SELF NAVIGATING WHEELCHAIR CONTROL
(OBSTACLE AVOIDANCE)

SOH YING WEI

A project report submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering (Hons.) Mechanical Engineering

Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

April 2013

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature : ___________________________

Name : Soh Ying Wei

ID No. : 09UEB08438

Date : 8th May 2013

APPROVAL FOR SUBMISSION

I certify that this project report entitled **"SELF NAVIGATING WHEELCHAIR CONTROL (OBSTACLE AVOIDANCE)"** was prepared by **SOH YING WEI** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Engineering (Hons.) Mechanical Engineering at Universiti Tunku Abdul Rahman.

Approved by,

Signature : _________________________

Supervisor : Prof. Dato. Dr Goh Sing Yau

Date : _________________________

ACKNOWLEDGEMENTS

I would like to thank everyone who had contributed to the successful completion of this project. I would like to express my gratitude to my research supervisor, Prof. Dato Dr. Goh Sing Yau for his invaluable advice, guidance and his enormous patience throughout the development of the research. Special thanks to Mr Danny Ng Wee Kiat, Mr Ho Chan Cheong and Mr Ho Kok Wai who had provided us their guidance and advices.

In addition, I would also like to express my gratitude to my loving family and friends who had helped and given me encouragement throughout the whole process.

SELF NAVIGATING WHEELCHAIR CONTROL (OBSTACLE AVOIDANCE)

ABSTRACT

Obstacle avoidance is one of the crucial features in a self-navigating wheelchair. The main aim of this project is develop an obstacle avoidance system for a self-navigating wheelchair in an indoor environment. A fuzzy logic controller for the obstacle avoidance system of a wheelchair was designed, developed and tested. Six ultrasonic sensors were used to provide feedback of the distance between the obstacle and the wheelchair. The controller processes the movement algorithm in the C language. Feedbacks from the ultrasonic sensors were used to determine the appropriate wheelchair movement to avoid the obstacle. Tests were conducted for an obstacle near to the left, near to the right and for no obstacle in an indoor environment. In all these three cases, the self-navigating wheelchair travelled smoothly, recognising and avoiding the obstacle.

TABLE OF CONTENTS

DECLARATION		**ii**
APPROVAL FOR SUBMISSION		**iii**
ACKNOWLEDGEMENTS		**v**
ABSTRACT		**vi**
TABLE OF CONTENTS		**vii**
LIST OF TABLES		**ix**
LIST OF FIGURES		**x**
LIST OF SYMBOLS / ABBREVIATIONS		**xi**
LIST OF APPENDICES		**xii**

CHAPTER

1	**INTRODUCTION**		**1**
	1.1	Project Background	1
	1.2	Aims and Objectives	2
	1.3	Scope of Work	2
	1.4	Problems Identification	2
2	**LITERATURE REVIEW**		**3**
	2.1	Introduction	3
	2.2	Development of Wheelchair with Obstacle Avoidance	4
	2.3	Control Method by Using Fuzzy Logic System	6
3	**METHODOLOGY**		**8**
	3.1	Overview	8

	3.2	Hardware Implementation	8
		3.2.1 Microcontroller (dsPIC33FJ128MC802)	9
		3.2.2 Digital Analog Converter, DAC (MCP4822)	10
		3.2.3 USB to UART Converter	11
		3.2.4 Ultrasonic Sensors (LV-MaxSonar-EZ1)	11
	3.3	Software Implementation	12
	3.4	Firmware	13
	3.5	Fuzzy Logic Controller (FLC)	13
		3.5.1 Membership Function (MF) / Fuzzy Sets	14
		3.5.2 Fuzzification	16
		3.5.3 Fuzzy Inference	17
		3.5.4 Fuzzy Rule Based System	17
		3.5.5 Defuzzification	17
		3.5.6 Design of Obstacle Avoidance FLC	18
4	**RESULTS AND DISCUSSION**		**21**
5	**CONCLUSION AND RECCOMENDATIONS**		**27**
	5.1	Conclusion	27
	5.2	Problems	27
	5.3	Recommendation	28
REFERENCES			**29**
LIST OF APPENDICES			**31**

LIST OF TABLES

TABLE	TITLE	PAGE
4.1	Right Obstacle Avoidance Data	23
4.2	Right and Left Obstacle Avoidance Data	25
A.1	Left Obstacle Avoidance Fuzzy Logic Table	36
A.2	Right Obstacle Avoidance Fuzzy Logic Table	36
B.1	Right Obstacle Avoidance	37
C.1	Left and Right Obstacle Avoidance	47

LIST OF FIGURES

FIGURE	TITLE	PAGE
2.1	Experimental Setup	5
2.2	Design Structure	6
3.1	SPI	10
3.2	The FLC and Relation of FLC with Conventional	14
3.3	Types of Membership Function	15
3.4	Fuzzification of a Car Speed	16
4.1	Voltage vs Input Distance (Right Avoidance)	26
4.2	Voltage vs Input Distance (Left Avoidance)	26
A.1	Circuit Design (Schematic)	31
A.2	Circuit Design (Printed Circuit Board)	32
A.3	LV-MaxSonar-EZ1 Ultrasonic Range Finder	32
A.4	Ultrasonic Sensor Beam Characteristic	33
A.5	Position of the Ultrasonic Sensors (Top View)	33
A.6	Flow Chart for Obstacle Avoidance with Fuzzy Logic Approach	34
A.7	Input Distance Membership Functions	35
A.8	Joystick Output Voltage at Different Position	35

LIST OF SYMBOLS / ABBREVIATIONS

R_i	Range in meter
V_{cc}	Supplied Voltage
V_i	Volts per Inch
V_m	Measured Voltage
ADC	Analogue to Digital Converter
DAC	Digital to Analogue Converter
FLC	Fuzzy Logic Controller
GUI	Graphic User Interface
I/O	Input or Output
RFID	Radio Frequency Identification
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus

LIST OF APPENDICES

APPENDIX	TITLE	PAGE
A	Figures and Tables	31
B	Right Obstacle Avoidance Data	37
C	Left and Right Obstacle Avoidance Data	47
D	Programming Source Code	59

CHAPTER 1

INTRODUCTION

1.1 Project Background

A self- navigating wheelchair refers to a wheelchair that has the ability to safely move towards a desired location using its knowledge and sensorial information of surrounding environment. Such a system enables a user to have more flexibility in terms of mobility and boost their independence which is able to reverse the cognitive impairment and depression. This results in improved quality of life. (Bates, Boot and Beverly 2004)

This project focuses on the control of a self- navigating wheelchair which has the capability of obstacle avoidance. The programming part of the obstacle avoidance system will be carried out using C language.

This project is a subpart of a main project of a wheelchair with Intelligent Distributed Controller and incorporated with a Brain Computer Interface (BCI). The obstacle avoidance system is crucial for the wheelchair in order for it to detect and avoid obstacles during its movement towards the desired location.

1.2 Aims and Objectives

The objectives of the project are:-

- To develop a controller for a self-navigating wheelchair to travel in an indoor environment
- To develop an obstacle avoidance Fuzzy Logic Controller (FLC) for the wheelchair in an indoor environment

1.3 Scope of Work

A fuzzy logic controller will be developed to receive inputs from ultrasonic sensors and send an output voltage to drive the motor so that the wheels which are actuated by the motor are able to response according to the feedback distance received from the ultrasonic sensors.

1.4 Problems Identification

For an autonomous system, one of its basic tasks will be obstacle avoidance. Obstacle avoidance plays an important role in the design of a self-navigating wheelchair with safety, comfort and smoothness as well. Therefore, the main challenge is to develop an algorithm and strategy which are able to assist a wheelchair to recognise obstacles and to go around it by maintaining an optimum distance. The obstacles are walls and solid objects.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

A self- navigating wheelchair that is developed for people with impaired mobility must be able to navigate autonomously within a certain workspace. The wheelchair is equipped with multiple sensor used for mapping, path planning and obstacle avoidance. (Simpson 2004)

A self -navigating wheelchair with obstacle avoidance refers to the one that can shape the path to avoid unforeseen obstacles. The actual location and sensor readings will determine the motion of the wheelchair. Besides that, there are various algorithms ranging from basic re-planning to reactive changes in the control strategy that can be used in obstacle avoidance. The techniques used will depend on the use of sensorial data and strategy of controlling the motion to overcome obstacles. (Riberio 2005)

2.2 Development of Wheelchair with Obstacle Avoidance

Various algorithms for enabling autonomous tasks of wheelchair in an unknown environment have been developed such as occupancy grid, global map, virtual force field vector field histogram and fuzzy logic.(Elfes 1989), (Borenstein 1990), (Borenstein, 1991) .

In 2002, a robotic wheelchair that was capable of following a specified target while moving in the middle of free space was developed. The robotic wheelchair was able to determine the target positioning with respect to itself via the colour sequence image which was processed by a panoramic camera. Several sonars installed on the wheelchair were used to measure the distance of the wheelchair from the target. (Argyros et al., 2002)

An agent based robotic wheelchair has the function of path planning, navigation and obstacle avoidance in its controller. The path planning is developed by using the A* algorithm whereas the obstacle avoidance is carried out by using fuzzy logic. (C. H. Kuo, H. L. Huang and M. Y. Lee 2003)

The Intelligent Powered Wheelchair system was developed to move in a track which was formed by black tape which is laid down of the floor so that the wheelchair can be lead to the desired destinations. (Ng 2011)

Camera sensors were installed on the wheelchair to guide the wheelchair along the track so that the signals can be sent to the data acquisition board. Then, the fuzzy logic system will use the signals as feedback to make sure that the wheelchair is moving along the guided track. Thus, when the wheelchair position is at the most right or most left, the wheelchair needs to adjust its position to rectify the error. These are carried out by setting the rules in the fuzzy logic which determines the control action of the wheelchair. However the number of rules should be constrained to certain number in order to avoid slowing down of the whole process. (Ng 2011)

The fuzzy logic plays a role in controlling the speed of the wheelchair as well. The purpose of speed control is to maintain the speed of wheel at constant speed when it is moving in a straight line and slowing down when there are obstacles nearby or when making a turn. The data for the controller to adjust the speed is obtained from the encoder sensor and camera sensor. In addition, a RFID reader and ultrasonic sensors are mounted on the wheelchair so that the wheelchair current location can be identified and obstacles along the track can be detected. (Ng 2011)

Figure 2.1 shows that the tuning of fuzzy logic controller is done in an indoor environment where the position of lines and obstacles are pre-set. From Figure 2.2, author used the triangular membership function to define it's fuzzy set for simplicity and ease of using. (Ng, 2011)

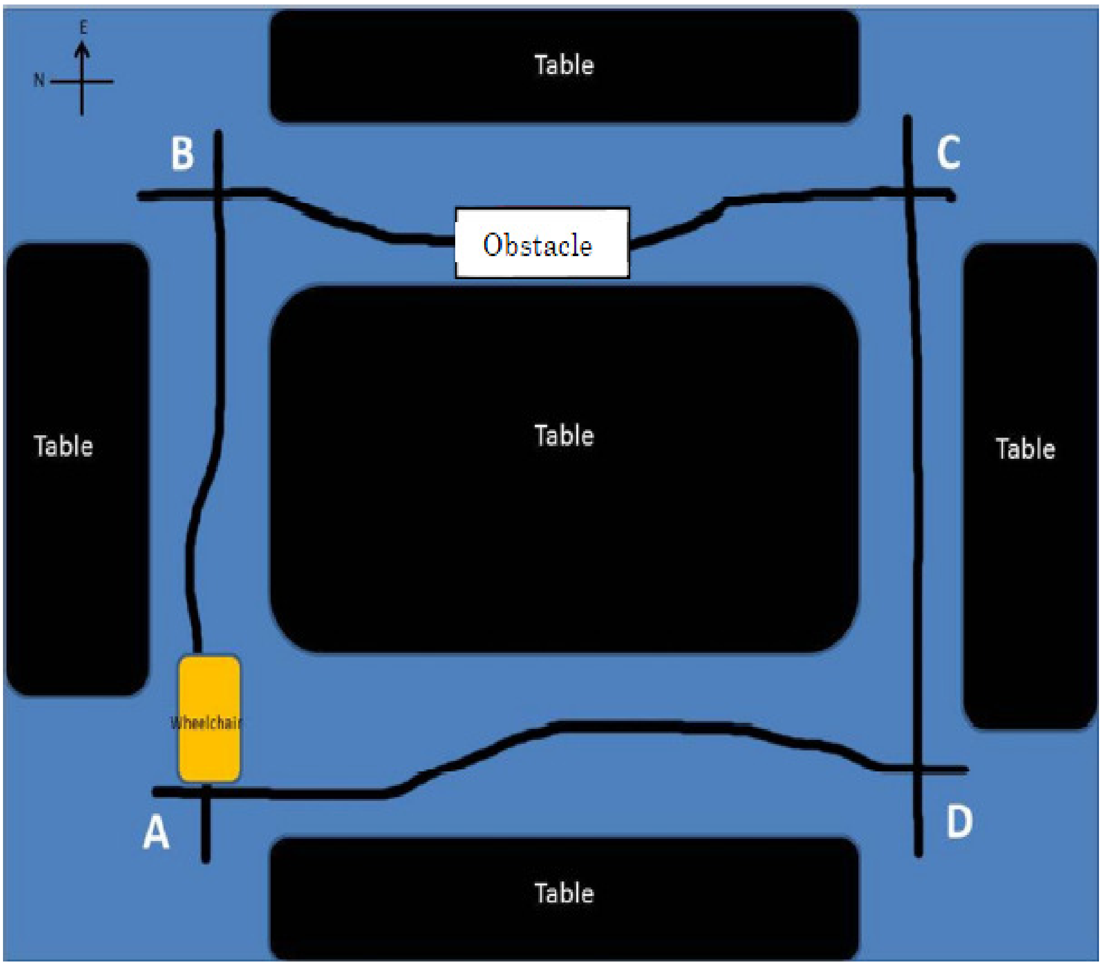

Figure 2.1: Experimental Setup

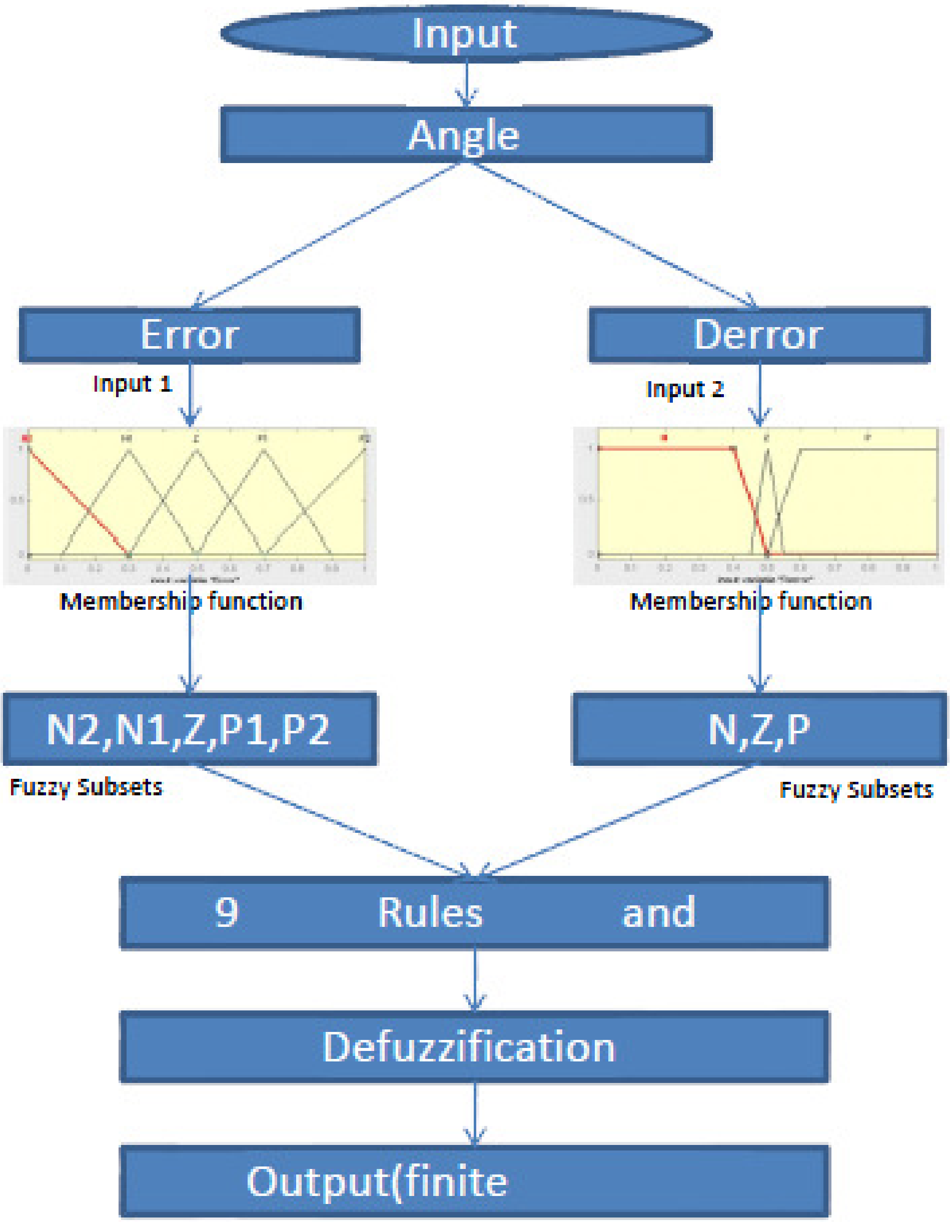

Figure 2.2: Design Structure

2.3 Control Method by Using Fuzzy Logic System

Most of the obstacle avoidance system in wheelchair implemented the fuzzy logic system that was introduced by Lofti Zadeh in a 1965's seminal paper. It is because fuzzy logic is a method used to solve problem in various system such as embedded micro-controllers which are simple and small, computers or workstation-based data acquisition which are large, networked, multi-channel and control systems. Although the input is imprecise, vague, and noisy or contains information that is missing, fuzzy logic is still able to reach a definite conclusion in a simple way. (Xu, W.L., Tso, S.K, and Fung, Y.H., 1998)

Furthermore, fuzzy logic method is similar to the ways of a human makes decision in solving problems. It is because fuzzy logic controller processes user defined rules which can be modified and tweaked easily. (Xu, W.L., Tso, S.K, and Fung, Y.H., 1998)

Obstacle avoidance systems that do not use fuzzy logic work by calculating the probability of an obstacle present in each square of the grid based on the data obtained via dividing an area surrounding wheelchair into a grid pattern. This requires immense calculations which lead to increase in response time of the autonomous wheelchair. Systems that operate using fuzzy logic will minimise the calculations and instructions required which results in increase processing speed of microcontroller and reaction time of autonomous wheelchair. Other than that, fuzzy logic is suitable for controlling non-linear system such as obstacle avoidance because it is difficult to model non-linear systems mathematically. (Xu, W.L., Tso, S.K, and Fung, Y.H., 1998)

CHAPTER 3

METHODOLOGY

3.1 Overview

The Fuzzy Logic Controller (FLC) for obstacle avoidance consisted of two ultrasonic sensors mounted to the front, two on the left and two on the right of the wheelchair for detecting obstacles (Figure A.5) and a PCB board for the controller (Figure A.2). The system is controlled by using fuzzy logic algorithm which was tested and tuned for functionality verification and performance improvement.

3.2 Hardware Implementation

The fuzzy logic control system was executed by using a fuzzy logic controller board (Figure A.1) consisting of the following components:

1. Microcontroller (dsPIC33FJ128MC802)
2. Digital Analog Converter (MCP4822)
3. USB to UART converter
4. Ultrasonic Sensors (LV-MaxSonar-EZ1)

3.2.1 Microcontroller (dsPIC33FJ128MC802)

The microcontroller dsPIC33FJ128MC802 was the master controller of the whole fuzzy logic system for obstacle avoidance. This controller was chosen because it incorporated all the elements needed for obstacle avoidance which was built in as modules. The following modules were used for obstacle avoidance fuzzy logic controller:

1. Analog to Digital Module (ADC)
2. Serial Peripheral Interface Module (SPI)
3. Universal Asynchronous Receiver/Transmitter Module (UART)

3.2.1.1 Analogue to Digital Module (ADC)

The ADC was used to convert continuous analogue voltage obtained from ultrasonic sensors to digital number which is proportional to the magnitude of the voltage. Then, the digital values will be used as input for fuzzy logic control algorithm. The configuration in microcontroller ADC is set to 12-bits so that it is compatible the DAC which has 12-bits configuration as well. For 12-bits ADC, it only has one sample/hold amplifier which means that there is only one channel available for sampling. Besides that, it only contains a single–word result buffer, ADC1BUF0 where the results is continuously updated and stored in the respectively variables for fuzzy logic algorithm purpose.

3.2.1.2 Serial Peripheral Interface Module (SPI)

The SPI is named by Motorola and it is a standard form of synchronous serial data link which works in full duplex mode. It was used as a communication interface between microcontroller and DAC in this project. The microcontroller communicates in master mode where it initiates the data frame. Since there is only a single slave device which is DAC, the slave select pins (SS) is fixed to logic low. Figure 3.1 SPI shows the connection diagram of SPI interface with single slave device.

Figure 3.1 SPI

(Source: http://www.byteparadigm.com)

3.2.1.3 Universal Asynchronous Receiver/ Transmitter Module (UART)

The UART is one of the serial I/O modules available in the dsPIC33F device family. For the FLC, UART is used as a communication tool between microcontroller and computer where ultrasonic sensors output will be transmitted to computer from time to time via the USB UART converter.

3.2.2 Digital Analog Converter, DAC (MCP4822)

The wheelchair that was used in this project is a powered wheelchair which is controlled by joystick via its own wheelchair driver. A MCP4822 was used to bypass the joystick control. It receives instructions of the fuzzy logic algorithm from master microcontroller. After it receives the digital signal from microcontroller, it converts the digital signal from fuzzy logic algorithm to an analog voltage for driving the wheelchair for the obstacle avoidance purpose.

3.2.3 USB to UART Converter

The USB to UART Converter is an interface used for communication between the computer and the microcontroller. The purpose of using this converter is to enable visualisation of the location of obstacles sensed by all the ultrasonic sensors. The data will be transmitted from the ultrasonic sensors to the computer from time to time. Thus, the visualisation of the data can be obtained by sending the ultrasonic sensors analogue output values to Hyper Terminal software.

3.2.4 Ultrasonic Sensors (LV-MaxSonar-EZ1)

Ultrasonic Sensors were used for obstacle avoidance by detecting objects via ultrasound beams. Basically the beam will move in a uniform medium in a straight direction until it hits an object which will reflect and transmit back the beam. The distance of an object from the wheelchair is determined by calculating the time interval between the sending of ultrasonic waves and receiving the echo.

LV-MaxSonar-EZI (Figure A.3) was chosen for this project due to its low power consumption and its ability to provide the range of sonar detection from 0.1524 m up to 6.451 m with 0.0254 m resolution. The characteristic of ultrasonic beams is shown in Figure A.4.

Therefore, the wheelchair is equipped with six ultrasonic sensors for obstacle avoidance which will be controlled by the FLC. The analog output which is in terms of voltage from ultrasonic sensors will be transmitted to the analog pins of microcontrollers so that the distance of obstacles from wheelchair can be determined.

The formulas used for calculating the distance using the voltage output from ultrasonic sensors are as following:

$$V_i = \frac{V_{cc}}{512} \tag{3.1}$$

Where

V_{cc} = Supplied voltage

V_i = Volts per inch (Scaling)

The range being detected is calculated as followed:

$$R_i = \frac{V_m}{V_i} \times 0.0254 \tag{3.2}$$

Where

V_m = Measured voltage

V_i = Volts per inch (Scaling)

R_i = Range in meter

3.3 Software Implementation

MPLAB X IDE (version 1.60) was used for programming the microcontroller by using C language whereas EAGLE 6.4.0 was used for the controller board design. Other than that, the user interface for visualising the output of ultrasonic sensors was carried out by using Hyper Terminal software.

3.4 Firmware

The development of firmware was carried out by using MPLAB X IDE and being programmed in terms of C language. MPLAB X IDE will assist the translation of codes being written in terms of C language to machine codes which will be burned into the PIC microcontroller's flash program memory.

3.5 Fuzzy Logic Controller (FLC)

Fuzzy logic system will be implemented in the wheelchair navigation for obstacle avoidance because fuzzy logic is a good method for inconsistent problems, easy to understand and flexible. The steps involved in fuzzy logic are membership functions or fuzzy sets, fuzzification, fuzzy reasoning (including rule base and fuzzy inference) and defuzzification. (Ibrahim 2004).

The FLC is shown in Figure 3.2 where there is similarity between FLC system and basic conventional control system. It is because FLC procedures is designed to emulate the traditional controller actions by applying fuzzy rules and adding features to resolve special cases such as non-linearity. (Ibrahim 2004)

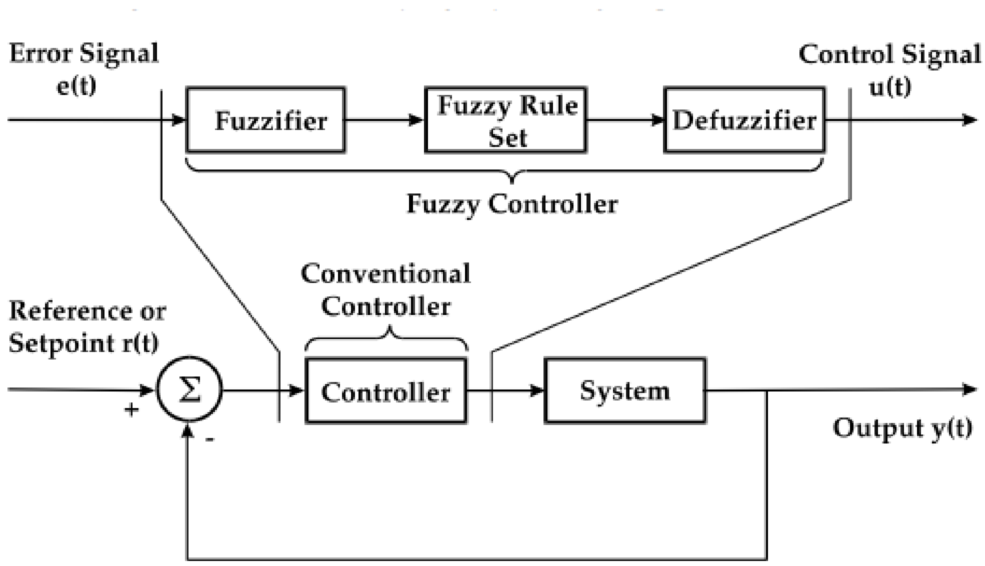

Figure 3.2: The FLC and Relation of FLC with Conventional Control Loop (Ng 2011)

3.5.1 Membership Function (MF) / Fuzzy Sets

A fuzzy set is the classical set theory's extension where there are varying degrees of membership for elements. Sometimes, it is insufficient to describe human reasoning with two truth values, true and false. Therefore, fuzzy logic describes human reasoning in the whole interval between 0 and 1. This enables linguistic term like tall, short, hot and cold to be formed in terms of membership value. There are three control parameters suggested in this project which are distance, acceleration and direction. The input data from sensors will be the input of FLC. Then, the error which is the difference between set position and current position and rate of error are translated from input of FLC. (Ibrahim 2004)

$$Z[n] = SetP[n] - Pos[n] \quad (3.3)$$

Where,

Z[n] = Error,

SetP[n] = Set point (the desired point for wheelchair)

Pos[n] = Current position of wheelchair

Rate of Error is the difference between previous error and current error

$$Z'[n] = Z[n] - Z[n-1] \quad (3.4)$$

Where,
Z'[n] = Rate of Error,
[n] = positive integer
(Ng, 2011)

There are various shapes of membership functions such as trapezoidal, triangular, Gaussian and etc. which can be used as shown in Figure 3.3. However, the most common membership functions are triangular and trapezoidal.

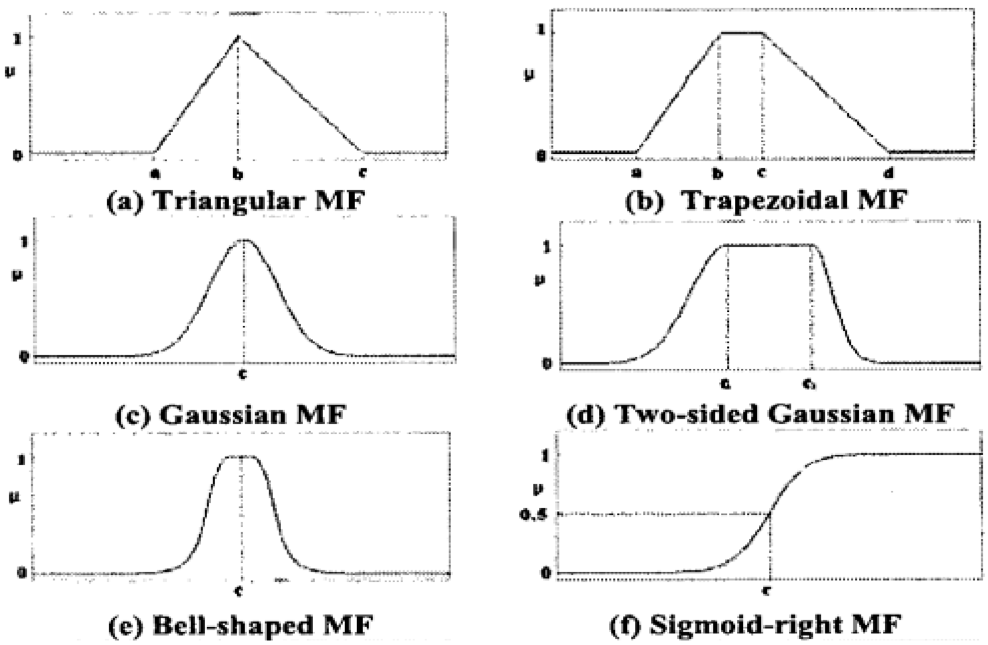

Figure 3.3: Types of Membership Function

Source: (http://power.eecs.utk.edu/pubs/bose_iecon_ies_ieee_ac_2002.pdf)

Basically there are two axes which form the membership function graph where the horizontal axis represents the input variable which can be any range of numbers according to user preference. Whereas the vertical axis represents the fuzzy value ranging from 0 to 1 where zero degree represents that the value is totally not in the set and degree of one represents that the value is in the set completely. Then, intermediate values indicate partial membership in the set. (Ibrahim 2004)

3.5.2 Fuzzification

Crisp values being transformed into membership grades for fuzzy set's linguistic terms is the process contained in fuzzification. The purpose is to allow a fuzzy condition in a rule to be interpreted. (Ibrahim 2004)

Based on the example shown in Figure 3.4, when the car speed x_o=40 km/h, the grade $\mu_a(x_o)$= 0.6 belongs to low and $\mu_b(x_o)$ = 0.4 belongs to fuzzy medium.

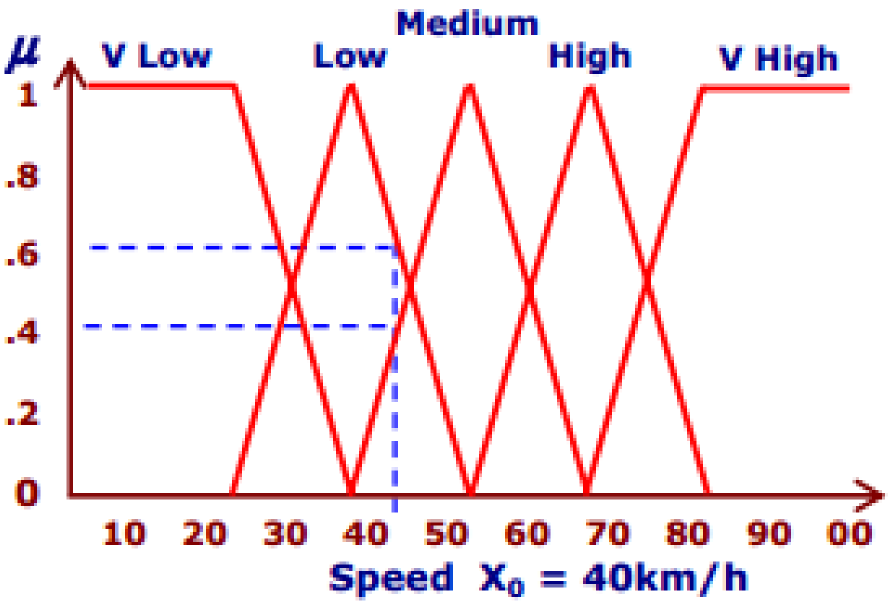

Figure 3.4: Fuzzification of a Car Speed

(Source : http://www.atp.ruhr-uni-bochum.de/rt1/syscontrol/node122.html)

3.5.3 Fuzzy Inference

Fuzzy Inference is the core element of a fuzzy system where it executes the process of fuzzy reasoning by combining the fact acquired from fuzzification and the rule base. It is also known as approximate reasoning which is a computational procedures used for evaluating linguistic description. (Ibrahim 2004)

3.5.4 Fuzzy Rule Based System

The fuzzy linguistic descriptions are formal representation of systems made through fuzzy IF –THEN rule. They encode knowledge about a system in statements of the form: IF (a set of conditions) are satisfied THEN (a set of consequents) can be inferred. IF ($x_1 is\ A_1$, $x_2 is\ A_2$, $x_n is\ A_n$) THEN ($y_1\ is\ B_1$, $y_2\ is\ B_2$, $y_n\ is\ B_n$) where linguistic variables x_i, y_j take the values of fuzzy sets of A_i and B_j respectively.

(Ibrahim 2004)

3.5.5 Defuzzification

A rule-based system which is written in terms of if-then rules is known as fuzzy logic. These rules are stored in the knowledge base of the system. The input to the fuzzy system is a scalar value that is fuzzified. The set of rules is applied to the fuzzified input. Fuzzy is each rule's output. Fuzzy outputs had to be transformed into a scalar output quantity so that the system can determine the type of the action to be executed. On the other hand, the course of transforming the fuzzy output is called defuzzification. (Ibrahim 2004)

1. Centroid method – It is also known as center of mass or center of gravity (COG). One of the defuzzification methods which is quite commonly used.

$$x^* = \frac{\int \mu_F(x).x dx}{\int \mu_F(x) dx} \tag{3.5}$$

Where $\int$ denotes algebraic integration

2. First maxima – this method is used for peaked output where the smallest value of the domain with highest membership is chosen.

3. Weighted average method- also known as Sugeno-Defuzzification. This method is usable for membership functions with symmetrical output. Each membership is weighted by its maximum membership value.

$$x^* = \frac{\sum \mu_F(\bar{x}_i)\bar{x}_i}{\sum \mu_F(\bar{x}_i)} \tag{3.6}$$

$\sum$ denotes algebraic summation and $\bar{x}_i$ is the maximum of *i*th membership function.

4. Centre of largest area - this technique is good when the output contains of at least two convex fuzzy subsets which do not overlap.

3.5.6 Design of Obstacle Avoidance FLC

The wheelchair obstacle avoidance method was based on the wall following control method where the wheelchair would maintain at a desired distance from the wall or obstacle to navigate in a closed indoor environment. The range of input distance for fuzzy logic algorithm would be set from 0 cm to 120 cm where 60 cm will be the desired distance.

The ultrasonic sensors are placed at approximately 20 cm horizontal distance apart at the front side of the wheelchair and the minimum width of obstacle that can be detected by the ultrasonic sensors will be approximately 21 cm. For the two ultrasonic sensors which are located at the left side of the wheelchair, they are mounted at 40 cm horizontal distance apart. The two ultrasonic sensors at the right side of the wheelchair are mounted at 40 cm horizontal distance apart as well. The minimum width of obstacle that can be detected by the ultrasonic sensors on the left and right will be approximately 23 cm.

Initially the wheelchair will be set on moving straight without any obstacles. Then, the front ultrasonic sensors will be used for detecting obstacles at the front of wheelchair. If there is no obstacle within one meter range, then the wheelchair will move straight. If there is an obstacle within one meter, the front left and right ultrasonic sensors input distance will be compared. If the obstacle is nearer to the front left ultrasonic sensor, then the left obstacle avoidance fuzzy logic algorithm will be executed whereas if obstacle is nearer to the front right ultrasonic sensor, then the right obstacle avoidance fuzzy logic algorithm will be executed (Figure A.6).

Basically there are three membership functions for input distance which are near, just right and far (Figure A.7).For example the obstacle is detected too near to the left of wheelchair, then the acceleration of wheelchair will be slow and wheelchair will turn right until the distance between the left side of wheelchair and obstacle is 60 cm. However, when left obstacle avoidance is still executed yet the wheelchair is far from the obstacle, the wheelchair acceleration will be slow and it will turn left until the distance between the left side of the wheelchair and obstacle is 60 cm.

Therefore, rules play a very crucial role in fuzzy logic design as it will determine the control action of the wheelchair. The rules formed for left obstacle avoidance and right obstacle avoidance are as shown in Table A.1 and Table A.2.

<u>Left Obstacle Avoidance</u>

1. If the obstacle is near, then the acceleration is low and the wheelchair turns right.
2. If the obstacle is just right, then the acceleration is fast and the wheelchair goes straight.
3. If obstacle is far, then the acceleration is low and the wheelchair turns left.

Right Obstacle Avoidance

1. If the obstacle is near, then the acceleration is low and the wheelchair turns left.
2. If the obstacle is just right, then the acceleration is fast and the wheelchair goes straight.
3. If the obstacle is far, then the acceleration is low and the wheelchair turns right.

Weighted average method is used for the defuzzication process of acceleration and direction output due to its ease of computation.

CHAPTER 4

RESULTS AND DISCUSSION

Tests were conducted for validating the FLC system on obstacle avoidance and the results are included in (Table B.1, Table C. 1). Table 4.1 and Table 4.2 as shown in this chapter are the values extracted from the Appendix. The results are discussed as below.

The voltages shown are the voltage X and voltage Y of the joystick controller. The FLC system was used to bypass the original controller of the wheelchair by emulating the joystick output voltage.

Therefore, voltage Y around 2.467 V indicates stop (neutral position). Voltage Y greater than neutral position indicates that the wheelchair accelerates until it reaches the maximum voltage of 3.709 V. For voltage X, the voltage around 2.467 V indicates neutral moving straight (neutral position). Voltage X lesser than 2.467 V indicates the wheelchair will move towards the left direction where highest speed for left direction is about 1.160 V. Voltage greater than 2.467 V indicates the wheelchair will move towards the right direction where maximum speed for right direction is about 3.904 V (Figure A.8). In short, voltage Y represents the acceleration and voltage X represents the left and right direction.

The tests were conducted at low acceleration and medium acceleration where the voltage X ranges from 2.3 V to 2.7 V whereas the voltage Y ranges from 2.3 V to 3.1 V. The voltage range is shown in the yellow region of Figure A.8.

For Table 4.1, Sensor 3 and Sensor 4 are the ultrasonic sensors on the right as indicated in Figure A.5. It can be observed that:

1. From row one to eight, the minimum distance for all the eight data are from sensor 4. Since the distance shown are less than 60 cm (near), voltage Y is around 2.70 V to 2.80 V for slowing down and voltage X is around 2.20 V to 2.30 V which indicates that the wheelchair is moving towards the left or far from the obstacle or wall.

2. From row 9 to 19, the minimum distances for the data are from sensor 3. Since distance shown are greater than 60 cm (far), voltage Y is around 2.70 V to 2.80 V for slowing down and voltage X is around 2.50 V to 2.60 V which indicates that the wheelchair is moving towards the right so that it can reduce the distance to about 60cm from the obstacles or wall.

3. From row 20 to 29, the minimum distances for the data are from sensor 4. Since distance shown are approximately 60 cm, the voltage Y is around 3.00 V to 3.10 V which indicates that the wheelchair is moving fast when it has keep optimum distance from obstacle or wall. The voltage X is about 2.40 V to 2.45 V which indicates that the wheelchair is moving straight when obstacles or wall are kept at optimum distance from wheelchair.

4. Algorithm for right obstacle avoidance can be visualised more clearly by viewing the graph as shown in Figure 4.1.

Table 4.1: Right Obstacle Avoidance Data

	Sensor 3 Distance (cm)	Sensor 4 Distance (cm)	Voltage Y (V) Acceleration	Voltage X (V) Acceleration
1	101.605	42.865	2.846	2.316
2	101.605	37.467	2.764	2.268
3	101.605	36.832	2.755	2.262
4	101.605	36.832	2.755	2.262
5	119.386	37.149	2.760	2.265
6	71.123	37.467	2.764	2.268
7	42.865	37.467	2.764	2.268
8	104.463	39.689	2.798	2.288
9	46.357	37.467	2.764	2.268
10	80.014	99.382	2.796	2.590
11	79.379	99.382	2.806	2.586
12	77.474	208.608	2.834	2.574
13	77.474	102.875	2.834	2.574
14	82.554	105.415	2.758	2.605
15	74.299	129.546	2.882	2.555
16	74.934	99.382	2.872	2.559
17	75.251	99.382	2.867	2.561
18	73.029	96.842	2.901	2.547
19	91.444	113.353	2.650	2.650
20	87.317	96.525	2.687	2.634
21	71.441	61.280	3.076	2.476
22	59.375	59.375	3.095	2.464
23	59.375	59.375	3.095	2.464
24	59.375	55.883	3.042	2.432
25	152.725	55.883	3.042	2.432
26	58.423	57.153	3.061	2.444
27	84.459	57.470	3.066	2.447
28	87.317	54.295	3.018	2.418
29	84.459	53.025	2.999	2.407
30	65.408	57.153	3.061	2.444

For Table 4.2, the columns highlighted represent the position of ultrasonic sensors (Figure A.5). From Table 4.2, it can be observed that:

1. From row 1 to 13, the minimum distances by comparing Sensor 1 and Sensor 2 are the distances detected from Sensor 1. Thus, left obstacle avoidance algorithm is executed. The minimum distance obtained from sensor 5 and 6 will be taken into consideration. Since the minimum distances obtained are less than 60 cm, the wheelchair acceleration is low. Voltage X is about 2.8 V which indicates that the wheelchair kept turning right so that it moved away from the wall or obstacles to obtain optimum distance.

2. From row 14 to 28, the minimum distances by comparing Sensor 1 and Sensor 2 are the distances detected from Sensor 2. Thus, right obstacle avoidance algorithm is executed. By comparing Sensor 3 and 4, Sensor 3 detects the minimum distance. When the distance detected was around 35 cm, the wheelchair acceleration is low and it turned left. When the distance detected was greater than 60 cm, the wheelchair acceleration is low and it turned right so that the wheelchair moved towards the wall or obstacles to obtain optimum distance.

3. From row 29 to 42, the minimum distances by comparing Sensor 1 and Sensor 2 are the distances detected from Sensor 1. Thus, left obstacle avoidance algorithm is executed. From Sensor 5 and 6, the distances detected are still lesser than 60 cm. Thus, the acceleration is low and it turned right where it moved away from the wall or obstacles to obtain optimum distance.

4. Algorithm for left obstacle avoidance can be visualised more clearly by the graph as shown in Figure 4.2.

Table 4.2: Right and Left Obstacle Avoidance Data

	Relative Distance between Obstacle and Wheelchair (cm)								Voltage Y (V) Acceleration	Voltage X (V) Direction
1	1	64.456	2	205.75	5	32.704	6	44.452	2.693	2.768
2	1	32.387	2	212.1	5	38.102	6	6.350	2.650	2.800
3	1	81.919	2	218.451	5	24.766	6	8.890	2.650	2.800
4	1	90.492	2	218.451	5	23.179	6	106.685	2.650	2.800
5	1	96.525	2	218.451	5	23.179	6	167.648	2.650	2.800
6	1	90.492	2	215.91	5	21.274	6	40.642	2.650	2.800
7	1	90.492	2	236.231	5	22.861	6	15.241	2.650	2.800
8	1	80.966	2	215.91	5	24.131	6	35.562	2.650	2.800
9	1	90.492	2	238.772	5	22.861	6	71.123	2.650	2.800
10	1	86.364	2	212.1	5	18.098	6	35.562	2.650	2.800
11	1	87.317	2	213.37	5	22.861	6	35.562	2.650	2.800
12	1	87.952	2	212.418	5	23.179	6	8.255	2.650	2.800
13	1	58.423	2	218.451	5	18.098	6	32.704	2.650	2.800
14	1	100.65	2	96.525	3	35.244	4	165.108	2.731	2.248
15	1	124.78	2	100.017	3	94.302	4	193.049	2.650	2.650
16	1	139.71	2	97.477	3	101.605	4	152.725	2.650	2.650
17	1	130.82	2	96.842	3	113.988	4	182.889	2.650	2.650
18	1	41.91	2	39.689	3	71.123	4	105.733	2.929	2.536
19	1	105.10	2	66.361	3	68.901	4	105.415	2.962	2.522
20	1	71.44	2	71.123	3	66.996	4	104.463	2.991	2.511
21	1	88.59	2	64.456	3	71.123	4	87.317	2.929	2.536
22	1	94.30	2	71.123	3	66.996	4	101.922	2.991	2.511
23	1	106.37	2	66.043	3	66.996	4	101.922	2.991	2.511
24	1	93.67	2	63.821	3	68.266	4	99.382	2.972	2.519
25	1	131.45	2	61.916	3	66.996	4	100.017	2.991	2.511
26	1	109.23	2	72.711	3	64.456	4	99.065	3.028	2.495
27	1	111.77	2	58.740	3	53.343	4	99.382	3.004	2.410
28	1	111.77	2	60.010	3	74.616	4	102.875	2.877	2.557
29	1	87.634	2	208.29	5	20.321	6	32.704	2.650	2.800
30	1	87.634	2	200.987	5	22.861	6	9.843	2.650	2.800
31	1	93.985	2	195.589	5	23.179	6	30.481	2.650	2.800
32	1	87.952	2	190.509	5	20.639	6	30.481	2.650	2.800
33	1	87.634	2	212.1	5	17.781	6	11.113	2.650	2.800
34	1	32.069	2	180.349	5	18.098	6	30.799	2.650	2.800
35	1	85.094	2	212.1	5	18.098	6	0.000	2.650	2.800
36	1	85.094	2	211.148	5	21.274	6	17.781	2.650	2.800
37	1	85.094	2	256.552	5	16.511	6	34.292	2.650	2.800
38	1	96.525	2	137.167	5	18.098	6	20.321	2.650	2.800
39	1	83.824	2	208.29	5	20.639	6	17.781	2.650	2.800
40	1	82.554	2	208.29	5	20.639	6	0.953	2.650	2.800
41	1	82.554	2	121.608	5	20.639	6	0.000	2.650	2.800
42	1	80.966	2	115.258	5	18.098	6	20.639	2.650	2.800

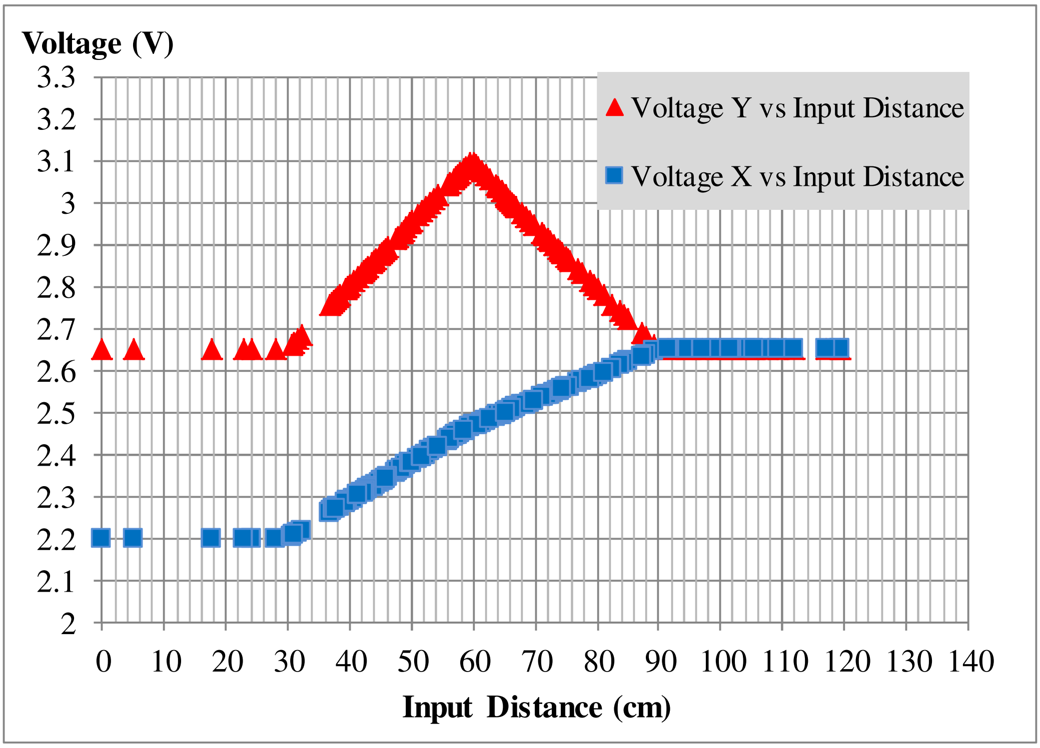

Figure 4.1: Voltage vs Input Distance (Right Avoidance)

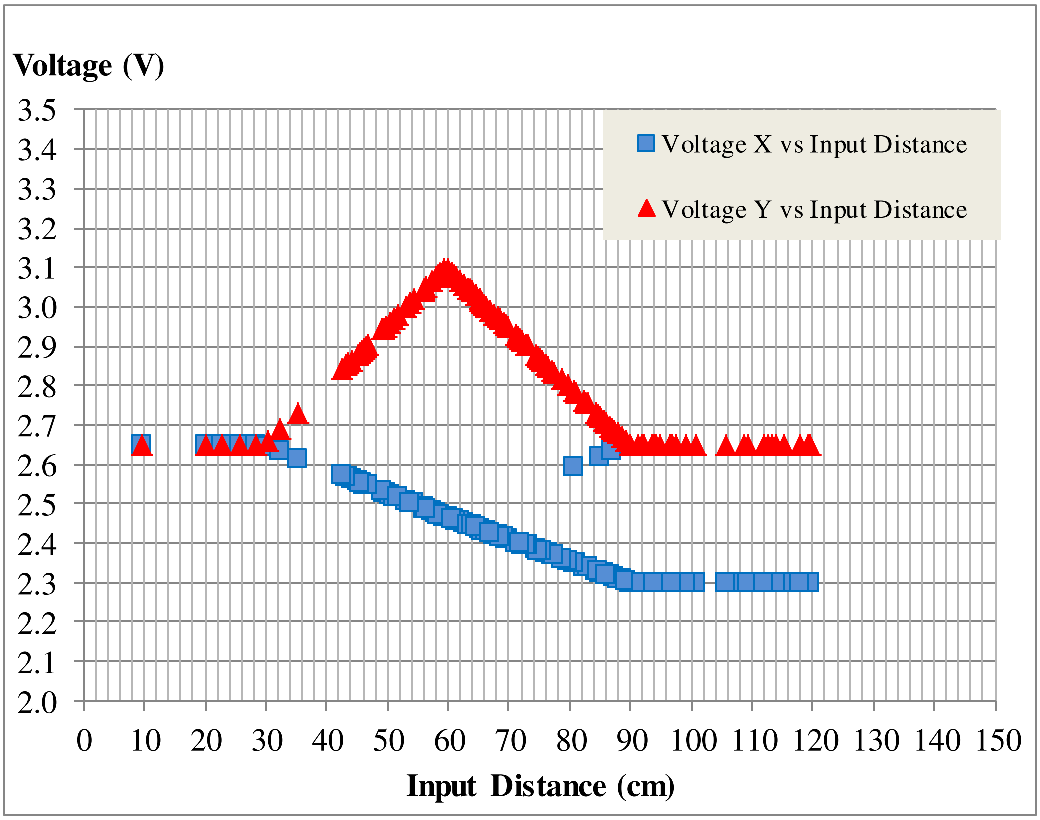

Figure 4.2: Voltage vs Input Distance (Left Avoidance)

CHAPTER 5

CONCLUSION AND RECCOMENDATIONS

5.1 Conclusion

A Fuzzy Logic Controller (FLC) for obstacle avoidance was designed, built and tested on a self-navigating wheelchair in an indoor environment. Six ultrasonic sensors were used to provide feedback of the distance between the obstacle and the wheelchair to the FLC. The wheelchair successfully manoeuvred past the obstacles when they were placed on the left or right of wheelchair in a closed indoor environment.

5.2 Problems

The FLC for obstacle avoidance performed as expected but it still had certain limitations. The ultrasonic sensors were high in sensitivity where 6.445mV is equivalent to 0.0254 m. However, when the ultrasonic sensor measured distances which produced equivalent voltages in the range of 100 mV to 200 mV, it would lead to inaccuracy when this voltage was being converted in microcontroller due to noise which appeared in the 100 mV to 200 mV range. Consequently, the voltage detected in microcontroller would be slightly lower than the actual voltage sent out by ultrasonic sensors which could lead to 5 to 20 cm difference between the actual distance and distance captured by microcontroller. Therefore the desired distance between the wheelchair and obstacle or wall was set to 60 cm.

Other than that, the ultrasonic sensors worked by emitting ultrasonic waves and received the reflected ultrasonic waves from the surface of object for distance measurement. However, it only worked perfectly for flat surface. When the waves hit the surface which was uneven or corners, it would lead to fluctuation or inaccuracy in the distance measured.

5.3 Recommendation

The FLC for obstacle avoidance performed satisfactorily as expected in indoor environment. However it has certain limitations as mentioned in the problems. Thus, the controller for obstacle avoidance can be greatly improved by integrating ultrasonic sensors with a laser range finder for greater view of the position of obstacles.

The distance measured by the ultrasonic sensors, Voltage Y output (acceleration) and Voltage X output (direction) were sent to Hyper Terminal software for data collection and monitoring purposes by using UART USB converter which is not so convenient. Hence, a wireless Bluetooth USB communication should be used to replace it.

REFERENCES

Argyros, A. et al., 2002. Semi-Autonomous Navigation of a Robotic Wheelchair. *Journal of Intelligent and Robotic Systems*, 39(2), pp. 315–329.

Borenstein, J., 1990. Real-time Obstacle Avoidance for Fast Mobile Robots in Cluttered Environments. *IEEE International Confe Conference on Robotics and Automation*, pp. 572 - 577.

Borenstein, J., 1991. The Vector Field Histogram-Fast Obstacle Avoidance for Mobile Robots. *IEEE Transactions on Robotics and Automation*, 7(3), pp. 278 - 288.

C. H. Kuo, H. L. Huang and M. Y. Lee., 2003. Development of Agent-Based. Autonomous Robotic Wheelchair Control Systems. *Journal of Biomedical Engineering - Applications, Basis, Communications*, 15(6), pp. 12–23.

Elfes, A., 1989. Using Occupancy Grids for Mobile Robot. *Computer*, 22(6), pp.46-57. Available through http://biorobotics.ri.cmu.edu/papers/sbp_papers/integrated4/elfes_occup_grids.pdf [Accessed 30th July 2012]

Xu W.L., Tso S.K, and Fung Y.H, 1998. Fuzzy Reactive Control of a Mobile Robot Incorporating a Real /Virtual Target Switching Strategy. *Robotics and Automation Systems*, 23, pp. 171-186.

Ibrahim, A. M., 2004. *Fuzzy Logic for Embedded Systems Applications.*USA: Elsevier Science.

Bates, J., Boote, J., and Beverly, C., 2004. Psychosocial Interventions for People with Dementing Illness: A systematic review. *Journal of Advanced*, 45(6), pp. 644–658.

Martin, W., 2009. *Autonomous Robot Obstacle Avoidance Using a Fuzzy Logic Control Scheme.* [Online]. Available through http://www.cs.cmu.edu/~wmartin/files/fuzzy_control_robot.pdf [Accessed 30th June 2012]

Mendel, J., 1995. Fuzzy logic systems for engineering: a tutorial. *Proceedings of the IEEE*, 83(3), 345-377.

Ng, C.S., 2011. *An Intelligent Distributed Controller for a Wheelchair that is Operated by a Brain Computer Interface.* KL. Available through http://eprints.utar.edu.my/378/1/ESA-2011-0808106-1.pdf [Accessed 5th August 2012]

Simpson, R. et al., 2004. The Smart Wheelchair Component System. *Journal of Rehabilitation Research & Development*, 41(3B), pp. 429–442.

Riberio, M. I., 2005. *Obstacle Avoidance.* [Online]. Available through http://users.isr.ist.utl.pt/~mir/pub/ObstacleAvoidance.pdf [Accessed 5th August 2012]

LIST OF APPENDICES

APPENDIX A: Figures and Tables

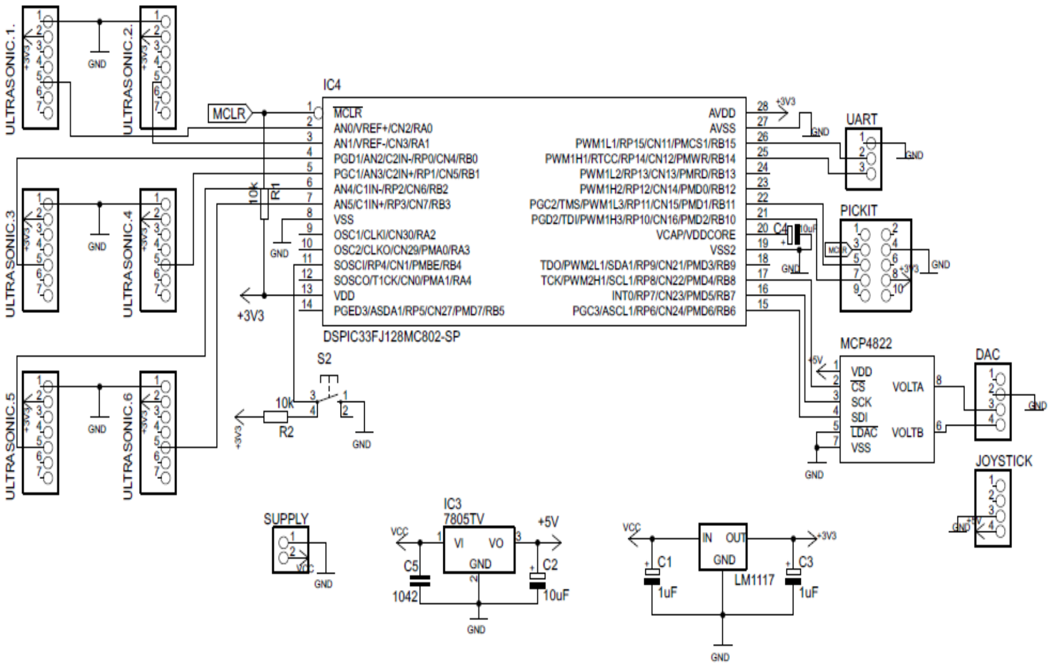

Figure A.1: Circuit Design (Schematic)

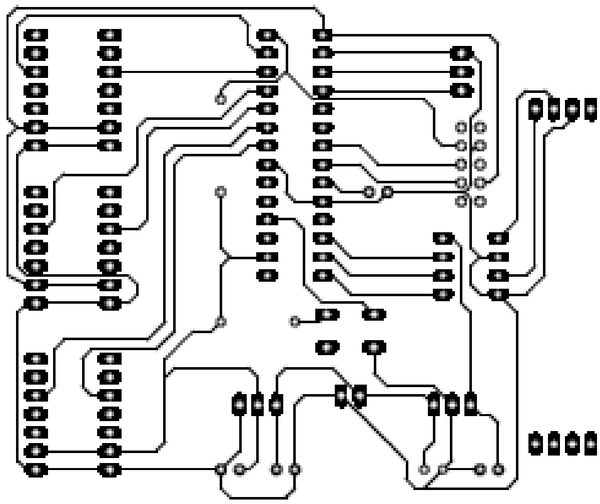

Figure A.2: Circuit Design (Printed Circuit Board)

Figure A.3: LV-MaxSonar-EZ1 Ultrasonic Range Finder

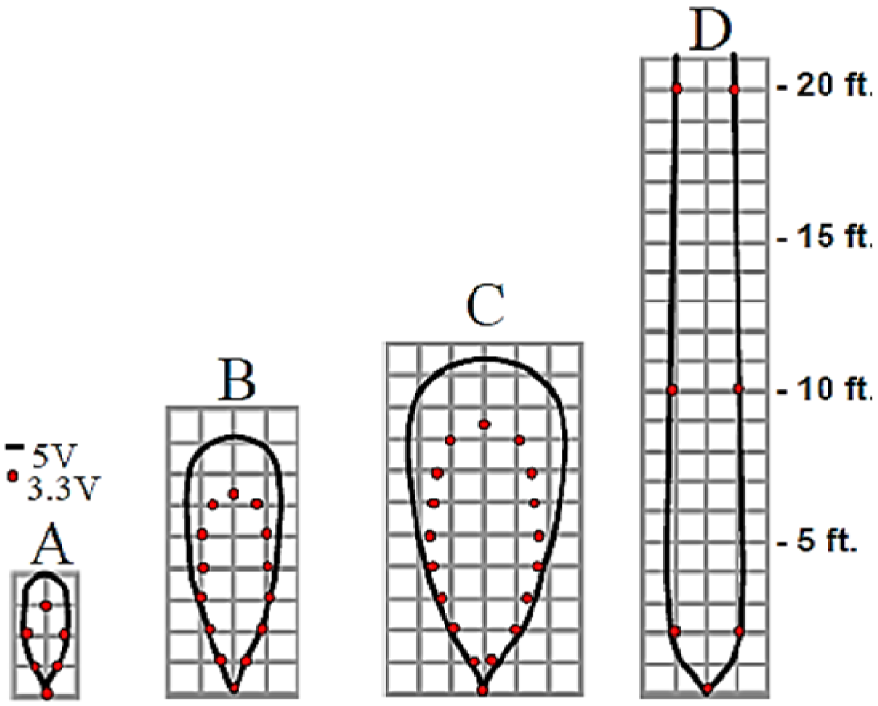

Figure A.4: Ultrasonic Sensor Beam Characteristic

Source: (http://www.maxbotix.com/documents/MB1010_Datasheet.pdf)

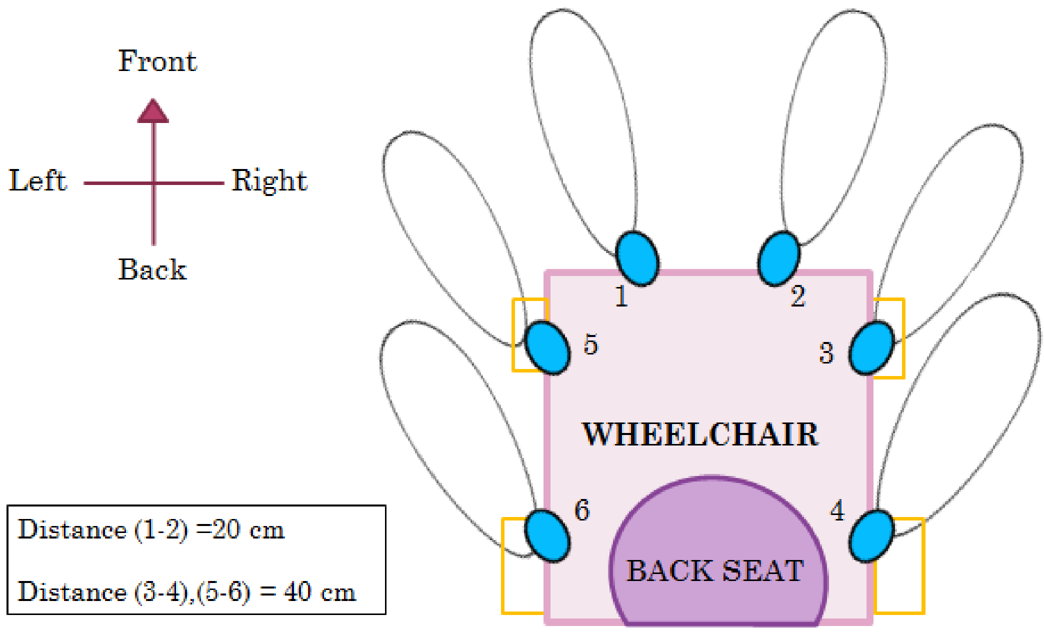

Figure A.5: Position of the Ultrasonic Sensors (Top View)

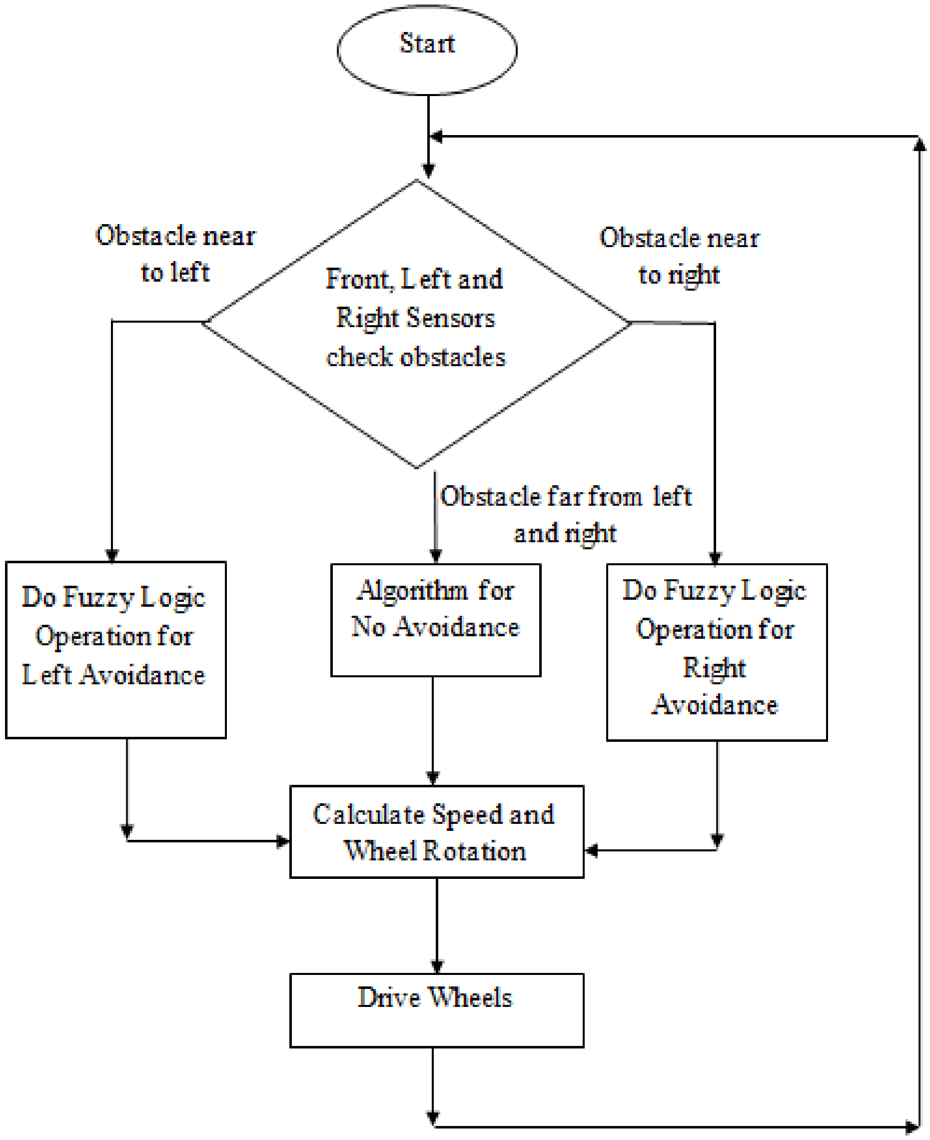

Figure A.6: Flow Chart for Obstacle Avoidance with Fuzzy Logic Approach

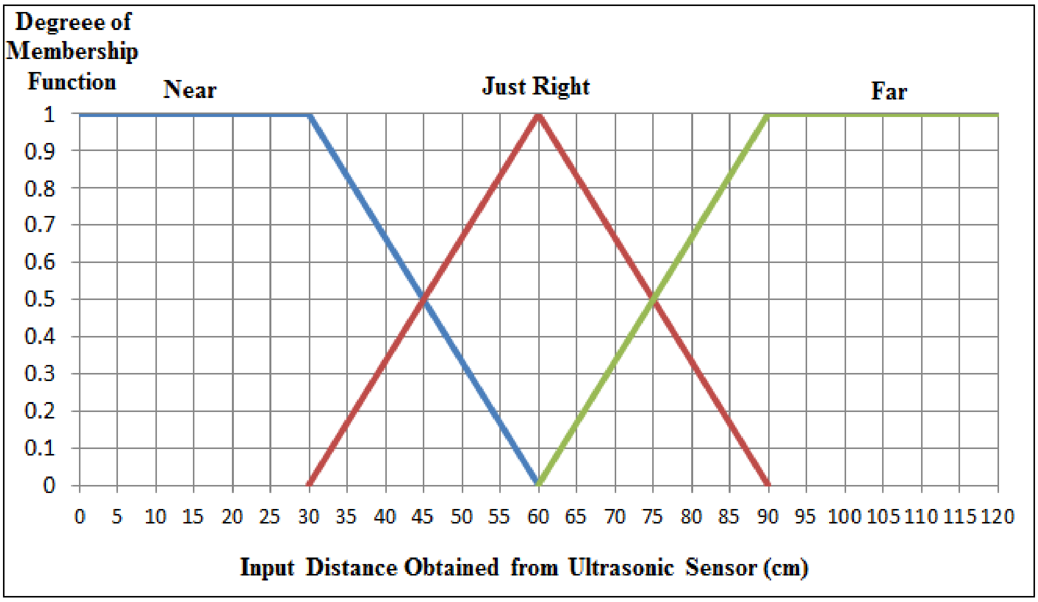

Figure A.7: Input Distance Membership Functions

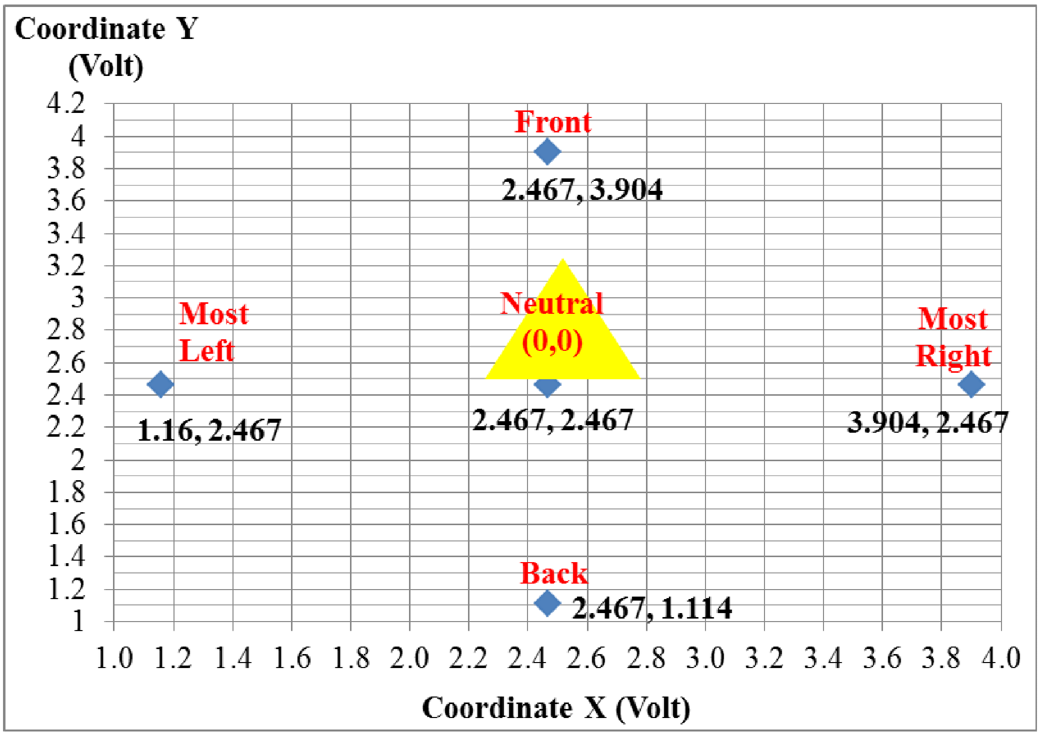

Figure A.8: Joystick Output Voltage at Different Position

Table A.1: Left Obstacle Avoidance Fuzzy Logic Table

Input Distance (Left)	Direction	Acceleration
Near	Turn Right	Slow
Just Right	Go Straight	Fast
Far	Turn Left	Slow

Table A.2 : Right Obstacle Avoidance Fuzzy Logic Table

Input Distance (Right)	Direction	Acceleration
Near	Turn Left	Slow
Just Right	Go Straight	Fast
Far	Turn Right	Slow

APPENDIX B: Right Obstacle Avoidance Data

Table B.1: Right Obstacle Avoidance

Sensor (1) Distance (cm)	Sensor (2) Distance (cm)	Sensor (3) Distance (cm)	Sensor (4) Distance (cm)	Voltage Y (V) Acceleration	Voltage X (V) Direction
186.699	41.912	61.280	141.294	3.076	2.476
186.699	41.912	119.386	137.167	2.650	2.650
185.429	71.123	119.386	144.787	2.650	2.650
191.779	43.500	119.386	137.167	2.650	2.650
180.666	41.277	117.163	63.821	3.038	2.492
181.936	41.595	115.258	53.025	2.999	2.407
181.936	40.959	117.163	55.883	3.042	2.432
181.301	56.200	117.163	154.630	2.650	2.650
186.699	42.230	116.846	58.423	3.080	2.455
181.936	40.642	124.466	55.883	3.042	2.432
180.349	39.689	114.306	56.200	3.047	2.435
180.349	40.642	140.659	56.835	3.056	2.441
180.349	39.689	68.901	57.153	3.061	2.444
154.948	36.832	114.623	58.423	3.080	2.455
173.998	38.419	59.375	58.740	3.085	2.458
177.809	39.372	60.328	76.521	3.090	2.470
174.316	38.419	102.875	57.153	3.061	2.444
173.681	38.102	24.131	166.378	2.650	2.200
171.458	58.423	52.390	165.108	2.989	2.401
143.517	32.069	50.167	72.076	2.956	2.381
168.918	38.102	47.627	131.134	2.918	2.359
169.871	38.102	120.338	74.299	2.882	2.555
169.236	38.102	48.262	71.123	2.927	2.364
175.268	39.372	45.722	163.838	2.889	2.342
167.648	36.832	42.865	42.865	2.846	2.316
201.940	39.689	51.755	39.689	2.798	2.288
163.838	49.532	42.865	39.372	2.793	2.285
165.108	29.211	42.547	37.467	2.764	2.268

165.108	27.941	38.102	37.467	2.764	2.268
571.528	124.466	45.722	37.467	2.764	2.268
85.729	9.525	46.357	37.149	2.760	2.265
535.966	90.809	42.547	37.467	2.764	2.268
91.762	11.113	38.102	38.102	2.774	2.273
96.525	31.434	101.605	42.865	2.846	2.316
108.273	12.701	101.605	37.467	2.764	2.268
96.525	8.890	101.605	36.832	2.755	2.262
123.513	15.876	119.386	37.149	2.760	2.265
134.627	32.704	71.123	37.467	2.764	2.268
94.937	6.350	42.865	37.467	2.764	2.268
150.502	18.098	104.463	39.689	2.798	2.288
90.492	24.131	46.357	37.467	2.764	2.268
87.317	10.160	101.922	53.660	3.009	2.413
168.283	34.927	71.123	42.865	2.846	2.316
180.666	37.467	68.266	43.500	2.855	2.322
187.969	51.120	105.098	39.689	2.798	2.288
193.049	47.945	104.145	38.419	2.779	2.276
198.447	37.467	47.945	40.642	2.812	2.296
210.830	14.606	39.689	27.941	2.650	2.200
89.857	17.781	60.328	0.000	2.650	2.200
226.388	39.372	41.912	42.547	2.831	2.307
241.312	53.025	55.883	66.361	3.042	2.432
246.392	68.901	51.755	67.948	2.980	2.395
243.217	80.966	57.153	69.218	3.061	2.444
245.122	54.295	49.850	92.397	2.951	2.378
238.772	53.025	47.310	96.525	2.913	2.356
238.772	53.025	48.262	86.364	2.927	2.364
238.772	53.025	90.809	56.200	3.047	2.435
236.231	51.755	51.120	58.740	2.970	2.390
236.231	51.755	45.087	124.466	2.879	2.336
259.093	73.346	48.262	64.456	2.927	2.364
231.151	51.120	49.850	64.456	2.951	2.378
233.691	51.755	49.850	51.755	2.951	2.378
231.469	51.120	56.200	61.916	3.047	2.435
263.220	74.934	51.755	61.916	2.980	2.395
238.772	60.645	52.390	61.916	2.989	2.401
241.312	53.025	51.755	59.375	2.980	2.395
236.231	51.755	74.934	49.850	2.951	2.378
383.559	85.412	65.726	55.883	3.042	2.432
248.932	72.711	72.076	60.328	3.090	2.470
396.259	87.317	72.711	52.390	2.989	2.401
226.071	49.532	78.744	60.010	3.095	2.468

607.089	137.167	72.711	54.295	3.018	2.418
563.907	124.784	137.167	49.215	2.942	2.373
403.880	89.539	65.408	51.755	2.980	2.395
343.234	104.780	71.123	69.853	2.948	2.528
241.312	68.266	71.123	57.788	3.071	2.449
335.614	74.934	72.076	68.583	2.967	2.520
261.633	58.423	71.123	57.153	3.061	2.444
322.913	72.076	72.711	54.295	3.018	2.418
327.676	72.394	91.762	75.569	2.863	2.563
223.531	49.532	77.474	59.375	3.095	2.464
327.676	89.857	74.299	59.375	3.095	2.464
314.975	71.123	72.076	58.423	3.080	2.455
337.836	74.934	71.123	72.076	2.929	2.536
335.614	74.934	47.627	58.423	2.918	2.359
330.216	73.346	80.966	55.883	3.042	2.432
314.975	87.634	66.996	57.153	3.061	2.444
312.435	71.123	67.948	57.788	3.071	2.449
370.858	82.872	64.456	57.153	3.061	2.444
322.596	71.758	61.916	57.153	3.061	2.444
330.216	73.029	61.916	48.580	2.932	2.367
306.085	68.266	60.645	17.781	2.650	2.200
269.253	59.375	57.470	71.123	3.066	2.447
180.349	39.689	44.135	63.503	2.865	2.327
241.312	53.660	57.788	73.346	3.071	2.449
302.275	82.554	51.755	75.569	2.980	2.395
238.772	72.076	51.755	74.299	2.980	2.395
317.515	71.123	50.167	76.204	2.956	2.381
292.114	64.456	49.850	74.934	2.951	2.378
314.975	72.076	49.850	137.167	2.951	2.378
315.293	71.123	48.262	61.916	2.927	2.364
292.114	64.456	38.102	69.536	2.774	2.273
248.932	55.883	53.025	72.076	2.999	2.407
284.176	63.503	50.167	77.474	2.956	2.381
312.435	71.123	49.850	72.394	2.951	2.378
293.702	65.408	50.167	157.488	2.956	2.381
294.972	65.726	37.467	99.382	2.764	2.268
292.114	80.966	71.123	74.934	2.929	2.536
312.435	71.123	71.123	76.204	2.929	2.536
304.815	68.266	66.996	74.299	2.991	2.511
305.132	68.583	64.456	80.014	3.028	2.495
302.592	66.996	64.456	53.025	2.999	2.407
302.592	67.631	73.664	68.583	2.967	2.520
302.592	68.266	63.503	69.218	3.043	2.490

284.176	80.014	59.375	65.726	3.095	2.464
551.207	139.707	60.010	65.091	3.095	2.468
488.021	108.273	57.788	61.916	3.071	2.449
614.710	137.167	58.423	60.328	3.080	2.455
505.485	112.083	58.423	59.375	3.080	2.455
649.319	145.740	55.883	49.215	2.942	2.373
256.552	56.200	41.912	68.583	2.831	2.307
505.485	129.546	53.025	58.423	2.999	2.407
426.741	94.937	53.025	54.930	2.999	2.407
530.886	119.386	53.025	55.883	2.999	2.407
640.111	144.787	56.200	55.883	3.042	2.432
657.892	147.645	52.390	48.262	2.927	2.364
558.827	124.784	36.832	63.821	2.755	2.262
543.586	120.973	53.660	55.883	3.009	2.413
627.410	142.247	61.280	57.153	3.061	2.444
657.892	152.407	61.916	54.295	3.018	2.418
650.272	146.375	60.328	76.204	3.090	2.470
665.512	149.867	58.423	57.153	3.061	2.444
586.768	130.816	57.153	58.423	3.061	2.444
652.812	146.375	94.937	71.123	2.929	2.536
657.892	147.327	56.200	73.029	3.047	2.435
563.907	129.546	42.865	64.456	2.846	2.316
657.892	163.838	49.850	64.456	2.951	2.378
586.768	147.645	48.262	61.916	2.927	2.364
523.265	117.163	48.262	60.010	2.927	2.364
602.009	134.627	48.262	59.375	2.927	2.364
533.426	119.386	45.087	57.788	2.879	2.336
586.768	131.769	45.722	58.423	2.889	2.342
599.469	134.627	45.722	47.945	2.889	2.342
213.053	47.945	44.770	22.861	2.650	2.200
236.231	53.025	31.434	56.835	2.673	2.214
617.250	139.707	44.770	57.153	2.875	2.333
268.301	87.634	45.405	57.153	2.884	2.339
283.224	78.744	46.040	56.200	2.894	2.344
293.384	65.726	61.280	57.153	3.061	2.444
320.056	71.758	61.916	59.375	3.095	2.464
347.997	77.474	61.916	56.200	3.047	2.435
252.742	60.645	59.375	142.564	3.095	2.464
383.876	85.729	123.513	58.423	3.080	2.455
396.259	87.634	56.200	60.010	3.047	2.435
496.912	111.765	58.423	58.423	3.080	2.455
408.960	119.386	59.375	58.423	3.080	2.455
354.347	78.744	60.010	63.821	3.095	2.468

584.228	129.546	59.375	59.375	3.095	2.464
454.682	101.605	60.010	61.280	3.095	2.468
618.838	137.167	83.189	57.788	3.071	2.449
525.805	116.846	71.441	57.153	3.061	2.444
609.312	152.407	60.645	55.883	3.042	2.432
497.864	127.006	59.375	54.295	3.018	2.418
561.367	124.784	60.010	51.755	2.980	2.395
170.506	38.102	108.273	51.755	2.980	2.395
355.617	78.744	108.273	51.120	2.970	2.390
208.290	45.722	108.908	49.850	2.951	2.378
222.578	49.532	112.718	51.755	2.980	2.395
247.662	72.711	107.003	50.167	2.956	2.381
267.030	61.916	106.685	54.295	3.018	2.418
279.731	61.916	109.225	51.755	2.980	2.395
320.056	71.441	111.765	53.025	2.999	2.407
465.160	104.145	140.977	54.295	3.018	2.418
393.719	87.317	107.003	61.916	3.066	2.480
393.719	87.634	99.065	66.996	2.991	2.511
330.216	105.098	101.922	66.996	2.991	2.511
343.234	77.474	104.145	63.821	3.038	2.492
340.377	80.014	101.605	63.821	3.038	2.492
383.559	85.094	101.922	61.916	3.066	2.480
353.077	77.474	96.842	76.839	2.844	2.571
350.537	77.474	106.368	53.025	2.999	2.407
446.744	100.017	111.765	66.043	3.005	2.505
378.478	77.474	97.477	61.916	3.066	2.480
445.792	115.258	100.017	61.916	3.066	2.480
368.318	81.919	166.061	61.916	3.066	2.480
309.895	68.583	104.463	61.916	3.066	2.480
347.997	76.839	96.842	63.821	3.038	2.492
439.441	97.477	263.220	60.645	3.085	2.472
248.932	55.883	160.663	61.916	3.066	2.480
320.056	72.711	100.017	64.456	3.028	2.495
360.697	80.014	144.787	104.145	2.650	2.650
306.085	68.266	132.404	107.638	2.650	2.650
320.056	71.441	142.564	109.225	2.650	2.650
323.866	73.029	150.820	111.765	2.650	2.650
424.201	94.302	152.725	105.415	2.650	2.650
627.410	157.488	78.744	105.098	2.815	2.582
553.747	124.466	149.867	104.463	2.650	2.650
388.639	87.317	139.707	105.415	2.650	2.650
307.355	68.583	114.623	101.287	2.650	2.650
302.275	67.631	148.280	101.605	2.650	2.650

386.099	90.492	89.222	101.922	2.659	2.646
475.003	107.003	87.952	101.605	2.678	2.638
607.089	137.167	104.780	101.922	2.650	2.650
647.731	145.105	147.645	94.302	2.650	2.650
464.843	105.098	152.407	94.937	2.650	2.650
383.876	87.952	98.112	100.652	2.650	2.650
287.034	76.204	145.105	99.382	2.650	2.650
419.120	93.667	116.846	101.605	2.650	2.650
403.880	91.127	124.466	101.605	2.650	2.650
343.234	77.474	94.302	100.017	2.650	2.650
342.917	76.839	91.762	94.937	2.650	2.650
408.960	91.762	104.463	96.842	2.650	2.650
340.377	76.204	89.222	80.014	2.796	2.590
408.960	94.302	87.317	100.017	2.687	2.634
408.960	92.397	87.317	99.382	2.687	2.634
405.467	91.762	85.094	101.922	2.721	2.621
406.102	91.762	84.459	93.985	2.730	2.617
343.234	78.744	83.824	114.623	2.740	2.613
405.150	91.444	82.554	99.065	2.758	2.605
403.880	89.857	80.014	99.382	2.796	2.590
405.150	91.444	79.379	99.382	2.806	2.586
403.880	91.127	77.474	208.608	2.834	2.574
403.880	91.127	77.474	102.875	2.834	2.574
401.339	90.492	82.554	105.415	2.758	2.605
403.880	110.178	74.299	129.546	2.882	2.555
445.792	99.382	74.934	99.382	2.872	2.559
396.259	88.587	75.251	99.382	2.867	2.561
403.880	90.174	73.029	96.842	2.901	2.547
401.339	89.857	91.444	113.353	2.650	2.650
401.339	89.539	87.317	96.525	2.687	2.634
398.799	89.539	71.123	127.006	2.929	2.536
167.966	42.547	69.218	124.466	2.957	2.524
172.411	38.737	66.996	99.065	2.991	2.511
198.130	45.722	66.996	246.392	2.991	2.511
215.910	48.897	43.817	97.477	2.860	2.324
236.231	78.744	65.091	122.878	3.019	2.499
246.709	55.883	64.456	93.985	3.028	2.495
271.793	60.645	64.456	116.846	3.028	2.495
280.366	61.916	58.423	91.127	3.080	2.455
307.355	68.266	61.916	89.539	3.066	2.480
309.895	83.824	61.916	83.824	3.066	2.480
231.151	51.120	67.948	86.364	2.976	2.517
368.318	80.966	61.916	87.317	3.066	2.480

378.478	83.824	61.916	77.474	3.066	2.480
403.880	89.857	74.299	104.780	2.882	2.555
332.756	73.346	57.153	82.236	3.061	2.444
401.339	89.222	78.744	82.236	2.815	2.582
383.559	99.382	79.379	80.014	2.806	2.586
356.887	80.014	80.966	80.014	2.796	2.590
381.018	85.094	79.379	80.014	2.806	2.586
393.719	87.952	77.474	71.441	2.924	2.538
391.179	87.317	74.299	73.664	2.891	2.551
386.099	86.364	72.394	74.934	2.910	2.544
383.559	100.652	77.474	72.394	2.910	2.544
274.333	61.280	74.299	72.076	2.915	2.542
200.987	45.087	77.474	72.076	2.915	2.542
373.398	76.204	74.934	56.200	3.047	2.435
381.018	84.459	60.328	85.729	3.090	2.470
381.018	85.094	78.744	71.123	2.929	2.536
381.018	99.065	71.123	71.123	2.929	2.536
347.997	93.667	71.123	67.948	2.976	2.517
115.258	27.941	68.583	66.996	2.991	2.511
118.433	32.704	68.266	66.678	2.995	2.509
129.546	30.481	39.372	78.744	2.793	2.285
137.167	30.799	63.821	64.456	3.038	2.492
139.707	48.262	64.456	64.456	3.028	2.495
149.867	33.974	72.076	61.916	3.066	2.480
157.488	35.244	61.916	66.361	3.066	2.480
161.298	36.197	78.744	72.076	2.915	2.542
167.966	37.467	71.441	61.280	3.076	2.476
180.349	54.295	59.375	59.375	3.095	2.464
187.969	41.595	59.375	59.375	3.095	2.464
188.287	41.912	59.375	55.883	3.042	2.432
175.268	38.737	152.725	55.883	3.042	2.432
187.969	41.912	58.423	57.153	3.061	2.444
221.943	62.551	84.459	57.470	3.066	2.447
173.998	38.102	87.317	54.295	3.018	2.418
181.301	39.372	84.459	53.025	2.999	2.407
183.206	39.372	65.408	57.153	3.061	2.444
181.619	39.372	5.080	55.883	2.650	2.200
180.666	53.025	57.153	54.295	3.018	2.418
193.049	42.547	64.456	59.375	3.095	2.464
175.268	38.102	76.204	56.200	3.047	2.435
281.954	61.916	80.649	73.981	2.886	2.553
281.954	61.280	72.076	71.123	2.929	2.536
279.414	76.204	81.919	71.123	2.929	2.536

264.173	74.299	82.236	78.744	2.815	2.582
271.793	61.916	80.966	60.645	3.085	2.472
289.574	63.821	80.966	68.583	2.967	2.520
274.333	60.328	81.919	65.726	3.010	2.503
271.793	61.280	80.966	80.966	2.782	2.596
322.913	72.076	81.919	63.821	3.038	2.492
271.793	61.916	80.649	64.456	3.028	2.495
271.793	62.551	81.919	64.456	3.028	2.495
322.596	72.076	80.014	64.456	3.028	2.495
297.194	66.996	79.379	64.456	3.028	2.495
303.545	68.901	80.014	64.456	3.028	2.495
332.756	74.934	85.094	74.299	2.882	2.555
322.596	73.346	96.525	65.091	3.019	2.499
342.917	77.474	79.379	64.456	3.028	2.495
332.756	75.569	79.379	63.821	3.038	2.492
332.756	75.251	78.744	64.456	3.028	2.495
332.756	75.569	79.379	64.456	3.028	2.495
322.913	73.664	79.379	65.091	3.019	2.499
323.866	73.664	79.379	64.456	3.028	2.495
324.183	73.664	80.014	64.456	3.028	2.495
322.596	73.346	79.379	64.456	3.028	2.495
330.216	75.569	81.919	65.091	3.019	2.499
330.216	74.934	78.744	64.456	3.028	2.495
322.913	73.664	79.379	64.456	3.028	2.495
320.373	73.346	78.744	64.456	3.028	2.495
274.333	61.916	80.014	65.091	3.019	2.499
322.596	73.664	78.744	64.456	3.028	2.495
322.913	73.346	80.014	64.456	3.028	2.495
322.596	73.346	79.061	63.821	3.038	2.492
322.596	73.346	78.744	64.456	3.028	2.495
322.913	73.664	80.014	64.456	3.028	2.495
323.866	73.664	79.379	64.456	3.028	2.495
322.596	73.664	79.379	64.456	3.028	2.495
324.818	74.299	80.014	65.091	3.019	2.499
264.173	60.645	80.014	64.456	3.028	2.495
332.756	75.251	79.379	64.456	3.028	2.495
330.216	75.569	80.014	64.456	3.028	2.495
332.756	75.251	79.379	64.456	3.028	2.495
332.756	74.934	80.014	64.456	3.028	2.495
273.063	61.916	79.379	64.456	3.028	2.495
251.472	57.153	79.379	64.456	3.028	2.495
272.111	61.916	79.379	64.456	3.028	2.495
330.216	74.934	80.014	64.456	3.028	2.495

222.578	51.120	79.379	71.123	2.929	2.536
309.895	71.123	79.379	64.456	3.028	2.495
259.093	58.740	79.379	64.456	3.028	2.495
299.735	66.996	79.379	64.456	3.028	2.495
332.756	75.569	79.379	64.456	3.028	2.495
320.373	73.346	79.379	64.456	3.028	2.495
332.756	76.839	80.014	68.583	2.967	2.520
332.756	75.569	80.014	65.091	3.019	2.499
335.614	76.204	80.014	64.456	3.028	2.495
274.333	61.916	78.744	64.456	3.028	2.495
330.216	74.934	80.014	64.456	3.028	2.495
323.866	73.664	79.379	65.091	3.019	2.499
335.296	76.204	79.379	63.821	3.038	2.492
324.183	73.981	79.379	63.821	3.038	2.492
322.913	73.664	79.379	64.456	3.028	2.495
323.866	73.664	79.379	64.456	3.028	2.495
322.596	73.346	82.872	64.456	3.028	2.495
322.913	73.664	79.379	64.456	3.028	2.495
322.913	73.664	80.014	65.091	3.019	2.499
324.183	73.346	80.014	64.456	3.028	2.495
325.136	73.664	79.379	64.456	3.028	2.495
322.913	73.664	79.379	64.456	3.028	2.495
322.913	73.664	80.014	64.456	3.028	2.495
302.275	68.583	79.379	63.821	3.038	2.492
323.866	73.664	81.919	64.456	3.028	2.495
323.866	73.664	80.014	64.456	3.028	2.495
322.596	73.029	82.554	64.456	3.028	2.495
274.333	61.916	62.551	63.821	3.057	2.484
324.183	73.664	71.123	65.091	3.019	2.499
325.453	73.664	51.755	64.456	2.980	2.395
324.183	73.664	57.788	42.865	2.846	2.316
325.453	73.664	73.029	39.372	2.793	2.285
80.014	20.639	79.696	40.007	2.803	2.290
87.317	22.861	79.379	42.547	2.841	2.313
94.302	24.131	82.554	39.372	2.793	2.285
322.596	71.123	79.379	39.689	2.798	2.288
113.988	27.941	74.934	42.230	2.836	2.310
119.386	29.211	81.919	39.372	2.793	2.285
127.006	30.481	61.916	39.689	2.798	2.288
137.167	32.704	79.379	39.372	2.793	2.285
320.056	72.711	83.189	39.372	2.793	2.285
153.360	36.197	80.014	39.689	2.798	2.288
157.488	36.197	78.744	39.372	2.793	2.285

309.895	71.123	84.459	42.230	2.836	2.310
175.268	40.642	81.919	38.102	2.774	2.273
183.206	42.230	80.014	39.372	2.793	2.285
191.779	42.865	79.379	41.277	2.822	2.302
200.987	45.722	80.014	39.689	2.798	2.288
205.750	47.310	83.824	45.722	2.889	2.342
215.910	50.167	81.919	39.689	2.798	2.288
220.991	50.167	82.236	39.372	2.793	2.285
231.151	52.390	80.014	37.467	2.764	2.268
241.312	54.295	78.744	39.689	2.798	2.288
243.534	55.883	80.014	39.689	2.798	2.288
254.012	57.788	79.379	40.642	2.812	2.296
263.220	59.375	80.014	39.372	2.793	2.285
272.111	61.280	80.014	39.372	2.793	2.285
274.333	60.010	79.379	39.372	2.793	2.285
287.034	63.186	80.014	39.689	2.798	2.288
293.384	64.456	82.554	39.372	2.793	2.285
226.071	52.390	79.379	39.372	2.793	2.285
242.582	55.883	80.014	54.295	3.018	2.418
317.515	72.076	71.123	32.069	2.683	2.219
226.388	51.755	72.711	30.481	2.659	2.205
345.457	80.014	77.474	37.467	2.764	2.268
330.216	73.664	78.744	39.372	2.793	2.285
330.216	75.569	77.474	37.784	2.769	2.271
332.756	73.664	87.317	58.423	3.080	2.455
299.735	66.361	109.225	66.361	3.000	2.507
309.895	68.901	99.065	41.277	2.822	2.302
274.333	61.916	80.331	41.277	2.822	2.302
323.866	73.346	71.123	30.481	2.659	2.205
271.793	61.916	71.441	30.799	2.664	2.208
269.253	61.280	71.123	30.799	2.664	2.208
312.435	72.076	120.338	69.853	2.948	2.528
355.617	80.649	101.605	65.408	3.014	2.501

APPENDIX C: Left and Right Obstacle Avoidance Data

Table C.2: Left and Right Obstacle Avoidance

Relative Distance between Wheelchair and Obstacle (cm)								Voltage Y (V) Acceleration		Voltage X (V) Direction	
A	93.67	B	152.725	G	57.470	H	27.941	I	3.066	J	2.480
A	111.77	B	106.685	C	153.360	D	80.014	E	2.796	F	2.590
A	94.94	B	119.386	G	59.375	H	27.941	I	3.095	J	2.468
A	85.41	B	130.816	G	60.645	H	30.799	I	3.085	J	2.461
A	85.41	B	149.867	G	50.485	H	20.321	I	2.961	J	2.523
A	74.30	B	166.061	G	63.186	H	27.941	I	3.047	J	2.447
A	82.87	B	169.236	G	61.916	H	29.529	I	3.066	J	2.454
A	75.57	B	204.797	G	64.456	H	27.306	I	3.028	J	2.440
A	72.08	B	211.148	G	63.186	H	32.704	I	3.047	J	2.447
A	74.30	B	226.388	G	63.186	H	32.704	I	3.047	J	2.447
A	85.41	B	246.392	G	61.280	H	29.211	I	3.076	J	2.458
A	85.73	B	259.093	G	61.916	H	29.211	I	3.066	J	2.454
A	83.82	B	259.093	G	62.551	H	26.671	I	3.057	J	2.451
A	79.70	B	218.451	G	62.551	H	30.481	I	3.057	J	2.451
A	86.68	B	269.571	G	64.456	H	30.481	I	3.028	J	2.440
A	85.09	B	269.253	G	64.456	H	30.481	I	3.028	J	2.440
A	85.41	B	266.713	G	65.091	H	30.481	I	3.019	J	2.437
A	80.97	B	139.707	G	65.091	H	30.481	I	3.019	J	2.437
A	90.49	B	293.384	G	71.123	H	34.292	I	2.929	J	2.403
A	107.64	B	267.030	G	68.266	H	31.434	I	2.972	J	2.419
A	101.29	B	220.991	G	71.441	H	32.704	I	2.924	J	2.401
A	104.46	B	215.910	G	63.821	H	30.481	I	3.038	J	2.444

A	101.92	B	286.081	G	69.853	H	30.481	I	2.948	J	2.410
A	96.52	B	247.662	G	71.441	H	36.832	I	2.924	J	2.401
A	106.69	B	253.060	G	66.996	H	31.434	I	2.991	J	2.426
A	80.33	B	215.910	G	67.631	H	30.799	I	2.981	J	2.423
A	83.82	B	144.787	G	71.123	H	32.069	I	2.929	J	2.403
A	107.64	B	143.517	G	59.375	H	27.941	I	3.095	J	2.468
A	109.23	B	129.546	G	71.441	H	32.704	I	2.924	J	2.401
A	105.10	B	137.167	G	71.123	H	26.036	I	2.929	J	2.403
A	98.75	B	131.134	G	71.123	H	30.481	I	2.929	J	2.403
A	105.73	B	125.101	G	62.551	H	30.799	I	3.057	J	2.451
A	99.38	B	140.977	G	71.123	H	32.704	I	2.929	J	2.403
A	114.31	B	116.846	G	112.083	H	46.992	I	2.650	J	2.300
A	74.30	B	115.258	G	65.091	H	27.941	I	3.019	J	2.437
A	85.41	B	154.948	G	75.569	H	34.927	I	2.863	J	2.379
A	85.73	B	187.969	G	69.536	H	25.084	I	2.953	J	2.412
A	86.68	B	159.075	G	87.317	H	38.102	I	2.687	J	2.314
A	87.32	B	139.707	G	58.105	H	29.211	I	3.076	J	2.476
A	73.35	B	85.729	G	55.883	H	25.719	I	3.042	J	2.490
A	104.46	B	142.564	G	87.317	H	35.244	I	2.687	J	2.314
A	80.01	B	113.353	G	58.740	H	25.719	I	3.085	J	2.472
A	83.82	B	115.258	G	84.459	H	37.149	I	2.730	J	2.329
A	111.77	B	121.608	G	115.258	H	40.324	I	2.650	J	2.300
A	105.42	B	119.386	G	44.135	H	24.131	I	2.865	J	2.562
A	74.30	B	120.338	G	60.010	H	23.179	I	3.095	J	2.465
A	76.84	B	89.857	G	113.353	H	46.357	I	2.650	J	2.300
A	86.05	B	171.458	G	66.361	H	27.941	I	3.000	J	2.430
A	55.88	B	160.028	G	66.043	H	27.941	I	3.005	J	2.431
A	78.74	B	129.546	G	87.317	H	38.102	I	2.687	J	2.314
A	87.32	B	114.623	G	112.718	H	39.372	I	2.650	J	2.300
A	67.00	B	117.798	G	53.660	H	11.113	I	3.009	J	2.503
A	74.93	B	101.605	G	113.353	H	41.277	I	2.650	J	2.300
A	87.32	B	105.733	G	120.973	H	46.040	I	5.535	J	5.535

A	80.65	B	104.145	G	80.966	H	35.562	I	2.782	J	2.349
A	47.94	B	111.765	G	66.361	H	11.748	I	3.000	J	2.430
A	32.70	B	96.842	G	60.963	H	17.781	I	3.081	J	2.459
A	11.11	B	91.762	G	61.280	H	0.000	I	3.076	J	2.458
A	72.71	B	106.368	G	108.590	H	42.865	I	2.650	J	2.300
A	73.66	B	80.014	G	75.569	H	15.241	I	2.863	J	2.379
A	52.39	B	58.740	G	113.988	H	38.737	I	2.650	J	2.300
A	89.86	B	104.463	G	61.280	H	0.000	I	3.076	J	2.458
A	0.00	B	104.145	G	45.087	H	15.558	I	2.879	J	2.556
A	39.37	B	101.922	G	115.258	H	40.959	I	2.650	J	2.300
A	51.44	B	102.557	G	109.225	H	41.595	I	2.650	J	2.300
A	80.97	B	166.696	G	91.762	H	34.927	I	2.650	J	2.300
A	74.93	B	71.123	C	45.405	D	152.725	E	2.884	F	2.339
A	119.70	B	99.065	C	64.456	D	152.407	E	3.028	F	2.495
A	54.30	B	100.017	G	119.386	H	87.317	I	2.650	J	2.300
A	100.65	B	96.525	C	35.244	D	165.108	E	2.731	F	2.248
A	124.78	B	100.017	C	94.302	D	193.049	E	2.650	F	2.650
A	139.71	B	97.477	C	101.605	D	152.725	E	2.650	F	2.650
A	130.82	B	96.842	C	113.988	D	182.889	E	2.650	F	2.650
A	46.36	B	99.065	G	119.703	H	36.832	I	2.650	J	2.300
A	55.88	B	112.083	G	119.703	H	44.135	I	2.650	J	2.300
A	123.51	B	94.302	C	177.809	D	112.083	E	2.650	F	2.650
A	122.24	B	76.204	C	117.163	D	215.910	E	2.650	F	2.650
A	116.85	B	89.857	C	117.798	D	212.418	E	2.650	F	2.650
A	137.48	B	92.397	C	117.163	D	233.691	E	2.650	F	2.650
A	139.71	B	96.525	C	114.623	D	259.093	E	2.650	F	2.650
A	20.64	B	96.525	G	54.295	H	0.000	I	3.018	J	2.500
A	0.00	B	96.525	G	51.120	H	0.000	I	2.970	J	2.519
A	0.00	B	96.842	G	51.755	H	0.000	I	2.980	J	2.515
A	0.00	B	96.525	G	59.375	H	0.000	I	3.095	J	2.468
A	0.00	B	97.477	G	49.850	H	19.368	I	2.951	J	2.527
A	0.00	B	96.842	G	50.485	H	15.876	I	2.961	J	2.523

A	0.00	B	91.127	G	49.532	H	0.000	I	2.946	J	2.529
A	0.00	B	96.842	G	53.025	H	0.000	I	2.999	J	2.507
A	0.00	B	93.032	G	84.459	H	32.387	I	2.730	J	2.329
A	61.28	B	94.937	G	49.215	H	21.274	I	2.942	J	2.531
A	64.46	B	92.397	G	49.850	H	23.179	I	2.951	J	2.527
A	67.95	B	92.397	G	50.485	H	21.274	I	2.961	J	2.523
A	49.53	B	89.857	G	51.120	H	6.350	I	2.970	J	2.519
A	24.77	B	89.857	G	49.215	H	71.123	I	2.942	J	2.531
A	54.30	B	89.857	G	25.719	H	9.843	I	2.650	J	2.650
A	34.93	B	89.857	G	66.996	H	0.000	I	2.991	J	2.426
A	0.00	B	91.127	G	45.722	H	44.452	I	2.889	J	2.552
A	37.47	B	89.222	G	113.988	H	83.824	I	2.650	J	2.300
A	0.00	B	86.364	G	43.500	H	0.000	I	2.855	J	2.566
A	0.00	B	86.682	G	56.518	H	0.000	I	3.052	J	2.486
A	0.00	B	87.952	G	56.200	H	0.000	I	3.047	J	2.488
A	13.02	B	89.857	G	43.182	H	0.000	I	2.851	J	2.568
A	0.00	B	82.554	G	28.259	H	0.000	I	2.650	J	2.650
A	0.00	B	82.872	G	46.040	H	34.292	I	2.894	J	2.550
A	0.00	B	82.872	G	46.675	H	0.000	I	2.903	J	2.546
A	0.00	B	80.649	G	46.357	H	0.000	I	2.898	J	2.548
A	0.00	B	80.014	G	46.675	H	0.000	I	2.903	J	2.546
A	0.00	B	80.649	G	20.321	H	0.000	I	2.650	J	2.650
A	0.32	B	80.014	G	71.123	H	0.000	I	2.929	J	2.403
A	0.00	B	77.474	G	55.883	H	0.000	I	3.042	J	2.490
A	0.00	B	83.189	G	53.025	H	15.241	I	2.999	J	2.507
A	50.48	B	77.474	G	86.682	H	34.292	I	2.697	J	2.317
A	67.00	B	76.204	G	85.094	H	21.274	I	2.721	J	2.326
A	42.23	B	72.711	G	85.094	H	30.799	I	2.721	J	2.326
A	61.92	B	73.664	G	85.729	H	88.587	I	2.711	J	2.322
A	46.04	B	72.711	G	85.094	H	28.259	I	2.721	J	2.326
A	65.73	B	71.123	G	94.302	H	34.927	I	2.650	J	2.300
A	63.19	B	71.123	G	82.554	H	32.387	I	2.758	J	2.340

A	64.46	B	85.729	G	87.317	H	30.481	I	2.687	J	2.314
A	56.20	B	59.375	G	85.094	H	30.481	I	2.721	J	2.326
A	83.82	B	63.186	C	85.094	D	235.279	E	2.721	F	2.621
A	124.78	B	111.765	C	80.649	D	131.134	E	2.787	F	2.594
A	83.82	B	68.266	C	86.999	D	127.006	E	2.692	F	2.632
A	1.91	B	71.123	G	117.798	H	0.000	I	2.650	J	2.300
A	0.00	B	71.123	G	87.317	H	0.000	I	2.687	J	2.314
A	0.00	B	71.123	G	87.317	H	0.000	I	2.687	J	2.314
A	3.81	B	71.123	G	87.317	H	26.036	I	2.687	J	2.314
A	55.88	B	71.123	G	87.317	H	18.733	I	2.687	J	2.314
A	45.09	B	71.123	G	87.317	H	5.398	I	2.687	J	2.314
A	30.80	B	66.996	G	87.317	H	26.671	I	2.687	J	2.314
A	67.00	B	71.123	G	87.317	H	38.102	I	2.687	J	2.314
A	85.73	B	66.996	C	84.459	D	124.784	E	2.730	F	2.617
A	133.36	B	68.901	C	84.142	D	124.784	E	2.735	F	2.615
A	40.32	B	107.955	G	85.094	H	0.000	I	2.721	J	2.326
A	0.00	B	66.996	G	87.317	H	0.000	I	2.687	J	2.314
A	0.00	B	66.996	G	87.317	H	0.000	I	2.687	J	2.314
A	0.00	B	68.266	G	87.317	H	9.208	I	2.687	J	2.314
A	34.61	B	66.996	G	87.317	H	24.766	I	2.687	J	2.314
A	38.74	B	66.996	G	87.952	H	27.941	I	2.678	J	2.310
A	38.10	B	67.631	G	87.317	H	35.562	I	2.687	J	2.314
A	77.47	B	66.996	C	82.872	D	119.703	E	2.754	F	2.607
A	137.48	B	82.236	C	33.974	D	120.656	E	2.712	F	2.236
A	76.84	B	66.996	C	42.230	D	120.338	E	2.836	F	2.310
A	88.59	B	64.456	C	64.456	D	112.083	E	3.028	F	2.495
A	109.23	B	64.456	C	80.966	D	228.611	E	2.782	F	2.596
A	32.07	B	64.456	G	87.317	H	0.000	I	2.687	J	2.314
A	26.67	B	66.996	G	85.094	H	24.131	I	2.721	J	2.326
A	49.85	B	66.996	G	85.729	H	32.704	I	2.711	J	2.322
A	83.51	B	206.703	G	76.204	H	24.131	I	2.853	J	2.375
A	53.66	B	457.540	G	85.729	H	30.481	I	2.711	J	2.322

A	57.15	B	66.361	G	85.094	H	30.799	I	2.721	J	2.326
A	65.73	B	69.536	G	87.317	H	26.671	I	2.687	J	2.314
A	48.90	B	63.186	G	82.236	H	48.262	I	2.763	J	2.342
A	68.90	B	63.821	C	58.105	D	47.945	E	2.922	F	2.361
A	91.44	B	64.456	C	75.251	D	49.215	E	2.942	F	2.373
A	61.92	B	63.821	G	85.094	H	8.255	I	2.721	J	2.326
A	11.11	B	63.821	G	82.236	H	7.620	I	2.763	J	2.342
A	44.13	B	64.456	G	87.317	H	87.317	I	2.687	J	2.314
A	66.36	B	62.551	C	48.262	D	94.302	E	2.927	F	2.364
A	101.60	B	64.456	C	72.711	D	117.163	E	2.905	F	2.545
A	86.05	B	64.456	C	72.711	D	109.543	E	2.905	F	2.545
A	82.55	B	59.375	C	72.076	D	107.003	E	2.915	F	2.542
A	127.01	B	77.791	C	72.711	D	107.003	E	2.905	F	2.545
A	50.48	B	107.955	G	88.587	H	11.113	I	2.668	J	2.307
A	18.10	B	55.883	G	82.236	H	0.953	I	2.763	J	2.342
A	0.00	B	66.361	G	74.616	H	15.558	I	2.877	J	2.384
A	41.91	B	39.689	C	71.123	D	105.733	E	2.929	F	2.536
A	105.10	B	66.361	C	68.901	D	105.415	E	2.962	F	2.522
A	71.44	B	71.123	C	66.996	D	104.463	E	2.991	F	2.511
A	88.59	B	64.456	C	71.123	D	87.317	E	2.929	F	2.536
A	94.30	B	71.123	C	66.996	D	101.922	E	2.991	F	2.511
A	106.37	B	66.043	C	66.996	D	101.922	E	2.991	F	2.511
A	93.67	B	63.821	C	68.266	D	99.382	E	2.972	F	2.519
A	131.45	B	61.916	C	66.996	D	100.017	E	2.991	F	2.511
A	109.23	B	72.711	C	64.456	D	99.065	E	3.028	F	2.495
A	111.77	B	58.740	C	53.343	D	99.382	E	3.004	F	2.410
A	111.77	B	60.010	C	74.616	D	102.875	E	2.877	F	2.557
A	111.77	B	60.010	C	59.375	D	101.605	E	3.095	F	2.464
A	111.77	B	57.788	C	59.375	D	93.032	E	3.095	F	2.464
A	106.69	B	67.631	C	57.470	D	92.397	E	3.066	F	2.447
A	94.30	B	69.536	C	58.105	D	86.999	E	3.076	F	2.452
A	105.73	B	55.883	C	72.076	D	92.397	E	2.915	F	2.542

A	105.10	B	47.310	C	27.941	D	105.733	E	2.650	F	2.200
A	97.48	B	51.755	C	53.660	D	61.916	E	3.009	F	2.413
A	97.48	B	49.850	C	51.120	D	85.094	E	2.970	F	2.390
A	106.69	B	50.167	C	50.167	D	82.236	E	2.956	F	2.381
A	124.47	B	48.262	C	51.755	D	82.872	E	2.980	F	2.395
A	116.85	B	63.503	C	44.135	D	77.474	E	2.865	F	2.327
A	105.10	B	65.091	C	45.087	D	77.474	E	2.879	F	2.336
A	127.01	B	62.551	C	25.401	D	71.123	E	2.650	F	2.200
A	105.10	B	54.295	C	42.547	D	96.525	E	2.841	F	2.313
A	139.71	B	305.132	G	93.985	H	42.547	I	2.650	J	2.300
A	91.76	B	199.717	G	100.652	H	40.007	I	2.650	J	2.300
A	89.22	B	431.821	G	100.652	H	41.595	I	2.650	J	2.300
A	82.55	B	249.250	G	96.842	H	40.959	I	2.650	J	2.300
A	89.22	B	251.472	G	97.477	H	42.230	I	2.650	J	2.300
A	83.82	B	54.295	C	39.689	D	69.218	E	2.798	F	2.288
A	101.60	B	53.025	C	39.689	D	35.562	E	2.736	F	2.251
A	93.03	B	53.025	C	39.689	D	66.996	E	2.798	F	2.288
A	94.94	B	267.983	G	96.842	H	41.912	I	2.650	J	2.300
A	82.24	B	51.755	C	39.372	D	66.996	E	2.793	F	2.285
A	100.02	B	54.295	C	42.547	D	66.996	E	2.841	F	2.313
A	127.01	B	167.648	G	96.525	H	31.434	I	2.650	J	2.300
A	112.08	B	310.213	G	94.937	H	36.514	I	2.650	J	2.300
A	101.60	B	233.691	G	96.525	H	36.197	I	2.650	J	2.300
A	87.32	B	320.373	G	96.525	H	42.865	I	2.650	J	2.300
A	80.97	B	274.333	G	93.667	H	43.500	I	2.650	J	2.300
A	69.22	B	266.713	G	92.397	H	36.197	I	2.650	J	2.300
A	72.08	B	209.878	G	89.857	H	35.562	I	2.650	J	2.300
A	69.22	B	220.991	G	89.857	H	35.562	I	2.650	J	2.300
A	68.58	B	293.702	G	89.857	H	34.927	I	2.650	J	2.300
A	77.47	B	215.910	G	89.222	H	38.737	I	2.659	J	2.303
A	90.49	B	363.238	G	89.857	H	38.102	I	2.650	J	2.300
A	78.74	B	368.635	G	89.857	H	32.704	I	2.650	J	2.300

A	80.01	B	304.815	G	89.857	H	34.609	I	2.650	J	2.300
A	86.36	B	374.668	G	89.857	H	36.832	I	2.650	J	2.300
A	87.32	B	383.559	G	89.857	H	33.974	I	2.650	J	2.300
A	80.97	B	383.876	G	87.317	H	37.149	I	2.687	J	2.314
A	84.14	B	317.515	G	87.317	H	30.481	I	2.687	J	2.314
A	78.74	B	287.351	G	87.317	H	37.467	I	2.687	J	2.314
A	83.19	B	396.259	G	87.317	H	35.244	I	2.687	J	2.314
A	58.74	B	86.364	G	87.317	H	32.704	I	2.687	J	2.314
A	79.38	B	93.985	G	87.952	H	28.576	I	2.678	J	2.310
A	79.70	B	101.605	G	84.459	H	34.927	I	2.730	J	2.329
A	61.92	B	80.966	G	82.872	H	23.496	I	2.754	J	2.338
A	58.42	B	137.167	G	84.777	H	24.766	I	2.725	J	2.328
A	40.32	B	175.586	G	87.952	H	23.179	I	2.678	J	2.310
A	25.08	B	195.589	G	89.222	H	16.828	I	2.659	J	2.303
A	28.26	B	148.597	G	80.649	H	31.434	I	2.787	J	2.350
A	54.30	B	180.349	G	75.569	H	22.861	I	2.863	J	2.379
A	30.48	B	198.447	G	86.047	H	16.511	I	2.706	J	2.321
A	41.59	B	178.761	G	78.744	H	21.274	I	2.815	J	2.361
A	27.94	B	203.845	G	76.204	H	23.179	I	2.853	J	2.375
A	47.94	B	246.392	G	91.127	H	27.941	I	2.650	J	2.300
A	51.44	B	253.060	G	78.744	H	20.003	I	2.815	J	2.361
A	41.59	B	264.490	G	97.477	H	37.784	I	2.650	J	2.300
A	42.86	B	280.366	G	96.842	H	9.525	I	2.650	J	2.300
A	2.54	B	293.702	G	71.123	H	9.525	I	2.929	J	2.403
A	42.86	B	322.596	G	89.222	H	0.000	I	2.659	J	2.303
A	0.00	B	320.056	G	79.696	H	16.828	I	2.801	J	2.356
A	15.24	B	228.611	G	79.696	H	20.321	I	2.801	J	2.356
A	0.00	B	348.314	G	78.744	H	19.368	I	2.815	J	2.361
A	28.26	B	350.537	G	77.156	H	0.000	I	2.839	J	2.370
A	0.00	B	358.475	G	76.521	H	0.000	I	2.848	J	2.373
A	25.08	B	421.660	G	77.474	H	51.755	I	2.834	J	2.368
A	0.00	B	434.361	G	77.474	H	0.000	I	2.834	J	2.368

A	2.54	B	444.839	G	77.474	H	27.941	I	2.834	J	2.368
A	57.79	B	455.952	G	74.934	H	32.069	I	2.872	J	2.382
A	60.01	B	259.093	G	75.251	H	33.974	I	2.867	J	2.380
A	96.52	B	480.083	G	74.934	H	33.974	I	2.872	J	2.382
A	88.59	B	492.784	G	75.569	H	25.719	I	2.863	J	2.379
A	86.05	B	497.864	G	74.934	H	34.609	I	2.872	J	2.382
A	87.32	B	518.185	G	73.029	H	32.704	I	2.901	J	2.393
A	87.00	B	518.185	G	72.076	H	33.974	I	2.915	J	2.398
A	85.73	B	553.747	G	72.711	H	32.704	I	2.905	J	2.394
A	87.32	B	505.485	G	72.076	H	93.032	I	2.915	J	2.398
A	89.22	B	223.531	G	72.711	H	86.047	I	2.905	J	2.394
A	127.01	B	576.925	G	72.076	H	86.047	I	2.915	J	2.398
A	101.92	B	502.944	G	72.076	H	85.094	I	2.915	J	2.398
A	87.95	B	445.474	G	72.711	H	85.729	I	2.905	J	2.394
A	111.77	B	607.089	G	71.123	H	86.682	I	2.929	J	2.403
A	105.73	B	618.520	G	71.123	H	31.434	I	2.929	J	2.403
A	90.49	B	632.491	G	71.123	H	32.704	I	2.929	J	2.403
A	87.95	B	647.731	G	71.123	H	32.704	I	2.929	J	2.403
A	87.32	B	647.731	G	71.123	H	32.069	I	2.929	J	2.403
A	87.00	B	647.731	G	71.123	H	32.387	I	2.929	J	2.403
A	86.68	B	647.731	G	71.123	H	32.069	I	2.929	J	2.403
A	66.36	B	82.554	G	71.123	H	32.069	I	2.929	J	2.403
A	109.54	B	533.426	G	66.996	H	31.434	I	2.991	J	2.426
A	78.74	B	94.302	G	66.996	H	30.799	I	2.991	J	2.426
A	85.09	B	116.846	G	66.996	H	30.799	I	2.991	J	2.426
A	86.05	B	129.864	G	68.266	H	23.179	I	2.972	J	2.419
A	107.00	B	142.564	G	68.583	H	27.941	I	2.967	J	2.417
A	101.92	B	154.630	G	69.218	H	30.481	I	2.957	J	2.414
A	93.03	B	166.061	G	68.583	H	30.481	I	2.967	J	2.417
A	68.90	B	243.534	G	72.711	H	30.481	I	2.905	J	2.394
A	74.30	B	182.571	G	45.405	H	22.861	I	2.884	J	2.554
A	71.44	B	172.093	G	51.755	H	24.449	I	2.980	J	2.515

A	84.78	B	234.009	G	35.244	H	20.639	I	2.731	J	2.616
A	83.82	B	223.848	G	58.423	H	26.036	I	3.080	J	2.474
A	12.70	B	226.071	G	30.481	H	0.000	I	2.659	J	2.646
A	30.48	B	252.425	G	65.408	H	81.919	I	3.014	J	2.435
A	43.50	B	211.148	G	65.726	H	22.544	I	3.010	J	2.433
A	53.66	B	279.414	G	66.361	H	30.481	I	3.000	J	2.430
A	68.90	B	102.557	G	63.821	H	29.211	I	3.038	J	2.444
A	87.95	B	211.148	G	64.456	H	30.799	I	3.028	J	2.440
A	93.98	B	317.515	G	65.726	H	30.481	I	3.010	J	2.433
A	67.00	B	142.247	G	63.821	H	30.481	I	3.038	J	2.444
A	105.73	B	147.645	G	66.996	H	33.339	I	2.991	J	2.426
A	107.64	B	157.805	G	65.091	H	34.927	I	3.019	J	2.437
A	103.51	B	173.998	G	66.996	H	33.974	I	2.991	J	2.426
A	105.10	B	175.268	G	65.726	H	32.704	I	3.010	J	2.433
A	111.77	B	264.173	G	68.583	H	23.179	I	2.967	J	2.417
A	58.42	B	208.608	G	62.551	H	23.814	I	3.057	J	2.451
A	65.73	B	224.483	G	56.518	H	31.434	I	3.052	J	2.486
A	71.12	B	246.392	G	43.182	H	27.941	I	2.851	J	2.568
A	76.84	B	215.910	G	22.861	H	16.511	I	2.650	J	2.650
A	71.12	B	269.571	G	32.387	H	73.346	I	2.688	J	2.634
A	72.08	B	264.173	G	63.186	H	89.857	I	3.047	J	2.447
A	33.34	B	242.899	G	63.186	H	96.525	I	3.047	J	2.447
A	28.58	B	309.895	G	63.186	H	13.018	I	3.047	J	2.447
A	58.42	B	143.517	G	67.631	H	25.719	I	2.981	J	2.423
A	83.82	B	210.830	G	63.821	H	30.481	I	3.038	J	2.444
A	81.92	B	398.799	G	64.456	H	28.576	I	3.028	J	2.440
A	23.18	B	401.339	G	64.456	H	2.858	I	3.028	J	2.440
A	0.00	B	568.035	G	63.821	H	0.000	I	3.038	J	2.444
A	6.35	B	221.308	G	62.551	H	11.113	I	3.057	J	2.451
A	30.80	B	469.923	G	68.583	H	0.000	I	2.967	J	2.417
A	0.00	B	416.580	G	64.456	H	21.909	I	3.028	J	2.440
A	35.24	B	415.310	G	65.091	H	29.211	I	3.019	J	2.437

A	80.65	B	574.068	G	64.456	H	15.241	I	3.028	J	2.440
A	15.24	B	533.426	G	65.091	H	18.416	I	3.019	J	2.437
A	26.04	B	574.068	G	65.091	H	18.733	I	3.019	J	2.437
A	46.36	B	637.571	G	61.916	H	12.701	I	3.066	J	2.454
A	37.78	B	248.932	G	63.503	H	22.861	I	3.043	J	2.445
A	66.04	B	614.710	G	61.916	H	17.781	I	3.066	J	2.454
A	42.86	B	635.031	G	61.280	H	16.193	I	3.076	J	2.458
A	38.10	B	640.111	G	63.821	H	17.781	I	3.038	J	2.444
A	33.97	B	220.991	G	60.645	H	24.131	I	3.085	J	2.461
A	87.32	B	210.830	G	61.280	H	26.036	I	3.076	J	2.458
A	40.96	B	233.691	G	60.645	H	21.274	I	3.085	J	2.461
A	84.46	B	211.783	G	63.186	H	11.113	I	3.047	J	2.447
A	25.40	B	202.257	G	65.726	H	27.941	I	3.010	J	2.433
A	85.09	B	193.049	G	63.186	H	27.306	I	3.047	J	2.447
A	67.00	B	283.541	G	53.660	H	23.179	I	3.009	J	2.503
A	63.19	B	223.531	G	71.758	H	30.481	I	2.920	J	2.400
A	89.86	B	594.389	G	89.222	H	35.562	I	2.659	J	2.303
A	80.97	B	472.463	G	22.861	H	27.941	I	2.650	J	2.650
A	87.32	B	215.910	G	64.456	H	68.266	I	3.028	J	2.440
A	82.87	B	386.099	G	65.726	H	25.084	I	3.010	J	2.433
A	58.74	B	173.998	G	64.456	H	24.131	I	3.028	J	2.440
A	91.76	B	421.660	G	66.361	H	31.434	I	3.000	J	2.430
A	94.30	B	607.089	G	63.821	H	30.481	I	3.038	J	2.444
A	84.14	B	525.805	G	64.456	H	30.481	I	3.028	J	2.440
A	93.03	B	640.111	G	64.456	H	30.481	I	3.028	J	2.440
A	87.32	B	647.731	G	66.996	H	30.481	I	2.991	J	2.426
A	85.41	B	648.049	G	66.996	H	30.799	I	2.991	J	2.426
A	75.57	B	256.552	G	66.996	H	30.799	I	2.991	J	2.426
A	76.20	B	72.076	C	383.559	D	33.657	E	2.707	F	2.234
A	61.28	B	374.668	G	91.762	H	38.102	I	2.650	J	2.300
A	119.39	B	228.611	G	42.547	H	68.266	I	2.841	J	2.572
A	89.22	B	228.929	G	96.525	H	34.927	I	2.650	J	2.300

A	63.50	B	163.520	G	9.525	H	0.000	I	2.650	J	2.650
A	5.08	B	161.298	G	99.065	H	41.595	I	2.650	J	2.300
A	73.98	B	160.663	G	99.065	H	33.974	I	2.650	J	2.300
A	50.17	B	160.345	G	94.302	H	31.434	I	2.650	J	2.300
A	18.73	B	161.298	G	112.083	H	6.668	I	2.650	J	2.300
A	13.97	B	160.345	G	105.733	H	26.671	I	2.650	J	2.300
A	96.52	B	160.663	G	96.842	H	49.850	I	2.650	J	2.300
A	71.12	B	161.298	G	93.985	H	42.865	I	2.650	J	2.300

APPENDIX D: Programming Source Code

Main Program

```
#include <p33Fxxxx.h>
#define FCY 23031250UL
#define BAUDRATE 9600
#define BRGVAL ((FCY/BAUDRATE)/16)-1
//#define FCY 82912500UL
#include <libpic30.h>
#include "IO_UNLOCK.h"
#include "SPI.h"

// For Ultrasonic => Volts/Inch = Vcc/512
# define Volts_per_Inch 8


// DSPIC33FJ128MC802 Configuration Bit Settings
int FBS __attribute__((space(prog), address(0xF80000))) = 0xCF ;
int FSS __attribute__((space(prog), address(0xF80002))) = 0xCF ;
int FGS __attribute__((space(prog), address(0xF80004))) = 0x7 ;
int FOSCSEL __attribute__((space(prog), address(0xF80006))) = 0x1 ;
int FOSC __attribute__((space(prog), address(0xF80008))) = 0xC3 ;
int FWDT __attribute__((space(prog), address(0xF8000A))) = 0x5F ;
int FPOR __attribute__((space(prog), address(0xF8000C))) = 0xF7 ;
int FICD __attribute__((space(prog), address(0xF8000E))) = 0xFFDE ;
```

```
//structure for Input_Distance
 typedef struct {
     float min;
     float peak;
     float max;
 } Input_Distance;

 // Setting for  Input_Distance

   // Input Distance for Near MF (0cm, 30cm, 60cm)
   Input_Distance Left_Negative = {0, 0X5E, 0XBC};

   // Input Distance for Just Right MF (30cm, 60cm, 90cm)
   Input_Distance Just_Right = {0X5E, 0XBC, 0X11B};

   // Input Distance for Far MF (60cm, 90cm, 120cm)
   Input_Distance Right_Positive = {0XBC, 0X11B, 0x7FF};


 // Structure for Input Distance DOM

 typedef struct {
     float Positive;
     float Just_Right;
     float Negative;

 } Input_Distance_Membership;



 // Triangle_Membership_Function
 float Triangle_Membership ( float a,float b, float c, float x);
```

```
// Trapezoidal_Membership_Function
float Trapezoidal_Membership (float a, float b, float c ,float d, float x);
//Fuzzy Rules checking MIN
float MIN( float min1, float min2); //float min3

void Uart_Trans (unsigned int z);
void New_Line ();
void Send_a ();
void Program_On ();
void Program_Off ();
void Left_Avoidance();
void Right_Avoidance();
float Init_State;


int main()
{
  IO_UNLOCK ();

  AD1CON1bits.FORM = 0b00;          //integer form
  AD1CON1bits.SSRC=0b000;
  AD1CON1bits.ASAM=0;               //Sampling begin when SAMP bit is set
  AD1CON1bits.AD12B=1;              //12bit, 1 channel ADC operation
  AD1CON2bits.VCFG = 0b000;         //voltage reference AVdd & AVss
  AD1CON2bits.CSCNA = 0b0;          //Do not scan inputs
  AD1CON2bits.SMPI = 1;
  AD1CON2bits.ALTS = 0;      //always uses channel input selects for Sample A
  AD1CON3bits.ADRC = 0;
  AD1CON3bits.ADCS = 64;
  AD1CHS0bits.CH0NA = 0;
  AD1PCFGLbits.PCFG0 = 0;    //Set to analog mode
  AD1PCFGLbits.PCFG1 = 0;
  IFS0bits.AD1IF = 0;
  IEC0bits.AD1IE = 0;
```

```
AD1CON1bits.ADON = 1;

CS_Output=0;    //set RB8 to output
CS=1;           //set RB8 (connected to CS) = 1 (high) (refer to figure 1-1 MCP4822)
SCK = 0b01000;  //set RP7 to SCK1
SDO = 0b00111;  //set RP6 to SDO1 (PIC) and SDI (DAC)
SDO_Output= 0;

SPI_1_Init();

RPOR7bits.RP15R=0b00011;        //U1TX
TRISBbits.TRISB15       = 0;    //U1TX

U1MODEbits.STSEL=0;             // 1 Stop bit
U1MODEbits.PDSEL=0;             // No Parity, 8 data bits
U1MODEbits.ABAUD=0;             // Auto-Baud Disabled
U1MODEbits.BRGH=0;              // Low Speed Mode

U1BRG = 149;                    // BAUD Rate Setting for 9600

U1STAbits.UTXISEL0=0;           // Interrupt after one TX character is transmitted
U1STAbits.UTXISEL1=0;
IFS0bits.U1TXIF=0;
IEC0bits.U1TXIE=0;              // Disable UART TX Interrupt

U1MODEbits.UARTEN=1;            // Enable UART
U1STAbits.UTXEN=1;              // Enable UART TX



TRISBbits.TRISB4=1;             // Set RB4 as input
  while (1)
  {
    if (PORTBbits.RB4==0)
      {
```

```
                Program_Off();
            }

        else if (PORTBbits.RB4==1)
            {
                Program_On();
            }


    }
}

 void Program_Off ()
 {
  int b;

     CS=0;
     SPI1BUF = (0x1000 |2467);
     while (IFS0bits.SPI1IF == 0);
     b=SPI1BUF;
     IFS0bits.SPI1IF = 0;
     CS=1;



     CS=0;
     SPI1BUF = (0x9000|2467);
     while (IFS0bits.SPI1IF == 0);
     b=SPI1BUF;
     IFS0bits.SPI1IF = 0;
     CS=1;

  return;
 }
```

```
void Program_On ()
{

  float USN_FL,USN_FR,R1,R2,L1,L2,MIN_LEFT,MIN_RIGHT;


    //AN0 (FRONT LEFT) INPUT
    AD1CHS0bits.CH0SA=0;
    AD1CHS0bits.CH0NA=0;
    __delay_us(1000);
    AD1CON1bits.SAMP=1;
    __delay_us(1000);
    AD1CON1bits.SAMP=0;

    while (IFS0bits.AD1IF==0);
    IFS0bits.AD1IF=0;

    USN_FL = ADC1BUF0;
    U1TXREG = 'A';

    Send_a ();
    Uart_Trans (ADC1BUF0);
    New_Line ();


    //AN1 (FRONT RIGHT) INPUT
    AD1CHS0bits.CH0SA=0b00001;
     AD1CHS0bits.CH0NA=0;
    __delay_us(1000);
    AD1CON1bits.SAMP=1;
    __delay_us(1000);
    AD1CON1bits.SAMP=0;
```

```
while (IFS0bits.AD1IF==0);
IFS0bits.AD1IF=0;

USN_FR = ADC1BUF0;
U1TXREG = 'B';
Send_a ();
Uart_Trans (ADC1BUF0);
New_Line ();


// AN2 RIGHT FRONT INPUT
AD1CHS0bits.CH0SA=0b00010;
AD1CHS0bits.CH0NA=0;
__delay_us(1000);
AD1CON1bits.SAMP=1;
__delay_us(1000);
AD1CON1bits.SAMP=0;
while (IFS0bits.AD1IF==0);
IFS0bits.AD1IF=0;
R1= ADC1BUF0;


//AN3 RIGHT BACK INPUT
AD1CHS0bits.CH0SA=0b00011;
AD1CHS0bits.CH0NA=0;
__delay_us(1000);
AD1CON1bits.SAMP=1;
__delay_us(1000);
AD1CON1bits.SAMP=0;
while (IFS0bits.AD1IF==0);
IFS0bits.AD1IF=0;
R2= ADC1BUF0;

//AN4 LEFT FRONT INPUT
```

```
AD1CHS0bits.CH0SA=0b00100;
AD1CHS0bits.CH0NA=0;
__delay_us(1000);
AD1CON1bits.SAMP=1;
__delay_us(1000);
AD1CON1bits.SAMP=0;
while (IFS0bits.AD1IF==0);
IFS0bits.AD1IF=0;
L1=ADC1BUF0;

//AN5 LEFT BACK INPUT
AD1CHS0bits.CH0SA=0b00101;
AD1CHS0bits.CH0NA=0;
__delay_us(1000);
AD1CON1bits.SAMP=1;
__delay_us(1000);
AD1CON1bits.SAMP=0;
while (IFS0bits.AD1IF==0);
IFS0bits.AD1IF=0;
L2=ADC1BUF0;
MIN_LEFT = MIN (L1,L2);
MIN_RIGHT= MIN (R1,R2);

if (USN_FL<=400||USN_FR<=400 || MIN_LEFT <=200 || MIN_RIGHT<=200 )
{
  if (USN_FL<USN_FR || MIN_LEFT<MIN_RIGHT)
  {
    Left_Avoidance();
  }

  if (USN_FL>USN_FR||MIN_LEFT>MIN_RIGHT)
  {
    Right_Avoidance();
  }
```

```
}

else if (USN_FL>400&&USN_FR>400&&MIN_LEFT>200&&MIN_RIGHT>200)
  {
    int b;
    CS=0;
    SPI1BUF = (0x1000 |2650);
    while (IFS0bits.SPI1IF == 0);
    b=SPI1BUF;
    IFS0bits.SPI1IF = 0;
    CS=1;


    CS=0;
    SPI1BUF = (0x9000|2375);
    while (IFS0bits.SPI1IF == 0);
    b=SPI1BUF;
    IFS0bits.SPI1IF = 0;
    CS=1;
  }

  else
  {
    int b;
    CS=0;
    SPI1BUF = (0x1000 |2385);
    while (IFS0bits.SPI1IF == 0);
    b=SPI1BUF;
    IFS0bits.SPI1IF = 0;
    CS=1;
```

```
        CS=0;
        SPI1BUF = (0x9000|2375);
        while (IFS0bits.SPI1IF == 0);
        b=SPI1BUF;
        IFS0bits.SPI1IF = 0;
        CS=1;

  }
 return;
}

void Right_Avoidance ()
 {
    float USN_R1,USN_R2,USN_RIGHT_MIN;
    float Output_FB;
    int OutputA_FB;
    float Output_LR;
    int OutputB_LR;

    //AN2 (RIGHT_FRONT) INPUT
    AD1CHS0bits.CH0SA=0b00010;
    AD1CHS0bits.CH0NA=0;
    __delay_us(1000);
    AD1CON1bits.SAMP=1;
    __delay_us(1000);
    AD1CON1bits.SAMP=0;
    while (IFS0bits.AD1IF==0);
    IFS0bits.AD1IF=0;

    USN_R1 = ADC1BUF0 ;
    U1TXREG = 'C';

    Send_a ();
    Uart_Trans (ADC1BUF0);
```

```
New_Line ();


//AN3 (RIGHT_BACK) INPUT
AD1CHS0bits.CH0SA=0b00011;
AD1CHS0bits.CH0NA=0;
__delay_us(1000);
AD1CON1bits.SAMP=1;
__delay_us(1000);
AD1CON1bits.SAMP=0;
while (IFS0bits.AD1IF==0);
IFS0bits.AD1IF=0;

USN_R2 = ADC1BUF0 ;

U1TXREG = 'D';
Send_a ();
Uart_Trans (ADC1BUF0);
New_Line ();
USN_RIGHT_MIN = MIN (USN_R1,USN_R2);

// Fuzzification

Input_Distance_Membership Input_Right;

Input_Right.Negative = Trapezoidal_Membership (Left_Negative.min, Left_Negative.min, Left_Negative.peak, Left_Negative.max, USN_RIGHT_MIN);

Input_Right.Just_Right = Triangle_Membership (Just_Right.min, Just_Right.peak, Just_Right.max, USN_RIGHT_MIN);

Input_Right.Positive = Trapezoidal_Membership (Right_Positive.min, Right_Positive.peak, Right_Positive.max,Right_Positive.max, USN_RIGHT_MIN);
```

```
Output_FB = ((Input_Right.Positive*2600)+(Input_Right.Just_Right*2650)
+(Input_Right.Negative*2600))/
(Input_Right.Negative+Input_Right.Just_Right+Input_Right.Positive);

OutputA_FB = (int) Output_FB;

    U1TXREG = 'E';
    Send_a ();
    Uart_Trans (OutputA_FB);
    New_Line ();

    int b;

    CS=0;
    SPI1BUF = (0x1000 |OutputA_FB);
    while (IFS0bits.SPI1IF == 0);
    b=SPI1BUF;
    IFS0bits.SPI1IF = 0;
    CS=1;

Output_LR = ((Input_Right.Positive*2600)+(Input_Right.Just_Right*2467)
+(Input_Right.Negative*2250))/
(Input_Right.Negative+Input_Right.Just_Right+Input_Right.Positive);

OutputB_LR =(int) Output_LR;

    U1TXREG = 'F';
    Send_a ();
    Uart_Trans (OutputB_LR);
    New_Line ();
    U1TXREG = 'a';
    while (IFS0bits.U1TXIF==0);   //clear interrupt flag
```

```
    IFS0bits.U1TXIF=0;

    CS=0;
    SPI1BUF = (0x9000 |OutputB_LR);
    while (IFS0bits.SPI1IF == 0);
    b=SPI1BUF;
    IFS0bits.SPI1IF = 0;
    CS=1;

    return;
 }

void Left_Avoidance ()
 {

    float USN_L1,USN_L2,USN_LEFT_MIN;
    float Output_FB;
    int OutputA_FB;
    float Output_LR;
    int OutputB_LR;

    //AN4(LEFT_1) INPUT
    AD1CHS0bits.CH0SA=0b000100;
    AD1CHS0bits.CH0NA=0;
    __delay_us(1000);
    AD1CON1bits.SAMP=1;
    __delay_us(1000);
    AD1CON1bits.SAMP=0;
    while (IFS0bits.AD1IF==0);
    IFS0bits.AD1IF=0;

    USN_L1 = ADC1BUF0 ;
    U1TXREG = 'G';
    Send_a ();
```

```
Uart_Trans (ADC1BUF0);
New_Line ();
//AN5 (LEFT_2) INPUT
AD1CHS0bits.CH0SA=0b00101;
AD1CHS0bits.CH0NA=0;
__delay_us(1000);
AD1CON1bits.SAMP=1;
__delay_us(1000);
AD1CON1bits.SAMP=0;
while (IFS0bits.AD1IF==0);
IFS0bits.AD1IF=0;

USN_L2 = ADC1BUF0 ;
U1TXREG = 'H';

Send_a ();
Uart_Trans (ADC1BUF0);
New_Line ();
USN_LEFT_MIN = MIN (USN_L1,USN_L2);

// Fuzzification


Input_Distance_Membership Input_Left;

Input_Left.Negative = Trapezoidal_Membership (Left_Negative.min,
Left_Negative.min, Left_Negative.peak, Left_Negative.max, USN_LEFT_MIN);

Input_Left.Just_Right = Triangle_Membership (Just_Right.min, Just_Right.peak,
Just_Right.max, USN_LEFT_MIN);

Input_Left.Positive = Trapezoidal_Membership (Right_Positive.min,
Right_Positive.peak,Right_Positive.max, Right_Positive.max, USN_LEFT_MIN);
```

```
Output_FB = ((Input_Left.Positive*2600)+(Input_Left.Just_Right*2650)
+(Input_Left.Negative*2600))/
(Input_Left.Negative+Input_Left.Just_Right+Input_Left.Positive);
OutputA_FB = (int) Output_FB;

    U1TXREG = 'I';

    Send_a ();
     Uart_Trans (OutputA_FB );
    New_Line ();

    int b;
    CS=0;
    SPI1BUF = (0x1000 |OutputA_FB);
    while (IFS0bits.SPI1IF == 0);
    b=SPI1BUF;
    IFS0bits.SPI1IF = 0;
    CS=1;

Output_LR = ((Input_Left.Positive*2250)+(Input_Left.Just_Right*2467)
+(Input_Left.Negative*2600))/
(Input_Left.Negative+Input_Left.Just_Right+Input_Left.Positive);

OutputB_LR =(int) Output_LR;

    U1TXREG = 'J';
    Send_a ();
    Uart_Trans (OutputB_LR );
    New_Line ();
    U1TXREG = 'a';
    while (IFS0bits.U1TXIF==0);   //clear interrupt flag
    IFS0bits.U1TXIF=0;
    CS=0;
    SPI1BUF = (0x9000 |OutputB_LR);
```

```
    while (IFS0bits.SPI1IF == 0);
    b=SPI1BUF;
    IFS0bits.SPI1IF = 0;
    CS=1;
    return;
 }

void Uart_Trans (unsigned int z)

 {
    char a[] = "XXXX";
    *a=(z%10000) / 1000| 0x30;
    *(a+1)= (z%1000) / 100| 0x30;
    *(a+2)= (z%100) / 10| 0x30;
    *(a+3)=(z%10)|0x30;

    int i;
    for (i=0;i<4;i++)
    {
       U1TXREG = a[i];
       while (IFS0bits.U1TXIF==0);   //clear interrupt flag
       IFS0bits.U1TXIF=0;
    }

 }


void New_Line ()

 {
    U1TXREG = '\t';
    while (IFS0bits.U1TXIF==0);   //clear interrupt flag
    IFS0bits.U1TXIF=0;
 }
```

```
void Send_a ()

 {
    while (IFS0bits.U1TXIF==0);   //clear interrupt flag
    IFS0bits.U1TXIF=0;
    U1TXREG = 'a';
    while (IFS0bits.U1TXIF==0);   //clear interrupt flag
    IFS0bits.U1TXIF=0;

 }



float Triangle_Membership ( float a,float b, float c, float x)

 {
    float Degree_Membership;

    if (x>a && x<b)
    {
       Degree_Membership = (x-a)/(b-a);
    }

    else if (x==b)
    {
       Degree_Membership = 1;
    }

    else if (x>b && x<c)
    {
       Degree_Membership = (c-x)/(c-b);
    }
```

```
    else
    {
        Degree_Membership = 0;
    }

    return Degree_Membership;
}


float Trapezoidal_Membership (float a,float b, float c, float d, float x)

{
    float Degree_Membership;

    if (x>a && x<b)
    {
        Degree_Membership = (x-a)/(b-a);
    }



    else if (x>=b && x<=c)
    {
        Degree_Membership = 1;
    }

    else if (x>c && x<d)
    {
        Degree_Membership = (d-x)/(d-c);
    }

    else
    {
        Degree_Membership = 0;
    }
```

```
    return Degree_Membership;
 }

float MIN( float min1, float min2)
 {
    float min_value;

    min_value = min1;

    if (min2<min_value)
      {
        min_value = min2;
      }

    else
      {
       min_value = min1;
      }

    return min_value;
 }
```

SPI Configuration

```
#include <p33Fxxxx.h>
#include "SPI.h"

void SPI_1_Init()
{
    IFS0bits.SPI1IF=0;                  //Clear interrupt flag
//    IFS0bits.SPI1EIF=0;
    //IEC0bits.SPI1IE=1;                //Enable the interrupt

    SPI1CON1bits.DISSCK=0;              //internal serial clock is enabled
    SPI1CON1bits.DISSDO=0;              //SDO1 pin is controlled by module
    SPI1CON1bits.MODE16=1;              //communication is word-wide (16 bits)
    SPI1CON1bits.SMP=1;
    SPI2STATbits.SPISIDL = 0;           //Continue module operation in Idle mode
    SPI1CON1bits.SSEN=0;                // SSx pin is not used by the module
    SPI1CON1bits.SPRE = 3 ;             //secondary prescale bits
    SPI1CON1bits.PPRE = 2;              //primary prescale bits
    SPI1CON1bits.CKP=0;
    SPI1CON1bits.CKE=1;

    SPI1CON1bits.MSTEN=1;               //Master Mode enabled
    SPI1STATbits.SPIEN=1;               //Enable SPI module
}
```

IO_UNLOCK

```
void IO_UNLOCK ()

{
    __builtin_write_OSCCONL(0x46);
    __builtin_write_OSCCONL(0x57);
    __builtin_write_OSCCONL(0x80);
}
```

SPI Initialization

```
# define CS LATBbits.LATB8
# define CS_Output TRISBbits.TRISB8
# define SCK  RPOR3bits.RP7R
# define SDO  RPOR3bits.RP6R
# define SDO_Output TRISBbits.TRISB6

void SPI_1_Init();
```