جامعة الملك سعود
King Saud University

College of Computer and Information Sciences

Computer Engineering Department

Controlling a Robot Using Electroencephalogram (EEG) Signals

CEN 493- Graduation project

Team Members

Abdullah Almingash (441102329)

Saad Aldobaian (441102889)

Abdelrahman Fatouh (441105960)

2^{nd} Semester 2024

Advisor:

Prof. Ghulam Muhammad

Table of Contents

Abstract 5
Chapter 1 The Problem Statement 6
Section 1.1 Need Statement 6
Section 1.2 Objective Statement 6
Chapter 2 Research Survey 7
Section 2.1 Introduction 7
Section 2.2 What is EEG? 7
Section 2.3 Application of EEG 8
Section 2.4 Our Application for EEG (EEG-Based Robotics System) 8
Section 2.5 EEG and AI 9
Section 2.6 Related Research 9
Section 2.7 Limitations of Current Designs or Technology 10
Section 2.8 Current Systems 11
Chapter 3 Requirements Specification 12
Section 3.1 User Needs 12
Section 3.2 Objective Tree 13
Section 3.3 Requirement Specification 14
Chapter 4 The Design 16
Section 4.1 Concept Overview 16
Section 4.2 Design Specifications 21
Design Level-0: 21
Design Level-1: 22
Section 4.3 Preliminary Work 27
Section 4.4 Flowchart 31
Chapter 5 Implementation and Verification 32
Section 5.1 Prototype Implementation 32
Section 5.2 Design Verification 39
References 42
Appendix A The Project Plan 44
Section A.1 Work Breakdown Structure 44
Section A.2 Gantt Chart 45
Section A.3 Cost Estimation 45
Appendix B Program code 46

List of Figures

Figure 2.1: Brainwave types 7

Figure 3.1: Objective Tree 13

Figure 4.1: Emotiv EPOC+ VS Emotiv Insight 2.0 20

Figure 4.2: Level-0 21

Figure 4.3: Level-1 22

Figure 4.4: Lithium Battery 23

Figure 4.5: Emotiv EPOC+ 24

Figure 4.6: Raspberry Pi 25

Figure 4.7: Servo motor, mg996r 26

Figure 4.8: Effective MI bands 27

Figure 4.9: (A) Sensor locations. (B) Recorded MI-EEG. 27

Figure 4.10: Confusion matrix 29

Figure 4.11: (A) ATCNet model. (B) Detailed ATCNet model. 30

Figure 4.12: System Flowchart 31

Figure 5.1: Humanoid A CM530 robot 33

Figure 5.2: (A) 17b Dof robot before fitting the parts. (B) 17b Dof robot while fitting the parts 33

Figure 5.3: 17b Dof robot after fitting the parts 34

Figure 5.4: Recording Dataset 35

Figure 5.5: Dataset recording program 35

Figure 5.6: (A) and (B) shows the first attempt results 36

Figure 5.7: (A) and (B) shows the second attempt results 37

Figure 5.8: (A) and (B) shows the third attempt results 37

Figure 5.9: (A) and (B) shows the fourth attempt results 38

Figure 5.10: Integrated system components 38

Figure A.1: Gantt Chart 45

List of Tables

Table 3.1: Requirement Specification 14
Table 4.1: Concept table 16
Table 4.2: Power Source (Strengths vs Weaknesses) 17
Table 4.3: Controller Board (Strengths vs Weaknesses) 18
Table 4.4: Connectivity (Strengths vs Weaknesses) 19
Table 4.5: Headset (Strengths vs Weaknesses) 20
Table 4.6: Level-0 table 21
Table 4.7: Level-1 Lithium Battery 23
Table 4.8: Level-1 Emotiv EPOC+ 24
Table 4.9: Level-1 Raspberry Pi 25
Table 4.10: Level-1 Robot motors 26
Table 5.1: Unit Test #1, Servo Motors 39
Table 5.2: Acceptance Test #1, Robot Motions 40
Table A.1: Work Breakdown Structure 44
Table A.2: Cost Estimation 45

Abstract

The project goal is to control a robot using Electroencephalogram (EEG) Signals to help disabled and elderly individuals accomplish tasks that they cannot do. When they start imagining moving a part of their body, even when they cannot move it, the brain generates electrical signals, this concept is called Motor Imagery (MI). Electrical signals can be recorded using an EEG headset to have the ability to study them, we preferred to use Emotiv EPOC+ to record the signals. The Raspberry Pi analyzes the signals collected from the headset by applying deep learning with an attention mechanism for the MI classification, resulting in the classification of the commands that disabled and elderly individuals want to do. Therefore, the Raspberry Pi controls the robot to perform this command.

We develop a small MI dataset for the project. The proposed MI dataset consists of three commands: raise the left hand, raise the right hand, and raise both hands. Based on the command, the robot acts accordingly. We are using a deep learning model, which is an Attention Temporal Convolutional Network (ATCNet). ATCNet is a model that can deal with MI-EEG data with high accuracy; it consists of three main blocks and a sliding window to augment MI data and boost the performance of MI classification efficiently.

Chapter 1 The Problem Statement

Disabilities and elderly individuals pose significant challenges, making it difficult for them to efficiently carry out their daily tasks. These individuals often require assistance, but unfortunately, suitable help may not always be readily available, leading to disruptions in their routines. Consequently, there is a pressing need for an innovative solution that can effectively enhance their quality of life and support them in living more independently and effortlessly.

Section 1.1 Need Statement

Many disabled or elderly individuals face challenges in performing daily tasks and often rely on caregivers for assistance. However, caregivers may not be available all day. According to the World Health Organization, an estimated 16% of the global population, or approximately 1.3 billion people, currently live with some form of disability, which can significantly impact their daily activities. Therefore, there is a clear need to develop a robot control system using electroencephalogram (EEG) signals, to empower individuals with disabilities in controlling and interacting with their environment, enhancing their independence and quality of life.

However, existing technology for controlling robots using EEG signals faces limitations. While some systems offer good accuracy, they often come with high costs, making them inaccessible to many individuals. On the other hand, low-cost alternatives often suffer from poor accuracy and require lengthy calibration times. According to that, not all caregivers can afford the good ones, so they need to buy the ones with limitations.

Section 1.2 Objective Statement

The project is focused on designing a humanoid robot controlled through EEG signals utilizing deep learning for pattern recognition. The objective is to apply EEG-based control for diverse applications, with potential applications in the medical and robotics fields. The project involves the use of a specialized headset to capture brain signal activity across various frequencies and channels. These signals will be input into a trained deep learning model to classify motion patterns, and subsequently, the identified classes will be transmitted to the humanoid robot for the execution of corresponding actions. Therefore, in this project, we develop a system involving a humanoid robot, EEG signals, and deep learning algorithms.

Chapter 2 Research Survey

Section 2.1 Introduction

A brain-computer interface (BCI), where EEG signals play a great role, has been a major topic since the 1920s [1], and since then the technology has developed rapidly till nowadays. Now we find it developed to a level where even a regular person -not a researcher- can buy a BCI device and experience the technology by himself.

Section 2.2 What is EEG?

An EEG, or electroencephalogram, is a signal that quantifies the electrical impulses generated by our brain. This process is non-invasive and utilizes little metallic discs known as electrodes, which are affixed to the surface of our scalp. The electrodes detect the minuscule electrical impulses generated by our brain cells and transmit them to a device that captures them as oscillating lines on a display or paper.

Once the electrodes are in place, the participant will be asked to relax and close his eyes. He may also be asked to do some tasks, such as opening and closing his eyes, breathing deeply, or reading aloud. These tasks can help to stimulate different areas of our brain and produce different types of brain waves.

Each channel will read different frequencies of brain activity; the frequency bands are delta (0.5 to 4Hz), theta (4 to 7Hz), alpha (8 to 12Hz), sigma (12 to 16Hz), and beta (13 to 30Hz). Each frequency band corresponds to certain activities [2]. Figure 2.1 shows a list of different frequency bands and the corresponding state of the brain.

Brainwave Type	Frequency Range (Hz)	State of the brain
Delta (δ)	0.1Hz to 3Hz	Deep, dreamless sleep, non-REM sleep, unconscious
Theta (θ)	4Hz to7Hz	Intuitive, creative, recall, fantasy, imaginary, dream
Alpha (α)	8Hz to12Hz	Relaxed, but not drowsy, tranquil, conscious
Low-range Beta (β)	12Hz to 15Hz	Formerly SMR, relaxed yet focused, integrated
Mid-range Beta (β)	16Hz to 20Hz	Thinking, aware of self & surroundings
High-range Beta (β)	21Hz to 30Hz	Alertness, agitation
Gamma (γ)	30Hz to 100+Hz	Motor Functions, higher mental activity

Figure 2.1 Brainwave types [2]

Section 2.3 Applications of EEG

Since the development of EEG, it has been used in various applications. Here we list some of them:

- Epilepsy: showing abnormal brain wave patterns that occur during seizures.
- Brain tumors: EEGs can help to locate brain tumors by showing areas of the brain where there is abnormal electrical activity.
- Head injury: EEGs can be used to assess the severity of a head injury and to monitor for brain damage.
- Encephalitis: This is an inflammation of the brain that can be caused by a virus or other infection. EEGs can help to diagnose encephalitis by showing widespread abnormalities in the brain waves.
- Sleep disorders: EEGs can help to diagnose sleep disorders such as narcolepsy and sleep apnea by monitoring the brain waves during sleep.

Section 2.4 Our Application for EEG (EEG-Based Robotics System)

Apart from the previous applications of EEG, which mostly focus on medical fields, we have another interesting application that uses it for controlling robotics and giving orders. It can be useful for disabled people to move and control things using only their thoughts [3].

The history of robotics-controlled EEG goes back to the 1970s when the researchers Hans Berger and José Delgado did some experiments. In 1988, Stevo Bozinovski and his colleagues at the University of Belgrade demonstrated the first successful control of a robot using EEG signals [4].

EEG-based robotic systems have the potential to transform a wide range of applications, including:

- Rehabilitation: EEG-controlled robots can provide individuals with motor impairments with new ways to regain lost mobility. For instance, EEG-controlled exoskeletons can assist individuals with paralysis in walking or performing other tasks.
- Prosthetics: EEG-based control can enhance the functionality of prosthetic limbs, allowing users to control their prostheses with their thoughts, making them more intuitive and natural to use.
- Assistive Technologies: EEG-controlled robots can assist individuals with disabilities in performing everyday tasks, such as operating computers, controlling appliances, or navigating their environment.

- Human-Robot Collaboration: EEG-based robotic systems can facilitate seamless collaboration between humans and robots, enabling them to work together more effectively in tasks that require shared decision-making and coordination.

Section 2.5 EEG and AI

AI has made an advance in the field of EEG and BCI where it was to determine the pattern of the signals and classify the thoughts into desired categories. AI works by giving inputs and labeled output and letting the deep learning Algorithm discover the pattern or the equation that led to the labeled outputs.

Nowadays using AI to analyze EEG signals has been the industry's standard [5].

Section 2.6 Related Research

In the field of EEG-based robotics systems, researchers go for two main concepts to achieve their application, Photo-Imagery and Motor-Imagery [6].

- Photo-imagery involving the mental visualization of pictures or scenes is indeed utilized in EEG research to understand and leverage the brain's response to internal, imagined stimuli.

- Motor-imagery is a widely employed technique in EEG-based robotics systems. It revolves around mentally simulating particular movements, like the movement of a hand or foot, without actually executing the physical action [7]. Through the detection and analysis of the associated patterns of brain activity, MI enables the control and interaction between the human brain and robotic devices.

Section 2.7 Limitations of Current Designs or Technology

There are several limitations associated with current designs and technology that we aim to address and overcome.

One limitation is the absence of wireless capabilities in some existing systems. Many current EEG-based robot control systems require wired connections between the EEG headset and the processing unit. This wired setup can restrict the mobility and freedom of movement for the users, limiting the practicality and usability of the system.

Another limitation is the physical size of the devices used in current designs. EEG headsets often consist of multiple electrodes and sensors, resulting in bulkier and less ergonomic designs. This can be particularly challenging for individuals with limited mobility or dexterity. The size and weight of the devices can cause discomfort and hinder the ease of use, especially for long-term applications.

Here are some ways to overcome these limitations that will be used in our system:

- Wireless communication between the EEG headset and the control unit is crucial for enhancing user comfort and enabling a wider range of applications.
- More compact and lightweight devices that prioritize user comfort and convenience.

Section 2.8 Current Systems

Many systems control robots using EEG signals. We will explore the similarities and differences between our system and those existing systems.

1- Brain EEG Signal Processing for Controlling a Robotic Arm

This research presented a new non-invasive system for controlling a robotic arm using Brain EEG signal processing, moving his right hand using the Emotive EPOC headset device.

Similarities between our system and this existing system:

- Filtering the signal to remove noise and unwanted data.
- Both systems use Emotiv EPOC+.

Differences between our system and this existing system:

- Our system focuses on controlling a humanoid robot, while the existing system focuses on controlling a robotic arm.
- Our system should perform three tasks (raise the left hand, raise the right hand, and raise both hands), while the existing system performs three different tasks (close, open arm, and close hand).

2- EEG Based Mobile Robot Control through an Adaptive Brain Robot Interface

This project mentioned a couple of brain-controlled automaton-supported BCIs. BCIs square measure systems that may bypass standard channels of communication (i.e., muscles and thoughts) to produce direct communication and management between the human brain and physical devices.

Similarities between our system and this existing system:

- Support BCI.
- Use a DC-servo motor.
- Translating different patterns of brain activity into commands in real-time.

Differences between our system and this existing system:

- Our system data transfers through Wi-Fi while the existing system data transfers through Bluetooth.
- Our system is reliable and low-cost.

Chapter 3 Requirements Specifications

In this chapter, we present the essential requirements that our project should meet. It is important to ensure that the project meets the needs of the customers.

Section 3.1 User Needs

- The robot should be low-cost.
- The interface between EEG signals and the robot should be wireless.
- The interface solution should use state-of-the-art algorithms.
- The robot should be lightweight.
- The system should be reliable.
- The system should be capable of recognizing EEG signals with high accuracy.
- The robot should be controlled using EEG signals to perform different actions.
- The system should be able to analyze and process the signals in real-time.

Section 3.2 Objective Tree

Figure 3.1 represents an Objective Tree that prioritizes three groups based on customer preferences, with the highest rank indicating the most important group. According to the figure, the most crucial group is 'Operation' (0.48), followed by 'Performance' (0.40), and the least important group is 'Durability' (0.12).

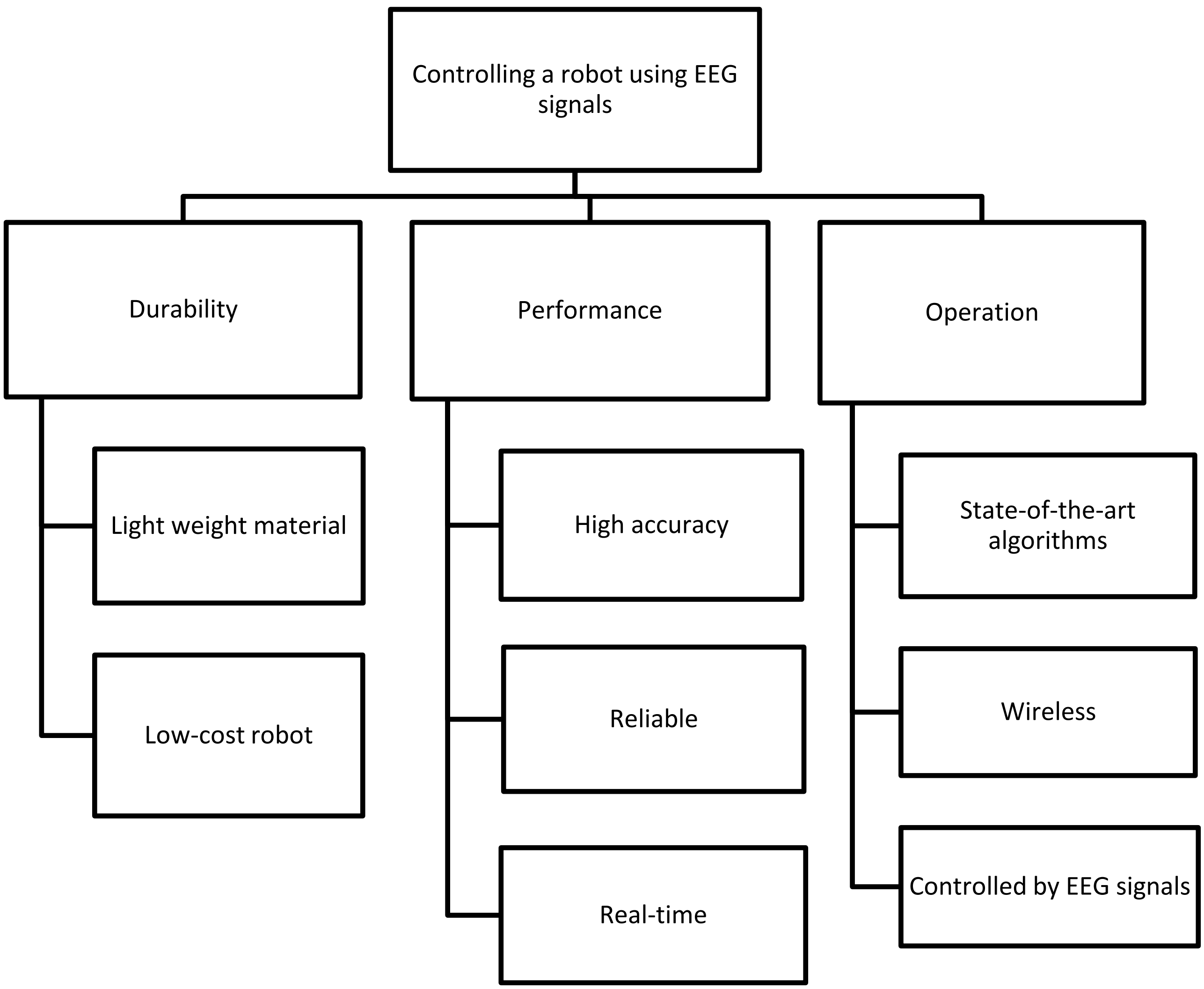


Figure 3.1 Objective Tree

Section 3.3 Requirements Specifications

Presented below are the project's requirements specifications that show the complete set of all system requirements.

Table 3.1 Requirements Specifications

Marketing Requirement	Engineering Requirement	Justification
1	1. Cost of the system should be under 3000$.	It should be doable with available hardware in the market.
1,4	2. Robot should weigh no more than 10 kg.	One person can move the whole system from one place to another.
2	3. System should use wireless technology.	We should be able to send raw data to the robot wirelessly.
5	4. System data rate should not be less than 100Mbs.	We must send a good amount of data reliably to avoid latency.
5,8	5. System latency should be no more than 0.4sec.	Action should be taken as soon as the user selects it.
5,6,7	6. System should have no less than 12 EEG sensor channels.	The more channels the more data streams we have, and we can get more accuracy.
1,2,4	7. System should be deployed on a small board computer.	System should be standalone so we can move it around easily.
3,6	8. System should be able to classify EEG into 3 main motion classes at a minimum.	We have to classify the EEG into 3 motions so the robot can perform them.
7	9. Robot should be able to perform 3 actions minimum.	We have to design the robot to perform the predicted output of the EEG signal.

Marketing Requirements

1. The robot should be low-cost.
2. The interface between EEG signals and the robot should be wireless.
3. The interface solution should use state-of-the-art algorithms.
4. The robot should be lightweight.
5. The system should be reliable.
6. The system should be capable of recognizing EEG signals with high accuracy.
7. The robot should be controlled using EEG signals to perform different actions.
8. The system should be able to analyze and process the signals in real-time.

Chapter 4 The Design

In this chapter, we present various design concepts considered for our project and outline the decisions made to select the optimal design that fulfills the desired requirements.

Section 4.1 Concept Overview

In this section, we present various alternatives for our project in Table 4.1. Our decision-making process involves a careful evaluation of each alternative, identifying and listing their strengths and weaknesses. This analytical approach ensures that we choose the most suitable option, aligning with our project's objectives.

Table 4.1 Concept table

Power Source	Controller Board	Motor type	Connectivity	headset
Lithium Battery	Arduino	Double shaft DC motor	Bluetooth	Emotiv Insight
AC Power	Raspberry Pi	DC Servo Motor	Wi-Fi	Emotiv EPOC+
Solar Power	Jetson nano	DC Motor		
Kinetic Energy		Dynamixel servo		

- Each column is independent of the others.

4.1.1 Power Source

The criteria to choose between the four power sources depends on portability and the chargeability as we can see the best option to choose is the **Lithium battery**. We described it in Table 4.2.

Table 4.2 Power Source (Strengths vs Weaknesses)

Power source	Strengths	Weaknesses
AC power	• High Power Capacity. • Availability.	• Lack of Portability. • Vulnerable to Power Outages.
Kinetic Energy	• Portability. • Renewable and Sustainable.	• difficult to control. • Limited Availability.
Lithium battery	• Rechargeable. • Portability. • Lightweight. • Low initial cost.	• Low energy density. • Limited Capacity.
Solar power	• Renewable. • Availability.	• High initial cost. • Limited Power generation during cloudy Days.

Lithium batteries can power wireless systems effectively. As we are working on a wireless setup for controlling the robot, lithium batteries will provide the necessary power to transmit and receive EEG signals wirelessly without the constraints of a wired power source. This enhances the mobility and freedom of movement of the robot.

4.1.2 Controller Board

Table 4.3 Controller Board (Strengths vs Weaknesses)

Method	Strengths	Weaknesses
Arduino	• Cheap. • Easy to use. • Low power.	• Low processing power. • Limited support for AI applications. • No parallel processing. • Difficult to connect wirelessly. • CPU speed 16MHz.
Raspberry Pi	• Mid-price. • Easier than Jetson Nano. • Higher processing power than Arduino. • Support screen and input devices. • Support camera. • Support parallel processing. • Easy wireless connectivity. • Support AI applications.	• Consume more power than Arduino. • Lower processing power than Jetson Nano. • More complex to use than Arduino.
Jetson Nano	• Most suitable for AI applications. • Have CUDA cores. • Support NVidia libraries. • Support screen and input devices. • Support camera. • Easy wireless connectivity. • Support for parallel processing.	• High power consumption. • Less community support than Raspberry Pi. • High cost. • More complex to use than Arduino and Raspberry Pi.

Since we are focusing on a low-cost system, we went with the most available controller board that supports AI applications and it was **Raspberry Pi** since it has a processing power that is enough to deploy our deep learning algorithm in it. This credit-card-sized computer packs a surprising amount of processing power into its compact design. With a variety of models available, ranging from the entry-level Raspberry Pi Zero to the more powerful Raspberry Pi 5, it caters to a wide range of projects and applications. Raspberry Pi is equipped with a Broadcom system-on-chip (SoC) processor, RAM, USB ports, HDMI output, GPIO pins for interfacing with external hardware, and support for various operating systems. Jetson Nano also can be used as a second option. We described it in Table 4.3.

4.1.3 Motor type

There are many different motors we could choose from. However, our decision is based on the torque range, input voltage, and control of the angles. As a result, we decided to choose the **DC Servo Motor** to use in our project.

4.1.4 Connectivity

Table 4.4 Connectivity of Bluetooth and Wi-Fi (Strengths vs Weaknesses)

Method	Strengths	Weaknesses
Bluetooth	• Cheap. • Low power consumption.	• Limited range. • Low data rate.
Wi-Fi	• High range. • High data rate. • Built-in for chosen controller board like Raspberry Pi.	• Higher power consumption.

We want our data to be transmitted nearly real-time, so we went for the most efficient solution which is **Wi-Fi**. It is easy to implement since we are using Raspberry Pi or Jetson Nano. We described it in Table 4.4.

4.1.5 Headset

Emotiv EPOC+ [8] and Emotive Insight 2.0 are both electroencephalogram (EEG) devices. These devices are designed to monitor and record electrical activity produced by the brain's neurons, commonly known as brainwaves. Table 4.5 shows the strengths and weaknesses of the two EEG devices.

Table 4.5 Headset (Strengths vs Weaknesses), derived from [9]

Method	Strengths	Weaknesses
Emotiv Insight 2.0	• Shorter set up time (1-2 min). • longer battery life (up to 20 hours).	• Only 5 recording sensors. • Good but lower data quality. • Fixed headband
<u>Emotiv EPOC+</u>	• 14 recording sensors. • High data quality. • Rotating headband.	• Longer set up time (3-5 min). • Shorter Battery life (up to 12 hours).

Based on the described strengths and weaknesses, we chose **Emotive EPOC+**.

The **Emotiv EPOC+** (a sample is shown in Figure 4.1) is a 14-channel wireless headset designed for scalable and contextual human brain research and provides access to professional-grade brain data with a quick and easy-to-use design. It offers several key features that make it a valuable tool for brain research and can be used for a wide range of applications, including brain-computer interfaces (BCIs), education, and gaming and entertainment.

Figure 4.1 Emotiv EPOC+ VS Emotiv Insight 2.0 [8]

Section 4.2 Design Specifications

4.2.1 Design Level-0

At this level, Figure 4.2 shows the block diagram that includes all inputs and outputs of the overall system. Table 4.6 describes the level-0 design.

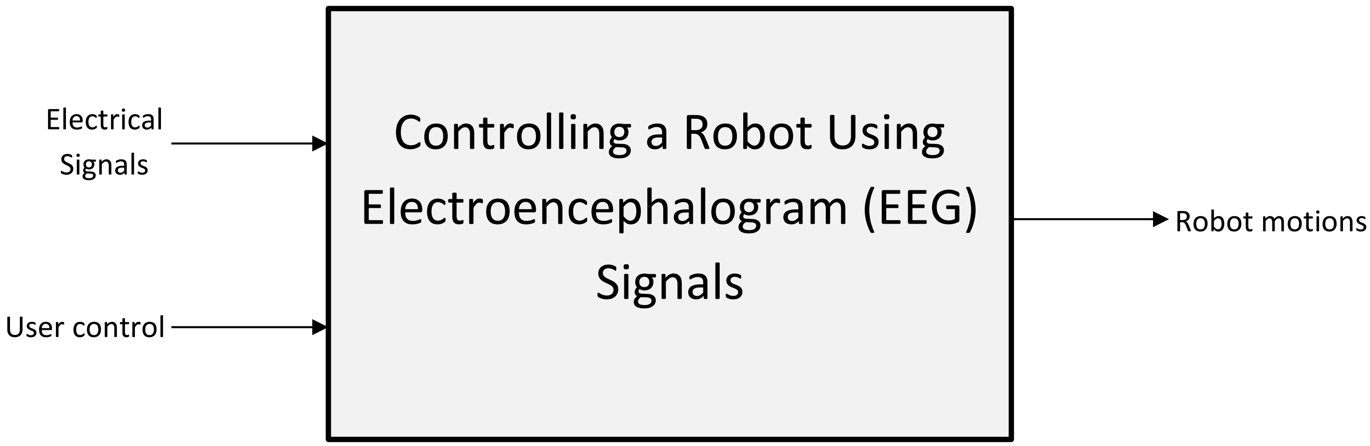


Figure 4.2 Level-0 design

Table 4.6 Level-0 table

Module	Controlling a Robot Using Electroencephalogram (EEG) Signals
Inputs	-EEG Signals -User control: on/off
Outputs	-Robot motions
Functionality	The system captures and interprets the electrical signals generated by the user's brain and analyzes them to control the robot's motions.

4.2.2 Design Level-1

In this level, Figure 4.3 shows the major modules inside the system and the interconnect between them. Table 4.7 describes the level-1 design.

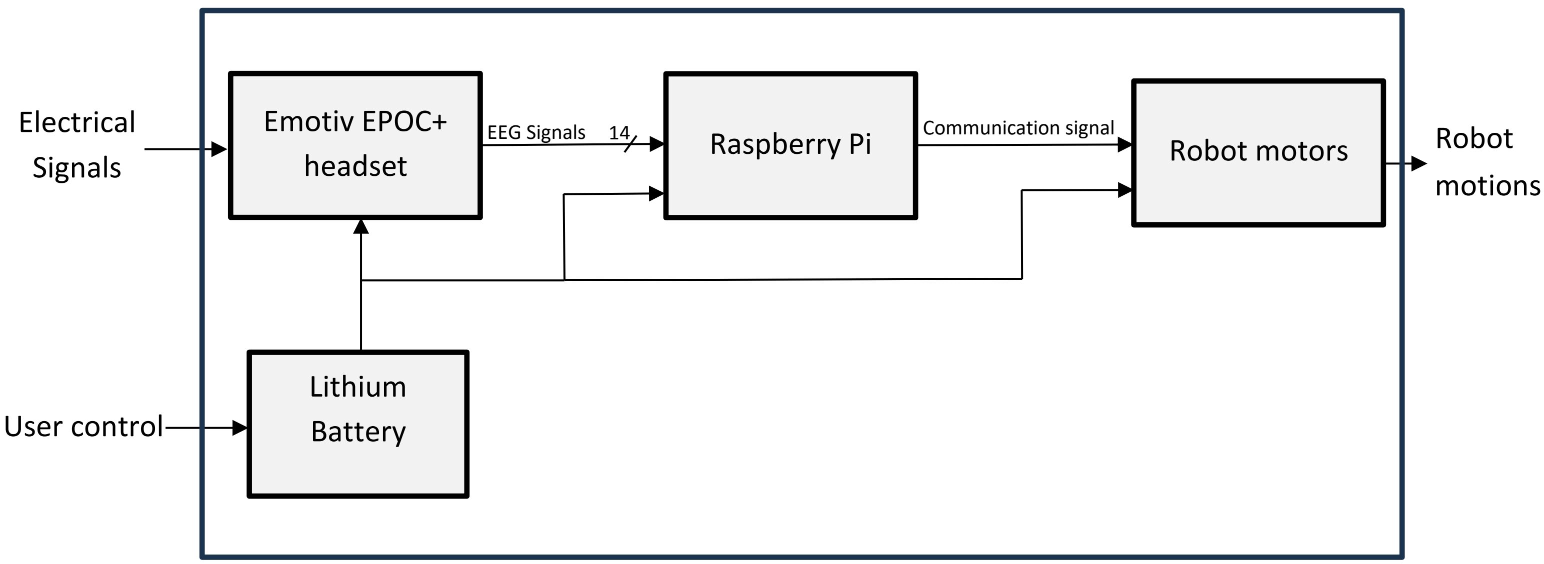


Figure 4.3 Level-1 design

Table 4.7 Level-1 Lithium Battery component

Module	Lithium Battery (a sample is shown in Figure 4.4)
Inputs	-User control: on/off
Outputs	-DC Voltage: 5V for power
Functionality	The lithium battery serves as the lifeblood of the system, providing power to essential components such as Emotiv EPOC+, Raspberry Pi, and the motors that drive the robot.

Figure 4.4 illustrates the lithium battery component that we will use in our system.

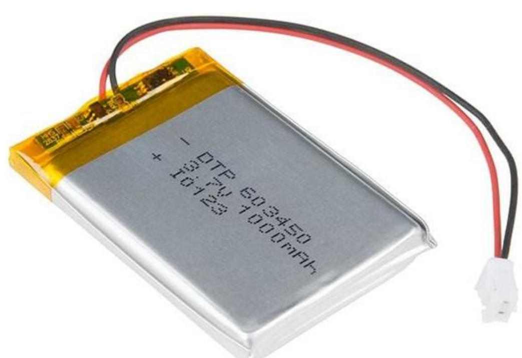


Figure 4.4 Lithium Battery

Table 4.8 Level-1 Emotiv EPOC+ component

Module	Emotiv EPOC+ (Figure 4.5)
Inputs	-Electrical Signals -DC Voltage: 5V for power
Outputs	-EEG Signals
Functionality	An EEG device is designed to capture and interpret the electrical signals generated by the user’s brain. These signals are detected by the electrode sensors on the headset, which are in contact with the user's scalp. The EEG signals collected by the Emotiv EPOC+ (Figure 4.5) include information about different brainwave frequencies, such as alpha, beta, gamma, delta, and theta waves. These signals are then transmitted to a compatible device for processing and analysis.

Figure 4.5 illustrates the Emotiv EPOC+ headset that we will use in our system. Table 4.8 describes Level-1 Emotiv EPOC+ component.

Figure 4.5 Emotiv EPOC+ headset

Table 4.9 Level-1 Raspberry Pi component

Module	Raspberry Pi 5 (Figure 4.6)
Inputs	-EEG Signals -DC Voltage: 5V for power
Outputs	-Communication Signals
Functionality	The Raspberry Pi 5 works by receiving EEG signals as input via Wi-Fi from the Emotiv EPOC+. The signals will be input into a trained deep learning model to classify motion patterns and transmit the identified classes as a communication signal.

Figure 4.6 illustrates the Raspberry Pi 5 component that we will use in our system. Table 4.9 describes Level-1 Raspberry Pi component.

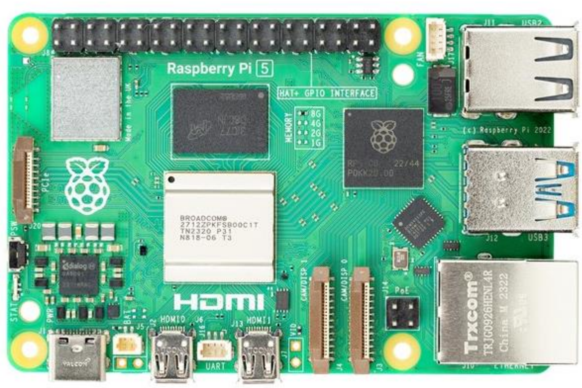


Figure 4.6 Raspberry Pi 5

Table 4.10 Level-1 Robot motors component

Module	Robot motors (a sample is shown in Figure 4.7)
Inputs	-Communication Signals -DC Voltage: 5V for power
Outputs	-Robot motions
Functionality	The robot receives communication signals from the Raspberry Pi. Based on the signal information, the motors work to execute the desired command, whether it be raising the left hand, raising the right hand, or raising both hands.

Figure 4.7 illustrates the Servo motor, mg996r that we will use in our system. Table 4.10 describes Level-1 Robot motor's component.

Figure 4.7 Servo motor, mg996r

Section 4.3 Preliminary Work

Based on our design decisions, we used the Emotiv Epoc+ for the data acquisition, which is an EEG device designed to capture and interpret the electrical signals generated by the user's brain. These signals are detected by 14 electrode sensors that are placed on the human scalp (Figure 4.9) during Motor Imagery tasks, collected EEG signals include information about different brainwave frequencies, but the effective MI bands are only beta and alpha waves (Figure 4.8). These signals (Figure 4.10) are then transmitted via Wi-Fi to the Raspberry Pi controller board that is placed in the robot. The EEG signals will be input as raw signal values (because raw signal values give more accuracy than other input formulations) into deep learning with an attention mechanism (that enables models to selectively focus on relevant parts of input data) for the MI classification. The output will be the result of this classification, based on that, the Raspberry Pi controls the robot to perform one of the three commands: raise the left hand, raise the right hand, and raise both hands. The robot receives communication signals from the Raspberry Pi. Based on that signal, the motors work to execute the desired command.

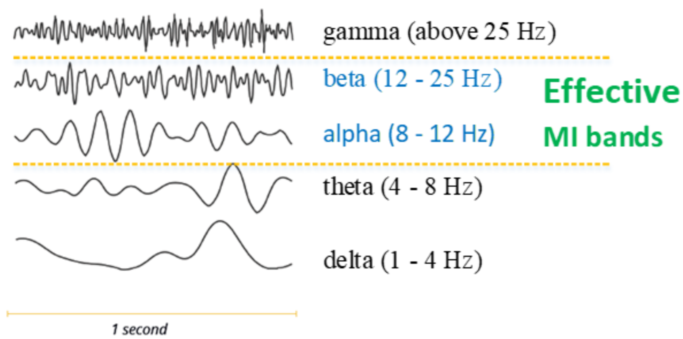


Figure 4.8 Effective MI bands

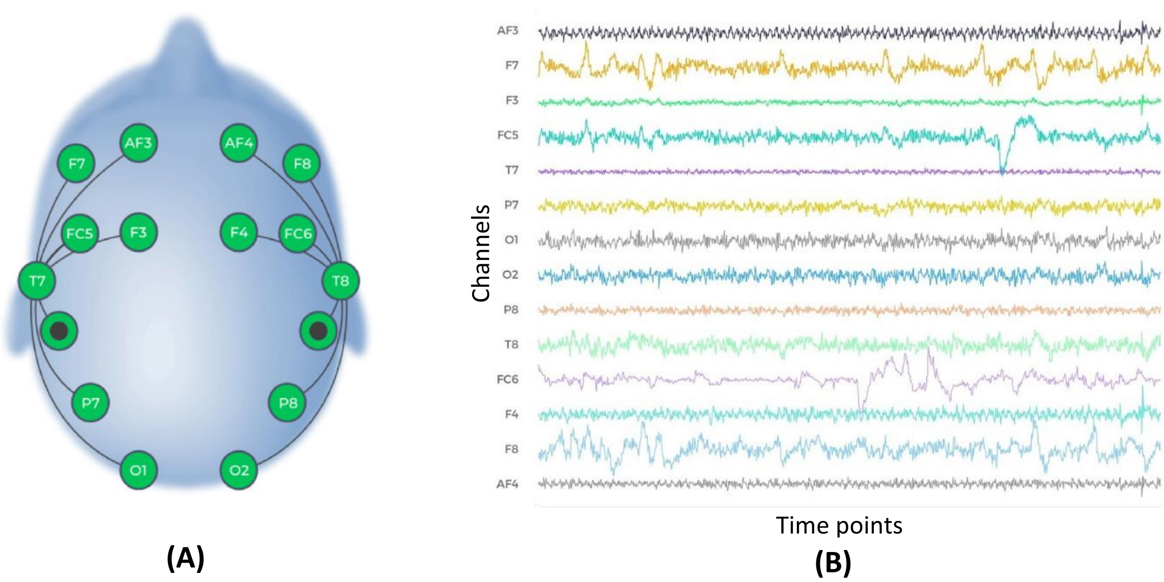


Figure 4.9: (A) Sensor locations. (B) Recorded MI-EEG.

Some MI datasets are public, but we will use the proposed MI dataset; however, first, we will train the model using the BCI-C IV-2a benchmark dataset. The proposed MI dataset will consist of three commands: raise the left hand, raise the right hand, and raise both hands, and will be recorded using 14 EEG electrodes. Data validation depends on the accuracy of each subject, we will classify the accuracy using the ATCNet model. However, we cannot rely only on the accuracy to determine if our data is valid, we will calculate the confusion matrix. A confusion matrix (Figure 4.11) will be used to see how the model predicts the dataset and evaluates them based on four criteria:

True Positive (TP): The predicted label is positive, and it is in the correct label.

False Positive (FP): The predicted label is positive, but in reality the label is negative.

True Negative (TN): The predicted label is negative, and it is in the correct label.

False Negative (FN): The predicted label is negative, but in reality the label is positive.

Based on the results, for each class in the confusion matrix, if the true positive rate is higher than the false positive rate for all other classes, the data is considered “Valid”; otherwise, it is considered “Invalid”.

The performance is evaluated using multiple formulas, such as:

Precision: It is a measure of correctness that is achieved in true prediction.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall: It is a measure of actual observations which are predicted correctly**.**

$$\text{Recall} = \frac{TP}{TP + FN}$$

F1 Score: It combines the precision and recall scores of a model.

$$F1\ Score = 2 \times \frac{recall \times precision}{recall + precision}$$

Accuracy: It is a measure of how often the classifier makes the correct prediction. It is the ratio between the number of correct predictions and the total number of predictions.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

		Predicted	
		Negative **(N)** -	Positive **(P)** +
Actual	Negative -	True Negative **(TN)**	**False Positive (FP) Type I Error**
	Positive +	**False Negative (FN) Type II Error**	True Positive **(TP)**

Figure 4.10 Confusion matrix

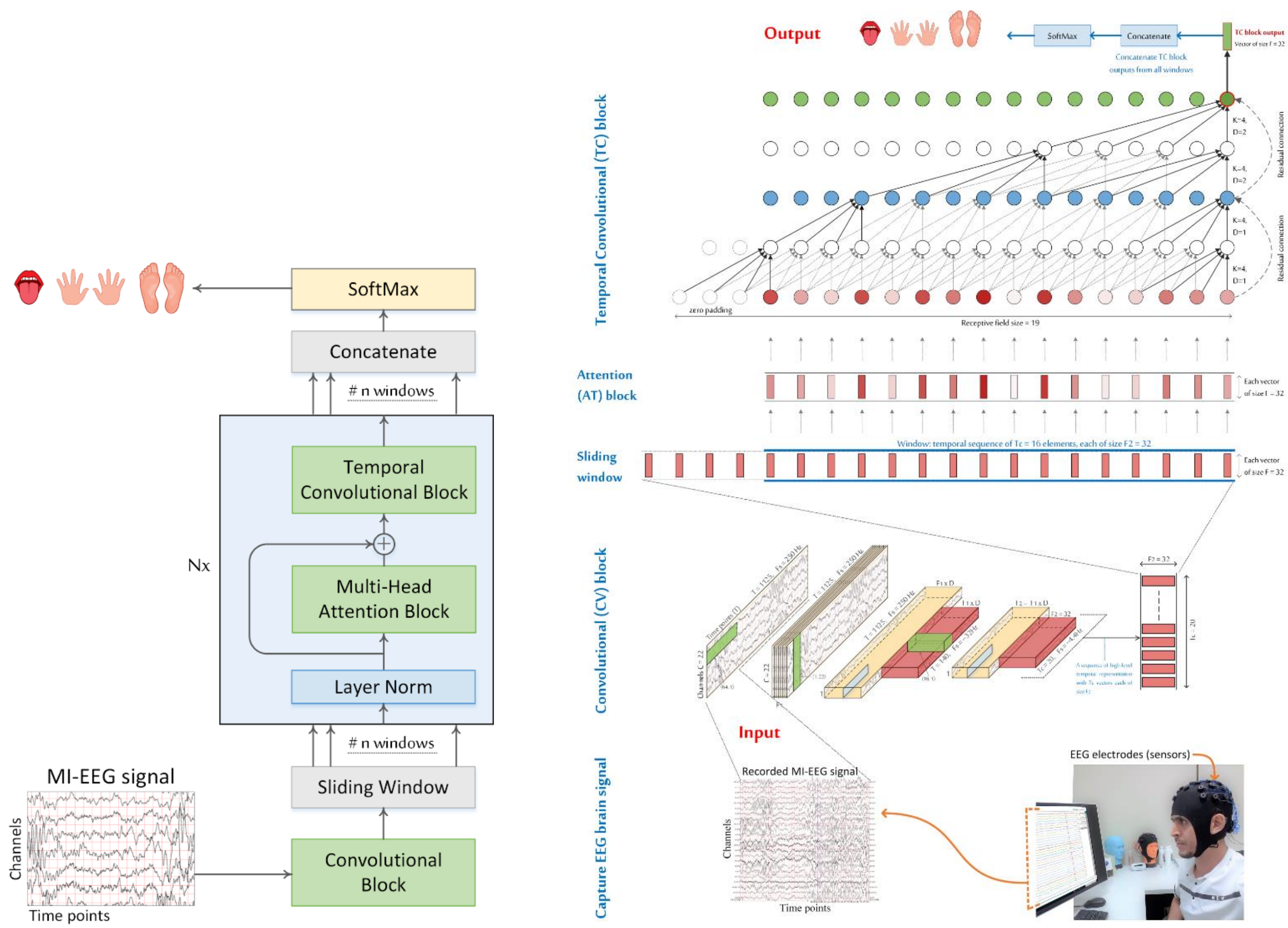
In our design, we will use deep learning with an attention mechanism for the MI classification, which is the Attention Temporal Convolutional Network (ATCNet) [13][14]. ATCNet is a model that can deal with MI-EEG data. This model consists of three main blocks:

1- Convolutional (CV) block: It has different convolutional layers that can deal with MI data and extract high-level features.

2- Attention (AT) block: This block highlights the most important information in the temporal sequence using a multi-head self-attention (MSA).

3- Temporal convolutional (TC) block: extracts high-level temporal features from the highlighted information using a temporal convolutional layer.

ATCNet model also utilizes the convolutional-based sliding window to augment MI data and boost the performance of MI classification efficiently. It works in parallel blocks, each block receives one window from the sliding window. Then, the outputs of each block are concatenated and fit into a SoftMax layer to produce the classification output.



(A) (B)

Figure 4.11: (A) ATCNet model [13]. (B) Detailed ATCNet model [13].

We also collaborated with a YouTube content creator to produce an entertaining video explaining the concept of EEG. You can watch the video at the following link: https://youtu.be/MpciypwiX2k?si=mRSx5eI_nRRE1vZH

Section 4.4 Flowchart

In this section, we present a flowchart (Figure 4.12) illustrating how the system works.

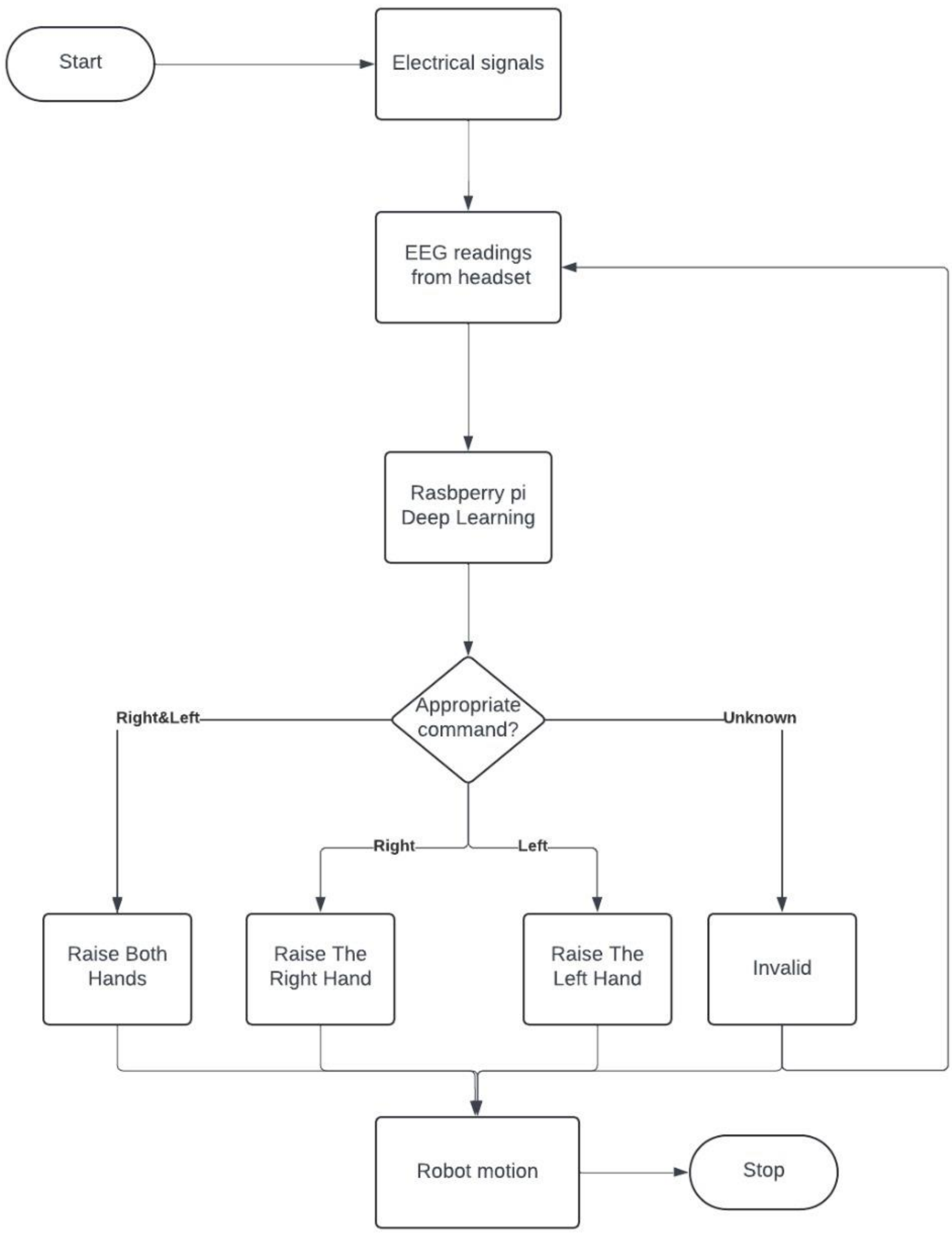


Figure 4.12 System Flowchart

Chapter 5 Implementation and Verification

In the Implementation and Verification phase, we detail how we turned our conceptual plans into tangible outcomes and ensured their alignment with our objectives. Our aim is to provide clear steps for the project's execution while ensuring verification standards. We will also address the challenges we faced on our journey and how we encountered them along the way, giving an overview of our journey.

Section 5.1 Prototype Implementation

In this section, we will show how we implemented each module in the system and describe these modules. We will also include the implementation challenges and how they were handled on our journey.

5.1.1 Raspberry Pi Installation

Regarding the Raspberry Pi 5, our selection was based on its affordability, high processing power featuring GPU capabilities, and the great community support it offers.

We opted for the installation of Raspberry Pi OS, the official operating system provided by the Raspberry Pi website. Subsequently, we installed Miniconda Manager to manage our AI libraries and their dependencies.

We used built-in optimization techniques from TensorFlow, such as quantization, to convert our models into smaller ones, enabling deployment on the Raspberry Pi.

After setting up the model, we can designate it as the inference engine, which will receive a stream of data and output results in real-time. The accuracy is usually affected by these techniques, but as we observed, we have not noticed much decrease in accuracy.

5.1.2 Robot

Initially, we attempted to utilize the humanoid A cm530 robot (Fig. 5.1) for our project. However, when we tried to connect and test it, we encountered issues. After contacting the company, we learned that the robot's outdated version was no longer supported, which explained the technical limitations we faced. This lack of support meant we could not access necessary updates and advancements, making the robot incompatible with our project.

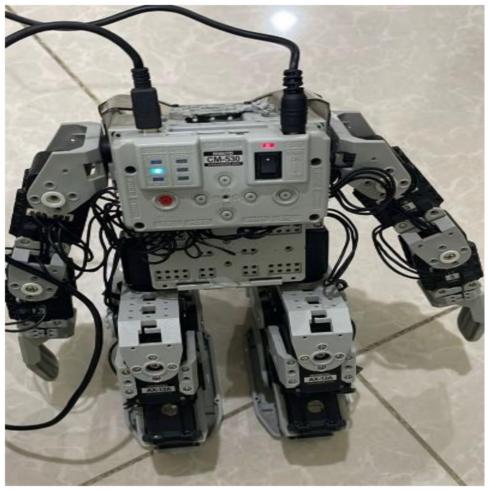
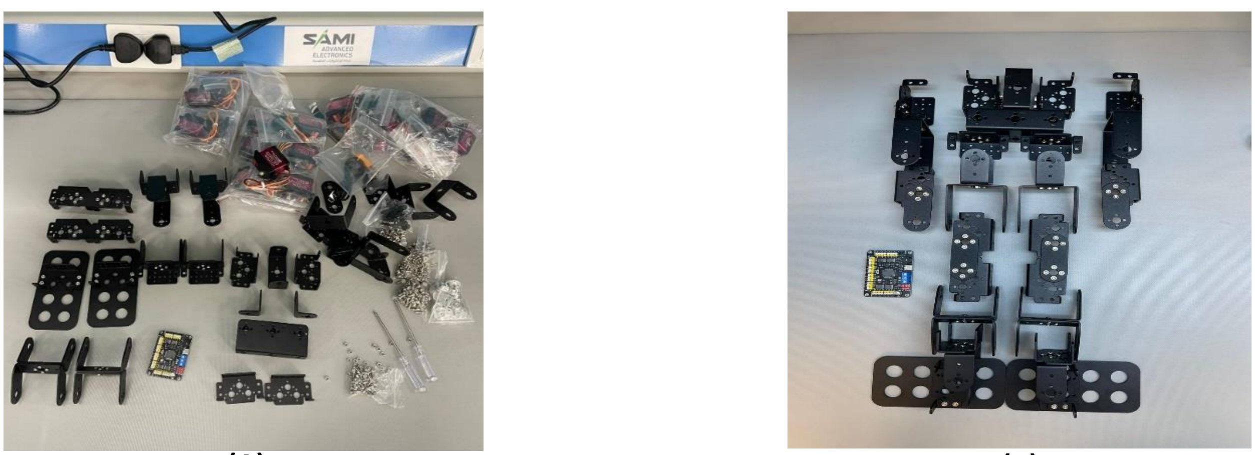


Figure 5.1 Humanoid A CM530 robot

Consequently, we made the decision to acquire a new robot that could better fulfill our needs. We purchased the 17B Dof robot with RTtobot 32 servo Controller VER 3.4, specifically chosen for its improved functionality and compatibility with our project's objectives (Fig. 5.2). However, upon receiving the robot, we encountered an immediate challenge: the parts and servos were not properly fitted.



(A) **(B)**

Figure 5.2: (A) 17b Dof robot before fitting the parts. (B) 17b Dof robot while fitting the parts.

This required us to carefully and methodically fit all the necessary parts and servos together. So, we carefully fitted all the necessary parts and servos (Fig. 5.3). This process involved ensuring proper integration and alignment to guarantee optimal functionality within the system.

After correctly assembling the parts, we turned on the robot and initiated various commands and actions to ensure that the servos functioned correctly. We closely monitored the robot's behavior, paying particular attention to the smoothness, precision, and synchronization of its movements.

Figure 5.3 17b Dof robot after fitting the parts

We were pleased to observe that the servos were indeed functioning correctly. The robot exhibited fluid and responsive movements, accurately performing the desired actions. This outcome served as confirmation of the successful integration and proper functioning of the fitted parts and servos. Additionally, we were able to make the robot stand up and maintain balance. However, we encountered difficulties in making the robot move forward because it was challenging to balance on one foot while taking a step. Therefore, we preferred to make the robot raise both hands instead.

After successfully fitting the parts and confirming the proper functioning of the 17B Dof robot, we connected it with the Raspberry Pi. This connection allowed us to leverage the capabilities of the Raspberry Pi and integrate it into our project.

Since our deep learning is implemented on the Raspberry Pi, we developed a code that takes the output classification of the deep learning model as an input, and perform the task that is assigned with this classification with the help of the Raspberry Pi.

5.1.3 Dataset Recording and Analyzing

Due to the unavailability of an existing dataset online, we had to record and create our own (Figure 5.4). Instead of manually recording it, we wrote a program that is capable of automatically recording EEG data corresponding to four distinct action classes.

Figure 5.4 Recording Dataset

The program operates by guiding users through each of the four classes, with a brief relaxation interval of 2 seconds separating each iteration. During each 4-second interval, participants are prompted to imagine performing the action associated with the respective class.

To enhance user experience and optimize data collection, the program randomizes the presentation of classes and provides advanced cues regarding the upcoming class during relaxation phases. This is achieved through the display of transparent images hinting at the forthcoming action, transitioning to clearer images when the class commences (Figure 5.5). Additionally, the program prevents consecutive repetitions of the same class, ensuring a balanced distribution of recordings across all classes.

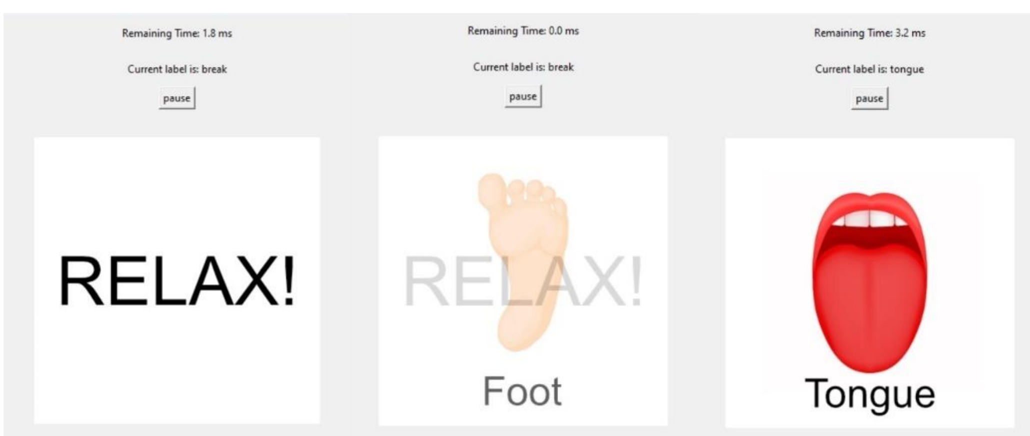
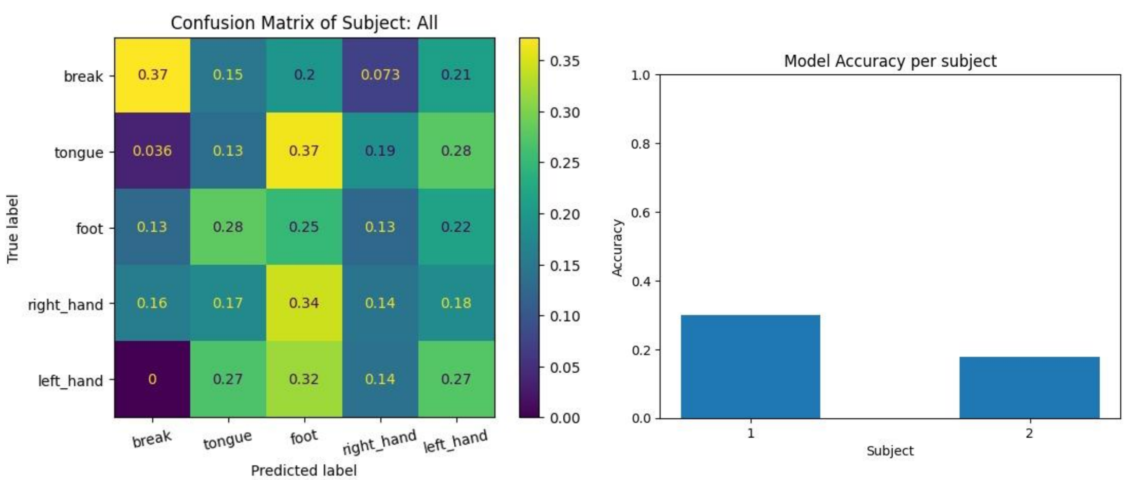


Figure 5.5 Dataset recording program

By implementing this automated system, we significantly minimized the time and effort required for data collection compared to manual methods. This was a crucial step in gathering the data we needed to train our model.

Due to the novelty of the technology, we encountered significant limitations. One major challenge was the technology's inability to achieve high accuracy with subject-independent methods. Consequently, we decided to adopt a subject-dependent approach, which means that each subject’s data has to perform training and testing on it individually to output a specifically trained model for that subject. Training and testing were performed on the Google Colab platform. For the following paragraphs, we will show only some of the best results for each attempt.

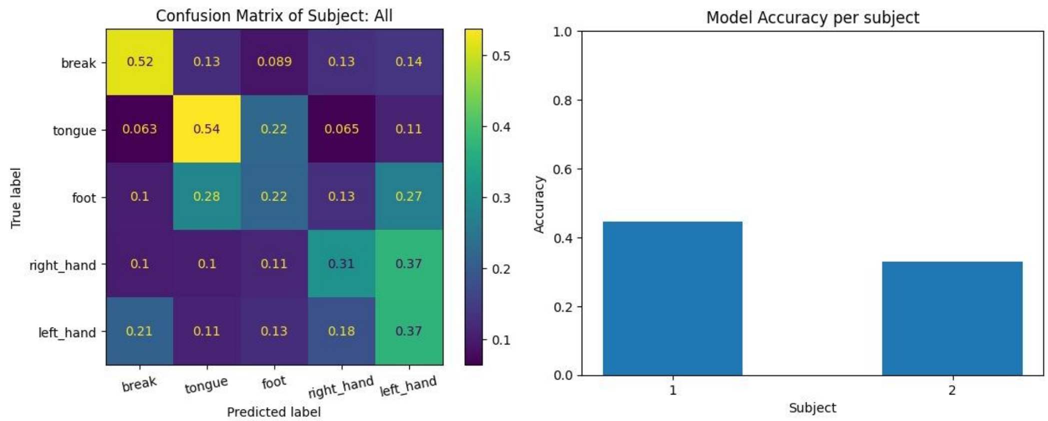
Initially, we began recording data at a sample rate of 125 with a vector input of 500 channels every 4 seconds. Each of the three participants contributed around 300 samples, evenly divided among the classes: Relax, Move Forward, Raise The Right Hand, Raise The Left Hand, and Raise Both Hands, with 60 samples for each class. However, the resulting accuracy after training and testing was 30% for subject 1, and 18% for subject 2, falling short of our expectations. The confusion matrix and the accuracies are shown in Fig. 5.6.



(A) **(B)**

Figure 5.6: (A) Confusion matrix and (B) accuracy, shows the first attempt results

In an effort to improve accuracy, we increased the sample rate to 280, resulting in a corresponding increase in the vector input to 1125 channels every 4 seconds. We recorded approximately 500 samples, allocating 100 samples to each class. Despite these efforts, the accuracy remained 43% for subject 1, and 32% for subject 2, failing to meet our desired expectations (Figure 5.7).



(A) **(B)**

Figure 5.7: (A) Confusion matrix and (B) accuracy, shows the second attempt results

To address this challenge, we conducted binary classification experiments, focusing on two classes at a time, which increased accuracy (Figure 5.8). However, we consistently found that including a Relax class decreased the accuracy. Based on this insight, we decided to streamline the number of classes to three (excluding the Relax class) instead of four. The new classes are: Raise The Right Hand, Raise The Left Hand, or Raise Both Hands.

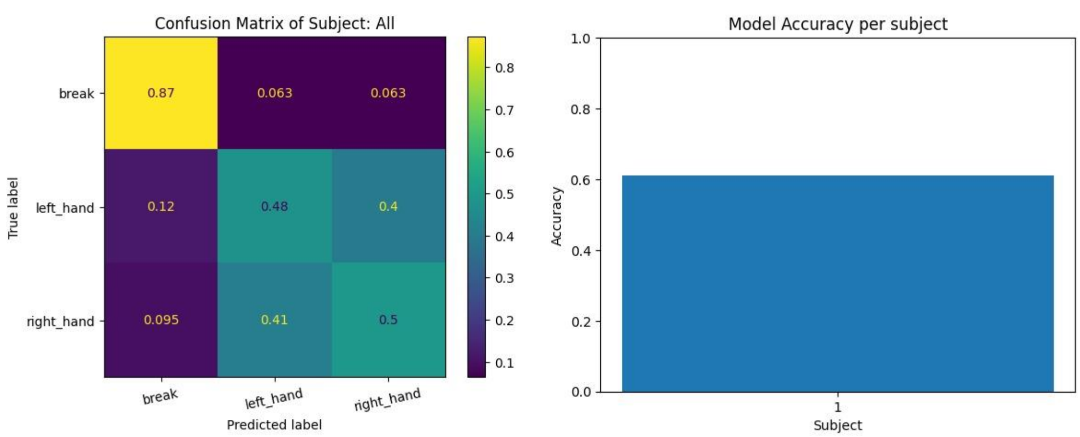
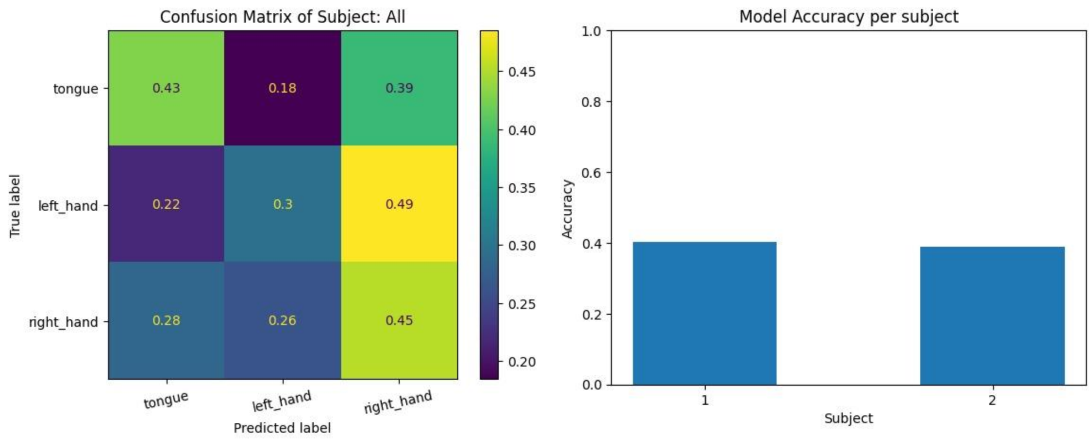


Figure 5.8: (A) Confusion matrix and (B) accuracy, shows the third attempt results

Following these observations, we used 300 samples, evenly divided among the classes: Raise The Right Hand, Raise The Left Hand, and Raise Both Hands, with 100 samples for each class. This resulted in an accuracy of 40% for subject 1, and 39% for subject 2.



(A) **(B)**

Figure 5.9: (A) Confusion matrix and (B) accuracy, shows the fourth attempt results

5.1.3 Fitting The System Components

Once each component of the system was implemented, we proceeded to integrate these components, resulting in the final prototype illustrated in Figure 5.10.

Figure 5.10 Integrated system

Section 5.2 Design Verification

In this section, we will present the results of testing the system's prototype to verify the design requirements. We conducted two types of tests: Unit Test (Table 5.1) and Acceptance Test (Table 5.2).

5.2.1 Unit Test

Table 5.1 Unit Test #1, Servo Motors

Test Writer: Saad Aldobaian							
Test Case Name:		Servo Motors – *Test #1*	**Test ID #:**			CRUES-01	
Description:		Before connecting the entire system together, it is crucial to ensure that the servos in the robot are functioning correctly.	**Type:**			☐ white box ☑ black box	
Tester Information							
Name of Tester:		Saad Aldobaian - Abdullah Almingash	**Date:**			3/4/2024	
Hardware Version:		Controlling a Robot Using (EEG) Signals 1.0	**Time:**			11:11 AM	
Setup:		Ensure that the robot is fully fitted with all its components properly assembled. Additionally, make sure that the robot is connected to the battery for power and the controller for control and communication.					
Step	**Action / Variable**	**Expected Results**	**Pass**	**Fail**	**N/A**	**Comments**	
1	Initialize the servo software library.	The servo library should be successfully initialized without any errors or warnings.	✓				
2	Try to run the servos in the right hand.	the servos in the right hand move correctly and accurately.	✓				

3	Try to run the servos in the left hand.	the servos in the left hand move correctly and accurately.	✓			
4	Try to run all the servos in the robot.	All the servos move correctly and accurately.	✓			
	Overall test result:		✓			

5.2.2 Acceptance Test

Table 5.2 Acceptance Test #1, Robot Motions

Test Writer: Abdullah Almingash						
Test Case Name:		Robot Motions – *Test #1*	**Test ID #:**			CRUES-01
Description:		Checks the engineering requirement: Robot should be able to perform 3 actions	**Type:**			☐ white box ☑ black box
Tester Information						
Name of Tester:		Abdullah Almingash – Abdelrahman Fatouh	**Date:**			4/5 /2024
Hardware Version:		Controlling a Robot Using (EEG) Signals 1.0	**Time:**			9:48 AM
Setup:		The robot should be fully assembled, the Raspberry Pi connected successfully with the robot's servo controller, and the battery should be fully charged.				
Step	**Action / Variable**	**Expected Results**	**Pass**	**Fail**	**N/A**	**Comments**
1	Write a program to test if the robot can perform the desired actions.	The program should be executed without any errors.	✓			

2	Execute the "Raise The Right Hand" command and verify if the robot raises its right hand and returns it to its initial position.	Upon execution, the robot should raise its right hand and then return it to its initial position.	✓			
3	Execute the "Raise The Left Hand" command and verify if the robot raises its right hand and returns it to its initial position.	Upon execution, the robot should raise its left hand and then return it to its initial position.	✓			
4	Execute the "Raise Both Hands" command and verify if the robot raises both hands and returns them to its initial position.	Upon execution, the robot should raise both hands and return them to their initial position.	✓			
	Overall test result:		✓			

References

[1] *Disability World Health Organization*. Available at: https://www.who.int/news-room/fact-sheets/detail/disability-and-health.

[2] *Brain EEG signal processing for controlling a robotic arm - Researchgate*. Available at: https://www.researchgate.net/publication/261481313_Brain_EEG_signal_processing_for_controlling_a_robotic_arm

[3] Shyaa, N.S. *Electroencephalography (EEG) based mobile robot control through an Adaptive Brain Robot Interface, American Scientific Research Journal for Engineering, Technology, and Sciences*. Available at: https://asrjetsjournal.org/index.php/American_Scientific_Journal/article/view/3489

[4] Chang, Y. et al. Research on two-class and four-class action recognition based on EEG signals, Mathematical Biosciences and Engineering. Available at: https://www.aimspress.com/article/doi/10.3934/mbe.2023455.

[5] Tariq, M., Trivailo, P.M. and Simic, M. (2018) EEG-based BCI control schemes for lower limb assistive-robots, Frontiers. Available at: https://www.frontiersin.org/articles/10.3389/fnhum.2018.00312/.

[6] Gopura, R.A.R.C. et al. (2022) Editorial: EMG/EEG signals-based control of assistive and Rehabilitation Robots, Frontiers. Available at: https://www.frontiersin.org/articles/10.3389/fnbot.2022.840321/full.

[7] Yazeed K. Musallam a *et al.* (2021) *Electroencephalography-based motor imagery classification using temporal convolutional network fusion*, *Biomedical Signal Processing and Control*. Available at: https://www.sciencedirect.com/science/article/abs/pii/S1746809421004237.

[8] "EPOC+ - 14 Channel EEG," *EMOTIV*. https://www.emotiv.com/epoc/

[9] "Headset Comparison Chart," *EMOTIV*. https://www.emotiv.com/comparison/

[10] *Emotiv EPOC+ 14 Channel Mobile brainwear* (2023) *EMOTIV*. Available at: https://www.emotiv.com/product/emotiv-epoc-x-14-channel-mobile-brainwear/

[11] *Emotiv+ license* (2023) *EMOTIV*. Available at: https://www.emotiv.com/emotivpro/

[12] *"CanaKit Raspberry Pi 5 Starter Kit - Turbine Black,"* *CanaKit*. Available at: https://www.canakit.com/canakit-raspberry-pi-5-starter-kit-turbine-black.html

[13] H. Altaheri, G. Muhammad, and M. Alsulaiman, “Dynamic convolution with multilevel attention for EEG-based motor imagery decoding,” IEEE Internet of Things Journal, Vol. 10, No. 21, pp. 18579-18588, November 2023. DOI: 10.1109/JIOT.2023.3281911

[14] H. Altaheri, G. Muhammad, and M. Alsulaiman, “Physics-Informed Attention temporal convolutional network for EEG-based motor imagery classification,” IEEE Transactions on Industrial Informatics, vol. 19, no. 2, pp. 2249-2258, February 2023. DOI: 10.1109/TII.2022.3197419

Appendix A The Project Plan

Section A.1 Work Breakdown Structure

Table A.1 Work Breakdown Structure

Controlling a robot using (EEG) signals						
ID	Activity	Description	Deliverable/ Checkpoints	Duration (days)	People	Predecessors
1	**The Problem Statement**	Issue identification and clarification		7		
1.1	Need Statement	Justify project necessity.	Acknowledgment of limitations	4	• Saad	
1.2	Objective statement	Proposed solution overview	Describe desired imperative	3	•Abdelrahman	1.1
2	**Research Survey**	Comprehensive technology review	Limitations identification	6	• Abdullah •Abdelrahman	
3	**Requirements Specification**			7		
3.1	User need	Pinpoint client demands	Customer satisfaction	2	• Abdullah	
3.2	Requirement Specification	Technical requirements compilation	Linking with user needs	4	• Abdelrahman • Saad	3.1
4	**The Design**			11		
4.1	Concept Overview	Clarifying the proposed products	Choose the appropriate products	6	• Saad • Abdullah • Abdelrahman	3.2
4.2	Design Specifications	Detailed system design explanation	Enhance project clarity.	5	• Abdullah • Saad	4.1
5	**Implementation and verification**	Ensure precise design realization		55		
5.1	Prototype Implementation			30	• Saad • Abdullah • Abdelrahman	
5.2	Desing Verification			25	• Saad • Abdullah • Abdelrahman	5.1

Section A.2 Gantt Chart

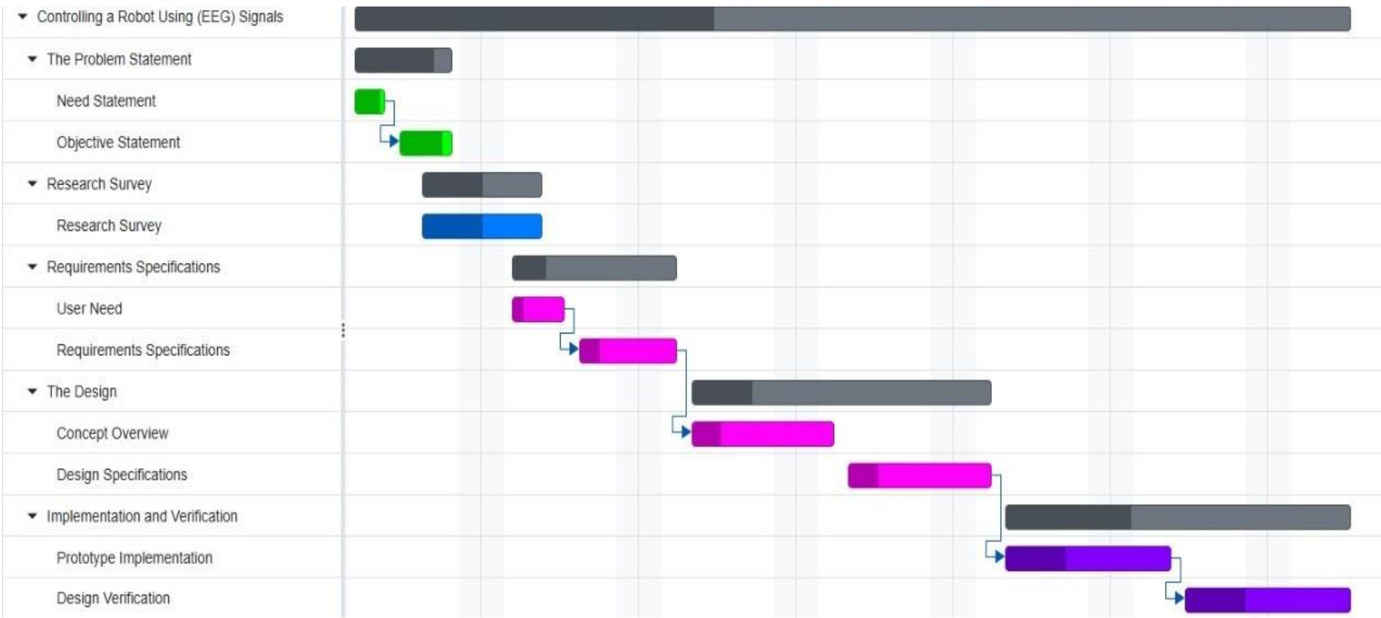


Figure A.1 Gantt Chart

Section A.3 Cost Estimation

Table A.2 Cost Estimation

Equipment:	Amount:	Cost:
Emotiv EPOC+	1	3,748 SAR
Emotive software subscription	1	5,400 SAR
17B Dof robot	1	563 SAR
32-way steering gear control	1	193 SAR
Micro servo (MG995)	3	36 SAR
CanaKit Raspberry Pi 5 8GB Starter Kit	1	600 SAR
TOTAL COST		**10,540 SAR**

Appendix B Program code

Since the codes are too long, we will provide the GitHub link and the QR code of the program code for each one.

ATCNet model for EEG-based motor imagery classification: https://github.com/abdo20050/EEG-ATCNet

EEG Dataset Collector program: https://github.com/abdo20050/EEG_dataset_collector