The Design, Development, and Analysis of a Wearable, Multi-modal Information Presentation Device to Aid Astronauts in Obstacle Avoidance During Surface Exploration

by

Alison Eve Gibson

Submitted to the Department of Aeronautics and Astronautics in partial fulfillment of the requirements for the degree of Master of Science in Aeronautics and Astronautics at the MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2017

© Massachusetts Institute of Technology 2017. All rights reserved.
The Design, Development, and Analysis of a Wearable, Multi-modal Information Presentation Device to Aid Astronauts in Obstacle Avoidance During Surface Exploration

by

Alison Eve Gibson

Submitted to the Department of Aeronautics and Astronautics on May 1, 2017, in partial fulfillment of the requirements for the degree of Master of Science in Aeronautics and Astronautics

Abstract

The future of human space exploration will involve extra-vehicular activities (EVA) on foreign planetary surfaces (i.e. Mars), an activity that will have significantly different characteristics than exploration scenarios on Earth. These activities become challenging due to restricted vision and limitations placed on sensory feedback from altered gravity and the space suit. The use of a bulky, pressurized EVA suit perceptually disconnects human explorers from the hostile environment, increasing navigation workload and risk of collision associated with traversing through unfamiliar terrain. Due to the hazardous nature of this work, there is a critical need to design interfaces for optimizing task performance and minimizing risks; in particular, an information presentation device that can aid in obstacle avoidance during surface exploration and way-finding. Multi-modal displays are being considered as cues to multiple sensory modalities enhance cognitive processing through taking advantage of multiple sensory resources, and are believed to communicate risk more efficiently than unimodal cues. This thesis presents a wearable multi-modal interface system to examine human performance when visual, vibratory, and visual-vibratory cues are provided to aid in ground obstacle avoidance. The wearable system applies vibrotactile cues to the feet and visual cues through augmented reality glasses to convey obstacle location and proximity. An analysis of obstacle avoidance performance with the multi-modal device was performed with human subjects in a motion capture space. Metrics included completion time, subjective workload, head-down time, collisions, as well as gait parameters. The primary measures of performance were collision frequency and head-down time, as these both must be minimized in an operational environment. Results indicate that information displays enhance task performance, with the visual-only display promoting the least head-down time over tactile-only or visual-tactile displays. Head-down time was the highest for trials without a display. Results provide implications for presenting information during physically active tasks such as suited obstacle avoidance.

Thesis Supervisor: Leia Stirling
Title: Assistant Professor of Aeronautics and Astronautics
Acknowledgments

This work was supported through graduate research fellowships from the National Science Foundation and Draper. The author would like to thank her research advisors Dr. Leia Stirling (MIT) and Dr. Andrea Webb (Draper) for their help and guidance throughout the project. The author would also like to thank and recognize MIT undergraduate students Bradley Jomard and Charlotte Sun for their assistance in data collection and post-processing.
Contents

1 Introduction
   1.1 Future Manned Missions to Mars ...................................... 15
   1.2 Multi-Modal Information Presentation ............................. 17
       1.2.1 Vibrotactile Displays ........................................ 19
   1.3 Human Obstacle Avoidance Strategy ................................. 22
   1.4 Project Objectives and Hypotheses ................................. 23

2 Examination of Vibrotactile Perception Thresholds of the Feet 27
   2.1 Experimental Methods ............................................... 28
       2.1.1 Participants .................................................. 28
       2.1.2 Materials ...................................................... 28
       2.1.3 Experimental Protocol ...................................... 29
       2.1.4 Statistical Methods ......................................... 31
   2.2 Results ............................................................... 32
       2.2.1 Perceived Location Accuracy ................................. 32
       2.2.2 Perceived Type Accuracy .................................... 34
       2.2.3 Undetected Vibrations ....................................... 36
   2.3 Discussion ............................................................ 37
   2.4 Implications for Vibrotactile Displays ............................ 40

3 Design of the Multi-modal Interface Device .......................... 43
   3.1 Display Cue Logic .................................................... 44
   3.2 Hardware Design and Development ................................ 45
List of Figures

2-1 Fritzing diagram of haptic display and electrical connections \( I^2C \) SCL (green), \( I^2C \) SDA (yellow), power (red), ground (black), and digital outputs (pink) ........................................... 30

2-2 Example participant wearing the experimental device and operating the GUI .................................................. 31

2-3 Average perceived Location Accuracy and standard error for each foot and attention state (pooled across Type and Order) 33

2-4 Perceived vibration Type Accuracy mean and standard error for each Location, Type, and Foot (pooled across Attention) 35

2-5 Percentage of undetected vibrations for each subject and vibration Location ..................................................... 37

3-1 Display cue characteristics for each level and modality .......... 44

3-2 The vibrotactile boot base assembly (left) and CAD models of the 3D-printed front base (top right) and back base (bottom right) ................................................................. 46

3-3 The vibrotactile boot and integrated electronics .................... 47

3-4 Fritzing diagram of electrical connections for the vibrotactile boot, showing connections for \( I^2C \) SCL (green), \( I^2C \) SDA (yellow), power (red), ground (black), digital outputs (pink), analog input (blue), and serial RX/TX (grey). Connectors are not shown. ......................................................... 48

3-5 The vibrotactile boot algorithm flowchart .......................... 49
3-6 A participant wearing the augmented Epson Moverio BT-200 Smart Glasses with attached shields

4-1 The obstacle blocks with retroreflective markers

4-2 An example user wearing the multi-modal display (left) and the motion capture layout (right) with obstacle (OB) and associated starting point locations for experimental trials (1-3) and training trials (T1-T4). Each square in the grid is approximately 23x23 inches.

4-3 The initial head-level state $H_0$ for each participant based on initial position marker readings (left) and the geometry showing the head-down angle measurement $H$ (right)

4-4 NASA TLX Overall Workload scores for all sixteen subjects and display conditions

4-5 NASA TLX Workload averages for all workload types and display conditions

4-6 NASA TLX Workload scores for mental and temporal workload types, where shapes denote significance groupings. Black circles represent overall mean and standard deviation while grey circles represent individual participant means.

4-7 Post-Experiment Survey workload rankings (Scale of 1-5 increasing with workload), where shapes denote significance groupings. Black circles represent overall mean and standard deviation while grey circles represent individual participant means.

4-8 Histogram data of participant responses to display preferences
4-9 Normalized Completion Time for all display conditions, where shapes denote significance groupings. Black circles represent overall mean and standard deviation while grey circles represent individual participant means (pooled across Order, Location, Size, and Sex). ................................. 67

4-10 Head-Down Percentage of Trial for all display conditions, where shapes denote significance groupings. Black circles represent overall mean and standard deviation while grey circles represent individual participant means (pooled across Order, Location, Size, and Sex). ................................. 68

4-11 Toe Clearance and Toe-Off Distance for all display conditions, where shapes denote significance groupings. Black circles represent overall mean and standard deviation while grey circles represent individual participant means (pooled across Order, Location, Size, and Sex). ................................. 70

4-12 Step characteristics for all display conditions, where shapes denote significance groupings. Black circles represent overall mean and standard deviation while grey circles represent individual participant means (pooled across Order, Location, Size, and Sex). ................................. 72

A-1 Normalized Completion Time for all sixteen subjects and display conditions ........................................... 96

A-2 Head-Down Percentage of Trial for all sixteen subjects and display conditions ........................................... 96
## List of Tables

2.1 Non-parametric tests used to determine effects of independent variables on non-spherical dependent variables ........................................ 32

2.2 Statistics from a Friedman test on the effect of Vibration Location on Perceived Vibration Location Accuracy (pooled across Attention, Type, Order, and Foot) ........................................ 33

2.3 Statistics from Wilcoxon Signed Rank and Rank Sum tests on the effects of Attention and Order on Perceived Vibration Type Accuracy (pooled across Location, Type, and Foot) ........................................ 34

2.4 Statistics from a Friedman test on the effect of Vibration Type on Perceived Vibration Type Accuracy (pooled across Attention, Location, Order, and Foot) ........................................ 35

2.5 Statistics from a Friedman test on the interaction of Foot*Location on Perceived Vibration Type Accuracy (pooled across Attention, Type, and Order) ........................................ 36

2.6 Percentages of undetected vibrations across all trials and participants. All dashed cells and cells not shown have zero undetected responses for those trials. ........................................ 36

4.1 Non-parametric tests used to determine effects of select independent variables on non-spherical motion capture metrics ........................... 60

4.2 Statistics from Friedman tests on the effect of display (d.f.=3) on NASA TLX Workload Scores, where asterisks denote significant p-values 63

4.3 Statistics from a Friedman test on Survey Workload scores .............. 63
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.4</td>
<td>Statistical results from a Friedman test on display preference rankings</td>
<td>64</td>
</tr>
<tr>
<td>4.5</td>
<td>Significant effects on Normalized Completion Time</td>
<td>66</td>
</tr>
<tr>
<td>4.6</td>
<td>Significant effects on Head-Down Percentage of Trial</td>
<td>68</td>
</tr>
<tr>
<td>4.7</td>
<td>Statistics from a Friedman test on the effect of Display on Toe Clearance</td>
<td>69</td>
</tr>
<tr>
<td>4.8</td>
<td>Statistics from a Friedman test on the effect of Display on Toe-Off Distance</td>
<td>70</td>
</tr>
<tr>
<td>4.9</td>
<td>Statistics from a Friedman test on the effect of Display on Number of Steps</td>
<td>71</td>
</tr>
<tr>
<td>4.10</td>
<td>Statistics from a Friedman test on the effect of Display on Normalized Average Step Length</td>
<td>71</td>
</tr>
<tr>
<td>4.11</td>
<td>Statistics from a Friedman test on the effect of Display on Normalized Step Length Variance</td>
<td>71</td>
</tr>
<tr>
<td>4.12</td>
<td>Statistics from Wilcoxon Rank Sum tests on the effect of Order on step characteristics</td>
<td>73</td>
</tr>
<tr>
<td>4.13</td>
<td>Statistics from Wilcoxon Signed Rank tests on the effect of Size on step characteristics</td>
<td>73</td>
</tr>
<tr>
<td>4.14</td>
<td>Percentage of trials with a leading right foot for each Display and Order level</td>
<td>74</td>
</tr>
<tr>
<td>4.15</td>
<td>Contingency Table for obstacle Size and Collision frequency with (expected cell total) and [chi-square statistic] for each cell</td>
<td>74</td>
</tr>
<tr>
<td>A.1</td>
<td>Contingency table containing frequency data for lead foot between TO and VT displays</td>
<td>93</td>
</tr>
<tr>
<td>A.2</td>
<td>Contingency table containing frequency data for lead foot between TO and NC displays</td>
<td>93</td>
</tr>
<tr>
<td>A.3</td>
<td>Repeated Measures ANOVA table for Normalized Completion Time, where asterisks denote significant p-values</td>
<td>94</td>
</tr>
<tr>
<td>A.4</td>
<td>Repeated Measures ANOVA table for Head-Down Percentage of Trial, where asterisks denote significant p-values</td>
<td>95</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Future Manned Missions to Mars

Future manned missions to Mars and the Moon are projected to involve space and surface operations that impose much higher risk and workload on astronauts than similar activities on Earth; the most complex of these operations involve Extra-Vehicular Activity (EVA), which occurs when astronauts exit the protected environment of the spacecraft and enter the vacuum of space or the thin atmosphere of another planet while wearing a bulky, pressurized spacesuit [1]. These activities are challenging due to the perceptual restrictions placed on somatosensory and proprioceptive feedback from altered gravity and the pressurized suit, visual restrictions from the helmet, and the absence of auditory information from the environment [2]. Somatosensory feedback involves physical sensations such as pressure, pain, or temperature, and proprioceptive feedback involves the kinaesthetic sense of limb position, orientation, and movement. Surface EVA operations on Mars will include assembly and construction of structures, geologic exploration, and protective shelter excavation [1]. These activities come with serious risk of injury or damage to imperative life-support equipment since trips and falls are likely to occur on unfamiliar, rocky terrain.

As if the task of locomoting while wearing a bulky spacesuit in a different gravito-inertial environment is not already difficult for a healthy person, the six to twelve month trip to Mars could cause physiological changes to astronauts that further in-
crease task difficulty. Long-duration spaceflight has various detrimental effects on the human body, including musculoskeletal deconditioning [3, 4, 5] and sensorimotor system alterations that affect locomotor function [6, 7, 8, 9]. These studies suggest that changes in locomotion post-flight include impaired multi-joint coordination, increased sway, decreased postural control, decreased head-trunk coordination, decreased agility, and impaired neuromuscular activation of lower limbs. Additionally, studies show that sensorimotor adaptation to microgravity environments and the resultant visual-vestibular reweighting alters static and dynamic balance stability during locomotion [9, 8, 10, 11]. Peters et al. [12] suggests that long-duration spaceflight modifies dynamic visual acuity and gaze stabilization due to alterations in the vestibular-ocular reflex (VOR). Additionally, researchers have found that head-trunk coordination is modified during walking and obstacle avoidance post-flight [13] and there is an increased risk of tripping due to changes in gait (i.e. less toe clearance) after returning to Earth [14]. Since astronauts’s bodies will likely be affected in these ways while in transit to Mars, the task of planetary exploration after landing on Mars becomes even more precarious.

Falls on the moon throughout the Apollo missions were common due to the altered human locomotion from the pressurized suit and reduced gravity. There is approximately twice as much gravity on Mars compared to the moon and the Martian terrain is far less forgiving. An unanticipated fall could puncture or damage critical life-support equipment, risking astronaut safety and mission timelines. Due to the physically demanding nature of suited surface exploration in combination with restricted time, resources, and perceptual capabilities of crewmembers, a critical need has been identified to design interfaces for supporting task performance and minimizing hazards during operations [1]. The information necessary to efficiently carry out a surface EVA extends far beyond what can be directly seen and felt through the suit, so presenting useful and relevant information to astronauts via displays could improve terrain traversal performance and safety. Of particular interest is the development of a multi-modal information presentation device to aid in the task of obstacle avoidance during surface exploration and way-finding. The term multi-modal is used
to describe information that is displayed over multiple input modalities, typically considering visual, auditory, and/or tactile displays. To assist astronauts in safely traversing to another crewmember, station, or rover while navigating through rugged terrain, multi-modal interfaces must be assessed to understand how the presentation of information affects cognitive function and operational performance [15].

1.2 Multi-Modal Information Presentation

One of the major challenges in the field of Human Factors is in understanding how to present critical information in the most efficient way possible while minimizing workload and distraction. An increasingly popular method for improving information delivery is the use of multisensory devices [16]. These devices present information to the user via multiple sensory modalities in order to support quick and robust interpretation through sensory reinforcement. The idea of presenting information to multiple modalities to improve task performance and information processing was first presented by Navon and Gopher in 1979 [17]. Research on this topic provoked the development of Wickens’ Multiple Resource Theory (MRT) [18], which asserts that there are multiple pools of attentional resources that are drawn from to perform a task, including resources allocated to sensory input stages, cognitive processing, and response selection. MRT asserts an existence of dissimilar cognitive resources for processing sensory information from different modalities, implying that cognitive workload may be reduced through distributing information across different sensory modalities. Van Erp et al. [19] suggests that this distribution of information across other modalities offloads the often overworked visual channel, reducing cognitive processing efforts required for effective task performance. Rather than consider the perspective of interference within the processing stages due to information coming from various input modalities, multi-modal displays follow MRT through considering the enhancement of processing due to synergistic information coming from the different attentional input resources. As a result, the hypothesis is that multi-modal information presentation could induce more robust information processing and improve
task performance, particularly in circumstances when the primary modality (usually vision) is overloaded.

Support for the benefits of multi-modal information presentation was first shown in several studies that distributed concurrent tasks across both the visual and auditory modalities, showing performance benefits for presenting redundant information as well as complementary information [18]. These results have made visual-audial displays a popular choice for decades. However, studies have more recently shown performance benefits of adding tactile cues to visual cues, a combination that is shown to result in faster reaction times than visual cues alone in simulation experiments of operator performance, decision-making, and alert recognition [20, 21, 22, 23]. Three meta-analyses of multi-modal cues for task interruption (i.e. combinations of auditory, visual, and tactile stimuli) showed faster response time and improved response accuracy for displays involving tactile warnings compared to those with auditory warnings [24]. While this makes tactile cues an appropriate choice for alarms and alerts, the addition of tactile cues to visual cues has also been shown to enhance operator performance in studies simulating more complex tasks such as communications, driving, land navigation, teleoperation in virtual reality environments, and spatial orientation in flight [16]. An important factor in increasing multi-sensory integration is spatio-temporal and/or semantic congruency, referring to display logic where redundant information is given across the different sensory channels. A study utilizing a tactile belt and visual display to present five standard military arm and hand signals demonstrated slower response times when the visual-tactile cues were incongruent (i.e. in different directions or mismatching information) [25]. A review of multiple studies utilizing visual-tactile cues to improve cognitive performance across various domains also suggested support for the benefit of information congruency, as redundant cues improved response speed and accuracy while conflicting cues were distracting to users and induced response error [26].

Multi-modal alerts and threat warnings are also shown to enhance perception of risk more efficiently than unimodal cues. One study [27] used a tactile torso belt to apply unique vibrations corresponding to the direction (i.e. front, back, right, or left)
of a threat during combat simulation; results showed that response time was reduced and threat detection precision increased when either the tactile cues or 3D audio cues were presented along with visual cues. Another study examined unimodal, bi-modal, and tri-modal alerts incorporating vibrotactile, audial, and visual signals, and found that bi-modal and tri-modal cues were perceived as more urgent than unimodal cues [28]. A review of displays utilizing multi-modal alerts and warnings also found support for the notion that multi-modal cues are more effective in conveying risk [19].

1.2.1 Vibrotactile Displays

The auditory system and associated cognitive pathways have a sophisticated ability in translating sound sensation to meaning (e.g. language) and learning new information mappings. However, the use of tactile channels for information mapping has not been characterized as thoroughly, and could be of great use in intuitively conveying alerts about surface features, inclination, and obstacles in a path. Other advantages of tactile cues include its omnidirectionality (i.e. signal salience regardless of spatial orientation and attentional load), ability to be perceived simultaneously with visual and auditory signals, and the small number of competing demands for this resource [23]. The auditory channel has the potential to be highly loaded during EVA operations due to radio communication protocols, so the addition of tactile cues may be a more appropriate design choice in this operational scenario. Elliot et al. [16] points out that many studies have demonstrated tactile cue success for waypoint land navigation, especially during low visibility conditions or when attention was focused on surrounding ground terrain. It has been hypothesized that tactile information can complement or replace other senses when interference is present in the primary modality (e.g. glares on visual displays or limited range of view) [26]. Additionally, studies utilizing a tactile display on the torso for direction commands and spatial orientation assistance have demonstrated faster reaction times, improved situation awareness, and increased stability during navigation tasks that burden the visual channel compared to using visual cues alone [29].

To ensure that information is easily and quickly recognized, it is recommended
that the placement of tactors correspond with task demands [16]. As a result, it's hypothesized that tactile presentation to the front of the feet may be most intuitive and task-relevant during surface obstacle avoidance. Cutaneous afferent inputs from the feet during walking are naturally involved in reflexes that alter swing limb trajectory to avoid stumbling and falling [30], but this feedback is mostly unavailable to the perceptually impoverished astronaut during an EVA. Since vibrotactile displays on the feet are uncommon, there is a need to better understand applications of tactile cues to the feet individually and in combination with other modalities. Vibrotactile displays for alerts, bodypart orientation, and directional navigation tasks have been used to guide successful task performance on the arms [31, 32, 33, 34, 35, 36], shoulder [37], and waist [38, 39, 40, 41, 42]; the tactile cues in these studies ranged from single vibratory alerts to multiple location and vibration patterns, conveying information such as navigation direction (up to four directions), position targeting, trajectory following, forearm rotation, arm/hand gestures, and joint angle errors for body orientation corrections. In all cases, no more than 5-10 unique vibratory signals were used. One of the studies used a vibrotactile bracelet to guide forearm orientation, and found higher accuracy in visuotactile-guided trajectories compared to visually-guided trajectories [32].

While the arms, shoulder, and waist have shown successful vibrotactile detection and comprehension for various tasks, there have been very few applications specifically utilizing the feet. One study successfully demonstrated the ability of a sandal-like vibration interface to promote and maintain a specific walking pace through mapping vibration pulse rate to a predetermined step rate [43]. Another study used vibrating toe rings to signal direction changes while walking towards a preset destination, where the toe ring on the right or left foot would vibrate to indicate a turn in the corresponding direction [44]. Other navigation studies have used vibrotactile signals to command direction changes (right/left) or inform approximate distances to a destination, but none have tried to convey detailed information about small obstacles directly in one's walking path, which may require increased information presentation on this sensory channel. The task of active obstacle avoidance presents different chal-
lenges than those in simple navigation since different information would be needed; task-relevant information may include the distance to obstacle(s), location of obstacle(s), and size of obstacle(s). Additionally, different information would be required for different strategies (e.g. stepping over an obstacle versus walking around it). To narrow the scope of this problem, the current body of work focuses on information display for the strategy of approaching and stepping over an obstacle. Stepping over an object is believed to be more challenging than walking around an object, as successful obstacle avoidance requires real-time visual recognition of foot placement in the approach phase [45].

Since advanced haptic technologies have emerged only in the last decade and their applications in mapping information to human feet in particular are not well studied, it isn’t clear what kind of information mapping characteristics (i.e. vibration frequencies, patterns, or locations) would be best to support robust cue salience. Factors that affect vibration perception include signal features (e.g. frequency, duration, and temporal/spatial distance) as well as individual factors such as gender and age [46]. Regarding frequency, a feature that determines intensity, vibrations ranging from 50 to 300 Hz have been found to be the most detectable on the skin [47]. Although, most vibration perception research has examined detection thresholds for the fingers, hands, and torso, while little research has been done specifically with the feet. Some studies have assessed vibration perception at the sole of the foot [48, 49, 50] and top of the foot [46, 51], supporting detection capabilities within the 20 to 250 Hz range. These findings support the notion that the sensory systems of the feet enable sufficient vibrotactile perception, although research has not examined perception thresholds at the locations of interest for the multi-modal device (i.e. the front, back, lateral, and medial sides of the foot).

Before vibrotactile signals can be implemented in a multi-modal navigational aid, it is necessary to first understand the tactile perception capabilities of the feet for the locations and signal types under consideration, in particular during states of divided attention. The detection and comprehension of the vibrotactile signals must be robust enough to withstand various cognitive and perceptual loads since the practical use
of such an interface would undoubtedly occur while the user is multi-tasking. Load theory [52, 53, 54] suggests that perceptual and cognitive demands (or loads) each have a limited capacity beyond which selective attention can fail, negatively affecting sensory perception and cognitive performance. Fitousi and Wenger [55] emphasize that most modern research attributes failures in attention and task performance to overloaded sensory channels and/or overloaded cognitive resources. A reduction in perceptual and attentional capacity can also take place during excessive workload; this phenomenon is referred to as attentional narrowing, and is characterized by an involuntary reduction in the range of cues able to be perceived by an individual [56]. In the case of an astronaut under high cognitive and visual load from multi-tasking while navigating unknown terrain, it is therefore imperative that the vibrotactile stimuli carrying critical information be salient enough to enable adequate perception and comprehension during narrowed attentional focus. This need for cue salience also justifies the consideration of a multi-modal interface, as inputs to multiple sensory modalities increase the likelihood of cue perception and comprehension; if one modality is overloaded and perceptually narrowed, other modalities are still able to receive the information.

1.3 Human Obstacle Avoidance Strategy

While performing an EVA, astronauts wear a helmet that occludes regions of the field of view. Obstacle avoidance studies have previously shown that when the lower visual field is restricted with basketball goggles and obstacle position cues in the environment are provided (i.e. vertical poles next to the obstacle), foot placement before the obstacle returns to the optimal values seen for full vision conditions [57, 58]. This phenomenon suggests that visual information of obstacle location relative to the self compensated for the loss of visual information about lower limb movement. Several obstacle avoidance studies examining the visual control of foot placement have shown that people need these cues about an approaching obstacle within at least two steps prior to the obstacle in order to maintain balance and avoid collisions. Some
of these studies used an eye tracker mounted on the head to determine gaze fixation throughout obstacle avoidance tasks, and found that people fixate their gaze two steps ahead on average [59, 60]. Other studies demonstrated that removing obstacle information within two steps distance did not affect foot placement or strategy [61, 45, 62, 63]. Obstacle avoidance studies have also confirmed that high failure rates in avoiding obstacles are not caused by inappropriate limb elevation, but incorrect foot placement before the obstacle [45, 64]. While general target location is important during approach, research by Rietdyk et al. [58] demonstrated that for a single-foot targeting task, the foot-relative-to-target information was more important than the absolute target-location in the environment. These results can provide guidance on the development of multi-modal displays for obstacle avoidance and suggest that cues provided at least two steps prior to an oncoming obstacle would aid participants in stepping over obstacles while minimizing head down time. It is expected that surface mission operations will involve tasks that require one to keep most of their attention on features ahead (e.g. following a rover, walking to another crewmember or structure, etc.), so it would be beneficial if one could do so without having to continuously divert their focus to the ground to avoid tripping over unanticipated objects. As a result, it was determined that obstacle proximity to the front of the foot would be the most relevant information to present during obstacle avoidance tasks with limited lower vision.

1.4 Project Objectives and Hypotheses

The primary objective of the current work was to examine the effectiveness of a visual-tactile display for a surface obstacle avoidance scenario similar to what astronauts may encounter during future missions to Mars. This was accomplished through experimentally evaluating the differences in obstacle avoidance performance and workload during use of unimodal information displays (visual only and tactile only), a multi-modal information display (visual-tactile combination), and no display. In order to carry out the experiments, a multi-modal information presentation device capable of
providing meaningful input to both the visual and tactile modalities was designed and developed. Since vibrotactile displays for the feet are uncommon, a preliminary study evaluating the vibrotactile perception capabilities of the feet was performed in order to guide adequate design of the tactile information mapping for the device. The body of work in this thesis presents results from the vibrotactile perception study (Chapter 2), the design and development of the multi-modal interface device (Chapter 3), and results from an analysis of obstacle avoidance performance across different information display conditions (Chapter 4).

Aside from obstacle collision frequency, the most operationally relevant measure of performance for this study was head-down time, where less head-down time was associated with better task performance, as the task priorities consisted of 1) avoiding contact with an obstacle and 2) reducing head-down time. These priorities were relevant to the operational scenario of interest (a suited astronaut navigating rocky terrain on Mars) since astronauts will need to keep their attention and focus on completing EVA tasks as opposed to keeping their focus on the ground. For the vibration perception study, the hypotheses were that attention could alter perception accuracy, and that certain vibration locations and signal properties may enable better perception accuracy than others. For the obstacle avoidance study with the multi-modal device, the main hypothesis was that real-time obstacle proximity information could enable visually-restricted individuals to avoid obstacles without continuously looking down at the ground. The presentation of cues beyond two steps distance to an obstacle and during approach could alter the gait accordingly and guide optimal foot placement before step-over. It was also hypothesized that the use of multi-modal cues (i.e. visual-tactile) might result in better obstacle avoidance performance compared to unimodal (i.e. tactile only or visual only) cues.

**Experiment 1: Vibrotactile Perception of the Feet**

- $H1_A$: Attentional loads may negatively affect vibration perception accuracy

- $H1_B$: Specific locations on the foot may promote better vibration perception accuracy than others
• $H_{1C}$: Specific types of vibration signals may be more perceivable than others

Experiment 2: Analysis of the Multi-modal Information Device on Obstacle Avoidance Performance

• $H_{2A}$: Real-time obstacle location and proximity cues can decrease head-down time during obstacle avoidance

• $H_{2B}$: Multi-modal cues may decrease head-down time more than unimodal cues during obstacle avoidance

• $H_{2C}$: Real-time obstacle location and proximity cues may alter the gait strategy during obstacle avoidance
Chapter 2

Examination of Vibrotactile Perception Thresholds of the Feet

Before the vibrotactile display could be implemented on novel locations of the foot, it was necessary to first understand the tactile perception capabilities of the feet for the locations and signal types under consideration, in particular during states of divided attention. The detection of the perceived tactile signal must be robust enough to withstand various cognitive and perceptual loads since the practical use of such an interface would undoubtedly occur while the user is multi-tasking. In order to better understand how to incorporate tactile signals for robust signal detection in the multimodal interface device, this study examined four types of vibrations at six different locations per foot under varying attention loads. Independent variables consisted of vibration signal Type (High, Low, Increase, Decrease), Location (1-6), Attention state (Focused or Distracted), Foot (Right or Left) and Order of attention condition assignment. The dependent variables were perceived Location Accuracy (ability to detect vibration at a specific location), and perceived vibration Type Accuracy (ability to detect the type of vibration signal). It was hypothesized that attentional load would negatively affect perception accuracy due to attentional narrowing, and that certain locations and vibration types may be more detectable than others.
2.1 Experimental Methods

2.1.1 Participants

The participants consisted of ten healthy adults (3 females, 7 males) between the ages of 19 and 27 (M=23.3, SD=2.4). The experimental protocol was approved by the MIT Committee on the Use of Humans as Experimental Subjects (COUHES) and all participants provided written consent. Participants were excluded from the study if they reported irregularity or abnormalities with tactile perception on the feet or any injuries to the lower extremities. All participants were compensated with $20 USD. Participants were mostly right-handed (8 of 10), while one was left-handed and one was ambidextrous.

2.1.2 Materials

A custom haptic display was developed that applied four kinds of vibrations at six locations on each foot: one on the tip of the big toe, three on the lateral side, one on the back of the heel, and one in the center of the medial side of the foot (Fig. 2-1). Locations on the perimeter of the foot were selected as they provided regions where obstacle contact may occur and would be intuitive in conveying obstacle proximity to the foot. While little is known about the perceptual characteristics on the front of the big toe (Location 1), heel (Location 5), and medial part of the foot (Location 6), it’s been suggested that cutaneous reflexes from stimulation of the sural nerve (lateral side of foot, Locations 2-4) during walking stabilize gait against perturbations from uneven terrain or obstacle collision [65]. Additionally, the lateral side of the foot would likely encounter more obstacles compared to the medial side, so half of the locations were assigned here. Vibrations were created by small haptic motors (Vibrating Mini Motor Disc, Adafruit, New York City, NY) of 10mm diameter and 2.7mm thickness. All vibrations were 1.5 seconds in duration and consisted of one of the four vibration levels at an amplitude of roughly 0.8-1.2g: High (11000 RPM), Low (2750 RPM), Increase and Decrease. This amplitude and frequency range is typical
for vibrotactile displays, including the few that have been applied to different parts of the foot [48, 51, 49]. The Increase vibration level went from the Low level to the High level and the Decrease vibration level did the opposite. Each of the haptic motors were controlled by an individual driver board (DRV2605, Texas Instruments, Dallas, TX), and all driver boards for the same foot were connected in series through a digital I²C line. The driver boards received input in the form of a Pulse Width Modulation (PWM) signal from the digital output pins on a microcontroller (Arduino UNO, Arduino, Massimo, Italy). The High vibration level had a 100% duty cycle at 5V while the Low vibration level had a 25% duty cycle at 5V. The haptic motors were placed on participants’ feet with double-sided tape and reinforced with athletic wrap for the duration of the experiment (Fig. 2-2). Participants used a custom graphical user interface (GUI) to complete the experiment, (Fig. 2-1), which commanded the motors via serial ports and recorded participant responses for each trial. The buttons around the foot icons on the GUI were for vibration location responses, while the buttons on the right side were for vibration type responses. The Python code for the GUI and the C code for the microcontroller algorithm can be found in Appendix A.

2.1.3 Experimental Protocol

Participants completed the experiment while in a focused state of attention (Focused condition) and in a distracted state of attention (Distracted condition), where the order of these conditions were counterbalanced (i.e. participants completed the trials for one or the other first). Participants randomly assigned to order DF performed the Distracted trials before the Focused trials, and vice versa for order FD. During the Focused trials, participants were instructed to focus on their feet and pay close attention to the vibration sensations. For the Distracted trials, participants were presented a random number between 0 and 100 at the beginning of each trial and instructed to count up from that number in increments of three until they felt the vibration.
Figure 2-1: Fritzing diagram of haptic display and electrical connections
$\text{I}^2\text{C}$ SCL (green), $\text{I}^2\text{C}$ SDA (yellow), power (red), ground (black), and
digital outputs (pink)
Four different vibration levels for six different locations per foot resulted in 48 unique vibration combinations, and participants experienced each combination for six trials during each attention condition, totaling to 576 trials overall. The 288 trials for each attention condition (Focused or Distracted) were randomized in a predetermined order. The numbers used in the Distracted trials were the same for all participants to maintain the total difficulty of each trial (e.g. some numbers may be harder to count up from). To ensure that participants could not predict when the vibration would occur during each trial, each vibration took place at a random point in time between 2-7 seconds after pressing the “Next Trial” button on the GUI. Once a vibration was felt, participants selected the location on the foot where they perceived the vibration and then selected the type of vibration felt. Each response was recorded and deemed correct or incorrect, but participants did not receive this feedback. If a participant did not perceive a vibration, they were instructed to select a button labeled “I didn’t feel it”, and that trial’s response was recorded as undetected.

Figure 2-2: Example participant wearing the experimental device and operating the GUI

2.1.4 Statistical Methods

Independent variables consisted of Attention (Focused, Distracted), Order (FD or DF), Type (High, Low, Increase, Decrease), Location (1-6) and Foot (Right, Left). Order was the only between-subject variable and the rest were within-subject variables. Each trial resulted in two measurements: perceived vibration location and
perceived vibration type. The location responses were used to calculate perceived Location Accuracy scores while the vibration type responses were used to calculate perceived Type Accuracy scores. As a result, each participant had two separate accuracy scores (i.e. Type and Location) for each of the 48 unique vibration combinations. Each accuracy score was independently calculated by dividing the number of correct responses by the total number of trials for that type/location combination (i.e. six). An undetected trial counted as an incorrect response in both calculations for accuracy. The independent variables of Type and Location did not pass Mauchly’s Test of Sphericity, an important assumption in a Repeated Measures Analysis of Variance (RMANOVA), so non-parametric tests were performed. These tests included the Wilcoxon Signed Rank, Wilcoxon Ranked Sum, and the Friedman test (Table 2.1). All post-hoc tests utilized a Bonferroni Correction and the family significance level was $\alpha = 0.05$. Effect sizes for significant pairwise differences were measured with Hedges’ $g$.

Table 2.1: Non-parametric tests used to determine effects of independent variables on non-spherical dependent variables

<table>
<thead>
<tr>
<th>Independent Variable</th>
<th>Variability</th>
<th>Levels</th>
<th>Statistical Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attention</td>
<td>Within-Subjects</td>
<td>2</td>
<td>Wilcoxon Signed Rank</td>
</tr>
<tr>
<td>Order</td>
<td>Between-Subjects</td>
<td>2</td>
<td>Wilcoxon Rank Sum</td>
</tr>
<tr>
<td>Vibration Type</td>
<td>Within-Subjects</td>
<td>4</td>
<td>Friedman</td>
</tr>
<tr>
<td>Vibration Location</td>
<td>Within-Subjects</td>
<td>6</td>
<td>Friedman</td>
</tr>
<tr>
<td>Foot</td>
<td>Within-Subjects</td>
<td>2</td>
<td>Wilcoxon Signed Rank</td>
</tr>
<tr>
<td>Interactions</td>
<td>Within-Subjects</td>
<td>8-24</td>
<td>Friedman</td>
</tr>
</tbody>
</table>

2.2 Results

2.2.1 Perceived Location Accuracy

Overall, participants performed well in discriminating the locations where vibrations occurred (Fig. 2-3). A Friedman test supported a significant main effect of Location
on Location Accuracy (Table 2.2). Post-hoc tests support significantly less accuracy for Location 3 compared to Location 1 ($p=0.0163$, $g=1.0$), Location 5 ($p=0.0331$, $g=0.91$), and Location 6 ($p=0.0045$, $g=1.04$). On the right foot, the lower accuracy seen for Location 3 can mostly be attributed to undetected Low vibrations here (discussed further in next sections). On the left foot, lower accuracy is a result of participants confusing Location 3 with neighboring Locations 2 and/or 4.

Table 2.2: Statistics from a Friedman test on the effect of Vibration Location on Perceived Vibration Location Accuracy (pooled across Attention, Type, Order, and Foot)

<table>
<thead>
<tr>
<th>Source</th>
<th>SS</th>
<th>d.f.</th>
<th>MS</th>
<th>Chi-Sq.</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vibration Location</td>
<td>2547.99</td>
<td>5</td>
<td>509.60</td>
<td>69.09</td>
<td>&lt;0.0001</td>
</tr>
<tr>
<td>Interactions</td>
<td>1819.89</td>
<td>45</td>
<td>40.44</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Error</td>
<td>4114.13</td>
<td>180</td>
<td>22.86</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Total</td>
<td>8482.00</td>
<td>239</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Figure 2-3: Average perceived Location Accuracy and standard error for each foot and attention state (pooled across Type and Order)
2.2.2 Perceived Type Accuracy

Wilcoxon Signed Rank and Rank Sum tests support significant main effects of Attention and Order on Type Accuracy (Table 2.4). Participants had higher accuracy in discriminating vibration types during Focused trials (M=79.82%, SD=27.98%) compared to Distracted trials (M=70.76%, SD=32.78%), with a p <0.0001 and small effect size of $g=0.30$. Additionally, those assigned to Order DF (Mean=80.34%, SD=32.78%) had higher accuracy scores than those in Order FD (Mean=70.24%, SD=27.80%), with a p <0.0001 and small effect size of $g=0.33$.

Table 2.3: Statistics from Wilcoxon Signed Rank and Rank Sum tests on the effects of Attention and Order on Perceived Vibration Type Accuracy (pooled across Location, Type, and Foot)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Level</th>
<th>Mean</th>
<th>SD</th>
<th>Rank</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attention</td>
<td>Focused</td>
<td>79.82%</td>
<td>27.98%</td>
<td>Signed Rank:</td>
<td>&lt;.0001</td>
</tr>
<tr>
<td></td>
<td>Distracted</td>
<td>70.76%</td>
<td>32.78%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Order</td>
<td>FD</td>
<td>70.24%</td>
<td>32.78%</td>
<td>Rank Sum:</td>
<td>&lt;.0001</td>
</tr>
<tr>
<td></td>
<td>DF</td>
<td>80.34%</td>
<td>27.80%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A Friedman test supported a significant effect of Type on Type Accuracy (Table 2.3). Accuracy scores are higher for the static vibrations of High and Low than those for dynamic vibrations of Increase and Decrease (Fig. 2-4). While post-hoc tests did not show significant pairwise comparisons, a contrast test (Wilcoxon Signed Rank) comparing static vibrations (Low and High pooled scores) and dynamic vibrations (Increase and Decrease pooled scores) supported a significant difference (p <0.0001, $g=0.95$).
Table 2.4: Statistics from a Friedman test on the effect of Vibration Type on Perceived Vibration Type Accuracy (pooled across Attention, Location, Order, and Foot)

<table>
<thead>
<tr>
<th>Source</th>
<th>SS</th>
<th>d.f.</th>
<th>MS</th>
<th>Chi-Sq.</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vibration Type</td>
<td>125897.78</td>
<td>3</td>
<td>41965.93</td>
<td>188.45</td>
<td>&lt;0.0001</td>
</tr>
<tr>
<td>Interactions</td>
<td>141152.81</td>
<td>27</td>
<td>5227.88</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Error</td>
<td>367617.92</td>
<td>920</td>
<td>399.58</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Total</td>
<td>634668.50</td>
<td>959</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Figure 2-4: Perceived vibration Type Accuracy mean and standard error for each Location, Type, and Foot (pooled across Attention)

A significant interaction between Foot and Location was supported by a Friedman test (Table 2.5). Post-hoc tests do not show any significant pairwise comparisons. The significant omnibus test may have been influenced by lower Vibration Type Accuracy at Location 3 for Increase and Low vibrations on the right foot, as the overall mean for Location 3 on the right foot was lower than the rest (Fig. 2-4). A possible explanation for this phenomenon is highlighted in the next section.
Table 2.5: Statistics from a Friedman test on the interaction of Foot*Location on Perceived Vibration Type Accuracy (pooled across Attention, Type, and Order)

<table>
<thead>
<tr>
<th>Source</th>
<th>SS</th>
<th>d.f.</th>
<th>MS</th>
<th>Chi-Sq.</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Foot*Location</td>
<td>18453.76</td>
<td>11.00</td>
<td>1677.61</td>
<td>27.62</td>
<td>0.0037</td>
</tr>
<tr>
<td>Interaction</td>
<td>45947.49</td>
<td>99.00</td>
<td>464.12</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Error</td>
<td>570267.25</td>
<td>840.00</td>
<td>678.89</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Total</td>
<td>634668.50</td>
<td>959.00</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

2.2.3 Undetected Vibrations
Vibrations at select locations were sometimes undetected, and occurred most frequently (i.e. approximately 20% of the time) for Low vibrations at Location 3 on the right foot (Table 2.6). While some participants occasionally did not detect vibrations at other locations, the majority of participants had difficulty perceiving Low vibrations at Location 3 on the right foot (Fig. 2-5). It is important to take these data into consideration while examining the main effect of Location on vibration Location Accuracy and the interaction of Location and Foot on vibration Type Accuracy, as an undetected vibration response resulted in an incorrect score for each accuracy type. While Location 3 sometimes caused location discrimination difficulties on the left foot, the undetected Low vibrations on the right foot at Location 3 certainly contributed to the lower accuracy scores seen for this Location, Type, and Foot.

Table 2.6: Percentages of undetected vibrations across all trials and participants. All dashed cells and cells not shown have zero undetected responses for those trials.

| Type   | Focused | | | | | | Distracted | | | |
|--------|---------|--------|--------|--------|--------|---------|--------|--------|--------|--------|--------|--------|
|        | R3      | R5     | L1     | L2     | L5     |        | R3      | R5     | L2     |        |        |        |
| High   | -       | -      | 1.7%   | -      | -      | -       | -       | -      | -      | -      | -      | -       |
| Low    | 21.7%   | 1.7%   | -      | 6.7%   | -      | 20%     | 3.3%   | 5%     |        |        |        |        |
| Increase| -      | -      | -      | -      | -      | -       | -       | -      | -      | -      | -      | -       |
| Decrease| -     | -      | -      | -      | 1.7%   | -       | -       | -      | -      | -      | -      | -       |
2.3 Discussion

This study examined the effect of Attention, Foot, vibration Location, and vibration Type on vibrotactile foot perception. It was hypothesized that distracted attention states would decrease perception accuracy due to limited attentional resources, and that certain vibrations types would be easier to detect than others. Overall, these results provide useful implications for vibrotactile interface design for the feet.

Results show that tactile perception of vibration patterns (i.e. Type) is degraded during distracted states of attention. Mean Type Accuracy was approximately 9% lower during the distracted trials, although the small effect size may be due to pooling of other variables. This has implications when designing interfaces that map critical information to sensory signals. Detection of time critical signals must be robust enough to eliminate ambiguity and ensure adequate signal interpretation. While attention state affected perceived vibration Type Accuracy in the current study, it did
not significantly affect perceived Location Accuracy. This implies that signal detection was robust, but comprehension of secondary information was slightly impaired (i.e. small effect size). Therefore, when designing vibrotactile information presentation devices, it may be more reliable to use location to convey critical information while other details in the signal can provide additional sub-critical information; for example, in the context of obstacle avoidance, vibration location could convey the location of an obstacle while the vibration signal type could convey proximity to the obstacle. Once the user is adequately notified of the presence and location of the item (critical component), they could choose to interrupt current tasks to examine additional details of the obstacle (sub-critical components) if signal type perception wasn’t salient. It’s important to note that small effect sizes may be a result of pooling data across groups for non-parametric tests, which results in higher overall variability.

The significant effect of Order on Type Accuracy is an interesting finding. Participants who completed Distracted trials first (Order DF) were 10% more accurate on average in distinguishing vibration types than those who completed the Focused trials first (Order FD). The effect size was small ($g=0.33$), although could have been due to the pooling of other variables. Type Accuracy did not show any trend with time or trial number, but nonetheless there may be some effect of attention on learning and/or detection over time. For example, if participants became tired or their feet became less sensitive over time (each set of trials consisted of 288 vibrations), it’s possible that ending the experiment with the information processing load of the Distracted trials could have further impaired perception.

With average perceived Location Accuracy scores close to 100% for most foot locations, vibrations at these locations are promising for applications in tactile displays. Although, limitations exist with the locations on the lateral side of the foot. Sensation in this area may not have fine enough location resolution, as some participants often confused it with neighboring Locations 2 or 4 on both feet. Additionally, this location may be less sensitive, as participants often did not feel the Low vibration here on the right foot, which contributed to the lower perceived Location Accuracy for Location 3 and lower perceived Type Accuracy for Location 3 on the right foot.
The poor detectability at this location may also have been due to a mechanical issue with that particular vibration motor; it is possible that the Low vibration level of this motor was different from that of the other motors (e.g. lower magnitude). Detection difficulties at other locations were much less common and specific to the participant. Subject-specificity of tactile thresholds on other foot locations has been observed in previous research [50, 46]. Going forward, it may be favorable to only include one or two locations on the lateral side of the foot for vibrotactile inputs (i.e. Locations 2 and/or 4).

User abilities in perceived vibration Type Accuracy vary by Type - static vibration types (High and Low) were more easily detected than dynamic vibration types (Increase and Decrease). On average, the static types were detected with higher accuracy by roughly 26% and a contrast between the two sets (i.e. static vs dynamic) had a large effect size. Participants reported occasional difficulty in detecting the difference between High vibrations and the dynamic vibrations (Increase and Decrease), which was observed in their trial responses. Participants stated that they usually selected High during these moments of confusion, which is consistent with lower accuracy for dynamic vibrations, with Decrease types being the hardest to distinguish. Participants reported that when a vibration started at the High level and then decreased, it was harder to detect this change than in the case for an increasing vibration, which could be due to the cutaneous sensation of the High level desensitizing the skin to the proceeding lower levels. A study simulating a monitoring task for an anesthesiologist in an operating room employed a tactile display to indicate changes in patient blood pressure, and results showed that tactile “change blindness” (i.e. failure to recognize a change in signal) occurred more often when the change in vibration intensity was gradual as opposed to discrete [66]. These results suggest that decreasing vibration signals are not adequately detectable and should not be used for critical information presentation. It is hypothesized that if the Decrease type was removed from the current study, perceived Type Accuracy for High vibrations may have been higher than those for Low vibrations, especially since High levels were rarely undetected. Regarding the Increase vibration type, distinguishing these from High types may be
easier than is the case for Decrease types, but there is still insufficient evidence that it’s reliable enough to convey critical information in a haptic interface application.

Vibrations were rarely undetected except for those of type Low at Location 3 on the right foot, which had an unusually high occurrence (about 20% of all R3-Low trials). It is unclear why the undetected signals are only for the right foot. The vast majority of participants were right-handed, so it is possible that the lateral side of the dominant foot is less sensitive and has a higher tactile detection threshold, but this is just speculation. The Low, Increase, and Decrease vibration types on the Right foot appear to have more variability in perceived Type Accuracy across locations, suggesting that the perception capabilities of the Right and Left feet do differ in some ways. It is also worth speculating that this phenomenon could have been device related (e.g. the motor at this location was below specification during Low vibrations). While the same type of motor was used at each location, it would be useful to confirm consistent vibration output levels with an accelerometer in the future.

2.4 Implications for Vibrotactile Displays

Overall, results demonstrate that the haptic perception capabilities of the feet for selected locations and vibration types is sufficient for use in a vibrotactile interface. High and Low vibrations are successfully perceived at most of the locations examined. Careful consideration should be taken when utilizing quickly increasing/decreasing continuous vibrations or multiple locations on the lateral side of the foot. Most importantly, signal perception should be robust enough to withstand attentional loading, so vibrotactile signal location should convey the critical information while the more subtle signal properties (e.g. intensity, duration, and patterns) can supplement with less critical information. Future work could examine different locations on the foot, different sizes in applied vibration area, or alternate dynamic patterns such as pulsing vibrations with unique interval timing.

The results of the current study were used to aid the tactile mapping design
of the multi-modal device for obstacle avoidance, where sensory reinforcement via visual channels was also incorporated. To narrow the large scope of an obstacle avoidance task, which could involve walking around an obstacle or stepping over it, the current body of work focused on multi-modal information display for the strategy of approaching and stepping over an obstacle. As a result, only one location from the current perception study was selected for use in the device - the front of the foot. Other locations examined in the current perception study could certainly be utilized in a version that supports the strategy of walking around an obstacle (e.g. vibrations on the perimeter of the foot could help guide participants away and/or around obstacles). While the current application of interest involves a wearable interface for obstacle avoidance, the growing range of computing devices, computational power and input/output capabilities opens doors for numerous haptic applications in human-computer interaction [67]. Some other applications that could benefit from vibrotactile inputs to the feet include advanced navigational devices, gait correction devices, or immersive virtual reality environments for entertainment and gaming.
Chapter 3

Design of the Multi-modal Interface Device

In order to assess whether multi-modal cues could benefit astronaut obstacle avoidance performance during planetary EVA, a multi-modal information presentation device was designed and developed to use in human subject research experiments. The multi-modal display was designed to provide cues to the visual and tactile modalities through two separate devices that were integrated. The visual display was implemented with augmented reality glasses (AR Glasses) while the tactile display was implemented through custom-made, adjustable vibrotactile boots that applied vibrations to the front of the foot. The vibrotactile boots contained embedded sensors that were used to determine the obstacle location and proximity information presented to both displays. Each boot was equipped with sensors and vibration motors, all connected to a small microcontroller that processed the sensor data and determined which cue to send to the user. To simulate the perceptual limitations of a suited astronaut, the AR Glasses were modified to block the lower peripheral vision and the vibrotactile boots were made rigid enough to resist deflection.
3.1 Display Cue Logic

Both of the displays provided four different cues based on distance information from two sensors, with each corresponding to a particular distance range for objects directly in front of the boots (Fig. 3-1). Previous research investigating detection and discriminability for different vibration patterns on various body parts while walking found that most people can discrimination among 4-5 different vibration stimuli [46]. As a result, only four different cue levels were chosen. The first cue signified an object at a distance range between 30 to 60 inches, the second cue signified a range between 8 to 30 inches, the third cue signified a range between 1.5 to 8 inches, and the fourth cue signified the boot was within 1.5 inches of the object. Since people need information about approaching obstacles roughly two steps prior [61, 68, 62, 63] and the average step size for a human is roughly 30 inches [69], the first cue level was set to 60 inches. Although, the vibrotactile boot’s rigid base alters the gait and significantly shortens step length, so the first cue was actually presented far beyond two steps distance. Spacesuit testing reveals that step length is also significantly reduced when fully suited [70]. Pilot testing showed that optimal toe-off distance for the leading foot was roughly between 8 inches and 1.5 inches while wearing the vibrotactile boots, and anything closer than 1.5 inches was prone to collision. This is consistent with previous findings [45, 64] that incorrect foot placement is a primary factor in obstacle contact.

<table>
<thead>
<tr>
<th>Display</th>
<th>Cue 1 (30-60 in)</th>
<th>Cue 2 (8-30 in)</th>
<th>Cue 3 (1.5-8 in)</th>
<th>Cue 4 (0-1.5 in)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Visual</td>
<td>2 Hz blink</td>
<td>4 Hz blink</td>
<td>8 Hz blink</td>
<td>Steady</td>
</tr>
<tr>
<td>Tactile</td>
<td>2 Hz low pulse</td>
<td>4 Hz medium pulse</td>
<td>8 Hz med-high pulse</td>
<td>Steady high vibration</td>
</tr>
</tbody>
</table>

Figure 3-1: Display cue characteristics for each level and modality

The augmented reality glasses displayed simple outlines of feet in the bottom right and left corners of the screen, and each of the four cues showed up on the tip of the
foot icons (Fig. 3-1), where the opaqueness of the circle and blinking rate increased as distance decreased. The vibrotactile boots applied a cue-specific vibration signal on the front of the foot. Vibration cues varied in frequency (perceived as intensity) where Low intensity was 4,000 RPM (67 Hz), Medium intensity was 6,000 RPM (100 Hz), Medium-High intensity was 8,000 RPM (133 Hz), and High intensity was 11,000 RPM (183 Hz). The amplitude of all vibrations was roughly 0.8-1.2g. Previous research has found that perceived intensity (a function of frequency) and perceived duration are the two dominant features affecting vibration discriminability [46]. Additionally, research on bimodal warning signals has shown that increased signal pulse rate is associated with perceived urgency [28]. These findings helped guide the information mapping, resulting in assigned pulse frequencies of 2Hz, 4Hz, 8Hz, and steady for cues 1-4, respectively. This decision to incorporate discrete, pulsing vibrations was also informed by results from the vibrotactile study (Ch. 2), which demonstrated participants’ difficulty in discriminating changes in vibration intensity when the signal was continuous. The lowest vibration level of 4,000 RPM was selected to be higher than the low vibration type (2750 RPM) in the vibrotactile perception study in order to increase cue salience as robust detection could be affected through the physical action of walking.

3.2 Hardware Design and Development

To ensure that the vibrotactile boots could be worn by participants of varying foot sizes, the boots were designed similarly to crampons; that is, a rigid front base and rigid back base were connected through a flexible spring-steel center bar with holes along its length (Crampon Flex Center Bar, Black Diamond, Salt Lake City, Utah) and a metal spring clip (Crampon Spring Clip, Grivel, Courmayeur, Italy) on the back base was used to insert a stiff pin through the hole for a specified size in order to modify the length. The custom, 3D-printed front and back bases were designed to support this mechanism as well as accommodate sensor housing and neoprene strap integration. Adjustable neoprene vamps and velcro ankle straps were sewn onto the
boots to support a comfortable fit for each participant. The rigid, crampon-like design was chosen in order to make the boots adjustable as well as to simulate a pressurized EVA boot with limited deflection capability.

Figure 3-2: The vibrotactile boot base assembly (left) and CAD models of the 3D-printed front base (top right) and back base (bottom right)

The front of each vibrotactile boot contained an ultrasonic rangefinder (MB1040 LV-MaxSonar, MaxBotix, Brainerd, MN) and a proximity sensor (VCNL4010 Proximity/Light Sensor, Adafruit, New York City, NY). The rangefinder was used to discriminate distances to objects between 8 inches to 60 inches and the proximity sensor was used to measure distances to objects closer than 8 inches. To ensure that the vibrotactile boots did not mistake the floor for an obstacle during toe-down gait maneuvers, a six degree-of-freedom Inertial Measurement Unit (IMU Digital Combo Board ITG32/ADXL345, SparkFun, Niwot, Colorado) was embedded in the front of the boot and used to determine orientation. Each vibrotactile boot applied vibrations on the front of the foot with two small haptic motors (Vibrating Mini Motor Discs, Adafruit, New York City, NY) of 10mm diameter and 2.7mm thickness. Haptic motors were controlled by individual driver boards (DRV2605, Texas Instruments, Dallas, TX) that received input in the form of a Pulse Width Modulation (PWM) signal from the microcontroller’s digital output pins.
The proximity sensor, haptic driver boards, and inertial measurement unit on each boot all operated on 3.3V and were connected in series on a digital I²C line. The analog ultrasonic rangefinder operated on 5V and was connected to the analog input pins of the microcontroller. The protoboard and microcontroller (Arduino UNO Wi-Fi, Arduino, Massimo, Italy) for each boot were housed in a box that was secured to the user’s calf with velcro straps. Wiring coming out of the housing box was consolidated through a flexible braided cable sleeve, and male Hitachi Style connectors were used to terminate the wires. On the vibrotactile boot, wiring coming from the sensors and motors was consolidated with electrical tape and terminated in female Hitachi style connectors. The use of connectors allowed all electronics to be disconnected during boot sizing and fitting, minimizing damage. The integrated vibrotactile boot is shown in Fig. 3-3 and a schematic of the electrical connections is summarized in Fig. 3-4.
3.3 Algorithms

The vibrotactile boot microcontroller’s main algorithm was written in the C language and involved a loop operating on the logical flow in Fig. 3-5. Readings from the triple-
axis ADXL345 accelerometer and ITG-3200 gyroscope on the 6-DOF IMU were used to determine orientation of the rigid vibrotactile boot base through the use of a complementary filter, which was compared to the baseline orientation of the boot base during power-on. If the boot was tilted forwards or backwards more than two degrees from the baseline orientation, the obstacle proximity cue was not updated. If the boot was within two degrees of the baseline orientation, all of the sensors were read and this data was used in a sequence of if-else statements to determine the appropriate cue level. This orientation logic assumes that the participant is walking on flat ground, as small tilts in orientation are treated as changes in the gait cycle as opposed to changes in ground inclination or declination. The cues were sent over Wi-Fi to the AR Glasses in real-time. Once the cue changed, the appropriate digital signals were sent to the haptic driver boards to actuate the haptic motors accordingly and the AR Glasses updated the cues on the display. Since the boots were almost always at a tilt angle greater than two degrees when in the air during the gait cycle, the boots would update cues every time they touched back down on the ground. The C Code implementing this algorithm can be found in Appendix A.

Figure 3-5: The vibrotactile boot algorithm flowchart
The microcontroller’s processing speed wasn’t fast enough to transmit information over Wi-Fi to the AR Glasses in real-time, so a small computer (Edison Compute Module, Intel, Santa Clara, CA) was used to take over the Wi-Fi data transmission role. This solution was non-optimal, but came about in order to meet software requirements without changing the hardware implemented on the vibrotactile boot. The Edison computer was attached to a breakout board (Intel Edison R2 Kit for Arduino, Intel, Santa Clara, CA), allowing serial communication with the Arduino UNO Wi-Fi microcontroller. The board combination allowed real-time cue data transmission to the AR Glasses via a Wi-Fi connection. The C Code implementing the Wi-Fi functionality on the Edison computer can be found in Appendix A.

3.4 Augmented Reality Glasses

The visual cues were implemented through augmented reality glasses (Moverio BT-200 Smart Glasses, Epson, Nagano, Japan) that received Wi-Fi updates of the cue information from both vibrotactile boots in real-time. A custom Unity 3D application was loaded onto the Epson BT-200 system’s Central Processing Unit, which runs Android. Unity 3D was selected due to it being a popular cross-platform game engine that works well on Android operating systems. The program was responsible for reading cue data over Wi-Fi and projecting the appropriate cue images in the lower left and right corners of the glasses in such a way that did not require users to converge their eyes to see the images; rather, users could keep their natural focus ahead and still be able to see the cues with visual fidelity. To simulate the effect of limited vision and restricted range of view caused by a helmet, 45mm x 27mm 3D-printed shields were adhered to the bottom of the glasses to block peripheral vision of the lower limbs (Fig. 3-6). The Epson Moverio BT-200 UV Shades were installed on the glasses in order to create visual contrast effects similar to those caused by the UV filter on spacesuit helmets.
3.5 Summary and Limitations

The vibrotactile boots were designed specifically for controlled human subject experiments in a laboratory setting (i.e. on smooth, flat ground). Since the boots were almost always at a tilt angle greater than two degrees when in the air during the gait cycle, the boots would update cues every time they touched back down on the ground. This characteristic made the boots only appropriate for flat ground, as opposed to inclined/declined terrain that would likely be encountered on Mars. A more advanced boot would need to be developed for obstacle avoidance in a realistic environment, and should convey distance information while the boot is in the air in order to increase spatial awareness during the approach and step-over phase. This could be incorporated in a future design through the use of pressure sensors or force sensors on the bottom of the boot instead of relying on orientation data from an IMU. While many technical advances could be made, the design and functionality of the vibrotactile boots were sufficient in addressing the research requirements of the current study.
Chapter 4

Analysis of the Multi-modal Interface Device

This study examined how people utilize relative obstacle proximity information via visual cues and tactile cues on the front of the feet (alone and in combination) using the wearable multi-modal interface during obstacle avoidance trials. The trials simulated aspects of walking in pressurized boots (i.e. the rigidity of the foot plantar surface) with partially occluded vision, a potentially common future astronaut scenario during planetary exploration. It was hypothesized that the visual and tactile cues would improve participants' ability to avoid obstacles while reducing head-down time during limited vision conditions. It was also hypothesized that use of visual and tactile cues would modify the gait strategy in ways that reduce risks of obstacle contact. Results from this study can contribute to guidelines for the design and implementation of multi-modal information presentation for complex, perceptually-constrained physical tasks.
4.1 Experimental Methods

4.1.1 Participants

The participants consisted of sixteen healthy adults (6 females, 10 males) between the ages of 23 and 33 (M=25.63, SD=1.03). The experimental protocol was approved by the MIT Committee on the Use of Humans as Experimental Subjects (COUHES) and all participants provided written consent. Participants were excluded from the study if they reported any physical injuries or cognitive impairments that could make the obstacle avoidance tasks a hazard to their health. Participants received up to $20 USD in compensation.

4.1.2 Experimental Protocol

A 10-camera motion capture system was used (Bonita Cameras, Vicon Motion Systems, Oxford, United Kingdom) to capture and quantify the human kinematics of each trial at 100 Hz. Retro-reflective position markers (Pearl Markers, B&L Engineering, Santa Ana, CA) were placed on the head and torso of the participants, as well as the obstacles and vibrotactile boots. Each participant completed a set of 16 trials in each of the four display conditions: Tactile Only (TO), Visual Only (VO), Visual and Tactile Combined (VT), and with No Cues (NC). Each participant was randomly assigned to one of two display type trial set orders: NC, VO, TO, VT (Order A) or VT, VO, TO, NC (Order B). These two orders were selected in order to examine the potential confounding factor of display order. Participants wore the multi-modal display device for all trials in order to maintain consistent perceptual and mobile characteristics; the smart glasses were put to sleep during TO and NC trial sets, and the vibration motors were unplugged for the VO and NC trial sets. The first four trials in each set were training trials while the remaining 12 were experimental trials. For all trials, participants were instructed to approach and step over the obstacle from one of three starting locations; while the participants were under the impression that the location of the obstacle was always random, it was ac-
tually placed in a specific location corresponding to the starting location (Fig. 4-2). Participants were instructed to look straight ahead once at the starting location to signify that they were ready for the next trial to start. Participants were instructed to make obstacle avoidance their first priority (i.e. not to touch the obstacle in any way) and reduce head-down time as their second priority (i.e. avoid looking down at the obstacle during the trial). They were also told that completion time was not a performance metric and to take as much time as needed to finish each trial. For each trial, the obstacle consisted of one block (small obstacle) or two stacked blocks (large obstacle), where the dimensions of each block were 9x6x4 inches for length, width, and height, respectively (Fig. 4-1). Obstacle size and location were randomized for each set of trials, but kept consistent across participants. Before starting trials, each subject went through the Epson BT-200 calibration application protocol with the augmented reality glasses to ensure proper projection of the visual cues.

**Figure 4-1: The obstacle blocks with retroreflective markers**

After each display type set of trials, participants completed a NASA Task Load Index (TLX) [71] survey where they ranked the workload of the preceding set of trials based on six types: Mental, Physical, Temporal, Performance, Effort, and Frustration. The NASA TLX uses these six types of workload to generate an Overall Workload score. After the entire experiment was complete, participants completed a Post-Experiment Survey regarding the workload of each display, intuitiveness of the cues,
and display preferences. They were asked to rank the workload of each display on a scale from 1-5, to rank the intuitiveness of the visual cues and tactile cues on a scale of 1-5, and to rank their display preference from 1-4 where 1 represents most preferred display and 4 represents least preferred display. They were also asked to select which display they found most useful during the obstacle avoidance trials.

Figure 4-2: An example user wearing the multi-modal display (left) and the motion capture layout (right) with obstacle (OB) and associated starting point locations for experimental trials (1-3) and training trials (T1-T4). Each square in the grid is approximately 23x23 inches.
4.1.3 Data Processing

Kinematic data from the Vicon motion capture system were used to calculate eight metrics: Normalized Completion Time, Head-Down Percentage of Trial, Toe-Off Distance, Toe Clearance, Number of Steps, Normalized Step Length, Normalized Step Variance, Lead Foot (i.e. the one that crossed over obstacle first) and whether or not a collision occurred for each trial. The Head-Down Percentage of Trial metric was chosen out of relevance to the operational scenario of interest, where an astronaut must navigate rocky terrain while not continuously looking down at the ground. The gait metrics of Toe-off Distance, Toe Clearance, and Step length are considered key measures when examining the control of lower limb trajectories over obstacles [57]. Number of Steps, Step Variance, and Normalized Completion Time were chosen simply to provide more details about obstacle avoidance strategies across display conditions. All of the metrics were calculated in MATLAB during post-processing, and the code can be found in Appendix B.

Completion time was defined as the amount of time it took the participant to go from the starting location to the end of the motion capture space. A trial officially started once the participant was standing at the appropriate starting location, looking straight ahead, and verbally told “Go”. A position marker was placed at the end of the motion capture space, and the end frame of the trial occurred when the right or left boot position marker crossed the end marker. These times were normalized by subject through dividing them by the longest completion time of all the trials for each subject. Data were normalized by subject in order to control for confounds related to differences in strategy or innate walking speed.

Head-Down Percentage of Trial was defined as the percentage of trial time the participant spent looking down at an angle \( H \) of 10 degrees or more (Fig. 4-3). This angle was determined based on pilot testing with the motion capture system, which indicated that the head naturally moves within the 0-10 degree range even while keeping the gaze straight ahead. The first several data points from the three head markers were used to solve for an initial head plane, a vector \( N \) normal to the plane,
and the angle between this normal vector and the vertical Z axis, $H_0$. The changing angle $H$ between the head plane’s normal vector and the vertical axis was calculated for each motion capture frame, and all of the frames where $H$ was 10 degrees or more beyond $H_0$ were summed and divided by the capture rate (100 Hz) to get head-down time in seconds. These values were later divided by the completion time for each trial.

Figure 4-3: The initial head-level state $H_0$ for each participant based on initial position marker readings (left) and the geometry showing the head-down angle measurement $H$ (right)

Step characteristics were calculated using the retroreflective marker on the front of the vibrotactile boot and four markers on the obstacle. The Lead Foot for a given trial was defined as the first foot to step over the obstacle, and Toe-Off Distance was defined as the horizontal distance between the front of the Lead Foot boot and the obstacle during toe-off to step over the obstacle. Toe Clearance was defined as the minimum vertical distance between the top of the obstacle and the bottom of the Lead Foot boot while stepping over the obstacle. After calculations were made, a distance of 5.3cm was subtracted from Toe Clearance and a distance of 7cm was subtracted from Toe-Off Distance to account for the placement of markers relative to the boot and obstacle. The variable Number of Steps simply refers to the number of steps taken by both feet during the trial. The other two step characteristics, Normalized Step Length and Normalized Step Variance, refer to the average step length during the trial and the variance in step length throughout the trial, respectively. These two step characteristics were normalized by subject in order to control for confounds related
to the effect of participant height/leg length on gait differences, and normalization was performed through dividing these data by the leg length of each subject. A trial was considered to have a collision if one of the four position markers on the obstacle moved more than 3mm during the duration of the trial. Each trial could have a maximum of one collision.

4.1.4 Statistical Methods

Subjective Measures

Since the Post-Experiment Survey workload scores for each display were on a scale from 1-5 and the display preference responses were rankings from 1-4, non-parametric Friedman tests were used to compare these data. The cue intuitiveness scores were on a scale of 1-5 with only two levels (i.e. visual cues and tactile cues), so a Wilcoxon Signed Rank test was used to compare the data. NASA TLX Workload scores were not normally distributed so Friedman tests were used to determine the effect of Display on workload measures. All post-hoc tests utilized a Bonferroni Correction and the family significance level was $\alpha = 0.05$. No correction was made for multiple dependent variable tests. Effect sizes for significant pairwise differences were measured with Hedges' $g$.

Motion Capture Metrics

Independent variables consisted of Display (TO, VO, VT, NC), Order (A or B), Size of obstacle (Small or Large), Sex (M or F), and the blocking factor of obstacle Location (1,2,3). Order and Sex were between-subject variables while the rest were within-subject variables. Kolmogorov-Smirnov tests showed that none of the motion capture metric data were normally distributed. Additionally, only two variables passed Mauchly’s Test of Sphericity: Normalized Completion Time and Head-Down Percentage of Trial. Since Sphericity is thought to be a more important assumption for Repeated Measures Analysis of Variance (RMANOVA), a five-factor RMANOVA was performed on Normalized Completion Time and Head-Down Percentage of Trial.
Results matched those of non-parametric tests for the two variables, so it was determined that the RMANOVA results were acceptable. The RMANOVA was preferred over non-parametric tests due to the benefit of accounting for variance from all sources and assessing interaction effects.

For all of the other dependent variables, non-parametric tests were used. These tests included the Friedman test, Wilcoxon Signed Rank test, Kruskal-Wallis test, and Wilcoxon Rank Sum test (Table 4.1). Only the effects of Display, Order, and Size were considered important and operationally relevant to the current study. Location was simply a blocking factor to randomize paths. To test interactions (i.e. Display*Size and Display*Order), a level was created for each unique combination of the variables in the interaction. All post-hoc tests utilized a Bonferroni Correction and the family significance level was $\alpha = 0.05$. Effect sizes for significant pairwise differences were measured with Hedges’ $g$.

Table 4.1: Non-parametric tests used to determine effects of select independent variables on non-spherical motion capture metrics

<table>
<thead>
<tr>
<th>Independent Variable</th>
<th>Variability</th>
<th>Levels</th>
<th>Statistical Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display</td>
<td>Within-Subjects</td>
<td>4</td>
<td>Friedman</td>
</tr>
<tr>
<td>Size</td>
<td>Within-Subjects</td>
<td>2</td>
<td>Wilcoxon Signed Rank</td>
</tr>
<tr>
<td>Order</td>
<td>Between-Subjects</td>
<td>2</td>
<td>Wilcoxon Rank Sum</td>
</tr>
<tr>
<td>Display*Size</td>
<td>Within-Subjects</td>
<td>8</td>
<td>Friedman</td>
</tr>
<tr>
<td>Display*Order</td>
<td>Between-Subjects</td>
<td>8</td>
<td>Kruskal-Wallis</td>
</tr>
</tbody>
</table>
4.2 Results

4.2.1 Subjective Measures

NASA TLX Workload Scores

There was not a significant effect of Display on the Overall Workload scores generated by the NASA TLX. The workload scores were unique to the participant and similar across all Display conditions in most cases (Fig. 4-4).

While there weren’t any significant effects on the Overall Workload scores, there was a significant effect of Display on Mental Workload \( (p=0.0022) \) and Temporal Workload \( (p=0.0118) \) (Table 4.2 and Figs. 4-5 and 4-6). Post-hoc tests suggest that Mental Workload was significantly higher during use of the TO display compared to the NC display \( (p=0.0009, g=1.45) \) and Temporal Workload was also significantly higher \( (p=0.0073, g=0.50) \) for the TO display compared to the NC display. While the Workload scores for VO and VT displays tend to fall between these extremes (Fig. 4-5), there were no statistically significant differences.

![Figure 4-4: NASA TLX Overall Workload scores for all sixteen subjects and display conditions](image-url)
Figure 4-5: NASA TLX Workload averages for all workload types and display conditions.

Figure 4-6: NASA TLX Workload scores for mental and temporal workload types, where shapes denote significance groupings. Black circles represent overall mean and standard deviation while grey circles represent individual participant means.
Table 4.2: Statistics from Friedman tests on the effect of display (d.f.=3) on NASA TLX Workload Scores, where asterisks denote significant p-values

<table>
<thead>
<tr>
<th>TLX Workload Type</th>
<th>Chi-Sq.</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall</td>
<td>6.36</td>
<td>.0954</td>
</tr>
<tr>
<td>Mental</td>
<td>14.61</td>
<td>.0022*</td>
</tr>
<tr>
<td>Physical</td>
<td>2.39</td>
<td>.4959</td>
</tr>
<tr>
<td>Temporal</td>
<td>10.98</td>
<td>.0118*</td>
</tr>
<tr>
<td>Performance</td>
<td>1.37</td>
<td>.7132</td>
</tr>
<tr>
<td>Effort</td>
<td>5.36</td>
<td>.1473</td>
</tr>
<tr>
<td>Frustration</td>
<td>3.48</td>
<td>.3230</td>
</tr>
</tbody>
</table>

Post-Experiment Survey Measures

On a scale from 1-5 that increases with intuitiveness, participants ranked the intuitiveness of the visual cues (M=3.88, SD=0.89) higher than that of the tactile cues (M=3.00, SD=1.03) with a Wilcoxon Signed Rank of 58.5 (p=0.0195, g=0.89). The Survey Workload scores (also on a scale from 1-5) had a significant effect of Display (Table 4.3, Fig. 4-7). Post-hoc tests suggested that participants ranked workload for the TO Display (M=3.75, SD=1.06) significantly higher than that for the VO Display (M=2.94, SD=1.18) (p=0.0394, g=3.71). There was not a significant effect of Order on intuitiveness ratings or workload rankings.

Table 4.3: Statistics from a Friedman test on Survey Workload scores

<table>
<thead>
<tr>
<th>Source</th>
<th>SS</th>
<th>d.f.</th>
<th>MS</th>
<th>Chi-Sq.</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display</td>
<td>14.22</td>
<td>6.493</td>
<td>4.73</td>
<td>9.82</td>
<td>.0202</td>
</tr>
<tr>
<td>Error</td>
<td>55.28</td>
<td>45</td>
<td>1.23</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Total</td>
<td>69.5</td>
<td>63</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
A Friedman test on Display Preference (ranking 1-4) also showed statistical significance (Table 4.4). Post-hoc tests suggest that the VO display was ranked significantly higher (i.e. more preferred) than the NC display ($p=0.004$, $g=3.78$) and the VT display was ranked significantly higher than the NC display ($p=0.004$, $g=3.78$). The rankings for the TO Display were not significantly different than those for the other display conditions. Figure 4-8 reflects histogram data of survey responses for the most preferred display, least preferred display, and most useful display.

Table 4.4: Statistical results from a Friedman test on display preference rankings

<table>
<thead>
<tr>
<th>Source</th>
<th>SS</th>
<th>d.f.</th>
<th>MS</th>
<th>Chi-Sq.</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display</td>
<td>19.19</td>
<td>63</td>
<td>6.40</td>
<td>11.66</td>
<td>.0087</td>
</tr>
<tr>
<td>Error</td>
<td>59.81</td>
<td>45</td>
<td>1.33</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Total</td>
<td>79</td>
<td>63</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
4.2.2 Motion Capture Metrics

Normalized Completion Time

A Repeated Measures ANOVA showed significant main effects of Subject, Location, Size, Display, Order, and Sex, along with significant interactions of Subject*Display and Display*Sex (Table 4.5). The full ANOVA table can be found in Appendix A (Fig. A.3). While Order, Sex, and the Display*Sex interaction do have significant p-values ($p<0.05$), Order and Sex are between-subject variables and therefore the effects on a normalized variable must be interpreted with caution. Post-hoc tests on the effect of Order suggest a higher average Normalized Completion Time for participants in Order B ($M=14.61\%$, $SD=9.55\%$) compared to participants in Order A ($M=11.86\%$, $SD=4.23\%$), but with a small effect size of $g=0.37$. Post-hoc tests on the effect of Sex suggest that females ($M=70.58\%$, $SD=14.63\%$) had a higher average Normalized Completion Time than males ($M=51.67\%$, $SD=20.77\%$), with a large effect size of $g=1.01$. 

Figure 4-8: Histogram data of participant responses to display preferences
Table 4.5: Significant effects on Normalized Completion Time

<table>
<thead>
<tr>
<th>Effect</th>
<th>F</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject</td>
<td>4.84</td>
<td>0.0001</td>
</tr>
<tr>
<td>Location</td>
<td>19.67</td>
<td>&lt;0.0001</td>
</tr>
<tr>
<td>Size</td>
<td>10.07</td>
<td>0.0073</td>
</tr>
<tr>
<td>Order</td>
<td>6.08</td>
<td>0.0297</td>
</tr>
<tr>
<td>Sex</td>
<td>11.55</td>
<td>0.0053</td>
</tr>
<tr>
<td>Display*Sex</td>
<td>3.06</td>
<td>0.0394</td>
</tr>
<tr>
<td>Subject*Display</td>
<td>9.75</td>
<td>&lt;0.0001</td>
</tr>
</tbody>
</table>

The main effect of Subject and the interaction between Subject and Display suggest that subjects were very different in their overall speed of trial completion across displays. This high variability can be recognized in Fig. A-1 in Appendix A, and further confirms the need for a Repeated Measures approach. While Location is a significant main effect, this was solely a blocking factor to generate a feeling of random walking trajectories in a controlled manner. Inferences on this parameter are not related to the driving hypotheses, but the factor was considered in the model to account for a known potential source of variance. As one might expect, participants had faster completion times for trials with the obstacle in Location 3 compared to trials with the obstacle in Location 1 ($p=0.0001$, $g=0.18$) or Location 2 ($p<0.0001$, $g=0.20$). This is likely due to the starting location for the Location 3 trials being slightly closer to the end of the motion capture space. Figure 4-2 depicts this characteristic of the experimental layout.

The main effect of Size suggests that participants finished trials with the small obstacle ($M=57.5$, $SD=20.43$) slightly quicker than those with the large obstacle ($M=60.02$, $SD=21.14$), although this was a small effect size of $g=0.12$ and may not be operationally relevant. Post-hoc tests on the main effect of Display (Fig. 4-9) suggest that Normalized Completion Time was significantly different between the TO and VO displays ($p<0.0001$, $g=0.50$), the TO and VT displays ($p=0.0017$, $g=0.20$), the TO and NC displays ($p<0.0001$, $g=1.14$), the VO and VT displays ($p=0.0004$, $g=0.26$),
the VO and NC displays \( p<0.0001, g=0.68 \), as well as the VT and NC displays \( p<0.0001, g=0.90 \). In short, all display conditions were significantly different from each other.

![Figure 4-9: Normalized Completion Time for all display conditions, where shapes denote significance groupings. Black circles represent overall mean and standard deviation while grey circles represent individual participant means (pooled across Order, Location, Size, and Sex).](image)

**Figure 4-9:** Normalized Completion Time for all display conditions, where shapes denote significance groupings. Black circles represent overall mean and standard deviation while grey circles represent individual participant means (pooled across Order, Location, Size, and Sex).

**Head-Down Percentage of Trial**

A Repeated Measures ANOVA supported main effects of Subject, Location, and Display on Head-Down Percentage of Trial. The significant effect of Subject and the interaction between Subject and Display show that subjects varied in their overall head-down time across displays, and this subject variability can be recognized in Fig. A-2 in Appendix A. The full ANOVA table can also be found in Appendix A (Fig. A.4).
Table 4.6: Significant effects on Head-Down Percentage of Trial

<table>
<thead>
<tr>
<th>Effect</th>
<th>F</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject</td>
<td>8.37</td>
<td>&lt;0.0001</td>
</tr>
<tr>
<td>Location</td>
<td>5.20</td>
<td>0.0126</td>
</tr>
<tr>
<td>Display</td>
<td>4.31</td>
<td>0.0102</td>
</tr>
<tr>
<td>Subject*Display</td>
<td>3.76</td>
<td>&lt;0.0001</td>
</tr>
</tbody>
</table>

Figure 4-10: Head-Down Percentage of Trial for all display conditions, where shapes denote significance groupings. Black circles represent overall mean and standard deviation while grey circles represent individual participant means (pooled across Order, Location, Size, and Sex).

While Location was included in the model to account for a potential source of variation, post-hoc tests on this effect suggest that participants spent a slightly lower percentage of time head-down during trials with the obstacle in Location 2 compared to trials with the obstacle at Location 1 ($p=0.0002$, $g=0.27$) or Location 3 ($p=0.0084$, $g=0.20$). Post-hoc tests on the main effect of Display suggest that Head-Down Percentage of Trial was significantly different between the TO and VO displays.
(p<0.0001, g=0.34), the TO and NC displays (p=0.0443, g=0.24), the VO and VT displays (p=0.0194, g=0.23), the VO and NC displays (p<0.0001, g=0.64), as well as the VT and NC displays (p<0.0001, g=0.37). In short, the TO and VT displays were in their own significance group (i.e. they aren’t significantly different from each other), and both the VO and NC displays were significantly different from this TO/VT grouping as well as each other (Fig. 4-10).

**Toe Clearance and Toe-Off Distance**

A non-parametric Wilcoxon Signed Rank test supported a significant effect of Size on Toe Clearance, where clearance was slightly lower for the large obstacle (M=6.84", SD=2.30") compared to the small obstacle (M=7.63", SD=2.30"), with a Signed Rank of 50771 and Z =6.35 (p<0.0001, g=0.33). A non-parametric Friedman test supported a significant effect of Display on Toe Clearance (Table 4.7). Bonferroni-corrected post-hoc tests do not show significance during multiple comparisons (all p>0.05), although it appears that the means for the display groups decrease from TO to VO to VT to NC (Fig. 4-11).

**Table 4.7: Statistics from a Friedman test on the effect of Display on Toe Clearance**

<table>
<thead>
<tr>
<th>Effect</th>
<th>SS</th>
<th>d.f.</th>
<th>MS</th>
<th>Chi-Sq.</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display</td>
<td>9434.41</td>
<td>3</td>
<td>3144.80</td>
<td>48.13</td>
<td>&lt;0.0001</td>
</tr>
<tr>
<td>Interactions</td>
<td>33062.63</td>
<td>45</td>
<td>734.73</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Error</td>
<td>104894.46</td>
<td>704</td>
<td>149.00</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Total</td>
<td>147391.50</td>
<td>767</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

A non-parametric Friedman test also supported a significant effect of Display on Toe-Off Distance (Table 4.8). Post-hoc tests support significant differences between the TO and NC displays (p=0.0132, g=1.35), the VO and NC displays (p=0.0436, g=1.18) as well as the VT and NC displays (p=0.0223, g=1.26). These results suggest that TO, VO, and VT displays are in their own significance grouping, and are significantly different from the NC condition (Fig. 4-11).
Table 4.8: Statistics from a Friedman test on the effect of Display on Toe-Off Distance

<table>
<thead>
<tr>
<th>Effect</th>
<th>SS</th>
<th>d.f.</th>
<th>MS</th>
<th>Chi-Sq.</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display</td>
<td>29650.36</td>
<td>3.00</td>
<td>9883.45</td>
<td>151.28</td>
<td>&lt;0.0001</td>
</tr>
<tr>
<td>Interactions</td>
<td>23358.31</td>
<td>45.00</td>
<td>519.07</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Error</td>
<td>94382.33</td>
<td>704.00</td>
<td>134.07</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Total</td>
<td>147391.00</td>
<td>767.00</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Figure 4-11: Toe Clearance and Toe-Off Distance for all display conditions, where shapes denote significance groupings. Black circles represent overall mean and standard deviation while grey circles represent individual participant means (pooled across Order, Location, Size, and Sex).

Step Characteristics

Non-parametric Friedman tests support a significant effect of Display on Number of Steps, Normalized Step Length, and Normalized Step Variance (Tables 4.9-4.11).
Table 4.9: Statistics from a Friedman test on the effect of Display on Number of Steps

<table>
<thead>
<tr>
<th>Effect</th>
<th>SS</th>
<th>d.f.</th>
<th>MS</th>
<th>Chi-Sq.</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display</td>
<td>47732.26</td>
<td>3</td>
<td>15910.75</td>
<td>248.21</td>
<td>&lt;0.0001</td>
</tr>
<tr>
<td>Interactions</td>
<td>24930.70</td>
<td>45</td>
<td>554.02</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Error</td>
<td>71949.04</td>
<td>704</td>
<td>102.20</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Total</td>
<td>144612.00</td>
<td>767</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 4.10: Statistics from a Friedman test on the effect of Display on Normalized Average Step Length

<table>
<thead>
<tr>
<th>Effect</th>
<th>SS</th>
<th>d.f.</th>
<th>MS</th>
<th>Chi-Sq.</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display</td>
<td>48989.91</td>
<td>3</td>
<td>16329.97</td>
<td>249.95</td>
<td>&lt;0.0001</td>
</tr>
<tr>
<td>Interactions</td>
<td>21950.46</td>
<td>45</td>
<td>487.79</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Error</td>
<td>76451.13</td>
<td>704</td>
<td>108.60</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Total</td>
<td>147391.50</td>
<td>767</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 4.11: Statistics from a Friedman test on the effect of Display on Normalized Step Length Variance

<table>
<thead>
<tr>
<th>Effect</th>
<th>SS</th>
<th>d.f.</th>
<th>MS</th>
<th>Chi-Sq.</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display</td>
<td>5547.14</td>
<td>3</td>
<td>1849.05</td>
<td>28.30</td>
<td>&lt;0.0001</td>
</tr>
<tr>
<td>Interactions</td>
<td>30219.48</td>
<td>45</td>
<td>671.54</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Error</td>
<td>111624.88</td>
<td>704</td>
<td>158.56</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Total</td>
<td>147391.50</td>
<td>767</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

For Number of Steps, post-hoc tests support significant differences between the TO and NC displays ($p=0.0001, g=1.00$), the VO and NC displays ($p=0.0137, g=0.74$), as well as the VT and NC displays ($p=0.0075, g=1.26$), where the NC trials resulted in a lower Number of Steps than trials for the other display conditions (Fig. 4-12). For Normalized Step Length, post-hoc tests also supported significant differences between the TO and NC displays ($p=0.0001, g=1.58$), the VO and NC displays ($p=0.0081,$
as well as the VT and NC displays \( (p=0.0039, g=1.33) \), where Normalized Step Length was larger during NC trials compared to the rest. Since Number of Steps and Normalized Step Length are related, these consistent statistical findings further support a change in stepping strategy across display type. For Normalized Step Variance, post-hoc tests did not show any significant comparisons, although it appears that trials during the NC condition had lower Normalized Step Variance than those for the display conditions (Fig. 4-12).

Figure 4-12: Step characteristics for all display conditions, where shapes denote significance groupings. Black circles represent overall mean and standard deviation while grey circles represent individual participant means (pooled across Order, Location, Size, and Sex).

Non-parametric Wilcoxon Rank Sum tests supported a significant effect of Order on both Number of Steps \( (p <0.0001, g=0.52) \) and Normalized Step Length \( (p <0.0001, g=0.36) \) (Table 4.12). Participants in Order B took significantly more steps than those in Order A. Normalized Step Length was significantly shorter for participants in Order B compared to those for Order A, which is consistent with the difference in Number of Steps.
Table 4.12: Statistics from Wilcoxon Rank Sum tests on the effect of Order on step characteristics

<table>
<thead>
<tr>
<th>Variable</th>
<th>Level</th>
<th>Mean</th>
<th>SD</th>
<th>Rank Sum</th>
<th>Z</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Steps</td>
<td>Order A</td>
<td>9.61</td>
<td>4.73</td>
<td>133321</td>
<td>-4.67</td>
<td>&lt;0.0001</td>
</tr>
<tr>
<td></td>
<td>Order B</td>
<td>13.74</td>
<td>12.73</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Normalized Step Length</td>
<td>Order A</td>
<td>52.07</td>
<td>22.54</td>
<td>163036</td>
<td>5.01</td>
<td>&lt;0.0001</td>
</tr>
<tr>
<td></td>
<td>Order B</td>
<td>43.84</td>
<td>23.72</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Non-parametric Wilcoxon Signed Rank tests supported a significant effect of Size on both Number of Steps ($p=0.0121, g=0.08$) and Normalized Step Length ($p=0.0147, g=0.08$) (Table 4.13). Participants took slightly more steps during trials with the large obstacle than they did during trials with the small obstacle, with an almost negligible effect size of $g=0.08$. Normalized Step Length was shorter for trials with the large obstacle compared to those with the small obstacle, with a small effect size of $g=0.36$. While significant, these small effects of Size on Number of Steps and Normalized Step Length are not considered operationally relevant differences.

Table 4.13: Statistics from Wilcoxon Signed Rank tests on the effect of Size on step characteristics

<table>
<thead>
<tr>
<th>Variable</th>
<th>Level</th>
<th>Mean</th>
<th>SD</th>
<th>Signed Rank</th>
<th>Z</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Steps</td>
<td>Small</td>
<td>11.28</td>
<td>8.96</td>
<td>22114</td>
<td>-2.51</td>
<td>&lt;0.0001</td>
</tr>
<tr>
<td></td>
<td>Large</td>
<td>12.07</td>
<td>10.60</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Normalized Step Length</td>
<td>Small</td>
<td>48.90</td>
<td>23.53</td>
<td>42272</td>
<td>2.44</td>
<td>0.0147</td>
</tr>
<tr>
<td></td>
<td>Large</td>
<td>47.01</td>
<td>23.43</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

All 16 participants reported right-foot dominance in a post-experiment survey. A significant effect of Display on dominant Lead Foot frequency was supported with a Chi-Square statistic of $X^2=8.39$ ($p=0.0385$). Pairwise comparisons suggested that the dominant foot was the Lead Foot more often during TO trials compared to NC trials ($p=0.0058, X^2=7.60$) and VT trials ($p=0.0321, X^2=4.59$). Contingency tables for this analysis are shown in Appendix A (Tables ??-A.2). Dominant Lead Foot
frequencies across Display and Order are displayed in Table 4.14.

**Table 4.14: Percentage of trials with a leading right foot for each Display and Order level**

<table>
<thead>
<tr>
<th></th>
<th>TO</th>
<th>VO</th>
<th>VT</th>
<th>NC</th>
<th>Order Means</th>
</tr>
</thead>
<tbody>
<tr>
<td>Order A</td>
<td>57.3%</td>
<td>39.6%</td>
<td>40.%</td>
<td>49.0%</td>
<td>46.6%</td>
</tr>
<tr>
<td>Order B</td>
<td>54.2%</td>
<td>55.2%</td>
<td>49.0%</td>
<td>34.4%</td>
<td>48.2%</td>
</tr>
<tr>
<td>Display Means</td>
<td>55.8%</td>
<td>47.4%</td>
<td>44.8%</td>
<td>41.7%</td>
<td></td>
</tr>
</tbody>
</table>

**Collisions**

A significant effect of obstacle Size on Collisions was supported with a Chi-Square statistic of $X^2=5.03 (p=0.0248)$, suggesting that there were more obstacle collisions during trials with the large obstacle. The continency table is shown in Table 4.15. Contingency tables and Chi-Square statistics did not support any differences for the other variables.

**Table 4.15: Contingency Table for obstacle Size and Collision frequency with (expected cell total) and [chi-square statistic] for each cell**

<table>
<thead>
<tr>
<th></th>
<th>Small Obstacle</th>
<th>Large Obstacle</th>
<th>Row Totals</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Collision</td>
<td>349 (339.00) [0.29]</td>
<td>329 (339.00) [0.29]</td>
<td>678</td>
</tr>
<tr>
<td>Collision</td>
<td>35 (45.00) [2.22]</td>
<td>55 (45.00) [2.22]</td>
<td>90</td>
</tr>
<tr>
<td>Column Totals</td>
<td>384</td>
<td>384</td>
<td>768 (Grand Total)</td>
</tr>
</tbody>
</table>

**4.3 Discussion**

This study examined the effect of cue display modality (visual, tactile, both in combination, or none), order of display presentation, and obstacle characteristics on a participant’s ability to step over an obstacle while minimizing head-down time (i.e. looking at the obstacle). To assess obstacle avoidance performance, the dependent variables of trial completion time, head-down time, toe clearance, toe-off distance, step characteristics, obstacle collisions, and subjective workload were measured.
Normalized Completion Time

For trial completion time, results suggest that completion time was dependent on the display, where times increased from NC to VO to VT to TO, each significantly different from each other. Since participants were essentially forced to look down at the obstacle in order to avoid it during NC trials, they selected the strategy of looking down and walking faster. Conversely, participants took more time to complete the trials involving cue presentation, suggesting that they may have paid attention to cue information as best as possible for the purpose of minimizing head-down time. During TO trials, 56% of participants (9 out of 16) reported that it took longer for them to confidently detect which tactile cue they were receiving, which could explain the longer completion times for that display type. This could also explain why the trials involving visual cues (VO and VT) had faster completion times, implying that visual cues may be processed quicker for complex mappings (i.e. mappings beyond a single alert cue). The completion times for the VT trials fell between those for the VO and TO trials, which suggests that the use of tactile cues in the multi-modal display may have slowed down processing time as a result of paying attention to both of the cue sensations.

Previous research comparing visual-only alerts to visual-tactile alerts in operational scenarios found quicker reaction times during the multi-modal conditions [20, 21, 22, 23], so it is likely that the current study may have imposed additional mental processing loads or perceptual limitations. Mental processing for tactile cues may have taken longer due to the novel, complex tactile mapping that participants were experiencing for the first time. A study examining the use of a vibrotactile sleeve for arm orientation implemented a complex mapping to convey joint angle errors, and participants reported that more conscious effort was required in tactile-only trials [36]. Another study found slower reaction times during the use of multi-modal cues compared to unimodal cues, and the researchers proposed that this was due to the unimodal cue only requiring one decision while the multi-modal cues required multiple decisions (i.e. one for each modality) [72]; this cognitive process of sensation-
to-mapping decision congruency between modalities is believed to be a component in longer reaction times. There may have also been physiological effects such as decreased vibrotactile perception, as previous research has found perception thresholds on the top of the foot to be significantly higher while walking compared to sitting [51]. Another factor that could have influenced longer processing times is participant confidence in cue comprehension; previous research has shown that participants often perform better than they think (i.e. rate themselves) in distinguishing different vibration signals [46]. It may take additional training and experience with a multi-cue tactile display to gain confidence in the vibration-to-information mapping. The significant effect of obstacle size on completion times suggests that participants may have proceeded with more caution during the trials with the large obstacle.

**Head-Down Percentage of Trial**

Results support that head-down time was highest during trials without cues, lowest during trials with only visual cues, and somewhere in the middle for trials with tactile only or multi-modal cues. The differences seen between the display conditions and NC condition suggest that the presence of any type of cue alters the strategy of the user. While there wasn't a significant difference between the head-down time for TO and VT trials as was the case with completion time, it appears that the overall head-down time trends are consistent with the completion time trends. There appears to be a trade-off between finishing quickly and looking down, as completions times were slowed when there was less time looking down. While tactile cues certainly help reduce head-down time, the addition of visual cues may not improve this metric as the VT head-down time mean was not significantly different. While tactile cues still reduce head-down time compared to no cues, it's possible that performance with visual cues alone may have reached a maximum, and there may be less potential for improvement with the addition of vibrotactile stimulation. This "ceiling effect" was observed in previous multi-modal research utilizing visual and tactile feedback to complete a cursor-tracking task [73]. The authors of the tracking study highlight the possibility of a Colavita effect [74], where responses are dominated by the visual channel during
discrimination tasks involving redundant multimodal feedback [72], and could be due to the higher information processing capability of vision. Increased head-down time during displays with tactile cues may have also been a result of decreased participant confidence in tactile cue comprehension. These results suggest that multi-modal displays should always employ the visual channel as the primary information processor. Once the visual channel is overloaded, the benefits of additional sensory cues are more apparent, and it is possible that participants in the current study were not visually burdened enough to offload attention to the foot vibrations.

**Toe Clearance and Toe-Off Distance**

The lower toe clearance for trials with the large obstacle seem appropriate considering that the obstacle height was twice that of the small obstacle, requiring more energy to clear than the small obstacle. While the vibrotactile boots did not convey information about obstacle height, participants could see the height of the obstacle while at the starting locations and could therefore prepare their gait accordingly. Post-hoc tests on the effect of display did not support any significant comparisons for toe clearance, but the trend between display conditions for head-down time may be consistent with the trends seen for toe clearance. Previous obstacle avoidance research [57] demonstrated that toe clearance increases while wearing basketball goggles that cut off the lower peripheral vision, implying that people increase toe clearance when visual information is reduced. An increase in head-down time suggests that participants were looking at the obstacle more, which may have allowed them to feel more confident during step-over and reduce clearance. This behavior likely took place in the current study, as clearance for the NC trials appears to be much less than that for the display conditions and there is also evidence that they spent more time looking down during these trials.

The significant difference in toe-off distance between the cue displays and the NC condition is most likely due to the presence of Cue 3, which signified that a boot was in the optimal 2-8 inch toe-off range. The average toe-off distance during NC trials (M=15.64", SD=7.04") was almost twice that of the cue display conditions, and
far beyond the optimal step-off range required for a steady obstacle step-over with low risk of collision. The majority of toe-off distances across cue display conditions was between 5-10 inches, suggesting that participants were successfully using the cues to determine when to step over the obstacle. As soon as one or both boots were in the optimal step-off range, participants chose which foot to step over the obstacle with - and sometimes this was the trailing foot, which could explain why toe-off distance sometimes occurred just beyond the optimal range (i.e. greater than 8 inches) but still stayed significantly shorter than in the NC trials. It is important to note that while toe-off distances varied between the cue display conditions and the NC condition (a metric related to collision risk), results did not support a significant effect of Display on collision frequency. This suggests that participants changed their obstacle avoidance strategy according to the display condition and did so in a way that maintained adequate task performance.

**Step Characteristics**

Participants took longer steps during the NC trials compared to the cue display trials, which also resulted in fewer steps for those trials. This result makes sense considering that participants were also fastest during the NC trials. Participants also had significantly lower step length variability during the NC trials compared to the cue display trials, which has implications for balance. Previous research \[75, 76, 77, 60\] has shown that increased step length variability is associated with better balance during walking. This suggests participants may have maintained better balance during obstacle avoidance for the cue display trials. This is not surprising considering that they were also walking slower, taking smaller steps, increasing toe clearance, and stepping over at optimal toe-off distances during cue display use. In short, results show that participants had a more conservative gait pattern during trials with the cue displays.

The significant effect of Order on step length and frequency is interesting; participants in Order A (NC, VO, TO, VT) took fewer steps and longer steps than those in Order B (VT, VO, TO, NC). This may be a consequence of starting with NC trials,
since longer and less frequent steps are associated with that display condition. It is possible that participants established a specific walking style during their first set of NC trials and attempted to maintain this throughout the rest of the trials. Conversely, participants starting with the novel and sometimes overloading VT display may have began their trials slowly and cautiously, setting this as the standard for the rest of their trials. While the significant effect of the between-subject factor of Order on Normalized Completion Time must be interpreted with caution, this result may suggest this exact behavior - steadier and more consistent times across trials for participants in Order B but larger within-subject completion time discrepancies for those in Order A. There was not a significant effect of Order on workload rankings for the displays or cue intuitiveness scores, so the difference in strategy was likely unrelated to these factors.

**Workload and Display Preferences**

While the Overall Workload scores generated by the NASA TLX did not show any significant differences, the Post-Experiment Survey Workload scores did, with participants rating the workload of the TO display significantly higher than the workload of the NC condition. The significant difference between the TO display and NC condition for NASA TLX Mental Workload and Temporal Workload scores may provide insight on these Survey Workload differences. While participants were told not to worry about speed or completion time, they moved slowest during TO trials, which could explain the difference in Temporal Workload. This could potentially be a result of higher Mental Workload as slower processing time could slow the gait and decision-making process, which was likely the case in the TO trials for the aforementioned reasons. The intuitiveness of the cues was ranked significantly higher for the visual cues compared to the tactile cues, which seems plausible considering that the tactile modality is not often used for information display. It is important to consider that the tactile display was a novel form of information mapping for the participants, as haptic communication devices are relatively new and uncommon compared to visual displays. The processing time required to establish new vibration-to-information
connections and perceive the new tactile sensations could have been enough to slow down participants’ processing of the tactile cues. Moreover, participants may have experienced varying levels of confidence in the tactile cues throughout the learning process.

The vast majority of participants preferred the VO or VT display and found it most useful for obstacle avoidance, a result that suggests that they felt the displays were helpful in avoiding obstacles while keeping their head up even though they had to move slower and more cautiously. The finding that most participants ranked the NC condition as their last preference demonstrates that walking in clunky, rigid boots during limited vision while having to avoid obstacles is not an enjoyable activity, albeit one that future astronauts will have to perform regularly during planetary exploration and surface operations.

Summary

It was hypothesized that the presentation of obstacle proximity cues to visual and tactile modalities alone and in combination would elicit different effects on a participant’s head-up obstacle avoidance capability. The experimental data support a significant decrease in head-down time when visual cues, tactile cues, or a combination of the two are presented. The data also show that the use of display cues slows participants down and promotes a more conservative gait pattern. Displays with the tactile cues had more head-down time and longer completion times compared to the visual display, and this could have been a result of increased mental processing time, perceptual limitations/ambiguities of vibrations during walking, and/or cue comprehension confidence; nonetheless, many of the participants preferred the VT display, demonstrating that tactile cues were not necessarily a hindrance. Additional training and vibration-to-information mapping reinforcement may have been needed to elicit the same performance seen with the VO display.

The use of display cues also significantly decreases toe-off distance to a more optimal range before stepping over an obstacle and may also increase toe clearance. Interestingly, the order of display presentation affected participants’ step frequency
and length throughout the trials, which may suggest an attempt to maintain the pace of the first set of trials (either VT or NC). Compared to the small obstacle, trials with the large obstacle slightly increased step frequency and decreased step length in the associated trials, as well as resulted in more collisions, less toe clearance, and longer completion times. These characteristics are most likely associated with the obstacle height and are not considered operationally relevant. Subjective data reveals that most participants preferred the displays with visual cues (VO or VT) over having no cues, and thought those displays were useful. Overall, the results show that the use of obstacle proximity cues during limited lower vision allows participants to successfully avoid obstacles while minimizing head-down time and can promote optimal foot placement compared to nominal scenarios without cues.
Chapter 5

Conclusions

The primary objective of the current work was to examine the effectiveness of a visual-tactile display for a surface obstacle avoidance scenario. While the experimental environment was simplified, results can help inform design considerations for astronaut planetary exploration tasks during future missions to Mars. A wearable multi-modal display device incorporating visual cues and tactile cues to convey obstacle location and proximity in real-time was developed. A preliminary study evaluating the vibrotactile perception capabilities of the feet was performed first in order to develop requirements for the tactile information mapping for the device. Human subjects research in a motion capture space was performed to evaluate the differences in obstacle avoidance performance and workload during use of unimodal information displays, a multi-modal information display, and no display.

The most operationally relevant measures of performance included obstacle collision frequency and head-down time, where less head-down time was associated with better performance. These metrics were relevant to the application of interest (a suited astronaut navigating rocky terrain on Mars) since astronauts will need to keep their attention and focus on completing EVA tasks as opposed to keeping their focus on the ground. The main hypothesis was that real-time obstacle location and proximity information could improve obstacle avoidance performance during limited lower peripheral vision. Blocking the peripheral vision of the lower limbs was necessary in order to simulate perceptual circumstances of a suited astronaut on Mars. It was
also hypothesized that the use of multi-modal cues (i.e. visual-tactile) might result in better obstacle avoidance performance compared to unimodal (i.e. tactile only or visual only) cues, and that any form of cue presentation might increase performance over having no cues at all. To assess differences in strategy, gait metrics included toe clearance, toe-off distance, number of steps, normalized average step length, and normalized step variance. Workload was subjectively reported through the NASA Task Load Index (TLX) and a post-experiment survey.

For each experimental trial, priorities were 1) to avoid the obstacle, and 2) to minimize head-down time. The data do not support a significant effect of display on obstacle collision frequency, suggesting that participants successfully accomplished their primary task across all display conditions. Differences in performance were mainly seen in the secondary task of reducing head-down time. The experimental data support a significant decrease in head-down time when obstacle location and proximity cues are provided. The least head-down time occurred during use of the visual-only display, with a 44% decrease on average from having no display. The visual-tactile display decreased head-down time by 27% on average and the tactile-only display decreased head-down time by 15% on average, although head-down times for these two displays were not significantly different. These results suggest that while multi-modal cues effectively increase task performance, the visual unimodal cues induce the best performance. The tactile unimodal cues did not increase performance over visual unimodal cues or visual-tactile multimodal cues. A meta-analysis of multisensory display research found that replacing visual cues with tactile cues produced mixed results of performance that relate to cue information complexity; tactile alerts can successfully replace visual alerts, but tactile direction cues cannot fully replace visual direction cues [16]. Data from the current study further support this phenomenon; while the cues weren’t necessarily direction cues, they were considered to be of the same complexity as direction cues (i.e. information promoting a change in body motion as opposed to a simple interrupting alert during a seated task).

The data also show that the use of display cues slowed participants down. Trends in completion time are consistent with those seen in head-down time, where increased
head-down time was associated with quicker completion times. This suggests that there may have been a trade-off between speed and looking down while avoiding obstacles. The visual-only display increased completion time by 32% on average from no display, the visual-tactile increased completion time by 41% on average, and the tactile-only display increased completion time by 49% on average. Completion time for each display was significantly different from the others. Since participants were told to take as much time as needed throughout the trials, completion time was not considered a performance metric, but these results do suggest that participants acquired a different strategy throughout display trials.

Experimental data also support a change in gait strategy during use of the various displays. The use of display cues significantly decreased toe-off distance to a more optimal range before stepping over an obstacle and may have also increased toe clearance. The display cues also caused participants to take shorter steps, resulting in more steps throughout these trials. The display cues may have also decreased step length variance, indicating that participants were more balanced while using the displays (significant omnibus test but non-significant pairwise comparisons). Overall, these differences in step characteristics suggest that participants had a more conservative gait pattern during trials with the displays. This is relevant to the finding that they were looking down less, as reduced lower vision may have affected confidence and increased caution.

Subjective data from surveys reveal that most participants preferred the displays with visual cues (visual-only or visual-tactile) over having no cues, and thought those displays were useful. NASA TLX workload scores suggest that the tactile-only display induced higher mental and temporal workloads compared to having no display. The increased mental workload of the tactile-only display may have implications for the performance trends, notably that tactile-only trials resulted in more head-down time and longer completion times compared to the other displays. The vibration-to-information mapping may have been complex enough to increase cognitive processing loads and/or decrease confidence in perception and comprehension abilities. Moreover, participants ranked the visual cues as more intuitive than the tactile cues.

85
Overall, results suggest that the use of obstacle location and proximity information cues during limited lower vision allows participants to avoid obstacles while minimizing head-down time, promotes a more conservative gait, and can guide optimal foot placement compared to scenarios without information display.

5.1 Implications for Multi-Modal Information Presentation Devices

5.1.1 Astronauts

In the case of future astronauts navigating rocky terrain during an EVA on the surface of Mars, it is important for them to be able to keep their visual focus at the destination ahead or mission task at hand. If astronauts are cautiously locomoting while keeping their visual focus on the ground, workload may increase and mission timelines may suffer. On the other hand, if an astronaut is attentionally narrowed with tasks at hand and not thinking about obstacles, the risk of falling and damaging life-support equipment may increase. The current body of work supports the notion that astronauts could benefit from built-in interfaces inside the suit to help with obstacle avoidance. Providing cue information about obstacle location and proximity resulted in successful obstacle avoidance with less head-down time in the laboratory setting, and this behavior might extend to a suited scenario on the rocky, Martian terrain.

While the current study did not find better performance with the multi-modal display compared to the visual-only display, the utilization of multiple sensory channels in a wearable interface should still be considered. Participants were still able to use visual-tactile cues and tactile cues alone to avoid obstacles and minimize head-down time. Further, the effect on gait was very similar between the various displays (i.e. more conservative). Multi-modal information may elicit different effects on response and performance during physically active tasks compared to seated monitoring and control tasks. The vast majority of research utilizing visual and tactile cues for a
locomotive task involves the use of vibrotactile vests/belts for navigation purposes [38, 29, 39, 40, 41]. While the feet were thought to be more intuitive for obstacle avoidance, a tactile vest or belt may be more complementary to visual cues in physical tasks since tactile sensations are uncommon here. Additional multi-modal configurations should be assessed (e.g. different vibration locations or patterns) to better understand the implications of design decisions for this application. Additionally, performance may have been better with the multi-modal display compared to the visual-only display if workload was higher on the visual channel. Future astronauts will likely have head-up displays embedded in their helmet assembly, and a lot of other pertinent information may be displayed (e.g. checklists and equipment statuses), leaving little space and salience for obstacle-specific warning cues. In such a case of visual clutter, the tactile cues may actually offer more benefit. Moreover, extensive training on a novel tactile display may be an important element in improving cue perception, comprehension, and confidence. Aside from obstacle avoidance, vibrotactile cues on the feet may benefit astronaut performance in other tasks. Miller et al. [14] found that astronauts are prone to tripping post-flight due to decreased toe clearance during mid-swing in the gait, so tactile cues on the front of the foot could also be used to elevate clearance and minimize tripping during reintroduction to walking after long duration space flight.

5.1.2 Other Applications

The results from the current work can also inform multi-modal device design for other applications that involve operation during limited vision and mobility. The same visual and tactile cue logic could be implemented for similar devices to aid firefighters in smoke-filled rooms, visually impaired individuals, and heavily-equipped soldiers in the dark. Variations of the information mapping for the specific application would likely be needed. Individuals with impaired vision have particular difficulty in discerning distance to objects [78], so providing object proximity information may aid walking and way-finding for these individuals. While multi-modal devices can augment and/or enhance perceptual and cognitive capabilities, they also have great
potential for use in sensory substitution applications (e.g. tactile-audial cues for a blind person or visual-tactile cues for a deaf person).

5.1.3 Takeaways

In conclusion, the following takeaways can be used to guide future multi-modal displays for obstacle avoidance:

- Obstacle location and proximity cues enable participants to avoid obstacles while reducing head-down time

- Obstacle location and proximity cues guide optimal foot placement (i.e. toe-off distance) before stepping over an obstacle

- Obstacle location and proximity cues promote a slower and more conservative gait strategy during obstacle avoidance

- The visual-only cues evoked the lowest head-down time compared to the tactile-only and multi-modal cues

- The tactile-only cues required higher mental and temporal workload than having no display

- The tactile cues on the feet were perceived as less intuitive than the visual cues on the augmented reality display

- The visual-tactile and visual-only cues were preferred over the tactile-only cues and having no display by most participants

It’s important to note that many of these results may be specific to the chosen information mapping for this work.

5.2 Limitations

It is important to highlight some limitations of this work. Participants often differed in their cue perception capabilities and this created differences in preference.
Many participants reported a preference for the visual display due to difficulty in consistently perceiving the tactile cues, while some participants preferred the tactile display because it felt less distracting. Hancock et al. [26] highlights that individual differences in tactile perception and “perceptual style” create challenges for novel information mappings; despite the designer’s intended message, the user’s interpretation may differ. The vibration perception study outlined in Chapter 2 suggested that user ability in perceived vibration accuracy may vary by vibration type and attentional loading. The obstacle avoidance task provided a cognitive load through the use of the novel tactile and visual information mappings as well as through locomoting through the environment. These additional load factors may have been enough to degrade perception of the vibrational cues while walking and decrease confidence in cue comprehension. It would be useful to better understand how vibratory thresholds of the feet change during walking and other physically active tasks.

Sarter et al. [79] posits that crossmodal matching is a critical step in multi-modal information display design, an activity that refers to equating perceived intensities of stimuli across the sensory modalities. The current study did not perform crossmodal matching between the tactile and visual cues, which could have caused confounding differences in cue salience and perception. As there is currently no agreed-upon matching technique, Sarter emphasizes a need for standardizing crossmodal matching procedures in future studies.

Participants had limited exposure to the display technology, and were required to learn how to utilize the information cues in a short period of time. The feet are not often used for vibrotactile displays, so participants may have benefited from more extensive training and experience with the system to enforce the unique signal-to-information mapping. Additional practice with the system could elicit different performance characteristics over time. It is also important to consider that participants did not have the same mobility impediments as a full-suited astronaut would have from a pressurized, bulky space suit. It would be useful to carry out a similar study that could replicate this physical workload, as it could provide further insight into how the cue information is used during a more realistic EVA scenario. Addi-
tionally, measures of situational awareness or secondary task performance could be recorded as additional performance metrics with implications for multi-modal display use strategies. It is possible that the cue timing and level logic would have to be altered for a full-suited scenario, as locomotion would be slower and more deliberate. Moreover, the use of a realistic head-up display presenting additional information and increasing visual workload may evoke differences in performance.

5.3 Future Work

It would be useful to examine different information mappings for the vibrotactile cues on the feet. Rather than relying solely on perceived vibration intensity and pulse frequency, more signal characteristics could be added to create unique temporal patterns (e.g. differing pulse lengths and intervals). There may be an information mapping that feels much more intuitive to the majority of users. It would also be useful to examine other body parts for the tactile display - while the feet seemed relevant for obstacle avoidance, the waist, wrist, or thigh may end up being a better choice. The ideal vibration location should relate to task demands, have low inter-subject variability, be sensitive enough for robust cue salience, and have a low probability of interfering with task demands.

It would also be useful to examine how performance changes when there is a cluttered head-up display that induces a higher visual workload, as this would be more similar to a potential scenario on Mars. In this circumstance, the tactile cues or visual-tactile cues may increase performance over the visual-only cues. Additionally, it would be advantageous to develop and assess multi-modal interfaces for other obstacle avoidance strategies, such as walking around the obstacle. In a realistic scenario, the strategies of stepping over and walking around would both be adopted. The vibrotactile boots could support the walk-around strategy if equipped with additional sensors and tactors on the lateral and medial sides of the boot.
5.4 Research Contributions

The majority of multi-modal information display studies integrating the visual and tactile channels have simulated operational tasks that involve sitting or standing in a quiet laboratory environment. Several studies have examined multi-modal information for the physically active task of directional navigation, but none thus far have assessed information display for obstacle avoidance. The current body of work provides insight on how multi-modal and unimodal information displays affect the performance of the physically active task of obstacle avoidance, a scenario requiring real-time cue perception, comprehension, and precise corrective actions while locomoting through the environment. While tactile interfaces have grown in popularity over the past decade, little work has been done specifically with the feet, a stimuli location with great potential for improving performance on tasks that require precise gait corrections. The current work contributes insight on vibrotactile perception and cue comprehension on the feet, although there is still a lot of work that needs to be done in determining optimal information mappings for tactile cues on the feet. Overall, results from the current body of work can help inform aspects of wearable multi-modal interface design for aiding obstacle avoidance during restricted lower peripheral vision; notably, that both visual and tactile unimodal and multimodal cues are capable of reducing head-down time during obstacle avoidance, but additional visual workloads and/or improvements in tactile information mappings may be necessary to see an increase in performance with multi-modal cues over visual unimodal cues.
Appendix A

Additional Data

Table A.1: Contingency table containing frequency data for lead foot between TO and VT displays

<table>
<thead>
<tr>
<th></th>
<th>TO</th>
<th>VT</th>
<th>Row Totals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Right Lead</td>
<td>107 (96.50)</td>
<td>86 (96.50)</td>
<td>193</td>
</tr>
<tr>
<td>Left Lead</td>
<td>85 (96.50)</td>
<td>106 (95.50)</td>
<td>191</td>
</tr>
<tr>
<td>Column Totals</td>
<td>192</td>
<td>192</td>
<td>384 (Grand Total)</td>
</tr>
</tbody>
</table>

Table A.2: Contingency table containing frequency data for lead foot between TO and NC displays

<table>
<thead>
<tr>
<th></th>
<th>TO</th>
<th>NC</th>
<th>Row Totals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Right Lead</td>
<td>107 (93.50)</td>
<td>80 (93.50)</td>
<td>187</td>
</tr>
<tr>
<td>Left Lead</td>
<td>85 (98.50)</td>
<td>112 (98.50)</td>
<td>197</td>
</tr>
<tr>
<td>Column Totals</td>
<td>192</td>
<td>192</td>
<td>384 (Grand Total)</td>
</tr>
</tbody>
</table>
Table A.3: Repeated Measures ANOVA table for Normalized Completion Time, where asterisks denote significant p-values

<table>
<thead>
<tr>
<th>Source</th>
<th>SS</th>
<th>d.f.</th>
<th>MS</th>
<th>F</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject</td>
<td>54483.67</td>
<td>12</td>
<td>4540.31</td>
<td>4.84</td>
<td>0.0001*</td>
</tr>
<tr>
<td>Location</td>
<td>3570.02</td>
<td>2</td>
<td>1785.01</td>
<td>19.67</td>
<td>&lt;0.0001*</td>
</tr>
<tr>
<td>Size</td>
<td>1253.44</td>
<td>1</td>
<td>1253.44</td>
<td>10.07</td>
<td>0.0073*</td>
</tr>
<tr>
<td>Display</td>
<td>42646.85</td>
<td>3</td>
<td>14215.62</td>
<td>15.61</td>
<td>&lt;0.0001*</td>
</tr>
<tr>
<td>Order</td>
<td>27603.43</td>
<td>1</td>
<td>27603.43</td>
<td>6.08</td>
<td>0.0297*</td>
</tr>
<tr>
<td>Sex</td>
<td>52428.11</td>
<td>1</td>
<td>52428.11</td>
<td>11.55</td>
<td>0.0053*</td>
</tr>
<tr>
<td>Subject*Location</td>
<td>2359.87</td>
<td>26</td>
<td>90.76</td>
<td>0.97</td>
<td>0.505</td>
</tr>
<tr>
<td>Subject*Size</td>
<td>1618.19</td>
<td>13</td>
<td>124.48</td>
<td>1.33</td>
<td>0.1883</td>
</tr>
<tr>
<td>Subject*Display</td>
<td>35510.12</td>
<td>39</td>
<td>910.52</td>
<td>9.75</td>
<td>&lt;0.0001*</td>
</tr>
<tr>
<td>Location*Size</td>
<td>114.01</td>
<td>2</td>
<td>57.01</td>
<td>0.61</td>
<td>0.5434</td>
</tr>
<tr>
<td>Location*Display</td>
<td>756.77</td>
<td>6</td>
<td>126.13</td>
<td>1.35</td>
<td>0.2324</td>
</tr>
<tr>
<td>Location*Order</td>
<td>305.45</td>
<td>2</td>
<td>152.73</td>
<td>1.68</td>
<td>0.2055</td>
</tr>
<tr>
<td>Location*Sex</td>
<td>286.33</td>
<td>2</td>
<td>143.16</td>
<td>1.58</td>
<td>0.2257</td>
</tr>
<tr>
<td>Size*Display</td>
<td>298.91</td>
<td>3</td>
<td>99.64</td>
<td>1.07</td>
<td>0.3624</td>
</tr>
<tr>
<td>Size*Order</td>
<td>16.2</td>
<td>1</td>
<td>16.2</td>
<td>0.13</td>
<td>0.7241</td>
</tr>
<tr>
<td>Size*Sex</td>
<td>0.27</td>
<td>1</td>
<td>0.27</td>
<td>0</td>
<td>0.9636</td>
</tr>
<tr>
<td>Display*Order</td>
<td>4672.41</td>
<td>3</td>
<td>1557.47</td>
<td>1.71</td>
<td>0.1807</td>
</tr>
<tr>
<td>Display*Sex</td>
<td>8356.65</td>
<td>3</td>
<td>2785.55</td>
<td>3.06</td>
<td>0.0394*</td>
</tr>
<tr>
<td>Order*Sex</td>
<td>682.9</td>
<td>1</td>
<td>682.9</td>
<td>0.15</td>
<td>0.7049</td>
</tr>
<tr>
<td>Error</td>
<td>60229.73</td>
<td>645</td>
<td>93.38</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Total</td>
<td>309508.97</td>
<td>767</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Table A.4: Repeated Measures ANOVA table for Head-Down Percentage of Trial, where asterisks denote significant p-values

<table>
<thead>
<tr>
<th>Effect</th>
<th>SS</th>
<th>d.f.</th>
<th>MS</th>
<th>F</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject</td>
<td>83907.00</td>
<td>12</td>
<td>6992.25</td>
<td>8.37</td>
<td>&lt;0.0001*</td>
</tr>
<tr>
<td>Location</td>
<td>2905.93</td>
<td>2</td>
<td>1452.96</td>
<td>5.20</td>
<td>0.0126*</td>
</tr>
<tr>
<td>Size</td>
<td>11.60</td>
<td>1</td>
<td>11.60</td>
<td>0.05</td>
<td>0.8196</td>
</tr>
<tr>
<td>Display</td>
<td>9439.85</td>
<td>3</td>
<td>3146.62</td>
<td>4.31</td>
<td>0.0102*</td>
</tr>
<tr>
<td>Order</td>
<td>1567.66</td>
<td>1</td>
<td>1567.66</td>
<td>0.22</td>
<td>0.6444</td>
</tr>
<tr>
<td>Sex</td>
<td>14.86</td>
<td>1</td>
<td>14.86</td>
<td>0.00</td>
<td>0.9640</td>
</tr>
<tr>
<td>Subject*Location</td>
<td>7269.26</td>
<td>26</td>
<td>279.59</td>
<td>1.44</td>
<td>0.0746</td>
</tr>
<tr>
<td>Subject*Size</td>
<td>2783.66</td>
<td>13</td>
<td>214.13</td>
<td>1.10</td>
<td>0.3544</td>
</tr>
<tr>
<td>Subject*Display</td>
<td>28502.90</td>
<td>39</td>
<td>730.84</td>
<td>3.76</td>
<td>&lt;0.0001*</td>
</tr>
<tr>
<td>Location*Size</td>
<td>335.48</td>
<td>2</td>
<td>167.74</td>
<td>0.86</td>
<td>0.4225</td>
</tr>
<tr>
<td>Location*Display</td>
<td>1245.76</td>
<td>6</td>
<td>207.63</td>
<td>1.07</td>
<td>0.3804</td>
</tr>
<tr>
<td>Location*Order</td>
<td>574.43</td>
<td>2</td>
<td>287.21</td>
<td>1.03</td>
<td>0.3721</td>
</tr>
<tr>
<td>Location*Sex</td>
<td>448.52</td>
<td>2</td>
<td>224.26</td>
<td>0.80</td>
<td>0.4592</td>
</tr>
<tr>
<td>Size*Display</td>
<td>905.61</td>
<td>3</td>
<td>301.87</td>
<td>1.55</td>
<td>0.1998</td>
</tr>
<tr>
<td>Size*Order</td>
<td>543.05</td>
<td>1</td>
<td>543.05</td>
<td>2.54</td>
<td>0.1353</td>
</tr>
<tr>
<td>Size*Sex</td>
<td>3.10</td>
<td>1</td>
<td>3.10</td>
<td>0.01</td>
<td>0.9060</td>
</tr>
<tr>
<td>Display*Order</td>
<td>1581.23</td>
<td>3</td>
<td>527.08</td>
<td>0.72</td>
<td>0.5454</td>
</tr>
<tr>
<td>Display*Sex</td>
<td>1845.27</td>
<td>3</td>
<td>615.09</td>
<td>0.84</td>
<td>0.4794</td>
</tr>
<tr>
<td>Order*Sex</td>
<td>1143.63</td>
<td>1</td>
<td>1143.63</td>
<td>0.16</td>
<td>0.6930</td>
</tr>
<tr>
<td>Error</td>
<td>125413.55</td>
<td>645</td>
<td>194.44</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Total</td>
<td>274616.28</td>
<td>767</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Figure A-1: Normalized Completion Time for all sixteen subjects and display conditions

Figure A-2: Head-Down Percentage of Trial for all sixteen subjects and display conditions
Appendix B

Computer Code

B.1 Vibration Perception Study Arduino UNO C Code

```c
#include <Wire.h>
#include "Adafruit_DRV2605.h"
#include <stdio.h>

Adafruit_DRV2605 drv;

// ----------------- PWM PINS ----------------
int VibeM1 = 3;
int VibeM2 = 5;
int VibeM3 = 6;
int VibeM4 = 9;
int VibeM5 = 10;
int VibeM6 = 11;

int vibe_locations[7] = {VibeM1, VibeM1, VibeM2, VibeM3, 
                         VibeM4, VibeM5, VibeM6};

int n_vibes = 6;
char buff[128];
int location_num = 0;

void setup() {
    Serial.begin(9600);
    Serial.println("Welcome to the MIPOA Vibe Test!");
    drv.begin();
    drv.selectLibrary(1);
```

97
pinMode(VibeM1, OUTPUT);
pinMode(VibeM2, OUTPUT);
pinMode(VibeM3, OUTPUT);
pinMode(VibeM4, OUTPUT);
pinMode(VibeM5, OUTPUT);
pinMode(VibeM6, OUTPUT);
analogWrite(VibeM1, 127);
analogWrite(VibeM2, 127);
analogWrite(VibeM3, 127);
analogWrite(VibeM4, 127);
analogWrite(VibeM5, 127);
analogWrite(VibeM6, 127);

drv.go();
Serial.print("Starting...");
}

uint8_t counter = 1;

//-----------------VIBRATION TYPES
//-----------------

// PWM input of 0-255 representing duty cycle where 127-255 (0-100%) different
// direction than 127-0 (0-100%)

// HIGH
void high(int VibeM) {

drv.setMode(DRV2605_MODE_PWMANALOG);
Serial.print("Vibe level: HIGH \n");
analogWrite(VibeM, 240);
delay(1500);
analogWrite(VibeM, 127); // turn off motor
drv.setMode(DRV2605_MODE_INTTRIG);
delay(1000);
}

// LOW
void low(int VibeM) {

drv.setMode(DRV2605_MODE_PWMANALOG);
Serial.print("Vibe level: LOW \n");
analogWrite(VibeM, 160);
delay(1500);
analogWrite(VibeM, 127); // turn off motor
drv.setMode(DRV2605_MODE_INTTRIG);
delay(1000);
}
// INCREASE
void increase(int VibeM) {
    drv.setMode(DRV2605_MODE_PWMANALOG);
    Serial.print("Vibe level: INCREASE \n");
analogWrite(VibeM, 150);
delay(250);
analogWrite(VibeM, 170);
delay(250);
analogWrite(VibeM, 190);
delay(250);
analogWrite(VibeM, 210);
delay(250);
analogWrite(VibeM, 230);
delay(250);
analogWrite(VibeM, 245);
delay(250);
analogWrite(VibeM, 127); // turn off motor
    drv.setMode(DRV2605_MODE_INTTRIG);
delay(1000);
}

// DECREASE
void decrease(int VibeM) {
    drv.setMode(DRV2605_MODE_PWMANALOG);
    Serial.print("Vibe level: DECREASE \n");
analogWrite(VibeM, 245);
delay(250);
analogWrite(VibeM, 230);
delay(250);
analogWrite(VibeM, 210);
delay(250);
analogWrite(VibeM, 190);
delay(250);
analogWrite(VibeM, 170);
delay(250);
analogWrite(VibeM, 150);
delay(250);
analogWrite(VibeM, 127); // turn off motor
    drv.setMode(DRV2605_MODE_INTTRIG);
delay(1000);
}

// DEMO
void demo() {

Serial.print("You are about to feel all four types of vibration at each foot location in the following order: High, Low, Increase, Decrease 
");

delay(2000);
for (int i = 1; i <= n_vibes; i++) {
  sprintf(buff, "Now testing vibe motor %d \n", i);
  Serial.print(buff);
  delay(1000);
  high(vibe_locations[i]);
  delay(2000);
  low(vibe_locations[i]);
  delay(2000);
  increase(vibe_locations[i]);
  delay(2000);
  decrease(vibe_locations[i]);
  delay(2000);
}
Serial.print("Demo Finished");
}

void from_serial() {
  if (Serial.available()) {
    char ch = Serial.read();
    Serial.println(ch);
    if (isdigit(ch)) {
      location_num = ch - '0';
    } else if (ch == 'H') {
      delay(random(3000, 7000));
      high(vibe_locations[location_num]);
    } else if (ch == 'L') {
      delay(random(3000, 7000));
      low(vibe_locations[location_num]);
    } else if (ch == 'I') {
      delay(random(3000, 7000));
      increase(vibe_locations[location_num]);
    } else if (ch == 'D') {
      delay(random(3000, 7000));
      decrease(vibe_locations[location_num]);
    } else if (ch == 'E') {
      exit(0);
    } else if (ch == 'X') {
      demo();
    }
  }
}
B.2 Vibration Perception Study Python GUI Code

```python
import Tkinter
import random
import datetime
import time
import serial
from random import randint

class ExperimentData:
    def __init__(self, start_trial=0, filename='exp Trials.txt'):
        # 3 sets of 48 randomized trials mirrored, making 6 sets of 48 trials total
        self.locations = []
        self.vibetypes = []
        self.recordcount = start_trial
        self.start_trial = start_trial
        self.participant = None
        self.readfile(filename)
        self.read_random_ints_file()

    def readfile(self, fname):
        condition_seq = []
        with open(fname, 'r') as file:
            for line in file.readlines():
                for n in line.split(', '):
                    condition_seq.append(int(n))
        trials = len(condition_seq)
        self.trials = trials
        self.condition_seq = condition_seq
        self.trial_nums = range(trials)

    def read_random_ints_file(self, fname='random ints.txt'):
        random_ints = []
        with open(fname, 'r') as file:
            for line in file.readlines():
                for n in line.split(', '):
                    random_ints.append(int(n))
        self.random_ints = random_ints

    def get_trials(self):
```
return self.trials

def get_recordcount(self):
    return self.recordcount

def record(self, location, vibetype):
    self.locations.append(location)
    self.vibetypes.append(vibetype)
    self.recordcount = self.recordcount + 1

def get_condition(self, i=None):
    if i == None:
        i = self.recordcount
    print i
    return self.condition_seq[i]

def get_random_int(self, i=None):
    if i == None:
        i = self.recordcount
    return self.random_ints[i]

def set_participant(self, participant):
    self.participant = participant

def get_participant(self):
    return self.participant

def writetofile(self, filename):
    f = open(filename, 'w')
    f.write("participant: " + str(self.participant) + "\n")
    f.write("Trial \t Condition \t Location Response \t Vibe Type Response\n")
    for i in range(self.start_trial, len(self.vibetypes) + self.start_trial):
        f.write(            str(self.trialnums[i]) + "\t" + str(self.condition_seq[i]) + "\t" + str(self.locations[i - self.start_trial]) + "\t" + str(self.vibetypes[i - self.start_trial]) + "\n")
    f.close()

class simpleapp_tk(Tkinter.Tk):
    def __init__(self, parent):
        Tkinter.Tk.__init__(self, parent)
self.parent = parent
self.initialize()
self.location = None
self.vibetype = None

def initialize(self):
    # We create widgets (button, text field, etc) in this method
    self.grid()  # Create grid layout manager and tell the window to use it
    self.myexperiment = ExperimentData()
    self.location = None
    self.vibetype = None
    self.rightserial = serial.Serial("/dev/tty.usbmodem1451", 9600)
    self.leftserial = serial.Serial("/dev/tty.usbmodem1411", 9600)
    isConnected = False
    while not isConnected:
        ser_in = self.rightserial.readline()
        print ser_in
        isConnected = True
    isConnected = False
    while not isConnected:
        ser_in = self.leftserial.readline()
        print ser_in
        isConnected = True

self.isrightdemo = True

#--------Add widgets----------------
#this should create a box to enter text
self.entryVariable = Tkinter.StringVar()
self.entry = Tkinter.Entry(self, textvariable=self.entryVariable)  # create the entry
self.entry.grid(column=0, row=0, sticky='EW')  # add to layout manager
# can ask widget to stick to edges of the cell, E-East W-West, etc.
self.entry.bind("<Return>",
        self.OnPressEnter)  # Event handler for pressing ENTER
#self.entryVariable.set(u"Enter text here.")

#create main buttons
self.Gobutton = Tkinter.Button(
    self, text=u"Enter", command=self.OnGoButtonClick)
#this part is Event handler
self.GoButton.grid(column=1, row=0)  # location of button

self.NextButton = tkinter.Button(
    self, text=u"Next Trial", command=self.OnNextButtonClick)
self.NextButton.grid(column=1, row=2)

self.EndButton = tkinter.Button(
    self, text=u"End", command=self.OnEndButtonClick)
self.EndButton.grid(column=1, row=15, sticky='E')

self.DemoButton = tkinter.Button(
    self, text=u"Demo", command=self.OnDemoButtonClick)
self.DemoButton.grid(column=0, row=3, sticky='W')

self.NoVibeButton = tkinter.Button(
    self,
    text=u"I didn't feel anything",
    command=self.ButtonNoVibeClick)
self.NoVibeButton.grid(column=0, row=15, sticky='W')
self.NoVibeButton.config(state=[u'default'])

self.StartButton = tkinter.Button(
    self, text=u"Begin Experiment", command=self.OnStartButtonClick)
self.StartButton.grid(column=0, row=2, sticky='W')

self.HighButton = tkinter.Button(
    self, text=u"High", command=self.HIButtonClick, width=10)
self.HighButton.grid(column=1, row=6)
self.HighButton.config(state=[u'default'])

self.LowButton = tkinter.Button(
    self, text=u"Low", command=self.LOBButtonClick, width=10)
self.LowButton.grid(column=1, row=7)
self.LowButton.config(state=[u'default'])

self.IncreaseButton = tkinter.Button(
    self, text=u"Increase", command=self.IncreaseButtonClick, width=10)
self.IncreaseButton.grid(column=1, row=8)
self.IncreaseButton.config(state=[u'default'])

self.DecreaseButton = tkinter.Button(
    self, text=u"Decrease", command=self.DecreaseButtonClick, width=10)
self.DecreaseButton.grid(column=1, row=9)
self.DecreaseButton.config(state=[u'default'])
#create label

self.labelVariable = Tkinter.StringVar()

label = Tkinter.Label(self,
    textvariable=self.labelVariable,
    anchor="c",
    fg="white",
    bg="blue",
    height=2,
    font=("Helvetica", 20))

label.grid(column=0, row=1, columnspan=2, sticky='EW')

self.labelVariable.set(u"Please enter participant ID.")

#Tell layout manager to resize columns when window is resized

self.grid.columnconfigure(0, weight=1)  #weight specified how much column can grow

self.resizable(True, True)  #prevents vertical resizing of window

self.update()

#make sure Tkinter has finished rendering widgets and evaluating their sizes

self.geometry()

self.geometry()  #set window size so doesn’t change with input

#create canvas and selection buttons

canvas = Tkinter.Canvas(width=900, height=650)
canvas.grid(column=0, row=4, rowspan=10)

#canvas.create_rectangle(0,0,900,700,fill="blue")

self.photo = Tkinter.PhotoImage(file='feetimage.gif')
canvas.create_image(450, 350, image=self.photo)

self.buttonR1 = Tkinter.Button(self,
    text=u"R1", command=self.ButtonR1Click)

buttonR1_window = canvas.create_window(600, 75, window=self.buttonR1)

self.buttonR1.config(state=['disabled'])

self.buttonL1 = Tkinter.Button(self,
    text=u"L1", command=self.ButtonL1Click)

buttonL1_window = canvas.create_window(300, 75, window=self.buttonL1)

self.buttonL1.config(state=['disabled'])

self.buttonR2 = Tkinter.Button(self,
    text=u"R2", command=self.ButtonR2Click)

buttonR2_window = canvas.create_window(775, 340, window=self.buttonR2)

self.buttonR2.config(state=['disabled'])

self.buttonL2 = Tkinter.Button(self,
    text=u"L2", command=self.ButtonL2Click)

self.buttonL2.config(state=['disabled'])
buttonL2_window = canvas.create_window(125, 340, window=self.buttonL2)
self.buttonL2.config(state=['disabled'])

self.buttonR3 = Tkinter.Button(
    self, text=u'"R3"', command=self.ButtonR3Click)
buttonR3_window = canvas.create_window(740, 430, window=self.buttonR3)
self.buttonR3.config(state=['disabled'])

self.buttonL3 = Tkinter.Button(
    self, text=u'"L3"', command=self.ButtonL3Click)
buttonL3_window = canvas.create_window(160, 430, window=self.buttonL3)
self.buttonL3.config(state=['disabled'])

self.buttonR4 = Tkinter.Button(
    self, text=u'"R4"', command=self.ButtonR4Click)
buttonR4_window = canvas.create_window(710, 530, window=self.buttonR4)
self.buttonR4.config(state=['disabled'])

self.buttonL4 = Tkinter.Button(
    self, text=u'"L4"', command=self.ButtonL4Click)
buttonL4_window = canvas.create_window(190, 530, window=self.buttonL4)
self.buttonL4.config(state=['disabled'])

self.buttonR5 = Tkinter.Button(
    self, text=u'"R5"', command=self.ButtonR5Click)
buttonR5_window = canvas.create_window(600, 625, window=self.buttonR5)
self.buttonR5.config(state=['disabled'])

self.buttonL5 = Tkinter.Button(
    self, text=u'"L5"', command=self.ButtonL5Click)
buttonL5_window = canvas.create_window(300, 625, window=self.buttonL5)
self.buttonL5.config(state=['disabled'])

self.buttonR6 = Tkinter.Button(
    self, text=u'"R6"', command=self.ButtonR6Click)
buttonR6_window = canvas.create_window(530, 350, window=self.buttonR6)
self.buttonR6.config(state=['disabled'])

self.buttonL6 = Tkinter.Button(
    self, text=u'"L6"', command=self.ButtonL6Click)
buttonL6_window = canvas.create_window(370, 350, window=self.buttonL6)
self.buttonL6.config(state=['disabled'])

def send_vibesignal(self):
    condition = self.myexperiment.get_condition()
isRight, command = self.num_to_vibe(condition)
if isRight:
    self.rightserial.write(command)
else:
def num_to_vibe(self, condition):
    numStr = str(condition)
    modeList = ['H', 'H', 'L', 'I', 'D']
    vibeInt = 0
    modeInt = 0
    type
    if len(numStr) > 2:
        print numStr
        vibeInt = int(numStr[:2])
        modeInt = int(numStr[2])
    else:
        vibeInt = int(numStr[0])
        modeInt = int(numStr[1])
    modeStr = modeList[modeInt]
    isRight = True
    if vibeInt > 6:
        isRight = False
        vibeInt -= 6
    vibeStr = str(vibeInt)
    print "isRight", isRight
    print "command ", vibeStr + modeStr
    return isRight, vibeStr + modeStr

def write_random_int(self):
    if self.myexperiment.get_participant()[-1] == 'D':
        self.labelVariable.set(u"Start counting from: " + str(
            self.myexperiment.get_random_int()))

#Methods to bind to widget event handlers

def EnableVibeTypeButtons(self):
    self.HIbutton.config(state=['active'])
    self.LObutton.config(state=['active'])
    self.Increasebutton.config(state=['active'])
    self.Decreasebutton.config(state=['active'])

def DisableLocationButtons(self):
    self.buttonR1.config(state=['disable'])
    self.buttonL1.config(state=['disable'])
    self.buttonR2.config(state=['disable'])
    self.buttonL2.config(state=['disable'])
    self.buttonR3.config(state=['disable'])
    self.buttonL3.config(state=['disable'])
self.buttonL3.config(state=['disable'])
self.buttonR4.config(state=['disable'])
self.buttonL4.config(state=['disable'])
self.buttonR5.config(state=['disable'])
self.buttonL5.config(state=['disable'])
self.buttonR6.config(state=['disable'])
self.buttonL6.config(state=['disable'])
self.NoVibeButton.config(state=['disable'])

def EnableLocationButtons(self):
    self.buttonR1.config(state=['active'])
    self.buttonL1.config(state=['active'])
    self.buttonR2.config(state=['active'])
    self.buttonL2.config(state=['active'])
    self.buttonR3.config(state=['active'])
    self.buttonL3.config(state=['active'])
    self.buttonR4.config(state=['active'])
    self.buttonL4.config(state=['active'])
    self.buttonR5.config(state=['active'])
    self.buttonL5.config(state=['active'])
    self.buttonR6.config(state=['active'])
    self.buttonL6.config(state=['active'])
    self.NoVibeButton.config(state=['active'])

def DisableVibeTypeButtons(self):
    self.HIbutton.config(state=['disabled'])
    self.LObutton.config(state=['disabled'])
    self.Increasebutton.config(state=['disabled'])
    self.Decreasebutton.config(state=['disabled'])

def OnGoButtonClick(self):
    # clicking the button will trigger this method
    self.labelVariable.set("Participant # + self.entryVariable.get()")  # set or get sets or reads value
    self.myexperiment.set_participant(self.entryVariable.get())

def OnPressEnter(self, event):
    # pressing enter will trigger this method
    self.labelVariable.set("Participant # + self.entryVariable.get()")
    self.myexperiment.set_participant(self.entryVariable.get())

def OnNextButtonClick(self):
    self.labelVariable.set(" ")
    self.DisableVibeTypeButtons()
    self.Nextbutton.config(state=['disable'])
    self.myexperiment.record(self.location, self.vibetype)
if self.myexperiment.get_recordcount() < self.myexperiment.get_trials():
    print self.myexperiment.get_condition()
    self.EnableLocationButtons()
    self.write_random_int()
    self.send_vibesignal()
else:
    self.labelVariable.set("Trials complete!")
    self.Nextbutton.config(state=['disable'])
    self.myexperiment.writetofile(self.myexperiment.get_participant() + str(datetime.datetime.now()) + '.txt')

def OnEndButtonClick(self):
    self.myexperiment.writetofile(self.myexperiment.get_participant() + str(datetime.datetime.now()) + '-INCOMPLETE.txt')
    self.rightserial.write('E')
    self.leftserial.write('E')
    self.rightserial.close()
    self.leftserial.close()
    self.labelVariable.set("Thank you for participating!")
time.sleep(5)
    self.destroy()

def OnDemoButtonClick(self):
    self.labelVariable.set("Demo")
    self.labelVariable.set("You will feel each of the four signal types in sequence at each foot location")
    if self.isrightdemo:
        self.isrightdemo = False
        self.rightserial.write('X')
    else:
        self.leftserial.write('X')
        self.isrightdemo = True

#write demo program

def OnStartButtonClick(self):
    self.labelVariable.set("")
    self.EnableLocationButtons()
    self.Startbutton.config(state=['disable'])
    self.Demobutton.config(state=['disable'])
    self.write_random_int()
    self.send_vibesignal()

def HIButtonClick(self):
    self.labelVariable.set("High")
self.Nextbutton.config(state=['active'])
self.DisableVibeTypeButtons()
self.vibetype = 1

def LOButtonClick(self):
    self.labelVariable.set("Low")
    self.Nextbutton.config(state=['active'])
    self.DisableVibeTypeButtons()
    self.vibetype = 2

def IncreaseButtonClick(self):
    self.labelVariable.set("Increase")
    self.Nextbutton.config(state=['active'])
    self.DisableVibeTypeButtons()
    self.vibetype = 3

def DecreaseButtonClick(self):
    self.labelVariable.set("Decrease")
    self.Nextbutton.config(state=['active'])
    self.DisableVibeTypeButtons()
    self.vibetype = 4

def ButtonR1Click(self):
    self.labelVariable.set("You selected location R1. What kind of vibration?")
    self.EnableVibeTypeButtons()
    self.DisableLocationButtons()
    self.location = 1

def ButtonL1Click(self):
    self.labelVariable.set("You selected location L1. What kind of vibration?")
    self.EnableVibeTypeButtons()
    self.DisableLocationButtons()
    self.location = 7

def ButtonR2Click(self):
    self.labelVariable.set("You selected location R2. What kind of vibration?")
    self.EnableVibeTypeButtons()
    self.DisableLocationButtons()
    self.location = 2

def ButtonL2Click(self):
    self.labelVariable.set(110
"You selected location L2. What kind of vibration?"
self.EnableVibeTypeButtons()
self.DisableLocationButtons()
self.location = 8

def ButtonR3Click(self):
    self.labelVariable.set(
        "You selected location R3. What kind of vibration?"
    )
    self.EnableVibeTypeButtons()
    self.DisableLocationButtons()
    self.location = 3

def ButtonL3Click(self):
    self.labelVariable.set(
        "You selected location L3. What kind of vibration?"
    )
    self.EnableVibeTypeButtons()
    self.DisableLocationButtons()
    self.location = 9

def ButtonR4Click(self):
    self.labelVariable.set(
        "You selected location R4. What kind of vibration?"
    )
    self.EnableVibeTypeButtons()
    self.DisableLocationButtons()
    self.location = 4

def ButtonL4Click(self):
    self.labelVariable.set(
        "You selected location L4. What kind of vibration?"
    )
    self.EnableVibeTypeButtons()
    self.DisableLocationButtons()
    self.location = 10

def ButtonR5Click(self):
    self.labelVariable.set(
        "You selected location R5. What kind of vibration?"
    )
    self.EnableVibeTypeButtons()
    self.DisableLocationButtons()
    self.location = 5

def ButtonL5Click(self):
    self.labelVariable.set(
        "You selected location L5. What kind of vibration?"
    )
    self.EnableVibeTypeButtons()
    self.DisableLocationButtons()
def ButtonR6Click(self):
    self.labelVariable.set(
        "You selected location R6. What kind of vibration?"
    )
    self.EnableVibeTypeButtons()
    self.DisableLocationButtons()
    self.location = 6

def ButtonL6Click(self):
    self.labelVariable.set(
        "You selected location L6. What kind of vibration?"
    )
    self.EnableVibeTypeButtons()
    self.DisableLocationButtons()
    self.location = 12

def ButtonNoVibeClick(self):
    self.labelVariable.set("You did not feel anything")
    self.DisableLocationButtons()
    self.Nextbutton.config(state=['active'])
    self.location = 99
    self.vibetype = 9

if __name__ == "__main__":
    app = simpleapp_tk(
        None)  # Parent is 'None' (no parent) since first GUI element
    app.title('MIPOA Experiment 1')  # Give title to window
    app.mainloop()
# program will loop indefinitely, waiting for events in order to respond

B.3 Vibrotactile Boot Edison Wi-Fi C Code

#include <SPI.h>
#include <WiFi.h>
#include <WiFiUdp.h>

int cue = 0;
int port = 25000;
int porta = 2222;
int previous_cue = 1;
int count = 0;

WiFiUDP Udp;
char ssid[] = "MIT";
char pass[] = "";
int status = WL_IDLE_STATUS;

char packetBuffer[255]; // buffer to hold incoming packet
String ReplyBuffer = "hello";

IPAddress remoteIPA;

int received = 0;

String receivedInput;

void setup() {
    Serial.begin(19200);
    pinMode(LED_BUILTIN, OUTPUT);
    Serial1.begin(19200); // listen to Arduino UNO
    // attempt to connect to Wifi network:
    Serial.print("status: ");
    Serial.println(status);
    status = WiFi.status();
    Serial.print("new status: ");
    Serial.println(status);
    while (status != WL_CONNECTED) {
        Serial.print("Attempting to connect to SSID: ");
        Serial.println(ssid);
        // Connect to WPA/WPA2 network. Change this line if using open or WEP network:
        status = WiFi.begin(ssid);
        // wait 10 seconds for connection:
        delay(10000);
    }

    int success = Udp.begin(port);
    Serial.println(success);
    long rssi = WiFi.RSSI();
    Serial.print("Signal Strength : ");
    Serial.print(rssi);
    Serial.println(" dBm");
    Serial.print("localIP : ");
    Serial.println(WiFi.localIP());
}
Serial.println("Waiting for Moverio Signal");
digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage level)
delay(500);  // wait for a second
digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW
delay(500);
digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage level)
delay(500);
digitalWrite(LED_BUILTIN, LOW);  // turn the LED on (HIGH is the voltage level)
delay(500);
digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage level)
LED on (HIGH is the voltage level)
digitalWrite(LED_BUILTIN, LOW);  // turn the LED on (HIGH is the voltage level)
delay(500);
digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage level)
// wait for a second
digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW
delay(500);
digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage level)
delay(500);
digitalWrite(LED_BUILTIN, LOW);  // turn the LED on (HIGH is the voltage level)
delay(500);
digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage level)
// wait for a second
digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW
delay(500);
digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage level)
delay(500);
digitalWrite(LED_BUILTIN, LOW);  // turn the LED on (HIGH is the voltage level)
delay(500);
digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage level)
// wait for a second
digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW
delay(500);
digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage level)
delay(500);
digitalWrite(LED_BUILTIN, LOW);  // turn the LED on (HIGH is the voltage level)
delay(500);
digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage level)
// wait for a second
digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW
delay(500);
digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage level)
delay(500);
digitalWrite(LED_BUILTIN, LOW);  // turn the LED on (HIGH is the voltage level)
delay(500);
digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage level)
// wait for a second
digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW
delay(500);
digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage level)
delay(500);
digitalWrite(LED_BUILTIN, LOW);  // turn the LED on (HIGH is the voltage level)
delay(500);
digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage level)
// wait for a second
digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW
delay(500);
digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage level)
delay(500);
digitalWrite(LED_BUILTIN, LOW);  // turn the LED on (HIGH is the voltage level)
delay(500);
digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage level)
// wait for a second
digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW
delay(500);
digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage level)
delay(500);
digitalWrite(LED_BUILTIN, LOW);  // turn the LED on (HIGH is the voltage level)
delay(500);
digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage level)
// wait for a second
digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW
delay(500);
digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage level)
delay(500);
digitalWrite(LED_BUILTIN, LOW);  // turn the LED on (HIGH is the voltage level)
delay(500);
digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage level)
// wait for a second
digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW
delay(500);
digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage level)
delay(500);
digitalWrite(LED_BUILTIN, LOW);  // turn the LED on (HIGH is the voltage level)
delay(500);
digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage level)
// wait for a second
digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW
delay(500);
digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage level)
delay(500);
digitalWrite(LED_BUILTIN, LOW);  // turn the LED on (HIGH is the voltage level)
delay(500);
digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage level)
// wait for a second
digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW
delay(500);
digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage level)
delay(500);
digitalWrite(LED_BUILTIN, LOW);  // turn the LED on (HIGH is the voltage level)
delay(500);
digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage level)
// wait for a second
digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW
delay(500);
digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage level)
delay(500);
digitalWrite(LED_BUILTIN, LOW);  // turn the LED on (HIGH is the voltage level)
delay(500);
digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage level)
// wait for a second
digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW
delay(500);
digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage level)
delay(500);
digitalWrite(LED_BUILTIN, LOW);  // turn the LED on (HIGH is the voltage level)
delay(500);
digitalWrite(LED_BUILTIN, HIGH);  // turn the LED on (HIGH is the voltage level)
// wait for a second
digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW
delay(500);

delay(500);

while (received < 2) {
    int packetsize = Udp.parsePacket();
    if (packetsize) {
        received += 1;
        IPAddress remotelP = Udp.remoteIP();
        if (Serial.available()) {
            Serial.print("Received packet of size : ");
            Serial.println(packetsize);
            Serial.print("From ");
            Serial.print(remotelp);
        }
    }

    // read the packet into packetBuffer
    int len = Udp.read(packetBuffer, 255);
    if (Serial.available()) {
        Serial.println("Contents:");
        for (int i = 0; i < len; i++) {
            Serial.print(packetBuffer[i]);
        }
        Serial.println(" ");
    }
}

remotelP = remotelP;
Udp.beginPacket(remotelP, porta);
Udp.print(cue);
Udp.endPacket();
}

void loop() {

digitalWrite(LED_BUILTIN,
            HIGH); // turn the LED on (HIGH is the voltage level)

    if (Serial1.available() > 0) {
        cue = Serial1.read();
        Serial.write(cue);
        Udp.beginPacket(remotelP, porta);
        Udp.write(cue);
        Udp.endPacket();
    }
B.4 Vibrotactile Boot Arduino UNO C Code

```c
#include <Wire.h> // Supposedly Edison version
#include <Wire.h> // I2C library, gyroscope
#include "Adafruit_DRV2605.h" // haptic driver boards
#include "Adafruit_VCML4010.h" // proximity sensor (front)
#include <SparkFun_VL6180X.h> // proximity sensor (back)
#include <stdio.h>
#include <avr/wdt.h>

// To see the cues being sent over wi-fi, go to http://<IP>/arduino/webserver/

float PITCH_MAX = 2.0; // degrees (point at which to ignore rangefinder if +/-
                        // ....alternative: have separate ones for pitching up
                        // and down, as shown below)
float DIST1 = 120; // (first and farthest distance level at which cues begin)
                    // approx. 60 inches
float DIST2 = 55; // second distance level at which CUE 2 begins, approx. 30 inches
float DIST3 = 22; // approx. 8 inches on rangefinder
float PROX_CLOSE_FRONT = 3900; // Reads approx 3750 at 2 inches (distance at which cue becomes 5)

// external variables
char str[512];
boolean firstSample = true;
boolean calibrate = true;
boolean read_range = false;
float zero_pt = 0.0;
float pitch = 0.0;
float range_reading = 0.0;
float range_dist = 1000;
float prox_front_reading = 0.0;
float prox_front_dist = 0.0;
int tstart_cue1 = 0;
int tstart_cue2 = 0;
int tstart_cue3 = 0;
int cue = 0; // can take a numbers 0, 1, 2, 5, 6, 7

//------------------SET UP PERIPHERAL HARDWARE------------------

// Vibe motor haptic driver class instance and setup
```
Adafruit_DRV2605 drv;
int VibePin_F1 = 3;
int VibePin_F2 = 5;
int VibePin_B1 = 6;
int VibePin_B2 = 9;

// Front proximity sensor class instance
Adafruit_VCNL4010 vcnl;

// Back proximity sensor
#define VL6180X_ADDRESS 0x29
VL6180xIdentification identification;
VL6180x sensor(VL6180X_ADDRESS);

void printIdentification(struct VL6180xIdentification *temp) {
  Serial.print("Model ID = ");
  Serial.println(temp->idModel);

  Serial.print("Model Rev = ");
  Serial.print(temp->idModelRevMajor);
  Serial.print(".");
  Serial.println(temp->idModelRevMinor);

  Serial.print("Module Rev = ");
  Serial.print(temp->idModuleRevMajor);
  Serial.print(".");
  Serial.println(temp->idModuleRevMinor);

  Serial.print("Manufacture Date = ");
  Serial.print(((temp->idDate >> 3) & 0x001F) + 1950);
  Serial.print("/");
  Serial.print(((temp->idDate >> 8) & 0x000F) + 100);
  Serial.print("/");
  Serial.print(((temp->idDate >> 12) & 0x000F) + 2000);
  Serial.print(" Phase: ");
  Serial.print((temp->idDate & 0x0007) + 1995);
  Serial.print("Manufacture Time (s)= ");
  Serial.println(temp->idTime * 2);
  Serial.println();
}

// Accelerometer ADXL345
#define ACC
(0x53) // ADXL345 ACC I2C address according to "I2C" section in manual
#define A_TO_READ
(6) // number of bytes we are going to read each time (two bytes for each
// axis)

// Gyroscope ITG3200
#define GYRO
0x68 // gyro address, binary = 11101001 when ADO is connected to Vcc (see
// schematics of your breakout board)
#define G_SMPLRT_DIV 0x15
#define G_DLPF_FS 0x16
#define G_INT_CFG 0x17
#define G_PWR_MGM 0x3E
#define G_TO_READ 8 // 2 bytes for each axis x, y, z

// offsets are chip specific - external variables - ADJUST THESE AT SOME POINT
int g_offx = 120;
int g_offy = 20;
int g_offz = 93;

// Vibe at level 5 - High and Steady
void cue_5(int vibe_pin) {
    drv.setMode(DRV2605_MODEPWMANALOG);
    analogWrite(vibe_pin, 220);
}

// Vibe at level 3 - Medium-high 225 @ 8Hz
void cue_3(int vibe_pin) {
    drv.setMode(DRV2605_MODEPWMANALOG);

    if ((millis() % 125) > 63) {
        analogWrite(vibe_pin, 210);
    } else {
        analogWrite(vibe_pin, 127);
    }
}

// Vibe at level 2 - Medium 200 @ 4Hz
void cue_2(int vibe_pin) {
    drv.setMode(DRV2605_MODEPWMANALOG);

    if ((millis() % 250) > 125) {
        analogWrite(vibe_pin, 190);
    } else {
analogWrite(vibe_pin, 127);
}
}

// Vibe at level 1 - Medium-low 170 @ 2Hz
void cue_1(int vibe_pin) {
    drv.setMode(DRV2605_MODEPWMANALOG);
    if ((millis() % 500) > 250) {
        analogWrite(vibe_pin, 170);
    } else {
        analogWrite(vibe_pin, 127);
    }
}

// Turn off vibration - Level 0
void cue_off(int vibe_pin) {
    analogWrite(vibe_pin, 127); // turn off motor
    drv.setMode(DRV2605_MODE_INTTRIG);
}

//----------------ACCEL/GYRO FUNCTIONS-------------------

// Allocate memory storage for accel/gyro data and external variables
float RwAcc[3]; // projection of normalized gravitation force vector on x/y/z axis, as measured by accelerometer
float Gyro_ds[3]; // Gyro readings in degrees/sec
float RwGyro[3]; // Rw obtained from last estimated value and gyro movement
float Awxz[2]; // angles between projection of R on XZ/YZ plane and Z axis (deg)
float RwEst[3]; // Estimated values from combining accel and gyro information
int lastTime = 0;
int interval = 0;
float wGyro = 10.0;

// Function to turn on the ADXL345
void initAcc() {
    writeTo(ACC, Ox2D, 0); // Format: WriteTo(DEVICE, address, value) --> writes // val of 0 to POWER_CTL
    writeTo(ACC, Ox2D, 16); // writes val of 16 to POWER_CTL
    writeTo(ACC, Ox2D, 8); // writes val of 8 to POWER_CTL
    // Note: by default the device is in +/2g range reading
}

// Function to read raw data from accelerometer
void getAccelerometerData(int *result) {
    int regAddress = 0x32; // first axis (X) acceleration data register on the
byte buff[A_TO_READ]; // buff is array of six bytes to store data

readFrom(ACC, regAddress, A_TO_READ, buff); // read the acceleration data from
// the ADXL345. (device, register
// address, num bytes to read,
// store bytes in buff)

// each axis reading comes in 10 bit resolution, ie 2 bytes. Least
// Significant is the first Byte in buff (buff[0])
// thus we are converting both bytes into one 8-byte int
// cast of (int) before buff obtains the buff byte in int form before
// left-shifting (bitwise operation)
result[0] = (((int) buff[1]) << 8) | buff[0]; // x-axis
result[1] = (((int) buff[3]) << 8) | buff[2]; // y-axis
result[2] = (((int) buff[5]) << 8) | buff[4]; // z-axis

// Normalize raw accelerometer data (convert to g)
void rawAccToG(int *raw, float *RwAcc) {
    RwAcc[0] = ((float) raw[0]) / 256.0; // for 10-bit, +/-2g, the sensitivity at
    // Xout, Yout, Zout is 256 LSB/g (pg 3 of
    // manual)
    RwAcc[1] = ((float) raw[1]) / 256.0;
    RwAcc[2] = ((float) raw[2]) / 256.0;
}

// Initialize the gyroscope
void initGyro() {
    /*******************************
    * ITG 3200
    * power management set to:
    * clock select = internal oscillator
    * no reset, no sleep mode
    * no standby mode
    * sample rate to = 125Hz
    * parameter to +/- 2000 degrees/sec
    * low pass filter = 5Hz
    * no interrupt
    *******************************
    writeTo(GYRO, G_PWR_MGM, 0x00);
    writeTo(GYRO, G_SMPLRT_DIV, 0x07); // EB, 50, 80, 7F, DE, 23, 20, FF
    writeTo(GYRO, G_DLPF_FS, 0x1E); // +/- 2000 dgrs/sec, 1kHz, 1E, 19
    writeTo(GYRO, G_INT_CFG, 0x00);

}
// Read data from gyroscope
void getGyroscopeData(int *result) {
    /*
     * ITG-3200 I2C
     * registers:
     * temp MSB = 1B, temp LSB = 1C
     * x axis MSB = 1D, x axis LSB = 1E
     * y axis MSB = 1F, y axis LSB = 20
     * z axis MSB = 21, z axis LSB = 22
     */
    int regAddress = 0x1B;
    int temp, x, y, z;
    byte buff[G_TO_READ]; // two bytes per axis + two bytes for temperature

    readFrom(GYRO, regAddress, G_TO_READ, buff); // read the gyro data from the ITG3200 and store in buff

    result[0] = ((buff[2] << 8) | buff[3]) + g_offx;
    result[3] = (buff[0] << 8) | buff[1]; // temperature reading
}

// Convert raw gyro data to degrees/sec
void rawGyroToDegsec(int *raw, float *gyro_ds) {
    gyro_ds[0] = ((float)raw[0]) / 14.375;
}

// Normalize a 3D vector
void normalize3DVec(float *vector) {
    float R;
    vector[0] /= R; // normalized x component
    vector[1] /= R; // normalized y component
    vector[2] /= R; // normalized z component
}

// Square a number x
float squared(float x) { return x * x; }

121
void getInclination() {
    int w = 0;
    float tmpf = 0.0;
    int currentTime, signRzGyro;

    currentTime = millis(); // number of milliseconds since Arduino board began
                          // running the program
    interval =
    currentTime -
    lastTime; // time since beginning of code where lastTime initialized to 0
    lastTime = currentTime;

    if (firstSample) { // the NaN check is used to wait for good data from the
                       // Arduino, initialized to True in beginning of code
    for (w = 0; w <= 2; w++) {
        RwEst[w] = RwAcc[w]; // initialize with accelerometer readings
    }
    }

    // Otherwise, evaluate RwGyro vector
    else {
        // evaluate RwGyro vector
    if (abs(RwEst[2]) < 0.1) {
        // Rz is too small and because it is used as reference for computing Axz,
        // Ayz it's error fluctuations will amplify leading to bad results
        // in this case skip the gyro data and just use previous estimate
        for (w = 0; w <= 2; w++) {
            RwGyro[w] = RwEst[w];
        }
    } else {
        // get angles between projection of R on ZX/ZY plane and Z axis, based on
        // last RwEst
        for (w = 0; w <= 1; w++) {
            tmpf = Gyro_ds[w]; // get current gyro rate in deg/s
            tmpf *= interval / 1000.0f; // get angle change in deg
            Awz[w] = atan2(RwEst[w], RwEst[2]) * 180 / PI; // get angle and convert to degrees
            Awz[w] += tmpf; //get updated angle
            // according to gyro movement
        }

        // estimate sign of RzGyro by looking in what quadrant the angle Axz is,
// RzGyro is positive if Axz in range -90 .. 90 -> cos(Awz) >= 0
signRzGyro = (cos(Awz[0] * PI / 180) >= 0) ? 1 : -1;

// reverse calculation of RwGyro from Awz angles, for formula deductions
// see http://starlino.com/imu_guide.html
for (w = 0; w <= 1; w++) {
    RwGyro[0] = sin(Awz[0] * PI / 180);
    RwGyro[0] /= sqrt(1 +
        squared(cos(Awz[0] * PI / 180)) *
        squared(tan(Awz[1] * PI / 180)));
    RwGyro[1] = sin(Awz[1] * PI / 180);
    RwGyro[1] /= sqrt(1 +
        squared(cos(Awz[1] * PI / 180)) *
        squared(tan(Awz[0] * PI / 180)));
}
RwGyro[2] =
    signRzGyro * sqrt(1 - squared(RwGyro[0]) - squared(RwGyro[1]));
}

// combine Accelerometer and gyro readings
for (w = 0; w <= 2; w++)
    RwEst[w] = (RwAcc[w] + wGyro * RwGyro[w]) / (1 + wGyro);

normalize3DVec(RwEst); // Final, normalized 3D vector
}

firstSample = false;
}

//******************************************MAIN PROGRAM******************************************/

// Begin serial communication
void setup() {
    wdt_reset();
    wdt_enable(WDTO_2S);
    Serial.begin(19200); // Open serial comm, specify baud rate
}

Wire.begin(); // Use I2C protocol
initAcc(); // initialize Accel
initGyro(); // initialize Gyro

// Initialize vibration motor driver boards
drv.begin();

123
drv.selectLibrary(1);

// Initialize front proximity sensor
if (!vcnl.begin()) {
    while (1)
    ;
}

// Set up PWM pins as output for vibe motor driver boards
pinMode(VibePin_F1, OUTPUT);
analogWrite(VibePin_F1, 127);

pinMode(VibePin_F2, OUTPUT);
analogWrite(VibePin_F2, 127);

drv.go();
}

// ---------------- MAIN LOOP ----------------/

void loop() {
    wdt_reset();

    if (!Serial.available()) {

        int acc[3];
        int gyro[4];

        getAccelerometerData(acc);
        rawAccToG(acc, RwAcc);
        normalize3DVec(RwAcc);

        getGyroscopeData(gyro);
        rawGyroToDegsec(gyro, Gyro_ds);

        getInclination();
        pitch = Awz[1];

        if (calibrate && (pitch != 0.00)) {
            zero_pt = pitch;
            calibrate = false;
            read_range = true;
        }

        if (pitch > (zero_pt + PITCH_MAX) || pitch < (zero_pt - PITCH_MAX)) {
            read_range = false;
        }
    }
}
} else {
    read_range = true;
}

if (read_range) {
    range_reading = analogRead(A0);
    range_dist = range_reading;
    prox_front_reading = vcnl.readProximity();
    prox_front_dist = prox_front_reading;
}

if (prox_front_dist >= PROX_CLOSE_FRONT) // TURN MOTORS ON LEVEL 5
{
    cue = 5;
    cue_5(VibePin_F1);
    cue_5(VibePin_F2);
}

else if (((prox_front_dist < PROX_CLOSE_FRONT) &&
          (range_dist <= DIST3))) // TURN MOTORS ON LEVEL 3
{
    if (cue != 3)
        tstart_cue3 = millis();
    cue = 3;
    cue_3(VibePin_F1);
    cue_3(VibePin_F2);
}

else if ((range_dist <= DIST2) &&
          (range_dist > DIST3)) // TURN MOTORS ON LEVEL 2
{
    if (cue != 2)
        tstart_cue2 = millis();
    cue = 2;
    cue_2(VibePin_F1);
    cue_2(VibePin_F2);
}

else if ((range_dist <= DIST1) &&
          (range_dist > DIST2)) // TURN MOTORS ON LEVEL 1
{
    if (cue != 1)
        tstart_cue1 = millis();
    cue = 1;
    cue_1(VibePin_F1);
```c
void writeTo(int DEVICE, byte address, byte val) {
    Wire.beginTransmission(DEVICE); // start transmission to ACC
    Wire.write(address); // send register address
    Wire.write(val); // write the value
    Wire.endTransmission(); // end transmission
}
```
Wire.write(val); // send value to write to address
Wire.endTransmission(); // end transmission
}

// reads num bytes starting from address register on ACC in to buff array
void readFrom(int DEVICE, byte address, int num, byte buff[]) {
    Wire.beginTransmission(DEVICE); // start transmission to ACC
    Wire.write(address); // sends address to read from
    Wire.endTransmission(); // end transmission

    Wire.beginTransmission(DEVICE); // start transmission to ACC
    Wire.requestFrom(DEVICE, num); // format: requestFrom(address, quantity of
    // bytes) --> so this requests 6 bytes from ACC
    int i = 0;
    while (Wire.available()) // ACC may send less than requested (abnormal), but
    // should now know from requestFrom to send 6 bytes
    {
        buff[i] = Wire.read(); // receive a byte from ACC and store in buff array
        i++;
    }
    Wire.endTransmission(); // end transmission
}

B.5 MATLAB Post-Processing Code
clear all; close all; clc;
participants = 10;
for p=1:participants
    pnum = num2str(p);
    eval(['''F'' num2str(p) = impordata(''' num2str(p) 'F.txt');']);
    eval(['''D'' num2str(p) = impordata(''' num2str(p) 'D.txt');']);
    eval(['''Rcorrect'' pnum, ''Lcorrect'' pnum, ''Rcorrect_Type'' pnum, ''Lcorrect_Type'' pnum, ''Rnotfelt'' pnum, ''Lnotfelt'' pnum]);
    eval(['''Rcorrect'' pnum, ''Lcorrect'' pnum, ''Rcorrect_Type'' pnum, ''Lcorrect_Type'' pnum, ''Rnotfelt'' pnum, ''Lnotfelt'' pnum]);
    [Rcorrect_Type3'' pnum, Lcorrect_Type3'' pnum, Rnotfelt'' pnum, Lnotfelt'' pnum] = parsedata(''' num2str(p) 'F.txt');
    [Rcorrect_Type3'' pnum, Lcorrect_Type3'' pnum, Rnotfelt'' pnum, Lnotfelt'' pnum] = parsedata(''' num2str(p) 'D.txt');
    end

% re-format for copy/paste into SPSS format
    eval(['''Rcorrect'' pnum, ''Rcorrect_Type'' pnum, ''Lcorrect'' pnum, ''Lcorrect_Type'' pnum, ''Rnotfelt'' pnum, ''Lnotfelt'' pnum]);
    eval(['''Rcorrect'' pnum, ''Rcorrect_Type'' pnum, ''Lcorrect'' pnum, ''Lcorrect_Type'' pnum, ''Rnotfelt'' pnum, ''Lnotfelt'' pnum]);
    end

DRcorrect_TypeTotal = zeros(6,4);
FRcorrect_TypeTotal = zeros(6,4);
DLcorrect_TypeTotal = zeros(6,4);
FLcorrect_TypeTotal = zeros(6,4);
for i=1:participants
    eval(['''DRcorrect_Type_Total'' = DRcorrect_Type_Total + D' num2str(i)'' _Rcorrect_Type';']);
    eval(['''FRcorrect_Type_Total'' = FRcorrect_Type_Total + F' num2str(i)'' _Rcorrect_Type';']);
    eval(['''DLcorrect_Type_Total'' = DLcorrect_Type_Total + D' num2str(i)'' _Lcorrect_Type';']);
    eval(['''FLcorrect_Type_Total'' = FLcorrect_Type_Total + F' num2str(i)'' _Lcorrect_Type';']);
end
DRcorrect_Type_Total = DRcorrect_Type_Total./participants;
FRcorrect_Type_Total = FRcorrect_Type_Total./participants;
DLcorrect_Type_Total = DLcorrect_Type_Total./participants;
FLcorrect_Type_Total = FLcorrect_Type_Total./participants;

% Type accuracy across all participants and locations
DRcorrect_Type_Total_ALL = [sum(DRcorrect_Type_Total(:,1))/6, ... sum(DRcorrect_Type_Total(:,2))/6, sum(DRcorrect_Type_Total(:,3))/6, ... sum(DRcorrect_Type_Total(:,4))/6];
FRcorrect_Type_Total_ALL = [sum(FRcorrect_Type_Total(:,1))/6, ... sum(FRcorrect_Type_Total(:,2))/6, sum(FRcorrect_Type_Total(:,3))/6, ...
sum(FR correct TypeTotal(:,4))/6;
DL correct TypeTotal_ALL = [sum(DL correct TypeTotal(:,1))/6, ...
sum(DL correct TypeTotal(:,2))/6, sum(DL correct TypeTotal(:,3))/6, ...
sum(DL correct TypeTotal(:,4))/6];
FL correct TypeTotal_ALL = [sum(FL correct TypeTotal(:,1))/6, ...
sum(FL correct TypeTotal(:,2))/6, sum(FL correct TypeTotal(:,3))/6, ...
sum(FL correct TypeTotal(:,4))/6];

%Location accuracies for each subject across all types
Focused_Rscores = [...
F1 Rcorrect(1) F2 Rcorrect(1) F3 Rcorrect(1) F4 Rcorrect(1) F5 Rcorrect(1)
F6 Rcorrect(1) F7 Rcorrect(1) F8 Rcorrect(1) F9 Rcorrect(1) F10 Rcorrect(1);
...
F1 Rcorrect(2) F2 Rcorrect(2) F3 Rcorrect(2) F4 Rcorrect(2) F5 Rcorrect(2)
F6 Rcorrect(2) F7 Rcorrect(2) F8 Rcorrect(2) F9 Rcorrect(2) F10 Rcorrect(2);
...
F1 Rcorrect(3) F2 Rcorrect(3) F3 Rcorrect(3) F4 Rcorrect(3) F5 Rcorrect(3)
F6 Rcorrect(3) F7 Rcorrect(3) F8 Rcorrect(3) F9 Rcorrect(3) F10 Rcorrect(3);
...
F1 Rcorrect(4) F2 Rcorrect(4) F3 Rcorrect(4) F4 Rcorrect(4) F5 Rcorrect(4)
F6 Rcorrect(4) F7 Rcorrect(4) F8 Rcorrect(4) F9 Rcorrect(4) F10 Rcorrect(4);
...
F1 Rcorrect(5) F2 Rcorrect(5) F3 Rcorrect(5) F4 Rcorrect(5) F5 Rcorrect(5)
F6 Rcorrect(5) F7 Rcorrect(5) F8 Rcorrect(5) F9 Rcorrect(5) F10 Rcorrect(5);
...
F1 Rcorrect(6) F2 Rcorrect(6) F3 Rcorrect(6) F4 Rcorrect(6) F5 Rcorrect(6)
F6 Rcorrect(6) F7 Rcorrect(6) F8 Rcorrect(6) F9 Rcorrect(6) F10 Rcorrect(6)];
Divided_Rscores = [...
D1 Rcorrect(1) D2 Rcorrect(1) D3 Rcorrect(1) D4 Rcorrect(1) D5 Rcorrect(1)
D6 Rcorrect(1) D7 Rcorrect(1) D8 Rcorrect(1) D9 Rcorrect(1) D10 Rcorrect(1);
...
D1 Rcorrect(2) D2 Rcorrect(2) D3 Rcorrect(2) D4 Rcorrect(2) D5 Rcorrect(2)
D6 Rcorrect(2) D7 Rcorrect(2) D8 Rcorrect(2) D9 Rcorrect(2) D10 Rcorrect(2);
...
D1 Rcorrect(3) D2 Rcorrect(3) D3 Rcorrect(3) D4 Rcorrect(3) D5 Rcorrect(3)
D6 Rcorrect(3) D7 Rcorrect(3) D8 Rcorrect(3) D9 Rcorrect(3) D10 Rcorrect(3);
...
D1 Rcorrect(4) D2 Rcorrect(4) D3 Rcorrect(4) D4 Rcorrect(4) D5 Rcorrect(4)
D6 Rcorrect(4) D7 Rcorrect(4) D8 Rcorrect(4) D9 Rcorrect(4) D10 Rcorrect(4);
...
D1 Rcorrect(5) D2 Rcorrect(5) D3 Rcorrect(5) D4 Rcorrect(5) D5 Rcorrect(5)
D6 Rcorrect(5) D7 Rcorrect(5) D8 Rcorrect(5) D9 Rcorrect(5) D10 Rcorrect(5);
...
D1 Rcorrect(6) D2 Rcorrect(6) D3 Rcorrect(6) D4 Rcorrect(6) D5 Rcorrect(6)
D6 Rcorrect(6) D7 Rcorrect(6) D8 Rcorrect(6) D9 Rcorrect(6) D10 Rcorrect(6)];
Focused_Lscores = [...
F1 Lcorrect(1) F2 Lcorrect(1) F3 Lcorrect(1) F4 Lcorrect(1) F5 Lcorrect(1)
F6 Lcorrect(1) F7 Lcorrect(1) F8 Lcorrect(1) F9 Lcorrect(1) F10 Lcorrect(1);
...
F1 Lcorrect(2) F2 Lcorrect(2) F3 Lcorrect(2) F4 Lcorrect(2) F5 Lcorrect(2)
F6 Lcorrect(2) F7 Lcorrect(2) F8 Lcorrect(2) F9 Lcorrect(2) F10 Lcorrect(2);
...
F1 Lcorrect(3) F2 Lcorrect(3) F3 Lcorrect(3) F4 Lcorrect(3) F5 Lcorrect(3)
F6 Lcorrect(3) F7 Lcorrect(3) F8 Lcorrect(3) F9 Lcorrect(3) F10 Lcorrect(3);
...
F1 Lcorrect(4) F2 Lcorrect(4) F3 Lcorrect(4) F4 Lcorrect(4) F5 Lcorrect(4)
F6 Lcorrect(4) F7 Lcorrect(4) F8 Lcorrect(4) F9 Lcorrect(4) F10 Lcorrect(4);
...
F1 Lcorrect(5) F2 Lcorrect(5) F3 Lcorrect(5) F4 Lcorrect(5) F5 Lcorrect(5)
F6 Lcorrect(5) F7 Lcorrect(5) F8 Lcorrect(5) F9 Lcorrect(5) F10 Lcorrect(5);
...
F1 Lcorrect(6) F2 Lcorrect(6) F3 Lcorrect(6) F4 Lcorrect(6) F5 Lcorrect(6)
F6 Lcorrect(6) F7 Lcorrect(6) F8 Lcorrect(6) F9 Lcorrect(6) F10 Lcorrect(6));
Divided_Lscores = [...]
    D1_Lcorrect(1) D2_Lcorrect(1) D3_Lcorrect(1) D4_Lcorrect(1) D5_Lcorrect(1)
    D6_Lcorrect(1) D7_Lcorrect(1) D8_Lcorrect(1) D9_Lcorrect(1) D10_Lcorrect(1);
    ...
    D1_Lcorrect(2) D2_Lcorrect(2) D3_Lcorrect(2) D4_Lcorrect(2) D5_Lcorrect(2)
    D6_Lcorrect(2) D7_Lcorrect(2) D8_Lcorrect(2) D9_Lcorrect(2) D10_Lcorrect(2);
    ...
    D1_Lcorrect(3) D2_Lcorrect(3) D3_Lcorrect(3) D4_Lcorrect(3) D5_Lcorrect(3)
    D6_Lcorrect(3) D7_Lcorrect(3) D8_Lcorrect(3) D9_Lcorrect(3) D10_Lcorrect(3);
    ...
    D1_Lcorrect(4) D2_Lcorrect(4) D3_Lcorrect(4) D4_Lcorrect(4) D5_Lcorrect(4)
    D6_Lcorrect(4) D7_Lcorrect(4) D8_Lcorrect(4) D9_Lcorrect(4) D10_Lcorrect(4);
    ...
    D1_Lcorrect(5) D2_Lcorrect(5) D3_Lcorrect(5) D4_Lcorrect(5) D5_Lcorrect(5)
    D6_Lcorrect(5) D7_Lcorrect(5) D8_Lcorrect(5) D9_Lcorrect(5) D10_Lcorrect(5);
    ...
    D1_Lcorrect(6) D2_Lcorrect(6) D3_Lcorrect(6) D4_Lcorrect(6) D5_Lcorrect(6)
    D6_Lcorrect(6) D7_Lcorrect(6) D8_Lcorrect(6) D9_Lcorrect(6) D10_Lcorrect(6);

Focused_Rscores_all = [mean(Focused_Rscores(1,:)); ...
    mean(Focused_Rscores(2,:)); ...
    mean(Focused_Rscores(3,:)); ...
    mean(Focused_Rscores(4,:)); ...
    mean(Focused_Rscores(5,:)); ...
    mean(Focused_Rscores(6,:))];

Focused_Rscores_all_std = [std(Focused_Rscores(1,:)); ...
    std(Focused_Rscores(2,:)); ...
    std(Focused_Rscores(3,:)); ...
    std(Focused_Rscores(4,:)); ...
    std(Focused_Rscores(5,:)); ...
    std(Focused_Rscores(6,:))];

Divided_Rscores_all = [mean(Divided_Rscores(1,:)); ...
    mean(Divided_Rscores(2,:)); ...
    mean(Divided_Rscores(3,:)); ...
    mean(Divided_Rscores(4,:)); ...
    mean(Divided_Rscores(5,:)); ...
    mean(Divided_Rscores(6,:))];

Divided_Rscores_all_std = [std(Divided_Rscores(1,:)); ...
    std(Divided_Rscores(2,:)); ...
    std(Divided_Rscores(3,:)); ...
    std(Divided_Rscores(4,:)); ...
    std(Divided_Rscores(5,:)); ...
    std(Divided_Rscores(6,:))];

Focused_Lscores_all = [mean(Focused_Lscores(1,:)); ...
    mean(Focused_Lscores(2,:)); ...
    mean(Focused_Lscores(3,:)); ...
    mean(Focused_Lscores(4,:)); ...
    mean(Focused_Lscores(5,:)); ...
    mean(Focused_Lscores(6,:))];

Focused_Lscores_all_std = [std(Focused_Lscores(1,:)); ...
    std(Focused_Lscores(2,:)); ...
    std(Focused_Lscores(3,:)); ...
    std(Focused_Lscores(4,:)); ...
    std(Focused_Lscores(5,:)); ...
    std(Focused_Lscores(6,:))];

Divided_Lscores_all = [mean(Divided_Lscores(1,:)); ...
    mean(Divided_Lscores(2,:)); ...
    mean(Divided_Lscores(3,:)); ...
mean(Divided_Lscores(4,:)); ... 
mean(Divided_Lscores(5,:)); ... 
mean(Divided_Lscores(6,:));

Divided_Lscores_all_std = [std(Divided_Lscores(1,:)); ... 
std(Divided_Lscores(2,:)); ... 
std(Divided_Lscores(3,:)); ... 
std(Divided_Lscores(4,:)); ... 
std(Divided_Lscores(5,:)); ... 
std(Divided_Lscores(6,:))];

% UNDETECTED VIBRATIONS
FR_notfelt = [...
    sum(F1_Rnotfelt(1,:)) sum(F2_Rnotfelt(1,:)) sum(F3_Rnotfelt(1,:))
    sum(F4_Rnotfelt(1,:)) sum(F5_Rnotfelt(1,:)) sum(F6_Rnotfelt(1,:))
    sum(F7_Rnotfelt(1,:)) sum(F8_Rnotfelt(1,:)) sum(F9_Rnotfelt(1,:))
    sum(F10_Rnotfelt(1,:)); ...
    sum(F1_Rnotfelt(2,:)) sum(F2_Rnotfelt(2,:)) sum(F3_Rnotfelt(2,:))
    sum(F4_Rnotfelt(2,:)) sum(F5_Rnotfelt(2,:)) sum(F6_Rnotfelt(2,:))
    sum(F7_Rnotfelt(2,:)) sum(F8_Rnotfelt(2,:)) sum(F9_Rnotfelt(2,:))
    sum(F10_Rnotfelt(2,:)); ...
    sum(F1_Rnotfelt(3,:)) sum(F2_Rnotfelt(3,:)) sum(F3_Rnotfelt(3,:))
    sum(F4_Rnotfelt(3,:)) sum(F5_Rnotfelt(3,:)) sum(F6_Rnotfelt(3,:))
    sum(F7_Rnotfelt(3,:)) sum(F8_Rnotfelt(3,:)) sum(F9_Rnotfelt(3,:))
    sum(F10_Rnotfelt(3,:)); ...
    sum(F1_Rnotfelt(4,:)) sum(F2_Rnotfelt(4,:)) sum(F3_Rnotfelt(4,:))
    sum(F4_Rnotfelt(4,:)) sum(F5_Rnotfelt(4,:)) sum(F6_Rnotfelt(4,:))
    sum(F7_Rnotfelt(4,:)) sum(F8_Rnotfelt(4,:)) sum(F9_Rnotfelt(4,:))
    sum(F10_Rnotfelt(4,:)); ...
    sum(F1_Rnotfelt(5,:)) sum(F2_Rnotfelt(5,:)) sum(F3_Rnotfelt(5,:))
    sum(F4_Rnotfelt(5,:)) sum(F5_Rnotfelt(5,:)) sum(F6_Rnotfelt(5,:))
    sum(F7_Rnotfelt(5,:)) sum(F8_Rnotfelt(5,:)) sum(F9_Rnotfelt(5,:))
    sum(F10_Rnotfelt(5,:)); ...
    sum(F1_Rnotfelt(6,:)) sum(F2_Rnotfelt(6,:)) sum(F3_Rnotfelt(6,:))
    sum(F4_Rnotfelt(6,:)) sum(F5_Rnotfelt(6,:)) sum(F6_Rnotfelt(6,:))
    sum(F7_Rnotfelt(6,:)) sum(F8_Rnotfelt(6,:)) sum(F9_Rnotfelt(6,:))
    sum(F10_Rnotfelt(6,:));
]

DR_notfelt = [...
    sum(D1_Rnotfelt(1,:)) sum(D2_Rnotfelt(1,:)) sum(D3_Rnotfelt(1,:))
    sum(D4_Rnotfelt(1,:)) sum(D5_Rnotfelt(1,:)) sum(D6_Rnotfelt(1,:))
    sum(D7_Rnotfelt(1,:)) sum(D8_Rnotfelt(1,:)) sum(D9_Rnotfelt(1,:))
    sum(D10_Rnotfelt(1,:)); ...
    sum(D1_Rnotfelt(2,:)) sum(D2_Rnotfelt(2,:)) sum(D3_Rnotfelt(2,:))
    sum(D4_Rnotfelt(2,:)) sum(D5_Rnotfelt(2,:)) sum(D6_Rnotfelt(2,:))
    sum(D7_Rnotfelt(2,:)) sum(D8_Rnotfelt(2,:)) sum(D9_Rnotfelt(2,:))
    sum(D10_Rnotfelt(2,:)); ...
    sum(D1_Rnotfelt(3,:)) sum(D2_Rnotfelt(3,:)) sum(D3_Rnotfelt(3,:))
    sum(D4_Rnotfelt(3,:)) sum(D5_Rnotfelt(3,:)) sum(D6_Rnotfelt(3,:))
    sum(D7_Rnotfelt(3,:)) sum(D8_Rnotfelt(3,:)) sum(D9_Rnotfelt(3,:))
    sum(D10_Rnotfelt(3,:)); ...
    sum(D1_Rnotfelt(4,:)) sum(D2_Rnotfelt(4,:)) sum(D3_Rnotfelt(4,:))
    sum(D4_Rnotfelt(4,:)) sum(D5_Rnotfelt(4,:)) sum(D6_Rnotfelt(4,:))
    sum(D7_Rnotfelt(4,:)) sum(D8_Rnotfelt(4,:)) sum(D9_Rnotfelt(4,:))
    sum(D10_Rnotfelt(4,:)); ...
    sum(D1_Rnotfelt(5,:)) sum(D2_Rnotfelt(5,:)) sum(D3_Rnotfelt(5,:))
    sum(D4_Rnotfelt(5,:)) sum(D5_Rnotfelt(5,:)) sum(D6_Rnotfelt(5,:))
    sum(D7_Rnotfelt(5,:)) sum(D8_Rnotfelt(5,:)) sum(D9_Rnotfelt(5,:))
    sum(D10_Rnotfelt(5,:)); ...
    sum(D1_Rnotfelt(6,:)) sum(D2_Rnotfelt(6,:)) sum(D3_Rnotfelt(6,:))
    sum(D4_Rnotfelt(6,:)) sum(D5_Rnotfelt(6,:)) sum(D6_Rnotfelt(6,:))
    sum(D7_Rnotfelt(6,:)) sum(D8_Rnotfelt(6,:)) sum(D9_Rnotfelt(6,:))
    sum(D10_Rnotfelt(6,:));
]

FL_notfelt = [...
sum(F1_Lnotfelt(1,:)) sum(F2_Lnotfelt(1,:)) sum(F3_Lnotfelt(1,:))
sum(F4_Lnotfelt(1,:)) sum(F5_Lnotfelt(1,:)) sum(F6_Lnotfelt(1,:))
sum(F7_Lnotfelt(1,:)) sum(F8_Lnotfelt(1,:)) sum(F9_Lnotfelt(1,:))
sum(F10_Lnotfelt(1,:)); ...
sum(F1_Lnotfelt(2,:)) sum(F2_Lnotfelt(2,:)) sum(F3_Lnotfelt(2,:))
sum(F4_Lnotfelt(2,:)) sum(F5_Lnotfelt(2,:)) sum(F6_Lnotfelt(2,:))
sum(F7_Lnotfelt(2,:)) sum(F8_Lnotfelt(2,:)) sum(F9_Lnotfelt(2,:))
sum(F10_Lnotfelt(2,:)); ...
sum(F1_Lnotfelt(3,:)) sum(F2_Lnotfelt(3,:)) sum(F3_Lnotfelt(3,:))
sum(F4_Lnotfelt(3,:)) sum(F5_Lnotfelt(3,:)) sum(F6_Lnotfelt(3,:))
sum(F7_Lnotfelt(3,:)) sum(F8_Lnotfelt(3,:)) sum(F9_Lnotfelt(3,:))
sum(F10_Lnotfelt(3,:)); ...
sum(F1_Lnotfelt(4,:)) sum(F2_Lnotfelt(4,:)) sum(F3_Lnotfelt(4,:))
sum(F4_Lnotfelt(4,:)) sum(F5_Lnotfelt(4,:)) sum(F6_Lnotfelt(4,:))
sum(F7_Lnotfelt(4,:)) sum(F8_Lnotfelt(4,:)) sum(F9_Lnotfelt(4,:))
sum(F10_Lnotfelt(4,:)); ...
sum(F1_Lnotfelt(5,:)) sum(F2_Lnotfelt(5,:)) sum(F3_Lnotfelt(5,:))
sum(F4_Lnotfelt(5,:)) sum(F5_Lnotfelt(5,:)) sum(F6_Lnotfelt(5,:))
sum(F7_Lnotfelt(5,:)) sum(F8_Lnotfelt(5,:)) sum(F9_Lnotfelt(5,:))
sum(F10_Lnotfelt(5,:)); ...
sum(F1_Lnotfelt(6,:)) sum(F2_Lnotfelt(6,:)) sum(F3_Lnotfelt(6,:))
sum(F4_Lnotfelt(6,:)) sum(F5_Lnotfelt(6,:)) sum(F6_Lnotfelt(6,:))
sum(F7_Lnotfelt(6,:)) sum(F8_Lnotfelt(6,:)) sum(F9_Lnotfelt(6,:))
sum(F10_Lnotfelt(6,:));]

DL_notfelt = [...
    sum(D1_Lnotfelt(1,:)) sum(D2_Lnotfelt(1,:)) sum(D3_Lnotfelt(1,:))
    sum(D4_Lnotfelt(1,:)) sum(D5_Lnotfelt(1,:)) sum(D6_Lnotfelt(1,:))
    sum(D7_Lnotfelt(1,:)) sum(D8_Lnotfelt(1,:)) sum(D9_Lnotfelt(1,:))
    sum(D10_Lnotfelt(1,:)); ...
    sum(D1_Lnotfelt(2,:)) sum(D2_Lnotfelt(2,:)) sum(D3_Lnotfelt(2,:))
    sum(D4_Lnotfelt(2,:)) sum(D5_Lnotfelt(2,:)) sum(D6_Lnotfelt(2,:))
    sum(D7_Lnotfelt(2,:)) sum(D8_Lnotfelt(2,:)) sum(D9_Lnotfelt(2,:))
    sum(D10_Lnotfelt(2,:)); ...
    sum(D1_Lnotfelt(3,:)) sum(D2_Lnotfelt(3,:)) sum(D3_Lnotfelt(3,:))
    sum(D4_Lnotfelt(3,:)) sum(D5_Lnotfelt(3,:)) sum(D6_Lnotfelt(3,:))
    sum(D7_Lnotfelt(3,:)) sum(D8_Lnotfelt(3,:)) sum(D9_Lnotfelt(3,:))
    sum(D10_Lnotfelt(3,:)); ...
    sum(D1_Lnotfelt(4,:)) sum(D2_Lnotfelt(4,:)) sum(D3_Lnotfelt(4,:))
    sum(D4_Lnotfelt(4,:)) sum(D5_Lnotfelt(4,:)) sum(D6_Lnotfelt(4,:))
    sum(D7_Lnotfelt(4,:)) sum(D8_Lnotfelt(4,:)) sum(D9_Lnotfelt(4,:))
    sum(D10_Lnotfelt(4,:)); ...
    sum(D1_Lnotfelt(5,:)) sum(D2_Lnotfelt(5,:)) sum(D3_Lnotfelt(5,:))
    sum(D4_Lnotfelt(5,:)) sum(D5_Lnotfelt(5,:)) sum(D6_Lnotfelt(5,:))
    sum(D7_Lnotfelt(5,:)) sum(D8_Lnotfelt(5,:)) sum(D9_Lnotfelt(5,:))
    sum(D10_Lnotfelt(5,:)); ...
    sum(D1_Lnotfelt(6,:)) sum(D2_Lnotfelt(6,:)) sum(D3_Lnotfelt(6,:))
    sum(D4_Lnotfelt(6,:)) sum(D5_Lnotfelt(6,:)) sum(D6_Lnotfelt(6,:))
    sum(D7_Lnotfelt(6,:)) sum(D8_Lnotfelt(6,:)) sum(D9_Lnotfelt(6,:))
    sum(D10_Lnotfelt(6,:));]

% PERCENTAGE NOT FELT ACROSS ALL SUBJECTS
Focused_NF_L =
    ((F1_Lnotfelt*6)+(F2_Lnotfelt*6)+(F3_Lnotfelt*6)+(F4_Lnotfelt*6)+(F5_Lnotfelt*6)
    +(F6_Lnotfelt*6)+(F7_Lnotfelt*6)+(F8_Lnotfelt*6)+(F9_Lnotfelt*6)+(F10_Lnotfelt
    *6))/60
Distracted_NF_L =
    ((D1_Lnotfelt*6)+(D2_Lnotfelt*6)+(D3_Lnotfelt*6)+(D4_Lnotfelt*6)+(D5_Lnotfelt*6)
    +(D6_Lnotfelt*6)+(D7_Lnotfelt*6)+(D8_Lnotfelt*6)+(D9_Lnotfelt*6)+(D10_Lnotfelt
    *6))/60
Distracted_NF_R =
    ((D1_Rnotfelt*6)+(D2_Rnotfelt*6)+(D3_Rnotfelt*6)+(D4_Rnotfelt*6)+(D5_Rnotfelt*6)
    +(D6_Rnotfelt*6)+(D7_Rnotfelt*6)+(D8_Rnotfelt*6)+(D9_Rnotfelt*6)+(D10_Rnotfelt
    *6))/60
Focused_NF_R =
Nonparametric_vibe_stats.m

clear all; close all; clc;

% Vibration Type Accuracy
% data = load('type_acc.csv');
% attention = typea(data);
% order = typeo(data);
% type = typet(data);
% location = typel(data);
% side = types(data);
% att_type = typeat(data);
% att_loc = typeal(data);
% type_loc = typetl(data);
% side_loc = typesl(data);

% Location Accuracy
data = load('loc_acc.csv')
attention = tab2a(data);
order = tab2o(data);
location = tab2l(data);
side = tab2s(data);
att_loc = tab2al(data);
side_loc = tab2sl(data);

[p_attention, tbl_attention, stats_attention] = signrank(attention(:,1),
attention(:,2))
means_attention = [mean(attention)' std(attention)'];

[p_ord, tbl_ord, statsord] = ranksum(order(:,1),order(:,2))
means_ord = [mean(order(:,1)) std(order(:,1)); mean(order(:,2))
std(order(:,2))];

[p_side, tbl_side, stats_side] = signrank(side(:,1), side(:,1))
means_side = [mean(side)' std(side)'];

[p_type, tbl_type, statstype] = friedman(type2,
length(type2(:,1))/10); %TYPE ACCURACY ONLY
means_type = [mean(type2)' std(type2)'];
[C_type, m_type, h_type] = multcompare(statstype, 'CType', 'bonferroni')

[p_location, tbl_location, stats_location] = friedman(location,
length(location(:,1))/10);
means_location = [mean(location)' std(location)'];
[C_location, m_location, h_location] = multcompare(stats_location, 'CType',
'bonferroni')

[p_att_type, tbl_att_type, stats_att_type] = friedman(att_type,
length(att_type(:,1))/10); %TYPE ACCURACY ONLY
means_att_type = [mean(att_type)' std(att_type)'];
[C_att_type, m_att_type, h_att_type] = multcompare(stats_att_type, 'CType',
'bonferroni')

[p_att_loc, tbl_att_loc, stats_att_loc] = friedman(att_loc,
length(att_loc(:,1))/10);
means_att_loc = [mean(att_loc)' std(att_loc)'];
[C_att_loc, m_att_loc, h_att_loc] = multcompare(stats_att_loc, 'CType', 'bonferroni')

% [p_type_loc, tbl_type_loc, stats_type_loc] = 
friedman(type_loc, length(type_loc(:,1))/10); % TYPE ACCURACY ONLY
% means_type_loc = [mean(type_loc)' std(type_loc)'];
% [C_type_loc, m_type_loc, h_type_loc] = multcompare(stats_type_loc, 'CType', 'bonferroni')
%
% [p_type_side, tbl_type_side, stats_type_side] = 
friedman(type_side, length(type_side(:,1))/10); % TYPE ACCURACY ONLY
% means_type_side = [mean(type_side)' std(type_side)'];
% [C_type_side, m_type_side, h_type_side] = multcompare(stats_type_side, 'CType', 'bonferroni')

[p_side_loc, tbl_side_loc, stats_side_loc] = 
friedman(side_loc, length(side_loc(:,1))/10);
means_side_loc = [mean(side_loc)' std(side_loc)'];
[C_side_loc, m_side_loc, h_side_loc] = multcompare(stats_side_loc, 'CType', 'bonferroni')
Calculate_Metrics.m

clear all; close all; clc;
Trial = zeros(12,1);
HeadDownTime1 = zeros(12,1);
HeadDownTime2 = zeros(12,1);
HeadDownTime3 = zeros(12,1);
HeadDownTime4 = zeros(12,1);
HeadDownTime5 = zeros(12,1);
CompletionTimes = zeros(12,1);
ObstacleCollisions = zeros(12,1);
Seconds_dropped = zeros(12,1);
Past_obstacle_times = zeros(12,1);
Toe_Off_Distances = zeros(12,1);
R_lead_toe = zeros(12,1);
L_lead_toe = zeros(12,1);
ToeClearance_lead = zeros(12,1);
R_lead_clr = zeros(12,1);
L_lead_clr = zeros(12,1);
R_crossed = zeros(12,1);
L_crossed = zeros(12,1);
Num_Steps = zeros(12,1);
Step_Length = zeros(12,1);
Step_Var = zeros(12,1);
Data_sets = {'TO'; 'VO'; 'VT'; 'NC'};
for set = 1:length(Data_sets)
    for trial = 1:12
        filename = [Data_sets{set} ' ' num2str(trial) '.csv']
        if exist(filename, 'file') ~= 2
            sprintf('WARNING! THIS FILE DOES NOT EXIST IN THE FOLDER: 
%s', filename)
            continue
        end
        data = importdata(filename, ',', 5);
        [Hdtimel, Hdtime_2, Hdtime_3, Hdtime_4, Hdtime_5, Seconds_dropped(trial),
         Past_obstacle_times(trial)] = HeadDown(data.data);
        Trial(trial) = trial;
        HeadDownTime1(trial) = Hdtime_1;
        HeadDownTime2(trial) = Hdtime_2;
        HeadDownTime3(trial) = Hdtime_3;
        HeadDownTime4(trial) = Hdtime_4;
        HeadDownTime5(trial) = Hdtime_5;

        CompletionTimes(trial) = CompletionTime(data.data);
        ObstacleCollisions(trial) = ObstacleCollision(data.data);
        [Toe_Off_Distances(trial), R_lead_toe(trial), L_lead_toe(trial)] = 
            DistanceToObstacle(data.data);
        [ToeClearance_lead(trial), R_lead_clr(trial), L_lead_clr(trial),
         R_crossed(trial), L_crossed(trial)] = ToeClearance(data.data);
        [Num_Steps(trial), Step_Length(trial), Step_Var(trial)] = 
            StepLength(data.data);

    end

eval([Data_sets{set} ' = table(Trial, Seconds_dropped, Past_obstacle_times,
        HeadDownTime1, HeadDownTime2, HeadDownTime3, HeadDownTime4, HeadDownTime5,
        CompletionTimes, ObstacleCollisions, Toe_Off_Distances, R_lead_toe, L_lead_toe,
        Num_Steps, Step_Length, Step_Var, ToeClearance_lead, R_lead_clr, L_lead_clr,
        R_crossed, L_crossed);']);

eval(['writetable(' Data_sets{set} ',"'' Data_sets{set} 'Results.csv'')']);
end

AllResults = [TO; VO; VT; NC];
writetable(AllResults, '16_Results.csv');
function [n_collisions] = ObstacleCollision(data)
time = data(:,1);
Obstacle_1 = data(:,6:8);
Obstacle_2 = data(:,9:11);
Obstacle_3 = data(:,12:14);
Obstacle_4 = data(:,15:17);
Collision_thresh = 3; [m, n] = size(Obstacle_1);

Collisons1_index_1 = [];
Collisons1_index_3 = [];
for k = 2:(m-1)
    if (abs(Obstacle_1(k,1)-Obstacle_1(k-1,1)) >= 1) &&
        (abs(Obstacle_1(k,1)-Obstacle_1(k-1,1)) < Collision_thresh)
        Collisons1_index_1 = [Collisons1_index_1 time(k+1)];
    elseif (abs(Obstacle_1(k,2)-Obstacle_1(k-1,2)) >= 1) &&
        (abs(Obstacle_1(k,2)-Obstacle_1(k-1,2)) < Collision_thresh)
        Collisons1_index_1 = [Collisons1_index_1 time(k+1)];
    elseif (abs(Obstacle_1(k,3)-Obstacle_1(k-1,3)) >= 1) &&
        (abs(Obstacle_1(k,3)-Obstacle_1(k-1,3)) < Collision_thresh)
        Collisons1_index_1 = [Collisons1_index_1 time(k+1)];
    elseif abs(Obstacle_1(k,1)-Obstacle_1(k-1,1)) >= Collision_thresh
        Collisons1_index_3 = [Collisons1_index_3 time(k+1)];
    elseif abs(Obstacle_1(k,2)-Obstacle_1(k-1,2)) >= Collision_thresh
        Collisons1_index_3 = [Collisons1_index_3 time(k+1)];
    elseif abs(Obstacle_1(k,3)-Obstacle_1(k-1,3)) >= Collision_thresh
        Collisons1_index_3 = [Collisons1_index_3 time(k+1)];
    end
end

Collisons2_index_1 = [];
Collisons2_index_3 = [];
for k = 2:(m-1)
    if (abs(Obstacle_2(k,1)-Obstacle_2(k-1,1)) >= 1) &&
        (abs(Obstacle_2(k,1)-Obstacle_2(k-1,1)) < 3)
        Collisons2_index_1 = [Collisons2_index_1 time(k+1)];
    elseif (abs(Obstacle_2(k,2)-Obstacle_2(k-1,2)) >= 1) &&
        (abs(Obstacle_2(k,2)-Obstacle_2(k-1,2)) < 3)
        Collisons2_index_1 = [Collisons2_index_1 time(k+1)];
    elseif (abs(Obstacle_2(k,3)-Obstacle_2(k-1,3)) >= 1) &&
        (abs(Obstacle_2(k,3)-Obstacle_2(k-1,3)) < 3)
        Collisons2_index_1 = [Collisons2_index_1 time(k+1)];
    elseif abs(Obstacle_2(k,1)-Obstacle_2(k-1,1)) >= 3
        Collisons2_index_3 = [Collisons2_index_3 time(k+1)];
    elseif abs(Obstacle_2(k,2)-Obstacle_2(k-1,2)) >= 3
        Collisons2_index_3 = [Collisons2_index_3 time(k+1)];
    elseif abs(Obstacle_2(k,3)-Obstacle_2(k-1,3)) >= 3
        Collisons2_index_3 = [Collisons2_index_3 time(k+1)];
    end
end

Collisons3_index_1 = [];
Collisons3_index_3 = [];
for k = 2:(m-1)
    if (abs(Obstacle_3(k,1)-Obstacle_3(k-1,1)) >= 1) &&
        (abs(Obstacle_3(k,1)-Obstacle_3(k-1,1)) < 3)
        Collisons3_index_1 = [Collisons3_index_1 time(k+1)];
    elseif (abs(Obstacle_3(k,2)-Obstacle_3(k-1,2)) >= 1) &&
        (abs(Obstacle_3(k,2)-Obstacle_3(k-1,2)) < 3)
        Collisons3_index_1 = [Collisons3_index_1 time(k+1)];
    elseif (abs(Obstacle_3(k,3)-Obstacle_3(k-1,3)) >= 1) &&
        (abs(Obstacle_3(k,3)-Obstacle_3(k-1,3)) < 3)
        Collisons3_index_1 = [Collisons3_index_1 time(k+1)];
    elseif abs(Obstacle_3(k,1)-Obstacle_3(k-1,1)) >= 3
        Collisons3_index_3 = [Collisons3_index_3 time(k+1)];
    elseif abs(Obstacle_3(k,2)-Obstacle_3(k-1,2)) >= 3
        Collisons3_index_3 = [Collisons3_index_3 time(k+1)];
    elseif abs(Obstacle_3(k,3)-Obstacle_3(k-1,3)) >= 3
        Collisons3_index_3 = [Collisons3_index_3 time(k+1)];
    end
end

n_collisions = size(Collisons1_index_1,1) + size(Collisons2_index_1,1) + size(Collisons3_index_1,1);
elseif abs(Obstacle_3(k,1)-Obstacle_3(k-1,1)) >= 3
    Collisions3_index_3 = [Collisions3_index_3 time(k+1)];
elseif abs(Obstacle_3(k,2)-Obstacle_3(k-1,2)) >= 3
    Collisions3_index_3 = [Collisions3_index_3 time(k+1)];
elseif abs(Obstacle_3(k,3)-Obstacle_3(k-1,3)) >= 3
    Collisions3_index_3 = [Collisions3_index_3 time(k+1)];
end
end

Collisons_1 = [Collisons1_index_1 Collisons2_index_1 Collisons3_index_1];
[m_c1,n_c1] = size(Collisons_1);
Collisons_3 = [Collisons1_index_3 Collisons2_index_3 Collisons3_index_3];
[m_c3,n_collisions] = size(Collisons_3);
end

HeadDown.m
function [Hdtimel, Hdtime_2, Hdtime_3, Hdtime_4, Hdtime_5, secs_dropped, past_obs_time] = HeadDown(data)
    [data, secs_dropped] = cleandata(data, [48:56]);
    time = data(:,1);
    Head_1 = data(:,48:50);
    Head_2 = data(:,51:53);
    Head_3 = data(:,54:56);
    Obstacle_1_y = data(1,7);

    h1 = transpose(Head_1(1:25,:));
    hv1 = mean(transpose(h1));
    h2 = transpose(Head_2(1:25,:));
    hv2 = mean(transpose(h2));
    h3 = transpose(Head_3(1:25,:));
    hv3 = mean(transpose(h3));
    head_vlr = hv1 - hv3;
    head_v2r = hv2 - hv3;
    normal_v = cross(head_vlr, head_v2r);

    Hdtime_1 = 0;
    Hdtime_2 = 0;
    Hdtime_3 = 0;
    Hdtime_4 = 0;
    Hdtime_5 = 0;

    [m, n] = size(Head_1);
    past_obs_row = m;
    found_obs_row = false;
    for i = 1:m
        if isnan(Head_3(i,2))
            continue
        elseif (Head_3(i,2) > Obstacle_1_y)
            past_obs_row = i;
            past_obs_time = time(past_obs_row)/100.0;
            found_obs_row = true;
            break
        end
    end
    if found_obs_row == false;
        for i = 1:m
            if isnan(Head_2(i,2))
                continue
            elseif (Head_2(i,2) > Obstacle_1_y)
                past_obs_row = i;
                past_obs_time = time(past_obs_row)/100.0;
                break
            end
        end
    end
end
\[
\text{found_obs_row} = \text{true}; \\
\text{break}
\]
\end{verbatim}
end
end
end

if found_obs_row == false;
\text{disp('Error in HeadDown: Never found past\_obs\_row');}
end

\text{tilt\_ang\_list} = \text{zeros(past\_obs\_row,1)};
\begin{verbatim}
for i = 1:past\_obs\_row
\text{head\_v1} = \text{Head\_1(i,:) - Head\_3(i,:)};
\text{head\_v2} = \text{Head\_2(i,:) - Head\_3(i,:)};
\text{head\_vn} = \text{\text{cross(head\_v1, head\_v2)}};
\text{head\_dot} = \text{dot(head\_vn, normal\_v)};
\text{head\_cross} = \text{\text{cross(head\_vn, normal\_v)}};
\text{tilt\_ang} = \text{(atan2(norm(head\_cross),head\_dot))*180/pi};
\text{if tilt\_ang} <= \text{10}
\text{Hdtime\_1} = \text{(Hdtime\_1)+(1/100)};
\text{elseif tilt\_ang} <= \text{20}
\text{Hdtime\_2} = \text{(Hdtime\_2)+(1/100)};
\text{elseif tilt\_ang} <= \text{30}
\text{Hdtime\_3} = \text{(Hdtime\_3)+(1/100)};
\text{elseif tilt\_ang} <= \text{60}
\text{Hdtime\_4} = \text{(Hdtime\_4)+(1/100)};
\text{else}
\text{Hdtime\_5} = \text{(Hdtime\_5)+(1/100)};
\text{end}
\text{tilt\_ang\_list}(i) = \text{tilt\_ang};
\end{verbatim}
end
end

\textbf{DistanceToObstacle.m}
\begin{verbatim}
function \text{[Ob_dist, R\_lead, L\_lead]} = \text{DistanceToObstacle(data)}
\text{time} = \text{data(:,1)};
\text{FootR} = \text{data(:,18:20)};
\text{FootL} = \text{data(:,27:29)};
\text{FootR\_z} = \text{FootR(:,3)};
\text{FootL\_z} = \text{FootL(:,3)};
\text{Obstacle\_3} = \text{data(:,12:14)};
\text{Obstacle\_4} = \text{data(:,15:17)};
\text{FloorR} = \text{\text{min(FootR\_z(FootR\_z>0))}};
\text{FloorL} = \text{\text{min(FootL\_z(FootL\_z>0))}};
\text{R\_lead} = \text{0};
\text{L\_lead} = \text{0};
\text{z\_error} = \text{25};
\text{num} = \text{1};
\text{[m,n]} = \text{size(time)};
\text{\%Obstacle marker 3}
\text{while num <= 10 \&\& R\_lead == 0 \&\& L\_lead == 0}
\text{for j = 1:m}
\text{if (FootR(j,2) >= Obstacle\_4(j,2))}
\text{i = j-1;}
\text{while i<j}
\text{if FootR(i,3) <= (FloorR+z\_error)}
\text{R\_lead} = \text{1};
\text{\text{\%Obstacle marker 4}}
\text{end}
\text{end}
\text{end}
\end{verbatim}
Ob_index = i;
Ob_foot = FootR(i,:);
R_lead = 1;
break
end
i = i-1;
if i == 0
  z_error = z_error + 1
  num = num + 1;
break
end
break

elseif FootL(j,2) >= Obstacle_4(j,2)
i = j-1;

if i == 0
  z_error = z_error + 1
  num = num + 1;
break
end
while i<j
  if FootL(i,3) <= (FloorL+z_error)
    Ob_index = i;
    Ob_foot = FootL(i,:);
    L_lead = 1;
    break
  end
  i = i-1;
  if i == 0
    z_error = z_error + 1
    num = num + 1;
    break
  end
end
break

end
break

end
if L_lead, break, end
if R_lead, break, end

end

end

if R_lead == 0 && L_lead == 0
  sprintf('Error in DistanceToObstacle: Neither foot detected as lead, outputs invalid')
end

Ob_dist = abs(Obstacle_3(Ob_index,2)-Ob_foot(1,2));
Ob_time_stamp = time(Ob_index);
End

CompletionTime.m
function Completion_time = CompletionTime(data)
time = data(:,1);
FootR = data(:,18:20);
FootL = data(:,27:29);
EndMarker = data(1,3:5);
[m, n] = size(FootR);
Completion_frame = 0;

count = 0;
while count<1
    for j = 1:m
        if -isnan(FootR(j,2)) && (FootR(j,2) >= EndMarker(2))
            Completion_frame = time(j);
            break
        elseif -isnan(FootL(j,2)) && (FootL(j,2) >= EndMarker(2))
            Completion_frame = time(j);
            break
        end
    end
    count = 1;
end
Completion_time = Completion_frame/100.0;

ToeClearance.m

function [clearance, R_lead, L_lead, R_crossed, L_crossed] = ToeClearance(data)

time = data(:,1);
FootR = data(:,18:20); %right boot toe marker
FootL = data(:,27:29); %left boot toe marker
Obstacle_1 = data(:,6:8); Obs1_z = Obstacle_1(:,3);
Obstacle_2 = data(:,9:11); Obs2_z = Obstacle_2(:,3);
Obstacle_3 = data(:,12:14); Obs3_z = Obstacle_3(:,3);
Obs3_y = Obstacle_3(:,2);
Obstacle_4 = data(:,15:17); Obs4_z = Obstacle_4(:,3);
Obs_z = max([mean(Obs1_z(-isnan(Obs1_z))), mean(Obs2_z(-isnan(Obs2_z))),
              mean(Obs3_z(-isnan(Obs3_z))), mean(Obs4_z(-isnan(Obs4_z))))];

[m,n] = size(time);
toe_clearanceR = zeros(m,1);
toe_clearanceL = zeros(m,1);
L_lead = 0;
R_lead = 0;
reach_obs_R = 0;
reach_obs_L = 0;
R_reached = 0;
L_reached = 0;
R_crossed = 0;
L_crossed = 0;

 clearance = 0;

for i = 1:m
    x_max = max(Obstacle_2(i,1), Obstacle_3(i,1));
    x_min = min(Obstacle_1(i,1), Obstacle_4(i,1));
    y_max = max(Obstacle_1(i,2), Obstacle_2(i,2));
    y_min = min(Obstacle_3(i,2), Obstacle_4(i,2));

    if ((FootR(i,2) >= y_min) && (FootR(i,2) <= y_max))
        toe_clearanceR(i) = FootR(i,3) - Obs_z;
        R_reached = 1;
        if (FootR(i,1) >= x_min) && (FootR(i,1) <= x_max)
            R_crossed = 1;
        end
    end

    if (FootL(i,2) >= y_min) && (FootL(i,2) <= y_max)
        toe_clearanceL(i) = FootL(i,3) - Obs_z;
    end

end
L_reached = 1;
if (FootL(i,1) >= x_min) && (FootL(i,1) <= x_max)
    L_crossed = 1;
end
end

%Determine when each foot first crossed over obs to determine lead/trail
if R_reached == 1
    reach_obs_R = find(toe_clearanceR,1);
end
if L_reached == 1
    reach_obs_L = find(toe_clearanceL,1);
end

if R_reached == 1 && L_reached == 1
    if reach_obs_R > reach_obs_L
        toe_clearance_lead = toe_clearanceL;
        toe_clearance_trail = toe_clearanceR;
        L_lead = 1;
    elseif reach_obs_R < reach_obs_L
        toe_clearance_lead = toe_clearanceR;
        toe_clearance_trail = toe_clearanceL;
        R_lead = 1;
    end

elseif R_reached == 1 && L_reached == 0
    toe_clearance_lead = toe_clearanceR;
    toe_clearance_trail = toe_clearanceL;
    R_lead = 1;
end
elseif R_reached == 0 && L_reached == 0
    sprintf('Error in ToeClearance: Neither foot crossed obstacle y position')
end
clearance = min(toe_clearance_lead(toe_clearance_lead==0));
end

StepLength.m
function [NumSteps,StepLength,StepVar] = StepLength(data)
time = data(:,1);
FootR = data(:,18:20);
FootL = data(:,27:29);
Obs_3y = data(1:250,13); Obs3 = mean(Obs_3y(~isnan(Obs_3y)));
Obs_4y = data(1:250,16); Obs4 = mean(Obs_4y(~isnan(Obs_4y)));
Obs_y = mean([Obs3, Obs4]); % Obs y location

[MinR, IndR] = findpeaks(FootR(:,2), 'MinPeakDistance', 110);
[MinL, IndL] = findpeaks(FootL(:,2), 'MinPeakDistance', 110);
%Find steps
StepsR = MinR(MinR < Obs_y);
StepsL = MinL(MinL < Obs_y);
NumStepsR = length(StepsR)-1;
NumStepsL = length(StepsL)-1;
NumSteps = NumStepsR + NumStepsL;
Step_lengthsR = zeros(1, length(NumStepsR));
Step_lengthsL = zeros(1, length(NumStepsL));

%Calculate step lengths
for i = 2:length(StepsR)
    Step_lengthsR(i-1) = abs(StepsR(i) - StepsR(i-1));
end
for i = 2:length(StepsL)
    Step_lengthsL(i-1) = abs(StepsL(i) - StepsL(i-1));
end
Step_lengths = [Step_lengthsR, Step_lengthsL];
StepLength = mean(Step_lengths);
StepVar = std(Step_lengths);

CleanData.m

function [data, nDropped] = cleanData(data, specialCols)

nDropped = -1;
minRow = 1;
for i = 1:length(specialCols)
    [ix, iy] = find(~isnan(data(:, specialCols(i))), 1);
    minRow = max(minRow, ix);
end
data = data(minRow:end,:);
nDropped = (minRow-1)/100.0;

RMANOVA_stats.m

clear all; close all; clc;
num_subjects = 16;
order_assignments = {'B' 'A' 'A' 'A' 'B' 'B' 'A' 'B' 'A' 'B' 'A' 'B' 'A' 'B' 'A'};
sex_assignments = {'M' 'F' 'M' 'M' 'F' 'F' 'M' 'M' 'M' 'F' 'F' 'M' 'M' 'M'};
leg_lengths = [1011.42 1078.35 942.13 959.09 903.93 918.89 824.91 1028.57
960.67 967.07 919.5 967.07 966.017 1032.93 988.78];

% ORDER
order = cell(1, 48*num_subjects);
order2v = zeros(48*num_subjects, 1);
for i = 1:num_subjects
    order(((i-1)*48+1):(48*i)) = order_assignments(i);
    order2v(((i-1)*48+1):(48*i)) = order2v((i-1)*48+1):
end

% SEX
sex = cell(1, 48*num_subjects);
sex2v = zeros(48*num_subjects, 1);
for i = 1:num_subjects
    sex(((i-1)*48+1):(48*i)) = sex_assignments(i);
    sex2v(((i-1)*48+1):(48*i)) = sex2v((i-1)*48+1):
end
% SUBJECT
subject2v = zeros(48*num_subjects, 1);
for i = 1:num_subjects
    subject(((i-1)*48+1):(48*i)) = i;
    subject2v(((i-1)*48+1):(48*i)) = i;
end

% OBSTACLE LOCATION, SIZE, AND DISPLAY TYPE ORDER FOR ALL 48 TRIALS
location_order = [3 3 1 1 2 2 3 2 1 1 2 3 1 3 1 1 3 2 2 3 3 2 3 3 1 2 2 1 3 1 1 2 3];
size_order = {'L' 'S' 'S' 'L' 'L' 'S' 'S' 'S' 'L' 'L' 'S' 'S' 'L' 'L' 'S' 'S' 'L' 'L' 'L' 'L'
              'L' 'S' 'S' 'L' 'L' 'S' 'S' 'L' 'L' 'S' 'S' 'L' 'L' 'S' 'S' 'L' 'L' 'L' 'L';
display_order = {'TO' 'TO' 'TO' 'TO' 'TO' 'TO' 'TO' 'TO' 'TO' 'TO' 'TO' 'TO' 'TO' 'TO'
                 'TO' 'TO' 'TO' 'TO' 'TO' 'TO' 'TO' 'TO' 'TO' 'TO' 'TO' 'TO' 'TO' 'TO';
location = repmat(location_order, 1, num_subjects);
size = repmat(size_order, 1, num_subjects);
display = repmat(display_order, 1, num_subjects);

% -------------- LOAD DATA-----------------
datalength = 48*num_subjects;
hdtime = zeros(1, datalength);
completiontime = zeros(1, datalength);
collisions = zeros(1, datalength);
toeoffdist = zeros(1, datalength);
umsteps = zeros(1, datalength);
stepnorm = zeros(1, datalength);
stepvar = zeros(1, datalength);
toe-clearance = zeros(1, datalength);
R_lead = zeros(16, 48);
L_lead = zeros(1, datalength);

for i = 1:num_subjects
    filename = [num2str(i) 'Results.csv'];
data_file = importdata(filename,';',1);
data = data_file.data;
    leglen = leglengths(i);
    hdtimesub = zeros(1, 48);
    CTnormalizedsub = zeros(1,48);
collisionsub = zeros(1,48);
    for trial = 1:48
        hdtimesub(trial) = (((data(trial,5)+data(trial,6)+(data(trial,7))/data(trial,9)))/100.0;
        CTnormalizedsub(trial) = (data(trial,9)/(max(data(:,9))))*100.0;
        if data(trial,10) > 5
            collisionsub(trial) = 1;
        else
            collisionsub(trial) = 0;
        end
    end
    hdtime(((i-1)*48+1):(48*i)) = hdtimesub;
    CT_normalized(((i-1)*48+1):(48*i)) = CTnormalizedsub;
end
completion_time(((i-1)*48+1):(48*i)) = data(:,9)';
collisions(((i-1)*48+1):(48*i)) = collisions_sub;
toe_off_dist(((i-1)*48+1):(48*i)) = data(:,11)';
num_steps(((i-1)*48+1):(48*i)) = data(:,14)';
lead_foot(((i-1)*48+1):(48*i)) = data(:,14)';
num_steps_norm(((i-1)*48+1):(48*i)) = ((data(:,14)')/leg_len)*100;
step_length(((i-1)*48+1):(48*i)) = data(:,15)';
step_length_norm(((i-1)*48+1):(48*i)) = ((data(:,15)')/leg_len)*100;
stepvar(((i-1)*48+1):(48*i)) = data(:,16)';
stepvar_norm(((i-1)*48+1):(48*i)) = ((data(:,16)')/leg_len)*100;
toeclearance(((i-1)*48+1):(48*i)) = data(:,17)';
R_lead(i,1:48) = data(:,12)';
L_lead(((i-1)*48+1):(48*i)) = data(:,13)';

end

toe_off_dist_inches = (toe_off_dist-70)*0.0393701;
toe_clearance_inches = (toe_clearance-100)*0.0393701;

% % % %---------------------- Repeated Measures ANOVA ---------------------
% % % Matrix M showing order and sex nested within subject, subject is random variable
M = zeros(6,6); M(1,5) = 1; M(1,6) = 1;

% Completion Time Normalized
[p_CTNorm, tbl_CTNorm, stats_CTNorm] = anovan(CT_normalized, {subject, location, size, display, order, sex}, 'model', 'interaction', 'random', [1], 'nested', M, 'varnames', {'SUBJECT', 'Location', 'Size', 'Display', 'Order', 'Sex'})

% Head-Down Percentage
[p_HD, tbl_HD, stats_HD] = anovan(hd_time, {subject, location, size, display, order, sex}, 'model', 'interaction', 'random', [1], 'nested', M, 'varnames', {'SUBJECT', 'Location', 'Size', 'Display', 'Order', 'Sex'})

nonparam_stats.m

clear all; close all; clc;
num_subjects = 16;
new_order = [1 1 5 1 7 10 9 6 4 2 8 3 12 19 17 15 14 13 24 16 20 21 28 31 27 36 26 29 25 32 30 33 34 35 41 43 42 89 45 46 38 47 37 44 40 48];

%DATA REFORMATTED
completion_time2 = zeros(16,48);
hd_time2 = zeros(16,48);
collisions2 = zeros(16,48);
toe_off_dist2 = zeros(16,48);
num_steps2 = zeros(16,48);
num_steps_norm2 = zeros(16,48);
step_length2 = zeros(16,48);
step_length_norm2 = zeros(16,48);
stepvar2 = zeros(16,48);
step_var_norm2 = zeros(16,48);
toeclearance2 = zeros(16,48);

leg_lengths = [1011.42 1078.35 942.13 959.09 903.93 918.89 824.91 1028.57
960.67 967.07 919.5 922.07 825.98 966.017 1032.93 988.78];

for i = 1:num_subjects
    filename = [num2str(i) ' Results.csv'];
data_file = importdata(filename,';',1);
data = data_file.data;
hd_time_sub = zeros(1, 48);
CT_normalized_sub = zeros(1, 48);
collisions_sub = zeros(1, 48);
leg_len = leg_lengths(i);

for trial = 1:48
    hd_time_sub(trial) = ((data(trial,5)+data(trial,6)+data(trial,7))/data(trial,9))*100.0;
    CT_normalized_sub(trial) = (data(trial,9)/(max(data(:,9))))*100.0;
    if data(trial,10) > 5
        collisions_sub(trial) = 1;
    else
        collisions_sub(trial) = 0;
    end
end

k = zeros(1,1);
for j = 1:48
    k = find(new_order==j);
    completion_time2(i,j) = data(k,9);
    Norm_completion_time2(i,j) = CT_normalized_sub(k);
    hd_time2(i,j) = hd_time_sub(k);
    collisions2(i,j) = collisions_sub(k);
    toe_off_dist2(i,j) = data(k,11);
    num_steps2(i,j) = data(k,14);
    num_steps_norm2(i,j) = ((data(k,14))/leg_len)*100;
    step_length2(i,j) = data(k,15);
    step_length_norm2(i,j) = ((data(k,15))/leg_len)*100;
    step_var2(i,j) = data(k,16);
    step_var_norm2(i,j) = ((data(k,16))/leg_len)*100;
    toe_clearance2(i,j) = data(k,17);
end

% ---------------- NON-PARAMETRIC TESTS -----------------

% Normalized Completion Time
% display = reformat_friedman(Norm_completion_time2);
% order = reformat_friedman3(completion_time2);
% size = reformat_friedman5(Norm_completion_time2);
% disporder = reformat_friedman2(completion_time2);
% dispsize = reformat_friedman4(Norm_completion_time2);

% Head-Down Time
% display = reformat_friedman(hd_time2);
% order = reformat_friedman3(hd_time2);
% size = reformat_friedman5(hd_time2);
% disporder = reformat_friedman2(hd_time2);
% dispsize = reformat_friedman4(hd_time2);

% Toe-Off Distance
% display = reformat_friedman(toe_off_dist2);
% order = reformat_friedman3(toe_off_dist2);
% size = reformat_friedman5(toe_off_dist2);
% disporder = reformat_friedman2(toe_off_dist2);
% dispsize = reformat_friedman4(toe_off_dist2);

% Toe Clearance
% display = reformat_friedman(toe_clearance2);
% order = reformat_friedman3(toe_clearance2);
% size = reformat_friedman5(toe_clearance2);
surveysr
clear all; close all; clc;
num_subjects = 16;
order_assignments = {'B' 'A' 'A' 'A' 'B' 'B' 'A' 'B' 'B' 'A' 'B' 'A' 'B' 'B' 'A'};
sex_assignments = {'M' 'F' 'M' 'M' 'F' 'F' 'F' 'M' 'M' 'M' 'F' 'F' 'F' 'M' 'M'};

% ORDER
order = cell(1,4*num_subjects);
for i=1:num_subjects
    order(((i-1)*4+1):(4*i)) = order_assignments(i);
end

% SEX
sex = cell(1,4*num_subjects);
for i=1:num_subjects
    sex(((i-1)*4+1):(4*i)) = sex_assignments(i);
end

% OBSTACLE LOCATION, SIZE, AND DISPLAY TYPE
display_order_work = {'TO' 'VO' 'VT' 'NC'};
display_work = repmat(display_order_work, 1, num_subjects);

% Friedman test on TLX overall workload scores
tlx_data = load('TLX_ANOVA2_data.csv');
[p_tlx, tbl_tlx, stats_tlx] = friedman(tlx_data,1);
[C_tlx, m_tlx, h_tlx] = multcompare(stats_tlx, 'CTYPE','bonferroni');

% Friedman tests for six workload measures in TLX
tlx_all_data = load('TLX_WithinDisplay.csv');
[p_mental, tbl_mental, stats_mental] = friedman(tlx_all_data(:,1:6:end),1);
[C_mental, m_mental, h_mental] = multcompare(stats_mental, 'CTYPE','bonferroni');

%Friedman test on survey workload rankings
surv_data = load('Survey_Workload.csv');
[p_surv, tbl_surv, stats_surv] = friedman(surv_data,1);
[C_surv, m_surv, h_surv] = multcompare(stats_surv, 'CTYPE', 'bonferroni');

% Friedman test on Survey Display Preference Data (since ranked)
pref_data = load('preference_rankings.csv');
[p_pref, tbl_pref, stats_pref] = friedman(pref_data,1);
[Cpref, m_pref, h_pref] = multcompare(stats_pref, 'CTYPE', 'bonferroni');

%Wilcoxon Signed Rank for display cues intuitiveness
visual_int = [4 5 4 4 5 5 4 4 3 3 2 5 4 3 3 4];
tactile_int = [2 2 4 2 3 3 4 1 4 4 2 4 4 4 3 2];
[p_intuitive, tbl_intuitive, stats_intuitive] = signrank(visual_int, tactile_int);
all_int = [visual_int ' tactile_int'];
visual_int_mean = mean(visual_int); visual_std = std(visual_int);
tactile_int_mean = mean(tactile_int); tactile_std = std(tactile_int);
Bibliography


[32] Fabrizio Sergi, Dino Accoto, Domenico Campolo, and Eugenio Guglielmelli. Forearm orientation guidance with a vibrotactile feedback bracelet: On the direction-


[70] C Cullinane, R Rhodes, and L Stirling. Mobility and agility during locomotion in the mark iii space suit.


