“ I hereby declare that I have read through this report entitle “Side by side speed and distance control person following robot” and found that it has comply the partial fulfilment for awarding the degree of Bachelor of Electrical Engineering (Mechatronic)

Signature : ……………………............................

Supervisor’s Name : ……………………………………..

Date : ……………………………………..

SIDE BY SIDE SPEED AND DISTANCE CONTROL FOR PERSON FOLLOWING ROBOT

TEOH YEW CHONG

A report is submitted in partial fulfillment of the requirement for the Bachelor of Mechatronic Engineering

Faculty of Electrical Engineering

UNIVERSITY TEKNIKAL MALAYSIA MELAKA

YEAR 2015

I declare that this report entitle "Side by Side Speed and Distance Control Person Following Robot" 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

First of all, I would like to express appreciation to Miss Nur Maisarah Binti Mohd Sobran for serving as my supervisor of this project and giving me opportunity to learn and apply engineering knowledge in this project. Her guidance encourages me to accomplish my work on time. With her valuable suggestion and comment that enable me to write a good final year report. Besides, I also would like to thanks to my friend, Koh Guan Keong for technical guidance, sharing his knowledge and advice in this project. Lastly, I would like to thanks to those who are giving me support and motivation to accomplish this project.

Abstract

Nowadays, the elderly in whole world is increasing year by year. Therefore, assistive robot is introduced and applied in many applications such as in hospital and home to help elderly. There are two types of person following robot; robot following behind and human and side by side person following robot. But only side by side person following that has interact more with human compare to robot following behind human. However, the problems will occur when design a side by side person following robot which is the distance between human and robot when robot following target human. Besides, speed control of side by side person following robot is also necessary. There are three objectives in this project which are to develop prototype of a side by side person following, design a fuzzy logic controller of speed control on robot and analysis performance of robot in term of distance and speed control. In methodology, Arduino Mega ATmega1280 as microcontroller, ultrasonic sensor as distance measurement and brushless DC motor with encoder as motor to move robot are used to build a robot. First experiment is to determine the rpm of motor by adjusted the input PWM. Second experiment is to test ultrasonic sensor and speed control on prototype while the third experiment is to determine the performance of side by side person following robot. The result of first experiment is the PWM able to control rpm of motor. Next, the result of second experiment showed that the maximum distance measured by front sensor and back sensor are 301cm and 316cm. Moreover, overshoot occurred in the result of experiment 2.

Abstrak

Pada masa kini, orang tua di seluruh dunia meningkat setiap tahun. Oleh itu, robot bantuan diperkenalkan dan digunakan dalam pelbagai aplikasi seperti di hospital dan rumah untuk membantu warga tua. Terdapat dua jenis robot yang boleh ikut manusia; robot berikut di belakang manusia dan robot yang ikut di sampingan manusia. Tetapi hanya robot yang ikut di sampingan manusia mempunyai lebih berinteraksi dengan manusia daripada robot berikut di belakang manusia. Walau bagaimanapun, masalah akan berlaku apabila mereka bentuk robot yang ikut di sampingan manusia yang boleh menjarakkan jarak di antara manusia dan robot apabila robot sedang ikut manusia. Selain itu, kawalan kelajuan dengan algoritma kawalan robot yang ikut di sampingan manusia juga diperlukan. Terdapat tiga objektif dalam projek ini iaitu untuk membangunkan prototaip robot yang ikut di sampingan manusia, mereka bentuk algoritma kawalan untuk kawalan kelajuan pada robot berkaitan dan analisis prestasi robot dari segi jarak kawalan dan kelajuan kawalan. Dalam metodologi, Arduino Mega ATmega1280 sebagai pengawal mikro, sensor ultrasonic sebagai pengukuran jarak dan brushless DC motor dengan encoder sebagai motor untuk menggerakkan robot. Eksperimen pertama adalah untuk analisis rpm motor diselaraskan oleh PWM input. Eksperimen kedua adalah untuk menguji sensor ultrasonik dan kawalan kelajuan pada prototaip manakala eksperimen yang ketiga ialah untuk mengujikan prestasi robot yang ikut di sebelah manusia. Hasil daripada eksperimen pertama adalah PWM yang dapat mengawal rpm motor. Seterusnya, hasil daripada eksperimen kedua menunjukkan bahawa maksimum jarak pengesanan bagi sensor depan dan sensor belakang adalah 301cm dan 316cm. Selain itu, terlajak berlaku ketika menjalankan experiment kedua.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE
	ACKNOWLEDGEMENT	**i**
	ABSTRACT	**ii**
	ABSTRAK	**iii**
	TABLES OF CONTENTS	**iv**
	LIST OF TABLES	**vi**
	LIST OF FIGURES	**vii**
	LIST OF ABBREVIATIONS	**ix**
	LIST OF APPENDICES	**x**
1	**INTRODUCTION**	**1**
	1.1 Motivation	1
	1.2 Problem Statement	2
	1.3 Objective	3
	1.4 Scope	3
2	**LITERATURE REVIEW**	**5**
	2.1 Introduction	5
	2.2 Review on Previous Study of Person Following Robot	5
	2.2.1 Comparison	12
	2.2.2 Summary	13
3	**METHODOLOGY**	**14**
	3.1 Introduction	14
	3.2 General Overview of Hardware	14
	3.3 Material	17
	3.3.1 Arduino Mega ATMega 1280	17
	3.3.2 Brushless DC motor with Encoder	18

	3.3.3 Motor Driver BLHD50K	19
	3.3.4 Ultrasonic Sensor	20
	3.4 Circuit	22
	3.5 Software	23
	3.6 Fuzzy Logic	24
	3.6.1 Input	24
	3.6.2 Output	25
	3.6.3 Fuzzy Rules	26
	3.7 Overview of Side by Side Following Robot	28
	3.7.1 Flowchart of Program of Side by Side Person Following Robot	28
	3.7.2 Block Diagram	29
	3.8 Project Setup	30
	3.4.1 Experiment 1	30
	3.4.2 Experiment 2	32
4	**RESULT AND ANALYSIS**	**35**
	4.1 Overview	35
	4.2 Experiment 1	35
	4.2.1 Analysis of Data	38
	4.3 Experiment 2	40
	4.3.1 Analysis of Data	44
5	**CONCLUSION**	**47**
	5.1 Conclusion	47
	5.2 Recommendation	47
	REFERENCES	**49**
	APPENDICES	**50**

LIST OF TABLES

TABLE	TITLE	PAGE
2.1	The details of previous study	12
3.1	Other material in hardware	20
3.2	Software used to build side by side person following robot	23
3.3	Membership function for ultrasonic sensor	25
3.4	Membership function for PWM of motor	26
4.1	Percentage error of distance for front sensor	40
4.2	Percentage error of distance for back sensor	42

LIST OF FIGURES

FIGURE	TITLE	PAGE
1.1	Graph of percentage of total population versus year	1
2.1	People walking side by side with a mobile robot	6
2.2	Table I and II about the linguistic term and fuzzy rules for reference linear velocity	7
2.3	Table III and IV about the linguistic term and fuzzy rules for turning-gain	8
2.4	Example of fuzzy rule base used in study	9
2.5	Baseline of stereo vision system	10
2.6	Three speed stages of human and robot	11
3.1	Overview of side by side person following robot hardware	14
3.2	CAD drawing of side by side person following robot hardware	16
3.3	Hardware of side by side person following robot	17
3.4	Arduino Mega ATmega 1280	18
3.5	BLH450K DC motor	18
3.6	Speed of motor with external DC motor	19
3.7	Motor driver BLHD50K	19
3.8	Ultrasonic Ranging Module HC-SR04	20
3.9	Connection Arduino with component	22
3.10	Circuit of side by side person following robot	22
3.11	Elements of fuzzy logic	24
3.12	Matlab Fuzzy Inference System	24
3.13	Rules set in Matlab Fuzzy Inference System	27

FIGURE	TITLE	PAGE
3.14	Flowchart of side by side person following robot distance control system	28
3.15	Block diagram of side by side person following robot	29
3.16	The circuit of brushless DC motor with motor driver and Arduino	31
3.17	Schematic circuit of brushless DC motor	31
3.18	Experiment 2 Setup	32
3.19	Location of hardware in a room	33
3.20	Experiment 2 setup and hardware tested in a room	33
4.1	First reading of RPM versus Duty cycle	34
4.2	Second reading of RPM versus Duty cycle	35
4.3	Third reading of RPM versus Duty cycle	35
4.4	Fourth reading of RPM versus Duty cycle	36
4.5	Fifth reading of RPM versus Duty cycle	36
4.6	Sixth reading of RPM versus Duty cycle	37
4.7	Average reading of RPM versus Duty cycle	37
4.8	Distance from front sensor versus RPM	39
4.9	Distance from back sensor versus RPM	39
4.10	Overshoot in graph of distance versus RPM	40
4.11	Target moving location	44

LIST OF ABBREVIATIONS

RPM	=	Revolution Per Minute
PWM	=	Pulse-Width Modulation

LIST OF APPENDICES

APPENDIX	TITLE	PAGE
A	Grantt Chart	50
B	Programming for experiment of motor speed control	51
C	Programming of experiment 2	53
D	Data Result of experiment 2	71

CHAPTER 1

INTRODUCTION

1.1 Motivation

Nowadays, assistive robot is widely developed to assist human in daily life such as use to help patient in hospital or elderly in home nursing. But if the assistive robot can have interacted with human, the robot can assist human more effectively [1]. Besides, assistive robot also is introduced by Japan government for the purpose of assist elderly in daily life, it is due to from the World Bank population aging research Japan has a highest percentage of aging of total population (25%) compare to other country [2] [3]. Moreover, in Malaysia the data show 1.5 million Malaysian are elderly.

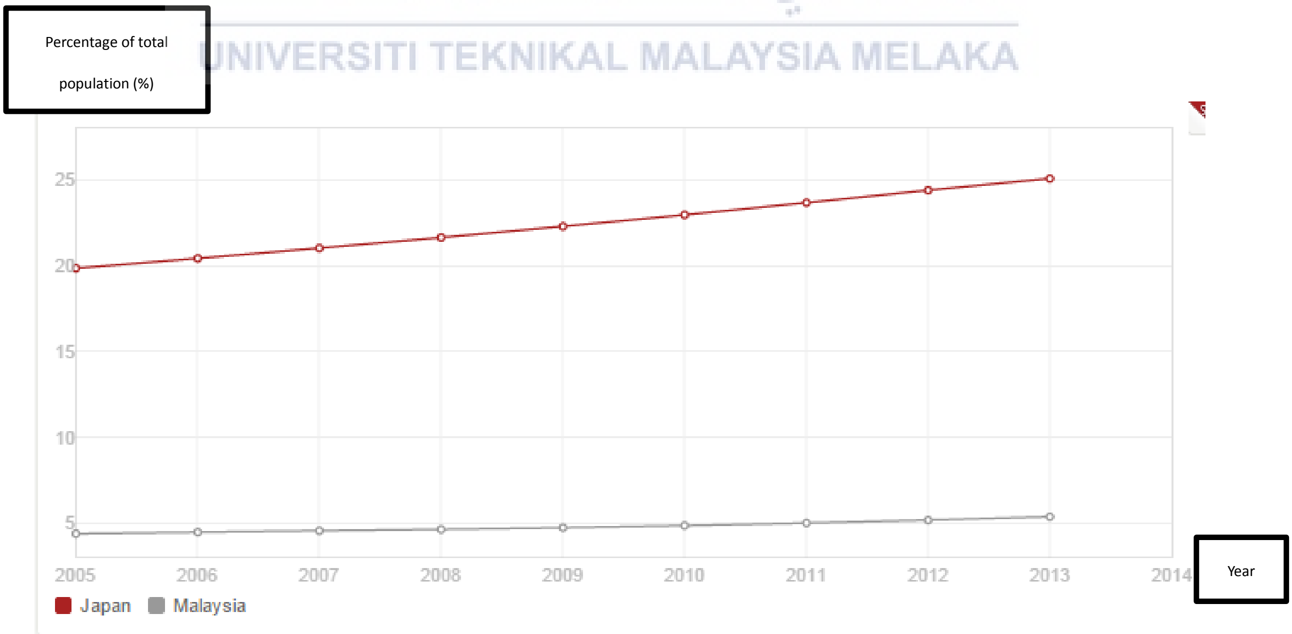


Figure 1.1: Graph of percentage of total population versus year

Person following robot is one of the assistive robot which can follow and assist human. There are 2 types of person following robot; first type is robot following behind human and second type is side by side person following robot. Robot following behind human has the most researches compare to side by side person following robot. But robot following behind human has less interacted with human due to this robot only follow at behind of human and when human walking, human does not know the status of robot whether it is follow or not. So, side by side person following robot is most suitable due to it is following target human beside him/her has more interact with human.

This type of robot can widely develop in many applications such as factories, office and supermarket. Side by side person following robot can be used in office, work as an assistant of human help them record and remind human. Besides, side by side person following robot also can use in the field of tourism such as guide human walking to reach their destination. Furthermore, side by side person following robot as an assistive robot can use as home nursing robot. [4]

1.2 Problem statement

From the previous researches, most only focus on the robot following behind a human, however it seems not much advantage for human since human and robot less interact at all. So to let human and robot have interacted is to design the robot side by side following a human. Then both human and robot can get information from each other easily.

Furthermore, a side by side person following robot that able to follow target human and can always besides target when following target. Therefore the motor speed control is needed to follow target human. This part of speed control is depends on the signal of human tracking sensor. The signal of human tracking sensor sends to microcontroller and then is calculated. Afterward the feedback signal will control speed of motor.

When a side by side person following robot is following a target human, the robot must able to measure distance, d between robot and human. In order to to control speed of robot, a suitable control algorithm is needed with inputs signal from distance sensor, so that output is the speed of both motors, v_m, can be adjusted according the sensors signal. For example, if target is far around 150cm from robot, the robot can speed up due to target may walk away from the view of distance sensor. If the target is near around 50cm from robot, the robot may using medium speed to follow human.

1.3 Objective of Project

There are three objectives in this project.

i. To develop prototype of a side by side person following robot
ii. To design control algorithm for a side by side person following robot that has speed control.
iii. To analysis the performance of developed side by side person following robot in term of speed and distance control.

1.4 Scope

The scopes of this project are outlines as follows.

1. From this project is to design a side by side person following robot which able to measure distance between human when following human and able to control the speed to follow human.

2. Besides, the path of human walking is straight line only when followed by side by side person following robot.

3. Moreover, there is no obstacle between human and side by side person

following robot when robot is following target human. The side by side person following robot is tested in a room of no obstacle.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Person following robot is a famous research due to it is very useful in assist human future life. The information from the previous research is use as guide to build a better side by side person following robot in this project. This chapter is about review, comparison and summary of previous researches on person following robot.

2.2 Review on previous study of person following robot

A study about side by side person following robot with the objective of able to follow beside target person and can keep a safe distance between target person and robot when both robot and human are moving. In this study, laser range finder with the detection range of 30m is used to detect target human. One unit of laser range finder can cover 180 degree of field of view, so two units laser range finder will built inside the robot which can cover 360 degree means it can detect surrounding of robot. Eight parameters are considered to design a better person following robot such as social relative distance, relative angle, relative velocity, distance to obstacle, sub goal, velocity, angular velocity and acceleration [5].

Besides, the robot is tested in three conditions by using three methods. First method is the standard prediction method which is to guess next position of target by using mathematical methods, linear extrapolation of the velocity. Next, second method is the self-anticipation. This method is use to plan the next step of robot movement to enlarge its utility with preferred linear velocity, angular velocity and acceleration. After these two methods the next method is the partner and self-anticipation. It plans next step movement of robot to maximize robot utility and target utility. There are few shortcomings in the study [5] such as parameters are not systematically calibrate, standard prediction method and the self-anticipation have the problem of lacks of ability to make the robot take a lead toward the target. Figure 2.1 is shows a success side by side person following robot by considering eight parameters [5].

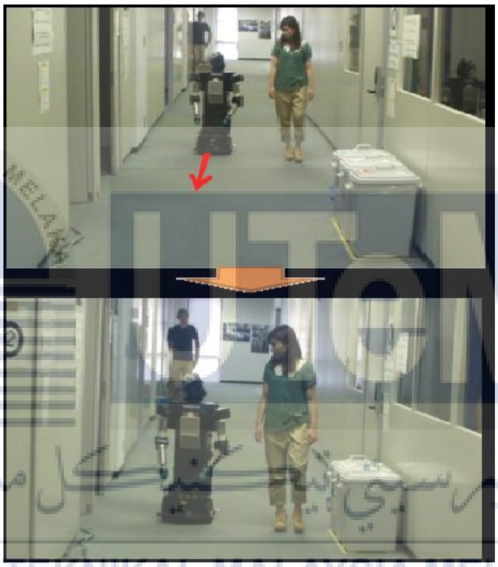


Figure 2.1: People walking side by side with a mobile robot

Source:[5]

In [6] proposed a person following robot with fuzzy based control system. The paper used Radio-frequency identification (RFID) with antenna is able to detect the position target human by giving the target person with an ID and stereo camera to detect human. But when human is out of the detection range of stereo camera, only can depends RFID detect the position target human. Moreover, two control strategies are used in this study; intelligent control strategy and fuzzy based controller. Intelligent control strategy is to control the robot for turning toward target when target is away from robot, two parameters are used which are the turning gain and linear velocity. The direction and distance from tracking part is obtained and from these two to do adjustment for turning gain and linear velocity. Then, the turning gain is adjusted online to make sure the target is in the view of camera, as the direction between target and robot is increased the turning gain is also increased while turning radius is decreased. There are two main parts using fuzzy based controller to control the motion of person following robot; reference linear velocity and turning-gain.

Furthermore, in order to get the reference linear velocity, $v_{re,}$ two inputs is considered and there are the distance, $x_{r,}$ between target and robot when both are moving and the vertical velocity of target, v_x. The rules between distance, x_r, and vertical velocity of target, v_x to get reference linear velocity is shown in Figure 2.2 with linguistic term. For example, if the distance, x_r, is very near (VN) and the vertical velocity of target, v_x is very slow (VS), then reference linear velocity, v_{re} is very slow (VS) [6].

TABLE I

LINGUISTIC TERMS FOR REFERENCE LINEAR VELOCITY.

x_r	Linguistic terms	v_x	Linguistic terms	v_{re}	Linguistic terms
VC	Very Close	VS	Very Slow	VS	Very Slow
C	Close	S	Slow	S	Slow
Z	Safe Distance	Z	Normal	Z	Normal
F	Far	F	Fast	F	Fast
VF	Very Far	VF	Very Fast	VF	Very Fast

TABLE II

FUZZY LOGIC FOR REFERENCE LINEAR VELOCITY CONTROLLER.

v_{re}		v_x /m • s^{-1}				
		VS	S	Z	F	VF
x_r /m	VN	VS	VS	S	S	Z
	N	VS	S	S	Z	Z
	Z	VS	Z	Z	F	F
	F	Z	F	VF	VF	VF
	VF	F	VF	VF	VF	VF

Figure 2.2: Table I and II about linguistic term and fuzzy rules for reference linear velocity

Source:[6]

For the turning-gain controller, the absolute direction, y_r and horizontal velocity of target, v_y will considered as inputs to obtain output turning-gain of robot, k. The rules used to obtain turning-gain, k is shown in Figure 2.3. The example of the rules is if the absolute direction, y_r is negative far (NF) and velocity of target, v_y is negative slow (NS), and then the output turning-gain, k is large [6].

TABLE III
LINGUISTIC TERMS FOR TURNING-GAIN ADJUSTMENT CONTROLLER.

y_r	Linguistic terms	v_y	Linguistic terms	k	Linguistic terms
NF	Negative Far	NF	Negative Fast		
NN	Negative Near	NS	Negative Slow	S	Small
Z	Center	Z	Zero	N	Normal
PN	Positive Near	PS	Positive Slow	L	Large
PF	Positive Far	PF	Positive Fast		

TABLE IV
FUZZY LOGIC FOR TURNING-GAIN ADJUSTMENT CONTROLLER.

k		v_y /m • s^{-1}				
		NF	NS	Z	PS	PF
y_r /m	NF	L	L	N	N	N
	NN	L	L	N	S	S
	Z	N	N	S	N	N
	PN	S	S	N	L	L
	PF	N	N	N	L	L

Figure 2.3: Table III and IV about linguistic term and fuzzy rules for turning-gain

Source:[6]

Moreover, the other research about person following robot is proposed with goal of ability to follow target person and able to keep safe distance between target and robot when both are moving. The sensor used to track human position is IR based co-ordinate system, while the distance sensor used is ultrasonic sensor. Fuzzy rules is applied in this research which means the inputs of position (Left, Mid and Right) and distance (Near, Locked and Far) to obtain the output, robot velocity. The Figure 2.4 show the fuzzy rule based in the control system [7].

1. If (Direction is Left) and (Distence is Locked) then (Rigth_wheel is incr)(left_wheel is Mild) (1)
2. If (Direction is Right) and (Distence is Locked) then (Rigth_wheel is Mild)(left_wheel is inc) (1)
3. If (Direction is Mid) and (Distence is Near) then (Rigth_wheel is dec)(left_wheel is dec) (1)
4. If (Direction is Mid) and (Distence is Far) then (Rigth_wheel is incr)(left_wheel is inc) (1)
5. If (Direction is Mid) and (Distence is Locked) then (Rigth_wheel is Mild)(left_wheel is Mild) (1)
6. If (Direction is Right) and (Distence is Near) then (Rigth_wheel is dec)(left_wheel is inc) (1)
7. If (Direction is Left) and (Distence is Near) then (Rigth_wheel is incr)(left_wheel is dec) (1)
8. If (Direction is Right) and (Distence is Far) then (Rigth_wheel is dec)(left_wheel is inc) (1)
9. If (Direction is Left) and (Distence is Far) then (Rigth_wheel is incr)(left_wheel is dec) (1)
10. If (Direction is Left) and (Distence is Dangerous_distance) then (Rigth_wheel is Mild)(left_wheel is stop) (1)
11. If (Direction is Right) and (Distence is Dangerous_distance) then (Rigth_wheel is stop)(left_wheel is Mild) (1)
12. If (Direction is Mid) and (Distence is Dangerous_distance) then (Rigth_wheel is stop)(left_wheel is stop) (1)

Figure 2.4: Example of fuzzy rule based used in the study

Source:[7]

From those rules the person following robot is tested in three conditions such as target in centre location of robot, target is in left side of robot and target is in right side of target. At the first conditions, the robot able to follow up human speeds with full power and can keep a safe distance with target. Besides, the robot able to make a curve turn to follow target by controlling left and right motor in the next two conditions [7].

Furthermore, a study about person following Omni-directional vehicle robot is focus more on position of the target. So input of position controller is based on 2 stereo camera sensors and reference position which is X_R, Y_R and Z_R. Typically there are two methods of control to calculate the distance between target human and robot, namely, position-based control and features-based control but this study is focus on the position-based control. Image captured is extracted, geometric model and camera model is used to predict the target pose. The study apply PD controller for the calculation of the position controller, while the position reference is the distance between person following robot and target human. After that the output position controller is derived by using Jacobain matrix, and then will become the input of velocity controller. The RT-Linux is used for calculate the position information from the stereo camera position and control the motion of robot [8].

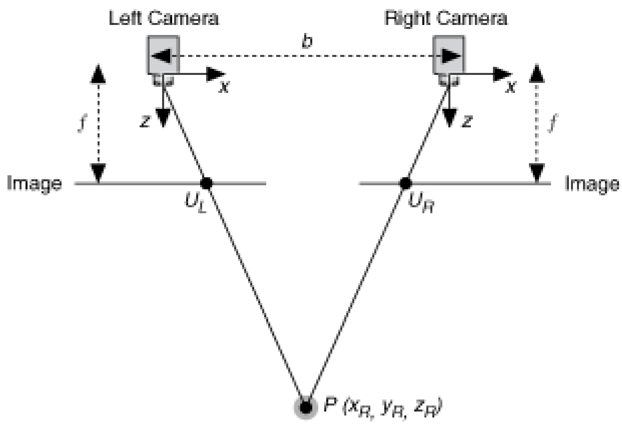


Figure 2.5: Baseline of stereo vision system

Source:[8]

The speed control of person following robot is mention in [9]. Speed control of this study is set according to three conditions such as death zone, speed control and adaptive deceleration. Death zone is a zone of errors occurs in robot like the distance control is different from the desired value. This is very important due to error is needed to consider building a better side by side person following robot. Furthermore, speed control is same as death zone where it also considered the response time of robot speed up and motion robot is slower than human walking speed. At the same time, adaptive deceleration algorithm is also considered depends on the speed decrease stage.

Three speed stages was introduced in the paper [9] and there are speed increase stage, speed saturation stage and speed decrease stage. These three stages are about the categories of robot speed and robot speed will changed according the output signal from the Kinect sensor. In order to find the error of human relative angle, PD controllers is used and apply on the servo motors control of robot. After the person following robot is done, few experiments is needed to test;

1. Target human move forward and reverse only for examining speed control.
2. Target human turn around the robot with constant distance is for testing angle control.
3. Target human walk along a straight path for speed control and angle control

testing.

4. Forward and reverse path.

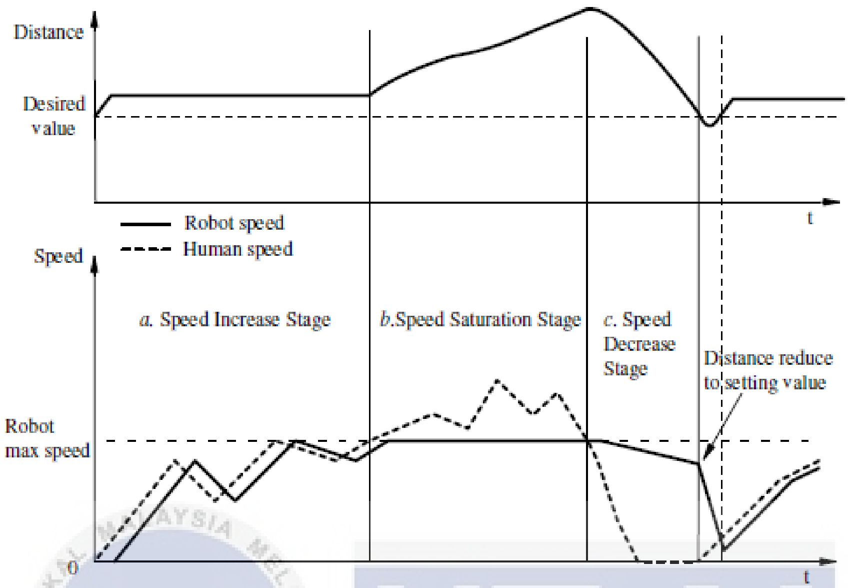


Figure 2.6: Three speed stages of human and robot

Source:[9]

2.2.1 Comparison of previous study

Table 2.1: The details of previous study

Title	Tracking algorithm	Distance algorithm	Control algorithm
How Do People Walk Side By Side? – Using a computational model of human behaviour	Laser range finder : Leg detection	Laser range finder: Distance detection	-
Fuzzy-based intelligent control strategy for a person following robot	Stereo camera: human tracking RFID: Position tracking	RFID: Position tracking	Fuzzy inference system
Modelling and Robust analysis of a fuzzy based person following robot	IR with co-ordinate: human detection	Ultrasonic sensor: Distance detection	Fuzzy inference system
A controller design on person following omni-directional vehicle robot	Stereo vision sensor: position tracking	Stereo vision sensor: position tracking	PD controller in position control
Robot human-following limited speed control	Kinect sensor: human joints detection	Kinect sensor: Distance detection	PD controller in angle control

2.2.2 Summary section of previous study

From the above researches can conclude that fuzzy logic system is more suitable use on the side by side person following robot. It is due to next step of human is unpredictable, only fuzzy logic can handle. Fuzzy logic control system will be used in speed control and position control. Besides, the Kinect sensor, one of stereo vision camera has the maximum range detection 4m longer range detection than ultrasonic sensor which is only maximum 3m range detection, but the Kinect sensor is much costly than ultrasonic sensor. Furthermore, the feedback from the journal [10] is the limitation measurement range of Kinect in depth and width. The different type of RFID have different range of detection and it is depends which type of RFID is used to determine position of target human. But the RFID is also expensive than ultrasonic sensor, so ultrasonic sensor is preferable component than can used in the side by side person following robot.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter is about the methodology of project. Main goal of this project is to build a prototype which able to follow target person and with control algorithm that able to keep a safe distance between human and robot when the robot moving.

3.2 General overview of hardware

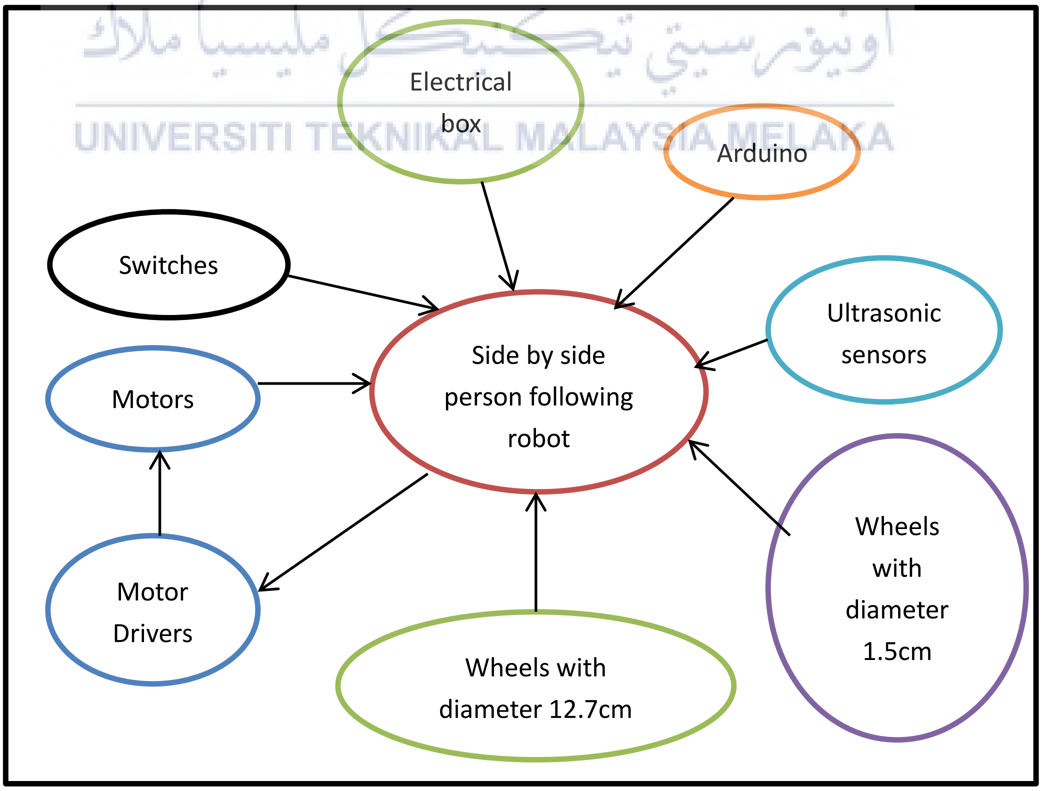


Figure 3.1: Overview of side by side person following robot hardware

For the prototype of side by side person following robot, Arduino will be used as the control center for whole robot. Besides, Arduino will be programmed to control the motor of the robot based on fuzzy logic controller. The fuzzy logic controller is actually based on the input from human tracking sensor and distance sensor.

In order to control the speed of motor accurately, a brushless DC motor with encoder and motor driver is used. Although Arduino can control the PWM, but the desired speed of a motor still cannot obtain 100% accurate. So motor with encoder is chosen due to with encoder, the speed of motor can be specified.

Besides, two ultrasonic sensors are installed on the left side of robot and separated with a distance 20cm due to and maximum separation of two sensors will increase the detection view of target.

Moreover, an electrical box is placed on top of prototype. An Arduino and motor drivers are installed inside the electrical box while two switches is installed beside of the box. One of switch is the power button for Arduino and the other switch is for motors.

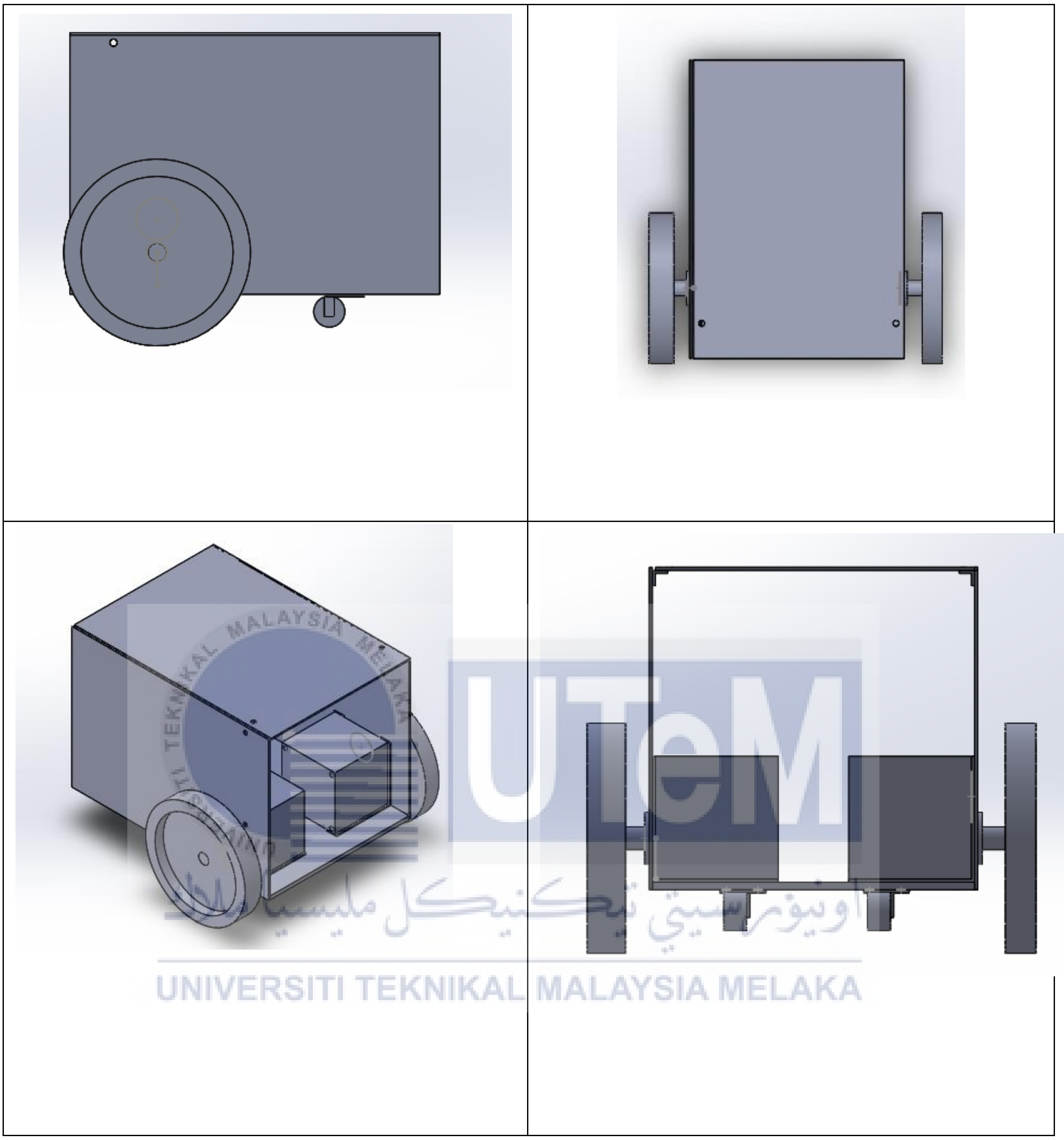


Figure 3.2: CAD drawing of side by side person following robot

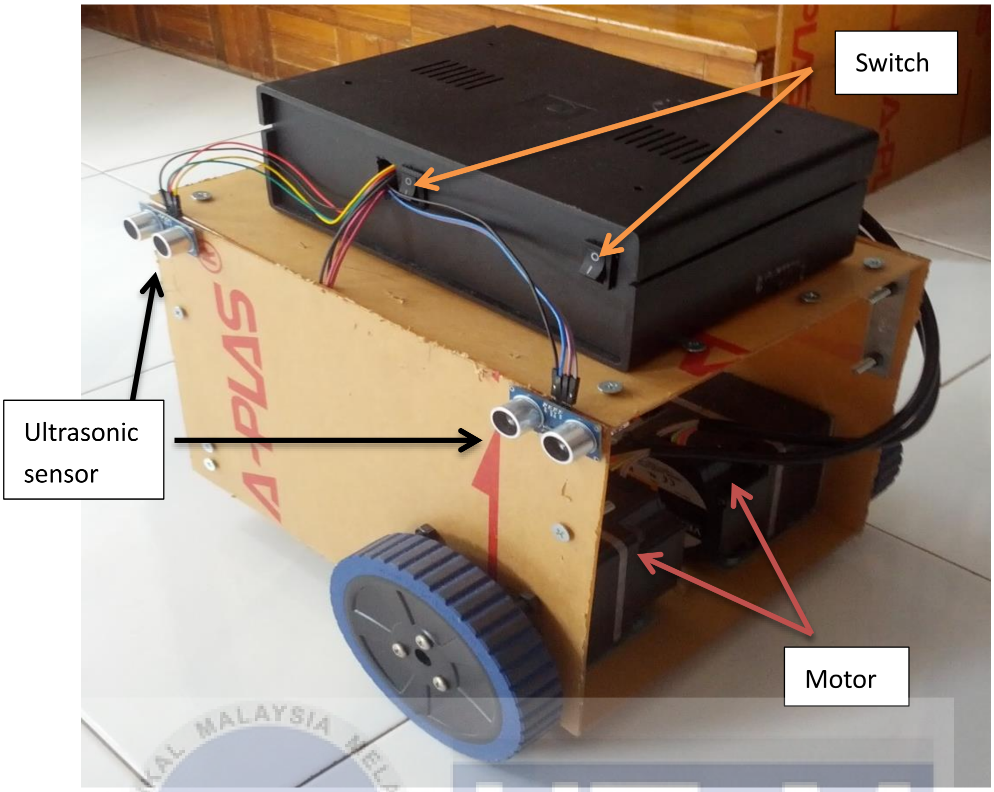


Figure 3.3: Hardware of side by side person following robot

3.3 Material

3.3.1 Arduino Mega ATmega 1280

Arduino Mega is a microcontroller board with 54 digital input/output pins, 16 analog inputs, 4 UARTs (hardware serial ports). 5V can be supply and operate in this microcontroller, but it may unstable. So the best input voltage is in the range between 7V to 12V. If more than 12V, the microcontroller will be overheated. Moreover, 15 out of 54 digital input/output pins of the Arduino Mega are PWM output pin which can use to control motor speed. Besides, it built with a 16MHz crystal oscillator and flash memory with 128KB.

Furthermore, it can be operated and programmed with software by connect it to a computer with a USB cable.

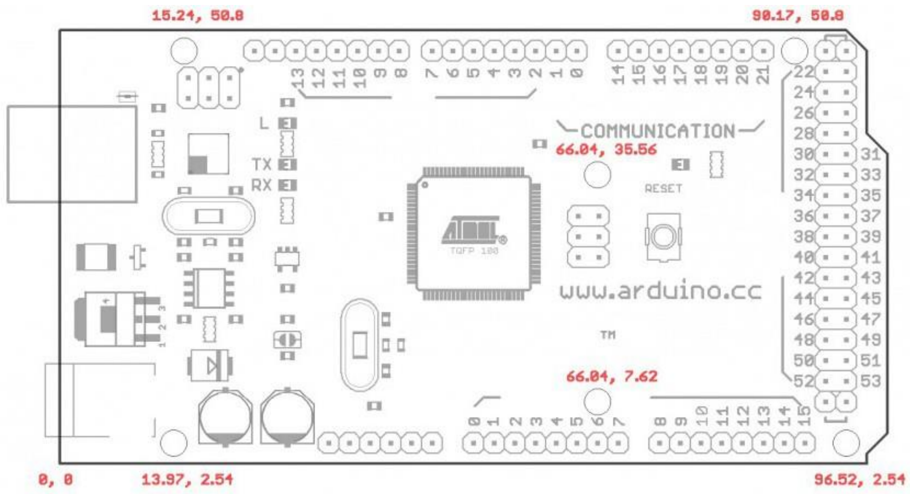


Figure 3.4: Arduino Mega with ATmega1280 microcontroller

3.3.2 Brushless DC motor with Encoder

Figure 3.5: BLH450K DC motor

BLH450K is used as actuator of side by side person following robot. Combination type parallel shaft gearhead is installed inside the motor. The speed reduction ratio is 100:1 used in the gearhead type. Furthermore, from the data sheet the range of motor speed is from 100 to 3000 r/min for the case of the motor turning in load condition. Besides, this motor needs motor driver with 24V power supply to activate it. Brushless DC motor with encoder is chosen due to with encoders, the accuracy of motor speed can be controlled more accurate.

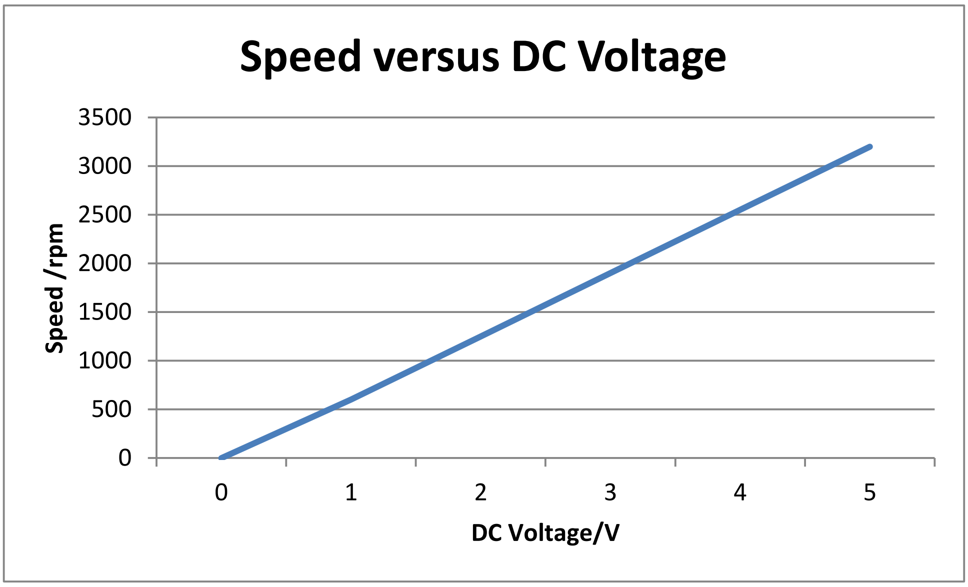


Figure 3.6: Speed of motor with external DC voltage

Figure 3.8 is about the theoretical graph from manual book of motor BLH450K [12]. DC voltage show in figure is the external DC supply use to signal the speed of motor. As the external DC voltage is increasing, the speed of motor is also increasing.

3.3.3 Motor Driver BLHD50K

Figure 3.7: Motor driver BLHD50K

Motor driver BLHD50K is a type motor driver only for oriental motor BLHM450K model. This type of motor driver has few external functions on the board of driver such as internal speed potentiometer and acceleration/deceleration time potentiometer. The function for internal speed potentiometer is to set the motor speed while the acceleration/deceleration time potentiometer has function to set motor started acceleration time and time of deceleration when motor stopped. Heat sink is installed to prevent overheating [11].

3.3.4 Ultrasonic Sensor

Figure 3.8: Ultrasonic Ranging Module HC - SR04

Ultrasonic Ranging Module HC - SR04 with the input power supply 5V able to detect in the range from 2cm to 4m. Besides, the surface are of objects must more than 0.5 m^2; otherwise it will affect accuracy of the measuring result.

Table 3.1: Other material in hardware

Material	Size	Unit	Image
Wheel	Diameter 127mm	2	
	Diameter 15mm	2	
Acrylic	297mm x 210mm (Thickness 2mm)	3	
	297mm x 210mm (Thickness 5mm)	1	

Screw with nut	Diameter 4mm	32	
	Diameter 2mm	8	
Electrical Box	225mm x 150mm	1	
Steel connector joint	78mm x 78mm	8	
	40mm x 40mm	2	

3.4 Circuit

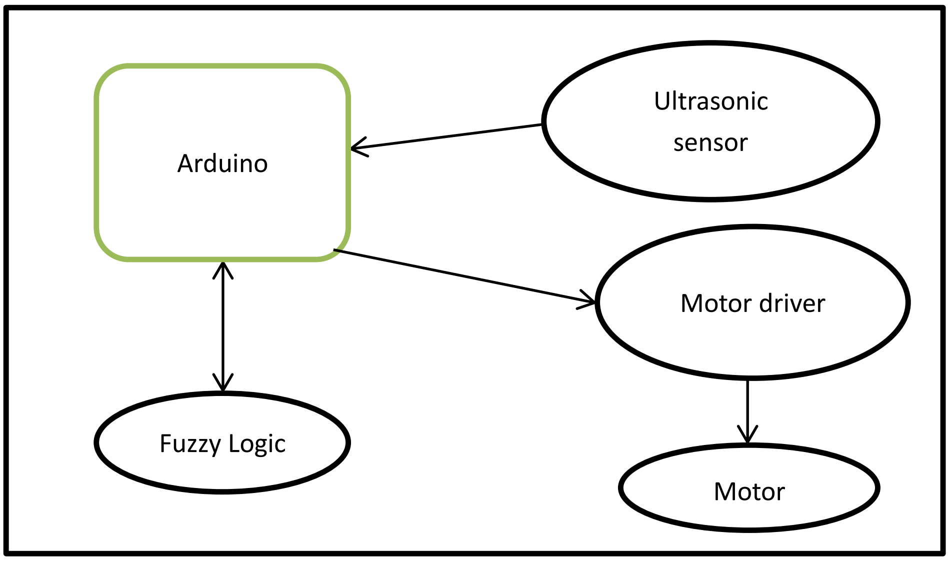


Figure 3.9: Connection Arduino with component

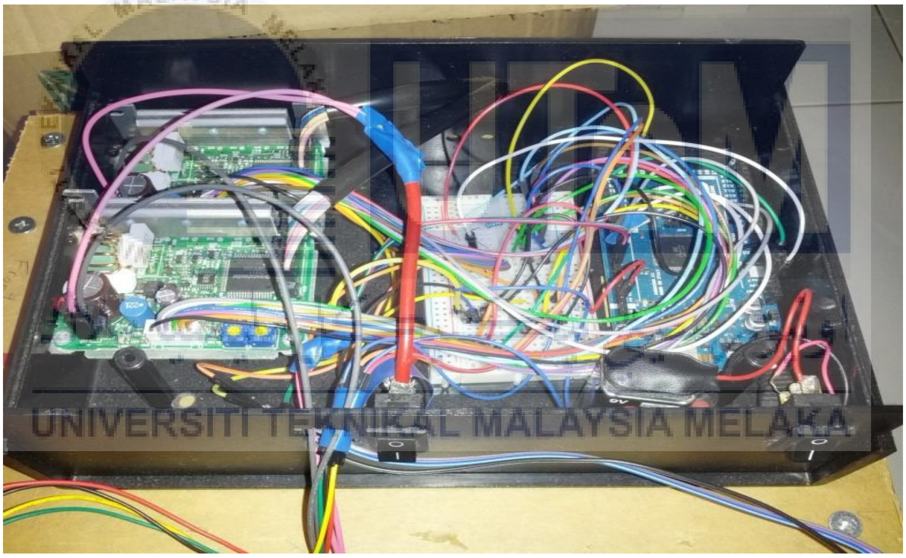


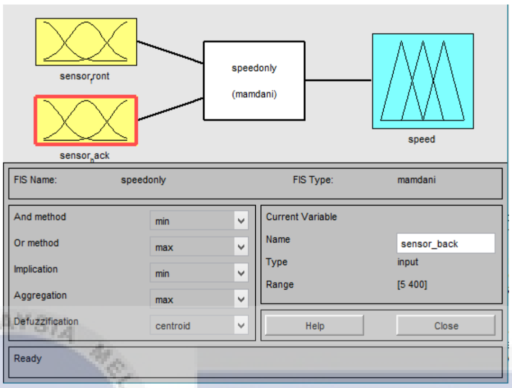
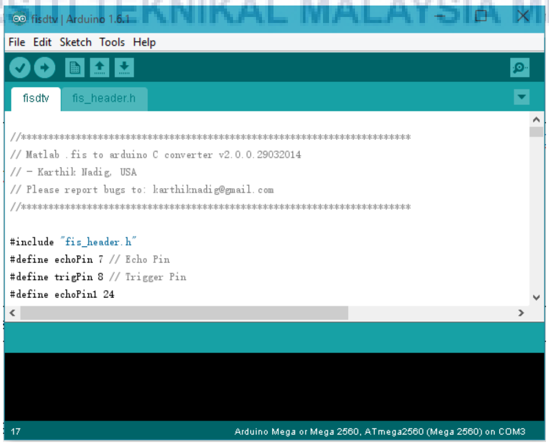
Figure 3.10: Circuit of side by side person following robot

The circuit design of side by side person following robot is shown in Figure 3.10. Arduino is the microcontroller for side by side person following robot and it will receive signal from ultrasonic sensor. The signal received by Arduino will be process based on fuzzy rules. After that, the other signal will be produced by Arduino to motor driver. Besides, motor will not connect directly with Arduino due to motor driver is controller for motor and it is to control speed of motor, direction of motor movement and protect motor. In this case, motor driver is received signal from Arduino and provide signal to control motor speed and direction.

3.5 Software

In this section will discuss function of the software used to program the side by side person following robot.

Table 3.2: Software used to build side by side person following robot

Software	Figure	Function
FIS Editor (Matlab tools)		To design speed controller by using fuzzy
Online software-MakeProto		To convert Matlab fuzzy system to Arduino C
Arduino software		To write coding and upload the coding to Arduino ATmega2560

3.6 Fuzzy Logic

A set of mathematical principles is represented based on degrees of membership function is called fuzzy logic. Fuzzy logic has 4 major elements; fuzzification, rule-base, inference mechanism and defuzzification interface. Besides, defuzzification use centroid method to obtain output value.

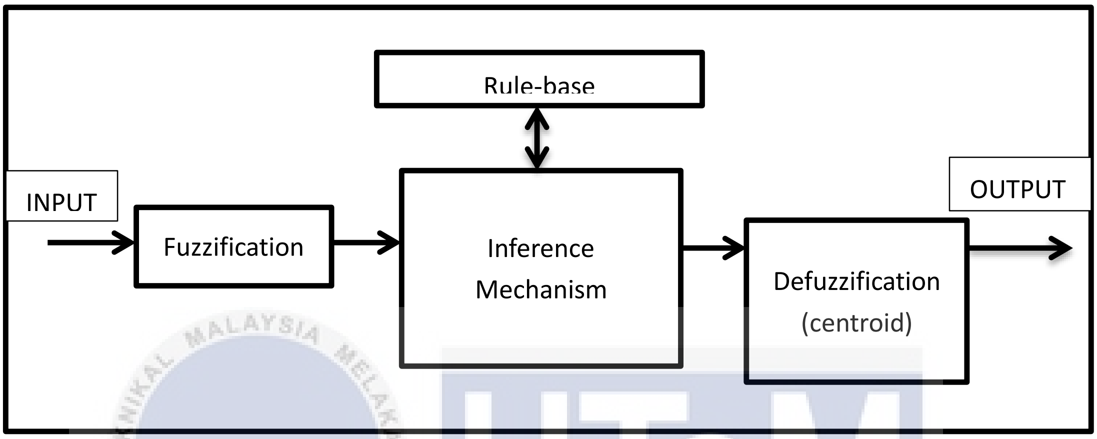


Figure 3.11: Elements of fuzzy logic

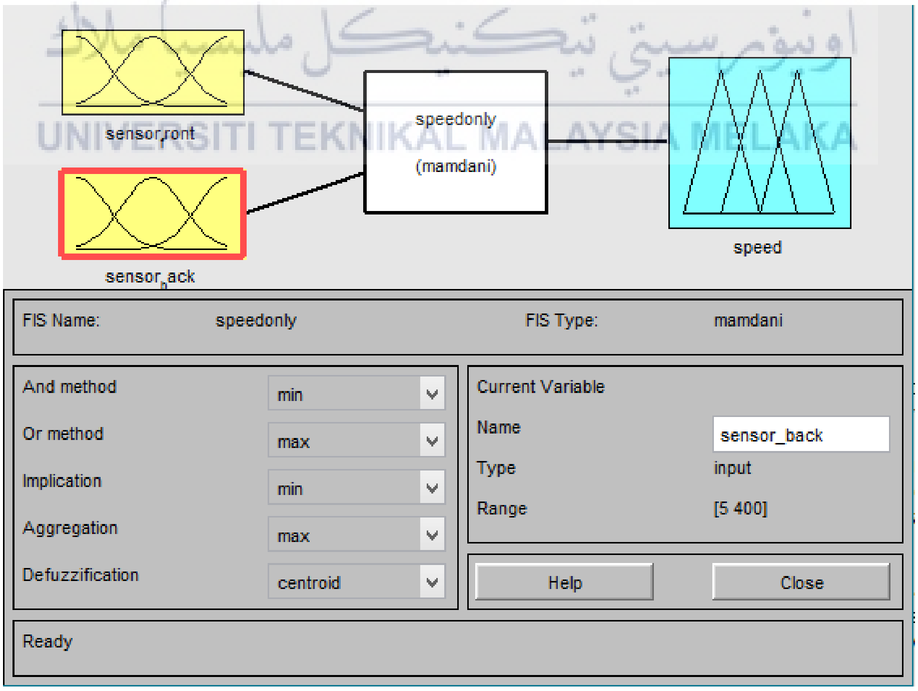
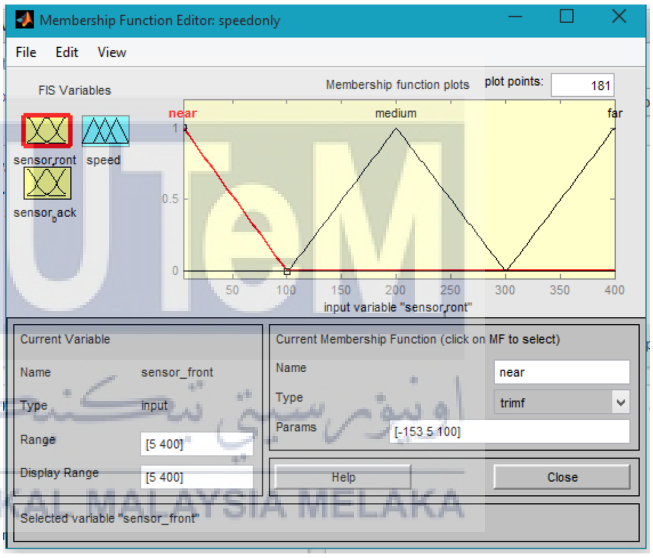
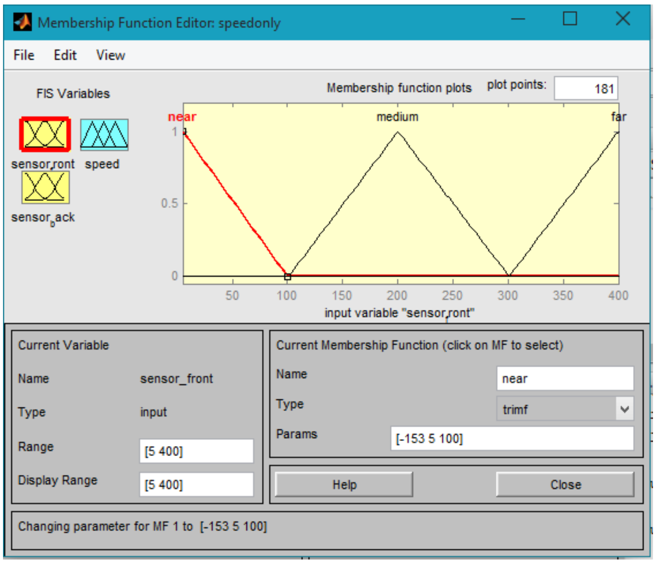


Figure 3.12:Matlab Fuzzy Inference System

3.6.1 Input

Distance detected by ultrasonic sensor is the input of fuzzy logic. One of ultrasonic sensor near the front of hardware will named as front sensor and the other near the back of hardware will named as back sensor. The range of membership function is from 5cm until 400cm maximum distance measurement of ultrasonic sensor. The membership function of ultrasonic sensor will listed in Table

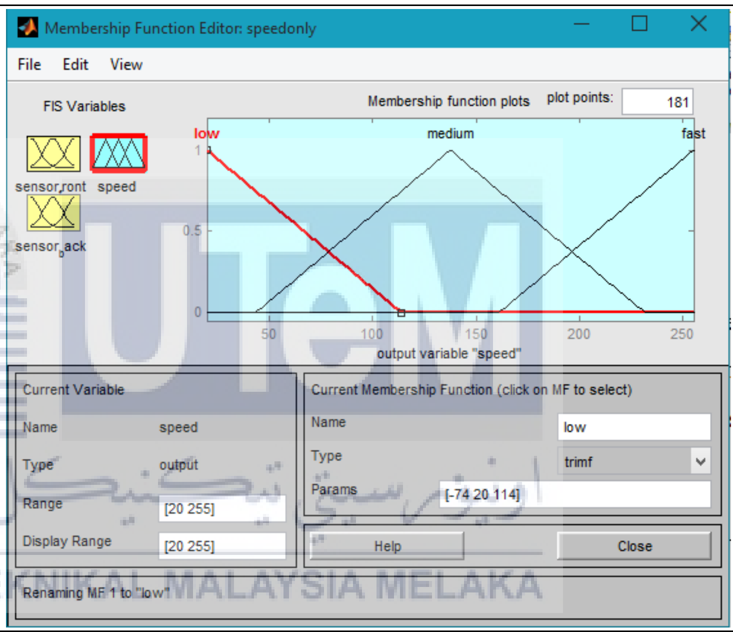
Table 3.3: Membership function for ultrasonic sensors

Name	Range of membership function	Figure
Front sensor	Near: 5 to 100 Medium: 101 to 300 Far: 301 to 400	
Back sensor	Near: 5 to 100 Medium: 101 to 300 Far: 301 to 400	

3.6.2 Output

Speed of motor is the only parameter which is needed to be controlled, so the speed as the output for fuzzy logic. The range of membership is based on value of PWM in Arduino which from 20 to 255. The speed membership function is listed in table.

Table 3.4: Membership function for PWM of motor

Name	Range of membership function	Figure
Speed	Low: 20 to 114 Medium: 43.5 to 231.5 Fast: 161 to 255	

3.6.3 Fuzzy Rules

There are total 12 rules is set based on 12 conditions. IF and THEN statements are used but the because of two inputs for fuzzy logic so an AND statement is used when happened of two input conditions.

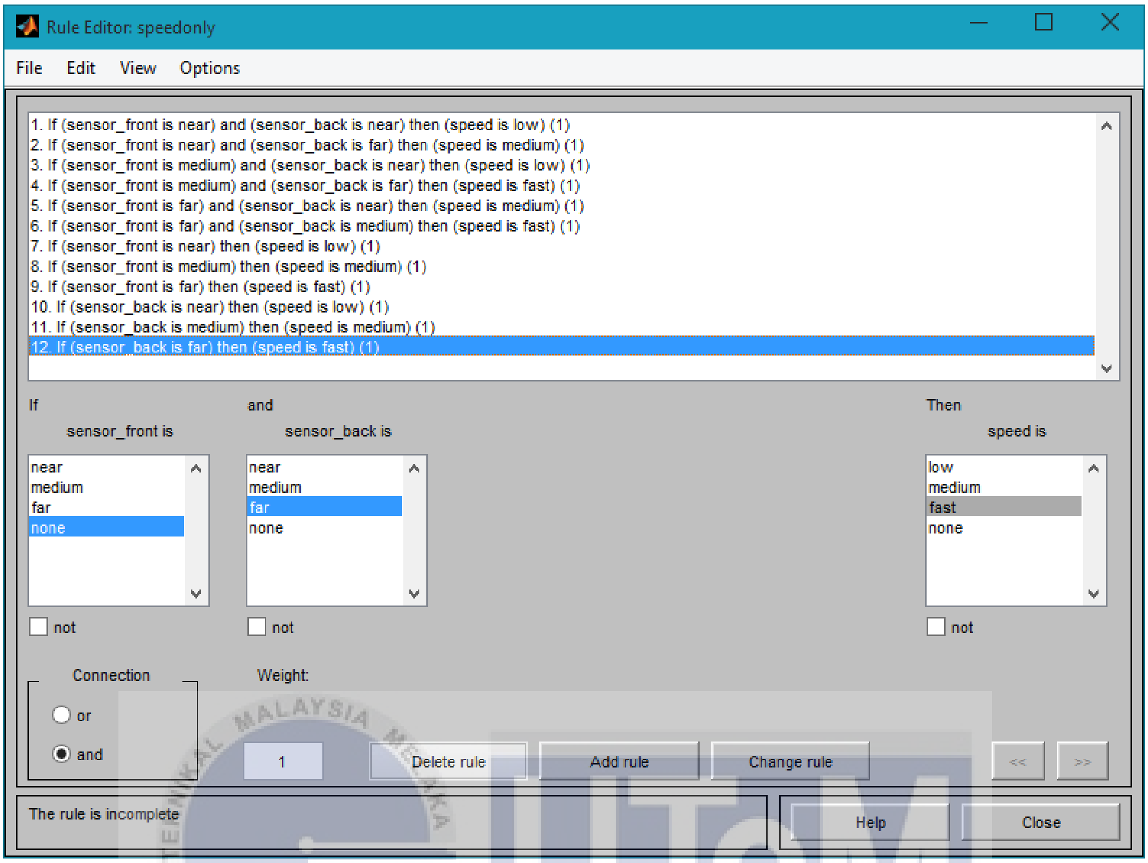


Figure 3.13: Rules set in Matlab Fuzzy Inference System

Fuzzy logic is used in speed control of side by side person following robot. There are 12 rules are set:

1. If (sensor_front is near) and (sensor_back is medium) then (speed is low)
2. If (sensor_front is near) and (sensor_back is far) then (speed is medium)
3. If (sensor_front is medium) and (sensor_back is near) then (speed is low)
4. If (sensor_front is medium) and (sensor_back is far) then (speed is fast)
5. If (sensor_front is far) and (sensor_back is near) then (speed is medium)
6. If (sensor_front is far) and (sensor_back is far) then (speed is fast)
7. If (sensor_front is near) then (speed is low)
8. If (sensor_front is medium) then (speed is medium)
9. If (sensor_front is far) then (speed is fast)
10. If (sensor_back is near) (speed is low)
11. If (sensor_back is medium) (speed is medium)
12. If (sensor_back is far) (speed is fast)

3.7 Overview of side by side person following robot function

3.7.1 Flowchart of program of side by side person following robot

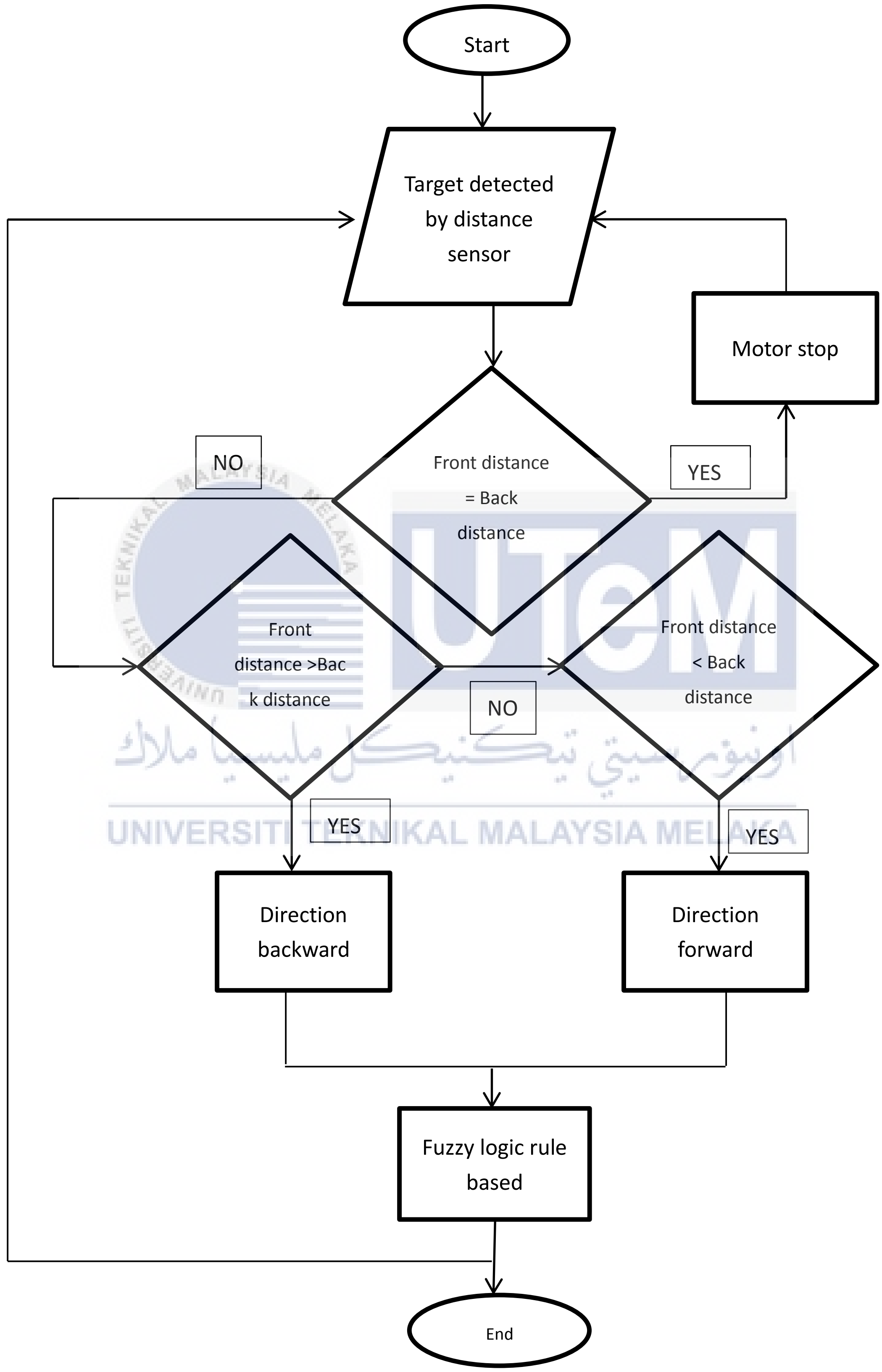


Figure 3.14: Flowchart of side by side person following robot distance control system

Figure 3.14 is the flowchart of side by side person following robot speed control and speed is controlled by fuzzy logic rule based control algorithm. At the beginning, the distance sensor will detect the whether the target is moving. The robot will be stopped when target human is at middle of ultrasonic sensor. If the distance detected by front sensor has shorter distance than distance measured by back sensor then robot will move forward. In addition, the robot will move backward when distance measured by back sensor has shorter distance than distance measured by front sensor. Once the direction robot moving is fixed and distance measured will through the fuzzy logic first before entering next stage. Moreover, the distance will be classified into far, medium and near. When distance, D is more than 300cm the motor will moving fast while when distance is less than 300cm but more than 100cm then the robot will moving with medium speed. If the distance detected is less than 100cm then the motor will started with low speed. The process will be repeated until the OFF button is pressed.

3.7.2 Block Diagram

Figure 3.15: Block diagram of side by side person following robot

In Figure 3.19 indicates open loop block diagram for side by side person following robot. Fuzzy logic will as the controller for whole system and motor as the actuator of robot. Besides, the ultrasonic sensor is to measure the distance while the input is distance and output is speed.

3.8 Project Setup

In this project setup will described the data collected through experiment. The purpose of experiment 1 is to control the rpm of motor by controlling the PWM. Besides, the experiment 2 is to analysis the performance side by side person following robot.

3.8.1 Experiment 1

In this experiment 1 is to control the speed of motor and then collect the data produced by speed output of GFS4G100 DC motor. The purpose of this experiment is to determine rpm of motor by adjusted the input PWM. By controlling motor speed, the both motors of side by side person following robot is also can be control independently, so the robot is able to increase and decrease the speed to follow the target human.

The components used are GFS4G100 DC motor, Arduino microcontroller and a 24V AC adapter. In experiment, the motor will be tested in no load condition. Besides, from the motor data sheet, it shows that the motor rpm is from 100 rpm to 3000 rpm for the motor moving without load. First of all, Arduino are used to control the input voltage for motor by adjusting the PWM. The PWM will be adjusted from 0 to 255 with increment of 1. The duty cycle is increasing when PWM is increasing. After that, the speed output signal from motor will send to Arduino and show in serial monitor of Arduino. The data collected is the average rpm from 10 readings when PWM is increase by 1, all the average rpm values is taken six times to obtain overall average rpm values. Besides, the data is included the motor rotation in clockwise and counterclockwise and after that the data will be recorded in Appendix C. The most important things need to take note is speed output signal is need to pull up the signal to 5V.

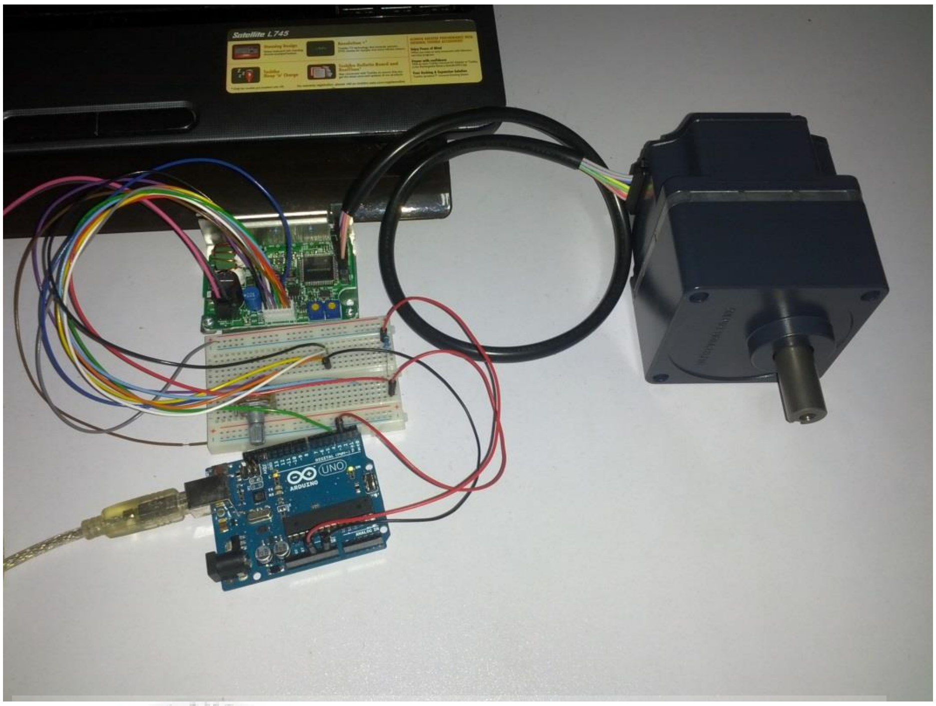


Figure 3.16: The circuit of brushless dc motor with motor driver and Arduino

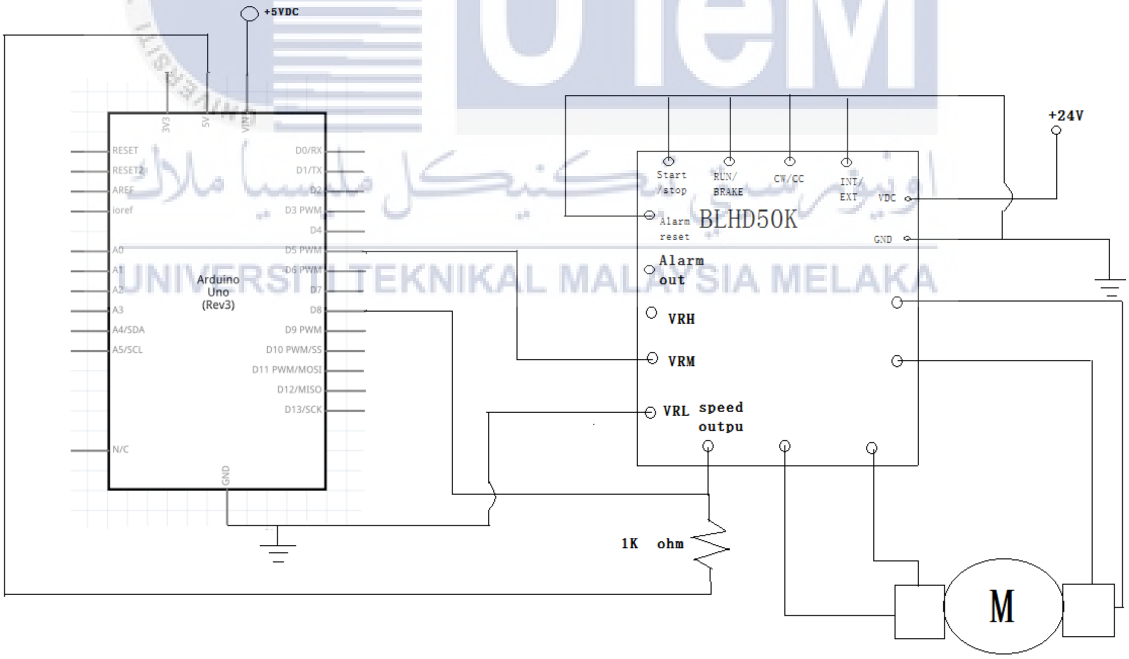


Figure 3.17: Schematic circuit of brushless dc motor

3.8.2 Experiment 2

In the experiment 2 is to measure the accuracy of ultrasonic sensor and examine the speed of person following robot when human walking. This experiment is related to third objective the performance of side by side person following robot in term of distance and speed control. The speed control is depends on the distance between robot and human. In order to control the speed of robot, the accuracy of distance sensor is necessary to build a better side by side person following robot.

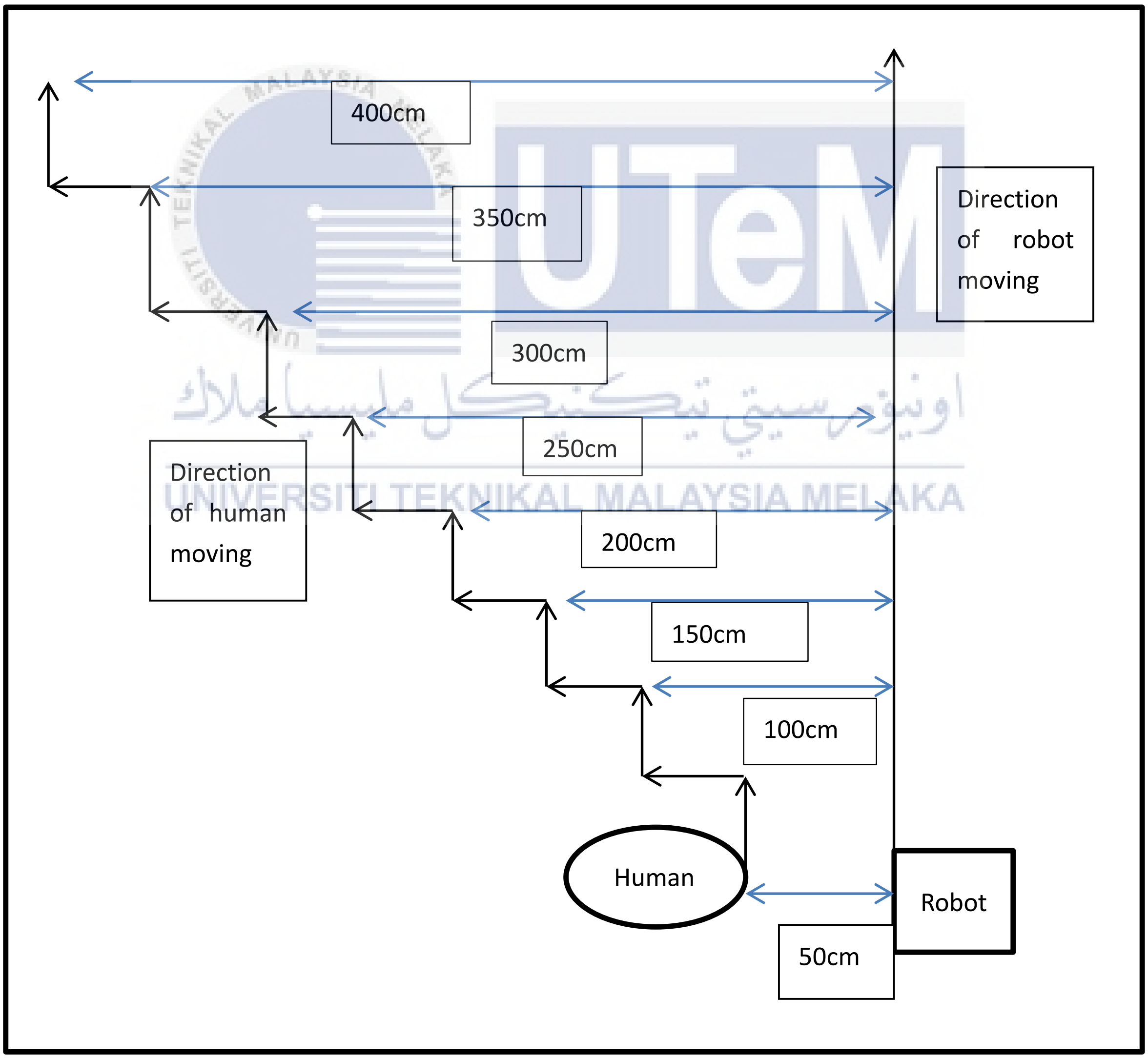


Figure 3.18: Experiment 2 setup

Figure 3.19:Location of hardware in a room

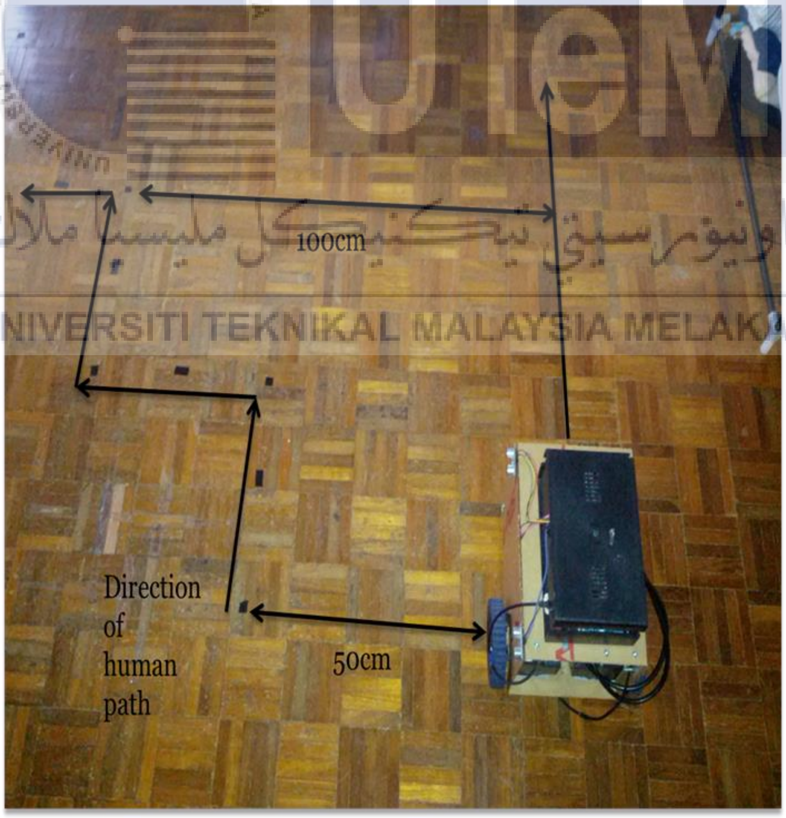


Figure 3.20: Experiment 2 setup and hardware tested in a room

First of all, human and robot will stand side by side with a distance of 50cm. After that, human will start to walk forward a distance and then stop and move away from robot until distance between human and robot is 100cm. Moreover, human will repeat the step of move forward and then more away from robot. The distance between human and robot will increase with the increment of 50cm until human is 400cm distance from robot. The robot is set to get data of distance between target and speed output of robot.

CHAPTER 4

RESULT AND ANALYSIS

4.1 Overview

In the section is described about the data collected from the experiment. Besides, the table of data will be put in Appendix and the graphs are plotted according the data. The graph will be analysed and explained in this section.

4.2 Experiment 1

In this experiment is to control rpm of motor. The PWM is adjusted slowly to determine the changes of speed output from motor. The graphs rpm versus duty cycle is plotted as below.

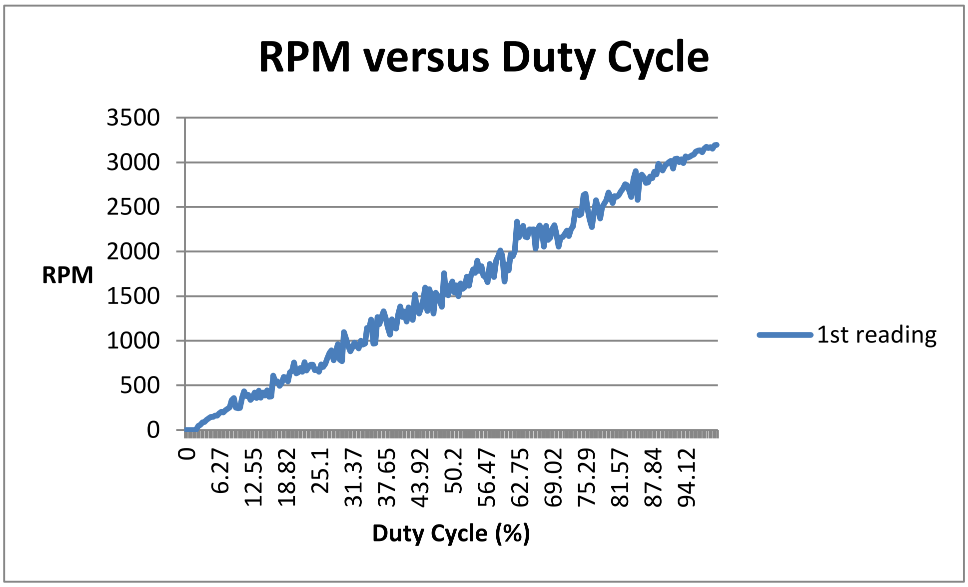


Figure 4.1: First reading of RPM versus Duty cycle

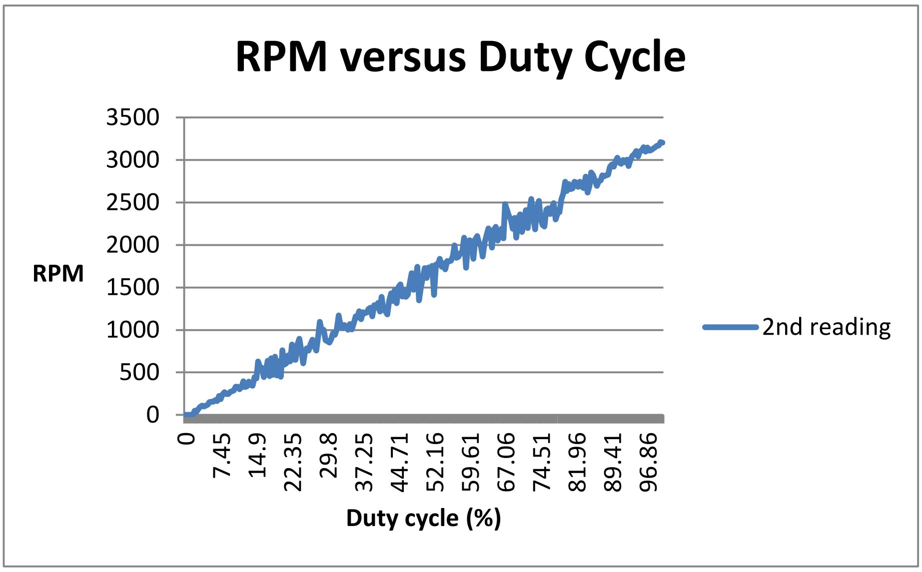


Figure 4.2: Second reading of RPM versus Duty cycle

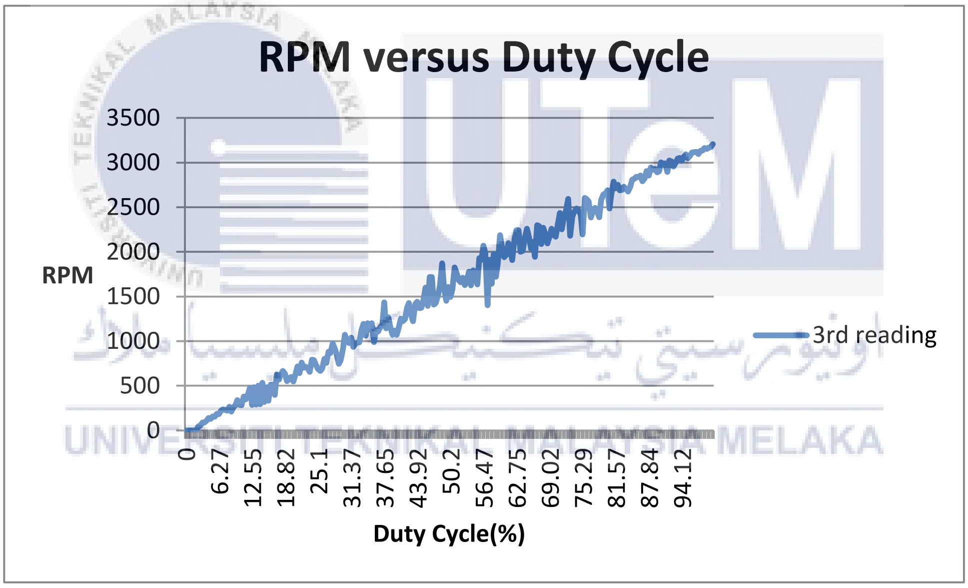


Figure 4.3: Third reading of RPM versus Duty cycle

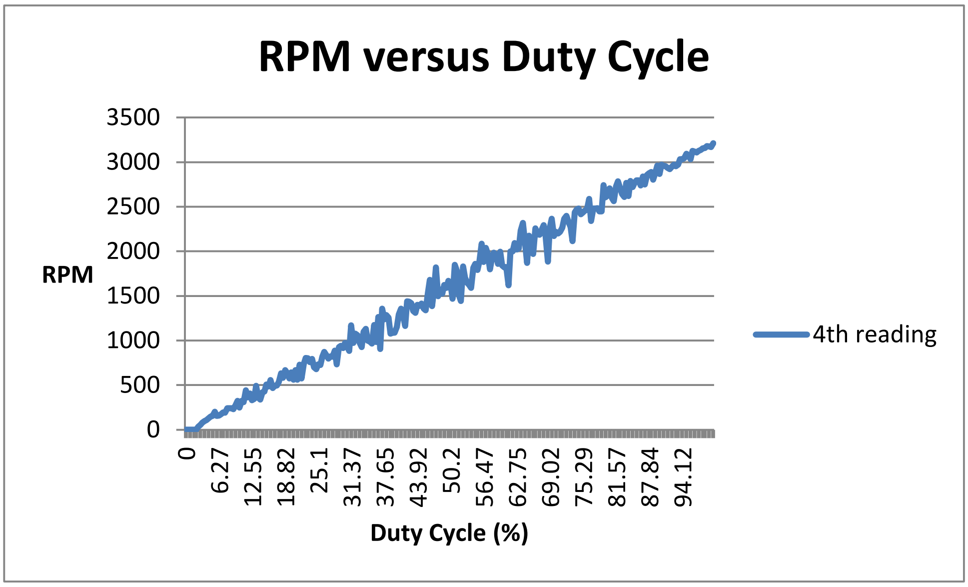


Figure 4.4: Fourth reading of RPM versus Duty cycle

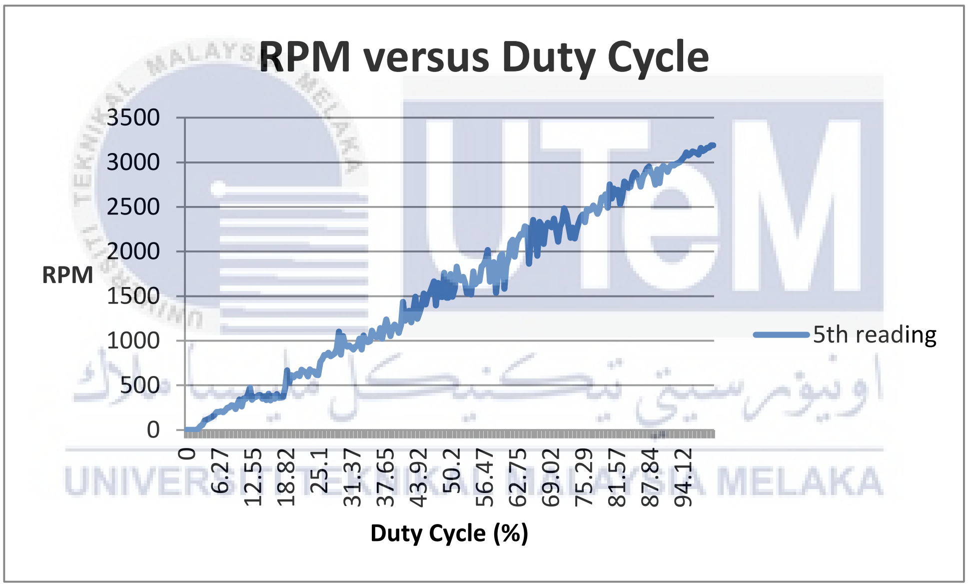


Figure 4.5: Fifth reading of RPM versus Duty cycle

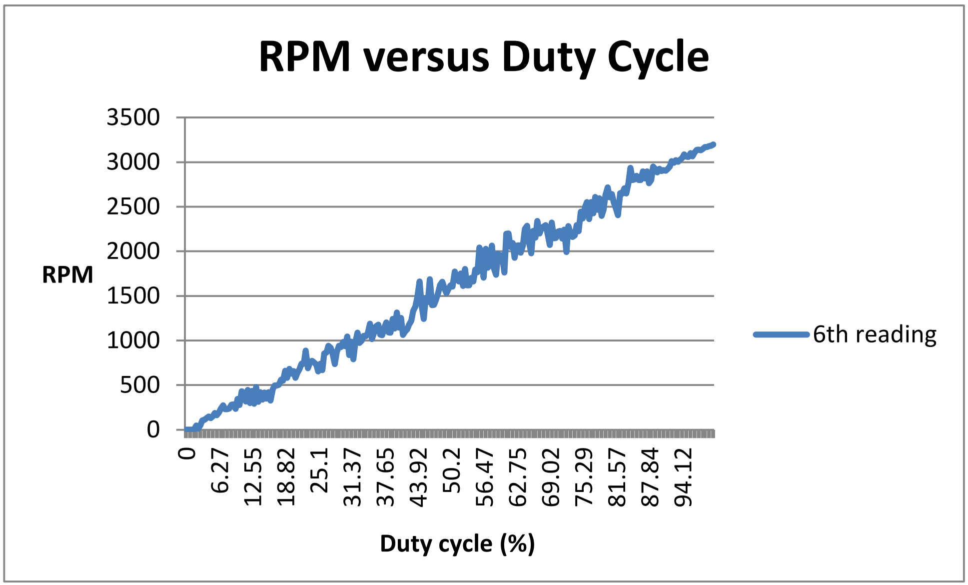


Figure 4.6: Sixth reading of RPM versus Duty cycle

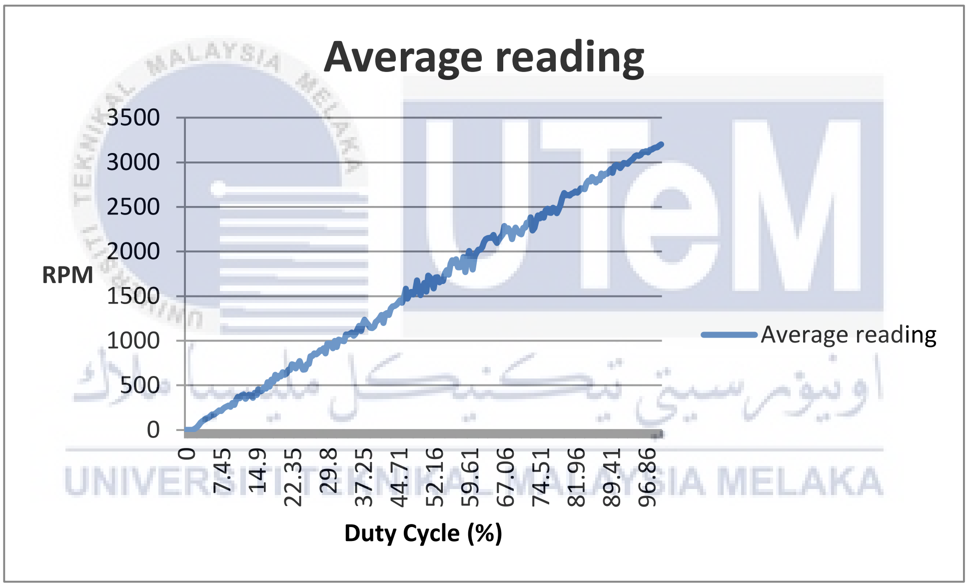


Figure 4.7: Average reading of RPM versus Duty Cycle

4.2.1 Analysis of Experiment 1

From the Figure 4.1 until Figure 4.6 are representing the data recorded from the motor speed output signal in PWM. The graphs show when the duty cycle is increasing, the rpm of motor is also increasing but there are not increasing linearly compare to theoretical result. The average data from figure 4.1 to figure 4.6 is calculated and presented on Figure 4.7.

From the Figure 4.7, when the duty cycle is around 1.96% the revolution per minute of motor is started increasing from 0. It is due to the when duty cycle is 0% means the input voltage of motor is 0V. When the duty cycle is within the 0% and the 1.96%, the input voltage is not enough to move the rotor of motor, so the rpm is still 0. The rpm of motor is started drop at the duty cycle at 5.88%. While after the rpm drop and the next reading of rpm is increase again. It shows not all rpm is decrease when the increasing of duty cycle. The situation of increase and decrease of rpm is continuing until the duty cycle is 97.65%. It is due to the low frequency PWM of Arduino and not high enough to let the rotor turn smoothly lead motor running with rattle. After 97.65% of duty cycle, the rpm is started increasing linearly this can explain that the low frequency of PWM is ignored when the duty cycle is almost full, thus the motor can move smoothly. Lastly, when the duty cycle is 100% the average maximum rpm is 3200.16.

In addition, the standard deviation of the data is calculated. The highest standard deviation is 219.3725 when the duty cycle is at 57.25%. It means that the largest different of rpm readings at duty cycle 57.25%. Although revolution per minute is not increasing smoothly, but the rpm still can controlled by PWM.

4.3 Experiment 2

In experiment 2 is testing the distance control and speed control. Target human will walk from 50cm away from robot until 400cm from robot. Furthermore, the speed of robot is set as the distance between target and robot is increasing, the speed is also increasing.

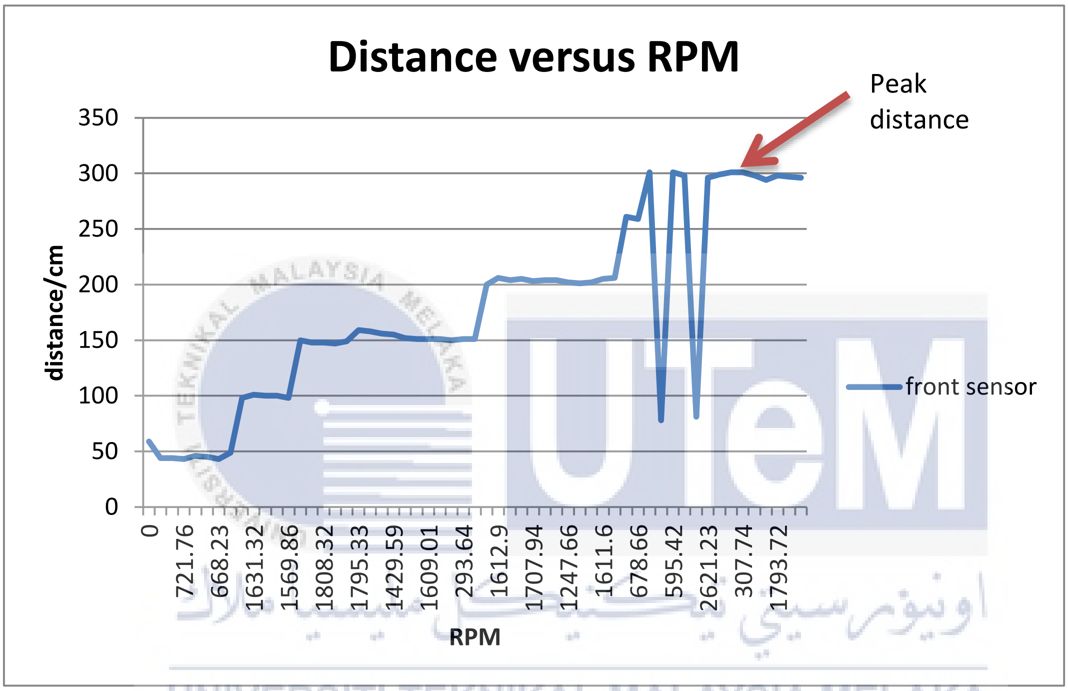


Figure 4.8: Distance from front sensor versus RPM

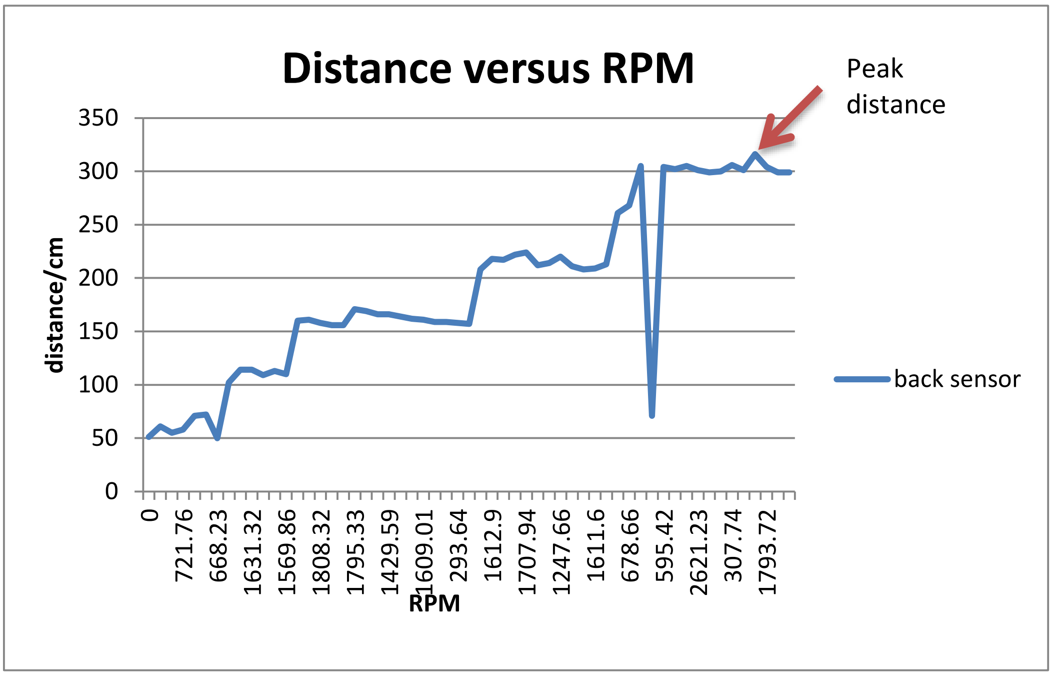


Figure 4.9: Distance from back sensor versus RPM

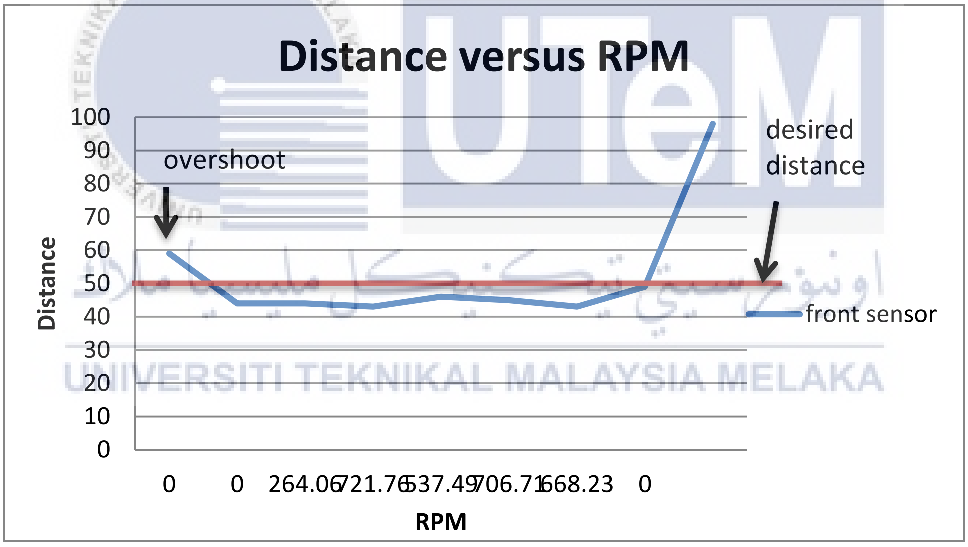


Figure 4.10: Overshoot in graph of distance versus RPM

Table 4.1: Percentage error of distance for front sensor

Case	Actual distance(cm)	Desired Distance (cm)	Percentage Error (%)	RPM
Case 1: Human is 50cm away from robot	59	50	18	0
	44	50	12	0

	44	50	12	264.06
	43	50	14	721.76
	46	50	8	537.49
	45	50	10	706.71
	43	50	14	668.23
	49	50	2	0
Case 2: Human is 200cm away from robot.	200	200	0	0
	206	200	3	1612.9
	204	200	2	1661.13
	205	200	2.5	1571.09
	203	200	1.5	1707.94
	204	200	2	1684.92
	204	200	2	1684.92
	202	200	1	1247.66
	201	200	0.5	1919.39
	202	200	1	1527.88
	205	200	2.5	1611.6
	206	200	3	0
Case 3: Human is 300cm away from robot.	301	300	0.33333	684.7
	78	300	74	2472.19
	301	300	0.33333	595.42
	298	300	0.66667	1813.24
	81	300	73	2195.39

	296	300	1.33333	2621.23
	299	300	0.33333	1088.14
	301	300	0.33333	1833.18
	301	300	0.33333	307.74
	298	300	0.66667	2631.58
	294	300	2	1690.62
	298	300	0.66667	1793.72
	297	300	1	772.2
	296	300	1.33333	2030.46

Table 4.2: Percentage error of distance for back sensor

Case	Actual distance(cm)	Desired Distance (cm)	Percentage Error (%)	RPM
Case 1: Human is 50cm away from robot	51	50	2	0
	61	50	22	0
	55	50	10	264.06
	58	50	16	721.76
	71	50	42	537.49
	72	50	44	706.71
	50	50	0	668.23
	102	50	104	0
Case 2: Human is 200cm away from robot.	208	200	4	0
	218	200	9	1612.9
	217	200	8.5	1661.13
	222	200	11	1571.09
	224	200	12	1707.94
	212	200	6	1684.92
	214	200	7	1684.92
	220	200	10	1247.66

	211	200	5.5	1919.39
	208	200	4	1527.88
	209	200	4.5	1611.6
	213	200	6.5	0
Case 3: Human is 300cm away from robot.	305	300	1.66667	684.7
	71	300	76.33333	2472.19
	304	300	1.33333	595.42
	302	300	0.66667	1813.24
	305	300	1.66667	2195.39
	301	300	0.33333	2621.23
	299	300	0.333333	1088.14
	300	300	0	1833.18
	306	300	2	307.74
	301	300	0.33333	2631.58
	316	300	5.33333	1690.62
	304	300	1.33333	1793.72
	299	300	0.333333	772.2
	299	300	0.333333	2030.46

4.3.1 Analysis of Experiment 2

First of all, from beginning Figure 4.8 and 4.9 indicate the distance between target and robot is started from around 50cm slowly increase with increment of 50cm until the target having distance around 300cm away from side by side person following robot. In Table 4.11 and 4.12, the zero rpm is means that the robot is stopped. Besides, the lowest distance measured by front sensor 43cm in Case 1 of the Table 4.11 with 14% of error. For the lowest distance measured by back sensor is 50cm in Case 1 in Table 4.12 with 0% of error. Furthermore, the target is started moving forward and the motor is started moving with 264.06 rpm in Case 1 of both Table 4.11 and 4.12. Next overshoot is occurred in Figure 4.10 shows that during the starting of experiment.

Essentially the ultrasonic sensor has the maximum 400cm of distance measurement and the experiment is set for target human moving away from robot until 400cm to test the accuracy of ultrasonic sensors, but the Figure 4.8 and 4.9 peak distance detected by front sensor is 301cm while peak distance showed by back sensor is 316cm. The side by side robot is actually stopped moving, when target human is moving to 350cm away from robot. The maximum sensing distance is affected by material, for example the jeans wear by target is a material absorbs sound wave, so it limits the maximum sensing distance.

In addition, the maximum percentage error obtained from front sensor is 74% which is shown in Case 3 of Table 4.11 and the distance measured is 78cm. At the same time, maximum percentage error from back sensor is 76.3333% in Table 4.12 Case 3 also and it is higher than percentage error of front sensor. These errors may be due to disturbance from nearby environment.

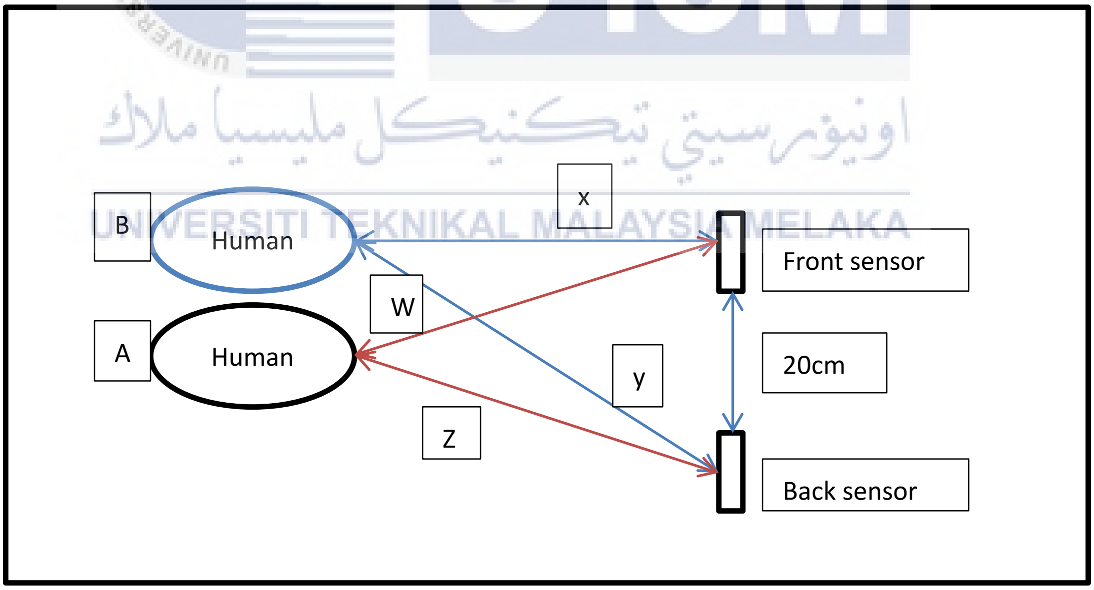


Figure 4.11: Target moving location

From figure 4.8 and 4.9, there are slightly difference distance detected by front sensor and back sensor. The distance detected by back sensor is slightly higher than front sensor. In figure 4.13 explain how the back sensor is slightly higher than front sensor. At point A, target human is stop and target is at the middle of between front sensor and back sensor. W is the distance detected by front sensor and Z is the distance detected by back sensor. When W and Z is almost same distance, the robot will stop. Moreover, when target is moving forward as point B, the front sensor will detect target with distance X and back sensor will detect target with distance Y. Distance Y is bigger than X so that why the back sensor is slightly higher than front sensor.

Moreover, the highest rpm of motor in Table 4.11 and Table 4.12 is 2631.58 rpm when the target has a distance 298cm with 0.666667% error detected by front sensor and 301cm measured by back sensor with error 0.3333%. However, there is a problem not showed in result which is the real speed of motor. The speed output, rpm directly produced by motor is excluded the calculation of gear ratio. Therefore, the actual speed of motor produced is lower than expected result for example taking the maximum rpm from figure 4.8 and 4.9, the wheel radius is 6.35cm.

$$\frac{2631.58}{100} \times 2\pi(6.35cm) \div 60 = 17.5cm/s$$

The side by side person following robot is moving with 17.5cm/s which is maximum speed when target human is around 300cm away from robot.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

The prototype of side by side person following robot is designed with two ultrasonic sensors installed on the left side of robot. Besides, two motors with motor drivers are used as the actuator of robot. In order to control speed of motor fuzzy logic controller is implemented into side by side person following robot where the input of fuzzy logic is distance and output is speed. 12 fuzzy rules are set based on 12 condition may happened. In experiment 1, the highest standard deviation is 219.3725 when the duty cycle is at 57.25% although rpm is increasing with fluctuation, but the rpm still can control by PWM. From experiment 2, the maximum distance detection by the front sensor is only 301cm while the maximum distance detection by back sensor is 316cm. Moreover, the maximum error distance error by front sensor and back sensor are 74% and 76.3333%. These large errors are occurred when human is 300cm away from robot where the distance measured by front sensor and back sensor are 78cm and 71cm. Furthermore, overshoot is occurred in fuzzy logic controller during the second experiment.

5.2 Future Work

For the future research, the side by side person following robot need to improve the type of motor used. Although the motor used has a larger torque but the gear ratio of motor has lower the speed of motor and lead the robot not able to follow human maximum speed.

Therefore, the gear ratio of motor should be lower. Lastly, overshoot result can be improved by change the range of membership function.

REFERENCES:

1. David Feil-Seifer and Maja J Mataric, Defining Socially Assistive Robotics, IEEE 9th International c0nference on rehabilitation robotics, July 2005.
2. The World Bank Group (2014), *Population ages 65 and above (% of total)*, Available: http://data.worldbank.org/indicator/SP.POP.65UP.TO.ZS/countries/JP?display=graph
3. Reno J. Tibke, "Japan's Taking Robot Action: Honda, Sekisui House, and New Government Funding", Copyright © 2001 – 2013.
4. Tapus, A., & Mataric, M. (n.d.). Towards Socially Assistive Robotics. Journal of the Robotics Society of Japan, 576-578.
5. Yoichi Morales, Satoru Satake, Rajibul Huq, Dylan Glas, Takayuki Kanda & Norihiro Hagita, How Do People Walk Side By Side?-Using A Computational Model Of Human Behavior For A Social Robot, USA, 2012.
6. Songmin Jia, Lijia Wang, Shuang Wang & Congxuan Bai, Fuzzy-based intelligent control strategy for a person following robot, IEEE International Conference on Robotics and Biomimetics (ROBIO), 2013.
7. Tarik Bin Alamgir, Ibrahim Jawad Alfar & Muhammad Mahbubur Rashid, Modeling and robust analysis of a fuzzy based person following robot, IEEE International Conference on Computer and Communication Engineering, July 2012.
8. Prasarn Kiddee & Akira Shimada, A controller design on person following Omni-directional Vehicle Robots, IEEE SICE annual conference, 2007.
9. Jianzhao Cai & Takafumi Matsumaru, Robot human-following limited speed control, IEEE International Symposium on robot and human interactive communication, 2013.
10. TranSafety, Inc., Road Engineering Journal, October, 1997
11. *Brushless Motor and Driver Package BLH Series Operating Manual*, Oriental Motor Co., Ltd, 2011

APPENDIX A

Grantt chart

FYP 1

Week	1	2	3	4	5	6	7	8	9	10	11
Briefing											
Registration											
Grantt Chart											
Literature Review											
Motivation											
Objective											
Scope											
Methodology											
Preliminary Result											
Conclusion											
Submit Report											
Presentation											

FYP 2

Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Prototype Development																
Locate Position of Ultrasonic Sensor																
Design Control Algorithm																
Programming																
Experiment																
Analysis & Discussion																
Submit Draft Report																
Slides Preparation																
Presentation																

APPENDIX B

Programming for experiment of motor speed control

```
int pin = 2;
unsigned long duration;
float totaltime;
float rpm;
float averagerpm;
void setup()
{
 pinMode(pin, INPUT_PULLUP);    // change “pin” to electrical behavior
 Serial.begin(9600);
 Serial.println("Started");            // print “Started”
}

void loop()
{
  for (int i=0;i<256;i++)                              // PWM increasing with increment 1
  {

     averagerpm = 0;
     analogWrite(3,i);
     for (int times = 0;times<10;times++)              // get the readings 10 times
     {
        duration = pulseIn(pin, HIGH);                 //timing when HIGH state
        duration = (duration+300);                     //timing plus pulse width
        totaltime= (duration/1000000.000000);
        rpm= (2.000000/totaltime);
```

```
            if (rpm >=4000){
              rpm = 0;


            }
            averagerpm = averagerpm + rpm;              // total rpm
        }
        averagerpm = averagerpm / 10;                   // average rpm


      Serial.print(i/2.55);                             // print duty cycle
      Serial.print("\t");
      Serial.println(averagerpm);                       // print average rpm
      delay(100);
    }
    Serial.println("Finished!");
}
```

APPENDIX C

```
//************************************************************************
// Matlab .fis to arduino C converter v2.0.0.29032014
// - Karthik Nadig, USA
// Please report bugs to: karthiknadig@gmail.com
//************************************************************************

#include "fis_header.h"
#define echoPin 7 // Echo Pin
#define trigPin 8 // Trigger Pin
#define echoPin1 24
#define trigPin1 26
long duration, distan,distance; // Duration used to calculate distance
long durat,distan1,distance1;
int pin = 28;
unsigned long time;
float totaltime;
float rpm;
// Number of inputs to the fuzzy inference system
const int fis_gcI = 2;
// Number of outputs to the fuzzy inference system
const int fis_gcO = 1;
// Number of rules to the fuzzy inference system
const int fis_gcR = 12;

FIS_TYPE g_fisInput[fis_gcI];
FIS_TYPE g_fisOutput[fis_gcO];
```

```
// Setup routine runs once when you press reset:
void setup()
{
     pinMode(pin, INPUT_PULLUP);    // change "pin" to electrical behavior
     Serial.begin (9600);
     pinMode(trigPin, OUTPUT);
     pinMode(echoPin, INPUT);
     pinMode(trigPin1, OUTPUT);
     pinMode(echoPin1, INPUT);
    // initialize the Analog pins for output.
    // Pin mode for Output: speed
    pinMode(2 , OUTPUT);
    // Pin mode for Output: LM
    pinMode(3, OUTPUT);
    // Pin mode for Output: RM
    pinMode(4, OUTPUT);
    // Pin mode for Output: LM_direction
    pinMode(5, OUTPUT);
    // Pin mode for Output: RM_Direction
    pinMode(6, OUTPUT);

}

// Loop routine runs over and over again forever:
void loop()
{
  distance=0;
  distance1=0;
   for (int times = 0;times<10;times++)
    {   digitalWrite(trigPin, LOW);
```

```
    delayMicroseconds(2);

    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);

    digitalWrite(trigPin, LOW);
    duration = pulseIn(echoPin, HIGH);

    //Calculate the distance (in cm) based on the speed of sound.
    distan = duration/58.2;

    distance = distance + distan;
}
distance = distance / 10;

for (int times = 0;times<10;times++)
{   digitalWrite(trigPin1, LOW);
    delayMicroseconds(2);

    digitalWrite(trigPin1, HIGH);
    delayMicroseconds(10);

    digitalWrite(trigPin1, LOW);
    durat = pulseIn(echoPin1, HIGH);

    //Calculate the distance (in cm) based on the speed of sound.
    distan1 = durat/58.2;

    distance1 = distance1 + distan1;
}
```

```
  distance1 = distance1 / 10;


  // Read Input: sensor_front
  g_fisInput[0] = distance1;
  // Read Input: sensor_back
  g_fisInput[1] = distance;

  g_fisOutput[0] = 0;


  fis_evaluate();

  // Set output vlaue: speed
  analogWrite(2 , g_fisOutput[0]);


if(distance1<=100 && distance<=100)
{
    Serial.println(distance1);
    Serial.print("\t");
    Serial.println(rpm);                    // print average rpm
    Serial.println(distance);
    Serial.print("\t");
    Serial.println(rpm);                    // print average rpm
  if(abs(distance-distance1)<=8)
  {
     digitalWrite(3,HIGH);
     digitalWrite(4,HIGH);
```

```
    }
    else if(distance1<distance)
    {
    digitalWrite(3,LOW);
    digitalWrite(4,LOW);
    digitalWrite(5,LOW);//Left
    digitalWrite(6,HIGH);//Right
    }
    else if(distance1>distance)
    {
    digitalWrite(3,LOW);
    digitalWrite(4,LOW);
    digitalWrite(5,HIGH);//Left backward
    digitalWrite(6,LOW);//Right backward
    }
}
  else if(distance1<=300 && distance<=300 && distance1>=101 && distance>=101)
{
    Serial.println(distance1);
      Serial.print("\t");
      Serial.println(rpm);                    // print average rpm
      Serial.println(distance);
      Serial.print("\t");
      Serial.println(rpm);                    // print average rpm
if(abs(distance-distance1)<=8)
    {
        digitalWrite(3,HIGH);
        digitalWrite(4,HIGH);
    }
```

```
    else if(distance1<distance)
    {
    digitalWrite(3,LOW);
    digitalWrite(4,LOW);
    digitalWrite(5,LOW);//Left
    digitalWrite(6,HIGH);//Right
    }
    else if(distance1>distance)
    {
    digitalWrite(3,LOW);
    digitalWrite(4,LOW);
    digitalWrite(5,HIGH);//Left backward
    digitalWrite(6,LOW);//Right backward
    }
}
else if(distance1<=400 && distance<=400 && distance1>=301 && distance>=301)
{
    Serial.println(distance1);
     Serial.print("\t");
     Serial.println(rpm);                          // print average rpm
     Serial.println(distance);
     Serial.print("\t");
     Serial.println(rpm);                          // print average rpm
    if(abs(distance-distance1)<=8)
    {
       digitalWrite(3,HIGH);
       digitalWrite(4,HIGH);
    }
    else if(distance1<distance)
    {
```

```
  digitalWrite(3,LOW);
  digitalWrite(4,LOW);
  digitalWrite(5,LOW);//Left
  digitalWrite(6,HIGH);//Right
  }
  else if(distance1>distance)
  {
  digitalWrite(3,LOW);
  digitalWrite(4,LOW);
  digitalWrite(5,HIGH);//Left backward
  digitalWrite(6,LOW);//Right backward
  }
}
else if(distance1<=100 && distance>100 && distance<=400 && distance1>=30)
{
  Serial.println(distance1);
   Serial.print("\t");
   Serial.println(rpm);                    // print average rpm
   Serial.println(distance);
   Serial.print("\t");
   Serial.println(rpm);                    // print average rpm
 // if(abs(distance-distance1)<=70)
 // {
  //    digitalWrite(3,HIGH);
 //     digitalWrite(4,HIGH);
 // }
  //else
  //{
  digitalWrite(3,LOW);
  digitalWrite(4,LOW);
```

```
   digitalWrite(5,LOW);//Left
   digitalWrite(6,HIGH);//Right
   //}
}
else if(distance1<=300 && distance>300    && distance<=400 && distance1>=30)
{
     Serial.println(distance1);
     Serial.print("\t");
     Serial.println(rpm);                    // print average rpm
     Serial.println(distance);
     Serial.print("\t");
     Serial.println(rpm);                    // print average rpm
   digitalWrite(3,LOW);
   digitalWrite(4,LOW);
   digitalWrite(5,LOW);//Left
   digitalWrite(6,HIGH);//Right

}
else if(distance1>100 && distance<=100    && distance1<=400 && distance>=30)
{
     Serial.println(distance1);
     Serial.print("\t");
     Serial.println(rpm);                    // print average rpm
     Serial.println(distance);
     Serial.print("\t");
     Serial.println(rpm);                    // print average rpm
   digitalWrite(3,LOW);
   digitalWrite(4,LOW);
   digitalWrite(5,HIGH);//Left
   digitalWrite(6,LOW);//Right
```

```
    }
    else if(distance1>300 && distance<=300 && distance1<=400 && distance>=30)
    {
        Serial.println(distance1);
        Serial.print("\t");
        Serial.println(rpm);                // print average rpm
        Serial.println(distance);
        Serial.print("\t");
        Serial.println(rpm);                // print average rpm
      digitalWrite(3,LOW);
      digitalWrite(4,LOW);
      digitalWrite(5,HIGH);//Left
      digitalWrite(6,LOW);//Right
    }
  else if (distance1>400 && distance >400)
  {

    digitalWrite(3,HIGH);
    digitalWrite(4,HIGH);

  }

}
//***********************************************************************
// Support functions for Fuzzy Inference System
//***********************************************************************
// Triangular Member Function
FIS_TYPE fis_trimf(FIS_TYPE x, FIS_TYPE* p)
{
```

```
    FIS_TYPE a = p[0], b = p[1], c = p[2];
    FIS_TYPE t1 = (x - a) / (b - a);
    FIS_TYPE t2 = (c - x) / (c - b);
    if ((a == b) && (b == c)) return (FIS_TYPE) (x == a);
    if (a == b) return (FIS_TYPE) (t2*(b <= x)*(x <= c));
    if (b == c) return (FIS_TYPE) (t1*(a <= x)*(x <= b));
    t1 = min(t1, t2);
    return (FIS_TYPE) max(t1, 0);
}

FIS_TYPE fis_min(FIS_TYPE a, FIS_TYPE b)
{
    return min(a, b);
}

FIS_TYPE fis_max(FIS_TYPE a, FIS_TYPE b)
{
    return max(a, b);
}

FIS_TYPE fis_array_operation(FIS_TYPE *array, int size, _FIS_ARR_OP pfnOp)
{
    int i;
    FIS_TYPE ret = 0;

    if (size == 0) return ret;
    if (size == 1) return array[0];

    ret = array[0];
    for (i = 1; i < size; i++)
```

```
    {
        ret = (*pfnOp)(ret, array[i]);
    }

    return ret;
}


//***********************************************************************
// Data for Fuzzy Inference System
//***********************************************************************
// Pointers to the implementations of member functions
_FIS_MF fis_gMF[] =
{
    fis_trimf
};

// Count of member function for each Input
int fis_gIMFCount[] = { 3, 3 };

// Count of member function for each Output
int fis_gOMFCount[] = { 3 };

// Coefficients for the Input Member Functions
FIS_TYPE fis_gMFI0Coeff1[] = { -153, 5, 30 };
FIS_TYPE fis_gMFI0Coeff2[] = { 30, 40, 50 };
FIS_TYPE fis_gMFI0Coeff3[] = { 50, 400, 411 };
FIS_TYPE* fis_gMFI0Coeff[] = { fis_gMFI0Coeff1, fis_gMFI0Coeff2,
fis_gMFI0Coeff3 };
FIS_TYPE fis_gMFI1Coeff1[] = { -153, 5, 30 };
```

```
FIS_TYPE fis_gMFI1Coeff2[] = { 30, 40, 50 };
FIS_TYPE fis_gMFI1Coeff3[] = { 50, 400, 558 };
FIS_TYPE* fis_gMFI1Coeff[] = { fis_gMFI1Coeff1, fis_gMFI1Coeff2, fis_gMFI1Coeff3 };
FIS_TYPE** fis_gMFICoeff[] = { fis_gMFI0Coeff, fis_gMFI1Coeff };

// Coefficients for the Input Member Functions
FIS_TYPE fis_gMFO0Coeff1[] = { -74, 20, 114 };
FIS_TYPE fis_gMFO0Coeff2[] = { 43.5, 137.5, 231.5 };
FIS_TYPE fis_gMFO0Coeff3[] = { 161, 255, 349 };
FIS_TYPE* fis_gMFO0Coeff[] = { fis_gMFO0Coeff1, fis_gMFO0Coeff2, fis_gMFO0Coeff3 };
FIS_TYPE** fis_gMFOCoeff[] = { fis_gMFO0Coeff };

// Input membership function set
int fis_gMFI0[] = { 0, 0, 0 };
int fis_gMFI1[] = { 0, 0, 0 };
int* fis_gMFI[] = { fis_gMFI0, fis_gMFI1};

// Output membership function set
int fis_gMFO0[] = { 0, 0, 0 };
int* fis_gMFO[] = { fis_gMFO0};

// Rule Weights
FIS_TYPE fis_gRWeight[] = { 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 };

// Rule Type
int fis_gRType[] = { 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 };

// Rule Inputs
```

```
int fis_gRI0[] = { 1, 1 };
int fis_gRI1[] = { 1, 3 };
int fis_gRI2[] = { 2, 1 };
int fis_gRI3[] = { 2, 3 };
int fis_gRI4[] = { 3, 1 };
int fis_gRI5[] = { 3, 2 };
int fis_gRI6[] = { 1, 0 };
int fis_gRI7[] = { 2, 0 };
int fis_gRI8[] = { 3, 0 };
int fis_gRI9[] = { 0, 1 };
int fis_gRI10[] = { 0, 2 };
int fis_gRI11[] = { 0, 3 };
int* fis_gRI[] = { fis_gRI0, fis_gRI1, fis_gRI2, fis_gRI3, fis_gRI4, fis_gRI5, fis_gRI6, fis_gRI7, fis_gRI8, fis_gRI9, fis_gRI10, fis_gRI11 };

// Rule Outputs
int fis_gRO0[] = { 1 };
int fis_gRO1[] = { 2 };
int fis_gRO2[] = { 1 };
int fis_gRO3[] = { 3 };
int fis_gRO4[] = { 2 };
int fis_gRO5[] = { 3 };
int fis_gRO6[] = { 1 };
int fis_gRO7[] = { 2 };
int fis_gRO8[] = { 3 };
int fis_gRO9[] = { 1 };
int fis_gRO10[] = { 2 };
int fis_gRO11[] = { 3 };
int* fis_gRO[] = { fis_gRO0, fis_gRO1, fis_gRO2, fis_gRO3, fis_gRO4, fis_gRO5, fis_gRO6, fis_gRO7, fis_gRO8, fis_gRO9, fis_gRO10, fis_gRO11 };
```

```
// Input range Min
FIS_TYPE fis_gIMin[] = { 5, 5 };

// Input range Max
FIS_TYPE fis_gIMax[] = { 400, 400 };

// Output range Min
FIS_TYPE fis_gOMin[] = { 20 };

// Output range Max
FIS_TYPE fis_gOMax[] = { 255 };

//***********************************************************************
// Data dependent support functions for Fuzzy Inference System
//***********************************************************************
FIS_TYPE fis_MF_out(FIS_TYPE** fuzzyRuleSet, FIS_TYPE x, int o)
{
    FIS_TYPE mfOut;
    int r;

    for (r = 0; r < fis_gcR; ++r)
    {
        int index = fis_gRO[r][o];
        if (index > 0)
        {
            index = index - 1;
            mfOut = (fis_gMF[fis_gMFO[o][index]])(x, fis_gMFOCoeff[o][index]);
        }
        else if (index < 0)
```

```
            {
                  index = -index - 1;
                  mfOut = 1 - (fis_gMF[fis_gMFO[o][index]])(x, fis_gMFOCoeff[o][index]);
            }
            else
            {
                  mfOut = 0;
            }

            fuzzyRuleSet[0][r] = fis_min(mfOut, fuzzyRuleSet[1][r]);
      }
      return fis_array_operation(fuzzyRuleSet[0], fis_gcR, fis_max);
}

FIS_TYPE fis_defuzz_centroid(FIS_TYPE** fuzzyRuleSet, int o)
{
      FIS_TYPE step = (fis_gOMax[o] - fis_gOMin[o]) / (FIS_RESOLUSION - 1);
      FIS_TYPE area = 0;
      FIS_TYPE momentum = 0;
      FIS_TYPE dist, slice;
      int i;

      // calculate the area under the curve formed by the MF outputs
      for (i = 0; i < FIS_RESOLUSION; ++i){
            dist = fis_gOMin[o] + (step * i);
            slice = step * fis_MF_out(fuzzyRuleSet, dist, o);
            area += slice;
            momentum += slice*dist;
      }
```

```
    return ((area == 0) ? ((fis_gOMax[o] + fis_gOMin[o]) / 2) : (momentum / area));
}

//***********************************************************************
// Fuzzy Inference System
//***********************************************************************
void fis_evaluate()
{
    FIS_TYPE fuzzyInput0[] = { 0, 0, 0 };
    FIS_TYPE fuzzyInput1[] = { 0, 0, 0 };
    FIS_TYPE* fuzzyInput[fis_gcI] = { fuzzyInput0, fuzzyInput1, };
    FIS_TYPE fuzzyOutput0[] = { 0, 0, 0 };
    FIS_TYPE* fuzzyOutput[fis_gcO] = { fuzzyOutput0, };
    FIS_TYPE fuzzyRules[fis_gcR] = { 0 };
    FIS_TYPE fuzzyFires[fis_gcR] = { 0 };
    FIS_TYPE* fuzzyRuleSet[] = { fuzzyRules, fuzzyFires };
    FIS_TYPE sW = 0;

    // Transforming input to fuzzy Input
    int i, j, r, o;
    for (i = 0; i < fis_gcI; ++i)
    {
        for (j = 0; j < fis_gIMFCount[i]; ++j)
        {
            fuzzyInput[i][j] =
                (fis_gMF[fis_gMFI[i][j]])(g_fisInput[i], fis_gMFICoeff[i][j]);
        }
    }

    int index = 0;
```

```
for (r = 0; r < fis_gcR; ++r)
{
    if (fis_gRType[r] == 1)
    {
        fuzzyFires[r] = FIS_MAX;
        for (i = 0; i < fis_gcI; ++i)
        {
            index = fis_gRI[r][i];
            if (index > 0)
                fuzzyFires[r] = fis_min(fuzzyFires[r], fuzzyInput[i][index - 1]);
            else if (index < 0)
                fuzzyFires[r] = fis_min(fuzzyFires[r], 1 - fuzzyInput[i][-index -
1]);
            else
                fuzzyFires[r] = fis_min(fuzzyFires[r], 1);
        }
    }
    else
    {
        fuzzyFires[r] = FIS_MIN;
        for (i = 0; i < fis_gcI; ++i)
        {
            index = fis_gRI[r][i];
            if (index > 0)
                fuzzyFires[r] = fis_max(fuzzyFires[r], fuzzyInput[i][index - 1]);
            else if (index < 0)
                fuzzyFires[r] = fis_max(fuzzyFires[r], 1 - fuzzyInput[i][-index -
1]);
            else
                fuzzyFires[r] = fis_max(fuzzyFires[r], 0);
```

```
                }
            }

            fuzzyFires[r] = fis_gRWeight[r] * fuzzyFires[r];
            sW += fuzzyFires[r];
        }

        if (sW == 0)
        {
            for (o = 0; o < fis_gcO; ++o)
            {
                g_fisOutput[o] = ((fis_gOMax[o] + fis_gOMin[o]) / 2);
            }
        }
        else
        {
            for (o = 0; o < fis_gcO; ++o)
            {
                g_fisOutput[o] = fis_defuzz_centroid(fuzzyRuleSet, o);
            }
        }
    }
```

APPENDIX D

Data Result of Experiment 2

Front sensor	RPM	Back sensor	RPM
59	0	51	0
44	0	61	0
44	264.06	55	264.06
43	721.76	58	721.76
46	537.49	71	537.49
45	706.71	72	706.71
43	668.23	50	668.23
49	0	102	0
98	0	114	0
101	1631.32	114	1631.32
100	1863.93	109	1863.93
100	1870.91	113	1870.91
98	1569.86	110	1569.86
150	269.4	160	269.4
148	1464.13	161	1464.13
148	1808.32	158	1808.32
147	1492.54	156	1492.54
149	1615.51	156	1615.51
159	1795.33	171	1795.33
158	1556.42	169	1556.42
156	1282.87	166	1282.87
155	1429.59	166	1429.59
152	1537.28	164	1537.28
151	1644.74	162	1644.74
151	1609.01	161	1609.01
151	1616.81	159	1616.81
150	1456.66	159	1456.66
151	293.64	158	293.64
151	1742.16	157	1742.16
200	0	208	0
206	1612.9	218	1612.9
204	1661.13	217	1661.13
205	1571.09	222	1571.09
203	1707.94	224	1707.94

204	1684.92	212	1684.92
204	1684.92	214	1684.92
202	1247.66	220	1247.66
201	1919.39	211	1919.39
202	1527.88	208	1527.88
205	1611.6	209	1611.6
206	0	213	0
261	0	261	0
259	678.66	268	678.66
301	684.7	305	684.7
78	2472.19	71	2472.19
301	595.42	304	595.42
298	1813.24	302	1813.24
81	2195.39	305	2195.39
296	2621.23	301	2621.23
299	1088.14	299	1088.14
301	1833.18	300	1833.18
301	307.74	306	307.74
298	2631.58	301	2631.58
294	1690.62	316	1690.62
298	1793.72	304	1793.72
297	772.2	299	772.2
296	2030.46	299	2030.46