Brain-Controlled Wheeled Chair Path Planning for Indoor Environments

Haider Abdulkarim [1], Mohammed Z. Al-Faiz [2]

[1] *Electrical and Communication Engineering Dept.,* Uruk University, Baghdad, Iraq.
Email: haiderasadi@daad-alumni.de

[2] Al Nahrain University, Iraq.
Mzalfaiz12@gmail.com

Abstract **- There has been intensive research in the field of ElectroEncephaloGram (EEG) and its applications to enhance the quality of life for fully and partially disabled patients. In recent years, the focus has been driven to fully control a wheeled chair using Brain Computer Interface (BCI). Many works have been introduced in the pursuit of this goal. This paper proposes a Brain Computer Interface for fully automated indoor wheelchair navigation and path planning system. The proposed BCI system uses Convolutional Neural Network for feature extraction and shallow neural network for classification. The experimental results are compared to simulated MATLAB model and showed an average error of 3 cm.**

Keywords - EEG, BCI, CNN, Wheeled Chair, Indoor Navigation.

I. INTRODUCTION

Mobile robots have witnessed a lot of interest and research in the last decades. They have been used in wide range of applications, starting from educational robots, up to far-planet excavations, such as curiosity rover. In spite of their wide variety of types, all mobile robots share common principles: they need to know where they are (navigation) and the route to their destination (path planning). The navigation phase provides the robot with its location as a point in the two-dimensional space (*xy*). If the navigation takes place in an outside area, Global Positioning System (GPS) receiver is used, which utilizes the triangulation principle to locate the object and calculate its three coordinates: longitude, latitude as well as elevation. However, in indoor environments, the positioning requires centimeter accuracy. This accuracy cannot be provided using the GPS receivers because received GPS satellite signal suffers from high attenuation inside buildings. Several indoor navigation solutions implement a positioning technology that uses ultrasonic beacons. The beacons continuously exchange location information with other in-sight beacons as well as with the mobile robot and hence they can calculate the instantaneous location of a mobile robot instantaneously. Two mostly-used indoor positioning systems exist: POSYXZ and Marvelmind Robotics.

Recently, Brain Computer Interface (BCI) field has increasing number of researches and applications. BCI uses extracted electroencephalogram (EEG) signals from brain to perform specific tasks, based on the intention of the person's mind. This field has promising applications for people who are fully disabled or amputated limbs.

Many studies have been done in the field of mobile-robot driving using BCI. For instance, K. Choi and A. Cichocki [1] implemented a Brain-Machine Interface (BMI) to capture EEG signals. They used Common Spatial Pattern (CSP) for feature extraction and Support Vector Machine for classification. They also used EMG for emergency stopping. They were able to move the wheelchair in four directions without colliding with obstacles. R. Leeb et al. [2] demonstrated the use of EEG to control a wheelchair in virtual reality using beta frequency band. The extracted feature was logarithmic band power and a threshold was used to decide between moving and stopping state. D. Craig and H. Nguyen [3] proposed a real-time BCI wheelchair controller for patients with Spinal Cord Injury (SCI). The BCI system was able to classify three different mental tasks to support the head-movement commands, using neural network classifier. They reached a classification accuracy of 82%. Y. Chae et al. [4] proposed a BCI system to control a simulated robot to navigate through a maze. They used Linear Discriminant Analysis (LDA) as feature selector and both Intentional Activity Classifier and the Motor Direction Classifier for direction classification, and they hit a best accuracy of 87.3%. K. Choi [5] implemented a wheelchair BCI system capable of avoiding obstacles using combined SVM, Principle Component Analysis (PCA) and Singular Value Decomposition (SVD). JR. Millán et al [6] used a statistical classifier that characterize each class with its Probability Density Function (pdf). The training phase lasted for 3-5 days, with three mental classes. Th robot did navigate through rooms, but with limitations related to turning angles and non-use of motor controller to accommodate for navigation errors. Y. Yu et al [7] proposed a BCI system that controls a car system by starting and stopping engine and direction control. They used binary LDA classifier with best accuracy of 90.3%. However, the proposed system was not tested in real-time driving conditions. K. Tanaka et al. [8] studied the control of an electric wheelchair moving direction using recursive training algorithm, while navigating through a room. The workspace is divided into 60cm×90cm regions and the subject had to choose his next target goal, either left or right, mentally. They reached an accuracy of 80%. A. Barbosa et al. [9] implemented probabilistic neural network using Discrete Wavelet Transform (DWT) features in Delta band. Four types of movements were classified to control the movement of a robot in all directions, achieving hit rate of 91%. R. Ron-Angevin et al. [10] implemented a self-paced BCI that uses two MI movements to control virtual robot movements in four directions and to avoid collisions with virtual street lanes, using LDA classifier.

This paper proposes an indoor path planning algorithm that uses the captured EEG signal via EMOTIV EPOC headset to navigate an electric wheelchair inside a building. The proposed algorithm is able select one of four different locations based on the user's mental intension. The algorithm then auto-drives the wheelchair via an implemented controller, avoiding static as well as moving obstacles, arriving at the final goal without further intervention from the user.

II. BRAIN CONTROLLED WHEEL CHAIR

The proposed model includes two main phases:

1. EEG signal feature extraction and classification (training and classification).
2. Indoor navigation and path planning.

In the training phase, the subject sits relaxing on a chair, then a que is shown to him on the laptop screen to produce one of the mental state commands: right hand, left hand, right leg, left leg or relax (idle) state. The que is ON for only five seconds. The EMOTIV EPOC is used for recording the EEG signals and storing them as matrices of dimension 14 channels x 640 time samples, with a sampling rate of 128 samples per seconds. This training session is repeated for five times for each mental state, and the procedure is repeated for each of the other three mental states. The resulting dataset is stored as the training EEG dataset, with the corresponding label that defines the state. Those captured EEG signals are then fed to the next step: feature extraction, which is the Convolutional Neural Network (CNN) structure. The extracted features are then stored with the corresponding labels, that identifies each class.

In the classification phase, the above steps are repeated and the resulting feature vector (of the unknown-class) is fed to the shallow Neural Network and compared to each training feature vector via a binary classification tree algorithm. The NN output is the winning class label, which is related to one of four house location that the user needs to navigate. In the second phase, the indoor house plan is plotted in advance using EasyPlanPro or any software. In addition, the Marvelmind robotics starter kit beacons (Fig. 1) are placed on a high position on the wall, each one is in Line-of-Sight with other beacons in order to exchange measured power messages. The navigation system consists of four wall beacons, one modem and one mobile beacon. The modem is connected to a PC and receives all of the beacons-generated messages.

Figure 1: Marvelmind robotics beacon.

By analyzing the Received Signal Strength Indicator (RSSI) level on all beacons, the indoor map can be formed, and the location of each of the anchor beacons, as well as the *xyz* location of the mobile beacon can be located. Many parameters of the indoor positioning system can be adjusted, such as the transmit and receive power level, position update rate, the channel selection of each of the transmit and receive bands and so on. The electric wheelchair is shown in Fig. 2. It uses two DC drive motors in the rear and two smaller castor wheels in the front. The normal control of the wheelchair is achieved using a joystick moving in all directions. This joystick is connected to a controller that generates Pulse Width Modulation (PWM) output. This output is fed to each motor. According to the direction of the joystick and the tensity of pressure, each DC motor is rotated with a given speed. For example, if the user wants to turn right, the left motor rotates forward while the right motor rotates backwards. This model is known as differential drive.

To achieve a wheelchair robot fully controlled from within the proposed BCI system, the wheelchair built-in controller is replaced with our proposed controller. It consists of an Arduino Uno microcontroller board, which is based on the ATmega328P, connected to several modules. The controller schematic diagram is shown in Fig. 3. The BTS 7960 full H-bridge module receives its control signal from Arduino Uno. Then, it generates PWM output, with rated current of 5A and fixed voltage of 12V. Depending on the Forward and Reverse Drive Enable inputs, the motor can move forward or backward, at the speed level received from Arduino and adjusted by the width of the PWM output. In addition, the wheelchair is provided with three ultrasonic radars, one at the front and two at both side corners of the chair. The ultrasonic sensors, in addition to the collision avoidance procedure, provides a secure level against collision with obstacles. The side sensors are used to turn in the direction that has the furthest or no obstacles.

To achieve a wheelchair robot fully controlled from within the proposed BCI system, the wheelchair built-in controller is replaced with our proposed controller. It consists of an Arduino Uno microcontroller board, which is based on the ATmega328P, connected to several modules. The controller schematic diagram is shown in Fig. 3.

Figure 2. Mobile wheelchair with two differential-drive motors.

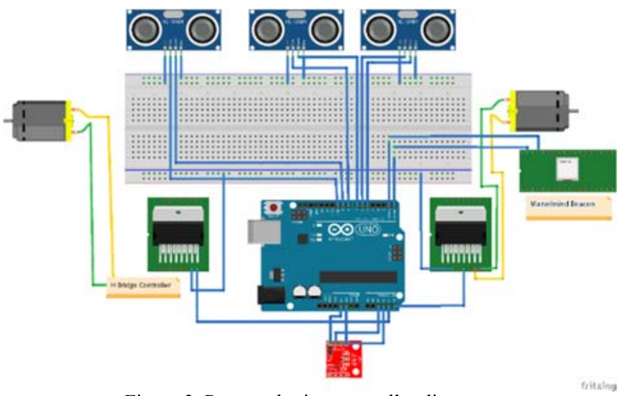

Figure 3. Proposed microcontroller diagram.

The BTS 7960 full H-bridge module receives its control signal from Arduino Uno. Then, it generates PWM output, with rated current of 5A and fixed voltage of 12V. Depending on the Forward and Reverse Drive Enable inputs, the motor can move forward or backward, at the speed level received from Arduino and adjusted by the width of the PWM output. In addition, the wheelchair is provided with three ultrasonic radars, one at the front and two at both side corners of the chair. The ultrasonic sensors, in addition to the collision avoidance procedure, provides a secure level against collision with obstacles. The side sensors are used to turn in the direction that has the furthest or no obstacles. The compass module MPU9265 3-axis magnetometer and gyroscope is used for measuring the instantaneous robot direction and calculating the calibration angle for correcting the wheelchair head alignment. The built-in gyroscope is used to calibrate the compass readings, as the magnetic fields from other indoor electric devices interferes with the compass reading and introduces nonlinear error.

III. NAVIGATION AND PATH PLANNING

The wheelchair robot controller has two targets: estimate its indoor location in the x-y coordinates system (navigation), and find a free route to a position inside the target room (path planning). The navigation information is retrieved from the mobile beacon units wirelessly via the modem connected to the PC. The modem periodically calculates the current position of the mobile beacon. The path planning, on the other hand, has advanced knowledge of all of the target positions. This knowledge is provided by making an indoor map of all the indoor space and marking fixed obstacles (like walls) as occupied objects in the occupancy map, that need to be excluded from the solution workspace. Upon selecting a mental state related to a specific position, the target coordinates are fed to the path planning algorithm, as well as the current position and reading of the compass. The heading of the wheelchair is estimated from compass reading, so that the controller is able to initially rotate the wheelchair in a direction of the bearing angle of the next target position. After this step, the Rapidly Exploring random Tree (RRT) algorithm is used to generate random paths to the target and them uses the prior knowledge of the indoor map to eliminate the paths that go through fixed obstacles, like walls, tables, coaches and others. Then, the shortest and only valid paths (free-of-obstacles) are selected. The controller then issues multiple forward commands, each for one meter length at an average speed of 0.27 m/s and consistently check for mobile obstacles, like chairs. If an obstacle is found, obstacle-avoidance strategy is committed, by retracting backward, rotate left or right by 90 degrees angle, depending on which direction is free, and moves forward. At this point, the algorithm recalculates the bearing angle and the operation repeated until reaching the target. To overcome instantaneous error readings in both the compass module and the Marvelmind positioning system, sensor readings are averaged. This minimizes the immediate and final target errors and reduces the time spent in aligning the wheelchair between intermediate points.

The navigation algorithm steps are as follows:

1. Calculate bearing angle:

$$\theta = \tan^{-1}\frac{x_2 - x_1}{y_2 - y_1} \quad (1)$$

If $\theta < 0$, then:

DOI 10.5013/IJSSST.a.22.01.10 ISSN: 1473-804x online, 1473-8031 print

$$\theta = \theta + 2\pi \tag{2}$$

2. Wheeled-chair heading alignment
3. Is Dist > thrshld ? if no, stop.
4. Calculate bearing angle (as in step 1)
5. Is Dist>0.5 total_Dist? If no, use fine align_error (5°) Else, use coarse align_error (10°)
6. Car_heading alignment (as in step 2)
7. Check obstacle: if yes, check obstacle direction {front, left or right}? Avoid obstacle (reverse drive, then rotate in a direction free-of obstacle)
8. Is target reached? If no, go to step 4 Else, stop.

IV. WHEELCHAIR KINEMATIC MODEL

To predict the future path of the robot and to maneuver around obstacles, robot kinematic modeling is needed. The robot is modeled as a rigid body on wheels, moving on a horizontal plane. The robot chassis moves on a two-dimensional plan, with coordinates $\{x, y\}$. Additional degrees of freedom such as the wheel axels, the joints of the steering wheels are neglected to simplify the model [11].

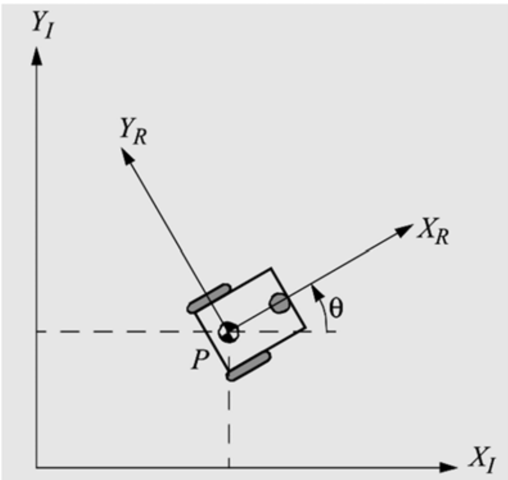

Figure 4. Global reference frame $\{X_I, Y_I\}$ and the robot local reference frame $\{X_R, Y_R\}$.

The relationship between the plane global reference frame and the robot local refence frame is shown in Fig. 4. The axes X_I-Y_I defines the global reference frame at origin *O*: $\{X_I, Y_I\}$. Point *P* on the robot chassis is chosen as the robot reference point. The basis axes $\{X_R, Y_R\}$ are relative to *P* on the robot chassis and they represents the local reference plane of the robot. The position of reference point *P* in the global frame is determined by *x*,*y* and the angle between global and local reference frames, *θ*. The robot position can be expressed as a vector as:

$$\xi_I = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \tag{3}$$

The subscript I represents the pose related to the global reference frame. The mapping is a function of the current position of the robot, and it is described by using orthogonal rotation matrix:

$$R(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

The matrix of Equation [4] is used to map the robot motion from the reference frame $\{X_I, Y_I\}$ to a motion term in the local reference frame $\{X_R, Y_I\}$. This mapping is expressed using the velocity vector $\dot{\xi}_I$:

$$\dot{\xi}_I = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} \tag{5}$$

Then, the relation between the motion in the two frames is formulated as:

$$\dot{\xi}_R = R(\theta)\dot{\xi}_I \tag{6}$$

The mapping formula described in Equation [6] **Error! Reference source not found.**answers the following question: given a robot geometry and speed of its wheels, what is the predicted robot path? The wheelchair kinematic model is shown in Fig. 5.

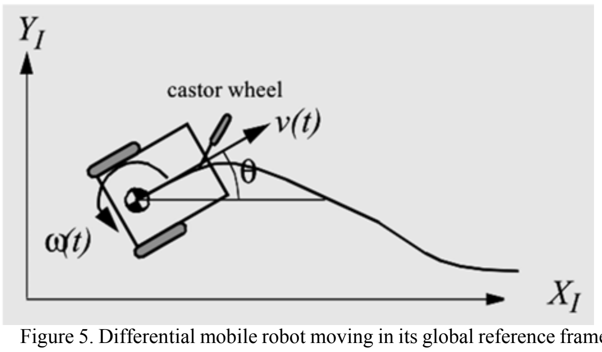

Figure 5. Differential mobile robot moving in its global reference frame.

This wheelchair has two wheels, each has diameter r. The reference point P is centered at distance l from each wheel. Therefore, given r, l, θ and the rotational speed of each wheel, $\dot{\phi}_1$ and $\dot{\phi}_2$, the forward kinematic model can estimate the robot's overall speed in the global reference frame as:

$$\dot{\xi}_I = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = f(l, r, \theta, \dot{\phi}_1, \dot{\phi}_2) \tag{7}$$

The final kinematic model for the differential-drive robot is given by:

$$\dot{\xi}_I = R(\theta)^{-1} \begin{bmatrix} \frac{r\dot{\phi}_1}{2} & \frac{r\dot{\phi}_2}{2} \\ 0 & \\ \frac{r\dot{\phi}_1}{2l} & \frac{-r\dot{\phi}_2}{2l} \end{bmatrix} \quad (8)$$

Where $R(\theta)^{-1}$ is calculated as:

$$R(\theta)^{-1} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (9)$$

V. RESULTS AND DISCUSSION

A. MATLAB Kinematic Model Results

MATLAB R2019b with Robotics System Toolbox is used to simulate the wheelchair robot movement within the indoor map. The house map is created by EasyPlanPro software, then it is exported as a bmp file, with large static obstacles (such as walls and sofas) are represented by shadow areas. This map file is then imported in MTALAB workspace and a binary occupancy map is created. The occupancy map simply a matrix of two dimensions, equal to the dimension of the original map. Each pixel in the map is marked as occupied (binary 1) or non-occupied (binary 0). the probabilistic roadmap (PRM) object is then defined to find a network graph of possible paths to a destination based on free and occupied spaces, as shown in Fig. 6. The mobileRobotPRM object randomly generates nodes and then creates connections between them if:

- no obstacles between each pair of nodes
- the connection length is less than or equal to the maximum specified connection distance.

After that, the shortest-free path from the current position to the destination is chosen and the mobile robot object simulates the instantaneous move with given speed and rotation angle based on the predefined kinematic parameters of the robot chassis (Fig 7). At the end of each moving step, the Euclidean distance between the current position and the target is calculated. If it is less than or equal to 10cm (required accuracy), the algorithm stops, otherwise the object is moved to the next node and the threshold is checked again.

The average accuracy for MATLAB model is calculated as follows: the model is run within a loop. At the start, the robot is positioned at location 1. A small laptop is fixed in front of the user and a cue is shown to start the EEG mental state selection. The EEG is recorded for five seconds, and the BCI classifier outputs one of the target positions: 2,3 or 4. After goal is reached, the error is calculated based on Euclidean distance. To evaluate the performance, the user issues navigation mental commands to all targets within one loop. Next, the loop is repeated again and the repetition is done for 10 times. All of the wheeled chair parameters are taken into account, such as the speed and radius of the wheel, which governs the minimum turning radius. The average accuracy of the simulated model is 6.74 cm.

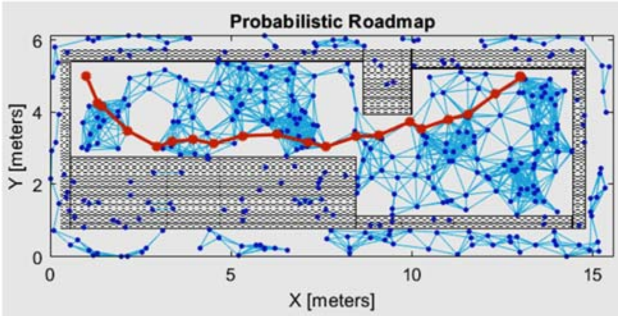

Figure 6. Generated probabilistic roadmap to reach target point, with nodes lying on obstacles are omitted from the solution space.

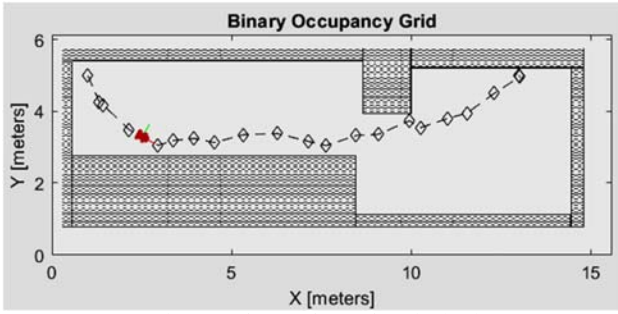

Figure 7. Shortest-free path to destination.

B. Practical Wheeled chair Robot Results

In the actual experiment, the EEG signal capturing and classification is identical to the simulated model. However, the positions of the targets in the map are stored in the Arduino code in advance. In addition, a specific driving strategy is proposed to overcome hardware limitations. Since the readings from obstacle sensors, compass and indoor positioning beacons cannot be checked continuously, the proposed strategy is to divide the calculated distance between two target positions to two-meter sections, move the robot forward for 0.5 second and then check for new readings. Different strategy comparisons are shown in Table I. When arriving near the end of each of the two-meter section, the movement is done in a 0.15 second-step. Several strategies are run and compared, in terms of the reaching-target error. The worst scenario occurs when moving 80% of the two-meter section with 0.5 sec step, giving an average error of 22 cm. The best accuracy is reached when moving 50% of the section (one meter) with 0.5 second step and then 0.15 second for the other meter, reaching an accuracy of 3 cm. Furthermore, all the scenarios are repeated with obstacles are put in random positions. In all cases, the obstacle-avoidance succeeded to avoid all of them with 0% collision rate.

TABLE I. DIFFERENT SCENARIOS ERROR.

Scenario	Duration	Error (cm)	notes
Dist ≥ 0.2 *Dist < 0.2*	0.5 sec 0.15 sec	**22**	**worst**
Dist ≥ 0.3 *Dist < 0.3*	0.5 sec 0.15 sec	21	
Dist ≥ 0.4 *Dist < 0.4*	0.5 sec 0.15 sec	17	
Dist ≥ 0.5 *Dist < 0.5*	0.5 sec 0.15 sec	**3**	**best**

VI. CONCLUSIONS AND FUURE WORK

A novel BCI for automated-navigation wheeled chair users has been proposed. The proposed system is a user-friendly because it requires short session time for training. In addition, due to the use of indoor navigation and the automated obstacle avoidance mechanism, the disabled user does not need to continuously issue navigation commands. He would rather only issue the target position for once and the combined path planning-navigation algorithm will drive him in an obstacle-free route. This is a major advantage, compared to other proposed BCI system that require continuous navigation commands. In addition, the centimeter- accuracy makes the system very usable in highly crowded indoor areas.

The major limitation of the proposed system is that the EPOC headset needs to be installed by a proficient each time the disabled user wants to use the wheeled chair. It the future, simpler headset with no installation requirement would make the proposed BCI more suitable to use.

REFERENCES

[1] K. Choi and A. Cichocki, “Control of a wheelchair by motor imagery in real time,” vol. 5326 LNCS, pp. 330, 2008, doi: 10.1007/978-3-540-88906-9-42.

[2] R. Leeb, D. Friedman, G. R. Müller-Putz, R. Scherer, M. Slater, and G. Pfurtscheller, “Self-paced (asynchronous) BCI control of a wheelchair in virtual environments: A case study with a tetraplegic,” vol. 2007, 2007, doi: 10.1155/2007/79642.

[3] D. A. Craig and H. T. Nguyen, “Adaptive EEG thought pattern classifier for advanced wheelchair control,” pp. 2544, 2007, doi: 10.1109/IEMBS.2007.4352847.

[4] Y. Chae, S. Jo, and J. Jeong, “Brain-actuated humanoid robot navigation control using asynchronous Brain-Computer Interface,” pp. 519, 2011, doi: 10.1109/NER.2011.5910600.

[5] K. Choi, “Control of a vehicle with EEG signals in real-time and system evaluation,” vol. 112, pp. 755, 2012, doi: 10.1007/s00421-011-2029-6.

[6] J. D. R. Millán, F. Renkens, J. Mouriño, and W. Gerstner, “Noninvasive brain-actuated control of a mobile robot by human EEG,” vol. 51, pp. 1026, 2004, doi: 10.1109/TBME.2004.827086.

[7] Y. Yu et al., “Toward brain-actuated car applications: Self-paced control with a motor imagery-based brain-computer interface,” vol. 77, pp. 148, 2016, doi: 10.1016/j.compbiomed.2016.08.010.

[8] K. Tanaka, K. Matsunaga, and H. O. Wang, “Electroencephalogram-based control of an electric wheelchair,” vol. 21, pp. 762, 2005, doi: 10.1109/TRO.2004.842350.

[9] A. O. G. Barbosa, D. R. Achanccaray, and M. A. Meggiolaro, “Activation of a mobile robot through a brain computer interface,” pp. 4815, 2010, doi: 10.1109/ROBOT.2010.5509150.

[10] R. Ron-Angevin, F. Velasco-Alvarez, S. Sancha-Ros, and L. Da Silva-Sauer, “A two-class self-paced BCI to control a robot in four directions,” pp. 0, 2011, doi: 10.1109/ICORR.2011.5975486.

[11] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, Introduction to Autonomous Mobile Robots, Second Edition. 2011.