University of Tartu

Faculty of Science and Technology

Institute of Technology

Kasper Kaljuste

Virtual Reality Gloves Based on the LucidGloves Project

Bachelor's Thesis (12 ECTS)

Computer Engineering

Supervisor:

Professor of Distributed Systems Ulrich Norbisrath

Tartu 2023

Abstract/Resümee

Virtual Reality Gloves Based on the LucidGloves Project

The outcome of the thesis was the written process of building a LucidGloves prototype 4 haptic glove and an analysis of what could be improved if someone were to build it again. While a video guide to building the glove exists, there were some issues faced while following it and solutions proposed. A Half-Life: Alyx map was made to gather feedback on the glove. Four test subjects tested the map, the results were analysed and potential improvements were presented, some of which were possible by modifying this prototype or the software, and some requiring a completely new design.

CERCS: T480 - Technology of other products, T120 - Systems engineering, computer technology

Keywords: virtual reality, haptic glove, immersion, force feedback, wearable technology

Virtuaalreaalsuse kindad LucidGloves projekti põhjal

Selle lõputöö tulemusena valmis kirjeldus LucidGloves prototüüp 4 jõu tagasisidega kinda ehitamisest ja tulemuse analüüs. Kuigi õpetus kinda ehitamiseks on video näol olemas, esinesid selle järgimisel probleemid, millele pakuti välja lahendused. Kindale tagasiside saamiseks valmis Half-Life: Alyx mängule testimiskeskkond, mida testisid neli inimest. Tagasisidet analüüsiti ja esitati potentsiaalsed lahendused, millest mõned vajasid selle prototüübi riistvara või tarkvara muutmist ning mõned täiesti uut kinda disaini.

CERCS: T480 - Muude toodete tehnoloogia, T120 - Süsteemitehnoloogia, arvutitehnoloogia

Märksõnad: virtuaalreaalsus, haptiline kinnas, kaasahaaratus, jõu tagasiside, kantav tehnoloogia

Contents

Acronyms

XR - Extended Reality

DIY - Do It Yourself

SC - Sensorimotor Contingency

VR - Virtual Reality

HMD - Head Mounted Display

DLC - Downloadable Content

HL:A - Half-Life: Alyx

1 Introduction

The Computer Graphics and Virtual Reality Study Lab at the University of Tartu supports the development of immersive extended reality (XR) experiences [1]. One big part of immersion is haptic feedback and spatial correctness, which can be achieved using different ways of interacting with virtual objects. Due to the educational setting, students can experiment with these various modes of interaction in the lab, and do-it-yourself (DIY) solutions are a good teaching tool to test and understand the technology behind them. This thesis aims to document the process of building such a device to increase immersion, which could be improved upon or recreated.

1.1 Immersion in Virtual Reality

Immersion in virtual reality can be characterised by the sensorimotor contingencies (SCs) they support. SCs refer to the actions that we know and perform to perceive the world [2], for example, moving your head to change the gaze direction or reaching your hand out to grab something. So the more meaningful SCs a VR system supports, the more immersive it is. Immersion breaks when the user moves his head, but the visual image does not change as it would in the real world or reaches out a hand which then goes through an object. All head-mounted displays (HMDs) from the major consumer electronics companies nowadays provide an immersive experience [3]. Tracking the user's hand movements is usually the next priority in developing an immersive VR experience. Most consumer VR systems have controllers that provide six degrees of freedom (6DoF), allowing users to see their hand position and interact with virtual objects. The combination of head and hand tracking allows many SCs to be carried over into the virtual world. Still, conventional controllers do not convey realistic hand posture and gestures in the virtual world, which can break immersion. [4]

The design of a virtual reality controller is similar to classical controllers with joysticks, buttons and triggers, so users cannot perform all SCs they would in the real world to interact with objects in the virtual world. This is partly solved by hand tracking, which has made massive improvements over the last few years. For example, hand tracking allows for natural conversational gesturing. Still, the lack of force feedback and weight of objects makes it feel imprecise and challenging to use, which is supported by findings showing that hand tracking with the Leap Motion Controller scores lower on the System Usability Score than the Oculus Touch controller. [5, 6] Hand tracking is quite advanced, as seen with the Quest 2 and Quest Pro; therefore, a comparison with the presented solution is necessary (in section 3.1 I compare Meta Quest 2 hand tracking to the proposed solution).

1.2 Related Work

VR has reached many consumers with nearly 20 million Quest headsets sold [7]. For example, 12 years ago, a head-mounted VR display with 6DoF head and hand tracking could cost tens of thousands of euros, but a similar system today can only cost hundreds of euros. [8] This has resulted in more hardware and software available for VR and different haptic feedback solutions for hand tracking.

1.2.1 Further Development of Controllers

One possibility to increase immersion is the advancement of controllers. For the existing HTC Vive controller, Stellmacher, Bonfert, Kruijff, and Schöning [9] made an enhancement that changed the required force of pressing the controller trigger according to the object being held. Holding a heavier object required more pressure from the user to press and hold down the trigger than holding a lighter object. This feedback method allowed the user to distinguish light objects from heavy objects intuitively, but the feedback was not expressed within the whole finger movement, only within the trigger. [9] Providing feedback to one or more fingers using a controller does allow for better immersion, but having no controller at all would be more immersive, as said by Xueni and Antonia [4]: it allows for natural conversational gesturing and richer hand motion, which a controller providing feedback to partial finger movement is not able to do.

1.2.2 Virtual Reality Gloves

To track hand movements without holding something but still providing feedback depending on user actions, haptic gloves can be used.

Commercial Gloves

SenseGlove is a company that originates from the graduation project of its two founders, Johannes Luijten and Gijs den Butter, at the Delft University of Technology. The first working prototype was created in 2017 and nowadays the company sells a haptic glove kit called the SenseGlove Nova (Figure 1.1). [10] The glove provides force feedback for four fingers from the thumb to the ring finger by applying resistance through its magnetic friction brakes and vibrotactile feedback with voice coil actuator technology to the thumb and index fingers [11].

Figure 1.1: SenseGlove Nova [11]

The SenseGlove Nova is on the cheaper side of commercial force feedback haptic gloves, costing 4999 $, with the more expensive options like the Dexmo glove costing 36000 $ [12]. Being closed source and expensive these are not ideal for the educational setting of the Computer Graphics and Virtual Reality Study Lab at the University of Tartu.

DIY Gloves

An alternative to expensive commercial haptic gloves is DIY gloves or gloves made as a result of research.

Uddin, Zhang and Wang [13] produced a glove with a pneumatic cylinder and membrane pump that provided realistic feedback when touching virtual objects. The pneumatic cylinder was attached to the glove on the back of the hand and, through a linked structure, allowed manipulation of the index finger movement. A diaphragm pump was located on the palm side of the hand, which was connected to a bladder on the inside of the index finger. Inflating the bladder simulated haptic feedback. The pneumatic cylinder and diaphragm pump were controlled by a solenoid valve using pulse width modulation. [13] Abad, Ormazabal, Reid, and Ranasinghe [14] achieved feedback using a different method. Instead of a pneumatic cylinder, they used retractable card holders with a cord attached to the tip of the fingers, where the cord could be restricted from coming out by a solenoid. There was no diaphragm pump type of feedback on this glove, but instead, there was a Peltier element at the tip of the index finger, which sent a cold or warm signal to the element depending on the object's temperature. In addition, a vibration actuator at the end of each finger was activated when the object was touched. [14]

As a result, the pneumatic glove [13] provided feedback only to the index finger but had the possibility to extend the functionality to the other fingers. The pneumatic glove [13] did not monitor the entire hand position, but the glove with retractable cords [14] used a Leap Motion controller to determine the hand position in a virtual environment.

The retractable cord glove [14] took inspiration from the LucidGloves [15] project, also using retractable card holders to provide feedback to fingers. The fourth prototype of the LucidGloves project seen on Figure 1.2, however, uses servo motors instead of solenoids to provide feedback to the fingers. The retractable card holder cord is connected to a potentiometer to determine the finger's position. Unlike the glove by Abadi *et al.*, prototype 4 of the LucidGloves project has no vibration or temperature feedback. As this is an open-source project, there is a parts list and instructions for glove assembly available. The cost per glove to make the prototype is also given, which is around $30. Being open source and low-cost means that this project is accessible for students to contribute and develop the project further; therefore it was also chosen as the base of this thesis.

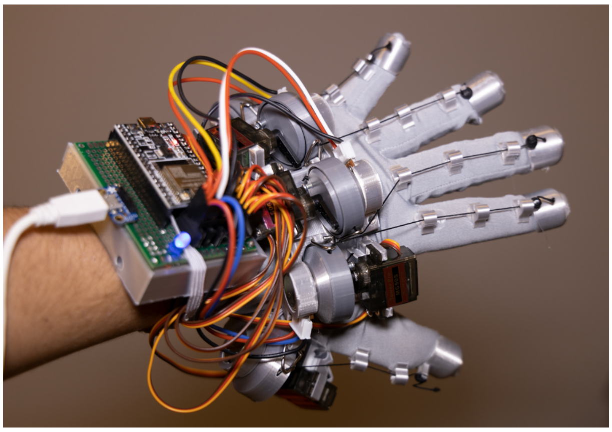

Figure 1.2: LucidGloves Prototype 4 [15]

1.3 Problem Overview

Because consumer VR headsets become more and more accessible, the advancement of technology behind making them more immersive is also intensifying. A consumer-oriented virtual reality headset includes controllers to interact with the system, which do not provide realistic haptic feedback to users. A glove instead of a controller or hand tracking could increase immersion by providing realistic feedback and allowing for more natural hand motions. This thesis aims to study the challenges involved in building a haptic glove based on the LucidGloves prototype 4 to make the process easier for future students who wish to develop haptic gloves further or understand the aspects of building one in the context of VR education.

2 Implementation

The design of the LucidGloves prototype 4 consists of smaller design sections, which the author of the glove, Lucas, explained in the video tutorial [16]. The video guide explained the parts required, assembling the haptic modules and the glove, wiring, and software to get the glove working. The working principle of the glove is as follows: the development board reads values from the potentiometers (finger position), which it sends to the OpenGloves driver emulating controllers, and when grabbing objects, the driver sends a command back to the development board to activate force feedback. I separated the design a little differently than described in the video. At first, I wanted to get the software with only the potentiometers working. After that, I planned on mounting and wiring the potentiometers on the glove as I knew the finger tracking would work by then, adding haptics and finally testing the glove. The following is an overview of what I did and the problems I faced when building the glove. I also highlighted the modifications I made in comparison to the video tutorial.

2.1 Software Setup and First Test

At first, I set out to test the development board with five potentiometers as inputs. I used the ESP32 Devkit V1 development board by Espressif Systems, which uses the ESP32-WROOM-32 module, instead of an Arduino Nano because it allowed me to use Bluetooth instead of USB serial. Most ESP32 boards use the Silabs CP210x USB to UART serial converter chip, which requires an additional driver as explained in the firmware setup and customisation tutorial on the LucidGloves Github wiki page [18]. After installing the driver, I added the ESP32 package to the Arduino IDE and downloaded the main build from the LucidGloves GitHub page [19]. I used a breadboard to connect the potentiometers and changed the default pin definitions in `lucidgloves/firmware/lucidgloves-firmware` because my development board differed from the one used in the video. After that, I flashed the build onto the board. As the default setting used USB serial, the potentiometer values were shown in the Arduino IDE serial monitor. The values changed as they were supposed to.

With the potentiometer test setup complete, the OpenGloves Force Feedback Demo in SteamVR could be tested for finger movement. I downloaded the OpenGloves base application and Force Feedback Demo [20, 21]. I launched OpenGloves from the desktop, set up the settings to enable left (or right) hand only in the settings, disabled force feedback as I had not implemented it yet, saved the settings, closed OpenGloves, launched OpenGloves again but in Force Feedback Demo mode with the Meta Quest 2 connected to the PC and rotated the potentiometers. The OpenGloves driver emulates the Steam Index Knuckles controllers, which support separate finger tracking. As a result, each finger on the left hand could be moved separately, as opposed to the Meta Quest 2 controller in the right hand. I repeated the same process with Bluetooth serial. I replaced the COMM_SERIAL definition in `lucidgloves/firmware/lucidgloves-firmware.ino` to COMM_BTSERIAL and changed the communication method in the Steam OpenGloves application to Bluetooth serial. Then I paired the ESP32 to the PC, and the demo worked using a battery pack to power the ESP32. The video of the result is in the appendices section 6.3.

Each of the fingers worked in the OpenGloves Force Feedback Demo in both USB and Bluetooth mode when rotating the corresponding potentiometers, so the next logical step was to start the process of mounting the potentiometers on the glove, which required printing and assembling parts from prototype 4 and 4.1.

2.2 Retractable Spool Mechanism

With the software set up, I printed the required parts for finger tracking. I did all the printing with PLA and the recommended printer settings specified in the printing guide [22]. I took the parts from the hardware folder of the main build [23]. As the first step, I made the mechanism for the retractable spool. This required printing the following parts:

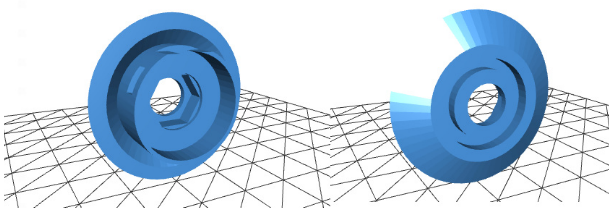

Figure 2.1: Tensioner for Potentiometer [24]

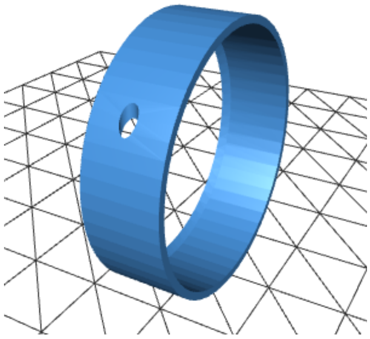

Figure 2.2: Spool Cover [25]

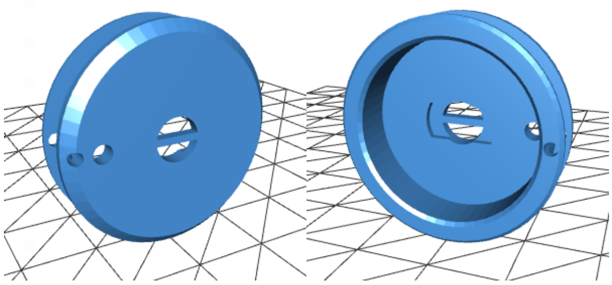

Figure 2.3: Haptic Spool [26]

The printer I used was a Prusa i3 MK3S. Each part printed successfully, but with the haptic spool and tensioner assembled, the spool cover did not click onto the tensioner as the brim of the tensioner had too many imperfections and broke off easily. The printing was repeated twice with more or less the same results. I then tried to print again using another 3D printer, the Weedo Tina 2. Before printing, I decided to switch the potentiometers to smaller ones, as they could be connected using a JST-XH-3Y connector, making the wiring more modular when reaching that part of the design process. This meant switching the tensioner to the one on Figure 2.4. Printing the three parts with the Weedo Tina 2 produced better results, as the spool cover clicked into place on the first try.

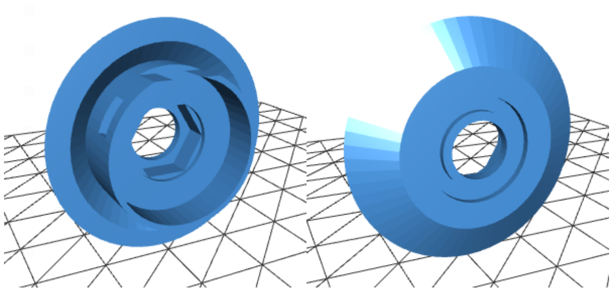

Figure 2.4: Tensioner for Smaller Potentiometer [27]

Next I disassembled a retractable badge reel with pliers, from which I extracted the string and torsion spring. I put the torsion spring into the tensioner and a nut into the tensioner slot. I attached a potentiometer from the other side, twisted the end of the torsion spring along its axis, and put the end into the potentiometer knob slot. I inserted the string from the disassembled badge reel into the haptic spool from the inside, with the knot holding the string in place. Then I pushed the haptic spool into the potentiometer knob slot and finally clicked the spool cover into place with the string coming out of the spool cover opening. I then wound up the string by turning the spool cover and pulling it allowed it to retract, although with limited speed. I solved this by disassembling the mechanism and lubing the potentiometer with WD-40. To achieve haptic feedback, the spools had screw holes in them. in the video guide Lucas recommended the M2x10 mm screws that came packaged with the servo [16], but because they fell out easily, I used M2.5x8mm socket screws which I could screw in further thanks to the flat ends.

The retractable spools required a high-quality print because the spool had to click on the tensioner. I did not succeed in printing these parts with the Prusa i3 MK3S and used the Weedo Tina 2 instead, which gave better results. Alternatively, one could also decrease the printing speed, which could increase the precision and quality of the print. In total, I made five working retractable spool mechanisms.

2.3 First glove

After assembling five retractable spools, I printed five slide holders, one rigid mount, five end caps and ten guide nodes.

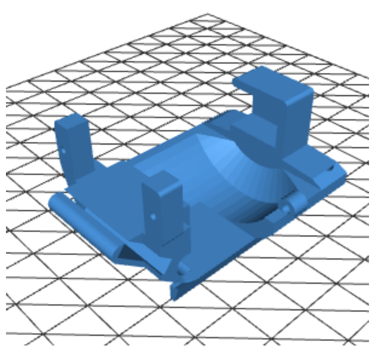

Figure 2.5: Smaller Potentiometer Slide Holder [28]

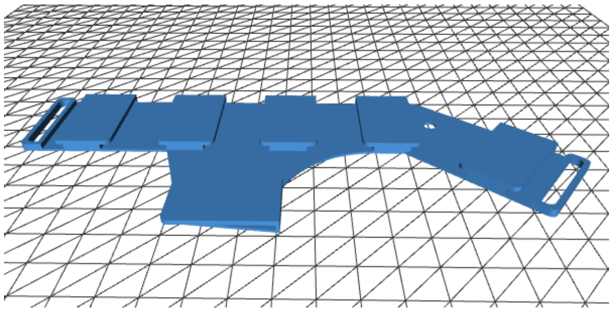

Figure 2.6: Left Hand Rigid Mount [29]

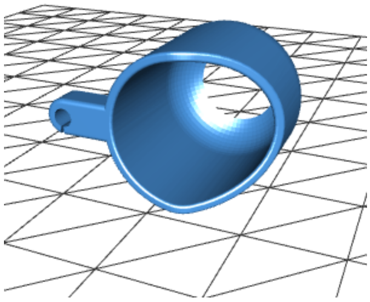

Figure 2.7: Finger End Cap [30]

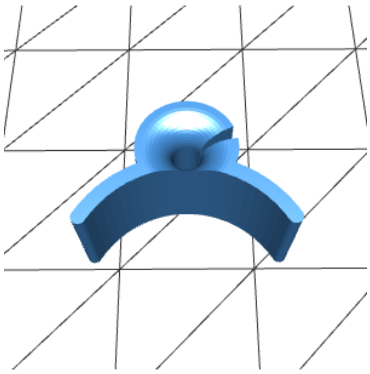

Figure 2.8: Guide Node [31]

I installed the previously assembled retractable spools on the slide holders, which I printed using the Weedo Tina 2. I put the metal loops that came with the badge reels on the slide holders to guide the string. Then I mounted the slide holders on the rigid mount by sliding them into the slots. As the Weedo Tina 2 print bed was too small for the rigid mount, I used the Prusa i3 MK3S instead. Using an elastic waistband, I temporarily fixed the rigid mount with retractable spools in place to a PU-coated textile glove. On the underside of the rigid mount, I cut a foam rubber to size and hot glued it on. I tied the strings onto the end caps, which I separately scaled before printing to fit each finger. I printed multiple iterations to ensure the best fit for my fingers. The strings needed guide nodes on fingers to ensure smooth lengthening and retracting. I hot glued two guide nodes on each finger, except the pinky finger, which had one guide node. The result was unsatisfactory as the glove was not rigid enough and hot glue did not stick well. The end caps and guide nodes detached easily, so in the next iteration, I used another type of glove, which would fit my hand better, be more rigid and allow hot glue to stick better. The first iteration of the glove can be seen on Figure 2.9.

Figure 2.9: First Glove Iteration

2.4 Second glove

In this implementation step, I aimed to install the 3D printed parts and the Quest 2 controller securely onto the glove.

Modification 1: The Type of Glove

As the PU-coated textile glove was not rigid enough and hot glue did not stick well, I needed to change the type of glove for the next prototype. The parts list [17] recommended nylon inspection gloves or cycling/golf gloves, but no exact model was specified. As I already had the TAMREX PRO6000 tactical glove, which was a good fit for my hands, I used this. The tactical glove made out of PU-coated synthetic leather allowed the gluable parts to stick better and was more rigid overall.

Modification 2: Guide Node on the Slide Holder

On the rigid mount, I turned the index finger potentiometer slide holder around to make the string travel at a straight line parallel to the finger instead of an angle. I decided to add an additional guide node with the wings snipped off on this slide holder because otherwise, the string rubbed against the slide holder edge in a way that restricted the retraction of the string.

I printed some of the guide nodes again because they had hot glue on them and when I tried to peel it off a couple of them snapped in half. I also resized and printed some new finger end caps to make them fit better on this glove. Instead of the 9 guide nodes on the previous glove, I printed additional guide nodes and placed them on some fingers so the string could travel in a straight line. The guide node placement can be seen in Figure 2.10.

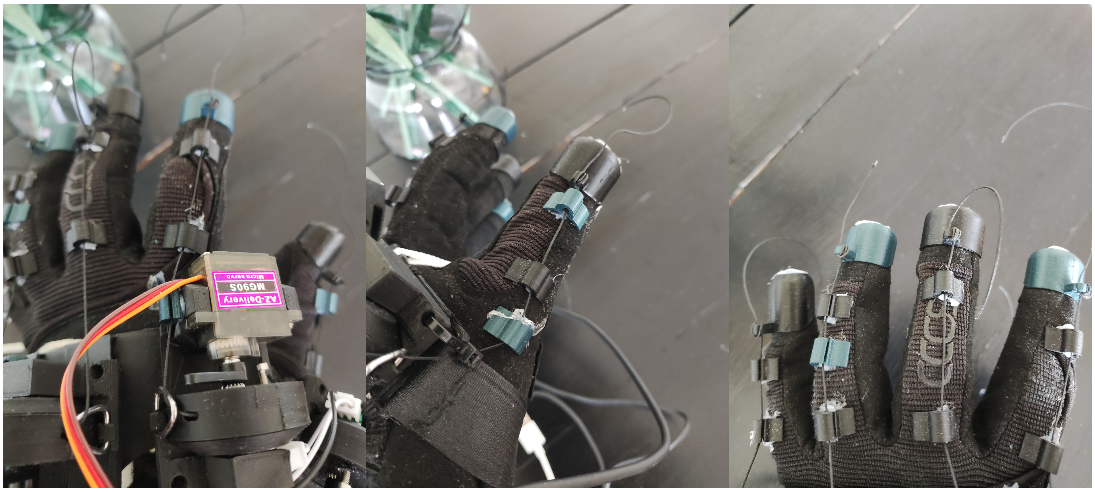

Figure 2.10: Second Glove Iteration Guide Nodes

Modification 3: Sewing of the Velcro Strip

To stop the rigid mount from moving forward and up when bending fingers towards the wrist, I sewed a velcro strip on the underside of the glove and on the sides of the rigid mount in a way that allowed tightening the rigid mount. This allowed me to fine-tune the position of the rigid mount on the glove and provided a more secure fit than the one in the video guide, which only tightened the sides of the rigid mount together. The result can be seen in Figure 2.11.

To track hand position I printed the Quest 2 controller mount, slid it on the rigid mount slot and put the controller in.

Figure 2.11: Second Glove Iteration Strap

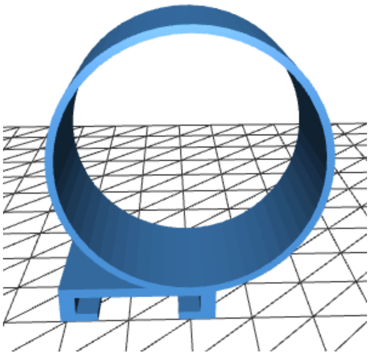

Figure 2.12: Quest 2 Controller Slider Mount [32]

Modification 4: Zip-tying the Controller Mount

When the controller was in the controller mount, it was wobbly if I moved or rotated my hand. I also noticed the lack of accuracy during fast movements when testing the glove in the Force Feedback Demo. To fix this, the slide holders had slots in them and I zip-tied the middle and ring finger spool mounts together and to the controller mount. This made tracking the hand more accurate as the hand position was only dependent on the controller, which did not wobble as much during fast movements anymore.

The TAMREX PRO6000 tactical glove was overall a better fit for my hand, the retracting mechanism was smooth, I got the parts printed and the Quest 2 controller was successfully installed.

My own design decisions were the choice of a glove (1), to sew the velcro strap on the glove so it would be more secure (2), and although Lucas talked about the idea of turning some of the slide holders around in the video, I had to make the decision based on my build. I also modified one printed guide node and hot-glued it onto the index finger slide holder, to make the string retraction smoother (3), and zip-tied the controller mount to two slide holders (4).

2.5 Wiring

The goal for this section was to get the ESP32 mounted on the glove and the potentiometers wired. To test if finger tracking worked I tested it with the Force Feedback Demo.

After assembling the mechanical parts of the glove the next step was to wire the potentiometers. To approximately know the length of wires, I had to think of the mounting mechanism for the development board. One possibility was to mount the development board to the wrist, but raising the back of the hand towards the back of the forearm would have resulted in the controller mount colliding with the board and moving the board further up the forearm would have made wiring awkward and unnecessarily long. An alternative that was also used in the video tutorial was to mount the board to the velcro straps in the palm of the hand, but that would have restricted finger movement when closing the user's hand as seen on Figure 2.13. In addition, this method would have not made much sense because my velcro strap modification (3) would have made it hard to tighten the rigid mount with the board in the way.

Figure 2.13: Pose for When the Board Would Restrict Finger Movement

Modification 5: Mounting the Development Board

To keep the wire length short and not dependent on the position of the hand, I chose to have the board mounted on the controller holder on the top of the glove with zip ties through the mounting holes on the board.

On the first attempt at wiring the glove, I used the daisy chain wiring method [33], as suggested in the video guide, to keep the wire management cleaner than each power and ground connection starting from the board. I cut the wires to length, used a Dupont style 2.54 mm female connector on the board and to connect to the potentiometers I used JST-XH 2.54 mm 3-pin connectors. The development board I used with the ESP32-WROOM-32 microcontroller had the following notice in the datasheet:

* Pins SCK/CLK, SDO/SD0, SDI/SD1, SHD/SD2, SWP/SD3 and SCS/CMD, namely, GPIO6 to GPIO11 are connected to the integrated SPI flash integrated on the module and are not recommended for other uses. [34]

So the GPIO numbers of the pins that were not recommended were 9, 10, 11, 6, 7, 8. With that in mind, I defined the final board side connections in the code:

```
#define PIN_PINKY    32
#define PIN_RING     35
#define PIN_MIDDLE   34
#define PIN_INDEX    39
#define PIN_THUMB    36
```

I daisy-chained the power and ground connections meaning that I crimped two wires to one JST-XH 2.54 mm connector and inserted 2 of those into the 3-pin connector. Testing the connections using a multimeter indicated that one power connection was faulty, so I repeated the crimp for that connection. Testing the glove in the Force Feedback Demo showed that multiple fingers were inaccurate and looking at the serial monitor the values were indeed jumping around even when not moving any fingers. Crimping two wires into one JST-XH 2.54 mm connector was unreliable as the connectors kept breaking and it was difficult to get working connections, so I abandoned this idea.

I rewired the power and ground lines using one power and one ground wire coming out of the board and soldered it to diverge into five separate wires to connect to the JST-XH 2.54 mm 3-

pin connectors. Rewiring each potentiometer power and ground connection separately resulted in less noisy serial monitor readings and the demo had more accurate finger tracking. Faulty connections were still present and ruined the calibration of some fingers. In addition, the spools did not retract the string fast enough, so the tracking was inaccurate. The potentiometers had to be lubed more thoroughly so I repeated the lubing process with WD-40 for each potentiometer again.

Another problem was that the hand model in the Force Feedback Demo did not allow for certain finger movements. The maximum extension of a finger in the demo was less than a real finger could extend. The calibration was set to always calibrate by default in the `lucidgloves/firmware/lucidgloves-firmware/AdvancedConfig.h` file meaning fully extending fingers and making a fist gave the potentiometer edge values that defined the range of motion. Real-life fingers had a bigger range of motion than that of the hand model in the Force Feedback Demo and it mapped the realistic bigger range to the smaller range on the demo.

Modification 6: Calibrating String Length

To solve this I tied the ends of the strings to the fingers so that the retraction from the spools started at the point in which the demo's maximally extended fingers matched the real finger's position and extending the fingers further did not retract the strings but allowed them some slack.

With these problems solved I crimped the faulty connections once again, verified in the demo that they worked and moved on to add haptics to the glove. My own design decisions were the mounting place of the board (5) and the calibration method of the string lengths (6).

2.6 Adding Haptics

To achieve haptic feedback on the glove I installed servos in the slide holder slots opposite of the spools. As the servos required 4.8 V to 6 V input and the development board maxed out at about 3.6 V in addition to not providing enough output current for 5 servos, I needed an extra power source [34]. I stripped a USB type-A to micro-B cable at the micro-B end and crimped the power and ground wires to two 1-pin Dupont style 2.54 mm connectors. Then I connected this in turn to a power bank.

Modification 7: Servo Wiring Method

For the power and ground connections, I soldered pin header strips together on one end and covered them with heat shrink tubing. After that, I defined correct servo pins in `lucidgloves/firmware/lucidgloves-firmware.ino`, which were all next to each other so I could wire the servos as seen in Figure 2.14.

Overall the wiring process included a lot of trial and error mainly because I could not get reliable connections with the daisy chain method. I got the glove to a point where the servos were wired, the power and ground wires for the potentiometers came from the development board and were split into five, the mechanical part was done and finger tracking worked after tying the strings to be at a correct length. With the wiring and mechanical part done, I could continue to get the haptics working.

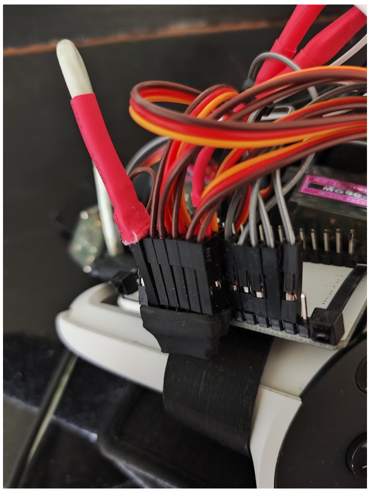

Figure 2.14: Servo Wiring

2.7 Software

This part of the implementation process focused on getting the haptics and a working test environment for the glove ready. The goal for the test environment was to be in the Force Feedback Demo, but later on, I made the switch to Half-Life: Alyx (HL:A) instead.

The first step was to enable force feedback in the `lucidgloves/firmware/lucidgloves-firmware.ino` file by changing the following define to true:

```
#define USING_FORCE_FEEDBACK true
//Force feedback haptics allow you to feel the solid objects you hold
```

The next step was to enable force feedback in the OpenGloves driver application. The functions tab in the OpenGloves application had two functions: to retract and extend the servos fully. The instructions in the application to calibrate the servo hands were as follows: ”As a guide, extend the servos fully, then place the servo horn to where you want the fingers to be fully restricted. After, retract the servos. They should not impede in any finger movement.” After calibrating the servos I launched the Force Feedback Demo, but the servos did not restrict the movement of the fingers nor do anything at all when grabbing objects. The power bank that supplied power to the servos shut off after a while when having a low current output, so it had to be switched on again. The servos then restricted the finger movement when grabbing objects, but the feedback restricted the finger motion so much that no objects could be grabbed. In the Discord LucidVR Tech Server support forum, some people had the same problem, but no solution was presented at the time of testing. I tried to switch to the Opengloves Beta branch in Steam but the same problem persisted. In the `lucidgloves/firmware/lucidgloves-firmware.ino` file changing the following define to true:

```
#define SERVO_SCALING true //dynamic scaling of servo motors
```

had no effect.

Modification 8: Inverting Button Inputs

In the meantime, I remembered another problem when running the Force Feedback Demo. Whenever the trigger gesture registered SteamVR took a screenshot. This meant that the system button was pressed down, as the screenshot shortcut in SteamVR is the trigger + system button. At the time I just disabled notifications and continued work on fixing the wiring. While debugging this I had a closer look at the serial monitor, which showed the finger, button and gesture values. This was one example line from the serial monitor when holding my hand in a neutral position:

```
A991B470C339D0E320F2047G2047P0HJKN
```

And this when making a fist:

```
A4004B4079C4070D2952E3869F2047G2047P4064HIJKLMN
```

In the `lucidgloves/firmware/Encoding.ino` the values from the serial monitor output are defined:

```
sprintf(stringToEncode, "A%dB%dC%dD%dE%dF%dG%dP%d%s%s%s%s%s%s%s%s\n",
  flexion[0], flexion[1], flexion[2], flexion[3], flexion[4], joyX, joyY,
  trigger, joyClick?"H":"", triggerButton?"I":"", aButton?"J":"",
  bButton?"K":"", grab?"L":"", pinch?"M":"", menu?"N":"", calib?"O":""
  );
```

The A-E (flexion[0-4]) values seemed to change values normally as I moved my fingers and because the joystick was disabled the F and G values were also normal. P, I, L, and M values for the trigger, triggerButton, grab and pinch were also correct, as making a fist should mean the trigger is at its maximum level and the trigger, grab and pinch gestures were activated. As the other buttons were pressed at all times, it meant that I had to invert them by changing the following defines:

```
#define INVERT_A true
#define INVERT_B true
#define INVERT_JOY true
#define INVERT_MENU true
#define INVERT_CALIB true
```

After fixing this I could delete the 355 screenshots made while testing the glove and prevent problems later on.

Modification 9: Changing the Software Branch

Going through the forum again there was one potential solution to the force feedback problem presented by Lucas, the author of the gloves, who suggested using the grab scaling branch instead [36]. As the grab scaling branch has prototype 5 support, which entered the beta stage after I had started working on this prototype [1], the configuration needed to be customized to support prototype 4. Below are the most notable definitions for prototype 4:

```
#define USING_SPLAY false
//whether or not your glove tracks splay. - tracks the side to side
//"wag" of fingers. Requires 5 more inputs.
```

[1]The change to prototype 5 was not feasible at this time, because was in an experimental beta phase and it works in a fundamentally different way: using hall-effect sensors instead of potentiometers, thus it required different 3D printed parts [39] and parts for the finger tracking mechanism [40].

```
#define USING_MULTIPLEXER false
//Whether or not you are using a multiplexer for inputs
...
#define USING_FORCE_FEEDBACK true
//Force feedback haptics allow you to feel the solid objects you hold
#define FFB_SCALE_GRAB true
//Whether or not the FFB data changes the threshold for the grab gesture
#define FLIP_FORCE_FEEDBACK false
#define SERVO_SCALING false //dynamic scaling of servo motors
...
//For potentiometers use MIXING_NONE.
#define FLEXION_MIXING MIXING_NONE
```

This branch introduced, as the name says, grab scaling. In the main branch triggering the grab gesture required the potentiometer values to be at 50 % or more of the calibrated value range, so because the main branch restricted the motion of the fingers way before reaching that value, no objects could be grabbed. The grab scaling branch had the possibility to scale the grab gesture threshold when grabbing an object instead of being a constant 50 % or more. Following was the new grab scaling method and the classic method function from `lucidgloves/firmware/lucidgloves-firmware/gesture.ino`:

```
bool grabGesture(int *flexion, int *savedHapticLimits){
  #if FFB_SCALE_GRAB //if we're dynamically scaling for haptics
  int total = 0;
  for (int i = 0; i <5; i++){
    int xServo = savedHapticLimits[i] * 0.9; //10% wiggle room to prevent
    //objects stuck in hand
    int xfscaled = flexion[i] * (1- xServo/ANALOG_MAX);
    total += xfscaled;
  }
  int avg = total/5;
  return avg <= ANALOG_MAX/2 ? 0:1;
  #else //classic grab gesture
  return (flexion[PINKY_IND] + flexion[RING_IND] + flexion[MIDDLE_IND] +
  //flexion[INDEX_IND]) / 4 <= ANALOG_MAX/2 ? 0:1;
  #endif
}
```

An alternative solution also pointed out in the forum was to decrease the force feedback multiplier. This had been solved in the Force Feedback Demo Github page in the less-ffb branch [41].

As I liked the idea and convenience of downloading the Demo from Steam and not building it myself, I tried the first solution before. Testing the Force Feedback Demo with the grab scaling branch introduced a new problem: grabbing an object and the feedback worked, but only for a split second. When the glove was moved near an object, the servos retracted as they should and right when I actually grabbed an object, they extended again. After getting no suggestions from the forum nor finding the bug myself, I decided to test the glove in Half-Life: Alyx integration mode.

Modification 10: Changing the Test Environment

Getting the Half-Life Alyx (HL:A) integration to work required installing the corresponding DLC for the OpenGloves application [42], setting the path in the DLC to
`C:\ProgramFiles(x86)\Steam\steamapps\common\Half-LifeAlyx`
and creating a file
`C:\ProgramFiles(x86)\Steam\steamapps\common\Half-LifeAlyx\game\hlvr\console.log`, as the DLC produced an error without it. Because the glove I made did not have any external buttons or a joystick, I changed the SteamVR controller bindings for HL:A to Dual Controllers (Movement On Weapon Hand), and the in-game setting for weapon hand to right to be able to move with the joystick on the right controller. Begin Half-Life: Alyx Integration had to be pressed in the DLC before starting HL:A.

I calibrated the hand pose as the real and virtual hand did not match up. There was a function in the Opengloves application: Pose auto-calibration. To calibrate SteamVR had to be started and by pressing the pose calibration button the controller in SteamVR froze. While the controller was in place I moved my hand in place of the controller to calibrate the mounted Meta Quest 2 controller offset.

The finger tracking and force feedback worked in HL:A, so I chose to continue with this rather than the Force Feedback Demo. Moreover because HL:A is based on the Source 2 engine, it gave me the possibility to make a demo map for it myself with the Valve Hammer Editor, which is included with every Source engine game [43], and publish it in the Steam Workshop. The main goal for the map was to have more varied objects than in the Opengloves Force Feedback Demo. The map could then be used to get first impressions from people using the glove and analyse what could be done better.

In conclusion to get the test environment and haptics working I needed to invert the button inputs (8), switch from the main LucidGloves branch to the grab-scaling one (9) and switch the test environment to HL:A (10) as the Force Feedback Demo did not work correctly.

2.8 Summary

The main variations from the video guide versus my glove were my choice of the glove (1), sewing of the velcro straps (3), the mounting place and wiring type of the development board (5), the software build (9) and the testing environment being a custom HL:A map instead of the Force Feedback Demo (10). The final glove is shown on Figure 2.15.

Figure 2.15: Final Glove Prototype With and Without the Controller

3 Analysis

I based the analysis of the haptic glove on my own thoughts and the feedback from 4 test subjects, who experienced the haptic glove for the first time. The first round of feedback included 2 people who tried the glove in HL:A and compared it to the Meta Quest 2 hand tracking in Hand Physics Lab [44]. I chose to compare the haptic glove to hand tracking in Hand Physics Lab because it provided a similar experience to the HL:A map I made, which was interacting with different objects using hands. The story mode levels of HL:A were too linear and did not include many objects in one place, so I made and published my own map which I made with the Valve Hammer Editor to the Steam Workshop. A link to the final iteration of the demo map can be found in the appendices section 6.4.

3.1 First Round of Feedback

For the first test I gave the test subjects free will of what to interact with in the HL:A map. Both of the test subjects were unable to reliably grab objects at first, but as they got more comfortable with the glove they said that it helped to grab objects more aggressively and with intent. Some objects were difficult to grab until the end. For example, a half-assembled gun grip was easy to grab with the controller, but with the glove, they found it hard to predict how to gun would be positioned in the hand. Every object had a predefined hand pose for when the object would stay grabbed so when they tried to grab something but that pose was not expected by the object it broke immersion as they were unable to reliably grab the object. If they could trigger the grab gesture the object would be held awkwardly and not in a way they intended. Some objects handled grabbing better, for example grabbing a plate provided accurate feedback with servos and had a similar hand pose to the real hand.

After about 10 minutes the glove became uncomfortable for both of the test subjects due to their hand becoming too hot and all the weight of the glove being at the top. The weight distribution of the glove bothered me too. Without the controller, the glove felt more like a normal glove but mounting the controller put a lot of weight on top of the glove, which made fast movements uncomfortable and after a while the glove became bothersome to use.

Another immersion-breaking interaction was pushing the doors of cabinets and car doors while not actually grabbing them. This provided no feedback to the palm of the hand but could not be solved with the current build of the glove.

After the HL:A test I let them try to interact with objects in the Hand Physics Lab and compare the feel and immersion of the two experiences. Hand tracking on the Meta Quest 2 is a native feature in some games including Hand Physics Lab meaning it does not emulate a controller.

Grabbing objects in the Hand Physics Lab similar to the haptic glove required a little bit of practice. Although comfortable compared to the glove the overall feedback was that hand tracking was more weightless and flimsy while the glove gave a better feel of interaction due to the weight of the glove and the tracking accuracy of the mounted controller during fast movements. While the haptic feedback provided with servos was interesting, the test subjects agreed that most of the immersion compared to hand tracking came from the tactile feel of a glove being on the hand, having some weight and more precise hand tracking when doing fast movements.

In conclusion of the first test, the haptic glove had a better feel for grabbing objects than hand tracking although some objects were difficult to grab reliably. The glove became uncomfortable after about 10 minutes because the hand got too hot and the weight distribution was all on top of the glove. Feedback from the servos was interesting for the test subjects but not the main reason for increased immersion, which was actually the tactile feel, weight and accurate tracking of the glove.

3.2 Second Round of Feedback

The two rounds of feedback were three weeks apart. The second round of feedback included only the HL:A map, which I had improved by adding more interactive objects like whiteboards, markers, erasers, bottles, and removing some of the problematic objects like the gun grip. The reasoning for removing some objects was to give the users a better experience with the glove working as intended. I did not think it was necessary to compare the haptic glove with hand tracking at this point, because gathering more feedback on the glove was the main priority.

The main problem we encountered at this stage of testing was the force feedback. As the glove was made considering my hand size, I calibrated the length of the strings and the positioning of the servo hands for my needs. When grabbing objects they both expressed that they felt very little or no resistance, although the servos worked as expected. The glove fit the test subject's hands but not as well as mine and the result was that the feedback restricted their finger motion when their hand was already close to a fist. The objects they grabbed were larger than the feedback indicated, so this broke immersion. The main factors for this were that the first test subject's fingers were shorter than mine and the seconds did not fit fully into the end caps. With shorter fingers making a fist did not retract the strings as much as with my hand because having shorter fingers meant the distance moved from an open hand to a fist was smaller. The problem with fingers not fitting the end caps was that the only point of resistance was pulling on the end caps, but the test subjects did not feel it. On Figure 3.1 the servo hands restrict the retraction of the strings the same amount, but because my fingers are not fully in the end caps, the hand can be closed more.

Figure 3.1: Problem When Fingers Were Not in End Caps

With the limited feedback they experienced they also said that even when the servos limited the movement of fingers, they could still be moved too much. As the retraction of the string was stopped by the servo, the finger could still be moved sideways or by changing the amount of flex in the finger. As the feedback was only on the fingertip but fingers have three joints, the same amount of string retraction could be achieved by a different finger position. This can be seen on Figure 3.2, where the servo hands are at the same position, but the finger position is different.

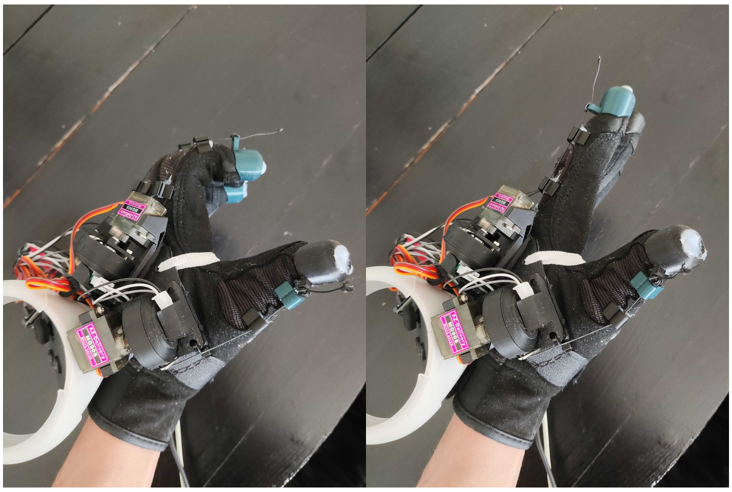

Figure 3.2: Lack of Precision in Force Feedback

Having more experience and understanding of how the glove works, I knew what to expect and how to interact with objects so the feedback stops my fingers close to where the in-game fingers were, so I did not notice this as much. When the feedback did work however, the test subjects tried to flex their fingers more than me and this introduced a small problem.

During testing one retractable spool cover came loose. I did not experience any kind of breaking during testing myself, but that was probably because I was using the glove more gently. The problem was that the index finger's slide holder was the other way around and when the feedback activated, the string pulled on the spool cover and it came off. A quick fix was to rotate the cover so the tension did not pull on the cover but the metal loop instead as seen on Figure 3.3.

Overall the second round of feedback introduced new problems in addition to the glove becoming uncomfortable to use after a while as was seen in the first round of feedback. The glove size and calibration suited my hand but variations in finger size or thickness proved to break immersion as the feedback did not work as expected.

While the glove provided a more immersive experience compared to hand tracking in some ways, it behaved worse in others. Of course, having no glove on when using hand tracking is

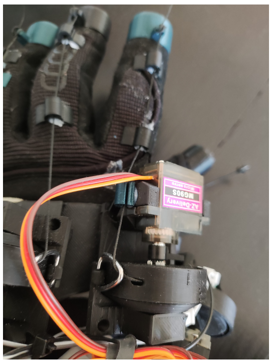

Figure 3.3: Rotated Cover to Put Tension on Metal Loop

more comfortable, but having the glove on could still be a lot more enjoyable experience. In addition to the bad weight balance and the glove being too warm to use indoors, there was an issue with the in-game hand models and for two test subjects the way the glove provided feedback. Some of these problems could be solved with changes in hardware and some in software.

3.3 Potential solutions

3.3.1 Hardware

I will first list the solutions to problems with my build of the glove. The Quest 2 controller put a lot of weight on the top of the glove, which was unnatural and made using the glove uncomfortable. An alternative would be to use the Vive Tracker 3, which weighs 75 grams [45] compared to the Quest 2 controllers 147 grams with the battery included [46]. The Vive Tracker 3 also has a lower profile while the Quest 2s controller has a tracking ring which sits quite high up on the glove. An extra thing to point out is that the Vive Tracker would not rotate inside the controller mount as the Quest 2 did. Transporting the glove sometimes rotated the controller and to have an accurate calibrated offset sometimes the controller had to be rotated back to the original position. This could also be solved by making a new Quest 2 controller mount where it could not rotate.

Another problem was that the user's hand became too hot inside the glove. As this was an outdoor glove there may be a less thick alternative which still provides enough rigidity. As an alternative, the glove could be replaced with a glove skeleton. LucidGloves prototype 5 would be the best option because it is a glove skeleton [37]. In addition, the LucidGloves wiki has a community design list [47], where a GitHub user Valsvirtuals has made designs like the ZipLine [48] and ProtoGlove [49], which are more similar to an exoskeleton and could be modified to remove the need for a glove. Prototype 5, Zipline and ProtoGlove designs can be seen on Figure 3.4

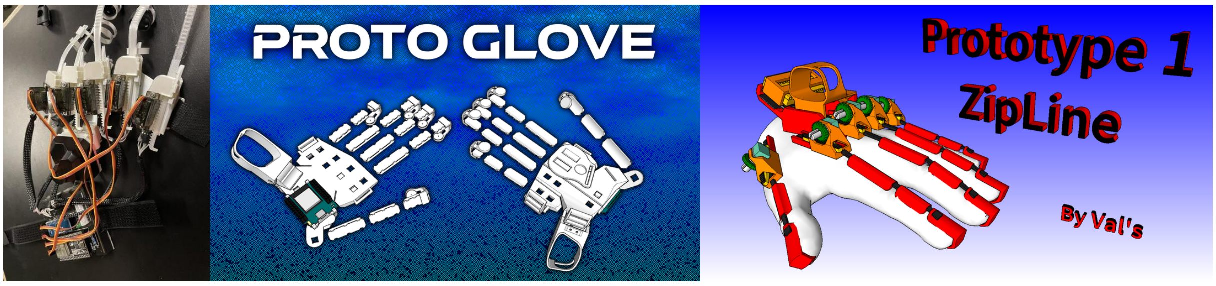

Figure 3.4: From Left to Right: Prototype 5, ProtoGlove and ZipLine Designs

Although I tried to find test subjects whose hand size was close to mine, two of them still experienced issues. The glove was made for my hand specifically, so variations in the user's hand size would mean a need for a complete change of the glove size, re-calibrating the servo hands, and/or printing the end caps in a different size. Having the end caps made out of a flexible material like silicone would help make it suitable for different hand sizes. Prototype 5 uses silicone guitar fingertip guards [40], so switching the end caps for these and adding a way to tie the strings to the fingertip guards could solve this problem. Silicone end caps would also make moving fingers more comfortable, as the rigid 3D-printed end caps collide with each other and restrict some movements.

The way force feedback worked on this glove did provide feedback, but fingers could still move too much according to two test subjects. This could be solved with a different design of the glove, for example, prototype 5, which is in beta, supports splay tracking, and implementing feedback on the splay motion would be possible. Having feedback in the palm of the hand would require additional parts, for example, a diaphragm pump and a bladder.

Overall some of the problems with the build of the glove like the weight distribution and the user's hand becoming too hot are possible to solve without needing a completely different design. Additional features like a more precise feedback system and palm feedback would require a different prototype design.

3.3.2 Software

Regarding the problem with smaller hands experiencing feedback with their fingers almost closed, the OpenGloves driver could be improved by adding a servo calibration feature, where the user could change the angle of rotation on servos when grabbing objects.

Another problem with the software was that some objects could not be grabbed reliably. This was an issue with HL:A and the object's hand position when grabbed. As this only impacted some of the objects, they could just be removed as adding new objects was possible to increase the variety.

A feature in the Force Feedback Demo, which I tried again after all the tests and analysis and finally got working, was that different objects had different feedback values. So smaller objects would restrict finger movement later than bigger objects. This feature was not implemented in HL:A. Unfortunately, I could feel the feature working in the Force Feedback Demo when grabbing only one type of object, the grenade which exploded into flowers. The main objects in Force Feedback Demo were still spheres and cubes and grabbing them did not feel as immersive as more realistic objects in HL:A. The main reason for this was that grabbing them did not activate a premade hand model as it did in HL:A and fingers could occlude into the objects. Also, in the HLA map I made, most of the objects still provided quite accurate feedback for me and I did not really notice the lack of this feature, because at the same string retraction length, the hand model could be usually closely matched with my real hand and I would feel resistance. This however was after many hours of testing and already knowing what the hand models look like for most objects on the map. So the lack of precision seen on Figure 3.2 was a positive in a sense. Still one of the bigger improvements to the existing prototype would be to improve the Force Feedback Demo by adding more differently sized objects or implementing the variable force feedback feature to the HL:A Force Feedback Integration DLC.

An alternative would be to get the glove working in Boneworks [50] with the OpenGloves mod [51], which also supports this feature.

4 Summary

The prototype I built was based on the LucidGloves prototype 4, which is an open-source project. The LucidGloves project included all the necessary guides and resources to build a finished glove, but I experienced issues following the guides. I accomplished the aim of this thesis, which was to document the process of building a DIY haptic glove. An extensive overview of the whole process makes building the glove more repeatable. I also made a set of 10 useful modifications and gave a critical review of the glove.

Overall the description of the process of building the glove and an analysis of it is going to help students and researchers improve this glove or make a better prototype taking into account the feedback and issues I faced. This prototype gives a glimpse into the future of open-source haptic gloves and what true haptic feedback in ideal situations would feel like.

5 Acknowledgements

I would like to express my gratitude to my supervisor, Ulrich Norbisrath, Professor of Distributed Systems at the University of Tartu, for guiding and helping me keep a positive outlook even when things did not go according to plan. I would also like to thank Tanel Tokman, Karl Aleksander Kaplinski, Ingvar Drikkit and Urmo Schults-Kopp for testing the glove and providing valuable feedback. Finally, I would like to thank the whole community behind the LucidGloves project for making a budget DIY haptic glove possible.

Bibliography

[1] The Computer Graphics and Virtual Reality Study Lab @ University of Tartu - About https://cgvr.cs.ut.ee/about/ 05.04.2023, 14:41 (UTC)

[2] J. K. O'Regan, A. Noë, "A sensorimotor account of vision and visual consciousness," *Behavioral and Brain Sciences 24, no. 5*, 2001, pp. 939–973, DOI:10.1017/S0140525X01000115.

[3] M. Slater, "Place Illusion and Plausibility Can Lead to Realistic Behaviour in Immersive Virtual Environments," *Philosophical Transactions: Biological Sciences 364, no. 1535*, 2009, pp. 3549–3557. DOI:10.1098/rstb.2009.0138.

[4] P. Xueni, A. F. de C. Antonia, "Why and how to use virtual reality to study human social interaction: The challenges of exploring a new research landscape," *British Journal of Psychology 109, no. 3*, 2018, pp. 395-417, DOI:10.1111/bjop.12290

[5] B. Gavin, "Hand Tracking for Immersive Virtual Reality: Opportunities and Challenges," *Frontiers in Virtual Reality 2*, 2021, DOI:10.3389/frvir.2021.728461.

[6] A. Masurovsky, P. Chojecki, D. Runde, M. Lafci, D. Przewozny, M. Gaebler, "Controller-Free Hand Tracking for Grab-and-Place Tasks in Immersive Virtual Reality: Design Elements and Their Empirical Study," *Multimodal Technologies and Interaction 4, no. 91*, 2020, DOI:10.3390/mti4040091.

[7] A. Heath, The Verge - This is Meta's AR / VR hardware roadmap for the next four years https://www.theverge.com/2023/2/28/23619730/meta-vr-oculus-ar-glasses-smartwatch-plans 01.03.2023, 2:33 (UTC).

[8] M. Slater, "Immersion and the illusion of presence in virtual reality," *British journal of psychology 109, no. 3*, 2018, pp. 431-433. DOI:10.1111/bjop.12305.

[9] S. Carolin, B. Michael, K. Ernst, and S. Johannes, “Triggermuscle: Exploring Weight Perception for Virtual Reality Through Adaptive Trigger Resistance in a Haptic VR Controller,” *Frontiers in Virtual Reality 2*, 2022, DOI:10.3389/frvir.2021.754511.

[10] SenseGlove - About Us https://www.senseglove.com/about-us/ 20.05.2023, 12:59 (UTC).

[11] SenseGlove - SenseGlove Nova https://www.senseglove.com/product/nova/ 20.05.2023, 13:01 (UTC).

[12] M. Caeiro-Rodríguez, I. Otero-González, F. A. Mikic-Fonte, and M. Llamas-Nistal, “A Systematic Review of Commercial Smart Gloves: Current Status and Applications,” *Sensors 21, no. 8*, 2021, pp. 2667. DOI:10.3390/s21082667.

[13] M. W. Uddin, X. Zhang and D. Wang, “A Pneumatic-Driven Haptic Glove with Force and Tactile Feedback,” *2016 International Conference on Virtual Reality and Visualization (ICVRV)*, Hangzhou, China, 2016, pp. 304-311, DOI:10.1109/ICVRV.2016.57.

[14] A. C. Abad, M. Ormazabal, D. Reid and A. Ranasinghe, “An Untethered Multimodal Haptic Hand Wearable,” *2021 IEEE Sensors*, Sydney, Australia, 2021, pp. 1-4, DOI:10.1109/SENSORS47087.2021.9639526.

[15] Hackaday - LucidGloves: VR Haptic Gloves on a budget https://hackaday.io/project/178243-lucidgloves-vr-haptic-gloves-on-a-budget 15.03.2023, 11:05 (UTC).

[16] Youtube - How to build cheap VR Haptic Gloves to FEEL VR. https://www.youtube.com/watch?v=2yF-SJcg3zQ&t=90s 10.05.2023, 13:02 (UTC).

[17] Github - LucidGloves Prototype 4 Parts List https://github.com/LucidVR/lucidgloves/wiki/Prototype-4-Parts-List 15.03.2023, 13:21 (UTC).

[18] Github - LucidGloves Firmware Setup and Customization Tutorial https://github.com/LucidVR/lucidgloves/wiki/Firmware-Setup-and-Customization-Tutorial 12.04.2023, 12:11 (UTC).

[19] Github - LucidGloves https://github.com/LucidVR/lucidgloves 12.04.2023, 12:21 (UTC).

[20] Steam - OpenGloves
https://store.steampowered.com/app/1574050/OpenGloves 12.04.2023, 12:47 (UTC).

[21] Steam - OpenGloves - Force Feedback Demo
https://store.steampowered.com/app/1654950/OpenGloves__Force_Feedback_Demo/
12.04.2023, 12:47 (UTC).

[22] Github - LucidGloves Prototype 4 Printing Guide
https://github.com/LucidVR/lucidgloves/wiki/Prototype-4-Printing-Guide 13.04.2023,
12:44 (UTC).

[23] Github - LucidGloves Prototype Hardware
https://github.com/LucidVR/lucidgloves/tree/main/hardware 15.04.2023, 19:04 (UTC).

[24] Github - LucidGloves Prototype 4 Hardware - Tensioner_WLPot.STL
https://github.com/LucidVR/lucidgloves/blob/main/hardware/Prototype4/Tensioner_WLPot.STL
17.05.2023, 11:27 (UTC).

[25] Github - LucidGloves Prototype 4 Hardware - SpoolCover_open.STL
https://github.com/LucidVR/lucidgloves/blob/main/hardware/Prototype4/SpoolCover_open.STL
17.05.2023, 12:09 (UTC).

[26] Github - LucidGloves Prototype 4.1 Hardware - HapticSpool_Proto4.1.STL
https://github.com/LucidVR/lucidgloves/blob/main/hardware/Prototype4.1/HapticSpool_Proto4.1.STL
17.05.2023, 12:10 (UTC).

[27] Github - LucidGloves Prototype 4 Hardware - Tensioner_GreenPot.STL
https://github.com/LucidVR/lucidgloves/blob/main/hardware/Prototype4/Tensioner_GreenPot.STL
17.05.2023, 12:11 (UTC).

[28] Github - LucidGloves Prototype 4.1 Hardware - GreenPotSlideHolder.STL
https://github.com/LucidVR/lucidgloves/blob/main/hardware/Prototype4/GreenPotSlideHolder.STL
17.05.2023, 12:18 (UTC).

[29] Github - LucidGloves Prototype 4 Hardware - RigidMount_leftHand.STL
https://github.com/LucidVR/lucidgloves/blob/main/hardware/Prototype4/RigidMount_leftHand.STL
17.05.2023, 12:19 (UTC).

[30] Github - LucidGloves Prototype 4 Hardware - Prot4_EndCap.STL
https://github.com/LucidVR/lucidgloves/blob/main/hardware/Prototype4/Prot4_EndCap.STL
17.05.2023, 12:22 (UTC).

[31] Github - LucidGloves Prototype 4.1 Hardware - GuideNode_upgraded.STL
https://github.com/LucidVR/lucidgloves/blob/main/hardware/Prototype4.1/GuideNode_upgraded.STL
17.05.2023, 12:23 (UTC).

[32] Github - LucidGloves Prototype 4 Hardware - Quest2_MountSlider.STL
https://github.com/LucidVR/lucidgloves/blob/main/hardware/Prototype4/Quest2_MountSlider.STL
17.05.2023, 12:26 (UTC).

[33] Wikipedia - Daisy chain (electrical engineering)
https://en.wikipedia.org/wiki/Daisy_chain_(electrical_engineering) 10.05.2023, 13:48 (UTC).

[34] Espressif - Esp32-Wroom-32 Datasheet
https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf 04.05.2023, 11:18 (UTC).

[35] G. Burdea, "Haptic Feedback for Virtual Reality," *Proceedings of International Workshop on Virtual prototyping*, Laval, France, 1999, pp. 87-96.

[36] Github - LucidGloves (grabscaling branch)
https://github.com/LucidVR/lucidgloves/tree/grabscaling 03.05.2023, 10:27 (UTC).

[37] Github - Prototype 5 Beta Assembly Guide
https://github.com/LucidVR/lucidgloves/wiki/Prototype-5-BETA-Assembly-Guide
18.05.2023, 10:29 (UTC).

[38] Github - LucidGloves Community Design List
https://github.com/LucidVR/lucidgloves/wiki/Community-Design-List 18.05.2023, 10:29 (UTC).

[39] Github - LucidGloves Prototype 5 Beta Printing Guide
https://github.com/LucidVR/lucidgloves/wiki/Prototype-5-Beta-Printing-Guide
18.05.2023, 10:24 (UTC).

[40] Github - LucidGloves Prototype 5 Beta Parts List https://github.com/LucidVR/lucidgloves/wiki/Prototype-5-Beta-Parts-List 18.05.2023, 10:24 (UTC).

[41] Github - OpenGloves Force Feedback Unity Demo (less-ffb branch) https://github.com/LucidVR/opengloves-force-feedback-unity-demo/tree/less-ffb 03.05.2023, 10:49 (UTC).

[42] Steam - OpenGloves - Half-Life: Alyx Force Feedback Integration https://store.steampowered.com/app/1658630/OpenGloves__HalfLife_Alyx_Force_Feedback_Integration/ 03.05.2023, 13:24 (UTC).

[43] Valve Developer Community - Valve Hammer Editor https://developer.valvesoftware.com/wiki/Valve_Hammer_Editor 15.05.2023, 13:30 (UTC).

[44] Oculus - Hand Physics Lab https://www.oculus.com/experiences/quest/3392175350802835/ 11.05.2023, 10:55 (UTC).

[45] Vive - Vive Tracker https://www.vive.com/us/accessory/tracker3/ 14.05.2023, 17:54 (UTC).

[46] GSMArena - Oculus Quest 2 review https://www.gsmarena.com/oculus_quest_2_review-news-46255.php 14.05.2023, 17:56 (UTC).

[47] GitHub - LucidGloves Community Design List https://github.com/LucidVR/lucidgloves/wiki/Community-Design-List 18.05.2023, 8:17 (UTC).

[48] GitHub - Valsvirtuals ZipLine https://github.com/Valsvirtuals/ZipLine 18.05.2023, 8:19 (UTC).

[49] GitHub - Valsvirtuals ProtoGlove https://github.com/Valsvirtuals/ProtoGlove 18.05.2023, 8:20 (UTC).

[50] Steam - Boneworks https://store.steampowered.com/app/823500/BONEWORKS/ 20.05.2023, 13:39 (UTC).

[51] GitHub - LucidVR OpenGloves Boneworks Mod https://github.com/LucidVR/opengloves-boneworks-mod 20.05.2023, 13:39 (UTC).

6 Appendices

6.1 3D Printed Parts List

Table 6.1: 3D Printing Parts List for One Glove

Part name	Quantity
HapticSpool_Proto4.1	5
SpoolCover_open	5
Tensioner_GreenPot	5
GuideNode_upgraded	10-13
GreenPotSlideHolder	5
RigidMount_leftHand	1
Quest2_MountSlider	1
Prot4_EndCap	5

6.2 Parts List

Table 6.2: Parts List for One Glove

Part name	Quantity
ESP32 Dev Board	1
Potentiometers RK097N-3-10K	5
MG90S servos	5
Retractable badge reels	5
TAMREX PRO6000 tactical glove	1
Velcro strips 20 cm hook	1
Velcro strips 20 cm loop	1
Power bank if using Bluetooth	2
Elastic waistband 30 cm	1
Foam rubber 20x20 cm	1

6.3 Videos

Potentiometer Test With OpenGloves Force Feedback Demo

LucidGloves Force Feedback Test in HL: Alyx

LucidGloves HL: Alyx Test Map Overview With Left Glove

6.4 Half-Life: Alyx Map

LucidGloves Force Feedback Demo Map

Non-exclusive licence to reproduce thesis and make thesis public

I, Kasper Kaljuste

1. grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright, my thesis

   "Virtual Reality Gloves Based on the LucidGloves Project"

   supervised by Ulrich Norbisrath

2. grant the University of Tartu a permit to make the thesis specified in point 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 4.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. am aware of the fact that the author retains the rights specified in points 1 and 2.

4. confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Kasper Kaljuste

20.05.2023