University of Glasgow | School of Computing Science

Honours Individual Project Dissertation

Modelling Mouse Movements With Machine Learning

Anith Manu Ravindran

April 6, 2020

Abstract

This paper primarily documents the design of a comprehensive test suite comprising of a diverse set of representative mouse pointing tasks, and the evaluation of data gathered from conducting participant trials with these tests. By doing this, we are trying to uncover information about different types of interaction behaviour and motion dynamics with a mouse which can be useful for improving the accuracy of machine learning models in endpoint and motion prediction tasks.

Acknowledgements

I would like to thank Prof. Roderick Murray-Smith for his continued support and guidance throughout the year. Additionally, I would like to thank all the participants who took part in this project for their time and patience.

Education Use Consent

I hereby grant my permission for this project to be stored, distributed and shown to other University of Glasgow students and staff for educational purposes. **Please note that you are under no obligation to sign this declaration, but doing so would help future students.**

Signature: Anith Manu Ravindran Date: 4 April 2020

Contents

1 Introduction

1.1 Overview

In this paper, we will initially explore various approaches that have been used to model aimed mouse movements to increase the efficiency of mouse pointing tasks. Primarily, this is done to improve user experience by predicting the endpoint of an aimed movement in advance of its delivery by a mouse-click. We will also briefly explore basic, yet fundamental interaction models and paradigms in Human-Computer Interaction (HCI) such as Fitts' Law and the Steering Law. Apart from this, an important goal of the project is to gain an insight into the user interaction behaviour and motion dynamics of aimed mouse movements. This is done by first creating a diverse test suite of representative pointing tasks, followed by data collection with participants. The collected data will then be analyzed using various visualization techniques with which we will examine how a user interacts with the mouse in some scenarios and explore the dynamics of unfolding mouse movements in others. Finally, we will conclude by identifying prospective areas of further research using the findings from the collected data.

1.2 Aims

The aim of this project is to first gain an understanding of the fundamental models of human movement that will lay the groundwork for building the set of pointing tasks. We will then examine research that has been conducted with respect to mouse endpoint and motion prediction. Here we will also gain a perspective into the dynamic and continuous nature of mouse motion. Having understood the implications of all this in pointing tasks, we then layout the design and implementation of a test suite using Pygame that meet the following criteria:

- Produce data that follow the findings from established predictive models such as Fitts' and Steering laws.
- Produce data that help us understand the dynamics of mouse movements by examining aspects of motion such as movement time, pointer position, velocity, and acceleration.
- Obtain useful and reliable information about mouse interaction that can be used to improve machine learning models.
- Has parameterizable tests that can be modified to be used in future research of pointing tasks.

1.3 Motivation

Mouse pointing is the dominant interaction technique with computers (Johnson 1993). It is arguably the most simplest and well studied case of motion in the field of HCI. Studies show that mouse pointing comprises up to 65% of our desktop computer usage and that our mouse usage is 3-5 times that of our keyboard usage (Chang et al. 2007).

Machine learning models aim to make mouse interaction more convenient by performing tasks like target and endpoint prediction. By studying these models, we can gain an understanding into the basic metrics and techniques necessary to implement and improve such models. As we will see later, some machine learning models will implement prediction based on learning some overall summary statistic like peak velocity, movement time or distance. These models abstract the dynamics of the motion. This has seen success in a lot of cases with models being able to perform spacial and endpoint predictions, and through improvements in the underlying mathematical models, the predictions have gotten successively more accurate.

However, nowadays interaction with computers are getting more dynamic and in addition to the target (or endpoint), we have to start giving importance to the pointing process itself. And so, we have to think of interaction as more than a discrete task. As we bring interaction out of the computer into the real world, we have to design for and understand the dynamics of the interaction by taking into account the laws of physics and behaviour of the users. Müller et al. (2017) studies the dynamics of one-dimensional mouse movements from a control theoretic perspective. By creating a more expansive and varied test suite, this project will also try to also explore some aspects of mouse motion dynamics. Equally important, we will also try to uncover user behaviour in specific scenarios by means of this extensive test suite. By doing this, we hope to gain valuable information that could have the potential to improve the accuracy of existing machine learning models in endpoint and prediction tasks.

2 Background

2.1 Models of Human Movement

Before beginning to implement the test suite, it was important to understand the well established predictive models of human movement. These models have a big effect on pointing movements and interface elements, and subsequently, on the design of the tests for this project.

2.1.1 Fitts' Law

Fitts' Law (Fitts 1954; Mackenzie and Buxton 1992) is one of the cornerstones of user interface research. It describes the time to move to a target area with a rapid, aimed movement. The law states that the amount of time required for a person to move a pointer to a target area is a function of the distance to the target divided by the size of the target. Given the amplitude A of the motion (i.e. the distance to reach the target), and the width W of the target measured along the axis of motion, the average movement time MT required is given by the formula:

$$MT = a + b \log_2 \left(\frac{A}{W} + K \right) \tag{2.1}$$

a and b are empirically determined regression coefficients. The log term is the index of difficulty (ID), which is in bits (as the base is 2). The movement time of a task is linearly correlated with its ID. From this, we can see that a pointing task will get harder the farther away the target is and the smaller it is. This fits well with the common perception of such tasks. The test suite that will be reviewed later, has implemented interactive tests that will illustrate Fitts' law through controlled experiments.

2.1.2 Steering Law

Sometimes, when we use interface elements like cascading menus, scroll bars or video-playback scrubber, we may overstep the border assigned to these elements, which can result in some mild consequences as a result of this human error. These consequences can hinder the user experience and delay user intent. It is not the fault of the individual user, but is a fact that such interface elements are fundamentally tricky. The Steering Law provides an explanation for the difficult nature of such path-steering tasks.

The Accot-Zhai Steering Law (Accot and Zhai 1997) (known simply as the Steering Law) is a corollary of Fitts's Law. It is a predictive model of movement that predicts the time necessary to steer a pointer (such as a mouse cursor) through a bounded tunnel. According to the Steering Law, a tunnel is any user-interface control that requires the user to move the cursor along a path that has borders (such as menu, scroll bar or slider). Given the length of the tunnel A and width W, the overall movement time T to move through the tunnel is given by the formula:

$$T = a + b \left(\frac{A}{W} \right) \tag{2.2}$$

a and b are empirical constants. The term (A \W) is the Steering's index of difficulty. We can infer from this formula that it takes more time to move a pointer through a narrow tunnel.

Our arms tend to follow a natural curve when in motion. The elbow and wrist, which enable the movement of the hand, describe an arc, not a line (Laubheimer 2019). Moving in a precisely straight line goes against this tendency which makes it difficult to move a pointer through a tunnel.

The steering law predicts that when we have to move precisely through a complex menu where each individual tunnel is narrow, it takes us more time. And if we try to move more quickly, we will be less accurate and cross the borders of the tunnel.

2.2 Previous Approaches to Mouse Movement Prediction

Previous work has seen the implementation of endpoint prediction using regression-based extrapolation and other kinematics-based approaches. Each of these methods has benefits and shortcomings, which we describe in this section. But perhaps the most important part of this section is the review of control theory (explained in Section 2.2.3). It is based on this that we will analyze some of the results from the data collected using the test suite designed later on.

2.2.1 Regression Based

Asano et al. (2005) implemented the Delphian Desktop, which is a mechanism for online spatial prediction of cursor movements in a Windows homepage. It is based on the relationship between the linear peak velocity of a movement and its final distance (Takagi et al. 2002). First the system is calibrated for each user based on previously collected pointing tasks. Then, taking into account that the user reaches the target by a straight line, the distance of the target is given to be :

$$D = a \cdot PV + b \tag{2.3}$$

Where a and b are constants provided by the linear regression model from the calibration stage, and PV is the peak velocity of the movement.

Keuning et al. (2001) found that the distance of the target can be assessed by multiplying the distance of the peak velocity by a factor of two. So they created their model on the basis that doubling the distance of a movement's peak velocity roughly predicts the final movement length. They were able to make predictions of the target distance when the cursor traversed about half that distance. Both this and the Delphian Desktop techniques are based on linear regression. In general, they are simple but did not produce very accurate, high-quality predictions.

2.2.2 Kinematics Based

Lank et al. (2007) took a more advanced approach that extends established laws of motion kinematics like minimum jerk law (Flash and Hogan 1985) and the stochastic optimized sub-movement model (Meyer et al. 2018) to develop a predictive equation. While the full details of these are beyond the scope of this paper, essentially their approach involves a two-step process. The first step is to fit a quadratic curve to a velocity versus distance profile that is created using the points of the unfolding movement. This fitting is done using least-squares regression. The predicted endpoint is given by the non-zero x-intercept of this curve. One of the issues in this step is that the velocity versus distance profile is not a perfect parabolic function and because of this, the model overestimates when predicting the end point. The second step involves determining a coefficient to account for this and correct the prediction. Using this model, a 42% target accuracy was achieved when about 80% of the motion had been completed.

Ruiz and Lank (2009) later improved upon this model and came up with an alternate method to account for the overestimation by checking the stability of a prediction directly. They were able to improve the accuracy to 51.4% with 90% of the motion complete using this new model. The issue with this approach apart from the mediocre accuracy is that the model did not allow for user-specific adaptability.

Pasqual and Wobbrock (2014) addressed this issue with their model using a method called Kinematic Template Matching. Many approaches to endpoint prediction require knowledge of possible targets, making them target-aware; conversely, this approach is target-agnostic. Their approach is also user-specific and is both easier to implement and more accurate than prior techniques. The model predicts the target that a user intends to click when using a mouse by looking at the mouse movement velocity profile and modeling mouse movement velocity time series as a 2D stroke gesture. By recording several prior mouse movements as templates and then applying template matching, the model was able to predict the endpoint location within 48 pixels when 75% of the mouse movement has been completed.

2.2.3 Control Theory

Interaction with computers is getting increasingly more dynamic. For example augmented and virtual reality devices nowadays contain intricate sensors capable of delivering very rich information about motion (Perret and Vander Poorten 2018). Such devices are capable of performing feats such as physics simulation, thereby bringing the interaction out of the computer into the real world. To model interaction with these systems, there is a need to understand the dynamics of the motion. Müller et al. (2017) looked into the dynamics of a simple one-dimensional mouse pointing task derived from Fitts' reciprocal tapping task (see Figure 3.4a). They dissected the anatomy of a mouse movement using time series plots, velocity profiles, acceleration profiles, phase space plots and Hooke plots. We will be using similar methods to examine the dynamics of mouse motion in the tests that will be explained in the next chapter.

In the models discussed in Sections 2.2.2 and 2.2.1, there is not much attention paid to the complexity of the motion in the interaction and the dynamics of the unfolding motion are ignored. The regression and kinematics based models mostly summarize the motion of the interaction using statistics like velocity of the movement and distance to the target. However, control theory takes into account multiple aspects of the interaction such as movement time, pointer position, velocity, and acceleration on a moment-to-moment basis.

Figure 2.1 provides a framework laid out by Müller et al. (2017) for a control theoretic model of interaction, to help understand interaction as a dynamical system. In the figure, there is the pointer (or stimulus) on the display, with its position changing over time. The variable that changes over time is known as the *signal*. The signals are given by the black arrows. The user continuously perceives the pointer position over time. The user also has a certain indented pointer position that they subtract from the observed pointer position. This is called the *error signal*. This signal is used by the subject to drive their bio-mechanical system to move their arm. The arm motion is measured with an input device; in this case, the input device being a mouse. The mouse signal is then processed inside the computer.

So in general, the stimulus appears on the display and the response is measured in the computer which changes the state of the stimulus based on the signal processing steps. Then, the final step is the current state of the stimulus being fed back to the user via a display. So at any point in time, the stimulus affects the response of the user and the response also affects the stimulus. Based on this model Müller et al. (2017) provides the definition of continuous interaction as follows:

> *"If we can identify a continuous signal loop between user and interface, and the user is continuously trying to minimise the error between his intention and the system state, we can say that the user is continuously interacting with the system."*

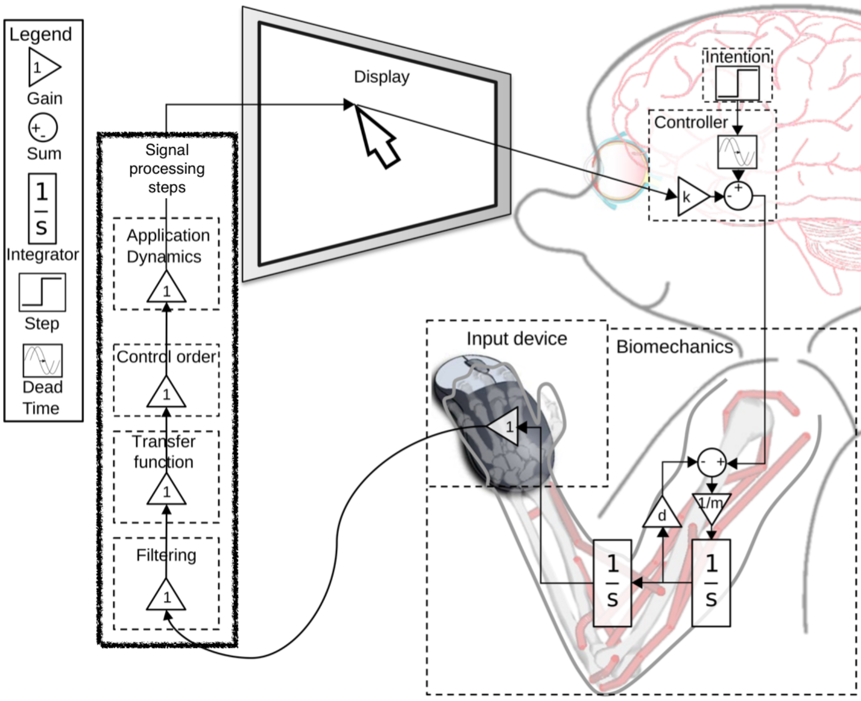


Figure 2.1: Control theoretic framework for continuous human-computer interaction laid out by Müller et al. (2017). Cursor on the display indicates the stimulus. Subject has an intent, while perceiving the current state of the stimulus. The subject then calculates the error signal as the difference between the intent and the stimulus using the controller. This signal then drives their bio-mechanical system (arm), which creates force and motion that is sensed by the input device (mouse). The signal from the input device is then processed inside the computer and the state of the stimulus is updated. The current state is then fed back to the user through the display.

Müller et al. (2017) then instantiated this framework using four existing models from manual control theory, namely McRuer's Crossover, Costello's Surge, second-order lag (2OL), and Bang-bang model. They used these models to predict the pointer position, velocity and acceleration overtime using a starting position and target as inputs.

All four models were able to predict the pointer position with good accuracy. The root mean square error (RMSE) of the predicted pointer position to the actual pointer position was used as the metric for measuring performance. The RMSE was 8.8% for 2OL, 9.1% for McRuer , 8.4% for Bang-bang, and 8.9% for Surge. All four models also reproduced human-like velocity profiles to some degree. However, some characteristic deviations were visible in phase space plots. None of the models were able to accurately predict the acceleration behaviour. The reason for this is that since the models are of second order, they can generate a step change in acceleration. This is not possible by human beings due to cognitive constraints, as well as due to the fact that muscle activation and thus force and acceleration builds up over time.

The point Müller et al. (2017) is trying to get across is to embrace the dynamics of motion more in interaction. This will enable us to understand the dynamics of more complex pointing techniques like mid-air gestures in VR, and will allow us to design for interaction with complex dynamical systems involving physics simulations. With this in mind, we will explore the design of a test suite in the next chapter that consists of many two-dimensional pointing tasks. The purpose of these tests are to explore the dynamics of a greater variety of motions.

3 Test Suite Implementation

3.1 Choice of technology

Before development on the test suite began, there was exploratory work conducted to determine which tools to use for the implementation. Initially it was thought to use an open-source cross-platform library called ‘libpointing’ to get raw events from the mouse. This was the same one that was used by Müller et al. (2017) . However, by advice of the project supervisor, who was involved in the aforementioned project, it was determined that this library was not compatible with python, and was also buggy at the time.

OpenSesame was looked into as another option as it was a free and open-source software for creating experiments via a graphical user interface (Mathôt et al. 2011). It also came with a plugin called ‘mousetrap’ that was intended specifically for easily building mouse-tracking experiments (Kieslich and Henninger 2017). These were tried but OpenSesame was found to be quite restrictive in creating experiments as the tests were not parameterizable. The icons and shapes used had to be directly taken from the collection provided by OpenSesame, which made the tests static.

Finally, it was decided that Pygame would be used to design the test suite. It is a set of Python modules designed to create fully featured games and multimedia programs in the python language. Pygame also comes with an inbuilt module called ’pygame.mouse’, that can be used to get the current state of a mouse device such as cursor position and button presses. There are also functions to alter the system cursor for the mouse by which it is possible to set the cursor position to our choosing. These libraries provided the tools necessary to create a modifiable and easy to use test suite without having to use any additional plugins. All the visualizations for the evaluation of the gathered data were done with the help of Python libraries such as pandas, NumPy and Matplotlib.

3.2 Tests

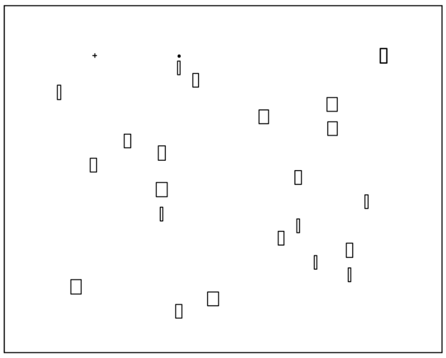
This section will depict each of the 6 tests that make up the test suite and also explain the reasoning behind the design and implementation of each test.

3.2.1 Test #1

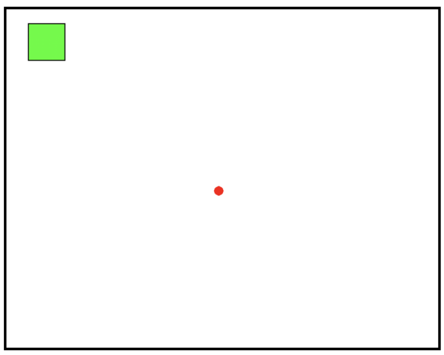
In a discrete pointing task, the subject first moves the mouse to a specified starting point and then initiates a speeded movement to the target. This procedure is then repeated multiple times. Wright and Lee (2013) implement their version of a discrete pointing task, as illustrated in Figure 3.1a. In their task, the dot is the starting point. The dot is filled in red when the cursor is too far away and is filled in green when the cursor is within 3px away. A green dot indicates that the movement is ready to be initiated. The 22 rectangles in Figure 3.1a are the targets. Once the dot turned green, the subjects were required to move the mouse to bring the cursor within a target rectangle, click it and bring the cursor back to the starting point. This was repeated for all 22 targets.

The test designed for this project is derived from this task, however, with are some changes and improvements. Figure 3.1b shows the implementation of the discrete pointing task in the test suite.

In Figure 3.1a, the targets are cluttered and it was reported that the subjects found it difficult to locate successive targets without practice. To improve on this, the test suite implementation displays only a red dot to indicate the starting point, and a single target at a time. The aim of the test is to click on the target as quickly and accurately as possible. Since we are only interested in the movement from the starting point to the target, instead of having the subject bring the cursor back to the red dot after clicking the target, the cursor automatically resets to the starting point and the target is displayed at a different location. By doing this, the time for conducting the test reduces and the amount of useful data increases as the motion can be repeated more number of times in the same time frame. Furthermore, the test has been parameterized so that the targets can be modified to any size. The coordinates and number of the targets can also be changed easily to fit the requirements.



(a) Implementation by Wright and Lee (2013). Comprised of 22 rectangular targets with the dot as the starting point. The cross represents the cursor. Subjects were required to bring the cursor to the starting point, then click on a target rectangle, and bring the cursor back to the starting point. This was repeated for all 22 targets.



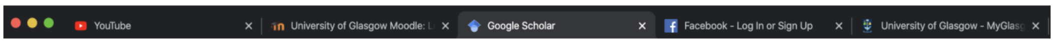
(b) Implementation in test suite. Comprises of a single green square target with the red dot as the starting point. Subjects were required to bring the cursor from the dot to the target, click on it, after which the cursor resets to the starting point and a new target appears.

Figure 3.1: *Discrete pointing task implementations*

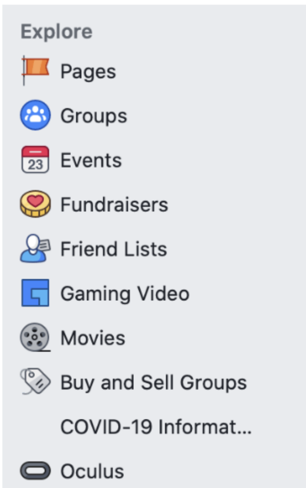
3.2.2 Test #2

One of the most widely used menu types in desktop applications and consumer electronics are linear menus. These are where items are arranged in a clustered manner as a horizontal or vertical list, as illustrated in Figures 3.2a and 3.2b, respectively. This test is also a discrete pointing task, but aims to gather mouse pointing data when interacting with linear interface elements. Figures 3.3a and 3.3b shows the design layout of this test that represent horizontal and vertical layouts, respectively.

In each case, there are 5 rectangular boxes and one of the boxes will be filled in with green to indicate the target to be clicked. The red boxes are the distractor targets. The black dot along the middle of the targets, indicate the starting point and the aim of the test is to click on the green target as quickly and accurately as possible. Once the target has been clicked, the mouse cursor resets to the starting point and another box will be filled in with green. This repeats for all 5 boxes, after which the black dot shifts to a new starting point further down the middle and the test is repeated. There are 4 different positions of starting points. By this we apply Fitts' Law and can compare the performances of attaining the targets from each starting position.

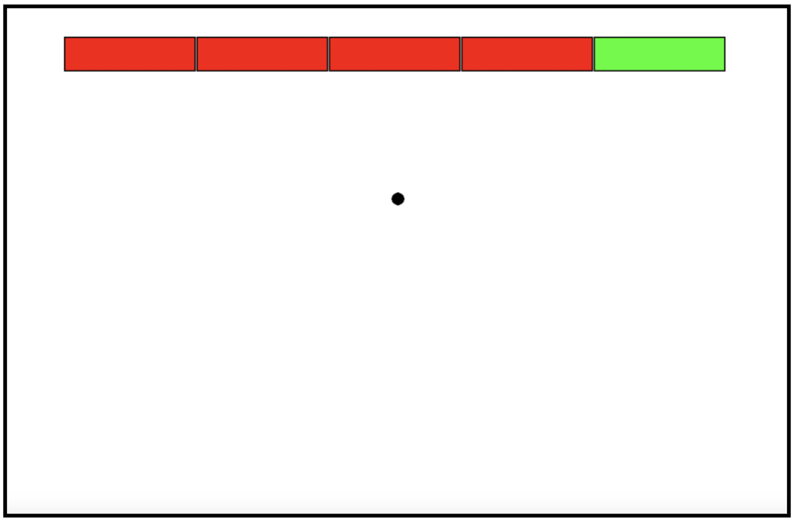


(a) Horizontal linear layout of tabs in Google Chrome



(b) Vertical linear menu found on Facebook's homepage

Figure 3.2: *Examples of linear layouts*



(a) Horizontal linear layout that represents tabs/menus similar to that depicted in 3.2a.

(b) Vertical linear layout that represents menus similar to that depicted in 3.2b.

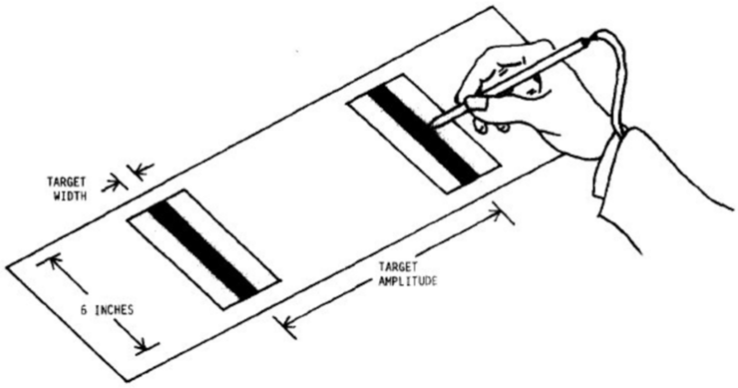
Figure 3.3: *Test suite implementations of linear layout discrete pointing tasks. Green rectangle is the target to be clicked and the red ones are the distractor targets. The black dot along the middle of the targets, indicate the starting point. Subjects were required to click on the target as quickly and accurately as possible, after which another box becomes the target. After all 5 boxes are eventually targeted and clicked, the test is repeated with a new starting point further down the middle.*

3.2.3 Test #3

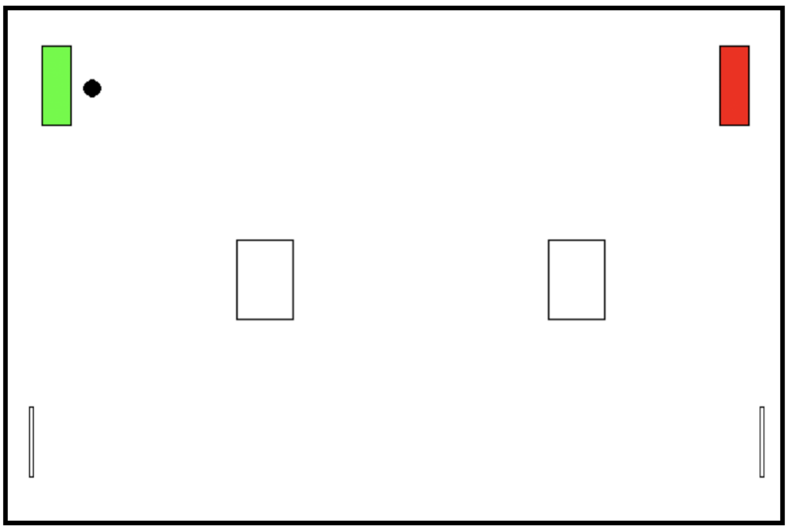
This task replicates the Fitts' reciprocal tapping task, that is illustrated in Figure 3.4a. In the original tapping task, Fitts had subjects move a stylus alternately back-and-forth between two separate target regions, tapping one or the other target at the end of each movement. These types of tasks are called *continuous* or *serial* tasks, where the subject is not expected to stop after finishing one movement but instructed to repeat the same task symmetrically as quickly as possible.

This task is replicated as a one-dimensional pointing test, as shown in Figure 3.4b. There are 3 pairs of bars with varying target width and amplitude. Only one pair is used at a time. The user is to use the mouse to move the black circle (radius: 8px) back and forth between bars, clicking on them as many times as possible within 90 seconds. Only the x-dimension of the mouse motion is used to move the circle, that is restricted to only move horizontally. Moving to a bar and tapping is intended to be an abstraction of actions such as selecting icons.

This task will help us examine the dynamics of a simple target acquisition task and the effect of the varying target widths and amplitudes.



(a) *The original reciprocal tapping task used by Fitts (Fitts 1954). Subjects were required to move a stylus back and forth between two plates as quickly as possible and tapped the plates at their centres.*



(b) *One-dimensional reciprocal tapping task implementation in test suite. 3 pairs of bars, each pair with different target width and amplitude. Subjects were required to move the black circle using the mouse and click back-and-forth between the colored bars as many times as possible.*

Figure 3.4: *Reciprocal tapping task implementations*

3.2.4 Test #4

This is a two-dimensional continuous task. In Figure 3.5, the green dot represents the cursor position and the green box represents the target. When the cursor is over the target, the alternate box gets filled in green and becomes the target. The aim of the test is to move back-and-forth between the two boxes as many times possible within 90 seconds. The two lines indicate boundaries of a tunnel and the subject should try to stay within the boundaries while performing the task. There are 3 target and tunnel widths. After every 90 seconds the size of the target reduces and the tunnel becomes narrower. By doing this, the test combines the effects of Fitts' Law and Steering Law.

Figure 3.5: *Two-dimensional continuous pointing task. Green circle follows the cursor. The two lines indicate boundaries of a tunnel. Subjects were required to use the mouse to move the cursor back-and-forth between the two boxes as many times possible while trying to stay inside the tunnel.*

3.2.5 Test #5

A cascading menu system displays submenus off to the side when selected, as illustrated in Figure 3.6a. As discussed in section 2.1.2, the Steering Law has a big effect on the usability of cascading menus. By the Steering Law, it will take more time to move precisely through a menu with narrow tunnels; and moving quickly will result in more accidental crossings of the tunnel borders. In most cases, crossing the tunnel boundaries interrupts the users intent. For example, in a hierarchical dropdown menu, if the user moves the cursor outside the menu area, the menu disappears.

The menu in Figure 3.6a is the hierarchical menu structure in MacOS. Moving the cursor using a mouse through this menu involves a series of linear path-steering tasks separated by 90-degree turns. A submenu opens on hover over an item within the main menu. In this L-shaped sequence, moving from 'Shuffle' to its child menu includes moving the cursor through the narrowest tunnel, which makes the task slower and more difficult to perform without errors.

Figure 3.6b illustrates how users may tend to move diagonally from 'Shuffle' to an item in the child menu. Doing this will result in the mouse crossing the area for 'Repeat' which will cause the submenu for 'Shuffle' to close and the one for 'Repeat' will open instead.

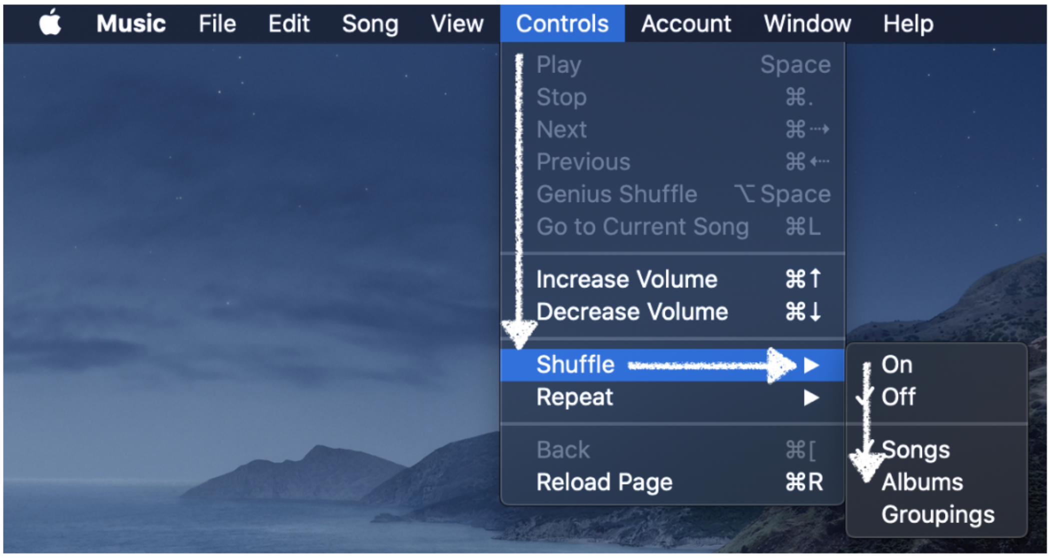
The L-shape tunnel is what we are interested in this scenario and the effect of the width and length of the tunnels on the performance of this motion. Figures. 3.7a and 3.7b shows how this is replicated in the test suite. Figure 3.7a replicates the L-shape with longer horizontal tunnels and Figure 3.7b is a variation of the same with longer vertical tunnels. The red dot represents the cursor position and moves along with the cursor. The aim of the test is to move the cursor through the tunnel as quickly as possible to the area given by the green square and click it, while always trying to keep a part of the red circle within the tunnel boundaries. So even if the cursor crosses into the black region, there can still be a part of the red circle inside the tunnel. There are 3 different tunnel widths so we can test the impact of the Steering Law. After every 90 seconds the tunnels becomes narrower.

Although only two tunnel patterns are included in the test suite, this test has been designed such that the tunnels can be modified to have a large number of variations as per the requirements. This has been done by representing the tunnel and the black region surrounding it as a maze. The maze is essentially similar to a binary matrix. In the matrix, 0 produces a black block, 1 produces a white block and 2 produces the target green block. Following this, the representation of Figures. 3.7a and 3.7b in the program are respectively given by:

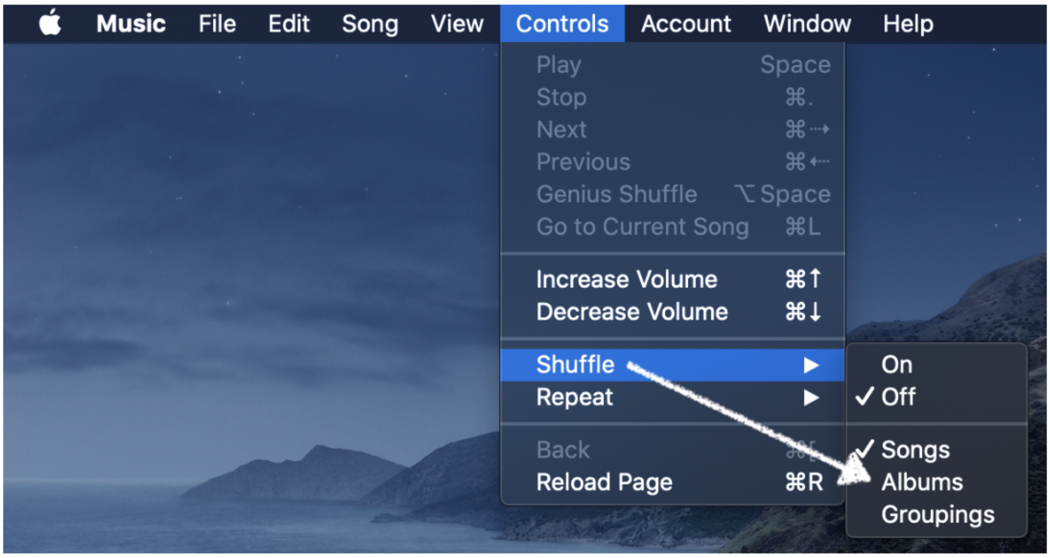
[1,1,1,0,0,0,0,0,0,0,
0,0,1,1,1,0,0,0,0,0,
0,0,0,0,1,1,1,0,0,0,
0,0,0,0,0,0,1,1,2,0,
0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,], and

[1,0,0,0,0,0,0,0,0,0,
1,0,0,0,0,0,0,0,0,0,
1,1,0,0,0,0,0,0,0,0,
0,1,0,0,0,0,0,0,0,0,
0,1,1,0,0,0,0,0,0,0,
0,0,1,0,0,0,0,0,0,0,
0,0,2,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0,0,]

By changing the positions of the 0's, 1's and 2's, we can create a vast number of patterns (not restricted to only tunnels) to test cursor motion in a constrained space using this test alone.

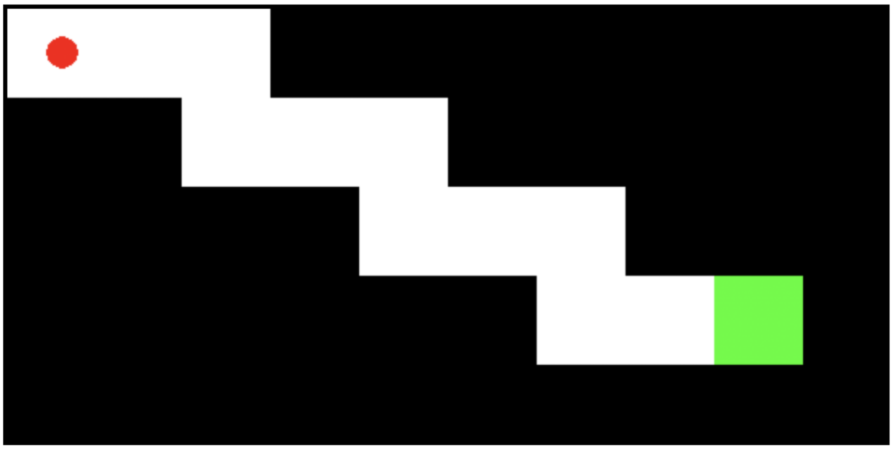


(a) Hierarchical menu structure in MacOS. Arrows indicate moving the mouse cursor through a series of linear path-steering tasks separated by 90-degree turns in an L-shaped sequence. The second step in this sequence involves the narrowest tunnel, which can be slow and difficult for users to move through without errors.

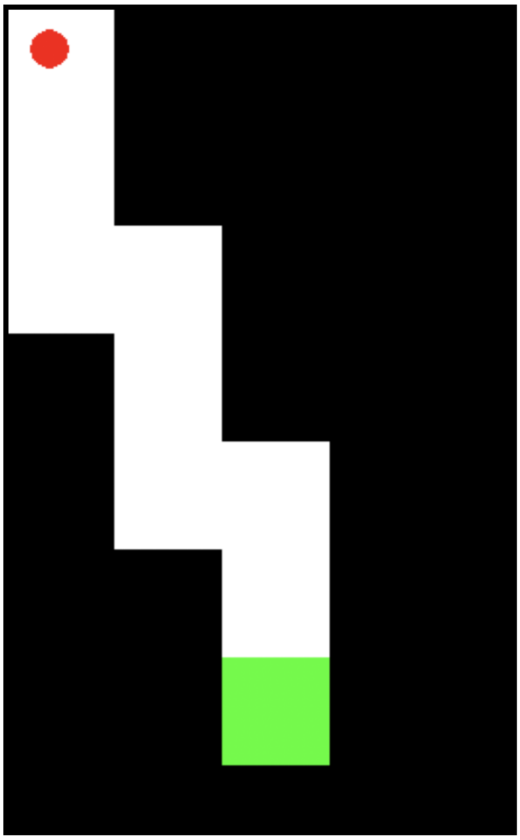


(b) Users may tend to move diagonally from 'Shuffle' to an item in its child menu. But in doing so, the mouse cursor will cross 'Repeat' and open its submenu and the target submenu will be lost.

Figure 3.6: *Effect of Steering Law on hierarchical menus (figures inspired by Laubheimer (2019))*



(a) L-shape sequence with longer horizontal tunnels.



(b) L-shape sequence with longer vertical tunnels.

Figure 3.7: *L-shape tunnel implementation in test suite. Red circle follows the cursor. Subjects were required to move the cursor as quickly as possible through the tunnels and click on the green square while always trying to keep a part of the red circle within the tunnel.*

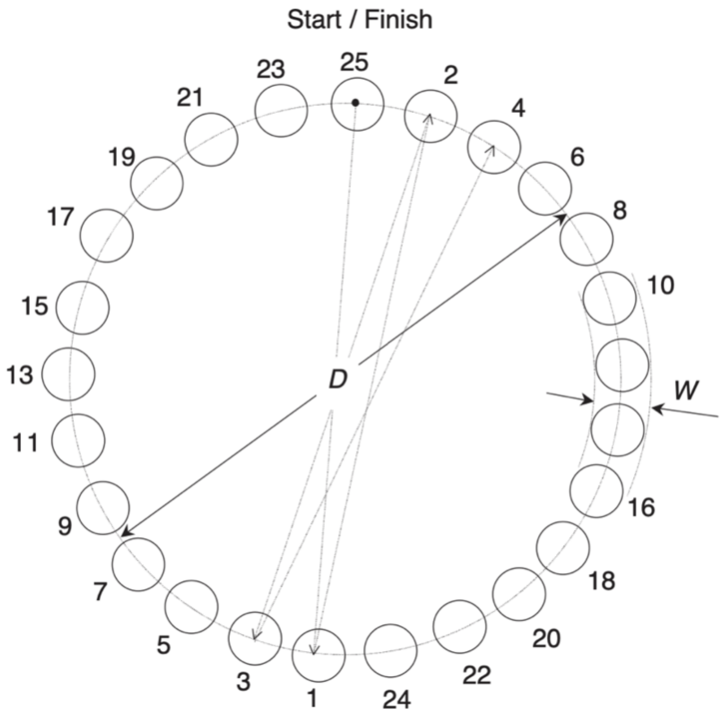
3.2.6 Test #6

This test is derived from the multi-directional tapping task described in the ISO9241 standard (ISO 2019), that is illustrated in Figure 3.8a. The International Organization for Standardization (ISO) published the ISO/TS 9241-411 in 2012. It is called, “Ergonomics of human-system interaction: Evaluation methods for the design of physical input devices”. This standard helps researchers in adopting a uniform measurement technique and avoid any bias in the results; also establish procedures of testing for evaluating pointing devices created by different companies.

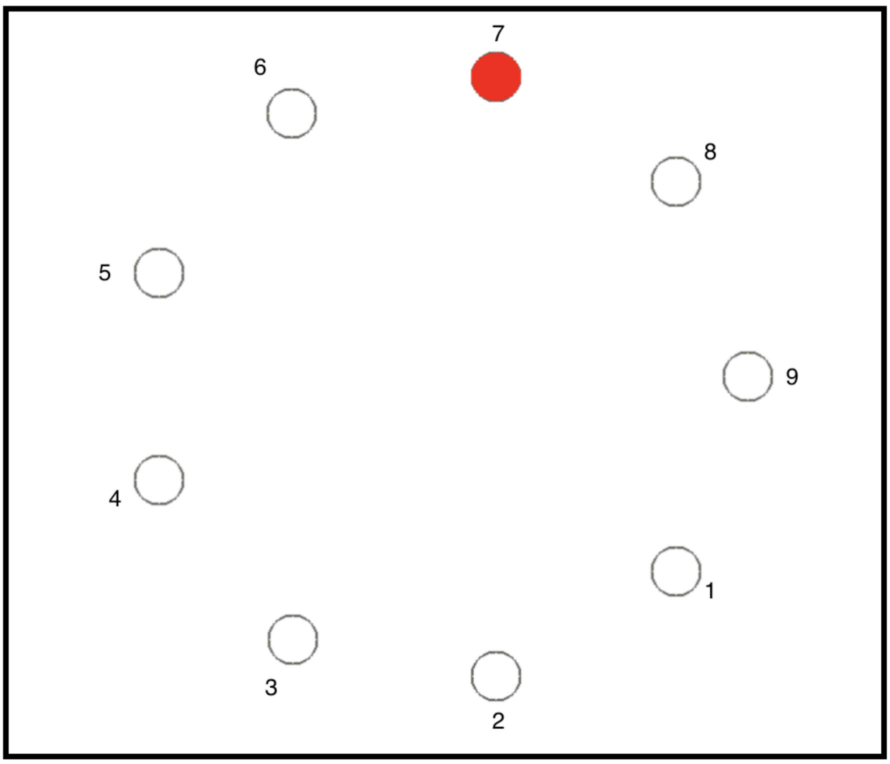
As this project is not funded, it was not feasible to purchase the standard. So the implementation of this tapping task in the test suite does not adhere to ISO9241. However, it mimics the same behaviour.

By the ISO9241 standard, the task involves tapping the circle at the top and continues clockwise following the ordered number. The arrows indicate the path subjects follow using the mouse, to alternating targets clockwise around the circle. Figure. 3.8b shows the implementation of this task in the test suite. The red-circle is the graphical indicator for the target the subject should proceed to next. The numbers indicate the target numbers (not the target order) as reference for later. The target order follows the similar alternating pattern as laid out by the ISO standard.

This test has the advantage of controlling for the effect of direction as the movement is not restricted to a specific area. There is also a variety of movement distances (D) and target widths (W) that have been chosen so that the subjects face a large and representative range of ID values.



(a) Multidirectional tapping task described in ISO (2019) standard. The lines indicate the path followed using a pointing device, to alternating between targets clockwise around the circle. Path begins and ends in the top target.



(b) Multidirectional tapping task implementation in test suite. Subjects were required to use the mouse and move the cursor to click on the target, given by the red-circle that alternates and goes around in a clockwise direction. Numbers indicate the target number (not target order as depicted in the Figure 3.8a) as reference for later.

Figure 3.8: *Multidirectional tapping task implementations*

4 Creating The Dataset

Here we detail the procedure and apparatus used to carry out data collection by participant trials.

4.1 Participants

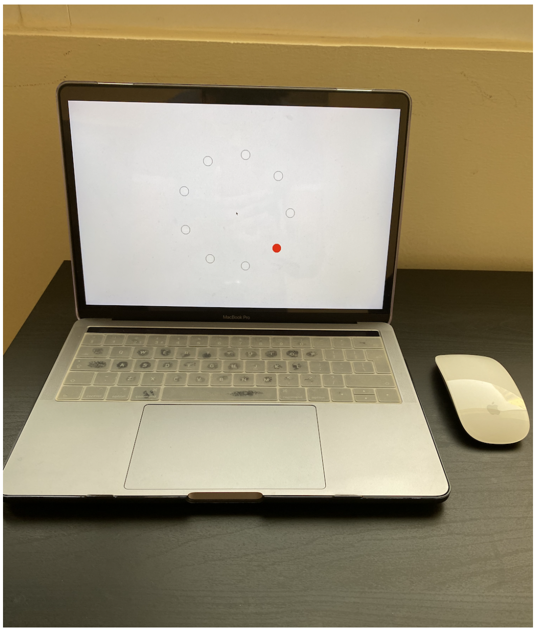
6 able-bodied participants and 1 disabled participant (mean age 29.42 years, std. dev. 11.58, 5 male, all normal or corrected to normal eyesight) took part in the experiment. Although one of the participants was disabled, it did not affect their hand-eye coordination in any manner as the disability did not extend to the upper-body. All the participants were right-handed and used the mouse with their right hands. Before beginning the experiment, all participants were asked to self-rate their computer proficiency on a scale of 0-10, 0 being the lowest. These scores ranged from 5 to 10 (mean score 9, std. dev. 1.69). The initial aim was to gather at least 12 participants so that a large enough data set could be formed that captured a variety of user behaviour with the mouse. However, as social distancing measures were put into place due to the COVID-19 outbreak, it was not possible or safe to ask any more people to take part in the experiment.

4.2 Apparatus

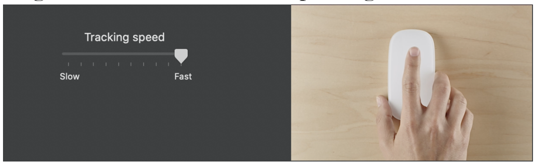
The test suite was run and data captured on a 2017, 13-inch Macbook Pro, running on MacOS Catalina, displaying a 2560x1600 native screen resolution at 227 pixels per inch, with a 60 Hz refresh rate. Further specifications include a 3.1 GHz Dual-Core Intel Core i5, with 8GB RAM. The Apple wireless Magic Mouse 2 was used as the input pointing device. This mouse is optimized to be used with Apple computers and also has an enhanced foot design for easier tracking and to move with less resistance across a desk. For this reason a mouse pad was not additionally bought and used for the experiment. The tracking speed of the mouse was set to maximum setting for this experiment. The setup is shown in Figure. 4.1.

4.3 Procedure

Participants were initially requested to adjust the table and display to their preference. They were then required to sign a consent form. The form detailed the purpose of the experiment and all necessary background information relating to how the experiment would be carried out. Rather than give a demonstration of all 6 tests in one go, each test was demonstrated first and then the subject was asked to perform the task. This allowed for small breaks in between each test to prevent the participants from experiencing any form of strain or cramps in their hands. It also allowed to keep the instructions fresh in their memory since each test had a different set of requirements. A demonstration protocol document was formulated before carrying out data collection, which focused on providing the exact same instructions to every participant for the purpose of ensuring a controlled experiment. The protocol gives explicit details of the step-by-step manner by which each test was demonstrated and the instructions that were given to each participant before asking them to perform the task. The consent form used and the demonstration protocol is given in the appendix Section A.1.



(a) The apparatus used. Includes a 2017, 13-inch Macbook Pro to run the tests and capture data. An Apple wireless Magic Mouse 2 was used as the pointing device.



(b) Mouse tracking speed setting.

(c) A participant interacting with the apparatus.

Figure 4.1: The setup used for the experiment.

5 Evaluating The Data

In this chapter, we work directly with the raw dataset obtained from the experiments. The data from each task was pre-processed appropriately to facilitate further analysis. The explicit details of the set of conditions and parameters that were used for each test like index of difficulty, target widths, amplitude of the motions and tunnels widths and lengths will also be specified.

5.1 Test #1

There were a total of 10 targets, all identical, but in different locations on the screen. One iteration consists of clicking on all 10 targets. The target size became smaller after 10 such iterations. There were three different target sizes. The amplitude (distance from starting point to target), width and ID values of each target are summarized in Table 5.1:

Table 5.1: *Summary of test #1 targets*

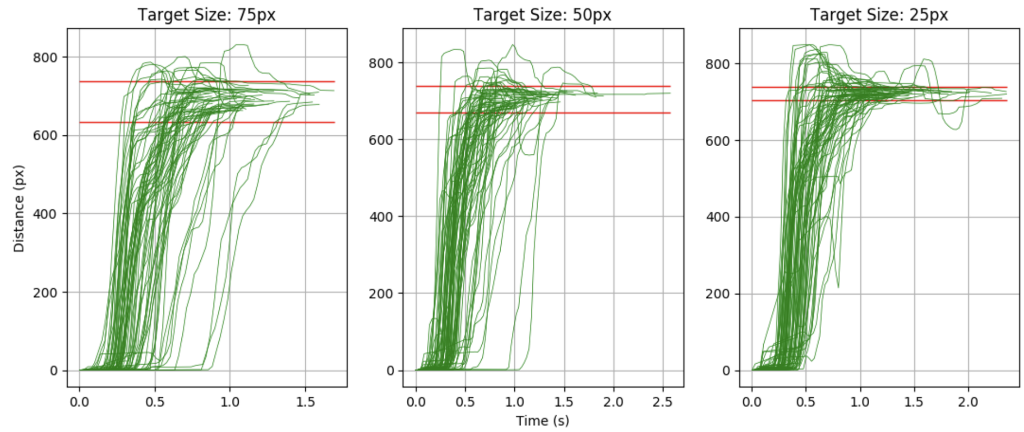
	ID		
	Width *(in px)*		
Amplitude *(in px)*	75	50	25
737.83	3.43	3.97	4.93
225.61	2.00	2.46	3.32
546.71	3.05	3.57	4.51
637.88	3.24	3.78	4.72
748.86	3.45	3.99	4.95
493.35	2.92	3.44	4.37
399.24	2.66	3.16	4.08
212.13	1.93	2.39	3.24
200.00	1.87	2.32	3.16
350.0	2.50	3.00	3.90

Let us now examine how the pointer position changes over time in the case of target 1, that is at a distance of 737.83px from the starting position. We will do this using time series plots. Figure 5.1a shows how the distance of the cursor changes relative to the starting point for all participants in all their trials across each target width. These plots seem a bit too cluttered; so to examine the motion more clearly, we will first look at the plot for only one participant in one of their trials.

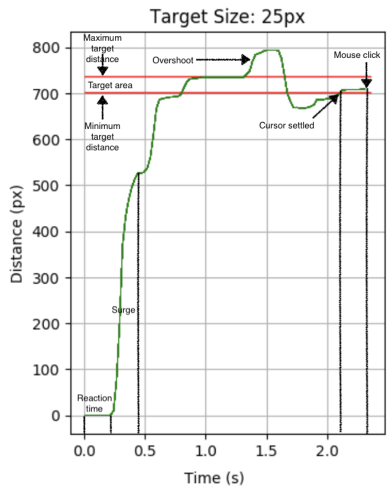
Figure 5.1b shows the mouse motion for participant 1 in trial 3 (target size: 25px). We will analyze the motion in a similar way to that done by Müller et al. (2017). From this figure, we can see how there is an initial reaction time before the pointer starts moving. Then there is a ballistic phase or a *surge*, when the cursor moves towards the target. The two red lines indicate the minimum and maximum distance to the target. Essentially, we can consider the space in between these to lines to be the target area. As the cursor approaches the target, there may be *overshooting*. To correct this, there are some further oscillations around the target, until finally the cursor has *settled* on the target area and the user clicks.

Figure 5.1c depicts all the trials for participant 1 for all 3 target widths. We can see as the target size gets smaller, overshooting increases. Also as the target size reduces, the time taken to click the target increases. This is explained by the Fitts' Law equation (Equation 2.1), when MT increases as target widths W decreases.

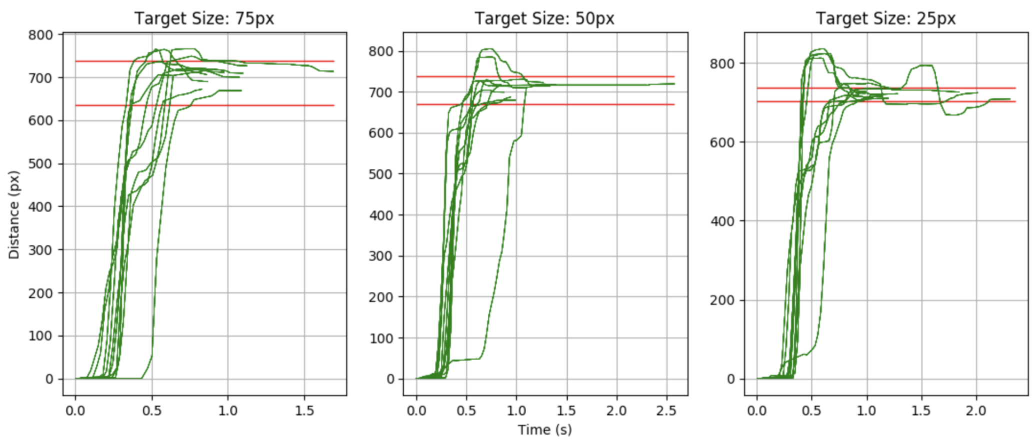
So, this test has been helpful in dissecting a simple discrete pointing task. We also explored the effects of target size on accuracy and movement time, thereby using this test to confirm and supplement Fitts' law.



(a) All participant trials for all widths of target 1.



(b) *Trial 3 for participant 1 for target 1 of width 25px. There is an initial reaction time, after which the user moves the cursor towards the target in a ballistic manner, called the surge phase. There is then some overshooting and subsequent corrective oscillations towards the target area. Once the cursor is on the target, it rests for sometime before the user clicks the mouse.*



(c) *All trials of participant 1 for all three target widths. As target width gets smaller, overshooting increases and so does the the time taken to click the target.*

Figure 5.1: *Time series plots depicting the change in the distance of the cursor relative to the starting point.*

5.2 Test #2

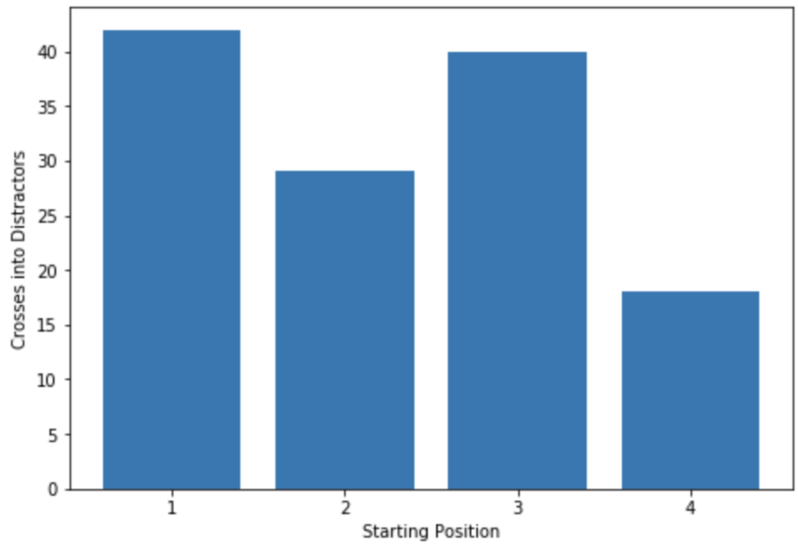
Unlike test #1, the target sizes for this test were kept constant and instead, the starting position was shifted. As depicted in Figure 3.3, there were 2 variations of this test. For each, there were 4 sets of starting positions. The starting position would change by moving further away along the middle of the targets, after all 5 targets were successively clicked. In the case of the horizontal layout, this meant the starting point shifting lower along the middle, and in the case of the vertical layout, shifting towards the right. An iteration for this test consisted of clicking on all 5 targets from each of the 4 starting points. There were a total of 10 iterations. For the horizontal layout, the ID's ranged from 1.23 to 2.48 and for the vertical layout, the range was from 3.02 to 4.29.

This test revealed some interesting results that gave some insight into the interaction pattern of the users. The users were instructed to click on the targets as quickly as possible. For any of the given 5 targets in either layouts, this means moving the mouse in a straight line directly to the target, without crossing any of the distractor (red) targets. Figure 5.2 shows bar plots of the number of readings observed when the cursor crossed into the distractor targets. The x-axis labels are the starting points, with 1 being the closest starting position and 4 being the furthest. From Figure 5.2a (that shows the statistic for the horizontal layout), we don't observe a considerable difference between each of the starting points. But in the vertical layout for the test, the results get surprising. Here, there is a significant increase in the number of crossings inside the distractor targets from the closest start point. From the 1st starting position a total of 116 readings were measured inside distractor targets.

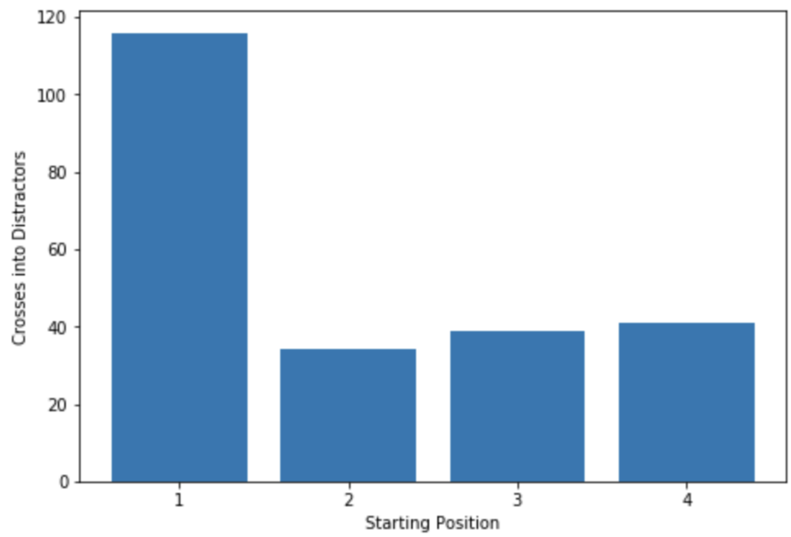
The reason this surprising is because trivially, we would expect the accuracy to be higher when the amplitude of the movement is short as the ID is lower. But for this test, the results are conflicting. To explore this further, we can look at Figure 5.3 that depicts the cursor movements from position 1 to each of the targets (target 1 is the bottom target and target 5 is the topmost) for all trials. We notice that as the target is further away from the centre, there are successively more crosses into the distractor targets. So the most number of crosses into the distractors are observed in bottom-most and topmost targets, i.e targets 1 and 5, respectively. If we observe Figure 5.4, that depicts the cursor movements from position 4, we see that the plots are much less dramatic with most of the plots following a straight path to the target without many crosses into the distractors.

One possible explanation for why the accuracy is lower for starting position 1 could be the effect of the Steering Law. As explained in Section 2.1.2, the natural tendency of the human arm is to follow a curved motion. This is because the motion of our wrists, describes an arc and not a straight line. And so, for short movements, the user may tend to only use their wrist and not lift their arms to drive the mouse, hence making it even more difficult to move the mouse in a straight line. The motion would therefore describe an arc, which in turn results in the cursor crossing into other targets. So even though this test did not require moving the cursor through a tunnel, the Steering effect may have potentially played a role in decreasing the accuracy.

When creating machine learning models to predict endpoints, it is important to consider these anomalies. In the case of this test, the targets used were arguably larger and were more spaced out than the vertical menus implemented in websites. So when designing machine learning models for interface elements, it would be important to consider even the smallest effects of human movement that may not strictly follow traditional predictive models like Fitts'. By using tests like this and gathering more data, perhaps the nature of these motions could be studied even more, which would help in creating more reliable and accurate machine learning models.



(a) Readings in distractors for targets in horizontal layout from each starting point.



(b) Readings in distractors for targets in vertical layout from each starting point.

Figure 5.2: *Bar plots depicting the number of readings when the cursor crossed into the distractor targets from each starting point. 1 is the closest starting point and 4 is the farthest.*

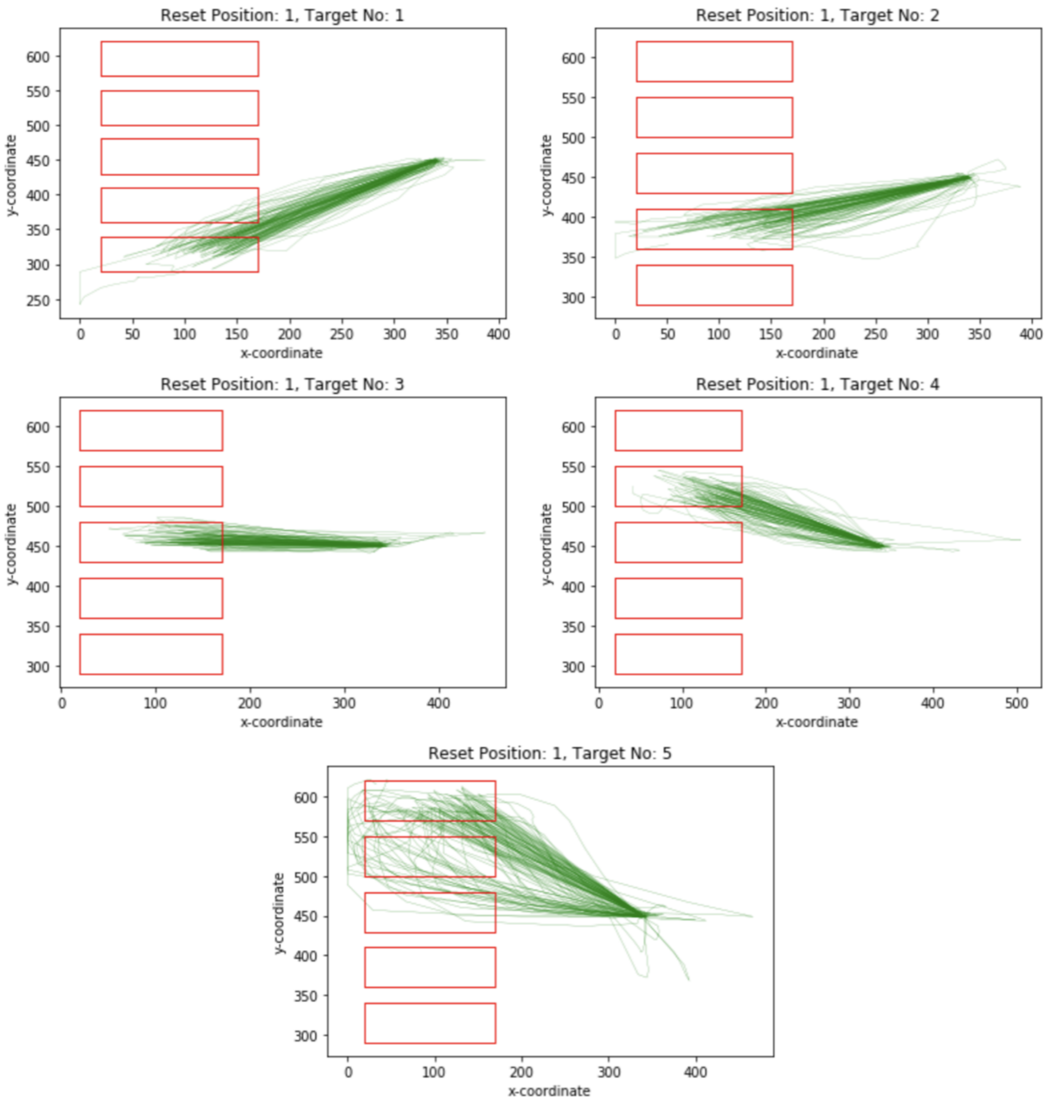


Figure 5.3: *Plots of cursor movement for all trials from the closest starting position (1) to each of the 5 targets. Red boxes indicate the targets. As the target gets further away from the centre, there are successively more crosses into the distractor targets. The most number of crosses into the distractors are observed in targets 1 and 5.*

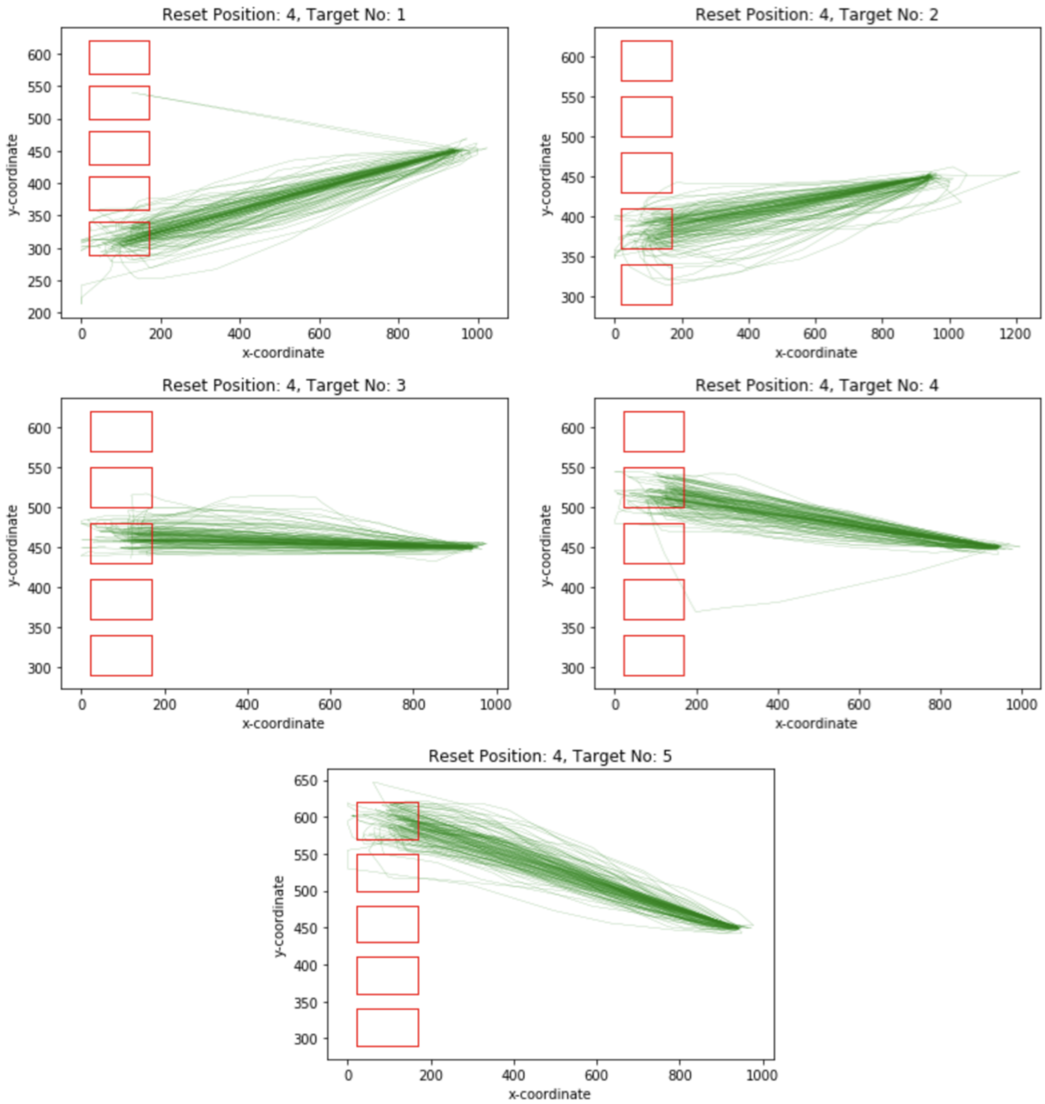


Figure 5.4: *Plots of cursor movement for all trials from the farthest starting position (4) to each of the 5 targets. Most of the plots follow a straight path to the respective target without many crosses into the distractors.*

5.3 Test #3

For the reciprocal mouse tapping task, 3 pairs of bars were used. The target length was kept constant at 100px, while the target width and amplitude (distance between the targets) was different for each pair. The details of these are given in Table 5.2.

Table 5.2: *Summary of test #2 targets*

Width *(in px)*	Amplitude *(in px)*	ID
35	1345	5.30
70	910	3.80
10	1350	7.08

The participants were given 90 seconds to move the black circle back and forth between the targets for each pair of bars, clicking on them as many times as possible. Figure 5.5 shows the total number of hits on the targets for each of the ID's. As we would expect, as the ID got higher, the number of hits got lower (ID:3.80, Hits:670; ID:5.30, Hits:465; ID:7.08, Hits:337).

Now, lets try to examine some aspects of the dynamics of the motion. Using phase space plots, we plot the pointer velocity over position. Figure 5.6 shows the phase space plots for one of the participants (participant 3) for all three ID's. From the plots, there seems to be a lot of noise in the data. To filter this out and also smooth the data, a Savitzky-Golay (savgol) filter (Savitzky and Golay 1964) was applied to the initial x-coordinate pointer positions. Many window lengths and polynomial orders were tried to fit the data as best as possible. This proved to be quite tricky as the initial sampling rate itself was not high enough to produce smooth enough data. For comparison, Müller et al. (2017) conducted their similar experiment using a screen with 1000 Hz update rate and then down-sampled to 500 Hz. In the trials here, the experiment was conducted using a MacBook Pro screen that only supports a maximum of 60 Hz. After many trial and error, the savgol filter was used with a 3rd degree polynomial and a window size of 21 samples. After doing this, the velocity was recalculated and new phase space plots were generated. The new plots for all participants for all ID's are shown in Figure 5.7. Even with the filter applied, there still seems to be a lot of noise and the plots aren't as smooth as the ones obtained by Müller et al. (2017). Due to the low number of samples, the plots loose quality which make it difficult to identify the corrective movements the participant makes if they overshoot or undershoot.

However, we can still identify some motion patterns. We can see from the plots that as the ID increases, so does the peak velocity of the movement in all cases, except for participant 2. So we could say that as the distance between the targets increases, the peak velocity also rises. As we discussed in Section 2.2, a lot of machine learning models perform endpoint prediction using the peak velocity metric. So identifying patterns like this and finding correlation between ID and velocity may potentially help in increasing the accuracy of such models.

We can also identify some patterns that indicate the strategies that different participants use to move towards the targets. Müller et al. (2017) was able to identify two different strategies that we can also see here. In the case of ID 3.80 for participants 2 and 5, we can see the plots are nearly circular. This indicates one big symmetric ballistic surge movement that ends close to the target. This is what is known as an *open loop strategy*. In other cases, the participants accelerate slower and in a less ballistic manner. This might indicate that they move slower to reduce the chance of overshooting and to land closer to the target. This is known as a *closed loop strategy*.

Hooke plots were also generated for all participants. These plotted acceleration over position. However, as the filtered data in itself was noisy, taking the derivative of the velocity data to get acceleration greatly amplified the noise in these plots. So in most cases it was hard to identify any clear patterns at all. Nevertheless, the Hooke plots generated for participant 3 are given in Figure 5.8 and the plots for all participants are given in the appendix Section A.2. Participant 3's

Hooke was chosen to be displayed here since from all the phase space plots, this participants data seemed relatively less noisy and so the generated Hooke plots also had some shape to them. The N shape is clearly visible in all the plots but the noise in the data makes it difficult to comment and compare the true nature of the acceleration and deceleration.

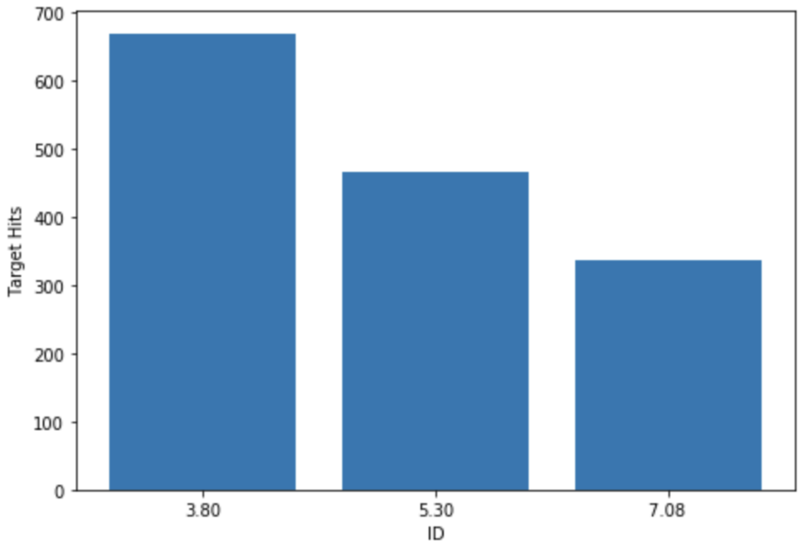


Figure 5.5: *Number of target hits for each ID. As the ID gets larger, the number of target hits decreases.*

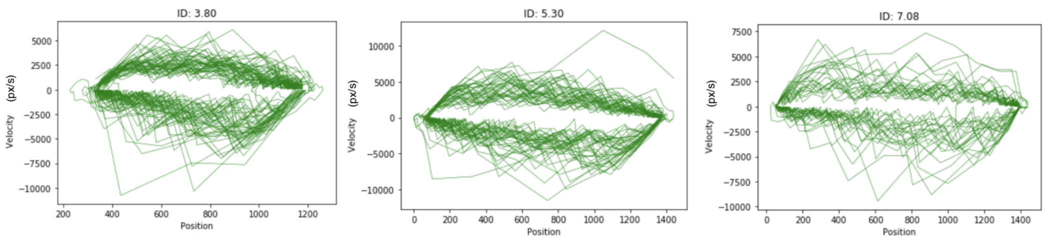


Figure 5.6: *Phase space plots of pointer velocity over position (without Savitzky-Golay filter) for all trials of the participant 3 for each ID.*

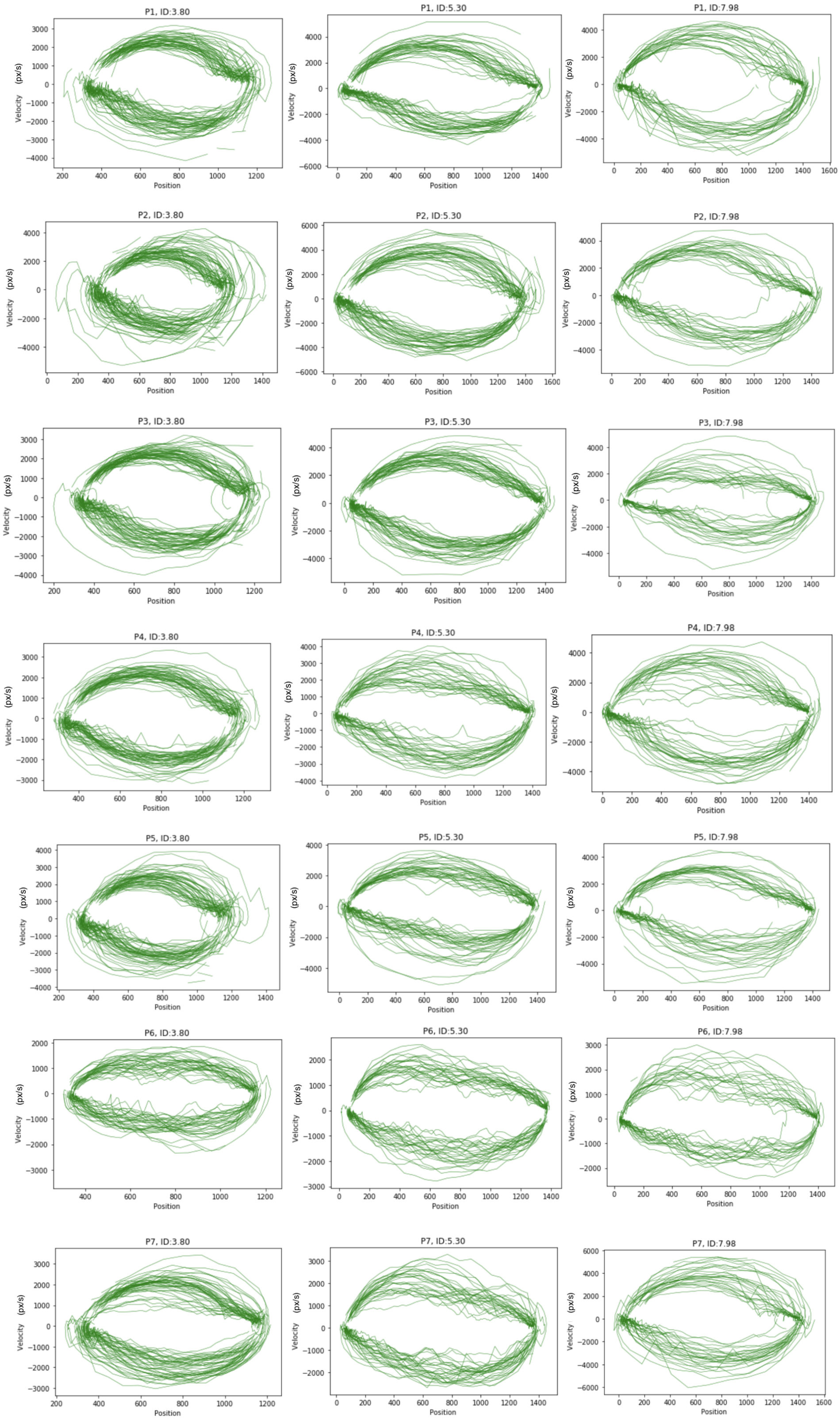


Figure 5.7: *Phase space plots (after applying Savitzky-Golay filter) for each participant in each ID. For all participants (except participant 2) as ID increases, the peak velocity increases. For, ID 3.80 participants 2 and 5, have nearly circular plots, indicating an open loop strategy. All other plots are indicative of closed loop strategy.*

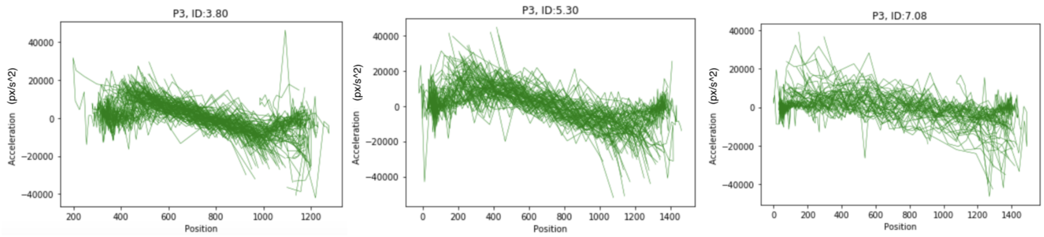


Figure 5.8: Hooke plots of acceleration over position (after applying Savitzky-Golay filter) for all trials of the participant 3 for each ID. N-shape indicates the acceleration and deceleration of the movements.

5.4 Test #4

In this test, the participants were once again given 90 seconds to move back and forth between the two targets as many times as possible. However, for this test, the movement of the circle is not restricted to one direction and the user must keep the circle within the boundaries of the tunnel. The tunnel length is kept constant at 1240px. There are 3 different tunnel widths and the the tunnel gets narrower after every 90 seconds. The target width is the same as the tunnel width. From the Steering Law formula (Equation 2.2), the Steering index of difficulties (A\W) for the tunnels are given in the Table 5.3.

Table 5.3: *Steering's index of difficulties for each tunnel width.*

Tunnel Width	Steering's ID
150	8.26
100	12.40
50	24.80

Table 5.4 below gives the average time taken by each participant to move through the tunnels and click back and forth between the targets for each ID.

Table 5.4: *Average time taken by each participant to move through the tunnels for each ID.*

	Average Time *(in s)*		
	ID		
Participant	8.26	12.40	24.80
1	0.87	0.88	1.58
2	0.80	0.82	1.47
3	0.75	0.76	0.95
4	0.80	0.85	0.87
5	0.70	0.82	0.85
6	0.90	0.84	2.27
7	0.79	0.85	1.18
Total Average	0.80	0.83	1.31
Standard Deviation	0.067	0.037	0.51

As expected, when the width of the tunnel decreases, the time to move through the tunnel increases. Hence, this test has helped confirm the Steering Law.

The bar plot in Figure 5.9 shows the number of readings gathered for each participant when they crossed the boundaries of the tunnel. Note that this is not the number of times the participants crossed the boundary, but the number of readings from the dataset when a collision was detected between the circle and the boundary lines. From these plots, we can clearly see the vast majority of the collisions were for ID 24.80, when the width of the tunnel was narrowest. Another thing to note from this plot is that participants 3, 4, and 5 have collided with the boundaries the most. These participants are also the ones that had the least average times moving through the tunnel for ID 24.80, as we can see from Table 5.4. So although these participants moved through the tunnel quickest, they were also the ones to cross the boundaries the most number of times. So subsequently, they are the participants with the least accuracy.

Incidentally, it will be the data sets of these participants that will be the most useful when creating machine learning models, because the goal when creating these models is to improve the efficiency of the task by making the prediction quicker even if the user is less accurate. And since these participants have the quickest movement times and were less accurate, we can deduce that their main goal was efficiency and not accuracy. And this is why predictive models are needed in the first place; so that even if the user is less accurate and sloppy in their movement, the model should still be able to correctly identify their intent. And this is only possible if the data used to train the model, suits that behaviour as well. This is not to say that movements with greater accuracy are any less important. But the point to take away from the results of this test is that sloppy movements also need to be given equal consideration when creating models for endpoint and mouse motion predictions as training the models with this type of data will improve usability by being able to identify intent to a greater accuracy, even if the user is not.

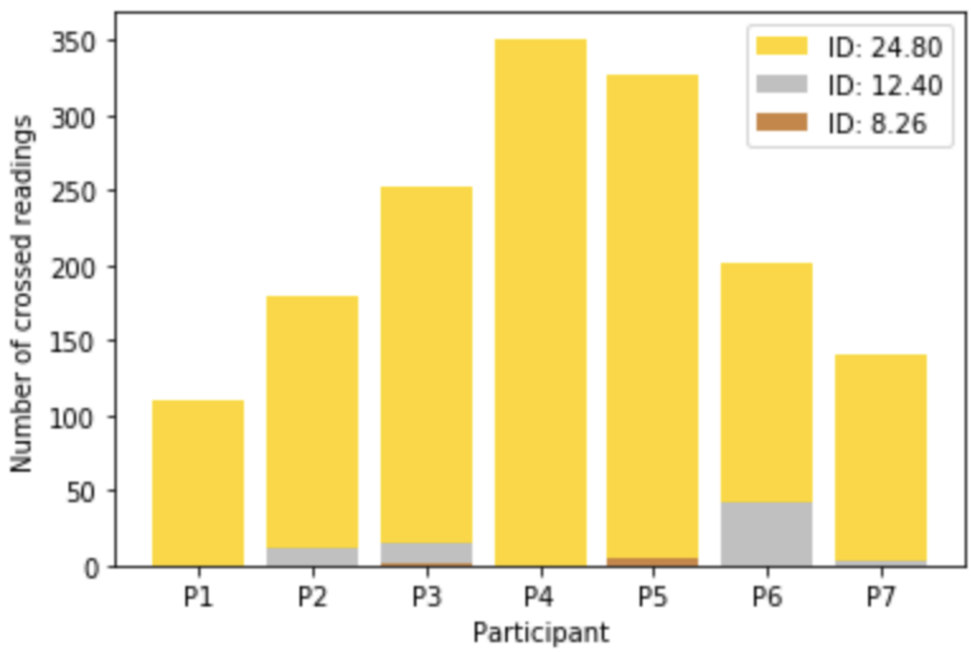


Figure 5.9: *Grouped bar plots indicating the number of readings when the circle cursor collided with the tunnel boundaries for each participant for each ID.*

5.5 Test #5

In this test the participants were given 90 seconds to move the cursor through the tunnel and click the green target. There is a red dot (radius: 10px) that moves with the cursor. The participants were told that at any point in time during the movements, they should see to it that some part of the red dot is within the tunnel. So even if the cursor crossed the boundaries of the tunnel, a part of the red dot could still be within the boundaries.

For both variations of the test, there were 3 different tunnel widths (40px, 55px and 70px) and the the tunnels gets wider and longer after every 90 seconds. Figures 5.10a and 5.10b depict the movement of the cursor through the tunnels for all participant trials. The red lines indicate the boundaries of the tunnel. By examining these plots, we can see that in general, for all the cases, the plot mostly stays within the boundaries of the tunnel during the first two 90 degree turns. But towards the end of the movement, the participants tend to cut through the final turn in a straight diagonal line rather than move in a zig-zag to get to the target. From the plots in the Figure 5.10b, that depict the movements through the second tunnel shape, we can see more trials cutting through all the turns through the corner parts of the lower border. This may tell us something interesting about the interaction behaviour. The path to the target in the second tunnel shape has two less 90-degree turns than the first tunnel shape. This may indicate that when the distance to the target is shorter, the user may have a tendency to cut through the turns rather than follow the tunnel path. Even in Figure 5.10a, the movements mostly stay within the boundaries for the first three turns, but the corners of the final couple of turns of the top border get successively less visible, indicating once again, as the cursor gets closer to the target, there is a tendency of the user to cut through the turns diagonally.

Table 5.5 gives the average number of target hits across all tunnel variations for each participant.

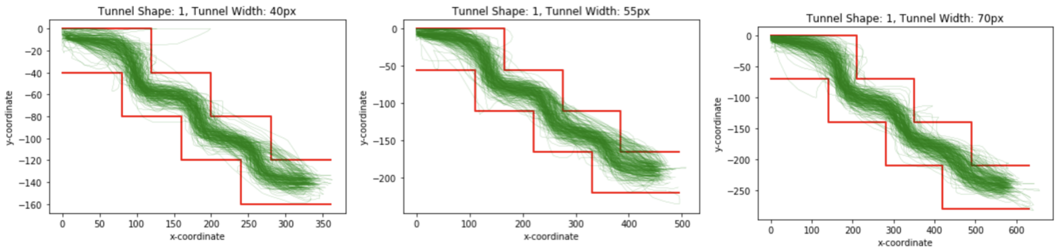
Table 5.5: *Average number of target hits for each participant across both tunnel shapes and all three widths.*

Participant	Average Hits
1	36.33
2	35.0
3	39.33
4	42.33
5	52.0
6	41.16
7	44.16
Total Average Hits	41.47
Standard Deviation	5.65

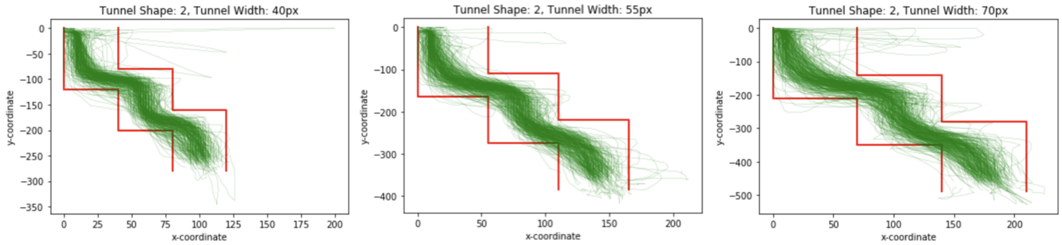
Since participant 5 has a much higher average number of hits than the rest, we will examine what strategy this subject employed to get such a high average. Figures 5.10c and 5.10d shows the cursor movements of participant 5 for all their trials. The plots for all other participants are given in the appendix Section A.3. For the first tunnel shape (Figure 5.10c), the participant generally follows the path of the tunnel without crossing the borders for the first three turns. But in the last three 90 degree turns of the movement, the participant cuts through the top boundary of the tunnel to get to the target in a straight line. This is especially prominent for the tunnel width of 55px, where for almost all the trials, the final part of the movement is a straight diagonal line.

In the case of the second tunnel shape (Figure 5.10d), the plots rarely follow the zig-zag nature of the tunnel. As the tunnel width increases, the cursor path gets successively more straight as the user cuts through the bottom boundary at nearly all the turns. This follows from what we discussed before, when a user tends to cut through the turns as the distance to the target decreases. Another interesting thing to note from this participants plots is that for the second tunnel shape of tunnel width 70px, although the user follows a straight path, there seems to be a decreased number of boundary crossings as we can see more parts of the bottom red line. This may indicate that for some of the trials, the user may have actually gotten close to the optimal path, which is a straight line through the tunnel without crossing any of the boundaries. This could explain why this participant has a much higher average number of target hits than all the others.

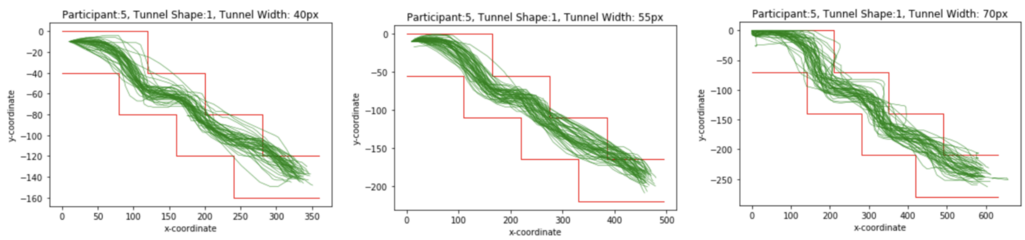
There was a lot of discussion conducted in this section on the interaction behaviour of users when moving a cursor through a tunnel. We evaluated plots that indicate user tendency to cut through the boundaries of the tunnel when the distance to the target reduces, and also that a user is likely to find and move in an optimal path when the tunnel is wide enough. Needless to say, there are many more interaction behaviours that could potentially be explored using this test with different tunnel shapes and width variations. As explained in the design for this test in Section 3.2.5, there can be a wide array of tunnel shapes created, and we can potentially uncover different behaviours for different shapes. Understanding these behaviours have a great implication when designing interface elements like cascading menus that follow the Steering Law. If a user may cut through the menu to get to an item in a sub menu quicker (like shown in Figure 3.6b), the interface element needs to be designed in a way that understand this behaviour and appropriately adapts to it. In the case of machine learning models, there was extensive background research conducted as part of this project to learn about the different types of mouse prediction models (as discussed in Section 2.2). There hasn't yet been models created to predict mouse movements through tunnels. But understanding the behaviour of users and gathering mouse data using tests like this, and coupling the results with manual control theory, could perhaps help create models capable of such prediction tasks.



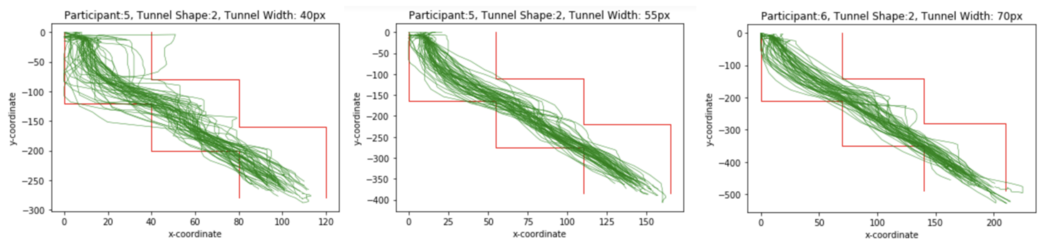
(a) Cursor movement through tunnel shape 1 for all participant trials for each tunnel width. The cursor mostly stays within the boundaries of the tunnel during the first three 90 degree turn. But towards the end of the movement, the participants tend to cut through the final turn of the top border in a straight diagonal line rather than move in a zig-zag.



(b) Cursor movement through tunnel shape 2 for all participant trials for each tunnel width. The cursor mostly stays within the boundaries of the tunnel during the first 90 degree turn. But there are more trials for this tunnel shape where the cursor cuts through all turns of the bottom border to avoid moving in a perfect zig-zag.



(c) Cursor movement through tunnel shape 1 for all participant 5 trials for each tunnel width. Participant follows a zig-zag motion without crossing the tunnel borders for the first three turns. But in the last three 90 degree turns of the movement, the participant cuts through the top tunnel boundary diagonally to get to the target in a straight line. This is especially prominent for the tunnel width of 55px, where the final part of the movement is a straight diagonal line for almost all trials.



(d) Cursor movement through tunnel shape 2 for all participant 5 trials for each tunnel width. The participant rarely follows the zig-zag path of the tunnel. As the tunnel width increases, the cursor path gets successively more straight as the participant cuts through the bottom boundary at all the turns. For width 70px, the cursor moves in a near straight line but with a decreased number of boundary crossings indicating that for some of the trials, the participant may have gotten close to the optimal path (a straight diagonal line through the tunnel without crossing any of the boundaries).

Figure 5.10: *Plots depicting the movement of the cursor through the tunnels. Red lines indicate the boundaries of the tunnel.*

5.6 Test #6

This test has three different ID's, each with a different target width and amplitude. One iteration is completed after all 9 circles have become the target once and are clicked on. There are 15 such iterations for each ID. The circles get bigger and closer after 15 iterations of each ID. The target widths, amplitude and ID's are given in Table 5.7.

Table 5.6: *Summary of test #6 targets*

Width (in px)	Amplitude (in px)	ID
40	500	3.75
60	350	2.77
80	200	1.80

Figure 5.11 shows the cursor movements for all participants for all trials when moving from target 1 to 6 for all three ID's. The red line indicates the ideal straight path between the centre of two targets. The two circle-like shapes represent the target themselves (not perfect circles as the axes are not equal). We can see for the highest ID (3.75), in most of the trials, the plot starts and ends very close to or on the red line, and near the centre of the targets. This is because due the small target size, the user is constrained by the target area and will click near the centre. As the ID reduces and the target width gets larger, we see this behaviour deviates. For ID's 2.77 and 1.80 the plots successively move away from the red line and the start and end of the motion gets further away from the centre of the targets. This is because as the target gets larger, the user has the freedom to click on a larger area to complete the motion. And so, they begin to click on the target at points further away from the centre.

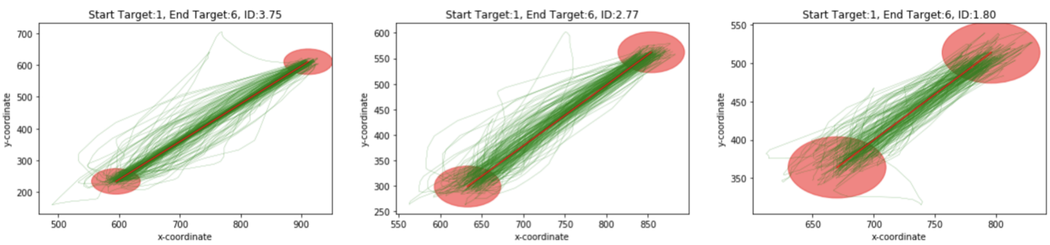


Figure 5.11: *Cursor movements for all participants for all trials when moving from target 1 to 6 for each ID. Lower circle is target 1 and upper circle is target 6. Red line indicates the ideal straight path between the centre of two targets. As the ID gets smaller (and target width get bigger), the plots successively start and end further away from the red line, indicating that the participants click further away from the target centre.*

Table 5.7 shows the overall average time it took each participant to complete the test for all three ID's. We can see that participant 2 has the lowest average time and participant 6 has the highest average time. So we will examine the plots for these two participants to compare their strategies.

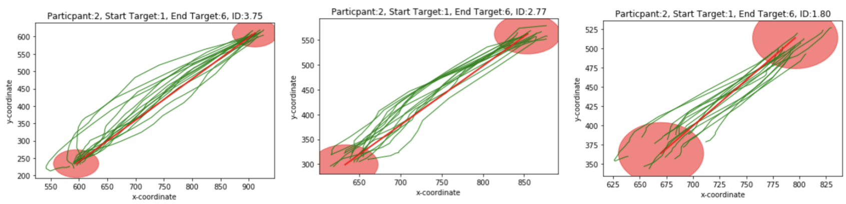
Figures 5.12a and 5.12b depict the plots for the trials for participants 2 and 6, respectively. Once again, we are looking at the movement from target 1 to 6 across all ID's. For all ID's participant 6's plots are closer to the red line and target centres than 2's, indicating that this users motion is close to a straight path. Whether the user did this intentionally or not cannot be inferred without a qualitative interview. However, since the average time difference between the two participants is nearly 40 seconds, we can say with some degree of confidence that this participant has made a conscious effort to move the cursor along a straight path. It also appears that participant 6 has made more of an effort to click at the centre of the target. This effort is especially visible for ID's 2.77 and 1.80. For the same ID's in the case of participant 2, we can see the plots start and end relatively further away from the red line and target centre.

Table 5.7: *Avereage test completion time for each participant across all three ID's.*

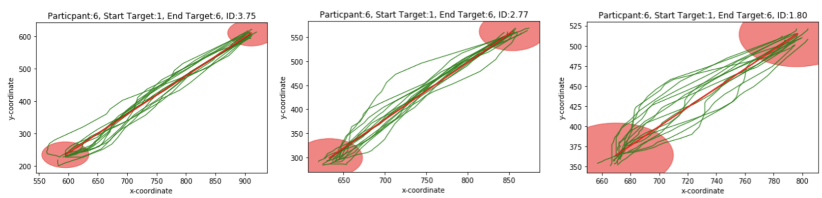
Participant	Average Test Time (in s)
1	88.13
2	80.17
3	87.90
4	91.21
5	82.33
6	119.48
7	91.79
Total Average	91.57
Standard Deviation	12.06

So similar to test #4, we have two types of users here. One that gives preference to accuracy (participant 6) and the other that has gone for efficiency (participant 2). In general, for this test, the plots for rest of the subjects have shown a behaviour similar to that of participant 2. In that, when constrained by the size of the target for the higher ID's, the motion will start and end close to the target centre. But as the target size increases and distance between the targets decrease (and the ID decreases), the users tend to put less effort into clicking on the centre. The plots for all participants are given in the appendix Section A.4 (only for motions from target 1 to 6).

This test has provided with another way of showing the interaction behaviour of different types of users. With more participant trials we could potentially uncover even more types of user behaviour. For example, a user that is extremely dexterous could perform this test with a high accuracy and low test time. At the same time a user with a shivering hand (eg. people of old age) would not able to move the mouse in a straight line and click on the centre of the target even with a very slow rate of motion. Understanding how a wider variety of users perform on this task could help train machine learning models to gather the necessary data to adapt to all kinds of scenarios with different user groups.



(a) *Participants 2's cursor movements. For ID's 3.75 and 2.77 the plots are far from the red line and target centre, indicating that the motion does not follow a straight line. For ID's 2.77 and 1.80, the plots begin and end successively further away from the centre of the target.*



(b) *Participants 6's cursor movements. For ID's 3.75 and 2.77 the plots are close to the red line and target centre, indicating that the motion is close to a straight path. For ID's 2.77 and 1.80, the plots begin and end very close to the centre of the target.*

Figure 5.12: *Cursor movements for participants 2 and 6 for all trials when moving from target 1 to 6 for each ID.*

6 Conclusion

6.1 Summary

The primary goal of this project was to design and implement a comprehensive and flexible test suite of representative pointing tasks that would aid in understanding the dynamics of pointing motions, as well give us an insight into interaction behaviours. First we conducted some background reading on previous implementations to endpoint and mouse motion predictions to gain an insight into what metrics and techniques are important when trying to implement machine learning models. We then move on to explain the design of the test suite and the inspiration behind each test. Using the implemented suite of tests, we conducted participant trials and captured mouse data from 7 test subjects. We then evaluated the gathered data using an array of visualization techniques to highlight the most interesting aspects and behaviours observed from each test.

For test #1, we implemented a simple discrete pointing task using one target at a time. We used this test to dissect each part of the mouse motion from the starting point to a target and compared the nature of the movements across different index of difficulties (ID). Test #2 was another discrete pointing task, where we had multiple targets (1 main target to be clicked on and 4 distractor targets) laid out linearly, either horizontally or vertically to replicate the layout of linear interface elements, such as menus and tabs. The most interesting observation from this test was that despite designing the test as a Fitts' task, we observed effects from the Steering Law that gave us an interesting insight into the users' motion with a mouse. We observed in the vertical layout of the test, subjects crossed into the distractor targets more often from the closest starting point, going against the trivial assumption that as the distance reduces, accuracy increases. In test #3, we implemented a version of Fitts' one-dimensional reciprocal task. Using this test, we used techniques from manual control theory to assess the velocity and acceleration of one-dimensional cursor movements by plotting these using phase space and Hooke plots, respectively. It was in this test, where we realized that the sampling rate of the equipment used for the test was too low to attain smooth velocity and acceleration data. To account for this, we tried to smooth our initial mouse data using a Savitzky-Golay filter. After doing this, the phase space plots improved drastically and looking at these plots, we were able to identify the two different strategies (open loop and closed loop) that the participants used to click back and forth between the targets. Even with the filtered data however, there was still a considerable amount of noise. When plotting acceleration in the Hooke plots, this noise greatly amplified which made it difficult to comment on the specific behaviours in the acceleration of the movements. Test #4 was a simple reciprocal pointing task in which there were two identical targets connected buy a tunnel the same width as that of the targets. We confirmed the effect of the Steering Law using this test by observing that the time to move through the tunnel increased as the width of the tunnel decreased. Test #5 was another tunnel task, this time comprising of multiple 90-degree turns to emulate interaction with interface elements like cascading menus. This test revealed the users' tendency to cut through the turns diagonally as the distance to the target reduces and also, that a user is likely to move in an optimal straight line path through the tunnel if it is wide enough. Test #6 was a multi-directional tapping task, similar to the one described in ISO9241 standard. From this test, we observed that users are less likely to put effort into clicking on the centre of the target if they are not constrained by the target area, as the target size increases.

Many of the behaviours we observed from these tests have drastic implications when designing machine learning models for endpoint and motion dynamics prediction. For example in test #2, when we observed that for certain trials, users were crossing through the distractors to get to the target for short distances, we would need to appropriately train a machine learning model to adapt to such an anomaly to improve the accuracy and reliability of the model. In the case of the tunnel tests, where we saw users crossing the tunnel boundaries to get to the target quicker, if we were to design a model to predict endpoints in a menu system that comprised of moving through similar tunnels, then we would have to train the model appropriately to adjust to the inaccuracies of the user but still be able to correctly predict their intent. Each test on its own can be analyzed in numerous different ways. Perhaps with more participant trials coupled with better a apparatus setup, we can use this exact test-suite to reveal much more interesting motion dynamics and user behaviour.

6.2 Reflection

This project has taught me a great deal about using a whole array of visualization techniques to extract valuable information about interaction behaviour from nothing but a raw data set comprising primarily of cursor and target coordinates. It has also taught me how to process data appropriately to derive hidden information about the dynamics of motion and also adjust to issues like low sampling. Although, I believe the analysis of the dynamics of cursor motion was hindered due to the lack of a more sophisticated screen with a higher refresh rate. This was discovered when first plotting the phase space plots. I initially thought that applying the savgol filter would solve this issue completely. However, there was still a considerable amount of noise in the data even after applying the filter. This successively amplified the noise in the derived velocity and acceleration values. I chose to run the experiments on my laptop for the sake of convenience as it was easier to gather participants to come perform the tests at my home where I could assure a very controlled environment. But if I were to repeat the experiment, I would've looked for better alternative apparatus. Regardless, conducting participant trials made me realize the importance of having a robust set of tests and a concrete data collection plan to keep the experiments as controlled as possible and to make sure the instructions given to each participant and their understanding of the tasks were identical.

Although a lot was achieved in this project, my initial goal was to use the gathered data to also implement a model capable of endpoint prediction. My plan was to train a model using kinematic template matching (described in Section 2.2.2). Then in the second semester, I was enrolled in the Deep Learning course and I hoped to use what I learned from this module to perhaps implement a neural network instead. A recurrent neural network can be used for sequence output tasks like time-series prediction. Using such a model, it may have been possible to achieve endpoints prediction as well as predict the dynamics of the motions like velocity and acceleration. However, by the time data collection had started, it was mid-March and the focus then shifted towards the dissertation write-up. From discussions with the project supervisor, it was also deduced that creating such a neural network would not be a trivial task as there aren't pre-trained neural networks capable of such tasks and perhaps creating such a model could be a whole another project of its own. Furthermore, as the situation with COVID-19 worsened, I had to abruptly leave the UK to return home, by which point it did not seem feasible to spend further time working on the project to build a neural network model.

6.3 Future work

If I was to continue working on this project, the most obvious next step would be to create a recurrent neural network from scratch, capable of endpoint and motion dynamics prediction. Aside from this, there is also the option of using this test suite with many different types of pointing apparatuses. For example, the results we would obtain by performing these tests using a touch pad or joystick would be much different than what we have attained here. Conducting the experiments with an entirely different and more focused group of participants would also potentially yield different results. If we were to conduct the trials with people of old age or a group that suffers from hand tremors, we could gather data to observe their motion dynamics with different pointing apparatuses. We could then use the data to train models to adapt to such users, thereby greatly increasing the ease and quality of interaction for them.

There is also scope to implement these tests in a virtual environment and perform the tasks using a VR headset and joystick. We could design the tests in a Unity environment and introduce haptic feedback and perform the tasks using force-feedback gloves. I am currently an applicant for a PhD program at the university, and my project proposal is titled 'Modelling Dynamic Real-world Systems as Tactile Representations with Dexmo Haptic Force-feedback Gloves'. My proposal aims to use the Dexmo gloves that the university has invested in, to create complex virtual systems from simple models like buttons and levers by first understanding their interaction dynamics in the real-world, and then using participant trials to gather data to map the real-world interaction to a virtual interaction space. Using the set of tests that I have created here and implementing them in a virtual environment is something that I would definitely see myself doing if I am to secure a place in the program.

A Appendix

A.1 Consent Forms and Demonstration Protocol

Participant Consent Form: Mouse Tracking Tests

Purpose
Mouse pointing is a dominant interaction technique. Studies show that mouse pointing comprises up to 65% of our desktop computer usage and that our mouse usage is 3-5 times that of our keyboard usage. But the pointing process is rarely of importance. Rather the target we want to point at is the main objective. Sometimes, when we use interface elements like dropdown menus, scroll bars or video-playback scrubber, we may overstep the border assigned to these elements, which can result in some mild consequences as a result of this human error. These consequences can however, hinder the user experience and delay the user intent. This project aims to achieve mouse endpoint/target prediction by creating a machine learning model that knows an endpoint in advance of its delivery by a mouse-click. With this, we could also increase the efficiency of mouse pointing considerably by predicting the user intent and controlling the interface elements to adjust to errors in mouse movement and positioning.

Background information
The machine learning model will aim to implement a technique called kinematic template matching, which will treat the unfolding velocity profile of a pointing movement as a 2-D stroke gesture and will use template matching to predict the endpoint based on prior observed movements. The model and technique selections used are subject to change as development progresses. However, the mouse data collected through this experiment will be vital in training any model selected.

Target audience
This experiment is open to all persons without any disease or illness that hampers their ability to grip and move a mouse.

What will you be doing?
The test-suite consists of 6 pointing tasks. You will be given a demonstration of a task and then be asked to perform it yourself using the screen and mouse in front of you. After you complete the task, the demonstration for the next task will be given and so on until you have completed all 6 pointing tasks.

How long will it take?
The whole test-suite will take approximately 15-20 minutes to complete. Each test itself can be completed in a matter of seconds, however, they are repeated many times so that more data can be gathered.

Will you receive compensation?
Unfortunately, as this is an undergraduate dissertation project, there are limited resources and a lack of funding. As a result of which it is not possible to provide any compensation for your efforts.

Please note that it is not your performance that will be evaluated. The mouse data collected will only be used for training the machine learning model. You may withdraw from the experiment at any time without prejudice, and any data already recorded will be discarded. Feel free to ask any questions or address any concerns you may have at any stage of the experiment. If you have any further queries or wish to express any comments relating to the experiment, please take a note of the following emails:

Anith Manu Ravindran
2341228r@student.gla.ac.uk

Prof. Roderick Murray-Smith *(Project Supervisor)*
roderick.murray-smith@glasgow.ac.uk

I have read this information sheet, and agree to voluntarily take part in this experiment:

Name: MANU RAVINDRAN

Signature:

Email: manu.ravindran@gmail.com

Date: 21-Mar-2020

Will you receive compensation?
Unfortunately, as this is an undergraduate dissertation project, there are limited resources and a lack of funding. As a result of which it is not possible to provide any compensation for your efforts.

Please note that it is not your performance that will be evaluated. The mouse data collected will only be used for training the machine learning model. You may withdraw from the experiment at any time without prejudice, and any data already recorded will be discarded. Feel free to ask any questions or address any concerns you may have at any stage of the experiment. If you have any further queries or wish to express any comments relating to the experiment, please take a note of the following emails:

Anith Manu Ravindran
2341228r@student.gla.ac.uk

Prof. Roderick Murray-Smith *(Project Supervisor)*
roderick.murray-smith@glasgow.ac.uk

I have read this information sheet, and agree to voluntarily take part in this experiment:

Name: Raidha Fayaz

Signature:

Email: raidhafayaz04@gmail.com.

Date: 12/3/2020

Will you receive compensation?
Unfortunately, as this is an undergraduate dissertation project, there are limited resources and a lack of funding. As a result of which it is not possible to provide any compensation for your efforts.

Please note that it is not your performance that will be evaluated. The mouse data collected will only be used for training the machine learning model. You may withdraw from the experiment at any time without prejudice, and any data already recorded will be discarded. Feel free to ask any questions or address any concerns you may have at any stage of the experiment. If you have any further queries or wish to express any comments relating to the experiment, please take a note of the following emails:

Anith Manu Ravindran
2341228r@student.gla.ac.uk

Prof. Roderick Murray-Smith *(Project Supervisor)*
roderick.murray-smith@glasgow.ac.uk

I have read this information sheet, and agree to voluntarily take part in this experiment:

Name: Sindhu Manu Ravindran

Signature: Sindhu

Email: sindh.manu@gmail.com

Date: 21/03/2020

Will you receive compensation?
Unfortunately, as this is an undergraduate dissertation project, there are limited resources and a lack of funding. As a result of which it is not possible to provide any compensation for your efforts.

Please note that it is not your performance that will be evaluated. The mouse data collected will only be used for training the machine learning model. You may withdraw from the experiment at any time without prejudice, and any data already recorded will be discarded. Feel free to ask any questions or address any concerns you may have at any stage of the experiment. If you have any further queries or wish to express any comments relating to the experiment, please take a note of the following emails:

Anith Manu Ravindran
2341228r@student.gla.ac.uk

Prof. Roderick Murray-Smith *(Project Supervisor)*
roderick.murray-smith@glasgow.ac.uk

I have read this information sheet, and agree to voluntarily take part in this experiment:

Name: HASHMAT ALI

Signature:

Email: hashmat_ali@hotmail.co.uk

Date: 10/03/2020

Will you receive compensation?
Unfortunately, as this is an undergraduate dissertation project, there are limited resources and a lack of funding. As a result of which it is not possible to provide any compensation for your efforts.

Please note that it is not your performance that will be evaluated. The mouse data collected will only be used for training the machine learning model. You may withdraw from the experiment at any time without prejudice, and any data already recorded will be discarded. Feel free to ask any questions or address any concerns you may have at any stage of the experiment. If you have any further queries or wish to express any comments relating to the experiment, please take a note of the following emails:

Anith Manu Ravindran
2341228r@student.gla.ac.uk

Prof. Roderick Murray-Smith *(Project Supervisor)*
roderick.murray-smith@glasgow.ac.uk

I have read this information sheet, and agree to voluntarily take part in this experiment:

Name: Gouri Venu

Signature:

Email: gouri1998@hotmail.com

Date: 12/03/2020 .

Will you receive compensation?
Unfortunately, as this is an undergraduate dissertation project, there are limited resources and a lack of funding. As a result of which it is not possible to provide any compensation for your efforts.

Please note that it is not your performance that will be evaluated. The mouse data collected will only be used for training the machine learning model. You may withdraw from the experiment at any time without prejudice, and any data already recorded will be discarded. Feel free to ask any questions or address any concerns you may have at any stage of the experiment. If you have any further queries or wish to express any comments relating to the experiment, please take a note of the following emails:

Anith Manu Ravindran
2341228r@student.gla.ac.uk

Prof. Roderick Murray-Smith *(Project Supervisor)*
roderick.murray-smith@glasgow.ac.uk

I have read this information sheet, and agree to voluntarily take part in this experiment:

Name: Isha Prasad

Signature:

Email: ahsi.prasad@gmail.com

Date: 12-8-2020

Will you receive compensation?
Unfortunately, as this is an undergraduate dissertation project, there are limited resources and a lack of funding. As a result of which it is not possible to provide any compensation for your efforts.

Please note that it is not your performance that will be evaluated. The mouse data collected will only be used for training the machine learning model. You may withdraw from the experiment at any time without prejudice, and any data already recorded will be discarded. Feel free to ask any questions or address any concerns you may have at any stage of the experiment. If you have any further queries or wish to express any comments relating to the experiment, please take a note of the following emails:

Anith Manu Ravindran
2341228r@student.gla.ac.uk

Prof. Roderick Murray-Smith *(Project Supervisor)*
roderick.murray-smith@glasgow.ac.uk

I have read this information sheet, and agree to voluntarily take part in this experiment:

Name: MOHAMMED KHABIR HUSSAIN

Signature: KHABIRH

Email: KHABIR_H@HOTMAIL.CO.UK

Date: 12/03/2020

Demonstration Protocol

This section gives a step-by-step description of how I intend to conduct the demonstration for each test. Rather than give a demonstration of all 6 tests in one go, each test will be demonstrated first and then the participant will be asked to perform the task. This will allow for small breaks in between each test so that the participants hands and fingers don't get strained/cramped and will also keep the instructions fresh in their memory.

Initialization

1. Open file “Mouse Tracking.py”.
2. Enter participant number given by the variable “participant”.

Test-1

3. Start running test-1 (test-ID 1).
4. Points to highlight:
   a. Click the target as quickly and accurately as possible.
   b. Mouse cursor will reset to the default position after clicking the target.
   c. There are 10 targets.
   d. An iteration consists of clicking on all 10 targets.
   e. Target positions will not change.
   f. There are 3 sets of target sizes.
   g. Target size will become smaller after 10 iterations.
5. Click the “esc” button to exit after 1 iteration.

Test-2

6. Start running test-2 (test-ID 1).
7. Points to highlight:
   a. Red buttons are distractors and the green button is the target.
   b. Click the target as quickly as possible.
   c. Mouse cursor will reset to a default position after clicking the target.
   d. Default position will change after all 5 targets have been clicked.
   e. There are 4 sets of default positions.
   f. An iteration consists of clicking on the 5 targets for each of the 4 default positions.
   g. There are 10 iterations.
   h. There are 2 variations of the test.
8. Click the “esc” button to exit after 1 iteration.
9. Run test-ID 2 briefly to show the second variation.

Test-3

10. Start running test-3 (test-ID 1).

11. Points to highlight:
    a. One distractor and one target.
    b. Time limit of 90secs.
    c. Click back and forth between the changing targets as quickly and accurately as possible.
    d. 3 pairs of targets of different sizes and distances.
12. Click the "esc" button to exit after 15 seconds of demonstration.
13. Run test-ID 2 and 3 briefly to show the other variations.

Test-4

14. Start running test-4 (test-ID 1).
15. Points to highlight:
    a. One distractor and one target.
    b. Time limit of 90secs.
    c. Move cursor back and forth between the changing targets as quickly and accurately as possible without crossing the tunnel borders.
    d. Do not click.
    e. Tunnel gets smaller after 90secs.
    f. 3 tunnel sizes.
16. Click the "esc" button to exit after 15 seconds of demonstration.

Test-5

17. Start running test-5 (test-ID 1.1).
18. Points to highlight:
    a. Target is the green button at the end of the tunnel.
    a. 90 second time limit.
    b. Move the red circle through the tunnel as quickly and accurately as possible and click the green target at the end.
    c. 2 tunnel shapes.
    d. 3 sizes of each tunnel. Tunnel will get bigger after 90secs.
    e. Tunnel shape will change after the first 3 variations of first tunnel shape.
19. Click the "esc" button to exit after 15 seconds of demonstration.
20. Run test-ID 2.1 for 15 seconds to demonstrate the second tunnel shape.

Test-6

21. Start running test-6 (test-ID 1).
22. Points to highlight:
    a. Target is red circle.
    b. Order of targets do not change.

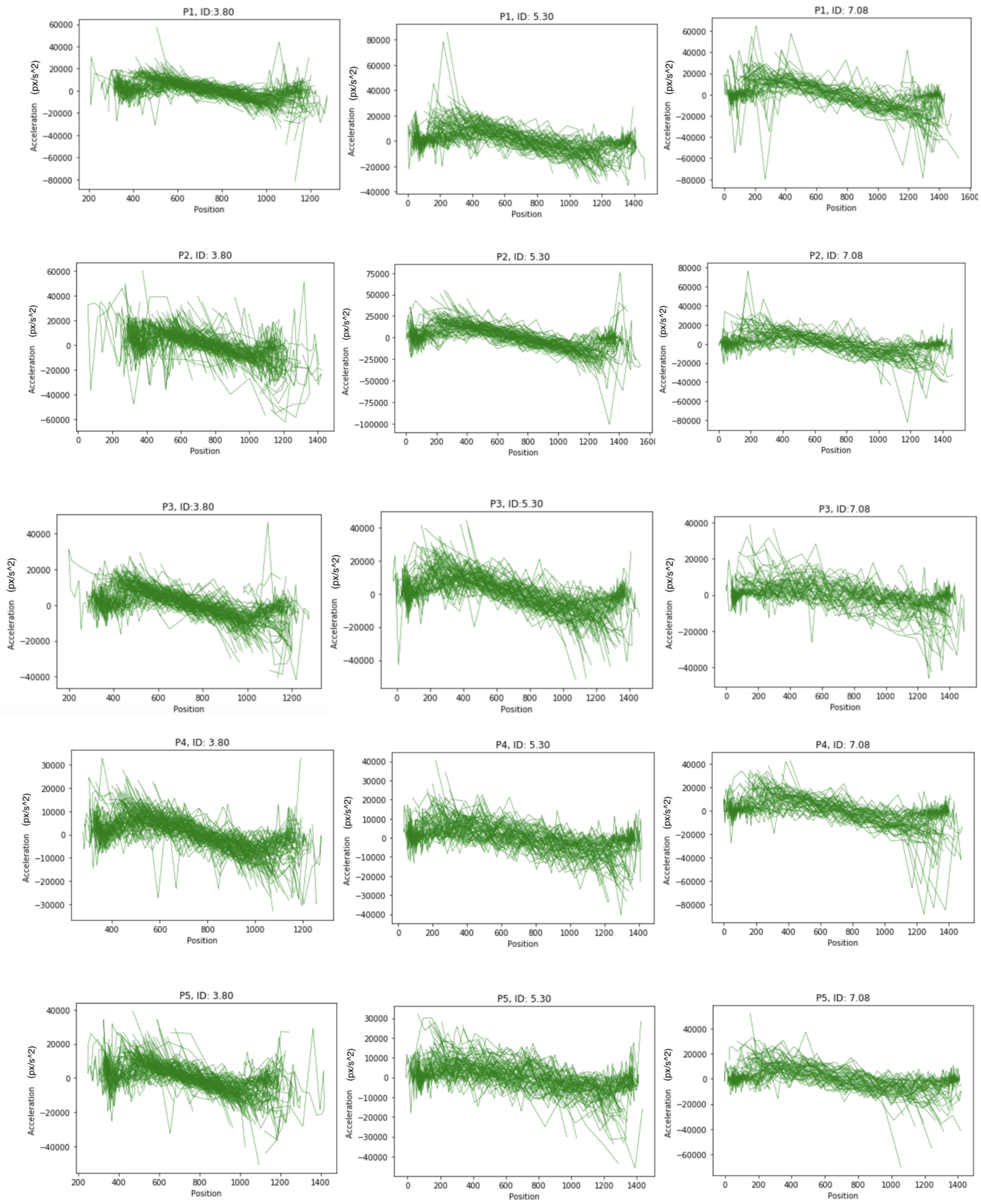
c. An iteration is completed after all circles are highlighted and clicked.

d. There are 15 iterations.

e. 3 different variations. Circles will get bigger and closer after 15 iterations of each variation.

23. Click the “esc” button to exit after 1 iterations.
24. Run test-ID 2 and 3 to show the other 2 variations.

A.2 Test #3 Hooke Plots



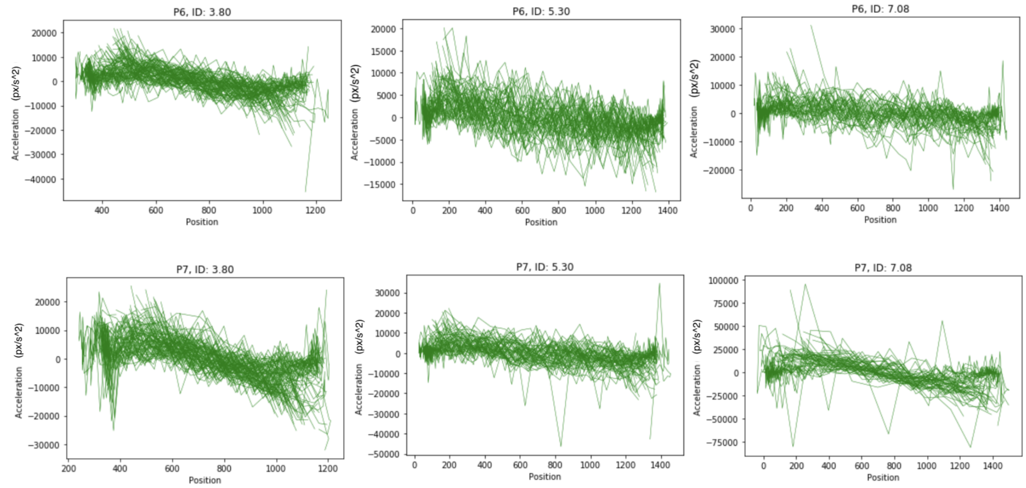
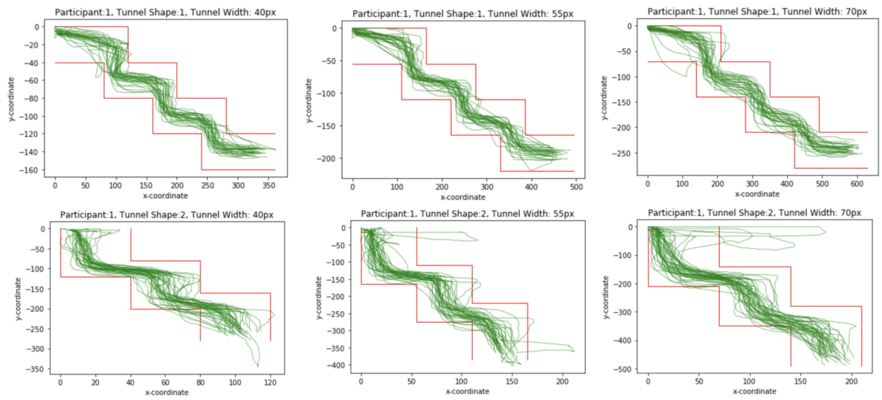
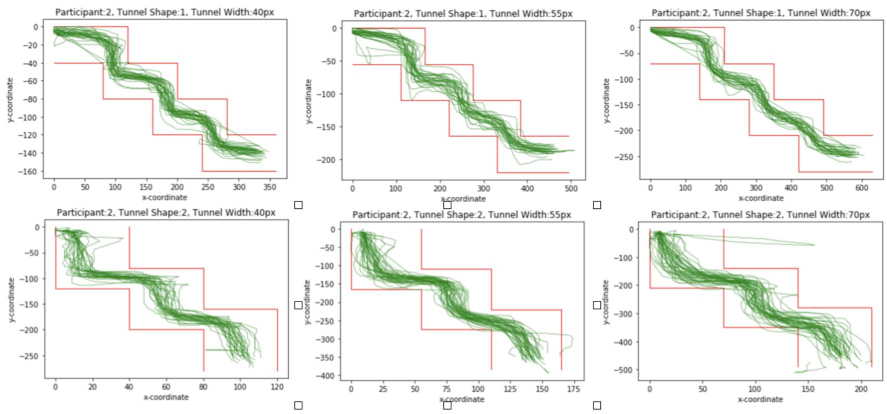
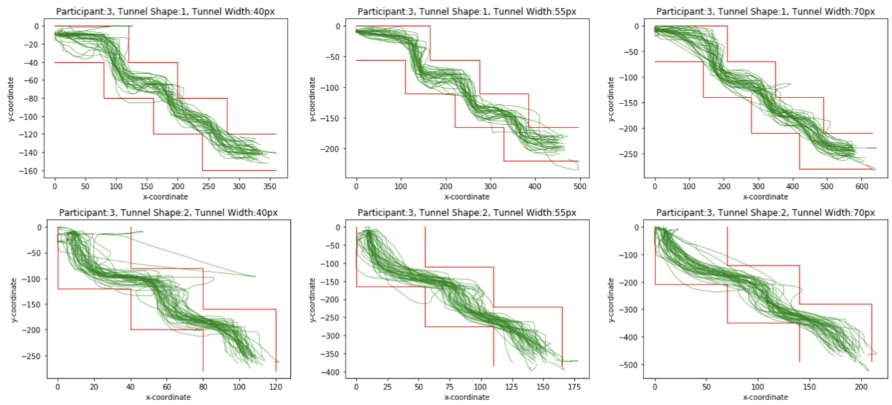


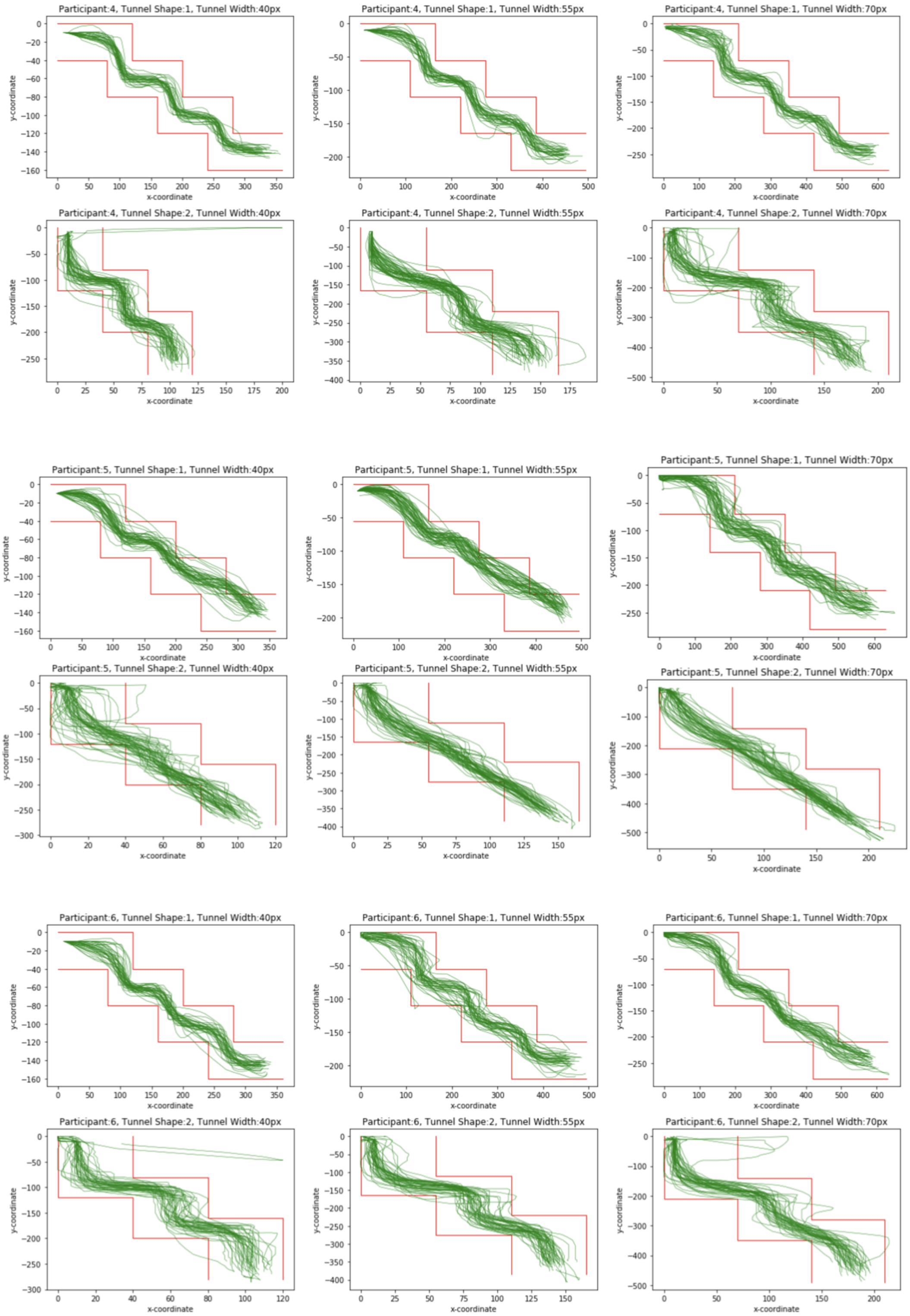
Figure A.2: *Hooke plots of acceleration over position (after applying Savitzky-Golay filter) for all trials for each participant for each ID.*

A.3 Test #5 Tunnel Plots








Participant:4, Tunnel Shape:1, Tunnel Width:40px
Participant:4, Tunnel Shape:1, Tunnel Width:55px
Participant:4, Tunnel Shape:1, Tunnel Width:70px
Participant:4, Tunnel Shape:2, Tunnel Width:40px
Participant:4, Tunnel Shape:2, Tunnel Width:55px
Participant:4, Tunnel Shape:2, Tunnel Width:70px
Participant:5, Tunnel Shape:1, Tunnel Width:40px
Participant:5, Tunnel Shape:1, Tunnel Width:55px
Participant:5, Tunnel Shape:1, Tunnel Width:70px
Participant:5, Tunnel Shape:2, Tunnel Width:40px
Participant:5, Tunnel Shape:2, Tunnel Width:55px
Participant:5, Tunnel Shape:2, Tunnel Width:70px
Participant:6, Tunnel Shape:1, Tunnel Width:40px
Participant:6, Tunnel Shape:1, Tunnel Width:55px
Participant:6, Tunnel Shape:1, Tunnel Width:70px
Participant:6, Tunnel Shape:2, Tunnel Width:40px
Participant:6, Tunnel Shape:2, Tunnel Width:55px
Participant:6, Tunnel Shape:2, Tunnel Width:70px
x-coordinate
y-coordinate

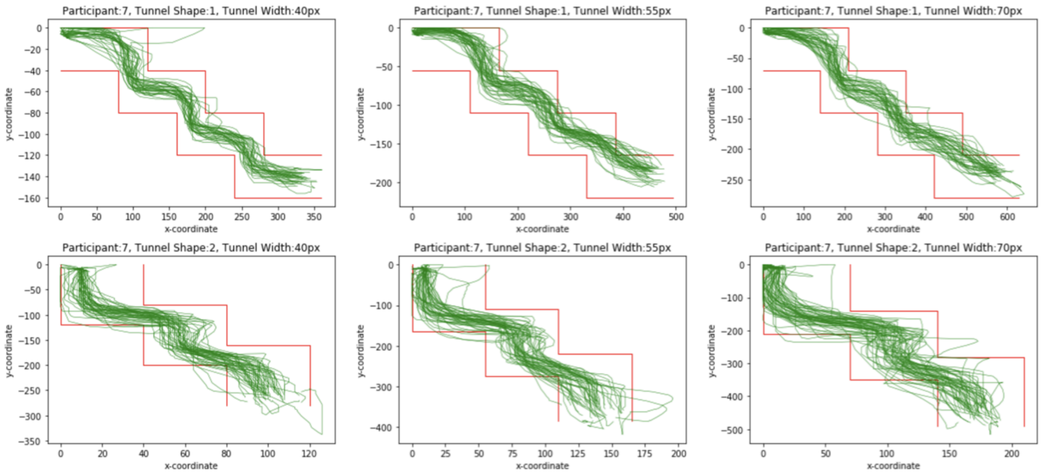
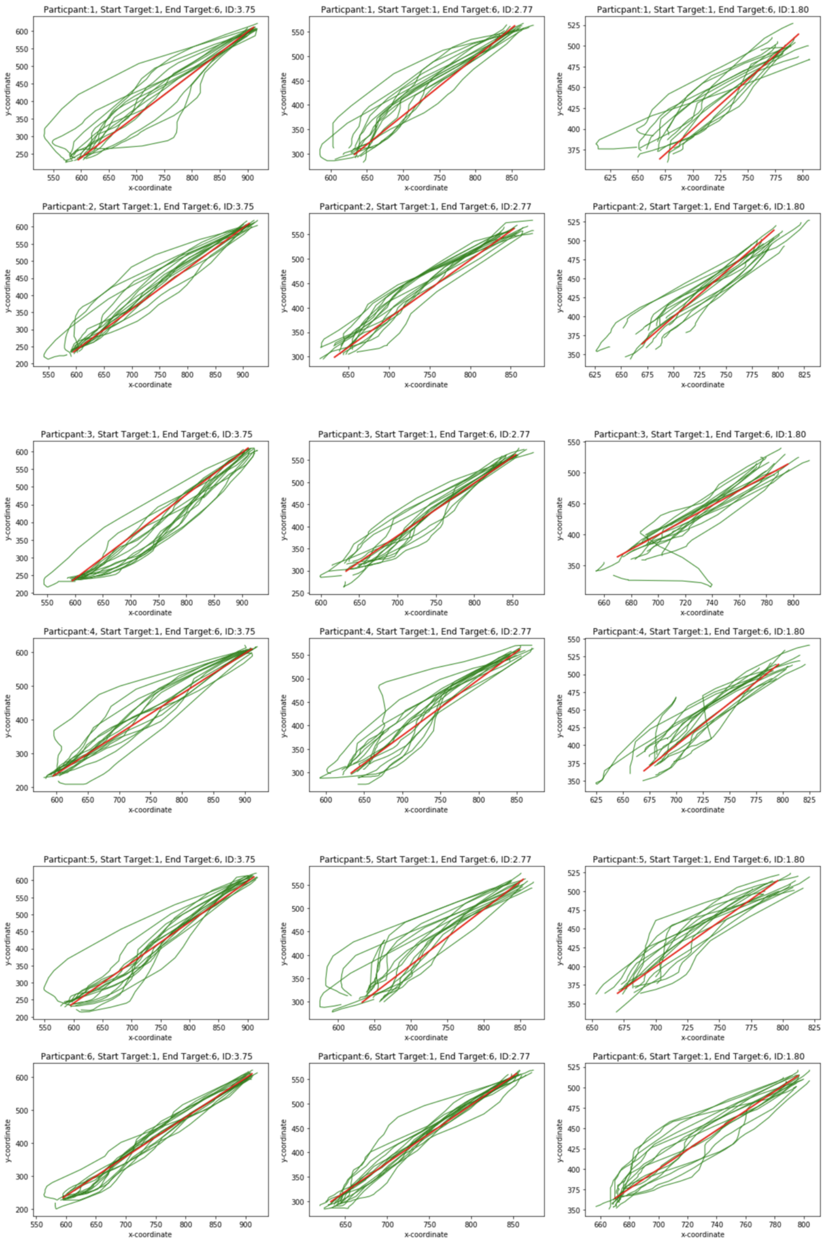


Figure A.5: *Cursor movement through both tunnel shapes for each participant for each tunnel width.*

A.4 Test #6 Plots



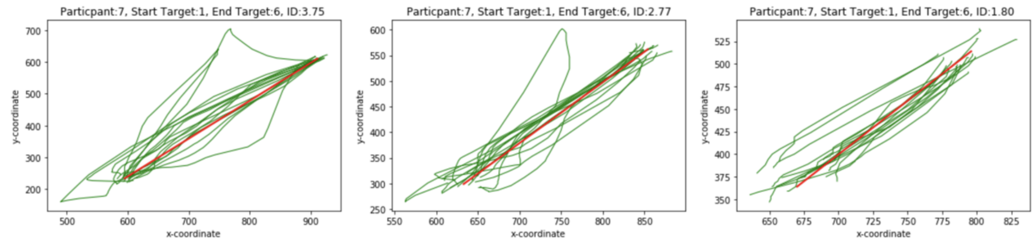


Figure A.7: *Cursor movements for each participants for all trials when moving from target 1 to 6 for each ID.*

Bibliography

Accot, J. and Zhai, S. (1997), 'Beyond Fitts' Law: Models for trajectory-based HCI tasks', *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems* p. 295–302.

Asano, T., Sharlin, E., Kitamura, Y., Takashima, K. and Kishino, F. (2005), 'Predictive interaction using the Delphian Desktop', *Proceedings of the 18th annual ACM symposium on User interface software and technology (UIST 05)* pp. 133–141.

Chang, C.-H. J., Amick, B. C., Menendez, C. C., Katz, J. N., Johnson, P. W., Robertson, M. and Dennerlein, J. T. (2007), 'Daily computer usage correlated with undergraduate students musculoskeletal symptoms', *American Journal of Industrial Medicine* **50**(6), 481–488.

Fitts, P. M. (1954), 'The information capacity of the human motor system in controlling the amplitude of movement.', *Journal of Experimental Psychology* **47**(6), 381–391.

Flash, T. and Hogan, N. (1985), 'The coordination of arm movements: an experimentally confirmed mathematical model', *The Journal of Neuroscience* **5**(7), 1688–1703.

ISO (2019), 'ISO/TS 9241-411:2012 Ergonomics of human-system interaction — part 411: Evaluation methods for the design of physical input devices', `https://www.iso.org/standard/54106.html`.

Johnson, P. W. (1993), 'Office ergonomics: Motion analysis of computer mouse usage', *Proceedings of the American Industrial Hygiene Conference and Exposition (AIHCE '93)* pp. 12–13.

Keuning, H., Oirschot, V. and Houtsma, A. (2001), 'Cursor displacement and velocity profiles for targets in various locations', *In Proceedings of Eurohaptics* pp. 108–112.

Kieslich, P. J. and Henninger, F. (2017), 'Mousetrap: An integrated, open-source mouse-tracking package', *Behavior Research Methods* **49**(5), 1652–1667.

Lank, E., Cheng, Y. N. and Ruiz, J. (2007), 'Endpoint prediction using motion kinematics', *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 07)* p. 637–646.

Laubheimer, P. (2019), 'Accot-Zhai Steering Law: Implications for UI design', `https://www.nngroup.com/articles/steering-law/`. Last accessed: 2020-03-12.

Mackenzie, I. S. and Buxton, W. (1992), 'Extending Fitts' Law to two-dimensional tasks', *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI 92)* pp. 219–226.

Mathôt, S., Schreij, D. and Theeuwes, J. (2011), 'OpenSesame: An open-source, graphical experiment builder for the social sciences', *Behavior Research Methods* **44**(2), 314–324.

Meyer, D. E., Smith, J. E. K., Kornblum, S., Abrams, R. A. and Wright, C. E. (2018), 'Speed—accuracy tradeoffs in aimed movements: Toward a theory of rapid voluntary action', *Attention and Performance XIII* p. 173–226.

Müller, J., Oulasvirta, A. and Murray-Smith, R. (2017), 'Control theoretic models of pointing', *ACM Transactions on Computer-Human Interaction* **24**(4), 1–36.

Pasqual, P. T. and Wobbrock, J. O. (2014), 'Mouse pointing endpoint prediction using kinematic template matching', *Proceedings of the 32nd annual ACM conference on Human factors in computing systems (CHI 14)* p. 743–752.

Perret, J. and Vander Poorten, E. (2018), 'Touching virtual reality: A review of haptic gloves', *ACTUATOR 2018; 16th International Conference on New Actuators* pp. 1–5.

Ruiz, J. and Lank, E. (2009), 'Effects of target size and distance on kinematic endpoint prediction', *Technical Report CS-2009-25. Cheriton School of Computer Science, University of Waterloo* .

Savitzky, A. and Golay, M. J. E. (1964), 'Smoothing and differentiation of data by simplified least squares procedures.', *Analytical Chemistry* **36**(8), 1627–1639.

Takagi, R., Kitamura, Y., Naito, S. and Kishino, F. (2002), 'A fundamental study on error-corrective feedback movement in a positioning task', *Proceedings of the 5th Asia Pacific Conference on Human Computer Interaction (APCHI 2002)* p. 160–172.

Wright, C. E. and Lee, F. (2013), 'Issues related to HCI application of Fitts' Law', *Human–Computer Interaction* **28**(6), 548–578.