The Design of Wearable Fingertip Haptic Device That Can Exert Various Softness

---

Gorkem ANIL AL

Supervised by: Dr. Antonia Tzemanaki and Prof. Sanja Dogramadzi

A dissertation submitted in partial fulfilment of the requirements of the University of Bristol and the University of the West of England for the Degree of Master of Science

Module Number: UFMED4-60-M

Faculty of Mathematics and Engineering, University of Bristol

Faculty of Environment and Technology,the University West of England, Bristol

October 22, 2017

ABSTRACT

This thesis presents a fingertip haptic device that can exert various levels of stiffness. The designed device provides softness haptic feedback via an inflated material. This makes the device different from other devices that provide force feedback. The device was developed for virtual reality applications-and teleoperated applications. Such devices can especially benefit surgeons in a robot-assisted surgical setting. Rendering the level of softness of different materials provides more realistic haptic perception. This can help the surgeons distinguish what type of organ or tissue they manipulate. Fingertip haptic device can also be beneficial for surgical training as surgeons can employ techniques used in open surgery where use of hands and their haptic sensation is possible.

Acknowledgements

I would like to express my special thanks of gratitude to my supervisors Dr. Antonia Tzemanaki, and Prof Sanja Dogramadzi for the useful remark, guidance and patience through this master thesis .

I would also express my gratitude to my parents and friends for their support.

Contents

1 Introduction

1.1 Overview

Minimally invasive surgery (MIS) offers advantages over conventional operations such as reducing blood loss, tissue trauma, pain and recovery time. In conventional open surgery, the surgeons use their hands or surgical instruments to interact with the internal tissues through a large incision. Although MIS brings advantages, it also has some disadvantages; the small incisions that are used during operation limits the degrees-of-freedom (DOF)and the dexterity of the surgeon. Detection of the forces between surgical tools and tissues in MIS is significant for better performance and timing (Abushagur et al. (2014)). While surgeons take advantage of the sense of touch in open surgery, and force feedback which is exerted by medical device in manual MIS, current robot-assisted (RAMIS) instruments do not provide haptic feedback to the surgeons when they interact with tissues. In the last decade, crucial robotic innovations emerged for medical applications which integrated with computers, are used for local surgery and tele-surgery, and enhancing MIS and RAMIS (Speich & Rosen (2004)).

The idea of the master-slave robotic mode is that robots (slave) are controlled by surgeons (master). The state of the art in RAMIS is the da Vinci system,and it is designed based on such a configuration Speich & Rosen (2004). They enhance the manipulation accuracy of surgeons, and tool-tissue interaction. However, it does not provide haptic feedback which might cause tissue trauma and damage Abushagur et al. (2014). Generally, as an alternative to haptic feedback, 3D visual feedback and auditory feedback are used to provide pseudo haptic feedback Lim et al. (2015).

Some of the master devices used in surgical robot systems of the literature exert kinesthetic and tactile forces to surgeon's fingers or hand. PHANTOM is one of the frequently used master devices. It provides highly accurate position control, and force feedback. However, it is a desktop device and requires cognitive manipulating training. Instead of this kind of device, hand exoskeleton hardware and fingertip haptic devices can be used by surgeons. These can decrease the training duration for surgery because they are worn and can mimic the hand movements of the surgeons Van der Meijden & Schijven (2009).

Haptic feedback and VR can be implemented for medical training and clinical practice. Some, haptic training possibilities are robotic surgery, anatomy learning and diagnostics (tumor diagnostic). Moreover, some studies support that collaboration of haptic feedback and VR simulators would be valuable for surgeon training Van der

Meijden & Schijven (2009). For instance, Lemole Jr et al. (2007) developed a virtual reality platform that includes real-time haptic feedback. for neurosurgical education. A head tracking protocol can provide realistic virtual environment as the user sees his/her own hand holding a virtual catheter. The system reads position and orientation of the catheter. When the collision happens between catheter and imported 3D polygonal isosurfaces which symbolises skin, bone or brain, the haptic stylus (Sensable Technologies, Woburn, MA) generates force. Lemole Jr et al. (2007) mentions that neurosurgical education requires long time hand-on training. With the help of computer based technology, and haptic devices, realistic neurosurgical operations can be operated.

Based on the need to improve surgical systems to be more efficient and which will provide the surgeons with better economics, this project focuses on designing a fingertip haptic device includes VR and hand tracking. The aim of this device is that, the user can sense the various levels of stiffness of different objects in virtual environment. The device should create displacement change of a material that is attached on fingertip. When user touches objects in virtual environment, user should sense the softness variety of objects.

1.2 Aims and Objectives

This project aims to design a haptic interface to develop human robot interaction. The designed haptic device is tested in virtual environment. The device exerts force to the fingertip of the user when the user touches the objects on the virtual environment in order to emulate the stiffness of the material.

The project consists of the design of fingertip haptic device, linear motion system, fingertip motion tracking, and virtual environment application.

In this project the various kinds of haptic devices, hand-exoskeletons, force feedback methods were reviewed, and the new fingertip haptic device was developed. Relative to these information a pillow system was created to give the sense of different stiffness levels of various objects in virtual environment applications. Also, its fabrication, and the actuator system rack and pinion mechanism was explained.

Force feedback is exerted via servo motor using rack and pinion mechanism. Also, the indentation is controlled with rack and pinion mechanism. The air is transferred with syringe pump. Several motion tracking methods were reviewed for syringe pump mechanism. This mechanism changes the softness of the pillow. Both motors

are controlled by Arduino Mega 2560.

Motion tracking was created for the haptic device using IMU sensor. It was attached on the haptic device, and it provided compact design. The acceleration information was sent to virtual environment via Arduino to translate the cursor which symbolises the fingertip.

A 3D environment was prepared to simulate the cursor for users and objects. This work is about testing the interaction of the haptic device and virtual objects on the software. Unity was selected for this purpose. The virtual sense was transferred to physical environment using Arduino board. The serial communication between Unity and Arduino offers motor control and motion tracking. The rigid body, and collider components can be attached to objects in Unity, and they offer to execute the physical tasks.

1.3 Thesis Roadmap

Chapter 2 describes different types of haptic devices their drawbacks and advantages are mentioned. Also, for each type of device, various designs are demonstrated. Later, motion capture methods are explained to implement virtual environment. In the last section of this chapter, virtual environment applications are mentioned.

Chapter 3 presents the materials and methods that were used for the fabrication of the fingertip haptic device.Printing materials, actuating motors, sensors and software that were chosen explained.

Chapter 4 proposes the mechanism design of the project. First, the design of fingertip haptic device is explained with focus on of the platform, the fabrication of the pillow, and rack and pinion mechanism. Following this, the actuation of the pillow and the syringe pump design are explained. These two main mechanisms are assembled, and the characterization of the fingertip haptic device is created. In the last section, the virtual environment is presented.

Chapter 5 presents the experiment of the device. Ten different conditions are created from the tests of the device and they are conducted. Chapter 6 evaluates the result of experiment. Chapter 7 summarizes entire project and discusses further work.

2 Background Information and Literature Review

Haptic feedback refers to the perception of the sense of touch. Haptics can include tactile and kineasthetic feedback, although it can also refer to temperature perception among others. Tactile refers to identifying the shape or texture of an object, while kineasthetic refers to force (Tzemanaki (2016)). Force feedback systems transmit kineasthetic information, while tactile systems offer cutaneous sensations (Vander Poorten et al. (2012)).

Human hands have a complex skeleton structure and provide high dexterity and precise manipulation ability. Moreover, hands include high sensitive touch due to mechanoreceptors which are embedded in skin that help to sense vibration, stretching and cutaneous stimuli. Humans can also distinguish shape, stiffness, and weight of an object thanks to these mechanoreceptors (Achibet (2015*a*)). Thus, hands play an important role in human life, and they help us to do precision work. Haptic sense is vital for surgeons because they rely on touch in open surgeries, when they palpate the tissue. Haptics applications can be used in medical training and robotic surgery (Vander Poorten et al. (2012)).

In addition to haptic feedback applications for surgeons, haptics can improve patient rehabilitation. Generally, haptic feedback is applied via hand-exoskeletons to treat patients who have neuromuscular problems (Iqbal et al. (2014)). Hand based haptic interaction can consist of force feedback, and tactile feedback.

2.1 Haptic Devices

Devices that can provide force feedback to the fingers or hands of users can be utilized in order to transmit the simulated haptic information from a virtual environment (Srinivasan & Basdogan (1997)). It is seen that the early haptic devices include fewer degrees of freedom (DOF); a haptic doorknob includes 1 DOF, and it applies force relative to users' action (MacLean & Roderick (1999)). Another application is 2 DOF joystick which reflects the force for machining process (Balakrishman et al. (1994)). Object manipulation and touching a virtual object require different kinds of forces from wrist, whole hand and each finger. The majority of devices cannot provide all types of feedback and they have limitations. Thus, different force feedback devices have been developed those aiming at the arm, the fingers, desktop-base interfaces and exoskeleton type devices (Achibet (2015*b*)).

In this project, various force feedback methods have been investigated to design the new device.

2.1.1 Desktop Force Feedback Devices

The robotic arm type desktop devices that exert haptic feedback are highly popular for VR applications. The user can control the designed cursor in a VR environment, and the devices transmit the physical interaction to the user by applying force to their hands. For instance, the PHANTOM (**Figure 1b**) is a 3 DOF remotely controlled device which exerts precise force to the users' fingers (Massie et al. (1994)), and Falcon (**Figure 1a**) is also 3 DOF haptic device which was designed based on parallel robot configuration (Martin & Hillier (2009)). Although these devices provide compelling force, their prices are expensive for customers (Pacchierotti et al. (2017)).

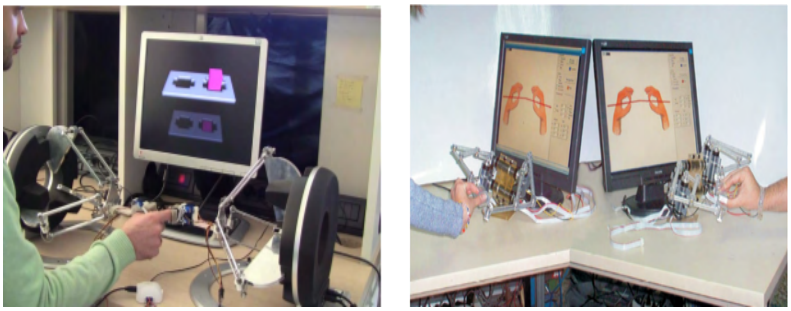
(a) (b)

Figure 1: (a) Novint Haptic Device (© Novint), (b) Phantom Haptic Device (© geomagic)

Desktop haptic devices have been designed for the specific action to support fingers. For example, MasterFinger-2 (**Figure 2b**) is a haptic interface that users can place their index and thumb. This device can provide 6 DOF for two fingers. It permits the user to grasp virtual objects in comfortable manner (Monroy et al. (2008)).

2.1.2 Wearable Haptic Devices

Desktop devices provide restricted workspace because they are grounded on a certain surface. Thus, to improve the workspace of users, wearable haptic devices have been developed. Some force feedback interfaces can be wearable by users such as HapticGEAR (**Figure 3a**). This device is attached the back of the user and exerts force pulling the finger of user (Hirose et al. (2001)). Moreover, FlexTorque (**Figure 3b**) haptic interfaces apply from virtual environments. A cable connects the arm and the wrist

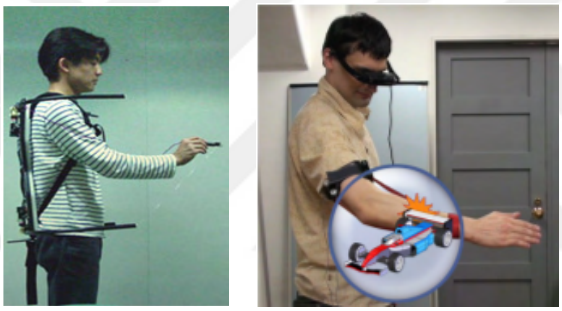


(a) (b)

Figure 2: Grounded haptic devices (a) Two grounded device provide haptic feedback to fingers (Pacchierotti et al. (2012)). (b) MasterFinger-2, two fingers are handled by two (Monroy et al. (2008))

.

of the user, and a motor is mounted on the arm. The motor pulls the cable to exert force on the wrist (Tsetserukou et al. (2010)).



(a) (b)

Figure 3: Human-Scale Haptic Devices (a) HapticGear is portable force display Hirose et al. (2001), (b) FlexTorque wearable haptic device for VR applications Tsetserukou et al. (2010)

Desktop haptic devices can provide multi-finger force feedback attaching some equipment such as a thimble. That kind of system can be implemented with virtual environment. For instance, Phantom base haptic interaction is utilized for the surgeon trainings (Salisbury & Srinivasan (1997)). Also, two desktop haptic devices can be used for two fingers to grasp a virtual object. Such systems can provide both kinesthetic and cutaneous feedback to the users (Pacchierotti et al. (2012)) (**Figure 2a**).

Hand Exoskeletons

In addition to desktop multi-finger interfaces, some types of multi-finger designs are wearable. Hand exoskeletons are most well-known multi-finger interfaces.

The hand exoskeletons can be divided into different categories according to actuator type, power transmission method, degrees of freedom, intention sensing method, and control method (Heo et al. (2012)).

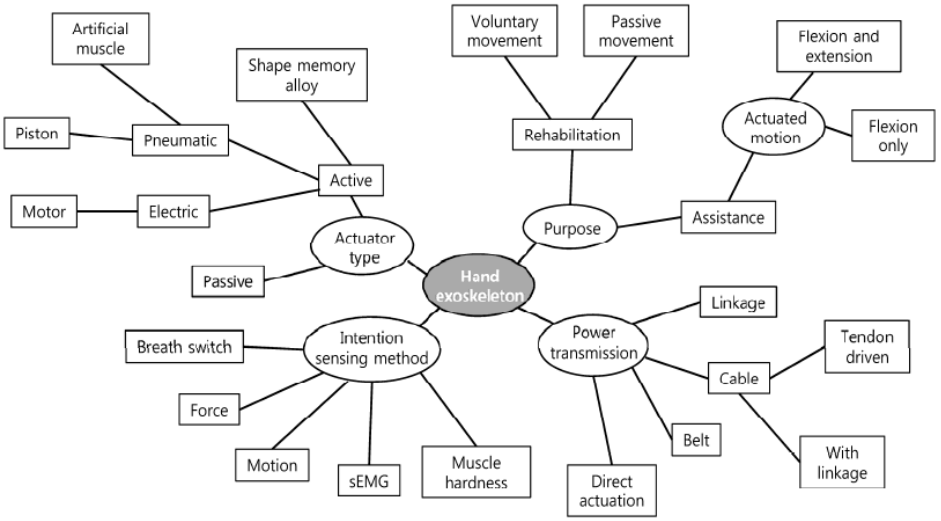
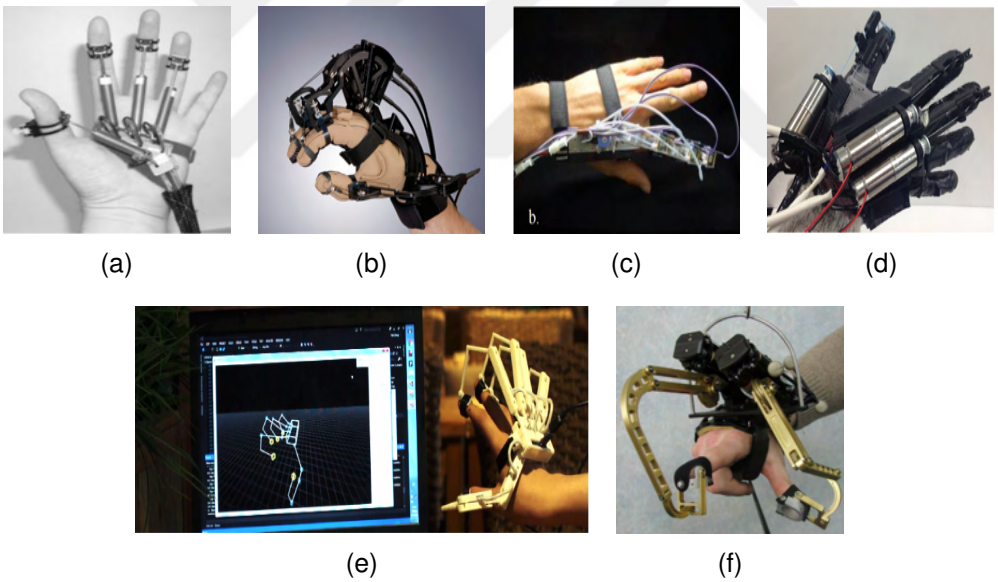


Figure 4: Hand exoskeleton classification (Heo et al. (2012))

Research groups have developed hand exoskeletons for rehabilitation, assistance, teleoperation, and VR applications. However, haptics can also be used for teleoperation and VR applications (Heo et al. (2012)). For example, Master II-ND (**Figure 5a**) is a pneumatic actuated hand exoskeleton (HE). Force feedback is provided by pistons which are nested inside the palm. The HE can exert force up to 16 N. To make the design simple, pulleys and cables are not used, and the design is compact and lightweight (Bouzit et al. (2002)). At the same time, the components of the HE are shifted to the back of hand to allow for comfortable gesture facilities to the users. The CyberGrasp (**Figure 5b**) (CyberGlove Systems, USA) is a commercially available exoskeleton glove which exerts force feedback pulling fingertips via cables. However, the volume of the exoskeleton is large, and expensive. Especially, the cost of CyberGrasp restricts using it specific applications (Achibet (2015*b*)). Fontana et al. (2013) developed a lightweight HE which applies force the thumb and index finger. It can be used both with virtual reality and teleoperation applications. Force is delivered to the fingers via rigid mechanism. Lelieveld & Maeno (2006) designed a lightweight 4 DOF HE to be used in master-slave applications. Force is delivered via cables, and for the motion tracking position sensors are utilized. The mechanism comprises more DOF when the system is compared with other designs (**Figure 5c**). Springer & Ferrier (2002) introduced multi-finger force-reflecting haptic interface. The four-bar mechanism delivers force to the fingertip, and it provides 1 DOF motion. Zhou & Ben-Tzvi (2015) developed the RML which is a haptic glove used for mobile navigation application. The gesture

of the user controls the mobile robot. The distance information between obstacles and the robot is sent back to users as haptic feedback, and users are informed about the robot's proximity. Angular position encoders are used for the position estimation. Another glove actuated by dual cable was developed by Daejeon Convention Center. Two cable systems are implemented to the HE. One of them to measure the position of the fingers, the other cable delivers the force to the fingers. Altough its volume is large, it is lightweight, 238 g Park et al. (2016) (**Figure 5d**). DEXMO (**Figure 5e**) is also a lightweight, it weights less than 270 g, and inexpensive HE, also developed for VR applications. Two stopping sliders are actuated by micro servo unit. When the force feedback is activated, the ratchet wheels are locked.Despite its high motion accuracy the disadvantage of this HE is that it applies binary haptic feedback as same as most of haptic HE do (Gu et al. (2016)).

Some HE provides tactile sense. These systems combine large forces and small-scale effects. For example, WHIPFI stimulates pinching by constraining thumb and index fingers, and provides tactile feedback pressing the fingertips Gosselin et al. (2005*a*) (**Figure 5f**).



(a) (b) (c) (d)

(e) (f)

Figure 5: Hand-exoskeleton (a) MasterII,a pneumatic hand exoskeleton Bouzit et al. (2002). (b) CyberGrasp is commercial haptic glove (CyberGlove Systems, USA) Achibet (2015*b*). (c)4 DOF hand exoskeleton Lelieveld & Maeno (2006). (d) Dual-Cable haptic glove Park et al. (2016). (e) DEXMO hand exoskeleton for VR applications Gu et al. (2016). (f) WHIPFI is a two finger exoskeleton that includes tactile feedback Gosselin et al. (2005*a*)

Fingertip Haptic Devices

Although exoskeletons are wearable, they can be heavy, and this reduces the effectiveness of these devices. However, fingertip haptic devices are easy to wear. Moreover,fingertip is the most sensitive part of body when people grasp and manipulate objects. Thus, fingertip haptic devices have appropriate design possibilities to apply cutaneous and tactile feedback Pacchierotti et al. (2017).

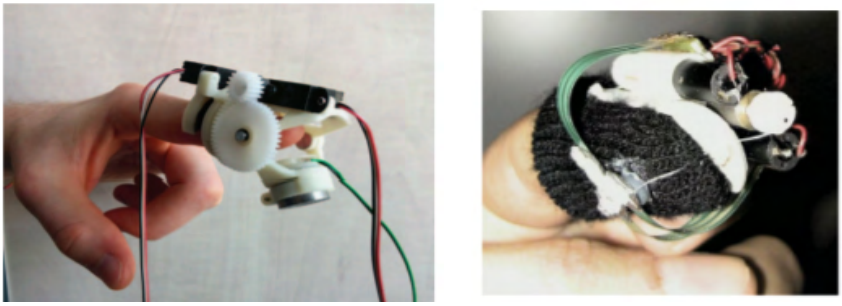
Cutaneous feedback provides a way to develop effective and compact haptic devices, because the density of mechanoreceptors is high on the fingertip skin. Cutaneous cues are also more informative than kinesthetic ones for curvature discrimination and fine manipulation Wijntjes et al. (2009).

There are four main cutaneous sensation;lateral skin stretch, normal indentation, relative tangential motion, and vibration. Tactile sense can be a combination of these categories. Fingertip haptic devices can exert different types of haptic feedback; i.e. displaying different contact force, and pressure, displaying curvature, applying vibration to distinguish materials or objects, distinguishing softness of materials, displaying friction, and exploring the surface geometry Pacchierotti et al. (2017).

Fingertip haptic devices can be classified according to the cutaneous sensation that they provide: normal indentation, lateral skin stretch, tangential motion, vibration Pacchierotti et al. (2017). Normal indentation is applied from one or multiple moving parts in design. These moving platforms or mechanical parts provide softness, curvature and pressure sense on the fingertip.

Moving platforms are popular method, and they are translated and oriented to apply cutaneous feedback to fingertip. The first fingertip haptic device was presented by Frisoli et al., in 2008. The device is capable of exploring curvature discrimination of virtual shapes. The system consists of a parallel platform and a moving platform under the fingertip. The parallel platform change the position of the fingertip device. The fingertip device is attached to this platform as the end effector. In the meantime,the moving platform contacts the fingertip in different positions to emulate the virtual shape Solazzi et al. (2007). Gabardi et al. (2016) presented Haptic Thimble (**Figure 6a**) which is wearable fingertip haptic device. The device allows surface exploration in virtual environment. Also, the device has compact size 66 x 35 x 38 mm and it weights 30 g.

Prattichizzo et al. (2013) (**Figure 6b**), presented a 3-DoF wearable fingertip device that consists of two platforms. One platform is static while the other platform is mobile and is applying requested force. The two platforms are supported by three motors via cable. Motors manipulate the cables to simulate different oriented contact. Also, three force sensing sensors measure the fingertip contact force. Pacchierotti



(a) (b)

Figure 6: Fingertip Haptic Devices (a)Haptic Thimble (Gabardi et al. (2016)), (b) 3 DoF wearable device(Prattichizzo et al. (2013))

et al. (2014) presented (**Figure 7a**) a similar device, with a higher accuracy. This device also consists of two platforms and three motors. Motors have position encoders and the length of cables can be adjusted. The platform can be moved towards the fingertip using the cables. One force sensor is attached to the platform's center to measure the cutaneous force perpendicular to the skin surface of fingertip. In another project, Kim et al. (2016) (**Figure 7b**) use the same idea for the design of a fingertip haptic device and add four Inertial Measurement Unit (IMU) sensors to track the palm and index finger rotation in virtual environment. One IMU sensor is attached to the contact plane, one at the palm, between the proximal interphalangeal (PIP) joint and the metacarpophalangeal (MCP) joint, and to the body part. Also, a force sensor is placed on the contact plane to measure contact force.

(a) (b)

Figure 7: Fingertip Haptic Devices (a) DoF cutaneous haptic device (Pacchierotti et al. (2014)), (b) 3 DoF cutaneous haptic device device(Kim et al. (2016))

Although the two platform devices can be applied in different scenarios, the end-effectors are always in contact with the fingertip. Thus, they do not offer the sensation of interrupting and making contact with virtual objects. In this respect, Chinello et al. (2015) presented the 3RRS wearable fingertip device which is composed of two platforms. Three servo motors are fixed to the upper body, and the mobile platform exerts force to the volar skin surface of fingertip. These two parts are connected by

three legs. Motors actuate the legs to render forces from the virtual environment.

In addition to applying force to a fingertip using moving platforms, the devices that are designed for lateral skin stretch can be also used to exert vertical force to fingertip. Minamizawa et al. (2007) (**Figure 8a**) developed a wearable fingertip device to present tangential and normal cutaneous feedback to the finger pulp. The device applies force using two motors. The motors move a belt which under the fingertip. When the motors spin in the opposite direction, the belt exerts vertical stress. If they spin in the same direction, motors generate shearing stress. This provides the information about the mass of the virtual objects. A similar design was used by Bianchi et al. (2016) (**Figure 8b**) for fabric based wearable device. The device presents different softness by stretching of the fabric under the fingertip. Two motors are used to manipulate the fabric. Also, the device can apply tangential force.

(a) (b) (c)

Figure 8: Fingertip Haptic Devices (a)Wearable Haptic Display (Minamizawa et al. (2007)), (b)Wearable Fabric Yielding Display (Bianchi et al. (2016)) (c) Wearable Fingertip Haptic Device (Leonardis et al. (2015))

Furthermore, Leonardis et al. (2015) (**Figure 8c**) presented the 3RSR wearable haptic device. The device has both contact and no contact capabilities. The parallel mechanism is located under the finger as a moving platform, and the device can provide both position and orientation information. Three motors are used to actuate the leg of parallel mechanism. Schorr & Okamura (2017) used the same type of design to develop a fingertip device that has parallel mechanism. The device exerts normal, lateral and longitudinal (distal and proximal)skin deformation. The normal force that the device can apply is 2 N.

2.1.3 Conclusion

Desktop haptic devices can be used in restricted workspace and the commercially available ones tend to be expensive.In order to provide more freedom of movement to

users, wearable haptic devices more suitable.Although the devices developed by [Hirose et al. (2001)] and [Tsetserukou et al. (2010)] (Figures 3a and 3b) are wearable, the weight can reduce their appli-cability. Hand exoskeletons are instead developed as compact and comfortable devices.There are different types of designs, for example glove or rigid designs, for all fingers or some of them. Their applications range rehabilitation to teleoperated applications. However, some exoskeletons are bulky, and cover large volume on fingers and hand. While others cannot provide haptic feedback. Thus, it is not possible to feel the softness of objects in a virtual envi-ronment or physical objects in a tele-operation scenario. Fingertip haptic devices are also wearable, and can be compact. Tthey are especially effective for tactile applications, and controllable cutaneous feedback. The developed fingertip devices provide shear stress, tangential or normal force. However, the majority of the literature on finger-tip haptic devices does not include rendering of various levels of stiffness. Therefore, in this project, a fingertip mechanism which includes adjustable stiffness using a silicone material, was designed.

There are different types of designs, i.e. glove or rigid designs. They have different range application from rehabilitation to teleoperated applications. However, some exoskeletons are bulky, and cover large volume on fingers and hand. Also, they cannot provide touch sense. Thus, it is not possible to feel the softness of the materials if they are used in virtual environment. Fingertip haptic devices are also wearable, and can be designed compact. They can be worn for all fingers. Especially, they are effective for tactile applications, and controllable cutaneous feedback. The developed fingertip devices provide shear stress, tangential force, normal force. However, the most of the research about fingertip device do not include rendering various levels of softness. Therefore, in this project fingertip mechanism which includes adjustable silicone material, was designed.

2.2 Motion Capture

Motion capture is ”the process of capturing the motion of a human body, at some resolution”. The application areas of motion capture are military, medical, entertainment, and advertising (Moeslund (2001)). Motion capture interfaces transpose the finger motions to a virtual environment. They are commonly used to provide interaction between the user and virtual environment. Motion capture can be produced using various methods, i.e. using data gloves, IMU sensors, optical camera, and hall effect sensor.

2.2.1 Data Gloves

Glove-based systems can be used to acquire hand movement data. These systems include an array of sensors. Data gloves vary in terms of sensor technology, location on the fingers, and mounting method. Some data gloves measure the angle, others measure average curvature. Also, some use 'cloth' (fabric), others use plastic, rubber, silicone etc. Some well known commercial glove systems are, CyberGlove (**Figure9a**),HumanGlove (**Figure 9b**), 5DT Data Glove (**Figure 9c**) and Pinch Glove (**Figure 9d**) (Dipietro et al. (2008)).

(a) (b) (c) (d)

Figure 9: Data Gloves (a)CyberGlove3 (b) HumanGlove (c)5DT Data Glove (d) Pinch Glove

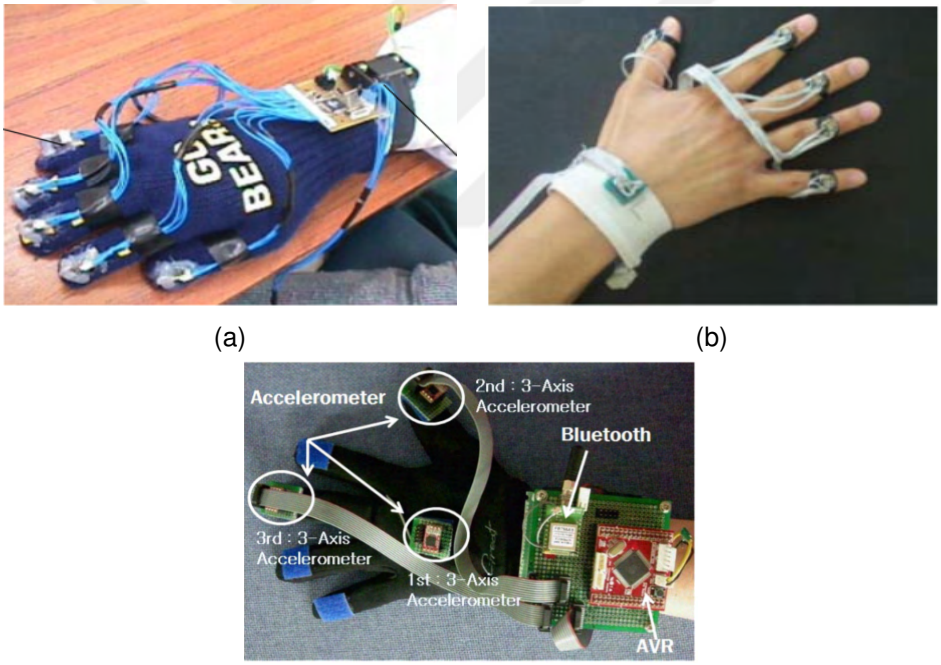
2.2.2 Optical Tracking

Optical tracking is applied using both marker based or markerless techniques. The user wears a device which carries markers, or coloured dots are placed on the finger for the marker based optical tracking technique. Kry & Pai (2006) apply the marker based technique to capture contact forces. Also, Tzemanaki (2016) tracked the index finger motion placing coloured dots to each joint of the index finger by applying colourgrading, colour detection and blob tracking.

Optical tracking without markers offers multi finger motion capturing using computer vision. Wang & Popovic (2009) presented an optical motion capture system that consists of a regular camera and a colourful glove. The pose of hand was determined from a single frame. Several hand tracking systems are commercially available, e.g. LeapMotion (Leap Motion Inc, USA) and CamBoard pico (pmdtechnologie GmbH, Germany). Scheggi et al. Scheggi et al. (2015) used the LeapMotion to mimic the pose of a hand in virtual environment. Through LeapMotion, the fingertip haptic device interacts with virtual objects. Another example for markerless motion tracking is Digits, presented by Kim et al. (2012). Digits is a wrist worn device,a and has infrared cameras to track hand gestures.

2.2.3 IMU Sensor Based Motion Tracking

Inertial Measurement Units (IMU)consist of tri-axis gyroscopes and accelerometers. IMU enables tracking rotational and translational movements. There are some projects that use IMU sensors for motion tracking. Perng et al. (1999) presented a glove which is called acceleration sensing glove (ASG) (**Figure 10a**). The glove includes 2-axis accelerometers on the fingertip. The accelerometer sensors are used to detect and translate the finger motion into a computer. b Hernandez-Rebollar et al. (2002) produced The AcceleGlove (**Figure 10b**) which can be used to manipulate objects in virtual environment. This glove can be used with a virtual keyboard which includes 26 postures of American Sign Language (ASL). The glove consists of a set of dual-axis accelerometers. Also, a paper published by Kim et al. (2009), demonstrated KHU-1 data glove which is used for 3D hand motion tracking and gesture recognition system. Three axis accelerometers are attached to the fingernail of middle finger and thumb. (**Figure 10c**)



(a) (b)

(c)

Figure 10: Gloves that includes IMU sensor for motion tracking (a) Acceleration Sensing Glove, (b) The Accele Glove (c) KHU-1 data glove

Apart from these hand gesture recognizing and motion tracking applications, IMU sensors have been used for some of the haptic devices to control them in virtual environment. Tavares et al. (2016) designed a glove for the rehabilitation of hand movement impairments. An IMU board which comprise 3-axis accelerometer and gyroscopes is used to track the hand position and orientation in the virtual environment.

Kim et al. (2016) (**Figure 7b**) developed 3 DoF cutaneous haptic device. The device is attached to the fingertip of the user. The three IMU sensors are attached to the index finger to track the position and orientation of the finger.

2.2.4 Conclusion

Motion tracking can be done implementing different methods. There are commercial data gloves, but they are expensive. Beside this, using different types of wearable device may cause drawback, because two wearable material will be different restriction. The combination of them should be cumbersome for virtual reality applications. Optical tracking method requires camera system and image processing, or hand tracking system like LeapMotion. The drawback of this method is, when fingers overlap, the camera cannot detect the position of fingers appropriately. While, attaching sensor to fingers can solve this problem because they give the position data via electronic board. For instance, if the fingertip device is used for three fingers with LeapMotion, it is not necessary to see five fingers in the virtual environment. Because of thee reasons, IMU sensor was used for the project.

2.3 Virtual Environment Applications

When the haptic feedback is integrated with an virtual environment, it can make virtual applications more believable. Haptic feedback can be applied for various scenarios such as teleoperation systems, gaphical user interfaces, training,rehabilitation and games (Ott (2009)).

Frisoli et al. (2009) developed a force-feedback exoskeleton system for upper-limb rehabilitation. When the user exercises, he/she controls an avatar representing his/her movement in the virtual environment. By receiving acoustic and visual feedback, the patient can enhance their awareness. The virtual environment was developed for that project using XVR Development Studio.

Gosselin et al. (2005*b*) demonstrated wearable haptic interface that can provide finger interaction with virtual environment. The device allows manipulation of virtual objects. This project can be used for teleoperation applications by controlling a robot arm.

Kuhnapfel et al. (2000) developed a surgical training based on virtual environment and simulation techniques in the simulation software called KISMET. Surgeons

can advance their surgery abilities using this software.

The RoSS (Robotic Surgical Simulator) is a portable system that was commercially produce by Simulated Surgical Systems LLC (San Jose,CA, USA). This device includes 52 exercises and helps surgeons to improve their basic surgical skills with hands-on surgical training (Bric et al. (2016)).

Another haptic device Falcon Novint has been used in several haptic research projects. Also, the device can be utilized in the game industry. The company provides own software. This device can be implemented using DLL files to different type of software such as Unity. DelBrocco (2013) used Novint Falcon for virtual applications that they designed in Unity.

2.4 Conclusion

The presence of a virtual environment can play a crucial role in certain applications in the medical field. A large part of the applications is intended for rehabilitation, surgical training as well game industry.In order to provide more experience to surgeons, haptic devices can provide sensitive and accurate force feedback, which can benefit surgical training.

2.5 Summary

Traditional desktop devices simulate force feedback to the entire arm. However they are grounded, and provide restricted workspace to users. Multi-finger interfaces can be designed both in desktop form or wearable,i.e., exoskeletons. Especially, hand exoskeletons provide extended workspace to users. The designed exoskeletons should be lightweight and compact because they are attached to hands. In this project, light materials will be chosen for the design of hardware. Some HE's cover both VR and haptic feedback; however, they cannot provide the sense of object stiffness appropriately. Especially, in surgical operations, sensing the stiffness of tissues can be significant for the surgeons. This approach will be investigated for implementation in the hardware.

Data gloves are suitable for VR applications and motion tracking. However, the high cost of these devices reduce the accessibility. Among several motion tracking methods, IMU sensor based motion tracking provides compact design. Virtual environment was created on Unity to test the device.

3 Materials and Methods

This chapter presents the materials that are used for the design of the fingertip haptic device and the syringe pump design as well as the methods employed for motion tracking, virtual environment and characterization of the device.

3.1 3D Printed and Acrylic Materials

3D printing process is fast method for prototyping at low cost. In this project rack and pinion of haptic device and syringe pump, and the support that is attached to syringe pump is produce acrylic while others were printed in PLA. Also, the flat surface attached under the pillow is the same material.

3.2 Silicone

DragonSkin silicone is used in various applications such as creating skin effects, movie special effects, and creating soft robots and materials. Moreover, because of the physical properties, they can be used for medical applications. Also, this rubbers can be applied for various industrial projects.

The silicone materials have different shore hardness. Shore hardness is the resistance of materials to indentation. In **Figure 19** the scale of shore hardness of various materials is illustrated. 'A' indicates what type of Durometer is used to find the shore hardness of the material. For example, the shore hardness of a rubber band is 20 A.

There are various types of DragonSkin silicone with different hardness ranging from 10A to 30A. In this project, 10A DragonSkin silicone is used because it is softer which is appropriate of soft haptics. 10 A silicone is soft, strong and stretchy. These are the key properties to produce variable softness. The tensile strength of the DragonSkin 10 A is 475 psi, and Die B Tear Strength is 120 pli.

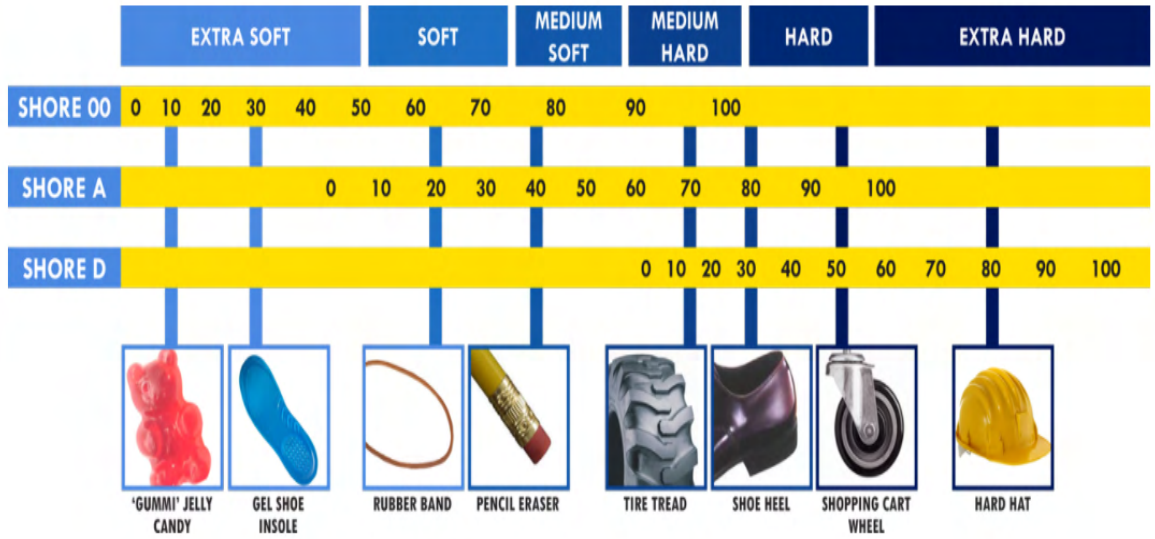


Figure 11: Shore Harness Scales

Figure 12: DragonSkin 10A

3.3 Silicone Glue

Sil-Poxy is an adhesive used for bonding silicone rubber and other subtances such as plastic or plaster. It can be used for repairing torn silicone moulds. Strong and flexible bonds can be created by Sil-Poxy. Shore hardness of this product is 40 A, and the tensile strength is 750 psi. Moreover, 12 minutes curing time is the significant feature of the product when it is compared to curing time of DragonSkin silicone. Curing time of DragonSkin silicone changes from 30 minutes to 16 hours. It is transcendent and specific gravity is 1.12 g/cc.

3.4 Servomotors

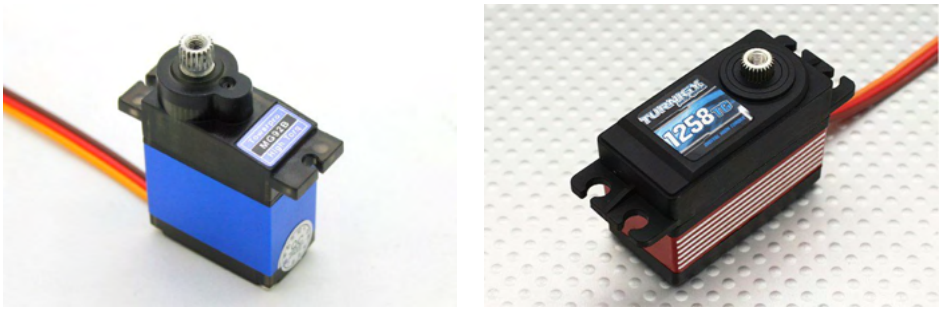
Two types of motors were considered for this project; DC motor and servomotor. As position information is adequate for the control of force, servomotors were researched.

Two servomotors were chosen; one for actuating the rack and pinion mechanism in the fingertip haptic device and another servomotor for the syringe pump. Table 1 shows the selection criteria of different servomotor for the fingertip haptic device. The motor is attached to the finger's distal phalange. Thus, the motor should have small size and lightweight. These features are important to provide an easy wearable, compact and lightweight haptic device. Also, when the user interact with virtual objects, motor should apply force as fast as possible. Because of this, the response time of servo motor is important for quick and realistic haptic feedback. Moreover, the torque of motor should be adequate to withstand the desired force.

Table 1: Market search of servo motors for the haptic device

Brand	Stall Torque	Operating Speed	Dimension	Weight
TowerPro SG90	1.8 kgf.cm	0.1 s/60 degree	22.2x11.8x31 mm approx.	9 g
TowerPro MG92B	3.1 kgf.cm	0.13 s/60 degree (5.0v)	22.8x12x31 mm	13.8 g
Turnigy MG90S	1.8 kgf.cm	0.10 s/60 degree	22.8x12.2x28.5 mm	13.4 g
TowerPro MG16R	2.7 kgf.cm	0.10 s/60 degree	29x11.2x29mm	18.8 g
TowerPro MG90D	2.1 kgf.cm	0.10 s/60 degree	22.8x122x28.5 mm	13 g

TowerPro MG92B was chosen for the fingertip haptic device. It provides the highest torque among the searched servomotors. Also, the weight and size of the motor are appropriate for this project.



(a) (b)

Figure 13: Selected servo motors(a) TowerPro MG92B, (b) Turnigy 1258TG

Additional market research was carried out for the syringe pump mechanism. For this mechanism, the motor should also provide high speed and torque. These are the key parameters for the syringe pump mechanism. The size and weight of the motor are not crucial for the selection of motor because this mechanism will be placed on a table. Table 2 shows the selection criteria for the servo motor for the syringe pump. For the syringe pump mechanism Turnigy 1258TG was chosen. With the budget of the project considered, this servomotor offers enough speed and torque to inflate the

designed silicone material. The selected two servomotors are actuated by an Arduino board.

Table 2: Market search of servo motor for the syringe pump mechanism

Brand	Stall Torque	Operating Speed	Dimension	Price
Turnigy 1258TG	12 kgf.cm	0.11 s/60 degree	40.3x20.2x27.2 mm	26.66 £
Hitec HS-7955TG	18/24 kgf.cm	0.19/0.15 s/60 degree	39.88x19.81x36.83 mm	81.12 £
Hitec HS-5645MG	10.3 kgf.cm	0.23 s/60 degree	40.6x19.8x37.8 mm	31.16 £
Hitec HS-8335SH	18.5/24 kgf.cm	0.16/0.07 s/60 degree	40x20x37 mm	53.8 £
Futaba S3306MG	24 kgf.cm	0.16 s/60 degree	66x60x57.1 mm	47.84 £
Power HD WP 23	23 kgf.cm	0.12 s/60 degree	40.x20.5x38.5 mm	57.59 £

3.5 Arduino

The Arduino MEGA 2560 board is used to provide communication with the virtual environment, while an IMU sensor is also connected to it.

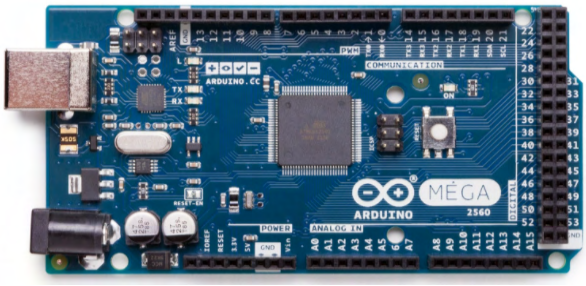


Figure 14: Arduino Mega 2560 Board

Arduino can be powered via USB cable externally. The recommended voltage range is between 7-12 volt for the external (non-USB) supply. If the board is supplied less than 7, then it may be unstable because the 5V pin of Arduino may supply less than 5 V. If more than 12 V is supplied, the voltage regulator may overheat and damage the board. Arduino mega offers 54 Digital I/O pins. Two servo motors are connected to digital pins 8 and 9. 4 pins can be used for receiving data (RX), and 4 pins work for transmitting data (TX). When data is being transmitted, LEDs of the RX and TX on the board flash. In the project, gyro and acceleration data of the IMU sensor are sent via RX and TX pins. Also, communication between IMU sensor and Arduino is created via I2C serial communication. Arduino offers Wire library which simplifies the use of I2C bus. Furthermore, Arduino software includes a serial monitor that can be used for data

and receiving data. In this project the data transmission between Arduino and Unity is provided via this serial monitor.

3.6 Inertial Measurement Unit (IMU) Sensor

IMU sensors measure linear and angular motion using 3 axes gyroscopes and 3 axis accelerometers. These sensors are chosen for different applications such as gesture recognition, wearable sensors for health, and motion enabled games. In this project, the IMU 6050 sensor is used for the motion tracking of the fingertip. The IMU 6050 sensor provides different sensitivity for gyroscopes ± 250, ± 500, ± 100, ± 2000. Also, the sensitivity for the acceleration is $\pm 2g$, $\pm 4g$, $\pm 8g$, $\pm 16g$. The operating voltage of the sensor is between 2.38 and 3.46 Volt. The raw data from IMU sensor are sent to Arduino via I2C bus .

Figure 15: IMU 6050 sensor

3.7 Pressure Sensor

The pressure that pillow when it is inflated was measured using Honeywell HSC-SAAN015PDAA5. This sensor has pressure range between -15 and +15 psi. Total band error is ± 1.

3.8 High-accuracy CCD Laser Displacement Sensor

The inflation of the silicone material is measured with a laser displacement sensor as these sensors offer high resolution and accurate measurement. For this purpose, Keyence sensor head is used to measure how much silicone material inflates. The

product number is LK-G402. It can measure the ± 400 mm distance, and it is a high speed sensor. The laser beam come to the surface of the material that is put under the head, and the sensor measures the reflection. The sensor is connected to computer via controller.

Figure 16: Keyence LK-G402 Laser Displacement Sensor

3.9 Unity

Unity 3D is a game engine that can work with various platforms such as PC, Web, IOS, Android, and Xbox [1]. Unity can be engaged 2D, 3D, VR and AR games and apps. 770 million gamers play Unity based games all over the world [2]. The toolset, editing and testing on the fly in Unity makes developers easy to use that program. [1] Also, Unity provides high visual fidelity and rendering, lighting, visual effects. [3] Beside these visual features, Unity offers physic engines for both 2D and 3D game design. The physics engines provide collisions, dynamic and kinematic features for objects that programmer create. In Unity, the standard Mono runtime is used for scripting. Scripting can be done using either C# or JavaScript. Also, using script, programmers can create their own Components to add different features to designed object. i.e. rigid body features [5].

Coding is done in scripts and two programming languages can be used; C# and UnityScript. Also, .NET languages which can compile DLL file can be used. In Unity, programmer can create own components using scripts. By coding in script, programmer can modify components, and add more properties to components [9].

Unity was chosen for the virtual environment application because it offers object oriented programming via C#. Another reason is, the program provides, collision and rigid body features which make easy the interaction between real life and virtual environment.Moreover, Unity can be connected to electronic board via serial

port. Thereby, collision and physical features can be used easily to actuate the motors.

4 Mechanism Design

4.1 Fingertip Haptic Device Design

The goal of the wearable device is to provide normal skin deformation to the fingertip of the index finger. The designed fingertip device consists of two main parts. The first part is a platform. The force is applied by the platform, and it de-forms the skin. The second part is a rack and pinion mechanism. The platform is actuated linearly by the rack and pinion mechanism.

4.1.1 The Design of The Platform

Adult humans have different finger distal extent and length. The research conducted by Peters and Mackenzie Peters et al. (2002) indicates the mean size of female and male index finger extent that was measured from 98 males and 402 females. The research shows that the mean extent of male's index finger tip is 12.7 mm and 12.0 mm for the left and right finger respectively; while for female is 10.4 mm and 10.2 mm. Moreover, Peters et al. (2009) gave data for the index fingertip area of males and females which was measured from 50 males and 50 females. The research shows that the index fingertip area for females is around $360mm^2$, and for males is $420mm^2$. The platform located under the fingertip, was designed according to these data. The area is $478.5mm^2$ and the extent of the platform is $25mm$. The extent of the platform is larger than the measured human's finger extent, because when the soft material is inflated, it is considered that the inflated surface area of the soft material will not the be same as the platform's surface area and the extent will be different.

The platform was designed to be placed into the inflated material, in order to restrict the sinking of the finger in the inflated material. The platform has six holes above the surface, and one hole below. In (**Figure 18a**), the six holes that are illustrated distribute the air flow. One big large hole was not preferred as the user's finger might have sunk inside the inflated material, and feel a shape instead of a flat surface. Also, six holes provide quick homogeneous distribution of air in the inflated material. Under of the platform a 7 mm tube was created. The diameter and the length of this tube was designed considering the plastic tubing that is attached to it. From this tube, the

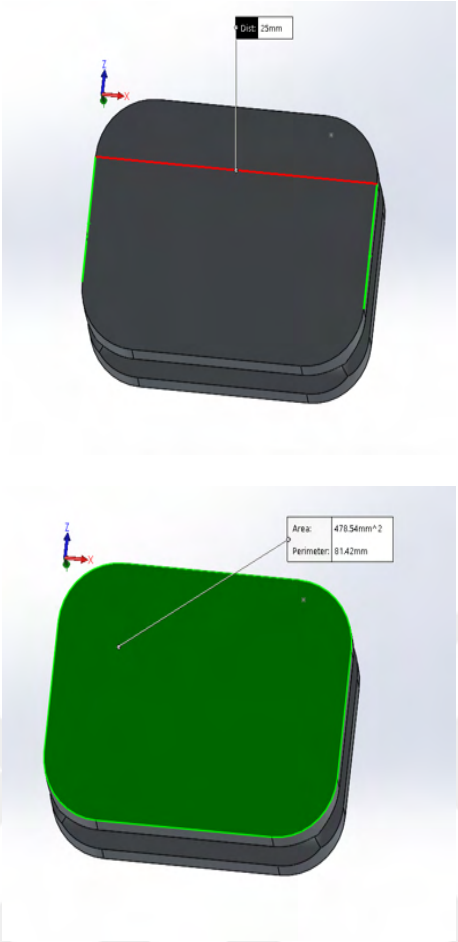


Figure 17: Platform Design (a) The extent of the designed platform is 25 mm, (b) The surface area of the designed platform is $478.5mm^2$

material is inflated by a syringe pump.

Moreover, each side of the platform has a 2 mm slit to hinder the air flow to the side of the platform. This, prevents expansion of the silicone material to horizontal axes. When the silicone material gets in contact with the fingertip, air starts to flow towards the horizontal axes. Therefore, a design without such a slit would not provide the appropriate stiffness variance.

4.1.2 The Fabrication of Pillow

Haptic feedback is applied to the fingertip using belt and parallel systems. Tsetserukou et al., Leonardis et al. Schorr & Okamura (2017) use rigid material under fingertip to apply force. Instead, Pacchierotti et al. ,Minamizawa et al. (2007) design their haptic device with fabric belt. Moreover, Koo et al. and Frediani et al. use dielectric elastomer actuators to exert force. Rigid materials and belt mechanism provide softness

(a)

Figure 18: CAD drawing of fingertip haptic device's platform

(a) (b)

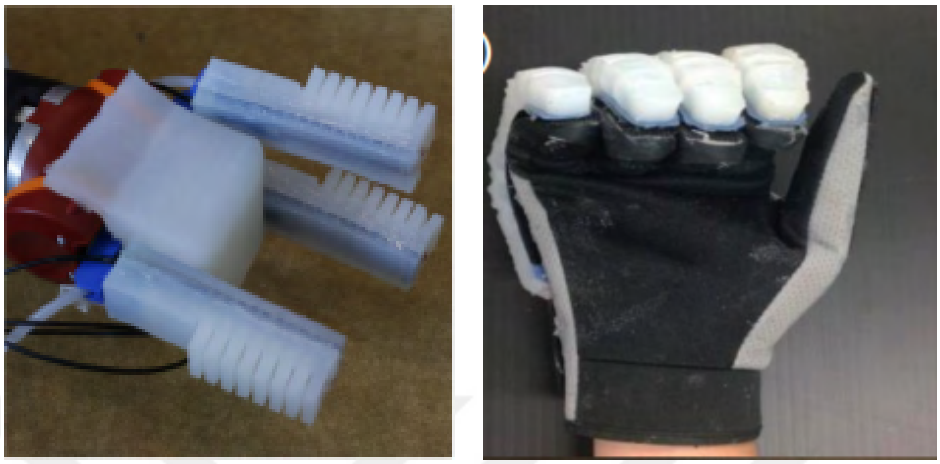
Figure 19: 3D printed platform (a) Isometric view of 3D printed platform, (b) Side view of 3D printed platform

perception controlling the motors position. However, these devices do not offer actual indentation sense because belt and rigid material cannot be deformed. In order to provide indentation sense and different stiffness sense to users, inflated material was used in this project.

First, a piece of latex glove was used as inflated material. This material was thin; and thus, it deformed after inflating it a number of times. The most challenging issue using latex material was sticking it to the platform. Glueing did not provide satisfying solution. Also, air leakage problems could not be solved. Because of these reasons, silicone material was considered instead.

In addition to the appropriate structure to prevent the air leakage, silicone materials are popular in the soft robotics and medical fields. Especially, the silicone

called DragonSkin is prefered. Homberg et al. (2015) presented a soft hand like gripper that includes three fingers. Moreover, DragonSkin was applied by some researhers to the design of soft exoskeleton. Yap et al. (2015) developed a soft exoskeleton for rehabilitation purpose. Fingers are bent using the designed soft actuators.



(a) (b)

Figure 20: Some applications that were designed using DragonSkin (a) Soft hand actuator (Homberg et al. (2015)) (b) Soft hand exoskeleton (Yap et al. (2015))

In this project, 10 A and 30 A silicone materials were considered, with 10 A silicone being chosen. Although 30 A is more durable than 10 A, it also requires more pressure to inflate as the tensile strength of 30A silicone is more than 10A silicone. According to the datasheet of the adhesive, 30A silicone has maximum 500 psi tensile strength whereas 10A silicone has 475 psi. Moreover, 10 A can provide softer object sense. Furthermore, as this project is aimed for medical applications, 10 A was considered as it can more accurately emulate the softness of organs. For example, silicone is selected to create artificial organs. Gregory et al. (2009) designed a naturally shaped silicone ventricle. This suggests that, the designed silicone haptic device could fit the surgical applications, and simulations.

The silicone that covers the plastic platform was moulded. Two various moulds were designed, and 3D printed.

First Mould Design

First 3D printed mould design is illustrated in Figure 21 . The mould consisting of 4 main parts. The aim of the moulding process is the producing a silicone material to put the platform inside of it. Therefore, 2 pieces were designed that have larger than platform size. **Figure 42b** shows the measure of the bottom piece, unit1. The

shape of Unit 1 is as same as the platform. Unit 2 is assembled to the unit 1 using M2 screw. 4 holes designed screwing. These two parts should be fitted well to hinder the undesirable flow of silicone to between Unit 1 and Unit2. The each side distance between unit3 and mould is 2 mm. illustrates this. Also, the depth from the bottom of Unit 1 to surface of Unit 2 is 11 mm.

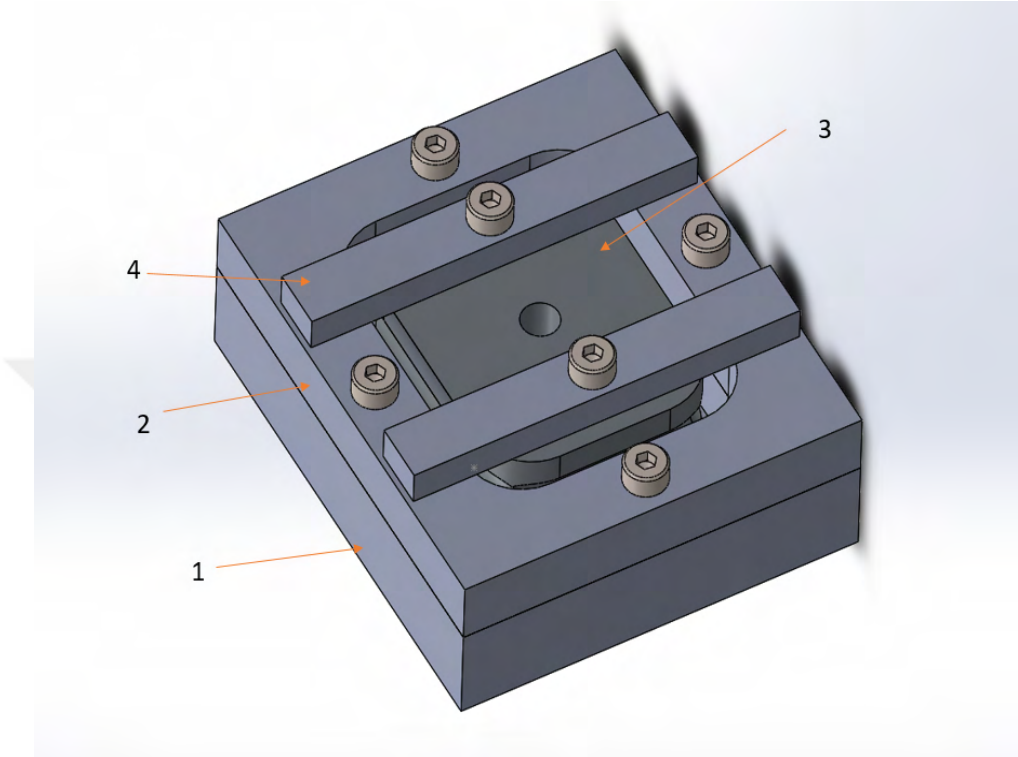


Figure 21: Moulding Process, first design

The Unit 3 design is the significant section to create soft material. The Unit 3 is put into the mould to give the shape of platform to the inside section of the pillow design. The size of the Unit 3 is same as the platform's size. However, the only difference is the depth of Unit 3. In order to inflate the silicone material easily, there should be gap between platform's surface and the silicone design. The depth of Unit 3 designs is longer than platform. The depth of three designs are shown in **Figure 33** The depth of the platform is 8 mm. According to these data, the gap between platform and pillow design is 1 mm, 1.5 mm and 2 mm respectively. **Figure 24** gives the illustration of this idea. In the figure, the first design of Unit 3 was put into the mould, and the gap is 1 mm. The 2 mm is the thickness of the surface of the pillow. From first to third Unit 3 design, the thickness of the surface of each pillow is 2 mm, 1.5 mm and 1 mm respectively. In this project the pillow that has 1 mm surface thickness was used. It is difficult to produce thin silicone material due to having bubble possibility in the silicone mixture. Thus, it was not considered to design thinner than 1 mm surface.

Unit 4 is attached to Unit 3 to stabilize it in silicone and to give the shape of

the platform. All pieces that are used for moulding process for the first mould design are illustrated in Figure 22.)

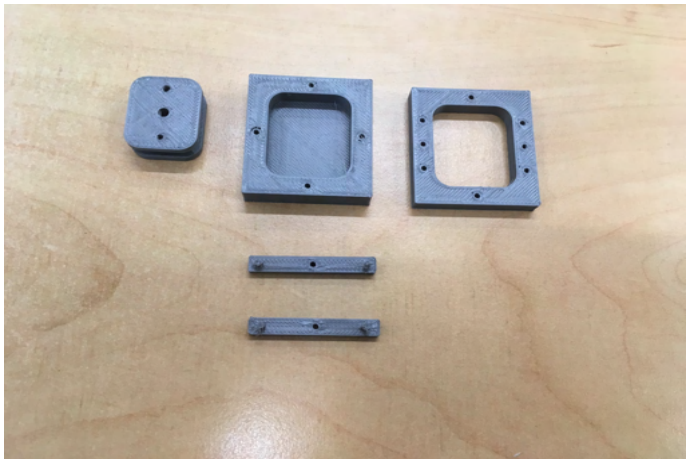


Figure 22: 3D printed first mould design

Second Mould Design

The second design is same of the first mould design (shown in Figure 23). However, to produce tight pillow, Unit 1, Unit 2 and Unit 3 were designed smaller. Unit 1 and 2 were reduced 2 mm. Unit 3 has 21 x 16 mm dimension (shown in Figure 45). When the pillow is inflated that was created from this mould, it becomes stiff faster. Because the range of softness of this pillow is not rich, producing a pillow using this mould design was not considered.

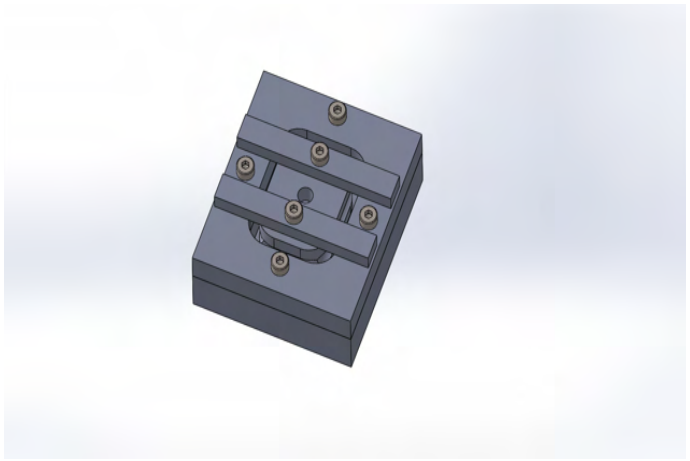


Figure 23: 3D design of second mould

Moulding Process

DragonSkin consists of two parts, A and B. The instruction which is provided from producer was followed to create the pillow material. First, the volume of each moulds was calculated. Later, the volume was multiplied with Specific Gravity that is given in the data-sheet of the product. This value gives information of how much weight silicone mixture should be prepared. Later, 1A:1B weight from each part was poured in a plastic glass, and they were mixed. When the silicone is mixed, bubbles arise in the mixture. The bubbles might cause porous product. Thus, after mixing the part A and B, the silicone was put into the vacuum machine to reduce the number or bubbles. After completing this process, the silicone was poured into the moulds, and Unit 3 was placed in the moulds (shown in **Figure 24**). Cure time of DragonSkin 10 A is 7 hours. After waiting one day, moulds were disassembled.

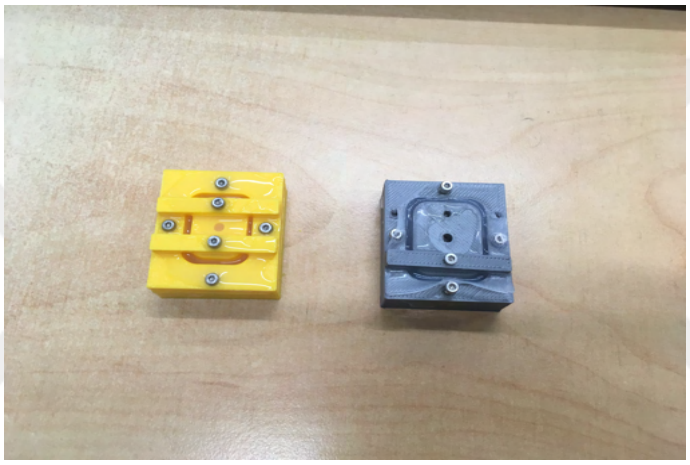


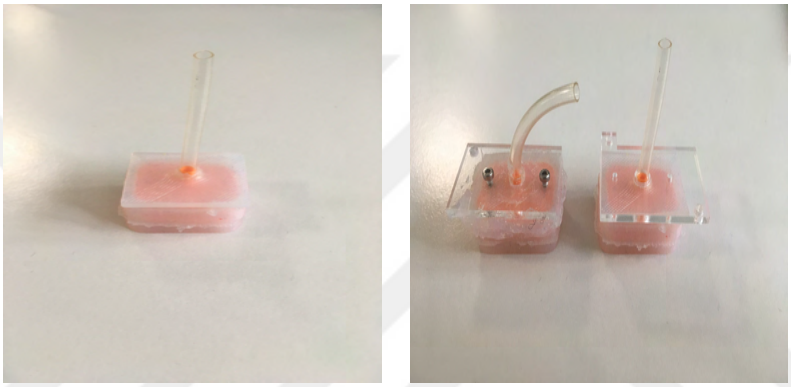
Figure 24: Moulding Process for mould design 2 and 1

The pillow was glued to 3D printed material. Sil-poxy adhesive was used for gluing. The glue is applied to bond silicone rubber to silicone, plastics, plasters, ceramics. However, this method could not hinder the air leakage. **Figure 25** shows how the pillow attached to the platform. In this figure, the platform was designed with flat surface under of it. This flat surface requires to attach pillow and platform to the rack.

This flat surface was changed with acrylic design (shown in **Figure 26b**). Acrylic piece was put under the platform, and it was glued to the pillow part. This also could not prevent the leakage. Another solution is designing a flat silicone surface. The platform was placed between this silicone plane and the pillow.

Pillow was glued to silicone plane, and tubing plastic was attached to the tube of platform. The plastic tube was glued to the silicone plane. To actuate this system via rack and pinion mechanism, this part was placed on the acrylic piece. This method prevented the air leakage of the pillow (**Figure 26b**).

Figure 25: Platform was placed into the pillow



(a) (b)

Figure 26: Assembly of pillow and platform (a) The idea of how silicone flat surface and attached to the pillow , (b) Assemblied pillow and platform ware shown on the right, on the left, non assemblied parts

4.1.3 Rack and Pinion Mechanism

Some of the fingertip haptic devices consists of fingerbelt, and fingerpad. Another design method is parallel mechanism that is attached to fingertip. This mechanism can apply cutaneous feedback to different side of the fingertip.

In this project, rack and pinion system was designed to apply haptic feedback to the fingertip of index finger. The belt system cannot be used because the inflated material and the pressure of the air can cause drift on the belt design. Also, parallel mechanism would be bulky when it is attached to the platform. Moreover, parallel mechanism's legs should be rigid not to be affected from the air pressure because pillow design should be attached the end-effector of that parallel mechanism. Rack and pinion system was designed for this project to actuate the platform linearly, and apply normal force. Also, it is easy to control the indentation with this mechanism.

Rack and pinion was designed considering the space limitations shown in figure. Because of the space constrain, and motor features the diameter of the pinion was chosen 13.6 mm, the pinion pitch circle is 12 mm. The pitch height of the rack is 6.1 mm. The thickness of both pieces is 5 mm. If the motor diameter and rack pitch height of the rack are selected more than 20 mm, the compactness of the system might be deteriorated. The system can expand to the down relative to the Figure 26.

The length of the rack was determined considering the inflation of the pillow because even pillow inflates, it should not exert force at initial position. The length of rack is 30 mm.

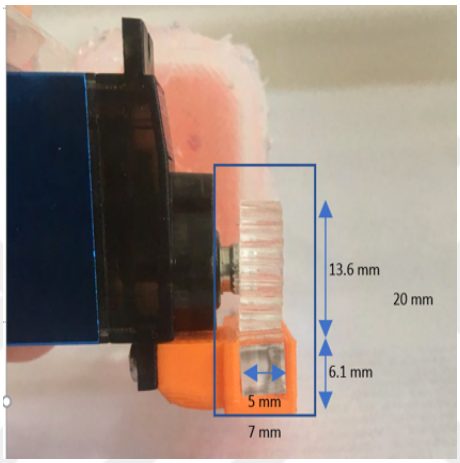


Figure 27: Space limitation of rack and pinion mechanism

The motor of this system can actuate 0.31 Nm stall torque. When the pressure of the pillow is considered, this torque is enough. If the pillow is inflated $4(mm)^3$, the pressure inside of the pillow is 5.17. If the user touches the pillow until feeling the platform, the measured pressure is 13.45 kPa. With respect to the diameter of the pinion, the system can provide more than 3.1kgf.cm torque, and can provide that amount of pressure. The number of teeth of the pinion was determined as 20. Device should be compact to be wearable. Thus, the pinion was attached to the shaft of the servo motor. Circular pitch of the pinion is 18 degree, and the pinion pitch circle is 12 mm. Controlling the position of the servo motor, the linear motion of the platform is changed. Linear motion of the platform can be calculated;

$$\delta = (2\pi\frac{d}{2})\frac{\theta}{360} \tag{1}$$

In this equation δ is displacement of the platform. θ is the angle of the servo motor, and d is the diameter of the pinion.

For example, for 5 degree change of the motor provides 0.52 mm displace-

ment. Also, the speed of the motor is 0.1 sec/60 degree. In 0.1 sec the platform can move around 6.2 mm according to equation (1). Period of the motor is 0.78 sec. The angular velocity is;

$$w = (2\pi f)w = \frac{2\pi}{0.78} = 8\frac{rad}{s} \tag{2}$$

The linear velocity is;

$$V = (wr)$$

$$V = 8\frac{rad}{s}x0.006m = 0.048\frac{m}{s} \tag{3}$$

The mass of the pillow to be moved is 15 gram. Also, the motor should move up the pillow mechanism to exert force the fingertip. When the maximum required pressure converted to the force, 1.345 $\frac{N}{(cm)^2}$ is calculated. The membrane of the pillow is $4(cm)^2$, and the force should be 5.4 N. The motor should provide force for both the weight and force.

In the data sheet of the motor acceleration and declaration time is not mentioned. The acceleration time is taken the half of 0.13 sec according to the triangular operating pattern.

In order to find the required force, the following formula is used,

$$accelerationtime = 0.13sec/2$$

$$a = \frac{V}{t} = \frac{0.048}{0.065} = 0.73\frac{m}{(sec)^2}$$

$$F = (mg + ma) + haptic$$

$$F = (0.015x9.81 + 0.015x0.73) + 5.4 = 5.55N \tag{4}$$

The torque is,

$$Torque = Fxdistance$$

$$T = 5.55x0.66cm = 3.66Ncm \tag{5}$$

After designing the rack and pinion in the CAD software, they were 3D printed. However, the teeth were not printed accurately. Thus, these pieces were created with laser cut (shown in figure 24).

4.2 The Design of Syringe Pump System

The pillow was designed to inflate using air rather than liquid. Liquid cannot provide the intended result. If water is used as a liquid material, when the amount of water increases in the pillow, the mechanism becomes more stiffer, and the weight of the system increases. However, the density of volume is less than air, and it can be compressible.

The air can be transferred via several devices. Small pneumatics pump can be utilized for this application. However,the transferred air should be discharged from the pillow. This discharge operation should close to the pillow. Also, for this operation electronic switch should be used. This method causes to increase the haptic device's weight. Thus, other methods were investigated. Syringe pump systems are significant in the medical field. Syringe is actuated using linear systems. The infusion accuracy of these devices is high. Generally, they can be controlled via their own software. However, these systems can be expensive. For example model NE-1000 programmable syringe pump is 830 dollars. Moreover, the linear system X-LSQ150B-E01-PTB2 that was produced by ZABER is 2,508 dollars. Also, the company OPENBUILDS offers belt driven linear actuators. The cost of these mechanism is reasonable but not appropriate for the project budget. Thus, syringe pump mechanism was designed to infuse air to the pillow. The syringe pump mechanism that was designed is similar the idea of the researcher from Washington University. Rack and pinion mechanism was designed to actuate syringe. In addition to price analysis, this system should work integrated virtual environment. The designed mechanism is actuated using servomotor controlled from Arduino.

For the pump system, 20 ml syringe was chosen. Motor was assembled to the Unit 1, whereas syringe was assembled to Unit 2. Pinion was set to the motor shaft, and rack actuates the syringe oin the horizontal axes. All these parts, then, attached to the plastic flat floor (Unit 3) to stable the system because when the motor actuates the syringe, all mechanism slips.

The infusion time is important for the syringe pump mechanism. Also, the system require high torque relative to positional servo motors. Thus, high torque servo motors reviewed and Turnegy 1258 TG was selected.

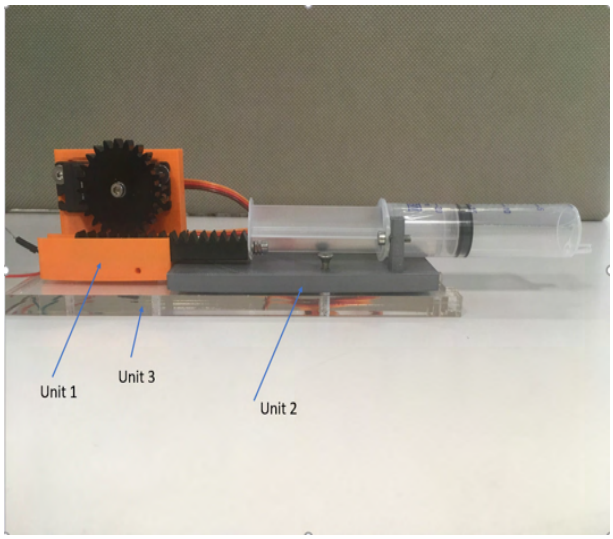


Figure 28: Syringe pump

In order to determine speed criteria, a sample example was calculated. What should be the velocity of syringe to inflate 10 ml air in 0.5 sec?

The syringe diameter is 20 mm, and needle diameter of the syringe is 2 mm. The air flow speed at the needle of the syringe should calculated below.

$$\dot{V} = \frac{Volume}{\Delta t} = \frac{0.01l}{0.5sec} = 0.02l/sec$$
$$A = \pi\frac{d_2}{2} = \pi(\frac{0.002m}{2})^2 = 3.14x10^{-6}m^2 \tag{6}$$

$$Velocity = \frac{\dot{V}}{A} = \frac{2x10^{-5}m^3/s}{3.14x10^{-6}m^2} = 6.3m/s \tag{7}$$

$6.3m/s$ is the velocity of air that is drained from the needle. The required air speed for the syringe is calculated using the area relation of the syringe diameter and the needle diameter.

Syringe diameter = A_1 and Syringe Velocity = V_1, Needle Diamater = A_2 and Needle Velocity = V_2

$A_1V_1 = A_2V_2$

$(0.02)^2V_1 = (0.002)^26.3m/s$

$V_1 = 0.063m/sec$

The motor velocity is calculated as 0.098 $\frac{rad}{s}$ which is enough to provide 10 ml air in 0.5 sec.

$$w = (2\pi f)w = \frac{2\pi}{0.66} = 9.5\frac{rad}{s} \tag{8}$$

$$V = (wr) \tag{9}$$

$$V = 9.5\frac{rad}{s}x0.0104m = 0.098\frac{m}{s} \tag{10}$$

The required force measured using digital weight scale. The force was applied with digital weight scale, and the required force was 850 gram, 8.33 newton. The selected motor can apply 12 kg force for 1 cm distance.

The rack and pinion designed according to required force and speed. The circular pitch diameter of the pinion is 36 mm. If the mechanical loss is not considered, that mechanism can provides 6.66 kgf.

T = radius of circular pitch x force

$$12kg1cm = F1.8cm, F = 6.66kg \tag{11}$$

The number of teeth of rack is 24. When the motor turns 1 degree, rack moves 0.8 mm. Motor turns 0.11 sec/60 degree according to its data-sheet. In the best circumstances, motor provides 18.8 mm linear motion in 0.11 sec.

The number of theeth of rack = 24. Circular pitch is 15 degree. When motor turns one pitch, linear motion of rack;

$$(36mmxpi)/24mm = 4.71mm(60/15) * 4.71 = 18.84mm \tag{12}$$

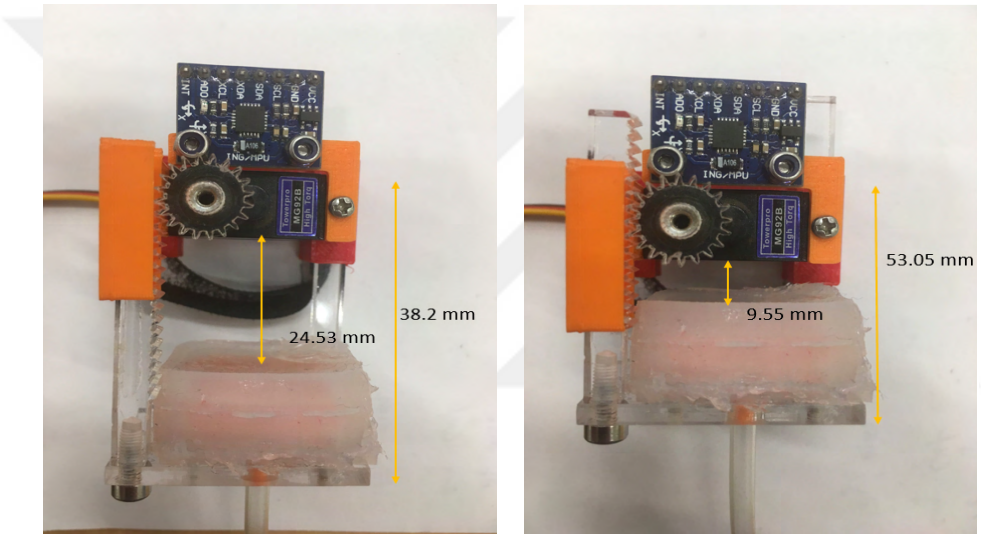
The distance between 10 ml and 15 ml on syringe is 17.8 mm, and 60 degree change is 18.8 (no mechanical loss). Although mechanical loss, according to calculation of that pinion design, this mechanism can provide adequate speed. 4 ml was considered to the maximum volume of air for this project because the silicone material inflation is not comfortable to use more than 4 ml air.

4.3 Features and Characterization of the Fingertip Device

4.3.1 Features of the Device

Designed fingertip haptic device weights 39.83 grams. Servomotor, pinion and the apparatus that support the winger weight 24.06 grams, while platform and the pillow weight 15.77 grams.

The dimension of the device is changeable. Figure **??** shows the maximum (24.53) and minimum (9.55) linear motion of the rack. Minimum condition is not adequate for usage. The dimension of the device, 38.55 mm x 38.21-53.05 mm x 40.33 mm.



(a) (b)

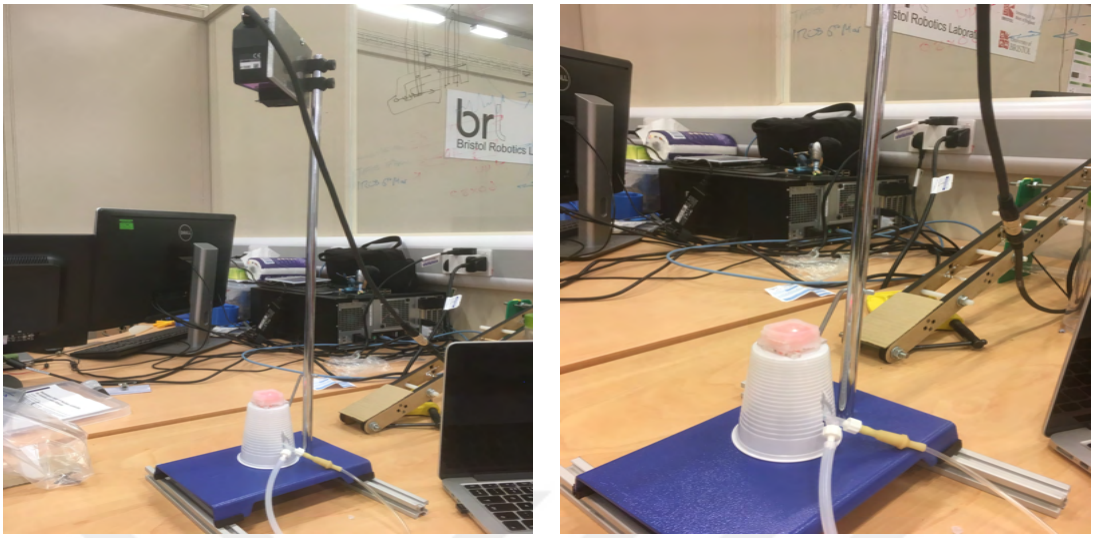
Figure 29: Workspace of the haptic device (a) Maximum distance is shown(degree of the motor is 170) (b) Minimum distance is shown(degree of the motor is 20)

4.3.2 Characterization of the Fingertip Device

In order to correctly use the device, characterization is important. The response and parameters of the device were measured for its characterization.

First, the relation between the indentation and volume of the pillow was determined. The indentation that was provided by the pillow was measured using high accuracy CCD laser displacement sensor. The pillow mechanism was attached to a plastic glass to stabilise it, and they were placed together under the laser displacement sensor (shown in figure 30a and 30b). The volume of air in the pillow was increased

from 0 to 4000 mm^3 inflating the pillow 1000 mm^3 each time. The outer surface increase of the pillow was measured. 30a and 30b show the inflated pillow design.



(a) (b)

Figure 30: The pillow was attached a glass and put under laser displacement sensor

The graph shows Volume-Indentation relation. From 0 to 1000 m^3 the membrane of the pillow increase linearly, then the graph become polynomial because tension in the pillow increases as it was inflated. When the volume of air in the pillow is 4000 mm^3, the indentation is around 8 mm. Figure shows the change of membrane for different volume of air inside of the pillow.

The second measurement is the pressure change inside of the pillow with respect to the different volume of air. The fingertip haptic device, air pressure sensor, and syringe was connected as it shown in 33a. Using the syringe, the pillow was actuated and the analog data that are taken from the pressure sensor was converted to digital data via National Instrument device. Beside this measurement, pushing the pillow till platform, the pressure change was measured. The results are given in the graphs;

The result of volume and pressure measurement is illustrated in 34a. While blue line shows the pressure change inside of the pillow, red line shows the pressure change when the force is applied with finger. The volume of air change between 1000 $mm3$ and 2000 $mm3$ provides high indentation change. Thus, the pressure change is significantly increased from 1000 $mm3$, nearly 1.5 kPa to 6 kPa when the force is applied till the platform. When the silicone inflated more, because the indentation does not change linearly, the pressure also change as a polynomial function. The max pressure is slightly more than 9 kPa, for the second situation.

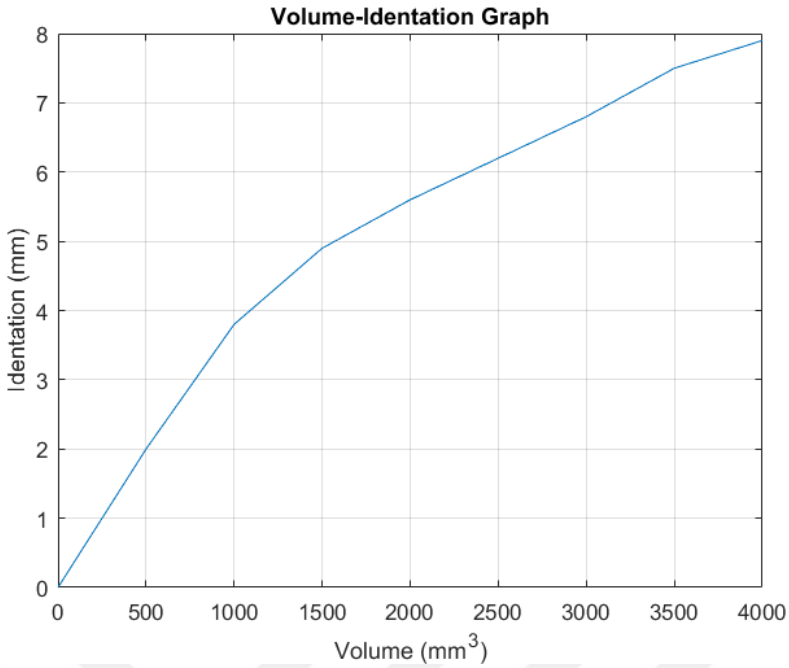
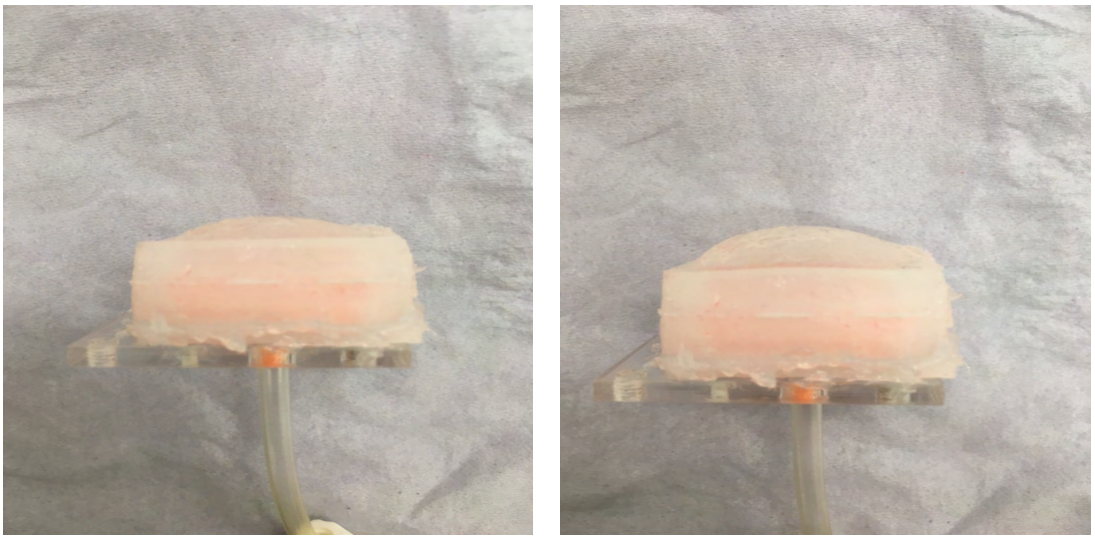


Figure 31: Space limitation of rack and pinion mechanism

From these two condition; volume-indentation and volume-normal pressure were used to identified Indentation-Pressure relation. The 34a gives the relation of Indentation-Pressure. The pressure of air inside the pillow do not change when the indentation change from 3 mm (around 700 $mm3$) in to 5 mm (1500 $mm3$). Because of the gap between platform surface and the pillow membrane, the material was not stretched. When the indentation rises steadily, the increase of the pressure is sharply.

After carrying out these tests, the pressure and indentation relation of the mechanism was created on two people (male-female) to prepare the experiment. The finger of the user was placed to the motor, opposite the middle of the pillow. Servo motor was actuated 5 degree for the each test. It equals to 0.52 mm indentation. The test was carried out between 1000 $mm3$ air (1 ml) to 4000 $mm3$(4 ml). The motor position was started from 120 degree, when there is no pressure, then the silicone mechanism's position was increased until the user feels the platform. The graphs show the finger pressure and indentation relation for male and female candidate. The red line indicates the result which was derived from the female candidate. The measured pressure from female candidate is higher than male candidate because the female candidate's finger surface area is smaller than the male's finger surface are. During the tests, the male candidate commented generally 8.32 mm is reasonable high pressure. Above this causes painful pressure on finger surface. However, female candidate found 9.36 mm indentation reasonable high pressure. For 2000 $mm3$ air and 4000 $mm3$, both candidates experienced similar trend. The upper limit for the candidates are different



(a) (b)

Figure 32: Different volume of air in the pillow (a)2000 mm^3, (b) 4000 mm^3

in each graph. For instance, for graph a, the indentation was 9.36, while for male candidate it was 7.9. This varies because the finger size of candidates is different., and sensed force is different.

4.4 Virtual Environment

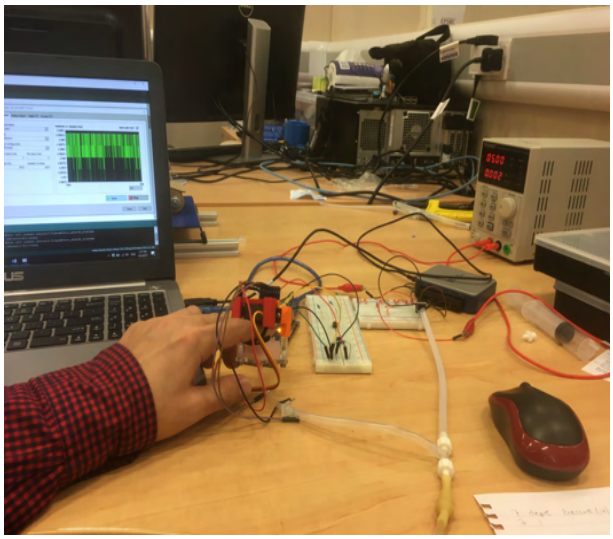
In this section the interaction between haptic interface and virtual environment is mentioned. First, collider functions that are used for physical interaction between virtual environment and the haptic device is explained, and servo motor control is mentioned with respect to collision. After these, motion tracking method is implemented to virtual environment.

4.4.1 Arduino Unity Interaction

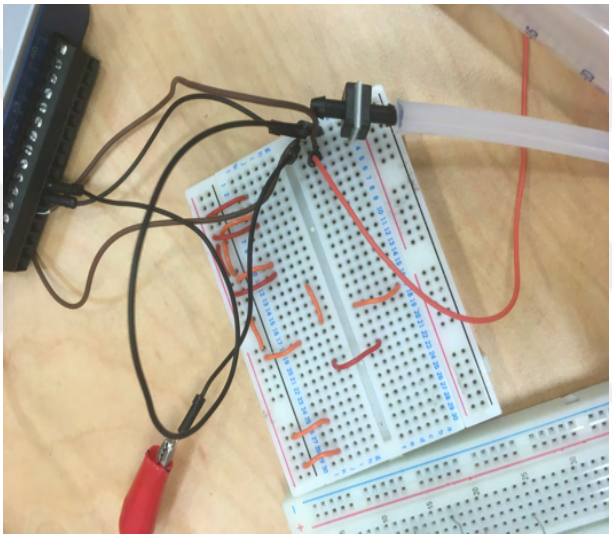
Unity's physic engine offers realistic haptic sense in the real world. In this project rigidbodies and colliders object components were used to realised the virtual force in real life.

Unity offers three collision functions; OnCollisionEnter, OnCollisionStay and OnCollisionExit. Using this functions, and serial communication between Arduino and Unity, interaction of virtual environment and real world was created.

In the designed virtual environment, the manipulated object is spherical red



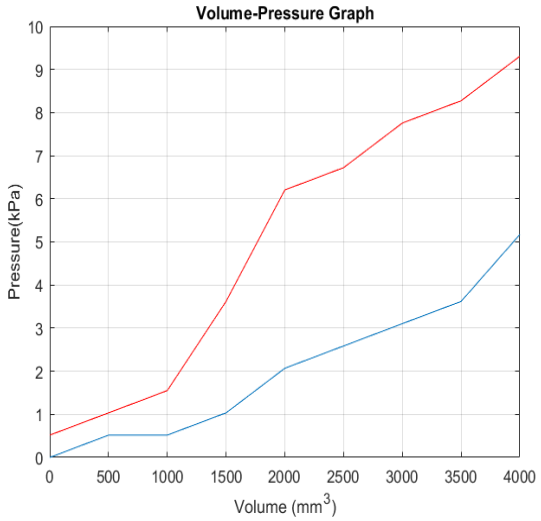
(a)



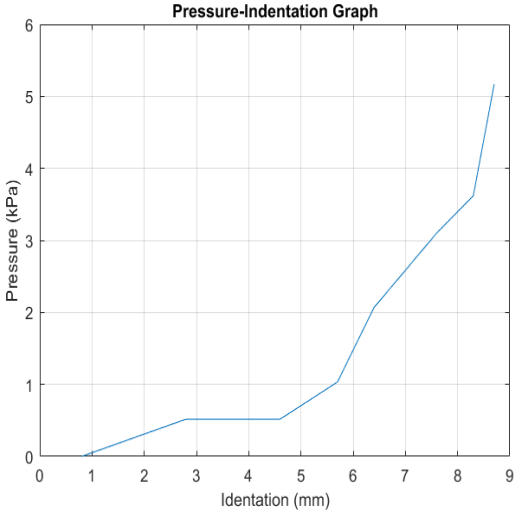
(b)

Figure 33: Measuring the pressure inside of the pillow(a) Connection of syringe, sensor and the pillow, (b) Air pressure sensor

object (fingertip). When it is collides one of the boxes, a character is sent from Unity to Arduino via serial monitor.



(a)

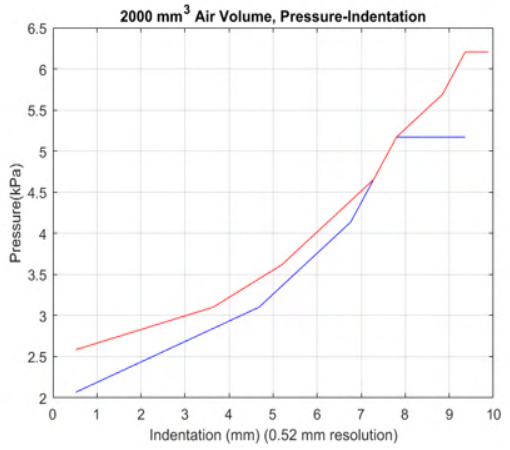


(b)

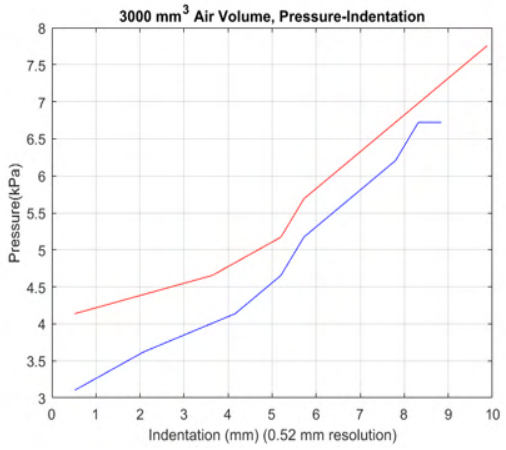
Figure 34: Measuring the pressure inside of the pillow(a) Connection of syringe, sensor and the pillow, (b) Air pressure sensor

```
0 references
private void OnCollisionEnter(Collision collision)
{
        if (collision.collider.name == "Object1")
   {
        rend.material.color = Color.red;
        int azz = Convert.ToInt16(curr_offset_z);
        azz = 7 * azz + 55;
        string ayy = Convert.ToString(azz);
        data.WriteLine("V" + ayy);
    }
```

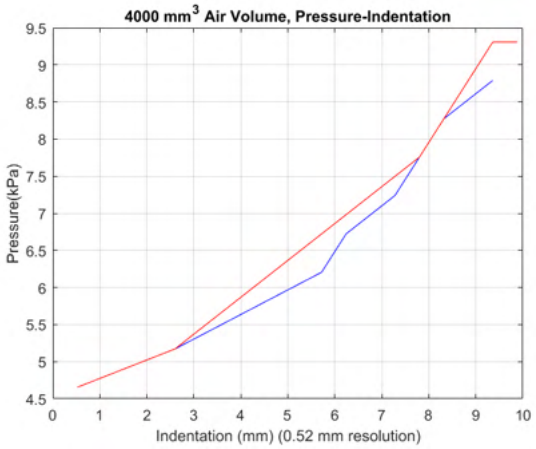
Figure 35: Used OnCollisionEnter and OnCollisionStay codes in Unity



(a)



(b)



(c)

Collision component was added to the fingertip object. If it collides Object1, character ”V” and the y axes position of the object is sent to Arduino. In order to provide stay on collision, OnCollisionStay function is used. This function is used same as OnCollisionEnter function. Unity is sensing still same data to Arduino. However, when the contact is broken, Unity sends different data. ”B” character is sent in addition to ”V” + ayy.

```
0 references
private void OnCollisionExit(Collision collision)
{
    if (collision.collider.name == "Object1")
    {
        int azz = Convert.ToInt16(curr_offset_z);
        azz = 7 * azz + 55;
        string ayy = Convert.ToString(azz);
        data.WriteLine("V" + ayy + "B");

    }
```

Figure 36: Used OnCollisionExit code in Unity

The data from Unity was sent via serial monitor as string. In Arduino, first these data are read as string, then servomotors are actuated according to designed scenario.

In Unity script, azz = 7 * azz + 55; code is used to control the position of the servomotor attached to fingertip haptic device. In a sample scenario, when the user touches the object, the y axes position azz is multiplied with 7, and added to 55. If the fingertip touches the box at 4, in real life motor is actuated

$$azz = 28 + 55 = 83\text{degree}.$$

. This equation is important to provide indentation to users. By changing the value 7, the steps of the indentation can be changed. From the Unity script, position control of the fingertip servo motor can be done.

Motion Tracking

The motion tracking of the fingertip is provided by GY-521 board which includes IMU 6050 sensor. The board is attached to the front section fingertip haptic device. (shown in 29a).

The acceleration feature of the sensor was used for the motion tracking of the fingertip haptic device. The position data are sent from Arduino to Unity. Two different methods can be applied to IMU sensor for motion tracking in Unity. One method is, filtering the raw acceleration data first, and sending them as a position data to Unity. This method requires filtering process, Kalman filter or complementary filter. The neat information that was provided by the User Debra, on the website geekmomprojects was viewed about filtering methods. Complementary filter consists of approximation of both the accelerometer and gyroscopes data. This can provide accurate motion tracking. Because this project is prototype, high sensitive position control was not considered .

The other method is removing the noise of the sensor in Unity. The method is similar high pass filter idea. Small amount of changes were filtered. The raw acceleration values, come from IMU sensor, are between -20000 and 2000 according to test of the sensor. Normalization is required to reduce the noise of acceleration data. In Unity script these values were reduced multiplying with 0.0002. This number was decided trial and error method. When this value was chosen 0.05 the control of the object in virtual environment was not easy because the high distortion. When the normalizer was chosen low such as 0.00005, the motion of the object is slow. The object does not response the motion of the IMU sensor appropriately. Therefore, normalizer of the IMU sensor is 0.0002.

After deciding normalize value, high pass filter was done. Although the sensor is not move, the raw data of acceleration changes. This causes to position change in the Unity. In order to prevent it following code was used.

```
if (Mathf.Abs(ax) - 1 < 0) ax = 0;
```

Figure 37: 3D design of second mould

Moreover, to move the fingertip in Unity with constant speed, Time.deltaTime function was used. This function is also necessary because in the project the acceleration values are added and subtracted every frame changes. Time.deltaTime object is moved 1 meter per second rather than 1 meter per frame. In the below code, moveSpeed helps to set the velocity of object in Unity.

curr_offset_x+=$ax *$ Time.moveSpeed $* Time.deltaTime$

In Unity scene, the object can move out of the camera view. In order to solve this problem, translation was limited for each axes of the spherical object. In the below code, the position of the spherical object was restricted between -3 and 10 on x-axes in Unity scene.

```
float clampedX = Mathf.Clamp(curr_offset_x, -3.0f, 10.0f);

curr_offset_x = clampedX;
```

Figure 38: The position of the object was limited

$curr_offset_x, curr_offset_y and, curr_offset_z$ values were implemented as a translation axes of fingertip(spherical object) in Unity.

The raw data were sent from Arduino Mega 2560 to Unity via serial port, using I2C protocol. Sensor data is written serial port as string. In unity, these data are divided for each axes and converted from string to float.

4.4.2 Test Scenario

The evaluation of the effectiveness of the haptic device can be tested using virtual environment. The scenario is about distinguishing the softness of different objects in virtual environment.

4.5 Softness Identification of Objects in Virtual Environment

The task consisted in identifying the softness of three different objects in virtual environment. The figure illustrates the three turquoise coloured objects. They have the same colour and size. However, each object has different softness. The aim of designing the objects same size and colour is, testing the ability of device more accurate.

In the figure red spherical object symbolises the fingertip which is controlled by the user using IMU sensor. When the user tilts the device to forward, the spherical object moves down. When the sensor is tilted to backward, the spherical object moves up. In the (**Figure 40a**), the user moves the fingertip up in the virtual environment.In the figure (**Figure 40b**), the user changes the position of the fingertip to down.

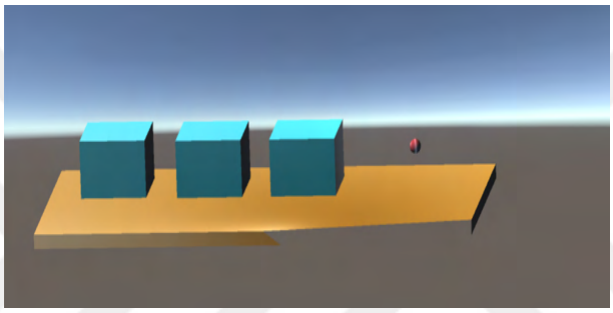
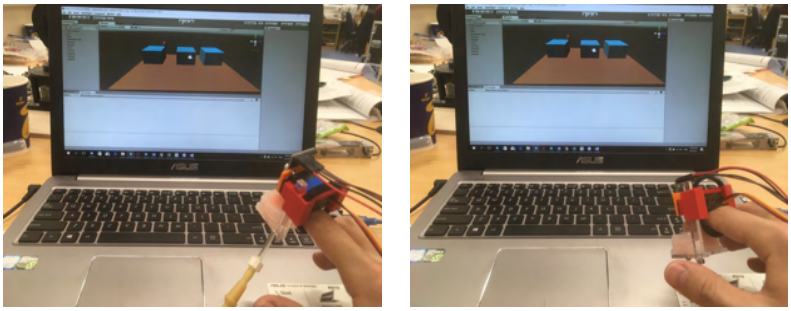


Figure 39: Virtual Environment which was designed for experiment



(a) (b)

Figure 40: The position change of fingertip in the Virtual Environment (a) Moving up the fingertip, (b) Moving down the fingertip

When the spherical object touches one of these three boxes, then the volume of air in the silicone material is changed, and the rack mechanism moves up. If the object is not soft, the silicone material is not inflated. The silicone material is inflated, and rack is moved up to provide different softness. For each virtual object, silicone material has different amount of volume. When user changes the position of the spherical object on y axes to down, it means that the fingertip sinks. In order to provide sinking

sense, the platform moves up in a controlled manner. The device does not provide binary feedback; thus the user senses the softness of materials. However, if the object is rigid, the haptic device apply binary force because there is no air inside of silicone material. After touching each objects, the user will classify the softness of objects.

5 Testing

5.1 Experiment

The experiment was conducted to scale the softness that the device provides relative to the control of the volume of air inside of the pillow and the position of the rack. The aim is to compare the same amount of pressure that obtained for different volume of air. For instance, 3.5 kPa pressure can be provided with 2 ml and 3 ml volume of air. However, the indentation for 2 ml in that scenario is 5.2 mm, while for 3 ml the indentation is 2.08 mm. In order to find an appropriate softness scale for the device, and compare them with measured data, the experience of the users with the haptic device was investigated.

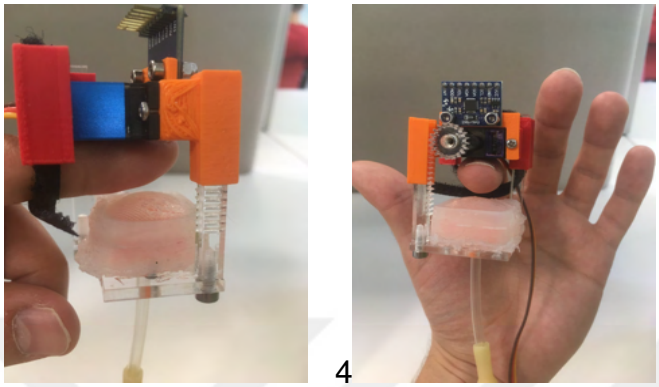
According to graph which was shared in section 4, the rack position and the volume of air were determined for the different conditions. The nine conditions were created as it showed in the table. In the graph 35a, the pressure change is between 2-5.5 kPa for both candidates. In order to compare the same amount of pressure for different volume of air 5 kPa was selected. Also, 3.5 and 4.5 kPa were measured for both 2 ml and 3 ml. In each scenario has own code in the table, i.e. 1.1 means 3.5 kPa pressure and 2 ml volume air. Also 5.2 mm indicates the indentation. If the candidates press the pillow 5.2 mm when the volume of air is 2000 $mm3$, the pressure will be 3.5 kPa.

	2 ml	3 ml	4 ml
3.5 kPa	1.1 (5.2 mm)	1.2 (2.08 mm)	x
4.5 kPa	2.1 (7.28 mm)	2.2 (5.2 mm)	x
5 kPa	3.1 (7.8 mm)	3.2 (5.72 mm)	3.3 (2.08 mm)
7 kPa	x	4.2 (8.32 mm)	4.3 (6.76 mm)

Table 3

Five candidates (3 males, 2 females) attended the experiment. The device

was attached to the candidates' finger as it shown in the 41a and 41b . Before starting the experiment, each candidate touched the pillow, gently or until the platform. The volume of air was changed from 0 to 4000 $mm3$ during this test. Candidates scaled the softness of the pillow relative to their own touch sense, from 1 (hard) to 5 (softness) for different kinds of conditions. When they are ready, the experiment was started.



4

(a) (b)

Figure 41: How the device was attached to the finger (a) , (b)

Ten conditions were experimented by candidates. Nine conditions are from the table. In the additional condition, the pillow does not have air. On the result table, the order of conditions are showed at the first column. The experiment start with 3.1 and finishes with 3.2. The order of the conditions were determined randomly.

Each scenario was written as a loop in Arduino Mega 2560. The program was started with the first scenario (4.2). The rack moved 8.32 mm up, and exerted the force to the finger of the candidate. After waiting 1 second, rack moved down. Between each scenario there was 4 seconds break for candidates to tell their sense scale.

5.2 Results

The table 4 illustrates the result of the experiment.It is noticeable that all candidates scored "no air", condition 1. In contrast, condition 3.3 was ranked as the softest in general trend. When the pillow inflated with 3000 $mm3$ air, it provided soft material sense according to the softness rank of candidates. When the conditions 1.2, 2.2, 3.2 and 4.2 are compared, condition 1.2 was voted as softer than other conditions for 3000 $mm3$. Condition 3.2 followed this.

Other conditions' score vary according to candidates' touch sense. Condition

3.1, 3.2 and 3.3 provide similar pressure but they have different indentation, and different amount of volume. According to candidate's scaling, 3.3 provides the softness among these three conditions. Moreover, condition 3.3 was scaled the softness among ten conditions. 3.1 were voted the as hard. It is considered that, when the indentation increases, the force that the pillow exerts the finger increase, and candidates sensed as hard. Contrast, 4000 $mm3$ should be more rigid, but the indentation is 2.8. Thus, candidates sensed the conditions of 4000 $mm3$ softer than other conditions.

When the volume of air inside in the pillow rose and the indentation increased, the candidates commented that the touch sense was hard. The conditions 1.1, 2.1 and 3.1 are example of this inference. While the majority of the candidates scored 1.1 as soft, this trend fell gradually. Only candidate three scaled 3.1 softer than 1.1 and 1.2.

	C1	C2	C3	C4	C5
3.1	2	1	3	1	2
2.2	4	2	4	2	3
4.3	3	2	5	2	2
2.1	2	1	2	3	2
1.1	3	2	2	4	4
no air	1	1	1	1	1
4.2	2	3	3	1	2
3.3	5	4	4	4	3
1.2	4	3	4	5	3
3.2	4	3	5	2	2

Table 4

6 Conclusion and Future Work

6.1 Conclusion

This thesis presented the investigation for the development of the fingertip haptic device. The research focused on the design of a new fingertip haptic device, and interaction of the device with a virtual environment application.

The background research question of the thesis is, how users can sense the various levels of stiffness of different object in virtual environment. In section 2, the

background information of different types of haptic devices were presented. Generally, haptic devices exert force, but they do not provide the indentation of different sense of soft material. In answer to this, an inflated silicone material which is called ”pillow’, was designed. 3D printed different moulds were designed, and selected the adequate design. The design of the mould is crucial to provide various range of softness. Also, a platform was designed and it was placed into the pillow. The aim of this, not only provide soft material sense, but also offer hard material sense. This developed system was attached to a rack and pinion mechanism. The servomotor of the mechanism actuate the rack upward to exert force to the fingertip of the user. The mechanism also offers adequate motion resolution.

In order to inflate the pillow, a syringe pump was designed. The mechanism is actuated a position controlled servomotor. It was designed rather than buying a linear actuator because it is more compact, and the mechanism actuates the syringe quickly. The pillow mechanism was actuated using plastic tubing. After this, the motion tracking was implemented to the device. Inertial Measurement Sensor (IMU) was selected for this object. The acceleration data was used to control the position of the fingertip object in virtual environment, Unity. Unity offers adequate features to create real life and virtual reality application. In unity, the application that covers three boxes and one spherical object was designed. In determined scenario, boxes are different materials with various softness, the spherical object is the fingertip. When the user touches one of the boxes in virtual environment, syringe pump inflates the pillow, and rack moves upward. If the user sinks his/her finger in the objects, according to position of the spherical object, rack exerts more force to finger.

After designing the virtual reality application, the characterization of the device was carried out. The volume-indentation, and pressure-indentation relation were investigated. In addition to this, device was tested on two people, and pressure-volume character of the device determined. In the experiment section, using the data of tests, nine conditions were created. The volume-pressure-indentation relation were compared. The experiment was conducted among five people. The device exerted force the users’ finger, and they scaled the softness relative to their own touch sense. General trend of the device was depicted.

One of the challenges of the project was designing inflated material. Especially, placing a rigid material into a soft material is not easy. The silicone material that is used for fabrication of the pillow, offers satisfying solution for air leakage problem. Also, rack and pinion mechanisms were produced with acrylic material using laser-cutting. The pinion of the fingertip device is accurate. However, if a pinion design requires high number of small teeth, then different methods can be more beneficial than laser-cut.

Moreover, motion tracking can be developed and made more accurate. In this thesis, motion tracking is adequate to take desired response, but it does not highly accurate finger motion. Unity includes appropriate functions to provide the physical interaction between the virtual reality and real life. The object components, and easy control of it helped to create virtual environment.

6.2 Future Work

One of the future work of this project is improving the hardware. The system was actuated by pinion. There is a passive stick at the opposite side of the rack. Because of the passive stick, when the platform moves upward, the other side of the platform may not move the same position. The passive pinion could be useful for the passive stick to move it to the same position. Also, the support which was designed to hold the user's finger would be comfortable. There is no soft material between motor and the finger. If motor is actuated this causes excessive force on the upper surface of finger.

The device would be tested in virtual environment with different scenarios. Two scenarios can be the future work of this thesis. One scenario is developing the scenario that was mentioned in the section 4. The object surface change can be added to provide visual haptic sense. This would help to users to scale the softness level of haptic feedback that is exerted by the device. Additionally, the softness of the object in real life can be emulated to the virtual environment. This application would be helpful to test the imitation capabilities of the device.

References

Abushagur, A. A., Arsad, N., Reaz, M. I. & Bakar, A. (2014), 'Advances in bio-tactile sensors for minimally invasive surgery using the fibre bragg grating force sensor technique: A survey', *Sensors* **14**(4), 6633–6665.

Achibet, M. (2015*a*), Contributions to the design of novel hand-based interaction techniques for virtual environments, PhD thesis, INSA de Rennes.

Achibet, M. (2015*b*), Contributions to the design of novel hand-based interaction techniques for virtual environments, PhD thesis, INSA de Rennes.

Balakrishman, R., Ware, C. & Smith, T. (1994), Virtual hand tool with force feedback, *in* 'Conference companion on Human factors in computing systems', ACM, pp. 83–84.

Bianchi, M., Battaglia, E., Poggiani, M., Ciotti, S. & Bicchi, A. (2016), A wearable fabric-based display for haptic multi-cue delivery, *in* 'Haptics Symposium (HAPTICS), 2016 IEEE', IEEE, pp. 277–283.

Bouzit, M., Burdea, G., Popescu, G. & Boian, R. (2002), 'The rutgers master ii-new design force-feedback glove', *IEEE/ASME Transactions on mechatronics* **7**(2), 256–263.

Bric, J. D., Lumbard, D. C., Frelich, M. J. & Gould, J. C. (2016), 'Current state of virtual reality simulation in robotic surgery training: a review', *Surgical endoscopy* **30**(6), 2169–2178.

Chinello, F., Malvezzi, M., Pacchierotti, C. & Prattichizzo, D. (2015), Design and development of a 3rrs wearable fingertip cutaneous device, *in* 'Advanced Intelligent Mechatronics (AIM), 2015 IEEE International Conference on', IEEE, pp. 293–298.

DelBrocco, M. V. (2013), A Virtual Haptics Environment for Simulating Anxiety-Inducing Phenomena, PhD thesis, Case Western Reserve University.

Dipietro, L., Sabatini, A. M. & Dario, P. (2008), 'A survey of glove-based systems and their applications', *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* **38**(4), 461–482.

Fontana, M., Fabio, S., Marcheschi, S. & Bergamasco, M. (2013), 'Haptic hand exoskeleton for precision grasp simulation', *Journal of Mechanisms and Robotics* **5**(4), 041014.

Frisoli, A., Salsedo, F., Bergamasco, M., Rossi, B. & Carboncini, M. C. (2009), 'A force-feedback exoskeleton for upper-limb rehabilitation in virtual reality', *Applied Bionics and Biomechanics* **6**(2), 115–126.

Gabardi, M., Solazzi, M., Leonardis, D. & Frisoli, A. (2016), A new wearable fingertip haptic interface for the rendering of virtual shapes and surface features, *in* 'Haptics Symposium (HAPTICS), 2016 IEEE', IEEE, pp. 140–146.

Gosselin, F., Jouan, T., Brisset, J. & Andriot, C. (2005*a*), Design of a wearable haptic interface for precise finger interactions in large virtual environments, *in* 'Eurohaptics Conference, 2005 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2005. World Haptics 2005. First Joint', IEEE, pp. 202–207.

Gosselin, F., Jouan, T., Brisset, J. & Andriot, C. (2005*b*), Design of a wearable haptic interface for precise finger interactions in large virtual environments, *in* 'Eurohaptics Conference, 2005 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2005. World Haptics 2005. First Joint', IEEE, pp. 202–207.

Gregory, S., Timms, D., Pearcy, M. & Tansley, G. (2009), 'A naturally shaped silicone ventricle evaluated in a mock circulation loop: a preliminary study', *Journal of medical engineering & technology* **33**(3), 185–191.

Gu, X., Zhang, Y., Sun, W., Bian, Y., Zhou, D. & Kristensson, P. O. (2016), Dexmo: An inexpensive and lightweight mechanical exoskeleton for motion capture and force feedback in vr, *in* 'Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems', ACM, pp. 1991–1995.

Heo, P., Gu, G. M., Lee, S.-j., Rhee, K. & Kim, J. (2012), 'Current hand exoskeleton technologies for rehabilitation and assistive engineering', *International Journal of Precision Engineering and Manufacturing* **13**(5), 807–824.

Hernandez-Rebollar, J. L., Kyriakopoulos, N. & Lindeman, R. W. (2002), The acceleglove: a whole-hand input device for virtual reality, *in* 'ACM SIGGRAPH 2002 conference abstracts and applications', ACM, pp. 259–259.

Hirose, M., Hirota, K., Ogi, T., Yano, H., Kakehi, N., Saito, M. & Nakashige, M. (2001), Hapticgear: the development of a wearable force display system for immersive projection displays, *in* 'Virtual Reality, 2001. Proceedings. IEEE', IEEE, pp. 123–129.

Homberg, B., Katzschmann, R., Dogar, M. & Rus, D. (2015), Haptic identification of objects using a modular soft robotic gripper. ieee, *in* 'RSJ International Conference on Intelligent Robots and Systems (IROS)'.

Iqbal, J., Khan, H., Tsagarakis, N. G. & Caldwell, D. G. (2014), 'A novel exoskeleton robotic system for hand rehabilitation–conceptualization to prototyping', *Biocybernetics and biomedical engineering* **34**(2), 79–89.

Kim, D., Hilliges, O., Izadi, S., Butler, A. D., Chen, J., Oikonomidis, I. & Olivier, P. (2012), Digits: freehand 3d interactions anywhere using a wrist-worn gloveless sensor, *in* 'Proceedings of the 25th annual ACM symposium on User interface software and technology', ACM, pp. 167–176.

Kim, J.-H., Thang, N. D. & Kim, T.-S. (2009), 3-d hand motion tracking and gesture recognition using a data glove, *in* 'Industrial Electronics, 2009. ISIE 2009. IEEE International Symposium on', IEEE, pp. 1013–1018.

Kim, M., Jang, I., Lee, Y., Lee, Y. & Lee, D. (2016), Wearable 3-dof cutaneous haptic device with integrated imu-based finger tracking, *in* 'Ubiquitous Robots and Ambient Intelligence (URAI), 2016 13th International Conference on', IEEE, pp. 649–649.

Kry, P. G. & Pai, D. K. (2006), Interaction capture and synthesis, *in* 'ACM Transactions on Graphics (TOG)', Vol. 25, ACM, pp. 872–880.

Kuhnapfel, U., Cakmak, H. K. & Maass, H. (2000), 'Endoscopic surgery training using virtual reality and deformable tissue simulation', *Computers & graphics* **24**(5), 671–682.

Lelieveld, M. & Maeno, T. (2006), Design and development of a 4 dof portable haptic interface with multi-point passive force feedback for the index finger, *in* 'Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on', IEEE, pp. 3134–3139.

Lemole Jr, G. M., Banerjee, P. P., Luciano, C., Neckrysh, S. & Charbel, F. T. (2007), 'Virtual reality in neurosurgical education: part-task ventriculostomy simulation with dynamic visual and haptic feedback', *Neurosurgery* **61**(1), 142–149.

Leonardis, D., Solazzi, M., Bortone, I. & Frisoli, A. (2015), A wearable fingertip haptic device with 3 dof asymmetric 3-rsr kinematics, *in* 'World Haptics Conference (WHC), 2015 IEEE', IEEE, pp. 388–393.

Lim, S.-C., Lee, H.-K. & Park, J. (2015), 'Role of combined tactile and kinesthetic feedback in minimally invasive surgery', *The International Journal of Medical Robotics and Computer Assisted Surgery* **11**(3), 360–374.

MacLean, K. E. & Roderick, J. B. (1999), Smart tangible displays in the everyday world: A haptic door knob, *in* 'Advanced Intelligent Mechatronics, 1999. Proceedings. 1999 IEEE/ASME International Conference on', IEEE, pp. 203–208.

Martin, S. & Hillier, N. (2009), Characterisation of the novint falcon haptic device for application as a robot manipulator, *in* 'Australasian Conference on Robotics and Automation (ACRA)', Citeseer, pp. 291–292.

Massie, T. H., Salisbury, J. K. et al. (1994), The phantom haptic interface: A device for probing virtual objects, *in* 'Proceedings of the ASME winter annual meeting, symposium on haptic interfaces for virtual environment and teleoperator systems', Vol. 55, Citeseer, pp. 295–300.

Minamizawa, K., Fukamachi, S., Kajimoto, H., Kawakami, N. & Tachi, S. (2007), Gravity grabber: wearable haptic display to present virtual mass sensation, *in* 'ACM SIGGRAPH 2007 emerging technologies', ACM, p. 8.

Moeslund, T. B. (2001), 'Interacting with a virtual world through motion capture', *Virtual Interaction: Interaction in Virtual Inhabited 3D Worlds, Qvortrup, L.(ed), Springer Verlag, London* .

Monroy, M., Oyarzabal, M., Ferre, M., Campos, A. & Barrio, J. (2008), 'Masterfinger: Multi-finger haptic interface for collaborative environments', *Haptics: Perception, Devices and Scenarios* pp. 411–419.

Ott, R. (2009), 'Two-handed haptic feedback in generic virtual environments'.

Pacchierotti, C., Chinello, F., Malvezzi, M., Meli, L. & Prattichizzo, D. (2012), Two finger grasping simulation with cutaneous and kinesthetic force feedback, *in* 'International Conference on Human Haptic Sensing and Touch Enabled Computer Applications', Springer, pp. 373–382.

Pacchierotti, C., Sinclair, S., Solazzi, M., Frisoli, A., Hayward, V. & Prattichizzo, D. (2017), 'Wearable haptic systems for the fingertip and the hand: taxonomy, review, and perspectives', *IEEE Transactions on Haptics* .

Pacchierotti, C., Tirmizi, A. & Prattichizzo, D. (2014), 'Improving transparency in teleoperation by means of cutaneous tactile force feedback', *ACM Transactions on Applied Perception (TAP)* **11**(1), 4.

Park, Y., Jo, I. & Bae, J. (2016), Development of a dual-cable hand exoskeleton system for virtual reality, *in* 'Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on', IEEE, pp. 1019–1024.

Perng, J. K., Fisher, B., Hollar, S. & Pister, K. S. (1999), Acceleration sensing glove (asg), *in* 'Wearable Computers, 1999. Digest of Papers. The Third International Symposium on', IEEE, pp. 178–180.

Peters, M., Mackenzie, K. & Bryden, P. (2002), 'Finger length and distal finger extent patterns in humans', *American journal of physical anthropology* **117**(3), 209–217.

Peters, R. M., Hackeman, E. & Goldreich, D. (2009), 'Diminutive digits discern delicate details: fingertip size and the sex difference in tactile spatial acuity', *Journal of Neuroscience* **29**(50), 15756–15761.

Prattichizzo, D., Chinello, F., Pacchierotti, C. & Malvezzi, M. (2013), 'Towards wearability in fingertip haptics: a 3-dof wearable device for cutaneous force feedback', *IEEE Transactions on Haptics* **6**(4), 506–516.

Salisbury, J. K. & Srinivasan, M. A. (1997), 'Phantom-based haptic interaction with virtual objects', *IEEE Computer Graphics and Applications* **17**(5), 6–10.

Scheggi, S., Meli, L., Pacchierotti, C. & Prattichizzo, D. (2015), Touch the virtual reality: using the leap motion controller for hand tracking and wearable tactile devices for immersive haptic rendering, *in* 'ACM SIGGRAPH 2015 Posters', ACM, p. 31.

Schorr, S. B. & Okamura, A. (2017), 'Three-dimensional skin deformation as force substitution: Wearable device design and performance during haptic exploration of virtual environments', *IEEE Transactions on Haptics* .

Solazzi, M., Frisoli, A., Salsedo, F. & Bergamasco, M. (2007), A fingertip haptic display for improving local perception of shape cues, *in* 'EuroHaptics Conference, 2007 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics 2007. Second Joint', IEEE, pp. 409–414.

Speich, J. E. & Rosen, J. (2004), 'Medical robotics', *Encyclopedia of biomaterials and biomedical engineering* **983**, 993.

Springer, S. L. & Ferrier, N. J. (2002), 'Design and control of a force-reflecting haptic interface for teleoperational grasping', *TRANSACTIONS-AMERICAN SOCIETY OF MECHANICAL ENGINEERS JOURNAL OF MECHANICAL DESIGN* **124**(2), 277–283.

Srinivasan, M. A. & Basdogan, C. (1997), 'Haptics in virtual environments: Taxonomy, research status, and challenges', *Computers & Graphics* **21**(4), 393–404.

Tavares, R., Souse, P. J., Abreu, P. & Restivo, M. T. (2016), Virtual environment for instrumented glove, *in* 'Remote Engineering and Virtual Instrumentation (REV), 2016 13th International Conference on', IEEE, pp. 311–312.

Tsetserukou, D., Sato, K. & Tachi, S. (2010), Exointerfaces: novel exosceleton haptic interfaces for virtual reality, augmented sport and rehabilitation, *in* 'Proceedings of the 1st Augmented Human International Conference', ACM, p. 1.

Tzemanaki, A. (2016), Anthropomorphic surgical system for soft tissue robot-assisted surgery, PhD thesis, University of the West of England.

Van der Meijden, O. & Schijven, M. (2009), 'The value of haptic feedback in conventional and robot-assisted minimal invasive surgery and virtual reality training: a current review', *Surgical endoscopy* **23**(6), 1180–1190.

Vander Poorten, E., Demeester, E. & Lammertse, P. (2012), Haptic feedback for medical applications, a survey, *in* 'Proceedings Actuator 2012'.

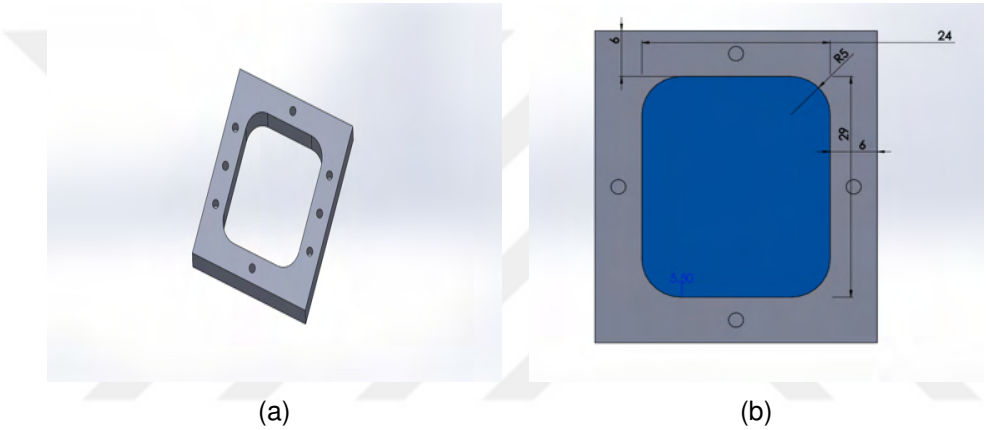
Wang, R. Y. & Popovic, J. (2009), 'Real-time hand-tracking with a color glove', *ACM Transactions on Graphics* **28**(3).

Wijntjes, M. W., Sato, A., Hayward, V. & Kappers, A. M. (2009), 'Local surface orientation dominates haptic curvature discrimination', *IEEE transactions on haptics* **2**(2), 94–102.

Yap, H. K., Lim, J. H., Nasrallah, F., Goh, J. C. & Yeow, R. C. (2015), A soft exoskeleton for hand assistive and rehabilitation application using pneumatic actuators with variable stiffness, *in* 'Robotics and Automation (ICRA), 2015 IEEE International Conference on', IEEE, pp. 4967–4972.

Zhou, M. & Ben-Tzvi, P. (2015), 'Rml glove—an exoskeleton glove mechanism with haptics feedback', *IEEE/ASME Transactions on Mechatronics* **20**(2), 641–652.

7 Appendix



(a) (b)

Figure 42: First Mould Design (a)Unit 2, (b)Unit 1

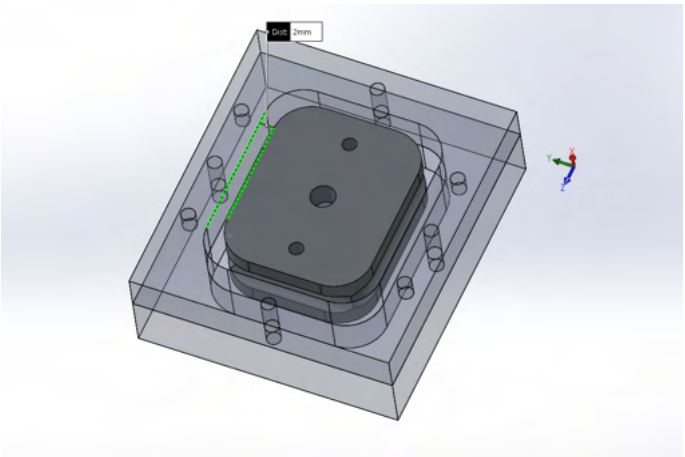


Figure 43: The distance from the side of the unit3 and mould.

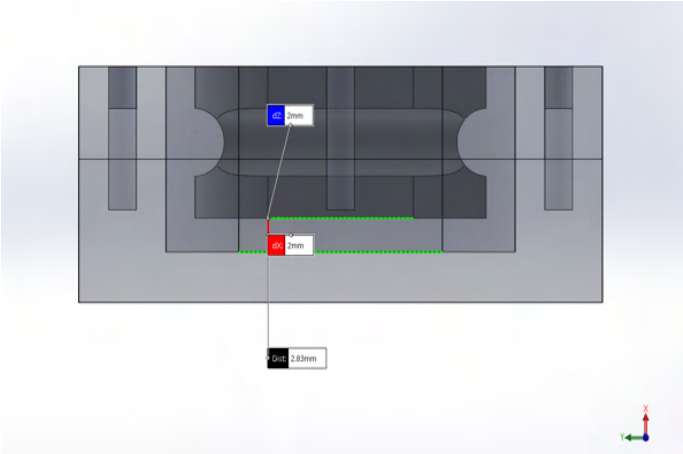
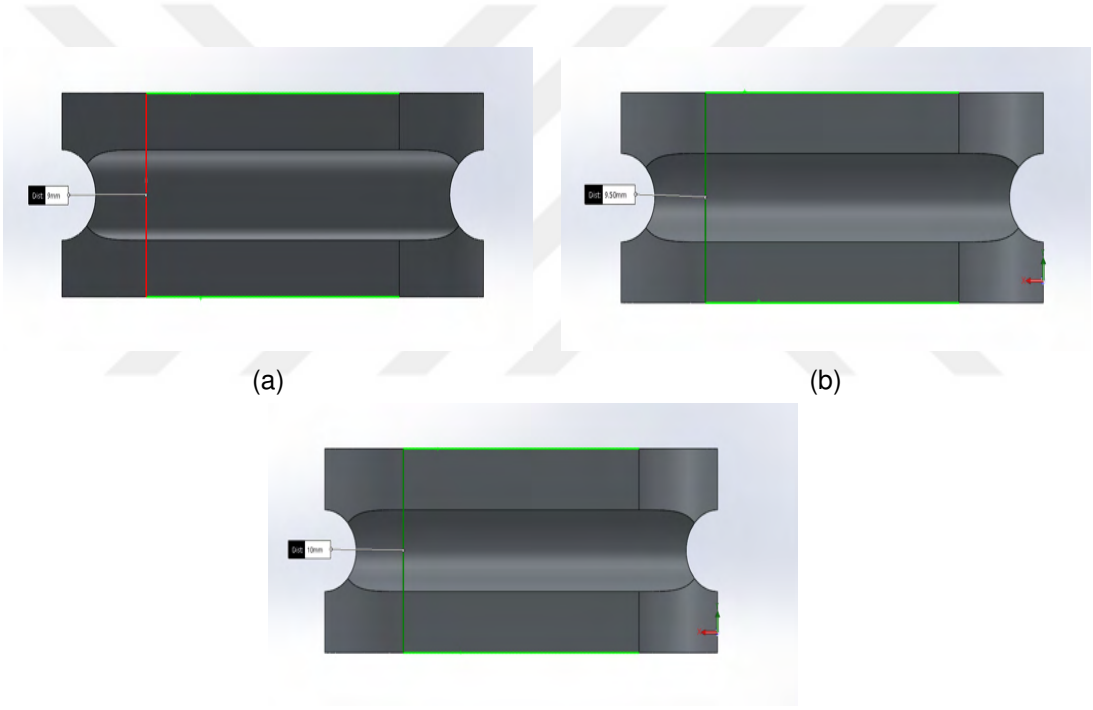


Figure 44: The gap showes the thickness of the designed pillow for first design of Unit 3



(a) (b)

(c)

Figure 45: Different design ideas for Unit 3, and their depth (a) Depth is 9mm, (b) Depth is 9.5 mm (c) Depth is 10 mm

Arduino Experiment Code

```
#include <Servo.h>

Servo myservo;  // create servo object to control a servo
Servo myservo2;
// twelve servo objects can be created on most boards
```

```
int posss = 0;
int pos = 0;    // variable to store the servo position
String dataunity;
String anglemotor;
int angle;
int16_t  acz_ofset;
 String poss;
char buf[20];
 char * pch;
static char outstr[15];


#include<Wire.h>


void setup(){

 Serial.begin(9600);
  myservo.write(100); //  initial value
  myservo.attach(9);  // attaches the servo on pin 9 to the servo object


}

char command[1024];
char commandBuffer[128];
int commandBufferSize = 0;

void loop(){

     readCommand();
  String ch = command;
  ch = Serial.read();
  if(command[0] == 'V')
    {
        delay(5);

     String line1 = String(command[1]);
     String line2 = String(command[2]);
      String line3 = String(command[3]);

     String totalLine = line1 + line2 + line3;

     int pos =  totalLine.toInt();

//SCENARIO 1

```

```
 myservo.write(140);
 delay(100);

    for (posss = 140; posss >=45 ; posss -= 1) { // goes from 140 degrees
   to 45 degrees
    myservo.write(posss);
    delay(30);  // waits 30 ms for the servo to reach the position
  }

delay(1000); // fingertip device exerts force

  for (posss = 45; posss <= 140; posss += 1) { // goes from 45 degrees to
   140 degrees
    // in steps of 1 degree
    myservo.write(posss);  // tell servo to go to position in variable '
   possss'
    delay(5);     // waits 5ms for the servo to reach the position
  }

delay(4000);  // In this break, the user will scale the softness of the
   haptic feedback

//SCENARIO 2

 myservo.write(140);
 delay(100);

    for (posss = 140; posss >=70 ; posss -= 1) { // goes from 180 degrees
   to 0 degrees
    myservo.write(posss);   // tell servo to go to position in variable '
   pos'
    delay(30);     // waits 15ms for the servo to reach the position
  }

delay(1000);

  for (posss = 70; posss <= 140; posss += 1) { // goes from 0 degrees to
   180 degrees
    // in steps of 1 degree
    myservo.write(posss); // tell servo to go to position in variable 'pos'
    delay(5);     // waits 15ms for the servo to reach the position
  }

delay(4000);
 //SCENARIO 3

 myservo.write(140);
```

```
  delay(100);

    for (posss = 140; posss >=55 ; posss -= 1) { // goes from 180 degrees
   to 0 degrees
    myservo.write(posss);              // tell servo to go to position in
   variable 'pos'
    delay(30);                       // waits 15ms for the servo to reach
   the position
  }

delay(1000);

  for (posss = 55; posss <= 140; posss += 1) { // goes from 0 degrees to
   180 degrees
    // in steps of 1 degree
    myservo.write(posss);  // tell servo to go to position in variable 'pos
   '
    delay(5);     // waits 15ms for the servo to reach the position
  }

delay(4000);
  //SCENARIO 4

 myservo.write(140);
  delay(100);

    for (posss = 140; posss >=50 ; posss -= 1) { // goes from 180 degrees
   to 0 degrees
    myservo.write(posss);  // tell servo to go to position in variable 'pos
   '
    delay(30);    // waits 15ms for the servo to reach the position
  }

delay(1000);

  for (posss = 50; posss <= 140; posss += 1) { // goes from 0 degrees to
   180 degrees
    // in steps of 1 degree
    myservo.write(posss); // tell servo to go to position in variable 'pos'
    delay(5);  // waits 15ms for the servo to reach the position
  }

delay(4000);
  //SCENARIO 5

 myservo.write(140);
  delay(100);
```

```
133
      for (posss = 140; posss >=70 ; posss -= 1) { // goes from 180 degrees
    to 0 degrees
135     myservo.write(posss); // tell servo to go to position in variable 'pos'
     delay(30);   // waits 15ms for the servo to reach the position
137   }

139 delay(1000);

141   for (posss = 70; posss <= 140; posss += 1) { // goes from 0 degrees to
    180 degrees
     // in steps of 1 degree
143     myservo.write(posss);   // tell servo to go to position in variable 'pos
    '
     delay(5);   // waits 15ms for the servo to reach the position
145   }

147 delay(4000);

149 //SCENARION HART
       myservo.write(140);
151   delay(100);

153      for (posss = 140; posss >=40 ; posss -= 1) { // goes from 180 degrees
    to 0 degrees
     myservo.write(posss);   // tell servo to go to position in variable 'pos
    '
155     delay(30);    // waits 15ms for the servo to reach the position
   }
157
158 delay(1000);
159
   for (posss = 40; posss <= 140; posss += 1) { // goes from 0 degrees to
    180 degrees
161     // in steps of 1 degree
     myservo.write(posss);   // tell servo to go to position in variable 'pos
    '
163     delay(5);   // waits 15ms for the servo to reach the position
   }
165
166 delay(4000);
167   //SCENARIO 6

169  myservo.write(140);
   delay(100);
171
      for (posss = 140; posss >=40 ; posss -= 1) { // goes from 180 degrees
```

```
    to 0 degrees
173     myservo.write(posss); // tell servo to go to position in variable 'pos'
        delay(30);    // waits 15ms for the servo to reach the position
175   }

177 delay(1000);

179   for (posss = 40; posss <= 140; posss += 1) { // goes from 0 degrees to
      180 degrees
        // in steps of 1 degree
181     myservo.write(posss);    // tell servo to go to position in variable '
      pos'
        delay(5);    // waits 15ms for the servo to reach the position
183   }
  delay(4000);
185 //SCENARIO 7

187  myservo.write(140);
    delay(100);
189
       for (posss = 140; posss >=100 ; posss -= 1) { // goes from 180 degrees
      to 0 degrees
191     myservo.write(posss);     // tell servo to go to position in variable '
      pos'
        delay(30);      // waits 15ms for the servo to reach the position
193   }

195 delay(1000);

197   for (posss = 100; posss <= 140; posss += 1) { // goes from 0 degrees to
      180 degrees
        // in steps of 1 degree
199     myservo.write(posss);    // tell servo to go to position in variable '
      pos'
        delay(5);      // waits 15ms for the servo to reach the position
201   }

203 delay(4000);
    //SCENARIO 8
205
  myservo.write(140);
207   delay(100);

209       for (posss = 140; posss >=100 ; posss -= 1) { // goes from 180 degrees
      to 0 degrees
        myservo.write(posss);     // tell servo to go to position in variable '
      pos'
```

```
211     delay(30);       // waits 15ms for the servo to reach the position
    }
213
delay(1000);
215
   for (posss = 100; posss <= 140; posss += 1) { // goes from 0 degrees to
     180 degrees
     // in steps of 1 degree
     myservo.write(posss);  // tell servo to go to position in variable 'pos
     '
219     delay(5);      // waits 15ms for the servo to reach the position
    }
221
delay(4000);
223   //SCENARIO 9

225  myservo.write(140);
   delay(100);
227
      for (posss = 140; posss >=65 ; posss -= 1) { // goes from 180 degrees
     to 0 degrees
     myservo.write(posss);     // tell servo to go to position in variable '
     pos'
     delay(30);        // waits 15ms for the servo to reach the position
231   }

233 delay(1000);

235   for (posss = 65; posss <= 140; posss += 1) { // goes from 0 degrees to
     180 degrees
     // in steps of 1 degree
237     myservo.write(posss);  // tell servo to go to position in variable 'pos
     '
     delay(5);       // waits 5ms for the servo to reach the position
239   }


241
      delay(5);
243         Serial.flush();

245     }
        if(command[0] == 'b')
247     {
          delay(5);
249      myservo.write(100);
     }
251
```

```
    }
```

Arduino Code for Virtual Environment Application

```
#include <Servo.h>

Servo myservo;  // create servo object to control a servo
Servo myservo2;
// twelve servo objects can be created on most boards

int pos = 0;    // variable to store the servo position
String dataunity;
String anglemotor;
int angle;
int16_t  acz_ofset;
 String poss;
char buf[20];
 char * pch;
static char outstr[15];


#include<Wire.h>

const int MPU=0x68;  // I2C address of the MPU-6050
int16_t AcX,AcY,AcZ,Tmp,GyX,GyY,GyZ,aczoffset;

 GyX and others for gyro_normalizer_factor


void setup(){
   Wire.begin();
  Wire.beginTransmission(MPU);
  Wire.write(0x6B);  // PWR_MGMT_1 register
  Wire.write(0);     // set to zero (wakes up the MPU-6050)
  Wire.endTransmission(true);
 Serial.begin(9600);
myservo.write(100); //  initial value motor value
  myservo.attach(9);  // attaches the servo on pin 9 to the servo object
  myservo2.attach(8);

}

char command[1024];
char commandBuffer[128];
int commandBufferSize = 0;
```

```
42
void loop(){
44   fsrReading = analogRead(fsrPin);

46  Wire.beginTransmission(MPU);
    Wire.write(0x3B);  // starting with register 0x3B (ACCEL_XOUT_H)
48  Wire.endTransmission(false);
    Wire.requestFrom(MPU,14,true);  // request a total of 14 registers
50  AcX=Wire.read()<<8|Wire.read();  // 0x3B (ACCEL_XOUT_H) & 0x3C (
      ACCEL_XOUT_L)
    AcY=Wire.read()<<8|Wire.read();  // 0x3D (ACCEL_YOUT_H) & 0x3E (
      ACCEL_YOUT_L)
52  AcZ=Wire.read()<<8|Wire.read();  // 0x3F (ACCEL_ZOUT_H) & 0x40 (
      ACCEL_ZOUT_L)
    Tmp=Wire.read()<<8|Wire.read();  // 0x41 (TEMP_OUT_H) & 0x42
54


56    String acx = String(AcX);
      String acy = String(AcY);
58    String acz = String(AcZ);

60  //Serial.print(AcX); Serial.print(";"); Serial.print(AcY); Serial.print("
      ;"); Serial.print(AcZ); Serial.print(";");Serial.println("");
    Serial.print(acx); Serial.print(";"); Serial.print(acy); Serial.print(";"
      ); Serial.print(acz); Serial.print(";");Serial.println("");
62  Serial.print(";"); Serial.print(gyz); Serial.println("");
    Serial.flush();
64


66     }

68 void serialEvent() {

70   while (Serial.available())
{
72


74    readCommand();
    String ch = command;
76  ch = Serial.read();
    if(command[0] == 'V')
78    {
          delay(5);
80

       String line1 = String(command[1]);
82     String line2 = String(command[2]);
        String line3 = String(command[3]);
```

```
84
      String totalLine = line1 + line2 + line3;
86    int pos =  totalLine.toInt();

88   myservo.write(pos); // tell servo to go to position in variable 'pos'
    myservo2.write(70);     // activate syringe pump
90  delay(5);
    Serial.flush();
92

    }
94


96 // Second Object
 if(command[0] == 'v')
98    {
        delay(5);
100     int angle1 = command[1]*3;
        int angle2 = 90 + angle1;
102
     String line1 = String(command[1]);
104  String line2 = String(command[2]);
      String line3 = String(command[3]);
106
     String totalLine = line1 + line2 + line3;
108
     int pos =  totalLine.toInt();
110  myservo.write(pos); // tell servo to go to position in variable 'pos'
   myservo2.write(80); // activate syringe pump
112
    // myservo.write(angle2);
114  delay(5);
       Serial.flush();
116   }


118
// Third Object
120 if(command[0] == 'c')
    {
122     delay(5);

124   String line1 = String(command[1]);
     String line2 = String(command[2]);
126   String line3 = String(command[3]);

128  String totalLine = line1 + line2 + line3;
     int pos =  totalLine.toInt();
130
```

```
    myservo.write(pos);  // tell servo to go to position in variable 'pos'
   myservo2.write(55);  // activate syringe pump

     delay(5);
       Serial.flush();
    }



//Syringe pump
 if(command[4] == 'B' || command[3] == 'B') {//|| command[4]=='b') the
  position can be 90 or 100. Thus either look command4 or command3
  myservo2.write(60);

  }

 if(command[4] == 'b' || command[3] == 'b') {//|| command[4]=='b') the
  position can be 90 or 100. Thus either look command4 or command3
  myservo2.write(60);

  }

   if(command[0] == 'k') {   //|| command[4]=='b')
   the position can be 90 or 100. Thus either look command4 or command3
    delay(5);
      String line1 = String(command[1]);
     String line2 = String(command[2]);
      String line3 = String(command[3]);

     String totalLine = line1 + line2 + line3;
     int pos =  totalLine.toInt();
  myservo2.write(60);
  myservo.write(90);



  }

}
}
```

Unity Code for Virtual Environment Application

```
using UnityEngine;
```

```
using System.Collections;
//using System.Collections.Generic;
using System.IO.Ports;
using System;
using System.Text;

public class balss : MonoBehaviour
{

    public float moveSpeed = 10f;
    private Rigidbody rbody;
    private Renderer rend;

    public Vector3 impactvariable;
    public float pos;

    public AudioClip clip;

    SerialPort data;

    public GameObject fingertip; // is the gameobject to

    //float normalizer = 0.00020f;  //0.00025f

    /// </summary>

    float curr_angle_x = 0;
    float curr_angle_y = 0;
    float curr_angle_z = 0;

    public float factor = 7; //7


    public bool enableRotation;
    public bool enableTranslation;

    // SELECT YOUR COM PORT AND BAUDRATE
    string port = "COM3";
    int baudrate = 9600;
    int readTimeout = 250;
    private float normalize;


    // Use this for initialization

    void Start()
    {
```

```
49          rbody = GetComponent<Rigidbody>();
            rend = GetComponent<Renderer>();
51          // open port. Be shure in unity edit > project settings > player is
        NET2.0 and not NET2.0Subset
            data = new SerialPort("\\\\.\\" + port, baudrate);
53
            try
55          {
                data.ReadTimeout = readTimeout;
57          }
            catch (System.IO.IOException ioe)
59          {
                Debug.Log("IOException: " + ioe.Message);
61          }

63          data.Open();
            foreach (string str in SerialPort.GetPortNames())
65          {

67              Debug.Log(string.Format("Port : (0)", str));
            }
69
        }
71
        // Update is called once per frame
73      void Update()

75      {
            string dataString = "null received";
77
            if (data.IsOpen)
79          {
                try
81              {
                    dataString = data.ReadLine();
83              }
                catch (System.IO.IOException ioe)
85              {
                    Debug.Log("IOException: " + ioe.Message);
87              }

89          }
            else
91              dataString = "NOT OPEN";
             Debug.Log("RCV_ : " + dataString);
93
            if (!dataString.Equals("NOT OPEN") )
```

```
95          {
                // recived string is  like  "accx;accy;accz;gyrox;gyroy;gyroz"
97              char splitChar = ';';
                string[] dataRaw = dataString.Split(splitChar);
99
                // accelerometer values are normalized
101             float axx = Int32.Parse(dataRaw[0]) ;
                float ayy = Int32.Parse(dataRaw[1]) ;
103             float azz = Int32.Parse(dataRaw[2]) ;
                normalize = Mathf.Sqrt(Mathf.Pow(axx, 2) + Mathf.Pow(ayy, 2) +
    Mathf.Pow(azz, 2));
105             float ax = Int32.Parse(dataRaw[0])/ normalize;
                float ay = Int32.Parse(dataRaw[1])/ normalize;
107             float az = Int32.Parse(dataRaw[2])/ normalize;

109             // high pass filter

111             if (Mathf.Abs(ax) - 1 < 0) ax = 0;
                if (Mathf.Abs(ay) - 1 < 0) ay = 0;
113             if (Mathf.Abs(az) - 1 < 0) az = 0;


115
                curr_offset_x += ax * moveSpeed * Time.deltaTime;
117             curr_offset_y += ay * moveSpeed * Time.deltaTime;
                curr_offset_z += az * moveSpeed * Time.deltaTime;
119

121             // Limit the position of the object

123             float clampedX = Mathf.Clamp(curr_offset_x, -3.0f, 10.0f);

125             curr_offset_x = clampedX;


127
                float clampedZ = Mathf.Clamp(curr_offset_z, 1.0f, 10.0f);  //
    never change 1
129
                curr_offset_z = clampedZ;
131

133               //  float clampedZz = Mathf.Clamp(curr_offset_z, 4.0f, 10.0f)
    ;  //never change 1

135 //curr_offset_z = clampedZz;


137
                if (enableTranslation) fingertip.transform.position = new
```

```
    Vector3(curr_offset_x, curr_offset_z, curr_offset_y);
139
         rbody.AddForce(curr_offset_x, curr_offset_z, curr_offset_y);
141
        }
143
    }
145


147


149 private void OnCollisionEnter(Collision collision)
    {
151         if (collision.collider.name == "Object1")
       {
153         rend.material.color = Color.red;
            int azz = Convert.ToInt16(curr_offset_z);
155         azz = 7 * azz + 55;
            string ayy = Convert.ToString(azz);
157         data.WriteLine("V" + ayy);
        }
159

161      if (collision.collider.name == "Object2") //Cube2
        {
163         impactvariable = collision.impulse;
            //stream.Write("V");
165         int azz = Convert.ToInt16(curr_offset_z);
            azz = 7 * azz + 55;
167         string ayy = Convert.ToString(azz);
            data.WriteLine("v" + ayy );
169

        }
171
        if (collision.collider.name == "Object3") //Cube3
        {
173         int azz = Convert.ToInt16(curr_offset_z);
            //azz = 4*azz + 50;
175         azz = 0 * azz + 70;
            string ayy = Convert.ToString(70);
177         data.WriteLine("c" + ayy);
179

        }
181 }


183
    private void OnCollisionStay(Collision collision)
```

```
185     {
            if (collision.collider.name == "Object1")
187         {
                int azz = Convert.ToInt16(curr_offset_z);
                azz = 7 * azz + 55;
                string ayy = Convert.ToString(azz);
191             data.WriteLine("V" + ayy);
            }
193
            if (collision.collider.name == "Object2") //Cube2
195         {
                rend.material.color = Color.blue;
                //stream.Write("V");
                int azz = Convert.ToInt16(curr_offset_z);
199             azz = 7 * azz + 55;
                string ayy = Convert.ToString(azz);
201             //string ayy = Convert.ToString(fingertip.transform.position.y)
    ;
               Debug.Log(ayy);
203             data.WriteLine("v" + ayy );


205
            }
207
            if (collision.collider.name == "Object3") //Cube3
209         {
                int azz = Convert.ToInt16(curr_offset_z);
211             //azz = 4*azz + 50;
                azz = 0 * azz + 70;
213             string ayy = Convert.ToString(70);
                data.WriteLine("c" + ayy);
215
            }
217
    }
219


221 private void OnCollisionExit(Collision collision)
    {
223     if (collision.collider.name == "Object1")
        {
225         int azz = Convert.ToInt16(curr_offset_z);
            azz = 7 * azz + 55;
227         string ayy = Convert.ToString(azz);
            data.WriteLine("V" + ayy + "B");
229
        }
```

```
231
233         if (collision.collider.name == "Object2")
            {
235             rend.material.color = Color.green;
                int azz = Convert.ToInt16(curr_offset_z);
237             azz = 7 * azz + 55;
                string ayy = Convert.ToString(azz);
239
                data.WriteLine("v" + ayy + "b");
241
            }
243
            if (collision.collider.name == "Object3") //Cube3
            {
245             int azz = Convert.ToInt16(curr_offset_z);
                azz = 0 * azz + 100;
247             string ayy = Convert.ToString(100);
249
                data.WriteLine( "k" );
251
            }
253     }

255 }
```