Indoor Navigation for Assistive Robots using EEG Signals as Feedback

Sunny Arokia Swamy Bellary*, David Grabowsky† and James M. Conrad‡
Department of Electrical and Computer Engineering, University of North Carolina at Charlotte
Charlotte, North Carolina 28223
Email: *sbellary@uncc.edu, †dgrabows@uncc.edu, ‡jmconrad@uncc.edu

***Abstract*—Assistive robots are used by various individuals with medical disabilities to help with tasks such as movement. A subset of these individuals are patients with the locked-in syndrome; these patients cannot communicate with a robot through traditional means, such as with a joystick. This work designs a navigation scheme which allows for an assistive robot to be controlled by patients suffering locked-in syndrome, thus allowing the patient to move about their environment. Navigation is accomplished using an algorithm that combines autonomous robot movement and communicated commands from the patient. To bridge the communication gap between the patient and robot, naturally occurring error-related potentials are used. These ERPs can be used to establish communication between the patient and robot without relying on the patient interacting with physical stimuli, such as a keyboard or joystick. The commands communicated to the robot comes in the form of a binary: correct or incorrect command in response to the movements of the robot at an intersection in a structured building. While more complicated commands can be classified from event-realted potentials (ERPs), such as directional movement, this simple command allows for fast reliable classifications and responses. To make up for the lack of complexity from patient commands, the robot is leveraged to handle tasks such as wall avoidance, while a navigation algorithm is designed to minimize the inputs required by the user when taking a commonly traveled path. The benefits of using a semi-controlled robot for navigation vs a fully autonomous robot is compared in terms of the time taken to discover and navigate an initial path to a destination. This work serves as a proof of concept for the proposed semi-autonomous navigation scheme to validate future work into the proposed design.**

***Index Terms*—assistive robots, error-related potentials, brain-computer interface, robot navigation, convolution nerual network**

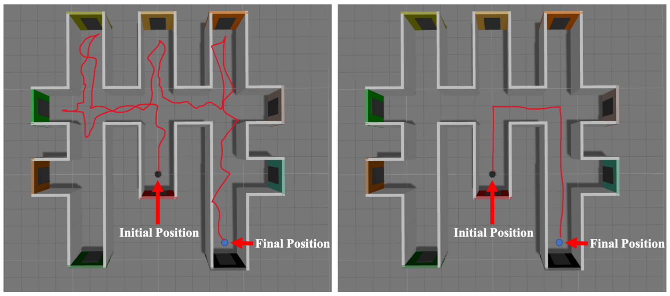
I. Introduction

Assistive robots pertain to a robot that helps individuals with disabilities by providing assistance with environmental interactions. The primary purpose of such robots is to enable individuals with disabilities to perform at a level relatively comparable to individuals without disabilities. Tasks such as: being able to navigate freely, pick and place objects from floors or shelves, eating/drinking, and personal hygiene [1] are examples of functions that robots can assist with. For most of these tasks, communication plays a vital role in understanding the requirements or needs of a disabled individual. Human communication is defined as the exchange of information verbally or non-verbally and can include utilizing keyboards, speaking commands, and more. Researchers have built systems that can be controlled using a variety of different communication methods such as joysticks, eye movements, head movements [2], [3]. Unfortunately, portions of the population can lose the capability to communicate via unfortunate circumstances such as accidents or birth disorders. Examples of conditions that can result in severe motor paralysis are strokes, severe cerebral palsy, motor neuron disease, amyotrophic lateral sclerosis, and encephalitis [4]. Many areas of research are actively focusing on ways to improve the life of these impaired. One particularly challenging condition is seen with patients who have neuromuscular disorders, either inherited or acquired by other factors. These patients have high needs and no practical way to communicate those necessities. These patients are often referred to as having locked-in syndrome [5] and have a complete loss of control over their voluntary muscles. Essentially, these patients are unable to speak and move but are conscious and can think and reason [6].

With the widespread and popular integration of robotics into daily life, it is no surprise to see robots now assisting disabled individuals. However, those who suffer from locked-in syndrome or patients with neuromuscular disorders still struggle to communicate with assistive robots, limiting the usefulness of these robots. This research focuses on patients who are affected by locked-in syndrome and seeks to help them achieve greater independence by enabling communication from the patient to assistive robots without relying on physical stimuli. This is accomplished through technologies such as Brain-Computer Interface (BCI)/Human-Robot Interaction (HRI) which provide an alternative communication bridge between a human brain and a robot.

BCIs enable persons to communicate using their human brain signal. There are various signal acquisition techniques among which Electroencephalography (EEG) signals is preferred or well known for its advantages which include: non-invasive, cost-effective and high temporal resolution [7]. The use of EEG signals for controlling robotics has seen many implementations, dating back to pioneering works such with Bozinovski et al. [8] where a methodology was implemented for controlling a robot with EEG Alpha Rhythm signals. EEG signals have many derivatives; one such is event-related potentials (ERP) signals. These signals are generated by a population of neurons in response to a perceptual, cognitive or motor event, in opposition to spontaneous activity that reflects the brain activity related to volunteer self-paced tasks. These

978-1-7281-6861-6/20/$31.00 ©2020 IEEE



(a) (b)

Fig. 1. Visualization comparing general types of navigation control (a) Fully autonomous method that simply wanders around the environment until a destination is reached (in this case a purple door); (b) Semi-controlled method that moves towards the center of any intersection detected then waits for user input to direct it towards an exit from the intersection.

events/stimuli can be either visual, auditive or vibrotactile [9]. ERP waveforms generated are categorized according to latency and amplitude after the presentation of stimuli, such as error-related potentials [10].

Utilizing these signals it becomes possible to control an assistive robot. Controlling an assistive robot could be subdivided into three broad categories:Completely controlled, Shared/Semi-controlled, and total autonomy. Among various tasks, we consider indoor navigation for locked-in syndrome patients. In this case, complete control isn't possible as these systems are either controlled using a joystick or keyboard. For these patients, only total autonomy is possible. The problem with enabling total autonomy for an assistive robot is that given a start point the robot must randomly explore each and every region in the hopes of stumbling upon the correct destination, as shown in Fig. 1(a). Hence we use a BCI system with a mobile robot to allow a patient to semi-control the navigation of the robot via EEG signals. Using the mobile robot and an hallway intersection detection algorithm we develop a navigation scheme that only requires a patient to input decisions at hallway intersections while relying on the mobile robot to handle local obstacle avoidance and movement, an example of such movement can be seen in Fig. 1(b). The overall focus of this work is the task of navigating an patient sitting on a mobile robot from a starting location to an ending destination inside a structured building while requiring minimum input from the patient.

In this paper our key contributions are:

- Novel use of error related potentials to control a robot for indoor navigation.
- User-based preference for the navigation.
- Navigation for locked-in patients without human aid.

II. Background and Related works

A. Error related potential

ErrPs are characterized by an initial positive peak occurring at about 200ms after an event followed by a massive negative deflection at about 200-250ms and a second positive peak at about 320ms [11]. These waveforms are reported to have a similar pattern in tasks using different modalities [12] [13] [14] and were almost identical when tested after several months [15]. Hence, the methods used to classify the signals can be very well generalized.

Studies conducted in past years intended to classify the ErrP signals using various classification algorithms. Errikos et. al [16] proposed a system which could extract features based on statistical measurements from averaged ERP recordings. Classification of the signals was accomplish using k-nearest neighbors (kNN) and support vector machines (SVM). Similarly, Ricardo and Millan [17] used a Gaussian Classifier by assigning the same prior probability for both correct and error classes. The most commonly used classifiers were the Gaussian Classifier, logistic regression, SVMs, and linear discriminant analysis and its variations. However, it should be noted that these works utilized varying prepossessing techniques, thus while the classification performances are noted between 70 and 80 %, the difference in prepossessing could be partially responsible for discrepancies in feature selection and performance metrics.

Deep learning based architectures have also seen their use in classifying EEG signals. Hubert and Axel [18] first used ConvNets, a deep learning based network, for detection of P300 in oddball paradigms. Later, Robin et al. [19] used Convolutional Neural Networks for decoding EEG signals which could classify left, right, feet and rest movements. Similar work was performed by Siavash et al. by using the parallel Convolutional neural network and energy based features [20]. However, not much work has been done towards using deep learning approaches for error related potential classification.

B. BCI in robotics

Early examples of robot control using bio-signals can be seen in works conducted by Bozinovski et al. such as controlling a robot using using EEG alpha rhythm signals [8]. Further work by Bozinovski can be seen as a robots trajectory was controlled using EEG/EOG signals gathered from the optical region of a human head [21]. An in depth summary of the pioneering works in this area of research is covered by Bozinovski [22], specifically focusing on early demonstrations of BCIs done in Europe. Iturrate et al. utilized P300 signals to control the navigation of a wheelchair. Using a virtual driving environment for the participants, the wheelchair was mobilized autonomously. To initiate movement, the user concentrates on a location on the map which is detected by the classifier and sent to the wheelchair. At this point, the wheelchair commences navigation to that point. Iturrates team also conducted experimentation with a robotic arm where the ErrP was considered as a reward signal. The robotic arm had 5 degrees of freedom for performing correct/incorrect reaching tasks. While the tasks were being performed the subjects brain data were recorded. Reach by Jiaxin and Zhang used a combination of EOG and EEGs signals to control a robot [2]. Zhang designed classifier which classified blink, gaze, wink, and frown. Linear discriminant analysis was used to classify the event-related potentials. Tests were conducted on two different robots: a humanoid robot NAO and a mobile robot Kobuki. [23] utilized a robotic quadcopter which was controlled in 3-D physical space using EEG signals. This was performed based on P300 signals which were used in order to classify left vs right hand, arm movement vs rest and constant forward velocity. These classified movements corresponded to commands used to navigate quadcopter. The research was also conducted by [24] on real-world car driving tasks which worked on the same principle as the previous work. In recent years it can be noted that most works were focused towards either EOG signals or P300 signals. There has not been much research conducted towards utilization error related potential up to the point of the mentioned research.

In late 2016s, Ehrlich and Cheng, came up with a neuro-based method for detecting erroneous robot action. Utilizing the concept of error related potentials in human-robot interaction, they were capable of determining whether robotic actions were incorrect. In their experimentation, a humanoid robot was placed such that it was gazing directly at a participant. A target stimuli was given in the form of a white rectangular size 3x3 cm appearing either left, right or above the robot head. Once the arrow blinked, the robot had to turn its head in that direction. Linear discriminant analysis was used to classify the erroneous signals [25]. Researchers from the Massachusetts Institute of Technology came up with a real-time system where the Baxter robot was used to perform specific tasks [26]. They proposed a feedback-based system that corrected the robots mistakes in real time. The robot simultaneously performed tasks such as classifying objects into two classes while utilizing the human brain signal as feedback. To classify the error related potentials they used an elastic net.With MITs research, the usage of error related potentials garnered interest in researchers for its various possible applications. Such applications include a multi-class object selection tasks where a robotic arm moves would hover over an object. A human subjects brain would then be read, and based on the response elicited by the subjects brain the robot would pick up the object its hand was hovering above [27]. Similar research was performed by [28] to pick and place objects with self-learning by using error-related potentials as feedback such that the robot learns simultaneously as per user choice. Another similar work was conducted by [29]. However, this work largely focused on navigating and identifying locations of interest in the local area of the patient and robot.

III. Methodology

The general methodology of the framework presented in this work can be seen in Figure 2. Robot motion is determined by a loop of navigation and mapping actions. The navigation is semi-controlled. User feedback generated from the visual stimuli of motion is used to command the robot at key points during navigation. The visual stimuli observed by the human eye is processed by the brain which generates signals that are in turn processed by an ErrP classifier. The classifier then outputs simple binary commands to the robot allowing the user to affect the motion of the robot at key points. The key points in this scenario are hallway intersections. When a hallway intersection is detected by the robot for the first time, it is added to a global map of intersections. The path the robot takes through each intersection on the map is stored and is used in the future to attempt to automatically navigate the robot to an end destination based on previous paths taken. If the robot ever reaches an intersection and navigates incorrectly, i.e. the robot begins to move toward the wrong exit of the intersection, then the user can correct that navigation.

A. ErrP Classifier

With the rapid growth and outperforming results of a machine and deep learning algorithms in the field of computer vision, natural language processing, speech analysis, there has been an interest towards the usage of these algorithms for EEG signal classification. As per the review of ten-year update in BCI applications for classification, convolution neural networks (CNN) have proved to have performed best even in EEG classification [30]. In this current work, we use Convolutional Neural Networks (CNNs), one of the deep learning architectures, for classification purposes. The significant component of CNNs is their ability to learn local patterns. Commonly CNNs are a set series of stages such as a convolutional layer, pooling layer, ReLU activation, and Fully connected. and some of the recent advancement layers such as Batch normalization and dropout layer. CNNs can have multiple layers, with their initial layers extracting the low-level features and as it goes deeper into the network more global and high-level features. The main advantage of using CNNs is its ability to learn from raw data which makes into the end-to-end analysis. Without any human supervision, CNNs can naturally detect the salient features from the input data. CNNs uses special convolution, pooling processes and also parameter sharing which makes it computationally efficient.

B. Navigation

In this work we assume that our environment is a structured building with four and three way intersections. We assume that our robot does not have a pre-existing map of its environment. The robot must build a map of intersection locations hich will be used later for the semi-controlled navigation scheme. The general problem of simultaneously localizing and mapping an

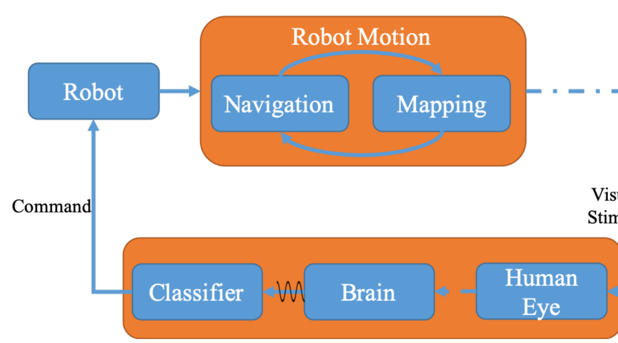


Fig. 2. Block diagram of overall system

environment is referred to as SLAM and is a common aspect of many robotic applications. SLAM involves estimating the pose of a robot while simultaneously building a map of its environment with no prior knowledge of the said environment. This task can be represented as a causality dilemma, where the pose of the robot is ascertained with respect to a global frame, or map, and the creation of a map requires knowledge of the robot's pose [31]. There are various algorithms used to solve this problem, many of which use a landmark-based approach [32]. Landmarks are a distinguishable feature in an environment which can be observed via devices such as a camera or laser rangefinder. Said devices can be used to measure properties such as the distance or bearing of the landmark relative to the robot. As the robot traverses the environment it continues to observe and collect landmarks while simultaneously building a map and localizing itself within the map. The sensed distance and bearing to each landmark can be combined with other factors like odometry readings to get a refined estimation of the robot pose and map. In this work, SLAM is accomplished using walls as landmarks, recognized with a laser rangefinder, and a particle algorithm, part of the ROS Gmapping package [33], accomplishes the localization and mapping. Our work builds on top of this package by identifying intersections when observed by the laser range finder and creating a history of paths taken based on which exits from each intersection are taken.

1) Intersection identification: Intersections are identified using the data received from the laser range finder. The data received by the laser range finder is broken down into groups of connected components, the raw dat taken from the laser range finder can be seen in Figure 3. These connected components are then considered to be candidate walls. Candidate walls are generally broken into two categories: corners and straight walls. Straight walls are easily identifiable as the majority of the points can be fit a to a simple linear equation. Corners are more complicated as they can be broken down into two walls separate walls that are perpendicular and intersect at a singel point. To accomplish this the RANSAC method outlined by Rissgard [34] is used. If the candidate corner is found to have two walls that intersect at a ninety degree angle then it is considered to be a corner. In order for a four-way intersection to exist there must be two corners and two walls, if this requirement is met then the algorithm moves onto the next stage. The two corners and the nearest edge of the other two walls are cataloged. If the distance between each of these nearest neighbors are all roughly equivalent, then the data received by the laser range finder is considered to be an intersection and we search for a location where lines drawn from one corner/edge to another corner/edge intersect. That point is the center of the intersection. A visualization of this process can be seen in Figure 4 . A similar algorithm is followed for three-way intersections.

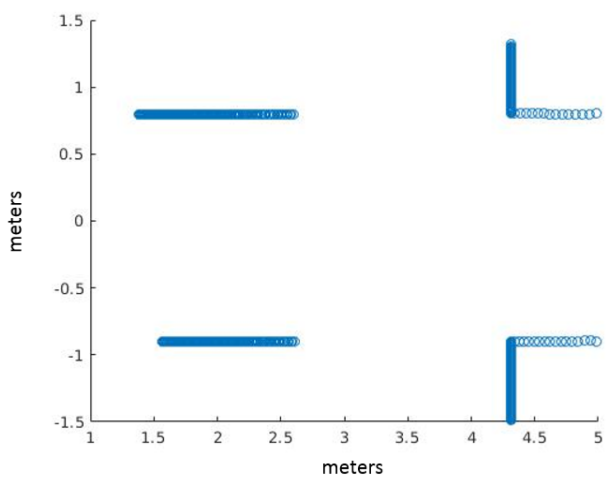


Fig. 3. Hallway intersection as seen by a laser range finder. Robot with laser range finder is located at position (0,0) and is orientated at 0 degrees.

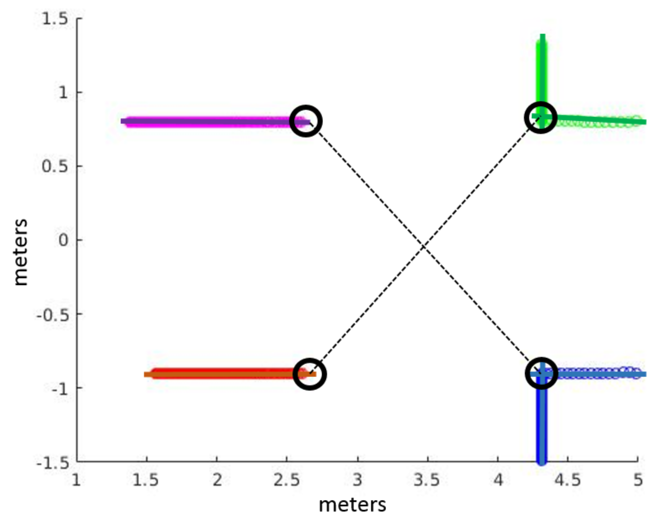
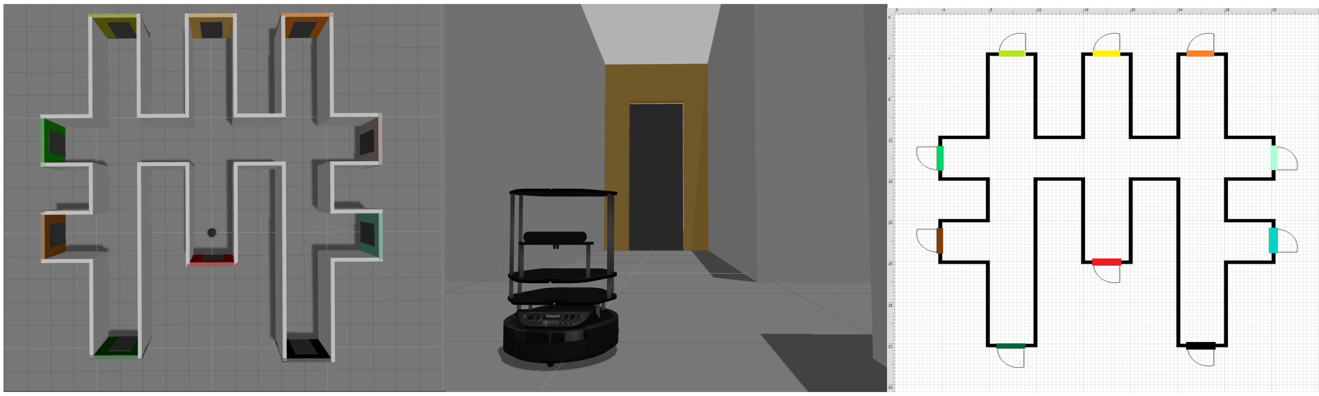


Fig. 4. Determining the center of the hallway intersection by identifying where lines drawn between the farthest neighbors of the four closest line ends and corners intersect

C. Intersection Exit Determination

When the robot approaches an intersection, ideally it would automatically rotate towards the most likely exit from the intersection that the user would pick based on the destination then continue on that path, thus minimizing the number of commands the user must give. However, the robot is unaware of the final destination. To achieve this goal a history of all previous paths taken are stored based on which exit of an intersection was taken. The robot will always automatically rotate towards whichever exit at an intersection has been taken the most number of times according to all previous paths. At each intersection, paths that do not include an exit taken will be removed from the pool of possible paths until eventually there is either a single path left, or no paths at all. Given the scenario where no paths remain the robot will randomly wander and adjust its navigation based on user input.



(a) (b) (c)

Fig. 5. Gazebo simulation visualization (a) Birds eye view; (b) Turtlebot view; (c) Schematic of simulated environment

1) Fully autonomous navigation: Fully autonomous navigation is used as a metric to compare our semi-controlled navigation algorithm against. It is accomplished using a simple explore algorithm that avoids walls and marks intersections while executing random motions. The robot attempts to move forward when it encounters a wall the robot rotates until there are no obstacles in front of it. If the robot detects an intersection then it marks the location of the intersection on the map and randomly exits the intersection. That exit direction for that particular intersection is then marked as 'used' and the robot will not exit from that direction in the intersection the next time it must move through the said intersection. This process continues until the destination is reached, which in the case of our experiment is a particularly colored door.

2) Semi-Controlled navigation: Semi-controlled navigation is accomplished using a combination of user input and movement. The robot moves along the center of a hallway until an intersection or wall is detected. In the case of a wall, the robot will rotate until the wall is no longer in front of the robot. In the case of the intersection, if it is the first time the intersection was detected, it gets marked and stored as part of the current path. If the intersection was already marked, then the robot searches the possible path pool and selects the exit from the intersection that matches the largest number of paths. The robot then rotates towards that intersection exit. If the user provides an 'incorrect' input once rotation is complete, then the robot rotates towards the next exit of the intersection in a clockwise fashion. If the robot ever exits an intersection in a way that no other path has done before, then the current path is considered new and is added to the robots memory. This process continues until the destination is reached.

IV. Experiment Protocol

A. Simulated Brain

EEG signals were collected from a publicly available dataset which contained 64 electrode recordings sampled at 512Hz of six subjects recorded over two sessions [35]. The procedure to classify the EEG data into ErrP or non-ErrP was as per the work is seen in [36]. The raw EEG signals were collected and preprocessed using EEGLAB [37]. Spatially, preprocessing was done using a common average filter and spectrally the preprocessing was done a using bandpass filter between 1Hz and 10Hz in order to remove the EEG artifacts. The electrode selection process was completed by visualizing the topographical maps of scalp activity. Out of 64 electrodes, two electrodes were chosen 'FCz' and 'Cz' and represented as a 2x512 matrix. A 5 layered CNN architecture was developed to classify the signals named ConvArch2 in [36].

Fig. 6. Hallway through which real world experimentation was conducted using a Turtlebot 2.

Fig. 7. Visualization of Turtlebot 2 identifying the destination door based on colored marker.

These classified signals were then filtered such that only correct signals were fed to the robotic agent and used for determining navigation direction at intersections. In other words, the ideal path for the robot form the start point to end point was predetermined, that path was used to pre-generate signals at each intersection corresponding to a correct exit, then those signals were fed to the robot during the experimental trials. The emphasis of this work being a proof of concept for future implementations with signals gathered in real time. The command from the signal comes in the form of a binary. Other more complicated classifiers can be used to determine directions of movement, such as forward/reverse/left/right, however, we instead choose to leverage the capability of the robot to complete simple navigation with the goal of reducing the effort and time needed for the user to send and have a signal successfully processed while maintining sucesfull global navigation to an end destination.

B. Simulation environment

Gazebo was used as the main simulation software for this work. An environment of colored doors and hallways was constructed and is visualized in Figure 5. The exact dimensions of the environment are visualized in Figure 5(c).

The walls represented obstacles in the environment while the colored doors represent destination. Various three-way and four-way intersections were designed as might commonly be seen in buildings. A simulated Turtlebot 2, also referred to as a turtlebot, was commanded via the Robot Operation System (ROS) for use as an agent in this simulation. The simulated turtlebot used a laser rangefinder and odometry information in order to map its surroundings. A simulated RGB color camera took pictures of the environment to determine if the turtlebot had reached the appropriate destination door. Tests conducted in this environment included running the Turtlebot autonomously with a gmapping SLAM algorithm [33] until it reached its destination, utilizing the methods outlined in section III subsection B sub-subsection i. As well as running the turtlebot as semi-controlled, using user input in the form of correct/incorrect at intersections in order to guide movement as described in section III subsection B sub-subsection ii.

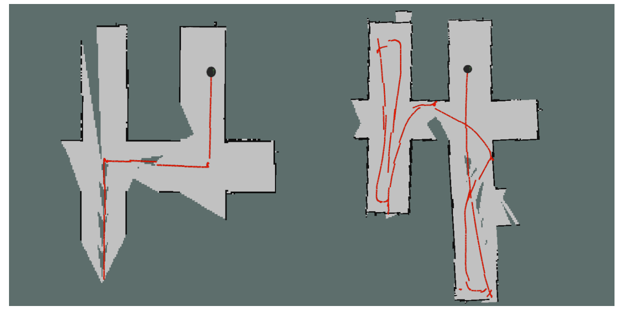
C. Real environment

Experimentation was also conducted in the real world using an 25-meter hallway with a single 90-degree turn leading to an 17-meter hallway. Colored markers were placed on doors to function as an icon in place of the colored doors in the simulation. No schematic of this hallway is given, however, a visual approximation of its shape and dimension can be viewed in Figure 9. Figures 6 and 7 also provide a visualization of the hallway environment.

V. Results

A. Simulation Results

The semi-autonomous navigation algorithm simulation results can be seen visualized in Figure 8(a). Results were measured in terms of the time required to complete the trial. The time to complete the simulation trial was 2 minutes and 34 seconds.



(a) (b)

Fig. 8. Results of simulation experimentation. The red line represents the path of the turtlebot. Black lines represent walls, white squares represent open space, and gray squares represent unknown. (a) Semi-Controlled Navigation results from simulated hallway using intersection identification and user inputs, visualized via Rviz.; (b) Autonomous navigation results from simulated hallway using randomized motion until a destination point is observed visualized via Rviz

(a) (b)

Fig. 9. Results of real world experimentation. The red line represents the path of the turtlebot. Back lines represent walls, white squares represent open space, and gray squares represent unknown. (a) Semi-Controlled Navigation results from real world hallway visualized via Rviz.; (b) Autonomous navigation results from real world hallway visualized via Rviz

The fully-autonomous navigation algorithm simulation results can be seen in Figure 8(b). The time to complete the simulation trial was 6 minutes and 15 seconds.

B. Real World results

The semi-controlled navigation algorithm results can be see in Figure 9(a). The time to complete the trial was 5 minutes and 56 seconds. The fully-autonomous navigation algorithm results can be seen in Figure 9(b). The time to complete the trial was 14 minutes and 32 seconds.

The results of the simulated and real world experimentation are summarized in Table I.

TABLE I
TIME TO REACH DESTINATION FOR SIMULATED AND REAL WORLD TRIALS OF SEMI-CONTROLLED AND FULLY AUTONOMOUS NAVIGATION SYSTEMS.

Simulated		Real World	
Semi-Controlled	Autonomous	Semi-Controlled	Autonomous
2min:34sec	6min:15sec	5min:56sec	14min:32sec

VI. Conclusions

This work shows proof of concept that the proposed semi-controlled navigation scheme has the potential to aid impaired user in reaching their destination inside a structured building with minimum effor required in terms of user input. The semi-controlled navigation scheme was compared against a simple autonomous navigation scheme. The semi-controlled navigation prioritized identifying and mapping intersections, then gathering simulated user input from a BCI, classifying the input, and using the resulting signal to determine the appropriate exit from an intersection for a given path. The autonomous navigation scheme simply avoided obstacles while randomly navigating the area in search of the specific destination represented as a colored door. Our results show that given basic simulation and experimentation the semi-controlled navigation scheme outperforms the autonomous navigation scheme based on the amount of time taken to reach a destination. It should be noted that the majority of the saved time comes from the autonomous navigation scheme taking an incorrect turn at an intersection and exploring an area of the environment that is not relevant to the destination of the user. If the simulation and experimentation had be done over a more extensive area without pre-filtering the classified signals then the results of the semi-autonomous navigation algorithm could be expected to decrease as incorrectly classified signals would cause the robot to exit an incorrect intersection. The results validate the potential for the proposed navigation scheme and provide the motivation to continue research into this work. Foremost, is the need for more extensive testing. Future work largely revolves around testing the proposed semi-controlled navigation scheme against a variety of complex navigation algorithms across much more detailed and complex environments. Optimistically, future work would also propose that inputs for controlling the robot be gathered in real time from a patient rather than through simulation.

References

[1] C. A. Stanger, C. Anglin, W. S. Harwin, and D. P. Romilly, "Devices for assisting manipulation: a summary of user task priorities," *IEEE Transactions on Rehabilitation Engineering*, vol. 2, pp. 256–265, Dec 1994.

[2] J. Ma, Y. Zhang, A. Cichocki, and F. Matsuno, "A novel eog/eeg hybrid human–machine interface adopting eye movements and erps: Application to robot control," *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 3, pp. 876–889, 2014.

[3] S. Manogna, S. Vaishnavi, and B. Geethanjali, "Head movement based assist system for physically challenged," in *2010 4th International Conference on Bioinformatics and Biomedical Engineering*, pp. 1–4, June 2010.

[4] A. Kübler, B. Kotchoubey, J. Kaiser, J. R. Wolpaw, and N. Birbaumer, "Brain–computer communication: Unlocking the locked in.," *Psychological bulletin*, vol. 127, no. 3, p. 358, 2001.

[5] J. R. Patterson and M. Grabois, "Locked-in syndrome: a review of 139 cases.," *Stroke*, vol. 17, no. 4, pp. 758–764, 1986.

[6] V. Gandhi, "Chapter 1 - introduction," in *Brain-Computer Interfacing for Assistive Robotics* (V. Gandhi, ed.), pp. 1 – 6, San Diego: Academic Press, 2015.

[7] T. C. Major and J. M. Conrad, "A survey of brain computer interfaces and their applications," in *IEEE SoutheastCon 2014*, pp. 1–8, March 2014.

[8] S. Bozinovski, M. Sestakov, and L. Bozinovska, "Using EEG alpha rhythm to control a mobile robot," in *IEEE/Engineering in Medicine and Biology Society Annual Conference*, 1988.

[9] E. Lopez-Larraz, M. Creatura, I. Iturrate, L. Montesano, and J. Minguez, "EEG single-trial classification of visual, auditive and vibratory feedback potentials in brain-computer interfaces," *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 4231–4234, 2011.

[10] S. Sur and V. Sinha, "Event-related potential: An overview," *Industrial psychiatry journal*, vol. 18, no. 1, p. 70, 2009.

[11] R. Chavarriaga, A. Sobolewski, and J. d. R. Milln, "Errare machinale est: the use of error-related potentials in brain-machine interfaces," *Frontiers in Neuroscience*, vol. 8, p. 208, 2014.

[12] M. Lehne, K. Ihme, A. Brouwer, J. B. F. van Erp, and T. O. Zander, "Error-related EEG patterns during tactile human-machine interaction," in *2009 3rd International Conference on Affective Computing and Intelligent Interaction and Workshops*, pp. 1–9, Sept 2009.

[13] R. Chavarriaga, I. Iturrate, Q. Wannebroucq, and J. d. R. Milln, "Decoding fast-paced error-related potentials in monitoring protocols," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 1111–1114, Aug 2015.

[14] J. Tessadori, L. Schiatti, G. Barresi, and L. S. Mattos, "Does tactile feedback enhance single-trial detection of error-related EEG potentials?," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 1417–1422, Oct 2017.

[15] P. W. Ferrez and J. del R. Millan, "Error-related EEG potentials generated during simulated braincomputer interaction," *IEEE Transactions on Biomedical Engineering*, vol. 55, pp. 923–929, March 2008.

[16] E. M. Ventouras, P. Asvestas, I. Karanasiou, and G. K. Matsopoulos, "Classification of error-related negativity (ERN) and positivity (Pe) potentials using knn and support vector machines," *Computers in Biology and Medicine*, vol. 41, no. 2, pp. 98 – 109, 2011.

[17] R. Chavarriaga and J. d. R. Millan, "Learning from EEG error-related potentials in noninvasive brain-computer interfaces," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 18, pp. 381–388, Aug 2010.

[18] H. Cecotti and A. Graser, "Convolutional neural networks for p300 detection with application to brain-computer interfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, pp. 433–445, March 2011.

[19] R. Schirrmeister, L. Gemein, K. Eggensperger, F. Hutter, and T. Ball, "Deep learning with convolutional neural networks for decoding and visualization of EEG pathology," in *2017 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, pp. 1–7, Dec 2017.

[20] S. Sakhavi, C. Guan, and S. Yan, "Parallel convolutional-linear neural network for motor imagery classification," in *2015 23rd European Signal Processing Conference (EUSIPCO)*, pp. 2736–2740, Aug 2015.

[21] S. Bozinovski, "Mobile robot trajectory control: From fixed rails to direct bioelectric control," in *Proceedings of the IEEE International Workshop on Intelligent Motion Control, IMC 1990*, 1990.

[22] S. Bozinovski and L. Bozinovska, "BrainComputer Interface in Europe: the thirtieth anniversary," *Automatika*, 2019.

[23] K. LaFleur, K. Cassady, A. Doud, K. Shades, E. Rogin, and B. He, "Quadcopter control in three-dimensional space using a noninvasive motor imagery-based brain–computer interface," *Journal of neural engineering*, vol. 10, no. 4, p. 046003, 2013.

[24] H. Zhang, R. Chavarriaga, Z. Khaliliardali, L. Gheorghe, I. Iturrate, and J. d R Millán, "Eeg-based decoding of error-related brain activity in a real-world driving task," *Journal of neural engineering*, vol. 12, no. 6, p. 066028, 2015.

[25] S. Ehrlich and G. Cheng, "A neuro-based method for detecting context-dependent erroneous robot action," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, pp. 477–482, IEEE, 2016.

[26] A. F. Salazar-Gomez, J. DelPreto, S. Gil, F. H. Guenther, and D. Rus, "Correcting robot mistakes in real time using eeg signals," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6570–6577, May 2017.

[27] H. Kolkhorst, M. Tangermann, and W. Burgard, "Guess what i attend: Interface-free object selection using brain signals," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7111–7116, IEEE, 2018.

[28] S. A. S. Bellary, "Human-robot cooperation using eeg signals with self-learning," Master's thesis, The University of North Carolina at Charlotte, 2019.

[29] X. Perrin, R. Chavarriaga, F. Colas, R. Siegwart, and J. d. R. Millán, "Brain-coupled interaction for semi-autonomous navigation of an assistive robot," *Robot. Auton. Syst.*, vol. 58, pp. 1246–1255, Dec. 2010.

[30] F. Lotte, L. Bougrain, A. Cichocki, M. Clerc, M. Congedo, A. Rakotomamonjy, and F. Yger, "A review of classification algorithms for EEG-based brain–computer interfaces: a 10 year update," *Journal of neural engineering*, vol. 15, no. 3, p. 031005, 2018.

[31] S. Thrun, "Probabilistic robotics," *Communications of the ACM*, vol. 45, no. 3, 2002.

[32] F. Bonin-Font, A. Ortiz, and G. Oliver, "Visual navigation for mobile robots: A survey," *Journal of Intelligent and Robotic Systems: Theory and Applications*, 2008.

[33] B. Gerkey and V. Rabaud, "Ros – gmapping." http://wiki.ros.org/gmapping, 2019. [Online; accessed 20-June-2019].

[34] S. Riisgaard, "SLAM for Dummies: A tutorial approach to Simultaneous Localization and Mapping," *citeulike.org*, 2005.

[35] R. Chavarriaga and J. d. R. Millan, "Learning from eeg error-related potentials in noninvasive brain-computer interfaces," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 18, pp. 381–388, Aug 2010.

[36] S. A. S. Bellary and J. M. Conrad, "Classification of error related potentials using convolutional neural networks," in *9th International Conference on Cloud Computing, Data Science and Engineering, Noida, India*, January 2019.

[37] A. Delorme and S. Makeig, "EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis," *Journal of Neuroscience Methods*, vol. 134, no. 1, pp. 9 – 21, 2004.