EMG CONTROLLED SYSTEM OF 4-DOF ROBOTIC ARM

JUNIOR BIN SINTAR

BACHELOR OF MECHATRONICS ENGINEERING WITH HONOURS
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2019

EMG CONTROLLED SYSTEM OF 4-DOF ROBOTIC ARM

JUNIOR BIN SINTAR

A report submitted
in partial fulfillment of the requirements for the degree of
Bachelor of Mechatronics Engineering with Honours

Faculty of Electrical Engineering

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2019

DECLARATION

I declare that this thesis entitled “EMG CONTROLLED SYSTEM OF 4-DOF ROBOTIC ARM is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature :
Name :
Date :

APPROVAL

I hereby declare that I have checked this report entitled "EMG CONTROLLED SYSTEM OF 4-DOF ROBOTIC ARM" and in my opinion, this thesis it complies the partial fulfillment for awarding the award of the degree of Bachelor of Mechatronics Engineering with Honours

Signature : ....

Supervisor Name : ....

Date : ....

DEDICATIONS

To my beloved mother and father

ACKNOWLEDGEMENTS

Throughout progressing in my final year project, I have encountered many outstanding individuals. They have helped me and contributed their knowledge in helping me complete my Final Year Project.

Firstly, I would like to give my deepest gratitude to my supervisor, Dr Fariz bin Ali@Ibrahim for his guidance and support throughout my Final Year Project. I am grateful for having him giving and sharing his knowledge and experiences as well as giving me helpful advices.

I would also like to thank to our Final Year Project coordinator, Assoc. Prof. Dr. Ahmad Zaki bin Haji Shukor for giving helpful insight during the Final Year Project. I would also like to take this opportunity to thank my panels, Dr. Hairol Nizam bin Mohd Shah and Dr. Nurdiana binti Nordin@Musa for consulting me during the seminar. Their guidance has encouraged me to improve myself to do better in this project.

I would also like to give credits to my friends whom I have share moments in facing difficulties throughout the final year project. Thank you for assisting me in completing this report. Lastly, it would be my greatest pleasure to give thanks to my parents who have never ceased to give me moral support. Without them, I would never have the confidence to take on new challenges in my life.

ABSTRACT

Electromyography (EMG) provides an alternative way of providing signal responses from the muscle. As such, the recent trend in developing myoelectric devices have spark the interest in this specific field of study. This is because the traditional controllers lack in certain parts which reduce the utilization of limbs to control devices mainly the robotic arm. However, noise such as crosstalk, motion artifact, ambient noise and inherent noise have become a major issue when handling EMG signals. The preparation of electromyography requires more attention in terms of muscle group selection, electrode placement and condition of the surrounding as it will affect the signal output. The aim of this study was to develop a 4 degree-of-freedom (DOF) robotic arm that can be controlled using EMG signals. The correlation between the EMG signal and the robotic arm are required to identified to analyze the performance of robotic arm. Review on the actuator, electromyography methods and microcontroller are done to evaluate the techniques used from past researches. The methods of this project include identification the error percentage of the actuator, classification and validation of EMG signals based on hand gestures, the correlation between the characteristic of the EMG signals with the functionality and performance of the robotic arm. The experiment showed that the actuator produced minor percentage error and does not affect the robotic arm accuracy significantly. The sampling rate and arm position affect the EMG signal output. In addition, the controllability of the robotic arm was low because the motors are controlled independently. The objectives of the project are achieved as the EMG-controlled robotic arm has been successfully developed. The robotic arm is still available for improvement by adding multiple channel sensor and implementing a wireless system.

ABSTRAK

Elektromilogi (EMG) menyediakan cara untuk mendapatkan bacaan isyarat daripada otot. Sehubungan dengan itu, penghasilan peranti myoelektrik telah menarik minat terhadap bidang ini. Hal ini kerana pengawal yang sedia ada mempunyai kekurangan yang menghadkan kadar penggunan anggota badan untuk mengawal pergerakkan peranti terutamanya tangan robotik. Walau bagaimanapun, isyarat EMG adalah terdedah kepada gangguan seperti pertembungan isyarat dari otot yang lain, pergerakkan otot dan radiasi daripada peranti elektrik. Penyediaan elektromilogi memerlukan lebih perhatian terutama pada pemilihan kumpulan otot, penempatan elektrod pada tubuh badan dan keadaan persekitaran kerana ia akan menggangu isyarat EMG. Tujuan utama projek ini ialah untuk membuat tangan robotik yang mempunyai empat darjah kebebasan yang dikawal menggunakan isyarat EMG. Hubungan antara isyarat EMG dan tangan robotik perlu diketahui untuk mengkaji prestasi tangan robotik. Kajian semula tentang mesin penggerak, teknik elektromilogi dan mikropengawal dilaksanakan untuk menilai teknik yang digunakan dalam kajian lepas. Antara kaedah yang digunakan ialah mengenalpasti tahap keberkesanan mesin penggerak, pengkelasan dan pengesahan isyarat EMG berdasarkan isyarat tangan dan mengenal pasti hubungan diantara ciri isyarat EMG dengan fungsi serta prestasi tangan robotik. Hasil eksperimen ini menunjukkan bahawa mesin penggerak mempunyai peratus kesilapan yang kecil dan tidak mengganggu ketepatan tangan robotik. Tambahan pula, kadar pengambilan sampel dan kedudukan tangan mempengaruhi isyarat EMG. Tahap kawalan tangan robotik pula adalah rendah kerana mesin penggerak perlu dikawal satu persatu. Objektif projek ini telah dicapai kerana tangan robotik yang dikawal menggunakan isyarat EMG dapat dihasilkan. Tangan robotik boleh lagi diperbaiki dengan menambahkan jumlah pengesan dan mengimplimentasikan sistem tanpa wayar.

TABLE OF CONTENTS

PAGE

DECLARATION

APPROVAL

DEDICATIONS

ACKNOWLEDGEMENTS iv

ABSTRACT v

ABSTRAK vi

TABLE OF CONTENTS vii

LIST OF TABLES ix

LIST OF FIGURES x

LIST OF SYMBOLS AND ABBREVIATIONS xiii

LIST OF APPENDICES xiv

CHAPTER 1 INTRODUCTION **1**
1.1 Overview 1
1.2 Background 1
1.3 Motivation 2
1.4 Problem Statement 3
1.5 Objectives 4
1.6 Scopes 5

CHAPTER 2 LITERATURE REVIEW **6**
2.1 Overview 6
2.2 Actuator 6
  2.2.1 Servo Motor 6
  2.2.2 DC Motor 7
  2.2.3 Stepper Motor 8
2.3 Electromyography (EMG) 9
  2.3.1 Invasive EMG 10
  2.3.2 Non-invasive EMG 11
  2.3.3 Electrode Placement 12
2.4 Microcontroller 12
  2.4.1 Arduino Uno R3 13
  2.4.2 Raspberry Pi 13
  2.4.3 Arduino Mega 2560 14
2.5 Literature Review Summary 15

CHAPTER 3 METHODOLOGY **18**
3.1 Overview 18

3.2 Project Flowchart 18
3.3 Task 1: Robotic arm Hardware Development 19
    3.3.1 Servo Motor 19
    3.3.2 Arduino Uno R3 21
    3.3.3 Myoware Muscle Sensor 21
    3.3.4 Robotic Arm 22
    3.3.5 Robotic Arm Circuit Design 23
    3.3.6 Robotic Arm Source Code Developement 24
        3.3.6.1 Motor Selection Subroutine 25
        3.3.6.2 Motor Run Subroutine 27
3.4 Task 2: Developing the Forward Kinematic of 4-DOF Robotic Arm 28
3.5 Task 3: Sensor calibration and Electrode Positioning 30
3.6 Experiment 1: Error Percentage of Motor Test 31
    3.6.1 Experiment 1 Setup 32
    3.6.2 Procedure 33
3.7 Experiment 2: EMG Signal Classification Test 34
    3.7.1 Experiment 2 Setup 34
    3.7.2 Procedure 36
3.8 Experiment 3: Hand Gesture Validation Test 37
    3.8.1 Procedure 37
3.9 Experiment 4: EMG Signal to Servo Motor Rotation Correlation Test 38
    3.9.1 Experiment 4 Setup 38
    3.9.2 Procedure 39
3.10 Experiment 5: Robotic Arm Accuracy Test Using Forward Kinematic 40
    3.10.1 Experiment 5 Setup 40
    3.10.2 Procedure 41
3.11 Experiment 6: Robotic Arm Performance Test 41
    3.11.1 Experiment 6 Setup 42
    3.11.2 Procedure 43
3.12 Objective Mapping 43

CHAPTER 4 RESULTS AND DISCUSSIONS 44
4.1 Experiment 1: Percentage Error of Motor Test 44
4.2 Experiment 2: EMG Signal Classification Test 45
4.3 Experiment 3: Hand Gesture Validation Test 46
4.4 Experiment 4: EMG Signal to Servo Motor Rotation Correlation Test 49
4.5 Experiment 5: Robotic Arm Accuracy Test Using Forward Kinematic 53
4.6 Experiment 6: Robotic Arm Performance Test 57

CHAPTER 5 CONCLUSION AND RECOMMENDATIONS 59
5.1 Conclusion 59
5.2 Recommendations 59

REFERENCES 61

APPENDICES 64

LIST OF TABLES

Table 1.1 List of recent projects related with electromyography 2

Table 2.1 Summary of actuator review 15

Table 2.2: Summary of types of electromyography review 16

Table 2.3 Summary of electrode placement review 16

Table 2.4 Summary of microcontroller review 17

Table 3.1: Specification of servo motor 20

Table 3.2 Arduino UNO Rev 3 specification 21

Table 3.3 Specification of Myoware muscle sensor 22

Table 3.4 DH parameters definition 29

Table 3.5 DH parameter of 4-DOF robotic arm 30

Table 3.6 Experiment 1 connection scheme 32

Table 3.7 Experiment 2,3 connection scheme 35

Table 3.8 Experiment 4,5,6 connection scheme 38

Table 3.9 Experiment mapped to objectives 43

Table 4.1 Percentage error of motor test 44

LIST OF FIGURES

Figure 1.1 Example of EMG signal waveform 1
Figure 2.1 Servo motor 7
Figure 2.2 DC motor 8
Figure 2.3 Stepper motor 9
Figure 2.4 Needle electrode 10
Figure 2.5 Surface electrode 11
Figure 2.6 Arduino Uno R3 13
Figure 2.7 Raspberry Pi 14
Figure 2.8: Arduino Mega 2560 14
Figure 3.1 Flowchart of the process of the project 19
Figure 3.2 Types of servo motor (a) SG90 (b) Futaba S3003 (c) MG946R (d) HD1501 MG 20
Figure 3.3 Myorware muscle sensor 22
Figure 3.4 Robotic arm 23
Figure 3.5 Robotic arm circuit diagram 24
Figure 3.6 Main program process flow 25
Figure 3.7 Motor selection subroutine 26
Figure 3.8 A section of the source code for the motor selection subroutine 26
Figure 3.9 Motor run subroutine flowchart 27
Figure 3.10 A section of the source code for the motor run subroutine 28
Figure 3.11 The robotic arm axis 29
Figure 3.12: Placement of EMG sensor (a) area of placement (b) electrode positioning 31

Figure 3.13 Experiment 1 circuit diagram 33

Figure 3.14 Experiment 1 Setup (a) components setup (b) angle sheet place in between servo motor and servo horn 33

Figure 3.15 Experiment 2 circuit diagram 35

Figure 3.16 Experiment 2 setup 35

Figure 3.17 Hand gestures (a) Hand rest (b) Thumb flex (c) Index finger flex (d) Middle finger flex (e) Ring finger flex (f) Small finger flex (g) Grasp (h) Wrist flexion (i) Wrist extension 37

Figure 3.18 EMG signal to servo motor rotation correlation test circuit diagram 39

Figure 3.19 EMG signal to servo motor rotation correlation test 39

Figure 3.20 Object A and object B 40

Figure 3.21: Experiment 5 setup (a) object location (b) target 41

Figure 3.22 Object C and target 42

Figure 3.23 Experiment 6 setup 42

Figure 4.1 Graph of EMG signal vs time 45

Figure 4.2 Graph of EMG signals versus time (30ms sampling rate) 46

Figure 4.3 Graph of EMG signals versus time (1000ms sampling rate) 47

Figure 4.4 Graph of EMG signal versus time (30 ms and arm hovering) 48

Figure 4.5 The graph of EMG signals versus servo motor A angle (a) Anticlockwise rotation (b) Clockwise rotation 49

Figure 4.6 The graph of EMG signals versus servo motor B angle (a) Anticlockwise rotation (b) Clockwise rotation 50

Figure 4.7 The graph of EMG signals versus servo motor C angle (a) Anticlockwise rotation (b) Clockwise rotation 51

Figure 4.8 The graph of EMG signals versus servo motor D angle (a) Clockwise rotation (b) Anticlockwise rotation 52

Figure 4.9 The graph of EMG signals versus servo motor E Angle (a) Anticlockwise rotation (b) Clockwise rotation 53

Figure 4.10 Comparison of actual coordinate with mean coordinate for coordinate (7,24) with object A 54

Figure 4.11 Comparison of actual coordinate with mean coordinate for coordinate (-7,24) with object A 55

Figure 4.12 Comparison of actual coordinate with mean coordinate for coordinate (7,24) with object B 56

Figure 4.13 Comparison of actual coordinate with mean coordinate for coordinate (7,24) with object B 57

Figure 4.14 Graph of location of object versus the time taken 58

LIST OF SYMBOLS AND ABBREVIATIONS

EMG - Electromyography
DOF - Degree of freedom
FDS - Flexor digitorum superficialis
V - Voltage
A - Ampere
bits - Binary digits
θ - Angle

LIST OF APPENDICES

APPENDIX A	MAIN SOURCE CODE FOR ROBOTIC ARM MOVEMENT CONTROL	64
APPENDIX B	SOURCE CODE FOR AUTOMATED CONTROL: LOCATION (-9,12)	74
APPENDIX C	SOURCE CODE FOR AUTOMATED CONTROL: LOCATION (-7,28)	76
APPENDIX D	SOURCE CODE FOR AUTOMATED CONTROL: LOCATION (10,14)	78
APPENDIX E	SOURCE CODE FOR AUTOMATED CONTROL: LOCATION (-10,20)	80
APPENDIX F	EMG SIGNALS DISTRIBUTION FOR DIFFERENT HAND GESTURES (EXPERIMENT 2)	82
APPENDIX G	EMG SIGNALS DISTRIBUTION FOR CONDITION 1 (EXPERIMENT 3)	83
APPENDIX H	EMG SIGNALS DISTRIBUTION FOR CONDITION 2 (EXPERIMENT 3)	84
APPENDIX I	EMG SIGNALS DISTRIBUTION FOR CONDITION 3 (EXPERIMENT 3)	85
APPENDIX J	EMG SIGNALS DISTRIBUTION WITH ANGLE ROTATION (DEGREES) FOR MOTOR A (EXPERIMENT 4)	86

APPENDIX K EMG SIGNALS DISTRIBUTION WITH ANGLE ROTATION (DEGREES) FOR MOTOR B (EXPERIMENT 4) 88
APPENDIX L EMG SIGNALS DISTRIBUTION WITH ANGLE ROTATION (DEGREES) FOR MOTOR C (EXPERIMENT 4) 90
APPENDIX M EMG SIGNALS DISTRIBUTION WITH ANGLE ROTATION (DEGREES) FOR MOTOR D (EXPERIMENT 4) 92
APPENDIX N EMG SIGNALS DISTRIBUTION WITH ANGLE ROTATION (DEGREES) FOR MOTOR E (EXPERIMENT 4) 94
APPENDIX O COORDINATE OF END EFFECTOR AT OBJECT A BASED ON THE ANGLE OF JOINTS (EXPERIMENT 5) 95
APPENDIX P COORDINATE OF END EFFECTOR AT OBJECT B BASED ON THE ANGLE OF JOINTS (EXPERIMENT 5) 96
APPENDIX Q TIME TAKEN TO TRANSPORT OBJECT C VIA MANUAL CONTROL (EXPERIMENT 6) 97
APPENDIX R TIME TAKEN TO TRANSPORT OBJECT C VIA AUTOMATED CONTROL (EXPERIMENT 6) 98

CHAPTER 1

INTRODUCTION

1.1 Overview

This section contains five parts which are project background, motivation problem statement, objectives and scopes. The project background highlights on some key aspects and fundamental of electromyography (EMG). The motivation discusses on the purpose of conducting the project based on the recent trend that associates with myoelectric devices. The problem statement discusses the problem regarding the preparation and development of EMG controlled devices. The objectives are sets of goal that needs to be achieved at the end of this project and is associated with limitation from the scope.

1.2 Background

Electromyography (EMG) has been commonly used in biomedical and clinical application. EMG is a process of measuring electrical activity in the muscles. An EMG signal[1] is basically the summation of all the action potential recorded from the muscle fiber. The motor units[2] which is made up of muscle fiber exhibits electrical properties when muscles undergoes contraction. The raw EMG signal is measured in terms of voltage and typically displayed in waveform. Figure 1.1 shows an example of EMG signal waveform.

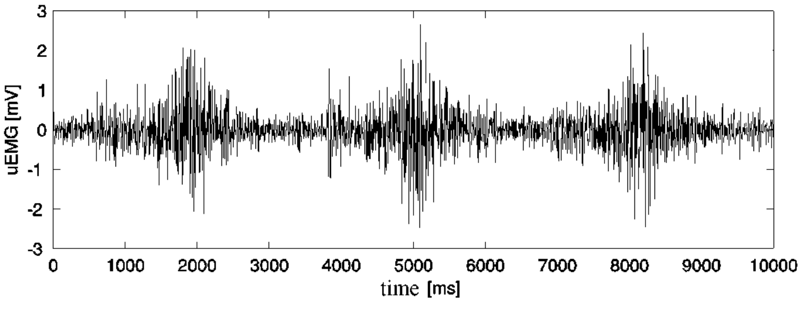


Figure 1.1 Example of EMG signal waveform

There are two known methods of extracting EMG signals which are the invasive and non-invasive methods. The invasive method uses wired electrodes that is inserted into the muscle’s fibers. The non-invasive method uses surface electrodes that are placed on the skin surface. The methods still use the same concept of recording EMG signal despite using different technique. The non-invasive method has been the more popular approach because the method is easier to conduct. However, both .techniques still share a common problem which is the EMG signals are highly exposed to noise [3]

There are several processes required before the EMG signal is readable. Pre-processing is the initial step in EMG signal data acquisition. The process includes amplification, filtration, rectification and analog to digital conversion. The EMG signal amplitude usually ranges from 0-10mV and the frequency ranges from 0 -500 Hz. The parameters depend of the type of electrode used. Signal that have been processed are transferred to the computer for real-time signal monitoring.

1.3 Motivation

In recent years, the usage of EMG signal for automation control, prosthetics and robotic arms has been more frequent. EMG signals have been used mostly in clinical application to diagnose muscles disorders in the past. Multiple researches have been conducted which aims to control devices using EMG signals. Table 1.1 shows a list of projects that have implemented electromyography in their device system.

Table 1.1 List of recent projects related with electromyography

Year	Projects
2015	Bilateral rehabilitation using an EMG-controlled robotic hand exoskeleton[4]
2016	Intelligent wheelchair controlled by EMG [5]
2017	EMG-controlled prosthetic hand with sensory system[6]
2018	Gait Robotic Exoskeleton[7]
2019	Robotic Arm movement using EMG signals[8]

The sudden interest in this field of studies is because the potential EMG signals have not yet been fully explored. The advantage of electromyography is that the signal is produced naturally by the movement of body parts. Controlling a device mainly a robotic arm would be more interactive compared to the standard controller because it utilizes the movement of the human body parts. In addition, this will also provide a more fluid control over the devices because traditional controllers restrict the movement to a certain body segment.

Lack of physical sensation of controlling a manipulator is also one of the factors that lead researches towards the usage of EMG signal. Most of the robotic arms are currently controlled by mechanical controllers and vision controllers. Examples of these controllers are joystick, buttons and image processing-based controllers. Both mechanical and vision controllers can control the robotic arm but lacks control over the transmission of force to the manipulator. EMG signals enable the user to control the devices via gesture which is influenced by the amount of force exerted by the muscle.

Hence, conducting this project will help in improving the control over myoelectric devices mainly the robotic arm. The behavior of manipulator based on the EMG signal from performing limb movement is the main aspect that is used to determine controller functionality and efficiency. The study on EMG-based controller will also help in overcoming the limitation of the previous controller.

1.4 Problem Statement

Electromyography exhibit several issues that affect the outcome of the signal. The placement of electrode is one of the main concerns of electromyography. Both method, invasive and non-invasive electromyography requires accurate positioning in order to get a quality EMG signal. The muscle group[9] needs to be identified initially before applying the electrode. The effect of electrode misplacement is cross talk which leads to misinterpretation of EMG signals.

Another problem when using electromyography is that the EMG signals are easily exposed to noise. Some examples of the noises are motion artifact, ambient noise and inherent noise. The motion artifact[10] produces the most noise compare to the other noises. Motion artifacts is the involuntary movement of the body that causes the muscles to contract randomly. This leads to the interference of the contact between the muscle fiber and the electrode.

The controller requires to read EMG signal which then will be transmitted to the actuator to produce movement. The EMG signals have an infinite value which mean that the signals produced are random and hardly the same for the same gesture. The signals need to be allocated into a certain range value for the controller to distinguish the gesture. This process requires proper EMG signal classification as the signal will also be affected by noise.

The precision of the robotic arm also needs to be considered. This is more focused on the selection of actuators as they handle the rotational movement of the robotic arm. Some of the motors may have major defects which will cause the motors to have a higher percentage error. A higher error will reduce the performance rate of the robotic arm as well as make it harder to manipulate. Hence, there are several problems in developing an EMG controlled robotic arm which are the electrode placement, noise exposure, classification of EMG signals and accuracy of the robotic arm.

1.5 Objectives

1. To design and develop a functional 4-DOF robotic arm that is controlled using EMG signals
2. To perform analysis on the operation of the Robotic Arm based on the characteristic of EMG signal collected from the forearm.
3. To test the effectiveness of the actuators based on the instruction given by the controller

1.6 Scopes

1. Developing a 4 degree of freedom (DOF) robotic arm with base, shoulder, elbow, wrist and gripper motion.
2. The movement of the robotic arm is controlled using EMG signals that are collected from the upper limb
3. The system will use a microcontroller to process the input and output data.
4. Surface electrodes will use to collect EMG signals that is interfaced by a single channel EMG sensor
5. A total of five actuators will be used to actuate the robotic arm joints.
6. The robotic arm forward kinematic will be developed using the DH convention method.

CHAPTER 2

LITERATURE REVIEW

2.1 Overview

This section discusses on the review of information from other journals and papers. The topic that is reviewed are actuator, electromyography and controller. The actuator consists of three parts which are servo motor, DC motor and stepper motor. The electromyography section consists of three parts which are invasive electromyography, non-invasive electromyography and electrode placement. Lastly, the microcontroller section consists of three parts which are Arduino UNO, Raspberry Pi and Arduino Mega 2560. The summary of the reviews is discussed in in the final section of this chapter.

2.2 Actuator

An actuator is a device that convert energy into motion. The actuators are required to move the robotic links and joints in order to replicate the real human limbs. Different types of actuator have their own advantage and disadvantages. Three types of actuator will be discussed in this part which are servo motor, DC motor and Stepper motor.

2.2.1 Servo Motor

The servo motor is a type of rotary actuator that can have a precise control over angular position. There are two types of servo motor, a DC servo motor and an AC servo motor. Both types operate on PWM (Pulse Width Modulation) which is activated when a pulse is sent to its respective output pins. The motor can produce high torque which can generate enough force to move medium weighted loads. The servo motors are commonly used in robotic applications. Figure 2.1 shows a typical servo motor.

Figure 2.1 Servo motor

Several research projects have used the servo motor for its ease of control. The robotic arm in [11] used servo motors as actuators which is controlled using internet connection. The angle of rotation is inserted into the program and transferred to the controller. The manipulator was able to rotate based on the value given by the researchers. The voiced controlled prosthetic arm[12] is another example of servo motor application. The prosthetic consists of six standard servo motors which are controlled by human speech. It is possible to generate a range of PWM signals by classification of different voice sample.

Servo motors also have a very high accuracy in terms of angle rotation. It uses a close loop negative feedback system using a potentiometer to return the current angle of the servo motor. In paper [13], the servo motor is used to actuate the gripper which used a geared system. The gripper is locked when the servo reached the desired angle. Another example of servo motor accuracy is the chess playing robot[14] which uses servo motor to lower down the end effector. The servo motor pushes the manipulator tip to reach the top of the chess piece without overshooting.

The servo motors in general have high torque, high accuracy and easy to control in which is suitable for a control operation. The common rectangular shape of a servo motor is also an advantage for some cases as it is easier to install it in the system. Any standard microcontroller with PWM output can control a servo motor.

2.2.2 DC Motor

A DC Motor is a rotary actuator that converts electrical energy into mechanical energy. The motor main components are axle, rotor, commutator, field magnets, and brushes. For most DC motor, the armature winding is placed between two permanent magnets that produces magnetic field. The armature will experience force when current flows through the windings and produce rotation. Fleming's left-hand rule is used to determine the direction of force acted on the armature winding. Figure 2.2 shows a typical DC motor.

Figure 2.2 DC motor

Commonly, DC motors are favored for its rotation speed rather than accuracy. Some DC motors are pre-installed with an encoder which can read the angle rotation of the motor. The exoskeleton in paper [15] uses DC motor in a close loop system. The DC motor need to react based on the change of dynamics in the human movement. In this case, speed is required because the human dynamics changes in a very fast pace. Most of the DC motor have average medium torque which is not favorable for load lifting. However, the torque can be increased by applying a gear system. In paper [6], the researchers used DC motors to actuate an exoskeleton using geared -based system.

In general, a DC motor have high speed, medium to high torque, controllable and flexible. The motor is also cheap and have a wide range of specification. The motor can be improved by using geared system.

2.2.3 Stepper Motor

Stepper motor is a DC motor that divides its rotation into discrete steps. The motor is equipped with multiple coils that are organized in groups. These groups are called "Phase", and this allow the motor to rotate in in steps when the coils are energized. A precise positioning and speed control can be achieved by using a stepper motor. There are two types of stepper motor, which are unipolar and bipolar. For unipolar, only half of the coil are energized at a time while bipolar type energizes all the coil to rotate the motor. Figure 2.3 shows a typical stepper motor.

Figure 2.3 Stepper motor

Some uses of stepper motor can be seen in robotics application. In paper [17],two stepper motor are used in a 2-DOF robotic arm for the base movement and shoulder movement. The type of motors is a 6 wired unipolar stepper motor and is powered by 12 V DC power supply. Motor driver is required to control the rotation of the motor. The advantage of stepper motor is that it has a high accuracy in rotation which is useful for positioning. Another example of usage of stepper motor can be found in paper [18] . A hybrid stepper motor is used to actuate the base, shoulder, elbow, and wrist of the pick and place robotic arm.

In general, the stepper motor has a high angular rotation accuracy medium to high torque and have medium rotation speed. However, stepper motor needs to be connected to a motor drive to function properly.

2.3 Electromyography (EMG)

Electromyography (EMG) is the process of recording muscle activity in terms of voltage. The EMG signals can be obtained from different muscle in the body. The topic that will be discussed in this section is the methods of extracting EMG signal and electrode placement.

2.3.1 Invasive EMG

The invasive EMG uses electrodes which are inserted in the muscle tissue to record muscle activity. Disposable needle electrodes that is attached with fine wires are mostly used for recording. Figure 2.4 show the typical needle electrode The radius of surface that the needle can detect is typically 1mm for a conventional EMG needle. They are about 100 muscle fiber [19] for the amount of surface covered by the needles while most of these fibers are distributed throughout a single muscle. This means that only about 4-6 fibers are detected in the cross section of the inserted needle.

Figure 2.4 Needle electrode

The method is focused more on detecting the activity of a small segment of the muscle. It can be an advantage or a disadvantage depending on the application. In research [20] , they used invasive electrode to record epileptic discharges that is present at a cortical level. The researcher stated that the recording of electroencephalography (EEG) using needle electrode is less reliable because they are required to correlate the electrode with scalp EEG. The practice of needle electrode also may cause harm. The paper [21] propose to limit the practice of invasive EMG electrode to qualified physician as the process requires more physicality. Consideration are made based on the harm towards user and the efficiency of the method.

Therefore, the invasive electrode is suitable for deep muscle signal analysis and a more focused muscle recording. The method however requires more precaution to take as needle insertion may cause harm to the body.

2.3.2 Non-invasive EMG

The non-invasive EMG uses electrode which are place on the surface of the skin. The electrode is covered with pre-gel substances that is attached with adhesive which can be stick to the skin surface. Figure 2.5 shows a typical surface electrode. The skin-mounted electrode[22] measures the electrical potential that occurs in all the muscle fiber that are located below the skin. This mean that the electrode covers a large amount of surface area of detection.

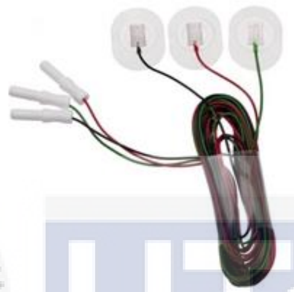


Figure 2.5 Surface electrode

Since the EMG signal are recorded through multiple layer of fibers, the electrode is more exposed noise. The characteristic of EMG signal depends a lot on the internal structure[23] of the body parts such as skin formation, blood flow velocity, tissue structure and the temperature of measuring surrounding. A controlled environment is required in order to get a more reliable EMG signal

Motion artifacts and electrode misplacement also effect the EMG signal [23] [24]. Motion artifact is the voluntary or involuntary movement of the human body. The movement may excite the muscle which will then produce electrical signal. Electrode misplacement tend to occur as the electrode cover a large surface are of the muscle. The result of misplacements is crosstalk which is the noise produce by activation of other muscle.

Therefore, the non-invasive method is suitable for a quick and overall muscle detection. The process of placing the electrode is also convenient as no penetration is required. The adhesive may cause a minor irritation to the body.

2.3.3 Electrode Placement

The human muscles consist of three categories[25] which are cardiac, skeletal and smooth muscles. Each of the muscle have their own traits. Smooth muscle are tissues that form the tissues of blood vessels and hollow organs. Cardiac muscle is muscles group which are not depended on the neural networks and contracts involuntary. Lastly, skeletal muscles are muscles which are attached to the bone by the tendons. Skeletal muscle is used to generate movement and force on the body. The human upper limb consists of three segment which are the arm segment, forearm segment and the hand segment. Each of the segment are interconnected by bones, muscles, tissues and vessel. The skeletal muscle contracts and relax to produce movement in the joints. The electrode is commonly placed at the midpoint of the muscle to get the optimum signal.

In research[26], a dual channel EMG sensor is placed on the Extensor Digitorum Muscle (EDM), Flexor Digitorum Superficialis muscle (FDS), Dorsal Interossei muscle (DI) , Plamar Interossei (PI) muscle for the movement of finger extension, flexion and abduction respectively. In research [27], a single channel EMG sensor is placed on the Adductor Policis (AP), First Pollicis Brevis (FPB),and Abductors Pollicis Brevis (APB) for thumb movement. In research [28], the researchers proposed two muscles which are the wrist muscle and finger muscles. The muscles chosen are the Flexor Carpi Radials (FCR), Extensor Carpi Ulnaris (ECU), Flexor Digitorum Superficial (FDS) and (EDC).

Therefore, the muscle group needs to be properly chosen before recording EMG signals from the muscle. The EMG signals produced is based on the movement of body parts that is produced by a certain muscle group. The common movement that will generate EMG signal from the arm is wrist movement and finger movement.

2.4 Microcontroller

A microcontroller is considered as a miniature computer that is on a single integrated circuit. The microcontroller has a central processing unit (CPU) which process data. There is also storage unit that can store data. A set of output and input pins are available for the controller to connect with other devices. In this part, we will discuss three types of controller which are the Arduino Uno R3, Raspberry Pi and Arduino Mega 2560.

2.4.1 Arduino Uno R3

The Arduino Uno is a microcontroller that is based on the ATmega328P. The total numbers of digital input/output of the controller is 14, in which 6 of the output can be used as PWM outputs. There are also 6 analog inputs in total. The Arduino Uno R3 operates in a 16MHz crystal oscillator and have a minimum operating voltage of 5V. The Arduino board can be powered by plugging a USB cable directly from a computer or by connecting it to the AC-to- DC adapter. Figure 2.6 shows an Arduino Uno R3.

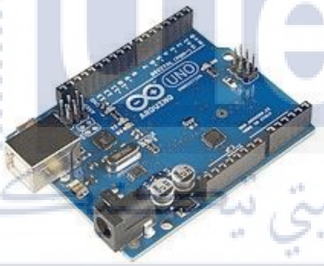


Figure 2.6 Arduino Uno R3

In journal [29], the researcher proposed the use of the Arduino Uno R3 to control the Automated Sorting Robotic Arm. The controller is proposed because the optimum processing capabilities of the robotic arm can be achieved while maintaining the simplicity of the system. The numbers of I/O are also enough to be able to connect multiple actuators and sensors.

2.4.2 Raspberry Pi

The Raspberry Pi is based on Broadcom BCM2837 Quad core (ARM Cortex-A53) SoC and have a memory of 1GB LPDDR2. The controller has a total of forty

GPIO header-pin, four USB ports, one Ethernet port and one HDMI port. It is powered by a USB connecter for 5.1V/ 2.5A dc. The Raspberry Pi have a compact design but maximize on its functionality. Figure 2.7 shows a Raspberry Pi microcontroller.

Figure 2.7 Raspberry Pi

The Maplin‟s OWI-535 [30], uses has four rotational joints which are the base, shoulder elbow and wrist and uses the Raspberry Pi as its main controller. The Raspberry Pi is chosen as it can interface a web cam and the robotic arm simultaneously via the internet. The L293D motor driver is connected in between the controller and the motor.

2.4.3 Arduino Mega 2560

The Arduino Mega is based on the ATmega2560 Microcontroller. It has a total of 54 input and output pins, 14 of the pins can be used as PWM output. It also has 16 analogue inputs and 4 UART ports. It runs on a 16MHz quartz crystal and is powered by DC USB power jack. The minimum operating voltage for this controller is 5V. Figure 2.8 shows an Arduino Mega 2560.

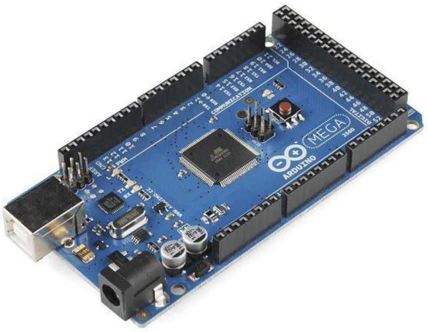


Figure 2.8: Arduino Mega 2560

The Arduino Mega 2560 is used in the mobile robot with a 6-DOF robotic arm [31] and is controlled wirelessly. The researchers use the controller as it requires multiple input and outputs for sensors, actuators and for the serial communicator. The researcher also recommend that the operating voltage supply is around 7V-12V as the board will be overheat if the voltage is above the limit.

2.5 Literature Review Summary

Table 2.1 shows the summary of the review of servo motor, DC motor and stepper motor which shows the advantages and disadvantages of each type of motor

Table 2.1 Summary of actuator review

Actuator	Advantages	Disadvantages
Servo Motor	• Easy to control [11] [12] • High accuracy [14] • High torque[13]	• Low angular velocity • Most of the servo can only rotate for 180°
DC Motor	• Can be controlled[15] • Flexible(can be modified) [6] • Can rotate 360° • High angular velocity	• Low torque[15]
Stepper Motor	• High accuracy[17] [18] • Can rotate 360°	• Medium angular velocity • Medium torque

Table 2.2 shows the summary of the review on invasive and non-invasive EMG which include the advantages and disadvantages of each method. Table 2.3 shows the summary of the review on electrode placement which shows the number of sensor channel, muscle group and the limb movement that is involved

Table 2.2: Summary of types of electromyography review

Type of Electrode		
Type	Advantages	Disadvantages
Invasive	- Suitable for deep muscle analysis[19] - Concentrated muscle recording[19]	- Potentially harmful [21] - Dependable with the application project [20]
Non-invasive	- Can record a large muscle group[22] - Easy to place on the skin	- Exposed to noise[23] [24]

Table 2.3 Summary of electrode placement review

Electrode Placement		
Number of channels	Muscle group	Movement involve
2	- FDS, EDM, DI, PI[26]	- Finger flexion, extension, abduction
1	- AP, FBP [27]	- Thumb movement
2	- FCR, ECU,FDS,EDC[28]	- Wrist and finger movement

Table 2.4 shows the summary of the Arduino Uno R3, Raspberry Pi and Arduino Mega 2560 which shows advantages and disadvantages of each microcontroller

Table 2.4 Summary of microcontroller review

Microcontroller	Advantages	Disadvantages
Arduino UNO Rev3	- Lower Price - The body size is compact	- Less I/O pins
Raspberry PI	- More I/O pins - Compact	- Higher price
Arduino Mega 2560	- More I/O Pins	- Higher price - Bigger in size

As a conclusion, the actuator that will be used is the servo motor. The servo motor is chosen because it is easy to control. The accuracy of servo motor is high which is suitable to control the robotic arm joints. The motor also has a high torque which is needed to lift the robotic arm. Next, the type of electrode chosen is the non-invasive electrode. This method is easier to prepare than the non-invasive method. The electrode is not harmful compared to the needle electrode. The surface of detection is also larger which is useful to detect the overall signal from a single muscle.

The skeletal muscle chosen is the Flexor Digitorum Superficialis (FDS) muscle using a single channel sensor. The hand movement targeted will be the finger movement. Lastly, the controller used will be the Arduino UNO because this project does not require many I/O pins. The controller has enough pins to connect all the components together. The price is reasonably cheaper than the other two controllers listed.

CHAPTER 3

METHODOLOGY

3.1 Overview

This chapter consists mainly of the tasks and experiments conducted in this project. There are three tasks in total which are the robotic arm hardware development, development of forward kinematic of the robotic arm, sensor calibration and electrode positioning. There is a total of six experiments conducted which are the servo motor error percentage test, EMG signal classification test, hand gesture validation test, EMG signal to servo motor rotation correlation test, robotic arm accuracy test using forward kinematic and performance test. The objective, setup and procedure of the experiments are discussed thoroughly in the experiment section

3.2 Project Flowchart

The project flowchart refers to the whole process of conducting the project. The project is conducted within a 28-week period that follows a Gantt chart. The process begins with identification of the problem based on the project background. Objective are formed to solve the problem that is faced while the scopes are set to according to the limitation of project. The hardware and source code development are done before the experiments are conducted. Any error that cause the data to become inconsistent will be fixed and the experiment will be repeated. All reliable data is documented, and the final report are prepared. Figure 3.1 shows the flowchart of the project.

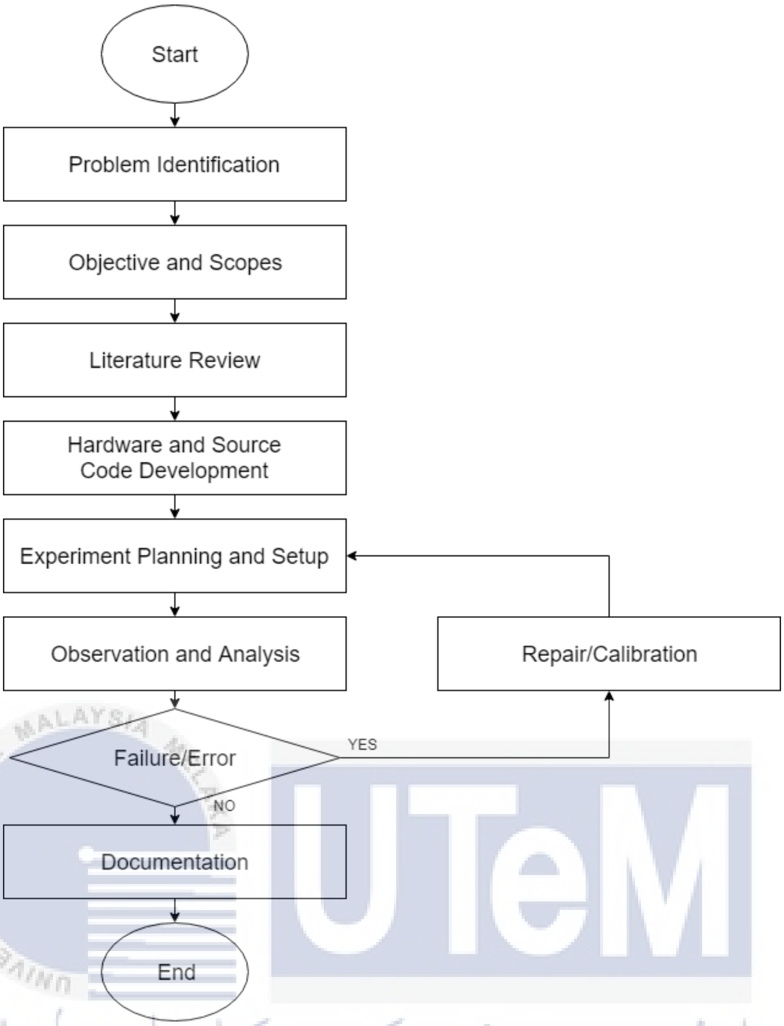


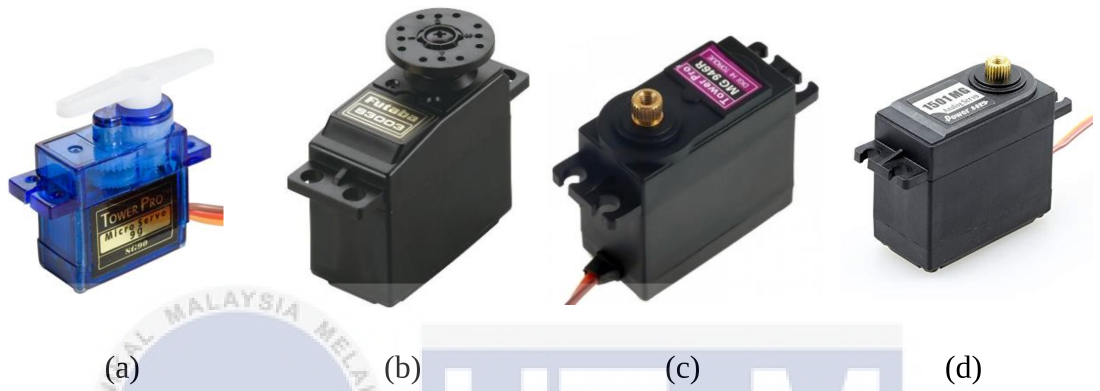
Figure 3.1 Flowchart of the process of the project

3.3 Task 1: Robotic arm Hardware Development

This section contains six section which discuss on the servo motor, Arduino Uno, Myoware muscle sensor, robotic arm, robotic arm circuit design and robotic arm source code development. The function and specification of each component are explained thoroughly.

3.3.1 Servo Motor

Servo motors are suitable for robotics arm because it has decent torque and precise control over angular rotation. The servo motors are controlled by sending pulse width modulation (PWM) through the control wires. The connection mount consists of three inputs which are positive supply, ground and signal. The robotic arm operates using 5 servo motor for each joint. The servo motors used are SG90, Futaba S3003, MG946R and two HD 1501 MG. Figure 3.2 shows the types of servo motor used and table 3.1 shows the specifications of each motors



(a) (b) (c) (d)

Figure 3.2 Types of servo motor (a) SG90 (b) Futaba S3003 (c) MG946R (d) HD1501 MG

Table 3.1: Specification of servo motor

Specification	SG90	Futaba S3003	MG946R	HD1501MG
Operating Voltage	+5V	+5V	+5V	+6V
Torque	2.5 kg/cm	4.1 kg/cm	13 kg/cm	17 kg/cm
Gear Type	Plastic	Plastic	Metal	Metal
Rotation	0° - 180°	0° - 180°	0° - 180°	0° - 180°
Weight	9g	37g	55g	60g
Current (idle)	5mA (4.8V) 6mA (6V)	7.2mA (4.8V) 8mA (6V)	10mA	4mA (4.8V) 5mA (6V)
Current (no load)	100mA (4.8V) 120mA (6V)	100mA	170mA	400mA (4.8V) 500mA (6V)
Current (stall current)	700mA (4.8V) 800mA (6V)	<1000mA	1200mA	2300mA (4.8V) 2500mA(6V)

3.3.2 Arduino Uno R3

The Arduino UNO Rev3 board is a microcontroller that is based on the Atmega32P. The board is commonly used in project as it is easy to upload and run programmer in the controller. The board runs on a 16MHz crystal oscillator and consists of mainly of 14 digital input/output pin, 6 analog inputs, USB input, reset button and power supply input. The I/O female header is provided to ease wire connection without the need of any soldering process. The Arduino UNO Rev 3 is an all-round microcontroller that is suitable for electronic projects. Table 3.2 shows the summary of Arduino UNO Rev3 specification.

Table 3.2 Arduino UNO Rev 3 specification

Parameter	Specification
Operating Voltage	5V
Input Voltage	7-12V
DC Current per I/O Pin	40 mA
Flash Memory	32 KB (0.5 KB used by bootloader)
EEPROM	1 KB
Clock Speed	16MHz

3.3.3 Myoware Muscle Sensor

The Myoware Muscle Sensor is used for this project because it is capable of measuring muscle activation in terms of electric potential. It is a single channel sensor which can be used to extract EMG signal from a single group of muscle. The sensor features a built-in electrode connector which makes the Myoware wearable. There are several shields that can be directly mounted on the module such as the power shield, cable shield, proto shield and LED shield. The sensor can produce two types of output which are amplified raw EMG signal and EMG envelope which has already been amplified, rectified and integrated. The adjustable gain potentiometer located on the module is used to adjust the gain of EMG signal. Figure 3.3 shows the Myoware muscle sensor and Table 3.3 shows the specifications of Myoware muscle sensor

Figure 3.3 Myorware muscle sensor

Table 3.3 Specification of Myoware muscle sensor

Parameter		Min	TYP	Max
Supply voltage		+2.9V	+3.3V / +5.5V	+5.7V
Supply current		-	9mA	14mA
Output signal voltage	EMG Envelope	0V	-	+Vs
	EMG Raw	0V	-	+Vs
Input bias		-	1pA	-

3.3.4 Robotic Arm

The robotic arm joints are connected by steel frames which have slots that fits the servo motor. The manipulator has a total of 4 degree of freedom not including the gripper. It has 5 type of rotation which is base rotation, shoulder rotation, elbow rotation, wrist rotation and gripper rotation. The servo motors are attached to each joint based on the amount of torque it can produce. Figure 3.5 shows the representation of each joint with the arm motion.

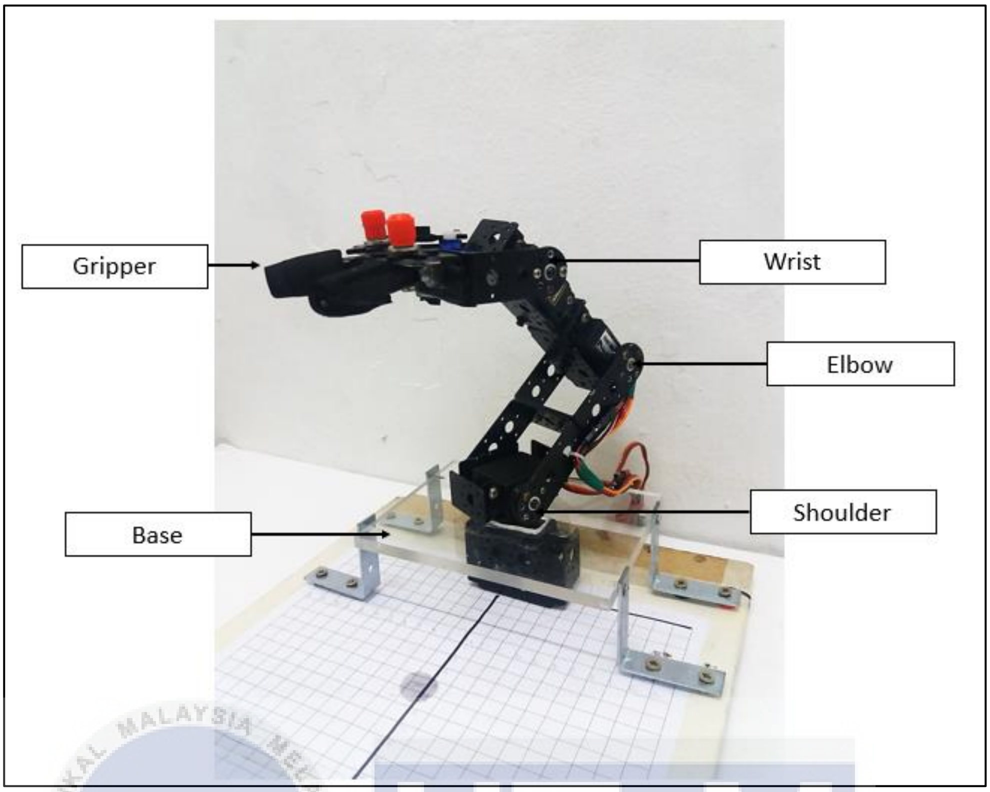


Figure 3.4 Robotic arm

3.3.5 Robotic Arm Circuit Design

The robotic arm has a total of two inputs (reset button, Myoware muscle sensor) and 6 outputs (servo motors, serial monitor). A 5V, 7A AC to DC power supply is used to power the servo motors. The Arduino UNO board is powered by a USB cable that is connected to the computer and the Myoware muscle sensor is connected to the 5V supply pin on the board. Servo motor MG946R, HD 1501 MG, Futaba S3003 and SG90 are connected to pin D9, D10, D11, D12 and D13 respectively while the sensor is connected to analog pin A0. A reset button is prepared to reset the robot to its original position. Figure 3.5 shows the circuit diagram of robotic arm.

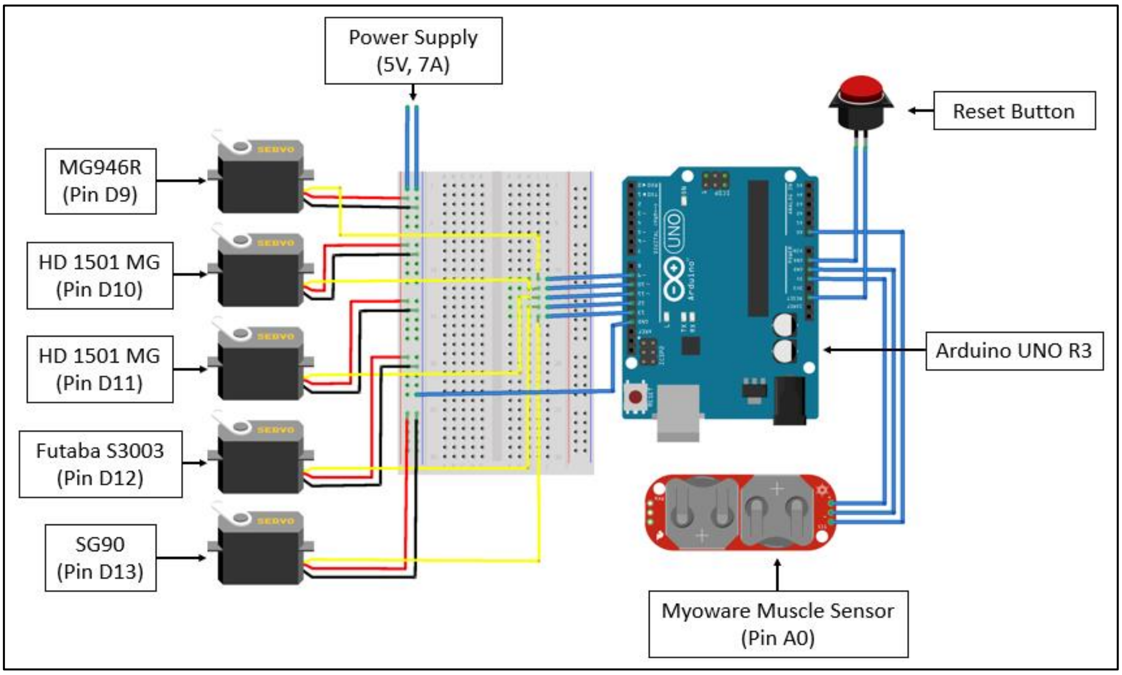


Figure 3.5 Robotic arm circuit diagram

3.3.6 Robotic Arm Source Code Developement

The source code consists of two main subroutines which are the motor selection subroutine and motor run subroutine. Initially, the robotic arm will move to its original position which has already been pre-programmed and will return to its original position whenever the reset button is pushed. Hand gesture is used to navigate through the program window. For each function, the EMG signal value (bits) are read from the analog input of the controller. The value is printed on the serial window that enable the user to view it. The program is also able to read the final value of the of each angle of rotation of the servo motor in the motor selection subroutine. Figure 3.6 shows the flowchart of the main program process flow.

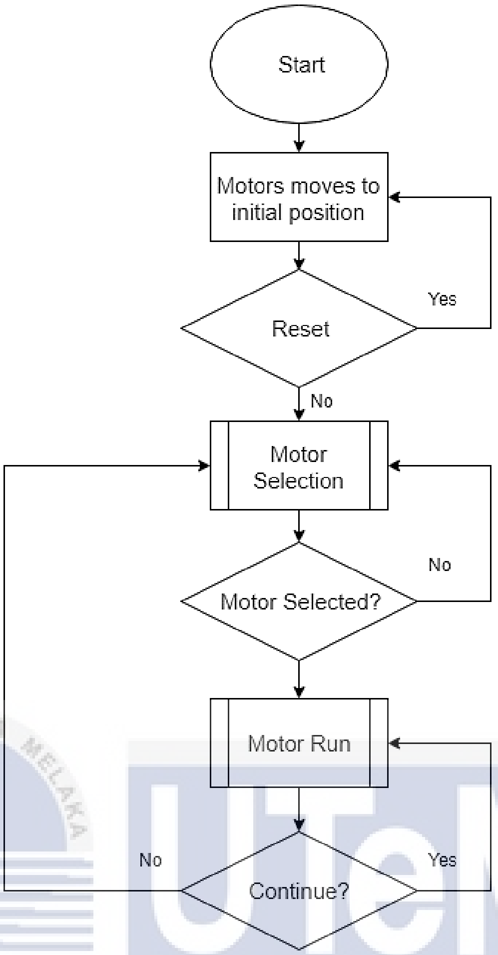


Figure 3.6 Main program process flow

3.3.6.1 Motor Selection Subroutine

The motor selection subroutine starts by reading value from the Myoware muscle sensor. When there is a gesture performed that produce signals that is within the range, the motor value will increase which also display the current selected servo motor. Servo motor MG946R, HD 1501 MG A, HD 1501 MG B, Futaba S3003 and SG90 are set to motor 1,2,3,4 and 5 respectively. Motor 0 is used to view the angle of each motor by subtracting the current angle with the initial angle read by the controller. The program will then continue with the motor run subroutine. If the signals exceed the threshold, the program will choose the current servo motor selected and exits the subroutine. Figure 3.7 and 3.8 shows the flowchart and source code for the motor selection subroutine.

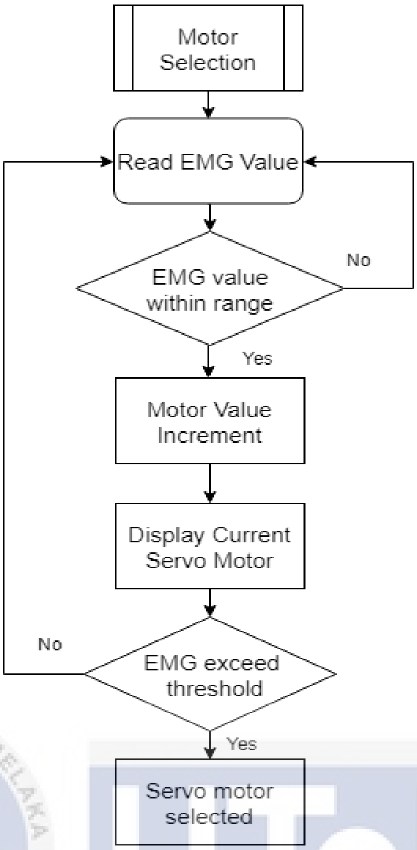
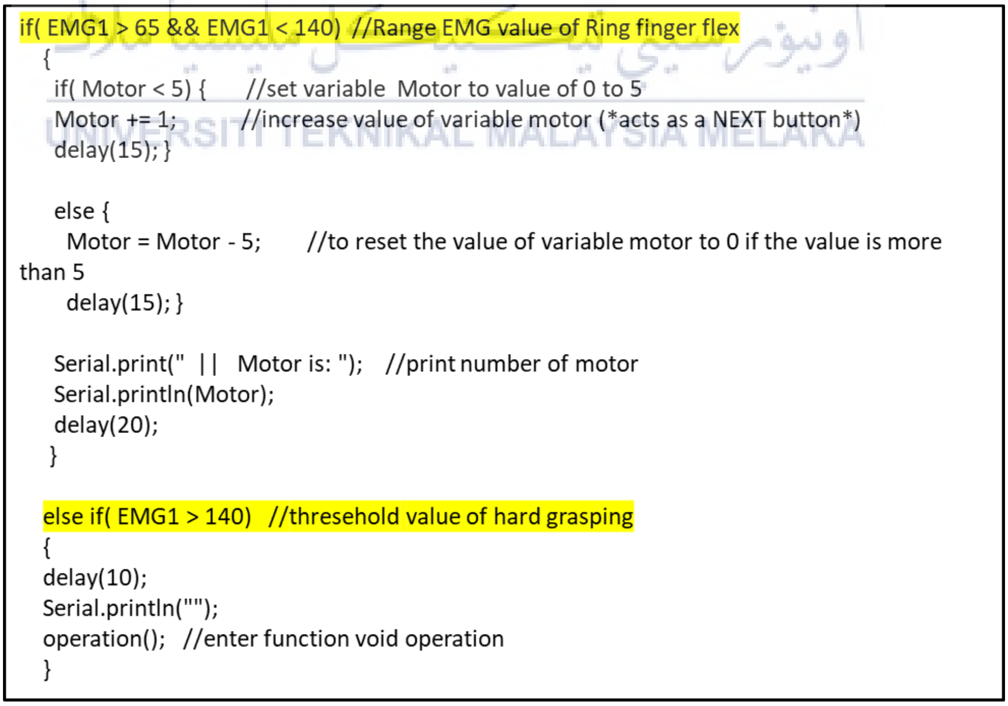


Figure 3.7 Motor selection subroutine



```
if( EMG1 > 65 && EMG1 < 140)  //Range EMG value of Ring finger flex
   {
    if( Motor < 5) {     //set variable  Motor to value of 0 to 5
    Motor += 1;         //increase value of variable motor (*acts as a NEXT button*)
    delay(15); }

    else {
      Motor = Motor - 5;      //to reset the value of variable motor to 0 if the value is more
than 5
      delay(15); }

    Serial.print("  ||   Motor is: ");    //print number of motor
    Serial.println(Motor);
    delay(20);
   }

  else if( EMG1 > 140)   //thresehold value of hard grasping
  {
  delay(10);
  Serial.println("");
  operation();   //enter function void operation
  }
```

Figure 3.8 A section of the source code for the motor selection subroutine

3.3.6.2 Motor Run Subroutine

The program starts by reading the EMG value from the Myoware muscle sensor. If the EMG value is within range A, the program will display the increment of the angle value of the selected servo motor on the serial monitor. The motor will rotate clockwise based on the value displayed. If the EMG value is within range B, the program will display the decrement of the angle value of the selected servo motor on the serial monitor. The motor will then rotate anticlockwise based on the value displayed. Lastly, if the EMG value exceeds the threshold value, then the program will exit the subroutine. Figure 3.9 and Figure 3.10 shows flowchart and source code for motor run subroutine respectively.

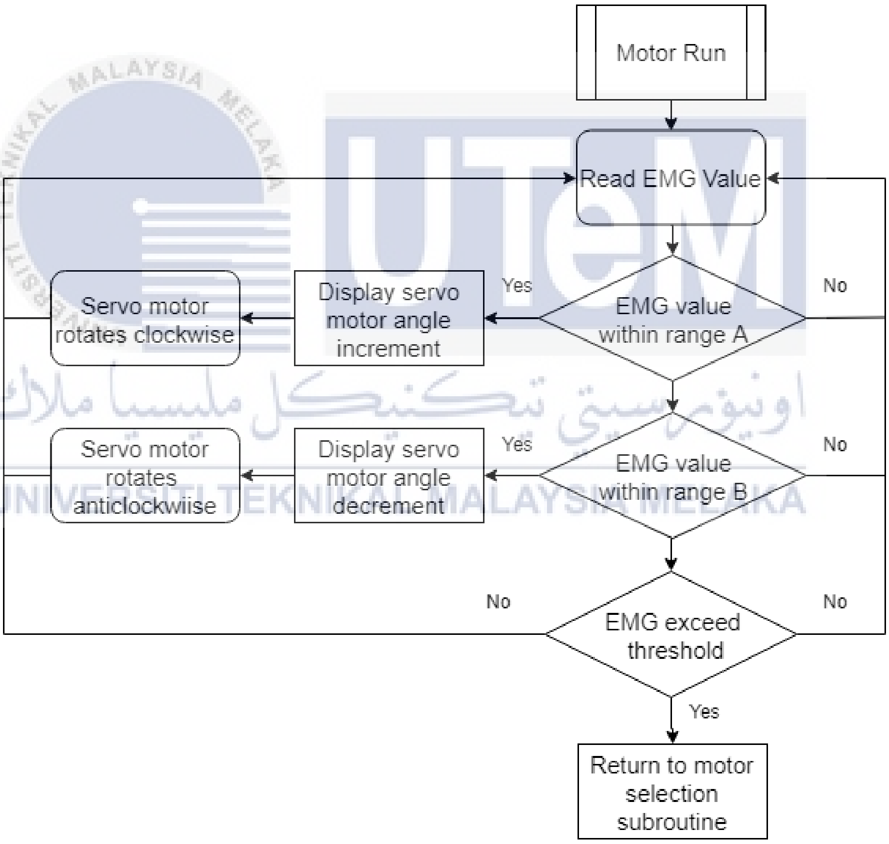
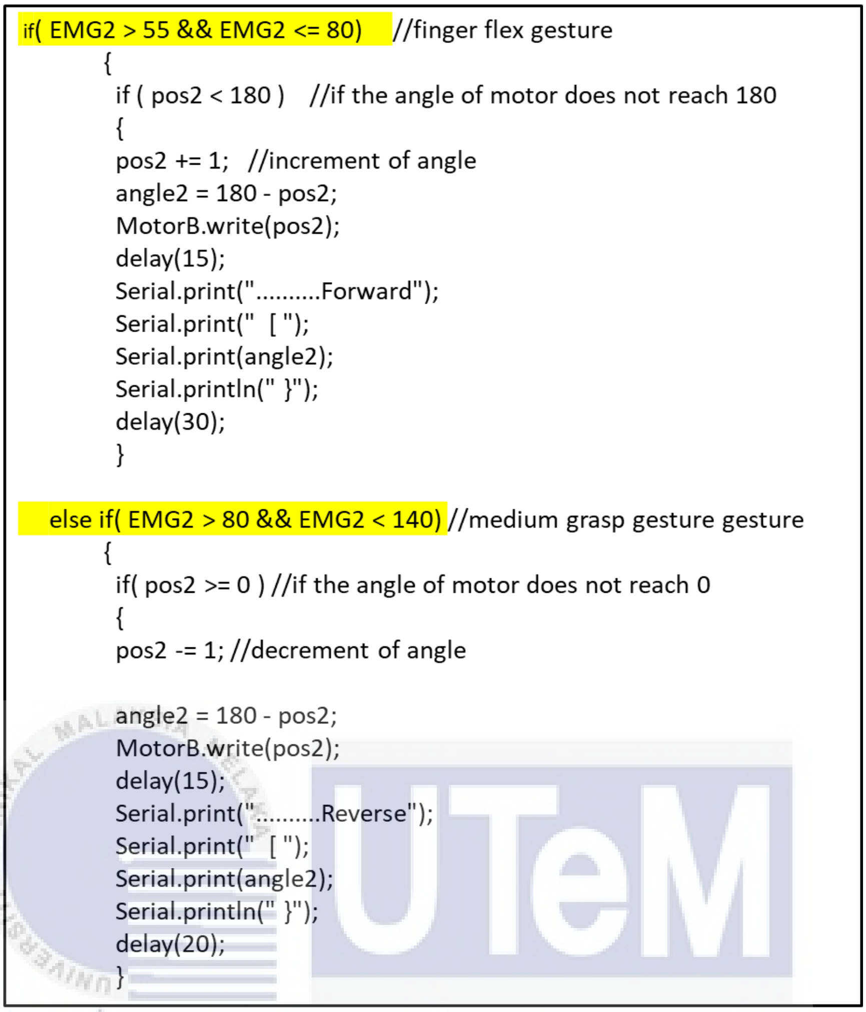


Figure 3.9 Motor run subroutine flowchart



```
if( EMG2 > 55 && EMG2 <= 80)    //finger flex gesture
         {
          if ( pos2 < 180 )    //if the angle of motor does not reach 180
          {
          pos2 += 1;   //increment of angle
          angle2 = 180 - pos2;
          MotorB.write(pos2);
          delay(15);
          Serial.print(".........Forward");
          Serial.print("  [ ");
          Serial.print(angle2);
          Serial.println(" }");
          delay(30);
          }

   else if( EMG2 > 80 && EMG2 < 140) //medium grasp gesture gesture
         {
          if( pos2 >= 0 ) //if the angle of motor does not reach 0
          {
          pos2 -= 1; //decrement of angle

          angle2 = 180 - pos2;
          MotorB.write(pos2);
          delay(15);
          Serial.print(".........Reverse");
          Serial.print("  [ ");
          Serial.print(angle2);
          Serial.println(" }");
          delay(20);
          }
```

Figure 3.10 A section of the source code for the motor run subroutine

3.4 Task 2: Developing the Forward Kinematic of 4-DOF Robotic Arm

The forward kinematic is used to obtain the coordinate of end effector and to determine the workspace of the manipulator. The angles rotation of each joints is used to compute the coordinate by using the transformation matrix equation. Figure 3.10 shows the robotic arm with each of its joint labelled with axis.

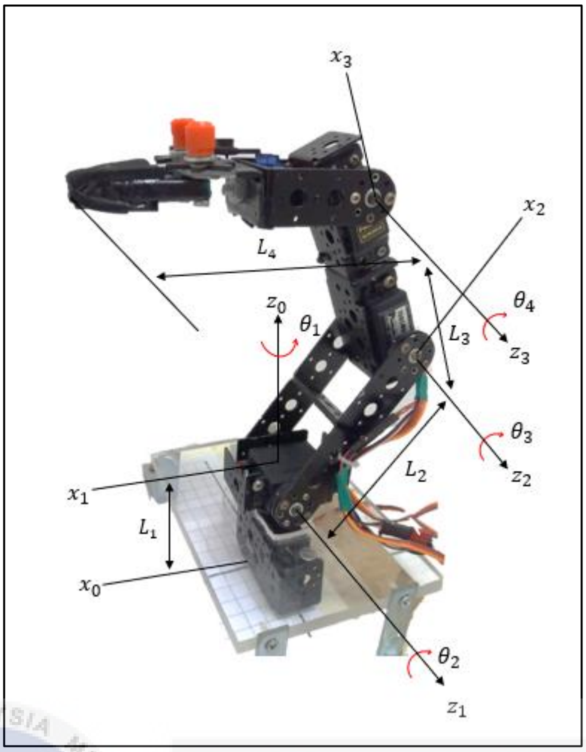


Figure 3.11 The robotic arm axis

The Denavit–Hartenberg (DH) parameter of the robotic is obtained by inserting the correct parameter in the DH parameter table. There are four parameters that is needed to be considered. Table 3.4 shows the definition of each parameters.

Table 3.4 DH parameters definition

Parameters	Definition
d	The distance of the previous x-axis to the current x-axis along the z-axis
θ	The angle of rotation about the z-axis between the previous x-axis to the current x-axis
a	The length between the between the previous z-axis to the current z-axis along the x-axis
α	The angle of rotation about the x-axis between the previous z-axis to the current z-axis

By referring to Figure 3.11, the DH parameters of the 4-DOF robotic arm is developed. Table 3.5 shows the DH parameters of the manipulator. The parameters are then inserted into Equation (3.1) which is the general formula of transformation matrix. The ${}^{0}_{4}T$ matrix is obtained as show in Equation (3.2). Cos and sin are substituted with c and s respectively to simplify the equation.

Table 3.5 DH parameter of 4-DOF robotic arm

Frame (i)	θ_i	α_i	a_i	d_i
1	θ_1	90	0	L_1
2	θ_2	0	L_2	0
3	θ_3	0	L_3	0
4	θ_4	0	L_4	0

$$ {}^{i-1}T_i = \begin{bmatrix} cos\theta_i & -sin\theta_i cos\alpha_i & sin\theta_i sin\alpha_i & a_i cos\theta_i \\ sin\theta_i & cos\theta_i cos\alpha_i & -cos\theta_i sin\alpha_i & a_i sin\theta_i \\ 0 & sin\alpha_i & cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.1} $$

$$ {}^{0}_{4}T = \begin{bmatrix} c_1 c_{123} & -c_1 s_{234} & s_1 & c_1(L_2 c_2 + L_3 c_{23}) + L_4 c_1 c_{234} \\ s_1 c_{123} & -s_1 s_{234} & c_1 & s_1(L_2 c_2 + L_3 c_{23}) + L_4 s_1 c_{234} \\ s_{234} & c_{234} & 0 & L_1 + L_2 s_2 + L_3 s_{23} + L_4 s_{234} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.2} $$

3.5 Task 3: Sensor calibration and Electrode Positioning

The sensor preparation is important to get a good reading of EMG signals from the muscle. The sensors need to be placed in the center of the selected muscle which in this project is the Flexor digitorum superficialis (FDS) muscle is chosen. Figure 3.12 shows the area of FDS muscle and the position of each electrode. The gain potentiometer can be calibrated after placing the electrode using a screwdriver.

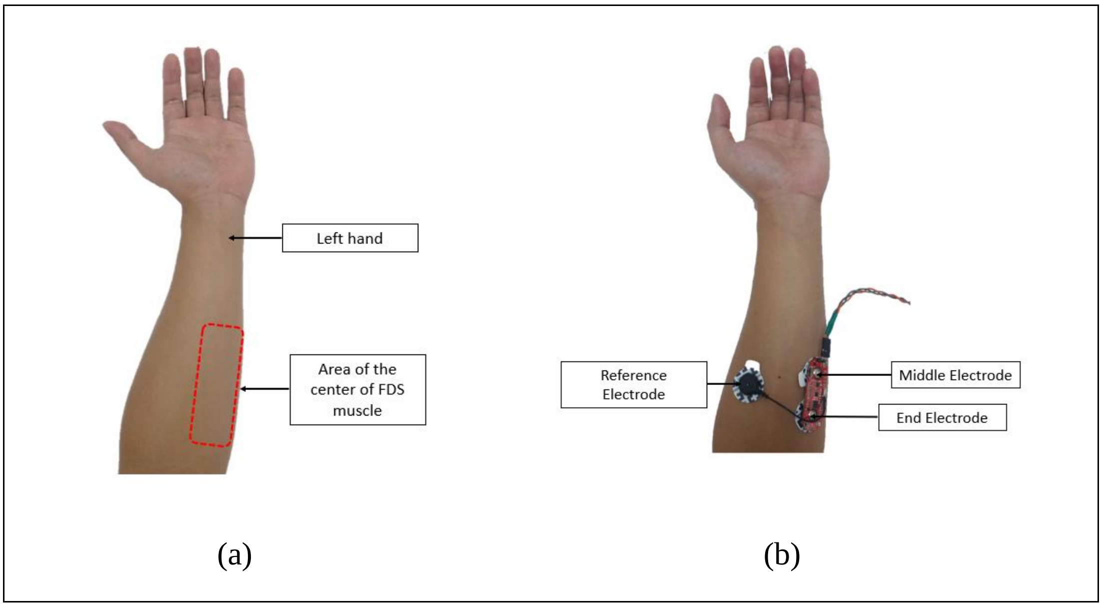


Figure 3.12: Placement of EMG sensor (a) area of placement (b) electrode positioning

Precaution needs to be taken before and after placing the electrode onto the skin as this will affect the signal output of the sensor. The excess oil from the skin needs to be cleaned. The usage of a hand sanitizer is able remove the oil. Thin hair on the arm should be shaved to add more contact between the electrode with skin. The body needs to be properly isolated from the floor to reduce the noise. The usage of velcro tape is advisable as it can help secure the sensor in place. Avoid using the sensor in a warm room as it will cause the skin to sweat. Instead, a well air-conditioned room can help the adhesive to stick better. The laptop charger needs to be unplugged from the laptop if the signal is unstable

3.6 Experiment 1: Error percentage of Motor Test

The aim of this experiment is to test the error percentage of each servo motor used in the robotic arm. This experiment also aims to partly achieve the first objective which is to design and develop a functional 4-DOF robotic arm that is controlled using EMG signals. For this experiment, each motor will be rotated from 0° to 180° for 10 times. The mean value will be calculated using Equation (3.3). The mean value will then be substituted as the actual value in Equation (3.4) to calculate the error percentage

$$\bar{x} = \frac{\Sigma f x}{n} \tag{3.3}$$

$$\% \ Error = \left(\frac{Theoretical\ value - Actual\ value}{Theroretical\ value}\right) x100\% \tag{3.4}$$

3.6.1 Experiment 1 Setup

The equipment and components used in this experiment are servo motor SG90, servo motor Futaba S3003, servo motor MG946R, servo motor HD 1501MG A and B, push button, Arduino UNO Rev3, USB cable and an angle sheet. The connection of each electrical component can be referred in Table 3.6. The circuit diagram for Experiment 1 can be referred in Figure 3.13. Figure 3.14 shows the setup before and after the angle sheet is placed on top.

Table 3.6 Experiment 1 connection scheme

	Input / Output	Connection
1.	Arduino pin	
	• +5v supply	Connected to supply wire of servo motor
	• +3.3 supply	Connected to positive wire of push button
	• Ground	Connected to push button and servo motor ground wire
	• Pin D8	Connected to servo motor data wire
	• Pin D3	Connected to push button data wire
2.	Arduino USB Port	Connected to the laptop (enabled for power supply)

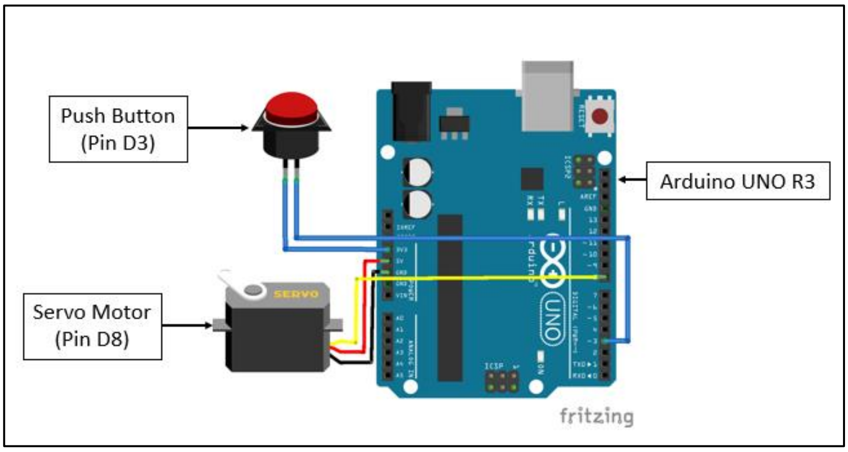


Figure 3.13 Experiment 1 circuit diagram

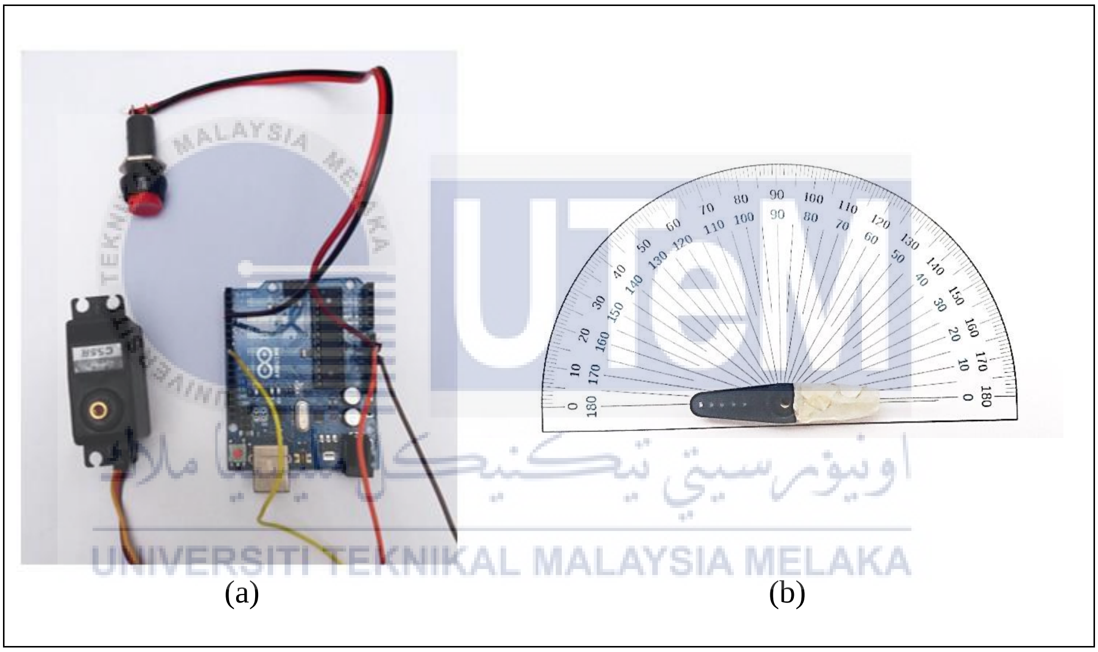


Figure 3.14 Experiment 1 Setup (a) components setup (b) angle sheet place in between servo motor and servo horn

3.6.2 Procedure

The experiment is constructed as shown in Figure 3.14. The angle sheet is placed in between the servo motor shaft and servo horn. The horn is then adjusted to the value 0°. After angle sheet is perfectly placed, the push button is pressed, and the motor rotates from 0° to 180°. The angle reading on the angle sheet is taken based on the last position of the servo horn and tabulated. The motor rotated automatically back to 0° after 5 seconds. The angle sheet is adjusted manually if the value is not equal to 0° The angle reading on the angle sheet is taken based on the last position of the servo horn and tabulated. The experiment is repeated for 10 rails. The servo motor is changed after the previous motor completed 10 trials.

3.7 Experiment 2: EMG Signal Classification Test

The aim of this experiment is to determine the EMG signal produced by different types of gestures and choose the best gesture which can be used to control the robotic arm. In addition, this experiment focuses on partly achieving the second objective which is to perform analysis on the operation of the robotic arm based on the characteristic of EMG signal collected from the forearm. For this test, 9 hand gesture is used which are hand rest, thumb flex, index finger flex, middle finger flex, ring finger flex, little finger flex, wrist extension, wrist flexion and grasping. The gestures are performed to produce EMG signals value (bits) which is recorded by the serial monitor. 20 samples are taken for each of the gesture. The graph of the EMG signals is plotted and classified.

3.7.1 Experiment 2 Setup

The components and equipment used for this experiment is Arduino UNO Rev3, Myoware muscle sensor, USB cable and laptop. The connection scheme is shown in Table 3.7. The circuit diagram and experiment are shown in Figure 3.15 and 3.16.

Table 3.7 Experiment 2,3 connection scheme

	Input / Output	Connection
1.	Arduino pin	
	• +5v supply	Connected to supply wire of Myoware muscle sensor
	• Ground	Connected to ground wire of Myoware muscle sensor
	• Pin A0	Connected to servo motor data wire
2.	Arduino USB Port	(enable for power supply and for viewing the serial monitor)

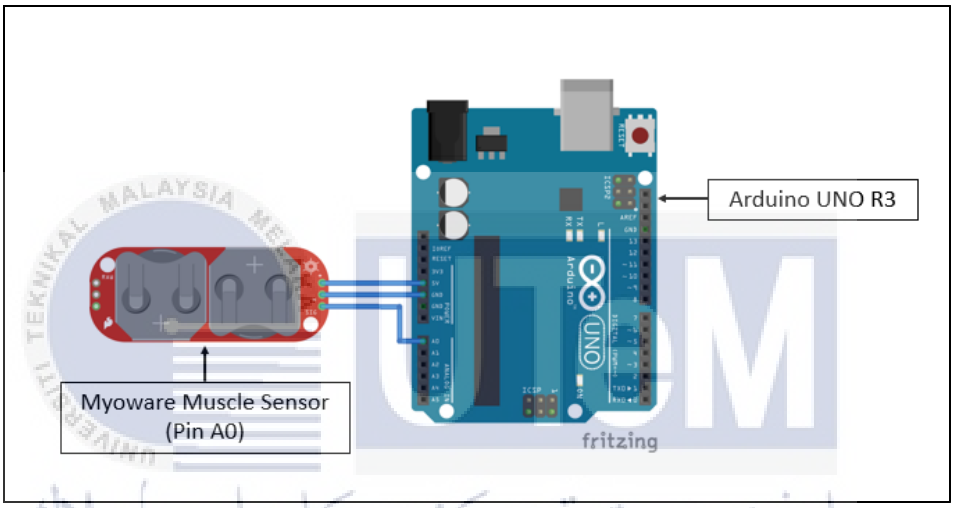


Figure 3.15 Experiment 2 circuit diagram

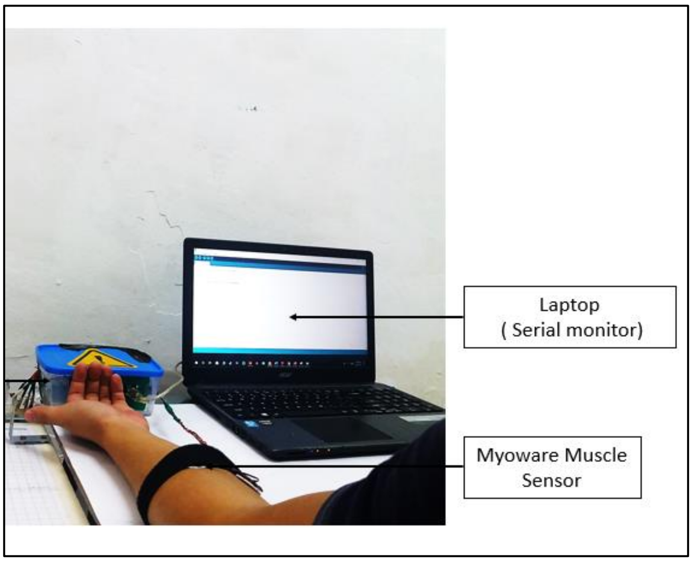
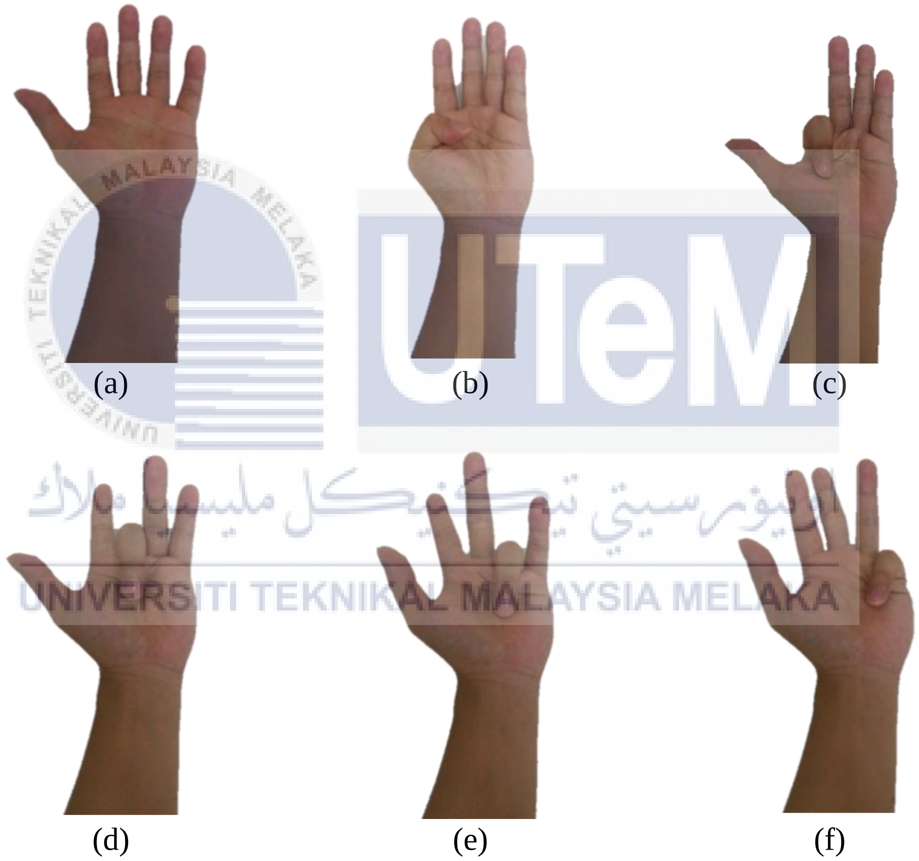


Figure 3.16 Experiment 2 setup

3.7.2 Procedure

The experiment is prepared as shown in Figure 3.16. The Myoware muscle is placed on the forearm (FDS muscle) and the USB cable is connected to the Arduino Uno and the laptop. The elbow is rested properly on the table and the hand rest gesture is performed for 20 seconds. The reading is observed and recorded. The experiment is repeated with hand gesture thumb flex, index finger flex, middle finger flex, ring finger flex, little finger flex, grasping, wrist extension and wrist flexion. The gesture can be referred in Figure 3.17. The EMG signals value(bits) is then tabulated into a table


(a) (b) (c)

(d) (e) (f)

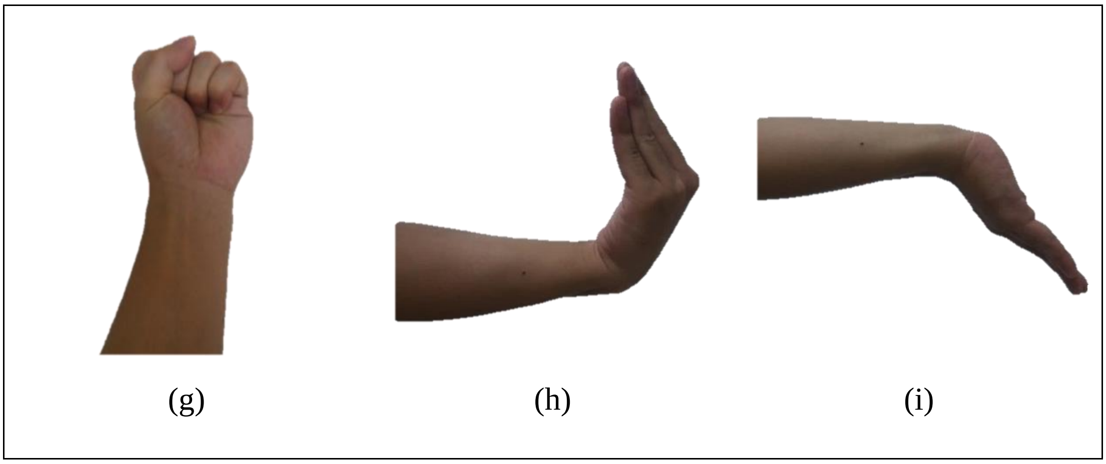

(g) (h) (i)

Figure 3.17 Hand gestures (a) Hand rest (b) Thumb flex (c) Index finger flex (d) Middle finger flex (e) Ring finger flex (f) Small finger flex (g) Grasp (h) Wrist flexion (i) Wrist extension

3.8 Experiment 3: Hand Gesture Validation Test

The objective of this experiment is to determine whether the hand gesture selected is accurate and usable with different sampling rate and arm position. In addition, this experiment is conducted to partly achieve the second objective which is to perform analysis on the operation of the robotic arm based on the characteristic of EMG signal collected from the forearm. For this experiment, the experiment setup is the same as in experiment 2 which is shown in Figure 3.16. Four gesture chosen for this test which are the hand rest, ring finger flex, medium grasp and hard grasp. 20 samples are taken for each of the sample

3.8.1 Procedure

The experiment is prepared the same as shown in Figure 3.15 and Figure 3.16. The Myoware muscle sensor is placed on the forearm and the USB cable is connected to the laptop. For the first test, the 20ms sample rate is used. The hand rest gesture, ring finger flex, medium grasping and hard grasping is performed to gather up to 20 sample for each gesture. The EMG signals values are recorded and tabulated. The experiment is repeated with 1000ms sampling rate and 20ms sampling rate with arm hovering.

3.9 Experiment 4: EMG Signal to Servo Motor Rotation Correlation Test

The aim of this experiment is to determine whether the robotic arm responds consistently with the EMG signal from the chosen gesture. In addition, the experiment is to achieve the second objective which is to perform analysis on the operation of the robotic arm based on the characteristic of EMG signal collected from the forearm. The experiment is done by moving each joint with 45° clockwise and 45° anticlockwise. The corresponding EMG signal value versus the angle rotation is then analyzed.

3.9.1 Experiment 4 Setup

The equipment used in this experiment is Arduino UNO Rev3, Myoware muscle sensor, laptop, USB cable, 4-DOF robotic arm. The connection schemes can be referred in Table 3.8. Figure 3.18 and 3.19 shows the experiment setup.

Table 3.8 Experiment 4,5,6 connection scheme

	Input / Output	Connection
1.	Arduino pin	
	+5v supply	• Connected to supply wire of Myoware muscle sensor
	Ground	• Connected to ground wire of Myoware muscle sensor and reset button
	Pin A0	• Connected to data wire of Myoware muscle sensor
	Pin D9	• Connected to data wire of servo motor MG946R
	Pin D10	• Connected to data wire of servo motor HD 1501 MG
	Pin D11	• Connected to data wire of servo motor HD 1501 MG
	Pin D12	• Connected to data wire of servo motor Futaba S3003
	Pin D13	• Connected to data wire servo motor SG90
2.	Arduino UNO USB port	• (enable for power supply and for viewing the serial monitor)
3.	Power Supply	
	+5V supply	• Connected to supply wire of servo motors
	Ground	• Connected to ground wire of servo motors

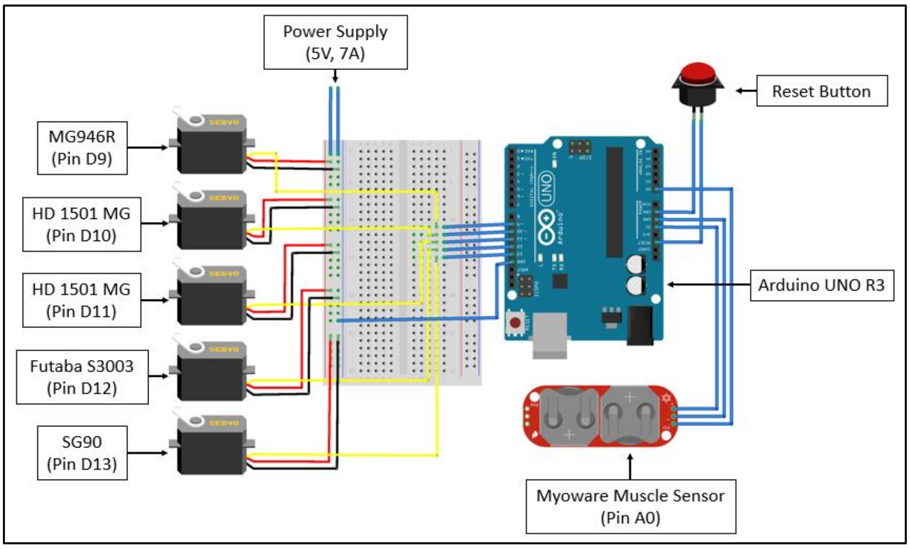


Figure 3.18 EMG signal to servo motor rotation correlation test circuit diagram

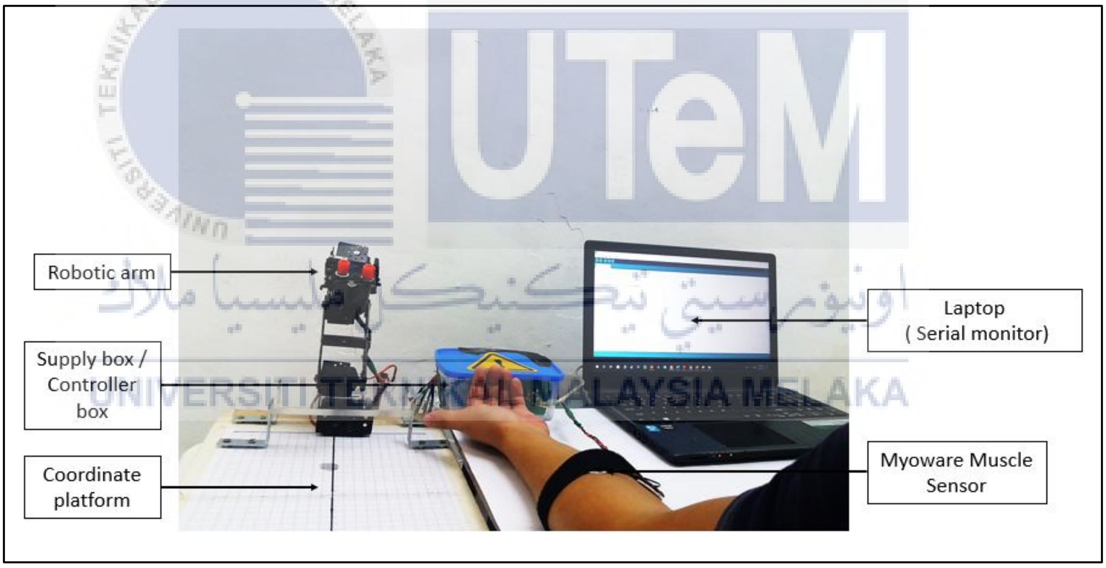


Figure 3.19 EMG signal to servo motor rotation correlation test

3.9.2 Procedure

The experiment is prepared as shown in Figure 3.19. The Myoware muscle sensor is placed on the forearm and the USB cable is connected from the control box to the laptop. Two hand gesture is used to complete the anticlockwise and clockwise rotation which is the ring finger flex and medium grasping respectively. The rotation of the joint is 45° to 0° and 0° to 45°. The EMG signals value and the servo motor angle are recorded. The experiment is repeated for joint B, C, D and E.

3.10 Experiment 5: Robotic Arm Accuracy Test Using Forward Kinematic

The aim of this experiment is to determine whether the robotic arm can reach the target accurately. In addition, this experiment aims to achieve the third objective which is to test the effectiveness of the actuators based on the instruction given by the controller. The forward kinematic that has been developed is used to calculate the coordinates of the end effector. The angle of each motor joints is collected from a menu in the program code which show all the angle.

3.10.1 Experiment 5 Setup

The equipment and component that is used in this experiment is the control box, Myoware muscle sensor, laptop, USB cable, 4-DOF robotic arm, object A and B, the coordinate platform. Figure 3.20 is the object used for this experiment. Figure 3.20 and 3.21 is the setup of the experiment. The connection scheme is the same as in experiment 4 which is shown in Figure 3.18.

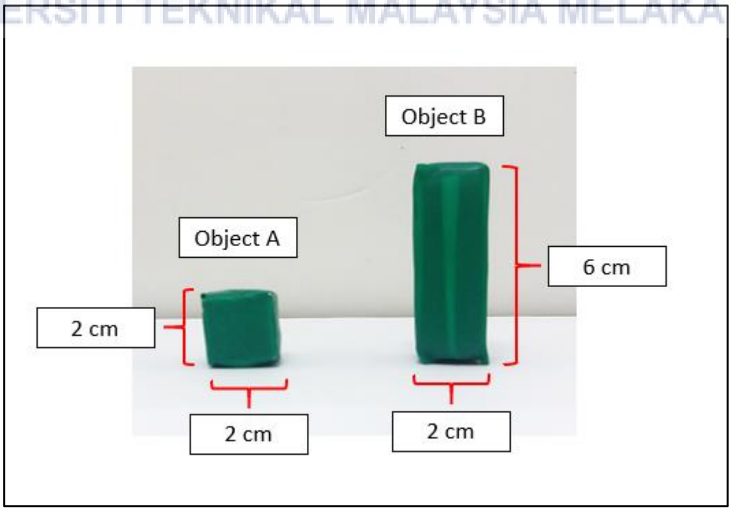


Figure 3.20 Object A and object B

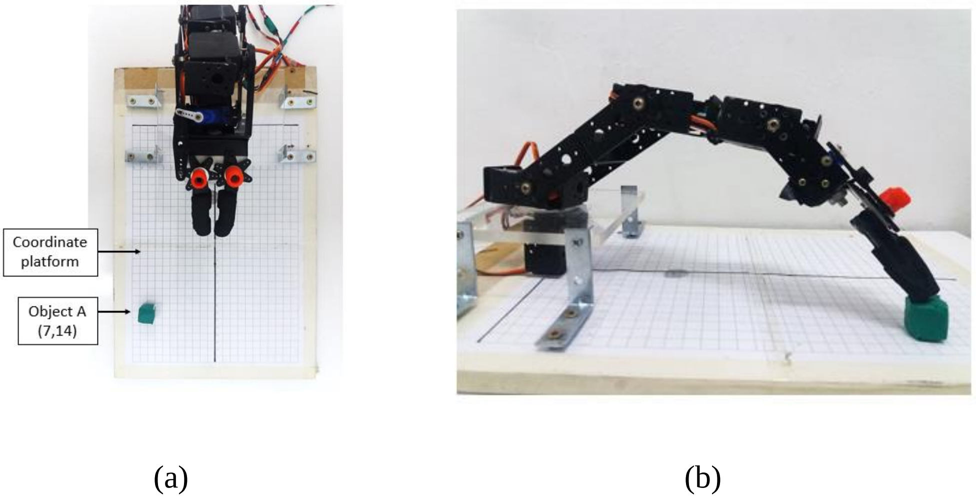


Figure 3.21: Experiment 5 setup (a) object location (b) target

3.10.2 Procedure

The experiment is prepared as shown in Figure 3.21. The Myoware muscle sensor is placed on the forearm and the USB cable is connected to the control box and the laptop. The robotic arm is controlled using the hand gesture ring finger flex, medium grasp and hard grasp. The robotic arm is controlled to reach the top of object A at coordinate (7,14). The angles of each joints each recorded and tabulated. The task is performed for 3 attempts. After the attempt is completed object A is allocated to coordinate (-7,14) and (0,14). The experiment is repeated with object B with the same coordinate.

3.11 Experiment 6: Robotic Arm Performance Test

The aim of this experiment is to determine the time taken for the robotic arm to grab and place an object to a target location using manual control. In addition, , this experiment aims to achieve the third objective which is to test the effectiveness of the actuators based on the instruction given by the controller. The time taken for automatic controlled robotic arm will be used to compare with the manual control.

3.11.1 Experiment 6 Setup

The equipment and component that is used in this experiment is the control box, Myoware muscle sensor, laptop, USB cable, 4-DOF robotic arm, object C, target and the coordinate platform. Figure 3.22 is the object used for this experiment. Figure 3.3 is the setup of the experiment. The connection scheme is the same as in experiment 4 which is shown in Figure 3.18.

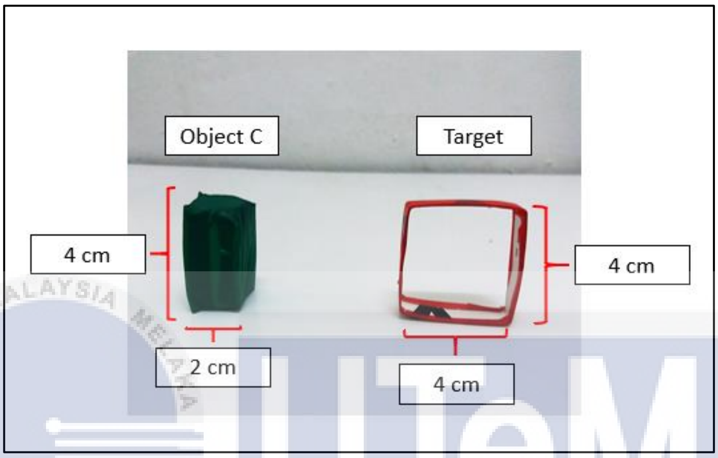


Figure 3.22 Object C and target

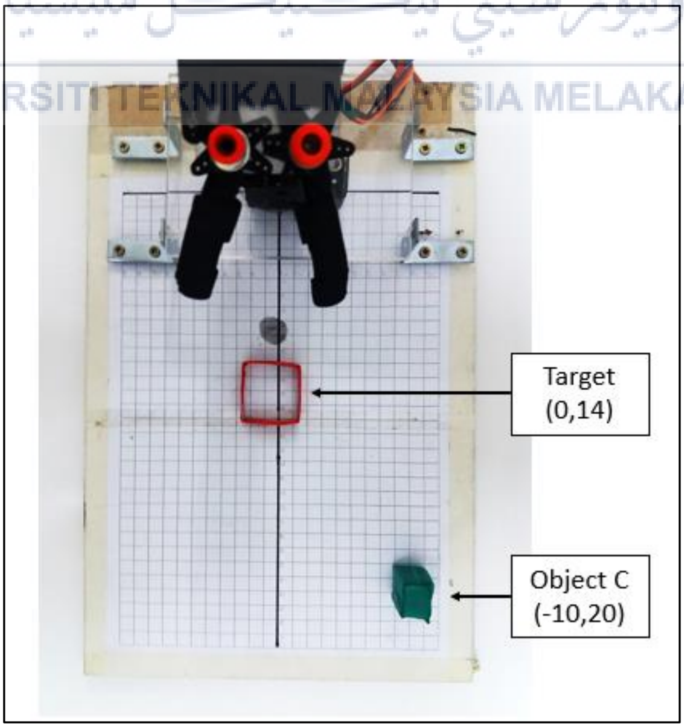


Figure 3.23 Experiment 6 setup

3.11.2 Procedure

The experiment is prepared as shown in Figure 3.23. The Myoware muscle sensor is placed on the forearm and the USB cable is connected to the control box and the laptop. The robotic arm is controlled using hand gesture ring finger flex, medium grasp and hard grasp. The robotic arm is controlled to grab object C at coordinate (-10,14) and place it to the target location at (0,14). The time taken to finish the action is taken and tabulated. The task is performed for 3 attempts and the mean time taken is calculated. After the manual control attempt is completed, the experiment is repeated using the automatic control. Experiment is repeated with coordinate (7,28), (-9,12) and (-10,20)

3.12 Objective Mapping

The task and experiment conducted in this project is designed to achieve the main objectives. Table 3.9 shows the task and experiment that has been mapped to the objective.

Table 3.9 Experiment mapped to objectives

	Objective 1	Objective 2	Objective 3
Task1: Robotic arm hardware development	√		
Task 2: Developing forward kinematic of the 4-DOF arm	√		
Task 3: Sensor calibration and electrode Positioning		√	
Experiment 1: Servo motor percentage error test	√		
Experiment 2: EMG signal classification test		√	
Experiment 3: Hand gesture validation test		√	
Experiment 4: EMG signal to servo motor rotation correlation test		√	
Experiment 5: Robotic arm accuracy test using forward kinematic			√
Experiment 6: Robotic arm performance test			√

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Experiment 1: Percentage Error of Motor Test

The Table 4.1 shows the number average angle of rotation for each motor based on 10 attempts.

Table 4.1 Percentage error of motor test

Attempt	SG90	Futaba S3003	MG946R	1501MG A	1501MG B
1	172	180	175	170	170
2	169	180	177	169	170
3	171	180	175	168	169
4	170	180	176	170	169
5	170	179	177	171	170
6	169	180	177	170	168
7	170	179	178	170	168
8	172	180	176	171	168
9	172	180	178	170	170
10	170	180	177	170	170
Average	170.5	179.8	176.6	169.9	169.4
% Error	5.3%	0.1%	1.9%	5.6%	5.9%

Based on the results, each servo motors have different percentage error. The servo motor with the highest amount of error is the 1501 MG (5.9%) while the lowest amount of error is the Futaba S3003(0.1%). Only two of the motors were below the 5% error threshold which are the Futaba S3003(0.1%) and MG946R (1.9%). However, the error percentage of SG90(5.3%), 1501MG A (5.6%) and 1501MG Bis still acceptable as it is still significantly close to the tolerance value. Therefore, the servo motors can be used for the robotic arm joint

4.2 Experiment 2: EMG Signal Classification Test

Based on the graph in Figure 4.1, the gesture which shows significant characteristic is the ring finger flex and grasping gesture. The ring finger flex gesture value range from 70-90 bits while the grasping gesture range from 90-130 bits. Another gesture which shows a lesser significant is the wrist extension gesture which range from 50-60 bits. The other gestures are less than 55 bits. The usable gesture based on this experiment is gesture ring finger flex and grasping because the EMG signal can be differentiated from the rest of the gestures.

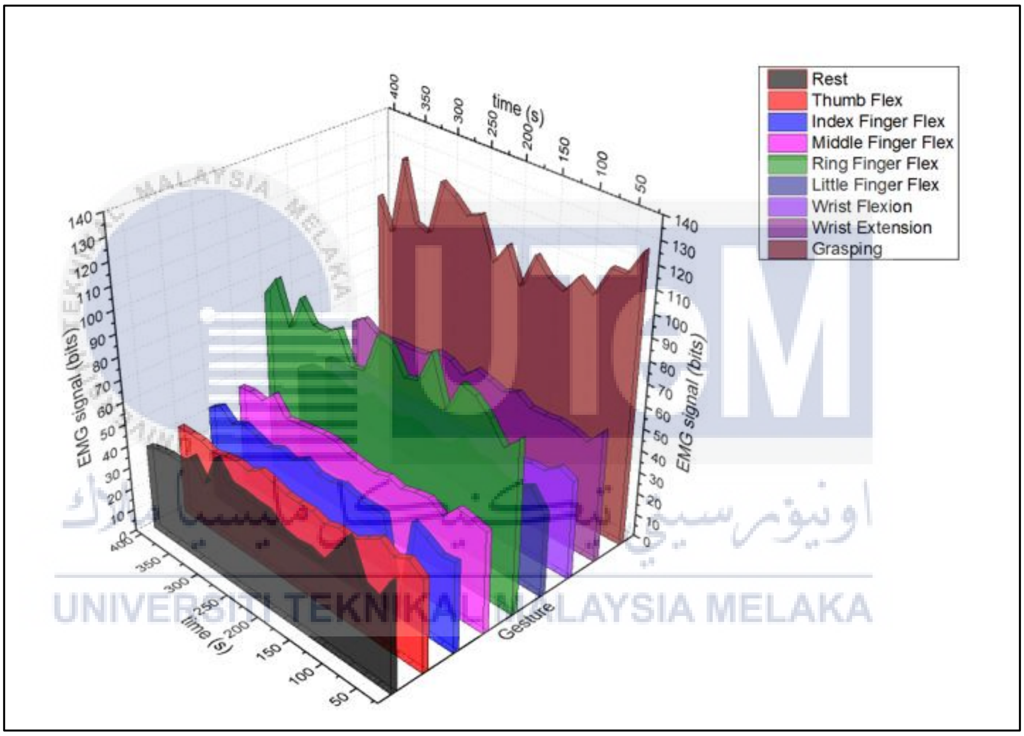


Figure 4.1 Graph of EMG signal vs time

The grasping gesture can be divided into two gesture which is the medium grasp and hard grasp because the EMG signal depends on the amount force exerted by the action. In addition, both gestures are very comfortable and natural to perform. The wrist extension is not chosen because the EMG signal value is very near to the minimum value and is harder to perform. The rest gesture is also chosen to act as the normal position where no activation is occurred. Therefore, 4 gesture has been chosen out of the 10 gesture to control the robotic arm which are the hand rest, ring finger flex, medium grasp and hard grasp gestures

4.3 Experiment 3: Hand Gesture Validation Test

The graph in Figure 4.2 shows that there 4 distinct EMG signal which is produced by performing 4 different gesture. The rest gesture produces the most stable gesture followed by the ring finger flex and medium grasp gesture while the hard grip gesture is moderately unstable. The range value of rest, ring finger flex, medium grasp and hard grasp is 40 – 44 bits, 69 – 92 bits, 91 – 129 bits, 192 – 306 respectively.

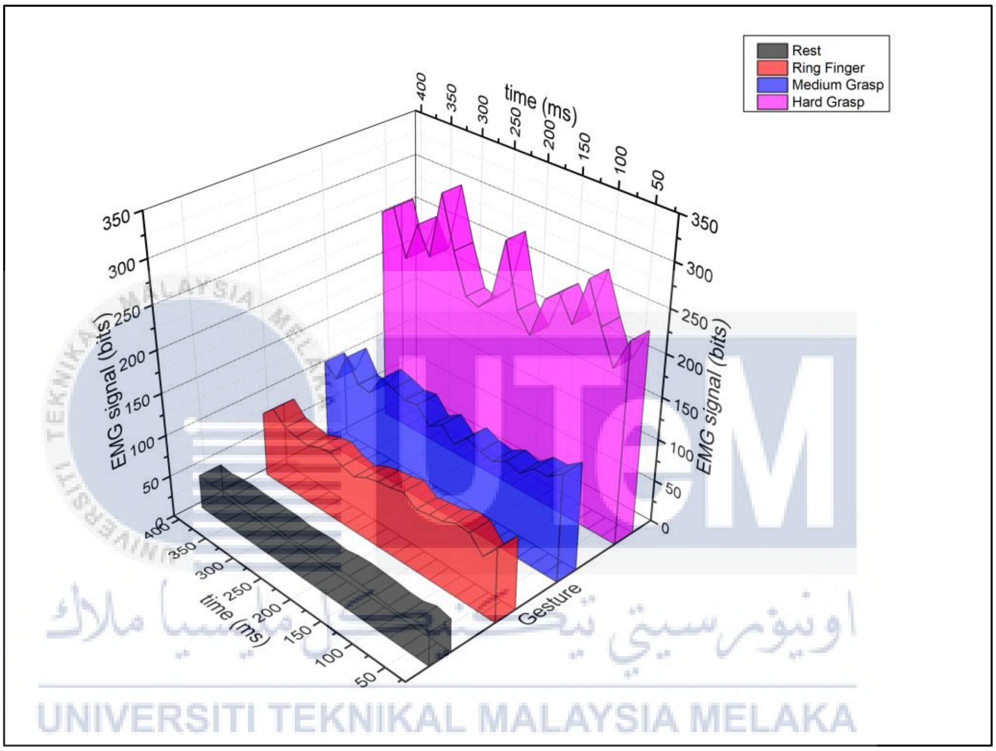


Figure 4.2 Graph of EMG signals versus time (30ms sampling rate)

The graph in Figure 4.3 shows that there 4 distinct EMG signal which is produced by performing 4 different gesture. The rest gesture, ring finger flex and medium grasp gesture produced a stable gesture but there occurs some signal spiking in some instances of the time frame. The hard grasp shows a more unstable graph than the graph in Figure 4.2. The range value of rest, ring finger flex, medium grasp and hard grasp is 38 – 57 bits, 66 – 115 bits, 105 – 132 bits, 141 – 306 respectively.

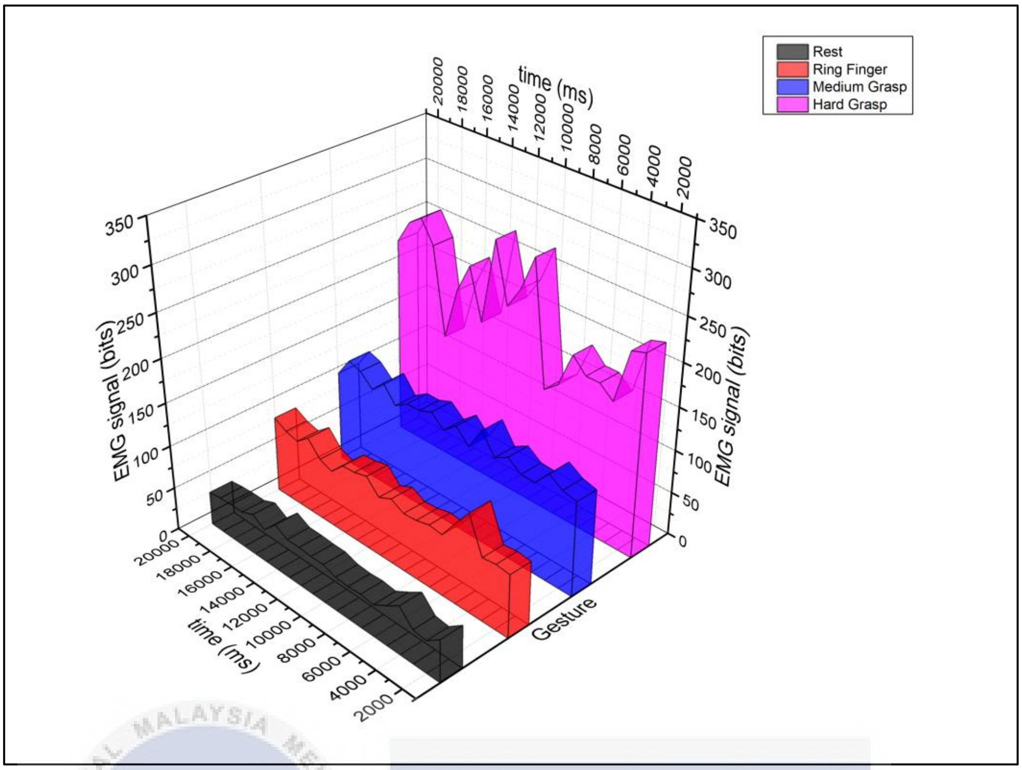


Figure 4.3 Graph of EMG signals versus time (1000ms sampling rate)

The Figure 4.4 shows that there 4 distinct EMG signal which is produced by performing 4 different gesture. The rest gesture is quite stable while ring finger flex and medium grasp gesture produces signals that overlaps with each other. The hard grasp signal is even more unstable than the graph in Figure 4.3. The range value of rest, ring finger flex, medium grasp and hard grasp is 40 – 52 bits, 69 – 112 bits, 116 – 182 bits, 102 – 328 respectively.

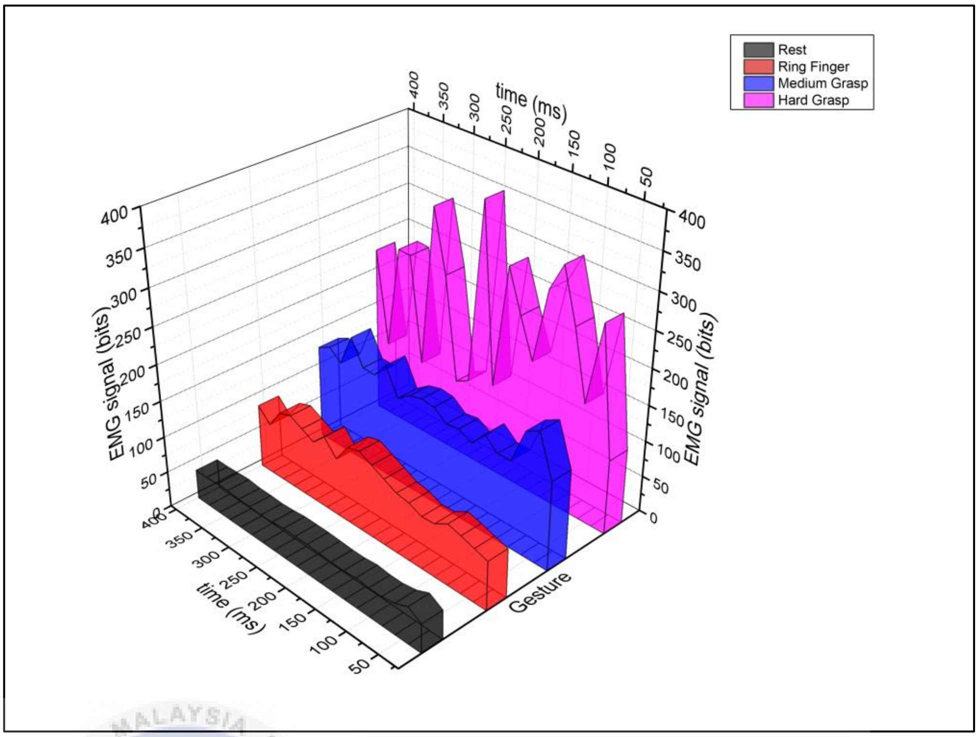


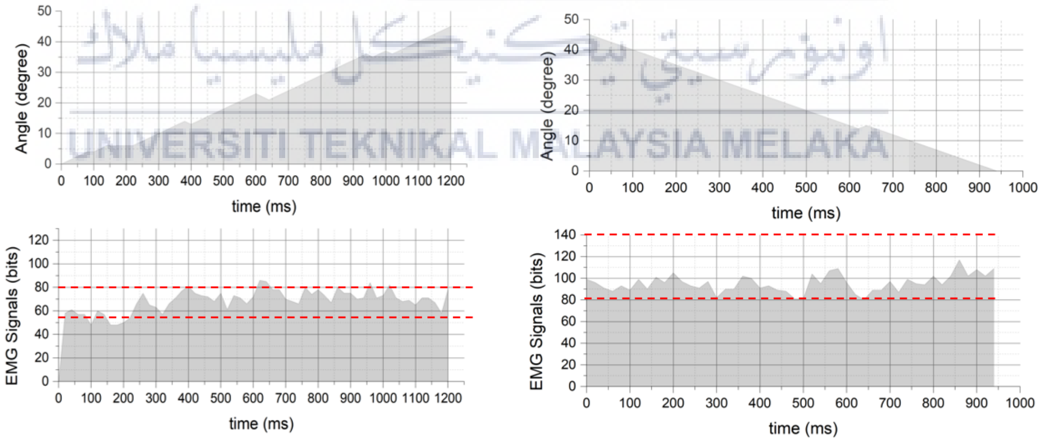
Figure 4.4 Graph of EMG signal versus time (30 ms and arm hovering)

Based on the results, the sampling rate and position of the arm can affect the EMG signal value. The standard deviation of EMG signals in condition A is smaller compared to the other two condition. In fact, the mean is just slightly higher or slightly lower than the mean in the other two condition. This shows that the distribution while maintains a consistent value of EMG signals. The graph in Figure 4.2 and Figure 4.3 are mostly identical except for the hard grasp gesture graph. The sampling rate affects by detecting unnecessary muscle action due to muscle fatigue. The standard deviation of EMG signals in condition C is quite large. This means that the distribution is not even. The arm movement produce more unwanted noise due to it not being restricted. Therefore, the four chosen gesture can still be used to control robotic arm because it still shows distinct EMG signal value. However, sample rate should be faster in which it is still readable in the serial monitor. The arm should also be restricted by resting the arm on flat surface.

4.4 Experiment 4: EMG Signal to Servo Motor Rotation Correlation Test

A total of five graph is produced from different table. Each joint is tested for EMG signal correlation with servo motor rotation. Servo motor A, B, C, D and E are servo motor MG946$, HD 1501 MG A, HD1501 MG B, Futaba S3003 and SG90 respectively

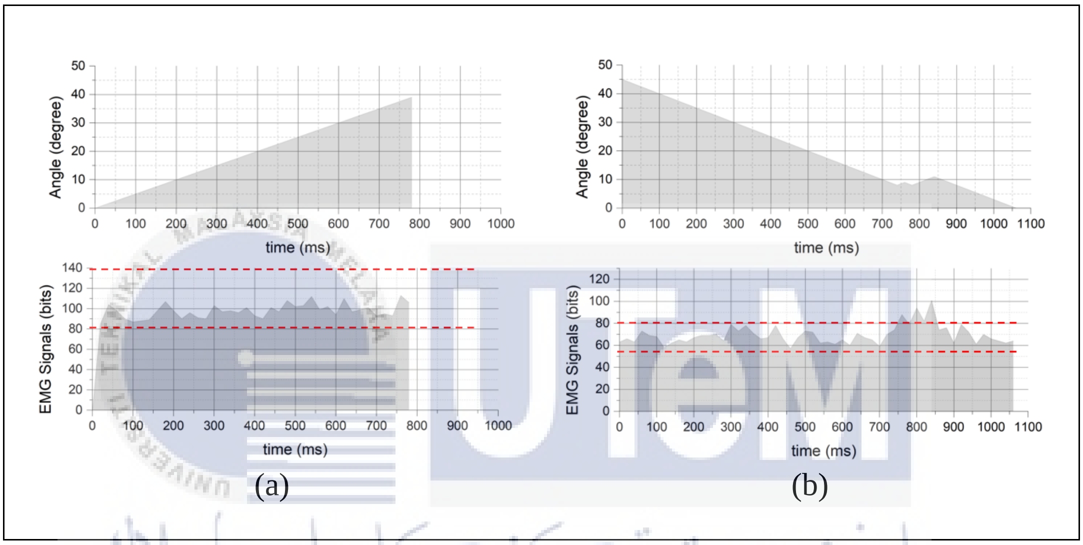
For servo motor A, the rotation is in the opposite direction. Based on the graph in Figure 4.5, the servo motor rotates clockwise when the EMG signals value is between 60 – 80 bits using the ring finger flex gesture and moves clockwise when the EMG signals value is between 80 -140 bits using the medium grasp gesture. There is some overshoot occur in the anticlockwise rotation which cause the rotation to drop by a few degrees. For the clockwise rotation, there is no overshoot but there some of the EMG signal did not reach the range which cause the servo motor to stop. The time taken to complete a 45° clockwise rotation is slower than the time taken to complete a 45° anticlockwise.



(a) (b)

Figure 4.5 The graph of EMG signals versus servo motor A angle
(a) Anticlockwise rotation (b) Clockwise rotation

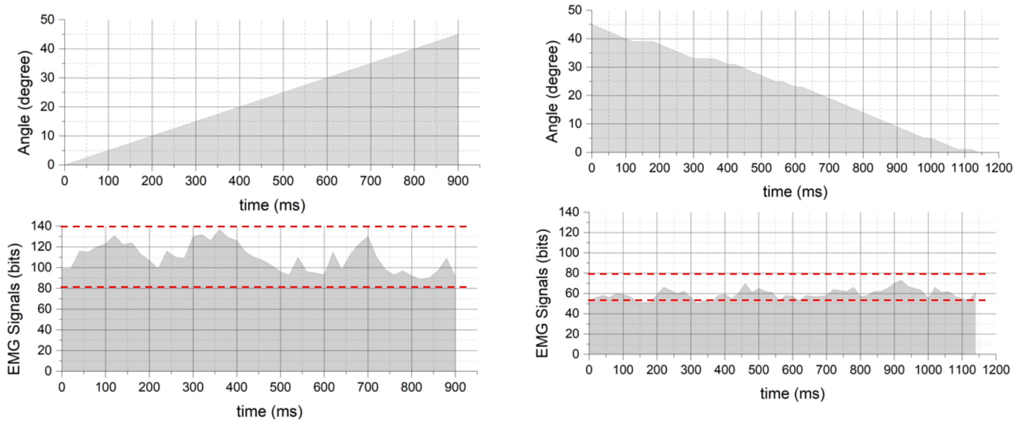
Based on the graph in Figure 4.6, the servo motor rotates clockwise when the EMG signals value is between 60 – 80 bits using the ring finger flex gesture and moves anticlockwise when the EMG signals value is between 80 -140 bits using the medium grasp gesture. For the anticlockwise rotation, the EMG signal value is consistently between the range of which makes the servo motor run smoothly without interruptions. For the clockwise rotation, there is overshoot that causes the motor to rotate anticlockwise. The time taken to complete a 45° clockwise rotation is faster than the time taken to complete a 45° anticlockwise



(a) (b)

Figure 4.6 The graph of EMG signals versus servo motor B angle
(a) Anticlockwise rotation (b) Clockwise rotation

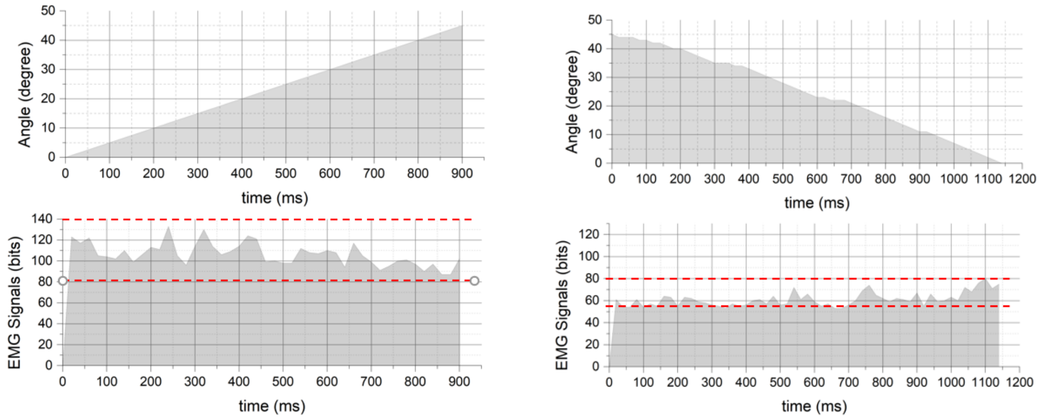
Based on the graph in Figure 4.7, the servo motor rotates clockwise when the EMG signals value is between 60 – 80 bits using the ring finger flex gesture and moves anticlockwise when the EMG signals value is between 80 -140 bits using the medium grasp gesture. For the anticlockwise rotation, the EMG signal value is consistently between the range of which makes the motor run smoothly without interruptions. For the clockwise rotation, there is no overshoot but some of the EMG signal did not reach the minimum range value. This causes the motor to stop rotating for some instances. The time taken to complete a 45° clockwise rotation is faster than the time taken to complete a 45° anticlockwise.



(a) (b)

Figure 4.7 The graph of EMG signals versus servo motor C angle
(a) Anticlockwise rotation (b) Clockwise rotation

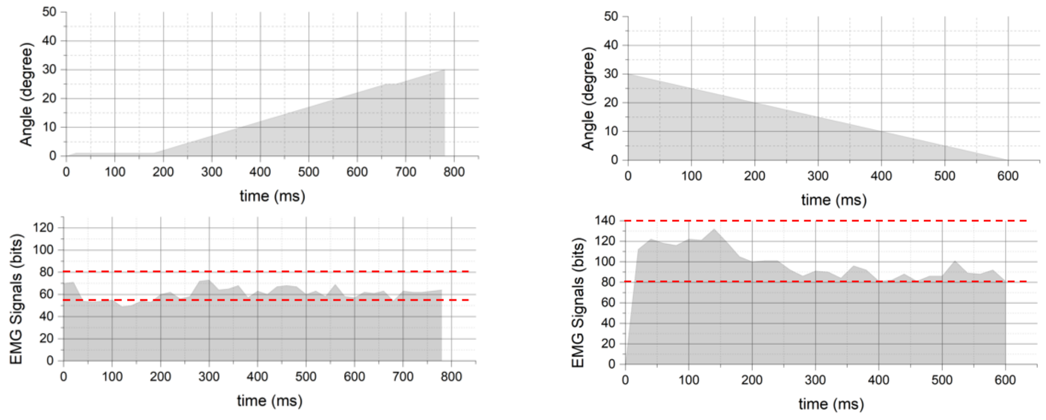
Based on the graph, the servo motor rotates clockwise when the EMG signals value is between 60 – 80 bits using the ring finger flex gesture and moves anticlockwise when the EMG signals value is between 80 -140 bits using the medium grasp gesture. For the anticlockwise rotation, the EMG signal value is consistently between the range of which makes the motor run smoothly without interruptions. For the clockwise rotation, there is no overshoot but some of the EMG signal did not reach the minimum range value. This causes the servo motor to stop rotating for some instances. The time taken to complete a 45° clockwise rotation is faster than the time taken to complete a 45° anticlockwise rotation



(a) (b)

Figure 4.8: The graph of EMG signals versus servo motor D angle
(a) Clockwise rotation (b) Anticlockwise rotation

For motor E, the servo motor rotates in the opposite direction. Based on the graph in Figure 4.9, the servo motor rotates clockwise when the EMG signals value is between 60 – 80 bits using the ring finger flex gesture and moves anticlockwise when the EMG signals value is between 80 -140 bits using the medium grasp gesture. For the clockwise rotation, the EMG signal value did not reach the minimum range value which cause the motor to stop rotating for a period. Afterwards, the motor continues to move smoothly. For the clockwise rotation, the EMG signal is consistently within the range value. The motor was able to run smoothly without interruption. The time taken to complete a 45° anticlockwise rotation is faster than the time taken to complete a 45° anticlockwise rotation



(a) (b)

Figure 4.9 The graph of EMG signals versus motor E Angle (a) Anticlockwise rotation (b) Clockwise rotation

Overall, the servo motor was able to rotate according to EMG signals value that is obtain for the muscle. The EMG signal value obtained using the medium grasp gesture is much more effective than the ring finger flex gesture which causes the motor to run smoother in less time. The servo motors can respond to the EMG signal but overshoot and undershoot may cause the motor to stop or rotates in the opposite direction.

4.5 Experiment 5: Robotic Arm Accuracy Test Using Forward Kinematic

Based on the graph in Figure 4.10, the red bar indicates the mean coordinates and the yellow bar indicates the actual coordinate. The mean coordinate for axis x is lower than the actual coordinate by 0.15 cm. The mean coordinate for axis y is higher than the actual coordinate by 0.96cm. The mean coordinate for axis z is higher than the actual coordinate by 0.04 cm. The average percentage error for axis x, y and z is 2.14%, 4% and 2% respectively. In conclusion the coordinate measurement from the forward kinematic is accurate with a 2.73% percentage error.

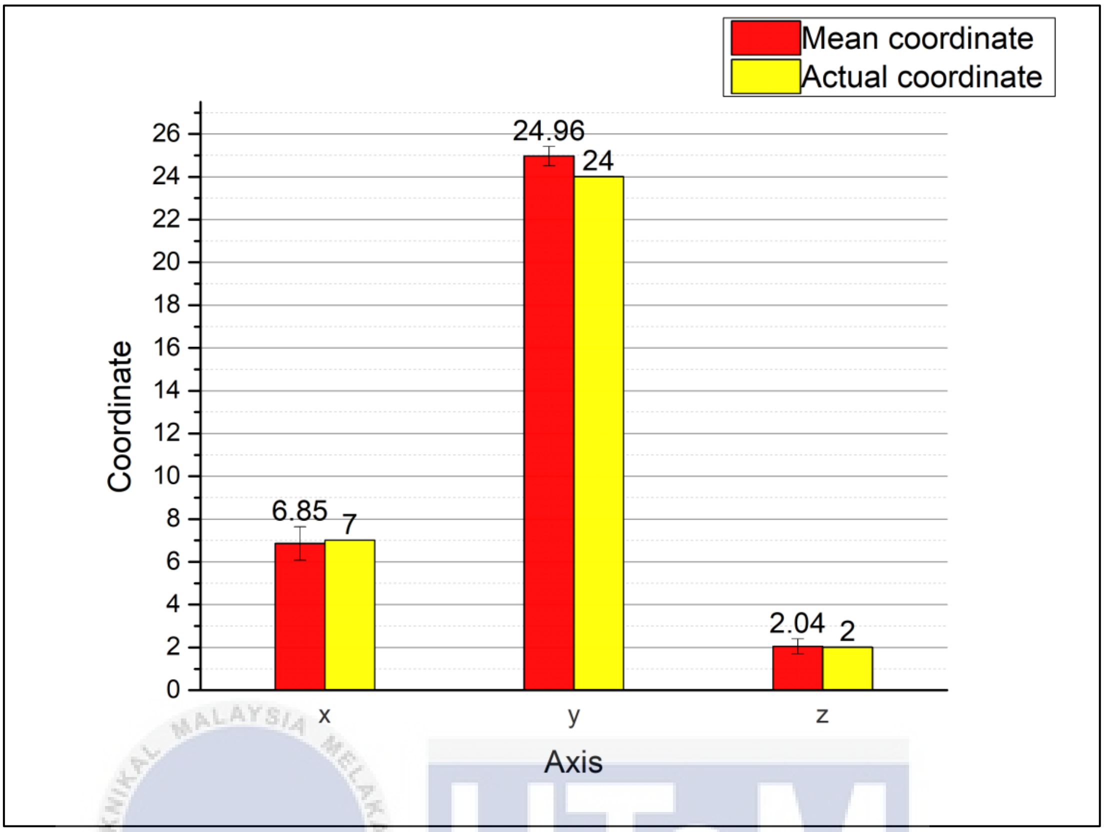


Figure 4.10 Comparison of actual coordinate with mean coordinate for coordinate (7,24) with object A

Based on the graph in Figure 4.11, the red bar indicates the mean coordinates and the yellow bar indicates the actual coordinate. The mean coordinate for axis x is higher than the actual coordinate by 0.01 cm. The mean coordinate for axis y is higher than the actual coordinate by 1.55 cm. The mean coordinate for axis z is higher than the actual coordinate by 0.1 cm. The average percentage error for axis x, y and z is 0.14%, 6.46% and 5% respectively. In conclusion, the coordinate measurement from the forward kinematic is accurate with a 3.87% percentage error.

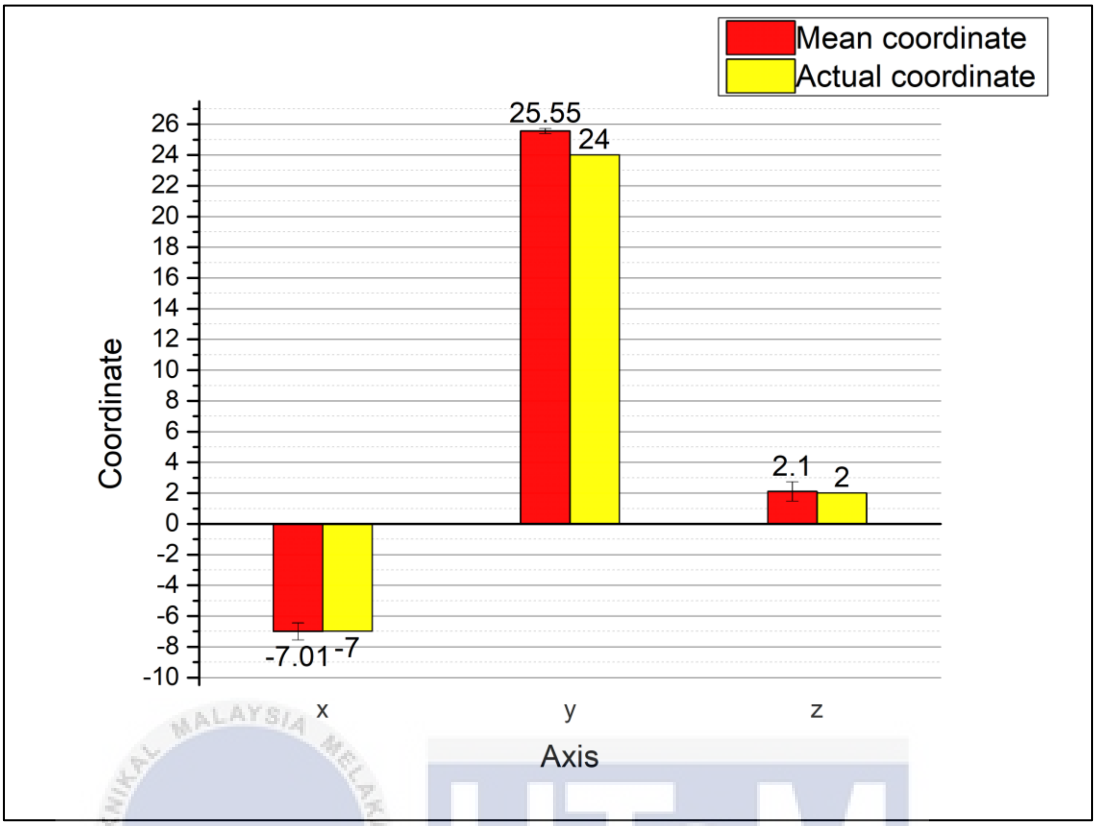


Figure 4.11 Comparison of actual coordinate with mean coordinate for coordinate (-7,24) with object A

Based on the graph in Figure 4.13, the red bar indicates the mean coordinates and the yellow bar indicates the actual coordinate. The mean coordinate for axis x is higher than the actual coordinate by 0.1 cm. The mean coordinate for axis y is higher than the actual coordinate by 1.3 cm. The mean coordinate for axis z is higher than the actual coordinate by 0.4 cm. The average percentage error for axis x, y and z is 1.42%, 5.41% and 6.67 respectively. In conclusion, the coordinate measurement from the forward kinematic is accurate with a 4.5% percentage error.

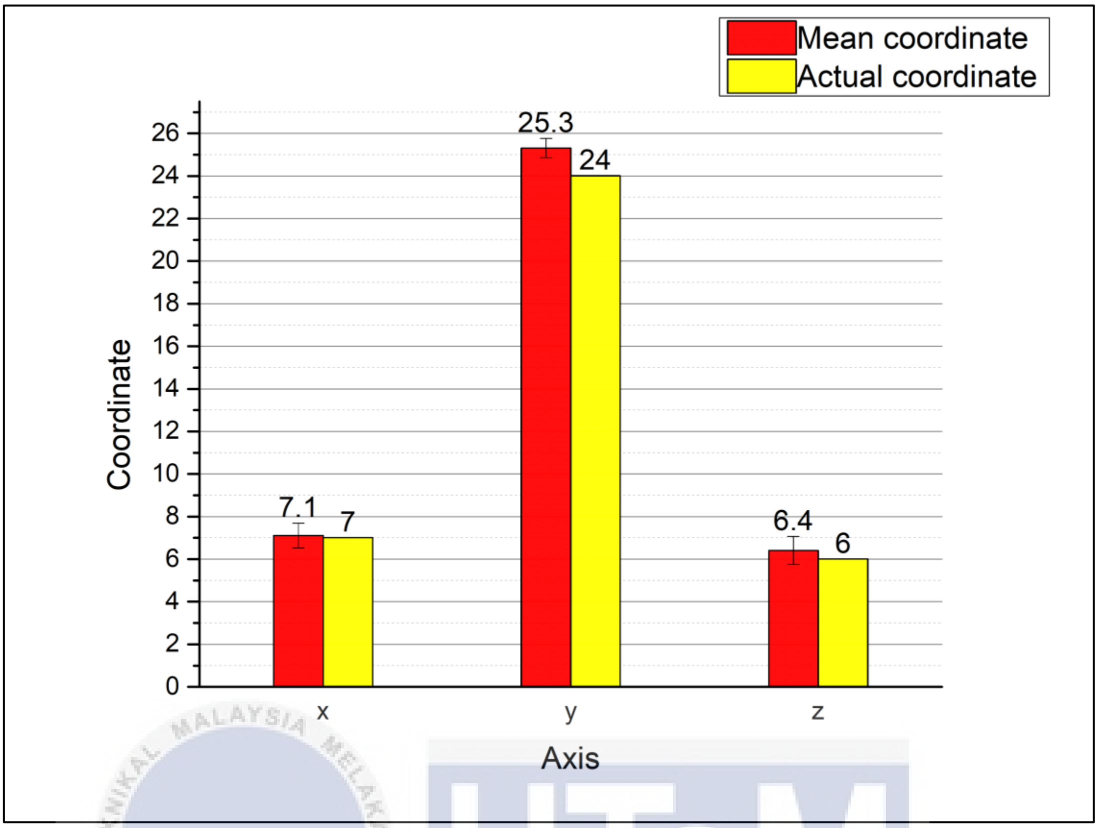


Figure 4.12 Comparison of actual coordinate with mean coordinate
for coordinate (7,24) with object B

Based on the graph in Figure 4.13, the red bar indicates the mean coordinates and the yellow bar indicates the actual coordinate. The mean coordinate for axis x is lower than the actual coordinate by 0.1 cm. The mean coordinate for axis y is higher than the actual coordinate by 1.4 cm. The mean coordinate for axis z is higher than the actual coordinate by 0.17 cm. The average percentage error for axis x, y and z is 1.42%, 5.83% and 2.83% respectively. In conclusion, the coordinate measurement from the forward kinematic is accurate with a 3.36% percentage error.

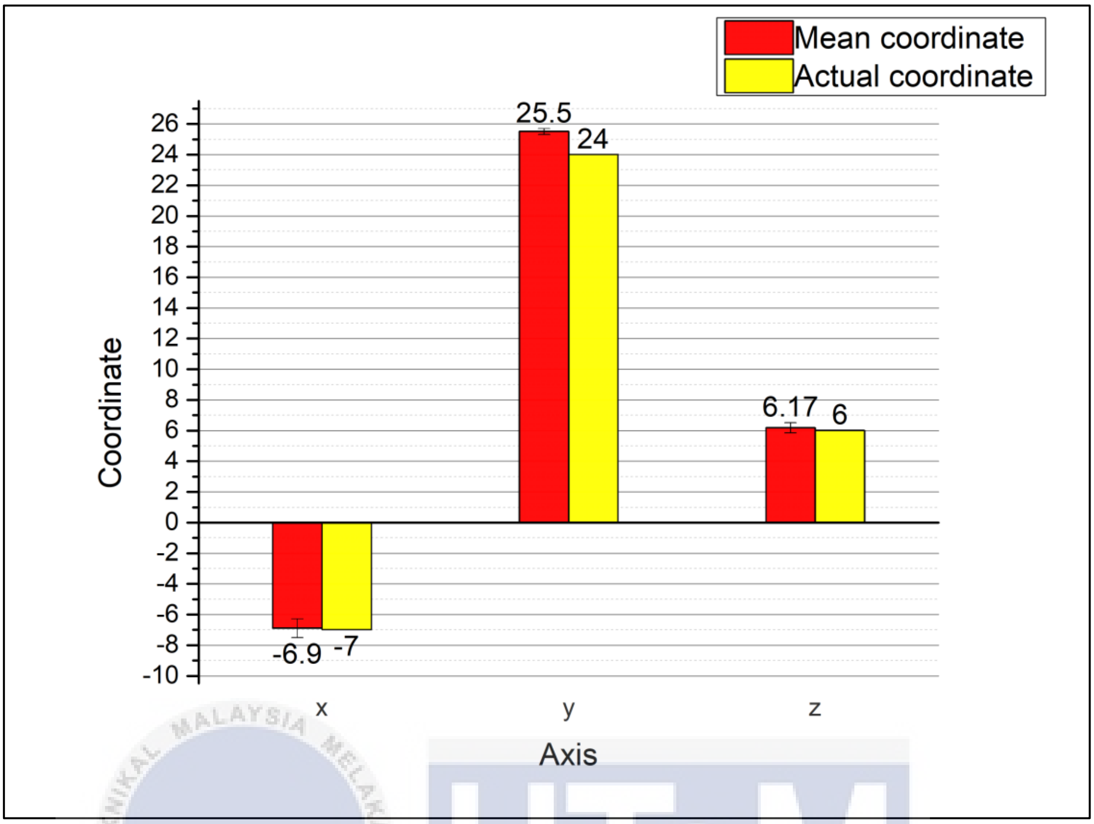


Figure 4.13 Comparison of actual coordinate with mean coordinate for coordinate (7,24) with object B

4.6 Experiment 6: Robotic Arm Performance Test

Based on the graph in Figure 4.14, Label A, B, C and D represents the coordinates of objects which are (7,28), (10,14), (-9,12) and (-10,20) respectively. The yellow bar represents the mean time of carrying the object via manual control while the red bar represents the mean time of carrying the object via preprogrammed control. The time taken to carry the object to the target area is higher when its manually controlled compare to when it uses a preprogrammed control. The total average time taken for manual controlled is 137.4s and 5.9s for the preprogrammed control. This is because the motor needs to be changed and selected to actuate different joints. In conclusion, the performance of the robotic arm is low.

Figure 4.14 Graph of location of object versus the time taken

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

As conclusion, all the objectives have been successfully achieved. The first objective is to design and develop a functional 4-DOF robotic arm that is controlled using EMG signals. The robotic arm is connected to the Myoware muscle sensor that sense EMG signals from the muscles. The manipulator can be controlled using hand gesture hand rest, ring finger flex, medium grasp and hard grasp to perform basic motion such as rotating joints and gripping objects. The joints can rotate to a specific angle using servo motors that is interfaced by the Arduino Uno R3. The second objective is to perform analysis on the operation of the robotic arm based on the characteristic of EMG signal collected from the forearm. The hand gesture chosen generates different range value of EMG signals and is used to control the robotic arm joints. In addition, the sampling rate and arm position affect the output of the signals. The ring finger flex gesture is used to rotate the joints clockwise while the medium grasp gesture is used to rotate the joints anticlockwise. The hard grasp gesture is used to select and unselect motors in. Undershoot and overshoot signals causes the joints to respond differently from the desired motion. The last objective is to test the effectiveness of the actuators based on the instruction given by the controller. The end effector of the robotic arm can reach to the target coordinate on the platform with minor percentage error. However, time taken to maneuver within the platform is higher compared to being automatically controlled. This is because the servo needs to be controlled independently.

5.2 Recommendations

As for recommendations, the single channel Myoware muscle sensor can be replaced with a module that can supports more than single channel. More muscle groups can be detected simultaneously with by having a multi-channel module. Method such as fuzzy logics and artificial neural network (ANN) can be applied to classify the gesture more accurately. Next, a wireless module can be implemented into the system to connect the robotic arm and controller. This is to allow the arm to move freely without being restricted by the length of the wire. The wireless system also helps reduce noise from the equipment that affects the signal.

REFERENCES

[1] I. Elamvazuthi, N. H. X. Duy, Z. Ali, S. W. Su, M. K. A. A. Khan, and S. Parasuraman, “Electromyography (EMG) based Classification of Neuromuscular Disorders using Multi-Layer Perceptron,” *Procedia Comput. Sci.*, vol. 76, no. Iris, pp. 223–228, 2015.

[2] F. Sadikoglu, C. Kavalcioglu, and B. Dagman, “Electromyogram (EMG) signal detection, classification of EMG signals and diagnosis of neuropathy muscle disease,” *Procedia Comput. Sci.*, vol. 120, pp. 422–429, 2017.

[3] N. Amrutha and V. H. Arul, “A Review on Noises in EMG Signal and its Removal,” vol. 7, no. 5, pp. 23–27, 2017.

[4] D. Leonardis *et al.*, “An EMG-controlled robotic hand exoskeleton for bilateral rehabilitation,” *IEEE Trans. Haptics*, vol. 8, no. 2, pp. 140–151, 2015.

[5] H. A. Yousif, N. A. Rahim, and A. Zakaria, “An Intelligent EMG Controlled Wheelchair – A Review,” no. July 2018, 2016.

[6] S. V Krivosheev and V. Borisova, “ScienceDirect of EMG-Controlled Prosthetic Prototyping of EMG-Controlled Prosthetic Prototyping of EMG-Controlled Prosthetic Prototyping of EMG-Controlled Prosthetic Hand with Sensory System Hand with Sensory System Hand with Sensory System Hand with Sensory System,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 16027–16031.

[7] G. J. Androwis, R. Pilkar, A. Ramanujam, and K. J. Nolan, “Electromyography Assessment During Gait in a Robotic Exoskeleton for Acute Stroke,” vol. 9, no. August, pp. 1–12, 2018.

[8] “Teleoperated Robotic Arm Movement Using Electromyography Signal With Wearable Myo Armband _ Elsevier Enhanced Reader.pdf.” .

[9] D. Roman-Liu and P. Bartuzi, “The influence of wrist posture on the time and frequency EMG signal measures of forearm muscles,” *Gait Posture*, vol. 37, no. 3, pp. 340–344, 2013.

[10] J. Lopes, M. Simão, N. Mendes, M. Safeea, J. Afonso, and P. Neto, “Hand/arm Gesture Segmentation by Motion Using IMU and EMG Sensing,” *Procedia Manuf.*, vol. 11, no. June, pp. 107–113, 2017.

[11] W. M. H. W. Kadir, R. E. Samin, and B. S. K. Ibrahim, “Internet controlled

robotic arm," *Procedia Eng.*, vol. 41, no. Iris, pp. 1065–1071, 2012.

[12] K. Gundogdu, S. Bayrakdar, and I. Yucedag, "Developing and modeling of voice control system for prosthetic robot arm in medical systems," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 30, no. 2, pp. 198–205, 2018.

[13] B. F. Erlingsson, I. Hreimsson, P. I. Pálsson, S. J. Hjálmarsson, and J. T. Foley, "Axiomatic Design of a Linear Motion Robotic Claw with Interchangeable Grippers," *Procedia CIRP*, vol. 53, pp. 213–218, 2016.

[14] F. Y. Ómarsdóttir, R. B. Ólafsson, and J. T. Foley, "The Axiomatic Design of Chessmate: A Chess-playing Robot," *Procedia CIRP*, vol. 53, pp. 231–236, 2016.

[15] N. Masud, C. Smith, and M. Isaksson, "Disturbance observer based dynamic load torque compensator for assistive exoskeletons," *Mechatronics*, vol. 54, no. September 2017, pp. 78–93, 2018.

[16] M. A. Fikri, S. C. Abdullah, and M. H. M. Ramli, "Arm exoskeleton for rehabilitation following stroke by learning algorithm prediction," *Procedia Comput. Sci.*, vol. 42, no. C, pp. 357–364, 2014.

[17] S. Ramasamy, R. Karthikesh, P. Manikandan, and P. Thiagarajan, "Development of stepper motor based two DOF robotic arm transferring liquid using peristaltic pump," *Int. J. Eng. Technol.*, vol. 5, no. 1, pp. 52–57, 2013.

[18] R. V Sharan and G. C. Onwubolu, "Simulating the Arm Movements of a Stepper Motor Controlled Pick- and-Place Robot Using the Stepper Motor Model," vol. 60, pp. 59–66, 2013.

[19] T. H. E. Basics and O. F. Electromyography, "THE BASICS OF ELECTROMYOGRAPHY," vol. 76, no. Suppl II, pp. 32–35, 2005.

[20] G. Ramantani, L. Maillard, and L. Koessler, "Correlation of invasive EEG and scalp EEG," *Seizure Eur. J. Epilepsy*, vol. 41, pp. 196–200, 2016.

[21] A. State *et al.*, "Needle Electromyography (EMG)," no. June, pp. 1–3, 2005.

[22] F. Sadikoglu, C. Kavalcioglu, and B. Dagman, "Electromyogram (EMG) signal detection, classification of EMG signals and diagnosis of neuropathy muscle disease," *Procedia Comput. Sci.*, vol. 120, pp. 422–429, 2017.

[23] R. H. Chowdhury *et al.*, "Surface Electromyography Signal Processing and Classification Techniques," pp. 12431–12466, 2013.

[24] L. S. Sudarsan and E. C. Sekaran, "Design and development of EMG

controlled prosthetics limb," *Procedia Eng.*, vol. 38, pp. 3547–3551, 2012.

[25] H. Ghapanchizadeh, S. A. Ahmad, A. J. Ishak, and M. S. Al-quraishi, "Review of surface electrode placement for recording electromyography signals .," 2017.

[26] A. Saikia *et al.*, "Combination of0EMG Features and Stability Index for Finger Movements Recognition," *Procedia Comput. Sci.*, vol. 133, pp. 92–98, 2018.

[27] S. N. Sidek, N. A. Jalaludin, and A. U. Shamsudin, "Surface electromyography (sEMG)-based thumb-tip angle and force estimation using Artificial Neural Network for prosthetic thumb," *Procedia Eng.*, vol. 41, no. Iris, pp. 650–656, 2012.

[28] S. Fani *et al.*, "Assessment of myoelectric controller performance and kinematic behavior of a novel soft synergy-inspired robotic hand for prosthetic applications," *Front. Neurorobot.*, vol. 10, 2016.

[29] R. Shah and A. B. Pandey, "ScienceDirect ScienceDirect ScienceDirect ScienceDirect Concept for Automated Sorting Robotic Arm Concept for Automated Sorting Robotic Arm Concept for Automated Sorting Robotic Arm Concept for Automated Sorting Robotic Arm Costing models for capacity opt," *Procedia Manuf.*, vol. 20, no. 2017, pp. 400–405, 2018.

[30] R. O. Thomas, "Remote Monitoring and Control of Robotic Arm with Visual Feedback using Raspberry Pi," *Int. J. Comput. Appl.*, vol. 92, no. 9, pp. 25–28, 2014.

[31] A. Swamardika, I. N. Budiastra, N. Setiawan, and N. I. Er, "Design of Mobile Robot with Robotic Arm Utilising Microcontroller and Wireless Communication," *Int. J. Eng. Technol.*, no. April, 2017.

APPENDICES

APPENDIX A MAIN SOURCE CODE FOR ROBOTIC ARM MOVEMENT CONTROL

```
#include <Servo.h>

Servo MotorA; //Declare Motor A to Motor E
Servo MotorB;
Servo MotorC;
Servo MotorD;
Servo MotorE;

int value;
int Motor = 0; //declare variable for motor selection (*initialize to 0
                 which indicates that no motor is selected)

int pos = 90; //declare motor position variable
int pos2 = 45;
int pos3 = 0;
int pos4 = 135;
int pos5 = 0;

int angle1; //declare secondary motor position variable
int angle2;
int angle3;
int angle4;
int angle5;

boolean pause = true; //declare variable pause for each funtion (void loop/
                          void operation/void print)
boolean pause2 = true;
boolean pause3 = true;

void operation(); //declare function operation
                    (*moves motor forward or reverse*)
void print (); //declare function (*select whhich motor to operate*)


void setup ()
{
  Serial.begin(9600); //initialize serial monitor

  MotorA.attach(9); //initialize motor A - motor E pin
  MotorB.attach(10);
  MotorC.attach(11);
  MotorD.attach(12);
  MotorE.attach(13);

  MotorA.write(pos); //initialize motor A - motor E inital position
  MotorB.write(pos2);
  MotorC.write(pos3);
  MotorD.write(pos4);
  MotorE.write(pos5);

}


void loop ()
{
```

```
  if (pause == true) //if function void loop is enabled
  {
    print (); //enter function print
  }

  else if (pause == false) //if function void loop is disabled
  {
    while(1){}   //pauses the function by entering an infinite while loop
  }

}



void print ()
{
  pause3 = true; //enables function operation ()

  if (pause2 == true) //if function print is enabled
  {
  int EMG1;
  pause = false; //pauses function void loop
  Serial.println("Flex ring finger to select Motor ");

  for(int i = 1; i < 1000; i++) //for loop to continuously read and
                                  print EMG signal
  {
    delay (15);
    EMG1 = analogRead(A0);
    Serial.print(i);
    Serial.print(".");
    Serial.print(EMG1);


    if (EMG1 > 65 && EMG1 < 140) //Range EMG value of ring finger flex
    {
      if (Motor < 5) //set variable Motor to value of 0 to 5
      {
      Motor += 1; //increase value of variable motor (*acts as a NEXT
                    button*)
      delay (15);
      }

      else
       {
        Motor = Motor - 5; //to reset the value of variable motor to 0
                              if the value is more than 5
        delay (15);
       }
       Serial.print(“||   Motor is: ");
       Serial.println(Motor);
       delay (20);
     }

     else if (EMG1 > 140)   //threshold value of hard grasping
     {
     delay (10);
     Serial.println("");
     operation (); //enter function void operation
     }

     else
     Serial.println("");
     delay (100);
   }
   Serial.println("");
```

```
 delay (1000);
 pause = true; //unpause function void loop ()
 return 0;
 }
 else if (pause2 == false)
 while (1) {}
}


void operation ()
{
 delay (500);
 if (pause3 == true)
 {
  pause2 =false; //pause function print ()
  int EMG2;
  delay (200);


  Switch(Motor) //choose which motor based on the value of variable
                  Motor
  {
   case 0:   delay (100);
             Serial.println(" "); //display angles of motor
             Serial.println("Angle of Motors");
             Serial.println("");
             Serial.print("Motor A: ");
             Serial.println(pos);
             Serial.print("Motor B: ");
             Serial.println(pos2);
             Serial.print("Motor C: ");
             Serial.println(pos3);
             Serial.print("Motor D: ");
             Serial.println(pos4);
             Serial.print("Motor E: ");
             Serial.println(pos5);



             Serial.println(" "); //display secondary angles of
       motor
             Serial.println("Angle of Motors");
             Serial.println("");
             Serial.print("Angle A: ");
             Serial.println(angle1);
             Serial.print("Angle B: ");
             Serial.println(angle2);
             Serial.print("Angle C: ");
             Serial.println(angle3);
             Serial.print("Angle D: ");
             Serial.println(angle4);
             Serial.print("Angle E: ");
             Serial.println(angle5);


             break;

   case 1:   delay (100);
             Serial.println("You have chosen motor 1");
             delay (100);

             for (int j = 1; j < 100000; j++) //for loop to continuously
         read and print EMG signal
           {
             EMG2 = analogRead(A0); //read EMG signals
             Serial.print(j);
```

```
Serial.print(".");
Serial.print(EMG2); //print EMG signals
delay (100);

if (EMG2 <= 55) //hand rest gesture range
{
  delay (15);
  Serial.println("..........Rest");
  delay (20);
}

else if(EMG2 > 55 && EMG2<=80) //ring finger flex gesture
                                 range

{


  if (pos < 180) //motor not more than 180 degrees
  {
  delay (15);
  pos += 1; //increase angle of rotation
  MotorA.write(pos);
  Serial.print("..........Forward");
  Serial.print(“[ ");
  Serial.print(angle1); //display angle on the serial
                          monitor
  Serial.println(“}");
  delay (20);



  }

  else if ( pos >= 180)  //motor more than 180 degrees
  {
    delay (15);
    Serial.println("..........Exceed Upper Limit");
    delay (20);
  }
}

else if(EMG2 > 80 && EMG2 < 140) //medium grasp gesture
                                   range
{
  if (pos > 0) //motor more than 0 degrees
  {
  delay (15);
  pos -= 1; //decrease angle of rotation
  MotorA.write(pos);
  Serial.print("..........Reverse");
  Serial.print(“[ ");
  Serial.print(pos); //display angle on the serial monitor
  Serial.println(“)");
  delay (20);
  }

  else if (pos <= 0) //motor less than 0 degrees
  {
    delay (30);
    Serial.println("..........Exceed Lower Limit");
    delay (20);
  }
}


else if (EMG2 > 140 )  //hard grasp gesutre threshold
{
```

```
                Serial.println("..........Return");
                pause = true;
                pause2 = true;
                pause3 = false;
                Serial.println("");
                delay (1000);
                loop (); //return to motor selection subroutine

              }

            }

            break;

    case 2:   delay (100);
              Serial.println("You have chosen motor 2");


              for (int k = 1; k < 100000; k++)
              {
              EMG2 = analogRead(A0); //read EMG signals
              Serial.print(k);
              Serial.print(".");
              Serial.print(EMG2); //print EMG signal on the serial monitor

              delay (15);

              if (EMG2 <= 55) //hand rest gesture range
              {
                delay (15);
                Serial.println("..........Rest");
                delay (20);
              }


              if(EMG2 > 55 && EMG2 <= 80) //ring finger flex gesture range

              {
                if (pos2 < 180) //motor less than 180 degrees
                {
                pos2 += 1; //increase angle of rotation
                angle2 = 180 - pos2; //substitute motor angle with angle
                                      at joint B axis
                MotorB.write(pos2);
                delay (15);
                Serial.print("..........Forward");
                Serial.print(“[ ");
                Serial.print(angle2); //display joint angle at serial
                                        monitor

                Serial.println(“}");
                delay (30);
                }

                else if (pos2 >= 180) //motor more than 180 degrees
                {
                  delay (15);
                  Serial.println("..........Exceed Lower Limit");
                  delay (20);
                }
              }

                else if(EMG2 > 80 && EMG2 < 140)   //medium grasp
                                                     gesture range
              {
                if (pos2 >= 0)   //motor more than 0 degrees
```

```
        {
        pos2 -= 1; //decrease angle of rotation
        angle2 = 180 - pos2; //substitute motor angle with angle
                          at joint B axis
        MotorB.write(pos2);
        delay (15);
        Serial.print(".........Reverse");
        Serial.print("[ ");
        Serial.print(angle2); //display joint angle at serial
                           monitor
        Serial.println("}");
        delay (20);
        }

        else if (pos2 < 0) //motor less than 0 degrees
        {
          delay (15);
          Serial.println(".........Exceed Upper Limit");
          delay (20);
        }
      }

       else if (EMG2 > 140) //hard grasp gesture range
      {
        Serial.println(".........Return");
        pause = true;
        pause2= true;
        pause3 = false;
        delay (1000);
        loop (); //return to motor selection subroutine

      }


    }
    break;


  case 3:   delay (100);
      Serial.println("You have chosen motor 3");
      for (int l = 1; l < 100000; l++)
    {
      EMG2 = analogRead(A0); //read EMG signal value
      Serial.print(l);
      Serial.print(".");
      Serial.print(EMG2); //print EMG signal value
      delay (10);

      if (EMG2 <= 55) //hand rest gesture range
      {
        delay (15);
        Serial.println(".........Rest");
        delay (15);
      }


      if(EMG2 > 55 && EMG2 <= 80) //ring finger flex gesture range

      {
        if (angle3 > -90) //joint angle more than -90 degrees
        {
        pos3 -= 1; //decrease angle of rotation
        angle3 = 90+pos3; //substitute motor angle with angle at
                         joint C axis
        MotorC.write(pos3);
        Serial.print(".........Forward");
```

```
                Serial.print(“[ ");
                Serial.print(angle3); //display joint angle on serial
                                        monitor
                Serial.println(“}");
                delay (15);
                }

                else if (angle3 <= -90) //motor less than -90 degrees
                {
                  delay (15);
                  Serial.println(".........Exceed Lower Limit");
                  delay (15);
                }
              }


              else if(EMG2 > 80 && EMG2 < 140) //medium grasp gesture
                                                 range
              {
                if (angle3 < 90)   //joint angle less than 90
                {
                pos3 += 1; //increase angle of rotation
                angle3 = 90+pos3; //substitute motor angle with angle at
                                    joint C axis
                MotorC.write(pos3);
                delay (15);
                Serial.print(".........Reverse");
                Serial.print(“[ ");
                Serial.print(angle3); //display joint angle on serial
                                        monitor
                Serial.println(“}");
                delay (15);
                }

                else if (angle3 >= 90) //motor more than -90 degrees
                {
                  delay (15);
                  Serial.println(".........Exceed Upper Limit");
                  delay (15);
                }
              }

              else if (EMG2 > 140) //hard grasp gesture range
              {
                Serial.println(".........Return");
                pause = true;
                pause2 = true;
                pause3 = false;
                Serial.println("");
                delay (1000);
                loop (); //return to motor selection subroutine

              }

            }

            break;


    case 4:   delay (100);
              Serial.println("You have choosen motor 4");

              for(int m = 1; m < 100000; m++)   //for loop to continuously
                                               read and print EMG signal
            {
              EMG2 = analogRead(A0); //read EMG signal value
```

```
Serial.print(m);
Serial.print(".");
Serial.print(EMG2); //print EMG signal on the serial monitor

delay (20);

 if (EMG2 <= 55) //hand rest gesture range
{
  delay (15);
  Serial.println("..........Rest");
  delay (20);
}


if(EMG2 > 55 && EMG2 <= 80) //ring finger flex gesture range

{
  if (angle4 > -90) //joint angle more than -90 degrees
  {
  pos4 += 1; //increase angle of rotation
  angle4 = 90 - pos4; //substitute motor angle with angle at
            Joint D axis
  MotorD.write(pos4);
  delay (15);
  Serial.print("..........Forward");
  Serial.print(“[ ");
  Serial.print(angle4); //display joint angle on serial
             monitor
  Serial.println(“)");
  delay (20);
  }

  else if (angle4 <= 90) //joint angle less than 90 degrees
  {
    delay (20);
    Serial.println("..........Exceed Lower Limit");
    delay (20);
  }
}

else if(EMG2 > 80 && EMG2 < 140) //medium grasp gesture
                  range
{
  if (angle4 < 90) //joint angle less than 90 degrees
  {
  pos4 -= 1; //decrease angle of rotation
  angle4 = 90 - pos4; //substitute motor angle with angle at
            joint D axis
  MotorD.write(pos4);
  delay (15);
  Serial.print("..........Reverse");
  Serial.print(“[ ");
  Serial.print(angle4); //display joint angle on serial
             monitor
  Serial.println(“}");
  delay (20);
  }

  else if (angle4 >= 90) //joint angle more than 90 degree
  {
    delay (20);
    Serial.println("..........Exceed Lower Limit");
    delay (20);
  }
}

```

```
         else if (EMG2 > 140) //hard grasp gesture range
         {
          Serial.println(".........Return");
          pause = true; //enables function void loop and void print
                          so that the program can continue
                          the funtion cycle
          pause2 = true;
          pause3 = false; //disables funtion void operation to
                           prevent the program from crashing
          Serial.println("");
          delay (1000);
          loop (); //return to motor selection subroutine

         }

        }

        break;


    case 5:   delay (100);
             Serial.println("You have chosen motor 5");

             for (int n = 1; n < 100000; n++) //for loop to continuously
                                                 read and print EMG signal
            {
             EMG2 = analogRead(A0); //read EMG signal value
             Serial.print(n);
             Serial.print(".");
             Serial.print(EMG2); //print EMG signal on the serial monitor
             delay (20);

             if (EMG2 <= 55) //hand rest gesture range
             {
              delay (15);
              Serial.println(".........Rest");
              delay (20);
             }

             if(EMG2 > 55 && EMG2 <= 80) //ring finger flex gesture range

             {
              if (pos5 < 20) //motor angle less than 20 degrees
              {
              pos5 += 1; //increase angle of rotation
              MotorE.write(pos5);
              Serial.print(".........Close");
              Serial.print(“[ ");
              Serial.print(pos5); //display joint angle on serial
                                   monitor
              Serial.println(“}");
              delay (30);
              }

              else if (pos5 >= 20) //motor angle more than 20 degrees
              {
               delay (20);
               Serial.println(".........Exceed Upper Limit");
               delay (20);
              }
             }

             else if(EMG2 > 80 && EMG2 < 140) //medium grasp gesture
                                                  range
             {
              if (pos5 > 0) //motor angle more than 0 degrees
```

```
               {
               pos5 -= 1; //decrease angle of rotation
               MotorE.write(pos5);
               Serial.print(".........Open");
               Serial.print(“[ ");
               Serial.print(pos5); //display joint angle on serial
                                      monitor
               Serial.println(“}");
               delay (30);
               }

               else if (pos5 <= 0) //motor angle less than 0 degrees
               {
                 delay (20);
                 Serial.println(".........Exceed Lower Limit");
                 delay (20);
               }
             }

             else if (EMG2 > 140)  //hard grasp gesture range
             {
               pause = true;
               pause2 = true;
               pause3 = false;
               Serial.println("");
               delay (1000);
               loop (); //return to motor selection subroutine

             }

           }

           break;



  }
 }
 else if (pause3 == false)
 while (1) {};
}
```

APPENDIX B SOURCE CODE FOR AUTOMATED CONTROL: LOCATION (-9,12)

```
#include <Servo.h>

Servo MotorA; //Declare Motor A to Motor E
Servo MotorB;
Servo MotorC;
Servo MotorD;
Servo MotorE;

int value;
int Motor = 0;   //declare variable for motor selection (*initialize to 0
                    which
                  indicates that no motor is selected)

int pos = 90;   //declare motor position variable (*all motors are set to
                  angle 0 degrees *)
int pos2 = 45;
int pos3 = 0;
int pos4 = 135;
int pos5 = 0;
void stop ();

void setup () {

  MotorA.attach(9);
  MotorB.attach(10);
  MotorC.attach(11);
  MotorD.attach(12);
  MotorE.attach(13);

  MotorA.write(pos);
  MotorB.write(pos2);
  MotorC.write(pos3);
  MotorD.write(pos4);
  MotorE.write(pos5);



}

void loop () {

delay (1000);
 MotorE.write(0);
 delay (50);
 MotorA.write(130);
 delay (50);
 MotorC.write(0);
 delay (50);
 for(;pos4<145;pos4++)
 {
 MotorD.write(pos4);
 delay (70);
 }
 delay (100);
 for(;pos2<101;pos2++)
 {
 MotorB.write(pos2);
 delay (20);
 }
 delay (200);
```

```
 MotorE.write(45);
 delay (500);
 MotorB.write(90);
 delay (50);
 MotorC.write(0);
 delay (400);
 MotorA.write(95);
 delay (400);
 MotorD.write(155);
 delay (1500);
 MotorE.write(0);
 delay (50);

 stop ();

}

void stop () {
 delay(10000);
  }
```

APPENDIX C SOURCE CODE FOR AUTOMATED CONTROL: LOCATION (-7,28)

```
#include <Servo.h>

Servo MotorA; //Declare Motor A to Motor E
Servo MotorB;
Servo MotorC;
Servo MotorD;
Servo MotorE;

int value;
int Motor = 0;    //declare variable for motor selection (*initialize to 0
                     which indicates that no motor is selected)

int pos = 90;    //declare motor position variable (*all motors are set to
                    angle 0 degrees *)
int pos2 = 45;
int pos3 = 0;
int pos4 = 135;
int pos5 = 0;
void stop ();

void setup () {

  MotorA.attach(9);
  MotorB.attach(10);
  MotorC.attach(11);
  MotorD.attach(12);
  MotorE.attach(13);

  MotorA.write(pos);
  MotorB.write(pos2);
  MotorC.write(pos3);
  MotorD.write(pos4);
  MotorE.write(pos5);

}

void loop () {


delay(1000);
 MotorE.write(0);
 delay(200);
 MotorA.write(84);
 delay(100);
 MotorC.write(90);
 delay(100);
 MotorD.write(125);
 delay(700);
 for (; pos4<169; pos4++)
 {
  MotorB.write(pos4);
  delay(50);
 }
 delay(50);
 MotorE.write(45);
 delay(1500);
 MotorB.write(90);
 delay(50);

 MotorC.write(0);
```

```
 delay (50);
 MotorA.write(100);
 delay (50);
 MotorD.write(145);
 delay (1000);
 MotorE.write(0);

 stop ();

}

void stop () {
 delay(10000);
  }
```

APPENDIX D SOURCE CODE FOR AUTOMATED CONTROL: LOCATION (10,14)

```
#include <Servo.h>

Servo MotorA; //Declare Motor A to Motor E
Servo MotorB;
Servo MotorC;
Servo MotorD;
Servo MotorE;

int value;
int Motor = 0;    //declare variable for motor selection (*initialize to 0
                     which indicates that no motor is selected)

int pos = 90;    //declare motor position variable (*all motors are set to
                  angle 0 degrees *)
int pos2 = 45;
int pos3 = 0;
int pos4 = 135;
int pos5 = 0;
void stop ();


void setup () {

  MotorA.attach(9);
  MotorB.attach(10);
  MotorC.attach(11);
  MotorD.attach(12);
  MotorE.attach(13);

  MotorA.write(pos);
  MotorB.write(pos2);
  MotorC.write(pos3);
  MotorD.write(pos4);
  MotorE.write(pos5);


}

void loop () {

delay (1000);
 MotorE.write(0);
 delay (200);
 MotorA.write(45);
 delay (200);
 MotorC.write(0);
 delay (200);
 MotorD.write(125);
 delay (300);
 MotorB.write(110);
 delay (1000);
 MotorE.write(45);
 delay (1000);
 MotorB.write(90);
 delay (200);

 MotorC.write(0);
 delay(200);
 MotorA.write(95);
 delay(200);
```

```
 MotorB.write(80);
 delay(200);
 MotorD.write(144);
 delay(200);
 MotorB.write(95);

 delay (1000);

 MotorE.write(0);

 stop();

}

void stop () {
 delay (5000);
  }
```

APPENDIX E SOURCE CODE FOR AUTOMATED CONTROL: LOCATION (-10,20)

```
#include <Servo.h>

Servo MotorA; //Declare Motor A to Motor E
Servo MotorB;
Servo MotorC;
Servo MotorD;
Servo MotorE;

int value;
int Motor = 0;   //declare variable for motor selection (*initialize to 0
                  which indicates that no motor is selected)

int pos = 90;   //declare motor position variable (*all motors are set to
                  angle 0 degrees *)
int pos2 = 45;
int pos3 = 0;
int pos4 = 135;
int pos5 = 0;
void stop ();

void setup () {

  MotorA.attach(9);
  MotorB.attach(10);
  MotorC.attach(11);
  MotorD.attach(12);
  MotorE.attach(13);

  MotorA.write(pos);
  MotorB.write(pos2);
  MotorC.write(pos3);
  MotorD.write(pos4);
  MotorE.write(pos5);

}

void loop () {

 delay (1000);
 MotorE.write(0);
 delay (50);
 MotorA.write(135);
 delay (50);
 MotorC.write(40);
 delay (50);
 MotorD.write(135);
 delay (50);
 MotorB.write(130);
 delay (1500);
 MotorE.write(45);
 delay (1500);
 MotorB.write(90);
 delay (50);

 MotorC.write(0);
 delay(50);
 MotorA.write(95);
 delay(50);
 MotorB.write(80);
 delay(50);
```

```
 MotorD.write(145);
 delay(1000);
 MotorE.write(0);


 stop();

}

void stop () {
 delay (10000);
  }
```

APPENDIX F EMG SIGNALS DISTRIBUTION FOR DIFFERENT HAND GESTURES (EXPERIMENT 2)

time (ms)	Hand Rest	Thumb Flex	Index Finger Flex	Middle Finger Flex	Ring Finger Flex	Little Finger Flex	Wrist Flexion	Wrist Extension	Grasping
20	50	42	41	46	78	38	45	60	129
40	38	47	40	48	71	47	48	52	117
60	40	44	44	49	80	43	43	54	117
80	57	48	47	42	86	43	43	55	103
100	47	44	4	46	88	47	41	54	108
120	38	40	43	45	74	46	44	54	99
140	40	41	43	43	95	51	46	58	102
160	39	39	43	45	80	44	43	55	109
180	40	41	45	43	78	42	42	57	93
200	45	41	43	44	89	46	42	59	108
220	45	41	43	45	90	46	44	53	98
240	46	43	46	47	72	43	43	57	116
260	46	45	42	47	70	45	42	58	113
280	54	41	44	46	84	48	41	54	119
300	39	42	41	45	80	43	41	53	123
320	48	41	43	44	80	44	41	53	98
340	42	39	45	52	89	46	42	52	100
360	42	41	42	46	73	46	40	52	126
380	41	41	47	47	92	46	41	56	91
400	39	41	43	46	83	42	40	52	105

APPENDIX G EMG SIGNALS DISTRIBUTION FOR CONDITION 1 (EXPERIMENT 3)

Sampling time (ms)`	Condition A			
	Hand rest	Ring finger flex	Medium rasp	Hard grasp
20	40	88	129	238
40	44	69	117	200
60	39	87	117	238
80	39	88	103	283
100	41	81	108	226
120	42	79	99	254
140	43	76	102	240
160	42	87	109	192
180	39	83	93	213
200	41	81	108	285
220	40	77	98	213
240	39	69	116	196
260	41	73	113	201
280	41	81	119	247
300	41	78	123	306
320	41	72	98	227
340	42	74	100	258
360	41	77	126	212
380	39	92	91	266
400	41	78	105	254
Total	876	1588	2303	4261
Mean	43.8	79.4	115.2	213.1
Standard deviation	1.40	6.53	10.95	32.05

APPENDIX H EMG SIGNALS DISTRIBUTION FOR CONDITION 2 (EXPERIMENT 3)

Sampling time (ms)`	Condition B			
	Hand rest	Ring finger flex	Medium rasp	Hard grasp
1000	50	74	111	234
2000	38	76	115	229
3000	40	74	125	166
4000	57	115	106	179
5000	47	94	107	179
6000	38	72	118	195
7000	40	74	109	154
8000	39	70	132	141
9000	40	69	105	281
10000	45	77	121	240
11000	45	70	108	214
12000	46	86	121	281
13000	46	82	115	182
14000	54	81	115	239
15000	39	66	105	205
16000	48	72	124	144
17000	42	89	109	243
18000	42	77	123	268
19000	41	81	128	257
20000	39	89	106	230
Total	845	1765	2756	4337
Mean	42.3	88.3	137.8	216.9
Standard deviation	5.39	11.25	8.42	44.00

APPENDIX I EMG SIGNALS DISTRIBUTION FOR CONDITION 3 (EXPERIMENT 3)

Sampling time (ms)`	Condition C			
	Hand rest	Ring finger flex	Medium rasp	Hard grasp
20	41	69	125	104
40	52	75	182	278
60	44	81	172	163
80	41	86	122	256
100	41	75	122	327
120	43	74	138	288
140	41	82	116	184
160	41	85	126	237
180	42	94	123	294
200	43	103	137	123
220	43	112	139	365
240	43	107	134	112
260	41	81	121	102
280	40	102	158	243
300	41	85	136	328
320	41	94	137	102
340	43	103	171	247
360	42	97	126	248
380	41	71	140	104
400	41	89	131	232
Total	816	1590	2174	4749
Mean	40.8	79.5	108.7	237.5
Standard deviation	2.53	12.72	18.67	86.31

APPENDIX J EMG SIGNALS DISTRIBUTION WITH ANGLE ROTATION (DEGREES) FOR MOTOR A (EXPERIMENT 4)

Anticlockwise		
EMG	Angle	Rotation
58	1	Forward
61	2	Forward
57	3	Forward
57	4	Forward
48	4	Rest
60	5	Forward
57	6	Forward
48	6	Rest
48	6	Rest
50	6	Rest
54	6	Rest
66	7	Forward
75	8	Forward
65	9	Forward
63	10	Forward
57	11	Forward
66	12	Forward
71	13	Forward
77	14	Forward
81	13	Reverse
75	14	Forward
73	15	Forward
72	16	Forward
68	17	Forward
75	18	Forward
61	19	Forward
73	20	Forward
71	21	Forward
66	22	Forward
72	23	Forward
86	22	Reverse
85	21	Reverse
78	22	Forward
78	23	Forward
70	24	Forward
68	25	Forward
66	26	Forward
80	27	Forward
74	28	Forward

Clockwise		
EMG	Angle	Rotation
83	89	Reverse
70	89	Rest
82	88	Reverse
96	87	Reverse
88	86	Reverse
83	85	Reverse
81	84	Reverse
83	83	Reverse
91	82	Reverse
94	81	Reverse
97	80	Reverse
102	79	Reverse
92	78	Reverse
102	77	Reverse
106	76	Reverse
94	75	Reverse
91	74	Reverse
92	73	Reverse
87	72	Reverse
82	71	Reverse
95	70	Reverse
93	69	Reverse
78	70	Forward
77	71	Reverse
90	70	Reverse
87	69	Reverse
91	68	Reverse
95	67	Reverse
93	66	Reverse
95	65	Reverse
91	64	Reverse
88	63	Reverse
88	62	Reverse
85	61	Reverse
93	60	Reverse
97	59	Reverse
91	58	Reverse
85	57	Reverse
88	56	Reverse

78	29	Forward
73	30	Forward
67	31	Forward
80	32	Forward
75	33	Forward
75	34	Forward
70	35	Forward
71	36	Forward
84	35	Reverse
71	36	Forward
73	37	Forward
82	36	Reverse
71	37	Forward
68	38	Forward
69	39	Forward
65	40	Forward
71	41	Forward
71	42	Forward
67	43	Forward
58	44	Forward
78	45	Forward
111	55	Reverse
104	54	Reverse
102	53	Reverse
96	52	Reverse
92	51	Reverse
107	50	Reverse
105	49	Reverse
106	48	Reverse
105	47	Reverse
100	46	Reverse
99	45	Reverse
69	46	Forward
63	45	Forward

APPENDIX K EMG SIGNALS DISTRIBUTION WITH ANGLE ROTATION (DEGREES) FOR MOTOR B (EXPERIMENT 4)

Clockwise		
EMG	Angle	Rotation
85	1	Reverse
104	2	Reverse
98	3	Reverse
90	4	Reverse
87	5	Reverse
88	6	Reverse
89	7	Reverse
98	8	Reverse
107	9	Reverse
98	10	Reverse
90	11	Reverse
96	12	Reverse
91	13	Reverse
90	14	Reverse
103	15	Reverse
97	16	Reverse
98	17	Reverse
96	18	Reverse
101	19	Reverse
93	20	Reverse
90	21	Reverse
101	22	Reverse
97	23	Reverse
108	24	Reverse
102	25	Reverse
103	26	Reverse
112	27	Reverse
99	28	Reverse
102	29	Reverse
94	30	Reverse
110	31	Reverse
97	32	Reverse
99	33	Reverse
101	34	Reverse
93	35	Reverse
95	36	Reverse
93	37	Reverse
113	38	Reverse

Antilockwise		
EMG	Angle	Rotation
59	89	Forward
60	88	Forward
59	87	Forward
74	86	Forward
75	85	Forward
66	84	Forward
73	83	Forward
68	82	Forward
69	81	Forward
73	80	Forward
74	79	Forward
70	78	Forward
61	77	Forward
68	76	Forward
65	75	Forward
70	74	Forward
63	73	Forward
59	72	Forward
64	71	Forward
64	70	Forward
63	69	Forward
61	68	Forward
59	67	Forward
68	66	Forward
64	65	Forward
57	64	Forward
55	64	Rest
58	63	Forward
60	62	Forward
61	61	Forward
56	60	Forward
58	59	Forward
51	59	Rest
55	59	Rest
54	59	Rest
56	58	Forward
64	57	Forward
79	56	Forward

106	39	Reverse
92	40	Reverse
95	41	Reverse
92	42	Reverse
101	43	Reverse
106	44	Reverse
112	45	Reverse

68	55	Forward
53	55	Rest
54	55	Rest
61	54	Forward
56	53	Forward
55	53	Forward
61	52	Forward
72	51	Forward
78	50	Forward
70	49	Forward
75	48	Forward
59	47	Forward
69	46	Forward
63	45	Forward

APPENDIX L EMG SIGNALS DISTRIBUTION WITH ANGLE ROTATION (DEGREES) FOR MOTOR C (EXPERIMENT 4)

Anticlockwise		
EMG	Angle	Rotation
55	89	Rest
64	88	Forward
57	87	Forward
54	87	Rest
50	87	Rest
56	86	Forward
59	85	Forward
56	84	Forward
64	83	Forward
62	82	Forward
61	81	Forward
61	80	Forward
61	79	Forward
58	78	Forward
57	77	Forward
62	76	Forward
59	75	Forward
55	75	Rest
50	75	Rest
50	75	Rest
60	74	Forward
58	73	Forward
60	72	Forward
58	71	Forward
57	70	Forward
56	69	Forward
55	69	Rest
59	68	Forward
58	67	Forward
59	66	Forward
57	65	Forward
55	65	Rest
56	64	Forward
52	64	Rest
65	63	Forward
59	62	Forward
56	61	Forward
59	60	Forward

Clockwise		
EMG	Angle	Rotation
99	1	Reverse
116	2	Reverse
115	3	Reverse
120	4	Reverse
123	5	Reverse
131	6	Reverse
122	7	Reverse
124	8	Reverse
113	9	Reverse
107	10	Reverse
99	11	Reverse
116	12	Reverse
110	13	Reverse
109	14	Reverse
130	15	Reverse
132	16	Reverse
126	17	Reverse
137	18	Reverse
129	19	Reverse
126	20	Reverse
115	21	Reverse
110	22	Reverse
107	23	Reverse
102	24	Reverse
96	25	Reverse
93	26	Reverse
110	27	Reverse
96	28	Reverse
95	29	Reverse
93	30	Reverse
115	31	Reverse
98	32	Reverse
113	33	Reverse
123	34	Reverse
130	35	Reverse
110	36	Reverse
98	37	Reverse
93	38	Reverse

55	60	Rest
54	60	Rest
56	59	Forward
69	58	Forward
67	57	Forward
62	56	Forward
68	55	Forward
62	54	Forward
56	53	Forward
64	52	Forward
65	51	Forward
54	51	Rest
57	50	Forward
57	49	Forward
58	48	Forward
63	47	Forward
64	46	Forward
58	45	Forward

97	39	Reverse
92	40	Reverse
89	41	Reverse
90	42	Reverse
97	43	Reverse
109	44	Reverse
91	45	Reverse

APPENDIX M EMG SIGNALS DISTRIBUTION WITH ANGLE ROTATION (DEGREES) FOR MOTOR D (EXPERIMENT 4)

Clockwise		
EMG	Angle	Rotation
95	1	Reverse
92	2	Reverse
93	3	Reverse
109	4	Reverse
99	5	Reverse
107	6	Reverse
98	7	Reverse
96	8	Reverse
94	9	Reverse
92	10	Reverse
97	11	Reverse
94	12	Reverse
105	13	Reverse
101	14	Reverse
88	15	Reverse
93	16	Reverse
93	17	Reverse
96	18	Reverse
105	19	Reverse
105	20	Reverse
106	21	Reverse
90	22	Reverse
85	23	Reverse
87	24	Reverse
87	25	Reverse
82	26	Reverse
90	27	Reverse
87	28	Reverse
98	29	Reverse
91	30	Reverse
91	31	Reverse
99	32	Reverse
90	33	Reverse
95	34	Reverse
101	35	Reverse
97	36	Reverse
90	37	Reverse
80	36	Forward
82	37	Reverse

Anticlockwise		
EMG	Angle	Rotation
52	89	Reverse
49	89	Reverse
55	89	Reverse
64	88	Forward
62	87	Forward
56	86	Forward
67	85	Forward
62	84	Forward
64	83	Forward
68	82	Forward
64	81	Forward
58	80	Forward
58	79	Forward
61	78	Forward
61	77	Forward
58	76	Forward
55	76	Reverse
55	76	Reverse
61	75	Forward
55	75	Reverse
70	74	Forward
59	73	Forward
54	73	Reverse
61	72	Forward
58	71	Forward
69	70	Forward
58	69	Forward
68	68	Forward
59	67	Forward
65	66	Forward
58	65	Forward
54	65	Reverse
55	65	Reverse
58	64	Forward
59	63	Forward
56	62	Forward
54	62	Reverse
58	61	Forward
60	60	Forward

78	36	Forward
84	37	Reverse
95	38	Reverse
90	39	Reverse
84	40	Reverse
89	41	Reverse
96	42	Reverse
98	43	Reverse
107	44	Reverse
98	45	Reverse

60	59	Forward
54	59	Reverse
51	59	Reverse
54	59	Reverse
53	59	Reverse
57	58	Forward
54	58	Reverse
57	57	Forward
56	56	Forward
66	55	Forward
71	54	Forward
63	53	Forward
63	52	Forward
68	51	Forward
65	50	Forward
59	49	Forward
55	49	Reverse
58	48	Forward
57	47	Forward
57	46	Forward
57	45	Forward

APPENDIX N EMG SIGNALS DISTRIBUTION WITH ANGLE ROTATION (DEGREES) FOR MOTOR E (EXPERIMENT 4)

Anticlockwise		
EMG	Angle	Rotation
71	1	Close
54	1	Rest
53	1	Rest
54	1	Rest
55	1	Rest
49	1	Rest
50	1	Rest
54	1	Rest
53	1	Rest
60	2	Close
62	3	Close
56	4	Close
58	5	Close
72	6	Close
73	7	Close
64	8	Close
65	9	Close
68	10	Close
57	11	Close
63	12	Close
60	13	Close
67	14	Close
68	15	Close
67	16	Close
60	17	Close
63	18	Close
58	19	Close
69	20	Close
58	21	Close
57	22	Close
62	23	Close
61	24	Close
63	25	Close
63	26	Close
62	27	Close
62	28	Close
63	29	Close
64	30	Close

Clockwise		
EMG	Angle	Rotation
112	29	Open
122	28	Open
118	27	Open
116	26	Open
122	25	Open
121	24	Open
132	23	Open
119	22	Open
105	21	Open
100	20	Open
101	19	Open
101	18	Open
92	17	Open
86	16	Open
91	15	Open
90	14	Open
84	13	Open
96	12	Open
92	11	Open
81	10	Open
82	9	Open
88	8	Open
81	7	Open
86	6	Open
86	5	Open
101	4	Open
89	3	Open
88	2	Open
92	1	Open
81	0	Open

APPENDIX O COORDINATE OF END EFFECTOR AT OBJECT A BASED ON THE ANGLE OF JOINTS (EXPERIMENT 5)

Output angle of joints	Coordinate A			Coordinate B		
	Attempt			Attempt		
	1	2	3	1	2	3
θ_1	76	73	75	106	104	106
θ_2	41	36	82	26	35	21
θ_3	-49	-38	-27	-20	-37	0
θ_4	-41	-45	-54	-55	-45	-71
Attempt	x	y	Z	x	y	z
1	6.10	24.47	1.64	-7.38	25.72	1.52
2	7.66	25.07	2.15	-6.37	25.53	2.00
3	6.79	25.35	2.32	-7.28	25.39	2.77
Average	6.85	24.96	2.04	-7.01	25.55	2.1
Standard deviation	0.78	0.45	0.35	0.56	0.17	0.63

APPENDIX P COORDINATE OF END EFFECTOR AT OBJECT B BASED ON THE ANGLE OF JOINTS (EXPERIMENT 5)

Output angle of joints	Coordinate A			Coordinate B		
	Attempt			Attempt		
	1	2	3	1	2	3
θ_1	75	73	75	104	107	105
θ_2	43	41	51	47	48	46
θ_3	-37	-27	-44	-40	-43	-41
θ_4	-45	-54	-45	-45	-45	-44
Attempt	x	y	Z	x	y	z
1	6.90	25.71	5.71	-6.35	25.45	6.55
2	7.75	25.34	6.47	-7.55	25.71	6.01
3	6.65	24.81	7.03	-6.79	25.35	5.94
Average	7.1	25.30	6.40	-6.90	25.50	6.17
Standard deviation	0.58	0.45	0.66	0.61	0.19	0.33

APPENDIX Q TIME TAKEN TO TRANSPORT OBJECT C VIA MANUAL CONTROL (EXPERIMENT 6)

Coordinate	Attempts (s)			Average (s)
	1	2	3	
A (7,28)	130.8	138.0	133.2	133.8
B (10,14)	144.0	142.8	139.8	142.2
C (-9,12)	123.0	127.2	124.8	134.8
D (-10,20)	141.0	138.0	137.3	138.6

APPENDIX R TIME TAKEN TO TRANSPORT OBJECT C VIA AUTOMATED CONTROL (EXPERIMENT 6)

Coordinate	Attempts (s)			Average (s)
	1	2	3	
A (7,28)	6.3	5.5	5.8	5.7
B (10,14)	5.2	6.1	6.3	5.9
C (-9,12)	6.5	6.2	5.9	6.2
D (-10,20)	5.6	5.8	5.9	5.8