UNIVERSITI MALAYSIA PAHANG

BORANG PENGESAHAN STATUS TESIS♦

JUDUL: **AUTOMATED CAR BRAKING SYSTEM USING FUZZY LOGIC CONTROLLER**

SESI PENGAJIAN: **2007/2008**

Saya **MARHATINIE BINTI MAMAT (850831-11-5230)**
(HURUF BESAR)

mengaku membenarkan tesis (Sarjana Muda/~~Sarjana~~ /~~Doktor Falsafah~~)* ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Tesis adalah hakmilik Universiti Malaysia Pahang (UMP).
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. **Sila tandakan (√)

☐	**SULIT**	(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)
☐	**TERHAD**	(Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)
√	**TIDAK TERHAD**	

Disahkan oleh:

_________________________ (TANDATANGAN PENULIS)

_________________________ (TANDATANGAN PENYELIA)

Alamat Tetap:

322-1 KG BUKIT KUANG
24000 KEMAMAN
TERENGGANU

NOR MANIHA ABD GHANI
(Nama Penyelia)

Tarikh: **30 NOVEMBER 2007**

Tarikh: : **30 NOVEMBER 2007**

CATATAN:
* Potong yang tidak berkenaan.
** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh tesis ini perlu dikelaskan sebagai atau TERHAD.
♦ Tesis dimaksudkan sebagai tesis bagi Ijazah doktor Falsafah dan Sarjana secara Penyelidikan, atau disertasi bagi pengajian secara kerja kursus dan penyelidikan, atau Laporan Projek Sarjana Muda (PSM).

“I hereby acknowledge that the scope and quality of this thesis is qualified for the award of the Bachelor Degree of Electrical Engineering (Power Systems)”

Signature : ____

Name : NOR MAHINA ABD GHANI

Date : 30 NOVEMBER 2007

AUTOMATED CAR BRAKING SYSTEM USING FUZZY LOGIC CONTROLLER

MARHATINIE BINTI MAMAT

"This thesis is submitted as partial fulfillment of the requirements for the award of the Bachelor of Electrical Engineering (Power Systems)"

Faculty of Electrical & Electronics Engineering
Universiti Malaysia Pahang

NOVEMBER, 2007

"All the trademark and copyrights use herein are property of their respective owner. References of information from other sources are quoted accordingly; otherwise the information presented in this report is solely work of the author."

Signature : ______________________________

Author : MARHATINIE BINTI MAMAT

Date : 30 NOVEMBER 2007

To my beloved parents

"Mr Mamat Hj Ibrahim and Mrs Rahana Abdullah"

For your love, care, support, and believe in me. Thank you so much.

ACKNOWLEDGEMENT

Alhamdulillah, His Willingness has made it possible for me to complete the final year project in time. I would like to take this opportunity to express gratitude to my dedicated supervisor, Mrs Nor Maniha Abdul Ghani for guiding me this project at every stage with clarity and that priceless gift of getting things done by sharing his valuable ideas as well as share his knowledge. I would also like to thank to all UMP lecturers and electrical technicians whom had helped directly or indirectly in what so ever manner thus making this project a reality.

Not forgotten to my heartfelt thanks to my beloved parents, Mr Mamat Hj Ibrahim and Mrs Rahana Abdullah and also the rest of my dearest family whom always support and prays on me throughout this project. Also to my best colleagues for their openhandedly and kindly guided, assisted, and supported and encouraged me to make this project successful.

Their blessing gave me the high-spirit and strength to face any problem occurred and to overcome them rightly. The great cooperation, kindheartedness and readiness to share worth experiences that have been shown by them will be always appreciated and treasured by me, thank you.

ABSTRACT

This paper deals with a Fuzzy Logic Controller (FLC) for an automated car braking system. The response of the system will be simulated by using Fuzzy Logic Toolbox in MATLAB and PID controller. The purpose of this controller is to brake a car when the car approaches for an obstacle at a specific range. For this, the Fuzzy Logic Controller is design using the Fuzzy Logic Toolbox in MATLAB. The system uses four rules and three membership function. The two parameters such as distance and speed will be observed for both controllers and the ability to attenuate disturbance will be simulated. Output of the controller will determine the force of the car brake. Base on the simulation, it can be concluded that the response of Fuzzy Logic Controller is better than PID. However, PID controller can be used to constitute a reference for the performance of the fuzzy logic controller.

ABSTRAK

Projek ini adalah berkaitan dengan system kawalan Fuzzy Logic (FLC) untuk sistem brek kereta automatik. Respon sistem ini akan disimulasikan dengan menggunakan *Fuzzy Logic Toolbox* di dalam MATLAB dan melalui sistem kawalan *PID*. Tujuan sistem kawalan ini adalah untuk membrek kereta apabila ia menghampiri halangan di dalam jarak yang telah ditetapkan. Untuk itu, pengawal 'Fuzzy Logic' dihasilkan menggunakan *Fuzzy Logic Toolbox* di dalam MATLAB. System ini menggunakan empat syarat dan tiga *membership function*. Dua parameter iaitu jarak dan kelajuan akan ditinjau untuk kedua-dua system kawalan ini dan kebolehan untuk mengatasi gangguan akan disimulasikan. Keluaran dari sistem kawalan ini akan menentukan daya untuk brek kereta tersebut. Berdasarkan hasil simulasi, boleh disimpulkan bahawa tindak balas pengawal 'Fuzzy Logic' adalah lebih baik berbanding *PID*. Tetapi, system kawalan *PID* boleh digunakan sebagai rujukan untuk pengawal 'Fuzzy Logic'.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS

F	-	Force
m	-	Mass
s, y	-	Displacement
v, y’	-	Velocity
a, y”	-	Acceleration
Ke	-	Kinetic Energy
W	-	Work
e	-	Error
Td	-	Derivative gain
Kp	-	Proportional gain
ζ	-	Damping ratio
ω_n	-	Natural frequency

LIST OF ABBREVIATIONS

FLC	-	Fuzzy Logic Controller
PID	-	Proportional Integral Derivative
FIS	-	Fuzzy Inference System
PD	-	Proportional Derivative
PI	-	Proportional Integral
P	-	Proportional
I	-	Integral

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Fuzzy Logic Controller (FLC)

Fuzzy logic was formulated by Lotfi Zadeh of the University of California at Berkeley in the mid-1960s, based on earlier work in the area of fuzzy set theory. Zadeh also formulated the notion of fuzzy control that allows a small set of 'intuitive rules' to be used in order to control the operation of electronic devices. In the 1980s fuzzy control became a huge industry in Japan and other countries where it was integrated into home appliances such as vacuum cleaners, microwave ovens and video cameras. Such appliances could adapt automatically to different conditions; for instance, a vacuum cleaner would apply more suction to an especially dirty area. One of the benefits of fuzzy control is that it can be easily implemented on a standard computer.

Fuzzy controllers appear in consumer products such as washing machines, video cameras, cars. As for in industry, for controlling cement kilns, underground trains, and robots. A fuzzy controller is an automatic controller, a self-acting or self-regulating mechanism that controls an object in accordance with a desired behavior. The object can be, for instance, a robot set to follow a certain path. A fuzzy controller acts or regulates by means of rules in a more or less natural language, based on the distinguishing feature: fuzzy logic. The rules are invented by plant operators or design engineers, and fuzzy control is thus a branch of intelligent control.

1.2 Proportional Integral Derivative (PID) Controller

A proportional-integral-derivative controller (PID controller) is a generic control loop feedback mechanism widely used in industrial control systems. A PID controller attempts to correct the error between a measured process variable and a desired set point by calculating and then outputting a corrective action that can adjust the process accordingly.

The PID controller calculation (algorithm) involves three separate parameters; the Proportional, the Integral and Derivative values. The Proportional value determines the reaction to the current error, the Integral determines the reaction based on the sum of recent errors and the Derivative determines the reaction to the rate at which the error has been changing. The weighted sum of these three actions is used to adjust the process via a control element such as the position of a control valve or the power supply of a heating element.

1.3 Car Braking System

Braking system is the most important system in a car. If the brakes fail, the result can be disastrous. The brakes are in essence energy conversion devices, which convert the kinetic energy of the vehicle into thermal energy.

In this project, a car brake system will be controlled by the Fuzzy Logic Controller (FLC) and the Proportional Integral Derivative (PID) controller. The purpose of the automated car braking system is to develop an automated control system that would maintain a safe driving distance from obstacles while in traffic. The system will successfully detect an obstacle ahead at a specific range and create a way for the system to avoid collision by braking the car. By that, it will results in a more enjoyable and less stressful drive. The system will be developed in fuzzy logic toolbox available in MATLAB and will be simulated to see the performance of the car braking system.

1.4 Problem Statement

The increasing rate of road accident had been increasing nowadays. At present, there are four deaths per 10,000 vehicles. In many such cases, the cause of the accident is driver distraction and failure to react in time. Generally, a car brake system operated manually as the driver push the brake pedal. Therefore, to overcome this problem, an automated car braking system will be implemented to avoid such accident.

1.5 Objectives

The objectives of this project are:

I. To develop a Fuzzy Logic Controller and Proportional Integral Derivative Controller using MATLAB for an automated vehicle due to an obstacle.

II. To evaluate and analyze the performance of the systems.

1.6 Scope of Project

This project is to design a Fuzzy Logic Controller and Proportional Integral Derivative Controller that can be use to control a car brake automatically. Thus, the scopes that need to be considered in this project are:

I. Car brake

The car brake will be controlled by the fuzzy logic controller from the MATLAB toolbox and PID controller designed according to the range detected to the obstacle ahead by reducing the speed from the specified speed desired.

II. Range

The range targeted for the obstacle to be detected is 25m from the car. Therefore, the car will be brake and stop before it hit the obstacle.

III. Obstacle

Obstacles in this project refer to any objects including cars, human or animal those were ahead the car. The obstacles will give input to the controller to brake the car.

1.7 Literature Review

1.7.1 Car Braking Issues

Traffic congestion is a worldwide problem. This problem is mainly due to human driving which involves reaction times, delays, and judgement errors that may affect traffic flow and cause accidents. [1] In many such cases, the cause of the accident is driver distraction and failure to react in time. Advanced system of auxiliary functions has been develop to help avoid such accident and minimize the effects of collision should one occur. This is done by reducing the total stopping distance. [2] By that means, the car brake itself should have a good software system to assist a driver along the road.

Electronic brake control system has been making the car safer for the past 25 years. In recent years, braking developments have led to significantly greater driving safety. [3] For the past few years, there are many car brake development that uses the involvement of the electronic roles such as the Intelligent Cruise Control (ICC), Antilock Braking Systems (ABS), Traction Control System (TCS) and the Sensotronic Brake Control (SBC).

Many studies in this field depend upon a precise mathematical model of the vehicle. In fact, behaviors of the drivers are mostly based on the experience, not the exact mathematic computation. The model of vehicle is highly nonlinear function; it is difficult to find the precise model. Therefore, fuzzy logic systems have been designed by many researchers for automated driving controller since fuzzy system emulates the performance of a skilled human operator in the linguistic tulles that do not need use a mathematic model. [1]

Ordinary cruise control systems for passenger cars are becoming less and less meaningful because of the increasing traffic density rarely make it possible to drive at a preselected speed. However, in order to achieve high customer acceptance an intelligent cruise control system has to perform similarly to an experienced human driver. Therefore, it is necessary to adjust the following distance and the control dynamics according to the individual driver's needs. Applying fuzzy logic to intelligent cruise control seems to be an appropriate way to achieve this human behavior, because driver's experience can be transformed easily into rules. [4]

1.7.2 Fuzzy Logic Toolbox

Fuzzy logic imitates the logic of human thought, which is much less rigid than the calculations computer generally perform. [5] Intelligent control strategies mostly involve a large number of inputs. Most of the inputs are relevant for some specific condition. Using fuzzy logic, this input is only considered in the relevant rule. This keep the complex system transparent.[6] Whereby using fuzzy logic, the concept will be much easily to understand as it was based on natural language.

The objective of using fuzzy logic has been to make the computer think like people. Fuzzy logic can deal with the vagueness intrinsic to human thinking and natural language and recognize its nature is different from randomness. Using fuzzy logic algorithm could enable machines to understand and respond to vague human concept such as hot, cold, large, small, etc. It also could provide a relative simple approach to reach definite conclusion from imprecise information. [7]

Fuzzy logic is very adequate to built qualitative models of many kind of system without an extensive knowledge of their mathematical models. The use of fuzzy controllers allows achieving a human like vehicle operation. [8]

There are two general types of fuzzy expert system:

I. Fuzzy control
II. Fuzzy reasoning

Although both make use of fuzzy sets, they differ qualitatively in methodology. [9] Fuzzy control comprises the steps of sense, preprocess, fuzzify, evaluate, activate, aggregate, defuzzify and act. However, difficulty occurs with the using of fuzzy logic system. Usually it is difficult to determine the membership function and fuzzy logic rules. Many cycles of trail-and-error are required to achieve the desired performance. [1]

The fuzzy logic toolbox is a collection of function built on the MATLAB numeric computing environment. It provides tools for us to create and edit fuzzy interference system with the framework of MATLAB or integrate the fuzzy system into simulation with simulink. The fuzzy logic toolbox for use with MATLAB is a tool for solving problems with fuzzy logic. It is a fascinating area of research because it does a good job of trading off between significant and precision. [10]

Although it is possible to use Fuzzy Logic Toolbox by working strictly from the command line, in general it is much easier to build a system graphically. [10] There are five primary GUI tools for building, editing, and observing fuzzy inference systems in Fuzzy Logic Toolbox:

I. Fuzzy Inference System (FIS) Editor
II. Membership Function Editor
III. Rule Editor
IV. Rule Viewer
V. Surface Viewer

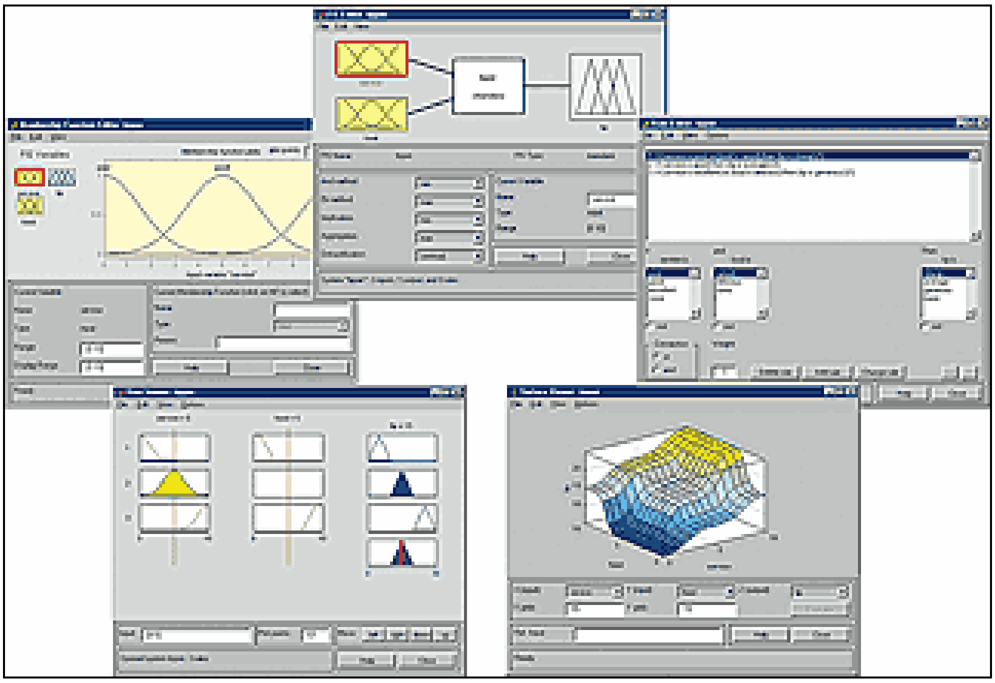


Figure 1.1 FIS Editor Toolbox

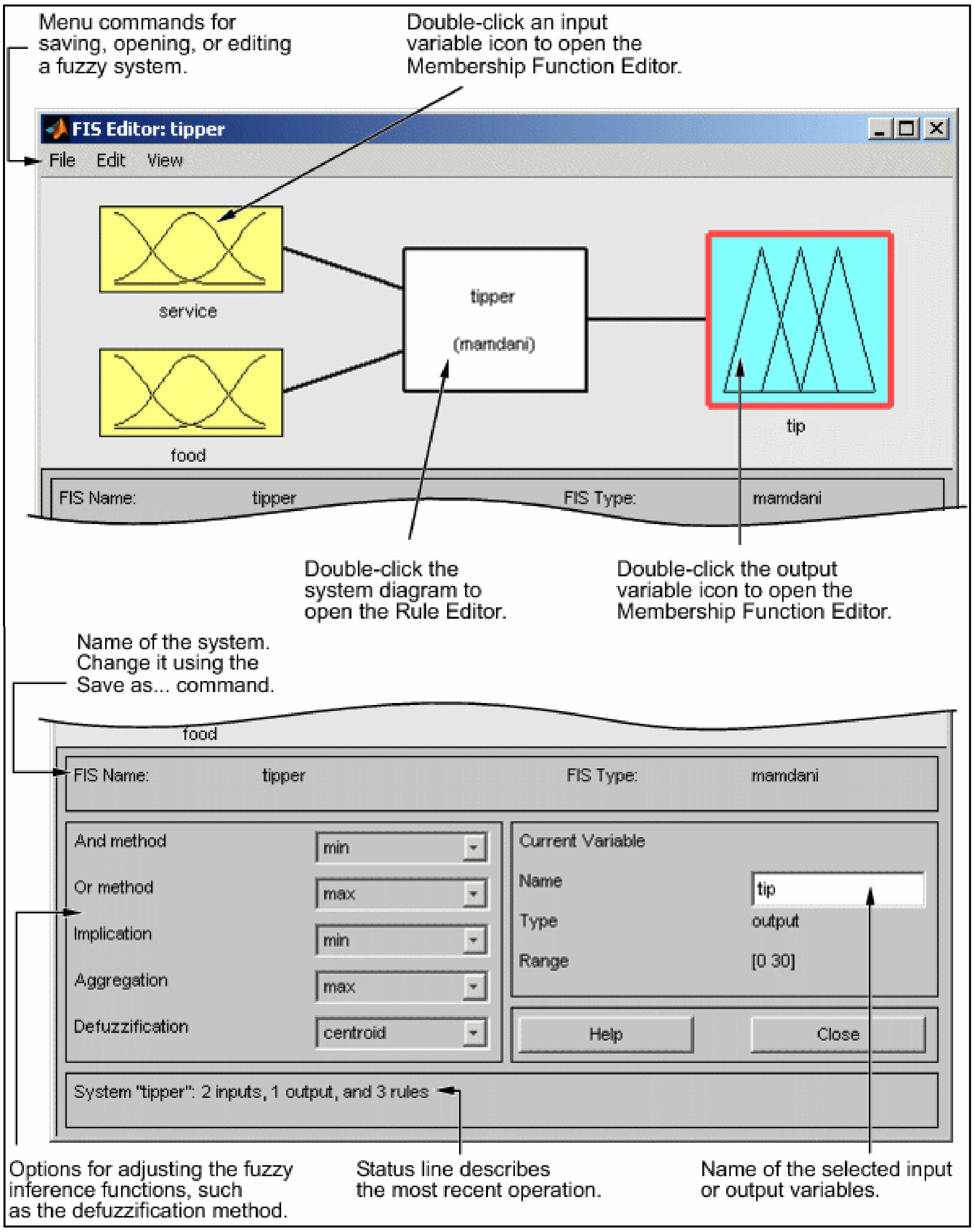


Figure 1.2 FIS Editor in MATLAB Toolbox

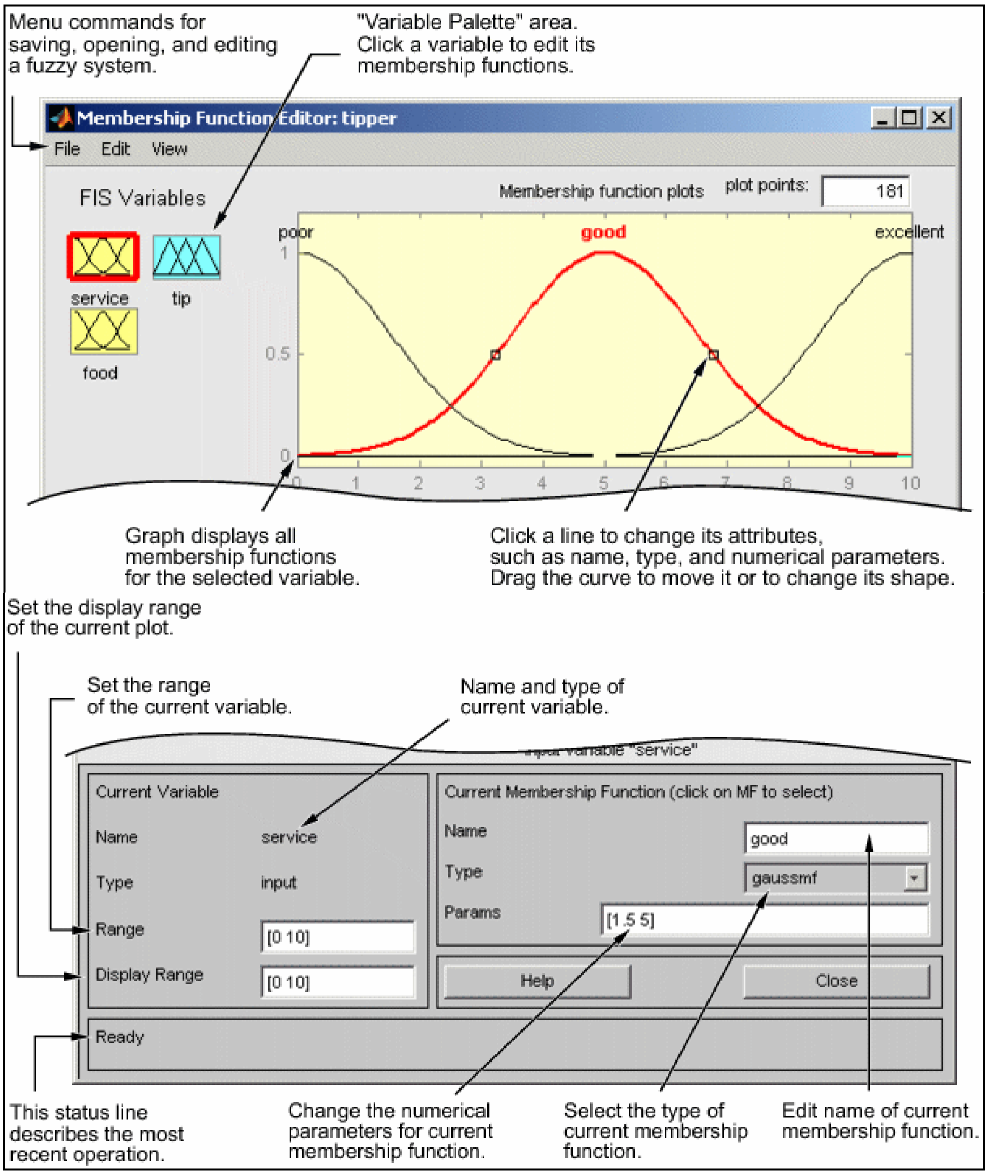


Figure 1.3 Membership Function Editor in MATLAB Toolbox

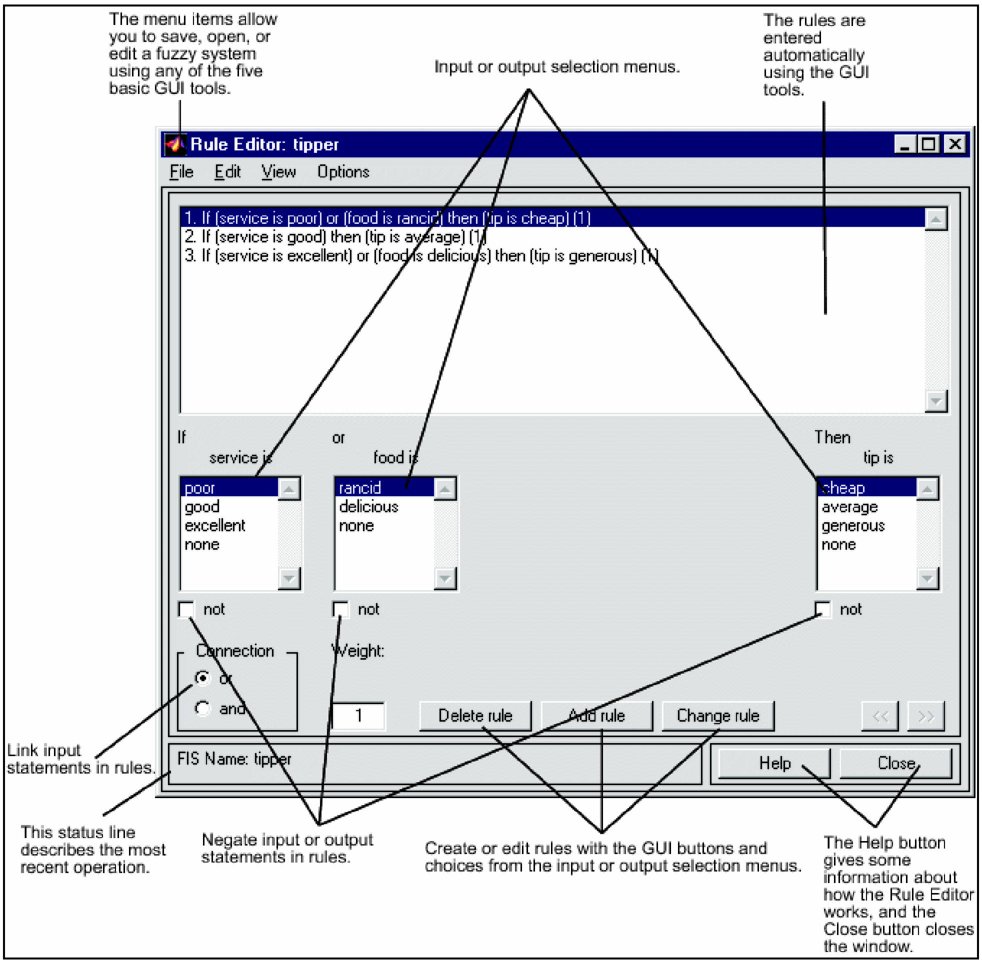


Figure 1.4 Rule Editor in MATLAB Toolbox

1.7.3 PID Controller

PID Control is the widest type of automatic control used in industry. There are many subtle variations in how it is applied in industry even though it has a relatively simple algorithm and structure. A PID controller can be used for regulation of speed, temperature, flow, pressure and other process variables.[11]

Some applications may require using only one or two modes to provide the appropriate system control. This is achieved by setting the gain of undesired control outputs to zero. A PID controller will be called a PI, PD, P or I controller in the absence of the respective control actions. [12]

The PID can provide control action designed for specific process requirements simply by tuning the three constants in the PID controller algorithm. The response of the controller can be described in terms of the responsiveness of the controller to an error, the degree to which the controller overshoots the set point and the degree of system oscillation.[12]

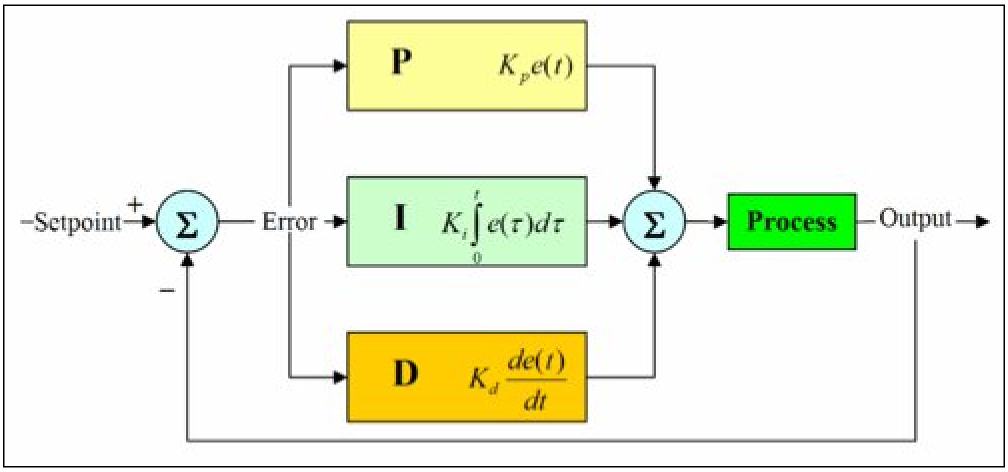


Figure 1.5 Block Diagram of a PID Controller

1.8 Methodology

1.8.1 Introduction

For a project to be done successfully, a methodology is one of the important elements. Methodology is the set of procedure of the project flow which includes the theories, concepts or ideas, comparative study of different approaches and critique of the individual methods will make sure that the project will run smoothly according to plan and to make sure that we can get the expected result.

1.8.2 Methodology

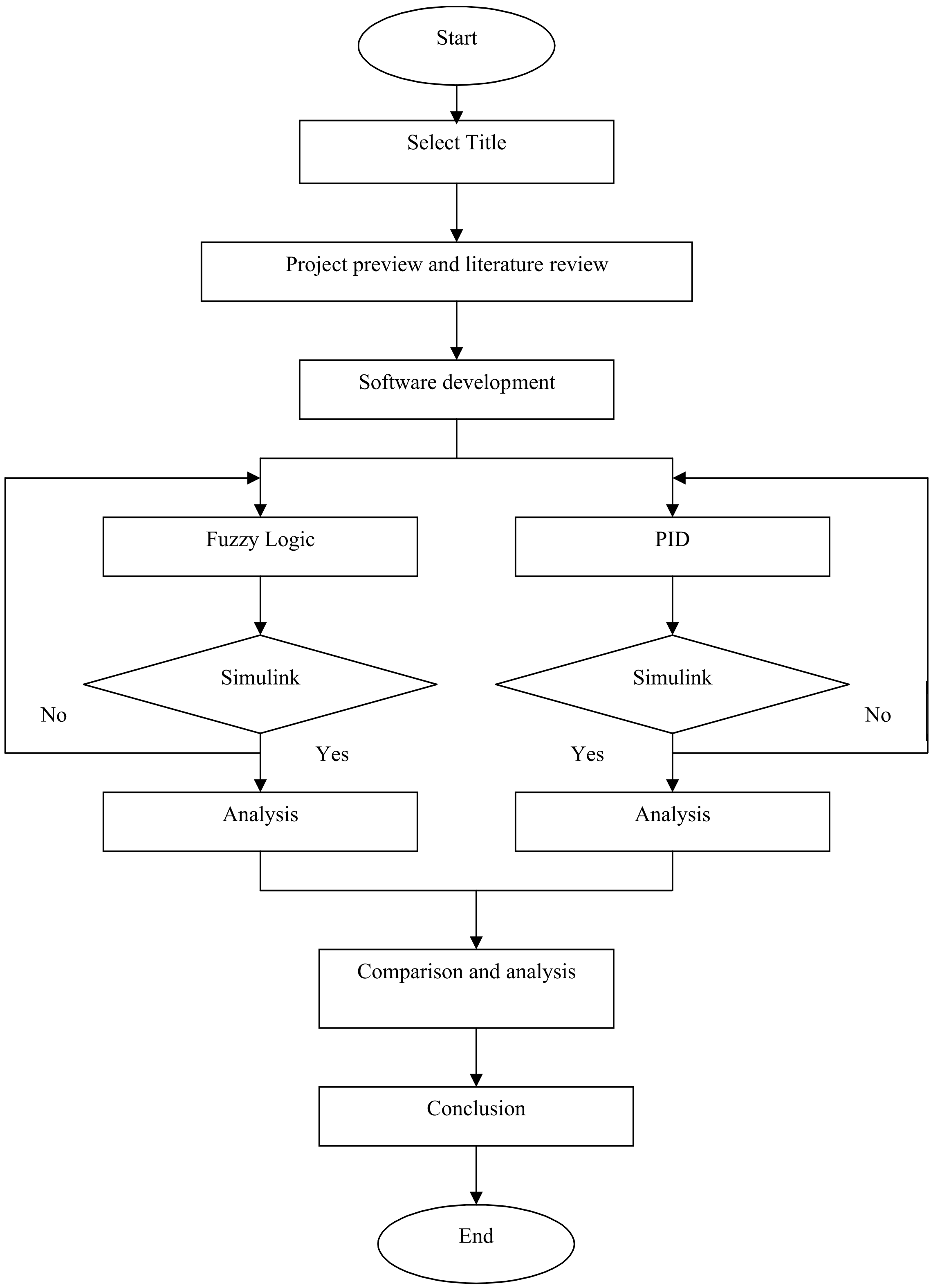


Figure 1.6 Flowchart of the Project Operation

1.8.3 Software Development

For this project, the controller will be implemented in two ways by using Fuzzy Logic Toolbox and PID. The result will be compared to know well both of the performance of the system. But first, a mathematical modeling of the car and brake system will be done to be implemented in the system.

1.8.4 System Model

When dealing with a control problem, often one of the first tasks that need to be undertakes is the development of the system model. Here, a system model represents a mathematical model of the process to be controlled, in order to gain a clear understanding of the problem. By then, a clearer view of how the system could work is developed as it provides method for the control system. Therefore, a descriptive model of the system as a hypothesis of how the system could work is built.

1.8.5 Control Design

After developing the system model for the car, a control design is developed. The control design consists of the controller for the car brake and also the simulink model for the car brake in both controller, PID and Fuzzy Logic Controller.

For PID controller, a controller is modeled as proportional derivative (PD) controller. Then, it is simulated in the MATLAB simulink to get the response for the controller.

As for Fuzzy Logic Controller (FLC), the controller is designed in the Fuzzy Logic Controller Toolbox in MATLAB. Then, it is integrated with the simulink using fuzzy logic controller block to see the output response of the controller so that it can be compared to the PID controller response.

1.8.6 Comparison and Analysis

From the result that we get in both controllers, a comparison will be done to evaluate the performance of both controllers. The output response that will be observed are:

I. Force of the car brake
II. Position of the car
III. Velocity of the car

1.8.7 Conclusion

After the system is developed, and the analysis is carried out, a conclusion is made in order to see the successfulness of this project based on the objectives that were set earlier. Thus, a recommendation is made for future progress for enhancement of this system.

CHAPTER 2

SYSTEM MODEL

2.1 Introduction

In order to design this project, a system modeling is necessary to provide method for the control system. By this, a descriptive model of the system as a hypothesis of how the system could work is built. For both controllers, the car will be modeled according to the Newton's second law of motion, F=ma.

2.2 System Model

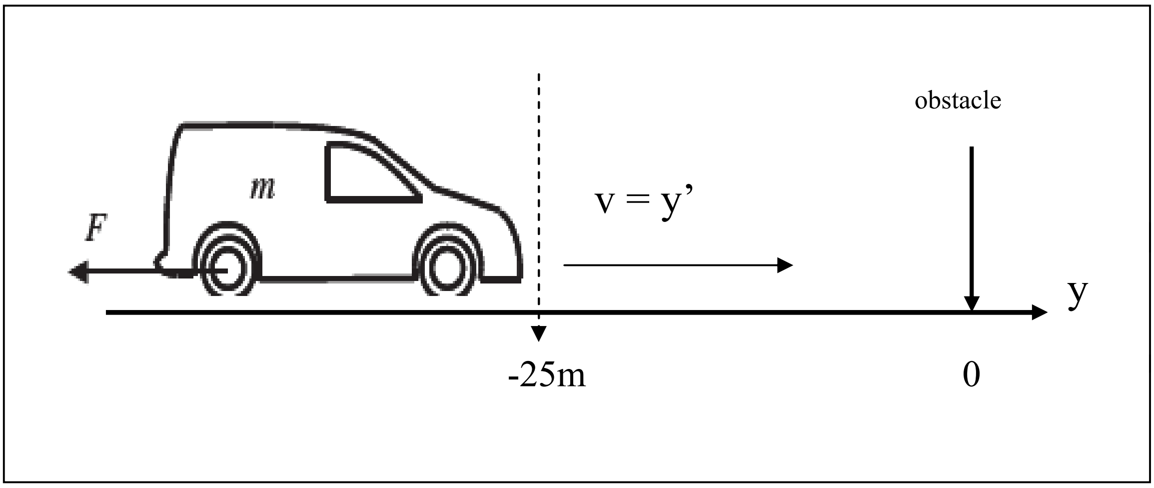


Figure 2.1 Model of a Car

To model the car, the engine dynamic, skidding, slip and friction of the car is disregard. Therefore, using Newton's second law of motion, the force F causes acceleration:

$F = ma$

Acceleration is the derivative of velocity y'. y' is the derivative of the position, y. Thus, a equals to y". Therefore, the differential equation models the motion of the car as

$F = my''$

Assuming the mass of the car is 1500kg, the initial position which the car will be controlled to brake is 25m and the initial velocity is 10ms-1. We have thus identified the following constants:

$m = 1500kg$
$y(0) = -25m$
$y'0) = 10ms^{-1}$

Considering that once the speed is zero, the car will not move anymore. The variable force, F is thus negative or zero, since the brake is the only means of control.

According to specification, when a brake is applied to a car of a speed of 80 km/h which in turn is $22.22ms^{-1}$ will bring the speed to zero at a distance of 27.3 m.

$v = 80km/h = 22.22ms^{-1}$
$y = 27.3m$

Knowing that kinetic energy is converted to work:

$Ke = ½ mv^2$
$W = Fs$

Therefore;

$\frac{1}{2} mv^2 = Fy$

$\frac{1}{2} (1500kg)(22.22ms^{-1})^2 = F(27.3m)$

$$F = \frac{my'^2}{2y}$$

$$= \frac{(1500kg)(22.22ms^{-1})^2}{2(27.3m)}$$

$$= 13600 \text{ N}$$

Therefore, it was assume that the automatic brake system limits the magnitude of the brake force to 13600N. The control signal is thus subject to the constraints

$-13\,600 \leq F \leq 0$

CHAPTER 3

CONTROL DESIGN

3.1 Introduction

Control design is to be aimed as a stage of designing the brake controller of both systems, Fuzzy Logic Controller (FLC) and Proportional Integral Derivative (PID). Afterward, these controllers are integrated into MATLAB simulink to simulate the car brake controller and to see the effectiveness of the car brake controller system.

3.2 Proportional Integral Derivative (PID) Controller Design

3.2.1 Introduction

On behalf of this type of controller, a PID is designed base on the basic block diagram of PID system in figure 3.1. Plant is a system that is to be controlled. Meanwhile, the controller is the means that provides the excitation for the plant and is designed to control the overall system behavior.

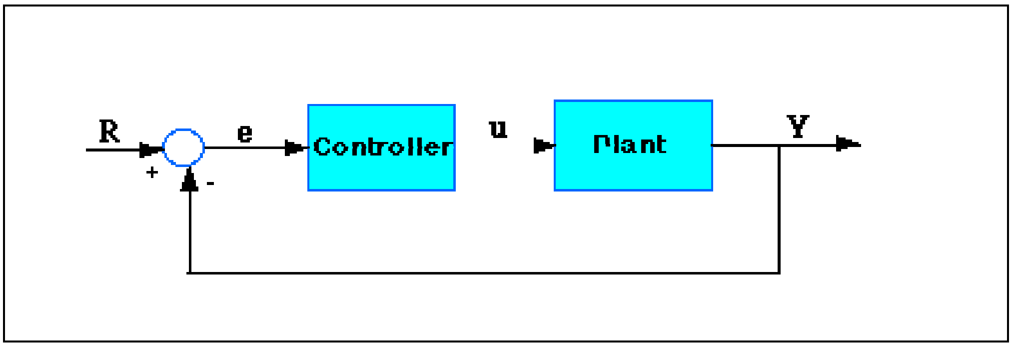


Figure 3.1 Block Diagram of PID System

3.2.2 PID Design

For the design of a PID controller, the driver is modeled as proportional-derivative (PD) controller. Thus, as for PD controller, the equation that were use is F=Kp(e+Td e').

The closed-loop system equations are

F=ma

F=Kp(e+Td e')

Rearranging both equations:

$$y'' = \frac{Kp\,(e + Td\,e')}{m}$$

$$= -\frac{KpTd}{m}(y') - \frac{Kp}{m}(y)$$

As for the steady state solution, since in steady state the system is at rest, that is, y" = y' = 0, and insertion yields the solution *y* = 0; this is in accordance with the problem specification. The transfer function of the closed-loop system is:

$$\frac{y(s)}{Ref} = \frac{\frac{K_p}{m}}{s^2 + \frac{K_p}{m}T_d s + \frac{K_p}{m}} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

The last expression is a general second-order system with natural frequency ωn as the frequency of oscillation of the system without damping and damping ratio, ζ. It is useful, because the car brake suppose to have a solution without overshoot, which is as fast as possible. This corresponds to the case $\zeta = 1$, which yields a critically damped response. Therefore, from deriving equation above resulting:

$$\zeta = \frac{1}{2}\sqrt{\frac{K_p}{m}}T_d$$

Applying $\zeta = 1$ results an optimal tuning relationship:

$$T_d = \frac{2}{\sqrt{\frac{K_p}{m}}}$$

It ensures that the response has no overshoot and there is a horizontal tangent at $y = 0$. Consequently the velocity will be zero when the car is stop. Assuming $Td = 1$, we will get $Kp = 6000$.

$Kp = 6000$

$Td = 1$

3.2.3 PID Simulink Model

Based on F=Kp(e+Td e'), the simulink model of PID controller is done after the value of Kp and Td are known. Thus, from the calculation, we get $Kp = 6000$ and

Td = 1. If the PID controller parameters which are the gains of the proportional and derivative terms are chosen incorrectly, the controlled process input can be unstable. Its output can be diverges, with or without oscillation, and is limited only by saturation or mechanical breakage. Tuning a control loop is the adjustment of its control parameters to the optimum values for the desired control response.

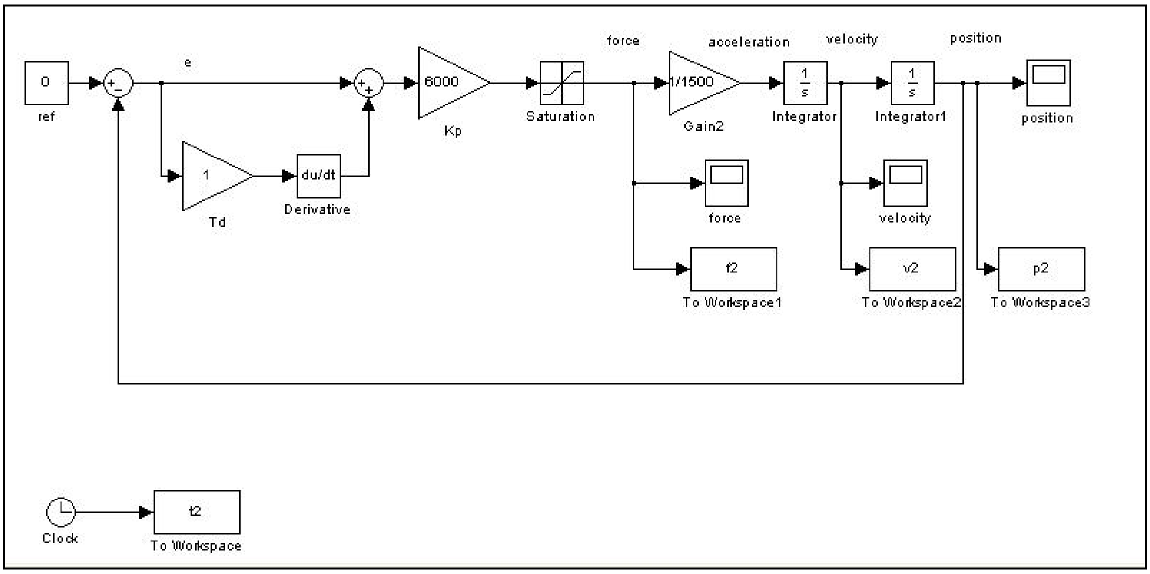


Figure 3.2 PID Simulink Model

3.3 Fuzzy Logic Controller (FLC) Design

3.3.1 Introduction

Fuzzy logic toolbox from MATLAB is used to develop a controller for fuzzy logic. Using the Fuzzy Inference System Editor (FIS), the editor involve is FIS editor, membership function editor and rule editor. Meanwhile, rule viewer and surface viewer are used to display the output of the controller designed.

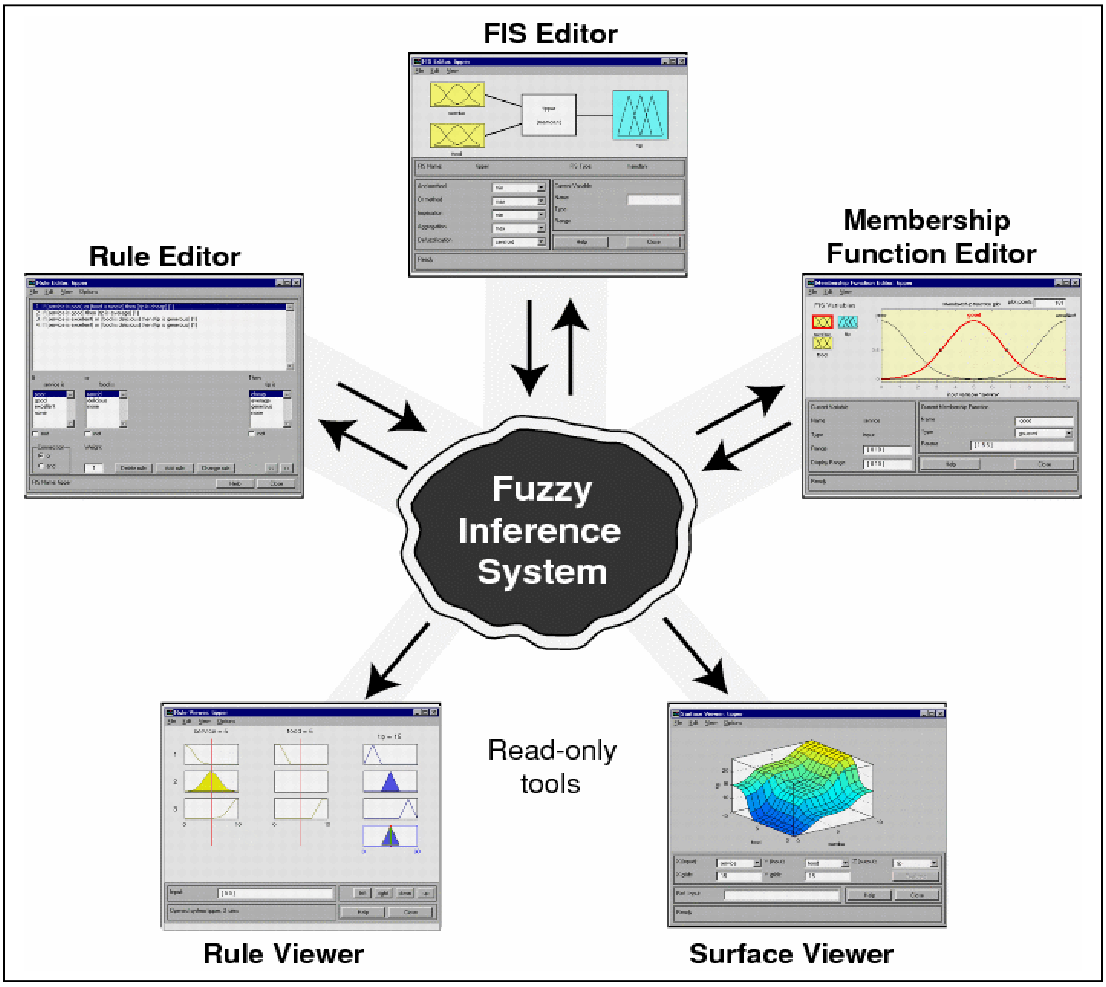


Figure 3.3 Fuzzy Inference Systems (FIS)

Afterwards, once the controller is complete, it is integrated with MATLAB simulink. This is done to simulate the controller of the car brake towards the car model itself. Thus, the performance of the car brake system is evaluated.

3.3.2 Fuzzy Logic Controller

3.3.2.1 FIS Editor

Using the fuzzy logic toolbox, the first things to be done is at the FIS Editor shown in figure 3.4. The FIS Editor handles the high-level issues for the system. It displays general information about a fuzzy inference system.

For the car brake controller design, there are two inputs and one output that are design through the toolbox. The inputs are position and velocity. The output is the brake.

The position represents the distance of the car from the obstacle detected. Velocity is measured from the velocity of the car towards the obstacle and brake represents the force of the car brake needed to stop the car. Defuzzication method used for this controller is mean of maximum (mom) method.

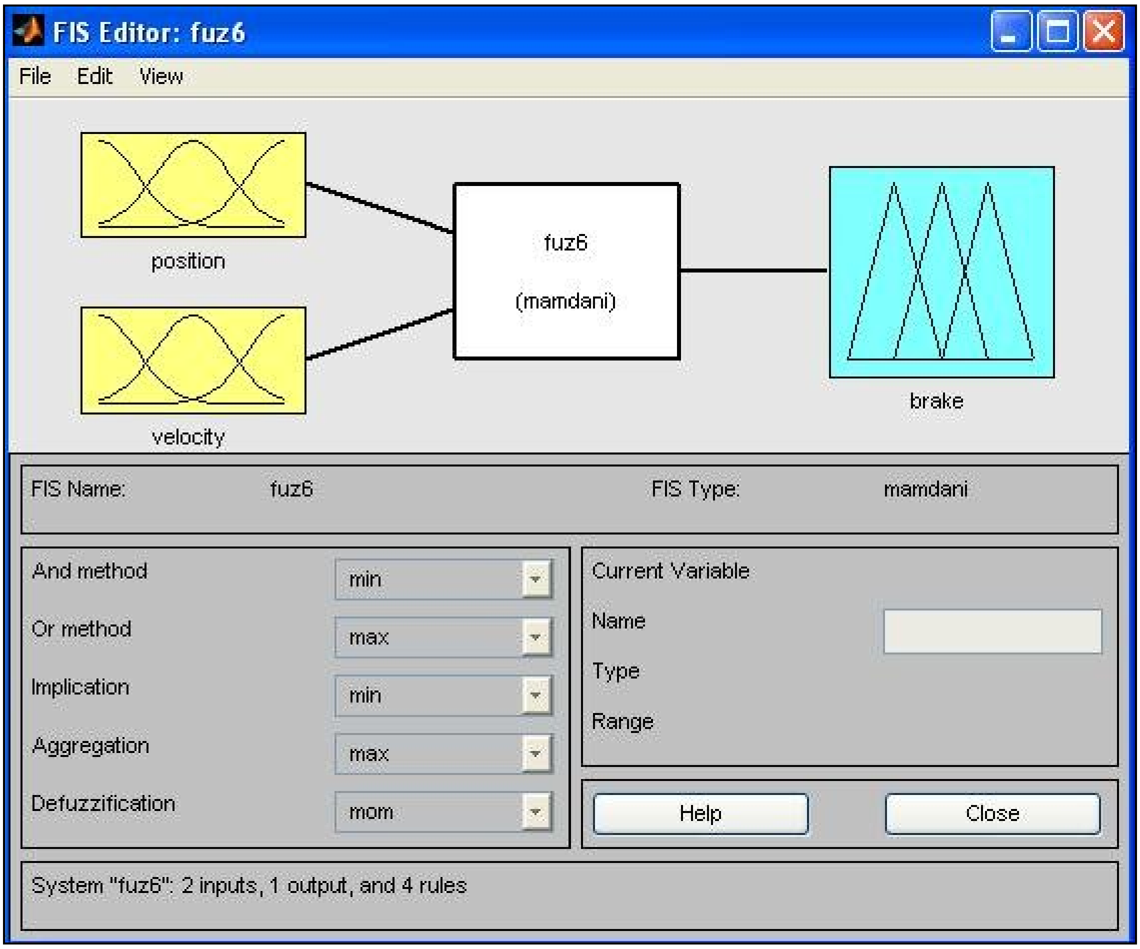


Figure 3.4 FIS Editor

3.3.2.2 Membership Function Editor

The Membership Function Editor is the tool that display and enable user to edit all of the membership functions associated with all of the input and output variables for the entire fuzzy inference system. It is used to define the shapes of all the membership functions associated with each variable.

For this car brake controller, as shown in figure 3.5 is the first input which is position that uses two membership functions:

I. Short
II. Long

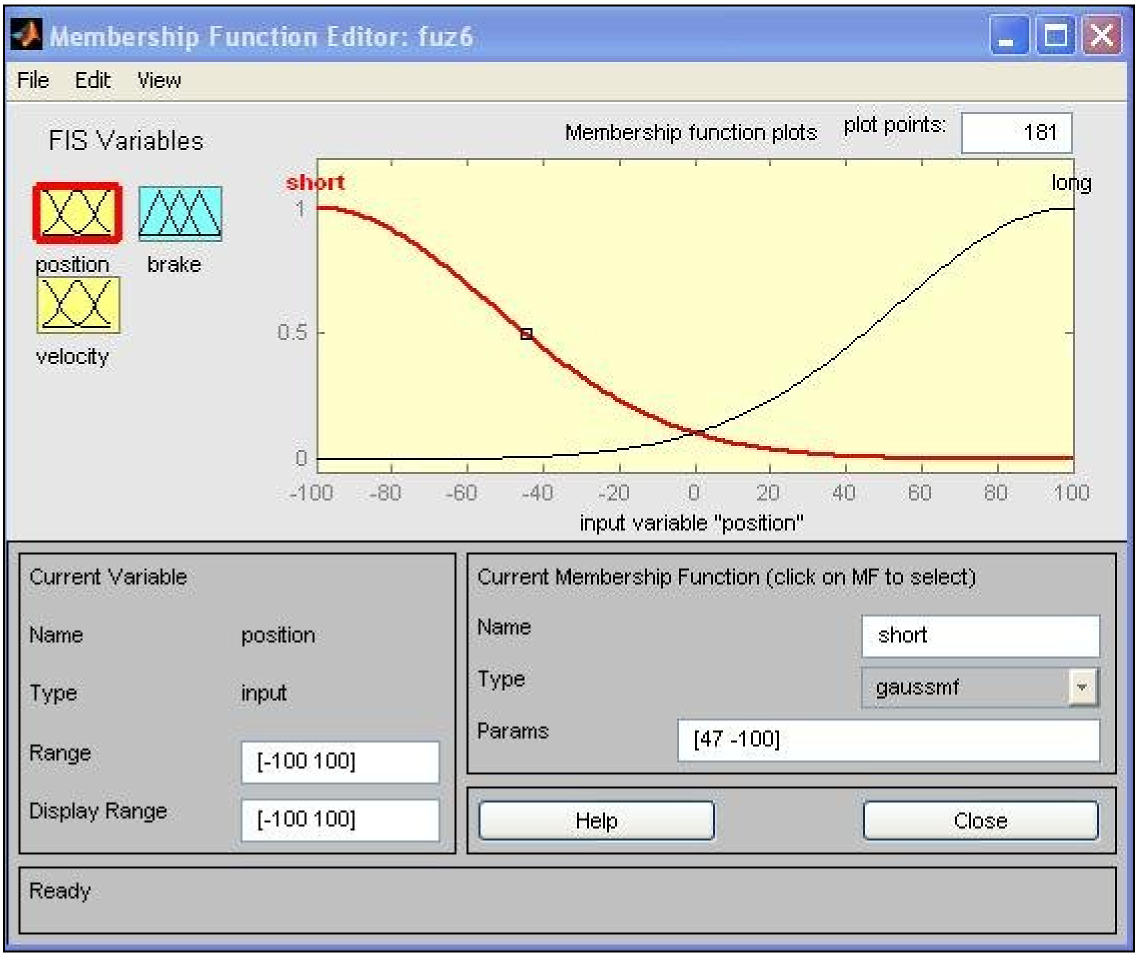


Figure 3.5 Membership Function Editor for Input Position

The figure 3.6 shows the second input, velocity. For this input, it also uses two membership functions which are:

I. Fast
II. Slow

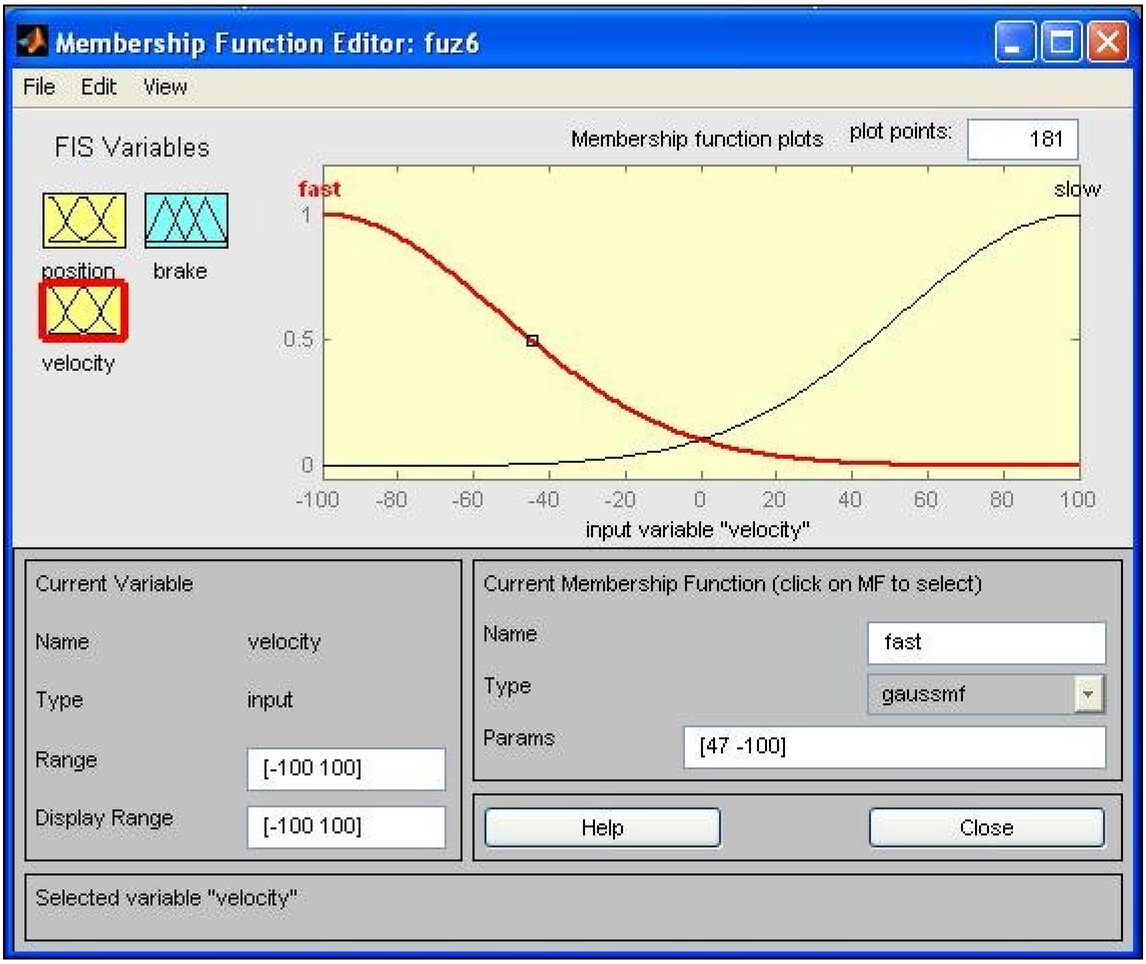


Figure 3.6 Membership Function Editor for Input Velocity

On the other hand, figure 3.7 shows the output, brake. Two membership functions are use which are:

I. Zero
II. Hard

As for the shape, trapezoidal-shaped built-in membership function (trapmf) shape is use. The parameters for the inputs and output membership functions are tune according to needs.

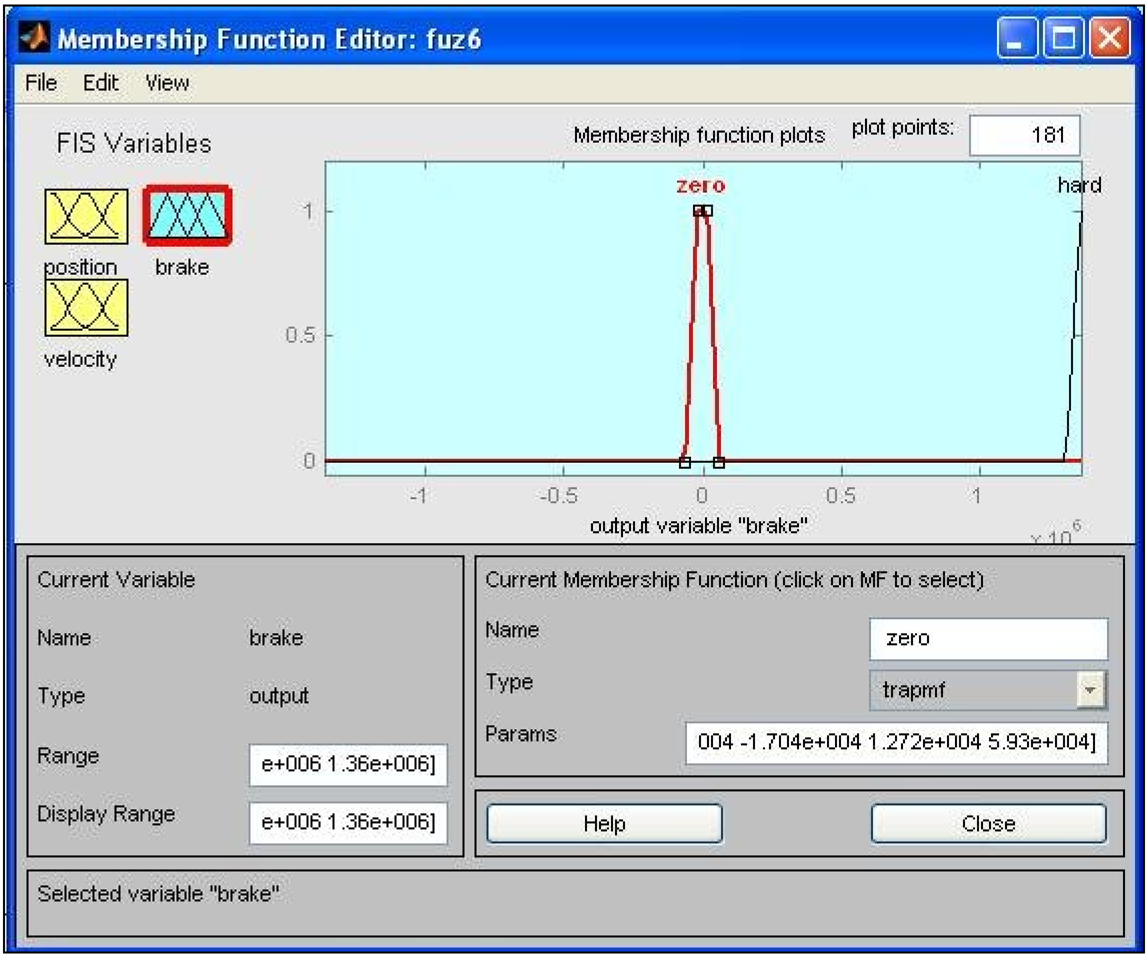


Figure 3.7 Membership Function Editor for Output Brake

3.3.2.3 Rule Editor

Rule Editor is use for editing the list of rules that defines the behavior of the system. Based on the descriptions of the input and output variables defined with the FIS Editor, the Rule Editor is used to construct the rule statements automatically, by clicking on and selecting one item in each input variable box, one item in each output box, and one connection item.

The rules that were developed are as shown in figure 3.8. It consist of four different rules which are:

I. Rule 1: If position is long and the velocity is fast, then brake is zero.
II. Rule 2: If position is long and the velocity is slow, then brake is zero
III. Rule 3: If position is short and the velocity is fast, then brake is hard
IV. Rule 4: If position is short and the velocity is slow, then brake is zero

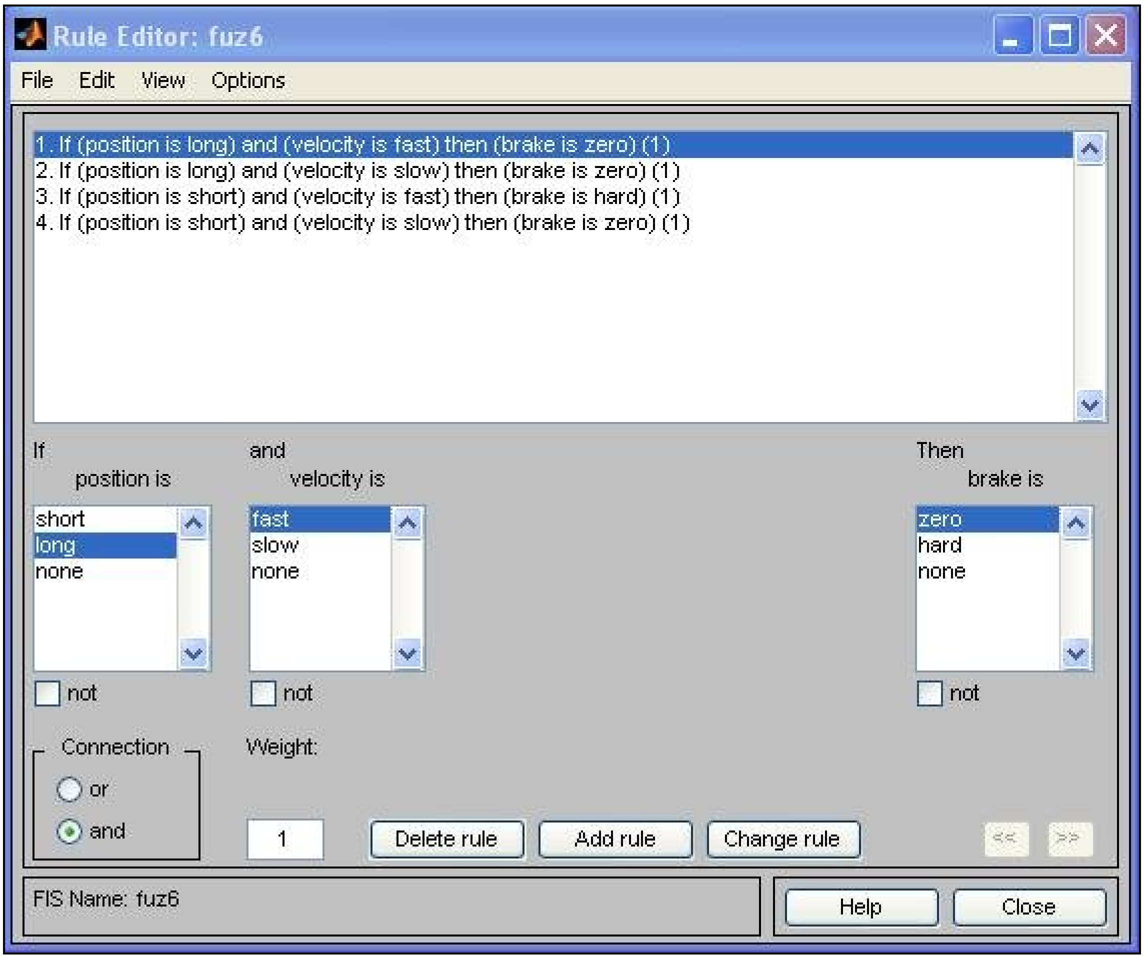


Figure 3.8 Rule Editor

3.3.2.4 Rule Viewer

The Rule Viewer displays a roadmap of the whole fuzzy inference process. Each rule is a row of plots, and each column is a variable. The rule numbers are displayed on the left of each row. The Rule Viewer interprets the entire fuzzy inference process at once. It also shows how the shape of certain membership functions influences the overall result.

Because it plots every part of every rule, it can become unwieldy for particularly large systems. But, for a relatively small number of inputs and outputs, it performs well as we can the entire rule. The Rule Viewer shows one calculation at a time and in great detail. In this sense, it presents a sort of micro view of the fuzzy inference system.

Rule 1: If position is long and the velocity is fast, then brake is zero

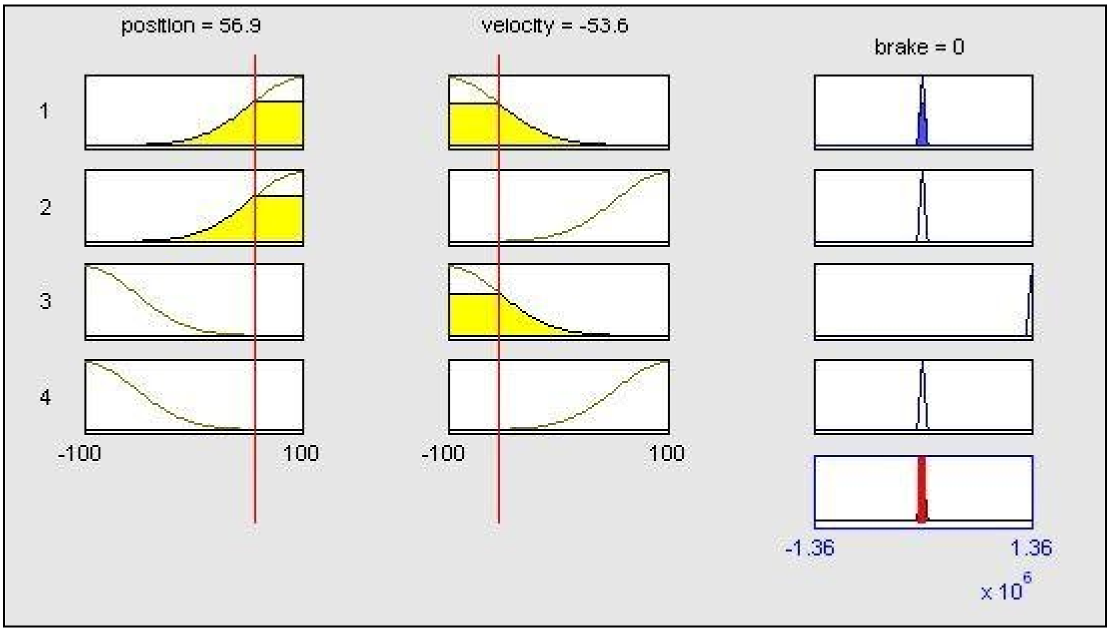


Figure 3.9 Rule Viewer with Rule 1

Rule 2: If position is long and the velocity is slow, then brake is zero

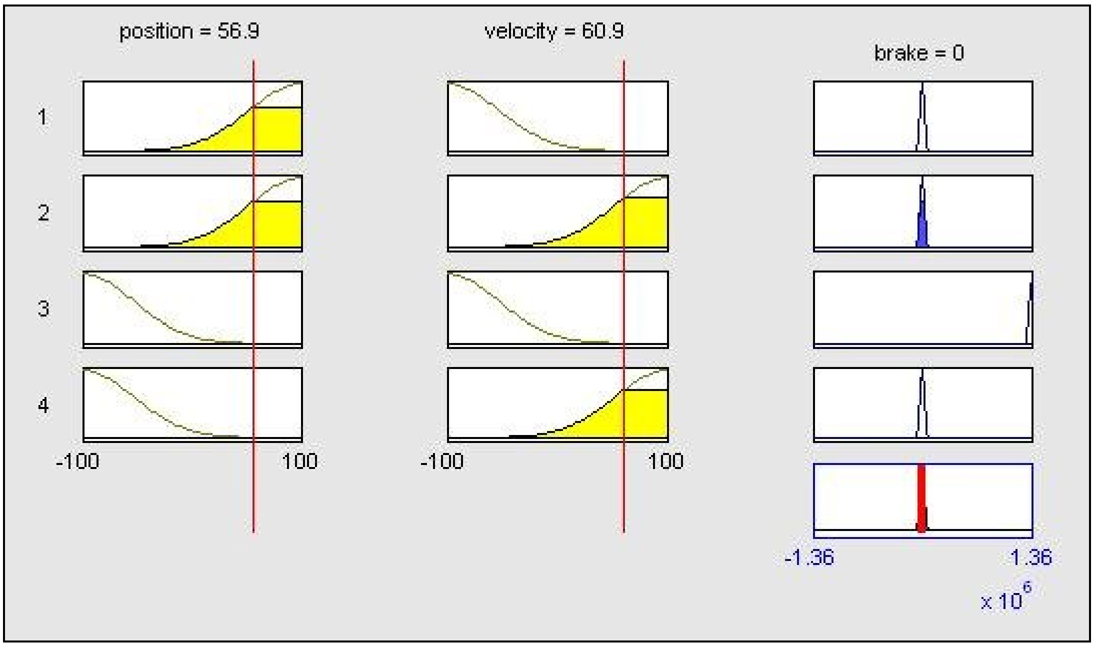


Figure 3.10 Rule Viewer with Rule 2

Rule 3: If position is short and the velocity is fast, then brake is hard

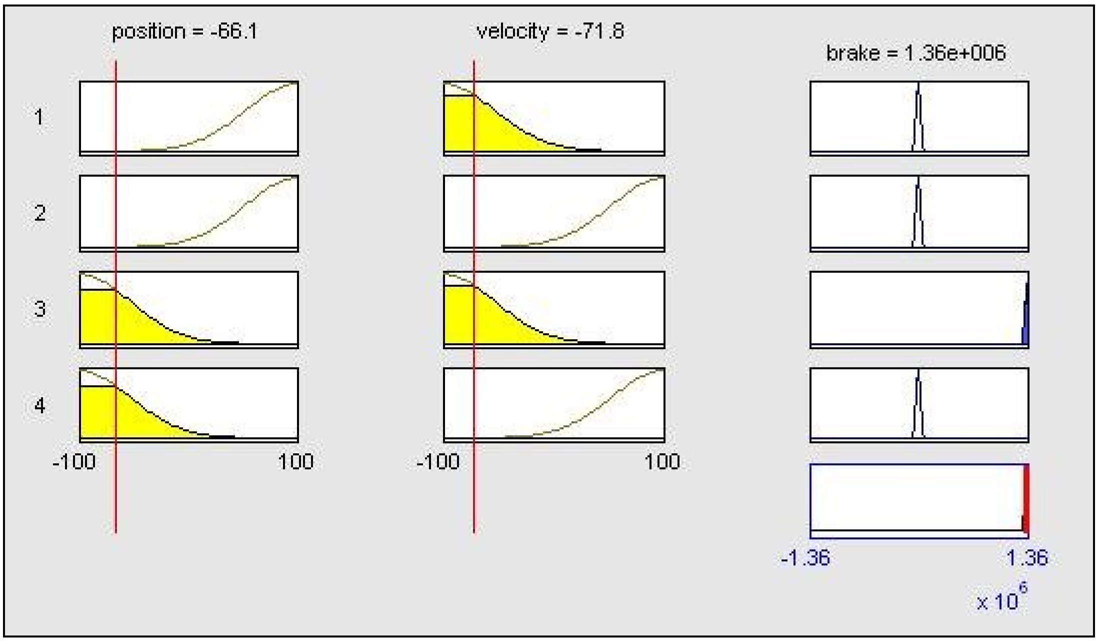


Figure 3.11 Rule Viewer with Rule 3

Rule 4: If position is short and the velocity is slow, then brake is zero

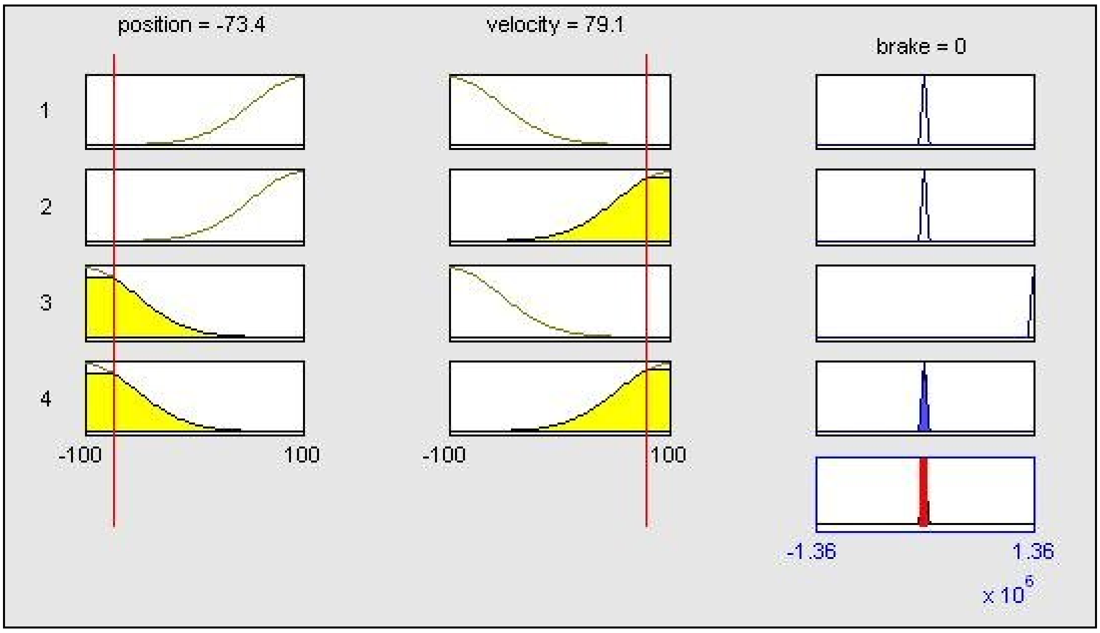


Figure 3.12 Rule Viewer with Rule 4

For each output of the Fuzzy Logic Controller, it can be simply observe by the rule viewer. Thus, correspond with the rules that are available, for braking equals to zero means 0N of brake force is applied to the system. For hard brake output, 13600N of braking force is given to the system. Hence, from the surface viewer, it is clear that for rule 1, 2 and 4 will results in 0N of braking force. As for rule 3, the brake output is 13600N.

3.3.2.5 Surface Viewer

The Surface Viewer in figure 3.13 is used to display the dependency of one of the outputs on any one or two of the inputs. The Surface Viewer is equipped with drop-down menus X (input), Y (input) and Z (output).

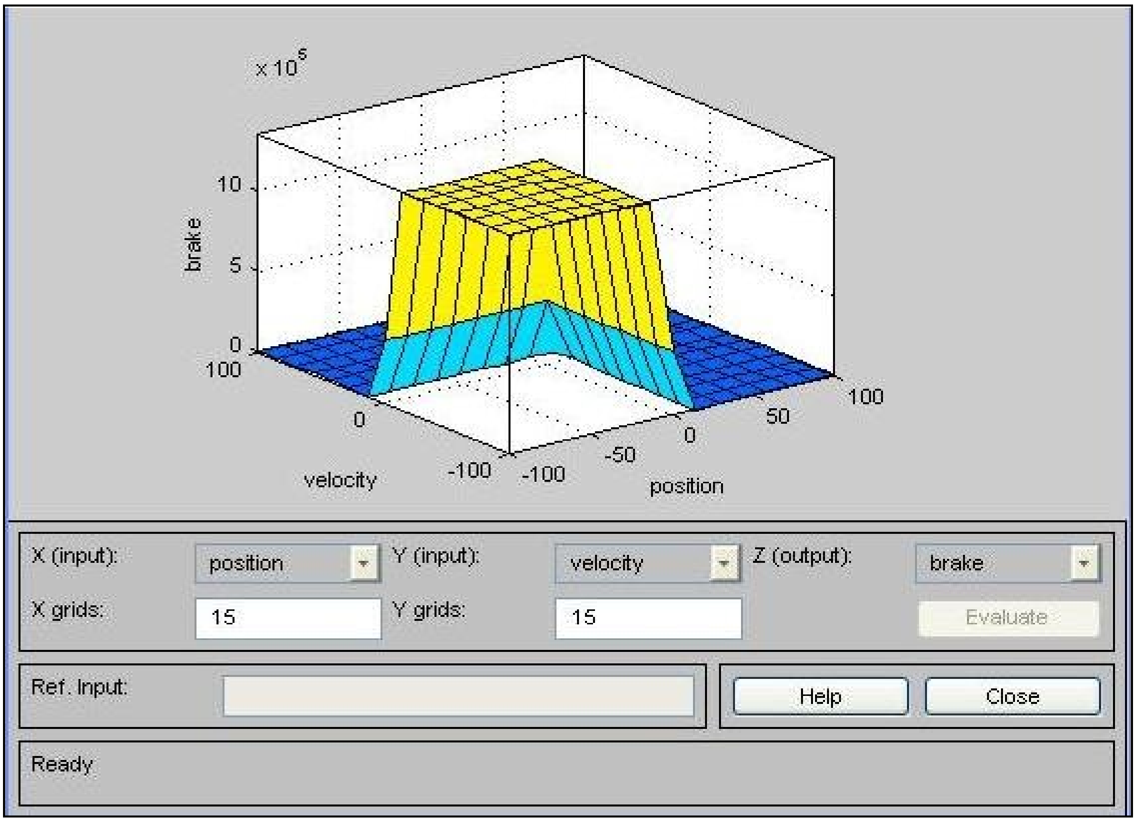


Figure 3.13 Surface Viewer

3.3.3 Fuzzy Logic Controller Simulink

To implement fuzzy inference system that were built from fuzzy logic toolbox into simulink, The Fuzzy Logic Controller block is used as a controller part to integrate the Fuzzy Logic Controller developed in the Fuzzy Logic MATLAB toolbox for the simulink model.

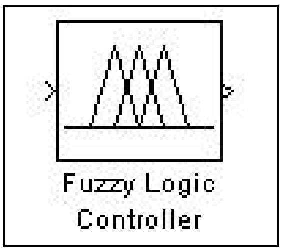


Figure 3.14 Fuzzy Logic Controller Block

Along with the simulink design, a mux, and a gain are used together with the fuzzy logic controller block in order to get an appropriate input to the fuzzy logic controller and to get a desired output of the controller together with the other blocks such as saturation and integrator.

Besides fuzzy logic controller block, fuzzy logic controller with rule viewer that is available in the simulink library can also be used in this simulation. The functions of them are nearly the same.

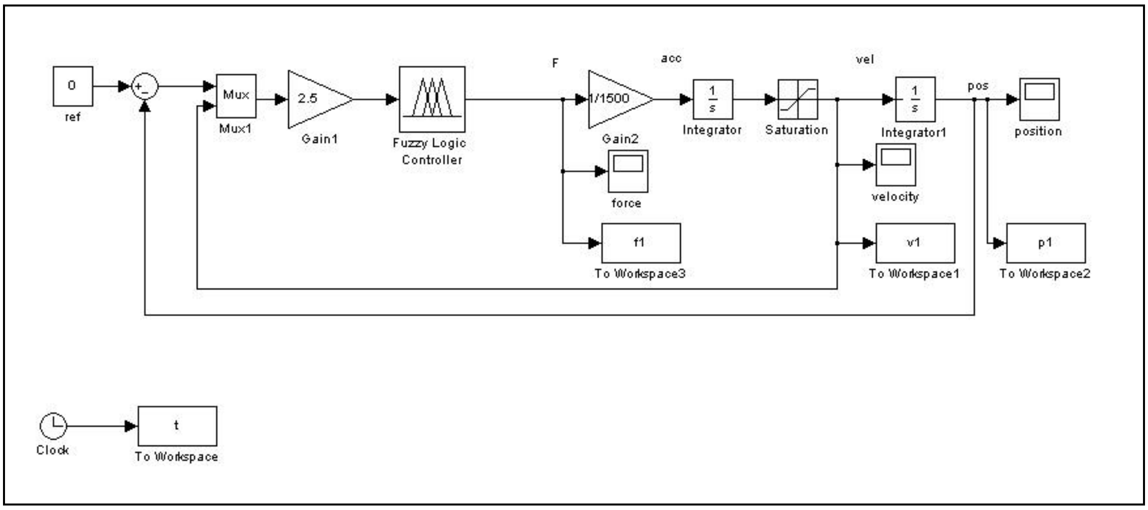


Figure 3.15 Fuzzy Logic Controller Simulink Model

CHAPTER 4

RESULTS

4.1 Introduction

The subject that is to be measured are braking force, car velocity, and a position of the car ahead of the obstacle. Hence, by the simulation done, these three elements will be observed to measure the ability and the performance for both PID and Fuzzy Logic Controller.

4.2 Fuzzy Logic Controller Simulink Result

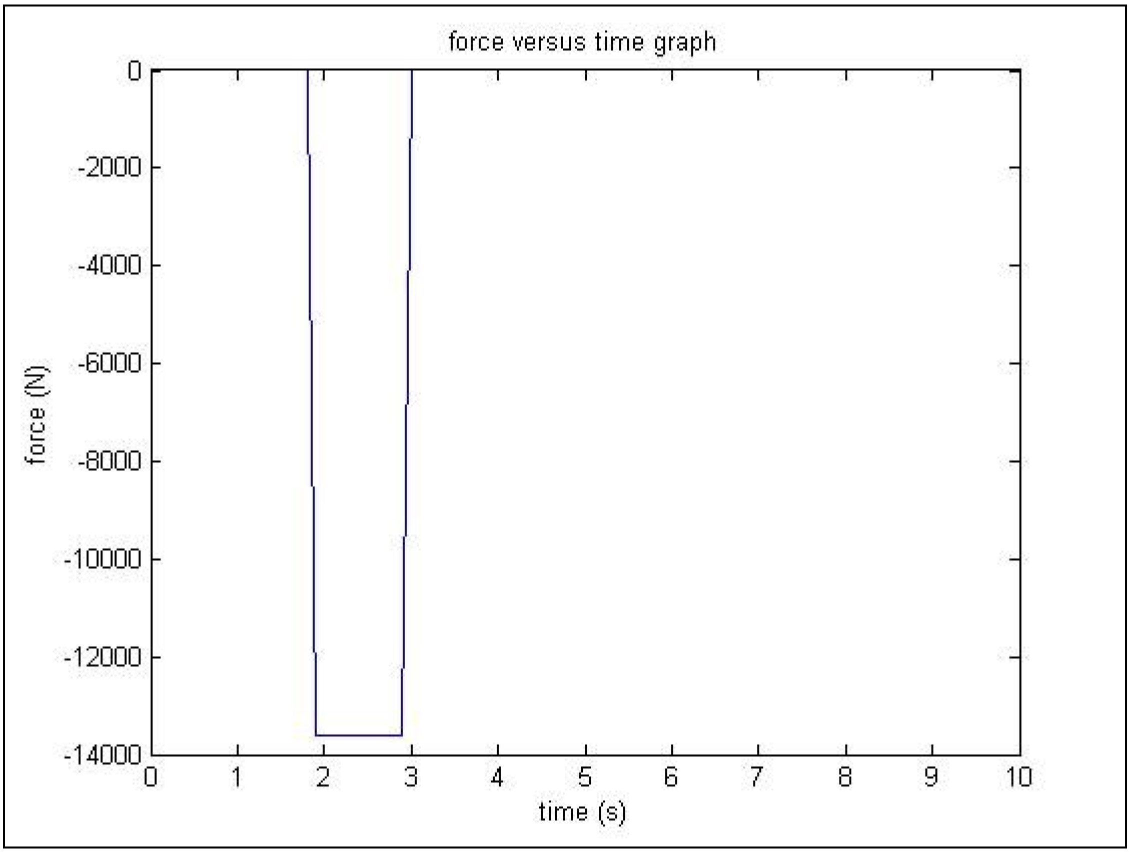


Figure 4.1 Force versus Time Graph

By looking at the resultant graph in figure 4.1, it shows that the car waits a while and started to brake the car at approximately 1.9s with 13600N braking force until 3s and the car stopped after 3s.

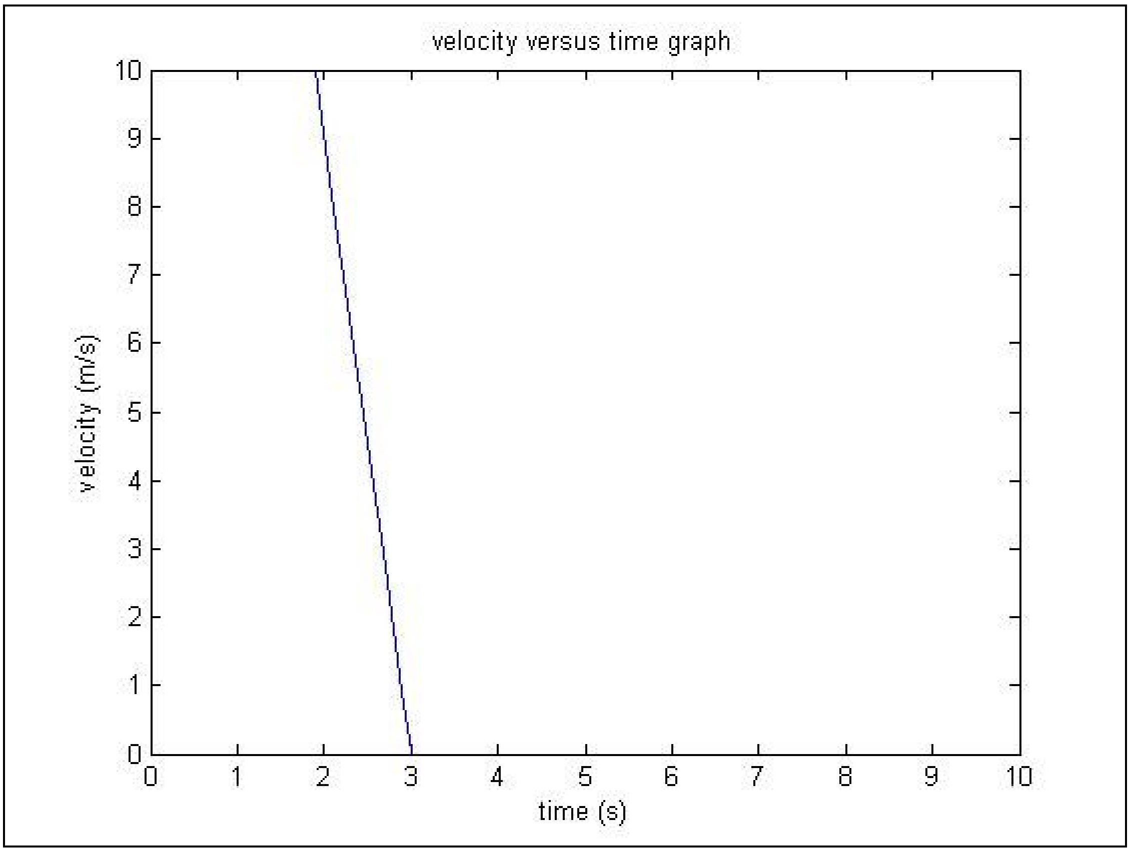


Figure 4.2 Velocity versus Time Graph

From the graph in figure 4.2, initial velocity of the car is 10m/s. The velocity is then reduce to 0m/s due to the braking force applied to the car during 1.9s until 3s.

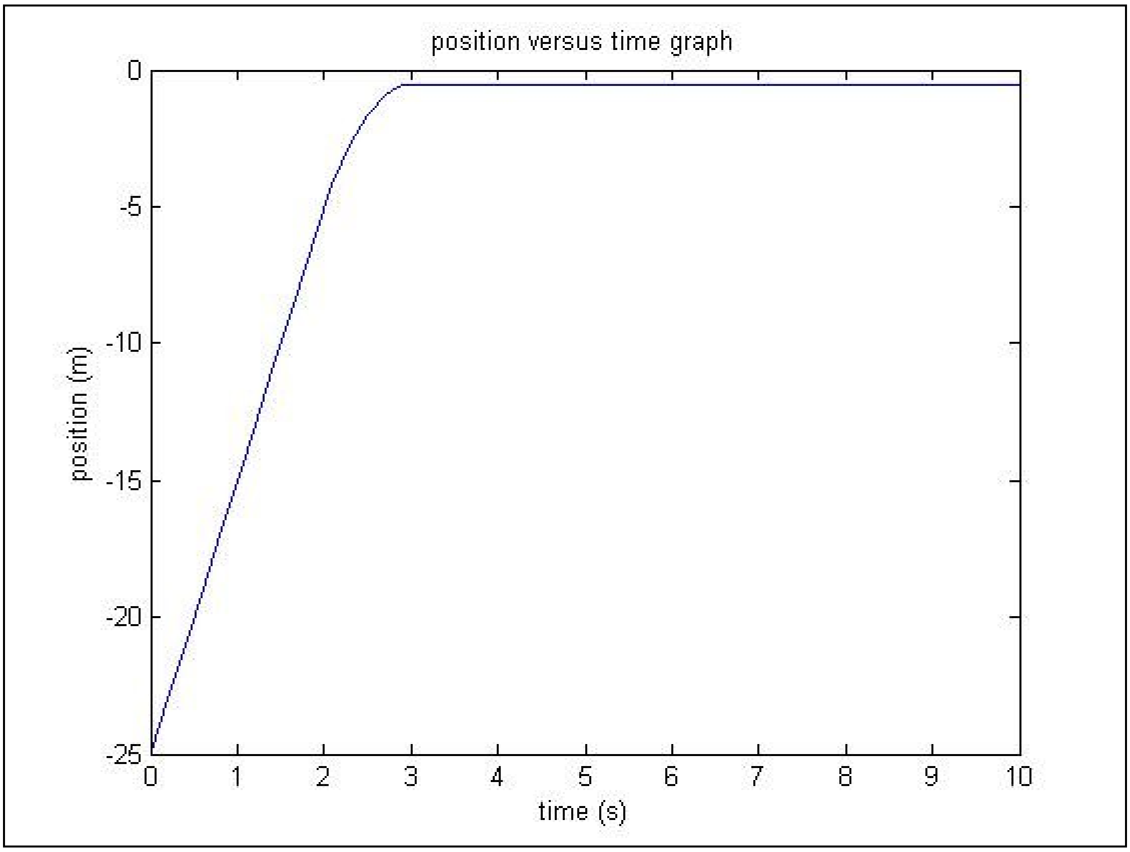


Figure 4.3 Position versus Time Graph

From the graph shown in figure 4.3, it shows that the car started to sense the obstacle at 25m distance. As the car approaching to the obstacle, the car is brake slowly according to previous graph at 1.9s, which is around 5m to 10m from the obstacle. The car stops at approximately less than 1m from the obstacle.

4.3 PID Simulink Result

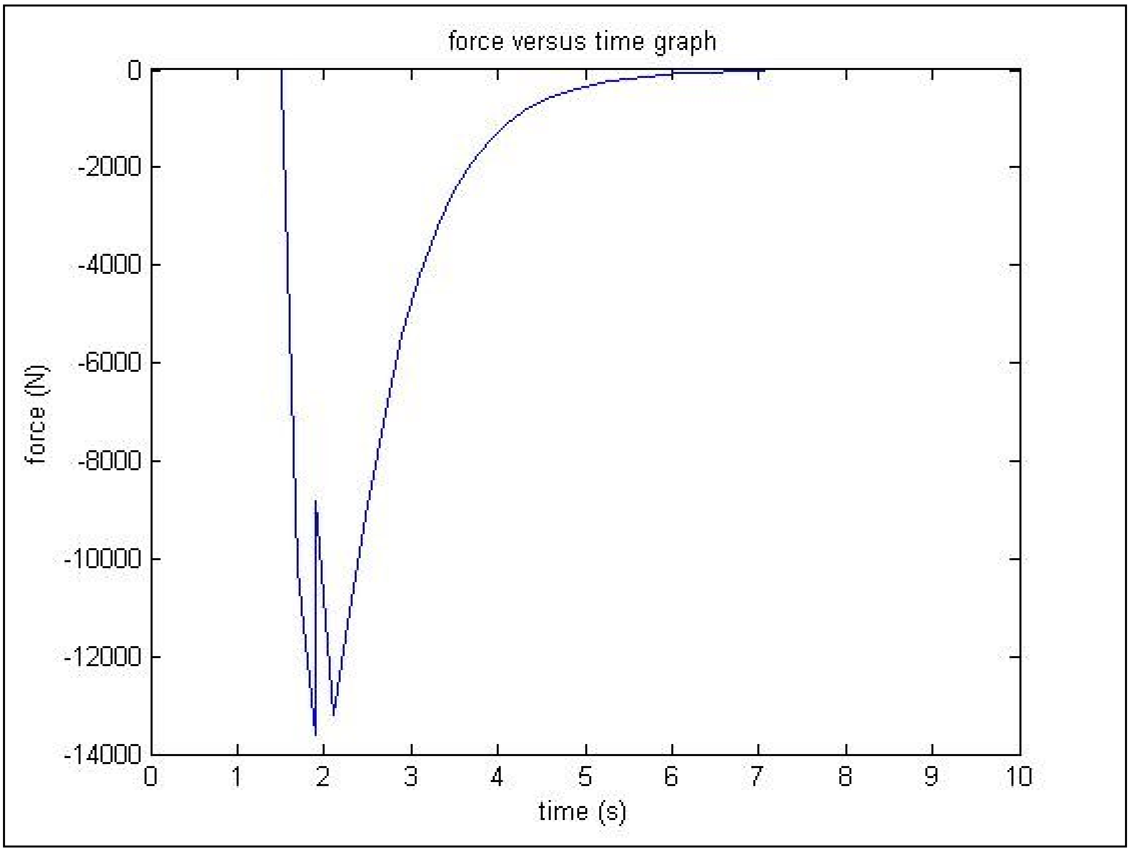


Figure 4.4 Force versus Time Graph

The figure 4.4 above shows the force versus time graph. During the first 1.5s, the control signal is zero, because the proportional action is positive and larger in magnitude than the derivative action, which is negative. Since the resultant action is positive, the saturation limits the signal to zero and the velocity remains constant. At 1.5s, the derivative action takes over and starts to brake the car. The car is sully stop at 5s.

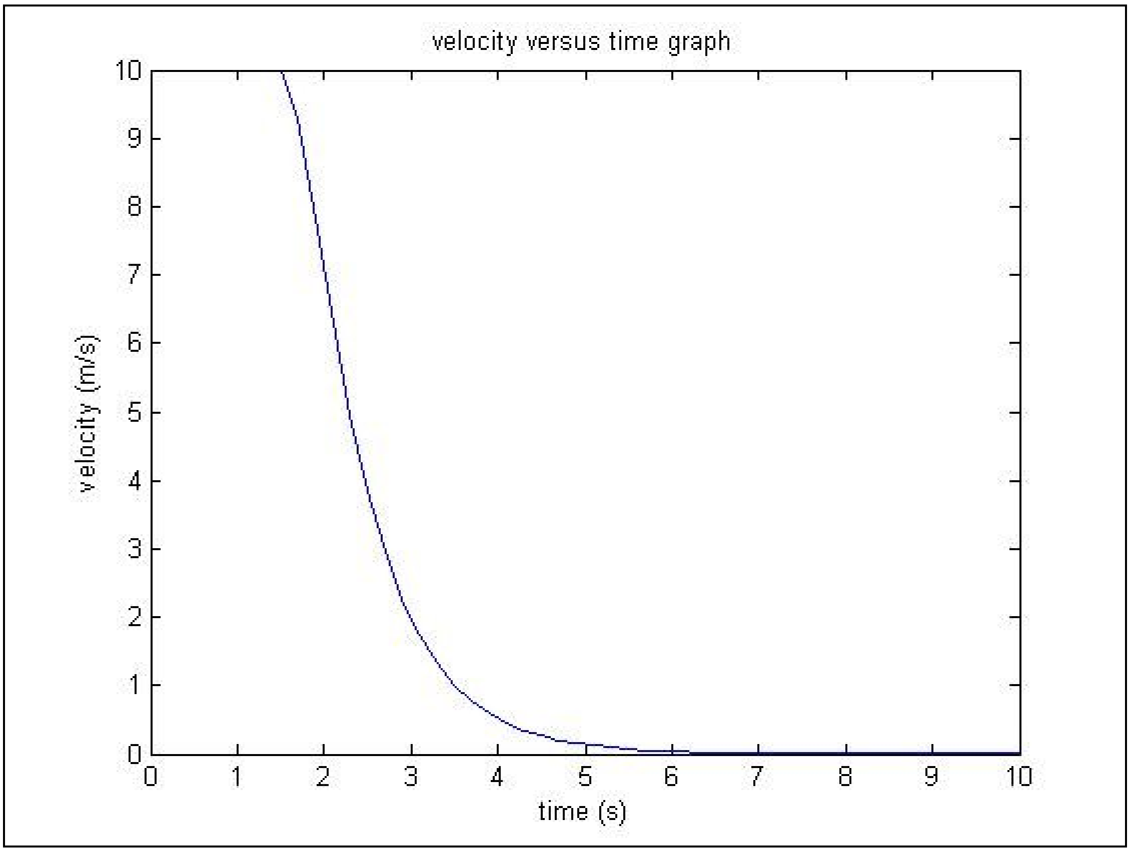


Figure 4.5 Velocity versus Time Graph

On behalf of the velocity of the car, as referring to the figure 4.5, the velocity is constantly at 10m/s during the first 1.5s by using PID controller. Afterward, when the brake is applied, the velocity is decrease continuously until the car stop and the velocity decrease to0m/s at 5s.

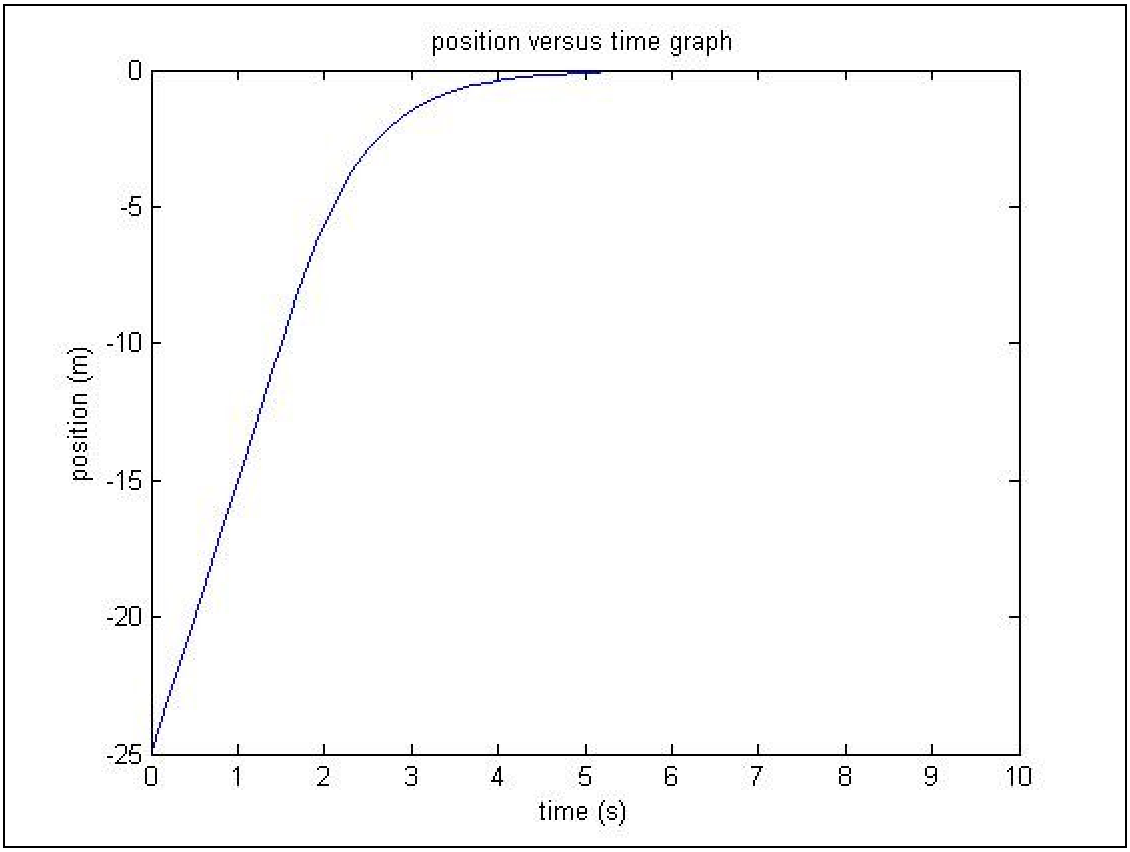


Figure 4.6 Position versus Time Graph

For the position of the car using PID controller, the car is approaching the obstacle at 25m away. Brake is applied at which is nearly 10m ahead the obstacle. Thus, the car slow down until it fully stop at 5s.

4.4 Comparisons of FLC and PID Output

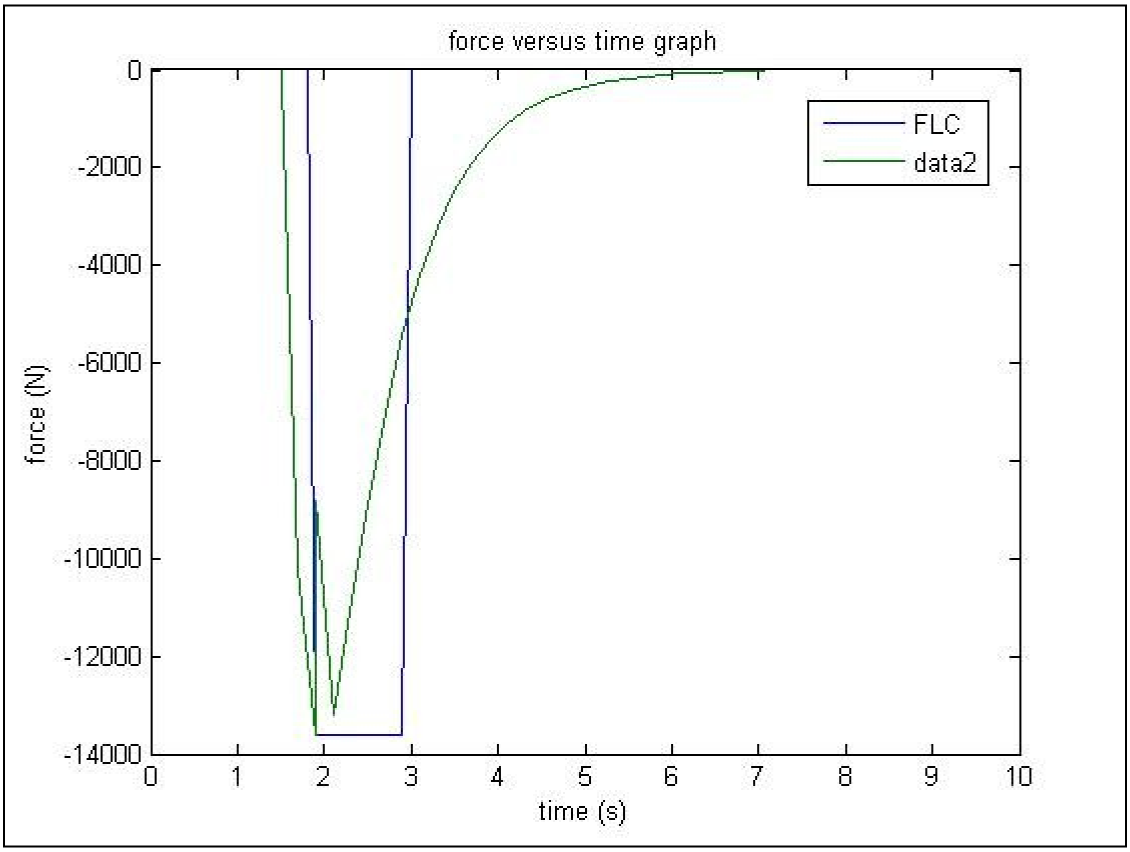


Figure 4.7 Force versus Time Graph

From the graph of figure 4.7, it is clear that both of the controllers produce different output. For PID, it started to brake the car at 1.5s while FLC brake the car at 1.9s which is a little bit slower than PID. But then, FLC uses 13600N brake force whereas PID use less than 13600N for its maximum brake force. Then, FLC are able to brake the car faster at 3s compared to PID which is at 5s.

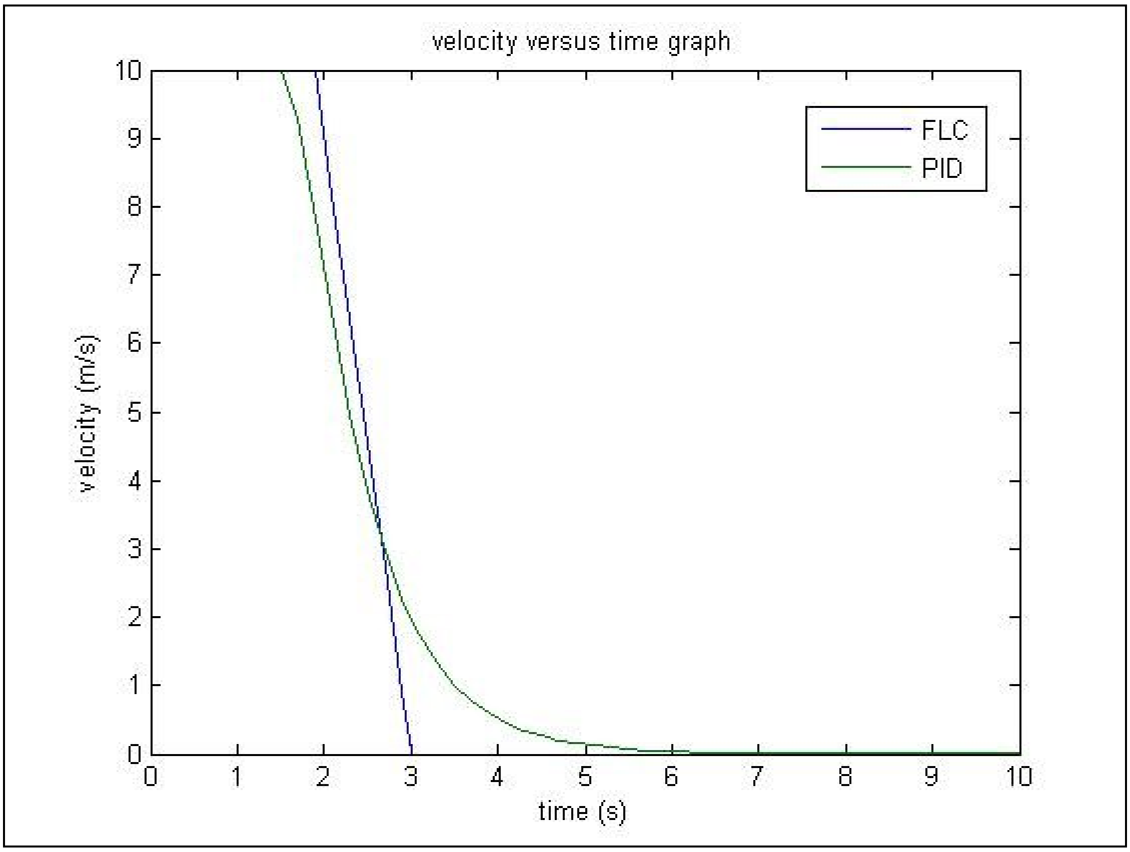


Figure 4.8 Velocity versus Time Graph

Referring to figure 4.8, velocity of both controllers decrease as the brake is applied by the controller. But, the settling time of Fuzzy Logic Controller is better than the PID. This is because the FLC are able to reach 0m/s faster than PID which enable the car to stop faster than PID controller.

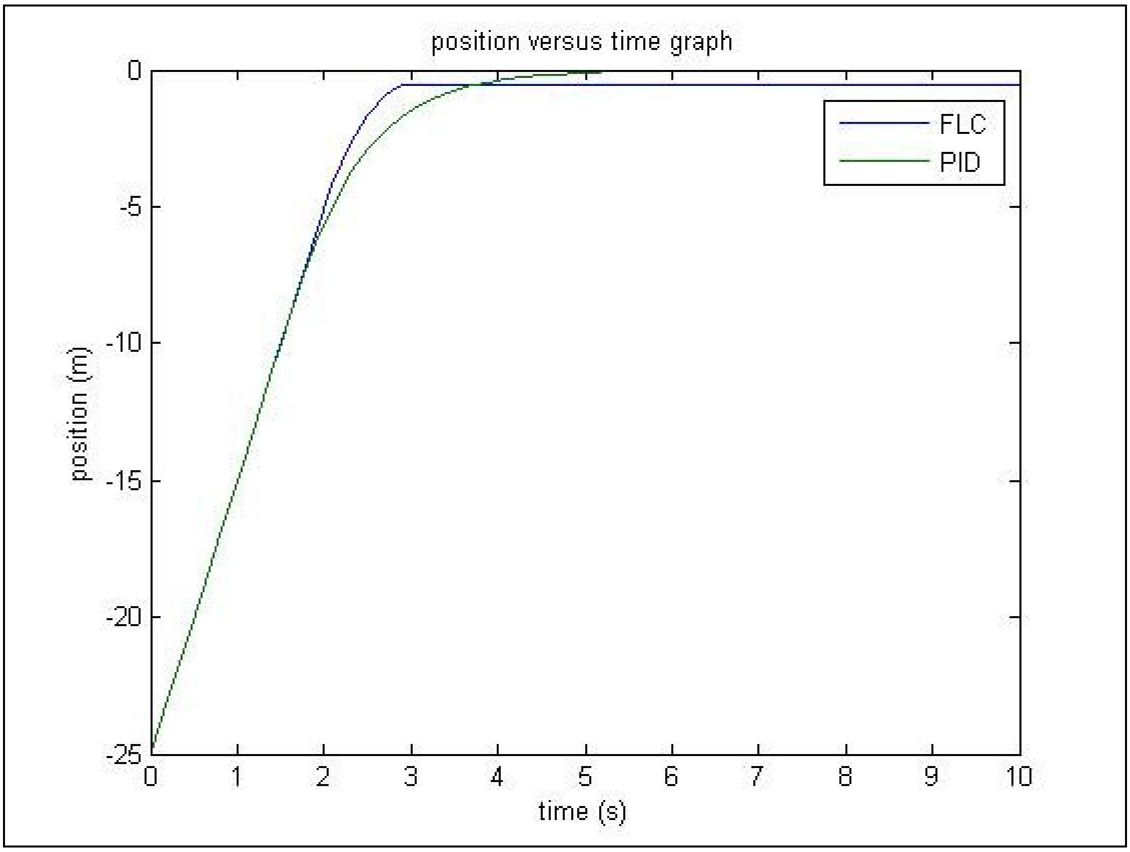


Figure 4.9 Position versus Time Graph

For the position result in figure 4.9, both of the controllers perform nearly the same. But, it is clearly seen that Fuzzy Logic Controller is better that it can brake faster before hit the obstacle at approximately 1m ahead the obstacle.

4.5 Discussion

For both position and velocity, Fuzzy Logic Controller gives better results in term of faster response and stability as compared to PID controller.

From this project, in order to develop the car braking system, there are many elements that needs to be considered. To get the system model, the mean of control should be considered. Though, factors that could effect the car brake also need to be considered such as the velocity, car mass and the load mass and also the driver behaviour. Hence, there are many difficulties regarding to succeed this project.

From the Fuzzy Logic Controller itself, there are many challenging steps to be take in order to develop the system. When it is integrated with the MATLAB Simulink, the things that should be considered is how well the system could meet the project objective. The problem regarding the controllers is to maintain the stability of the output result. Thus, a certain ‘try and error’ procedure are done to develop the system that really could achieve the main objective of the automated car braking system.

According to the theoretical aspect, the Fuzzy Logic Controller should be better than the PID controllers. However, by developing the PID controller itself, it helps to design the Fuzzy Logic Controller better. By transferring PID data into rules, a good Fuzzy Logic Controller could be develop as the PID design helps to provide the earlier stage of designing the Fuzzy Logic Controller.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Introduction

Conclusion and recommendation are made after the analyses of the result are done. The conclusion involve the summary of the project taken. Then, recommendations are prepared for future progress and improvement of the projects done.

5.2 Conclusion

By comparing both of the controllers that have been simulated, it can be said that Fuzzy Logic Controller response are better than the PID response. Fuzzy Logic Controller is able to brake the car faster than PID controller. Moreover, Fuzzy Logic Controller is able to brake the car from quickly in a short time due to the obstacle.

Still, from this project, it can be seen that to develop a Fuzzy Logic Controller, less mathematical expression and calculation are needed. From the toolbox, we can develop a controller which use our logic thought to transform it into rules that will integrate with the other method to develop and design the required controller.

On the other hand, to develop PID controller, it require more mathematical solution that can somehow results in difficulties to design a more difficult and complex controller. This is why the usage of Fuzzy Logic Controller is better than the PID controller. However, the PID controller constitutes a reference for the assessment of the performance of the fuzzy controller.

5.3 Recommendation

For a future development of this project, the Fuzzy Logic Controller designed can be enhanced by applying more rules. By then, it can produce better response. By applying more rules, it means that more parameters and measurement should be considered. Also, it is recommended that the PID controller had to be integrated with Fuzzy Logic Controller to produce better controller.

5.4 Commercialization

For the past few years, there are many car brake development that uses the involvement of the electronic roles such as the Intelligent Cruise Control (ICC), Antilock Braking Systems (ABS), Traction Control System (TCS) and the Sensotronic Brake Control (SBC). Therefore, by the results of this project, it is seen that this project has the potential to be enhance and commercialize in the automotive industry.

REFERENCES

[1] Cai, L., Rad, A,. B., Chan, W., L., Ho, M., L., *A Neural-Fuzzy Controller for Intelligent Cruise Control of Vehicle in Highways*
Available at: http://www.ieee.org

[2] 23th March 2007, Citing Internet Sources URL http://www.auto-innovation.com/actu-news/

[3] Jost, K. (2003). *Bosch Expands Brake Offering with Electronics.* Automotive Engineering Online Article
Available at: http://www.sae.org/automag/electronics/09-2003/

[4] Muller, R., Nocker, G., *Intelligent Cruise Control with Fuzzy Logic.*
Available at: http://www.ieee.org

[5] Simon, D., (2003). *Embedded System Programming: Introduction to Fuzzy Control.*
Available at: http://www.academic.csuohio.edu/simond/fuzzyintro

[6] Altrock, V., C., *Fuzzy Logic in Automative Engineering: Fuzzy Application Library.*
Available at: http://www.fuzzytech.com/e/e_a_esw.html

[7] Ahmad, M., I., (2004). *Fuzzy Logic for Embedded System Application* : Elsevier Science (USA).

[8] Naranjo, J., E., Reviejo, J., Gonzalez, C., Garcia, R., Pedro, T., D., (2003). *A Throttle and Brake Fuzzy Controller: Towards the Automatic Car*: Springer Berlin/ Heidelberg.

[9] Siler, W., Buckley, J., J., (2005). *Fuzzy Expert System and Fuzzy Reasoning*: John Wiley and Sons Inc.

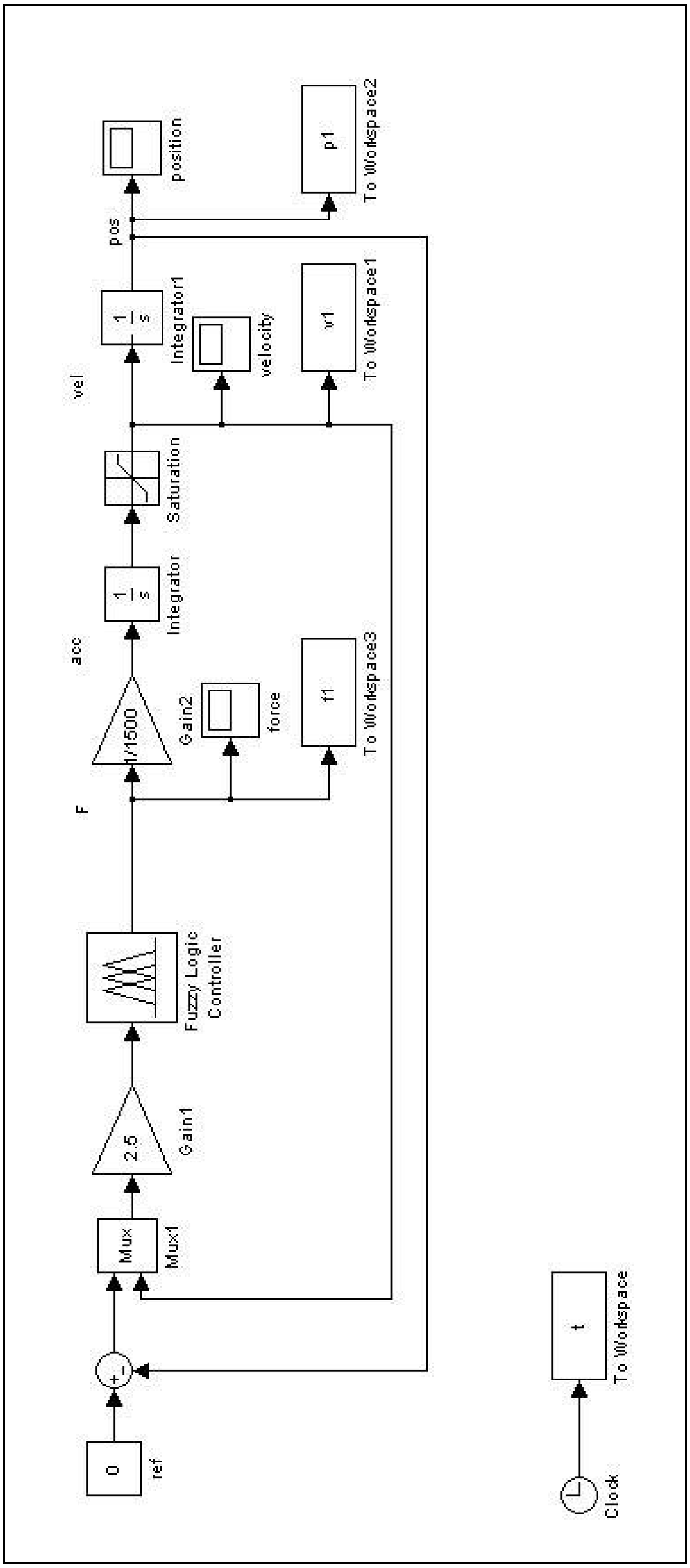
[10] (2006), *Fuzzy Logic Toolbox for Use with MATLAB*: The Math Work Inc

[11] 13th January 2007, Citing Internet Sources URL
http://en.wikipedia.org/wiki/PID_controller

[12] 6th February 2005, Citing Internet Sources URL
http://www.expertune.com/tutor.html

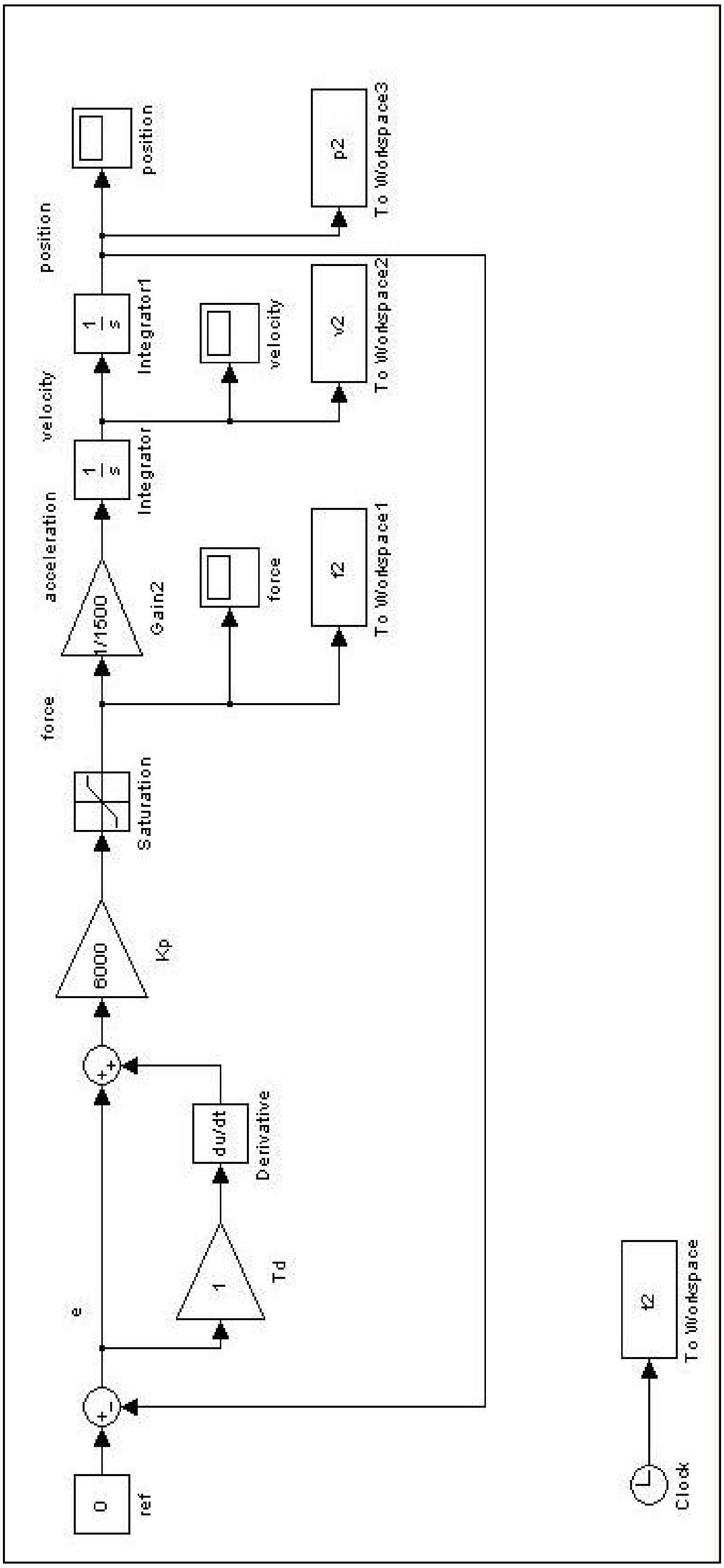
APPENDIX A

FUZZY LOGIC CONTROLLER (FLC) SIMULINK BLOCK DIAGRAM


0
ref
Mux
Mux1
2.5
Gain1
Fuzzy Logic Controller
F
1/1500
Gain2
force
f1
To Workspace3
acc
1/s
Integrator
Saturation
vel
1/s
Integrator1
velocity
v1
To Workspace1
pos
position
p1
To Workspace2
t
To Workspace
Clock

APPENDIX B

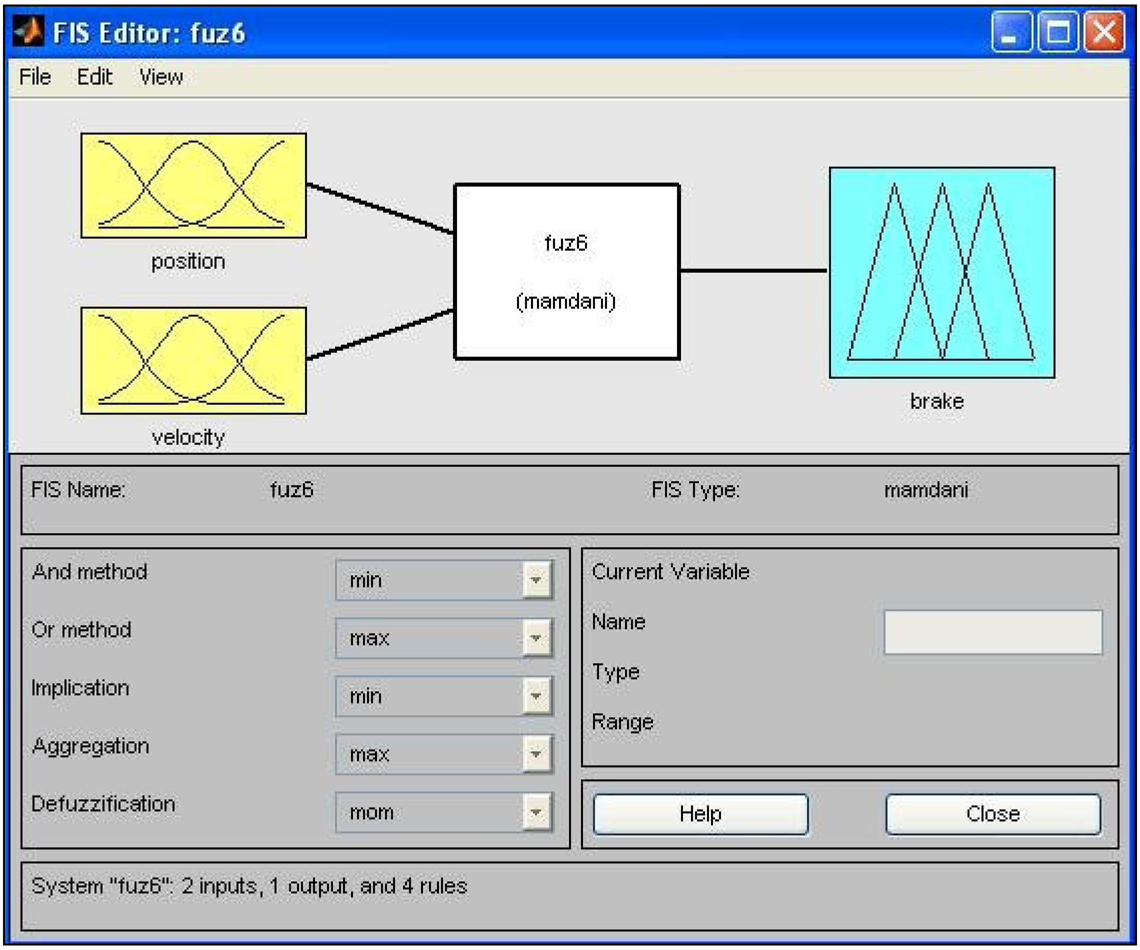
PROPORTIONAL INTEGRAL DERIVATIVE (PID) SIMULINK BLOCK DIAGRAM


0
ref
e
1
Td
du/dt
Derivative
6000
Kp
Saturation
force
1/1500
Gain2
acceleration
Integrator
velocity
Integrator1
position
position
velocity
force
f2
To Workspace1
v2
To Workspace2
p2
To Workspace3
Clock
t2
To Workspace

APPENDIX C

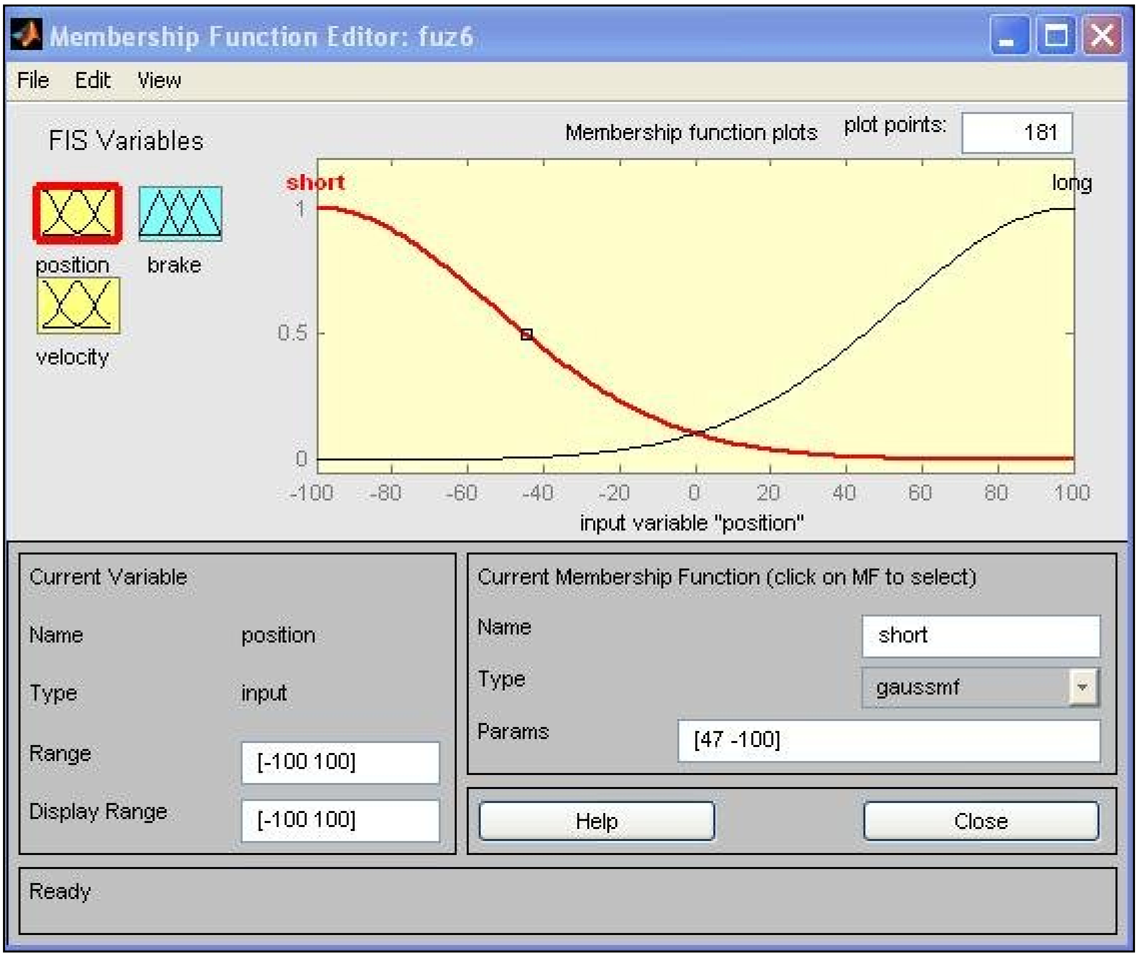
FUZZY LOGIC CONTROLLER TOOLBOX

FIS EDITOR

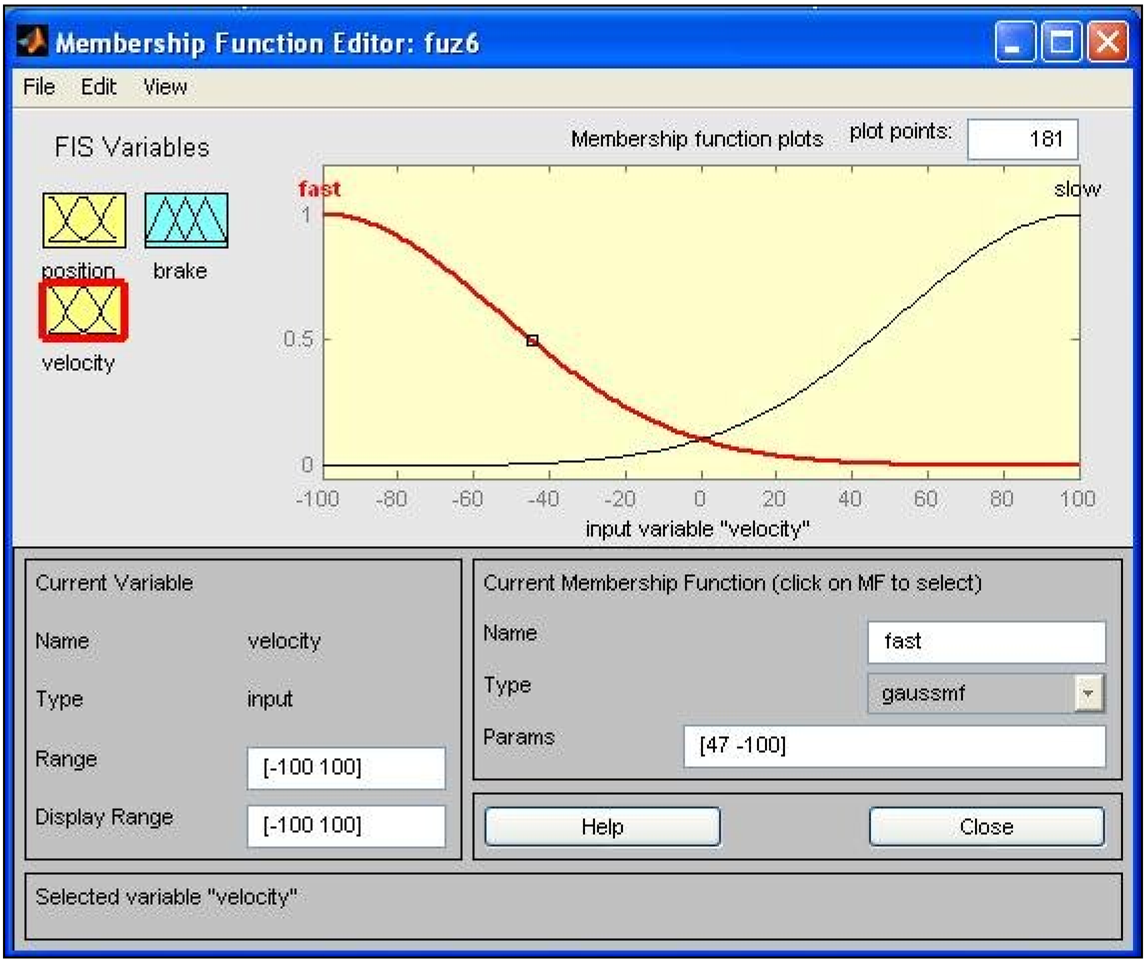

FIS Editor: fuz6
File
Edit
View
position
velocity
fuz6
(mamdani)
brake
FIS Name:
fuz6
FIS Type:
mamdani
And method
min
Or method
max
Implication
min
Aggregation
max
Defuzzification
mom
Current Variable
Name
Type
Range
Help
Close
System "fuz6": 2 inputs, 1 output, and 4 rules

MEMBERSHIP FUNCTION EDITOR

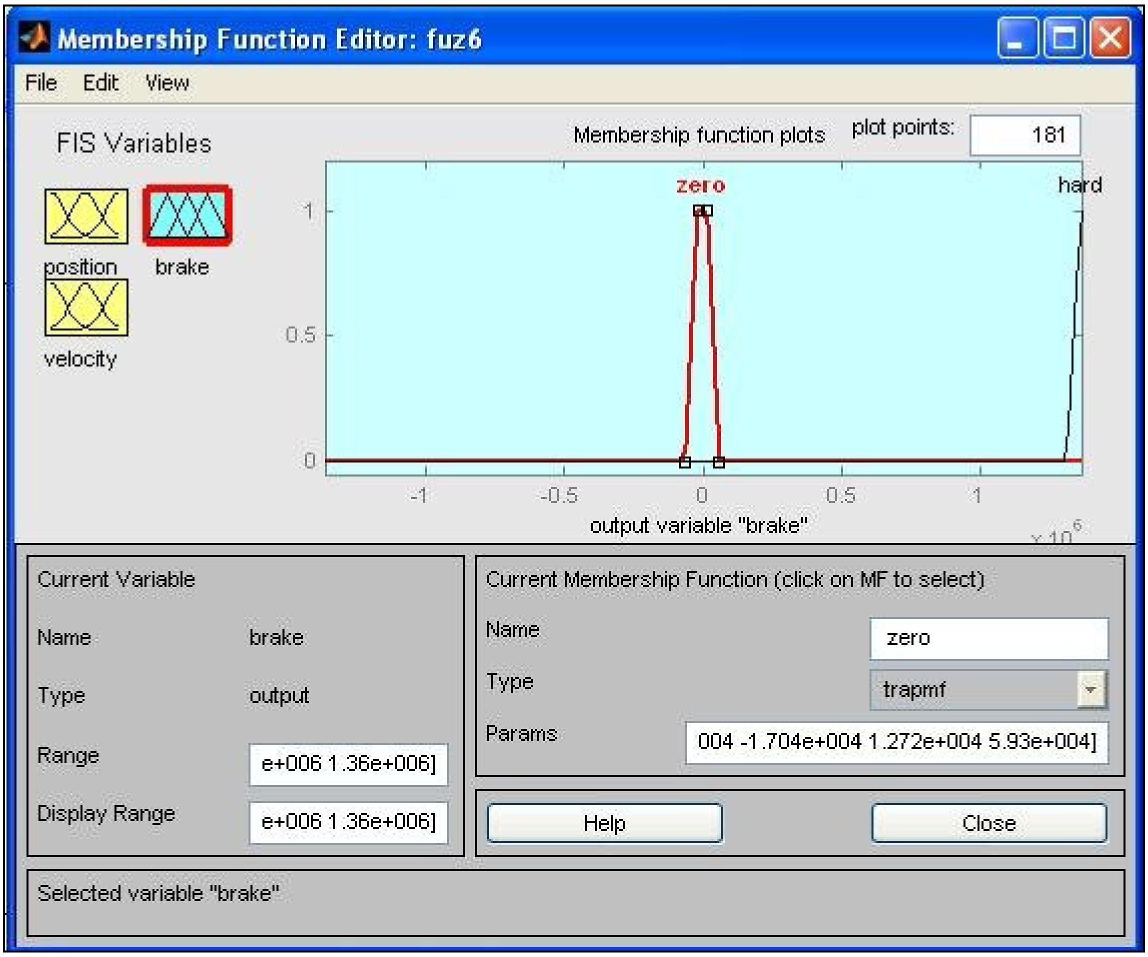
a) Input 1: position



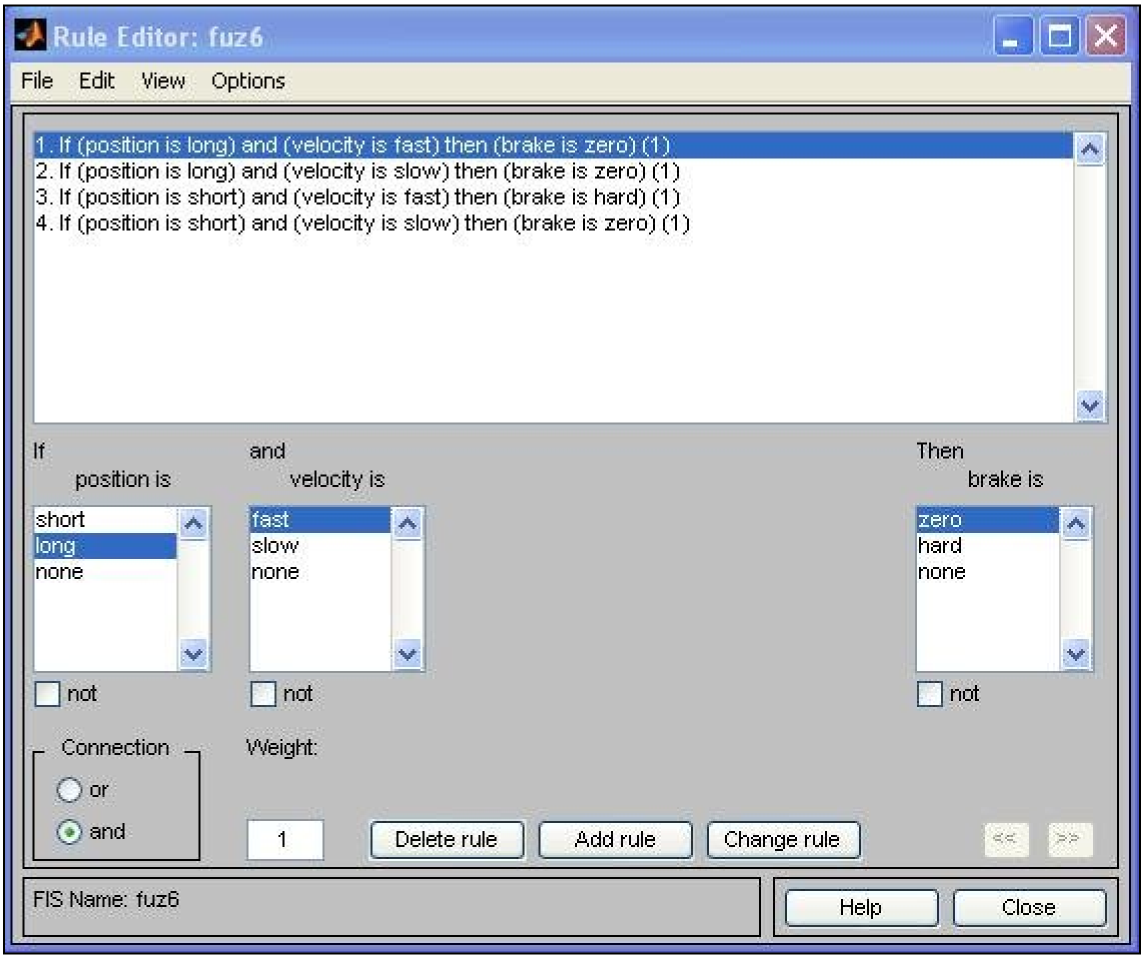
b) Input 2: velocity



c) Output: force

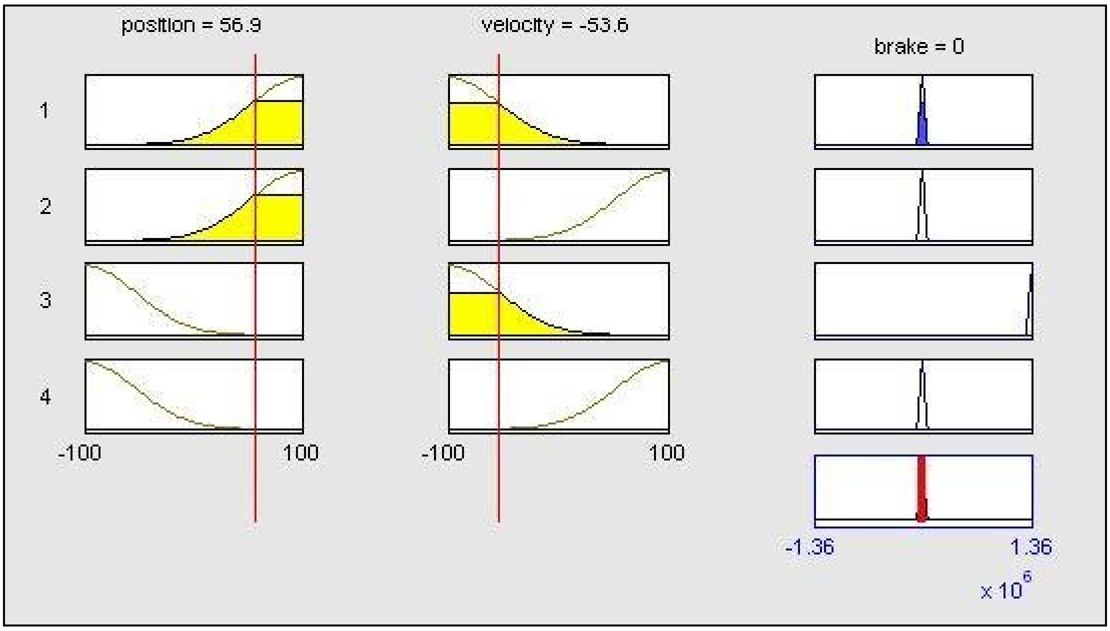


RULE EDITOR

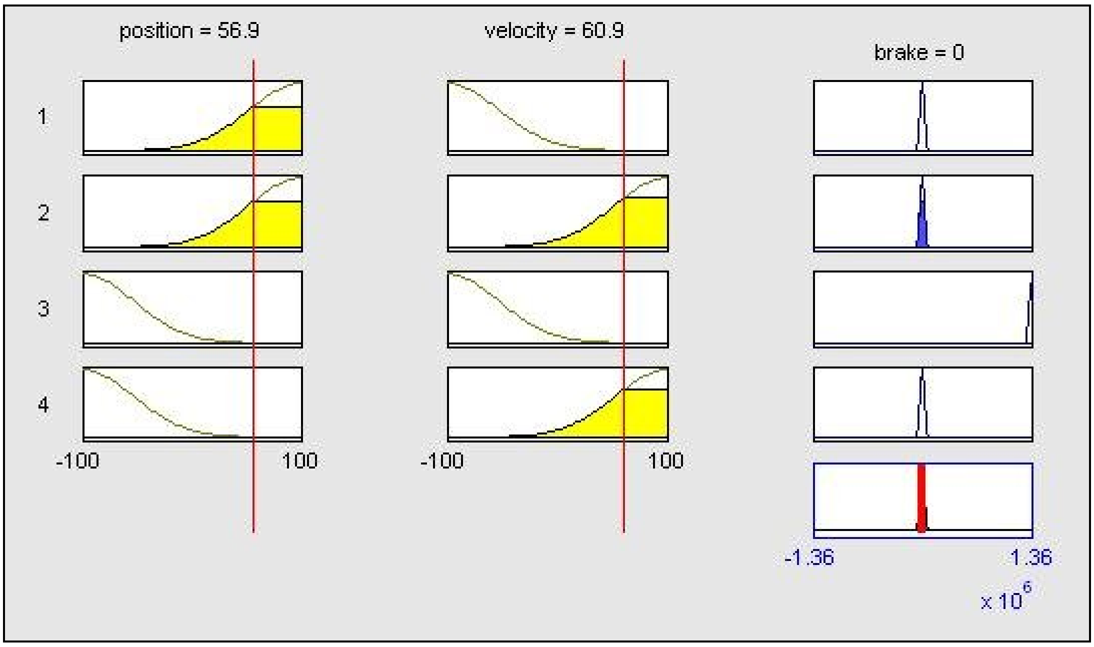

Rule Editor: fuz6
File Edit View Options
1. If (position is long) and (velocity is fast) then (brake is zero) (1)
2. If (position is long) and (velocity is slow) then (brake is zero) (1)
3. If (position is short) and (velocity is fast) then (brake is hard) (1)
4. If (position is short) and (velocity is slow) then (brake is zero) (1)
If
position is
short
long
none
not
and
velocity is
fast
slow
none
not
Then
brake is
zero
hard
none
not
Connection
or
and
Weight:
1
Delete rule
Add rule
Change rule
FIS Name: fuz6
Help
Close

RULE VIEWER

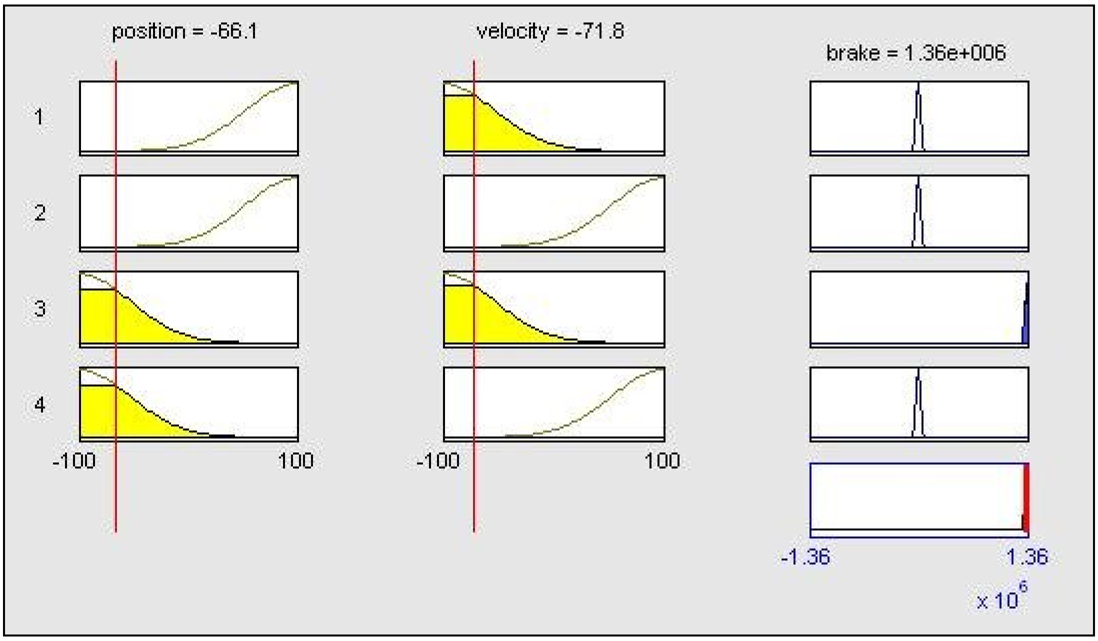
Rule 1: If position is long and the velocity is fast, then brake is zero



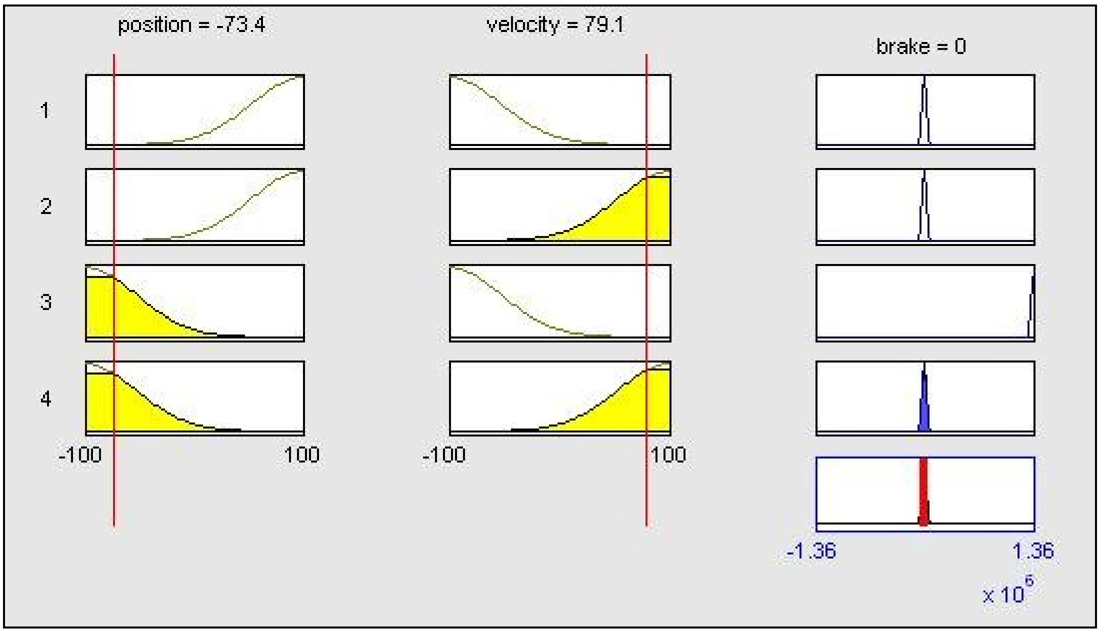
Rule 2: If position is long and the velocity is slow, then brake is zero



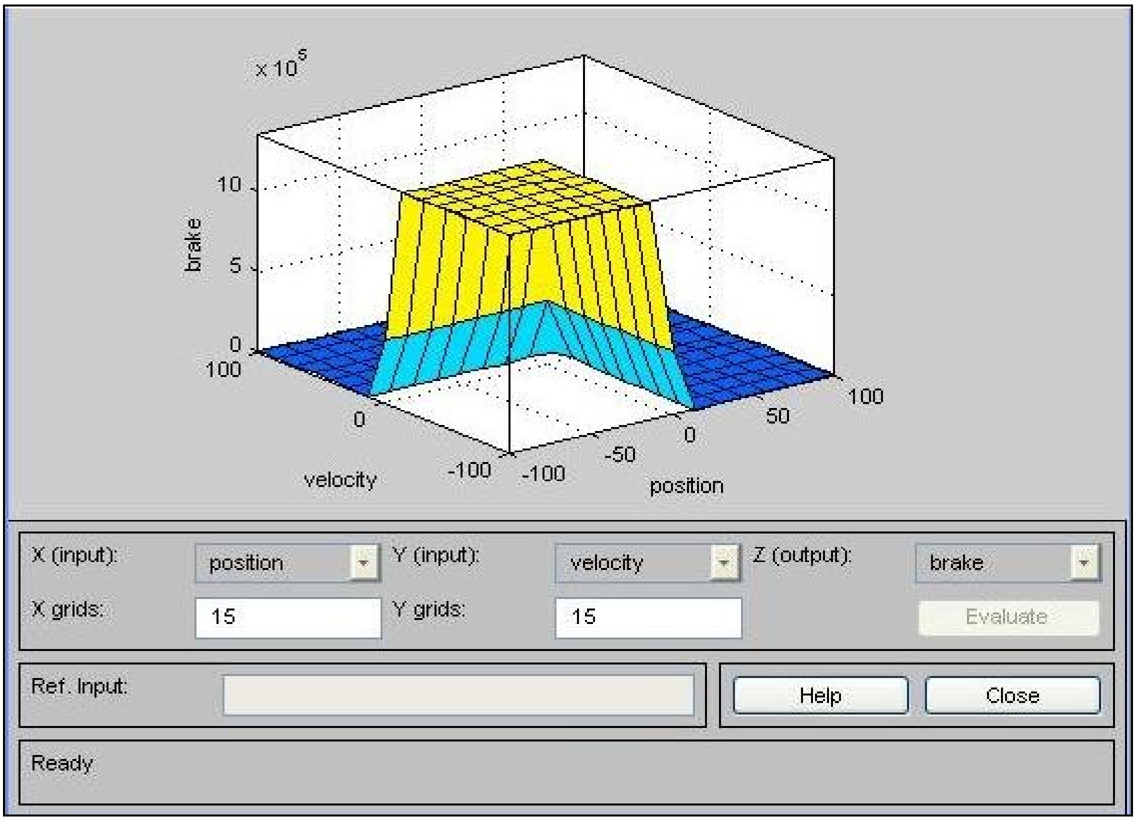
Rule 3: If position is short and the velocity is fast, then brake is hard



Rule 4: If position is short and the velocity is slow, then brake is zero



SURFACE VIEWER



APPENDIX D

FUZZY LOGIC CONTROLLER M-FILE

FUZZY LOGIC CONTROLLER M-FILE

```
[System]
Name='fuz6'
Type='mamdani'
Version=2.0
NumInputs=2
NumOutputs=1
NumRules=4
AndMethod='min'
OrMethod='max'
ImpMethod='min'
AggMethod='max'
DefuzzMethod='mom'

[Input1]
Name='position'
Range=[-100 100]
NumMFs=2
MF1='short':'gaussmf',[47 -100]
MF2='long':'gaussmf',[47 100]

[Input2]
Name='velocity'
Range=[-100 100]
NumMFs=2
MF1='fast':'gaussmf',[47 -100]
MF2='slow':'gaussmf',[47 100]

[Output1]
Name='brake'
Range=[-1360000 1360000]
NumMFs=2
```

MF1='zero':'trapmf',[-63620 -17040 12720 59300]

MF2='hard':'trapmf',[1308000 1360000 1394000 1447000]

[Rules]

2 1, 1 (1) : 1

2 2, 1 (1) : 1

1 1, 2 (1) : 1

1 2, 1 (1) : 1

APPENDIX E

MATLAB M-FILE COMMAND FOR OUTPUT GRAPH

I) Fuzzy Logic Controller simulink output graph

```
plot (t,f1)        %graph for force%
plot (t,v1)        %graph for velocity%
plot (t,p1)        %graph for position%
```

II) PID simulink output graph

```
plot (t2,f2)       %graph for force%
plot (t2,v2)       %graph for velocity%
plot (t2,p2)       %graph for position%
```

III) Combination of Fuzzy Logic Controller and PID simulink output graph

```
plot (t,f1,t2,f2)    %graph for force%
plot (t,v1,t2,v2)    %graph for velocity%
plot (t,p1,t2,p2)    %graph for position%
```