

**DESIGN OF A PRECISION MOTION CONTROL FOR AN UPPER LIMB OF  
ROBOTIC ARM**

**TEO TING HUAN**



**A report submitted in partial fulfillment of the requirements for the Bachelor Degree**

**of Mechatronics Engineering**  
اونيورسي تيكنيكل ماليسيا ملاك

**UNIVERSITI TEKNIKAL MALAYSIA MELAKA**

**Faculty of Electrical Engineering  
Universiti Teknikal Malaysia Melaka**

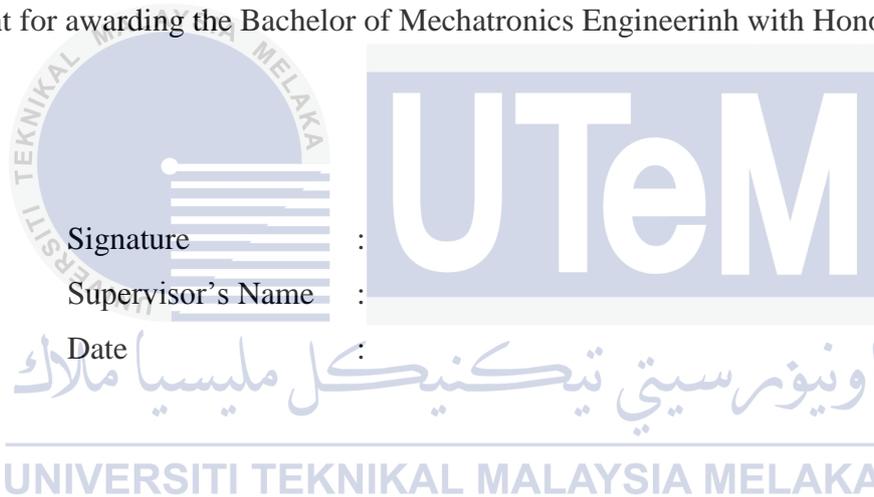
**June 2014**

I declare that this report entitle “Design of a Precision Motion Control for An Upper Limb of Robotic Arm” is the result of my own research except as cited in the references. The report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

اونيورسيتي تيكنيكل مليسيا ملاك

Signature :  
Author's Name :  
Date :

I hereby declare I have read through this report entitle “Design of A Precision Motion Control for An Upper Limb of Robotic Arm” and found that it has comply the partial fulfillment for awarding the Bachelor of Mechatronics Engineerinh with Honours.



## ACKNOWLEDGEMENT

I would like to express my greatest gratitude to the people who have helped and supported me throughout this project. I am grateful to my supervisor, **Dr. Mariam Md Ghazaly**, for her continuous support for the project, from initial advice and contacts in the early stages and through ongoing advice and encouragement all the way. Her willingness to give her time so generously has been very much appreciated.



## ABSTRACT

The objective of this project is to design a controller which is able to control the output angle for an upper limb of robotic arm. A model of robotic arm of 1 degree of freedom is designed and fabricated. Comparison between few types of motor is carried out and DC geared motor is chosen as the motor to be used in this project. Study is carried out based on previous research to investigate which type of controller is suitable to be used in this project. PID controller and fuzzy logic controller are chosen and compared in terms of their performance such as the steady-state error, settling time, rise time and overshoot. The experimental setup is carried out. The equipments used are Micro-Box 2000/2000C, Cytron DC geared motor, motor driver circuit. Micro-Box module acts as the interface between hardware component and programming software installed in a computer. The software involved in this project is MATLAB R2009a. Open loop simulations are first carried out to obtain the transfer function of motor. It is then substituted into the system for further analysis. Simulation for an uncompensated system is carried out to observe the closed loop system characteristic without using the controllers. After that, closed loop simulations are carried out for compensated systems using PID controller and fuzzy logic controller. Two types of experiments are carried out, namely point to point trajectory control and tracking control experiments. Analysis is made based on the results obtained and the best type of controller is chosen for achieving precise motion control for an upper limb of robot manipulator.

## ABSTRAK

Objektif projek ini adalah untuk mereka bentuk pengawal yang sesuai yang mampu mengawal tork keluaran dan kelajuan untuk bahagian lengan atas robot. Model lengan robot 1 darjah kebebasan direka dan dibina. Perbandingan antara beberapa jenis motor dijalankan dan DC motor dipilih sebagai motor yang akan digunakan dalam projek ini. Kajian dibuat berdasarkan penyelidikan untuk menyiasat jenis pengawal yang akan digunakan dalam projek ini. Pengawal PID dan pengawal logik kabur (fuzzy logic) dipilih dan dibandingkan dari segi prestasi mereka seperti keadaan mantap dan masa pengenapan. Eksperimen dijalankan. Peralatan yang digunakan adalah Micro-Box 2000/2000C, Cytron DC motor menjurus, litar pemacu motor. Modul Mikro-Box bertindak antara komponen perkakasan dan perisian pengaturcaraan dipasang dalam komputer. Perisian yang terlibat dalam projek ini adalah Matlab R2009a. Ujian gelung terbuka pertama yang dijalankan untuk mengawal kelajuan dan kawalan tork dalam sistem uncompensated. Selepas ujian gelung terbuka selesai, ujian gelung ditutup akan dilaksanakan untuk sistem pampasan menggunakan pengawal PID dan pengawal logik kabur. Perbandingan akan dibuat dan jenis pengawal yang sesuai akan dipilih untuk mencapai kawalan gerakan tepat untuk bahagian lengan atas robot.

## TABLE OF CONTENTS

CHAPTER	TOPIC	PAGE
	PROJECT TITLE	i
	SUPERVISOR DECLARATION	ii
	DECLARATION	iii
	ACKNOWLEDGEMENT	iv
	ABSTRACT	v
	ABSTRAK	vi
	TABLE OF CONTENTS	vii
	LIST OF FIGURES	ix
	LIST OF TABLES	xi
	LIST OF APPENDICES	xii
1	<b>INTRODUCTION</b>	
	1.1. Motivation	1
	1.2. Problem Statement	1
	1.3. Objectives	2
	1.4. Scope	2
	1.5. Chapter Overview	3

<b>2</b>	<b>LITERATURE REVIEW</b>	<b>4</b>
	2.1. Introduction	4
	2.2. Robotics	4
	2.3. Robotic Arm	5
	2.4. Motor	6
	2.5. Case Study of Controller	8
	2.5. Summary	16
<b>3</b>	<b>METHODOLOGY</b>	<b>18</b>
	3.1. Introduction	18
	3.2. Research Methodology	18
	3.2.1 Project Methodology	18
	3.2.2 Experiment Methodology	19
	3.3. Structure of Robotic Arm	19
	3.4. Hardware Implementation	20
	3.4.1 Cytron 12V DC Geared Motor with Hall Effect Encoder	21
	3.4.2 Micro-Box 2000/2000C (xPC Target Machine)	22
	3.4.3 Motor Driver Circuit	22
	3.5. System Overview	24
	3.6. Mathematical Modeling	26
	3.6.1 Modeling of a DC Geared Motor	26
	3.6.2 Torque	30
	3.6.3 Moment of Inertia	31
	3.6.4 Calculation of Moment of Inertia	32
	3.7. Open Loop Control	33
	3.8. Controller Design	34
	3.8.1 PID Controller	34
	3.8.2 Tuning Methods	36
	3.8.2.1 Trial and Error Method	36
	3.8.2.2 Ziegler-Nichols Method	37
	3.8.3 Fuzzy Logic Controller (FLC)	38
	3.8.3.1 Fuzzification Interface	39
	3.8.3.2 Knowledge Base/ Rule-Based System	41

	3.8.3.3 Inference Engine	41
	3.8.3.4 Defuzzification Interface	41
	3.9. Performance Characteristic (Time Domain)	42
	3.10. Output Angle Measurement	44
	3.11. Uncertainty and Reliability	45
<b>4</b>	<b>RESULTS AND DISCUSSION</b>	<b>46</b>
	4.1. Introduction	46
	4.2. Open Loop Simulation	47
	4.2.1 Transfer Function of Motor	54
	4.2.2 Linearity of the System	54
	4.3. Uncompensated System	55
	4.3.1 Point to Point Trajectory Control for Uncompensated System	55
	4.3.2 Tracking Control for Uncompensated System	58
	4.4. Compensated System with PID Controller	61
	4.4.1 Point to Point Trajectory Control with PID Controller	61
	4.4.2 Tracking Control with PID Controller	78
	4.4.3 Summary for PID Controller	88
	4.5. Compensated System with Fuzzy Logic Controller	89
	4.5.1 Point to Point Trajectory Control with Fuzzy Logic Controller	89
	4.5.2 Tracking Control with Fuzzy Logic Controller	93
	4.5.3 Summary for Fuzzy Logic Controller	95
<b>5</b>	<b>CONCLUSION AND RECOMMENDATION</b>	<b>96</b>
	5.1. Conclusion	96
	5.2. Recommendations	97

**REFERENCES****98****APPENDICES****100**

## LIST OF TABLES

NO.	TITLE	PAGE
2.1	Classification and characteristics of robotic arms	5
2.2	Comparison between the functions of motors	7
2.3	Summary of research papers	17
3.1	Specifications of Cytron 12V DC geared motor	21
3.2	Specifications of Micro-Box 2000/2000C	22
3.3	Parameters in the armature circuit of DC geared motor	27
3.4	Parameters relative to the torque of DC geared motor	30
3.5	Parameters relative to the moment of inertia of robotic arm	31
3.6	Mass of components of robotic arm	32
3.7	Parameters of the open loop simulation	34
3.8	Parameters related to PID controller	35
3.9	Parameters of transient response and the effects caused by manipulating $K_p$ , $K_i$ and $K_d$ values	37
3.10	Controller parameters of Ziegler-Nichols step response method	37
3.11	Components of fuzzy logic controller	38
3.12	Rule base system for fuzzy logic controller	41

3.13	Parameters associated with step response of the system	42
4.1	Results of system identification for DC motor speed model (1V)	49
4.2	Results of system identification for DC motor speed model (2V)	50
4.3	Results of system identification for DC motor speed model (3V)	51
4.4	Results of system identification for DC motor speed model (4V)	52
4.5	Results of system identification for DC motor speed model (5V)	53
4.6	Data obtained from open loop simulation	54
4.7	Parameters for point to point trajectory control experiments	55
4.8	Parameters for tracking error experiments	58
4.9	Parameters for point to point experiments using PID controller	61
4.10	Parameters of PID controller obtained from simulation results	75
4.11	Parameters for tracking error experiments for PID-controlled system	78
4.12	Summary of PID-controlled system	88
4.13	Parameters for point to point experiments using fuzzy logic controller	89
4.14	Parameters for tracking error experiments for fuzzy-controlled system	93
4.15	Summary of Fuzzy Logic controlled system	95
5.1	Comparison between the performance of PID controller and fuzzy logic controller	97
6.1	Name and functions of motor pins	33

## LIST OF FIGURES

NO.	TITLE	PAGE
2.1	Robotic mechanism	5
2.2	Illustration of a 1-DOF upper limb of robotic arm	6
2.3	Types of control methods for robotic arm	8
2.4	Different arm movements	9
2.5	Comparison of three controllers for RMS error. Bar charts represent mean values for twenty two motions and error bars represent maximum and minimum values	10
2.6	Comparison of three controllers for correlation factors	10
2.7	Comparison of three controllers for mean absolute error	11
2.8	Simulation of a third-order system (demonstrates robustness)	12
2.9	Tracking an ellipse	14
2.10	Experimental results in the upper displacement zone	15
3.1	Design of robotic arm (a) Top view (b) Bottom view (c) Side view (d) Side view	20
3.2	DC geared motor with encoder and its removable cover	21
3.3	Components of Micro-Box module	23

3.4	Motor driver circuit	23
3.5	System concept	24
3.6	Experimental setup of the project	25
3.7	Schematic diagram of the DC geared motor	26
3.8	Block diagram of the DC geared motor	29
3.9	Block diagram of open loop system	33
3.10	Block diagram of a typical PID controller	35
3.11	Block diagram of closed-loop PID control method	36
3.12	Basic structure of fuzzy logic systems	38
3.13	Fuzzy logic control system	39
3.14	The input and output variables of fuzzy logic controller	40
3.15	The surface view of the fuzzy logic controller	40
3.16	The output membership functions	42
3.17	Second order underdamped response specifications	43
3.18	The experimental set up of robotic arm	44
4.1	Structure of Chapter 4	46
4.2	Graph of input voltage (1V) and output angle against time	49
4.3	Graph of input voltage (2V) and output angle against time	50
4.4	Graph of input voltage (3V) and output angle against time	51
4.5	Graph of input voltage (4V) and output angle against time	52
4.6	Graph of input voltage (5V) and output angle against time	53
4.7	Graph of output angles against input voltages	54
4.8	Results of point to point trajectory control for an uncompensated	56

	system with input angle of $15^\circ$	
4.9	Results of point to point trajectory control for an uncompensated system with input angle of $30^\circ$	57
4.10	Results of tracking error experiment for an uncompensated system with input angle of $15^\circ$	59
4.11	Graph of steady-state error against time for an uncompensated system with input angle of $30^\circ$	60
4.12	Results of point to point trajectory control experiment for a PID control system with input angle of $15^\circ$ and $K_p$ value of 1	63
4.13	Results of point to point trajectory control experiment for a PID control system with input angle of $15^\circ$ and $K_p$ value of 10	64
4.14	Results of point to point trajectory control experiment for a PID control system with input angle of $15^\circ$ and $K_p$ value of 14	65
4.15	Results of point to point trajectory control experiment for a PID control system with input angle of $15^\circ$ and $K_p$ value of 14.6	66
4.16	Results of point to point trajectory control experiment for a PID control system with input angle of $30^\circ$ and $K_p$ value of 1	67
4.17	Results of point to point trajectory control experiment for a PID control system with input angle of $30^\circ$ and $K_p$ value of 10	68
4.18	Results of point to point trajectory control experiment for a PID control system with input angle of $30^\circ$ and $K_p$ value of 14	69
4.19	Results of point to point trajectory control experiment for a PID	70

	control system with input angle of 30 °and Kp value of 14.6	
4.20	Results of point to point trajectory control experiment for a PID control system with input angle of 60 °and Kp value of 1	71
4.21	Results of point to point trajectory control experiment for a PID control system with input angle of 60 °and Kp value of 10	72
4.22	Results of point to point trajectory control experiment for a PID control system with input angle of 60 °and Kp value of 14	73
4.23	Results of point to point trajectory control experiment for a PID control system with input angle of 60 °and Kp value of 14.6	74
4.24	Results of point to point trajectory control experiment for a PID control system with input angle of 15 °and Kp value of 8.76	75
4.25	Results of point to point trajectory control experiment for a PID control system with input angle of 15 °and Kp value of 8.76	76
4.26	— Results of tracking error experiment for a PID control system with input angle of 15 °	79
4.27	Results of tracking error experiment for a PID control system with input angle of 30 °	80
4.28	Results of tracking error experiment for a PID control system with input angle of 60 °	81
4.29	Comparison between output angles for a PID control system with different frequencies and input angle of 15 °(Top to bottom: 0.1Hz, 0.5Hz, 1Hz, 5Hz, 10Hz, 10Hz)	82

- 4.30 Comparison between output angles for a PID control system with 83  
different frequencies and input angle of  $30^\circ$  (Top to bottom: 0.1Hz,  
0.5Hz, 1Hz, 5Hz, 10Hz, 10Hz)
- 4.31 Comparison between output angles for a PID control system with 84  
different frequencies and input angle of  $30^\circ$  (Top to bottom: 0.1Hz,  
0.5Hz, 1Hz, 5Hz, 10Hz, 10Hz)
- 4.32 Comparison between steady-state error for a PID control system 85  
with different frequencies and input angle of  $15^\circ$  (Top to bottom: 0.1Hz,  
0.5Hz, 1Hz, 5Hz, 10Hz, 10Hz)
- 4.33 Comparison between steady-state error for a PID control system 86  
with different frequencies and input angle of  $30^\circ$  (Top to bottom: 0.1Hz,  
0.5Hz, 1Hz, 5Hz, 10Hz, 10Hz)
- 4.34 Comparison between steady-state error for a PID control system 87  
with different frequencies and input angle of  $60^\circ$  (Top to bottom: 0.1Hz,  
0.5Hz, 1Hz, 5Hz, 10Hz, 10Hz)
- 4.35 Graph of input voltage, output angle and steady-state error 90  
against time for a compensated system with input angle of  $15^\circ$
- 4.36 Graph of input voltage, output angle and steady-state error 91  
against time for a compensated system with input angle of  $30^\circ$
- 4.37 Graph of input voltage, output angle and steady-state error 92  
against time for a compensated system with input angle of  $60^\circ$
- 4.38 Graph of steady-state error against time for a fuzzy logic control 94  
system with an input angle of  $15^\circ$  (Top to bottom: 1Hz, 3Hz, 5Hz)

4.39	Graph of steady-state error against time for a fuzzy logic control system with an input angle of $30^\circ$ (Top to bottom: 1Hz, 3Hz, 5Hz)	94
4.40	Graph of steady-state error against time for a fuzzy logic control system with an input angle of $60^\circ$ (Top to bottom: 1Hz, 3Hz, 5Hz)	95
6.1	Project research methodology flow chart	101
6.2	Project experiment methodology flow chart	102
6.3	Rear view of the motor with encoder and cover	103
6.4	Block diagram of driver circuit	104
6.5	Open loop Simulink block diagram	105
6.6	Open loop Simulink block diagram with transfer function	105



## LIST OF APPENDICES

NO.	TITLE	PAGE
A	Final Year Project Research Gantt Chart	100
B	Project Research Methodology Flow Chart	101
C	Project Experiment Methodology Flow Chart	102
D	Connections of Cytron DC Geared Motor	103
E	Block Diagram of Driver Circuit Board	104
F	Procedure of Open Loop Control System	105
G	Fuzzy Rule-Based System	106

## CHAPTER 1

### INTRODUCTION

#### 1.1 Motivation

For accurate servo-positioning of mechanical actuators in realistic engineering systems, high precision motion is required to achieve both high speed and high torque. Once an adequate control loop is designed, the system basically has the ability to achieve the required precise positioning as the errors between the reference and the controlled variables because of fluctuations or disturbances can be discovered and minimized correspondingly [1].

The motivation for this project is to improve the motion for an upper limb of robotic arm using positioning control and analyze the performance of the controllers in terms of steady-state error, settling time and rise time. With improved motion control, the robot manipulator can be applied in various fields such as medical fields and semiconductor industry, in which a precise motion using robot manipulator is more preferred rather than human labor.

#### 1.2 Problem Statement

Robotic arm requires precise motion controls which enable it to determine the exact trajectory and the torque needed to achieve a targeted outcome. Currently in robotic assembly cell for small production, there are still some limitations for robots arms. For example, they cannot work efficiently in complicated environments without knowing any environmental information. They often rely on an external sensor system to help with the assembly work [2]. Also, improper motion control results in injuries or fatality. Thus, it is important to improve the capability of robotic manipulator.

With improved motion control, the robotic arm can be used in wider range applications and with increase efficiency. For example, it can be used in semiconductor industry in which a precise motion is required. The problem faced is in deciding and designing an appropriate controller to control the output angle of the upper limb of robotic arm correspondingly.



## 1.1 Objective

The main objectives of this project are:

1. To design a suitable controller to control the output angle for an upper limb of robotic arm.
2. To run the point to point trajectory control experiment and tracking control experiment.
3. To analyze and compare the performance of the controllers in terms of steady-state error, settling time and rise time.
4. To decide the best type of controller to be used.

## 1.2 Scope

The scopes covered in this project are

1. Design a controller to control the motion of robotic arm using MATLAB version 2009a.
2. Run the open-loop simulation and obtain the transfer function of motor.
3. Run the closed-loop simulation with uncompensated system.
4. Run the closed-loop simulation with compensated system for PID controller and fuzzy logic controller.
5. Analyze the performance of PID controller and fuzzy logic controller by manipulating the input angle and observe the graphs of output angle.
6. Compare the performance of PID controller and fuzzy logic controller in terms of steady-state error, settling time and rise time.

## 1.5 Chapter Overview

Chapter 1 introduces the motivation of this project, the overall problem faced by people and how to solve that specific problem. The objectives and scope covered in this project are listed accordingly.

Chapter 2 gives an overview on the current context of this research topic and the relating past studies based on similar topics. Background theory is included in this section. It is divided into topic-related sub-sections and the sections are discussed one by one.

Chapter 3 shows the methodology of project implementation including project implementation flow chart, project experiment flow chart and the overview of the system. The equipments used in this project are introduced, whereas the related equations regarding this project is outlined and derived.

Chapter 4 shows the results and discussion including the open-loop and closed loop simulation results. The results are shown textually and visually. The technique of system identification using Simulink block diagram is used. System transfer function is obtained for further analysis. PID and fuzzy logic controllers are implemented and compared in terms of their performance to decide whether which type of controller is more suitable to control the motion of robotic arm.

Chapter 5 is the conclusion made based on the findings on the previous chapters. Recommendation is made based on the conclusion to get improvement for the future research.

## CHAPTER 2

### LITERATURE REVIEW

#### 2.1 Introduction

A human arm motion has been taken into consideration during the set-up designing stage. It was considered that it should allow for the greatest angular displacement possible, and that it should be able to transport the greatest possible mass at the tip. In this section, the structure and components which make up this whole project are investigated. Research is done in selecting the proper type of controllers to be used in the system.

#### 2.2 Robotics

In the manufacturing process, most of the industrial automated tasks are carried out by specialized machines which are designed to carry out predetermined functions. The inflexibility and generally high cost of these machines have led to a public interest in robots which are capable of performing a variety of manufacturing tasks at lower production costs and greater flexibility in works.

A definition used by the Robot Institute of America gives a more precise description of industrial robots: “A robot is a reprogrammable multi-functional manipulator designed to move materials, parts, tools, or specialized devices, through variable programmed motions for the performance of a variety of tasks [3].” Figure 2.1 shows the mechanism of robots. It is a closed loop system with feedback path.

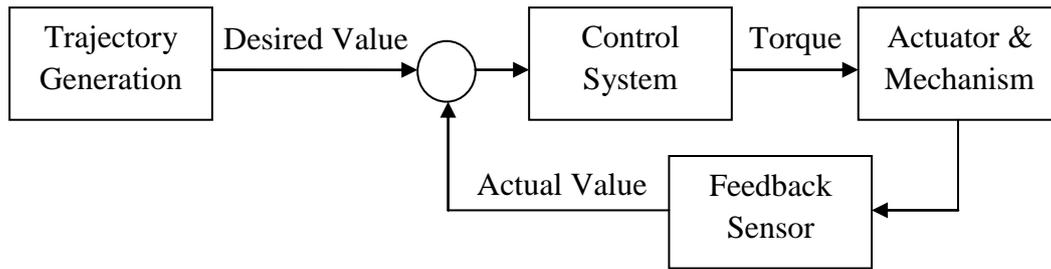
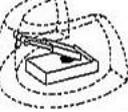
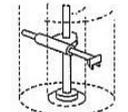
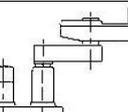
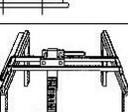
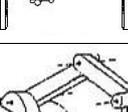


Figure 2.1: Robotic mechanism [3]

### 2.3 Robotic Arm

The main function of human's arm is to move the hand and fingers from one place to another. Likewise, the function of a robotic arm is to move the end effectors from one place to another. Generally the robotic arms are classified as six types. Table 2.1 shows the classification and characteristics of robotic arms.

Table 2.1: Classification and characteristics of robotic arms [4]

	Type	Diagram	Number of Joint	Characteristic
1	Rectangular Coordinate Robot		2 prismatic joints	Principal axes of control are linear
2	Spherical Coordinate Robot		1 prismatic joint & 2 revolute joints	Allow full rotation throughout a spherical range
3	Cylindrical Coordinate Robot		2 prismatic joints & 1 revolute joint	Operate on a cylindrical axis
4	SCARA Robot		2 parallel revolute joints & 1 additional prismatic joint	For pick-and-place work
5	Cartesian/Gantry Robot		3 cylinder joints	Coincident with the standard X-Y-Z Cartesian axis
6	Articulated Robotic Arm		All revolute joints	Used for complex assembly operations

In this section, an upper limb of robotic arm with 1 degree of freedom is discussed as the project only focus on the upper limb part. Generally the arm is connected to a motor and the rotation of motor leads to the motion of arm. Figure 2.2 illustrates an upper limb of robotic arm with 1 degree of freedom.

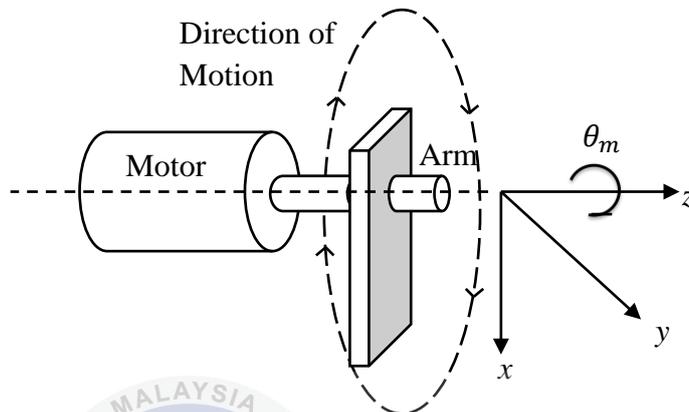


Figure 2.2: Illustration of a 1-DOF upper limb of robotic arm

#### 2.4 Motor

Motors turn electrical energy into mechanical energy and produce the torque required to move to the desired target position. Motor selection and mechanical design is a crucial part in the process of designing motion control system [5]. Table 2.2 shows the comparison between several types of motors in terms of their pros and cons.

Table 2.2: Comparison between the functions of motors [5]

	<b>Stepper Motor</b>	<b>Brushed DC Servo Motor</b>	<b>Brushless DC Motor</b>	<b>Brushless Servo Motor</b>
Pros	<ul style="list-style-type: none"> <li>✓ Inexpensive</li> <li>✓ No feedback is required</li> <li>✓ Good low-end torque</li> <li>✓ Clean rooms</li> </ul>	<ul style="list-style-type: none"> <li>✓ Inexpensive</li> <li>✓ Moderate speed</li> <li>✓ Good high-end torque</li> <li>✓ Simple drives</li> </ul>	<ul style="list-style-type: none"> <li>✓ Excellent torque at low speed</li> <li>✓ Don't need complex power supply</li> <li>✓ Low maintenance</li> <li>✓ High efficiency</li> <li>✓ Long lifespan</li> </ul>	<ul style="list-style-type: none"> <li>✓ Maintenance-free</li> <li>✓ Long lifetime</li> <li>✓ No sparking</li> <li>✓ High speeds</li> <li>✓ Clean rooms</li> <li>✓ Quiet</li> </ul>
Cons	<ul style="list-style-type: none"> <li>✗ Noisy and resonant</li> <li>✗ Rough performance at low speeds</li> <li>✗ Poor high-speed torque</li> <li>✗ Not for hot environments</li> <li>✗ Not for variable loads</li> </ul>	<ul style="list-style-type: none"> <li>✗ Maintenance required</li> <li>✗ No clean rooms</li> <li>✗ Brush sparking causes EMI and danger in explosive environments</li> </ul>	<ul style="list-style-type: none"> <li>✗ High initial cost</li> </ul>	<ul style="list-style-type: none"> <li>✗ Expensive</li> <li>✗ Complicated drives</li> <li>✗ Require tuning of control loop parameters</li> </ul>
Fields	Positioning, micro-movement	Velocity control, high-speed position control	Position control	Robotics, pick-and-place, high-torque applications

Actuation of a robotic arm requires high torque, relatively moderate speed and to achieve accurate positioning. After comparing these motors with each other, brushless DC motor is chosen to be used in this project due to its excellent torque performance and apparently least disadvantages.

## 2.5 Case Study of Controllers

During the past few decades, modeling and control of flexible robot arms have attracted increasing attention. Various basic research and control techniques for flexible manipulators have been investigated to certain extent of success [4].

There are many methods of control available for robotic arm. Types of control of such non-linear systems can be divided into two major categories: (i) traditional control; and (ii) non-traditional (intelligent) control. Traditional control includes: (1) adaptive control; (2) robust control, and (3) robust-adaptive 'hybrid' control. Intelligent control can be classified into: (1) learning control such as neural network-based control; (2) fuzzy-logic control; and (3) genetic control [6]. These methods of control are summarised in Figure 2.3.

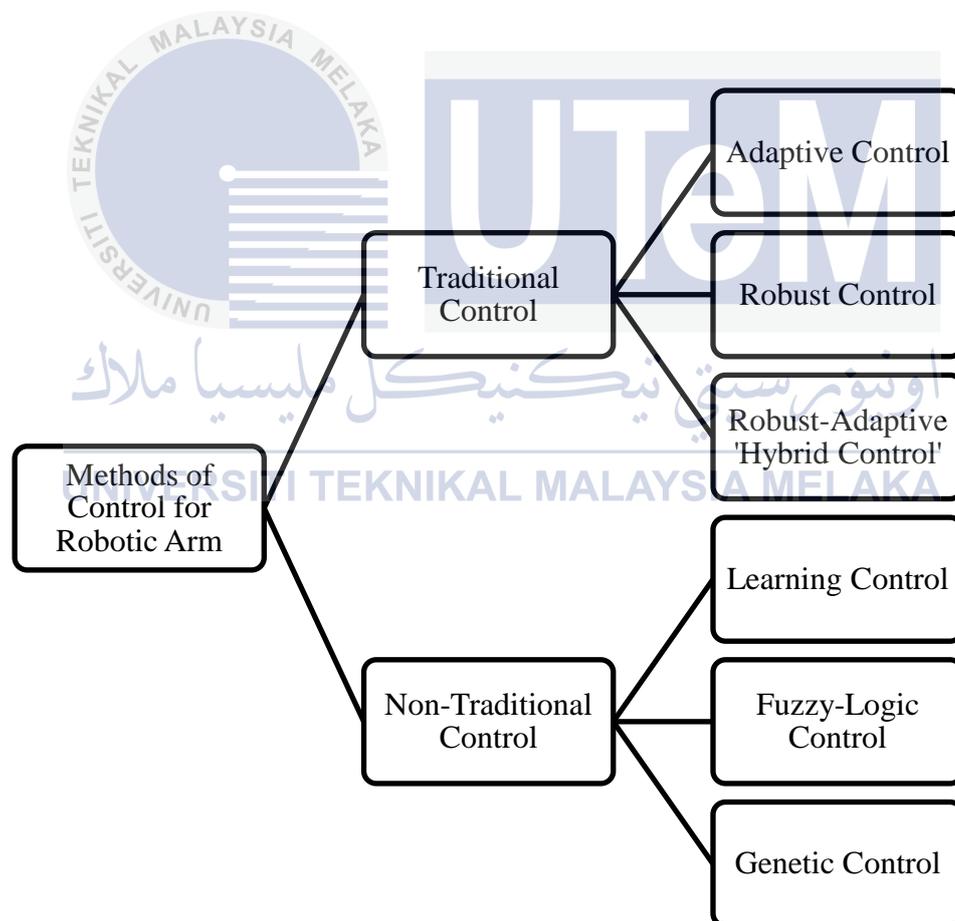


Figure 2.3: Types of control methods for robotic arm

C.S. Lee and R.V. Gonzalez studied and compared three different control methods for a muscle-like actuated arm in two degrees-of-freedom for elbow flexion/extension (f/e) and forearm pronation/supination (p/s) [7]. Figure 2.4 shows the movement of arm for elbow flexion/extension and elbow pronation/supination.

In their study, electromyogram (EMG) signal is employed to determine the control signal used to actuate the muscle cylinders. The first algorithm is fuzzy controller with EMG signals and position error as control inputs. The second algorithm is fuzzy-MA controller, which incorporated moment arm information into the existing fuzzy logic controller. The third algorithm is the conventional Proportional-Integral-Derivative (PID) controller, which operated solely on position and integration error.

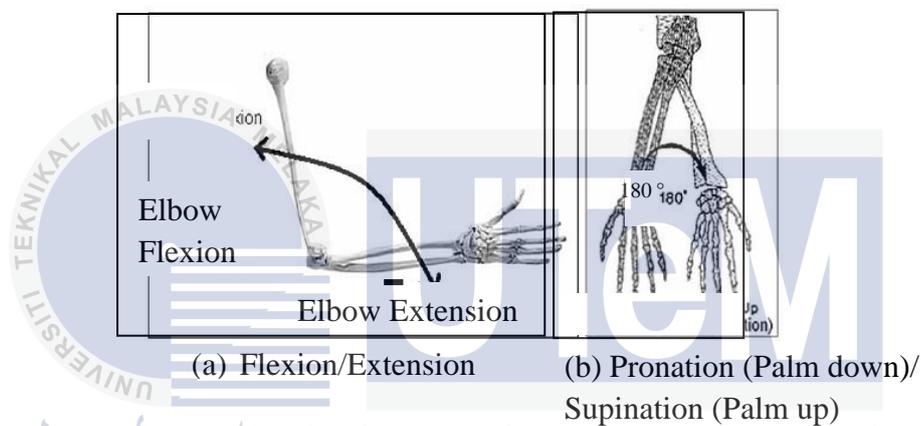
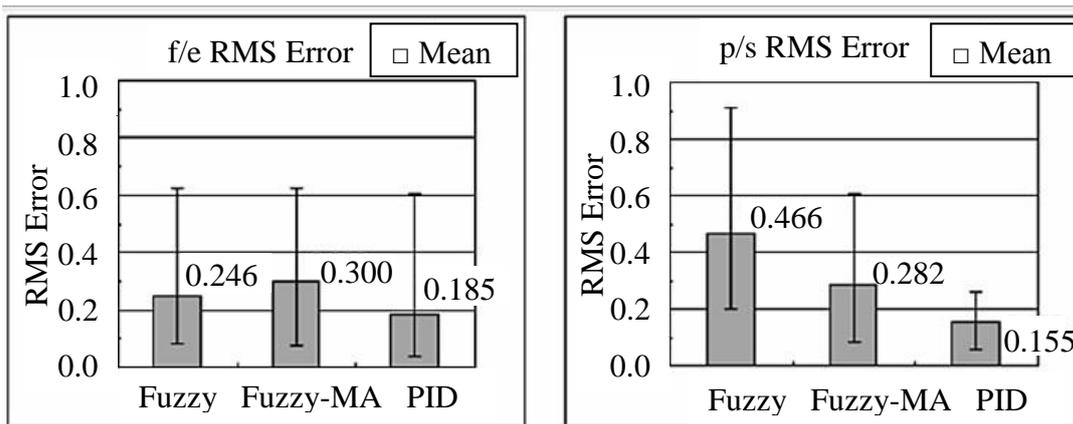


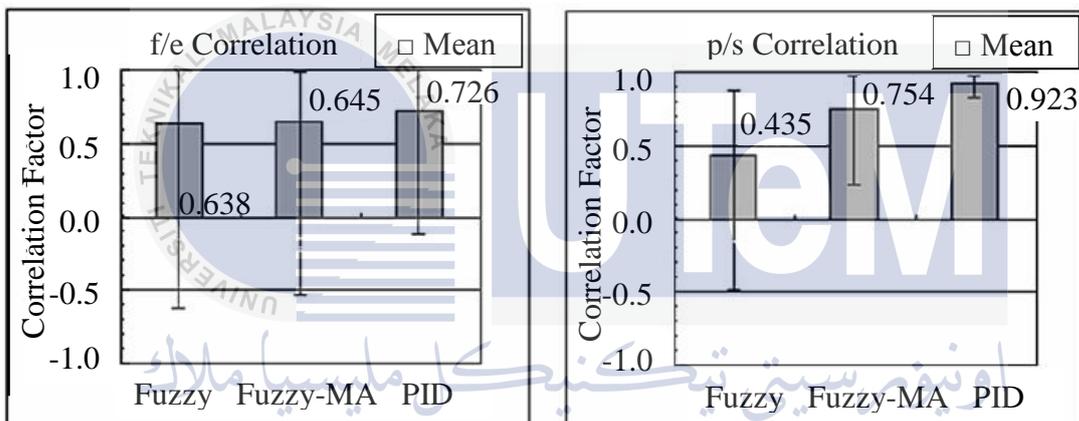
Figure 2.4: Different arm movements [7]

The results are analyzed in terms of (a) Root-Mean-Square (RMS) error, (b) correlation factor between the actual and the desired positions for each degree of freedom, and (c) mean absolute error (MAE) between the actual and desired positions for each degree of freedom. The results of analysis are shown in Figure 2.5 which compares the RMS error using three controllers. Figure 2.6 shows the comparison of three controllers for correlation factor, whereas Figure 2.7 shows the comparison of three controllers in terms of mean absolute error.



(a) Flexion/extension (f/e) RMS error (b) Pronation/supination (p/s) RMS error

Figure 2.5: Comparison of three controllers for RMS error. Bar charts represent mean values for twenty two motions and error bars represent maximum and minimum values [7]



(a) Flexion/extension (f/e) correlation factor (b) Pronation/supination (p/s) correlation factor

Figure 2.6: Comparison of three controllers for correlation factors [7]

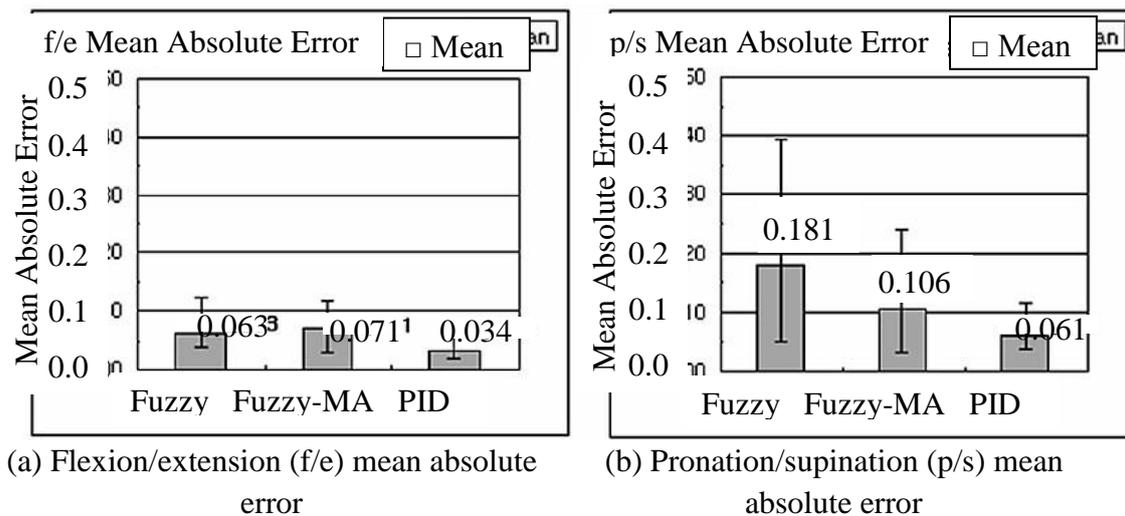


Figure 2.7: Comparison of three controllers for mean absolute error [7]

The results indicated that using the moment arm information in the fuzzy controller noticeably improved p/s motion control. However, better control results were obtained in the PID controller compared to the fuzzy controllers. It is because PID Controller quickly recovered the position error that was initially present. The fuzzy controller and Fuzzy-MA controller could not quickly recover the initial error.

This study showed that moment arms information incorporated into the fuzzy logic control technique improved the mechanical arm's response. However, a PID controller provides better accuracy than the EMG driven fuzzy based controllers [7].

Despite the fact that all these controllers are usually designed individually, there are also researches which combine the usage of two or more controllers. James Carvajal et al. proposed a new fuzzy logic proportional-integral-derivative (PID) controller by combining the structure of conventional PID controller and fuzzy logic controller. The main purpose of their study is to control some known non-linear system, such as a robot manipulator [8].

This controller is developed by first describing the discrete-time linear PID control law. Then, the PID structure is modified by progressively deriving the steps necessary to incorporate a fuzzy logic control mechanism. Throughout the study, the proposed fuzzy PID controller is examined for its ability to control linear and nonlinear plants. Also, its performance is evaluated in comparison with the corresponding conventional PID controller [8].

To investigate the robustness of the two controllers, experiments are carried out for fuzzy PID controller and conventional PID controller. Figure 2.8 shows the results of simulation.

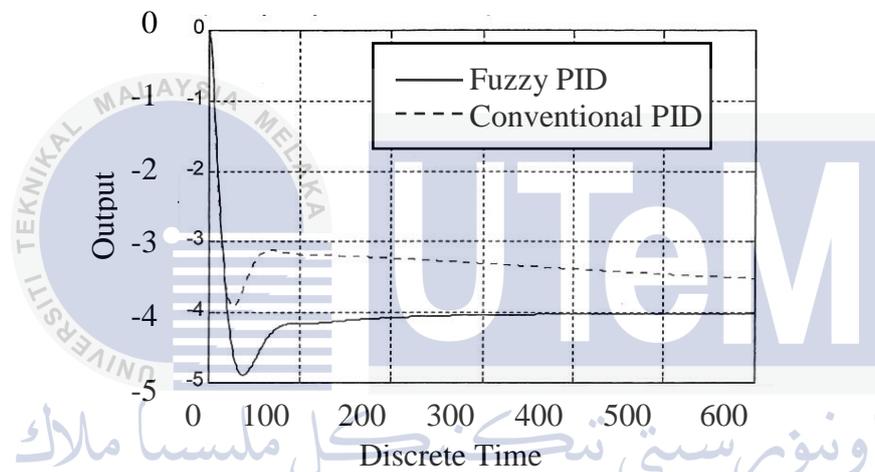


Figure 2.8: Simulation of a third-order system (demonstrates robustness) [8]

The result indicates that the fuzzy controller reached the desired setpoint an order of magnitude faster than the linear controller, implying that the fuzzy controller is more robust in terms of hitting the setpoint in a reasonable amount of time. The fuzzy PID controller derived in this research successfully demonstrated better performance than the conventional PID controller for many cases.

A study by P. Sooraksa and Guanrong Chen proposed a new control technique which is similar as the previous research. A mathematical model for the “shoulder-elbow-like” single flexible-link robot arm with damping is investigated. A fuzzy-logic-based  $(PI+D)^2$  control scheme is introduced and developed for vibration suppression and set-point tracking. Results of computer simulation have demonstrated that the designed controller can suppress the arm vibration and control the tip to track a set-point successfully with satisfactory accuracy and robustness [9].

Another study by Hassan B. Kazemian introduced the self organizing fuzzy (SOF) controller. It is an extension of the rule-based fuzzy logic controller with an additional learning capability. The application of the self-organizing fuzzy PID (SOF-PID) controller to a 2-link non-linear revolute joint robotic arm is studied through path tracking trajectories at the setpoint [10].

The rule-based fuzzy PID is a non-learning controller, and the rules in the rule base are designed for a specific process in mind. However, the self-organising fuzzy PID controller is a learning controller. The rule creation of the SOF-PID controller produces its own control strategies, where there are no rules in the rule base at first.

For the purpose of comparison, the same experiments were repeated using the self-tuning controller subject to the same data supplied at the setpoint. Figure 2.9 shows the comparison between self-organizing fuzzy PID controller and self-tuning controller.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

The results show that the SOF-PID controller is better in tracking a path and adapting itself to constant changes at the setpoint and from the process under control than self-tuning controller. The TITO self-tuning controller initially displays a noticeable amount of oscillation before it is quickly settles.

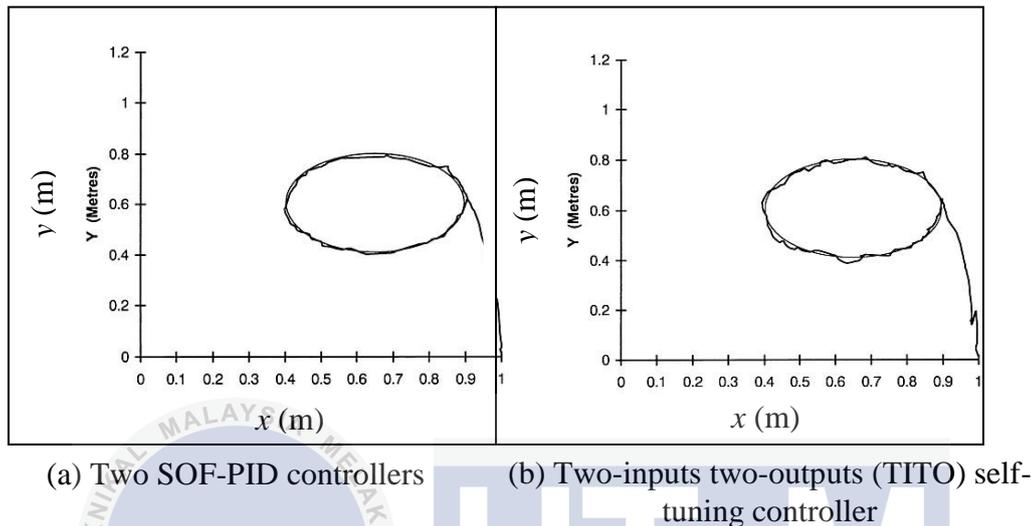


Figure 2.9: Tracking an ellipse [10]

Aron Pujana-Arrese et al. investigated the characteristics of pneumatic artificial muscles by building an experimental one-degree-of-freedom set-up based on pneumatic muscles manufactured by Festo. The experimental set-up is non-linear and very difficult to control properly. As a reference, an enhanced PID controller was designed. A robust controller  $H_\infty$  and a sliding-mode controller based on an observer were designed and implemented at the same time. Also, a position controller based on an internal pressure loop for each pneumatic muscle was tuned up. Comparison is made by means of experimental results regarding each of the four position controllers [11].

Throughout the experiments, total three different areas of the displacement range are being tested. Figure 2.10 shows the experimental response to a ramp input of  $10^\circ$  and a slope of 20 %s applied in the upper displacement zone, where the mass at the tip is 3kg.

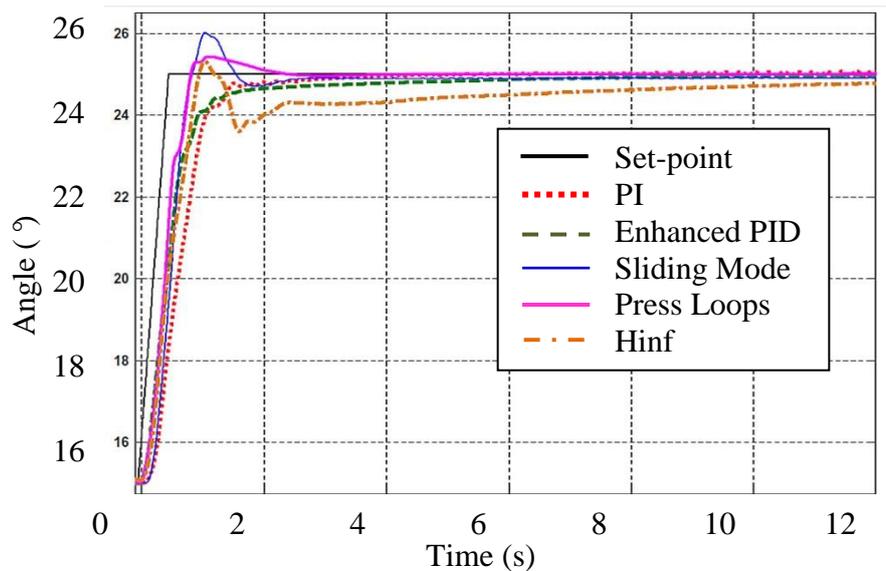


Figure 2.10: Experimental results in the upper displacement zone [11]

With the PI and enhanced PID controllers being implemented in the system, there is no overshoot. However, the settling time is quite high. The sliding-mode algorithm is the one with the biggest overshoot. On the other hand, the  $H_\infty$  (Hinf) controller takes a long time to eliminate the steady-state error. The algorithm based on the pressure loops takes the arm to 25° before any other.

Based on the analysis on the experiments for four types of controller, the results obtained with the classic PI, enhanced PID controller and  $H_\infty$  controller would have to be tuned in different operating zones. Despite the change in the pneumatic circuit, a control algorithm based on the independent control of the pressure of each muscle is designed. The results obtained are the best in terms of performance levels and for compensating the non-linearity of the prototype. The study of robustness in comparison with the load did produced satisfactory results [11].

## 2.6 Summary

A robotic arm of single degree-of-freedom (1 DOF) is constructed in this project. DC geared motor is chosen to be used after comparing with other types of motors. Throughout the study on research papers, better understanding on the characteristics of each type of controllers are obtained. The information regarding the research papers are summarized into Table 2.3.

For the implementation of this project, the PID controller and fuzzy logic controller are chosen to be compared in terms of their performance such as steady-state error and settling time. This is to verify that the advanced technique might achieve more precise motion control compared to conventional control technique for robotic arm.



Table 2.3: Summary of research papers

	<b>Journal</b>	<b>Controller</b>	<b>Application</b>	<b>Advantages</b>	<b>Disadvantages</b>
1	Fuzzy logic Versus a PID Controller for Position Control of a Muscle-Like Actuated Arm	PID controller, Fuzzy-MA controller, Fuzzy logic controller	Muscle-like actuated arm of two degree of freedom	PID controller quickly recovered the position error that was initially present	Fuzzy controller and the Fuzzy-MA controller could not quickly recover the initial error
2	Fuzzy PID Controller: Design, Performance Evaluation, and Stability Analysis	Fuzzy PID controller	Non-linear plants	<ul style="list-style-type: none"> <li>• More robust in terms of hitting the setpoint in a reasonable amount of time</li> <li>• Able to tolerate the poor selections of the controller gains</li> </ul>	Requires careful tuning to get the set of gains that worked for each case
3	Comparative Study of a Learning Fuzzy PID Controller and a Self-Tuning Controller	Self-organizing fuzzy (SOF) PID controller	2-link non-linear revolute joint robotic arm	Better in tracking a path and adapting itself to constant changes at the setpoint from the process under control than the self-tuning controller	Need to learn the appropriate control rule strategy by itself by going through a series of training (RUNS)
4	Mathematical Modeling and Fuzzy Control of a Flexible-Link Robot Arm	Fuzzy PI+D controller	Shoulder-elbow-like single flexible link robotic arm	<ul style="list-style-type: none"> <li>• Suppress the arm vibration</li> <li>• Control the tip to track the setpoint with satisfactory accuracy and robustness</li> </ul>	Need to be tune manually for working control parameters
5	Modelling in Modelica and Position Control of a 1-DoF Set-up Powered by Pneumatic Muscles	PI controller, Enhanced PID controller, Hinf ( $H_{\infty}$ ) controller, Sliding-mode controller	Pneumatic artificial muscles/McKibben-muscles	<ul style="list-style-type: none"> <li>• With the PI and enhanced PID controller, there is no overshoot</li> <li>• Sliding-mode controller is capable of eliminating the steady-state error.</li> </ul>	<ul style="list-style-type: none"> <li>• Sliding-mode is the algorithm with biggest overshoot</li> <li>• Enhanced PID and <math>H_{\infty}</math> controllers have to be designed or tuned in different operating zones</li> </ul>

## CHAPTER 3

### METHODOLOGY

#### 3.1 Introduction

Research is carried out at the previous chapter. The types of controllers and components to be used are chosen. In this chapter, the procedures during research and experimental setup are introduced. Modeling of the system is done in this chapter.

#### 3.2 Research Methodology

Generally the research methodology is divided into two parts, namely project methodology and experiment methodology. Project methodology is the overall processes taken to complete this final year project, whereas experiment methodology is the procedures of the experimental setup. The research methodology of this final year project is summarized into a Gantt Chart shown in Appendix A.

##### 3.2.1 Project Methodology

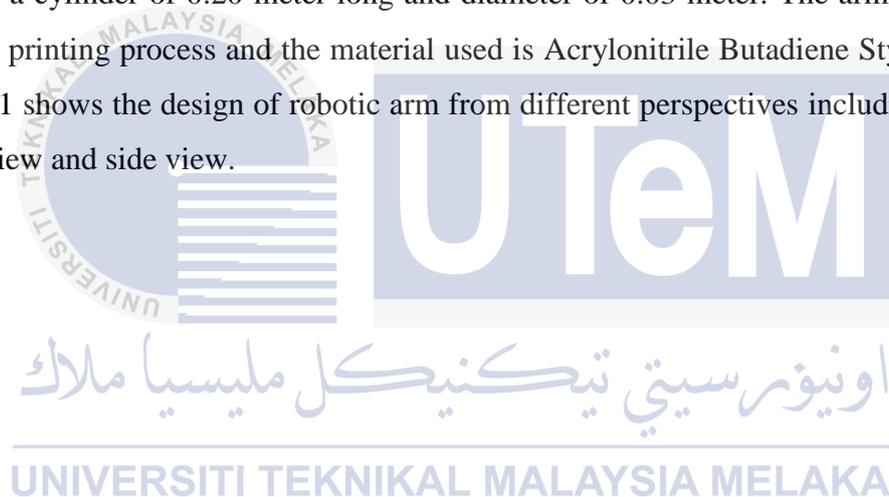
The project research starts with literature review, which is the study on other researchers' findings about this topic. This is useful as it will provide extra knowledge about the research topic. After the literature review, types of components to be used are chosen. Experimental setup is conducted. At the same time, report is drafted whereas analysis is made based on the results obtained. Finally, the final report is written and the overall project is concluded. The project research methodology is summarized into Appendix B.

### 3.2.1 Experiment Methodology

The experimental setup started with the design and fabrication of robotic arm. Then, selection of components is made and the types of controller to be used are decided. Experimental setup is done inside the laboratory. Simulations are conducted using simulation software (MATLAB) installed in host computer. Analysis is made based on the results obtained. The project experiment methodology is summarized into Appendix C.

### 3.3 Structure of Robotic Arm

The robotic arm is assembled to human arm and is divided into two parts which are shoulder and arm. The shoulder is a cube with side length of 0.05 meter, whereas the shape of arm is a cylinder of 0.20 meter long and diameter of 0.03 meter. The arm is fabricated using 3D printing process and the material used is Acrylonitrile Butadiene Styrene (ABS). Figure 3.1 shows the design of robotic arm from different perspectives including top view, bottom view and side view.



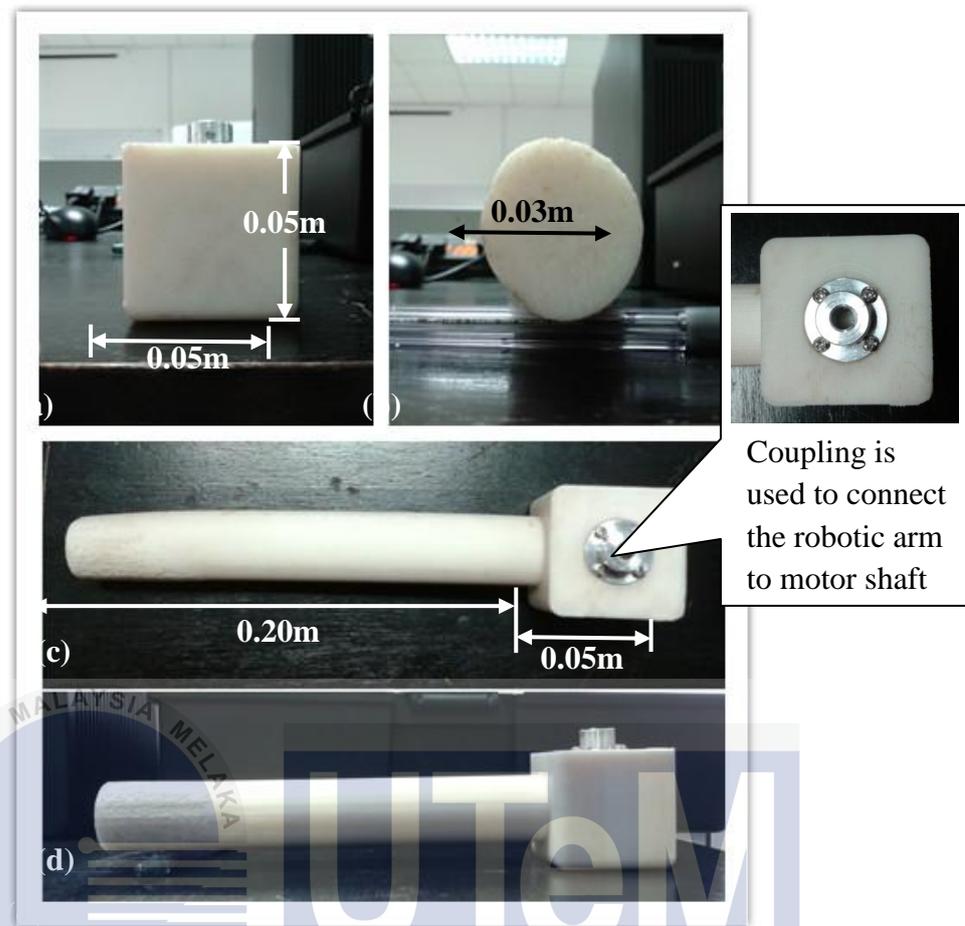


Figure 3.1: Design of robotic arm (a) Top view; (b) Bottom view; (c) Side view; (d) Side view

UNIVERSITI TEKNIKAL MALAYSIA MELAKA  
 اونیورسیتی تکنیکل ملیسیا ملاک  
 UNIVERSITI TEKNIKAL MALAYSIA MELAKA

### 3.4 Hardware Implementation

The hardware components used in this project are shown in this section. A DC geared motor is used to actuate the robotic arm. It has an attached Hall Effect encoder which is used to monitor the movement of arm. Figure 3.2 shows a Cytron 12V DC geared motor with Hall Effect Encoder.

### 3.4.1 Cytron 12V DC Geared Motor with Hall Effect Encoder

When the instruction is set inside the software, Micro-Box will send a voltage output to motor. The motor will then actuates the robotic arm. The 5V Quadrature Hall Effect Encoder is applied to monitor the position and direction of rotation [9]. Table 3.1 denotes the specifications of this motor. The DC geared motor has 6 connection ports. Its detailed connection of is summarized into Appendix D.



Figure 3.2: DC geared motor with encoder and its removable cover [9]

Table 3.1: Specifications of Cytron 12V DC geared motor with encoder [9]

Parameter	Specification
Product Code	SPG30E – 300K
Operating Voltage	4.5V ~ 5.5V
Weight	160 g
Output Power	1.1 Watt
Encoder Resolution	<ul style="list-style-type: none"> <li>• 3 pulses per rear shaft revolution, single channel output</li> <li>• 810 counts per main shaft revolution</li> </ul>
Gear Ratio	270:1
Rated Voltage	12 VDC
Rated Speed	12 rpm
Rated Current	410 mA
Rated Torque	1176 mN.m

### 3.4.2 Micro-Box 2000/2000C (xPC Target Machine)

Micro-Box 2000/2000C used in this project is an element of Electro-Mechanical Engineering Control System (EMECS) by TeraSoft Inc. EMECS provide a platform for investigating a variety of control related problems such as system modeling, system identification, linear control, nonlinear control etc. In addition to hardware, Simulink blocks for the experiments are provided to help users in control design and simulation. [10]

EMECS is made up of three components, the Micro-box 2000/2000C, driver circuit, and servo motor module. In this project, DC geared motor is used and thus, the servo motor provided in EMECS is not used. A Micro-Box module is connected between the motor and computer to act as an interface between them. Table 3.2 lists out the specifications of the Micro-Box module, whereas Figure 3.3 shows its components.

Table 3.2: Specifications of Micro-Box 2000/2000C [10]

Parameter	Specification
Power Supply	Min. 48W
Operating Voltage	9 ~ 36 VDC
Dimension	255(W) x 152(D) x 82(H) mm
Net Weight	2.0kg

### 3.4.3 Motor Driver Circuit

The driver circuit is used to drive the DC geared motor. Once the instructions from host computer are received, the driver circuit will actuate the movement of motor. Figure 3.4 shows a close-up view of the driver circuit.

There are two cable connected to this circuit: the motor cable (yellow colour) and the encoder cable (grey colour). Both cables are required to connect to motor in order to actuate the robotic arm. The detailed connection of driver circuit board is shown in Appendix E.

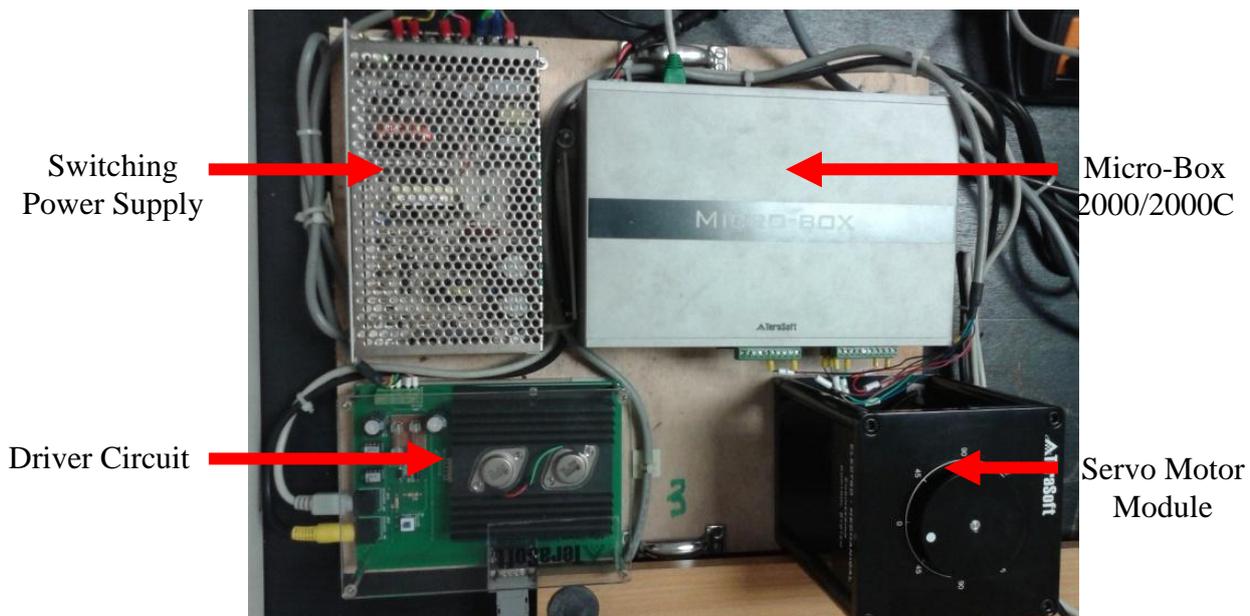


Figure 3.3: Components of Micro-Box module

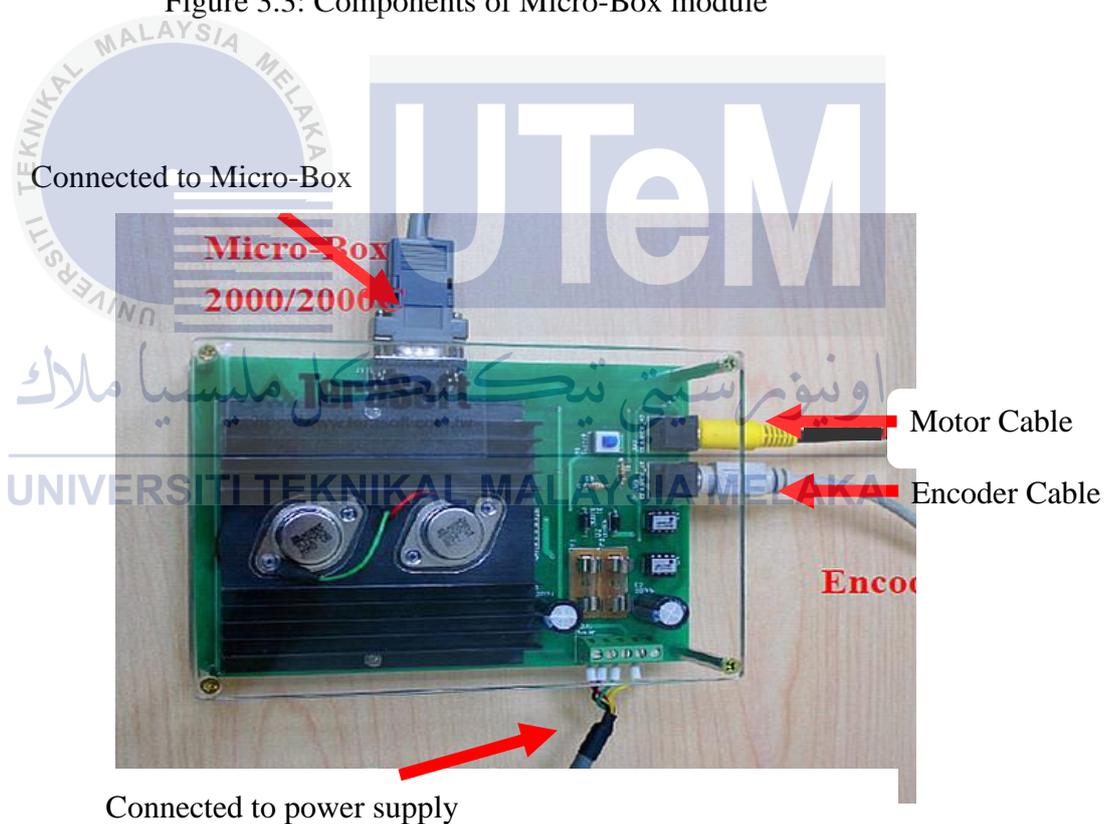


Figure 3.4: Motor driver circuit

### 3.5 System Overview

The experimental setup of the position control system is as shown in Figure 3.5. The purpose of this system is to have the output angle of the motor follow the input angle which is being set by the user. First, the robotic arm is connected to Micro-box, whereas the Micro-box module acts as the interface between the hardware (robotic arm) and the software installed inside the computer.

The Micro-box also acts as a data acquisition unit which obtain data from the host computer and transferred the information as voltage output to the motor driver circuit. The driver circuit will then actuate the movement of the arm. Figure 3.5 indicates the relationship between the elements of the project, whereas Figure 3.6 shows the experimental setup of the project.



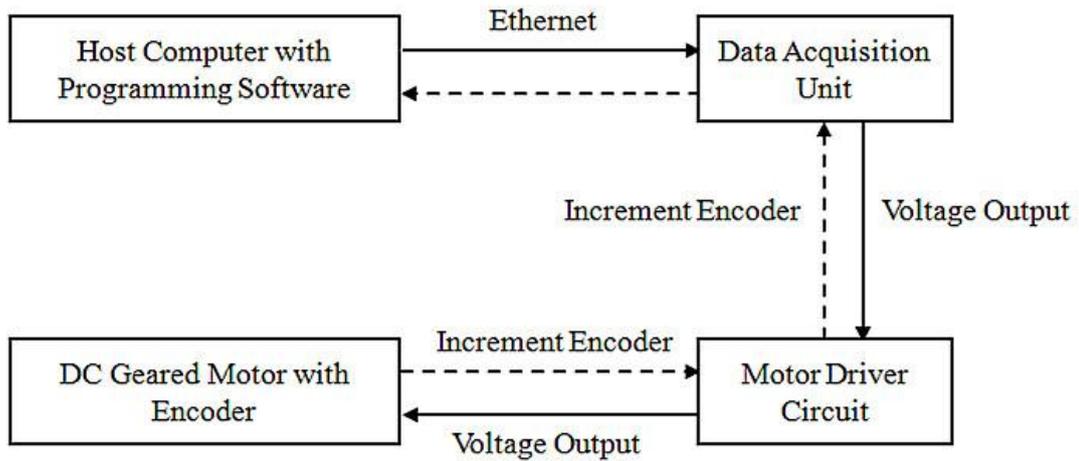


Figure 3.5: System concept

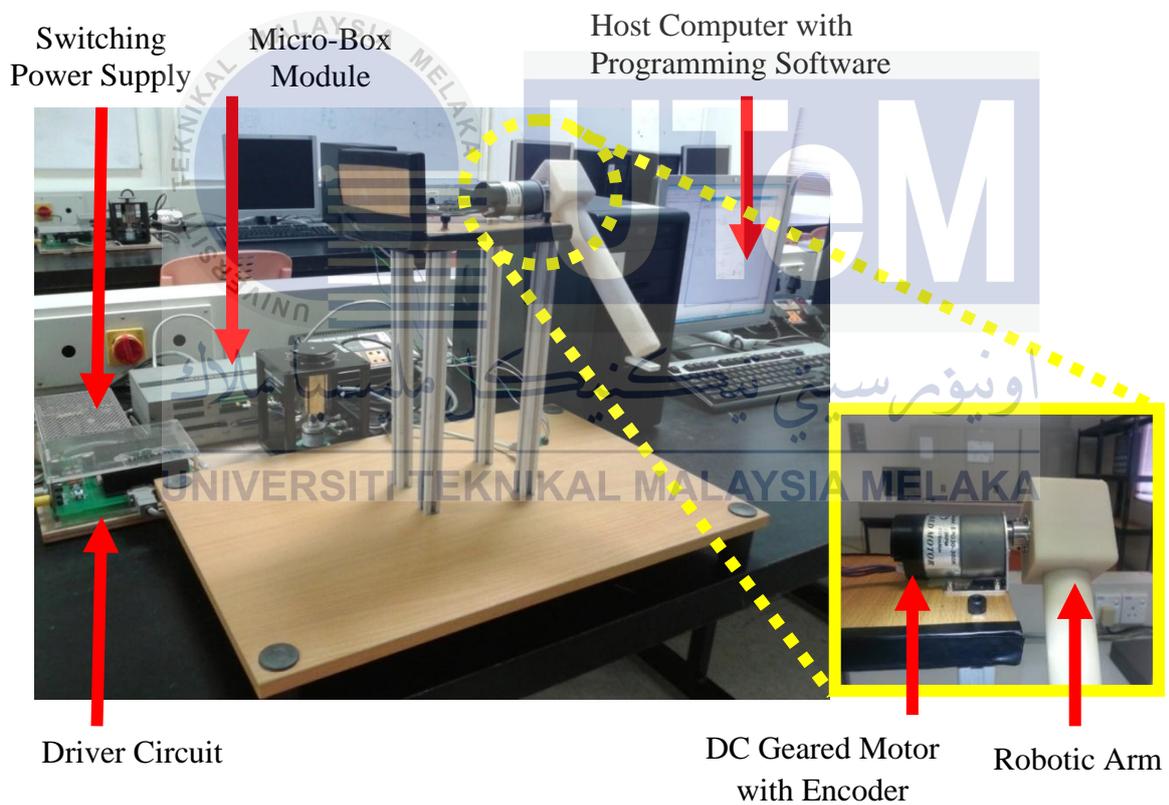


Figure 3.6: Experimental setup of the project

### 3.6 Mathematical Modeling

Generally there are two ways to obtain the transfer function from the system, namely mathematical modeling and System Identification. In this project, both methods are used. Mathematical modeling is the process of deriving transfer function from the system block diagram respectively. It is important to obtain the mathematical equations from the model in order to understand the characteristics and working principles of the model.

#### 3.6.1 Modeling of a DC Geared Motor

In this section, a linear model of a DC geared motor is developed and analyzed. Consider a DC motor, in which its armature circuit is depicted in Figure 3.7. Both the rotor and the shaft of motor are assumed to be rigid. The input of the system is the armature voltage (in Volts) which is driven by an external voltage source.

In this project, the variables being measured are the angular velocity of the shaft (in radians per second) and the shaft angle (in degree). All the parameters involved in mathematical modeling of DC geared motor are shown in Table 3.3.

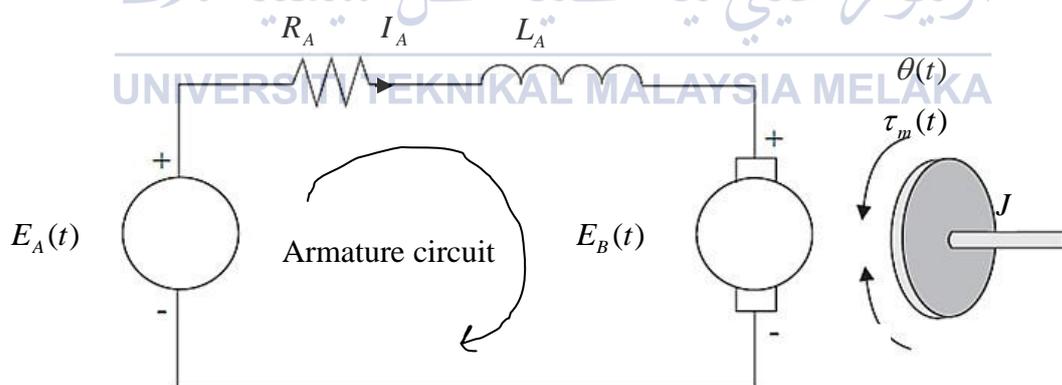


Figure 3.7: Schematic diagram of the DC geared motor [11]

Table 3.3: Parameters in the armature circuit of DC geared motor

Symbol	Definition	S.I. Unit
$E_A$	Armature voltage	V
$E_B$	Back electromotive force (emf)	V
$R_A$	Armature resistance	$\Omega$
$L_A$	Armature Inductance	H
$I_A$	Armature current	A
$\tau_m$	Torque developed by motor	Nm
$\theta$	Angular displacement of motor shaft	rad
$\omega$	Angular velocity of motor	rad / s
$J$	Moment of inertia of motor and load	kgm <sup>2</sup>
$b$	Frictional constant of motor and load	Nm

The torque generated by the motor is proportional to the armature current by a constant of proportionality which named as motor torque constant,  $K$

$$\tau_m = KI_A \quad (3.1)$$

The back electromotive force (emf) is related to the angular velocity of the motor. Since the current carrying armature is rotating in a magnetic field, its voltage is proportional to the angular velocity.

$$E_B = K\omega = K \frac{d\theta}{dt} \quad (3.2)$$

By combining Newton's law and Kirchoff's law,

$$KI_A = J \frac{d^2\theta}{dt} + b \frac{d\theta}{dt} \quad (3.3)$$

$$L \frac{dI_A}{dt} + RI_A = E_A - K \frac{d\theta}{dt} \quad (3.4)$$

By using Laplace Transform, Equation 3.3 and 3.4 now become

$$KI_A(s) = Js^2\theta(s) + bs\theta(s) \quad (3.5)$$

$$LsI_A(s) + RI_A(s) = E_A(s) - Ks\theta(s) \quad (3.6)$$

From Equation 3.6,

$$(Ls + R)I_A(s) = E_A(s) - Ks\theta(s) \quad (3.7)$$

$$I_A(s) = \frac{E_A(s) - Ks\theta(s)}{Ls + R} \quad (3.8)$$

With Equation 3.8 being substituted into Equation 3.5,

$$K \frac{E_A(s) - Ks\theta(s)}{Ls + R} = Js^2\theta(s) + bs\theta(s) \quad (3.9)$$

From Equation 3.9,

$$K \frac{E_A(s) - Ks\theta(s)}{Ls + R} = (Js + b)s\theta(s) \quad (3.10)$$

$$KE_A(s) - K^2s\theta(s) = (Ls + R)(Js + b)s\theta(s) \quad (3.11)$$

$$KE_A(s) = [(Ls + R)(Js + b) + K^2]s\theta(s) \quad (3.12)$$

Hence, the transfer function of the input voltage to the output angle is

$$G_1(s) = \frac{\theta(s)}{E_A(s)} = \frac{K}{(Ls + R)(Js + b)s + K^2s} \quad (3.13)$$

The block diagram of the DC geared motor is shown at Figure 3.8.

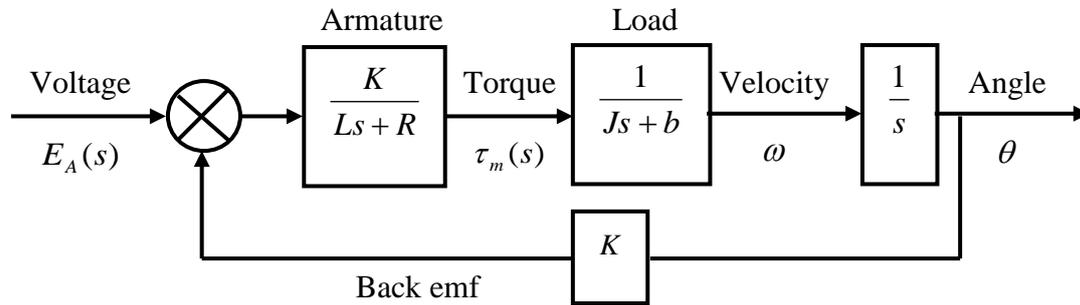


Figure 3.8: Block diagram of the DC geared motor [11]

From the block diagram, the transfer function of the input voltage to the angular velocity is

$$G_2(s) = \frac{\omega(s)}{E_A(s)} = \frac{K}{(Ls + R)(Js + b) + K^2} \quad (3.14)$$

From the block diagram of the motor, it can be observed that the transfer function of the motor is a third order transfer function. However, the structure of Equation 3.13 can be relatively simple as follows, which is a second order transfer function [13].

$$G_1(s) = \frac{\theta(s)}{E_A(s)} = \frac{K}{s(s + \alpha)} \quad (3.15)$$

For the simplicity of this project, the second order transfer function is applied throughout the experiments. During the stage of system identification, the numerical values for the transfer function of motor are obtained through experiments and are used for further analysis at Chapter 4.

### 3.6.2 Torque

In this section, the formulae of the torque generated by the motor are discussed. Table 3.4 shows the parameters involved in this section.

Table 3.4: Parameters relative to the torque of DC geared motor

Symbol	Definition	S.I. Unit
$L$	Angular momentum	$kgm^2$
$J$	Moment of inertia at motor and load	$kgm^2$
$\omega$	Angular velocity	$rad/s$
$\alpha$	Angular acceleration	$rad/s^2$
$b$	Frictional constant of motor and load	$Nm$

Angular momentum of the robotic arm is the product of the moment of inertia of an object and its angular velocity.

$$L = I\omega \quad (3.16)$$

Torque,  $\tau$  of the robotic arm is a time derivative of angular momentum.

$$\tau = \frac{dL}{dt} \quad (3.17)$$

If  $I$  is constant, then

$$\tau = \frac{dL}{dt} = J \frac{d\omega}{dt} = J\alpha \quad (3.18)$$

From Equation 3.18, the angular acceleration can be obtained by adding a derivative block after the angular speed derivative block in Simulink. On the other hand, equation of torque in terms of motor shaft angle position is shown.

$$\tau = J \frac{d^2\theta}{dt^2} \quad (3.19)$$

Hence, the torque can be obtained by adding two derivative blocks after the angular position of the motor.

### 3.6.3 Moment of Inertia

The robotic arm fabricated in this project consists of two parts, a cube with 0.05m side length and a cylinder with 0.20m length. The standard formula of moment of inertia of the cube and cylinder as listed as follows. Table 3.5 shows the parameters involved in calculating the moment of inertia of the robotic arm.

$$I_1 = \frac{m_1 s^2}{6} \quad (3.20)$$

$$I_2 = \frac{m_2 l^2}{3} \quad (3.21)$$

Table 3.5: Parameters relative to the moment of inertia of robotic arm

Symbol	Definition	S.I. Unit
$I_1$	Moment of inertia of the cube	$kgm^2$
$I_2$	Moment of inertia of the cylinder	$kgm^2$
$m_1$	Mass of the cube	$kg$
$m_2$	Mass of the cylinder	$kg$
$s$	Side length of the cube	$m$
$l$	Length of the cylinder	$m$

Total moment of inertia of robotic arm,

$$I = I_1 + I_2 = \frac{m_1 s^2}{6} + \frac{m_2 l^2}{3} \quad (3.22)$$

Thus, the moment of inertia can be calculated by knowing the numerical values of these parameters.

### 3.6.4 Calculation of Moment of Inertia

The equation of moment of inertia for robotic arm is obtained in the previous section. In this section, the value of moment of inertia is calculated. Firstly, the mass of cube and cylinder are measured using electronic weighing scale. Then, the values obtained are substituted into the derived equation. The data is tabulated in Table 3.6.

Table 3.6: Mass of components of robotic arm

Repeatability	Mass of Cube (kg)	Mass of Cylinder (kg)
1	0.155	0.184
2	0.157	0.184
3	0.156	0.183
4	0.157	0.185
5	0.156	0.182
6	0.155	0.184
7	0.157	0.183
8	0.157	0.183
9	0.156	0.185
10	0.157	0.182
Mean	0.156	0.184
Std Dev	0.000823	0.00108

From Equation 3.7, total moment of inertia of robotic arm,

$$I = \frac{m_1 s^2}{6} + \frac{m_2 l^2}{3}$$

where  $m_1 = 0.156$  kg,  $m_2 = 0.184$  kg,  $s = 0.05$  m,  $l = 0.20$  m

$$\text{Total moment of inertia, } I = \frac{(0.156)(0.05)^2}{6} + \frac{(0.184)(0.20)^2}{3}$$

$$= 0.0024 \text{ kg} \cdot \text{m}^2$$

$$= 0.024 \text{ Nm}^2$$

### 3.7 Open Loop Control

The purpose of open loop control is to study the dynamic system behavior. In this project, the open loop control system is used to obtain the transfer function of motor. Open loop simulations are carried out using the technique of System Identification. System Identification is a MATLAB toolbox which is used to obtain mathematical models from the measured input and output data. It helps to identify the system transfer function for further analysis. Generally the simplified transfer function of the motor is a second order transfer function as shown below.

$$G(s) = \frac{As + B}{Cs^2 + Ds + E} \quad (3.23)$$

The process of System Identification is carried out in open loop condition. The procedures are repeated for 10 times to test the repeatability. The values of mean and standard deviation are obtained. The parameter value closest to the mean value and has smallest standard deviation is chosen as the transfer function of motor. This transfer function will then be substituted into the system for further analysis. Figure 3.9 shows the block diagram of the open loop simulation.

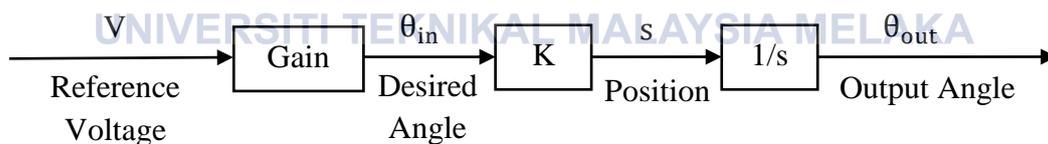


Figure 3.9: Block diagram of open loop system

For the open loop simulation, the sampling time is set to 0.001 second. This means that the reading is taken at every millisecond. A delay of 0.1 second is set to enable a clearer view of the input and output signals. The input voltage is the only parameter being varied, which ranged from 1 Volt to 5 Volts. The output data is obtained in degree form. Table 3.7 shows the parameters and each of their numerical values in open loop simulation.

Table 3.7: Parameters of the open loop simulation

Parameter	Numerical Value
Input Voltage	1 ~ 5Volt
Simulation time	1s
Delay	0.1s
Sampling time	1ms
Input type	Step input

### 3.8 Controller Design

Once the system is set up experimentally, it is crucial to design a controller to control the parameters in the system. In this project, the parameter that is being controlled is output angle of the motor. Thus, the controller must be able to achieve precise motion control and to minimise steady-state error.

#### 3.8.1 PID Controller

A PID controller uses proportional, integral and derivative functions to control the input signal before sending it to the plant unit. The angular position of DC motor can be controlled to drive the robotic arm [10]. Figure 3.10 shows the block diagram of a typical PID controller. The PID controller has remained as the most commonly used controller in practically all industrial control applications due to its simplicity in architecture. Also, it is conceptually easy to understand and explicit tuning procedures [13].

In this project, an uncompensated system is constructed and analyzed. A PID controller is designed and implemented in the same system. Then, the compensated system is analyzed again to compare the performance in terms of steady-state error and settling time. After the analysis of PID is done, an analysis based on fuzzy logic controller is carried out.

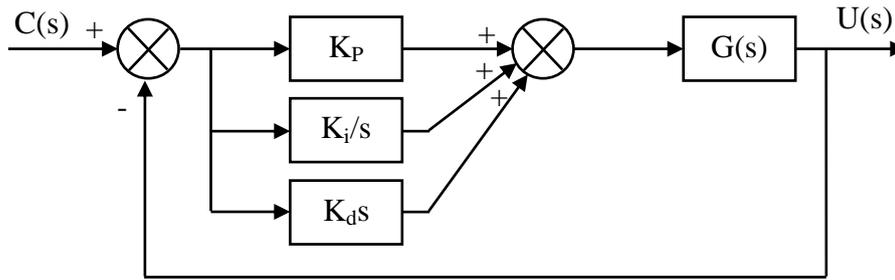


Figure 3.10: Block diagram of a typical PID controller [13]

Closed loop system is formed by adding a feedback to the current system. Figure 3.11 shows the closed loop system with PID controller. Open loop simulation is first carried out and then closed loop simulation. The Simulink block diagrams constructed in this project are shown in Appendix F.

The equation of an ideal PID controller is as follows.

$$u(t) = Kp e(t) + Ki \int_0^t e(\tau) d\tau + Kd \frac{de}{dt} \quad (3.24)$$

It can also be expressed as:

$$u(t) = Kp \left( e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right) \quad (3.25)$$

where Table 3.8 shows the parameters involved in Equation 3.25.

Table 3.8: Parameters related to PID controller

Parameter	Name
$Kp$	Proportional gain
$Ki$	Integral gain
$Kd$	Differential gain
$u$	Control signal
$e$	Control Error
$Ti$	Integral time
$Td$	Derivative time

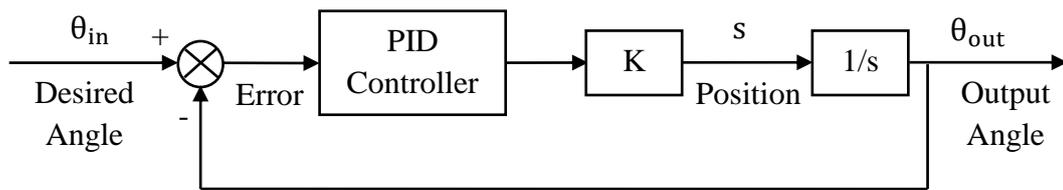


Figure 3.11: Block diagram of closed-loop PID control method

The parameters of PID controller need to be properly tuned to improve the performance of the plant or system. Generally  $K_p$  is used to decrease the rise time, whereas  $K_i$  is used to reduce the steady-state error and settling time.  $K_d$  is used to eliminate the steady-state error. Conventional PID controller shows superb performance when dealing with linear systems. However, it becomes more difficult to achieve the same performance in case of non-linear and highly dimensional systems [12].

### 3.8.2 Tuning Methods

In this project, both trial and error method and Ziegler-Nichols method are implemented to compare whether which method is useful in this case.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

#### 3.8.2.1 Trial and Error Method

In practice control, engineers often use trial and error for the tuning process [14]. It is a relatively easier way to tune the controller. However, it can take a lot of time and does not ensure that the system performance is satisfactory. Table 3.9 shows the effects of manipulating the parameters of PID controller.

Table 3.9: Parameters of transient response and the effects caused by manipulating  $K_p$ ,  $K_i$  and  $K_d$  values [14]

Response	Overshoot	Settling Time	Rise Time	Stability	Transient Response	Steady-State Error
$K_p$	Increase	Small change	Decrease	Small change	Small change	Decrease
$K_i$	Increase	Increase	Decrease	Decrease	Degrade	Eliminate
$K_d$	Decrease	Decrease	Small change	Increase	Improve	No change

### 3.8.2.2 Ziegler-Nichols Method

One of the design methods is based on the process dynamics which can be acquired experimentally. Ziegler-Nichols method is one of the common methods used to determine the process dynamics based on the step response of the system. Ziegler and Nichols proposed two methods for designing PID controllers at 1942, one is time-domain frequency method and one is frequency-domain method [15].

In this project, Ziegler-Nichols frequency-domain method is implemented. The frequency-domain method is based on the frequency response of the system. The parameters of the controller have been given formula in terms of ultimate gain  $K_u$  and ultimate period  $T_u$ . A controller is connected in the system. Parameters are set so that the control action is proportional. The gain is increased slowly until the system starts to oscillate. The gain when oscillation occurs is  $K_u$  whereas the period of oscillation is  $T_u$ . Table 3.10 shows the controller parameters obtained from Ziegler-Nichols frequency response method.

Table 3.10: Controller parameters of Ziegler-Nichols step response method [15]

Controller	$K$	$T_i$	$T_d$
P	$0.5K_u$	-	-
PI	$0.4K_u$	$0.8T_u$	-
PID	$0.6K_u$	$0.5T_u$	$0.12T_u$

### 3.8.3 Fuzzy Logic Controller (FLC)

Fuzzy logic controller is one of the useful control techniques for uncertain and ill-defined non-linear systems. Control actions of a fuzzy controller are described by some linguistic rules [12]. The fuzzy controller uses a form of quantification of imprecise information (input fuzzy sets) to generate by an inference scheme, which is based on a knowledge base of control force to be applied on the system [14]. Figure 3.12 shows the structure of a fuzzy logic controller, whereas Figure 3.13 shows a fuzzy logic control system. Fuzzy logic controller is made up of four main components as depicted in Table 3.11.

Table 3.11: Components of fuzzy logic controller [17]

	<b>Component</b>	<b>Function</b>
1	Fuzzification interface	To transform a real-valued variable into a fuzzy set.
2	Knowledge base containing fuzzy IF-THEN rules and membership functions	To express ambiguous and qualitative relationships
3	Fuzzy reasoning	To derive conclusions from a set of fuzzy IF-THEN rules and one or more premises.
4	Defuzzification interface	To transform a fuzzy set to a real-valued variable.

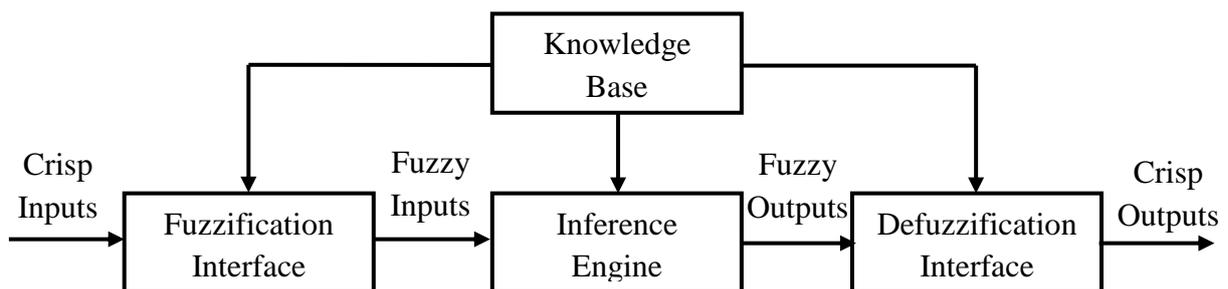


Figure 3.12: Basic structure of fuzzy logic systems [17]

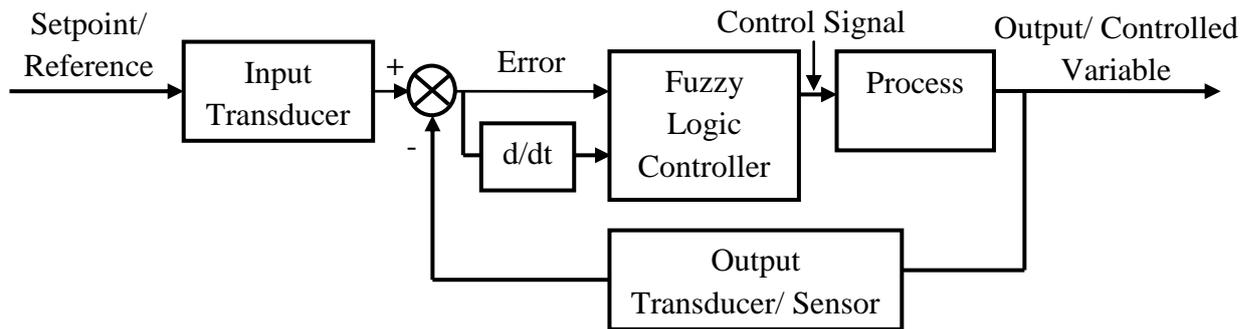


Figure 3.13: Fuzzy logic control system

### 3.8.3.1 Fuzzification Interface

The Mamdani fuzzy system is implemented in this project. In order to control the output angle of the motor, two types of variables are considered, which are error and the rate of change of error. Error is defined as reference angle subtracted by output angle, where rate of change of error is the current error values subtracted by previous error values.

$$\text{Error, } e(t) = u(t) - c(t) \quad (3.26)$$

$$\text{Rate of change of error, } \frac{de(t)}{dt} = e_2(t) - e_1(t) \quad (3.27)$$

These two types of variables are defined in Gaussian membership functions as shown in Figure 3.14, namely:

1. Large Positive (LP)
2. Small Positive (SP)
3. Zero Error (ZE)
4. Small Negative (SN)
5. Large Negative (LN)

The surface view of the fuzzy logic controller enables a clearer view about the range of the input and output variables as shown in Figure 3.15. Both the range of error and rate of change of error are from -5 to 5.

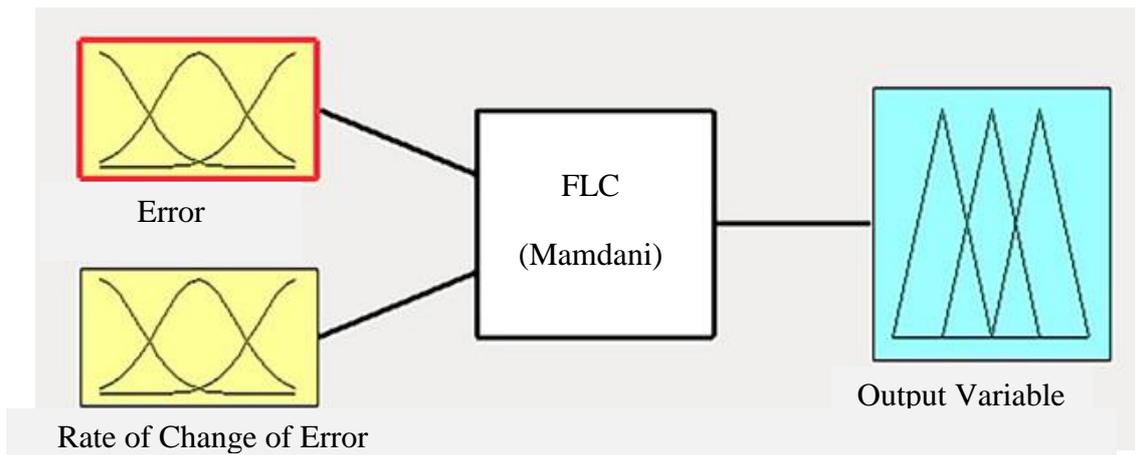


Figure 3.14: The input and output variables of fuzzy logic controller

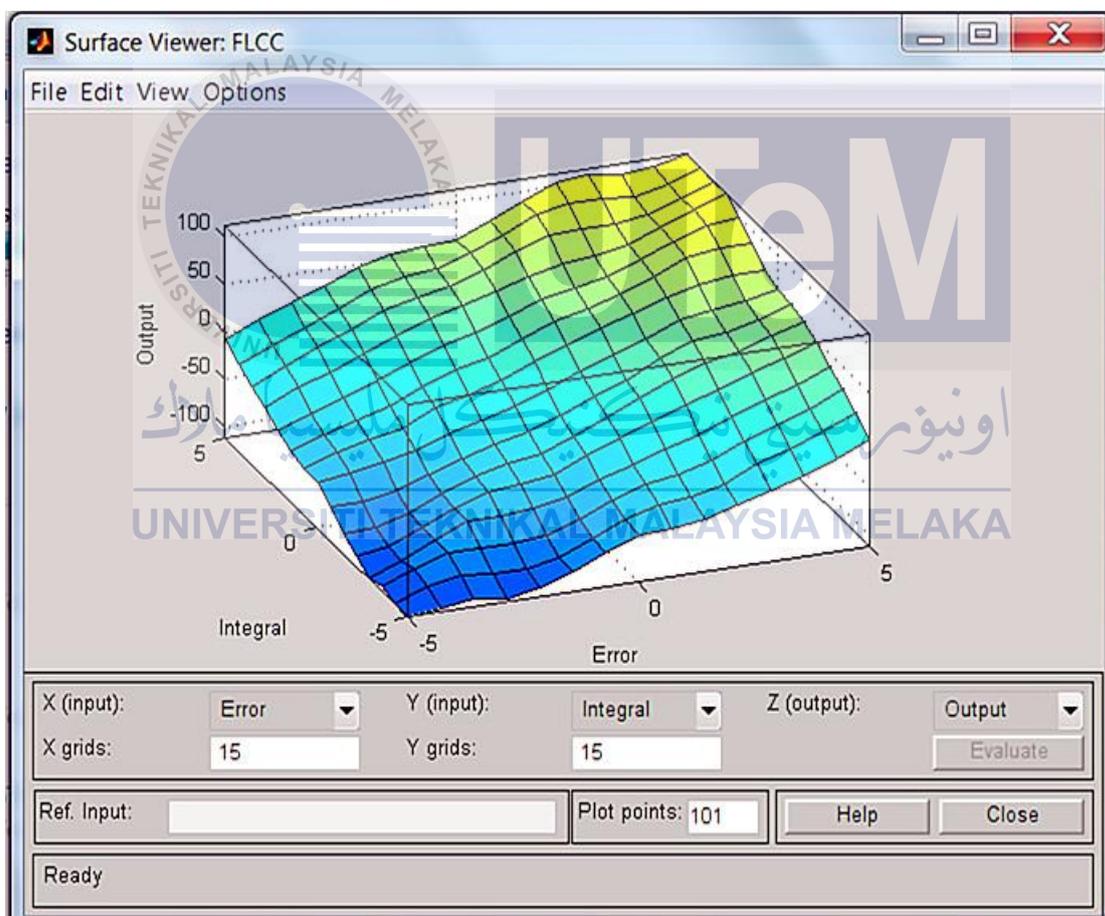


Figure 3.15: The surface view of the fuzzy logic controller

### 3.8.3.2 Knowledge Base/ Rule-Based System

In this system, total 25 rules are generated as shown in Table 3.12. The fuzzy rules is created in the form of “IF  $x$  is A, THEN  $y$  is B” in the rule based system. The complete sets of rules are summarized into Appendix G.

Table 3.12: Rule base system for fuzzy logic controller

		Rate of Change of Error				
		LN	SN	ZE	SP	LP
Error	LN	LN	LN	SN	SN	ZE
	SN	LN	SN	SN	ZE	SP
	ZE	SN	SN	ZE	SP	SP
	SP	SN	ZE	SP	SP	LP
	LP	ZE	SP	SP	LP	LP

### 3.8.3.3 Inference Engine

A typical fuzzy reasoning of the Mamdani fuzzy system is based on the max-min inference method. It is implemented in this project.

### 3.8.3.4 Defuzzification Interface

The output of motor is also defined in Gaussian membership functions as shown in Figure 3.16, namely:

1. Large Positive (LP)
2. Small Positive (SP)
3. Zero Error (ZE)
4. Small Negative (SN)
5. Large Negative (LN)

The range of the motor output is set to -200 to 200 as shown in Figure 3.16.

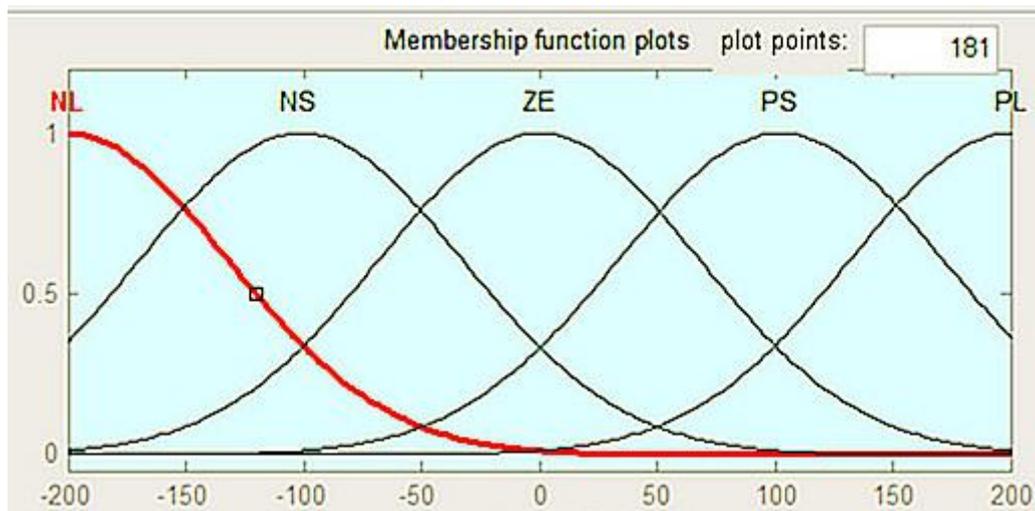


Figure 3.16: The output membership functions

### 3.9 Performance Characteristic (Time Domain)

To analyze a system, its performances or characteristics needed to be known. The parameters shown in Table 3.13 and Figure 3.17 give an adequate description on the step response of a system.

Table 3.13: Parameters associated with step response of the system [13, 16]

	Parameter	Definition
1	Overshoot (%OS)	The amount that the waveform overshoots the steady-state, or final value at the peak time, expressed as the percentage of the steady-state value
2	Rise Time (Tr)	The time taken for the waveform to rise from 0.1 of the final value to 0.9 of the final value
3	Settling Time (Ts)	The time taken until the output falls within and remains within $\pm 2\%$ of the steady-state value
4	Peak Time (Tp)	The time required to reach the first, or maximum, peak

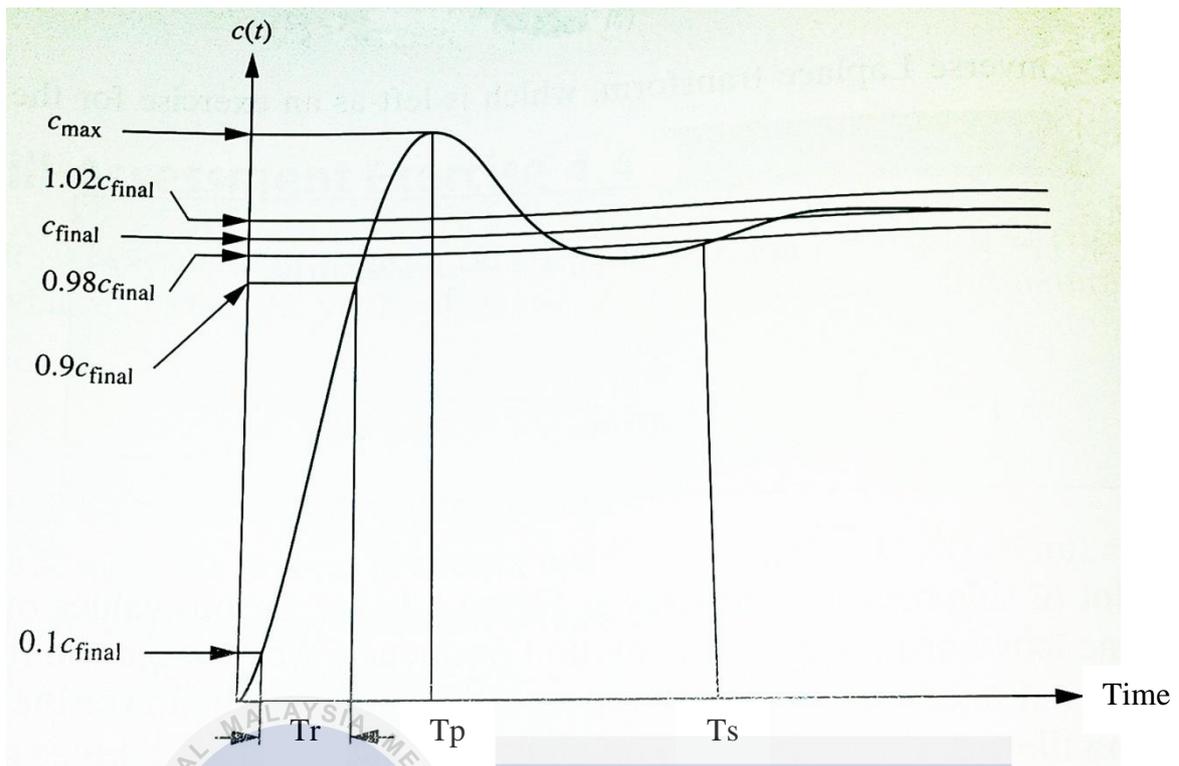


Figure 3.17: Second order underdamped response specifications

Based on Figure 3.17, the compensated system with controllers are analyzed in terms of their performance such as settling time, rise time and steady-state error. Then, the PID controller and Fuzzy Logic controller are compared and the one with better performance is selected.

### 3.10 Output Angle Measurement

The main objective of this project is to control the output angle for the robotic arm. In order for the rotation of the robotic arm be more clearly seen, a card board with angle notation is attached at the motor shaft to measure the corresponding output angle, as shown in Figure 3.18.

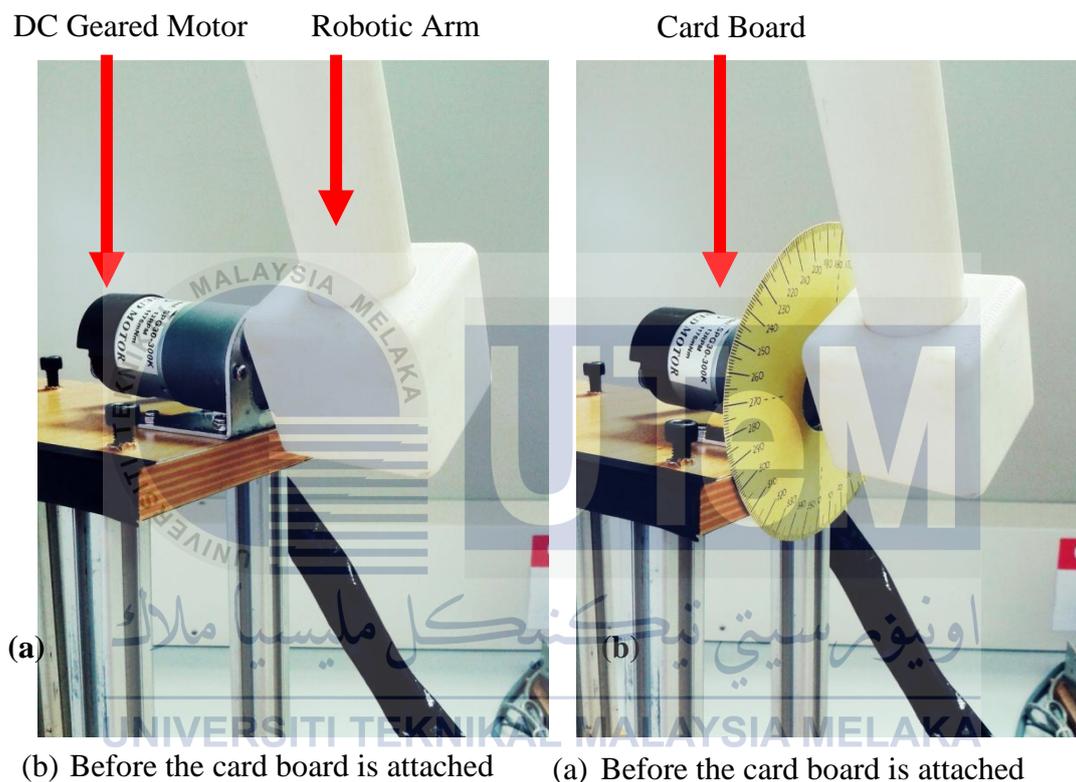


Figure 3.18: The experimental set up of robotic arm

### 3.10 Uncertainty and Reliability

There are several uncertainties that might occur during the experiments during this stage such as human errors, systematic errors and control errors. Human errors can cause incorrect results without noticing by the user. For example, improper wiring can cause failure in obtaining real time signals. Systematic errors might occur when estimation is made based on mathematical modeling. Control errors such like error in constructing Simulink block diagram can cause problems in obtaining output data.

To avoid these errors, extra care need to be taken when the simulation is run. The process of data measurement is repeated for 10 times and an average value is taken from the results. This can ensure the reliability of the data.



## CHAPTER 4

### RESULTS AND DISCUSSION

#### 4.1 Introduction

The project research methodology is discussed in previous chapter. In this section, the simulation results obtaining from the robotic arm model is shown and discussed. Both open loop and closed loop control system procedure are carried out using Simulink block diagrams in MATLAB.

The open loop simulation is first carried out to observe the open loop characteristics of the system. Closed-loop simulation is then carried out (with and without using controller) for different input angles. The angles being tested are  $15^\circ$ ,  $30^\circ$  and  $60^\circ$ . After that, PID controller and fuzzy logic controller are implemented to observe the changes in the system for the same batch of input angles. Figure 4.1 shows the structure of this chapter and the topics which are presented in this chapter.

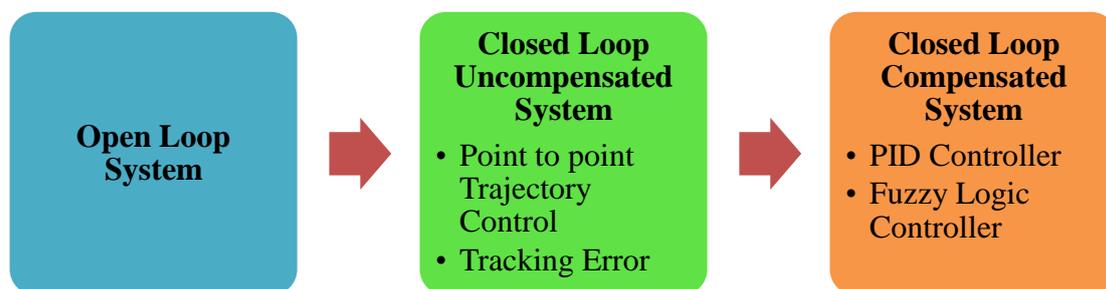


Figure 4.1: Structure of Chapter 4

## 4.2 Open Loop Simulation

In this section, the results of open loop simulation are shown. The objectives of carrying out the open loop simulation are:

1. To obtain the transfer function of motor
2. To observe the characteristics of open loop systems

When open loop simulations are carried out, several voltages (1V to 5V) are applied to the system. The purpose of varying the voltage is to observe the experimental value and simulation value. Then, the voltage value where the experimental value is closest to the simulation value is selected.

Table 4.1 shows the results of system identification for DC motor model when the input voltage is 1 Volt. From the experimental results, the second experimental values are closest to the mean value. Thus the transfer function of motor for 1V is:

$$G_1(s) = \frac{-0.06289s + 9.728}{s^2 + 41.03s - 1.505} \quad (4.1)$$

Figure 4.2 shows the input voltage and output angle of the open loop simulation with the substituted transfer function,  $G_1(s)$ . From the graph, it is observed that when 1 Volt is applied to the system, the maximum output angle achieved is about  $10^\circ$ . Apparently there are errors between the real time signal and simulated signal.

Table 4.2 shows the results of system identification for DC motor model when the input voltage is 2 volts. The experimental results show that the eighth experimental values are closest to the mean value. Thus the transfer function of the motor for 2V is:

$$G_2(s) = \frac{-0.7873s + 84.5}{s^2 + 318.7s + 10.78} \quad (4.2)$$

Figure 4.3 shows the graphs of input voltage and output angle for the open loop simulation with the transfer function  $G_2(s)$  being substituted into the system. It can be seen from the graphs that when 2 volts is applied to the system, the maximum output angle achieved is about  $26^\circ$ . The simulated signal is closely followed by the real time signal. However, the value of voltage still needs to be varied to validate whether which voltage value can give the best result.

Table 4.3 shows the results of system identification for DC motor model when the value of input voltage is set to 3 volts. The experimental results indicate that the ninth experimental values are closest to the mean value. Thus the transfer function of motor for 3V is:

$$G_3(s) = \frac{-1.221s + 204.3}{s^2 + 749.6s + 7.292} \quad (4.3)$$

Figure 4.4 shows the graphs of input voltage and output angle for 3V where the transfer function  $G_3(s)$  being substituted into the system. It can be seen from the graphs that when 3 volts is applied to the system, the maximum output angle achieved is about 44 °.

Table 4.4 shows the results of system identification for DC motor model when the input voltage is 4 volts. From the experimental results, it is observed that the fifth experimental values are closest to the mean value. Hence, the transfer function of the system is:

$$G_4(s) = \frac{-0.07351s + 15.82}{s^2 + 57.2s - 0.4846} \quad (4.4)$$

Figure 4.5 shows the graphs of input voltage and output angle when the transfer function  $G_4(s)$  is substituted into the current system to compare the experimental value and simulated value. When 4 volts is applied to the system, the maximum output angle achieved is about 60 °.

Table 4.5 shows the results of system identification for DC motor model. The experimental results indicate that the third experimental values are closest to the mean value. Hence, the transfer function of the system is:

$$G_5(s) = \frac{-0.06382s + 15.37}{s^2 + 53.39s + 0.3334} \quad (4.5)$$

Figure 4.6 shows the graphs of input voltage and output angle for 5V when the transfer function  $G_5(s)$  being substituted into the system. It shows that the maximum output angle achieved is about 75 ° when 5 volts is applied to the system.

Table 4.1: Results of system identification for DC motor model ( $V_{in} = 1V$ )

Repeatability	A	B	C	D	E
1	-0.05839	6.982	1	36.71	0.6297
2	-0.06289	8.728	1	41.03	-1.505
3	-0.03201	3.713	1	20.90	0.6472
4	-0.1128	8.626	1	68.08	-3.779
5	-0.3716	31.69	1	131.2	-1.151
6	-0.05986	6.991	1	31.26	1.407
7	-0.0429	4.088	1	28.02	-0.4906
8	-0.04318	3.702	1	24.15	0.1015
9	-0.05682	6.187	1	33.94	-1.828
10	0.02748	10.34	1	43.59	0.5742
Mean	-0.081297	9.1047	1	45.888	-0.5394
Std. Dev	0.107692653	8.252314356	0	32.75613286	1.552004531

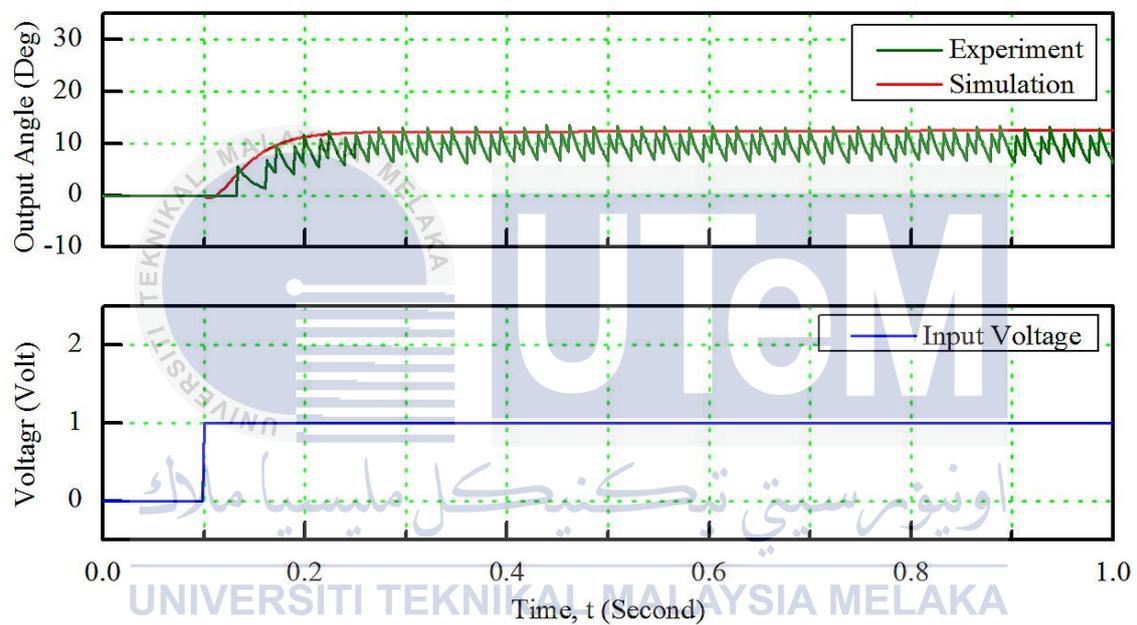


Figure 4.2: Graph of input voltage (1V) and output angle against time

Table 4.2: Results of system identification for DC motor model ( $V_{in} = 2V$ )

Repeatability	A	B	C	D	E
1	-1.577	116.2	1	495.4	-1.355
2	-0.324	46.51	1	228.8	-6.999
3	-0.348	43.39	1	214.8	-11.73
4	-1.215	92.12	1	424	-22.3
5	-0.7374	89.54	1	420.3	-8.424
6	-0.4993	58.54	1	224.2	7.143
7	-0.295	40.36	1	195.7	-8.845
8	-0.7873	84.5	1	318.7	10.78
9	-1.994	137.8	1	522.2	1.191
10	-2.032	132.5	1	563.5	-20.17
Mean	-0.9809	84.146	1	360.76	-6.0709
Std. Dev	0.680896	36.49585401	0	141.1141634	10.75909274

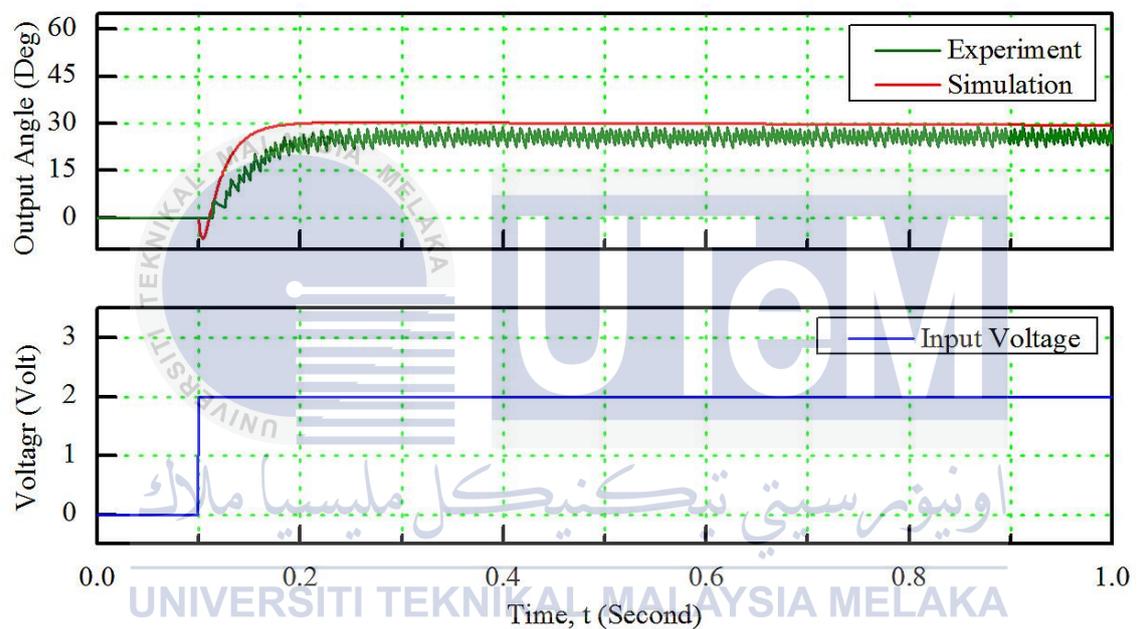


Figure 4.3: Graph of input voltage (2V) and output angle against time

Table 4.3: Results of system identification for DC motor model ( $V_{in} = 3V$ )

Repeatability	A	B	C	D	E
1	-0.8023	127.5	1	454.8	8.548
2	-0.7127	99.85	1	419.6	-17.58
3	-0.2971	41.52	1	146.4	3.905
4	-2.493	390.9	1	1522	-4.068
5	-2.064	299.1	1	1093	18.49
6	-2.587	383.5	1	1571	-46.15
7	-0.3698	55.52	1	215.7	-5.469
8	-0.2493	46.42	1	161.5	2.86
9	-1.221	204.3	1	749.6	7.292
10	-0.605	97.64	1	399.1	-14.31
Mean	-1.14012	174.625	1	673.27	-4.6482
Std. Dev	0.91071653	136.973839	0	542.0515495	18.15246003

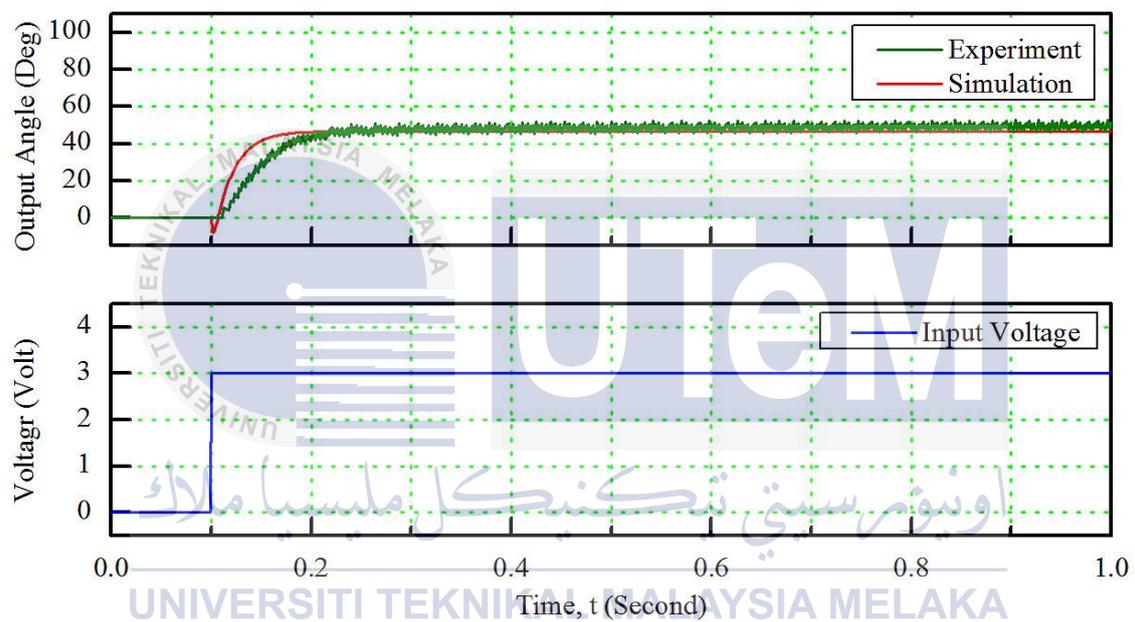


Figure 4.4: Graph of input voltage (3V) and output angle against time

Table 4.4: Results of system identification for DC motor model ( $V_{in} = 4V$ )

Repeatability	A	B	C	D	E
1	-0.06043	15.32	1	53.96	0.2958
2	-0.1019	19.51	1	73.35	-1.575
3	-0.1026	22	1	81.39	-1.09
4	-0.07097	17.68	1	60.92	0.951
5	-0.07351	15.82	1	57.2	-0.4846
6	-0.1222	23.02	1	87.72	-2.177
7	-0.03601	8.531	1	39.85	-1.233
8	-0.09833	23.96	1	85.57	0.6859
9	-0.06625	15.14	1	53.12	0.6077
10	-0.1006	19.23	1	73.6	-1.663
Mean	-0.08328	18.0211	1	66.668	-0.56822
Std. Dev	0.025937539	4.599387748	0	15.97512149	1.13194357

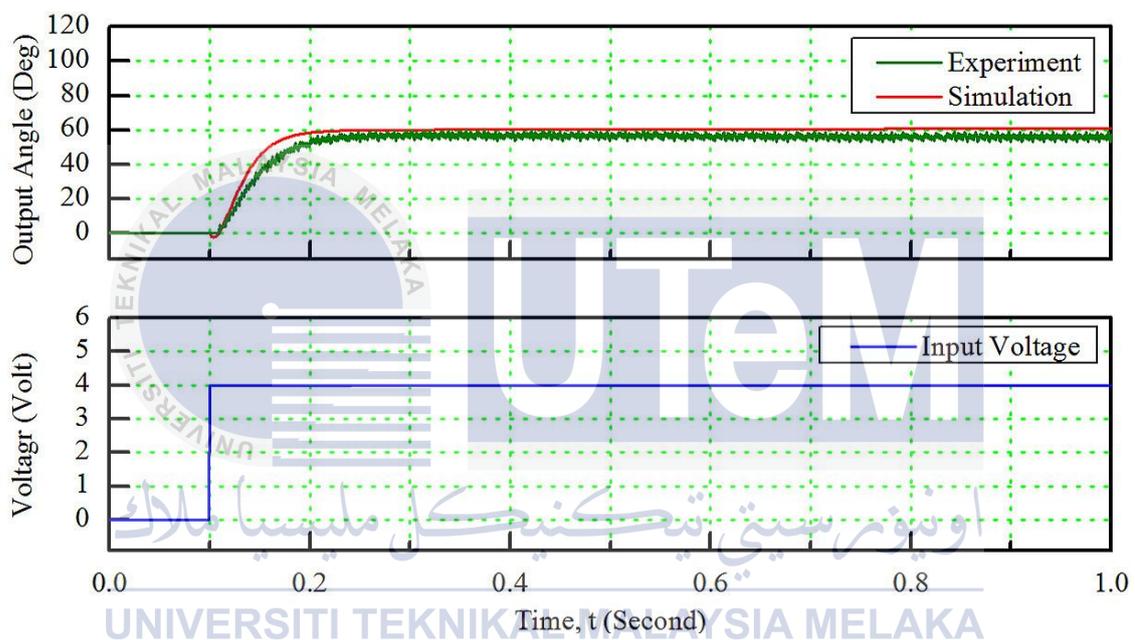


Figure 4.5: Graph of input voltage (4V) and output angle against time

Table 4.5: Results of system identification for DC motor model ( $V_{in} = 5V$ )

Repeatability	A	B	C	D	E
1	-0.08854	19.58	1	72.25	-1.037
2	-0.05626	14.14	1	51.85	-0.6075
3	-0.06382	15.37	1	53.39	0.3334
4	-0.07011	17.95	1	64.09	-0.3312
5	-0.08647	18.16	1	66.49	-0.9248
6	-0.05336	13.95	1	50.02	-0.2852
7	-0.05986	14.69	1	51.21	0.304
8	-0.07791	18.11	1	63.55	0.1302
9	-0.06802	17.23	1	62.21	-0.6139
10	-0.02745	12.72	1	45.32	-0.07472
Mean	-0.06518	16.19	1	58.038	-0.310672
Std. Dev	0.017846135	2.295454445	0	8.758388741	0.48631514

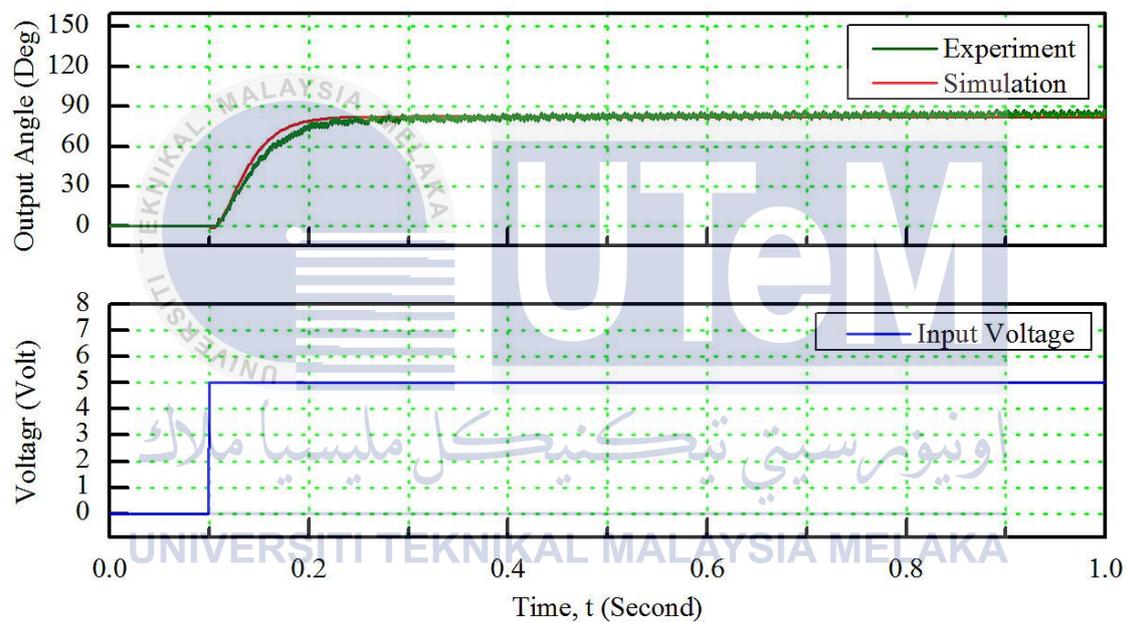


Figure 4.6: Graph of input voltage (5V) and output angle against time

#### 4.2.1 Transfer Function of Motor

Both the simulation of 4V and 5V give significant results. However, 5V is chosen as the reference input voltage as the results shown are of less noise or disturbances compared to results of 4V.

Equation 4.5 is chosen as the transfer function of motor. It is then substituted into the system and use for further analysis during closed-loop simulation. The Simulink block diagram with transfer function is as shown in Appendix G.

#### 4.2.2 Linearity of the System

To determine whether the system is linear or non-linear, the results of output angle (in degree) versus input voltage (in volt) are plotted. Table 4.6 shows the data obtained from open loop simulation. Figure 4.7 shows the graph of output angles against the input voltages. It is observed that the graph of output angle versus input voltage is linear. Thus the system is a linear system.

Table 4.6: Data obtained from open loop simulation

Input Voltage (V)	Output Angle (°)
1	10
2	26
3	44
4	60
5	75

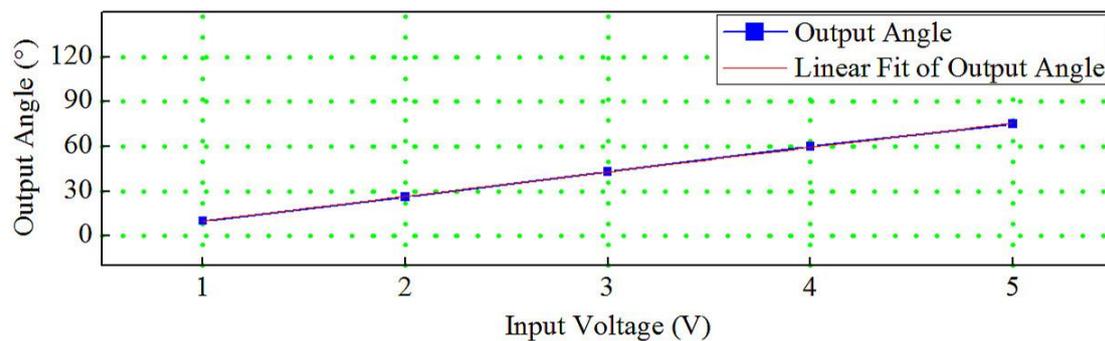


Figure 4.7: Graph of output angles against input voltages

#### 4.3 Uncompensated System

### 4.3.1 Point to Point Trajectory Control for Uncompensated System

Trajectory generation is the process of selecting a motion and the associated input controls to provide a complete and precise description of the robot motion using a suitable model of robots.

In this section, a closed-loop uncompensated system (without using controller) is designed and simulated. Step signal is given as the input signal and the output graph is observed to verify whether the output signal follows the selected input signal. Table 4.7 shows the parameters being fixed as well as being varied. Figure 4.7 and 4.8 shows the results of point to point trajectory control experiments for input angles of  $15^\circ$  and  $30^\circ$ .

Table 4.7: Parameters for point to point trajectory control experiments

Parameter	Numerical Value
Input Angle	$15^\circ, 30^\circ$
Simulation time	1s
Delay	0.1s
Sampling time	1ms
Input type	Step input
Controller	None

From Figure 4.8 and 4.9, it is observed that the experimental signal is stationary. This is because the robotic arm did not rotate due to large friction at the motor shaft. When the input angle is increased from  $15^\circ$  to  $30^\circ$ , the robotic arm is still in stationary position.

In short, the uncompensated system in this experiment does not produce satisfactory results as the output signals do not follow the input signals. There are large errors in both cases.

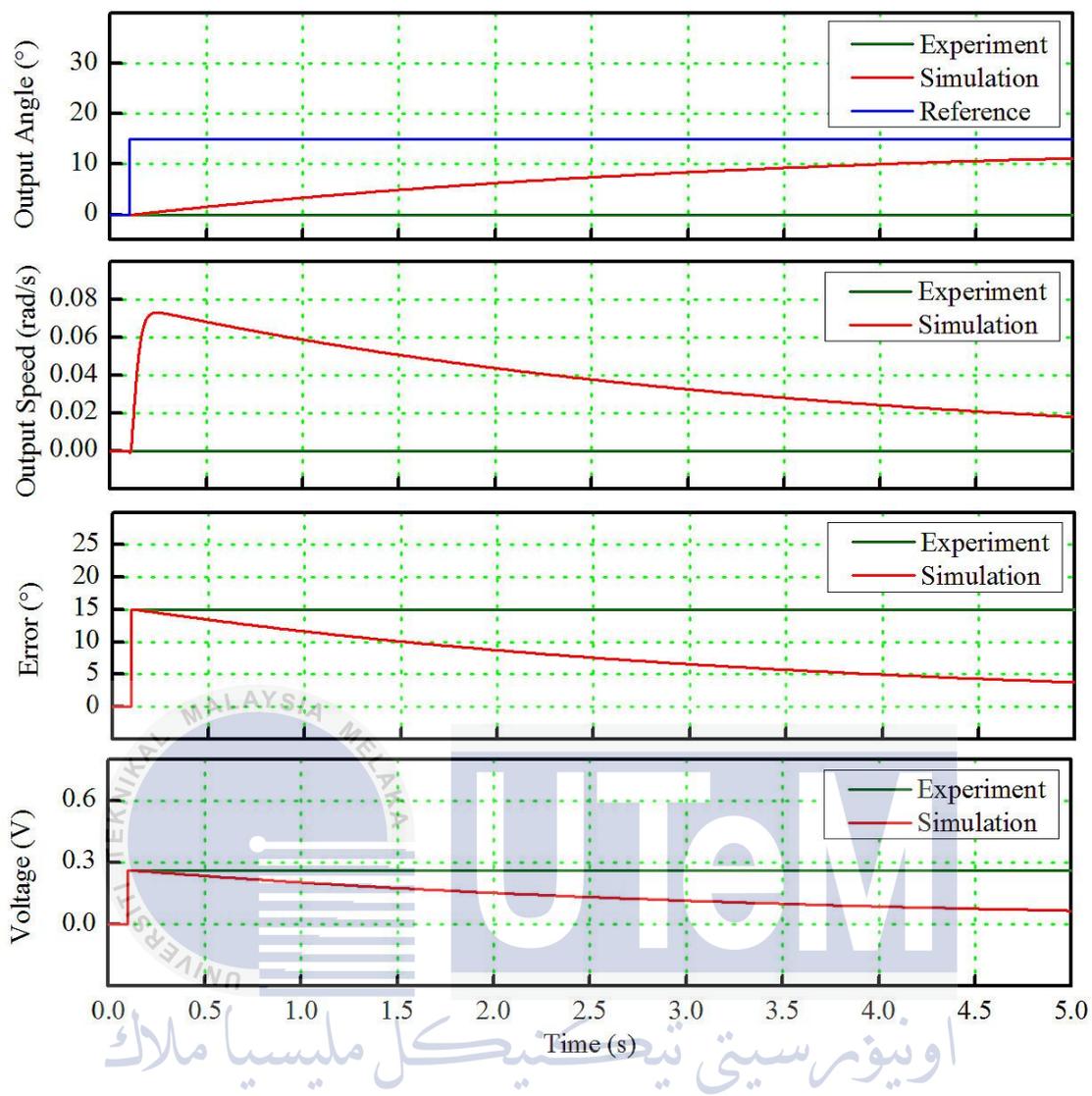


Figure 4.8: Results of point to point trajectory control for an uncompensated system with input angle of  $15^\circ$

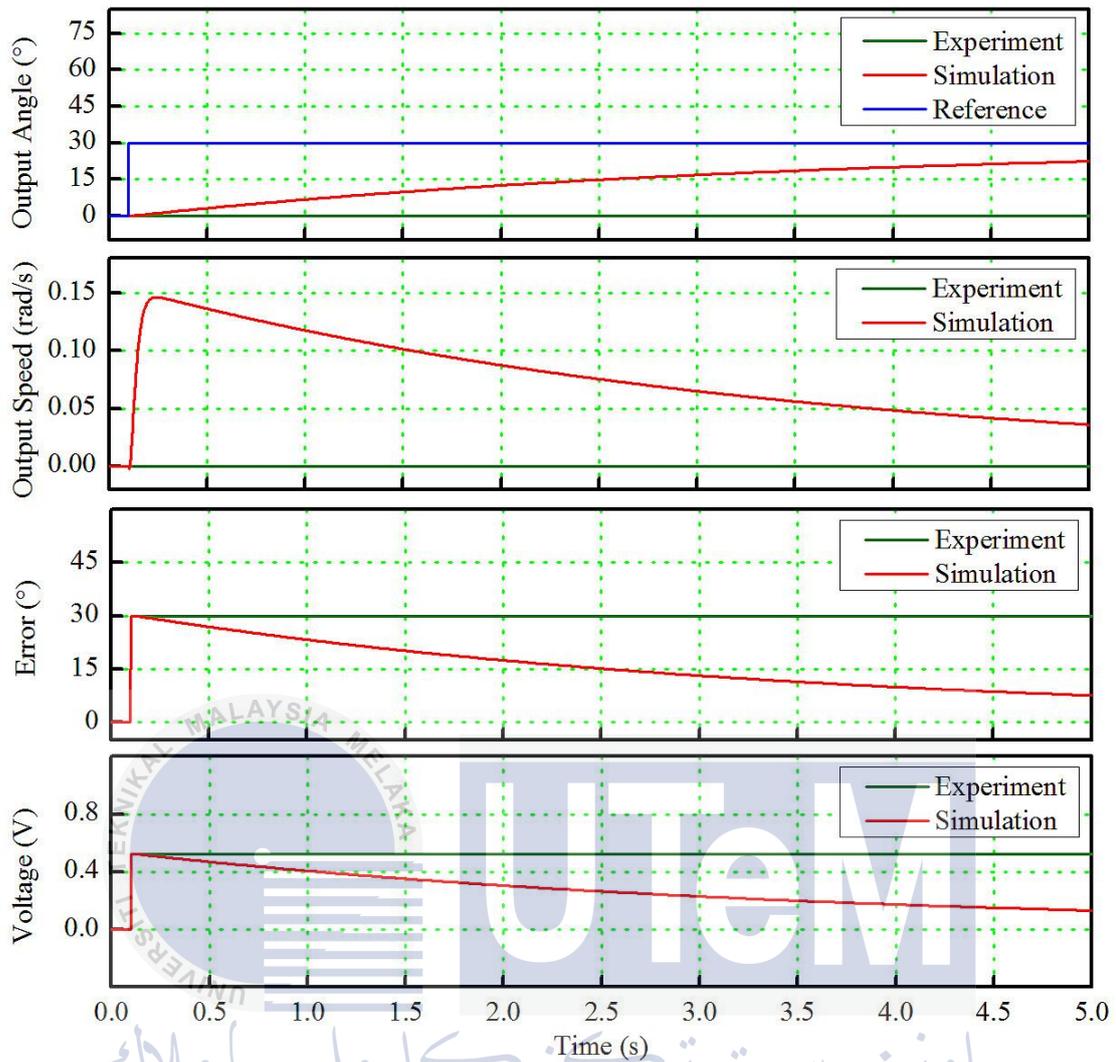


Figure 4.9: Results of point to point trajectory control for an uncompensated system with input angle of  $30^\circ$

### 4.3.2 Tracking Control for Uncompensated System

In this section, the tracking control experiments are carried out for different input angles. Sine wave is given as the input signal in each case. The frequency of the system is varied to observe the effects of different frequencies to the output signals. Table 4.8 shows the parameters being fixed and varied for the tracking control experiments. Figure 4.10 indicates the results of tracking error experiment when the input angle is  $15^\circ$ .

Table 4.8: Parameters for tracking error experiments

Parameter	Numerical Value
Input Angle	$15^\circ, 30^\circ$
Simulation time	5s
Sampling time	1ms
Input type	Sine wave
Controller	None

From the results of simulation for uncompensated system for both experiments, it is summarized that there are large steady-state errors occurred in each cases, in which the output angles are not proportional with the input angles. Hence, a controller is needed to improve the transient response of the system.

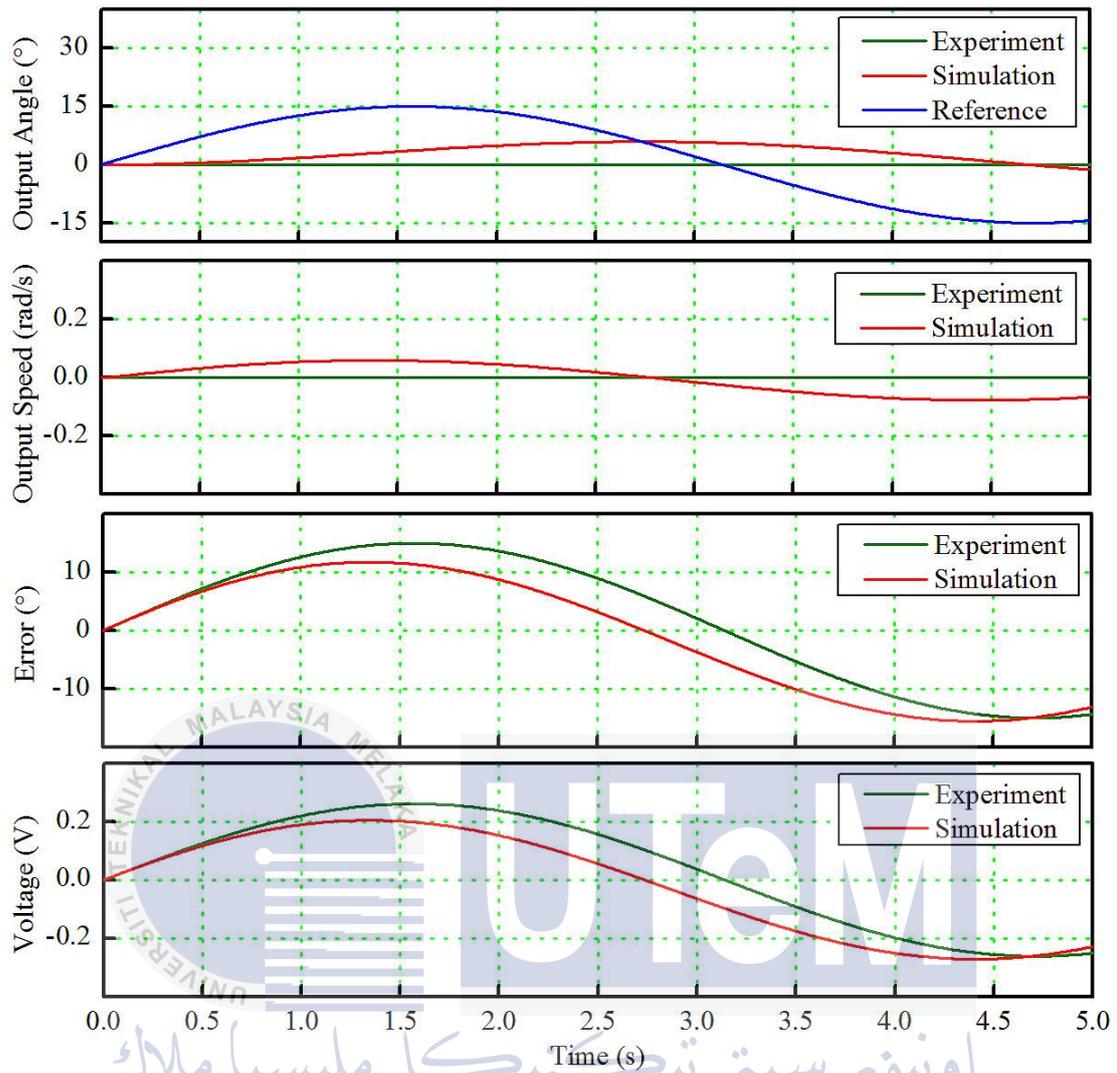


Figure 4.10: Results of tracking error experiment for an uncompensated system with input angle of  $15^\circ$

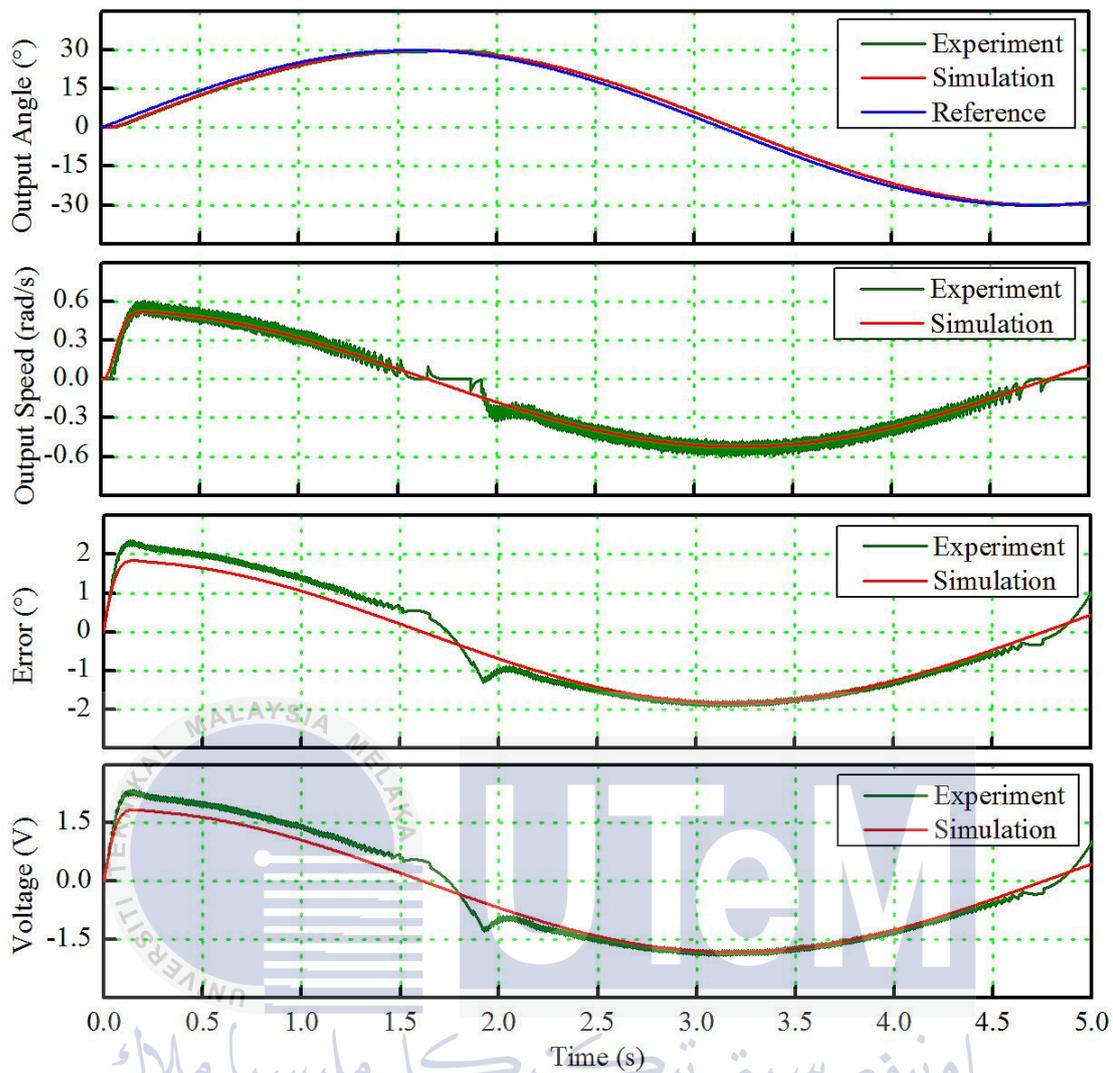


Figure 4.11: Graph of steady-state error against time for an uncompensated system with input angle of  $30^\circ$

#### 4.4 Compensated System with PID Controller

In this section, the system is implemented with PID controller. Two types of experiments are carried out, namely point to point trajectory control and error tracking experiments. This is to test the capability of the robot manipulator as well as to control its motion precisely.

##### 4.4.1 Point to Point Trajectory Control with PID Controller

To determine suitable  $K_p$  value, simulation is run with  $K_i = 0$  and  $K_d = 0$ . The gain is increased slowly until the system starts to oscillate. Table 4.9 shows the parameters being fixed and also varied in this compensated system with PID controller. Figure 4.11 shows the variation of  $K_p$  value by means of manual tuning where the input angle is  $15^\circ$ .

Table 4.9: Parameters for point to point experiments using PID controller

Parameter	Numerical Value
Input Angle	$15^\circ, 30^\circ, 60^\circ$
Simulation time	1s
Sampling time	1ms
Input type	Step input
Controller	PID controller

Figure 4.12 to 4.15 shows the results of simulation with a constant input angle of  $15^\circ$  and varying  $K_p$  values. From Figure 4.13, it is observed that when the  $K_p$  value is increased to 10, the system starts to oscillate. The value of  $K_p$  is increased continuously until 14 as shown in Figure 4.14. The oscillation of system increased too.

Finally when the  $K_p$  value is increased to 14.6, the system reached complete oscillation. The gain value during this condition is named as ultimate gain,  $K_u$  as discussed previously in Chapter 3, whereas the period of oscillation is named as  $T_u$ . Then, the  $K_p$  value is calculated from the  $K_u$  value and the values of  $K_i$  and  $K_d$  are calculated based on Equation 3.25 in Chapter 3.

Figure 4.15 to 4.18 shows the results of simulation with a constant input angle of  $30^\circ$  and varying  $K_p$  values. From Figure 4.15, it is observed that when the system starts to

oscillate. The value of  $K_p$  is increased continuously until 14 as shown in Figure 4.17. The oscillation of system increased too. When the  $K_p$  value is increased to 14.6 as in Figure 4.18, the system reached complete oscillation.

Figure 4.19 to 4.22 shows the results of simulation with a constant input angle of  $60^\circ$  and varying  $K_p$  values. The results are similar to the previous experiments. When the  $K_p$  value is increased to 14.6 as in Figure 4.22, the system reached complete oscillation.



$K_p = 1, K_i = 0, K_d = 0$

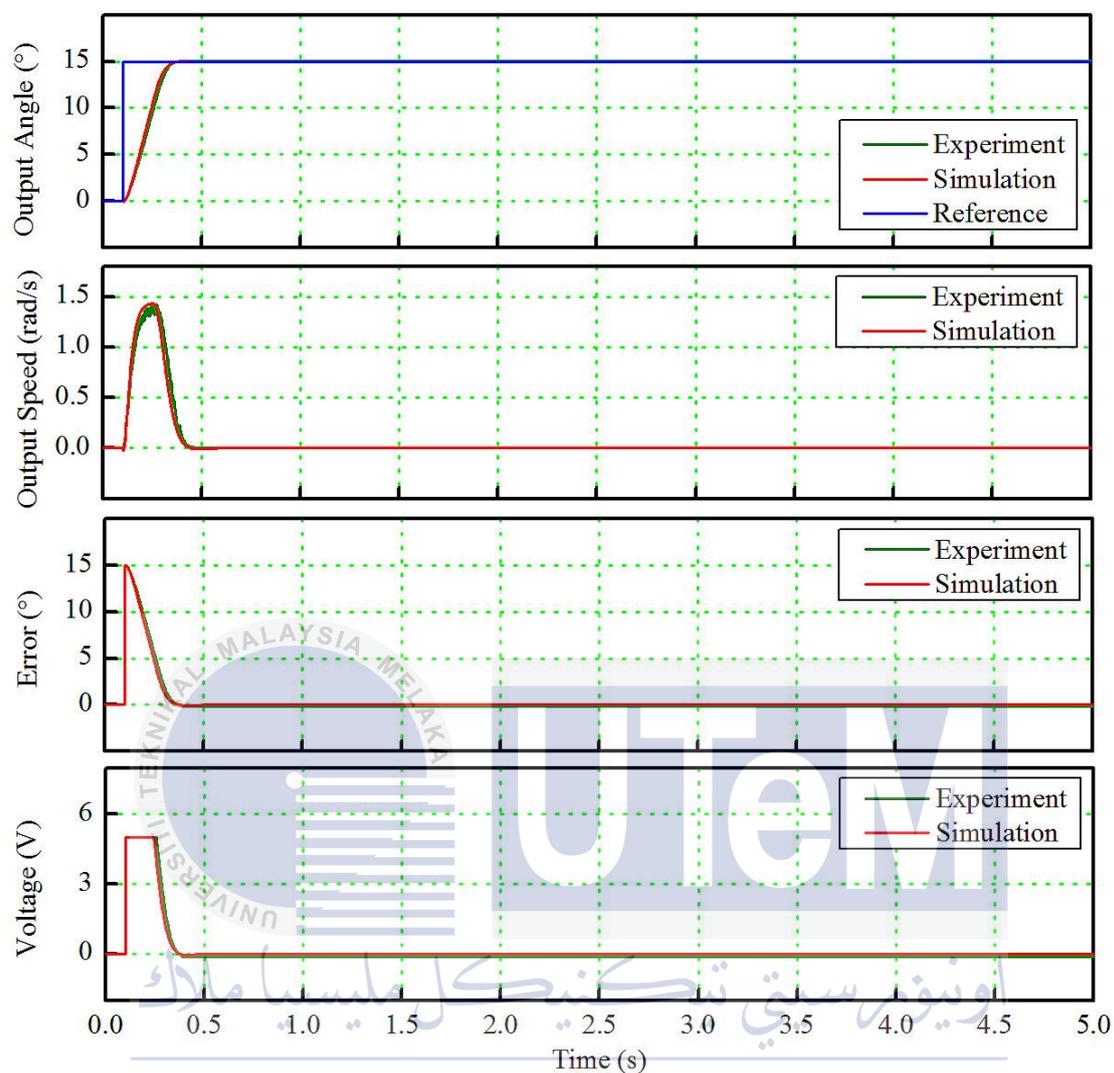


Figure 4.12: Results of point to point trajectory control experiment for a PID control system with input angle of  $15^\circ$  and  $K_p$  value of 1

$K_p = 10, K_i = 0, K_d = 0$

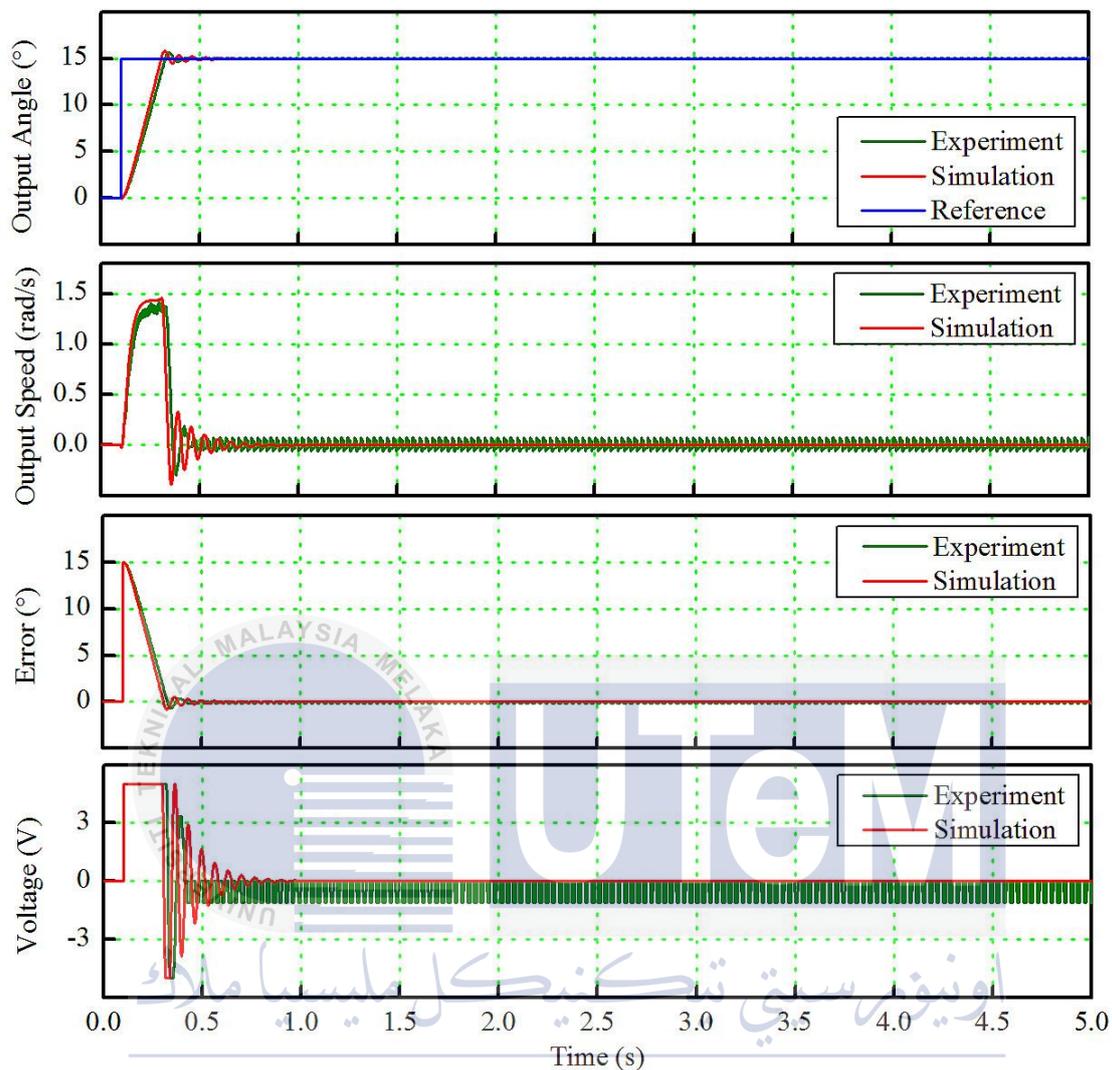


Figure 4.13: Results of point to point trajectory control experiment for a PID control system with input angle of  $15^\circ$  and  $K_p$  value of 10

$K_p = 14, K_i = 0, K_d = 0$

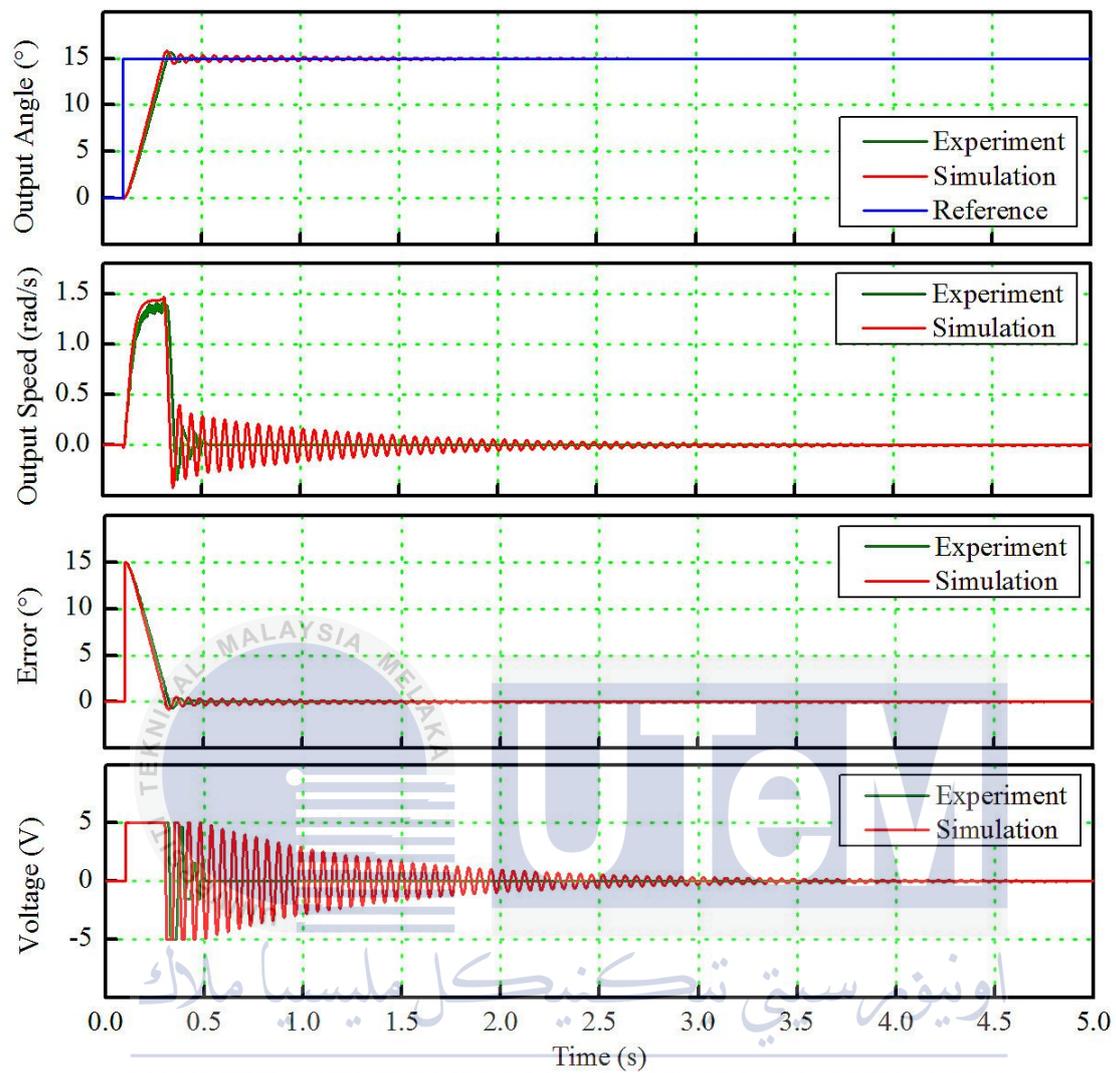


Figure 4.14: Results of point to point trajectory control experiment for a PID control system with input angle of  $15^\circ$  and  $K_p$  value of 14

$K_p = 14.6, K_i = 0, K_d = 0$

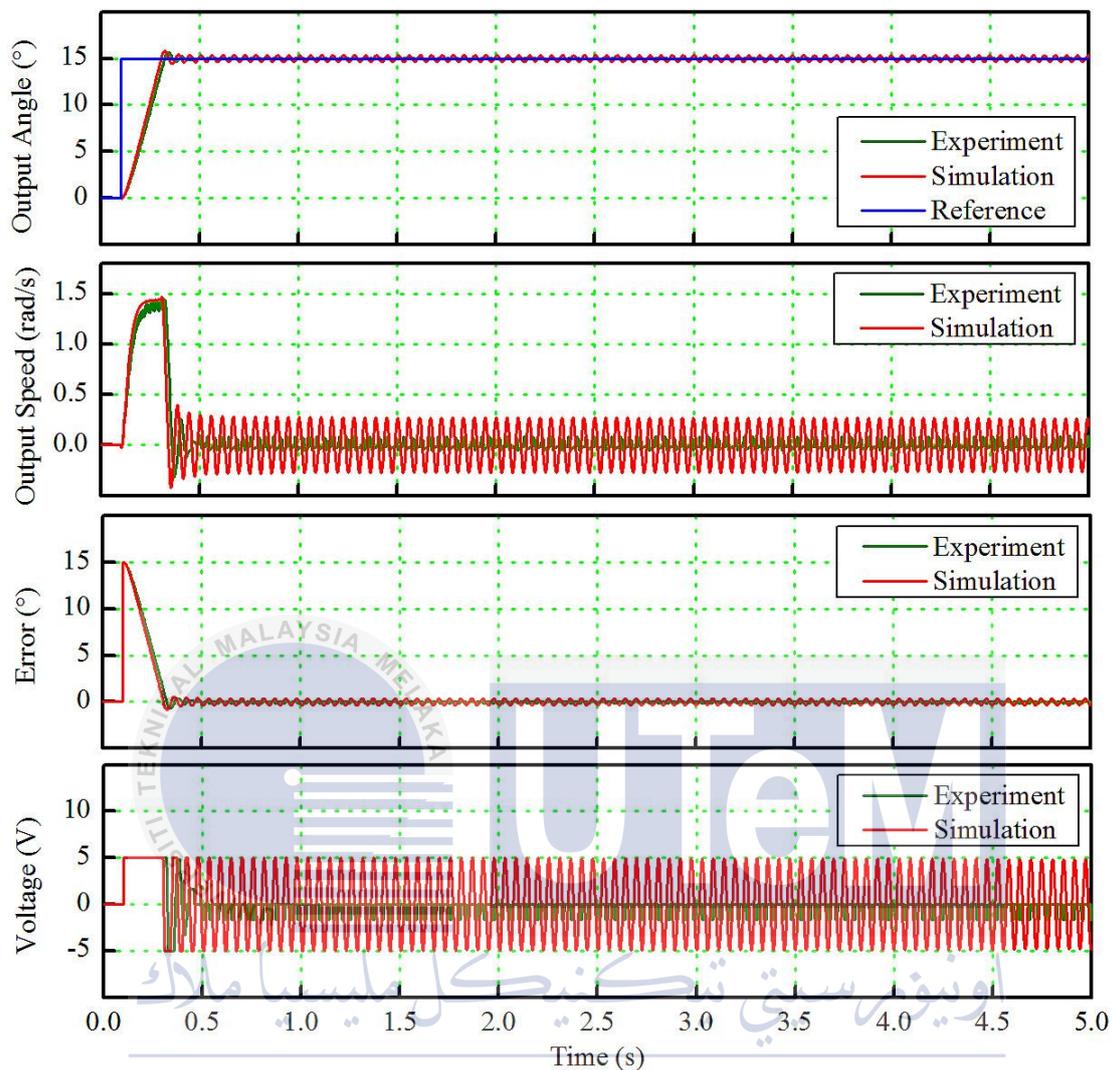


Figure 4.15: Results of point to point trajectory control experiment for a PID control system with input angle of  $15^\circ$  and  $K_p$  value of 14.6

$K_p = 1, K_i = 0, K_d = 0$

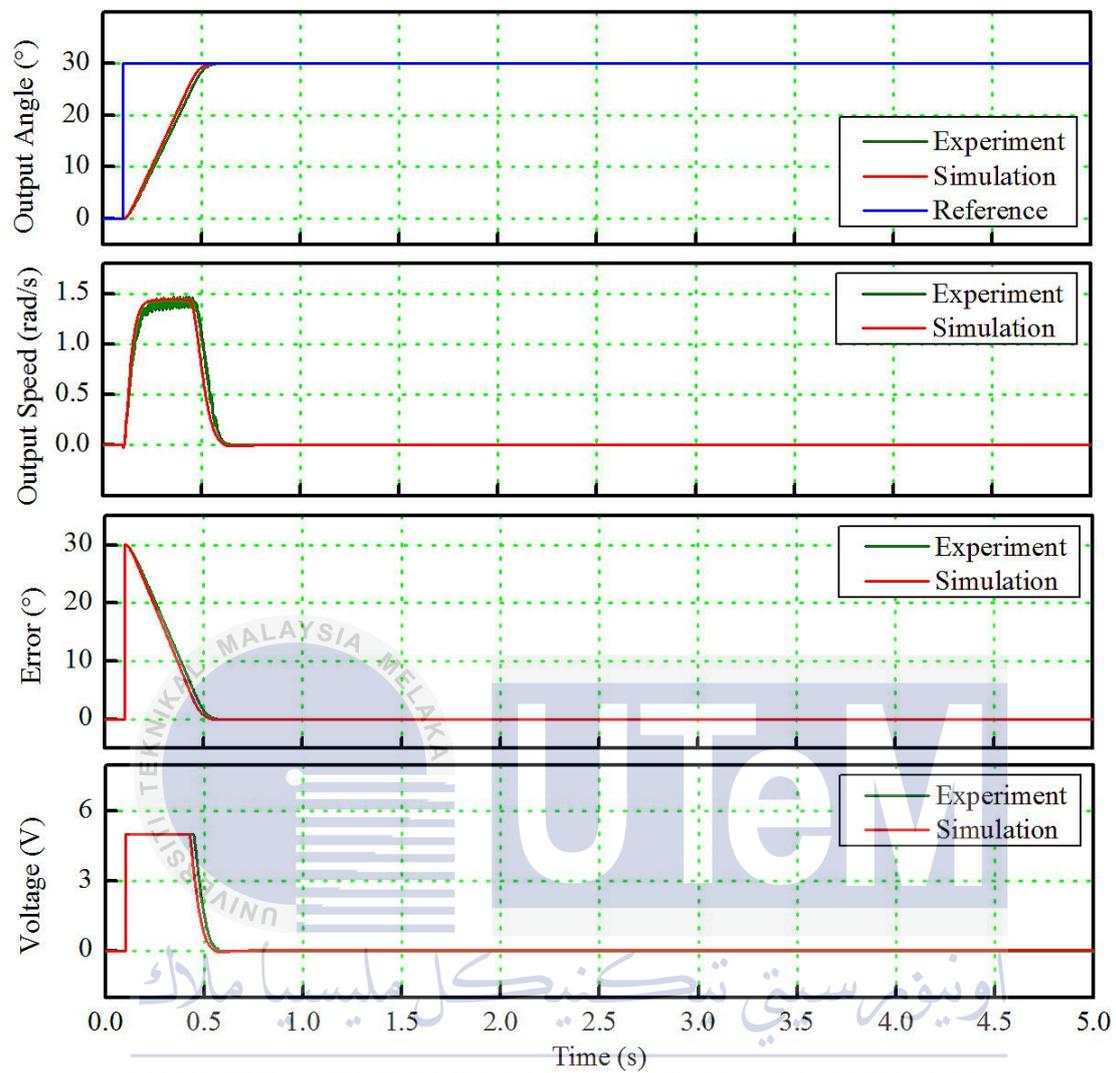


Figure 4.16: Results of point to point trajectory control experiment for a PID control system with input angle of 30° and  $K_p$  value of 1

$K_p = 10, K_i = 0, K_d = 0$

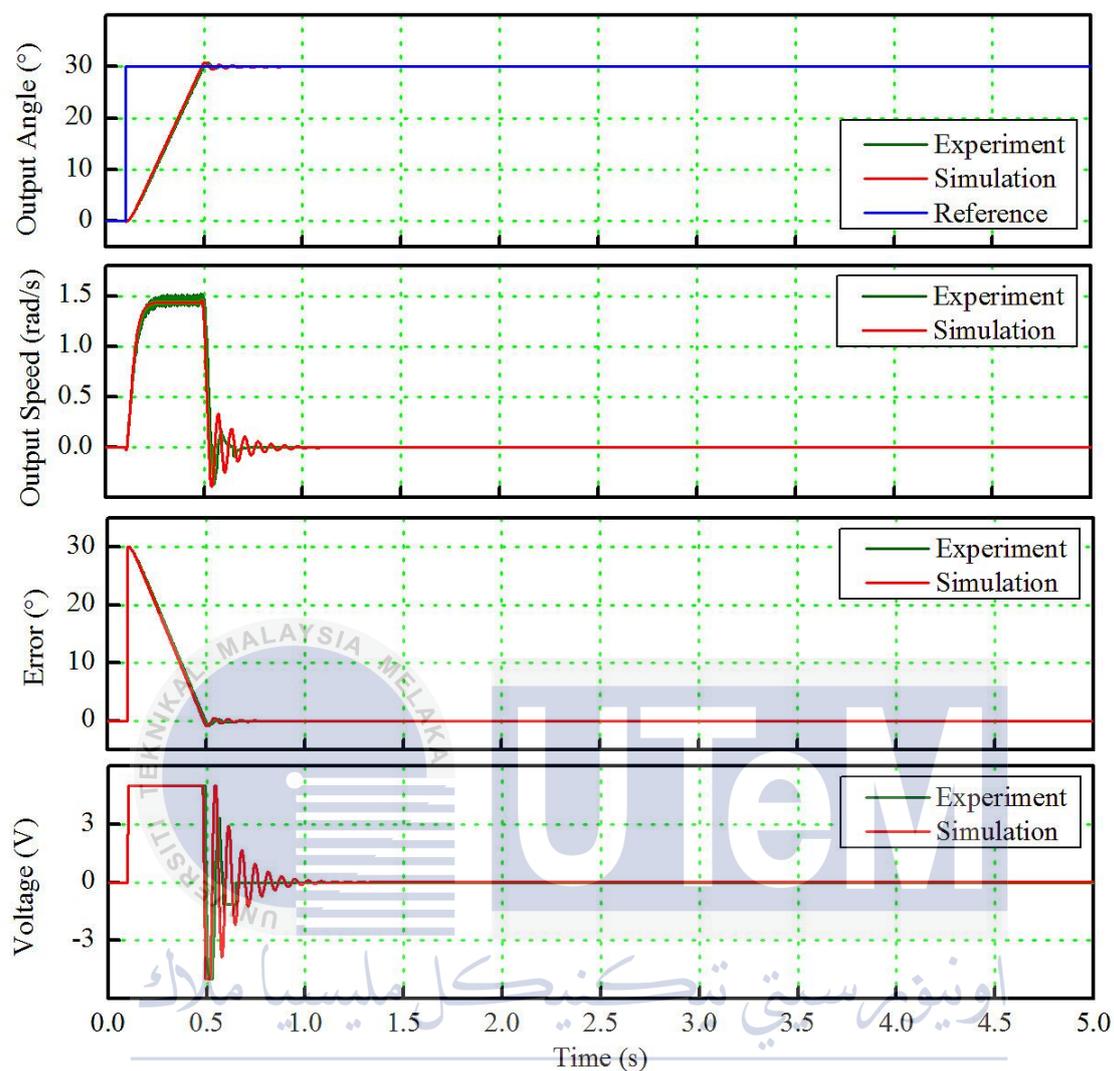


Figure 4.17: Results of point to point trajectory control experiment for a PID control system with input angle of  $30^\circ$  and  $K_p$  value of 10

$K_p = 14, K_i = 0, K_d = 0$

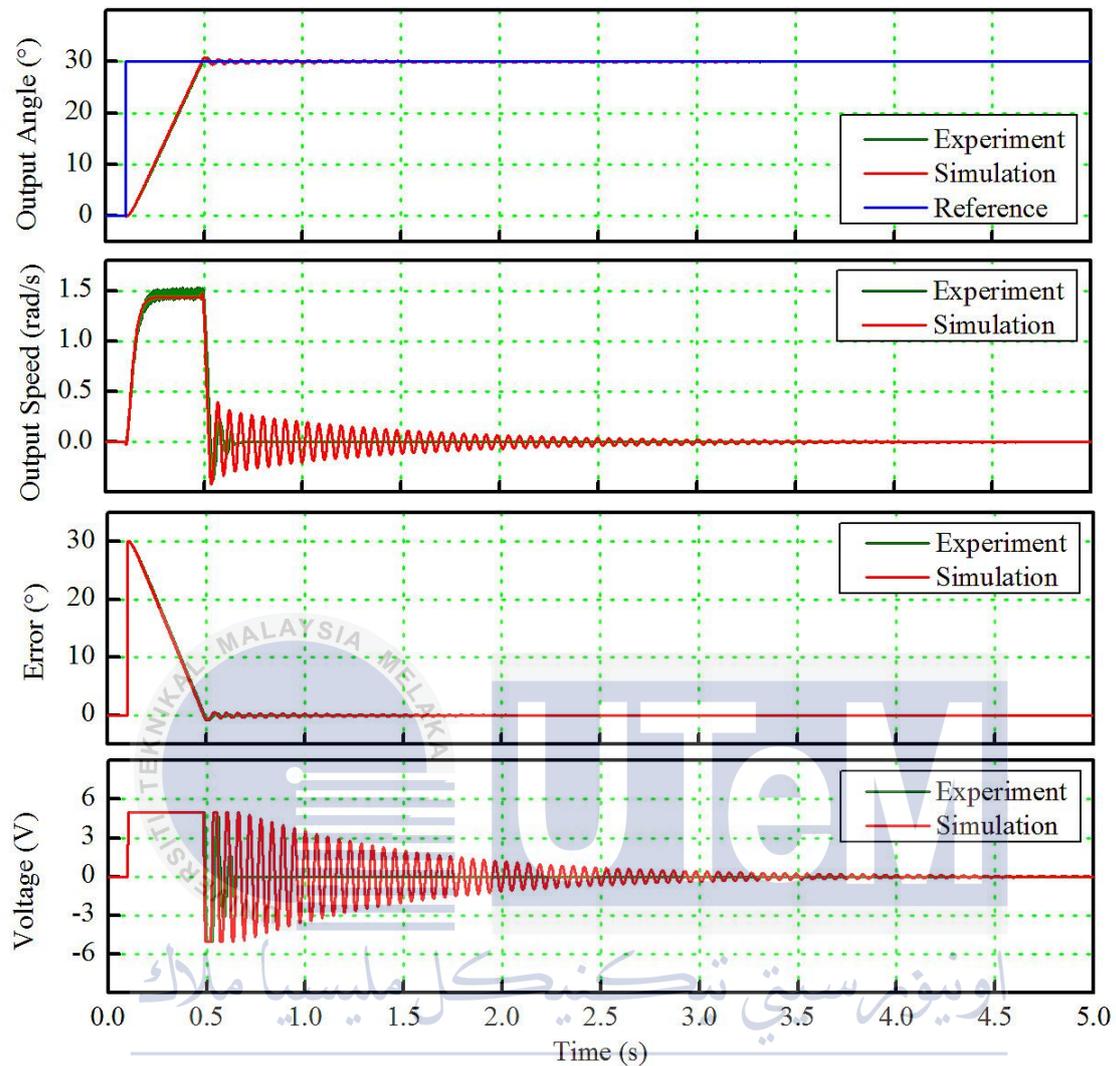


Figure 4.18: Results of point to point trajectory control experiment for a PID control system with input angle of  $30^\circ$  and  $K_p$  value of 14

$K_p = 14.6, K_i = 0, K_d = 0$

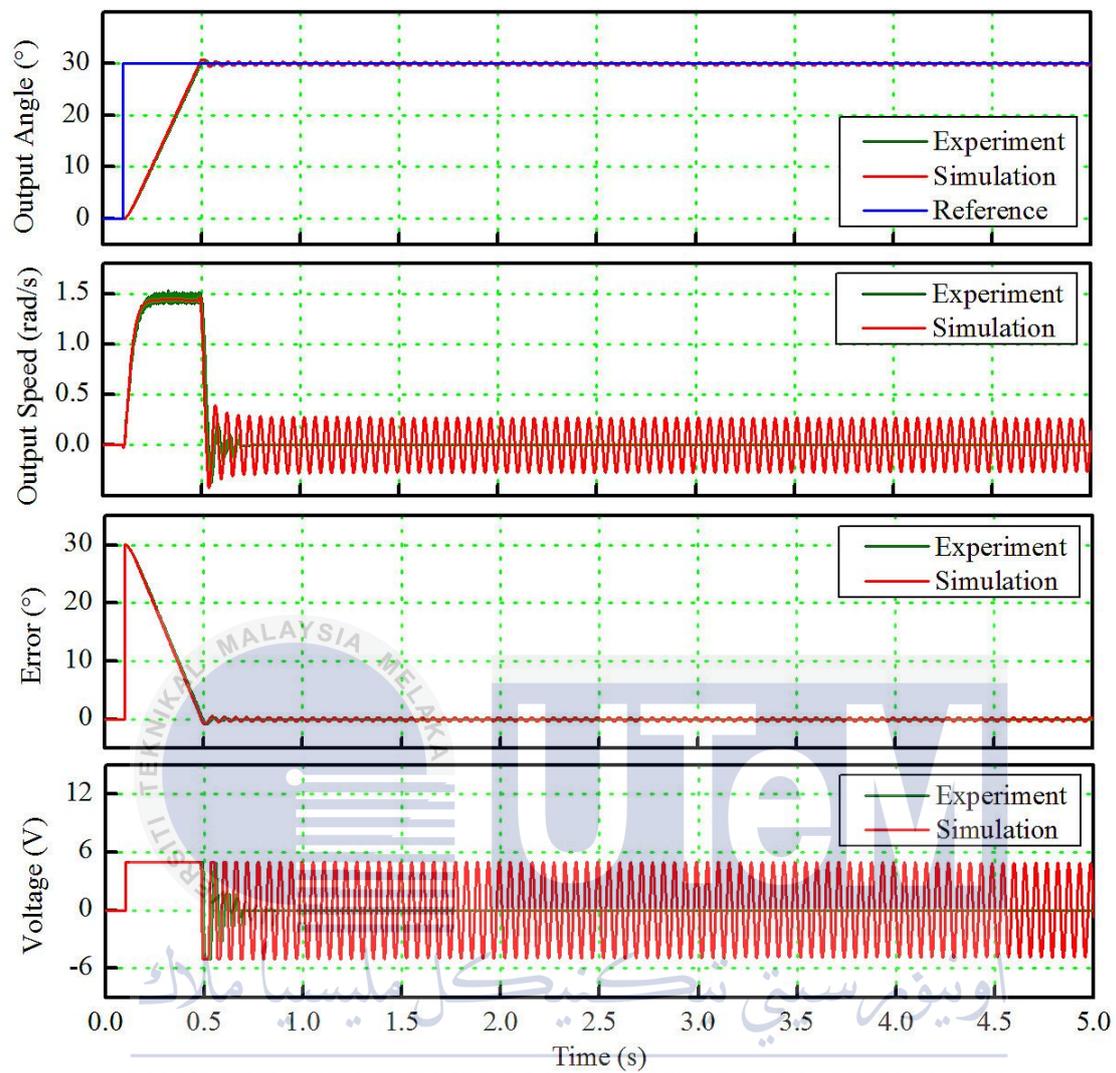


Figure 4.19: Results of point to point trajectory control experiment for a PID control system with input angle of  $30^\circ$  and  $K_p$  value of 14.6

$K_p = 1, K_i = 0, K_d = 0$

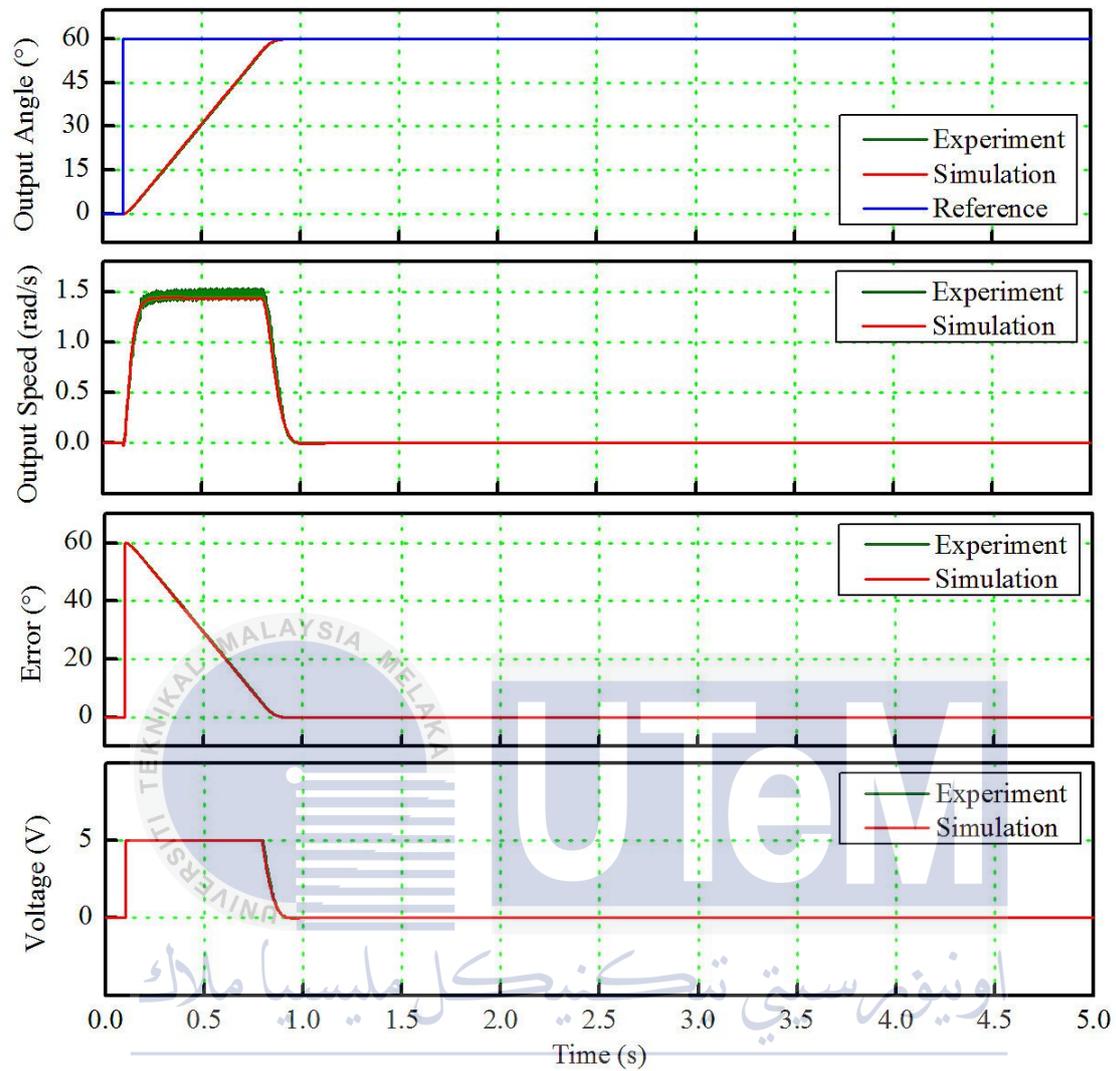


Figure 4.20: Results of point to point trajectory control experiment for a PID control system with input angle of  $60^\circ$  and  $K_p$  value of 1

$K_p = 10, K_i = 0, K_d = 0$

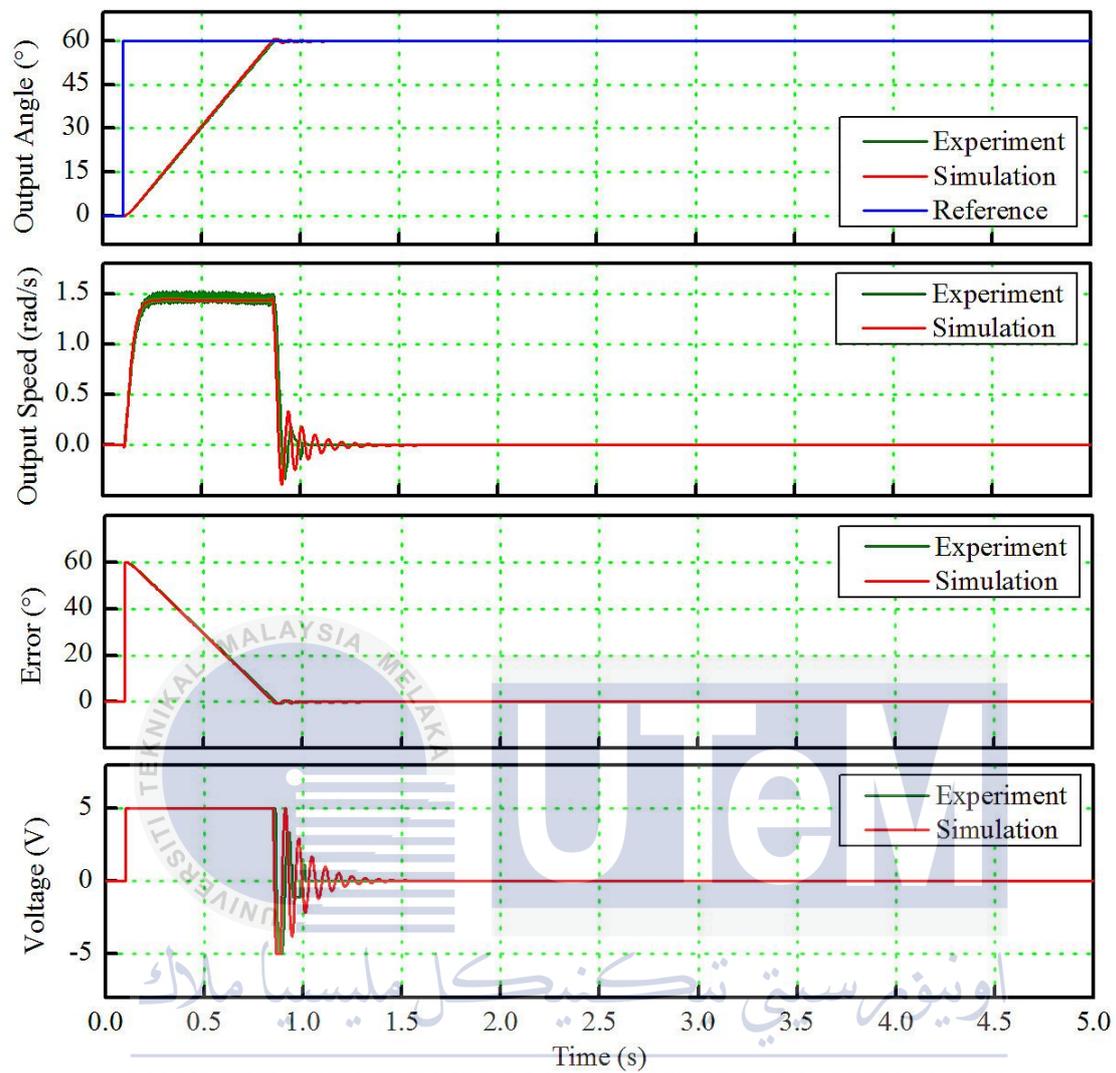


Figure 4.21: Results of point to point trajectory control experiment for a PID control system with input angle of  $60^\circ$  and  $K_p$  value of 10

$K_p = 14, K_i = 0, K_d = 0$

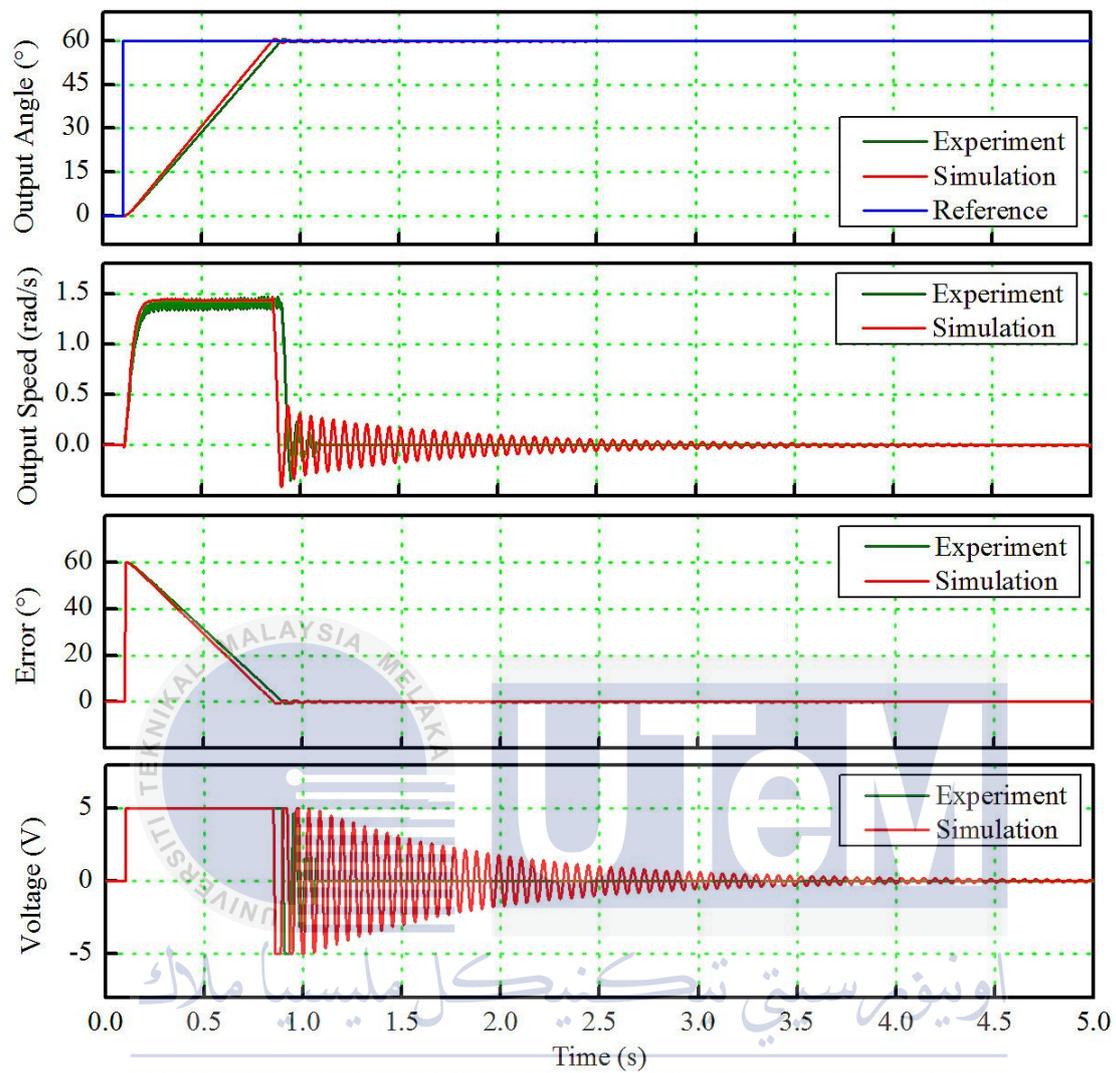


Figure 4.22: Results of point to point trajectory control experiment for a PID control system with input angle of  $60^\circ$  and  $K_p$  value of 14

$K_p = 14.6, K_i = 0, K_d = 0$

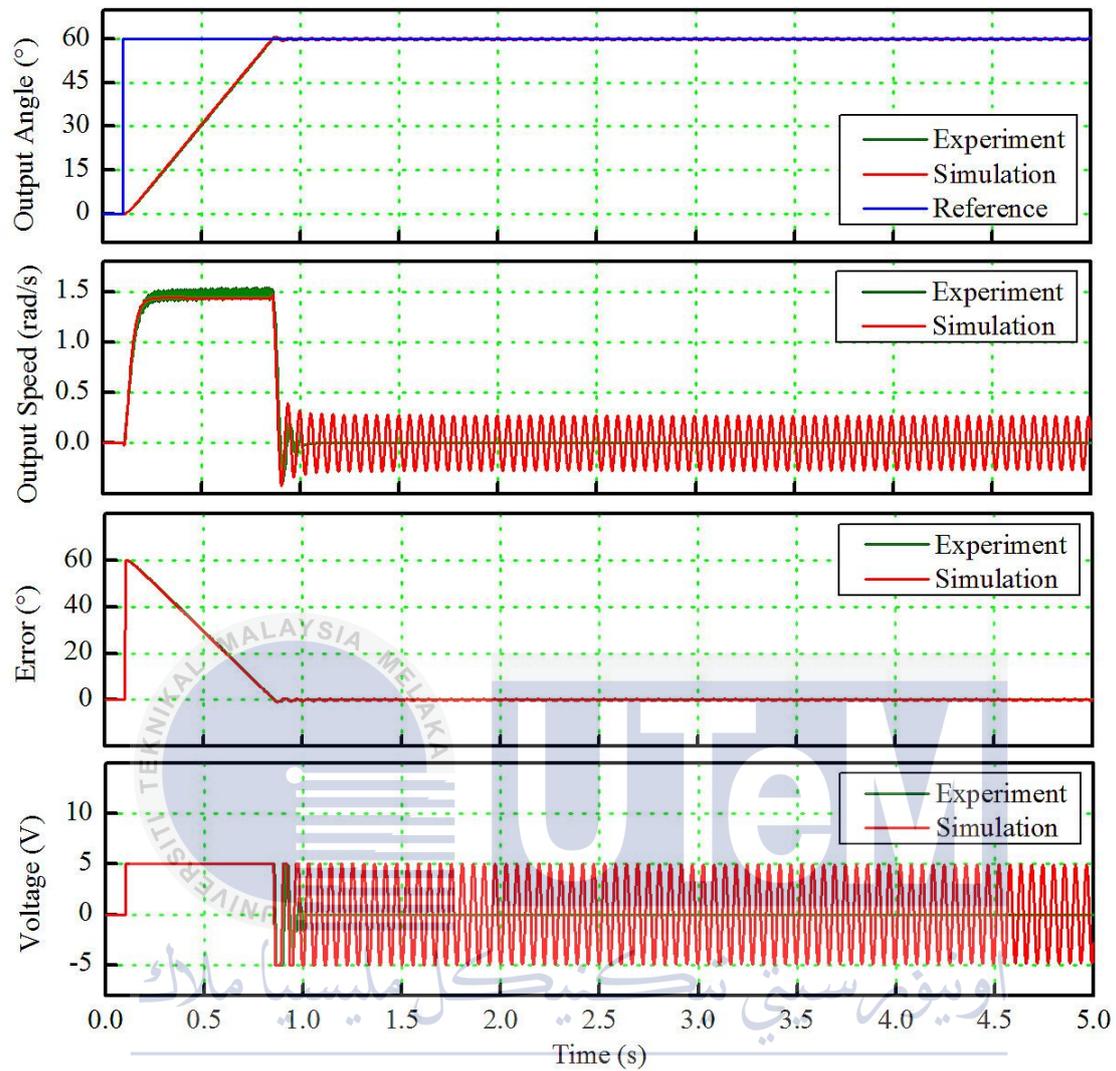


Figure 4.23: Results of point to point trajectory control experiment for a PID control system with input angle of  $60^\circ$  and  $K_p$  value of 14.6

Since the results for three types of input angles are identical from each other, the parameters of PID controller can be calculated.

Ultimate gain,  $K_u = 14.6$

From Table 3.10: Proportional gain value,  $K_p = 0.6T_u = (0.6)(14.6) = 8.76$

From the graph: Integral time,  $T_i = 0.852 - 0.796 = 0.056s$

From Table 3.10:  $T_i = 0.5T_u = (0.5)(0.056) = 0.028$

From Equation 3.25: Integral gain,  $K_i = \frac{K_p}{T_i} = \frac{8.76}{0.056} = 156.4$

From Table 3.10:  $T_d = 0.12T_u = (0.12)(0.056) = 0.00672 = 6.72 \times 10^{-3} s$

From Equation 3.25: Derivative gain,  $K_d = T_d K_p = (0.00672)(8.76) = 0.06$

Thus from the results of simulation of input angle  $15^\circ$ , the following data of PID controller can be obtained as shown in Table 4.10.

Table 4.10: Parameters of PID controller obtained from simulation results

Parameter	Symbol	Numerical Value
Proportional gain	$K_p$	8.76
Integral gain	$K_i$	156.4
Derivative gain	$K_d$	0.06

The calculation is just an approximation based on the experiments being carried out. To validate the reliability of the results, the  $K_p$  value is fixed on the next experiment and the  $K_i$  value is varied to reduce the steady-state error of the system.

Figure 4.24 and 4.25 shows the results of tuning  $K_i$  values. From Figure 4.24, it can be observed that when  $K_i$  value is 1, there are some vibrations of the movement of robotic arm. The vibration is not necessary. Thus, the value of  $K_i$  is kept constant at zero.

$K_p = 8.76, K_i = 0, K_d = 0$

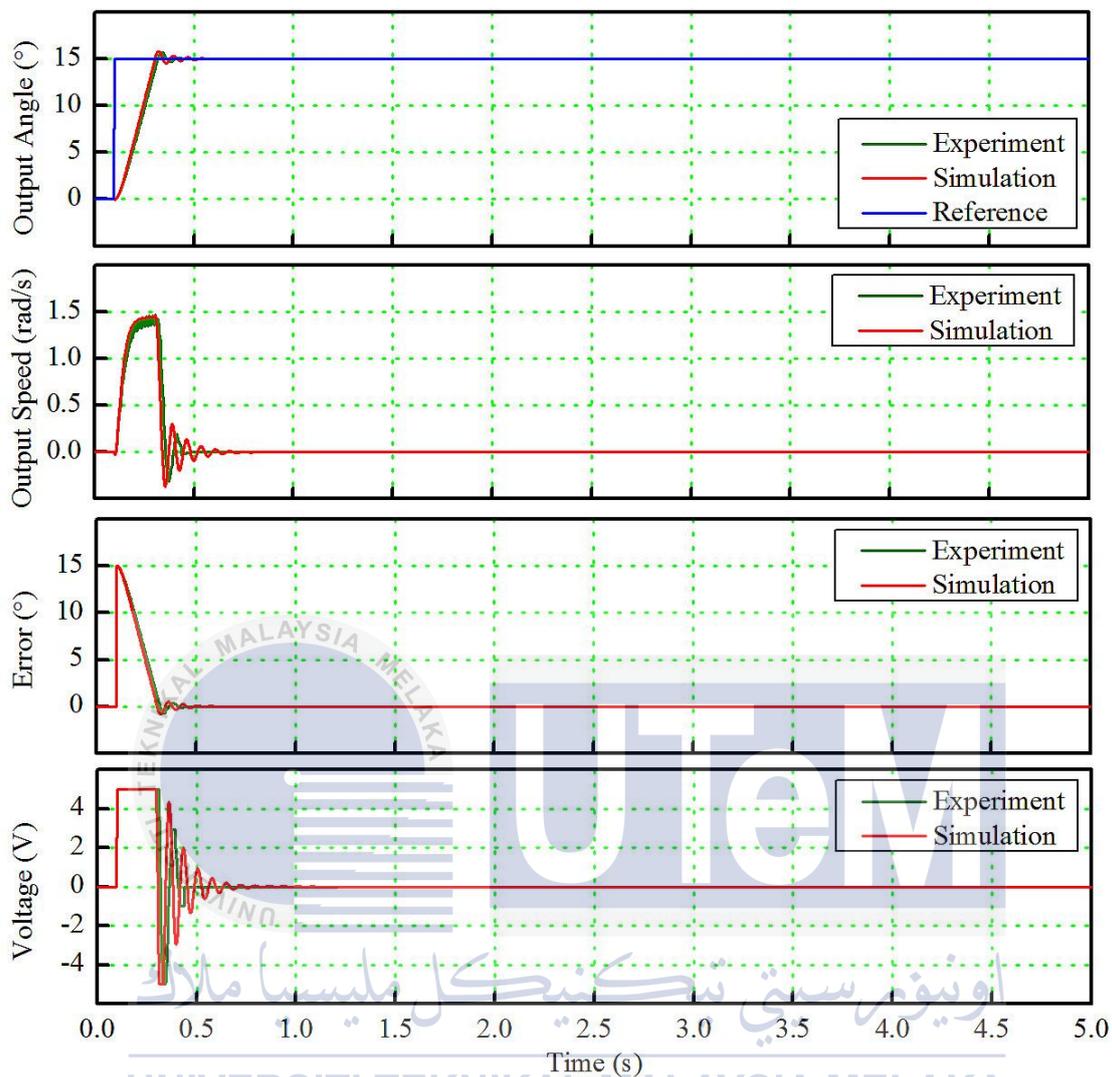


Figure 4.24: Results of point to point trajectory control experiment for a PID control system with input angle of  $15^\circ$  and  $K_p$  value of 8.76

$K_p = 8.76, K_i = 1, K_d = 0$

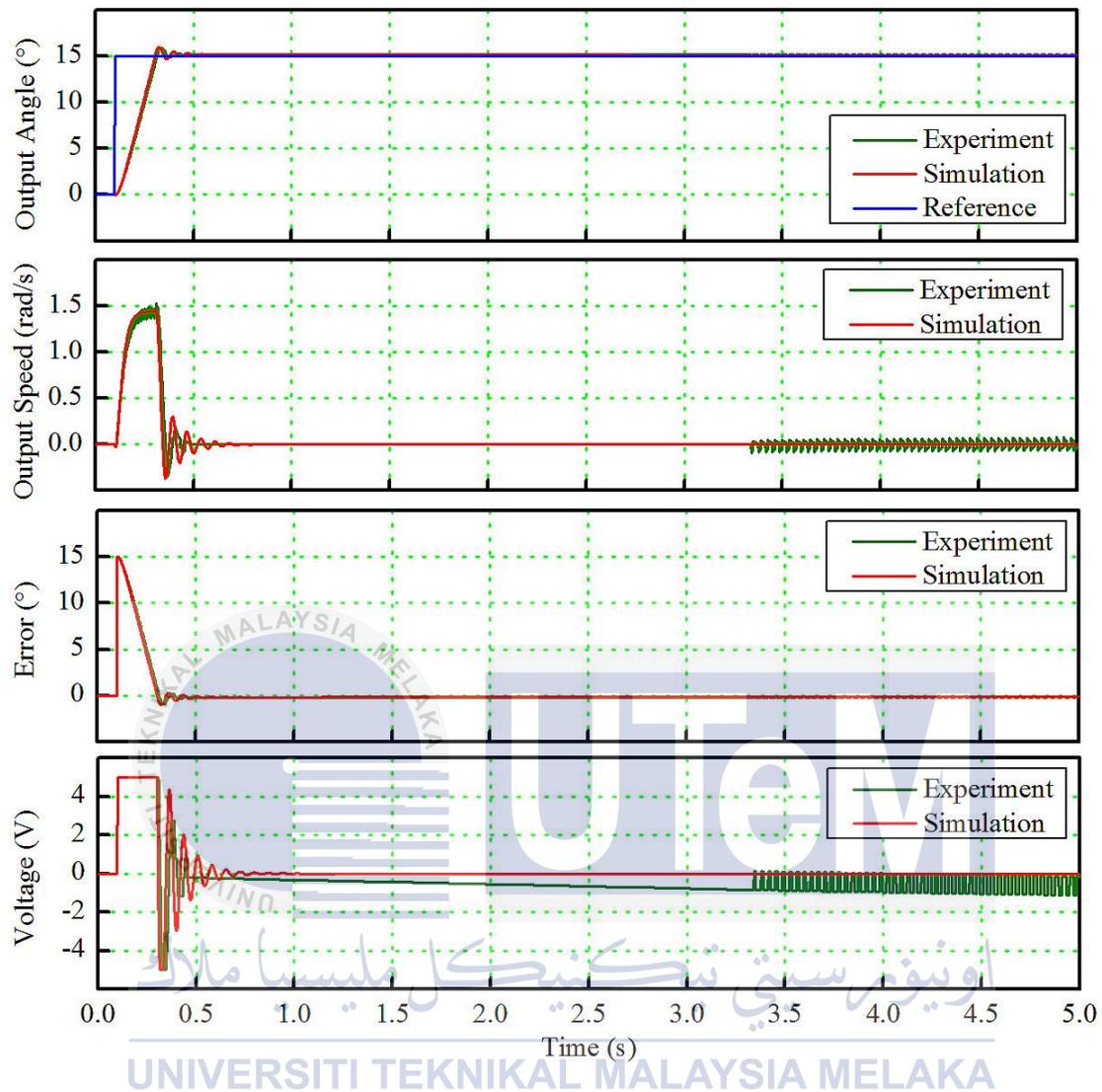


Figure 4.25: Results of point to point trajectory control experiment for a PID control system with input angle of  $15^\circ$  and  $K_p$  value of 8.76

#### 4.4.2 Tracking Control with PID Controller

In this section, the tracking error experiments are carried out for different input angles. Sine wave is given as the input signal in each case. The frequency of the system is varied to observe the effects of different frequencies to the output signals. Table 4.11 shows the parameters being fixed and varied for the tracking error experiments. Figure 4.26 to 4.28 indicates the results of steady-state error when the input angle is  $15^\circ$ ,  $30^\circ$  and  $60^\circ$ .

Table 4.11: Parameters for tracking error experiments for PID-controlled system

Parameter	Numerical Value
Input Angle	$15^\circ$ , $30^\circ$ , $60^\circ$
Simulation time	5s
Sampling time	1ms
Input type	Sine wave
Controller	PID controller

From the results of tracking control experiments, it is observed that the results are satisfactory as the experimental value of steady-state error is almost reduced to zero no matter how the value of input angle is changed.

Figure 4.29 to 4.31 shows the comparison in terms of output angles between the signals of different frequencies. The frequencies of the systems are varied to observe the effect of changing in frequency to the output angles of the motor. It can be concluded that the output angle increases when the frequency of the system is increased. However, when the frequency is very high (For example, 50Hz), the output signals are unstable.

Figure 4.32 to 4.34 shows the comparison in terms of steady-state error between the signals of different frequencies. It is observed that when the frequency of the system increased, the steady-state error also increased as the system started to become unstable.

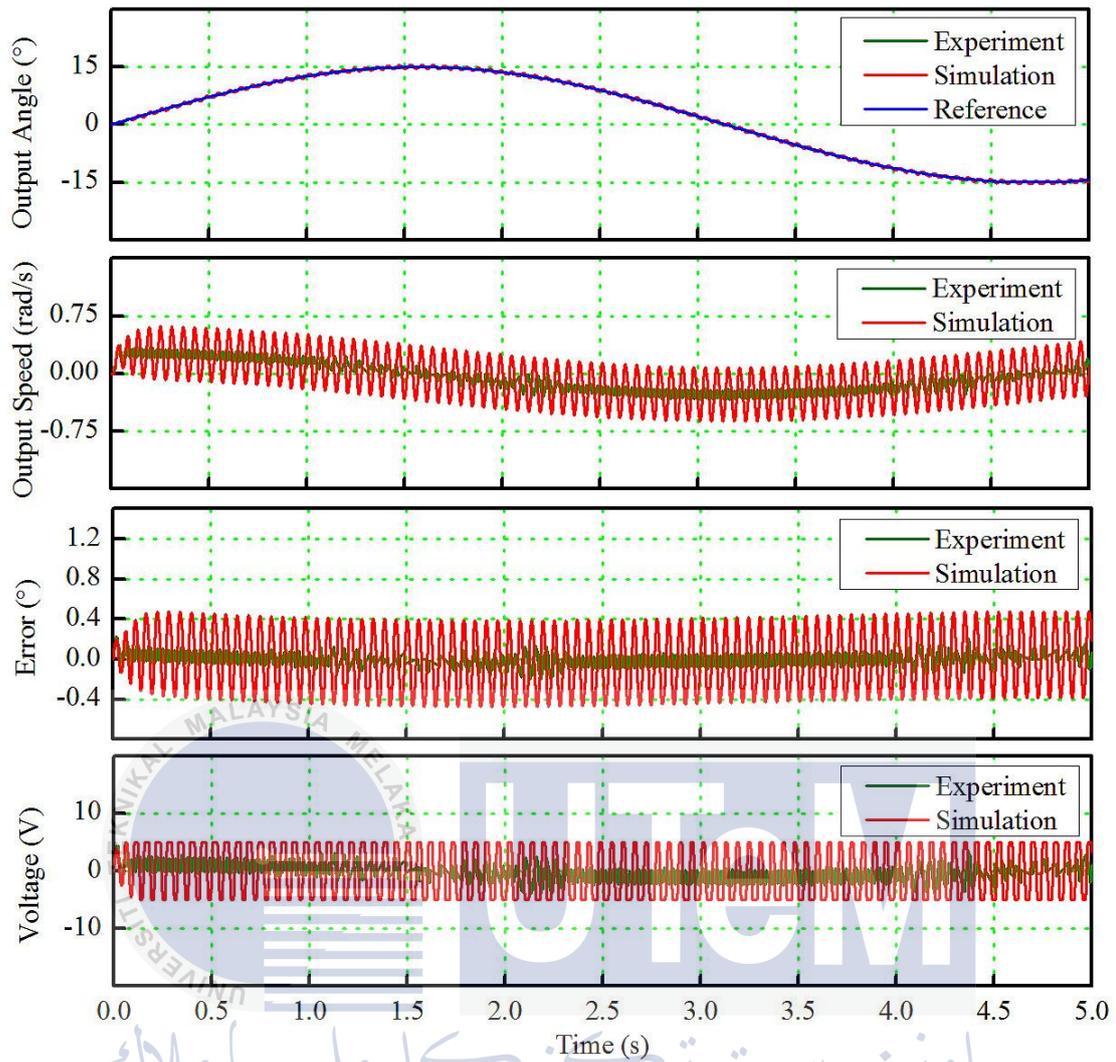


Figure 4.26: Results of tracking error experiment for a PID control system with input angle of  $15^\circ$

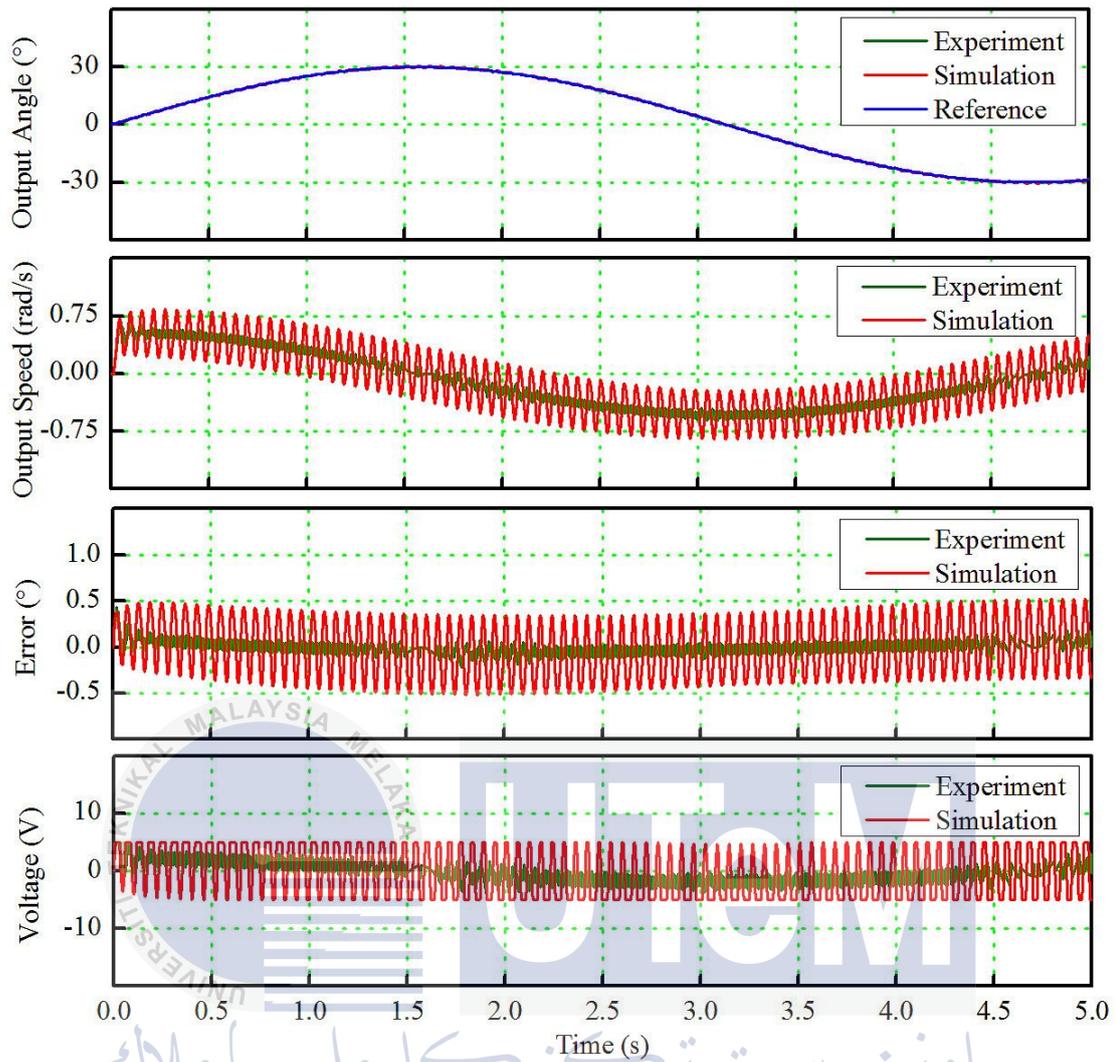


Figure 4.27: Results of tracking error experiment for a PID control system with input angle of  $30^\circ$

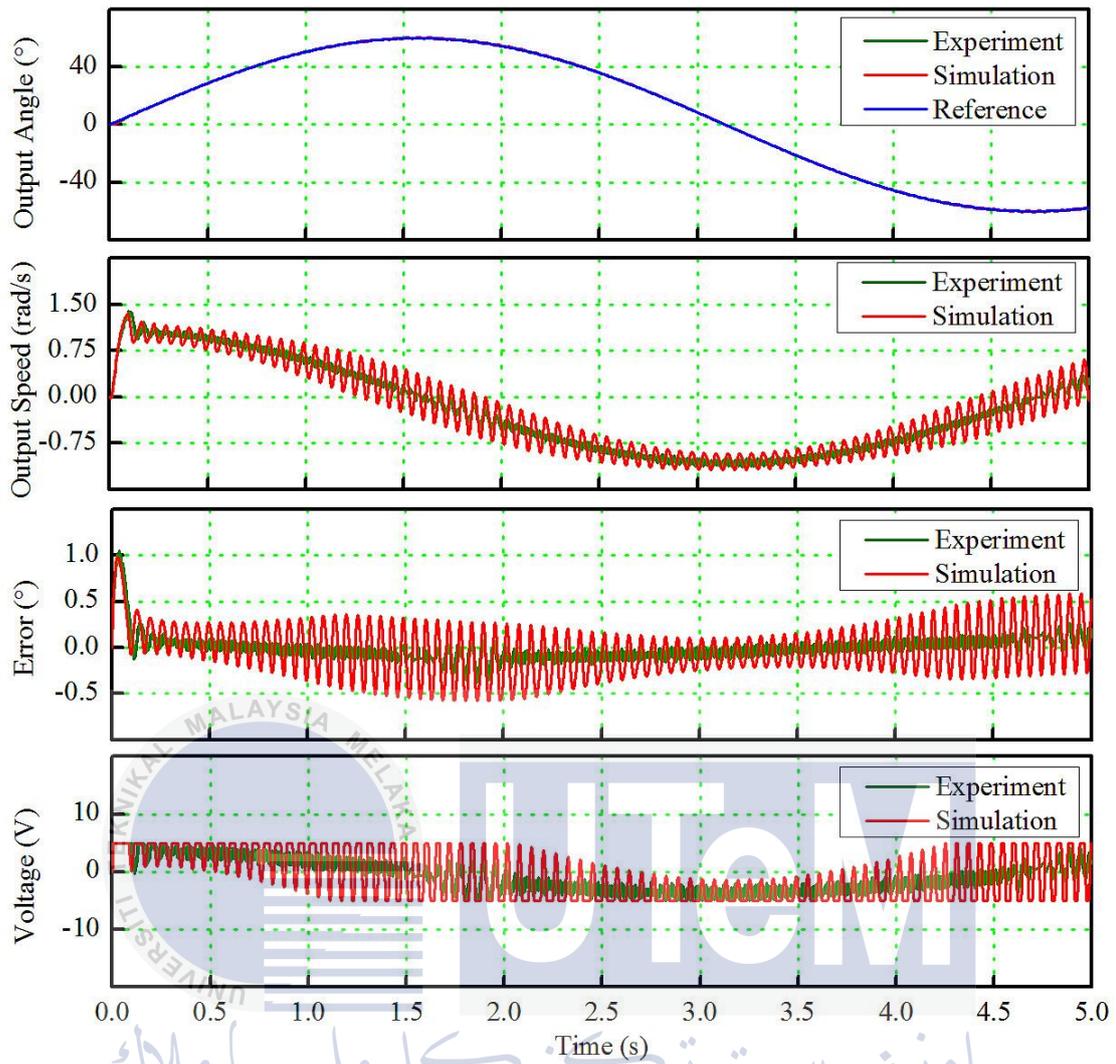


Figure 4.28: Results of tracking error experiment for a PID control system with input angle of  $60^\circ$

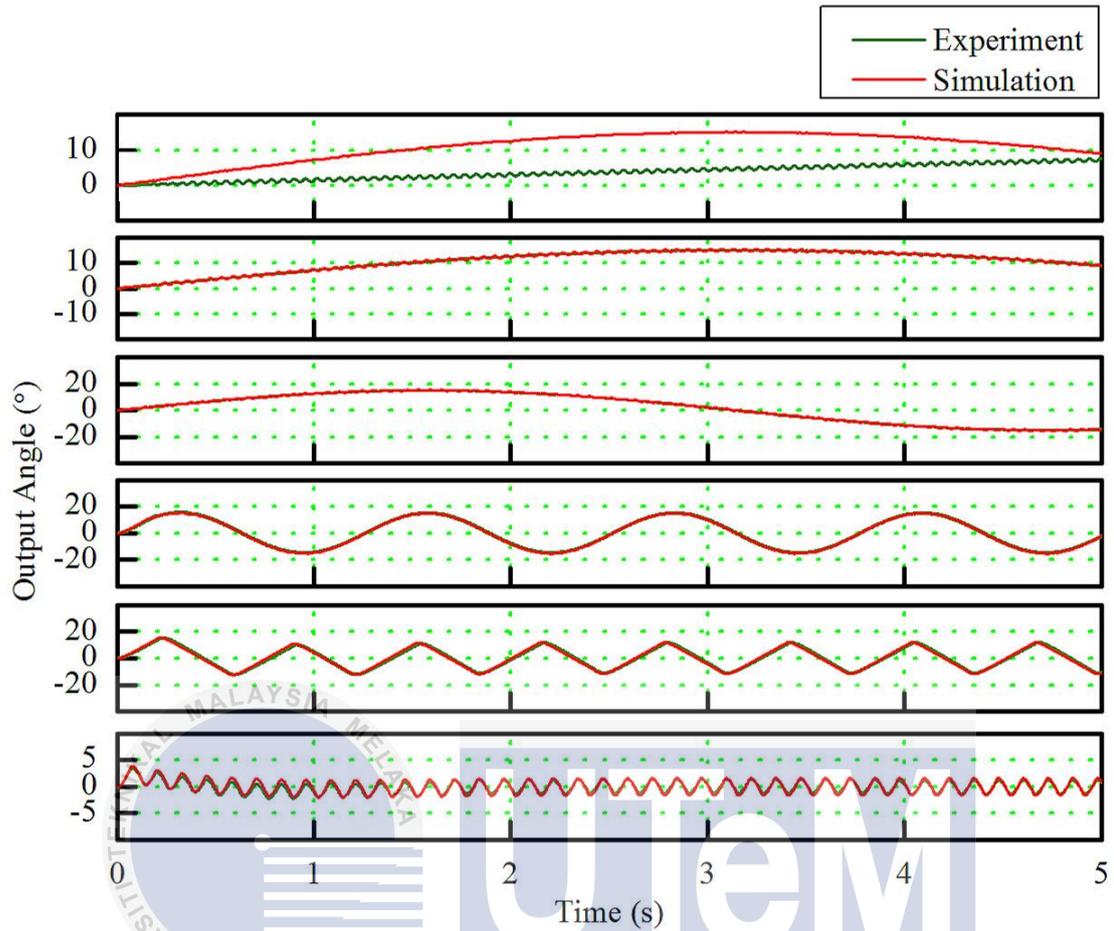


Figure 4.29: Comparison between output angles for a PID control system with different frequencies and input angle of  $15^\circ$  (Top to bottom: 0.1Hz, 0.5Hz, 1Hz, 5Hz, 10Hz, 10Hz)

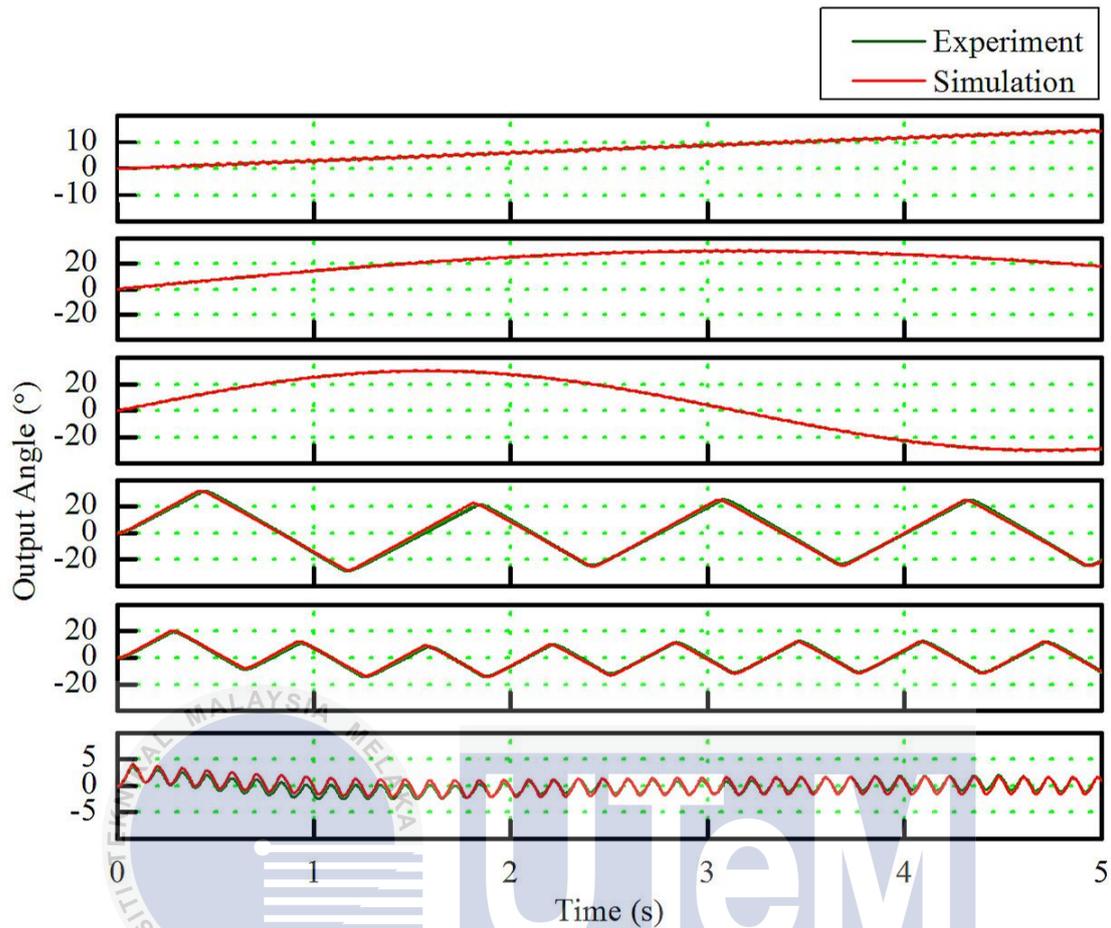


Figure 4.30: Comparison between output angles for a PID control system with different frequencies and input angle of  $30^\circ$  (Top to bottom: 0.1Hz, 0.5Hz, 1Hz, 5Hz, 10Hz, 10Hz)

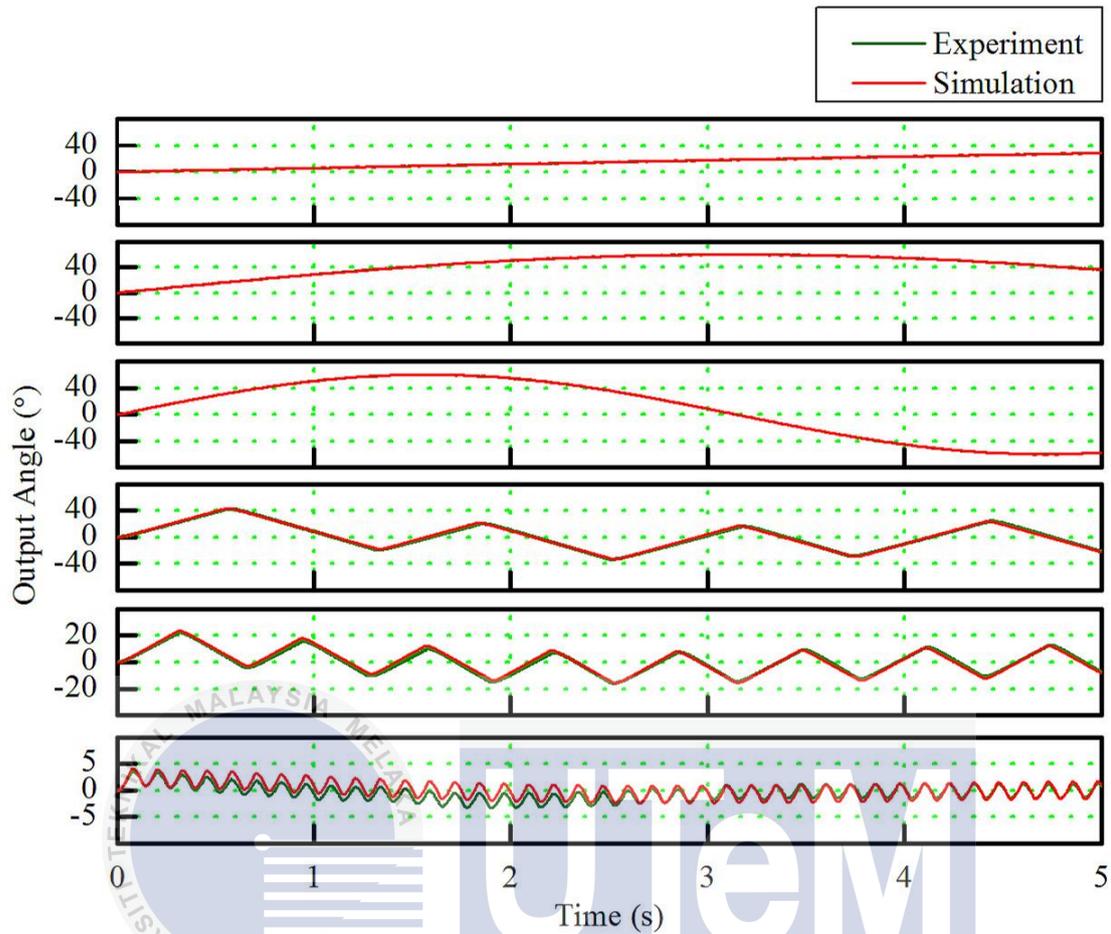


Figure 4.31: Comparison between output angles for a PID control system with different frequencies and input angle of  $60^\circ$  (Top to bottom: 0.1Hz, 0.5Hz, 1Hz, 5Hz, 10Hz, 10Hz)

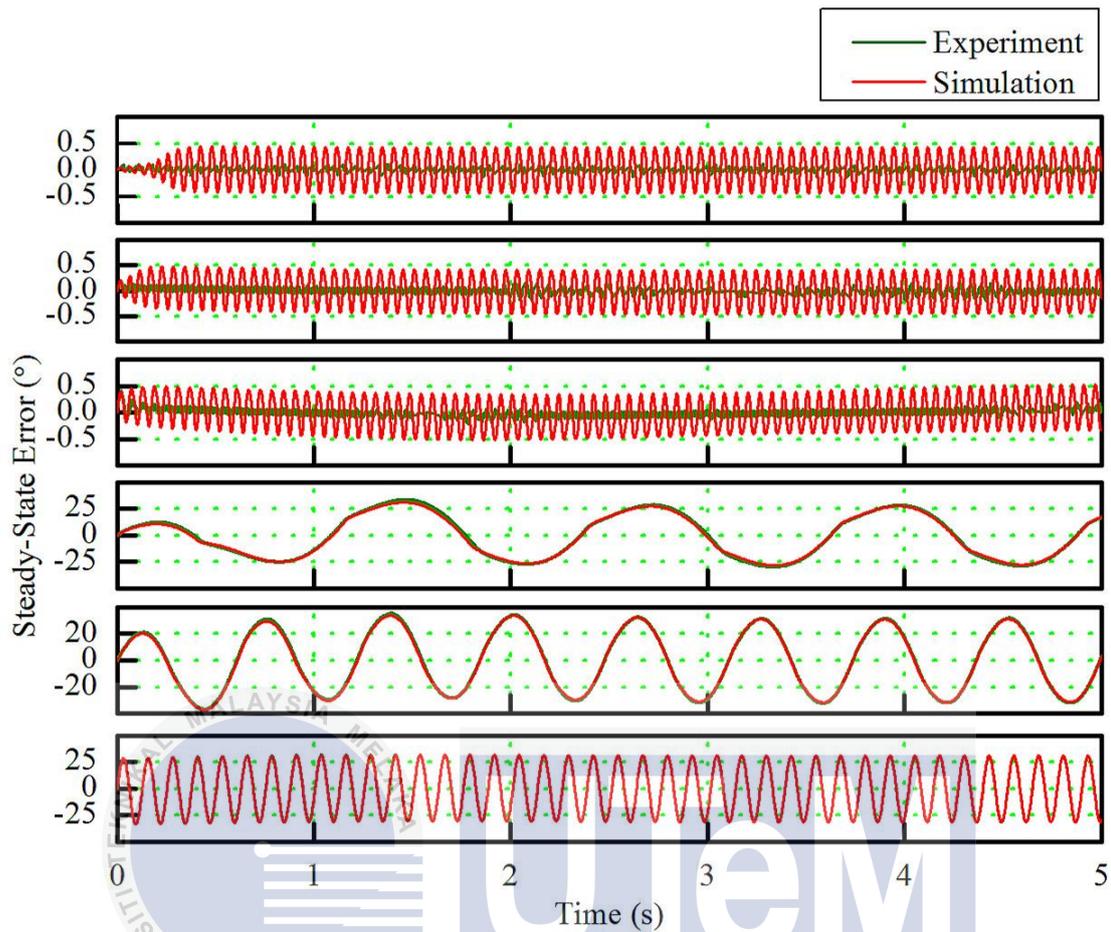


Figure 4.32: Comparison between steady-state errors for a PID control system with different frequencies and input angle of  $15^\circ$  (Top to bottom: 0.1Hz, 0.5Hz, 1Hz, 5Hz, 10Hz, 10Hz)

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

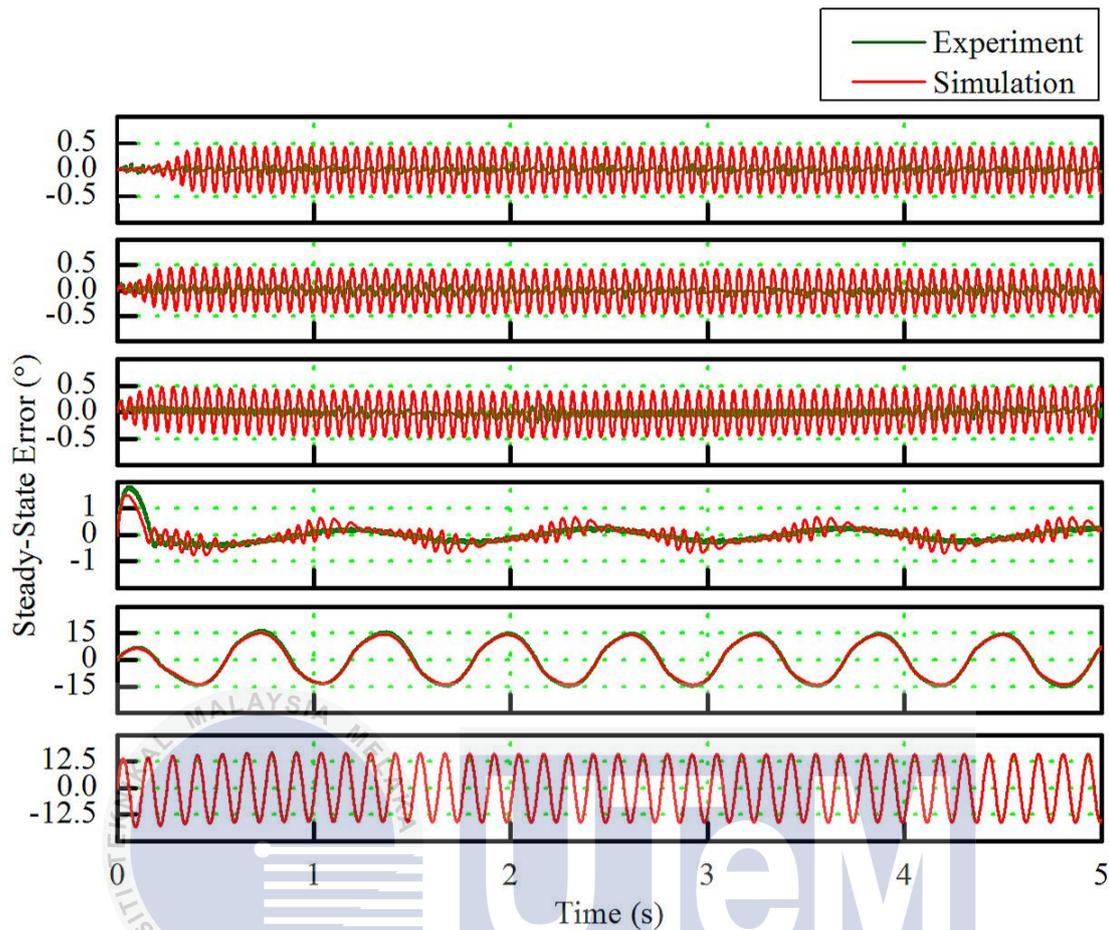


Figure 4.33: Comparison between steady-state errors for a PID control system with different frequencies and input angle of  $30^\circ$  (Top to bottom: 0.1Hz, 0.5Hz, 1Hz, 5Hz, 10Hz, 10Hz)

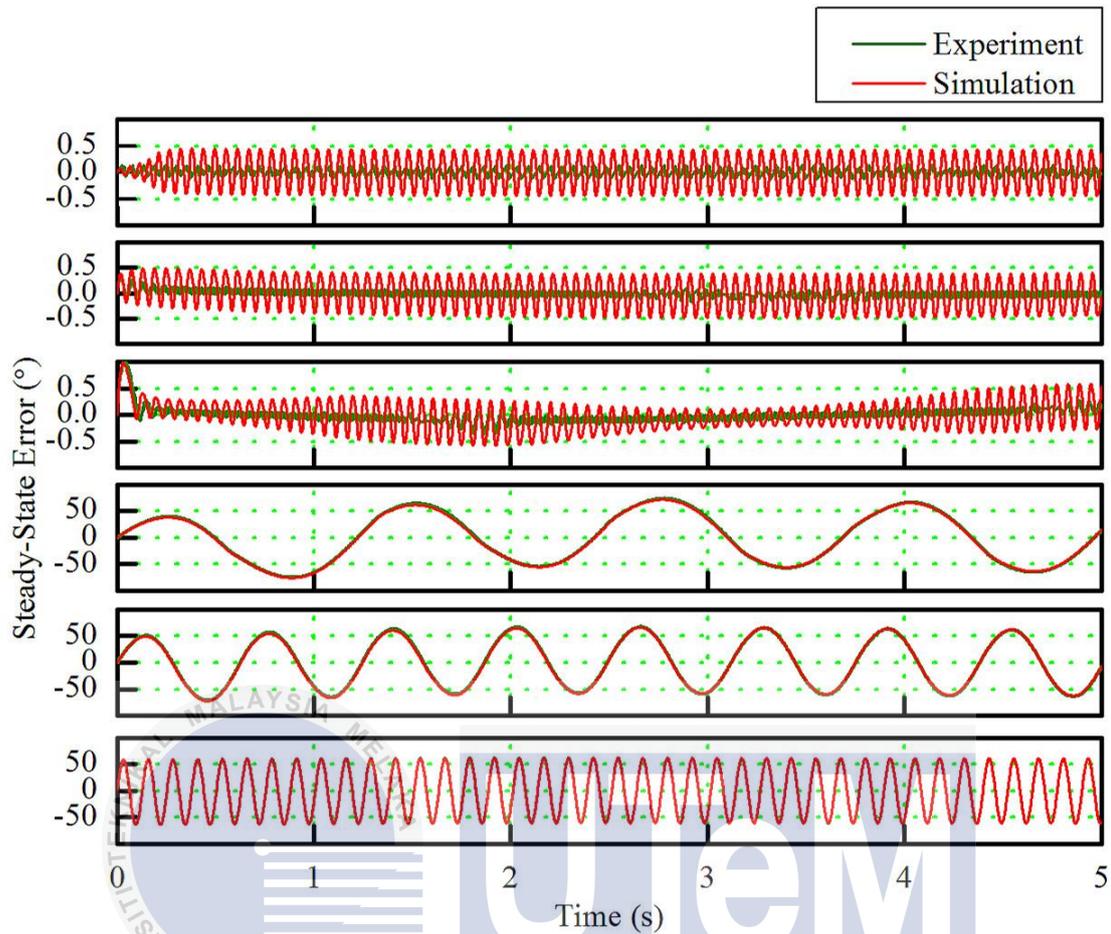


Figure 4.34: Comparison between steady-state errors for a PID control system with different frequencies and input angle of  $60^\circ$  (Top to bottom: 0.1Hz, 0.5Hz, 1Hz, 5Hz, 10Hz, 10Hz)

#### 4.4.3 Summary for PID Controller

After carrying out various types of experiments, the performance of the controller is observed. The results of simulation for PID controller in this chapter are summarized into Table 4.12.

Table 4.12: Summary of PID-controlled system

Performance	Point to Point Trajectory Control			Tracking Control		
	15 °	30 °	60 °	15 °	30 °	60 °
Steady-state error (°)	0.01	0.01	0.02	0.002	0.01	0.01
Settling time (s)	0.5	0.7	1.1	0.25	0.1	0.2
Rise time (s)	0.25	0.4	0.8	-	-	-
Overshoot (%)	13.33	6.67	3.33	-	-	-



## 4.5 Compensated System with Fuzzy Logic Controller

### 4.5.1 Point to Point Trajectory Control with Fuzzy Logic Controller

For the compensated with fuzzy logic controller, a two input two output (TITO) system is implemented. The inputs of the system are steady-state error and rate of change of error, whereas the output is the angle of the robotic arm. Table 4.13 shows the parameters being fixed and also varied in this compensated system with fuzzy logic controller. Figure 4.35 shows the results of output angle, steady-state error and input voltage when the input angle is  $15^\circ$ .

Table 4.13: Parameters for point to point experiments using fuzzy logic controller

Parameter	Numerical Value
Input Angle	$15^\circ, 30^\circ, 60^\circ$
Simulation time	5s
Sampling time	1ms
Input type	Step input
Controller	Fuzzy logic controller

From the graphs, it can be observed that the two input two output fuzzy logic controller is able to produce significant results as the output angle is  $15^\circ$ , which is the same as the input angle. The steady-state error is almost 0 radians. However, this perfect condition only occurs after 1 second. At the period of 0 second to 1 second, the results are not satisfactory. It is due to the motor need high initial torque to actuate the arm. Figure 4.36 shows the results when the input angle is  $30^\circ$ . It is observed that the system shows significant results as the steady-state error is almost 0 radians and the output angle is  $30^\circ$ , which is equivalent to the input angle.

Figure 4.37 shows the results when the input angle is  $60^\circ$ . From the graphs, it can be seen that the output angle is  $60^\circ$ , which is the same as the input angle. The steady-state error is approaches to 0 radians. Thus, the value of input angle is increased again to observe is there any changes in the output results. Figure 4.40 shows the results when the input angle is  $90^\circ$ .

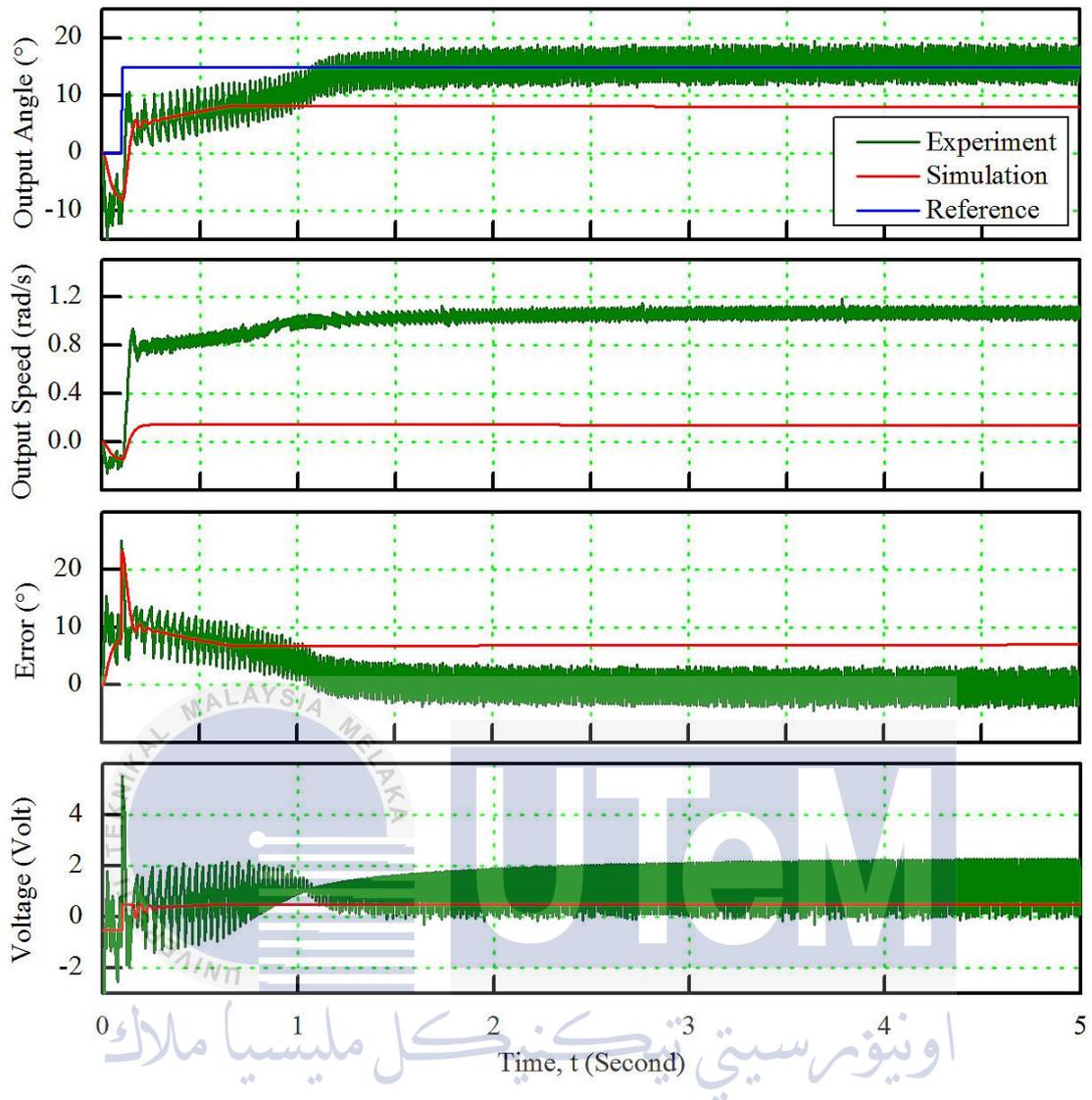


Figure 4.35: Graph of input voltage, output angle and steady-state error against time for a compensated system with input angle of  $15^\circ$

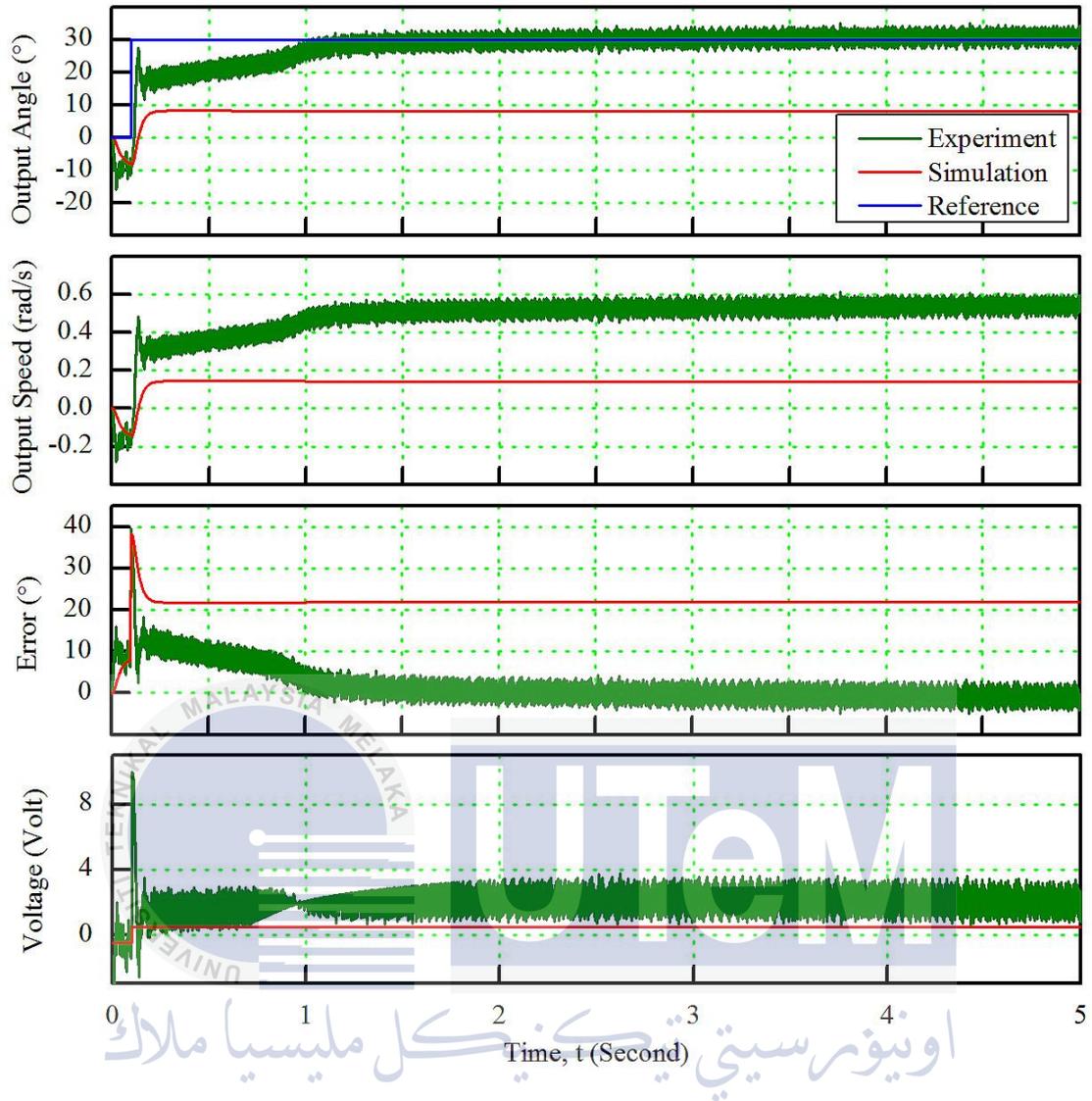


Figure 4.36: Graph of input voltage, output angle and steady-state error against time for a compensated system with input angle of  $30^\circ$

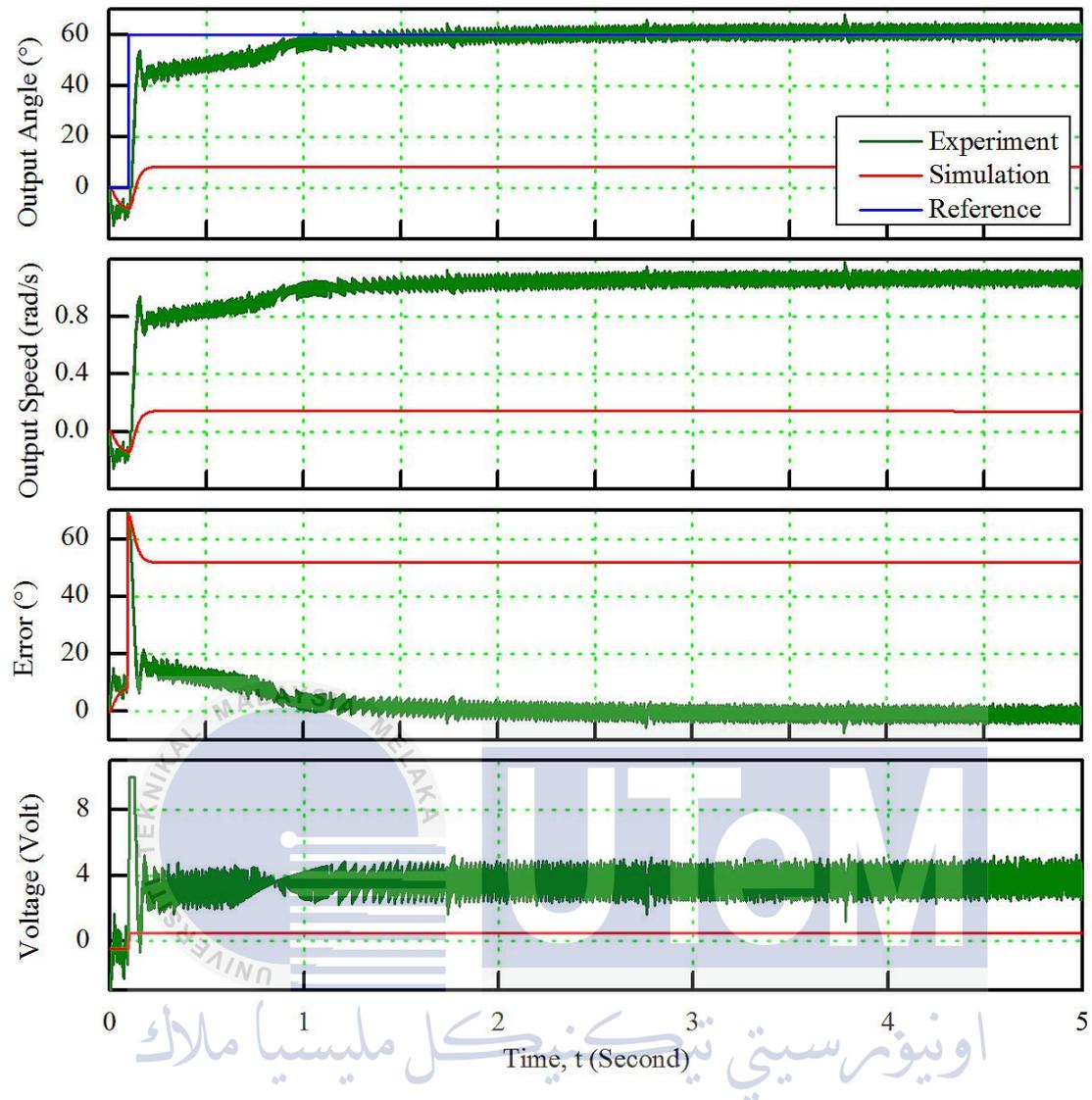


Figure 4.37: Graph of input voltage, output angle and steady-state error against time for a compensated system with input angle of  $60^\circ$

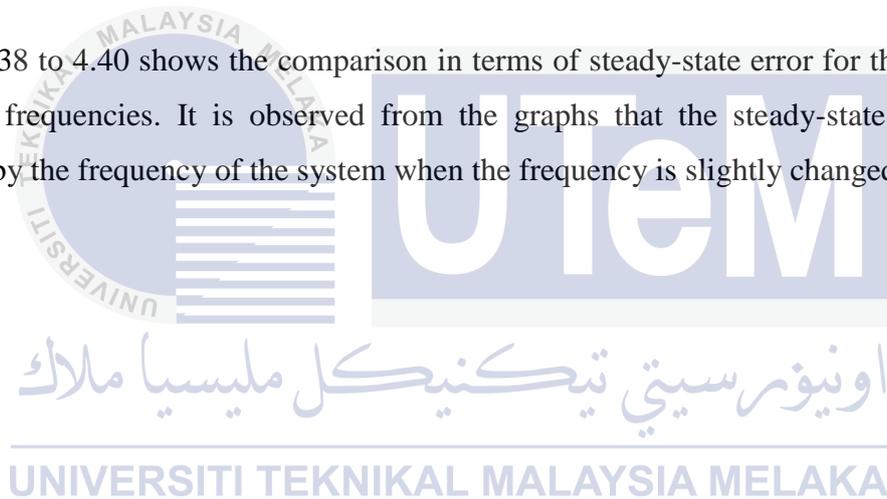
#### 4.5.2 Tracking Control with Fuzzy Logic Controller

The tracking error experiments are carried out for different input angles for a fuzzy-controller system. The frequency of the system is varied to observe the effects of different frequencies to the output signals. Table 4.14 shows the parameters being fixed and varied for the tracking error experiments.

Table 4.14: Parameters for tracking error experiments for fuzzy-controlled system

Parameter	Numerical Value
Input Angle	15 °, 30 °, 60 °
Simulation time	5s
Sampling time	1ms
Input type	Sine wave
Controller	Fuzzy logic controller

Figure 4.38 to 4.40 shows the comparison in terms of steady-state error for the systems of different frequencies. It is observed from the graphs that the steady-state error is not affected by the frequency of the system when the frequency is slightly changed.



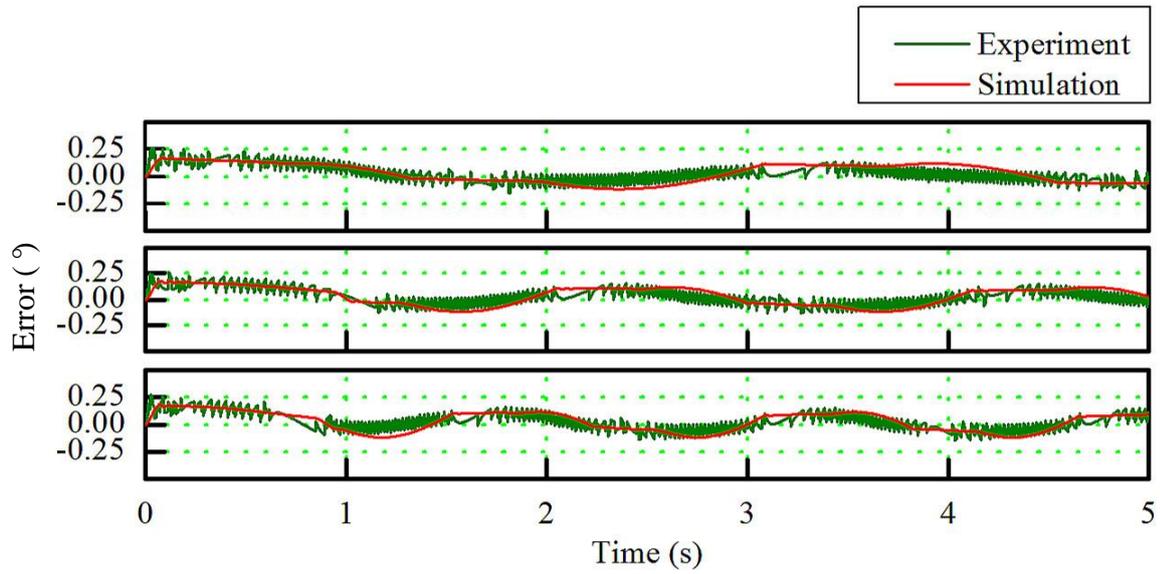


Figure 4.38: Graph of steady-state error against time for a fuzzy logic control system with an input angle of  $15^\circ$  (Top to bottom: 1Hz, 3Hz, 5Hz)

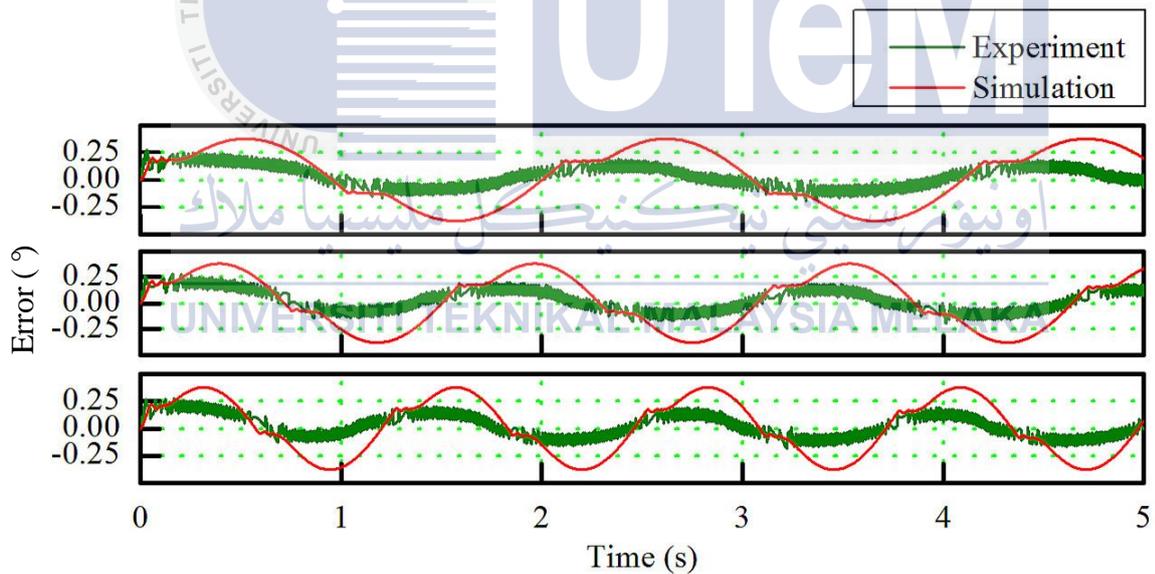


Figure 4.39: Graph of steady-state error against time for a fuzzy logic control system with an input angle of  $30^\circ$  (Top to bottom: 1Hz, 3Hz, 5Hz)

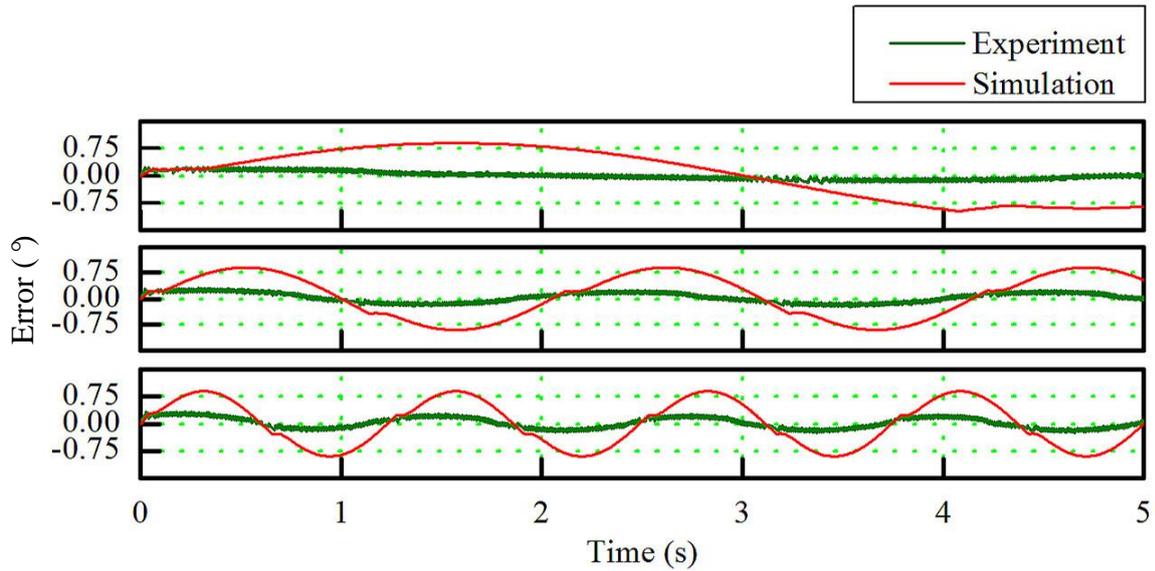


Figure 4.40: Graph of steady-state error against time for a fuzzy logic control system with an input angle of  $60^\circ$  (Top to bottom: 1Hz, 3Hz, 5Hz)

#### 4.5.3 Summary for Fuzzy Logic Controller

After carrying out various types of experiments, the performance of the controller is observed. The results of simulation for fuzzy logic controller in this chapter are summarized into Table 4.15.

Table 4.15: Summary of Fuzzy Logic controlled system

Performance	Point to Point Trajectory Control			Tracking Control		
	$15^\circ$	$30^\circ$	$60^\circ$	$15^\circ$	$30^\circ$	$60^\circ$
Steady-state error ( $^\circ$ )	0.03	0.03	0.04	0.12	0.15	0.06
Settling time (s)	0.6	0.4	1.0	0.2	0.2	0.1
Rise time (s)	0.8	0.8	0.9	-	-	-
Overshoot (%)	33.33	42.86	25	-	-	-

## CHAPTER 5

### CONCLUSION AND RECOMMENDATION

#### 5.1 Conclusion

In this project, a prototype of upper limb of robotic arm is analyzed. PID controller and fuzzy logic controller are chosen to be compared in terms of their performance such as steady-state error, settling time and rise time. Several types of experiments are carried out involving open loop systems, uncompensated systems and compensated systems. Two types of experiments are carried out to control the motion of the robotic arm, namely point to point trajectory control and tracking control.

The comparisons between two types of controllers are summarized in Table 5.1. It is found that PID controller is more capable in eliminating the steady-state error, whereas fuzzy logic controller demonstrates shorter settling time compared to PID controller. However, the rise time of fuzzy logic controller is higher compared to PID controller. In short, PID is a better choice in precision motion control compared to fuzzy logic controller.

Table 5.1: Comparison between the performance of PID controller and fuzzy logic controller

Performance of PID Controller	Point to Point Trajectory Control			Tracking Control		
	15 °	30 °	60 °	15 °	30 °	60 °
Steady-state error (°)	0.01	0.01	0.02	0.002	0.01	0.01
Settling time (s)	0.5	0.7	1.1	0.25	0.1	0.2
Rise time (s)	0.25	0.4	0.8	-	-	-
Overshoot (%)	13.33	6.67	3.33	-	-	-
Performance of Fuzzy Logic Controller	Point to Point Trajectory Control			Tracking Control		
	15 °	30 °	60 °	15 °	30 °	60 °
Steady-state error (°)	0.03	0.03	0.04	0.12	0.15	0.06
Settling time (s)	0.6	0.4	1.0	0.2	0.1	0.1
Rise time (s)	0.8	0.8	0.9	-	-	-
Overshoot (%)	33.33	42.86	25	-	-	-

## 5.2 Recommendations

For future research, it is recommended that a more precise controller can be implemented to control the motion of the upper limb of a robotic arm. For example, fuzzy-PID controller can be used to control the speed as well as the output angle of the robotic arm. It is not implemented in this project due to its complexity and difficulty in tuning.

In addition, a more precise motor can be used in the system to yield higher performance and sensitivity of the robot manipulator. For example, a servo motor can be used although it is more complicated than DC motor.

## REFERENCES

- [1] Yamaguchi, T., Hirata, M., Chee, K.P., *High-Speed Precision Motion Control*, pp. 1-8. USA: CRC Press, Taylor & Francis Group, 2011
- [2] Chen, F., Sekiyama, K., Di, P., Huang, J., Fukuda, T., “i-Hand: an intelligent robotic hand for fast and accurate assembly in electronic manufacturing”, presented at the proceedings of IEEE International Conference on Robotics and Automation, River Centre, Saint Paul, Minnesota, USA, May 14-18, 2012.
- [3] Fu, K. S., Gonzalez, R. C., Lee, C.S.G. *Robotics: Control, Sensing, Vision and Intelligence*, pp. 1-6. Singapore: McGraw-Hill, Inc., 1987.
- [4] U.S. Department of Labor, *OSHA Technical Manual (OTM) Section IV: Chapter 4*, 1999.
- [5] National Instruments Corporation, *Fundamentals of Motion Control*, 2013.
- [6] Kirchoff, S., Melek, W.W., A Saturation-Type Robust Controller for Modular Manipulators Arms. *Mechatronics 17*, pp. 175–190, 2007.
- [7] Lee, C. S., Gonzalez, R.V., Fuzzy Logic Versus A PID Controller for Position Control of A Muscle-Like Actuated Arm. *Journal of Mechanical Science and Technology 22* 1475 – 1482, 2008.
- [8] Kazemian, H. B. Comparative Study of a Learning Fuzzy PID Controller and a Self-Tuning Controller. *ISA Transactions 40*, pp. 245 – 253, 2001.
- [9] Cytron Technologies Sdn. Bhd., *DC Geared Motor with Encoder: Product User's Manual - MO-SPG-30E-XXXX*, 2013.
- [10] Solutions4U, *Electro-mechanical Engineering Laboratory Manual Version 1.1*, 2009.
- [11] Robert, B., Stefano, S., *Matlab and Simulink for Modeling and Control*. Delft University of Technology, Netherlands, 1999.

- [12] Aamir, M., On replacing PID controller with ANN controller for DC motor position control. *International Journal of Research Studies in Computing, Volume 2 Number 1*, pp. 21-29, 2013.
- [13] Nise, N. S., *Control Systems Engineering*, 6<sup>th</sup> ed., Asia: John Wiley & Sons Pte Ltd., 2011.
- [14] The MathWorks Inc., *Introduction: PID Controller Design*, 2013.
- [15] Levine, W.S., *The Control Handbook*. USA: CRC Press LLC., 1995.
- [16] Schwarzenbach, J., Gill, K.F., *System Modelling and Control*, 3rd ed. Great Britain, 1992.
- [17] Feng, G., *Analysis and Synthesis of Fuzzy Control Systems: A Model-based Approach*. USA: CRC Press LLC., 2010.



## APPENDIX A

## Final Year Project Research Gantt Chart

	Year	2013				2014						
		Task/ Month	Sep	Oct	Nov	Dec	Jan	Feb	Mac	Apr	May	Jun
<b>Final Year Project 1</b>												
1	Search for Supervisor and get the title											
2	Modeling of the system											
3	Motor selection											
4	Material selection											
5	Experimental setup											
6	One-day Microbox workshop											
7	Open loop simulation											
8	Submit draft of report											
9	Seminar											
10	Submit FYP 1 report											
<b>Final Year Project 2</b>												
11	Closed loop simulation for uncompensated system											
12	Compensated system with PID controller											
13	Compensated system with Fuzzy Logic controller											
14	Results and Analysis											
15	Submit draft of report											
16	Seminar											
17	Submit FYP 2 report											

## APPENDIX B

## Project Research Methodology Flow Chart

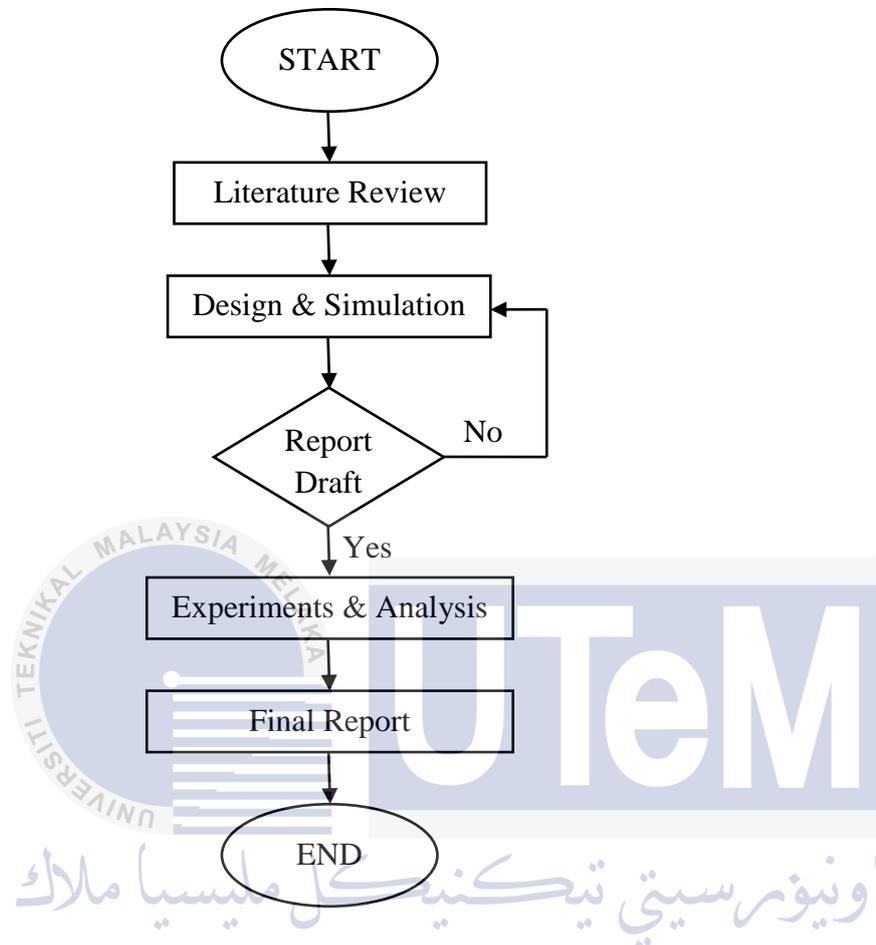


Figure 6.1: Project research methodology flow chart

## APPENDIX C

## Project Experiment Methodology Flow Chart

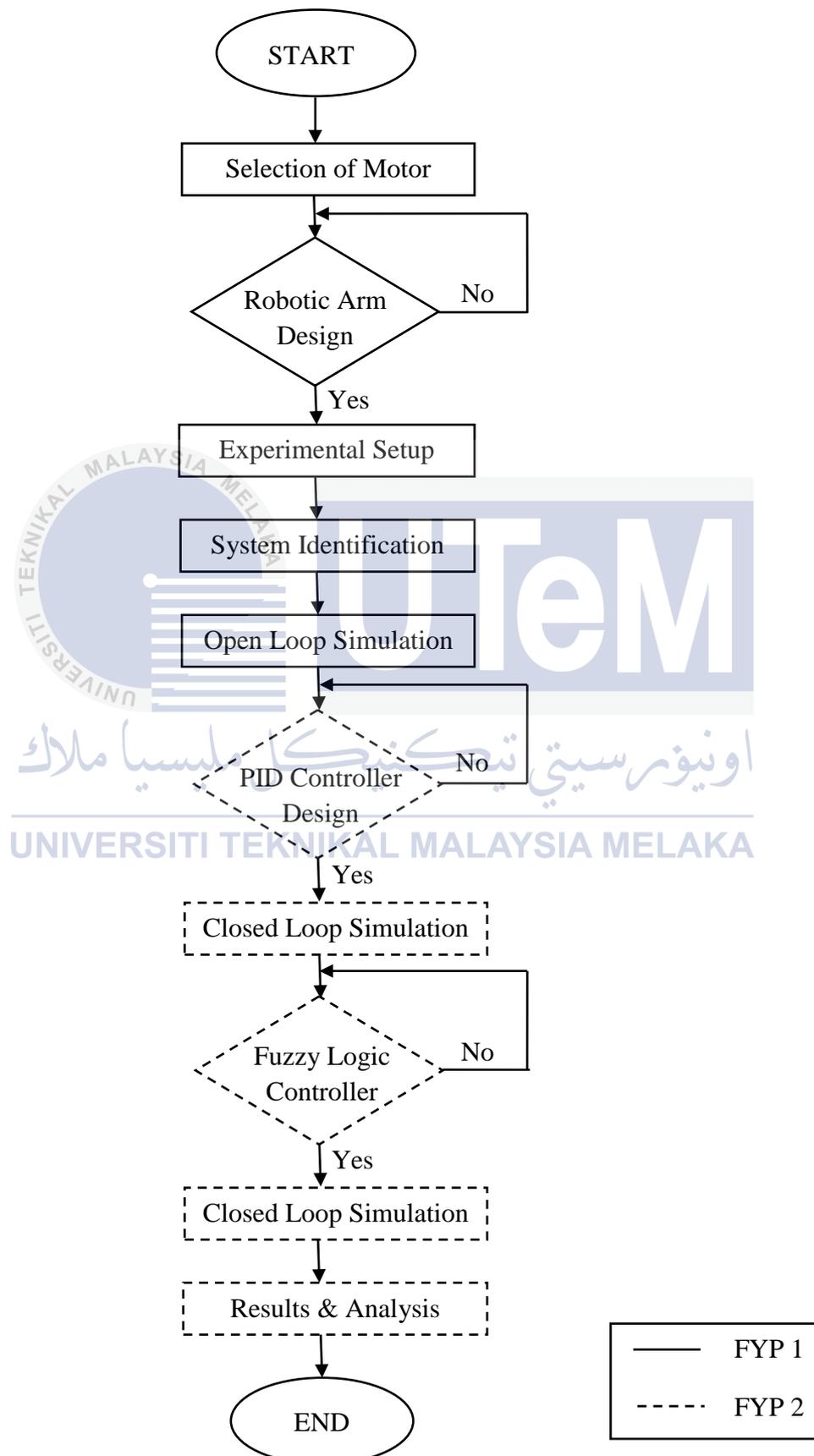


Figure 6.2: Project experiment methodology flow chart

## APPENDIX D

### Connections of Cytron DC Geared Motor

Cytron DC geared motor has 6 connection ports. Pin 1 and Pin 2 are the output ports of the motor driver circuit in which the voltage is supplied. Pin 3 and Pin 4 supply a constant 5V voltage to Hall Effect sensor. Pin 5 and Pin 6 are the output channels of Hall Effect sensor.

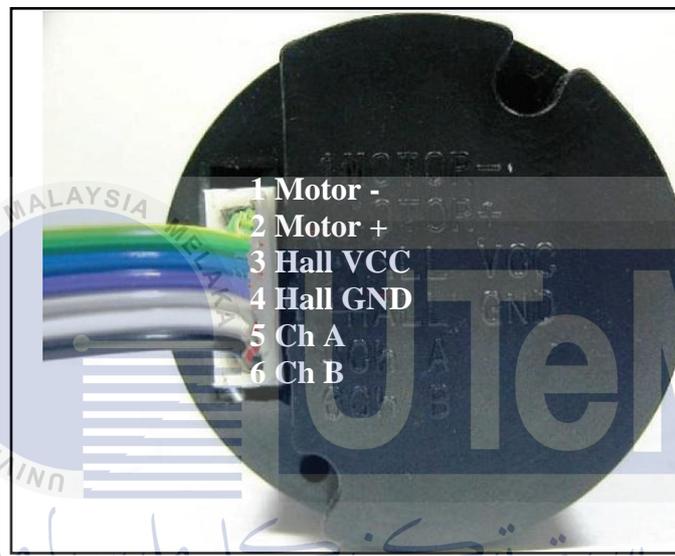


Figure 6.3: Rear view of the motor with encoder and cover

Table 6.1: Name and functions of motor pins

Pin	Name	Function
1	Motor -	Output of motor driver (Negative terminal)
2	Motor +	Output of motor driver (Positive terminal)
3	Hall Effect Sensor Vcc	Supply voltage for sensor circuit (4.5 ~ 5.5V)
4	Hall Effect Sensor Ground	Ground
5	Channel A	Output of the encoder
6	Channel B	Output of the encoder

## APPENDIX E

### Block Diagram of Driver Circuit Board

Driver circuit is one of the components in Micro-Box module. The block diagram of driver circuit is necessary for the user to know exactly which pin is connected to which wire of motor. In this project, only robotic arm is implemented. Hence, only Pin 1 to Pin 7 (Arm) is taken into consideration. In other words, Pin 8 and Pin 9 (Pendulum) are unused.

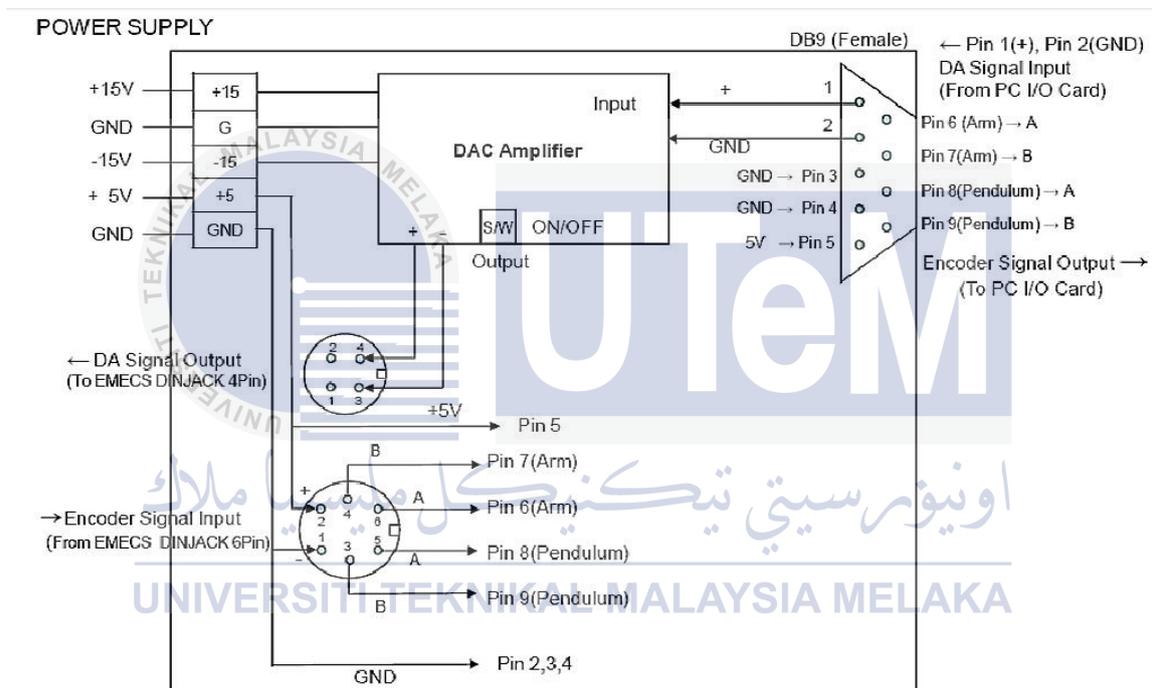


Figure 6.4: Block diagram of driver circuit

APPENDIX F

Procedure of Open Loop Control Simulation

- i. Open loop electrical model (experiment)

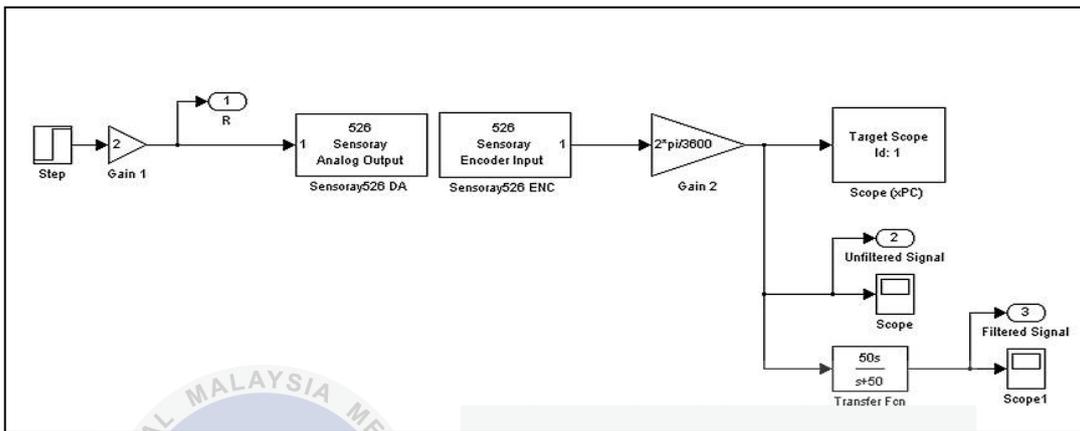


Figure 6.5: Open loop Simulink block diagram

- ii. Open loop electrical model (Combine experiment and simulation)

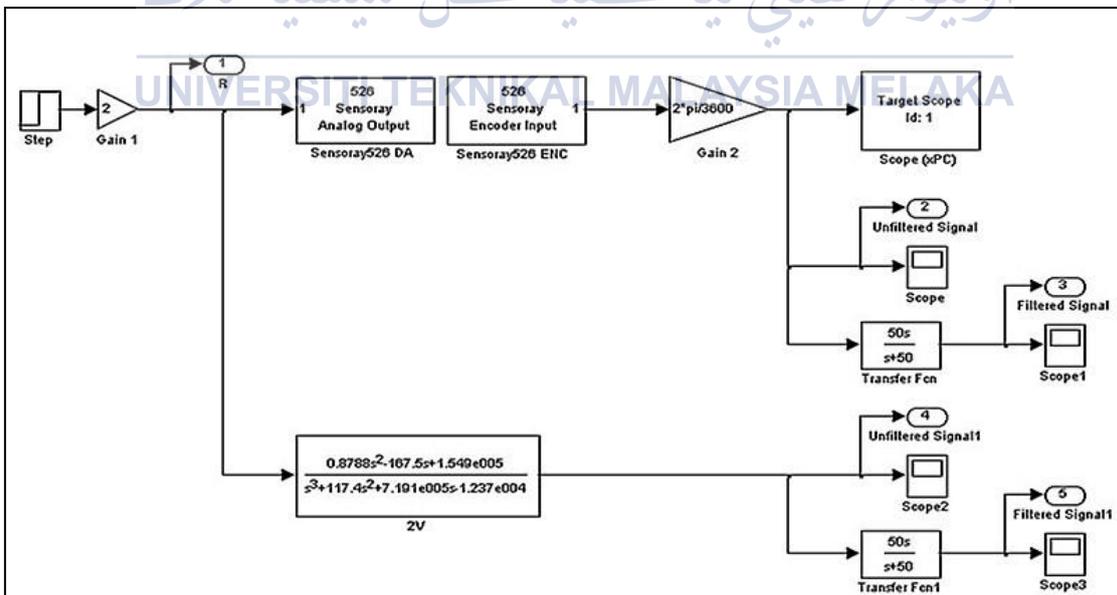


Figure 6.6: Open loop Simulink block diagram with transfer function

## APPENDIX G

### Fuzzy Rule-Based System

1. If (Error is LN) and (Integral is LN) then (Output is LN)
2. If (Error is SN) and (Integral is LN) then (Output is LN)
3. If (Error is ZE) and (Integral is LN) then (Output is SN)
4. If (Error is SP) and (Integral is LN) then (Output is SN)
5. If (Error is LP) and (Integral is LN) then (Output is ZE)
6. If (Error is LN) and (Integral is SN) then (Output is LN)
7. If (Error is SN) and (Integral is SN) then (Output is SN)
8. If (Error is ZE) and (Integral is SN) then (Output is SN)
9. If (Error is SP) and (Integral is SN) then (Output is ZE)
10. If (Error is LP) and (Integral is SN) then (Output is SP)
11. If (Error is LN) and (Integral is ZE) then (Output is SN)
12. If (Error is SN) and (Integral is ZE) then (Output is SN)
13. If (Error is ZE) and (Integral is ZE) then (Output is ZE)
14. If (Error is SP) and (Integral is ZE) then (Output is SP)
15. If (Error is LP) and (Integral is ZE) then (Output is SP)
16. If (Error is LN) and (Integral is SP) then (Output is SN)
17. If (Error is SN) and (Integral is SP) then (Output is ZE)
18. If (Error is ZE) and (Integral is SP) then (Output is SP)
19. If (Error is SP) and (Integral is SP) then (Output is SP)
20. If (Error is LP) and (Integral is SP) then (Output is LP)
21. If (Error is LN) and (Integral is LP) then (Output is ZE)
22. If (Error is SN) and (Integral is LP) then (Output is SP)
23. If (Error is ZE) and (Integral is LP) then (Output is SP)
24. If (Error is SP) and (Integral is LP) then (Output is LP)
25. If (Error is LP) and (Integral is LP) then (Output is LP)