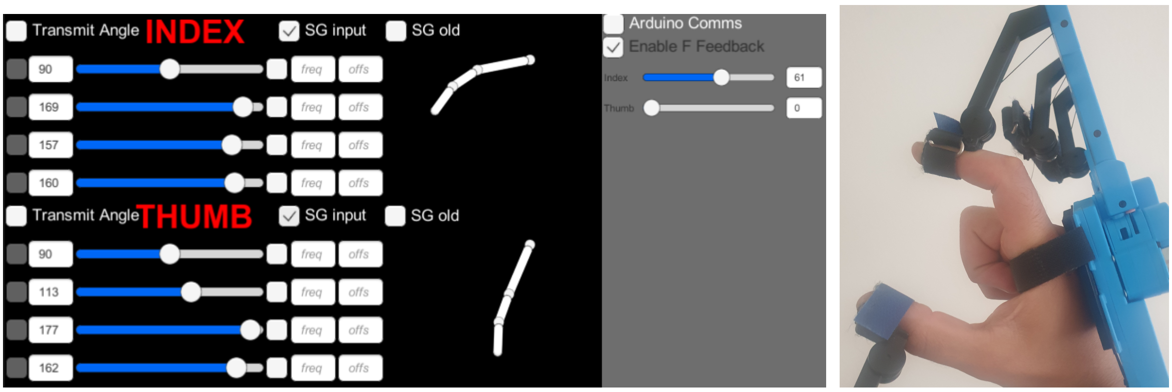


Figure 59: The index finger receives force feedback which prevents the user from moving it further. At the thumb no force is applied.

Chapter 6

Evaluating the robotic hands performance

This chapter deals with experiments conduction using the robotic hands. Two versions of experiments are conducted: experiments where angles were generated in Unity either through manual input or through frequency control, and secondly experiments where the SenseGlove was used to let a user control the robotic hands movements. Using a high speed camera with 240 FPS, high speed footage of the robotic hand was recorded to accurately track each joints position. Marked points on the finger (Figure 60) allowed tracking the coordinates of these points in video editing software (Figure 61)after which the joint angles were derived in MATLAB.

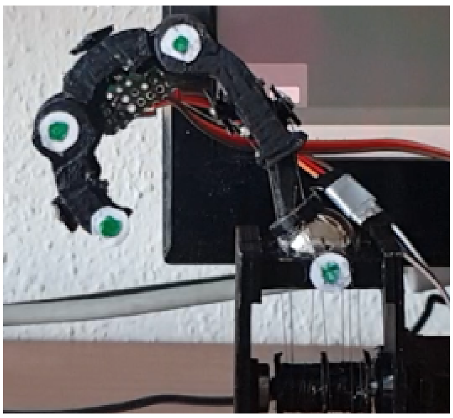


Figure 60: Four points are marked on the finger. This makes tracking these four points easier in the video editing software.

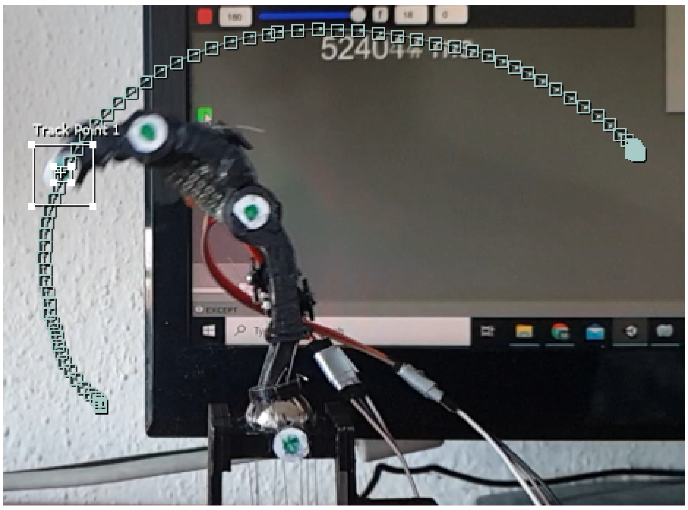


Figure 61: The high speed footage is being analyzed in the video editing software. Currently the marked finger tip is being tracked.

6-1 Experiments using manual control

6-1-1 Opening and closing at maximal speed

The first experiments that was conducted was to simply open and close the finger as quickly as possible. To do this is interface (Figure 52 in Chapter 5) was used to manually input values. With the click of button a command would be sent that flexes all the joints 90°. Using a 240 FPS camera the maximum opening and closing speed is recorded. The results are shown in Figure 62 and 63. In these figures the time that the movement actually starts is taken as the starting time. It should be noted that closing the finger is a tad faster than opening it, this is probably the case due to the tendons undergoing more friction when in the flexed position due to there being more angles. Another interesting observation is that the inner two joints tend to finish their rotation around the same time, whereas the third outer joint takes 30 to 40 ms longer. While a larger bobbin is used for this joint to increase its speed, it is still not as fast as the other joints due to the extra rotation that is required (as discussed in Chapter 4, Section 4-3-1).

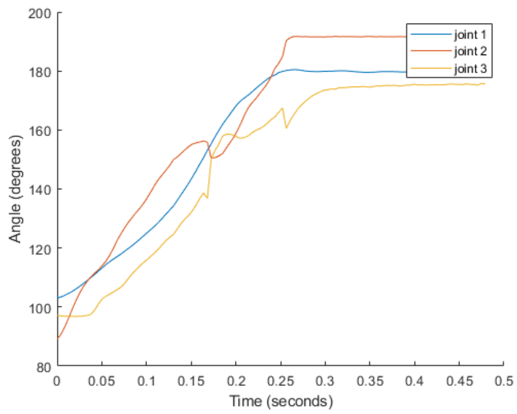


Figure 62: The finger extending its joint for the closed finger position as fast as possible. Both the inner 2 DOF joint (joint 1) and the middle joint (joint 2) take approximately 260 ms complete their rotation, the last joint (joint 3) takes about 300 ms.

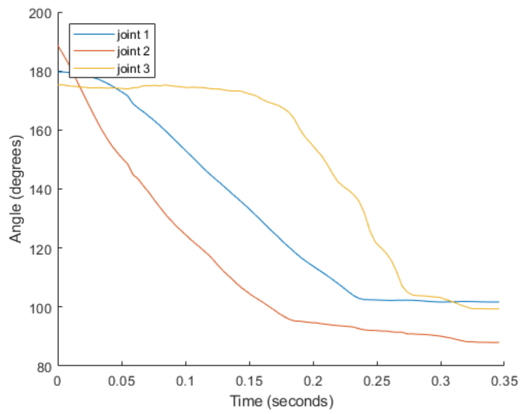


Figure 63: The finger flexing its joint for the open finger position as fast as possible. Both the inner 2 DOF joint (joint 1) and the middle joint (joint 2) take approximately 245 ms to complete their rotation, the last joint (joint 3) takes about 275 ms.

6-1-2 Opening and closing the finger periodically

After having tested the maximum speed that the joints can rotate, frequency control experiments were done. Similarly to the previous tests a high speed camera is used to track the joint locations. Through the frequency control a finger was opened and closed at frequencies ranging from 0.5 Hz to 1.5 Hz.

During recording of the high speed footage the screen was also recorded in the back ground. The screen showed the angle that was at that time being transmitted. This was used to sync the input signal to movement of robot hand. However after doing this there appeared to be a great amount of latency between the input and robotic movements. This was the case because of display latency of the monitor and other devices. To counteract this an experiment was conducted where a button was pressed on the Arduino. When this button was pressed a light on the display would light up. By listening to the high speed footage of this experiment the click of the button could be perfectly timed and the latency from the display to the Arduino was measured. This latency would be in the range of 70 to 100 ms. Using this measured latency an offset was applied to the input signal.

The results of three frequency tests ranging from 0.5 Hz to 1.5 Hz are presented in Figures 64 to 66. Each if the figures shows the paths of the tracked point of the robotic on the left, the angles of the three joints as derived from the footage, and on the right the delay between the input signal sending an angle of 135° and the joint hitting this angle.

For a low frequency of 0.5 Hz the joint angles can fully open and close and the delay between input and output is on average a little over 100 ms. As the frequency is increased to opening and closing the finger at 1 Hz, average delay rises to around 150 ms. Also it appears the outer joint (joint 3) is starting to struggle to complete its full rotation. The amplitude of its movement has decreased when compared to the 0.5 Hz case. This amplitude decrease coincides with a noticeable longer delay than the other two joint at 1.0 Hz. Lastly at 1.5 Hz delay for the first two joints is still around the 150 ms mark, however the thirds joints has increased to approximately 220 ms. Compared to the previous test the amplitude of its motion has decreased even further. This is to be expected because before the joint has reached its final position it is already directed to turn back in this case. For higher frequency this effect will only increase.

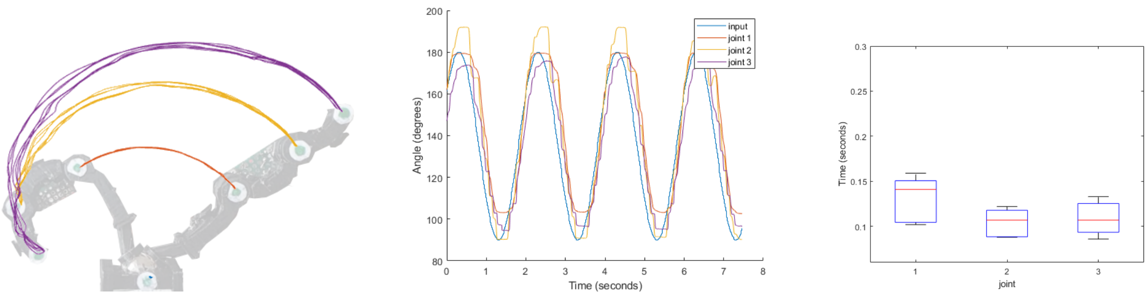


Figure 64: 0.5 Hz experiments

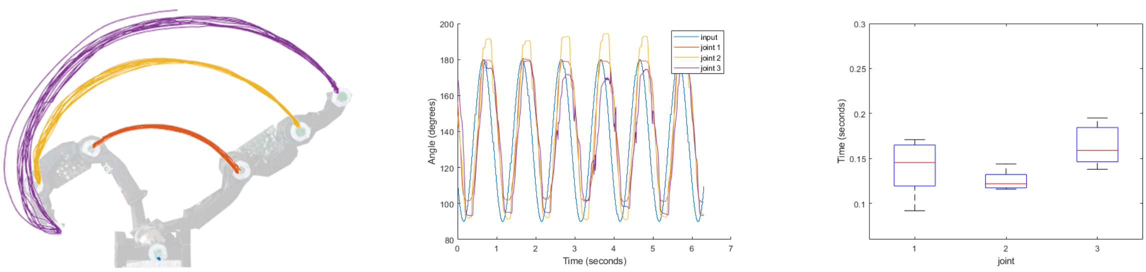


Figure 65: 1.0 Hz experiments

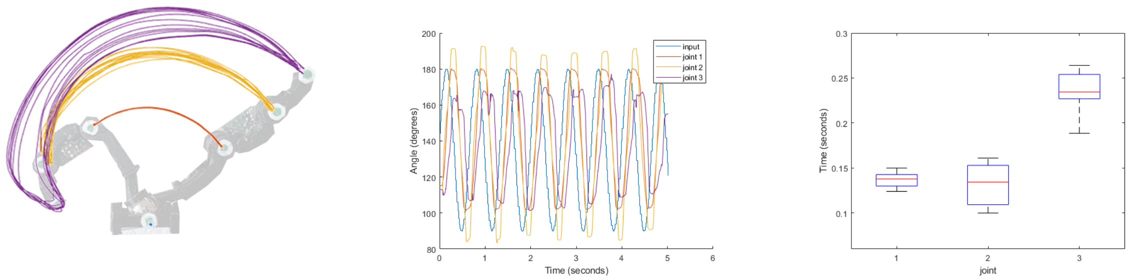


Figure 66: 1.5 Hz experiments

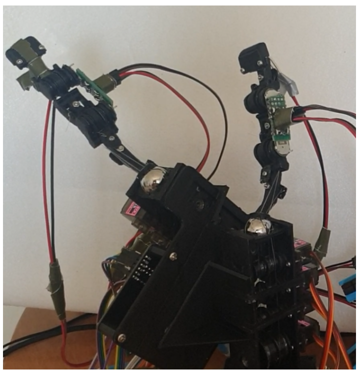
6-2 Experiments using the SenseGlove

6-2-1 Copying hand posture using SenseGlove

Getting the SenseGlove to work correctly with the robotic hand is a hard task by itself. The process is described in Chapter 5. This section first illustrates how well the translation of the SenseGlove angles to the angles of the robotic hand works and then tests where objects are grabbed are conducted.

Figure 67 shows the robotic hand copying the users open hand position. In this position both the thumbs and index fingers joint angles do translate well to the robotic hand.

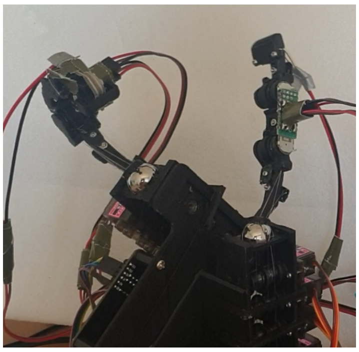
As the user slowly closes their index finger (Figure 68) the two outer 1 DOF joints move well. The 2 DOF ball and socket joint however does not rotate sufficiently. In Figure 69 where the user has almost completely closed their index finger the error is even more pronounced with the angle being almost 45° off. While this problem might at first seem like an error with the robotic hand itself, Figure 70 shows that hand is capable of completely closing its index finger. Here Unity is used directly to input the joint angles. This shows that the translation algorithm used to calculate the robotic hands joint angles from the SenseGlove joints angles is not correct. Getting this algorithm working correctly is a hard task, especially when multiple users with different hand sizes are considered, and is also not the main task of this project.



(a) The robotic hand.

(b) The SenseGlove users hand.

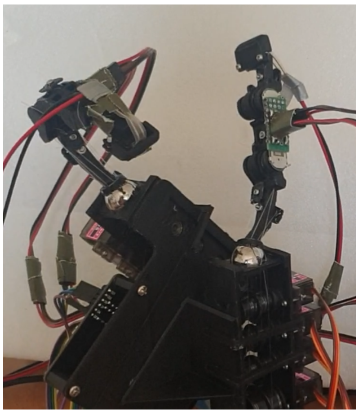
Figure 67: The user has an open hand which the robotic hand copies well.



(a) The robotic hand.

(b) The SenseGlove users hand.

Figure 68: The user is flexing their index finger. While the outer two joint angles translate well to the robotic arm, the ball and socket joints flexion angle is too small.



(a) The robotic hand.

(b) The SenseGlove users hand.

Figure 69: The user has almost completely closed their index finger. The two outer joints angles are acceptable, but the ball and socket joint is almost 45° off.

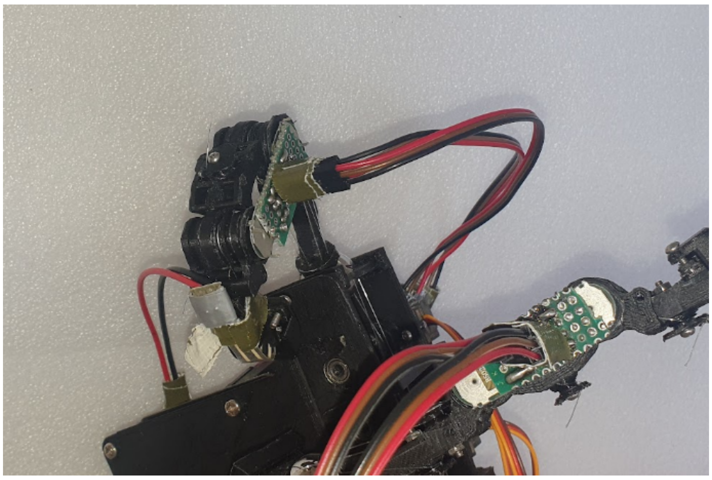


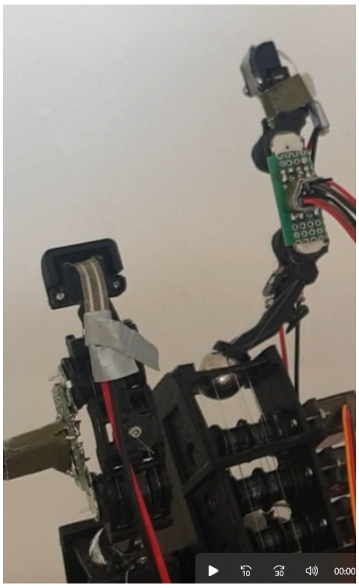
Figure 70: Using Unity to input the joint angles directly shows the robotic hand can close its hand completely.

Figures 71 and 72 show two more cases from the side. In the first case the users hand is almost completely extended with a slight bent in the second 1 DOF joint. This can also be seen on the robotic hand, however here the angle is a few degrees too sharp. In some cases the algorithm has difficulty differentiating between a completely straight for or a finger with slight bent. One of the factors that also influences this is how the user wears the SenseGlove. Sometimes the SenseGlove may be worn a bit higher or lower than at other times. This slight difference causes the SenseGloves joint angles to be slightly different, which may translate to these issues. In Figure 72 the user bends their second 1 DOF joint further. In this hand position the robotic hand performs significantly better.

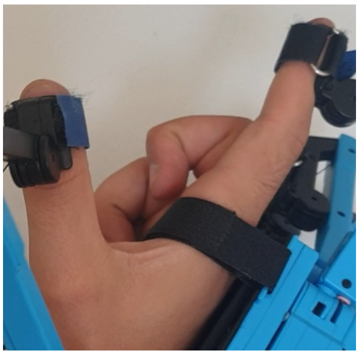
In Figures 73 and 74 the SenseGlove user flexes their thumb from the initial open hand position of Figure 67. In Figure 75 the thumbs outer joint is straightened, after which in Figure 76 the thumb is moved further over the other closed fingers. The first three figures show the robotic hand generally following the users thumbs correctly. In Figure 76 however the robots thumb does not fully flex. This appears to be an issue at the thumbs middle joint. The users middle thumb joint is bend at almost 90° while the robots thumb joint appears to only bend at around 135°. Similarly to the index finger this problem appears to be a problem with translation as Figure 77 shows. Here both the 2 DOF joint and the middle joint are shown to make an angle of approximately 90°.

6-2-2 Grabbing objects using the SenseGlove

In next set of test we take a look at grabbing objects. Initially was tried to grab objects whilst the user was wearing the SenseGlove. Due to the translation from the SenseGloves angles to the angles of robotic being in incorrect positions however, most objects were hard to grab. One of the first objects that was attempted to grab was a big rubber die. This is shown in Figure 78. Next the same rubber die held with the thumb only this time as seen in Figure 79. While the user can hold the die indirectly the size of die is not correctly communicated to the user. This is most likely also caused by the hand palm of the robotic hand being partially missing. The force feedback while holding the die works very well though as the the tip of



(a) The robotic hand.

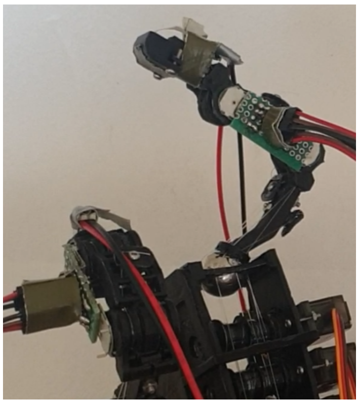


(b) The SenseGlove users hand.

Figure 71: A view from the side. The index finger is almost completely extended, there only is a slight bent. This is present in both the robotic hand and the users hand. However the angle of the robotic hand is a few degrees too sharp.

thumb (where the strain gauge is located), is pressed directly against it.

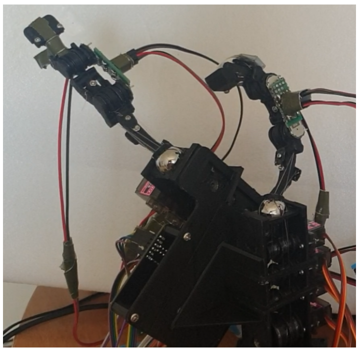
After grabbing the die in two different ways, other more complex objects were attempted to be grabbed. First a cup was attempted to be held and then a pen. Both of these objects were very hard to grab whilst wearing the SenseGlove due to the translation of joint angles not working correctly. Nevertheless it is shown in Figure 80 that holding a cup can still be achieved if the translation of joint angles can be improved. Similarly Figure 81 shows the same for holding a pen. Grabbing an object as small as a pen was impossible with the incorrect translation. Using manual inputs through Unity however this object could still be grabbed.



(a) The robotic hand.

(b) The SenseGlove users hand.

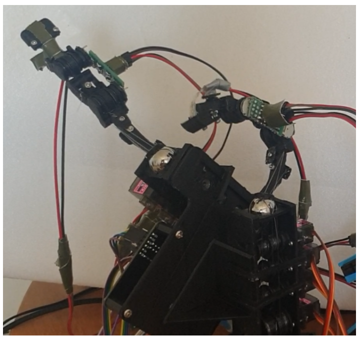
Figure 72: The second 1 DOF joint is flexed to $90°$. This position translates well to the robotic hand.



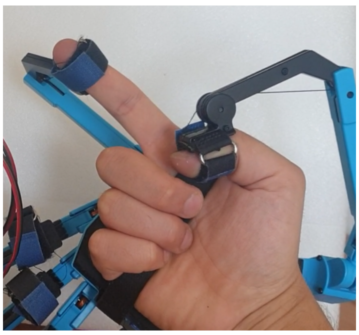
(a) The robotic hand.

(b) The SenseGlove users hand.

Figure 73: From the open hand position the thumb is slowly moved towards the hand palm. This position translate reasonably well to the robotic hand.

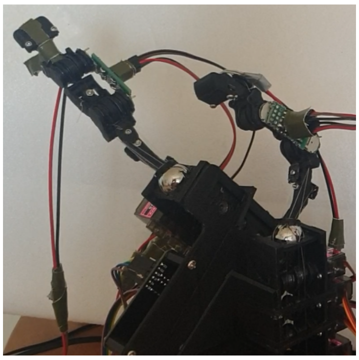


(a) The robotic hand.

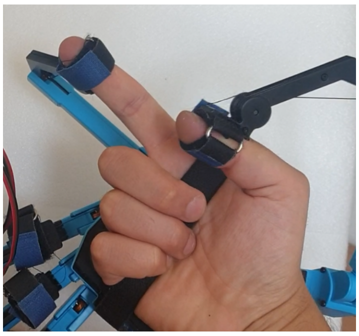


(b) The SenseGlove users hand.

Figure 74: The thumb is brought close to the other fingers. The angles translate well.

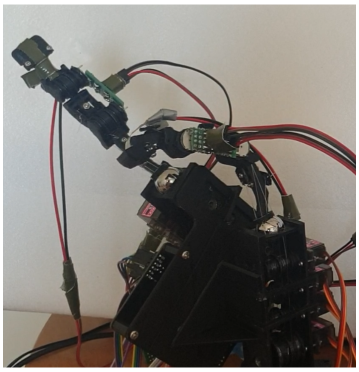


(a) The robotic hand.



(b) The SenseGlove users hand.

Figure 75: The outer IP joint of the thumb is extended, which the robotic hand mimics.



(a) The robotic hand.

(b) The SenseGlove users hand.

Figure 76: When the thumb is closed on top of the other fingers the robotic hand struggles.

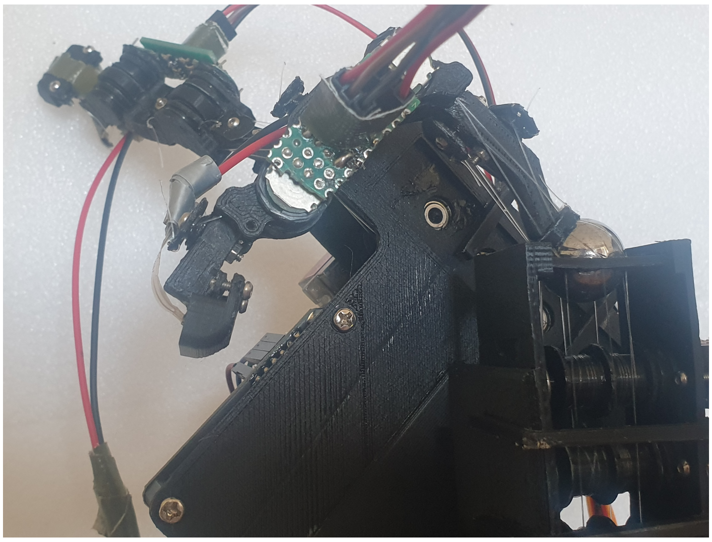


Figure 77: Using manual control of the joint angles through Unity the thumbs can in fact close further.

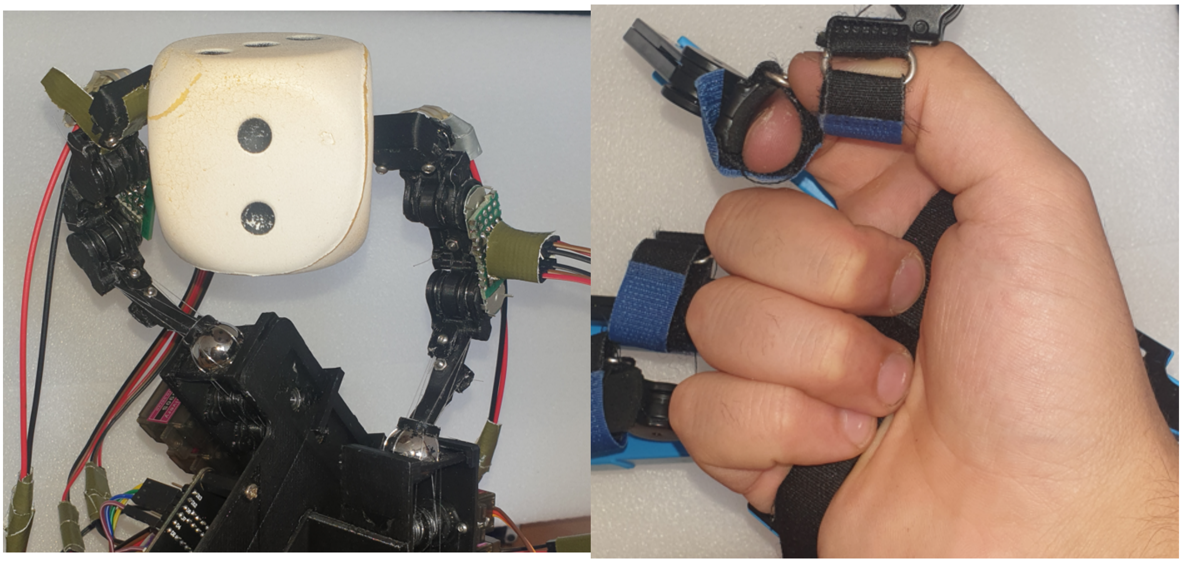


Figure 78: The user is holding a rubber die through the robotic hand. As can be seen the precise hand posture is not correctly copied by the robotic hand, however holding this object is still possible and force feedback is felt that hinders the user from closing their finger further.

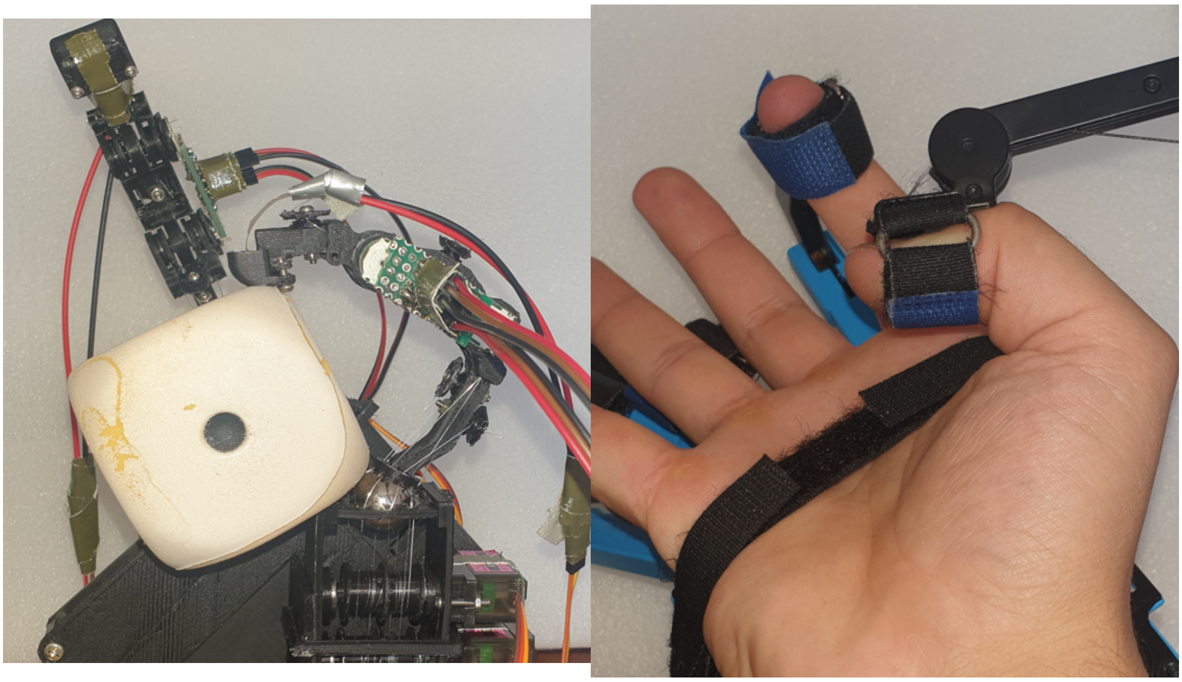


Figure 79: The user is holding the rubber die between their thumb and hand palm. the size of the thumb as felt by the user is too small, however this is also the case because the the correct shape of the hand palm was not designed in this project. Nevertheless during this test the force feedback works exceptionally well because the die is pressed directly against the tip of the robotic finger that contains the strain gauges.

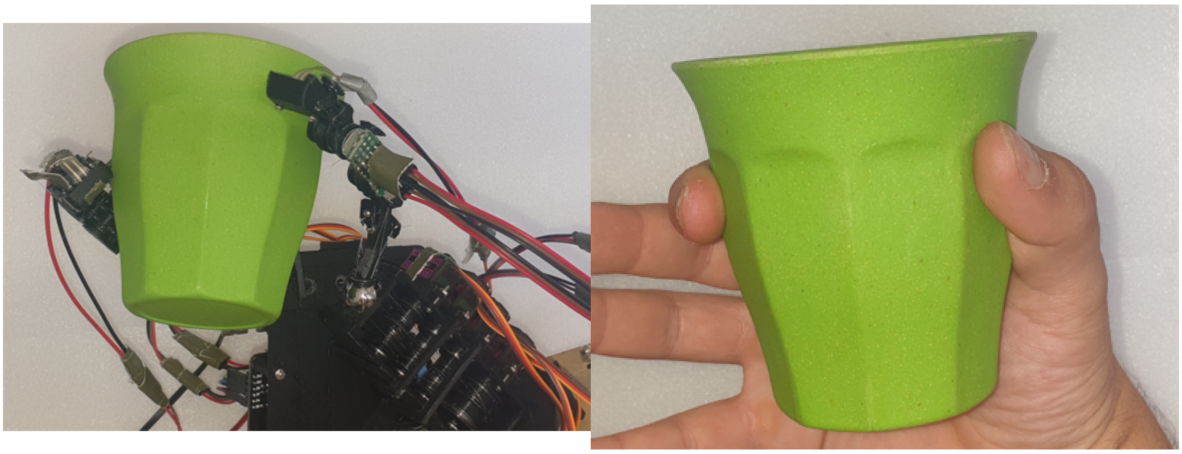


Figure 80: Grabbing a cup with the SenseGlove proved extremely hard due to the translation errors. So instead the cup was hold using manual inputs of Unity. The image shows that the position required to hold the cup can still be attained.

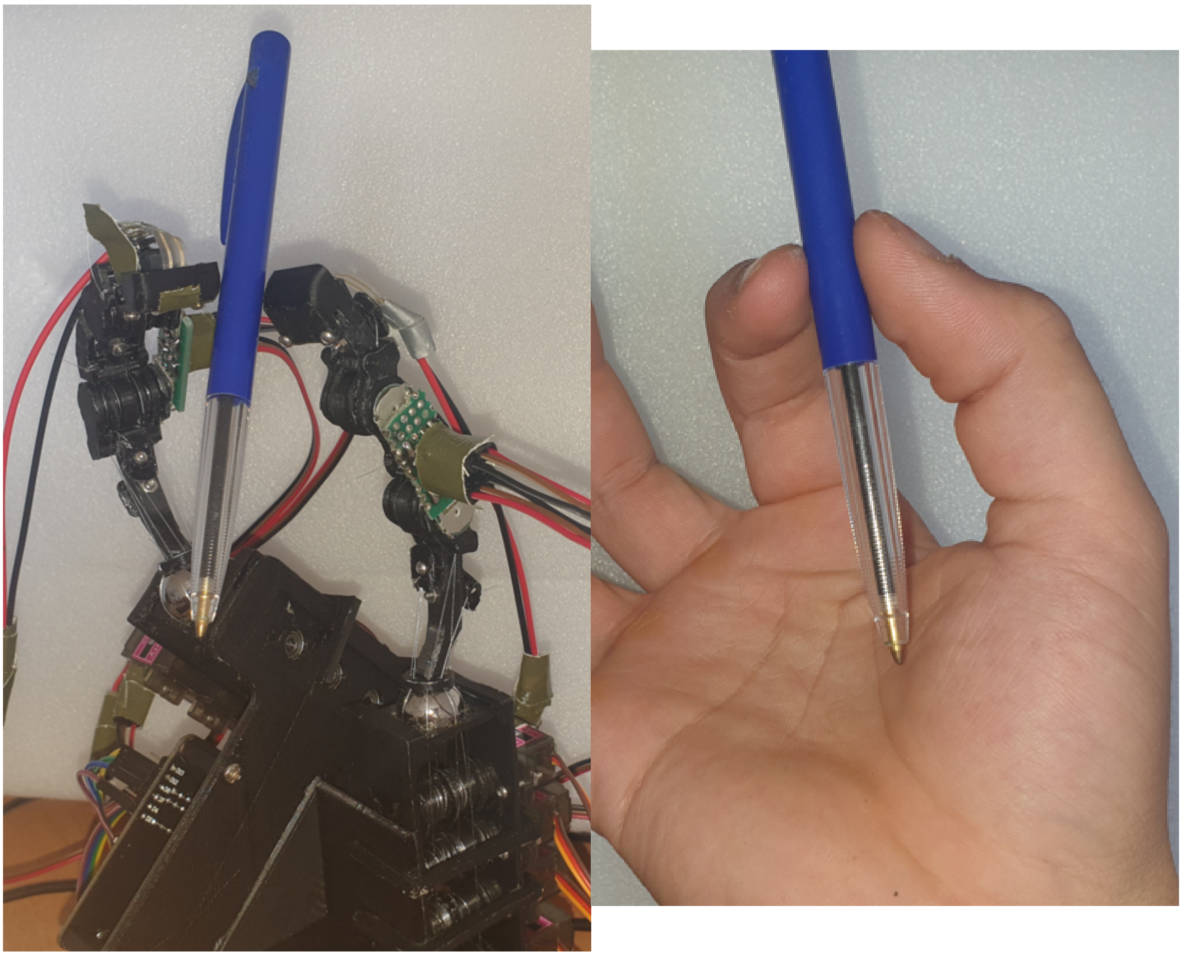


Figure 81: Grabbing a pen with the SenseGlove was impossible due to the small size of the pen and the error in the translation of joint angles. Instead Unity's manual input was used to attempt hold the pen. As shown in the image this was succesful, suggesting that with improved translation holding a pen could work.

Chapter 7

Conclusion

In this project a anthropomorphic, low-cost, fast tactile robotic hand was designed. First in Chapter 2 other robotic hand designs were discussed. It was concluded that many robotic hand designs tend to be underactuated. This generally leads to the robotic hand not being fully capable of moving like the human hand. Other designs often tended to be extremely expensive. This undesirable because it limits research and commercial options. An extremely well known example of this is the Shadow Hand [6] which costs upwards of $60 000 [25] even without actuators.

Chapter 3 discussed in depth the design of the actuation and transmission system to realize movement. First some anatomy of the human hand was discussed. For this a kinematic model was derived with simplifications that would still allow the robotic hand to move similarly to a human hand. For the two outer joints of each finger each tendon is routed along pulleys. This is to improve the accuracy and speed of the robotic hand by ensuring the tension on tendons is unaffected by joint position. Other hands such a the Cable-driven Anthropomorphic Robot Hand [21], discuss these issues, but simply accept the inaccuracies this causes and present no solution to this. Later in the chapter, design possibilities for the 2 DOF joint are discussed. Making sure tension on tendons remains equal across all joint angles is an often ignored problem in other hands. The DART hand [23] presents a solution by routing the tendons through the center of multiple degree of freedom joint. This however comes with drawbacks. In the Thesis a novel solution this problem is proposed: routing the tendons along a hemisphere. This overcomes the spatial issues associated with routing to a single point. The 2 DOF joint is for convenience again depicted in Figure 82.

In Chapter 4 the designs as discussed in the previous chapter are put into practise. The 3D printed parts to allow the tendons to be routed in very specific way are discussed in depth. This is also a part where other robotic hand designs could draw inspiration from. The way tendons are routed have huge effects on friction these tendons undergo as well as efficient transfer of forces. All these factors matter, especially for tactile devices. The chapter ends with the assembly of the complete hand and its costs. The cost, apart from the novel joint and routing design, is perhaps the strongest asset of this robotic hand. With a cost of less than €110, the only other robotic hand that come close is the Open-source, anthropomorphic,

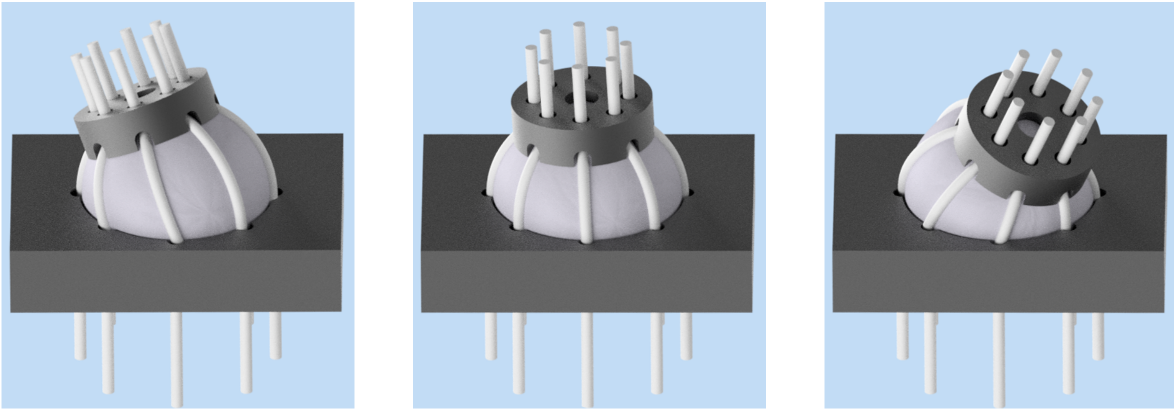


Figure 82: A 2-DoF joint with 8 tendons routed through it in three different angles. Tendons forming a tendon pair are routed on opposite side of the ball joint. The ball joint consists of a half sphere along which the tendons are routed. This ensures the total tendon length of a pair remains unaffected by the angle of the joint.

underactuated robot hand [17]. This hand however has only a single actuator to control 13 degrees of freedom. Our hand contains two fully actuated fingers with each consisting of 4 DOF.

Chapter 5 deals with getting the hand to actually move. This is implemented in two ways, either manually or using the tactile SenseGlove [4]. Manually the user can move sliders or input direct joint angles which will be transmitted to the robotic hand. A second option allows the joint angles to varied according to a frequency of the users choosing. This allows experiments to be conducted where the speed and accuracy of the hand can be tested. Lastly the robotic hand is enabled to be controlled using the SenseGlove. The robotic hand is in this case made to follow the SenseGlove users hand. On top of this pressure sensor attached to the finger tips of the robotic hand provide the user with tactile feedback.

Finally Chapter 6 discussed various tests that were conducted with the robotic hand. The fingers were found to have an opening (finger completely extended) and closing (finger completely closed) speed of less than 300 ms. Then tests using the frequency controls were conducted. Frequencies of 0.5 Hz, 1 Hz, and 1.5 Hz are discussed and the exact motion analyzed. It was shown that at an opening and closing frequency of 1 Hz the outer joint had trouble keeping up. Then, the SenseGlove was used and tested was whether the users hand posture was accurately copied. Due to the algorithm used to convert sensory data from the SenseGlove to joint angles sometimes giving incorrect results, not every hand posture could be accurately copied. Lastly, grabbing objects using the SenseGlove was tested. While not many objects could easily be held accurately (due to the angle conversion algorithm), when objects could be held using the SenseGlove the force feedback performed well to feel when an object was held.

7-1 Future work and additional insights

In terms of costs and novelties I believe this project to have been extremely successful. I truly believe the novelties in joint design and routing of tendons discussed in Chapter 3 and 4, can be implemented by other robotic hands to increase their performance. Though, I believe the superiority of these design choices needs to be evaluated better. This requires mostly better actuators. For this thesis this would have taken too much additional time, but if this work were to be continued I would recommend the use of better motors designed with custom control parameters through software. This is because the used servomotors are not designed to be disassembled as was done in this project. The additional delay between the angular sensor, that was placed on the joints, and the actuator under some circumstances caused jittery motion or sometimes even oscillations. Another thing that would need improving is translation algorithm to derive the human hands joint angles from the SenseGlove sensory data. This is most likely a hard problem to find a well performing solution for, but is required to accurately connect the two tactile devices.

Bibliography

[1] Algorithm to calulate the intersection points of two circles. https://forum.unity.com/threads/calulate-the-intersection-points-of-two-circles.611032/. Accessed: 10-7-2022.

[2] Arduino servo library. https://www.arduino.cc/reference/en/libraries/servo/. Accessed: 10-7-2022.

[3] github repository for this project. https://github.com/2Meric/SG_Hand_Unity. Accessed: 10-7-2022.

[4] Senseglove. https://www.senseglove.com/. Accessed: 10-7-2022.

[5] Senseglove-unity. https://github.com/Adjuvo/SenseGlove-Unity. Accessed: 10-7-2022.

[6] Shadow hand. https://www.shadowrobot.com/. Accessed: 10-7-2022.

[7] Unity. https://unity.com/. Accessed: 10-7-2022.

[8] Konstantinos Antonakoglou, Xiao Xu, Eckehard Steinbach, Toktam Mahmoodi, and Mischa Dohler. Toward haptic communications over the 5g tactile internet. *IEEE Communications Surveys Tutorials*, 20(4):3034–3059, 2018.

[9] Michael J. Vande Weghe* Benjamin H. Brown Jonathan Ko Lillian Y. Chang David D. Wilkinson Sean M. Bidic Ashish D. Deshpande*, Zhe Xu* and Yoky Matsuoka. Mechanisms of the anatomically correct testbed hand. *Transactions on Mechatronics*, 18(1):238–250, 2011.

[10] G. I. Bain, N. Polites, B. G. Higgs, R. J. Heptinstall, and A. M. McGrath. The functional range of motion of the finger joints. *Journal of Hand Surgery (European Volume)*, 40(4):406–411, 2015.

[11] J. Butterfass, M. Grebenstein, H. Liu, and G. Hirzinger. Dlr-hand ii: next generation of a dextrous robot hand. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, volume 1, pages 109–114 vol.1, 2001.

[12] L.Y. Chang and Y. Matsuoka. A kinematic thumb model for the act hand. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 1000–1005, 2006.

[13] Abdulmotaleb El Saddik. The potential of haptics technologies. *Instrumentation Measurement Magazine, IEEE*, 10:10 – 17, 03 2007.

[14] Vineet Gokhale, Mohamad Eid, Kees Kroep, R. Venkatesha Prasad, and Vijay S. Rao. Toward enabling high-five over wifi: A tactile internet paradigm. *IEEE Communications Magazine*, 59(12):90–96, 2021.

[15] Oliver Holland, Eckehard Steinbach, R. Venkatesha Prasad, Qian Liu, Zaher Dawy, Adnan Aijaz, Nikolaos Pappas, Kishor Chandra, Vijay S. Rao, Sharief Oteafy, Mohamad Eid, Mark Luden, Amit Bhardwaj, Xun Liu, Joachim Sachs, and José Araújo. The ieee 1918.1 "tactile internet" standards working group and its standards. *Proceedings of the IEEE*, 107(2):256–279, 2019.

[16] Nestor Jarque-Bou, Margarita Vergara, and Joaquín Sancho-Bru. Estimation of the abduction/adduction movement of the metacarpophalangeal joint of the thumb. *Applied Sciences*, 11:3158, 04 2021.

[17] George P. Kontoudis, Minas V. Liarokapis, Agisilaos G. Zisimatos, Christoforos I. Mavrogiannis, and Kostas J. Kyriakopoulos. Open-source, anthropomorphic, underactuated robot hands with a selectively lockable differential mechanism: Towards affordable prostheses. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5857–5862, 2015.

[18] Dong-Hyuk Lee, Jae-Han Park, Sung-Woo Park, Moon-Hong Baeg, and Ji-Hun Bae. Kitech-hand: A highly dexterous and modularized robotic hand. *IEEE/ASME Transactions on Mechatronics*, 22(2):876–887, 2017.

[19] Pamela K. Levangie, Cynthia C. Norkin, and Michael D. Lewek. *Joint Structure amp; Function: A comprehensive analysis.* F. A. Davis Company, 2005.

[20] C.S. Lovchik and M.A. Diftler. The robonaut hand: a dexterous robot hand for space. In *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, volume 2, pages 907–912 vol.2, 1999.

[21] Sungjae Min and Sooyeong Yi. Development of cable-driven anthropomorphic robot hand. *IEEE Robotics and Automation Letters*, 6(2):1176–1183, 2021.

[22] Tetsuya Mouri, Haruhisa Kawasaki, K. Yoshikawa, J. Takai, and Soken Ito. Anthropomorphic robot hand: Gifu hand iii. *Proc. of Int. Conf. ICCAS2002*, pages 1288–1293, 01 2002.

[23] Nicholas Thayer and Shashank Priya. Design and implementation of a dexterous anthropomorphic robotic typing (dart) hand. *Smart Materials and Structures*, 20(3):035010, 2011.

[24] Daniël Van Den Berg, Rebecca Glans, Dorian De Koning, Fernando A. Kuipers, Jochem Lugtenburg, Kurian Polachan, Prabhakar T. Venkata, Chandramani Singh, Belma

Turkovic, and Bryan Van Wijk. Challenges in haptic communications over the tactile internet. *IEEE Access*, 5:23502–23518, 2017.

[25] Zhe Xu, Vikash Kumar, and Emanuel Todorov. A low-cost and modular, 20-dof anthropomorphic robotic hand: design, actuation and modeling. In *2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pages 368–375, 2013.

[26] Zhe Xu and Emanuel Todorov. Design of a highly biomimetic anthropomorphic robotic hand towards artificial limb regeneration. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3485–3492, 2016.

[27] I. Yamano and T. Maeno. Five-fingered robot hand using ultrasonic motors and elastic elements. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 2673–2678, 2005.