

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

## **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.

## **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 tuelles 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

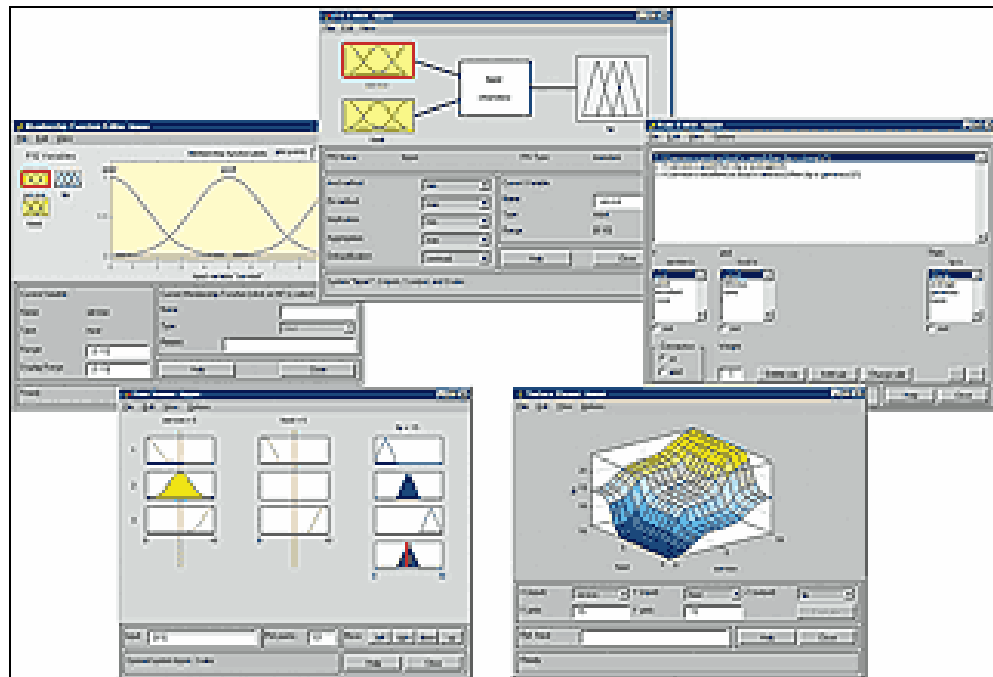
Although both make use of fuzzy sets, they differ qualitatively in methodology. [9] Fuzzy control comprises the steps of sense, preprocess, fuzzify, evaluate, activate, aggregate, defuzzify and act. However, difficulty occurs with the using of fuzzy logic system. Usually it is difficult to determine 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 inference 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]

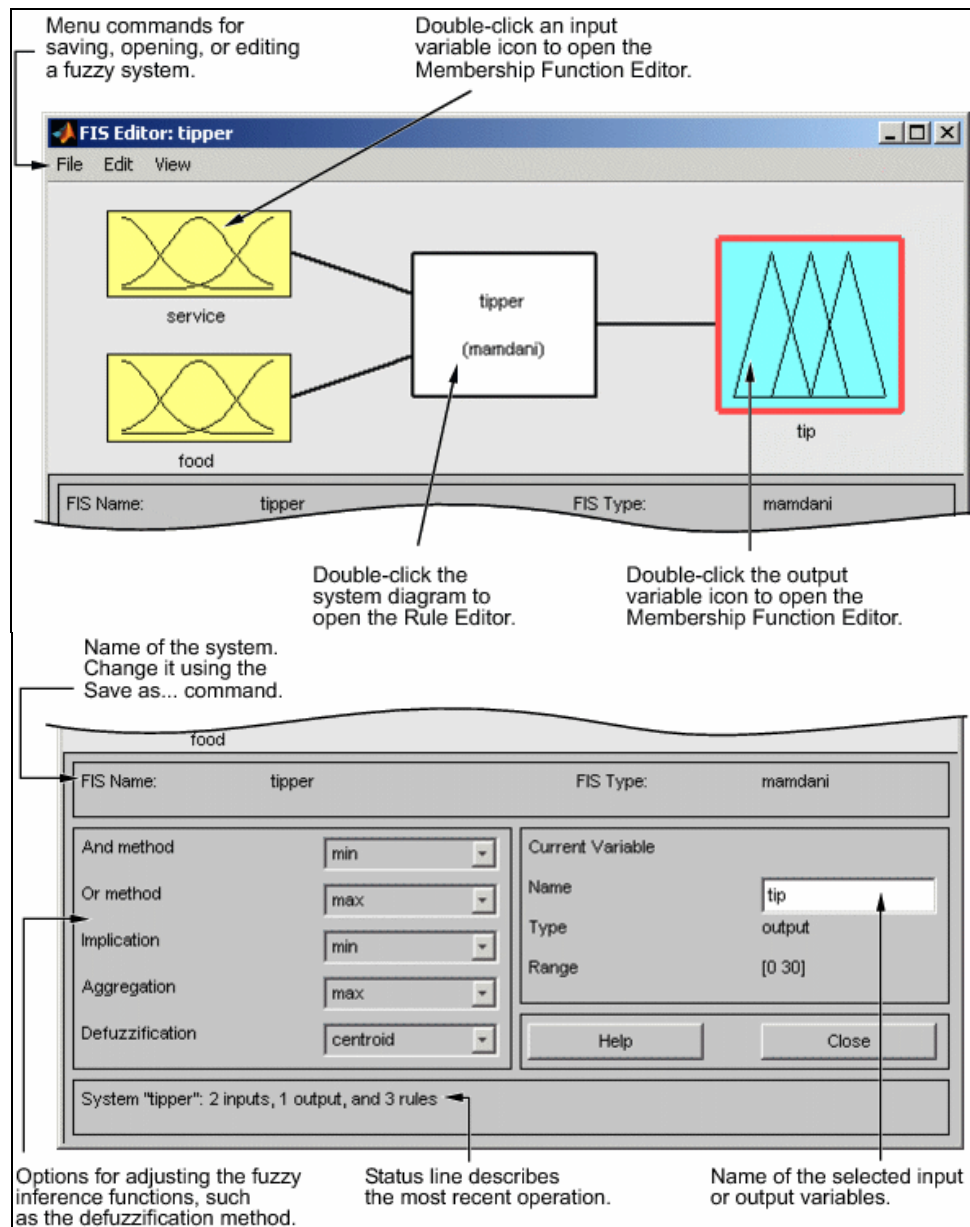
Although it is possible to use Fuzzy Logic Toolbox by working strictly from the command line, in general it is much easier to build a system graphically. [10] There are five primary GUI tools for building, editing, and observing fuzzy inference systems in Fuzzy Logic Toolbox:

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

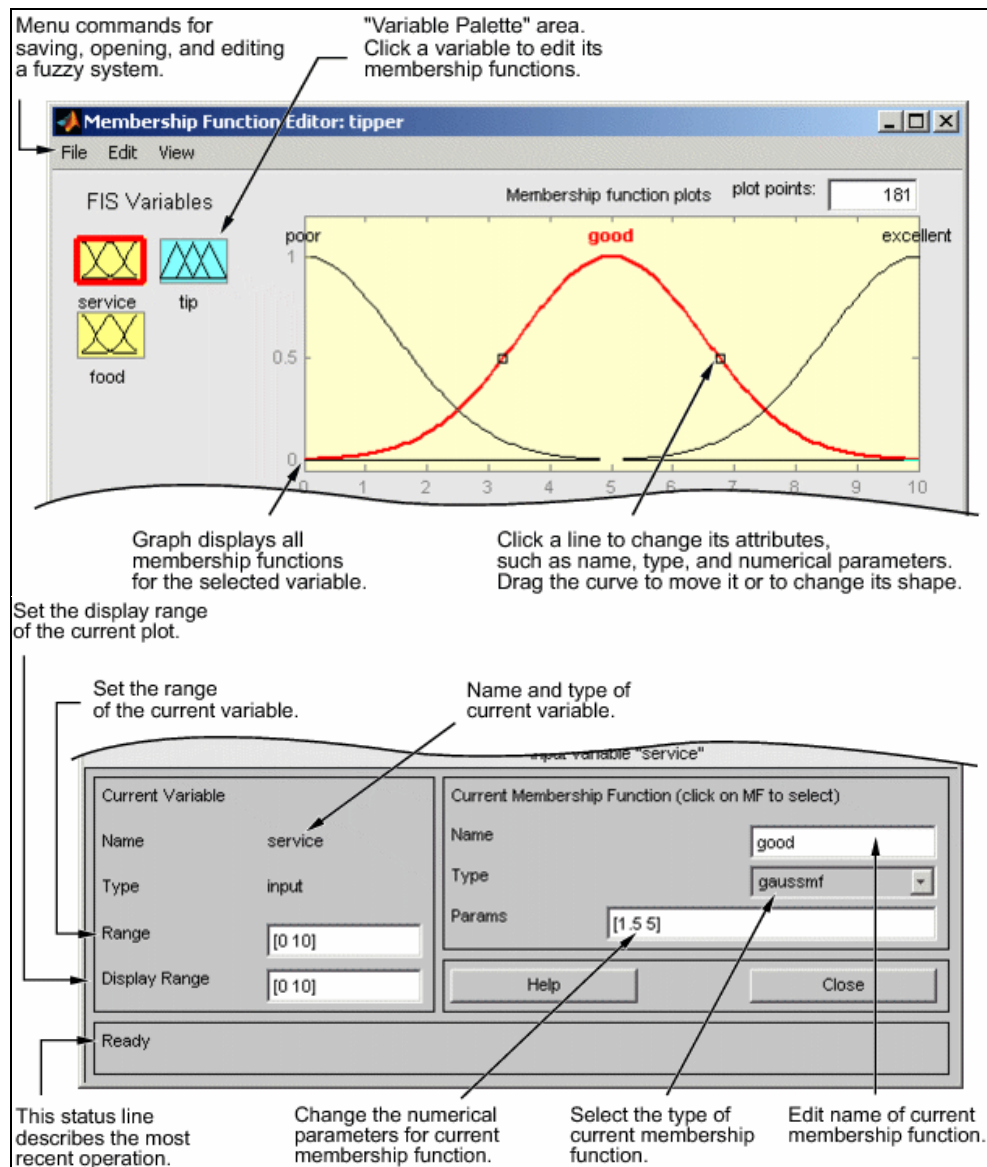




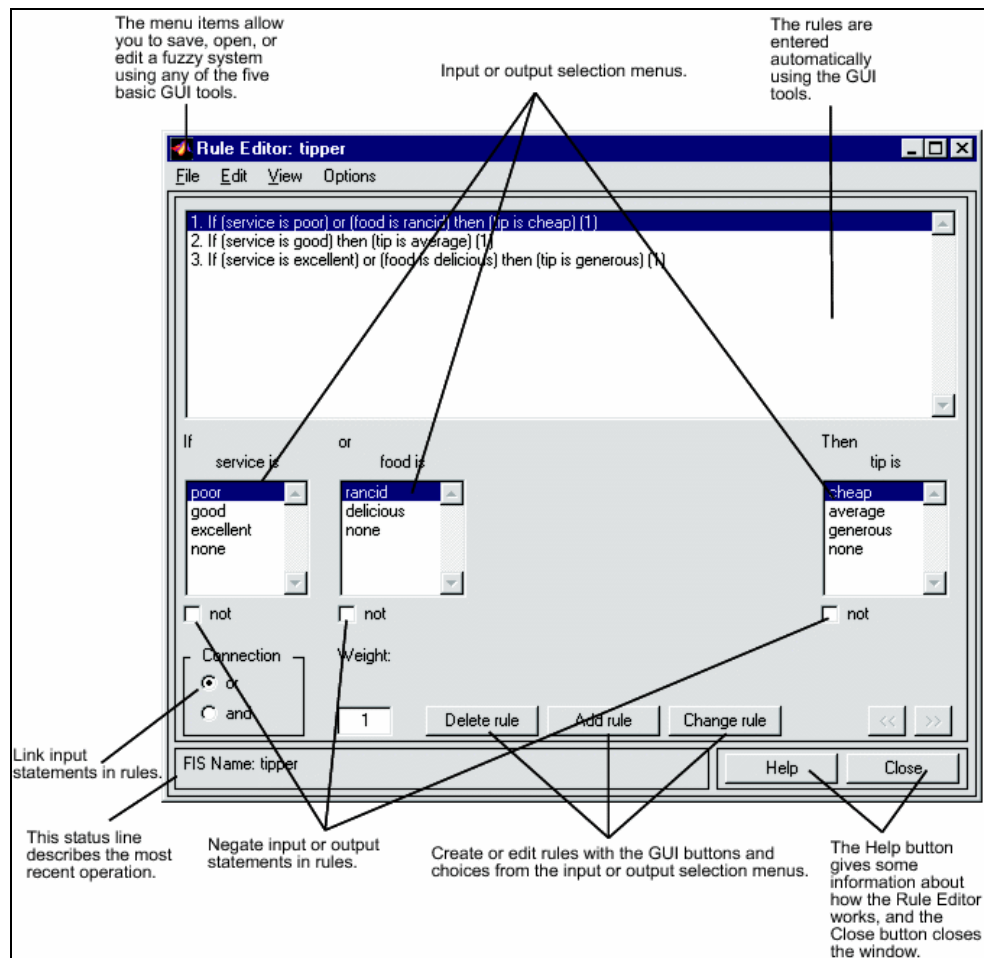
**Figure 1.1** FIS Editor Toolbox



**Figure 1.2** FIS Editor in MATLAB Toolbox



**Figure 1.3** Membership Function Editor in MATLAB Toolbox



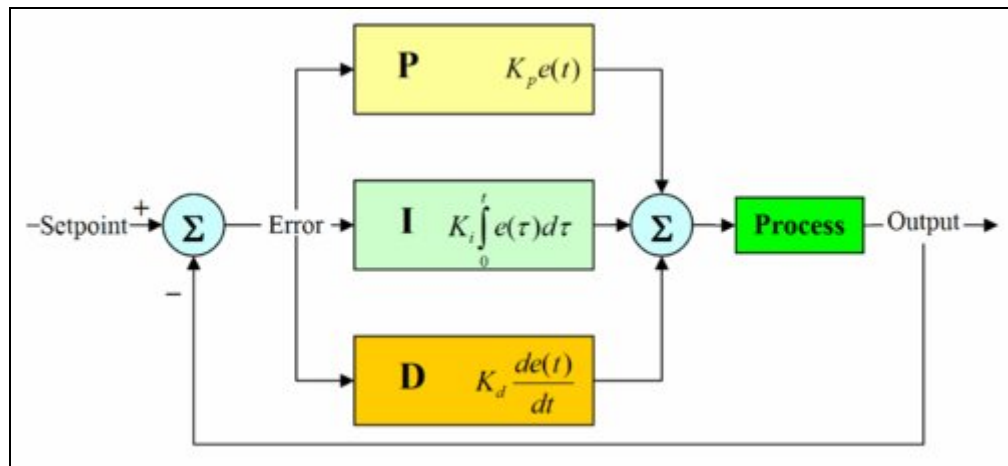
**Figure 1.4** Rule Editor in MATLAB Toolbox

### 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]



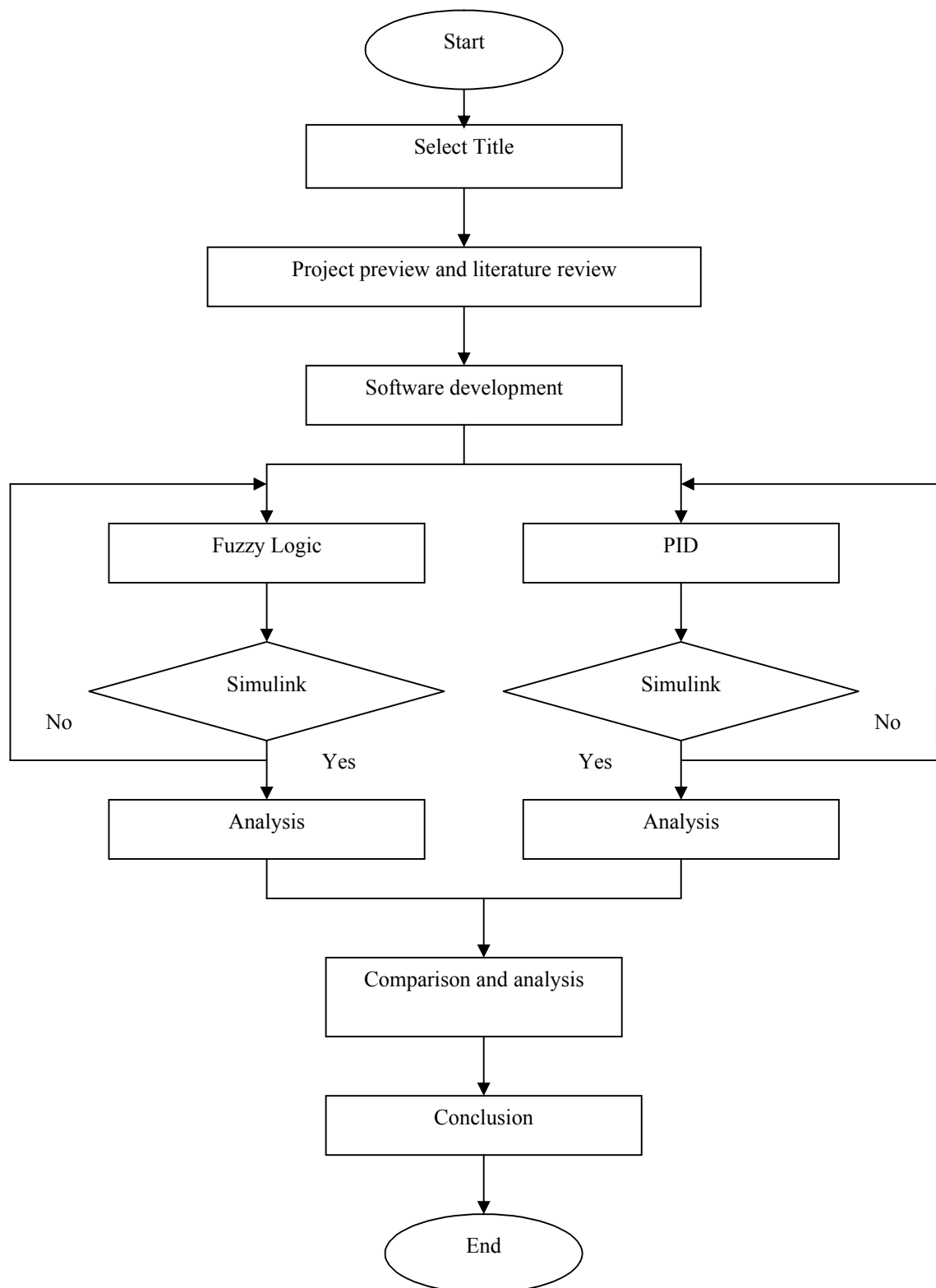
**Figure 1.5** Block Diagram of a PID Controller

## 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 undertaken 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.



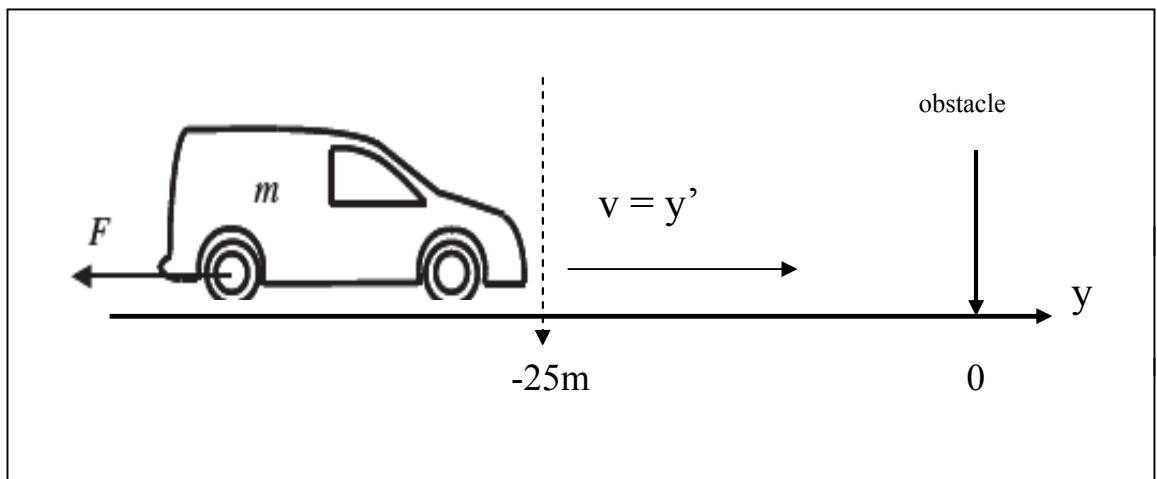
## CHAPTER 2

### 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 disregarded. 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<sup>-1</sup>. We have thus identified the following constants:

$$m = 1500\text{kg}$$

$$y(0) = -25\text{m}$$

$$y'(0) = 10\text{ms}^{-1}$$

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<sup>-1</sup> will bring the speed to zero at a distance of 27.3 m.

$$v = 80\text{km/h} = 22.22\text{ms}^{-1}$$

$$y = 27.3\text{m}$$

Knowing that kinetic energy is converted to work:

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

$$W = Fs$$

Therefore;

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

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

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

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

$$= 13600 \text{ N}$$

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**

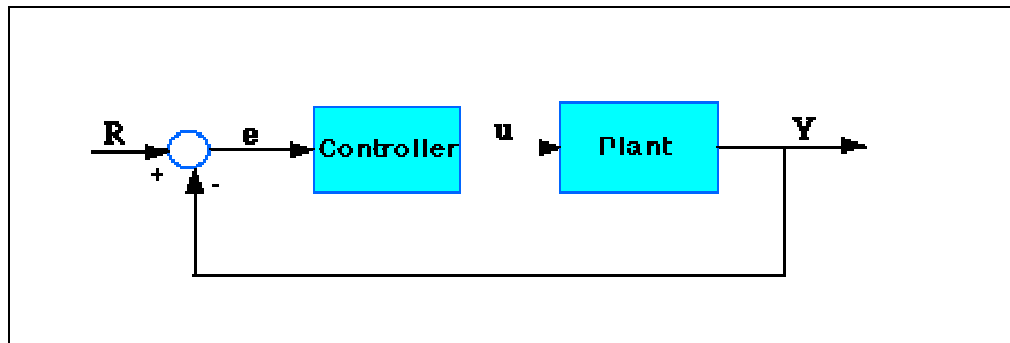
#### **3.1 Introduction**

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  $F = K_p(e + T_d \dot{e})$ .

The closed-loop system equations are

$$F = ma$$

$$F = K_p(e + T_d \dot{e})$$

Rearranging both equations:

$$y'' = \frac{K_p (e + T_d \dot{e})}{m}$$

$$= \frac{-K_p T_d}{m} (y') - \frac{K_p}{m} (y)$$

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  $\omega_n$  as the frequency of oscillation of the system without damping and damping ratio,  $\zeta$ . 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  $T_d = 1$ , we will get  $K_p = 6000$ .

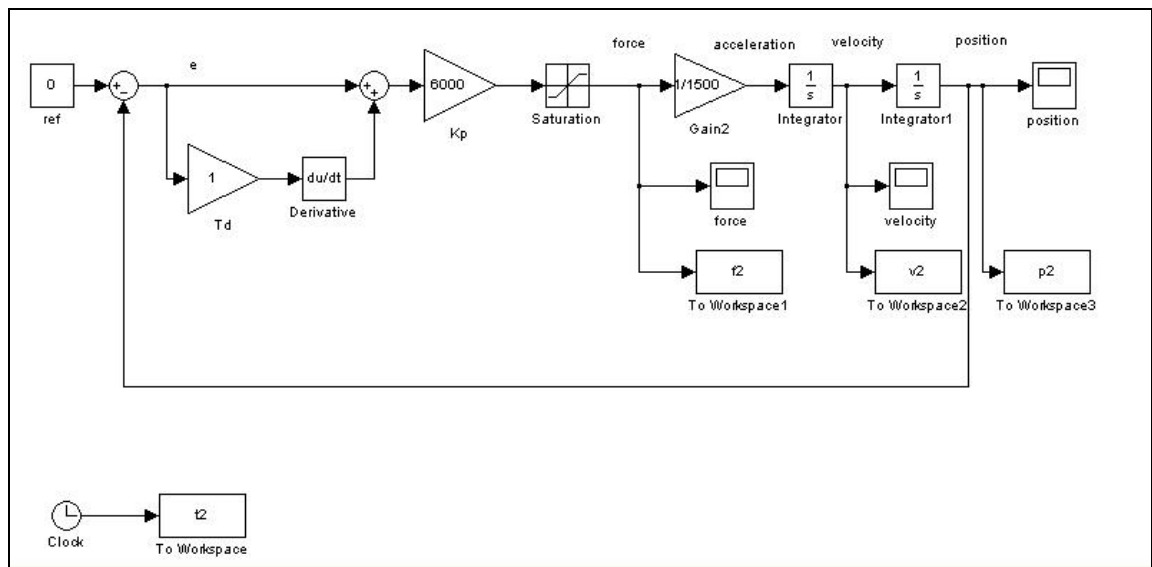
$$K_p = 6000$$

$$T_d = 1$$

### 3.2.3 PID Simulink Model

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

$T_d = 1$ . 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.

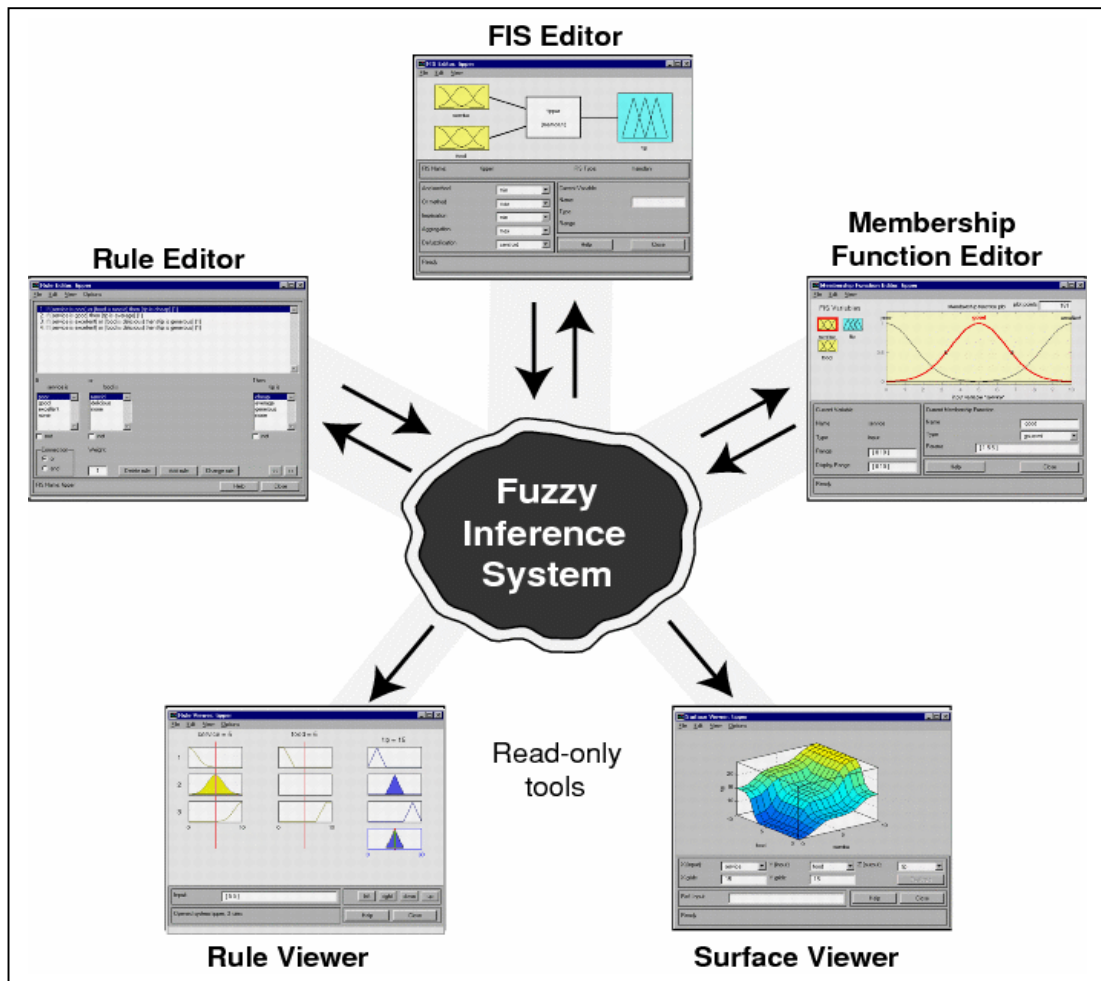


**Figure 3.2** PID Simulink Model

### 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

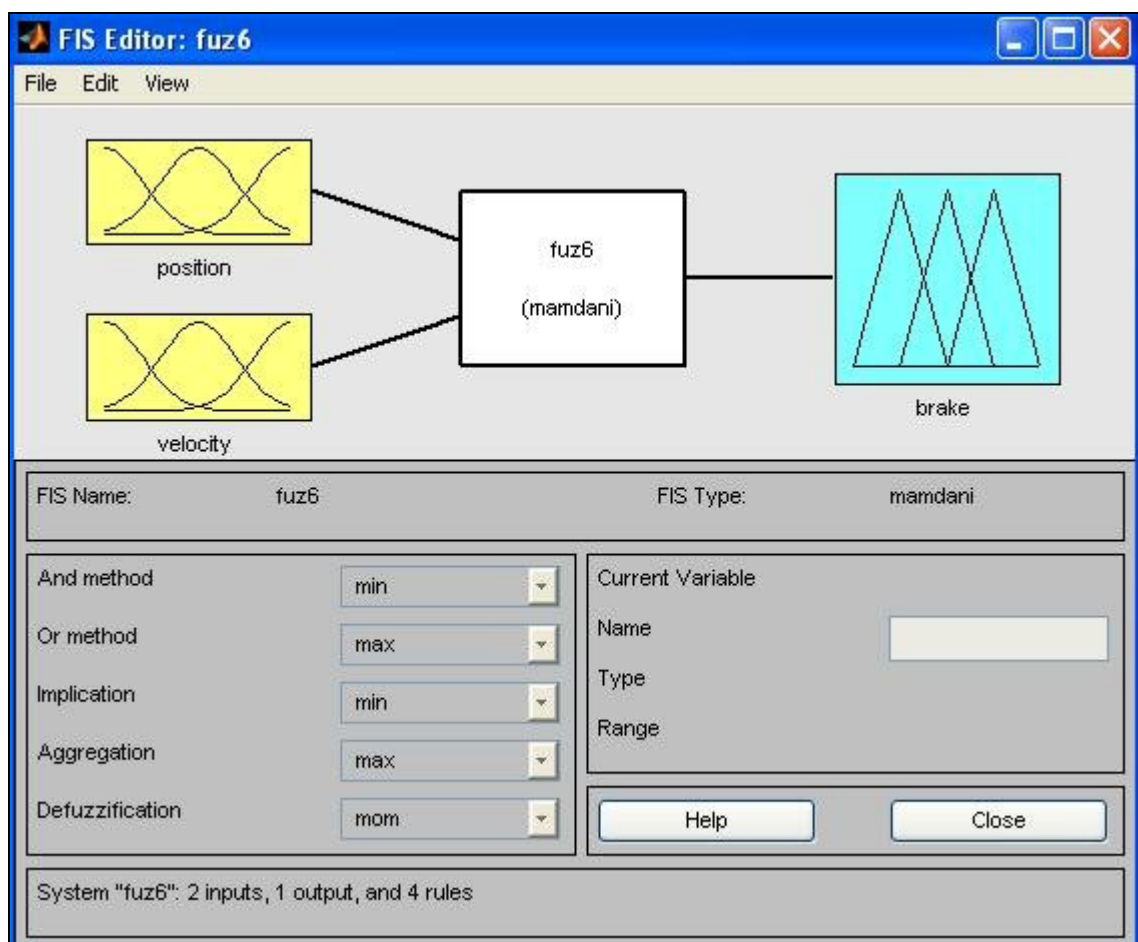
#### 3.3.2.1 FIS Editor

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. Defuzzification method used for this controller is mean of maximum (mom) method.



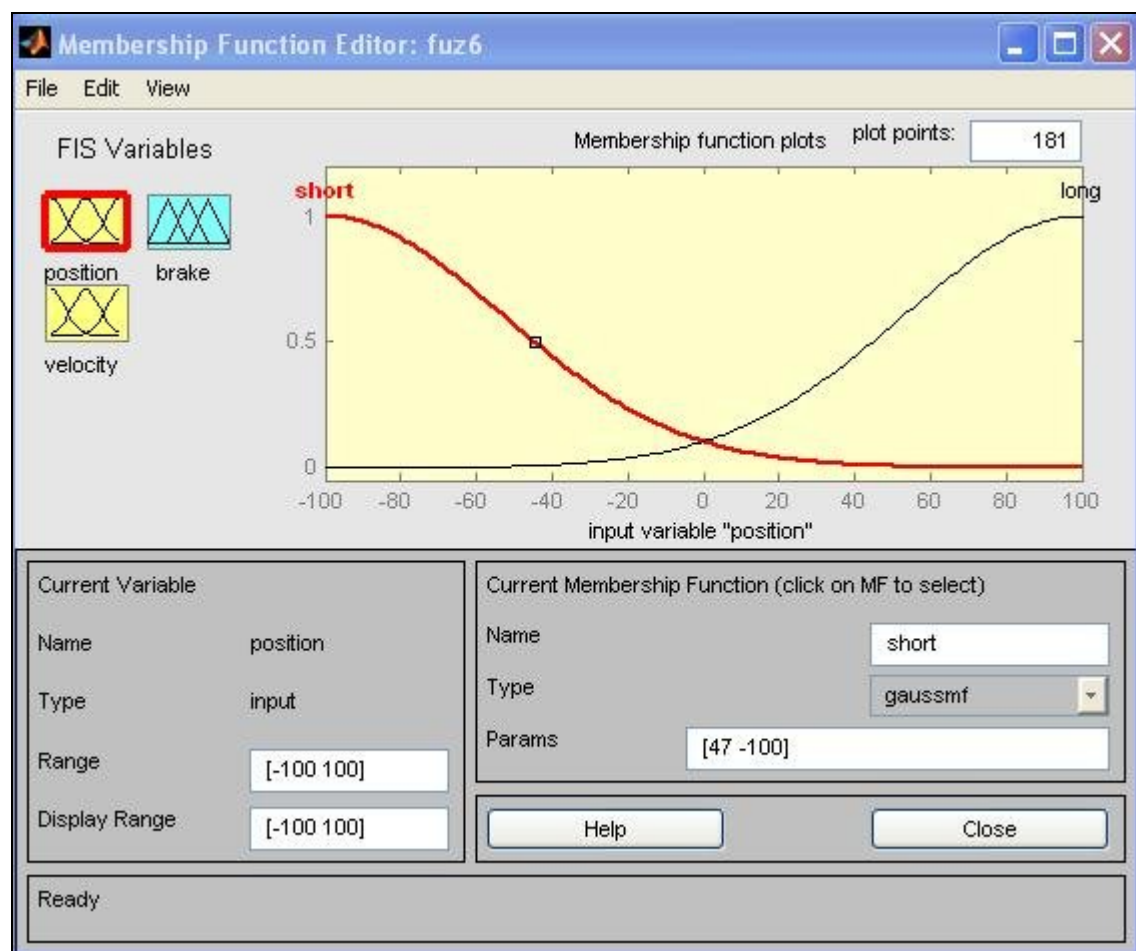
**Figure 3.4** FIS Editor

### 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