ANALYSIS REAL TIME CONTROL OF BEHAVIOR ROBOT IN ASSITING HUMAN IN CLEANING HOUSEHOLD

HEE KEE MING

BACHELOR OF MECHATRONICS ENGINEERING WITH HONOR

UNIVERSITY TEKNIKAL MALAYSIA MELAKA

" I hereby declare that I have read through this report entitle "ANALYSIS REAL TIME CONTROL OF BEHAVIOR ROBOT IN ASSITING HUMAN IN CLEANING HOUSEHOLD" and found that it has comply the partial fulfillment for awarding the degree of Bachelor of Mechatronic Engineering with Honor"

Signature : ....

Supervisor's Name : ....

Date : ....

ANALYSIS REAL TIME CONTROL OF BEHAVIOR ROBOT IN ASSITING HUMAN IN CLEANING HOUSEHOLD

HEE KEE MING

A report submitted in partial fulfillment of the requirements for the degree of Bachelor of Mechatronic Engineering with Honor

FACULTY OF ELECTRICAL ENGINEERING
UNIVERSITY TEKNIKAL MALAYSIA MELAKA

2015

I declare that this report entitle “ANALYSIS REAL TIME CONTROL OF BEHAVIOR ROBOT IN ASSITING HUMAN IN CLEANING HOUSEHOLD” 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.

Signature : ....

Name : ....

Date : ....

ACKNOWLEDGEMENT

I would like to thanks Universiti Teknikal Malaysia Melaka (UTeM) give me an opportunity to take part in final year project (FYP). FYP give me a platform that able to learn new knowledge. Thank You for the facility that provided by UTeM so that I able to learn or conduct final year project in a suitable environment.

Besides, I would like to express my deepest gratitude to my final year project Supervisor, Nur Maisarah Binti Mohd Sobran who guides me to accomplishing this final year project topic. She gave me a lot of advice and opportunity to complete my FYP topic under her supervisions.

ABSTRACT

Technology of service robot keep improving from year by year due to need of human to solve difference kind of problem that face in daily life. Service robot in market is moving randomly when operate and this will cause it miss out some of the part in the room and increase the time needed to completely clean up the room. In order to overcome this problem, a prototype that able to learn by demonstration will be create to make the cleaning robot able to clean the room more efficiency. Cleaning robot that design will not include avoid obstacle sensor to reduce the complex of this project because main focus for project is learn by demonstration and human will avoid obstacle when doing demonstrate trajectory. Prototype will have two mode that is record mode that memory demonstrate trajectory by user and replay mode that redo the trajectory did by user. Basic sensor like encoder and digital compass sensor will be used in this project to detect distance travel and facing angle of cleaning robot. Number of step of encoder for one complete cycle and accuracy of digital compass sensor had been test in this project. Two difference system that is open-loop system and closed-loop system had been test to find the best system to operate cleaning robot. Cleaning robot able to show out the trajectory by using both of the system but closed-loop system is much more better compare with open-loop system because the end point of desired and actual point for closed-loop system is closer compare to open-loop system.

ABSTRAK

Teknologi perkhidmatan robot terus meningkatkan dari tahun ke tahun kerana keperluan manusia untuk menyelesaikan perbezaan jenis masalah yang dihadapi dalam kehidupan seharian. Robot perkhidmatan seperti robot vakum di pasaran bergerak secara rawak apabila beroperasi dan ini akan menyebabkan ia terlepas beberapa bahagian di dalam bilik. Jalan rawak akan terlepas beberapa kawasan pembersihan dan akan meningkatkan keperluan masa diperlukan untuk membersihkan bilik. Untuk mengatasi masalah ini, prototaip yang dapat program jalan pembersihan akan mencipta untuk membuat robot pembersihan dapat membersihkan bilik kecekapan lanjut. Prototaip ini tidak akan mengandungi sebarang sensor untuk mengelakkan halangan kerana pengguna akan mengelakkan halangan apabila pengaturcaraan jalan pembersihan untuk robot. Prototaip akan mempunyai dua mod iaitu mod rekod memori mencatat trajektori oleh pengguna dan ulangan mod yang buat semula trajektori yang lakukan oleh pengguna. Sensor asas seperti pengekod sensor dan kompas digital akan digunakan dalam projek ini untuk mengesan perjalanan jarak dan menghadap sudut pembersihan robot. Bilangan langkah pengekod untuk satu kitaran lengkap dan ketepatan sensor kompas digital telah ujian dalam projek ini. Dua sistem perbezaan iaitu sistem gelung buka dan sistem gelung tertutup telah ujian untuk mencari sistem yang terbaik untuk mengendalikan pembersihan robot. Pembersihan robot dapat menunjukkan daripada trajektori dengan menggunakan kedua-dua sistem itu tetapi sistem gelung tertutup adalah lebih baik berbanding dengan sistem gelung buka kerana titik akhir titik dikehendaki dan sebenar untuk sistem gelung tertutup adalah lebih dekat berbanding dengan membuka gelung system.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Introduction

Research background, motivation, problem statement, objectives and scope will be present in this chapter.

1.2 Research Background

Cleaning is an important activity that happens in our daily life. Clean environment able to maintain human body stay in health and comfortable. To maintain housing area always in clean condition although human does not have time do household chores, cleaning robot is one of the best solution. There have many types of cleaning robot that exists in the market and able to assist human doing household chore likes road cleaning robot, vacuum robot, window cleaning robot, floor cleaning robot and swimming pool cleaning robot. The image of the cleaning robot then mention was shown in Table 1. The reason that cleaning robot able to assist human doing household chores is because it is able to operate automatically and clean desired place. Some of the cleaning robot is able to detect surrounding obstacle that blocking the way and avoid it. Besides, cleaning robot like vacuum robot contain dust sensor that able to detect surrounding dust and clean the dust area. After the statement mention above, it show that cleaning robot able to clean housing area although no body in the house. Unfortunately, most of the cleaning robot in the market moving trajectory are in random motion. To overcome the problem, designer include some sensor to prevent collision occur. In this project, a bathroom cleaning robot that able to learn from human demonstrate trajectory and redo the demonstrate trajectory will be develop. With this kind of cleaning robot, user is able to program trajectory of the cleaning robot compare with cleaning robot in the market that move randomly. Besides, the sensor that detect surrounding environment will not be include in this project.

Table 1.1: Cleaning Robot exists in Market

Road Cleaning Robot	
Vacuum Robot	
Window Cleaning Robot	
Floor Cleaning Robot	
Swimming Pool Cleaning Robot	

1.3 Motivation

Cleaning is an important task for human to maintain healthy life. Unfortunately, human now are busy working and lack of time to clean their housing environment. To overcome this problem, they need a cleaning robots that able to assist them to conduct household chores. Although market already have many types of cleaning robot for household chores but cleaning robot that designed to clean bathroom still not familiar.

One of the motivation for me to conduct this project is because one of the magazine [1] title "When Will We Have Robots To Help With Household Chores?" from IEEE Spectrum. From the title of the magazine, it show that human are willing to have a cleaning robot that able to reduce their working task to finish their household chores. In this magazine, it mention that big company like Google has recently start buying spree of Robotics Company. It show that cleaning robot become important for the future and it will improve quality of life of human. With the research of big company, it will greatly improve or accelerate robotics technology in the world. This is because big company willing to pay huge amount of money for research.

Another magazine [2] from IEEE Spectrum "So, Where Are My Robot Servants?" mention that the idea of robots doing chores around the house has long captured by people's imaginations and robot would help them live independently longer and providing care. From both of the magazine, it show that the demand of human on cleaning robot are keep on increasing and many research for household chore robot are in progress.

Besides, most of the cleaning that sell in the market are move randomly. Cleaning robot that move randomly does not able program their trajectory. They are moving in random path and using sensor to detect surrounding environment to prevent collision like wall and obstacle. Besides of that, when cleaning robot move randomly path, some of the area will be left out. Therefore, to ensure cleaning robot able to clean housing area more efficient, an idea to develop a bathroom cleaning robot that able to program trajectory by human was coming out. Cleaning robot that able to program trajectory able to clean housing area as user like. Therefore, cleaning robot will not move randomly and miss out some area like corner of the room. To provide better housing environment for human, bathroom cleaning robot that able to learn to do what they do will be design in this project to reduce time taken to finish chores and user is able to program cleaning robot trajectory motion as they like.

The cleaning robot that fix to clean bathroom is just a prototype, the main focus of this project is the technology that able to program trajectory of cleaning robot as user demonstrate and redo the trajectory. This technology able to include into every cleaning robot if successful.

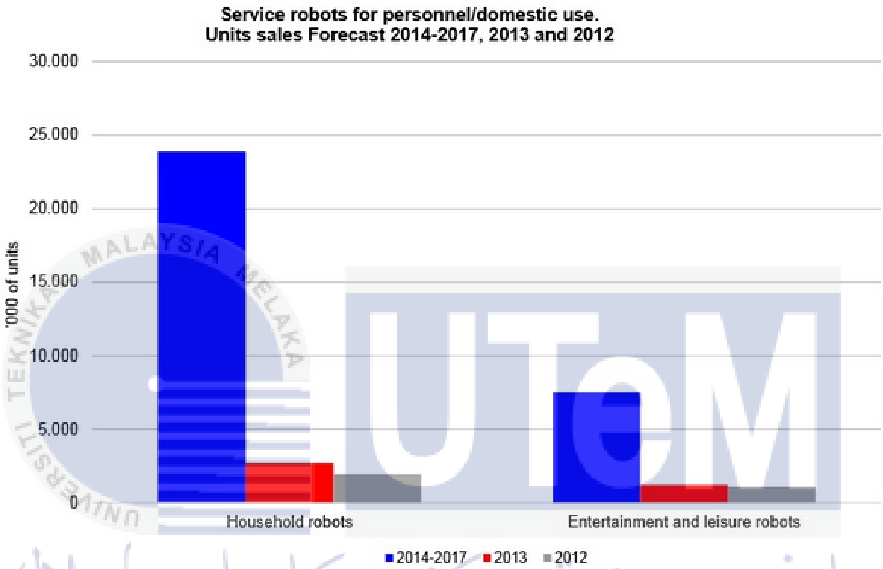


Figure 1.1: Service Robot for Personal/Domestic Use[3]

From the figure 1 it show that units of household robots are keep on increasing year by year. It show that demand of human to household robot will keep on increasing. Therefore, human will need household robot to help them solve daily life in their life.

1.4 Problem Statement

The main function of cleaning robot is to clean the selected area nicely. Cleaning robot should be able to clean the surface that passes by perfectly. Most of the bathroom floor is rough surface to prevent human fall down because of slippery. Therefore, the material of the brush is important enough when clean the surface. Material for brush to clean rough surface should be strong enough to clear the dirt inside slit.

Besides, weight of cleaning robot is also important when design the prototype. The weight of bathroom cleaning robot should be big enough to pull down the body of cleaning robot. This is because when body of cleaning robot is light, surface of brush and floor does not able to contact nicely to clean the floor. Besides, bigger weight will increase the grip between the tires and floor. Tires that have proper grip on the floor is able to prevent cleaning robot move out of desired trajectory. The brushes that rotate to clean surface will also well contact when weight pull down the body of robot.

DC motor with encoder was used to detect the motion of tires and store it into memory. Data that store in memory will be used when bathroom cleaning robot operate. Rotation motion of brush will create unwanted force that will affect BCR does not move in desired trajectory. Compass sensor was used to overcome the problem when desired angle and real-time angle are difference. Compass sensor that use able to record the angle of cleaning robot facing when human demonstrate the trajectory.

The position of digital compass sensor place in body of cleaning robot is important. If the sensor place position not suitable, data that collected is not accurate and bathroom cleaning robot will not face to selected angle. When bathroom cleaning robot faces to difference angle and move, this will increase the path error when cleaning robot redo trajectory that programed. Therefore, position to place digital compass sensor should be test for few times and get the best position to place the sensor.

Encoder will detect one step when rotary encoder rotates in a specific value of angle. For the encoder that use in this project it will detect one encoder step when it rotate 1.44 degree. Therefore, one whole cycle that contains 360 degree will contains 250 steps of encoder value.

1.5 Objectives

The objectives for this project are:

1. To design a prototype cleaning robot that able to clean bathroom environment.

2. To develop a bathroom cleaning robot that able to learn from human demonstrate trajectory and redo the demonstrate trajectory.

3. To analysis the performance of developed cleaning robot in terms of direction, maneuverings movements and path error.

1.6 Scope

To achieve the objectives that mention, scope for this project was list as below:

1. Material to design prototype will be plastic, rubber and metal that easily get from market.

2. Bathroom cleaning robot (BCR) that design only allows to operate on floor that is flat surface. The BCR does not able to climb up or move to another level when operate.

3. Analysis of cleaning robot will be done in three difference shapes like straight line, round shape and square shapes.

4. Angle that able to detect by digital compass sensor in cleaning robot only in X- Axis and Y-Axis.

5. Bathroom cleaning robot does not include avoid obstacle function.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In this chapter the concept to design learns by demonstrate bathroom cleaning robot will discuss. Besides, theory and component used will be discussed in this chapter.

2.2 Design of Cleaning Robot in Market

There are many cleaning robot for household chores in the market. Unfortunately, most of the cleaning robot are moving in random position and use sensor to detect surrounding to prevent collision. After go though some journal, it show that most of the cleaning robot in market are using collision sensor, infrared sensor, dust sensor, ultrasonic sensor, sonar range sensor, laser ranger finder and acceleration sensor to detect surrounding environment when moving in random[4]. Many types of sensor will be include in the cleaning robot just to prevent cleaning robot stuck or collide of cleaning robot and obstacle. This is because cleaning robot that sell in market is design to clean user house and every house will have difference size and contain obstacle.

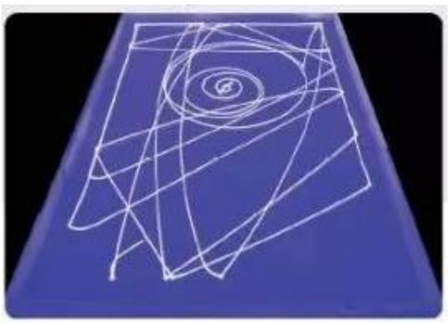


Figure 2.1: Random Path Planning

2.3 Idea to Develop Bathroom Cleaning Robot (BCR)

The general idea to design path learning bathroom cleaning robot (BCR) is based on the journal [7]. It writes about the idea to program robot trajectory follow a person demonstration. Therefore, this project is to design a prototype of BCR to learn human demonstrate trajectory and show the trajectory movement when operate. The way they use to program cleaning robot is using a broomstick to move cleaning robot. Cleaning robot will automatically generated trajectory that demonstrate and matches the trajectory when operate.

This is because cleaning robot that move randomly contain some disadvantages. That is cleaning robot will miss out some part when doing cleaning process[5]. Beside of that, time taken for cleaning robot that moving randomly to finish cleaning process will be longer compare with cleaning robot that able to program trajectory[6]. To reduce the component use to build up a cleaning robot and reduce time taken for cleaning robot finish cleaning process, a bathroom cleaning robot that able to learn from demonstrate trajectory and redo demonstrate trajectory will be create.

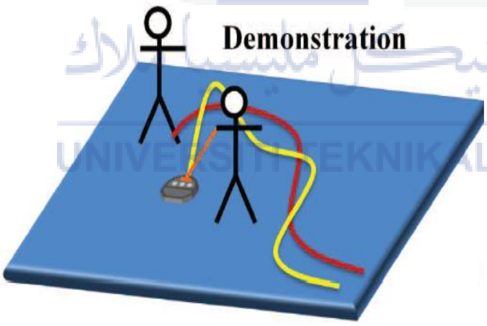


Figure 2.2: Demonstration for Cleaning Robot.

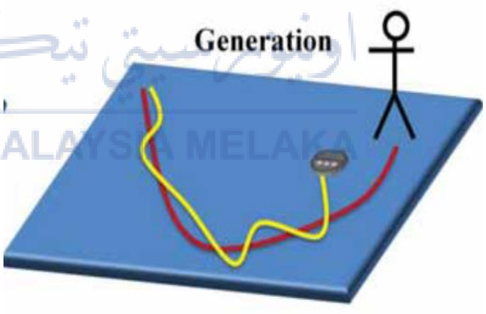


Figure 2.3: Path Learning of Cleaning Robot

2.4 Component Selection

2.4.1 Type of Distance Travel Sensor

There are many types of sensor that able to measure travel distance of an object. Sensor like infrared sensor[8], ultrasonic sensor [9], acceleration sensor[10] and encoder [11] that are able to measure the travel distance of robot.

The infrared sensor and ultrasonic sensor are using the concept like detect surrounding and calculate move distance. This kind of sensor is not suitable for difference types of room space. For acceleration sensor, it is able to calculate the distance travel according to the velocity that detected.

Besides, sensor like encoder able to measure travel distance of object according to number of step detected. In journal [11], it mention that encoder was used to estimate the size of the cleaning room. They attach collision detector and encoder on the body of cleaning robot to detect the size of room. They create two environment of room to conduct experiments: with furniture and without any furniture. For experiment in room without furniture, cleaning robot is able to measure the size without any obstacle and get better results. Encoder is able to calculate the horizontal distance and vertical distance of the room according to the value of encoder come out with some mathematical calculation and get the actual size of the room.

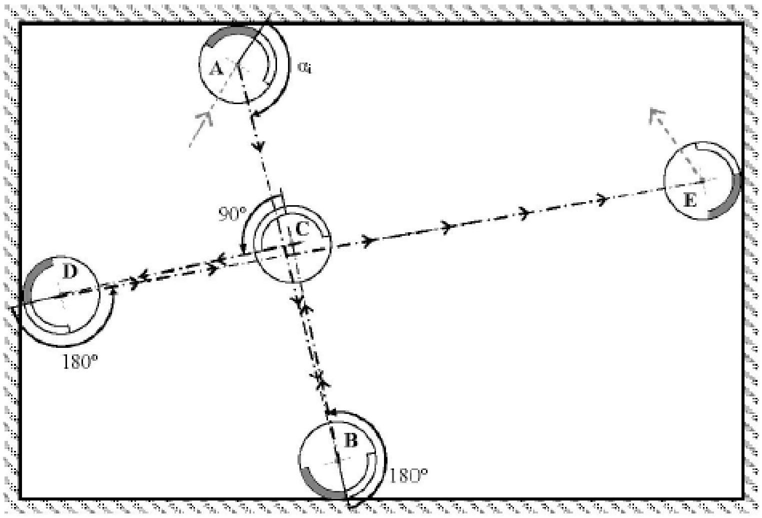


Figure 2.4: Example of “cross exploration”.

Data that collected from encoder sensor is much more accurate compare with other infrared sensor, ultrasonic sensor and acceleration sensor that mention at above. This is because encoder is measure the rotation angle on the wheel but infrared sensor, ultrasonic sensor and acceleration sensor is measure distance travel of the robot with mathematical formula. If the position of sensor attach on body of cleaning robot is not suitable, it will affect the results of whole project. Therefore, encoder sensor was selected to measure the travel distance of bathroom cleaning robot.

2.4.1.1 Digital Compass Sensor

When cleaning robot moving, the trajectory maybe does not fully follow demonstrate trajectory did by user. One of the way to tune the position or trajectory of bathroom cleaning robot to follow demonstrate trajectory is using compass sensor that able to modified error that occurs [12]. Compass Sensor able to detect angle of machine face to when user start to demonstrate trajectory until the end. Data that collected will be used to compare with actual angle of bathroom cleaning robot facing when testing prototype. Angle of bathroom cleaning robot will be tune when actual angle and desired angle are difference.

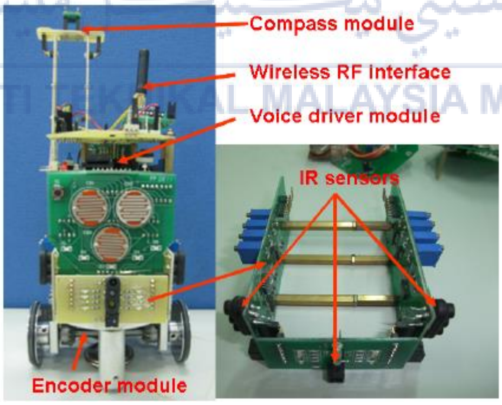


Figure 2.5: Robot with Compass Module

Navigate trajectory motion system needs digital compass and angle sensor to acquire the position and posture information of the BCR[13]. The angle of BCR facing will be calculated according to distance robot move in X-Y Axis by using triangle rule. When direction or angle contain error, two input data from encoder and compass sensor will be used by controller and control robot toward the desired path.

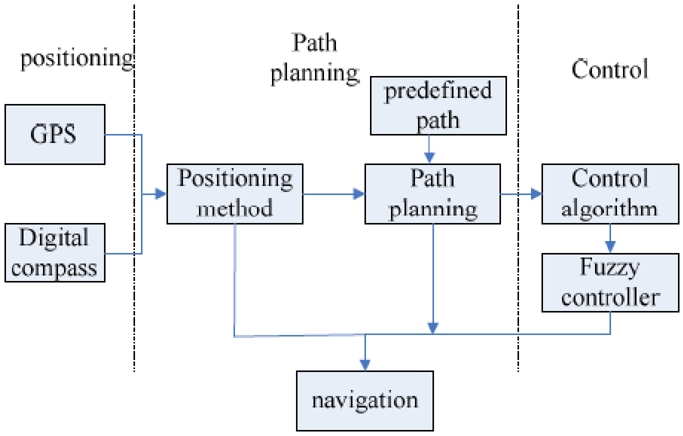


Figure 2.6: Block Diagram of Navigate Trajectory Motion System

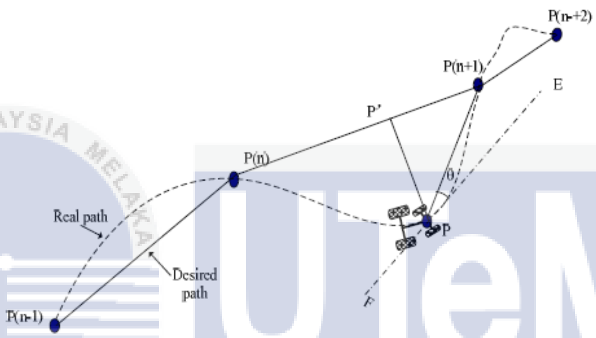


Figure 2.7: Path Tracking Method for Vehicle

Journal that wrote by students from University Malaysia Pahang (UMP) [14] show that compass module is able to track position of robot face. In their journal it is discuss to develop a straight line robot movement by using digital compass module. Although unexpected external force changing the moving path of robot, robot still able to veer back to original set-point direction to achieve smooth and stabilized movement.

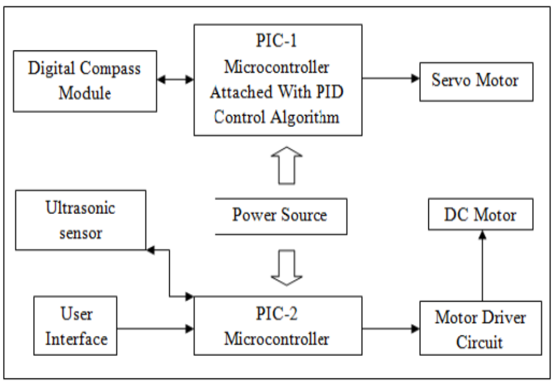


Figure 2.8: General System Block Diagram

Center of gravity is important to ensure total weight of robot is located at the center due to avoid robot skidding to another path or move in unbalanced situation. From the center of gravity, the angle was able to calculate according to X-Y plan.

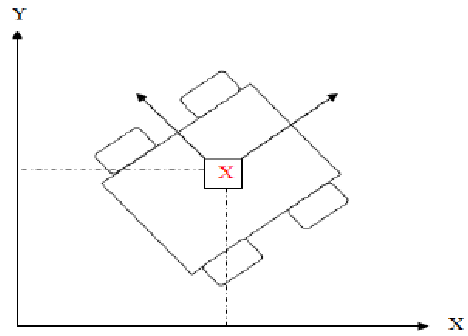


Figure 2.9: Center of Gravity of Robot

2.4.1.2 Microcontroller

Microcontroller is a device that contains processor core, memory and programmable input and output pin. There have many types of microcontroller that had been use to program robot like RCX [15], MCS-51 mircochip[12], arduino Uno [11] and PIC[14]. This type of microcontroller was previously used to conduct robot that able to control path planning. RCX and MCS-51 mircochip are seldom use by user because it is not a common microcontroller that easily get from market. For PIC and Arduino microcontroller, Arduino microcontroller has much more user compare with PIC microcontroller. This is because the programming language for arduino is using C language while PIC is using executive code in binary and hexadecimal format. C language is common languages that use to program device. Besides, information for c language also easily find from internet.

Journal wrote by Aguilar-Acevedo got discuss the way to control trajectory of DC motors [11]. From journal, they select Arduino Uno as microcontroller to control trajectory of DC gear motors with encoder. Figure 2.10 show system block diagram to control trajectory of DC motors.

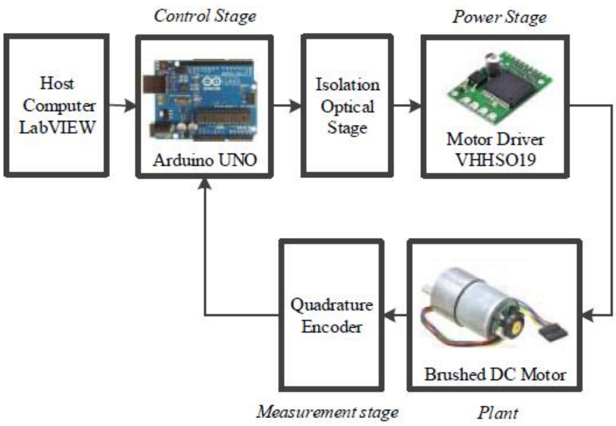


Figure 2.10: Control System of Trajectory Control

Arduino platform has been selected in [11] because of its popularity and performance. Besides, arduino board is more popular in hardware control compare with system analysis that used FPGA. This is because arduino is using serial program while FPGA is using parallel program. Arduino Uno is selected to program whole system. Arduino Uno will compute control signal received from DC motors with encoder and compass sensor. PID controller system had been use in their project to overcome the error.

The experiment that carries on in journal [11] is to compare the trajectory of simulation trajectory and actual trajectory motion. Conclusion from the writer was microcontroller software architecture allowed the evaluation of the path in real time.

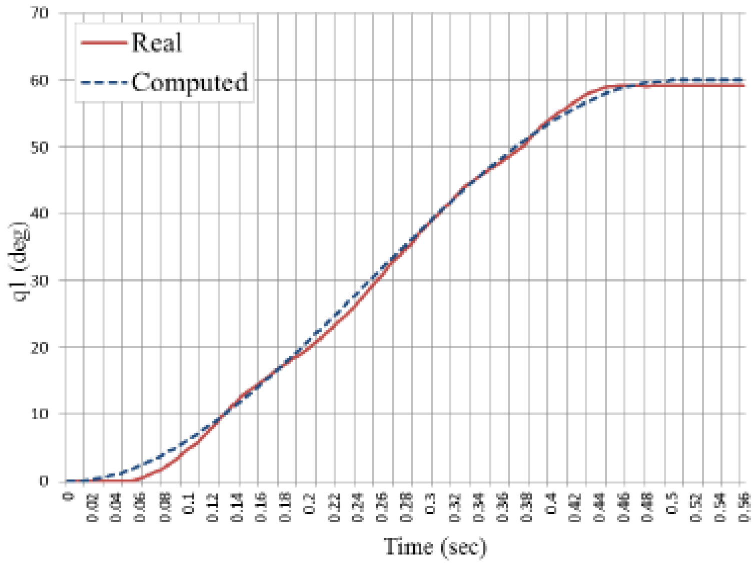


Figure 2.11: Compare Real and Computed Trajectory in Real Time

2.5 Conclusion

This project is more focus on the learning demonstration function. Besides, user will avoid the obstacle when demonstrate the trajectory of bathroom cleaning robot. Therefore, function like detect obstacle to prevent collision, controller and fix path planning will not be included for this project.

After go through many journal, components that decide to build up a prototype will be Arduino Uno as microcontroller, dc motor with encoder and digital compass sensor. Arduino Uno board as the microcontroller because from journal [11] mention that arduino platform suitable to control hardware device and it is common for research.

For the memories trajectory motion that demonstrate by human function will use DC motor with encoder and digital compass sensor. This is because encoder that attach with DC Motor able to calculate the distance travel by counting the numbers of step encoder collected. Besides, the compass sensor will be used to detect direction of cleaning robot facing. Collected data from encoder and digital compass sensor will store in memory of arduino board. The data that store in arduino's memory will be used to redo the trajectory motion that program by user. Data that collected from before and after will be used to compare the accuracy of bathroom cleaning robot trajectory.

CHAPTER 3

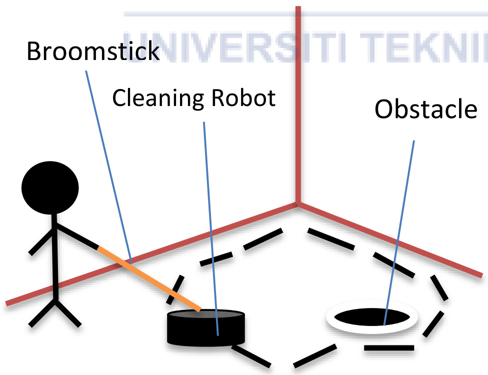
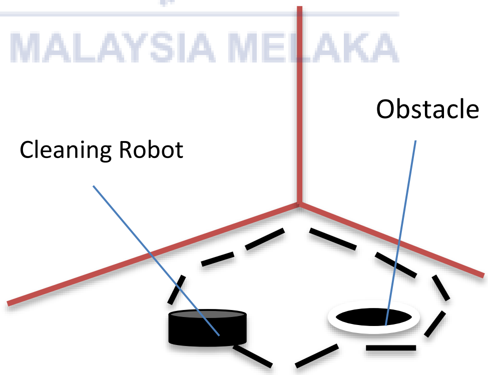
METHODOLOGY

3.1 Introduction

In chapter 3 will discuss about the design of bathroom cleaning robot (BCR). BCR motion control is difference compare to cleaning robot that exists in market and in this chapter will explain the way BCR operate. In order to meet the requirement to develop a prototype of BCR, component used will be discussed.

3.2 Method

Table 3.1: Way to Program Trajectory of Cleaning Robot

<table>
<tr>
<td>
</td>
<td>
</td>
</tr>
<tr>
<td>User Demonstrate the trajectory of Cleaning Robot</td>
<td>Bathroom Cleaning Robot will move according the demonstrate trajectory did by user automatically.</td>
</tr>
</table>

From table 2 show that the proper way to program bathroom cleaning robot that by using a broomstick. Broomstick will plug into the design hole that contains a switch before start demonstrate trajectory. Once switch was pressed, cleaning robot will start to learn or record the trajectory did by human until the broomstick was release from that switch. Trajectory that record will be store in memory of Arduino Board although turn off power supply of the robot until reset button pressed. Besides, function like obstacle avoidance will not include in this project. This is because when human demonstrate the trajectory of bathroom cleaning robot, human eye will differentiate obstacle and human will automatically avoid it. Therefore, the function of obstacle avoidance is not useful for this project.

The method to control bathroom cleaning robot will be show in figure 13 as show below.

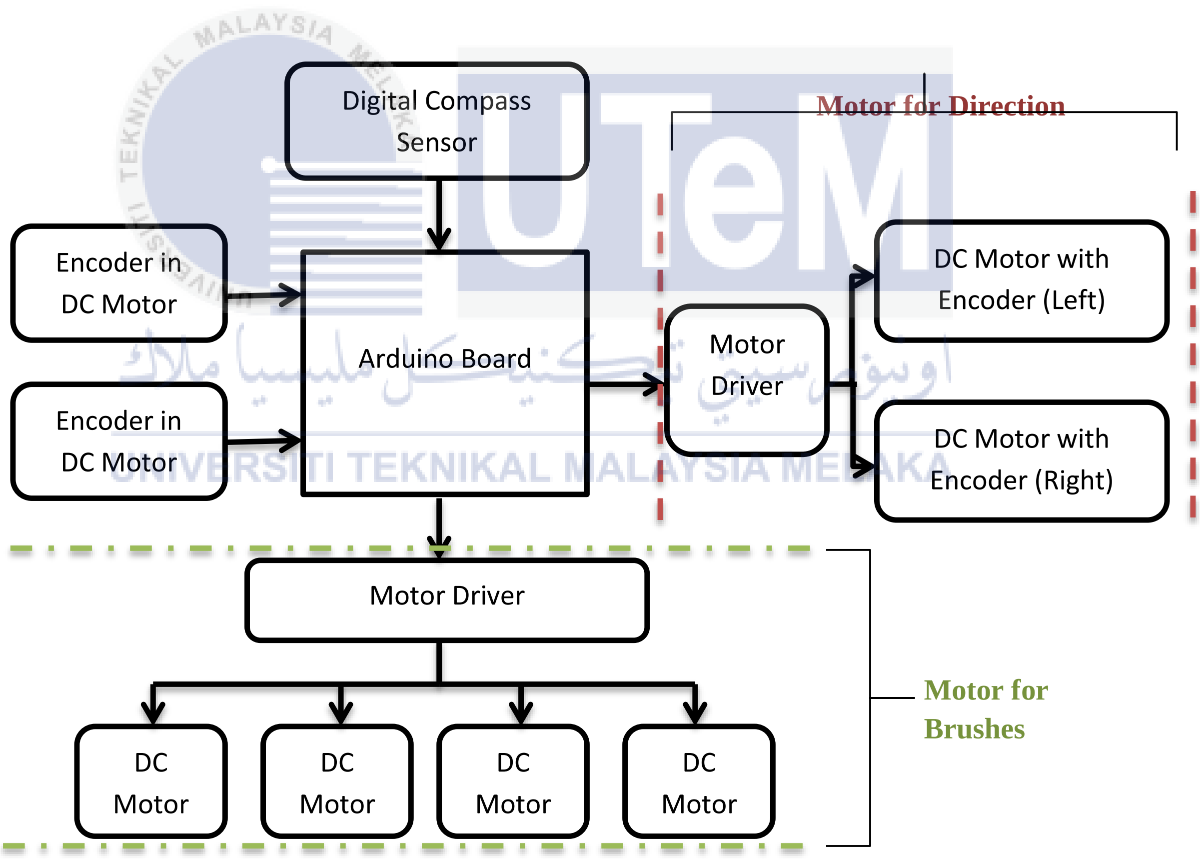


Figure 3.1: System Design for Bathroom Cleaning Robot

To create the function learns from demonstration, encoder will be used to detect the motion and angular position of wheels when user demonstrates the trajectory of cleaning robot. Digital code that occurs from encoder when user demonstrates trajectory will be send to arduino board and store into memory in board. When user demonstrate trajectory, angle of cleaning robot facing will be detect by digital compass sensor and store in memory on Arduino board too.

When bathroom cleaning robot operate or cleaning bathroom floor, Arduino Board will compare the data record in memory with actual angle of cleaning robot facing and number of step encoder rotate. When the value that detected mismatch, DC motor with encoder will adjust the facing angle and encoder steps by rotate the wheel forward and backward until it match the value that program. Therefore, encoder and digital compass sensor value that store in memory can assume as desired input for close-loop control system.

Method to calculate the angle when using DC motor with encoder will be show in figure 3.2 as below.

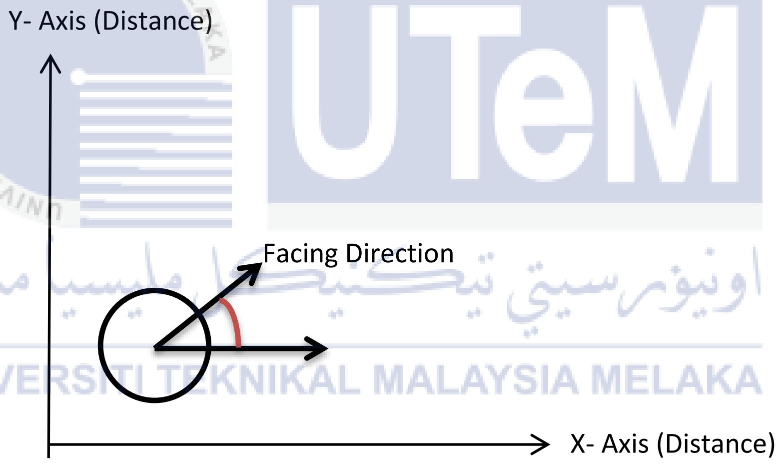


Figure 3.2: X-Y Axis Plan

Formula used to calculate distance travel:

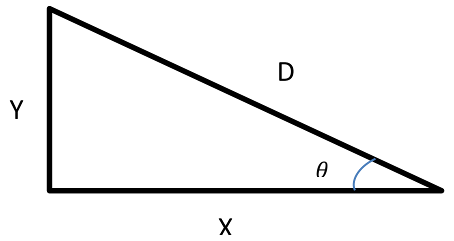


Figure 3.3: Trigonometry Method

$$X^2 + Y^2 = D^2 \quad (1)$$

$$D = \sqrt{X^2 + Y^2} \quad (2)$$

Where $X = distance\ of\ X - Axis$

$Y = distance\ of\ Y - Axis$

$D = displacement\ of\ robot\ move$

Formula used to calculate angle:

$$\tan \Theta = \frac{Distance\ of\ Y - Axis}{Distance\ of\ X - Axis} \quad (3)$$

$$\Theta = \tan^{-1} \frac{Distance\ of\ Y - Axis}{Distance\ of\ X - Axis} \quad (4)$$

Method to calculate moving distance of encoder:

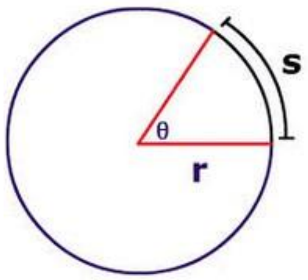


Figure 3.4: Angular Distance

Get the degree of wheel rotate until one step of encoder detected. With the angle, calculate the distance of wheel rotate by using the equation list below:

$$S = r\theta \quad (3)$$

Where, S = Arc Length

r = radius of the circle

θ = angle in rad

3.3 Flow Chart

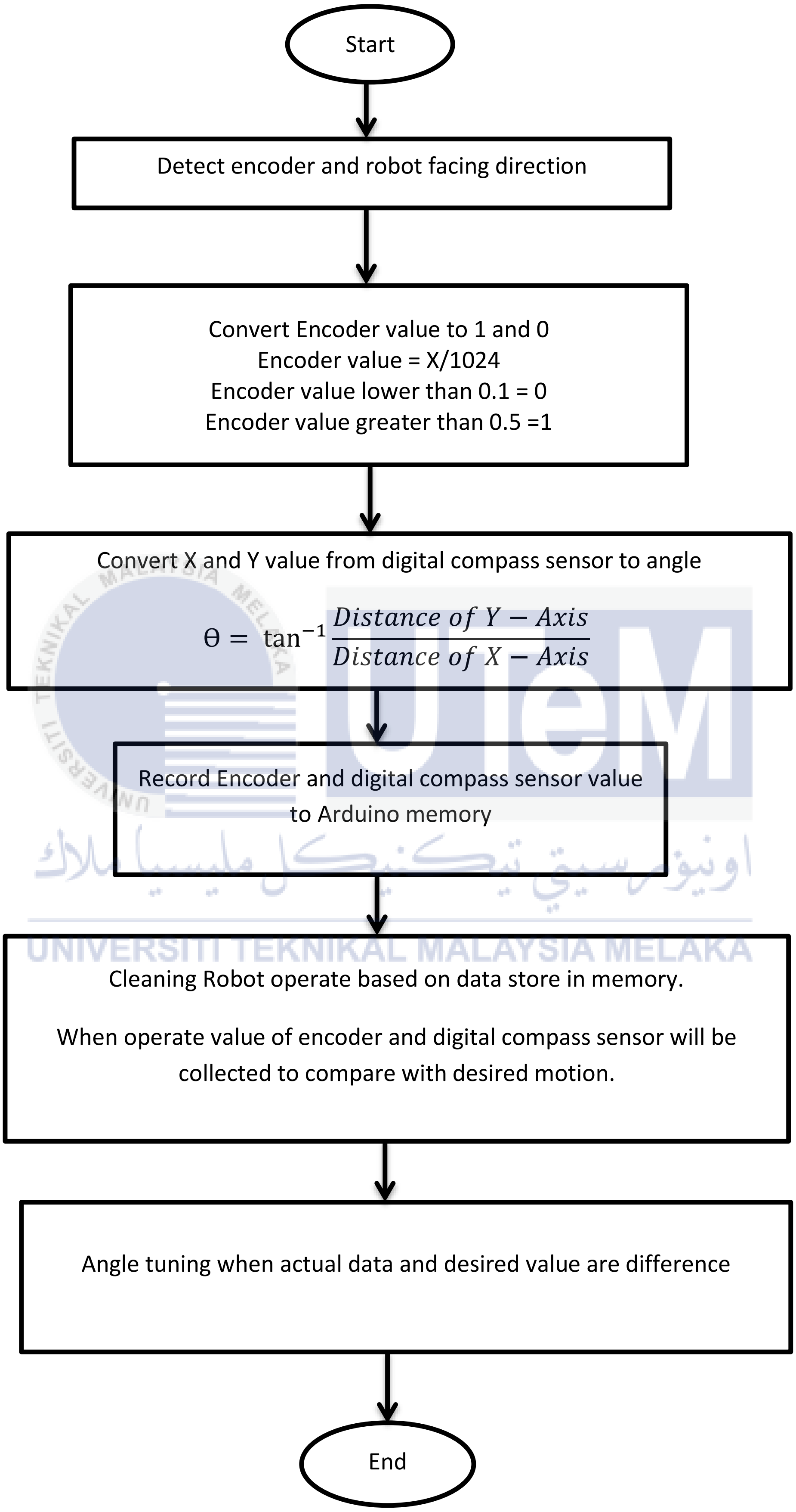


Figure 3.5: Flow Chart of Prototype

3.4 Hardware Design

Material used to build up prototype will be plastic, rubber and metal that easily gets from market. Drawing software like SolidWork will be used to design bathroom cleaning robot as show in figure 3.6 as below.

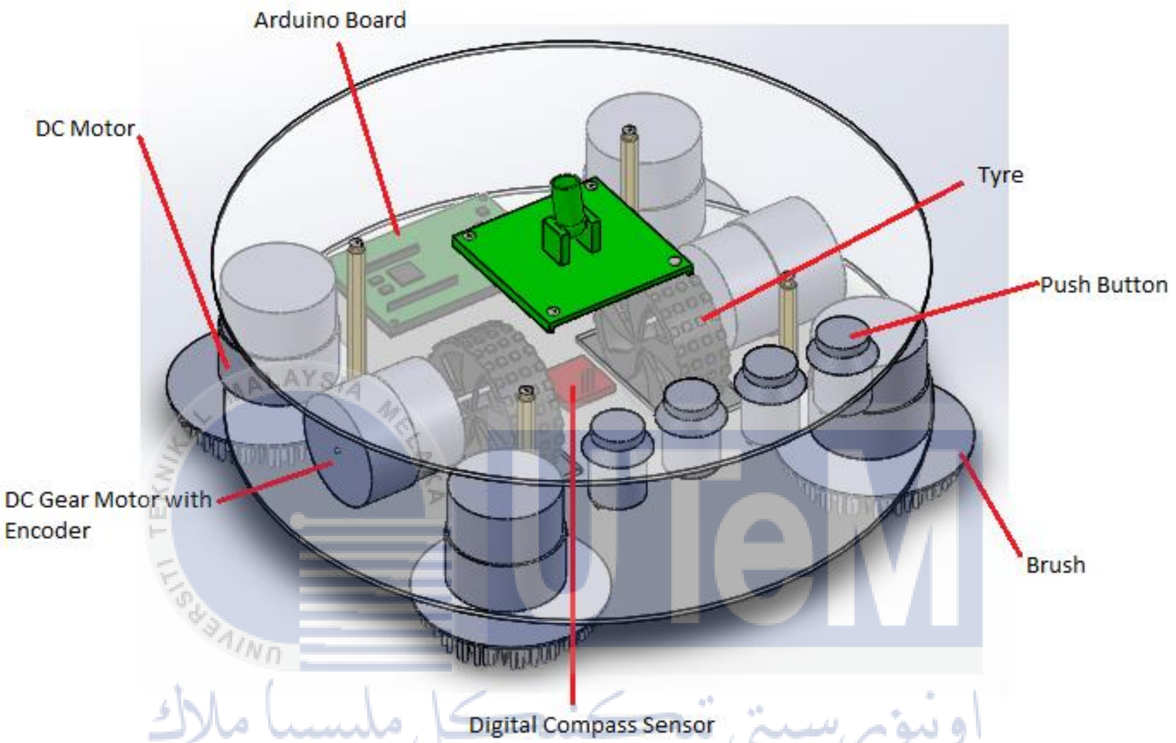


Figure 3.6: Prototype Design Drawing

The prototype design will be in round shape to prevent collision between bathroom cleaning robot and wall of bathroom. Component that included are round shape acrylic board, digital compass sensor, push button, Arduino board, 4 DC motors, 2 tires and 2 DC Motor with encoder. The body size of bathroom cleaning robot prototype will be 25cm diameter long. The hole on the top of cleaning robot show in figure 3.7 is slot for broomstick to push the cleaning robot. By using broomstick, human is able to program bathroom cleaning robot like brushing floor.

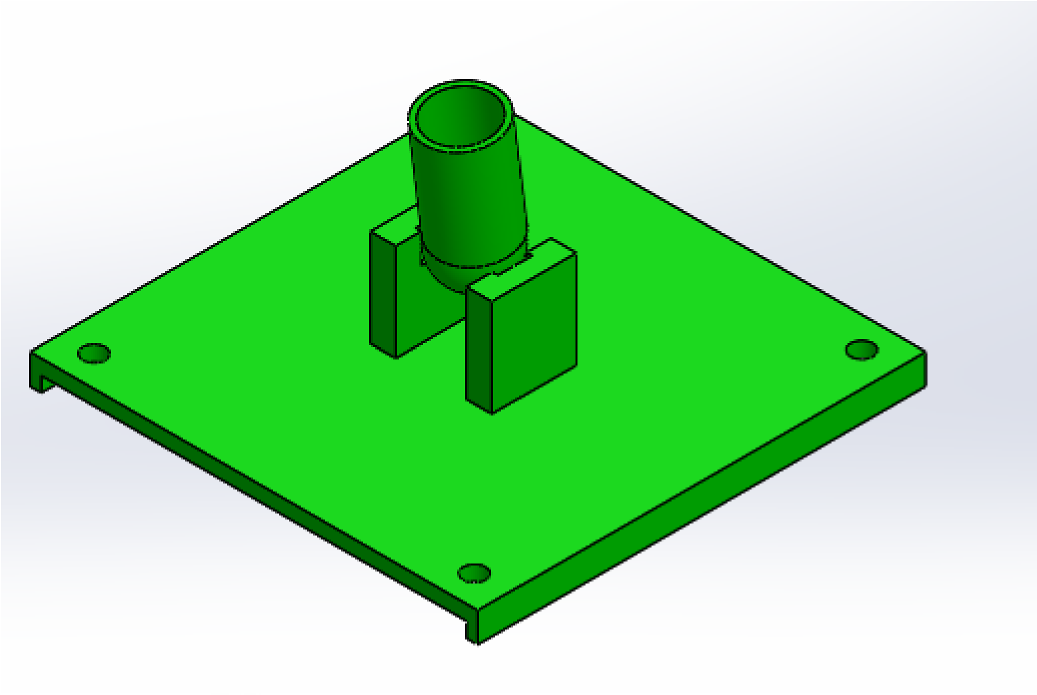


Figure 3.7: Slot for Broomstick

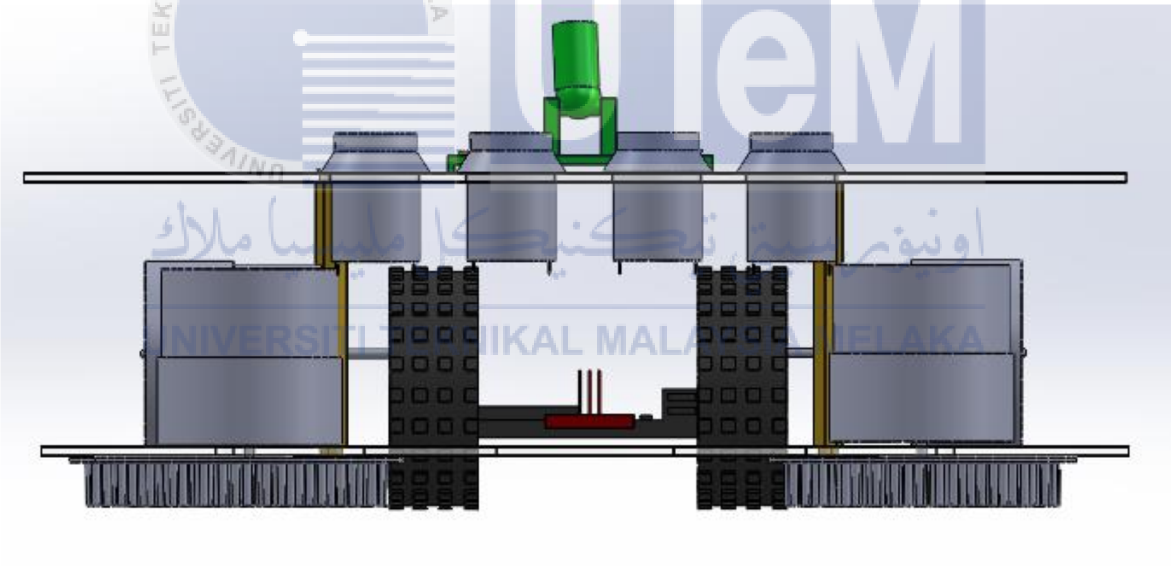


Figure 3.8: Front View of Prototype Design

Figure 3.9: Top View of Prototype Design

As show in figure 3.8, the horizontal level of tires and brush are the same. This is to ensure that the tires have enough grip force when brushes on dc motor rotate. Four brushes will place at four side of BCR with DC Motor to clean the surface of floor and corner of bathroom. Arduino Board and digital compass sensor will place at the front cleaning robot. Position to place digital compass sensor is important to detect the direction bathroom cleaning robot facing. Therefore, digital compass sensor will be place in between of both tyres to reduce the error when adjust the direction using DC Motor with encoder.

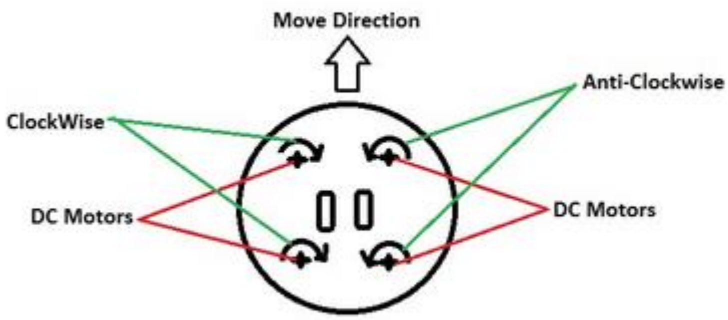


Figure 3.10: Brush Rotate Direction

3.5 Prototype

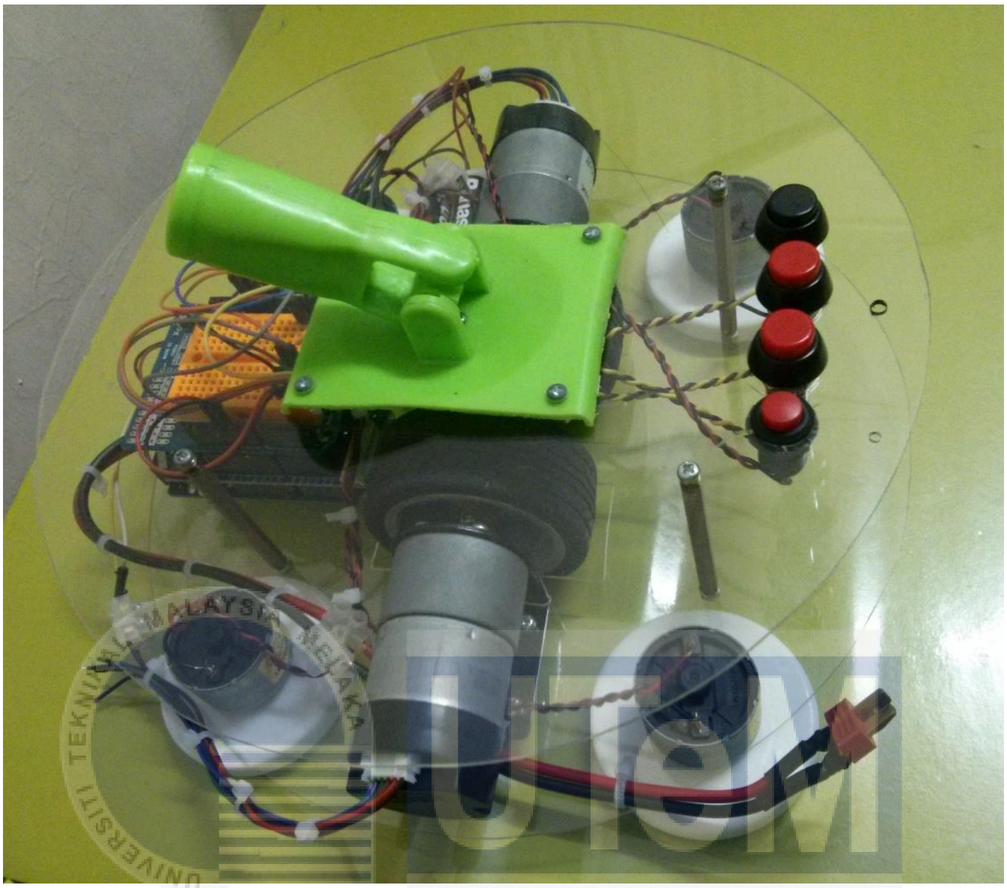


Figure 3.11: Prototype of Bathroom Cleaning Robot (BCR)

Prototype of this project was show in Figure 3.11. Acrylic was used to design this prototype to let use look through the body of prototype and understand way of prototype operate for this project. Three push button and one latching button are attach on the surface. Function of that three push button will be Start button, Record button and Replay button. While latching button will be on and off button for the small dc motor that attach with brush.

Li-Po Battery will be used as external power supply for DC gear motor with encoder. This is because DC gear motor with encoder need 12V to operate it and Arduino only able to supply maximum 5V. Circuit for four small DC motor that attach with brush are separate with the main circuit. Therefore, the power supply of the DC motor also separate with the main circuit. Two 1.5V AA battery will be used to run the motor.

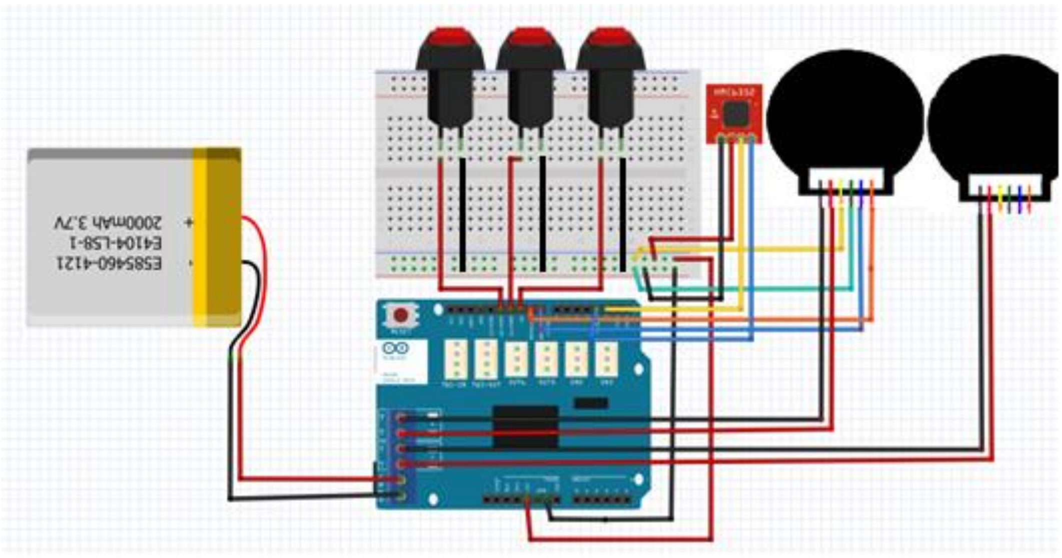


Figure 3.12: Main Circuit to Operate Prototype

Figure 3.12 show the circuit of the prototype. Arduino motor driver shield will attach together with Arduino Leonardo. Two DC gear motor with encoder will connect to the motor driver so that Pulse-Width-Modulation (PWM) of motor can be control by using motor driver shield. For digital compass sensor, SDA (data line) will connect to pin 2 while SCL (clock line) will connect to pin 3 of the board. Pin 10 will be START button, pin 11 will be RECORD button and pin 12 will be REPLAY button to select record and replay mode for the prototype. Basic parallel circuit used to control the rotation of dc motor with brush was shown in Figure 3.13.

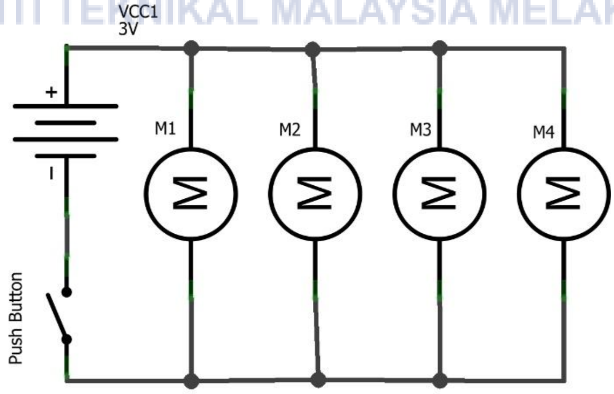


Figure 3.13: Circuit Diagram for Motor with Brush

3.5.1 Weight of Hardware Design

Weight of bathroom cleaning robot is one of the tasks to prevent wheel slipping. Body weight of cleaning will be show in table 3.2 at below.

Table 3.2: Total Weight of Bathroom Cleaning Robot

Part	Unit	Mass Per Unit (g)	Total Mass (g)
Acrylic Board	2	20	40
Arduino Board	1	28	28
Digital Compass Sensor	1	12	12
DC Motor	4	28	112
DC Motor with Encoder	2	110	220
Wheel	2	23	46
Brush	4	21	84
Battery	1	350	350
			892

Formula to calculate force:

$$\text{Force, } F = mg \quad (4)$$

Where F = force (N)

m = mass in kg

g = gravitational force in ms^{-2}

Let g = 9.81 ms^{-2}

F = (0.892) (9.81) = 8.75 N

Total Weight of bathroom cleaning robot will be around 8.75 N as design. With this weight, body of bathroom cleaning will being pull downward by downward force and increase contact surface between brush and floor. Besides, this grip on wheel will increase as the weight of prototype increase.

3.6 Component and Function

3.6.1 DC Motor with Encoder

Figure 3.14: DC Motor with Encoder

DC Motor with encoder is a programmable dc motor. This kind of motor mostly uses to detect the position or angular motion of shaft when rotate. Encoder that attach together with DC Motor is able to convert the position or angular motion into digital code.

3.6.2 Compass Sensor

Figure 3.15: Compass Sensor

Compass sensor combines x-y axis magneto-resistive sensor that support analog and digital circuits. Compass sensor able to detect bathroom cleaning robot face position and data that collect will be store in memory for future use.

3.6.3 DC Motor

Figure 3.16: DC Motor

Normal DC Motor will be used to rotate the brush under bathroom cleaning robot. Four DC Motors will start to operate when start button on surface of cleaning robot being pressed.

3.6.4 Arduino Board

Figure 3.17: Arduino Uno

Arduino Board is a common used microcontroller board for learner. Programming language that use to program board is C language. Therefore, it is easy for learner to program the board.

3.6.5 Motor Driver

Figure 3.18: DC Motor Driver

12 Volts DC motor driver able to operate DC motor that need 12 Volts to operate. Power supply from the Arduino board only 5 Volts but the DC Motor needs 12 Volts to operate. Therefore, external power supply will be needed to start up DC Motors. Besides, this motor driver able to control PWM of DC Motor with encoder when operate.

3.7 Control Algorithm of Bathroom Cleaning Robot

3.7.1 Open-Loop System (Method 1)

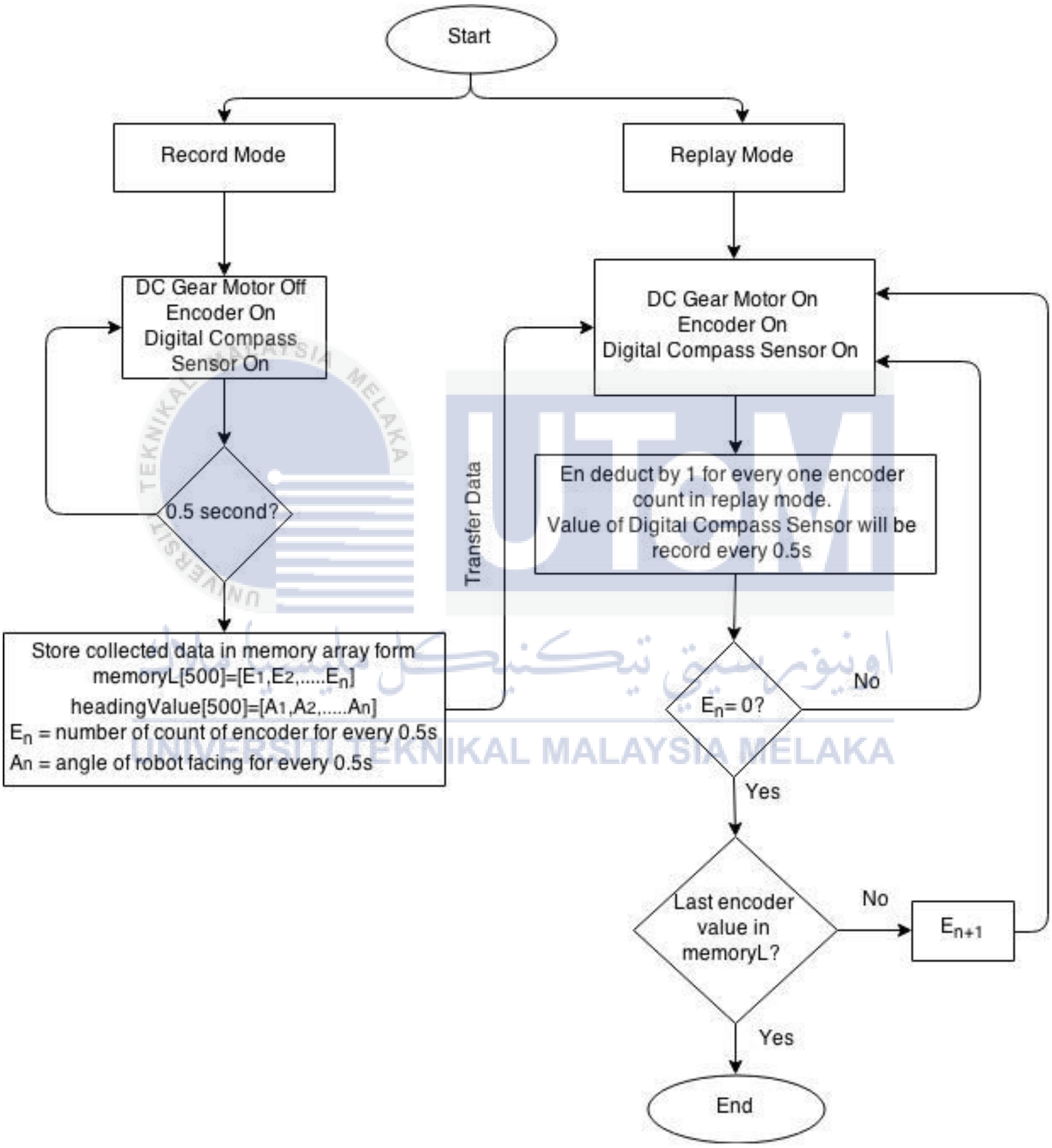


Figure 3.19: Control Algorithm for Open-Loop System

3.7.2 Closed-Loop System (Method 2)

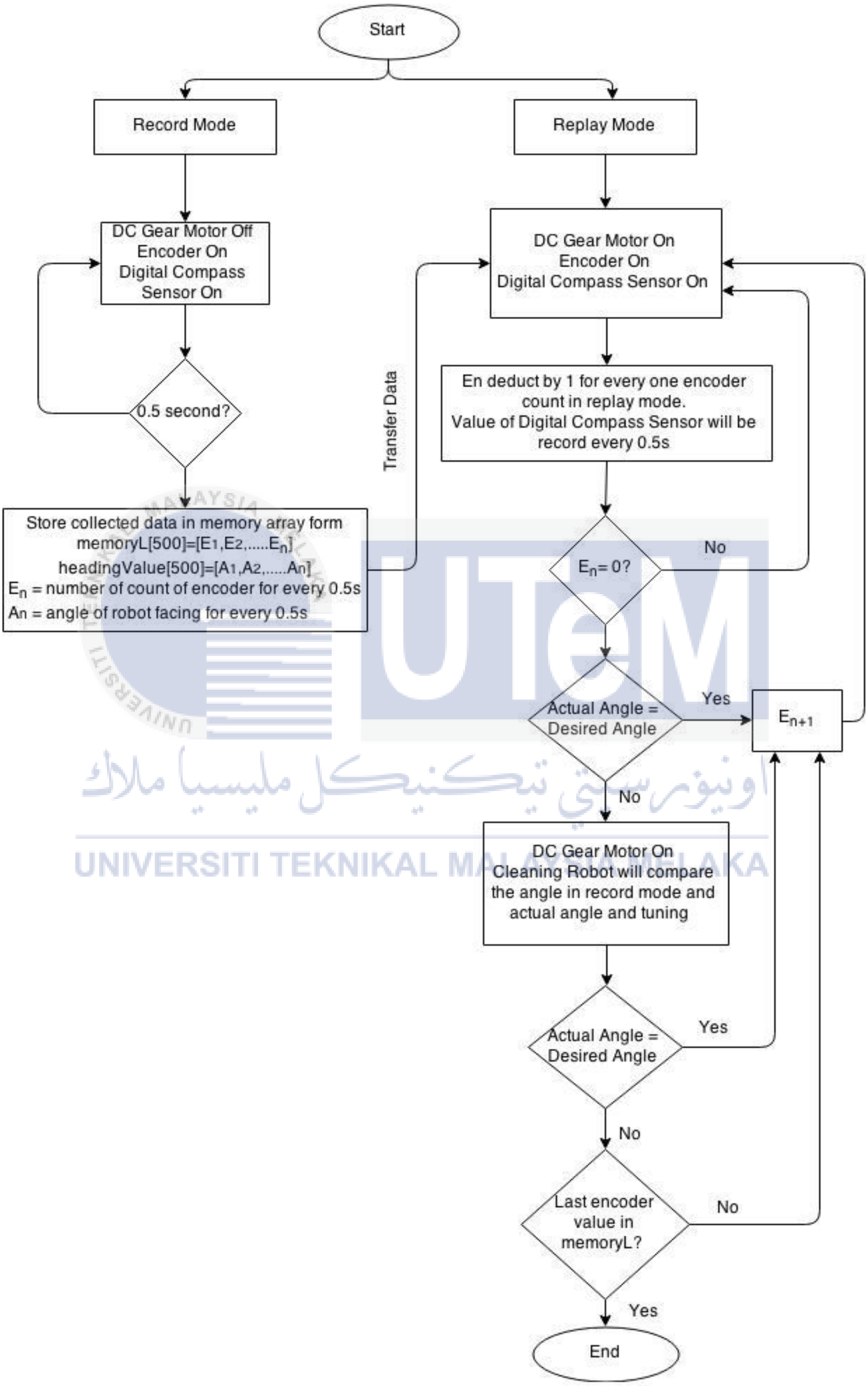


Figure 3.20: Control Algorithm for Closed-Loop System

3.8 Experiment 1

3.8.1 Wheel Move Distance

This experiment is to test the distance of wheel move then encoder rotate. From the distances that calculate, path error will be count to achieve one part of the third objective for this project. Microcontroller like Arduino will be used to program the dc motor with encoder. Protractor will be used to measure the angle of wheel rotate.

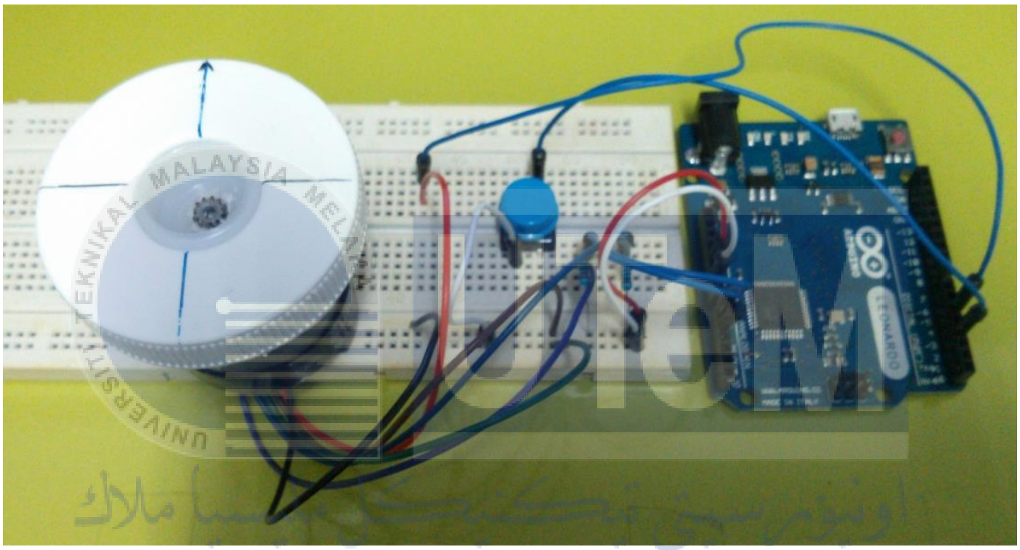


Figure 3.21: Experiment Prototype

3.9 Experiment 2

3.9.1 Digital Compass Sensor

In this experiment, digital compass sensor will be test to detect the direction of bathroom cleaning robot facing. The purpose of this experiment is to analysis the accuracy of the compass sensor and understand the way of digital compass function.

Digital compass sensor will connect with Arduino so that data of digital compass sensor is able to collect and the data will use for analysis. Compass sensor will be put on protractor and facing 90 degree. After that change the direction of compass sensor facing to 30 degree, 60 degree and 0 degree. With the protractor, actual degree of compass sensor facing and angle calculate by Arduino will be compare.

Trigonometry formula will be program into Arduino for calculation.

$$\tan \Theta = \frac{Distance\ of\ Y{-}Axis}{Distance\ of\ X{-}Axis}$$

$$\Theta = \tan^{-1}\frac{Distance\ of\ Y - Axis}{Distance\ of\ X - Axis}$$

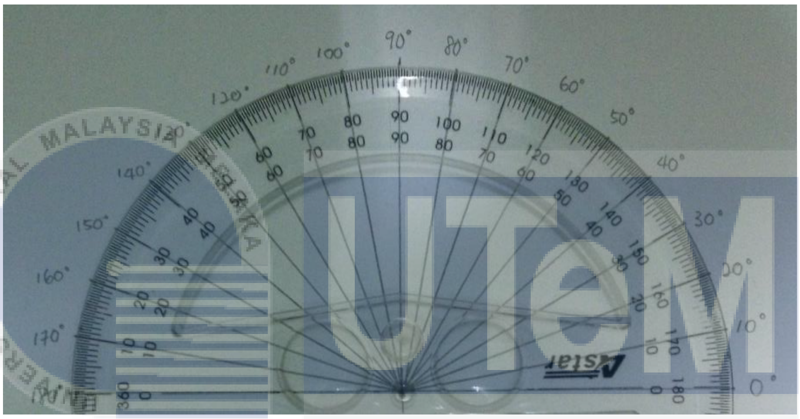


Figure 3.22: Draw Angle on Paper

Difference angle will be draw on a piece of plane paper by using protractor as show in Figure 3.22.

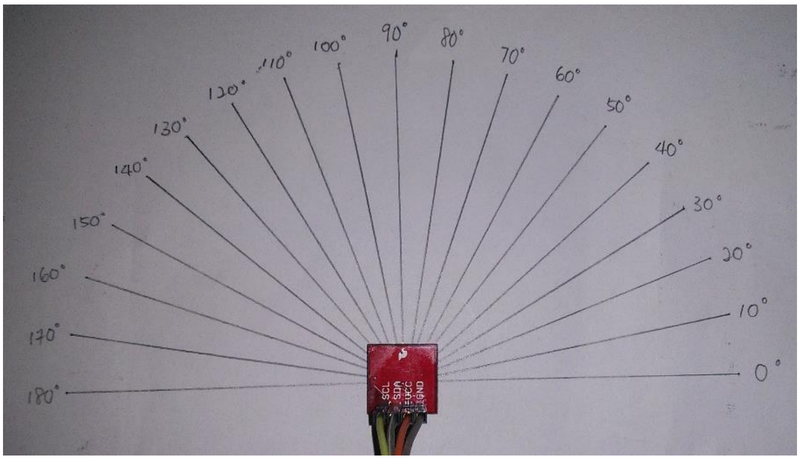


Figure 3.23: Digital Compass Sensor on Paper

Digital compass sensor that connect with Arduino will be place at the center of draw protractor on the paper. The center point of the chip should be on the center point of the draw protractor. This is because rotation point of the digital compass sensor is at the center point of the chip. This is to ensure angle that detected by compass sensor is accurate when doing experiment.

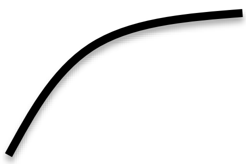
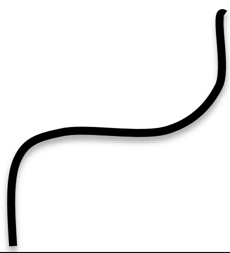
3.10 Experiment 3

3.10.1 Method to Analysis Trajectory Motion

The main analysis for this project is to analysis trajectory of cleaning robot when operate in replay mode. Three difference types of shape will be done to detect accuracy of bathroom cleaning robot to achieve user demonstrate trajectory.

Three difference types of trajectory shape that conduct in project will be straight line, curve and S shape.

Table 3.3: Types of Shapes

Straight Line	
Curve	
S Shape	

Two difference method will be used to analysis the performance of bathroom cleaning robot (BCR). Method 1 will be an open-loop system while method 2 will be a closed-loop system.

For method 1 and method 2, the way to record the number of encoder steps and direction of BCR is the same. The only difference between this two methods is method 1 use the data that store in the Arduino memory directly to operate the BCR without any adjustment while method 2 will make some adjustment about the direction BCR facing with the desired angle store in memory and actual angle of BCR face. The feedback of method 2 will be the facing angle in real time.

In this experiment, black colour sticker will stick on the floor according to the desired trajectory. A piece of Majong paper will be place on the surface of the black colour line. This is to let the BCR draw out the trajectory in replay mode on the Majong paper. After that, user will demonstrate the trajectory according to the shape of the black sticker so that BCR will record the number of encoder step and direction in record mode and redo it in replay mode.

According to the desired trajectory and actual trajectory that draw by BCR, the data about number of encoder step count and direction of BCR facing will be collect and use it to analysis BCR. Besides, comparison between actual trajectory that draw by using method 1 and method 2 will be make. From the error that calculate from the data, mean error and root mean square error (RMSE) will be used to compare error between open-loop and closed-loop system.

CHAPTER 4

RESULT AND DISCUSSION

4.1 Experiment 1

This experiment is one of the minor part to achieve the objective number 3 for this project that is path error when cleaning robot operate. In this experiment, the length of wheel travel in one complete cycle and number of step of encoder will we detected. Diameter of wheel use in this experiment will be 6.4cm. To calculate the distance move by the wheel for every step of encoder detected, formula $s = r\theta$ had been use.

After go through the experiment, it show that one complete cycle 6.4cm wheel rotate contain 250 numbers of step. With the results get from the experiment and do some calculation by using formula, $s = r\theta$, it show that one step of encoder is equal to 1.44 degree or 0.025 radian and 0.08cm. Therefore, bathroom cleaning robot will travel 20.11cm with one complete cycle.

One complete cycle with 360 degree contain 250 numbers of step of encoder value show that distance travel can be control very precise. The data table for this experiment will be attached in the appendix of this report.

4.2 Experiment 2

This experiment is to analysis the digital compass sensor able to detect the direction in proper way or not. Objective number 3 mention that direction of bathroom cleaning robot facing is one of the criteria that will be include.

Results that collected will be collected and put into table as show below. Three difference types of angle like 30 degree, 60 degree and 90 degree will be test in this experiment.

Table 4.1: Digital Compass Sensor Rotate 30, 60 and 90 Degree

Number of Set	Desired Angle (°)	Actual Angle (°)	Error (°)	Desired Angle (°)	Actual Angle (°)	Error (°)	Desired Angle (°)	Actual Angle (°)	Error (°)
1	30	31	1	60	61	1	90	90	0
2	30	30	0	60	60	0	90	90	0
3	30	30	0	60	60	0	90	90	0
4	30	29	1	60	59	1	90	89	1
5	30	30	0	60	60	0	90	90	0
6	30	30	0	60	60	0	90	91	1
7	30	30	0	60	60	0	90	90	0
8	30	30	0	60	59	1	90	90	0
9	30	31	1	60	60	0	90	89	1
10	30	29	1	60	59	1	90	90	0
Total Error			4			4			3
Mean Error			0.4			0.4			0.3
RMSE			0.6325			0.6325			0.5477

Formula:

Mean Error = $\left(\frac{Total\ Error}{Number\ of\ Set}\right)$

Root Mean Square Error, RMSE = $\sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}$

From the Root Mean Square Error (RMSE) show by 3 difference angles that are 0.6325 for both 30 and 60 degree and 0.5477 for 90 degree. Overall of the RMSE value of three difference angle are below 1. The lower the RMSE value mean the lower the error will occur when operate. With the RMSE value that less than 1 show that the digital compass sensor is very accurate and suitable to use it to calculate BCR facing direction.

4.3 Experiment 3

In this experiment, it will include two different programming method to test the error of the prototype. The method 1 is using the open loop method while method 2 is using close loop method. In record mode, every 0.5 seconds that number of step detected and degree BCR facing will be recorded.

Method 1 (Open-Loop System):

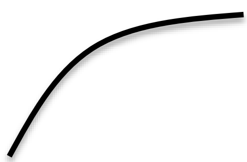
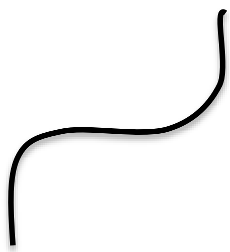
Bathroom Cleaning Robot (BCR) use the data that collected in record mode and run the data without make any changes or correction in replay mode.

Method 2 (Closed-Loop System):

For record mode, the method use will be the same as method one. Store the data into the Arduino microcontroller. The difference between method one and method two are method 2 compare the degree store in record mode and replay mode. When the BCR moving from one point to another point, it will compare the designed degree and actual degree. If the degree is not the same, BCR will automatically adjust its facing position until it actual angle is the same as designed angle only the BCR will move to another point.

To test the error of the BCR, three difference types of trajectory had been made, that are straight line, curve and S shapes.

Table 4.2: Three Difference Kind of Trajectory Shape.

Straight Line	 Curve	 S Shape

4.3.1 Method 1 (Open-Loop System)

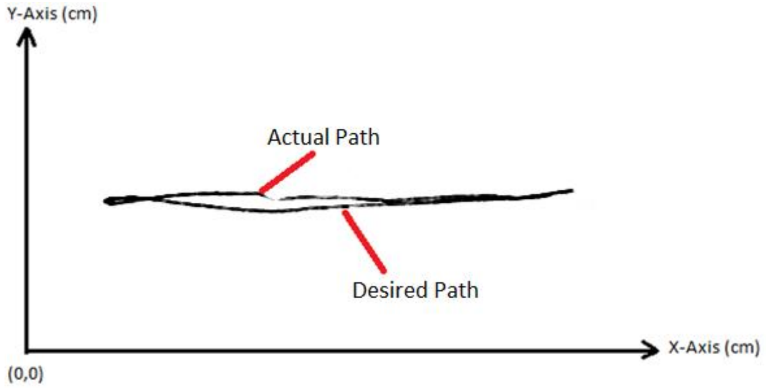


Figure 4.1: Straight Line with Method 1

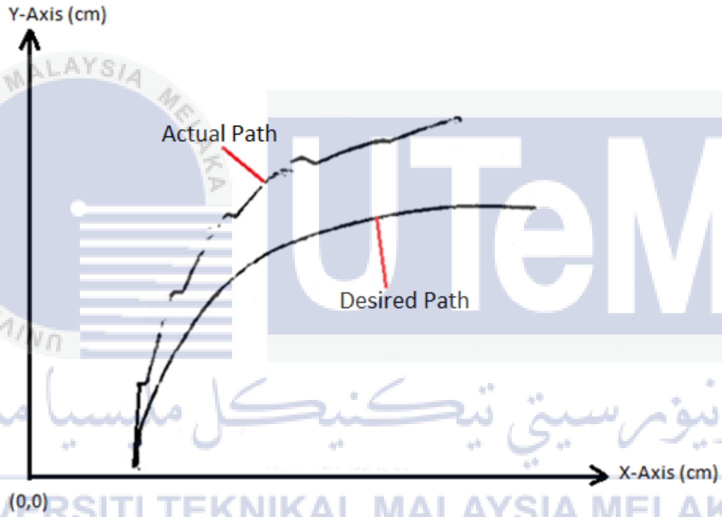


Figure 4.2: Curve Line with Method 1

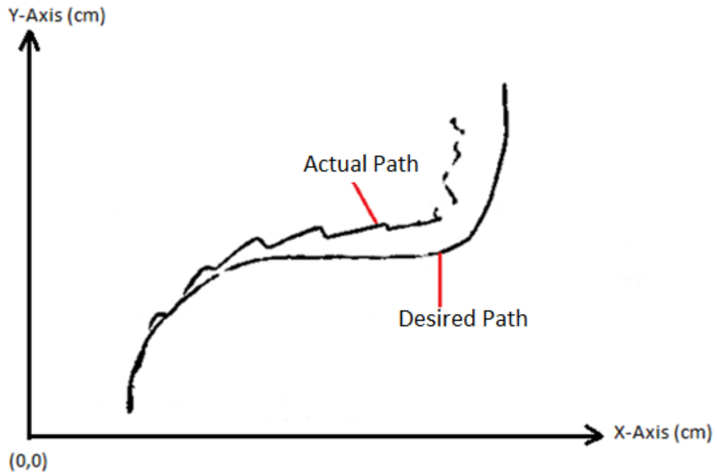


Figure 4.3: S Shape with Method 1

In method 1, three difference type of trajectory ran four times to get the data. From the data, the error of number of step and angle of BCR facing will be discuss. From the figure as show above, the ending point for designed path and actual path are far from each other. For the angle, Figure 4.2 and 4.3 shows that without the feedback of the angle, BCR will move out of the designed trajectory but it still will show out the shape of the trajectory in replay mode. The analysis about the number of step and angle will be discuss at below.

4 3.1.1 Straight Line

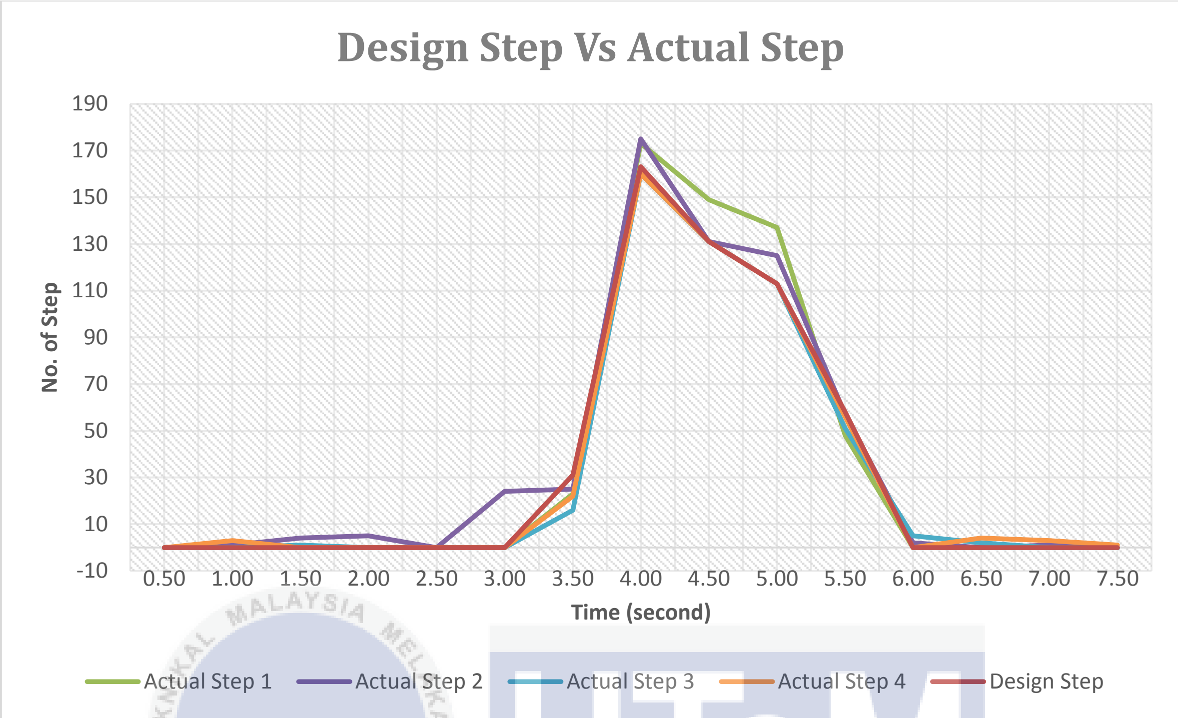


Figure 4.4: Number of Step Value Detected for Every 0.5 Second for Straight Line

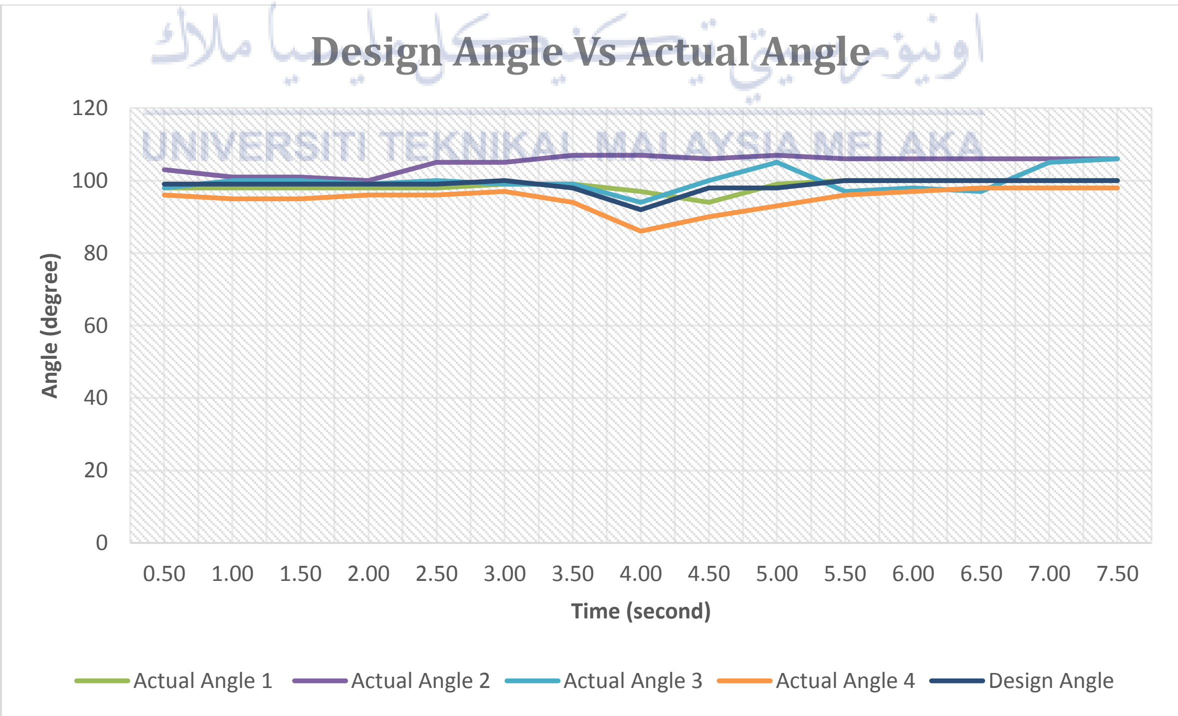


Figure 4.5: Angle of BCR Face Every 0.5 Second Along Straight Line

<u>Error of Number of Step (Method 1 – Straight Line)</u>

Table 4.3: Error of Number of Step (Straight Line)

Time	Design Step	Actual Step 1	Error 1	Actual Step 2	Error 2	Actual Step 3	Error 3	Actual Step 4	Error 4
0.50	0	0	0	0	0	0	0	0	0
1.00	0	0	0	1	1	0	0	3	3
1.50	0	0	0	4	4	1	1	0	0
2.00	0	0	0	5	5	0	0	0	0
2.50	0	0	0	0	0	0	0	0	0
3.00	0	0	0	24	24	0	0	0	0
3.50	31	23	-8	25	-6	16	-15	22	-9
4.00	163	173	10	175	12	160	-3	160	-3
4.50	131	149	18	131	0	131	0	131	0
5.00	113	137	24	125	12	113	0	113	0
5.50	58	48	-10	57	-1	51	-7	56	-2
6.00	0	0	0	2	2	5	5	0	0
6.50	0	0	0	0	0	2	2	4	4
7.00	0	0	0	1	1	0	0	3	3
7.50	0	0	0	0	0	0	0	1	1
Total Error			70		68		33		25
Mean Error			4.6667		4.5333		2.2		1.66667

Overall Mean Error = 3.26667

Root Mean Square Error, RMSE = $\sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}$

RMSE = $\sqrt{\frac{7.2622}{4}}$ = 1.3474

After go through some of the calculation, it show that the overall mean error after go through 4 times same replay mode is 3.26667 number of steps. The mean error is mean that every 0.5 second the average error should be around 3.26667 number of steps. One number of step is equal to 0.08cm. Therefore, the error when BCR travel along the path will have 0.26cm error per 0.5 seconds

<u>Error of Facing Angle (Method 1 - Straight Line)</u>

Table 4.4: Error of Facing Angle (Straight Line)

Time	Design Angle	Actual Angle 1	Error 1	Actual Angle 2	Error 2	Actual Angle 3	Error 3	Actual Angle 4	Error 4
0.50	99	98	-1	103	4	98	-1	96	-3
1.00	99	98	-1	101	2	100	1	95	-4
1.50	99	98	-1	101	2	100	1	95	-4
2.00	99	98	-1	100	1	99	0	96	-3
2.50	99	98	-1	105	6	100	1	96	-3
3.00	100	99	-1	105	5	99	-1	97	-3
3.50	98	99	1	107	9	99	1	94	-4
4.00	92	97	5	107	15	94	2	86	-6
4.50	98	94	-4	106	8	100	2	90	-8
5.00	98	99	1	107	9	105	7	93	-5
5.50	100	100	0	106	6	97	-3	96	-4
6.00	100	100	0	106	6	98	-2	97	-3
6.50	100	100	0	106	6	97	-3	98	-2
7.00	100	100	0	106	6	105	5	98	-2
7.50	100	100	0	106	6	106	6	98	-2
Total Error			17		91		36		56
Mean Error			1.13333		6.06667		2.4		3.7333

Overall Mean Error = 3.3333

Root Mean Square Error, RMSE = $\sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}$

RMSE = $\sqrt{\frac{13.3422}{4}}$ = 1.8264

For the error of BCR facing angle every 0.5 seconds, it show that every 0.5 seconds will have average 3.3333 degrees difference compare with the designed angle that recorded in the record mode. The angle of BCR facing is very important in this project. If the BCR facing to the wrong direction or angle and the BCR will travel to the wrong path. The greater the number of the step, the greater the error will be.

4.3.1.2 Curve

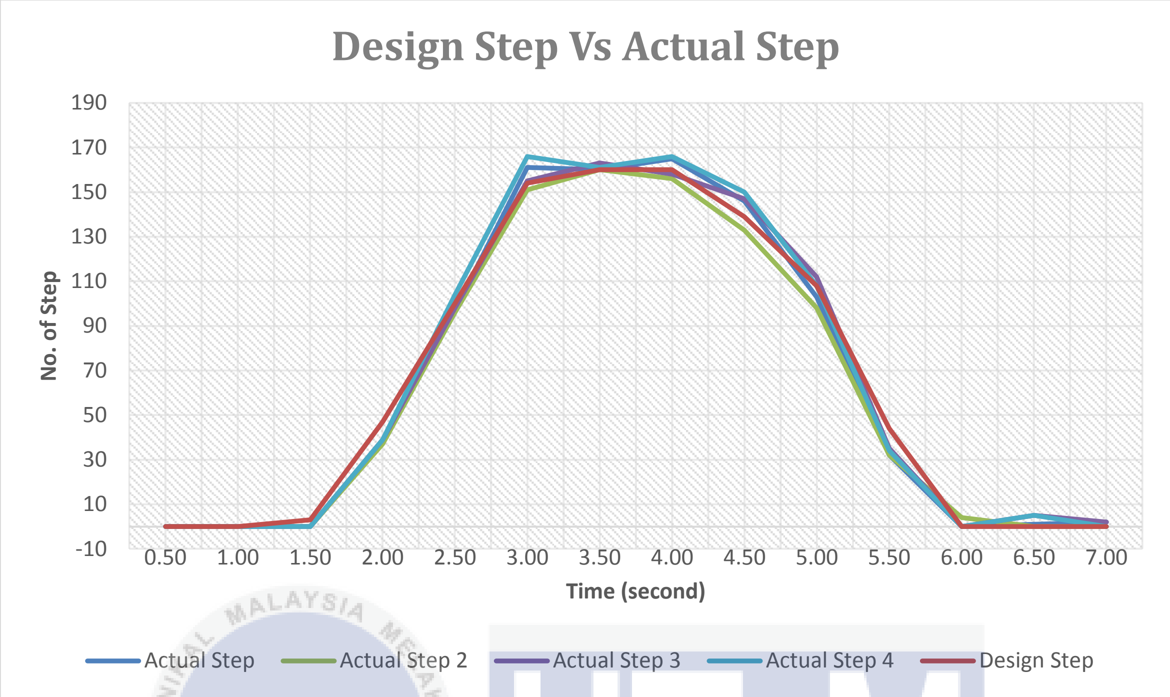


Figure 4.6: Number of Step Value Detected for Every 0.5 Second for Curve

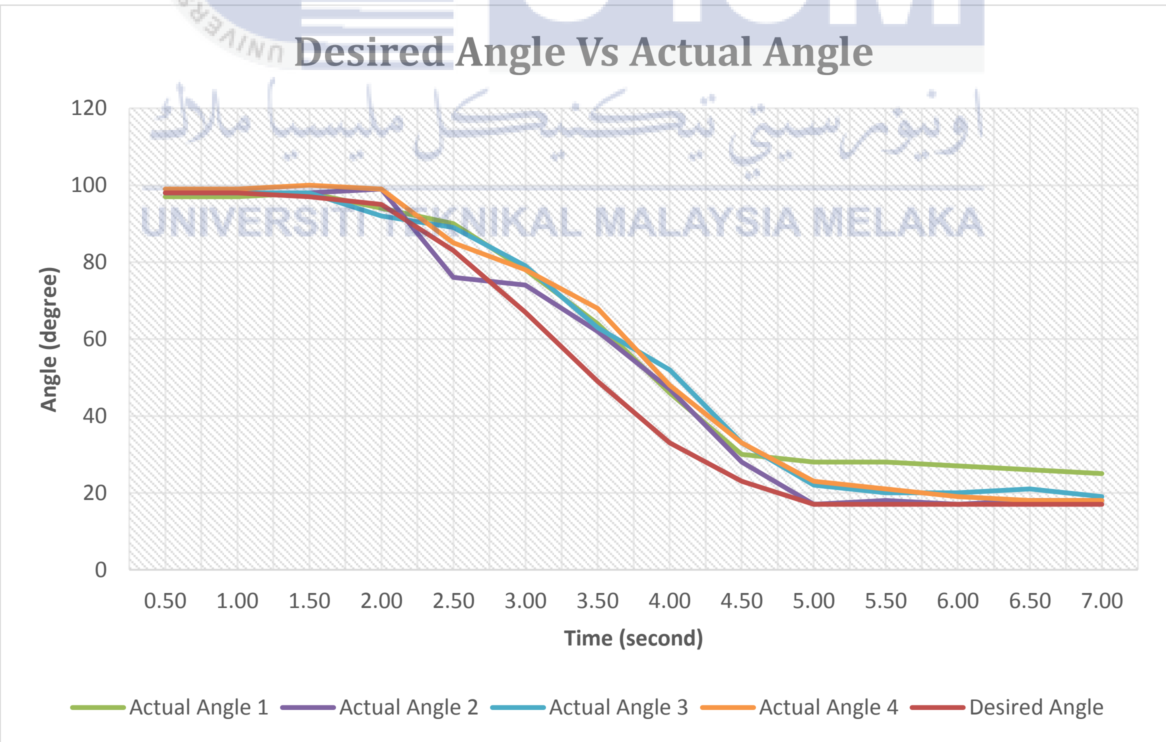


Figure 4.7: Angle of BCR Face Every 0.5 Second Along Curve

Error of Number of Step (Method 1 – Curve)

Table 4.5: Error of Number of Step (Curve)

Time	Design Step	Actual Step 1	Error 1	Actual Step 2	Error 2	Actual Step 3	Error 3	Actual Step 4	Error 4
0.50	0	0	0	0	0	0	0	0	0
1.00	0	0	0	0	0	0	0	0	0
1.50	3	0	-3	0	-3	0	-3	0	-3
2.00	47	38	-9	37	-10	39	-8	39	-8
2.50	100	97	-3	96	-4	98	-2	104	4
3.00	154	161	7	151	-3	155	1	166	12
3.50	160	160	0	160	0	163	3	161	1
4.00	160	165	5	156	-4	158	-2	166	6
4.50	139	146	7	133	-6	147	8	150	11
5.00	108	103	-5	98	-10	112	4	108	0
5.50	44	32	-12	32	-12	35	-9	34	-10
6.00	0	0	0	4	4	0	0	0	0
6.50	0	1	1	0	0	5	5	5	5
7.00	0	2	2	0	0	2	2	0	0
Total Error			54		56		47		60
Mean Error			3.8571		4		3.3571		4.2857

Overall Mean Error = 3.8750

Root Mean Square Error, RMSE = $\sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}$

RMSE = $\sqrt{\frac{0.4525}{4}}$ = 0.3363

From the table as show above, it show that the average error for BCR will be 3.8750 numbers of step for every step. It mean that every one step BCR travel, it will have 3.8750 number of steps error and 0.31cm.

Error of Facing Angle (Method 1- Curve)

Table 4.6: Error of Facing Angle (Curve)

Time	Desired Angle	Actual Angle 1	Error 1	Actual Angle 2	Error 2	Actual Angle 3	Error 3	Actual Angle 4	Error 4
0.50	98	97	-1	98	0	98	0	99	1
1.00	98	97	-1	98	0	98	0	99	1
1.50	97	98	1	98	1	98	1	100	3
2.00	95	94	-1	99	4	92	-3	99	4
2.50	83	90	7	76	-7	89	6	85	2
3.00	67	78	11	74	7	79	12	78	11
3.50	49	64	15	62	13	63	14	68	19
4.00	33	46	13	47	14	52	19	48	15
4.50	23	30	7	28	5	33	10	33	10
5.00	17	28	11	17	0	22	5	23	6
5.50	17	28	11	18	1	20	3	21	4
6.00	17	27	10	17	0	20	3	19	2
6.50	17	26	9	18	1	21	4	18	1
7.00	17	25	8	18	1	19	2	18	1
Total Error			60		54		82		80
Mean Error			4.2857		3.8571		5.8571		5.7143

Overall Mean Error = 4.9286

Root Mean Square Error, RMSE = $\sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}$

RMSE = $\sqrt{\frac{0.4525}{4}}$ = 0.8719

For the average error of the angle of BCR facing along the curve shape is 4.9286. It show that the error of BCR moving in curve shape is bigger than the error of BCR moving in straight line.

4.3.1.3 S-Shape

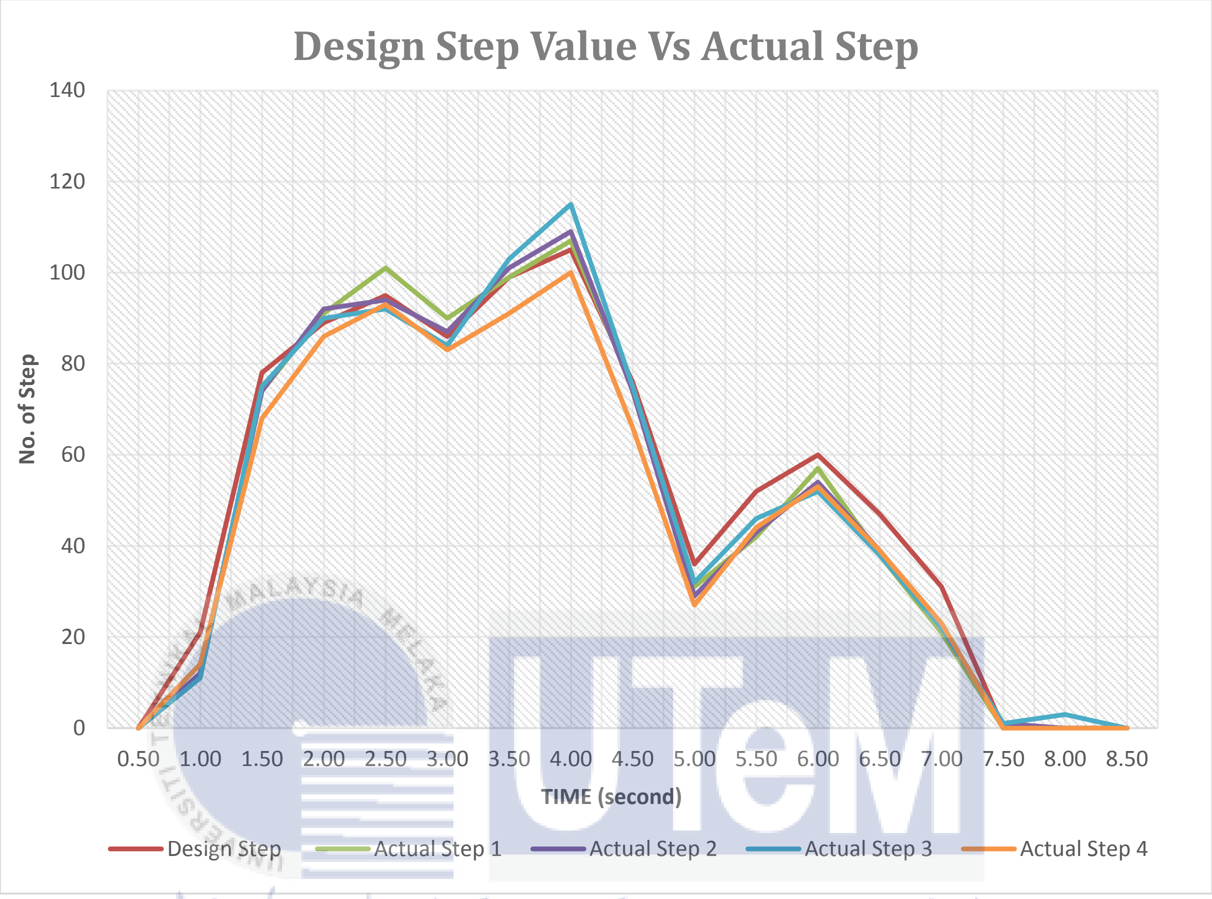


Figure 4.8: Number of Step Value Detected for Every 0.5 Second for S-Shape

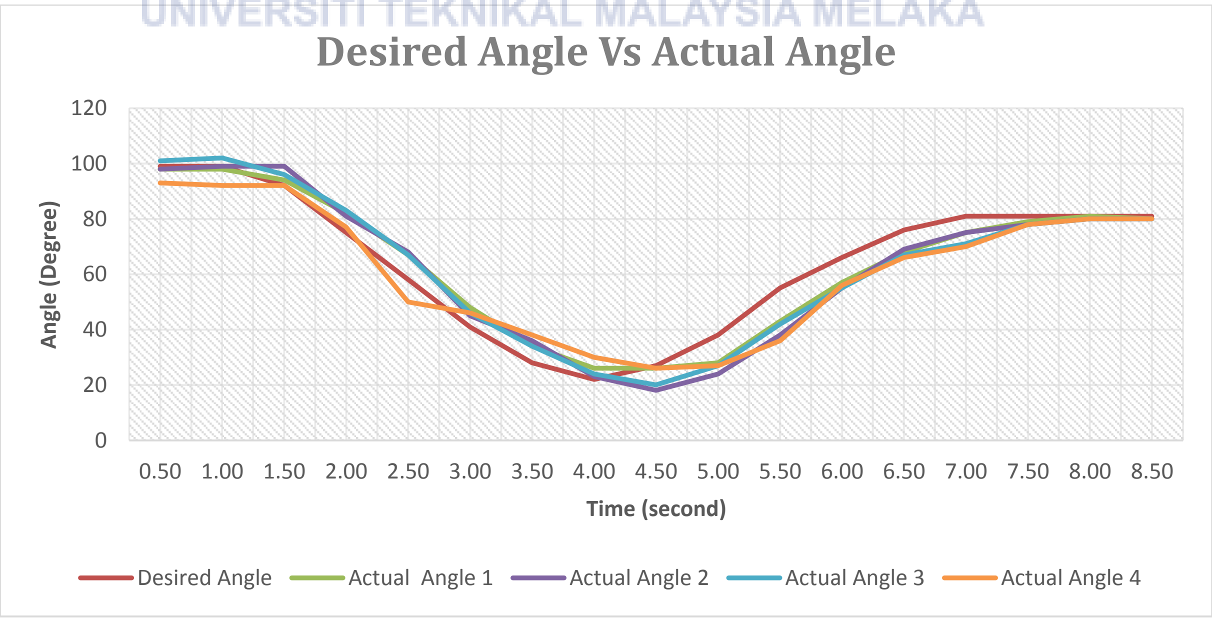


Figure 4.9: Angle of BCR Face Every 0.5 Second Along S-Shape

<u>Error of Number of Step (Method 1- S-Shape)</u>

Table 4.7: Error of Number of Step (S-Shape)

Time	Design Step	Actual Step 1	Error 1	Actual Step 2	Error 2	Actual Step 3	Error 3	Actual Step 4	Error 4
0.50	0	0	0	0	0	0	0	0	0
1.00	21	12	-9	12	-9	11	-10	14	-7
1.50	78	74	-4	74	-4	75	-3	68	-10
2.00	89	91	2	92	3	90	1	86	-3
2.50	95	101	6	94	-1	92	-3	93	-2
3.00	86	90	4	87	1	84	-2	83	-3
3.50	99	99	0	101	2	103	4	91	-8
4.00	105	107	2	109	4	115	10	100	-5
4.50	76	75	-1	74	-2	75	-1	66	-10
5.00	36	31	-5	29	-7	32	-4	27	-9
5.50	52	42	-10	43	-9	46	-6	44	-8
6.00	60	57	-3	54	-6	52	-8	53	-7
6.50	47	38	-9	39	-8	38	-9	39	-8
7.00	31	21	-10	22	-9	22	-9	23	-8
7.50	0	0	0	1	1	1	1	0	0
8.00	0	0	0	0	0	3	3	0	0
8.50	0	0	0	0	0	0	0	0	0
Total Error			65		66		74		88
Mean Error			3.8235		3.8824		4.3529		5.1765

Overall Mean Error = 4.3088

Root Mean Square Error, RMSE = $\sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}$

RMSE = $\sqrt{\frac{1.1722}{4}}$ = 0.5413

For the error of number of step BCR travel along the S shape the mean error is 4.3088 number of steps for one step. For every 0.5 seconds, BCR will have error about 0.34cm.

<u>Error of Facing Angle (Method 1-S-Shape)</u>

Table 4.8: Error of Facing Angle (S-Shape)

Time	Desired Angle	Actual Angle 1	Error 1	Actual Angle 2	Error 2	Actual Angle 3	Error 3	Actual Angle 4	Error 4
0.50	99	98	-1	98	-1	101	2	93	-6
1.00	99	98	-1	99	0	102	3	92	-7
1.50	92	94	2	99	7	96	4	92	0
2.00	75	82	7	81	6	83	8	77	2
2.50	58	67	9	68	10	67	9	50	-8
3.00	41	48	7	45	4	46	5	46	5
3.50	28	34	6	36	8	34	6	38	10
4.00	22	26	4	23	1	24	2	30	8
4.50	27	26	-1	18	-9	20	-7	26	-1
5.00	38	28	-10	24	-14	27	-11	27	-11
5.50	55	43	-12	38	-17	42	-13	36	-19
6.00	66	57	-9	55	-11	55	-11	56	-10
6.50	76	68	-8	69	-7	67	-9	66	-10
7.00	81	75	-6	75	-6	71	-10	70	-11
7.50	81	79	-2	78	-3	78	-3	78	-3
8.00	81	81	0	80	-1	80	-1	80	-1
8.50	81	80	-1	80	-1	80	-1	80	-1
Total Error			86		106		105		113
Mean Error			5.05882		6.23529		6.17647		6.64706

Overall Mean Error = 6.02941

Root Mean Square Error, RMSE = $\sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}$

RMSE = $\sqrt{\frac{1.3875}{4}}$ = 0.5890

For the error of BCR facing for one steps, it will have average error of 6.02941 degrees. After compare the data of the straight line, curve shape and S shape, it show that the longer the path travel by the BCR without feedback the greater the error it will occur. Besides, when compare the travel path as show in figure X, it show that the final position of desired and actual point are far apart to each other. To overcome this problem, another method that is method 2 was create to reduce the error of the BCR.

4.3.2 Method 2 (Closed-Loop System)

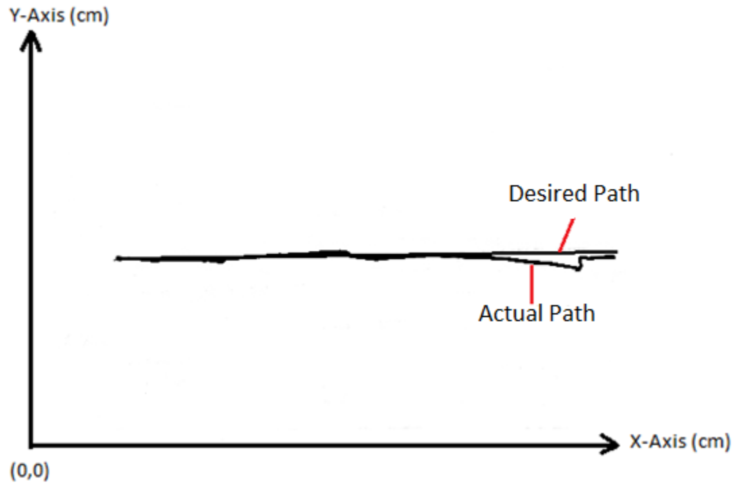


Figure 4.10: Straight Line with Method 2

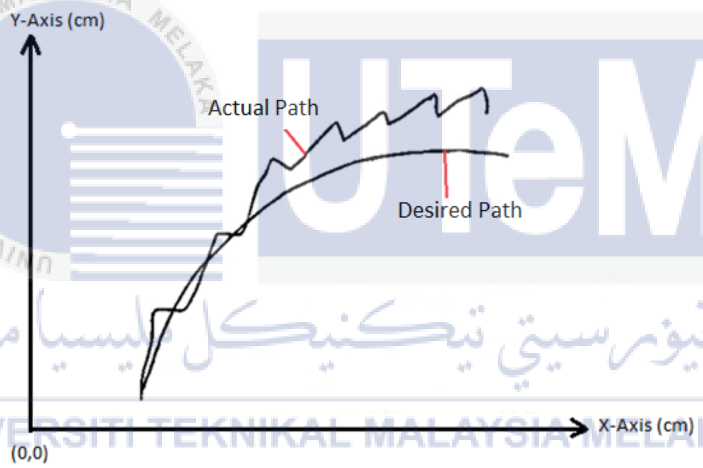


Figure 4.11: Curve Line with Method 2

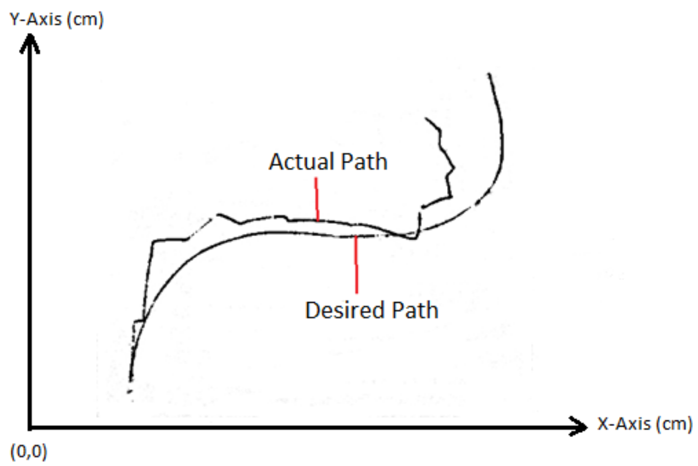


Figure 4.12: S Shape with Method 2

By using method 2, 3 difference type of trajectory that undergo in method 1 like straight line, curve and S shape will redo again. But the difference is the BCR will stop for every one step and adjust the angle same as desired angle only the BCR will move according to the data store in the Arduino. After go through this experiment, it show that the error of the desired path and actual path are become smaller. This is because the distance between final desired position and actual final position is smaller compare with method 1.

In this method 2, the angle of BCR facing will be 100 percentage exactly the same as the designed angle. This is because the BCR will adjust it position exactly the same as design angle only it will move to another point. Therefore, the angle error of BCR will not be discuss in this part.

4.3.2.1 Straight Line

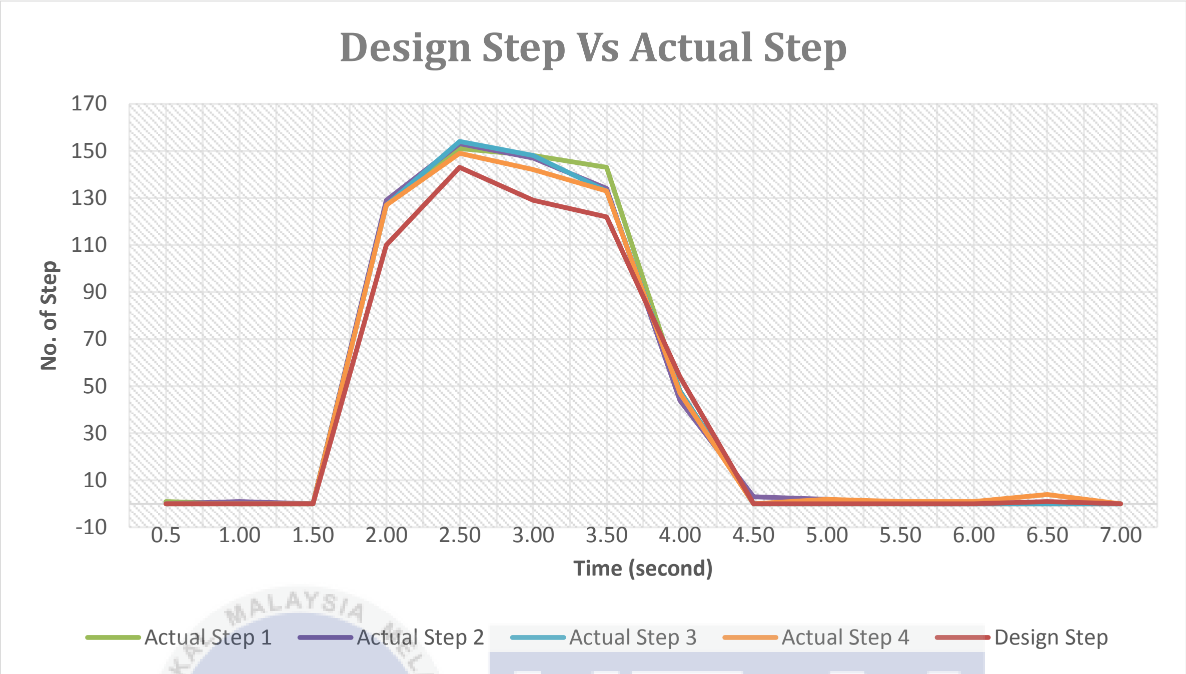


Figure 4.13: Number of Step Value Detected for Every 0.5 Second for Straight Line

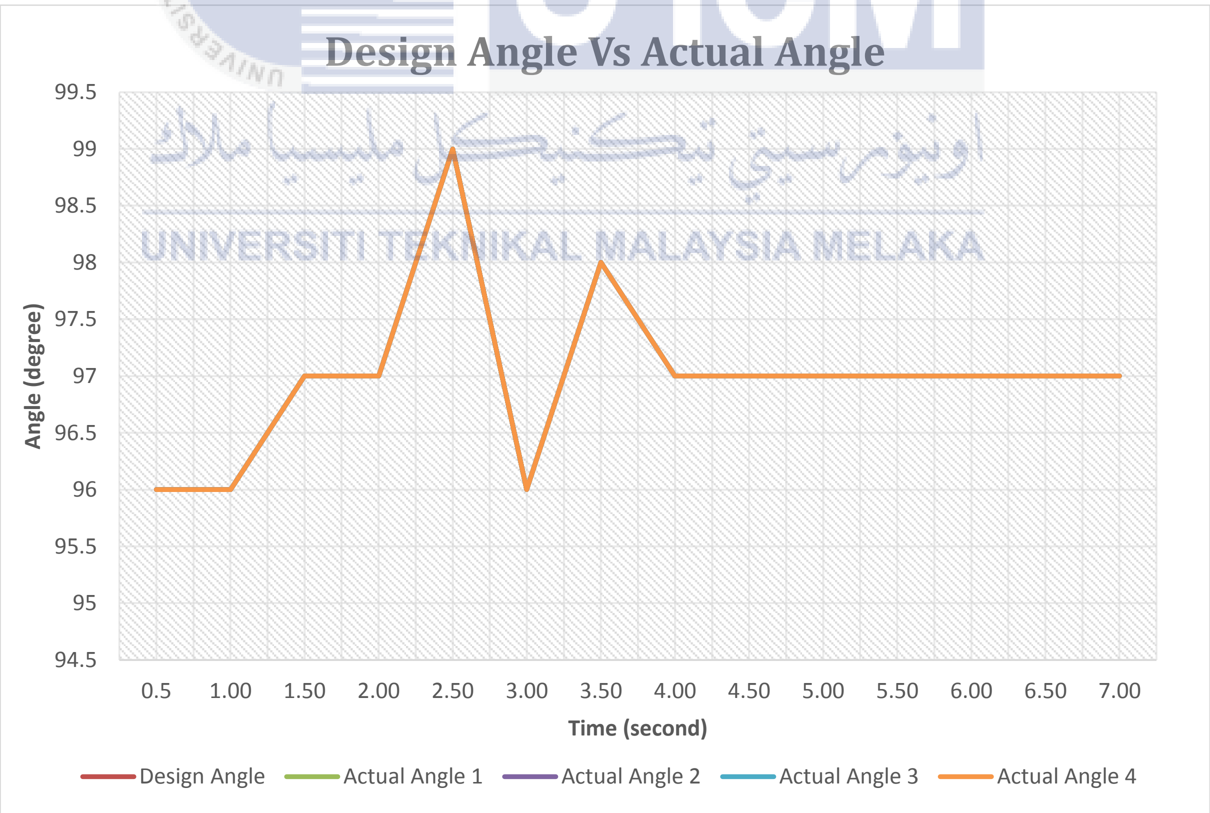


Figure 4.14: Angle of BCR Face Every 0.5 Second Along Straight Line

Error of Number of Step (Method 2- Straight Line)

Table 4.9: Error of Number of Step (Straight Line)

Time	Design Step	Actual Step 1	Error 1	Actual Step 2	Error 2	Actual Step 3	Error 3	Actual Step 4	Error 4
0.5	0	1	1	0	0	0	0	0	0
1.00	0	0	0	1	1	0	0	0	0
1.50	0	0	0	0	0	0	0	0	0
2.00	110	128	18	129	19	127	17	127	17
2.50	143	151	8	153	10	154	11	149	6
3.00	129	148	19	147	18	148	19	142	13
3.50	122	143	21	134	12	133	11	133	11
4.00	54	48	-6	44	-10	48	-6	47	-7
4.50	0	0	0	3	3	0	0	0	0
5.00	0	0	0	2	2	0	0	2	2
5.50	0	0	0	0	0	0	0	1	1
6.00	0	0	0	0	0	0	0	1	1
6.50	1	0	-1	0	-1	0	-1	4	3
7.00	0	0	0	0	0	0	0	0	0
Total Error			74		76		65		61
Mean Error			5.2857		5.4286		4.6429		4.3571

Overall Mean Error = 4.928571

Root Mean Square Error, RMSE = $\sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}$

RMSE = $\sqrt{\frac{0.7857}{4}}$ = 0.4432

In this experiment, it show that average mean error of BCR moving in straight line with method 2 is 4.928571 number of step for every step. It mean that the error of BCR finish one point to another point will have 0.39cm error.

4.3.2.2 Curve

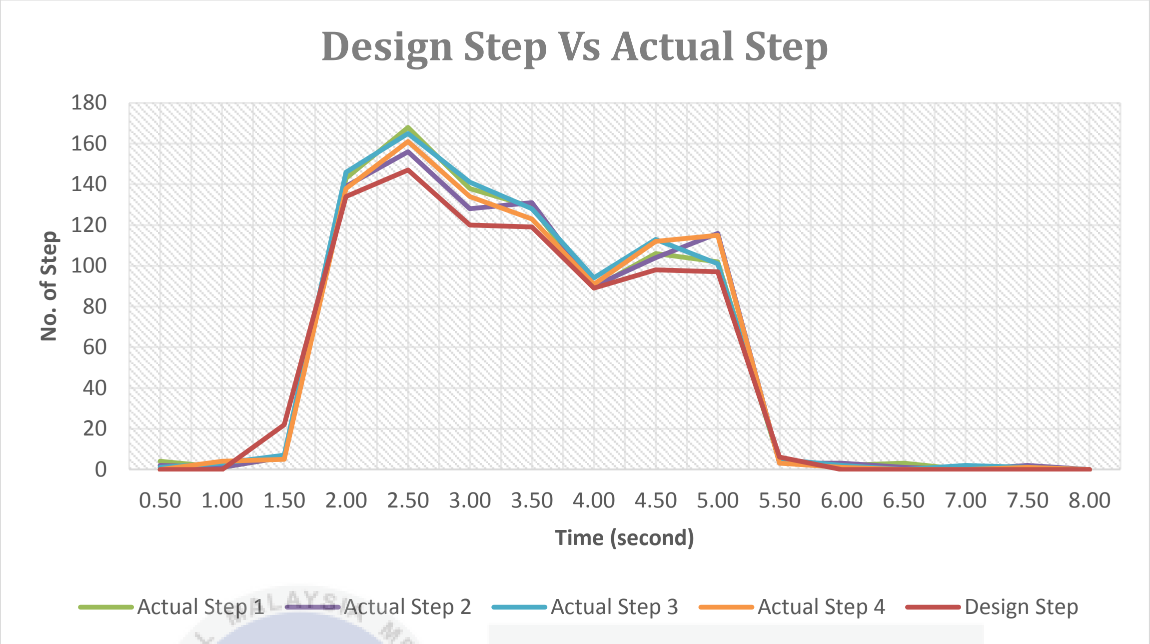


Figure 4.15: Number of Step Value Detected for Every 0.5 Second for Curve

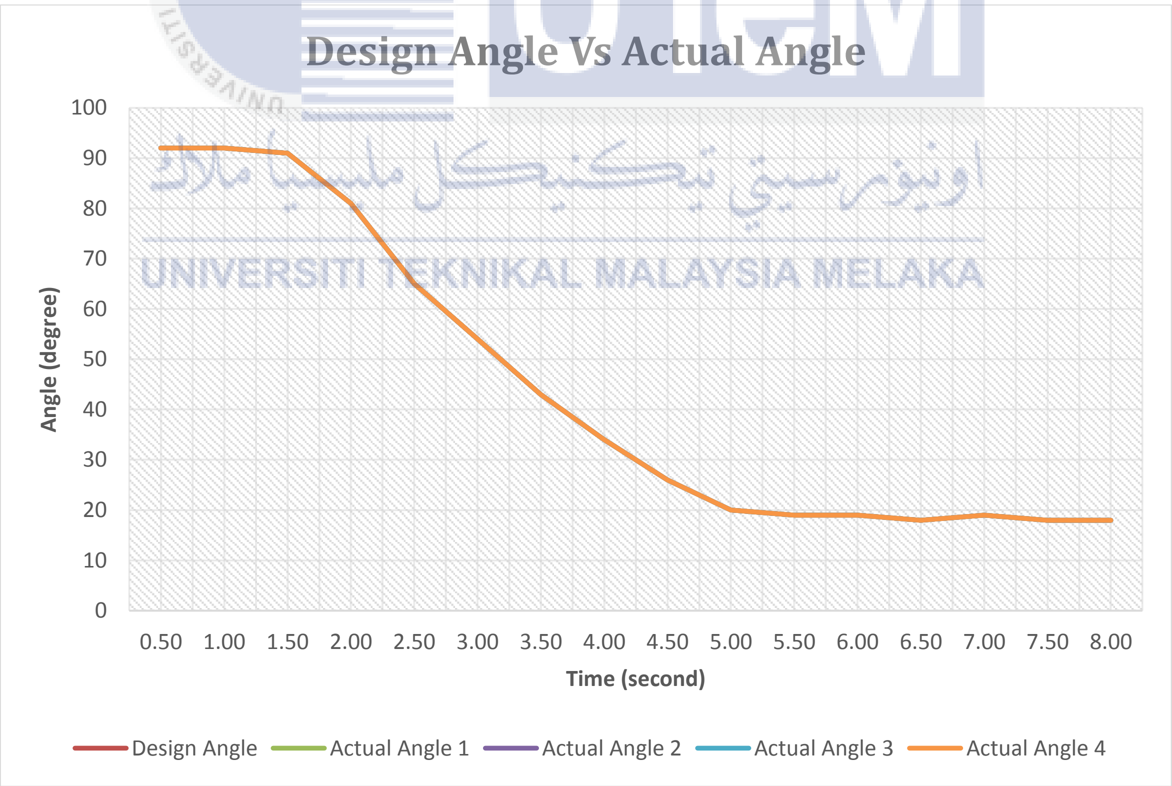


Figure 4.16: Angle of BCR Face Every 0.5 Second Along Curve

<u>Error of Number of Step (Method 2- Curve)</u>

Table 4.10: Error of Number of Step (Curve)

Time	Design Step	Actual Step 1	Error 1	Actual Step 2	Error 2	Actual Step 3	Error 3	Actual Step 4	Error 4
0.50	0	4	4	2	2	1	1	0	0
1.00	0	1	1	1	1	3	3	4	4
1.50	22	6	-16	6	-16	7	-15	5	-17
2.00	134	143	9	139	5	146	12	138	4
2.50	147	168	21	156	9	165	18	161	14
3.00	120	138	18	128	8	141	21	134	14
3.50	119	129	10	131	12	128	9	123	4
4.00	89	89	0	90	1	94	5	91	2
4.50	98	106	8	104	6	113	15	112	14
5.00	97	102	5	116	19	101	4	115	18
5.50	6	4	-2	3	-3	5	-1	3	-3
6.00	0	2	2	3	3	2	2	1	1
6.50	0	3	3	1	1	0	0	0	0
7.00	0	0	0	0	0	2	2	0	0
7.50	0	0	0	2	2	1	1	1	1
8.00	0	0	0	0	0	0	0	0	0
Total Error			99		88		109		96
Mean Error			6.1875		5.5		6.8125		6

Overall Mean Error = 6.125

Root Mean Square Error, RMSE = $\sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}$

RMSE = $\sqrt{\frac{0.8828}{4}}$ = 0.4698

In this experiment, it show that average mean error of BCR moving in curve shape with method 2 is 6.125 number of step for every step. It mean that the error of BCR finish one point to another point will have 0.49cm error.

4.3.2.3 S-Shape

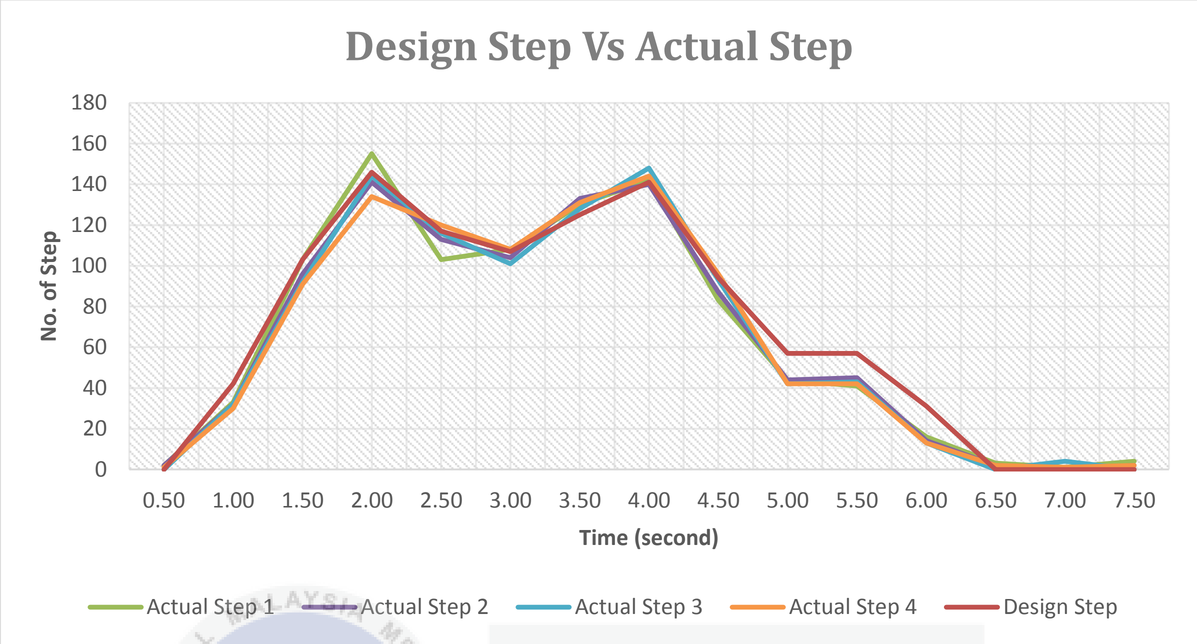


Figure 4.17: Number of Step Value Detected for Every 0.5 Second for S-Shape

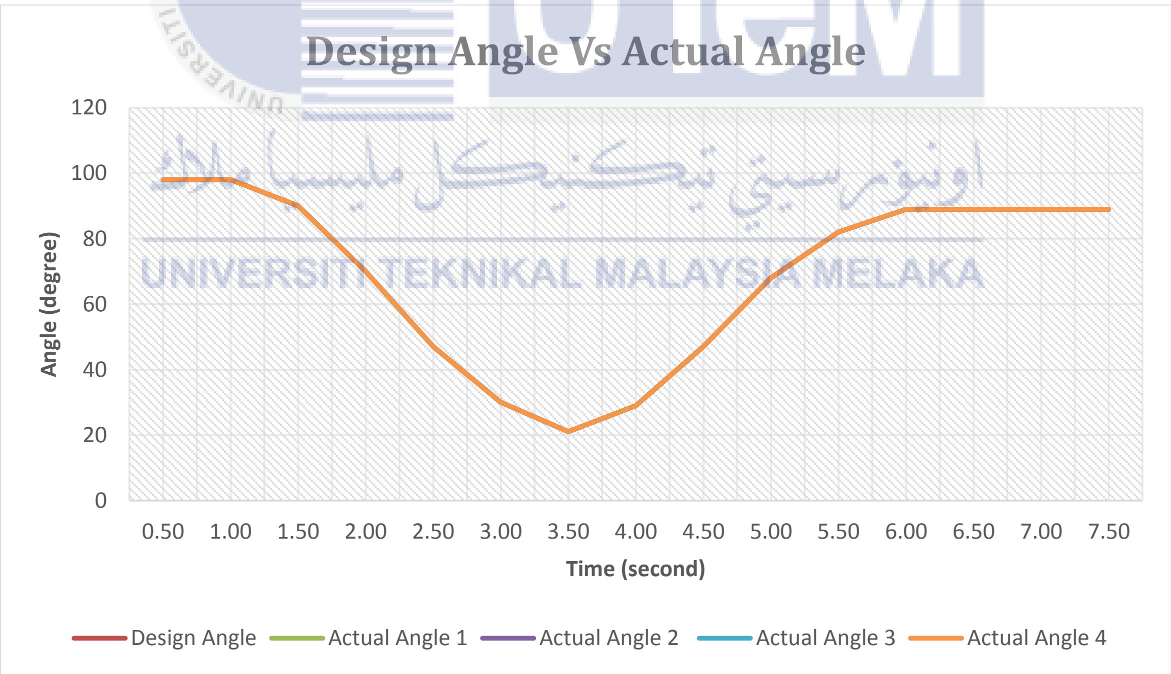


Figure 4.18: Angle of BCR Face Every 0.5 Second Along S-Shape

<u>Error of Number of Step (Method 2- S Shape)</u>

Table 4.11: Error of Number of Step (S-Shape)

Time	Design Step	Actual Step 1	Error 1	Actual Step 2	Error 2	Actual Step 3	Error 3	Actual Step 4	Error 4
0.50	0	2	2	2	2	0	0	1	1
1.00	42	33	-9	32	-10	32	-10	30	-12
1.50	103	103	0	96	-7	92	-11	91	-12
2.00	146	155	9	141	-5	144	-2	134	-12
2.50	117	103	-14	113	-4	116	-1	120	3
3.00	107	108	1	104	-3	101	-6	108	1
3.50	125	130	5	133	8	128	3	131	6
4.00	141	142	1	140	-1	148	7	144	3
4.50	94	83	-11	87	-7	93	-1	96	2
5.00	57	44	-13	44	-13	42	-15	42	-15
5.50	57	41	-16	45	-12	43	-14	42	-15
6.00	31	16	-15	14	-17	13	-18	13	-18
6.50	0	3	3	2	2	0	0	2	2
7.00	0	1	1	0	0	4	4	1	1
7.50	0	4	4	2	2	0	0	2	2
Total Error			104		93		92		105
Mean Error			6.9333		6.2		6.1333		7

Overall Mean Error = 6.56667

Root Mean Square Error, RMSE = $\sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}$

RMSE = $\sqrt{\frac{0.64444}{4}}$ = 0.4014

In this experiment, it show that average mean error of BCR moving in S shape with method 2 is 6.56667 number of step for every step. It mean that the error of BCR finish one point to another point will have 0.53cm error.

4.4 Discussion

Table 4.12: Mean Error of Number of Step of Encoder

	Mean Error(Number of Step)		Distance Error (cm)	
	Method 1	Method 2	Method 1	Method 2
Straight Line	3.26667	4.928571	0.26	0.39
Curve Shape	3.875	6.125	0.31	0.49
S-Shape	4.3088	6.56667	0.34	0.53

After go through method 1 and method 2, it show that the number of steps error for method 1 is less or smaller than method 2. This is because in method 2 the BCR have to adjust direction it facing exactly the same as desired angle before it move to another point. When undergo tuning, DC gear motor with encoder will move to tune angle and number of step of motor when adjust its direction will be included in the calculation. Although the mean error of method 2 is higher, trajectory that draw out by BCR as show in Figure 4.10, 4.11 and 4.12 for method 2 show that the error is smaller compare with Figure 4.1, 4.2 and 4.3 for method 1. Method 2 will make BCR face to desired direction and prevent BCR move far from desired trajectory while using method 1 BCR will move far from desired trajectory. Therefore, error about number of steps for method 2 greater than method 1 is acceptable.

As label in Table 16, the distance error keep on increase from straight line, curve shape to S-shape. From the data it show that most of the error occur when turning. The greater the number for turning in trajectory, the higher the error will occur.

Table 1.13: Mean Error of Facing Angle

	Mean Error(Face Angle)	
	Method 1	Method 2
Straight Line	3.3333	0
Curve Shape	4.9286	0
S-Shape	6.02941	0

About the angle error of BCR facing, method 2 will have 100 percentage accurate in angle compare with method 1. Besides, after go through all the data, it show that error for method 1 will keep on increase base on number of turn of trajectory. Therefore, method 2 that contain feedback is much better compare with method 1. More detail graph about method 1 and method 2 will be attach in appendix.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

Prototype for this project Bathroom Cleaning Robot (BCR) is able to operate as expected way. With combination of encoder and digital compass sensor, trajectory make by human demonstration is able to detect and replay it.

Two difference mode were designed in this project that is learn human demonstration name as RECORD Mode while redo human demonstration name as REPLAY Mode. In RECORD Mode, every 0.5 seconds will record it number of steps of encoder detected and angle of BCR facing to. For REPLAY Mode, BCR will demonstrate the trajectory based on the data that collected in RECORD Mode by control both of the DC gear motor with encoder. Two difference system were designed to undergo REPLAY Mode that is open-loop system and closed-loop system.

When operate the bathroom cleaning robot in open-loop system (method 1) it show that the end point of desired and actual are far apart. To overcome this problem the closed loop system was introduce in this project to reduce the distance between desired and actual end point. According to the distance between end point of desired and actual point, distance of method 2 is closer compare with method 1. After compare the data collected by using two difference method, it show that method 2 that is using closed-loop system is much more better than method 1. Closed-loop system will adjust BCR facing angle to prevent it move far from desired trajectory.

5.2 Recommendation

Bathroom cleaning robot hard to stop due to inertia force and low gear ration when moving along the trajectory. Therefore, gear ration of DC gear motor should be increase to acceptable value to increase the torque of robot. This is because higher the gear ratio or torque value, the bigger the force will stop the robot keep moving when do not have power supply to motor.

To smooth the trajectory of BCR when operate in replay mode, controller like PI, PD, PID or fuzzy logic should be implement into the robot. With controller, Pulse-Width-Modulation (PWM) of the motor will keep on changing base on the data that collected in record mode and reduce the settling time for the robot reach selected position.

REFERENCE

[1] S. K. Gupta, “When Will We Have Robots To Help With Household Chores? - IEEE Spectrum.” [Online]. Available: http://spectrum.ieee.org/automaton/robotics/home-robots/when-will-we-have-robots-to-help-with-household-chores. [Accessed: 30-Oct-2014].

[2] E. Guizzo, “So, Where Are My Robot Servants? - IEEE Spectrum.” [Online]. Available: http://spectrum.ieee.org/robotics/home-robots/so-where-are-my-robot-servants. [Accessed: 30-Oct-2014].

[3] “Robot-Era - Implementation and integration of advanced Robotic systems and intelligent environments in real scenarios for the ageing population.” [Online]. Available: http://www.robot-era.eu/robotera/index.php?pagina=news&id=185&blocco=70. [Accessed: 17-Nov-2014].

[4] J. Olson, I. Kirillov, and J. Sweeney, “Simple Mapping and Path-planning with the Roomba,” pp. 1–4.

[5] K. Skrzypczyk, R. Road, and B. Bd, “Surface Covering Algorithms for Semiautonomous Vacuum Cleaner,” pp. 294–298, 2002.

[6] M. Waanders, “Coverage Path Planning for Mobile Cleaning Robots,” 2011.

[7] J. E. Young, K. Ishii, T. Igarashi, and E. Sharlin, “Showing robots how to follow people using a broomstick interface,” in *2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2010, pp. 133–134.

[8] Chih-Hao Chen and Kai-Tai Song, “Complete coverage motion control of a cleaning robot using infrared sensors,” in *IEEE International Conference on Mechatronics, 2005. ICM ’05.*, 2005, pp. 543–548.

[9] M. Tresanchez, T. Palleja, M. Teixido, and J. Palacin, “Estimating a room size using encoders and collision detectors: application to a cleaning mobile robot,” in *2007 IEEE International Symposium on Intelligent Signal Processing*, 2007, pp. 1–6.

[10] T. Miyake, H. Ishihara, R. Shoji, and S. Yoshida, “Development of Small-Size Window Cleaning Robot A traveling direction control on Vertical surface Using accelerometer,” in *2006 International Conference on Mechatronics and Automation*, 2006, pp. 1302–1307.

[11] F. Aguilar-Acevedo and V. Granados Alejo, “Using open-source platform for trajectory control of DC motors,” in *2013 IEEE International Autumn Meeting on Power Electronics and Computing (ROPEC)*, 2013, pp. 1–5.

[12] Kuo-Lan Su, Sheng-Ven Shiau, Jr-Hung Guo, and Yi-Lin Liao, “Implement a Path Planning Method for Mobile Robots on the Unknown Environment,” in *2010 Fourth International Conference on Genetic and Evolutionary Computing*, 2010, pp. 445–448.

[13] Zhou Jianjun, Zhang Man, Cao Min'gang, Li Su, and Liu Bin, "Automatic navigation system for electric power vehicles with EPS," in *2008 IEEE Vehicle Power and Propulsion Conference*, 2008, pp. 1–3.

[14] M. R. Ghazali, W. I. Ibrahim, M. Z. M. Tumari, and C. W. Hong, "Development of straight line robot movement using Digital Compas Module," in *The 6th International Conference on Soft Computing and Intelligent Systems, and The 13th International Symposium on Advanced Intelligence Systems*, 2012, pp. 1409–1413.

[15] Y. Watanabe, "Development of small robot for home floor cleaning," in *Proceedings of the 41st SICE Annual Conference. SICE 2002.*, vol. 5, pp. 3222–3223.

APPENDICES

APPENDIX A

Distance travel by the wheel for difference number of step value.

No. of Steps	Degree	Radius	Distance
1	1.44	0.025132741	0.080424772
2	2.88	0.050265482	0.160849544
3	4.32	0.075398224	0.241274316
4	5.76	0.100530965	0.321699088
5	7.2	0.125663706	0.40212386
6	8.64	0.150796447	0.482548632
7	10.08	0.175929189	0.562973404
8	11.52	0.20106193	0.643398175
9	12.96	0.226194671	0.723822947
10	14.4	0.251327412	0.804247719
11	15.84	0.276460154	0.884672491
12	17.28	0.301592895	0.965097263
13	18.72	0.326725636	1.045522035
14	20.16	0.351858377	1.125946807
15	21.6	0.376991118	1.206371579
16	23.04	0.40212386	1.286796351
17	24.48	0.427256601	1.367221123
18	25.92	0.452389342	1.447645895
19	27.36	0.477522083	1.528070667
20	28.8	0.502654825	1.608495439
21	30.24	0.527787566	1.688920211
22	31.68	0.552920307	1.769344983
23	33.12	0.578053048	1.849769755
24	34.56	0.60318579	1.930194526
25	36	0.628318531	2.010619298
26	37.44	0.653451272	2.09104407
27	38.88	0.678584013	2.171468842
28	40.32	0.703716754	2.251893614
29	41.76	0.728849496	2.332318386
30	43.2	0.753982237	2.412743158

31	44.64	0.779114978	2.49316793
32	46.08	0.804247719	2.573592702
33	47.52	0.829380461	2.654017474
34	48.96	0.854513202	2.734442246
35	50.4	0.879645943	2.814867018
36	51.84	0.904778684	2.89529179
37	53.28	0.929911426	2.975716562
38	54.72	0.955044167	3.056141334
39	56.16	0.980176908	3.136566106
40	57.6	1.005309649	3.216990877
41	59.04	1.03044239	3.297415649
42	60.48	1.055575132	3.377840421
43	61.92	1.080707873	3.458265193
44	63.36	1.105840614	3.538689965
45	64.8	1.130973355	3.619114737
46	66.24	1.156106097	3.699539509
47	67.68	1.181238838	3.779964281
48	69.12	1.206371579	3.860389053
49	70.56	1.23150432	3.940813825
50	72	1.256637062	4.021238597
51	73.44	1.281769803	4.101663369
52	74.88	1.306902544	4.182088141
53	76.32	1.332035285	4.262512913
54	77.76	1.357168026	4.342937685
55	79.2	1.382300768	4.423362456
56	80.64	1.407433509	4.503787228
57	82.08	1.43256625	4.584212
58	83.52	1.457698991	4.664636772
59	84.96	1.482831733	4.745061544
60	86.4	1.507964474	4.825486316
61	87.84	1.533097215	4.905911088
62	89.28	1.558229956	4.98633586
63	90.72	1.583362697	5.066760632
64	92.16	1.608495439	5.147185404
65	93.6	1.63362818	5.227610176
66	95.04	1.658760921	5.308034948
67	96.48	1.683893662	5.38845972
68	97.92	1.709026404	5.468884492
69	99.36	1.734159145	5.549309264
70	100.8	1.759291886	5.629734036
71	102.24	1.784424627	5.710158807
72	103.68	1.809557369	5.790583579
73	105.12	1.83469011	5.871008351
74	106.56	1.859822851	5.951433123
75	108	1.884955592	6.031857895

76	109.44	1.910088333	6.112282667
77	110.88	1.935221075	6.192707439
78	112.32	1.960353816	6.273132211
79	113.76	1.985486557	6.353556983
80	115.2	2.010619298	6.433981755
81	116.64	2.03575204	6.514406527
82	118.08	2.060884781	6.594831299
83	119.52	2.086017522	6.675256071
84	120.96	2.111150263	6.755680843
85	122.4	2.136283005	6.836105615
86	123.84	2.161415746	6.916530386
87	125.28	2.186548487	6.996955158
88	126.72	2.211681228	7.07737993
89	128.16	2.236813969	7.157804702
90	129.6	2.261946711	7.238229474
91	131.04	2.287079452	7.318654246
92	132.48	2.312212193	7.399079018
93	133.92	2.337344934	7.47950379
94	135.36	2.362477676	7.559928562
95	136.8	2.387610417	7.640353334
96	138.24	2.412743158	7.720778106
97	139.68	2.437875899	7.801202878
98	141.12	2.463008641	7.88162765
99	142.56	2.488141382	7.962052422
100	144	2.513274123	8.042477194
101	145.44	2.538406864	8.122901966
102	146.88	2.563539605	8.203326737
103	148.32	2.588672347	8.283751509
104	149.76	2.613805088	8.364176281
105	151.2	2.638937829	8.444601053
106	152.64	2.66407057	8.525025825
107	154.08	2.689203312	8.605450597
108	155.52	2.714336053	8.685875369
109	156.96	2.739468794	8.766300141
110	158.4	2.764601535	8.846724913
111	159.84	2.789734277	8.927149685
112	161.28	2.814867018	9.007574457
113	162.72	2.839999759	9.087999229
114	164.16	2.8651325	9.168424001
115	165.6	2.890265241	9.248848773
116	167.04	2.915397983	9.329273545
117	168.48	2.940530724	9.409698317
118	169.92	2.965663465	9.490123088
119	171.36	2.990796206	9.57054786
120	172.8	3.015928948	9.650972632

121	174.24	3.041061689	9.731397404
122	175.68	3.06619443	9.811822176
123	177.12	3.091327171	9.892246948
124	178.56	3.116459913	9.97267172
125	180	3.141592654	10.05309649
126	181.44	3.166725395	10.13352126
127	182.88	3.191858136	10.21394604
128	184.32	3.216990877	10.29437081
129	185.76	3.242123619	10.37479558
130	187.2	3.26725636	10.45522035
131	188.64	3.292389101	10.53564512
132	190.08	3.317521842	10.6160699
133	191.52	3.342654584	10.69649467
134	192.96	3.367787325	10.77691944
135	194.4	3.392920066	10.85734421
136	195.84	3.418052807	10.93776898
137	197.28	3.443185549	11.01819376
138	198.72	3.46831829	11.09861853
139	200.16	3.493451031	11.1790433
140	201.6	3.518583772	11.25946807
141	203.04	3.543716513	11.33989284
142	204.48	3.568849255	11.42031761
143	205.92	3.593981996	11.50074239
144	207.36	3.619114737	11.58116716
145	208.8	3.644247478	11.66159193
146	210.24	3.66938022	11.7420167
147	211.68	3.694512961	11.82244147
148	213.12	3.719645702	11.90286625
149	214.56	3.744778443	11.98329102
150	216	3.769911185	12.06371579
151	217.44	3.795043926	12.14414056
152	218.88	3.820176667	12.22456533
153	220.32	3.845309408	12.30499011
154	221.76	3.870442149	12.38541488
155	223.2	3.895574891	12.46583965
156	224.64	3.920707632	12.54626442
157	226.08	3.945840373	12.62668919
158	227.52	3.970973114	12.70711397
159	228.96	3.996105856	12.78753874
160	230.4	4.021238597	12.86796351
161	231.84	4.046371338	12.94838828
162	233.28	4.071504079	13.02881305
163	234.72	4.09663682	13.10923783
164	236.16	4.121769562	13.1896626
165	237.6	4.146902303	13.27008737

166	239.04	4.172035044	13.35051214
167	240.48	4.197167785	13.43093691
168	241.92	4.222300527	13.51136169
169	243.36	4.247433268	13.59178646
170	244.8	4.272566009	13.67221123
171	246.24	4.29769875	13.752636
172	247.68	4.322831492	13.83306077
173	249.12	4.347964233	13.91348554
174	250.56	4.373096974	13.99391032
175	252	4.398229715	14.07433509
176	253.44	4.423362456	14.15475986
177	254.88	4.448495198	14.23518463
178	256.32	4.473627939	14.3156094
179	257.76	4.49876068	14.39603418
180	259.2	4.523893421	14.47645895
181	260.64	4.549026163	14.55688372
182	262.08	4.574158904	14.63730849
183	263.52	4.599291645	14.71773326
184	264.96	4.624424386	14.79815804
185	266.4	4.649557128	14.87858281
186	267.84	4.674689869	14.95900758
187	269.28	4.69982261	15.03943235
188	270.72	4.724955351	15.11985712
189	272.16	4.750088092	15.2002819
190	273.6	4.775220834	15.28070667
191	275.04	4.800353575	15.36113144
192	276.48	4.825486316	15.44155621
193	277.92	4.850619057	15.52198098
194	279.36	4.875751799	15.60240576
195	280.8	4.90088454	15.68283053
196	282.24	4.926017281	15.7632553
197	283.68	4.951150022	15.84368007
198	285.12	4.976282764	15.92410484
199	286.56	5.001415505	16.00452962
200	288	5.026548246	16.08495439
201	289.44	5.051680987	16.16537916
202	290.88	5.076813728	16.24580393
203	292.32	5.10194647	16.3262287
204	293.76	5.127079211	16.40665347
205	295.2	5.152211952	16.48707825
206	296.64	5.177344693	16.56750302
207	298.08	5.202477435	16.64792779
208	299.52	5.227610176	16.72835256
209	300.96	5.252742917	16.80877733
210	302.4	5.277875658	16.88920211

211	303.84	5.3030084	16.96962688
212	305.28	5.328141141	17.05005165
213	306.72	5.353273882	17.13047642
214	308.16	5.378406623	17.21090119
215	309.6	5.403539364	17.29132597
216	311.04	5.428672106	17.37175074
217	312.48	5.453804847	17.45217551
218	313.92	5.478937588	17.53260028
219	315.36	5.504070329	17.61302505
220	316.8	5.529203071	17.69344983
221	318.24	5.554335812	17.7738746
222	319.68	5.579468553	17.85429937
223	321.12	5.604601294	17.93472414
224	322.56	5.629734036	18.01514891
225	324	5.654866777	18.09557369
226	325.44	5.679999518	18.17599846
227	326.88	5.705132259	18.25642323
228	328.32	5.730265	18.336848
229	329.76	5.755397742	18.41727277
230	331.2	5.780530483	18.49769755
231	332.64	5.805663224	18.57812232
232	334.08	5.830795965	18.65854709
233	335.52	5.855928707	18.73897186
234	336.96	5.881061448	18.81939663
235	338.4	5.906194189	18.8998214
236	339.84	5.93132693	18.98024618
237	341.28	5.956459672	19.06067095
238	342.72	5.981592413	19.14109572
239	344.16	6.006725154	19.22152049
240	345.6	6.031857895	19.30194526
241	347.04	6.056990636	19.38237004
242	348.48	6.082123378	19.46279481
243	349.92	6.107256119	19.54321958
244	351.36	6.13238886	19.62364435
245	352.8	6.157521601	19.70406912
246	354.24	6.182654343	19.7844939
247	355.68	6.207787084	19.86491867
248	357.12	6.232919825	19.94534344
249	358.56	6.258052566	20.02576821
250	360	6.283185308	20.10619298

Method 1 (Straight Line)

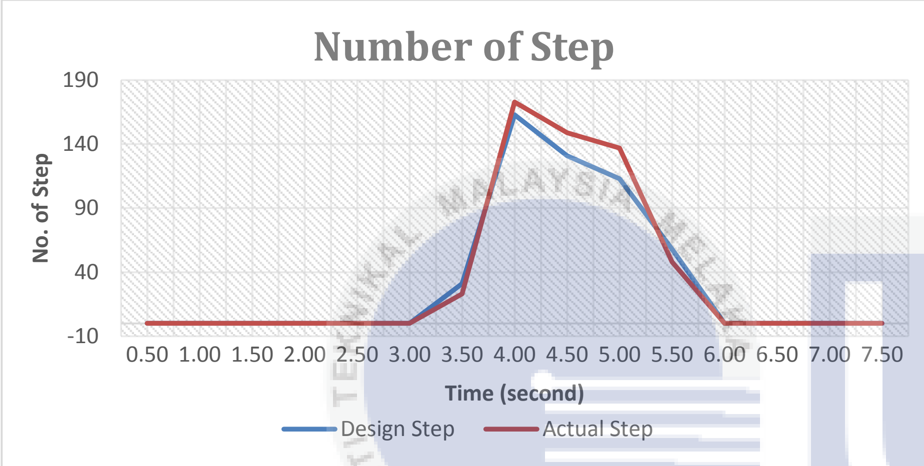


Figure 54: Number of Step for First Out of Four Experiment for Straight Line

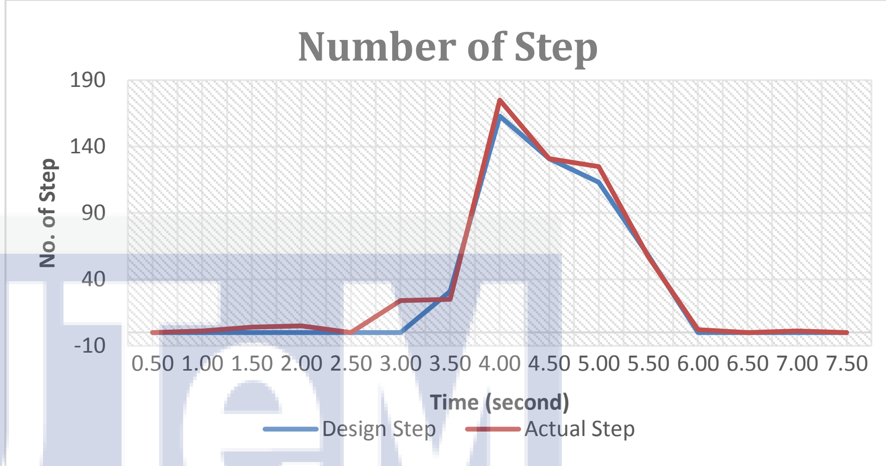


Figure 55: Number of Step for Second Out of Four Experiment for Straight Line

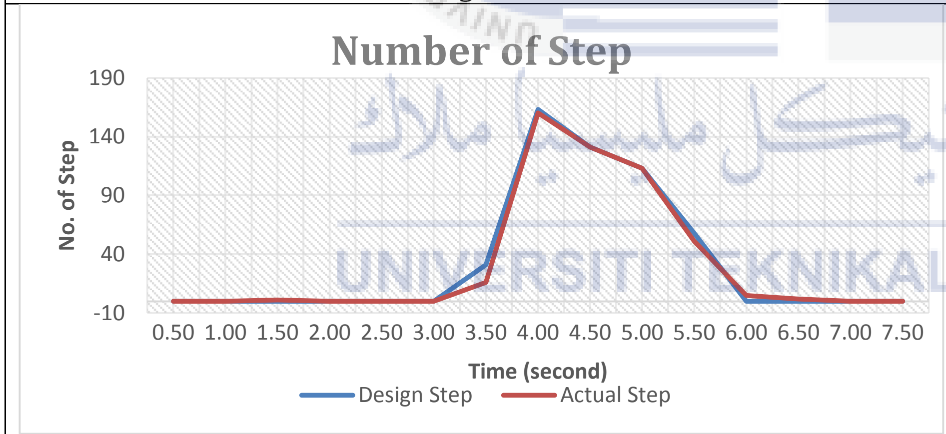


Figure 56: Number of Step for Third Out of Four Experiment for Straight Line

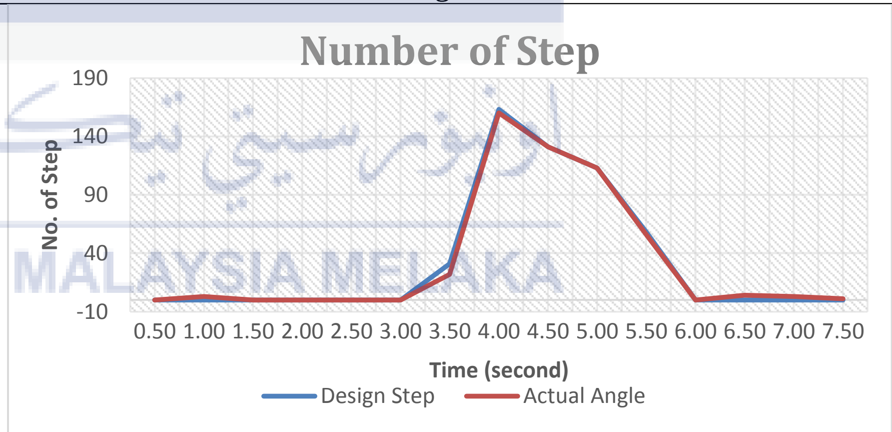


Figure 57: Number of Step for Forth Out of Four Experiment for Straight Line

Straight Line

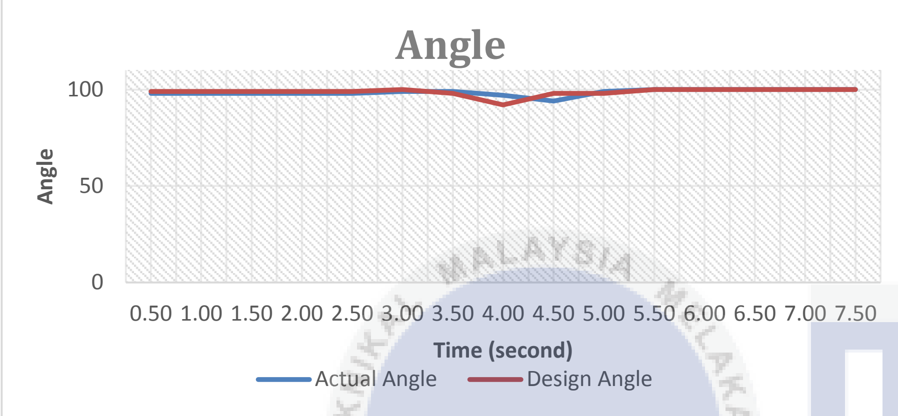


Figure 58: Facing Angle for First Out of Four Experiment for Straight Line

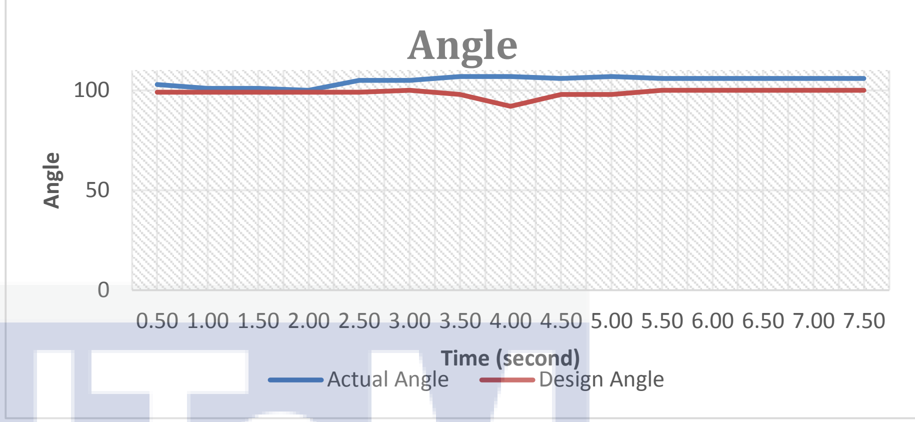


Figure 59: Facing Angle for Second Out of Four Experiment for Straight Line

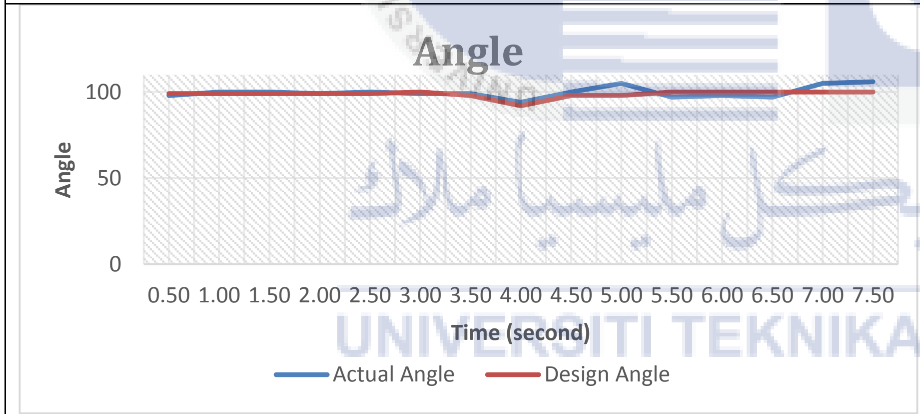


Figure 60: Facing Angle for Third Out of Four Experiment for Straight Line

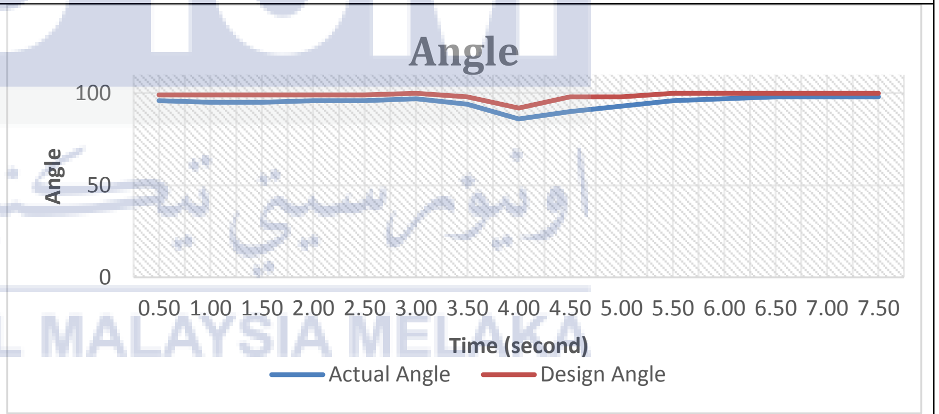


Figure 61: Facing Angle for Forth Out of Four Experiment for Straight Line

Curve

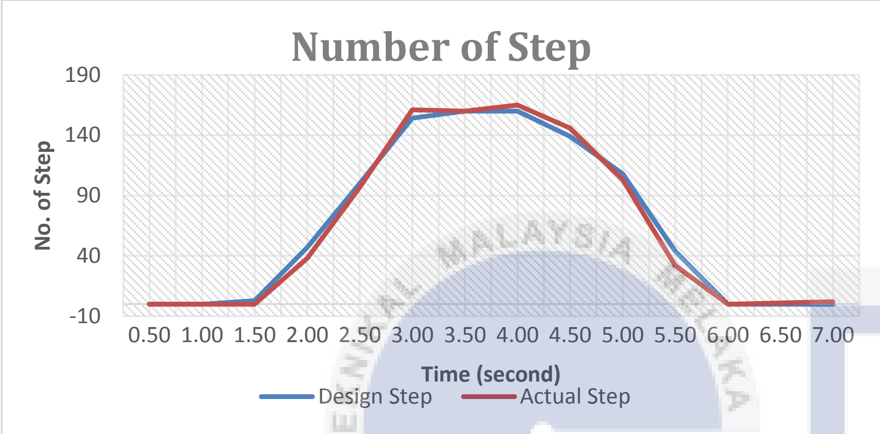


Figure 62: Number of Step for First Out of Four Experiment for Curve

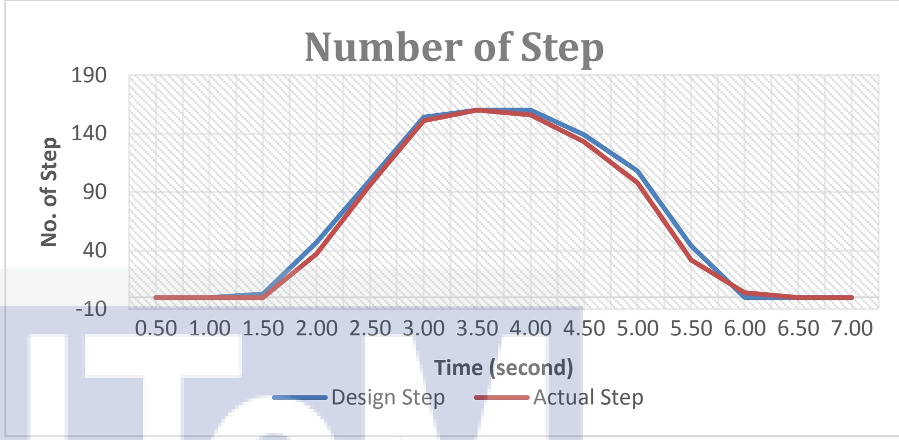


Figure 63: Number of Step for Second Out of Four Experiment for Curve

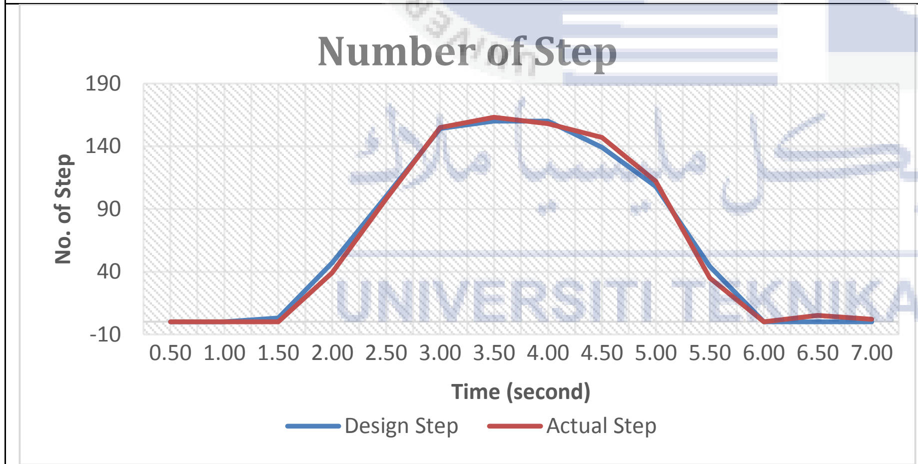


Figure 64: Number of Step for Third Out of Four Experiment for Curve

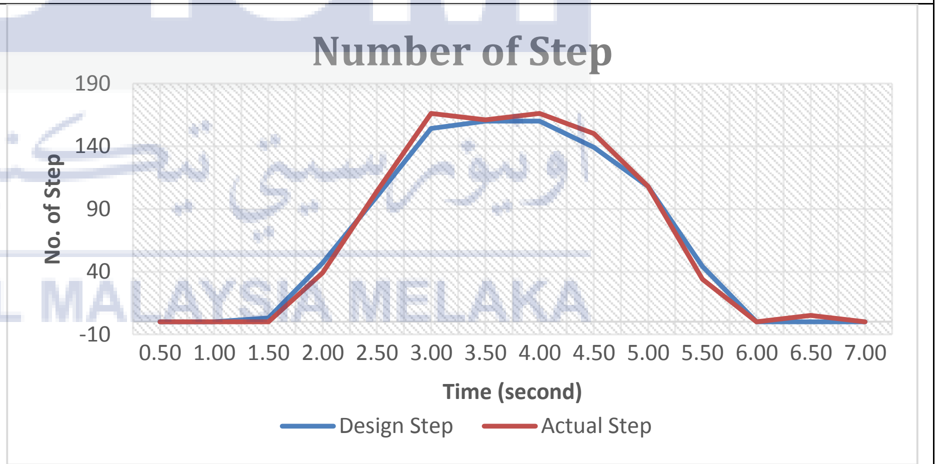


Figure 65: Number of Step for Forth Out of Four Experiment for Curve

Curve

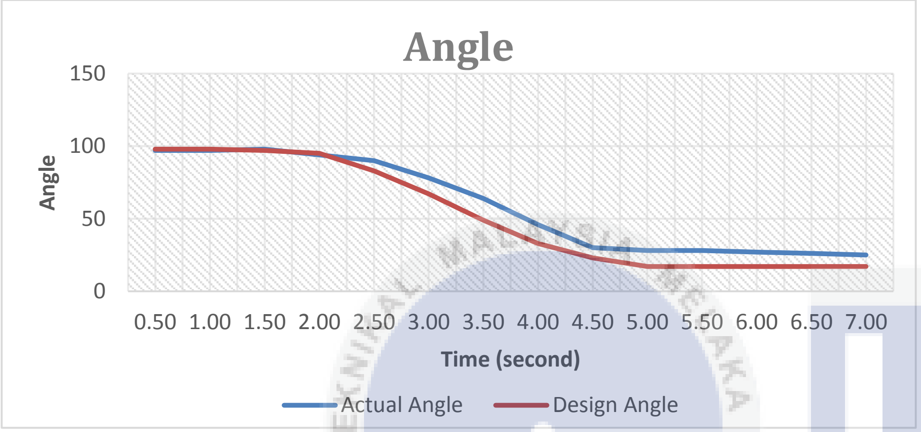


Figure 66: Facing Angle for First Out of Four Experiment for Curve

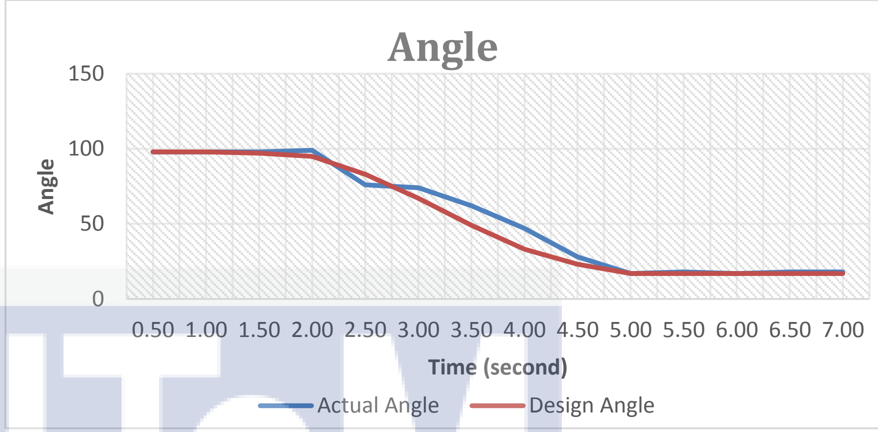


Figure 67: Facing Angle for Second Out of Four Experiment for Curve

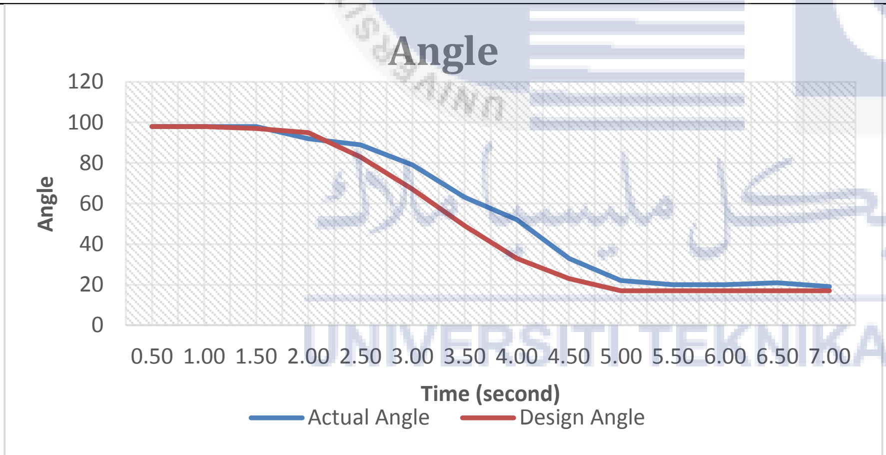


Figure 68: Facing Angle for Third Out of Four Experiment for Curve

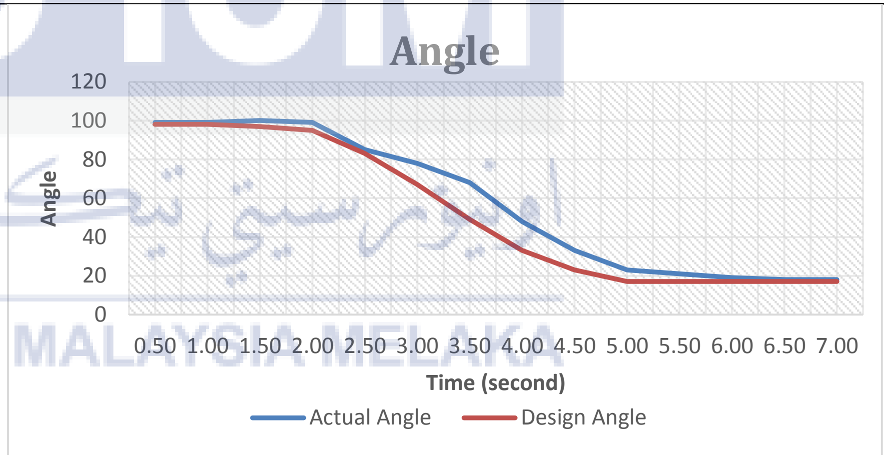


Figure 69: Facing Angle for Forth Out of Four Experiment for Curve

S-Shape

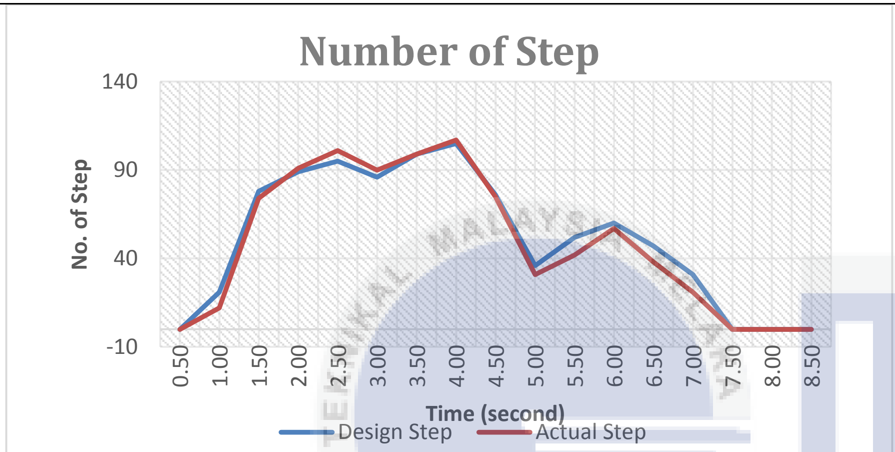


Figure 70: Number of Step for First Out of Four Experiment for S-Shape

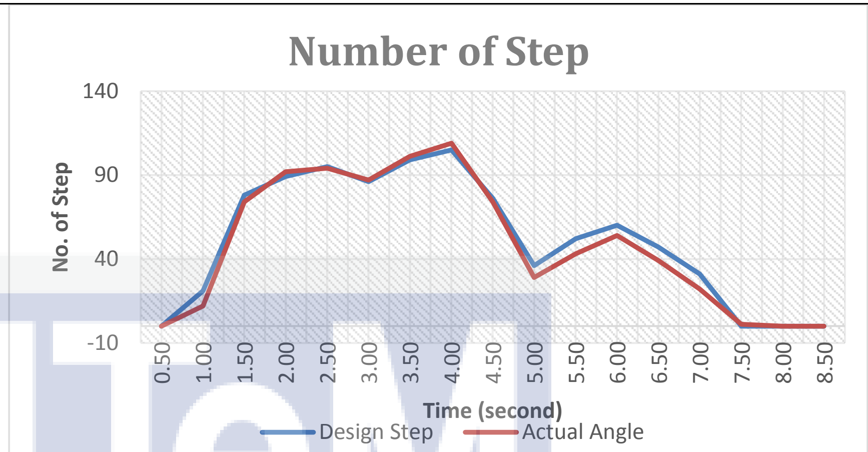


Figure 71:Number of Step for Second Out of Four Experiment for S-Shape

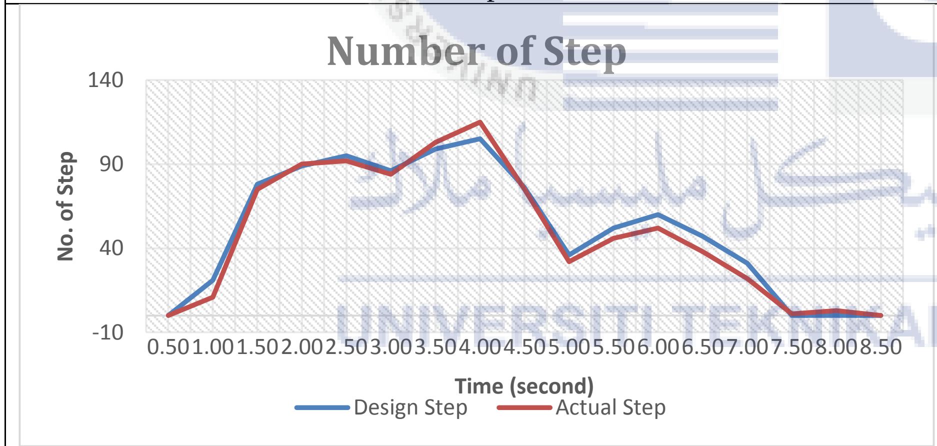


Figure 72: Number of Step for Third Out of Four Experiment for S-Shape

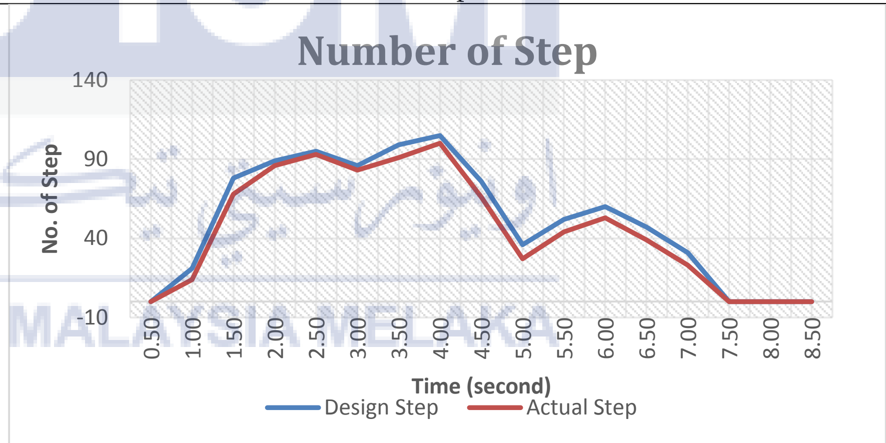


Figure 73: Number of Step for Forth Out of Four Experiment for S-Shape

S-Shape

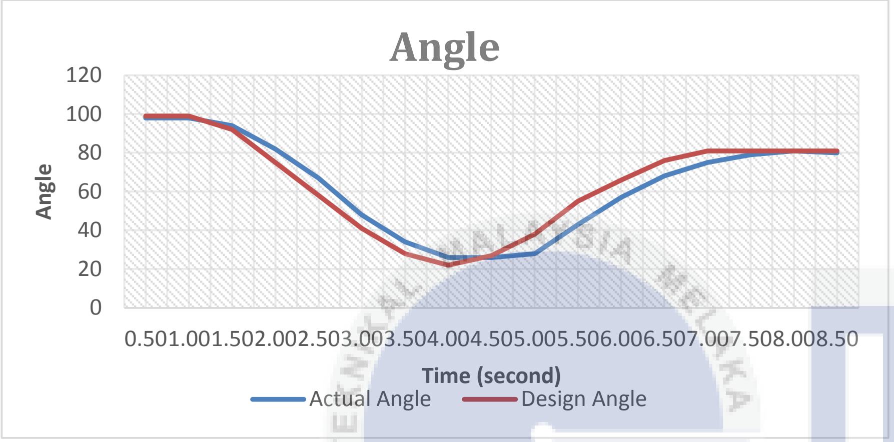


Figure 74: Facing Angle for First Out of Four Experiment for S-Shape

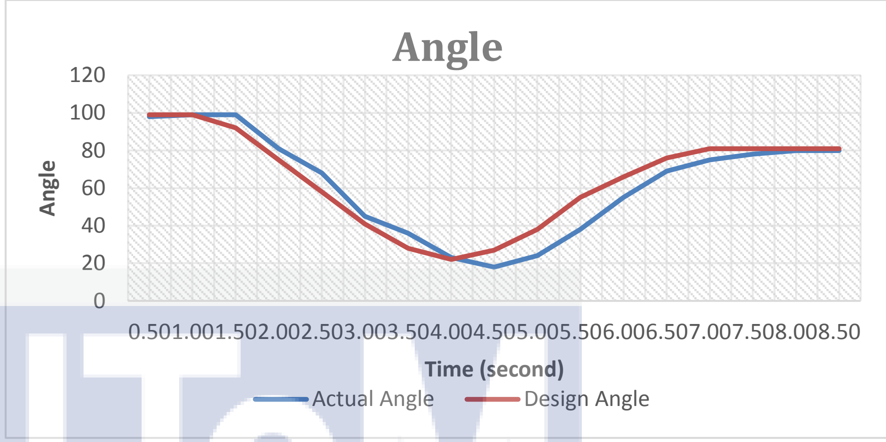


Figure 75: Facing Angle for Second Out of Four Experiment for S-Shape

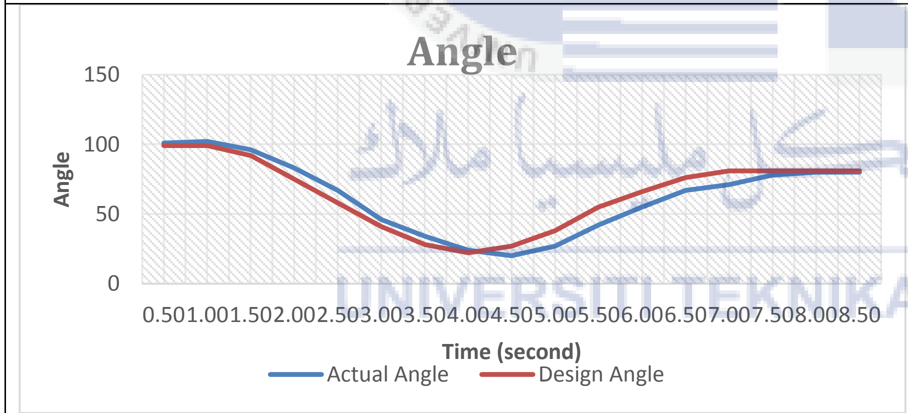


Figure 76: Facing Angle for Third Out of Four Experiment for S-Shape

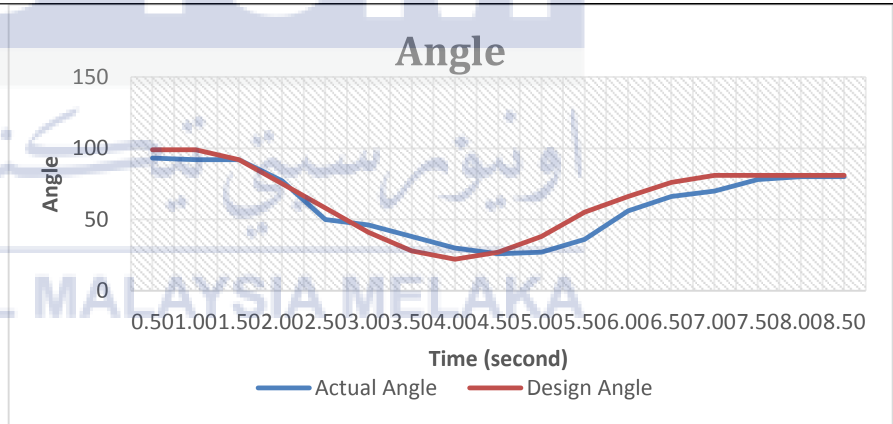


Figure 77: Facing Angle for Forth Out of Four Experiment for S-Shape

Method 2 Straight Line

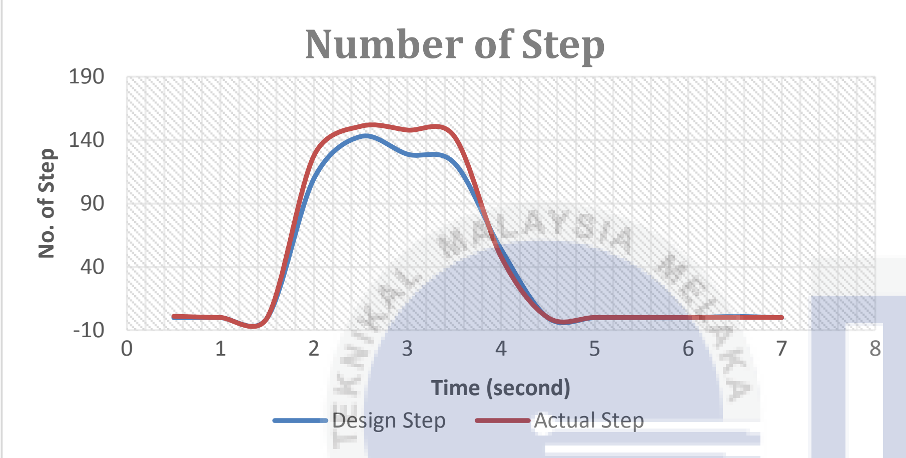


Figure 78: Number of Step for First Out of Four Experiment for Straight Line

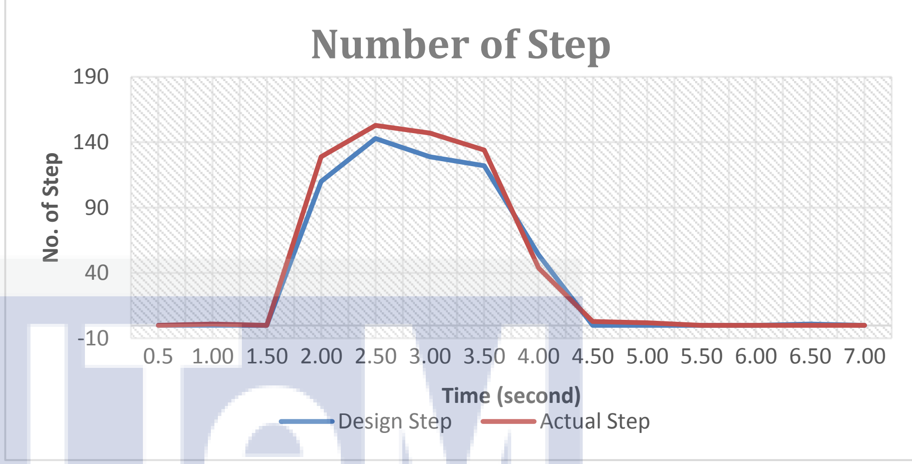


Figure 79: Number of Step for Second Out of Four Experiment for Straight Line

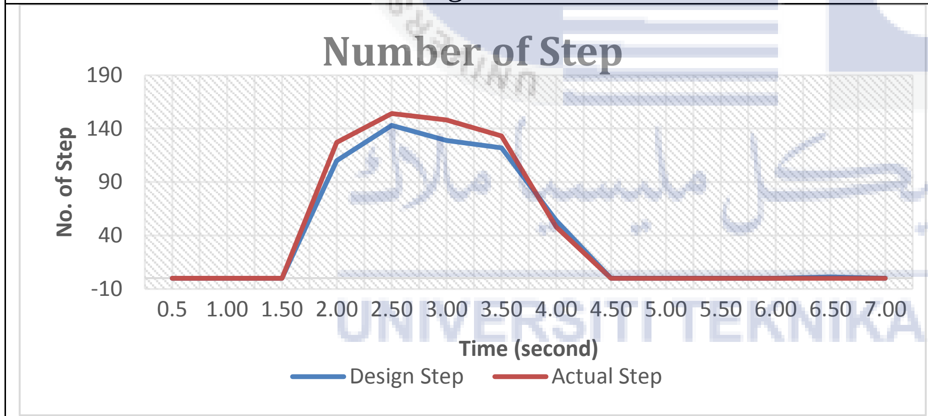


Figure 80: Number of Step for Third Out of Four Experiment for Straight Line

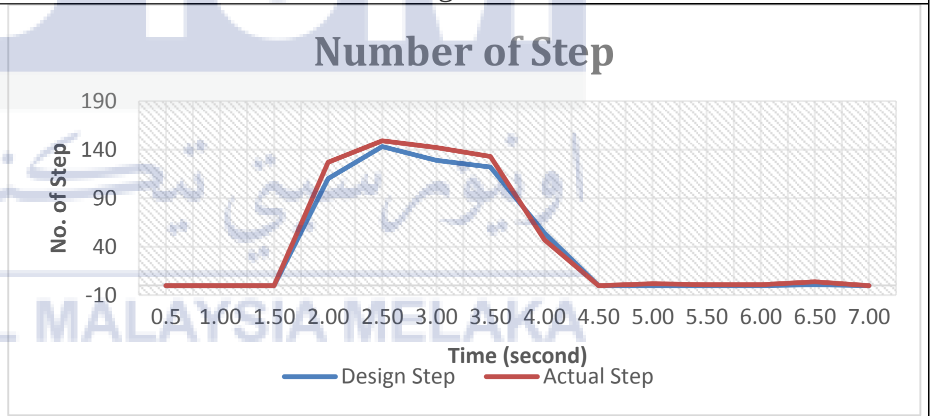


Figure 81: Number of Step for Forth Out of Four Experiment for Straight Line

Curve

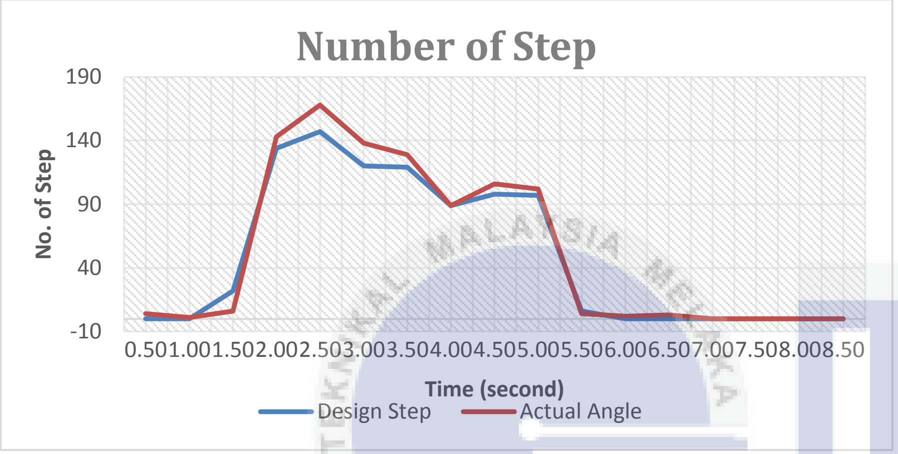


Figure 82: Number of Step for First Out of Four Experiment for Curve

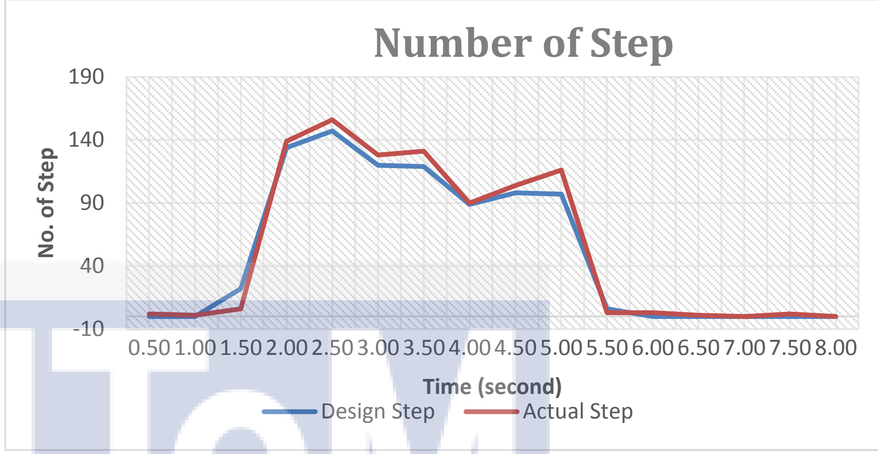


Figure 83: Number of Step for Second Out of Four Experiment for Curve

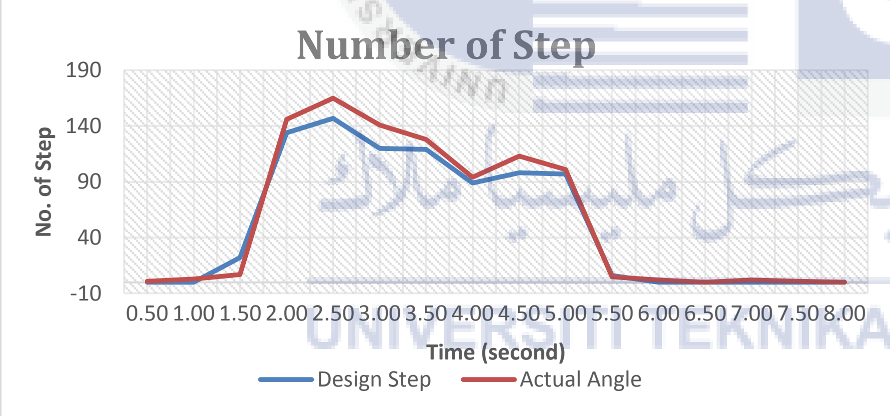


Figure 84: Number of Step for Third Out of Four Experiment for Curve

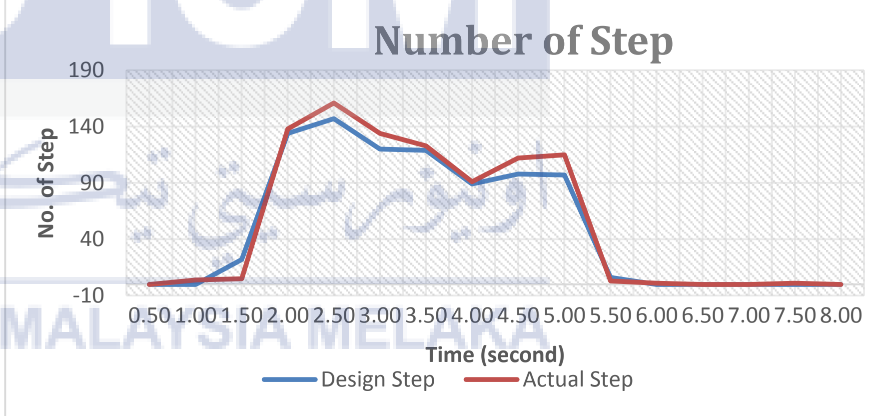


Figure 85: Number of Step for Forth Out of Four Experiment for Curve

S-Shape

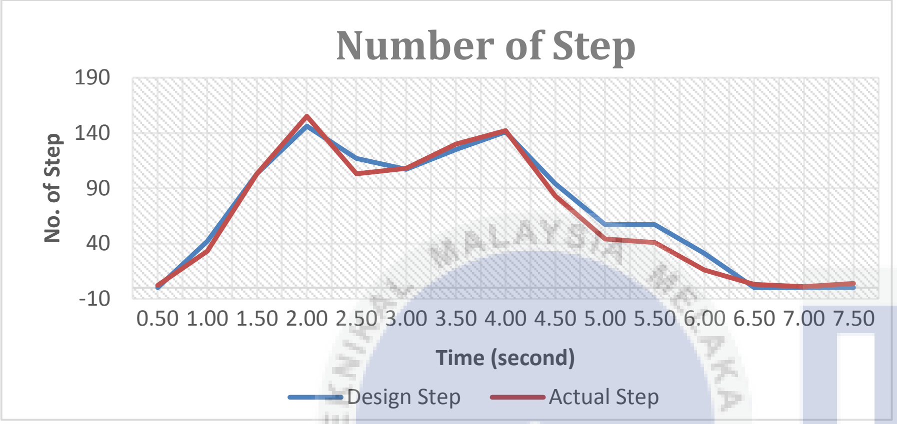


Figure 86: Number of Step for First Out of Four Experiment for S-Shape

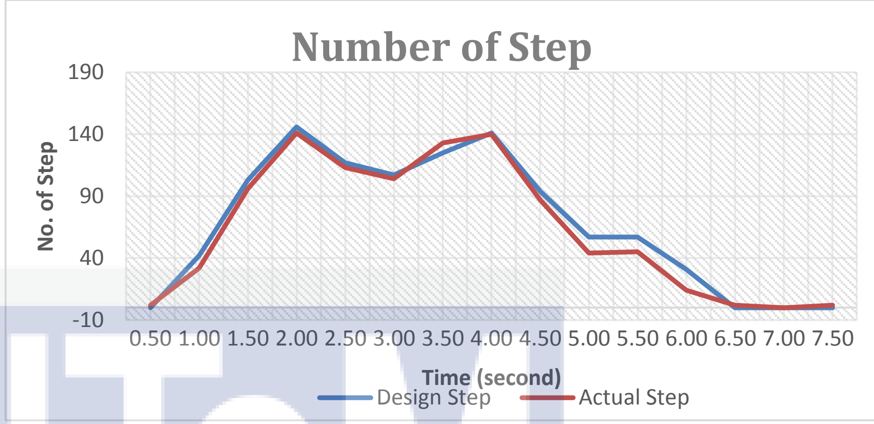


Figure 87: Number of Step for Second Out of Four Experiment for S-Shape

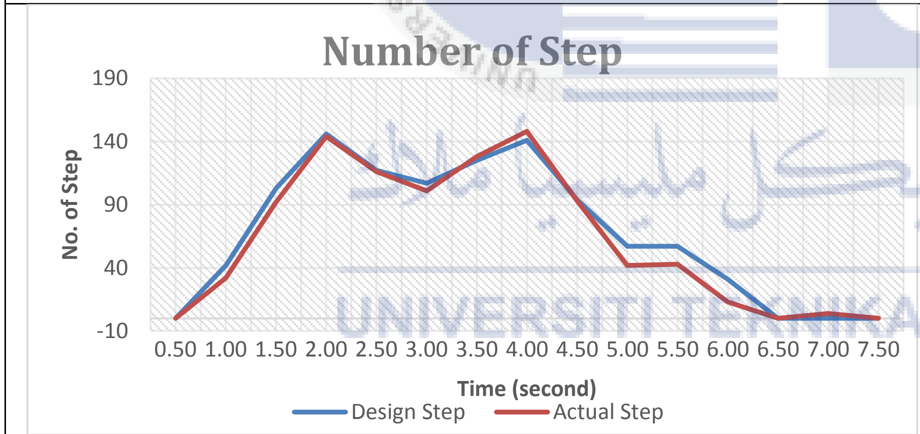


Figure 88: Number of Step for Third Out of Four Experiment for S-Shape

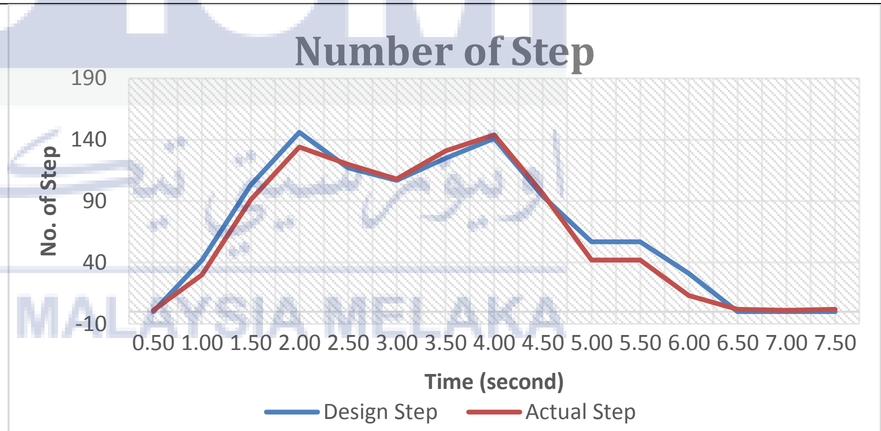


Figure 89: Number of Step for Forth Out of Four Experiment for S-Shape

APPENDIX B

Gantt chart of Project Planning

Project Activities	2014				2015					
	Sept.	Oct.	Nov.	Dec.	Jan.	Feb.	Mar.	Apr.	May.	Jun.
Research and study topic	■	■								
Study Journal	■	■	■							
Theory Use to Record and Execute Data	■	■	■							
Component Selection	■	■	■							
Type of Encoder	■	■	■							
Type of Angle Sensor	■	■	■							
Build up Prototype				■	■	■	■	■	■	
Programming Task		■	■	■	■	■	■	■	■	■
Experiment		■	■	■	■	■	■	■	■	■
Encoder		■	■	■	■	■	■			
Compass Sensor				■	■	■	■	■		
Trajectory Motion					■	■	■	■		
Analysis Performance of Prototype							■	■	■	■
Prepare Report and Presentation								■	■	■