**Eye Tracking Powered Wheelchair with Computer Vision and Gaze-to-Speech for People with Full Body Paralysis**

**Category: Engineering and Technology**

Asanshay Gupta

Atlanta International School

2980 North Fulton Dr.

Atlanta, GA, 30305

## Table of Contents

## Introduction

Imagine being totally dependent on somebody. Imagine having to speak through somebody. Imagine not being able to move. Over 115,000 people in the US alone have locked-in syndrome or tetraplegia, one of the worst forms of paralysis. Of these people, about 28% earn less than $15,000 a year (“Statistics about paralysis”). With current mobility solutions costing in the range of $12,000 or more, many of these people, especially those without insurance are forced to accept a future bound to somebody else (“I Series Pricing”, 2020).

When people are paralysed with locked-in syndrome, most of the time they retain control of some muscles such as those used to move the eyes and blink (Smith et al. 2005). Most current mobility solutions are based on the eye and tongue movements to communicate information to another person, who then can perform tasks (Evans, 2020).

**The goal of this project is to make a low cost, fully featured, modular, and adaptable powered wheelchair and gaze-to-speech system that is controlled with only the eyes to help people with locked in syndrome move around, interact with the world and communicate with caregivers with greater independence.**

Paralysis comes in many different forms, each affecting different parts of the body. Although the product targets people with classic locked in syndrome, every case is different and may allow for different ranges of movements. This means that the design needs to be modular to allow for easy changes in input method, and support the use of external switches, such as muscle twitch sensors.

### Overview of Available Options

In classic locked-in syndrome, the most common muscles within which motion is retained are the eyes, tongue, and those used to breathe (Smith et al., 2005). It is therefore essential to use one of those muscles to allow for communication. One of the most reliable and effective ways of doing this is through eye tracking, which uses the positions of two eyes to trigger actions on a screen based on where the eyes are looking. The most common type of eye tracking uses computer vision technology to detect the eyes, and is positioned somewhere near the screen, tracking the eyes from the screen's perspective. Other eye tracking solutions include goggles that track from the eye's perspective, which, even though they are more accurate, are more invasive, more expensive, and can be uncomfortable for long term use. Alternatives to eye tracking include tongue tracking and "sip-puff" technology, but these are limited in that they are invasive solutions that usually yield less data than eye tracking (Evans, 2020). Tongue tracking typically works by surgically implanting magnetic chips into the patient's tongue, which communicate with a chip embedded into the top of the patient's mouth (Kim et al., 2013). "Sip-puff" technology uses sensors at the end of a straw to measure whether the user is breathing in or out. This is then interpreted as a binary signal to trigger an action (Evans, 2020). These solutions are extremely invasive and expensive, and they also yield much less data than eye tracking. Screen-side eye tracking is therefore the best suited for use in a wheelchair.

**Cost**

A very important consideration was keeping the cost of the design under $4000 to make it accessible to the general public. For comparison, similar products can cost over $12,000 and have cumbersome features such as eye tracking goggles and long response times (Tobii Dynavox, 2020). This means that families already struggling under the weight of medical expenses have a tough choice: either shell out a huge sum of money, or give up their loved one's mobility. The astronomical price tags on current solutions mean that those who need them the most cannot afford them. Over 115,000 people in the US alone have a form of full body paralysis, and many of them are already being crippled with other medical expenses (Christopher and Dana Reeve Foundation, 2020). With current mobility solutions costing in the range of $12,000 or even more, many of these people without insurance are forced to accept being bound to somebody else (Tobii Dynavox, 2020). The target price point of $4000 was calculated based on the minimum cost of materials for such a product, while keeping accessibility in mind. In addition, the product was designed so it can easily be adapted to fit a user's existing power wheelchair and/or laptop for added cost benefit. For example, if a user already had a power wheelchair, this solution could easily be mounted to it, enabling eye control as well as eliminating the cost of purchasing a new power wheelchair for the final product.

**Sensing Technologies**

While eye tracking is used, the eyes, the user's primary method of judging their surroundings, need to be focused on the screen. Having the eyes completely focused on the screen can be a safety concern as the user would not be capable of recognizing and avoiding obstacles in the path of the wheelchair. This necessitates a robust sensing and stopping system to detect obstacles and either navigate the wheelchair around them or stop (Maurer, 2020). There are many possible sensing solutions available, among them LIDAR, computer vision, and time-of-flight sensors. LIDAR is one of the most effective at sensing depth, which is useful but can also easily be misrepresented. LIDAR technologies also result in a very high cost, because of the technology's novelty. Although time-of-flight sensors are much cheaper, they lack accuracy even more than LIDAR and require optimal conditions to work (Schweber, 2019). Often, computer vision can match or even exceed the performance of LIDAR while remaining at a fraction of the cost (Wang, Yan, et al.). In addition, computer vision operates in a way similar to that of the eyes, which makes it well suited for this task.

Taking all of this into consideration, the most effective solution for this would be to mount a screen and eye tracker onto a standard eye tracking powered wheelchair. On the screen, there should be an interface to control the wheelchair, as well as one for a speech system. To ensure user safety, there should be many cameras all around the wheelchair projecting onto the screen, which will be detecting and avoiding obstacles in the path of the wheelchair. The solution should be modular and adaptable to provide a customized experience for each user.

## Materials, Methods, and Procedures

### Choice of Eye Tracker

After an interview with a researcher from the University of British Columbia who has experience with various eye trackers, the most appropriate choice for an eye tracking power wheelchair was an eye tracker from Tobii Dynavox. The eye tracking system runs smoothly in any application on a PC running Windows, which allows for a lot of flexibility in terms of device selection. Tobii Dynavox is one of the biggest companies that manufactures eye trackers in the world and is known for their accuracy, reliability, and ease-of-use (Soroski, 2020). Of the various models offered, the PCEye Mini was chosen because of its simplicity and low cost compared to other options. It is capable of moving the mouse and performing simple mouse clicks on the screen, which is enough to control the wheelchair and allow for a gaze-to-speech system.

### Software-Hardware Interface

A web interface was selected for the UI for its portability to multiple devices and ease of development. It is hosted in Node.js, and is programmed using JavaScript, HTML, and CSS. To control the wheelchair, this software component of the system needs to be connected to the actual wheelchair hardware. As shown in figure 1.1, the main interface between the PC and the electronic components is a microcontroller, in this case an Arduino. From this, connecting to the wheelchair can be done in two main ways, internally and externally. Connecting to the wheelchair internally means to disassemble the controller of the wheelchair and directly interface with the signal coming from the joystick. Although this allows for better control, this solution is permanent and non-modular. Wheelchairs can be vastly different on the inside, so connecting internally results in a huge cost in ease of setup as most wheelchairs lack clear labeling inside. Connecting externally consists of physically manipulating the joystick using servo motors. This replicates the functionality of a human hand, and therefore works with a slew of different wheelchair controller designs. This solution makes it possible to easily switch the wheelchair without much difficulty. The Arduino microcontroller is connected to the Node.js server using the “johnny-five” module and a protocol called Firmata, which allows for nearly lag-free signal. The servos control the joystick through 3D printed “prongs” on either side of the stick (see figure 1.2), and are calibrated to drive the wheelchair straight.

*Figure 1.1: Flowchart Showing Component Relationships*

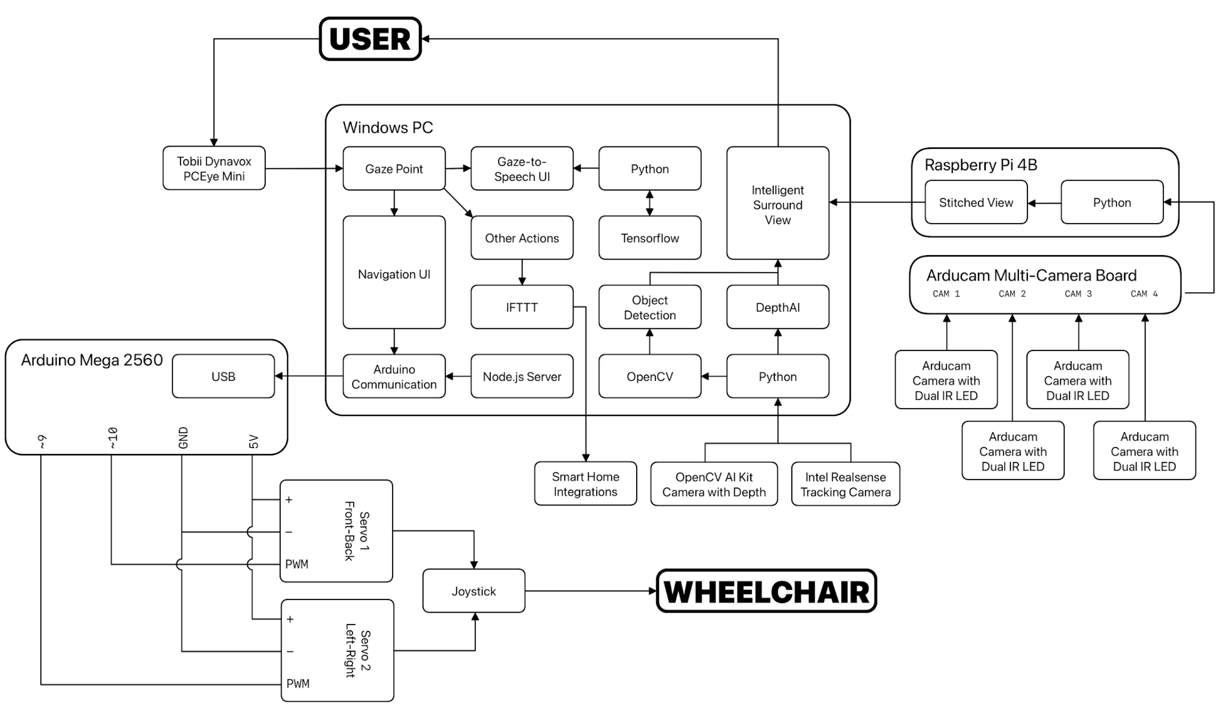
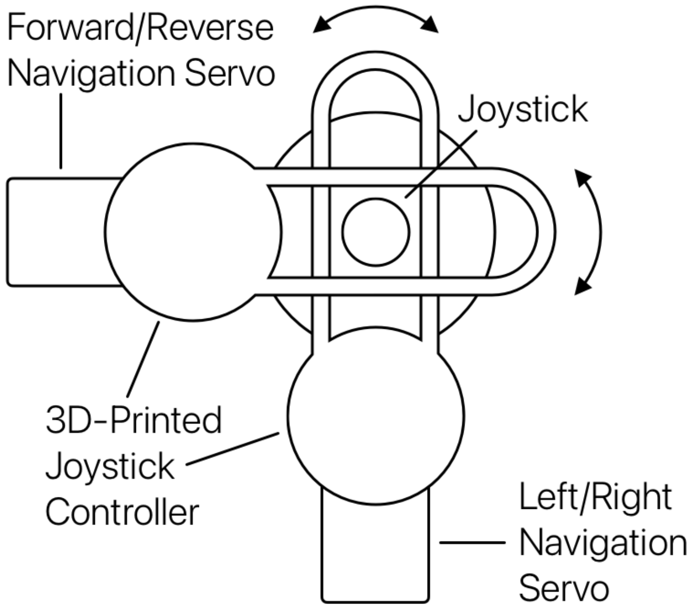


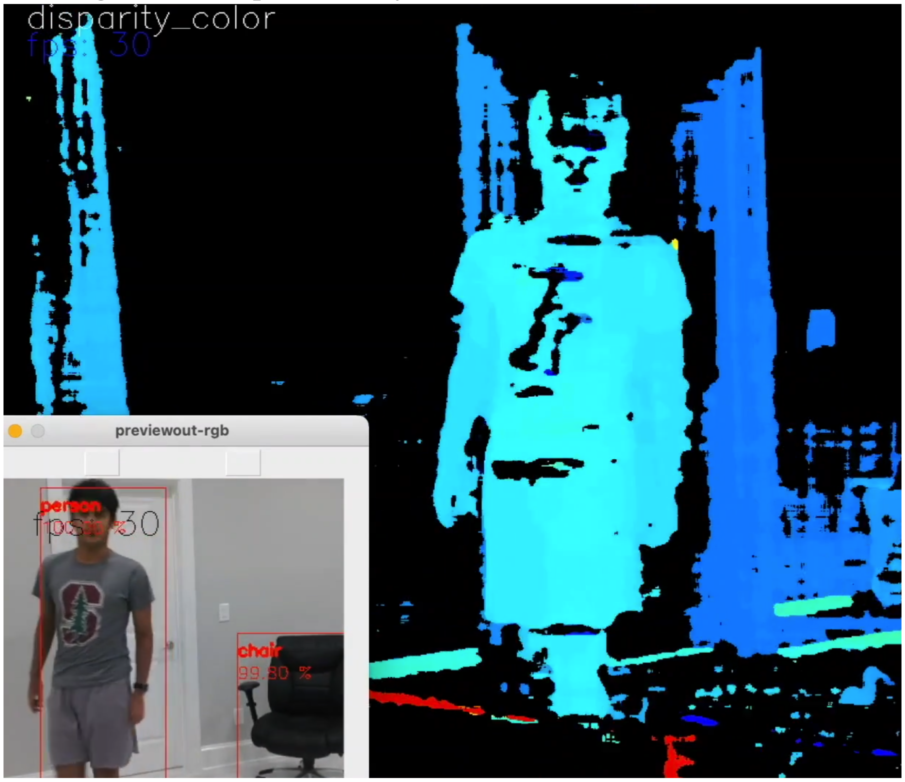
*Figure 1.2: Servo-Joystick Interface*




**AI Camera Systems**

As the eyes of the user will be focused on the screen during operation, the wheelchair system needed an effective way of both sensing depth and detecting objects. For this, the optimal solution was computer vision. The chosen camera was the OpenCV AI Kit with Depth, which consists of two infrared cameras, one 12 megapixel RGB camera, and a MyriadX vision processing unit. Using machine learning, the main camera identifies various objects such as people, pets, or furniture. The differences between the two stereo infrared cameras allow for a very accurate depth map, which can help with safety in the case that a user gets too close to a wall. As seen in figure 1.3, the camera is able to detect objects for collision avoidance. Although the depth image is not very high resolution, it can be used in conjunction with the RGB camera to precisely identify objects and their respective depths. For more reliable navigation, Intel's Realsense T265 tracking camera will be used for V-SLAM (Vision-based Simultaneous Localization And Map building), occupancy mapping, path planning, and collision avoidance.

*Figure 1.3: Depth and Object Detection View From Camera*




**List of Components**

*Table 1.1: List of Components and Their Purpose*

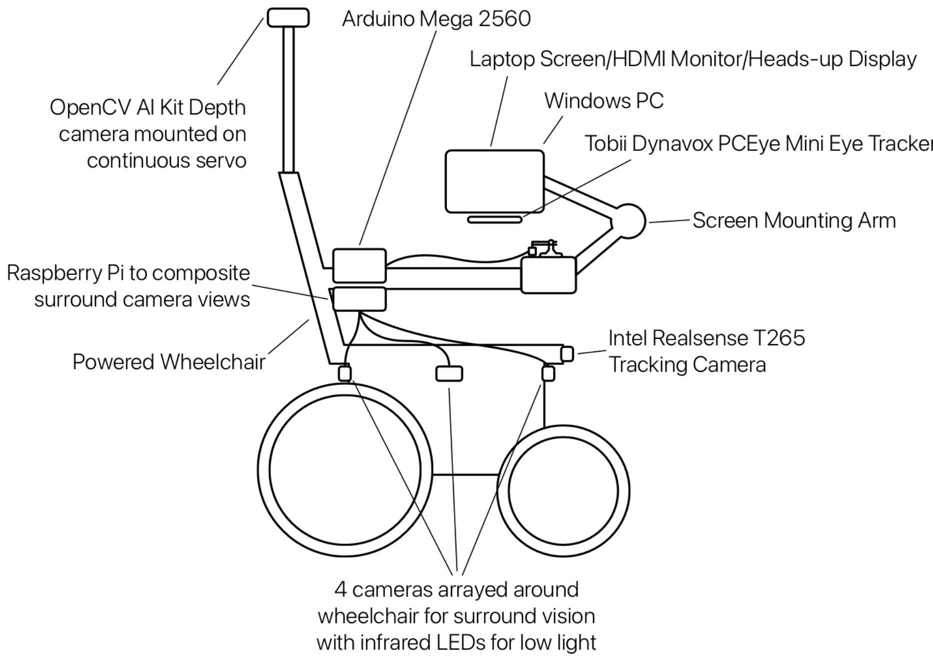
| **Component** | **Purpose** |
|---|---|
| Porto Mobility Ranger D-09S Wheelchair | Commercially available powered wheelchair |
| Tobii Dynavox PCEye Mini | Eye tracker with fast setup and reliable tracking |
| Asus Zenbook 14 Laptop PC | Windows PC with large screen |
| Arduino Mega 2560 | Microcontroller to control navigation servos |
| Raspberry Pi 4B | Microcomputer to stitch camera feeds |
| OpenCV AI Kit Depth | Camera with integrated machine learning |
| Continuous Rotation Servo | Servo to pan depth camera |

| | |
|---|---|
| Arducam Multi-Camera Board and Cameras | Multi-camera interface for Raspberry Pi |
| 3D printed Joystick Controller with Servos | External controller to move wheelchair joystick |
| Laptop Arm | Used to mount laptop to the wheelchair |
| Wires and USB cables | Needed to connect all components |
| Node.js | Programming environment for backend code |
| Web Interface | UI for navigation, gaze-to-speech and actions |
| Python with OpenCV and DepthAI | Used for object detection |
| Intel Realsense T265 Tracking Camera | Tracking camera to avoid collisions and map path |

**Assembly**

Once every component has been individually tested, the wheelchair needs to be assembled. Table 1.1 provides a list of essential components. As shown in figure 2.1, the eye tracker should be mounted to the Windows PC right below the screen, to allow for optimal tracking. The screen of the PC should be mounted directly in front of the user. If the PC used is a laptop PC, it can be mounted using a standard laptop car mount. The eye tracker and Arduino both need to be connected to the PC, so the use of a USB hub may be necessary. On many wheelchairs, the joystick is fixed to the arm using two screws. These screws should be used to mount brackets for the servos, then mount the servos and press-fit the prongs on top. Once these have been mounted, wires should be connected to the corresponding pins on the Arduino as per figure 1.1, then the USB cable should be routed from the Arduino to the USB hub on the other side.

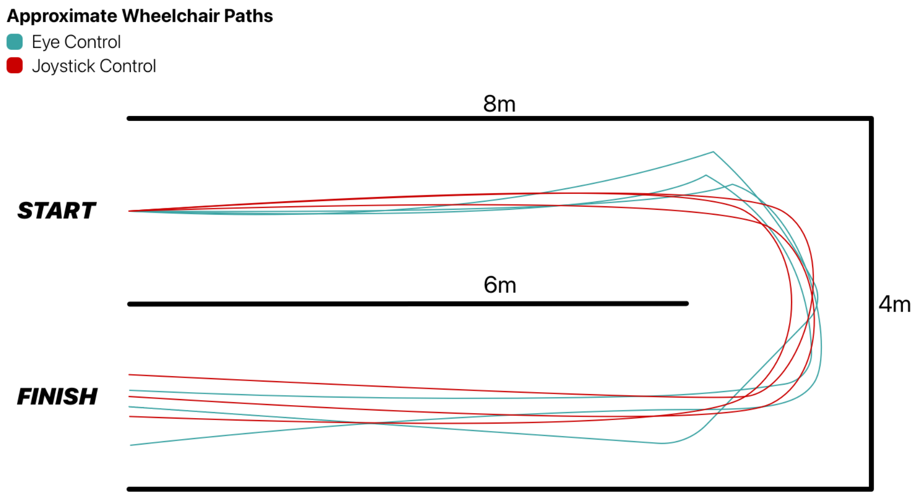
*Figure 2.1: Annotated Wheelchair Diagram*




**Testing**

To test the motion of the wheelchair, the wheelchair should be placed in a large room and an adult user should be instructed to travel through a course with set boundaries without moving any part of their body except their eyes. To set up the test, 4 objects should be placed to form a 4m by 8m rectangle. A 6m line should be marked in the middle from the left side (See figure 1.4). The wheelchair should start on the left side centered above the center line and travel around the line to right below its leftmost endpoint. The tester needs to travel along the centerline, perform a u-turn to the other side of the line, and then return to the start line on the other side of the centerline. The time should be measured to the nearest second from when the wheelchair leaves the start point to when any part of the wheelchair crosses the finish line. The time taken should then be compared to the time for the same users driving the wheelchair with the normal joystick. As well as giving a strong quantitative metric on the differences between joystick drive and eye tracking, the qualitative data from this test can highlight possible improvements for ease of use and reliability.

*Figure 1.4: Wheelchair Test Course*

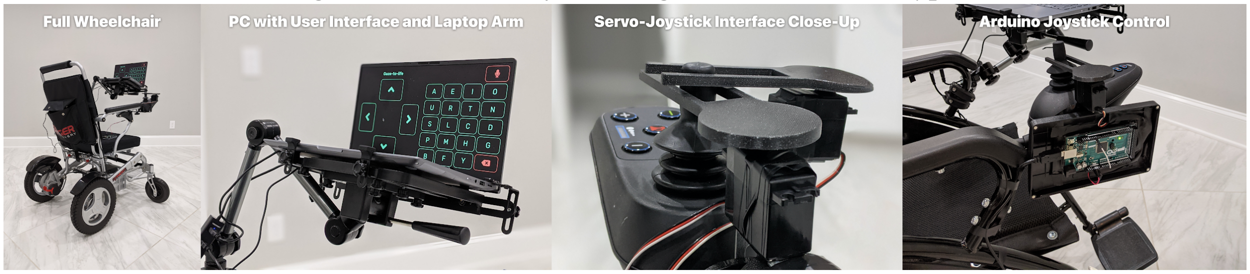



## Results

### Completed Wheelchair Prototype

The completed wheelchair prototype consists of a Windows PC mounted to a standard commercial wheelchair via a laptop arm, which is connected to an Arduino. The Arduino is mounted to the inside of the right handle, and controls two servos which each control one axis of joystick motion (See figure 1.2 for close-up). There are 4 cameras with infrared LEDs arranged around the wheelchair, which connect to a Raspberry Pi mounted below the wheelchair and will eventually show a surround view on a separate monitor. The depth camera with embedded spatial AI is mounted on the top of the wheelchair, and can turn using a servo. See figure 2.1 for photos of the working wheelchair prototype.

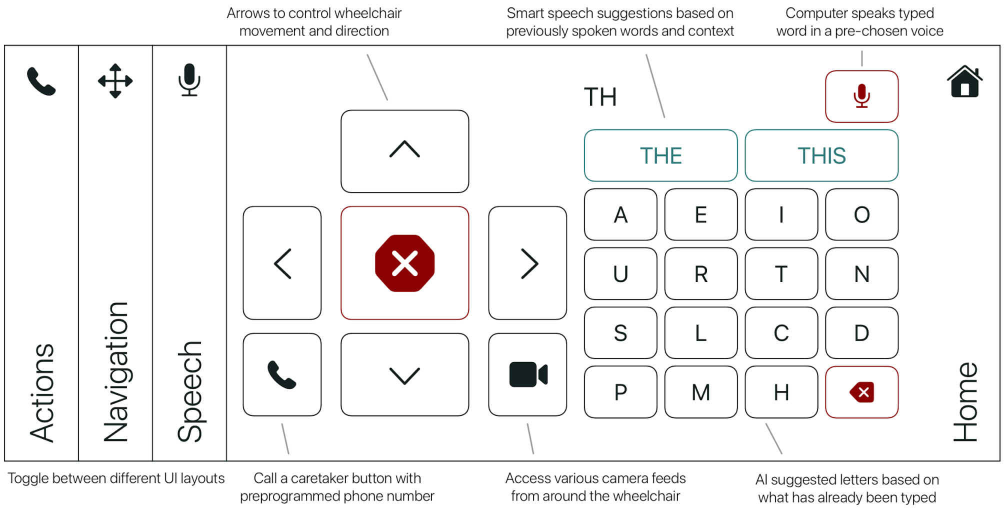
*Figure 2.1: Photos of Working Wheelchair Prototype*




**User Interface (UI)**

An important consideration while designing the user interface was to make large targets for the eye tracking, to prevent mis-clicks. The final user interface design (see figure 2.2) is composed of two main sections, which correspond to the two main tasks performed by the user. To the left of the screen, there are a set of arrows which allow the user to easily move the wheelchair, as well as a button to stop the wheelchair and to show the feeds from the various cameras arranged around the wheelchair. This design allows the user to smoothly drive the wheelchair while maintaining a level of safety. On the right side, there are some common letters arranged in a grid, which act like a keyboard for the user. For example, if the user looks at a letter, it is appended to the currently typed text at the top of the screen. For added speed, the user can select one of the smart suggestions at the top of the screen. The suggestions are added using next word prediction based on NLP (Natural Language Processing) and deep learning. The deep learning model is built using LSTM (Long Short-Term Memory), and more relevant training data for phrases will be used in the future. To allow for large eye tracking targets, only a limited selection of letters are shown at a time. The machine learning algorithm suggests some of the most appropriate letters, which eases the typing experience as there is less searching to do before the desired letter is found.

*Figure 2.2: Annotated User Interface Diagram*

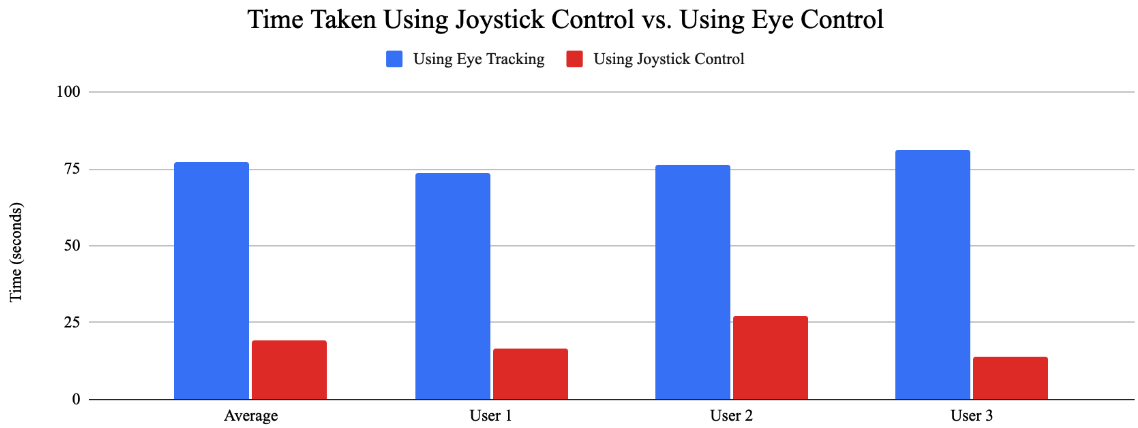



## Summary of Test Results

*Table 2.1: Table of Test Results*

| User | Time Taken to Complete Path (seconds) | | | |
|---|---|---|---|---|
| | **Trial 1** | **Trial 2** | **Trial 3** | **Average** |
| Using Eye Tracking | | | | 77 |
| 1 | 59 | 72 | 90 | 74 |
| 2 | 62 | 87 | 80 | 76 |
| 3 | 83 | 92 | 69 | 81 |
| Using Joystick Drive | | | | 19 |
| 1 | 12 | 25 | 13 | 17 |
| 2 | 26 | 34 | 22 | 27 |
| 3 | 12 | 15 | 14 | 14 |

*Figure 2.3: Graph Comparing Joystick Control and Eye Control*



As shown in figure 2.3, on average, the joystick controlled wheelchair outpaces the eye controlled wheelchair by 401% or by 58 seconds. There are 3 key reasons for this: the tuning of the wheelchair, the toggle system of driving, and the range of motion permitted by the current user interface. One observation from the testing was that instead of driving straight, the wheelchair would veer off to one side, presumably because of the servo prongs being slightly misaligned. The time spent correcting this error meant the eye tracking wheelchair needed to turn more than the non-eye tracking version. Another issue was the toggle system of steering, in which the user had to click the stop button in between different directions. This meant that the users frequently turned too much or too little as it was difficult to estimate the optimal stopping point. The final main factor was the capability to simultaneously turn and drive, which meant the joystick controlled wheelchair could have a much wider turn radius than the eye controlled one (See figure 1.4 for path differences).

## Discussion of Results

### Improvements Based on Testing

The three main factors that slowed down the eye tracking wheelchair can all be fixed to an extent. For the veering off, the servo and its prong attachment needs to be tuned so it can move to a position where it moves the joystick forwards and only forwards. In addition, a tracking camera can be added, which will provide real-time feedback to the navigation system. The tracking camera will be used for auto-navigation to navigate through saved indoor locations and quick-return to previous positions. To allow for more precise adjustments for steering, the acceleration curves in the toggle system can be adjusted. Finally, a wider turn radius can be achieved by adjusting the user interface to include "diagonal" buttons which allow for the user to simultaneously drive and turn.

**Improvements to Testing Method**

For more accurate testing results, testing should be conducted in a variety of situations using more people. These could include going through doors, completing other courses, or going along a sidewalk outside. Having more people try the wheelchair would increase the sample size, improving the reliability of the results. It should also be compared with other existing eye tracking powered wheelchairs, as well as other alternatives for people with full-body paralysis. In addition, the speech system should be tested using different words and commonly spoken phrases to test the accuracy and speed of the speech system.

**Projected Cost vs Achieved Cost**

The total cost of all the parts of the wheelchair is approximately $3,800, which is slightly under the targeted $4,000. However, there are multiple reasons that this value may not be totally representative of a market price for the wheelchair. Even though the cost of materials may be less than $4,000, this does not account for the cost of labor for fabrication, machine time, capital expenses, development costs, or the efficiencies of volume production. For example, even though many of the parts were optimized for quick printing, the wheelchair uses many 3D printed parts, which would cost approximately $67.34 to manufacture traditionally (Szyk, 2018). In addition, there are many circuits that are made using breadboard parts, which would need to be replaced with a custom soldered PCB on the final version, furthermore increasing the cost. Regulatory approval and safety testing is expected to further add to the price.

**Wheelchair-Agnostic Design**

The modular nature of the external control system allows the eye control system to be easily retrofitted onto any powered wheelchair. This was achieved through using an external control system in the servos as well as by creating a versatile mounting system. If the user does not have to pay for a new wheelchair, the total price of the product halves, making it much more accessible. In addition, the software was designed to be easily installable on any PC, which means if the user has a PC, the system can be adapted to that as well. The system can also function with one or more features excluded, such as the depth tracking cameras or gaze-to-speech system to control cost or adapt to user needs.

**Modular Design**

The design of the user interface also permits the use of any external switches if the user has the capability to activate one, such as twitching a muscle or using a sip-puff device, to augment or replace eye tracking. When activated, the visible selected state will cycle through the buttons at a regular interval, allowing for the user to trigger the highlighted button using the external switch. Alternate hardware is being tested to replace the laptop screen, including an HDMI monitor and heads-up displays (HUD), which projects the user interface onto a see-through surface in front of the user. This allows the user to maintain awareness of their surroundings while controlling the wheelchair.

**Smart Home Integration**

To further move the user towards independence, the goal is to connect the user interface to a smart home system enabling users to do tasks such as call a caregiver, switch lights on and off, moving curtains up or down, adjust home thermostat or use a home assistant device such as Amazon Alexa, Google, etc. through integration with the IFTTT (If This Then That Web Based Service) platform, which allows the user to control smart home devices such as lights or curtains. In addition, a button can be added to give a push notification to a companion device, further reducing the strain on the caregiver as they can stay further away from the user.

**Literature Cited**


Evans, Nicholas. Videoconference interview. Conducted by Asanshay Gupta, 3 Sept. 2020.

Galante, Adriano, and Paulo Menezes. A Gaze-Based Interaction System for People with Cerebral Palsy. Coimbra, Portugal, Procedia Technology, 2012. ScienceDirect, doi:10.1016/j.protcy.2012.09.099. Accessed 24 Sept. 2020.

"I-Series Pricing" Tobii Dynavox, www.tobiidynavox.com/about/about-us/https://us.tobiidynavox.com/pages/i-series-ap/. Accessed 14 Sept. 2020.

Kim, J., et al. “The Tongue Enables Computer and Wheelchair Control for People with Spinal Cord Injury.” Science Translational Medicine, vol. 5, no. 213, 27 Nov. 2013, pp. 213ra166-213ra166, www.ncbi.nlm.nih.gov/pmc/articles/PMC4454612/, 10.1126/scitranslmed.3006296. Accessed 11 Jan. 2021.

Maurer, Chris. Videoconference interview. 11 Sept. 2020.

"Node.js - Express Framework." TutorialsPoint, www.tutorialspoint.com/nodejs/nodejs_express_framework.htm. Accessed 13 Jan. 2021.

Rwaldron. "Johnny-five Wiki." Github, 8 May 2020, github.com/rwaldron/johnny-five/wiki/Getting-Started. Accessed 13 Jan. 2021.

Schweber, Bill. “LIDAR and Time of Flight, Part 1: Introduction.” Microcontrollertips.com, 2019, www.microcontrollertips.com/lidar-and-time-of-flight-part-1-introduction-faq/. Accessed 10 Jan. 2021.

Soroski, Tom. Videoconference interview. 20 Sept. 2020.

"Stats about Paralysis." The Christopher and Dana Reeve Foundation, www.christopherreeve.org/living-with-paralysis/stats-about-paralysis. Accessed 14 Sept. 2020.

Smith, Eimear, and Delargy, Mark. “Locked-in Syndrome.” BMJ, vol. 330, no. 7488, 17 Feb. 2005, pp. 406–409, www.ncbi.nlm.nih.gov/pmc/articles/PMC549115/., 10.1136/bmj.330.7488.406. Accessed 16 Jan. 2021.

Szyk, Bogna. “3D Printing Cost Calculator.” *Omnicalculator.com*, 2018, www.omnicalculator.com/other/3d-printing. Accessed 13 Jan. 2021.

Wang, Yan, et al. Pseudo-LiDAR from Visual Depth Estimation: Bridging the Gap in 3D Object Detection for Autonomous Driving.

## Acknowledgement of Major Assistance

This project was completed at my home from November 2020 to the present, January 2021. It was completed using funding from the International Baccalaureate’s MYP Innovators Grant. Scientific guidance and feedback was received from Dr. Peluchi Flores, Director of STEAM and Interdisciplinary Learning and MYP Science Teacher, Atlanta International School.