Artturi Lobski

Pneumatic controller prototype and haptic feedback

Master's Thesis in Information Technology

May 29, 2019

University of Jyväskylä

Faculty of Information Technology

Author: Artturi Lobski

Contact information: arlobski@student.jyu.fi

Supervisors: Lauri Kettunen, and Paavo Nieminen

Title: Pneumatic controller prototype and haptic feedback

Työn nimi: Pneumaattinen ohjainprototyyppi ja haptinen palaute

Project: Master's Thesis

Study line: Pelit ja pelillisyys

Page count: 35+7

Abstract: The main themes in this master's thesis are human computer interaction (HCI) and development of the HCI. The question this research tries to answer is, whether it is possible to portray the rigidness of virtual objects by exploiting pneumatics (pressurized air) better than by means of the more common vibrotactile feedback technology. So far, there has been little research on this matter, so it is interesting to find out if the new technology could improve the sense of immersion and the usability of HCI in virtual reality applications. This research is based on previous research on haptic controllers in the video game scene and exploitation of pneumatics in the controllers. A new prototype pneumatic controller is realised and compared with the existing HTC vive controller. Both controllers are tested on suitability of portraying the rigidness of virtual objects. The results showed that it is possible to exploit pneumatics in controllers, and that the prototype's feedback is more natural for portraying the rigidness of objects than vibrotactile feedback controllers. The research is in its beginning phase and different uses have to be tested and researched further to ensure the necessity of pneumatic haptic controllers. The results can be useful in pneumatic controller research. This research also provides a good basis for development and use of the presented prototype.

Keywords: Virtual reality, controllers, pneumatics

Suomenkielinen tiivistelmä: Tämän pro gradu -tutkielman keskeisenä aiheena on ihmisen ja tietokoneen välinen vuorovaikutus ja sen edistäminen. Tutkimuksella pyritään vastaamaan kysymykseen, voidaanko pneumatiikkaa eli paineilmaa hyödyntävällä ohjaimella kuvata virtuaalisten esineiden lujuutta paremmin kuin yleisemmin käytetyllä värinäpalautteella. Tätä yksityiskohtaa on tutkittu toistaiseksi vähän, joten on mielenkiintoista selvittää, voisiko uusi teknologia parantaa vuorovaikutuksen immersiota ja käytettävyyttä. Tutkimus perustuu aiemmin tehtyyn tutkimukseen haptisten ohjainten käytöstä pelitarkoituksissa ja pneumatiikan hyödyntämisestä erilaisissa ohjaimissa. Tutkimus on toteutettu luomalla prototyyppi pneumatiikkaa hyödyntävästä ohjaimesta. Prototyypin toimintaa verrataan olemassa olevaan HTC vive -ohjaimeen ja molempien ohjainten soveltuvuutta virtuaalisten esineiden lujuuden kuvaamiseen. Tutkimustulokset osoittivat, että pneumatiikkaa on mahdollista hyödyntää ohjaimissa ja että sitä hyödyntäen on mahdollista kuvata lujuutta luonnollisemmin kuin värinäpalautetta hyödyntäen. Tutkimus on alkuvaiheessa ja erilaisia käyttötarkoituksia on testattava ja tutkittava vielä jatkossa, jotta voidaan varmistaa pneumaattisten haptisten ohjainten tarpeellisuus. Tuloksia voi hyödyntää pneumaattisia ohjaimia käsittelevissä jatkotutkimuksissa. Lisäksi tutkimus tarjoaa hyvän pohjan prototyypin kehitykselle ja hyödyntämiselle.

Avainsanat: Virtuaalitodellisuus, ohjaimet, pneumatiikka

List of Figures

Contents

1 Introduction

This thesis will be discussing Human-Computer Interaction (HCI), different tactile feedbacks, and Virtual Reality (VR). HCI means the study of the relationship between a human and a machine and their combined effort to solve simple tasks. A simple task can be, for example, pushing a button to start a machine or using more complex button combinations to launch a Task manager in the Windows operating system. One way to study HCI is by generating a haptic sensation to the user and observing if the completion of the given task improves.

Haptic, or kinaesthetic, communication means the creation of an artificial sense of touch. By creating a haptic controller, the feedback that is generated from the controller to the user can be enhanced and by triggering as many as possible senses of the user. By this enhancement, the more "natural" feel of the performed task can be achieved.

That brings us to the next question: what is an appropriate task to test the haptic controller and how can the controller be employed? One of the technologies that is coming back to the market and getting more common is VR. VR means the creation of simulated, virtual environments that are perceived as real by their user. VR usually consists of a Head Mounted Display (HMD) to bring the visualization of the virtual environment, controllers to interact in the environment and with the environment, and tracking stations to locate the user to the right coordinates in the environment and to track user's movement.

This thesis will concentrate on exploring new possibilities with future technologies in the game industry and hopefully will also contribute to the next generation of controllers. At the moment, the problem with the most commonly used game controllers is the lack of intuitive controls and interaction with the environment. And so this study will bring ideas on how can pneumatic force be exploited in a controller. Pneumatic feedback will be put against vibrotactile feedback and will be tested as an alternative way of portraying rigidness of virtual objects. The more "natural" and intuitive controls will present users with more immersive environments and experience. So the main question is: "can controllers be made more intuitive and immersive by exploiting pneumotactile feedback".

Chapter 2 of this thesis will progress by first exploring the technological state at this point of time, and explore the journey to this point. Next there will be thorough explanation of the main themes of this thesis. In Chapter 3, there will be description of the prototype assembled for this thesis. In Chapter 4, testing of the prototype will be discussed. Finally, Chapter 5 concludes the thesis by explaining how it went and if the desired results were achieved and what are the next steps in the future of development.

2 Background

The work presented in this thesis builds on virtual reality, pneumatic tactile feedback and immersion enhancing. The desired features will be achieved by exploiting prior research and turning the ideas into a more practical product. In this chapter, the historical progress of HCI controllers will be discussed and virtual environments in general. The topics will also be tactile technology and already existing research and works.

2.1 History

The first game controller was made in 1961 for a game called *"spacewar!"*. The controller had a couple of digital switches and knobs for rotating. The first joystick that was based on an aircraft controller was introduced by Atari in 1977. The first gamepad came with the Atari 7800 console, but the actual potential of the gamepad was realised by the Nintendo Entertainment Systems (NES) gamepad in 1983. The NES controller had a D-pad (four directional buttons) and some buttons (start, select, A, and B). When 3D games arrived, controllers that were designed for 2D games were inadequate and analogue sticks were added to controllers. The first one was Nintendo 64 (N64) console in 1996. In the 1990s, companies started implementing haptic feedback into their controllers, and this was the first time controllers had a bi-directional interaction with a user. (Merdenyan and Petrie 2015)

At the same time, another technology was under intensive research – Virtual Reality (VR). The first virtual reality was *Sensorama*. It was designed in the 1950s and its first mechanical prototype was launched in 1962. Sensorama had 3D-visuals, audio, olfactory and haptic stimulus such as wind. Sensorama played a movie about riding a motorcycle in New York, but the viewer could not affect the environment in any way. About at the same time, in 1961, the first Head Mounted Display (HMD) called *Headsight* was designed by Phlico. Better known as HMD, or as Ivan Sutherland called it: BOOM (Binocular Omni Orientation Monitor), *The Sword of Damocles* was made in 1968. The name pointed to the size and the positioning of the device over the head as in the Greek story of Damocles. One of the first VR related controllers was the Wired Gloves. The first was *Sayre glove* in 1977 that was developed by Tom DeFranti and Daniel J. Sandin. It was based on fiber-optics technology. In the 1980s, Thomas G. Zimmerman worked on a *Dataglove* that later influenced, for example, an infamous controller *Power Glove* that was made for the NES. After Power Glove's bad reception, VR development regressed to mostly military and academic research until it became more affordable for the public at a much later date. (Boas 2013)

Nowadays the most used input devices are: Wired gloves, wands, and computer vision. Gloves are classified into three different technology categories. The first ones employ light and fiber optics. The measurements of a joint angle are determined by the amount of light received by a photocell at the end of the fiber, meaning less light received the higher is the bend of the joint. The second technology is based on measuring the electrical resistance of conductive ink. Resistance changes as the user moves the hand. This technology has the worst performance but it is very cost-effective. The third technology employs mechanical sensors. Gloves that are made using mechanical sensors are usually more precise but are larger and more expensive, such as one of the newest additions, *HaptX Glove*. The mechanical part that restricts the finger going through the objects works by using microfluidic actuators that can apply up to four pounds of force on each finger. The same microfluid technology is employed in HaptX microfluidic skin that has 130 actuators per glove. Microfluidic skin actuators push against the user's skin and provide force feedback generating a feeling of touch. HaptX Glove exploits magnetic motion tracking of the fingers (HaptX Inc. 2018).

The first popular wand like motion-sensor controller was introduced by Nintendo in the *Nintendo Wii* console. The *Wii Remote* has an infrared (IR) sensor that receives IR light from a sensor bar located near a television. The sensor bar has five IR emitters on each side. The calculation of the position of the wand takes place in the console. The controller relies on triangulation of IR light. Wii controller has also accelerometers to detect acceleration, gyroscopes for angular velocities, speakers, and a rumble pack to provide feedback to the user. Sony has also made a wand controller for their *Playstation* (PS). It is a combination of *PS eye* and *PS move*. PS move is a wand that has a magnetometer to calibrate the inertia sensors and a colorful ball shaped lights on top. The balls are for PS eye to recognize the controller and to get a better tracking of the controllers.

The article by Boas (2013) considers computer vision also as a controller. Boas (2013) mentions PS eye, but the emphasis is on Microsoft product *Kinect*, which came with *Xbox360*. Kinect has an RGB camera that recognizes faces, a depth sensor to scan the surroundings, a microphone for sound, and a microchip to process the movements of the user.

2.2 Human-computer interaction

Human Computer Interaction (HCI) examines the relationship between a human and a machine and their combined effort to resolve a simple task. A simple task can vary, but it always consists of a user, a controller, and a machine. What makes HCI good or bad is the ability to complete a task as well as possible, meaning that performance is the main objective. Performance of the task consists of human performance, i.e., learning the controls and efficient use of the tool, and system performance, i.e., the speed of calculations, complexity of structure, and intuitive design. Results of the performance can be measured with the time spent on learning controls, the time used on a task completion, the number of errors that happened during the completion, and the cost of evaluation, meaning how fast the user gets comfortable with the controls. (Card, Moran, and Newell 1983)

To present a concrete example of HCI planning, the prototype of this thesis will be used as an example. The first step in the design is to consider the psychology of the user and the design of the user interface (UI). Starting from determining a task for both the user and the machine which in this case is "assessing the rigidness of a material in a virtual environment". Next step is specifying the user and performance requirements of the system. The targeted users in this case are VR platform users that desire more immersion in their experience with objects and specifically the rigidness of materials. The system requirements are virtual environment and machine that runs and enables interaction with it. (Card, Moran, and Newell 1983, 418-420)

The next step is specifying methods in order to perform the task and analyzing the chosen methods. In the solution presented in this thesis, the method for this task completion is to construct a controller, which is going to inflate with air so much that the rigidness fits the object in the virtual environment. So, when the user squeezes the controller, the feeling of the rigidness will be experienced (Card, Moran, and Newell 1983, *Principle 5: Specify the* *methods to do the task*). As mentioned by Card, Moran, and Newell (1983), the analyzing of the methods changes in different stages of the progress in a HCI project. Initially, the described method seems like a good solution and analysis of its effectiveness will be decided by the users of the controller.

The next thing Card, Moran, and Newell (1983) propose is that the focus should be on reducing the time of task completion. Solutions in this case are: squeezing a chamber filled with pressurized air, pumping air into the chamber, pumping air out, detecting pressure in the chamber, selecting a specific pressure for an object. To reduce the time for task completion, several unit tasks may possibly be combined into a single unit task, or the number of unit tasks may by reduced.

To design HCI as flawlessly as possible, there should be alternative methods in order to perform the same task. That is one reason why in this project a vibrotactile solution is also present. The alternative method should be "clear to the user and easy to apply" as Card, Moran, and Newell (1983) write. Vibrotactile solution is common tactile feedback that is used in current phones, toys, tablets and controllers (Israr et al. 2015), and for that reason it is chosen as an alternative, easily applicable, method.

Errors and how to handle them is also part of planning the HCI (Card, Moran, and Newell 1983). In the HCI presented in this thesis, human error is probably more common and anticipated than machine error, but both are equally possible. Human errors in this case could be misreading the pressure of the chamber by the feel or performing erroneous actions for reasons unrelated to the feel. These errors can not be corrected beforehand or while testing is taking place but participants could be informed beforehand to be cautious while testing and ask them to state if a mistake is made consciously. When the product is available to the public, after that those mistakes are uncontrollable.

On the hardware side, there could be a couple of errors and most of them are related to the pressure in the device. Because pressure can change rapidly, there is the possibility that the used pressure sensor does not have time to react to the pressure changes and the pressure will resultingly rise higher than anticipated. Such delay could also be a result of processing delay in different parts of software. This could be fixed by resetting the pressure and trying again to try and fill the chamber to the right pressure or by switching to a more precise pressure sensor, but for this solution, the precision that is present was considered sufficient.

The last principle described by Card, Moran, and Newell (1983) is analyzing the sensitivity of performance predictions to assumptions. With the presented solution, there are not many variables that could change, so the analyzing is simpler than in more complex systems. User side performance could be affected by an unintentional use of the product. For example, squeezing the chamber before it is filled results in wrong pressure when the chamber is released. That could be a result of not reading a manual or listening to instructions. Using wrong PC hardware could possibly affect performance because VR is used and it has its own requirements for hardware and behaviour that is hard to anticipate in those cases.

2.3 Haptic controllers

Controllers are important as they are the tools for interaction with environments that are unreachable otherwise. If the most common and widely used controllers are compared with haptic controllers, that are getting more common these days, the difference in feedback is quite obvious. Haptic controllers are designed to give realistic feedback and reactions to the use of tools and of the environment. To feel like one is grabbing an object just pressing a button on a controller does not reflect well the real life scenario. This motivates the use of haptic controllers. A haptic controller allows the user to feel the resistance when an object is touched and feeling of dimensions of the object can be comprehended too. One of the ways to simulate such action is by using *artificial muscles* as in the controller introduced by Lee and Ryu (2008). The basic idea of the artificial muscles is to contract to feel resistance if an obstacle is met and relax when nothing is on the way. Contraction and relaxation of the artificial muscle is done by pumping a hydraulic fluid in and out of the muscle. However, only pumping fluids in and out is not making the device a haptic or a tactile one.

Haptic, or kinaesthetic, communication is an artificial system that recreates the sense of touch by mechanical simulation. Haptics can be divided into two parts. The first part is kinaesthetic (proprioception) which enables the user to feel the counter force to touch. The second one is tactile (cutaneous) feedback which enables the user the skin-sensation such as temperature, a vibration, or the feeling of the surface of the object (Ramsamy et al. 2006).

This thesis will concentrate on the tactile aspect of the haptics. As mentioned before, the prototype will exploit pneumatic force to model the rigidness of objects. Even though the prototype uses pneumatics and applies a pressure, it is still not considered kinaesthetic haptics but a tactile one. For me, the reasoning is, to be kinaesthetic the user should feel the weight of the object or sense squeeze. In the prototype, however, only the rigidness of the object is varied changing users 'feel' or 'sense' of the touch on objects. For this purpose, I find "tactile" to be the more appropriate wording for the presented type of feedback.

2.4 Tactile stimulus

Haans and IJsselsteijn (2006) explain that the human sense of touch can be divided into five different types of stimuli, most of which can be replicated technically. One of those stimuli is pain, which can be inflicted by deforming skin. Another, more uncommon, stimulus is the inclination which means that one can feel how the limbs are positioned with respect to the body. It is not known if there are haptic controllers that could imitate or exploit this stimulus, but inclination is usually imitated with visual cues. Haans and IJsselsteijn (2006) present also temperature, which can be imitated with different heating sensors and fans. The last two stimuli that will be discussed in more detail for the purposes of this thesis are vibration and pressure.

2.4.1 Vibrotactile sensation

A vibrotactile stimulus can be felt, for example, by touching loudspeakers when they are played loudly or by touching big machines when they are operating. A vibrotactile stimulus is usually generated by a motor with a weight at the end of a shaft. While the motor is running, the motion of the weight repeatedly displaces the centre point of mass, thus causing vibration.

In a study by Suhonen, Väänänen-Vainio-Mattila, and Mäkelä (2012), on one hand, the participants of a haptic feedback experiment felt that strong vibrations were loud and weak vibrations were sometimes not felt at all. On the other hand, different kinds of sequences of vibrations were easily distinguishable from each other.

In a study by Haller et al. (2013), vibrotactile feedback was considered *disturbing and disruptive* by the participants and the author. It was somewhat a desired reaction but some of the participants felt that vibration was so disturbing that in longer use they preferred to switch the device off.

Even though vibrotactile feedback is inexpensive and common (Israr et al. 2015), it is not suited for the task of getting a rigid feeling of an object. Rigidness could possibly be indicated by regulating the power of vibration but it is not natural, and soft objects would not be felt at all as explained by Suhonen, Väänänen-Vainio-Mattila, and Mäkelä (2012).

2.4.2 Pneumotactile sensation

Pneumotactile sensation can be generated by pressurized air, and there are many methods to create such a tactile sensation. One of the methods is described by Becke et al. (2015). Becke et al. (2015, 316) describe their device's pressure feedback as feedback that has a wide range of attention capture. If feedback is adjusted continuously, this can be employed to grasp the user's attention slowly. The feedback is compared with somebody grabbing the user by the hand, and this can be used even in intimate communication scenarios. If strong force is applied, this kind of feedback can be exploited to disable the movement of the user. A large variation in attachment positions is also one of the advantages of devices that exploit such pressure feedback. In their article, Becke et al. (2015, 316) point out that continuous pressure feedback is more pleasant than prolonged vibration, that is considered disturbing.

The area of pressure can also be point based, by strapping a small container straight to the desired area on the body. This can be used to indicate the point and the strength of the tactile sensation to the user. He et al. (2015) have an excellent example of the point based pneumatic feedback. He et al. (2015) made a *Pneumatic Armband* which has five air pouches, that can be inflated and deflated separately. In their project, the idea was to explore new kind of tactile stimuli and the focus was on three different sensations: tapping, holding and tracing. The feedback is versatile, and by using pressure instead of vibration, the problem with negative responses after long exposure was tackled.

It is also possible to separate the container from the skin and put it into a chassis. This chassis and the pressurized container combination can then be used as a button (Gohlke, Hornecker, and Sattler 2016). Gohlke, Hornecker, and Sattler (2016, 309) tell that soft robotics enable expressive control and the users in their study describe soft materials as organic and natural. The greatest advantage of soft actuators over mechanical rigid structures is the generation of isotropic force output which is very much needed for the prototype of this thesis as well. Gohlke, Hornecker, and Sattler (2016, 314) write about a couple of disadvantages of soft materials. One of those is the lack of simple hardware for pneumatic systems and the control of valves and compressors. Another weakness is the speed and precision of soft materials. This could be fixed with higher pressure, but then the advantages of the soft materials are lost. Separate pressurized containers can also be employed separately from the chassis. Changing the pressure in the container and squeezing it will make the user feel the change in rigidness of the squeezable object.

Furthermore, it is possible to employ *controllable inflatable fabric* (Ou, Heibeck, and Ishii 2016). In their work Ou, Heibeck, and Ishii (2016) introduce *Thermoplastic Polyurethane (TPU)* coated textile as a material that can be employed in different situations. By choosing a right textile, the elasticity of the inflatable materials can be changed. With this kind of technology and materials it is possible to make *"pressure operated wearables"*, which could also serve as feedback receptors. On top of that, this technology enables precise feedback to the desired location.

As mentioned in the work *Pneumatibles* by Gohlke, Hornecker, and Sattler (2016), soft pneumatic actuators are still a quite new and unexplored technology. By making a novel prototype and using the technologies mentioned before, the work in this thesis will bring some insight on where the feedback could be used and how. The soft robotics with its "natural feel" and possibility for a longer use time without disturbance generated by the feedback are the main reasons why this kind of technology was chosen in this thesis.

2.5 Related works

The prototype in this thesis was greatly inspired by the articles cited in what follows. The controllers in the gaming industry are usually not researched and when new ones are made all the data is kept in secret for patent reasons. Subsections of this section consist of the works that are done for research purposes and give insight on possible technologies to be used in the further development of the controllers in gaming and technological industries. These articles talk about new kinds of feedbacks and parameters that are crucial in haptic controller development.

2.5.1 Impacto haptic controller

Lopes, Ion, and Baudisch (2015) present *Impacto*, a device to 'render' the haptic sensation of hitting and being hit in virtual reality. Impacto achieves its goal by decomposing the stimuli in two parts: the first part concentrates on the tactile aspect by tapping the skin of the impact area, and the second concentrates on retracting the motion of the limb that the applied force generates. The first aspect, the tactile stimulus, is achieved by using a solenoid that snaps to the skin at the time a fist meets the blocking arm of the user. The retracting motion is achieved by applying a small electric shock to the retracting muscles.

Lopes, Ion, and Baudisch (2015) also write about all the applications they used this controller in. The simplest case was blocking a punch with an arm, but later the controller was applied to punching, kicking and hitting a ball with a baseball bat. Lopes, Ion, and Baudisch (2015) concluded that experiencing impact in virtual environment is possible and achievable in light format by decomposing it in two parts: tactile and retraction force. They also showed a proof-of-concept module in three VR applications and each of the applications demonstrated the variety of Impacto in haptic sensations.

A contribution of this article is the new way to utilize already existing simple technology to create a haptic controller. The idea for controller in this thesis is also similar in exploiting technology that have not been used before and in creating a new HCI that could be beneficial in the future.

2.5.2 SoEs gaming controller for immersive interaction

Chen et al. (2016) present an attachable augmented haptic device for HTC Vive called *SoEs (Sword of Elements)*. The idea of the device is to enhance gaming experience and to increase immersion for a first-person game, while simultaneously keeping the cost of the device as low as possible. The device was made because virtual reality lacks immersive tactile feedback regardless of HTC Vive having vibrotactile feedback. "The idea was to simulate the haptic feedback of player manipulation in the immersive environment such as striking while the iron is hot which the player could feel the heat and reaction force" (Chen et al. 2016).

The playable application for this device consisted of two different gamemodes, making the arrowheads for the bow and shooting intruders with the bow. "The Blacksmith: Environment" unity package was employed in a creation of the game that Chen et al. (2016) called "The ancient maker". In the first game mode, where a player had to make arrowheads, heat could be felt when an arrowhead was heated and hit with a hammer. When arrows were quenched, the vapour could be felt as heat and wind. Polishing and striking arrowheads also generated vibrotactile feedback. In the second mode, where the player shoots the arrows, the continuous reaction force could be felt when pulling an arrow, and a feel of a gust of wind was generated during release.

In this work, an attachment was made and it was also briefly described. It consists of the HTC Vive controller, a motor module for additional vibrotactile stimuli, electronic fans to get a feeling of the wind on the hands, and of a thermal module to get thermal feedback from virtual hot iron. The paper has no conclusions on how the device worked and whether there were any problems. Neither does it explain whether it actually enhanced the immersion and experience of the game, but there was discussion about future work and it is believed that this project is in its alpha stage. Real research appears to start from this point on.

This particular controller attachment implements the same idea as in this thesis. To concentrate on enhancement of the HCI communication, it is easier to use already available virtual environments and controllers for the position sensing. The idea with a cost efficient product was also considered while planning this work, because of the already high costs of existing consumer VR products.

2.5.3 Pneumatibles - Pneumatic button

Gohlke, Hornecker, and Sattler (2016) concentrate on exploring what they call *pneumatibles*. As Gohlke, Hornecker, and Sattler (2016) say in their article, the *pneumatibles* are "interactive, pneumatically driven actuator/sensor elements, made from pliable materials and inspired by soft-robotics principles", in another words, a sensor or actuator that on top of handling input is able to produce pneumotactile feedback. The article presents a pneumatible button, but the main focus of the article is on the technical parameters of the pneumatibles. The technical parameters consist of air pressure in the pneumatibles, material properties, the dimensions of the presented device, and of actuation sequences. The article also goes deeper in other technical parameters such as to the rigidness of the material, minimum and maximum pressure levels, and volume on air chamber in different situations (Gohlke, Hornecker, and Sattler 2016, Table 1.)

The Pneumatible introduced in the article is a button. A simple button consisting of a single chamber was chosen because it allows a systematic investigation of the structural design parameters. The single chamber solution helps to investigate all the parameters that influence the behaviour and tactile characteristics of the pneumatible, and to remove all the unnecessary variables that are related to multiple chamber solutions. In conclusion, Gohlke, Hornecker, and Sattler (2016) contributed to the pneumatibles by designing a *control system for pneumotactile applications* and by providing a basic list of parameters, as mentioned briefly before, that need to be considered when designing similar interfaces.

In case of the prototype in this thesis, pneumatibles have a significant role as a basis or as a starting point to which HTC Vive and different kind of air chambers will be added. Gohlke, Hornecker, and Sattler (2016) state that "pneumotactile actuators such as graspable or body-mounted pneumatibles deserve special attention and their control parameters have to be evaluated separately". On top of things that were mentioned earlier, the priority of implementation of pneumatible device in this thesis is the game aspect, and consequently, the theoretical part of the research is not profound in some parts.

3 Prototype

The prototype made for this thesis consists of three hardware parts: HTC Vive, Arduino, and additional external I/O. It also contains two software parts: Arduino I/O controlling code and code on top of the Unity game engine.

In this chapter, every part of the prototype will be discussed and the reasoning behind every decision will be brought up. The idea of this prototype changed during the process, and the final decisions and possible improvements will be discussed in the last section of this chapter.

3.1 Hardware

The controller prototype connects to an existing HTC Vive VR setup (HMD, base stations, controllers). Additional parts are needed to apply pneumatic feedback and control for the prototype. The self-made part of the prototype consists of a chamber, a valve for regulating pressure, a digital pressure meter, and, as a control logic, Arduino Uno Rev3. All these mentioned parts are for modeling pneumotactile sensation and feedback.

This section explains the idea behind each part, why the parts are needed, and which role each of them has in the final product. Basic layout of the additional hardware can be seen in Figure 1.

3.1.1 HTC Vive

The prototype was built by employing HTC Vive. The HTC Vive controller is made for wireless interaction with virtual environments. The controller includes 24 sensors for tracking position, a multi-function trackpad in front, dual-stage trigger behind, and an HD haptic feedback actuator (HTC Corporation 2011-2019). The base stations enable wireless tracking of the controllers and the HMD. The stations are usually placed in the corners of the room. While the distance between stations can be up to eight meters, the manufacturer suggests to keep the stations 5.5 meters from each other, resulting in a four by four meters working area. The base stations operate by sending alternatingly IR sweeps horizontally and verti-

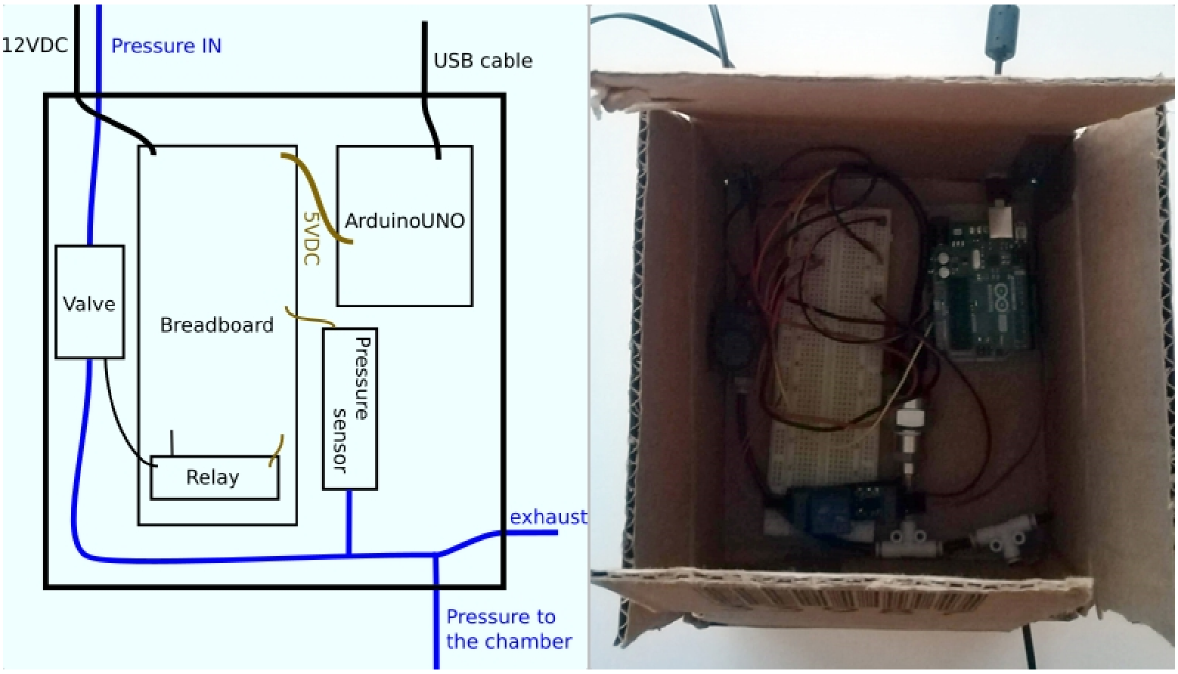

Figure 1. Schematic drawing of components in the control logic (left). Photo of the control logic (right).

cally (Niehorster, Li, and Lappe 2017, 3-4).

HTC Vive was chosen because the author had previous experience with using and programming software for the device. HTC Vive was also chosen because it was available and it is one of the common VR devices according to many reputable websites[1]. In this project, HTC Vive was used twice in different applications. First it was employed as meant by the manufacturers and by utilizing its own HD vibrotactile feedback. The second time the controllers were employed as tracers for the hand to enable movement of the arms inside the virtual environment. In the second case, an alternative device called *Vive Tracker* could have been employed to track the location of the hands, but it was not available for this project, and the controllers worked well for the purpose.

1. PCMag, TechRadar, Wareable

3.1.2 Additions

Additional parts employed in the prototype were: Pressure Transducer Sensor[2], electric valve[3], relay[4] for controlling the valve, and a basic self-made chamber.

The pressure sensor was chosen because its operating voltage fits well Arduino's five volts output and the operating pressure range is sufficient for this project. Finding a suitable valve for the project was a bit challenging. The search did not reveal any valve that could operate with five volts, so the operating voltage criteria was risen to 12 volts. Luckily, the two way valve was found that fit. The last part of the prototype was a chamber. Because TPU materials turned out to be expensive and difficult to employ, a rubber balloon and jeans fabric combination was chosen for the tests.

3.1.3 Control logic

In this thesis, all the additional components were controlled by Arduino. The device was chosen because it had all the features that were needed: possible communication with the computer (serial port), an analog input (pressure from the sensor), ability to control relays (digital output) and also property to power additional components with 5V (pressure sensor and relay). In this prototype, the connections were connected through the Breadboard. 5VDC came from Arduino and 12VDC came from a transformer connected to the power grid. Because the valve operates on 12VDC it could not be operated straight by Arduino, which is why the relay was used. The relay was connected to the 5VDC and the operational signal came from DigitalOutput pin "5". The other side of the relay was connected to regulate power coming to the valve. The connections of additional components to the Breadboard can be seen in Figure 2.

Operating the pressure sensor was a bit easier. The pressure sensor has three cables: plus (red), ground (black) and analog output (yellow). The output cable of the pressure sensor was connected to the "AO" pin. The received data value were varying from 100 to 150, and they were later translated into bars in the software for easier understanding of pressure

2. Pressure sensor
3. Two way normally closed electric solenoid air valve
4. Subminiature high power relay by songle (SRD-05VDC-SL-C)

values.

All the communication with a computer software and code uploading to the device were handled by serial port, which was connected to the computer via USB.

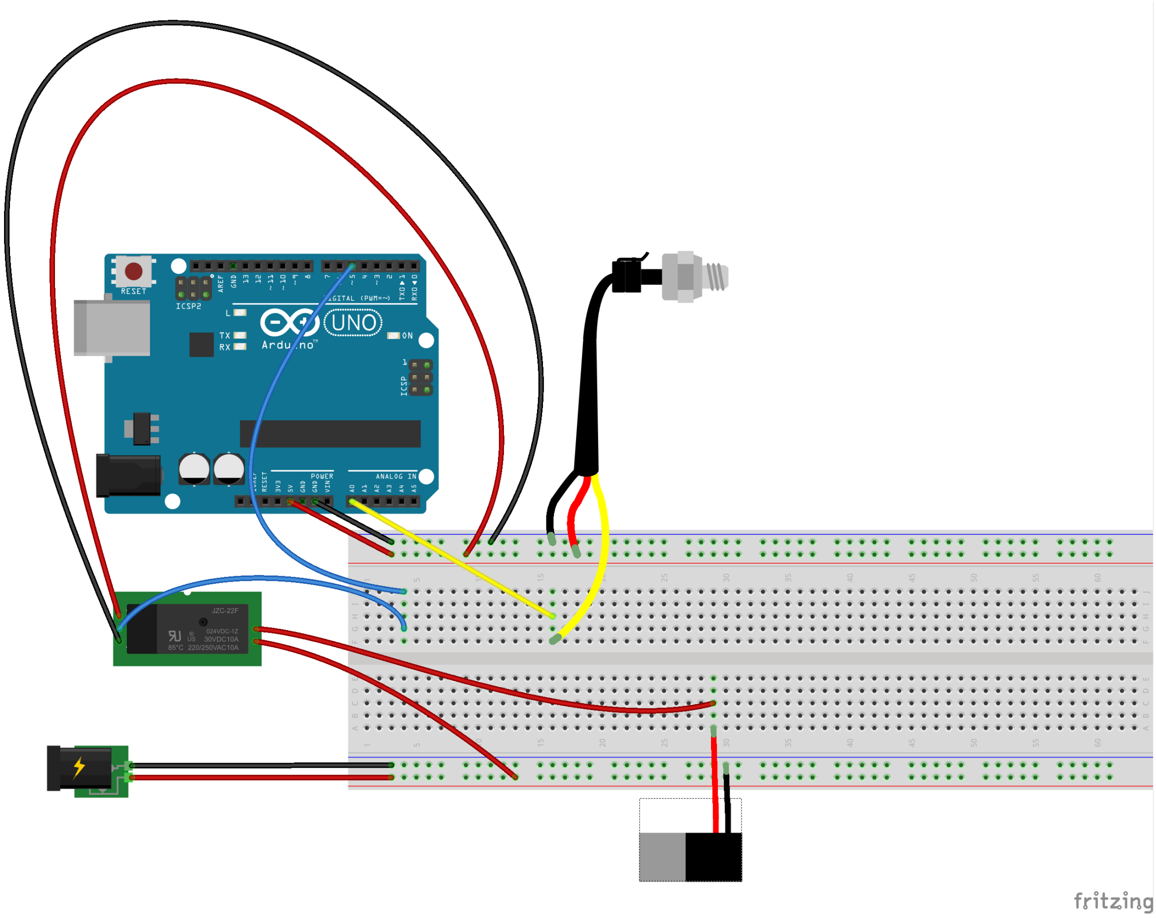

Figure 2. Electrical connections of the prototype made in Fritzing.

3.2 Software

To make the prototype react to changes in a virtual environment and to interact with it, an environment has to be created in the first place. The software for this solution consists of two codes that interact with each other. The code that is run in Arduino is mainly responsible for reading pressure data and operating the valve. On the Unity side, there is a testing environment and logic, that sends object data to Arduino. In this section, the software parts are discussed and the communication between different parts is explained. Parts of the software that require more attention are discussed more closely.

3.2.1 Arduino

The main function of the Arduino control logic was to let air flow into the chamber and close the valve when the pressure inside the chamber was correct. When making this feature, it was noticed that the value from the pressure sensor needed to be converted to bars with the following formula:

$$\text{bar} = \text{zeropoint} + \text{avrgPress} * \text{factor} \tag{3.1}$$

in which "zeropoint" value was calculated as -1.42307 and factor value was 0.01499. After the conversion to bars was ready and the maximum pressure could be set, there was a need for a way to communicate with the software on the computer. By following a tutorial made by Alan Zucconi [5], a program in Arduino was made to receive the maximum value of pressure and to send a confirmation message back to computer via the serial port. The whole code can be seen in Appendix A of this thesis.

3.2.2 Unity

For testing the prototype, a game was made. The game is a simple sorting game where the player has to separate balls into different holes by their attributes, in this case rigidness of the object. In the beginning of the game, the player is presented with a tutorial, where levels of rigidness are presented as colors of the balls: green is soft and red is hard. Balls are then put in the holes that are the same color. After nine sorted balls, the "real" game starts and the player has to sort 20 balls as well as possible.

After a full game (total of 84 balls) is played, the controller is changed, vibrotactile feedback is disabled, and HTC Vive controllers are placed on backside of the arm of the user. Additions to the prototype are added and the only thing that is changed in the game code is that Unity is now communicating with Arduino. Unity code is programmed to send maximum pressure values depending on the rigidness of the balls, and it then receives the confirmation message sent by the Arduino. Unity code can be seen in Appendix B.

5. https://www.alanzucconi.com/2015/10/07/how-to-integrate-arduino-with-unity/

3.3 Progression

In the beginning phase, when the prototype was just an idea, it was different. The plan was to make a prototype that would be mobile and wireless. But as always in the projects, modifications have to be made to match the original schedule. The original prototype idea had a different pneumatic system that was planned to be fixed on the arm of the user, and all the electronics would be powered by a battery fixed on the hip with a belt. The planned pneumatic system had two air chambers: one for squeezing the virtual object and another one was the one that should have supplied the first one with pressure. There was a couple of ideas for the supply chamber, for example one idea was that the chamber would be solid, bottle-like chamber with a cylinder inside pushing and pressurizing air. This idea was left behind because such component was not found and the cost of the product would probably have exceeded the budget. The second idea was that the supply chamber would be squeezed and this would generate pressurized air. The idea was left behind because such solution would take a lot of time to realize, and buying all the components (motors, drawstrings and chassis for squeezer) would be a gamble because there was no certainty that this implementation would generate enough pressure.

Because of the decision that pressure will be drawn from the compressor, there was no possible way to make the prototype wireless, and thus changes to the electrical circuit were made. The initial plan was to employ a Raspberry Pi Zero as the control logic that uses Bluetooth for transmitting, but because there was eventually no need for wirelessness, the Arduino was ultimately chosen. Initially there was an idea that all electrical components would work on 5V voltage and for the power the prototype would use 10Ah phone charging battery. The first problem occurred when there was no valves found that could operate on 5V or they were too expensive. In the end, a valve was found that operated on 12V and that meant that a separate power supply was needed, because Arduino has only 5V and 3.3V outputs. Luckily, the problem was solved by buying a transformer from 220VAC to 12VDC and so even the Arduino could have been powered from the same source. Control of the valve also required a relay that could be powered and controlled by the Arduino.

4 Evaluation of the pneumotactile controller

The goal of this study was to find an alternative and simple approach to emulate the rigidness of objects in a virtual environment. The success of this was tested with a reaction to the pneumotactile feedback. The main question was whether the emulation is as precise as desired, and whether it serves its purpose better than the more commonly employed vibrotactile feedback.

4.1 Risks and ethics

Because the object of the testing is a pressure operated self-made prototype, there is always some risks that can not be totally eliminated. In case of this particular prototype, there is a possibility, that the pressure chamber's inner layer rips apart due to the friction between the layers and it creates a loud noise. Ripping of the inside layer may also cause some pain to the user's hand holding the prototype.

To reduce the risks caused by the prototype, a risk analysis was made and the following practices were implemented: To protect the ears, earmuffs or protective headphones were used. Gloves are also used to protect from possible inflicted pain that could be felt in case of inner layer ripping.

There is a minor risk in testing, but it is believed that effects of them actualizing is extremely low. There is no other way to test the prototype and get the needed data of the feeling and experience of the pneumatic feedback.

4.2 Considerable issues in group testing

To test a new product, usually there is a group of people that do the initial testing and evaluation of the product. In case of the prototype in this thesis, the idea was the same, and full preparations for a group testing were made. Also, the test game was designed to provide measurable and quantifiable HCI performance data from user tests in accordance to Section 2.2. Because of the involved risks and also other limitations, not all within the author's control, eventually the testing had to be regretfully omitted from the scope of this thesis. As a back-up plan, it was decided that it is better to test the prototype locally first and build good basis for the future research.

There are a couple of things that are good to keep in mind when testing a controller. One thing is that the prototype should be compared to an existing product so that the need of the newer one can be justified. As there will be other people involved, on top of physical risks information security has to be addressed so that participants of the testing can feel themselves safe and be sure that their personal information is safe.

Of course it is important that group of participants in the testing is as large and as diverse as possible. Large group of participants enables diverse data, like first impression of the products and someone could notice something that no one else reacted to. Diversity of the age in the group is also one thing to consider. It is likely that the consumer of the product is a younger person, but the one that buys the product is a parent or an adult, and for them the purchase have to be justified and necessary.

Because the tests could not be performed with a group of people, the research lacked in thorough data in initialization of the prototype and first time experiences. There is hope that this research will inspire people to become interested in the field and possibly develop this technology and test it with large group.

4.3 Methods

The study contained two parts, which were now tested and recorded by only the author, because the actual group tests had to be postponed. The first controller in testing was the HTC Vive controller and the rigidness of an object was modelled by exploiting the vibrotactile feedback. A hands-on test was concluded in three parts. In the first part, soft material was modeled with 10% of maximum vibration power, medium rigidness with 50%, and hard material with 100%. In the second part, the powers were 50%, 75% and 100 %, and in the last part 60%, 80% and 100%. After testing with controller, information about the game scores, feelings and other remarks were written down as would have been done with each participant of the planned group test.

After testing with reference controller, it was changed to the self-made prototype and the same tests were conducted, but this time by exploiting pneumotactile feedback. After the test, the same questions were answered again as in the first time with vibrotactile controller.

The final questionnaire consisted of questions about which of the controllers did emulate rigidness better, and if there were any other observations that were made while testing either of controllers.

4.4 Results

The results were gathered by testing two controllers in the VR. All the tests were conducted by the author only because of the time restrictions. Desired result was to find an optimal solution to feeling rigidness of the objects in the virtual environment.

The tests in this thesis were conducted in two parts as planned. The first part was experiencing vibrotactile and establishing basis for testing and another part was testing self-made prototype and evaluating its feasibility to fulfill the task.

This section provides the results of the testings and comparison of the controllers.

4.4.1 Vibrotactile feedback

Testing started with the vibrotactile feedback. The results were a bit odd and did not fully reflect what the hypothesis was. The points from first three laps were: 18/20, 12/20, 16/20. It was not odd that the results went worse because of how close in the end the feedbacks were, but it was interesting to see that second round performance went worse than the last one. This phenomenon could have happened because some of the feedbacks were more distinguishable than others, for example the weakest one (soft object) most certainly went always right. For that reason if other rounds had more soft balls it would be easier to get closer to full score.

Other things that probably affected the testing were numbing and sound caused by the rumble pack. Because the rumble pack that emits the vibrotactile feedback is located under the touchpad on the front side of the controller, it was decided that it is optimal to hold thumb on touch pad where the feedback is strongest, to maximize the experience. The problem with getting whole feedback in one spot is that the receiving finger started getting numb in the end of the testing and so if the testing would carry on a little longer the results would probably decline. Also one thing that was noticed when testing, sometimes when making a decision on how strong the feedback was, it was easier to listen to the sound of the rumble pack and make decision upon sound, which is technically a totally different cue to rely on and could have affected the results.

The opinion on questions "how natural the feedback felt" and "did the feedback felt like rigidness of an object" is quite negative. The feedback did not feel as natural reaction to squeezing an object. As the author has used controllers a lot and is experienced in the gaming scene, it had to be admitted that it is possible to learn to "feel" the rigidness of the object with vibrotactile feedback but it is not an optimal situation in any way.

4.4.2 Pneunotactile feedback

The testing of pneumotactile feedback had to be moved to the next day because previous testing took more time than anticipated and the laboratory property closed earlier than was planned. The environment was set in the morning and testing was started straight away without testing again with vibrotactile feedback. The result points of the testing were: 20/20, 20/20, 15/20. The first round of testing had to be redone a couple of times because of technical problems, but the other rounds were done only once.

The results were quite surprising because the first two rounds went so well and even in the third round mistakes often were made when the feeling from the previous squeeze was forgotten. The problems that occurred with the prototype were mostly because of poor design and of course some mistakes were made because of the lack of experience with pneumotactile feedback. The reference to poor design means that the prototype had to be manually emptied every time between single squeezes because there were not enough valves and logic was not coded for the exhaust valve to operate. In the end, another design flaw was found regarding the first object in the game. For some reason the first object did not get the right pressure or it was not assigned at all. This kind of tactile feedback was new to the author even though the idea and build prototype were originals. Often when the chamber was filling with the air it was forgotten to concentrate on the feeling of the rigidness and it was more enjoyable to feel out the sensation.

Because of the way the pneumotactile feedback works, some adjustments had to be done. The one thing was pressures, and limits used in the testing. Because the chamber had to be emptied between squeezes, there was a minimum pressure which worked as a reset point. Minimum pressure was assigned at 0.1bar which means 20% of maximum pressure in this case. So on the first round testing with 10% pressure would be impossible. The decision was made to change the pressures to: first round 50%,75%, 100%; second round 60%, 80%, 100%; and third round 80%, 90%, 100%. Meaning that in the last round the difference in pressures were only 0.05bar.

In general, there were couple of things that did not feel as desired. One of the things was slow change in the pressures. In the video games usually everything happens very fast and there is no time to wait for one second to grab another object. Because of the implemented solution, the rigidness could be also calculated by wait time until the chamber was full, and the valve switched on with loud click. One problem in the beginning was that it was scary to squeeze the chamber harder to get a better feel, in fear of it blowing up. In the end it was noticed that muscles in the arm started to get sore because of all the squeezing.

The natural feel of the pneumotactile was on point. The only thing that was mildly distracting was that the form of the object in the virtual environment and shape of the chamber were different. When picking up different objects in the virtual environment, they really felt like their rigidness was different and they were quite easy to tell apart when testing for a while. The pneumotactile feedback really corresponded to the expectations. Now the next step is to think of right implementations for it, and to do more tests to get more precise data on its usefulness.

4.4.3 Final results

To summarise the testing process, it was fair and it is possible to compare pneumo- to vibrotactile feedback. There are always some important viewpoints that can not be addressed here as testing was solely done by the author but some general linings can be derived.

It is hard to compare controllers because one is developed with existing technology and by large company, and another is one man job in limited schedule. Even if it is kept in mind that the prototype was made specifically for this task and the HTC Vive controller was made for universal use. For this specific task the prototype worked better in portraying the rigidness and telling objects apart. HTC Vive felt better in speed and ease of use in the author's opinion. It would be interesting to try the prototype on people that have not used any VR platforms before and to see if the presented prototype would be more natural to use and easier to learn than the HTC Vive controller. Unfortunately, that is the one question that can not be answered as the producer of the prototype and active and already experienced user of VR and HTC Vive.

5 Conclusions and Future Directions

What can be concluded from the work done for this thesis is that there is a room to grow in exploring HCI for VR, and solutions for enriching HCI can be made using inexpensive materials. The presented haptic controller is not yet a substitute controller for controllers on the market but it could possibly be in the future, and glimpses of its potential can be seen.

The main question of this thesis was answered by showing that by exploiting pneumotactile feedback it is possible to portray rigidness of objects naturally and more intuitively than with the vibrotactile feedback.

Even though the prototype could be called a success, proving the concept of inexpensive haptic controllers to be a realistic possibility, it is still far away from a complete product or even an all around working controller. The current prototype is not mobile, does not have any functionality other than changing rigidness, and its effect has so far been tested only by one user. The basic work is done on this topic and the "real" work is still to be done, so if the HCI or more technical view on gaming scene is in the reader's scope of interest, this could be an interesting thing to look into.

If the only thing that is desired is to change the prototype for it to work smoothly and its operation to be totally automatic, then following things need to be considered. At this moment, the exhaust for the prototype was realised manually with a hand-operated valve. Such solution is not optimal and it weakens the immersion because of the involvement of the person from the "outside". This problem can be resolved by attaching a second valve to the prototype and make it so that it opens when the object is released from the hand.

Another improvement on immersion could be done in picking up and releasing an object in the virtual environment. Currently objects are snapping to the "hand" when object is near the hand and the object is released when it is near the hole that it can be put in. The picking up of the object can be realised by the pressure sensor detecting when object is squeezed (pressure rises) and then attaching the object to the hand. Releasing then needs to be done by a button that gets pressed when the chamber of the controller is released from the hand.

Regarding HCI performance testing, the problem with remembering the feeling of the right pressure was often an issue, therefore other testing solutions could be presented. For example, it could be possible to make a game mode where there are many objects on the table that have different rigidness and they have to be organised from soft to hard. Because the objects would be in front of the user all the time, it would be possible to compare them to each other as many times as needed before making a final decision of the order. That way the user could concentrate better on the current feeling and not on the remembering of past feelings.

The last thing that could be tested, is reactions of the users when the controller reaches its full potential. For controller solution reaching its full potential it needs visual and audio cues. Objects could deform when squeezed and release some kinds of sounds in the process.

Bibliography

Becke, Dennis, Henning Pohl, Michael Rohs, Maximilian Schrapel, and Eugen Wagner. 2015. "Wrist Compression Feedback by Pneumatic Actuation". *Proc. CHI'15:* 315–318. doi:10.1145/2702613.2725427.

Boas, Yuri Antonio Gonçalves Vilas. 2013. "Overview of Virtual Reality Technologies". *Mms.Ecs.Soton.Ac.Uk.* https://pdfs.semanticscholar.org/4214/cb09e29795f5363e5e3b545750dce027b668.pdf.

Card, Stuart K., Thomas P. Moran, and Allen Newell. 1983. *The Psychology of Human-Computer Interaction.* Hillsdale, New Jersey: Lawrence Erlbaum Associates, Inc. ISBN: 0-89859-243-7.

Chen, Yang-Sheng, Ping-Hsuan Han, Jui-Chun Hsiao, Kong-Chang Lee, Chiao-En Hsieh, Kuan-Yin Lu, Chien-Hsing Chou, and Yi-Ping Hung. 2016. "SoEs: Attachable Augmented Haptic on Gaming Controller for Immersive Interaction". *Proceedings of the 29th Annual Symposium on User Interface Software and Technology - UIST '16 Adjunct:* 71–72. doi:10.1145/2984751.2985707.

Gohlke, Kristian, Eva Hornecker, and Wolfgang Sattler. 2016. "Pneumatibles: Exploring Soft Robotic Actuators for the Design of User Interfaces with Pneumotactile Feedback". *Proceedings of the TEI '16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction - TEI '16:* 308–315. doi:10.1145/2839462.2839489.

Haans, Antal, and Wijnand IJsselsteijn. 2006. "Mediated social touch: A review of current research and future directions". *Virtual Reality* 9 (2-3): 149–159. ISSN: 13594338. doi:10.1007/s10055-005-0014-2.

Haller, Michael, Christoph Richter, Peter Brandl, Sabine Gross, Gerold Schossleitner, Andreas Schrempf, Hideaki Nii, Maki Sugimoto, and Masahiko Inami. 2013. "Finding the right way for interrupting people to posture guidance". *Proceedings of the 13th IFIP TC 13 international conference on Human-computer interaction:* 1–18.

HaptX Inc. 2018. “Industrial-grade haptic technology”. https://haptx.com/technology/.

He, Liang, Cheng Xu, Ding Xu, and Ryan Brill. 2015. “PneuHaptic: Delivering Haptic Cues with a Pneumatic Armband”. *Proceedings of the 2015 ACM International Symposium on Wearable Computers - ISWC ’15:* 47–48. doi:10.1145/2802083.2802091.

HTC Corporation. 2011-2019. “Controller”. https://www.vive.com/eu/accessory/controller/.

Israr, Ali, Siyan Zhao, Kyna McIntosh, JaeKyun Kang, Zachary Schwemler, Eric Brockmeyer, Mark Baskinger, and Moshe Mahler. 2015. “Po2: Augmented Haptics for Interactive Gameplay”. *ACM SIGGRAPH 2015 Emerging Technologies:* 21:1–21:1. doi:10.1145/2782782.2792489.

Lee, Yongkwun, and Dongseok Ryu. 2008. “Wearable haptic glove using micro hydraulic system for control of construction robot system with VR environment”. *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems:* 638–643. doi:10.1109/MFI.2008.4648016.

Lopes, P, A Ion, and P Baudisch. 2015. “Impacto: Simulating physical impact by combining tactile stimulation with electrical muscle stimulation”. *UIST 2015 - Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology:* 11–19. doi:10.1145/2807442.2807443.

Merdenyan, Burak, and Helen Petrie. 2015. “User Reviews of Gamepad Controllers: a Source of User Requirements and User Experience”. *Proceedings of the 2015 Annual Symposium on Computer-Human Interaction in Play - CHI PLAY ’15:* 643–648. doi:10.1145/2793107.2810332.

Niehorster, Diederick C., Li Li, and Markus Lappe. 2017. “The accuracy and precision of position and orientation tracking in the HTC vive virtual reality system for scientific research”. *i-Perception* 8 (3): 1–23. ISSN: 20416695. doi:10.1177/2041669517708205.

Ou, Jifei, Felix Heibeck, and Hiroshi Ishii. 2016. "TEI 2016 Studio: Inflated Curiosity". *Proceedings of the TEI '16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction - TEI '16:* 766–769. doi:10.1145/2839462.2854119.

Ramsamy, Priscilla, Adrian Haffegee, Ronan Jamieson, and Vassil Alexandrov. 2006. "Using haptics to improve immersion in virtual environments". *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 3992 LNCS:603–609. ISSN: 16113349. doi:10.1007/11758525_81.

Suhonen, Katja, Kaisa Väänänen-Vainio-Mattila, and Kalle Mäkelä. 2012. "User experiences and expectations of vibrotactile, thermal and squeeze feedback in interpersonal communication". *Proceedings of BCS HCI 2012:* 205–214. http://dl.acm.org/citation.cfm?id=2377916.2377939.

Appendices

A Arduino code

```
#include <SoftwareSerial.h>
#include <SerialCommand.h>

SerialCommand sCmd;

//relays pin number
int relay = 5;

// analog pin number for pressure metering
int pressPin = A0;

//next 4 initialized variables are for pressure handling
float pressure = 0.0f;
float bar= 0.0f;
float maxpress = 0.5f;
float avrgPress;

//count variable is for average pressure calculating
int count = 0;

float avrgPress;
void setup()
{
  pinMode(relay, OUTPUT);
  pinMode(pressPin, INPUT);

   Serial.begin(9600);
   while (!Serial);

   //What happens when "ECHO" or "PING" commands are received
  sCmd.addCommand("ECHO", echoHandler);
  sCmd.addCommand("PING", grabHandler);
}
```

```

void loop()
{
  //Add pressure value on top of previous each loop count
  pressure = pressure + analogRead(pressPin);
  count ++;

  //when there is 10 pressure values stacked,
  //calculate average pressure and reset state
  if(count >= 10)
  {
     avrgPress = pressure / 10.0f;
     pressure = 0.0f;
     count = 0;
  }

  //transform value to bars
  bar = -1.42307 + avrgPress * 0.01499;

  //Read serial if there is an input
  if (Serial.available() > 0) sCmd.readSerial();

  //if pressure goes over wanted value then valve closes
  if (bar > maxpress)
  {
    digitalWrite(relay, HIGH);
  }

  //open valve and let pressure flow in if
  //pressure inside is less than 0.1 bar
  if(bar < 0.1f)
  {
    digitalWrite(relay, LOW); //valve is open
  }
}

//how does arduino handle different different "ECHO arg" commands
```

```
void echoHandler () {
  char *arg;
  arg = sCmd.next();
  if (arg != NULL)
  {
    if(String(arg) == "1")
    {
      maxpress=0.25f;
    }
    if(String(arg) == "2")
    {
      maxpress=0.35f;
    }
    if(String(arg) == "3")
    {
      maxpress=0.5f;
    }

    //send value of max pressure and arg back to Unity as "confirmation"
    String msgdata = String(maxpress) + " I " + arg + " I ";
    Serial.println( msgdata );
  }
  else
    Serial.println("nothing to echo");
}

//if PING message is sent and pressure is higher than max pressure (
    object grabbed)
void grabHandler ()
{
  if(bar > (maxpress + 0.1f))
    {
     Serial.println("squeezed");
    }
}
```

B Unity code

```
/* Parts of this code are copied from:
 * ArduinoConnector by Alan Zucconi
 * http://www.alanzucconi.com/?p=2979
 */
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using System;
using System.IO.Ports;

public class ArduinoBridge : MonoBehaviour {

    private SerialPort stream;
    public bool msg = false;
    public bool grab = false;
    public bool grabbed = false;
    private float mass;
    BallTriggeringScript bts;

    //opening of the serialport to arduino
    public void Open()
    {
        // Opens the serial port COM4 with baud rate 9600
        stream = new SerialPort("COM4", 9600);
        stream.ReadTimeout = 50;
        stream.Open();
    }

    //Use this function to read from Arduino
    public string ReadFromArduino(int timeout = 0)
    {
        stream.ReadTimeout = timeout;
        try
        {
            return stream.ReadLine();
        }
```

```
        catch (TimeoutException e)
        {
            return null;
        }
    }

    //This function is for a response waiting time from arduino
    public IEnumerator AsynchronousReadFromArduino(Action<string>
callback, Action fail = null, float timeout = float.PositiveInfinity)
    {
        DateTime initialTime = DateTime.Now;
        DateTime nowTime;
        TimeSpan diff = default(TimeSpan);

        string dataString = null;

        do
        {
            try
            {
                dataString = stream.ReadLine();
            }
            catch (TimeoutException)
            {
                dataString = null;
            }

            if (dataString != null)
            {
                callback(dataString);
                yield break; // Terminates the Coroutine
            }
            else
                yield return null; // Wait for next frame

            nowTime = DateTime.Now;
            diff = nowTime - initialTime;
```

```

        } while (diff.Milliseconds < timeout);

        if (fail != null)
            fail();
        yield return null;
    }

    // Use this for initialization
    void Start () {
        //Opening port to arduino
        Open();
        //initializing ball mass
        bts = gameObject.GetComponent<BallTriggeringScript>();
    }

// Update is called once per frame
void Update () {
        mass = bts.ballMass;

        //if there is message to be sent then send mass of the ball to
 arduino via "ECHO arg" command
        if (msg == true) MessageTest(mass);

        //if scene is changed close the stream
        if (bts.sceneChange == true) stream.Close();
}

    //Sends message, waits for reply and sets message state to false
    void MessageTest(float ballmass)
    {
        //Sending arduino a message "ECHO arg"
        if (Mathf.Approximately(ballmass, 1.0f))
    WriteToArduino("ECHO 1");
        if (Mathf.Approximately(ballmass, 1.25f))
    WriteToArduino("ECHO 2");
        if (Mathf.Approximately(ballmass, 1.5f))
```

```
    WriteToArduino("ECHO 3");

        //Use this to wait for response from arduino
        StartCoroutine
        (
            AsynchronousReadFromArduino
            ((string s) => Debug.Log(s),         // Callback
                () => Debug.LogError("Error!"), // Error callback
                10000f                          // Timeout (milliseconds)
            )
        );

        //reset message state
        msg = false;
    }

    //Use this function to write to Arduino
    public void WriteToArduino(string message)
    {
        stream.WriteLine(message);
        stream.BaseStream.Flush();
    }
}
```