Point n Click: A Glove-Based Cursor Control Device

Camron Ong
De La Salle University
Manila, Philippines
camron_evan_ong@dlsu.edu.ph

Abstract

A clear and well-documented LaTeX document is presented as an article formatted for publication by ACM in a conference proceedings or journal publication. Based on the "acmart" document class, this article presents and explains many of the common variations, as well as many of the formatting elements an author may use in the preparation of the documentation of their work.

CCS Concepts

• **Hardware → Sensors and actuators**.

Keywords

pointer device, mouse, human interface device, accelerometer, gyroscope, magnetometer, capacitive sensors, glove, arduino, transfer function, cursor

ACM Reference Format:
Camron Ong. 2018. Point n Click: A Glove-Based Cursor Control Device. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX).* ACM, New York, NY, USA, 12 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Many devices, such as the widely used mouse, trackpad, pen and tablet pair, and VR controllers, are used for device interaction and pointer control. While these are popular and effective, they can pose challenges and limitations depending on the context. For example, mice generally require a decently sized flat surface, trackpads can be small or limited in size, while VR controllers are more optimized for immersive VR environments rather than the 2D space and screens, and also cost a decent amount of money.

There could also be several user examples, such as presentations, some creative work, or hands-free computer navigation, where the said devices can be limited. Inefficiencies may arise in the context of presentations, wherein multiple devices could be necessary to deliver the presentation. Mobility limitations within users could also lead to accessibility concerns in general.

To address these challenges, this study aims to extend an alternative interface, a glove-based pointing device, designed to provide an intuitive interface and a way to control the pointer through the use of various hand motions and simple gesture recognition. Unlike traditional input devices, the proposed glove-based pointer will use hand motions and simple pointing recognition, enabling interaction without reliance on physical surfaces.

Given this goal, the study will aim to maintain most ease of plug-and-play of the existing input devices while adding new features on top. Specifically, the following will be kept in mind while redesigning the previous version of the glove interface from [9] and combining some features from [20]:

- The pointer glove should still be lightweight and portable.
- It should operate without requiring extensive calibration or external hardware apart from the glove and any receivers.
- The glove should be able to handle mouse-like inputs, namely pointer manipulation and clicking, without requiring physical contact with a 2D surface.
- Latency should be generally low to allow smooth input-to-feedback translation.
- The solution should also work in general computer navigation.

To evaluate this, several research questions can be constructed:

- **RQ1** Does the proposed glove-based pointer provide the necessary functionality for general computing and accessibility purposes?
- **RQ2** How does the device's usability compare to traditional input methods and the previous version in terms of accuracy, reliability, and ease of use?
- **RQ3** How does the device perform compared to the original?
- **RQ4** Is the system intuitive for users?

2 Original Interface

The mouse is already a golden standard used for moving the pointer across the 2D screen. Moving to wireless mice also allows for a non-in-person interaction with the computer, wherein users can move the pointer while standing from a distance. That said, these traditional mice require a surface, usually flat, underneath the mouse to function properly. This can be troublesome when standing at a distance, especially in situations such as presenting from a distance - that requires both a clicker and a pointer, or simply using the computer from a distance.

A possible alternative to pointer movement is a gesture-based system that uses accelerometers, and gyroscopes, as shown by [9]. The solution focused on translating the gestures to pointer movement, specifically for scenarios of in-person presentation control. While innovative, the prototype turned out to be slower than the device it was being compared to, the CheerPod [7]. While this could be due to hardware limitations on the sensor used, the glove prototype also lacked some features available for the Cheerpod and consequently, the mouse, such as clicking capabilities, and wireless signal transmission.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
Conference acronym 'XX, Woodstock, NY
© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

3 Proposed Improvements and Redesigned Interface

The study aims to extend the previous glove-based pointer system by addressing its lack of certain mouse functionalities and improving cursor control. The redesigned system adds direct pointer manipulation, mouse-like clicking capabilities, and improved sensors. These changes also address the limitations of traditional mice, which require a flat surface to function efficiently. Gesture recognition will allow users to move the cursor to predefined screen areas by pointing, providing near-instantaneous repositioning similar to eye tracking. Additionally, adding mouse-like interactions ensures a more familiar user experience, making the glove a viable alternative to traditional pointing devices.

3.1 Initial Setup

The main Arduino Pro Micro [3] attached to the glove will still be the primary controller and pointer manipulator in the project. The motion sensing will also use the current MPU-6050 [10] sensor as a baseline to capture hand orientation and estimate hand gestures. This will be compared to the MPU-9250 [15], which has an added magnetometer, which could improve the movement or gesture-to-pointer translation accuracy while combating drift. Calibration will likely be used to measure the distance of the glove from the screen, which will contribute to the calculation of the user's point gesture to a spot on the screen. If, due to external circumstances, the sensor cannot be acquired, a stand-in HMC5883L [2] will be used alongside the MPU-6050 (or 6500) instead.

Infrared transmitters and receivers [14], [13] will be used for finger gesture recognition. It should be noted that this will be limited to only the normal pointing gesture (pointing finger extended). These are rated to have a range of about 10 meters at maximum. Capacitive touch sensors, specifically the TTP223B [12], will also be added to simulate mouse clicks. For example, touching the thumb and index finger could be mapped to the left mouse button, while touching the thumb and right finger could map to the right mouse button. The pin out of this is shown in Figure 1

An additional setup will be required to communicate the IR signals from the transmitter to the receivers. Given this, there are two options - 1) use a separate Pro Micro board or something similar to simulate the Mouse library, or 2) communicate the signals instead via Bluetooth. The second option was opted for to keep the Mouse movements limited only to one device, thus potentially improving stability. Given this, the ESP32 lineup, specifically the ESP32-WROOM [21], was chosen and interfaced with the IR receivers, as shown in Figure 2. Meanwhile, an HC-05 Bluetooth module [8] was added to the other board for wireless connection and communication.

3.2 Design Iterations

Preliminary testing of each iteration was conducted and was used as the basis for improvements made on the firmware and hardware choices or designs in the project. One main issue was the touch capacitive sensors. The initial prototype had intended for three capacitive touch sensors, mapping to left, middle, and right clicks, respectively. These were initially set up on a breadboard and could be tested individually, hence why the setup seemed fine. That said,

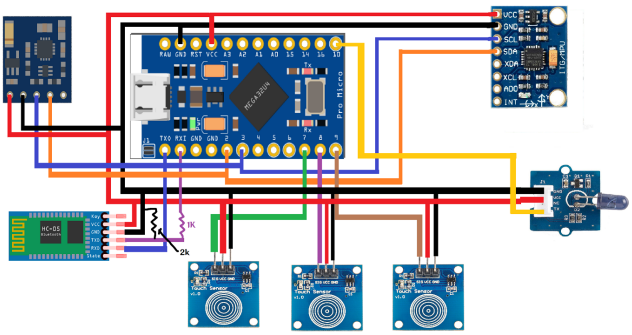

Figure 1: Sample Schematic of the Arduino Pro Micro controller (top-middle) connected to the HMC5883L compass (top-left), MPU-6050 (top-right), Bluetooth module (bottom left), IR Emitter (middle-right), and the Capacitive Touch Sensors (Bottom-row).

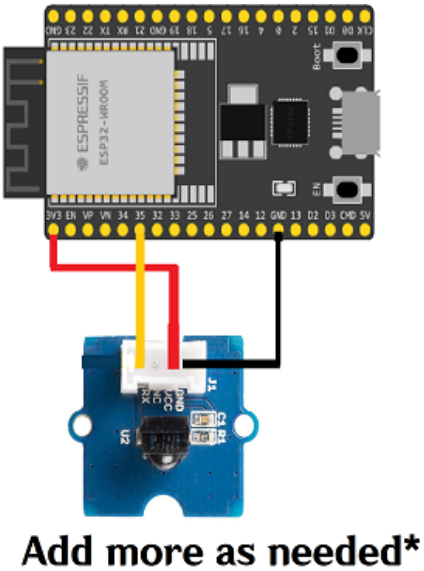

Figure 2: Sample Pinout Schematic of an ESP32-WROOM (top) connected with an IR Receiver. Multiple Receivers will be added as necessary.

when they were attached to the glove, having the three side-by-side felt unnatural. This was because the sensors themselves were too large, and they could block each other's boards when the user's fingers were curled.

Apart from this, the drift from the gyroscope and accelerometer was also investigated. Ideally, this would be fixed by adding a magnetometer and manually adding fusion functions for the three devices. That said, upon testing, popular Attitude and Heading Reference System (AHRS) libraries that do this, such as Mahony [5] and Madgwick [4], require a lot of storage space on the Arduino Pro Micro. An alternative could be to manually implement a lightweight

version, but this led to difficulties in program implementation and integration. Thus, a small alternative of simply re-setting the MPU based on the magnetometer's readings after a set period of time was opted for.

3.3 Final Setup

As discussed above, the glove-based pointer system underwent multiple refinements during development. For instance, after some testing, it was found that 9-axis sensors, including the MPU-9250 and other similar boards, were generally being phased out. Along the same line, getting a fake 9250 was also common, wherein the chip would only house an MPU-6500 and would lack the magnetometer. In fact, this occurred during the material-gathering portion of the project, in which a lookalike was purchased and obtained instead of the original intended chip. Thus, it was decided that the final design would reuse the previous MPU-6050 plus a stand-in HMC5883L to act as the missing magnetometer, as discussed in the previous subsection.

Along the same lines, as mentioned earlier, only two capacitive sensors were used. The initial version also had them unshielded, which led to unintentional registered taps. To address this, modifications in the form of a grounding shield was added on one side of each sensor. This caused them to be less sensitive and only receptive in one area, thus improving click accuracy but also limiting its sensitivity to one side's specific area.

The IR transmitter was the most stable piece and was left as is.

The Bluetooth module that was added in the final design also allows for easier upgrading to a full wireless solution. Wireless, in this case, refers to the removal of the cable attaching the Pro Micro board on the glove to the computer. This could lead to better usability but will not be explored in this project. Pictures of the final setup of the glove and receivers are shown in Figures 3 and 4.

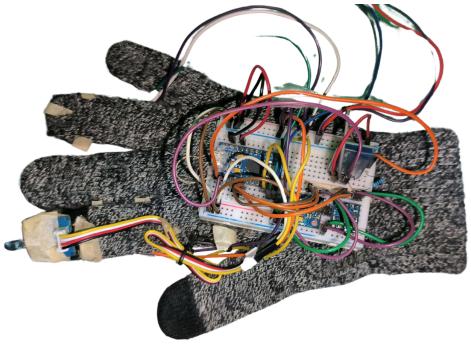

Figure 3: Final Physical Setup of the Updated Glove.

3.4 Summary of Firmware and Hardware in the Final Setup

The final system consists of the following firmware and hardware components:

3.4.1 Firmware and Libraries Used.

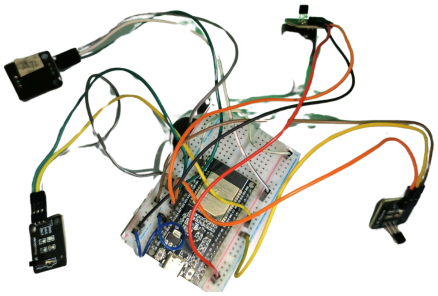

Figure 4: Physical Setup of the Receivers Connected to an ESP32-WROOM.

- **Mouse** [6]: Allows certain boards to control cursor movement of the connected computer. Positioning is based on the relative position of the cursor rather than the absolute position.
- **I2C Device Library** [18]: A collection of classes for streamlined communication for I2C devices. This library was specifically used to interface the MPU6050 with the microcontroller. Technically, the library also supports the HMC5883L.
- **Adafruit HMC5883 Unified** [1]: A sub-library that interfaces and abstracts data from the HMC5883L. This was chosen over the I2C Device Library version because sensor readings are converted into standardized units, thus leading to more simplified integration.

3.4.2 Hardware Used.

- **Arduino Pro Micro** [3]: Serves as the main microcontroller for processing sensor inputs and controlling the mouse pointer.
- **ESP32-WROOM** [21]: Bluetooth and WiFI capable microcontroller. Handles IR receiver signals and transmits data to the Pro Micro via Bluetooth.
- **TTP223B** [12]: Touch capacitive sensors that detect touch-based gestures to simulate mouse clicks.
- **HMC5883L** [2]: A magnetometer used to help correct drift in hand orientation tracking.
- **MPU6050** [10]: A 6-axis motion sensor used for tracking hand movement and angle.
- **IR Transmitter** [14]: Mounted on the glove to send signals for pointing.
- **IR Receivers** [13]: Placed around the screen to detect IR signals and determine pointing direction.

3.5 Scope and Limitations of the Device

Given this setup, the glove will be limited to only left and right-click inputs. Scrolling and actions like dragging will not be considered due to hardware design limitations and choices. Hence, the testing facility will also be created to match only these features - manipulating the mouse pointer and selecting or clicking on a target. Additionally, while wireless data transmission will be implemented for the IR setup, a full wireless device will also be out of scope for the project. This is because moving to a wireless solution would

require additional design considerations and could affect usability, such as requiring on-board power and charging.

4 Methodology

4.1 Experimental Design

A general-purpose testing facility was created to evaluate the proposed glove design. This would replicate general computer usage, wherein a user would move their mouse across the screen to navigate and click through buttons at random distances and from a set of shapes, similar to Fitts' Law [11]. Each test consists of five trials or shapes to click through. A sample screenshot is shown in Figure 5.

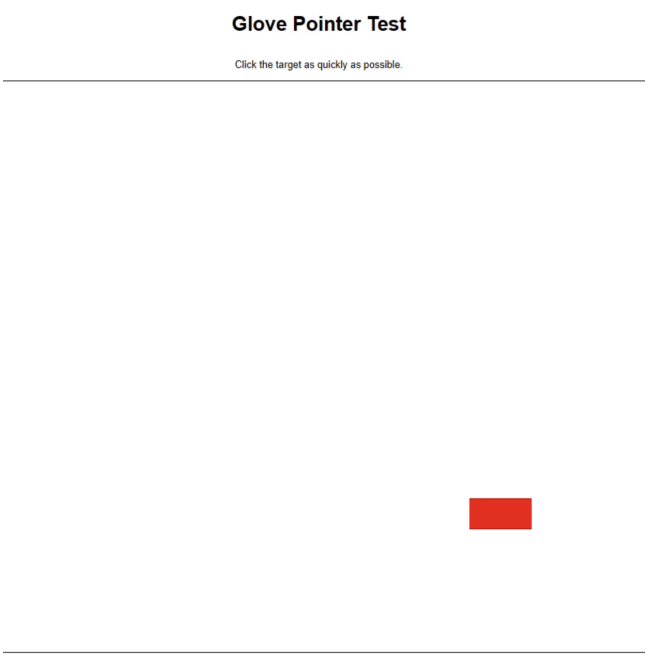

Figure 5: Sample Screenshot of the Testing Facility with a Rectangle Target generated.

A 13-inch MacBook Pro (2017) with a screen resolution of 1440 by **900** was used for testing. The mouse sensitivity is set to 4/10, while the trackpad sensitivity was set to 6/10. Both of these were constant throughout the tests. Normal room lights were also kept on during the tests, and the users sat at a comfortable distance, mimicking a normal desk sitting position.

10 participants were involved in the study. Each participant would do the task about five times each. The tests were carried out using the (1) baseline mouse, again set to a constant sensitivity, and (2) the final version of the glove, as mentioned above and shown in Figure 3. Note that while no formal testing was done for the previous iterations, the final design incorporates the earlier mentioned refinements to enhance usability and reduce errors.

4.2 Performance Metrics

The variables measured include total time (TT), which represents the time required to accomplish each button click, and the corresponding distance (D) between each button, and the button size (W). This aligns with Fitts' Law [16], which predicts the movement time (MT) required to move to a target based on D and W. Specifically, this is expressed as:

$$MT = a + b * \log_2(\frac{2D}{W})$$

Additionally, the Index of Difficulty (ID) with respect to the distance and width can be calculated as:

$$ID = \log_2(\frac{D}{W} + 1)$$

And the respective throughput per device is calculated using the overall total time:

$$TP = \frac{ID}{TT}$$

This theoretical framework is used to compare the efficiency of the glove-based pointer system to the traditional mouse by calculating the expected movement time based on target distance and size and comparing it to the measured movement time [16].

Other than Fitts' Law metrics, the errors, such as missed or extra clicks, are also recorded. Additionally, hover time is tracked to measure how long the cursor remains inside the target area before clicking, while correction time accounts for overshooting—when the cursor enters the target but exits before successfully clicking. The delay between entering the target to clicking the target is also logged, both in terms of first entry and most recent entry. To analyze movement patterns, the full cursor path is logged, allowing for the computation of path efficiency, i.e., how directly the cursor moves toward the target. With this, extra features such as the velocity, total overshooting time, the length of the path taken and its respective ratio compared to the straight-line distance, and the extra movement in percentage can also be calculated.

Additionally, a comparative analysis was performed on the usability and efficiency of input methods. This analysis, including a T-test [22] which assessed statistical significance, determined whether the glove-based pointer system offered comparable performance to a traditional mouse in terms of movement time, path efficiency, error rate, hover time, and correction time if applicable. While the mouse and glove-based pointer system differ in fundamental interaction mechanics, this comparison aimed to assess whether the glove system can serve as a feasible alternative for pointer-based tasks, particularly in scenarios where traditional mouse usage may be impractical.

4.3 Model Creation

Using the data gathered, regression models were also created, with variables such as task completion time and error counts as the target. This was done using the other variables, like distance, the device type, or the shape of the target button.

To explain each further, the model for task completion time was created using Ordinary Least Squares (OLS) regression [19]. This attempts to calculate and find the best fitting line given the data

gathered. Separate models were initially created to test the features, and any that showed a high Variance Inflation Factor (VIF) were removed. A regression combining the two datasets was also created.

Additionally, Poisson regression [17] models were created for the error modeling. Poisson regression is commonly used for count data, making it suitable for predicting the number of errors based on various task-related factors. Since error counts are discrete and often skewed toward lower values, this model estimates how the measured variables influence the likelihood of errors occurring.

To account for overdispersion—where the variance of the error counts exceeds the mean—a Negative Binomial regression model was also tested. However, after evaluation, the Poisson model was retained as it provided better overall performance in predicting error counts for both the glove and traditional mouse datasets.

5 Results and Discussion

A total of 51 sessions, resulting in 255 trials, were gathered for the mouse. For the glove, 61 sessions were recorded, totaling 305 trials. However, two glove sessions had to be removed due to minor bugs caused by the random-target generator of the testing facility. Additionally, trials with optimal path ratios higher than 15, along with error counts greater than 10, were also removed. The final counts of each was 250 for the mouse and 292 for the glove.

5.1 Fitts' Law Modelling

Table 1 shows the regression fitting Fitts' Law across the various tests. A graph showing the models of MT versus ID for the two devices is also shown in Figure 6 and 7

Table 1: Fitts' Law Regression Results and Throughput

Metric	Mouse	Glove
	Regression Results	
Intercept	0.4456 ($p < 0.0001$)	-0.6235 ($p = 0.251$)
ID Coefficient	0.0662 ($p < 0.0001$)	0.4467 ($p < 0.0001$)
R^2	0.095	0.131
	Throughput (bits/s)	
Mean Throughput	8.47	3.07

Both models show a significant positive relationship between Index of Difficulty (ID) and movement time (MT), aligning with Fitts' Law predictions. This confirms that as tasks become more difficult, completion time increases for both input devices. This also makes sense as the conversion and translation from either glove input or mouse input is inherently the same. Both directly manipulate the mouse pointer, which implies both would be affected by "higher difficulty" tasks.

However, a difference appears when comparing the two input devices' coefficients. Specifically, the glove has a much higher coefficient for ID (0.447) compared to the mouse (0.066), indicating that task difficulty affects the glove much more than the mouse. Additionally, the lower R^2 values, 0.095 for the mouse and 0.131 for the glove, suggest that other factors also influence completion time beyond just task difficulty. This result makes sense, as the task itself involves not only moving the pointer to the target but also clicking on the target itself, which could add delays, as to be discussed later.

Apart from this, the average throughput of the two devices was 8.47 bits/second and 3.07 bits/second for the mouse and glove, respectively. This is a large difference, with the mouse having over double the throughput of the glove. That said, this large difference could be due to two reasons. The first is the inherent difference in familiarity with the two devices. Many of the participants were avid long-term mouse users, but the glove had a learning curve that required getting used to. The second possible reason is because of how the inputs translate to pointer movements. The movements of a mouse and other "2 dimensional" input devices can be translated easier since it is direct 2D to 2D. This means that stopping the pointer, or making a sharp movement to make the pointer move in a different direction is much easier. On the other hand, the glove relies on the gyroscope's tilt, which can be harder to translate from 3D to 2D, as well as being harder to control, especially when the users are not particularly familiar with the device.

Another aspect was that the IR setup was not used much. This could be due to unfamiliarity or that the receivers were too far or inconvenient to point to. The translated positions of the IR to the screen were also set to the corners of the screen, which was generally far from the actual testing facility's test area size mentioned earlier.

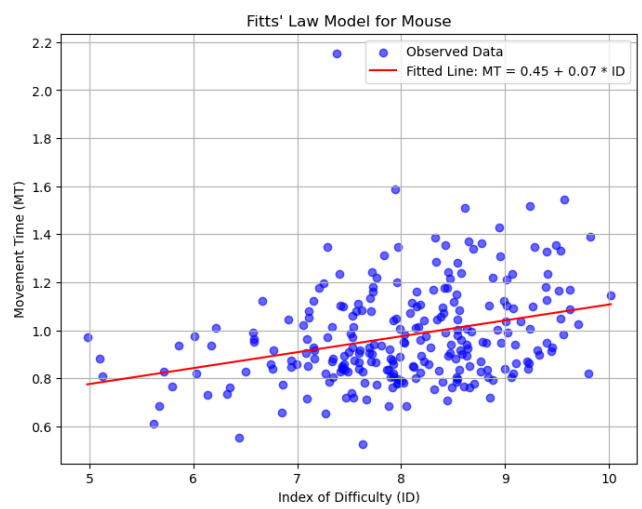

Figure 6: Graph of Fitts Law Model for the Mouse

5.2 T-test for Metrics

Table 2 shows the comparison of various movement and error metrics between the mouse and glove tests. Boxplots showing the distributions are also included in the appendix. Overall, all three categories have several significant performance differences between the two devices.

For movement, the mouse had faster total times and, correspondingly, higher velocity despite the similar average distances per trial. This directly indicates that accomplishing tasks with the mouse is

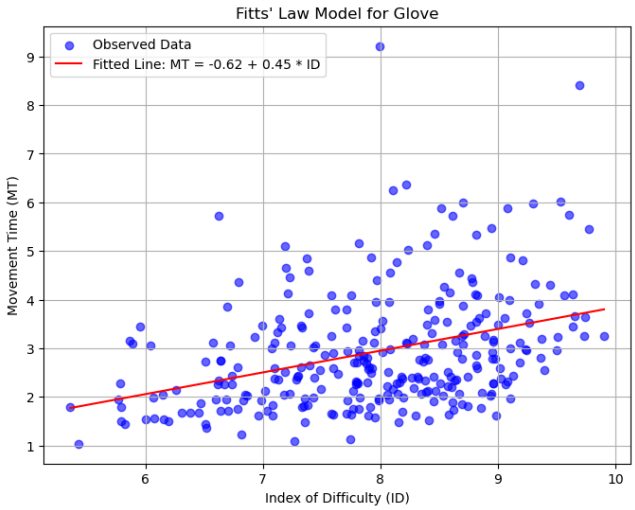

Figure 7: Graph of Fitts Law Model for the Glov

Table 2: Comparison of Performance Metrics between Mouse and Glove

Movement				
Metric	**Mouse**	**Glove**	**t-stat.**	**p-value**
Total Time	0.9765	2.9295	-27.3785	0.0000
Distance	813.76	812.84	0.0293	0.9766
Velocity	543.56	178.29	21.10	0.0000
Optimal Path Ratio	3.7455	3.7701	-0.1330	0.8943
Error Metrics				
Metric	**Mouse**	**Glove**	**t-stat.**	**p-value**
Total Misclicks	0.0360	0.1747	-4.7006	0.0000
Total Overshoots	0.1520	0.1884	-1.0351	0.3011
Time-Based Error Handling				
Metric	**Mouse**	**Glove**	**t-stat.**	**p-value**
Hover Time	0.3484	0.5096	-5.4307	0.0000
Total Overshoot Time	0.0377	0.2924	-5.6669	0.0000
First Entry Click Delay	0.3424	0.7349	-7.3324	0.0000
Most Recent Click Delay	0.2953	0.4101	-5.3591	0.0000

faster than with the glove, which matches the throughput calculations seen earlier. That said, both devices had an average path ratio of about 3.7, indicating both devices took a path that was more than three times longer than what was optimal, though this is likely just due to user inefficiency.

As for the error metrics, the glove did have significantly more total misclicks compared to the mouse, averaging about 0.175 compared to 0.036 misclicks per trial, thus resulting to a p-value of 0.0. On the other hand, both devices had a fair share of overshoots at 0.152 and 0.188 for the mouse and glove, respectively. This means that there was no significant difference, as the p-value only resulted in 0.301.

Finally, for the time-based metrics, all four measured values had significant differences between the mouse and the glove. Hover time, overshoot time, and delays between entering and clicking on a target were all significantly lower for the mouse compared to the glove. This generally aligns with the finding earlier, wherein the overall completion time and throughput of the mouse was higher than the glove.

Overall, while movement efficiency remained largely unchanged, the glove introduced significantly more errors and required longer adjustment times before users could complete a trial. These findings suggest that while the glove-based input method may be functional, its current form results in greater difficulty in precise control and selection, especially with higher ID tasks, as mentioned in 5.1.

5.3 Modelling the Completion Time

Regression models were created to predict the total completion time based on the various features earlier. Initially, the models used all of the features, including those engineered, such as the optimal path ratio. These were eventually removed due to high VIF, thus resulting in strong multicollinearity. Tables 3 and 4 show the breakdown of the regressions and how the various factors contribute to the task modelling.

Table 3: Refined OLS Regression Results for Completion Time (Mouse)

Variable	Coef.	Std. Err.	t-stat.	p-value
Constant	-0.023	0.125	-0.182	0.856
Distance	3.36e-05	2.54e-05	1.323	0.187
Index of Difficulty (ID)	0.096	0.014	6.993	0.000
Extra Movement %	0.022	0.006	3.948	0.000
Hover Time	0.472	0.038	12.522	0.000
Error Count	0.234	0.050	4.697	0.000
Target Shape	-0.014	0.006	-2.092	0.037
R-squared	0.503			
Adjusted R-squared	0.491			
Condition Number	1.33e+04			

Table 4: Refined OLS Regression Results for Completion Time (Glove)

Variable	Coef.	Std. Err.	t-stat.	p-value
Constant	-2.895	0.722	-4.011	0.000
Distance	3.96e-05	0.000	0.284	0.777
Index of Difficulty (ID)	0.601	0.088	6.812	0.000
Extra Movement %	0.124	0.038	3.256	0.001
Hover Time	1.371	0.116	11.792	0.000
Error Count	0.668	0.110	6.086	0.000
Target Shape	-0.075	0.045	-1.659	0.098
R-squared	0.527			
Adjusted R-squared	0.517			
Condition Number	1.22e+04			

The p-values of ID, Extra Movement %, Hover Time, and Error Count show that they are all significant weights in the model. In

particular, ID is the strongest predictor for both devices, with a significant positive coefficient. This aligns with Fitts' Law, as shown earlier, where task difficulty directly affects movement time.

Hover Time has a much larger effect in the glove model, with a coefficient of 1.371 compared to the mouse, which only had 0.472. This suggests that delays due to hovering are more prominent in glove-based interactions. Specifically, the time increases due to more hovering, which is caused by delays in user mouse click input.

Finally, Extra Movement Percentage is also more influential in the glove model, which translates to glove trials having more unnecessary or longer movements. This can also be due to overshooting caused by the difficulties in control mentioned in 5.2. Along the same vein of overshooting, Error Counts have stronger effects on the glove than on on the mouse, which indicates that missed clicks are also harder to correct.

Surprisingly, both the Distance and Target Shape seem to have weak effects for both models. This could also be due to the fact that the two values are technically part of the ID calculation, which would also explain the relatively high condition numbers experienced.

It should also be noted that both models had R^2 values of 0.503 and 0.527, respectively. This means that both models explain only about half of the variance in the task, which is considerably lacking but decent at the same time. This is because there are other external factors due to human and input behavior that are undeniably unaccounted for in the tests. The multicollinearity is also still high, at over 10,000 for both models.

Apart from this, the two datasets can be combined to analyze whether the features behave differently across the two respective devices. Table 5 shows these values, confirming the earlier findings. Specifically, that the mouse is faster than the glove, that ID affects the glove more, and that hover times and the extra movements are bigger issues found for the glove compared to the mouse. Surprisingly, the R^2 improves to about 0.780. This could be due to the larger and more varied data, which, along with device types, can make certain relationships more easily identifiable. The smaller multicollinearity between the variables could also be the factor, which can be seen as the value is only 307 for this graph, compared to the over 10,000 mentioned above.

Graphs showing the overall prediction versus observed total times for the three models are also shown in Figure 9, 8, and 10.

5.4 Modelling Missed Clicks

The missed clicks done with both devices can also be investigated and modeled. The main features are shown in Table 6 and 7, respectively.

The model generated for the mouse performs poorly, as the calculated pseudo-R^2 is very low, at 0.132. Additionally, none of the variables are statistically significant as the p-values are also very high. There is also instabilities for the overshoot time and clicking delays, as the standard errors calculated were at 12 and 16.

As for the glove model, this models the misclicks better, with a pseudo-R^2 of about 0.320. The hover time and the most recent click delay are the most significant. It should be noted that these two values are somewhat related, and equal in many cases.

Table 5: OLS Regression Results for Completion Time (Combined Model)

Variable	Coef.	Std. Err.	t-stat.	p-value
Constant	-0.036	0.523	-0.069	0.945
Index of Difficulty (ID)	0.097	0.058	1.661	0.097
Extra Movement %	0.022	0.024	0.935	0.350
Hover Time	0.473	0.161	2.927	0.004
Error Count	0.232	0.213	1.090	0.276
ID × Device	0.414	0.077	5.362	0.000
Extra Movement % × Device	0.075	0.035	2.178	0.030
Hover Time × Device	0.904	0.183	4.944	0.000
Error Count × Device	0.444	0.228	1.946	0.052
Device Type	-2.190	0.690	-3.173	0.002
R-squared		0.782		
Adjusted R-squared		0.779		
Condition Number		307		

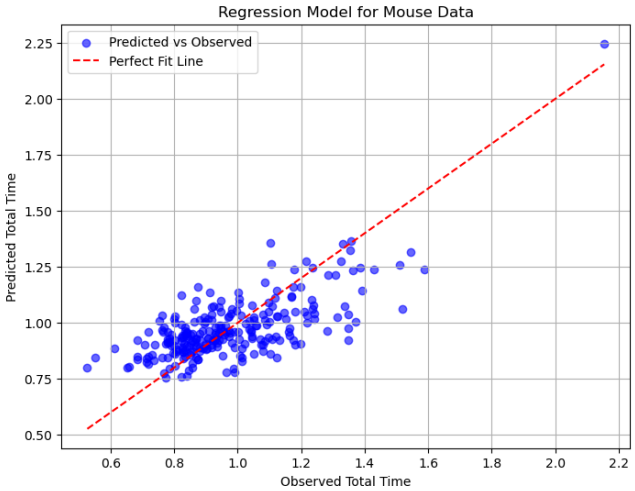

Figure 8: Regression Model for Mouse

This result is also likely due to the low sample size of errors encountered in the observations. In further investigation, only 9 trials out of the 250 had at least one error. In comparison, 51 out of 292 glove trials had at least one misclick. This difference, where the mouse trails had fewer errors and the glove trials had more errors, likely contributed to the difference in the model fits.

Apart from this, the longer hover times and clicking delays for the glove could indicate that the users are ensuring that they are within the target before clicking. That said, the inherent fewer amount of misclicks when using the mouse already shows how users are more comfortable using the mouse and that the glove is outperformed by the golden standard.

Finally, it should be noted that the variables total overshoot time, first entry click delay, and most recent click delay were added to the feature list after doing some trial and error. These engineered features are technically already related to some of the previous

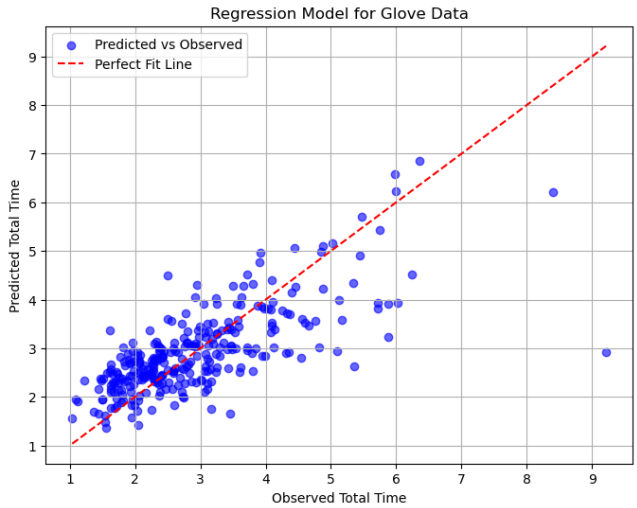

Figure 9: Regression Model for Glove

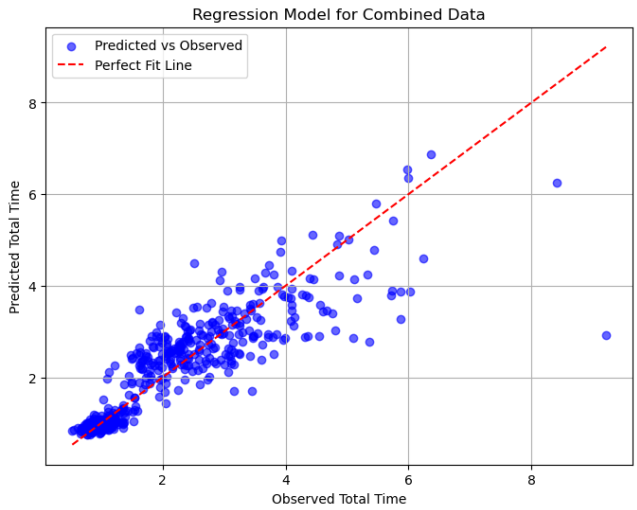

Figure 10: Regression Model for Combined Data

features. That said, adding them appears to have significantly increased the modelling performance for the glove data at the cost of slightly decreasing the performance of the model for the mouse. Hence, it was decided to leave them in for consistency across the results.

The graphs showing the predicted versus the observed error counts is shown in Figure 11 and 12 for the mouse and glove models, respectively. Note that Poisson regression predicts the expected counts given the input values, hence why many of the predictions are non-integer values.

5.5 Performance Insights

The glove is functional for general computer navigation that involves pointer manipulation and simple target selection. However, in terms of accuracy and ease of use, the study has shown that it falls short compared to the traditional mouse in all measured

Table 6: Poisson Regression Results for Error Count - Mouse

Variable	Coef	Std Err	P-value
Intercept	-6.8680	5.505	0.212
distance	-0.0005	0.001	0.645
ID	0.4902	0.571	0.391
extra_movement_percentage	-0.2212	0.357	0.536
hover_time	-5.5222	6.396	0.388
target_shape	-0.0711	0.248	0.774
total_overshoot_time	-0.2235	16.005	0.989
first_entry_click_delay	7.2083	16.515	0.663
most_recent_click_delay	-0.9631	12.778	0.940
Log-Likelihood		-33.779	
Pseudo R-squared		0.1321	
Observations		250	

Table 7: Poisson Regression Results for Error Count - Glove

Variable	Coef	Std Err	P-value
Intercept	-1.8657	2.237	0.404
distance	0.0003	0.000	0.410
ID	0.0632	0.273	0.817
extra_movement_percentage	0.0931	0.109	0.395
hover_time	4.4308	1.700	0.009
target_shape	-0.1026	0.142	0.471
total_overshoot_time	1.6557	2.457	0.500
first_entry_click_delay	-1.3153	2.498	0.598
most_recent_click_delay	-9.4164	2.001	0.000
Log-Likelihood		-99.526	
Pseudo R-squared		0.3199	
Observations		292	

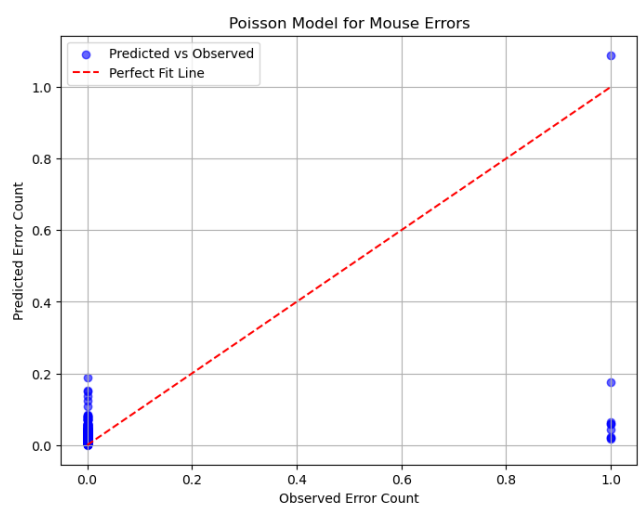

Figure 11: Regression of Total Errors for Mouse

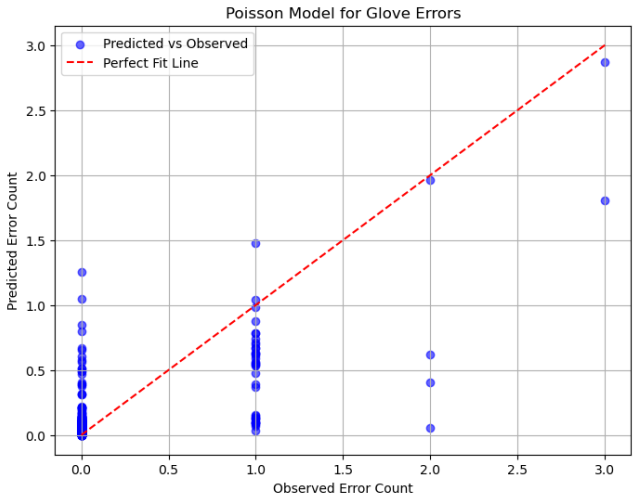

Figure 12: Regression of Total Errors for Glove

aspects. This suggests that while the glove is already usable, further refinements could enhance its overall performance and user experience.

Additionally, while not explicitly measured, it is likely that users would require some time to adjust to the glove's movement and input method. Given that participants only had a few minutes of exposure to the device, their initial performance may not fully reflect its potential with extended use and familiarity. In particular, while the pointer movement latency may be low, the hover times were still higher for the glove compared to the mouse. This indicates that more time is required for the user to accomplish the clicking part of the task, translating to possibly higher cognitive processing time for the glove.

Users also largely preferred to use the normal hand movements and gestures to move the pointer rather than using the IR setup. As stated earlier, this could be due to the lack of familiarity or possibly the inconvenience of suddenly pointing the fingers. Improvements to the transitioning between normal movement and pointing movement could also help lessen this inconvenience.

Apart from that, some drift still occurs. This is largely due to several hardware limitations. As mentioned earlier, the setup of an MPU6050 plus HMC5883L was used instead of a single chip with all three sensors - gyroscope, accelerometer, and magnetometer. Using two separate modules results in a requirement to have an external fusion function, which, unfortunately, due to hardware limitations in flash storage, was not possible. While this does not have much direct impact on short-form tests, long form usage of the glove would definitely require anti-drift mechanisms for better user experience.

Overall, even with the numerous challenges, the glove presents a unique alternative to traditional pointing devices. It offers a hands-free, 2D-surface-independent method of interaction that could be beneficial in scenarios where conventional input devices are less practical. Improved versions could become a niche substitute for the usual input devices used today.

6 Conclusion and Recommendations

6.1 Recommendations

Several improvements could be made to make the glove device more usable as an alternative to the conventional pointer-movement and selecting devices. On the hardware side, moving to full wireless as the Bluetooth module and other Bluetooth board is already available. This could involve swapping the setups of the boards to have the ESP32 connected to the glove, then having the Pro Micro connected with the computer.

Swapping the two boards could also improve or eliminate the effects of drift on the MPU. To explain, the ESP32 has more flash storage, which was a limitation mentioned earlier. Hence, with the ESP32 on the glove instead, the AHRS libraries could then be implemented properly, thus potentially leading to calibration and fusion between the MPU6050 and the HMC5883L. Along the same vein, just using a proper module with all three sensors - the accelerometer, gyroscope, and magnetometer, integrated would likely be the best fix.

Additionally, reducing the size of the touch sensor or area could be investigated. This would make it more natural and easier to move the fingertips, without having a board attached. Potentially, metallic sheets wrapped around the fingertips could be used instead.

Apart from this, adding more IR receivers and improving the IR locations, both physically and where they translate to on screen, or using some form of gesture or pointing angle recognition could be an area to explore. That said, the angle recognition could be difficult to implement, as the main glove relies on the MPU6050's gyroscope as the main controller of the pointer movement.

In terms of testing, increasing the number of trials and potentially diversifying the testing facility can improve the performance of the models on the respective tasks. Other than that, conducting a qualitative assessment could result in a more comprehensive study of how the glove compares with other input methods, such as various pain points annoyances for improvement, or areas where the users could find themselves preferring the glove.

6.2 Conclusion

The new iteration of the glove has shown an avenue for non-in-person computer usage that requires pointer manipulation and target selection, which is evident in the task completion of RQ1. That said, in terms of RQ2 and RQ3, the glove performed worse in all aspects compared to the mouse, namely requiring more time to complete the tasks and also having considerably more missed clicks and overshoots. However, it performed better than the previous iteration in terms of mimicking a mouse, as clicking could now be done. The device was also less intuitive for users, as they appeared to have some difficulties in familiarizing themselves with the new input device for RQ4, compared to their familiarity with a mouse.

The glove's performance does not yet match that of a traditional mouse and has many areas for improvement, particularly in reducing error rates, improving user intuitiveness, and optimizing total task completion time. That said, this study demonstrates its potential as a feasible alternative in specific contexts. It can cater to niche scenarios where traditional mice may be impractical, such as hands-free or non-in-person environments.

Acknowledgments

I would like to thank the DLSU Center for Automation Research (CAR) laboratory for their assistance in providing insights, tools, and solutions to several hardware-based difficulties and issues encountered during the designing phase of the project.

I would also like to thank the previous researchers of Point n Move, who created the initial version of the glove, for being easy to contact and providing the exact model specifications, along with passing on the physical glove, the code that was outputted from the project, and the extra materials that were leftover from their study.

References

[1] Adafruit. n.d. Adafruit HMC5883 Unified Library. https://github.com/adafruit/Adafruit_HMC5883_Unified
[2] Honeywell Aerospace. n.d. HMC5883L Datasheet. https://www.digikey.com/en/htmldatasheets/production/786600/0/0/1/asd2613-r.html
[3] Arduino. 2024. Micro | Arduino Documentation. https://docs.arduino.cc/hardware/micro/
[4] Arduino. n.d. Madgwick Library for Arduino. https://docs.arduino.cc/libraries/madgwick/
[5] Arduino. n.d. Mahony Library for Arduino. https://docs.arduino.cc/libraries/mahony/
[6] Arduino. n.d. Mouse | USB Functions | Arduino Language Reference. https://docs.arduino.cc/language-reference/en/functions/usb/Mouse/
[7] Cheerdots. n.d. *CheerPod Instruction Book.*
[8] HC-05 Developers. n.d. HC-05 Bluetooth Module Datasheet. https://components101.com/sites/default/files/component_datasheet/HC-05%20Datasheet.pdf
[9] S. B. Dy, R. J. O. Encinas, D. J. L. Go, K. C. C. Lasala, B. A. Y. Lu, M. M. Manlises, and J. A. Deja. 2024. Point n Move: Designing a Glove-Based Pointing Device. *arXiv preprint* arXiv:2412.00501 (2024).
[10] ElectronicWings. 2020. MPU6050 Accelerometer and Gyroscope Sensor Guide with Arduino Programming. https://www.electronicwings.com/sensors-modules/mpu6050-gyroscope-accelerometer-temperature-sensor-module
[11] Paul M Fitts. 1954. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology* 47, 6 (1954), 381–391.
[12] FL. n.d. TTP223-BA6 Touch Pad Detector IC Datasheet. https://vakits.com/sites/default/files/TTP223B%20Touch%20Switch.pdf
[13] Generic. n.d. IR Receiver Module for Arduino. https://components101.com/infrared-receiver-tsop1738
[14] Generic. n.d. IR Transmitter Module for Arduino. https://components101.com/infrared-transmitter-and-receiver-pair
[15] TDK InvenSense. 2016. MPU-9250 Product Specification Revision 1. https://invensense.tdk.com/wp-content/uploads/2015/02/PS-MPU-9250A-01-v1.1.pdf
[16] I. S. MacKenzie. 1989. A note on the information-theoretic basis for Fitts' law. *Journal of Motor Behavior* 21 (1989), 323–330.
[17] Paul Roback and Julie Legler. n.d. Beyond Multiple Linear Regression: Chapter on Poisson Regression. https://bookdown.org/roback/bookdown-BeyondMLR/ch-poissonreg.html
[18] J. Rowberg. 2022. I2C Device Library. https://github.com/jrowberg/i2cdevlib
[19] G. A. F. Seber and Alan J. Lee. 1989. *Linear Regression Analysis.* Vol. Series in Probability and Statistics. Wiley.
[20] A. Shih and H. Lee. 2012. Glove Mouse - A wireless, wearable computer pointing device. https://people.ece.cornell.edu/land/courses/ece4760/FinalProjects/f2012/as986_hl525/as986_hl525/index.htm
[21] Espressif Systems. n.d. ESP32-WROOM-32E/32UE Datasheet. https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32e_esp32-wroom-32ue_datasheet_en.pdf
[22] B. L. Welch. 1947. The generalization of 'Student's' problem when several different population variances are involved. *Biometrika* 34, 1-2 (1947), 28–35. doi:10.1093/biomet/34.1-2.28

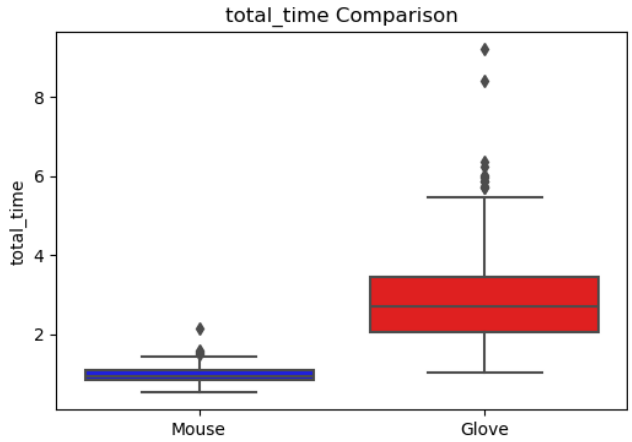

Figure 13: Boxplot of Total Times of Mouse and Glove

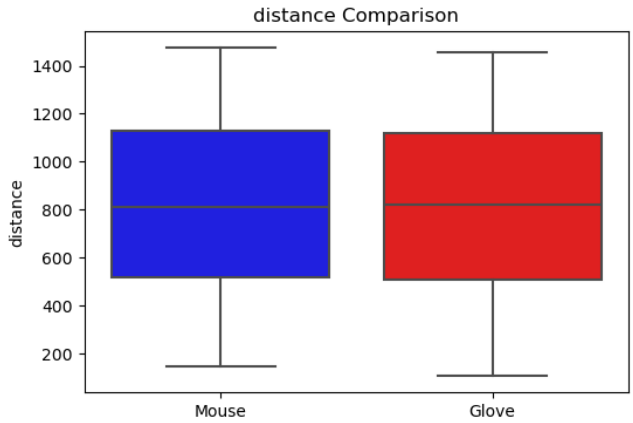

Figure 14: Boxplot of Target Distances of Mouse and Glove

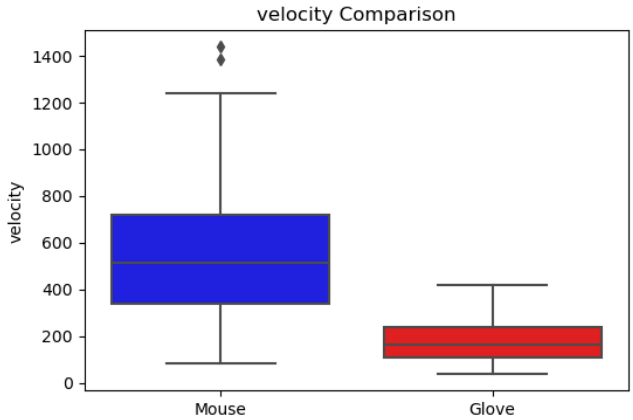

Figure 15: Boxplot of Velocities of Mouse and Glove

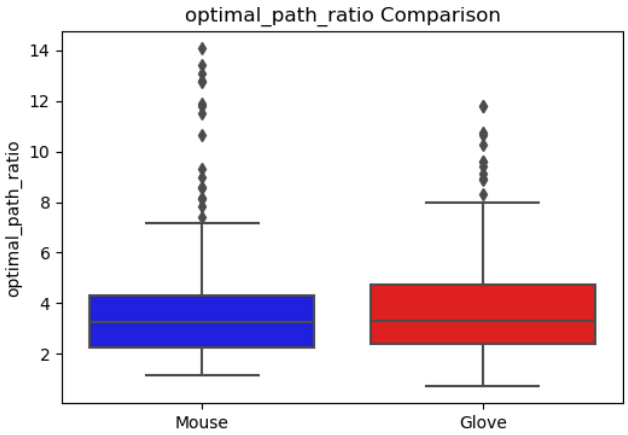

Figure 16: Boxplot of Optimal Path Ratios of Mouse and Glove

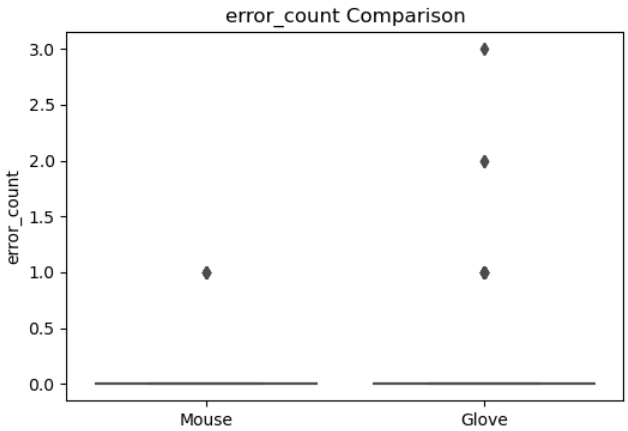

Figure 17: Boxplot of Error Counts of Mouse and Glove

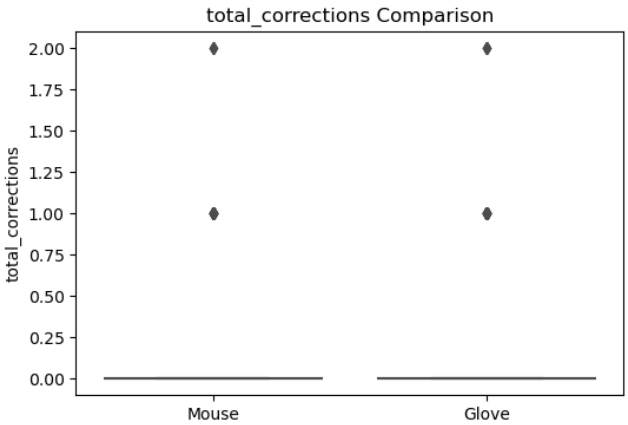

Figure 18: Boxplot of Correction Counts of Mouse and Glove

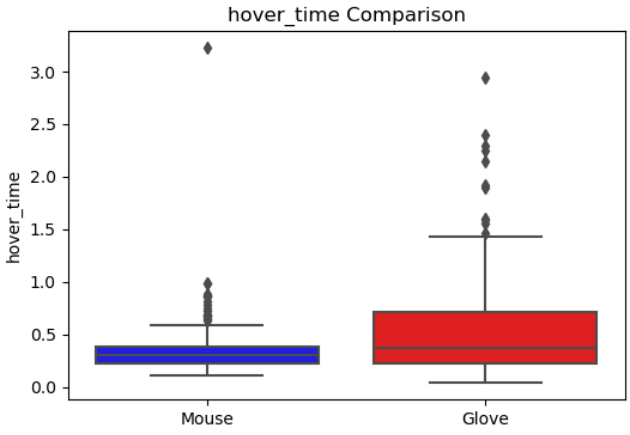

Figure 19: Boxplot of Hover Times of Mouse and Glove

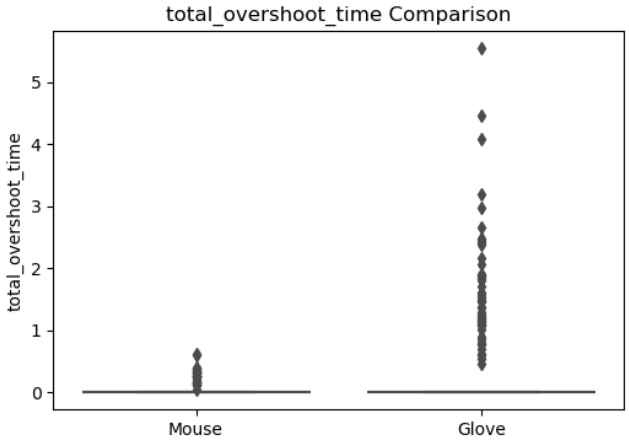

Figure 20: Boxplot of Overshoot Times of Mouse and Glove

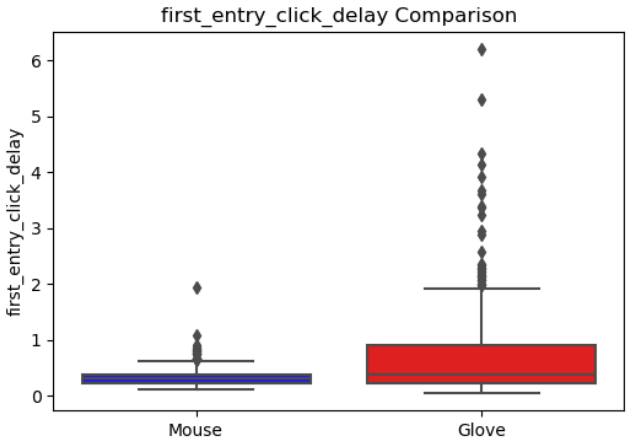

Figure 21: Boxplot of First Entry Delays of Mouse and Glove

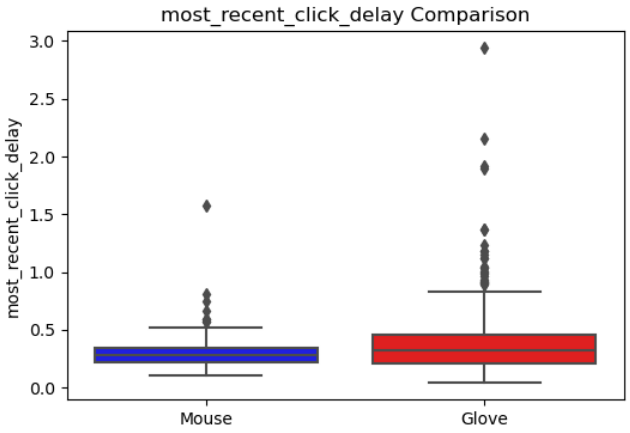

Figure 22: Boxplot of Most Recent Entry Delays of Mouse and Glove