

FingerSight: A Vibrotactile Wearable Ring to Help the Blind Locate and Reach Objects in Peripersonal Space

by

Shantanu Avinash Satpute

Bachelor of Engineering, Savitribai Phule Pune University, 2018

Submitted to the Graduate Faculty of
Swanson School of Engineering in partial fulfillment
of the requirements for the degree of
Master of Science in Bioengineering

University of Pittsburgh

2019

UNIVERSITY OF PITTSBURGH
SWANSON SCHOOL OF ENGINEERING

This thesis was presented

by

Shantanu Avinash Satpute

It was defended on

March 29, 2019

and approved by

Roberta L. Klatzky, Ph.D., Professor, Department of Psychology, Carnegie Mellon

Zhi-Hong Mao, PhD., Professor, Department of Electrical and Computer Engineering and
Department of Bioengineering

Thesis Advisor: George Stetten, M.D., Ph.D., Professor, Department of Bioengineering

Copyright © by Shantanu Avinash Satpute

2019

FingerSight: A Vibrotactile Wearable Ring to Help the Blind Locate and Reach Objects in Peripersonal Space

Shantanu Avinash Satpute, M.S.

University of Pittsburgh, 2019

Visually impaired people need a solution to compensate for the lack of visual information. Although assistive technologies exist to help them navigate through the environment, blind people rely on groping to locate and reach objects in peripersonal space. FingerSight solves this problem using visual-to-tactile substitution. Our prototype consists of four haptic tactors embedded into a ring worn on the index finger, with a tiny camera mounted on top. The camera image is processed using computer vision to control haptic feedback to the user. The four tactors are evenly spaced around the finger. Users are instructed to move their hand towards the target by vibrating the tactor in that direction, guiding their motion until they reach the target. At that point, all tactors vibrate simultaneously.

Two experiments were conducted on normally-sighted participants, to test the functionality of our prototype. The first revealed that participants could discriminate between the five different haptic stimulations with a mean accuracy of 89.4%, which improved with additional training. In the second experiment, participants were blindfolded and instructed to move their hand wearing the device to reach one of four Light Emitting Diodes (LEDs) mounted on a cardboard sheet within arm's reach. Infrared markers mounted on the device enabled its location to be recorded by an optical tracker. A computer vision algorithm located the LED in the camera image and controlled the tactors using two different strategies: (1) Worst Axis First and (2) Adjacent Tactor Pair. Results

revealed that participants could follow the haptic instructions to reach the target with similar accuracy for both strategies, but that the time to reach the target was significantly different.

Using control systems analysis, a closed loop proportional-integral-derivative (PID) controller and plant was simulated. A model for the plant was computed on the experimental data using autoregressive-moving average with exogenous terms (ARMAX), with the human subject acting as the plant. The control system was then optimized to find the best strategy for factor activation, laying the groundwork for a future generation of FingerSight.

Table of Contents

Preface.....	xi
1.0 Introduction.....	1
1.1.1 Wearable haptic systems on the waist.....	2
1.1.2 Wearable haptic systems on the wrist	3
1.1.3 Wearable haptic systems on the tongue	4
1.1.4 Wearable haptic/audio systems on the head.....	5
1.1.5 Wearable haptic systems on the hand	6
2.0 Hardware Design	9
3.0 Computer Vision Algorithm	17
4.0 Motion Tracking	18
5.0 Experimental Methods	22
5.3.1 Experiment 1: Tactor Localization	24
5.3.2 Experiment 2: Guidance to Target.....	25
6.0 Results	31
7.0 Control System Analysis.....	39
8.0 Conclusion	48
Appendix A Program for FingerSight	50
A.1 Program for FingerSight with Worst Axis First guidance strategy	50
A.2 Program for FingerSight with Adjacent Tactor Pair guidance strategy	56
A.3 Program to control the Arduino Uno Board.....	62
Appendix B NDI Optotrack Certus Setup using Ethernet	66

Appendix C SWAT Analysis of Technologies Commercially available to Visually

Impaired People. 71

Bibliography 72

List of Tables

Table 1 Confusion matrix of error for first attempt	32
Table 2 Confusion matrix of erros for best attempt.....	32
Table 3 Statistical comparisons of experimental variables using Student's <i>t</i> -test.....	36
Table 4 Performance of PID controller.....	46

List of Figures

Figure 1 Flexible 3D printed ring	11
Figure 2 Second generation of Ring	12
Figure 3 Velcro ring and camera holder	13
Figure 4 FingerSight Circuit Schematic	14
Figure 5 Array of ten markers embedded in a holder	19
Figure 6 Marker Array attached to camera holder	20
Figure 7 External Trigger and Synchronization Circuit	21
Figure 8 LED Placement.....	23
Figure 9 Velcro belt around haptic factors.....	24
Figure 10 Numeric Keypad to report the vibrating factor(s)	25
Figure 11 Guidance to Target	26
Figure 12 End of trial.....	27
Figure 13 Axes for guidance.....	28
Figure 14 Comparision Between Mean Error Proportions: Experiment 1	33
Figure 15 Histogram of R^2 for linear approach to the initial stopping point.	35
Figure 16 Means of the five parameters for each strategy. The error bar is 1 s.e.m.....	35
Figure 17 Illustration of regression of Euclidean distance	37
Figure 18 Motion in x and y directions for each guidance strategy.....	38
Figure 19 Classical control system block diagram for FingerSight.....	39
Figure 20 Illustration of input and output signals of the Plant	40
Figure 21 ARMAX model with 65.02% fit	43

Figure 22 Simulation of control system in Simulink	44
Figure 23 Step response of the PID controller	45
Figure 24 Output of the plant regulated by the tuned PID controller	46
Figure 25 SWAT Analysis.....	71

Preface

This thesis has been written to fulfill the graduation requirement of the Master of Science in Bioengineering. FingerSight is a patented technology by the University of Pittsburgh. I am glad that I could make some contributions to the development of the technology.

This thesis would not have been possible without the excellent guidance and persistent motivation from my research advisor, Dr. George Stetten. I am grateful to him for granting me an opportunity to conduct research in the Visualization and Image Analysis (VIA) Laboratory, and for the continued help and wise counsel in making this thesis a reality. Development of the prototype for FingerSight would have been impossible without his vision and assistance. I wish to thank Dr. Roberta Klatzky for introducing me to the whole new world of Psychophysics and for all the support and feedback during the past two years. I wish to express sincere gratitude towards Dr. Zhi-Hong Mao, for his exceptional guidance and co-operation in the control system analysis of FingerSight.

I would especially like to thank my parents and my sister, who deserve a special note of thanks. Thanks for motivating me to dream big, and for the endless encouragement and support. I thank my friends in Pittsburgh for accompanying me in the ups and downs. Lastly, I would like to thank my lab-mates for maintaining a positive and friendly environment, leading to more productive research space. Special thanks to Janet Canady, for helping me with research and the experimental setup.

I hope FingerSight will eventually be a successful commercial product and make a difference in the lives of people who are deprived of sight. It was a pleasure contributing towards the development of a novel device, meant for a noble cause.

1.0 Introduction

It is estimated that 1 million Americans were legally blind in the year 2015, while 3.2 Million Americans had visual impairment [1]. According to a recent study by the National Eye Institute, it is estimated that these numbers will skyrocket to 8 Million in 2050 [2]. Such a large prevalence of visually impaired people in the society, provides us researchers an opportunity to bring a change in their lives. It is common to see a blind person struggle with finding things due to the lack of vision. Blind people have to grope around to reach objects. The motivation for FingerSight, was to quickly and reliably guide blind people to locate and reach objects in peripersonal space.

1.1 Related Work

Visually impaired people have relied on white canes and watch dogs as the primary assistive tool for more than eighty years. Even though a lot of technologies have been researched and implemented, it is a common sight that blind people still rely on them for assistance.

Various electronic technologies have been researched and implemented with the goal of improving the lives of visually impaired people. Some, such as magnification systems and optical character recognitions systems, have focused on helping the subject to read the printed page. Others are focused on helping blind people navigate in unknown environment using haptic feedback, which is our approach. The sections below include a review of related work in this area.

1.1.1 Wearable haptic systems on the waist

Around 1998 a wearable tactile navigation system was developed at MIT's Media Lab. They devised a haptic directional display with a matrix of 4x4 vibrators embedded in the back of a vest [3]. The matrix of vibrators could generate five instructions to guide blind people: four cardinal directions and a stop signal. The guidance system consisted of infrared transceivers and a wearable computer, apart from the wearable haptic array. IR was used to determine the location of the user which was transmitted to the computer. The computer, already knowing the location of the target, then decided the shortest path to that target and activated corresponding vibrators to instruct the participant to move accordingly.

Later, a belt worn around the waist called 'ActiveBelt' was developed at Keio University, Japan. [4]. The belt had 8 vibrators controlled by a microcomputer. The 8 vibrators were designed to communicate the four cardinal directions and the four primary intercardinal directions. In an experiment, they used four different pulse intervals of vibrations, all with a 50% duty cycle. For walking, it was found that a pulse interval of 1000 ms (vibrator on for 500 ms and off for 500 ms) worked the best for most participants. Further, based on the feedback from the participants and the outcomes the experiment, it was concluded that, for practical applications, four vibrators were sufficient instead of 8, and that it is desirable for the vibrators to be activated only when the users were lost.

The Tactile Wayfinder was another haptic belt, designed and tested in Germany [5]. The Tactile Wayfinder had 6 vibrators embedded in a belt. However, unlike ActiveBelt, where only one vibrator was turned on at a time, two vibrators could be turned on simultaneously, giving participants a sense of an in-between direction. In an experiment with 13 sighted participants, it was found that people could perceive the indicated direction with 15 degrees mean deviation.

Furthermore, a navigation experiment was conducted on seven participants on two different routes with 5 and 6 waypoints. It was reported that in 99% of the recorded positions of the users via GPS, the deviation from the desired path was less than 15 meters. The user only received directional information of the next waypoint. We thought that such a strategy of giving users an in-between direction could be used to guide participants using FingerSight. We called it the “Two Tactor Vector Method,” as discuss it in Chapter 5.3.2.

1.1.2 Wearable haptic systems on the wrist

In 2007, a basic single wrist-worn vibrator was tested on 8 legally blind patients at the University of California, Santa Barbra [6]. It was tested in two modes, while walking on an 80 m path with 45 and 90 degree turns. In the on-course mode, the vibrator was activated when the user was oriented in the desired direction (towards the next waypoint). In the off-course mode, the vibrator was activated when the participant was oriented in the wrong direction. Even admitting the limitations of using a single vibrator, the authors demonstrated that a single binary signal (on/off) could guide legally blind participants. Based on such a binary feedback, we have designed a warmer-cooler strategy for hand guidance using FingerSight, which is also discussed in Chapter 5.3.2. The authors also reported that haptic feedback was just as efficient as audio feedback in guiding legally blind participants.

Another strap with 8 vibrators that could be worn around the wrist was developed at the University of California, Santa Cruz, in 2015, and tested on 10 bind participants [7]. The authors stressed that haptic instructions did not negatively impact user’s ability to listen to sounds from the environment, since blind people depend on such auditory feedback from the environment to accomplish various tasks in life thereby proving haptic instructions to be superior than auditory

instructions not only in terms of practicality of use, but also in terms of ease of following instructions. The authors reported that with haptic instructions participants could more closely follow the desired path while walking. However, accuracy in following a desired path can be achieved at the expense of higher time to reach the destination. In interviews with participants, it was noted that they preferred haptic feedback to audio since it does not interrupt their hearing. In addition to directional control while walking by activating a single vibrator to direct the user, they also implemented a rotational control when the user was misoriented by more than 67.5 degrees. In order to indicate the user to rotate in the direction of the waypoint, vibrators were sequentially activated one after another in clockwise or anticlockwise direction

The Kahru tactile indoor navigator was developed at Carnegie Mellon University in 2001, in the form of a harness consisting of 6 vibrating motors [8]. The vibrating motors generated six different factors that could instruct the user to move in the four cardinal directions as well as to speed up or slow down. Communication with the vest/harness was established using an infrared receiver, which activated appropriate vibrators.

1.1.3 Wearable haptic systems on the tongue

A Tongue Display Unit was introduced by the University of Wisconsin. The authors had proposed that the way the brain processes visual information could be retained by stimulating the tongue with an electrode array. Signals could be sent to the brain via a different pathway that deals with touch. It was hypothesized that blind people could learn to perceive touch as sight in the visual cortex. An array of 12x12 flexible electrodes was developed that stimulated the nerves on the tongue in order to convey visual information picked up by a camera to the participant. It was reported that after about 15 hours of training, the user loses the on-the-tongue sensations and

perceives stimulations as shapes and features in space [9]. A few other researchers have continued to work on tongue display to help blind people navigate around obstacles and to detect correct stability and posture [10] [11]. However, it is evident upon market records that tongue displays did not find a viable expression in terms of practicality of use and commercialization.

1.1.4 Wearable haptic/audio systems on the head

Head mounted assistive devices are the most widely used kind of wearable devices. The vOICe system is a technology that was developed and patented by Philips Co. It converts visual information from images into sound patterns. Loudness and frequency of sound represent corresponding visual information such brightness, position and elevation [12].

Intelligent glasses (IG) are an application of vision and touch stimulating interface developed as a novel electronic travel aid (ETA). The intelligent glasses provide tactile maps of spaces, in turn helping blind people navigate in the environment. The visuo-tactile ETA consists of stereo cameras placed on the eyes (glasses) that acquire binocular visual information. This visual information is processed by a scene analyzer that implements computer vision algorithms whose output is fed to a braille-like tactile display based on shape memory alloy. An experiment was conducted on 20 healthy sighted participants to evaluate performance on (1) Shape and direction recognition, (2) Environment structure understanding, and (3) Navigation using tactile maps. The mean correct discrimination was different for different shapes and the worst for circle. Exploration times ranged from about 5 seconds to 55 seconds depending on the shape. All subjects could successfully recognize all the directions. A task where participants had to touch the tactile display and then match it with one of the visual maps handed to them demonstrated that 90% of

participants could acquire a good understanding of the environment by touch. Further, 5 participants were taken to the entrances of 4 different rooms and asked to explore the tactile display of the room and then report what they felt as a drawing. Out of 5, three subjects could successfully recognize the navigation environment [13]. It is evident that placing stereo cameras on glasses is a promising approach for head mounted assistive devices.

“Seeing AI” is an ongoing research project at Microsoft in collaboration with Pivothead that proposes to utilize the power of AI to describe text, colors, currency and object to the low vision community using audio feedback. While wearing these unique glasses, the user can just swipe the touch panel on the eyewear to take a photo. It then can analyze the image and can translate image to speech to describe attributes of the person or object or just read text [14] [15]. The company wishes to commercialize the device and help the blind community. Seeing AI can also be used just as a smartphone application, using the phone camera to take pictures. It is reported that the application can scan barcodes, identify people and objects and can even describe facial expressions of people and can describe scenes.

1.1.5 Wearable haptic systems on the hand

Our approach is to mount the tactors on the hand, in part because of the high density of mechanoreceptors. Out of the four mechanoreceptors responsible for sensory information conveyed using touch, Pacini and Meissner receptors are involved in sensing vibrations. Meissner receptors respond to touch or high amplitude, low frequency stimuli, whereas Pacini receptors respond to vibration or low amplitude, high frequency stimuli [16]. We believe the vibrators used in FingerSight stimulate the Pacini receptors. The two-point discrimination threshold (TPDT) is the least for fingers at 2.5 mm, meaning that two different points on the finger must be only 2.5

mm apart to perceive them as two different points. The TPDT of skin is different for different body parts. It is extremely high for calf, thigh, belly, back, chest and arms, but substantially lower for fingers, thumbs and on different parts of the face [17]. Given that the fingers have the best (highest) haptic resolution, they are an appropriate target for haptic devices. In 2008 an innovative tactile device that could be wrapped around the fingertip was developed [18]. The device was constructed using electroactive polymer and soft actuators, which can stimulate the skin without any electromagnetic transmission. Their softness and flexibility make them adaptable to any contours of the human body, especially the fingertip where the haptic resolution is the best. The device can simulate braille code or other codes that can provide haptic instructions to blind people. Even though the device sounds very flexible, its high voltage requirement and low bandwidth present a challenge to their practicality for use in everyday lives.

Two prototypes for a vibrotactile display to be worn around the wrist and the forearm were developed at the University of British Columbia, Canada. Both the displays consisted of 2 vibrating DC motors similar to the ones used in FingerSight, spaced 60mm apart. The vibrators were worn by clinicians and activated intermittently at 140 Hz to alert the clinicians about undesirable heart rate changes of a patient during treatment. It was proposed that the vibrations could be used to send such basic alerts without distracting the clinicians, as a good way to replace the auditory alarms that are generally used. Experiments were conducted using the prototypes and it was concluded that both the prototypes worked equally well [19]. Such an interface can be modified to help in navigating blind people.

Several wearable and smart technologies have been researched and developed. However, there have been mixed reviews about the viability of such wearable acoustic/tactile assistive devices. Cognitive overload can be one of the major problems when arbitrary mappings from the

senses to spatial information are used. Furthermore, auditory feedback potentially interferes with the user's ability to pick up auditory cues from the environment. Additionally, tactile interfaces work precisely only on the fingers and tongue. Thus, it is the most practical to stimulate receptors on the fingers [20].

1.2 Related work from our Lab

PalmSight, an assistive technology helping the blind to locate and grasp objects, was developed at the Carnegie Mellon University. PalmSight consisted of binocular cameras mounted on the user's hand and five vibrators attached to the back of the same hand. It used computer vision methods to generate depth map and then identify and locate the target object in 3D space, and activating corresponding vibration to help grab/grasp an object [21].

2.0 Hardware Design

The prototype developed for FingerSight in this thesis consists of a ring worn around the index finger embedded with four small tactile buzzers/vibrators (Adafruit part number 1201) commonly found in cell phones and mounted on which is a small camera. These vibrators, which in contact with the skin are known in the haptics community as *tactors*, are interfaced with an Arduino Uno board that controls the pattern of vibration. The Arduino Uno board and the camera are each connected by universal serial bus (USB) to a computer that runs an algorithm to decide which vibrator(s) is/are turned on depending on the visual information acquired from the camera. The camera provides a video frame with a resolution of 480 x 640 to the computer every 0.5 seconds. The frame is processed using a computer vision algorithm that is described in Chapter 3. The computer vision algorithm detects a target object and based on the location of that object in the frame, activates one or more of the tactors, thereby delivering haptic instructions to guide the user to move the hand in left, right, up, down or straight forward directions relative to finger. The tactors are activated depending on various guidance strategies that we have developed, which are explained in the chapter 5.3.2.

We have devised experiments to evaluate the functionality of our prototype and the user's ability to follow haptic instructions using an optical tracking system (NDI Optotrack Certus), which is capable of tracking the 3D location of infrared (IR) light emitting diodes (LEDs). We mounted an array of 10 IR-LEDs on the ring, thus permitting continuous tracking of its location and orientation, as described in Chapter 4.

2.1 Iterations of the FingerSight Ring Prototype

We have gone through multiple iterations of the design of the FingerSight ring prototype. Within the basic design of the ring with four tactors and a camera, we needed to ensure that the vibration of a given tactor could be localized. Considering the small surface area around the base of a finger, one of the biggest challenges of designing the ring was taking into account that the ring could transmit the vibration from a given tactor to other parts of the finger, making it difficult for the user/participant to discriminate and localize these vibrations.

The first version of the ring was 3D-printed using a flexible material (Ninjaflex Thermoplastic Elastomer). It did not accommodate a camera, since the initial objective was to test if the tactors can be embedded into the ring in such a way that it could fit around the index finger. The challenging part was accommodating the relatively big size of the tactors (10 mm diameter), as compared to the small diameter of the index finger and space between it and the middle finger, leaving an extremely small permissible gap between each of the vibrators.

In our first prototype, a ring was printed (refer to Figure 1) and a circuit was built to interface the haptic tactor with an Arduino UNO board. At this point, we realized that the variations between the diameters of index fingers of different users is a particular challenge, especially since the tactors have to be in the four cardinal directions around the index finger, for users to correctly perceive the haptic instruction. Thus, a ‘one size fits all’ ring made of a non-stretchable material did not work, because, depending on the diameter of the particular index finger, the four tactors would not end up spaced by 90 degrees around that finger. Also due to the extremely small gap between each tactor and the large width of the printed material, vibrations were transmitted through the ring to the entire finger, making the user feel as if all the vibrators were activated. Moreover, the limited accuracy of the 3D printer (LULZBOT Taz6) in manufacturing such a small device

holding four factors was difficult. Thus, factor discrimination and localization were challenging using this version of the ring.

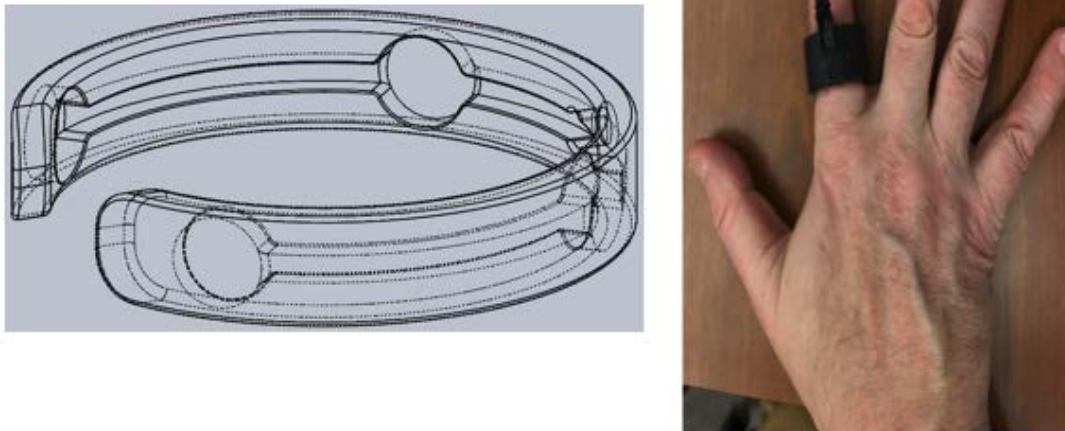


Figure 1 Flexible 3D printed ring

In the second prototype we used a flexible rubber hair-band made from Silicone (Part number: 38205_A) to act as a ring as shown in Figure 2. Each factor was individually attached the hair band ring using heat shrink tubing. Since the only link between the factors was the flexible rubber band, localization of vibrations was better. Separate from the vibrator band, a second ring was designed, and 3D printed to act as a camera holder. After comparing a number of camera models, we settled on a small cylindrical USB camera commercially available for inspecting plumbing systems, because of its small size (lens diameter= 7mm), visual angle of 70 degrees and low cost (\$13). The two rings worn together served as a prototype which we tested in the laboratory. Although this iteration of prototype was uncomfortable to wear and bulky, since it consisted of

two different rings, it demonstrated that the vibrations of the right intensities could be localized and related to images processed from the camera.



Figure 2 Second generation of Ring

We next considered using a more comfortable and flexible material for the ring. We chose a flexible white elastic material commercially available for use in clothing (width: 19mm, Model number: Dritz 9322W) sewed into a ring, with the four tactors glued to its inner surface as shown in Figure 3, so that the vibrators were directly in contact with the skin of the index finger and remained equally spaced over a range of finger sizes as the elastic stretched evenly between each tactor. This, in turn, ensured better vibration localization, making it easy for the user to discriminate vibration between one tactor and another. A small loop of elastic on the top (north) of the ring acted as the camera holder. The prototype was tested, and an optimum intensity of vibration was used as discussed in Section 2.3. The elastic ring suffered from an unfortunate

tendency to cut off the blood supply to the finger, especially in subjects with larger fingers, so we abandoned this prototype

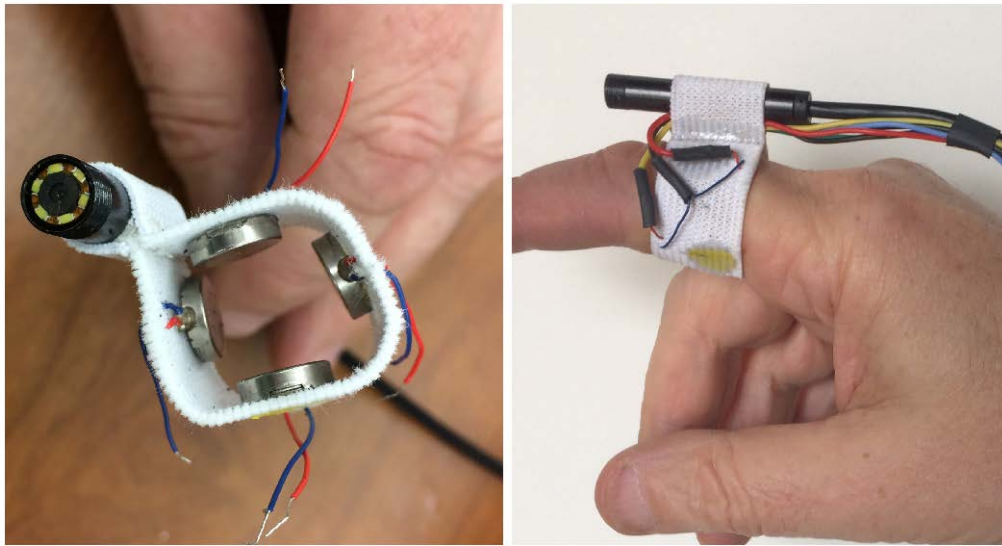


Figure 3 Velcro ring and camera holder

Our final prototype consisted of a Velcro strap (Product Number: 90340) and a custom camera holder. (See Figure 6 in chapter 4) Under the strap, the four factors were individually attached to the finger using a two-sided Scotch tape with a width of 0.5 inch. A separate camera holder was 3D printed, such that the Velcro passed through a notch in its base. The camera housed in the holder had a resolution 640x 480, with a visual angle of 70 degree. At the top of the camera holder, we attached a curved housing for an array of ten infrared diodes to track the movements of the user during the experiments. For more details on 3D tracking refer to chapter 4.0.

2.2 Hardware Design

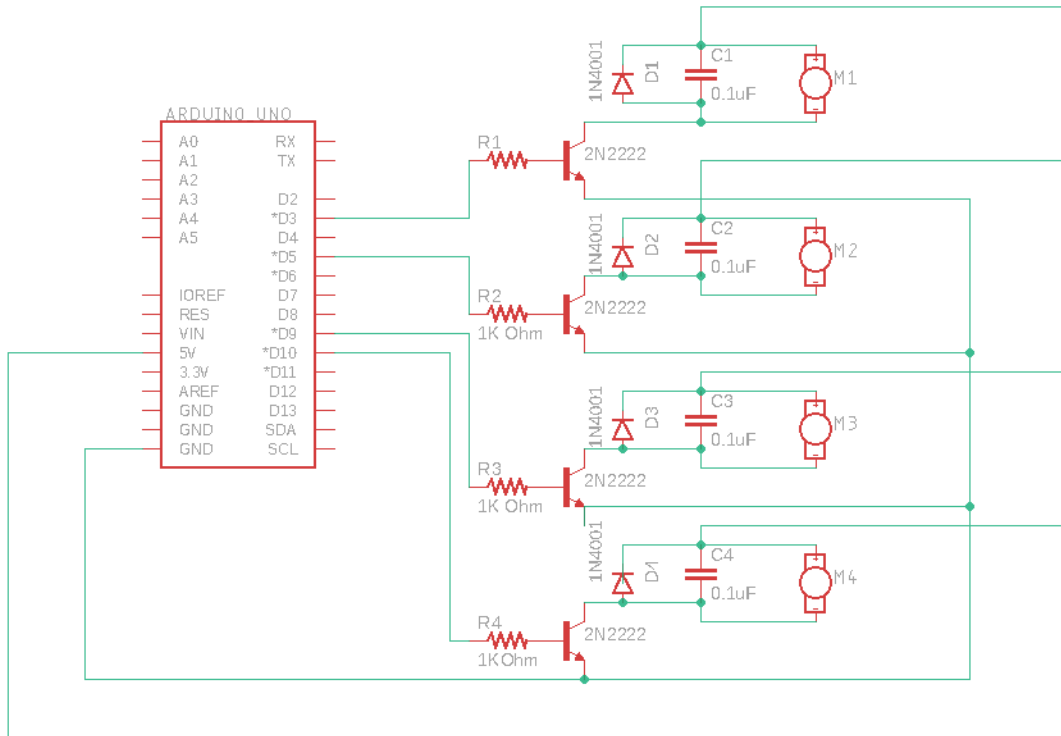


Figure 4 FingerSight Circuit Schematic

The four factors M1, M2, M3 and M4 are driven by an Arduino Uno. Voltage spikes are produced by the motor and the diodes connected in reverse bias act as surge protectors, which protects the microcontroller. The $0.1\mu\text{F}$ ceramic capacitors connected in parallel also absorb these voltage spikes. To drive the factors, 75 mA of current is needed. Since the Arduino can source current only up to 40 mA, a transistor is used to amplify current.

2.3 Design Parameters

In the design of FingerSight, there are multiple cognitive and ergonomic factors to consider to ensure the functioning of the device. During our initial testing trials, we explored parameters such as which finger is best suited for the ring, as well as intensity of vibrations at which haptic instructions are delivered to ensure correct perception of these vibrations.

The two-point discrimination threshold of all the five fingers is almost the same, approximately 2 mm [17][22]. People are usually accustomed to pointing at things using their index finger. Also, as compared to the other fingers, the index finger has the advantage of freedom of large movements. Even though the pinky finger has the same advantage, its use raises the problem of the other fingers occluding the camera. Also, the index finger has larger circumference than the pinky, which ensures a bigger gap between the tactors. Further, mounting the camera on the top of the index finger guarantees that other fingers do not occlude the camera or the target.

The tactors were mounted in the north (dorsal), east (medial), west (lateral) and south (ventral) directions on the index finger. In all cases, we used the dominant hand. During initial testing in the laboratory, it was evident that when the tactors were not aligned in the cardinal directions, it was more difficult to discriminate between vibrations from different tactors. For example, in order to perceive the north (top) vibration and to distinguish it from other vibrations, it was important that the tactor was clearly located in the north direction. We believe this is due to a greater ease in using easily defined, and therefore more easily differentiable, directions.

As discussed earlier, one of the problems faced was transmission of vibration to the entire base of the finger by the ring, and also the skin to an extent. To overcome this problem, we decided to explore different intensities of vibration. The intensity of vibration felt by the user depends on DC voltage applied to the tactors, as they are essentially vibrating motors. When 5 Volt DC voltage

was applied to the factors, the intensity of vibration was high but it also lead to higher transmission of vibration. To reduce transmission and better localize the vibration, a voltage of 4V was applied, which made the motor slower by 20% which reduced the transmission of vibration ultimately making it easy to discriminate one factors vibration from another. It also benefited by reducing the distracting vibration noise substantially. However, when all four haptic factors were activated simultaneously (which we did to instruct the user to move straight or in order to indicate that they are very close to the target object), we applied only 3.33 V across the factor in order to reduce the noise and facilitate better haptic instruction discrimination.

The next design parameter that needed to be optimized was the number of haptic instructions delivered to the user every second. Initially we felt that the higher the number of instructions per second, the faster the response would be. However, during testing on sighted/healthy people, we realized that operating at faster than 2 instructions per second lead to redundant signals. For example, even if 5 haptic instructions were delivered a second, people would only perceive and act on two of them. Since, the subject pool for our experiments would consist of sighted/healthy people, we were constrained to no more than this number of instructions per second. The participants were also not habitual to the haptic instructions on the finger. With persistent use, one might be able to respond to more than 2 haptic instructions per second.

3.0 Computer Vision Algorithm

As stated earlier, the camera captures a frame every half a second. To detect the target object, a computer vision algorithm was implemented using OpenCV. Connected pixels in a frame that have a similar property are called a *blob*. In our case, connected pixels with a hue-saturation-value (HSV) in a defined range formed blobs. Since we used LEDs mounted on a large cardboard sheet as the target objects in our experiments, an algorithm was implemented to detect circular blobs.

First of all, we define an HSV range to determine whether a given pixel is a member of the blob or not. Upper and lower limits of HSV differ from color to color. Further, the frame acquired from the camera is converted from red-green-blue (RGB) to HSV color space. A binary mask is then created using the defined range of HSV values and a series of erosions and dilations is used to remove small areas due to noise. The result is an effective segmentation of the pixels within the LED from the black background.

Further, the contour of the blob is computed. A circle with minimum radius that encloses the blob is computed and displayed on the frame. The (x,y) coordinates and the radius of the circle are extracted to compute the correct haptic instruction to be delivered to the user. The (x,y) image coordinates based indicate the direction in which to move in the x and y directions towards the detected blob. Since the target appears bigger as one moves towards it, the radius of the circle indicated how far the user is from the target. If there is no blob detected in a given frame, no haptic instruction is delivered, indicating to the user that the target object is not in the field of view of the camera.

4.0 Motion Tracking

To verify the functionality of FingerSight, and to evaluate performance of different guidance strategies, tracking the location of the finger of participants while performing the experiment was necessary. NDI (Northern Digital Inc.) Optotrack Certus, is a high precision, high accuracy and high-speed optical tracker that was used for the purpose of tracking [23].

In order to collect 3D position of the finger of participants, software developed by Northern Digital named *First Principles* was used. Detailed instructions on setting up the Optotrack Certus is included in the Appendix. The Optical tracking system consists of a position sensor, a system control unit (SCU), a strober, and a set of markers. The position sensor is comprised of three cameras that determine the location of markers. Markers used in the experiment were special infrared (IR) LEDs, each containing four independent light-emitting elements in a small square to improve accuracy. The markers are connected to the system control unit via a strober, which is responsible for activating and deactivating infrared light emitting diodes. The three cameras built into the optical tracker sense the location of these IR-LEDs, and report the 3D positions to the computer, which are then displayed and stored using First Principles software.

To correctly record hand motion of participants in response to haptic instructions that are delivered, we designed a rigid body containing an array of ten markers to be attached to the ring. A minimum of three of the markers have to be properly oriented in the field of view of the three cameras of the position sensor to correctly determine the pose (orientation and location) of the rigid body. To ensure this minimum over a range of hand orientations, a 3D structure shaped as a curved cap was printed in which were embedded the ten markers as shown in Figure 5.

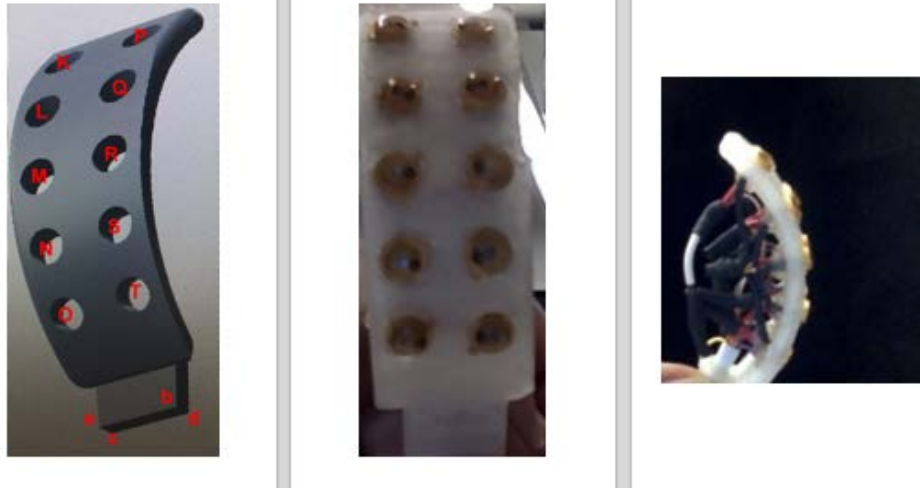


Figure 5 Array of ten markers embedded in a holder

The advantage of having ten markers was that even if participants rotated their finger or wrist during experiments, the position sensor would still be able to track at least three of the markers. The First Principles software records the position of each marker that is visible, and from these computes and records the position of origin of the rigid body and orientations of the x,y and z axes of the rigid body., which are defined by the user relative to the markers. Marker R in Figure 5 was set as the origin of our rigid body and markers M and Q used in addition to define the x, y and z axes. Further, a rectangular surface defined by points *abcd* was attached to the camera holder, which in turn attached to the ring (as shown in figure 6).

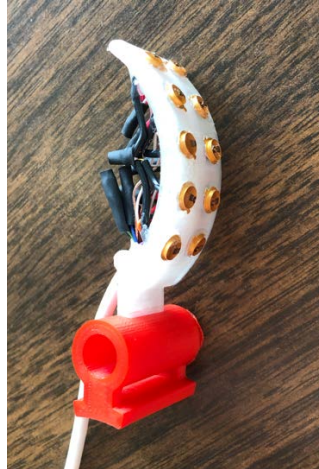


Figure 6 Marker Array attached to camera holder

The data collected from the optical tracker consisted of the coordinates of each visible marker and the coordinates of the origin of the rigid body in global coordinates (optical tracker coordinates) and the orientations of the x, y and z axes of the array. Using the data collection software, the frequency at which these measurements are recorded can be adjusted. For our experiment we used a tracking frequency of 20 Hz. Participants were asked to move their hand in particular tactor vibrations, with these haptic instructions being delivered at a frequency of 2 per second. Thus, a tracking frequency of 20 Hz was sufficient to supply 10 locations for every haptic instruction.

A challenge we faced was synchronizing the tracker with the FingerSight system, in order to correctly analyze data. We needed to time-stamp the haptic instructions to know when a certain haptic instruction was delivered relative to the movement of the participant based on the tracking data. To accomplish this, we used the external triggering function of the tracker. An RS232 connection was established between the system control unit (SCU) of the tracker and an Arduino Uno that controlled the Target LEDs for the experiment (see Fig. 7). This is the same Arduino

used to control the factors (see Figure 4 in Chapter 2) Using the RS232, the SCU was triggered by the Arduino to start data collection. To synchronize the tracker with the FingerSight experiment, we used the internal clock of the tracker. The tracker was operated at 20 Hz and the SCU clock line was connected to a decade counter, whose output thus went high twice a second and was used to trigger the haptic instruction by the Arduino. In this manner, complete synchronization was achieved between the tracking and the haptic instructions.

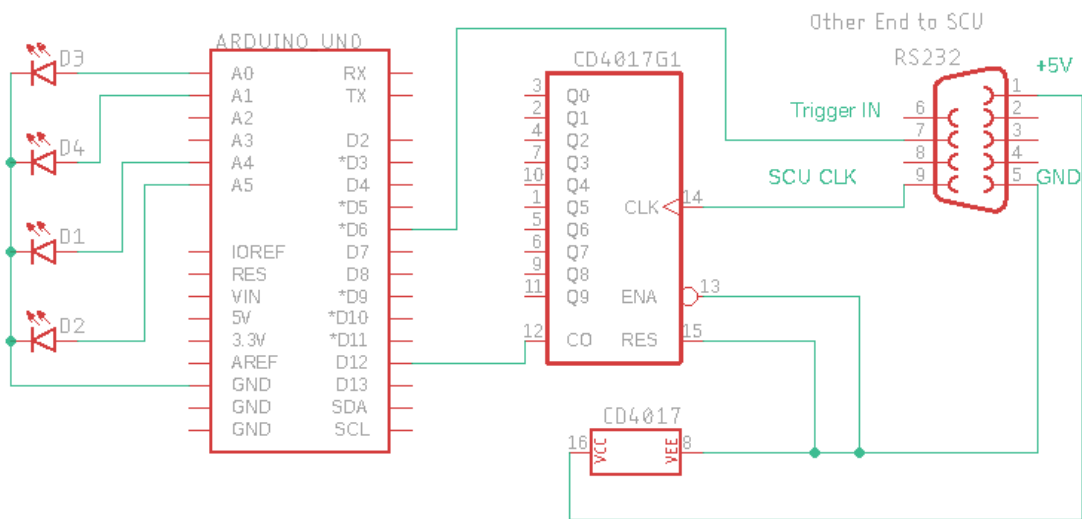


Figure 7 External Trigger and Synchronization Circuit

5.0 Experimental Methods

This chapter discusses the experimental methods in detail. It describes the subject population, experimental setup, design, procedure, and measures.

5.1 Subject population

With IRB approval at Carnegie Mellon University, experiments were conducted with volunteer undergraduates recruited from the Psychology Department subject pool. The participant pool consisted of 19 males and 11 females. All 30 participants were sighted with no known visual impairments and had the ability to feel vibration in their hand and control hand movement normally. All participants signed a consent form before the experiment. The experiments lasted for less than an hour.

5.2 Stimuli

Four white Light Emitting Diodes (LEDs) inserted into a black cardboard poster acted as the target objects (refer to schematic shown in Figure 7 in Chapter 4). The four LEDs were placed in arbitrary locations to form a quadrilateral with no sides parallel or equal, as shown in Figure 8, to ensure that participant could not easily predict target location.

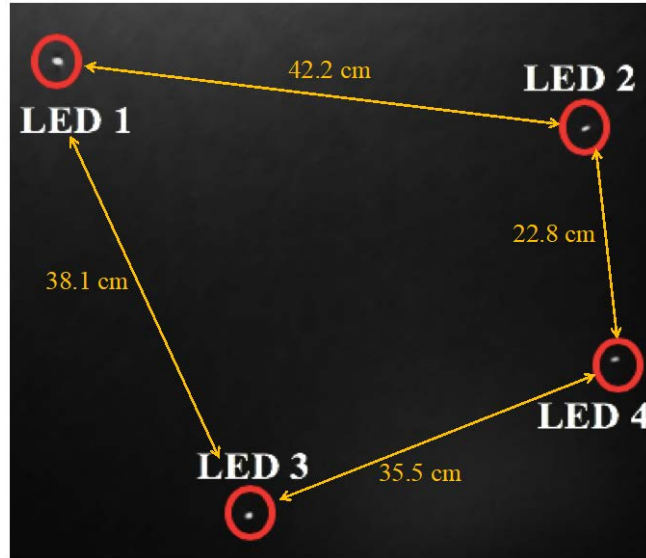


Figure 8 LED Placement

5.3 Design

Participants were seated at a table (91 x 60) facing the poster containing the LED targets. Four tactors (vibrators) were taped against the skin around the base of the index finger of their dominant hand, oriented in the 4 cardinal directions around the axis of the finger. A disposable Velcro belt, attached to the camera-holder and array of tracker markers, was then wrapped around the tactors (as shown in Figure 9). A detailed hardware description is reported in Chapter 2.1. The subsequent experiment was divided into two parts, which are explained in the following sections (5.3.1 and 5.3.2).

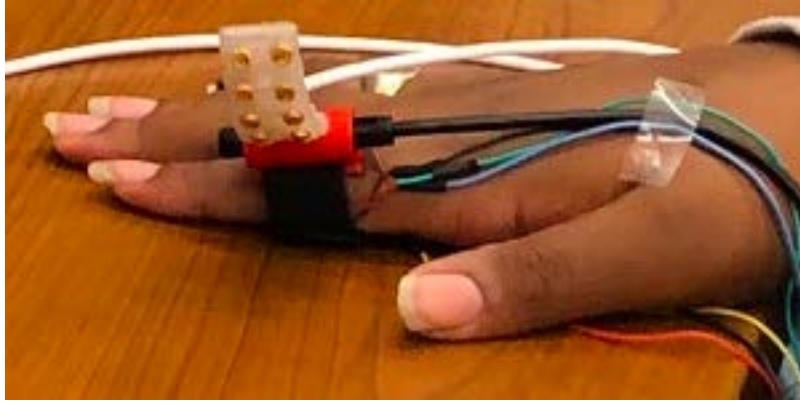


Figure 9 Velcro belt around haptic factors

5.3.1 Experiment 1: Tactor Localization

In the first experiment, we tested if the participants could tell which tactor(s) were activated. The participants held their dominant hand steady. Initially, the various tactor positions (left, top, right, bottom, all 4) were demonstrated to familiarize the participant. Then, over a series of 15 trials, each haptic instruction was tested three times, in random order. For each test, the tactor(s) vibrated for 90 milliseconds, after which participants reported what they felt by means of a numeric keypad. As shown in Figure 10, keys with arrows pointing in the four cardinal directions were used to report single pulses, and the central key (number 5) corresponded to all 4. The corresponding responses were recorded by the program along with the correct signals. If a participant made more than 3 errors, the 5 possible stimuli were again presented and named. The experiment was then repeated for these participants.



Figure 10 Numeric Keypad to report the vibrating factor(s)

5.3.2 Experiment 2: Guidance to Target

In this experiment, the participant attempted to move the hand wearing the FingerSight apparatus in the direction of the perceived vibration in order to reach the target. If they felt the left factor vibrating, they moved their hand in left direction, and so on, for each cardinal direction. When all four factors were activated, it indicated that their finger was aligned with the target object, in which case they were supposed to move straight forward, closer to the target object. Before the first target approach, each of the five haptic instructions was manually executed three times and labeled, to re-familiarize participants.

The participants were then blindfolded, and a curtain that covered the four target objects during the initial part of the study was removed. To begin each trial, participant sat with their palm

resting at the center of the table. All the target LEDs were approximately at a distance of 76.2 cm from the starting point. One of the four target LEDs was turned on. At this point, participants had to first search for the target by sweeping their hand in different directions so that the target object would become visible to the camera, similar to how a sighted person might search for a target. As soon as the target object was in the frame of the camera, participants got a vibration indicating the direction of the target object. The participants received two haptic instructions per second. When participants were properly aligned and within approximately 10 centimeters of the target, all the tactors were activated for 3 seconds, instructing the participant to move their finger forward to touch the target.

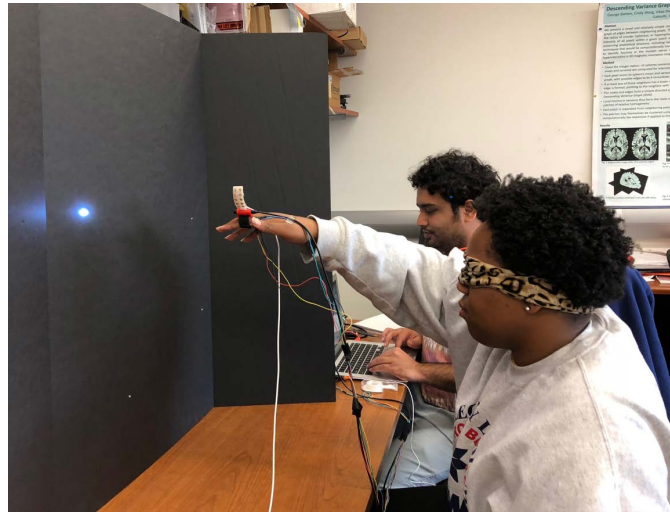


Figure 11 Guidance to Target

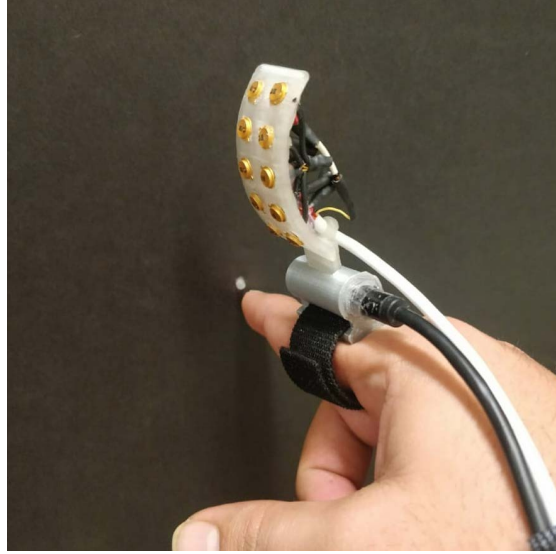


Figure 12 End of trial

In order to guide the participant's hand, we initially considered four different guidance strategies, determining the pattern of vibrations used to instruct the participant how to move to the target. The four guidance strategies (described below) are the following: (1) Worst Axis First, (2) Adjacent Pair, (3) Two Tactor Vector, and (4) Warmer-Cooler.

For ease of understanding, we refer to the left/right axis as x , the up/down axis as y axis, and the forward/backwards axis as z . So, to correct the left/right location of the target in the camera frame, participants would move their hand along the x axis. To correct the up/down location of the target in the camera frame, participants would move their hand along the y axis. When all the four tactors were activated, the participants would move towards the target along the z axis. Figure 13 shows the three axes with respect to the 4 LED targets on the black poster.

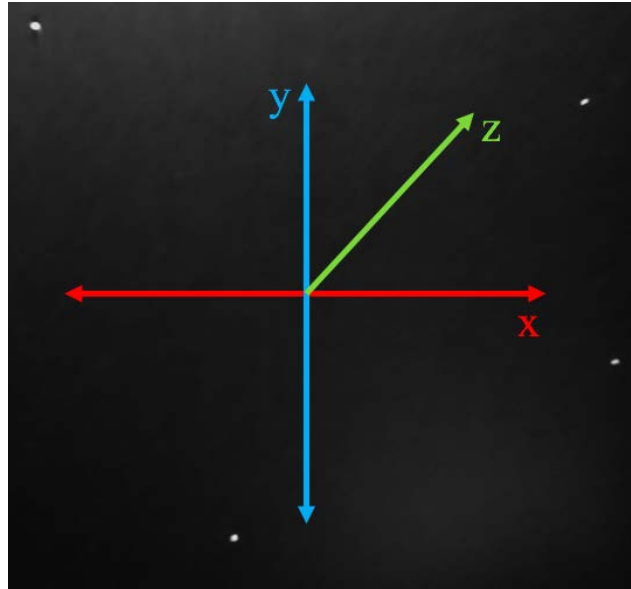


Figure 13 Axes for guidance

(1) Worst Axis First, as the name suggests, is based on an algorithm that determines if the participant is further away in x direction or in y direction and indicates the user correct in the worst direction first. For example, if the Cartesian coordinates of the target object relative to the center of the frame (where the target is desired to be) are $(20,50)$ the user gets a signal to move their hand upward first, in turn minimizing the distance (error) in y direction. Once the y distance falls below a certain threshold, the participant is indicated to move in the x direction i.e. towards the right. Thus, with each new frame received from the camera, we determine the worst axis and correct it first. The threshold was based on the size (radius) of the target object in the acquired frame. If the center of an axis was a pixel that fell within the contours of the *blob*, the algorithm considered the subject to be aligned in that specific direction. (It was hypothesized that this strategy would make it easier for participants to attend to the signal since it focused on moving their hand only in one cardinal direction at a time.

(2) The Adjacent Pair strategy (also called the Adjacent Tactor Pair) is based on signaling the participant to move in both the x and y directions, by alternating signals between right-left and up-down, one after the other. Thus, instead of first correcting one of the directions completely, participants are guided in both x and y directions, if appropriate, over a sequence of two instructions. From the earlier example, if the Cartesian co-ordinates of the target object were (20,50), the signal would guide them to move right and up with the next two instructions. Once the finger was correctly aligned in one of the directions, the participants would get the same directional indication twice or more in succession, which in turn also informed them that they were aligned in the other direction. Subsequent mis-alignment would reinstate the alternation of both instructions.

(3) The Two Tactor Vector strategy was designed to test if two adjacent tactors could be activated at the same time to stimulate a vibration in the intermediate direction of the two tactors. Thus, if the target object was located at (20,50), we would simultaneously activate both the top and the right tactor. It was hypothesized that this would indicate the participant to move in an in-between direction similar to tracking a vector in between the two direction. Ideally this would minimize errors in both the directions simultaneously, which meant the participants would get close to the target object more quickly. If the participant was already aligned in one of the axes, only one tactor was activated.

(4) The Warmer-Cooler strategy was based on determining if the participant was moving closer the target object or away from it. Participants would search for the object and try to reach it without any directional haptic feedback. The only haptic instruction was whether the distance from the target was getting smaller or larger. If the finger was moving closer to the target object, then a tactor would be activated to indicate that they were moving in the correct direction.

In pilot testing all four of the hand guidance strategies, we found that the Two Tactor Vector and Warmer-Cooler strategies were ineffective. The problem with the Two Tactor Vector strategy was that it was difficult to determine which two tactors were activated; the likely problem was cross-talk due to transmission of vibration through bone and soft tissue. The Warmer-Cooler strategy failed because the lack of a directional feedback led to unsystematic search for improvement. Thus, it was evident that these two strategies could not effectively guide participants to reach the target object, and would lead to arm and hand fatigue in any reasonable experiment searching for the target.

Following the pilot study, the formal experiment was conducted with the two remaining guidance strategies, namely Worst Axis First and Adjacent Pair. This experiment consisted of four trials performed by subjects, two trials per guidance strategy. To make sure that subject could not predict the location of target, a different target was used in each trial. For a particular guidance strategy, the target object was LED 1 and LED 4 (diagonal pair), while for the other strategy the target objects were LED 2 and LED 3 (refer to Figure 13). The sequence of strategy and targets was counterbalanced across participants, to ensure there was no bias in the experiment.

6.0 Results

This chapter discusses the results of the two experiments described in the previous chapter.

6.1 Tactor Localization Results

All 30 participants completed Experiment 1. To analyze if participants got better at tactor discrimination with some additional training, we compared the first attempt and the best attempt for each participant. 18 participants repeated the experiment, since they initially reported three or more signals incorrectly. Table 1 and table 2 show the confusion matrix of errors after the first and best (last) training periods. The numbers 1 through 5 each represent a particular haptic instruction: (1) top, (2) right, (3) bottom, (4) left, and (5) all tactors activated. Only incorrect responses are shown, in which the reported instruction does not correspond to the activated tactor; thus, the diagonals of the matrices are left blank.

Table 1 Confusion matrix of error for first attempt

Activated Reported	1	2	3	4	5	Total Errors	Error Proportion
1		0	7	3	2	12	0.13
2	4		9	10	2	25	0.28
3	15	5		0	0	20	0.22
4	5	6	0		0	11	0.12
5	10	0	3	4		17	0.19

Table 2 Confusion matrix of errors for best attempt

Activated Reported	1	2	3	4	5	Total Errors	Error Proportion
1		0	0	1	0	4	0.04
2	1		9	4	2	16	0.18
3	4	2		0	2	8	0.09
4	7	5	0		1	13	0.14
5	5	0	1	1		7	0.08

The bar chart shown below in Figure 14 shows the comparison between the first and best attempts. It is evident that the mean number of errors were generally reduced significantly with additional training, especially for haptic instructions 2, 3, and 5, for which errors were larger at first. In summary the total proportion of errors reduced from 18.75% to 11.25%

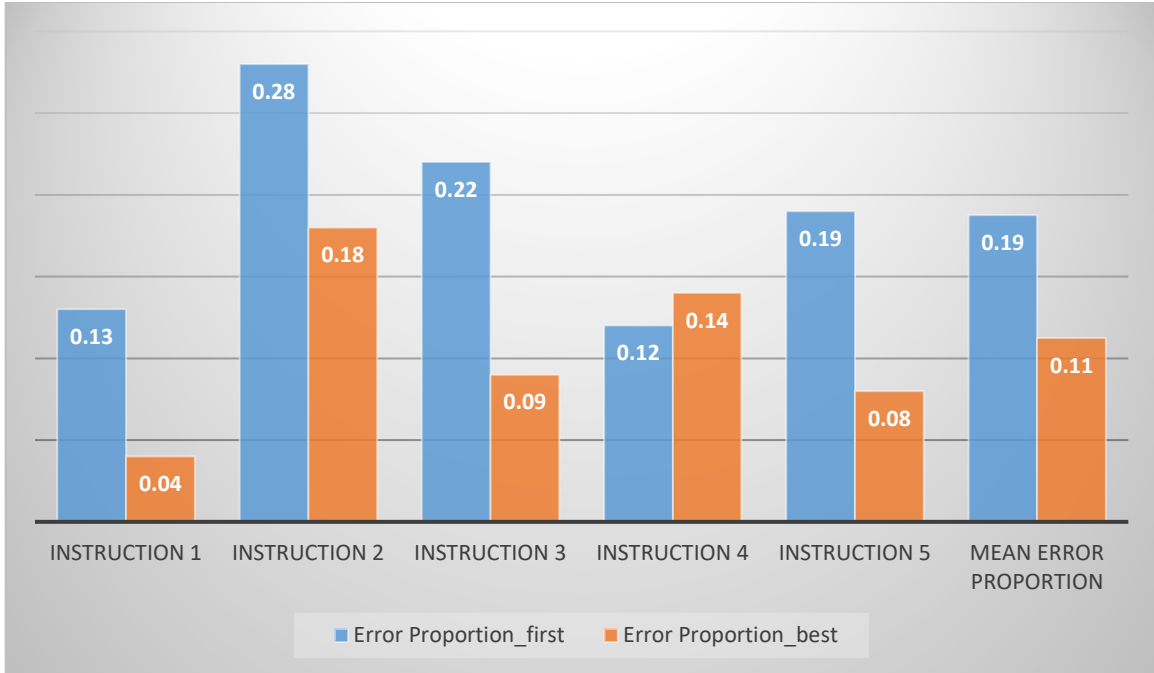


Figure 14 Comparison Between Mean Error Proportions: Experiment 1

6.2 Guidance to Target Results

The same thirty blindfolded participants who had completed Experiment 1 performed Experiment 2. During each trial, with a specific guidance strategy and a specific target, the haptic instruction delivered to the participant and the corresponding tracked locations of the markers were recorded. In order to analyze guidance to the target, we considered the distance of the participant's finger from the target in the x, y, and z directions, as well as the Euclidean distance of the finger from the target in three-dimensional space, given by the equation below, where (x_t, y_t, z_t) are the co-ordinates of the target location.

$$euclidean\ distance = \sqrt{((x - x_t)^2 + (y - y_t)^2 + (z - z_t)^2)} \quad (6-1)$$

From the data it was evident that subjects generally approached the target linearly with respect to time (as shown in Figure 17) in terms of the Euclidean distance from the point in time when they received the first haptic instruction until they were 10 cm from the target. A regression line was fit to the Euclidean distance vs. time during these periods. Figure 17 shows a representative example with the two guidance strategies. The goodness of fit (R^2) of this linear regression is shown as a histogram for the participant population in figure 15 for each guidance strategy. The slope of the linear fit and the time to approach the stopping point below 10 cm were measured for each trial. We also measured movement beyond that point, in terms of the minimum Euclidean distance reached relative to the LED and the total trial duration.

Out of the 30 participants, only data collected from 22 could be analyzed. Data collected from the remaining 8 participants had to be discarded, because 6 subjects had missing data from the tracker for an entire guidance strategy consisting of two trials, and 2 subjects had an unacceptable linear fit ($R^2 < 0.6$). For one of the subjects, who apparently had difficulty at the beginning of a trial responding correctly to the tactors, we kept the overall results after adjusting the period of descent to begin only when he started moving correctly in response to the haptic instruction.

The graphs shown below in Figure 16 summarizes the results. It shows, for each guidance strategy, the mean R^2 and the slope of the regression on Euclidean distance to the initial stopping at below 10 cm, along with the duration of that approach. Also shown is total time for the trial and the minimum Euclidean distance, where the trial ceased.

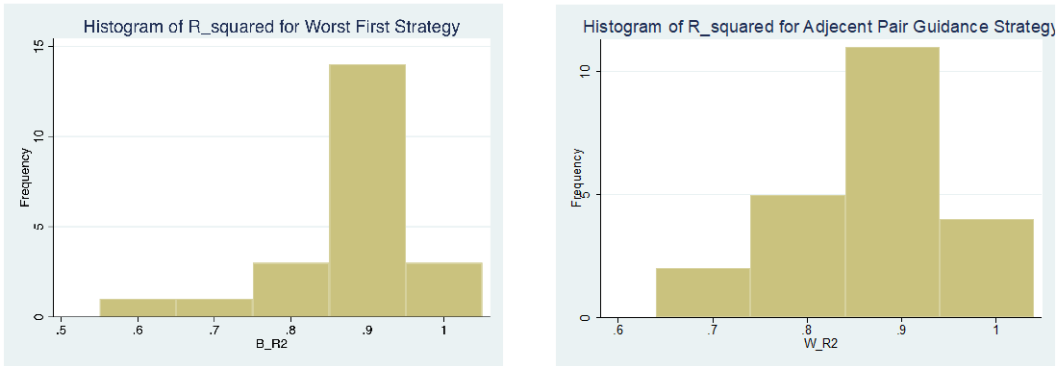


Figure 15 Histogram of R² for linear approach to the initial stopping point.

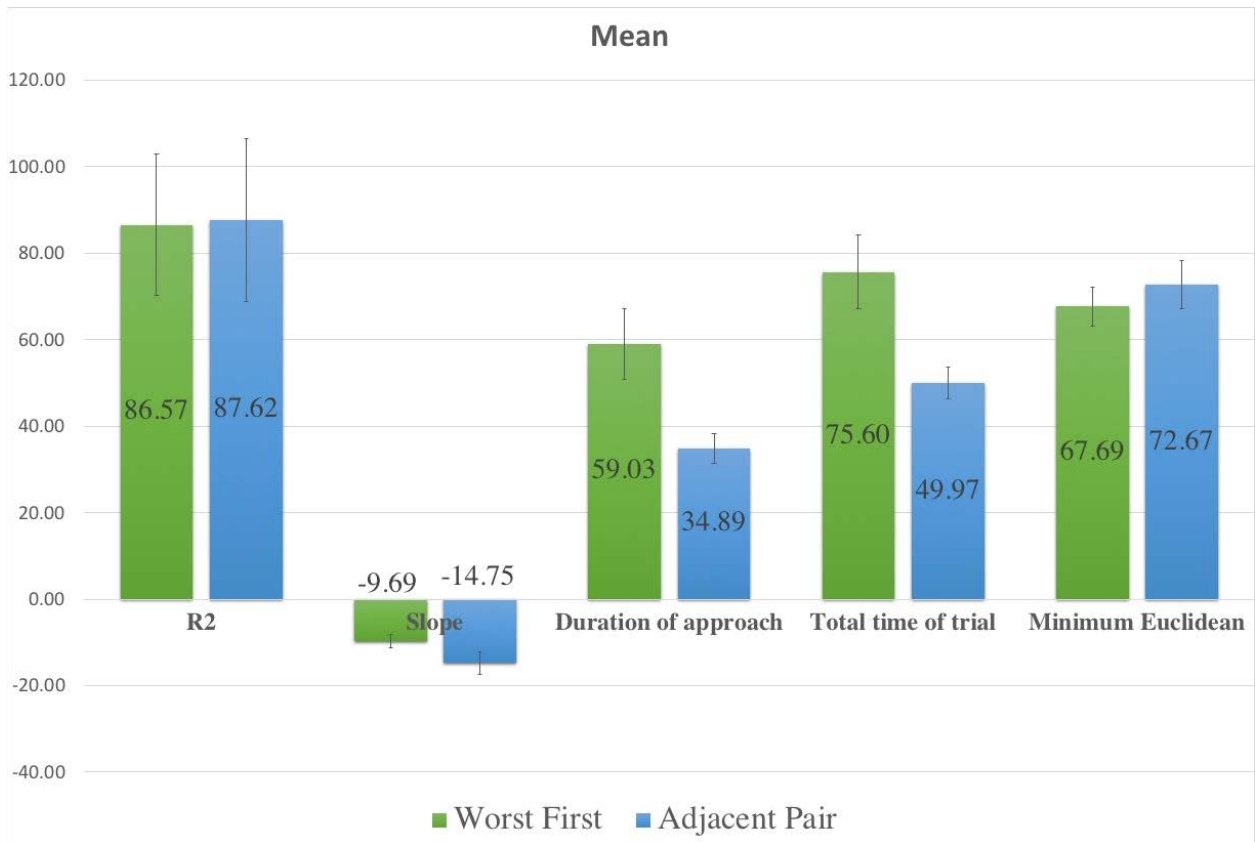


Figure 16 Means of the five parameters for each strategy. The error bar is 1 s.e.m.

From Figure 16 it appears that the mean slope, duration, and total time are all lower for the Adjacent Pair strategy, as compared to the Worst Axis First strategy. The correlation (R^2) and minimum Euclidean distance at the end of the trial, in contrast, are very similar. A paired t-test was performed on each of the five variables to test if they were significantly different for the two strategies. The results are summarized in Table 3 below.

Table 3 Statistical comparisons of experimental variables using Student's *t*-test

Mean difference: Worst-First minus Adjacent-Pair	Test Statistic (<i>t</i>)	<i>p</i>
R^2 for linear approach	-0.57	0.57 (ns)
Slope of approach	-2.24	0.04*
Duration of approach	3.24	0.004**
Total trial time	3.28	0.004**
Minimum Euclidean distance	-0.62	0.54 (ns)

The differences in slope and approach duration to reach the target location, as well as total trial time, are significantly different for the two guidance strategies, with an advantage for the Adjacent Pair guidance. As noted above, R^2 and minimum Euclidean distance were similar for the two guidance strategies, indicating comparable regularity in approach toward the final stopping

point and the final distance from the target. This conclusion was supported by the finding that the t-test failed to reach significance.

Figure 17 shows a representative example of the slopes and correlation of the regression line for the two different strategies. While it appears that an exponential might be a better fit for the Worst-First strategy, the linear fit meets our criterion, and it is clear that the regressions reflect the slower approach to the target for Worst-Fit.

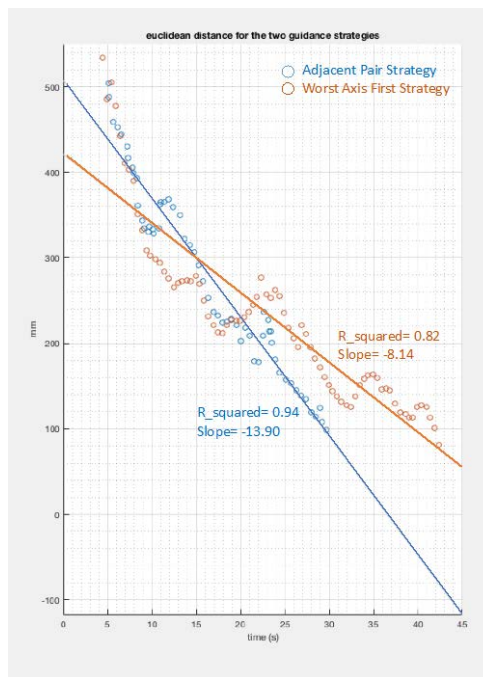


Figure 17 Illustration of regression of Euclidean distance

An explanation for the disadvantage of the Worst Axis First is that while correcting the more distant direction from the target, participants could not adequately monitor the other, causing them to drift, with the results of decreasing overall accuracy and increasing time. Just this trend

is clearly visible in the sample below showing a participant's trajectory on the x and y axes for each strategy. There is a clear zig-zag as corrections trade off in Worst First, as opposed to the regular approach on both axes for Adjacent Pair, as shown in Figure 18.

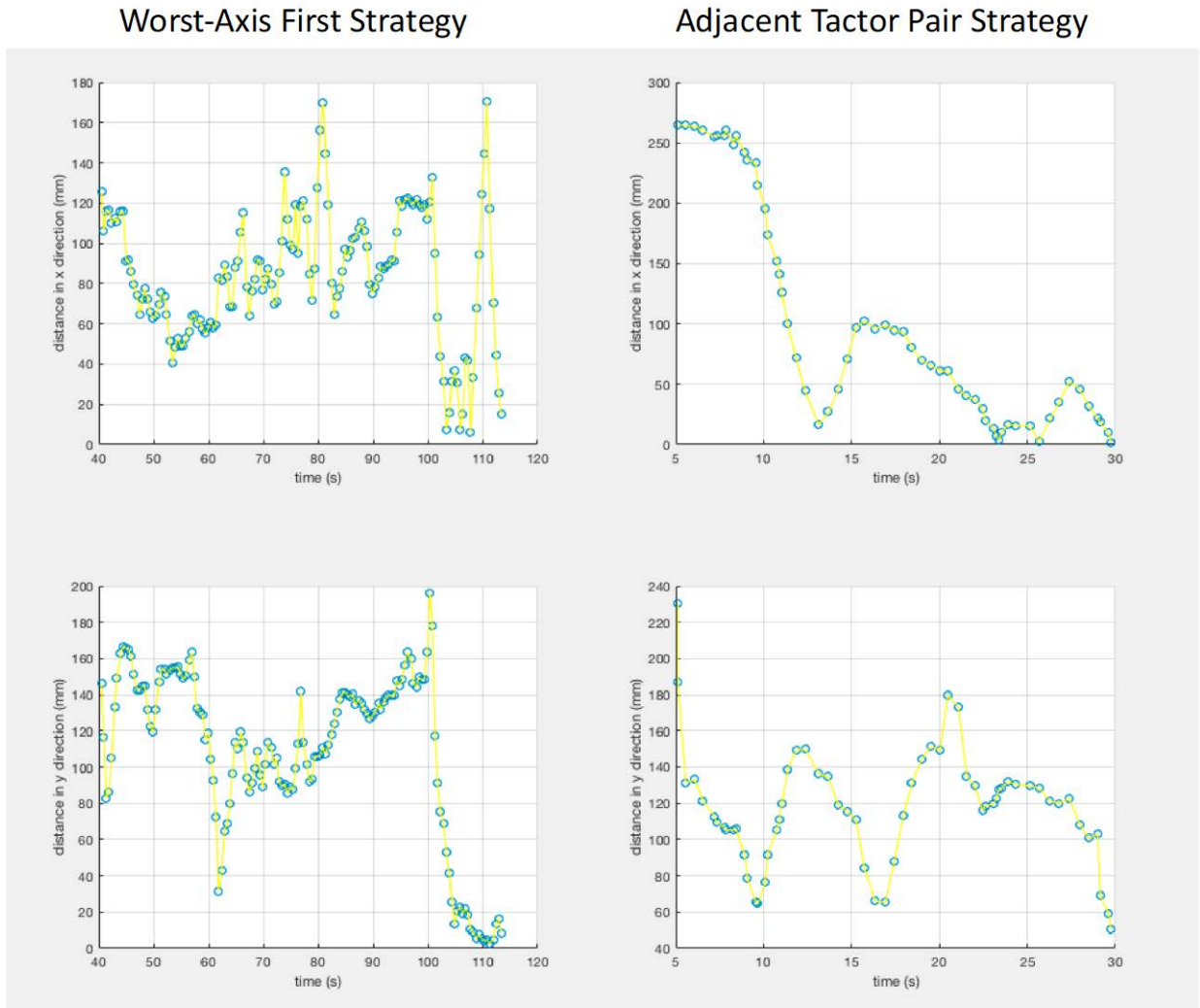


Figure 18 Motion in x and y directions for each guidance strategy

7.0 Control System Analysis

Classical control theory principles can be used to understand the behavior of a dynamic system and to regulate that system. The system that is being controlled is called a *plant*, and the system controlling the plant is referred to as the *controller*. Feedback is fed from the output of the plant back to the controller, informing the controller about the current state of the plant (see Fig. 19). Depending on the reference signal X_{ref} and the output of the plant X , the controller can feedforward an appropriate signal u to the plant, in turn regulating the plant. In the framework of classical control, the human subject who is being guided to reach the target acts as the plant. The FingerSight, system consisting of the camera, Arduino Uno and the four haptic factors, act as the controller, as shown in Figure 19.

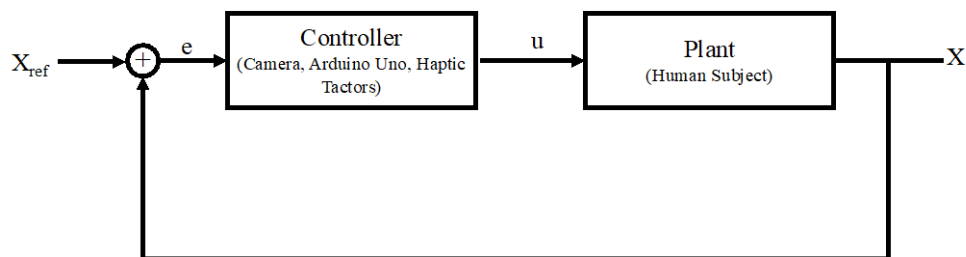


Figure 19 Classical control system block diagram for FingerSight

7.1 Model of the Plant

Since a human being is involved in the control loop, the output of the plant contains unpredictable noise and disturbances. Data acquired by the optical tracker during a representative trial with the Adjacent-Pair guidance strategy was used to model the plant. The two directions of guidance, namely x and y , were decoupled to analyze the response of the plant. The input signal to the plant is a rectangular pulse with a duration of 90 ms, delivered once each second. During this pulse, one of the factors in x direction (left/right) or in y direction (top/bottom) is activated. The output of the plant is the location of the finger in that corresponding direction. In other words, if the input to the plant is a pulse directing the human to move their hand in the left or right direction, the corresponding output of plant is the responding changes in the x coordinate of the hand. This continues until the output of the plant is matched with the x_{target} , the x coordinate of the target object. Figure 20 shows the input and output of the plant for a representative example using the Adjacent-Pair guidance strategy.

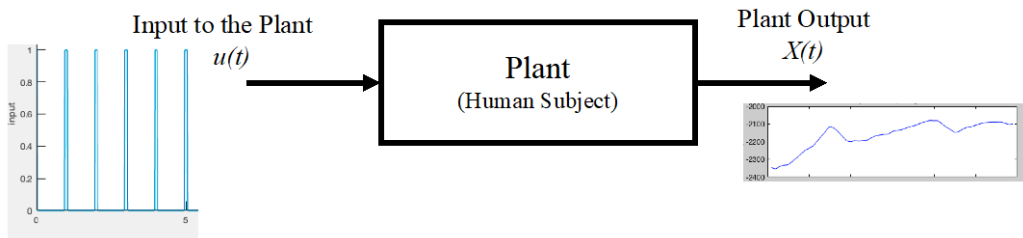


Figure 20 Illustration of input and output signals of the Plant

As already stated, since the plant consists of a human user, the output of the plant is prone to large errors and disturbances. This can be clearly seen in the representative output of the plant as shown in in Figure 21. In order to model the relationship between the input and output of the plant, an ARMAX model is used. ARMAX stands for *autoregressive-moving average with exogenous terms*. ARMAX modelling is useful in the reliable estimation of stochastic dynamics of a system in the presence of strong noise and dominating disturbances [24] [25]. The relationship between input and output and the error are estimated using a recursive extended last square (RELS) method. This method minimizes the sum of squares of prediction errors [26]. For a system with input $u(k)$, also termed the *exogenous variable*, output $y(k)$, also called the autoregressive variable, and moving average or white noise $e(k)$, the linear differential equation of ARMAX model can is given by the following equation:

$$\begin{aligned}
 y(k) + a_1y(k - 1) + \dots + a_{n_a}y(k - n_a) & \quad (7-1) \\
 = b_1u(k - n_k) + \dots + b_{n_b}u(k - n_k - n_b + 1) + c_1e(k - 1) + \dots \\
 + c_{n_c}e(k - n_c) + e(k)
 \end{aligned}$$

$n_a = \text{number of poles}$

$n_b = \text{number of zeros} + 1$

$n_c = \text{number of } C \text{ coefficients}$

$n_k = \text{number of samples that occur before the input affects the output}$

For ease of understanding, the above equation can be rewritten as follows, where q is the delay operator.

$$A(q)y(k) = B(q)u(k - n_k) + C(q)e(k) \quad (7-2)$$

ARMAX (n_a, n_b, n_c, n_k) models are defined by the four parameters n_a, n_b, n_c, n_k . The parameter n_a describes the number of previous output samples that affect the present output. Similarly, the parameter n_b+1 describes the number of previous inputs that affect the current output.

The system identification toolbox in MATLAB provides a user-friendly graphical user interface to formulate ARMAX models based on the input and output data [27]. Using this toolbox, ARMAX models of various orders were formulated, trading the complexity of the model off against the goodness of the corresponding fit. ARMAX (2,1,1,1) was found to yield an acceptable fit, in which the input to the plant and white noise explained 65.02% of variation in the output of the plant. ARMAX(2,1,1,1) as shown in Figure 21. The equation for this model and the transfer function of the plant obtained from it is given by the equations (7-3) and (7-4) respectively.

$$\begin{aligned} & (1 - 1.839q^{-1} + 0.839q^{-2})y(k) \\ & = (-9.785q^{-1} + 8.394q^{-2})u(k) + (1 + 0.768801q^{-1})e(k) \end{aligned} \quad (7-3)$$

$$\text{Transfer Function } G(z) = \frac{Y(Z)}{X(z)} = \frac{(-9.785z^{-1} + 8.394z^{-2})}{(1 - 1.839z^{-1} + 0.839z^{-2})}. \quad (7-4)$$

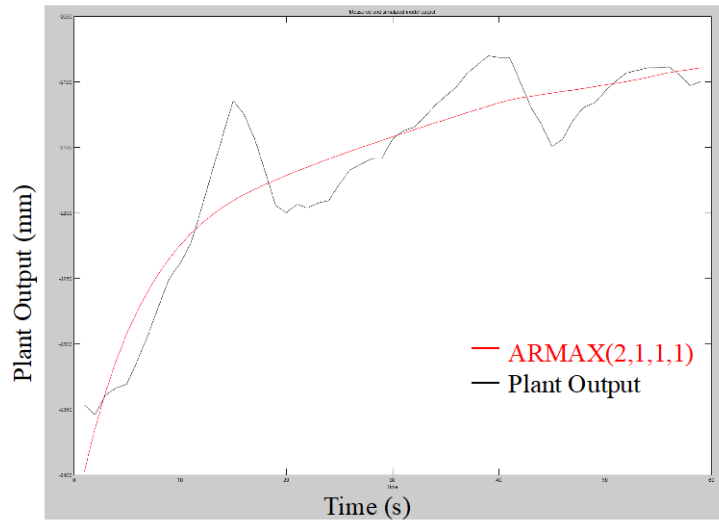


Figure 21 ARMAX model with 65.02% fit

7.2 Simulation of the Control System

The transfer function for the plant obtained from the ARMAX model was used to simulate a closed-loop control system based on a proportional-integral-derivative (PID) controller to regulate the plant.

The transfer function of a discrete PID controller is as follows:

$$C(Z) = K_p + K_i \frac{T_s Z + 1}{2 Z - 1} + K_d \frac{Z - 1}{Z T_s} \quad (7-5)$$

$K_p = \text{Proportional Gain}$

$K_i = \text{Integral Gain}$

$K_d = \text{Derivative Gain}$

$TS = \text{Sample time}$

The output of controller $u(t)$ for a PID controller is equal to

$$u(t) = K_P e(t) + K_I \int e(t) dt + K_D \frac{de(t)}{dt} \quad (7-6)$$

Using the MATLAB Simulink program, a closed loop control system was simulated as shown in Figure 22. The PID controller was tuned using the built-in PID tuner in Simulink. Manual adjustments were made to the response time and the transient behavior was optimized to minimize rise time and settling time. Figure 23 shows the step response of the tuned PID controller, along with the control parameters.

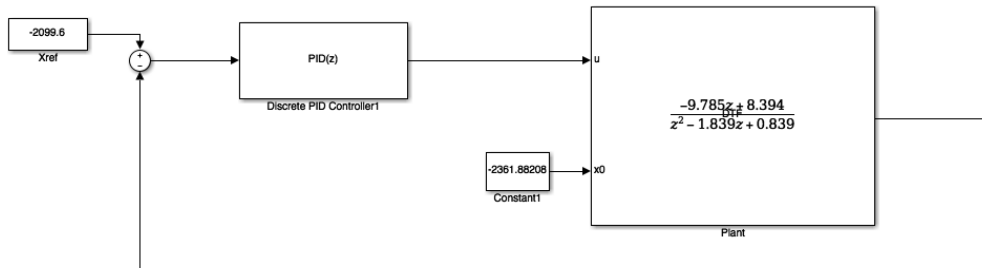


Figure 22 Simulation of control system in Simulink

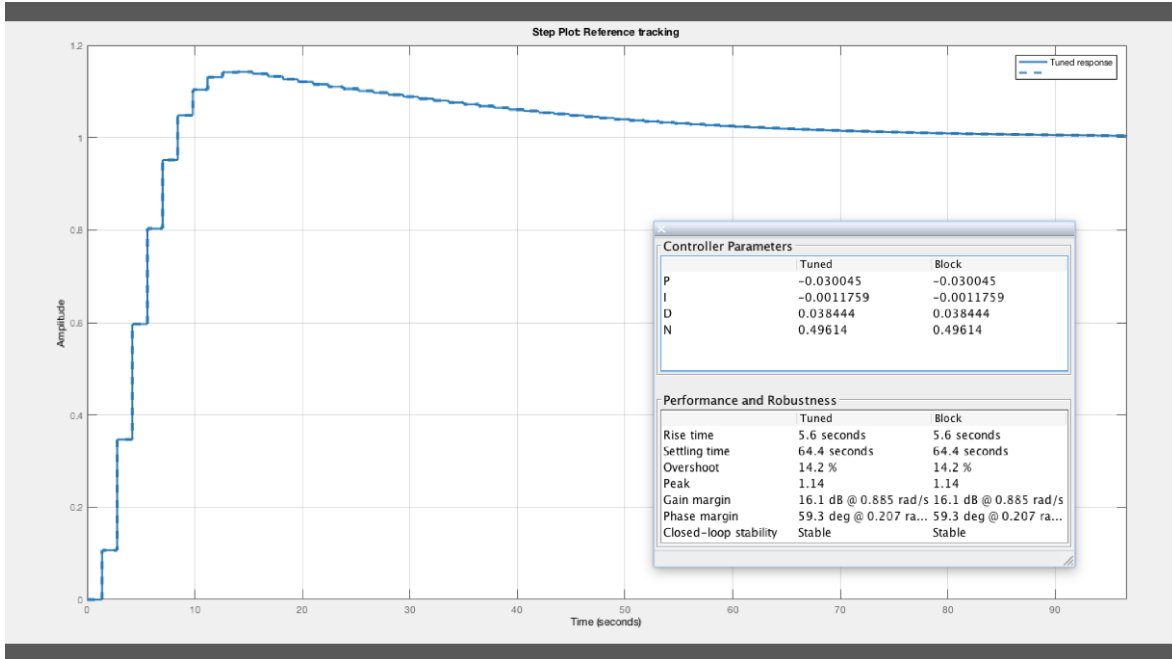


Figure 23 Step response of the PID controller

The values of the gains and the performance parameters of the PID controller are summarized in Table 4 shown below. The output of the plant regulated by the tuned PID controller exhibits much less disturbance and noise, as shown in Figure 24. The transfer function of the PID controller can be written as follows, with a sample time of 0.5 s:

$$C(Z) = (-0.0300) - (0.00117) \frac{(0.5)Z + 1}{2} \frac{Z + 1}{Z - 1} + (0.03844) \frac{Z - 1}{0.5Z} \quad (7-7)$$

$$C(Z) = \frac{(0.0466)Z^2 - (0.1242)Z + 0.0769}{Z^2 - Z} \quad (7-8)$$

Table 4 Performance of PID controller

K_P	-0.0300
K_I	-0.00117
K_D	0.03844
Rise Time	5.6 s
Settling Time	64.4 s
Closed Loop Stability	Stable

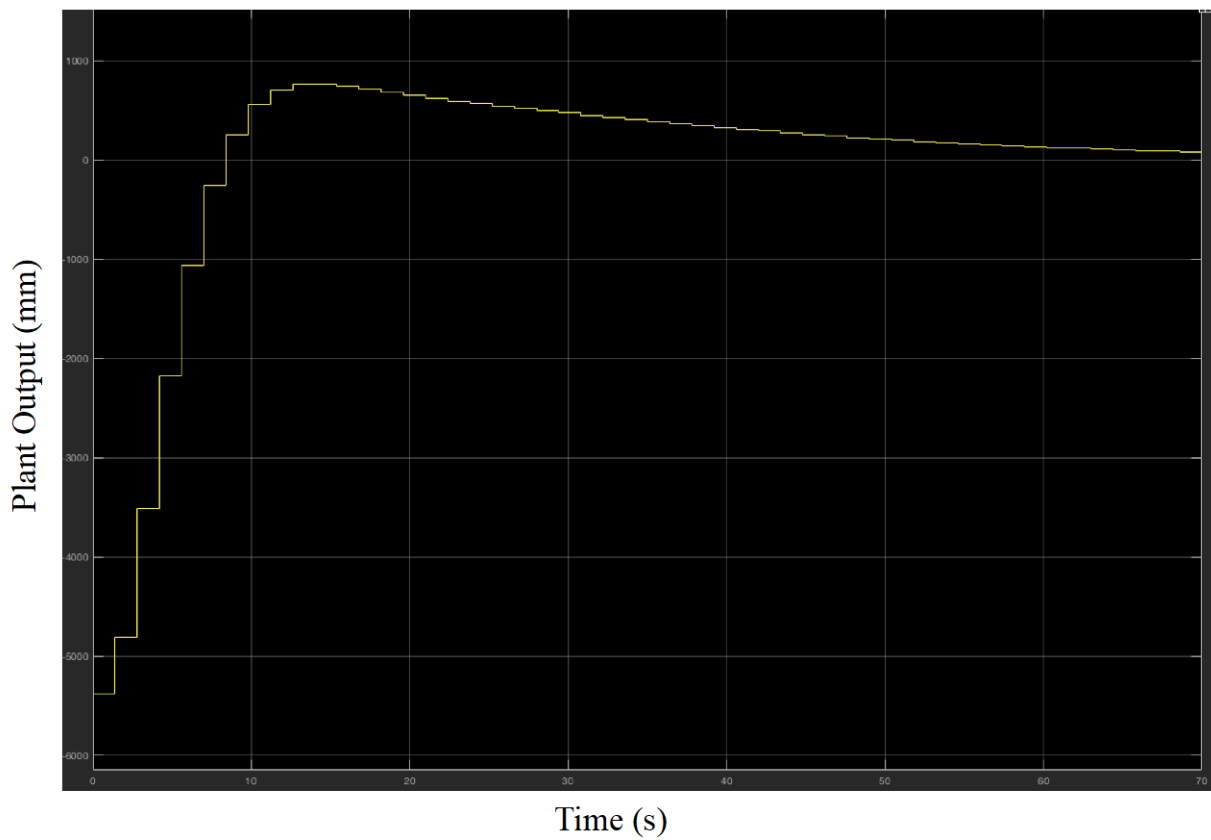


Figure 24 Output of the plant regulated by the tuned PID controller

Since the plant model was based on actual motion data collected from the tracker, this simulation of the control system may provide insights into possible improvements to the FingerSight prototype and hand guidance strategy.

8.0 Conclusion

This thesis demonstrates that FingerSight can be used by visually impaired people to locate and reach objects in their peripersonal space. The experiments demonstrated that untrained people can discriminate haptic feedback in four cardinal directions, and that this feedback can be used to guide them to target objects. Presumably, a habitual visually impaired user would quickly capitalize on experience to perform even more effectively. Since FingerSight does not rely on audio/sonic feedback, nor does it generate high-dB sound, FingerSight should minimally if at all interfere with the sounds in the environment that blind people rely on for performing other tasks. Objects used in day to day activities tend to be substantially larger than the LEDs used as target objects in our experiment. We believe that the task of locating and guiding the hand in the direction of larger objects is a much simpler task that can likely be accomplished in a significantly smaller time.

FingerSight is a commercially viable device, in terms of prospective expense of fabrication, power consumption, and ease of use. Its current cost of components, \$60, would be reduced by mass fabrication. Computation could be offloaded to a cloud-based agent or a smartphone application. Such accessible enabling technologies are critical in alleviating the known linkage between poverty and vision loss [28].

8.1 Future Work

FingerSight can be trained to recognize objects used in day-to-day lives by training a neural network, which a neural network on a case to case basis to recognize objects that a given visually impaired person uses. This would enhance commercial value. As noted above, FingerSight could be interfaced with a smart phone, using an Android and an OIS application, based on the principles of the current algorithms for object detection and guidance strategy. The application could be enhanced with the capability to receive audio commands. The visually impaired user could just say “FingerSight: Find my Wallet”, and the application would respond to haptic feedback to locate and reach the object, as done by users in Experiment 2. The guidance strategy can further be improved on the basis of the PID controller shown in chapter 7.

During the course of the development of the prototype for FingerSight, we presented FingerSight at two networking and commercialization events. Investors and companies thought that FingerSight is a promising device and has the potential to change the lives of those deprived of sight.

Appendix A Program for FingerSight

Appendix A.1 and A.2 consist of the program for FingerSight, used in the experiments, written using Python. Appendix A.3 consists of the program used to interface the Arduino, programmed using the Arduino IDE.

A.1 Program for FingerSight with Worst Axis First guidance strategy

```
from collections import deque
import numpy as np
import argparse
import imutils
import cv2
import serial
import time
import math
#print("THIS IS DONE")
arduino = serial.Serial('/dev/tty.usbserial-DN02MZ74',9600)
time.sleep(2) #wait for 2 seconds for the communication to get established
arduino.readline() #read the serial data and print it as li
ap = argparse.ArgumentParser()
ap.add_argument("-b", "--buffer", type=int, default=64,
                help="max buffer size")
args = vars(ap.parse_args())

if not args.get("video", False):
    camera = cv2.VideoCapture(0)

LED_HSV_Lower = (0,0,235)
LED_HSV_Upper = (180,18,255)
pts = deque(maxlen=args["buffer"])
see=0

import xlwt
wb=xlwt.Workbook()
```

```

sheet=wb.add_sheet("Experiment_Part2")

count=[]
rad=[]
erx=[]
ery=[]
instr=[]
cenlocx=[]
cenlocy=[]
#print("Start Data Collection")
a=time.time()
arduino.write(('6').encode())
time.sleep(0.1)

while True:
    arduino.write(('8').encode())
    ser_data = arduino.readline()
    (grabbed, frame) = camera.read()

    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)

    # construct a mask for the color "green", then perform
    # a series of dilations and erosions to remove any small
    # blobs left in the mask
    mask = cv2.inRange(hsv, greenLower, greenUpper)
    mask = cv2.erode(mask, None, iterations=2)
    mask = cv2.dilate(mask, None, iterations=2)
    # find contours in the mask and initialize the current
    # (x, y) center of the ball
    cnts = cv2.findContours(mask.copy(), cv2.RETR_EXTERNAL,
        cv2.CHAIN_APPROX_SIMPLE)[-2]
    center = None

    # only proceed if at least one contour was found
    if len(cnts) > 0:
        see=1
        # find the largest contour in the mask, then use
        # it to compute the minimum enclosing circle and
        # centroid
        c = max(cnts, key=cv2.contourArea)
        ((x, y), radius) = cv2.minEnclosingCircle(c)
        M = cv2.moments(c)
        center = (int(M["m10"] / M["m00"]), int(M["m01"] / M["m00"]))
        sf = frame.shape

```

```

# only proceed if the radius meets a minimum size
#if radius > 10:
# draw the circle and centroid on the frame,
# then update the list of tracked points
cv2.circle(frame, (int(x), int(y)), int(radius),(0, 255, 255), 2)
cv2.circle(frame, center, 5, (0, 0, 255), -1)
cv2.circle(frame, (int(sf[1]/2),int(sf[0]/2)), 5, (0,0,255), -1)
cv2.line(frame, (int(sf[1]/2),0),(int(sf[1]/2),int(sf[0])),(255,0,0),1,1)
cv2.line(frame, (0,int(sf[0]/2)),(int(sf[1]),int(sf[0]/2)),(255,0,0),1,1)
if radius>0:
    #print("Radius= %d",radius)
    Xerror= abs((sf[1]/2)-center[0])
    Yerror= abs((sf[0]/2)-center[1])
    #print(Xerror,Yerror)
    if radius>140:
        b=time.time()
        c=b-a
        count.append(c)
        rad.append(radius)
        erx.append(Xerror)
        ery.append(Yerror)
        cenlocx.append(center[0])
        cenlocy.append(center[1])
        instr.append("SH")
        #arduino.write(('7').encode())

        #print("Go straight and Hit the Dot")
        #arduino.write(('1').encode())
        #arduino.write(('2').encode())
        #arduino.write(('3').encode())
        arduino.write(('9').encode())
        time.sleep(2)
        arduino.write (('0').encode())
        time.sleep(0.5)
        #arduino.write(('1').encode())
        #arduino.write(('2').encode())
        #arduino.write(('3').encode())
        arduino.write(('9').encode())
        time.sleep(2)
        arduino.write(('0').encode())
        #print("***** THE END *****")
        break

    if center[0]<(sf[1]/2)+1*radius and center[0]>(sf[1]/2)-1*radius and
center[1]<(sf[0]/2)+1*radius and center[1]>(sf[0]/2)-1*radius:

```

```

print("Go Straight and Hit The DOT")
b=time.time()
c=b-a
count.append(c)
rad.append(radius)
erx.append(Xerror)
ery.append(Yerror)
cenlocx.append(center[0])
cenlocy.append(center[1])
instr.append("S")
arduino.write(('7').encode())
#arduino.write(('1').encode())
#arduino.write(('2').encode())
#arduino.write(('3').encode())
arduino.write(('5').encode())
time.sleep(0.2)
arduino.write(('0').encode())

```

elif Xerror>Yerror:

```

print("Fix LEFT RIGHT")
if center[0]<(sf[1]/2):
    print("Go Left <----- <---- <----- <-----")
    b=time.time()
    c=b-a
    count.append(c)
    rad.append(radius)
    erx.append(Xerror)
    ery.append(Yerror)
    cenlocx.append(center[0])
    cenlocy.append(center[1])
    instr.append("L")
    arduino.write(('7').encode())

    arduino.write(('4').encode())
    time.sleep(0.07)
    arduino.write(('0').encode())
elif center[0]>(sf[1]/2):
    #print("Go RIGHT ----> -----> ----->")
    b=time.time()
    c=b-a
    count.append(c)
    rad.append(radius)
    erx.append(Xerror)
    ery.append(Yerror)
    cenlocx.append(Xerror)

```


else:

```
b=time.time()
c=b-a
count.append(c)
rad.append(0)
erx.append(0)
ery.append(0)
cenlocx.append(0)
cenlocy.append(0)
instr.append("M")
```

```
pts.appendleft(center)
```

```
# show the frame to our screen
cv2.imshow("Frame", frame)
```

```
key = cv2.waitKey(1) & 0xFF
```

```
# if the 'q' key is pressed, stop the loop
```

```
for i in range (0,len(rad)):
    sheet.write(i,0,count[i])
    sheet.write(i,1,rad[i])
    sheet.write(i,2,erx[i])
    sheet.write(i,3,ery[i])
    sheet.write(i,4,cenlocx[i])
    sheet.write(i,5,cenlocy[i])
    sheet.write(i,6,instr[i])
wb.save("Part2_Worst_LED3_Sub29")
```

```
# cleanup the camera and close any open windows
camera.release()
cv2.destroyAllWindows()
```

A.2 Program for FingerSight with Adjacent Tactor Pair guidance strategy

```
from collections import deque
import numpy as np
import argparse
import imutils
import cv2
import serial
import time
import math

arduino = serial.Serial('/dev/tty.usbserial-DN02MZ74',9600)
time.sleep(2) #wait for 2 seconds for the communication to get established
arduino.readline() #read the serial data and print it as line

ap = argparse.ArgumentParser()
ap.add_argument("-b", "--buffer", type=int, default=64,
                help="max buffer size")
args = vars(ap.parse_args())

if not args.get("video", False):
    camera = cv2.VideoCapture(0)

LED_HSV_Lower = (0,0,235)
LED_HSV_Upper = (180,18,255)
pts = deque(maxlen=args["buffer"])
see=0
import xlwt
wb=xlwt.Workbook()
sheet=wb.add_sheet("Exp_both")

count=[]
rad=[]
inst=[]
cenx=[]
ceny=[]
a=time.time()
arduino.write(('6').encode())
time.sleep(0.1)

while True:
    arduino.write(('8').encode())
    ser_data = arduino.readline()
```

```
(grabbed, frame) = camera.read()
```

```
hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
```

```
# construct a mask for specified HSV limits
# a series of dilations and erosions to remove any small
# blobs left in the mask
mask = cv2.inRange(hsv, LED_HSV_Lower, LED_HSV_Upper)
mask = cv2.erode(mask, None, iterations=2)
mask = cv2.dilate(mask, None, iterations=2)
# find contours in the mask and initialize the current
# (x, y) center of the ball
cnts = cv2.findContours(mask.copy(), cv2.RETR_EXTERNAL,
                        cv2.CHAIN_APPROX_SIMPLE)[-2]
center = None

# only proceed if at least one contour was found
if len(cnts) > 0:
    see=1
    # find the largest contour in the mask, then use
    # it to compute the minimum enclosing circle and
    # centroid
    c = max(cnts, key=cv2.contourArea)
    ((x, y), radius) = cv2.minEnclosingCircle(c)
    M = cv2.moments(c)
    center = (int(M["m10"] / M["m00"]), int(M["m01"] / M["m00"]))
    print("centre=", center)
    print("radius=",radius)
    sf = frame.shape
    cv2.circle(frame, (int(x), int(y)), int(radius),(0, 255, 255), 2)
    cv2.circle(frame, center, 5, (0, 0, 255), -1)
    cv2.line(frame, (int(sf[1]/2),0),(int(sf[1]/2),int(sf[0])),(255,0,0),1,1)
    cv2.line(frame, (0,int(sf[0]/2)),(int(sf[1]),int(sf[0]/2)),(255,0,0),1,1)
    if radius > 0:
        print("circle found with raduis r= %d",radius)
        if radius>140:
            b=time.time()
            c=b-a
            count.append(c)
```

```

rad.append(radius)
cenx.append(center[0])
ceny.append(center[1])
inst.append("SH")
print("Go straight and hit the target")
#arduino.write(('1').encode())
#arduino.write(('2').encode())
#arduino.write(('3').encode())
arduino.write(('9').encode())
time.sleep(2)
arduino.write (('0').encode())
time.sleep(0.5)
#arduino.write(('1').encode())
arduino.write(('9').encode())
#arduino.write(('3').encode())
#arduino.write(('4').encode())
time.sleep(2)
arduino.write (('0').encode())
print("also the end of the program")
break

```

```

if center[0]<(sf[1]/2)+2*radius and center[0]>(sf[1]/2)-2*radius:
    print("left right aligned-----")
    if center[1]<(sf[0]/2)+2*radius and center[1]>(sf[0]/2)-2*radius:

```

```

        print("UP DOWN ALIGNE||||||||||||||||||||||||||||||||||||")
        print(".....GO STRAIGHT AHEAD.....all buzz")
        b=time.time()
        c=b-a
        count.append(c)
        rad.append(radius)
        cenx.append(center[0])
        ceny.append(center[1])
        inst.append("S")
        #arduino.write(('1').encode())
        #arduino.write(('2').encode())
        #arduino.write(('3').encode())
        arduino.write(('5').encode())
        time.sleep(0.07)
        arduino.write (('0').encode())

```

```

elif center[1]<(sf[0]/2)-2*radius:

```

```

print("GO UP")
b=time.time()
c=b-a
count.append(c)
rad.append(radius)
cenx.append(center[0])
ceny.append(center[1])
inst.append("U")
arduino.write(('1').encode())
time.sleep(0.07)
arduino.write (('0').encode())
updown=1

elif center[1]>(sf[0]/2)+2*radius:
print("GO DOWN")
b=time.time()
c=b-a
count.append(c)
rad.append(radius)
cenx.append(center[0])
ceny.append(center[1])
inst.append("D")
updown=2
arduino.write(('3').encode())
time.sleep(0.07)
arduino.write (('0').encode())

elif center[0]<(sf[1]/2)-2*radius:
print("Go Left  <-----  <----  <-----  <-----")
leftright=1
b=time.time()
c=b-a
count.append(c)
rad.append(radius)
cenx.append(center[0])
ceny.append(center[1])
inst.append("L")
arduino.write(('4').encode())
time.sleep(0.07)
arduino.write (('0').encode())

if center[1]<(sf[0]/2)+2*radius and center[1]>(sf[0]/2)-2*radius:
print("ALREADY ALIGNED IN THE VERTICAL DIRECTION")
elif center[1]<(sf[0]/2)-2*radius:
print("go up ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^")

```

```

b=time.time()
c=b-a
count.append(c)
rad.append(radius)
cenx.append(center[0])
ceny.append(center[1])
inst.append("U")
updown=1
arduino.write(('1').encode())
time.sleep(0.07)
arduino.write (('0').encode())

```

```

elif center[1]>(sf[0]/2)+2*radius:
print("go DOWN \\\\\\")
updown=2
b=time.time()
c=b-a
count.append(c)
rad.append(radius)
cenx.append(center[0])
ceny.append(center[1])
inst.append("D")
arduino.write(('3').encode())
time.sleep(0.07)
arduino.write (('0').encode())

```

```

elif center[0]>(sf[1]/2)+2*radius:
print("Go RIGHT ----> -----> ----->")
leftright=2
b=time.time()
c=b-a
count.append(c)
rad.append(radius)
cenx.append(center[0])
ceny.append(center[1])
inst.append("R")
arduino.write(('2').encode())
time.sleep(0.07)
arduino.write (('0').encode())

```

```

if center[1]<(sf[0]/2)+2*radius and center[1]>(sf[0]/2)-2*radius:
print("ALREADY ALIGNED IN TOP-BOTTOM AXIS")
elif center[1]<(sf[0]/2)-2*radius:
print("go up ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^")

```

```
b=time.time()
c=b-a
count.append(c)
rad.append(radius)
cenx.append(center[0])
ceny.append(center[1])
inst.append("U")
updown=1
arduino.write(('1').encode())
time.sleep(0.07)
arduino.write (('0').encode())
```

```
elif center[1]>(sf[0]/2)+2*radius:
    print("go down \\\\\\\\\\\\\\\")
    b=time.time()
    c=b-a
    count.append(c)
    rad.append(radius)
    cenx.append(center[0])
    ceny.append(center[1])
    inst.append("D")
    updown=2
    arduino.write(('3').encode())
    time.sleep(0.07)
    arduino.write (('0').encode())
```

```
cv2.line(frame, (int(sf[1]/2),0),(int(sf[1]/2),int(sf[0])),(255,0,0),1,1)
cv2.line(frame, (0,int(sf[0]/2)),(int(sf[1]),int(sf[0]/2)),(255,0,0),1,1)
```

else:

```
    b=time.time()
    c=b-a
    count.append(c)
    rad.append(0)
    cenx.append(0)
    ceny.append(0)
    inst.append("M")
# update the points queue
pts.appendleft(center)

# show the frame to our screen
cv2.imshow("Frame", frame)

key = cv2.waitKey(1) & 0xFF
```



```

    # if the 'q' key is pressed, stop the loop

for i in range (0,len(rad)):
    sheet.write(i,0,count[i])
    sheet.write(i,1,rad[i])
    sheet.write(i,2,cenx[i])
    sheet.write(i,3,ceny[i])
    sheet.write(i,4,inst[i])
wb.save("Part2_both_LED2_Sub30")

# cleanup the camera and close any open windows
camera.release()
cv2.destroyAllWindows()

```

A.3 Program to control the Arduino Uno Board

```

int data;
const int motorPina = 3; //a
const int motorPinb = 5; //b
const int motorPinc = 9; //c
const int motorPind = 10; //d
const int buff1 = 6; //From SCU via Certus
const int buff2 = 7; //From SCU via certus
int LED3 = A0;
int LED4 = A1;
int LED1 = A4;
int LED2 = A5;
int clk_decade = 12; //output of the decade counter which is (freq of SCU clock/10),
active high
//int storedState_clk_decade;
bool waiting_for_negative_edge;

void setup()
{
    Serial.begin(9600); //initialize serial COM at 9600 baudrate
    pinMode(LED_BUILTIN, OUTPUT); //make the LED pin (13) as output
    pinMode(motorPina, OUTPUT);
    pinMode(motorPinb, OUTPUT);
    pinMode(motorPinc, OUTPUT);
    pinMode(motorPind, OUTPUT);
}

```

```

pinMode(buff1, OUTPUT);
pinMode(buff2, OUTPUT);
digitalWrite (LED_BUILTIN, LOW);
Serial.println("Hi!, I am Arduino");

pinMode(LED1, OUTPUT);

}

void loop()
{
  analogWrite(LED2, 150);
  //analogWrite(LED4,150);

  if (Serial.available())
  {
    data = Serial.read();

    if (data == '1')
    {
      analogWrite(motorPina, 204);
      //delay(90);
      delay(120);
      analogWrite(motorPina, 0);
    }

    else if (data == '2')
    {
      analogWrite (motorPinb, 204);
      delay(90);

      analogWrite (motorPinb, 0);
    }

    else if (data == '3')
    {

      analogWrite (motorPinc, 204);
      delay(90);
      analogWrite (motorPinc, 0);
    }

    else if (data == '4')
    {
      analogWrite (motorPind, 204);
      //delay(90);

```

```

    analogWrite (motorPind, 0);
}

else if (data == '0')
{
    analogWrite (motorPina, 0);
    analogWrite (motorPinb, 0);
    analogWrite (motorPinc, 0);
    analogWrite (motorPind, 0);
}

else if (data == '5')
{
    analogWrite (motorPina, 220);
    analogWrite (motorPinb, 220);
    analogWrite (motorPinc, 220);
    analogWrite (motorPind, 220);
    //190
    delay(90);
    //delay(180);
    analogWrite (motorPina, 0);
    analogWrite (motorPinb, 0);
    analogWrite (motorPinc, 0);
    analogWrite (motorPind, 0);
}
else if (data == '9')
{
    analogWrite (motorPina, 170);
    analogWrite (motorPinb, 170);
    analogWrite (motorPinc, 170);
    analogWrite (motorPind, 170);
    //delay(15000);
    //analogWrite (motorPina, 0);
    //analogWrite (motorPinb, 0);
    //analogWrite (motorPinc, 0);
    //analogWrite (motorPind, 0);

}

else if (data == '6')
{

    digitalWrite(buff1, HIGH);
//delay(10);
    digitalWrite(buff1, LOW);
}

```

```

//delay(10);
//digitalWrite(buff1,HIGH);

}

else if (data == '7')
{
digitalWrite(buff2,HIGH);
delay(10);
digitalWrite(buff2, LOW);
//delay(10);
//digitalWrite(buff2, HIGH);
}
}

else if (data == '8')
//Look for falling edge on clk_decade
{
while(digitalRead(clk_decade)==LOW); //Wait until HIGH

waiting_for_negative_edge = true;

while(waiting_for_negative_edge)
{
if (digitalRead(clk_decade)==LOW )
{
Serial.println('1');
waiting_for_negative_edge = false;
}
}
}
}
}
}

```

Appendix B NDI Optotrak Certus Setup using Ethernet

i. Configure the host computer.

Configure the [Optotrak System] and [Ethernet Options] sections of the optotrak.ini file at c:\ndigital\settings as follows.

[OPTOTRAK System]

ResetTimeout= 500

Driver= etherlink.dll

[Ethernet Options]

Server Address=172.16.3.30

Control Port=9999

Data Port=10000

Reset Delay=10000

Data Timeout=10000

ii. Configure the SCU

- Connecting Tera Term to Optotrak SCU
 - Use a female DB9 to female DB9 cross linked/null modem cable (and usb to serial adapter if required) to connect host computer to Optotrak SCU
 - Open Tera Term
 - Select Serial option and select serial port connected to Optotrak SCU
 - Select Setup – Terminal from top tool bar
 - Under new line select CR+LF for Receive and Transmit
 - Select checkbox “Local Echo” and press okay
 - Press enter in command line

- Check current IP Address with Tera Term
 - Type eniprd with a space afterwards (there is always a space after a command) and press enter
 - Note: commands are extremely sensitive and must include space. If a correction is made to the command using backspace the command will not complete correctly and must be reentered with no corrections

- Disable DHCP with Tera Term
 - Type endhwr 0
 Ensmrd
- Check DHCP status with Tera Term
 - Type endhrd (space afterwards)
 - Returns example 01400 where the first 0 indicates DHCP disabled

- Write the desired IP address to Optotrak SCU with Tera Term
 - Type enipwr 192.168.101.2
 - Important: Power cycle Optotrak SCU after this step!

- Check current IP address with Tera Term to ensure the IP address has been written correctly
 - Type eniprd (space afterwards)
 - Returns 192.168.101.2(plus 4 numbers)

- Check current subnet mask with Tera Term and note it for later
 - Type ensmrd (space afterwards)
 - Returns 255.255.255.0 or 255.255.252.0 usually

iii. Configure the Computer network card

- Open network and sharing center on host computer

- Under local area connection options, uncheck internet protocol version 6
- Under local area connection options go to internet protocol version 4
- Set ip address manually to the same network but different ip address of the SCU
 - Example: 192.168.101.11
- Set subnet mask the same as SCU
 - Example: 255.255.255.0

iv. Configure First Principles

- Open First Principles
- Open settings – communications- ethernet
- Enter IP address of Optotrak SCU (192.168.101.2 in example)
- Query system to identify SCU connected

1. Network connection – through intermediate router, hub or switch

i. Connections

Use category 5 ethernet cable to connect the host computer and the SCU to the router.

ii. Configure the host computer.

Configure the [Optotrak System] and [Ethernet Options] sections of the optotrak.ini file at c:\ndigital\settings as follows.

```
[OPTOTRAK System]
ResetTimeout= 500
```

Driver= etherlink.dll

[Ethernet Options]

Server Address=172.16.3.30

Control Port=9999

Data Port=10000

Reset Delay=10000

Data Timeout=10000

iii. Configure the SCU using Tera Term

- Disable DHCP with Tera Term
 - Type endhwr 0
Ensmrd
- Check DHCP status with Tera Term
 - Type endhrd (space afterwards)
 - Returns example 01400 where the first 0 indicates DHCP disabled

iv. Configure the computer network card

- Open network and sharing center on host computer
 - Under local area connection options, uncheck internet protocol version 6
 - Under local area connection options go to internet protocol version 4
 - Check 'Obtain ip address manually'

v. First Principles configuration

- Open command prompt and run **ipconfig** command to verify the host ip address

- Run **arp -a** to get the ip address assigned to the SCU (You can get this from the router's default router settings webpage as well)
- Open First Principles
- Open settings – communications- ethernet
- Enter IP address of Optotrak SCU (192.168.101.2 in example)
- Query system to identify SCU connected

Appendix C SWAT Analysis of Technologies Commercially available to Visually Impaired People.

The following SWAT analysis (strength, weakness, opportunities and threats analysis) was conducted on commercially available technologies by Janet Canady.

Competitive Product	Product/ Technology	Price	Strengths	Weaknesses
TapTapSee	App that detects and announces objects	\$0, in app purchases	Free, good for objects	Must know location of object to use
Ultrasonic canes	Cane providing haptic feedback	\$ 816.69	Aerial and ground obstacle prevention, don't need to touch obstacles, "hall pass"	expensive, heavy, must lift cane, must sweep side to side, hand strain, uncomfortable, weather
BuzzClip	Buzzer that detects obstacles in ones path	\$ 249.00	Can use multiple, small, lightweight	only detects items directly in front of it, more expensive
BlindSquare.com	Foursquare app for blind people	\$ 40.00	Announces intersections	Need headset or have to touch screen, relies on FourSquare
Or-cam MyEye	Camera attached to eye that reads text aloud	\$ 2,500.00	Can also recognize currency and faces, small, lightweight, doesn't require smart phone	Has two separate units, uses headphones, limited to 150 products, expensive
Argus II Retinal Prosthesis	Bionic eye	\$150,000.00	Makes shades of lights visible	Requires invasive surgery, still requires external equipment
NavCog3	Smartphone based navigation	Free	Turn by turn navigation	Requires smart phone with barometer
EyeSee	Eyetracking and facial tracking software	Not available	Can map eye movement and facial expressions	No real time feedback
Ford "Feel the View"	A Braille-like device for car windows	N/A	Fun, good for all ages	No real time feedback, must learn new haptic language
Opticon	Barcode scanner system	N/A	Phone application, easy barcode generation, can label nearly anything	Information must be preinput, not real time

Created by Janet Canady

Figure 25 SWAT Analysis

Bibliography

1. "Visual impairment, blindness cases in U.S. expected to double by 2050", *National Institutes of Health (NIH)*, 2019. [Online]. Available: <https://www.nih.gov/news-events/news-releases/visual-impairment-blindness-cases-us-expected-double-2050>. [Accessed: 26- Mar- 2019].
2. Varma, R et al, "Visual impairment and blindness in adults in the United States: Demographic and Geographic Variations from 2015 to 2050," *JAMA Ophthalmology*, DOI:10.1001/jamaophthalmol.2016.1284.
3. S. Ertan, C. Lee, A. Willets, H. Tan, and A. Pentland, "A wearable haptic navigation guidance system," *Digest of Papers. Second International Symposium on Wearable Computers (Cat. No.98EX215)*, 1988.
4. K. Tsukada and M. Yasumura, "ActiveBelt: Belt-Type Wearable Tactile Display for Directional Navigation," *UbiComp 2004: Ubiquitous Computing Lecture Notes in Computer Science*, pp. 384–399, 2004.
5. W. Heuten, N. Henze, S. Boll, and M. Pielot, "Tactile wayfinder," *Proceedings of the 5th Nordic conference on Human-computer interaction building bridges - NordiCHI 08*, 2008.
6. J. R. Marston, J. M. Loomis, R. L. Klatzky, and R. G. Golledge, "Nonvisual Route following with Guidance from a Simple Haptic or Auditory Display," *Journal of Visual Impairment & Blindness*, vol. 101, no. 4, pp. 203–211, 2007.
7. G. Flores, S. Kurniawan, R. Manduchi, E. Martinson, L. M. Morales, and E. A. Sisbot, "Vibrotactile Guidance for Wayfinding of Blind Walkers," *IEEE Transactions on Haptics*, vol. 8, no. 3, pp. 306–317, 2015.
8. F. Gemperle, N. Ota, and D. Siewiorek, "Design of a wearable tactile display," *Proceedings Fifth International Symposium on Wearable Computers*, 2001.
9. P. Bach-y-Rita, K. A. Kaczmarek, M. E. Tyler, and J. Garcia-Lara, "Form perception with a 49-point electrotactile stimulus array on the tongue: A technical note," in *Journal of Rehabilitation Research and Development* , vol. 35, No. 4, 1998, pp. 427–430.
10. M. Ptito, "Cross-modal plasticity revealed by electrotactile stimulation of the tongue in the congenitally blind," *Brain*, vol. 128, no. 3, pp. 606–614, 2005.
11. P. Bach-y-Rita, K. A. Kaczmarek, M. E. Tyler, and J. Garcia-Lara, "Form perception with a 49-point electrotactile stimulus array on the tongue: A technical note," in *Journal of Rehabilitation Research and Development* , vol. 35, No. 4, 1998, pp. 427–430.

12. “The vOICe - New Frontiers in Sensory Substitution,” *The vOICe - New Frontiers in Sensory Substitution*. [Online]. Available: <http://www.seeingwithsound.com/>. [Accessed: 14-Mar-2019].
13. R. Velazquez, E. Fontaine, and E. Pissaloux, “Coding the Environment in Tactile Maps for Real-Time Guidance of the Visually Impaired,” *2006 IEEE International Symposium on MicroNanoMechanical and Human Science*, 2006.
14. Seeing AI Project — Pivothead Wearable Imaging", *Pivothead Wearable Imaging*, 2019. [Online]. Available: <http://www.pivothead.com/seeingai/>. [Accessed: 13- Mar- 2019].
15. G. Ng, P. Barralon, G. Dumont, S. Schwarz, and J. M. Ansermino, “Optimizing the Tactile Display of Physiological Information: Vibro-Tactile vs. Electro-Tactile Stimulation, and Forearm or Wrist Location,” *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2007.
16. K. Myles and M. S. Binseel, *The Tactile Modality: A Review of Tactile Sensitivity and Human Tactile Interfaces*. United States: Army research lab aberdeen proving ground md human research and engineering directorate, 2007.
17. Weinstein and Sidney, *Intensive and extensive aspects of tactile sensitivity as a function of body part, sex, and laterality*. 1968. The skin senses Proceedings of the First International Symposium March, 1966 Tallahassee, Fla 195-222
18. Y. J. Liu, X. R. Wu, J. S. Leng, R. Davidson, and P. Taylor, “Tactile Display Based on Controllable Fluid Actuator,” *Materials Science Forum Progress in Light Metals, Aerospace Materials and Superconductors*, pp. 1669–1672, 2007.
19. G. Ng, P. Barralon, G. Dumont, S. Schwarz, and J. M. Ansermino, “Optimizing the Tactile Display of Physiological Information: Vibro-Tactile vs. Electro-Tactile Stimulation, and Forearm or Wrist Location,” *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2007.
20. R. Velázquez, “Wearable Assistive Devices for the Blind,” *Wearable and Autonomous Biomedical Devices and Systems for Smart Environment Lecture Notes in Electrical Engineering*, pp. 331–349, 2010.
21. Z. Yu, S. Horvath, A. Delazio, J. Wang, R. Almasi, R. Klatzky, J. Galeotti, G. Stetten,”PalmSight: an assistive technology helping the blind to locate and grasp objects” *Technical Report, Robotics Institute, CMU*, December, 2016
22. S. Lederman and R. Klatzky, "Haptic perception: A tutorial", *Attention, Perception & Psychophysics*, vol. 71, no. 7, pp. 1439-1459, 2009. Available: 10.3758/app.71.7.1439
23. “Optotrak Certus User Guide,” *manualzz.com*. [Online]. Available: <https://manualzz.com/doc/28945333/optotrak-certus-user-guide>. [Accessed: 14-Mar-2019].

24. “ARMAX Model Definitions (System Identification Toolkit),” *ARMAX Model Definitions (System Identification Toolkit) - LabVIEW 2013 System Identification Toolkit Help - National Instruments*. [Online]. Available: <http://zone.ni.com/reference/en-XX/help/372458D-01/lvsysidconcepts/modeldefinitionsarmax/>. [Accessed: 23-Mar-2019].
25. M. Mignolet and J. Red-Horse, “ARMAX identification of vibrating structures - Model and model order estimation,” *35th Structures, Structural Dynamics, and Materials Conference*, 1994.
26. E. H. Fung, Y. Wong, H. Ho, and M. P. Mignolet, “Modelling and prediction of machining errors using ARMAX and NARMAX structures,” *Applied Mathematical Modelling*, vol. 27, no. 8, pp. 611–627, 2003.
27. “data,” *Estimate parameters of ARMAX model using time-domain data - MATLAB*. [Online]. Available: <https://www.mathworks.com/help/ident/ref/armax.html#btdf35s-9>. [Accessed: 24-Mar-2019].
28. Kuper, H., et. al. “Does Cataract Surgery Alleviate Poverty? Evidence from a Multi-Centre Intervention Study Conducted in Keyna, the Philippines and Bangladesh.” *PloS Medicine*. 5.11 (2010).