



Department of Precision and Microsystems Engineering

# Haptic feedback in virtual reality rehabilitation for Parkinson's Disease

Myla van Wegen

Report no : 2022.026  
Professors : J.L. Herder, T. Vanbellingen  
Specialisation : MSD  
Type of report : MSc Thesis  
Date : 1st of July 2022

# Thesis Report

## Haptic feedback in virtual reality rehabilitation for Parkinson's Disease

by

Myla van Wegen

4473078

Supervisors: J. Herder, T. Vanbellingen  
Project Duration: September, 2021 - July, 2022  
Faculty: Faculty of Mechanical, Maritime and Materials Engineering, Delft and  
ARTORG Centre for Biomedical Engineering Research, Bern

# Preface

*A little over a year ago I first got in touch with this project. Not completely by accident it was located at the University of Bern, in Switzerland. It combined all my interest; a challenging design project which involved hands-on work and at the same time creating something that would help others in a new and inspiring environment. It took me sometime to convince my friends that I was actually going to be working on a research project and not the projects one works on in the mountains..*

*It was a year full of new experiences. From never having worn a VR headset, to designing a VR experiment. From living just five minutes from the ocean to living 500 m above sealevel with mountain view. From never having done a nine month long project, to leading, planning and executing a nine month long project. From almost never eating cheese fondue to eating cheese fondue at least once a week. From not having my own desk at uni, to having two great places in Bern and Zurich to work on my project. From speaking a bit of German to still speaking a bit of German.*

*During the year I was so grateful for all the great people around me. First of all, thanks to my dad for helping me brainstorm and discuss the project, and, of course, getting me in touch with Tim. Thanks a lot to Tim who took the chance on a foreigner to take on one of his projects and supervising me with great energy along the way. Big thanks to Just, for finding the time to supervise, discuss and brainstorm with full engagement. Of course, Rolf, for having the cool VR glove startup and providing me with great tips and tricks. And all the people that participated in my experiments, thanks a lot.*

*Then, not to forget, many thanks to the great people at ARTORG who were lovely enough to speak English, organise things including food and help me to explore Bern. To my amazing roommates in The Hague who would encourage me to work hard so that we could go camping and bouldering together. To my mom for cycling all the way to Bern to visit and my brother and sister for hugging me when I got home. For all the peers and my fellow board members in Delft spending so much time drinking coffee with me. And not to forget, Lukas who joined the adventure not knowing he was the one that had to endure all my down times. Who still wanted go running, skiing and climbing with me even though I was too stressed to be nice. Last but not least, thanks Switzerland, for having the best mountains ever in which I could empty my mind and gain new energy, ideas and adventures.*

*Looking back, it feels crazy how much can change after just one opportunity. The thesis project has brought me so much new knowledge and memories. Not only did I learn to manage a huge project by myself, I learned that it is possible do to both a research project and mountain projects at the same time. Most of all, I learned that you never really do it alone. I hope you enjoy reading my work.*

*All the best,*

*Myla van Wegen  
Delft, July 2022*

# Summary

*It is hypothesised that providing Haptic Feedback (HFB) during VR dexterity rehabilitation can further enhance the performance of PD patients in a Virtual Environment (VE) and eventually during ADL, due to the extra immersion that might occur due to the feedback. The overarching goal of the project is to determine the usability of HFB in VR rehabilitation for PD patients to evaluate whether it can enhance their dexterity and consequently would improve their ability to perform ADL. First, an extensive literature search is done, resulting in a paper called: Virtual interaction; an overview of wearable haptic technologies and their performance in virtual object exploration. Then, a prototype Haptic Device (HD) was designed and build where Vibrotactile Feedback (VTF) was implemented as a means of haptic feedback. With the prototype a usability study was performed both on a technical level as well as on a user level with healthy subjects and patients with Parkinson's Disease (PD). This was reported by means of a paper called: . A novel virtual reality glove system with integrated vibro-tactile feedback for Parkinson's Disease: A usability study.. The results of this study indicated that indeed VTF as a means of haptic feedback during rehabilitation could enhance dexterity in Activities of Daily Living (ADL) in PD. However, there is still improvement possible in terms of the design of the HD and further studies will be needed to evaluate its value as a rehabilitation tool to improve QoL in PD. This report acts as a complementary document to the paper on the conducted study with the designed HD and elaborates on the design steps, prototype construction and data analysis.*

# Contents

<b>Preface</b>	<b>i</b>
<b>Summary</b>	<b>ii</b>
<b>1 Introduction</b>	<b>2</b>
<b>2 Theory</b>	<b>4</b>
2.1 Parkinson's Disease	4
2.1.1 Symptoms	4
2.1.2 Severity Diagnosis	4
2.1.3 Treatment	5
2.1.4 Rehabilitation	5
2.2 Virtual Reality	5
2.3 Haptics	5
2.3.1 Haptic perception	6
2.3.2 Interaction tasks	7
2.3.3 Object Exploration	7
2.3.4 Haptic feedback technologies	8
<b>3 Design</b>	<b>9</b>
3.1 Design Requirements	9
3.1.1 Technological requirements	9
3.1.2 Practical requirements	9
3.2 Design decisions	9
3.2.1 Type of haptic feedback	9
3.2.2 Type of actuator	10
3.3 Preliminary Tests	11
3.3.1 Execution	13
3.3.2 Results	13
<b>4 Prototype</b>	<b>14</b>
4.1 Hardware	14
4.1.1 VRFRee Glove	14
4.1.2 Head Mounted Display (HMD)	14
4.1.3 Electronics	14
4.1.4 Casing	16
4.2 Software	18
4.3 Short HD Manual	18
<b>5 Experiments</b>	<b>20</b>
5.1 Technical tests	20
5.1.1 Latency	20
5.1.2 Intensity	20
5.1.3 Compatibility	20
5.1.4 Scripts	21
5.2 User tests	21
5.2.1 Participants	21
5.2.2 Virtual Reality (VR) test	21
5.2.3 Questionnaire	22
5.2.4 Data Analysis	22
5.3 Unforeseen Events	24

---

<b>6 Conclusion and future recommendations</b>	<b>25</b>
<b>References</b>	<b>26</b>
<b>A Scripts</b>	<b>28</b>
<b>B Questionnaires</b>	<b>49</b>
<b>C 3D Models</b>	<b>54</b>

# Acronyms

**ADL** Activities of Daily Living  
**DBS** Deep Brain Stimulation  
**ERM** Eccentric Rotating Mass  
**HD** Haptic Device  
**HF** Haptic Factor  
**HFB** Haptic Feedback  
**HMD** Head Mounted Display  
**HX** Haptic Experience  
**IMU** Inertial Measurement Unit  
**LKA** Limb Kinetic Apraxia  
**LRA** Linear Resonant Actuators  
**NE** norepinephrine  
**PD** Parkinson's Disease  
**QoL** Quality of Life  
**SUS** System Usability Scale  
**UPDRS** Unified Parkinsons Disease Rating Scale  
**USEQ** User Satisfaction Evaluation Questionnaire  
**VE** Virtual Environment  
**VR** Virtual Reality  
**VTF** Vibrotactile Feedback

# Introduction

PD is a very common neuro-degenerative disease, which affects many people all over the world. As there is no cure available yet, treatment of the disease is aimed at reducing the symptoms in order to increase the performance of ADL and consequently enhance the Quality of Life (QoL) of PD patients. Rehabilitation is used as a means to improve overall physical functioning to increase performance of ADL. This rehabilitation can be focused on improving dexterity, as dextrous deficits are a common symptom in PD and very important during the execution of ADL. In the recent years Virtual Reality (VR) has become a great solution for providing more engaging and patient-tailored programs. It is hypothesised that providing Haptic Feedback (HFB) during VR dexterity rehabilitation can further enhance the performance of PD patients in a Virtual Environment (VE) and eventually during ADL, due to the extra immersion that might occur due to the feedback.

The overarching goal of the project is therefore to determine the usability of HFB in VR rehabilitation for PD patients to evaluate whether it can enhance their dexterity and consequently would improve their ability to perform ADL. During the project, the main goal has been converged to the following: *Implementation of HFB in a wearable and usable manner to perform a pilot study on haptic VR dexterity rehabilitation for patients with PD*, from which the following sub-questions were derived: "What type and technology of HFB are adequate and suited for enhancing PD dexterity rehabilitation?" and "How can we examine the enhancement in dexterity performance, compliance of the patients and usability of the system?". This research objective is carried out through an extensive literature review on virtual object interaction and through the design of a novel HD which is tested both on technical requirements and by users.

This report acts as a complementary document to both the paper in which the literature study is conducted (*Virtual interaction; an overview of wearable haptic technologies and their performance in virtual object exploration*) and the paper on the conducted usability study with the designed HD (*A novel virtual reality glove system with integrated vibro-tactile feedback for Parkinson's Disease: A usability study*). The paper contains all necessary information on the design of the system with HD and the experimental method, as well as the results of the experiments and an extensive discussion on the results and the conclusion that is drawn. The aim of this report is to describe the project process, background and decisions in more detail, where that was not possible in the paper. This is done by first providing relevant theory on PD and rehabilitation, haptics and haptic technologies in chapter 2. In chapter 3 the design requirements are discussed, together with the design decisions that have been made in terms of type of feedback and actuators. Some preliminary testing has been done with patients which is also discussed in chapter 3. After the design of the HD, the prototype is build, which is described in detail in chapter 4 and includes the implemented hardware and software as well as a manual to use the device. In chapter 5 the design, execution and analysis of the technical and user tests is discussed. Finally a conclusion on the overall project is drawn, together with recommendations for future studies.

In addition to this report, a digital folder is created, which contains all the digital information concerning the project, such as scripts and obtained user data. In this report, this folder will be referred when



applicable.

# 2

## Theory

### 2.1. Parkinson's Disease

PD is a common neuro-degenerative disease, where dopaminergic neuronal loss in the substantia nigra leads to a dopamine deficiency. Next to that, selective degeneration of neurons of the locus coeruleus and sympathetic ganglia causes a deficiency in norepinephrine (NE) [24] [12]. These deficiencies account for the development of several symptoms, which affect the QoL of PD patients.

#### 2.1.1. Symptoms

PD patients can develop a wide variety of symptoms during the progression of the disease. In the early stages, the most evident symptoms are movement-related, such as tremor, stiffness, problems with walking and gait and bradykinesia. Also, it has shown that PD affects the reach-to-grasp movement [14] and dexterity due to impairments in individual finger force and torque control [8]. Cognitive and behavioural problems may arise later on, together with sleep and emotional problems, impaired balance and coordination, depression, speech difficulties, and fatigue. Also, sensory disturbances are a common symptom in PD and altered tactile-motor integration and proprioceptive-integration, that is the temporal coupling between haptic information and motor output, could contribute to the motor symptoms that induce movement impairments [6].

These symptoms can cause inability of the patient to perform the essential ADL both in the beginning as well as more advanced stages of the disease [10]. Especially, manual dexterity is essential in the physical functioning during ADL, such as getting dressed, cooking and opening doors. Together with treatment induced complications, these symptoms can consequently affect the QoL of a patient [21], which is further explained in subsection 2.1.2.

#### 2.1.2. Severity Diagnosis

To determine the stage and severity of Parkinson's disease of a patient, multiple rating scales have been developed. Each scale has its own particular method and focus area. The most widely used rating scale that contains both an impairment, a disability section and a mental and behavioural section is the Unified Parkinsons Disease Rating Scale (UPDRS) [25]. This is a rating instrument used to map the motor and behavioural aspects of the disease. This scale focuses on the patient experiences from the past week and contains a total of 42 items over four sections; I mental functioning, behaviour and mood, II ADL, III motor functioning, IV complications of therapy [28]. A scale that focuses on the QoL as perceived by the patient is the PD Questionnaire39, which evaluates 39 items over eighth sections; mobility, ADL, emotional well-being, stigma of the disease, back in the next, cognitive, communication, bodily discomfort [21]. The QoL is an important measure of the severity of the disease as it describes how the patient experiences the disease personally. It is a multi-dimensional term which contains physical, mental and social aspects to describe the patients satisfaction of life [21]. Schrag et al. conclude that the QoL in idiopathic Parkinson's Disease is primarily determined by depression, disability, postural instability, and cognitive impairment [27]. The focus of the treatment and rating of the disease should therefor be mainly in these areas.

### 2.1.3. Treatment

Currently, a cure for PD does not exist. However, there exist a wide variety of treatment strategies, including pharmacologic approaches and Deep Brain Stimulation (DBS), as well as non-pharmacologic approaches, such as rehabilitation therapies. Medicinal dopamine replacement has proven to improve bradykinesia [20] by enhancing motor power in movements. However, it is not able to reconstitute the loss in coordinated motor output [26] during, for example, reach-and-grasping tasks or balance and gait and manual dexterity [14]. DBS interventions, on the other hand, have shown to positively affect motor symptoms both in gait performance, standing postural control [5] and upper limb motor performance, including dexterity [20]. This suggests that impairment in dexterity in PD might be related to Limb Kinetic Apraxia (LKA), which is the decrease in fine motor skills not explained by basic motor impairments. Unfortunately, both dopamine treatment and DBS come with negative side effects. Rehabilitation, including exercises interventions, however, has proven to significantly improve overall physical functioning and balance without the negative effects of medicinal and DBS treatment and has shown to be an essential component in the treatment strategy of Parkinson patients [2]. This stresses the importance of neuro-rehabilitation to address dextrous deficits [14].

Because PD is a neuro-degenerative disorder, it has become clear that different approaches and content of the interventions are necessary at different stages of the disease. This means that a rehabilitation program should not be standardised, but tailored to the specific patient and targeted at their personal motor impairments [4]. Also, the program should be goal-based and focused on learning and practicing specific activities that are necessary to achieve and automatise functionality in daily routines and consequently enhance the QoL [2] of a specific patient.

Individual rehabilitation programs can be very time intensive to develop for each individual patient and execute with, for instance, a physical therapist. This may lead to these programs to become too expensive and limited by the number of therapists [7]. VR technology is a solution to facilitate rehabilitation programs for neurological diseases such as PD and to overcome the described disadvantages of personalized rehabilitation. This is further discussed in 2.1.4.

### 2.1.4. Rehabilitation

Rehabilitation, including exercise interventions, has proven to significantly improve overall physical functioning and balance without the negative effects of medicinal and DBS treatment. As the formation and retention of motor skills are impaired in PD, this should be taken into account when developing rehabilitation programs. In a review of Marinelli et al. on the many facets of motor learning and their relevance for PD they, describe the underlying mechanism of motor learning and how this knowledge should be implemented in developing suitable rehabilitation strategies for PD patients. They conclude that motor learning without awareness or cognitive strategies is normal in PD, in contrast to the retention of new skills, which is disturbed. Rehabilitation can therefore be focused on practicing already known activities to improve the ADL by addressing the dextrous deficits and consequently improving the QoL of patients with PD [14] [2]. Such dextrous tasks include, for example, getting dressed, cooking and opening a door. Because PD is a neuro-degenerative disorder, different approaches and content of the interventions are necessary at different stages of the disease and rehabilitation program should be tailored to the specific patient and their personal motor impairments [4].

## 2.2. Virtual Reality

VR can be described as computer-designed environments with which its users can interact in real-time through the use of a virtual interface. Such an experience immerses the user into their virtual surroundings and can be perceived as very real. The flexibility that VR offers in terms of the design of the environment, but also the possibility to give immediate feedback and track user performances, makes it very suitable for application in fields as rehabilitation. The implementation of VR into rehabilitation for neurological diseases has therefore been growing tremendously and shows increased participation, motivation and compliance during the interventions [3]. Also, VE environments create the possibility to create more fun and enjoyable experiences.

## 2.3. Haptics

Lately, the interest in creating VR environments that enhance the users' immersion even further by adding HFB to the experience has been growing. This section elaborates on the human haptic per-

ception system, the definition of haptic interaction tasks and what types of haptic feedback exist. An overview of interaction between a user and a virtual interface is shown in Figure 2.1 and focused on the haptic interaction during object exploration in Figure 2.1.

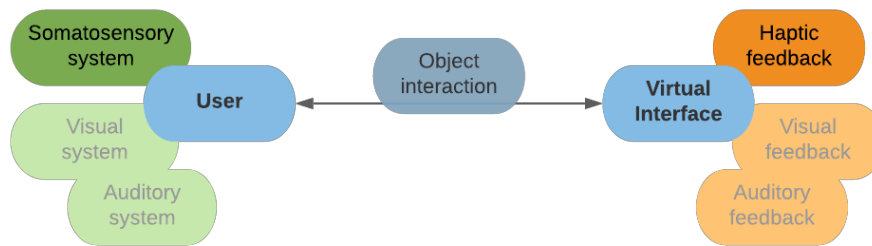


Figure 2.1: Interaction of user and interface

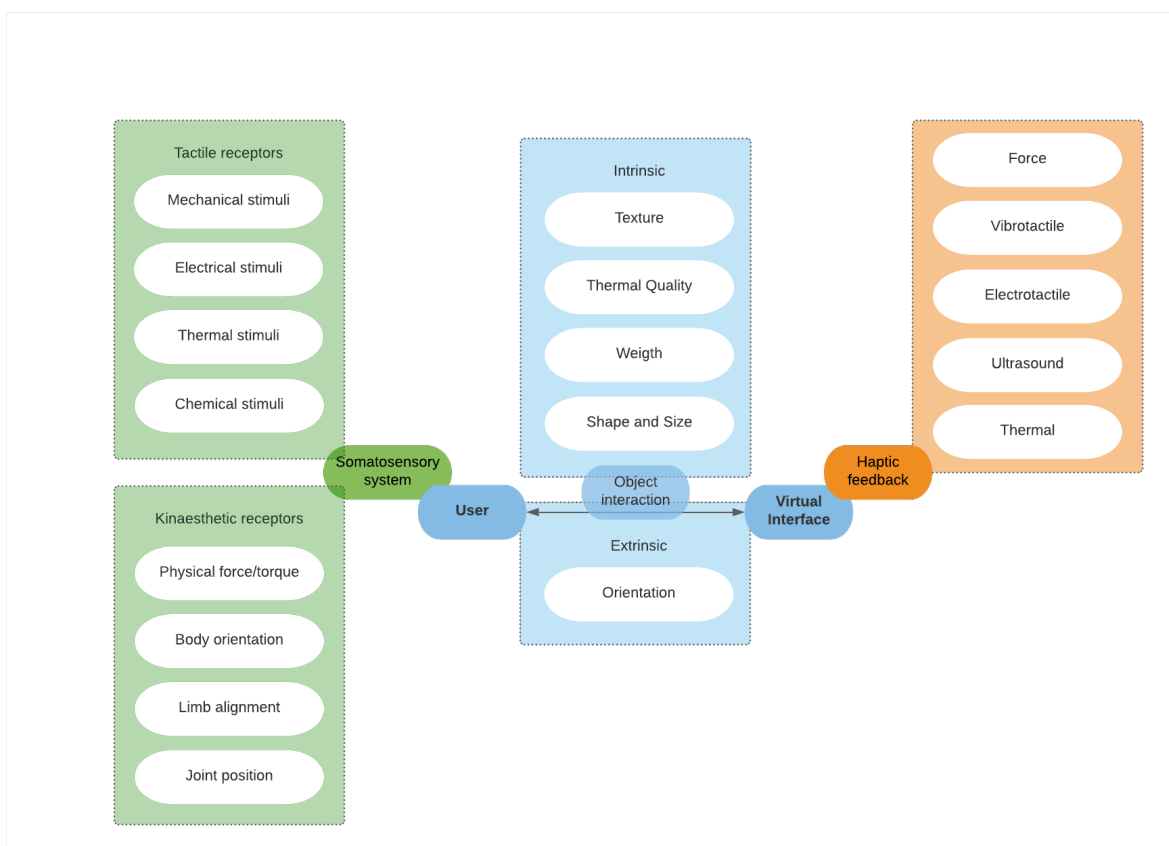


Figure 2.2: Overview of the haptic interaction between user and interface

### 2.3.1. Haptic perception

The haptic perception system of humans, also known as the somatosensory system, can be physiologically classified into *kinaesthetic* and *tactile perception* and is based on the location of the sensory receptors. For tactile perception these are the receptors in the human skin, also known as cutaneous receptors, that can perceive mechanical stimuli, such as high/low frequency vibrations, pressure and shear deformation, as well as electrical stimuli, temperature and chemicals [30]. As for kinaesthetic perception, there are sensory receptors in muscles, tendons, and joints that describe the operational state of the human locomotor system, such as joint positions, limb alignment, body orientation and

muscle tension. Since the operational state of the locomotor system is essential for the positioning of limbs, kinaesthetic sensing usually involves the motor control primitives [17].

The mechanoreceptors in the skin, responsible for the tactile perception, can be divided into four populations, SA I, RA I, SA II, RA II, with different primary functions each. SA I and II are involved in detecting sustained pressure and spatial deformation, such as perceiving object patterns, coarse texture and movement and are sensitive to frequencies below 5Hz. RA I and II are sensitive to higher frequencies and are involved in detecting temporal change of the skin, such as perceiving fine texture. Experiments have shown that for static tactile stimuli, the fingertip has a spatial sensitivity of 2 to 4 mm and the perception sensitivity for fingertip skin deformation is 10  $\mu\text{m}$  and 0.8 mN. For dynamic tactile stimuli the maximum perceivable frequency at the fingertip and palm is between 200 and 300 Hz with a perceivable threshold of 0.1-0.2  $\mu\text{m}$ . [19]

### 2.3.2. Interaction tasks

Hatzfeld and Kern described that the interaction between a user and a VE interface can be described through interaction tasks. The user can receive feedback, which can be haptic, visual and/or audio, from the interface while performing these tasks. In this chapter we will focus on the haptic interaction between the user and the interface, where the user perceives the haptic feedback through it's haptic sensory system, as shown visually in 2.1. Both the interaction tasks, as well as the classification of the somatosensory system and the types of haptic feedback are discussed in the following subsections. An overview of the interaction is shown in Figure 2.2. Haptic interaction tasks can be classified into perception operations and motion control operations, which can subsequently be classified into so called primitives. Operations that involve the perception of stimuli can be subdivided into four primitives [17]:

**Detection** This primitive describes how a stimulus is detected by a user and depends on the type and intensity of the stimulus together with the sensory organs and neural processing of the user. Only after a stimulus is detected, the other primitives apply.

**Discrimination** The primitive discrimination describes how information of multiple stimuli included in properties, such as frequency of a signal or hardness, mass, or texture of an object are compared to each other and consequently how this information is perceived.

**Identification** The primitive identification describes how the information of multiple stimuli is identified and compared to practical or abstract knowledge and consequently how this information is perceived. For example, the identification of geometric properties, such as size and shape.

**Scaling** The primitive scaling describes how the different properties of stimuli are are rated and how they relate to the perception.

Operations that involve motor control can be divided into three primitives:

**Travel** By movement, the user searches or reaches for an object or explores unknown environments.

**Selection of objects and functions** This primitive describes a direct interaction to a virtual environment by marking and selecting an object or function.

**Manipulation** The modification primitives describes a change in orientation, position, shape or other properties of an object.

### 2.3.3. Object Exploration

As most virtual interaction that includes haptic feedback involves the interaction with virtual objects, this sections elaborates on the properties of objects and how they can, as Lederman and Klatzky described, be manually explored through the human kinaesthetic and tactile perception. These object properties consist of intrinsic properties containing surface texture, thermal quality, compliance, weight, size, shape and the external property, orientation.

**Surface texture** The perceived texture on the surface of an object can be characterized by its roughness, stickiness, slipperiness or friction and is important in determining the way an objects should be handled to, for example, prevent it from slipping out of your hand. The texture of an object is mainly explored through lateral motion of the fingertips on the objects' surface and is therefor mostly dependent on tactile feedback.

**Thermal quality** The perceived thermal properties of an object are perceived through the thermal receptors of the skin and determine the apparent coolness or warmth of an object under contact due to thermal conduction. The thermal quality of an object is mainly determined by static contact between the object and a large skin surface and is therefore mostly dependent on tactile feedback

**Compliance** The compliance, or stiffness, of an object is its deformation under certain applied force. Research has shown that the compliance of continuously deformable materials, such as rubber, can be perceived through the tactile receptors of the skin only. On the other hand, rigid object surfaces require both tactile and kinaesthetic feedback in order to discriminate the spring constant of the object. (Srinivasan and LaMotte 1995). The compliance of an object is mainly explored by applying pressure to the object and, as described above, is dependent on tactile and kinaesthetic feedback, depending on the type of material.

**Weight** The weight of an object represents its density and structure and is important in determining the amount of force required to lift or move an object. Weight is mainly explored by unsupported holding of the object, but more detailed information on distribution of the mass can be better perceived by wielding an object. *kinaesthetic* feedback is the main haptic feedback that is used during weight determination.

**Size and shape** The size and shape of an object can be divided into two types of objects: those that are within the scale of a fingertip and therefore can rely on tactile feedback only and those that are larger than the fingertip, and have to be explored with the input of kinaesthetic feedback. The curvature of an object is also part of its size and shape.

**Orientation** The orientation of an object is the way in which the object is located and rotated in space. This is mostly determined by kinaesthetic feedback that occurs from gravitational forces of the object within a certain reference frame.

#### 2.3.4. Haptic feedback technologies

Haptic devices can be designed to provide haptic feedback by stimulating the tactile or the kinaesthetic perception of the user, or both. The general types of active haptic feedback can be characterized as force, vibrotactile, electrotactile, ultrasound and thermal feedback [16]. Each technology has its own characteristics, which are discussed together with the haptic perception it utilizes:

**Force feedback** This type of feedback can affect both the kinaesthetic receptors of the user and the tactile receptors, depending on the type of applied pressure. Indentation of the skin is perceived by tactile receptors, SA I for normal pressure and SA II for lateral skin stretch. Larger forces applied to the finger or hand are sensed by the kinaesthetic system of the user. Devices that use *passive force feedback* restrict movements of the user with the use of brakes, whereas active devices restrict or move the user by means of actuation.

**Vibrotactile feedback** This type of feedback relies on the tactile receptors, RA I and RA II, that can sense high frequency vibrations. An advantage of this type of feedback is that it is simple, easily implemented and controlled. However, vibrations may become irritating and the actuators are hard to efficiently miniaturize.

**Electrotactile feedback** This type of feedback not only relies on tactile receptors but also effects nerve endings with electrical action potential. An advantage compared with other types of feedback is that it can provide a wide range of sensations without any mechanical or moving parts. [18]

**Ultrasound feedback** Ultrasound is a high frequency sound wave. By using multiple ultrasound emitters, the user can feel ultrasound feedback through the generated turbulence which create invisible tangible interfaces in the air. This type of feedback is usually not as perceptible compared to other types.

**Thermal feedback** This type of feedback is based on thermal actuators that are in close contact with the skin. Because the human does not have high density thermal receptors, few thermal actuators are needed to stimulate the skin. However, the heat needs to be moved around due to the law of energy conservation. Therefore, this method generally costs more energy, because the heat needs to be moved around very quickly for realistic feel.

# 3

## Design

This chapter elaborates on the approach of the design of the HD during the course of the thesis project.

### 3.1. Design Requirements

To address the main design goal provided in chapter 1, a list of requirements is made for the HD to be designed. The requirements are divided into three main categories: Technological and practical requirements.

#### 3.1.1. Technological requirements

These requirements contain aspects that concern the type of haptic feedback, the actuation technology and characteristics, degrees of freedom etc. Should be based on the sensory thresholds in PD patients.

- Haptics should stimulate the tactile perception
- Latency of less than 50ms
- Intensity should be variable
- Compatible with the VRFree

#### 3.1.2. Practical requirements

These requirements concern the comfort and usability of the glove. Paccierotti et al. described a few target objectives for the design of wearable interfaces [22], which are important in the wearability and usability of the device.

- Weight; Wearable devices should tire the wearer as little as possible.
- Form factor; Wearable devices should alter the body size of the wearer as little as possible.
- Impairment; Wearable devices should limit the motion of its wearer as little as possible
- Comfort; Wearable devices should be comfortable to wear and easy to adapt to the wearers limb.

### 3.2. Design decisions

#### 3.2.1. Type of haptic feedback

From the literature review done on haptic device, the following insights were gained. There are many haptic technologies out there. Many of which, are technically tested but not yet verified in actual VR implementation, showing that the designing a device is one thing, but implementing is a whole challenge by itself. For the purpose of the project, the HFB types, that would be feasible are tactile force feedback, kineasthetic force feedback, vibrotactile feedback, thermal feedback and electrotactile feedback. The four types of HFB are rated against their versatility, usability and realism in order to determine which type to use in the design of the HD.

**Comparison** Figure 3.1 shows an overview of the types of haptic feedback, rated on their versatility, usability and realism. The colour scheme is based on the values stated in Table 3.1. Versatility

	<i>Green</i>	<i>Orange</i>	<b>Red</b>
<b>Versatility</b>	Addresses three or more object properties	Addresses two object properties	Addresses one object property
<b>Usability</b>	Is very small and only one component is needed to implement in a glove	More than one component is needed to implement, but it would still fit in a glove	Is very bulky and needs multiple components to implement, making it difficult to implement in a glove
<b>Realism</b>	The perceived feedback is very close to natural haptics	The perceived feedback is some what close to natural haptics	The perceived feedback is very far from natural haptics

**Table 3.1:** Haptic feedback type decision table complementary information

	<i>Versatility</i>	<i>Usability</i>	<i>Realism</i>
Tactile Force Feedback	Orange	Orange	Green
Kinesthetic Force Feedback	Orange	Red	Green
Vibrotactile feedback	Green	Green	Orange
Thermal Feedback	Red	Orange	Green
Electrotactile feedback	Orange	Red	Orange

**Figure 3.1:** Haptic feedback type decision table

describes how many types of object properties the technology type has proven to address. The usability in terms of the design of the device, how many components are needed. The realism describes the perceived realism of the users during the experiments on different object tasks.

The results from the review show that both tactile and kinaesthetic force feedback generally provides a higher realism, however the devices are less versatile in terms of feedback due to a low amount of variation in feedback possibilities. Next to that, they are relatively big and due to the exerted forces on the hand, a counter force is perceived on another part of the hand or fingers. Vibrotactile feedback is implemented in many of the studies from the review, as it is very versatile in conveying different feedback intensities and due to the size of the actuators easy to use, however vibrations are perceived as less real. Thermal feedback was only tested for thermal object qualities and therefore less versatile, but scores high in sense of realism. Electrotactile feedback can be perceived as highly irritating and is not yet tested with many object interactions. There were no general guidelines on testing the technologies, so it was difficult to compare and this overview is based on the conclusions that the different studies have made. Also, proper visual feedback and training plays a very important role in the performance of the tasks, which was not similar in every case.

From this overview and all information from the studies from the literature review, VTF seems to be the most fitting and implementable type of feedback for the pilot purpose of this project.

### 3.2.2. Type of actuator

VTF is usually implemented through the use of vibrational actuators. Three main type of actuators are Eccentric Rotating Mass (ERM) actuators, Linear Resonant Actuators (LRAs) and lately, also piezo actuators have been used [1].



### **ERM actuators**

ERM actuators consist, as the name suggests, of a mass that rotates around a motor shaft, causing vibration in two directions. They can be either brushed or brush-less, depending on the construction of the motor. In a brushed motor, the shaft rotates due to a current through armature windings that are inside a magnetic field induced by a static magnet around the moving shaft. A commutator on the rotating shaft is connected to static brushes, causing reversal in current, creating continuous rotation of the actuator. Brush-less actuators, on the other hand, avoid the use of brushes by having the magnet attached to the rotating shaft. A stator contains coils to which power is applied consecutively in a pattern and the occurring forces cause the shaft to rotate. Because a ERM actuator is driven with DC current, the vibration frequency and amplitude is directly related to the input voltage, creating low versatility. Also, they are relatively noisy when actuated. These actuators can be bought of the shelf relatively small, with a diameter of 10mm.

### **LRAs**

LRAs work through the principle of a voice coil that induces a magnetic field and interact with a magnetic mass connected to a spring. This combination, however, causes resonance in the system, which is where the name comes from. A LRA works optimal around its' specific resonance frequency, which is usually around 200 hz, but is actuator dependent. The frequency and amplitude of the input signal are independent of each other, which allows for creating more detailed vibrations. These actuators can be bought of the shelf relatively small, with a diameter of 8mm.

### **Piezo actuators**

Piezo actuators utilize the piezoelectric effect of piezo materials. These materials expand when a voltage is applied and are therefore able to vibrate when the voltages are varied. The advantages of piezo actuators are that no magnetic field is induced, minimizing interference with other electronics, very minimal amount of moving parts and they have low latency. However, they need a very high driving voltage, and are only available off-the-shelf with a minimum diameter of 20mm.

### **Comparison**

Figure 3.2 shows the three types of actuators, rated against five important characteristics for the design of the HD, sorted on highest importance on top. The colour scheme compares the actuators based on the values stated in Table 3.2. The size is based on the size that would fit on a fingertip. The latency is based on the requirement of a maximum latency of 50ms between visual and haptic feedback to not be detected [9]. The energy consumption during a buzz is based on the average consumption of the three actuators [15]. The bandwidth is based on the three characteristics of the actuators.

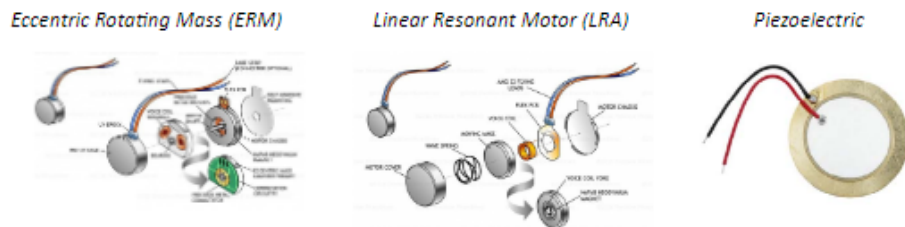
In terms of size, both the ERM and LRA score best, because the piezo actuator has not been found with a smaller diameter than 20mm. This would not fit on a users' finger, which is not desirable. In terms of latency, the piezo actuator scores highest, as it does not require the response of a moving mass, but is directly related to the applied Voltage. The LRA scores second best as it has a faster response time than an ERM, which needs time to reach its rotational velocity. Both the ERM and LRA can be driven with low voltages between 1 and 5V. A piezo actuator needs a voltage in the order of tens to hundreds of Volts. On the other side, the piezo consumes less power than a typical ERM actuator during a buzz. Because a LRA actuator works optimal around its resonance frequency, it has a relatively low bandwidth. A piezo has a very high resonance frequency, providing a high bandwidth. From this overview, it is concluded that for the objective of this project, the LRA seems like the best option.

## **3.3. Preliminary Tests**

To determine the needs of the PD patients more clearly, specifically in terms of location and intensity of the VTF, some preliminary testing has been done. A testing plan has been made before hand and is shown below. The tests are performed on five patients whom have participated in a VR rehabilitation test just before.

	<i>Green</i>	<i>Orange</i>	<i>Red</i>
<b>Size</b>	D<15 mm and t<3 mm	D<15mm and t<3mm	D>15mm and t<3mm
<b>Latency</b>	Latency <10 ms	10-50 ms	50 ms >
<b>Working Voltage</b>	< 2 V	2-5 V	5 V >
<b>Energy consumption during buzz</b>	< 4 $\mu$ Ah	4-15 $\mu$ Ah	15 $\mu$ Ah
<b>Bandwidth</b>	Frequency does not affect vibration and resonance frequency is 400Hz >	Frequency and amplitude are independent, resonance frequency <400Hz	Frequency and amplitude are dependent, resonance frequency <400Hz

**Table 3.2:** Actuator decision table complementary information



Size			
Latency			
Working Voltage			
Energy consumption during buzz			
Bandwidth			

**Figure 3.2:** Actuator decision table [29]



**Figure 3.3:** Locations of the actuators

### 3.3.1. Execution

*Two Point Discrimination (5 min):*

1. Varying between 1 point and two points on locations (1,2,8) of patients hand
2. Patient indicates the amount of points felt
3. Patient has three tries per distance
4. Note minimum distance

*Vibration Fork Test (5 min):*

1. Vibrating the fork
2. Placing it on patient on the two locations
3. Patient indicates when no more vibration is felt
4. Note the scale value (1-8)

*Vibration Actuator Test (5 min)*

1. Putting on glove
2. Starting actuator sweep from zero to highest amplitude vibration on at a time on on locations (1,2,8)
3. Patient indicates when feeling vibration
4. Patient indicates when uncomfortable
5. Patient indicates where vibration is felt

*Questions (5 min)* The corresponding questionnaire is depicted in Figure B.1

### 3.3.2. Results

The results from the preliminary testing confirm that indeed the perception of vibrations is not very disturbed according to the vibration fork test. The two-point discrimination test showed varying results per patient, and thus no conclusions were taken from here. The vibration actuator test made clear that the design of the implementation of the actuators needs to be optimized. The lower intensities were not very well perceived, due to the fact that the actuators were located outside of the glove and that due to the differences in hand sizes, the actuators did not always fit close to the fingers and especially the palm. All intensities were perceived as comfortable, non-irritating and not distracting by all patients. The opinions on the facts if the vibrations could make a VR experience more realistic and easier were varied. The idea of having vibrations during a VR task was difficult to imagine and not all patients were convinced that it would work. Most patients stated that the most logic location of the actuators would be at the fingertips as that would be where they touch an object during a dexterity task. Therefore it is chosen to locate one actuator at each fingertip, as shown in Figure 3.3

# 4

## Prototype

This chapter contains information on the hardware and software used in the build prototype, including all the scripts and 3D models that have been made. It also contains a manual on how to use the HD in combination with a VE. The overall communication and layout of the system is shown in Figure 4.1.

### 4.1. Hardware

This section contains all the details on the hardware that is integrated into the prototype. An image of the prototype in use including all the used hardware is shown in Figure 4.2

#### 4.1.1. VRFree Glove

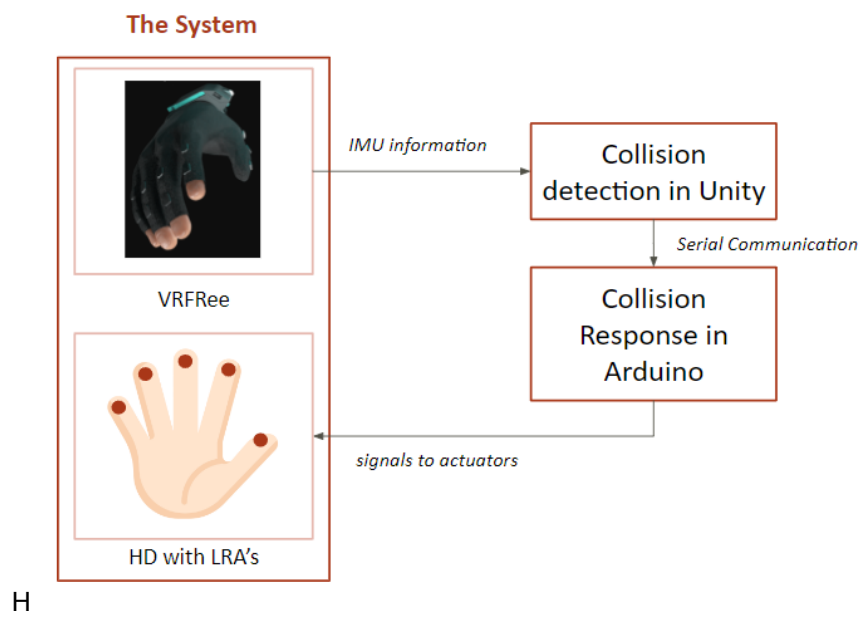
As a basis for the HD prototype, the VRFree gloves by Sensoryx are used. The VRFree is a glove that allows for plug and play ease of use and is highly wearable because it is ungrounded and wireless. It consists of two gloves with eleven Inertial Measurement Unit (IMU)s each, that send data to two wrist-modules which communicates with a head-module mounted on the Head Mounted Display (HMD). It is able to track the orientation of the individual phalanges and use this data to render the hands in VR in a realistic manner.

#### 4.1.2. HMD

At first, a HTC vive is used as it was readily available. Unfortunately, the integration of the VRFree gloves with the HTC vive did not seem to work due to the controller bindings that were not able to synchronise how they should. The VRFree gloves come with a Unity Plugin which controls the physics and sensor data and converts it to the rendering of the hands in VR. This plugin unfortunately is only compatible with the Oculus and not with the HTC vive, as the HTC vive works through a platform called SteamVR. A way to work around this, is by using a Unity SteamVR package to use the Steam Environment. However, this package is not compatible with the VRFree unity plugin, so an additional VRFree SteamVR plugin is necessary to communicate with the gloves through SteamVR. Unfortunately, this construction was very unreliable as it showed a lot of inconsistent results in terms of communication that got disrupted with the gloves etc. Also, the controller bindings were not properly saved after defining them, making it a very unreliable method. Next to that, the HTC Vive needs an extra external box as a grounded reference point to the HMD, making it a less flexible set-up for applying in different locations. These setbacks made it necessary to find another HMD, which became the Oculus Quest 2. This HMD does not need an external reference point, and is very user friendly and compatible with the Unity Platform, making the integration of the VRFree glove with the Plugin much more straight forward.

#### 4.1.3. Electronics

This section elaborates on the choice of electronics. Figure 4.3 shows an overview of how the electronics part looks.



**Figure 4.1:** Overview of system communication



**Figure 4.2:** Prototype in use with all the hardware

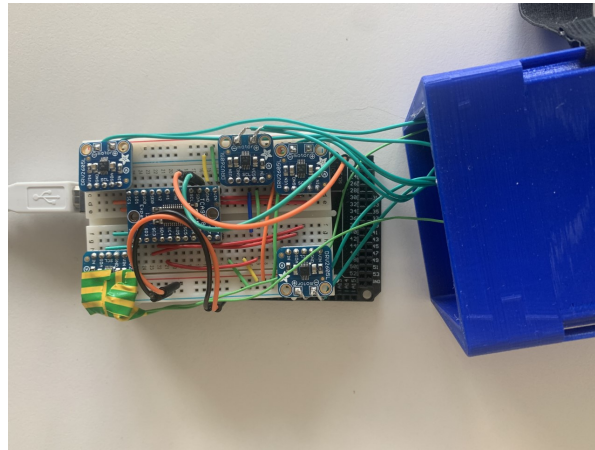


Figure 4.3: Overview of electronics inside the HD

### Micro-controller

As main controller of the haptic feedback, a micro-controller is used. The Joy-it MEGA2560R3 was available and fits the needs.

### LRAs

The choice for LRA actuators has been elaborated on in chapter 3. To minimize weight and size, and consequently maximize wearability, the smallest available off-the-shelf LRAs are used for the purpose of this study. These are round actuators with a diameter of 8mm that have a rated voltage of 1.2V. In total five of these actuators are used to convey the VTF. The actuators are located in 3D printed fingertip parts, which is further elaborated in subsection 4.1.4.

### Haptic Drivers

In order to control the actuators, a DRV2506L is used. This is a haptic motor controller that is specifically designed to drive haptic motors, such as ERMs or LRAs. The driver is controller over I2C through the SDA and SCL ports of the micro-controller. Each individual actuator needs it's own driver to be able to control the actuators individually, which comes down to a total of five drivers.

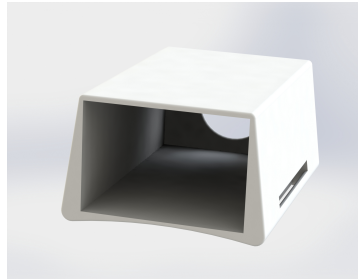
### Multiplexer

Most micro-controllers, including the Joy-it MEGA, only have one SDA and one SCL port. To be able to control all the actuators individually through a single master, a multiplexer is needed to communicate to multiple I2C slave devices. For this purpose, the TCA9548A 1-to-8 I2C multiplexer is used.

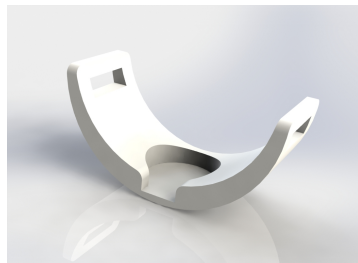
#### 4.1.4. Casing

All Solidworks models can be found in the attached digital folder in *Myla Thesis Addition Digital/Solidworks*. To create a wearable device, a 3D printed case is designed in SolidWorks to be worn around the wrist, as shown in Figure 4.4 and the dimensions shown in Figure C.1. The micro-controller and the breadboard with the drivers and multiplexer are fitted inside this case. The case is designed such that it fits around a medium sized wrist and can be adjusted with Velcro straps that are attached to the case.

The cases in which the actuators are held to be properly fitted around the finger are also designed using SolidWorks. All fingers have different sizes, which is why it is chosen to design a total of six sizes. This way, the best fitting finger cases can be used for each individual patient. At first the finger cases are designed with an extra elastic band to keep the actuator in place, as shown in Figure 4.6. However, this meant that a constant pressure was put on the users' fingers, which made it uncomfortable to wear. Therefore, the next iteration was without the elastic bands and where the parts were already fitted inside of the glove, to increase the ease of use and having the ability to work the cables around in the glove, so that they are not in the way during putting them on and off. Figure 4.7 shows an overview of how the actuators are arranged inside the glove.



**Figure 4.4:** 3D design of wrist case



**Figure 4.5:** 3D design of finger part



**Figure 4.6:** Printed finger part with elastic band

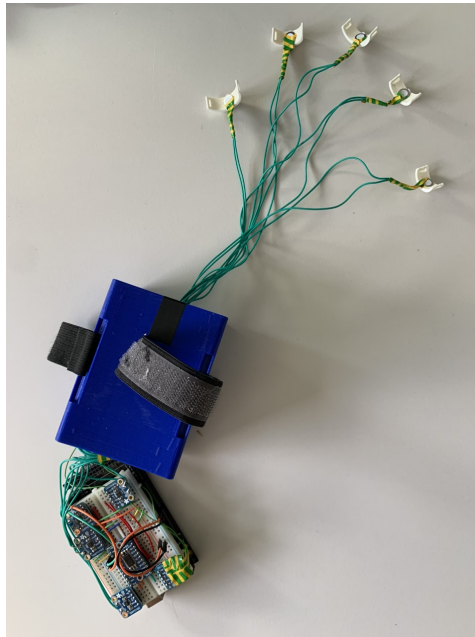


Figure 4.7: Overview of actuators as arranged in the glove

## 4.2. Software

The different scripts that describe how the haptic feedback is initialized in the Unity Software and communicated to the micro-controller are shown in this Section. Using the unity Software together with the VRFree Unity Plugin, the grabbing detection in VR is done. After a "haptic" object is grabbed, a signal is sent to the micro-controller through the serial port to which the controller is connected.

The code in Listing A.3 contains the code in which the serial communication is handled in Arduino. The code in Listing A.4 contains the initialization and calibration of the actuators in Arduino.

## 4.3. Short HD Manual

This section elaborated shortly on the steps that are taken when using the system. Ofcourse, during this process many small steps can go wrong when using a different laptop or different HMD, but in general for using the designed device in a VR environment, the following steps should be taken:

1. The headmodule of the VRFree should be connected to the HMD and the laptop that is used.
2. The HMD should be tuned on according to the manual.
3. The gloves with HD and headmodule need to be worn and turned on.
4. The gloves need to be calibrated according to the VRFree setup calibration.
5. The Arduino software (UserTestsWithHaptics.ino) should be loaded onto the HD.
6. For use in the virtual dexterity grasp scene, the TestingGrabbingBug project should be opened in Unity.
7. The intensity of the vibrations can be chosen in the Unity Game as shown in Figure 4.8.
8. The game can be tested.



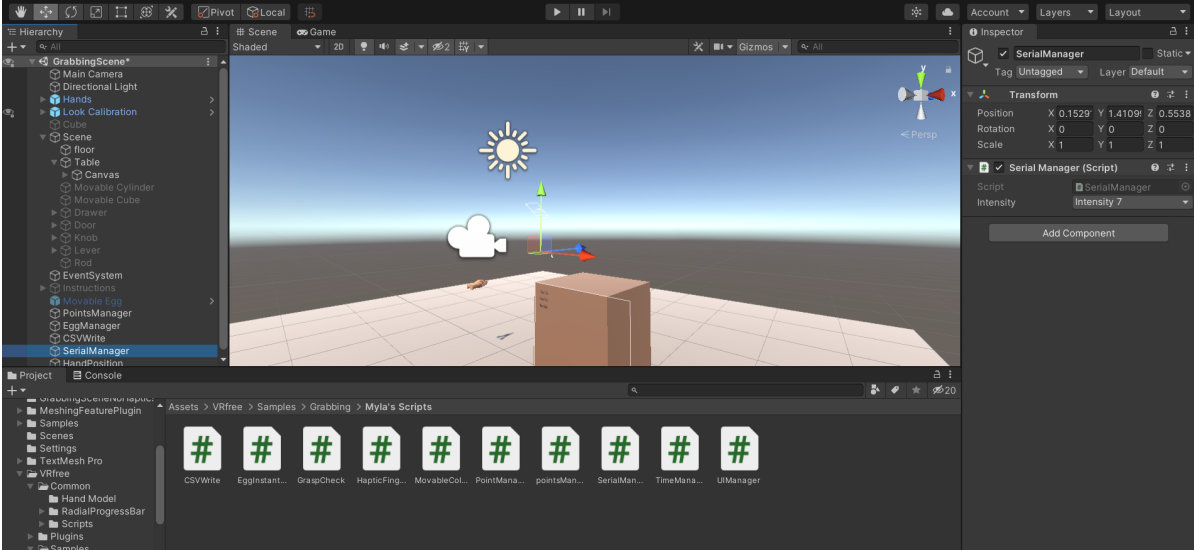


Figure 4.8: Image of Unity Game Screen

# 5

## Experiments

This chapter contains information on the setup of the experimental parts of the project and includes the scripts that are used for data collection, and how the data is visualised and analysed using MATLAB. Both the set-up and design of the technical tests as well as the user tests are discussed.

### 5.1. Technical tests

This section elaborates on the design and execution of the technical test that have been performed on the HD.

#### 5.1.1. Latency

The setup that has been used for testing the latency and intensities of the HD is visually shown in Figure 5.1. With this set-up it is possible to measure the acceleration of the LRA after a signal has been started. As a LRA moves only along one axis, only one axis needs to be measured. However, from the off the shelf accelerometers, the Adafruit LIS3DH Triple-Axis Accelerometer seemed the most suitable for the purpose. It is able to sense accelerations in a range of 2,4,8 or 16G, of which 2G is chosen as setting. The micro-controller is programmed through the Arduino IDE and the data from the accelerometer is collected using a terminal program for serial communication and stored in a .csv file. This is then analysed using MATLAB. The code that has been build for the micro-controller is depicted in subsection 5.1.4.

#### 5.1.2. Intensity

The code that has been build for the is for the acceleration mainly similar to the one from the latency tests as depicted in subsection 5.1.4.

#### 5.1.3. Compatibility

To test the compatibility of the HD with the VRFRee sensors, a compatibility test is performed. The main goal is to find out whether or not the magnetometers of the IMU's inside the glove do not get disturbed due to magnetic interference.

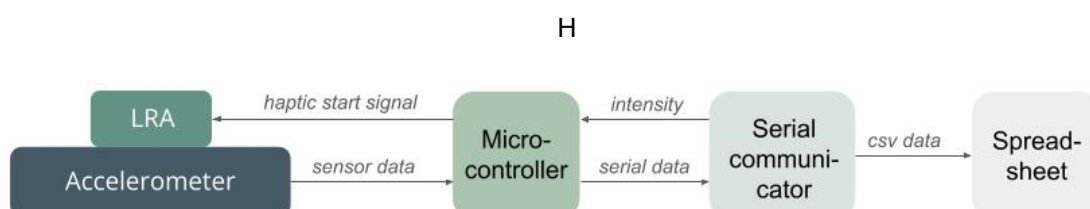


Figure 5.1: A visual representation of the latency and intensity testing setup

### 5.1.4. Scripts

This section contains all the scripts that have been made for the technical tests. The micro-controller is programmed through the Arduino IDE in programming language C++. The collision detection and consequently haptic start signal is programmed through the Unity software and Visual Studio in programming language C#. The VR experiment scene is also programmed and build within Unity.

The code in Listing A.1 contains the parameters that are set for both the actuators and the accelerometers during the latency and the intensity tests. The code in Listing A.2 is the accelerometer specific code for both the latency and the intensity testing. It contains the initialization function of the accelerometer as well as how the data is handled and visualized.

## 5.2. User tests

To actually test the compliance and impact on the dexterity of PD patients, the HD is also tested with users. This section elaborates on the design and execution of the user test that have been performed on the HD.

### 5.2.1. Participants

The participants for this study are five patients with PD and six healthy subjects. It was observed during another study in the same lab with both PD patients, that a bias in the results occurred due to the fact that the healthy subjects were around the age of 20 to 30 and had more technological experience than a general PD patient group. To prevent this and filter out a any bias that may occur due to familiarity and experience with the used VR technologies and games, for this study, the healthy subjects are age-matched. The user test consist of a practical part where a VE is used and numerical data is obtained as well as a questionnaire in which subjective feedback on the HD and system is collected.

### 5.2.2. VR test

The VE for the VR test is designed and build using Unity 3D software. This software is highly user-friendly in creating VEs.

#### VE design

A simplistic environment is designed to make it as clear and non-distracting as possible for patients. The aim is have it in such a way that the task is very self-explanatory, obvious and can be full-filled without any extra instructions during the test. The VE exists of a brown table on which a real-life-sized egg appears at random locations. The random locations prevents automation of the movement and makes sure the brain has to evaluate and execute a new grasping path each time. It is made sure, however, that the distance from the user to the egg is not too far away to prevent fatigue. After the egg appears, the user has to grasp the egg with the hand and hold it for three seconds. This is done to practise the force control during the grasp of an object. When the fingers "penetrate" the egg too much, the egg "breaks" and disappears, and counts as a bad grasp. When the egg is held for three seconds it also disappears, and counts as a good grasp. In total the subjects perform ten grasps per session, with a total of three sessions. First a session with the haptic feedback, than a session without the feedback, and concluding with a session with haptic feedback. Prior to the three sessions, the subjects are asked to practise the grasps without feedback until they are familiar with the device. This is done to prevent the results to be influenced through a learning curve.

Also, an actual real grasp off an egg grasped as base line for the finger distance to determine whether a grasp is bad. Here, a real egg is placed at random distances from the user, similar to the VR setting, with eyes closed. After opening the eyes, they grasp and lift the egg for three seconds and place it back. The data of the real grasp is collected.

#### Scripts

This is a short discription of the scripts for how the user data has been measured and how the VR scene is made in Unity. In Unity, the code is written in C, which is an object oriented language. All the used scripts in unity can be found in the attached digital folder in *Myla Thesis Addition Digital/Unity/Testing-GrabbingBug/Assets/VRfree/Samples/Grabbing/Myla's Scripts*.

The basis of the virtual grasping game consists of the table and egg. To manage the different eggs appearing, an EggManager script is made, Listing A.8. To manage the amount of good, bad and total grasps, a PointsManager is created, Listing A.11. The moment that an egg is grabbed, the haptic

feedback is started. This is done by checking if the hand is actually grabbing the egg, which is already implemented into the VRFree Unity Plugin. When the egg is grabbed, the positions of all fingers with respect to the thumb is compared to a certain minimum value below which the egg will break. Listing A.5. This value is determined during some grasps with a real egg. To send the data for the haptic feedback to the micro controller, a Serial Manager is created, Listing A.9.

### 5.2.3. Questionnaire

The questions that have been implemented in the questionnaire have been selected carefully. The main objectives of the questionnaire are to evaluate the haptic experience, and the usability and compliance of the user to the system. Figure B.2 shows all the questions that have been originally considered and which ones have eventually been chosen for use in the questionnaire. For the Haptic Experience a Haptic Experience (HX) questionnaire is found. Even though this questionnaire is not yet validated, it contains very relevant questions that are categorized into five categories: Autotelic, Immersion, Realism, Expressivity and Harmony. The questions regarding expressivity do not fit the purpose of the designed HD and are left out. For compliance, some questions from the User Satisfaction Evaluation Questionnaire (USEQ) questionnaire are adopted. A presence questionnaire is also considered, but eventually left out, as it does not comply to the purpose of the questionnaire with respect to the HD. To test the usability of the system, the complete System Usability Scale (SUS) questionnaire is used. This is a standardised questionnaire to evaluate the usability of a system and is widely used in research. Therefore, the factor that comes out of this questionnaire can be interpreted as a validated outcome of the systems usability. Most questions are evaluated through a 5-point Likert Scale ranging from *Strongly Disagree* to *Strongly Agree* and *Completely Not* to *Very Much*. With this, the results can be collected and evaluated in a quantitative manner. The final questionnaire is found in Appendix B

### 5.2.4. Data Analysis

The collected data from the user test contains the position of the wrist, the finger distances relative to the thumb, the position of the egg, which grasps were good and which were bad and the time. This data is analysed through the use of MATLAB, for which a script is written for both left hand analysis and right hand analysis. All the used matlab scripts can be found in *Myla Thesis Addition Digital/Matlab*. From this data, *the average completion time per grasp* is calculated, as well as *the average completion time per grasp per travelled distance*, *the jerk* and *the amount of good grasps*. From all these characteristics the Haptic Factor (HF) is calculated, which is described in Equation 5.1. This describes the ratio of a measured characteristic where haptic feedback was applied, T, to the no feedback condition, NH. A factor of unity means they were similar. With this factor the relative performance can be evaluated easily.

At first, the grasp path was compared with a calculated straight path going from the start point of the wrist to the location of the egg. The average difference between the straight path and the actual travel path was calculated as a measure of optimal and smoothest grasping path and consequently confidence. This was based on an experimentally confirmed mathematical model on the coordination of arm movements during a point-to-point task, which was based on the jerk integrated over the entire movement. This showed that the smoothest path was mathematically a straight line and experimentally almost exactly similar [13] However during the VR experiments of this project, the grasping paths differed quite a lot and change when the egg moves after touching. Therefore it is chosen to directly use the jerk factor during the grasp as a measure of smoothness. This is already widely used to evaluate grasping trajectories [23] [11].

$$HF = \frac{T}{NH} \quad (5.1)$$

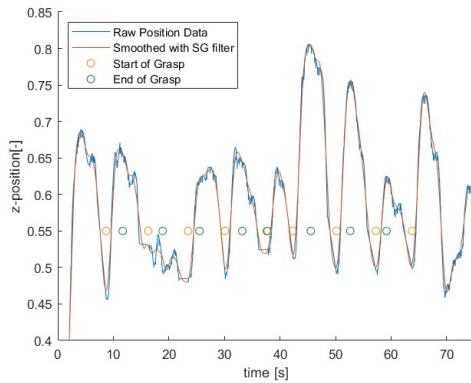
**Filtering** After importing the position data, the data is first linearly interpolated to create a uniform data set. This is done so that a Fourier Transform can be performed to analyse the data in the frequency spectrum and to be able to filter out noise. However, the data shows no obvious bandwidth which should be targeted. Therefore it is decided to use a Savitzky-Golay filter on the time domain data. This is a digital moving average filter that uses the least-squares method to fit successive subsets of data points to a low-degree polynomial. This increases the precision of the data without changing the original tendency of the data set, which is important as we want to eventually calculate the jerk as a measure of

smoothness. Figure 5.2a and 5.2b show a set of position data, with the smoothed data and the marked beginning and end times of the grasp.

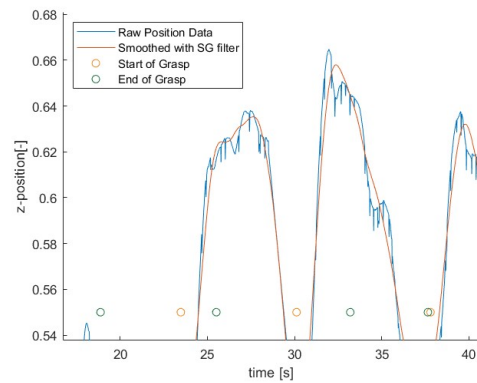
**Extracting Jerk** The derivative is taken from the smoothed position data to obtain the velocity, following Equation 5.2. The position data is again smoothed and the derivative is taken to obtain the acceleration. This is repeated to obtain the jerk.

$$v(i) = \frac{p(i+1) - p(i)}{t(i+1) - t(i)} \quad (5.2)$$

Only the average jerk during the actual grasp is wanted. Because the egg has to be held for three seconds, the marked end-time of the grasp is not the actual end of the grasping trajectory and so the grasping trajectory is taken between the starting grasp time and ending grasp time minus three seconds. For future studies it would be easier to store the acceleration data of the IMUs directly, instead of the position. It would make the calculations of the jerk more accurate, as the data would be less manipulated. Also, it could be interesting to try out more types of smoothness characteristics and see what each one shows as result. Figure 5.3a and Figure 5.3b show an example of smoothed position and jerk data from a healthy subject respectively. Figure 5.4a and Figure 5.4b show an example of smoothed position and jerk data from a PD patient respectively.

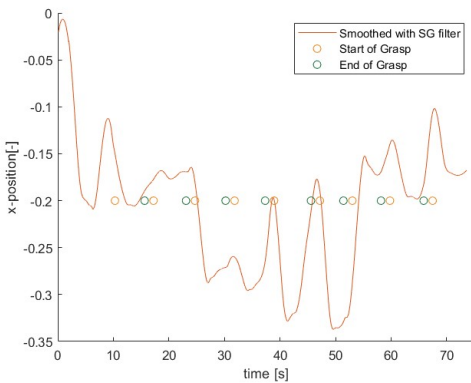


(a) Graph of position data and grasp timing

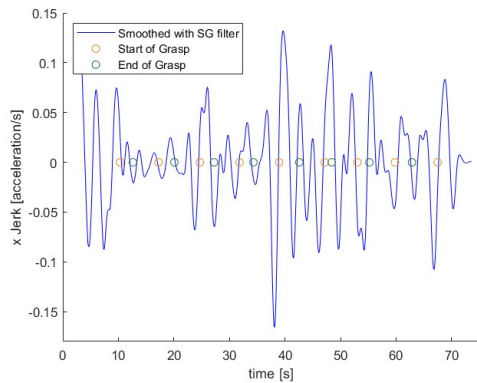


(b) Close-up graph of position data and grasp timing

Figure 5.2a

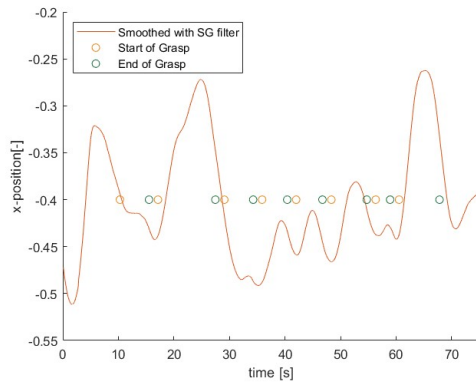


(a) Graph of smoothed position data healthy subject

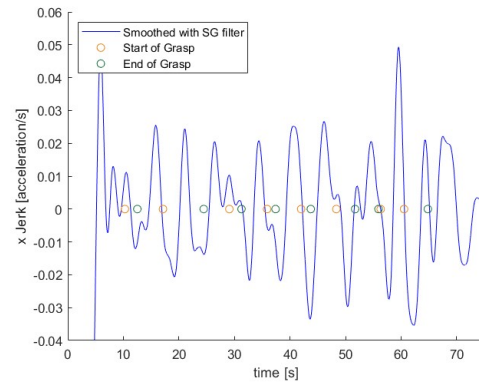


(b) Graph of smoothed jerk data healthy subject

**Extracting overall completion time and per travelled distance** The completion times are calculated by finding the difference in time stamps between the appearing of the egg, which is marked as the start of the grasp, and the moment the grab is completed, ie. the egg has been held for three seconds, and then subtracting three seconds to find the end time of the grasping trajectory. The travelled distance is the distance between the wrist at the start of the grasp and the position of the egg at the start of the grasp.



(a) Graph of smoothed position data of PD patient



(b) Graph of smoothed jerk data of PD patient

**The amount of good grasps** The amount of good grasps is already stored in the saved data during the test and can be directly evaluated. However, it is checked that no strange behaviour, such as extremely short grasps of less than 1 second, has occurred during the test.

### 5.3. Unforeseen Events

During the experiments, multiple unforeseen events have happened. This section will elaborate on the unforeseen events, how they have been handled and how they can be prevented in the future.

*Gloves* During the tests with patients, unfortunately the gloves did not work properly. It might be nice to always have back-up parts during an important test.

*Actuators* The actuators were delivered with very small cable attached to it. These cables seemed to be very vulnerable, as they would sometimes get loose during taking the glove on and off. Therefore it was chosen to make them stronger by adding hot glue and electrical tape and bringing spare actuators to the tests.

*Accelerometer* It was tried to use an external accelerometer during the user experiments to measure the acceleration of the wrist. However, the amount of serial data that needed to be sent from unity to arduino became too high, creating a lot of noise and inconsistencies. Therefore it was decided to use the position data from the IMUs. However, looking back, it would have been smart to use the accelerometer data from the IMU's.

*Data Analysis* Prior to the tests it was decided which data would be relevant for further analysis. However, it would be recommended to check one sample set of acquired data, to actually check if you have collected everything you wanted. The sampling time of the data during this study was not uniform, making it necessary to interpolate between the data points to create uniform data which could then be transformed to the frequency domain.

# 6

## Conclusion and future recommendations

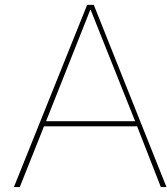
The overarching goal of the thesis project was to determine the usability of HFB in VR rehabilitation for PD patients to evaluate whether it can enhance their dexterity and consequently would improve their ability to perform ADL. This was done by first performing an extensive literature studies as reported in a paper called: *Virtual interaction; an overview of wearable haptic technologies and their performance in virtual object exploration*. Then, a prototype is designed that utilizes VTF to implement HFB in the VRFree gloves. With the prototype a usability study was performed both on a technical level as well as on a user level with healthy subjects and patients with PD. This was reported by means of a paper called: *A novel virtual reality glove system with integrated vibro-tactile feedback for Parkinson's Disease: A usability study*. The results of this study indicated that indeed VTF as a means of haptic feedback during rehabilitation could enhance dexterity in ADL in PD. However, there is still improvement possible in terms of the design of the HD and further studies will be needed to evaluate its value as a rehabilitation tool to improve QoL in PD.

# References

- [1] *AB-003: Driving Linear Resonance Vibration Actuators - Precision Microdrives*. URL: <https://www.precisionmicrodrives.com/ab-003>.
- [2] Giovanni Abbruzzese et al. "Rehabilitation for Parkinson's disease: Current outlook and future challenges". In: *Parkinsonism and Related Disorders* 22 (Jan. 2016), S60–S64. ISSN: 18735126. DOI: 10.1016/j.parkreldis.2015.09.005.
- [3] Jashvini Amirthalingam et al. "Virtual Reality Intervention to Help Improve Motor Function in Patients Undergoing Rehabilitation for Cerebral Palsy, Parkinson's Disease, or Stroke: A Systematic Review of Randomized Controlled Trials". In: *Cureus* 13.7 (July 2021). DOI: 10.7759/CUREUS.16763.
- [4] Paolo Borrione et al. "Effects of physical activity in Parkinson's disease: A new tool for rehabilitation". In: *World Journal of Methodology* 4.3 (2014), p. 133. DOI: 10.5662/WJM.V4.I3.133.
- [5] A. Collomb-Clerc and M. L. Welter. "Effects of deep brain stimulation on balance and gait in patients with Parkinson's disease: A systematic neurophysiological review". In: *Neurophysiologie Clinique/Clinical Neurophysiology* 45.4-5 (Nov. 2015), pp. 371–388. ISSN: 0987-7053. DOI: 10.1016/J.NEUCLI.2015.07.001.
- [6] Antonella Conte et al. "Abnormal temporal coupling of tactile perception and motor action in Parkinson's disease". In: *Frontiers in Neurology* 8.JUN (2017). ISSN: 16642295. DOI: 10.3389/fneur.2017.00249.
- [7] Iulia Cristina Stanica et al. "Parkinson's Disease - How Virtual Reality Exercises can Improve the Neurorehabilitation Process". In: *IEEE International Conference on E-health and Bioengineering, 2019*.
- [8] Marcio Alves De Oliveira et al. "Strength and isometric torque control in individuals with Parkinson's disease". In: *Experimental Brain Research* 184.3 (2008), pp. 445–450. ISSN: 00144819. DOI: 10.1007/s00221-007-1212-9.
- [9] Massimiliano Di Luca and Arash Mahnan. "Perceptual Limits of Visual-Haptic Simultaneity in Virtual Reality Interactions". In: *2019 IEEE World Haptics Conference, WHC 2019*. Institute of Electrical and Electronics Engineers Inc., July 2019, pp. 67–72. ISBN: 9781538694619. DOI: 10.1109/WHC.2019.8816173.
- [10] Peter F Edemekong et al. "Activities of Daily Living". In: *Encyclopedia of the Neurological Sciences* (Dec. 2017), pp. 47–48.
- [11] Catherine Elliott et al. "Lycra® arm splints improve movement fluency in children with cerebral palsy". In: *Gait & Posture* 33.2 (Feb. 2011), pp. 214–219. ISSN: 0966-6362. DOI: 10.1016/J.GAITPOST.2010.11.008.
- [12] Alberto J. Espay, Peter A. LeWitt, and Horacio Kaufmann. "Norepinephrine deficiency in Parkinson's disease: The case for noradrenergic enhancement". In: *Movement Disorders* 29.14 (Dec. 2014), pp. 1710–1719. ISSN: 1531-8257. DOI: 10.1002/MDS.26048. URL: <https://onlinelibrary.wiley.com/doi/full/10.1002/mds.26048> <https://onlinelibrary.wiley.com/doi/abs/10.1002/mds.26048> <https://movementdisorders.onlinelibrary.wiley.com/doi/10.1002/mds.26048>.
- [13] Tamar Flash and Neville Hogans. "The Coordination of Arm Movements: An Experimentally Confirmed Mathematical Model". In: *The Journal of Neuroscience* 5.7 (1985), pp. 1688–1703.
- [14] Andreas Gebhardt et al. "Poor dopaminergic response of impaired dexterity in Parkinson's disease: Bradykinesia or limb kinetic apraxia?" In: *Movement Disorders* 23.12 (Sept. 2008), pp. 1701–1706. DOI: 10.1002/MDS.22199.



- [15] *Haptic Energy Consumption*. Tech. rep. Dallas, Texas: Texas Instruments, 2022. URL: [www.ti.com](http://www.ti.com).
- [16] *Haptic feedback (haptics) - What is it?* URL: [https://teslasuit.io/blog/haptic\\_feedback/](https://teslasuit.io/blog/haptic_feedback/).
- [17] Christian Hatzfeld and Thorsten A Kern Editors. *Springer Series on Touch and Haptic Systems Engineering Haptic Devices A Beginner's Guide Second Edition*. Tech. rep. URL: <http://www.springer.com/series/8786>.
- [18] Colin V. Keef et al. "Virtual Texture Generated Using Elastomeric Conductive Block Copolymer in a Wireless Multimodal Haptic Glove". In: *Advanced Intelligent Systems 2.4* (Apr. 2020), p. 2000018. ISSN: 2640-4567. DOI: 10.1002/aisy.202000018.
- [19] S J Lederman and R L Klatzky. "Haptic perception". In: *Attention, Perception & Psychophysics 71.7* (2009), pp. 1439–1459.
- [20] Takao Nozaki et al. "Effect of Subthalamic Deep Brain Stimulation on Upper Limb Dexterity in Patients with Parkinson Disease". In: *World Neurosurgery 115* (July 2018), e206–e217. ISSN: 1878-8750. DOI: 10.1016/J.WNEU.2018.04.014.
- [21] JA Opara et al. "Quality of life in Parkinson's Disease". In: *Journal of Medicine and Life 5.4* (2012), p. 375.
- [22] Claudio Pacchierotti et al. "Wearable haptic systems for the fingertip and the hand: Taxonomy, review, and perspectives". In: *IEEE Transactions on Haptics 10.4* (2017), pp. 580–600. DOI: 10.1109/TOH.2017.2689006.
- [23] Ying Chun Pan et al. "Feasibility of using acceleration-derived jerk to quantify bimanual arm use". In: *Journal of NeuroEngineering and Rehabilitation 17.1* (Mar. 2020). ISSN: 17430003. DOI: 10.1186/S12984-020-0653-2.
- [24] *Parkinson's Disease | National Institute on Aging*. URL: <https://www.nia.nih.gov/health/parkinsons-disease>.
- [25] Claudia Ramaker et al. "Systematic evaluation of rating scales for impairment and disability in Parkinson's disease". In: *Movement Disorders 17.5* (Sept. 2002), pp. 867–876. ISSN: 1531-8257. DOI: 10.1002/MDS.10248. URL: <https://onlinelibrary.wiley.com/doi/full/10.1002/mds.10248> <https://onlinelibrary.wiley.com/doi/abs/10.1002/mds.10248> <https://movementdisorders.onlinelibrary.wiley.com/doi/10.1002/mds.10248>.
- [26] Luis F. Schettino et al. "Hand preshaping in Parkinson's disease: Effects of visual feedback and medication state". In: *Experimental Brain Research 168.1-2* (2006), pp. 186–202. ISSN: 00144819. DOI: 10.1007/s00221-005-0080-4.
- [27] Anette Schrag, Marjan Jahanshahi, and Niall Quinn. "What contributes to quality of life in patients with Parkinson's disease?" In: *Journal of Neurology, Neurosurgery & Psychiatry 69.3* (Sept. 2000), pp. 308–312. ISSN: 0022-3050. DOI: 10.1136/JNRP.69.3.308. URL: <https://jnnp.bmj.com/content/69/3/308> <https://jnnp.bmj.com/content/69/3/308.abstract>.
- [28] *Unified Parkinson's Disease Rating Scale – Meetinstrumenten in de zorg*. URL: <https://meetinstrumentenzorg.nl/instrumenten/unified-parkinsons-disease-rating-scale/>.
- [29] *Vibration Motors - ERMs and LRAs - Precision Microdrives*. URL: <https://www.precisionmicrodrives.com/vibration-motors-erms-and-lras>.
- [30] Vibol Yem and Hiroyuki Kajimoto. "Wearable tactile device using mechanical and electrical stimulation for fingertip interaction with virtual world". In: *Proceedings - IEEE Virtual Reality* (Apr. 2017), pp. 99–104. DOI: 10.1109/VR.2017.7892236.



# Scripts

**Listing A.1:** Actuator and accelerometer parameters

```
1 #include <Wire.h>
2 #include "DRV2605L.h"
3 #include <Adafruit_LIS3DH.h>
4 #include <Adafruit_Sensor.h>
5
6 // -----Vibrator and Accelerometer parameters
7 // ----- //
8 const int amnt_vibrators = 5;
9 DRV2605L vib;
10 Adafruit_LIS3DH lis = Adafruit_LIS3DH(); //Default constructor is I2C, addr 0x19.
11
12 const int buttonPin = 2;
13 const int ledPin = 13;
14 int buttonState = 0;
15
16 boolean autoCal = true; //Autocalibration on or off
17 int voltage_rms = 1.2; //rms voltage
18 int voltage_peak = 1.9; //peak voltage
19 int frequency = 240; //rated resonance frequency
20 uint8_t vibMode = DRV2605_MODE_REALTIME; //mode selection
21 int startActuation = 0; //start signal
22
23 const int maxStorage = 500;
24 int state[maxStorage];
25 int microsecond[maxStorage];
26 float sensor[maxStorage];
27
28 float dataX;
29 float dataY;
30 float dataZ;
```

**Listing A.2:** Accelerometer functions

```
1 void initAccelerometer()
2 {
3   busSelect(vib_bus_accelerometer);
4   while (!Serial) delay(10); // will pause until serial console opens
5   Serial.println("LIS3DH test!");
6
7   if (!lis.begin(0x18)) { // change this to 0x19 for alternative i2c address
8     Serial.println("Couldnt start");
9     while (1) yield();
10  }
11
12  Serial.println("LIS3DH found!");
13  lis.setRange(LIS3DH_RANGE_2_G); // set the range to 2, 4, 8 or 16 G
14 }
```

```

15 Serial.print("Range = "); Serial.print(2 << lis.getRange());
16 Serial.println("G");
17
18 lis.setDataRate(LIS3DH_DATARATE_400_HZ);
19 }
20
21
22 void handleAccelerometer()
23 {
24     //Get all parameters
25     busSelect(vib_bus_accelerometer);
26     lis.read();
27     dataX = lis.x;
28     dataY = lis.y;
29     dataZ = lis.z;
30 }
31
32
33 void visualizeData(float dataX, float dataY, float dataZ, Adafruit_LIS3DH &lis)
34 {
35     // Display the results (acceleration is measured in m/s^2)
36     sensors_event_t event;
37     lis.getEvent(&event);
38     Serial.print(startActuation);
39     Serial.print(event.acceleration.z);
40
41     // Serial.print(":\\tX:"); Serial.print(event.acceleration.x);
42     // Serial.print(":\\tY:"); Serial.print(event.acceleration.y);
43     // Serial.print(":");
44     // Serial.println(" m/s^2 ");
45 }

```

Listing A.3: Serial Communication

```

1
2 void recvWithStartEndMarkers() { // Receive incoming serial input from start to end
   marker
3   static boolean recvInProgress = false;
4   static byte ndx = 0;
5   char startMarker = '<';
6   char endMarker = '>';
7   char rc;
8
9   while (Serial.available() > 0 && newData == false) {
10      rc = Serial.read();
11
12      if (recvInProgress == true) {
13          if (rc != endMarker) {
14              receivedChars[ndx] = rc;
15              ndx++;
16              if (ndx >= numChars) {
17                  ndx = numChars - 1;
18              }
19          }
20          else {
21              receivedChars[ndx] = '\\0'; // terminate the string
22              recvInProgress = false;
23              ndx = 0;
24              newData = true;
25          }
26      }
27
28      else if (rc == startMarker) {
29          recvInProgress = true;
30      }
31  }
32 }
33
34 void parseData() {
35     //Serial.println(receivedChars);
36     String receivedString = String(receivedChars);

```

```

37
38 if (receivedString == "start") {
39     startEvent = true;
40     Serial.println("New Grasp!");
41 }
42 else if (receivedString == "stop") {
43     startEvent = false;
44     Serial.println("Stop Grasp!");
45 }
46
47 else {
48     // split the incoming data into parts and convert to strings and integers
49     strcpy(tempChars, receivedChars);
50     char * strtokIndx;           // this is used by strtok() as an index
51
52     strtokIndx = strtok(tempChars, ","); // get the first part - the string
53     strcpy(vibLocation, strtokIndx);    // copy it to VibLocation
54
55     strtokIndx = strtok(NULL, ",");     // this continues where the previous call left
56     off                                           // off
57     vibAction = *strtokIndx;                // convert this part to a char
58
59     strtokIndx = strtok(NULL, ",");
60     vibEffectInt = atoi(strtokIndx);       // convert this part to an integer
61
62     parsed = true;                         // parsing has been done
63
64
65
66 //Serial.print("Location ");
67 //Serial.println(String(vibLocation));
68 //Serial.print("Action");
69 //Serial.println(String(vibAction));
70 //Serial.print("Effect");
71 //Serial.println(vibEffectInt);
72
73 if (*vibLocation == VIB_LCTN_THUMB) { // Check Location Input
74     vibLocationString = "Thumb";
75     vibLocationInt = vib_bus_thumb;
76     //Serial.println("This is correct");
77 }
78 else if (*vibLocation == VIB_LCTN_INDEX) {
79     vibLocationString = "Index";
80     vibLocationInt = vib_bus_index;
81 }
82 else if (*vibLocation == VIB_LCTN_MIDDLE) {
83     vibLocationString = "Middle";
84     vibLocationInt = vib_bus_middle;
85 }
86 else if (*vibLocation == VIB_LCTN_RING) {
87     vibLocationString = "Ring";
88     vibLocationInt = vib_bus_ring;
89 }
90 else if (*vibLocation == VIB_LCTN_PINKY) {
91     vibLocationString = "Pinky";
92     vibLocationInt = vib_bus_pinky;
93 }
94 else if (*vibLocation == VIB_LCTN_HANDPALM) {
95     vibLocationString = "Hand Palm";
96     vibLocationInt = vib_bus_handpalm;
97 }
98 else {
99     Serial.println("You have chosen an invalid vibrator");
100    vibLocationString = "Nothing";
101    parsed = false; // wrong input
102    // create code to change to correct bus
103 }
104
105 if (vibAction == VIB_GO) { // Check Action Input
106     vibActionString = "Go";

```

```

107     vibActionInt = 1;
108     //Serial.println("This is correct");
109 }
110 else if (vibAction == VIB_STOP) {
111     vibActionString = "Stop";
112     vibActionInt = 0;
113 }
114 else {
115     Serial.println("You have chosen an invalid action");
116     vibActionString = "Nothing";
117     parsed = false; // wrong input
118 }
119
120 if (vibEffectInt > 255) { // Check Effect Input
121     Serial.println("You have chosen an invalid value");
122     parsed = false;
123 }
124 else {
125     vibEffectInt = map(vibEffectInt, 0, 9, 0, 255);
126 }
127 }
128 }
129
130 void showParsedData()
131 { // Show the incoming data
132     Serial.print("Location ");
133     Serial.println(vibLocationString);
134     Serial.print("Action");
135     Serial.println(vibActionString);
136     Serial.print("Effect");
137     Serial.println(vibEffectInt);
138 }
139 }
140
141 void handleParseData()
142 { // Use the data to perform the necessary actions
143
144     busSelect(vibLocationInt);
145     switch (vibActionInt) {
146
147         case 0: // Stop vibrating
148
149             if (vibMode == DRV2605_MODE_REALTIME) {
150                 //Serial.println("Stop");
151                 vib[vibLocationInt].setRealtimeValue(0x00);
152             }
153             else if (vibMode == DRV2605_MODE_INTRIG) {
154
155                 vib[vibLocationInt].stop();
156             }
157
158             else {
159                 Serial.println("Choose another mode for now...");
160             }
161             break;
162
163         case 1: // Start Vibrating
164
165             busSelect(vibLocationInt);
166             if (vibMode == DRV2605_MODE_REALTIME) {
167                 for (int i = 0; i < amnt_vibrators; i++) {
168                     vib[i].setMode(DRV2605_MODE_REALTIME); // set to selected mode
169                 }
170                 if (*vibLocation == BAD_GRAB) {
171                     //Serial.println("Im here");
172                     for (int i = 0; i < amnt_vibrators; i++) {
173                         busSelect(i);
174                         vib[i].setRealtimeValue(vibEffectInt);
175                     }
176                 }
177             }

```

```

178     }
179
180     delay(60);
181     for (int i = 0; i < amnt_vibrators; i++) {
182         busSelect(i);
183         vib[i].setRealtimeValue(0x00);
184     }
185     delay(60);
186     for (int i = 0; i < amnt_vibrators; i++) {
187         busSelect(i);
188         vib[i].setRealtimeValue(vibEffectInt);
189     }
190     delay(60);
191     for (int i = 0; i < amnt_vibrators; i++) {
192         busSelect(i);
193         vib[i].setRealtimeValue(0x00);
194     }
195     delay(60);
196     for (int i = 0; i < amnt_vibrators; i++) {
197         busSelect(i);
198         vib[i].setRealtimeValue(vibEffectInt);
199     }
200     delay(60);
201     for (int i = 0; i < amnt_vibrators; i++) {
202         busSelect(i);
203         vib[i].setRealtimeValue(0x00);
204     }
205
206 }
207
208 else {
209     //Serial.println("Im her");
210     //vib[vibLocationInt].setRealtimeValue(vibEffectInt);
211     //delay(200);
212     //vib[vibLocationInt].setRealtimeValue(0x00);
213     for (int i = 0; i < amnt_vibrators; i++) {
214         busSelect(i);
215         vib[i].setRealtimeValue(vibEffectInt);
216     }
217
218     delay(400);
219     for (int i = 0; i < amnt_vibrators; i++) {
220         busSelect(i);
221         vib[i].setRealtimeValue(0x00);
222     }
223 }
224 }
225
226 else if (vibMode == DRV2605_MODE_INTRIG) {
227     Serial.println("Start Vibrating Internal");
228     vib[vibLocationInt].setWaveform(0, 48);
229     vib[vibLocationInt].setWaveform(1, 0);
230     vib[vibLocationInt].go();
231     //delay(500);
232
233 }
234
235 else {
236     Serial.println("Choose another mode for now...");
237
238 }
239 break;
240
241 default:
242     break;
243 }
244 }

```

Listing A.4: Vibrator parameters and initialization

```
1 //-----Vibrator initialization ----- //
```

```

2 void initVibrator(uint8_t vibmode)
3 {
4   Serial.println("Begin Initialization, set mode is:");
5   String modeString = String(vibmode);
6   //Serial.println(modeString);
7
8   for (int i = 0; i < amnt_vibrators; i++)
9   {
10    busSelect(i);           // select bus
11    Serial.println("Hello");
12    vib[i].begin();        // set-up the vibrators to standard configuration
13
14    if (vib[i].begin()) {
15      Serial.print( "Vibrator: "); Serial.print(i); Serial.print( "on bus: "); Serial.println
16        (i);
17
18      if (autoCal == true) {
19        vib[i].autoCalibrationLRA(voltage_rms, voltage_peak, frequency);           // start
20          auto calibration process
21
22        if (vib[i].autoCalibrationLRA(voltage_rms, voltage_peak, frequency)) {
23          Serial.println((String) "Calibration of vibrator " + i + " Complete");
24          // check if auto calibration worked
25        }
26        else {
27          Serial.println((String) "Calibration Failed, is tried again");
28          vib[i].autoCalibrationLRA(voltage_rms, voltage_peak, frequency);           // start
29            auto calibration process
30          if (vib[i].autoCalibrationLRA(voltage_rms, voltage_peak, frequency)) {
31            Serial.println((String) "Calibration of vibrator " + i + " Complete");
32            // check if auto calibration worked
33          }
34          else {
35            Serial.println((String) "Calibration Failed again");
36          }
37        }
38      }
39      else {
40        Serial.println("No autocalibration"); // check if all vibrators are properly
41          connected
42      }
43      vib[i].useLRA();           // set to LRA mode
44      vib[i].selectLibrary(6);   // select LRA library
45      vib[i].setMode(vibmode);   // set to selected mode
46      //Serial.println(vib[i].readRegister8(DRV2605_REG_MODE));
47    }
48    else {
49      Serial.println("No Actuator attached");
50    }
51  }
52 }
53
54 // ----- Set-up -----//
55 void setup()
56 {
57   Wire.begin();
58   Serial.begin(9600);
59   delay(500); //allow time for communication start
60   Serial.println("Hello");
61   //initAccelerometer();
62   initVibrator(vibMode);
63   Serial.println("Setup finished");
64
65   // receive input //
66   //Serial.println("This demo expects 3 pieces of data - which actuator, 's' for stop and 'g'
67     for go, and an effect value");
68   // Serial.println("Enter data in this style <t, g, 13> ");
69   // Serial.println();

```

```

66 }
67
68
69 //===== LOOPP =====//
70
71 void loop()
72 {
73   recvWithStartEndMarkers();
74   if (newData == true) {
75     startActuation = 0;
76     //Serial.println("New data!");
77     parseData();
78     //showParsedData();
79     handleParseData();
80     newData = false;
81   }
82 }
83 }
84
85 //-----Accelerometer ----- //
86
87 void initAccelerometer()
88 {
89   busSelect(vib_bus_accelerometer);
90   while (!Serial) delay(10); // will pause Zero, Leonardo, etc until serial console opens
91   Serial.println("LIS3DH test!");
92
93   if (!lis.begin(0x18)) { // change this to 0x19 for alternative i2c address
94     Serial.println("Couldnt start");
95     while (1) yield();
96   }
97   Serial.println("LIS3DH found!");
98
99   lis.setRange(LIS3DH_RANGE_2_G); // 2, 4, 8 or 16 G!
100
101   Serial.print("Range = "); Serial.print(2 << lis.getRange());
102   Serial.println("G");
103
104   lis.setDataRate(LIS3DH_DATARATE_400_HZ);
105 }
106
107 void handleAccelerometer()
108 {
109   //Get all parameters
110   busSelect(vib_bus_accelerometer);
111   lis.read();
112   dataX = lis.x;
113   dataY = lis.y;
114   dataZ = lis.z;
115 }
116
117
118 void visualizeData(float dataX, float dataY, float dataZ, Adafruit_LIS3DH &lis, bool
    visualize)
119 {
120
121   /* Display the results (acceleration is measured in m/s^2) */
122   if (visualize) {
123     sensors_event_t event;
124     lis.getEvent(&event);
125     //Serial.print(startActuation);
126     Serial.print("\tX:"); Serial.print(event.acceleration.x);
127     Serial.print("\tY:"); Serial.print(event.acceleration.y);
128     Serial.print("\tZ:"); Serial.println(event.acceleration.z);
129   }
130   else if (!visualize) {
131     return;
132   }
133 }
134
135 }

```



Listing A.5: Grasp Check

```

1
2 public class GraspCheck : MonoBehaviour
3 {
4     MovableCollisionHandler grab;
5     HandPosition handposition;
6
7     public Vector3 rawThumbPos;
8     public Vector3 rawIndexPos;
9     public Vector3 rawMiddlePos;
10    public Vector3 rawRingPos;
11    public Vector3 rawPinkyPos;
12
13    public Vector3 rawThumbPosLeft;
14    public Vector3 rawIndexPosLeft;
15    public Vector3 rawMiddlePosLeft;
16    public Vector3 rawRingPosLeft;
17    public Vector3 rawPinkyPosLeft;
18
19    public bool badGrab = false;
20    public bool grabCheck = false;
21    public float distanceFingersRight;
22    public float distanceFingersLeft;
23
24    public float[] distanceFingersRightArray;
25    public float[] distanceFingersLeftArray;
26
27    public Vector3 rawThumbPosLeftDebug;
28
29    float maxDistanceFingers = 0.03f;
30    bool counter = false;
31    //bool haptic = false;
32    public bool handCheckRight = false;
33    public bool handCheckLeft = false;
34    public float distLeft;
35    public float distRight;
36
37    EggInstantiate egg;
38    PointManagerReal pointsManager;
39    SerialManager serial;
40    CSVWrite csv;
41    HapticEgg haptic;
42    GameObject RightHand;
43    GameObject LeftHand;
44
45
46
47
48
49
50 // Start is called before the first frame update
51 void Start()
52 {
53     //colourRenderer = this.gameObject.GetComponent<Renderer>();
54     csv = GameObject.Find("CSVWrite").GetComponent<CSVWrite>();
55     grab = gameObject.GetComponent<MovableCollisionHandler>();
56     pointsManager = GameObject.Find("PointsManager").GetComponent<PointManagerReal>();
57     serial = GameObject.Find("SerialManager").GetComponent<SerialManager>();
58     handposition = GameObject.Find("HandPosition").GetComponent<HandPosition>();
59     //haptic = this.gameObject.GetComponent<HapticEgg>();
60
61     //RightHand = GameObject.Find("finger_index_end_r");
62     //LeftHand = GameObject.Find("finger_index_end_l");
63
64     grab.onGrab.AddListener(Haptic);
65
66 }
67
68 // Update is called once per frame
69 void Update()

```

```

70 {
71     rawThumbPosLeftDebug = new Vector3 (VRfree.HandBoneTransforms.RightHand.
        GetBoneWorldPosition(VRfree.HandBoneTransforms.BoneIndex.ThumbEnd, VRfree.
        VRfreeAPI.GetHandData(true)).x, VRfree.HandBoneTransforms.RightHand.
        GetBoneWorldPosition(VRfree.HandBoneTransforms.BoneIndex.ThumbEnd, VRfree.
        VRfreeAPI.GetHandData(true)).y, VRfree.HandBoneTransforms.RightHand.
        GetBoneWorldPosition(VRfree.HandBoneTransforms.BoneIndex.ThumbEnd, VRfree.
        VRfreeAPI.GetHandData(true)).z) ;
72     GetAndWritePosition();
73     GetFingerDistanceRight();
74     GetFingerDistanceLeft();
75     //GetFingerDistanceRightArray();
76     // GetFingerDistanceLeftArray();
77
78     grabCheck = grab.isGrabbed;
79     // handCheckRight = haptic.isRightHand();
80     //handCheckLeft = haptic.isLeftHand();
81
82     distRight = Vector3.Distance(gameObject.transform.position, GameObject.Find("root_r")
        .transform.position);
83     distLeft = Vector3.Distance(gameObject.transform.position, GameObject.Find("root_l").
        transform.position);
84
85     if (grabCheck)
86     {
87
88
89         counter = true;
90         StartCoroutine("HoldCounter");
91
92         if (distRight < distLeft) //grab is on right hand
93         {
94             handCheckRight = true;
95             handCheckLeft = false;
96             for (int i = 0; i < 4; i++)
97             {
98                 if (distanceFingersRightArray[i] < 0.03) //< HandPosition.
                    distanceFingersStartRight[i] * 0.2)
99                 {
100                     Debug.Log("Bad Grab!"); // do things
101                     badGrab = true;
102                 }
103                 /*if (distanceFingersRight <= maxDistanceFingers)
104                 {
105                     Debug.Log("Bad Grab!"); // do things
106                     badGrab = true;
107                 }*/
108             }
109         }
110         else if (distRight >= distLeft) //grab is with left hand
111         {
112             handCheckRight = false;
113             handCheckLeft = true;
114             for (int i = 0; i < 4; i++)
115             {
116                 if (distanceFingersLeftArray[i] < 0.025) //< HandPosition.
                    distanceFingersStartLeft[i] * 0.2)
117                 {
118                     Debug.Log("Bad Grab!"); // do things
119                     badGrab = true;
120                 }
121                 /* if (distanceFingersLeft <= maxDistanceFingers)
122                 {
123                     Debug.Log("Bad Grab!"); // do things
124                     badGrab = true;
125                 }*/
126             }
127         }
128
129
130     }

```

```
131     /*if (distanceFingers <= maxDistanceFingers)
132     {
133         Debug.Log("Bad Grab!"); // do things
134         badGrab = true;
135     }
136     else
137     {
138         badGrab = false;
139     }*/
140 }
141
142 else
143 {
144     StopCoroutine("HoldCounter");
145 }
146
147 if (this.transform.position.y <= 0.6) // check if egg has fallen of the table
148 {
149     Debug.Log("Off the table!"); // do things
150     badGrab = true;
151 }
152
153
154 if (badGrab == true)
155 {
156
157     gameObject.GetComponent<Renderer>().material.color = new Color(255, 0, 0);
158
159
160     StartCoroutine("DelayDestroy");
161
162     //StartCoroutine("HapticDelayWithDestroy");
163     badGrab = false;
164
165 }
166 }
167
168 private float calcFingerDistanceAvg(Vector3 thumbPos, Vector3 indexPos, Vector3
169     middlePos, Vector3 ringPos, Vector3 pinkyPos)
170 {
171     float distanceTI = Vector3.Distance(thumbPos, indexPos);
172     float distanceTM = Vector3.Distance(thumbPos, middlePos);
173     float distanceTR = Vector3.Distance(thumbPos, ringPos);
174     float distanceTP = Vector3.Distance(thumbPos, pinkyPos);
175
176     float avgDistance = (distanceTI+distanceTM+distanceTP+distanceTR)/4;
177
178     return avgDistance;
179 }
180
181 private float[] calcFingerDistanceArray(Vector3 thumbPos, Vector3 indexPos, Vector3
182     middlePos, Vector3 ringPos, Vector3 pinkyPos)
183 {
184     float distanceTI = Vector3.Distance(thumbPos, indexPos);
185     float distanceTM = Vector3.Distance(thumbPos, middlePos);
186     float distanceTR = Vector3.Distance(thumbPos, ringPos);
187     float distanceTP = Vector3.Distance(thumbPos, pinkyPos);
188
189     float[] distanceArray = { distanceTI, distanceTM, distanceTR, distanceTP};
190
191     return distanceArray;
192 }
193
194 private IEnumerator HoldCounter() // Coroutine that counts the length of the hold. After 3
195     s the hold is completed, a point is added, a new egg is instantiated and this current
196     egg disappears at a random new location
197 {
198     if (counter == true)
199     {
```

```

198     yield return new WaitForSeconds(3); // wait three seconds
199
200
201
202     //Call SetColor using the shader property name "_Color" and setting the color to
        red
203     gameObject.GetComponent<Renderer>().material.color = new Color(144, 238, 144);
204     //colourRenderer.material.SetColor("_Color", Color.red);
        // do things
205     Debug.Log("Congratualtions");
206
207     counter = false;
208     StartCoroutine("DelayDestroyGood");
209
210 }
211
212
213 }
214
215 private IEnumerator DelayDestroy() // Coroutine that couds the length of the hold. After
        3s the hold is completed, a point is added, a new egg is instantiated and this
        current egg disapears at a random new location
216 {
217     yield return new WaitForSeconds(0.2f); // wait three seconds
218     serial.sendHapticFeedback('b', 'g');
219     pointsManager.UpdateScoreBad();
220     Destroy(this.gameObject);
221
222 }
223
224 private IEnumerator DelayDestroyGood() // Coroutine that couds the length of the hold.
        After 3s the hold is completed, a point is added, a new egg is instantiated and this
        current egg disapears at a random new location
225 {
226     yield return new WaitForSeconds(0.2f); // wait three seconds
227
228
229     pointsManager.UpdateScoreGood();
230     Destroy(this.gameObject);
231
232 }
233
234 private void GetAndWritePosition()
235 {
236     float rawWristPosx = GameObject.Find("root_r").transform.position.x;
237     float rawWristPosy = GameObject.Find("root_r").transform.position.y;
238     float rawWristPosz = GameObject.Find("root_r").transform.position.z;
239
240     float rawWristPosxLeft = GameObject.Find("root_l").transform.position.x;
241     float rawWristPosyLeft = GameObject.Find("root_l").transform.position.y;
242     float rawWristPoszLeft = GameObject.Find("root_l").transform.position.z;
243     csv.WritePositionToCSV(rawWristPosx, rawWristPosy, rawWristPosz, rawWristPosxLeft,
        rawWristPosyLeft, rawWristPoszLeft );
244 }
245
246 private void GetFingerDistanceRight()
247 {
248     rawThumbPos = new Vector3 (Vrfree.HandBoneTransforms.RightHand.GetBoneWorldPosition(
        Vrfree.HandBoneTransforms.BoneIndex.ThumbEnd, Vrfree.VrfreeAPI.GetHandData(true))
        .x, Vrfree.HandBoneTransforms.RightHand.GetBoneWorldPosition(Vrfree.
        HandBoneTransforms.BoneIndex.ThumbEnd, Vrfree.VrfreeAPI.GetHandData(true)).y,
        Vrfree.HandBoneTransforms.RightHand.GetBoneWorldPosition(Vrfree.
        HandBoneTransforms.BoneIndex.ThumbEnd, Vrfree.VrfreeAPI.GetHandData(true)).z) ;
249     rawIndexPos = new Vector3(Vrfree.HandBoneTransforms.RightHand.GetBoneWorldPosition(
        Vrfree.HandBoneTransforms.BoneIndex.IndexFingerEnd, Vrfree.VrfreeAPI.GetHandData(
        true)).x, Vrfree.HandBoneTransforms.RightHand.GetBoneWorldPosition(Vrfree.
        HandBoneTransforms.BoneIndex.IndexFingerEnd, Vrfree.VrfreeAPI.GetHandData(true)).
        y, Vrfree.HandBoneTransforms.RightHand.GetBoneWorldPosition(Vrfree.
        HandBoneTransforms.BoneIndex.IndexFingerEnd, Vrfree.VrfreeAPI.GetHandData(true)).
        z);
250     rawMiddlePos = new Vector3(Vrfree.HandBoneTransforms.RightHand.GetBoneWorldPosition(

```

```

    VRfree.HandBoneTransforms.BoneIndex.MiddleFingerEnd, VRfree.VRfreeAPI.GetHandData
    (true)).x, VRfree.HandBoneTransforms.RightHand.GetBoneWorldPosition(VRfree.
    HandBoneTransforms.BoneIndex.MiddleFingerEnd, VRfree.VRfreeAPI.GetHandData(true))
    .y, VRfree.HandBoneTransforms.RightHand.GetBoneWorldPosition(VRfree.
    HandBoneTransforms.BoneIndex.MiddleFingerEnd, VRfree.VRfreeAPI.GetHandData(true))
    .z);
251 rawRingPos = new Vector3(VRfree.HandBoneTransforms.RightHand.GetBoneWorldPosition(
    VRfree.HandBoneTransforms.BoneIndex.RingFingerEnd, VRfree.VRfreeAPI.GetHandData(
    true)).x, VRfree.HandBoneTransforms.RightHand.GetBoneWorldPosition(VRfree.
    HandBoneTransforms.BoneIndex.RingFingerEnd, VRfree.VRfreeAPI.GetHandData(true)).y
    , VRfree.HandBoneTransforms.RightHand.GetBoneWorldPosition(VRfree.
    HandBoneTransforms.BoneIndex.RingFingerEnd, VRfree.VRfreeAPI.GetHandData(true)).z
    );
252 rawPinkyPos = new Vector3(VRfree.HandBoneTransforms.RightHand.GetBoneWorldPosition(
    VRfree.HandBoneTransforms.BoneIndex.PinkyFingerEnd, VRfree.VRfreeAPI.GetHandData(
    true)).x, VRfree.HandBoneTransforms.RightHand.GetBoneWorldPosition(VRfree.
    HandBoneTransforms.BoneIndex.PinkyFingerEnd, VRfree.VRfreeAPI.GetHandData(true)).
    y, VRfree.HandBoneTransforms.RightHand.GetBoneWorldPosition(VRfree.
    HandBoneTransforms.BoneIndex.PinkyFingerEnd, VRfree.VRfreeAPI.GetHandData(true)).
    z);

253
254
255 distanceFingersRight = calcFingerDistanceAvg(rawThumbPos, rawIndexPos, rawMiddlePos,
    rawRingPos, rawPinkyPos);
256 distanceFingersRightArray = calcFingerDistanceArray(rawThumbPos, rawIndexPos,
    rawMiddlePos, rawRingPos, rawPinkyPos);
257 csv.WritePositionFingerToCSV(rawThumbPos, rawIndexPos, rawMiddlePos, rawRingPos,
    rawPinkyPos);
258 }
259
260
261 private void GetFingerDistanceLeft()
262 {
263     rawThumbPosLeft = new Vector3(VRfree.HandBoneTransforms.LeftHand.GetBoneWorldPosition
    (VRfree.HandBoneTransforms.BoneIndex.ThumbEnd, VRfree.VRfreeAPI.GetHandData(false
    )).x, VRfree.HandBoneTransforms.LeftHand.GetBoneWorldPosition(VRfree.
    HandBoneTransforms.BoneIndex.ThumbEnd, VRfree.VRfreeAPI.GetHandData(false)).y,
    VRfree.HandBoneTransforms.LeftHand.GetBoneWorldPosition(VRfree.HandBoneTransforms
    .BoneIndex.ThumbEnd, VRfree.VRfreeAPI.GetHandData(false)).z);
264 rawIndexPosLeft = new Vector3(VRfree.HandBoneTransforms.LeftHand.GetBoneWorldPosition
    (VRfree.HandBoneTransforms.BoneIndex.IndexFingerEnd, VRfree.VRfreeAPI.GetHandData
    (false)).x, VRfree.HandBoneTransforms.LeftHand.GetBoneWorldPosition(VRfree.
    HandBoneTransforms.BoneIndex.IndexFingerEnd, VRfree.VRfreeAPI.GetHandData(false))
    .y, VRfree.HandBoneTransforms.LeftHand.GetBoneWorldPosition(VRfree.
    HandBoneTransforms.BoneIndex.IndexFingerEnd, VRfree.VRfreeAPI.GetHandData(false))
    .z);
265 rawMiddlePosLeft = new Vector3(VRfree.HandBoneTransforms.LeftHand.
    GetBoneWorldPosition(VRfree.HandBoneTransforms.BoneIndex.MiddleFingerEnd, VRfree.
    VRfreeAPI.GetHandData(false)).x, VRfree.HandBoneTransforms.LeftHand.
    GetBoneWorldPosition(VRfree.HandBoneTransforms.BoneIndex.MiddleFingerEnd, VRfree.
    VRfreeAPI.GetHandData(false)).y, VRfree.HandBoneTransforms.LeftHand.
    GetBoneWorldPosition(VRfree.HandBoneTransforms.BoneIndex.MiddleFingerEnd, VRfree.
    VRfreeAPI.GetHandData(false)).z);
266 rawRingPosLeft = new Vector3(VRfree.HandBoneTransforms.LeftHand.GetBoneWorldPosition(
    VRfree.HandBoneTransforms.BoneIndex.RingFingerEnd, VRfree.VRfreeAPI.GetHandData(
    false)).x, VRfree.HandBoneTransforms.LeftHand.GetBoneWorldPosition(VRfree.
    HandBoneTransforms.BoneIndex.RingFingerEnd, VRfree.VRfreeAPI.GetHandData(false)).
    y, VRfree.HandBoneTransforms.LeftHand.GetBoneWorldPosition(VRfree.
    HandBoneTransforms.BoneIndex.RingFingerEnd, VRfree.VRfreeAPI.GetHandData(false)).
    z);
267 rawPinkyPosLeft = new Vector3(VRfree.HandBoneTransforms.LeftHand.GetBoneWorldPosition
    (VRfree.HandBoneTransforms.BoneIndex.PinkyFingerEnd, VRfree.VRfreeAPI.GetHandData
    (false)).x, VRfree.HandBoneTransforms.LeftHand.GetBoneWorldPosition(VRfree.
    HandBoneTransforms.BoneIndex.PinkyFingerEnd, VRfree.VRfreeAPI.GetHandData(false))
    .y, VRfree.HandBoneTransforms.LeftHand.GetBoneWorldPosition(VRfree.
    HandBoneTransforms.BoneIndex.PinkyFingerEnd, VRfree.VRfreeAPI.GetHandData(false))
    .z);
268 //rawPinkyPosLeft = new Vector3(VRfree.HandBoneTransforms.LeftHand.
    GetBoneWorldPosition(VRfree.HandBoneTransforms.BoneIndex.Thumb2, VRfree.VRfreeAPI
    .GetHandData(true)).x, VRfree.HandBoneTransforms.LeftHand.GetBoneWorldPosition(
    VRfree.HandBoneTransforms.BoneIndex.Thumb2, VRfree.VRfreeAPI.GetHandData(true)).y

```

```

    , VRfree.HandBoneTransforms.LeftHand.GetBoneWorldPosition(VRfree.
    HandBoneTransforms.BoneIndex.Thumb2, VRfree.VRfreeAPI.GetHandData(true)).z);
269
270 distanceFingersLeft = calcFingerDistanceAvg(rawThumbPosLeft, rawIndexPosLeft,
    rawMiddlePosLeft, rawRingPosLeft, rawPinkyPosLeft);
271 distanceFingersLeftArray = calcFingerDistanceArray(rawThumbPosLeft, rawIndexPosLeft,
    rawMiddlePosLeft, rawRingPosLeft, rawPinkyPosLeft);
272 }
273
274 void Haptic()
275 {
276     serial.sendHapticFeedback('i', 'g');
277 }
278 }

```

Listing A.6: Hand Position

```

1
2 public class HandPosition : MonoBehaviour
3 {
4     [ReadOnly]
5     public static Vector3 rawThumbPosStartRight;
6     public static Vector3 rawIndexPosStartRight;
7     public static Vector3 rawMiddlePosStartRight;
8     public static Vector3 rawRingPosStartRight;
9     public static Vector3 rawPinkyPosStartRight;
10
11     public static Vector3 rawThumbPosStartLeft;
12     public static Vector3 rawIndexPosStartLeft;
13     public static Vector3 rawMiddlePosStartLeft;
14     public static Vector3 rawRingPosStartLeft;
15     public static Vector3 rawPinkyPosStartLeft;
16
17     public static float[] distanceFingersStartRight;
18     public static float[] distanceFingersStartLeft;
19
20     // Start is called before the first frame update
21     void Start()
22     {
23         //extract handData from finger tips position and store in finger positions
24         rawThumbPosStartRight = new Vector3(VRfree.HandBoneTransforms.RightHand.
            GetBoneWorldPosition(VRfree.HandBoneTransforms.BoneIndex.ThumbEnd, VRfree.
            VRfreeAPI.GetHandData(true)).x, VRfree.HandBoneTransforms.RightHand.
            GetBoneWorldPosition(VRfree.HandBoneTransforms.BoneIndex.ThumbEnd, VRfree.
            VRfreeAPI.GetHandData(true)).y, VRfree.HandBoneTransforms.RightHand.
            GetBoneWorldPosition(VRfree.HandBoneTransforms.BoneIndex.ThumbEnd, VRfree.
            VRfreeAPI.GetHandData(true)).z);
25         rawIndexPosStartRight = new Vector3(VRfree.HandBoneTransforms.RightHand.
            GetBoneWorldPosition(VRfree.HandBoneTransforms.BoneIndex.IndexFingerEnd, VRfree.
            VRfreeAPI.GetHandData(true)).x, VRfree.HandBoneTransforms.RightHand.
            GetBoneWorldPosition(VRfree.HandBoneTransforms.BoneIndex.IndexFingerEnd, VRfree.
            VRfreeAPI.GetHandData(true)).y, VRfree.HandBoneTransforms.RightHand.
            GetBoneWorldPosition(VRfree.HandBoneTransforms.BoneIndex.IndexFingerEnd, VRfree.
            VRfreeAPI.GetHandData(true)).z);
26         rawMiddlePosStartRight = new Vector3(VRfree.HandBoneTransforms.RightHand.
            GetBoneWorldPosition(VRfree.HandBoneTransforms.BoneIndex.MiddleFingerEnd, VRfree.
            VRfreeAPI.GetHandData(true)).x, VRfree.HandBoneTransforms.RightHand.
            GetBoneWorldPosition(VRfree.HandBoneTransforms.BoneIndex.MiddleFingerEnd, VRfree.
            VRfreeAPI.GetHandData(true)).y, VRfree.HandBoneTransforms.RightHand.
            GetBoneWorldPosition(VRfree.HandBoneTransforms.BoneIndex.MiddleFingerEnd, VRfree.
            VRfreeAPI.GetHandData(true)).z);
27         rawRingPosStartRight = new Vector3(VRfree.HandBoneTransforms.RightHand.
            GetBoneWorldPosition(VRfree.HandBoneTransforms.BoneIndex.RingFingerEnd, VRfree.
            VRfreeAPI.GetHandData(true)).x, VRfree.HandBoneTransforms.RightHand.
            GetBoneWorldPosition(VRfree.HandBoneTransforms.BoneIndex.RingFingerEnd, VRfree.
            VRfreeAPI.GetHandData(true)).y, VRfree.HandBoneTransforms.RightHand.
            GetBoneWorldPosition(VRfree.HandBoneTransforms.BoneIndex.RingFingerEnd, VRfree.
            VRfreeAPI.GetHandData(true)).z);
28         rawPinkyPosStartRight = new Vector3(VRfree.HandBoneTransforms.RightHand.
            GetBoneWorldPosition(VRfree.HandBoneTransforms.BoneIndex.PinkyFingerEnd, VRfree.
            VRfreeAPI.GetHandData(true)).x, VRfree.HandBoneTransforms.RightHand.

```

```

    GetBoneWorldPosition(VRfree.HandBoneTransforms.BoneIndex.PinkyFingerEnd, VRfree.
    VRfreeAPI.GetHandData(true)).y, VRfree.HandBoneTransforms.RightHand.
    GetBoneWorldPosition(VRfree.HandBoneTransforms.BoneIndex.PinkyFingerEnd, VRfree.
    VRfreeAPI.GetHandData(true)).z);
29
30    rawThumbPosStartLeft = new Vector3(VRfree.HandBoneTransforms.LeftHand.
    GetBoneWorldPosition(VRfree.HandBoneTransforms.BoneIndex.ThumbEnd, VRfree.
    VRfreeAPI.GetHandData(true)).x, VRfree.HandBoneTransforms.LeftHand.
    GetBoneWorldPosition(VRfree.HandBoneTransforms.BoneIndex.ThumbEnd, VRfree.
    VRfreeAPI.GetHandData(true)).y, VRfree.HandBoneTransforms.LeftHand.
    GetBoneWorldPosition(VRfree.HandBoneTransforms.BoneIndex.ThumbEnd, VRfree.
    VRfreeAPI.GetHandData(true)).z);
31    rawIndexPosStartLeft = new Vector3(VRfree.HandBoneTransforms.LeftHand.
    GetBoneWorldPosition(VRfree.HandBoneTransforms.BoneIndex.IndexFingerEnd, VRfree.
    VRfreeAPI.GetHandData(true)).x, VRfree.HandBoneTransforms.LeftHand.
    GetBoneWorldPosition(VRfree.HandBoneTransforms.BoneIndex.IndexFingerEnd, VRfree.
    VRfreeAPI.GetHandData(true)).y, VRfree.HandBoneTransforms.LeftHand.
    GetBoneWorldPosition(VRfree.HandBoneTransforms.BoneIndex.IndexFingerEnd, VRfree.
    VRfreeAPI.GetHandData(true)).z);
32    rawMiddlePosStartLeft = new Vector3(VRfree.HandBoneTransforms.LeftHand.
    GetBoneWorldPosition(VRfree.HandBoneTransforms.BoneIndex.MiddleFingerEnd, VRfree.
    VRfreeAPI.GetHandData(true)).x, VRfree.HandBoneTransforms.LeftHand.
    GetBoneWorldPosition(VRfree.HandBoneTransforms.BoneIndex.MiddleFingerEnd, VRfree.
    VRfreeAPI.GetHandData(true)).y, VRfree.HandBoneTransforms.LeftHand.
    GetBoneWorldPosition(VRfree.HandBoneTransforms.BoneIndex.MiddleFingerEnd, VRfree.
    VRfreeAPI.GetHandData(true)).z);
33    rawRingPosStartLeft = new Vector3(VRfree.HandBoneTransforms.LeftHand.
    GetBoneWorldPosition(VRfree.HandBoneTransforms.BoneIndex.RingFingerEnd, VRfree.
    VRfreeAPI.GetHandData(true)).x, VRfree.HandBoneTransforms.LeftHand.
    GetBoneWorldPosition(VRfree.HandBoneTransforms.BoneIndex.RingFingerEnd, VRfree.
    VRfreeAPI.GetHandData(true)).y, VRfree.HandBoneTransforms.LeftHand.
    GetBoneWorldPosition(VRfree.HandBoneTransforms.BoneIndex.RingFingerEnd, VRfree.
    VRfreeAPI.GetHandData(true)).z);
34    rawPinkyPosStartLeft = new Vector3(VRfree.HandBoneTransforms.LeftHand.
    GetBoneWorldPosition(VRfree.HandBoneTransforms.BoneIndex.PinkyFingerEnd, VRfree.
    VRfreeAPI.GetHandData(true)).x, VRfree.HandBoneTransforms.LeftHand.
    GetBoneWorldPosition(VRfree.HandBoneTransforms.BoneIndex.PinkyFingerEnd, VRfree.
    VRfreeAPI.GetHandData(true)).y, VRfree.HandBoneTransforms.LeftHand.
    GetBoneWorldPosition(VRfree.HandBoneTransforms.BoneIndex.PinkyFingerEnd, VRfree.
    VRfreeAPI.GetHandData(true)).z);
35
36    distanceFingersStartRight = calcFingerDistanceArray(rawThumbPosStartRight,
    rawIndexPosStartRight, rawMiddlePosStartRight, rawRingPosStartRight,
    rawPinkyPosStartRight);
37    distanceFingersStartLeft = calcFingerDistanceArray(rawThumbPosStartLeft,
    rawIndexPosStartLeft, rawMiddlePosStartLeft, rawRingPosStartLeft,
    rawPinkyPosStartLeft);
38    Debug.Log(distanceFingersStartLeft[1]);
39    Debug.Log(distanceFingersStartRight[1]);
40
41 }
42
43 private float[] calcFingerDistanceArray(Vector3 thumbPos, Vector3 indexPos, Vector3
    middlePos, Vector3 ringPos, Vector3 pinkyPos)
44 {
45     float distanceTI = Vector3.Distance(thumbPos, indexPos);
46     float distanceTM = Vector3.Distance(thumbPos, middlePos);
47     float distanceTR = Vector3.Distance(thumbPos, ringPos);
48     float distanceTP = Vector3.Distance(thumbPos, pinkyPos);
49
50     float[] distanceArray = { distanceTI, distanceTM, distanceTR, distanceTP };
51
52     return distanceArray;
53 }
54 }

```

Listing A.7: Haptic Egg

```

1 // Attach this script to the collider of the finger tips
2
3 using System.Collections;

```

```

4 using System.Collections.Generic;
5 using UnityEngine;
6 using VRfreePluginUnity;
7
8 public class HapticEgg : MonoBehaviour
9 {
10     // SerialManager haptic;
11     // public char location;
12     MovableCollisionHandler grab;
13
14
15     public bool RightHand;
16     public bool LeftHand;
17
18     // Start is called before the first frame update
19     void Start()
20     {
21
22         grab = this.gameObject.GetComponent<MovableCollisionHandler>();
23
24     }
25
26     // Update is called once per frame
27     private void OnCollisionEnter(Collision collision)
28     {
29         Debug.Log("Hello");
30
31         if (collision.transform.parent.tag == "RightHand")
32         {
33
34             Debug.Log("Hello");
35             //haptic.sendHapticFeedback(location, 'g');
36             RightHand = true;
37             LeftHand = false;
38
39         }
40         if (collision.transform.parent.tag == "LeftHand")
41         {
42             //haptic.sendHapticFeedback(location, 'g');
43             RightHand = false;
44             LeftHand = true;
45
46         }
47
48     }
49
50
51
52
53     public bool isRightHand()
54     {
55
56         return RightHand;
57     }
58     public bool isLeftHand()
59     {
60
61         return LeftHand;
62     }
63
64     //object1 = GameObject.Find("object1").GetComponent<>();*/
65 }

```

Listing A.8: Egg Instantiate

```

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class EggInstantiate : MonoBehaviour
6 {

```



```

7     private float maxDistanceX = 0.15f;
8     private float maxDistanceY = 0;
9     private float maxDistanceZ = 0.15f;
10
11    public int MaxPoints = PointManagerReal.MaxPoints;
12    public int TotalPoints;
13
14    private Vector3 startPos;
15    private Quaternion startQuat;
16
17    public GameObject egg;
18    SerialManager serial;
19    CSVWrite csv;
20    //PointManagerReal points;
21
22    private bool instantiate = false;
23
24    // Start is called before the first frame update
25    void Start()
26    {
27        serial = GameObject.Find("SerialManager").GetComponent<SerialManager>();
28        csv = GameObject.Find("CSVWrite").GetComponent<CSVWrite>();
29        startPos = transform.position;
30        startQuat = transform.rotation;
31        EggInstantiateMiddle();
32    }
33
34
35    public void EggInstantiateRandom()
36    {
37        TotalPoints = PointManagerReal.TotalPoints;
38        if (TotalPoints <= MaxPoints) {
39
40            instantiate = true;
41            StartCoroutine("InstantiateDelay");
42            //serial.sendStartEvent(true);
43
44        }
45        else
46        {
47            Debug.Log("Congratulations you have completed 15 grasps");
48        }
49    }
50
51    public void EggInstantiateMiddle()
52    {
53        Debug.Log("Egg is created");
54        Instantiate(egg, transform.position, transform.rotation);
55        serial.sendStartEvent(true);
56    }
57
58
59    private IEnumerator InstantiateDelay() // Coroutine that counts the length of the hold.
60    { // After 3s the hold is completed, a point is added, a new egg is instantiated and this
61        // current egg disappears at a random new location
62
63        yield return new WaitForSeconds(1.6f); // wait one and half seconds
64        if (instantiate == true)
65        { // do things
66            Debug.Log("Egg is created");
67            Vector3 position = startPos + new Vector3(Random.Range(-maxDistanceX,
68                maxDistanceX), Random.Range(-maxDistanceY, maxDistanceY), Random.Range(-
69                maxDistanceZ, maxDistanceZ));
70            Instantiate(egg, position, transform.rotation);
71            csv.WriteEggPositionToCSV(position);
72            instantiate = false;
73        }
74    }
75 }

```

Listing A.9: Serial Manager

```
1 using UnityEngine;
2 using System.Collections.Generic;
3 using System;
4 using System.Linq;
5 using System.IO.Ports;
6 using System.Collections;
7 using UnityEngine.EventSystems;
8
9 public class SerialManager : MonoBehaviour
10 {
11     public SerialPort serialPort;
12     public enumIntensity intensity;
13
14     // Start is called before the first frame update
15     void Start()
16     {
17         startSerial();
18     }
19
20     void startSerial()
21     {
22         // Creates Serial Connection to Port COM3
23         serialPort = new SerialPort("COM3", 9600);
24         serialPort.ReadTimeout = 100;
25         serialPort.Handshake = Handshake.None;
26         serialPort.Open();
27         Debug.Log("Serial Open");
28     }
29
30     public void sendHapticFeedback(char location, char action)
31     {
32         char[] vibComChar = new char[5];
33
34         vibComChar[0] = '<';
35         vibComChar[1] = location;
36         vibComChar[2] = action;
37         vibComChar[3] = getIntensity();
38         vibComChar[4] = '>';
39
40         string vibComString = string.Join(",", vibComChar);
41
42         sendThroughSerialPort(vibComString);
43     }
44
45     public void sendStartEvent(bool startEvent)
46     {
47         if (startEvent == true)
48         {
49             sendThroughSerialPort("<start>");
50         }
51         if (startEvent == false)
52         {
53             sendThroughSerialPort("<stop>");
54         }
55     }
56
57     void sendThroughSerialPort(string communication)
58     {
59         serialPort.Write(communication);
60         Debug.Log($"This information is send: {communication}");
61     }
62
63     public char getIntensity()
64     {
65         if (intensity == enumIntensity.intensity1)
66         {
67             return '1';
68         }
69         if (intensity == enumIntensity.intensity2)
```

```

70     {
71         return '2';
72     }
73     if (intensity == enumIntensity.intensity3)
74     {
75         return '3';
76     }
77     if (intensity == enumIntensity.intensity4)
78     {
79         return '4';
80     }
81     if (intensity == enumIntensity.intensity5)
82     {
83         return '5';
84     }
85     if (intensity == enumIntensity.intensity6)
86     {
87         return '6';
88     }
89     if (intensity == enumIntensity.intensity7)
90     {
91         return '7';
92     }
93     if (intensity == enumIntensity.intensity8)
94     {
95         return '8';
96     }
97     if (intensity == enumIntensity.intensity9)
98     {
99         return '9';
100    }
101
102    else return '0';
103
104    }
105
106    public enum enumIntensity
107    {
108        intensity1,
109        intensity2,
110        intensity3,
111        intensity4,
112        intensity5,
113        intensity6,
114        intensity7,
115        intensity8,
116        intensity9
117    };
118
119
120
121 }

```

Listing A.10: CSV writer

```

1
2 public class CSVWrite : MonoBehaviour
3 {
4
5     string dataFile = "";
6     string positionFile = "";
7     string positionFileFinger = "";
8     string positionEgg = "";
9
10
11     float currentTime;
12     int numberOfTry;
13     int amountGood;
14     int amountBad;
15
16     // Start is called before the first frame update

```

```

17 void Start()
18 {
19     dataFile = Application.dataPath + "/data.csv";
20     StreamWriter tw = new StreamWriter(dataFile, false);
21     tw.WriteLine("Time , Amount Good, Amount Bad, Total");
22     tw.Close();
23
24     positionFile = Application.dataPath + "/position.csv";
25     StreamWriter tw2 = new StreamWriter(positionFile, false);
26     tw2.WriteLine("Time, Posx, Posy, Posz, posxl, posyl, poszl");
27     tw2.Close();
28
29     positionFileFinger = Application.dataPath + "/positionFinger.csv";
30     StreamWriter tw3 = new StreamWriter(positionFileFinger, false);
31     tw3.WriteLine("Time, tPosx, tPosy, tPosz, iPosx, iPosy, iPosz, mPosx, mPosy, mPosz,
32         rPosx, rPosy, rPosz, pPosx, pPosy, pPosz ");
33     tw3.Close();
34
35     positionEgg = Application.dataPath + "/positionEgg.csv";
36     StreamWriter tw4 = new StreamWriter(positionEgg, false);
37     tw4.WriteLine("Time, Egg Position");
38     tw4.Close();
39 }
40
41 // Update is called once per frame
42 void Update()
43 {
44     /* if (Input.anyKeyDown)
45     {
46         Debug.Log("Start write");
47         WriteDataToCSV(1,2,3);
48     }*/
49 }
50
51 public void WriteDataToCSV(int numberOfTry, int amountGood, int amountBad)
52 {
53     currentTime = Time.time;
54     string currentTimeString = currentTime.ToString();
55
56     StreamWriter tw = new StreamWriter(dataFile, true);
57     tw.WriteLine(currentTimeString + "," + numberOfTry + "," + amountGood + "," +
58         amountBad);
59     tw.Close();
60 }
61
62 public void WritePositionToCSV(float posx, float posy, float posz, float posxl, float
63     posyl, float poszl)
64 {
65     currentTime = Time.time;
66     string currentTimeString = currentTime.ToString();
67
68     StreamWriter tw2 = new StreamWriter(positionFile, true);
69     tw2.WriteLine(currentTimeString + "," + posx + "," + posy + "," + posz + "," + posxl
70         + "," + posyl + "," + poszl);
71     tw2.Close();
72 }
73
74 public void WritePositionFingerToCSV(Vector3 thumb, Vector3 index, Vector3 middle,
75     Vector3 ring, Vector3 pinky)
76 {
77     currentTime = Time.time;
78     string currentTimeString = currentTime.ToString();
79
80     StreamWriter tw3 = new StreamWriter(positionFileFinger, true);
81     tw3.WriteLine(currentTimeString + "," + thumb.x + "," + thumb.y + "," + thumb.z + ","
82         + index.x + "," + index.y + "," + index.z + "," + middle.x + "," + middle.y + ","
83         + middle.z + "," + ring.x + "," + ring.y + "," + ring.z + "," + pinky.x + "," +
84         pinky.y + "," + pinky.z);
85     tw3.Close();

```

```

80     }
81
82     public void WriteEggPositionToCSV(Vector3 position)
83     {
84         currentTime = Time.time;
85         string currentTimeString = currentTime.ToString();
86
87         TextWriter tw4 = new StreamWriter(positionEgg, true);
88         tw4.WriteLine(currentTimeString + "," + position);
89         tw4.Close();
90     }
91
92
93
94 }

```

Listing A.11: Points Manager

```

1
2 public class PointManagerReal : MonoBehaviour
3 {
4     public static int TotalPoints = 0;
5     public static int TotalGood = 0;
6     public static int TotalBad = 0;
7     public const int MaxPoints = 10;
8     bool endGame;
9
10    public UIManager UIText;
11    private CSVWrite csv;
12    SerialManager serial;
13    public EggInstantiate egg;
14
15
16
17    // Start is called before the first frame update
18    void Awake()
19    {
20        TotalPoints = 0;
21    }
22
23
24    private void Start()
25    {
26        UIText = GameObject.Find("Canvas").GetComponent<UIManager>();
27        csv = GameObject.Find("CSVWrite").GetComponent<CSVWrite>();
28        serial = GameObject.Find("SerialManager").GetComponent<SerialManager>();
29        //egg = GameObject.Find("EggManager").GetComponent<EggInstantiate>();
30
31        if (UIText == null)
32        {
33            Debug.Log("UITEXT is null");
34        }
35    }
36
37
38
39
40    private void Update()
41    {
42
43        if (TotalPoints == MaxPoints & endGame == false)
44        {
45            Debug.Log("Congratulations");
46            endGame = true;
47        }
48    }
49
50
51
52
53    public void UpdateScoreGood()

```

```
54 {
55     if (TotalPoints < MaxPoints)
56     {
57         TotalGood++;
58         TotalPoints++;
59         Debug.Log(TotalGood);
60         UIText.ShowScore(TotalGood, TotalBad, TotalPoints);
61         egg.EggInstantiateRandom();
62         //serial.sendStartEvent(false);
63         csv.WriteDataToCSV(TotalGood, TotalBad, TotalPoints);
64         //here write end time
65     }
66 }
67
68 public void UpdateScoreBad()
69 {
70     if (TotalPoints < MaxPoints)
71     {
72         TotalBad++;
73         TotalPoints++;
74         Debug.Log(TotalBad);
75         UIText.ShowScore(TotalGood, TotalBad, TotalPoints);
76         egg.EggInstantiateRandom();
77         //serial.sendStartEvent(false);
78         csv.WriteDataToCSV(TotalGood, TotalBad, TotalPoints);
79         //here write end time
80     }
81 }
82
83 static public int getTotalPoints()
84 {
85     int totalPoints = TotalPoints;
86     return totalPoints;
87 }
88
89
90 public int getGoodPoints()
91 {
92     int totalGood = TotalGood;
93     return totalGood;
94 }
95
96
97 public int getBadPoints()
98 {
99     int totalBad = TotalBad;
100    return totalBad;
101 }
102
103
104
105 }
```

B

## Questionnaires

**1. Kennen Sie haptisches/vibratives Feedback?**

- Ja  
 Nein

**2. Wie haben Sie das vibrierende Gefühl erlebt?. Bitte geben Sie auf den folgenden Skalen eine Punktzahl von 1 bis 10 an.**

*Sehr angenehm* 1 2 3 4 5 6 7 8 9 10 *Sehr unangenehm*

*Sehr schmerzhaft* 1 2 3 4 5 6 7 8 9 10 *Nicht schmerzhaft*

*Sehr ablenkend* 1 2 3 4 5 6 7 8 9 10 *Nicht ablenkend*

**3. Sie haben bereits ein Spiel in VR erlebt. Glauben Sie, dass das Hinzufügen des Vibrationsgefühls zu einem Berührungseignis die Spiele realistischer machen würde? Bitte geben Sie auf den folgenden Skalen eine Punktzahl von 1 bis 10 an.**

*Gar nicht* 1 2 3 4 5 6 7 8 9 10 *Ja voll und ganz*

**4. Glauben Sie, dass das Hinzufügen des Vibrationsgefühls zu einem Berührungseignis die Spiele einfacher machen könnte?**

*Gar nicht* 1 2 3 4 5 6 7 8 9 10 *Ja voll und ganz*

**5. Haben Sie Empfehlungen zur Implementierung von Vibrationsfeedback in den Handschuh?**

- Standorte: \_\_\_\_\_  
 Intensitäten: \_\_\_\_\_  
 Anzahl Standorte: \_\_\_\_\_

**6. Gibt es weitere Anmerkungen zum Feedback, die zu den Virtual-Reality-Spielen hinzugefügt werden könnten? (visuell, haptisch, audio)**

---



---

**Figure B.1:** Questionnaire for preliminary testing



## Fragen speziell zum haptischen Feedback

Haptisches Feedback bezieht sich auf alles, was Sie mit dem Tastsinn fühlen. Es können Vibrationen, Kraft, Temperatur, Druck oder jede andere körperliche Empfindung sein <sup>1</sup>. Während des Tests erlebten Sie Vibrationen als haptisches Feedback als Hinweis auf den Beginn eines Berührungseignisses. Dieses Vibrationsgefühl ist das haptische Feedback, auf das wir uns in den folgenden Fragen beziehen.

	Lehne völlig ab	Stimme völlig zu
1. Das haptische Feedback war zufriedenstellend	1   2   3   4   5	
2. Mir gefällt, wie sich das haptische Feedback selbst anfühlt, unabhängig von seiner Rolle im System	1   2   3   4   5	
3. Ich mochte das haptische Feedback nicht	1   2   3   4   5	
4. Ich würde das System ohne das haptische Feedback bevorzugen	1   2   3   4   5	
5. Ich mag es, das haptische Feedback als Teil des Erlebnisses zu haben	1   2   3   4   5	
6. Ich fühlte mich aufgrund des haptischen Feedbacks mit dem System verbunden	1   2   3   4   5	
7. Das haptische Feedback half mir, mich auf die Aufgabe zu konzentrieren	1   2   3   4   5	
8. Das haptische Feedback verstärkte meine Beteiligung an der Aufgabe	1   2   3   4   5	
9. Das haptische Feedback hilft mir zu unterscheiden, was los war	1   2   3   4   5	
10. Das haptische Feedback war realistisch	1   2   3   4   5	
11. Das haptische Feedback war glaubwürdig	1   2   3   4   5	
12. Das haptische Feedback war überzeugend	1   2   3   4   5	
13. Das haptische Feedback fühlte sich vom Rest der Erfahrung abgekoppelt an	1   2   3   4   5	
14. Das haptische Feedback fühlte sich angemessen an, wann und wo ich es fühlte	1   2   3   4   5	
15. Das haptische Feedback fühlte sich fehl am Platz an	1   2   3   4   5	
16. Das haptische Feedback lenkte mich von der Aufgabe ab	1   2   3   4   5	
17. Das haptische Feedback hatte eine angenehme Intensität	1   2   3   4   5	
18. Das haptische Feedback erleichterte das Greifen	1   2   3   4   5	
19. Das haptische Feedback hat mich beim virtuellen Greifen sicherer gemacht	1   2   3   4   5	
20. Das haptische Feedback war irritierend	1   2   3   4   5	
21. Das haptische Feedback machte es nicht einfacher, meine Greifkraft zu kontrollieren	1   2   3   4   5	

<sup>1</sup>Sathiyamurthy, 2020.

## Fragen zum Gesamtsystem

Mit *dem System* meinen wir Folgendes: Der Handschuhe mit vibrierendem haptischem Feedback, die für die Rehabilitation in der virtuellen Realität verwendet werden könnte.

- |                                                                                                                        | Lehne<br>völlig ab | Stimme völlig zu |
|------------------------------------------------------------------------------------------------------------------------|--------------------|------------------|
| 22. Ich denke, ich würde das System regelmäßig nutzen                                                                  | 1   2   3   4   5  |                  |
| 23. Das System erscheint mir unnötig kompliziert                                                                       | 1   2   3   4   5  |                  |
| 24. Ich finde, das System ist einfach zu benutzen                                                                      | 1   2   3   4   5  |                  |
| 25. Ich denke, ich bräuchte technische Unterstützung um das System nutzen zu können                                    | 1   2   3   4   5  |                  |
| 26. Ich finde, dass die verschiedenen Funktionen des Systems gut integriert sind                                       | 1   2   3   4   5  |                  |
| 27. Das System erscheint mir zu uneinheitlich                                                                          | 1   2   3   4   5  |                  |
| 28. Ich glaube, dass die meisten Leute die Benutzung des Systems schnell erlernen können                               | 1   2   3   4   5  |                  |
| 29. Das System erscheint mir sehr umständlich zu benutzen                                                              | 1   2   3   4   5  |                  |
| 30. Ich fühle mich bei der Benutzung des Systems sehr sicher                                                           | 1   2   3   4   5  |                  |
| 31. Ich musste einiges lernen, um mit dem System zurecht zu kommen                                                     | 1   2   3   4   5  |                  |
|                                                                                                                        |                    |                  |
|                                                                                                                        | Gar nicht          | Sehr viel        |
| 32. Wie sehr schien Ihre Erfahrung in der virtuellen Umgebung mit Ihrer Erfahrung in der realen Welt übereinzustimmen? | 1   2   3   4   5  |                  |
| 33. Hat Ihnen Ihre Erfahrung mit dem System gefallen?                                                                  | 1   2   3   4   5  |                  |
| 34. Haben Sie sich während Ihrer Erfahrung mit dem System unwohl gefühlt?                                              | 1   2   3   4   5  |                  |
| 35. Glauben Sie, dass dieses System für Ihre Rehabilitation hilfreich sein wird?                                       | 1   2   3   4   5  |                  |
| 36. Haben Sie Empfehlungen zur Verbesserung des haptischen Feedbacks während der Virtual-Reality-Rehabilitation?       |                    |                  |

---



---



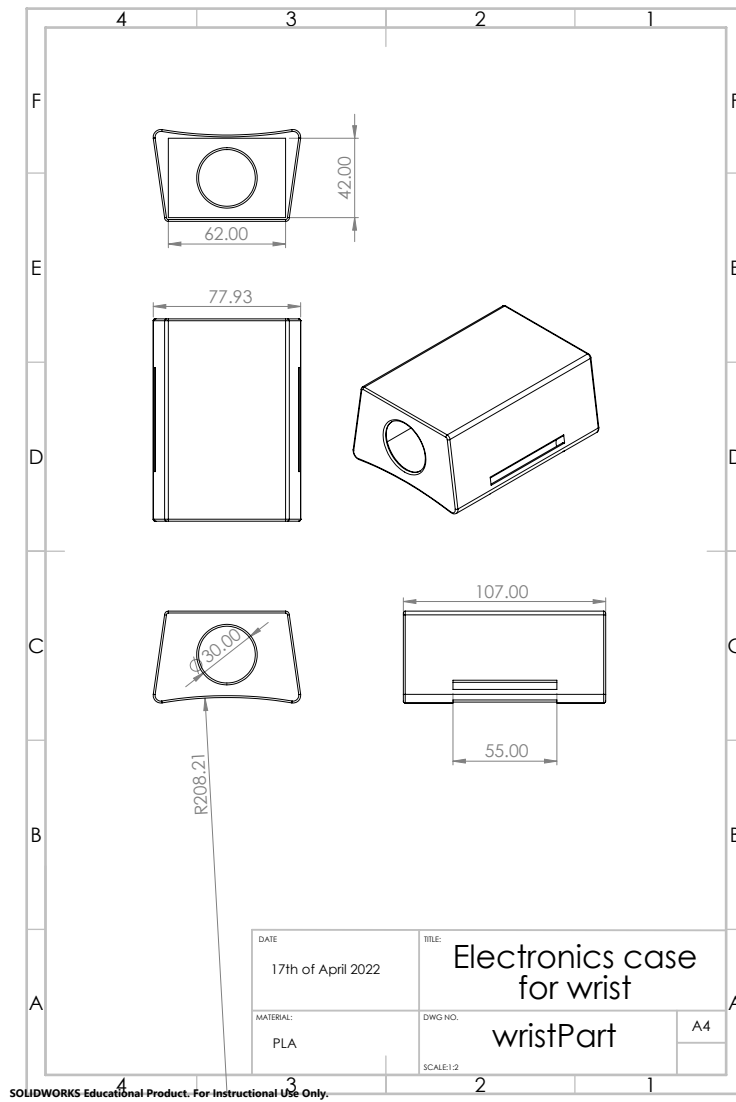
---

Questionnaire	System Focus	Domain	Question	Dutch	Answer method	Notes
HX scale model	Haptic	Autoteleic	The haptic feedback felt satisfying	De haptische feedback voelde bevredigend aan	Lehne volledig ab - stemme volledig zu	ja
HX scale model	Haptic	Autoteleic	I like how the haptic feedback itself feels, regardless of its role in the system	Ik hou van hoe de haptische feedback zelf aanvoelt, ongeacht zijn rol in het systeem	5 pt Likert Scale	ja
HX scale model	Haptic	Autoteleic	I disliked the haptic feedback	Ik hield niet van de haptische feedback	5 pt Likert Scale	ja
HX scale model	Haptic	Autoteleic	I would prefer the system without the haptic feedback	Ik zou het systeem liever hebben zonder de haptische feedback	5 pt Likert Scale	ja
HX scale model	Haptic	Autoteleic	I like having the haptic feedback as part of the experience	Ik vind het leuk om de haptische feedback als onderdeel van de ervaring te hebben	5 pt Likert Scale	ja
HX scale model	Haptic	Immersion	I felt engaged with the system due to the haptic feedback	Ik voelde me betrokken bij het systeem door de haptische feedback	5 pt Likert Scale	ja
HX scale model	Haptic	Immersion	The haptic feedback helped me focus in the task	De haptische feedback hielp me om me te concentreren op de taak	5 pt Likert Scale	ja
HX scale model	Haptic	Immersion	The haptic feedback increased my involvement in the task	De haptische feedback verhoogde mijn betrokkenheid bij de taak	5 pt Likert Scale	ja
HX scale model	Haptic	Immersion	The haptic feedback helps me distinguish what was going on	De haptische feedback helpt me te onderscheiden wat er aan de hand was	5 pt Likert Scale	ja
HX scale model	Haptic	Realism	The haptic feedback was realistic	De haptische feedback was realistisch	5 pt Likert Scale	ja
HX scale model	Haptic	Realism	The haptic feedback was believable	De haptische feedback was geloofwaardig	5 pt Likert Scale	ja
HX scale model	Haptic	Realism	The haptic feedback was convincing	De haptische feedback was overtuigend	5 pt Likert Scale	ja
HX scale model	Haptic	Harmony	The haptic feedback felt disconnected from the rest of the experience	De haptische feedback voelde los van de rest van de ervaring	5 pt Likert Scale	ja
HX scale model	Haptic	Harmony	The haptic feedback felt appropriate when and where I felt it	De haptische feedback voelde gepast waar en wanneer ik het voelde	5 pt Likert Scale	ja
HX scale model	Haptic	Harmony	The haptic feedback felt out of place	De haptische feedback voelde niet op zijn plaats	5 pt Likert Scale	ja
HX scale model	Haptic	Harmony	The haptic feedback distracted me from the task	De haptische feedback leidde me af van de taak	5 pt Likert Scale	ja
Self	Haptic		The haptic feedback had a comfortable intensity	De haptische feedback had een comfortabele intensiteit	5 pt Likert Scale	
Self	Haptic		The haptic feedback made the grasp easier	De haptische feedback maakte de greep gemakkelijker		
Self	Haptic		The haptic feedback made me more confident during the virtual grasp	De haptische feedback maakte me zelfverzekerder tijdens de virtuele greep	5 pt Likert Scale	
Self	Haptic		The haptic feedback was irritating	De haptische feedback was irritant	5 pt Likert Scale	
Self	Haptic		The haptic feedback did not make controlling my grabbing power easier	De haptische feedback maakte het niet gemakkelijker om mijn grijpkracht te beheersen	5 pt Likert Scale	
Tcha Tokey (2011) Overall		Presence	The device (glove with electronics) which controlled my movement distract me from performing assigned tasks		5 pt Likert Scale	Nee
Tcha Tokey (2011) Overall		Presence	I could concentrate on the assigned tasks rather than on the device (glove with electronics)		5 pt Likert Scale	Nee
Tcha Tokey (2011) Overall		Flow	I felt I could perfectly control my actions		5 pt Likert Scale	Nee
Tcha Tokey (2011) Overall		Flow	At each step I knew what to do		5 pt Likert Scale	Nee
Tcha Tokey (2011) Overall		Skill	I felt confident selecting objects in the virtual environment		5 pt Likert Scale	Nee
Tcha Tokey (2011) VR		Technological ad	If I use the same virtual Environment (-> device?) again, my interaction with the environment would be clear and understandable for me		5 pt Likert Scale	Nee
Tcha Tokey (2011) VR		Technological ad	It would be easy for me to become skillfull at using the virtual environment (-> device?)		5 pt Likert Scale	Nee
Tcha Tokey (2011) Overall			In your opinion, what where the positive points about your experience?		5 pt Likert Scale	Nee
Tcha Tokey (2011) Overall			In your opinion, what where the negative points about your experience?		5 pt Likert Scale	Nee
Tcha Tokey (2011) Overall			Do you have suggestions to improve this virtual reality experience?		5 pt Likert Scale	Nee
IPQ	VR	Realism	How much did your experience in the virtual environment seem consistent with your real world experience ?		völlig übereinstimmen - völlig unpassend	ja
SUS	Overall	Usability	I think that I would like to use this system frequently.	Ik denk dat ik dit systeem vaak zou willen gebruiken.	1-5 scale strongly agree-stornly disagree	ja
SUS	Overall	Usability	I found the system unnecessarily complex.	Ik vond het systeem onnodig ingewikkeld.	1-5 scale strongly agree-stornly disagree	ja
SUS	Overall	Usability	I thought the system was easy to use.	Ik dacht dat het systeem gemakkelijk te gebruiken was.	1-5 scale strongly agree-stornly disagree	ja
SUS	Overall	Usability	I think that I would need the support of a technical person to be able to use this system.	Ik denk dat ik de steun van een technisch persoon nodig heb om dit systeem te kunnen gebruiken.	1-5 scale strongly agree-stornly disagree	ja
SUS	Overall	Usability	I found the various functions in this system were well integrated.	Ik vond de verschillende functies in dit systeem goed geïntegreerd.	1-5 scale strongly agree-stornly disagree	ja
SUS	Overall	Usability	I thought there was too much inconsistency in this system.	Ik vond dat er teveel inconsistentie in dit systeem zat.	1-5 scale strongly agree-stornly disagree	ja
SUS	Overall	Usability	I would imagine that most people would learn to use this system very quickly.	Ik kan me voorstellen dat de meeste mensen dit systeem heel snel zouden leren gebruiken.	1-5 scale strongly agree-stornly disagree	ja
SUS	Overall	Usability	I found the system very cumbersome to use.	Ik vond het systeem erg omslachtig in gebruik.	1-5 scale strongly agree-stornly disagree	ja
SUS	Overall	Usability	I felt very confident using the system.	I felt very confident using the system.	1-5 scale strongly agree-stornly disagree	ja
SUS	Overall	Usability	I needed to learn a lot of things before I could get going with this system.	Ik moest veel dingen leren voordat ik met dit systeem aan de slag kon.	1-5 scale strongly agree-stornly disagree	ja
USEQ	Overall	Emotion	Did you enjoy your experience with the system (gloves with haptic feedback) ?	Heb je genoten van je ervaring met het systeem (handschoenen met haptische feedback)?	gar nicht - sehr viel	ja
USEQ	Overall		Did you feel discomfort during your experience with the system?	Voelde u zich ongemakkelijk tijdens uw ervaring met het systeem?	gar nicht - sehr viel	ja
USEQ	Overall	Autoteleic	Do you think that this system will be helpful for your rehabilitation?	Denkt u dat dit systeem nuttig zal zijn voor uw revalidatie?	gar nicht - sehr viel	ja
Open question			Do you have any recommendations to improve the haptic feedback during Virtual Reality Rehabilitation?	Heeft u aanbevelingen om de haptische feedback tijdens Virtual Reality Rehabilitation te verbeteren?		

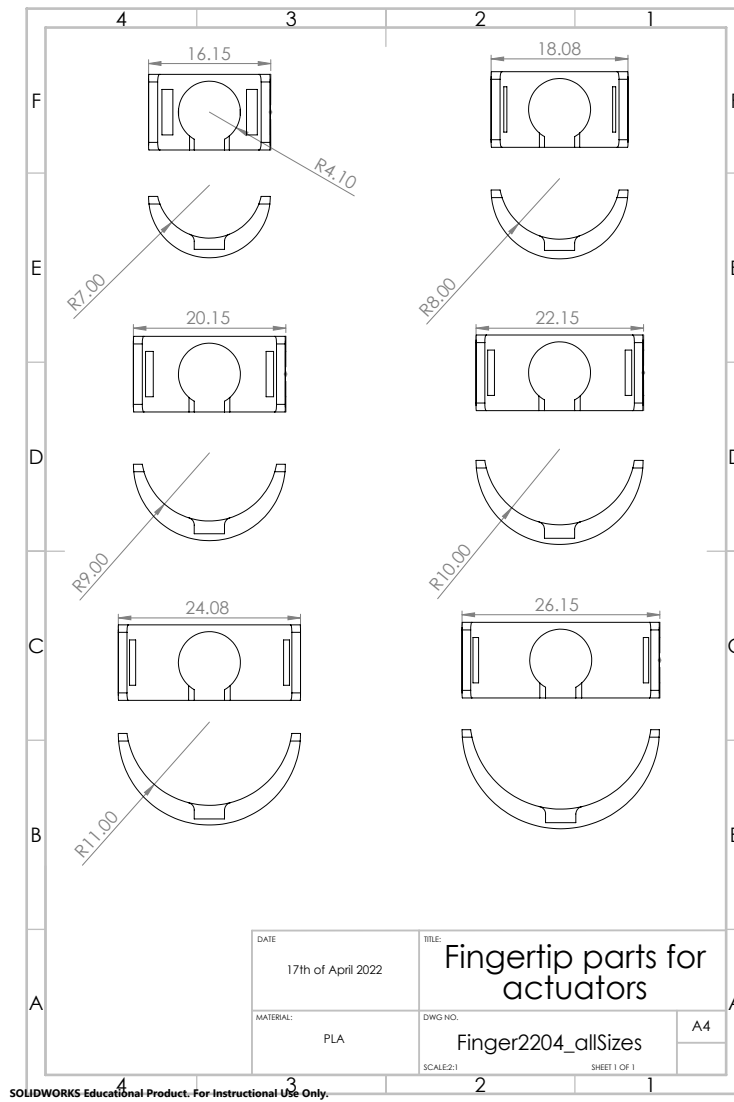
Figure B.2: Evaluated (all) and chosen (green) questions for the questionnaire

C

3D Models



**Figure C.1:** Drawings of all the designed fingertip parts for the actuators



**Figure C.2:** Drawings of all the designed fingertip parts for the actuators