DESIGN, CHARACTERIZATION, AND TESTING OF SKIN-STRETCH FEEDBACK INTEGRATED INTO A GAME CONTROLLER

by

Nathaniel Alexis Caswell

A thesis submitted to the faculty of
The University of Utah
in partial fulfillment of the requirements for the degree of

Master of Science

Department of Mechanical Engineering

The University of Utah

December 2013

THE UNIVERSITY OF UTAH GRADUATE SCHOOL

STATEMENT OF THESIS APPROVAL

The thesis of Nathaniel Alexis Caswell

has been approved by the following supervisory committee members:

William R. Provancher	, Chair	6/17/2013 Date Approved
Jake J. Abbott	, Member	6/18/2013 Date Approved
Stephen A. Mascaro	, Member	6/17/2013 Date Approved

and by Tim A. Ameel, Chair of

the Department of Mechanical Engineering

and by David B. Kieda, Dean of The Graduate School.

ABSTRACT

Haptic feedback in modern game controllers is limited to vibrotactile feedback. The addition of skin-stretch feedback would significantly improve the type and quality of haptic feedback provided by game controllers.

Skin-stretch feedback requires small forces (around a few newtons) and translations (as small as 0.5 mm) to provide identifiable direction cues. Prior work has developed skin-stretch mechanisms in two form factors: a flat form factor and a tall but compact (cubic) form factor. These mechanisms have been shown to be effective actuators for skin-stretch feedback, and are small enough to fit inside of a game controller. Additional prior work has shown that the cubic skin-stretch mechanism can be integrated into a thumb joystick for use with game controllers.

This thesis presents the design, characterization, and testing of two skin-stretch game controllers. The first game controller provides skin stretch via a 2-axis mechanism integrated into its thumb joysticks. This controller uses the cubic skin-stretch mechanism to drive the skin stretch. Concerns that users' motions of the joystick could negatively impact the saliency of skin stretch rendered from the joystick prompted the design of a controller that provides 2-axis skin stretch to users' middle fingers on the back side of the controller.

Two experiments were conducted with the two controllers. One experiment had participants identify the direction of skin stretch from a selection of 8 possible directions.

This test compared users' accuracies with both controllers, and with five different finger restraints on the back-tactor controller. Results show that users' identification accuracy was similar across feedback conditions.

A second experiment used skin stretch to rotationally guide participants to a randomized target angle. Three different feedback strategies were tested. Results showed that a strategy called sinusoidal feedback, which provided feedback that varied in frequency and amplitude as a function of the user's relative position to the tactor, performed significantly better on all performance metrics than the other feedback strategies. It is important to note that the sinusoidal feedback only requires two 1-axis skin-stretch actuators, which are spatially separated, in order to provide feedback. The other lower performing feedback strategies used two 2-axis skin-stretch actuators.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

This work was supported, in part, by the National Science Foundation under awards IIS-0746914 and IIS-0904456. Additional support was provided by the University of Utah Research Foundations Technology Commercialization Project.

1 INTRODUCTION

1.1 Thesis Overview

Haptic feedback in modern game controllers is currently limited to vibrotactile feedback. Vibrotactile feedback is widely used in video games to communicate in a wide range of situations. A few situations in which vibrotactile feedback is used include the sensation of: collision, taking enemy fire, nearby explosions, vehicle engine rumble, impact of landing on the ground, and firing a weapon. Because the vibrotactor (eccentric mass) motors enclosed in the game controller are used to communicate these situations, the information must be conveyed by varying the amplitude of vibration, which limits the patterns of touch feedback that can be presented. This is a very one-dimensional form of feedback.

Because game controllers use such a limited form of feedback, games and game developers are limited to conveying only simple information via touch. More detailed information must be communicated via vision or audio. If game controllers were capable of conveying more detailed haptic information, games could be more immersive and rely less on visual and audio communication channels, and more evenly distribute information to users.

Currently force-feedback devices are the preferred and accepted method for providing additional haptic information beyond that provided by vibration. While force-feedback devices are able to provide rich haptic information to users, these devices are too large to

fit inside of modern game controllers. If a force-feedback device were miniaturized to fit inside of a game controller, certain drawbacks of force feedback could limit the practical benefits of the addition of force feedback. Currently existing force-feedback devices for games deliver forces to the user through an input device such as a steering wheel or joystick. Note that in these cases the force feedback itself can influence the user's input to the device. So while some users may find this influence to be an acceptable drawback in exchange for the additional haptic information, others may not.

Ideally, we would like to present users with detailed haptic information without reducing their ability to precisely provide input. Skin-stretch feedback has the potential to do this. Skin-stretch feedback is a form of tactile feedback where a contactor placed against the user's skin is displaced in the plane of the skin by a few millimeters. Furthermore, people are generally able to perceive the direction of these skin-stretch cues. Hence, skin-stretch feedback can convey information similar to that conveyed by force-feedback devices. The primary difference between skin-stretch feedback and force feedback is that skin stretch usually applies smaller forces to a limited section of users' skin, while force feedback applies larger forces directly via the input device to the user's hand.

Lower amounts of force exerted by skin stretch means that skin-stretch feedback is less likely to affect a user's ability to provide precise input. Also, since skin stretch only requires small amounts of force, much smaller actuators can be used compared to force feedback. This makes the integration of skin stretch into game controllers more practical than force feedback from a device design perspective.

The small size of the actuators required to stretch skin means that the skin-stretch actuators can be oriented inside of the game controller in many orientations. Skin stretch could be integrated into the input device common in game controllers, the thumb joystick. This would be the skin-stretch equivalent of a force-feedback joystick. Integrating skin-stretch feedback in the thumb joystick of a game controller would allow users to receive feedback and provide input to the device at the same location. The co-location of feedback and input could be advantageous because it focuses the user's attention. A possible drawback is that the skin-stretch feedback could be difficult to sense while the user is moving the thumb joystick.

Instead of rendering skin-stretch feedback at the thumb joystick, we could instead render the feedback to another part of the hand. Thumbs and index fingers are commonly used to move the thumb joystick and press buttons, bumpers, and triggers, so they are not the most advantageous locations for integrating skin-stretch feedback.

The middle finger often rests on the back of game controllers, beneath the thumb joysticks on the front of the controller. The skin stretch applied to the middle finger on the back side of the controller could be oriented so that the skin-stretch plane matches the plane of motion of the thumb joystick, which should allow users to map the feedback to the plane of their thumb joystick inputs.

Users' palms are also usually stationary and in contact with the game controller. However, the palm contacts the controller at the handles, which means that skin-stretch feedback rendered at this location would stretch the skin in directions that do not match the plane of motion of the thumb joystick and users may have difficulty interpreting

feedback at these locations. That is, skin stretch applied to the palms could convey direction information out of the thumb joystick's plane of motion.

This thesis details the design, verification, and testing of two game controllers that provide skin-stretch feedback. One controller provides 2-axis skin stretch to both the user's thumbs at the thumb joysticks, which will be referred to as the front-tactor controller. The other controller provides 2-axis skin stretch to both of the user's middle fingers on the back of the device as well as 1-axis skin stretch to both of the user's palms at the handles, which will be referred to as the back-tactor controller. Both controllers are equipped with thumb joysticks, buttons, triggers, and bumpers so that they mimic the input capabilities of modern game controllers.

The skin-stretch mechanism in the front-tactor controller uses two servos to actuate a 2-axis flexible hinge, which will be referred to as a "flexure stage." The output of the flexure stage is attached to a ThinkPad TrackPoint cap (to be referred to as a "tactor") that interacts with the user's thumb. The tactor has a texture that generates high levels of friction between itself and skin. The skin-stretch mechanism is mounted inside of an instrumented 2-axis gimbal that allows the skin-stretch mechanism to operate as a thumb joystick input.

The skin-stretch mechanism for the back side of the back-tactor controller uses two servos that actuate a sliding plate via spring steel wires. The sliding plate is constrained to planar motion and is mounted to a tactor that is used to stretch the skin on the user's middle finger. The skin-stretch mechanism that renders skin stretch to the user's palm is actuated by a single servo that is directly attached to a tactor wheel. The tactor wheel is a textured wheel that stretches the user's skin as it rotates.

The skin-stretch mechanism in the front-tactor controller has a large amount of compliance and hysteresis between be the servos and the tactor. This compliance causes there to be error between the desired tactor position and the actual tactor position when a user's skin is being stretched. This error is minimized by trying to move the tactor a set amount further than the desired position to compensate for device hysteresis and compliance.

The skin-stretch mechanism for the back side of the back-tactor controller has very low levels of backlash and compliance. The mechanism's design that uses two servos attached to a sliding tactor plate via spring steel wires, however, has nonlinear kinematics. As one servo moves, the sliding plate moves along that servo's axis, and the sliding plate also rotates about the other servo. Accurate control of the motion of the tactor is achieved by using bi-linear interpolation of a lookup table comprised of servo positions and tactor positions to correct for the nonlinear kinematics.

The relative effectiveness of the skin-stretch game controllers at rendering direction information is tested using an 8-direction identification task. The effects of several finger restraints for the back-tactor controller on effectiveness were also evaluated. Directional skin-stretch cues were rendered to participants in one of 8 directions, and participants used the thumb joystick to identify the direction of the skin-stretch cue. The results showed no significant differences in participant response accuracy between either of the skin-stretch game controllers. This result is significant as it indicates that both feedback locations are equally capable of conveying direction information.

Finally, we evaluate how effective the skin-stretch game controllers are at providing targeting information. Three different feedback strategies are rendered to the front-and

back-tactor controllers and one feedback strategy is rendered to the palm tactors. These feedback strategies all guide the participants to randomized target angles ("targets"). Results show that one of the feedback strategies could guide participants to the targets quickly, efficiently, and consistently. This feedback moved the right or left tactor of each controller in a sinusoidal motion that increased in frequency and decreased in amplitude as participants got closer to the target. Only one tactor, either on the left or right side of the controller, would move at a time. The side that moved indicated the direction participants should rotate toward to most efficiently find the target.

1.2 Chapter Summaries

Brief summaries of the key information contained in the chapters of this thesis are provided below.

Chapter 2 provides background relevant to the design of a skin-stretch game controller. First, we present an overview of haptic game peripherals and their capabilities and limitations. This overview identifies a possible gap in haptic game feedback options that can be filled with skin stretch in a game controller. Next, we explore the results of prior skin-stretch studies in order to better understand skin-stretch feedback. Results from these studies provide information that explains many of the design decisions made in the design of the skin-stretch game controllers. Finally, we examine the characteristics of the skin-stretch mechanisms that are used in the game controllers. Explanations of the workings of these mechanisms and the methods used to accurately control their motions are critical pieces of information, which allow us to effectively integrate the mechanisms into the game controllers and improve upon their designs.

Chapter 3 presents the details of the front- and back-tactor controller designs. First we describe an existing skin-stretch game-controller prototype that is the predecessor of the front- and back-tactor controllers. The design goals for the skin-stretch game controllers are then specified with reasons for the design goals. Details of the mechanical, electrical, and software designs are then presented for the front- and back-tactor controllers. The mechanical design overviews provide photos of the controllers, exploded views of complex assemblies, labeled images of controller components, and detailed descriptions of the mechanical components. The electrical design overviews provide labeled photos of the printed circuit boards (PCBs) that control the game controllers, simplified electrical diagrams, and detailed descriptions of the electrical components and their purposes. The software design overview explains the flow of the software that operates on microcontrollers that control the game controllers. Finally, specifications and capabilities of the front- and back-tactor controllers are summarized.

Chapter 4 describes, in detail, the bi-linear interpolation methods that were implemented to enable accurate control of the back tactors for general motions. Before the bi-linear interpolation method could be implemented, data relating tactor position to servo positions had to be recorded. A calibration rig that uses linear encoders to measure the position of the tactor while controlling the servos is used to create a 5 by 5 lookup table of tactor positions and servo positions. A function that uses bi-linear interpolation of the 5 by 5 lookup table to convert desired tactor positions to servo positions is then described in detail. This function is implemented on the back-tactor controller along with code that moves the tactor by updating the tactor position/trajectory at a rate of 300 Hz.

Chapter 5 details an experiment that is designed to determine the relative effectiveness of the skin-stretch game controllers at rendering direction information. The experiment is an 8-direction identification task where participants identify the direction of skin-stretch cues. The front-tactor controller and the back-tactor controller with 5 different restraints comprise the six conditions tested. Directional skin-stretch cues were rendered to participants in one of 8 directions, and participants used the thumb joystick to identify the direction of the skin stretch. The results showed no significant differences in participant response accuracy between the skin-stretch game controllers, meaning that either could be used for gaming, as desired.

Chapter 6 details a second experiment that evaluates how effective the skin-stretch game controllers are at providing targeting information. Participants rotate right and/or left under rate control in a virtual environment and are tasked with finding a randomized target angle. The only feedback that participants receive is skin stretch. Three different feedback strategies are rendered to the front-and back-tactors and one feedback strategy is rendered to the palm tactors. The different locations and feedback strategies make a total of 7 conditions.

The first feedback strategy is referred to as sinusoidal feedback and was implemented in all three locations (front-, back-, and palm-tactors). Sinusoidal feedback moved the tactors in a sinusoidal motion that increased in frequency and decreased in amplitude as participants got closer to the target. Only one tactor, either on the left or right side of the controller, would move at time. The side that moved indicated the direction participants should rotate toward to most efficiently find the target.

The other two feedback strategies are called sustained and pulsed feedback. These feedback strategies move the tactor, relative to its center position, in the direction that the target is relative to the participant's current heading. The pulsed feedback pulses in the direction of the target, and the sustained feedback stays in an outbound position and follows the target around the edge of the tactor workspace. The results of the experiment show that sinusoidal feedback guided participants to the targets more quickly, efficiently, and consistently than the other two feedback strategies. The pulsed feedback guided participants to the target the slowest, likely due to the low bandwidth of the feedback. The sustained feedback guided participants to the targets slightly faster than pulsed feedback. Participants were able to rotate near the target quickly using the sustained feedback, but had difficulty finding the small target.

Chapter 7 concludes this thesis by first discussing the things learned from the design, verification, and testing of the skin-stretch game controllers. The contributions to the academic community are then described. Lastly, possible avenues of future work are discussed.

2 BACKGROUND

Prior work relevant to the design of a skin-stretch feedback game controller can be broken into three categories. First we will look at existing haptic game peripherals. By examining these commercial haptic devices, we can understand what capabilities current devices have, what these devices are commonly used for, and why a skin-stretch feedback game controller would be a useful addition. Next we will review a range of skin-stretch perception studies. The results of these studies provide critical information that has and will continue to guide the design and development of our game controller. Finally, we will review the compact skin-stretch actuators that have been integrated into our controllers. We will look at both the capabilities of these actuators and their limitations.

2.1 Haptic Game Peripherals

Haptic feedback has been used to enhance games since the 1970s, when vibrations were first used to convey impacts in arcade racing games. Since then, haptic feedback has found its way into personal computers and console games with the help of haptic game peripherals. These peripherals started with simple vibrotactile feedback, and can now provide force feedback in three dimensions.

Vibrotactile feedback is common in modern game controllers such as Microsoft's XBOX 360 controller and Sony's DualShock controllers. These controllers contain a pair of direct current (DC) motors that spin small off-center (eccentric) masses to generate

vibration. They use different size masses within the controller so that the two motors can generate different ranges of frequencies and amplitudes. Frequency and amplitude are coupled, however, since the motors vary both the frequency and amplitude by increasing the motor's output velocity.

Force feedback is another common form of haptic feedback found in modern game peripherals, such as Logitech's Driving Force GT force-feedback steering wheel, Microsoft's Sidewinder Force Feedback 2 force-feedback joystick, and the Novint Falcon 3d force-feedback robot. The above devices represent a progression of force feedback in games from 1, to 2, to 3 dimensions.

Force-feedback steering wheels apply torques to a steering wheel that users' primarily use to interact with driving games. The addition of torque to the steering wheel is commonly used to allow users to feel torques similar to driving a real car. Because the wheel is also used as a form of game input, the addition of torques can affect users' driving.

Force-feedback joysticks apply torques to the two axes of motion of joysticks. These joysticks are commonly used to simulate the torques felt through the control column of aircraft. Because the joystick is used as a two axis input, torques applied to the handle can reduce the user's ability to provide precise input as with force-feedback steering wheels. The Gravis Xterminator Force is a game controller that integrates the concept of a force-feedback joystick into its D-pad / thumb joystick. This has the same advantages and disadvantages of a force-feedback joystick, but in a smaller form factor.

The Novint Falcon applies forces in three dimensions to a user through a hand-held interface. The ability to render forces in three dimensions allows this device to render a

broad range of sensations. However, as with the force-feedback steering wheels and joysticks above, because the device is used as a 3-axis input, forces applied to the device affect users' positioning capability.

All of the force-feedback game peripherals can detrimentally affect users' abilities to provide precise input. However, they can also all provide a greater range of haptic interaction to users than vibrotactile feedback. This is where skin-stretch feedback has an opportunity to fill a potential need. This is because a skin-stretch feedback device can be used to provide 2-axis skin stretch in one or multiple locations to a user, without detrimentally affecting users' abilities to provide input.

2.2 Skin-Stretch Perception Studies

Skin stretch has been used to convey information in a wide range of devices, ranging from wearable and handheld devices to benchtop devices and even the steering wheel of a car. From these devices and the studies that use them, we learn some things that will help guide the design and development of skin stretch in a game controller.

2.2.1 Directional Fingertip Skin Stretch

Research on skin-stretch feedback applied to user's hands is an ongoing area of research. A study by Gleeson et al. [1] explores using skin stretch to convey direction information. Gleeson et al. looked at the effects of velocity and displacement on the communication accuracy of the four cardinal directions. They found that displacements as small as 0.05 mm could be perceived, and displacement as low as 0.2 mm resulted in communication accuracy rates above 95%. Cue velocities from 0.5 to 4 mm/s were

tested. The 2 and 4 mm/s cues achieved the highest communication accuracy, and the 0.5 and 1 mm/s cues performed significantly worse. A variation in performance based on stimulus direction was also observed. Participants appeared to perform better on the North and South cues than the East and West cues. A reanalysis of the data using d', a bias-free measure of detection accuracy, showed that after the effects of direction bias were removed, participants did respond more accurately to cues in the North and South directions.

Another study by Gleeson et al. [2] explored the effects of varying the orientation of the skin-stretch feedback relative to the participant. Users performed timed identification of directional cues with their hand in a range of orientations about a single axis. Response times with the wrist and finger rotated 90 degrees were about 0.2 s slower than response times with the arm straight forward. Arm rotation angles up to 36 degrees resulted in minimal increases to response times. This indicates that the orientation in which skin-stretch feedback is presented to a user can affect how it is perceived, and that moderate rotations from the ideal orientation can have relatively small impacts on users' perception of the feedback.

Montandon and Provancher [3] performed a study that rendered direction cues in one of 16 different directions. This more difficult task made it easier to observe rotational direction errors in users' responses. In one experiment, cues were only rendered to an extended right thumb (extended straight forward to be aligned with a "North" skin-stretch cue), resulting in an average accuracy of 29% for the 16-direction identification task. Users tended to respond in counter-clockwise direction to the direction cue as well for the right thumb experiment. A second experiment rendered cues in 16 directions to both

thumbs at the same time, resulting in an accuracy of 49%. No noticeable response rotation was observed in the dual-thumb case. This indicates that providing skin stretch to both hands in the same manner can greatly increase the communication accuracy of skin-stretch feedback.

Guinan et al. [4] performed an experiment that directly relates to providing skin-stretch feedback in a game controller. Guinan et al. embedded skin-stretch feedback into the thumb joystick inputs of a prototype game controller. The controller's handles can be mounted in various orientations. This device allowed them to test the effects of different handle orientations on users' ability to respond on the joysticks and their perceptions of skin-stretch cues. They found no significant difference in response times or response accuracies between the straight and angled handle configurations. They also found that there may be ergonomic advantages to the angled configuration.

Another experiment by Guinan et al. [5] used a prototype of the front-tactor controller detailed in this thesis to look at the effects of providing cues to both thumb joysticks with a range of delays between cues. In general, as the lag time between cues to the two thumbs increased, accuracy rates also increased. Lag times varied between 0 and 1500 ms and combined accuracy rates (meaning when both sequential cues were correctly identified and responded to correctly) varied between roughly 55% and 81%, respectively. This indicates that *different* cues provided to both thumbs with a small or no delay will reduce the accuracy of communication.

Guinan et al. [6] have also used skin-stretch feedback to communicate direction information in five degrees of freedom. This is achieved by placing two 2-axis tactors back-to-back in a way that the tactors can be grasped by a user's thumb and index finger.

Translational information is portrayed by moving both tactors in the same direction. Rotational information is portrayed by moving the tactors in opposing directions, to portray rotations out of the plane of tactor motion. A third axis of rotational information (in-plane to the tactor motion) is portrayed by simultaneously moving the tactors in a spiral, but this results in a less intuitive rotary cue than the out-of-plane rotation cues described above (some people intuitively just want to translate their hand in a circular pattern, though all participants learned to interpret this cue as an in-plane rotation with very high accuracy). Results from their first experiment showed that all five degrees of freedom provide > 98% communication rates. A second experiment showed that the device can be used to guide a user's hand motions under open-loop and closed-loop feedback. The important result from this study is that opposing skin-stretch cues rendered with back-to-back skin-stretch devices are intuitively interpreted as rotations or torques.

Another study by Guinan et al. [7] explored some of the uses of a skin-stretch game controller in gaming tasks. They used an earlier prototype of the front-tactor controller detailed in this thesis to provide directional information to aid users in two different gaming tasks. One task used skin stretch and visual feedback to indicate the direction to pass the football in a football simulator (user was the quarterback in an American-style football game). Users completed the task significantly faster when provided skin-stretch feedback than visual feedback. The second task required participants to move through a virtual environment using the thumb joysticks. The virtual environment contains obstacles and targets that the user must avoid and move to, respectively. Participants were provided visual feedback or visual plus one form of skin-stretch feedback. The form of skin-stretch feedback varied from pulsing cues similar to those provided in the above

directional skin-stretch experiments to cues that consisted of a sustained cue that does not return to center, but moves circumferentially around the edge of the tactor's 2 mm diameter workspace. The results showed that participants were significantly more "efficient" at finding the targets when provided with skin-stretch feedback. Participants did not exhibit significantly different performance levels under different skin-stretch feedback modes, but the sustained feedback had the highest numerical efficiency. These two experiments show that skin-stretch feedback can provide information that is additionally useful beyond the information provided by visual feedback.

Koslover et al. [8] developed a multimodal handheld device that provides direction information via visual, audio, skin-stretch, and vibrotactile feedbacks. Test participants use the direction information to navigate through a grid while verbally stating the direction of the cue. Results showed that participants were able to navigate accurately (> 96.9%) using any of the feedback modes. This experiment validates the use of skin-stretch feedback in a mobile device as a form of navigation.

2.2.2 Skin Stretch Used to Enhance Force Feedback

Skin stretch has also been used to enhance force feedback. This is interesting because it means that a new type of hybrid feedback device may be able to provide feedback that is superior. Gwilliam et al. [8] used skin-stretch feedback applied to the thumb to enhance direction information supplied by a force-feedback joystick. They provided users with skin stretch alone, force feedback alone, or a combination of both to convey direction cues to which participants responded with the joystick. Results showed that users responded with significantly higher accuracy when provided with force feedback and

significantly higher precision when provided with skin-stretch cues. When users were provided with both skin stretch and force feedback, precision was highest of the tested feedback conditions. This study indicates that skin-stretch feedback could be used to supplement low levels of force feedback in situations where large amounts of force feedback would conflict with a user's precise control.

Provancher and Sylvester [3] focused on how fingerpad skin stretch affects the perception of friction. They found that small amounts of skin stretch (0.25 to 0.75 mm) combined with force feedback via a PHANToM device significantly increased participants' perception of friction. This indicates that skin stretch can enhance (increase, or possibly decrease) the sensation of friction in an active or passive device.

Quek and Schorr [9] used a tactile display to augment kinesthetic-force feedback. Their device provided skin-stretch feedback to the thumb and index finger while participants grasped a device that also provided force feedback as they interact with a virtual springs. Their results showed that the skin-stretch feedback caused a significant increase in the participants' perception of the spring's stiffness.

2.2.3 Arm Skin Stretch

A study by Caswell et al. [10] looked at the communication accuracy rates for a variety of displacements for 4-directional skin-stretch cues applied to the palm, back of hand, distal dorsal forearm, and distal volar forearm. The results for the palm are of particular interest to a skin-stretch game controller. Communication accuracy rates for the palm exceed 95% with cues 0.5 mm in length. This result serves as an indicator of a minimum amount of skin-stretch displacement that is required to communicate basic

direction information via a users' palm, though lower displacements could possibly be used if trying to communicate less than 4-directions.

Bark et al. [11] use an arm-mounted device to provide rotational skin stretch to a user's arm. In one experiment, participants used the rotational skin-stretch feedback in a closed-loop manner to successfully orient virtual objects to within several degrees. In a second experiment, participants identified varying amplitudes of static rotational skin stretch. This second task was more difficult, resulting in lower response accuracy. Participants also occasionally confused clockwise and counter-clockwise rotations. This study shows that users may not be very good at sensing the amplitude (and occasionally direction) of static skin-stretch displacements. Also, it shows that if participants are allowed to move, skin stretch can be used to guide physical motions.

Related work by Wheeler et al. [12] uses the same type of arm-mounted device to provide rotational skin stretch. The skin-stretch cues are used to provide proprioceptive feedback from a virtual prosthetic arm (computer simulation of a prosthetic arm). The participants control the prosthetic arm using myoelectric sensors attached to their biceps and triceps muscles. Their results showed that the skin-stretch feedback was able to reduce blind targeting errors relative to no feedback. They also showed that participants experienced lower visual demand while following a 30 degree moving target range. This indicates that rotational skin-stretch feedback can be useful as a form of proprioceptive feedback for prosthetic devices, or possibly virtual limbs (a gaming application).

Minamizawa et al. [13] propose a wearable fingertip haptic display that conveys weight to a user's fingers. The device consists of two rollers that laterally move a flexible sheet across the user's finger to generate distributed skin stretch. The device is also

capable of generating forces normal to the finger surface when the flexible sheet is pulled taught by both motors. They find that this device is able to create a reliable weight sensation.

Biggs and Srinivasan [14] investigated the relative effectiveness of tangential and normal skin stretch as a tactile sensation. They used a 1 mm probe to generate slow skin displacements of varying magnitudes. Participants varied the magnitude to match the intensity of sensation generated by a reference normal skin displacement. At the fingerpad and forearm, participants ended up varying the skin-stretch displacements to 0.3 to 0.6 times the magnitudes of the normal skin displacement. This indicates that users are significantly more sensitive to tangential skin displacement than normal skin displacement.

Medeiros-Ward et al. [15] developed a steering wheel that provides skin-stretch cues that convey direction information. Participants were tasked with driving and making lane changes in response to auditory and tactile instructions. They found that participants were able to effectively interpret and follow the skin-stretch commands while driving. They also found that when participants were additionally tasked with conversing on a cell phone, that they more accurately responded to skin-stretch cues than verbal instructions. This indicates that skin stretch can be used to bypass an audio information bottleneck.

Vitello et al. [16] explored the effects of active exploration on tactile sensitivity. They used an integrated tactile/kinesthetic display to generate skin-stretch feedback while moving the participant's arm. Participants either moved their arm on their own or let the display move their arm for them. While their arm moved, participants had to identify the direction of the skin-stretch feedback. They found that when participants were actively

moving their arm, their tactile sensitivity was significantly decreased. This indicates that motions made while playing games may decrease the effectiveness of skin-stretch feedback.

2.3 Compact Skin-Stretch Actuators

The skin-stretch mechanisms used in the front- and back-tactor controllers are closely related to other skin-stretch devices created at the Haptics and Embedded Mechatronics Laboratory (HEML) at the University of Utah. Using the actuator designs developed for these other skin-stretch devices will allow for quicker implementation of skin-stretch feedback in game controllers. However, before using the skin-stretch actuator designs from the other devices, we should examine their effectiveness and weakness so that improvements may be made as appropriate.

2.3.1 Skin-Stretch Feedback Tactor and Aperture Research

Work by Gleeson et al. [4] explored the effects of tactor size and texture, and an aperture-based restraint on communication accuracy. They tested three sizes of tactors with rough and smooth textures. Tactor size did not have a significant effect, but the rough texture significantly improved communication accuracy over the smooth texture. Three sizes of aperture-based restraints (10, 12, and 16 mm dia.) were also evaluated on the index finger and thumb. The larger apertures (12 mm for index finger and 16 mm for thumb) resulted in the highest communication accuracies for both the index finger and thumb. They hypothesized that the smaller apertures reduced performance because the apertures masked away mechanoreceptors on the participants' skin. The 12 mm inner

diameter aperture has been used on all skin-stretch game controller designs as a compromise between performance and packaging space.

2.3.2 Servo-Driven Flexure Design

The servo-driven flexure used in the front-tactor controller is based on the design by Gleeson et al. [3]. Gleeson et al. developed their flexure-driven design to be portable and effective at communicating direction via skin stretch. They found that while the design decoupled the motions of the two servos, significant compliance and backlash inherent in the device reduced the amount of skin displacement produced at the tactor. They minimized the error in position by commanding positions beyond the position they desired. This feed-forward correction was found to result in average lengths of rendered stimuli and RMS error of 1.10 ± 0.10 mm and 0.99 ± 0.10 mm for the X (lateral to finger) and Y (along length of finger) axes. The primary source of the variation in stimulus length is variation in participant skin mechanical properties. Use of this actuator design requires attention to its compliance.

2.3.3 Flat Servo-Driven Sliding-Plate Design

Montandon and Provancher [3] developed a flatter servo, wire, and sliding-plate skin-stretch mechanism. Their skin-stretch mechanism is designed to effectively communicate direction via skin stretch to thumbs, while maintaining a small and flat form factor. This design has much less compliance than the servo-driven flexure design. Position measurements of uncorrected skin-stretch cues show evidence of nonlinear kinematics in the device. The error in cue motions was effectively eliminated by commanding a tactor

trajectory along a series of waypoints for each cue direction. This indicates that the same or similar correction will need to be made in order to accurately position the tactor of this device design.

This flat skin-stretch mechanism is also used in an arm-mounted device by Caswell et al. [10]. Caswell et al. developed a portable device that provides direction information via skin stretch to the forearm. The device is capable of positioning a 7 mm diameter ThinkPad TrackPoint tactor within a 4 mm diameter workspace with up to 40 N of force. This device serves as additional evidence that this skin-stretch mechanism can be effectively integrated into other devices.

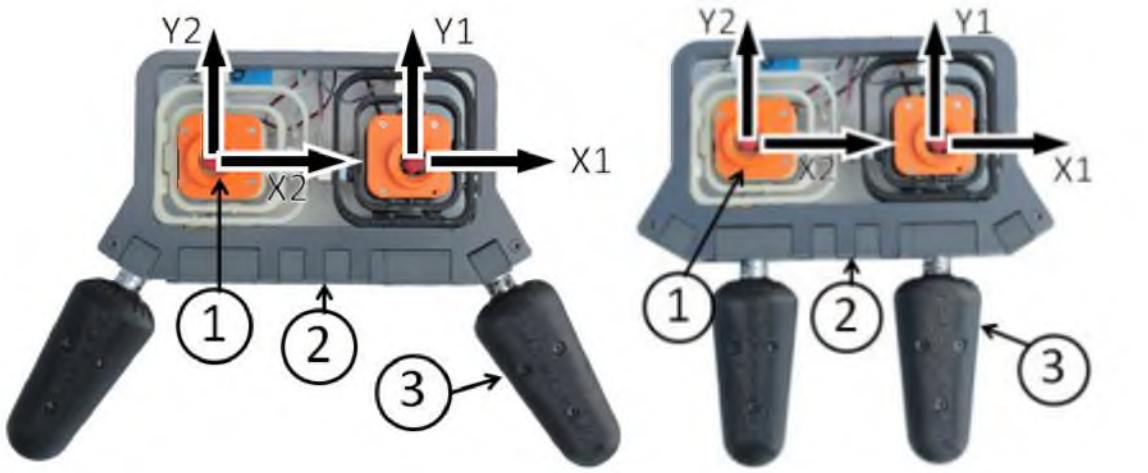
3 DESIGN

The designs of the skin-stretch game controllers detailed below were heavily influenced by the game-controller prototype used by Guinan et al. [4] and modern game controllers (XBOX 360 and PlayStation 3). They are the result of attempts to determine how skin-stretch feedback can most effectively be integrated into modern game controllers.

In order to fully detail the design of the skin-stretch game controllers, we will first look at how existing devices influenced the design goals of the controllers. The design of the skin-stretch game controllers will then be discussed in three areas: mechanical, electrical, and software.

3.1 Prior Device

The bimanual feedback and input game-controller prototype used by Guinan et al. [4] is the direct predecessor to the devices detailed below. As shown in Figure 3.1, it has two 2-axis skin-stretch feedback mechanisms, each of which is mounted inside of a 2-axis gimbal. The angle of each axis of the gimbal is measured by a potentiometer, and a torsional spring on each rotational axis provides torque to center the gimbal. The two gimbals are mounted in a frame that has handles that can move to change the angle at which a user's thumbs and hands interact with the device.



(a) (b)

Figure 3.1: Prior game-controller prototype in two configurations: with handles mounted in an angled orientation (a) and with handles mounted directly below the thumb joysticks in a straight orientation (b). The numbers refer to: (1) skin-stretch mechanism in a 2-axis gimbal (2) controller frame (3) adjustable handles.

Testing with this device found that the two different hand orientations had no significant effect on users' response times and response accuracy to directional skin-stretch cues. Participant comments and slightly faster (but not significantly so) response time results with angled handles indicate that there may be ergonomic advantages to the angled configuration. For this reason, our subsequent skin-stretch game controllers were designed with angled handles.

3.2 Design Goals for New Game-Controller Device

The primary goals of the new design were to enclose the two 2-axis gimbals inside of an ergonomic shell and to integrate skin-stretch feedback in an effective way. We chose to emulate the input and feedback capabilities of modern game controllers (XBOX and PlayStation) due to their familiarity and proven effectiveness at interacting with virtual

environments. These features include: dual 2-axis thumb joystick inputs with integrated tactile switches, multiple buttons, triggers, vibrotactile feedback, and an ergonomic enclosure. The inclusion of vibrotactile feedback is important because it allows us to evaluate the effects of combining vibrotactile feedback with skin-stretch feedback (and because it is used in modern game controllers).

The ergonomic shape of the device was chosen in an attempt to replicate the familiar and comfortable hand positions and grips found in popular game controllers. Key features of the ergonomics include: angled handles, thumb joysticks reachable by angled thumbs, triggers reachable by index or middle fingers on the top of the device, buttons on the face of the device reachable by thumbs, buttons above the triggers (referred to as bumpers) reachable by index fingers, and a rounded form factor for comfort.

Finally, all electronic components were to be embedded inside of the shell of the device in order to make the device more portable, durable, and easy to use.

3.3 Mechanical Design Overview of Game Controllers

Two different devices were designed to provide skin-stretch feedback in different ways. The first device uses the 2-axis skin-stretch mechanism in a 2-axis gimbal to provide skin stretch at the thumb joysticks. This device is similar to the device used by Guinan et al. [4] and uses a servo-driven flexure stage similar to Gleeson et al. [3]. The second device provides 2-axis skin stretch to the middle fingers on the back side of the device instead of at the thumb joysticks, and also provides 1-axis skin stretch to the palms at the handles of the device. The 2-axis skin-stretch mechanism used in the back-tactor design is based on the design by Montandon and Provancher [3].

Figure 3.2 shows the names by which the different sides of the controllers will be referred to. The face of the device with the thumb joystick will be referred to as the front side, and the opposite side as the back side. The side of the device with the triggers will be referred to as the top side, and the opposite side as the bottom.

3.3.1 Front-Tactor Controller

The front-tactor controller is designed to mimic the capabilities of modern game controllers, while additionally providing planar skin stretch to users' thumbs through the thumb joysticks. The design detailed below is the most recent iteration of a series of prototypes.

Figure 3.3 shows the locations of features on the front-tactor controller. The thumb joysticks are located so that users' thumbs are angled toward the centerline of the controller at approximately 45 degrees when users hold the device. Buttons on the face of the controller are located closer to the sides of the device and are reachable by users' thumbs. Triggers and bumpers are located on the top of the controller so that they can be easily reached by users' index fingers. The triggers are modified from an off-the-shelf trigger assembly (SparkFun Electronics part number COM-10314) to press a tactile switch at the end of their travel instead of using a potentiometer. This was done to reduce the size of the trigger assembly so that it would fit into the controller shell.

Power and communication lines are routed through cables that exit the device through the strain relief. A small, female single in-line pin (SIP) connector that is flush with the controller shell provides an easy way to reprogram the microcontroller inside the device.

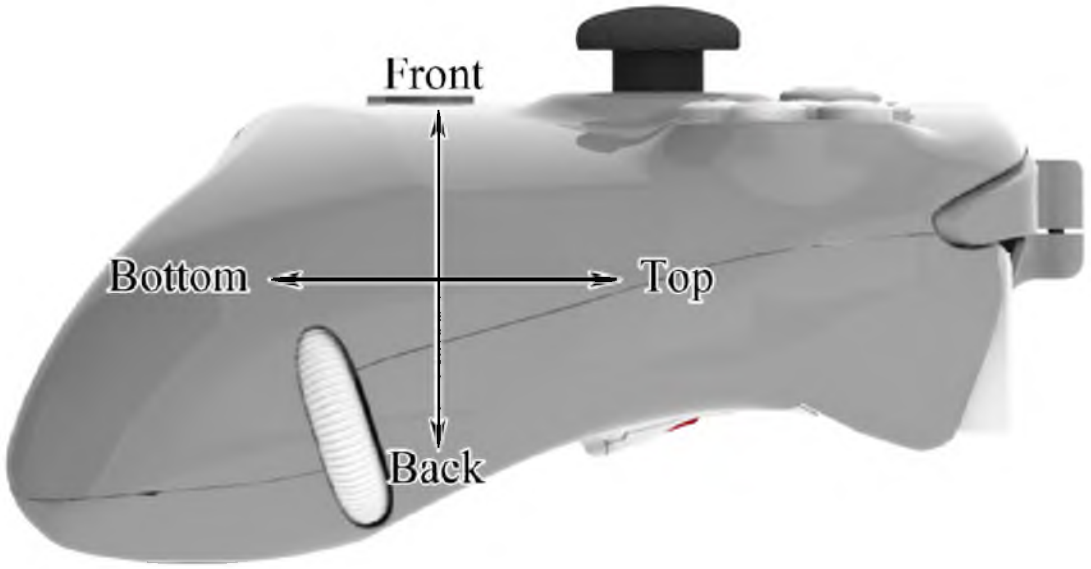


Figure 3.2: Side view of front-tactor controller with direction labels

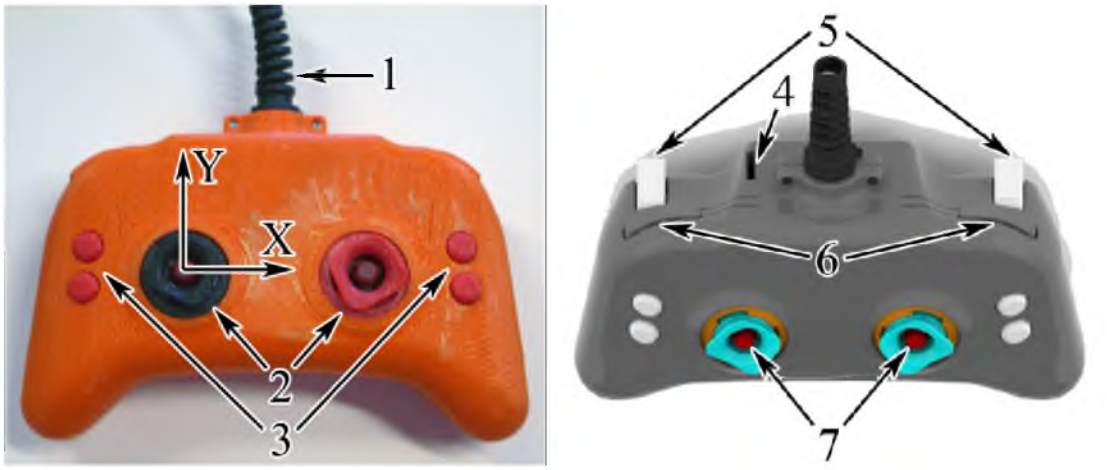


Figure 3.3: Photo and render of front-tactor controller with tactor axes labeled. (1) Cable strain relief (2) Thumb joysticks (3) Buttons (4) Programming port (5) Triggers (6) Bumpers (7) Tactors

Figure 3.4 shows the controller with the front half of the shell removed. Vibrotactors are located in each of the handles of the device. The offset mass on the left vibrotactor is smaller than the mass on the right vibrotactor. This is done so that each vibrotactor generates a different vibration amplitude and frequency when powered. Vibrotactors in modern game controllers also have two different masses for the same reason.

The PCB is located in the top of the device. Cables are routed between the PCB and the electronic components in the controller. The connectors on the top side of the PCB connect to cables that exit the top of the controller. The connectors on the bottom side of the PCB connect to cables that run to the tactile switches, servos, potentiometers, and vibrotactors.

Figure 3.5 shows an exploded view of the components critical to the rotation of the joystick. Further details are shown in Figure 3.6 and Figure 3.7. Each axis of the gimbal is supported on one end by a potentiometer (Digi-Key Corporation part number P3R7103-ND), and by a 2 mm steel shaft on the other end. The steel shafts pass through bronze bushings that are press fit into the gimbal. One end of the steel shaft is affixed to a bronze bushing to keep it in place.

The outer ring of the gimbal is grounded to the shell of the device and allows the thumb joystick to move in the Y direction as labeled in Figure 3.3. The center of the gimbal is grounded to the outer ring of the gimbal and allows the thumb joystick to move in the X direction. The range of motion of the thumb joystick is limited by the hole in the shell through which it extends. This limits the joystick to a ±10.5 degree maximum range of motion about each axis.

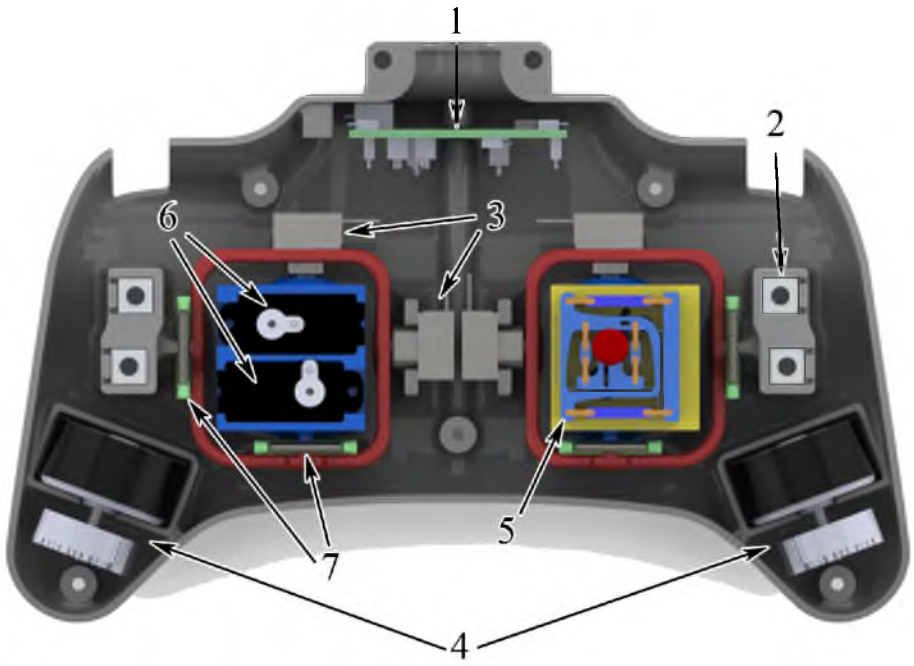


Figure 3.4: Front-tactor controller with front removed. (1) PCB (2) Tactile switch (3) Joystick potentiometers (4) Vibrotactor motors (5) Flexure stage (6) X- and Y-servos (7) Spring return mechanisms

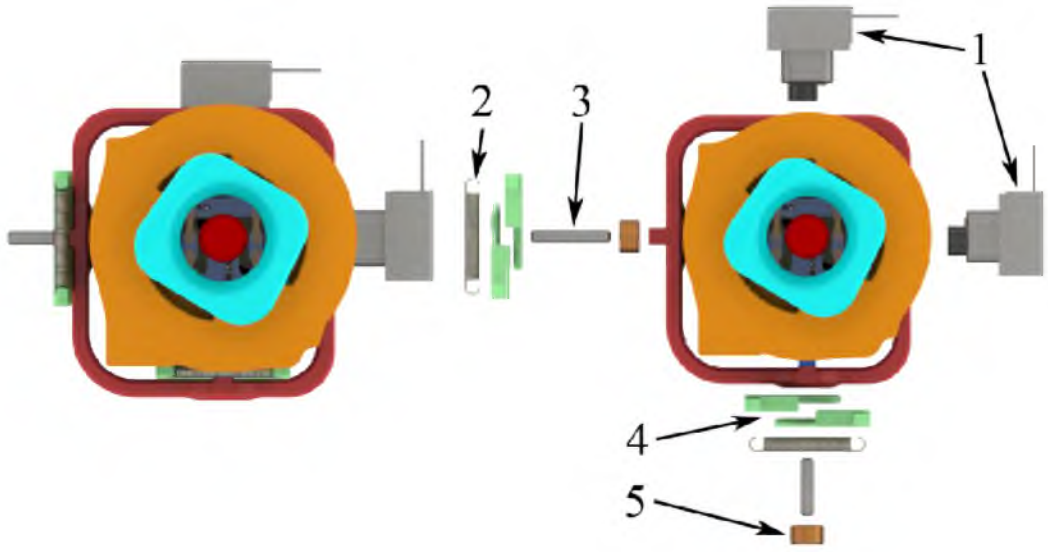


Figure 3.5: Front view of gimbal assembled and exploded. (1) Potentiometers (2) Tensile spring (3) 2 mm steel shaft (4) Spring return arms

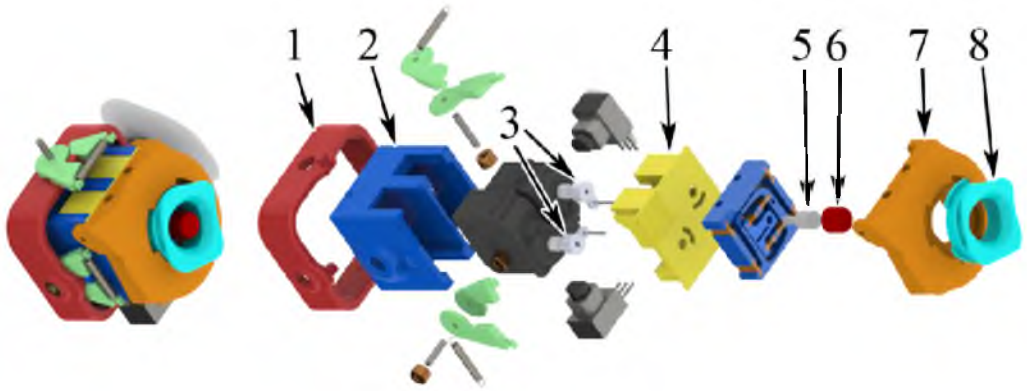


Figure 3.6: Diagonal view with gimbal assembled and exploded. (1) Gimbal ring (2) Inner gimbal block base (3) Servo Arms (4) Gimbal block insert (5) Machined brass threaded rod (6) ThinkPad TrackPoint cap (7) Gimbal block top dome (8) Adjustable thumb aperture

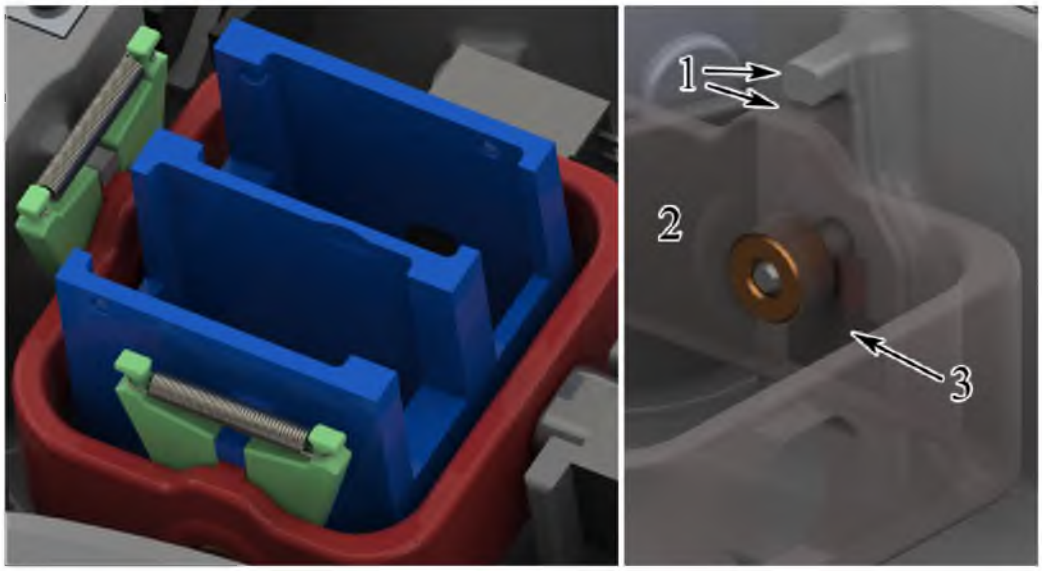


Figure 3.7: Close up views of spring return mechanism and tactile switch under gimbal. (1) Extrusions that the spring return arms align (2) Gimbal ring (3) Tactile switch

Figure 3.6 shows the components of the mechanism that moves the tactor. The inner gimbal block is designed so that the components assemble with light press fits. A few screws prevent the pressed assembly from disassembling. This design reduces the overall size of the inner gimbal block from Guinan et al. [4] from 33.4 mm x 33.7 mm x 37.4 mm to 27.1 mm x 27.6 mm x 35.7 mm.

The motion of the tactor is actuated by two Futaba S3154 servos. The servos apply forces to a flexible, 2-axis coupler (referred to as a flexure stage). The force is applied via a pin attached to the output arm of the servo. The radius of the servo arm is 4 mm. The rotation of the servo arm causes the pin to apply force to the flexure stage. The flexure stage is designed so that each servo moves the output of the flexure stage along a single axis. A TrackPoint ThinkPad pointing stick cap is attached to the output of the flexure stage via a machined piece of 10-32 threaded brass rod.

When force is applied to the side of the tactor by a user's thumb, compliance in the components between the servo and the tactor allows the tactor to move slightly from its intended position. This compliance results in some error between the commanded tactor position and the actual tactor position when a user's thumb is on the device. This is compensated for in software by commanding a position beyond the desired position.

Figure 3.7 shows a closer view of the spring return mechanism for the thumb joysticks. A spring is used to provide torque that centers the joystick. The spring is attached to two arms that rotate about the 2 mm steel shaft. The length of the spring was chosen so that it is always applying some amount of force to the two arms. The two arms act as a 'clamp' that applies an aligning force to the surrounding stages of the gimbal (controller shell and gimbal ring or gimbal ring and inner gimbal block). This spring

return mechanism was chosen over a simple torsional spring so that the joysticks have a 'hard' center position.

3.3.2 Back-Tactor Controller

The back-tactor controller is a completely separate design from the front-tactor controller. The back-tactor controller was designed to be easier to assemble, more ergonomically shaped, provide additional feedback, and use less compliant actuator mechanisms.

Figure 3.8 shows the locations of features on the front and top of the back-side tactor controller. The location of the thumb joysticks, triggers, and tactile bumpers are similar to the front-tactor controller design. The most noticeable difference is that there are no longer tactors integrated into the thumb joysticks. The thumb joysticks used in this device are commercial 2-axis thumb joysticks with an integrated tactile switch (SparkFun.com Electronics part number COM-09032).

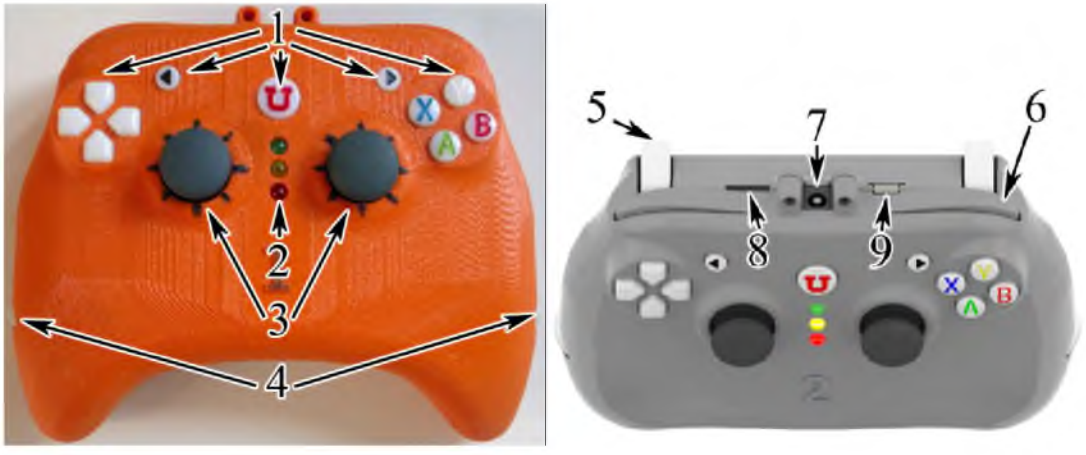


Figure 3.8: Photo and render of front-tactor controller. (1) Buttons (2) Light emitting diodes (LED) (3) Thumb joysticks (4) Palm tactors (5) Trigger (6) Bumper (7) Barrel power connector (8) Programming port (9) Universal Serial Bus (USB) mini connector

Future revisions of this controller will use a different 2-axis thumb joystick (Digi-Key Corporation part number 252A103B60NB-ND). This change was made because the potentiometers in the joysticks from SparkFun Electronics have a deadband in the center of their range where the signal does not change.

Also, there are several new features. The single power/communication cable has been replaced with separate barrel connector and USB mini connector for power and communication, respectively. This change was made to allow the cables to be easily detached from the device so that it can become wireless. Several new buttons have been added to the face of the device. This was done to make the input capabilities of this device equivalent to those of an XBOX 360 game controller. Three LEDs were also added to the face of the device. These LEDs allow the device to provide visual feedback as a form of communication, device status, troubleshooting, etc.

Figure 3.9 shows the inside of the controller with the front half removed from the front view. The biggest change from the front-tactor controller design is the PCB. All electrical components (except the actuators) are now directly soldered to the PCB. This significantly reduces the number of wires and connectors in the controller, and also makes assembly and maintenance/repairs much easier and quicker.

The palm tactors are directly attached and driven by Futaba S3156 servos. The palm tactors are located in the handle of the device so that users' palms are in contact with the palm tactors whenever they are holding the device. The vibrotactors were moved out of the handles to make room for the palm tactors, and are located next to the battery compartment. The battery compartment is included in the design so that the controller may be used wirelessly if so desired.

The back tactors are each driven by a pair of Futaba S3154 servos via 0.64 mm diameter spring steel wires. The spring steel wires are clamped between a sliding plate and the tactor post. An eyelet (McMaster-Carr part number 7113K239) attaches a custom servo arm to the end of the spring steel wire. This eyelet attaches to a custom servo arm with a 2 mm steel shaft oriented orthogonally to the spring steel wire. This creates a rotational joint with very low backlash between the spring steel wire and the servo. The servo arm has a radius of 5 mm, versus the 4 mm servo arm in the front-tactor design. This increases the maximum velocity of the tactor output in exchange for lower force. This was done because the maximum force output of the front-tactor controller was significantly higher than was necessary, and there are advantages to a higher maximum tactor velocity, such as sharper impulse motions.

Figure 3.10 shows the back of the controller. The area surrounding the tactor is designed so that it can be replaced with plastic part inserts of varying materials and shapes. This was done so that we could compare the effects of different finger restraints on users' perception of the skin stretch. The tactor is located so that the user's middle finger can rest on it while they hold the controller.

Figure 3.11 shows the top of the controller viewed from the back. The top half of the shell presses down on the servos and vibrotactors when assembled. This, in addition to the friction of the press fits between the actuators and the bottom half of the shell, holds the actuators in place when the controller is assembled. There is also space underneath the PCB for a Bluetooth module. This Bluetooth module can optionally be soldered to the PCB to enable communication over Bluetooth as an alternative to wired USB.

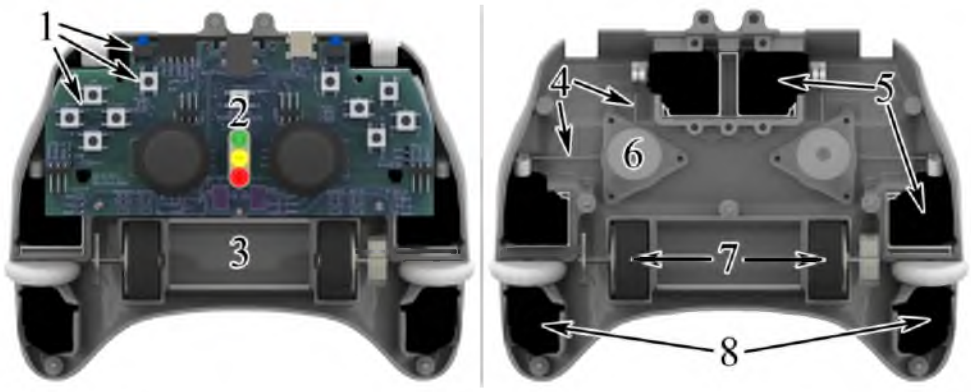


Figure 3.9: Back-tactor controller with front shell removed. (1) Tactile switches (2) PCB (3) Battery compartment (4) Push/pull wires (5) Servos for back-tactor (6) Sliding plate (7) Vibrotactor motors (8) Servos for palm tactors

Figure 3.10: Back view of back-tactor controller with tactor axes labeled. (1) Removable finger restraints (2) Battery compartment cover (3) Tactors

Figure 3.11: Back view of back-tactor controller with back shell removed. (1) Analog triggers (2) Optional Bluetooth module

3.3.3 Previous Design Iterations

The front-tactor controller was the first controller to be developed. Several non-functional and function iterations of this controller were designed and fabricated in order to improve the design. Early nonfunctional iterations tested both the ergonomics (shape, grip, location of key features) and suitability of fused deposition manufacturing (FDM) as the manufacturing method for the shell and many subcomponents. These iterations verified that the design concept was mechanically viable.

Later functional iterations varied the initial designs to make room for a PCB to act as an interface between the actuators and inputs of the device and a personal computer (PC). These designs also added a combined power and RS-232 communication cable that exits the device through a strain relief coupling. One variation attempted to reduce the size and weight of the device by using much smaller Nuke 3 servos to actuate the 2-axis skin-stretch flexures. While the design was smaller and lighter, the Nuke 3 servos proved to have a high failure rate, were louder, and were significantly slower than the Futaba S3154 servos used in the above designs.

3.4 Electrical Design Overview of Game Controllers

The front- and back-tactor controllers' PCBs are separate designs. The front-tactor controller's PCB was constrained to have a much smaller size, and due to its placement, must connect to the tactile switches and potentiometers via wires instead of directly soldering them to the PCB as was done with the back-tactor PCB.

3.4.1 Front-Tactor Controller PCB

The front-tactor controller's PCB was designed after the mechanical design of the device. Due to this, the size and location of connectors on the PCB were limited by the initial design of the controller's shell. Although a later redesign of the front-tactor controller increased the allowable size of the PCB, the original PCB design has not been resized. The schematic for this PCB is very similar the PCB used to control the front-tactor controller's predecessor. The use of the same microcontroller and very similar schematic layout allowed for the reuse of previous microcontroller code.

Figure 3.12 shows the components that interact with the dsPIC30F4011 microcontroller that is used in the front tactor game controller. Pull-up resistors cause the un-pressed state of the tactile switches to be 5 V. The output signals of the potentiometers for the 2-axis gimbals are capable of varying between 0 and 5 V, but do not exhibit this full range in practice because only 21 degrees of their 300 degree range is used.

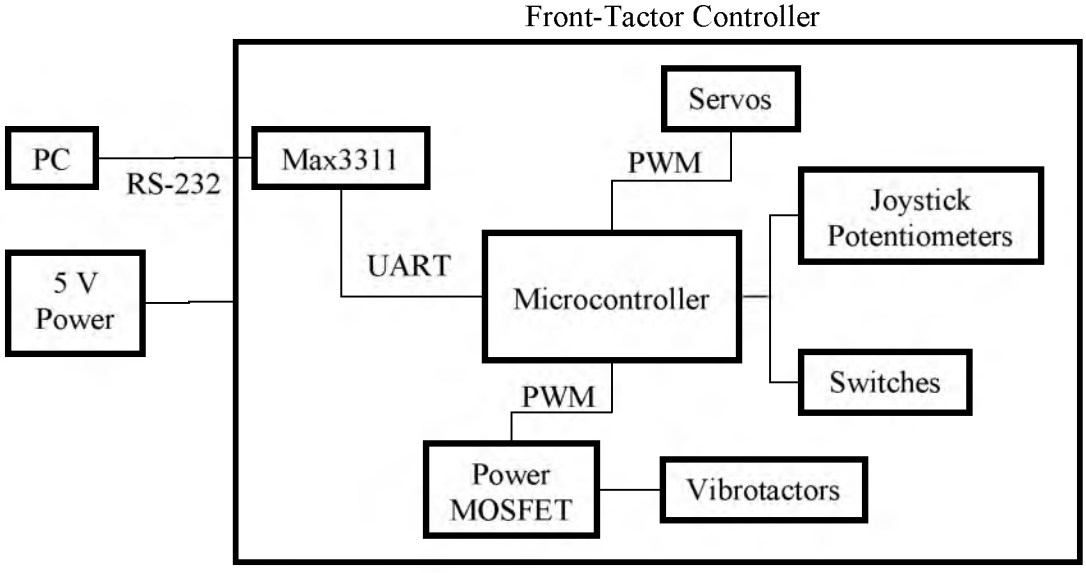


Figure 3.12: Simplified schematic of front-tactor PCB.

A dual channel TrenchFET power MOSFET allows the microcontroller to control the vibrotactors with a pulse-width modulated (PWM) signal. A MAX3311 integrated circuit (IC) allows the microcontroller to communicate with a PC by converting the microcontroller's 0 to 5 V Universal Asynchronous Receiver/Transmitter (UART) output to RS-232 voltage levels.

Figure 3.13 shows the layout of the front and back sides of the front-tactor controller PCB. The connectors for the servos, tactile switches, and vibrotactors are located on the front side of the PCB, and the connectors for power, RS-232 communication, and reprogramming of the microcontroller are located on the back of the PCB. This minimizes the lengths of wires required. A single mounting hole at the bottom of the PCB and the bottom edge of the PCB fully constrains the PCB to the controller.

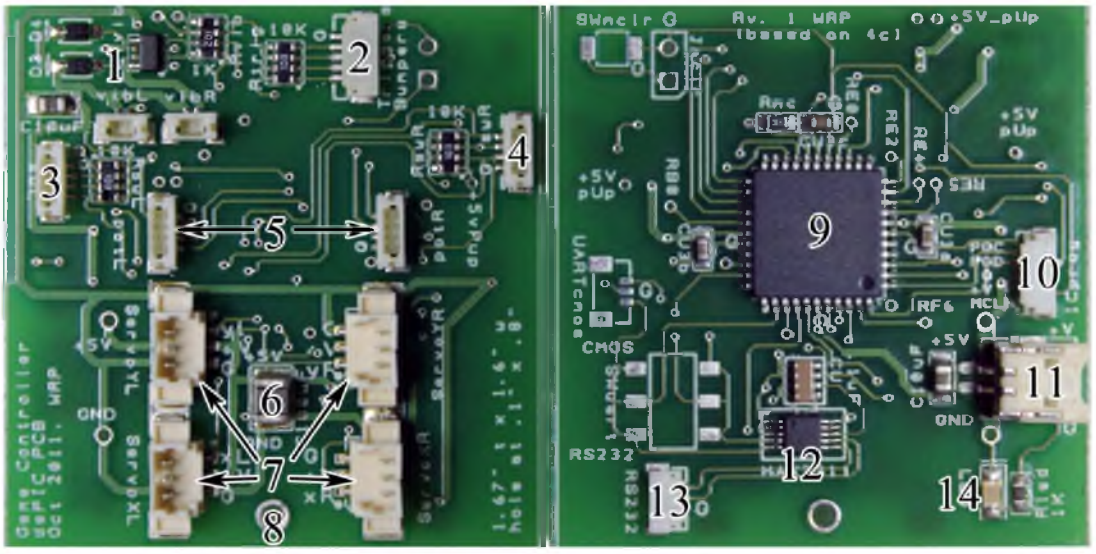


Figure 3.13: Top and bottoms of front-tactor controller PCB. (1) Vibrotactor power/control components (2) Trigger and bumper header (3) Left side switch header (4) Right side switch header (5) Potentiometer headers (6) 100 uF filter capacitor (7) Servo headers (8) Mounting hole (9) Microcontroller (10) Programming header (11) Power header (12) Max3311 (13) RS-232 communication header (14) Power LED

3.4.2 Back-Tactor Controller PCB

The back-tactor PCB is controlled by a dsPIC33EP256MU806 microcontroller. The dsPIC30F4011 was not chosen for the back-tactor controller PCB for a number of reasons. The primary reason is that the dsPIC30F4011 microcontroller used in the front-tactor PCB does not have enough pins to interact with the additional tactile switches, LEDs, and servos. Initially, the back-tactor PCB was designed to use the dsPIC30F6015, which contains more than enough pins and peripherals to control the additional features. The dsPIC33EP256MU806 displaced the dsPIC30F6015 due to its much more flexible pin assignments, and its built-in USB communication module. USB communication is preferred over serial communication due to its increased bandwidth capabilities, availability in modern PCs (RS-232 serial ports are no longer commonly found on PCs), and its more robust communication protocols.

The PCB acts not only as an intermediary between the PC and the actuators, but is also the physical mounting plane for the majority of the electrical and mechanical components in the controller. By making the PCB central to the controller, the PCB can be much larger in size, while reducing the number of wires inside the controller. The location of tactile switches, thumb joysticks, LEDs, triggers, and power and USB connectors, however, must line up exactly with their corresponding features on the controller's shell. Because of this, the back-tactor controller PCB was designed at the same time as the back-tactor controller.

Figure 3.14 shows all of the electrical components that interact with the microcontroller. Notable differences from the front-tactor controller are the additional 3 general purpose input/output (GPIO) controlled LEDs, a USB connector, 5 tactile

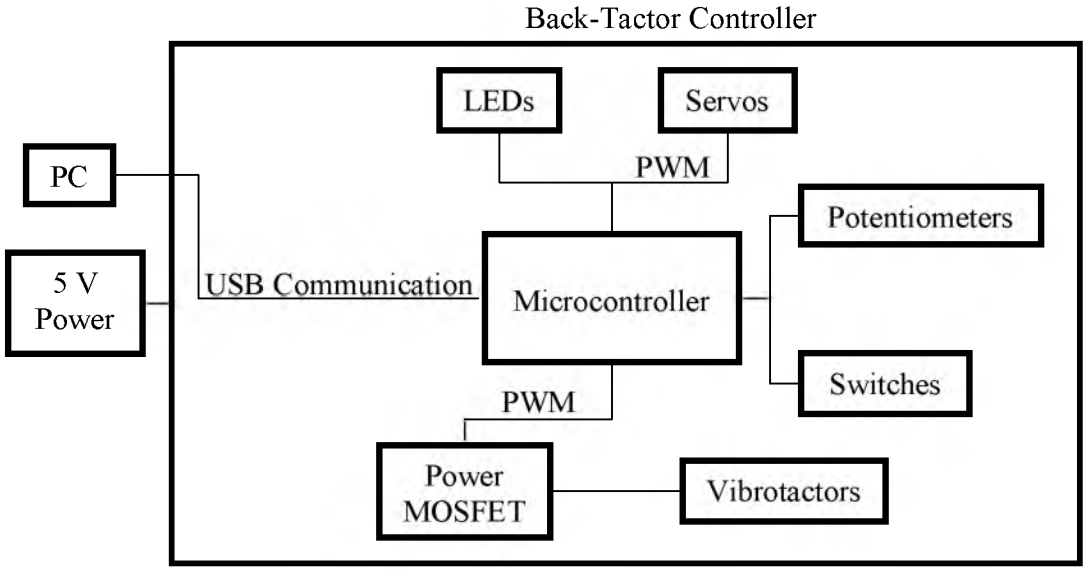


Figure 3.14: Simplified schematic of the back-tactor controller PCB

switches, an oscillator, a 3.3 V voltage regulator, 2 analog triggers, 2 palm tactor connectors, 2 spare servo connections, and 2 spare motor outputs (same as is used by the vibrotactors). The oscillator is required in order to create a more exact clock signal for the microcontroller so that USB communication is reliable. The 3.3 V regulator is required because the dsPIC33EP256MU806 operates on 3.3 V logic, with a maximum input voltage of 3.6 volts. A benefit of the regulator is that voltage drops at the power input due to high current draw of the servos should have less of an effect on the microcontroller.

Figure 3.15 shows the layout of the front and back sides of the back-tactor controller PCB. The PCB has 5 mounting points to the controller shell. Traces that transfer current to the servos are much larger than the traces for logic due to the potential of high current draw from the servos. Surface mount 100 uF capacitors are located between power and ground immediately next to each servo header to act as temporary current sources in the event of large load spikes from the servos. Prior to the addition of these capacitors, the servos could sometimes draw large enough amounts of current to drop the voltage at the regulators inputs, causing a voltage drop in the microcontroller's power, resulting in the microcontroller resetting.

3.5 Software Design Overview

The flow of the software on both the front- and back-tactor controllers is similar enough that they will be described together. Both controllers use a microcontroller to communicate with a PC, sense switch and potentiometer states, and control servos, vibrotactors, and LEDs. Figure 3.16 illustrates the flow of the software on the microcontroller.

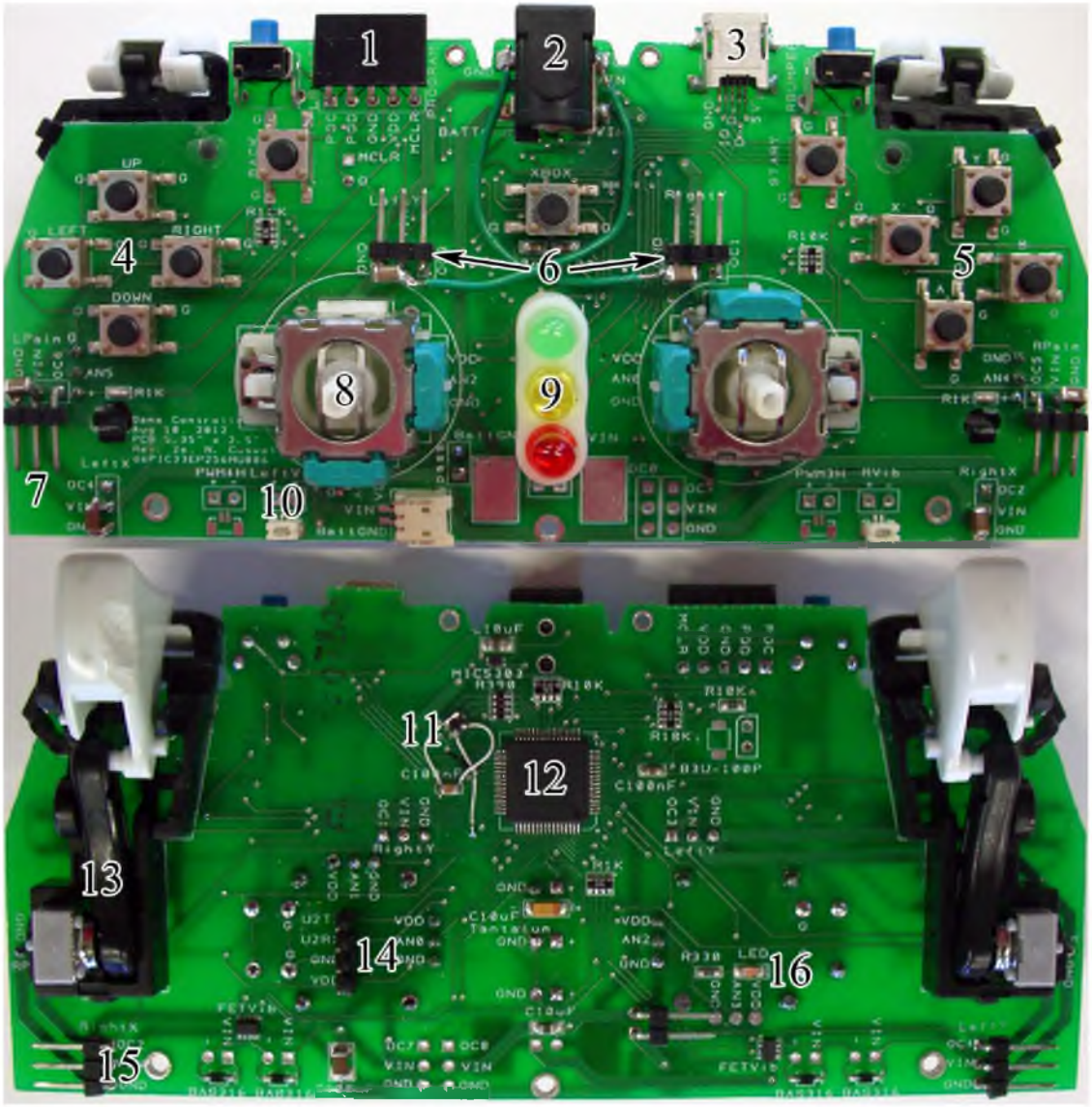


Figure 3.15: Front and back sides of back-tactor controller. (1) Programming header (2) Power connector (3) USB mini header (4) Tactile switches (5) More tactile switches (6) Back-tactor Y-axis servo connectors (7) Palm-servo connector (8) Thumb joystick (9) LEDs (10) Vibrotactor header (11) Oscillator (12) Microcontroller (13) Analog Trigger (14) Bluetooth connector (15) Back-tactor X-axis servo connectors (16) Power LED

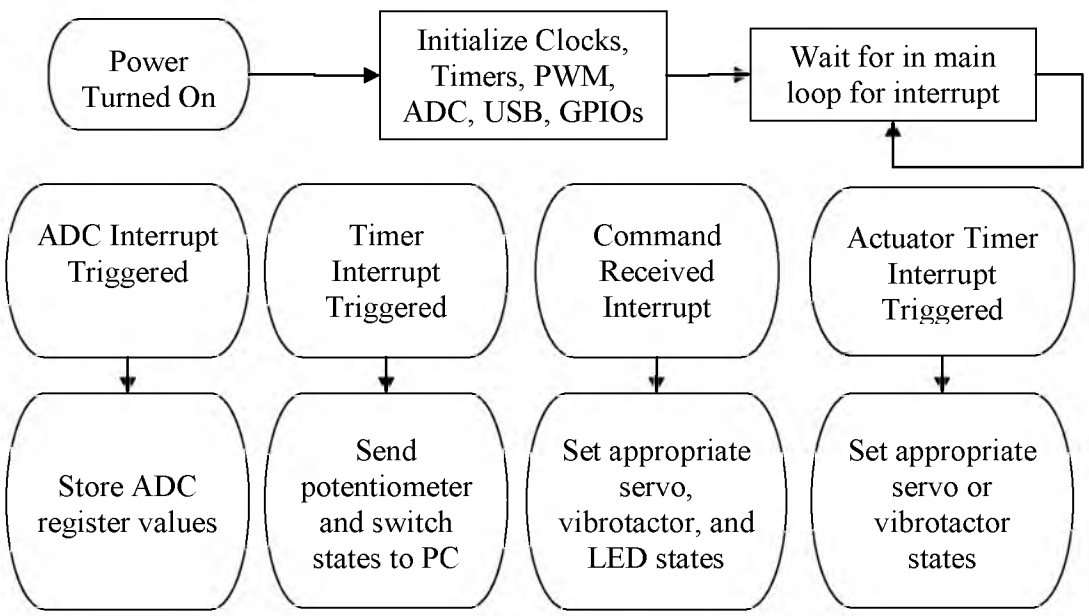


Figure 3.16: Software flow diagram

The software on the microcontrollers is event driven. Most of the time the microcontroller is waiting for one of its many interrupts to trigger a subroutine. The software sets up these timers during startup. It is important to note that the front-tactor controller sends state messages to the PC every 8 ms, but the back-tactor controller waits for the PC to send a command before replying with a state message. This difference is due to the fact that the back-tactor controller communicates using USB human interface device (HID) protocol, so it was simpler to allow the PC to allow flow control for the PC-microcontroller data exchange. The flow of the microcontroller software is as follows:

- On start up: Initialize peripherals (setup output compare and command servos to their center positions, setup PWM and set vibrotactors off, set LEDs off, setup communication peripherals (UART or USB), setup analog-to-digital converter (ADC)

to continuously sample, and setup timer interrupts to record potentiometer values and send state messages to the PC)

- Periodically triggered by interrupt: Add current ADC values of potentiometers to moving average filter
- Periodically triggered by interrupt or by PC: Communicate state of tactile switches and potentiometers to PC
- In the event of a PC command:
    a. Parse the command to determine what type of command was sent and associate order-dependent byte fields to appropriate controller functionality
    b. Set the appropriate PWM, output compare, and GPIO registers to command the vibrotactors, servos, and LEDs according to the parsed command
    c. If applicable, set timer interrupts for timed actuator motions

3.6 Device Specifications Summary

After many revisions, the front- and back-tactor controllers reached the state shown above. In general, both controllers are capable of providing dual 2-axis skin-stretch feedback to the fingerpad, accept user input via thumb joysticks, buttons, triggers, and bumpers, successfully enclose all electronics safely, and are ergonomically shaped so that they are comfortable and easy to hold. These controllers both meet the original design criteria, but still have room for improvement. The final specifications of the controllers are detailed below.

3.6.1 Front-Tactor Controller Specifications

The front-tactor controller is capable of providing skin-stretch feedback through the center of its dual thumb joysticks. It has enough buttons, triggers, and bumpers to satisfy simple game input requirements, but does not have as many buttons as modern game controllers. The specifications of the front-tactor controller are as follows:

- Weight: 280 g
- Input Voltage: 4.5 to 5.5 V
- Current Consumption: ~0.1 A - 2 A (depends on amount of torque applied to servos)
- Power Connector: 5.5 mm by 2.5 mm female barrel connector
- Communication Connector: Female DB9 connector
- Programming Connector: 5 pin SIP connector
- Maximum Unloaded Tactor Velocity (single axis): 42 mm/s
- Maximum Tactor Force (single axis): 36 N
- Tactor Workspace Size: 5.1 mm diameter circle
- X and Y Joystick Range of Motion: ±10.5 deg, ~33 mm radius
- X and Y Joystick Position Resolution: 8 bit, or ~0.08 deg
- Number of Tactile Switches: 10
- Communication Rate: 56700 baud rate, controller sends state to PC every 8 ms

3.6.2 Back-Tactor Controller Specifications

The back-tactor controller is capable of providing skin-stretch feedback through tactors on the back side of the controller. It has the same input capabilities as modern game controllers, which will allow this controller to potentially act as a replacement for

modern game controllers, provided the appropriate communication protocol is implemented on the PC and the microcontroller inside the controller. The specifications of the back-tactor controller are as follows:

- Weight: 340 g
- Input Voltage: 3.5 to 5.5 V
- Current Consumption: ~0.1 A - 3 A (depends on amount of torque applied to servos)
- Power Connector: 5.5 mm by 2.5 mm female barrel connector
- Communication Connector: USB mini
- Programming Connector: 5 pin SIP connector
- Maximum Unloaded Finger Tactor Velocity (single axis): 52.4 mm/s
- Maximum Finger Tactor Force (single axis): 29 N
- Finger Tactor Workspace Size: 5.1 mm diameter circle
- Maximum Unloaded Palm Tactor Velocity (single axis): 112.8 mm/s
- Maximum Palm Tactor Force: 14 N
- Palm Tactor Workspace Size: ~31.8 mm, or ~130 deg
- X and Y Joystick Range of Motion: ±20 deg
- X and Y Joystick Position Resolution: 8 bit, or ~0.16 deg
- Analog Trigger Resolution: 8 bit
- Number of Tactile Switches: 15
- Communication Rate: 64 byte packets at up to 1 kHz rate via USB, typical rates are 60 to 120 Hz

4 DEVICE VERIFICATION

The skin-stretch actuator design used in the back-tactor controller is based on the design presented by Montandon and Provancher [3]. They minimized significant inaccuracies in tactor paths due to the nonlinear kinematics of the design by moving the tactor along a trajectory with a series of waypoints (specified as bits to specify the PWM pulse width/length sent through the output compare module of the microcontroller, which determines the position of the respective servos). The result of using this series of waypoints can be seen in Figure 4.1.

While this method was successful in correcting the cue paths, the correction only applied to a limited set of motions for Montandon and Provancher (the 16 cues seen above). For the back-tactor controller, we instead used bi-linear interpolation of a lookup table between measured tactor positions and commanded servo positions to create a more general correction of the tactor workspace.

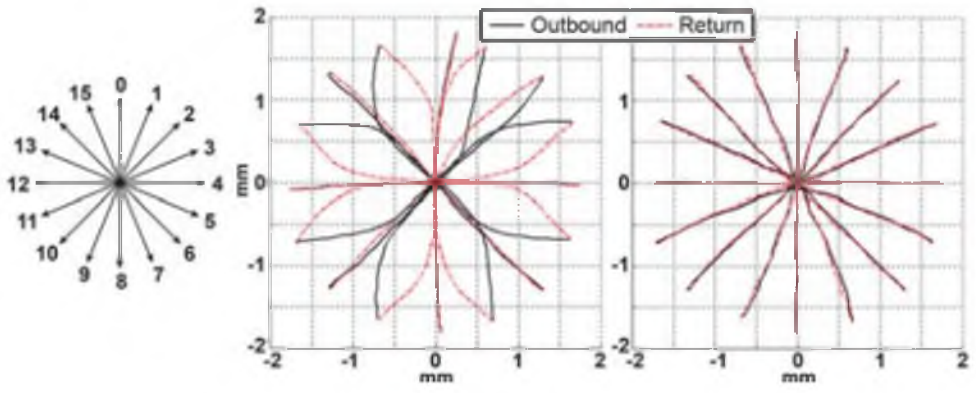


Figure 4.1: Uncorrected and corrected paths from Montandon and Provancher

4.1 Calibration Device and Setup

In order to obtain values for the lookup table, we needed a way to measure and log the positions of the tactor while it was controlled by the servos. A 2-axis measurement rig equipped with two linear probe encoders (model US Digital PE-500-2-I-S-L), PWM outputs to control the servos, and RS-232 communication to report values to a PC was used to obtain the tactor's position measurements. The linear probe encoders have a 0.0127 mm resolution with a 50.8 mm range of travel.

Figure 4.2 shows the layout of the calibration rig. Two linear encoders are mounted orthogonally and are linked to the tactor output via 0.5 mm diameter by 150 mm long spring steel wires. The spring steel wires attach to a custom tactor post. The custom tactor post allows the spring steel wires to rotate freely about the custom tactor post. The actuator assembly is mounted to the calibration rig.

Figure 4.3 shows the actuator assembly for the left tactor. The actuator assembly was created by taking the back-tactor controller design and cutting away all but the relevant parts of the sliding-plate mechanism. This was done so that the actuator assembly measured on the calibration rig is as similar as possible to the actuator assembly inside of the game controller (which would be more difficult to calibrate in its installed state/packaging). The entire servo, servo arm, spring steel wire, and sliding-plate assembly is directly transplanted from the game controller to the calibration rig. A mounting plate was also added so that the actuator assembly could be securely mounted to the calibration rig.

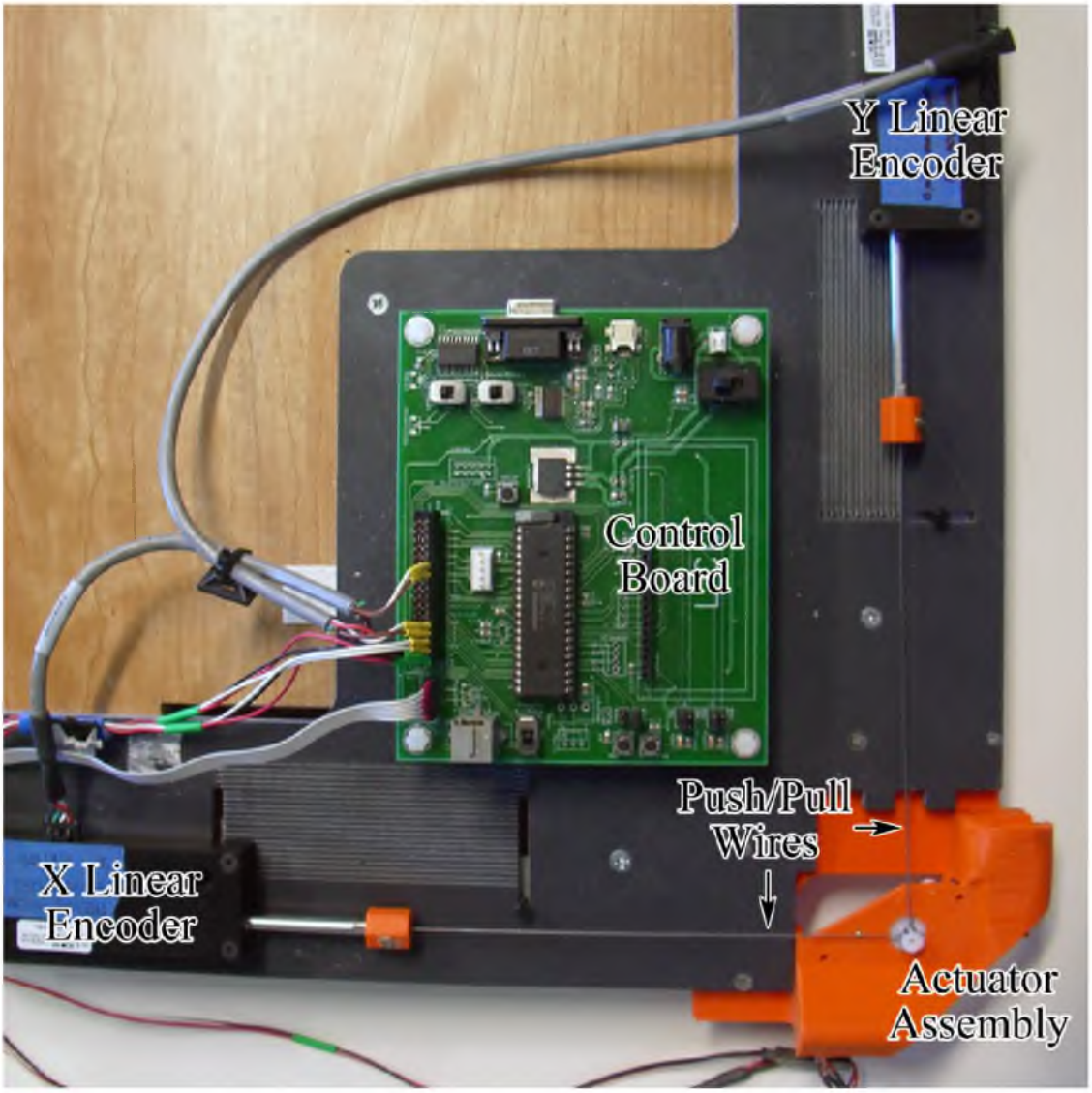


Figure 4.2: Labeled calibration rig

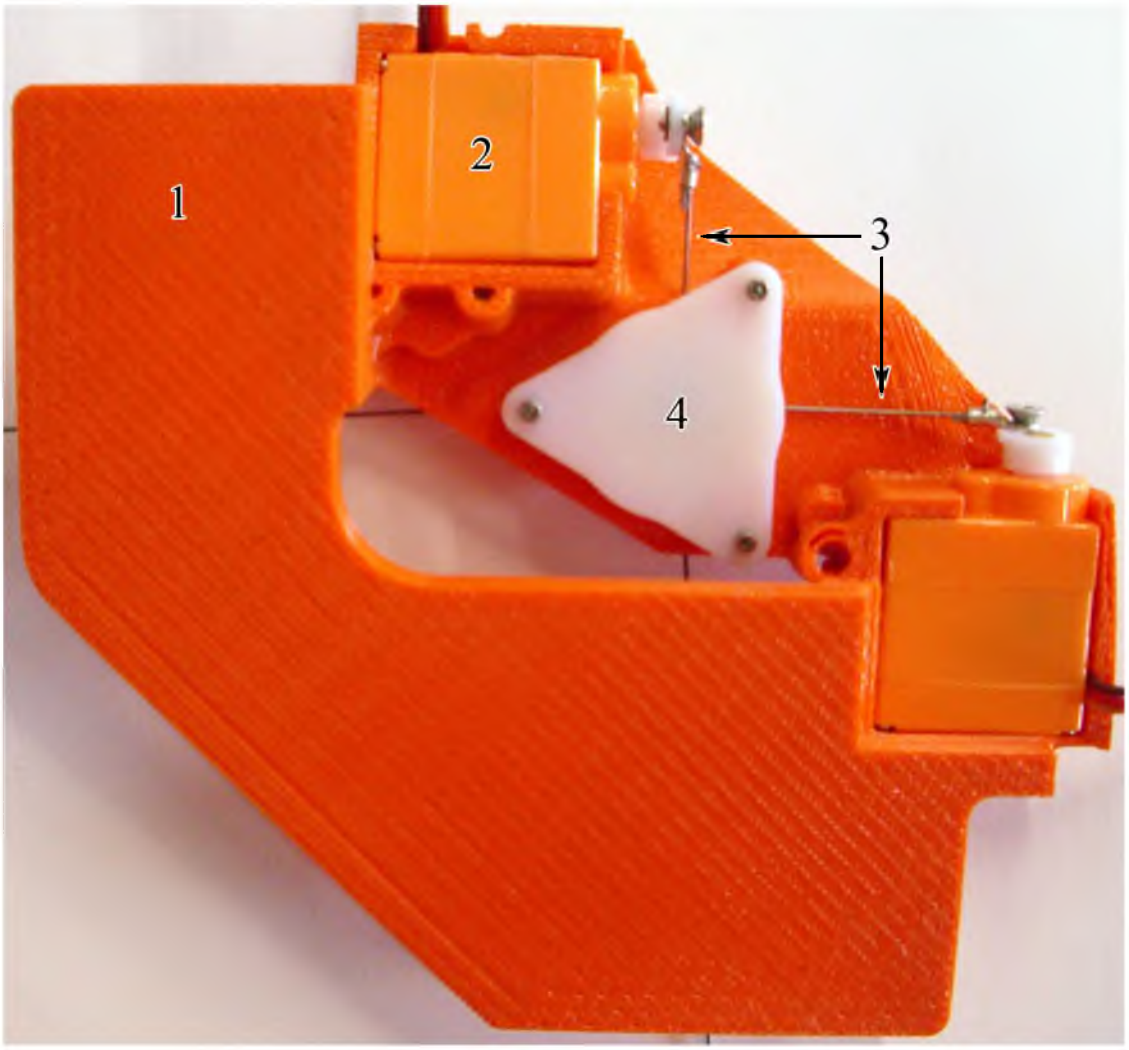


Figure 4.3: Labeled photo of actuator assembly. (1) Mounting plate (2) Servo (3) Spring steel wires (4) Support plate that supports the sliding plate

4.2 Bi-linear Interpolation Description

We used bi-linear interpolation of a 5 by 5 lookup table to convert desired tactor positions to servo commands (servo PWM pulse widths). Figure 4.4 is an illustration of the bi-linear interpolation process. Each point on the lookup table contains a pair of servo positions (the pulse widths sent to the X and Y servos) and a measured tactor position (X and Y positions measured by the linear encoders). To calculate the servo positions that should be commanded to reach a desired tactor position, we first determine which cell of the lookup table the desired tactor position is in.

We next calculate where the desired X tactor position lies in the cell. This is calculated according to:

$$X_{ratio} = \frac{X_{target} - X_{min}}{X_{max} - X_{min}} \tag{4.1}$$

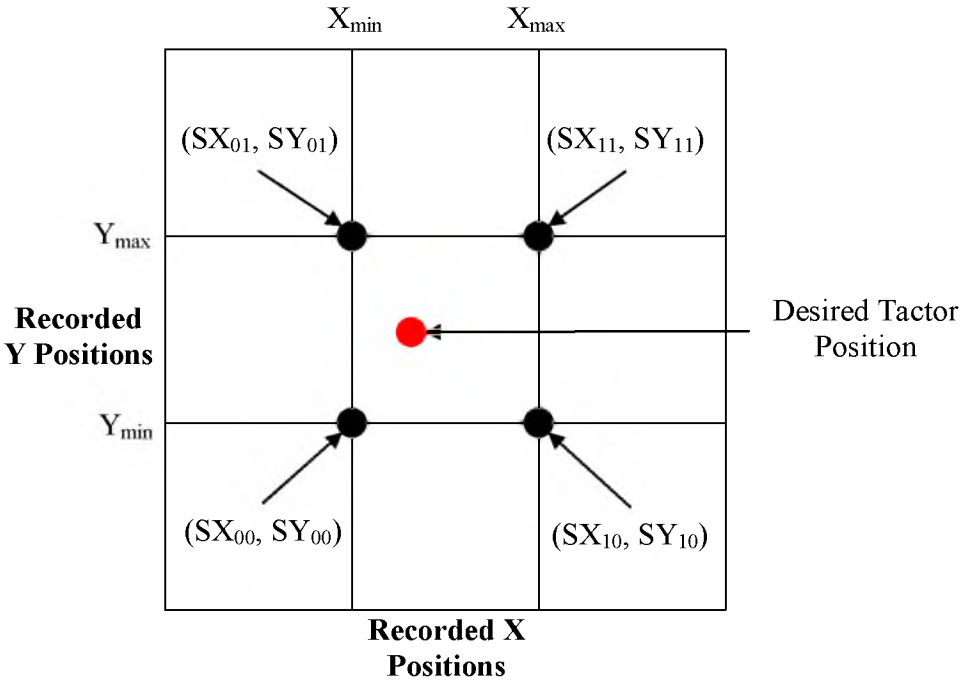


Figure 4.4: Illustration of lookup table

where X_{ratio} is the relative X position of the desired X tactor position within the cell, X_{max} is the larger X grid value, X_{min} is the smaller X grid value, and X_{target} is the desired X value. This same calculation is made using the corresponding Y values to obtain Y_{ratio}.

Some intermediate calculations are made using the following equations:

$$\begin{aligned} U_{x0} &= X_{ratio}(SX_{10} - SX_{00}) + SX_{00} \\ U_{y0} &= X_{ratio}(SY_{10} - SY_{00}) + SY_{00} \\ U_{x1} &= X_{ratio}(SX_{11} - SX_{01}) + SX_{01} \\ U_{y1} &= X_{ratio}(SY_{11} - SY_{01}) + SY_{01} \end{aligned} \tag{4.2}$$

where SX_{uv} and SY_{uv} are X and Y servo positions from the lookup table. The subscripts indicate where these values are in the lookup table (see Figure 4.4). The U values are then used to calculate the servo positions/commands according to:

$$\begin{aligned} X_{servo} &= Y_{ratio}(U_{x1} - U_{x0}) + U_{x0} \\ Y_{servo} &= Y_{ratio}(U_{y1} - U_{y0}) + U_{y0} \end{aligned} \tag{4.3}$$

where X_{servo} and Y_{servo} are the X- and Y-servo positions that should be commanded to the servos to move the tactor to the desired tactor position.

4.3 Original Versus Corrected Paths

The microcontroller inside the back-tactor controller was reprogrammed to implement the bi-linear interpolation method. Prior to the implementation of the bi-linear interpolation, the microcontroller would directly and proportionately command servo positions when messages are received from a PC at 60-120 Hz. After the implementation of the bi-linear interpolation, the microcontroller generates paths that update the commanded servo position at a rate of 300 Hz. The points along the path are calculated by first subdividing the vector between the desired tactor position and the current tactor position into a series of points. These points are then each fed through a function that

converts the desired tactor position into servo positions using the bi-linear interpolation method outlined in Section 4.2. The spacing between trajectory points is dependent on the desired velocity of the tactor motion.

Figure 4.5 shows the results of the bi-linear interpolation. The bi-linear interpolation did slightly improve the accuracy of the tactor motions. This is most noticeable for the right tactor, where the angles of some of the cues are slightly misaligned, and the lengths of the cues are not consistently the same.

It is important to note that the uncorrected motions for the back-tactor controller are noticeably more linear than the uncorrected motions in Figure 4.1. This is partially due to the use of longer spring steel wires (22.4 mm and 14 mm versus 7.7 mm and 18.9 mm) than those used by Montandon and Provancher [3]. The longer wires increase the radius of the arc on which the tactor travels when one of the servos moves the tactor. This reduces the nonlinearity of the tactor motion when actuated by a single servo. Also, Montandon and Provancher originally used a single outbound point for their uncorrected cues. By using a single outbound point, one servo would reach its target before the other for certain cue directions, resulting in a petal-shaped path for some cues. The uncorrected paths shown for the back-tactor controller utilize a series of waypoints. By commanding a series of positions, both servos reach their end target at nearly the same time.

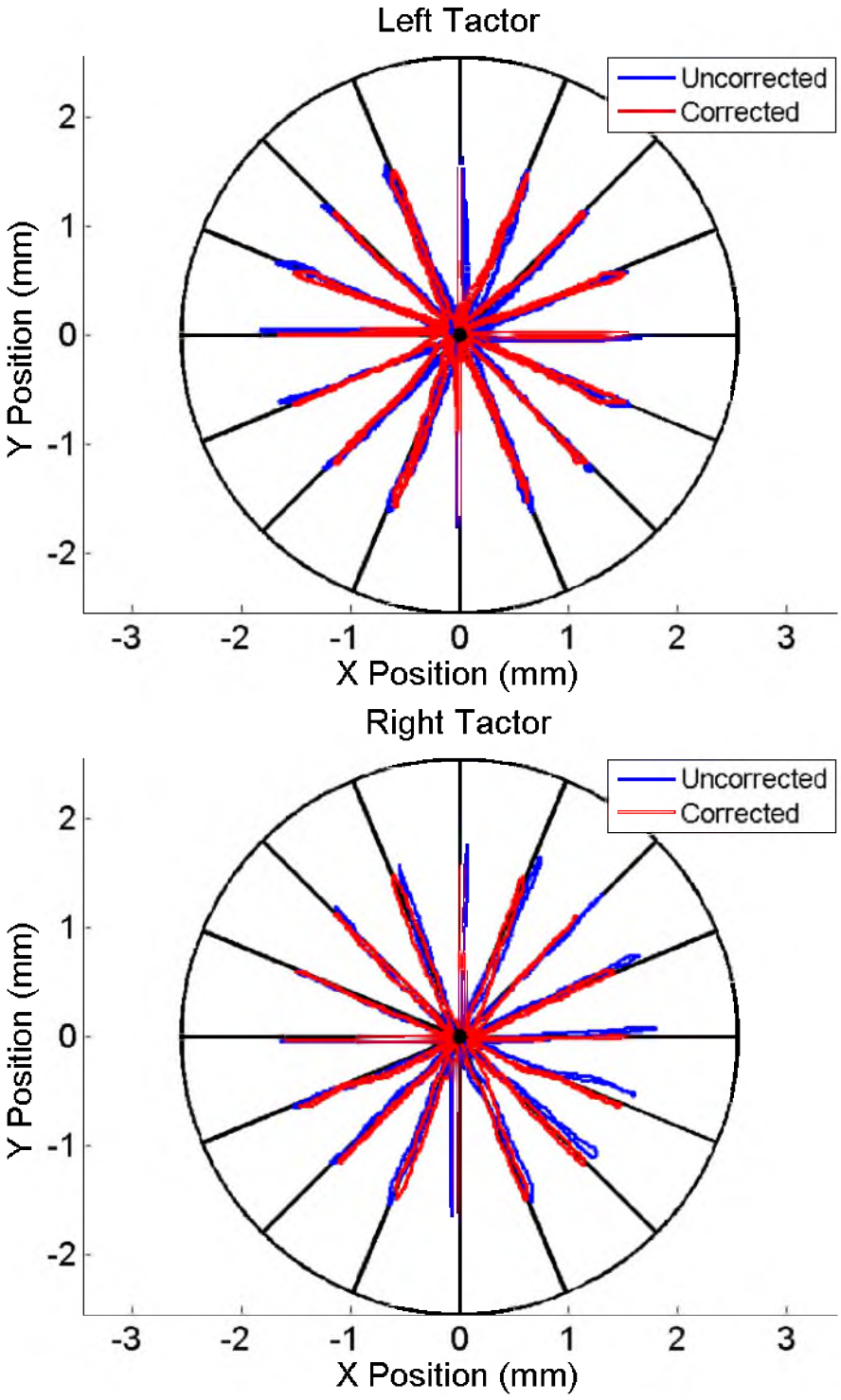


Figure 4.5: Uncorrected and corrected paths in 16 directions for right and left sides. The black lines and circle indicate ideal tactor angles and workspace limit, respectively.

5 EIGHT DIRECTION CUE EXPERIMENT

Prior work by Gleeson et al. [17] has shown that the direction of skin stretch applied to the index finger pad with displacements as low as 0.2 mm can be accurately perceived with >95% accuracy. Additional perception experiments by Guinan et al. [4] [5] [7] have measured how well skin-stretch feedback applied at a thumb joystick conveys direction information. Similar experiments, however, have not been performed with the back-tactor controller. The below experiment tests whether people are more accurate at identifying directional skin-stretch feedback given at the front or back side of the controller. Additionally, the effects of several different plastic inserts that restrain the middle finger in different ways are also tested.

This experiment was conducted under an institutionally approved human subjects IRB protocol.

5.1 Methods

This experiment measures the participants' abilities to perceive the direction of skin displacement cues on the pad of the thumbs or middle fingers. The directions rendered to the participant's fingers are limited to 8 evenly spaced directions, which will be referred to as N, NW, W, SW, S, SE, E, and NE. The direction cues each consist of an outbound motion in the cue direction at 2.5 mm/s, a 0.3 s pause, followed by a 40-50 mm/s (maximum speed of the actuators) motion back to the starting position, as seen in Figure

5.1. This direction cue is rendered to both tactors at the same time. The target displacement of the tactors during the cues is 1 mm. This particular cue length was chosen because we thought it would make the cues sufficiently difficult so that the participants' accuracy would not exhibit a ceiling effect at near 100%. This decision was made after running several pilot tests. Participants receive direction cues on both thumb/index fingers and respond to the skin-stretch cues via the right thumb joystick with the assistance of an on-screen selector that participants use to register their interpretation of the direction cue.

Figure 5.2 shows the differences between the 6 test conditions. For this experiment, cues are either rendered via the front-tactor controller or via the back-tactor controller with one of the 5 different finger restraints attached. The first condition shown is the front-tactor controller. The second image shows the back side of the back-tactor controller. The remaining 5 images show close up views of the left finger restraints removed from the back-tactor controller.

The flat restraint (Figure 5.2(c)) is a flat surface with a conical aperture. This restraint is the most similar to apertures used in a study by Guinan et al. [4]. The grooved restraint (Figure 5.2(d)) is curved so that it conforms to the width of the user's finger. A version of the grooved restraint (Figure 5.2(e)) with hard insert is included to determine whether there is a performance difference between the rubber (made of Objet's TangoPlus rubber-like material) and hard (made of Objet's VeroWhite Plus material) inserts. All other back-tactor finger restraints have a translucent rubber insert (made of Objet's TangoPlus rubber-like material). The bottom wall restraint (Figure 5.2(f)) is similar to the grooved restraint, but additionally has a bottom wall that prevents the user's finger from sliding

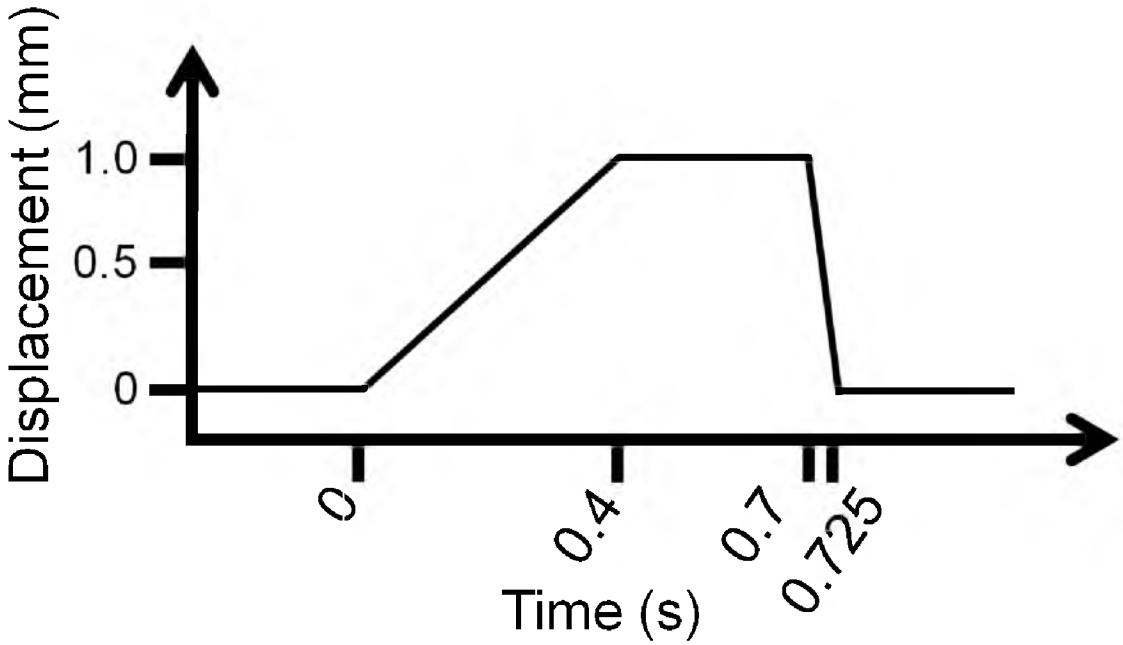


Figure 5.1: Plot of tactor displacement over time for direction cues

Figure 5.2: The 6 conditions tested included (a) the front-tactor controller (b) and the back-tactor controller with the following middle finger-restraint inserts: (c) Flat restraint with rubber insert (d) Grooved restraint with rubber insert (e) Grooved restraint with hard insert (f) Bottom wall restraint with rubber insert (g) Middle digit restraint with rubber insert

down from the aperture. The foam padding seen attached to the bottom wall was included to better center the user's finger on the aperture. Last, the middle digit restraint (Figure 5.2(g)) uses two foam padded walls to restrain the user's finger from sliding up or down from the aperture and has the advantage of having no restraint contact with the fingerpad near where the tactor makes contact with the fingerpad.

The test is broken into 6 sections; there is one section for each restraint condition. Each section starts with a practice session followed by a test session. During the practice session, cues are rendered in the four cardinal directions, N, S, E, and W. Each of these directions is rendered twice during the practice session. The order in which the 8 cues are rendered is random.

During the test session, cues are rendered in one of 8 directions. A total of 64 cues, or 8 repetitions of each direction, are rendered during this session. The cues are semi-randomly ordered; each of the 8 sets of 8 directions is randomized and added to a list of cues to be given. This ensures that participants will receive the same cue twice in a row at most.

We tested 24 participants, 4 of which were female. Participant ranged in age from 21 to 41 years, with a mean of 28.25 years with a standard deviation of 4.9 years. Participants were not compensated. Participants were chosen so that half had previous experience with the skin-stretch game controllers, and half did not. The experienced and inexperienced halves were each split into groups of 6. A 6 by 6 balanced Latin Square was used to determine the order in which conditions would be presented to each group of 6 participants. This was done to minimize possible ordering effects.

5.1.1 Hardware and Setup

Hardware for the experiment consisted of a PC, two monitors, noise canceling headphones, a chair, and the front- and back-tactor controllers. The microcontroller inside each of the game controllers was programmed with custom code that slowed the outbound motion of the tactors to 2.5 mm/s by commanding a position trajectory to the servos. The front-tactor controller was programmed to generate cues of length 1.15 mm so that the tactors actually displace approximately 1 mm due to the compliance in the actuator/flexure mechanism. The back-tactor controller generates cues of length 1 mm. The back-tactor controller uses bi-linear interpolation (i.e., the tactors' motions are calibrated and positions are commanded using a table lookup and interpolation) to improve the accuracy of the paths as was described in Chapter 4.

Figure 5.3 shows the setup of the hardware used in the experiment. Two monitors are on the desk. One monitor faces the participant and displays the direction selector experiment user interface and occasional instructions. The display of the second monitor is angled so that only the test proctor can see it. The second monitor displays both the direction selector and the most recent cue given to the participant. The noise canceling headphones play white noise and tones during the experiment.

5.1.2 Software Details

The experiment software for the PC was written in C#. The software is responsible for displaying instructions and a direction selector experiment user interface, recording participant responses, playing tones when participants select responses, and communicating with the game controllers. Figure 5.4 depicts the flow of the experiment

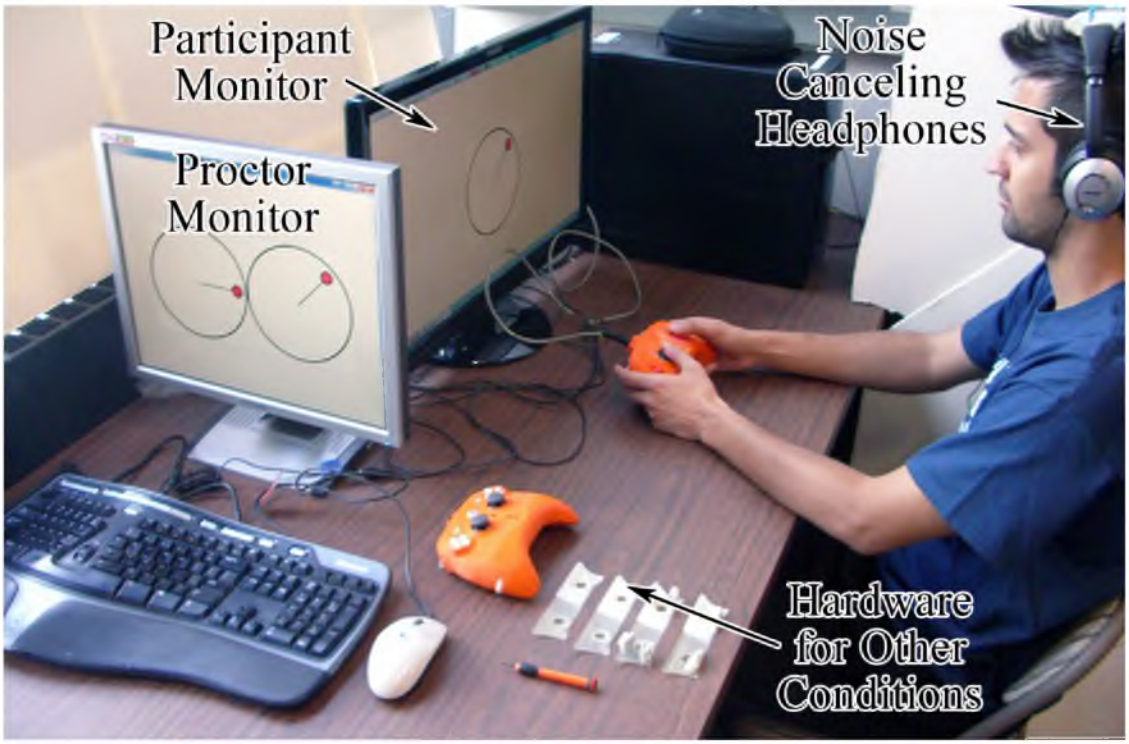


Figure 5.3: Labeled photo of experiment setup with participant

program. The program first asks for input about the participant that is recorded in the participant's data file.

The program then displays instructions for a short practice session. After the participant has responded to the 8 practice cues, instructions are displayed on the screen for the test session. After the participant completes the test session, the program pauses and displays instructions for the practice session again. This process repeats for the 5 remaining test conditions.

Before each session (practice or test) starts, the monitor displays instructions that say to press the trigger to begin. After the participant presses the right trigger, the session starts. The monitor displays a selection interface, and the controller renders a direction cue. Figure 5.5 shows the selection interface during various stages of selection. The selection interface indicates the direction that the right thumb stick is being pointed at by the participant. This on-screen selection interface is used to ensure that we record the participant's intended response. The on-screen indicator changes from red to green and a tone plays through the headphones after the participant selects a direction by pressing the left bumper. The next cue is rendered after a short pause after the participant returns the thumb stick to the center position. This pause varies randomly in duration from 800 to 2000 ms.

5.1.3 Proctor and Participant Interaction

The participant is seated in a chair at the desk. The monitor and a game controller are on the desk in front of the participant. The test proctor provides the participant written experiment instructions detailing the purpose of the experiment. The proctor then asks the

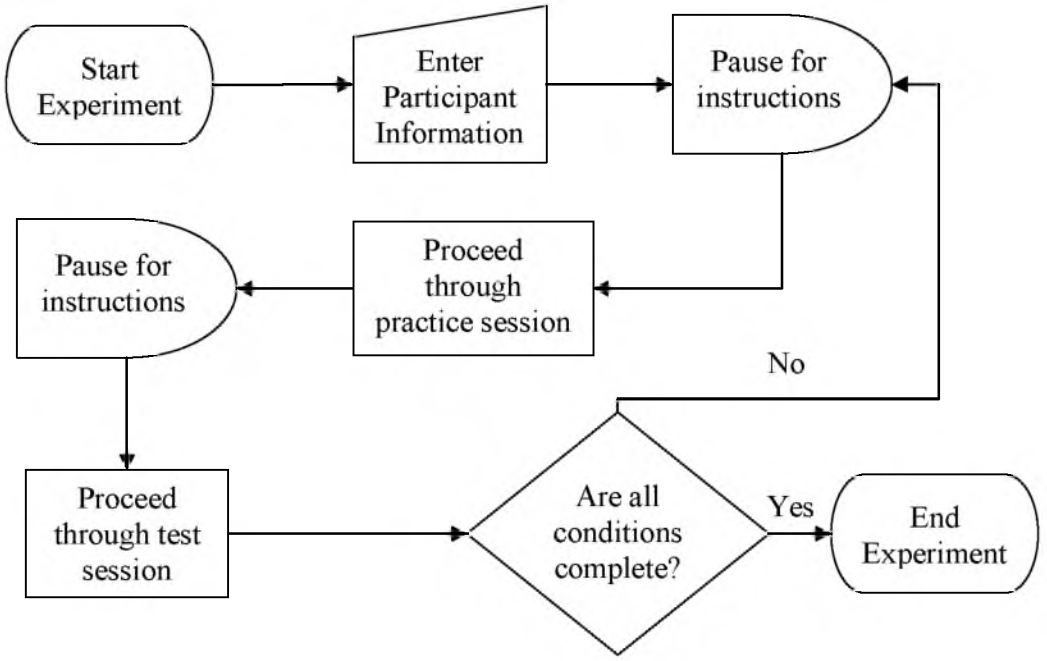


Figure 5.4: Flow diagram of experiment software

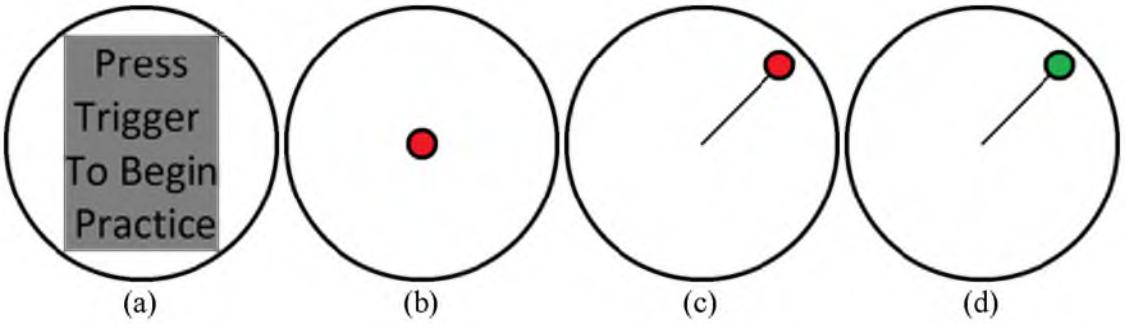


Figure 5.5: Examples of on-screen visual feedback. (a) Instruction screen (b) Centered selection interface (c) Selection interface showing current direction selection (d) Selection interface indicating a successfully recorded response

participant if they have any questions after the participant finishes reading the instructions. The proctor starts the experiment after answering any questions and reiterating key points of the instructions. The key points are:

- There are 6 sections to the experiment.
- Each section consists of a short practice session prior to a test session.
- Skin-stretch cues will be in one of 8 directions during the test sessions, but only one of 4 directions during the practice sessions.
- Keep your middle fingers (or thumbs for the front-tactor controller) on the tactors while the test is in progress.
- After feeling a direction cue on both fingers, move the right thumb joystick in the direction of the cue as soon as you think you know what the direction is.
- Press the left bumper after verifying that the on-screen selector is pointing in the direction you intend to respond with.

The proctor hands the participant a game controller and verifies that the participant is holding it correctly. The participant should have their thumbs on the thumb joysticks, their index fingers on or near the triggers, their middle finger on the back of the controller (resting on the tactor when holding the back-tactor controller), and remaining fingers wrapped around the handles.

The proctor changes the game-controller hardware for the next condition after the participant finishes each test session. The proctor either switches controller type or replaces the finger restraint for the back-tactor controller. The practice session, test session, and hardware change are repeated for the 6 different conditions. The participant

is asked to complete a survey after the experiment. The experiment lasts approximately 1 hour, depending on how fast the participant responds.

5.2 Results and Discussion

Response accuracy is the primary metric by which performance is judged for this experiment. A participant's response is considered correct if their response direction falls within ±22.5 degrees of the cue direction. Participant response speed was recorded, but is not necessarily a reliable measure of performance. This is because participants were not penalized for taking longer to respond to cues. Participants were instructed to move the thumb joystick in the direction they thought the cue was in as soon as they thought they knew what the direction was, but to not record the response until they were sure that the on screen selector was pointing in the direction they intended to respond with. Response times for both the initial joystick motion and final selection were both recorded.

Figure 5.6 shows the mean accuracy across all directions and conditions for each participant along with a 95% confidence interval. Participant 19 has a mean accuracy that is more than 3 standard deviations from the mean accuracies of the remaining participants. For this reason, it is suspected that the performance of participant 19 is not typical of the population. Participant 19's data has been excluded from the following analyses.

5.2.1 Results Based on Condition

Figure 5.7 shows the response time means and accuracy means of all participants combined for the 6 different conditions. A 2-way repeated measures ANOVA showed

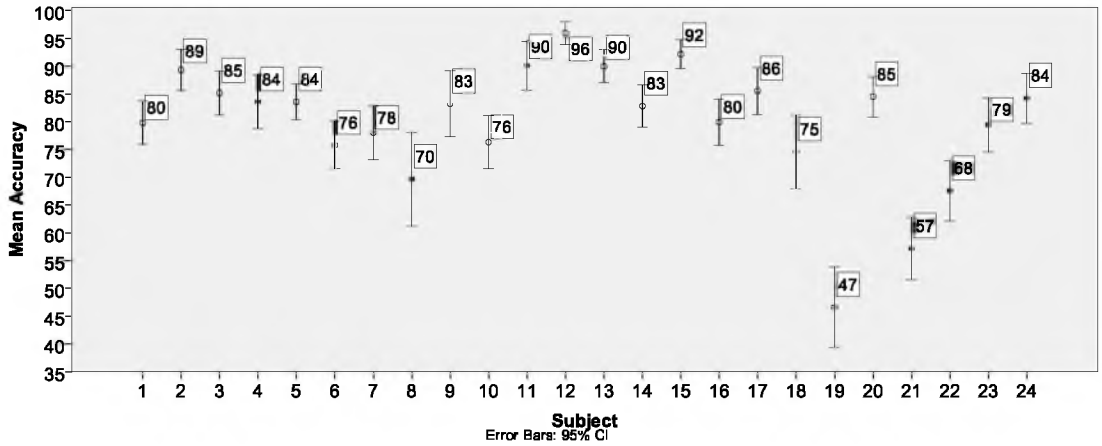


Figure 5.6: Plot of mean accuracy for each participant for all test conditions along with 95% confidence intervals.

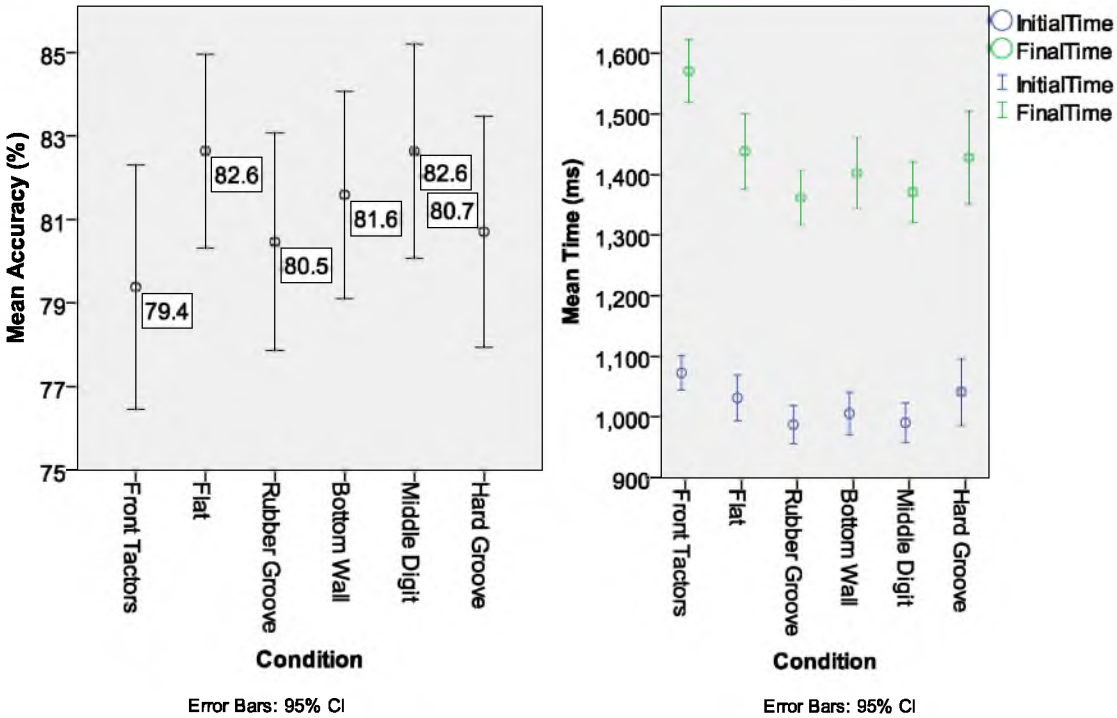


Figure 5.7: Plots showing participant response accuracy and response time for each restraint/feedback condition along with 95% confidence intervals.

that at least one of the restraint conditions was significantly different than the others [$F(5,17954)=3.471$, $p=0.004$], after accounting for cue direction, subject, and interaction between cue direction and condition. A post-hoc Tukey's test showed that the front-tactor condition had significantly lower accuracy than the flat and middle digit restraint conditions [$p=0.012$ for both].

The difference between the maximum (middle digit restraint) and minimum (front-tactor controller) accuracy means is only 3.2%. This small difference indicates that for this particular direction identification task, restraint condition is of minimal importance.

Both mean initial response time and mean final response time were plotted for each condition. The front-tactor condition has a significantly slower final response time than all of the other conditions. A 2-way repeated measures ANOVA showed that at least one of the restraint conditions was significantly different than the others [$F(5,17954)=3.471$, $p=0.004$], after accounting for cue direction and subject. A post-hoc Tukey's test showed that the front tactor restraint condition had significantly slower response times than all other conditions [$p<0.001$ for all].

This could be due to the fact that for the front-tactor condition, participants are responding at the same location at which they are receiving feedback. This could be causing participants to sometimes wait for the cue to completely finish returning to center before responding. Slight ergonomic differences between the two controllers could also be responsible for the difference. For example, the thumb joysticks in the two controllers have different angular ranges and radii of motion. The return spring force for the joysticks likely varies as well (we did not design for a specific spring force).

Figure 5.8 shows the differences in accuracy between experienced and inexperienced participants for each restraint condition. Experienced participants performed much more consistently across test conditions than inexperienced participants, and performed better overall. An independent-samples t-test showed that there was a significant difference in accuracy between experienced (M=0.835, SD=0.371) and inexperienced (M=0.787, SD=0.409) participants [$t(17662)=8.157$, $p<0.001$]. This indicates that as participants become more used to these controllers, their overall performance should improve and their performance should become less dependent on the ergonomics or orientation in which the skin-stretch feedback is given.

A survey was conducted after the experiment which asked participants questions about their preferences regarding the different restraint conditions.

Figure 5.9 is a plot of participants' mean responses to three types of questions. The questions asked participants how much they liked/disliked, their perceived performance, and ease of interpretation for each condition. Participants chose from a preference scale from 1 to 5. A 1 corresponded to very much disliked, very poor performance, or very difficult to interpret. A 5 corresponded to very much liked, very good performance, or very ease to interpret. Scores of 3 were neutral responses. The results show that the bottom wall restraint condition had the highest scores in likeability and ease of interpretation, and the middle digit restraint condition had the highest perceived performance. The flat restraint condition, on the other hand, had the lowest scores in all three categories. This is interesting, as the flat restraint condition was shown above to have significantly higher performance than the front tactor restraint condition, and performance equal to that of the middle digit restraint condition.

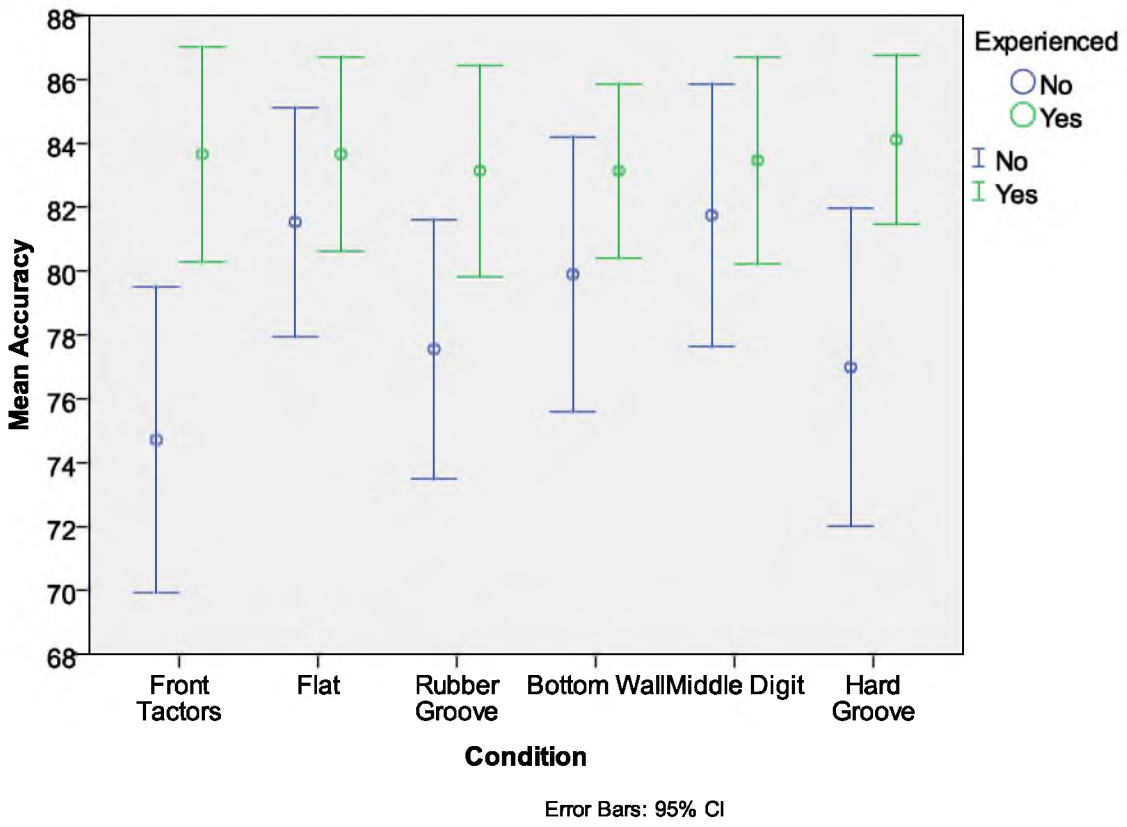


Figure 5.8: Plot of response accuracy for each condition based on participant experience level. Experienced users had uniform response accuracy regardless of restraint/feedback condition.

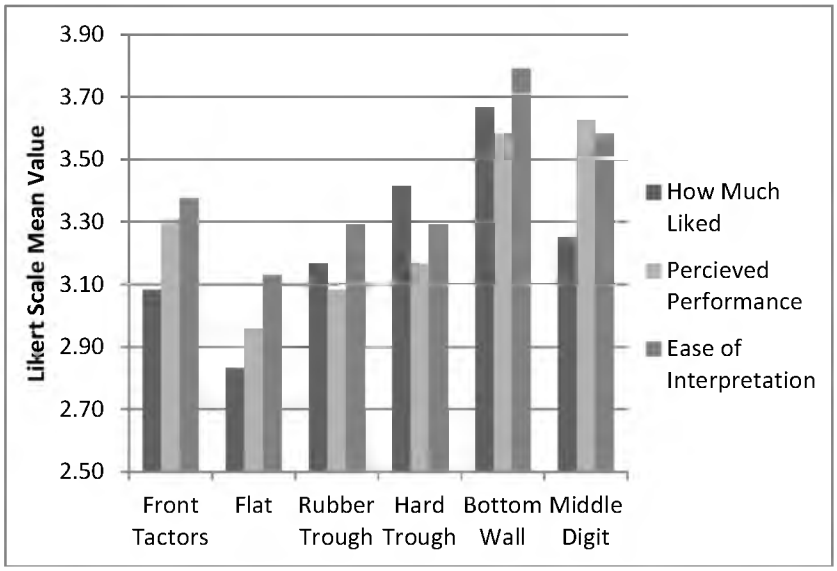


Figure 5.9: Plot of participants' opinions regarding condition preference, perceived performance, and ease of interpretation

5.2.2 Results Based on Cue Direction

Figure 5.10 shows the mean response time and mean accuracy of all participants combined for the 8 different cue directions. The mean accuracy plot for the cue directions clearly shows that cues in the cardinal directions have higher mean accuracies than the cues in diagonal directions An independent-samples t-test showed that there was a significant difference in accuracy between cardinal (M=0.859, SD=0.347) and diagonal (M=0.765, SD=0.424) directions [$t(17662)=16.113$, $p<0.001$]. Table 5-1 shows that this increased response accuracy for the cardinal directions exists for all conditions. It is possible that this is due to the "oblique effect," a tactile sensitivity bias toward the vertical or horizontal versus the oblique angles, as detailed by Gentaz et al. [18], and not due to increased tactile sensitivity in those directions due to relative density of mechanoreceptors in participants' fingers. If the oblique effect is the

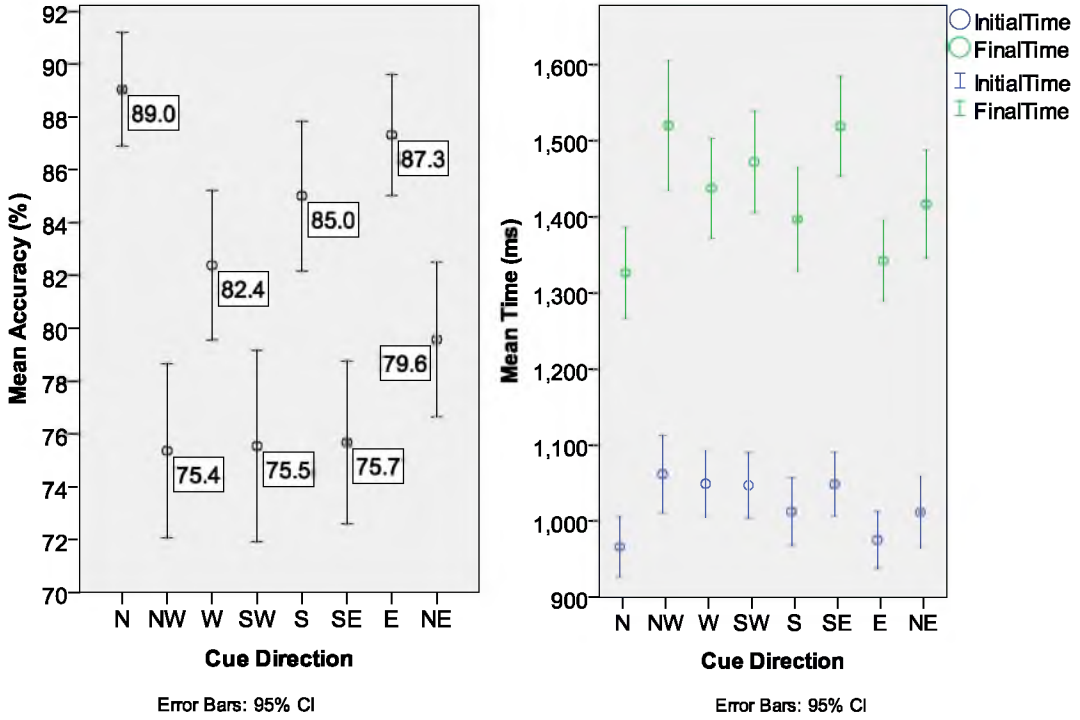


Figure 5.10: Plots showing participant response accuracy and response time for each cue direction along with 95% confidence intervals.

Table 5-1: Confusion matrices of aggregate cue directions and participant responses for each restraint condition

Front Tactor

Rendered Direction \ Response Direction	N	NW	W	SW	S	SW	E	NE	Sum
N	330	7	0	0	0	1	2	28	368
NW	60	268	35	0	0	0	1	4	368
W	3	59	291	8	2	0	2	3	368
SW	2	6	77	253	13	5	3	9	368
S	0	0	1	23	316	23	5	0	368
SE	0	3	0	1	35	248	80	1	368
E	0	0	1	0	1	7	317	42	368
NE	14	0	2	0	0	0	38	314	368
Sum	409	343	407	285	367	284	448	401	2944

Flat

Rendered Direction \ Response Direction	N	NW	W	SW	S	SW	E	NE	Sum
N	332	11	1	0	0	0	0	24	368
NW	45	281	42	0	0	0	0	0	368
W	1	34	311	21	0	1	0	0	368
SW	1	2	48	299	17	0	0	1	368
S	0	1	0	23	308	34	1	1	368
SE	0	0	0	1	36	279	52	0	368
E	0	1	2	0	0	32	325	8	368
NE	16	1	0	0	1	0	52	298	368
Sum	395	331	404	344	362	346	430	332	2944

Rubber Groove

Rendered Direction \ Response Direction	N	NW	W	SW	S	SW	E	NE	Sum
N	333	10	1	0	0	1	1	22	368
NW	56	274	34	1	1	0	1	1	368
W	1	47	300	18	0	0	2	0	368
SW	1	1	72	266	26	0	1	1	368
S	0	1	0	33	300	30	4	0	368
SE	0	0	0	1	30	277	60	0	368
E	0	0	0	0	0	24	329	15	368
NE	18	0	0	0	0	0	60	290	368
Sum	409	333	407	319	357	332	458	329	2944

Bottom Wall

Rendered Direction \ Response Direction	N	NW	W	SW	S	SW	E	NE	Sum
N	325	17	1	0	0	0	1	24	368
NW	49	289	30	0	0	0	0	0	368
W	0	37	316	15	0	0	0	0	368
SW	3	4	45	275	39	1	0	1	368
S	1	1	0	17	318	30	1	0	368
SE	0	0	1	0	33	280	52	2	368
E	1	0	0	0	2	30	323	12	368
NE	29	2	0	0	0	0	61	276	368
Sum	408	350	393	307	392	341	438	315	2944

Middle Digit

Rendered Direction \ Response Direction	N	NW	W	SW	S	SW	E	NE	Sum
N	337	13	0	1	0	0	0	17	368
NW	50	283	31	0	0	0	1	3	368
W	2	37	309	19	1	0	0	0	368
SW	1	5	50	285	26	1	0	0	368
S	1	1	0	22	319	22	3	0	368
SE	0	0	1	1	23	298	45	0	368
E	2	0	1	0	0	41	314	10	368
NE	21	0	1	0	0	0	58	288	368
Sum	414	339	393	328	369	362	421	318	2944

Hard Groove

Rendered Direction \ Response Direction	N	NW	W	SW	S	SW	E	NE	Sum
N	309	24	1	0	1	2	0	31	368
NW	53	269	38	1	2	0	2	3	368
W	2	50	292	19	1	0	1	3	368
SW	0	4	49	290	24	0	0	1	368
S	1	1	0	9	316	39	1	1	368
SE	2	0	1	2	22	289	51	1	368
E	4	0	0	3	0	15	320	26	368
NE	26	2	1	0	2	1	45	291	368
Sum	397	350	382	324	368	346	420	357	2944

result of relative mechanoreceptor density, we might expect to see increased response accuracy in the diagonal directions for the front-tactor results and increased response accuracy in the cardinal direction for the back-tactor controller (the user's thumb is angled toward the centerline of the front-tactor controller at approximately 45 degrees, while the user's middle finger is oriented approximately 90 degrees relative to the center plane of the back-tactor controller). Because we do not see this 45 degree rotation in response accuracy between the thumb and middle fingers, it is more likely that the increased response accuracy in the cardinal directions is due to a tendency to respond in the cardinal directions.

In order to investigate if the apparent "oblique effect" is due to variations in sensitivity in different directions, d' values (an unbiased measure of response accuracy) were calculated for each rendered cue direction and restraint condition, as seen in Table 5-2. According to Macmillan and Creelman [19], d' is the most widely used sensitivity measure of detection theory. The calculations are based on the ratios of correct responses (hit rate) and false-alarm responses (false-alarm rate) for a particular cue direction. The equation used to calculated d' is:

$$d' = z(H) - z(F) \tag{5.1}$$

where z(H) is the z score of the hit rate and z(F) is the z score of the false alarm rate. Table 5-2 shows that the cardinal directions have higher d' values than the diagonal directions. This suggests that the "oblique effect" shown above is due to higher sensitivity in the cardinal directions.

Figure 5.11 shows the mean angular error between user's responses and the direction cue for each cue direction. The plot shows angular error for both the front- and back-

Table 5-2: d' values for restraint conditions and rendered cue directions calculated using (5.1)

	N	NW	W	SW	S	SE	E	NE	Combined
Front Tactors	3.61	2.69	3.02	2.59	3.37	2.47	3.46	3.42	**3.03**
Flat	3.48	2.87	3.29	3.14	3.21	2.95	3.54	3.07	**3.18**
Rubber Trough	3.52	2.80	3.07	2.64	3.05	2.91	3.57	2.92	**3.03**
Bottom Wall	3.42	3.02	3.50	2.80	3.29	2.91	3.59	2.80	**3.14**
Middle Digit	3.65	3.01	3.21	2.88	3.37	3.23	3.28	2.88	**3.17**
Hard Trough	3.11	2.75	2.99	2.94	3.37	3.04	3.52	3.01	**3.07**
Combined	**3.46**	**2.85**	**3.17**	**2.82**	**3.27**	**2.90**	**3.49**	**3.01**	**3.10**

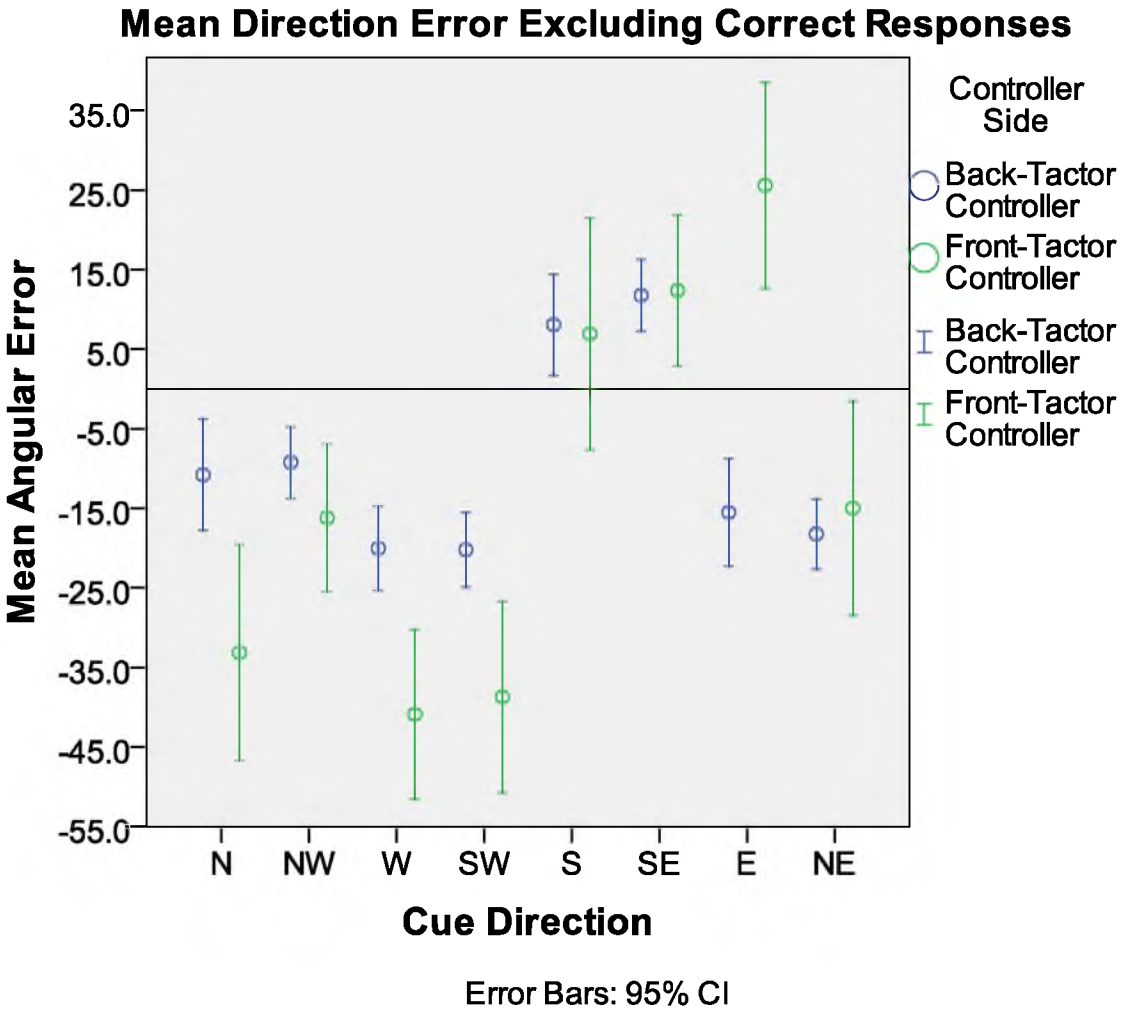


Figure 5.11: Plot of mean angular error in participant responses for each cue direction

tactor controller. Correct responses are excluded from the mean angular errors so that we can better see the relative differences in response direction error. A series of one-sample t-tests showed that for the back-tactor controller, all 8 directions have mean angular errors that are significantly different from 0 [$t(>203)>2.492$, $p<0.013$].

There is also an overall small negative (or clockwise) response direction bias. A one-sample t-test showed that there was a significant difference between the mean angular error ($M=-1.86$, $SD=22.647$) and 0 [$t(17663)=-10.944$, $p<0.001$]. It is unclear why there is an overall negative response direction error. We expected there to be a zero mean response direction error because feedback was simultaneously rendered to both hands, as happened in the study by Montandon and Provancher [3]. It is possible that the participants focused more on the right side of the controller since that is where they were responding to the cues.

By focusing on one side of the controller, participants could have biased their responses toward the cues from that side of the controller. For the front-tactor controller, participants' right thumbs were aligned approximately 45 degrees counter-clockwise relative to the Y-axis of the right tactor. This rotation causes the skin-stretch cues to be rotated clockwise relative to the user's right thumb, which may cause participants to respond in the clockwise direction relative to the rendered cue direction. For the back-tactor controller, participants' right middle fingers tended to be rotated counter-clockwise relative to the X-axis of the right tactor. This rotation causes the skin-stretch cues to be rotated clockwise relative to the user's right middle finger, which may cause participants to respond in the clockwise direction relative to the rendered cue direction.

5.2.3 Results Summary

The front tactor condition had significantly lower accuracy than the flat and middle digit conditions. Although significant, the differences in response accuracy between all constraint conditions are small (<3.2%). This indicates that for simple skin-stretch cues, both the front- and back-tactor controllers provide roughly equivalent communication of cues. This result also indicates that any of the back-tactor restraints can be used with roughly equal results, allowing users to choose the restraint they prefer without large consequence, though the highest numerical accuracy was found for the flat and middle digit restraint types (mean accuracy of 82.6 % for both) on the back-tactor controller.

We also found a large oblique effect where participants more accurately and more often responded in the cardinal directions, for all conditions. This implies that regardless of finger or tactor orientation, users may exhibit an increased sensitivity in the cardinal directions

Finally, we found that participants had a tendency to respond in the clockwise direction relative to direction cues. It is possible that by having users respond to cues on the right thumb joystick, we are drawing their attention away from tactors on the left side of the controller and toward tactors on the right side. This was not observed by Montandon [3] when providing cues to both thumb tips, but his participants responded verbally to direction cues. By shifting our participants' attention to the right side of the controller, their sensitivity may have been drawn toward the right side of the controller. Further tests that alternate which thumb joystick users respond with could determine whether this effect manifests for the left thumb joystick.

6 TARGETING PERFORMANCE EXPERIMENT

An experiment was designed to measure the effectiveness of three skin-stretch feedback strategies at directing a user to face a randomized target angle ("target") in a virtual environment. The feedback strategies provided direction and proximity information to the user. The feedback strategies were also rendered from 3 different locations on the game controllers to the user's hands.

This experiment was conducted under an institutionally approved human subjects IRB protocol.

6.1 Methods

The experiment measures how quickly participants can rotate under rate control, using the controller's left thumb joystick, to face target angles within a virtual environment when provided with a variety of skin-stretch feedback strategies. The experiment evaluates 3 different feedback strategies, applied at 3 different locations. All feedback strategies vary their feedback as a function of the angle between the target and the user's heading. Also, all feedback strategies become stationary when the user's heading is within a deadband window surrounding the target. The target window is a ± 7.5 ° band surrounding the target.

The 7 conditions being tested in this experiment are: front-sinusoidal, front-sustained, front-pulsed, back-sinusoidal, back-sustained, back-pulsed, and palm-sinusoidal

feedback. The palm location is only capable of rendering the sinusoidal feedback since the palm tactors only have 1 DOF.

We tested 14 participants, 1 of which was female. The participants had a mean age of 29.6 years, with a standard deviation of 6.1 years. All 14 participants self-identified as right-hand dominant. Participants were not compensated. A pair of 7 by 7 balanced Latin Squares was used to determine the order in which conditions would be presented to the 14 participants. This was done to minimize possible ordering effects.

The test was divided into two sections that participants completed on different days. Participants were presented with 3 conditions the first day, and 4 the second day. This was done to avoid overly fatiguing the participants. Each day's sections took between 30 minutes and 1 hour on average to complete.

One hundred (100) targets were presented for each of the 7 conditions. The 100 targets were broken into rounds of 10 targets with short breaks in between each round. Each round of 10 targets is comprised of 30, 65, 100, 135, or 170 ° clockwise (CW) and counter-clockwise (CCW) targets. These ten unique target headings are presented in a semirandom order.

To create each semirandom set of 10 targets, the list of allowable clockwise and counter-clockwise targets was first shuffled separately to create two randomly ordered lists. A combined list was then made by semirandomly selecting targets from one of the two lists, one target at a time. The chance of choosing a target from the CCW list instead of the CW list was defined by the following equation:

$$CCWchance = \frac{numCW+1}{numTotal+2} \tag{6.1}$$

where numCW is the number of targets that have been selected from the CW list and numTotal is the total number of targets that have been selected from both lists. Once all of the targets from one of the two lists have been depleted, the remaining targets in the other list are added to the end of the combined list. This semirandom selection reduces the chances of having a large number of targets in the same CW or CCW direction from occurring consecutively.

6.1.1 Feedback Strategies

There are three feedback strategies tested in this experiment. Two of the feedback strategies (sustained and pulsed) are nearly identical to feedback strategies used by Guinan et al. [7] in their gaming tasks. The third feedback strategy is substantially different than the other two in that it only provides feedback on a single tactor and requires only a single DOF of motion on that tactor, thus allowing it to be rendered using the palm tactors in the handles of the back-tactor controller.

The first feedback strategy moves the tactors around a circular (circumferential) path as a function of the angle between the user and the target, as shown in Figure 6.1. The circular path has a radius equal to the maximum displacement of the tactor and is concentric with the tactor aperture. This feedback strategy will be referred to as sustained feedback. When the user's heading is outside the target window, the angular position of the tactor on the circle is mapped to the angle between the user and the target. When the user's heading is within the target window, the tactor stops at the North (forward) direction. This heading and lack of motion indicates that the user is on target. The left and right sides of the controller are moved simultaneously and in the same manner.

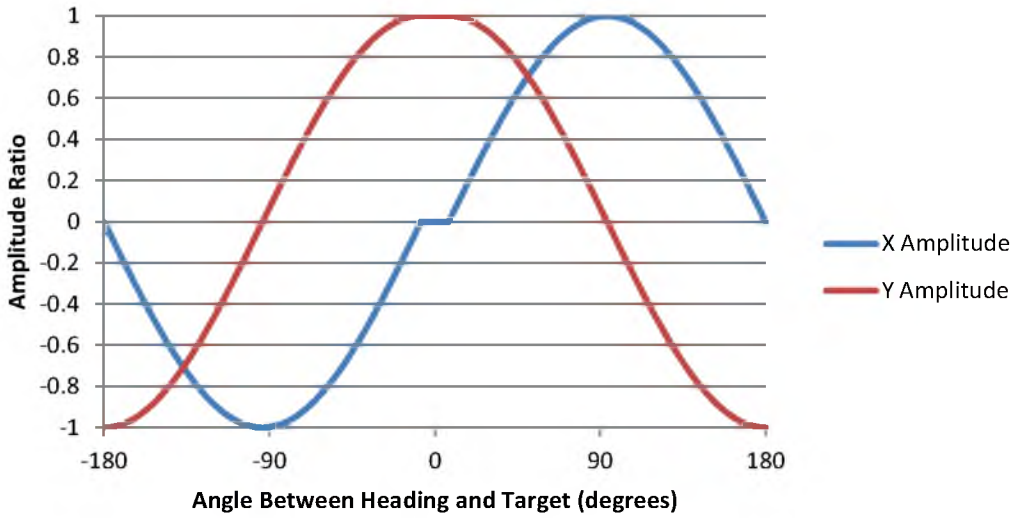


Figure 6.1: Plot of tactor position as a function of the angle between the user's heading and the target for the sustained feedback strategy. The horizontal slope for each curve near 0 degrees corresponds to the targeting deadband.

The second feedback strategy moves the tactors in a pulsed motion from center radially outward to the edge of a maximum tactor displacement radius circle, and back to center. Figure 6.2 depicts this pulsed motion by showing pulsing amplitudes. The angle of the motion is a function of the angle between the user and the target. This feedback strategy will be referred to as pulsed feedback. The cue consists of an outbound motion at maximum velocity (40-50 mm/s) and a return motion at the same velocity 0.4 s after the start of the outbound motion. This out and back cue repeats at 1 Hz. When the user's heading is within the target window, the tactor moves to and stops in the North (forward) position. Holding in the North direction indicates that the user is on target. The left and right sides of the controller are moved simultaneously and in the same manner.

The third feedback strategy is a single-axis sinusoidal tactor motion that varies in frequency and displacement amplitude as a function of the angle between the user's heading and the target, as seen in Figure 6.3. For the front- and back-tactor feedback

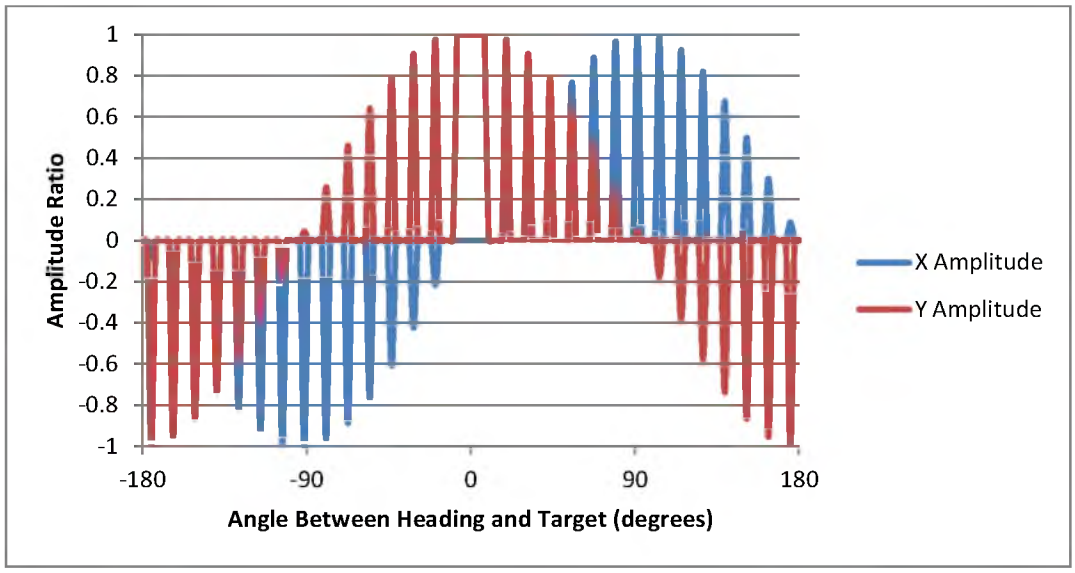


Figure 6.2: Plot of tactor position as a function of the angle between the user's heading and the target for the pulsed feedback strategy. The horizontal slope for each curve at 0 degrees corresponds to the tactor being held in the North (forward) position while the user is within the targeting deadband.

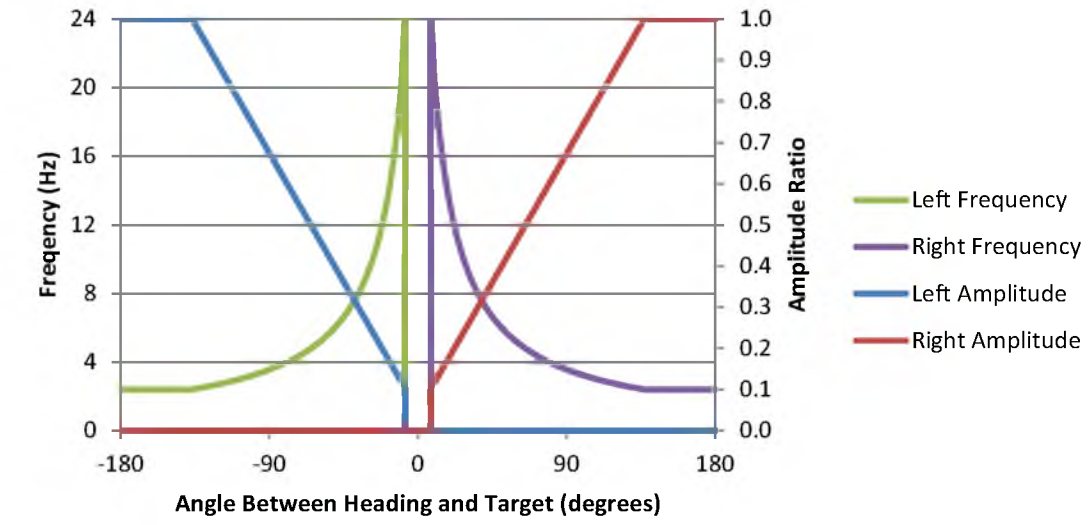


Figure 6.3: Plot of tactor frequency and amplitude as a function of the angle between the user's heading and the target for the sinusoidal feedback strategy. The drop in amplitude and frequency for each curve near 0 degrees corresponds to the targeting deadband.

locations, this feedback is presented on the X-axis of motion of either the right or left tactor. This feedback strategy will be referred to as "sinusoidal feedback".

Figure 6.3 shows how this feedback strategy varies with angle. As the angle decreases, the frequency increases and amplitude decreases. Amplitude is shown as a ratio of maximum displacement. This is done because the three feedback locations do not have the same maximum feedback displacement. When the user's heading is outside the target deadband window, the amplitude varies between 10% and 100% and the frequency varies between 2.4 and 23.9 Hz. When the user's heading is within the target window, the tactor rests in the center of its workspace. This lack of tactor motion indicates that the user is within the target window (i.e., is pointing at the target with sufficient accuracy to end the trial). Additionally, the feedback is applied to the left side of the controller when the target is counter-clockwise relative to the user's heading, and is applied to the right side of the controller when the target is clockwise relative to the user's heading. Only one of the sides moves at any given time.

6.1.2 Hardware and Setup

Hardware for the experiment consisted of a PC, a monitor, noise canceling headphones, a chair, and the front- and back-tactor controllers. The microcontroller inside of the front-tactor controller was programmed to use the tactors' full workspace (~5 mm diameter workspace when unloaded) to maximize the saliency of feedback. The microcontroller inside of the back-tactor controller was programmed to use a 4 mm diameter tactor workspace instead of the full workspace because the tactor would

sometimes pinch user's fingers between the tactor and the aperture if the tactor was moved too far.

Figure 6.4 shows the setup of the hardware used in the experiment. The monitor faces the participant and displays the virtual environment and occasional instructions. The noise canceling headphones play white noise and tones during the experiment.

6.1.3 Software Details

The experiment software for the PC was written in C#. The software is responsible for displaying instructions and a 3D virtual environment, recording participant responses, playing tones when participants successfully locate targets, and communicating with the game controllers.

Figure 6.5 depicts the flow of the experiment program. The program displays instructions and allows the participant to take a break before and after each round of 10 targets. All rounds consist of 10 targets in different directions. The practice program has the participant complete 2 rounds for each of the 3 feedback strategies, for a total of 60 targets. After the participant has located all of the targets in the initial practice session, the test program is started.

The test program tests the participants on 3 conditions the first day, and 4 conditions the second day. For each condition, participants are first required to complete 2 training rounds. After the participant completes the 2 training rounds, the program pauses and displays instructions for the test rounds. After the participant completes 10 test rounds, the program pauses and displays instructions for the next condition. This process repeats for the remaining test conditions.

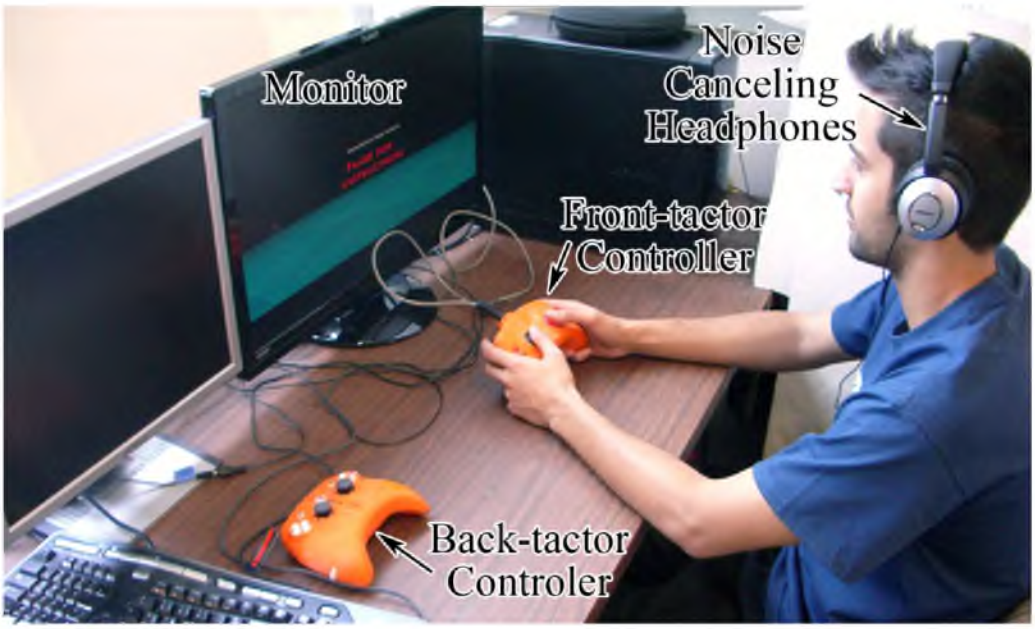


Figure 6.4: Experiment setup with participant

Before each round starts, the monitor displays instructions that say to press the trigger when ready to start, as shown in Figure 6.6. After the participant presses the right trigger, the round starts. The monitor displays an empty, but checkered 3D virtual environment. The controller provides feedback, which guides the participant to the target. The view of the virtual environment rotates when the participant moves the left thumb joystick to the left or right.

The rate of rotation varies between 0 and 92.8 °/s, and is linearly related to the X-position of the left thumb joystick. The maximum rate of rotation was selected in an attempt to try to minimize the total test length. If the maximum rate of rotation were set too high, it would be difficult to stop within the target window. If the maximum rate of rotation were set too low, finding targets would take longer. Once the participant rotates to the target angle by rotating to within ±7.5 degrees of the target and pressing the left

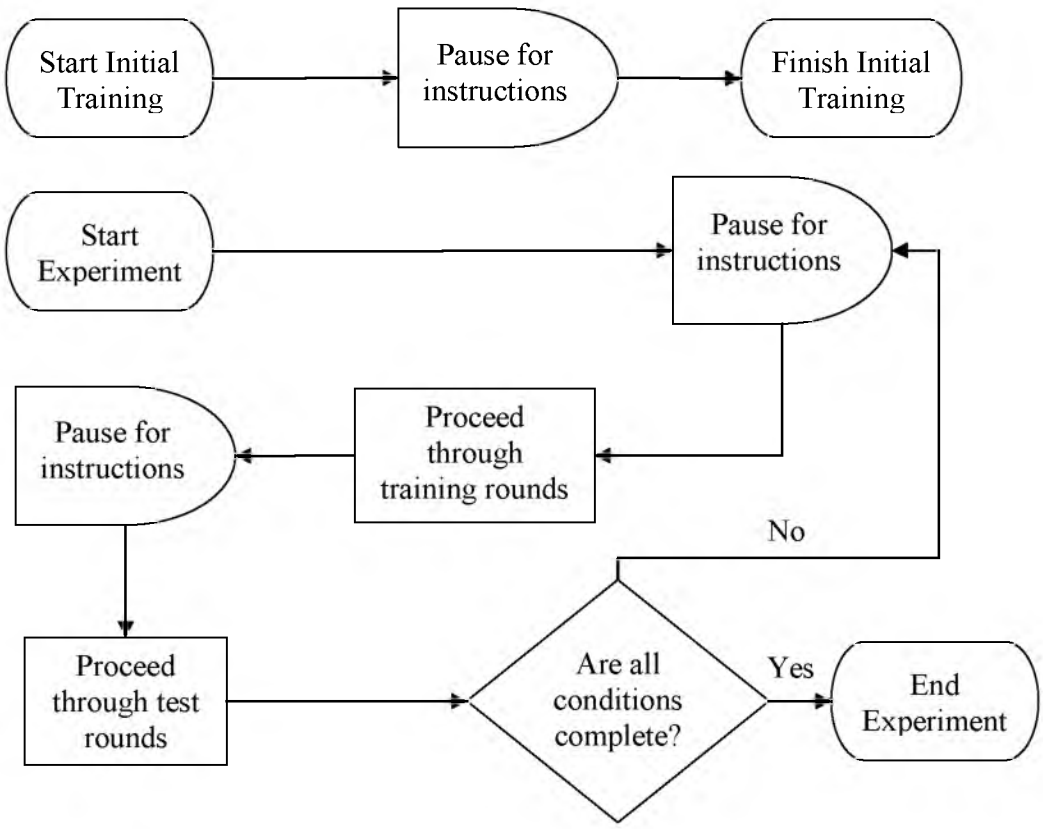
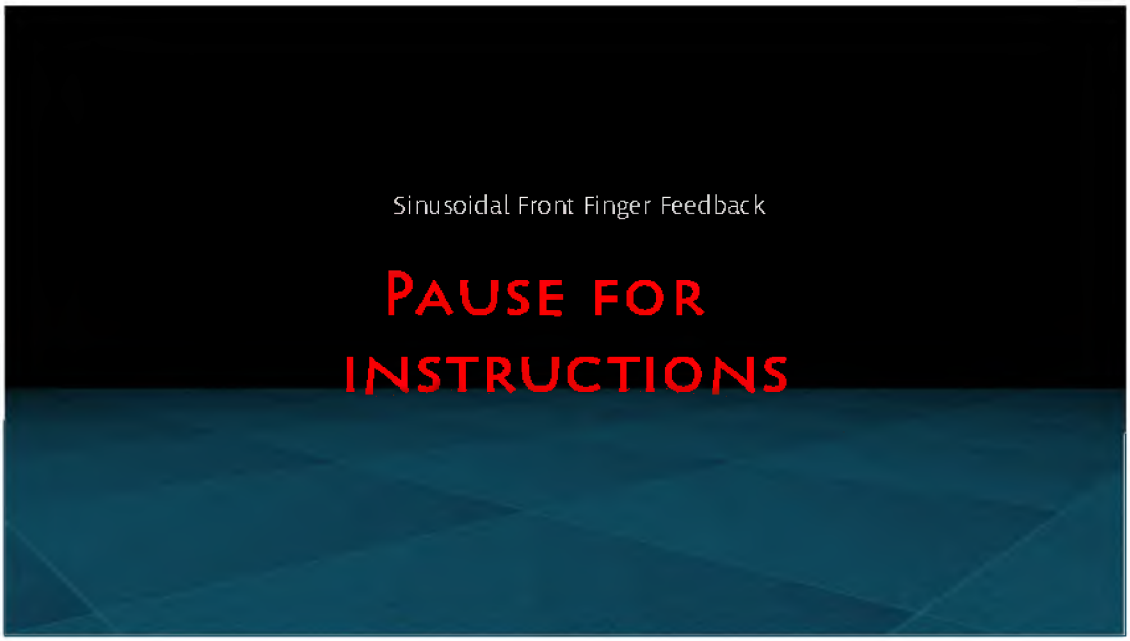
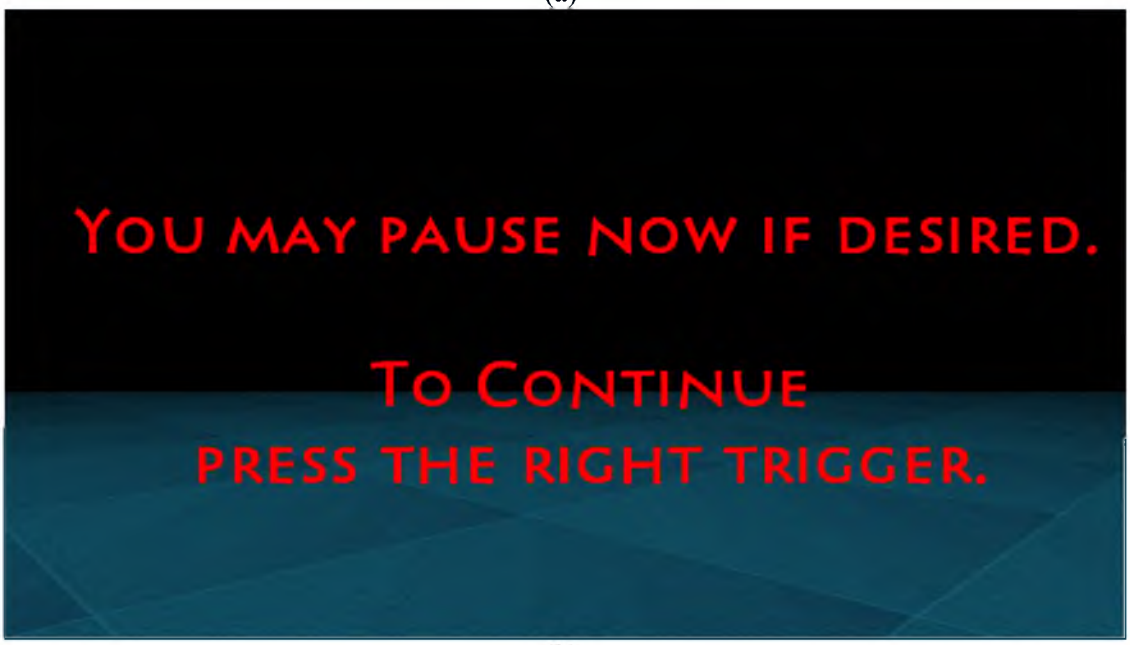


Figure 6.5: Flow diagram of experiment software



(a)



(b)

Figure 6.6: Examples of on screen visual feedback. (a) Instructions displayed at the start of a condition (b) Instructions displayed in between rounds

trigger, a tone plays through the headphones and the next target is presented. The next target's position is defined relative to the participant's current heading. After the participant has found the 10 targets, instructions are again displayed until the participant decides to continue with the test.

The participant is seated in a chair at the desk. The monitor and a game controller are on the desk in front of the participant. The test proctor provides the participant written experiment instructions, which explain the purpose of the experiment. The proctor then asks the participant if they have any questions after the participant finishes reading the paper. The proctor starts the experiment after answering any questions and reiterating key points of the instructions. The key points are:

- Three of the sections will be completed the first day, four on the second day
- Each section consists of a short practice session prior to a test session.
- Keep your middle fingers (or thumbs for the front-tactor controller) on the tactors while the test is in progress.
- Try to find the targets as quickly as possible
- Press the left trigger when you think you have found the target
- If you do not hear a tone play, continue looking for the target and try again
- There are 7 sections to the experiment.

The proctor hands the participant a game controller and verifies that the participant is holding it correctly. The participant should have their thumbs on the thumb joysticks, their index fingers on or near the triggers, their middle finger on the back of the controller (resting on the back-tactors when holding the back-tactor controller), and remaining fingers wrapped around the handles.

6.1.4 Proctor and Participant Interaction

During the practice rounds, the proctor explains how each feedback strategy works until the participant understands and is able to find the targets. The proctor may also change out which controller is being used between feedback conditions. The practice rounds, test rounds, and possible controller change are repeated for the different feedback conditions. The participant is asked to complete a survey after the experiment on the second test day.

6.2 Results and Discussion

The primary goal of this experiment was to measure the time it takes to rotate to and find a target for the 3 different feedback strategies, implemented with skin-stretch feedback at 3 different locations. In addition to time, some other notable measures include: target overshoot, sum of rotation, average velocity, number of instances of rotating the wrong direction, and number of instances of correcting wrong initial rotations (the latter several of which indicate misinterpreting the direction of cues). We will also look at the notable effects of target angle and feedback location.

6.2.1 Feedback Strategy

Feedback strategy is a strong factor in predicting the various measures of performance. Figure 6.7 shows mean completion times for the feedback conditions. A 2-way repeated measures ANOVA showed that at least one of the feedback conditions was significantly different than the others [$F(6,9752)=$, $p>0.001$], after accounting for absolute target-angle, participant, and condition/absolute target-angle interaction. A post-

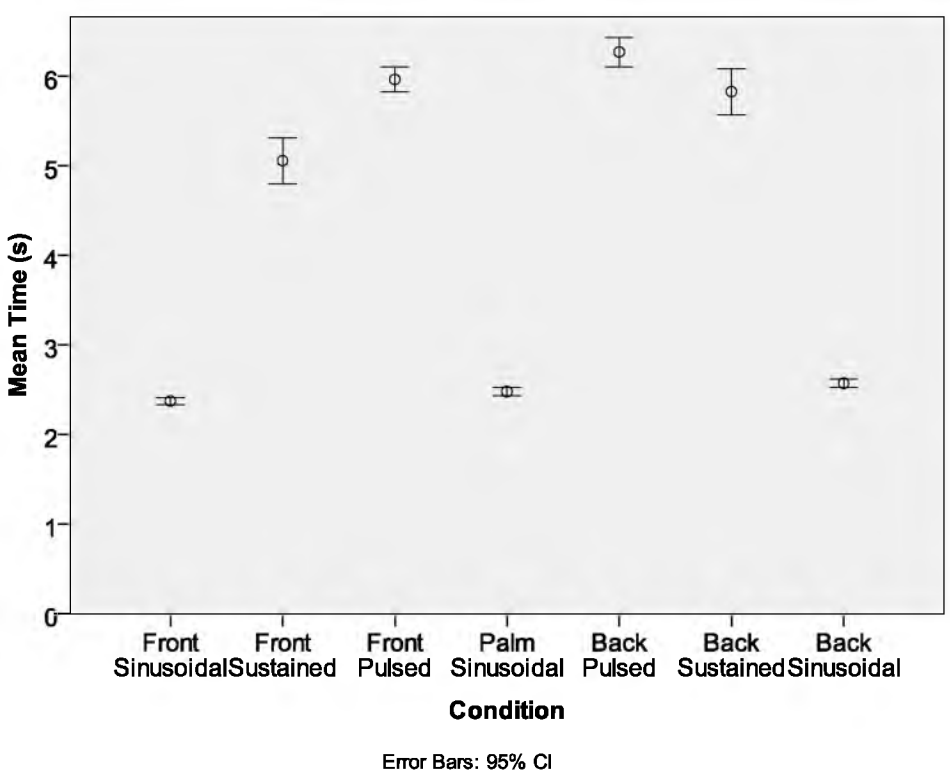


Figure 6.7: Plot of mean time to find each target for each feedback condition

hoc Tukey's test showed that the sinusoidal feedback conditions had significantly faster completion times than all other feedback conditions [$p<0.001$ for all]. It also showed that front pulsed feedback had significantly slower completion times than sustained feedback [$p<0.001$] and back pulsed feedback had significantly slower completion times than back sustained feedback [$p=0.001$]. Sinusoidal feedback mean completion time is nearly half that of sustained feedback. Note the symmetrical layout of the feedback conditions shown in Figure 6.7 and later related figures, with the sinusoidal, palm condition shown in the middle.

Figure 6.8 shows the mean rotational velocity for each feedback condition. A 2-way repeated measures ANOVA showed that at least one of the feedback conditions was significantly different than the others [$F(6,9752)=$, $p>0.001$], after accounting for absolute target-angle, participant, and feedback condition/absolute target-angle interaction. A post-hoc Tukey's test showed that the pulsed feedback conditions had significantly slower velocities than all other feedback conditions [$p<0.001$ for all]. This slow rotation is the reason that pulsed feedback has the slowest mean completion time. Participants likely rotate slower on the pulsed feedback because the feedback provides information at a much lower bandwidth. Updated direction information is only conveyed at 1 Hz, whereas sinusoidal and sustained feedback methods continuously provide feedback as the participant rotates (at 60 Hz). The high tactor velocity for the pulsed mode may have also made the cue directions more difficult to sense.

Figure 6.9 shows the mean total rotation for each feedback condition. A 2-way repeated measures ANOVA showed that at least one of the feedback conditions was

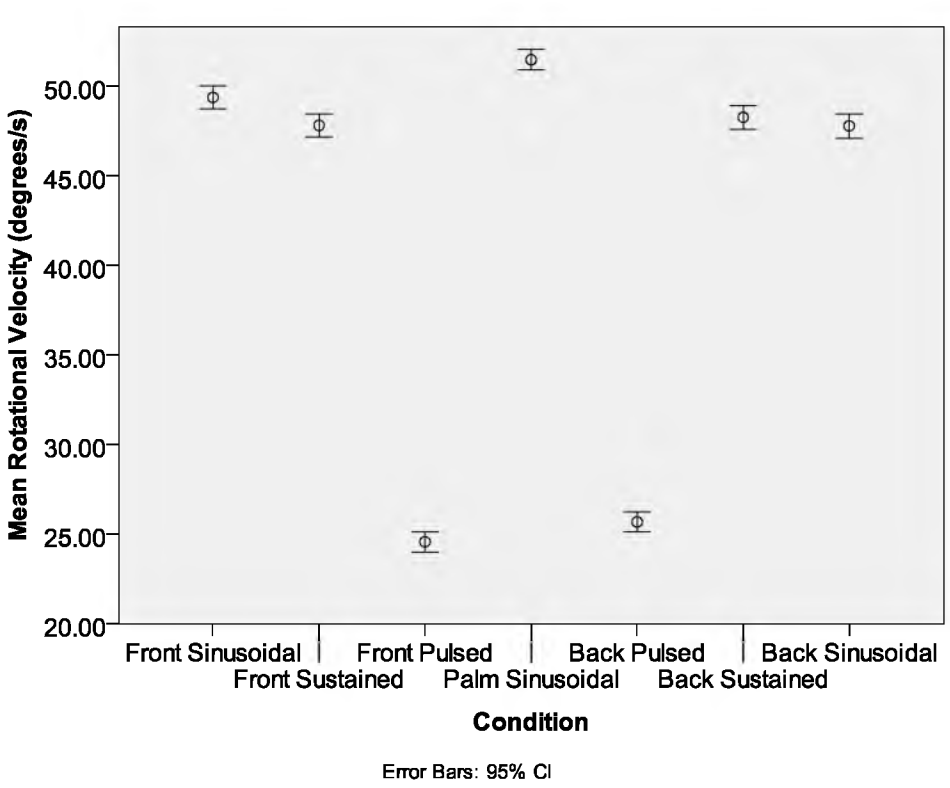


Figure 6.8: Plot of mean rotational velocity for each feedback condition.

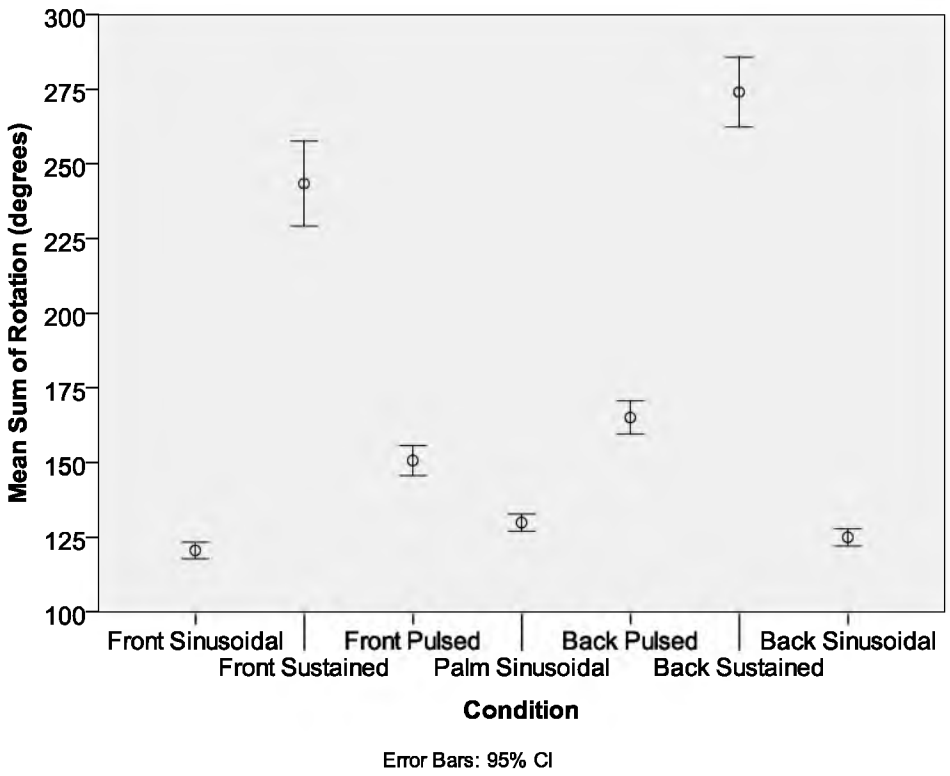


Figure 6.9: Plot of total rotation angle for each feedback condition.

significantly different than the others [F(6,9752)=, p>0.001], after accounting for absolute target-angle, participant, and condition/absolute target-angle interaction. A post-hoc Tukey's test showed that the sustained feedback conditions had significantly larger sums of rotation than all other feedback conditions [p<0.001 for all]. While sinusoidal and sustained feedback methods have similar rotational velocities, sinusoidal feedback has a lower completion time because participants more efficiently located the target.

Participants mentioned having difficulty sensing the 'on target cue' during the sustained feedback. The 'on target cue' for sustained feedback is only noticeable when the participant is rotating, and participants adopted a strategy of rotating back and forth over the target until they are able to sense the 'on target cue', hence why their total integrated rotation angle is so much higher in Figure 6.9. If the 'on target cue' for sustained feedback were more salient, performance would likely have been closer to that of the sinusoidal feedback, which could be incorporated in the future.

Figure 6.10 shows the mean target overshoot for each feedback condition. For this analysis, overshoot is defined as the angle by which the participant exceeded the minimum angle required to be on target. This overshoot considers only the overshoot from the last revolution of user motion. That is, if the participant passed the target and kept going any additional revolution(s), this would not have been taken into account in the reported values in Figure 6.10. However, it is unlikely that participants simply rotated in such a manner. A 2-way repeated measures ANOVA showed that at least one of the feedback conditions has significantly different target overshoot than the others [F(6,9752)=, p>0.001], after accounting for absolute target-angle, participant, and condition/absolute target-angle interaction. A post-hoc Tukey's test showed that the

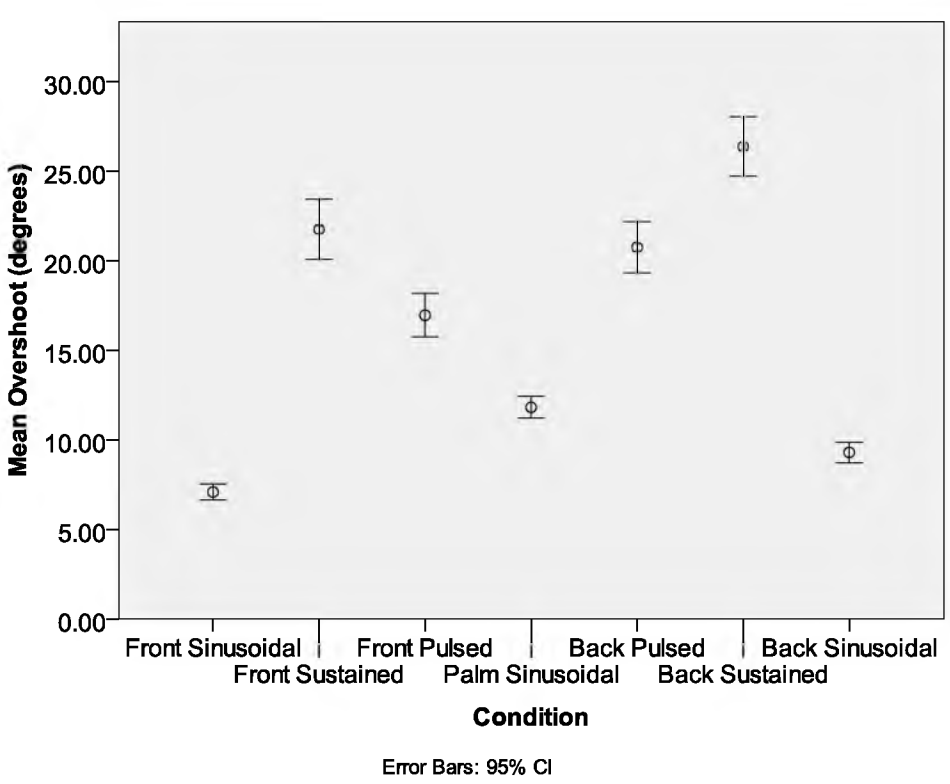


Figure 6.10: Plot of degrees of overshoot for each feedback condition.

sinusoidal feedback conditions had significantly smaller overshoot than all other feedback conditions [$p<0.001$ for all]. It also showed that front pulsed feedback had significantly less overshoot than front sustained feedback [$p<0.001$] and back pulsed feedback had significantly lower overshoot than back sustained feedback [$p=0.001$]. The amount of overshoot is an indicator of the saliency of the 'on target cues' and cue bandwidth for the feedback strategies. These relative differences in overshoot match participants' comments on the difficulty of sensing the 'on target cues' under sustained and pulsed feedback modes.

It is also interesting to look at if the participant is initially interpreting the direction cues correctly or whether they are just guessing and moving, and correcting after they

start moving. A participant is deemed to have made a wrong initial rotation if they rotate away from the target beyond their starting orientation. Figure 6.11 and Figure 6.12 illustrate how good each feedback strategy is at guiding participants in the optimal direction to the target. Sinusoidal and pulsed feedbacks have the lowest rate of incorrect initial rotations, while sustained feedback has over double the incidences of incorrect initial rotations. The higher rate of incorrect initial rotation for the sustained feedback is likely due to difficulty in determining the initial direction of the skin-stretch feedback.

The direction of the target indicated by sustained feedback is difficult to sense unless the participant is rotating. Often this means that participants guess which direction to rotate initially, and correct their input based on the relative motion of the tactor. This could be due to the velocity of the tactor motion (40-50 mm/s) being so high that participants had difficulty sensing the direction of the skin stretch.

Figure 6.12 shows that participants almost always correct an initial incorrect rotation input during sinusoidal feedback, but only correct an initial wrong rotation about 50% of the time during the sustained and pulsed feedbacks. This difference is an indicator that left/right rotation information is much more salient during the spatially separated sinusoidal feedback. Also, by the time participants realize that they rotated the wrong direction during the sustained and pulsed feedbacks, it sometimes takes less time to continue in the wrong direction than it would take to rotate back (e.g., if the target was at -170 degrees the will likely just continue in the CCW direction).

Figure 6.13 shows the mean rotation angle that participants rotate through before they correct an incorrect initial rotation for each feedback condition. A 2-way repeated measures ANOVA showed that at least one of the feedback conditions has significantly

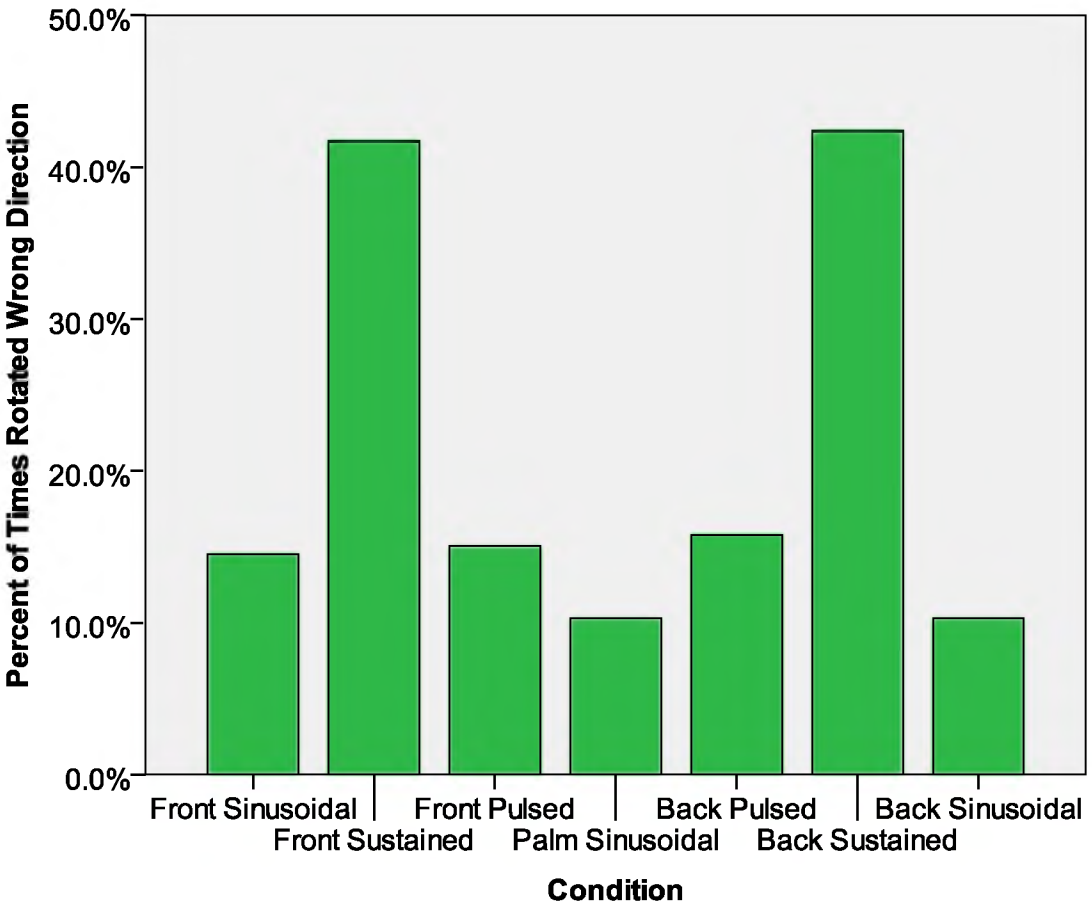


Figure 6.11: Plot of the percent of time that participants rotated the wrong direction for each feedback condition.

different target overshoot than the others [$F(6,9752)=$, $p>0.001$], after accounting for absolute target-angle, participant, and condition/absolute target-angle interaction . A post-hoc Tukey's test showed that the sinusoidal feedback conditions had significantly smaller overshoot than all other feedback conditions [$p<0.001$]. This is another indicator that left/right information supplied by the sinusoidal feedback is more salient than that of the other feedback strategies. This is probably due to the spatial separation of the feedback of the left/right information during sinusoidal feedback.

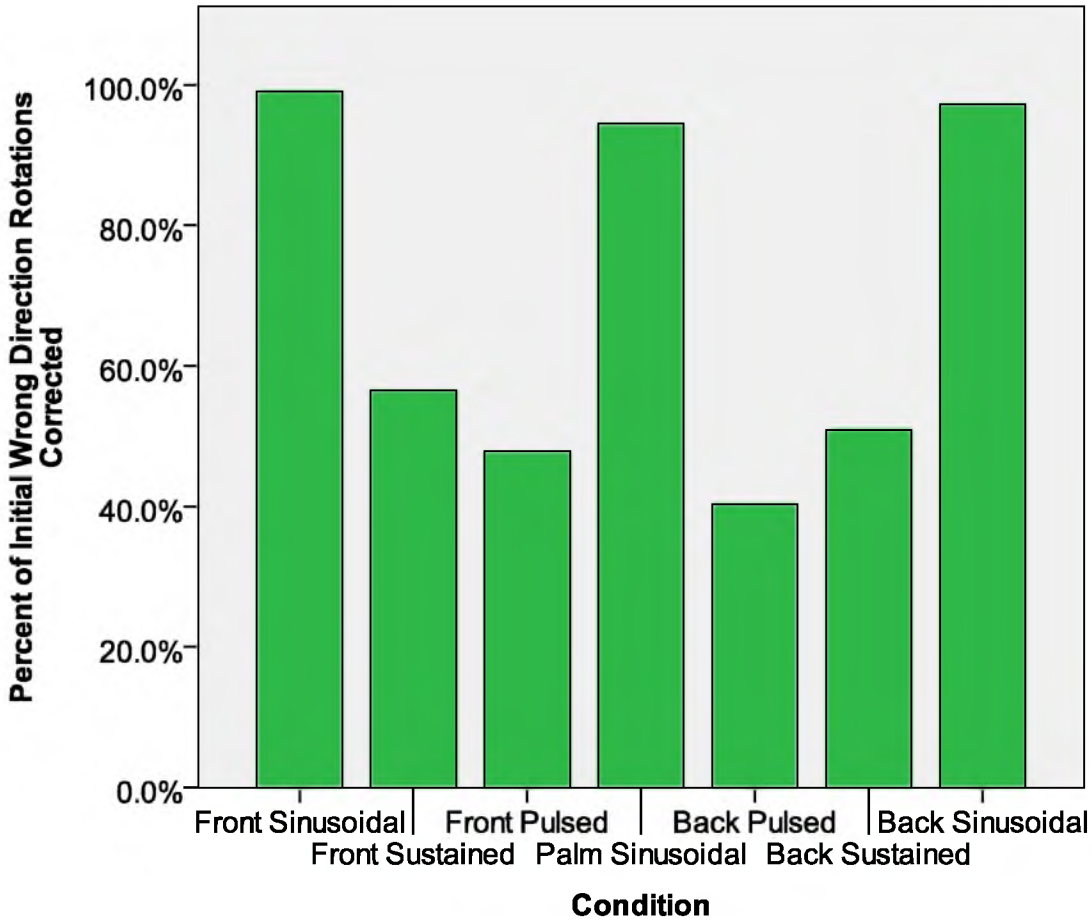


Figure 6.12: Plot of percent of times that participants corrected an incorrect initial rotation for each feedback condition.

A survey was conducted after the experiment which asked participants questions about their preferences regarding the different feedback conditions. Figure 6.14 is a plot of participants' mean responses to three types of questions. The questions asked participants how much they liked/disliked, their perceived performance, and ease of interpretation for each feedback condition. Participants chose from a preference scale from 1 to 5. A 1 corresponded to very much disliked, very poor performance, or very difficult to interpret. A 5 corresponded to very much liked, very good performance, or very ease to interpret. Scores of 3 were neutral responses. The results show that the

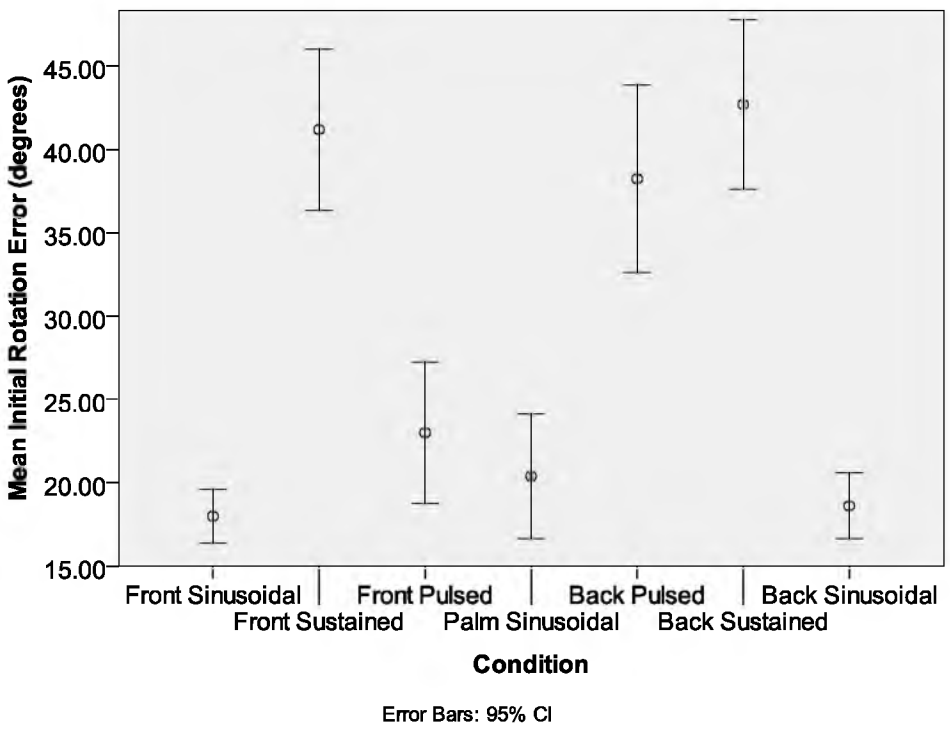


Figure 6.13: Plot of mean angle rotated in wrong direction before correcting direction for each feedback condition.

sinusoidal feedback strategies had the highest scores in likeability, ease of interpretation, and perceived performance. The pulsed feedback strategy, on the other hand, had the lowest scores in all three categories. These results match the performance results shown above.

6.2.1 Feedback Location

Feedback location was not as big of a factor in performance as feedback strategy, but still accounts for small differences in performance. Figure 6.15 shows the mean completion times for the sinusoidal feedback conditions and Figure 6.16 shows the mean completion times for the pulsed and sustained feedback conditions.

A 2-way repeated measures ANOVA showed that at least one of the feedback

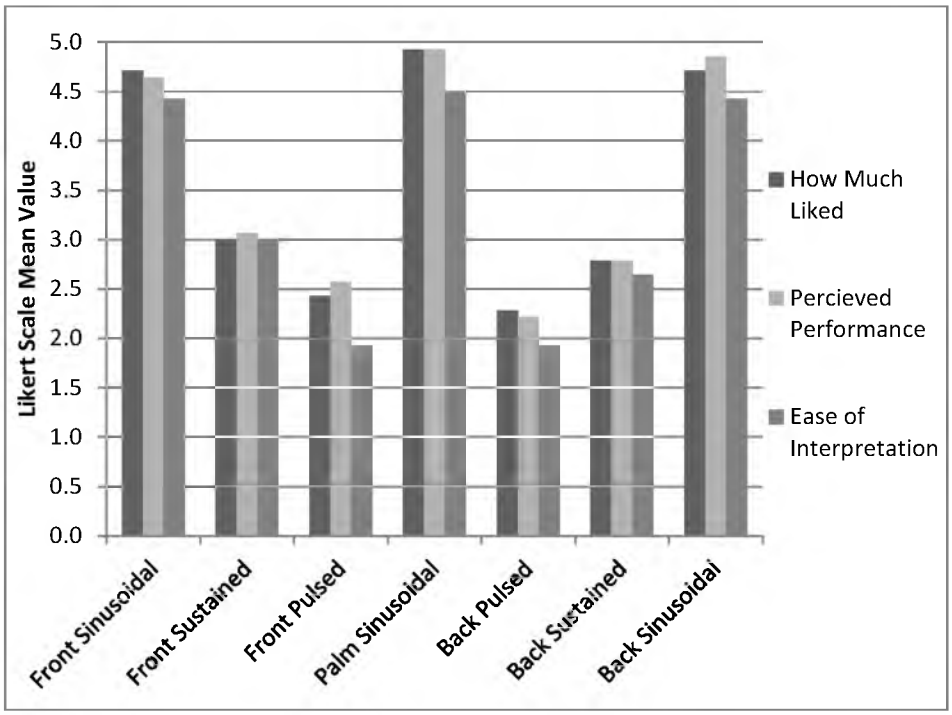


Figure 6.14: Plot of participants' opinions regarding feedback condition preference, perceived performance, and ease of interpretation

conditions was significantly different than the others [F(6,9752)=, p>0.001], after accounting for absolute target-angle, participant, and condition/absolute target-angle interaction. A post-hoc Tukey's test showed that the three sinusoidal feedback conditions do not have significantly different completion times between each other [p>0.544 for all]. It also showed that front pulsed feedback does not have significantly different completion times than back pulsed feedback [p=0.079], but that front sustained feedback has significantly faster completion times than back sustained feedback [p<0.001]. The slightly lower completion times for the front conditions could be due to the feedback and thumb joystick being co-located for that condition. Also, slight differences in ergonomics between the front and back tactor devices could contribute to this difference in completion time.

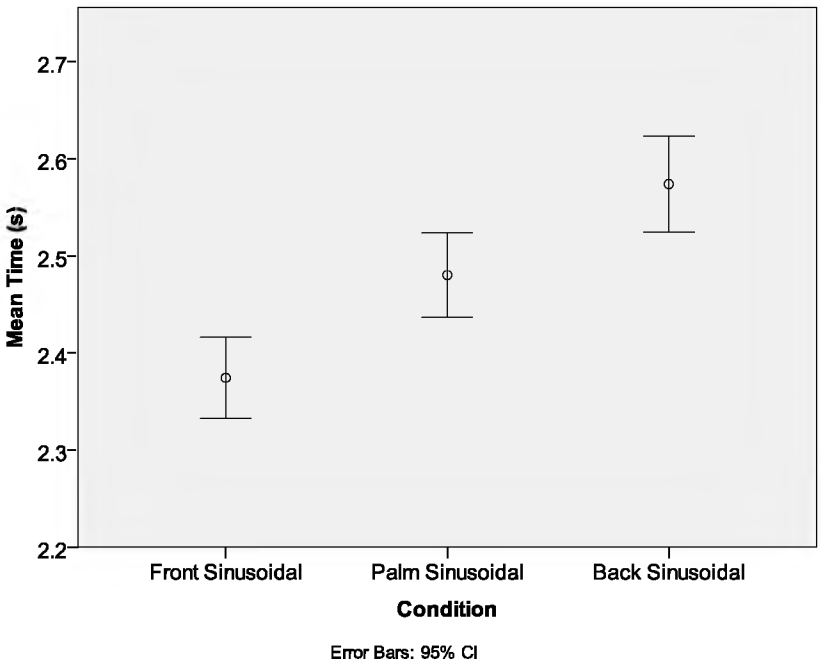


Figure 6.15: Plot of mean time for sinusoidal feedback at the three feedback locations.

Figure 6.17 shows that the front feedback location has lower overshoot than the back feedback location. An independent-samples t-test showed that the front feedback location has significantly lower overshoot than the back feedback location [$t(8398) = -5.287$, $p<0.001$]. Co-location of feedback and input on the front location supports this outcome as well (i.e., this is expected). By locating the input and feedback in the same place, attention may be focused more than when the feedback is in a different location than the input.

6.2.1 Target Angle

The initial angle of targets relative to the participant has several notable effects on performance that reinforce and further explain many of the observations above. Figure 6.18 shows that in general, as target angle increases, completion time increases. It is

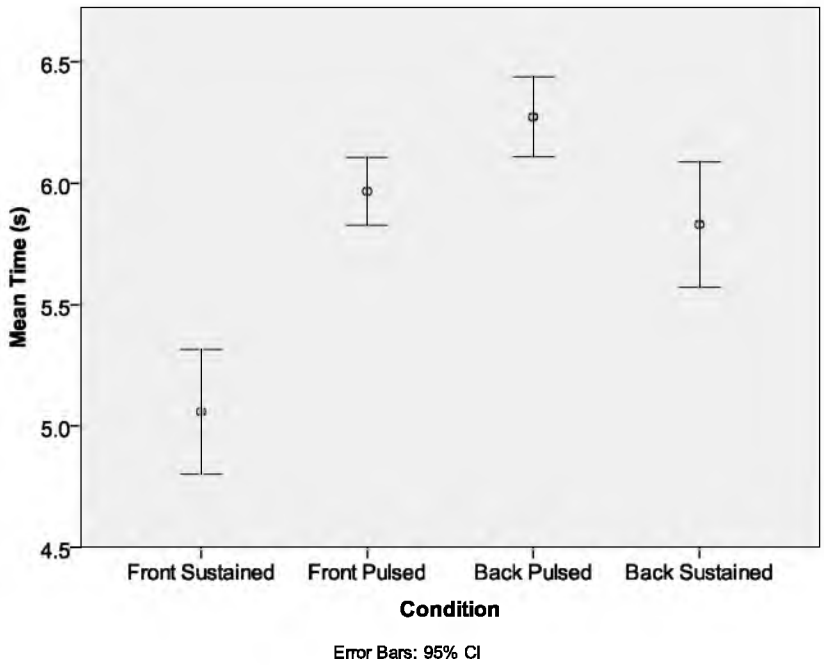


Figure 6.16: Plot of mean time for sustained and pulsed feedbacks at the front and back locations.

notable that for both sustained feedback conditions, target angle does not correlate with completion time [$F(1,1398)<0.728, p>0.394$]. This is likely due to difficulty in sensing the initial target direction of sustained feedback cues. This could result in participants rotating the wrong direction, as shown in Figure 6.11 or oscillating about the target window, as shown by the large total rotation in Figure 6.9. Occasionally rotating the wrong direction can negate any time benefit of a smaller initial target angle.

Figure 6.19 shows how overshoot changes with target angle for each feedback condition. Sinusoidal and sustained feedback methods have a decrease in overshoot as target angle increases, but pulsed feedback shows an opposite trend. The positive correlation between overshoot and target angle for the pulsed feedback is likely due to the low bandwidth of the feedback combined with the fact that participants tend to rotate faster when they start further away from the target.

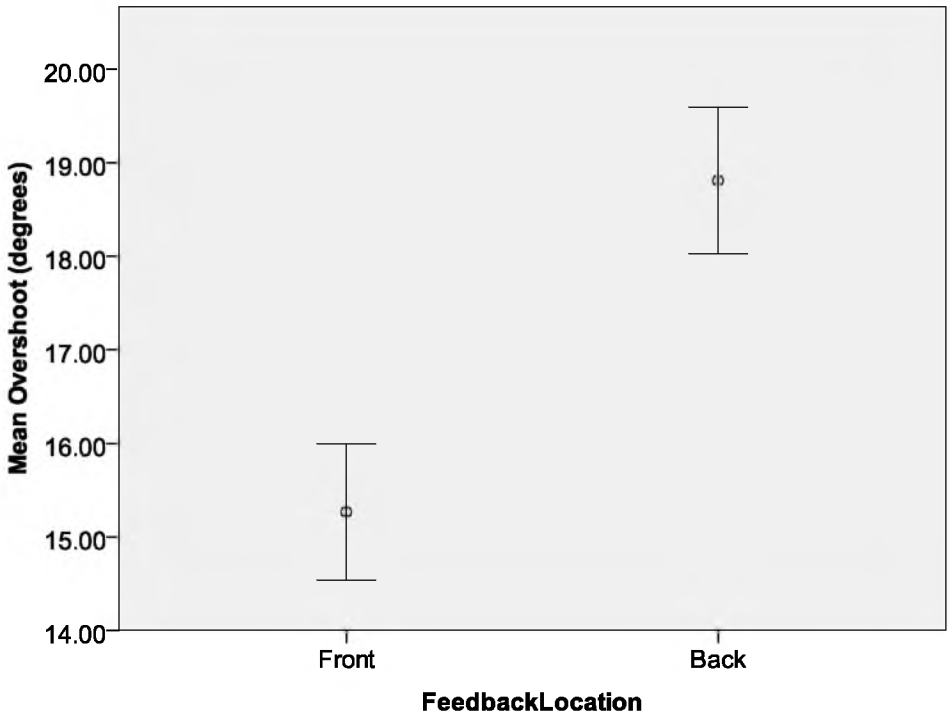


Figure 6.17: Plot of overshoot for each feedback location.

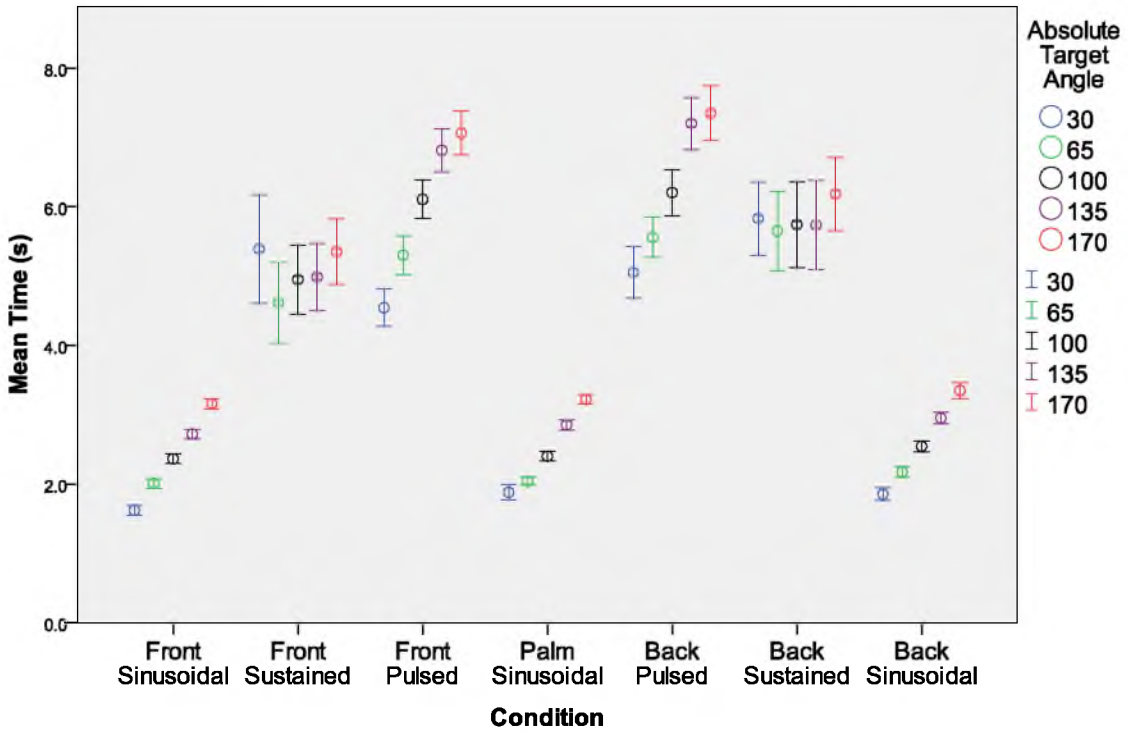


Figure 6.18: Plot of mean completion time for target angle and feedback condition.

The negative correlation between overshoot and target angle for sinusoidal and sustained feedback methods could be due to both feedback methods constantly varying as the participant approaches the target. This change in feedback as they approach the target may alert them that they are approaching the target. When participants start closer to the target, less of a change in feedback is sensed, which may reduce how effectively the participant perceives the approach of the target.

Figure 6.20 shows an increased number of incorrect direction rotations for the 30 and170 degree targets for the sustained and pulsed feedback conditions. This indicates that the left/right sidedness of the 30 and 170 degree targets, nearly straight forward and straight back, respectively, was particularly difficult to sense initially.

6.1 Summary

Sinusoidal feedback was by far the best feedback strategy of the three feedback strategies tested. Participants using this strategy found targets faster, rotated more quickly, rotated less overall (less wasted motion), overshot targets the least, and rarely rotated away from the target with their initial motions. These performance benefits are likely due to the feedback providing very salient direction and proximity information at a high level of bandwidth. Sidedness information provided by this feedback is very easy to interpret due to the spatial separation of the rotational direction information. Also, because this feedback strategy only requires one DOF to be implemented, simple devices can easily integrate this form of feedback.

While pulsed and sustained feedback conditions performed worse than sinusoidal feedback, useful information about each can be shown by comparing them to each other.

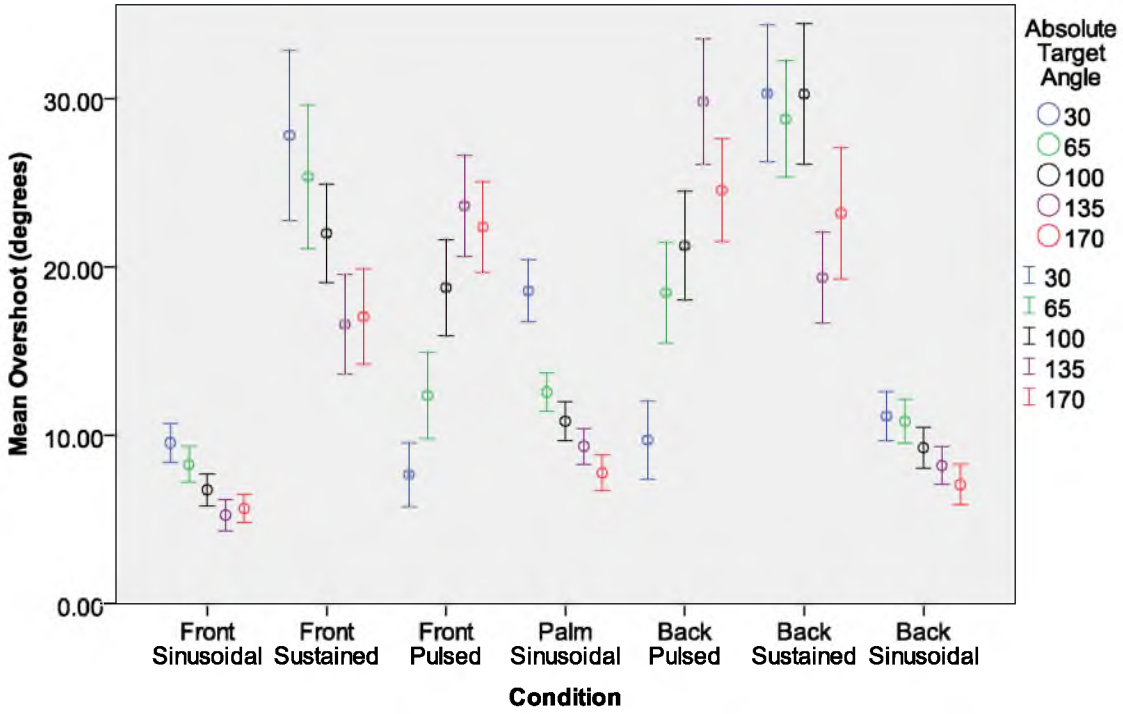


Figure 6.19: Plot of overshoot for each target angle and feedback condition.

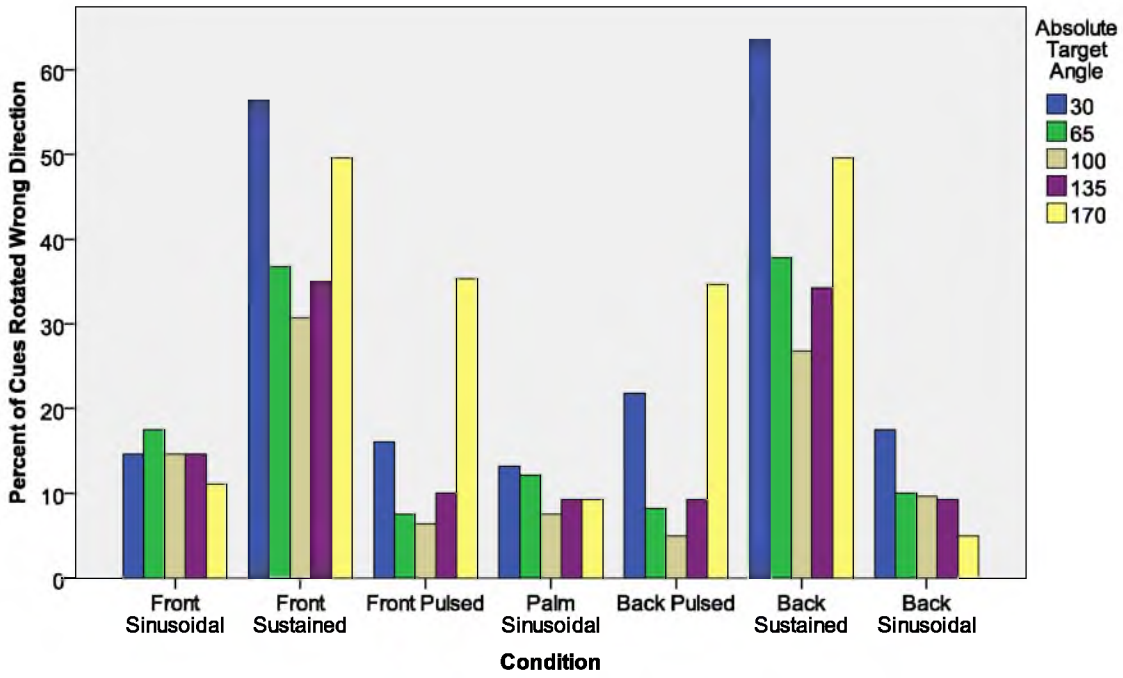


Figure 6.20: Plot of percent wrong initial direction rotations for each target angle and feedback condition.

Participants using sustained feedback had lower completion times and rotated faster than pulsed feedback. Participants using pulsed feedback had lower overshoot, rotated much less overall, and rotated away from the target less often. This indicates that the pulsed feedback was able to more efficiently guide participants to the target (no wasted/changing inputs from user), but did so slowly. This is not a surprise as the bandwidth of the pulsed feedback is rather limited by the pulse rate of 1Hz. Also, the large total rotation and high average rotation velocity for sustained feedback indicate that participants using sustained feedback spent a lot of time actively searching for the target. They rotated past the target multiple times before finding it, indicating that the "on target" cue provided by sustained feedback was difficult to sense. If the 'on target cue' for sustained feedback was modified to be easy to identify, the high average velocity of this feedback suggests it may become competitive with the sinusoidal feedback.

Feedback location was shown to also be a significant predictor of performance. Participants receiving feedback from the front-feedback (thumb joystick) location found targets faster with less overshoot than when receiving feedback from the back-feedback location. This performance difference could be due to the co-location of feedback and thumb joystick input on the front-tactor controller. Differences in ergonomics between the front- and back-tactor controllers could also contribute to this difference.

7 CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

This thesis presents the designs of two different game controllers with integrated skin-stretch feedback. One of the controllers packages a 2-axis skin-stretch mechanism inside of a thumb joystick that allows users to receive 2-axis skin-stretch information on their thumb pads while users simultaneously use their thumbs to move the thumb joysticks. The front-tactor mechanism is made compact by using two micro servos to actuate a flexure stage. The flexure stage is mounted to a tactor that can be accurately moved within a 5.1 mm diameter workspace at up to 42 mm/s. This game controller is also equipped with eight buttons, two on either side of the thumb joysticks, two triggers, and two bumpers, which allows the controller to flexibly interact with virtual environments and games.

The second controller uses the same micro servos used in the front-tactor controller to provide skin stretch to the user's middle fingers on the back side of the controller, behind the thumb joysticks located on the front the controller. The back-tactor mechanism is a flat design that uses servos to drive a sliding plate via spring steel wires. This design is flat enough that it fits in the back half of the controller. This allows the front half of the controller to house a large PCB to which all components except the actuators are mounted. Additional skin-stretch feedback actuators are located in the handles of the back-tactor controller and provide 1-axis skin-stretch feedback to the user's palms. These

palm tactors are each driven by a single servo that directly attaches to and drives a wheel that extends beyond the handle of the controller to the user's palm.

The two skin-stretch controllers were used to provide skin stretch to participants for the purpose of an 8-direction identification task. Results of the experiment showed that participants using the front- and back-tactor controllers have significant, but small differences in identification accuracy. This means that both controllers are approximately equally capable of providing directional skin-stretch information.

We also tested how well the game controllers could guide participants to heading to face toward a randomized target angle. Participants rotated within a virtual environment, using the skin-stretch feedback to guide them to the target's heading. Three different feedback strategies were tested that provide direction information in different manners. Sinusoidal feedback moved the tactors in a sinusoidal motion that increased in frequency and decreased in amplitude as participants got closer to the target direction. Only one tactor, either on the left or right side of the controller, would move at time. The side that moved indicated the direction participants should rotate toward to most efficiently reach the target. The other two feedback strategies are called sustained and pulsed feedback. These feedback strategies move the tactor in the direction of the target relative to the participant's current heading. The pulsed feedback pulses in the direction of the target, and the sustained feedback stays in an outbound position and follows the target around the edge of the tactor workspace.

The results of the experiment showed that the feedback strategy called sinusoidal feedback guided participants to the targets the fastest. This feedback also resulted in the least amount of overshoot and total rotation, fastest average rotational velocity, and the

fewest incidences of rotating away from the target. The pulsed feedback guided participants to the target the slowest, likely due to the low bandwidth of the feedback. The sustained feedback guided participants to the targets slightly faster than pulsed feedback. When participants were guided by the sustained feedback, they would quickly rotate to the target, but would take most of their time trying to locate the exact location of the target. If the sustained feedback were modified so that it becomes more apparent when participants are facing the target, the sustained feedback may be able to guide participants to the target as efficiently as the sinusoidal feedback mode.

7.2 Contributions

The work described in this thesis makes four contributions related to the fields of haptic perception and mechatronic design.

1. The first contribution is the development of a skin-stretch game controller that provides 2-axis skin stretch to a user's thumb pads centered within the thumb joysticks. This controller (or earlier prototypes developed as part of this thesis) has already shown that it is a valuable resource by its use for tactile feedback in the studies by Guinan et al. [5] [7] and in the two experiments detailed in this thesis. Also, future versions of skin-stretch game controllers will be able to build upon this design.
2. The second contribution is the development of a skin-stretch game controller that provides 2-axis skin stretch to a user's middle fingers on the back of the controller and 1-axis skin stretch to the user's palms at the handles of the controller. This controller has been used as a feedback device in the two experiments detailed in

this thesis. Also, future versions of skin-stretch games controllers will be able to build upon this design.

3. The third contribution is skin-stretch perception data that were collected from an 8-direction skin-stretch identification experiment comparing the relative effectiveness of the front- and back-tactor controllers at communicating direction information. Participants had to identify the direction of skin-stretch cue rendered via one of the 6 hardware conditions. Results from the experiment primarily showed that there were no significant differences in participant identification accuracy between the 6 hardware conditions. These results allow the designer to use either the front- or back-controller design with the confidence that user performance will be comparable, within the small variations seen between these designs.
4. The fourth contribution is data about the effectiveness of skin-stretch feedback at guiding participants rotationally to a randomized target angle. Participants were guided to the target angles in a virtual environment via 3 feedback strategies. Results from the data showed that a feedback strategy call "sinusoidal feedback" had the best performance in a wide range of metrics. This is particularly interesting to anyone who would like to guide participants along a single axis toward a target or position.

7.3 Future Work

There are several areas in which future work can be done to improve upon these skin-stretch game controllers and studies to be done to better understand participants'

perception of skin stretch and work to determine what scenarios and types of information can be effectively conveyed with skin stretch.

Future work improving the front-tactor controller design should attempt to modify the design so that as many of the components in the controller as possible are mounted to a single PCB (similar to the PCB in the back-tactor controller). This would significantly reduce the number of wires routed inside of the controller and reduce the need for the game-controller shell to house the components. This would make the disassembly and maintenance to the controller much easier and quicker to complete. Also, moving more components to the PCB will also make the controller much easier to manufacture, as a single PCB can be populated and set into the controller shell instead of placing multiple components into the game-controller housing and routing wires to the PCB.

The shape of the front-tactor controller can be improved to make it more ergonomic. The current version of the controller is somewhat ergonomic, but is not as ergonomic as an XBOX controller, for example. Increasing the comfort and ease of use of the controller to the standards of common modern game controllers would allow users to more comfortable use the front-tactor controller for longer periods of time, and would likely also reduce their response times to game events.

The back-tactor controller also has room for ergonomic improvements. The location of the tactors on the back-side of the controller is not in the ideal location for most users, and is likely reducing their ability to properly feel the skin-stretch cues. The ideal location could be determined by making a flexible model of a game controller and have participants modify the location and orientation of the back tactor apertures until they are

satisfied with the location. A new design could then be made that attempts to satisfy the location and orientation preferences of the participants.

Modifications of the sustained feedback from the second experiment could significantly improve its usefulness as a targeting cue. In the second experiment, the sustained cue simply ceased moving when participants were on target. This 'on target' cue was too subtle, and could be improved by additionally providing a pulse or detent sensation using the shear tactors or using the vibrotactors.

Further testing should be done to fully understand the perceptual biases inherent in these controllers, so that modifications to the angle at which participants interact with the tactors can be made in an attempt to minimize the difference between the feedback that is rendered and the feedback that is perceived. Modifications to the shape of the tactile workspace (perhaps nonsymmetric scaling) could also be made to try and make the skin stretch perceived as close as possible to the skin stretch that is desired to be conveyed.

Experiments could also be made that attempt to maximize various metrics of performance in game scenarios by using a combination of visual, audio, and skin-stretch information. These controllers have more potential to be used in more complex gaming tasks than has been taken advantage of in previous experiments. By testing skin-stretch feedback along with audio and visual feedback in real game situations, we can determine ways in which skin-stretch feedback can improve users' performance beyond their performance without skin-stretch feedback. Testing could also compare the effectiveness of skin-stretch, audio, and visual feedback at providing information in various scenarios.

The effects of combining vibrotactile and skin-stretch feedback should also be explored. Vibrotactile feedback is currently the standard form of haptic feedback in game

controllers, and is not likely to be displaced by the introduction of skin-stretch feedback into games controllers. Rather, both feedback types will likely be used in the same controller, sometimes at the same time. In order to determine how the use of skin-stretch feedback at the same time as vibrotactile feedback, tests should be performed that test for masking effects, priming effects, and other possible interesting interactions. Testing could also compare the effectiveness of the skin-stretch and vibrotactile feedback at providing information in various scenarios.

REFERENCES

[1] B. Gleeson, S. Horschel, and W. Provancher, "Perception of direction for applied tangential skin displacement: Effects of speed, displacement, and repetition," *IEEE Trans. Haptics*, vol. 3, no. 3, pp. 177-188, June-Sept., 2010.

[2] B. Gleeson and W. Provancher. "Mental rotation of directional tactile stimuli," in *2012 IEEE Haptics Symp.*, Vancouver, pp. 171-176.

[3] M. Montandon, and W. Provancher, "A smart phone peripheral with bi-manual skin stretch haptic feedback and user input," in *2013 IEEE Int. Conf. Consumer Electronics*, Las Vegas, NV, pp. 45-46.

[4] A. Guinan, R. Koslover, N. Caswell, and W. Provancher, "Bi-manual skin stretch feedback embedded within a game controller," in *2012 IEEE Haptics Symp.*, Vancouver, pp. 255-260.

[5] A. Guinan, N. Caswell, F. Drews, and W. Provancher, "A video game controller with skin stretch haptic feedback," in *2013 IEEE Int. Conf. Consumer Electronics*, Las Vegas, NV, pp. 456-457.

[6] A. Guinan, N. Hornbaker, M. Montandon, A. Doxon, and W. Provancher, "Back-to-back skin stretch feedback for communicating five degree-of-freedom direction cues," in *Proc. 2013 World Haptics Conf.*, Daejeon, p. 6.

[7] A. Guinan, M. Montandon, N. Caswell, and W. Provancher, "Skin stretch feedback for gaming environments," in *2012 IEEE Int. Workshop Haptic Audio Visual Environments Games*, Munich, pp. 101-106.

[8] R. Koslover, B. Gleeson, J. de Bever, and W. Provancher, "Mobile navigation using haptic, audio, and visual direction cues with a handheld test platform," in *IEEE Trans. Haptics*, vol. 5, no. 1, pp. 33-38, Jan.-Mar., 2012.

[9] Z. F. Quek, S. B. Schorr, I. Nisky, A. M. Okamura, and W. R. Provancher, "Sensory Augmentation of Stiffness using Fingerpad Skin Stretch," in *Proc. 2013 World Haptics Conf.*, Daejeon, pp. 467-472.

[10] N. Caswell, R. Yardley, M. Montandon, and W. Provancher, "Design of a forearm-mounted directional skin stretch device," in *2012 IEEE Haptics Symp.*, Vancouver, pp. 365-370.

[11] K. Bark, J. Wheeler, G. Lee, J. Savall, and M. Cutkosky, "A wearable skin stretch device for haptic feedback," in *Proc. World Haptics Conf. 2009*, Salt Lake City, UT, pp. 464-469.

[12] J. Wheeler, K. Bark, J. Savall, and M. Cutkosky, "Investigation of rotational skin stretch for proprioceptive feedback with application to myoelectric systems," in *IEEE Trans. Neural Systems Rehabilitation Engineering*, vol. 18, no. 1, pp. 58-66, Feb., 2010.

[13] K. Minamizawa, S. Fukamachi, H. Kajimoto, N. Kawakami, and S. Tachi, "Wearable haptic display to present virtual mass sensation," in *ACM SIGGRAPH 2007 Sketches*, San Diego, CA, p. 43.

[14] J. Biggs and M. Srinivasan, "Tangential versus normal displacements of skin: relative effectiveness for producing tactile sensations," in *2002 Symp. Haptic Interfaces Virtual Environment Teleoperator Systems*, Orlando, FL, pp. 121-128.

[15] N. Medeiros-Ward, J. Cooper, A. Doxon, D. Strayer, and W. Provancher, "Bypassing the bottleneck: The advantage of fingertip shear feedback for navigational cues," in *Proc. Human Factors Ergonomics Society Annu. Meeting*, vol. 54, no. 24, pp. 2042-2047, Sept., 2010.

[16] M. Vitello, M. Ernst, and M. Fritschi, "An instance of tactile suppression: Active exploration impairs tactile sensitivity for the direction of lateral movement," in *Proc. EuroHaptics*, France, pp. 351-355, 2006.

[17] B. Gleeson, S. Horschel, and W. Provancher, "Communication of direction through lateral skin stretch at the fingertip," in *Third Joint EuroHaptics Conf. Symp. Haptic Interfaces Virtual Environment Teleoperator Systems*, Salt Lake City, pp. 172-177, 2009.

[18] E. Gentaz, G. Baud-Bovy, and M. Luyat, "The haptic perception of spatial orientations," in *Experimental Brain Research*, vol. 187, no. 3, pp. 331-348, May, 2008.

[19] N. A. Macmillan and C. D. Creelman, "Basic detection theory and one-interval designs," in *Detection Theory: A User's Guide*, 2nd ed. New Jersey: Psychology Press, 2004, p. 7