EEG CONTROLLED ROBOTIC ARM. [Bachelor Thesis].

Thesis · June 2021

DOI: 10.13140/RG.2.2.29975.70566

6 authors, including:


Ahmad Maassarani
Bahçeşehir University
2 PUBLICATIONS **0** CITATIONS
SEE PROFILE


Irmak Akoglu
Bahçeşehir University
1 PUBLICATION **0** CITATIONS
SEE PROFILE


Issa Radwan Hoballah
Bahçeşehir University
1 PUBLICATION **0** CITATIONS
SEE PROFILE


Nadim Absi
Bahçeşehir University
1 PUBLICATION **0** CITATIONS
SEE PROFILE

T.C.
BAHÇEŞEHİR UNIVERSITY

FACULTY OF ENGINEERING AND NATURAL SCIENCES

CAPSTONE FINAL REPORT

PROJECT 1229: EEG CONTROLLED ROBOTIC ARM

Ahmad Maassarani

Irmak Akoğlu

Issa Hoballah

Nadim Absi

Yara Corky

Yusuf S. Kurt

Advisors:

Assoc. Prof. Armağan Fatih Karamanlı, Mechatronics Engineering

Assist. Prof. Hakan Solmaz, Biomedical Engineering

ISTANBUL, June 2021

STUDENT DECLARATION

By submitting this report, as partial fulfillment of the requirements of the Capstone course, the students promise on penalty of failure of the course that.

- they have given credit to and declared (by citation), any work that is not their own (e.g., parts of the report that is copied/pasted from the Internet, design or construction performed by another person, etc.).
- they have not received unpermitted aid for the project design, construction, report or presentation.
- they have not falsely assigned credit for work to another student in the group, and not take credit for work done by another student in the group.

ABSTRACT

Motor function loss is a crucial problem which decreases the quality of life drastically for the people who suffer from it. In this project, an alternative solution is proposed to improve the level of comfort in daily tasks for the people with upper limb motor function loss. The solution uses EEG signals collected from the brain activity of an individual to control the movements of an assistive robotic arm. Movement related mental commands and facial expressions are trained in Emotiv BCI. Emotiv EPOC+ headset captured the EEG signals of the individual, and they were analyzed in the developer tool of Emotiv named Cortex. Interface between robotic arm and BCI is made with Cortex, Python and Arduino programming. Robotic arm with 4-DOF is built up to imitate the natural-arm-like movements. An Arduino microcontroller is utilized to process the commands received through the interface.

Keywords: Motor function loss, EEG analysis, robotic arm, BCI, EEG controlled robotic arm

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

3D	Three Dimensional
ABS	Acrylonitrile-butadiene-styrene
API	Application Programming Interface
BCI	Brain Computer Interface
CAD	Computer aided design
CB	Certification Body
CE	Conformité Européenne (European Conformity in French)
CSV	Comma-separated values
DOF	Degree of Freedom
EMBS	Engineering in Medicine and Biology Society
EEC	European Economic Community
EEG	Electroencephalography
EMG	Electromyography
EU	European Union
FAST	Function Analysis and System Technique
FCC	Federal Communications Commission
FDA	Food and Drug Administration
FDM	Fused Deposition Modelling

FFT	Fast Fourier transform
GUI	Graphical user interface
ICES	Institute for Clinical Evaluative Sciences
IEEE	The Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization
MCP	Metacarpophalangeal joint
OSC	Open Sound Control
PC	Personal Computer
PCB	Printed Circuit Board
PIP	Proximal interphalangeal
PLA	Polylactic Acid
SLA	Stereolithography
SLS	Selective Laser Sintering
STL	Standard Triangle Language
OSHA	Occupational Safety and Health Administration
USB	Universal serial bus

1.OVERVIEW

Losing the function of a limb would restrict an individual's life experience in an irreversible way. Basic daily activities, like grabbing a cup, throwing a ball, hand shaking with a person might be uncomfortable activities for the people who lost the function on one of their upper extremities. To minimize the negative impacts of the functional or complete loss of an arm, this project proposes a solution: A robotic arm controlled by the captured thoughts of an individual with the help of electroencephalography (EEG) technique. Since this kind of product can only be achieved by a collaboration of people from different backgrounds, a multidisciplinary team has worked in this project. There are students from Biomedical Engineering and Mechatronics Engineering departments with different capabilities and interests to yield one common product as the result of this project.

Nadim Absi is the volunteer in the group whom the EEG signals are collected from after several training sessions. He also contributed to the project by bringing his problem-solving skills and optimization from his background on Industrial Engineering double major.

Irmak Akoğlu took a role in signal quality check and adjusting the headset during training sessions. She was the responsible person for budgeting and documentation. In addition, she was the contact person with the school when faculty needed to be informed to access to the lab.

Yara Corky is the double major student who had responsibilities from both departments. While she was working on the coding to distinguish different EEG commands on biomedical side, she also showed her efforts on mathematical modelling of the robotic arm on mechatronics side. She was the bridge between two departments with her knowledge on both fields and her key role was developing the interface which is the key point of integration.

Issa Hoballah managed the mathematical modelling and forward & inverse kinematics of the robotic arm. He was created Arduino code and put his efforts on building the prototype. Initial testing of robotic arm with a video game controller is also performed by him to verify the motions of the arm.

Yusuf S. Kurt was the designer of the robotic arm. He utilized computer aided design (CAD) to create parts that 3D printed . By his efforts, the full robotic arm concept has printed, and he made necessary changes on the design through the course of project. Furthermore, he dealt with power management for the operation of the robotic arm.

Ahmad Maassarani printed all the 3D parts in his printer and assembled the parts of the arm together. His contributions made it possible to have the physical model in real life based on the 3D designs. He was the main responsible person for material selection.

In the first semester of this project, it was aimed that the robotic arm would grab a glass of water and direct it to the individual with the help of his/her mental comments. Even though essence of the movement performed by the robotic arm is not changed dramatically, the scope of the project has narrowed down. Final result of this project is a robotic arm which can grab a given solid object, instead of drinking water from a cup. This change is made due to several reasons:

- Training the individual for specific commands took longer time than planned due to restrictions of Covid-19. Especially during lockdown, trainings were interrupted for at least a period of 2 weeks which was a precious amount of time that affected the progress in drastically.

- Servo motors which was ordered in the beginning of the semester were defected and they needed to be replaced with different models. This slowed down the assembly process.

-Due to the limitation of the commands which can be obtained, one degree of freedom (DOF) reduced on the design of the shoulder. This restricted the flexibility of the movement.

Due to the reasons mentioned above, the main aim of the project is to control the robotic arm as precise as possible with the EEG commands to grab an object.

1.1. Identification of the need

The EEG-controlled robotic arm is an electromechanical prosthetic device that is controlled by the brain signals from the motor imaginary activity of an individual. These signals are detected and recorded by an EEG headset. Raw signals are fed into the software interface to be processed and then classified commands are sent to the microcontroller of the robotic arm. The microcontroller is programmed to control the servo motors which will perform the motion according to received EEG commands. To perform these tasks, requirements of the project are discussed and presented in this section. Scope of the project is decided according to the possibilities and limitations to achieve a realistic goal while still serving a solution to the described problem on the previous section.

The robotic arm should perform the motion of a natural arm as close as possible to the real movement. However, budget and time restrictions and availability of mechanical components are considered to set a realistic goal. Due to that, robotic arm should perform at least lifting from the elbow, grabbing a given object and rotate or release this object. Robotic arm is not wearable in the design scope of this project. At the final presentation, the end-product of this project supposed to function as stated above.

This project aims contributing to the research of how to increase life quality of people who lost motor function in their arms. More detailed explanations of the target of this product are given as in the below questions:

WHO?

The arm can be used by any person who does not have any brain function loss since the control of the arm will be done by the EEG signals. In addition, individual should have healthy audio-visual sensory since the trainings are needed to yield EEG commands. Paralyzed people who are unable to control their body and limbs would be the desired target user to develop the product according to assistive needs of these people in daily tasks. However, in the scope of this project an induvial who does not have any incapability will be the subject.

WHAT?

Mental commands are used to control the robotic arm. These commands are generated by the brain activity of the user of the headset. Robotic arm is 3D printed and controlled according to mental commands to grab the object according to its microcontroller.

After the signals of commands are acquired from the brain of the user, the signals are transmitted to an interface software, and they are classified accordingly. Then the classified signals will initiate a specific movement in the robotic arm according to the task assigned to them. The motors and controllers of the robotic arm will perform the task given by the input signal and the specific movement which the user thought of.

WHY?

The robotic arm is designed to make the lives of paralyzed people easier by giving them the ability to get perform some tasks they need in their daily life. Even the final result would achieve this aim partially, it will be a good reference for future projects.

WHERE?

The robotic arm can be used in any place or environment in which the user is able to secure the arm around a solid object with space around it. Also, EEG headset should be worn and ideally it should be an environment with minimum signal interference.

WHEN?

The final state of the robotic arm will allow the user/patient to use the arm at any time s/he wants to grab an object in nearby surroundings of the robotic arm.

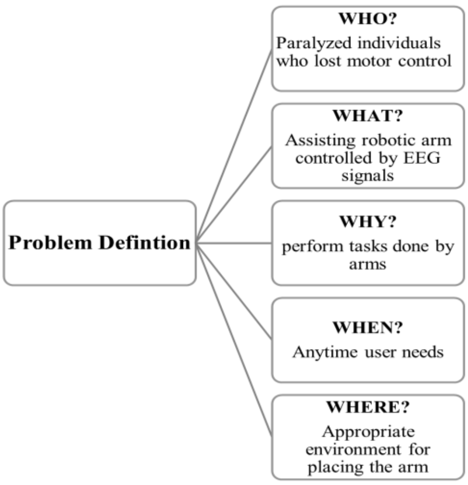


Figure 1. Problem Definition Diagram

1.2. Definition of the problem

Motor function loss is an eminent problem affecting millions around the globe. According to the World Health Survey, around 785 million people live with some form of disability. But around 190 million people of those have severe motor functions disability [1]. These numbers include people with various problems from quadriplegia , tetraplegia, blindness, etc. With each passing day, the number of people suffering from disabilities increases. Hence, this project aims to help the effected individuals, and solve such an abstruse problem: regaining some of their motion abilities. To match with this aim, there are some requirements to be satisfied in the boundary of the limitations of this project. These requirements and constraints are represented briefly in this section.

1.2.1. Functional requirements

Mental commands should smoothly control the movements of the robotic arm. They should be easy to train to not exhaust the user. Interface which translates those mental commands to mechanical commands should perform a classification of different commands.

Robotic arm should perform the basic tasks such as grabbing an object. It should have enough flexibility of the movement to convert the EEG commands into actual movements. This flexibility can be achieved by the number of DOF and angles of the servo motors.

1.2.2. Performance requirements

Highest accuracy on mental commands should be achieved with minimal misclassification between distinct ones. Otherwise, there would be deviations from desired movement of robotic arm if wrong commands are sent. To overcome this problem, number of trainings for each command should be maximized. Accuracy of the commands should be also ensured by limiting the intensity of the received signal and excluding the ones who are below a threshold. In addition, sending commands should be as close as possible to real time to take the most benefit from the control of the robotic arm.

Robotic arm should mimic the natural movements of an arm as maximum as possible. In the design, number of DOF should be considered accordingly to achieve that flexibility of the motion. Also, maximum angles should be limited to avoid overexertion of the motors if excessive mental commands are sent. Lastly, motion of the arm should be verified by another controller than EEG commands to see smoothness and all possible DOFs that this device can achieve when receiving the non-interrupted signals.

1.2.3. Constraints

Main constraints of this project can be classified under 3 categories: time, budget and physical constraints. Each of them will be discussed in this section.

In addition to that, economical, ethical, social and enviromental effects will be debated following to constraints.

Lastly, standards that are followed will be represented at the end of this section which limits the architecture. Though, they are still needed to guide the project to stay above a quality threshold.

1.2.3.1. Time, Budget, and Physical Constraints

-Time Constraints

Due to the iterative nature of EEG analysis, a clear scheduling is essential to complete all the tasks on time and to get optimal results. Also, system integration between the EEG signals and robotic arm is a time consuming and challenging procedure. Hence, all tasks performed in this project are going to be according to the Gannt Chart which is elaborated at the Work Plan in the next section. Globally experienced pandemic might slow down overall workflow. Possible delays are also taken in consideration while creating the Gannt Chart including the effect of lockdown in May.

-Budget Constraints

Ideally, this project should abide to the limit granted by the university for each student working on their capstone projects. Furthermore, budget should be formulated as minimal as possible while achieving optimal results. All facilities provided by the university are utilized with maximum possible efficiency regarding to the current ongoing pandemic. Possessions of team members are also used as a contributing factor for accelerating the procedure. Bahçeşehir University is offering up to 500₺ per student to support capstone projects. Additionally, we are permitted to use the EPOC+ available in the biomedical laboratory which helps contributing to the budget of this project. Also, a member of our team owns a 3D printer; in which contributes to printing some of the body parts needed to keep the project economical. Fragile EEG electrodes created an extra cost which did not predicted in the proposal of this project. Replacing them was an unexpected expenditure for the project. However, it is still possible to keep the project budget below the maximum threshold as planned at the beginning.

-Physical Constraints

Lockdowns and social measures for COVID-19 are heavily restricting the team's mobility overall. Meeting face to face is avoided as much as possible, abiding by the laws in the Turkish Republic regarding to health and safety regulations. Hence, regular online meetings are held over online streaming platforms such as MS Teams to organize workflow and make this project possible.

The probes of the EPOC+ are highly sensitive, causing the signal strength to be negatively impacted by hair density, and frequent sensor lubrication is required to ensure the accuracy of the signals. Also, due to the sensitivity of the probes, the headset must be firmly secured all the time. Additionally, regular maintenance should be done on the golden plates of the sensor electrodes to avoid corrosion formation. To minimize fragility on attaching parts of the electrodes, twists should be gentle while replacing them on the legs of the headset. Broken electrodes should be inspected and reported immediately and replaced in short term to not interrupt training progress.

To ensure human mimicked arm motion, the robotic arm will be constrained with angular limitations and certain flexibility rates (ex: human elbows extend from 0° to about 140°). The whole arm is going to have 4 DOF and an end effector. Number of DOF corresponds to the number of EEG commands that are managed to be generated. Also, in the physical architecture section, arm dimensions and motion mechanism descriptions are provided.

1.2.3.2. Environmental, Social, and Economic Effects, Ethical Issues

-Environmental effects

Due to the model is going to be printed out of PLA, there are some environmental effects to be discussed. PLA is compostable and not biodegradable, which means that microbes are not able to break PLA down to gas and biomass. On the other hand, being non-biodegradable means they take a lot of time to breakdown which can pose environmental threats. Hence, recycling any wasted PLA is taken into consideration when synthesizing the arm.

-Social effects

Robotics have been debated for decades whether they are intelligent enough to dominate the human race or replace people's job, but in this case, it is aimed to improve the quality of life without causing any scandals. Above that, all details from safety to standards will be provided, avoiding any potential misuse, and misunderstanding of the product, to clear all ambiguities. Also, the risk analysis shall provide all possible risks of the product, alongside how to prevent or protect oneself from such risks. Outcomes of this project can provide a futuristic projection for further projects such as full exoskeleton controlled by EEG. Keeping that in mind, the current mere EEG controlled robotic arm teaches us how to control robotics in generally by using brainwaves. Although this seems like such a positive impact to be made which is what is intended, some people might differ within opinions whether such a thing opposes their life perspective or against nature. Hence, social impact is not always going to be positive, in the end not all people can be satisfied. Remaining ethical and following the standards should avoid any controversy.

-Economic effects

This project is not going to be industrialized. The prototype is for personal use for those who are in the target group. The current version of the project does not oppose any economic effects as it will not be in use until it would be improved to in future projects to be a full functional arm or beyond. Once it reached that development level, additional maintenance and installation costs may apply in future. Taking in consideration that the arm is for personal use, all the equipment combined to make it work can be costly as all the parts need to be purchased individually and assembled, making it more on the expensive side.

-Ethical issues

In a project that has such a small scope and current applications, faced ethical issues are minimal. Staying eco-friendly by using PLA and recycling waste to avoid damaging the environment is the first ethical step to take. Also, physical injury possibilities are assessed, discussed, and limitations on motion are applied to reduce the possibility of undesired incidents. Most importantly, no animals are going to be used for experimentation, and noninvasive methods are used to prevent any potential harm to the subject/user. Chosen headset does not have any negative impact of the health and subject was one of the group members who was volunteered to be the subject. Lastly, EMBS and IEEE ethical codes are reviewed and everything is referenced to avoid plagiarism.

1.2.3.3 Standards

In this project standards from several organizations are reviewed. Code of Ethics from IEEE [2] and EMBS [3] are followed. OSHA [4] standards are reviewed for compliance with emergency termination switch on microcontroller. Quality of mechanical parts are ensured by ISO standards [5].

Standards for each sub-system are discussed in this section as the order of appearance of the sub-systems in this report.

Standards for Biomedical Engineering sub-system:

A medical device will not be developed in this sub-system. However, the EEG headset which collects the data has a direct contact with the individual. Hence, headset as an electronic device must satisfy some certain standards for the safety of the user. EPOC+ headset from Emotiv is the headset that is preferred in this project.

EPOC+

EPOC+ is a device designed to be used in research applications or personal use only. So, it is **NOT** classified as a Medical Device as defined in EU directive **93/42/EEC**.

It is tested for Electromagnetic Compliance to ensure it can work without creating any disturbance for the other devices on the same environment. Likewise, its Safety Compliance is also checked under the CB scheme. Safety and wireless interference standards are certified by worldwide recognized bodies in Australia, and North America. It also has a CE mark which ensures the quality in European Economic Community. Related ID numbers are provided by manufacturer as follows:

- FCC ID Number **2ADIH-EPOC02**
- IC ID Number: **12769A-EPOC02**
- USB Dongle - FCC ID Number **XUE-USBD01**

Manufacturer also states that the device does not cause any harmful interferences. If it receives any interference, there is a high probability that the interference is caused by a device nearby in the environment where headset is used. (e.g., other Bluetooth devices on the room might affect the recordings.)

It satisfies the regulations to be categorized as **Class B** digital device. This complies **ICES-003** in Canada, **part 15 FCC rules** in United States and EU radio equipment device directive **2014/53/EU.** So, it has medium risk level.

Although, radio frequency generated by headset is designed not to create any disturbance on other device performances, it is noted that it could interfere some radiofrequency communication around when the instructions are not followed as stated on the user manual [6].

Type	Standards Tested
EMC and Telecom: Class B	TSI EN 300 440-2 V1.4.1
	EN 301 489-1
	EN 301 489-3
	AS/NZS CISPR22 :2009
	AS/NZS 4268 :2008
	FCC CFR 47 Part 15C
Safety	EN 60950-1:2006
CB Certificate (TUV Rheinland Japan)	JPTUV-029914

Figure 2. EPOC+ Standards Scheme from User Manual

<u>Standards for Mechatronics Engineering sub-system:</u>

To ensure consistency, compatibility, effectiveness, and safety of the project, physical arm is constructed according to multiple different standards. According to OSHA Instruction PUB 8-1.3 SEP 21,1987 Office of Science and Technology Assessment [4], a dynamic brake system for emergency robot breaking must be implemented, to avoid potential hazards of the robotic arm’s motion. However, when the scale of the robot considered only a current breaker switch is used on the microcontroller of the robotic arm.

Standards of the components used in construction of the robotic arm are as follows:

- PLA: RAL9005.
- Arduino microcontroller: T.E S011RJ95V-0.
- Servo motor: 0V-CN-AA160.
- Copper threads: M3.
- Stainless steel screws: M3.
- Copper wires: TS9760.
- Limit switch: ISO9001, IATF16949, and IP67.
- Super glue: EN ISO/IEC17025:2017.

1.3. Conceptual solutions

Before choosing the design solution for this project, similar projects on the literature are reviewed. They will be briefly mentioned here before and more detailed representations will be given in the conceptualization parts of the sub-systems.

1.3.1. Literature Review

Coming up with a concept is dependent on the project's performance, technical and safety requirements. In addition, following an in-depth literature review of all components of the system, alternative solutions are generated. These solutions are examined separately to select the most suitable option for the project in according to the preset criteria.

There were various alternatives that inspired the conceptual solution of the robotic arm in this project. For instance, InMoov Project illustrates many design solutions for the robotic human-like parts [7]. Furthermore, there are different projects as well such as Reachy Project [8] which describes a different point of view for the conceptual solution.

For the BCI applications, several articles are covered to understand the working mechanism in more depth. St. Germain describes the capabilities of a BCI application to control a Lego arm [9]. This work illustrates a basic solution by using Emokey application to achieve simple movements. Another work has done by Beyrouthy *et al.* where they trained a user to open and close a robotic arm [10]. Last but not least, to understand the data processing with EEGLAB research has been conducted by Delorme and Makeig is reviewed. In their work, they explained how independent component analysis can be evaluated and how EEGLAB can be used for verification purposes [11]. However, some of the software tools used in these papers are replaced by newer alternatives which are employed in this project. Those solutions are described in detail in the embodiment design of the related sub-system in the next sections of the report.

1.3.2. Concepts

Starting with the EEG data collection hardware, the headsets available in the lab are the Emotiv EPOC+ and Emotiv FLEX. Although other solutions are available in the market, it would be wise to use what is already available to avoid additional cost. The EPOC+ is an EEG headset that carries 14 electrodes (+2 as references and +2 act as comforters), where each of those is capable of detecting EEG electric signal from a specific region, which is then wirelessly sent to PC through Bluetooth using a Universal USB Receiver (Dongle). Data is then presented through the Emotiv BCI applications [12]. The FLEX is the more advanced and robust headset of the two, and with its 32 electrodes, it can gather more accurate data that is more similar to traditional medical EEG caps, however it also comes with a cost of complexity and application challenges of gel-based electrodes. Data obtained by EPOC+ headset is in the same fashion as the FLEX with less complexity in application and analysis even though less number of electrodes might be a drawback [13].Ultimately, EPOC+ is chosen for the project applications due to its simplicity and ease of use. The complexity of the FLEX, although enticing, would hinder our progress. Explanation is elaborated further in the section 3.1.2.

For the EEG software, the BCI acts as main communication channel between the human brain and the computer interface. This is where the data collected by the chosen headset will be sent to be processed, analyzed, and transformed into a understandable signal by the microcontroller that would initiate the mechanical parts of the arm [14]. There are many open BCI available in the market thus, choosing the simplicity program that is capable to be integrate with the arm software would be the most sensible option. Firstly, there are the Emotiv PRO and BCI. These are the main apps accompanying the Emotiv headsets especially designed for them. The BCI allows the user to train and trigger events through mental commands and/or facial expressions. Since they are designed to be compatible with the headsets, they excel by having simple real-time viewing of the EEG waveforms [15], [16]. Also, the training of commands and facial expressions on these apps saves a lot of time. However, they fall behind in the processing and analysis part due to the lack of integrated filters and components analysis techniques. That is why another software is considered which is MATLAB-EEGLAB. EEGLAB an open-source toolbox and GUI designed for MATLAB. It allows the importing of raw EEG, easy visualization of the data, preprocessing, component analysis and time/frequency decomposition. The toolbox allows users to interact with the data directly and customize it according to their needs [11]. Basically, it can do what the BCI and PRO cannot, and it can also be used as a verification method. While both these programs cover for each other the process side of EEG signals, they do not solve the problem of communicating with other systems. In order to effectively utilize EEG signals, an interface is required to send the signals to the microcontroller controlling the arm. This is where Cortex API or Node-Red come in place. Both of these are new flagship programming tools/developer kits which allow communication between the Emotiv programs and other software. As such, it is chosen to an integrate system is proposed by where initial training and analysis can be done in the BCI and PRO and then, using Cortex API, the signals can be sent to the Arduino to control the arm. There are other options also available which are kept as alternatives if any revision will be needed. These are explained more in detail in the section 3.1.2.

The conceptual design chosen for this project consists of the hand, given in the Fig. 20, and the forearm, arm and shoulder given in the Fig. 23. These conceptual designs were selected out of the other alternatives and because these were considered as the optimal solutions to create similar design for this project. As illustrated in detail in the section 3.2.3, most of the design solutions for this project were eliminated because of the constraints established for the project in general.

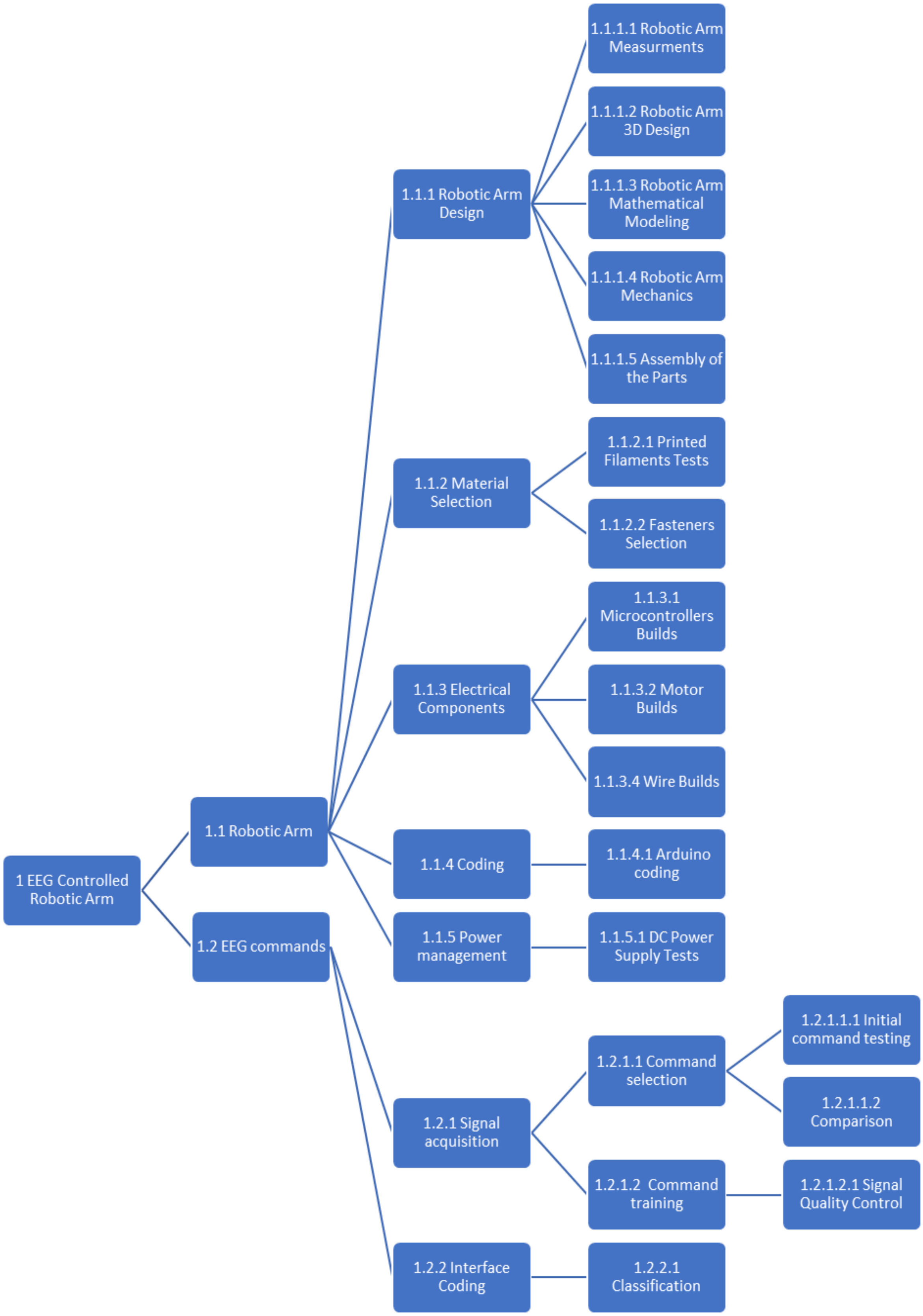
1.4. Architecture

Architecture of this project is utilizing Emotiv PRO and BCI applications for EEG data acquisition. Together with them development API called Cortex is used in the main part of coding in the Python language. Node Red toolbox is also another alternative solution which is getting discovered in the current course of the project to ease the way in interfacing. Also, EEGLAB is used to understand the characteristics of the data and planned to be used in the verification step later on. Further details will be given in the section 3.1.4.

Physical architecture of this project consists of the 3D printed parts being assembled and of course with the external parts such as the motors and the other components. Additive manufacturing is chosen because of many reasons such as low-cost and high flexibility. Furthermore, 3D printing method is chosen to be the fused deposition modelling (FDM) due to the same reason. Moreover, the tendons were chosen to be some strings being pulled by the servo motor that operates fingers to close or open. Forearm yaw and elbow pitch movements are obtained by the attached servo motors. Also, shoulder roll and shoulder pitch are provided by servo motors. More details are provided in the section 3.2.4.

2. WORK PLAN

2.1. Work Breakdown Structure (WBS)

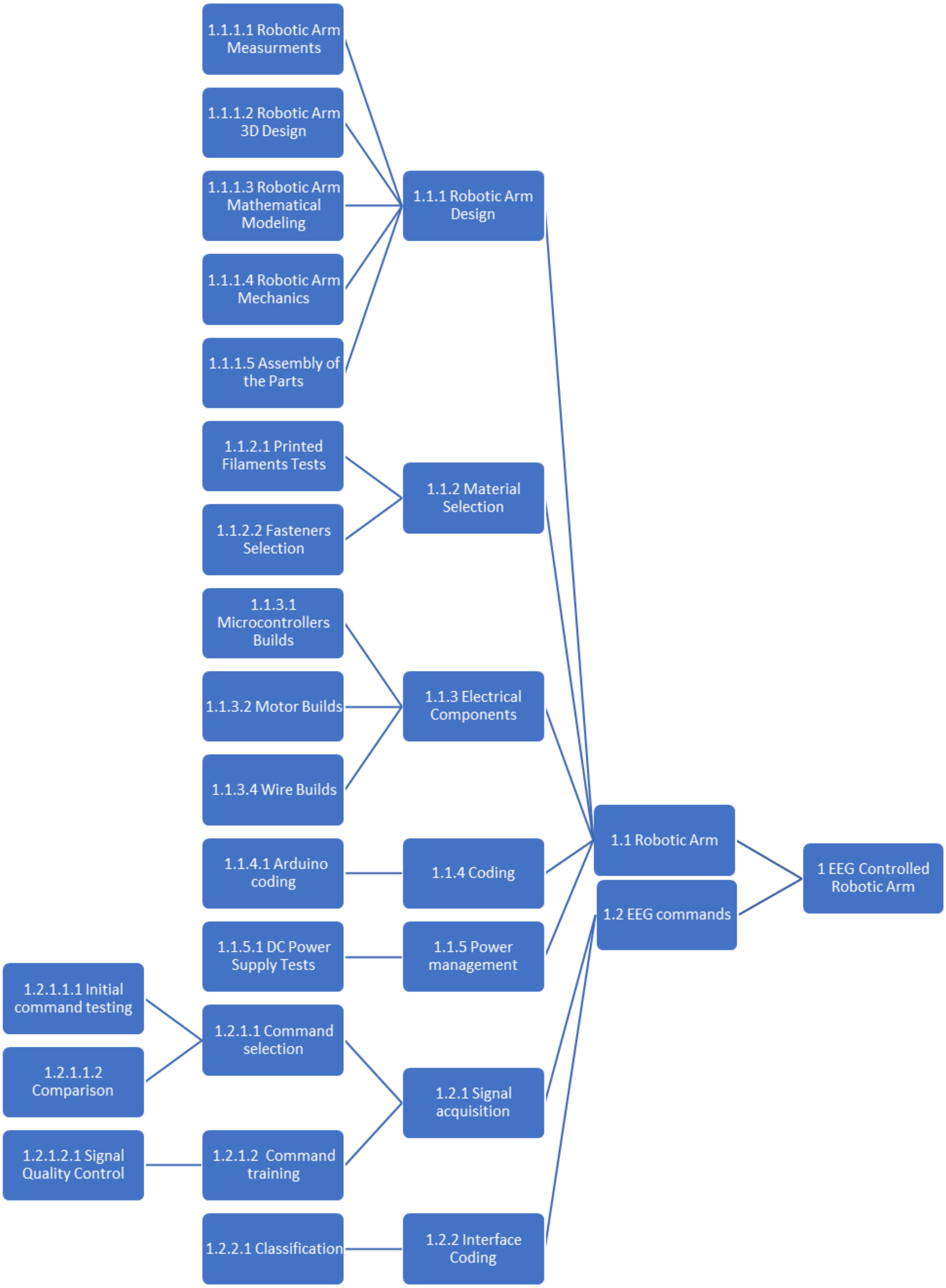


2.2 Responsibility Matrix (RM)

Task	Ahmad	Issa	Irmak	Nadim	Yara	Yusuf
A) Robotic Arm Measurements		S				R
B) Robotic Arm 3D Design	S				S	R
C) Robotic Arm Mathematical Modeling		S			R	S
D) Robotic Arm Mechanics	S	R			S	S
E) Material Selection & 3D Printing	R	S				S
F) Microcontrollers Builds	R	S			S	S
G) Assembly of the Parts	R	S				
H) Motor Builds	R	S				
I) Wire Builds	S	R				
J) Arduino Coding	S	R			S	S
K) Power Management	S					R
L) Command Training			S	R	S	
M) Command Testing			S	R	S	
N) Signal Quality Control			R	S	S	
O) Cortex Coding			S	S	R	
P) Project integration	S	S	S	S	R	S
Q) Project management		R				
R) Documentation			R			

R = Responsible, S = Support

2.3. Project Network (PN)



2.4. Gantt chart

<table>
<tr><th>Week</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th></tr>
<tr><td>Ahmad</td><td colspan="2">E</td><td colspan="7">F, G, H</td><td></td><td></td><td></td><td colspan="2" rowspan="6">Integration in the Extension Period</td></tr>
<tr><td>Issa</td><td colspan="3">D</td><td colspan="4">I</td><td colspan="3">J, Q</td><td></td><td></td></tr>
<tr><td>Irmak</td><td colspan="11">N, R</td><td></td></tr>
<tr><td>Nadim</td><td colspan="11">L, M</td><td></td></tr>
<tr><td>Yara</td><td colspan="3">C</td><td colspan="3">O</td><td colspan="6">P</td></tr>
<tr><td>Yusuf</td><td>A</td><td colspan="8">B</td><td></td><td colspan="2">K</td></tr>
</table>

Blue is for Mechatronics, Orange is for Biomedical, and Yellow is for mixed tasks.

All the planning of the project is given in this section. In the first constructed flow it was planned to finish the tasks by the 12th week. However, due to lockdown in May for 2 weeks an extension is taken an integration of both subsystems are done in this extension period. Some last-minute changes in the finger design and reprinting is also done in this period.

2.5. Costs

Table 1. Table of Costs

Components	Cost (TL - ₺)
Arduino Mega	398 ₺
6x FSR402 force sensor	330 ₺
8x MG995 servomotor	328 ₺
Wires, resistors, capacitors, others	50 ₺
PCB (20x30) sheet	31 ₺
TOTAL	**1137 ₺**

Table 2. Actual Costs

Electro-mechanical	
Arduino Uno R3	186.44 ₺
2x IDX-218	372.73 ₺
2x Servo Motor metal head	26.08 ₺
3x MG945	161.01 ₺
Emergency stop button	19.90 ₺
TOTAL	**766.16 ₺**
Utilities	
2kg PLA filament	231.28 ₺
3D printer nozzles and nozzle drill	35.4 ₺
Bolts, Screws, tools	64.90 ₺
Plastic strings and tubes	22.5 ₺
Super glue	20 ₺
Wire shrinks	26.95 ₺
TOTAL	**401.03 ₺**
EEG Headset	
16 Electrodes with Hydrator Pack	916.56 ₺
TOTAL	**916.56 ₺**
Overall Total	**2,083.75 ₺**

As shown in the tables 2 & 3, the components predicted to be used had changed when the project reached the execution phase. Even though actual costs are higher they are still in the scope of the project budget. The main increase was due to the need for a replacement of broken electrodes which was not predicted on proposal.

2.6. Risk Assessment

Possible risk factors on EEG recording procedure are represented below:

1. **Seizures / Epilepsy :** Although the EEG is a safe process with no risk of electric shocks or radiation exposure, depending on method of data collection, subject may experience a seizure due to epilepsy. This can happen if a high frequency, visual stimuli was used during testing. This risk has a low probability but an extremely high severity, so it is best to avoid using such stimuli in the first place and even if it is utilized, employed frequency should be limited within a safe threshold.
2. **Overexertion of subject:** Initial testing with several users revealed that wearing the headset for long period of time can exert the subject quite heavily. Also, the readings would start getting affected as the subject would not be able to concentrate. A moderate severity risk but with a high probability and as such, the best plan of action is to limit the number of sessions per week and to limit the recording intervals as short as possible.
3. **Environment affecting subject:** Initial testing also revealed that the lab environment can also heavily affect the user's concentration. Parameters such as noise, outside interference, lighting, number of people in the lab etc., affected the quality of recordings. This high probability, low severity risk can be avoided by limiting number of people in the lab and, setting a standardized procedure that would be followed in each recording to maintain the same environment on every session.
4. **Damage to headset:** The headset itself is not very sturdy and has multiple components which can be either lost or broken. Damaging the headset would be very costly and hinder any further progress. A medium probability but highly severe risk that can be prevented by taking extra caution when handling the headset and make sure only one person at a time interacts with it. Make sure no one from outside of the team uses it and it should be stored safely in its box when done using.
5. **Environment affecting headset:** The lab equipment and the hair on the subject's skull are hindering the readings. This is evident by the amount of noise and artifacts observed in initial testing. A high probability moderately severe problem with the only possible plan of action is to deal with it in the processing and analysis part. The possibility of using the headset outside the lab is impractical but would be the best solution. This can be explored further if permission was granted.

6. **Difficulty in interpreting result:** Due to the complexity of EEG, the result could be hard to understand even with all the analysis tools that are planned to be used. This stems from the fact that observing the Emotiv PRO is sometimes not enough to understand what the signals really mean. A possible probability, moderate risk problem which can be dealt with by consultation of experts or leaving the analysis part to the software by using Cortex.
7. **Obtaining non-repeatable results:** The project requires the signals to be repeatable to establish a set of inputs that correspond to specific EEG waveforms. However, repeating the exact same EEG waveforms is difficult due to a multitude of reasons which range from the user to software to environment. With a possible probability and moderate severity, best course of action is to use a simple set of commands for the user to perform and train over.
8. **Fragile electrodes:** After several sessions, it is observed that the electrodes used are extremely fragile and can break easily. Moreover, soaking them with saline was causing them to become extensively corroded over-time. Although the electrodes were still functional, it became extremely difficult to fit them on the headset. Instead, the electrodes had to be stabilized using against the subject's head until new set of electrodes are purchased. This problem can cause the contact quality to drop frequently which tinkers with the data recordings and EEG quality. As of the middle of the second semester, almost nine electrodes were at least partially damaged and needed a replacement. The solution to this likely probable and highly severe problem is to take utmost care when handling the electrodes and replace them with a new set of electrodes as soon as possible to ensure the contact quality remains acceptable when taking recordings.
9. **Difficulty in linking the headset to the mechanical arm:** Since this is the goal of the project, it poses a high possibility and high severity due to the project's complexity. It can be due to a multitude of reasons from software to hardware and from time to budget. However, the most prior one would be communicating several different software protocols with each other. Ultimately, the best plan of action is to determine early the software components that will interact with one another and, ensure they will be compatible.

Table 3. EEG Risk Assessment Table

Failure event	Probability	Severity	Risk level	Plan of action
Seizures / Epilepsy	Unlikely	Major	Medium	Avoid using visual Stimuli with high frequency
Overexertion of subject	Likely	Moderate	High	limit the number of sessions per week, limit the recording intervals to a max of 5 min each and up to an hour
Environment affecting subject	Likely	Minor	Medium	Setting standardized procedure, limiting number of people in the lab
Damage to headset	Medium	Major	High	Extra caution when handling the headset, only one person at a time interacts with it, tuck away safely
Environment affecting headset	Likely	Moderate	High	Deal with it in the processing and analysis part/ use headset outside the lab
Difficulty in interpreting result	Possible	Moderate	Medium	Consultation of experts or bypassing the analysis part by using different software
Results not repeatable	Possible	Moderate	Medium	Use a simple set of commands for the user to perform and train over
Fragile Electrodes	Likely	Major	High	Purchase a new set of electrodes and continue with the suggested solutions until then
Difficulty in linking the headset to the mechanical arm	Possible	Major	High	Ensure software and components on both sides are compatible before implementation

Possible risk factors in the working electro-mechanical system:

1. **Microcontroller failure:** microcontroller failure is always a possible event to occur with moderate severity. Such failure opposes no health risks to the user, but could have a few predicted outcomes, such as:
    - **System freeze:** the system could become totally unresponsive, which requires a reboot.
    - **System loop:** the system could get stuck in a loop and any given point of the code, which requires a reboot.
    - **System malfunction:** this event is highly unlikely but possible. In this event, the electromechanical system would be totally down, which requires a swap with another pre-coded working microcontroller.

2. **Short circuit:** It is an unlikely event to occur, but if it does, it usually comes with major severity to the sub-system or the user. In case of such an event, the components could get fried or even catch fire, which could lead to serious damage to the system or user. To avoid such a risk, installing a current breaker in the system would keep everything safe from such an event.

3. **Motor failure:** Another highly unlikely event to occur, with minor severity. If such an event occurs, it might be due to faulty manufactured motors or loose wires/low-high current. The aftermath of such an event is the buzzing and jamming of the motor. Changing the affected motors solves the problem.

4. **Interference in the signal**: It is unlikely for signals to mash up and interfere with each other. Such an event would have major problems though. If the signal to the arm gets interfered, it could cause unexpected arm movement or vibrations which could harm whoever is within the workspace of the robotic arm. Such an event can be solved by installing addition band-pass filters.

5. **Dangerous root movement due to power loss**: It is always possible to lose power due to a blackout or whatever reason, hence, an alternative powering battery needs to be implemented in the system to avoid severe outcome if such an event occurs. Uncontrolled or spasming motion of the arm can be very harmful to whomever is within the workspace of the robotic arm.

6. **Wire disconnection:** This is a possible event especially in a dynamic system like the robotic arm. Wire disconnection could lead to major problems such as power loss, dangerous movement, loss of functionality, and other problems. Proper wire management and selection should avoid this problem as a whole.

Table 4. Robotic Arm Risk Assessment Table

Failure event	**Probability**	**Severity**	**Risk level**	**Plan of action**
Microcontroller Failure	Possible	Moderate	Medium	Having multiple microcontrollers with pre-installed code on them, available on demand
Short Circuit	Unlikely	Major	Medium	Installing a current breaker
Motor Failure	Unlikely	Minor	Very Low	Replacing the damaged motors
Interference in the signal	Unlikely	Major	Medium	Additional band-pass filters considered
Dangerous robot movement due to power loss	Possible	Major	High	Implementing a dynamic brake system
Wire disconnection	Possible	Major	High	Proper wire management and selection

3. SUB-SYSTEMS

3.1. Biomedical Engineering

Introduction:

Individuals with extreme disabilities, including certain spinal cord injuries and paralyzed individuals, are particularly in need of electronic assistive technologies that increase their capabilities to accomplish certain simple tasks [17]. In this context, research methods to control robots have improved significantly in recent years, using biological signals such as electroencephalographic (EEG) signals to enable the human brain to communicate with controlled computers [18][19].

The purpose of this project is to control a robotic arm by using a Brain-Computer Interface (BCI) to train mental commands by utilizing the EPOC+ neuroheadset.

To succeed, distinction is needed between different EEG acts such as: cognitive controls for movement of the limbs. This goal could be achieved by applying the steps needed correctly. These steps include pre-processing of the signals, feature extraction, classification, and transferring the signals to the robotic arm.

<u>Pre-processing:</u> To retrieve the proper data contained in the signals, raw output is processed on this step to get cleaned and denoised. The elimination of artifacts is one of the essential elements of pre-processing. Often, artifacts arise from non-cerebral mechanisms. Even cortical energy potentials are often listed as objects. Non-physiological causes are often considered as artifacts, such as 50/60Hz power-line noise [20].

<u>Feature extraction:</u> The goal is to define essential components of the EEG signals that permit the evaluation of discrimination between actions. It is critical because the performance of the BCI system is decided.

<u>Classification:</u> This stage gives a class to a set of features derived from the signals and corresponds to the specified state of mind.

<u>End User Implementation:</u> It transfers the classifier's final output data to an external computer power. In this scenario, the output is sent to an Arduino microcontroller that controls the motion of a servo motor [21].

Based on all these steps conducted steps are represented in Figure 3.

Figure 3. Steps of the subsystem

3.1.1. Requirements

Objectives:

To achieve the aim of the project, EPOC+ headset by Emotiv is chosen to collect EEG signals required for establishing the system.

Emotiv PRO and BCI will be utilized to train and record the events related to mental imaginary commands, and for displaying the raw EEG signals with their characteristics.

Cortex API will be used for the real-time interaction between the headset and the robotic arm. Python will be the coding language.

Other alternatives for headset and software will be represented in the section 3.1.2.

All the components of the system are supposed to work together to succeed in executing the mental commands and make the robotic arm follow the commands.

i. Technical requirements

Product of mechatronics subsection of this project will be the robotic arm. Reliable control of the robotic arm heavily depends on the accurate choice of the input commands. Therefore, the product of biomedical subsection is yielded as in the form of processed, classified and digitally converted EEG commands.

Main functional requirement for the final product of this subsection is providing consistent and clear control of the robotic arm while operated by the target user. For a smooth operation of the designed system, following must be considered:

- Electrodes of the headset should be placed in correct positions.
- Recording environment should be arranged accordingly to minimize noise artifacts.(e.g., silence in the lab, no Bluetooth or electromagnetic devices are operating)
- Signal patterns should be recognized confidently.

- Suitable features should be highlighted from the accepted patterns.
- Classification of the features should be performed to obtain coherent input commands when those features are converted digitally.

Primary performance goal of input commands is allowing the user to control the robotic arm independently and consistently on every attempt with the minimum lagging from the signal. Performance of the product can reach to its highest potential if:

- The contact quality is nearly always 100% (momentary drops until 83% are still accepted)
- The process duration is minimized, ideally converges to real time for all steps from recording of the signal to communication with microcontroller.
- The input commands are designed to reduce or to eliminate the need of help from another individual in real life applications.

To meet with these performance goals EPOC+ headset is a suitable choice. Accuracy of this headset is dependent on the subject, environment, and placement of electrodes. However, it is indicated that it can provide high accuracy when only one mental command is trained at a time while adding more commands in short interval creates challenges to meet the same accuracy. Mood detection is more precise for excitement when compared with frustration or engagement as stated by the manufacturer. It can detect a signal on a sensitivity of minimum 0.51 μV, with 14-bit accuracy on 0.2-0.43 Hz bandwidth range [22].

Technical requirements of a design process can be illustrated by a Function Analysis and System Technique (FAST) diagram [23]. It is a graphical representation of how each step in a design process is done and why they are needed. Even more, it tells what other actions should be done simultaneously on specific levels when a task is performed. Functions must be defined with word couples to ensure each move is defined consciously. That will provide a clear understanding for the design process and ease the decision among alternative solutions [24]. A FAST Diagram is provided below to visualize technical requirements described earlier in a figure more briefly:

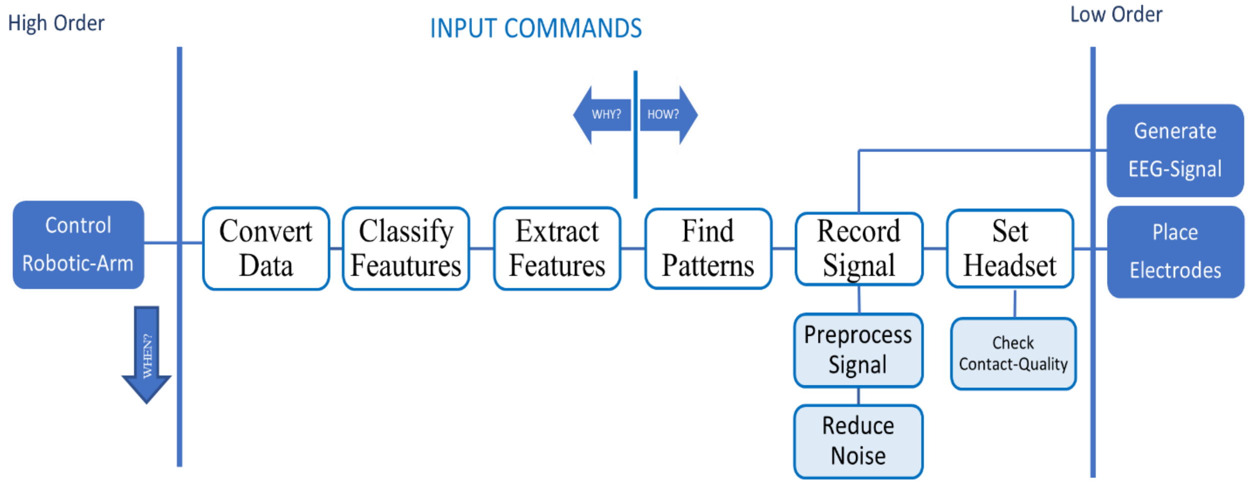


Figure 4. FAST Diagram of Functional and Performance Requirements

ii. Business requirements

Cost of the project can be shown as the main business requirement of the project. In this project required measurement and analysis tools are provided by the university. This is an enormous cost reduction to conduct all steps of the project. However, if a similar setting is conducted elsewhere, cost of all equipment should be reviewed for the budget. Resources provided by the university are listed below:

- Emotiv EPOC+ and EPOC FLEX Headsets
- Emotiv PRO Academic License and Emotiv BCI
- MATLAB License

Even tough, all devices and software planned to be used so far are under the roof of the university, according to future needs of the project other needs might be added up on cost.

Time management can be presented as the secondary business requirement of the project. Analysis of EEG signals is a highly time-consuming process. It needs several sessions to train the individual for specific mental commands. Also, noise reduction and finding the related signal patterns are labor-intensive tasks. Following that, feature extraction from patterns and classification of the data are all deliberate tasks. All these steps demand a well-designed scheduling to be able to yield the results on time.

An additional thing to be considered on business planning is external limitations. Due to the pandemic that is experienced worldwide, recording raw EEG signal at the university laboratory requires some additional adjustments. First, permissions should be taken to use laboratory facilities on campus. Besides, scheduling to avoid overlaps with other projects requires in advance planning. Finally, social distancing measures should be considered and number of the people in the laboratory and duration of recordings should be minimized.

3.1.2. Technologies and methods

This a review on project goals and technical requirements and, according to the review, variety of alternative solutions are generated, from which the final design will be chosen. To come up with a valid project design for the EEG system, the hardware and software alternatives will be examined separately.

i. Hardware: For the EEG side of the project, the main hardware component is the data collection mechanism. While the mechanism can be as simple as a couple sensor electrodes attached to the skull, this solution is too impractical to consider due to its inconsistency, low signal to noise ratio and general inaccuracy. Instead, we will investigate the two headsets available in the BAU BME laboratory, “Emotiv EPOC+” and “Emotiv EPOC FLEX”. For starters Emotiv Systems is a company that specializes in EEG headsets and BCI application, combining both in a single convenient unit.

a. EPOC+: It is an EEG headset that carries 14 electrodes (+2 that act as comforters), each of which can generate an EEG electric signal, that is then wirelessly sent to PC through Bluetooth using a Universal USB Receiver (Dongle). The data is

Figure 5. Emotiv EPOC+ Headset

presented, live, through the Emotiv BCI applications exactly like any other high-grade EEG. The head set also includes saline solution that is rubbed beforehand on each electrode to increase conductivity and accuracy.

Accurate placement of the electrodes is critical, and that is why the headset is shaped in such a way that the electrodes are aligned with the international 10-20 EEG placement system [9].

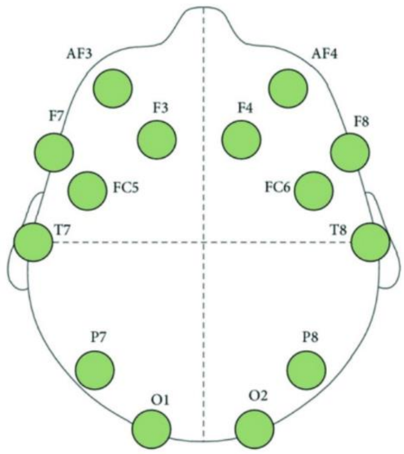


Figure 6. EPOC+ Electrode Positioning According to 10-20 System

b. EPOC FLEX: It is the more advanced, heavy EEG headset from Emotiv. It combines all the basic features of the EPOC+ headset with the accuracy and robustness of other high-quality EEG caps. Number of electrodes increase to 32, and this time they are gel electrodes which are more accurate. Unlike the EPOC+ headset, the electrode positioning here is configurable, allowing for the highest grade of data acquisition. The headset works in the same mechanism and software as the EPOC+ [13]. However, gel electrodes can create discomfort due to the residues of gel remained after usage.

Figure 7. Emotiv EPOC FLEX Headset

ii. Software – BCI: A Brain Computer Interface (BCI) is the main communication channel between the human brain and a computer interface. This is where the data collected by the chosen headset will be sent to be processed, analyzed, and transformed into a workable signal that can work with the mechanical part of the arm. The BCI relays the EEG signal from all electrodes and depending on the software used, the signals can be viewed in real-time in coupling apps [14]. However, mathematical analysis is not as simple and not all systems are capable of this. This section analyzes different EEG systems that could aid in the success of the project.

a. Emotiv PRO/BCI: accompanying the Emotiv headsets there are two applications, PRO AND BCI, especially designed for them. The BCI allows the user to train and trigger events through mental commands and/or facial expressions. These can be configured in real time according to the user needs. Performance metrics and emotional levels can also be tracked and, user-specific training profiles can be created and stored for different experiments [15]. However, the BCI mainly focuses on training and does not display any data collected. This is where the Emotiv PRO comes in. The PRO can display raw EEG, performance metrics, frequency bands, and emotional states, all in-real time. The software also allows quick data manipulation and filtering to a certain degree through amplitude bands and FFT. The PRO also offers additional features such as cloud data storage, CSV file exportation and event marking [16]. Both applications are available for free through the academic license that came with the headsets.

b. Cortex Application Programming Interface (API): Cortex API is a flagship application/developer kit from Emotiv that offers an extra, important feature that the previously mentioned applications lack. By manually developing a software/set of code that act as an interface, it allows direct communication between the data collected from the headset and other third-party software/devices. Mental commands or facial expression can be directly sent to other applications as command signals instead of going through the trouble of first exporting the data from the PRO, analyzing, and processing it in another software, and then sending the processed data to the electric components of the arm. After applying and registering for a license through Emotiv , you can start building the interface using coding software such as Python or C#[25].

c. Node-RED: A free accessible programming platform that enables the users to connect devices, networks, and robots without writing code. It has a drag-and-drop user interface with several nodes with different features, such as logical nodes, input, and output nodes to create custom flows. And they are combined to create an integration with computers or microcontrollers such as Arduino. The needed toolbox for our project is EMOTIV-BCI Node-Red toolbox which is a Node-RED input node library that enables the user to connect EMOTIV software to Node-RED and build a range of BCI based systems without having to code. Node-RED Toolbox can access trained Mental Commands and Facial Expressions via a cloud. In our case, the flow will consist of nodes of mental commands and facial expressions that each is assigned by a function to a specific string or character that is sent to the Arduino as a command to control a specific motor in Arduino [26].

d. MATLAB-EEGLAB: MATLAB by itself is great for numerical computations, algorithm applications and user interface design. However, instead of building a BCI interface from the ground up, the EEGLAB Plug-in exists to facilitate this process. EEGLAB is an open-source toolbox and GUI designed for MATLAB. It allows the importing of raw EEG, easy visualization of this data, preprocessing, component analysis and time/frequency decomposition. The toolbox allows users to interact with the data directly, independent from MATLAB. Customizable settings and functions can be used to process the data easily. With some training, users are further able to integrate the toolbox with MATLAB codes, thus enabling extra layer of communication to other devices [11].

e. Open Vibe, Python, C++, JAVA: These are some different programming languages that can be used as a BCI with the help of extra plug-ins, with Open Vibe being an actual open-source BCI. The problem with this lies in the fact that they cannot be integrated well with the Emotiv Headsets, that is especially true for Open Vibe since they use different file formats. And while the other programming languages due have a work around that, they require extensive training and knowledge, far more than what is needed in the previous example. That is why alternative options were left as worst-case scenario options that would only be considered if the previous alternatives fail.

3.1.3. Conceptualization and Embodiment Design:

Figure 8 shows the entire scheme of how the EEG data is collected, processed, and then sent to the arm. However, since EEG data analysis is rather a detailed process with several layers, a process flow chart would more accurately represent of the project. Through the suggested integrated BCI design, data would be using the headset though the trainings done by the subject in the BCI, this data is sent as separate command signals to the arm through the Cortex interface. The main advantage of such system is that it allows us to skip manually processing the data and go through the feature extraction and component analysis parts. This seems logical as they are already automatically performed by the BCI and any further analysis would only slow down the process and cause unnecessary delays.

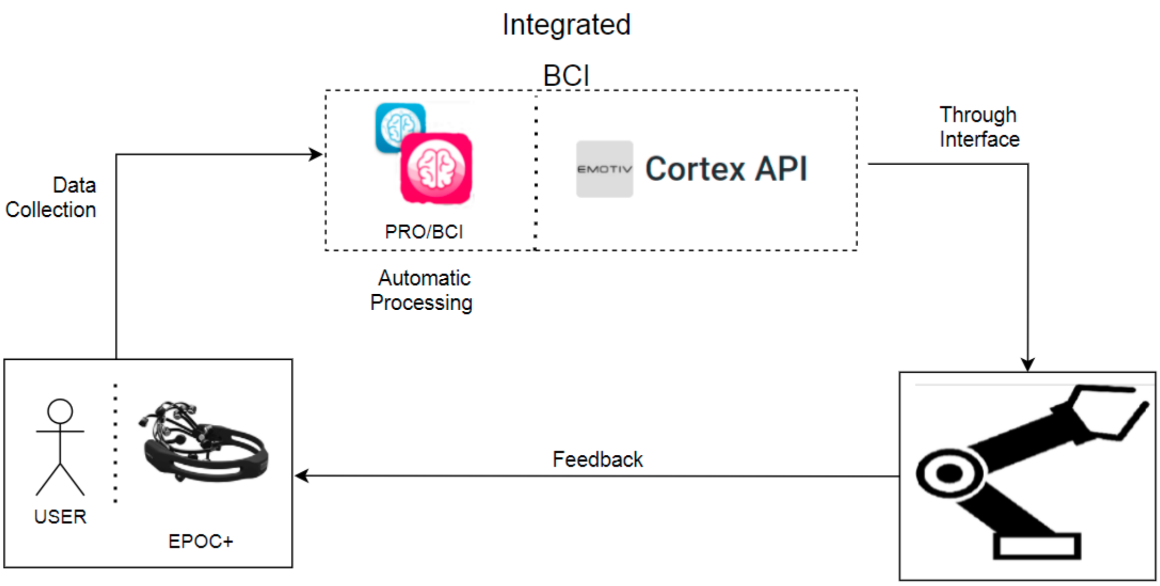


Figure 8. EEG Data Movement Representation

a. Data Collection: We decided to go with the EPOC+, instead of the FLEX. While the high accuracy of the FLEX is enticing, it is not necessarily helpful since, often, not all 32 or even 14 electrode channels are needed for picking up the mental commands. Furthermore, initial testing with the EPOC+ showed that setting it up takes a bit of time. Using a headset with a larger number of electrodes would only make this process more tedious. EEG signals based on thinking about some previously chosen photos during trainings for mental commands are the data that is decided to be collected. Facial expressions were also promising since they produced a distinct EEG shape every time. Coupling this two types is the most consistent solution.

b. Apps: Both the Emotiv PRO and BCI are going to be used extensively, since they are the main headset interface. While they do provide nice features such as the ability to train the user, sequential sampling, and filtering, they suffer greatly from not providing customizable processing and component analysis. That's why an interface is used for further classification and sending the signal to microcontroller

c. Interface: After the data has been collected. It needs to be processed, and the appropriate analysis methods should be carried out After that all what remains is how the data would be directly sent from the headset. Cortex allows us to use the live stream of EEG data to train the BCI and to control the Arduino which is something that was not achievable by MATLAB or other presented BCI apps. Node-Red also able to perform this function but further testing is required. Other solutions were also readily available, but they proved to be either too costly or complex. That's why Cortex is chosen as the interface which will learn the characteristics of the repeated signal and classify them accordingly. A serial connection to Arduino is also embedded in the main core.

Table 5 Comparison of BCI and Interface Concepts

	BCI/PRO	EEGLAB	Cortex API	Node-Red	Other
Cost	Free	Free	Free	Free	Free/High
Complexity	Low	High	Medium	Low	Medium/High
Performance	High	Low	Medium	Medium	Medium
Features	Low	Medium	Medium	Medium	Medium

3.1.4. Software architecture

Cortex codes was written in Python programming language. Through its code, training for mental commands and facial expressions was achieved, live display of EEG data streams according to the commands or expressions trained, recording, and marking the recorded data. There are codes in Cortex that shows Mental Command and facial expression training and live mode. Also, there are codes that shows the ability to get active actions, brain map, and training threshold and set action sensitivity in live mode.

In Cortex we have trained our user profile for four mental commands and neutral state, four facial expressions and neutral state. The aim is combining mental commands and facial expressions to utilize as many output signals as possible to control the robotic arm.

EMOTIV BCI that trains the user profile has a built in Artificial Neural Network that classifies the commands based on their power value comparing to the neutral state's power, and the neural network requires as many trainings as possible for it to get smarter and be able to distinguish the commands better. Our software architecture is connected to several EMOTIV apps like BCI, and PRO through a cloud of data storage and transfer which allowed the program to train EMOTIV BCI directly and access the neural network built inside it.

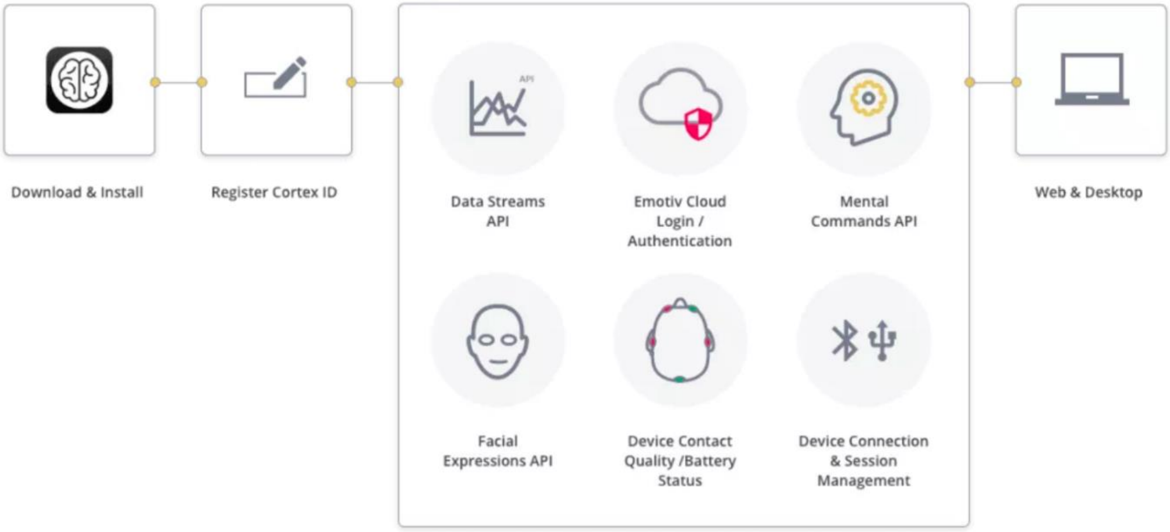


Figure 9. Cortex Architecture

3.1.5. Materialization

BCI: The entirety of the first semester and beginning of the second semester was spent experimenting with BCI/PRO to understand how they work and what works best when it comes to data acquisition. A problem frequently encountered was getting the mental commands trained during a trial, be repeatable in the next. While the subject was able to perform a sequence of acceptable trainings during the same day, he found it difficult to get the same commands working in another day or, would need a long time to get a single acceptable trial. According to Emotiv, thinking about the mental command as a "physical action", or trying to move the box presented in BCI would help with this issue, however, this solution proved to ineffective when performing it in the lab. After much experimentation, we found that using a specific picture that represents the action as a reference works. The subject would observe the picture for some time before the start of a session and then, would start training while thinking of picture. This method proved to be effective as the subject was able to repeat the results trained for a multiple of days and different commands. Using the above-mentioned methodology and as of now, the subject has trained for 4 mental commands (LIFT, DROP, ROTATECLOCKWISE and PUSH) and is able to perform acceptable trials without much effort in a smaller amount of time. Training facial expressions is a bit easier since they do not need an intense and rigorous thought process but, some calibration is nevertheless required. Some facial expressions (SMILE and RAISE BROWS) have been trained however they are not used at the end.

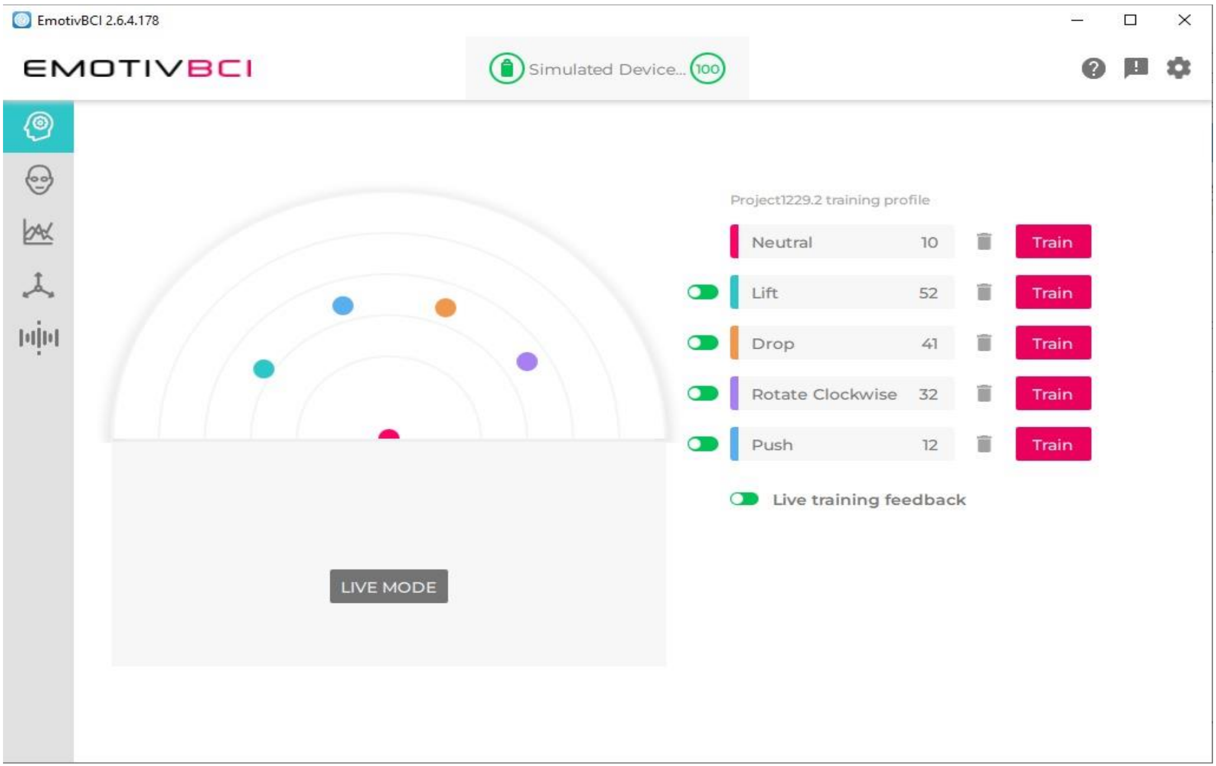


Figure 10. Example BCI interface of Trained Mental Commands

BCI application is replacing the trained commands in a range map as can be seen in Figure 10. When more commands are trained the distance between these commands are increased since the uniqueness of each command is understood better by the Neural Network of the software. By clicking on Train button next to each command trainings can be performed. In each training, a box is visible in the screen and if the subject is trained for that command, box moves in the command direction (for example it moves from bottom to top for Lift command). Otherwise, the box doesn't move. Continuous and coherent movements will be scored higher and only the scores higher than 70 will be accepted. Score from one of the successful trainings can be seen below at Figure 11. To ensure high scores, contact quality of electrodes are always checked. Trainings with 100% contact quality is accepted. Besides visual controls by the trainer, BCI also has a warning when the contact quality drops below 98%. This warning is represented in Figure 12.

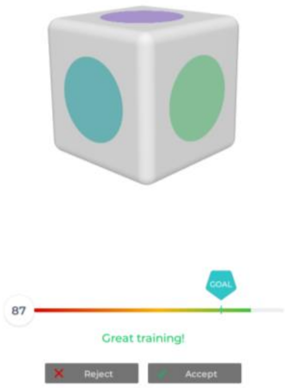


Figure 11. An accepted training with a score of 87/100

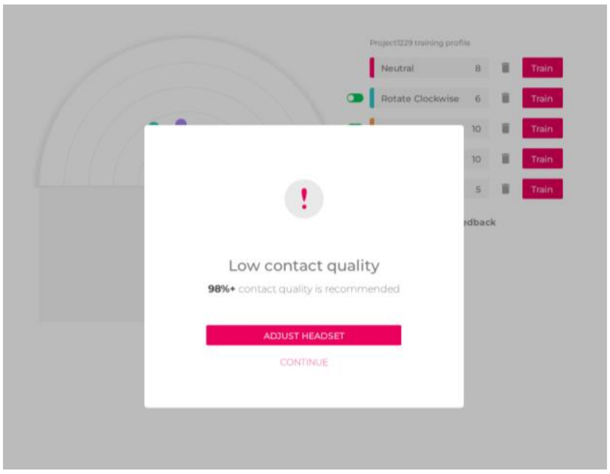


Figure 12. Low contact quality warning

PRO and EEGLAB: While experimenting with the BCI, the PRO accompanied the trainings by providing instantaneous EEG plots. In the beginning, we were manually marking the timings when the training took place and would then take download recording data to be analyzed on EEGLAB. Recordings from PRO can be seen at Figure 13 are synced to cloud. The process resulted in long training sessions and large data files that were difficult to process and analyze. With the introduction of Cortex, PRO utilization has decreased and is now used to see the activity of electrodes as a support tool at most. It helped us figure out that the frontal lobe/electrodes are the most utilized ones when training mental commands. The PRO might play a more important role down the future if research is to be expanded upon but, the fact that EEGLAB lacks the initiative of effectively processing the data like Cortex does, proved that it is not a solution for the current task at hand. In conclusion all analysis and serial communication with microcontroller is done in the code of Cortex which is given in Appendix A.

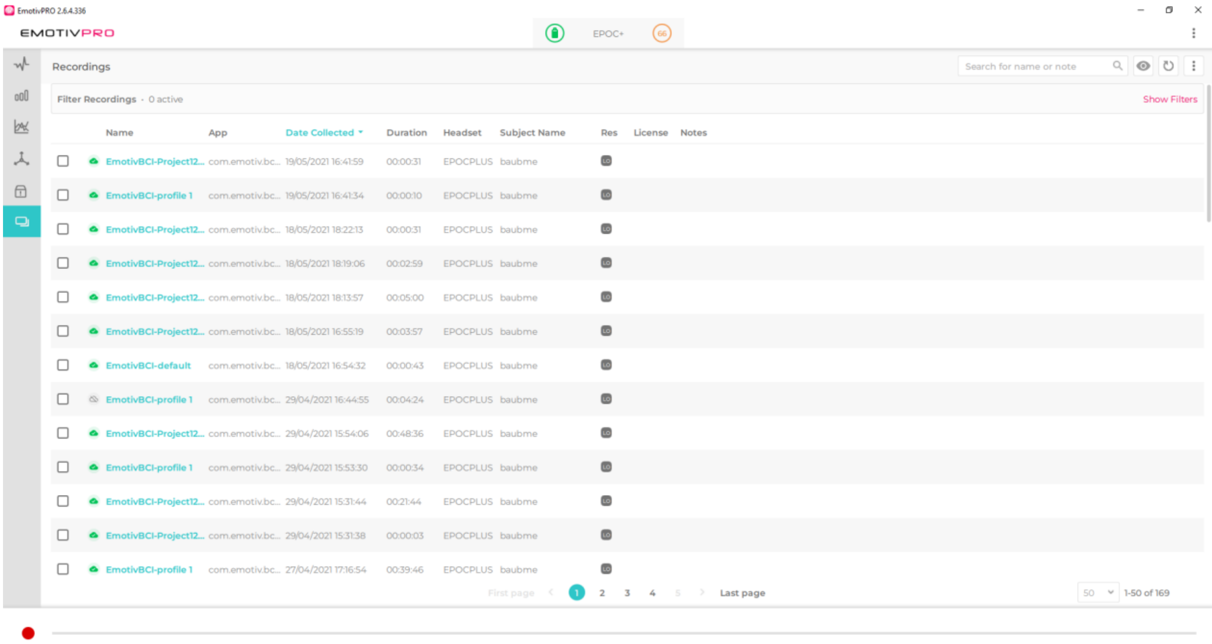


Figure 13. Emotiv PRO Recordings saved on cloud

3.1.6. Evaluation

Consistency of EEG commands is crucial for a healthy system integration and reliability of the movement. Due to that, highest number of trainings that are possible are done. However, it couldn't be reached the desired numbers due to the lockdown of nearly two weeks. During this period trails for a simulated device is carried on BCI interface. Based on simulations code is modified for serial communication during lockdown period based on the simulated training intensities.

Quality of the signals were main parameter to be concerned during trainings. This is ensured by the contact quality check on the headset and preprocessing of the signal on the software. Headset is adjusted every time when the contact quality dropped below 98%. Additional saline solution added to electrodes after a long training session if contact quality started to drop because of dryness.

Mental commands and facial expressions are categorized better when the number of training is increased. An appropriate distribution is achieved when the commands are trained in several sessions.

Achievements proves that an individual can be trained for mental commands and if the enough number of trainings achieved it can be used to control the arm when they are sent to the microcontroller with a serial port communication protocol.

3.2. Mechatronics Engineering

The application of biological signals like electroencephalographic (EEG) to control automated robotic human prosthetics is considered a breakthrough in biomechatronics which provides a better life quality for individuals with motor function loss. This technology allows the disabled to get assistance with some daily life tasks.

Our challenge for this project is to achieve almost perfect integration between the signals of EEG collected from a headset (EPOC+), with a robotic arm that is stationary in the elementary stage of the project, and able to perform normal arm movements by grabbing a glass of water.

The technologies that are planned to be used in this project are:

1. The EEG headset EPOC+ by Emotiv.
2. Emotiv software, PRO, and BCI.
3. MATLAB and EEGLAB toolbox for signal processing and connecting with Arduino.
4. Arduino Uno for controlling the robotic arm.
5. Five servomotors.
6. PLA structured robotic arm.

The plan is to get the signal from the headset and write the code of the robotic arm according to the type of input of the signal, then the specific signals made by the mental commands will determine the type and direction of the motion that the robotic arm will perform.

The robotic arm parts will mostly be 3D printed and assembled in a similar shape to normal human arm.

3.2.1. Requirements

There are some functional and performance requirements that robotic arm should satisfy.

Main function of the arm will be to grab a given the object. It should also hold the object while continue the motion and release when the user sent the appropriate mental commands. To achieve this function, it should have enough DOF to move flexibly.

Flexibility of the movement will determine how well the robotic arm perform the tasks. In addition to that, accuracy of the movement and duration of the task determines the performance of the robotic arm as well. Besides, strength of the grab needs to be arranged by the properties of the object.

Lastly, safety is an important parameter for the well-functioning of the arm both to protect arm and its surroundings against electrical and mechanical damage. To ensure that, angular limitations for servo motors are written on microcontroller code and a current breaker switch is used in the microcontroller.

3.2.2. Technologies and methods

The comparison between the major projects currently existing as supporting wearable devices and assistive prostheses could be seen here in this segment. The contrast highlights the differences between the modes and approaches employed in certain projects.

A team in Hong Kong Polytechnic University developed a mobile set of exoskeleton robotic arms which can be used or transported anywhere to support people with strokes to open or close their hands as shown in Figure 14 [27]. It acts in an active mode and has 2 degrees of freedom at the MCP and PIP for each finger. The robotic hand comprises of Velcro bands that maintain the arm in place, 5 linear finger actuators and a frame for palm reinforcement. Finger assemblies can be used to provide strength and flexibility for the finger and extension. It is intended for use in various finger lengths. There is also an integrated controller that performs different functions on the robot arm and tracks the EMG signals (sensor) used in the hand closing and opening operation. In addition, the set provides a portable remotely controlled system that allows the trainer to customize various training strategies and choose from them.

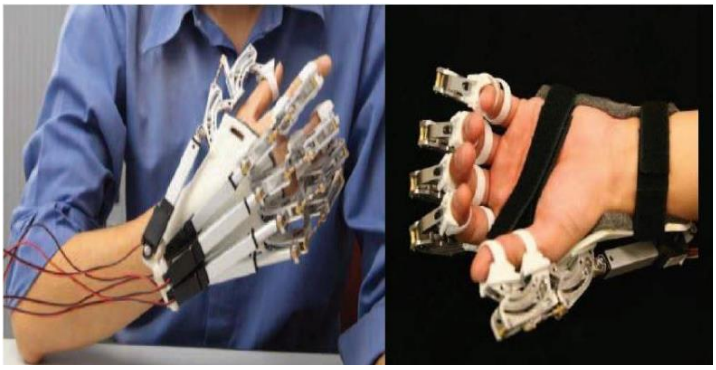


Figure 14. Hand Exoskeleton and Prototype

In University of Technology in Sydney a hand exoskeleton has been planned and built for post-stroke rehabilitation [28]. It is shown in Figure 15. Centered on the action of the equivalent parameters of the right hand (healthy), the machine achieves maximum flexion/extension movement of the five left hand fingers (impeded hand) . The machine was about 2 kg, and, because of its low weight, they preferred aluminum. There are 15 degrees of freedom in the hand exoskeleton (DOFs). Nevertheless, abduction/adduction motion could not be accomplished by the exoskeleton; thus, further research remains to be done on the project.

Figure 15. Recovery Robotic Arm

As illustrated in Fig. 16 and 17, a group from Kyushu University suggested a hand exoskeleton by utilizing a three-layered slipping spring mechanism [29]. The purpose of this article is to describe an easy-to-use, lightweight, and portable device, such that the under-actuated system is used to minimize the weight and size of the device by reducing the number of actuators to just one. The bionic hand weights 320 g. For durable gripping, each finger has three DOFs, which are flexion/extension, but the thumb is static. As a result, the 3 DOFs are controlled by one actuator. Therefore, the 4 fingers can operate simultaneously. The three-layered slipping spring comprised of 3 springs separated into inner (Si), middle (Sc) and external (So) springs and rigid bodies into portions of the tip (Rt), inner (Ri) and outer (Ro). The system allows flexion motion to be conducted.

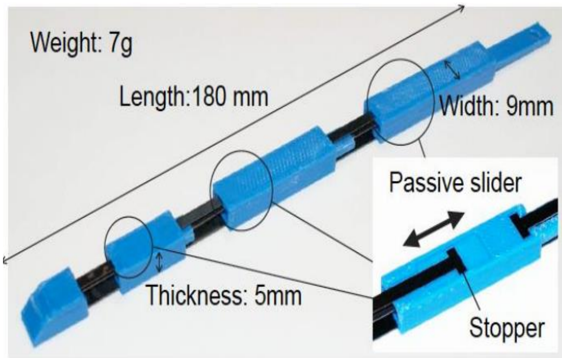


Figure 17. Mechanism of Three-layered Slipping Spring

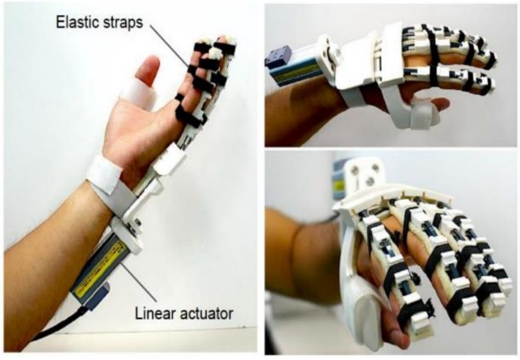


Figure 16. Hand Exoskeleton

A hand supportive exoskeleton running in active mode [30] has been developed by the University of Salford. The physiotherapy activities are done using a virtual reality trainer. This helps the patient to perform the recovery activities by exciting immersive games. This exoskeleton conveniently enables hand operation, observation, and recording. The exoskeleton possesses 7 functional DOFs. The product can be seen in Figure 18 below:

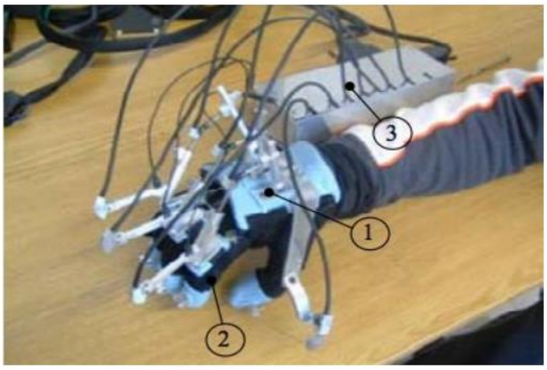


Figure 18. Supportive Hand

In 2016, a group at Twente University propose a simple, portable exoskeleton of a hand for support and recovery. They have designed it to be inexpensive, wearable, and compact for active rehabilitation as shown in Figure 19 [31]. They have used EMG system for this side, operated by the muscles. This could be for various sizes. They used a 3D printer for damaged elements of the exoskeleton. The purpose of this exoskeleton robotic hand is to help users with hand disorders.

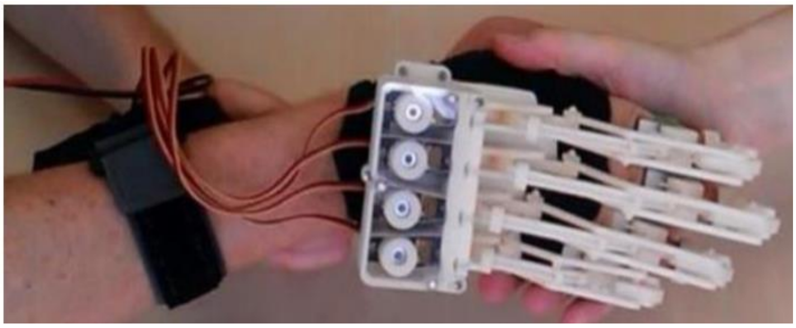


Figure 19. Supportive Recovery Exoskeleton

The robotic hand they used was taken from Thingiverse database (2012): STL Files were then loaded into HP 3D environmental software and fabricated by use of an HP DesignJet 3D printer in ABS and they have used Arduino UNO for the microcontroller of the robotic arm [32]. The design is provided in Figure 20.

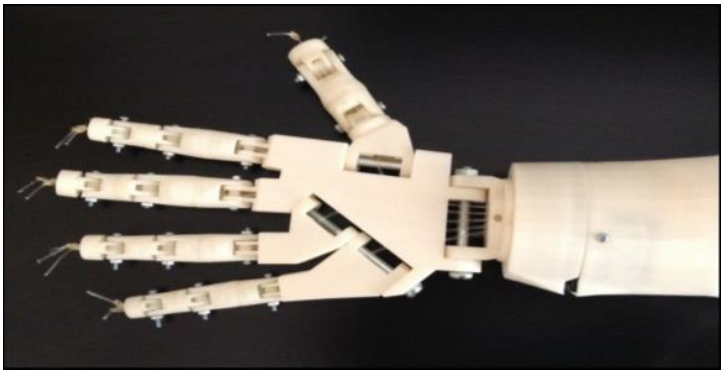


Figure 20. The 3D Printed 5 DOF Prosthetic Hand

Department of Electronics & Communication Engineering, Govt. Engineering College, have developed more similar project to ours, by using the same EEG headset. They have used Arduino UNO as well for the microcontroller. The stages of their process are similar to ours, but the robotic arm design is less detailed and with 3 DOF [33].

Virtual reality is used for the output from the signal processing and translating algorithm phase in the robotic arm, a real-life machine with the aid of HYPERTERMINAL software interface. Their workflow is provided in Figure 21 as following:

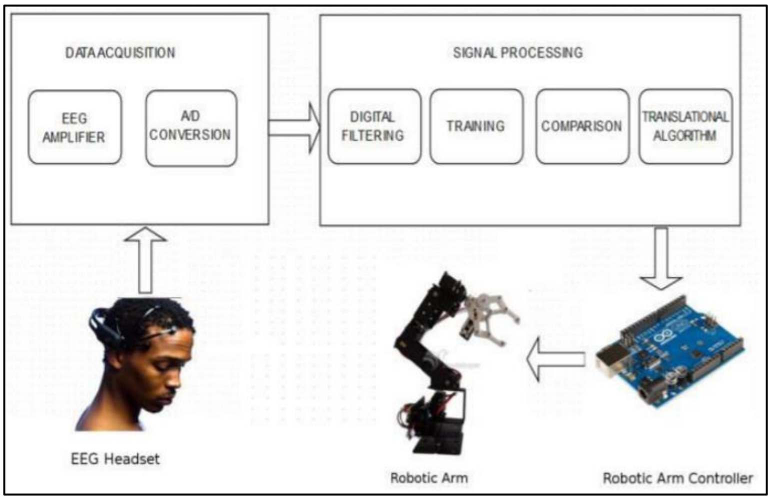


Figure 21. Block Diagram for System Implementation

The AREXX Engineering Educational Robot - Robot Arm Pro V3 was the robotic arm used. This robotic arm has been operated with an ATMEGA64 microcontroller, contains 6 Servo motors, and is fitted with programmable software of its own [34]. It was simpler to operate the robot from the Input voltage due to the integration of a voltage regulator, which makes this the optimal alternative when powering the arm. The robot design can be found in Figure 22.

Figure 22. The AREXX Engineering Educational Robot - Robot Arm Pro V3

3.2.3. Conceptualization

Conceptual design is a significant process for a project complex as this. Although it may change during further operations, it is a crucial step that identifies the 3D modeling goals and constitutes the main idea of the physical architecture. As a matter of fact, for this project, the intention of this process is that how well it converges to the human limb functions. Furthermore, the limbs that should be designed were hand, forearm, and upper arm.

For the hand, there were various design solutions to be considered. Since hand is the most detailed limb for this project, the options were considered thoroughly. Figure 23 shows the degrees of freedom structure of a human hand. Every finger has four degrees of freedom. However, it would be very difficult and costly to obtain that. As can be seen in the Figure 24, there is this hand design of the InMoov project and in this design, there is only one degree of freedom for each finger.

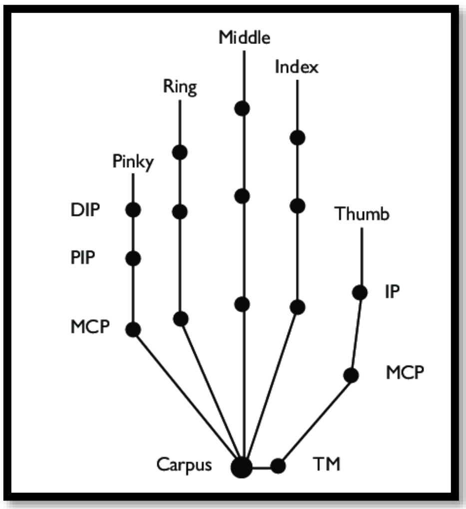


Figure 23. Kinematic Structure of Human Hand

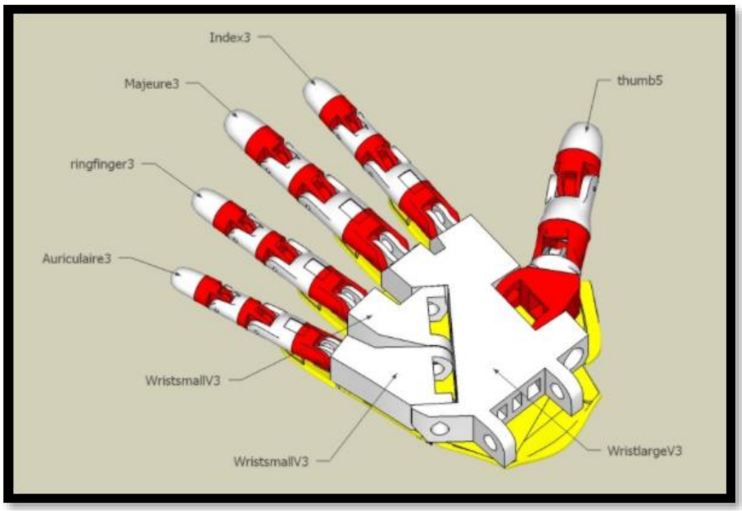


Figure 24. InMoov project Hand Design

Another issue had to be considered was the determination of the tendons. There are two options: motors can rotate one string per finger dynamically, or motors can rotate the strings statically by twisting two strings with extra apparatus. However, the second option would be very expensive and would require more time for trials and errors. Hence, the first option was chosen where the strings were guided by a hole going through every part of the finger and knotted at the tips of the fingers. The above-mentioned holes are given in the Figure 25, the palm and the fingertip cross-section can be seen in that figure.

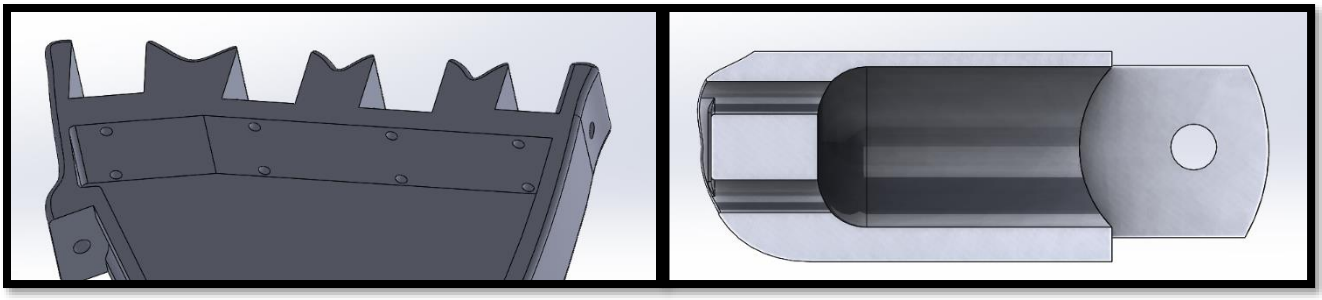


Figure 25. Holes to Guide the Strings, the Palm (left) and the Fingertip (right)

There is also another design at InMoov Project, given in Figure 27, this design is more similar design to the one which will be provided in this project. As can be inspected in this figure, this design does not require further parts for the fingers, which is more desirable on account of given limited budget and time.

Throughout the literature review phase, the number of simultaneously used fingers during the exploration phase has attracted the attention in the article named "A Systematic Comparison of Perceptual Performance in Softness Discrimination with Different Fingers" [35]. In the Figure 26 simultaneously used fingers can be seen, where the order is thumb, index, middle, ring, and pinky, respectively. So, one can make inference that the thumb, index, and middle fingers are the necessary fingers for defining objects. So, even utilizing a single motor for pinky and ring fingers can be considered but this will be discussed later.

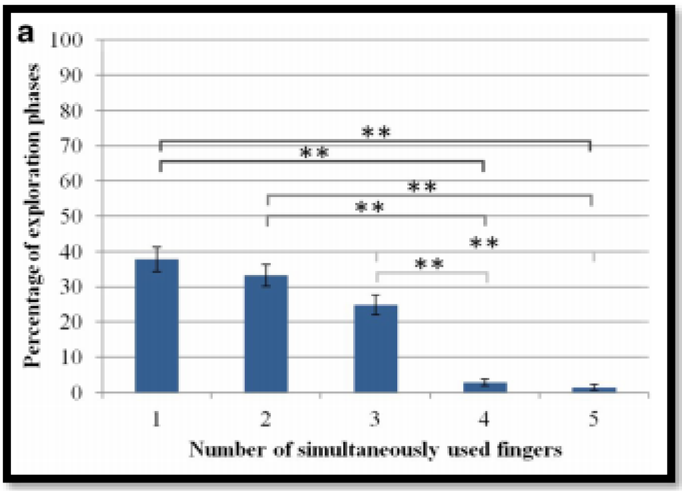


Figure 26. Finger Numbers from Thumb to Pinky

Figure 27. Another Hand Design at InMoov

So, all things considered, as stated in the Figure 28 an early conceptual design of the hand was specified even by evaluating the above issues. As might expected, this conceptual design was used for 3D designing in a CAD program.

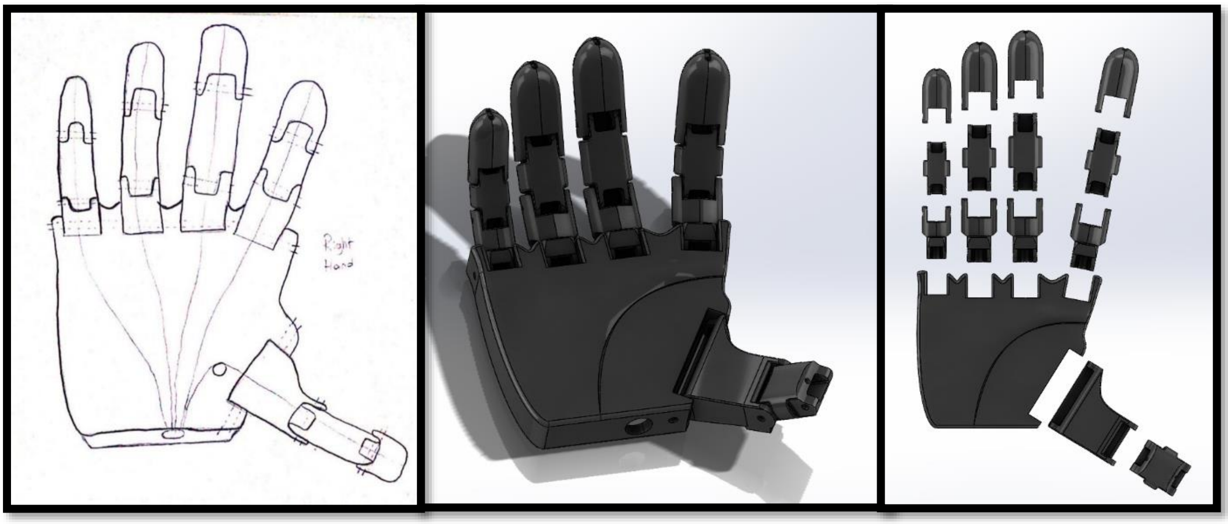


Figure 28. CAD Design in-process

For the forearm part, there are two options as well: using a mechanism at the wrist part to provide rolling motion to the hand or putting this mechanism to the elbow part. Both options have their pros and cons. Namely, first alternative would provide more sensitivity, but it may result in tangling of the strings. Second alternative would solve that issue, but the hand would lack precision. It is also beneficial to mention that inside of the forearm should contain holes for the servo motors to be mounted and this can be seen in the Figure 29. For the elbow pitch a single motor will be used and two motors will be used for shoulder pitch and roll.

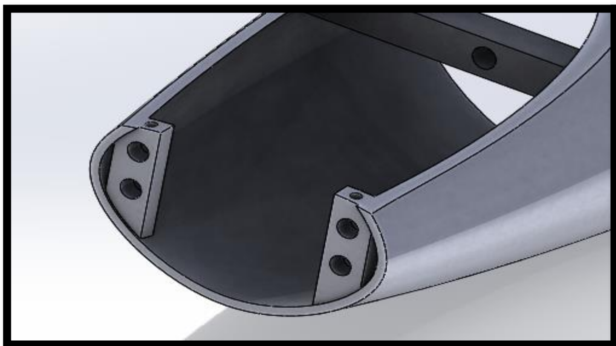


Figure 29. Forearm Servo Motor Mounting Area and the CAD Design in-process

For the forearm yaw, there were two options. Either a bearing would be utilized, or a different mechanism could be implemented. Due to many reasons regarding the assembly, designing a new mechanism has been chosen. This mechanism should make the yaw motion attainable while providing stability. A scribble of an early design of the forearm yaw bearing can be seen in the Figure 30. The same mechanism has been utilized for the shoulder roll as well. The CAD model will be shown in the Physical Architecture sub-topic.

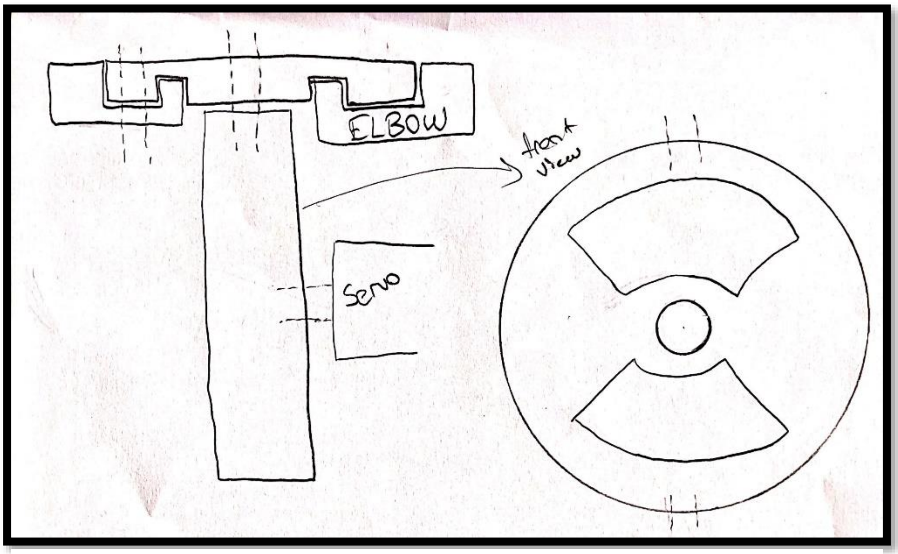


Figure 30. A Scribble of an Early Design of the Forearm Yaw Bearing Mechanism

3.2.4 Physical Architecture:

There are various ways to build this robotic arm. 3D printing method has been chosen to be able to have flexibility on what is being designed. Certainly, there are some advantages and drawbacks of 3D printing, which are summarized on Figure 31. Furthermore, fused deposition modelling (FDM) is preferred among the other 3D printing methods like SLA and SLS. The major reason is that FDM is way more budget friendly compared to other methods. It is beneficial to state that during FDM process, filament called a thermoplastic material is being utilized. Moreover, coil of a thermoplastic filament is around 120₺.

3D Printing	
PROS	CONS
Flexible design	Limited materials
Print on demand	Inaccurate prints
Low-cost	Takes long time to print

Figure 31. Pros and Cons of 3D Printing

For the physical structure, as can be seen in the Figure 33 general arm proportions are given by R. Bowns (2011) [36]. One head length is 8 inches and that is approximately 20.5 centimeters. A human hand is measured to be ¾ head length which is 15.375 cm from middle finger to wrist. As shown in the Figure 32, the CAD design for this project, that is the case as well.

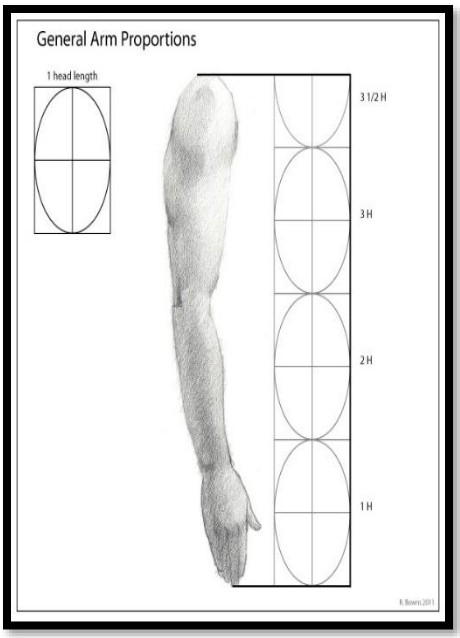


Figure 32. General Arm Proportions by Bowns

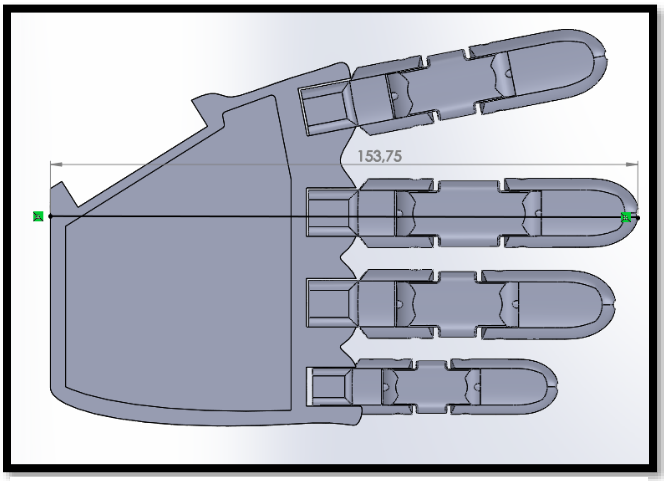


Figure 33. CAD design in-process

As can be seen in the Figure 34, these are the dimensions of the robotic arm of the Reachy project [8]. This was the chosen conceptual design of the robotic arm for this project. Also, shoulder and elbow roll mechanisms are decided to be like this since this is an optimal design.

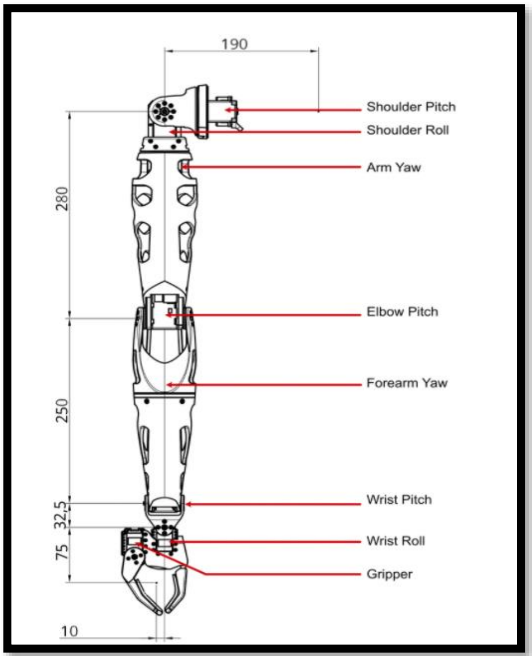


Figure 34. Physical Architecture of the Arm Courtesy of Reachy Project

As mentioned in the Conceptual Design sub-topic, strings will be used for the rotation of the fingers. In addition, springs will be utilized as the retainers of the fingers. As can be seen in the Figure 35, the servo motor will cause the fingers to rotate with the applied force by the strings attached to the servo motor horn.

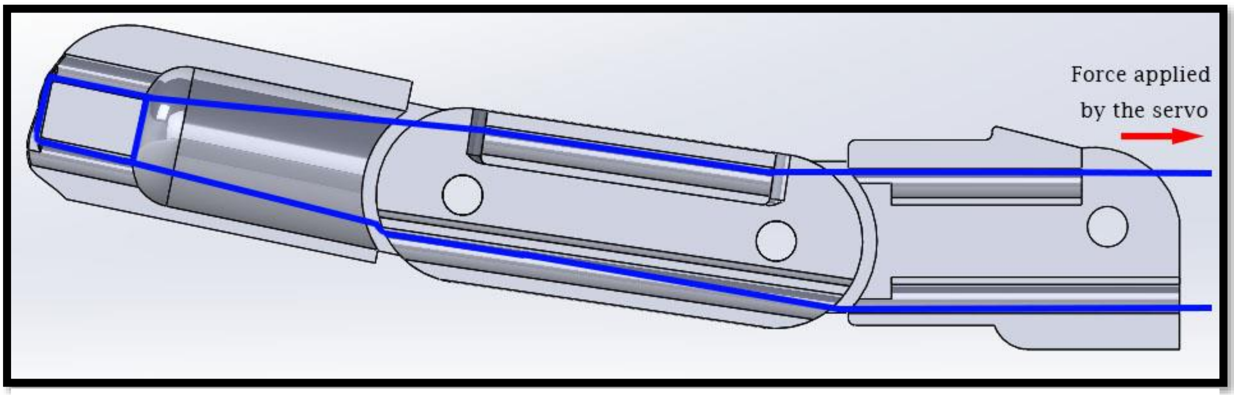


Figure 35. Strings Providing Rotation to the Fingers

For the forearm yaw, as mentioned in the Conceptualization sub-topic, it has been decided to design a bearing mechanism. In the Figure 36, the two parts that forms the bearing can be seen. The part on the left is an apparatus that connects the elbow, forearm, and the part on the right, which is a disc that forms a bridge between the servo and the apparatus. Furthermore, it has to be printed twice in order to intertwine around the disc.

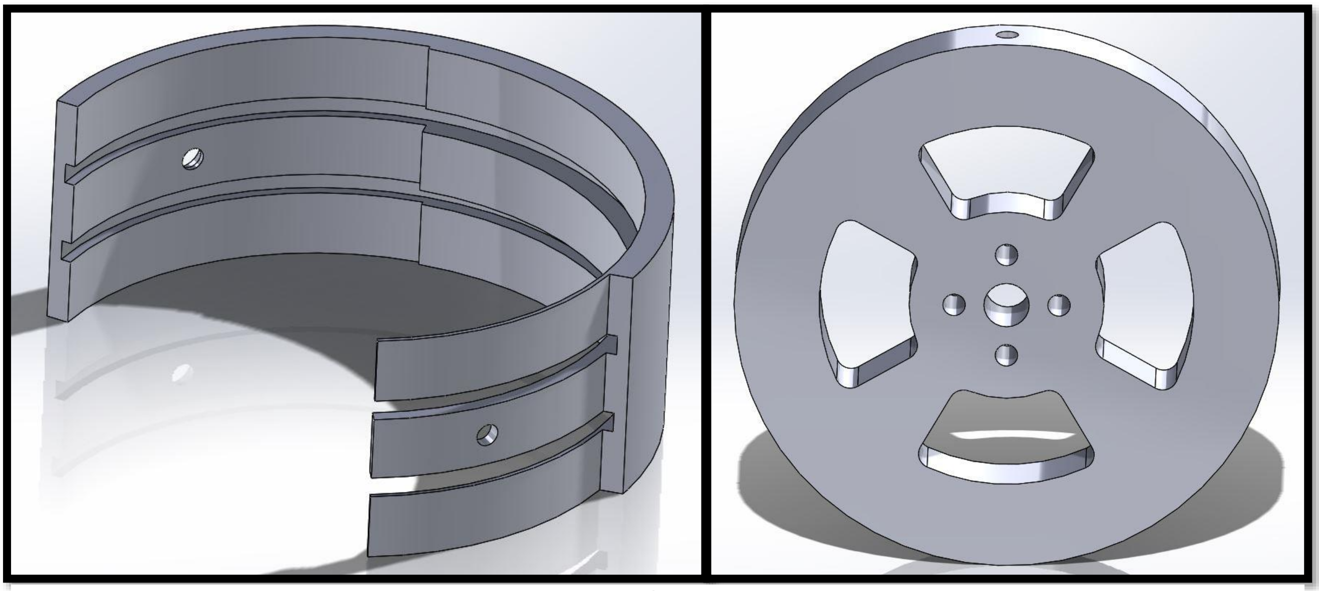


Figure 36. Apparatus and the Disc for the Forearm Yaw Bearing Mechanism

The assembly of these parts plays a crucial role for this mechanism. In the Figure 37, it can be observed that the parts that are highlighted with the blue color are attached to the forearm. Whereas the yellow parts are connected to the elbow. The green part is connected to both the disc and the servo motor, and it is called a servo head.

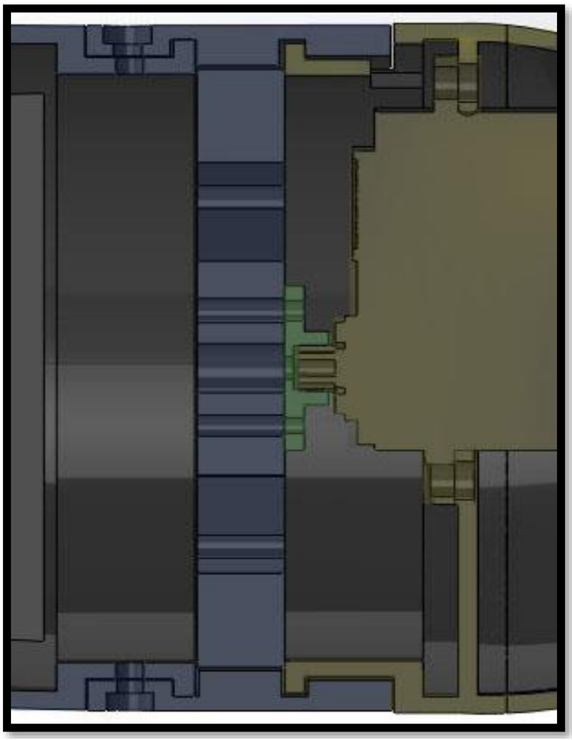


Figure 37. Cross Section of the Forearm Yaw Bearing Mechanism

The elbow part is the link between the biceps and the forearm. Also, there are two servo motors mounted on this part where one servo provides rotation for the forearm yaw and the other servo provides the elbow pitch. Figure 38 shows the elbow part with the two servo motors mounted on it.

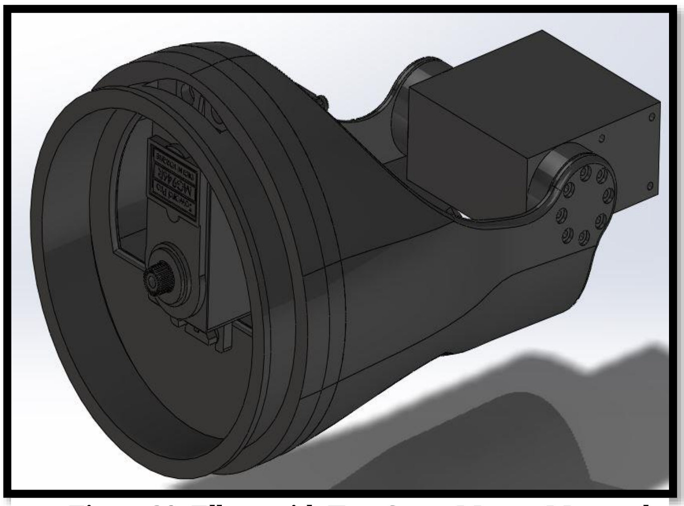


Figure 38. Elbow with Two Servo Motors Mounted

For the biceps and the shoulder, they are connected to a servo motor and the shoulder also has an additional servo motor that provides the shoulder rolling motion which is the same mechanism as the forearm yaw. So, with this part, the shoulder pitch and shoulder roll motions are achieved. In Figure 39, the joint between the shoulder and the biceps that provides the shoulder pitching motion can be seen. Also, the ending of the shoulder has a 25 mm hole so it can be attached to a metal rod.

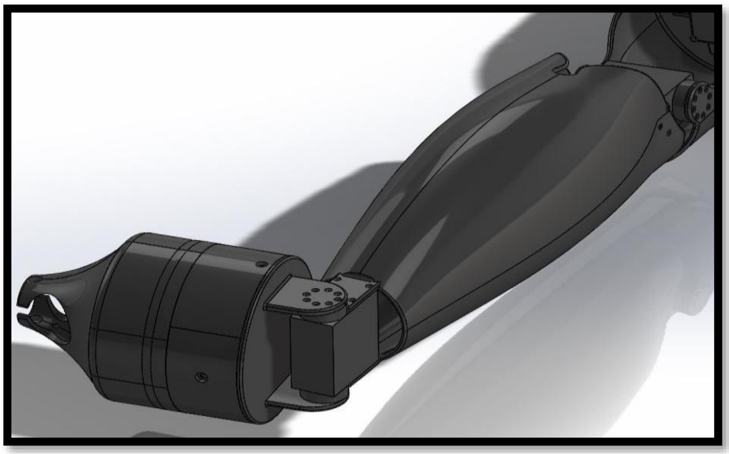


Figure 39. Biceps and the Shoulder Connection

All in all, the arm is designed to provide at least 4 degrees of freedom motion to the end effector. Moreover, the arm utilizes 3 one axis servo motors and 2 two axis servo motors in order to attain these motions. Furthermore, other than the above-mentioned servo motors, there are bolts, nuts, strings, and springs used at the assembly of this arm. In Figure 40, the final design of the arm with dimensions can be seen.

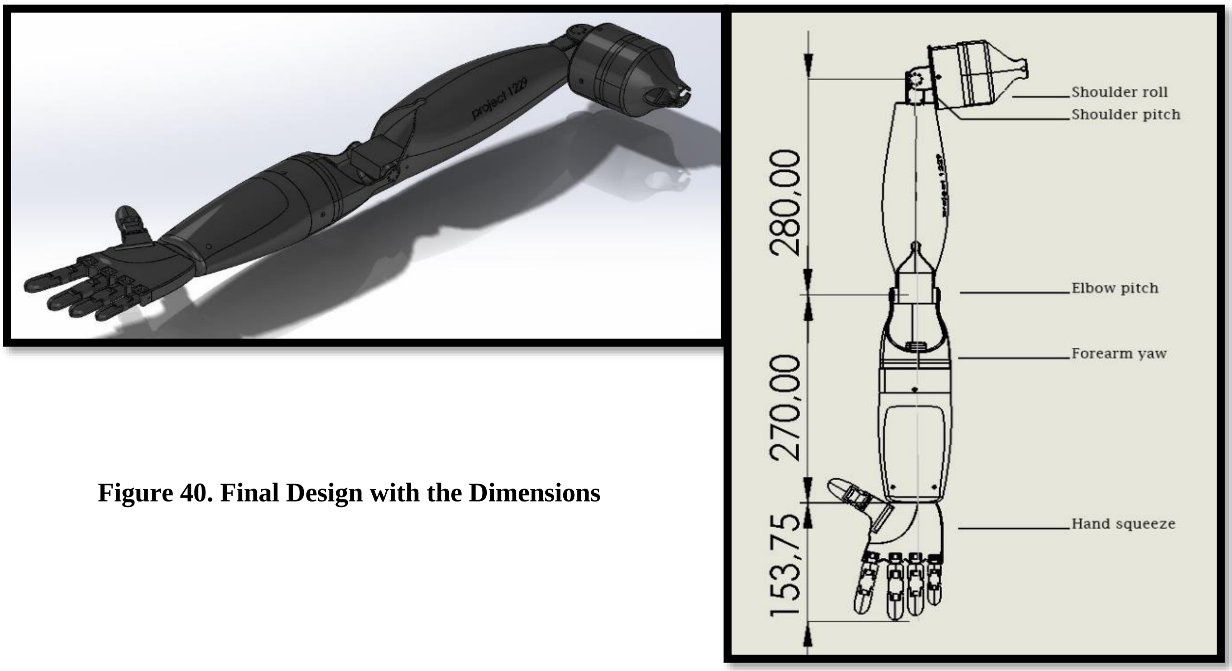


Figure 40. Final Design with the Dimensions

3.1.5. Materialization

The materialization process is where the arm comes to life. The main factor contributing to materializing the hand is using the ANYCUBIC Mega Zero V1 3D printer which is shown in Figure 41. This is a single nozzle printer with a build plate and an indirect extruder. The only downside in our case, is that we could not get our hands on a heat bed to use. If we had a heat-bed, we wouldn’t have had to reprint some of the parts such as shown in Figure 42, 43, & 44. In Figure 42, the shoulder piece detached from the printing bed; shifting the print sideways. In Fig. 43, the forearm cover detached from the printing bed, and got stuck to the printing nozzle at around 80% of the progress completion. In Fig. 44, warping occurred due the print failing to stick to the bed. While printing, due to the fact that there is not heating bed and the print being cold, the molten filament caused a warping effect in the printed part and lifted the part up of the printing bed, causing the print to fail.

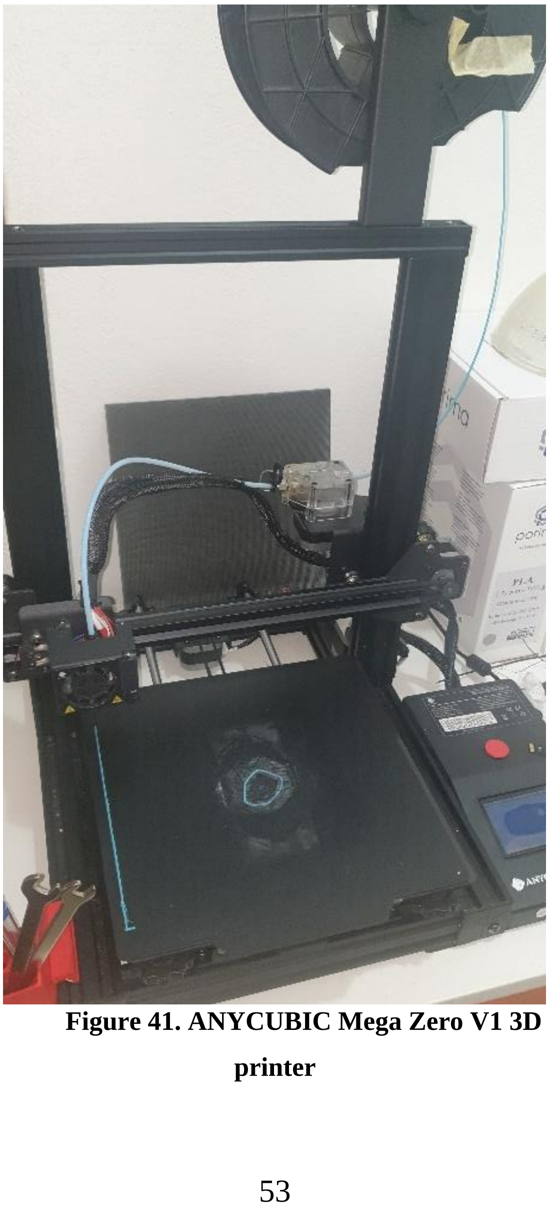


Figure 41. ANYCUBIC Mega Zero V1 3D printer

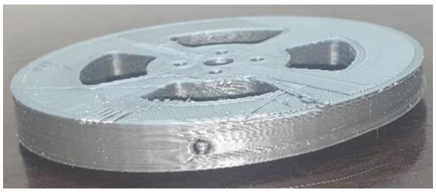


Figure 42. Rotary disk failed print

Due to all what was mentioned prior, these are some of the parts that were forced to be reprinted as they are in a beyond fixable state. Additionally, some prints failed due to wrong dimensions and change of motors (Which changes the internal dimensions of the print). Also, the printing times of different parts of the robotic arm varied between 30 minutes to 23 hours per part. Due to the mentioned prior, we tried our best to avoid reprints and save time. For example, we tried to sand off the edges of some of failed prints, as well as fixing the filament using a 3D pen. Some other issue faced while printing is electric power cut, causing the nozzle to jam. One of the problems we had faced is getting a defective part of the filament roll that was being used to print the parts. That could have been caused due to the fact that the filament being used in our prints is recycled PLA to minimize waste. Through the journey of bringing the arm to life, we stumbled upon different printing techniques and styles for optimal results and avoiding weak points. As shown in Figure 45 & 46, we found that printing in

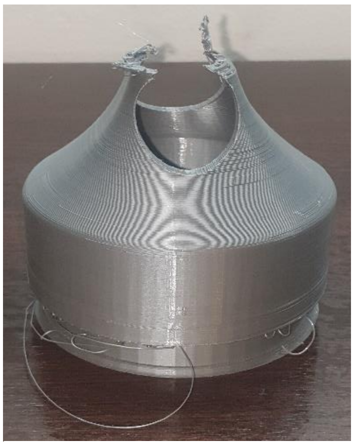


Figure 43. Shoulder Anchor failed print

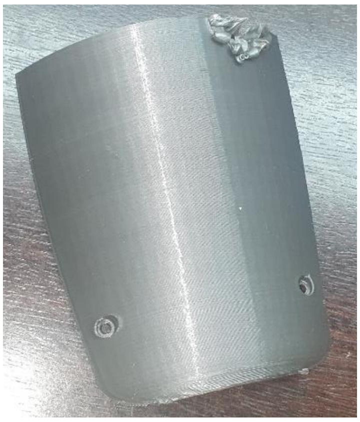


Figure 44. Forearm yaw cover failed print

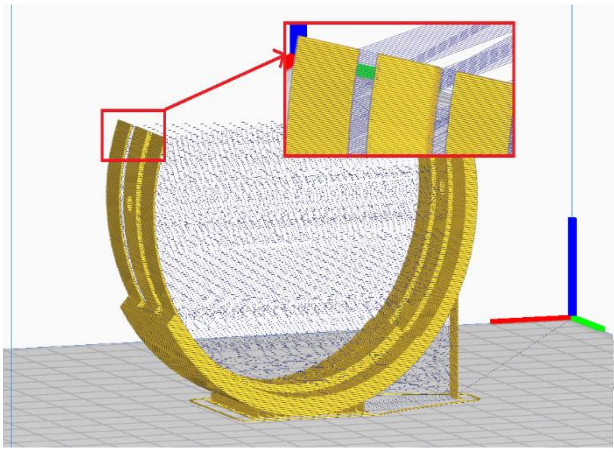


Figure 46. Rotary disk base in Cura software

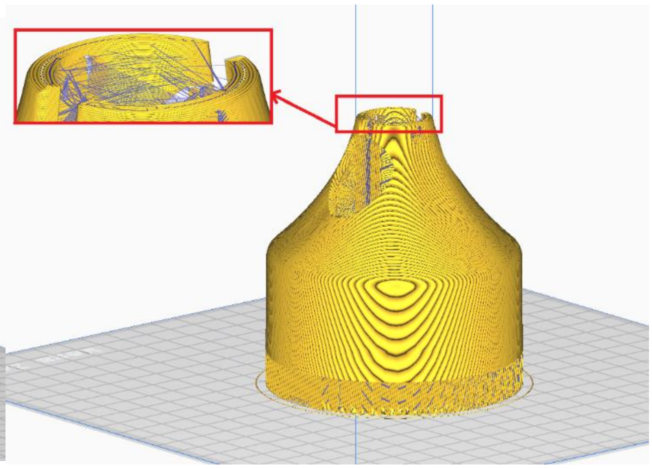


Figure 45. Shoulder Anchor in Cura software

this style leads to a weak structure in the printing layers. Hence, we had to research different printing methods for stronger structured layers, such as in Figure 47 & 48.

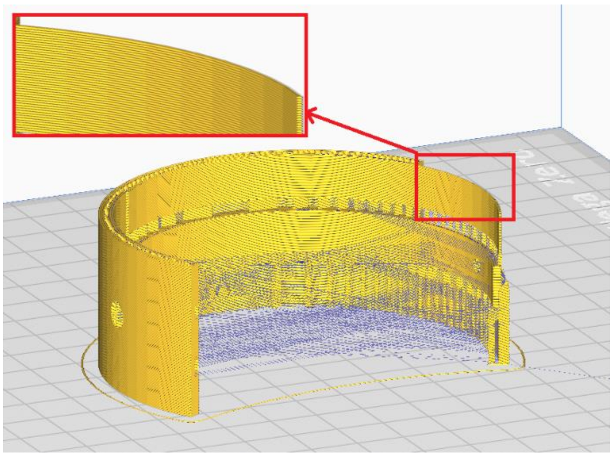


Figure 47. Strengthened printing method for the rotary disk base.

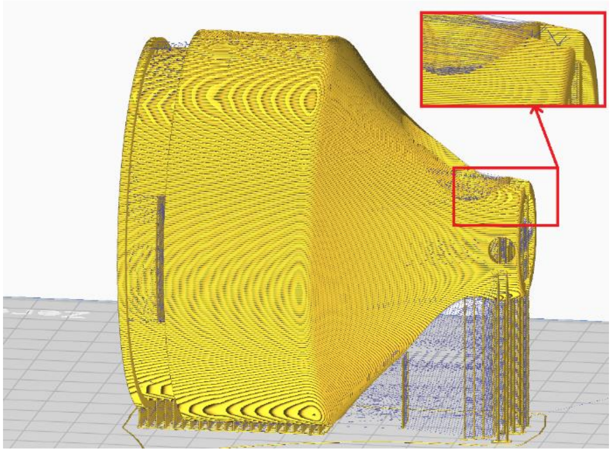


Figure 48. Strengthened printing method for shoulder anchor.

Another problem we had faced is regarding the servo motors. When we had first received the motors and installed them into our robotic arm, we noticed some slipping in the motion. After running a few basic tests, we noticed that the plastic gears of the interior of the motors themselves were defective and needed replacement. Due to the mentioned prior, we had to send the motors back to the company we bought them from and request new motors with metallic gears; avoiding any gear slipping problems, corrosion, and to increase the torque to carry more weight. In the next step, all the printed parts had been assembled together alongside the motors inside.

Figure 49. Interior of the forearm and palm

As shown in Figure 49, a simple mechanism where all the strings coming from the fingers are connected to a single pulley connected to one of the servo motors works to open and close the fist. The springs help in reopening the fist and expanding the fingers at ease. Also, the tubes don't only work as a connector of all strings (removes the branching and merges them into one tube), but also protects the filament from corrosion due to friction and the strings from wearing out when the arm is in motion.

This connection is a work in progress, and what is shown in Figure 49, is not the final look of the interior as there is still some work to be done regarding the connections to the rest of the system.

All the motors are fixed in place using the designated bolts and screws in the 3D design. Also, as shown in Figures 50 and 51, looking closely at the bicep a mismatched line in the print is visible. This misprint doesn't seem to oppose any structural issues.

Figure 51. Fully assembled arm with uncovered forearm.

Figure 50. Other side of the assembled arm.

3.2.6. Mathematical modelling

We have two revolute joints at the shoulder, one revolute joint at the Elbow, and one yaw at the forearm.

First, we assigned the global frame at the first shoulder joint responsible for the flexion extension motion with the Z axis as facing to the right of the model and the X axis facing upwards with the note of all the segments of the model are facing downwards as the initial state of the arm.

later we assigned frame number one on top of the global frame. the center of the second frame was placed on top of the first and global frame but in this case the Z2 axis is facing forward and the X2 axis is facing downwards. The rotation around Z2 axis in the 2nd frame is responsible for the abduction and adduction motion of the shoulder.

At the elbow joint we applied frame 3 with Z3 axis being the axis where the elbow performs the flexion and extension motion and the X3 pointing forward.

The forearm joint is the fourth frame, we assigned Z4 to be pointing downwards coinciding with the axis of the motor and X4 pointing forward.

The end effector has the same direction as the previous frame.

For obtaining the D-H table, we applied The Denavit-Hartenberg Steps.

Table 6. Denavit-Hartenberg table

i	a_{i-1}	α_{i-1}	d_i	θ_i
1	0	0	0	θ1
2	0	90	0	θ2
3	L1	90	D3	θ3
4	0	90	L2	θ4
e	0	0	L3	0

For link “i”, link length (ai−1) & link twist (αi −1), link offset (di) & joint angle (θi).

- (α_{i-1}) is the angle between (z_{i-1}) & (z_i) measured about (x_{i-1})
- (a_{i-1}) is the distance between the (o_{i-1}) to the intersection of (x_{i-1}) & (z_i) measured along (x_{i-1}) or the distance between (z_{i-1}) & (z_i) measured along (x_{i-1})
- (d_i) is the distance between the intersection of (x_{i-1}) & (z_i) to (o_i) measured along (z_i) or the distance between (x_{i-1}) & (x_i) measured along (z_i)
- (θ_i) is the angle between the axes (x_{i-1}) & (x_i) measured about (z_i)

Figure 52.Denavit-Hartenberg table components Assignments

The link transformation matrix under the modified D-H convention is:

$$
{}^{i-1}_{\ \ i}\mathbf{A} = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1} d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

Figure 53. Transformation Matrix

Then we multiply all the transformation matrices of all links until we reach the final transformation matrix which is ${}^{0}A_{e}$.

For the inverse kinematics:

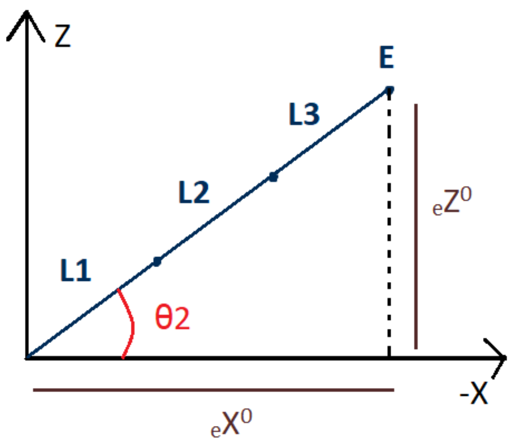


Figure 54. Front view

We applied the geometric approach and in the Front view of the arm, we can consider the three links that we have a X and Z axis of the global frame in the figure attached below.

from this side we can calculate the angle θ2 geometrically as it is equal to $\theta_2 = tan^{-1}(\frac{{}^0_eZ}{{}^0_eX})$.

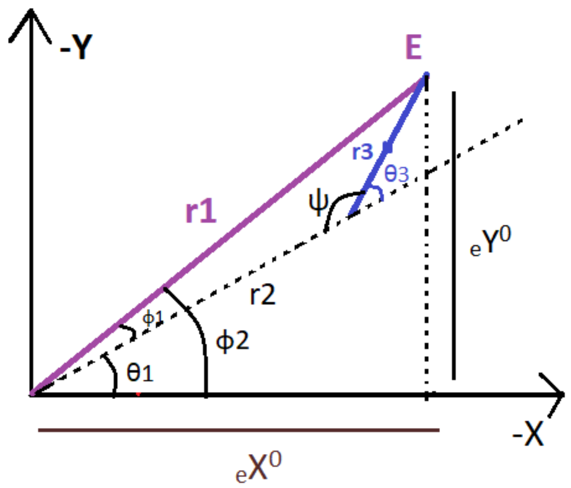


Figure 55. Side view

From the side view, we can geometrically calculate the rest of the unknown variables and angles.

By using regular law of cosines and Pythagorean in the figure attached below and we obtain the following equations.

$$r_2 = L_1$$

$$r_3 = L_2 + L_3$$

$$r_1 = \sqrt{({}^0_eX)^2 + ({}^0_eY)^2}$$

$$\varphi_2 = tan^{-1}(\frac{{}^0_eY}{{}^0_eX})$$

$$\psi = \frac{cos^{-1}\,(r_1{}^2 - r_3{}^2 - r_2{}^2)}{(-2r_3r_2)}$$

$$r_3{}^2 = r_2{}^2 + r_1{}^2 - 2r_1r_2\cos(\theta_1)$$

$$\theta_1 = \varphi_2 - \varphi_1$$

$$\cos(\varphi_1) = \frac{(-r_1{}^2 + r_3{}^2 - r_2{}^2)}{(-2r_1r_2)}$$

$$\theta_3 = 180 - \psi$$

Motors torque requirements: The motors being used in our prototype are LDX-218 Double Axis Servo Motor and MG945 Servo. The LDX-218 Double Axis Servo Motor has a torque of:

$$17\ kg.cm\ @7.4\ V, and\ 15\ kg.cm\ @6.6\ V.$$

The MG945 servo has a torque of:

$$10\ kg.cm\ @4.8\ V, and\ 12\ kg.cm\ @6\ V.$$

The torque of the servo at the shoulder joint required is according to the following calculation: The arm length being carried by the sLDX-218 servo is 70.375 cm. According to the torque formula:

$$\tau = rFsin(\theta)$$

To calculate the force acting on the object, first we need to take the weight of the arm and multiply it by the gravity constant “g” as:

$$F = ma$$

$$g = 9.80665\ m.s^{-2}$$

Substituting in the mass of our 3D printed arm provided by the Cura 3D printing program is + the weight of the motors:

$$m = 626 + 3(55) + 2(60) = 911\ g$$

We get the following result:

$$F = (0.911\ kg)(9.80665\ m.s^{-2}) = 8.9338515\ N$$

Assuming we are working at the hardest angle of 90°:

$$\tau = (70.375\ cm)(8.9338515\ N)(1) = 64.1116\ kg.cm$$

$\tau = 64.1116\ kg.cm$ is the approximate torque required for the motor at the shoulder point to be able to move the arm with ease.

3.2.6.1. Mathematical modelling notes and observations

Unfortunately, due to the ongoing pandemic, and the availability of motors in Turkey during such a difficult time, we had to settle for the strongest affordable motors we were able to find. The motor we are using at the shoulder join only have 17 kg.cm torque capability, which is almost 3.7x less than the required torque. On the other hand, the other motors (MG945) at the

elbow, forearm yaw, and grip operate with ease and are able to carry their load.
Due to the approximate 3.7x torque missing, the motors are occasionally able to lift and operate the arm at slow speeds and for short durations of time.

3.2.7. Power management

The power needed to power our system is fairly simple and straight forward. The LDX-218 Double Axis Servo Motor is operating at 7.4 V for maximum torque at maximum of 1 Amp, whereas the MG945 servo is operating at 5 V as maximum torque is not needed at a maximum of 1.2 Amp. We are using a 12 V- and 40-Amp power supply fed into a 5 V voltage regulator for the MG945 motors, and a 7.5 V regulator for the Double Axis Servo Motors. The motors' signal wires are 0.2 mm copper wires, whereas the power wires are 1.5 mm copper wires. The Arduino Uno is also powered from the same 5 V voltage regulator providing the motors, as the regulator can handle up to 20 Amps of load.

3.2.8. Mechatronics sub-system finalization

To simulate the brain wave signals as an input, we used a PS2 wireless controller for the finalization of this sub-system. The PS2 Controller uses a receiver chip that connects to the Arduino Uno directly, allowing the flexible use of the input commands of the wireless controller.

Figure 56. The PS2 controller with the receiver

The receiver connects to the Arduino is connected to the Arduino in the following matter:

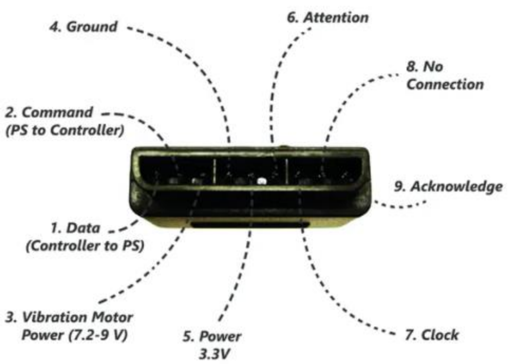


Figure 57. Port names and specifications

The numbering beside each port corresponds to the numbers that use color coded wires in the following Arduino Uno figure.

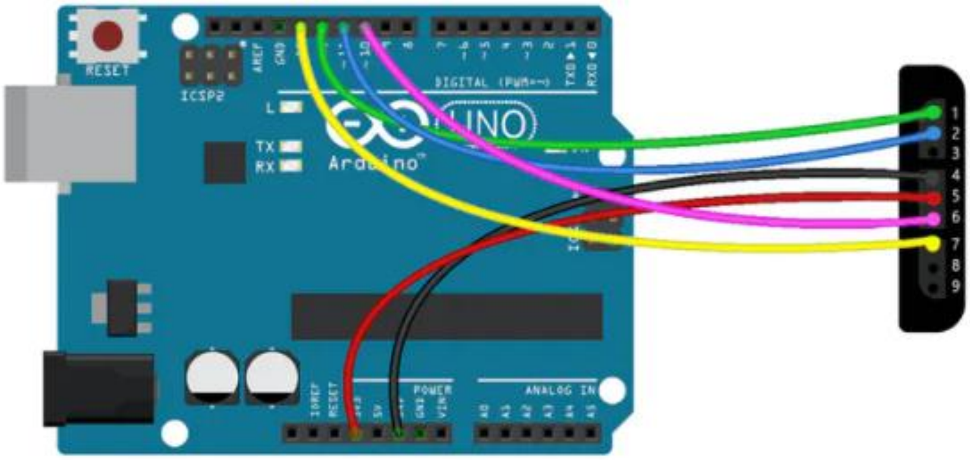


Figure 58. Arduino Uno to receiver connection

The code for the PS2 controller controlling the robotic arm can be found in APPENDIX A.

The idea behind using a PS2 wireless controller is to utilize the ability of the PS2 controller to produce many different continuous streams of analog/digital signals. This replaces the signals received the from the headset and serves a similar purpose.

```
int SY = 60;       //Shoulder Y counter. Controller commands: (Front & Down Arrows)
int SX = 90;       //Shoulder X counter                       (Left & Right Arrows)
int FAY = 60;      //Forearm yaw counter                      (Square & Circle)
int E = 60;        //Elbow counter                            (Triangle & X)
int F = 60;        //Fist counter                             (L2 & R2)
```

The previous code snippet demonstrates which button is controlling which motion. This allows the full control of the arm using the controller with ease.

3.2.9. Evaluation

Since this is a robotic arm, verifying the model requires its materialization as a prototype. After assembling the 3D printed arm with its motors and microcontroller, the first thing to test is the code of the robotic arm. So, first phase is testing the code functionality with the arm, by forcing an input instead of receiving a signal from the EEG headset; ensuring code viability. After the code is verified to achieve the desired motion, phase two starts which means the input of the robotic arm is acquired from the data sent from the headset through the microcontroller. Also, since we are dealing with brainwave commands, this process is going to mainly rely on trial and error, while tweaking everything (Code, data, and hardware) simultaneously as the test results show. Grip force is going to be tested by grabbing various objects with different forces, and measuring them using force sensors, and tweaking the force according to viability.

4. SUMMARY AND CONCLUSION

Motor function loss creates a huge drawback for people who are experiencing it, especially on their daily tasks. To reverse its negative impact, a robotic arm design by employing EEG signals is described on the scope of this project. Identification of the problem is detailly discussed to clarify the needs of the project from different departments. Several works form the literature are scanned to get inspired by them and to apply the best solution possible when all the design limitations are considered. Workflow plan is created according to the challenges on both biological and mechanical side. Responsibilities of each team member is distributed among their interests and talents. Constraints on different aspects such as time, budget and implementation are represented to set a realistic goal and achieve it on time with optimal design. Environmental, ethical, social, and economical parameters are followed on each step especially on material selection and 3D design processes. Required standards and ethical codes are followed in each step of the project. An architecture between headset and apps of Emotiv is designed and PRO; BCI apps are employed with Cortex on Python environment. A serial port connection protocol is written in the send the commands from Python to Arduino. Commands for the movements of the arm is coded on Arduino environment. Parts of the arm are designed by using CAD and produced by 3D printing. Several servo motors are used to achieve correct angles and 4 DOF. Design is created according to mathematical modelling. Electrical and mechanical safety is achieved by insulation and limited angles.

This project demonstrates how an individual can be trained to confidently perform a set of mental commands and how this can be used in the control of a robotic arm.

In conclusion, result of this capstone project will bring a better understanding on EEG analysis and construction of the suitable robotic arm by employing analyzed signal. It is expected that results of this project will be beneficial in future for bigger scale applications such as wearable arm supports or exoskeletons for different regions of body with motor function loss.

REFERENCES

[1] W. Health Organization, *WORLD REPORT ON DISABILITY WHO Library Cataloguing-in-Publication Data*. 2011.

[2] “Code of Ethics -IEEE.” [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 08-Jan-2021].

[3] “Ethics & EMBS – EMBS.” [Online]. Available: https://www.embs.org/ethics/. [Accessed: 08-Jan-2021].

[4] “Industrial Robot Safety Standards and Guidelines.” [Online]. Available: https://www.usabotics.com/osha-safety-standards.htm. [Accessed: 30-Dec-2020].

[5] B. Hellingsworth, P. Hall, and H. Anderson, “Quality management principles,” *High. Natl. Comput.*, pp. 313–390, 2020.

[6] “Regulatory Compliance - EPOC+.” [Online]. Available: https://emotiv.gitbook.io/epoc-user-manual/introduction-1/regulatory-compliance. [Accessed: 18-Dec-2020].

[7] “Build yours – InMoov.” [Online]. Available: https://inmoov.fr/build-yours/. [Accessed: 07-Jan-2021].

[8] A. Pitti *et al.*, “Reachy, a 3D-Printed Human-Like Robotic Arm as a Testbed for Human-Robot Control Strategies,” *Front. Neurorobotics | www.frontiersin.org*, vol. 1, p. 65, 2019.

[9] J. St. Germain, “Neuroprosthetics: An Investigation into Utilizing EEG Brain Waves to Control a Robotic Arm,” Boston College, 2015.

[10] T. Beyrouthy, S. K. Al Kork, J. A. Korbane, and A. Abdulmonem, “EEG Mind controlled Smart Prosthetic Arm,” *2016 IEEE Int. Conf. Emerg. Technol. Innov. Bus. Pract. Transform. Soc. EmergiTech 2016*, pp. 404–409, 2016.

[11] A. Delorme and S. Makeig, “EEGLAB: An open source toolbox for analysis of single-trial EEG dynamics including independent component analysis,” *J. Neurosci. Methods*, vol. 134, no. 1, pp. 9–21, 2004.

[12] B. Jake St Germain, “Neuroprosthetics: An Investigation into Utilizing EEG Brain Waves to Control a Robotic Arm,” Boston College, 2015.

[13] “EPOC Flex - 32-Channel Wireless EEG Device | EMOTIV.” [Online]. Available: https://www.emotiv.com/epoc-flex/#flex-kit-options. [Accessed: 16-Dec-2020].

[14] A. Vallabhaneni, T. Wang, and B. He, “Brain—Computer Interface,” in *Neural Engineering*, Springer US, 2007, pp. 85–121.

[15] "Emotiv BCI - Built for Insight and Epoc+ headsets | EMOTIV." [Online]. Available: https://www.emotiv.com/emotiv-bci/. [Accessed: 17-Dec-2020].

[16] "EMOTIVPRO | The Most Advanced EEG Software | EMOTIV." [Online]. Available: https://www.emotiv.com/emotivpro/. [Accessed: 17-Dec-2020].

[17] M. Venkata Suresh Kumar and B. Neelima, "A Portable Wireless Head Movement Controlled Human-Computer Interface for People with Disabilities," *Int. J. Adv. Res. Electr. Electron. Instrum. Eng. (An ISO*, vol. 3297, no. 7, pp. 10477–10484, 2014.

[18] P. Gargava, K. Sindwani, and S. Soman, "Controlling an arduino robot using Brain Computer Interface," *Proc. - 2014 3rd Int. Conf. Reliab. Infocom Technol. Optim. Trends Futur. Dir. ICRITO 2014*, pp. 3–7, 2015.

[19] D. S. Benitez, S. Toscano, and A. Silva, "On the use of the Emotiv EPOC neuroheadset as a low cost alternative for EEG signal acquisition," *2016 IEEE Colomb. Conf. Commun. Comput. COLCOM 2016 - Conf. Proc.*, pp. 4–9, 2016.

[20] M. Fatourechi, A. Bashashati, R. K. Ward, and G. E. Birch, "EMG and EOG artifacts in brain computer interface systems: A survey," *Clinical Neurophysiology*, vol. 118, no. 3. Elsevier, pp. 480–494, 01-Mar-2007.

[21] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan, "Brain-computer interfaces for communication and control," *Clinical Neurophysiology*, vol. 113, no. 6. Elsevier, pp. 767–791, 01-Jun-2002.

[22] "How accurate is your detection? | EMOTIV." [Online]. Available: https://www.emotiv.com/knowledge-base/how-accurate-is-your-detection/. [Accessed: 08-Jan-2021].

[23] S. Shefelbine, J. Clarkson, R. Farmer, and S. Eason, *GOOD DESIGN PRACTICE FOR MEDICAL DEVICES AND EQUIPMENT – REQUIREMENTS CAPTUREDGE ENGINEERING DESIGN CENTRE*. Cambridge: University of Cambridge Engineering Design Centre, 2002.

[24] B. John, "FAST Diagrames: The Foundation for Creating Effective Function Models," in *Trizcon 2011*, 2011, pp. 1–10.

[25] "Cortex API." [Online]. Available: https://emotiv.gitbook.io/cortex-api/. [Accessed: 21-May-2021].

[26] "NodeRED Toolbox for Emotiv BCI." [Online]. Available: https://emotiv.gitbook.io/emotivbci-node-red-toolbox/. [Accessed: 02-May-2021].

[27] N. S. K. Ho *et al.*, “An EMG-driven exoskeleton hand robotic training device on chronic stroke subjects: Task training system for stroke rehabilitation,” *IEEE Int. Conf. Rehabil. Robot.*, 2011.

[28] A. Rahman and A. Al-Jumaily, “Design and Development of a Bilateral Therapeutic Hand Device for Stroke Rehabilitation,” 2013.

[29] J. Arata, K. Ohmoto, R. Gassert, O. Lambercy, H. Fujimoto, and I. Wada, “A new hand exoskeleton device for rehabilitation using a three-layered sliding spring mechanism,” in *Proceedings - IEEE International Conference on Robotics and Automation*, 2013, pp. 3902–3907.

[30] I. Sarakoglou, N. G. Tsagarakis, and D. G. Caldwell, “Occupational and physical therapy using a hand exoskeleton based exerciser,” in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2004, vol. 3, pp. 2973–2978.

[31] M. Bianci *et al.*, “Development of low-cost portable hand exoskeleton for assistive and rehabilitation purposes,” in *Proceedings of eNTERFACE’16, Enschede*, 2016, no. July.

[32] D. Elstob and E. L. Secco, “A Low Cost Eeg Based Bci Prosthetic Using Motor Imagery,” Mar. 2016.

[33] T. D. Sunny, T. Aparna, P. Neethu, J. Venkateswaran, V. Vishnupriya, and P. S. Vyas, “Robotic Arm with Brain – Computer Interfacing,” in *International Conference on Emerging Trends in Engineering, Science and Technology (ICETEST - 2015)*, 2016, vol. 24, pp. 1089–1096.

[34] AREXX Engineering, “ROBOT ARM PRO V3 MANUAL.” [Online]. Available: http://www.arexx.com/robot_arm/downloads/manual/robot_arm_pro_manual_en.zip.

[35] A. C. Zoeller and K. Drewing, “A Systematic Comparison of Perceptual Performance in Softness Discrimination with Different Fingers,” *Attention, Perception, Psychophys.*, vol. 82, no. 7, pp. 3696–3709, Oct. 2020.

[36] “Drawing Basics: Proportions of the Arm.” [Online]. Available: http://www.paintdrawpaint.com/2011/01/drawing-basics-proportions-of-arm.html. [Accessed: 30-Dec-2020].

ACKNOWLEDGEMENTS

We wish to thank our advisers Assist. Prof. Hakan Solmaz are Assoc. Prof. Armağan Fatih Karamanlı. They supported us with their expertise and allowed us to use the university facilities.

A special thanks to Elie Chebel who is a master student in Biomedical Engineering Institute for his help and support for Python codes. He brought his positive attitude on our team brainstorming sessions in the last period of this project.

Also, we would like to acknowledge our university management office and security team for making the lab accessible during pandemic for us. This work was wholly funded by Bahçeşehir University which we appreciate.

Lastly, for replying even in weekends and fast delivery of replacement electrodes we would like to note the name of Cemil Müftüoğlu from CMSense.

APPENDIX A

CODES USED IN BIOMEDICAL ENGINEERING SECTION

Cortex Core code with modifications

Main template of Cortex is provided by Emotiv for software developers. Modifications are done by using unique credentials. Due to that, code is not provided in this report since it is not open source.

Cortex Live Stream Code with Project Profile Name

Main template of Cortex is provided by Emotiv for software developers. Modifications are done by using unique credentials. Due to that, code is not provided in this report since it is not open source.

APPENDIX B

CODES USED IN MECHATRONICS ENGINEERING SECTION

Arduino code used for verification of robotic arm with PS2 controller

```
#include <PS2X_lib.h>
#include <Servo.h>

PS2X ps2x;

                  //Initializing servos
Servo Elbow;
Servo Fist;
Servo Yaw;
Servo ShoulderX;
Servo ShoulderY;

int error = 0;
byte type = 0;
byte vibrate = 0;

int SY = 60;      //Shoulder Y counter. Controller commands: (Front & Down
Arrows)
int SX = 90;      //Shoulder X counter                        (Left & Right
Arrows)
int FAY = 60;     //Forearm yaw counter                       (Square &
Circle)
int E = 60;       //Elbow counter                             (Triangle & X)
int F = 60;       //Fist counter                              (L2 & R2)

void setup(){
 Serial.begin(57600);
                  //Attaching servos
 Elbow.attach(2);
 Fist.attach(5);
 Yaw.attach(6);
 ShoulderY.attach(4);
 ShoulderX.attach(3);

 error = ps2x.config_gamepad(13,11,10,12, true, true);   //GamePad(clock,
command, attention, data, Pressures?, Rumble?)

 if(error == 0){
  Serial.println("Found Controller, configured successful");
  Serial.println("Try out all the buttons;");
 }

  else if(error == 1)
   Serial.println("No controller found, check wiring.");

  else if(error == 2)
   Serial.println("Controller found but not accepting commands.");

  else if(error == 3)
```

```
   Serial.println("Controller refusing to enter Pressures mode, may not
support it.");

   type = ps2x.readType();
     switch(type) {
       case 0:
        Serial.println("Unknown Controller type");
       break;
       case 1:
        Serial.println("DualShock Controller Found");
       break;
     }
}

void loop(){

 if(error == 1)
  return;

 else { //DualShock Controller

    ps2x.read_gamepad(false, vibrate);          //read controller and set
large motor to spin at 'vibrate' speed

    if(ps2x.Button(PSB_START))                  //will be TRUE as long as
button is pressed
         Serial.println("Start is being held");
    if(ps2x.Button(PSB_SELECT))
         Serial.println("Select is being held");

     if(ps2x.Button(PSB_PAD_UP)) {              //will be TRUE as long as
button is pressed
       Serial.print("Up held this hard: ");
       Serial.println(ps2x.Analog(PSAB_PAD_UP), DEC);
       SY++;
                                                //Limit angle to 180
degrees
       if(SY > 180)
       {
         SY = 180;
       }

      ShoulderY.write(SY);
      }
      if(ps2x.Button(PSB_PAD_RIGHT)){
       Serial.print("Right held this hard: ");
       Serial.println(ps2x.Analog(PSAB_PAD_RIGHT), DEC);
       SX--;
                                                //Limit angle to 0 degrees
       if(SX < 0)
       {
         SX = 0;                                //Closing shoulder
       }

      ShoulderX.write(SX);
      }
      if(ps2x.Button(PSB_PAD_LEFT)){
```

```
        Serial.print("LEFT held this hard: ");
        Serial.println(ps2x.Analog(PSAB_PAD_LEFT), DEC);
        SX++;
                                                          //Limit angle to 180
degrees
        if(SX > 180)
        {
          SX = 180;                                       //Opening shoulder
        }

       ShoulderX.write(SX);
       }
       if(ps2x.Button(PSB_PAD_DOWN)){
        Serial.print("DOWN held this hard: ");
        Serial.println(ps2x.Analog(PSAB_PAD_DOWN), DEC);
        SY--;
                                                          //Limit angle to 0 degrees
        if(SY < 0)
        {
          SY = 0;
        }

        ShoulderY.write(SY);
       }

         if(ps2x.Button(PSB_L3))
          Serial.println("L3 pressed");
         if(ps2x.Button(PSB_R3))
          Serial.println("R3 pressed");

         if(ps2x.Button(PSB_L2))
         {
          Serial.println("L2 pressed");
          F--;
                                                           //Limit angle to 0 degrees
          if(F < 0)
          {
           F = 0;                                          //Opening fist
          }

           Fist.write(F);
          }

         if(ps2x.Button(PSB_R2))
         {
          Serial.println("R2 pressed");
          F++;
                                                           //Limit angle to 180
degrees
          if(F > 180)
          {
           F = 180;                                        //Closing fist
          }

           Fist.write(F);
          }
```

```
        if(ps2x.Button(PSB_GREEN)) {
         Serial.println("Triangle is pressed");
         E--;
                                                    //Limit angle to 0 degrees
         if(E < 0)
         {
          E = 0;                                    //Opening elbow
         }

          Elbow.write(E);
        }

        if(ps2x.Button(PSB_BLUE)) {
         Serial.println("X pressed");
         E++;
                                                    //Limit angle to 180
degrees
         if(E > 180)
         {
          E = 180;                                  //Closing elbow
         }

          Elbow.write(E);
        }

        if(ps2x.Button(PSB_PINK)) {
         Serial.println("Square is pressed");
         FAY--;
                                                    //Limit angle to 0 degrees
         if(FAY < 0)
         {
          FAY = 0;                                  //Opening yaw
         }

          Yaw.write(FAY);
        }

        if(ps2x.Button(PSB_RED)) {
         Serial.println("Circle pressed");
         FAY++;
                                                    //Limit angle to 180
degrees
         if(FAY > 180)
         {
          FAY = 180;                                //Closing elbow
         }

          Yaw.write(FAY);
        }
 }
 delay(50);
}
```

Arduino code used to link robotic arm to Cortex

```
#include <Servo.h>

                  //Initializing servos
Servo Elbow;
Servo Fist;
Servo Yaw;
Servo ShoulderX;
Servo ShoulderY;

int error = 0;
byte type = 0;
byte vibrate = 0;

int SY = 60;        //Shoulder Y counter      (Front & Down Arrows)
int SX = 90;        //Shoulder X counter      (Left & Right Arrows)
int FAY = 25;      //Forearm yaw counter      (L1 & R1)
int E = 0;        //Elbow counter             (TriangleT & X)
int F = 0;        //Fist counter              (L2 & R2)

void setup(){
 Serial.begin(9600);
                                //Attaching servos
 Serial.setTimeout(10);
 Elbow.attach(2);
 Fist.attach(5);
 Yaw.attach(6);
 ShoulderY.attach(4);
 ShoulderX.attach(3);
  Elbow.write(0);
  Yaw.write(90);
  Fist.write(90);

}
String data;
void loop(){


while(Serial.available()>0) { //DualShock Controller
   data=Serial.readString();
    Serial.println(data);
    data.trim();
    if(data.equals("drop")) {
         Serial.println("Closing Fist");
         F=F+5;;
                                              //Limit  angle  to
180 degrees
         if(F > 180)
```

```
                {
                 F = 180;                                  //Closing Fist
                }

                 Fist.write(F);
               }
               if(data.equals("push")) {
                Serial.println("Opening Fist");
                F=F-5;
                                                          //Limit angle to 0
degrees
                if(F <0)
                {
                 F = 0;                                   //Opening Fist
                }

                 Fist.write(F);
               }

          if(data.equals("lift")) {
                Serial.println("Elbow Closing");
                E++;
                                                          //Limit angle to 0
degrees
                if(E > 100)
                {
                 E = 100;                                  //Opening elbow
                }

                 Elbow.write(E);
               }
           if(data.equals("pull")) {
                Serial.println("Elbow Open");
                E--;
                                                          //Limit angle to 0
degrees
                if(E < 10)
                {
                 E = 10;                                   //Opening elbow
                }

                 Elbow.write(E);
               }
           if(data.equals("rotateClockwise")) {
                Serial.println("Yaw turning clockwise");
                FAY=FAY+5;
                      Serial.println(FAY);
//Limit angle to 0 degrees
                if(FAY > 135)
```

```
                {
                 FAY = 135;                                   //Opening yaw
                }
                 Yaw.write(FAY);

               }if(data.equals("rotateCounterClockwise")) {
                Serial.println("Yaw turning coutner-clockwise");
                FAY=FAY-5;
                      Serial.println(FAY);
//Limit angle to 0 degrees
                if(FAY < 25)
                {
                 FAY = 25;                                   //Opening yaw
                }
                 Yaw.write(FAY);
               }
       }
       delay(50);
      }
```