WOLKITE UNIVERSITY



College of Engineering and Technology
Department of Electrical and Computer Engineering

DESIGN AND CONTROL OF AUTOMATIC CRUISE CONTROL SYSTEM USING FUZZY LOGIC CONTROLLER

Prepared by:

Asmamaw Ayele [ENGR/145/07]

Enbakom Alehgn [ENGR/375/07]

Amide Worku [ENGR/121/07]

Advisor: Mr. Misker Shumeye

A Final-year Project report submitted for the partial fulfillment of the requirement for Bachelor of Science degree in Electrical and computer Engineering

June, 2019
Wolkite, Ethiopia

Declaration

This declaration is made on the……...day of……...2019.

Student's Declaration:

We **signature**

1. Asmamaw Ayele……
2. Amide Worku …….....
3. Enbakom Alehgn……....

Hereby declare that the work entitled **fuzzy logic-based design and simulation of automatic cruise control system** is our original work. We have not copied from any other students' work or from any other sources except where due reference or acknowledgement is made explicitly in the text, nor has any part been written for us by another person.

Supervisor's Declaration:

We **Signature**

1. ……
2. ……
3. …….

Hereby certifies that the work entitled **fuzzy logic-based design and simulation of automatic cruise control system** was prepared by the above-named student and was submitted to the department of Electrical and computer engineering as a partial fulfillment of the requirement for Bachelor of Science degree, to the best of our knowledge, is the said student's work.

Received for examination by: _____ Date: _____

(Name of the supervisor).

Acknowledgement

First and for most, we would like to thank **God** and his mother's **St. Marry** for all things and for his forgiveness, charity and strength he has given to us.

Secondly, we would like to express our deepest sincerity and respect full gratitude for our advisor ***Mr. Miskir for*** his valuable and fruit full advice in giving clue or hint and showing best direction in what manner that we could prepare this thesis project paper. Finally, we would like to thank our instructors and those who support us in different ways.

Table of contents

Contents **page**

Declaration i

Acknowledgement ii

Table of contents iii

Table of Figure v

Abstract vi

Acronyms vii

Chapter One 1

Introduction 1

1.1. Background of the project 1

1.2. Objectives of the project 2

1.2.1. General objective 2

1.2.2. Specific objective 2

1.3. Statement of problem 2

1.4. Significance of the project 3

1.5. Scope of the study 3

1.6. Limitations of the project 3

1.7. Organization of the project 4

Chapter Two 5

Literature Review 5

Introduction 5

Chapter Three 7

Methodology 7

3.1. Introduction 7
3.2. Block diagram of the system 7
3.2.1. System components 8
3.3. Matlab software 10
3.4. Control system 10
3.4.1. Mathematical modeling of Cruise control system 10
3.4.2. Controller design 19
3.4.2.1. Fuzzy logic toolbox in Matlab 23
3.4.2.2. Input variables 24
3.4.2.3. Output variable 26
3.4.2.4. Inference engine (Rule viewer) 27
3.4.2.4.1. Graphical User Interface (GUI) Tools 27
Chapter Four 29
Result and Discussion 29
4.1. Introduction 29
4.2. Output result from Matlab 29
4.3. Simulink model of the system 33
Chapter Five 38
Conclusion and Recommendation 38
5.1. Conclusion 38
5.2. Recommendation for future work 38
References 39
7. Appendices 40

Table of Figure

Figure 3.1 General block diagram of the system 7

Figure 3.2 detail design of the system 8

Figure 3.3 Model of the car in motion 11

Figure 3.5 Longitudinal dynamics of the vehicle 12

Figure 3.5 The graph of plant transfer function using Matlab 19

Figure 3.6 basic structure of fuzzy inference system (FIS) 21

Figure 3.7 basic shapes of membership function 22

Figure 3.8 Fuzzy logic tool box designs for cruise control 24

Figure 3.9 Input membership functions for distance 25

Figure 3.10 Input membership functions for friction 26

Figure 3.11 output membership functions for speed 27

Figure 3.12 Rules generated from inference engine (rules editor). 28

Figure 4.1 Fuzzy Output result (Rule viewer) at low speed 29

Figure 4.2 Fuzzy Output result (Rule viewer) at medium speed 30

Figure 4.3 Fuzzy Output result (Rule viewer) at high speed 31

Figure 4.5 Uncontrolled responses 33

Figure 4.6 Simulink model of the system 34

Figure 4.7 Simulation of overall system 35

Figure 4.8 Simulation result with system specifications 36

Figure 4.9 Speed control response using fuzzy logic 37

Abstract

This project presents the design and simulation of fuzzy logic based an automatic cruise control, which is a system that maintains the speed of a car and keeps a minimum distance to the closest vehicle in front by giving command to the driver. Ride comfort on the highway often interrupted because drivers need to adjust the vehicle speed. Safe distance between vehicles should be maintained is the main reason. The situation of monotonous and high speed will increase the risk of accidents on highway. A device is required by the driver to adjust the vehicle speed during the long distance (cruise) driving on highway without neglecting the safety aspects. The cruise control system is one of the most enduringly popular and important models for control system engineering. The system is very simple to understand and yet the control techniques cover much modern design method, which include modeling the plant and designing fuzzy controller for controlling the vehicle speed. Then dynamics of the system has modeled to provide a transfer function for the plant. Fuzzy controller is designed for linear model. The external disturbances such as road grades considered to stabilizing the system. The fuzzy inference method used in this study is Mamdani. The controller has modeled by using MATLAB/Simulink software and we can get performance characteristics values such as percentage overshoot 0.505%, rise time 396ms, this shows that our system has good characteristic performance.

Key words: *MATLAB/Simulink, fuzzy logic controller,*

Acronyms

ACC – Automatic Cruise Control

FIS – Fuzzy Inference System

GUI – Graphical User Interface

MATLAB – Matrix Laboratory

MFs – Membership Functions

PID – Proportional Integral Derivative

Chapter One

Introduction

1.1. Background of the project

The cruise control system is one of the most enduringly popular and important models for control system engineering. The system is widely used because it is very simple to understand and yet the control technique covers important modern design method. Cruise control is designed to solve several problems faced by the drivers and passengers such as fatigue duty and traffic accident, and it is performed only as a speed control [1]. Nowadays, the cruise control must become a common feature in most automobiles, this is because there is a great crowd of vehicles and passenger in most countries, mostly in our country Addis Ababa is a good example of this crowd. This system reduces the driver's tasks for adjusting the gas pedal and checking the speedometer frequently depend on the distance from the front vehicle. Cruise control works by using the principles of automatic control system on throttle position setting. The drawback of this system is that, it is not possible to automatically reduce speed in case of dangerous situation [2].

The historical background of automated cars was developed back in 1930's. In the late 1980 and beginning of 1990's, the main initiative was to improve the highway capacity and safety with automation in highway and vehicle level [3]. During 1990s, the capability for car automation systems was demonstrated in Europe, Japan and the United State [4]. The idea of fuzzy logic was invented by Professor L.A Zadeh of the University of California at Berkeley in 1965. The fuzzy theory provides a mechanism that represents a linguistic construction such as: many, high, low, medium, small. Fuzzy rules can come from brain or they can ask an expert for the if-then rules or the expert can ask themselves to state the rules and tune them. Fuzzy logic is one of expert system techniques and fixes its ability in solving most expert problem. A common fuzzy inference system is the Mamdani type. The result from Mamdani's inference for a given fuzzy rule is the part of the fuzzy set that lies under the activation degree. The linguistic variables are converted into crisp set via defuzzification process [5].

Fuzzy logic idea is similar to the human being's feeling and inference process unlike classical control strategy, which is a point-to-point control, fuzzy logic control is a range-to-point or range-to-range control. The output of a fuzzy controller is derived from fuzzification of both inputs and outputs using the associated membership functions. A crisp input will be converted to the different members of the associated membership functions based on its value. From this point of view, the output of a fuzzy logic controller is based on its memberships, which can be considered as a range of inputs. Fuzzy logic toolbox helps to create and edit fuzzy inference system. In addition, the effect of increasing input uncertainty may be easier to understand and control than to change parameters of membership functions on sets of fuzzy rules. Fuzzy logic has been used for different applications: It has been used for flow control systems of a manufacturing plant. By considering such huge applications of fuzzy logic we need to design our project by using such controller.

1.2. Objectives of the project

1.2.1. General objective

The main objective of this project is to design and simulate the automatic cruise control system by using fuzzy logic controller in MATLAB/Simulink.

1.2.2. Specific objective

Specifically, this project has undertaken in the following stages:

- ✓ Design Mathematical model of the plant
- ✓ Design fuzzy logic controller
- ✓ Simulation of the system using Fuzzy logic controller in MATLAB/Simulink
- ✓ Analysis of the simulation results.

1.3. Statement of problem

The main motivation to do this project is to maximize the performance of the system by using the appropriate controller by considering the distance and awkwardness of roads as input of the system to control the speed of the vehicle. Since as we have listed in the literature review most of the projects has been performed by using classical controller called PID, since this controller has played different role in controlling speed of the vehicle. But it has some limitation, since it

doesn't consider the distance between the car being driven and the front vehicle and the friction of the road, which become the input of ultrasonic sensor. And also the performance characteristic parameters like percentage overshot, steady state error, rise time and setting time are not minimized under such controller. In our project we use more advanced controller called fuzzy logic controller which include such features which PID controller cannot hold and it can stabilize the system performance by minimizing percentage overshoot, rise time and setting time parameters.

1.4. Significance of the project

The relevance of this project is to regulate the vehicle speed so that it follows the driver's command and maintains the speed at the commanded level. And also our project is to design and simulate automatic cruise control system in order to avoid car collision and minimize the number of deaths in each time. Typically, this project will play a great role in decreasing the number of deaths in each time and place due to inability to control the speed of the vehicle in the absence of sensors which give the driver to adjust the throttle setting. With the advancement of software's invention our thesis has been the one which will be designed to reduce such effect. From those software's we use Matlab/Simulink software in order to simulate the overall system. To give the best direction for the driver, the information given to the driver is either the distance from the front vehicle or the speed at which he is driving to control the speed of the vehicle. And the controller that we use is the most advanced controller which is fuzzy logic toolbox.

1.5. Scope of the study

The study covers the design, control and simulation of automatic cruise control system by using fuzzy logic toolbox under Matlab/Simulink software. In our project we have included mathematical modeling of the vehicle, fuzzy logic toolbox design and simulation, and also the Simulink model of the overall system by using such controller.

1.6. Limitations of the project

Listed below are client-specified limitations of our project:

- The system response will be easily affected by plant model parameters
- This project is limited to software simulation
- This project cannot automatically reduce speed of the car during dangerous situation, other than showing the driver the distance from his car and front vehicle.

1.7. Organization of the project

In this project a clear out line has been provided with descriptions of each chapter are given below: This report consists of 5 chapters. **Chapter 1** gives an overview of the entire project including statement of the problem, project objectives, and project scope. **Chapter 2** reviews and related works with compression with system have been discussed. **Chapter 3** discusses on the methodology and basic procedures to design and simulation of the system. In **chapter 4,** the result fuzzy logic controller for each input, output and, also the Simulink model simulation results are recorded. And the **last Chapter** summarizes Conclusion and recommendations.

Chapter Two

Literature Review

Introduction

This chapter reviews some reference from previous project, journal, articles, and book and data sheets. A lot of research work has been published in the field of automatic cruise control system and various techniques have been proposed.

A major challenge in designing intelligent vehicles capable of autonomous motion whether in air, water or land is a reliable collision avoidance system. Any mobile robot or vehicle needs an ability to perform reliable obstacle detection as well as have efficient and effective collision avoidance actuation ability when performing in an unknown or dynamic environment. Mobile robots and intelligent vehicles have recently been of great interest to commercial industries such as automotive and aircrafts industries as well as to the military. There has been a great amount of research devoted to the development of a reliable cruise control system.

Khalid Sailan and Klaus. Dieter Kuhnert, researchers provide Modeling and design of cruise control system with feed forward for all terrain vehicles, their research paper presents PID controller with feed-forward control in cruise control system as one of the most enduringly popular and important models for control system engineering [1]. Since this project held on using PID controller, which didn't consider the distance between the vehicles and this cause has low performance. Since the controller that we need to use has a special ability to reduce such performance characteristic like setting time, rise time and percentage overshoot.

A lot of researches have done on cruise control system, which is under the controller called PID controller. Since this controller is preferable than the other types, even though this controller cannot the ability to provide distance sensor and other special parameters, and also has some feedbacks like the ability to minimize system performance characteristic parameters is too low when we compared with the most advanced controller know as Fuzzy logic control. The result from Mamdani's inference for a given fuzzy rule is the part of the fuzzy set that lies under the activation degree. The linguistic variables are converted into crisp set via defuzzification process

[5].Since this controller provides ultrasonic sensor for detecting the distance from the front vehicle, and has ability to minimize the effect of performance characteristic parameters like percentage overshoot, steady state error, setting, and rising time.

Chapter Three

Methodology

3.1. Introduction

This chapter explains the methods used to design and simulate the project. For successful completion of this project some steps will be followed to carry out different tasks. Different literature will be revised relating to this project and data will be collected about automatic cruise control system in Matlab/Simulink using fuzzy logic tool box. To include some automation tasks and increase the performance of the system we need to design our project by using the more advanced controller called fuzzy logic controller.

3.2. Block diagram of the system

The project has two FIS (Fuzzy Inference System) variables in the controller. From these we have two input variables and one output variable. Those input variables are distance from the front car and awkwardness of road (Friction) and the output variable become Speed (throttle control, which is applied as control input for to the plant) to keep safe distance between cars. The other elements of the system block diagram are the plant and feedback as shown below.

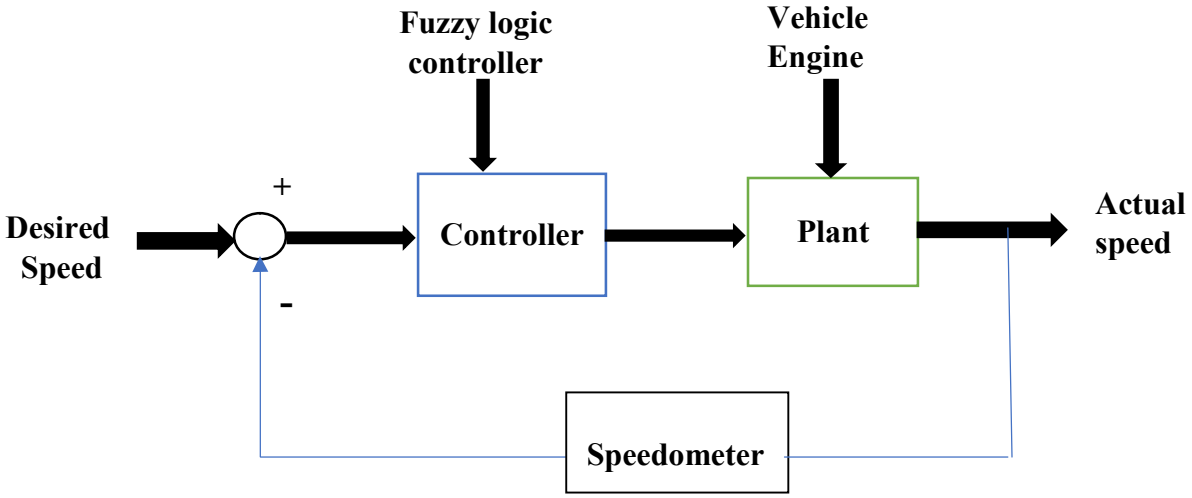


Figure 3.1 General block diagram of the system

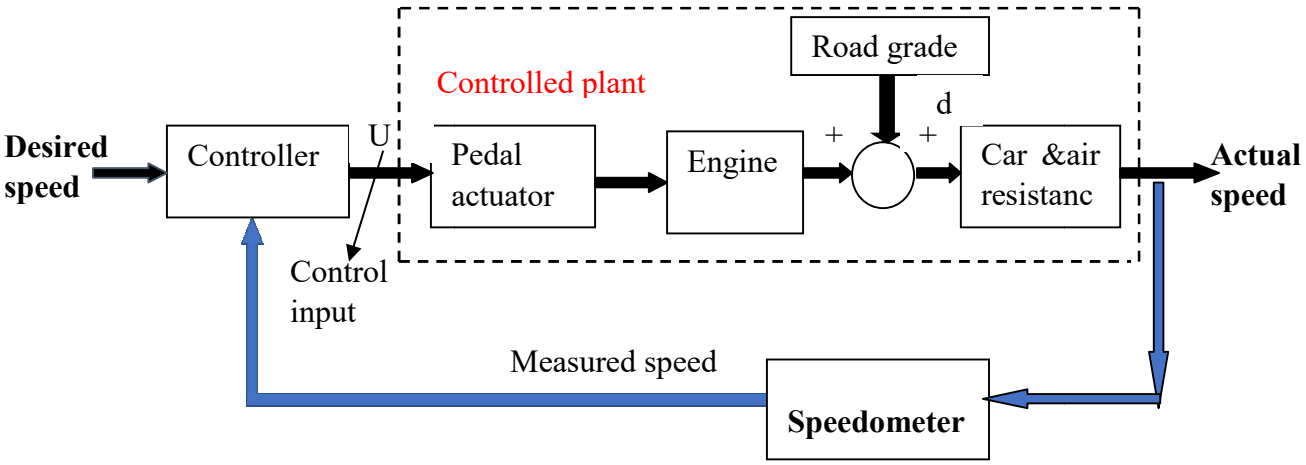


Figure 3.2 detail design of the system

From the above block diagram the following parameters are described.

- Desired speed is the speed which the vehicle is driven, which needed to be controlled
- The controller is fuzzy logic controller, which has two inputs so called distance and the road awkwardness or friction. And the output of the controller which is throttle used to move the car forward.
- The plant is automobile/vehicle, and the pedal actuator is which used to start the engine based on the throttle input, which convert the electrical signal in to mechanical motion.
- The output of the system is actual speed which is controlled by the fuzzy logic controller
- Speedometer is a device which record speed sensor measures, which is located on the dash board/control computer of the vehicle.
- The disturbance of the system/ road grade be the external factor for the driver for the sake of safe operation and that is varying in every point, but for our project we consider the ideal system, and neglect the road grade and wind speed as small effect on the driving process.

3.2.1. System components

A cruise control system basically consists of the following devices:

- ✓ Vehicle speed sensor,
- ✓ Ultrasonic sensor

- ✓ Control computer (Usually call it dashboard),
- ✓ Actuator to set throttle,
- ✓ Cancellation switches

Automobiles should consist the above devices that perform these necessary functions that allow a cruise control system to exist. The following paragraphs give a quick summary of the purpose of each component.

A. Vehicle speed sensor: The vehicle speed sensor measures how fast the vehicle is traveling. This is central to the control of speed, because in order to maintain a constant speed, the actual desired speed of the car must be known. When a cruise control system is activated, the vehicle speed sensor sends data of how fast the car is traveling to a computer where the value is stored as a reference.

B. Ultrasonic sensor: it is electronic device which used to sense the distance and the recorded data should read by the control computer, to guide the driver.

C. **The control computer**: is where the brains of the cruise control system are located. The control computer receives signals from the vehicle speed sensor, and Ultrasonic sensor and then sends signals out that adjust the throttle to maintain the desired speed that is determined when the system is first activated. The computer is also responsible for receiving input from the driver that allows the cruise control to increase or decrease the vehicle speed, as well as receiving signals that turn the cruise control off.

D. **The actuator**: receives electronic signals from the computer and turns them into mechanical motion that controls how much the throttle is open.

E. **Cancellation switches**: turn the cruise control system off. A switch is always installed on the brake pedal that breaks the cruise control circuit when the brake is depressed; likewise, for cars with manual transmission, a switch of the same sort is installed on the clutch pedal. The reasons for this are obvious it would be very dangerous and harmful for the cruise control system to stay activated while the driver is trying to stop or slow down the car. These switches allow the cruise control system to shut off as soon as the driver desires to slow down without the driver actually having to consciously turn it off.

3.3. Matlab software

MATLAB is a programming language developed by Math Works. It started out as a matrix programming language where linear algebra programming was simple. It can be run both under interactive sessions and as a batch job. **MATLAB** (matrix laboratory) is a ***fourth-generation*** high-level programming language and interactive environment for numerical computation, visualization and programming.

It allows:

- matrix manipulations;
- plotting of functions and data;
- implementation of algorithms;
- creation of user interfaces;
- interfacing with programs written in other languages, including C, C++, Java, and FORTRAN; analyze data; develop algorithms; and
- Create models and applications.

It has numerous built-in commands and math functions that help you in mathematical calculations, generating plots, and performing numerical methods. And It is widely used as a computational tool in science and engineering encompassing the fields of physics, chemistry, math and all engineering streams.

3.4. Control system

3.4.1. Mathematical modeling of Cruise control system

Early cruise control systems lacked the fancy computer programming that modern systems utilize. As a result, early systems were not as good at actually keeping the speed of the car constant; the throttle was held in position, but as other forces on the vehicle varied, the vehicle speed fluctuated some. The logic of modern control systems in cruise control comes out of the need to compensate for the physical forces that act on a car and try to change its speed. In a simplified ideal situation, forces would be present in the movement of a car the engine force (driving force), gravitational force, aerodynamics drug force, and the inertia force. In such a situation, the control system would be very simple. A simple proportional controller could

control the speed by simply having set throttle positions for each possible speed setting. However, this idealized situation is far from reality. Cars have varying drag forces due to wind, varying mechanical component friction due to speed and temperature, varying gravitational forces due to inclines, and varying engine force due to hydraulically activated steering pumps. For the sake of understanding the basics of how the controls work, the simplest case mentioned is a good starting point.

Forces are the engine force (driving force), gravitational force, aerodynamics drug force, the inertia force are considered, so the main goal of a cruise control system will be to get the car to the desired speed, like what happens does when a driver hits the resume button. Once this resume speed is achieved, the forces on the car will not be changing, so no logic will be needed for maintaining the speed. The logic instead will deal with making the car accelerate up to the desired speed rapidly without overshooting that desired speed by very much.

Based on the command from the driver $V_{d,}$ the feedback signal from speed sensor then **controller** maintain vehicle's speed V_a by adjusting the throttle control **U**, to increase or decrease the engine driver force $F_{d.}$

Look the figure below: the model of the vehicle indicating all forces acting on it when it is in motion.

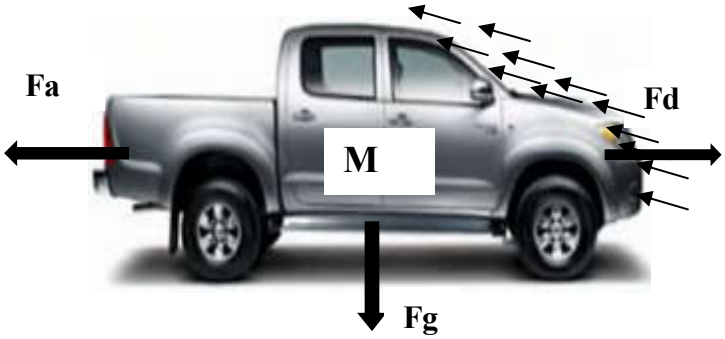


Figure 3.3 Model of the car in motion

According to Newton's first law of motion, "An object in motion continues in motion with the same speed and in the same direction unless acted upon by an unbalanced force." Any

passengers in the car will also be decelerated to rest if they are strapped to the car by seat belts." [11]. The purpose of the cruise control system is regulating the vehicle speed so that it follows the driver's command and maintains the speed at the commanded level. Based on the command signal from the driver and the feedback signal from the speed sensor, the cruise controller regulates vehicle speed by adjusting the engine throttle angle u to increase or decrease the engine drive force Fd.

The longitudinal dynamics of the vehicle is given as follows:

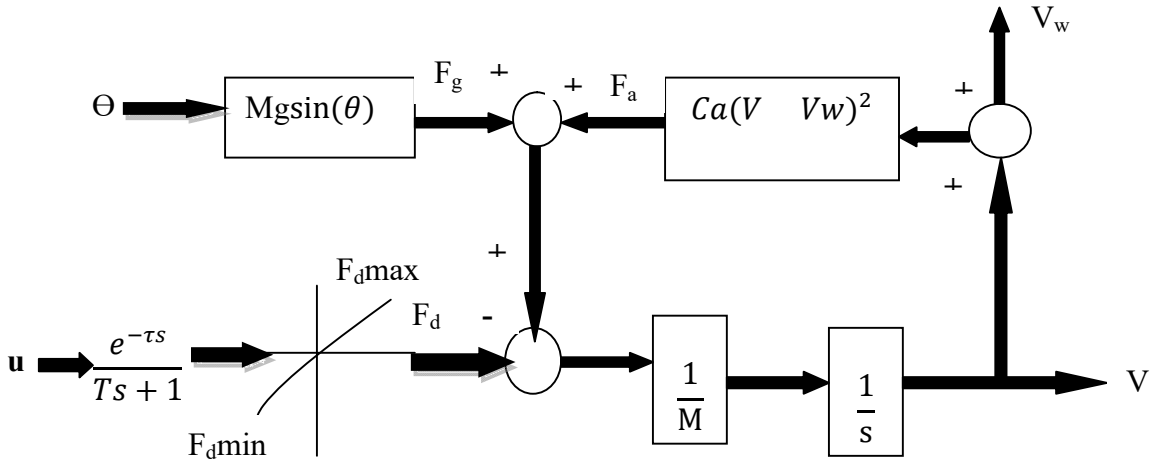


Figure 3.5 Longitudinal dynamics of the vehicle

From the above the longitudinal dynamics of the vehicle we can generate mathematical model of the system, regarding to Newton's law of motion as follows:

$F_d = M\frac{dv}{dt} + F_a + F_g$

Where $M\frac{dv}{dt}$ = the inertial force

M = total mass of the vehicle

F_a = the aerodynamics drug force

F_g = Force due to gravity

Then, $F_g = Mg \sin(90)$, *but consider no road inlination*

- Hence, it is assumed is no wind gust and no grading exist during the movement of the car. Applying this zero-initial condition to the block diagram, the model is left with the forward path and the unity feedback loop of the output speed. Since the state variables have been chosen to be the output speed and the drive force, the corresponding state and output equations are found to be:

$$F_d = M\frac{dv}{dt} + F_a + F_g \quad \text{..... (1)}$$

Where $F_g = mg\sin\theta$, when θ *is t e road inclination*

But for our design we assume that there is no more inclination exist.

The actuator and the vehicle propulsion system are modelled as cascade temporary lag of first order $\frac{C1e^{-\tau s}}{Ts+1}$

$F_a = Ca(V \quad Vw)^2$, where Vw is the wind speed, which is neglected for our design

$$\frac{dv}{dt} = \frac{1}{M}(F_d \quad C_aV^2) \quad \text{..... (2)}$$

$$\frac{dF_d}{dt} = \frac{1}{T}(C_1U(t-\tau) - F_d) \quad \text{..... (3)}$$

$y = v.$

However, a problem of non-linearity arises. There is a squared term in the equation (2). One way to overcome this problem is to linearize all the state-equations by differentiating both left and right hand sides of the equations with *M*, *Ca*, *C1*, *T* and *v* remain constant. After differentiating, the state-equations become:

$\frac{dv}{dt} = \frac{1}{M}(F_d \quad C_a V^2)$ and y= v

$\frac{d}{dt}\left(\frac{dv}{dt}\right) = \frac{d}{dt}(\frac{1}{M}(F_d \quad C_a V^2))$

$$\frac{d^2v}{dt^2} = \frac{1}{M}(\quad 2C_a V\delta V + \delta F_d)$$

Since, $\boldsymbol{\frac{dv}{dt} = \delta}$**. Then** $\boldsymbol{\frac{dCV^2}{dt} = 2CV\frac{dv}{dt} = 2CV\delta V}$

$$\frac{d^2v}{dt^2} = \frac{1}{M}(\quad 2C_a V\delta V + \delta F_d) \ldots\ldots\ldots\ldots\ldots\ldots (4)$$

From equation (3) we have

$\frac{dF_d}{dt} = \frac{1}{T}(C_1 U(t-\tau) - F_d)$…… (Differentiating with respect to **t**)

$\frac{d^2Fd}{dt(dt)} = \frac{d}{dt}(\frac{1}{T}(C_1 U(t-\tau)) - \frac{d}{dt}F_d$……. Since $\frac{d}{dt} = \delta$

$\frac{d^2F_d}{dt^2} = \frac{d}{dt}(\frac{1}{T}(C_1 \delta U(t-\tau) - F_d))$…… (5)

From Equation (4) and (5):

- ✓ δV is the discrete output speed
- ✓ δF_d *is t e discrete driving force*
- ✓ C_a *is aerodynamic drag force coeffcient*
- ✓ C_1 is natural vehicle constant (which is set to limit the variation during the change of gears by the pedal during driving)

Then the transfer function to the car model can be obtained by solving the state equation for $\frac{\Delta V(s)}{\Delta U(s)}$

Since from the above equations (4) & (5) we have:

$$\frac{d^2v}{dt^2} = \frac{1}{M}(\ \ 2C_aV\delta V + \delta \mathrm{F_d})$$

$$\frac{d^2F_d}{dt^2} = \frac{d}{dt}\left(\frac{1}{T}(C_1\delta U(\text{t-}\tau)\text{-}\delta F_d)\right)$$

The output equation becomes

$$Y=\delta v$$

In the above equations, δV means that the output is discrete and δF_d also it means the drive force if discrete. The symbol V means the discrete and δu(t-T) is the time delay of the engine. Up to this point the state and output equations are written in the time domain.

The linearized model provides a transfer function. It can be obtained by solving the state in S-domain equations to the ratio of $\frac{\Delta V(s)}{\Delta U(s)}$.

Generally, δV =the output is discrete

$$\delta F_d = t\ \ e\ driving\ force\ is\ discrete$$

V= the actual speed

δU(t-T) = the time delay of the engine

$$\frac{\Delta V(s)}{\Delta U(s)} = \ \ ration\ of\ state\ equation$$

Then by ignoring initial conditions transform into S domain (Laplace transform).

$$\frac{d^2v}{dt^2} = \frac{1}{M}\left(\ \ 2C_aV\delta V(t) + \delta \mathrm{F_{d\ (t)}}\right)$$

$$S^2V(s) = \frac{1}{M}(\ \ 2CaVSV(s) + SF_d\ (s))$$

$$S^2V(s) = \frac{-2C_aVSV(s)}{M} + \frac{S}{M}F_d\ (s)$$

$$S^2V(s) + \frac{2C_aVSV(s)}{M} = \frac{S}{M}F_d\ (s)$$

$$V(s)\ [s^2 + \frac{2C_aVSV}{M}\ ] = \frac{S}{M}F_d\ (s)$$

$$Fd\ (s) = V(s)[\frac{MS^2 + 2C_aVS}{S}]$$

$Fd\ (s) = V(s)[MS + 2C_aV]$…… (1*)

$Fd\ (s) = V(s)[ms + 2C_aV]$……. the S domain of the driving force equation

Since $Fd\ (s) = V(s)[ms + 2C_aV]$

From equation number (2)

$\frac{d^2F_d}{dt^2} = \frac{d}{dt}(\frac{1}{T}(C_1\delta U(t-\tau) - F_d(t)))$……Using Laplace transform convert to S-domain

$s^2F_d(s) = \frac{1}{T}(C_1SU(s)e^{-S\tau}\ \ \delta F_d(s))$

$s^2F_d(s) + \frac{SF_d(s)}{T} = \frac{C_1SU(s)e^{-S\tau}}{T}$

$F_d(s)\left(S^2 + \frac{S}{T}\right) = \frac{C_1SU(s)e^{-S\tau}}{T}$……. solve for $F_d(s)$

$F_d(s) = (\frac{C_1SU(s)e^{-S\tau}}{T})\ (\frac{T}{TS^2+S})$

$F_d(s) = \frac{C1U(s)e^{-S\tau}}{TS+1}$ ..... (2*)

Equating equation (1*) and (2*)

$Fd\ (s) = V(s)[ms + 2C_aV]$ and $F_d(s) = \frac{C1U(s)e^{-S\tau}}{TS+1}$

$\frac{C1U(s)e^{-S\tau}}{TS+1} = V(s)[MS + 2C_aV]$, by equating the two equations

The throttle actuator and vehicle propulsion system are modeled as a time delay in cascade with a first order lag and force saturation characteristic, the actuator and the vehicle propulsion system are modelled as cascade temporary lag of first order given below.

$\frac{V(s)}{U(s)} = \frac{C_1 e^{\ S\tau}}{(TS+1)(MS+2C_aV)}$, where $\frac{C_1 e^{-S\tau}}{TS+1}$ $for\ actuator\ first\ order\ system$

This is because; in order to show the time lag when, the actuator receives electronic signals from the computer and turns them into mechanical motion that controls how much the throttle is open.

Let us simplify the above equation into standard

$\frac{V(s)}{U(s)} = \frac{\frac{C_1}{MT}e^{-S\tau}}{(S+\frac{1}{T})(S+\frac{2C_aV}{M})}$, Let $\boldsymbol{a} = \frac{C_1}{M\tau}, and\ \boldsymbol{b} = \frac{2C_aV}{M}, \boldsymbol{d} = \frac{1}{T}$, then the above equation simplified as:

$$\frac{V(s)}{U(s)} = \frac{a^{e^{-S\tau}}}{(s+d)(s+b)}$$

From the above equation we have some standard values

Mass of the vehicle (M) = 2940Kg

Aerodynamics drug coefficient (Ca) = for hilux Toyota vehicles then take:

Ca = 0.37

C_1 = 743

T = 5sec

τ = 0.2 sec

g = 9.81m/s^2

V_{ref} = 10kph

$$\frac{V(s)}{U(s)} = \frac{\frac{C_1}{MT}e^{-S\tau}}{(S+\frac{1}{T})(S+\frac{2CaV}{M})}$$, By substituting the above values we get

$$\frac{V(s)}{U(s)} = \frac{0.0505e^{0.2s}}{s^2 + 0.203s + 0.00051}$$

Matlab command and output of the plant model:

$$\frac{V(s)}{U(s)} = \frac{0.0505}{s^2 + 0.202s + 0.00051}$$

Using such transfer function, we can plot the step response of the plant that has to be controlled in the editor window:

```
%% Obtain Step Response Characteristics
% Create the following continuous-time transfer function:
%
% $$V(s)/U(s) = {{0.0505} \over {s^2 + 0.203s + 0.00051}}$$
%
sys = tf([0.0505],[1 0.203 0.00051]);
%%
% Compute the step response characteristics for the system. Specify the
% lower and upper thresholds used in the rise time calculation as 0.05 and
% 0.95.
```

Stepplot(sys)

% plot the graph of the plant transfer function

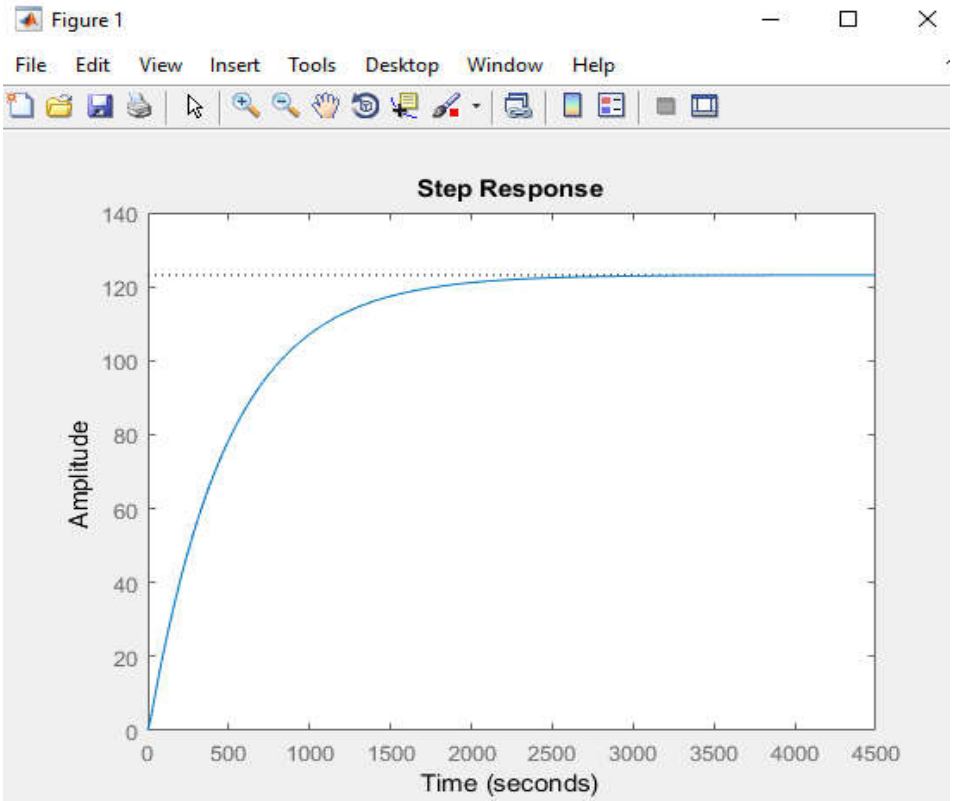


Figure 3.5 The graph of plant transfer function using Matlab

It is obvious that the system described by this transfer function is second order system with time delay. This is because the complexity of the system and the engine parameter system identification tool used to define the system transfer function.

3.4.2. Controller design

The problem of cruise control system is to maintain the output speed *v* of the system as set by input signal base on the command signal from the driver. Cruise controller design is applied assuming a single-loop system configuration with a linear model and nonlinear model. The fuzzy logic controller function is designed to augment or modify the open-loop function in a manner that produces the desired closed-loop performance characteristics. The plant functions represent the actuators and the controller part of the system, and the plant parameters are determined primarily by functional aspects of the control task.

Designing a controller for non-linear systems is difficult. For conventional controller design we require mathematical model and higher order differential equations. Sometimes we linearize mathematical model and design controller for system by applying well developed techniques for solving linear equations. But as we know that real life problems are non-linear in nature and their models are not easy to handle mathematically. We solve daily life problems by if-else rules. So, fuzzy logic rules are heuristic based which takes if-else conditions to solve daily life problems. It is linguistic based technique to design controller. It takes conditions from experience and intuition.

Since fuzzy logic is an approach to computing based on "degrees of truth" rather than the usual "True or False" (1or 0). Boolean logic on which is based the modern computer. The idea of fuzzy logic was first advanced by Dr. Lotfi Zadeh of the University of California at Berkeley in the 1960s [5].

A membership function (MF) is a curve that defines how each point in the input space is mapped to a membership value (or degree of membership) between 0 and 1. The input space is sometimes referred to as the universe of discourse. Or, the membership function is a graphical representation of the magnitude of participation of each input. The rules use the input membership values as weighting factors to determine their influence on the fuzzy output sets of the final output conclusion. Once the functions are inferred, scaled, and combined, they are defuzzified into a crisp output which drives the system. For any set X, a membership function on X is any function from X to the real unit interval [0, 1]. The membership function which represents a fuzzy set is usually denoted by μA. For an element x of X, the value μA(x) is called the membership degree of x in the fuzzy set. The membership degree μA(x) quantifies the grade of membership of the element x to the fuzzy set. The value 0 means that x is not a member of the fuzzy set; the value 1 means that x is fully a member of the fuzzy set. The values between 0 and 1 characterize fuzzy members; which belong to the fuzzy set only partially. A fuzzy set is completely characterized by its membership function (MF). Since most fuzzy sets in use have a universe of discourse X consisting of the real line R, it would be impractical to list all the pair defining a membership function. A more convenient and concise way to define an MF is to express it as a mathematical formula. The definition of the membership functions is a very

delicate point in the design of the FLC, because the only restriction that a membership function must satisfy is that its values must be in the [0,1]range. A fuzzy set can therefore, unlike a crisp one, be represented by an infinite number of membership functions.

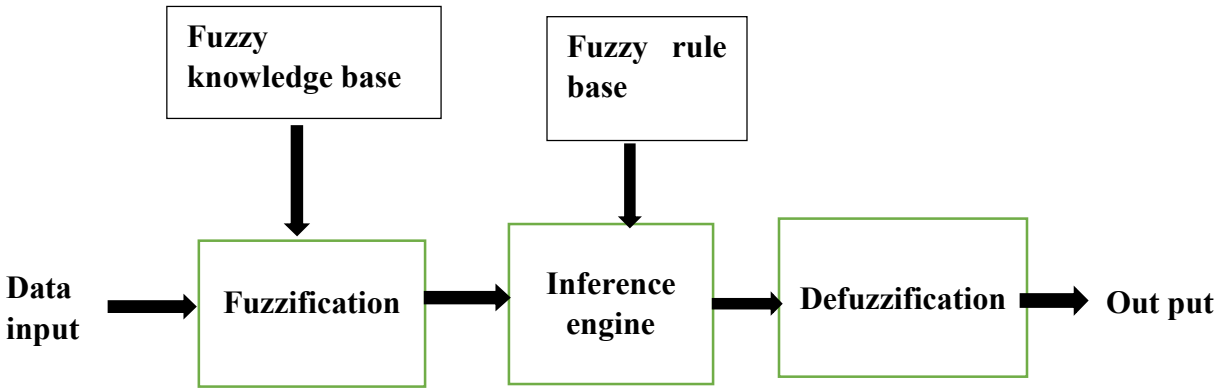


Figure 3.6 basic structure of fuzzy inference system (FIS)

A. Fuzzification

- Fuzzification is the first step of the fuzzy inference system which is used to convert crisp values to the linguistic value.

B. Rule Evaluation

- A fuzzy rule is defined as a conditional statement in the form:

  IF **x** is A

  THEN **y** is B

  Where:

  - ✓ x and y are linguistic variables;
  - ✓ A and B are linguistic values determined by fuzzy sets on the universe of discourse X and Y, respectively.

C. Defuzzification

- Defuzzification is the process of producing a quantifiable result in fuzzy inference system. It is typically needed in fuzzy control systems. These will have several rules that transform several variables into a fuzzy result, that is, the result is described in terms of membership in fuzzy sets [5].
- Defuzzification is the process of converting fuzzified result into crisp value.

Here is a list of general observations about fuzzy logic:

- Fuzzy logic is conceptually easy to understand.

  The mathematical concepts behind fuzzy reasoning are very simple. Fuzzy logic is a more intuitive approach without the far-reaching complexity.

- Fuzzy logic is a more intuitive
- Fuzzy logic is flexible.

  With any given system, it is easy to layer on more functionality without starting again from scratch.

- Fuzzy logic is tolerant of imprecise data.
- Fuzzy logic is based on natural language.

  The basis for fuzzy logic is the basis for human communication (Natural language, which is used by ordinary people in daily basis). This observation underpins many of the other statements about fuzzy logic. Because fuzzy logic is built on the structures of qualitative description used in everyday language, fuzzy logic is easy to use, etc.

The membership function belongs to the following shapes:

1. Triangular
2. Trapezoidal
3. Sigmoid
4. Gaussian

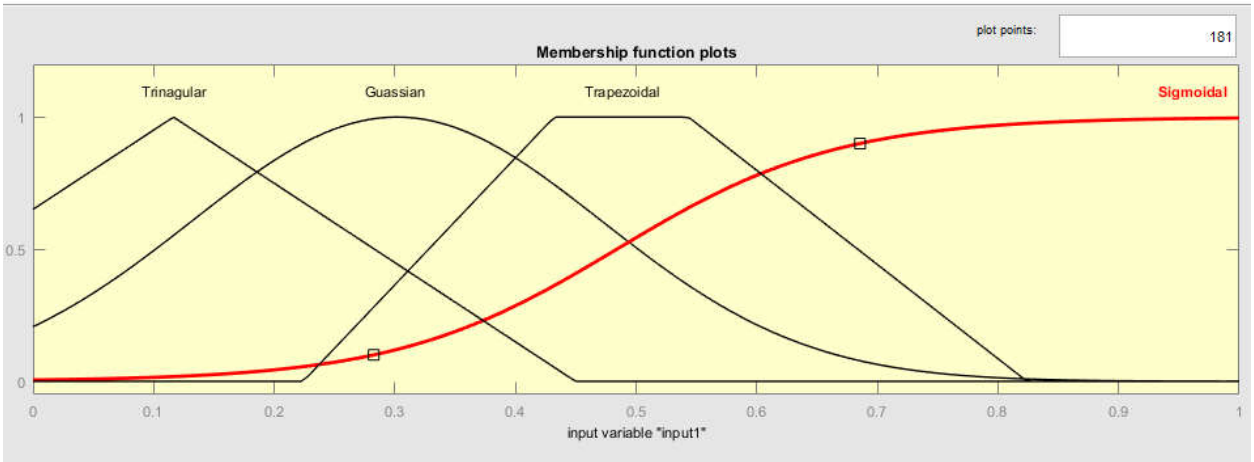


Figure 3.7 basic shapes of membership function

3.4.2.1. Fuzzy logic toolbox in Matlab

Fuzzy Logic Toolbox provides MATLAB functions, apps, and a Simulink block for analyzing, designing, and simulating systems based on fuzzy logic. The product guides us through the steps of designing fuzzy inference systems.

MATLAB fuzzy logic toolbox facilitates the development of fuzzy-logic systems using graphical user interface (GUI) tools command line functionality. There are five primary GUI tools for building, editing, and observing fuzzy inference systems in the Fuzzy Logic Toolbox: Fuzzy Inference System (FIS) Editor, Membership Function Editor, Rule Editor, Rule Viewer and Surface Viewer.

The toolboxes that we model complex system behaviors is by using simple logic rules, and then implement these rules in a FIS (fuzzy inference system). We can use it as a stand-alone fuzzy inference engine (which used to perform rules). Alternatively, we can use fuzzy inference blocks in Simulink and simulate the fuzzy systems within a comprehensive model of the entire dynamic system.

MATLAB has built-in fuzzy controller designing tool. This tool allows designing rule base for controller in linguistics. Inputs and outputs are defined and ***membership functions*** of each input and outputs.
This tool also allows you to select different membership functions and ***defuzzification*** **methods**.

- Write ***fuzzy*** in MATLAB command window and a new window will pop-up.
- Select number of inputs from ***Edit>>add variables>>input/output***
- Double Click the input and a new window will pop-up containing *membership functions* of that input. if you want to change *MFs* click ***Edit>>*** and find your required option for membership functions.
- Click on controller to change model parameters and add rule base for fuzzy controller.
- In first window, in left bottom ***defuzzification*** methods can be changed. We will use centroid method.

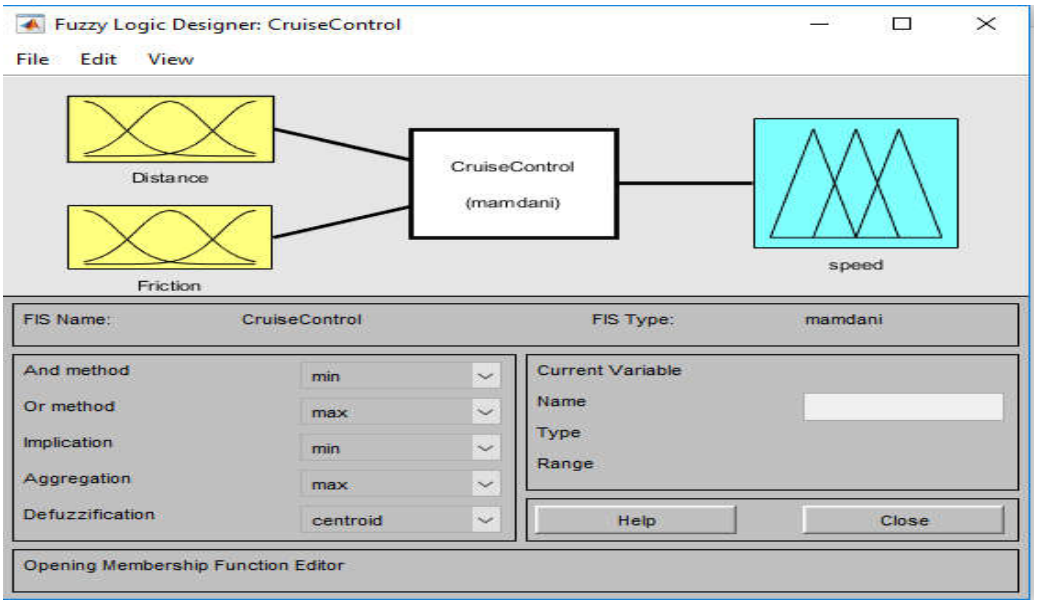


Figure 3.8 Fuzzy logic tool box designs for cruise control

3.4.2.2. Input variables

Using Fuzzy tool in MATLB we can input membership function of desire shape. In this example we have distance from the front vehicle and awkwardness of the road i.e. friction. The membership functions of both are given below in triangular shapes. We are considering three membership functions for each input.

For distance from the front vehicle:

The range which such linguistic variable belong is from 10m to 100m.

- Small distance
- Medium distance, and
- High distance, with all having triangular membership functions are considered.

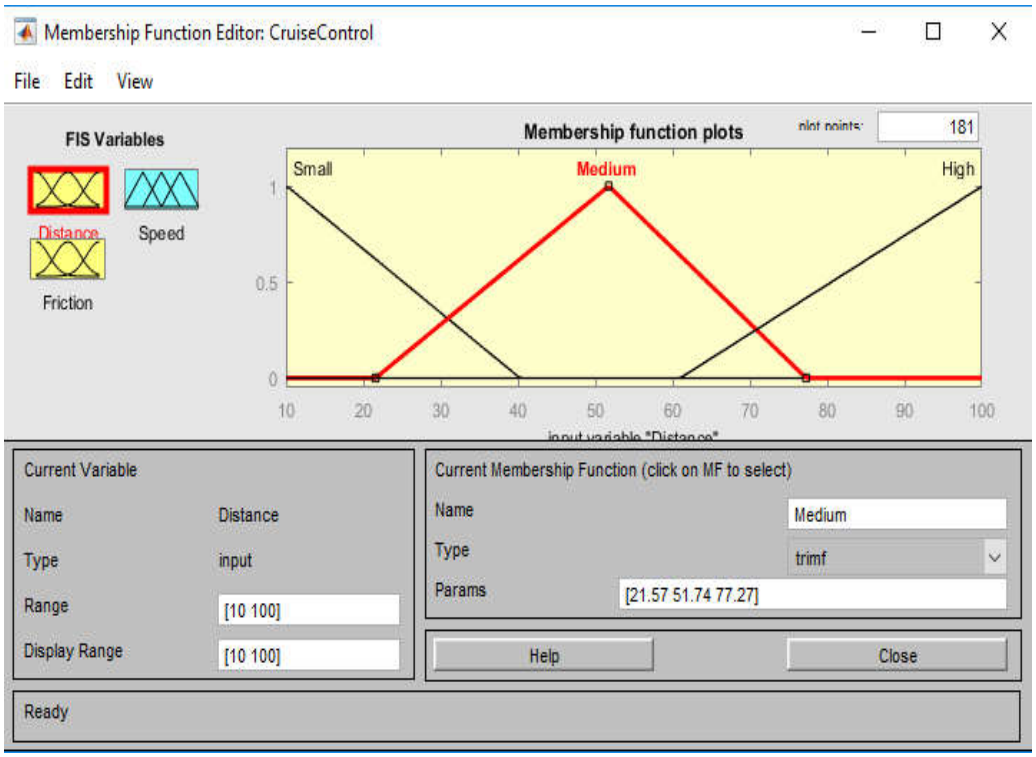


Figure 3.9 Input membership functions for distance

For friction:

Since, friction is the force that opposes motion between two objects that are in contact. And the rougher the surface, high friction we get than smooth surfaces.

The range which the friction value considered is from 0.1 to 1

- **Low** friction value, have low opposition force (Smooth surface)
- **Medium** friction value (partially smooth, and partially rough surfaces)
- **High** friction value (rough surfaces), which used to operate the brake to control the tire easily. And all linguistic terms having triangular membership functions.

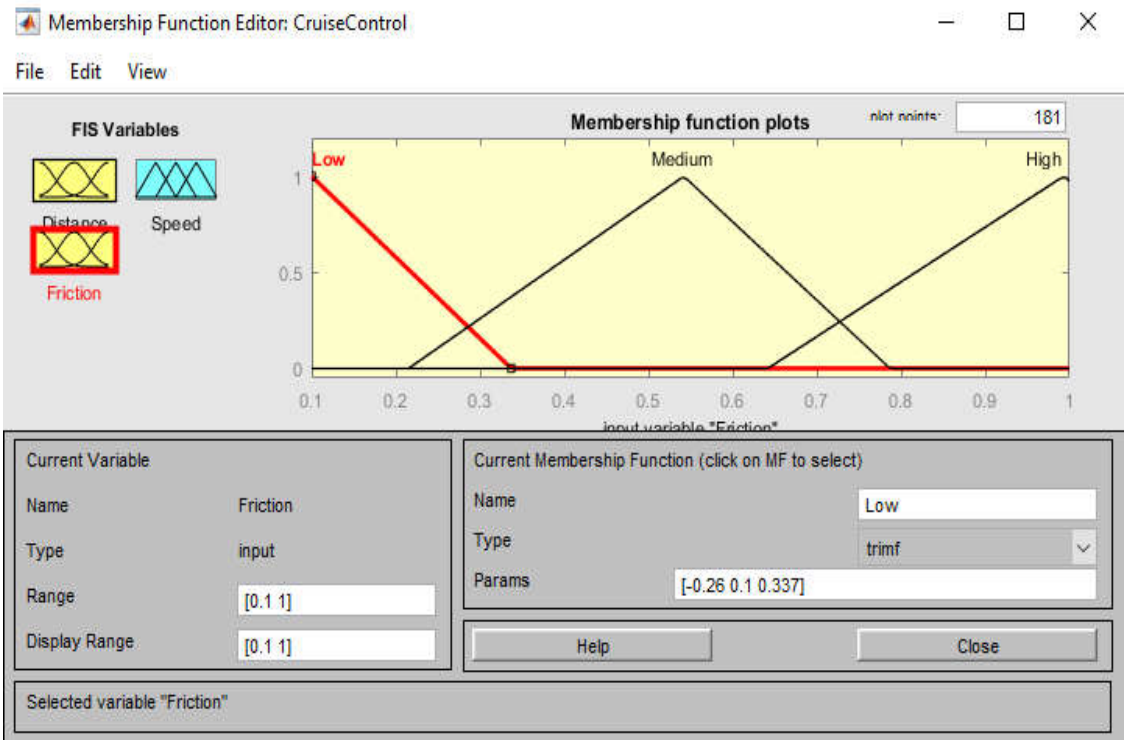


Figure 3.10 Input membership functions for friction

3.4.2.3. Output variable

The output variable speed (throttle control, control input of the plant) and it has its own linguistic term which it can be modified.

The range of the speed to be considered in our design is from 0 to 200Km/hr.

- Low speed
- Medium, and
- High speed with all having triangular membership functions.

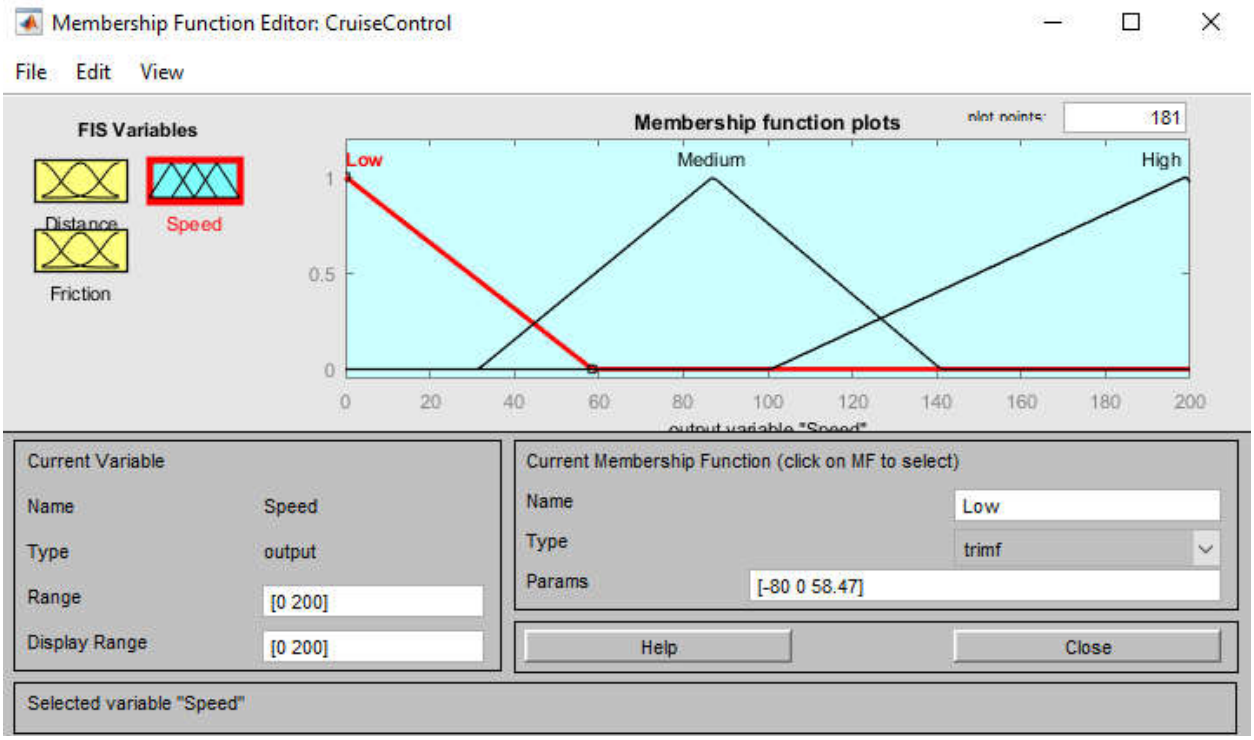


Figure 3.11 output membership functions for speed

3.4.2.4. Inference engine (Rule viewer)

As we have discussed before this section provides the rule generation by using fuzzy knowledge base.

3.4.2.4.1. Graphical User Interface (GUI) Tools

There are five primary **GUI** tools for building, editing, and observing fuzzy inference systems in the Fuzzy Logic Toolbox:

- Fuzzy Inference System (FIS) Editor
- Membership Function Editor
- Rule Editor
- Rule Viewer
- Surface Viewer

In Fuzzy Inference system we have two types of inference systems:

- Mamdani's inference method
- Sugeno inference method.

Since, Mamdani's fuzzy inference method, the most common method used in the inference engine to generate the rules.

Rules are influential in selecting the number of variables and membership functions to be modeled because the model becomes exponentially more complex as the number of variables or membership functions increases. This is because a rule must be available for each possible combination of input variable membership functions and potential outcome membership functions.

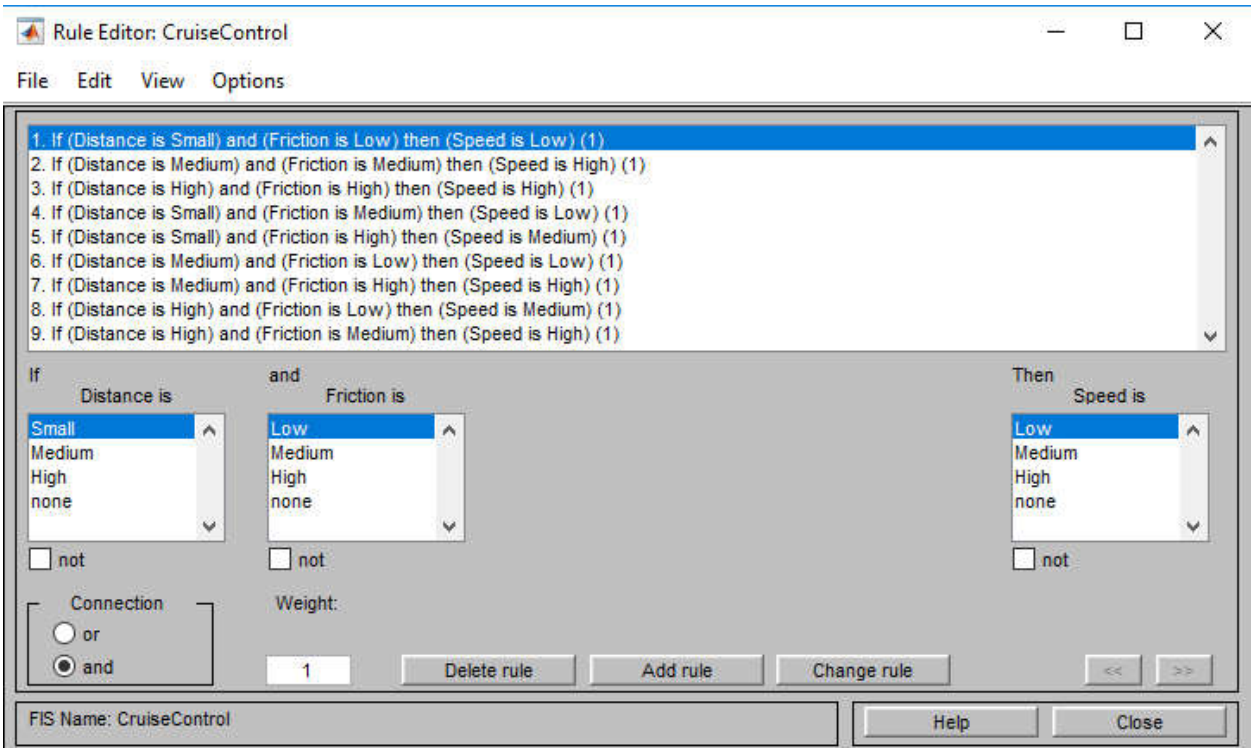


Figure 3.12 Rules generated from inference engine (rules editor).

Chapter Four

Result and Discussion

4.1. Introduction

In this chapter we try to briefly show the simulation result gained from Matlab/Simulink by using fuzzy logic controller and discuss about these results. Firstly, we proceed to show the result from fuzzy logic tool box configurations and then the Simulink model for the overall system.

4.2. Output result from Matlab

This section provides the output result which gained from Matlab by using the above designed input and output variables with their appropriate linguistic variables with their perspective membership functions. And we try to show the fuzzy logic-based rules generated from the front Matlab and Simulink result of the overall system. The following figure shows the rule result of such input variables from fuzzy logic toolbox.

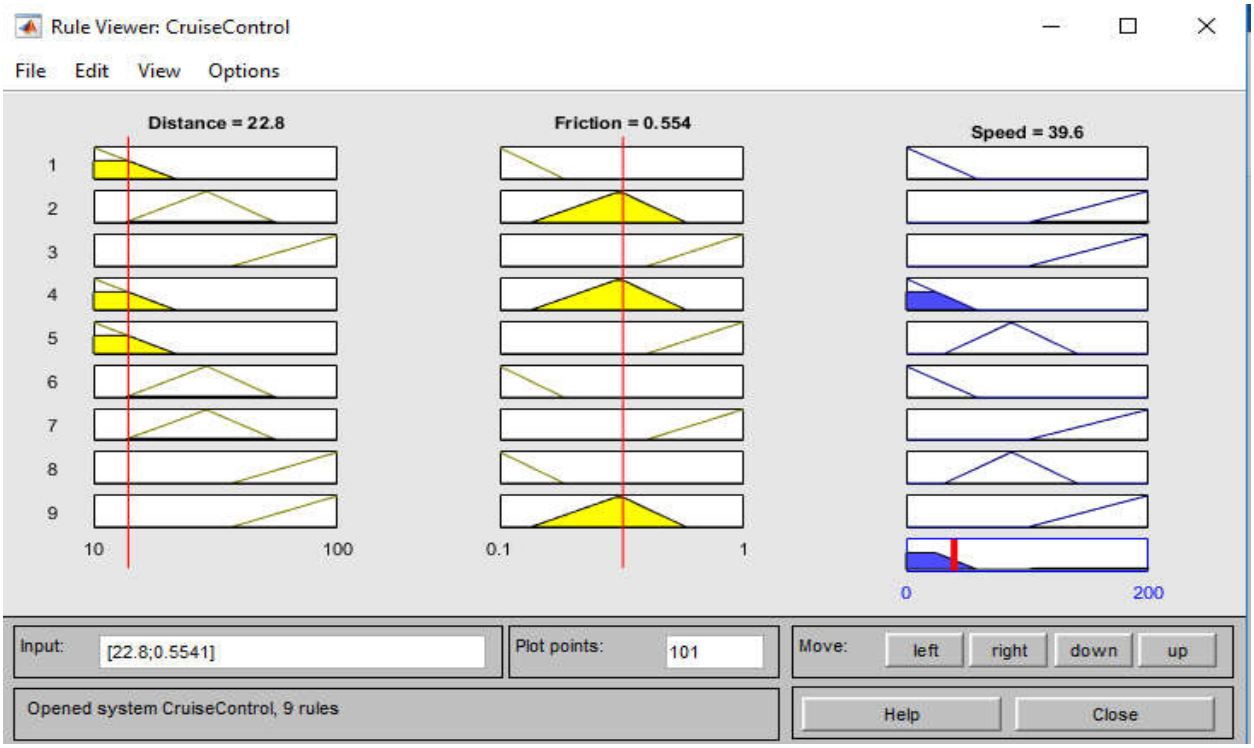


Figure 4.1 Fuzzy Output result (Rule viewer) at low speed

From the above diagram we have seen how to automatically control the speed (the throttle control) of the vehicle by adjusting the input variables. We know that the throttle setting is directly related with the speed of the vehicle, which implies that when we highly pressed the throttle setting the car tries to move at higher speed and vice versa. Since the output result is automatically gained by adjusting the input variable values of distance and friction and we can get low speed output.

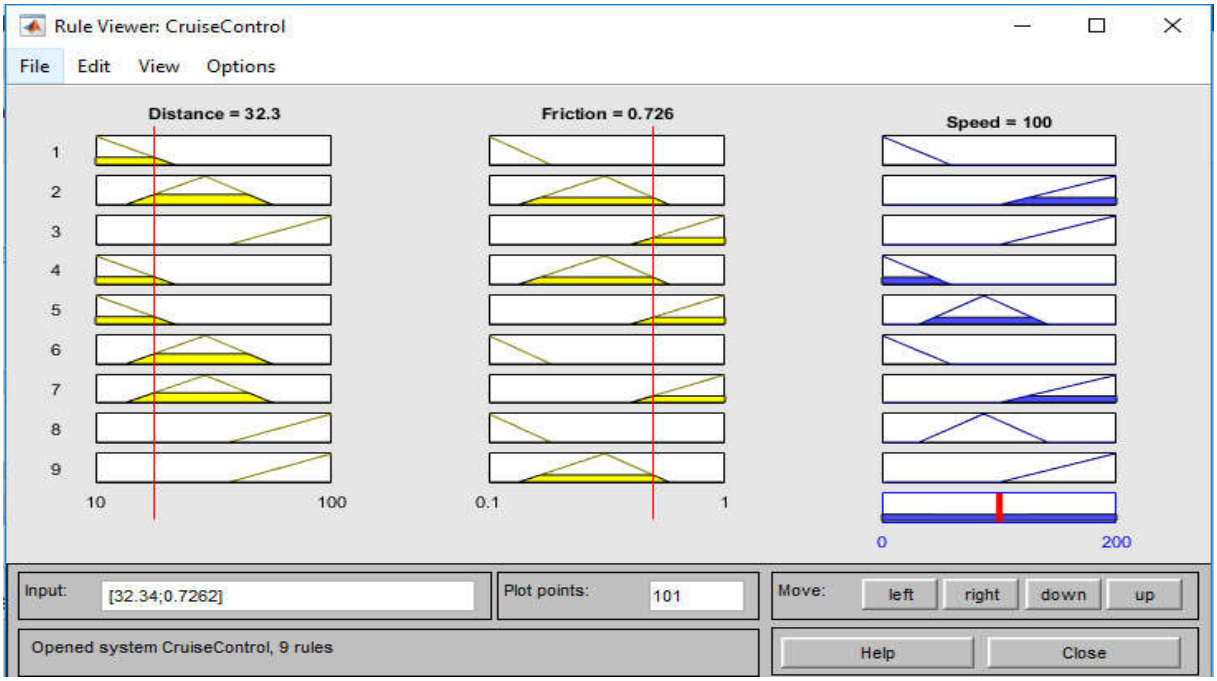


Figure 4.2 Fuzzy Output result (Rule viewer) at medium speed



Figure 4.3 Fuzzy Output result (Rule viewer) at high speed

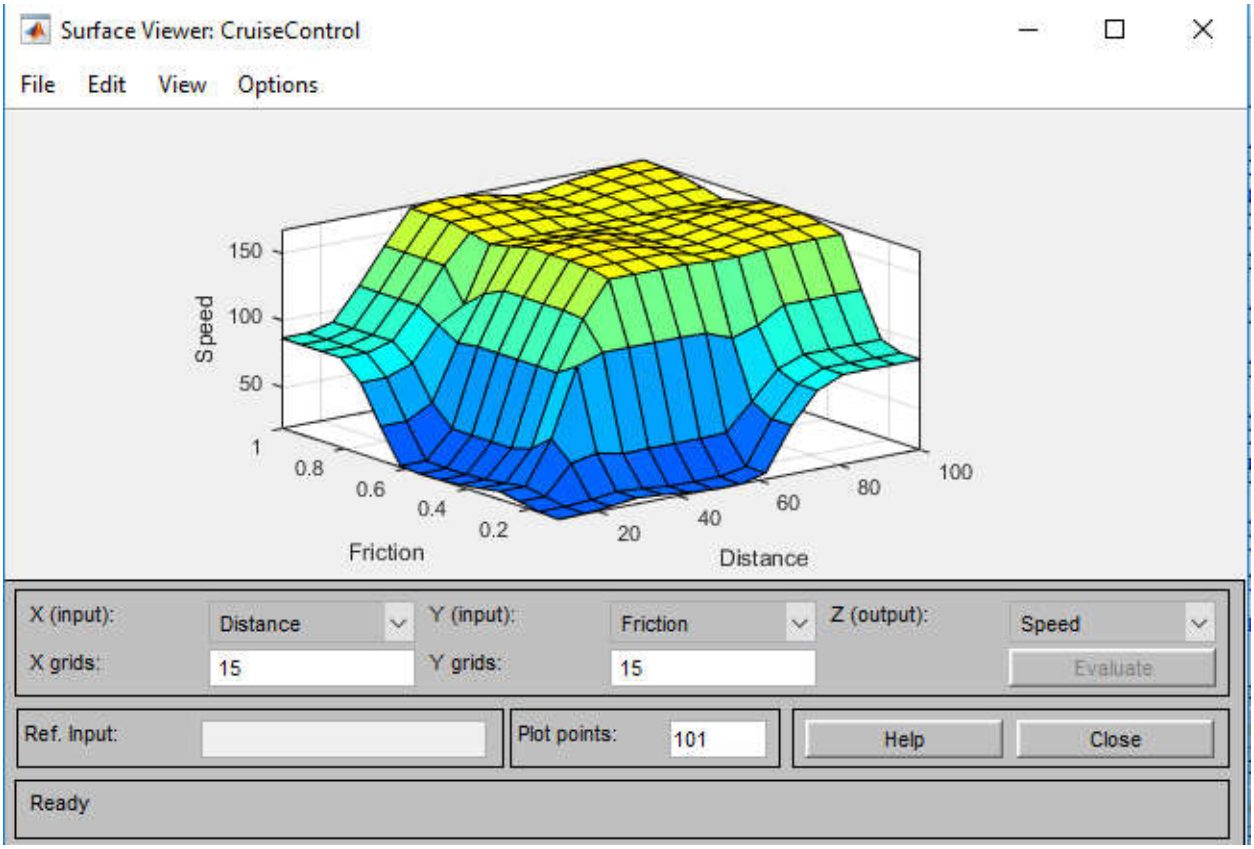


Figure 4.4 Fuzzy surface and output inferences for Fuzzy rules

From this diagram we have seen the surface of the system which is shows the monotonically increasing and decreasing values of the throttle setting (speed) based on the inputs.

For example: if the distance from the front vehicle is too small, and the awkwardness of the road (friction) is low, then speed of the vehicle becomes slow down (this implies the throttle is pressed slowly). And, if the distance is so far from the front vehicle and the friction is too high then the speed of the vehicle become monotonically increasing.

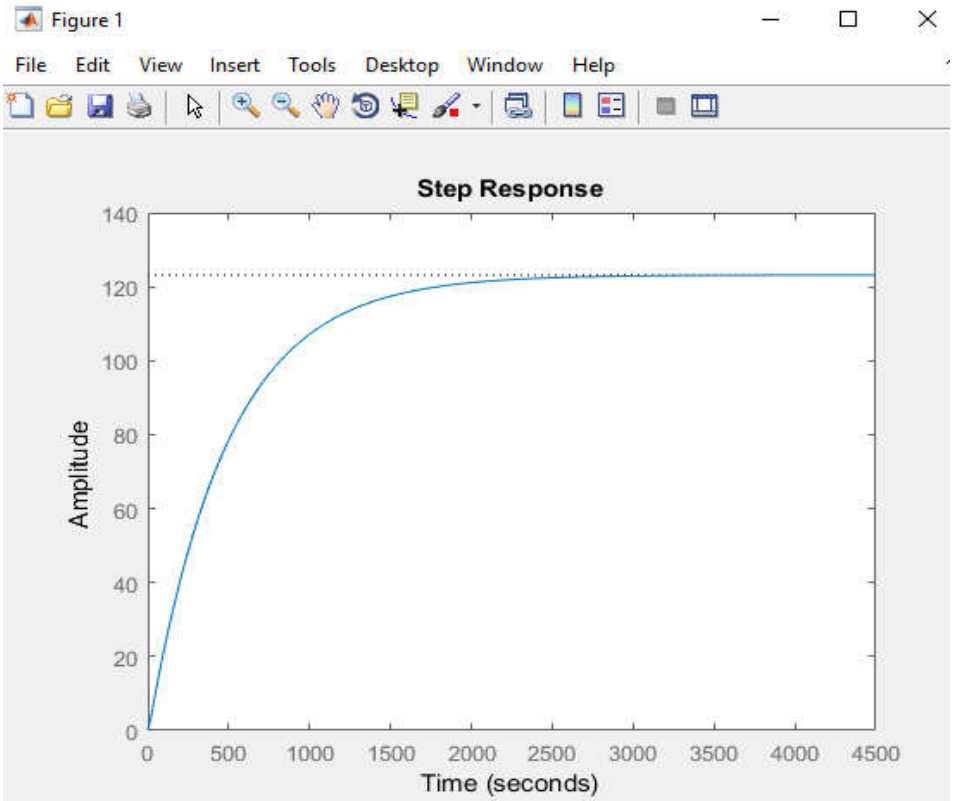


Figure 4.5 Uncontrolled responses

The above figure shows the response of the plant model without any controller applied on it.

4.3. Simulink model of the system

The Simulink model is created in Matlab under Fuzzy logic Toolbox with a feedback control system as shown below, is designed by following the steps which we show in the system block diagram. Since the Simulink model consists of the fuzzy logic controller, the plant, the step input the saturation (which gives the limit of the output), and scope. The input for fuzzy logic model is a unit step with the desired speed of 60 Km/h. The output of the model which is speed is plotted by the means of scope block.

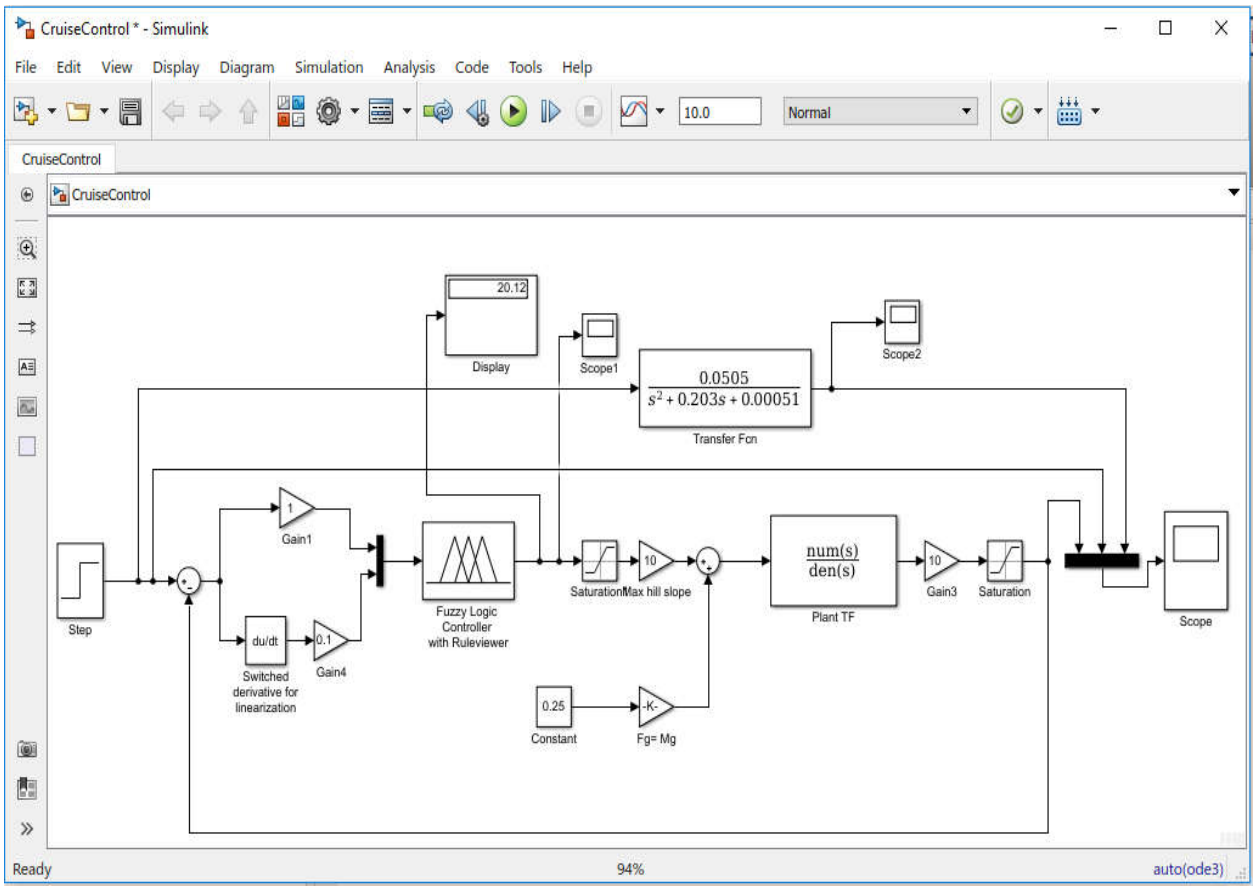


Figure 4.6 Simulink model of the system

From the above graph we have show Simulink model of the overall system under Matlab/Simulink software. The components of the block are directly related with the system block diagram, which are discussed in chapter three.

These are:

- ✓ Fuzzy logic controller
- ✓ The plant model (transfer function)
- ✓ Scope, used to display output signal
- ✓ Step input, the value that we give to the system as input
- ✓ Gain, which is added for system stabilizations
- ✓ Saturation ,which limits the output range

Generally, the system can only easily affected by the change of the plant model parameters and the controller; but the other components are added for system stabilization only.

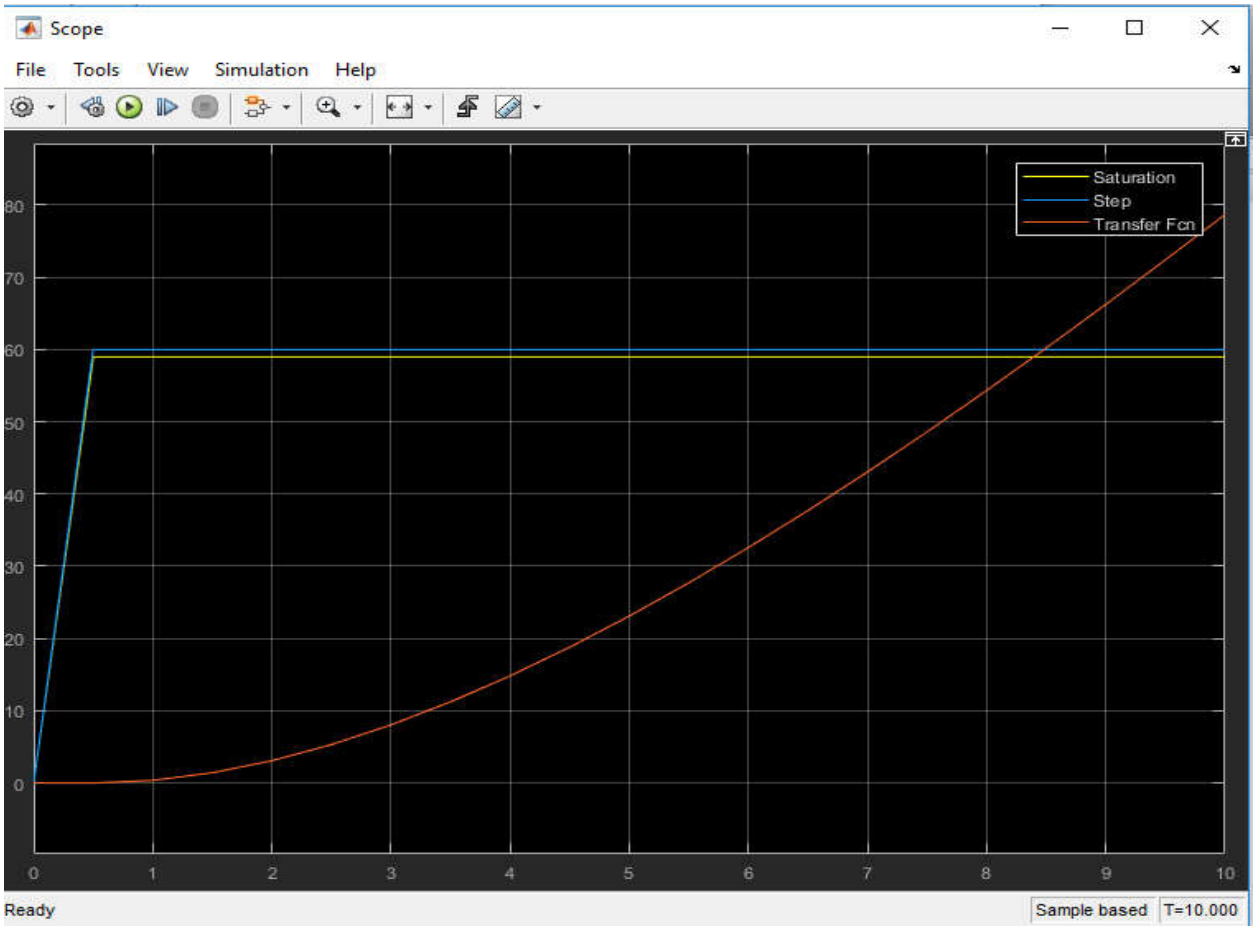


Figure 4.7 Simulation of overall system

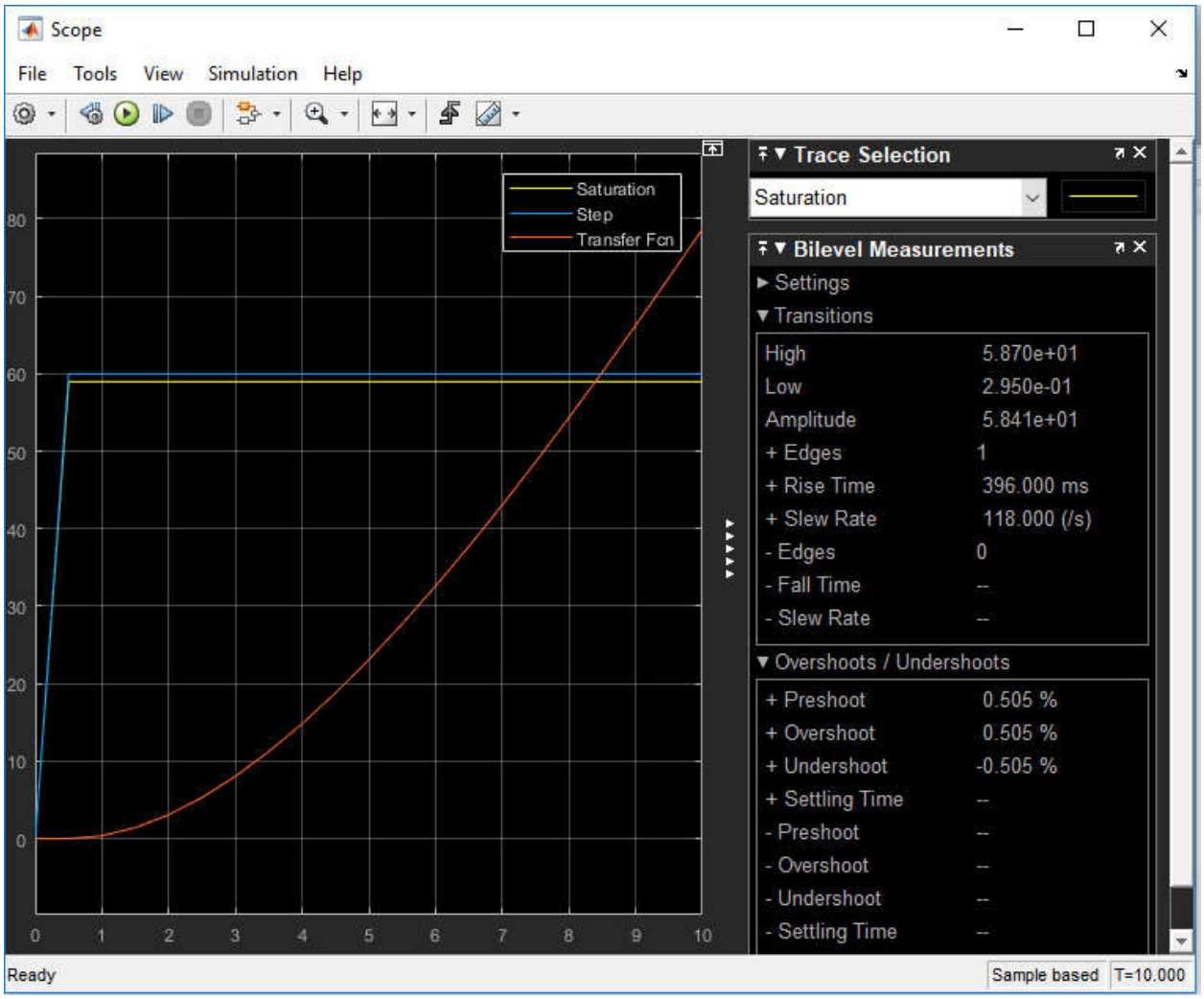


Figure 4.8 Simulation result with system specifications

The simulation output has three results, which represents:

Where: The Blue signal shows the step input

The Gray signal shows the plant transfer function (Uncontrolled plant)

The Yellow signal shows the controlled output

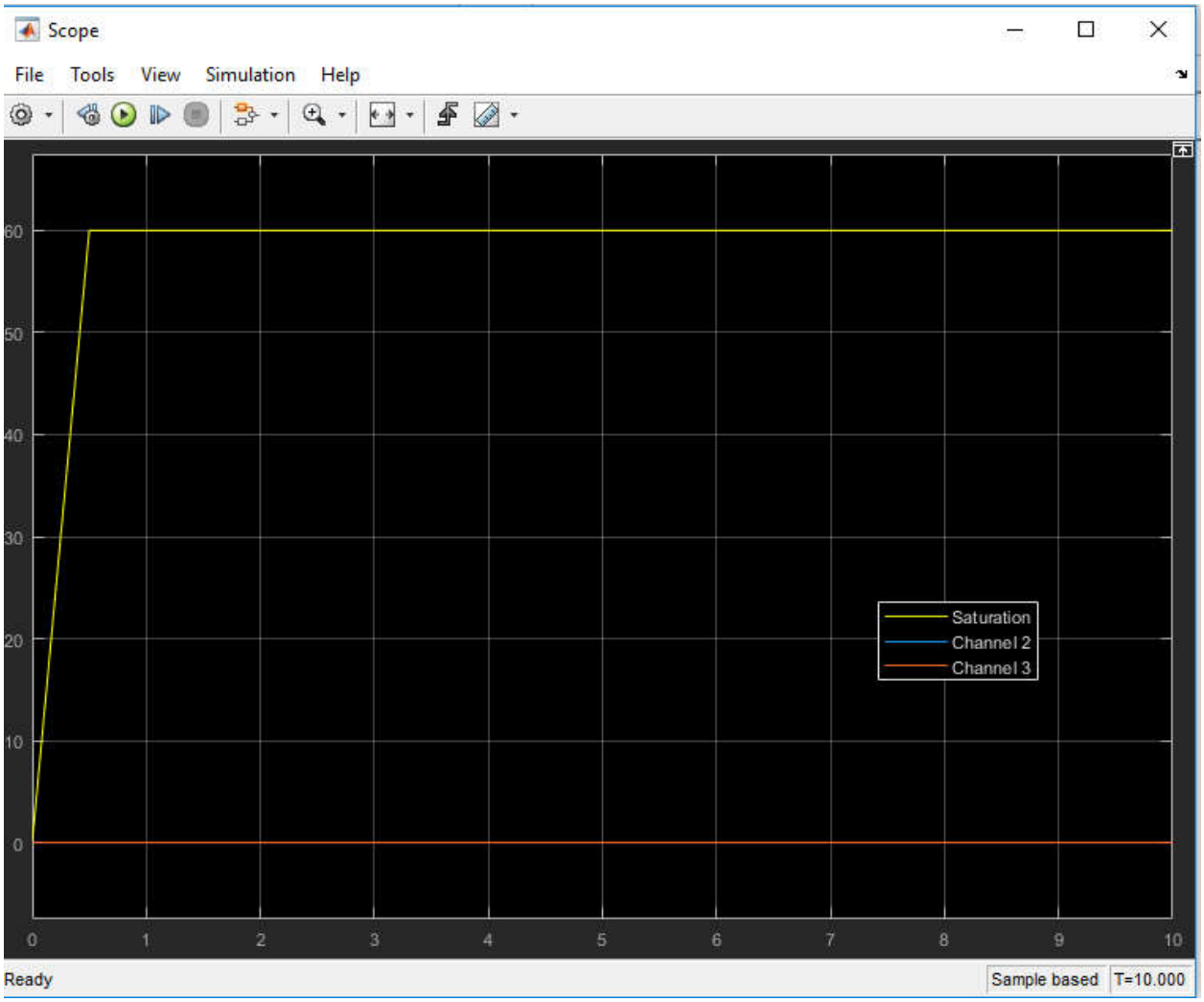


Figure 4.9 Speed control response using fuzzy logic

As we have seen the result of the system performance characteristics the percentage overshoot become reduced to 0.5% and rise time also 396.00 ms after applying fuzzy logic controller, Before make any decision of controller design, a few of design specification have been set. In this

design, we take two considerations to be met which are settling time *Ts* less than 1s and percentage of overshot *%OS* is less than 20% and rise time . So this advanced controller can control the system. We can conclude that the system is fully controlled.

Chapter Five

Conclusion and Recommendation

5.1. Conclusion

Automatic Cruise control system has designed and simulated using fuzzy logic controller to control the speed of the vehicle in desired manner. It has designed and simulated using Matlab/Simulink software. The mathematical model for cruise control system has been derived successfully. The controller has been designed for cruise control system. The characteristic of Fuzzy response has discussed. Better control performance and great stability has gained from fuzzy controller. Such fuzzy logic controller produced the responses with high rise time, but it offers low percentage overshoot and peak amplitude which results good performance of the system. The possibilities of what such a system can do seem nearly limitless. If every car on the road could one day be equipped with automatic cruise control, we could live in a world with drastically fewer traffic accidents than we ever have thought possible. Automatic cruise control has invented as a comfort or convenience item, it can eventually begin to double as a safety feature that changes the way we drive our cars.

5.2. Recommendation for future work

Though the designed system was able to automate cruise control, a few developments can be made on it to be more robust.

It involves:

- Extending the system to include a car with several sensors, to make the car speed control as smart as possible.
- Make the car self driving using different sensors and also adding radar control system, this will make cruise control system more advanced, which will have the ability to decrease the car accident.
- Generally this thesis can be made as smart as possible by adding extra features.

References

[1]. Khaled Sailan and Klaus.Dieter Kuhnert, “Modeling and design of cruise control system with feedforward for all terrian vehicles” pp. 339-349, (Mar. 2017).

[2]. AjeethaApparna, and Aishwarya A, “Adaptive cruise control for vehicle modelling using matlab”PP 82-88, Volume 12, Issue 2 Ver. II (Mar. – Apr. 2017).

[3]. Rohan Kumar and Rajan Pathak ,2012, Adaptive Cruise Control - Towards a Safer Driving Experience, International Journal of Scientific and Engineering Research Volume 3, Issue 8, August 2017.

[4]. V. V Sivaji, Dr. M. Sailaja,Adaptive Cruise Control Systems for Vehicle Modeling Using Stop and Go Manoeuvres,International Journal of Engineering,Research and Applications, Vol3 Issue4 ,(Jul. -Aug 2016).

[5]. G. Lai, Z. Liu, Y. Zhang, C. L. P. Chen, S. Xie, and Y.-J. Liu, “Fuzzy adaptive inverse compensation method to tracking control of uncertain nonlinear systems with generalized actuator dead zone,” *IEEE Transactions on Fuzzy Systems*, 2016.

[6]. R. C. Dorf, R. H. Bishop, Modern Control Systems, Ninth Edition. New Jersey: Prentice-Hall, pp. 366-420, Inc. 2001.

[7]. Y.-J. Liu and S. Tong, “Adaptive fuzzy control for a class of unknown nonlinear dynamical systems,” *Fuzzy Sets and Systems*, vol. 263, pp. 49–70, 2015.

[8]. C.-C. Tsai, S.-M. Hsieh, and C.-T. Chen, “Fuzzy longitudinal controller design and experimentation for adaptive cruise control and stop & Go, ”Journal of Intelligent and Robotic Systems: Theory and Applications, vol. 59, no. 2, pp. 167–189, 2010..

[10] Lee, C.-C., "Fuzzy logic in control systems: fuzzy logic controller-parts 1 and 2," IEEE Transactions on Systems, Man, and Cybernetics, Vol. 20, No. 2, pp 404-435, 1990.

[11]. Khairuddin Osman, Mohd. Fuaad Rahmat, Mohd Ashraf Ahmad, "Controller Design for a Cruise Control system," Kuala Lumpur Malaysia, 2009.

[12]. http//: www.ArmoredToyotahilux .com

7. Appendices

Appendix A

Datasheet on the vehicle we consider for our design

The Toyota Hilux - Specifications

ENGINE

	4x4 2.5 litre D-4D diesel	4x4 3.0 litre D-4D diesel
Engine code	2KD-FTV	1KD-FTV
Number of cylinders	4 in line	4 in line
Valve mechanism	16-valve DOHC	16-valve DOHC
Fuel injection system	Direct Injection common rail	Direct Injection common rail
Displacement (cc)	2494	2982
Bore & stroke (mm)	92.0 x 93.8	96.0 x 103.0
Compression ratio	17.4:1	17.9:1
Maximum output (DIN hp (KW)/rpm)	120 (88)/3600	171 (126)/3600
Maximum torque (Nm/rpm)	325/2000	343/1400–3400
Fuel tank capacity (litres)	80 (76 Extra Cab)	80
Drag coefficient	0.37	0.37

SUSPENSION

Front	Double wishbone with coil springs
Rear	Leaf springs with twin tube shock absorbers

TYRES AND WHEELS

	Single Cab 4x4 HL_2	Extra Cab 4x4 HL_2	Double Cab 4x4 HL_2	4x4 HL_3	4x4 Invincible
15" steel wheels w/205/70R15 tyres	–	–	–	–	–
15" steel wheels w/255/70R15C tyres	–	•	–	–	–
16" steel wheels w/205/R16 tyres	•	–	•	–	–
15" alloy wheels w/255/70R15C tyres	–	–	–	•	•

BRAKES

Front	Ventilated discs
Rear	Leading-Trailing Drums

TRANSMISSION

	HL_2	HL_3	Invincible
Gearbox type	5 M/T	5 M/T or 4 A/T (D-4D 170 only)	5 M/T or 4 A/T
Drive type	4x4	4x4	4x4

ENVIRONMENTAL PERFORMANCE

Fuel consumption (EU directive 80/1268-2004/3/EC)	4x4 2.5 litre D-4D diesel	4x4 3.0 litre D-4D diesel	4x4 3.0 litre D-4D diesel
	5 M/T	5 M/T	4 A/T
Fuel consumption combined (mpg)	34.0	34.0	30.1
Extra urban	39.2	39.8	35.3
Urban	28.0	27.4	23.9
Carbon dioxide CO_2 (g/km)	219/217 Single Cab	219	246

PERFORMANCE

	4x4 2.5 litre D-4D diesel	4x4 3.0 litre D-4D diesel	4x4 3.0 litre D-4D diesel
	5 M/T	5 M/T	4 A/T
Maximum speed (mph)	96.3	105.6	108.7
0–62 mph (sec.)	15.0 (SC,EC), 15.2 (DC)	12.0	11.9

DIMENSIONS

	Single Cab 4x4 HL_2	Extra Cab 4x4 HL_2	Double Cab 4x4 HL_2	4x4 HL_3	4x4 Invincible
Dimensions					
Overall length (mm)	5255	5255	5130	5255	5255
Overall width excluding mirrors (mm)	1760	1835	1760	1835	1835
Overall height unladen (mm)	1795	1795	1810	1810	1810
Wheelbase (mm)	3085	3085	3085	3085	3085
Front tread (mm)	1510	1540	1510	1540	1540
Rear tread (mm)	1510	1540	1510	1540	1540
Minimum turning circle kerb to kerb (m)	12.2	12.2	12.2	12.2	12.2
Minimum turning circle wall to wall (m)	13.0	13.0	13.0	13.0	13.0
Deck Space					
Deck height from ground unladen (mm)	860	855	850	850	850
Internal deck height (mm)	450	450	450	450	450
Internal deck width (mm)	1520	1515	1515	1515	1515
Internal deck length (mm)	2315	1805	1520	1520	1520
Width between wheel arches (mm)	1100	1100	1100	1100	1100
Weights					
Kerb weight (kg)*	1820	1830	1880 1895	1880– 1890	1885–
Gross vehicle weight (kg)	2680	2875	2940	2940	2940
Max. payload (kg)*	860	1045	1060 1060	1050– 1055	1050–
Axle load limit, front (kg)	1280	1280	1305	1305	1305
Axle load limit, rear (kg)	1600	1610	1635	1635	1635
Roof load limit (kg)	–	–	75	75	75
Towing Capacity					

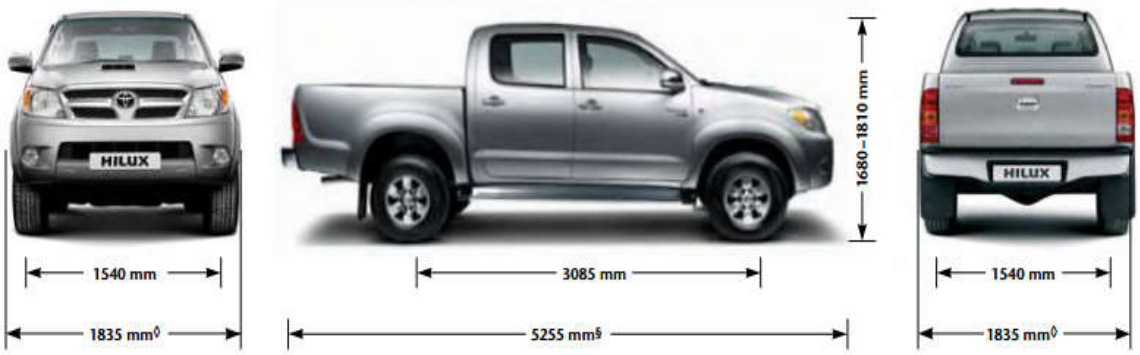
Braked trailer weight (kg)	2250	2250	2250	2250	2250
Unbraked trailer weight (kg)	750	750	750	750	750
Max. nose weight on tow ball (kg)	110	110	110	110	110
Gross train weight (kg)	4930	5125	5190	5190	5190

* Kerb weight and max. payload include a full tank of fuel and all fluids. They exclude driver, passengers and other added accessories.

All kerb masses quoted are subject to manufacturing tolerances and are for vehicles with minimum equipment.

§ 5130 mm HL_2 Double Cab with no rear step.

◊ 1760 mm HL_2 grades.



Please note: all width measurements given exclude door mirrors.

TODAY TOMORROW TOYOTA