Engineering Project Detailed Research Plan

Please complete the information/questions begun/seen below in red ink. Save this document to your computer, and add a printed hardcopy to your application.

Date: 9/15/2023
Student Name: **Ananya Mehta**
Project Title: **A Motor Imagery Controlled All-Terrain Mobility Platform using Machine Learning**

Parts of the generic engineering project are listed below with descriptions to the students in the boxes. Students may provide a detailed research plan by describing their specific project in response to each box below.

Engineering Goal PROBLEM BEING ADDRESSED: All engineering projects solve a problem or fill a need. This goal should be a simple statement that describes the product being designed, the customer it is for and the problem or need it satisfies. Example" "The goal is to design a solar powered lawn mower for inexpensive automated lawn care for homeowners"

My Project Goal is:

To design and build a scale model of an all-terrain mobility platform and employ machine learning (ML) to control it using motor imagery (MI) signals from a dataset pre-recorded by an electroencephalogram (EEG) brain-computer interface (BCI) headset for paralyzed or locked-in patients.

Design Criteria Design criteria define the product's required performance . Examples: " It will have a minimum speed of 10 KPH". The output will be within 15% of the mean of the experimental data". "It must withstand 15 repetitions of a 10N impact" The International System of units (SI) required.

My Project Design Criteria are the following:

The patient generates motor imagery data by imagining motor movements. This data is captured using an Electroencephalogram (EEG) BCI headset, processed in real time using a ML algorithm and will then be turned into movement commands for the mobility platform.

- The ML algorithm must be able to process and classify 4 seconds of accumulated motor imagery data before the next 4 seconds worth of data is available 80% of the time
- The ML algorithm must be able to correctly classify left vs right imagined movement 80% of time
- The mobility platform must be able to turn 45 degrees in the given direction within 4 seconds of generating the movement command 80% of the time
- The mobility platform must be able to walk over 2 cm high obstacles with all hardware attached, including battery, motor driver, and on-board microcontrollers 80% of the time

Constraints

Constraints are factors that limit the engineer's flexibility such as size, cost, and time limitations. Examples: "It must fit in a box no larger than 10x20x50 cm" "The maximum cost is $50" "The software must run in real time on a Raspberry Pi"

My Project Constraints are the following:

- Cannot generate real-time data from the actual brain activity due to prohibitive costs of BCI headsets - must use pre-recorded data from a published dataset
- Patient in pre-recorded data could be mentally fatigued, impacting the data, causing false positives and negatives
- ML model cannot be generic for all users. It has to be retrained for each user prior to predictions
- Can control movement using only 2 inputs - left and right imagined hand movement
- Can only create a scaled model of the mobility platform
- Cannot carry excessive weight due to cost constraints for high torque motors

Chosen design

Provide your chosen design. For hardware, provide a sketch. For software, provide a flowchart. Indicate the components you will develop, and the libraries you are using.

My Project Design is shown below: insert photos, diagrams, or illustrations below.

The design of the proposed platform includes developing both hardware and software components - a mobility platform (hardware) that can navigate rugged terrain and an ML data pipeline (software) that can process the motor imagery EEG signals from the brain and translate them into physical movement for the patient. The following sections provide more details on the design of hardware and software components.

Hardware Design

The hardware design includes the following components (Figure 1) of the mobility platform:

- Transport Mechanism
- Hardware Assembly
- Motor Imagery Dataset

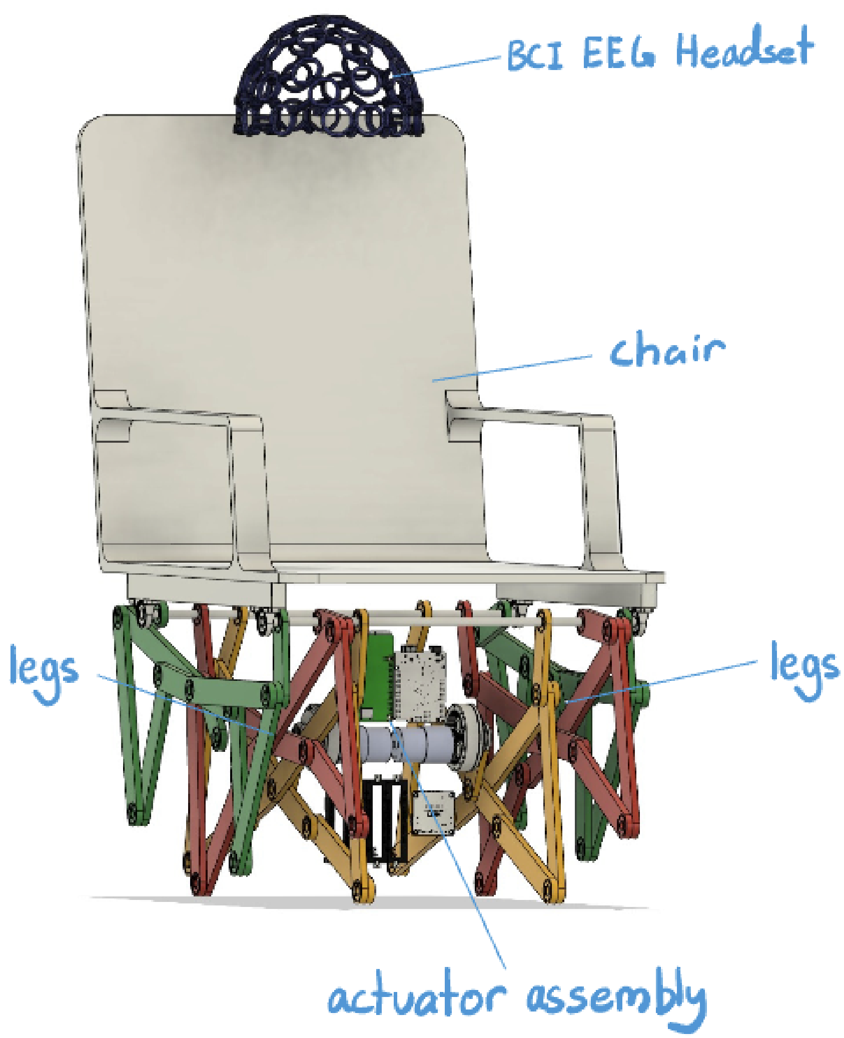

Figure 1: Physical Layout

The tentative size based on my CAD model is 600 mm (height) x 500 mm (width) x 500 mm (depth).

Transport Mechanism

This is the first component in the hardware design. The transport mechanism consists of a three-leg assembly on either side of the mobility platform corresponding to the right and left sides (Figure 7). The use of multiple legs will allow the mobility platform to travel over rugged terrain. The leg assembly for each side will be powered by a motor.

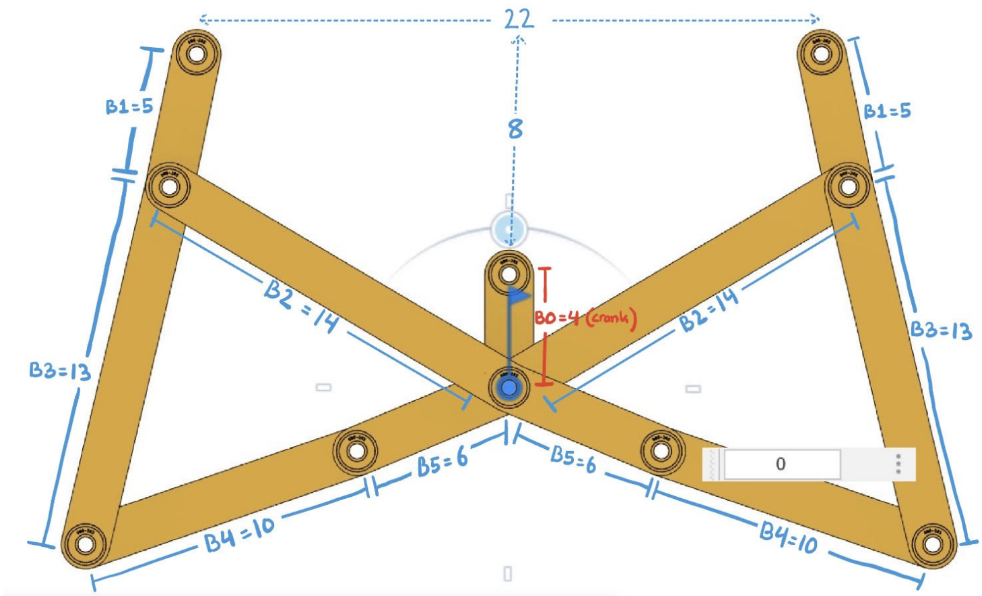

Figure 2: Single Leg Design

A single leg is a collection of linkages connected via revolute joints. The relative lengths of different linkages is shown above (Figure 2). Their absolute sizes will be determined once I have the materials, but the ratio will be maintained as specified. Linkages can be transformed into walking mechanisms by arranging the bars and pivot points in a specific configuration that mimics the leg movements of a walking creature. The linkage system to be used is called the ‘strider’ mechanism which has the advantage of consistent speed and less skidding when stepping on/off obstacles [2].

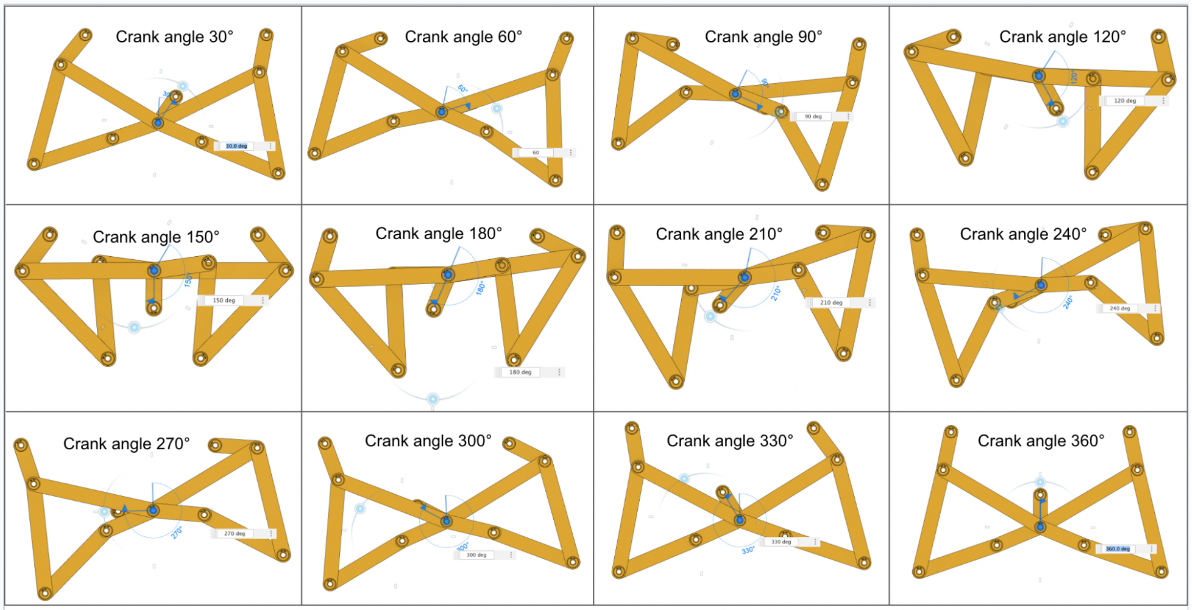

Figure 3: Single Leg Movement

Link to simulation: https://youtu.be/Dm06AQGp1ew

The movement in Figure 3 was captured from the simulation of the entire mechanism I modeled in Fusion 360 [6]. As the motor rotates the crank (the link directly attached to the motor), the leg moves to mimic a walking motion.

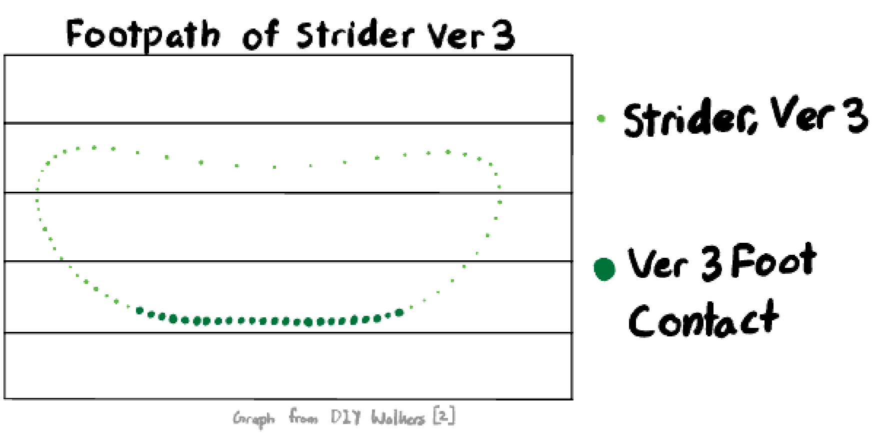

Figure 4: Trajectory of Strider Leg

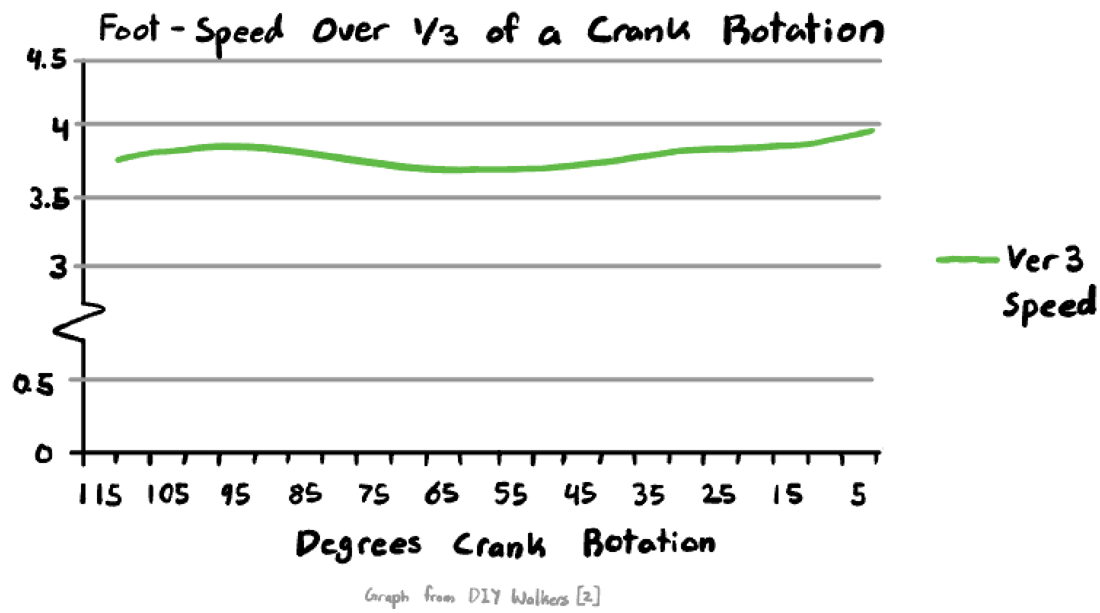

Figure 5: Speed of Strider Leg

The foot trajectory of a leg is shown in Figure 4. The speed is shown in Figure 5. To reduce the bounce during the walking motion, the trajectory should be as flat as possible, and the horizontal speed should be uniform to minimize skidding or shaking. The bounce can also be reduced using compressible feet. These requirements are well suited for a mobility platform. The horizontal foot-speed is calculated by taking the difference between the X values of adjacent points of the foot-path (Figure 4) when the feet are in contact with the ground. Therefore, when the plotted points are close together (Figure 4) the speed is less, and when they are far apart the speed is more [2]. The unit for the speed is length-units/sec where the 'length-units' are used to specify the link sizes in Figure 2.

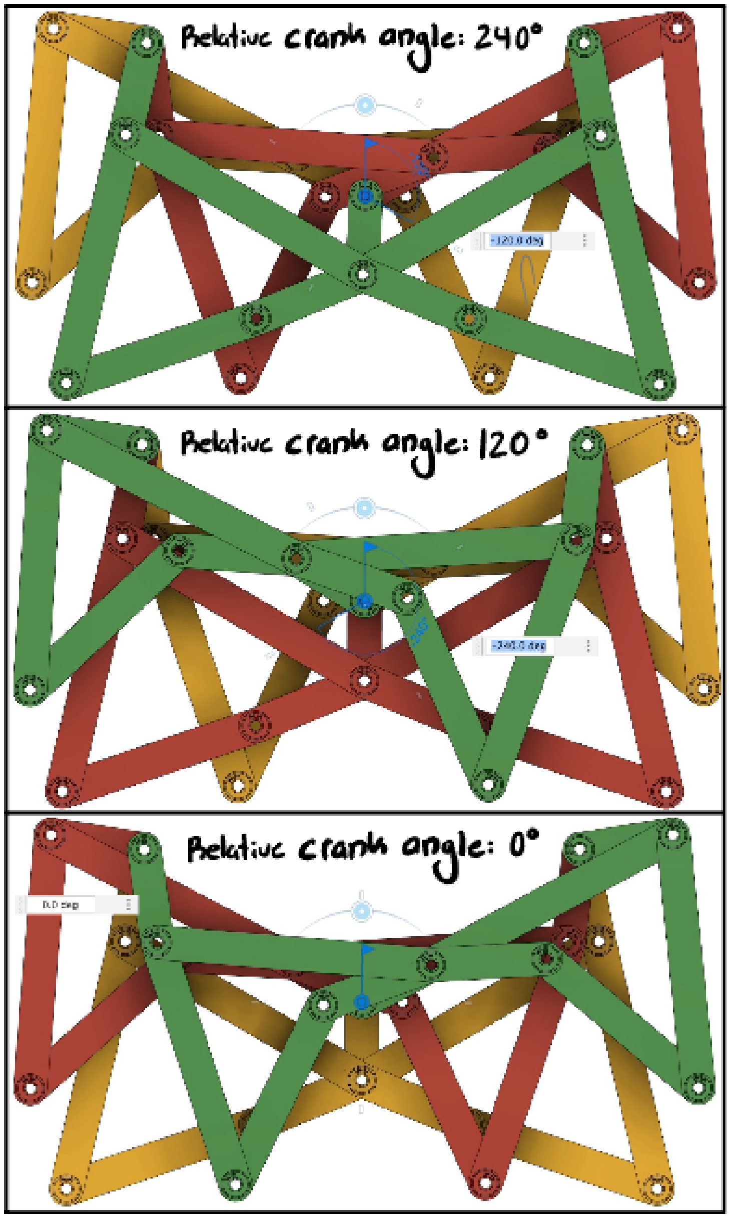

Figure 6: Movement of Strider Legs

Link to simulation: https://youtu.be/GfFwzZL4ReA

The three legs are connected together through a crankshaft that enables them to operate with a phase shift of 120 degrees relative to each other.

In the first image (Figure 6), the red leg is about to touch the ground, the yellow leg has just left the ground, and the green leg is completely on the ground.
In the second image (Figure 6), the yellow leg is about to touch the ground, the green leg has just left the ground, and the red leg is completely on the ground.

In the third image (Figure 6), the green leg is about to touch the ground, the red leg has just left the ground, and the yellow leg is completely on the ground.

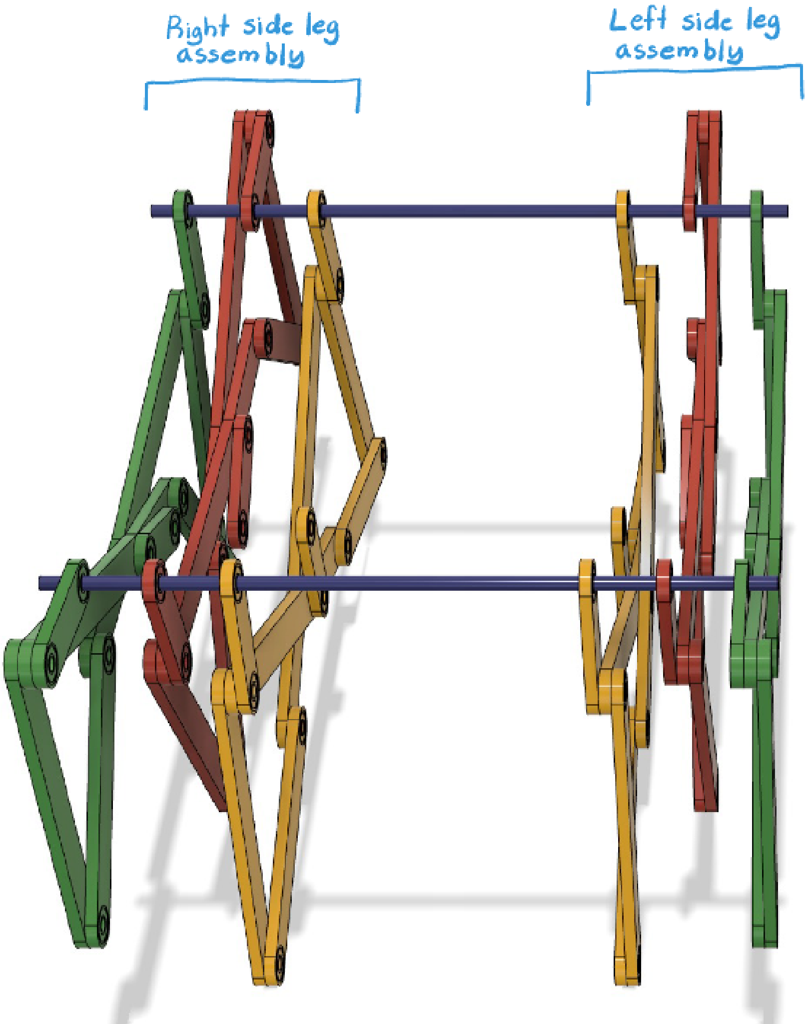

Figure 7: Six Leg Strider Leg Mechanism

The three legs on each side (Figure 7) provide additional stability by ensuring that there are at least 2 points of contact with the ground at all times.

Hardware Assembly

This is the second component in the hardware design. The hardware assembly consists of the following components (Figure 8):

- a pair of motors

- a pair of gearboxes
- motor driver
- motor controller (arduino)
- raspberry pi
- battery

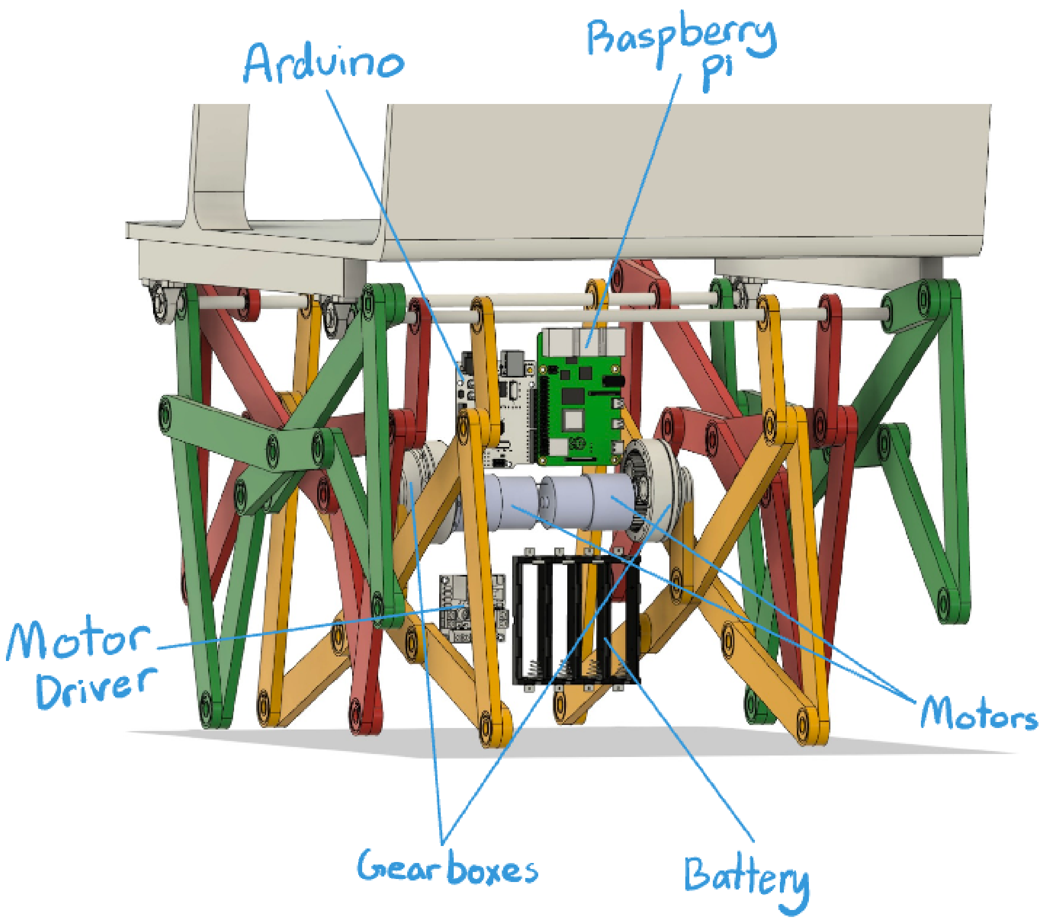

Figure 8: Hardware Assembly Layout

To drive the leg assemblies, I intend to employ two DC motors or stepper motors. Given the need for high torque, it will be necessary to increase the torque output at the expense of speed through gearboxes. Gearboxes are mechanical components designed to transmit and manipulate mechanical power or motion between rotating elements. They achieve this through the utilization of gear sets of varying sizes to alter the rotational speed, and torque magnitude.

For motor control, a motor driver circuit will be employed to precisely regulate both the speed and direction of the electric motor. An Arduino microcontroller will be responsible for generating motor movement commands, which will subsequently be transmitted to the motor driver for execution. Power will be supplied via a battery through a voltage regulator, providing 5V to Arduino and 12V to the motor driver.

Hardware Connections

Different components in the BCI headset and hardware assembly with their connections are shown in Figure 9.

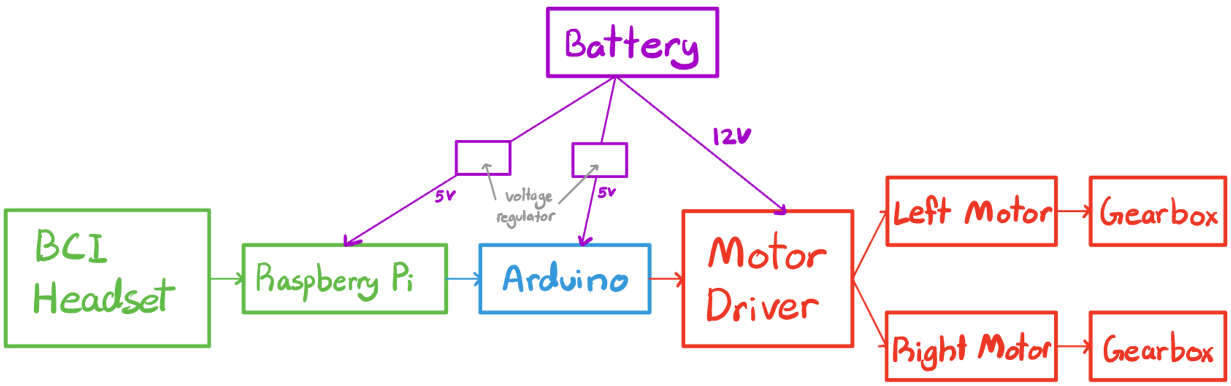

Figure 9: Hardware Connections

The Raspberry Pi is a single-board computer featuring various models, with typical specifications including a Broadcom CPU, RAM, USB ports, HDMI output, GPIO pins, and support for various operating systems, suitable for a wide range of computing projects. In this project, I will use Raspberry Pi 3B for computationally intensive operations, such as sourcing the EEG MI data from the stored dataset, and processing it using ML.

I will also use an Arduino microcontroller to exclusively control the hardware assembly using the movement commands generated by the Raspberry Pi. Arduino is an open-source microcontroller platform, typically equipped with analog and digital input/output pins, used for creating interactive electronic projects and prototypes. The Arduino and Raspberry Pi will be connected using a UART serial interface.

The Arduino interfaces with the motor driver through General Purpose Input Output (GPIO) pins. The movement commands, signaled through the GPIOs, will be used by the motor driver to control the direction and speed of the two motors.

A gearbox on either side will reduce the speed and amplify the torque.

Motor Imagery Dataset

This is the third component in the hardware design. Brain-Computer Interfaces (BCIs) are communication pathways that connect the human brain with external devices or computer systems. By decoding and interpreting the electrical activity of the brain, BCIs offer individuals with motor disabilities an alternative way to communicate and interact with the world, restoring a sense of independence and improving their quality of life.

Motor Imagery (MI) is a cognitive process involving the mental simulation of movement and has gained significant attention in the BCI field. It allows individuals to imagine performing specific motor tasks without physically executing them. This is made possible through event-related desynchronization/synchronization (ERD/ERS). Sensorimotor rhythms (SMRs) are synchronized brain waves over the sensorimotor cortex in three different frequency bands: μ (8–12 Hz), β (18–30 Hz), and γ (30–200 Hz). EEG recording is mostly limited to μ and β bands. SMR amplitude is higher during the idle stage called as event-related synchronization (ERS) and the amplitude decreases when the sensorimotor areas are active due to a certain motor task or even during motor imagery (MI). This decrease in SMR amplitude is called event-related desynchronization (ERD). The ERD signal is used for MI-related BCI.

The dataset I will use contains 64-channel EEG data, with MI data of left and right imagined hand movement obtained from 109 healthy patients [3]. The data will be stored on a Raspberry Pi, which will also perform the data processing.

The subjects in this dataset performed different motor/imagery tasks while 64-channel EEG were recorded using the BCI2000 system (http://www.bci2000.org). Each subject performed 14 experimental runs: two one-minute baseline runs (one with eyes open, one with eyes closed), and three two-minute runs of each of the four following tasks:

- Task 1: Open and close left or right fist
- Task 2: Imagine opening and closing left or right fist
- Task 3: Open and close both fists or both feet
- Task 4: Imagine opening and closing both fists or both feet

I will be using the data corresponding to Task 2. The data processing is explained further in the document (Figure 11).

What is a BCI Headset?

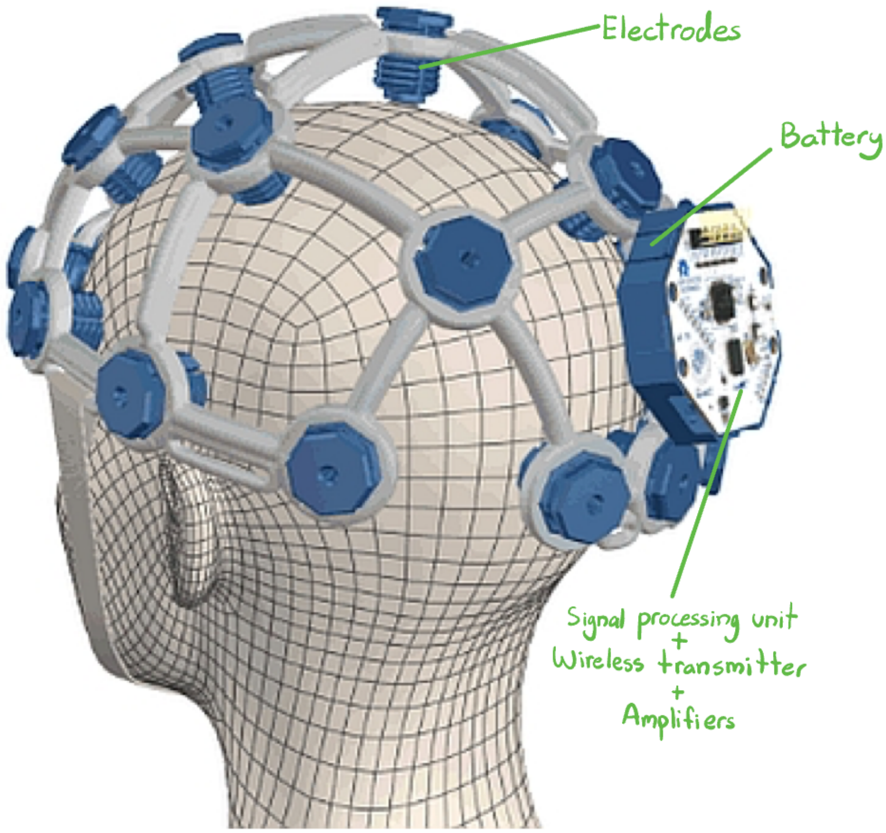

Figure 10: BCI Headset

A typical EEG BCI headset (Figure 10) consists of [9]:

- Electrodes - To sample electrical data
- Amplifiers - To amplify electrical data, as EEG signals are very weak
- Signal Processing Unit - To filter noise and prepare for further analysis
- Wireless Transmitter - To transmit data to external device for analysis
- Battery - To power headset independently

****I will not collect actual data using a BCI headset, but instead use pre-recorded data from a published dataset and store it on the Raspberry Pi.**

This concludes the hardware design of the 3 components to be developed for the mobility platform:

- Transport Mechanism
- Hardware Assembly
- Motor Imagery Dataset

Software Design

The software component consists of 2 pieces of software - one executing on the Raspberry Pi and the other on Arduino.

Raspberry Pi Software

This is the first software component to be developed. The entire process starts with the processing of pre recorded data, and classifying it into left or right commands to control the mobility platform. The data will be first processed using MNE, and will then be classified using ML on the Raspberry PI SBC. This will result in the movement commands that will be sent to Arduino for steering the mobility platform.

MNE (Magnetoencephalography and Electroencephalography) is a software library and toolbox used in the field of neuroscience and neuroimaging. It is specifically designed for the analysis of brain activity data acquired through electroencephalography (EEG) techniques. MNE provides a comprehensive set of tools for processing, analyzing, and visualizing EEG data [7].

Data Processing

I will use the following steps to process the MI data:

- Load data from Raspberry Pi into RAM for processing
  MNE provides a convenient way to import the MI data [3] using the load_data() API of the 'eegbci' module in the MNE package
- Preprocess or cleaning the MI data
  This step involves the application of various signal processing techniques to enhance the quality and extract relevant information from the recorded brain signals associated with motor imagery tasks. I will use filter_data() API in the filter module of the MNE package to discard the frequencies beyond 50 Hz and eliminate interference from the power line voltage.
- Detect experimental events
  The dataset includes an annotation file which includes timestamps for events, such as the onset of imagined movement. These tiestamps will be used to extract the relevant MI data from the consolidated capture using the events_from_annotations() API in the MNE package.
- Epoch the continuous data
  Epoching for MI data involves dividing the continuous EEG signal into discrete time segments or epochs, typically aligned with the onset of motor imagery tasks. This facilitates the analysis of brain activity patterns during specific events. I will use the epochs() API from the MNE package to separate the data into left and right epochs.
- Signal Processing. It involves the following steps:

- ○ The first step is to calculate the power spectral density (PSD) for the left and right epochs data and compare the two to verify the Event Related Desynchronization (ERD) in the μ band (8-12 Hz) of SMR frequencies. I will use compute_psd() API in the epochs module of the MNE package.
- ○ The second step is to use the Common Spatial Pattern (CSP) algorithm to identify specific combinations of 64 electrode positions on the scalp that can best distinguish between different MI tasks by analyzing the electrical activity of the brain [8].

ML Classification

Supervised learning is a type of machine learning where an algorithm is trained on a labeled dataset, meaning that each input data point is associated with the correct output or target. The algorithm learns to make predictions or decisions based on this training data. In other words, supervised learning involves teaching the algorithm to map input data to the correct output by providing it with examples of the correct answers during the training process. This type of learning is commonly used for tasks such as classification (assigning data points to predefined categories or classes) and regression (predicting continuous values) [5].
I will use Linear Discriminant Analysis (LDA) to maximize the difference between left and right data. LDA is a classifier which separates two or more classes or groups by finding a linear combination of features (variables) that maximizes the separation between these classes.
If the data is not clearly distinguishable as left or right, it will be classified as a command to keep moving forward. The resulting classifications will be communicated to the Arduino through simple text-based commands.

Arduino Software

This is the second software component to be developed. Arduino will take the classification results from the Raspberry Pi and convert them into commands for the motor driver. To write and compile the code, I will use Arduino IDE. I will choose between a DC motor or stepper motor based on their actual performance in the mobility platform.

I plan to use the following steps to program the motors.

- Identify the GPIOs connected to the motor driver, and configure them as output pins. This will allow them to be controlled using the Arduino.
- Set direction (clockwise or counterclockwise) and speed for both the motors, and start moving forward.
- Wait for a command from Raspberry Pi in a while loop. When classification is received, process into commands and increase the speed of the left or right

motors accordingly. When the left motor is sped up, the mobility platform will turn right, and vice versa. If the command received is to go straight forward, keep both motors at the same speed.

I will also write separate code to validate the movement related criteria independent of the MI data.

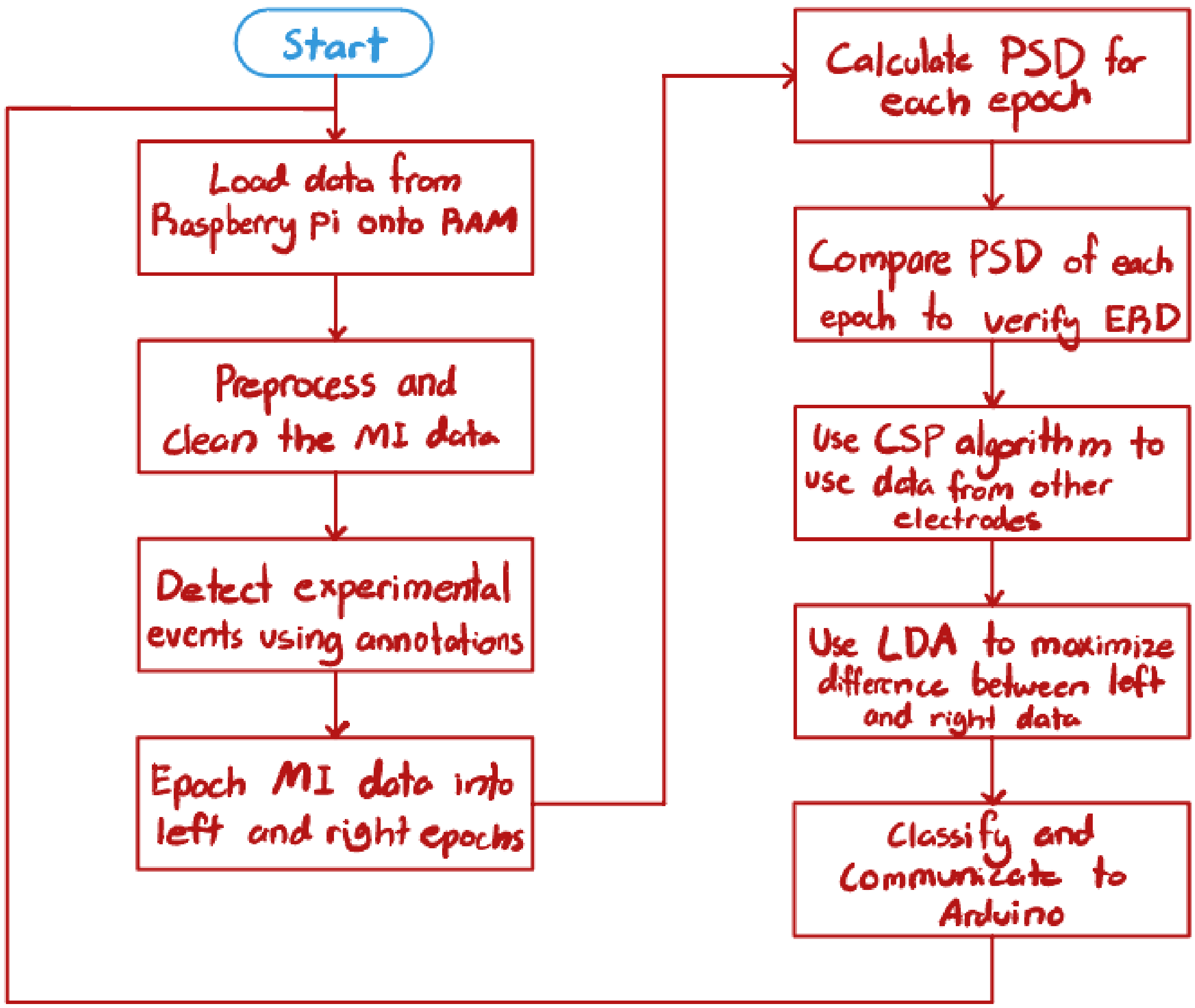

Figure 11: Raspberry Pi Flowchart

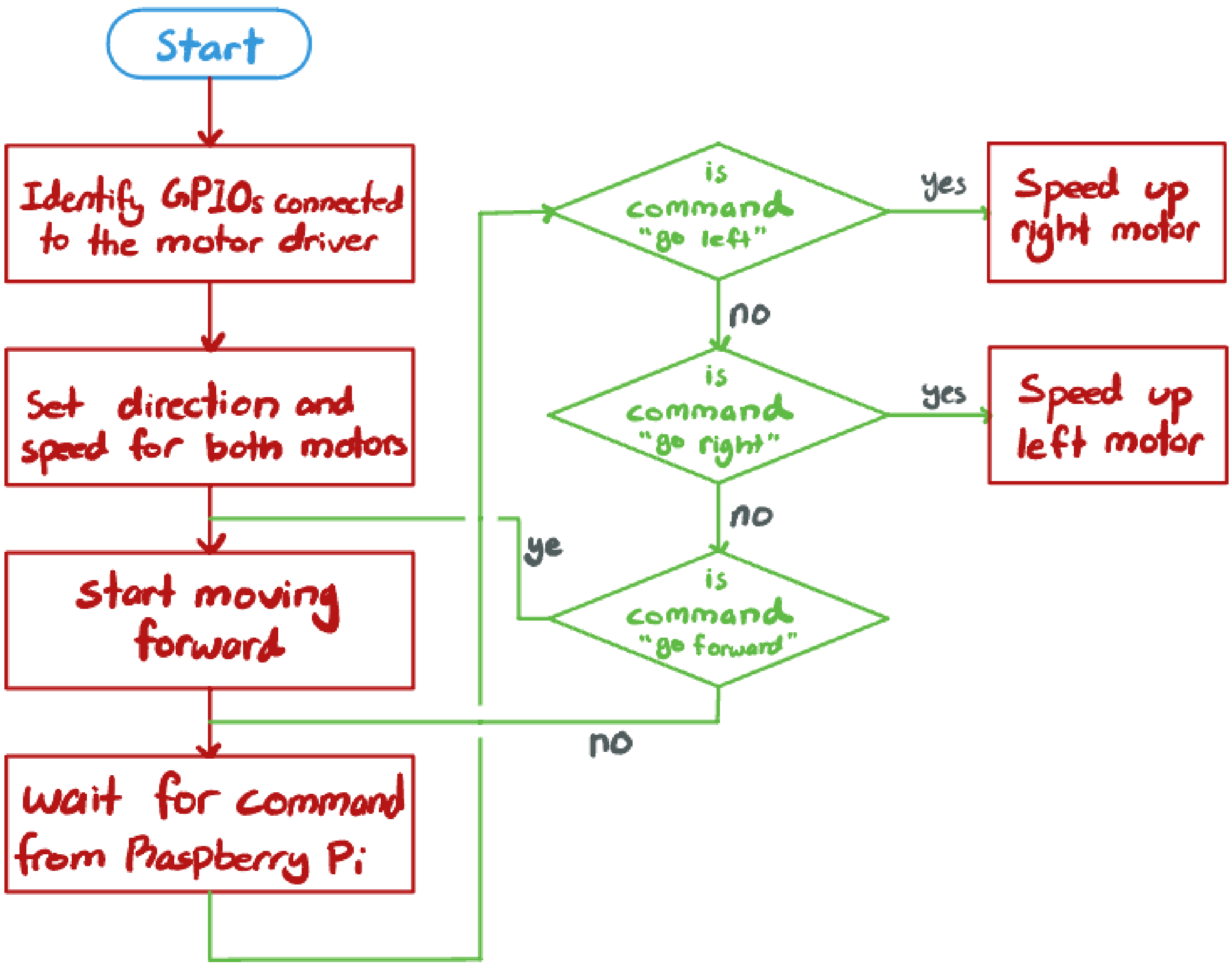

Figure 12: Arduino Flowchart

This concludes the software design of the 2 components to be developed for the data processing in the mobility platform:

- Raspberry Pi software
- Arduino software

Testing Procedures

Test and evaluate your prototypes against the design criteria listed above to show how well the product meets the need/goal. Provide a test plan describing how you will test the design criteria and constraints you listed above., How will you analyze the data? If the product requires human testing please fill out and append https://science-fair.org/wp/wp-content/uploads/2015/10/Research-Plan-Human-Participants.docx

I test and analyze my prototypes using the following methods:

The ML algorithm must be able to process and classify 4 seconds of accumulated motor imagery data before the next 4 seconds worth of data is available 90% of the time

1. Obtain 1 event (4 seconds) of motor imagery data from the dataset. An event records a single motor imagery hand movement (either left or right).
2. Record the timestamp when 4 seconds of data is accumulated and is ready for processing
3. Record timestamp when ML algorithm returns results
4. Observe whether ML algorithm returns results before next event (4 seconds) is accumulated
5. Record the results
6. Repeat steps 1-5 5 times
7. Repeat steps 1-6 for 10 patients

The ML algorithm must be able to correctly classify left vs right imagined movement 90% of time

1. Obtain 1 event (4 seconds) of motor imagery data from the dataset. An event records a single motor imagery hand movement (either left or right).
2. Feed accumulated data from event into ML algorithm
3. Record expected result vs result from the ML algorithm
4. Repeat steps 1-3 10 times, each with a different random event from a single patient. There are 30 events per patient to choose from.
5. Repeat steps 1-4 for 10 patients

The mobility platform must be able to turn the mobility platform 45 degrees in the given direction within 4 seconds of generating the movement command 90% of the time

1. Run program on the mobility platform to turn 45 degrees in a random direction (left or right)
2. Record time the platform takes to turn 45 degrees
3. Record results
4. Repeat steps 1-3 20 times

The mobility platform must be able to walk over 2 cm high obstacles with all hardware attached, including battery, motor driver, and on-board microcontrollers 90% of the time

1) Place 2 cm high obstacle in front of mobility platform
2) Run program on the platform to walk straight forward
3) Record whether the platform was able to successfully navigate through the obstacle
4) Repeat steps 1-3 20 times

Bibliography

List at least five (5) major references (e.g. science journal articles, books, internet sites & dates of review) from your literature review. If you plan to use vertebrate animals, one of these references must be an animal care reference.

1. Swee, Sim Kok, et al. "EEG controlled wheelchair." *MATEC Web of Conferences*, vol. 51, 2016, p. 02011, https://doi.org/10.1051/matecconf/20165102011.
2. Vagle, Wade. "Walker ABC's." *DIY Walkers*, DIY Walkers, 2023, www.diywalkers.com/walker-abcs.html.
3. "EEG Motor Movement/Imagery Dataset." *EEG Motor Movement/Imagery Dataset v1.0.0*, PhysioNet, 9 Sept. 2009, www.physionet.org/content/eegmmidb/1.0.0/.
4. Mulder, Th. "Motor Imagery and Action Observation: Cognitive Tools for Rehabilitation." *Journal of Neural Transmission (Vienna, Austria : 1996)*, U.S. National Library of Medicine, 2007, www.ncbi.nlm.nih.gov/pmc/articles/PMC2797860/.
5. Portilla, Jose. "Python for Machine Learning & Data Science Masterclass | Udemy." *Udemy*, Udemy, Sept. 2021, www.udemy.com/course/python-for-machine-learning-data-science-masterclass/.
6. Autodesk. (2023). Autodesk Fusion 360 [Computer software]. Autodesk
7. MNE Developers. "Python Homepage." *MNE*, MNE, 2023, https://mne.tools/stable/index.html.

8. MNE Developers. “Motor Imagery Decoding from EEG Data Using the Common Spatial Pattern (CSP).” *Motor Imagery Decoding from EEG Data Using the Common Spatial Pattern (CSP) - MNE 1.5.1 Documentation*, 2023, https://mne.tools/stable/auto_examples/decoding/decoding_csp_eeg.html#footcite-schalketal2004.
9. “3D-Printed Brain-Sensing Headset Is Open Source.” *eeNews Europe*, eeNews Europe, 15 Nov. 2015, www.eenewseurope.com/en/3d-printed-brain-sensing-headset-is-open-source/.