ME 416: Senior Design

Spring 2021

Quantitative analysis of Eye tracker testing rig

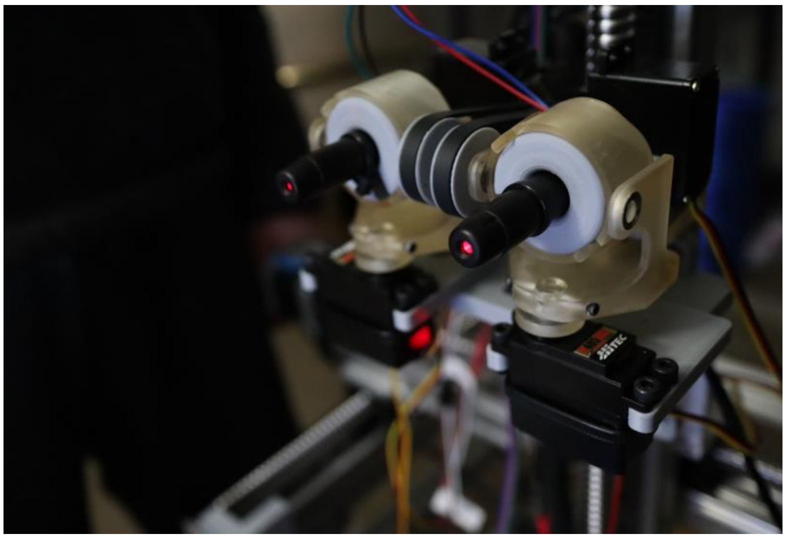


Figure 1 – Eye Subsystem of testing rig

Brenno Aguiar Bonfim Cruz

Chenhao Yu

George Sam Eralil

Kyle Butzerin

Mathew Ubachs

Matthew James

Date of Submission: 04/29/2021

1. Introduction

Amyotrophic Lateral Sclerosis (ALS) is a progressive neurodegenerative disease that destroys a patient's nerve cells in the brain and spinal cord. Consequently, it takes away the patient's ability to control motor functions including speaking, writing, and eventually breathing. Despite being classified as a rare disease, roughly 140,000 patients worldwide every year are diagnosed with ALS. This is equivalent to almost 384 new cases every day (International Alliance of ALS/MND Associations n.d.).

Although a decline of motor functions like moving and speaking is characteristic of ALS, studies have found a resistance of sphincter and sensory functions, as well as oculomotor functions (Proudfoot et al., 2015). Therefore, eye tracking systems have proved to be a great resource for ALS patients as they provide the ability to speak and write by controlling a PC using their eyes. However, studies note that most eye tracker systems only work under specific constraints, like a specific angle or spatial location of the user's head or the user's age (Dalrymple et al., 2018). This can seriously limit the options available to a consumer and oftentimes the user may be oblivious to these requirements at the time of purchase.

For our capstone project, we are building an eye tracker testing rig that will provide a means to collect data on a manufacturer's eye tracker and standardize the testing procedure for eye trackers. The testing rig can also be used to produce benchmarks for eye trackers from different manufacturers.

The eye tracker testing rig that we developed consists of three major subsystems: the eyes, the shoulder or gantry, and the screen subsystem. The eye subsystem consists of 2 individually actuated eyes that use 2 servos each, to control the pitch and yaw rotations of the eyes. A "parallax" function developed in Arduino enables the eyes to focus on a single point on the screen continuously. The gantry subsystem simulates a human shoulder and is actuated by 3 stepper motors, that can move the face or the eyes in the X, Y and Z axes. The screen subsystem consists of a Microsoft Surface tablet mounted with a Tobii eye tracker, which will be used to move a cursor on the PC using the eyes.

The objective of this experiment is to characterize the performance of our eye tracker testing rig by allowing it to type a predetermined message on the PC using a virtual keyboard. The process of typing out the message will be repeated at different shoulder positions to find the most optimal positional parameters that provide the best performance metrics. The performance metric we will be using is the accuracy rate obtained by determining the number of characters at their respective positions in a sentence that the robot got correct.

2. Literature Review

This section details a brief literature survey on the research efforts that have been conducted to measure the precision of eye-trackers. We describe what has been found on the underlying principle of eye trackers from Tobii and the necessary parameters required to interact with them.

The survey also attempts to identify the best parameters for the experiment based on work that has been previously done in the field of eye tracker analysis. Finally, we reflect on the information gap that currently exists in the field that will define the need for our experimentation.

Tobii eye tracker systems primarily rely on three main face parameters to obtain a precise location of where we are looking on a screen. These include, the face geometry, the eye location on the face and the pupil location. The exact coordinates of gaze on a screen are determined by using bright and dark pupil tracking (Tobii n.d.) performed using an array of infrared lasers.

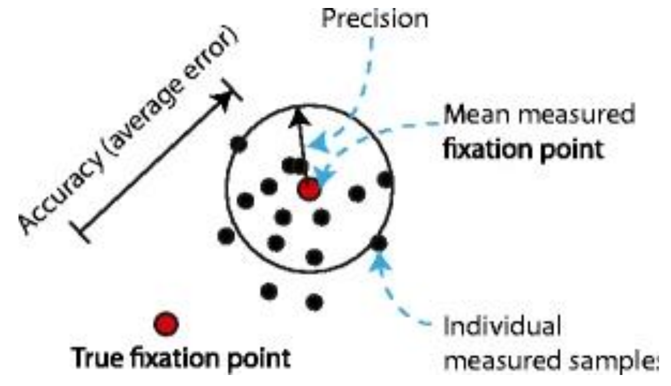


Figure 2 Accuracy vs Precision for eye trackers. Created by Wang et. al, 2017

Wang et. al. in their study of artificial eyes (AEs) for precision measurement of eye-trackers, use AEs from different manufacturers and determine the accuracy and precision of eye trackers. The authors note that spatial accuracy and precision are the most important performance metrics for eye data quality. The authors define accuracy as the difference between the actual gaze position and the gaze position estimated by the tracker, and precision as the repeated reliability to produce a given measurement of the eyes (Wang et. al, 2017). The authors found that when ranking the precision scores of AEs with human data, eye trackers with higher sample rates like the SMI HiSpeed and the EyeLink came out in the first four places, while the ones with lower sample rates like TobiiX2 and EyeTribe were ranked at the bottom.

In the comparison between the different AEs, it was noted that the OEMI-7 AE could be tracked by all the eye trackers. However, the authors go on to note that the OEMI-7 has a 7-mm pupil, whereas human pupil sizes range from 2 to 8mm. Wang et. al. propose that this could be a potential source of imprecision for the eye-trackers (Wang et. al., 2017). Since the authors obtained the best results with the OEMI-7, this will be the AEs that will be used in our experimentation. However, it would remain to be noted what effects an improved AE could have on the accuracy of our results.

In another study conducted on the influence of eye physiology on eye-tracking data quality, it was found that blue eyes performed poorly in comparison to brown eyes (Nystrom et al., 2013). The authors also found a similar decay in precision among participants who wore glasses when compared to uncorrected eyes. Therefore, brown unobstructed eyes will be used in our experimentation as well to minimize bias errors.

In the same study, Nystrom et. al. also note that eye tracker precision can be increased by increasing the pixel resolution at the eye by moving the eye trackers closer. However, being limited by the large pupil size of OEMI-7 AEs and uncertainties regarding the working range for the Tobii eye tracker, we do not quantitatively know the relationship between spatial position of the eyes and precision of the eye tracker. This would be the information gap that currently exists and hence defines the need for this experimentation.

3. Theory

As mentioned previously, our primary goal is to obtain a performance metric of our robot's interaction with the Tobii eye tracker. To achieve this goal, we used a mathematical concept called kinematic chain. This mathematical tool is widely use in robotic applications. In simple words, this mathematical tool allows for representation of joints of a robot in matrix form. This matrix contains the rotation and translation associated to that joint. Each one of these matrices are called frames of reference. (MARK W. SPONG 2020)

Every element is called Frame of Reference (FOR), which is essentially a matrix containing rotation denoted by "R" and translation, denoted by "T" as shown below.

$$g = \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix}$$

We can link two joints of a moving robot by using what is called transformation matrix. This transformation matrix is the result of rotation and translation from one frame of reference to another. For instance, let us take a matrix g_{ij}, where "i" and "j" are points in 3D space. The g_{ij} matrix represents a transformation from the point "i" to the point "j".

For systems with no rotation, the "R" portion of the matrix is equivalent to a three-by-three identity matrix.

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The translation portion of a g_{ij} matrix, indicates the translations in x, y, and z from "i" to "j".

$$T = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Now that we understand how the joints are linked, we can extrapolate this concept to our application.

First, we must identify all "joints" in our robot. To address every joint in this system, we will subdivide the robot in five elements: base, stage, eye mount, left eye, and right eye (Table 1, Fig. 4 and 5).

Naming Convention for Frames of Reference

B	Base situated on the chassis of the robot
C	Center of the stage sitting on the power screw
D	Imaginary point between the eyes

R	Center of the right eye
L	Center of the left eye
S	Screen

Table 1 – Naming Convention for Frames of Reference

Our objective here is to use the kinematic chain to obtain the coordinates and rotations of each one of the eyes, hence finding the input our eye servos need.

The transformation matrix from the eyes (L&R) to the screen (S) is given by the following matrix multiplication:

$$g_{LS} = [(g_{LD})^{-1}\,(g_{CD})^{-1}(g_{BC})^{-1}g_{BS}]\,P_s \ (1)$$

$$g_{RS} = [(g_{RD})^{-1}\,(g_{CD})^{-1}(g_{BC})^{-1}g_{BS}]P_s \ (2)$$

Where P_s is the target position on the screen; g_{LS} and g_{RS} matrices correspond to the location and orientation of the left and right eye with respect to the screen.

The final step is to convert these coordinates in angles using four-quadrant inverse tangent. This method returns angles that have their signs tied to the quadrant at which the point is at. For instance, if our point is on the first quadrant, the cosine and sine are positive.

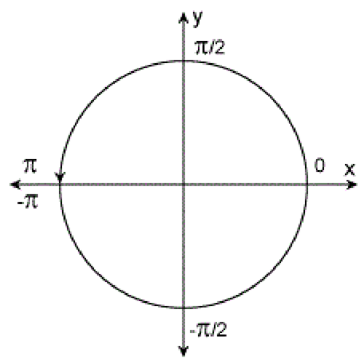


Figure 3 – Four-quadrant inverse tangent representation (MATLAB n.d.)

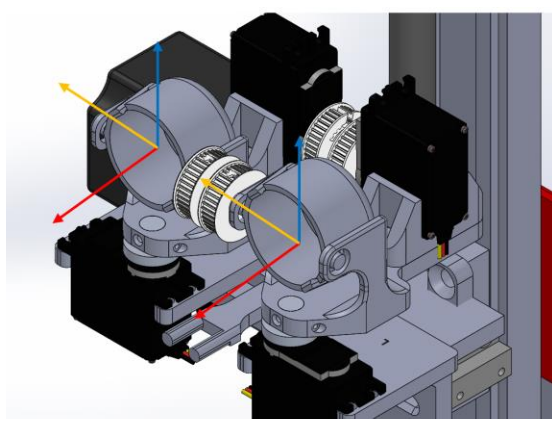


Figure 4 –"R" and "L" frames of reference

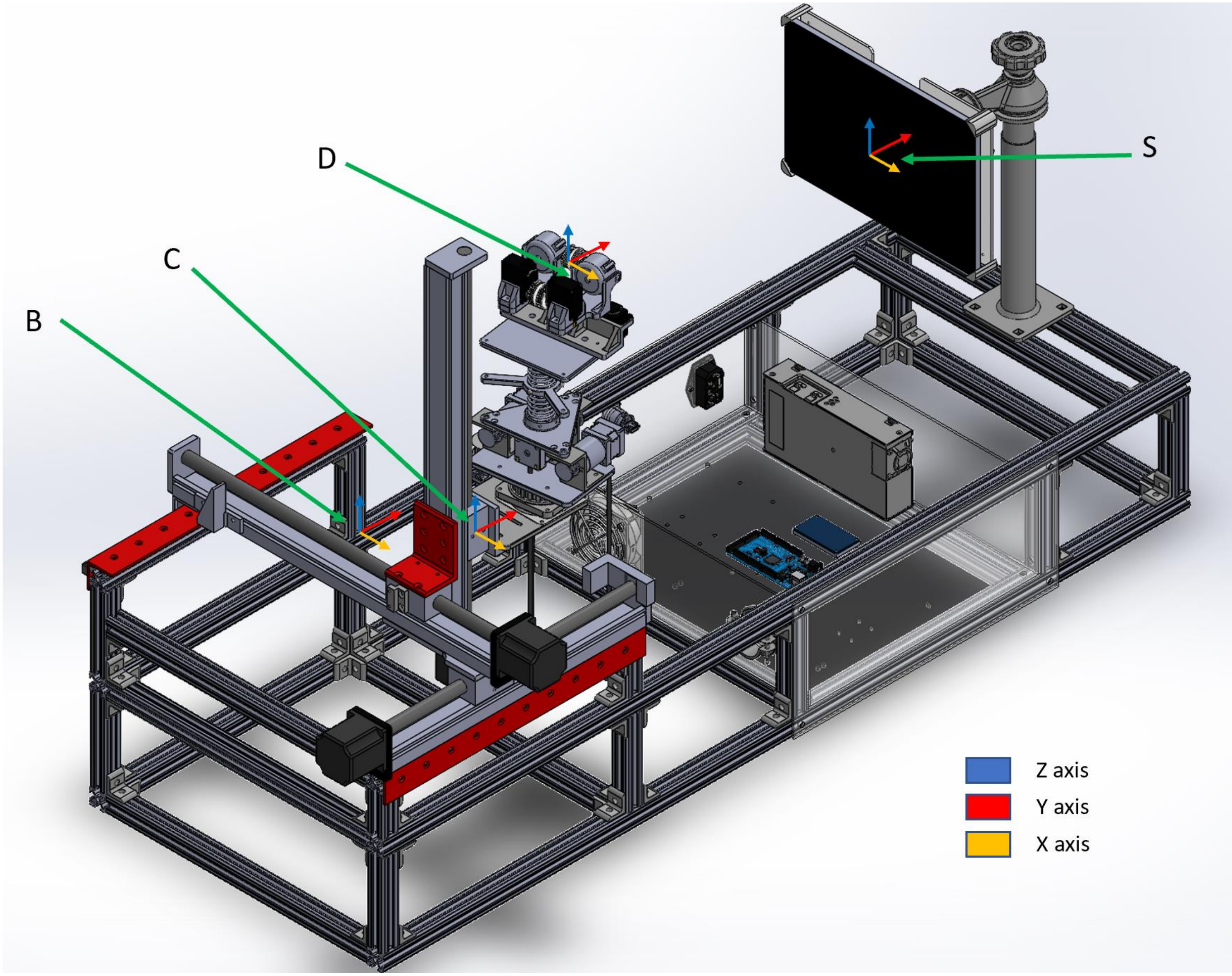


Figure 5 - “B”, “C”, “D”, and “S” frames of reference.

4 Experiment

4.1 Experimental Setup

Our experimental setup consists of three subsystems. 1) Eyes and mounts, 2) Gantry, 3) screen and Tobii. The eye subsystem is composed of four servos, a pair of lasers, and a pair of eyes. The gantry is composed of three steppers with their respective stages and power screws. The third subsystem consist of a Microsoft Surface and the Tobii eye tracker.

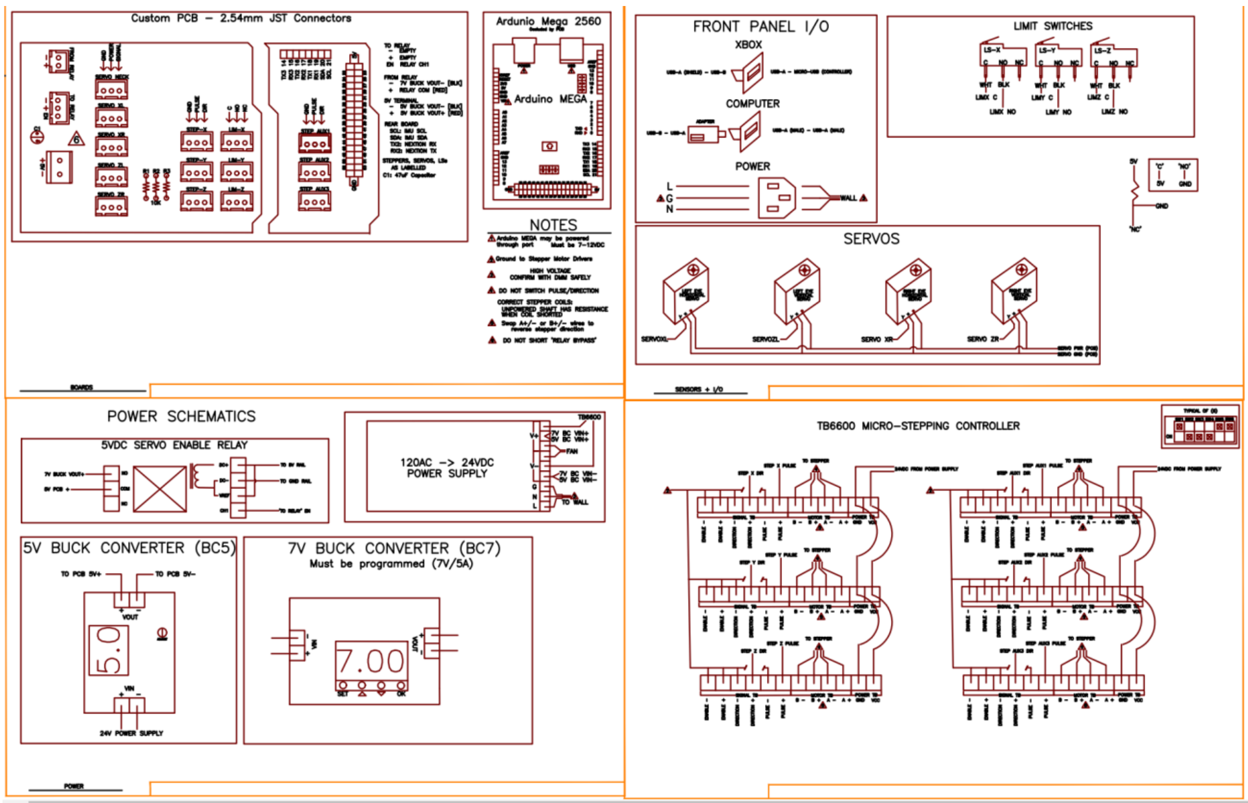


Figure 6 – Microsoft Eye tracker Wiring Diagram.

The eye subsystem relies on four servos to control the pitch and yaw for each one of the eyes. We are using micro lasers embedded in a cap that slides over the eyes as seen in Fig. 7. The lasers are solely used for calibration purposes. They should be stowed after calibration. The gantry subsystem is responsible for the translation of the eye mechanism in x, y, and z. These two subsystems are being controlled using an Arduino Mega, a USB Shield, and stepper drives. Finally, we have the screen subsystem. This subsystem will be treated as a black box throughout the paper due to restricted access to proprietary technology.

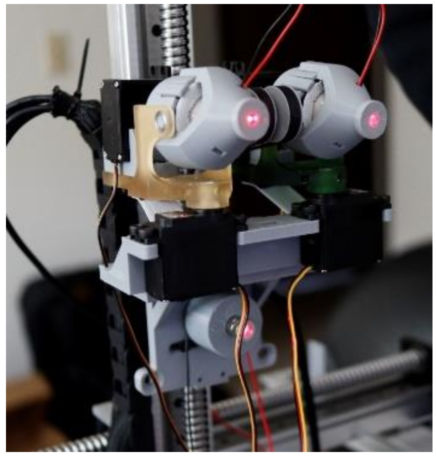


Figure 7 – Eye Subsystem

Before we started testing, we had to assure our system was calibrated. We used the wallpaper shown on the center to calibrate the eye convergence by sending both eye lasers to the blue dot at the center of the wallpaper. We also used the same wallpaper to cross-check the gantry positioning. Next step was to calibrate the eyes using the Tobii eye tracker. Finally, we were ready to make our robot start typing.

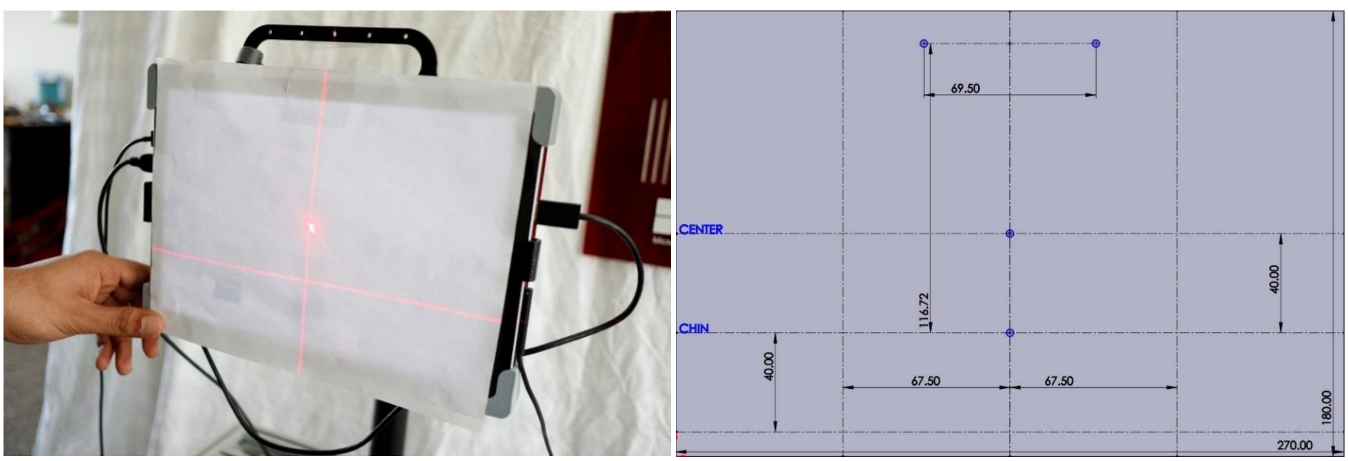
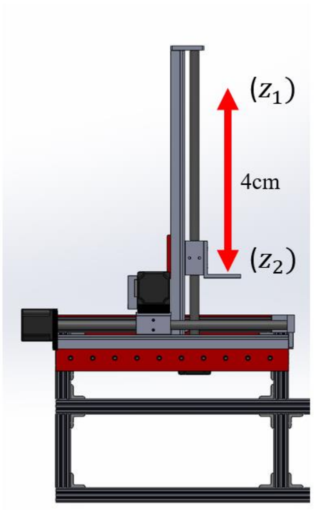


Figure 8 – Laser calibration (left picture) and wallpaper used for calibration (right picture).

The testing strategy was simple. We intend to test how the system can perform at different positions. To achieve that goal, we pre-determined six different locations to perform the test (Fig. 9). Each test would take approximately eight minutes. We essentially made our robot type the same sentence at each location. The sentence came from Steve Gleason, here is the sentence:

"Life is difficult. Not just for me or other ALS patients. Life is difficult for everyone. Finding ways to make life meaningful and purposeful and rewarding, doing the activities that you love and spending time with the people that you love, I think that is the meaning of this human experience." – Steve Gleason

We picked this sentence because besides resonating with the project, it has a wide range of letters that can challenge the robot. The total number of characters is 294.



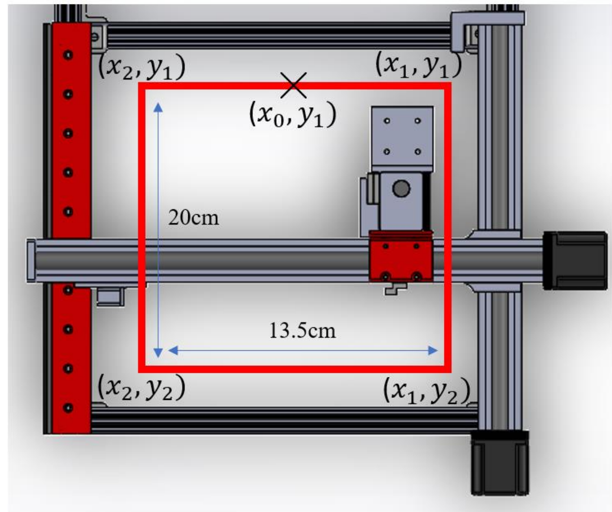


Figure 9 – Schematic of pre-determined locations used for testing.

4.2 Experimental Procedure

Start-Up Procedure:

1. Before beginning the experiment, make sure the robot is plugged onto 120 AC.
2. Make sure the screen is centered and it is positioned vertically.
3. Turn the Microsoft Surface on.
4. Insert the Tobii eye tracker onto the slots right below the Microsoft Surface. There are two magnets that can assure the Tobii is in place.
5. Turn the power supply on and plug in the Xbox controller using the USB cord.
6. Insert the laser caps onto the eyes until it clicks and plug the red and black cables to live and neutral female pins.
7. Press “Y” on the Xbox controller. This will initiate the stepper calibration sequence. Please wait until the robot stops moving.
8. Press “A” on the Xbox controller. This will initiate servo calibration sequence.
9. Open the calibration wallpaper located in the PowerPoint presentation on the desktop.
10. Move both laser points to the blue dot located on the center of the screen. Use the left and right arrows from the D-Pad to switch between the servo motors. Use the Up and down arrows to move the servos. Press “L1” to exit and save calibration.
11. Remove the laser caps.
12. Open Tobii Calibration and make sure the software can see the eyes.
13. Press “L2” on the Xbox to initiate Tobii eye calibration.
14. There are four calibration points. As soon as Tobii recognizes the first point, press “B” to move to the next one.
15. Press “L1” on the Xbox controller to initiate servo manual mode.
16. Press “R3” to initiate the testing.

Data Acquisition:

1. Press “R1” to enter stepper manual mode.
2. Move to the pre-determined positions using the analog stick.
3. Press “R3” to enter testing mode.
4. Repeat the operation for the other preset values.

Shut-Down Procedure:

1. Turn the power switch off.
2. Unplug the power cord out of the outlet.
3. Turn the Microsoft Surface off

4.3 Data Analysis & Uncertainty

After collecting the word output from our experiment, we calculated the success rate per sentence during every test. This calculation was performed by simply dividing the number of characters missed by the total number of characters. We plotted the success rate versus gantry position as we can see in Fig. X.

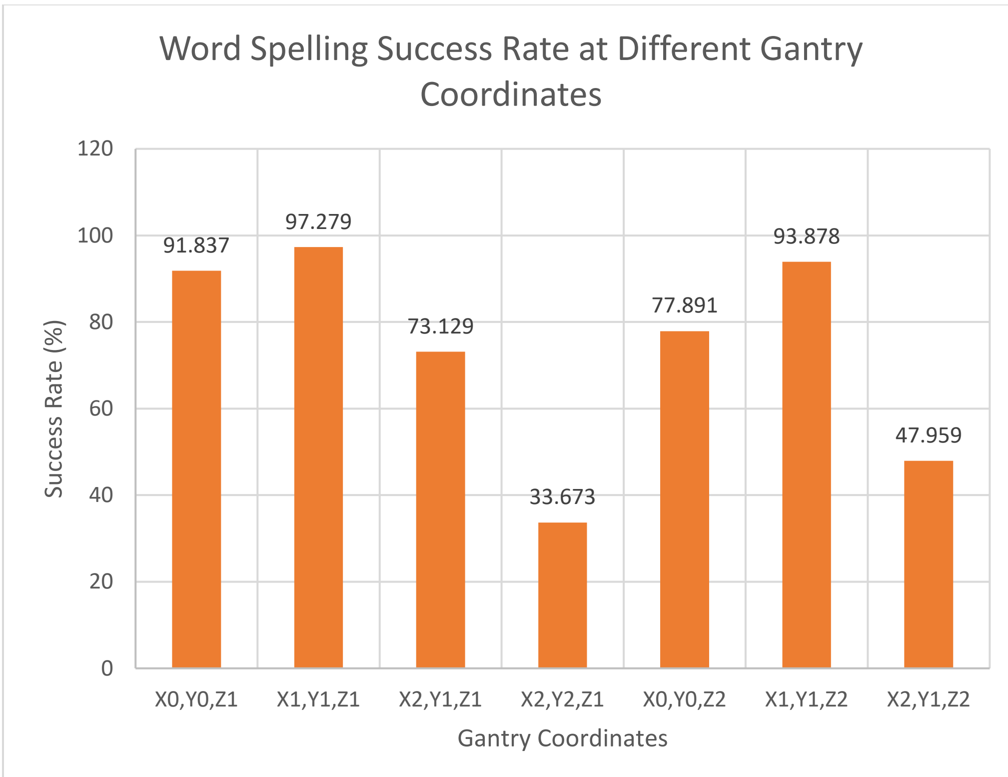


Figure 10 – Word Spelling Success Rate at Different Gantry Coordinates

5 Results and Recommendations

The best result was found at coordinate (X_1, Y_1, Z_1) which corresponds to the front right side of the screen. At this location, the experiment had a 97.28% success rate. The second best was at (X_0, Y_1, Z_1) which corresponds to front and center of the screen; the success rate was of 93.88%. The lowest accuracy was obtained at (X_2, Y_2, Z_1) at the back and left side of the screen; the success rate was of 33.67%

The results deviated slightly from our predictions. Initially we thought that the robot would perform the best at the center position, which was not the case. The best result was obtained at the right side of the screen instead. This led us to believe that there is a gap in understanding on how the Tobii works. As mentioned previously, the eye tracker is a black box, thus we do not know if the problem lies on our robot or the eye tracker.

We would like future groups to carry out this research and investigate the theory behind machine vision technologies used by eye trackers. We intend to use this information to understand the engineering requirements necessary to make the testing rig as unbiased and realistic as possible, and not to bypass the system.

Furthermore, more work needs to be done to expand on the testing procedure that we have developed. Future work would focus on expanding the testing region to cover the entire screen and use metrics like accuracy and precision mentioned before to analyze the system. Moreover, testing of different eye tracker systems and different/custom built AEs would be beneficial to identify the true sources of uncertainties in the testing.

Bibliography

Anna Maria Feit1, 2 Shane Williams2 Arturo Toledo2,3 Ann Paradiso2. 2017. *Toward Everyday Gaze Input: Accuracy and Precision of Eye Tracking and Implications for Design.* Research Paper, Microsoft Research.

n.d. *International Alliance of ALS/MND Associations.* Accessed 03 29, 2021. https://www.alsmndalliance.org/what-is-alsmnd/.

MARK W. SPONG, SETH HUTCHINSON. 2020. *Robot Modeling and Control.* Hoboken: John Wiley & Sons, Ltd.

MATLAB. n.d. *MathWorks.* Accessed April 25, 2021. https://www.mathworks.com/help/matlab/ref/atan2.html.

Nystrom, M., R Andersson, K Holmqvist, and J van de Weijer. 2013. "The influence of calibration method and eye physiology on eyetracking data quality." *Behavior Research Methods.*

Proudfoot, Malcolm, Ricarda A.L. Menke, Rakesh Sharma, Claire M. Berna, Stephen L. Hicks, Christopher Kennard, Kevin Talbot, and Martin R. Turner. 2015. "Eye-tracking in amyotrophic lateral sclerosis: A longitudinal study of saccadic and cognitive tasks." *Amyotroph Lateral Scler Frontotemporal Degener.*

Tobii. n.d. *Tobii Dynavox.* Accessed 03 25, 2021. https://www.tobiidynavox.com/about/about-us/how-eye-tracking-works/.

Wang, Dong, Fiona B. Mulvey, Jeff B. Pelz, and Kenneth Holmqvist. 2017. "A study of artificial eyes for the measurement of precision in eye-trackers." *Behavior Research Methods.*