Final Report

Group 2

111511245 Chu Yu-An

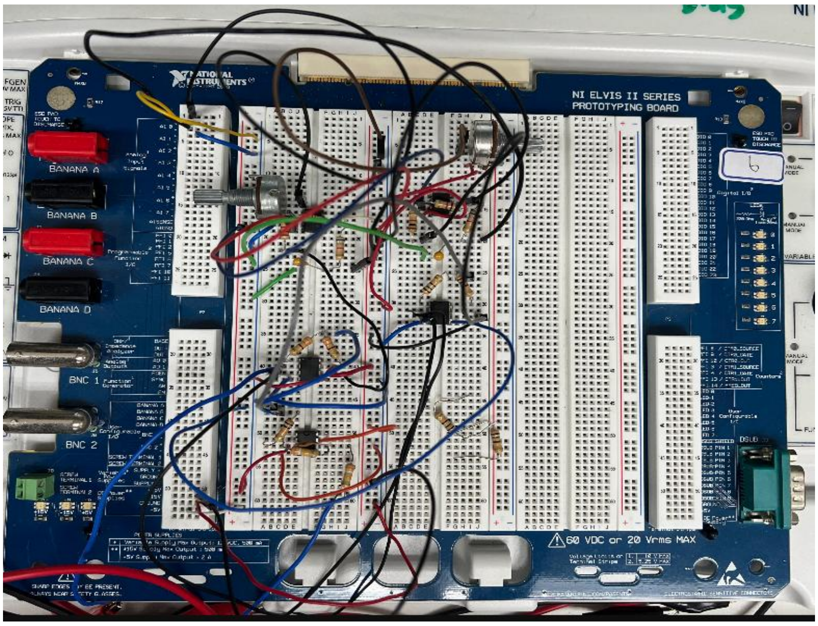
Introduction

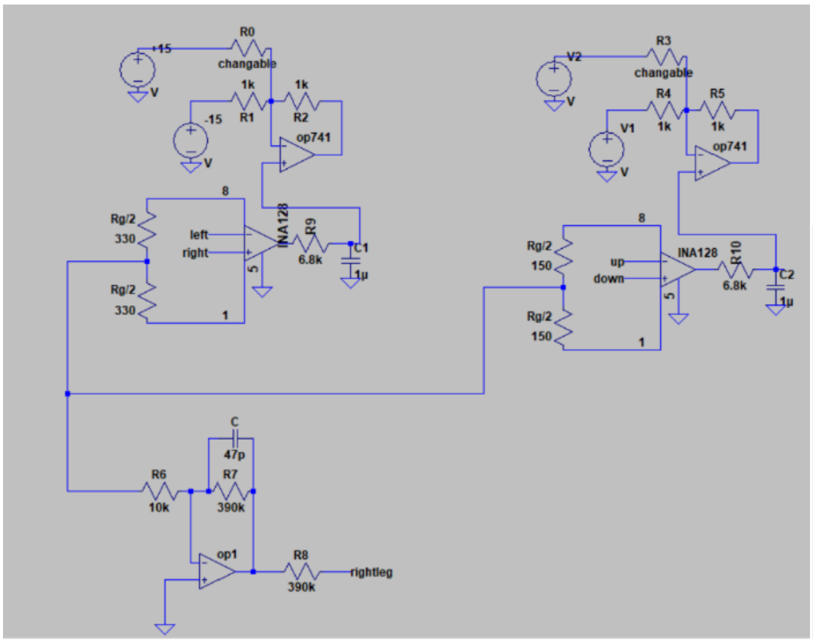
Since we know there are many ALS and aphasia patients suffering from communicating with medical crew, developing a system to assist them in through ways apart from speaking is an essential issue. The solution we came up with is to capture eye movement to read their need through monitor, which can significantly improve their quality of life and reduce the burden on caregivers and families.

To achieve the goal, our group plan to develop an eye-tracking-based communication system, which can enable the expression of basic needs, such as nature's calling or assistance requests. To meet their needs precisely, we need to evaluate the system's accuracy, usability, and user satisfaction. There are some instruments can be used to fix the problems, such as Tobii and Pupil Labs, but we still want to try to fix this problem by knowledge in BME lab. Thus, the focus of our project is to detect EOG signals of the patient, analyze it, and recognize the types of assistance they need.

Methods

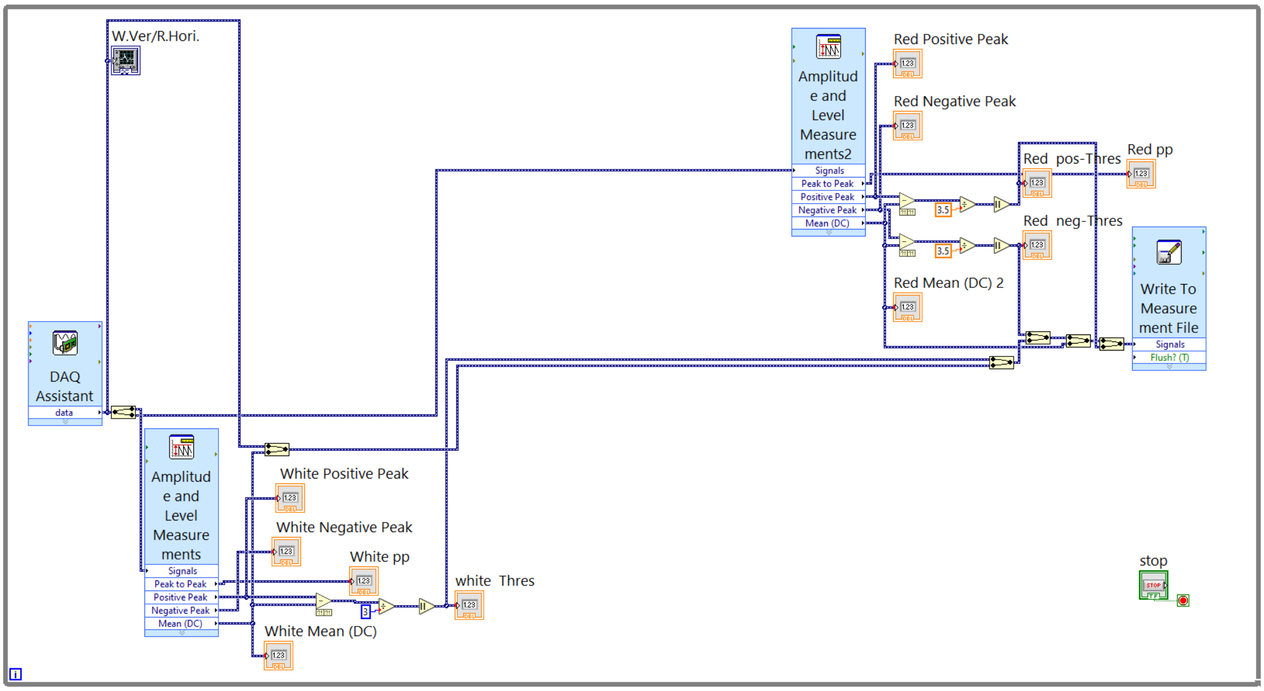
From what we've learned in Lab 8, the circuit we modified is as follows：





Rg(水平)	660歐姆
Rg(垂直)	300歐姆
$C_{濾波}$(水平)	1uF
$C_{濾波}$(垂直)	1uF
$R_{濾波}$(水平)	6.8k 歐姆
$R_{濾波}$(垂直)	6.8k 歐姆
3dB	23.417 Hz

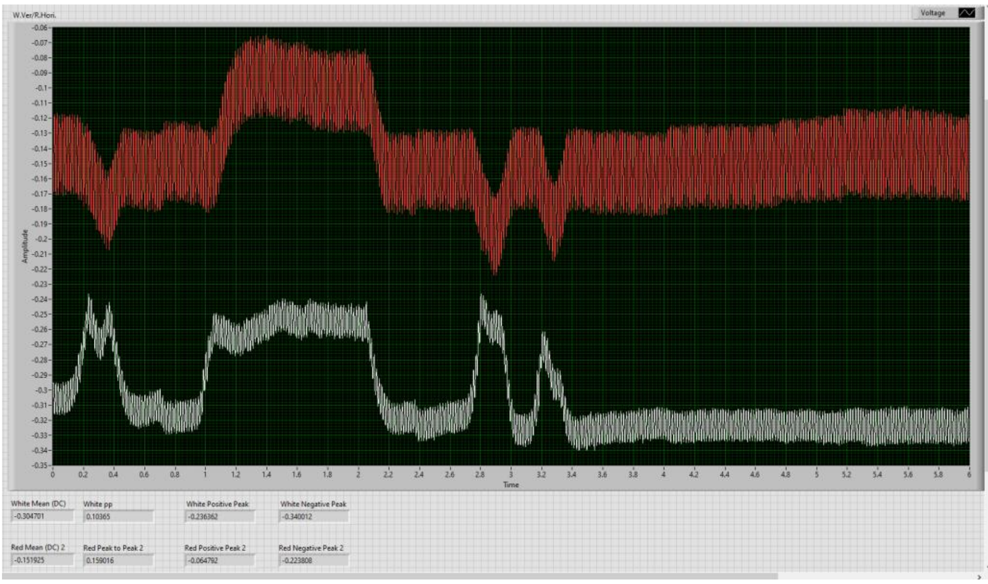
As the picture above shows, we connect two EOG detection circuits to one shared right-leg drive circuit whose function is to erase common mode noise, improve signal quality, and offer a virtual ground for EOG system. The two EOG systems are for horizontal and vertical eye movement signals individually, so we can get data from both directions. After we construct the circuit, we then recruit some volunteers to join our experiment. The procedures can be divided into five parts：



First, connect the circuit to LabVIEW software above in the computer. We will split the signals into two channels and define the red one as the horizontal signal and the white one as the vertical signal. We can divide the original EOG signal into two direction and analyze it, and thus we can react to the patient's demand and meet their needs to their satisfaction.

Secondly, initialize the oscillator. In this step, we need to erase the DC offset, and there are two ways to achieve it. We can connect variable resistors (R_0 、R_3) to $V^+ \ or \ V^-$ for roughly adjustment, and change values of the variable resistors to a specified value to erase DC offset at most.

(Before initialization)



(After initialization)

Then we can measure the signal and test the accuracy of our EOG circuit. After the initialization in the previous step, we will ask the subject to perform the following steps：

1. Look to the front：Looking to the front is to define the standard potential, which enable us to make sure that we can detect the signal’s accurate direction.
2. Blink：It is the sign that our measurements began, and we should detect the following EOG signal in both directions.
3. Look in the target direction：

Starvation	Roll over	Thirst
Number two	Reference	Number one
Change clothes	Sponge bath	Change pants

   Just like the above form, we will previously design a list of demands and utilize the EOG signal to acquire the true intend for the patients. For different tasks, the subjects need to look at different direction, which shows his or her needs. The principle to define the needs is similar for both channels. If the subject looks up or look left, the action potential will increase. Similarly, if the subject looks down or right, the action potential will decrease. Combine both channels, we can define eight directions like two crosses and thus confirm the correspond need.

4. Look to the front again：This step is to let us know the action potential that symbolize the patients’ needs came to an end and be ready to receive signal to end the examination.
5. Blink again：Like the previous blink signal, blink signal mark the start and the end of our exanimation. The former one is for the start sign and the latter is for the termination.

Next step, we will derive the result of our experiment by exporting excel files, and we need to link the output excel document to Python file manually. The output will then be shown in the laptop by picture and terminal port simultaneously. Hence, we can read the intend of our subjects accurately. I will introduce the Python code later in discussion, and the above is the procedure of our experiment.

	A	B	C	D	E	F	G
1	Voltage (Positive I	Voltage	Voltage_0	Voltage (DC Voltage)	Voltage_0 (Negative Peal	Voltage_0 (DC Voltage)	Voltage_0 (Positive Peak)
2	0.036808	0.826216	1.110129	0.843584	0.010145	1.121573	0.019148
3		0.830079	1.102726				
4		0.829113	1.106266				
5		0.822997	1.119786				
6		0.814628	1.136847				
7		0.809477	1.147791				
8		0.810765	1.144894				
9		0.817203	1.130731				
10		0.824928	1.114314				
11		0.830079	1.104013				
12		0.830079	1.104013				
13		0.824928	1.114636				
14		0.816559	1.131696				
15		0.808834	1.14586				

(Example form of output excel file)

Finally, to get the result, we will need to type something for making sure that we are analyzing the correct file. As the following two picture shows：

1. Choose which mode we are going to use. Mode 1 is to decide the data path of our input excel file, mode 2 is to type that where we plan to store our output picture, and mode 3 is to exit the program. And if we didn't type anything in Mode 2, the program will automatically set its output data path to be "C:\Users\Administrator\Desktop\.png".

```
--- EOG Event Detection ---
1. Analyze an Excel file
2. Set default output path
3. Exit
Enter your choice: 1
```

2. After choosing the mode, we enter the path to the excel file, and we can choose to enter the output path or not, which will not have any influence on the result.

```
Enter the path to the Excel file: C:\Users\Administrator\Desktop\excel\伊凡右下.xlsx
Enter the output path for the plot:
```

3. Finally, it will show our result on both terminal port and the picture, which will be illustrate in next paragraph.

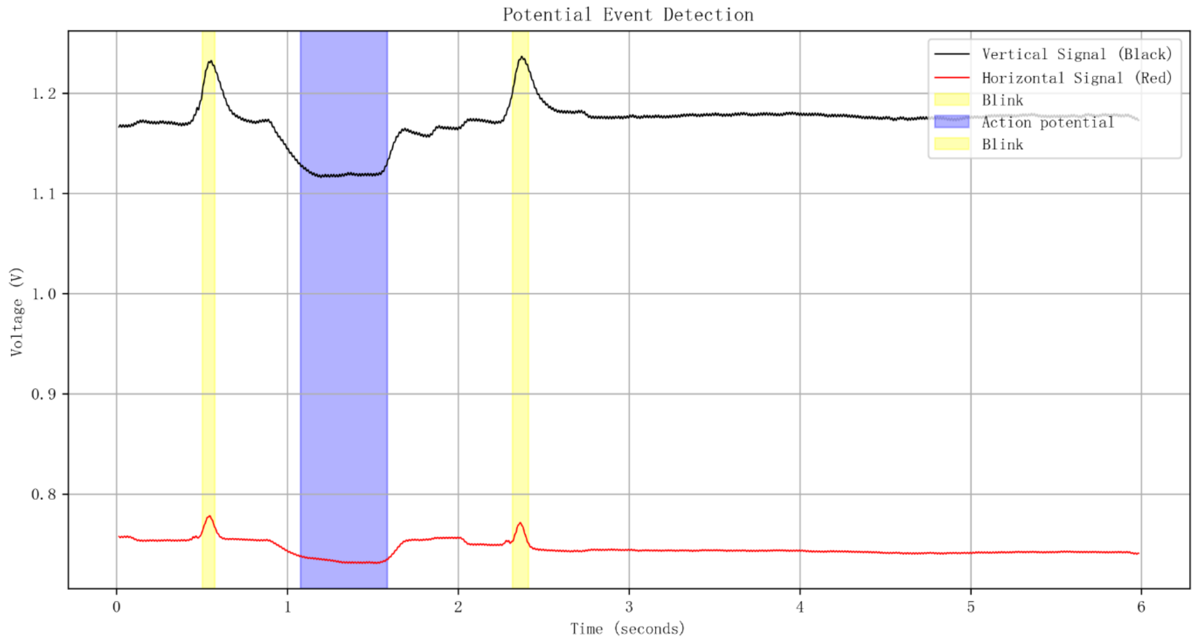
Results

The example output is shown in the following two pictures：

```
--- EOG Event Detection ---
1. Analyze an Excel file
2. Set default output path
3. Exit
Enter your choice: 1


Enter the path to the Excel file: C:\Users\Administrator\Desktop\excel\伊凡右下.xlsx
Enter the output path for the plot:
=== Detected Events ===
Event: Blink - Start: 0.502s, End: 0.576s
Event: Action potential - Start: 1.076s, End: 1.586s, Direction: Vertical: Looking down, Horizontal: Looking right
Event: Blink - Start: 2.318s, End: 2.412s
```

(Terminal port)



(Output picture)

The above two pictures show an example of looking at lower right. As shown in the output picture, there are two blink and one action potential in our experiment. Also, the terminal port shows the correspond output in words. Combining the principle from the above, we can speculate that the subject wants to change his or her pants. In conclusion, this is how we use EOG signal to detect patients who suffer from ALS and aphasia and thus meet their needs accurately.

Discussion

In this part, I’m going to (a)introduce and verify the usage of our project result and then (b)discuss some problems and difficulties we met.

(a)

From the result part, we can assume that our system is able to detect the signal and show the subjects’ needs. To verify its accuracy, we collect 4 subjects’ data to test the stability and reliability of our circuit. The following form shows that one of the subjects’ s data, and specify the accuracy：

Actual Direction	Blink	Experiment	Accuracy (Ver.)	Accuracy (Hor.)	Accuracy
Up	O/O	Up	O	O	O
Down	O/O	Down	O	O	O
Left	O/O	Left	O	O	O
Right	O/O	Right	O	O	O
Upper Left	O/O	Left	X	O	X
Upper Right	O/O	Upper Right	O	O	O
Lower Left	O/O	Lower Left	O	O	O
Lower Right	O/O	Down	O	X	X

We can see that the form indicates the result of our experiment, which shows the accuracy of blink, vertical, horizontal, and its summarization. It is obvious that if one of the directions is not detected correctly, we will count that one as failure. Just like the above form, we have other subjects’ data collected in the following link：

https://docs.google.com/document/d/1gG445lJaz2jO6pip3LLrXxOL4WkTN_2F/edit?usp=sharing&ouid=117590191802868281375&rtpof=true&sd=true

In short, we have accuracy of 65.625% for the full test, 84.375% for the vertical direction, and 78.125% for the horizontal direction. Before the experiment, we expect 80% of total accuracy.

However, the result shows that we only achieved 80% accuracy of each direction, but the total accuracy is not really satisfying. In conclusion, if we use this system to try to realize the needs of ALS and aphasia patients, we may not be able to acquire the correct result so accurately like other products (others' research result in reference), and the reason will be explained later in next paragraph. Since my main job is to design Python program, I will mainly discuss the problem I met during the designing process, and the possible reason why our accuracy didn't meet our expectations.

Our Python code can be mainly divided into four parts：collecting data, parameters rule, analyze data, and output form. This code also contains other parts like libraries, plot, and picture, but since it is common for signal analysis, I will not explain that in detail. Furthermore, the whole source code will be put in the following link：

https://drive.google.com/file/d/1Mdh0z_5t2NsSVHo8avdp_XHbnxG55OWv/view?usp=sharing

1. Collecting data：Since our data collected from LabVIEW is 6 seconds long in total, containing 3000 numbers of signal values. In other words, it is 0.002 seconds per signal value. After reading necessary data from excel file, we will set window size to be 15, which means that we see 0.03 seconds as one signal value to analyze it later. Also, we use "moving average" to certify the adaptability and reliability of constantly changing value of average. The following is the function of each input parameters.

Data name	Function
voltage	Input vertical signal
voltage_0	Input horizontal signal
voltage (dc voltage)	Average input vertical signal
voltage_0 (dc voltage)	Average input horizontal signal
voltage (positive voltage)	Deciding vertical threshold
voltage_0 (positive/negative voltage)	Deciding horizontal threshold

```python
# Check required columns
required_columns = [
    'voltage', 'voltage_0',
    'voltage (dc voltage)', 'voltage_0 (dc voltage)',
    'voltage (positive peak)', 'voltage_0 (negative peak)',
    'voltage_0 (positive peak)'
]
if not all(col in data.columns for col in required_columns):
    raise ValueError(Fore.RED + f"Excel file is missing required columns: {required_columns}")

# Assume each data point has a time interval of 0.002 seconds
time_interval = 0.002
data['time'] = [i * time_interval for i in range(len(data))]

# Smooth the data
window_size = 15
data['voltage_smooth'] = moving_average(data['voltage'], window_size)
data['voltage_0_smooth'] = moving_average(data['voltage_0'], window_size)
```

2. Parameters rule：After dealing with some input data, we need some parameters to define our baseline and threshold value. Applied in the previous step, vertical and horizontal baselines are defined by “voltage (dc voltage)” and “voltage_0 (dc voltage)”, and both directions’ threshold are defined by “voltage (positive voltage)” and “voltage_0 (positive /negative voltage)”. As for horizontal direction, it is worth noting that there are two kinds of peak to choose from. Since the horizontal signal is not as strong as vertical one, we will adapt the stronger one (identify the correct threshold) to be our horizontal threshold in case of failing to recognize the signal, and the adapted one will be named “t_horizontal _threshold”. Besides, there are three other parameters that we need to explain its usage.

    First, “threshold_duration” defines that whether our action is blink or not. If the duration that the difference of current vertical voltage value and vertical baseline value is larger than vertical threshold value is shorter than threshold duration, the program will define the signal as blink signal. If not, it will be action potential. After experiment, we find it best to set our “threshold_duration” to be 0.4 second.

    Secondly, “cooldown_time” is set because that the action potential will be chaotic for a while. In case that we capture the unwilling signal; we need to skip some signal in a specific duration. According to our experiment, the best “cooldown_time” is about 0.5

second. Thus, we set it to avoid false detecting.

Finally, to capture the correct signal, we will avoid rising and falling time, which we call it “time_advance”. When detecting rising signal, we will skip some initial signal. When detecting falling signal, we will skip some signal from the back. In other words, we will only capture the signal in the middle of action potential. And after doing some test and observation, we will define “time_advance” to be 0.05 second.

```
# Read baseline and threshold values
vertical_baseline = data['voltage (dc voltage)'].iloc[0]
horizontal_baseline = data['voltage_0 (dc voltage)'].iloc[0]
t_vertical_threshold = data['voltage (positive peak)'].iloc[0]
if data['voltage_0 (negative peak)'].iloc[0] < data['voltage_0 (positive peak)'].iloc[0]:
    t_horizontal_threshold = data['voltage_0 (negative peak)'].iloc[0]
else:
    t_horizontal_threshold = data['voltage_0 (positive peak)'].iloc[0]

# Set parameters
threshold_duration = 0.4  # Duration threshold to distinguish blink from action potential
cooldown_time = 0.5  # Cooldown time after action potential (seconds)
time_advance = 0.05  # Advance end_time by 0.05 seconds
vertical_threshold = t_vertical_threshold
horizontal_threshold = t_horizontal_threshold
```

3. Analyze data：This part is mainly divided into two steps：blink detection and action potential detection. Just like the above mentioned, only when we detected the signal of blink will we get to testify whether the next voltage change is action potential. Then, we will compare the difference between current voltage value and voltage baseline value to threshold value. Because the horizontal signal is weaker, we need to make the threshold value to be 3 times larger so that we can detect the horizontal action potential more precisely. As for the reason why it is 3 times, it’s from the numerous try and errors. After discussing and verifying, we make it 3 times bigger to make it more accurate. Moreover, the main discrepancy between blink and action potential is the time that the voltage difference exceeds the threshold value. If the time exceeds threshold duration, it is defined to be action potential; If not, it is blink signal.

```python
# Determine event type
if duration <= threshold_duration:
    event_type = "Blink"
    direction = None
else:
    event_type = "Action potential"
    direction = "Vertical: " + ("Looking up" if end_vertical > vertical_baseline else "Looking down")
    if abs(end_horizontal - horizontal_baseline) >= horizontal_threshold:
        direction += ", Horizontal: " + ("Looking left" if end_horizontal > horizontal_baseline else "Looking right")
```

4. Output form：It is an easy part, but to make the users easier to recognize the different information. Blue indicates the process and input information. Green separates output and other information. Yellow marks the blink event, and red marks the action potential's start time, end time, and both directions. The form and code are shown in the following picture.

```python
# Display event list
print(Fore.GREEN + "=== Detected Events ===")
for start_time, end_time, event, direction in events:
    if event == "Blink":
        print(Fore.YELLOW + f"Event: {event} - Start: {start_time:.3f}s, End: {end_time:.3f}s")
    else:
        print(Fore.RED + f"Event: {event} - Start: {start_time:.3f}s, End: {end_time:.3f}s, Direction: {direction}")
print("\n")
```

```
Enter the path to the Excel file: C:\Users\Administrator\Desktop\BME lab\excel\伊凡右下.xlsx
Enter the output path for the plot:
Analyzing the file and detecting events...
=== Detected Events ===
Event: Blink - Start: 0.502s, End: 0.576s
Event: Action potential - Start: 1.076s, End: 1.586s, Direction: Vertical: Looking down, Horizontal: Looking right
Event: Blink - Start: 2.318s, End: 2.412s
```

(b)

There are three significant difficulties we met in our experiment：

First, horizontal signal is not strong enough. From Lab8 we know that horizontal signal is always not as strong as vertical one. Though varying from each subject, both blink and action potential threshold are hard to define in horizontal direction. We came up with the solution to this problem is that measure the EOG signal of only left eye instead of on both eyes. Hence, we can capture signal that is stronger and more specific for the subject.

Secondly, each subject's parameters are not all the same. This problem can be solved by setting the parameters in advance. Like "threshold duration", "cooldown_time", and "time_advance", these parameters can be measured before application to the patient. By experiment in advance, we can get more accurate parameters for each subject, and then we can adjust them to a more precise level.

Last but not least, the difference between each subject's action potential standard is the key to the common usage of this system. The reason why it is hard to define the standard is that everyone's angle for looking at each direction is not all the same. If A looks right at 40 degrees, we can't use this standard to do the same task for B, who looks right at 30 degrees. Otherwise, it will severely effect the accuracy for our system. The only way to solve this problem is to define each subject's best fit standard in advance of the examination, so that we can clearly know how the subject's eyeball moves.

The above are three main problems we met, and there are still some difficulties we need to conquer. The most significant one is that how we deliver the needs of the patients to those who take care of them in real time. Our system requires many manual steps, and it is hard for every patient to tell their needs to medical crew or their family when it is emergent. So far, we haven't fix this yet. Maybe it requires more time to ensure the link between LabVIEW and Python is in real time, which can deliver the needs in time instead of using manual way to collect and analyze data.

Finally, the accuracy is not as high as we expected before final project, which we expected 80% accuracy in advance, and it turned out to be 65 percent. However, I think it resulted from the parameters and limited database. Once we adjust the parameters more precisely in advance of the experiment and recruit more volunteers to test our system, it will solve the above-mentioned problems to a quite large extent.

Conclusion

Just like the above-mentioned, we have accuracy of 65.625% for the full test, 84.375% for the vertical direction, and 78.125% for the horizontal direction. Thus, the system can be used to help ALS and aphasia patients to express their needs to some extent. Nearly 66% of accuracy shows that we don't have too much positive confidence that we can realize the function that we expected before experiment. In conclusion, our system can help the patients, but it still requires some adjustment.

These adjustments can be in parameters, functions, and the instruments, which all require further test and verifying. Our group is not so satisfied with the result, but we believe that it's the best performance in limited time.

Division of Labor

Member	Labor	% of contributions
伊凡沙韻	LabVIEW design、Measurement、Taking demo videos	33.3%
朱昱安	Topic inspiration、Measurement、Python programming	33.3%
薛油和	Circuit design、Measurement、Python revision	33.3%

Just like the above form, our group's division of labor is clear. I'm responsible for topic inspiration and Python programming. 伊凡沙韻 is responsible for LabVIEW design and taking demo video. 薛油和 is responsible for circuit and Python debugging. Thus, I think the contributions of each member is roughly the same. I came up with the final topic and spent most of my time writing the whole Python code, 伊凡沙韻 does almost every part in LabVIEW, and 薛油和 not only design the circuit but also help me debug my code. I'm glad to have such good partners, and I hope that we have some opportunity to cooperate with each other in the future.

References

- An Electro-Oculogram (EOG) Sensor's Ability to Detect Driver Hypovigilance Using Machine Learning

  https://pmc.ncbi.nlm.nih.gov/articles/PMC10058593/

- EOG-based eye movement detection and gaze estimation for an asynchronous virtual keyboard https://www.sciencedirect.com/science/article/pii/S1746809418301757

- Comparing In-ear EOG for Eye-Movement Estimation With Eye-Tracking: Accuracy, Calibration, and Speech Comprehension

  https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2022.873201/full

- 有口難言之疾 – 失語症

  https://www.ntuh.gov.tw/neur/Fpage.action?fid=4326

- 運動神經元疾病(漸凍人)

  https://www.mnda.org.tw/disease_main.php

Source link

Accuracy result：

https://docs.google.com/document/d/1gG445lJaz2jO6pip3LLrXxOL4WkTN_2F/edit?usp=sharing&ouid=117590191802868281375&rtpof=true&sd=true

Source code：

https://drive.google.com/file/d/1Mdh0z_5t2NsSVHo8avdp_XHbnxG55OWv/view?usp=sharing

Demo video：

https://www.youtube.com/watch?v=eACNCWlBFQs&ab_channel=%E4%BC%8A%E5%87%A1%E6%B2%99%E9%9F%BB