

ORIGINAL ARTICLES

Mind-Wave Controlled Robot: An Arduino Robot Simulating the Wheelchair for Paralyzed Patients

Chi Hang Cheng¹, Shuai Li*¹, Seifedine Kadry²

¹Department of Computing, Hong Kong Polytechnic University, Hong Kong, China

²Beirut Arab University, Beirut, Lebanon

Received: May 7, 2018

Accepted: June 6, 2018

Online Published: June 27, 2018

DOI: 10.5430/ijrc.v1n1p6

URL: <https://doi.org/10.5430/ijrc.v1n1p6>

ABSTRACT

This project attempts to implement an Arduino robot to simulate a brainwave-controlled wheelchair for paralyzed patients with an improved controlling method. The robot should be able to move freely in anywhere under the control of the user and it is not required to predefine any map or path. An accurate and natural controlling method is provided, and the user can stop the robot any time immediately to avoid risks or danger. This project is using a low-cost brainwave-reading headset, which has only a single lead electrode (Neurosky mind wave headset) to collect the EEG signal. BCI will be developed by sending the EEG signal to the Arduino Mega and control the movement of the robot. This project used the eye blinking as the robot controlling method as the eye blinking will cause a significant pulse in the EEG signal. By using the neural network to classify the blinking signal and the noise, the user can send the command to control the robot by blinking twice in a short period of time. The robot will be evaluated by driving in different places to test whether it can follow the expected path, avoid the obstacles, and stop on a specific position.

Key Words: Rain-computer interface, Electroencephalogram, Neural network, Neurosky sensor, Wheelchair

1. INTRODUCTION

Paralyzed patients face many difficulties in their daily life. It is hard for them to make use of motor neurons to control muscle. People suffer from motor disabilities may sometimes be very stiff and even cannot speak as they want. They need the help from others to perform daily activities. For example, fully paralyzed patients may need someone's help to control the wheelchair. In the past, many technologies have grown and become mature for disabled people to interact with physical devices, such as the electromyogram (EMG) arm, finger gesture recognition application and voice-controlled wheelchair.^[1] However, most of them are relying on muscles, body movements or speech commands. Obviously, they are not convenient for paralyzed people perform these actions.

Advances in the neural network and human computer interaction technologies have caused concern to brain computer interface (BCI).^[2] By employing BCI technology, human can use brain wave to interact with physical devices easily. In this project, authors will make an Arduino robot car that controlled by human brain wave using the BCI technique. Arduino Mega is chosen because it is a low-cost microcontroller and it is more powerful than an Arduino UNO.^[3] The human brain wave will be captured using a low-cost Neurosky mind wave headset.^[4] The techniques used in this project can be further extended to a wheelchair for paralyzed people.

The objective of this work is to implement an Arduino robot to simulate a brainwave-controlled wheelchair for paralyzed patients with an improved controlling method. The outcomes of this project should fulfill all the following requirements. In

*Correspondence: Shuai Li; Email: shuaili@polyu.edu.hk; Address: Department of Computing, Hong Kong Polytechnic University, Hong Kong, China.

terms of mobility, no any map or path need to be predefined and the robot should be able to move freely in anywhere under the control of the user. In terms of accuracy, a controlling method with at least 85% of accuracy should be adopted. In terms of safety, an immediate command should be provided to stop the robot immediately to avoid risks or danger. In terms of cost-effective, the time cost of each controlling command selection should be less than 1 second. In terms of simplicity, the robot should provide a natural controlling method that does not require left blinking or right blinking. Autonomous obstacle avoidance should be included to avoid the jerky movement.

2. NEUROSKY MIND WAVE HEADSET

Brain computer interface provides a communicating method between physical devices and human brain.^[5] This technology used the neural activity of the human brain as a signal. The objective of BCI is to create an output channel for the brain to benefit the people suffered from motor disabilities, so that they can control the physical devices with their mind.^[6] However, BCI is not a very stable interface and most of the time it needs some sort of reliable control.

Electroencephalography (EEG) is the most studied non-invasive interface of BCI. It is an instrument to record the electrical activity of the human brain. The EEG signal is represented in waveforms and they have a random and small amplitude. Generally, waveforms can be classified into several power spectrums such as alpha (closed eyes and relaxed), beta (thinking), theta (sleepy or disappointed), and delta (sleeping).^[7] Although this non-invasive brain wave measurement is not harmful to the body, the measurement is inaccurate because the noise of brain wave keeps producing miscellaneous signals. NeuroSky mind wave headset (see Figure 1) is a low-cost module for reading mental activities of the human using a single lead electrode. According to a BCI hardware study, the neurosky mind wave headset is the lowest price modular EEG device with the highest usability score.^[8] By recording the EEG signal, this headset is able to identify EEG power spectrum, and output the RAW signal, EEG power spectrum, attention strength and meditation strength.^[9] The raw brainwaves and the eSense data obtained from the NeuroSky mind wave headset are calculated to determine the strength of the eye blinks. It is possible to find other existing techniques for blink detection, such as EMG activity detection and image processing. However, the NeuroSky mind wave headset is more efficient than other eye blinks detection techniques and it requires fewer accessories with a faster processing time.^[10] The eSense value also indicating the strength of user's attention and meditation level. The range of the level is from 0 to 100. The measure-

ment of attention and meditation level are based on human mental activity. Tightening or relaxing the muscles may not have an immediate change in the strength of attention and meditation.^[11]



A single lead electrode on the EEG headset



TGAM chip

Figure 1. A NeuroSky mind wave headset

3. REQUIREMENTS OF PARALYZED PATIENTS

Paralysis refers to one or more muscle groups completely loss the muscle function. The patient may lose the feeling in the affected area which causes the mobility problem. For some of the patients, like the amyotrophic lateral sclerosis (ALS), the eye muscles are the only part that they can easily control even in the final stages of the disease.^[12] The reasons for paralysis are generally damaged the nervous system, stroke, trauma, poliomyelitis or ALS.^[13] A survey by the Christopher and Dana Reeve foundation indicated that in every 50 people, about 1 of them are paralyzed.^[13]

Paralyzed patients need help from others in the daily life. Especially the fully paralyzed patients, they require the support

at all the time. However, it is impossible to find someone that can provide the support in every moment. In the worst situation, they can only lie in bed when they are alone in a room. Although wheelchair is a good tool when walking is difficult or impossible, paralyzed patients cannot control them easily. For fully paralyzed patients, they can go to somewhere only under the help of someone. There is no doubt that a mind wave-controlled wheelchair can help in their daily life.

4. COMPARISON OF WHEELCHAIR CONTROLLING METHODS

Almost all the existing wheelchair controlling methods are relying on the physical motion of the human. Even the electric wheelchair, it required the control of the joystick. The muscles of the mouth are used for the speech-controlled wheelchair. These wheelchairs controlling methods are not suitable for the paralyzed patients in the final stage of the disease.

As mentioned in the problem statements, some of the paralysis like the ALS, their eye muscles are functional even in the final stages of the disease.^[12] Therefore, the preferable wheelchair controlling methods for them should be eye blinks and non-physical motion, like brain motion. A lot of techniques can be used for the eye blink detection while most of them have some limitations. One of a new approach is to use a build-in webcam for the face and eye detection base on the image processing, but the first tests showed some people cannot successfully control and communication through this solution.^[14]

Using image processing techniques to detect the eye blinks also has some limitations. First, the face should always be steady in a specified position. Otherwise, the camera cannot take a clear frame for computation. The second limitation is adequate light should be prepared for the detection. Furthermore, it is hard to implement to a real-time system as the time cost is varied. Based on these limitations, this design is not an effective way to obtain eye blinks for controlling the wheelchair. Another approach that commonly adopted at present for the eye blink detection is using three small electrodes.^[15] These electrodes are stuck to the human skin around the orbicularis oculi muscle to get the electromyography (EMG) data. Obviously, an EMG system can detect the eye blinks efficiently, but this approach is not very accurate.^[10] The accuracy of EMG system is easily affected because of variability skin conductance and sight positioning changes. The raw EMG data of the muscles are hard to obtain. It is because the baseline will easily drift when patient changing the sight positioning. Moreover, the skin conductance is not a constant as it can change dynamically,

and the biggest challenge of EMG system is the electrical noise.^[10] Therefore, the idea of EMG system is possible, but it is not an effective way to detect the eye blinks.

As mentioned the most commonly used techniques for blink detection before, it is obvious that the existing blink detection approach required lots of data and algorithm to handle the identification of eye blinks. A new way to detect eye blinks is using the EEG signals. By employing this method, only one brain wave reading device is required for the whole system to extract the EEG signals. Compared to use the built-in webcam with various lights, employing EEG approach has a higher mobility. This new technique can also overcome the deficiencies of the electromyography and image processing techniques for detecting blinks. For example, according to the website of Neurosky, the accuracy of the brainwave data obtained is about 96%.^[16] And this method involved fewer algorithms and computation, which means it has a less time cost and able to command the wheelchair rapidly.^[10] Also, it is a lightweight device which required fewer accessories.^[10] Hence, compared to the existing blink detection approach, analyzing the EEG signals is more efficient and effective. It should be the best solution to employ for controlling the wheelchair.

5. COMPARISON OF EEG WHEELCHAIR SYSTEMS

Brice Rebsamen and his team introduced an indoor wheelchair controlling method using thought.^[17] This system was introduced in 2007. The idea of this system is to build a mind-controlled wheelchair that able to navigate inside the hospital or a typical house autonomously. The control mechanism of this system is based on the P300 EEG BCI which allows the patients to choose a destination listed on the menu. The destination listed on the menu will flash one by one. By identifying the positive potential peak of the EEG signal, the system can know the patient is focusing on which destination. Although the system provides a simple controlling method, the usage of the wheelchair is only limited to a specific environment with predefined paths. If the environment changed, a new map is required to load into the system. Moreover, the wheelchair can only arrive at a specific point, but the patient cannot adjust the position. It seems that the usage environment is the biggest limitation of this system.

A new wheelchair controlling method through thought has been introduced by Vaibhav and his team.^[18] A monitor module is used in this system for patients to select the movement option. The selection pointer moves from column to column with a defined time interval. Patients can perform left-hand

motor imagery (MI) or right-hand motor imagery to select the row of options. If the patient does not take any MI in the time interval, the pointer will move to the next column. This system allows patients to control the wheelchair freely by selecting the command, but the time cost of each command selection may take up to 7 seconds. Also, the time cost will be double if the patient forgets to take an MI or the MI cannot be identified by the system. Although wheelchair is slow for safety purpose, the time spends of command selection should be minimized to enhance the cost-effective.

Arzak and his team introduced a method using single lead EEG module provided by Neurosky to collect the attention level of the patient.^[9] The attention level is used as a command of controlling the wheelchair to move forward, left or right according to the range of attention level. Although this system provides a real-time controlling method which is cost-effective, it is tired for patients to keep in a high attention level. Moreover, the attention level may easily be affected by emotions.

Ramya Stephygraph and his team purposed an eye blinks commanding method to control the wheelchair effectively.^[19] They used the NeuroSky mind wave sensor to obtain the eye blinks information by recording the EEG signal. Various eye blink characteristics can be used for different commands. Such as the frequency, strength and time duration. Although this controlling method seems cost-effective, it is not very convenient for the patients to perform only left blink and right blink. Moreover, the accuracy of the system is still a question. The success rate of the command has not been mentioned. It is also possible that the success rate will drop because of the inconvenience left blink and right blink. Besides, the stress blink is also hard to perform for some patients as this action may require tightening the muscles.

Base of the analysis of the related works. We can find that these systems have a lot of limitations. The most serious problem is first three systems do not provide an immediate command to stop the robot. This limitation may cause the safety issues. And these three systems required to wait for a long time to do the command selection, the time cost of each controlling command is too long so users cannot control the robot in real-time. And the last two systems have not included autonomous obstacle avoidance, so the user may need to send lots of commands in order to avoid the obstacles. Finally, users may have some possible difficulties when controlling the robots because the controlling method requires the user focus, imagine, keep in a high attention, and perform unnatural blinking like left blinking, right blinking or stress blinking.

6. SYSTEM ARCHITECTURE

Figure 2 showed the system architecture of this project. The system is formed by a Neurosky headset (EEG reader), a computer included MATLAB, and an Arduino robot car. The EEG reader contains a TGAM1 chip that can capture the human brain signal. TGAM1 also provided the signal filter and signal amplification. After that, the EEG signal will be digitized and sent to the Bluetooth transmission module (HC-06). Finally, the EEG signal will be transmitted to the computer for further analysis.

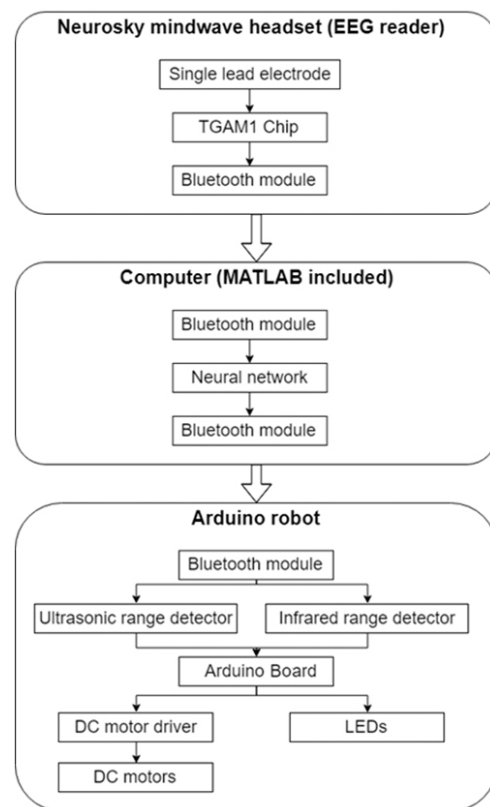


Figure 2. System architecture

MATLAB in the computer will be used for noise filtering. It will check whether the user is sending a robot controlling command, or the blinking are just natural blinking of human. If the blinking is belonging to a robot controlling command, MATLAB will also determine which command does the blinking representing.

The Arduino robot car is consisting of 7 components. They are Arduino Mega, one Bluetooth HC-05 module,^[20] two DC motors, one L298N DC motor driver,^[21] five ultrasonic range detectors (SR-04), three infrared range sensors, and one LEDs board. When the Bluetooth HC-05 module received the controlling command from the computer, the Arduino board will consider the data received from the ultrasonic

sensor and infrared sensor to make the final decision of the car movement. The ultrasonic sensors will detect the obstacles and avoid it automatically. The infrared sensors will detect the distance between the robot body and the ground to prevent falling from the stair. If the command from eye blinking is received, the robot will follow this command.

Otherwise, maintain the motion in the last time step and run obstacle avoidance. And the LEDs will display the status of the controlling command (e.g. the front LED will on when the user sends a “forward” movement command).

Below is the flow chart showing how the system works (see Figure 3).

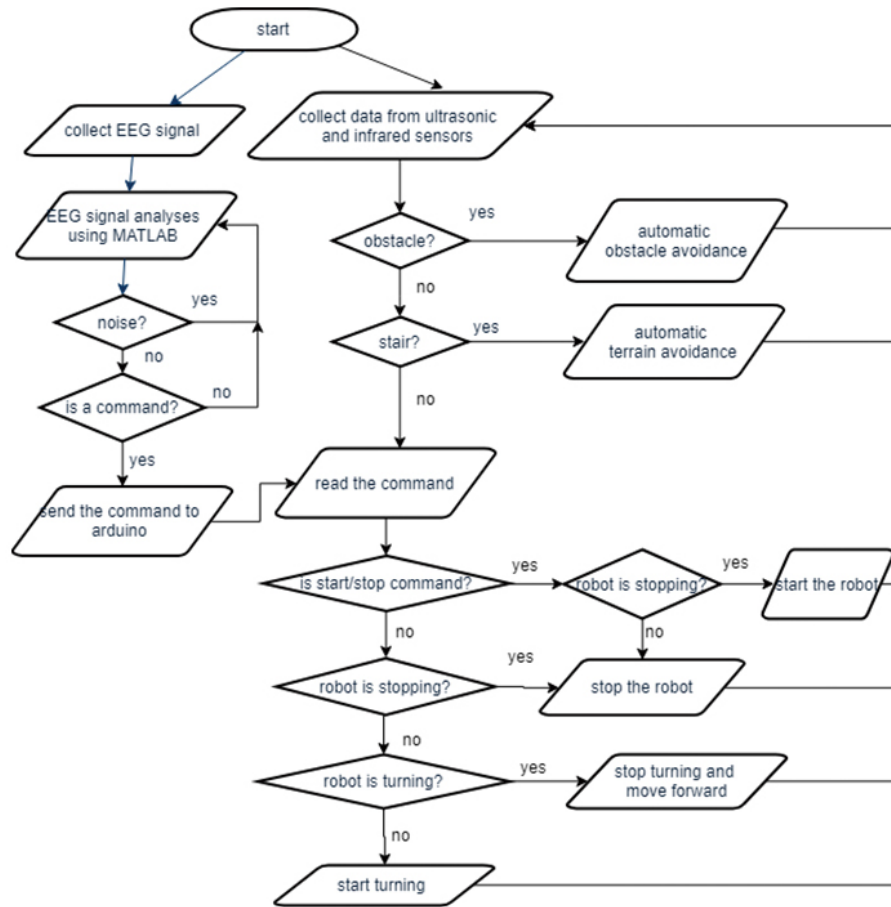


Figure 3. A descriptive flowchart

7. THE FIRST PROTOTYPE

For the first prototype, the RAW EEG signal can be collected and send to the Arduino through Bluetooth. After collecting the EEG signal, the Arduino will do the work of eye blinking detection by identifying the wave of eye blinking in the RAW EEG signal. Generally, when the eye blinking occurs, a pulse will be sent to the RAW EEG data, and this prototype is able to identify the pulse. After detecting the eye blinking, the robot will follow the eye blinking command to move in different directions. However, eye blinking is a natural activity of human. In order to identify whether the eye blinking is a controlling command or a natural human activity, some eye blinking identification strategy need to be used in the prototype.^[22] At the current stage, the eye blinking will be

considered as controlling command only if the two blinking were occurred within a short period and strength of that two blinking are higher than the natural blinking.

By default, the robot will stop until the user sends an eye blinking command to it. At the same time, the 4 LED installed on the robot will blink one by one with a short delay. 4 LEDs are working on the robot to represent the direction of forward, backward, left, and right. Users can blink twice in a short period to start the robot. The movement of the robot is depending on which LED is lighted up at that moment. At any time, the user can blink twice in a short period to change the direction of the robot. Once the user blinked twice, the movement of the robot will be changed immediately base on which LED is lighted up at that moment. As mentioned

before, a safety mechanism is the most important part of designing the robot. To avoid the danger caused by careless or incorrect control, the system required to provide an immediate command for the user to stop the robot. In this prototype, the user can blink three times or more in a short period to stop the robot. The robot will stop immediately, while the LED will keep lighting up one by one. User can blink three times or more in a short period to restart the robot. The movement of the robot is depending on which LED is lighted up at that moment.

This first prototype can work fine base on the eye blinking command. Although the accuracy is quite low, it can move in different directions and stop at any time. Also, it is a bit difficult for the robot to follow a specific path. It is because the 4 LEDs light up one by one with a short delay. Although the delay is short, the 4 LEDs light up one by one cost a huge of time. The user needs to wait until a proper LED light up and send the eye blinking command.

8. FOUR PROPOSED CONTROLLING METHODS

For the following prototypes, the stop/start function and eye blinking signal filtering functions are same as the first prototype mentioned above. For the brief testing part of the following prototype, the following common testing method will be used: level 1 (The robot should follow the rectangle drawn on the floor) and level 2 (The robot should follow the irregular path draw on the floor) are used.

8.1 Prototype using 7 LEDs

By default, the robot will move forward until the user sends an eye blinking command to it. At the same time, the LED installed on the robot will blinks one by one with a short delay. 7 LEDs are working on the robot to represent the direction of forward, various degrees of left, and various degrees of right. User can blink twice in a short period to change the direction of the robot. The movement of the robot is depending on which LED is lighted up at that moment. If the front LED lighted up, the robot moves forward. If other LED lighted up, the robot will turn in that direction in different turning degrees.

During the testing, although the accuracy is not very high, this prototype works fine base on the eye blinking command. Compared to the first prototype, it has more LEDs, so it can turn in a more accurate direction and do not require to stop and rotate. However, it is a bit difficult for the robot to follow a specific path. It is because the 7 LEDs lighted up one by one with a short delay. The user needs to wait until a proper LED lighted up and send the eye blinking command.

8.2 Prototype using 2 LEDs

By default, the robot will move forward until the user sends an eye blinking command to it. At the same time, the LED installed on the robot will blinks one by one with a short delay. 2 LEDs are working on the robot to represent the direction of left and right. User can blink twice in a short period to change the direction of the robot. The movement of the robot is depending on which LED is lighted up at that moment. If the left LED lighted up, the robot rotates left. If right LED lighted up, the robot rotates right. During the rotation of the robot, the user can blink twice at any time to end the rotation session. After that, the robot will go forward as a default. Again, during the testing, although the accuracy is not very high, this prototype can work fine base on the eye blinking command. Compared to the first prototype, it has fewer LEDs, so it can turn in just about one second, which is acceptable. Moreover, it can follow a specific path like a rectangle, and even some irregular path accurately. Overall, it is a good controlling method.

8.3 Prototype using stress blinking

By default, the robot will move forward until the user sends an eye blinking command to it. User can blink twice in a short period to change the direction of the robot. If the strength of first blinking is larger than the second one, the robot rotates left. If the strength of first blinking is less than the second one, the robot rotates right. During the rotation of the robot, the user can blink twice at any time to end the rotation session. After that, the robot will go forward as a default. Again, this prototype can work fine base on the eye blinking command. Compared to the other prototype, it does not rely on the LED. The user can send any command immediately and no need to wait for a proper LED light up. However, during the testing, the accuracy of the eye blinking command is a bit low because of the difficulty of performing the stress blinking.

8.4 Prototype using speed of blinking

By default, the robot will move forward until the user sends an eye blinking command to it. User can blink twice in a short period to change the direction of the robot. If the time of two blinking is less than 350 ms, the robot rotates left. Otherwise, if the time of two blinking is between 350-700 ms, the robot rotates right. During the rotation of the robot, the user can blink twice at any time to end the rotation session. After that, the robot will go forward as a default. During the testing, this prototype can work fine base on the eye blinking command. Compared to the other prototype, it does not rely on the LED. The user can send any command immediately and no need to wait for a proper LED light up. Moreover, it can follow a specific path like a rectangle, and even some

irregular path. Overall, it is also a good controlling method.

8.5 Comparison and summary of the prototypes

During the brief testing, method B and method D have the best performance.

For all the controlling methods proposed, an immediate command has been provided. The user can stop the robot at any time by blinking three times in a short period. Moreover, all the controlling methods proposed can move in different directions base on the command sent by users. It is not required to predefine any map or path.

Method B is relying on the LED and the user may need to take up to 1 second for the direction selection. It may see as a reasonable time cost even in a real situating when controlling the wheelchair,^[23] and the user can indicate a clear and accurate direction selection by the help of LED. Method D is not relying on the LED but relying on the speed of the eye blinking. The user does not need to wait for the LED light up to perform the direction selection. User can select the direction immediately by the rapid blinking. The speed of blinking determining the moving direction of the robot.

However, for all the prototypes above, the classification of the blinking and noise can be improved to enhance the accuracy. As in the prototypes, eye blinking will be considered as controlling command only if the two blinking is within a short period and strength is above the natural blinking, but some noise having the similar pattern of the blinking wave may be considered as human blinking which causes to the incorrect command.

To conclude, after implementing the controlling methods proposed, we can find that method B and method D have the best performance. However, in terms of immediate control, method D is better than method B. User may need to wait up to 1 second for the direction selection in method B, while method D does not. Considering some complicated path and the efficiency, method D is more safety and less time-consuming. Therefore, method D is chosen as the controlling method to be adopted in this project. However, in this stage, the accuracy of the prototype is still a big concern and it should be further optimized to have a better classification of the blinking and noise to enhance the accuracy.

9. NOISE FILTERING USING NEURAL NETWORK

The neural network has been adopted in this project to classify the blinking and the noise in the raw EEG signal. Below are some examples showing the wave pattern of human blinking and noises. A lot of noise will appear for several minutes after the user wears the headset (see Figure 4). Then, the

wave pattern will be stable.

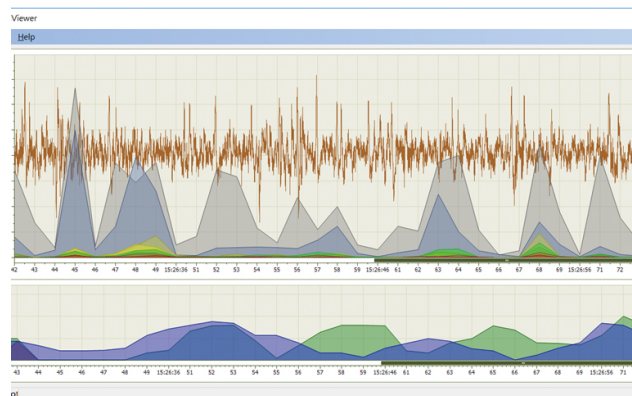


Figure 4. Example of EEG noise

In the graph below, waves labeled in green color are human blinking while red color is some significant noise that have a similar wave pattern as human blinking (see Figure 5).

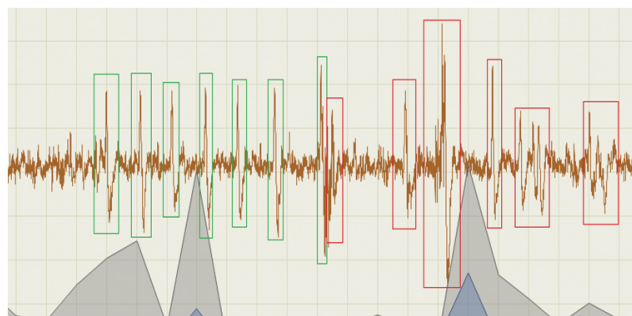


Figure 5. Example of blinking in EEG

In the prototype, eye blinking is considered as controlling command if the peak and trough of the wave are higher and lower than a specific value as the example (see Figure 6).

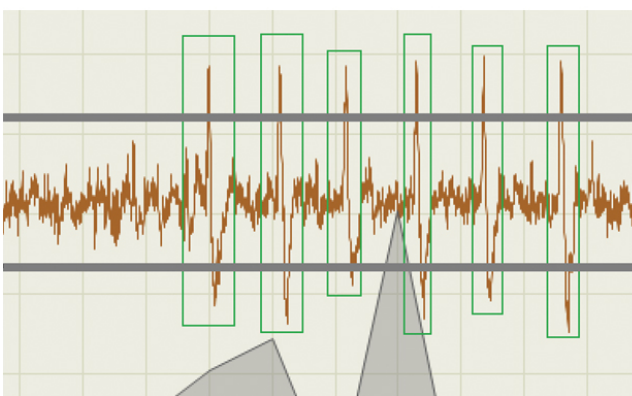


Figure 6. Noise filtering adopted in the prototype

However, the wave of noise may also have a peak higher than the specific value, so it is not a good classification method to identify whether the wave is human blinking or not. In order to have a higher accuracy in the robot controlling process, an artificial neural network was adopted.

In this project, MATLAB is used to build the neural network and send the controlling command after classifying the wave of noise and the human blinking. The wave pattern of human blinking will be further identified to different controlling commands, such as blinking twice rapidly belong to the turn left command, blinking twice slowly belong to the turn right command.

10. CLASSIFY THE BLINKING AND NOISE

Wave in the EEG signal is captured for two stages of filtering. Firstly, if the peak is lower than a specific value X as labeled in red color below, it will not count as a blinking (see Figure 7).

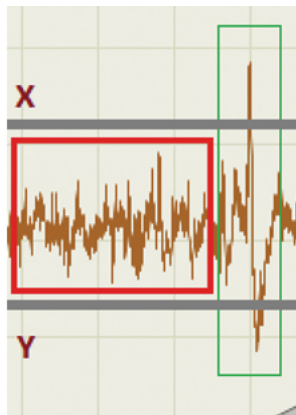


Figure 7. First-stage noise filtering

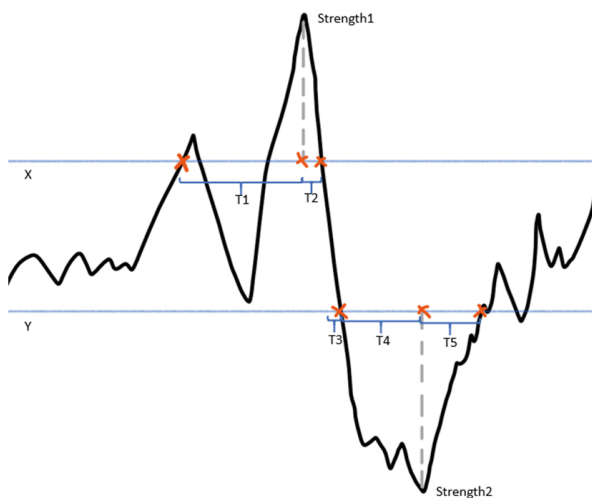


Figure 8. Wave pattern of a blinking wave

If the peak is higher than the specific value X, it will come to the next stage. At this stage, several points are used to record the wave patterns as the example (see Figure 8).

When the strength reaches value X, the timer will start to record the time until the peak is reached. The time spent in this part is represented by T1.

The peak of a wave is represented by Strength1. When the peak is reached, the timer will start again until the strength of wave drops back to value X. The time spent in this part is represented by T2.

When the strength drops below value X, the timer starts until the strength of the wave drops below value Y. The time spent in this part is represented by T3.

When the strength is below value Y, the timer starts until the trough is reached. The time spent in this part is represented by T4.

The trough of a wave is represented by Strength2. When the trough is reached, the timer starts until the strength of wave reach value Y. The time spent in this part is represented by T5.

Below are the pseudocode showing the wave pattern capturing process.

```

If(raw>X)
    Tic
    If(raw>peak)
        peak = raw
        T1 = toc
if(raw<X)
    T2=toc
    If(raw < trough)
through = raw
        T4 = toc
If(raw<Y)
    T3=toc
    If(raw < trough)
through = raw
        T4 = toc
If(raw>Y)
    T5 = toc
    wavePattern = (peak through T1 T2 T3 T4 T5)
    NetI= wavePattern
    
```

By collecting the time of points, peak and trough of blinking and noise, a training dataset can be formed to train a neural network to classify the blinking and noise.

A Levenberg-Marquardt backpropagation neural network function is used.^[24,25] According to the matlab documentation, it is the fastest backpropagation algorithm and is highly recommended as a first-choice supervised algorithm,

although it does require more memory than other algorithms.^[26] In this part, the neural network with 7 inputs, 4 hidden layer size is used. 75% of the dataset is used as training, while 15% is used as testing and 15% is used as validation.

The error rate of the training data, testing data and validating data are 3.59%, 3.22% and 4.97%, respectively, which is a great success in classifying the blinking and noise.

11. IDENTIFY THE COMMANDS

After classifying the blinking and noise, the blinking signal need to be further analyses to identify:

- (1) Whether the user is sending a robot controlling command, or the blinking are just natural blinking of human.
- (2) If the blinking is belonging to a robot controlling command, which command does the blinking representing.

To solve the first problem, the time between two blinking can be taken as a consideration to identify whether the user is sending a robot controlling command or not. Typically, the time between two natural blinking is greater than 1 second. While the robot controlling command requires the user blink twice or more rapidly. This differentiation is used to identify the natural blinking and the robot controlling command in this project.

To solve the second problem, the wave pattern and the time between two blinking can be taken as a consideration to identify which controlling command is sent by the user. Another neural network is created to solve these two problems. Firstly, if the wave pattern is classified as blinking, the wave pattern collected in the first neural network will be used again, but this time, two continues wave will be recorded which means we will have two datasets from the first neural network. Simply name that as blink_1 and blink_2. The time between two blinking will also be recorded.

Input of neural network II: [blink_1 blink_2 time]. Output of neural network II: different controlling commands.

A Levenberg-Marquardt backpropagation neural network function is used. The neural network with 15 inputs, 8 hidden layer size is used. 75% of the dataset is used as training, while 15% is used as testing and 15% is used as validation.

The error rate of the training data, testing data and validating data are 2.09%, 3.02% and 2.66%, respectively, which is a great success in identifying different commands.

Below are the pseudocode showing the wave pattern identification process.

```

if(isempty(blink_1))
    blink_1=wavePattern
    t1=clock
elseif(isempty(blink_2))
    blink_2=wavePattern
    t2=clock
    time = time between t1 and t2
    t1=clock
else
    blink_1=blink_2;
    t2=clock;
    blink_time_1=etime(t2,t1);
    t1=clock;
    NetII=(blink_1 blink_2 time)
    
```

12. PERSONAL AND UNIVERSAL NEURAL NETWORK

In order to enhance the accuracy, user can build their personal neural networking by doing the training process. In the training process, the blinking patterns of user will be captured and save as a training data set.

Three stages of training are required to build a personal neural network. The LED board is set up on the Arduino robot to guide the user to have the training.

In the first stage, the LED will on for 20-30 seconds. Users are required to avoid blinking in this period. The aim of this stage is to collect the noise. At the next stage, users are required to blink only when the LED is on. The LED will light up and down continuously with 1 second separation. Which means the user need to blink once within 1 second when the LED is on and keep opening their eyes for 1 second when the LED is off. The aim of this stage is to collect the blinking pattern of the user’s natural blinking and also to capture some noise. In the final stage, the left LED and right LED will light up one by one with 1 second separation. Users are required to blink twice rapidly when the left LED is on and blink twice in a bit slower speed when the right LED is on. The aim of this stage is to collect the blinking pattern of different controlling command.

A universal neural network was also provided for the user so that they can control the robot without any training procedure. Compare to the personal neural network, the accuracy of the universal one is lower, so it is not recommended for a long-term usage.

The universal neural network is formed by other users. Which means, when a user trained a personal neural network, the data set in that personal neural network will be reused with other users’ personal neural network to form a universal one. Therefore, the more the user trains their

personal neural network, the universal neural network should be more accurate in terms of classifying human blinking and noise. Also, the universal neural network can be a reference for building a personal neural network as it already has a good classification of human blinking and noise.

13. AUTONOMOUS CONTROLLING FUNCTION

Because of the safety issue, 5 ultrasonic sensors and 3 infrared sensors are added in the robot. The ultrasonic sensors in the front part of the robot will detect the obstacles. Each sensor will echo an ultrasonic wave with a time delay to avoid the wave-conflict problem. The infrared sensors are installed at the bottom part of the robot, they are used to detect the distance between the robot body and the ground to prevent falling from the stair (see Figure 9).

By including autonomous obstacle avoidance, the motion can be modified and require less controlling command even the road has lots of obstacles. If the command from eye blinking is received, the robot will follow this command. Otherwise, maintain the motion in the last time step and run obstacle avoidance. In terms of performance, the car will become smoother in motion.

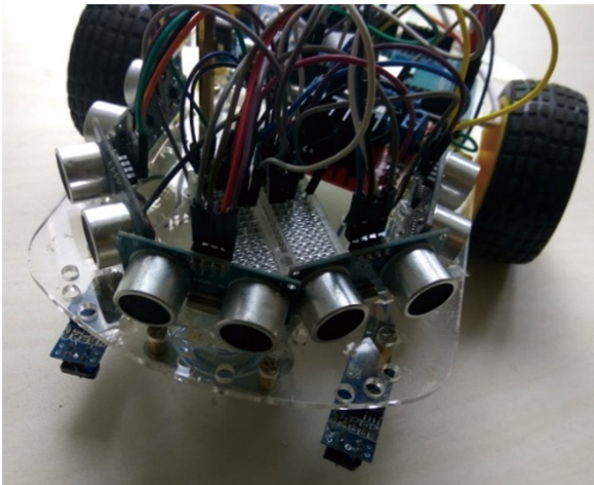


Figure 9. Ultrasonic sensors and infrared sensors

However, there are some limitations when adopting this method. First, it is impossible when user wants to get closer to the obstacle. Second, the robot will be totally out of control if one or more sensor has some unpredictable errors.

Therefore, this method has been modified to enhance the performance. If the robot is moving forward, it will include the autonomous obstacle avoidance as mentioned before. However, when the robot is turning left or right, the autonomous obstacle avoidance function will not be used.

By adopting the modified method, users are able to get closer to the obstacle as they may want. And users can control the robot by changing the direction even one or more sensors have an unexpected error. Therefore, both limitations are solved.

By including autonomous terrain detection, the motion can be modified and prevent the robot falling from the stair. If the command from eye blinking is received, the robot will follow this command. Otherwise, maintain the motion in the last time step and run terrain detection.

The same method will be used as mentioned above. By adopting the modified method, users are able to get closer to the stair as they may want. And users can control the robot by changing the direction even one or more sensors have an unexpected error.

14. TESTING

Accuracy and safety are the most important part of this project. In order to test the accuracy of the robot the following methods are used, and each method will be tested for three rounds.

During the test, the number of correct command means the motion of the robot match the command sent by the user; the number of the wrong command means either: 1) The motion does not match the command. 2) The command is seen as noise. 3) Noise is seen as a command.

The first test is focused on the autonomous terrain detection (see Figure 10). During the testing, the robot should detect the terrain and avoid falling from the stair. The testing criteria are to count the times of falling from the stair. In this test, a higher-level ground will be used to simulate the stair.



Figure 10. Autonomous terrain detection

The testing results are listed in Table 1.

Table 1. Testing results of autonomous terrain detection

Round	Number of times trying to fall from the high-level ground	Results	Accuracy
1	2	Both avoided	100%
2	2	Both avoided	100%
3	2	Both avoided	100%

During the test, the robot can avoid falling from stair successfully. When the sensors detected the distance between the robot body and the ground is too large, it will go back and turn to avoid falling from stairs.

The next test is based on a simple rectangular map (see Figure 11). During the testing, the robot should follow the rectangle drawn on the floor. The testing criteria are to count the times of incorrect command received. In this test, 5 checkpoints are labeled on the ground. The robot should reach each checkpoint in the order of: red, orange, yellow, green, and finally stop at blue.



Figure 11. Test on a simple map

The testing results are listed in Table 2.

Table 2. Testing results of running on a simple map

Round	The number of commands sent	Correct	Wrong	Accuracy
1	20	18	2	90%
2	19	18	1	94.7%
3	15	15	0	100%

During the test, the robot can pass through all the checkpoints successfully. Even sometime the robot may not recognize the user’s command correctly, the user can send the command again to avoid the robot being derailed or directly stop the robot by blinking three times and more rapid. The stop command is the most sensitive and accurate command so that it can use to prevent the risk happen.



Figure 12. Test on an irregular map

The next test is based on an irregular map (see Figure 12). During the testing, the robot should follow the irregular path on the floor. The testing criteria is to count the times of incorrect command received. In this test, 5 checkpoints are labeled on the ground. The robot should reach each checkpoint in the order of: red, orange, yellow, green, and finally stop at blue.

The testing results are listed in Table 3.

Table 3. Testing results of running on an irregular map

Round	The number of commands sent	Correct	Wrong	Accuracy
1	22	20	2	90.9%
2	26	23	3	88.4%
3	20	18	2	90%

Same with the previous test, the robot can pass through all the checkpoints successfully. The accuracy is around 90%.

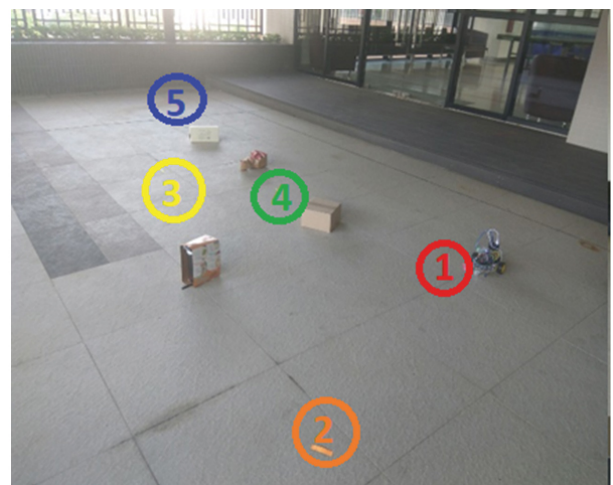


Figure 13. Static obstacle avoidance

The next test is focused on the static obstacle avoidance function (see Figure 13). During the testing, the robot should reach all checkpoints and avoid the obstacle automatically, even the user is not sending any command. The testing criteria is to count the times of incorrect command received. And counting the times of hitting the obstacle. In this test, 5 checkpoints are labeled on the ground. The robot should reach each checkpoint in the order of: red, orange, yellow, green, and finally stop at blue. Several static obstacles are placed on the path, the robot should avoid it automatically.

The testing results are listed in Table 4.

Table 4. Testing results of static obstacle avoidance

Round	The number of commands sent	Correct	Wrong	Accuracy	Obstacle hit
1	24	20	4	83.3%	0
2	25	22	3	88%	0
3	26	21	5	80.7%	0

Although the accuracy becomes lower in this test, the average accuracy can keep in around 85%, which is acceptable. One possible reason of the accuracy drop is that we need to find a good path to hit the obstacle, so we can test the obstacle avoidance function during the test. This abnormal motion may cause some confusion in terms of robot control.

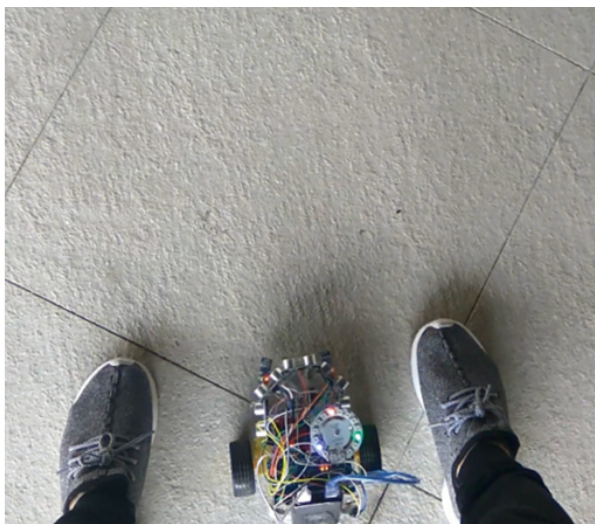


Figure 14. Dynamic obstacle avoidance

The final testing is focused on the dynamic obstacle avoidance function (see Figure 14). During the test, the robot should avoid the dynamic obstacle automatically, even the user is not sending any command. The testing criteria is to count the times of hitting the obstacle.

The testing results are listed in Table 5.

Table 5. Testing results of dynamic obstacle avoidance

Round	Number of times of placing the obstacles	Impacted	Avoided	Accuracy
1	22	1	16	94.1%

During the test, the robot can avoid almost all dynamic obstacles successfully. The robot only impacted when the speed of the dynamic obstacle is faster than the speed of the robot.

15. EVALUATION

In terms of mobility, this project provided a method that no any map or path need to be predefined. Although some other systems provided a simple controlling method, the usage of the wheelchair is limited to a specific environment because the system require the predefined paths.^[17] If the environment changed, a new map is required to load into the system. In this project, the robot can move anywhere like a real wheelchair. The user can blink three times or more to stop or start the robot. And blink twice to start turning or stop turning. This is a simple controlling method that allows robots to move in any direction and not to rely on any predefined path. However, it is important to ensure the Bluetooth connection between different devices must be established and the strength should be stable.

In terms of accuracy, this project provided a robot controlling method with around 85% accuracy on average, which is an acceptable performance. During the test, we found that the accuracy is relatively high in level 1 and level 2. The accuracy can reach 100%. However, start from level 4, the accuracy becomes lower. One possible reason behind is that we need to find a good path to hit the obstacle so that we can test the obstacle avoidance function. This abnormal motion may cause some confusion to the robot controller. And this situation may not happen in the real life as we will not want to hit the obstacle by using the wheelchair. Although the accuracy is getting lower from level 4, this method still provided 85% accuracy on average.

In terms of safety, this project provided an immediate command to stop the robot to avoid the risk. Also, obstacle avoidance and autonomous terrain detection are included to enhance the safety. Some other similar systems require users to select a command to stop while the command selection time may take up to 7 second. Therefore, compare to other similar system, this project has a better performance in terms of safety as it provided an immediate command to stop the robot.^[9, 17, 18]

In terms of cost-effective, all the controlling command of the robot are in real time. The user does not need to wait be-

fore sending any command. Also, all the commands are just simple blinking which can be sent by the user immediately. Some other similar systems require users to spend a long time to select a controlling command. And the commands are difficult to perform, for example, performing the motor imagery, keeping in a high attention and performing stress blinking.^[17-19] This project provided a set of simple and effective command for the user to control the robot. Blinking three times or more means start or stop the robot, blinking twice means start or stop turning. These two simple commands can perform rapidly, so that the user can control the robot in real time. Therefore, compare to other similar system, the method used in this project is more cost-effective as it provided an immediate command to stop the robot.

In terms of simplicity, this project provided a natural controlling method that does not require unnatural blinking (e.g. left blinking, right blinking, strong blinking, long blinking). The user can blink twice to turn and blink three times or more to stop. Also, as autonomous obstacle avoidance and autonomous terrain detection are included, the user can send less command to avoid the jerky blinking. Some other similar systems require users to avoid the obstacles manually. Considering the daily-life situation, there must be some static and dynamic obstacle in the street, the users may need to send a lot of command if they are required to avoid the obstacles manually. Therefore, the controlling method implemented in this project has included the autonomous obstacle avoidance and autonomous terrain detection function to reduce the frequency of sending the command.

Overall, all the objectives of this project are met, which is a huge success in the development of mind-wave controlled robot.

16. CONCLUSION

This project has provided a new method for paralyzed patients to control the wheelchair using a low-cost brainwave-reading headset. This project has implemented an Arduino robot to simulate a brainwave-controlled wheelchair for paralyzed patients with an improved controlling method. The robot can move freely in anywhere under the control of the

user and it is not required to predefine any map or path. An accurate and natural controlling method is provided, and the user can stop the robot any time immediately to avoid risks or danger.

This project used the eye blinking as the robot controlling method as the eye blinking will cause a significant pulse in the EEG signal. By using the neural network to classify the blinking signal and the noise, user can send the command to control the robot by blinking twice in a short period of time.

Autonomous obstacle avoidance and autonomous terrain detection are used to reduce the frequency of sending the controlling commands, and avoid the risks and danger immediately. This project has been evaluated by driving the robot in different places to test whether it can follow the expected path, avoid the obstacles, and stop on a specific position. The accuracy is around 85%, which is acceptable, and the robot can arrive all the checkpoint, avoid all the obstacles and stop at a specific point accurately.

17. FUTURE WORK

The future work will be implementing this method to a real wheelchair and take experiments with some paralysis patients like ALS. It is important to ensure that the patients can also be able to use our mind wave-controlled wheelchair system satisfactorily. As mentioned by a survey,^[27] the wheelchair functions for the ALS patients is different to the normal motorized wheelchair. When implementing this method to a real wheelchair for the ALS patients, collecting the user experience will be a very important step to make the wheelchair success.

Moreover, in terms of scalability, current system can be further modified to be more scalable and add more module to it. For example, it can be further extended to a wheelchair with an arm^[28] or combining other BCI application to this system.^[29]

To achieve higher mobility and lower cost, instead of a laptop, a powerful single-board computer like raspberry pi can be used to implement and train the neural network.^[30]

REFERENCES

- [1] Puviarasi R, Ramalingam M, Chinnavan E. Self assistive technology for disabled people – voice controlled wheel chair and home automation system. IAES International Journal of Robotics and Automation (IJRA). 2014; 3(1).
- [2] Tan D, Nijholt A. Brain-Computer Interfaces and Human-Computer Interaction. Brain-Computer Interfaces Human-Computer Interaction Series. 2010: 3-19.
- [3] Siswoyo A, Arief Z, Sulistijono IA. Application of Artificial Neural Networks in Modeling Direction Wheelchairs Using Neurosky Mindset Mobile (EEG) Device. EMITTER International Journal of Engineering Technology. 2017; 5(1).
- [4] Susan K. Mind over Matter: a \$199 Headset Controls Objects via Brain Waves. (NeuroSky MindSet). IEEE Spectrum. 2010; 47(5): 23.

- [5] McFarland DJ, Wolpaw JR. Brain-computer interfaces for communication and control. *Communications of the ACM*. 2011; 54(5): 60-66. PMID:21984822. <https://doi.org/10.1145/1941487.1941506>
- [6] He B. *Neural Engineering*. Boston (MA): Kluwer Academic/Plenum Publishers, 2005.
- [7] Girase P, Deshmukh M. Mindwave Device Wheelchair Control. *International Journal of Science and Research (IJSR)*. 2016; 5(6): 2172-2176. <https://doi.org/10.21275/v5i6.NOV164722>
- [8] Stamps K, Hamam Y. Towards Inexpensive BCI Control for Wheelchair Navigation in the Enabled Environment – A Hardware Survey. *Brain Informatics Lecture Notes in Computer Science*. 2010: 336-345.
- [9] Arzak M, Sunarya U, Hadiyoso S. Design and Implementation of Wheelchair Controller Based Electroencephalogram Signal using Microcontroller. *International Journal of Electrical and Computer Engineering (IJECE)*. 2016; 6(6): 2878. <https://doi.org/10.11591/ijece.v6i6.11452>
- [10] Ghorpade S, Patil A. Mindwave-A New Way to Detect an Eye Blink. *IJARCCCE*. 2015: 82-84.
- [11] Yasui Y. A Brainwave Signal Measurement and Data Processing Technique for Daily Life Applications. *Journal of Physiological Anthropology*. 2009; 28(3): 145-150. PMID:19483376. <https://doi.org/10.2114/jpa2.28.145>
- [12] Tjust A. *Extraocular Muscles in Amyotrophic Lateral Sclerosis*. PhD dissertation, Umeå; 2017.
- [13] Udayashankar A, Kowshik A, Chandramouli S, et al. Assistance for the Paralyzed Using Eye Blink Detection. 2012 Fourth International Conference on Digital Home. 2012: 104-108.
- [14] Ayudhya C. A Method for Real-Time Eye Blink Detection and Its Application. *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR)*; 2017.
- [15] Aramideh M, Eekhof J, Bour L, et al. Electromyography and recovery of the blink reflex in involuntary eyelid closure: a comparative study. *Journal of Neurology, Neurosurgery & Psychiatry*. 1995; 59(6): 662. <https://doi.org/10.1136/jnnp.59.6.662-b>
- [16] NeuroSky. *Exercise Equipment for Your Mind*. Exercise Equipment for Your Mind.
- [17] Rebsamen B, Burdet E, Guan C, et al. Controlling a wheelchair using a BCI with low information transfer rate. 2007 IEEE 10th International Conference on Rehabilitation Robotics. 2007: 1003-1008.
- [18] Gandhi V, Prasad G, Coyle D, et al. EEG-Based Mobile Robot Control Through an Adaptive Brain-Robot Interface. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. 2014; 44(9): 1278-1285. <https://doi.org/10.1109/TSMC.2014.2313317>
- [19] Stephygraph LR, Arunkumar N, Venkatraman V. Wireless mobile robot control through human machine interface using brain signals. 2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM); 2015.
- [20] "Bluetooth Module HC-05", 2017. [Internet]. Available from: https://wiki.eprolabs.com/index.php?title=Bluetooth_Module_HC-05 [Accessed: 19-Nov-2017].
- [21] L298 Dual H-Bridge Motor Driver. 2017. [Internet]. Available from: http://www.robotshop.com/media/files/pdf/data_sheet-mot103b1m.pdf [Accessed: 19-Nov-2017].
- [22] Nguyen HT, Trung N, Toi V, et al. An autoregressive neural network for recognition of eye commands in an EEG-controlled wheelchair. 2013 International Conference on Advanced Technologies for Communications (ATC 2013), Ho Chi Minh City. 2013. p. 333-338.
- [23] Ning B, Li MJ, Liu T, et al. Human Brain Control of Electric Wheelchair with Eye-Blink Electrooculogram Signal. *Intelligent Robotics and Applications Lecture Notes in Computer Science*. 2012: 579-5882.
- [24] Levenberg-Marquardt (trainlm): Backpropagation (Neural Network Toolbox). [Internet]. Available from: <http://matlab.izmiran.ru/help/toolbox/nnet/backpr12.html> [Accessed: 05-Jan-2018].
- [25] Yu H, Wilamowski BM. *LevenbergMarquardt Training Industrial Electronics Handbook*, vol. 5 *Intelligent Systems*, 2nd ed., chapter 12, pp. 12-1 to 12-15, CRC Press 2011.
- [26] Documentation, MATLAB & Simulink. [Internet]. Available from: <https://www.mathworks.com/help/nnet/ref/trainlm.html> [Accessed: 10-Jan-2018]
- [27] Trail M, Nelson N, Van JN, et al. Wheelchair use by patients with amyotrophic lateral sclerosis: A survey of user characteristics and selection preferences. *Archives of Physical Medicine and Rehabilitation*. 2001; 82(1): 98-102. PMID:11239293. <https://doi.org/10.1053/apmr.2001.18062>
- [28] Bousseta R, Ouakouak IE, Gharbi M, et al. EEG Based Brain Computer Interface for Controlling a Robot Arm Movement Through Thought. *Irbm*. 2018; 39(2): 129-135. <https://doi.org/10.1016/j.irbm.2018.02.001>
- [29] Sanjana M. *Brain Computer Interface and its Applications – A Review*. 2017 Jun.; 8(5).
- [30] Kucukyildiz G, Ocak H, Karakaya S, et al. Design and Implementation of a Multi Sensor Based Brain Computer Interface for a Robotic Wheelchair. *Journal of Intelligent & Robotic Systems*. 2017; 87(2): 247-263. <https://doi.org/10.1007/s10846-017-0477-x>