

**DEVELOPMENT OF A CAR-FOLLOWING MODEL FOR  
TRAFFIC STREAMS HAVING WEAK LANE DISCIPLINE**

**By**

**FATIMA AFIFAH**

**Student ID No.: 0416042413**

A thesis submitted to the Department of Civil Engineering,  
Bangladesh University of Engineering and Technology, Dhaka,  
in partial fulfillment of the requirements for the degree of

**MASTER OF SCIENCE IN CIVIL ENGINEERING  
(TRANSPORTATION)**



**Department of Civil Engineering**  
**BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY**  
**(BUET)**

---

**AUGUST, 2019**



The thesis/project titled “Development of a Car Following Model for Traffic Streams having Weak Lane Discipline” submitted by FATIMA AFIFAH, Student ID No.: 04016042413, Session: April 2016, has been accepted as satisfactory in partial fulfillment of the requirement for the degree of Master of Science in Civil Engineering (Transportation) on 6<sup>th</sup> August, 2019.

### BOARD OF EXAMINERS

---

**Dr. Md. Shamsul Hoque**  
Professor,  
Department of Civil Engineering,  
BUET, Dhaka-1000.

**Chairman  
(Supervisor)**

---

**Dr. Ahsanul Kabir**  
Professor & Head,  
Department of Civil Engineering,  
BUET, Dhaka-1000.

**Member  
(Ex-Officio)**

---

**Dr. Md. Hadiuzzaman**  
Professor,  
Department of Civil Engineering,  
BUET, Dhaka-1000.

**Member**

---

**Dr. Md. Shahid Mamun**  
Professor,  
Department of Civil Engineering  
Ahsanullah University of Science and  
Technology (AUST)

**Member  
(External)**

## DECLARATION

I hereby declare that the research work presented in this thesis has been performed by me and this thesis or any part of it has not been submitted elsewhere for any other purposes except for publication.

August, 2019

*Fatima Afifah*

Fatima Afifah

## ACKNOWLEDGEMENT

All praise to Almighty Allah, the most Gracious and most Merciful.

The author would like to express her sincere gratitude to her supervisor Dr. Md. ShamsulHoque, Professor, Department of Civil Engineering, Bangladesh University of Engineering & Technology (BUET), for his continuous guidance, invaluable suggestions and affectionate encouragement all stages of this study. Without this valuable direction and cordial assistance, this research work could never be materialized. The author's debt to him is immense.

The author would like to thank all the academic staff of Department of Civil Engineering, BUET including those of Transportation Engineering Division for their dedication and inspiration. Special thanks to Md. Rafiqul Islam, Senior Lab Instructor, Department of Civil Engineering BUET, who helped out greatly during the initial stage of thesis approval in BUET.

The author is also grateful to her brother A. B. M. Irad Hossain who encouraged her immensely and guided her whenever needed.

The author is grateful to the authors of different articles mentioned in the reference which proved to be very helpful throughout the whole thesis work.

The author is also grateful to Umma Hafsa and Abdul Mukit, students of University of Asia Pacific, who contributed during data collection phase of this thesis.

Finally, the author is willing to show her solemn gratitude to her parents for their continuous support and motivation throughout the thesis work.

## ABSTRACT

In recent years, Traffic engineers widely use microscopic traffic simulation tools to evaluate traffic operational problems. It is due to the ability of traffic simulation tool to generate with different traffic problem in different scenario without even disrupting traffic condition on the road. Car following model is one of the most important driver behavior model that composes traffic simulation model. The road traffic situation of Bangladesh is heterogeneous in nature, where vehicles vary widely in static and dynamic characteristics and share the same width of carriageway. Therefore, the car following behavior under this condition is expected to be quite different from that of homogeneous or car dominated traffic. This justifies the development of car following models for heterogeneous road traffic condition under weak lane discipline.

However, in reality, the car following behavior of a driver is influenced by a vague perception of the stimuli and his decision making mechanism is driven by fuzzy rules gained by knowledge and experience. Therefore, the deterministic models fail to emulate the complex and multi ruled behavior of the driver in an uncertain environment of the road. Considering these issues, this study aims to develop a Neuro-Fuzzy model to depict car following behavior in heterogeneous road traffic condition.

In this study, car following data have been collected in the form of video footage of vehicular movements at one of the busiest roads of Dhaka city. From the video footage, car following parameters have been extracted by converting the two dimensional coordinates of vehicle trajectories of the video footage into real world positions or into pixel coordinates as required for different parameters along the road using Python programming language and OpenCV library. From these positions, various car following parameters such as distance, speed and acceleration were calculated. A car-car following model was developed by using the 'Neuro-Fuzzy Designer' toolbox of MATLAB. The model was calibrated through the Adaptive Neuro-Fuzzy Inference System of the same toolbox, in order to alter the membership functions such that the models are able to replicate the real scenario of vehicular movement accurately.

In order to validate the models, at first few criteria were defined to perform statistical analysis of the proposed model. Furthermore, two new models were introduced and

statistical analysis was done for the new models. At last, performance of the proposed model was observed by comparing the errors of all the models.

## TABLE OF CONTENTS

DECLARATION .....	i
ACKNOWLEDGEMENT .....	ii
ABSTRACT.....	iii
TABLE OF CONTENTS.....	v
LIST OF TABLES .....	vii
LIST OF FIGURES .....	viii
CHAPTER 1 INTRODUCTION .....	1
1.1 Background .....	1
1.2 Objectives of the Study .....	4
1.3 Scope of the Study.....	4
1.4 Organization of the Thesis .....	5
CHAPTER 2 LITERATURE REVIEW .....	6
2.1 Introduction .....	6
2.2 Traffic Simulation Models .....	6
2.2.1 Car Following Models .....	7
2.3 Adaptive Neuro –Fuzzy Interface System.....	15
2.3.1 Fuzzy Logic .....	16
2.3.2 Neural Network.....	20
2.3.3 ANFIS Architecture .....	22
2.4 Limitation of State-of –the –art models .....	26
2.5 Summary .....	27
CHAPTER 3 METHODOLOGY AND DATA COLLECTION .....	28
3.1 Introduction .....	28
3.2 Neuro-Fuzzy Models.....	28
3.2.1 Advantages of Neuro-Fuzzy Models .....	28
3.3 Data Collection Techniques .....	29
3.3.1 Wire linked Vehicles Technique.....	30
3.3.2 Photographic and Video Techniques .....	30
3.3.3 Instrumented Vehicle Techniques:.....	31
3.3.4 The Use of Global Positioning System:.....	31
3.4 Site Selection for Data Collection .....	32
3.4.1 Site Selection Criteria .....	32
3.4.2 Reconnaissance Survey.....	33
3.4.3 Final Site Selection .....	35
3.5 Collection of Data .....	37
3.5.1 Video Recording .....	37
3.5.2 Auxiliary Data.....	38



3.6 Summary .....	39
CHAPTER 4 DATA ANALYSIS AND MODEL DEVELOPMENT .....	40
4.1 Introduction .....	40
4.2 Data Analysis .....	40
4.2.1 Data Extraction Methodologies .....	40
4.2.2 Image Processing with OpenCV .....	42
4.3 Fuzzy Inference System Development.....	51
4.3.1 Inputs and Outputs .....	52
4.3.2 Membership Functions.....	57
4.3.3 Fuzzy Rules.....	62
4.3.4 Neural Network Representation.....	68
4.3.5 Model Calibration .....	69
4.4 Summary .....	77
CHAPTER 5 VALIDATION .....	78
5.1 Introduction .....	78
5.2 Validation Methodology.....	78
5.3 Validation of the Proposed Model.....	79
5.3.1 Definition .....	79
5.3.2 Statistical Analysis of Proposed Model .....	83
5.3.3 Introduction of New Models .....	86
5.3.4 Performance Comparison.....	93
5.4 Summary .....	95
CHAPTER 6 CONCLUSIONS AND RECOMMENDATIONS .....	96
6.1 Conclusions .....	96
6.2 Observations on the Models Developed.....	96
6.3 Constraints of the Research.....	98
6.4 Recommendations for Future Research .....	99
REFERENCES .....	101
APPENDIX A .....	109
APPENDIX B .....	117

## LIST OF TABLES

<b>Table 4-1:</b>	Summary of Input Parameters .....	56
<b>Table 4-2:</b>	An Example of all the Input and Output that was used to Develop the Model.....	57
<b>Table 4-3:</b>	Predicted Outputs Generated by All the Membership Functions .....	57
<b>Table 4-4:</b>	Pre-calibration Parameters Pertaining to the Input Membership Functions .....	61
<b>Table 4-5:</b>	List of Fuzzy Rules Used in the study .....	64
<b>Table 4-6:</b>	Post-calibration Parameters Pertaining to the Input Membership Functions .....	73
<b>Table 4-7:</b>	Post-calibration Parameters Pertaining to the Output Membership Functions .....	74
<b>Table 5-1:</b>	Statistical Analysis of Proposed Model.....	84
<b>Table 5-2:</b>	Statistical Analysis of Stimulus-response Model .....	88
<b>Table 5-3:</b>	Details of Different Types of Dataset.....	90
<b>Table 5-4:</b>	The R and MSE for different type of Dataset.....	91
<b>Table 5-5:</b>	Statistical Analysis of Neural Network .....	92
<b>Table 5-6:</b>	Summery of all the performances of all the models.....	94

## LIST OF FIGURES

<b>Figure 2-1:</b>	Fuzzy Inference System .....	17
<b>Figure 2-2:</b>	Schematic diagram of a Biological Neuron .....	20
<b>Figure 2-3:</b>	A Feed Forward ANN with Input, Output and One Hidden Layer.....	21
<b>Figure 2-4:</b>	Different types of Activation Functions: (a) Threshold (b) Piecewise Linear (c) Gaussian (d) Sigmoid (e) Tan-hyperbolic .....	21
<b>Figure 2-5:</b>	ANFIS Architecture .....	23
<b>Figure 2-6:</b>	ANFIS structure .....	26
<b>Figure 3-1:</b>	Reconnaissance Survey in Different Roads of Dhaka City.....	34
<b>Figure 3-2:</b>	The View of the Final Site Selected for Survey.....	36
<b>Figure 3-3:</b>	Satellite view of the selected site .....	36
<b>Figure 4-1:</b>	Grayscale Image.....	44
<b>Figure 4-2:</b>	Blurred Image.....	45
<b>Figure 4-3:</b>	Hue, Saturation & Value .....	46
<b>Figure 4-4:</b>	The Upper and Lower Range for Hue, Value and Saturation is shown in form of Movable Bars .....	47
<b>Figure 4-5:</b>	Morphological Transformation .....	48
<b>Figure 4-6:</b>	Final contoured Area of the Bus .....	49
<b>Figure 4-7:</b>	Steps to Image Processing.....	49
<b>Figure 4-8:</b>	Imported data in the “Neuro-Fuzzy Designer” toolbox .....	54
<b>Figure 4-9:</b>	Formation of Gaussian Membership Function.....	59
<b>Figure 4-10:</b>	Choosing the type of membership function .....	59
<b>Figure 4-11:</b>	Initialization of Input Membership Functions.....	60
<b>Figure 4-12:</b>	Rule editor .....	63
<b>Figure 4-13:</b>	Neural Network Representation of the FIS .....	68
<b>Figure 4-14:</b>	Plot of Training Error against Epoch Number .....	71
<b>Figure 4-15:</b>	Plot of Model Outputs and Data set Outputs against Data Set Index ..	72
<b>Figure 4-16:</b>	Membership Function after Calibration (input: LLDS) .....	73
<b>Figure 4-17:</b>	Rule Viewer after Calibration .....	76
<b>Figure 4-18:</b>	Surface Viewer Showing the Output Surface .....	77
<b>Figure 5-1:</b>	Fitness of Data against Regression line.....	81
<b>Figure 5-2:</b>	The validation dataset being loaded in the trained model.....	84
<b>Figure 5-3:</b>	Fitness of the proposed ANFIS model .....	86
<b>Figure 5-4:</b>	(on left) Training Error of the Stimulus-response Model & (on right) ANFIS model structure for that model.....	87
<b>Figure 5-5:</b>	(on left) Plot of Trained Model Output against Data Set Index & (on right) Surface Viewer Showing the Output Surface.....	87
<b>Figure 5-6:</b>	Fitness of new ANFIS model .....	89
<b>Figure 5-7:</b>	Representation of Neural Network.....	90
<b>Figure 5-8:</b>	Error Histogram and Performance of the Neural Network .....	92
<b>Figure 5-9:</b>	Fitness of the NN prediction model .....	93

# CHAPTER 1

## INTRODUCTION

### 1.1 Background

Traffic engineers and transport professionals use microscopic traffic simulation tools to evaluate traffic operational problems and intelligent transportation system (ITS) applications. The ITS systems include collision warning systems, adaptive traffic management, traveler information and incident management systems. These systems are difficult to evaluate using analytical tools due to complex system dynamics involved in these applications. In traffic simulation tool, different scenarios can be generated with the roadway geometric and traffic data and traffic problems can be evaluated without disrupting traffic conditions on the road. Traffic simulation tools are composed of several driver behavior models including car-following, lane-change and travel path. In particular, car-following models have significant impact on the accuracy of the traffic simulation tools in replicating traffic behavior on the road.

Car following models describe how vehicles interact with each other in a traffic stream. Many researchers have been conducted for homogeneous traffic condition particularly in developed countries. The developed models may be grouped as car-following model (e.g. Chandler et al, 1958; Gazis et al., 1959; Edie, 1961, Gazis et al., 1961; Herman and Rothery, 1965; Ozaki, 1993; Ahmed, 1999, Choudhury et al, 2009, etc.), psychophysical model (e.g. Brackstone et al., 2002, etc.), fuzzy-logic model (e.g. Kikuchi and Chakroborty, 1992; Brackstone et al., 1997, Chakroborty and Kikuchi, 2003, etc.) and general acceleration model (e.g. Gipps, 1981; Ahmed, 1999). The models range from simplest car-following model (Chandler, 1958) with minimum parameters, to comparatively complex and comprehensive model with multi-variable consideration.

The state-of-the-art car following models has their own limitations. According to Kim et al. (2003), some models (e.g. Chandler, 1958; Gazis, 1958; Edie, 1961; Herman and Rothery, 1965, Gipps, 1981, etc.) are based on the assumption that drivers follow the same driving rules. The reality is, these rules may differ with different drivers, or even for the same driver under different circumstances, and in fact, possibly with the same driver and nearly identical situations. Some other limitations include absence of

significant stochastic components such as, ignorance of factors beyond vehicle kinematics and ignorance of past sequence of car motion that produce the current state and the inadequacy to represent the normal car-following behavior.

The aforementioned models are developed in homogeneous traffic condition. Despite being the state-of-the-art car following models, these models fail to address weak lane discipline and presence of non-motorized traffic (therefore difference in speed between the motorized and non-motorized vehicles) which depicts the real situation in heterogeneous traffic stream.

It is unfortunate that, the studies so far done on modeling heterogeneous traffic stream is quite scarce for its demand. Mallikarjuna (2007), on his research, has focused on the analysis and modeling of heterogeneous traffic observed on mid-block sections of urban and rural roads of India. Arasan and Koshy (2005) described a modeling methodology adopted to simulate the flow of heterogeneous traffic with vehicles of wide ranging static and dynamic characteristics. The studies conducted by Sayer et al. (2003), Ravishankar and Mathew (2011) have endeavored to identify the effect of lead-vehicle size on driver-following behavior in mixed traffic stream. Studies have even been conducted attempting to conjugate the longitudinal movement with lateral movement to define the lateral discomfort while longitudinal movement is in action (e.g Gunay, 2007).

In recent years, Islam (2013) has developed an acceleration model traffic streams having weak lane discipline using comprehensive disaggregate (e.g. second by second trajectory data) and aggregate data (e.g. traffic counts and speeds) and calibrated it using econometric approaches. Imran (2009), on his research, has focused on developing car-following models for mixed traffic using fuzzy-logic inference system of MATLAB. Earlier, there has been research on modeling heterogeneous traffic operations at two corridors of Dhaka city (e.g. Hossain, 1996). In this study, the effect of different composition of non-motorized traffic on corridor travel time and passenger movement capacity has been analyzed based on the developed simulation model (MIXNETSIM) (Hossain, 1996). Moreover, some factors have been identified (e.g. traffic density) which may result in higher delay for traffic for larger size vehicles. The research of Hoque (1994) was mainly on modeling of signalized intersections of Dhaka city. This model (MIXSIM) has been used in computation of PCU conversion factor and evaluation of alternative schemes to improve the capacity of an approach serving mixed traffic flow.

Many sophisticated computing techniques have been developed to revolutionize various field of transportation engineering (e.g. microscopic and macroscopic traffic simulation model). Microscopic traffic simulation models are convenient in assessing performance of traffic operations in various traffic conditions or environmental conditions in a cost effective way without disrupting the traffic. The car following theory that is of interest for this study suggests that other things being equal, the acceleration or deceleration of a following vehicle will directly be influenced by the stimuli induced by actions of the leading vehicle. Basically, this theory is based on the dynamic behavior of two consecutive vehicles moving on the same roadway without overtaking. According to Brackstone and McDonald (1999), this adjustment takes place after a time delay, which is a function of the reaction time of the follower and the vehicle's acceleration or deceleration capability.

Most of the models describing car following behavior are more or less deterministic in nature, which are unable to emulate the complex and multi ruled behavior of the driver in an ambiguous environment of the road. In fact, the car following behavior of a driver is influenced by a vague perception of the stimuli and his decision making mechanism is driven by fuzzy rules gained by knowledge and experience. Contrary to deterministic approaches, a Fuzzy Inference System (FIS) based approach is the one that can investigate any uncertain or ambiguous matter in terms of a quantifiable degree of uncertainty to reflect subjective or vague perception of human reasoning by expressing the variables in linguistic terms. For better optimization, an enhancement to this approach is to represent the FIS by an artificial neural network, which is structured in such a way that the parameters of the fuzzy sets and the logical operations are exactly replicated in the weights and operations of the neural network.

In Bangladesh, traffic stream follows weak-lane discipline even though almost in every road, lanes are well-defined with proper road markers. A common scenario in the roads of Bangladesh is, smaller vehicles (such as rickshaws, motorcycles, etc.) fill the gap between larger vehicles (such as buses, passenger cars, etc.) It so happens that a section of a road may contain five or even more vehicles depending on the size of vehicles. As a result, a driver not only follows the leader vehicle but also has to mind the gap between the vehicles on both sides (depending on the presence of any vehicle on both or either sides). Therefore, car-following behavior in heterogeneous traffic stream with weak lane discipline is believed to be represented effectively by addressing a new parameter (lateral gaps) with the existing parameters. Taking into account of the capability of artificial neural network and fuzzy logic to model car following behavior in a realistic manner, there is a huge prospect to use these

platforms which may unveil the new challenges and usher the way for new thoughts and techniques to proceed further.

## **1.2 Objectives of the Study**

The overall objective of this study is to develop a car following model by using the framework of Adaptive Neuro-Fuzzy Inference System (ANFIS), in order to model the aspect of car following behavior of heterogeneous vehicular traffic with special focus on situation with weak lane discipline and calibrate the model parameters using detailed data collected from selected locations of Dhaka city. The specific objectives are recapitulated as follows:

- a) To develop fuzzy rules for various types of vehicles in order to build car-following model structure
- b) Analysis of comprehensive aggregate data (e.g. speeds) which will be used for model development
- c) To calibrate the proposed model using Adaptive Neuro-Fuzzy Inference System.
- d) To validate the proposed models by using proper statistical analysis.

## **1.3 Scope of the Study**

The scope of this study is to develop a car following model. The model will be Neuro-Fuzzy model in Bangladesh condition, that is, under heterogeneous road traffic condition in a weak lane discipline. The comprehensive data will be collected from a busy location of this city and the data will cover the necessary information to identify the position of vehicles. In the first stage, the data will be used to generate the speed and acceleration of vehicles.

At the second stage, the parameters will be estimated using the comprehensive data and a Neuro-Fuzzy model will be developed. Later the model will be calibrated using artificial neural network. The model does not take into account of vehicular maneuvers other than following the leader such as lane changing, overtaking etc. The model not only considers the longitudinal movement of the vehicle but also adds lateral movement models to understand the driving pattern in a more logical way. The combined model created, then can be run in simulation tools to replicate the driving behavior for mixed traffic conditions with varying proportion of motorized and non-

motorized traffic. The proposed model can further be tuned and used for non-motorized vehicles. The transport planners and policy makers may be interested in such practice to make new plans or policy for traffic improvement and management scheme (e.g. reduction of congestion, effect of construction of new transport infrastructure on traffic movement prior to implementation).

#### **1.4 Organization of the Thesis**

This thesis comprises of seven chapters to illustrate the methodology for achieving the aforementioned objectives. The thesis is organized as below.

Chapter 1 gives the context of the study in a nutshell

Chapter 2 presents a review of available literature about various aspects of car following theories as well as of artificial neural network, fuzzy logic.

Chapter 3 describes the methodology employed in this study to achieve the stated objectives and the data collection techniques for this purpose.

Chapter 4 deals with different aspects of data analysis and development of models on a step by step basis.

Chapter 5 describes the validation of proposed models by performing statistical analysis for the proposed model.

Chapter 6 presents the conclusions of the entire study and suggests recommendations for future research.



## CHAPTER 2

### LITERATURE REVIEW

#### 2.1 Introduction

This study focuses on the application of Adaptive Neuro Fuzzy Inference System (ANFIS) to model car following behavior of heterogeneous road traffic condition in weak lane discipline. Therefore, it is important to have a clear understanding of not only the phenomenon and models of car following, but also the essential aspects of fuzzy set theory. The purpose of this chapter is to review previous studies pertaining to car following theory to gain knowledge about the vast amount of literature available about various aspects of this field. The chapter also includes a basic theoretical overview on fuzzy set theory including the prospect of using artificial neural network in it. This enables to understand the state of art of the subject and findings in previous studies. In this chapter, an extensive study of the literatures on relevant references both in Bangladesh and abroad has been briefed.

#### 2.2 Traffic Simulation Models

Traffic modeling is aimed to accurately recreate traffic as observed and measured on road where it is assumed that the appearance of a traffic system without replicating . Shanthikumar et. al. (1983) developed the model integrating the mathematic models into traffic system. Traffic modeling plays a significant role in traffic engineering, It can be utilized to plan and manage the traffic within certain road network

The models were grouped according to their areas of application. The three classifications of traffic simulation models, are microscopic modeling, macroscopic modeling and mesoscopic modeling. Macroscopic modeling represents traffic stream as an aggregate measured in terms of characteristics like speed, flow and density. Examples of macroscopic models include Greenshield (1935), Greenberg (1959), Underwood (1961). The advantage of such models is that they can describe the spatial and temporal extent of traffic congestion particularly that is caused by non-occurring incidents such as traffic crashes.

Microscopic modeling is based on the characteristics of various vehicle movements such as cars, buses, motorcycles and so on in the traffic flow. Microscopic modeling aimed to collect data parameters, such as, flow, density, speed, travel and delay time, long queues, stops, pollution, fuel consumption and shock waves. The characteristics of microscopic modeling methods were based on car-following model, lane-changing models and gaps of the individual drivers. The most important benefit of microscopic models is that they are used for evaluating traffic operational performance of the existing or future planned highways.

Mesoscopic modeling describes the analyzed transportation elements in small groups. The models coalesce with both characteristics of microscopic and macroscopic models. For example, average travel time and speed of a certain highway segment can be evaluated with this model using individual vehicles equipped with in-vehicle real-time travel information systems. The models are most favorable in evaluating traveler information systems. Kinetic theory based models are typical examples of mesoscopic models.

### **2.2.1 Car Following Models**

The concept of car following was first proposed by Reuschel (1950) and Pipes (1953). Car following models describe the interaction between adjacent vehicles in the same lane (Brackstone and McDonald, 1999). The subject vehicle follows the leader (vehicle in front) and responds to its action. These models are the major parts of the microscopic vehicular movements modeling which provide the foundation for traffic simulation systems.

Car-following models may further be divided into a number of sub-models like stimulus-response model, collision-avoidance model and psychophysical model etc.

#### **2.2.1.1 Stimulus-response models**

Stimulus –response models have been developed based on the assumption that the following vehicle driver can accurately perceive spacing and relative speed between the following and the lead vehicle without any scope for stochastic components. Herman and Rothery (1963) developed a mathematical formula of these models which is commonly known as fifth GM model. The formula is as follows:

$$a_{n+1}(t+T) = \frac{\alpha l m [v_{n+1}(t+T)]^m}{[x_n(t) - x_{n+1}(t)]^l} [v_n(t) - v_{n+1}(t)] \quad (2.1)$$

In the above equation,

- $a_i(j)$  = Acceleration of the  $i$ th vehicle at time  $j$
- $v_i(j)$  = Speed of the  $i$ th vehicle at time  $j$
- $x_i(j)$  = Position of the  $i$ th vehicle at time  $j$
- $\alpha$  = Sensitivity factor
- $l$  = Exponent of distance
- $m$  = Exponent of speed of following vehicle
- $T$  = Reaction time

It was Chandler et al. (1958) who put forth the first prototype leading to the above formula at the General Motors (GM) research labs in USA. At the same time in Japan, Kometani and Sasaki (1959), was working on their model. Gazis et al. (1959) made an endeavor to derive a macroscopic relationship describing speed and flow using the microscopic car following equation as a starting point. The mismatch between the two led to the hypothesis that the algorithm should be amended by the introduction of the inverse of distance term into the sensitivity constant, in order to minimize the discrepancy between the two approaches. Edie (1961) made an attempt to match the GM model to new macroscopic data in a similar manner finding that another amendment should be made to the sensitivity constant (based on the idea that the driver would be more sensitive to the relative speed as the speed of the traffic stream increases), namely, the introduction of the speed dependent term. This approach was used by Gazis et al. (1961) to investigate the sensitivity of their macroscopic relationships, to variations in the magnitude of the speed and distance terms. Researchers at the GM Research Laboratory introduced the sensitivity-stimulus framework that is the basis for most car following models to date (Toledo, 2003). According to this framework a driver reacts to stimuli from the environment. The response of the driver can be summarized by the following equation.

$$response(t) = sensitivity(t) \times stimulus(t - \tau) \quad (2.2)$$

Where,  $t$  is the time of observation and  $\tau$  is reaction time for drivers. The time interval between seeing, hearing, or feeling and the starting to do something in response to the stimulus of a traffic or highway situation are called “reaction time”. The psychological process constitutes four operations; perception, identification or

intellection, emotion or judgment and volition or reaction (means execution of decision).

### **2.2.1.2 Collision avoidance models**

To overcome the limitations of stimulus- response models, many advanced models have been introduced; collision avoidance models are one of them. The collision avoidance model is a car following behavior by modeling safe following distance based on the principle of Newtonian equations of motion. The essence of this model is that drivers select their speed in such a way that they can safely stop without collision. Research in this direction began in the late fifties by Kometani et al. and later continued by Gipps and others. Kometani and Sasaki (1959) proposed a car-following model in which the following vehicle driver seeks to maintain a safe following distance behind the lead vehicle. This safe distance would enable the following vehicle to avoid a collision with the lead vehicle if the driver of the vehicle in front were to act ‘unpredictably’.

The next major development of this model was made by Gipps (1981). He modeled the response of the following vehicle based on the assumption that each driver sets limits to his desired braking and acceleration rates. These limits enable the model to calculate a safe speed for the follower vehicle so that it can come to a safe stop if the vehicle ahead stops suddenly. The factors which he considered in the model are the ‘safety’ reaction time, maximum braking rate of lead vehicle and the maximum braking rate of the lead vehicle that the driver of the following vehicle believes is likely to be used.

The advantage of collision avoidance model is that longitudinal and lateral movement of vehicle both can be described by this model after a few adaptations. Gunay (2007) tried to integrate the lateral offset of the following vehicle into the Gipps (1981) following model. This study was a pioneer to describe the two-dimensional movement of cars. This model can allow flexibility to alter kinematic properties as well.

The greatest challenge to the validity of collision avoidance model emerged when it was appeared that Newtonian mechanics could fail to illustrate a short headway. This suspicion arouse when Brackstone et al. (2002) found that the minimum desired following distance was far lower than believed when they investigated the parameters for the action point model. In fact, it is obvious that a driver should be vigilant while

following a vehicle closely and the reaction time will be less than usual. An extremely short headway can be described by a collision avoidance model if the driver expects a low deceleration difference to the preceding vehicle (Lee, 2007).

### **2.2.1.3 Psychophysical models**

The concept of the psychophysical car-following models (also known as ‘Action Point model’) is similar that of the stimulus-response models. The concept was first brought up by Michaels (1963). The underlying assumption is, a leading vehicle influences longitudinal movement of a following vehicle in the same lane. A driver perception of relative movement of the leading vehicle, changes in separation and speed difference both influences the characteristics of the following vehicle. According to Wiedemann (1974), the drivers of the following vehicles follow the leaders even when the space headway is large and drivers perceive changes in separation only if the physical impulse exceeds a certain minimum value called threshold. A driver expects to drive to his target or desired speed in free flow condition. The response of the driver is influenced on a large scale by the term “perception threshold” which works in two mutually related and subsequent operations. First of all, a driver will escalate the speed until he realizes that the further increment will be intrepid enough to cause a rear end collision. Then he will try to maintain the speed with the leading vehicle. It is impossible to maintain the equal space or time headway all the time. In the second case, when the preceding vehicle will be far beyond the perception threshold or space headway will increase significantly, the same driver of the following vehicle will try to accelerate his vehicle and reduce the headway and maintain equal speed to that of the leader. The mechanisms are periodic and termed as “following spiral”. As following and leading vehicles are totally relative to each other, there is no way to stop the cycle. As a result, the psychophysical model is able to illustrate the oscillating phenomenon observed in car following experiments.

Todosiev (1963), a noteworthy researcher, whose contribution in developing psychophysical models is undeniable. Todosiev (1963) used the optic flow (the rate of change of the subtended angle of the lead vehicle) to determine the dynamic relationship with the lead vehicle. The study found that the relative acceleration was almost constant in certain portions and driver changed them at certain action points called thresholds. Todosiev conducted simulator experiments to determine the thresholds and human perception thresholds causing them. The results obtained were

compared from similar results obtained from psychophysical investigations in human perception of moving objects. Action point density (as a function of relative speed and relative acceleration) was used in the study for different speed levels to determine speed thresholds. Braunstein and Laughery (1964) investigated drivers' ability to detect relative speed and found that detection time increased with following distance and decreased with an increase in relative speed. Hoffmann (1966) found that drivers were able to detect a change in inter vehicle spacing as little as 12% of the initial spacing. Lee (1976) assumed in his study that drivers judge their decision to start braking on the basis of 'Time to Collision' (TIC) and distance with the vehicle ahead. The psychophysical approach and derivatives of it are now in a number of traffic simulation programs such as VISSIM, PARAMICS, etc.

#### **2.2.1.4 Fuzzy logic based models**

The models so far discussed are all deterministic models. Fuzzy logic has been accepted as an emerging technique which represents the next logical step in attempting accurately describes driver behavior. The use of fuzzy logic within car following models can explain the vagueness of perception of various inputs associated with car following from the point of view of motorists. The basic concept of fuzzy logic-based model is that it alters input factors into linguistic forms using particular membership functions. The models use normal language-based 'IF-THEN' type driving rules. Neural network approach is similar to fuzzy logic-based but it integrate past driving behavior through the process of learning and training.

Kikuchi and Chakroborty (1992) were first to produce neuro-fuzzy model. They attempted to 'fuzzify' the traditional GM model using 'Relative Speed', 'Distance' and 'Acceleration of Leading Vehicle' as inputs using 396 'IF-THEN' type rules. The outcome of the rule base was that the following vehicle will accelerate at the same rate as the leader, plus a small term to account for relative speed and distance. The model was then used to illustrate how the fuzzy logic system can be used to describe car following, with local stability being examined for various configurations of inputs, which were compared with traditional GM models.

Inokuchi et al. (1999) presented similar approach by integrating neural networks and fuzzy logic. This neuro-fuzzy model utilized similar linguistic definitions of IF-THEN fuzzy logic rules. The study also incorporated a neural network to enhance the control algorithm. The neural network model is representation of the process that learns by

experience and examples. In the proposed neuro-fuzzy car-following model, the characteristic of the following driver was observed from a data and the learning process was represented by changing the weight of the synapse or connection between neurons in the model. The neural network was incorporated with the fuzzy rules and produced an estimation of the actions of the driver. However, driver response time lag was considered as a constant in this model. Several other research efforts also considered fuzzy application of car-following theory including fuzzification of the MISSION model (Rekersbrink, 1995) and information of the MITRAM model (Yikai et.al., 1993).

### **2.2.1.5 Microscopic simulation models**

Several articles have focused on the comparison of urban road traffic simulators. Ghariani et al. (2014) have made a comparison of seven simulators according to their ability to simulate public transport according to certain criteria such as visualization, software category (commercial or open source), and the public transport infrastructure, the characteristics of public transport vehicles as well as passengers and pedestrians. The authors concluded that the simulators, AIMSUN, VISSIM and Paramics, offer the ability to import aerial images that can be used as background and as a guide in the development of road networks. 3D views offered in VISSIM and AIMSUN are better than Paramics. In Paramics, vehicles appear as 3D blocks without distinct characteristics. However, in VISSIM and AIMSUN, vehicles appear with distinct characteristics. In some simulation software, special attention was paid to the driver settings. For example, VISSIM offers different car-following or lane-changing models. Although VISSIM, AIMSUN and Paramic allow the simulation of bus circuit, only VISSIM simulates disembarkation and embarkation of passengers from the left side of roads. Therefore, only VISSIM allows bus stops on the left side or the right side of the road. The occupancy rate of the bus and the vehicle's capacity can also be defined in VISSIM and Paramics, while the number of doors in buses is only recognized in VISSIM.

Sun et al. (2013) studied the two simulators VISSIM and CORSIM based on an urban network in the North Bund area, Hong kou District, Shanghai, China. They concluded that the use of the software, publishing and network configuration in CORSIM is easier. Due to the existence of different simulation mechanisms, CORISM simulation results can be transmitted quickly, while VISSIM configuration provides an easy interface with a separate output file to the simulation results. While for the calibration

effort, the software provides multiple calibration parameters to allow the simulated network to reproduce the real situation. VISSIM is more appropriate for large intersections with broadband traffic, while CORSIM is good for modeling unsaturated intersections. VISSIM's simulation results are closer to the real situation.

Kokkinogenis et al. (2011) presented a comparison of seven simulators based on criteria such as visualization, expansion of the simulator and it is oriented agent. The classification of simulators based on their ability to simulate cars, buses, trucks, trains, bicycles and pedestrians, as the management of the sensors (such as traffic lights, electromagnetic loops, cameras, etc). The article found that MITSIM and SUMO do not have 3D visualization. VISSIM, Paramics and AIMSUN have a realistic 3D visualization which is very useful for presenting real-world scenarios. MITSIM does not provide expansion capabilities, but since the simulator is open-source researchers can modify and extend its core. Each simulator has advanced features, for example, VISSIM has parameters allowing the flexibility of its functions, Paramics is suitable for use of resources on distributed machines, AIMSUN provides various options for its extension, SUMO has a flexible architecture, and MAS -T2er Lab. and ITSUMO are oriented agent.

The aim of the research conducted in the article by Maciejewski (2010) was to provide a comparison of three selected platforms (SUMO, VISSIM and TRANSIMS) on a fragment of areal urban network (in Poland). In general, it seems that the model SUMO had too low capacity compared to the capacity of the actual network, but on the other hand, it is difficult to assess the capabilities of VISSIM and TRANSIMS were consistent with, or better than the actual network. However, despite some differences in quantitative measurements, effects similar to the spread of the traffic flow (appearance of network bottlenecks and traffic jam effects) were observed in all three models.

Ratrouf and Rahman (2009) offered a comparative analysis of fourteen simulators focusing on the simulation of traffic at highways, congested urban networks and adaptation of simulators to ITS (Intelligent Transport Systems) applications. This article revealed that AIMSUN, CORSIM and VISSIM are suitable for clogged arteries and highways, but AIMSUN is less suited compared to others. The AIMSUN characteristics are favorable to the creation of large urban and regional networks, but it requires difficult coding. The models AIMSUN, Paramics, INTEGRATION and CORSIM seem most suitable to ITS applications.



### 2.2.1.6 Mixed traffic models

Oketch (2000) made an attempt to formulate a different modeling approach for modeling heterogeneous traffic including non-motorized traffic. A car following model was adopted in this study similar to that proposed in Gipps (1981) model. Both longitudinal and lateral movements were considered in this study (two-dimensional movement). The lateral movement was taken as gradual process instead of an instantaneous one. This study acknowledged the type of the lead and following vehicles while maintaining a longitudinal gap (in motion and standstill state) and while performing overtaking. Cho and Wu (2004) made an attempt to develop a longitudinal movement model for motorcycles. This model was derived based on the thrust (desired speed of the follower), space headway and safety margin. Moreover, in this study the staggered following is described based on one weight function by considering the effects of lateral position difference on the longitudinal headway. Lan and Chang (2004) developed a model which deals with more than one leading vehicles. They made an attempt to use General Motors model to explain the motorcycle's following behaviors in two cases: (1) only one leading vehicle in front; (2) two or more leading vehicles in front and neighboring-front (including left-front, right-front, or both). Arasan and Koshy (2005) described a modeling methodology adopted to simulate the flow of heterogeneous traffic with vehicles of wide ranging static and dynamic characteristics. Lee et al. (2009) developed a model to study the longitudinal behavior of vehicles in mixed traffic conditions. This model considers behavior patterns, such as squeezing through small lateral gaps, moving abreast of other vehicles in the same lane, oblique following and swerving. To capture these behaviors, longitudinal headway model and oblique following model were developed using collision avoidance concept. Maurya (2011) developed comprehensive driver behavior model which considers concurrently both longitudinal and lateral interaction with roadway and traffic features. Mallikarjuna et al. (2011) analyzed microscopic data under heterogeneous traffic conditions such as vehicle composition in the traffic stream, lateral distribution of vehicles, lateral gaps and longitudinal gaps using image processing. Empirical relationships between the lateral gap and area occupancy have been proposed for various vehicle combinations. The influence of difference in the lateral positions of leading and following vehicles on the longitudinal gap has been analyzed. Asaithambiet al. (2018) implemented a microscopic traffic simulation models for mid blocksection. The performance of different vehicle-following models was evaluated based on different Measure of Effectiveness (MoE) using field data collected from a four-lane divided urban arterial road in Chennai city. Speed-concentration and flow-concentration relationships for different vehicle-following

models were developed and analyzed for different compositions. Capacity is higher when the proportion of smaller size vehicles is higher, since these vehicles use longitudinal and lateral gaps effectively. The simulation model was also applied to evaluate a range of traffic control measures based on vehicle type and lane (Ex: exclusion of auto-rickshaws, heavy vehicles, auto-rickshaws, heavy vehicles, etc.). An extensive study was conducted by Ravishankar and Matthew (2011) on the car-following behavior of vehicles in heterogeneous traffic stream consisting of diverse vehicle types with very weak lane discipline in India. The focus of their work was incorporation of some vehicle type dependent parameters to modify Gipps' (1981) Model in order to better represent the following behavior of vehicles. Performance of the model was investigated for both microscopic and macroscopic level using data collected from both homogeneous (for car and truck combination) and heterogeneous traffic stream.

In Bangladesh, Hoque (1994) was first to model a signalized intersection of Dhaka city. This model (MIXSIM) has been used in computation of PCU conversion factor and evaluation of alternative schemes to improve the capacity of an approach serving mixed traffic flow. Later, Hossain (1996) modeled heterogeneous traffic operation at two corridors. In this study, the effect of different composition of non-motorized traffic on corridor travel time and passenger movement capacity has been analyzed based on the developed simulation model (MIXNETSIM) (Hossain, 1996). Moreover, some factors have been identified (e.g. traffic density) which may result in higher delay for traffic for larger size vehicles. Imran (2009), on his research, has focused on developing car-following models for mixed traffic using fuzzy-logic inference system of MATLAB. In recent years, Islam (2013) has developed an acceleration model traffic streams having weak lane discipline using comprehensive disaggregate (e.g. second by second trajectory data) and aggregate data (e.g. traffic counts and speeds) and calibrated it using econometric approaches.

### **2.3 Adaptive Neuro –Fuzzy Interface System**

The objective of an ANFIS is to integrate the best features of Fuzzy Systems and Neural Networks. ANFIS is one of the best tradeoffs between neural and fuzzy systems, providing smoothness, due to the Fuzzy Control (FC) interpolation and adaptability due to the Neural Network Back propagation. Before describing ANFIS it is needed to explain Fuzzy Systems first.

### 2.3.1 Fuzzy Logic

Two distinct forms of problem knowledge exist for many problems: Objective knowledge, which is used in all engineering problem formulations (e.g. mathematical models), and Subjective knowledge, which represents linguistic information that is usually impossible to quantify using traditional mathematics (e.g. rules, expert information, design requirements).

Earlier system modeling problems were solved using conventional mathematical tools (like differential equations), but they are not well suited with vaguely defined and uncertain system. Takagi and Sugeno (1985) introduced fuzzy inference system which employs fuzzy if-then rules that can model the qualitative aspects of human knowledge and reasoning processes without employing quantitative analyses. So Fuzzy modeling, has found numerous practical applications in control, prediction and inference. Fuzzy logic is a form of multi-valued logic derived from fuzzy set theory to deal with reasoning that is approximate rather than precise. In contrast to binary sets having binary crisp logic, the fuzzy logic variables may have a membership value of not only 0 or 1. Just as in fuzzy set theory with fuzzy logic the set membership values can range (inclusively) between 0 and 1, in fuzzy logic the degree of truth of a statement can range between 0 and 1 and is not constrained to the two truth values {true (1), false(0)} as in classic propositional logic.

#### 2.3.1.1 Fuzzy inference system (FIS)

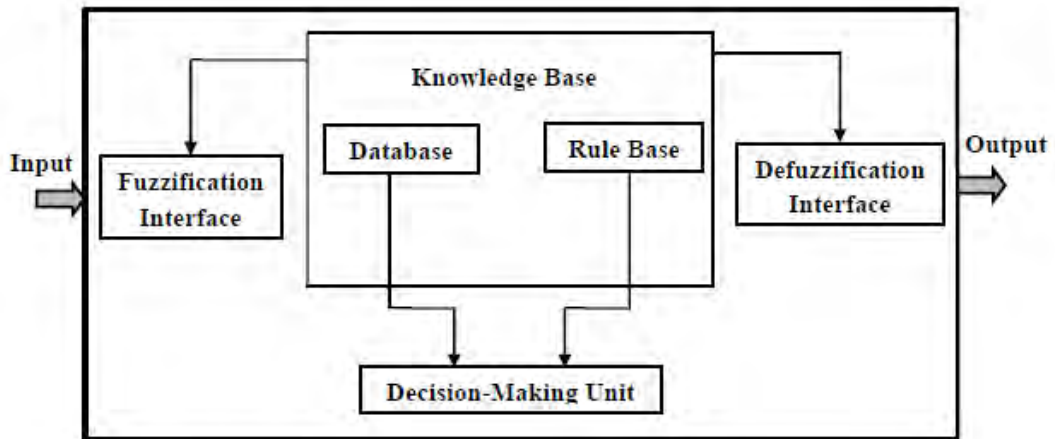
Fuzzy inference systems are also known as fuzzy-rule-based systems, fuzzy models, fuzzy associative memories (FAM), or fuzzy controllers when used as controllers. Fuzzy inference is the process which maps the given input into the output using fuzzy logic. In general, a Fuzzy Logic System (FLS) is a nonlinear mapping of an input data (feature) vector into a scalar output data. The richness of the FL is that there are enormous numbers of possibilities that leads to lots of different mappings. This richness does require a careful understanding of FL and the elements that comprise a FLS.

Basically a fuzzy inference system is composed of five functional blocks as shown in Figure 2-1. These are,

- I. a rule base containing a number of fuzzy if-then rules
- II. a database which defines the membership functions of

- III. a decision-making unit which performs the inference
- IV. a fuzzification interface which transforms the crisp inputs
- V. a defuzzification interface which transform the fuzzy.

Usually, the rule base and the database are jointly referred to as the knowledge base.



**Figure 2-1:** Fuzzy Inference System

Rules may be provided by experts or can be extracted from numerical data. Fuzzy if-then rules or fuzzy conditional statements are expressions of the form IF A THEN B, where A and B are labels of fuzzy sets characterized by appropriate membership functions. Due to their concise form, fuzzy if-then rules are often employed to capture the imprecise modes of reasoning that play an essential role in the human ability to make decisions in an environment of uncertainty and imprecision. This can be described by the examples.

If the speed is high, then apply the brake a little, where speed and brake are linguistic variables, high and little are linguistic values or labels that are characterized by membership functions.

### 2.3.1.2 Fuzzy set theory

#### 2.3.1.2.1 Crisp sets

The crisp set is a collection of objects (say U) having identical properties such as count ability and finiteness. A crisp set 'B' can be defined as a group of elements over the universal set U, where a random element can be a part of B or not. Which means there are only two possible ways, first is the element could belong to set B or it does

not belong to set B. The notation to define the crisp set B containing a group of some elements in U having the same property P is given below.

$$B = \{x: x \in U \text{ and } x \text{ has same property } P\} \quad (2.3)$$

It can perform operations like union, intersection, compliment and difference. The traditional approach (crisp logic) of knowledge representation does not provide an appropriate way to interpret the imprecise and non-categorical data. In another way, it cannot deal with the representation of human intelligence.

### 2.3.1.2.2 Fuzzy sets

A fuzzy set is a combination of the elements having a changing degree of membership in the set. Here “fuzzy” means vagueness, in other words, the transition among various degrees of the membership implies that the limits of the fuzzy sets are vague and ambiguous. Therefore, the membership of the elements from the universe in the set is measured against a function to identify the uncertainty and ambiguity.

A fuzzy set is denoted by a text having tilde under strike. Now, a fuzzy set X would contain the entire possible outcome from interval 0 to 1. Suppose a is an element in the universe is a member of fuzzy set X, the function gives the mapping by  $X(a) = [0,1]$ . The notion convention used for fuzzy sets when the universe of discourse U (set of input values for the fuzzy set X) is discrete and finite, for fuzzy set X is given by:

$$\mu_X(a) \in [0,1]$$

$$X = \left\{ \frac{\mu_A(a_1)}{a_1} + \frac{\mu_B(a_2)}{a_2} + \dots \right\} = \sum_i \frac{\mu_X(a_i)}{a_i} \quad (2.4)$$

The fuzzy set theory was initially proposed by a computer scientist Lotfi A. Zadeh in the year of 1965. After that lot of theoretical development has been done in a similar field. Previously the theory of crisp sets based in dual logic is used in the computing and formal reasoning which involves the solutions in either of two form such as “yes or no” and “true or false”.

### 2.3.1.3 Linguistic variables

Linguistic variables are variable whose values are not numbers but words or sentences in a natural or artificial language. In general, linguistic variables are less specific than numerical ones. Let  $u$  denote the names of linguistic variable, numerical values of a linguistic variable  $u$  are denoted  $x$ , where  $x \in U$ . Sometimes  $x$  and  $u$  are interchangeably used. A linguistic variable is usually decomposed into a set of terms,  $T(u)$ , which covers its universe of discourse.

### 2.3.1.4 Membership functions

Membership functions,  $\mu_F(x)$  for the most part, associated with terms that appear in the antecedents or consequents of rules, or in phrases. The most commonly used shapes for membership functions are triangular, trapezoidal, piecewise, linear and Gaussian. Usually, membership functions are chosen by the user arbitrarily, based on the user's experience; hence, the membership function for two users could be quite different depending upon their experiences, perspectives, cultures, etc. Fuzzy logic was introduced as a superset of standard Boolean logic by considering the fuzzy values that ranges from 0 to 1 instead of only considering two values true or false and applying the same logic operators such as AND, OR, NOT, etc. Thus the concept is extended from two valued logic to multi-valued logic.

The conditional statement commonly known as IF-THEN rules can be easily formulated using fuzzy logic. Rules consist of two parts: the antecedent or the IF part, and the consequent or the THEN part. The IFTHEN rules can take the following form:

$$\text{IF } x \text{ is } A \text{ and } y \text{ is } B \text{ THEN } z \text{ is } C \quad (2.5)$$

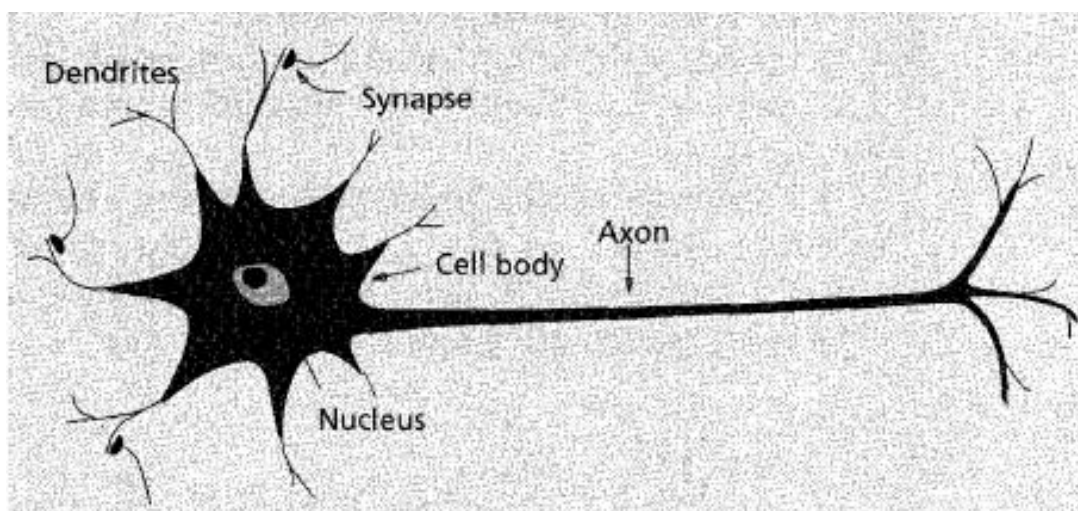
Where,  $A$ ,  $B$  and  $C$  are linguistic variables whose values are sentences in a natural language.

The main disadvantage of fuzzy classifier is that system time response slows down with the increase in number of rules. If the system does not perform satisfactorily, then the rules are reset again to obtain efficient results i.e. it is not adaptable according to the variation in data. The accuracy of the system is dependent on the knowledge and experience of human experts. The rules should be updated and weighting factors in the fuzzy sets should be refined with time. Neural networks,

genetic algorithms, swarm optimization techniques, etc. can be used to for fine tuning of fuzzy logic control systems.

### 2.3.2 Neural Network

A neural network (NN), often just called a “Artificial Neural Network” (ANN), is a mathematical model or computational model based on biological neural networks. It consists of an interconnected group of artificial neurons and processes information using a connectionist approach to computation. Neural networks can be defined as “A system composed of many simple processing elements operating in parallel whose function is determined by network structure, connection strengths, and the processing performed at computing element or nodes”.



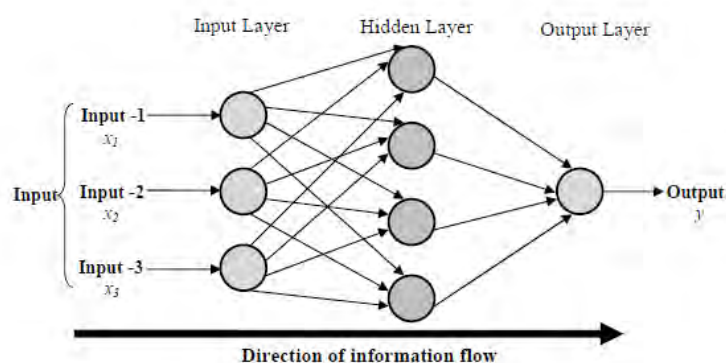
**Figure 2-2:** Schematic diagram of a Biological Neuron

Neural networks have the ability to adapt to changing input so that the network produces the best possible result without the need to redesign the output criteria. The original inspiration for the technique was from examination of the central nervous system and the neurons which constitute one of its most significant information processing elements (Figure 2-2). The neural networks resemble the brain mainly in two respects-knowledge is acquired by the network from its environment and interneuron connection strengths, known as synaptic weights are used to store the acquired knowledge.

Neural networks resemble the human brain in the following two ways:

1. A neural network acquires knowledge through learning.
2. A neural network's knowledge is stored within inter-neuron connection strengths known as synaptic weights

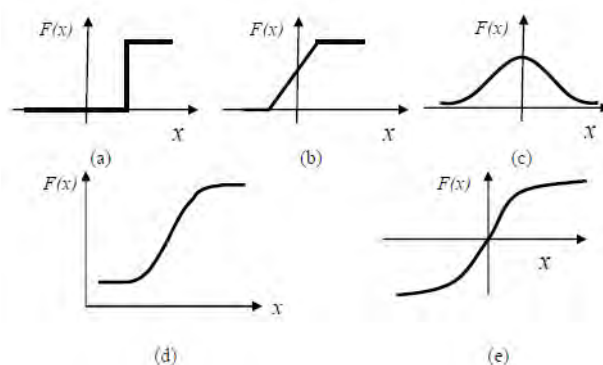
The true power and advantage of neural networks lies in their ability to represent both linear and non-linear relationships and in their ability to learn these relationships directly from the data being modeled. Traditional linear models are simply inadequate when it comes to modeling data that contains non-linear characteristics.



**Figure 2-3:** A Feed Forward ANN with Input, Output and One Hidden Layer

The most common neural network model is the multi-layer perceptron (MLP). This type of neural network is known as a supervised network because it requires a desired output in order to learn. The goal of this type of network is to create a model that correctly maps the input to the output using historical data so that the model can then be used to produce the output when the desired output is unknown. A graphical representation of an MLP is shown in Figure 2-3.

In a two hidden layer MLP, the inputs are fed into the input layer and get multiplied by interconnection weights as they are passed from the input layer to the first hidden layer. Within the first hidden layer, they get summed up and then processed by a nonlinear function also known as activation function. Figure 2-4 shows the activation functions used in NN.



**Figure 2-4:** Different types of Activation Functions: (a) Threshold (b) Piecewise Linear (c) Gaussian (d) Sigmoid (e) Tan-hyperbolic



As the processed data leaves the first hidden layer, again it gets multiplied by interconnection weights, then summed and processed by the second hidden layer. Finally the data is multiplied by interconnection weights then processed one last time within the output layer to produce the neural network output.

The MLP and many other neural networks learn using a learning algorithm. The backpropagation method uses a steepest descent technique, in order to update the parameters associated with the shape and position of both input and output membership functions for model calibration. The hybrid method uses a combination of backpropagation (steepest descent) method and least square method to update the membership function parameters for model calibration. With each presentation the output of the neural network is compared to the desired output and an error is computed. This error is then fed back to the neural network and used to adjust the weights such that the error decreases with each iteration and the neural model gets closer and closer to producing the desired output. This process is known as "training".

### **2.3.3 ANFIS Architecture**

ANFIS is a hybrid system incorporating the learning abilities of NN and excellent knowledge representation and inference capabilities of fuzzy logic (Jang 1993) that have the ability to self-modify their membership function to achieve a desired performance. An adaptive network, which subsumes almost all kinds of neural network paradigms, can be adopted to interpret the fuzzy inference system. ANFIS utilizes the hybrid-learning rule and manage complex decision-making or diagnosis systems. ANFIS has been proven to be an effective tool for tuning the membership functions of fuzzy inference systems.

ANFIS is a simple data learning technique that uses a fuzzy inference system model to transform a given input into a target output. This prediction involves membership functions, fuzzy logic operators and if-then rules. There are two types of fuzzy system, commonly known as the Mamdani and Sugeno models. There are five main processing stages in the ANFIS operation, including input fuzzification, application of fuzzy operators, application method, output aggregation, and defuzzification.

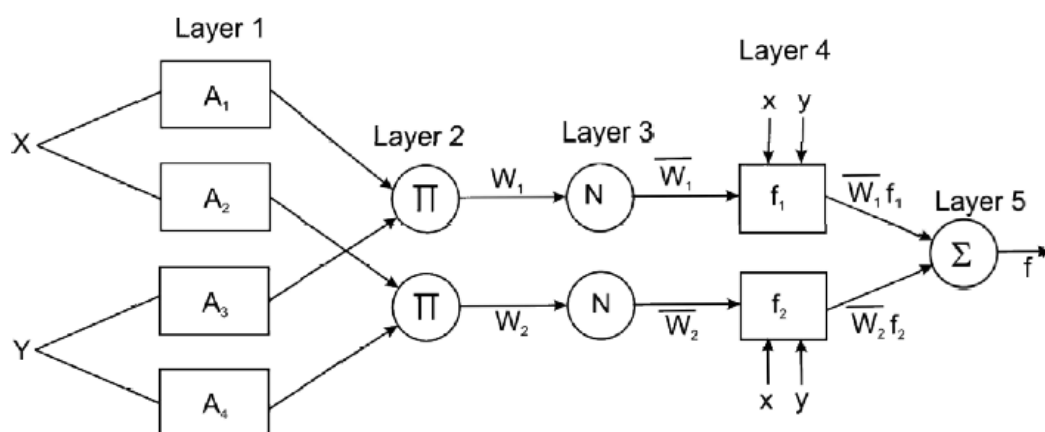
ANFIS utilizes “Representation of prior knowledge into a set of constraints (network topology) to reduce the optimization search space”, from Fuzzy Systems and “adaptation of back propagation to structured network to automate FC parametric

tuning”, from Neural Networks, to improve performance. The design objective of the fuzzy controller is to learn and achieve good performance in the presence of disturbances and uncertainties. The design of membership functions is done by the ANFIS batch learning technique, which amounts to tune a FIS with back propagation algorithm based on a collection of input–output data pairs.

Generally, ANFIS is a multilayer feed forward network in which each node performs a particular function (node function) on incoming signals. For simplicity, we consider two inputs 'x' and 'y' and one output 'z '. Suppose that the rule base contains two fuzzy if-then rules of Takagi and Sugeno type (Jang 1993):

Rule 1: IF x is A1 and y is B1 THEN  $f_1=P_1x+Q_1y+R_1$

Rule 2: IF x is A2 and y is B2 THEN  $f_2=P_2x+Q_2y+R_2$  (2.6)



**Figure 2-5: ANFIS Architecture**

The ANFIS architecture is a five layer feed forward network as shown in Figure 2-5. An adaptive network (Jang 1993) is a multilayer feed forward network in which each node performs a particular function (node function) on incoming signals as well as a set of parameters pertaining to this node. The formulas for the node functions may vary from node to node, and the choice of each node function depends on the overall input-output function which the adaptive network is required to carry out. Note that the links in an adaptive network only indicate the flow direction of signals between nodes; no weights are associated with the links.

To reflect different adaptive capabilities, we use both circle and square nodes in an adaptive network. A square node (adaptive node) has parameters while a circle node (fixed node) has none. The parameter set of an adaptive network is the union of the

parameter sets of each adaptive node. In order to achieve a desired input-output mapping, these parameters are updated according to given training data and a gradient-based learning procedure is used.

Layer 1: Each node in this layer generates membership grades of a linguistic label. For instance, the node function of the  $i^{\text{th}}$  node may be a generalized bell membership function:

$$O_{1,i} = \mu_{A_i}(x); \text{ for } i = 1,2 \quad (2.7)$$

$$O_{1,i} = \mu_{B_{i-2}}(y); \text{ for } i = 3,4 \quad (2.8)$$

So, the  $O_{1,i}(x)$  is essentially the membership grade for  $x$  and  $y$ .

$$\mu_A(x) = \frac{1}{1 + \left| \frac{x-c_i}{a_i} \right|^{2b_i}} \quad (2.9)$$

Where,  $x$  is the input to node  $i$ ;  $A_i$  is the linguistic label (small, large, etc.) associated with this node; and  $\{a_i, b_i, c_i\}$  is the parameter set that changes the shapes of the membership function. Parameters in this layer are referred to as the premise parameters.

Layer 2: Each node in this layer calculates the firing strength of a rule via multiplication and the nodes are fixed:

$$O_{2,i} = w_i = \mu_{A_i}(x)\mu_{B_i}(y); i = 1,2 \quad (2.10)$$

Layer 3: The nodes are fixed nodes. They are labeled with N, indicating that they play a normalization role to the firing strengths from the previous layer. The outputs of this layer can be represented as:

$$O_{3,i} = \overline{w_i} = \frac{w_i}{w_1 + w_2} \quad (2.11)$$

Layer 4: The nodes are adaptive nodes. The output of each node in this layer is simply the product of the normalized firing strength and a first-order polynomial (for a first-order Sugeno model). Thus, the outputs of this layer are given by:

$$O_{4,i} = \bar{w}_i f_i = \bar{w}_i (p_i x + q_i y + r_i) \quad (2.12)$$

Where  $O_{4,i}$  is the output of layer 4, and  $\{p_i, q_i, r_i\}$  is the parameter set. Parameters in this layer will be referred to as the consequent parameters.

Layer 5: There is only one single fixed node labeled with  $\Sigma$ . This node performs the summation of all incoming signals. Hence, the overall output of the model is given by:

$$O_{5,i} = \sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \quad (2.13)$$

The shown adaptive network in Figure 2-5 is functionally equivalent to a fuzzy inference system in Figure 2-1. The basic learning rule of ANFIS is the back propagation gradient descent (Aik and Jayakumar, 2008), which calculates error signals (the derivative of the squared error with respect to each node's output) recursively from the output layer backward to the input nodes. This learning rule is exactly the same as the back propagation learning rule used in the common feed forward neural networks. From the ANFIS architecture in Figures 2-1 and 2-5, it is observed that given the values of premise parameters, the overall output  $f$  can be expressed as linear combinations of the consequent parameters:

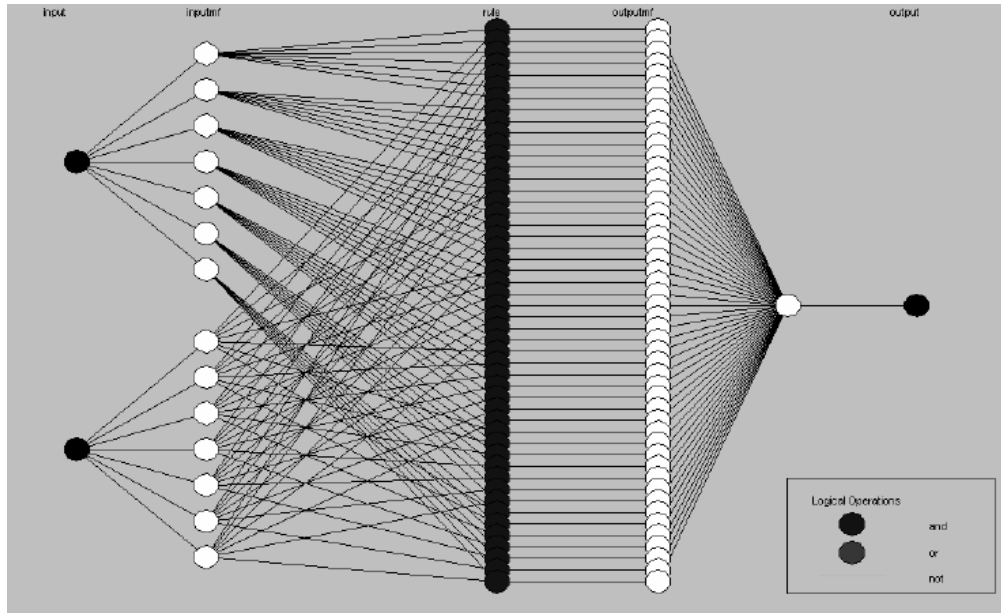
$$f = \bar{w}_1 f_1 + \bar{w}_2 f_2 \quad (2.14)$$

$$f = \bar{w}_1 (p_1 x + q_1 y + r_1) + \bar{w}_2 (p_2 x + q_2 y + r_2) \quad (2.15)$$

Here, it is to be observed that, the equations were developed based on two inputs and two categories of that inputs. If more inputs were defined with more or less categories, then the equation would vary accordingly.

### 2.3.3.1 ANFIS Learning Algorithm

In this subsection, the hybrid learning algorithm is explained briefly. The ANFIS Learning Algorithm uses a two-pass learning cycle. In the forward pass, S1 is unmodified and S2 is computed using a Least Squared Error (LSE) algorithm (Off-line Learning). In the Backward pass, S2 is unmodified and S1 is computed using a gradient descent algorithm (usually Back Propagation).



**Figure 2-6: ANFIS structure**

From the ANFIS structure shown in Figure 2-6, it has been observed that when the values of the premise parameters are fixed, the overall output can be expressed as a linear combination of the consequent parameters.

The hybrid learning algorithm is a combination of both back propagation and the least square algorithms. Each epoch of the hybrid learning algorithm consists of two passes, namely forward pass and backward pass. In the forward pass of the hybrid learning algorithm, functional signals go forward up to layer 4 and the consequent parameters are identified by the least squares estimate. The back propagation is used to identify the nonlinear parameters (premise parameters) and the least square is used for the linear parameters in the consequent parts.

## 2.4 Limitation of State-of-the-art models

In the following sections the limitations of state-of-the-art acceleration models for both homogeneous traffic and heterogeneous traffic are described.

- a) While developing the car following models it has been assumed that the drivers in a traffic stream obey traffic rules. But the actual scenario is quite different. In fact, these rules may differ with different drivers, or even for the same driver and with different conditions, and in some cases, possibly with the same driver and nearly identical situations.

- b) Most of the models only a set of kinematic variables, such as relative spacing and speeds, instantaneous speeds,, etc., to determine subsequent driving behavior (Kim et al., 2003). In fact, there are numerous other factors besides basic kinematics that may influence car-following behavior, such as various human characteristics. So it can be said that a number of significant stochastic components are missing from some state-of-the-art acceleration models (Kim et al., 2003).
- c) Most models described above are deterministic model.
- d) Most models are made for homogeneous road traffic condition.
- e) The continuous lateral movement of the vehicle is a common feature in a traffic stream where lane discipline is weak. This type of movement has effect on the car following behavior of the vehicle. But this phenomenon has not been discussed in the previous works.

## **2.5 Summary**

In this chapter, a comprehensive review of literature was carried out not only on car following theory but also on fuzzy logic. From the review, it is apparent that although substantial research has been carried out on the topic abroad, there has hardly been any research in Bangladesh so far on the topic. Few limitations of the state of the art models has been discussed at the end. Therefore, the development of car following models using Neuro-Fuzzy techniques under heterogeneous road traffic condition for weak lane discipline of Bangladesh is justified.

## CHAPTER 3

### METHODOLOGY AND DATA COLLECTION

#### **3.1 Introduction**

The roads in Bangladesh are typically non-lane based and heterogeneous road traffic condition in nature. To address the uncertainty of traffic condition, it is required to develop models to depict the acceleration behavior of vehicles following one another. Because of the ambiguity and vagueness of relevant variables on the part of the motorist following a leader, the use of Neuro-Fuzzy (ANFIS) techniques has a enormous prospect for application in the context of Bangladesh and other developing countries. The chapter focuses on development of Neuro-Fuzzy models of car following behavior and also on subsequent collection of data to calibrate the models.

#### **3.2 Neuro-Fuzzy Models**

This section describes the benefits of using Neuro-Fuzzy (ANFIS) for the modeling of car following behavior over deterministic models used for the same purpose.

##### **3.2.1 Advantages of Neuro-Fuzzy Models**

During the past decades, Fuzzy logic has been accepted as an emerging technique for its wide range of successful applications, such as in medical diagnosis system (Moein at el., 2009), industrial processes (Kaur & Kaur, 2012) etc. the application of artificial neural network to calibrate various parameters of the Fuzzy Inference System or FIS (e.g. shape and position of membership functions) makes this technique even more versatile and adaptive. Therefore, the use of Neuro-Fuzzy (ANFIS) has a huge potential for the modeling of vehicular movement.

There are innumerable benefits of using Adaptive Neuro-Fuzzy Inference System (ANFIS) for modeling car following behavior for heterogeneous road traffic condition, some of which are enlisted as follows:

- a) The use of fuzzy logic can explain the vagueness of perception of various inputs associated with car following from the point of view of motorists, thereby ensuring the ability to depict car following behavior, which is approximate in nature, in a realistic manner.
- b) The use of artificial neural network calibrate tune various parameters of the Fuzzy Inference System (FIS), thereby establishing a pattern between input variables and the output with a better fit to replicate field data of car following.
- c) For heterogeneous traffic, the presence of two or more models of car following representing various vehicle types can be easily managed using Neuro-Fuzzy techniques using the same architecture of the basic Fuzzy Inference System, while the only difference being the variation of field data for training.
- d) Different characteristics other than vehicular characteristic such as human characteristics, road characteristic etc. can be easily added to the model using Fuzzy Logic, thereby ensuring the ability to depict realistic car following behavior.

The aforementioned advantages cannot be benefited from a deterministic model. Therefore, it is justified to use Adaptive Neuro-Fuzzy Inference System (ANFIS) to develop models to represent car following behavior under heterogeneous road traffic condition in weak lane discipline, which is a typical feature of the roads of Bangladesh. It is expected that the models will be able to depict car following behavior in Bangladesh condition successfully and realistically.

It is important to mention here that the “Neuro-Fuzzy Designer” toolbox of MATLAB provides an excellent interface to develop Neuro-Fuzzy models. The toolbox is entirely in the form of graphical user interface, where every aspect of model development (e.g. membership functions, fuzzy rules etc) is handled graphically, without the need to write even a single line of code at the command prompt of MATLAB. Considering these points, the “Neuro-Fuzzy Designer” toolbox of MATLAB would be used in this study.

### **3.3 Data Collection Techniques**

To analysis car following behavior, it is of utmost importance to collect data which involves extraction of various parameters such as speed, acceleration, headways etc. on a continuous basis. From practical point of view, this is one of the biggest



challenges as the process of collecting data should not be a disruption for the normal vehicle movement on road. The process of vehicle data extraction goes back as far as fifty to sixty years from now. Over this period of time, a number of different data collection techniques have been devised to observe and record car following behavior. In this section, various procedures for the extraction of car following data will be elaborated. After discussing various aspects of these methods, the best one suited in the context of Bangladesh will be selected.

### **3.3.1 Wire linked Vehicles Technique**

Wire linked vehicle is one of the oldest methods used to extract vehicle data. The earliest works on car following theory (Chandler et al., 1958; Gazis et al., 1959; Gazis et al., 1961; Herman and Rothcry, 1963) used wire linked vehicles for the measurement of car following parameters. In this technique, the subject vehicle was fitted with a wire fastened to the rear bumper of a control vehicle, with a constant wire tension maintained by a slipping friction clutch. The downside of this method was inevitably that it was limited to test tracks and the accuracy of the data was in question.

### **3.3.2 Photographic and Video Techniques**

Photographic and video techniques are one of the most popular methods of car following data collection. The first incidence of this technique goes back late 1950s. Kometani and Sasaki (1959) photographed a pair of vehicles by a camera from the top of a roadside building to extract various parameters. In mid 1960s, Snider and Ernst (1965) mounted a camera on the rear end of the leader to record a following vehicle and gathered the parameters to calculate the number of pixels representing the width of the subject vehicle. More accurate results were found using the same technique with better frame rate of the video by Constantine and Young (1967). Treiterer and Myers (1974) went to great lengths and used aerial photography to take pictures at certain intervals, whereas Ozaki (1993) obtained car following data by making use of video films taken from a high rise building. Imran (2009) used video footage of vehicular movements at one of the busiest roads of Dhaka city to collect car following data. The advantage of these approaches is the ability to observe vehicles with influencing drivers' car following behavior. The biggest strength, however, is that it portrays the varying characteristics of drivers under normal condition on a roadway. The main disadvantage of this technique is that extraction of data from video footage is an extremely painstaking process. It is suggested that the analysis process can

consume to six times as long as the real time recording. Also, the computation of acceleration from such data potentially generates even larger error.

### **3.3.3 Instrumented Vehicle Techniques:**

One of the most sophisticated methods for the extraction of car following parameters has been the use of instrumented vehicles. The first instance of the use of an instrumented vehicle can be found in Helander (1975) even though not directly used for car following behavior analysis. Later, other studies successively improved the technique for car following data collection containing various enhancements (e.g. electro-optical sensors, microwave radar etc) with focus on the performance of adaptive cruise control, human factors and safety concerns. Brackstone et al. (1999) devised the most advanced instrumented vehicle equipped with optical speedometer, radar rangefinder and audio-video monitoring system for car following behavior analysis. Kim (2005) on his dissertation has developed anew data collection system using an instrumented test vehicle, with a synchronized user interface and data acquisition program coupled with two separate CAN networks, GPS, inertial distance measuring instrument, and digital video, has been developed to produce a sufficient quality and quantity of data on real driving behavior. Despite installation and maintenance costs being significantly high, instrumented vehicle technique provide a simpler and more straightforward way of data collection of parameters without any intermediate analysis.

### **3.3.4 The Use of Global Positioning System:**

Although the methodology was originally developed to conduct travel time studies for planning purposes (Quiroga, 1997), various studies such as Inokuchi et al. (1999) used GPS (Global Positioning System) successfully for car following behavior analysis. Imran (2009) and Islam (2013) simultaneously used video recording from an over-bridge and GPS equipped vehicles to collect necessary data. Gurusinghe et al. (2002) proved that the data accuracy from GPS receivers is superior to that of conventional techniques. However, the only downside of this technique is that it can collect data for only a limited number of vehicles at a time.

From the above discussion, a number of issues can be identified about the collection of data for car following behavior in the context of Bangladesh. Because of the cost of installation and high maintenance, it would be impractical to use instrumented vehicles. Despite its better accuracy, the use of GPS is not feasible due to

heterogeneity of traffic. Due to less cost and better possibility of capturing drivers' car following behavior, it was decided to use video technique for data collection.

### **3.4 Site Selection for Data Collection**

The most important step to video recording is to select a suitable site. The site for video recording of traffic flow should be such that it is convenient and efficient to observe vehicle movement. In this section, a description of the necessary steps taken for the selection of sites for video recording has been presented.

#### **3.4.1 Site Selection Criteria**

For successful extraction of data through video recording with minimum, it is a necessary to employ some criteria so that the best site can be identified for data collection. It will help to ensure that the video footage collected from the selected site is explicable with ease and efficiency. The criteria are briefly itemized as follows:

- i. Raised platform for placing camera: There should be an elevated place (e.g. building, foot-over Bridge) at the site from where video data can be collected conveniently.
- ii. Continuous movement of vehicle: There should be continuous movement in the site with a reasonable speed.
- iii. Straight portion of road: The road section should be free from any bend or curve portion so that flow for a reasonably straight road length can be obtained. The straight road should be with good visibility and there should be a divider to make it a dual carriageway
- iv. Minimal non- motorized vehicles: The site should be free from non-motorized vehicles and the composition of traffic should be such that various types of vehicles are represented in a significant proportion.
- v. Minimal side friction: The roadside should be free as much as possible from various side frictions such as pedestrian or vendor activity, bus stoppage, parked vehicles etc.
- vi. Avoidance of large commercial vehicles: Image processing software will be used for vehicle counting which works on the basis of only frontal view of traffic and therefore the large vehicle operating road section is not expected (as it obstructs the frontal view of small sized vehicles like private car and auto-rickshaw).

- vii. Minimal obstacle from trees: As most roads in Dhaka have trees on the divider and on roadside, the site in concern should have as less obstacle from trees as possible.
- viii. Less traffic volume: The traffic volume at the site should not be extensively high, so that vision of vehicles is not obstructed by adjacent vehicles placed laterally on the carriageway.
- ix. Minimal lateral traffic movement: Lateral movements of vehicles such as overtaking and lane changing should be a minimum at the site.
- x. Accessibility to lateral gap information: As one of the parameters of car following model would be lateral gap between two vehicles, the camera should be placed in such a platform that would provide the information without much complication.
- xi. Defined section of a road: As image processing would be used to extract data, the section of the road should be in proportion to the pixel numbers. If the section of the road is excessively long, it would be problematic to define the pixels against different sections of the road.

However, from practical point of view, it is impossible to find a site in Dhaka city meeting all these criteria. As a result, it is necessary to sacrifice one or two of these criteria to find a suitable site. However, some of these criteria cannot be sacrificed at all; otherwise the study will not be successful.

### **3.4.2 Reconnaissance Survey**

Before selecting a site for data collection, it is of utmost importance to do a preliminary survey. A reconnaissance survey was conducted at various locations of Dhaka city to check whether the aforementioned criteria are satisfied or to what extent they are satisfied. Few surveys were done from longitudinal view (camera parallel to traffic) and few were done from lateral view (camera perpendicular to traffic). Possible problems to collect the necessary data were identified.

#### **3.4.2.1 Problems and features identified from reconnaissance survey**

Some common problems and features are identified from the reconnaissance survey. Based on this features final site for data collection and methodology for data extraction will be selected.

#### Problems:

- a) One of the surveys was conducted from a foot-over bridge. The main problem with surveying on the foot-over bridge is vibration. While people walked over the bridge, it generated such vibrations that the camera could not be hold still. (Figure 3-1)
- b) Even though, the surveyed road was meant for motorized vehicle only, a large number of rickshaws still seemed to move in a row.
- c) The gap between the buses was so critical that it was impossible to identify from surveys done longitudinal view.
- d) The sunlight when changes its intensity affects the image processing. It is identified as an object in the image processing. Therefore, the time of the surveying should be chosen wisely.
- e) As image processing would be used to extract data, the section of the road being excessively long would be problematic to define the pixels against different sections of the road. Therefore camera should be set in perpendicular position to the traffic.
- f) The trees and light stands in the middle of the island act an obstacle for image processing. A section where least obstacles of this kind should be selected.
- g) The view of the vehicle should not be so small that it would be negligible for image processing. (Figure 3-1).



**Figure 3-1:** Reconnaissance Survey in Different Roads of Dhaka City.

#### Features:

- a) Even though the road was meant for only two cars to move side by side, the reality was that there were 5 to 6 vehicles occupying the same roadway.
- b) As there are multiple vehicles in a section of a road, the car following behavior of a particular vehicle not only depends on the front vehicle in front of it, but also influenced by the movement of side vehicles.

- c) While moving in a compact space, the vehicle not only has to follow the front vehicle but also has to mind the gap between vehicles on each side.
- d) There could be one or more lead vehicles for a subject vehicle.
- e) Overtaking and lane changing was frequent in the road. Therefore, these data need to be avoided while processing the collected data.
- f) The camera should be as perpendicular to the vehicles as possible. This will eliminate errors in a huge amount.

Reconnaissance survey not only serves the purpose of site selection, but also helps to locate vantage points for video recording. It also enables to become aware of the potential problems which may occur in subsequent work. Reconnaissance survey helps to find the correct methodology to extract data that will provide the most accuracy.

### **3.4.3 Final Site Selection**

Based on the reconnaissance survey and features identified, a site was finally selected for data collection through video recording, on the basis of which the membership functions of the Neuro- Fuzzy models will be calibrated. The site was located at Mirpur road near science laboratory, which is one of the busiest of Dhaka city. The site satisfies six of the seven criteria for site selection enumerated earlier and most importantly, because it is nearby the stop line of the signalized junction at Mirpur road, vehicles with all kinds of longitudinal maneuver such as those which were accelerating, decelerating or maintaining constant speed as well as lateral maneuver such as lane changing and overtaking were readily observed at the site. It is important to mention that, there is a separate lane dedicated only for non-motorized vehicles which limits the mixture of human-powered vehicles with motorized vehicles to some extent. As there is a well- furnished foot-over Bridge provided right above the intersection which eases the pedestrian movement in every side of the intersection, the presence of side friction by haphazard pedestrian movement is less in the selected site. To be more precise, the site selected for video recording was located on the approach side of a T-junction where the vehicles move from New-market area to Mirpur area. There were multi-storied buildings just beside the site, which acted as vantage point for video recording. A 13 storied building was selected where the video was recorded from the eleventh floor as it provides better visibility of the moving vehicles. In this configuration, the video footage would show that vehicles are moving from left to right. As mentioned earlier, this particular location will guarantee the availability of

vehicles with all the possible longitudinal maneuvers such as acceleration and deceleration. As the video is recorded from a high rise building, it avails in getting the lateral distance of both side of a vehicle. In Figure 3-2, a view of the selected site from the high rise building is provided where it shows that top of all vehicles are visible from the position where the camera was set.



**Figure 3-2:** The View of the Final Site Selected for Survey

Like most of the major roads of Dhaka city, it is a dual carriageway, on which non-motorized vehicles are banned even though there was seen a small percentage of non-motorized vehicles moving about by breaking the law. In order to obtain the best result from image processing, several surveys were conducted in different time of the day on the selected site to get the perfect time to record the video. Fortunately, the time of the video recording coincided with the time of the day when high volume of traffic moves about the road. In Figure 3-3, a satellite image of the selected site is shown along with the approximate field of view of the video footage superimposed on it in yellow arrows.



**Figure 3-3:** Satellite view of the selected site

### **3.5 Collection of Data**

In order to calibrate the parameters representing the shapes and positions of the input and output membership functions of the Fuzzy Inference System (FIS), it is necessary to collect significant amount of data at the selected site. In this section, various details about data collection at the selected site have been presented.

In this study, the principal source of raw data was the video footage, which would later be used for the conversion of vehicle coordinates of video footage into real world positions, from which various input and output parameters pertaining to car following theory such as distance, speed and acceleration will be obtained. In addition, some other auxiliary measurements were necessary for the purpose of camera calibration, which is a prerequisite for the aforementioned conversion. However, for a few cases, approximations were made.

#### **3.5.1 Video Recording**

At the selected site, extraction of raw data was carried out by video recording the vehicular movement. An added advantage of video recording was it provided a permanent and comprehensive record ensuring the advantage of the capability to replay the footage as many times as necessary. A Samsung Galaxy s6 edge phone with a camera of 3840 x 2160 pixels was used for video recording. The time for video recording was chosen to be after 4.30pm as sunlight does not affect image and the volume of traffic is moderately high at that hour. However, some part of the video footage was not useful for analysis. Therefore, the duration of video recording must be chosen such that significant amount of data can be extracted from moving vehicles. Considering these issues, the duration of video recording was chosen to be more or less near one hour and thirty minute.

However, although video recording offers the advantage of high quality, comprehensive and reliable data for analysis to convert coordinates of video footage into real world positions, it suffers from the disadvantage that it consumes significant time and labor for extraction of data. This can be reduced by writing a code with any reliable programming languages.

It is necessary to place the video camera on a tripod stand at a place, which provides best visibility. It may be a good idea to select a vantage point, which is not readily visible to the motorists, which will ensure that the driving behavior of motorists will



not be changed from natural behavior after seeing the video camera. As mentioned earlier, the video camera was placed at the eleventh floor of 13-storeied building just beside the selected site, which provided an elevated and unobstructed view of the traffic stream. Horizontally, the camera was placed as close as possible to the property line demarcating the building and footpath.

The following potential problems, which may affect subsequent data analysis, were present at the site of video recording:

- i. Quite often, a smaller vehicle or part of that smaller vehicle got covered by larger vehicles. On the later part of the video, the smaller vehicle could be seen due to the differences on the speed of the two concerned vehicles. As a remedial measure, either such instances were fully discarded, or in case of slight obstruction, approximations were made for the position of the edge, on the basis of its expected position.
- ii. Vehicles on some part of the video were not clearly visible due to presence of trees in the divider and light post. In this case, vehicles were either fully discarded or approximation of the position of the vehicles were made depending on the visibility of the vehicles.
- iii. In some cases, the video footage clearly showed that the following vehicle was following the leader but the stimulus was not from the leader, instead from a difference source such as pedestrians suddenly stepping on the road or the need for public transports loading (or unloading) passengers. These cases were simply discarded from analysis.

The advantage of recording the video with a mobile camera is that no need of digital format conversion in order to find performs frame-by-frame analysis of the video footage.

### **3.5.2 Auxiliary Data**

In addition to the main source of data in the form of video footage as discussed earlier, some other auxiliary data were also collected. These data were required mainly for coordinate conversion and for few other purposes related to data analysis.

In order to know the position of the vehicles, it is essential to know about the coordinates of those vehicles. In order to perform coordinate conversion few data were required. A simple measuring tape was used to obtain the road section covered

by the camera both in parallel and perpendicular direction of the moving vehicles. The same instrument was used to determine the dimensions of a few standard vehicles.

### **3.6 Summary**

In this chapter, the methodology adopted to achieve the objectives of the study has been discussed. This chapter also elaborately explains various aspects of data collection to calibrate the membership functions of Fuzzy Inference Systems (FIS). This includes site selection, video recording and collection of other data.

## CHAPTER 4

### DATA ANALYSIS AND MODEL DEVELOPMENT

#### 4.1 Introduction

As discussed in the previous chapter, deterministic models have few limitations which justify the use of Adaptive Neuro-Fuzzy Inference System for modeling the car following behavior of heterogeneous traffic condition in weak lane discipline. Before modeling, various aspects of analyzing raw video data of vehicular movement collected from field have been elaborately explained. This includes image processing, extraction of vehicle coordinates from the video footage and conversion of these coordinates into various parameters pertaining to car following theory. This discussion is followed by the in-depth description of the development of Neuro-Fuzzy models by using the ‘Neuro-Fuzzy Designer’ toolbox of MATLAB, intending to focus on fuzzy membership functions, fuzzy rules and model calibration using artificial neural network.

#### 4.2 Data Analysis

In order to ease up the process of data analysis, it was decided to use Python programming language to perform necessary coding and to extract data, image processing was used. Among all the programming languages, Python was used since it is quite user friendly and does not have to deal with jargons (unnecessary syntax which often are the root cause of errors) that most programming languages use. For image processing, OpenCV was decided to use since it has simple functionalities and is compatible enough with Python programming language.

##### 4.2.1 Data Extraction Methodologies

Due to the hard work of many researchers and scientists, there exist a few methods to extract nvehicle data. They are Inductive loops, RADAR gun, LASER Gun, Manual count (Adnan &Zainuddin, 2013). But these intrusive and non-intrusive sensors are either too expensive and requires high maintenance or damages the already existing

road. So to overcome the limitations in existing methods, several video-based approaches were proposed for estimating or measuring the speed of vehicles in roadways. Almost all the methods include a background/foreground segmentation step to detect image regions containing motion. The methods of background extraction from color image (Peijiang, 2009).based on average value, median filter (Zhiwei et al., 2007; Maduro et al., 2008) and common region. Few other common approaches for this task include simple frame differences (Madasu et al., 2010; Dogan et al., 2010; Xiao et al., 2007) as well as statistic models based on Gaussian distributions (Grammatikopoulos et al., 2005; Wang et al., 2008) or other measures (Schoepflin & Dailey, 2003). Image tracking is one of the steps to estimate vehicle speed. Some of the common tracking method involves image patches (Grammatikopoulos et al., 2005), edges (Dailey et al., 2000; Zhiwei et al., 2007), blobs (Maduro et al., 2008), corners (Maduro et al., 2008; ;Dogan et al., 2010). Few researchers have successfully tracked vehicles based on the license plate region (Czajewski et al., 2010; Garg & Goel, 2013) or a combination of such features (Palaio et al., 2009). Some methods are sensitive to conditions such as shadows, perspective and illumination variation. Blob analysis [Madasu et al., 2010; Xiao et al., 2007; Maduro et al., 2008; Schoepflin & Dailey, 2003) is one such method which can have high tracking error. Moreover, these methods produce satisfactory results only when the camera is positioned high above the roadway, with the blobs being tracked for many frames.

Many researchers have found ways to avoid the problem associated with blob analysis. Dogan et al. (2010), in his work directly tracked distinctive features using the Lucas-Kanade optical flow algorithm (Lucas & Kanade, 1981). The disadvantage of using this algorithm is that it can handle only a single vehicle at a time. Moreover, they require a side view of the vehicles and do not take perspective into account. Equation of spherical projection is used in (Madasu & Hanmandlu, 2010) to estimate the vehicle speed. Lucas-Kanade-Tomasi algorithm is used for motion tracking. Vehicle detection and tracking is one of the most important steps to estimate vehicle speed. The various algorithms which are developed for vehicle detection and tracking has reviewed in (Wang & Gu, 2008).

While using different methods and algorithms, vehicles might go undetected or wrong-fully detected. To remove misdetection of vehicle (Ling et al., 2012) due to vehicle travelling from other lane, or other small movements such as tree waving can be avoided using ROI extraction. The background is multiplied with ROI mask. So that vehicles are detected accurately. Furthermore thresholding and morphological

operations are used to reduce noise. In thresholding selection of threshold value is based on various methods. The threshold value can be selected manually or automatically by using thresholding. The histogram thresholding is explained in (Daniel et al., 2012). The next section has covered the proposed method for determining vehicle speed.

#### **4.2.2 Image Processing with OpenCV**

The parameters of the car-following model needed to be extracted using image processing by OpenCV libraries. Many image processing techniques, such as image segmentation, thresholding, Lucas-Kanade-Tomasi, HaarCasacade Classifier, SHIFT function, have been tried for extracting different parameters before using the proposed technique for heterogeneous traffic condition.

Image segmentation is a major preprocessing step in many vision based applications. It helps to extract the moving foreground from static background. This approach combines local and global thresholding techniques to take advantage of the computational efficiency of the former and the accuracy of the latter. It provides good results in segmentation, and allows automating the process when foreground color of images is not constant, as well as speeding it up significantly. The shortcoming of using image segmentation is it identifies the change in sunlight as a different object.

Thresholding means converting image into binary format. It is important for image processing. Thresholding image can separate dark and light side of the colorful image. A binary image can be converted to any format. In image processing, If pixel value is greater than a threshold value, it is assigned one value (may be white), else it is assigned another value (may be black). The function used is `cv2.threshold`. The limitation of using threshold is its incapability to define all the colors at the same time.

Lucas-Kanade-Tomasi algorithm is used for motion tracking. The disadvantage of using this algorithm is that it can handle only a single vehicle at a time. Moreover, they require a side view of the vehicles and do not take perspective into account. HaarCasacade Classifier, one of the techniques, has been used to detect and classify vehicles which can be a cumbersome technique. One other limitation is since it is difficult to crop the vehicles from road; the classifier identifies the road to as an object. SHIFT function is also cumbersome as HaarCasacade Classifier.

One of the common limitations of above mentioned technique is that with the change in the intensity of sunlight image processing gets complicated. Not only that sunlight is detected as an object with all the techniques but also it distorts the original color of a vehicle. It also adds shadow to each vehicle because of which the accurate parameters are never found. In order to overcome the problem, a specific time of the day has been selected for data collection when the intensity of the sunlight is minimal. On the following section, a new technique to extract data for heterogeneous traffic has been proposed where only the top of the vehicle is detected.

#### **4.2.2.1 Proposed technique**

For many reasons such as shadow of the vehicles, intensity of sunlight, the closeness of one vehicle to another etc., the parameters of car-following model on heterogeneous traffic condition is difficult to extract than for homogeneous traffic condition. A new technique has been proposed in order to fulfil the objective of this study. In this proposed method, only the top of the vehicle was identified and then used for necessary data extraction. The image processing technique for data extraction of car following model for heterogeneous traffic condition is described in the following. The whole process has been described using a bus as an example.

##### **4.2.2.1.1 Pre-processing**

A camera has been used for recording vehicle movement. Before even beginning the image processing, it is vital to convert the video into the frames. At first, cv2.VideoCapture function is used where the video is passed as an argument. This helps load up the video for image processing. In pre-processing the video has converted into the frames. In OpenCV, the read function is used to get each frame.

##### **4.2.2.1.2 Grayscale image generation**

To get better results, vehicle detection process should be performed in the grayscale image domain. There are few benefits towards using a grayscale image.

1. With modern computers, and with parallel programming, it is possible to perform simple pixel-by-pixel processing of a megapixel image in milliseconds. Facial recognition, OCR, content-aware resizing, mean shift

segmentation, and other tasks can take much longer than that. Grayscale image helps in this context.

2. Sometimes it is harder to process a colored image as it decreases the speed of processing an image. If color is not needed, it is considered a noise. Hence a RGB to grayscale conversion is performed on each video frame.

To achieve an appropriate threshold level and make results more suitable than the input image, each frame should be brought in contrast to background. Among several grayscale transformations, the simplest form of grayscale transformation has been used in this work. For color conversion we use the function `cv2.cvtColor(input_image, flag)` where `flag` determines the type of conversion. To convert to grayscale, we use flag `cv2.COLOR_BGR2GRAY`. Figure 4-1 demonstrates the grayscale image that can be obtained using OpenCV.



**Figure 4-1:** Grayscale Image

#### **4.2.2.1.3 Image blurring**

Image blurring is achieved by convolving the image with a low-pass filter kernel. It is useful for removing noise. It actually removes high frequency content (e.g: noise, edges) from the image resulting in edges being blurred when this filter is applied. OpenCV provides mainly four types of blurring techniques. They are – Averaging, Gaussian Filtering, Median Filtering, and Bilateral Filtering. Among these filters, Gaussian Filter has been used in this work. The Gaussian Filter is a type of image-blurring filters that uses a Gaussian function (which also expresses the normal distribution in statistics) for calculating the transformation to apply to each pixel in the image.

In this approach, instead of a box filter consisting of equal filter coefficients, a Gaussian kernel is used. It is done with the function, `cv2.GaussianBlur (image, kernel size, sigmaX, sigmaY)`. The width and height of the kernel should be positive and odd. The standard deviation in the X and Y directions are `sigmaX` and `sigmaY` respectively. If only `sigmaX` is specified, `sigmaY` is taken as equal to `sigmaX`. If both are given as zeros, they are calculated from the kernel size. For our proposal, `cv2.GaussianBlur (image, 5,5), 0)` has been used. Figure 4-2 shows the blurred image generated by OpenCV.



**Figure 4-2: Blurred Image**

#### **4.2.2.1.4 Vehicle detection by color determination**

Many researchers have attempted various methods to obtain the data required for model preparation. Due to congestion of traffic in the road and few other factors described in the previous section, data extraction of heterogeneous traffic stream is complicated and cumbersome than of homogeneous traffic stream. In this study, an attempt has been made to detect vehicles using color determination.

##### **4.2.2.1.4.1 What is the color method (Hue, Saturation, Value)**

Color is the visual byproduct of the spectrum of light as it is either transmitted through a transparent medium, or as it is absorbed and reflected off a surface. Color is the light wavelengths that the human eye receives and processes from a reflected source.



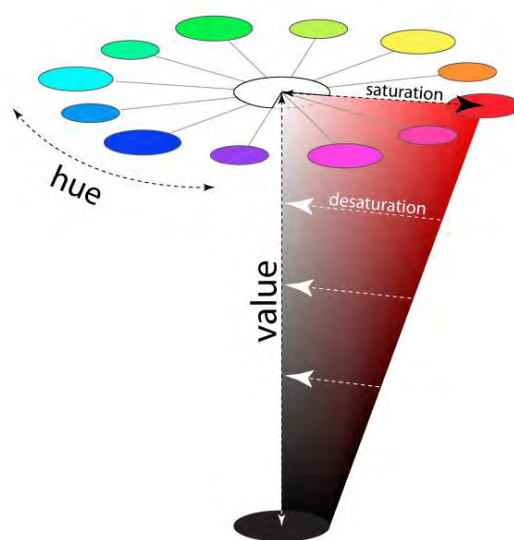
Color consists of three integral components:

1. Hue
2. Saturation
3. Value

Hue is one of the main properties a color, defined technically as the degree to which a stimulus can be described as similar to or different from stimuli that are described as red, green, blue and yellow. The hue (H) of a color refers to which pure color it resembles. The three primary hues in light are red, green, and blue.

Saturation defines intensity of a color. When a pigment hue is toned both white and black (grey) are added to the color to reduce the color's saturation. In terms of the additive light color model, though, saturation works on a scale based on how much or how little other hues are represented in the color.

Value refers to the lightness or darkness of a color. It indicates the quantity of light reflected. When referring to pigments, dark values with black when added are called "shades" of the given hue name. Light values with white pigment added are called "tints" of the hue name. Figure 4-3 demonstrate Hue, Saturation and Value using the color red.



**Figure 4-3:** Hue, Saturation & Value

#### 4.2.2.1.4.2 Steps to attain the color method

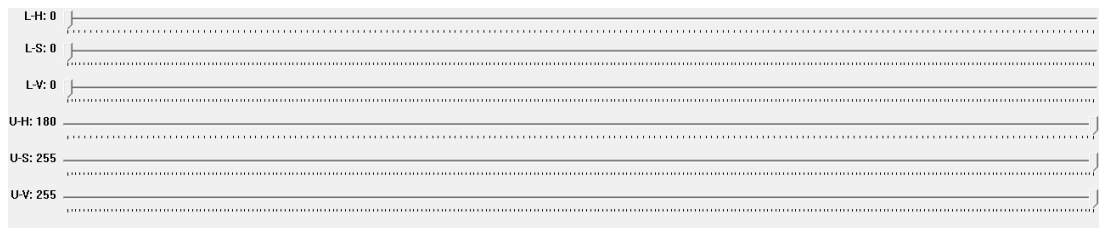
Color segregation can be achieved by extracting a particular HSV (hue, saturation, value) from an image. The algorithm is simple and the main steps are as follows:

### Step 1 - BGR to HSV Conversion

There are more than 150 color-space conversion methods available in OpenCV. For this study, only BGR to HSV conversion will be discussed briefly. For color conversion, the function used is `cv2.cvtColor(image, flag)` where the flag determines the type of conversion. For BGR to HSV conversion, `cv2.COLOR_BGR2HSV` flag is used.

### Step 2 – Apply a Threshold Mask

To mask the newly converted HSV image, the first step is to assign lower and upper range of the color which needed to be masked. For example, for the color red, the lower range is [170, 41, 115] and upper range is [180, 153, 255], both of which are assigned as numpy array. The next step is to mask the image using the function `cv2.inRange`. This function takes the image as the first argument and lower and upper range as the second and third argument respectively. Since color varies from one vehicle to another, it was decided to make a bar which would contain the upper and lower values of hue, saturation and value. Figure 4-4 shows the bars that were used to identify a color.

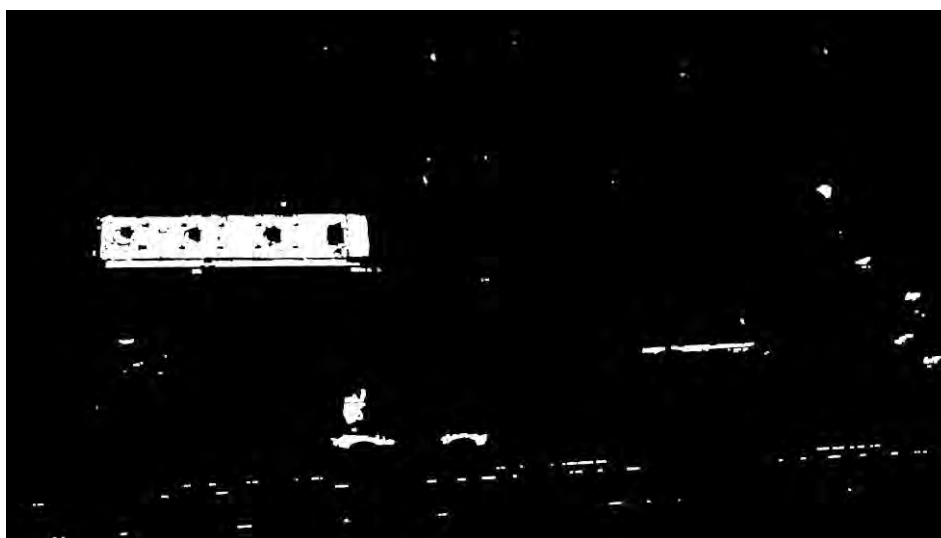


**Figure 4-4:** The Upper and Lower Range for Hue, Value and Saturation is shown in form of Movable Bars

#### 4.2.2.1.5 Morphological transformation

Morphological transformations are some simple operations based on the image shape. It is normally performed on binary images. It needs two inputs, one is our original image, and second one is called structuring element or kernel which decides the nature of operation. Two basic morphological operators are Erosion and Dilation. The basic idea of erosion is just like soil erosion only; it erodes away the boundaries of foreground object. It is just opposite of erosion. Here, kernel is taken 1 in compliance with pixel element. So it increases the white region in the image or size of foreground object increases. In this work, erosion is followed by dilation. It is because, erosion removes white noises, but it also shrinks our object. So we dilate it. Here, `cv2.dilate` and `cv2.erode` has been used for dilation and erosion respectively.

To complete image processing, morphological transformation such as opening and closing has been done. Opening and closing are the variant forms of dilation and erosion. Opening is just another form of erosion followed by dilation. It is useful in removing noise. Here the function `cv2.morphologyEx` has been used. First argument is the image processed so far. Second argument is a flag and the third argument is kernel. Flags such as `cv2.MORPH_OPEN` and `cv2.MORPH_CLOSE` are used for opening and closing respectively. Kernel here is a numpy matrix which consists value one. Figure 4-5 demonstrates the view of the image after all the morphological transformations.



**Figure 4-5:** Morphological Transformation

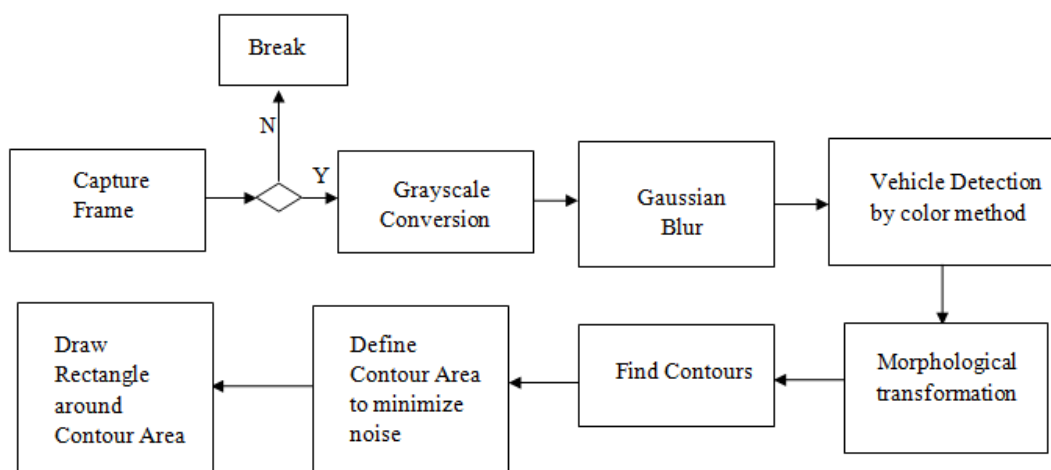
#### **4.2.2.1.6 Finding contours**

Each image (video frame) has three significant features to achieve detection goals. These features include: edges, contours and points. Among mentioned features, an appropriate option is to find contour. Contours can be easily drawn using the function `cv2.drawContours`. But for this work, `cv2.findContours` function has been used. This is Contour Approximation Method. This method helps to store all the coordinates of an object. The first argument is the processed image so far, the second argument is the algorithm used to find contour and the third argument is the flag used to determine how many coordinates we need to store. For the second argument, `cv2.RETR_TREE` has been passed. The flag, `cv2.CHAIN_APPROX_NONE`, needs to be used to store all the coordinates of the boundary. Since two end points of the line are required, the flag, `cv2.CHAIN_APPROX_SIMPLE`, has been used here. `cv2.boundingRect`

function has been used which helps extracting the coordinates of the contour. The coordinates can further be used to bind the vehicle with rectangle. It should be mentioned that all the other unnecessary noise can be eliminated by selecting a maximum area to identify. Figure 4-6 shows the final contoured area of the bus with green rectangle. Figure 4-7 summarizes the steps to image processing.



**Figure 4-6: Final contoured Area of the Bus**



**Figure 4-7: Steps to Image Processing**

#### 4.2.2.2 Feature extraction

After defining the contour, the next step is to extract features in order to make the model. The details into the feature extraction are discussed here.

#### 4.2.2.2.1 Coordinate conversion

In the previous section, it is described how to identify the vehicle and define contour around it. The four corners of the vehicle can be found from the contouring process. These coordinates need to be converted to real world data for further use. Here, the auxiliary data collected before in Chapter 3 would be utilized. In order to convert the coordinates, a ruler needed to be used which ease the process of conversion. Afterwards basic rule of linear proportionality was used to obtain the coordinates in pixel values.

$$\frac{x}{\text{Length of the point in ruler}} = \frac{\text{Pixel value of the whole screen}}{\text{Length of the whole screen in ruler}} \quad (4.1)$$

Here  $x$  is unknown pixel value of the point in concern. The pixel value of the whole screen of the laptop can easily be found using OpenCV. Since the real dimension which covers the whole screen of the laptop is already known (Auxiliary data), this can be further used to find the length of the road section in real world data using the same rule of linear proportionality. The distance between two points can be found by the difference between two point information. In this way, the longitudinal distance and the lateral distances of a vehicle from its surrounding vehicles can be obtained easily.

It is necessary to mention that, it was seen that the camera is slightly slanted which distort the view to be slightly non-parallel to the movement of vehicles. This error is insignificant as vehicles never maintain the parallel position with the lane or road section. But in order to eliminate this insignificant error, the error was first calculated using basic trigonometric formula and length of the road section was adjusted.

#### 4.2.2.2.2 Speed determination

Finally, once the vehicles are tracked, we can determine the speed of vehicles on the road. Vehicles are tracked from the distance of centers from one frame to another frame. The real world distance is needed to be mapped on the image. It is convenient to take the real world distance as the same length as the image width since image width can be found using OpenCV library. Euclidean distance of two consecutive center of a blob is taken as the pixel distance. Frame rate of the video can be found

from OpenCV library. Total time taken for a vehicle to pass the screen can be found from dividing image width by frame rate.

$$Totaltime = \frac{Image\ width}{Frame\ rate} \quad (4.2)$$

Speed of the vehicle is then found from dividing the multiplication of real world distance and pixel distance by total time.

$$Speed = \frac{(Pixel\ distance * real\ world\ distance)}{Total\ time} \quad (4.3)$$

#### 4.2.2.2.3 Acceleration determination

Acceleration of a vehicle can be defined as its rate of change of speed. In this study, the calculation of instantaneous acceleration of a vehicle requires three successive real world positions of one of its edges. However, there is no straightforward formula available for the calculation of acceleration, when the only available data is three successive positions of a vehicle. Fortunately, it is possible to manipulate basic equations of motion, with the intention to produce an expression for acceleration, based on the condition that a vehicle moving at an instantaneously uniform acceleration traverses known speeds in two successive position and equal time intervals (0.25 seconds in this study). If speed of two consecutive position ( $v_1$  and  $v_2$ ) is known, the acceleration,  $a$  can be found.

$$a = \frac{v_2 - v_1}{\Delta t^2} \quad (4.4)$$

### 4.3 Fuzzy Inference System Development

In the previous sections of this chapter, various issues pertaining to data analysis have been elaborately discussed, which leads to the measurement of position, speed and acceleration. After the phase of data analysis, comes the crucial phase of model development. In this study, a model representing car following behavior in heterogeneous road traffic condition in weak lane discipline will be developed by using the Adaptive Neuro-Fuzzy Inference System (ANFIS). The essence of the

platform of ANFIS is that the car following parameters will be represented by fuzzy membership functions and will be connected by a number of fuzzy rules. In order to replicate the relationships between input and output parameters as observed in field (from video footage snapshots), the geometric shapes and positions of input and output membership functions (depicted by various parameters) will be tailored by using artificial neural network.

It is widely known that the majority of motorized vehicles running on the roads of Dhaka city are mainly constitute auto-rickshaws, cars, buses and trucks. In this study, only car-car following behavior in the heterogeneous road traffic condition in weak lane discipline has been modeled using Adapting Neuro-Fuzzy Inference System. In the following subsections, various aspects of the development of the FIS have been discussed on a step by step basis.

### **4.3.1 Inputs and Outputs**

At this point, it is necessary to recall the basic stimulus-response based car following models discussed in Chapter 2. The features of the deterministic models suggest that, given a set of definitive or crisp inputs, the model will produce a definitive or crisp output. In this study, it was decided to use input parameters that replicate more of the heterogeneous road traffic condition and output parameters that were observed in the basic stimulus-response based car following models, although in an imprecise or fuzzy way.

As the models developed in this study were based on a Fuzzy Inference System, it is necessary to categorize each of the inputs into two or more uniquely named linguistic categories, which would be assigned separate membership functions automatically by the ‘Neuro-Fuzzy Designer’ toolbox of MATLAB, as discussed later. However, nothing similar was done for the output, because the system will be calibrated by artificial neural network to create an ANFIS, which requires all the output membership functions to be singletons.

#### **4.3.1.1 Inputs**

A total of six inputs were used in the study to produce the one and only output. A description of each of the inputs along with their various technical features is given as follows:

- a) The first input is the 'Relative Speed' (in meters per second), designated as RS in this study. It is widely accepted as one of the most primary inputs in any car following model. It is defined as the difference in speed between a vehicle and the one immediately in front of it at the same instant. It can have both positive and negative values, based on the convention that if the speed of the leader is greater than that of the follower, RS will be positive, while the reverse will be true for an opposite case.
- b) The second input is the 'Distance' (in meters), designated as DS in this study. It is merely the clear distance between the rear bumper of the leader and the front bumper of the follower. Unlike relative speed, this quantity can have positive values only. It can be calculated by simply subtracting the real world position of the follower from that of the leader.
- c) The third input is the 'Acceleration of the Leader' (in meters per second per second), designated as AL in this study. This self explanatory quantity is useful for the explanation of the car following behavior of a vehicle, subjected to various speed disturbances created by the leader. It can take both positive and negative values.
- d) The fourth input is the 'Speed of the Follower' (in meters per second), designated as SF in this study. This exclusively positive and self explanatory quantity is an ideal one to establish a stable following distance between the leader and follower for various speeds of the follower, which would later be useful for model validation.
- e) The fifth input is the 'Left Lateral Distance' (in meters), designated as LLDS in this study. This input is specially introduced to introduce weak lane discipline into this model. It is nothing but the lateral distance between the follower and the vehicle on its left side. Like 'Distance' designated by DS, this quantity can have positive values only. It can be calculated by subtracting the real world position of the left side vehicle from that of the follower.
- f) The sixth input is the 'Right Lateral Distance' (in meters), designated as RLDS in this study. This input is also introduced to introduce weak lane discipline into this model. It is nothing but the lateral distance between the follower and the vehicle on its right side. Like 'Distance' designated by DS and 'Left Lateral Distance' designated by LLDS, this quantity can have positive values only. It can be calculated by subtracting the real world position of the follower from that of the vehicle on its right side.



### 4.3.1.2 Outputs

As discussed earlier, the output parameters would be taken same as in the stimulus-response based car following models. The only output parameter of the models is the 'Acceleration of the Follower' (in meters per second per second), designated as AF in this study. Therefore, given a particular scenario perceived by the follower in terms of the aforementioned inputs, the response (or output) will be exhibited by the follower by accelerating in a certain manner. Although theoretically it is called 'acceleration', the numerical value of the output can be both positive and negative for acceleration and deceleration respectively.

It is necessary to mention in this stage that, in the development of ANFIS prediction model, the available data are usually divided into randomly selected subsets. The first subset is known as the training and testing subset. This data set is used to develop and calibrate the model. The second data set is known as the validation dataset. This dataset was not used in the development of the model, is utilized to validate the performance of the trained model. For this study, 80% of the master dataset was used for training and testing. The remaining 30% was set aside for model validation.

After the defining the inputs and output parameters, the next step of the development of the model is to import all inputs and outputs in the “Neuro-Fuzzy Designer” toolbox of MATLAB. There are two ways to do it. One of them is to import file from directory. The other one is to import it from MATLAB “Workspace.

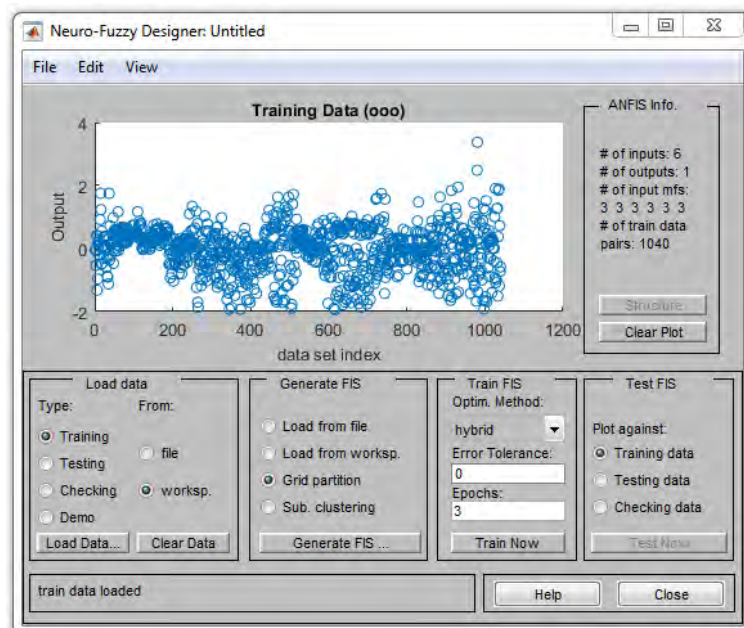


Figure 4-8: Imported data in the “Neuro-Fuzzy Designer” toolbox

It is more convenient to import from workspace than as file from directory as in the workspace the data are arranged in a spreadsheet with different columns dedicated for each parameters.

Data can be imported as file but might require rearranging. It is important to mention that all the parameters both from inputs and output needs to be imported in the “Neuro-Fuzzy Designer” toolbox of MATLAB. As there are six input parameters and one output parameters, all the data are imported together from workspace in the imported in the “Neuro-Fuzzy Designer” toolbox of MATLAB.

Figure 4-8 demonstrates the scattering of all the output parameter. As total data available was 1300 and 80% of it was used for training and testing purposes, there are exactly 1040 output data shown in Figure 4-8. It is important to mention here that in order to eliminate the possibility of over-fitting of the membership functions solely on the basis of the training data set and thereby to ensure generalization of the models, the ‘Neuro-Fuzzy Designer’ of MATLAB suggests that a significant and randomly selected portion of the overall data arrays should be separated from it and be designated as checking data set. The basic feature of the use of checking data set in the model calibration process is that it enables to adopt the configurations of input and output membership functions as final calibration result corresponding to the one, for which the checking error ('Root Mean Squared Error' of the outputs of the checking data against the corresponding model outputs) is minimum even though the training error may not be minimum at that point. It is contrary to the case, where no checking data is used and the membership function configurations corresponding to the point where training error ('Root Mean Squared Error' of the outputs of the training data against the corresponding model outputs) is smallest, is adopted as the final calibration result (usually the last epoch of overall calibration). For this particular no checking data is loaded to make the model. The next step towards making the model is to define the linguistic categories of all input and output parameters. The linguistic categories help to fuzzify the model. The behavior of a model may vary with the number of linguistic categories. The following paragraph provides the details to all the linguistic categories.

In Table 4-1, various features of all the inputs are summarized such as name and designation, which also includes the number and name of each linguistic categories into which each input is divided. It should be noted that all the input parameters discussed above are included in the Neuro-Fuzzy model, corresponding to the same

instant of time. That is, for a particular input-output combination, all these six input parameters were calculated at the same instant as suggested by classical models.

**Table 4-1:** Summary of Input Parameters

Serial Number	Input Name	Input Designation	Number of Categories	Category Names
1	Relative Speed	RS	3	Negative
				Nothing
				Positive
2	Distance	DS	2	Inadequate
				Adequate
3	Acceleration of Leader	AL	3	Deceleration
				Zero
				Acceleration
4	Speed of Follower	SF	2	Slow
				Fast
5	Left Lateral Distance	LLDS	2	Inadequate
				Adequate
6	Right Lateral Distance	RLDS	2	Inadequate
				Adequate

As mentioned earlier, no linguistic categories needed to be defined for the output parameter. An important aspect of the output is that it is temporally lagged from the instant representing all the inputs, as suggested by classical car following models discussed in Chapter 2. Throughout this study, a uniform time lag of 1 second will be adopted, although it can vary widely depending on various factors and there have been attempts to account for it. The uniform time lag of 1second was considered most reasonable one from practical point of view of data analysis. It could have been a good idea to try values of time lag greater than I second, but due to limited field of view of the camera, the selection of a value of time lag greater than second would lead to omission of significant amount of trajectory data. The step before defining the linguistic categories is shown in the following picture.

In Table 4-2 demonstrates an example which is the obtainment of all the input and output variables of the Fuzzy Inference System (FIS) in a usable format for model calibration. All the samples have been presented in Appendix B.

**Table 4-2:** An Example of all the Input and Output that was used to Develop the Model

RS	DS	AL	SF	LLDS	RLDS	AF
0	1.068719	0.036222	1.003941	0.441127	1.693262	-0.04358
0.02532	1.068095	0.037213	0.976295	0.447913	1.69749	-0.02396
0.056205	1.058694	-0.44964	0.973513	0.447913	1.701781	-0.53355
0.018982	1.045754	0	1.010736	0.441127	1.713158	-0.06051
0.00346	1.033511	1.348921	0.94887	0.447913	1.710549	0.423097
0.143255	1.030441	-1.22857	1.003941	0.4547	1.715027	-1.32173
0.028103	1.00101	1.87965	1.001615	0.447913	1.719567	0.395057
0.142069	0.994281	-0.92418	0.94541	0.441127	1.731195	-0.96941
-0.11149	0.624364	0.037622	1.115427	0.739735	1.33323	0.188483
-0.13819	0.597218	0	1.142128	0.739735	1.337254	0.072043
-0.08586	0.563285	0.037213	1.087479	0.739735	1.348126	0.27552
-0.08431	0.542925	-0.44964	1.114025	0.726162	1.365844	-0.2045
-0.03502	0.522565	0	1.064739	0.719376	1.376836	0.210983
-0.25292	0.515779	1.348921	1.198333	0.726162	1.374315	0.918396
0.047592	0.4547	1.828571	1.099604	0.726162	1.385428	1.756805

### 4.3.2 Membership Functions

The input membership functions of a fuzzy inference system are meant for the fuzzification of the input parameters. The membership value varies between 0 and 1 for each linguistic category. However, because the models will be calibrated by artificial neural network, the output membership functions, which are mainly meant for defuzzification of fuzzy values to produce a crisp output by using the weighted average method (the only option available for ANFIS), are required to be nothing but singletons, thereby not requiring to be assigned any particular shape of membership function. The ‘Neuro-Fuzzy Designer’ toolbox of MATLAB offers membership functions with various geometric shapes, such as triangular, trapezium, Gaussian, bell shaped, sigmoid etc. A set of sample were used to find out the membership function needed to be used to make the model.

**Table 4-3:** Predicted Outputs Generated by All the Membership Functions

Actual	gauss	disig	tri	tra	gbell	guass2mf	pimf	psigmf
-0.5454	-0.5454	-0.5452	-0.5453	-0.5454	-0.5454	-0.5455	-0.545	-0.5452
-0.3439	-0.3439	-0.344	-0.3439	-0.3439	-0.3438	-0.3438	-0.344	-0.344
-0.6111	-0.6112	-0.6021	-0.6108	-0.5956	-0.6116	-0.614	-0.312	-0.6021

<b>Actual</b>	<b>gauss</b>	<b>disig</b>	<b>tri</b>	<b>tra</b>	<b>gbell</b>	<b>guass2mf</b>	<b>pimf</b>	<b>psigmf</b>
0.22109	0.22068	0.2192	0.2207	0.2112	0.22103	0.22343	0.1001	0.2192
0.74149	0.74311	0.7597	0.7416	0.74148	0.74352	0.74324	0.7414	0.7597
-0.875	-0.8743	-0.8802	-0.875	-0.8757	-0.8737	-0.8743	-0.876	-0.8802
0.1076	0.10702	0.0495	0.1076	0.20349	0.10521	0.0991	0.1953	0.0495
0.30584	0.3014	0.1714	0.3059	0.16414	0.3021	0.25513	0.1515	0.1714
-0.2866	-0.2878	0.114	-0.2866	0.14447	-0.282	-0.2356	0.1234	0.114
0.30584	0.30556	0.3805	0.3056	0.115	0.30759	0.32129	0.0736	0.3805
0.38769	0.38615	0.0197	0.3876	0.08554	0.38372	0.43745	0.0267	0.0197
0.4517	0.4492	0.278	0.4517	0.43791	0.44584	0.39881	-0.207	0.278
-1.5183	-1.5182	-1.5186	-1.5183	-1.5185	-1.5182	-1.5181	-1.518	-1.5186
0.24961	0.2511	0.2437	0.2494	0.22362	0.21588	0.22834	0.2745	0.2437
0.03768	0.03837	0.0294	0.0384	0.07519	0.06628	0.05873	-0.034	0.0294
0.26496	0.26492	0.2311	0.2648	-0.0321	0.27016	0.30568	-0.067	0.2311
0.03852	0.06318	-0.1722	0.0389	-0.0492	0.0783	0.00164	-0.068	-0.1722
-0.3702	-0.3598	-0.2294	-0.3708	-0.2027	-0.3643	-0.3107	-0.217	-0.2294
0.03022	0.01303	-0.2143	0.0301	-0.1805	0.0225	-0.0595	-0.209	-0.2143
1.16546	1.16596	1.1801	1.1656	1.01434	1.16927	1.1668	1.118	1.1801
<b>RMSE</b>	<b>0.00028</b>	<b>0.1548</b>	<b>0.0072</b>	<b>0.1638</b>	<b>0.01374</b>	<b>0.03625</b>	<b>0.2379</b>	<b>0.1548</b>

\*RMSE = Root Mean Square Error

In Table 4-3 twenty random samples were taken and a test was conducted using all the membership function provided by MATLAB. Root mean square error was determined for all the membership functions. It was found that Gaussian membership function provides the least of error of all the membership functions. Therefore, Gaussian membership was chosen to make the ANFIS model.

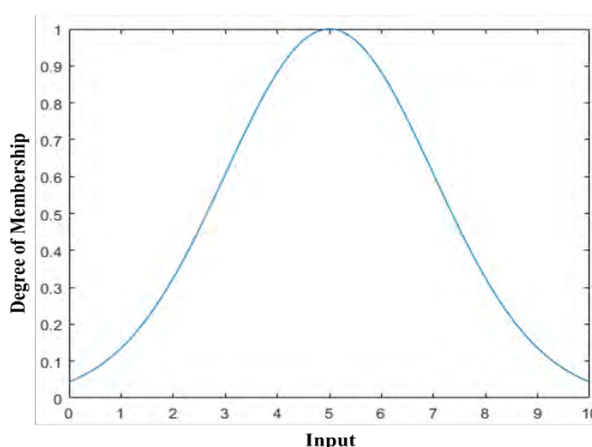
Gaussian membership function computes fuzzy membership values using a Gaussian membership function. A Gaussian membership function is not the same as a Gaussian probability distribution. For example, a Gaussian membership function always has a maximum value of 1. The advantage of Gaussian membership function is that they theoretically produce outputs ranging between 0 and 1 for any value of inputs. Although this is not a unique characteristic of Gaussian functions (also shared by some other types), the biggest advantage of using them is that they ensure smooth transition of outputs throughout its domain, thereby ensuring flexibility. As far as the calibration of the membership functions is concerned, this type of membership

functions do not distort their shape too much even after calibration, while ensuring a good fit between inputs and outputs.

Mathematically, the Gaussian functions contain two parameters governing their geometric shape and can be expressed as follows:

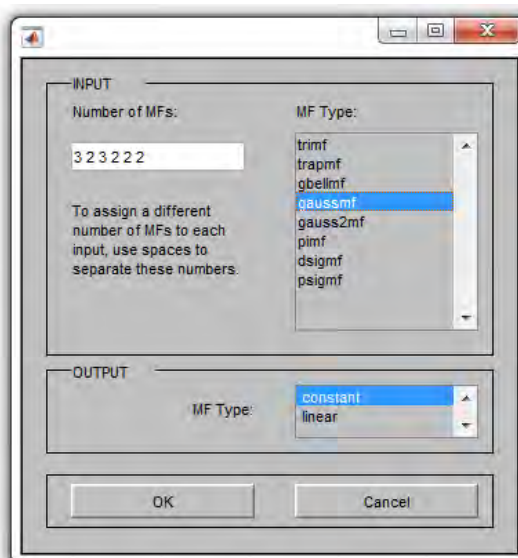
$$f(x, \sigma, c) = e^{-\frac{(x-c)^2}{2\sigma^2}} \quad (4.1)$$

In the above equation, the parameter ‘ $\sigma$ ’ defines the standard deviation of the Gaussian function and ‘ $c$ ’ defines the mean of the function.



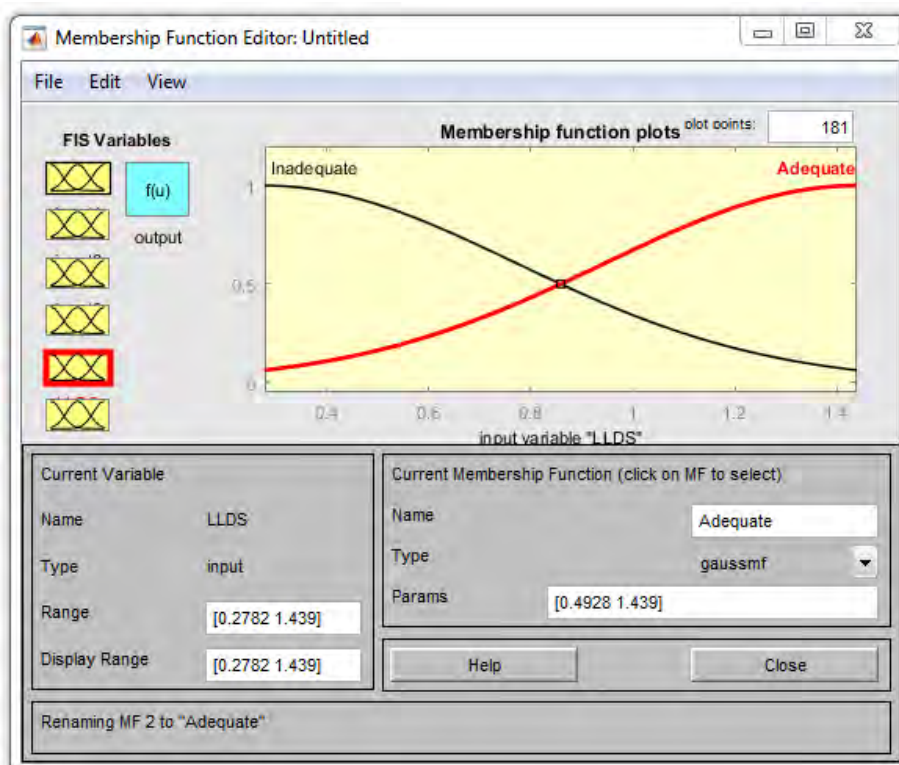
**Figure 4-9:** Formation of Gaussian Membership Function

Figure 4-9 shows the formation of a typical Gaussian membership function. The input values for which to compute membership are specified as scalar or vector. Membership values returned after the training are as scalar or vector.



**Figure 4-10:** Choosing the type of membership function

Figure 4-10 shows the choice provided as membership functions to build the model. It also shows how the number of membership is chosen for every input parameter. There are six digits put between spaces where each digit represents the number of membership functions provided for each input parameters chronologically imported by the workspace as mentioned earlier. There are two options provided for output membership functions. The option ‘constant’ is chosen over the option ‘linear’ as the latter one consumes a lot of time to operate.



**Figure 4-11:** Initialization of Input Membership Functions

The ‘Neuro-Fuzzy Designer’ toolbox of MATLAB provides a convenient way to assign and initialize membership functions. For a given variable, its range must be specified. The maximum and minimum limit of the range is usually convenient to keep close to the expected maximum and minimum value of that variable respectively. However, in case of ANFIS, the limits are automatically reset as equal to the maximum and minimum value of the variable as observed in the data set during training. Finally, the type and number of membership functions spanning over the range of the variables must be specified, which will automatically produce the specified number of membership functions in such a way that all the membership functions are equally spaced and the peak of the first and last membership functions coincide with the minimum and maximum limit of the variable respectively.

In Figure 4-11, an example of initialized Gaussian membership functions for one of the input variables of this study has been shown for one of the models. In Table 4-4 the pre-calibration parameters pertaining to the input membership functions have been presented for the model.

**Table 4-4:** Pre-calibration Parameters Pertaining to the Input Membership Functions

Input Name	Input Designation	Category Names	Membership Function Parameters	
			$\sigma$	c
Relative Speed	RS	Negative	1.622	-2.621
		Nothing	1.622	1.179
		Positive	1.622	5.018
Distance	DS	Inadequate	1.631	0.224
		Adequate	1.631	4.065
Acceleration of Leader	AL	Dec	1.258	-1.991
		Zero	1.258	0.9711
		Acc	1.258	3.933
Speed of Follower	SF	Slow	0.2654	0.6765
		Fast	0.2654	1.303
Left Lateral Distance	LLDS	Inadequate	0.4928	0.2782
		Adequate	0.4928	1.439
Right Lateral Distance	RLDS	Inadequate	0.6535	0.1923
		Adequate	0.6535	1.731

As mentioned earlier,  $\sigma$  defines the standard deviation and c defines the mean of the Gaussian membership function. It can be observed from Table 4-4 that  $\sigma$  of an input is same for all the categories that is under that particular input whereas c varies for all categories.

It is important to mention here that because the system will be calibrated by artificial neural network to form an ANFIS, the output membership functions (which are required to be singletons) cannot be initialized in the aforementioned process. As the total number of output membership functions would be equal to the number of fuzzy rules, as discussed in the next subsection, it is necessary to add each singleton manually. For each singleton, an initial value of zero was chosen, and because the FIS will follow 'Zero Order Sugeno' method. Here, it should be mentioned that, membership function depends on the fuzzy rule that needs to be defined initially.

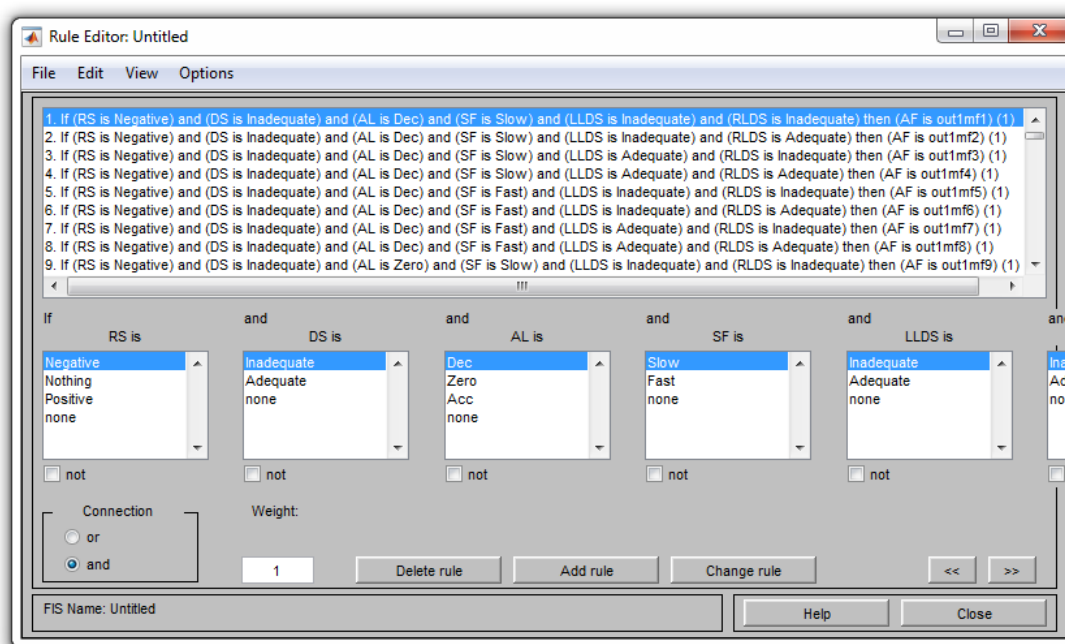


### 4.3.3 Fuzzy Rules

Fuzzy rules are the learning rules for an ANFIS model. It is the one of the characteristic of the ANFIS model that sets it apart from a deterministic model. The underlying principle of fuzzy logic is that it represents each of the input variables in linguistic terms to produce a fuzzy output, by combining one or more inputs with logical operators. The advantage of such principle enables the use of fuzzy logic in various systems (e.g. car following), where imprecision or vagueness is so predominant that the use of deterministic models will not be able to reflect the reality. For example, in the deterministic models of car following, the way of thinking from the point of view of motorist to initiate acceleration (or deceleration) is, "If the relative speed is -1.5 meters per second, the distance is 6 meters, acceleration of leader is -1 meters per second per second, speed of follower is 7 meters per second, the left lateral distance is 0.5354 meter and the right lateral distance is 0.789 meters then accelerate at the rate of -0.8 meters per second per second". It should be mentioned that such notion is unrealistic from the psychological point of view, because the motorist is hardly able to perceive the aforementioned input and output quantities with precision, instead they perceive these quantities vaguely by assigning various linguistic terms. Therefore, the correct way of perception will be, "If the relative speed is 'negative', distance is 'inadequate', acceleration of leader is 'decelerating' and speed of follower is 'fast', then 'decelerate', left lateral distance and right lateral distance is 'adequate' in such and such way". Although the perception thresholds of these variables may vary from motorist to motorist depending on various factors including the vehicle type of the follower, this is one of the established driving rules, which is gained by experience and knowledge.

In this study, each Fuzzy Inference System (FIS) contained a number of 'IF-THEN' type rules connecting the antecedent with consequent to reflect the aforementioned driving rules to model car following behavior. The antecedent part of the rules accommodated each of the six inputs, whereas the consequent part of the rules accommodated the sole output. The total number of rules was dependent on the number of membership functions assigned for each input, in order to ensure that each and every possible combination is included in the fuzzy rule base by means of permutation and combination. In the rule editor of the 'Neuro-Fuzzy Designer' toolbox of MATLAB, all the rules can be assigned interactively. Here, it is important to mention that 'Neuro-Fuzzy Designer' toolbox of MATLAB takes care of the rule making, therefore no need to do it manually. However, it is necessary to assign the linguistic terms of each input variables when input membership functions were

initialized. Figure 4-12 shows a typical example of inputting the name of the linguistic terms (negative, nothing and positive) of the RS input variable.



**Figure 4-12: Rule editor**

It is also important to mention that unlike a conventional Fuzzy Inference System (FIS), the ‘Neuro-Fuzzy Designer’ toolbox of MATLAB imposes a number of constraints on Adaptive Neuro-Fuzzy Inference System (ANFIS). This includes the mandatory requirement that the weights of each fuzzy rule must be unity. Another requirement is that different fuzzy rules must not share the same output membership function, which implies that the number of output membership functions must be equal to the number of fuzzy rules. Therefore, it is not practicable to assign separate linguistic categories for each output membership function. After all the linguistic categories for each input variables are set, ‘Neuro-Fuzzy Designer’ toolbox of Adaptive Neuro-Fuzzy Inference System (ANFIS) automatically creates the Fuzzy rules and each Fuzzy rules counts as a single output membership functions (e.g. mf1, mf2 etc.) as illustrated in the Figure4-12.

It has already been mentioned that the input variables RS and AL were grouped into three linguistic categories each, whereas the ones DS, SF, LLDS and RLDS were grouped into two linguistic categories each. Therefore, the entire rule base contained 144 ( $3 \times 2 \times 3 \times 2 \times 2 \times 2$ ) number of unique fuzzy rules, where each of the four input variables were connected with the logical operator 'AND' (chosen to be arithmetic minimum) ensuring that each of the linguistic terms of each input are connected with

those of all other inputs in every possible way. In Table 4-5, all the fuzzy rules of this study have been listed.

**Table 4-5:** List of Fuzzy Rules Used in the study

Rule Number	IF	AND	AND	AND	AND	AND	THEN
	RS	DS	AL	SF	LLDS	RLDS	AF
1	Negative	Inadequate	Dec	Slow	Inadequate	Inadequate	mf1
2	Negative	Inadequate	Dec	Slow	Inadequate	Adequate	mf2
3	Negative	Inadequate	Dec	Slow	Adequate	Inadequate	mf3
4	Negative	Inadequate	Dec	Slow	Adequate	Adequate	mf4
5	Negative	Inadequate	Dec	Fast	Inadequate	Inadequate	mf5
6	Negative	Inadequate	Dec	Fast	Inadequate	Adequate	mf6
7	Negative	Inadequate	Dec	Fast	Adequate	Inadequate	mf7
8	Negative	Inadequate	Dec	Fast	Adequate	Adequate	mf8
9	Negative	Inadequate	Zero	Slow	Inadequate	Inadequate	mf9
10	Negative	Inadequate	Zero	Slow	Inadequate	Adequate	mf10
11	Negative	Inadequate	Zero	Slow	Adequate	Inadequate	mf11
12	Negative	Inadequate	Zero	Slow	Adequate	Adequate	mf12
13	Negative	Inadequate	Zero	Fast	Inadequate	Inadequate	mf13
14	Negative	Inadequate	Zero	Fast	Inadequate	Adequate	mf14
15	Negative	Inadequate	Zero	Fast	Adequate	Inadequate	mf15
16	Negative	Inadequate	Zero	Fast	Adequate	Adequate	mf16
17	Negative	Inadequate	Acc	Slow	Inadequate	Inadequate	mf17
18	Negative	Inadequate	Acc	Slow	Inadequate	Adequate	mf18
19	Negative	Inadequate	Acc	Slow	Adequate	Inadequate	mf19
20	Negative	Inadequate	Acc	Slow	Adequate	Adequate	mf20
21	Negative	Inadequate	Acc	Fast	Inadequate	Inadequate	mf21
22	Negative	Inadequate	Acc	Fast	Inadequate	Adequate	mf22
23	Negative	Inadequate	Acc	Fast	Adequate	Inadequate	mf23
24	Negative	Inadequate	Acc	Fast	Adequate	Adequate	mf24
25	Negative	Adequate	Dec	Slow	Inadequate	Inadequate	mf25
26	Negative	Adequate	Dec	Slow	Inadequate	Adequate	mf26
27	Negative	Adequate	Dec	Slow	Adequate	Inadequate	mf27
28	Negative	Adequate	Dec	Slow	Adequate	Adequate	mf28
29	Negative	Adequate	Dec	Fast	Inadequate	Inadequate	mf29
30	Negative	Adequate	Dec	Fast	Inadequate	Adequate	mf30
31	Negative	Adequate	Dec	Fast	Adequate	Inadequate	mf31

Rule Number	IF	AND	AND	AND	AND	AND	THEN
	RS	DS	AL	SF	LLDS	RLDS	AF
32	Negative	Adequate	Dec	Fast	Adequate	Adequate	mf32
33	Negative	Adequate	Zero	Slow	Inadequate	Inadequate	mf33
34	Negative	Adequate	Zero	Slow	Inadequate	Adequate	mf34
35	Negative	Adequate	Zero	Slow	Adequate	Inadequate	mf35
36	Negative	Adequate	Zero	Slow	Adequate	Adequate	mf36
37	Negative	Adequate	Zero	Fast	Inadequate	Inadequate	mf37
38	Negative	Adequate	Zero	Fast	Inadequate	Adequate	mf38
39	Negative	Adequate	Zero	Fast	Adequate	Inadequate	mf39
40	Negative	Adequate	Zero	Fast	Adequate	Adequate	mf40
41	Negative	Adequate	Acc	Slow	Inadequate	Inadequate	mf41
42	Negative	Adequate	Acc	Slow	Inadequate	Adequate	mf42
43	Negative	Adequate	Acc	Slow	Adequate	Inadequate	mf43
44	Negative	Adequate	Acc	Slow	Adequate	Adequate	mf44
45	Negative	Adequate	Acc	Fast	Inadequate	Inadequate	mf45
46	Negative	Adequate	Acc	Fast	Inadequate	Adequate	mf46
47	Negative	Adequate	Acc	Fast	Adequate	Inadequate	mf47
48	Negative	Adequate	Acc	Fast	Adequate	Adequate	mf48
49	Nothing	Inadequate	Dec	Slow	Inadequate	Inadequate	mf49
50	Nothing	Inadequate	Dec	Slow	Inadequate	Adequate	mf50
51	Nothing	Inadequate	Dec	Slow	Adequate	Inadequate	mf51
52	Nothing	Inadequate	Dec	Slow	Adequate	Adequate	mf52
53	Nothing	Inadequate	Dec	Fast	Inadequate	Inadequate	mf53
54	Nothing	Inadequate	Dec	Fast	Inadequate	Adequate	mf54
55	Nothing	Inadequate	Dec	Fast	Adequate	Inadequate	mf55
56	Nothing	Inadequate	Dec	Fast	Adequate	Adequate	mf56
57	Nothing	Inadequate	Zero	Slow	Inadequate	Inadequate	mf57
58	Nothing	Inadequate	Zero	Slow	Inadequate	Adequate	mf58
59	Nothing	Inadequate	Zero	Slow	Adequate	Inadequate	mf59
60	Nothing	Inadequate	Zero	Slow	Adequate	Adequate	mf60
61	Nothing	Inadequate	Zero	Fast	Inadequate	Inadequate	mf61
62	Nothing	Inadequate	Zero	Fast	Inadequate	Adequate	mf62
63	Nothing	Inadequate	Zero	Fast	Adequate	Inadequate	mf63
64	Nothing	Inadequate	Zero	Fast	Adequate	Adequate	mf64
65	Nothing	Inadequate	Acc	Slow	Inadequate	Inadequate	mf65
66	Nothing	Inadequate	Acc	Slow	Inadequate	Adequate	mf66

Rule Number	IF	AND	AND	AND	AND	AND	THEN
	RS	DS	AL	SF	LLDS	RLDS	AF
67	Nothing	Inadequate	Acc	Slow	Adequate	Inadequate	mf67
68	Nothing	Inadequate	Acc	Slow	Adequate	Adequate	mf68
69	Nothing	Inadequate	Acc	Fast	Inadequate	Inadequate	mf69
70	Nothing	Inadequate	Acc	Fast	Inadequate	Adequate	mf70
71	Nothing	Inadequate	Acc	Fast	Adequate	Inadequate	mf71
72	Nothing	Inadequate	Acc	Fast	Adequate	Adequate	mf72
73	Nothing	Adequate	Dec	Slow	Inadequate	Inadequate	mf73
74	Nothing	Adequate	Dec	Slow	Inadequate	Adequate	mf74
75	Nothing	Adequate	Dec	Slow	Adequate	Inadequate	mf75
76	Nothing	Adequate	Dec	Slow	Adequate	Adequate	mf76
77	Nothing	Adequate	Dec	Fast	Inadequate	Inadequate	mf77
78	Nothing	Adequate	Dec	Fast	Inadequate	Adequate	mf78
79	Nothing	Adequate	Dec	Fast	Adequate	Inadequate	mf79
80	Nothing	Adequate	Dec	Fast	Adequate	Adequate	mf80
81	Nothing	Adequate	Zero	Slow	Inadequate	Inadequate	mf81
82	Nothing	Adequate	Zero	Slow	Inadequate	Adequate	mf82
83	Nothing	Adequate	Zero	Slow	Adequate	Inadequate	mf83
84	Nothing	Adequate	Zero	Slow	Adequate	Adequate	mf84
85	Nothing	Adequate	Zero	Fast	Inadequate	Inadequate	mf85
86	Nothing	Adequate	Zero	Fast	Inadequate	Adequate	mf86
87	Nothing	Adequate	Zero	Fast	Adequate	Inadequate	mf87
88	Nothing	Adequate	Acc	Fast	Adequate	Adequate	mf88
89	Nothing	Adequate	Acc	Slow	Inadequate	Inadequate	mf89
90	Nothing	Adequate	Acc	Slow	Inadequate	Adequate	mf90
91	Nothing	Adequate	Acc	Slow	Adequate	Inadequate	mf91
92	Nothing	Adequate	Acc	Slow	Adequate	Adequate	mf92
93	Nothing	Adequate	Acc	Fast	Inadequate	Inadequate	mf93
94	Nothing	Adequate	Acc	Fast	Inadequate	Adequate	mf94
95	Nothing	Adequate	Acc	Fast	Adequate	Inadequate	mf95
96	Nothing	Adequate	Acc	Fast	Adequate	Adequate	mf96
97	Positive	Inadequate	Dec	Slow	Inadequate	Inadequate	mf97
98	Positive	Inadequate	Dec	Slow	Inadequate	Adequate	mf98
99	Positive	Inadequate	Dec	Slow	Adequate	Inadequate	mf99
100	Positive	Inadequate	Dec	Slow	Adequate	Adequate	mf100
101	Positive	Inadequate	Dec	Fast	Inadequate	Inadequate	mf101

Rule Number	IF	AND	AND	AND	AND	AND	THEN
	RS	DS	AL	SF	LLDS	RLDS	AF
102	Positive	Inadequate	Dec	Fast	Inadequate	Adequate	mf102
103	Positive	Inadequate	Dec	Fast	Adequate	Inadequate	mf103
104	Positive	Inadequate	Dec	Fast	Adequate	Adequate	mf104
105	Positive	Inadequate	Zero	Slow	Inadequate	Inadequate	mf105
106	Positive	Inadequate	Zero	Slow	Inadequate	Adequate	mf106
107	Positive	Inadequate	Zero	Slow	Adequate	Inadequate	mf107
108	Positive	Inadequate	Zero	Slow	Adequate	Adequate	mf108
109	Positive	Inadequate	Zero	Fast	Inadequate	Inadequate	mf109
110	Positive	Inadequate	Zero	Fast	Inadequate	Adequate	mf110
111	Positive	Inadequate	Zero	Fast	Adequate	Inadequate	mf111
112	Positive	Inadequate	Zero	Fast	Adequate	Adequate	mf112
113	Positive	Inadequate	Acc	Slow	Inadequate	Inadequate	mf113
114	Positive	Inadequate	Acc	Slow	Inadequate	Adequate	mf114
115	Positive	Inadequate	Acc	Slow	Adequate	Inadequate	mf115
116	Positive	Inadequate	Acc	Slow	Adequate	Adequate	mf116
117	Positive	Inadequate	Acc	Fast	Inadequate	Inadequate	mf117
118	Positive	Inadequate	Acc	Fast	Inadequate	Adequate	mf118
119	Positive	Inadequate	Acc	Fast	Adequate	Inadequate	mf119
120	Positive	Adequate	Acc	Fast	Adequate	Adequate	mf120
121	Positive	Adequate	Dec	Slow	Inadequate	Inadequate	mf121
122	Positive	Adequate	Dec	Slow	Inadequate	Adequate	mf122
123	Positive	Adequate	Dec	Slow	Adequate	Inadequate	mf123
124	Positive	Adequate	Dec	Slow	Adequate	Adequate	mf124
125	Positive	Adequate	Dec	Fast	Inadequate	Inadequate	mf125
126	Positive	Adequate	Dec	Fast	Inadequate	Adequate	mf126
127	Positive	Adequate	Dec	Fast	Adequate	Inadequate	mf127
128	Positive	Adequate	Dec	Fast	Adequate	Adequate	mf128
129	Positive	Adequate	Zero	Slow	Inadequate	Inadequate	mf129
130	Positive	Adequate	Zero	Slow	Inadequate	Adequate	mf130
131	Positive	Adequate	Zero	Slow	Adequate	Inadequate	mf131
132	Positive	Adequate	Zero	Slow	Adequate	Adequate	mf132
133	Positive	Adequate	Zero	Fast	Inadequate	Inadequate	mf133
134	Positive	Adequate	Zero	Fast	Inadequate	Adequate	mf134
135	Positive	Adequate	Zero	Fast	Adequate	Inadequate	mf135
136	Positive	Adequate	Zero	Fast	Adequate	Adequate	mf136

Rule Number	IF	AND	AND	AND	AND	AND	THEN
	RS	DS	AL	SF	LLDS	RLDS	AF
137	Positive	Adequate	Acc	Slow	Inadequate	Inadequate	mf137
138	Positive	Adequate	Acc	Slow	Inadequate	Adequate	mf138
139	Positive	Adequate	Acc	Slow	Adequate	Inadequate	mf139
140	Positive	Adequate	Acc	Slow	Adequate	Adequate	mf140
141	Positive	Adequate	Acc	Fast	Inadequate	Inadequate	mf141
142	Positive	Adequate	Acc	Fast	Inadequate	Adequate	mf142
143	Positive	Adequate	Acc	Fast	Adequate	Inadequate	mf143
144	Positive	Adequate	Acc	Fast	Adequate	Adequate	mf144

#### 4.3.4 Neural Network Representation

After defining the inputs and all linguistic categories in the ‘Neuro-Fuzzy Designer’ toolbox of Adaptive Neuro-Fuzzy Inference System (ANFIS) , now comes the neural network part of ANFIS. In a typical artificial neural network model, an infinite number of combinations are possible for the number of layers and number of nodes in each layer, where the best combination can be found in a trial and error method. However, in case of Adaptive Neuro-Fuzzy Inference System (ANFIS), the entire system can be represented in the form of an artificial neural network (Kikuchi and Chakroborty, 1995; Chakroborty and Kikuchi, 2003), where the total number of layers is predetermined, each layer represents a particular component of the system and the number of nodes in each layer depends on key characteristics associated with the system.

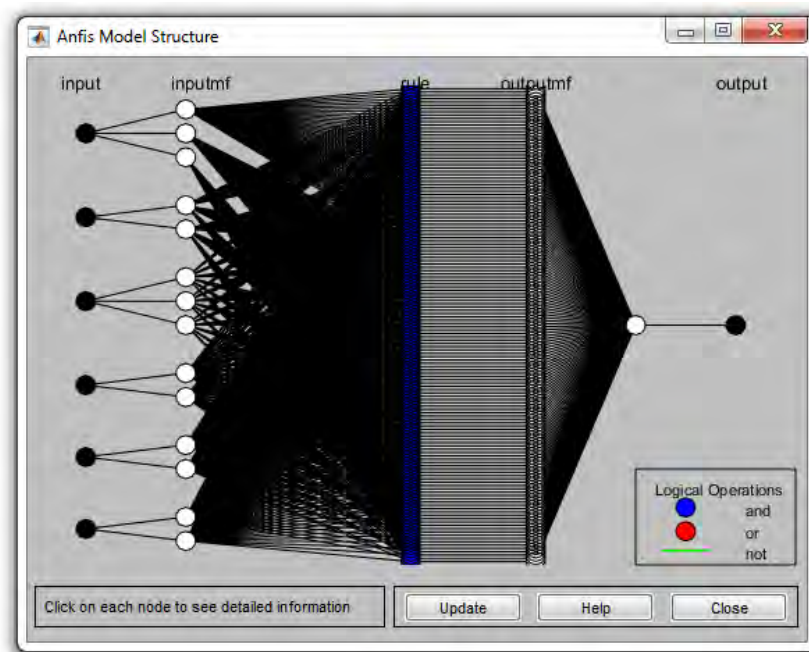


Figure 4-13: Neural Network Representation of the FIS

In Figure 4-13, the artificial neural network automatically produced by the ‘Neuro-Fuzzy Designer’ toolbox of MATLAB has been shown, which represents the basic characteristics of the Neuro-Fuzzy models developed in this study. The characteristics of each layer have been described as follows:

- i. The first layer is called 'input', which contains six nodes. Each node of the layer represents one of the inputs to the system, which are crisp in nature.
- ii. The second layer is called ‘inputmf’, which contains fourteen nodes. The nodes are arranged in groups such that each group is connected to one particular input and each node of the group produces the fuzzified value (degree of membership) of that particular input for one of the linguistic categories assigned in it.
- iii. The third layer is called ‘rule’, which contains one hundred and forty four nodes. Each node of the layer represents one of the fuzzy rules connected to one of the combinations of fuzzified linguistic terms of each input. The blue color of each node refers that fuzzified inputs are connected by the logical operator 'AND' meaning the node produces the degree of membership for that particular rule, which was chosen to be the minimum of the constituent degree of membership values for each of the inputs.
- iv. The fourth layer is called ‘outputmf’, which contains one hundred and forty four nodes. Each node of the layer represents one of the output membership functions. The input of each node is the degree of membership of the corresponding rule, which is multiplied with the location of the singleton to produce the output of the node.
- v. In the fifth layer, there is only one node, the function of which is to produce an aggregated value for all the node outputs of the fourth layer.
- vi. The sixth and last layer is called 'output', which produces the output of the model and is nothing but the same quantity produced by the fifth layer.

#### **4.3.5 Model Calibration**

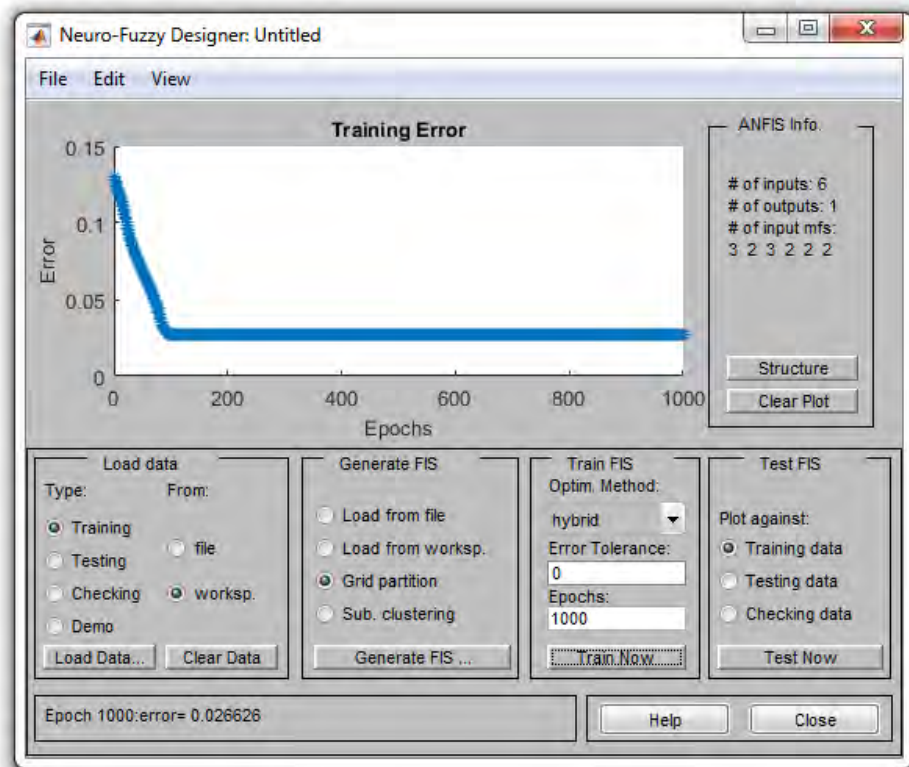
After the development of the basic architecture of the Fuzzy Inference System (FIS), the most important task is to calibrate it by using artificial neural network to produce an Adaptive Neuro-Fuzzy Inference System (ANFIS). Calibration of a ANFIS means the alteration of the shape and positions of input-output membership functions by using a suitable algorithm of artificial neural network in such a way that the input-output relationships replicate the field observed input-output relationships as



accurately as possible. However, it is practically impossible to calibrate the models to perfectly fit all the input-output relationships due to the presence of inherent noise in the data set. As shown earlier, the system can be presented in the form of an artificial neural network, which is structured such that the parameters of the membership functions of the fuzzy sets and the logical operations of the inference system are exactly replicated in the weights and the operations of the neural network. Modification of the weights of the artificial neural network results in changes in the shape of each membership function.

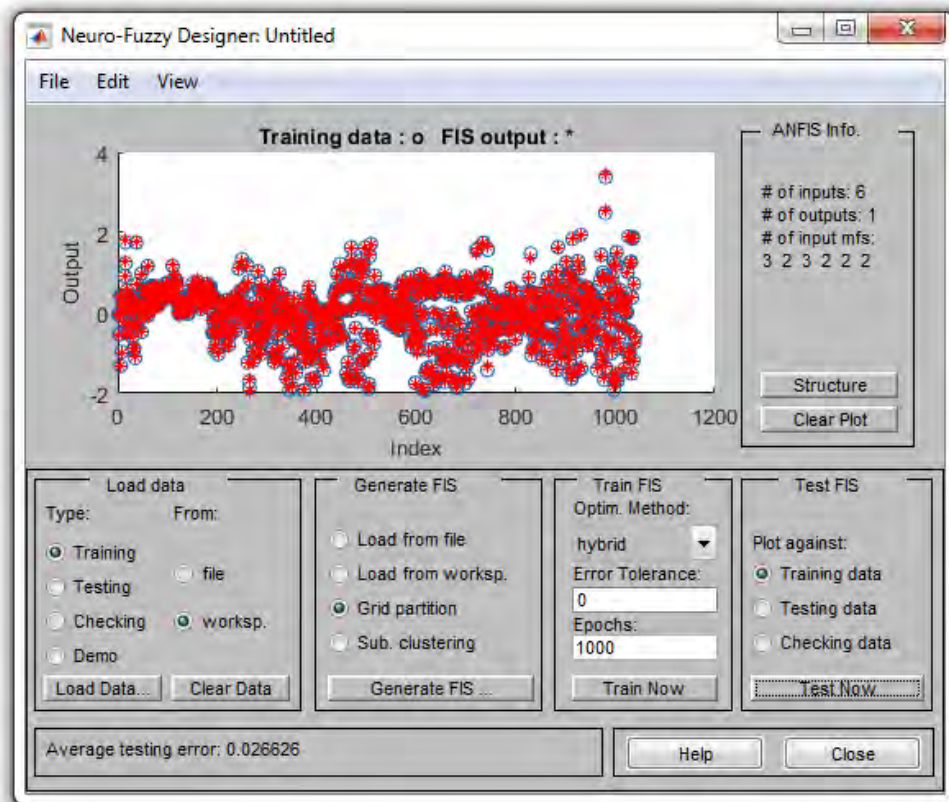
As mentioned earlier, all the input membership function of this study are mathematically expressed as “Gaussian” function, the key parameters to adjust here are  $\sigma$  and  $c$  which means standard deviation and mean accordingly. In case of output membership functions, all of which are singletons, the only parameter to adjust is the placement of the singletons on the horizontal axis. However, the fuzzy rule base was fixed before the phase of model calibration; therefore it has nothing to be calibrated. Fortunately, the ‘Neuro-Fuzzy Designer’ toolbox of MATLAB can automatically calibrate the membership functions provided the data set is loaded properly and some calibration parameters are specified.

First step to calibrate the model is to choose the best method to train the model. The ANFIS editor of the ‘Neuro-Fuzzy Designer’ toolbox of MATLAB provides two different optimization methods for model calibration. The backpropagation method uses a steepest descent technique, in order to update the parameters associated with the shape and position of both input and output membership functions for model calibration. The hybrid method uses a combination of backpropagation (steepest descent) method and least square method to update the membership function parameters for model calibration. The shape and position of the input membership functions are adjusted by using the backpropagation method, whereas the position of the singleton type output membership functions are adjusted by using the least square method. After testing with a random set of data it was observed that the hybrid method offers lesser error than the backpropagation method. Even though the backpropagation method shows a smooth error transition over each epoch, the calibration results in the backpropagation method show unrealistically large amount of scatter regarding the position of the output singletons, which is much larger than those in the hybrid method. These relative advantages of the hybrid method over the backpropagation one justify the preference of the former over the latter.



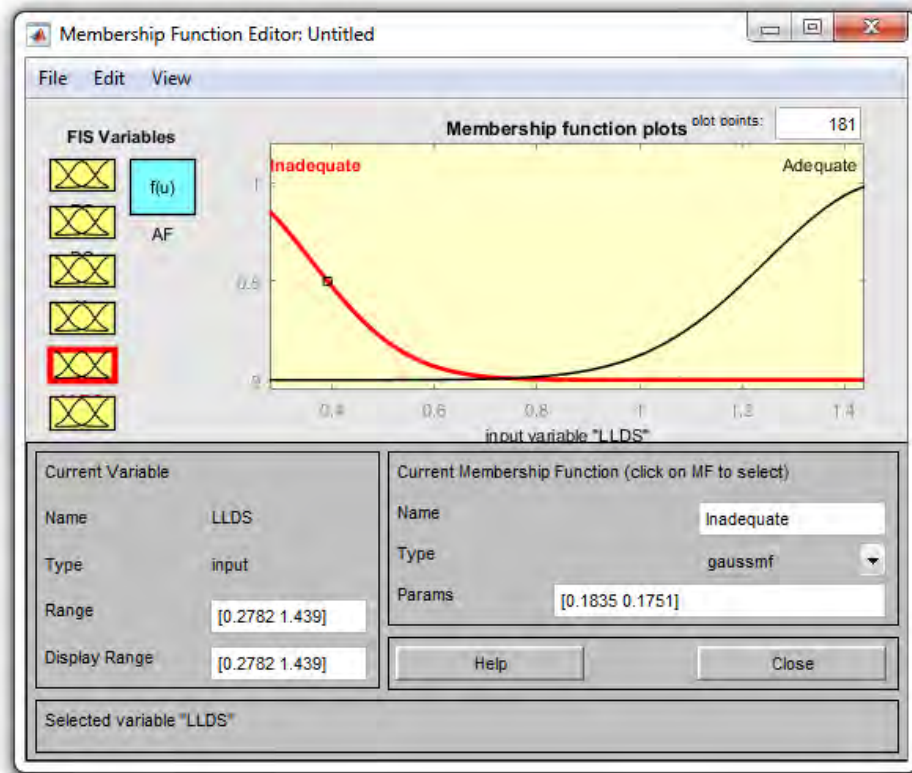
**Figure 4-14:** Plot of Training Error against Epoch Number

In addition to this option, the ANFIS editor also allows to specify the error tolerance and the number of epochs. The error tolerance is the value of error, the attainment of one below which during any epoch of the calibration will automatically end the calibration process at that point. It was chosen to be zero, which is the default value of the ANFIS editor. The effect of the input is passed forward through the network, then the error between targets and predicted output is known as an epoch. The number of epochs was chosen as 1000 based on the expectation that it will ensure model calibration to its ultimate potential by minimizing the training error as much as possible. Figure 4-14 shows the plot of training error against the epoch number during the calibration of one of the models. The epoch for the training model is 0.026626. it should be mentioned that, 1000 epoch was taken arbitrarily. The line became straight near 150 epochs. Therefore, lesser epoch could have been taken which would require lesser time.



**Figure 4-15:** Plot of Model Outputs and Data set Outputs against Data Set Index

Once the models are calibrated, the ANFIS editor is capable to show the response of the model for the training data sets with the click on the 'Test Now' button, while the option 'Training Data' is active. Figure 4-15 shows the plot of actual outputs of the training data points (in blue) as well as the model outputs for the same data points (in red) plotted against the data index for one of the models. The more the red points cover the blue circled points the model is more efficient to work with real field data. However, it is impossible to fit all the red points with the blue points due to the presence of noise in the data. A visual observation of the plot shows satisfactory agreement between the outputs of the training data set (as observed in field) and those of the model in most of the cases.



**Figure 4-16:** Membership Function after Calibration (input: LLDS)

It has previously been mentioned that the key parameters to be adjusted in each of the input membership functions of each model during model calibration are designated as  $\sigma$  and  $c$ , which govern the geometric shapes and positions of the input membership functions. Figure 4-16 shows the change in the shape of the input membership function (LLDS) after calibration. However, unlike the input membership functions, all the output membership functions were singletons having just a degree of membership value of unity at a particular point along the horizontal axis. Therefore, the only parameter to be calibrated in case of output membership functions was solely the placement of each of the singletons along the horizontal axis. In Table 4-6 and Table 4-7, the post-calibration parameters pertaining to the input and output membership functions respectively have been presented for the proposed model.

**Table 4-6:** Post-calibration Parameters Pertaining to the Input Membership Functions

Input Name	Input Designation	Category Names	Membership Function Parameters	
			$\sigma$	$c$
Relative Speed	RS	Negative	1.6	-2.619
		Nothing	1.592	1.204
		Positive	1.692	4.985
Distance	DS	Inadequate	1.625	0.1987

Input Name	Input Designation	Category Names	Membership Function Parameters	
			$\sigma$	c
		Adequate	1.699	4.014
Acceleration of Leader	AL	Dec	1.329	-1.978
		Zero	1.349	0.8882
		Acc	1.427	3.822
Speed of Follower	SF	Slow	0.2418	0.7076
		Fast	0.1446	1.395
Left Lateral Distance	LLDS	Inadequate	0.1835	0.1751
		Adequate	0.2395	1.489
Right Lateral Distance	RLDS	Inadequate	0.369	-0.08761
		Adequate	0.7643	1.565

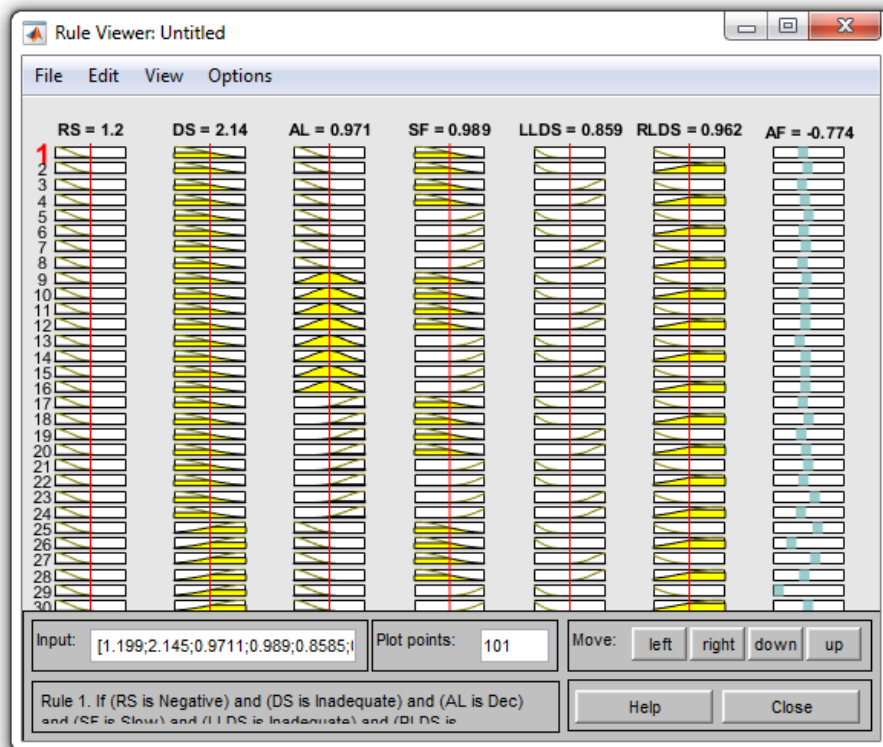
**Table 4-7:** Post-calibration Parameters Pertaining to the Output Membership Functions

Membership Function	Position	Membership Function	Position	Membership Function	Position
mf1	-16.07	mf49	-0.8736	mf97	-22.9
mf2	25.29	mf50	-8.608	mf98	43.62
mf3	-26.38	mf51	1.299	mf99	68.2
mf4	-0.2021	mf52	-3.443	mf100	36.16
mf5	33.7	mf53	-18.15	mf101	-59.52
mf6	-10.54	mf54	0.8408	mf102	-22.2
mf7	8.594	mf55	-4.413	mf103	-18.72
mf8	-20.6	mf56	5.513	mf104	92.32
mf	14.24	mf57	-3.597	mf105	138.6
mf10	-11.43	mf58	4.244	mf106	-52.07
mf11	5.631	mf59	-4.122	mf107	67.72
mf12	3.199	mf60	2.295	mf108	-35.73
mf13	-47.4	mf61	16.9	mf109	6.054
mf14	7.103	mf62	-1.761	mf110	20.13
mf15	2.84	mf63	-0.5568	mf111	26.97
mf16	3.487	mf64	-1.5	mf112	14.14
mf17	-18.3	mf65	-22.11	mf113	95.11
mf18	32.52	mf66	0.6355	mf114	304.3
mf19	-33.02	mf67	-25.21	mf115	7.19

Contd.... Table 4-7

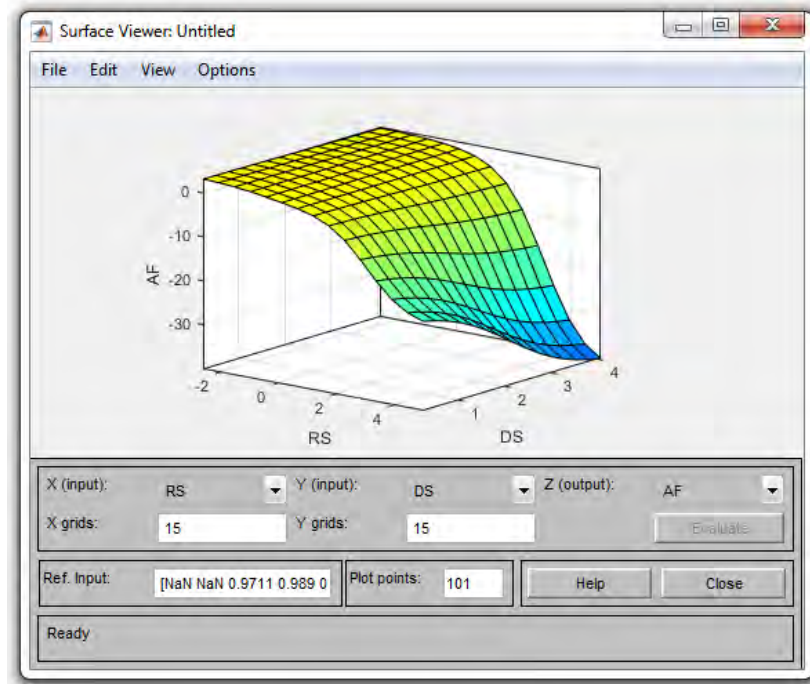
Membership Function	Position	Membership Function	Position	Mem Function	Position
mf20	11.15	mf68	-6.018	mf116	35.59
mf21	29.01	mf69	-7.603	mf117	5.772
mf22	-12.67	mf70	13.45	mf118	10.54
mf23	88.19	mf71	1.277	mf119	-38.59
mf24	-35.8	mf72	15.73	mf120	-11.54
mf25	114.8	mf73	-14.3	mf121	-63.41
mf26	-122	mf74	25.3	mf122	-124.3
mf27	92.91	mf75	-29.17	mf123	3.069
mf28	-5.468	mf76	7.613	mf124	-53.67
mf29	-235.6	mf77	116.1	mf125	12.35
mf30	28.39	mf78	-13.84	mf126	85.31
mf31	51.66	mf79	-24.78	mf127	-25.92
mf32	-24.24	mf80	8.143	mf128	-26.55
mf33	-13.7	mf81	-13.25	mf129	24.02
mf34	30.12	mf82	-2.662	mf130	40.26
mf35	-9.422	mf83	9.352	mf131	19.26
mf36	5.035	mf84	-3.923	mf132	-52.83
mf37	170.3	mf85	-61.32	mf133	-32.59
mf38	-35.13	mf86	14.61	mf134	-114.5
mf39	-8.089	mf87	4.143	mf135	-1.438
mf40	-16.96	mf88	8.514	mf136	-116.9
mf41	70.75	mf89	168.2	mf137	30.35
mf42	-30.54	mf90	-68	mf138	99.38
mf43	-22.88	mf91	-51.31	mf139	1.171
mf44	-3.522	mf92	43.89	mf140	12.83
mf45	158.3	mf93	-68.45	mf141	-7.199
mf46	71.53	mf94	-38.08	mf142	-15.98
mf47	-159.3	mf95	96.16	mf143	-6.819
mf48	12.02	mf96	1.217	mf144	-12.77

After the training, for any value of any input parameters, an output can be generated which can be observed in Figure 4-17. The shapes of all the inputs of 144 rules are presented in the figure. By moving the red bar of each input parameters, a unique output can be generated.



**Figure 4-17: Rule Viewer after Calibration**

Once the models are calibrated by adjusting the shapes and positions of membership functions, it is possible to visualize the output patterns in the surface viewer of the 'Neuro-Fuzzy Designer' toolbox of MATLAB in the form of a three dimensional and colorful surface which represents the output in the vertical axis and two inputs in two horizontal axes. However, because the number of inputs in this study is six, it is necessary to select two of the input variables as redundant variables by assigning them certain values at the input vector of the surface viewer. On the other hand, the remaining four variables being represented in two horizontal axes of the surface viewer were designated as 'Not a Number' (NaN). An example of the use of surface viewer for one of the models is shown in Figure 4-18, where the output surface is plotted against the input variables RS (Relative Speed) and DS (Distance), while keeping the remaining two input variables AL (Acceleration of Leader), SF (Speed of Follower), Left Lateral Distance (LLDS) and Right Lateral Distance (RLDS) in some constant values (For example, AL= 0.9711 meter square per second; SF= 0.989 meter per second). In a same way, it is possible to produce an infinite number of output surfaces for any of the models by using various combinations of input variables at the horizontal axes, while assigning various values for the remaining redundant input variables at the input vector.



**Figure 4-18:** Surface Viewer Showing the Output Surface

#### 4.4 Summary

In this chapter, the details of the analysis of raw data have been presented on a step by step basis. It involved details of image processing and proposed technique to analyze the raw data. It was followed by a detailed description of the process to convert the coordinates on the video footage into real world positions, which led to the calculation of various car following parameters including length, speed and acceleration.

This chapter also elaborately explained various issues pertaining to model development, which included the development of the basic architecture of the Fuzzy Inference System (FIS) and other relevant topics such as membership functions, fuzzy rules etc. Finally, the process of model calibration has been detailed by using the ANFIS editor of the 'Neuro-Fuzzy Designer' toolbox of MAT LAB.



## CHAPTER 5 VALIDATION

### 5.1 Introduction

Once the Neuro-Fuzzy model has been developed, which depict car following behavior under heterogeneous road traffic condition, it is necessary to validate them. In this study, the purpose of model validation is to assess the performance of the model and to ensure that the incorporation of the model outputs will produce acceptably realistic and consistent results replicating the car following behavior of real world. In this chapter, to examine the performance of the developed models, various criteria were used to calculate errors. At first, the definitions of all the criteria were provided. Then model performance was observed using the defined criteria. Later two models were developed and the performance of the new models was compared with the Neuro-Fuzzy model.

### 5.2 Validation Methodology

There are a number of ways to validate car following models. The simplest and most common way of model validation involves conventional methods such as checking the goodness of fit of the model results based on field data (external to the one used for model calibration) from the same site or from a similar site through statistical or econometric measures of effectiveness (Chen et al., Khodayari et al., Ghaffari et al., kamali et al.). Despite its simplicity, this method suffers from lack of credibility due to its being unable to predict the long term stability of the models temporally and spatially.

One of the better ways to validate car following models is by incorporating the model results into professional microscopic simulation packages such as PARAMICS, VISSIM, AIMSUN, CARSIM etc. and recalibrating their car following parameters for the condition prevailing in this study, which is quite different from the one for which the packages were originally calibrated, in order to observe their performance in the simulation framework and to compare it with field observations (Rao and Owen, 2000; Wu et al., 2003). However, the aforementioned professional microscopic

simulation packages are not only expensive, but also require significant effort for the coding of their operation. Also, because these packages are comprehensive in nature and not limited to the modeling of car following behavior only, it would not be wise to attempt this approach without readjusting traffic operation parameters other than car following (e.g. lane changing, overtaking etc) for heterogeneous road traffic condition. Therefore, considering the breadth of the scope of this study, it was not deemed feasible to adopt this approach.

The other way to validate the model is to develop a simple and easy to understand simulation model and to check the stability of the car following models by using it. Because the car following models have been developed by using the 'Neuro-Fuzzy Designer' toolbox of MATLAB, it is an ideal case and offers a golden opportunity for the development of a simulation model using SIMULINK (which is a simulation platform incorporated with MATLAB), in order to check the stability of the car following models (Imran, 2006). The limitation of SIMULINK is that, the parameters for lateral movement cannot be addressed by SIMULINK. Therefore, SIMULINK was not used to validate the model.

Due to the aforementioned limitation of SIMULINK and professional microscopic simulation packages, it was decided to use conventional method of finding error of the model by defining the criteria and compare the model performance with other similar model.

### **5.3 Validation of the Proposed Model**

The validation of the proposed ANFIS model is done in three steps. At first, few criteria had been defined for statistical analysis. Secondly, the proposed model had been checked for error using different criteria already defined in the previous step. Thirdly, two new models had been introduced in the later section and finally, performance of the models had been checked against the proposed ANFIS model using the same defined criteria.

#### **5.3.1 Definition**

Few criteria had been defined that were used later to check the performance of the proposed model and the new two models.

### 5.3.1.1 Mean absolute percentage error (MAPE)

The mean absolute percentage error (MAPE), also known as mean absolute percentage deviation (MAPD) is a statistical measure of how accurate a forecast system is. It measures this accuracy as a percentage, and can be calculated as the average absolute percent error for each time period minus actual values divided by actual values. Where A is the actual value and F is the forecast value, this is given by:

$$MAPE = \frac{100}{n} \sum_{i=0}^n \left| \frac{A-F}{A} \right| \quad (5.1)$$

The mean absolute percentage error (MAPE) is the most common measure used to forecast error, and works best if there are no extremes to the data (and no zeros).

Although the concept of MAPE sounds very simple and convincing, it has major drawbacks in practical application, and there are many studies on shortcomings and misleading results from MAPE.

- i. It cannot be used if there are zero values (which sometimes happen for example in demand data) because there would be a division by zero.
- ii. For forecasts which are too low the percentage error cannot exceed 100%, but for forecasts which are too high there is no upper limit to the percentage error.
- iii. MAPE puts a heavier penalty on negative errors,  $A < F$  than on positive errors. As a consequence, when MAPE is used to compare the accuracy of prediction methods it is biased in that it will systematically select a method whose forecasts are too low.

To overcome these issues with MAPE, one other measure is proposed here.

### 5.3.1.2 Symmetric Mean Absolute Percentage Error (SMAPE)

Symmetric mean absolute percentage error (SMAPE or sMAPE) is an accuracy measure based on percentage (or relative) errors. It is usually defined as follows:

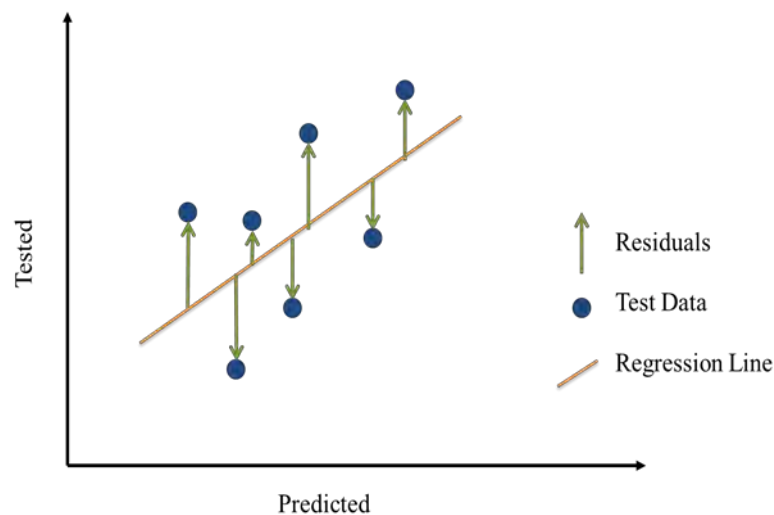
$$SMAPE = \frac{100\%}{n} \sum_{i=1}^n \frac{|F - A|}{\frac{(|A| + |F|)}{2}} \quad (5.2)$$

### 5.3.1.3 Root Mean Square Error (RMSE)

The root-mean-square error (RMSE) or root-mean-square deviation (RMSD) is a frequently used measure of the differences between values (sample or population

values) predicted by a model or an estimator and the values observed. In other words, Root Mean Square Error (RMSE) is the standard deviation of the residuals (prediction errors). Residuals are a measure of how far from the regression line data points are; RMSE is a measure of how spread out these residuals are. In other words, it tells you how concentrated the data is around the line of best fit. Root mean square error is commonly used in climatology, forecasting, and regression analysis to verify experimental results.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=0}^n (A - F)^2} \quad (5.3)$$



**Figure 5-1:** Fitness of Data against Regression line

When standardized observations and forecasts are used as RMSE inputs, there is a direct relationship with the correlation coefficient. For example, if the correlation coefficient is 1, the RMSE will be 0, because all of the points lie on the regression line (and therefore there are no errors). Figure 5-1 demonstrates the fitness of data against regression line.

#### 5.3.1.4 R-squared ( $R^2$ )

R-Squared ( $R^2$  or the coefficient of determination) is a statistical measure in a regression model that determines the proportion of variance in the dependent variable that can be explained by the independent variable. In other words, r-squared tells how well the data fit the regression model (the goodness of fit).

$$R^2 = \frac{[\sum_{i=0}^n (A - A_{mean})(F - F_{mean})]^2}{[\sum_{i=0}^n (A - A_{mean})^2 \sum_{i=0}^n (F - F_{mean})^2]} \quad (5.4)$$

Here,  $A_{mean}$  and  $F_{mean}$  define the mean values of A and P respectively. R-squared can take any value between 0 to 1. Although the statistical measure provides some useful insights regarding the regression model, the user should not rely only on the measure in the assessment of a statistical model. The figure does not disclose information about the causation relationship between the independent and dependent variables.

In addition, it does not indicate the correctness of the regression model. Therefore, the user should always draw conclusions about the model by analyzing r-squared together with the other variables in a statistical model.

### 5.3.1.5 Mean Absolute Deviation (MAD)

The MAD (Mean Absolute Deviation) measures the size of the error in units. It is calculated using the following formula.

$$MAD = \frac{1}{n} \sum_{i=0}^n |A - F| \quad (5.5)$$

The MAD is a good statistic to use when analyzing the error for a single item. However, if MADs are aggregated over multiple items, care should be taken about high-volume products dominating the results.

### 5.3.1.6 Mean Squared Error (MSE)

In statistics, the mean squared error (MSE) or mean squared deviation (MSD) of an estimator (of a procedure for estimating an unobserved quantity) measures the average of the squares of the errors—that is, the average squared difference between the estimated values and what is estimated.

$$MSE = \frac{1}{n} \sum_{i=0}^n (A - F)^2 \quad (5.6)$$

The mean squared error tells how close a regression line is to a set of points. It does this by taking the distances from the points to the regression line (these distances are the “errors”) and squaring them. The squaring is necessary to remove any negative

signs. It also gives more weight to larger differences. It's called the mean squared error as you're finding the average of a set of errors.

### 5.3.1.7 Standard Deviation Error (SDE)

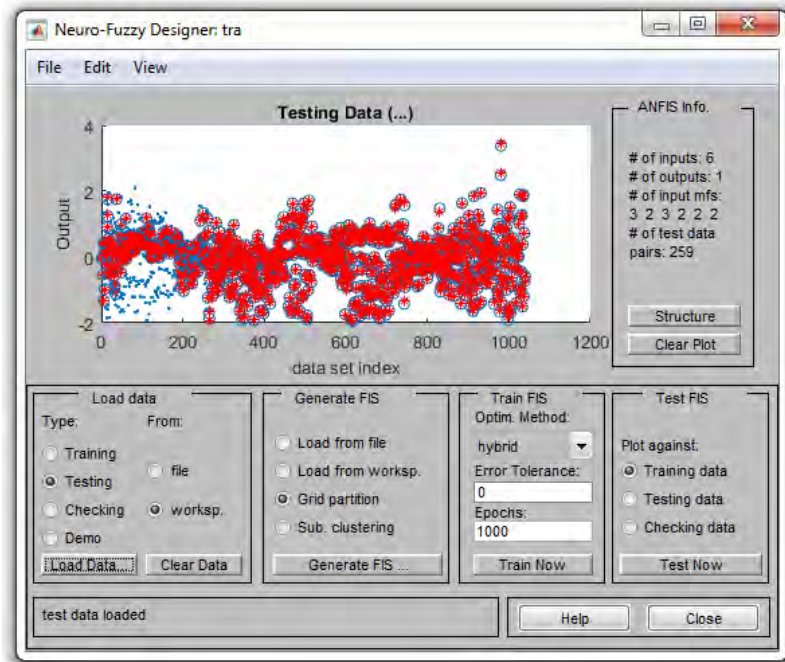
Standard deviation measures the amount of variability or dispersion of data from the mean. Standard deviation error (SDE) measures the error for disparity of data from the error. The more the data are apart, the more the error is. The formula for standard deviation error (SDE) is:

$$SDE = \sqrt{\frac{1}{n} \sum_{i=0}^n \left( \frac{|A-F|}{A} - \frac{MAPE}{100} \right)^2} \quad (5.7)$$

### 5.3.2 Statistical Analysis of Proposed Model

It is of utmost importance to validate a model after being proposed. The essence of validating a model is to see the proposed model works well if new real world data is introduced. As discussed in the previous chapter, the whole dataset was divided into two subsets. 80% of the whole dataset were used for training and testing. 20% of the remaining subset was left for validating the model. As 1300 data were used, 260 data remained for the validation purpose.

The validation dataset had been imported from the MATLAB workspace. The dataset being loaded in the trained model is demonstrated in the following figure.



**Figure 5-2:** The validation dataset being loaded in the trained model

Figure 5-2 demonstrates the red circles are the trained model and the blue spot on the left side of the picture are the validation dataset.

After the validation dataset being loaded in the model, the model would generate the output of the dataset based the rules that had been set by the training. Then the outputs were exported in the workspace and extracted using the function evalfis.

The 260 model output and the corresponding actual output were then further used to observe the errors of the model generated output where the errors have been defined earlier of the chapter. A sample is provided on how the errors were calculated in the following Table 5-1.

**Table 5-1:** Statistical Analysis of Proposed Model

Actual (A)	Forecasted (F)	(A-F)	$c/a$	$c^2$	$(A-\bar{A})$	$c*f$	$f^2$	$d-\frac{MAPE}{100}$	$i^2$
a	b	c	d	e	f	g	h	i	j
0.2569	0.269	-0.0128	0.049	0.00016	0.3298	-0.0042	0.108	0.021	0.0005
1.2851	1.292	-0.0069	0.005	4.8E-05	1.357	-0.0094	1.844	-0.022	0.0005
-0.6363	-0.6438	0.0074	-0.012	5.5E-05	-0.5636	-0.0042	0.318	-0.039	0.0016
-0.032	-0.03388	0.00187	-0.059	3.5E-06	0.0408	7.6E-05	0.002	-0.086	0.0076
-0.5619	-0.57277	0.010862	-0.019	0.0001	-0.489	-0.0053	0.239	-0.047	0.0023
0.2911	0.30197	-0.0108	0.037	0.0001	0.363	-0.0039	0.132	0.008	0.0073
-0.1103	-0.12098	0.01060	-0.096	0.00012	-0.0376	-0.0004	0.001	-0.124	0.0154
-0.8606	-0.87013	0.00943	-0.011	8.8E-05	-0.7879	-0.0074	0.621	-0.039	0.0015

Actual (A)	Forecasted (F)	(A-F)	c/a	c <sup>2</sup>	(A- $\bar{A}$ )	c*f	f <sup>2</sup>	d- $\frac{MAPE}{100}$	i <sup>2</sup>
a	b	c	d	e	f	g	h	i	j
1.3536	1.362402	-0.0088	0.006	7.7E-05	1.426	-0.0125	2.035	-0.022	0.0004
-1.5267	-1.53489	0.00816	-0.005	6.6E-05	-1.4539	-0.0118	2.114	-0.033	0.0011
0.7626	0.774244	-0.0116	0.015	0.00014	0.8353	-0.0097	0.698	-0.012	0.0002
1.7123	1.726925	-0.0146	0.009	0.00021	1.7851	-0.0260	3.187	-0.019	0.0004
-1.8497	-1.85795	0.00825	-0.004	6.8E-05	-1.777	-0.0146	3.158	-0.032	0.0011
1.0541	1.067101	-0.0129	0.012	0.00017	1.1267	-0.0146	1.270	-0.015	0.0003
-1.5837	-1.59488	0.01114	-0.007	0.00012	-1.511	-0.0168	2.283	-0.035	0.0012

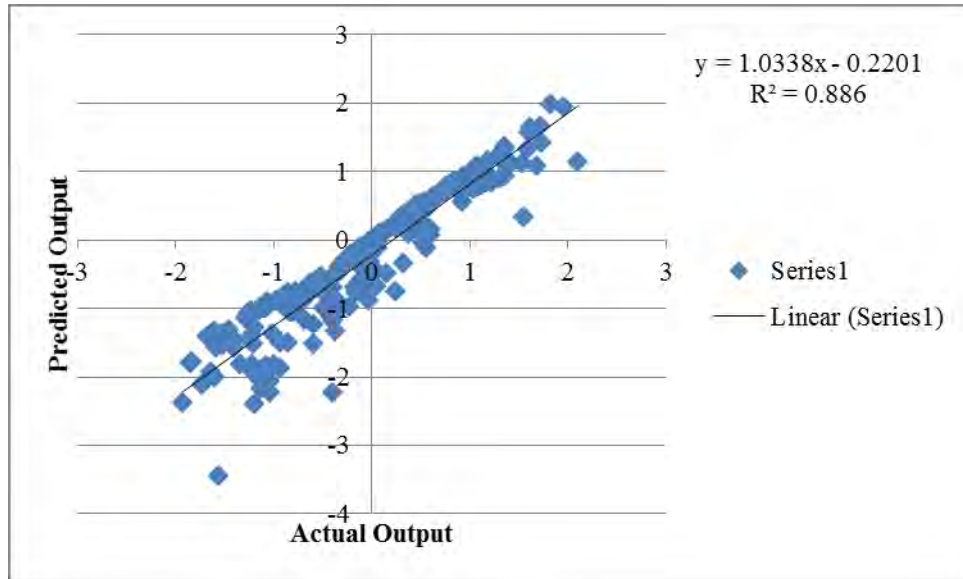
Table 5-1 shows a sample of analysis which was used to determine the error and deviation of the model from standard. The result of the analysis is provided below.

Results of analysis

MAPE	2.826947328
RMSE	0.013559199
R <sup>2</sup>	0.999852541
SDE	0.250551392
MAD	0.009147882
MSE	0.000183852
SMAPE	0.004701466

Root mean squared error (RMSE) was found for both training and validation dataset. RMSE for training data was 0.026626 where ad for validation set RMSE was 0.01356 as can be seen from the Table5-1. RMSE for validation dataset are less than the trained dataset.





**Figure 5-3:** Fitness of the proposed ANFIS model

Figure 5-3 shows the overall fitness of the Adaptive Neuro-Fuzzy Inference System (ANFIS), showing that the predicted outputs were plotted against the actual ones. The outputs lie around a straight line passing through the origin, which implies a very close agreement between the two.

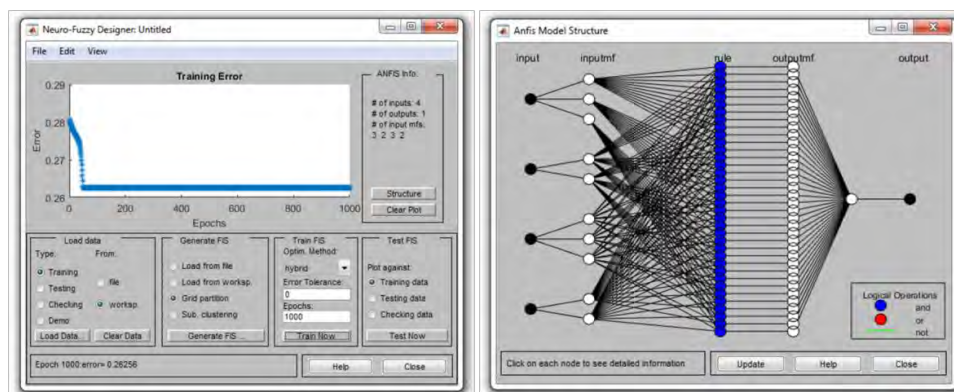
### 5.3.3 Introduction of New Models

The third step of the validation is to introduce two new model. The first model is an ANFIS model with two less input parameters than the proposed model. The second model is a neural network generated in MATLAB. The description of the model for the validation is provided below.

#### 5.3.3.1 A stimulus response model

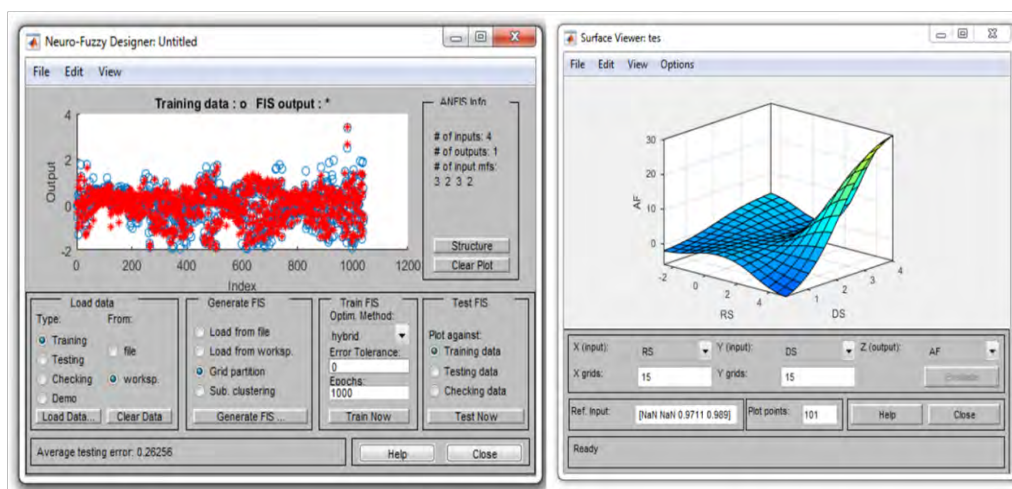
A new ANFIS model had been prepared for validation of the proposed model using the same parameter described in the study of Imran (2009) or described in Stimulus-response model in Chapter 2. The input parameters are relative speed, distance, acceleration of leader and speed of follower all of which have been described in Chapter 4. The output parameter is the same as the proposed model, acceleration of follower. At first, all the input and output parameters were imported from workspace in the “Neuro-Fuzzy Designer” toolbox of MATLAB. Here, it is important to mention that same set of data was used as in the proposed model. The membership function

had been kept the same as the proposed model, Gaussian membership function. The number of membership functions for each input parameter had also been kept the same as the proposed model depending on the input parameter being considered. Training had been done through 1000 epoch and by using hybrid method for error calculation and adjustment of the parameters of the membership functions.



**Figure 5-4:** (on left) Training Error of the Stimulus-response Model & (on right) ANFIS model structure for that model.

As from Figure 5-4, it can be seen that the Root Mean Square Error (RMSE) for trained new model is 0.26256 which is higher than the RMSE error found in the trained proposed model. Figure 5.4 also shows the ANFIS model structure of the new model which does not seem as complicated as the proposed model one. As from Figure 5-4, it can be seen that 36 ( $3*2*3*2$ ) number of unique fuzzy rules had been used where each of the four input variables were connected with the logical operator ‘AND’ (chosen the arithmetic minimum) ensuring that each of the linguistic terms of each input are connected with those of all other inputs in every possible way.



**Figure 5-5:** (on left) Plot of Trained Model Output against Data Set Index & (on right) Surface Viewer Showing the Output Surface

Once the model was calibrated, the response of the model for the training data sets had been observed in the ANFIS editor. On the left side of Figure 5-5 shows the plot of actual outputs of the training data points (in blue) as well as the model outputs for the same data points (in red) plotted against the data index for one of the models. As mentioned in Chapter 4, after the calibration of the model, the response of the model against different input parameters can be seen in the surface viewer. The right portion of Figure 5-5 shows the smooth surface of AF against RS (0.9711 meter per second) and DS (0.989 meter) whereas the other two inputs are designated as NAN. The summary of the model is given below.

Summary of ANFIS info

Number of nodes : 99  
 Number of linear parameters : 36  
 Number of nonlinear parameters : 20  
 Total number of parameters : 56  
 Number of training data pairs : 1040  
 Number of fuzzy rules : 36

Same as the proposed model, 260 model output and the corresponding output were used to perform statistical analysis in order to observe the performance of the new model. The criteria used for statistical analysis has been defined earlier in this chapter. A sample is provided about the steps to the statistical analysis are provided in Table 5-2.

**Table 5-2:** Statistical Analysis of Stimulus-response Model

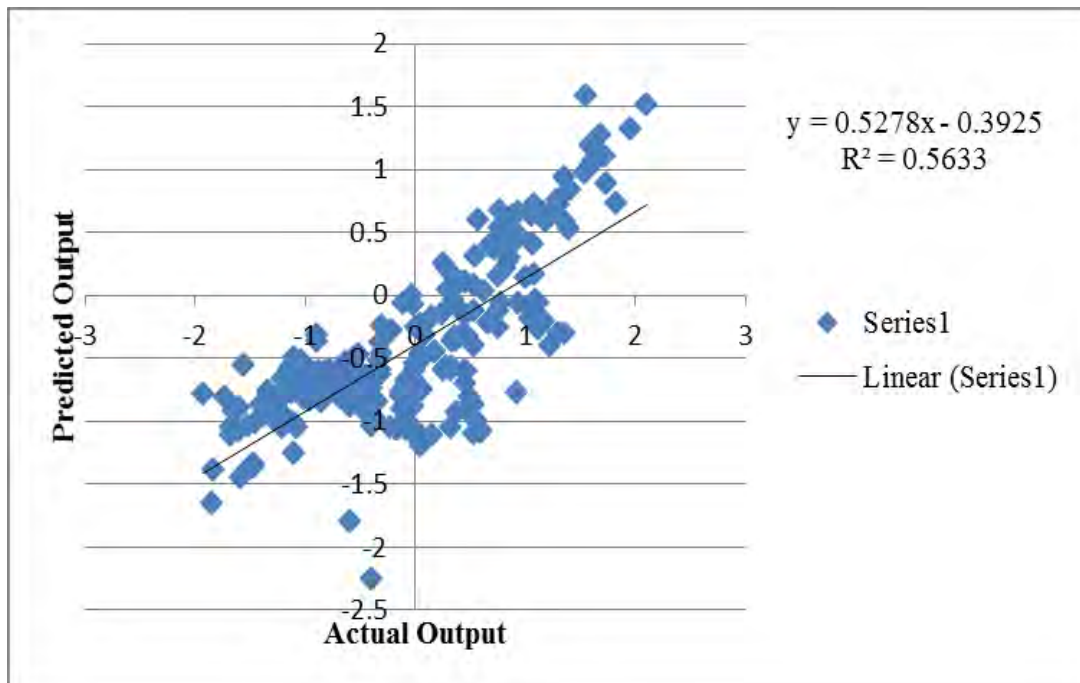
Actual(A)	Forecasted(F)	A-F	c/a	c <sup>2</sup>	(A-A <sub>mean</sub> )	c*f	f <sup>2</sup>	d- $\frac{MAPE}{100}$	i <sup>2</sup>
a	b	c	d	e	f	g	h	i	j
0.257	0.2576	-0.0007	0.0026	4E-07	0.3298	0.0002	0.1087	8.2516	68.089
1.2852	0.7609	0.5243	0.408	0.2749	1.358	0.712	1.844	8.657	74.9437
-0.636	-0.5246	-0.1117	-0.176	0.0125	-0.5636	0.0629	0.3176	8.0735	65.1814
-0.032	0.0016	-0.0336	-1.049	0.0011	0.0408	0.0014	0.0017	7.2001	51.8419
-0.562	-0.4956	-0.0663	-0.118	0.0044	-0.4891	0.0324	0.2393	8.1311	66.1148
0.2911	0.2016	0.0895	0.3074	0.008	0.3639	0.0326	0.1324	8.5565	73.2131
-0.11	-0.0525	-0.0579	-0.525	0.0034	-0.0376	0.0022	0.0014	7.7244	59.6664
-0.861	-0.8055	-0.0552	-0.064	0.003	-0.7879	0.0435	0.6208	8.1849	66.9922
1.3536	0.9461	0.4074	0.301	0.166	1.4263	0.5812	2.0345	8.55	73.1033

Actual(A)	Forecasted(F)	A-F	c/a	c <sup>2</sup>	(A-A <sub>mean</sub> )	c*f	f <sup>2</sup>	d- $\frac{MAPE}{100}$	i <sup>2</sup>
a	b	c	d	e	f	g	h	i	j
-1.527	-1.3822	-0.1445	-0.095	0.0209	-1.454	0.2101	2.114	8.1544	66.4939
0.7626	0.5466	0.216	0.2832	0.0466	0.8353	0.1804	0.6978	8.5323	72.7993
1.7123	1.1087	0.6036	0.3525	0.3643	1.7851	1.0775	3.1866	8.6015	73.9864
-1.85	-1.6415	-0.2082	-0.113	0.0434	-1.7769	0.37	3.1575	8.1365	66.2021
1.0541	0.6742	0.3799	0.3604	0.1443	1.1269	0.4281	1.2699	8.6094	74.1225
-1.584	-1.4443	-0.1394	-0.088	0.0194	-1.511	0.2107	2.283	8.161	66.6016

Table 5-2 shows a sample of steps that is required for a statistical analysis. The result of the analysis is provided below.

Result of analysis

MAPE	824.903
RMSE	0.693437
R <sup>2</sup>	0.563326
SDE	112.0451
MAD	0.556854
MSE	0.480855
SMAPE	0.301172



**Figure 5-6:** Fitness of new ANFIS model

Figure 5-6 shows the overall fitness of the Stimulus-response Model, showing that the predicted outputs were plotted against the actual ones. The outputs lie around a

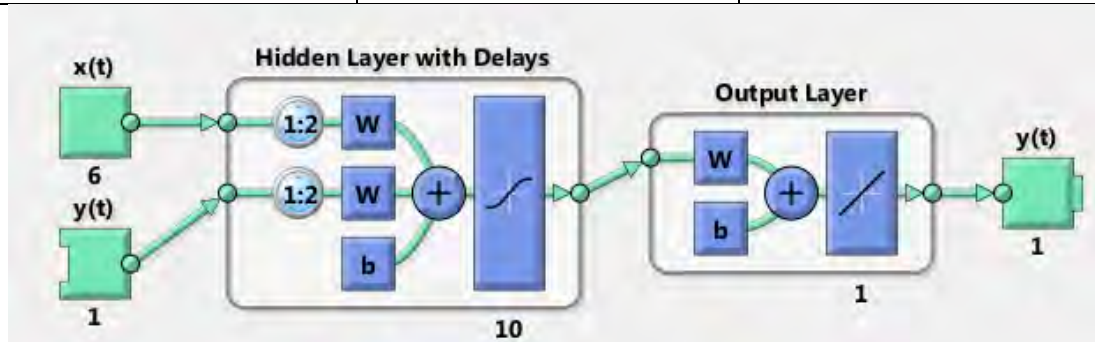
straight line passing through the origin, which implies a very close agreement between the two but not as close as the proposed model.

### 5.3.3.2 A neural network model

As part of the validation, a Neural Network model was created using the simple platform provided by MATLAB. Neural Network toolbox of MATLAB provides five tools to make models. The tools are Input-Output and curve-fitting, Pattern recognition and classification, Clustering and Dynamic Time series. Among all the five tools, Dynamic Time series only helps create a predictive model. Prediction is a dynamic filtering, in which past values of one or more time series are used to predict future values. Dynamic neural networks, which include tapped delay lines are used for nonlinear filtering and prediction. Predictive models are also used for system identification or dynamic modeling in which dynamic models are building for physical systems. Car following model is a dynamic model. In order to make the model, at first 1300 input and output data had been imported from directory in separate files. The data were automatically separated in three dataset. Table 5-3 gives the details into different dataset.

**Table 5-3:** Details of Different Types of Dataset

Training Dataset	Validation Dataset	Testing Dataset
70% of the data	15% of the data	15% of the data
910 data	195 data	195 data
These are presented to the network during training, and the network is adjusted according to its error	These are used to measure network generalization, and to halt training when generalization stops improving.	These have no effect on training and so provide an independent measure of network performance during and after training.



**Figure 5-7:** Representation of Neural Network

As from Figure 5-7, it is clear that six inputs and one output data had been passed which went through 10 hidden layers. Two delays are used one from the input and the other one from the output to help reach the target. There are three algorithms that can be used to make a Neural Network models. The description of each kind is provided below.

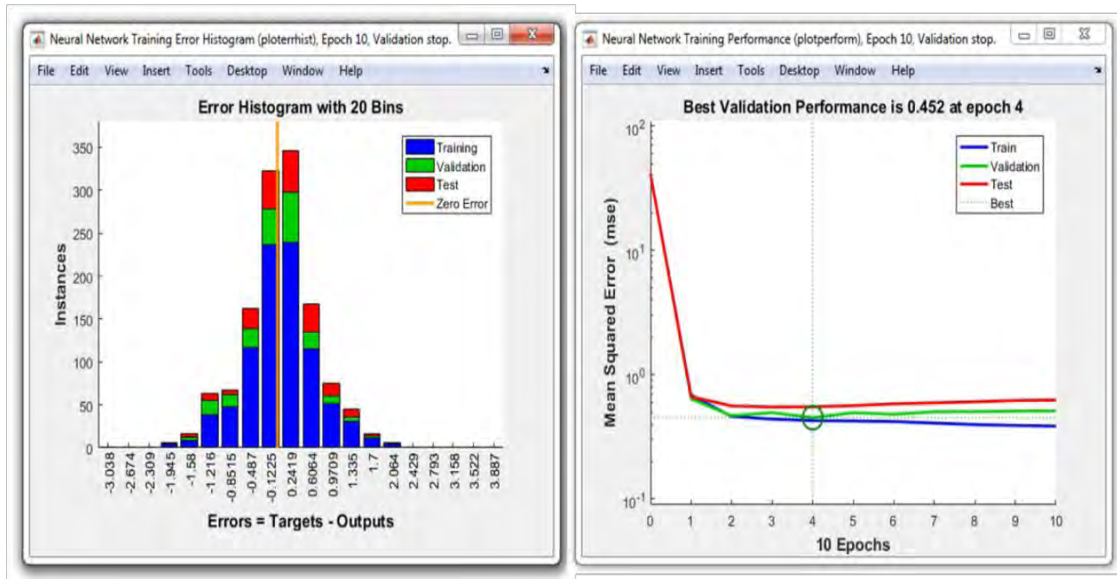
- i. Levenberg-Marquardt: This algorithm typically requires more memory but less time. Training automatically stops when generalization stops improving, as indicated by an increase in the mean square error of the validation samples.
- ii. Bayesian Regularization: This algorithm typically requires more time, but can result in good generalization for difficult, small or noisy datasets. Training stops according to adaptive weight minimization (regularization).
- iii. Scaled Conjugate Gradient: This algorithm requires less memory. Training automatically stops when generalization stops improving, as indicated by an increase in the mean square error of the validation samples.

Considering all the facts, Lavenberg-Marquardt algorithm was chosen to train the model since even if requires more memory, it requires less time. Internal performance was measured by mean squared error. 10 epoch was required to train the model and the time required to train was four seconds. The summary of the training, validation and testing model is provided in Table 5-4

**Table 5-4:** The R and MSE for different type of Dataset

<b>Dataset</b>	<b>Target value</b>	<b>MSE</b>	<b>R</b>
Training	910	0.427690	0.484149
Validation	195	0.452001	0.466673
Testing	195	0.551869	0.320135

Figure 5-8 shows the error histogram in one side and the performance of the model on the other side. In error histogram, the error is plot against different instances. The errors of different kind (training, testing & validation) are plotted against the instances in the form of bars. In performance figure on the right side, it can be seen mean squared errors of training, testing and validation data are plotted against epoch. It should be noted that, the training of the model stops when the validation error become still.



**Figure 5-8:** Error Histogram and Performance of the Neutral Network

Same as the proposed model and the new ANFIS model, 260 model output and the corresponding output were used to perform statistical analysis in order to observe the performance of the new model. The criteria used for statistical analysis has been defined earlier in this chapter. A sample is provided about the steps to the statistical analysis are provided in Table 5-5.

**Table 5-5:** Statistical Analysis of Neural Network

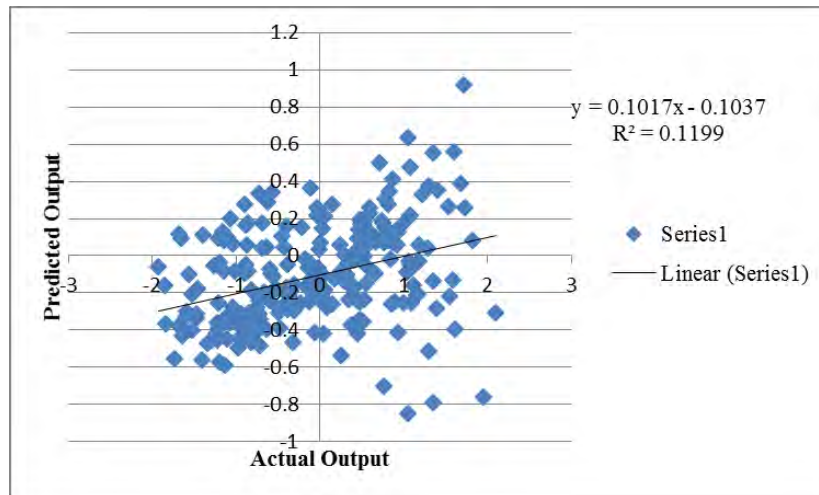
Actual	Forecast	A-F	c/a	c <sup>2</sup>	A- $\bar{A}$	c*f	f <sup>2</sup>	d- $\frac{MAPE}{100}$	i <sup>2</sup>
a	b	c	d	e	f	g	h	i	j
-0.729	0.3378	-1.066	-1.464	1.137	-0.656	0.700	0.432	0.7398	0.5472
-0.584	-0.094	-0.489	-0.838	0.2395	-0.512	0.2506	0.2622	1.3653	1.86393
0.257	-0.534	0.7918	3.0813	0.627	0.3288	0.2604	0.1081	5.2847	27.9284
1.2852	0.0431	1.2421	0.9665	1.5428	1.357	1.6856	1.8416	3.1699	10.0484
-0.636	0.0468	-0.683	-1.074	0.4667	-0.564	0.3856	0.3187	1.1299	1.27671
-0.032	0.2616	-0.293	-9.174	0.0862	0.0398	-0.011	0.0016	-6.971	48.5944
-0.562	-0.109	-0.455	-0.805	0.2048	-0.490	0.2218	0.2402	1.3981	1.95471
0.2911	-0.137	0.4284	1.4713	0.1835	0.363	0.1555	0.1318	3.6748	13.5041
-0.11	0.3647	-0.475	-4.304	0.2256	-0.038	0.0183	0.0015	-2.1004	4.41187
-0.861	-0.438	-0.422	-0.491	0.1784	-0.788	0.3332	0.6223	1.7128	2.93356
1.3536	0.5544	0.7992	0.5904	0.6387	1.4254	1.1392	2.0318	2.7939	7.80575
-1.527	-0.205	-1.321	-0.865	1.746	-1.454	1.9224	2.1167	1.338	1.79015
0.7626	-0.698	1.4608	1.9157	2.134	0.8344	1.2189	0.6962	4.1191	16.9672
1.7123	0.9159	0.7964	0.4651	0.6342	1.7842	1.4209	3.1833	2.6685	7.1211

Actual	Forecast	A-F	c/a	c <sup>2</sup>	A- $\bar{A}$	c*f	f <sup>2</sup>	d- $\frac{MAPE}{100}$	i <sup>2</sup>
a	b	c	d	e	f	g	h	i	j
-1.85	-0.161	-1.688	-0.913	2.8514	-1.777	3.0021	3.1607	1.2905	1.66548

Table 5-5 shows a sample of steps that is required for a statistical analysis. The result of the analysis is provided below.

Result of analysis

MAPE	220.345
RMSE	0.839679
R <sup>2</sup>	0.119881
SDE	31.99392
MAD	0.681848
MSE	0.705061
SMAPE	0.53607



**Figure 5-9:** Fitness of the NN prediction model

Figure 5-9 shows the overall fitness of the Neural Network, showing that the predicted outputs were plotted against the actual ones. The outputs lie around a straight line passing through the origin, which implies a very close agreement between the two.

### 5.3.4 Performance Comparison

So far, few criteria were defined for validation purposes. The performance of the proposed model was determined by the statistical analysis set by the defined criteria. Two more models were made and the performance of those models was calculated by statistical analysis. In Table 5-6 summarizes the performances of all the models.



**Table 5-6:** Summary of all the performances of all the models

	MAPE	RMSE	R <sup>2</sup>	SDE	MAD	MSE	SMAPE	R <sup>2</sup> from graph
Proposed Model	2.8269	0.0135	0.8868	0.2505	0.0091	0.0002	0.0047	0.886
New ANFIS model	824.90	0.6934	0.5633	112.045	0.5568	0.4808	0.3011	0.563
Neural Network model	220.35	0.8396	0.1198	31.993	0.6818	0.7050	0.5360	0.119

- i. The criterion mean absolute percentage error (MAPE) shows the mean absolute error that can be considered as a criterion to model risk to use it in real-world condition. Mean absolute percentage error (MAPE) is higher for the proposed model (2.827) as can be seen in Table 5.6. This error is lower for the new ANFIS model (-824.903). As the error for both new ANFIS model and Neural Network model is over 100%, it is advised to check for Symmetric Mean Absolute Percentage Error (SMAPE).
- ii. Symmetric Mean Absolute Percentage Error (SMAPE) is lower for the proposed model (0.004701) than the other two models. This error is higher for Neural Network model (0.53607).
- iii. Root mean squared error (RMSE) is a criterion for comparing error dimension in various models. Root mean squared error is lower for the proposed model (0.013559) than the other models where RMSE is relatively high for both models. For new ANFIS model RMSE is 0.693437. The error is higher for the neural network model (0.839679).
- iv. R-squared tells how well the data fit the regression model (the goodness of fit). R-squared can take any values between 0 to 1. The closer the R<sup>2</sup> value is to 1, the better the data fit the model. From Table 5.6, we can conclude that the validation data for proposed model fit the model best (0.886853). Neural network model provides a poor fit of the data to the model (0.119881).
- v. Standard deviation error (SDE) indicates the persistent error even after calibration of the model. Proposed model shows better less SDE which is 0.250551 whereas the new ANFIS model made for validation of the proposed model shows higher error (112.0451)?
- vi. Mean absolute error (MAD) measures the size of the error of a model. It can be seen from Table 5.6 that, this error is less for the proposed model (0.009148). MAD seems to be over 0.5 for both the new ANFIS model and the Neural Network model.

- vii. The mean squared error tells how close a regression line is to a set of points. From Table 5.6, it can be observed that this error is less for the proposed model (0.000184) which means the data set are very close to the regression line. The error is higher for the rest two models (the new ANFIS model: 0.480855 & for Neural Network model: 0.705061).
- viii. R-squared has also been measured by plotting the actual and predicted outputs. The values of r-squared were nearly same as the one found from the equation discussed earlier. As before,  $R^2$  is higher for the proposed value which means the data fit close to the regression line.  $R^2$  value for the new ANFIS model is higher than the one for Neural Network.
- ix.

#### **5.4 Summary**

In this chapter, the details of validation of the proposed model have been presented. In order to validate the Adaptive Neuro-fuzzy Inference System (ANFIS), first the criteria were set and the statistical analysis of the proposed model was performed based on the set criteria. Then two more models were made and analyzed statistically. Finally, the performance of the proposed model was compared with the new two models created.

## CHAPTER 6

### CONCLUSIONS AND RECOMMENDATIONS

#### 6.1 Conclusions

The primary objective of this study was the development of car following models for heterogeneous road traffic condition, by using the platform of Adaptive Neuro-Fuzzy Inference System (ANFIS). The outcome of the study was a Neuro-Fuzzy models to depict car-car following behavior, which has later been validated by finding error of the model. In this section, the entire study has been summarized and the findings have been highlighted.

An extensive review of literature has been carried out about car following as well as other topics related to this study (e.g. fuzzy logic) from previous research work in Bangladesh and abroad. It was observed that not much research has been carried out about this field in the heterogeneous road traffic situation of Bangladesh, whereas much research work has been carried out abroad. The focus of this study is to make a model a car following model for traffic stream having weak lane discipline.

The car following data have been collected at a suitable site by video recording vehicular movements and the car following parameters have been extracted by converting the video footage coordinates into real world positions by using a code written in Python programming, which have subsequently been converted into length, speed, acceleration etc by the same written code. The ‘Neuro-Fuzzy Designer’ toolbox of MATLAB has been used for the development and calibration of a car-car model based on the data extracted from video footage. Finally, the validation of model has been performed by checking the performance of the model against two other models.

#### 6.2 Observations on the Models Developed

By using the ‘Neuro-Fuzzy Designer’ toolbox of MATLAB for their development, each of the Neuro-Fuzzy model of car following behavior mainly intended to produce the output in the form of acceleration of the follower, subjected to a given set of inputs. The main features of the car following models are listed as follows:

- a) Each model involved six inputs namely ‘Relative Speed’ (RS), ‘Distance’ (DS), ‘Acceleration of Leader’ (AL) and ‘Speed of Follower’ (SF), ‘Left Lateral Distance’ (LLDS), ‘Right Lateral Distance’ (RLDS), while the only output was the ‘Acceleration of the Follower’ (AF) subjected to a constant time lag of 1 second.
- b) Each of the inputs were categorized into two or three linguistic categories and definite names were assigned to each category, whereas the outputs were categorized as one hundred and forty-four different singletons using default names, in order to apply the 'Zero Order Sugeno' method.
- c) All the input membership functions were represented as the geometric properties of Gaussian function which being initialized automatically, whereas each of the singletons of the output membership functions were initialized as zero.
- d) In each model, there were a total of one hundred and forty-four fuzzy rules through permutation and combination of the linguistic categories of each input corresponding to a unique output membership function, where all the input variables were connected with the ‘AND’ operator (arithmetic minimum).
- e) The models were calibrated in the ANFIS editor of the ‘Neuro-Fuzzy Designer’ toolbox and the optimization technique was chosen as hybrid method with 1000 epochs. Although lesser epoch could have been taken which would save much time.

The validation of the model was done defining some criterion to find error of the model. The main features of the validation are:

- a) Root mean squared error (RMSE) for the training dataset seems higher than the validation dataset. As a random set of data were used for validation of the model, the less error in the validation proves better performance of the model.
- b) Even though the criterion mean absolute percentage error (MAPE) is greater than one for the proposed error, MAPE for the proposed model shows better result than the two models made for validation both of which had error over 100%.
- c) As mean absolute percentage error (MAPE) is greater than 100%, it is advised to measure Symmetric Mean Absolute Percentage Error (SMAPE). It is found that the SMAPE for the proposed model is much lower ( $>0.1$ ) than the other models created for validation.

- d) Root mean squared error (RMSE) is a standard measure to find model error. It can be observed that RMSE for the proposed model is lower than 0.1 whereas the same error for the other models is much higher.
- e) Standard deviation error (SDE) which indicates the persistent error after calibration of the model. The error after the calibration seems quite less for the proposed model than the other models.
- f) Mean absolute error (MAD) measures the size of the error of a model. Mean absolute error of the proposed models is insignificant compared to other two models made for validation of the proposed model.
- g) The results from mean squared error for the proposed model shows that the data are really close to the regression line as compared to the other two models.
- h) R-squared was determined both using equation and graph. For both the methods, the data used for proposed model fits close to the regression line.  $R^2$  for other models indicates the data are far from the regression line and scattered widely.
- i) It can be sum up from the above discussion that, the error for the proposed model is lower than the other models and the data shows a good fitness of the model.
- j) Adaptive Neuro-Fuzzy Inference System which is used for this study is an appropriate method to observe the car following behavior of traffic
- k) The model developed in this study can further be used to simulate traffic.

### **6.3 Constraints of the Research**

Although the study was able to successfully model the car following behavior of heterogeneous road traffic by using Adaptive Neuro-Fuzzy Inference System (ANFIS), it had a number of inherent constraints as discussed below:

- a) In the study, all the models were developed using only trajectory data of vehicle types on the roads of Dhaka city. Due to unavailability during daytime, trucks, one of the most important motorized vehicle types that affect the roadway capacity significantly, were not addressed in the study. Non-motorized vehicles and motorcycles, which form the majority of traffic on the roads, are also not included in the study.

- b) Because the study used video technique for data collection, the field of view on the video footage was limited. This posed a maximum limitation on the speed of vehicles that can be covered in the study.
- c) In this study, car following data were decided to be collected at only one site, namely just upstream of one of the approaches of the signalized T- junction at Science lab at a particular time only. This decision regarding site selection for data collection is unable to address the possibility that the car following behavior of vehicles may be changed under a new set of traffic conditions and environmental conditions at other sites.
- d) Because the image quality of the captured video footage was mediocre, using image processing, it was difficult to obtain certain data from the video. Therefore, that data needed to be discarded. Had the video footage been of high quality, such drawbacks would not have been present.
- e) Due to the presence of obstacles in the road, few data needed to be discarded.
- f) Since image processing is used for data collection and sunlight acts as an object itself, the duration and time of the data collection needed to be limited to a specific time of the day.
- g) While analyzing with image processing, the occlusion effect and the camouflage effect was not accounted in this study. Nathanail and Karakikes (2018) explained them best in their chapter on problems with heterogeneous traffic.
- h) While developing the ANFIS model, from membership function to layer development in every step Gaussian function was used.
- i) Stability (local & asymptotic stability) of the car following model was not tested.

#### **6.4 Recommendations for Future Research**

The contribution of this study in the field of microscopic modeling of vehicular traffic flow can draw a number of research lines for future studies in the same field, which were not addressed in this study due to its limited scope and due to various constraints. The recommendations for future research are listed in the following points:

- a) The parameters will vary with different lane. For example, the lane close to the divider can have a parameter which defines the distance between the subject vehicle and the divider as a driver has to be aware of the divider while

driving. Therefore, separate models can be defined for each lane which will provide more accurate models.

- b) As the distresses on the road are abundant, a driver has to either decelerate each time or make a maneuver to avoid a distress. Since it influences the driving behavior of a driver, it is logical to consider distress as a parameter of the model.
- c) Traffic in the road can be simulated and more data can be generated for heterogeneous road traffic conditions using professional microscopic simulation packages like PARAMICS, VISSIM or AIMSUN. The models developed in this study should be used for the adjustment of car following parameters in those packages for subsequent testing of them in heterogeneous road traffic situations.
- d) The effect of demographic factors (e.g. age, gender, driving experience, alcohol addiction) could not be addressed. The inclusion of these factors can make the models more reasonable.
- e) The type of side vehicle on both sides of the subject vehicle varies in a section of a road. The effect of the type of side vehicles on the subject vehicle can be analyzed and used to make more advanced car-following models.
- f) For the model development, only the lateral distance of the side vehicle has been considered. The effect of the velocity of the side vehicles on the subject vehicle can be analyzed further to generate a more efficient car-following model.
- g) The study did not use any data of motorcycle and rickshaw to develop the model. The trajectory data of motorcycle and rickshaw can be obtained by redefining the contour area of the data extraction model.
- h) Models can be developed for different times of the day and days of the week as the traffic varies with both time and day.
- i) Gaussian function was used in every step of model development. Different functions can be used for different steps and the behavior of the model can be tested.
- j) Stability of the car following model can be tested in further studies.

## REFERENCES

- Adnan, M. A., Sulaiman, N., Zainuddin, N. I., and Besar, T. B. H. T. (2013, April). Vehicle speed measurement technique using various speed detection instrumentation. In 2013 IEEE Business Engineering and Industrial Applications Colloquium (BEIAC) (pp. 668-672). IEEE.
- Ahmed, K. I. (1999). Modeling drivers' acceleration and lane changing behavior (Doctoral dissertation, Massachusetts Institute of Technology).
- Arasan, V. T., and Koshy, R. Z. (2005). Methodology for modeling highly heterogeneous traffic flow. *Journal of Transportation Engineering*, 131(7), 544-551.
- Asaithambi, G., Kanagaraj, V., Srinivasan, K. K., and Sivanandan, R. (2018). Study of traffic flow characteristics using different vehicle-following models under mixed traffic conditions. *Transportation letters*, 10(2), 92-103.
- Brackstone, M., Sultan, B., and McDonald, M. (2002), Motorway Driver Behaviour: Studies on Car-Following, *Transportation Research Part F*, 5, pp. 329–344.
- Brackstone, M., and McDonald, M. (1999). Car-following: a historical review. *Transportation Research Part F: Traffic Psychology and Behaviour*, 2(4), 181-196.
- Braunstein, M. L., and Laughery, K. R. (1964). Detection of vehicle velocity changes during expressway driving. *Human factors*, 6(4), 327-331.
- Chakroborty, P., & Kikuchi, S. (2003). Calibrating the membership functions of the fuzzy inference system: instantiated by car-following data. *Transportation Research Part C: Emerging Technologies*, 11(2), 91-119.
- Chandler, R. E., Herman, R., and Montroll, E. W. (1958). Traffic dynamics: studies in car following. *Operations research*, 6(2), 165-184.
- Chen, D., Yuan, Y., Li, B., and Wu, J. (2005, August). Validation and comparison of microscopic car-following models using beijing traffic flow data. In *International Conference on Fuzzy Systems and Knowledge Discovery* (pp. 1008-1011). Springer, Berlin, Heidelberg.
- Cho, H. J., and Wu, Y. T. (2004, October). Modeling and simulation of motorcycle traffic flow. In 2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat.No. 04CH37583) (Vol. 7, pp. 6262-6267). IEEE.



- Choudhury, C. F., Ramanujam, V., and Ben-Akiva, M. E. (2009). Modeling acceleration decisions for freeway merges. *Transportation research record*, 2124(1), 45-57.
- Constantine, T.; Young, A.P (1967).*Traffic Dynamics: Car Following Studies*. *Traffic Engineering and Control*, Vol. 8(1), pp. 551-554.
- Czajewski, W., and Iwanowski, M. (2010, September). Vision-based vehicle speed measurement method. In *International Conference on Computer Vision and Graphics* (pp. 308-315). Springer, Berlin, Heidelberg.
- Raju, P. D. R., and Neelima, G. (2012). Image segmentation by using histogram thresholding. *International Journal of Computer Science Engineering and Technology*, 2(1), 776-779.
- Doğan, S., Temiz, M. S., and Külür, S. (2010). Real time speed estimation of moving vehicles from side view images from an uncalibrated video camera. *Sensors*, 10(5), 4805-4824.
- Eddie, L. C. (1961). Car-following and steady-state theory for noncongested traffic. *Operations research*, 9(1), 66-76.
- Garg, M., and Goel, S. (2013). Real-time license plate recognition and speed estimation from video sequences. *ITSI Transactions on Electrical and Electronics Engineering*, 1(5), 1-4.
- Gazis, D. C., Herman, R., and Potts, R. B. (1959). Car-following theory of steady-state traffic flow. *Operations research*, 7(4), 499-505.
- Gazis, D. C., Herman, R., and Rothery, R. W. (1961). Nonlinear follow-the-leader models of traffic flow. *Operations research*, 9(4), 545-567.
- Ghaffari, A., Alimardanii, F., and Khodayari, A. (2012, September). Predicting the future state of a vehicle in a stop&go behavior based on ANFIS models design. In *2012 6th IEEE International Conference Intelligent Systems* (pp. 368-373). IEEE.
- Ghariani, N., Elkosantini, S., Darmoul, S., and Said, L. B. (2014, May). A survey of simulation platforms for the assessment of public transport control systems. In *2014 International Conference on Advanced Logistics and Transport (ICALT)* (pp. 85-90). IEEE.
- Gipps, P. G. (1981). A behavioural car-following model for computer simulation. *Transportation Research Part B: Methodological*, 15(2), 105-111.
- Greenberg, H. (1959). An analysis of traffic flow. *Operations research*, 7(1), 79-85.

- Greenshields, B. D., Channing, W., and Miller, H. (1935). A study of traffic capacity. In Highway research board proceedings (Vol. 1935). National Research Council (USA), Highway Research Board.
- Gunay, B. (2007). Car following theory with lateral discomfort. *Transportation Research Part B: Methodological*, 41(7), 722-735.
- Gurusinghe, G. S., Nakatsuji, T., Azuta, Y., Ranjitkar, P., and Tanaboriboon, Y. (2002). Multiple car-following data with real-time kinematic global positioning system. *Transportation Research Record*, 1802(1), 166-180.
- Helander, M. G. (1975). Physiological reactions of drivers as indicators of road traffic demand. *Transportation research record*, 530, 1-17.
- Herman, R., and Rothery, R. W. (1965). Car-Following and Steady-State Flow. in: J. Almond (Ed.) *Proceedings of the 2nd International Symposium on the Theory of Traffic Flow*, pp. 1–11 (Paris: Organization for Economic Co-operation and Development (OECD)).
- Herman, R., Montroll, E. W., Potts, R. B., and Rothery, R. W. (1959). Traffic dynamics: analysis of stability in car following. *Operations research*, 7(1), 86-106.
- Hoffmann, E. R. (1966). Note on detection of vehicle velocity changes. *Human factors*, 8(2), 139-142.
- Hoque, M. S. (1997). The modelling of signalized intersections in developing countries. (Doctoral dissertation, University of Southampton).
- Hossain, M. (1996). Modelling of traffic operations in urban networks in developing countries (Doctoral dissertation, University of Southampton).
- Imran, A. (2009). Neuro-Fuzzy Model for Car-Following Behaviour in Heterogeneous Road Traffic Condition (Master's Thesis, Bangladesh University of Engineering and Technology).
- Inokuchi, H., Kawakami, S., Ogino, H., Noda, K. and Kurimoto, Y. (1999). The Car Following Models with the Consideration of the Driver's Attribute. In proceedings of the 6th World Congress on Intelligent Transport Systems, Toronto, Canada.
- Islam, M. (2013). Development of an acceleration model for traffic streams having weak lane discipline. (Master's Thesis, Bangladesh University of Engineering and Technology).
- Jang, J. S. (1993). ANFIS: adaptive-network-based fuzzy inference system. *IEEE transactions on systems, man, and cybernetics*, 23(3), 665-685.

- Kaur, A., and Kaur, A. (2012). Comparison of fuzzy logic and neuro-fuzzy algorithms for air conditioning system. *International journal of soft computing and engineering*, 2(1), 417-20
- Khodayari, A., Ghaffari, A., Kazemi, R., and Manavizadeh, N. (2010, September). ANFIS based modeling and prediction car following behavior in real traffic flow based on instantaneous reaction delay. In *13th International IEEE Conference on Intelligent Transportation Systems* (pp. 599-604). IEEE.
- Ghaffari, A., Khodayari, A. R., Kamali, A., and Tajdari, F. A. (2015). A New Model of Car Following Behavior Based on Lane Change Effects using Anticipation and Evaluation Idea. *Iranian Journal of Mechanical Engineering*, 16(2), 26-38.
- Kikuchi, S., and Chakroborty, P. (1992). Car-following model based on fuzzy inference system. *Transportation Research Record*, 82-82.
- Kikuchi, C., and Chakroborty, P. (1992). Fuzzy neural network control of complex systems: a study on longitudinal vehicle control. In *Proc World Congr Neural Networks* (Vol. 3, pp. 84-92).
- Kim, T. (2005). Analysis of variability in car-following behavior over long-term driving maneuvers (Doctoral dissertation, Department of Civil and Environmental Engineering, University of Maryland, College Park)
- Kokkinogenis, Z., Passos, L. S., Rossetti, R., and Gabriel, J. (2011). Towards the next-generation traffic simulation tools: a first evaluation. In *Proceedings of the 6th Doctoral Symposium In Informatics Engineering*. FEUP, University of Porto, Portugal (pp. 77-90).
- Kometani, E. I. J. I., and Sasaki, T. S. U. N. A. (1958). On the stability of traffic flow (report-I). *J. Oper. Res. Soc. Japan*, 2(1), 11-26.
- Kometani, E., and Sasaki, T. (1959). A safety index for traffic with linear spacing. *Operations research*, 7(6), 704-720.
- Lan, L. W., and Chang, C. W. (2004). Motorcycle-following models of general motors (gm) and adaptive neuro-fuzzy inference system. *Transportation Planning Journal*, 33(3), 511-536.
- Lee, D. N. (1976). A theory of visual control of braking based on information about time-to-collision. *Perception*, 5(4), 437-459.

- Lee, T. C. (2007). An agent-based model to simulate motorcycle behaviour in mixed traffic flow (Doctoral dissertation, Imperial College London (University of London)).
- Lee, T. C., Polak, J. W., and Bell, M. G. (2009). New approach to modeling mixed traffic containing motorcycles in urban areas. *Transportation Research Record*, 2140(1), 195-205.
- Huang, C. L., and Ma, H. N. (2012, August). A Moving Object Detection Algorithm for Vehicle Localization. In 2012 Sixth International Conference on Genetic and Evolutionary Computing (pp. 376-379).IEEE.
- Lucas, B. D., and Kanade, T. (1981).An iterative image registration technique with an application to stereo vision.Joint Conference on Artificial Intelligence, pp. 674–679
- Maciejewski, M. (2010). A comparison of microscopic traffic flow simulation systems for an urban area. *Transport Problems*, 5, 27-38.
- Madasu, V. K., & Hanmandlu, M. (2010, June). Estimation of vehicle speed by motion tracking on image sequences. In 2010 IEEE Intelligent Vehicles Symposium (pp. 185-190).IEEE.
- Maduro, C., Batista, K., Peixoto, P., and Batista, J. (2008, October). Estimation of vehicle velocity and traffic intensity using rectified images. In 2008 15th IEEE International Conference on Image Processing (pp. 777-780).IEEE.
- Mallikarjuna, C. (2007). Analysis and Modeling of Heterogeneous Traffic. PhD Thesis, Department of Civil Engineering, Indian Institute of Technology, New Delhi. (Doctoral dissertation, Department of Civil Engineering, Indian Institute of Technology, New Delhi).
- Mallikarjuna, C., Ramachandra, R. K., and Seethepalli, N. V. S. K. (2011). Analysis of microscopic data under heterogeneous traffic conditions. *Transport*, 25(3), 262-268.
- Maurya, A. K. (2011, June). Comprehensive approach for modeling of traffic streams with no lane discipline. In 2nd International Conference on Models and Technologies for Intelligent Transportation Systems.
- Michaels, R. M. (1963). Perceptual Factors in Car-following. *Proceedings of the Second International Symposium on the Theory of Road Traffic Flow*, pp. 44-59.

- Moein, S., Monadjemi, S. A., and Moallem, P. (2009). A novel fuzzy-neural based medical diagnosis system. *International Journal of Biological & Medical Sciences*, 4(3), 146-150.
- Nathanail, E. G. and Karakikes, I. D. (Eds.). (2018). *Data Analytics: Paving the Way to Sustainable Urban Mobility: Proceedings of 4th Conference on Sustainable Urban Mobility (CSUM2018)*, 24-25 May, Skiathos Island, Greece (Vol. 879). Springer.
- Oketch, T. G. (2000). New Modeling Approach for Mixed Traffic Streams with Non-motorized Vehicles. *Transportation Research Record: Journal of the Transportation Research Board*, No. 1705, Transportation Research Board of the National Academies, Washington, D.C., pp. 61-69.
- Ozaki, H. (1993). Reaction and anticipation in the car-following behavior. In *Proc. of 12th International Symposium on Theory of Traffic Flow and Transportation* (pp. 349-366).
- Peijiang, C. (2009, January). Moving object detection based on background extraction. In *2009 International Symposium on Computer Network and Multimedia Technology* (pp. 1-4). IEEE.
- Pipes, L. A. (1953). An operational analysis of traffic dynamics. *Journal of applied physics*, 24(3), 274-281.
- Quiroga, C. A. (1997). An integrated GPS-GIS methodology for performing travel time studies. (Doctoral dissertation, Louisiana State University, Louisiana, USA.)
- Rao, L., and Owen, L. (2000). Validation of high-fidelity traffic simulation models. *Transportation research record*, 1710(1), 69-78.
- Ratrout, N. T., and Rahman, S. M. (2009). A comparative analysis of currently used microscopic and macroscopic traffic simulation software. *The Arabian Journal for Science and Engineering*, 34(1B), 121-133.
- Ravishankar, K. V. R., and Mathew, T. V. (2011). Vehicle-type dependent car-following model for heterogeneous traffic conditions. *Journal of transportation engineering*, 137(11), 775-781.
- Rekersbrink, A. (1995). Mikroskopische Verkehrssimulation mit Hilfe der Fuzzy-logic. *Strassenverkehrstechnik*, 39(2).
- Reuschel, A. (1950). Fahrzeugbewegungen in der Kolonne. *Osterreichisches IngenieurArchiv*, 4, 193-215.

- Sayer, J. R., Mefford, M. L., and Huang, R. W. (2003). The effects of lead-vehicle size on driver following behavior: Is ignorance truly bliss? Proceedings of the Second International Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design.
- Schoepflin, T. N., and Dailey, D. J. (2003). Dynamic camera calibration of roadside traffic management cameras for vehicle speed estimation. *IEEE Transactions on Intelligent Transportation Systems*, 4(2), 90-98.
- Shanthikumar, J. G., and Sargent, R. G. (1983). A unifying view of hybrid simulation/analytic models and modeling. *Operations research*, 31(6), 1030-1052.
- Snider, J. N., and Ernst, R. L. (1965). A Study of Driver Variability in Car Following and Open Road Driving. *Highway Research Record*, (84).
- Sun, D. J., Zhang, L., and Chen, F. (2013). Comparative study on simulation performances of CORSIM and VISSIM for urban street network. *Simulation Modelling Practice and Theory*, 37, 18-29.
- Takagi, T., and Sugeno, M. (1993). Fuzzy identification of systems and its applications to modeling and control. In *Readings in fuzzy sets for intelligent systems* (pp. 387-403). Morgan Kaufmann.
- Todosiev, E. P. (1963). The action point model of the driver-vehicle system (Doctoral dissertation, The Ohio State University).
- Toledo, T. (2002). Integrated driving behavior modeling. (Doctoral dissertation, Massachusetts Institute of Technology).
- Treiterer, J., and Myers, J. (1974). The hysteresis phenomenon in traffic flow. *Transportation and traffic theory*, 6, 13-38.
- Underwood, R. T. (1960). Speed, volume, and density relationships. In *Quality and Density of Traffic Flow*, Yale Bureau of Traffic, 66-76.
- Wang, G., Xiao, D., and Gu, J. (2008, September). Review on vehicle detection based on video for traffic surveillance. In *2008 IEEE International Conference on Automation and Logistics* (pp. 2961-2966). IEEE.
- Wiedemann, R. (1974). Simulation des verkehrsflusses (Doctoral dissertation, IfV).
- Xiao, H., C. and Yung, N. H. (2007, February). A novel algorithm for estimating vehicle speed from two consecutive images. In *2007 IEEE Workshop on Applications of Computer Vision (WACV'07)* (pp. 12-12). IEEE.

Yikai, K., Satoh, J. I., Itakura, N., Honda, N., and Satoh, A. (1993, December). A fuzzy model for behaviour of vehicles to analyze traffic congestion. In Proceedings of the International Congress on Modelling and Simulation.

Zhiwei, H., Yuanyuan, L., and Xueyi, Y. (2007, December). Models of vehicle speeds measurement with a single camera. In 2007 International Conference on Computational Intelligence and Security Workshops (CISW 2007) (pp. 283-286). IEEE.

## **APPENDIX A**

### **PYTHON CODE AND OPENCV LIBRARIES TO EXTRACT NECESSARY CAR FOLLOWING PARAMETERS**



### ***#CODE TO GET THE RANGE OF A COLOR***

```
import cv2
import numpy as np
from matplotlib import pyplot as plt

def nothing(x):
    # any operation
    pass

frame = cv2.imread("grey1.PNG")
height , width , layers = frame.shape
new_h=height/2
new_w=width/2
resize = cv2.resize(frame, (new_w, new_h))

cv2.namedWindow("Trackbars")
cv2.createTrackbar("L-H", "Trackbars", 0, 180, nothing)
cv2.createTrackbar("L-S", "Trackbars", 0, 255, nothing)
cv2.createTrackbar("L-V", "Trackbars", 0, 255, nothing)
cv2.createTrackbar("U-H", "Trackbars", 180, 180, nothing)
cv2.createTrackbar("U-S", "Trackbars", 255, 255, nothing)
cv2.createTrackbar("U-V", "Trackbars", 255, 255, nothing)

while True:
    median = cv2.medianBlur(resize, 5)
    hsv = cv2.cvtColor(median, cv2.COLOR_BGR2HSV)

    l_h = cv2.getTrackbarPos("L-H", "Trackbars")
    l_s = cv2.getTrackbarPos("L-S", "Trackbars")
    l_v = cv2.getTrackbarPos("L-V", "Trackbars")
    u_h = cv2.getTrackbarPos("U-H", "Trackbars")
    u_s = cv2.getTrackbarPos("U-S", "Trackbars")
    u_v = cv2.getTrackbarPos("U-V", "Trackbars")

    lower_red = np.array([l_h, l_s, l_v])
    upper_red = np.array([u_h, u_s, u_v])

    mask = cv2.inRange(hsv, lower_red, upper_red)

    kernel = np.ones((2, 2), np.uint8)
    erosion = cv2.erode(mask, kernel, iterations=1)

    key = cv2.waitKey(1) & 0xff

    # if not ret:
    #     break
```

```

        #cv2.namedWindow( 'frame', cv2.WINDOW_NORMAL);
        #cv2.imshow('frame', frame)
cv2.namedWindow( 'Trackbars', cv2.WINDOW_NORMAL);
cv2.imshow("Trackbars", erosion)

if key == ord('p'):

while True:

        key2 = cv2.waitKey(1) or 0xff

if key2 == ord('p'):
break

if key == 27:
break

cap.release()
cv2.destroyAllWindows()

```

*#CODE TO EXTRACT NECESSARY PARAMETERS*

```

import cv2
import numpy as np
import time
import uuid
import datetime

# Create a VideoCapture object and read from input file
# If the input is the camera, pass 0 instead of the video
file name
cap = cv2.VideoCapture('05.05 PM.mp4')
fps = cap.get(cv2.CAP_PROP_FPS)

kernelSize=1
DISTANCE = 85.5
# The weighting to apply to "this" frame when averaging.
A higher number
# here means that the average scene will pick up changes
more readily,

```

```

# thus making the difference between average and current
scenes smaller.
DEFAULT_AVERAGE_WEIGHT = 0.04
MIN_AREA= 45000
# The maximum distance a blob centroid is allowed to move
in order to
# consider it a match to a previous scene's blob.
BLOB_LOCKON_DISTANCE_PX = 80
# The number of seconds a blob is allowed to sit around
without having
# any new blobs matching it.
BLOB_TRACK_TIMEOUT = 0.7
# The left and right X positions of the "poles". These
are used to
# track the speed of a vehicle across the scene.
LEFT_POLE_PX = 320
RIGHT_POLE_PX = 500
# Constants for drawing on the frame.
LINE_THICKNESS = 1
CIRCLE_SIZE = 5
font = cv2.FONT_HERSHEY_SIMPLEX
# A variable to store the running average.
avg = None
# A list of "tracked blobs".
tracked_blobs = []
#kernel =
cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (3,3))
#sift = cv2.xfeatures2d.SIFT_create()
from itertools import tee, izip
def pairwise(iterable):
    "s -> (s0,s1), (s1,s2), (s2, s3), ..."
    a, b = tee(iterable)
    next(b, None)
    return izip(a, b)
def get_speed (pixel,distance,fps):
    image_width =3840
    sec= image_width/fps

    #sec= abs(last_seen-first_seen)
    #print ("sec",sec)
    #mph
    speed = ((pixel*distance)/sec)*0.681818* 0.44704
    #speed = ((pixel*distance)/sec)
    #kmph
    #speed = ((pixel*distance)/sec)*3.6
    return speed
if (cap.isOpened()== False):
print("Error opening video stream or file")

# Read until video is completed

```

```

while(cap.isOpened()):
    # Capture frame-by-frame
    ret, frame = cap.read()
    dtime = cap.get(cv2.CAP_PROP_POS_MSEC)
    #fps = cap.get(cv2.CAP_PROP_FPS)
    #print (fps)
    #total_frames = cap.get(cv2.CAP_PROP_FRAME_COUNT)
    print(float(dtime))
    font = cv2.FONT_HERSHEY_SIMPLEX
    text= "Time: {} sec".format("%.0f" % int(dtime))
    cv2.putText(frame,text,(900,50), font,
    2,(0,255,0),2,cv2.LINE_AA)

    if ret == True:
        frame_time = time.time()

        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        blur = cv2.medianBlur(frame, 5)

        hsv = cv2.cvtColor(blur, cv2.COLOR_BGR2HSV)

        lower_red = np.array([2, 2, 180])
        upper_red = np.array([90, 53, 218])

        mask = cv2.inRange(hsv, lower_red, upper_red)

        kernel = np.ones((5, 5), np.uint8)

        #dilation = cv2.dilate(mask, kernel)
        #kernel = np.ones((12, 12), np.uint8)
        #erosion = cv2.erode(dilation, kernel)
        opening = cv2.morphologyEx(mask, cv2.MORPH_OPEN, kernel,
        iterations=2)

        image,contours, hier =
        cv2.findContours(opening,cv2.RETR_TREE,cv2.CHAIN_APPROX_S
        IMPLE)

        biggest_area = 0
        defcontour_find(found_area ):
            if (cv2.contourArea(found_area) > MIN_AREA) and
            (cv2.contourArea(found_area) >biggest_area):
                #biggest_area = cv2.contourArea(found_area)
        return True

```

```

blobs = filter(lambda c: contour_find(c) , contours)

    # examine the contours, looking for the largest one
    '''for c in contours:
        (x1, y1, w1, h1) = cv2.boundingRect(c)
        # get an approximate area of the contour
found_area = w1*h1
        # find the largest bounding rectangle

if (found_area> MIN_AREA) and (found_area>biggest_area):
biggest_area = found_area
blobs = filter(lambda c: biggest_area = found_area,
contours)'''

if blobs:
for c in blobs:
    # Find the bounding rectangle and center for
each blob
    (x, y, w, h) = cv2.boundingRect(c)
center = (int(x + w/2), int(y + h/2))
cv2.circle(frame, ((x+w/2), (y+h/2)), 8, (0,0,255), -1)
cv2.rectangle(frame, (x,y), (x+w,y+h), (0,255,0), 2)

# Look for existing blobs that match this one
closest_blob = None
iftracked_blobs:
    # Sort the blobs we have seen in previous
frames by pixel distance from this one
closest_blobs = sorted(tracked_blobs, key=lambda b:
cv2.norm(b['trail'][0], center))

forclose_blob in closest_blobs:
distance = cv2.norm(center, close_blob['trail'][0])

    # Check if the distance is close
enough to "lock on"
if distance < BLOB_LOCKON_DISTANCE_PX:
    # If it's close enough, make sure
the blob was moving in the expected direction
expected_dir = close_blob['dir']
ifexpected_dir == 'left' and close_blob['trail'][0][0] <
center[0]:
continue
elifexpected_dir == 'right' and close_blob['trail'][0][0]
> center[0]:
continue
else:
closest_blob = close_blob
break

```

```

ifclosest_blob:
    # If we found a blob to attach this
blob to, we should
    # do some math to help us with speed
detection
prev_center = closest_blob['trail'][0]
    #print
('prev_center[0]',prev_center[0])
    #print ('center[0]',center[0])
    #print
('closest_blob[direction]',closest_blob['dir'])
if center[0] <prev_center[0]:
    # It's moving left
closest_blob['dir'] = 'left'
closest_blob['bumper_x'] = x
else:
    # It's moving right
closest_blob['dir'] = 'right'
closest_blob['bumper_x'] = x + w

    # ...and we should add this centroid
to the trail of
    # points that make up this blob's
history.
closest_blob['trail'].insert(0, center)
closest_blob['last_seen'] = frame_time
if not closest_blob:

    # If we didn't find a blob, let's make a
new one and add it to the list
    b = dict(
id=str(uuid.uuid4())[:8],
first_seen=frame_time,
last_seen=frame_time,
dir=None,
bumper_x=None,
trail=[center],
    )
tracked_blobs.append(b)

iftracked_blobs:
    # Prune out the blobs that haven't been seen in
some amount of time
fori in xrange(len(tracked_blobs) - 1, -1, -1):
ifframe_time - tracked_blobs[i]['last_seen'] >
BLOB_TRACK_TIMEOUT:
    #print "Removing expired track
{}".format(tracked_blobs[i]['id'])
deltracked_blobs[i]

```

```

for blob in tracked_blobs:
    for (a, b) in pairwise(blob['trail']):
        print a[0]
        print a[1]
        print b[0]
        print b[1]
        dist = cv2.norm(a, b)
            #print ('dist', dist)
        velocity= get_speed(dist,DISTANCE,fps)
        print velocity
            #text= '{} kmph'.format(velocity)
            #cv2.putText(dst,text,((x+w/2),(y+h/2)),
font, 2, (0,0,255),3,cv2.LINE_AA)

cv2.circle(frame, a, 3, (255, 0, 0), LINE_THICKNESS)

if blob['dir'] == 'left':
cv2.line(frame, a, b, (255, 255, 0), LINE_THICKNESS)
else:
cv2.line(frame, a, b, (0, 255, 255), LINE_THICKNESS)

bumper_x = blob['bumper_x']
ifbumper_x:
cv2.line(frame, (bumper_x, 100), (bumper_x, 500), (255,
0, 255), 3)

cv2.namedWindow( 'frame', cv2.WINDOW_NORMAL);
cv2.imshow('frame',frame)
    #cv2.imshow ('original', frame)

    # Press Q on keyboard to exit
if cv2.waitKey(25) & 0xFF == ord('q'):
break

    # Break the loop
else:
break

# When everything done, release the video capture object
cap.release()

# Closes all the frames
cv2.destroyAllWindows()

```

## **APPENDIX B**

### **DATA SHEETS FOR THE CALCULATION OF CAR FOLLOWING PARAMETERS FROM VEHICLE POSITIONS**



### Training and Testing Data

RS	DS	AL	SF	LLDS	RLDS	AF
0	1.068719	0.036222	1.003941	0.441127	1.693262	-0.04358
0.02532	1.068095	0.037213	0.976295	0.447913	1.69749	-0.02396
0.056205	1.058694	-0.44964	0.973513	0.447913	1.701781	-0.53355
0.018982	1.045754	0	1.010736	0.441127	1.713158	-0.06051
0.00346	1.033511	1.348921	0.94887	0.447913	1.710549	0.423097
0.143255	1.030441	-1.22857	1.003941	0.4547	1.715027	-1.32173
0.028103	1.00101	1.87965	1.001615	0.447913	1.719567	0.395057
0.142069	0.994281	-0.92418	0.94541	0.441127	1.731195	-0.96941
-0.11149	0.624364	0.037622	1.115427	0.739735	1.33323	0.188483
-0.13819	0.597218	0	1.142128	0.739735	1.337254	0.072043
-0.08586	0.563285	0.037213	1.087479	0.739735	1.348126	0.27552
-0.08431	0.542925	-0.44964	1.114025	0.726162	1.365844	-0.2045
-0.03502	0.522565	0	1.064739	0.719376	1.376836	0.210983
-0.25292	0.515779	1.348921	1.198333	0.726162	1.374315	0.918396
0.047592	0.4547	1.828571	1.099604	0.726162	1.385428	1.756805
-0.19672	0.468273	1.87965	1.226435	0.732949	1.376241	1.23588
-0.02655	0.420767	-0.92418	1.114025	0.726162	1.387474	-0.49739
-0.09638	0.41398	0.647308	1.143403	0.719376	1.398768	0.554245
-0.08431	0.386834	-0.6224	1.17023	0.726162	1.403335	-0.47525
-0.06127	0.376474	0	1.147196	0.719376	1.414749	0.078394
-0.11066	0.372901	0.421647	1.17023	0.732949	1.412651	0.345634
-0.08383	0.372576	0	1.143403	0.719376	1.424185	0.076202
-0.08431	0.36854	-0.42165	1.17023	0.726162	1.428994	-0.3079
-0.08383	0.36685	0.421647	1.143403	0.719376	1.440649	0.393589
-0.11183	0.364676	0	1.171399	0.726162	1.445578	0.015285
-0.05069	0.363375	-0.08271	1.115427	0.726162	1.450568	0.107326
-0.17261	0.359566	0.110707	1.23043	0.726162	1.455617	-0.02338
0.251122	0.766882	1.534442	1.183411	1.004411	0.432753	0.387295
0.30447	0.807601	0	1.23676	0.990838	0.443528	0.028401
0.203874	0.861893	-0.75595	1.183411	0.970478	0.42037	-0.60039
0.061795	0.895826	-0.33177	1.212068	0.963692	0.417571	-0.30745
0.203874	0.963692	-1.63311	1.183411	0.963692	0.473053	-0.87497
0.110544	0.997625	-1.49328	1.183411	0.956905	0.483828	-0.90303
0.136551	1.017984	-0.0424	1.212068	0.956905	0.487816	0.026366
-0.00098	1.045131	-1.74204	1.183411	0.943332	0.491804	-1.12319
0.061795	1.045131	0.545944	1.212068	0.936546	0.502579	0.308555
0.285168	1.058704	3.573959	1.212068	0.936546	0.506567	1.753479
0.031412	1.10621	-0.60159	1.183411	0.929759	0.517341	-0.36025

RS	DS	AL	SF	LLDS	RLDS	AF
0.107895	1.112996	1.223725	1.183411	0.929759	0.528116	0.720632
0.110544	1.133356	0.042395	1.183411	0.922973	0.532104	0.138016
0.098508	1.153716	-0.65109	1.212068	0.916186	0.542879	-0.22559
0.067714	1.174075	-0.0342	1.183411	0.916186	0.553653	0.099362
0.035097	1.187648	0	1.150795	0.916186	0.557641	0.119905
0.098508	1.194435	0.034199	1.212068	0.909399	0.575203	0.186895
0.067714	1.214795	-0.0342	1.183411	0.895826	0.585977	0.145351
0.069851	1.228368	0.034199	1.183411	0.88904	0.589965	0.190167
0.155219	1.241941	1.365886	1.183411	0.882253	0.60074	0.98077
0.067714	1.269087	-1.40009	1.183411	0.882253	0.611515	-0.47032
0.098508	1.28266	0.034199	1.212068	0.882253	0.615503	0.245137
0.069851	1.30302	0	1.183411	0.875467	0.626277	0.220865
0.075987	1.316593	0.608699	1.151504	0.875467	0.630265	0.586177
0.097095	1.330166	-0.6429	1.212793	0.861893	0.63115	-0.00722
0.163591	1.350526	0.685293	1.236458	0.861893	0.644723	0.60405
0.067232	1.384459	-0.68529	1.18293	0.855107	0.65151	-0.07574
0.111453	1.398032	0.685293	1.18432	0.84832	0.617577	0.604557
0.12076	1.418392	-0.68529	1.236458	0.84832	0.658297	0.113644
0.07076	1.445538	0.034199	1.18432	0.841534	0.665083	0.285683
0.14578	1.459111	0.608699	1.221296	0.841534	0.685443	0.599955
0.097095	1.486257	-0.6429	1.212793	0.834747	0.705803	0.124435
0.183491	1.506617	1.400085	1.211683	0.834747	0.699016	1.168844
0.099233	1.54055	-1.36589	1.212793	0.834747	0.699016	-0.055
0.137277	1.560909	0.608699	1.212793	0.827961	0.712589	0.620589
0.069369	1.588056	-0.6087	1.18293	0.827961	0.726162	0.0662
0.16438	1.601629	0.608699	1.239896	0.821174	0.732949	0.635199
0.137277	1.635562	0	1.212793	0.821174	0.739735	0.412829
0.099233	1.662708	-0.6087	1.212793	0.814388	0.760095	0.189117
0.163591	1.683068	0.651094	1.236458	0.814388	0.766882	0.665349
0.100361	1.717	-0.68529	1.216059	0.807601	0.773668	0.177611
0.12076	1.73736	0	1.236458	0.807601	0.794028	0.413215
0.139926	1.764506	0.685293	1.212793	0.807601	0.807601	0.684991
0.164463	1.791653	0	1.23733	0.800814	0.814388	0.471479
0.095986	1.825585	-0.68529	1.211683	0.800814	0.827961	0.181277
0.160941	1.845945	0.642898	1.236458	0.794028	0.834747	0.659272
0.139926	1.879878	0.042395	1.212793	0.787241	0.827961	0.431568
0.098123	1.907024	-0.65109	1.211683	0.787241	0.84832	0.156322
0.12076	1.927384	-0.0342	1.236458	0.787241	0.84832	0.398696
0.187019	1.95453	0.685293	1.259886	0.780455	0.855107	0.664189

<b>RS</b>	<b>DS</b>	<b>AL</b>	<b>SF</b>	<b>LLDS</b>	<b>RLDS</b>	<b>AF</b>
0.121632	1.995249	-0.68529	1.23733	0.780455	0.875467	0.206502
0.160941	2.022396	0.642898	1.236458	0.773668	0.882253	0.63095
0.121632	2.056328	-0.6429	1.23733	0.773668	0.88904	0.178936
0.12076	2.083475	0	1.236458	0.766882	0.902613	0.366578
0.121632	2.110621	0	1.23733	0.773668	0.902613	0.39168
0.163591	2.137767	0.685293	1.236458	0.773668	0.916186	0.651938
0.12076	2.1717	-0.68529	1.236458	0.780455	0.916186	0.207417
0.143693	2.198846	0.034199	1.257253	0.787241	0.922973	0.472115
0.097095	2.232779	-0.0342	1.212793	0.794028	0.922973	0.404325
0.146325	2.253139	0.034199	1.259886	0.794028	0.936546	0.492404
0.12076	2.287072	-0.0342	1.236458	0.800814	0.956905	0.460441
0.142223	2.314218	0	1.257921	0.800814	0.956905	0.496881
0.12377	2.348151	0.034199	1.23733	0.800814	0.970478	0.487413
0.143693	2.375297	0	1.257253	0.807601	0.977265	0.511975
0.142223	2.40923	-0.0342	1.257921	0.807601	0.984052	0.505107
0.143693	2.443163	0.034199	1.257253	0.807601	0.984052	0.523577
0.12076	2.477095	-0.0342	1.236458	0.807601	0.997625	0.485384
0.107886	2.504242	-0.58083	1.259886	0.814388	0.997625	0.36828
0.141555	2.531388	0.580826	1.257253	0.814388	1.011198	0.725971
0.105254	2.565321	-0.58083	1.257253	0.814388	1.017984	0.365342
0.142223	2.592467	0.580826	1.257921	0.821174	1.031558	0.739278
0.105254	2.6264	-0.58083	1.257253	0.821174	1.038344	0.383528
0.107649	2.653546	0.027629	1.257921	0.821174	1.045131	0.534122
0.107649	2.680692	0	1.257921	0.827961	1.051917	0.534998
0.12076	2.707838	0.553197	1.236458	0.827961	1.06549	0.745667
0.105254	2.734985	-0.58083	1.257253	0.834747	1.06549	0.410508
0.105254	2.762131	0	1.257253	0.841534	1.072277	0.549793
0.107649	2.789277	0.027629	1.257921	0.84832	1.072277	0.565762
0.107649	2.816423	0	1.257921	0.84832	1.08585	0.559944
0.105254	2.84357	-0.02763	1.257253	0.855107	1.092637	0.555055
0.105922	2.870716	0	1.257921	0.855107	1.099423	0.56602
0.074249	2.897862	-0.49607	1.257253	0.855107	1.10621	0.395603
0.143693	2.918222	1.111095	1.257253	0.861893	1.10621	1.171753
0.105922	2.952155	-0.61503	1.257921	0.86868	1.119783	0.441855
0.074249	2.979301	-0.49607	1.257253	0.86868	1.126569	0.407859
0.100182	2.999661	0.414928	1.257253	0.86868	1.126569	0.738024
0.146325	3.026807	0.696167	1.259886	0.875467	1.133356	0.870194
0.074249	3.06074	-1.1111	1.257253	0.875467	1.146929	0.30677
0.112967	3.081099	0.49607	1.264967	0.882253	1.153716	0.792427

RS	DS	AL	SF	LLDS	RLDS	AF
0.094339	3.148965	-0.02763	1.246339	0.88904	1.187648	0.571188
0.02868	3.243977	-0.49607	1.211683	0.895826	1.194435	0.272305
-0.17253	3.250763	0.49607	0.97947	0.895826	1.201222	0.796393
0.056169	3.291483	-0.5237	1.239173	0.916186	1.255514	0.376492
0.063082	0.936546	0	1.212046	0.604968	0.392323	0.023401
0.010728	0.977265	1.633114	1.236743	0.615627	0.40135	0.608106
0.101042	1.038344	-0.46451	1.212046	0.619499	0.417224	-0.03481
0.131448	1.072277	-0.66189	1.212046	0.630158	0.396372	0.03118
0.188783	1.099423	-1.23849	1.236743	0.647603	0.442366	-0.02371
0.188783	1.119783	0	1.212046	0.651475	0.458421	0.198973
0.167746	1.133356	0.45298	1.212046	0.655347	0.474536	0.260294
0.188783	1.153716	-0.45298	1.212046	0.65922	0.483925	0.091067
0.190068	1.167289	-0.02763	1.236743	0.669878	0.506947	0.203365
0.188783	1.187648	0.027629	1.212046	0.680537	0.523243	0.186864
0.163102	1.201222	0.553197	1.212046	0.684409	0.532813	0.291711
0.190068	1.221581	-0.58083	1.212046	0.688281	0.542443	0.00797
0.188783	1.235154	0.027629	1.236743	0.705727	0.565707	0.210457
0.190068	1.255514	-0.02763	1.212046	0.709599	0.575457	0.182454
0.188783	1.269087	0.027629	1.236743	0.713471	0.598841	0.221215
0.190068	1.289447	-0.02763	1.212046	0.72413	0.408713	0.160916
0.188783	1.30302	0.027629	1.236743	0.728002	0.425431	0.186071
0.190068	1.32338	-0.02763	1.237626	0.738661	0.442209	0.167547
0.188783	1.343739	0.027629	1.212793	0.746522	0.459048	0.156666
0.21424	1.357313	-0.54594	1.236458	0.760095	0.46916	-0.01576
0.188783	1.370886	0.545944	1.23733	0.760095	0.48612	0.335762
0.21424	1.391245	-0.54594	1.236458	0.773668	0.503139	-0.01916
0.2132	1.404818	0.022245	1.239896	0.780455	0.520219	0.225822
0.190068	1.418392	0.49607	1.24401	0.766882	0.530573	0.382899
0.21424	1.438751	-0.51831	1.23733	0.787241	0.55456	0.078338
0.235565	1.452324	-0.45551	1.257253	0.787241	0.565034	0.16826
0.2132	1.465898	0.477755	1.236458	0.787241	0.582355	0.482002
0.21424	1.479471	-0.02225	1.267164	0.787241	0.59295	0.329779
0.21424	1.49983	0	1.257253	0.821174	0.610392	0.373014
0.253347	1.52019	-0.83406	1.23733	0.821174	0.627894	0.274592
0.21424	1.52019	0.834059	1.257253	0.827961	0.652243	0.749203
0.234733	1.54055	-0.43776	1.257921	0.834747	0.663079	0.417026
0.21424	1.554123	0.437756	1.236458	0.855107	0.673975	0.589176
0.254003	1.567696	-0.84801	1.284788	0.855107	0.691718	0.573908
0.234733	1.581269	0.410253	1.24401	0.88904	0.709522	0.613734

RS	DS	AL	SF	LLDS	RLDS	AF
0.235565	1.588056	-0.01775	1.257921	0.86868	0.727386	0.553286
0.254003	1.601629	-0.3925	1.257253	0.875467	0.738523	0.604561
0.254003	1.608415	0	1.257253	0.882253	0.756507	0.597694
0.234733	1.615202	0.410253	1.257253	0.882253	0.774552	0.635529
0.269015	1.628775	-0.72891	1.257253	0.88904	0.785871	0.785487
0.255935	1.628775	0.277609	1.263029	0.88904	0.804036	0.63364
0.254003	1.635562	0.041047	1.257921	0.909399	0.815475	0.630316
0.269519	1.642348	-0.32934	1.212793	0.909399	0.833761	0.657425
0.269015	1.628775	0.010687	0.995696	0.902613	0.852107	0.261734
0.269519	1.574483	-0.01069	1.031427	0.916186	0.863727	0.296575
0.253347	1.526977	0.343293	1.028121	0.929759	0.882194	0.499128
0.283174	1.486257	-0.63242	1.028121	0.929759	0.900721	0.102279
0.269015	1.431965	0.299815	1.028121	0.929759	0.919308	0.481013
0.269519	1.384459	-0.01069	0.995696	0.936546	0.931169	0.31381
0.269519	1.330166	0	1.041035	0.950119	0.949877	0.376934
0.282084	1.28266	-0.26608	0.983648	0.963692	0.961859	0.16855
0.269015	1.221581	0.276766	1.031427	0.956905	0.980687	0.540458
0.282084	1.174075	-0.27677	1.028121	0.963692	0.99279	0.231685
0.282084	1.119783	0	0.983648	0.963692	1.011739	0.353498
0.282084	1.058704	0	1.028121	0.970478	1.023962	0.407782
0.281712	1.004411	0.007858	1.041035	0.970478	1.043032	0.44251
0.283174	0.950119	-0.03091	0.97947	0.984052	1.055375	0.359771
0.29142	0.88904	-0.1743	1.056099	0.990838	1.074565	0.382864
0.271001	0.827961	0.432013	0.97947	1.017984	1.08703	0.728518
0.29142	0.773668	-0.43201	1.028121	1.017984	1.106341	0.127234
0.29142	0.712589	0	0.983648	1.017984	1.125712	0.41654
0.282084	0.644723	0.197352	1.028121	1.024771	1.138357	0.621521
0.292419	0.590431	-0.21844	0.995696	1.031558	1.157849	0.246666
0.281712	0.522565	0.226303	1.028121	1.051917	1.170614	0.664032
0.291675	0.468273	-0.21059	0.983648	1.051917	1.197013	0.245492
0.291675	0.400407	0	1.028121	1.058704	1.209899	0.507511
0.282084	0.339328	0.202734	0.983648	1.058704	1.222846	0.651179
0.281712	0.278249	0.007858	1.031427	1.06549	1.235853	0.535593
0.291675	0.223957	-0.21059	0.983648	1.072277	1.255706	0.254916
0.142038	1.73736	1.363893	1.028192	0.466271	0.746249	-0.5004
0.024609	1.703427	-0.07443	1.150273	0.466271	0.746892	-0.17396
0.05667	1.689854	0.074427	1.11356	0.466271	0.74777	-0.04277
-0.11326	1.669494	2.216247	1.144979	0.459484	0.748872	-0.48338
0.014574	1.683068	-1.13157	1.212863	0.48663	0.723041	-0.92883

RS	DS	AL	SF	LLDS	RLDS	AF
-0.09884	1.669494	1.8146	1.212863	0.48663	0.731344	0.305247
-0.09884	1.683068	0	1.212863	0.378045	0.726267	-0.14814
-0.06512	1.696641	-0.91798	1.236523	0.384832	0.721371	-0.80543
0.022494	1.696641	-0.43094	1.175838	0.350899	0.757366	-0.50797
-0.1021	1.676281	1.348921	1.216129	0.378045	0.732444	0.348986
-0.06889	1.689854	-0.08905	1.188483	0.500204	0.721247	-0.20757
-0.12453	1.696641	0.06662	1.23996	0.513777	0.710193	-0.08614
0.0383	1.717	-0.87685	1.13193	0.411978	0.699273	-1.03527
-0.101	1.696641	1.324018	1.188483	0.391619	0.722409	0.310513
-0.14329	1.710214	-0.42474	1.257311	0.378045	0.738874	-0.51653
-0.08836	1.73736	0.424737	1.175838	0.378045	0.741873	0.128189
-0.17325	1.744147	-0.42474	1.287279	0.405192	0.717823	-0.46676
-0.04305	1.784866	0	1.157071	0.411978	0.714222	-0.13885
-0.12583	1.784866	0.44964	1.211754	0.425551	0.703916	0.145767
-0.17185	1.805226	-0.47207	1.287279	0.425551	0.707257	-0.45269
-0.10177	1.845945	-0.06662	1.221365	0.418765	0.717449	-0.17171
-0.09847	1.859518	0.538687	1.184394	0.398405	0.741272	0.150468
-0.20831	1.873091	-0.0249	1.295787	0.411978	0.738	-0.26243
-0.19946	1.913811	0	1.286943	0.459484	0.687264	-0.09257
-0.05788	1.981676	1.369317	1.115698	0.479844	0.677761	-0.0253
-0.20681	1.988463	0.44964	1.236523	0.473057	0.688206	0.168696
-0.14443	2.029182	-0.998	1.236523	0.473057	0.691873	-0.79998
-0.12694	2.056328	1.098716	1.212863	0.473057	0.695543	0.204554
-0.14395	2.076688	-0.44964	1.257979	0.479844	0.692424	-0.2987
-0.18115	2.103834	1.316966	1.212863	0.479844	0.689295	0.319454
-0.17957	2.137767	-0.41769	1.237395	0.493417	0.686151	-0.37332
-0.11458	2.1717	-1.36932	1.257979	0.493417	0.682987	-0.60612
-0.20781	2.19206	2.231385	1.211754	0.48663	0.69337	0.035148
-0.18869	2.232779	-1.31171	1.274608	0.48663	0.696936	-0.7987
-0.21262	2.273498	0	1.29854	0.500204	0.69368	-0.0451
-0.09245	2.327791	0.44964	1.150273	0.466271	0.724315	0.098997
-0.18603	2.341364	-0.92171	1.301462	0.466271	0.727759	-0.52466
-0.17695	2.38887	0.893713	1.236523	0.513777	0.683648	0.259695
-0.21679	2.422803	0.027994	1.274608	0.513777	0.686988	0.012121
-0.11391	2.470309	-1.36932	1.257311	0.50699	0.697057	-0.51394
-0.24283	2.490668	1.369317	1.300651	0.50699	0.700277	0.317376
-0.15877	2.558534	0.44964	1.188483	0.50699	0.703432	0.122789
-0.25809	2.58568	-0.03195	1.289809	0.52735	0.692945	-0.04901
-0.16202	2.646759	-1.31697	1.276049	0.540923	0.675599	-0.28883

<b>RS</b>	<b>DS</b>	<b>AL</b>	<b>SF</b>	<b>LLDS</b>	<b>RLDS</b>	<b>AF</b>
-0.14592	2.680692	0	1.259942	0.534136	0.69211	0.059847
-0.20987	2.707838	0.788574	1.274608	0.54771	0.681396	0.291927
-0.27299	2.755344	1.009987	1.274608	0.534136	0.69096	0.41497
-0.18857	2.816423	-1.37382	1.276049	0.54771	0.686867	-0.43454
-0.13071	2.857143	-0.89477	1.274117	0.561283	0.675901	0.087891
-0.22823	2.884289	1.787003	1.259942	0.561283	0.67842	0.246006
-0.18176	2.931795	-1.31697	1.295787	0.574856	0.667274	-0.03266
-0.14395	2.979301	0	1.257979	0.568069	0.669609	0.249134
-0.15404	3.006447	0.44964	1.23996	0.581642	0.665063	0.306152
-0.31197	3.033593	1.754038	1.288261	0.568069	0.680781	1.099212
-0.16062	3.108246	-1.2261	1.276049	0.568069	0.669255	0.042826
-0.11763	3.142178	-0.1105	1.23996	0.561283	0.684777	0.129253
-0.12164	3.169325	-0.21344	1.257311	0.574856	0.673053	0.259672
-0.25523	3.196471	1.663337	1.286943	0.568069	0.674802	0.567785
-0.11391	3.25755	-1.787	1.257311	0.568069	0.67645	0.056032
-0.16058	3.27791	0.470037	1.274608	0.568069	0.677997	0.221018
-0.32525	3.311843	1.642489	1.274117	0.568069	0.672655	1.340505
-0.26645	3.386495	-0.39429	1.23996	0.574856	0.673998	-0.2538
-0.31544	3.440787	0.237349	1.274117	0.588429	0.661667	0.582958
-0.23898	3.515439	0.212291	1.184394	0.588429	0.662807	0.323613
-0.31949	3.562945	-0.49416	1.295787	0.581642	0.670633	-0.13593
-0.2611	3.644384	0	1.237395	0.581642	0.671572	0.192179
-0.25404	3.698677	-0.44233	1.257979	0.588429	0.665625	-0.02601
-0.25199	3.752969	0.442331	1.228285	0.595215	0.65958	0.652511
-0.27599	3.800475	0.044523	1.249503	0.561283	0.68737	0.09892
-0.22579	3.854768	-0.59557	1.236523	0.595215	0.660771	-0.19108
-0.31235	3.902273	0.989861	1.261218	0.595215	0.661225	1.182857
-0.21219	3.949779	-0.88114	1.216129	0.534136	0.722664	-1.23461
-0.29911	3.990499	0.037213	1.300729	0.540923	0.709361	-0.07326
-0.26745	4.065151	0.899281	1.212863	0.588429	0.668823	0.790521
-0.34915	0.318968	-1.87556	1.276063	0.421052	0.42261	-1.70486
-0.17624	0.386834	-1.98664	1.28982	0.435521	0.423963	-1.86849
0.033582	0.43434	-0.5027	1.11142	0.457224	0.418529	-0.65678
-0.1007	0.427553	0.368285	1.222682	0.486162	0.406309	-0.33201
0.11607	0.536138	0.100218	0.999651	0.493397	0.414449	0.020123
-0.10257	0.563285	0.034199	1.21615	0.551272	0.390009	-0.51389
-0.07278	0.583644	-0.0342	1.188505	0.565741	0.391362	-0.41995
-0.10966	0.597218	-0.5532	1.259959	0.58021	0.392715	-0.9677
-0.20499	0.624364	1.072276	1.288272	0.594679	0.394068	-0.10682

<b>RS</b>	<b>DS</b>	<b>AL</b>	<b>SF</b>	<b>LLDS</b>	<b>RLDS</b>	<b>AF</b>
-0.08042	0.678656	-1.12594	1.296571	0.609148	0.395421	-1.18248
-0.12296	0.753308	0.825829	1.236542	0.645321	0.406267	0.07826
-0.1567	0.780455	0.484881	1.239979	0.645321	0.421193	-0.18681
-0.10597	0.814388	-1.09991	1.257996	0.65979	0.422546	-0.99165
-0.20853	0.841534	1.099905	1.291805	0.667024	0.430685	0.259195
-0.16264	0.895826	-0.51908	1.278357	0.695962	0.418465	-0.67629
-0.12633	0.936546	-0.58082	1.278357	0.717665	0.413032	-0.56121
-0.12837	0.970478	0.580825	1.244092	0.739369	0.407598	0.155373
-0.14455	0.997625	-0.58082	1.296571	0.761072	0.402165	-0.47108
-0.18987	1.045131	1.92803	1.221386	0.775541	0.403518	1.037401
-0.10525	1.079063	-1.42409	1.288272	0.797245	0.398084	-0.91731
-0.04137	1.112996	1.076885	1.157094	0.811714	0.399437	0.54785
-0.22103	1.058704	-0.17313	1.296571	0.876824	0.383137	-0.10869
-0.05095	1.289447	-0.79163	1.21615	0.912996	0.393983	-0.64582
-0.18059	1.30302	1.077126	1.278463	0.927465	0.395336	0.562636
0.032376	1.411605	-1.11132	1.259959	0.978107	0.362756	-1.02571
-0.13698	1.493044	1.400108	1.165198	1.028748	0.389881	0.778574
-0.26522	1.608415	-1.99115	1.212885	1.06492	0.3858	-0.72493
-0.23902	1.655921	-1.2888	1.267232	1.072155	0.39394	-0.63963
-0.36968	1.703427	1.526659	1.239979	1.101093	0.38172	0.556978
-0.20943	1.764506	-1.23786	1.157094	1.115562	0.383073	-0.375
-0.53674	1.798439	1.99115	1.297457	1.130031	0.384426	0.058052
-0.22915	1.900238	-1.56828	1.212885	1.151734	0.378993	-0.50081
-0.42018	1.940957	1.622947	1.239979	1.158968	0.387132	0.263232
-0.08751	2.008823	-1.33461	1.115721	1.173437	0.388485	-0.33726
-0.26849	2.090261	-1.99115	1.21615	1.195141	0.397978	-0.36611
-0.28058	2.137767	0.245699	1.212885	1.20961	0.399331	0.17856
-0.32765	2.185273	0	1.259959	1.216844	0.407471	-0.03867
-0.2868	2.246352	0.992164	1.157094	1.231313	0.408824	0.406761
-0.31717	2.287072	-0.90593	1.244092	1.245782	0.410177	-0.14942
-0.07278	2.341364	-1.02084	1.188505	1.267486	0.404743	-0.50524
-0.38382	2.354937	1.491109	1.28135	1.281955	0.406096	0.15459
-0.28486	2.429589	-0.47027	1.211775	1.253017	0.448169	-0.12508
-0.2747	2.477095	0.905926	1.145002	1.253017	0.463095	0.400923
-0.18718	0.318968	0	1.228305	0.772263	0.517184	-0.24804
-0.23241	0.352901	0.918311	1.21615	0.793981	0.495466	0.337023
-0.2251	0.393621	0.909069	1.152022	0.802127	0.48732	0.479491
-0.22804	0.427553	-0.90907	1.211775	0.803486	0.485961	-0.91036
-0.27622	0.468273	0	1.259959	0.798058	0.491389	-0.20136



RS	DS	AL	SF	LLDS	RLDS	AF
-0.14862	0.522565	0.909069	1.075539	0.779057	0.51039	0.669752
-0.1524	0.542925	-0.84222	1.131953	0.780416	0.509031	-0.89458
-0.13624	0.563285	-1.53572	1.211775	0.795348	0.494099	-1.2286
-0.21799	0.590431	1.291698	1.150295	0.78992	0.499527	0.697951
-0.16187	0.624364	-1.2917	1.237414	0.784493	0.504954	-1.09503
-0.09377	0.658297	0.757203	1.121984	0.785851	0.503595	0.458752
-0.23222	0.67187	0.778514	1.211775	0.793997	0.49545	0.302822
-0.20347	0.712589	0	1.183026	0.788569	0.500878	-0.14666
-0.218	0.746522	-0.83142	1.249521	0.783141	0.506305	-0.82685
-0.06731	0.787241	-1.34721	1.183026	0.80486	0.484587	-1.06262
0.038439	0.800814	-0.58082	1.113583	0.799432	0.490015	-0.54739
-0.05906	0.794028	-0.08114	1.21615	0.794005	0.495442	-0.13511
-0.05579	0.807601	0	1.212885	0.80215	0.487297	-0.07895
-0.1178	0.821174	1.580934	1.113583	0.789936	0.499511	0.977731
0.005071	0.841534	-1.58093	1.152022	0.784508	0.504939	-1.15349
0.068696	0.841534	-0.43716	1.115721	0.785867	0.50358	-0.36685
-0.03371	0.827961	-0.30853	1.237414	0.787226	0.502221	-0.25796
0.031411	0.841534	0.243117	1.157094	0.788585	0.500862	0.136025
0.062589	0.834747	-0.39007	1.150295	0.79673	0.492717	-0.30517
0.005071	0.821174	0.892655	1.152022	0.791302	0.498145	0.4821
0.060681	0.821174	-1.27118	1.175861	0.779088	0.510359	-0.84193
0.004375	0.800814	0.326283	1.211775	0.800807	0.48864	0.127204
0.120822	0.800814	-0.32628	1.115721	0.795379	0.494068	-0.27131
0.102566	0.773668	0.326283	1.113583	0.796738	0.492709	0.212121
-0.02438	0.753308	0.442315	1.212885	0.79131	0.498137	0.169464
-0.14979	0.468273	0	1.296571	0.77778	0.749196	-0.23624
-0.08712	0.536138	-0.05616	1.237414	0.777106	0.749871	-0.12781
-0.28623	0.427553	-1.11403	1.300654	0.781869	0.745108	-1.6169
-0.29781	0.508992	0.236427	1.297457	0.767621	0.759356	-0.16846
-0.24038	0.590431	-0.90471	1.296571	0.773733	0.753244	-1.26502
-0.36379	0.658297	1.982191	1.296097	0.773058	0.753918	0.309873
-0.31777	0.760095	-0.82283	1.301501	0.785957	0.74102	-1.41283
-0.35751	0.84832	0.822832	1.28982	0.778496	0.748481	0.077208
-0.26427	0.922973	-1.5874	1.295795	0.784608	0.742369	-1.92354
-0.30528	0.990838	0.831416	1.284841	0.77036	0.756617	0.093439
-0.31701	1.051917	0	1.296571	0.790045	0.736932	-0.57325
-0.26972	1.126569	-0.83142	1.301244	0.78937	0.737606	-1.31292
-0.24785	1.201222	0.052903	1.276063	0.795482	0.731494	-0.41456
-0.34891	1.255514	1.288797	1.296571	0.794808	0.732169	0.101425

<b>RS</b>	<b>DS</b>	<b>AL</b>	<b>SF</b>	<b>LLDS</b>	<b>RLDS</b>	<b>AF</b>
-0.24454	1.336953	-1.73649	1.300732	0.794133	0.732843	-1.86761
-0.21538	1.411605	-0.26711	1.288272	0.786672	0.740305	-0.64692
-0.24163	1.465898	1.233667	1.237414	0.785997	0.740979	0.371782
-0.10566	1.513403	-1.91897	1.221386	0.778536	0.74844	-1.22725
-0.21391	1.533763	0.519081	1.297188	0.784648	0.742328	-0.11251
-0.18457	1.615202	-0.45027	1.295995	0.79076	0.736216	-0.70821
-0.12211	1.676281	-0.16903	1.244092	0.783299	0.743677	-0.26931
-0.30546	1.703427	1.019186	1.301244	0.789411	0.737565	-0.09353
-0.18241	1.784866	-1.88477	1.295995	0.795523	0.731454	-1.49049
-0.21175	1.839158	0.608706	1.287291	0.788062	0.738915	-0.10067
-0.18583	1.893451	-0.31971	1.28135	0.780601	0.746376	-0.63547
-0.10023	1.940957	-0.98516	1.257328	0.793499	0.733477	-0.67128
-0.13493	1.968103	0.081141	1.286954	0.779252	0.747725	-0.19228
-0.2285	2.008823	1.266125	1.301393	0.765004	0.761973	0.254842
-0.16444	2.083475	-0.0424	1.239979	0.737183	0.789794	-0.376
-0.22192	2.117408	0	1.297457	0.736508	0.790468	-0.57241
-0.17126	2.178487	-0.60871	1.284841	0.74262	0.784356	-0.75268
-0.19177	2.205633	0.288998	1.287291	0.789452	0.737525	-0.30547
-0.15278	2.259925	-0.3232	1.268497	0.781991	0.744986	-0.51103
-0.20581	2.348151	0.642905	1.28135	0.808462	0.718514	-0.11941
-0.18027	2.395657	-0.64291	1.295995	0.821361	0.705616	-0.8504
-0.2904	1.011198	0	1.274117	0.278249	0.856815	-0.32258
-0.25909	1.072277	-0.71164	1.287279	0.386834	0.860884	-1.0427
-0.12384	1.133356	-1.95328	1.274117	0.393621	0.858077	-1.92473
-0.18797	1.167289	0.587396	1.301529	0.393621	0.861965	0.244201
-0.1423	1.228368	-0.5027	1.287279	0.41398	0.838617	-0.71375
-0.08292	1.275874	-1.08615	1.295787	0.420767	0.835538	-1.22244
-0.07442	1.309807	0	1.287279	0.427553	0.839155	-0.16439
-0.07442	1.336953	0	1.287279	0.318968	0.835895	-0.16599
-0.06421	1.364099	-0.37855	1.300729	0.312182	0.846118	-0.51308
-0.1111	1.404818	0.970948	1.286943	0.318968	0.842677	0.440994
-0.07966	1.445538	-0.64464	1.295787	0.318968	0.845932	-0.77861
-0.10808	1.479471	0.442334	1.296564	0.427553	0.849097	0.101177
-0.05858	1.52019	-0.82364	1.29854	0.441127	0.838598	-0.90749
-0.16872	1.547336	1.728476	1.300651	0.298609	0.868728	0.642955
-0.10557	1.608415	-0.90484	1.294052	0.298609	0.871622	-1.02742
-0.03848	1.642348	-1.10125	1.295787	0.332542	0.840492	-1.10999
-0.1227	1.662708	1.303553	1.29854	0.332542	0.843205	0.356754
0	1.710214	-1.78305	1.287279	0.305395	0.872973	-1.6333

<b>RS</b>	<b>DS</b>	<b>AL</b>	<b>SF</b>	<b>LLDS</b>	<b>RLDS</b>	<b>AF</b>
-0.12987	1.710214	1.083337	1.286943	0.312182	0.868718	0.229578
-0.08679	1.750933	-0.87493	1.29854	0.312182	0.871159	-0.95455
-0.00851	1.784866	-1.20841	1.295787	0.339328	0.846364	-1.1514
-0.05698	1.791653	1.054629	1.278343	0.339328	0.848624	0.224411
-0.10255	1.812012	0.591536	1.286943	0.318968	0.871154	0.038923
0.014451	1.845945	-1.7823	1.281337	0.325755	0.873233	-1.61913
-0.00485	1.845945	0.141513	1.291796	0.346115	0.848076	-0.04365
-0.14161	1.886664	1.963953	1.257311	0.318968	0.880647	-0.05431
-0.03759	1.920597	-1.9332	1.274117	0.318968	0.882274	-1.71277
-0.02079	1.93417	0	1.257311	0.318968	0.883811	-0.1935
-0.04708	1.940957	0.37855	1.259942	0.318968	0.885257	-0.00964
-0.00924	1.95453	-0.72184	1.267216	0.332542	0.87304	-0.80675
-0.05923	1.95453	0.721842	1.272097	0.352901	0.847159	0.074784
-0.04005	1.968103	-0.3925	1.277441	0.325755	0.882267	-0.52448
0.000668	1.97489	-0.32934	1.257311	0.359688	0.842632	-0.40182
-0.04622	1.97489	0.739596	1.257979	0.359688	0.843626	0.085954
0.01663	1.988463	-1.00567	1.257979	0.352901	0.851316	-1.03713
0.009961	1.981676	-0.38291	1.28858	0.366474	0.845343	-0.40624
-0.12717	1.968103	1.372287	1.277441	0.318968	0.886785	-0.02213
0.02402	1.995249	-1.41903	1.277441	0.359688	0.846698	-1.28763
-0.04092	1.97489	1.039023	1.277441	0.318968	0.887959	0.08875
0.000491	1.981676	-0.60937	1.274117	0.359688	0.847691	-0.63451
0	1.981676	0.276766	1.257311	0.359688	0.848052	0.01347
0.026042	1.981676	-0.69345	1.274608	0.359688	0.848323	-0.68886
-0.06883	1.961317	1.794691	1.257311	0.366474	0.841716	-0.18976
0.015201	1.981676	-1.62122	1.274608	0.373261	0.841806	-1.4178
0.018738	1.97489	0.220164	1.257311	0.359688	0.848591	0.011425
-0.01467	1.968103	0.257711	1.274608	0.366474	0.8485	-0.01422
0.01663	1.97489	-0.23466	1.257979	0.359688	0.855105	-0.31371
-0.01022	1.968103	0	1.284829	0.359688	0.848046	-0.15184
0.001932	1.968103	-0.02305	1.274117	0.332542	0.881616	-0.17585
0	1.968103	0.030907	1.274117	0.332542	0.881163	-0.14029
-0.01418	1.968103	0.226803	1.274117	0.332542	0.88062	-0.04123
0.02167	1.97489	-0.57353	1.274117	0.332542	0.879986	-0.63216
-0.01614	1.961317	0.604944	1.274117	0.325755	0.879261	0.102436
-0.04138	1.968103	0.288296	1.281337	0.332542	0.878446	-0.05188
0.009918	1.981676	-0.77282	1.278343	0.359688	0.850395	-0.77185
0	1.97489	0.195395	1.276049	0.325755	0.869759	-0.0348
-0.00954	1.97489	0.577424	1.249503	0.346115	0.855099	0.105623

<b>RS</b>	<b>DS</b>	<b>AL</b>	<b>SF</b>	<b>LLDS</b>	<b>RLDS</b>	<b>AF</b>
0.184444	1.97489	-0.27761	1.072867	0.318968	0.881069	-0.35803
0.214076	1.93417	-0.47412	1.072867	0.312182	0.879802	-0.55124
0.229118	1.879878	0.474118	1.028192	0.312182	0.878444	0.131547
0.201742	1.832372	-0.27677	1.072867	0.312182	0.876996	-0.30121
0.20125	1.784866	0.007858	1.072867	0.305395	0.875458	0.004187
0.947791	1.601629	-0.02225	1.301503	0.675073	0.682239	0.194895
1.104125	1.608415	0.523699	0.93229	0.678693	0.678619	0.451515
1.581886	1.574483	-1.00145	0.870282	0.683897	0.673416	-0.69159
1.399314	1.52019	0.39009	0.877467	0.682314	0.674999	0.519896
1.349881	1.479471	0	0.9269	0.687517	0.669796	0.034426
1.166132	1.445538	0.611363	0.870282	0.685934	0.671379	0.605779
1.211864	1.404818	-0.5237	1.028192	0.684351	0.672962	-0.21638
0.97377	1.377672	1.076896	0.870282	0.682767	0.674545	0.987919
1.02309	1.343739	0.034199	0.809683	0.681184	0.676128	0.134149
1.104125	1.30302	-0.5874	0.93229	0.679601	0.677711	-0.3107
1.114282	1.269087	-0.02763	0.93229	0.684804	0.672508	0.065358
0.905873	1.235154	0.615025	0.9269	0.690008	0.667305	0.567898
1.176289	1.208008	-0.61503	0.870282	0.688425	0.668888	-0.44937
0.819448	1.167289	1.26612	0.809683	0.686841	0.670471	1.218667
0.771528	1.133356	-0.0424	0.870282	0.685258	0.672054	0.029992
0.771528	1.10621	0	0.870282	0.683675	0.673637	0.070331
0.75885	1.06549	-0.66189	0.870282	0.692499	0.664814	-0.56637
0.75885	1.038344	0	0.870282	0.690915	0.666397	0.051926
0.884028	1.011198	0	0.745103	0.689332	0.66798	0.055563
0.56968	0.970478	0.661891	0.870282	0.687749	0.669563	0.623464
0.75885	0.950119	-0.66189	0.870282	0.686166	0.671147	-0.62051
0.56968	0.922973	0.661891	0.870282	0.684583	0.67273	0.648126
0.351567	0.902613	0.831391	0.870282	0.683	0.674313	0.693567
0.374994	0.88904	0	0.846855	0.681416	0.675896	0.047181
-0.03065	0.86868	1.633114	0.870282	0.693406	0.663906	1.079567
0.672849	0.86868	-1.61825	0.809683	0.691823	0.665489	-1.50589
0.351567	0.841534	0.985132	0.870282	0.69024	0.667073	0.826904
0.19842	0.827961	0.842187	0.819788	0.688657	0.668656	0.679757
1.291311	0.814388	-1.57396	0.745103	0.69386	0.663452	-1.22514
0.429019	0.766882	-1.69996	0.809683	0.683907	0.673405	-1.60924
0.160905	0.746522	0.8228	0.877467	0.682324	0.674988	0.669574
-0.00522	0.739735	0.992119	0.819788	0.687527	0.669785	0.731002
0.516425	0.732949	-1.74807	0.705424	0.679158	0.678155	-1.69722
-0.03065	0.699016	1.633114	0.870282	0.684361	0.672951	1.150921

RS	DS	AL	SF	LLDS	RLDS	AF
0.418915	0.699016	-1.69996	0.819788	0.682778	0.674535	-1.63244
-0.463	0.678656	1.817782	0.870282	0.687981	0.669331	1.634297
0.138024	0.665083	1.733204	0.676542	0.684815	0.672498	1.313237
0.147925	0.644723	0	0.870282	0.681648	0.675664	0.060091
0.004883	0.637937	0.905885	0.809683	0.680065	0.677247	0.678203
0.160905	0.63115	-0.99212	0.877467	0.678482	0.67883	-0.86976
0.273104	0.624364	0.086234	0.745103	0.683685	0.673627	0.141325
-0.03065	0.604004	0.790927	0.870282	0.682102	0.67521	0.59743
0.228688	0.604004	-0.87716	0.809683	0.680519	0.676793	-0.81562
0.069463	0.590431	0.992119	0.745103	0.678936	0.678377	0.773178
0.23417	0.576858	-0.43564	0.676542	0.677353	0.67996	-0.37146
0.361236	0.556498	-1.37928	0.877467	0.67577	0.681543	-1.39087
0.101029	0.542925	1.379281	0.809683	0.680973	0.67634	1.095012
-0.15635	0.542925	0.114958	0.970918	0.718526	0.638787	0.232106
0.412165	0.536138	-1.74807	0.809683	0.68301	0.674303	-1.71505
0.14074	0.515779	0.842187	0.877467	0.681427	0.675886	0.726202
0.033245	0.508992	0.470247	0.877467	0.68663	0.670682	0.424126
-0.08726	0.508992	0.320681	0.9269	0.67826	0.679052	0.273564
0.114207	0.855107	-1.74108	0.9269	0.802576	0.520803	-1.63306
0.224781	0.841534	-1.85541	0.93229	0.805743	0.517637	-1.64545
0.07273	0.807601	1.519052	0.9269	0.802122	0.521257	1.310106
-0.0918	0.570071	1.210868	1.075517	0.804835	0.518545	0.900226
0.184671	1.343739	1.41405	1.028192	0.803927	0.519452	1.143998
-0.03176	1.336953	-1.74108	1.072867	0.796687	0.526693	-1.19497
0.081554	1.343739	-1.85541	1.075517	0.799853	0.523526	-1.18956
-0.11393	1.330166	1.519052	1.11356	0.796233	0.527147	1.041346
0.88904	0.88904	0	1.031499	0.506449	0.781508	0.08306
-2.62075	0.970478	1.591226	1.212863	0.501696	0.776756	1.239678
2.055147	1.038344	-1.0168	1.212863	0.50079	0.77585	-1.1596
-0.86719	0.997625	1.864811	1.295787	0.496944	0.772003	1.630365
0.520029	1.038344	0.518315	1.212863	0.499884	0.774943	0.567975
1.570232	1.051917	-0.51831	1.257311	0.496038	0.771097	-0.5345
1.072277	1.072277	0	1.237395	0.498978	0.774037	0.155155
1.151269	1.08585	-0.06542	1.257979	0.495131	0.770191	0.077878
0.603618	1.10621	0.502592	1.236523	0.491285	0.766344	0.556142
1.733959	1.126569	-0.60739	1.244073	0.494225	0.769285	-0.64624
4.192412	1.235154	-1.95726	1.237395	0.486533	0.761592	-1.88739
5.018356	1.316593	-1.70176	1.221365	0.482686	0.757746	-1.77322
1.031558	1.031558	0	1.237395	0.488567	0.763626	0.166642

<b>RS</b>	<b>DS</b>	<b>AL</b>	<b>SF</b>	<b>LLDS</b>	<b>RLDS</b>	<b>AF</b>
-2.46466	1.126569	1.591226	1.237395	0.490601	0.76566	1.414973
2.156945	1.140143	-1.0168	0.983715	0.489694	0.764754	-1.00209
-0.80611	1.058704	1.864811	0.983715	0.499421	0.774481	1.721
0.506456	1.024771	0.518315	0.979537	0.495575	0.770634	0.497532
1.51594	0.997625	-0.51831	1.11356	0.498515	0.773574	-0.54673
0.984052	0.984052	0	0.979537	0.501455	0.776515	0.106131
1.015538	0.950119	-0.06542	0.979537	0.504395	0.779455	0.010651
0.413594	0.916186	0.502592	0.983715	0.507336	0.782395	0.491289
1.49643	0.88904	-0.60739	1.014406	0.503489	0.778549	-0.76763
0.306927	0.916186	0	0.877467	0.922973	1.27562	0.328558
0.120501	0.86868	0.518315	1.031499	0.929759	1.255706	0.930651
0.139997	0.84832	-0.97382	1.072867	0.943332	1.235853	-0.13027
0.034122	0.821174	0.455509	1.150273	0.956905	1.222846	0.574237
0.001727	0.814388	0.518315	1.150273	0.956905	1.209899	0.58055
0.035109	0.814388	-0.08114	1.121961	0.956905	1.197013	0.379135
0.061481	0.807601	-0.87493	1.150273	0.970478	1.170614	0.027931
0.059754	0.794028	0	1.151999	0.977265	1.157849	0.389774
0.04476	0.780455	0.267539	1.150273	0.990838	1.138357	0.494141
0.067306	0.773668	0.192462	1.115698	0.990838	1.125712	0.572934
0.005071	0.760095	0.414928	1.151999	0.997625	1.106341	0.521027
0.034122	0.760095	-0.43717	1.150273	1.017984	1.08703	0.144537
0.059754	0.753308	-0.43776	1.151999	1.017984	1.074565	0.19893
0.034122	0.739735	0.437756	1.150273	1.024771	1.055375	0.560469
0.032395	0.732949	0	1.151999	1.031558	1.043032	0.346873
0.001727	0.726162	0.518315	1.150273	1.038344	1.023962	0.573689
0.032395	0.726162	-0.51831	1.151999	1.045131	1.011739	0.095729
0.062591	0.719376	-0.45551	1.150273	1.051917	0.99279	0.190644
-0.03239	0.705803	0.973824	1.184394	1.058704	0.980687	0.606788
0.034122	0.712589	-0.51831	1.150273	1.06549	0.961859	0.094224
0	0.705803	0	1.184394	1.072277	0.949877	0.210169
0	0.705803	0.022245	1.183004	1.08585	0.931169	0.221524
0.032395	0.705803	-0.02225	1.151999	1.08585	0.919308	0.324273
0	0.699016	0	1.184394	1.092637	0.900721	0.204755
-0.02847	0.699016	0	1.212863	1.10621	0.882194	0.087809
0	0.705803	0.022245	1.183004	1.112996	0.863727	0.212301
-0.06243	0.705803	0.976678	1.184394	1.112996	0.852107	0.575877
-0.00139	0.719376	-0.97668	1.184394	1.126569	0.833761	-0.24863
-0.05213	0.719376	-0.02225	1.236523	1.133356	0.815475	-0.01592
0	0.732949	0	1.184394	1.133356	0.804036	0.186173

<b>RS</b>	<b>DS</b>	<b>AL</b>	<b>SF</b>	<b>LLDS</b>	<b>RLDS</b>	<b>AF</b>
-0.05975	0.732949	0.518315	1.211754	1.146929	0.785871	0.266014
-0.03239	0.746522	0	1.184394	1.160502	0.774552	0.126891
-0.05439	0.753308	-0.49607	1.237395	1.167289	0.756507	-0.26257
-0.02736	0.766882	-0.02225	1.211754	1.174075	0.738523	0.059579
-0.06086	0.773668	0.518315	1.212863	1.174075	0.727386	0.258353
-0.053	0.787241	-0.51831	1.237395	1.180862	0.709522	-0.28188
-0.0854	0.800814	0.518315	1.237395	1.194435	0.691718	0.181645
-0.06148	0.821174	0.027629	1.211754	1.201222	0.673975	0.016268
-0.10531	0.834747	-0.02763	1.257311	1.208008	0.663079	-0.13627
-0.08796	0.861893	0	1.23996	1.208008	0.652243	-0.07513
-0.0854	0.882253	0	1.237395	1.221581	0.627894	-0.06808
-0.07358	0.902613	-0.51831	1.257979	1.235154	0.610392	-0.36747
-0.10704	0.922973	0.545944	1.257311	1.241941	0.59295	0.17345
-0.10598	0.950119	-0.02763	1.257979	1.241941	0.582355	-0.11797
-0.08452	0.977265	0	1.236523	1.255514	0.565034	-0.0622
-0.12405	0.997625	0	1.276049	1.255514	0.55456	-0.11801
-0.10531	1.031558	0	1.257311	1.269087	0.530573	-0.08381
-0.10771	1.058704	0.027629	1.257979	1.275874	0.520219	-0.06491
-0.10598	1.08585	-0.02763	1.257979	1.28266	0.503139	-0.09109
-0.09021	1.112996	-0.51831	1.274608	1.289447	0.48612	-0.40099
-0.14161	1.140143	1.099141	1.257311	1.296233	0.46916	0.492507
-0.12434	1.174075	-0.5532	1.274608	1.296233	0.459048	-0.43437
-0.12261	1.208008	-0.02763	1.274608	1.30302	0.442209	-0.07155
-0.15891	1.241941	0.580826	1.274608	1.32338	0.425431	0.281165
-0.24592	1.28266	1.400085	1.274117	1.330166	0.408713	0.695742
-0.29651	1.336953	0.778489	1.276049	1.330166	0.398841	0.363371
-0.30323	1.398032	-0.06685	1.286943	1.350526	0.375457	0.001342
-0.38247	1.465898	1.699962	1.259942	1.357313	0.365707	0.788247
-0.35465	1.533763	-0.87716	1.286943	1.370886	0.342443	-0.12761
-0.36004	1.608415	0.086234	1.286943	1.370886	0.332813	0.173437
-0.34232	1.683068	-0.08623	1.274608	1.370886	0.323243	0.149009
-0.30455	1.750933	-0.8228	1.288261	1.384459	0.306947	-0.06733
-0.34722	1.676281	0.909035	1.274117	1.404818	0.283925	0.485829
-0.36038	1.655921	0	1.287279	1.404818	0.274536	0.273061
-0.33963	1.594842	-0.33192	1.287279	1.411605	0.258421	0.193105
-0.29507	1.560909	-0.51026	1.274608	1.418392	0.242366	0.06483
-0.34183	1.526977	0.755953	1.274117	1.425178	0.226372	0.508293
-0.30774	1.493044	-0.75595	1.287279	1.425178	0.217224	0.041619
-0.35465	1.459111	0.755953	1.286943	1.438751	0.20135	0.535928

RS	DS	AL	SF	LLDS	RLDS	AF
-0.30356	1.425178	-0.8228	1.287279	1.438751	0.192323	0.017458
-0.05227	0.536138	0	1.274117	0.875467	0.487652	-0.08097
-0.43497	0.597218	1.633114	1.274608	0.875467	0.479469	0.893521
-0.23977	0.67187	-0.79093	1.257979	0.882253	0.46456	-0.67292
-0.25591	0.732949	0	1.274117	0.895826	0.449711	-0.07489
0.181984	0.800814	-1.67358	1.257979	0.88904	0.441709	-0.84941
0.029333	0.84832	0.509899	1.274608	0.882253	0.440554	0.175161
0.287804	0.916186	-0.90468	1.263084	0.882253	0.439459	-0.44107
0.469977	0.963692	-0.59103	1.257979	0.909399	0.411278	-0.30971
-0.21832	1.017984	1.659388	1.236523	0.916186	0.396731	0.780202
-0.03546	1.072277	-0.84219	1.257311	0.916186	0.38903	-0.67089
0.013333	1.126569	-0.25964	1.274608	0.922973	0.38139	-0.24929
0.202568	1.187648	-0.57175	1.237395	0.916186	0.380597	-0.57835
0.79902	1.228368	-1.90038	1.237395	0.929759	0.366291	-1.73838
0.991315	1.248728	-0.54594	1.257979	0.936546	0.352045	-0.61336
1.207175	1.269087	-0.43776	1.236523	0.929759	0.351432	-0.32536
1.03643	1.275874	0.437756	1.212863	0.936546	0.344093	0.471047
1.479522	1.28266	-0.95486	1.237395	0.943332	0.330028	-0.90637
1.239305	1.28266	0.499353	1.212863	0.956905	0.316024	0.648365
1.389361	1.28266	-0.37855	1.257979	0.963692	0.302079	-0.48935
1.240883	1.289447	0.326306	1.236523	0.963692	0.301768	0.395964
1.871058	1.296233	-0.98826	1.237395	0.963692	0.294731	-0.90153
1.230834	1.296233	1.058255	1.212863	0.956905	0.294541	0.965105
1.403266	1.296233	-0.3963	1.244073	0.970478	0.280838	-0.5695
1.417765	1.296233	-0.01395	1.237395	0.990838	0.253622	0.052442
1.207175	1.296233	0.410253	1.236523	1.004411	0.240039	0.433796
1.434476	1.30302	-0.3963	1.212863	1.004411	0.233303	-0.55342
0.223007	0.610791	0	0.760708	0.562531	0.455453	-0.0921
0.401659	0.583644	-1.60877	0.745103	0.569379	0.448605	-1.89459
0.47373	0.549712	-0.05616	0.676542	0.569379	0.448605	-0.05593
0.091641	0.502206	1.013586	0.870282	0.569379	0.448605	0.592283
0.326211	0.468273	-1.75487	0.877467	0.569379	0.448605	-1.93967
0.161216	0.468273	1.75487	0.870282	0.576227	0.441758	0.855162
0.004178	0.447913	0.764543	0.979537	0.576227	0.441758	0.47404
0.140577	0.447913	-1.42643	0.93229	0.576227	0.441758	-1.48003
0.056618	0.427553	1.335468	0.870282	0.583074	0.43491	0.635149
-0.16619	0.43434	1.65908	0.9269	0.583074	0.43491	0.729911
0.298476	0.461486	-1.11623	0.9269	0.583074	0.43491	-1.37241
0.148616	0.420767	1.397755	0.9269	0.583074	0.43491	0.69668



<b>RS</b>	<b>DS</b>	<b>AL</b>	<b>SF</b>	<b>LLDS</b>	<b>RLDS</b>	<b>AF</b>
-0.05482	0.427553	1.168791	0.93229	0.583074	0.43491	0.612745
0.181797	0.43434	-0.79093	0.745103	0.589922	0.428062	-0.9187
-0.10207	0.41398	0.790927	0.979537	0.589922	0.428062	0.515605
0.052637	0.427553	-1.63311	0.9269	0.589922	0.428062	-1.53147
-0.03478	0.420767	1.312434	0.93229	0.589922	0.428062	0.622042
-0.10129	0.427553	-0.47025	1.028192	0.59677	0.421215	-0.4013
0.244494	0.441127	-1.12098	0.877467	0.59677	0.421215	-1.33669
-0.03243	0.407194	1.019147	1.028192	0.603617	0.414367	0.538522
-0.10957	0.441127	1.760912	0.870282	0.603617	0.414367	0.66626
0.052637	0.461486	1.362546	0.9269	0.610465	0.40752	0.635758
-0.04725	0.4547	0.755953	0.979537	0.610465	0.40752	0.480562
-0.0959	0.461486	0	1.028192	0.610465	0.40752	0.003169
0.170825	0.475059	-1.74108	0.870282	0.610465	0.40752	-1.7827
-0.09598	0.4547	0.985132	1.075517	0.610465	0.40752	0.471562
0.101292	0.468273	-0.77849	0.9269	0.617313	0.400672	-0.87133
-0.18464	0.4547	1.901405	1.031499	0.617313	0.400672	0.728922
-0.10207	0.481846	-0.4898	0.979537	0.62416	0.393824	-0.38356
0.113432	0.495419	-1.69996	0.870282	0.62416	0.393824	-1.69644
-0.09333	0.481846	0.066847	1.072867	0.62416	0.393824	-0.04889
-0.11343	0.495419	1.748072	0.983715	0.62416	0.393824	0.733062
0	0.508992	-1.74807	0.979537	0.631008	0.386977	-1.59185
-0.10925	0.508992	1.748072	0.979537	0.631008	0.386977	0.746109
0.05424	0.522565	-1.68276	0.983715	0.631008	0.386977	-1.62474
-0.05143	0.536138	1.690641	0.983715	0.637855	0.380129	0.762856
0.054823	0.542925	0	0.877467	0.631008	0.386977	-0.05894
-0.16619	0.536138	1.745314	0.9269	0.637855	0.380129	0.824479
0.118318	0.542925	1.613822	0.979537	0.644703	0.373281	0.80492
-0.03607	0.563285	-1.24067	0.983715	0.644703	0.373281	-1.18372
0.179842	0.590431	-0.83176	0.819788	0.651551	0.366434	-1.02261
0.114207	0.590431	-0.72549	0.9269	0.658398	0.359586	-0.8687
-0.08203	0.576858	1.297566	0.979537	0.658398	0.359586	0.718948
-0.00862	0.590431	-1.17454	0.979537	0.658398	0.359586	-1.1813
-0.04778	0.597218	-0.20474	1.031499	0.658398	0.359586	-0.23788
0.222942	0.604004	-1.50023	0.979537	0.665246	0.352738	-1.72122
0.072112	0.604004	0.919997	1.072867	0.665246	0.352738	0.542239
0.161216	0.597218	1.815687	0.870282	0.665246	0.352738	1.027139
-0.26318	0.576858	1.549045	1.072867	0.665246	0.352738	0.7868
-0.04866	0.610791	-1.71765	1.028192	0.665246	0.352738	-1.52699
0.143598	0.617577	-0.98513	0.89751	0.665246	0.352738	-1.16825

RS	DS	AL	SF	LLDS	RLDS	AF
0.085159	0.597218	-0.37741	0.979537	0.678941	0.339043	-0.5149
0.103719	0.597218	-0.29695	0.979537	0.678941	0.339043	-0.43935
0.051962	0.583644	0.828108	0.979537	0.678941	0.339043	0.476586
0.113575	0.576858	-0.98581	0.979537	0.678941	0.339043	-1.18773
-0.1046	0.576858	1.659388	1.031499	0.678941	0.339043	0.801407
0.004178	0.590431	-0.90903	0.979537	0.685789	0.332195	-1.03988
0.061571	0.590431	-0.91828	0.979537	0.685789	0.332195	-1.09503
0.047784	0.583644	0.153741	0.983715	0.685789	0.332195	0.003915
-0.10207	0.576858	1.464505	0.979537	0.692637	0.325348	0.762345
0.047784	0.590431	-1.46451	0.983715	0.692637	0.325348	-1.60697
0.09333	0.583644	-0.66189	0.979537	0.699484	0.3185	-0.89069
0.091802	0.570071	-0.0424	0.983715	0.699484	0.3185	-0.2236
-0.04866	0.556498	1.535676	1.028192	0.706332	0.311653	0.72845
0	0.563285	0	0.979537	0.706332	0.311653	-0.17153
0.047324	0.563285	-1.53568	1.028192	0.706332	0.311653	-1.7479
0.093769	0.556498	-0.74312	1.028192	0.706332	0.311653	-1.00937
0.009609	0.556498	1.29366	1.031499	0.706332	0.311653	0.649473
0.004178	0.556498	0.918285	0.979537	0.713179	0.304805	0.485237
-0.14597	0.556498	0.909035	1.072867	0.713179	0.304805	0.412261
-0.03441	0.583644	-1.82732	1.075517	0.713179	0.304805	-1.86691
0.022631	0.590431	-0.87025	1.072867	0.720027	0.297957	-1.14948
-0.07324	0.590431	1.533888	1.072867	0.720027	0.297957	0.704652
-0.04732	0.610791	-0.457	1.075517	0.720027	0.297957	-0.68886
-0.01934	0.617577	-0.44768	1.075517	0.726875	0.29111	-0.70941
-0.05739	0.624364	0	1.11356	0.733722	0.284262	-0.28588
-0.03804	0.637937	-0.3095	1.11356	0.733722	0.284262	-0.56047
0.087496	0.65151	-1.35755	1.072867	0.733722	0.284262	-1.6362
-0.1888	0.644723	1.735412	1.115698	0.733722	0.284262	0.770368
-0.13616	0.67187	-0.84219	1.115698	0.74057	0.277414	-1.08385
-0.33048	0.692229	1.555983	1.150273	0.747418	0.270567	0.750729
-0.1954	0.739735	-0.92287	1.072867	0.747418	0.270567	-1.24876
-0.21819	0.766882	-0.32068	1.115698	0.747418	0.270567	-0.54681
-0.18127	0.814388	-0.55648	1.11356	0.754265	0.263719	-0.75637
-0.26165	0.841534	0.992119	1.13193	0.754265	0.263719	0.448475
-0.41526	0.875467	1.002863	1.160363	0.733722	0.284262	0.508939
-0.22601	0.922973	-1.62802	1.072867	0.767961	0.250024	-1.76011
-0.37059	0.956905	1.62802	1.115698	0.767961	0.250024	0.909226
-0.21605	1.004411	-1.4385	1.11356	0.774808	0.243176	-1.33553
-0.25308	1.051917	1.24355	1.072867	0.774808	0.243176	0.59905

<b>RS</b>	<b>DS</b>	<b>AL</b>	<b>SF</b>	<b>LLDS</b>	<b>RLDS</b>	<b>AF</b>
-0.17536	1.08585	-1.24355	1.072867	0.774808	0.243176	-1.2361
-0.33041	1.112996	1.438501	1.075517	0.774808	0.243176	0.709825
-0.20258	1.153716	-1.00286	1.072867	0.781656	0.236328	-0.94135
-0.22182	1.180862	0.969583	1.031499	0.781656	0.236328	0.444524
-0.10129	1.208008	-1.87547	1.028192	0.788504	0.229481	-1.61939
-0.26318	1.235154	1.875467	1.072867	0.788504	0.229481	0.701666
-0.32277	1.269087	1.668147	1.028192	0.788504	0.229481	0.497796
-0.27079	1.309807	-0.88453	1.031499	0.788504	0.229481	-0.49866
-0.15791	0.658297	0	1.028192	0.642779	0.375206	0.040837
-0.16393	0.678656	0.807911	0.983715	0.64504	0.372944	0.628745
-0.10925	0.699016	-0.80791	0.979537	0.654089	0.363896	-0.66235
-0.2084	0.712589	0.807911	1.028192	0.65635	0.361634	0.614265
-0.08005	0.739735	-0.43307	0.9269	0.658612	0.359372	-0.3555
-0.10207	0.753308	-0.4898	0.979537	0.660874	0.357111	-0.38888
-0.18848	0.766882	-0.1107	1.072867	0.663135	0.354849	-0.13081
0.09598	0.800814	-1.05809	0.979537	0.665397	0.352587	-1.11128
-0.19805	0.814388	1.168791	1.075517	0.667659	0.350326	0.775991
-0.20258	0.882253	0	1.072867	0.599792	0.418193	-0.01618
-0.10129	0.909399	-0.90588	1.028192	0.602053	0.415931	-0.64531
-0.29377	0.929759	1.713796	1.11356	0.604315	0.413669	1.568229
-0.0918	0.970478	-1.62283	1.075517	0.606577	0.411408	-0.95894
-0.19805	0.984052	1.699962	1.075517	0.615625	0.402359	1.601527
-0.20258	1.011198	0.114958	1.072867	0.6111	0.406884	0.093614
-0.10195	1.038344	-1.61018	1.072867	0.613362	0.404623	-0.8548
0.01837	1.058704	-1.57619	1.11356	0.615624	0.402361	-0.73824
-0.15669	1.058704	1.212122	1.212863	0.617885	0.400099	0.632057
-0.0918	1.072277	1.159327	1.075517	0.674439	0.343545	0.914311
0.008401	1.08585	-1.21194	1.11356	0.683488	0.334497	-0.56719
-0.12985	1.10621	1.211945	1.11356	0.685749	0.332235	0.909504
-0.04137	1.126569	-0.76454	1.072867	0.688011	0.329973	-0.45543
-0.01669	1.133356	-0.39478	1.072867	0.690273	0.327712	-0.29505
-0.22143	1.140143	0.96653	1.217196	0.692534	0.32545	0.597101
-0.10374	1.167289	-0.51884	1.13193	0.64729	0.370694	-0.22001
-0.1888	1.180862	1.620677	1.115698	0.629192	0.388792	1.748852
-0.05739	1.208008	-1.06836	1.11356	0.624667	0.393317	-0.35983
-0.05953	1.221581	0	1.115698	0.626929	0.391055	0.006087
-0.11596	1.235154	0.241042	1.157071	0.635977	0.382007	0.144004
-0.09333	1.255514	0.985132	1.072867	0.624666	0.393319	0.839485
-0.01669	1.269087	-1.22617	1.072867	0.626928	0.391057	-0.45215

RS	DS	AL	SF	LLDS	RLDS	AF
-0.16828	1.275874	1.159327	1.151999	0.629189	0.388795	1.003519
-0.06201	0.942518	0.114958	0.93229	0.529352	0.468273	0.061858
0.051425	0.949304	-1.81492	0.93229	0.536138	0.468273	-1.42412
-0.10925	0.942518	1.81492	0.979537	0.536138	0.461486	1.572508
-0.21798	0.956091	-0.99212	1.150273	0.536138	0.461486	-0.91775
-0.05264	0.983237	0.086234	0.979537	0.529352	0.468273	0.021111
-0.08915	0.990024	-0.90903	1.072867	0.529352	0.468273	-0.69216
-0.0918	1.003597	0	1.075517	0.522565	0.475059	-0.05972
-0.1529	1.01717	-0.76454	1.184394	0.522565	0.468273	-0.68897
-0.07476	1.044316	-0.70429	1.150273	0.522565	0.468273	-0.50065
0.104599	1.057889	0.704286	0.9269	0.515779	0.475059	0.473379
-0.07648	1.044316	-0.70429	1.151999	0.508992	0.475059	-0.51197
0.040181	1.057889	-0.6429	1.075517	0.508992	0.475059	-0.52947
0.00265	1.051103	0.642898	1.072867	0.508992	0.468273	0.381542
0.038439	1.051103	-1.22372	1.11356	0.502206	0.475059	-0.78131
0	1.044316	0.580826	1.115698	0.495419	0.475059	0.304902
0.036302	1.044316	-0.58083	1.115698	0.495419	0.475059	-0.41177
0.038439	1.03753	0	1.11356	0.495419	0.468273	-0.04406
0.068696	1.030743	-0.51831	1.115698	0.488633	0.475059	-0.37101
0.038439	1.01717	0.518315	1.11356	0.488633	0.468273	0.275522
0.123807	1.010383	0	1.028192	0.481846	0.475059	-0.10914
0.036302	0.990024	0	1.115698	0.475059	0.475059	-0.0494
0.038439	0.983237	0	1.11356	0.475059	0.475059	-0.05066
0.070834	0.976451	-0.51831	1.11356	0.468273	0.481846	-0.4099
0.076483	0.962878	0.518315	1.075517	0.461486	0.488633	0.310652
0.068696	0.949304	-0.51831	1.115698	0.461486	0.481846	-0.42107
0.111527	0.935731	0	1.072867	0.461486	0.481846	-0.06402
0.041373	0.915372	0.437173	1.115698	0.4547	0.488633	0.243145
0.110137	0.908585	-0.41493	1.072867	0.447913	0.481846	-0.41147
0.068696	0.888225	-0.02225	1.115698	0.447913	0.488633	-0.06176
0.070834	0.874652	0	1.11356	0.447913	0.481846	-0.04689
0.108877	0.861079	0	1.075517	0.441127	0.488633	-0.05649
0.068696	0.840719	0	1.115698	0.441127	0.488633	-0.04875
0.123807	0.827146	0.518315	1.028192	0.441127	0.481846	0.338708
0.099303	0.806787	-0.97382	1.11356	0.43434	0.488633	-0.89631
0.108877	0.813573	0.455509	1.075517	0.427553	0.495419	0.304923
0.038439	0.833933	0.518315	1.11356	0.427553	0.495419	0.306123
0.081554	0.840719	-0.08114	1.075517	0.420767	0.495419	-0.12591
0.136237	0.854293	-0.87493	1.075517	0.41398	0.488633	-0.84109

RS	DS	AL	SF	LLDS	RLDS	AF
0.098194	0.881439	0	1.11356	0.420767	0.495419	-0.03195
0.122166	0.901798	0.267539	1.072867	0.420767	0.502206	0.15732
0.067306	0.922158	0.192462	1.115698	0.400407	0.508992	0.097865
0.041373	0.935731	0.414928	1.115698	0.407194	0.502206	0.231305
0.111527	0.942518	-0.43717	1.072867	0.400407	0.502206	-0.4256
0.138887	0.962878	-0.43776	1.072867	0.393621	0.502206	-0.4254
0.068696	0.990024	0.437756	1.115698	0.393621	0.515779	0.256353
0.108877	1.003597	0	1.075517	0.393621	0.508992	-0.05962
0.079132	1.023957	0.518315	1.072867	0.393621	0.508992	0.307446
0.070834	1.03753	-0.51831	1.11356	0.386834	0.515779	-0.40549
0.097166	1.051103	-0.45551	1.115698	0.380048	0.522565	-0.34628
0.079132	1.071463	0.973824	1.072867	0.380048	0.515779	0.61541
0.068696	1.085036	-0.51831	1.115698	0.373261	0.522565	-0.38736
0.108877	1.098609	0	1.075517	0.373261	0.522565	-0.06283
0.067306	1.118968	0.022245	1.115698	0.373261	0.522565	-0.01379
0.068696	1.132542	-0.02225	1.115698	0.366474	0.529352	-0.04144
0.068696	1.146115	0	1.115698	0.366474	0.536138	-0.02703
0.111527	1.159688	0	1.072867	0.366474	0.529352	-0.06852
0.067306	1.180048	0.022245	1.115698	0.359688	0.542925	-0.01243
0.008401	1.193621	0.976678	1.11356	0.366474	0.536138	0.54383
0.067306	1.193621	-0.97668	1.115698	0.352901	0.536138	-0.64871
0.070834	1.207194	-0.02225	1.11356	0.359688	0.536138	-0.03941
0.068696	1.220767	0	1.115698	0.352901	0.549712	-0.02507
0.036302	1.23434	0.518315	1.115698	0.352901	0.542925	0.268544
0.038439	1.241127	0	1.11356	0.352901	0.542925	-0.03486
0.069443	1.247913	-0.49607	1.11356	0.396115	0.549712	-0.34814
0.068696	1.261486	-0.02225	1.115698	0.396115	0.556498	-0.03826
0.038439	1.275059	0.518315	1.11356	0.396115	0.549712	0.26466
0.068696	1.281846	-0.51831	1.115698	0.389328	0.556498	-0.35563
0	1.295419	0.518315	1.151999	0.389328	0.549712	0.255114
0.034575	1.295419	0.027629	1.115698	0.389328	0.549712	-0.01487
0	1.302206	-0.02763	1.151999	0.382542	0.563285	-0.01993
0.036302	1.302206	0	1.115698	0.382542	0.563285	-0.0352
0	1.308992	0	1.151999	0.382542	0.556498	0.001376
0.034122	1.308992	-0.51831	1.150273	0.382542	0.556498	-0.25742
0	1.315779	0.545944	1.150273	0.375755	0.563285	0.270022
0.036302	1.315779	-0.02763	1.115698	0.375755	0.563285	-0.05199
0	1.322565	0	1.151999	0.375755	0.563285	-0.00031
0	1.322565	0	1.151999	0.375755	0.563285	-0.00031

RS	DS	AL	SF	LLDS	RLDS	AF
0.001727	1.322565	0	1.150273	0.375755	0.556498	0.003002
0	1.322565	0.027629	1.150273	0.368968	0.563285	0.012703
0	1.322565	-0.02763	1.151999	0.368968	0.570071	-0.02056
0.032395	1.322565	-0.51831	1.151999	0.368968	0.570071	-0.279
-0.06731	1.329352	1.099141	1.183004	0.368968	0.563285	0.544005
0.036712	1.315779	-0.5532	1.11356	0.362182	0.570071	-0.42792
0	1.322565	-0.02763	1.151999	0.362182	0.576858	-0.02566
-0.06731	1.322565	0.580826	1.183004	0.362182	0.570071	0.295505
-0.12888	1.308992	1.400085	1.157071	0.355395	0.563285	0.875965
-0.13402	1.288633	0.778489	1.11356	0.368968	0.542925	0.508445
-0.19929	1.268273	-0.06685	1.183004	0.368968	0.563285	-0.18889
-0.27453	1.23434	1.699962	1.151999	0.362182	0.570071	1.478002
-0.2521	1.193621	-0.87716	1.184394	0.362182	0.563285	-1.05084
-0.22337	1.152901	0.086234	1.150273	0.362182	0.556498	-0.10075
-0.21971	1.118968	-0.08623	1.151999	0.362182	0.549712	-0.25783
-0.19929	1.085036	-0.8228	1.183004	0.362182	0.549712	-0.89287
-0.2251	1.051103	0.909035	1.151999	0.355395	0.556498	0.518311
-0.25749	1.01717	0	1.184394	0.388609	0.556498	-0.31974
-0.20263	0.976451	-0.33192	1.150273	0.381822	0.563285	-0.49603
-0.17074	0.942518	-0.51026	1.150273	0.368249	0.570071	-0.62354
-0.25071	0.915372	0.755953	1.183004	0.368249	0.576858	0.278348
-0.17753	0.874652	-0.75595	1.157071	0.361463	0.583644	-0.86183
0.00539	0.847506	0.755953	0.9269	0.375036	0.563285	0.492812
0.106248	0.847506	-0.8228	0.877467	0.368249	0.563285	-0.82943
-0.01719	0.766882	-0.01395	1.274608	0.570071	0.400407	-0.24005
0.009502	0.780455	0.256473	1.286943	0.576858	0.441127	0.016398
-0.01397	0.807601	-0.24252	1.274608	0.563285	0.461486	-0.34212
0.013162	0.821174	-0.14624	1.287279	0.515779	0.461486	-0.2153
-0.02226	0.84832	0.132289	1.274117	0.522565	0.420767	-0.10573
0.01123	0.861893	-0.41103	1.287279	0.522565	0.461486	-0.35788
0.01663	0.882253	-0.02893	1.274608	0.522565	0.407194	-0.09145
-0.01446	0.902613	0.696433	1.274117	0.529352	0.41398	0.196103
0.002451	0.922973	-0.78652	1.287279	0.529352	0.366474	-0.38283
-0.00373	0.943332	0.922551	1.274608	0.495419	0.400407	0.281691
1.167289	0.914252	-0.92287	0.870282	0.515779	0.427146	-1.14989
0.062364	0.855107	-0.46	1.211754	0.505449	0.751309	-0.50577
0.080247	0.834747	-0.07	1.216129	0.502696	0.745599	-0.09165
0.063186	0.807601	0.052244	1.212863	0.50069	0.726375	0.015836
0.045115	0.787241	0	1.212863	0.496945	0.713999	-0.04819

<b>RS</b>	<b>DS</b>	<b>AL</b>	<b>SF</b>	<b>LLDS</b>	<b>RLDS</b>	<b>AF</b>
0.105576	0.773668	0.477755	1.183004	0.499874	0.701682	0.415106
0.048306	0.746522	-0.8563	1.236523	0.496048	0.696213	-0.94428
0.08986	0.732949	0.76864	1.188483	0.498968	0.690804	0.641262
0.091605	0.705803	0.087664	1.183004	0.495231	0.671882	0.066172
0.076197	0.678656	-0.53	1.216129	0.491285	0.666593	-0.5927
0.075716	0.658297	0.052244	1.212863	0.494123	0.647792	0.012921
0.148409	0.637937	1.001453	1.150273	0.485533	0.64941	0.870778
0.041182	0.604004	-1.0537	1.216129	0.482169	0.644303	-1.18088
0.103843	0.590431	0.442334	1.188483	0.487567	0.625682	0.402301
0.082822	0.563285	0.065419	1.184394	0.49016	0.634268	0.028549
0.089905	0.542925	-0.43776	1.211754	0.489269	0.622555	-0.53065
0.073584	0.522565	0.437756	1.184394	0.499042	0.617688	0.368587
0.072917	0.502206	0	1.184394	0.495575	0.606096	-0.05846
0.052129	0.508992	0	1.184394	0.498515	0.594563	-0.08295
0.05656	0.536138	-0.47309	1.213962	0.502455	0.596664	-0.6043
0.0543	0.542925	0.164547	1.203678	0.50444	0.558106	0.06533
0.053519	0.556498	0.33079	1.183004	0.507234	0.580687	0.230688
0.053519	0.576858	0	1.183004	0.503349	0.576183	-0.07969
0.053001	0.597218	-0.02225	1.184394	0.497406	0.571738	-0.10119
0.005236	0.610791	-0.50775	1.216129	0.497459	0.560568	-0.65753
0.079452	0.637937	0.944926	1.157071	0.497511	0.542671	0.703377
0.029534	0.644723	-0.94493	1.216129	0.497563	0.524834	-1.0507
0.048912	0.665083	0.442334	1.188483	0.497616	0.534204	0.287839
0.149523	0.692229	1.19876	1.11356	0.497668	0.516488	0.886764
0.053138	0.719376	-1.5711	1.211754	0.497721	0.512405	-1.42459
0.107805	0.746522	1.571096	1.11356	0.497773	0.508383	1.056604
0.154825	0.732949	-0.0342	1.115698	0.497825	0.504421	-0.03528
0.280574	0.760095	1.934528	0.93229	0.497878	0.493733	1.636281
0.277774	0.719376	-0.75595	0.979537	0.49793	0.483105	-0.90774
0.285963	0.699016	0.842187	0.9269	0.497983	0.479324	0.664723
0.232217	0.665083	-0.84219	0.979537	0.498035	0.475603	-0.9783
0.305105	0.65151	0.755953	0.93229	0.498087	0.471942	0.611217
-0.2904	1.011198	0	1.274117	0.856815	0.278249	0.113445
-0.25909	1.072277	-0.71164	1.287279	0.860884	0.386834	-0.42248
-0.12384	1.133356	-1.95328	1.274117	0.858077	0.393621	-1.15994
-0.18797	1.167289	0.587396	1.301529	0.861965	0.393621	0.292149
-0.1423	1.228368	-0.5027	1.287279	0.838617	0.41398	-0.3687
-0.08292	1.275874	-1.08615	1.295787	0.835538	0.420767	-0.76472
-0.07442	1.309807	0	1.287279	0.839155	0.427553	-0.03048

RS	DS	AL	SF	LLDS	RLDS	AF
-0.06421	1.364099	-0.37855	1.300729	0.835895	0.312182	-0.29675
-0.1111	1.404818	0.970948	1.286943	0.846118	0.318968	0.610487
-0.07966	1.445538	-0.64464	1.295787	0.842677	0.318968	-0.49019
-0.10808	1.479471	0.442334	1.296564	0.845932	0.427553	0.257874
-0.05858	1.52019	-0.82364	1.29854	0.849097	0.441127	-0.60833
-0.16872	1.547336	1.728476	1.300651	0.838598	0.298609	1.255571
-0.10557	1.608415	-0.90484	1.294052	0.868728	0.298609	-0.66318
-0.03848	1.642348	-1.10125	1.295787	0.871622	0.332542	-0.96414
-0.1227	1.662708	1.303553	1.29854	0.840492	0.332542	0.997447
0	1.710214	-1.78305	1.287279	0.843205	0.305395	-1.4985
-0.12987	1.710214	1.083337	1.286943	0.872973	0.312182	0.83529
-0.08679	1.750933	-0.87493	1.29854	0.868718	0.312182	-0.69881
-0.00851	1.784866	-1.20841	1.295787	0.871159	0.339328	-1.14865
-0.05698	1.791653	1.054629	1.278343	0.846364	0.339328	0.94004
-0.10255	1.812012	0.591536	1.286943	0.848624	0.318968	0.511335
0.014451	1.845945	-1.7823	1.281337	0.871154	0.325755	-1.64357
-0.00485	1.845945	0.141513	1.291796	0.873233	0.346115	0.149401
-0.14161	1.886664	1.963953	1.257311	0.848076	0.318968	1.842664
-0.03759	1.920597	-1.9332	1.274117	0.880647	0.318968	-1.5398
-0.02079	1.93417	0	1.257311	0.882274	0.318968	0.029715
-0.04708	1.940957	0.37855	1.259942	0.883811	0.318968	0.396753
-0.00924	1.95453	-0.72184	1.267216	0.885257	0.332542	-0.73715
-0.05923	1.95453	0.721842	1.272097	0.87304	0.352901	0.706031
-0.04005	1.968103	-0.3925	1.277441	0.847159	0.325755	-0.32851
0.000668	1.97489	-0.32934	1.257311	0.882267	0.359688	-0.30391
-0.04622	1.97489	0.739596	1.257979	0.842632	0.359688	0.739328
0.01663	1.988463	-1.00567	1.257979	0.843626	0.352901	-0.95706
0.009961	1.981676	-0.38291	1.28858	0.851316	0.366474	-0.36021
-0.12717	1.968103	1.372287	1.277441	0.845343	0.318968	1.290809
0.02402	1.995249	-1.41903	1.277441	0.886785	0.359688	-1.43924
-0.04092	1.97489	1.039023	1.277441	0.846698	0.318968	1.105328
0.000491	1.981676	-0.60937	1.274117	0.887959	0.359688	-0.6201
0	1.981676	0.276766	1.257311	0.847691	0.359688	0.334453
0.026042	1.981676	-0.69345	1.274608	0.848052	0.359688	-0.69543
-0.06883	1.961317	1.794691	1.257311	0.848323	0.366474	1.948491
0.015201	1.981676	-1.62122	1.274608	0.841716	0.373261	-1.37607
0.018738	1.97489	0.220164	1.257311	0.841806	0.359688	0.284373
-0.01467	1.968103	0.257711	1.274608	0.848591	0.366474	0.300319
0.01663	1.97489	-0.23466	1.257979	0.8485	0.359688	-0.18914



RS	DS	AL	SF	LLDS	RLDS	AF
-0.01022	1.968103	0	1.284829	0.855105	0.359688	0.038373
0.001932	1.968103	-0.02305	1.274117	0.848046	0.332542	0.014338
0	1.968103	0.030907	1.274117	0.881616	0.332542	0.055704
-0.01418	1.968103	0.226803	1.274117	0.881163	0.332542	0.262756
0.02167	1.97489	-0.57353	1.274117	0.88062	0.332542	-0.64311
-0.01614	1.961317	0.604944	1.274117	0.879986	0.325755	0.657161
-0.04138	1.968103	0.288296	1.281337	0.879261	0.332542	0.313063
0.009918	1.981676	-0.77282	1.278343	0.878446	0.359688	-0.80346
0	1.97489	0.195395	1.276049	0.850395	0.325755	0.241161
-0.00954	1.97489	0.577424	1.249503	0.869759	0.346115	0.630434
0.184444	1.97489	-0.27761	1.072867	0.855099	0.318968	-0.16385
0.214076	1.93417	-0.47412	1.072867	0.881069	0.312182	-0.47797
0.229118	1.879878	0.474118	1.028192	0.879802	0.312182	0.477894
0.201742	1.832372	-0.27677	1.072867	0.878444	0.312182	-0.25259
0.20125	1.784866	0.007858	1.072867	0.876996	0.305395	0.039857
0.222117	1.017984	0.654696	1.014406	0.41398	0.578482	0.464736
0.360469	0.970478	-0.34329	0.89751	0.427553	0.575206	-0.40693
0.124336	0.882253	-0.60937	1.150273	0.441127	0.569034	-0.52751
0.102871	0.84832	0.234661	1.157071	0.441127	0.566134	0.182645
0.02986	0.821174	0.75326	1.183004	0.427553	0.563356	0.473182
0.115217	0.814388	-0.86965	1.151999	0.420767	0.560699	-0.80756
0.107943	0.787241	0.116389	1.151999	0.420767	0.558161	0.092232
0.051477	0.760095	0.319714	1.188483	0.420767	0.555741	0.200347
0.068828	0.746522	-0.27761	1.188483	0.43434	0.553436	-0.29699
0.085395	0.726162	0.318656	1.151999	0.41398	0.551247	0.236238
0.017831	0.705803	0.392499	1.195033	0.420767	0.54917	0.208389
0	0.699016	0.455509	1.184394	0.441127	0.547205	0.232918
0.12405	0.699016	-1.46648	1.151999	0.4547	0.54535	-1.42282
0.076938	0.665083	0.257711	1.183004	0.43434	0.543604	0.182221
0.074307	0.644723	0.042105	1.183004	0.434345	0.541965	-0.01581
0.087961	0.624364	0.277609	1.151999	0.434347	0.540431	0.210705
0.053001	0.604004	0.041047	1.184394	0.427553	0.539002	-0.05013
0.076938	0.590431	-0.36076	1.183004	0.427455	0.537674	-0.44221
0.073584	0.570071	0.031418	1.184394	0.427655	0.536448	-0.03888
0.034122	0.549712	1.177352	1.150273	0.43434	0.535322	0.822574
0.053001	0.542925	-0.84801	1.184394	0.427553	0.534294	-1.01492
-0.07083	0.529352	1.981349	1.184394	0.420767	0.533362	1.44993
0.055566	0.542925	-1.0224	1.184394	0.427553	0.532525	-1.19609
0.108968	0.529352	-0.91985	1.188483	0.427545	0.531781	-1.13036

<b>RS</b>	<b>DS</b>	<b>AL</b>	<b>SF</b>	<b>LLDS</b>	<b>RLDS</b>	<b>AF</b>
0.02736	0.488633	1.371148	1.184394	0.41398	0.53113	1.008144
0.054391	0.481846	-0.41025	1.183004	0.427553	0.530569	-0.5639
-0.03273	0.468273	1.393953	1.183004	0.420767	0.530097	0.922914
0.025998	0.475059	-1.435	1.213962	0.420777	0.529713	-1.66976
0.02986	0.461486	0.433547	1.183004	0.4547	0.529414	0.260103
-0.1515	0.4547	2.901835	1.183004	0.447913	0.5292	2.48719
-0.16257	0.447913	3.933367	1.203678	0.420767	0.52902	3.378037
0.053001	0.475059	-1.14059	1.184394	0.447913	0.52905	-1.37786
-0.01928	0.461486	0.848009	1.203678	0.4547	0.529159	0.495412
0.042373	0.461486	-0.65572	1.183004	0.427553	0.529344	-0.86115
0.0442	0.461486	-0.48925	1.211754	0.427455	0.529606	-0.70385
0.053001	0.4547	0.296955	1.184394	0.427555	0.529941	0.179047
-0.07629	0.441127	1.366324	1.228285	0.427655	0.530348	0.833929
-0.01928	0.4547	-0.51831	1.203678	0.4547	0.530827	-0.77333
-0.00139	0.4547	0.022245	1.184394	0.427553	0.531374	-0.17286
0.030959	0.4547	-0.49534	1.183004	0.423755	0.53199	-0.68515
-0.05718	0.4547	0.407673	1.245663	0.425755	0.532672	0.020278
-0.0398	0.461486	0	1.228285	0.468273	0.533419	-0.29919
0.053001	0.468273	-0.78259	1.184394	0.43434	0.534229	-0.99842
-0.1686	0.4547	1.567814	1.183004	0.441127	0.535101	0.885394
0.03121	0.502206	-1.45409	1.212863	0.447913	0.538074	-1.64928
-0.02327	0.495419	0.889443	1.211754	0.446791	0.539178	0.52932
-0.1078	0.515779	1.795216	1.221365	0.468273	0.542812	1.274144
0.021266	0.536138	-1.98135	1.216129	0.441127	0.544124	-1.91952
-0.1139	0.529352	1.232644	1.211754	0.4547	0.545485	0.659519
0.046225	0.563285	-1.56199	1.211754	0.441127	0.546893	-1.66085
-0.06086	0.549712	1.695667	1.212863	0.447913	0.548346	1.211991
0.028206	0.563285	-1.40737	1.211754	0.45463	0.549843	-1.53179
-0.09606	0.556498	1.988197	1.211754	0.45447	0.551382	1.520357
-0.30138	0.576858	1.020762	1.228285	0.45467	0.552962	0.142915
-0.18341	0.624364	-1.77688	1.221365	0.481846	0.554581	-1.69069
0.222117	1.017984	0.654696	1.014406	0.41398	0.756815	0.473504
0.360469	0.970478	-0.34329	0.89751	0.427553	0.760884	-0.2692
0.124336	0.882253	-0.60937	1.150273	0.441127	0.758077	-0.61577
0.102871	0.84832	0.234661	1.157071	0.441127	0.761965	0.205151
0.02986	0.821174	0.75326	1.183004	0.427553	0.738617	0.578197
0.115217	0.814388	-0.86965	1.151999	0.420767	0.735538	-0.88196
0.107943	0.787241	0.116389	1.151999	0.420767	0.739155	0.105275
0.051477	0.760095	0.319714	1.188483	0.420767	0.735895	0.254474

RS	DS	AL	SF	LLDS	RLDS	AF
0.068828	0.746522	-0.27761	1.188483	0.43434	0.746118	-0.30962
0.085395	0.726162	0.318656	1.151999	0.41398	0.742677	0.27413
0.017831	0.705803	0.392499	1.195033	0.420767	0.745932	0.303343
0	0.699016	0.455509	1.184394	0.441127	0.749097	0.345638
0.12405	0.699016	-1.46648	1.151999	0.4547	0.738598	-1.40067
0.076938	0.665083	0.257711	1.183004	0.43434	0.768728	0.225617
0.074307	0.644723	0.042105	1.183004	0.43434	0.771622	0.019681
0.087961	0.624364	0.277609	1.151999	0.43434	0.740492	0.247372
0.053001	0.604004	0.041047	1.184394	0.427553	0.743205	0.00375
0.076938	0.590431	-0.36076	1.183004	0.427553	0.772973	-0.39467
0.073584	0.570071	0.031418	1.184394	0.427553	0.768718	0.009321
0.034122	0.549712	1.177352	1.150273	0.43434	0.771159	1.014409
0.053001	0.542925	-0.84801	1.184394	0.427553	0.746364	-0.92202
-0.07083	0.529352	1.981349	1.184394	0.420767	0.748624	1.904449
0.055566	0.542925	-1.0224	1.184394	0.427553	0.771154	-1.08516
0.108968	0.529352	-0.91985	1.188483	0.427553	0.773233	-0.98302
0.02736	0.488633	1.371148	1.184394	0.41398	0.748076	1.268147
0.054391	0.481846	-0.41025	1.183004	0.427553	0.780647	-0.45519
-0.03273	0.468273	1.393953	1.183004	0.420767	0.782274	1.281274
0.025998	0.475059	-1.435	1.213962	0.420767	0.783811	-1.50743
0.02986	0.461486	0.433547	1.183004	0.4547	0.785257	0.379431
-0.1515	0.4547	1.901835	1.183004	0.447913	0.77304	1.847331
-0.16257	0.447913	1.933367	1.203678	0.420767	0.747159	1.885219
0.053001	0.475059	-1.14059	1.184394	0.447913	0.782267	-1.18802
-0.01928	0.461486	0.848009	1.203678	0.4547	0.742632	0.725385
0.042373	0.461486	-0.65572	1.183004	0.427553	0.743626	-0.72853
0.0442	0.461486	-0.48925	1.211754	0.427553	0.751316	-0.58386

**Validation Data**

RS	DS	AL	SF	LLDS	RLDS	AF
0.053001	0.4547	0.296955	1.184394	0.427553	0.745343	0.256984
-0.07629	0.441127	1.366324	1.228285	0.427553	0.786785	1.285188
-0.01928	0.4547	-0.51831	1.203678	0.4547	0.746698	-0.63634
-0.00139	0.4547	0.022245	1.184394	0.427553	0.787959	-0.032
0.030959	0.4547	-0.49534	1.183004	0.427553	0.747691	-0.56191
-0.05718	0.4547	0.407673	1.245663	0.427553	0.748052	0.291139
-0.0398	0.461486	0	1.228285	0.468273	0.748323	-0.11037
0.053001	0.468273	-0.78259	1.184394	0.43434	0.741716	-0.86069
-0.1686	0.4547	1.567814	1.183004	0.441127	0.741806	1.353574
0.03121	0.502206	-1.45409	1.212863	0.447913	0.748591	-1.52673

<b>RS</b>	<b>DS</b>	<b>AL</b>	<b>SF</b>	<b>LLDS</b>	<b>RLDS</b>	<b>AF</b>
-0.02327	0.495419	0.889443	1.211754	0.447913	0.7485	0.762561
-0.1078	0.515779	1.795216	1.221365	0.468273	0.755105	1.712326
0.021266	0.536138	-1.98135	1.216129	0.441127	0.748046	-1.84969
-0.1139	0.529352	1.232644	1.211754	0.4547	0.781616	1.054114
0.046225	0.563285	-1.56199	1.211754	0.441127	0.781163	-1.58373
-0.06086	0.549712	1.695667	1.212863	0.447913	0.78062	1.603307
0.028206	0.563285	-1.40737	1.211754	0.4547	0.779986	-1.47704
-0.09606	0.556498	1.988197	1.211754	0.4547	0.779261	1.950003
-0.30138	0.576858	1.020762	1.228285	0.4547	0.778446	0.705566
-0.18341	0.624364	-1.77688	1.221365	0.481846	0.750395	-1.83577
-0.16836	1.06549	0	1.041108	0.612505	0.510706	-0.02594
-0.15876	1.08585	-1.32871	1.031499	0.624042	0.522244	-1.07418
-0.00472	1.10621	1.358913	0.877467	0.61522	0.513422	1.07713
-0.15876	1.10621	-1.02	1.031499	0.619971	0.518173	-0.85879
-0.14415	1.126569	-0.89559	1.072867	0.624722	0.522924	-0.78303
-0.20277	1.146929	0.89559	1.075517	0.622687	0.520888	0.669715
-0.10679	1.174075	-1.01981	0.979537	0.613865	0.512066	-0.89647
-0.10679	1.187648	-1.32871	0.979537	0.611829	0.510031	-1.09989
-0.20277	1.201222	1.358913	1.075517	0.609794	0.507995	1.18035
-0.24922	1.228368	-1.01981	1.121961	0.614545	0.512746	-0.96823
-0.16836	1.248728	1.019807	1.041108	0.653229	0.55143	0.775336
-0.11097	1.269087	-1.35891	0.983715	0.63762	0.535821	-1.28586
-0.05416	1.28266	1.328715	0.9269	0.642371	0.540573	1.045572
-0.16836	1.28266	1.019807	1.041108	0.660695	0.558897	0.77639
-0.22275	1.30302	-1.35891	1.095498	0.672233	0.570434	-1.46117
-0.07091	1.330166	-0.89559	0.99963	0.649838	0.548039	-0.97612
-0.15545	1.330166	0.89559	1.028192	0.620656	0.518857	0.73421
-0.18343	1.350526	-1.35891	1.056173	0.61862	0.516822	-1.16006
-0.24297	1.370886	0.952635	1.056173	0.636944	0.535146	0.867627
-0.20277	1.398032	-0.95264	1.075517	0.614549	0.512751	-0.85802
-0.28762	1.418392	-1.35891	1.160363	0.63966	0.537861	-1.51967
-0.05416	1.452324	1.328715	0.9269	0.603692	0.501893	1.352464
-0.10679	1.459111	1.019807	0.979537	0.601656	0.499858	0.934997
0.052955	1.472684	-1.01981	0.819788	0.599621	0.497822	-1.06271
-0.00472	1.465898	-1.32871	0.877467	0.604372	0.502573	-1.23651
-0.08431	1.465898	0.952635	0.89751	0.609123	0.507324	0.832529
-0.00357	1.459111	-1.84822	0.93229	0.634233	0.532435	-1.67477
-0.05708	1.459111	1.848225	0.870282	0.638984	0.537186	1.822943
0.12764	1.465898	-0.95264	0.745103	0.636949	0.53515	-1.10665

RS	DS	AL	SF	LLDS	RLDS	AF
0.025889	1.452324	-1.01981	0.846855	0.634913	0.533115	-1.15867
0.235819	1.445538	-1.73087	0.745103	0.619305	0.517506	-1.6042
-0.17269	1.425178	0.650424	0.819788	0.64238	0.540581	0.539209
-0.25041	1.445538	-1.12923	0.89751	0.633558	0.531759	-1.26533
0.0681	1.472684	-1.65763	0.745103	0.645095	0.543297	-1.65331
0.002461	1.465898	-0.95264	0.870282	0.64306	0.541261	-1.13058
0.102368	1.398032	1.319863	0.9269	0.638989	0.53719	1.059394
0.264028	1.370886	-0.71108	0.809683	0.609807	0.508009	-0.90061
0.323844	1.336953	0.711081	0.705424	0.607772	0.505973	0.563809
0.313003	1.296233	-0.71108	0.760708	0.612523	0.510724	-0.90101
0.166356	1.133356	-0.83528	0.870282	0.940626	0.601297	-0.57274
0.336951	1.119783	-0.70939	0.819788	0.937459	0.598131	-0.51234
0.166356	1.099423	0.887024	0.870282	0.927506	0.588178	0.900949
0.284747	1.08585	0.835277	0.819788	0.92434	0.585012	0.76853
0.246683	1.072277	-0.83528	0.846855	0.914387	0.575059	-0.66449
0.141287	1.051917	0.887024	0.877467	0.897647	0.558319	0.869585
0.116945	1.038344	-1.77636	0.884386	0.887695	0.548366	-1.47622
-0.66089	1.017984	1.776357	1.075517	0.904888	0.56556	1.675247
0.141287	1.004411	-0.88702	0.877467	0.854216	0.514887	-0.88169
0.166356	0.963692	-0.88702	0.870282	0.905796	0.566467	-0.77064
-0.03729	0.950119	0.709389	0.9269	0.902629	0.563301	0.821365
0.385298	0.936546	-0.77354	0.819788	0.919822	0.580494	-0.63598
-0.30999	0.909399	1.608815	0.983715	0.90987	0.570541	1.72711
0.141287	0.916186	-0.83528	0.877467	0.899917	0.560588	-0.75133
0.166356	0.902613	1.776357	0.870282	0.903537	0.564209	1.607878
-0.2065	0.88904	-0.06466	0.970918	0.90037	0.561042	0.149742
-0.81769	0.882253	-1.13284	1.13193	0.876844	0.537516	-1.20862
-0.52162	0.882253	1.359302	1.056173	0.894038	0.55471	1.386757
-0.25778	0.882253	0.773538	0.983715	0.870512	0.531184	0.841406
-0.89656	0.882253	-0.70939	1.121961	0.860559	0.521231	-0.68272
-0.20943	0.882253	-0.77354	0.983715	0.904898	0.56557	-0.62064
-0.32574	0.875467	-1.3593	1.031499	0.894945	0.555617	-1.21571
-0.67814	0.861893	1.863707	1.056173	0.898566	0.559237	2.100809
-0.47474	0.882253	0.952635	0.995765	0.875039	0.535711	1.081376
-0.49677	0.902613	-1.84822	1.028192	0.8583	0.518972	-1.67977
-0.03729	0.916186	-0.83528	0.9269	0.855134	0.515805	-0.87635
-0.52162	0.909399	-0.77354	1.056173	0.851967	0.512639	-0.74078
-0.56996	0.909399	0.773538	1.056173	0.828441	0.489113	0.781794
-0.50161	0.916186	-1.70939	1.041108	0.845635	0.506306	-1.63868

<b>RS</b>	<b>DS</b>	<b>AL</b>	<b>SF</b>	<b>LLDS</b>	<b>RLDS</b>	<b>AF</b>
-0.24093	0.922973	1.887024	0.979537	0.856041	0.516713	1.592605
-0.52162	0.922973	-0.77354	1.056173	0.852875	0.513547	-0.73159
-0.20943	0.922973	1.421085	0.983715	0.870068	0.53074	1.301378
-0.32302	0.916186	0.773538	0.99963	0.860115	0.520787	0.822654
-0.41069	0.909399	-0.77354	1.031499	0.829803	0.490475	-0.8209
-0.39622	0.909399	1.421085	1.028192	0.833423	0.494095	1.149924
-0.20943	0.909399	-0.21725	0.983715	0.830256	0.490928	-0.21194
-0.0091	0.902613	-1.07514	0.93229	0.833877	0.494549	-1.19976
-0.59986	0.88904	1.421085	1.072867	0.837497	0.498169	1.184931
-0.5681	0.895826	-0.71108	1.075517	0.834331	0.495002	-0.71144
-0.69869	0.88904	0.711081	1.093112	0.85831	0.518982	0.74568
-0.65424	0.882253	-0.71108	1.093112	0.814425	0.475097	-0.72265
-0.3368	0.86868	0.711081	1.014406	0.851978	0.51265	0.744917
0.035339	0.861893	-0.71108	0.93229	0.869171	0.529843	-0.7202
-0.8582	0.841534	0	1.13193	0.872791	0.533463	0.061643
-0.21423	0.84832	-1.11591	0.995765	0.849265	0.509937	-1.22251
-0.02293	0.834747	1.115908	0.947645	0.832526	0.493198	0.932494
-0.35178	0.814388	0	1.028192	0.815786	0.476458	-0.00014
-0.12448	0.807601	-0.64822	0.983715	0.81262	0.473292	-0.8368
-0.40882	0.787241	0.648221	1.041108	0.81624	0.476912	0.615976
-0.10762	0.780455	-0.64822	0.979537	0.7995	0.460172	-0.88035
-0.17372	0.760095	-1.06581	0.995765	0.796334	0.457006	-1.39011
-0.72983	0.739735	-1.94351	1.115698	0.806741	0.467413	-1.92745
-0.28917	0.739735	-0.58512	1.031499	0.810361	0.471033	-0.77541
-0.32574	0.719376	0.58512	1.031499	0.813981	0.474653	0.523431
-0.49102	0.705803	-0.58512	1.075517	0.817601	0.478273	-0.71739
-0.52848	0.692229	1.268497	1.083255	0.821222	0.481894	1.070816
-0.44572	0.678656	-0.52186	1.072867	0.804482	0.465154	-0.6886
-0.66064	0.678656	-1.55795	1.115698	0.801316	0.461988	-1.73734
-0.86316	0.712589	1.629008	1.151999	0.804936	0.465608	1.529716
-0.44572	0.705803	-1.10543	1.072867	0.794983	0.455655	-1.40676
-0.44572	0.685443	1.105427	1.072867	0.791817	0.452489	0.850326
-1.84255	0.665083	-0.4585	1.274117	0.78865	0.449322	-0.596
-0.44572	0.678656	0.458504	1.072867	0.866923	0.527595	0.533654
-0.62071	0.705803	-0.4585	1.11356	0.863756	0.524428	-0.44717
-0.62071	0.685443	-1.13163	1.11356	0.86059	0.521262	-1.22122
-0.8345	0.712589	1.131628	1.151999	0.857424	0.518095	1.086347
-0.66064	0.699016	0.458504	1.115698	0.847471	0.508143	0.523283
-0.66541	0.685443	-0.4585	1.121961	0.837518	0.49819	-0.47708

RS	DS	AL	SF	LLDS	RLDS	AF
-0.63198	0.665083	1.131628	1.115698	0.820778	0.48145	0.999307
-0.62071	0.644723	0	1.11356	0.824398	0.48507	0.027108
-0.63198	0.624364	-1.13163	1.115698	0.821232	0.481904	-1.33821
-0.82435	0.604004	1.131628	1.150273	0.811279	0.471951	1.115662
-0.63198	0.590431	0	1.115698	0.808113	0.468785	-0.00653
-1.06471	0.570071	-1.10543	1.188483	0.811733	0.472405	-1.03635
-0.91401	0.563285	-1.02505	1.165175	0.82214	0.482812	-1.06222
-0.79357	0.549712	1.096101	1.144979	0.798614	0.459286	1.039579
-0.91401	0.563285	-1.10543	1.165175	0.822594	0.483265	-1.14581
-1.06471	0.570071	-1.02505	1.188483	0.799067	0.459739	-1.00172
-1.02799	0.590431	1.096101	1.183004	0.829834	0.490506	1.297761
-0.79357	0.604004	-1.10543	1.144979	0.826668	0.487339	-1.21408
-1.23163	0.624364	-1.02505	1.211754	0.796355	0.457027	-0.93032
-0.91401	0.644723	1.096101	1.165175	0.793189	0.45386	1.040657
-1.01253	0.665083	-0.39507	1.184394	0.810382	0.471054	-0.35115
-0.86468	0.665083	0.395074	1.157071	0.814002	0.474674	0.451876
-1.2401	0.678656	1.096101	1.212863	0.797263	0.457934	1.220663
-1.02799	0.678656	-1.10543	1.183004	0.800883	0.461555	-1.07767
-1.03723	0.678656	1.105427	1.184394	0.797716	0.458388	1.12146
-1.20694	0.685443	-0.39507	1.211754	0.801337	0.462008	-0.37163
-1.06471	0.685443	0.395074	1.188483	0.79817	0.458842	0.375712
-0.91401	0.685443	1.105427	1.165175	0.808577	0.469249	1.083004
-1.55776	0.692229	-1.13163	1.249503	0.785051	0.445723	-1.1084
-1.02799	0.699016	1.131628	1.183004	0.809031	0.469703	1.167025
-1.2401	0.705803	0	1.212863	0.805864	0.466536	0.041046
-1.26876	0.705803	0.458504	1.212863	0.809485	0.470157	0.533597
-1.01253	0.705803	-0.85358	1.184394	0.799532	0.460204	-0.85881
-1.26876	0.712589	0.853578	1.212863	0.803152	0.463824	0.920752
-1.2401	0.712589	-0.4585	1.212863	0.806772	0.467444	-0.39246
-1.36382	0.719376	1.131628	1.228285	0.810392	0.471064	1.354868
-1.26876	0.739735	0.458504	1.212863	0.834372	0.495044	0.596888
-0.86316	0.739735	-1.10543	1.151999	0.837992	0.498664	-1.15061
-1.23163	0.746522	-0.4585	1.211754	0.841613	0.502284	-0.37574
-1.26876	0.746522	0.458504	1.212863	0.838446	0.499118	0.593227
-1.03723	0.753308	-0.4585	1.184394	0.842066	0.502738	-0.42176
-1.1098	0.753308	-1.02505	1.195033	0.845687	0.506358	-1.03717
-1.26029	0.753308	0.458504	1.211754	0.86288	0.523552	0.589124
-1.05664	0.760095	-1.10543	1.183004	0.859714	0.520385	-1.20059
-1.29399	0.760095	-1.55795	1.216129	0.856547	0.517219	-1.56784

<b>RS</b>	<b>DS</b>	<b>AL</b>	<b>SF</b>	<b>LLDS</b>	<b>RLDS</b>	<b>AF</b>
-0.86468	0.780455	-0.4585	1.157071	0.866954	0.527626	-0.4884
-1.06588	0.787241	0.458504	1.184394	0.877361	0.538033	0.476846
-1.08926	0.807601	0.521863	1.183004	0.867408	0.52808	0.527501
-1.03723	0.814388	-0.98037	1.184394	0.864241	0.524913	-1.13069
-0.853	0.834747	0.458504	1.150273	0.867862	0.528533	0.440084
-0.86316	0.841534	1.105427	1.151999	0.864695	0.525367	0.932383
-0.89333	0.84832	-1.06	1.157071	0.854742	0.515414	-1.1372
-0.66064	0.861893	0.700062	1.115698	0.865149	0.525821	0.661845
-1.02799	0.861893	-0.4585	1.183004	0.868769	0.529441	-0.59029
-0.64936	0.86868	0.458504	1.11356	0.865603	0.526275	0.472231
-0.88491	0.875467	0.510517	1.150273	0.862437	0.523108	0.460033
-0.61998	0.882253	-0.98063	1.11356	0.85927	0.519942	-1.0156
-2.68066	0.882253	-0.37863	1.301162	0.856104	0.516776	-0.40612
-0.84029	0.882253	-0.8342	1.157071	0.860178	0.52085	-0.93798
-0.65973	0.882253	0.834204	1.115698	0.870585	0.531256	0.751015
-0.42051	0.882253	-0.59162	1.075517	0.874205	0.534877	-0.53818
-0.61998	0.88904	0.136052	1.11356	0.877825	0.538497	0.173611
-0.63237	0.88904	0.017757	1.115698	0.874659	0.53533	0.054873
-0.91329	0.88904	-0.01776	1.165175	0.864706	0.525378	-0.0826
-0.66469	0.895826	0	1.121961	0.881899	0.542571	0.016958
-0.64984	0.902613	0.477818	1.11356	0.892306	0.552978	0.476229
-0.40191	0.902613	-0.91146	1.075517	0.889139	0.549811	-0.84577
-0.82362	0.902613	0.433647	1.150273	0.879186	0.539858	0.333516
-0.71912	0.909399	0	1.13193	0.87602	0.536692	-0.01277
-0.44281	0.909399	-0.37863	1.083255	0.852494	0.513166	-0.33657
-0.61671	0.909399	0.326381	1.11356	0.862901	0.523573	0.336596
-0.64102	0.909399	-0.32638	1.121961	0.859734	0.520406	-0.31859
-0.63126	0.909399	0.378634	1.115698	0.870141	0.530813	0.377314
-0.59544	0.909399	-0.39259	1.11356	0.873761	0.534433	-0.3775
-0.62109	0.916186	0.410345	1.11356	0.870595	0.531267	0.405997
-0.64102	0.916186	-0.39639	1.121961	0.867429	0.5281	-0.39954
-0.46648	0.916186	0.378634	1.083255	0.877835	0.538507	0.444836
-0.9144	0.916186	0.017757	1.165175	0.888242	0.548914	-0.13336
-0.59632	0.922973	-0.39639	1.11356	0.864716	0.525388	-0.3806
-0.6176	0.922973	-0.37484	1.121961	0.86155	0.522222	-0.37571
-0.60672	0.916186	0.360891	1.115698	0.871956	0.532628	0.354405
-0.59632	0.916186	0.013954	1.11356	0.875577	0.536249	0.045044
-0.83941	0.916186	-0.01395	1.157071	0.87241	0.533082	-0.10271
-0.59632	0.909399	0.013954	1.11356	0.882817	0.543489	0.046131



<b>RS</b>	<b>DS</b>	<b>AL</b>	<b>SF</b>	<b>LLDS</b>	<b>RLDS</b>	<b>AF</b>
-0.88875	0.909399	-0.01395	1.165175	0.879651	0.540323	-0.13853
-0.44281	0.902613	0.013954	1.083255	0.856125	0.516796	0.081156
-1.20724	0.902613	0	1.211754	0.866531	0.527203	-0.17688
-0.5868	0.902613	-0.33272	1.115698	0.863365	0.524037	-0.31798
-1.2309	0.895826	0.711355	1.211754	0.866985	0.527657	0.48645
-0.41939	0.895826	-0.75348	1.083255	0.863819	0.524491	-0.74275
-0.81011	0.895826	0.374845	1.151999	0.874226	0.534897	0.286556
-1.01886	0.88904	-0.34341	1.188483	0.864273	0.524945	-0.48416
-0.80924	0.88904	0.329459	1.151999	0.874679	0.535351	0.252695
-0.78932	0.882253	-0.31877	1.151999	0.8783	0.538971	-0.40426
-0.78865	0.882253	-0.01069	1.151999	0.88192	0.542592	-0.07957
-0.78932	0.882253	0.010692	1.151999	0.88554	0.546212	-0.06357
-2.04716	0.875467	0.332721	1.288261	0.875587	0.536259	0.244368
-0.99204	0.86868	0.042124	1.184394	0.876041	0.536713	-0.07424
-0.57552	0.86868	0	1.11356	0.879661	0.540333	0.038567
-1.20269	0.861893	-0.01069	1.213962	0.876495	0.537167	-0.15881
-0.99204	0.861893	0.010692	1.184394	0.839396	0.500067	-0.0179
-1.20269	0.861893	-0.01069	1.213962	0.829443	0.490114	-0.03943
-0.99137	0.861893	0	1.184394	0.860209	0.520881	-0.06899
-0.81011	0.855107	0.343413	1.151999	0.850256	0.510928	0.315642
-1.22015	0.84832	-0.33272	1.216129	0.853876	0.514548	-0.40446
-0.9828	0.84832	0	1.183004	0.864283	0.524955	-0.06207
-0.99137	0.841534	-0.01069	1.184394	0.861117	0.521789	-0.06003
-0.81882	0.841534	0	1.157071	0.851164	0.511836	-0.00744
-1.19492	0.841534	0.010692	1.212863	0.861571	0.522242	-0.0502
-0.99137	0.834747	-0.01069	1.184394	0.851618	0.512289	-0.03394
-1.22015	0.834747	0.010692	1.216129	0.855238	0.51591	-0.02206
-0.78932	0.834747	0	1.151999	0.865645	0.526316	-0.01371
-1.12564	0.827961	-0.01069	1.203678	0.869265	0.529937	-0.07327
-1.47488	0.827961	0.010692	1.245663	0.838952	0.499624	0.042846
-1.19492	0.827961	0	1.212863	0.876505	0.537177	-0.08374
-1.0169	0.827961	-0.04212	1.188483	0.880125	0.540797	-0.12992
-1.19492	0.827961	0.042124	1.212863	0.890532	0.551204	-0.07996
-1.1872	0.821174	-0.12342	1.212863	0.880579	0.541251	-0.23407
-1.27257	0.814388	0.185316	1.221365	0.877867	0.538538	0.153794
-1.23201	0.807601	0.554488	1.211754	0.89506	0.555732	0.511546
-1.4727	0.807601	1.715411	1.212863	0.891894	0.552565	1.554014
-1.23823	0.800814	0.525993	1.216129	0.865655	0.526327	0.534788
-1.26572	0.787241	0.547481	1.216129	0.904116	0.564787	0.549339

<b>RS</b>	<b>DS</b>	<b>AL</b>	<b>SF</b>	<b>LLDS</b>	<b>RLDS</b>	<b>AF</b>
-1.20637	0.780455	0.318767	1.211754	0.901403	0.562075	0.317694
-1.22328	0.773668	0	1.213962	0.898237	0.558909	-0.03713
-0.21056	0.766882	-0.32946	1.037955	0.861138	0.521809	-0.31666
-0.40448	0.766882	0.329459	1.075517	0.885117	0.545789	0.424374
-0.37188	0.760095	-0.31877	1.072867	0.854805	0.515477	-0.31046
-0.20007	0.760095	0.277708	1.031499	0.851638	0.51231	0.297959
0.019218	0.753308	-0.2884	0.983715	0.855259	0.51593	-0.3115
-0.31356	0.753308	0.329459	1.056173	0.845306	0.505978	0.344132
-0.21381	0.753308	0.410345	1.028192	0.862499	0.523171	0.453845
-0.38226	0.746522	-0.76587	1.075517	0.859333	0.520005	-0.82274
-0.18875	0.746522	0.364923	1.028192	0.862953	0.523625	0.40519
-0.31414	0.739735	0	1.056173	0.859786	0.520458	0.049787
-0.44253	0.739735	1.563194	1.083255	0.87698	0.537652	1.400556
-0.39239	0.732949	-1.56319	1.072867	0.86024	0.520912	-1.60487
-0.58468	0.732949	0	1.111397	0.857074	0.517746	0.053642