ON INTERACTIVE CONTROL FOR INTELLIGENT COLLISION EVASIVE EMERGENCY INTERVENTION IN SMART VEHICLES

A thesis submitted to the
Department of Electrical Engineering, University of Moratuwa
in partial fulfillment of the requirements for the Degree of
Master of Science

by

SAMARANATH RAVIPRIYA RANATUNGA

Supervised by: Dr Sisil Kumarawadu

Department of Electrical Engineering
University of Moratuwa
Sri Lanka

January ~~2006~~ 2007

Contents

Declaration

The work submitted in this thesis is the result of my own investigation, except where otherwise stated.

It has not already been accepted for any other degree, and is also not being concurrently submitted for any other degree.

S R Ranatunga
(Candidate)

Date: 29/01/2007

I endorse the declaration by the candidate.

UOM Verified Signature

Dr. Sisil Kumarawadu
(Supervisor)

Abstract

This research study focuses on finding a solution for collision avoidance of smart vehicular systems. The main paradigm that is used to establish the solution is the interactive control of vehicular systems for negotiating a collision scenario for taking evasive actions. In this study, an interactive controller proposed negotiates collision scenarios between two vehicular systems leading to cooperative maneuvers. Thus, the interactive control actions lead to some maneuvers mutually beneficial to both the vehicles. The objective of this study is to develop a fully operational intelligent interactive controller for the smart vehicles. An Inter-Vehicle Communication (IVC) system plays a pivotal role in exchanging the necessary information in between the vehicles. The IVC system is assumed to be with enough versatility for dealing with multiple collisions in the channel transmitting information.

This study is focusing on the vehicles outside the usually considered platoon environment. It is considering for emergency intervention maneuvers for collision evasive solutions.

The hierarchical differentiation in control of the participatory vehicles is done by using the Master-Slave concept. The master is given more power in comparison to the slave. But these states are moment-bound and are to change fast.

There are two main controllers which have been developed for braking and steering. The two controllers are based on Adaptive Neuro-Fuzzy Inference System (ANFIS). The top tier of this controller includes all important auxiliary functional components for processing the primary sensory variables. The ANFIS controller has been offline trained in the Matlab-7 environment.

A simulation study has been done for the controllers in the Matlab/Simulink environment for various categories of collisions between the two vehicles. Even though the above paradigm is discussed for two participatory vehicular sub-systems, it is emphasized, that the same approach can effectively be extended without any major conceptual breakthrough to any number of vehicles for reliable evasion of collisions. In similar way, multiple vehicles can be considered as a multiplication of the number of pairs of vehicles for applying the results of the above study.

Two fully autonomous prototypes were realized with full capability for testing intelligent interactive collision avoidance trials. Here, all sensor types and equipment were tested for expected functionality to be used in the integrated environment. To this end, software were developed for testing each component in the provided platform.

Acknowledgement

I specially thank my supervisor, Dr. Sisil Kumarawadu for his unwavering guidance, support and advice for carrying out this research work successfully. I am also very appreciative for his extensive help in fulfillment of some publications related to this research work, in some prestigious international forums.

I am indebted to my parents and my wife for constant support and encouragement for successfully carrying out this work.

My gratitude is also due to Prof. H.Y.R. Perera, Head/Electrical Engineering, for offering me a Research Assistantship in the Department of Electrical Engineering, in support of my studies.

My sincere thanks are also due to the chairman and the committee members of the SRC grant committee, University of Moratuwa, for the grant they extended, which existed as an extensive support in my research studies.

I would like to take this opportunity to extend my thanks to Dr. Amith Munindradasa, Dr. Rohan Munasingha, Dr. Thrishantha Nanayakkara and Dr. Lanka Udawatta for being the members of the review committee for my research. If not for their guidance and advice this work wouldn't have been a success at the end.

Dr. N Munasinghe and his staff, at the Engineering Post Graduate Unit, are also thanked for all assistance extended.

I have been assisted extensively by Mr. Geeth Jayendra, in realizing the prototypes for the research work. My appreciations are also due to him.

I would also like to thank Mr. Buddhika Jayasekera and Mr. Dharshana Prasad, who have been my colleagues at the Departmental Research Lab, for helping me in various ways for successfully carrying out this work.

Finally, my thanks go to various other personnel without whose help this work wouldn't have been a success. Understandably, their individual names cannot be mentioned here due to being large in number.

List of Figures

Figure **Page**

List of Tables

CHAPTER 1

Introduction

1.1 Collision Avoidance of Vehicles

Collision avoidance studies for vehicles are carried out for preventing any life threat or material loss from vehicular collisions. Traffic accidents have been taking thousands of lives each year around the globe, outnumbering any deadly disease or natural disaster. This shows the real picture of vehicle accidents. It is expected, the results would be worsened in the future if no effective counter action was taken.

Studies show that about 60% roadway collisions could be avoided if the operator of the vehicle was provided warning at least one-half second prior to a collision [27]. This is due to the fact that human drivers suffer from perception limitations on roadway emergency events, resulting in large delay in propagating emergency warnings. The intelligent control methods can do an effective work, where the human perception limitations hinder emergency reactions. This study takes on the advantage of some computational intelligence methods giving in a promising solution for the discussed problem.

1.2 Interactive Control of Vehicular Systems

An interactive controller plays the prime role in collision avoidance of smart ground vehicles, in this study. The main concern is for proving the paradigm for two vehicles. Further, it can be extended for more vehicles without a major conceptual breakthrough.

The importance of the interactive control can be elaborated. For instance, when a vehicle tries to enter the main road from a side road, an on-coming vehicle on the main road can make the former wait by means of intelligent maneuvering, until the passing-by is over. Similar situations can arise in typical lane changing event too. In order to effectively control the flow around a roundabout, the interactive control will prove to be immensely useful.

The so called controller creates some cooperative maneuvers within each

vehicle intended to create a setup for optimally avoiding a probable collision situation between two vehicles. It can be heuristically verified that more effective collision avoidance is achievable if both the vehicular systems in the verge of a collision take evasive measures after negotiating for a common solution with the other participating vehicles, rather than acting alone by each vehicle. This has led to explore different collision avoidance criteria that will ensure collaboration of all the vehicles on the verge of an accident. The resulting interactive nature of control enhances the perspectives of collision avoidance yielding better results.

In the past, there have been a number of occasions where the cooperative control of vehicles with regard to collision avoidance has been studied. Considering world's number one program for developing Automated Highway Systems (AHS) i.e., the California Partners for Advanced Transit and Highways (PATH) program in cooperation with the State of California Department of Transportation (Caltrans) and the United States Federal Highway Administration (FHWA), a multilayer AHS architecture has been discussed in [7], [19] and [18]. But this control architecture has been discussed with relation to *platoon* environment, i.e., organization of traffic in groups of up to 20 tightly spaced cars. In a platoon environment, to maintain close proximity while traveling at relatively high speeds (90 km/h), the vehicles must be fully automated, since people cannot react quickly enough to drive safely with such small headways as 1-2 [m]. Apart from that, these works have been mainly based on elaborating the architecture of the control design rather than the full design of the controller with development details. In [30], the details were given on three different categories of vehicle controllers for collision avoidance. Here also, the main concern was to elaborate on the control system architectures giving some comparative analysis on the different categories.

To the best of my knowledge, this research is the first attempts to realize interactive control of vehicular systems giving full details of the controller performance with resulting maneuvers of cooperative nature suiting for emergency intervention in collision avoidance, outside a platoon environment. Moreover, the methodology suggested here is very well adaptable not only to the fully-automated vehicles but also for the semi-automated vehicles. More importantly, the controller that has been realized in this study is working well for avoiding all-directional collisions without being amenable to a specific direction [20].

1.3 Intelligent Applications in Vehicles

Generally speaking, as far as collision avoidance is concerned, there are three main strata: collision warning, where it advises or warns the driver by means of audio-visual outputs, partially controlling the vehicle (driver assistance in the steady-state conditions or as an *emergency intervention* to avoid a collision), and fully controlling the vehicle (vehicle automation).

Collision warning systems include functions such as forward-collision warning, blind-spot warning, lane-departure warning, lane-change or lane-merge warning, rear-impact warning and rollover warning for heavy vehicles. A special category of collision warning is driver monitoring to detect and warn of drowsiness or other impairments that prevent the driver from safely operating the vehicle. If the driver does not adequately respond to warnings, collision-avoidance systems might take control of steering, braking or throttle control to maneuver the vehicle back to a safe state. Driver-assistance systems include functions such as adaptive cruise control (ACC), i.e., one of the recent attractions available in the prestigious commercial vehicles where, ACC senses slower vehicles ahead and adjusts speed to establish a safe following distance [11], [21], lane keeping, precision docking, etc.

Vehicle automation systems include low-speed automation, autonomous driving, and close-headway platooning etc. A fully autonomous vehicle will control its own steering and speed all the time [23].

1.4 Hierarchy of Control among Vehicles

In Master-Slave systems, one vehicle or node gives the commands, and another node or collection of nodes executes them. A node can be a master one moment and then be reconfigured at another time. The master has more control on the other vehicle systems during the period of action. On the other hand, slave obeys the latter on selected maneuvers during the time of that stay. This Master-Slave concept which gives the benefit of the centralized control approach, is common in Robotics whereas it is quite novel to the field of smart vehicular systems [24].

Comparing the Master-Slave model with the other hierarchical systems, there is the Peer-to-Peer model, in which there is no designated master; all vehicle systems are equal in rank as far as hierarchy is concerned. Each node can both transmit (if the communication bus is active) and receive messages. Similar to the multi-master

principle, a node can both transmit messages to several other nodes and vice versa with the multicast (one to many, many to one). In peer-to-peer systems, participants rely on one another for service, rather than solely relying on a dedicated and centralized infrastructure. Further, since the communication and control are decentralized, it offers scalability, robustness, and limits requirements for central administration. Even though, peer-to-peer configurations offer greatest flexibility, they are the most difficult to control. Therefore, in this study, the Master-Slave concept has been used in order to differentiate the control hierarchy among the vehicles. [31]

The Inter-Vehicle Communication (IVC) [14] facilitates receipt/transmission of data by/from each vehicular system. The internal auxiliary processing units, associated with the controller, process data relevant to each of the vehicle. The exchange of information is only relevant within a pre-specified perimeter as agreed by both the vehicular systems. Beyond this perimeter, the information received is merely neglected without any acknowledgment.

1.5 Inter-Vehicle Communication (IVC)

The integration of communication technology in state-of-the-art vehicles has already begun and in existence for years. The areas of usage include: car phones, internet access from cellular technologies and Bluetooth adapters for integration of mobile devices. But, the reality of direct communication between vehicles using a wireless ad hoc network generally known as inter-vehicle communication (IVC) or car-to-car communication (C2CC) is relatively new [14]. One advantage of this over the cellular technologies is the low communication delay. This is very important in distribution of time-critical data. The advantage is that a communication network is established by the vehicles themselves within a given perimeter between two or many vehicles. The systems become alerted and begin to establish a communication network between them. This network is based on single-hop communication. The vehicle communication is decentralized and is a self-organizing information system. Inter-vehicle ad hoc networks are highly volatile due to high mobility rates of vehicles. These wireless ad hoc networks need to be spontaneously created and re-configured in a variety of traffic situations [2]. When the relative distance is less than a critical

distance within the above limits, the vehicles start exchanging the relevant sensor and other secondary level information with due 'acknowledgement' from each other.

The approach for disseminating information using the IVC is by flooding the local area of the vehicle (as against the method of traditional routing-based). This is quite feasible as the concerned area is within a perimeter of 40[m] radius in the study of this research.

Under the general study of IVC systems, the five basic types of messages that are exchanged between vehicles are [6]:

1) Basic Safety Messages containing data describing the sender characteristics and driving conditions.
2) Warning Messages containing critical warning on emergency situations that has occurred or could possibly occur within the traffic.
3) Info-Entertainment Messages containing data about services and resources available and offered by other vehicles and information of general interest (e.g. traffic conditions and meteorological data).
4) Routing Messages containing data used by routing protocols.
5) Inter-Personal Messages containing different profiles of the drivers and the passengers in the vehicles.

But in this specific study, the expected main features relevant to the discussion include mainly the numbers 1) and 4).

The exchange of information is by means of 'packets' of data, which has a generic format. This format for such communication might include: vehicle identity information, current latitude/longitude, lane-code, current hierarchy of the vehicle, i.e., master/slave statuses or empowerment details, current velocity, acknowledgement information etc.

The segment of information in a standard code of format is exchanged in between the vehicles within an agreed upon frequency band. On receipt of valid information within the perimeter, they are acknowledged with a 'handshake' for confirmation by the neighboring vehicular system.

As an infrastructure to create such communication, it is required to create an air interface, protocols and applications required.

The communication with infrared and millimeter waves are usually directional. Therefore, they are not preferred in this study for IVC usage. Besides, those with VHF and microwaves are of broadcast type. Although VHF waves (e.g. 220[MHz])

have been used because of their long communication distance, the mainstream nowadays is microwaves. It is convincing that the most suitable wave medium for the kind of application as this study requires microwave [13].

A significant challenge in IVC networks would be the implementation of suitable routing protocols so as to ensure successful data packet delivery and lower packet delivery delay [2]. The multiple access schemes for the protocols are preferable here. In order to avoid destructive interference with already established channels, the protocols need to transmit additional information to let all nodes be aware of the status of each slot. Under appropriate code assignment and spreading–code schemes, the primary collisions i.e., two nodes with the same code try to access the channel together, can be avoided. For avoiding multi-access interference (MAI) leading to secondary collisions at a receiver, the channel is split into control and data channels. RTS/CTS (Request to Send/Clear to send) are transferred over control channels to let all potentially interfering nodes be aware of the channel status [13].

1.6 Modern Sensor Technologies for Smart Vehicular Systems

Each vehicular sub-system is endowed with most modern surveillance and navigational equipment and sensors for detecting information related to effective collision avoidance. Each IVC unit onboard involves in exchange of information between the vehicles [14]. The speed of the vehicle is measured by speedometer onboard [29]. Laser range scanners are used to detect the distance to each and every moving and non-moving obstacle around and on-path of each vehicle [9], [10]. The navigation sensors based on Differential Global Positioning System (DGPS) detect the latitude/longitude information of each vehicular system [1], [26]. The required electronics and software are integrated for each module for error corrections. A digital compass onboard each vehicle detects the heading angle of it [26]. It is common practice to integrate GPS with inertial sensors and possibly other motion sensors to bridge GPS outage gaps and enhance the system integrity. The technology exists today for GPS receivers with processors and electronics contained on a single chip. Progress in the development of semiconductor-processing technology using micro-electromechanical systems (MEMS) has led to the introduction of new inertial sensors in the automotive industry [28].

1.7 Main Controller Studies

The main controller is synthesized using an Adaptive Neuro-Fuzzy Inference System (ANFIS) of Takagi-Sugeno type [12]. This option is preferred because it not only allows the proven ability of fuzzy logic in handling the areas of uncertainty and nonlinearities in real world systems but also gives way for the controller to adapt during the training process due to its neural network representation. The ANFIS controller has been trained offline with a satisfactory number of data sets.

There are two main ANFIS controllers onboard each of the two vehicular systems: one for the control of braking and the other for control of steering.

1.8 Changes in the Mode of Control and the Human Factors Considerations

The controller mode of the vehicle is normally in the Driver Control Mode. When specific conditions arise conducive for a probable collision as discussed in the algorithm, the mode of control of the vehicle can change into Collision Avoidance Mode. It is understood that the sudden changes like the above must not interfere with the normal driving habits. One potential solution is to give a constant visual feedback to the driver whenever a change in the control mode occurs. The driver is likely to get startled when a sudden change of mode of control occurs. But the most promising feature of the controller is that it can even intervene in avoiding a series of likely consecutive collisions which may have arisen due to incapability of the driver to respond positively to the changes of control modes. The most essential requirement of the driver to be efficiently interactive in the environment discussed is to get trained and customized to the sudden changes in the control modes by getting an enough exposure in a simulator environment, beforehand. With enough experience, the driver will be in a position to take the changes in modes with the normal sense [22].

1.9 Simulations and Results

The simulation study for the controller has been carried out in the Matlab/Simulink environment. This platform is heavily used in a wide range of applications, including signal and image processing, communications and control design etc. This high-level technical computing environment is well suited for algorithm development, data visualization, data analysis, numerical computation and

simulations. With the extensive add-on features of Matlab/Simulink, it allows development of technical solutions faster than with traditional programming languages, such as C/C++ and Fortran.

1.10 Prototype Studies/ Realization of Prototypes

In this study, two fully autonomous prototypes were developed with all the required sensory and communication capabilities. After assembling the platforms, each component was individually tested for their performance related to the intended performance. The communication between the prototypes and the central PC is through RF broadcasting. The prototype was intended to detect the obstacles as well as the other prototype in a verge of a collision. The obstacle detection was intended to be done with ultrasonic sensors. Detection of the other prototype was intended to be done with the relative distance information plus the heading information that are fed through the RF communication. The required software were developed and used with each component testing. The most important hardware component i.e., RF receiver/transmitter modules, were extensively tested for their expected performance, after developing some supporting circuits for communication with the central PC and RF modules onboard the prototypes.

CHAPTER 2

ANFIS: Adaptive Neuro-Fuzzy Inference Systems

2.1 Neuro-Fuzzy Hybrid Systems

Neuro-Fuzzy refers to hybrids of artificial neural networks and fuzzy logic. Hybridization of neuro-fuzzy results in a hybrid intelligent system that synergizes these two techniques by combining the human-like reasoning style of fuzzy systems with the learning and adaptive structure of neural networks. Neuro-fuzzy hybridization is widely termed as Fuzzy Neural Network (FNN) or Neuro-Fuzzy System (NFS) in the literature. Neuro-fuzzy system incorporates the human-like reasoning style of fuzzy systems through the use of fuzzy sets and a linguistic model consisting of a set of IF-THEN rules. The adaptive or learning capability comes from the strength of Neural Networks. One of the well known examples is ANFIS.

2.2 Fuzzy logic

In the Fuzzy Set Theory, if X is a collection of objects and denoted generally as x, i.e., $x \in X$, then X is referred to as the *universe of discourse* or simply as the universe. X may consist of discrete objects or continuous space. Then a fuzzy set A can be defined as,

$$A = \{(x, \mu_A(x)) \mid x \in X\}, \tag{1}$$

where $\mu_A(x)$ is called the membership function for the fuzzy set A. The membership function maps each element of X to a membership grade between 0 and 1, i.e.

$$\mu_A(x) : X \to [0,1]. \tag{2}$$

A fuzzy set is completely characterized by its membership function. There are a various classes of membership functions and are usually parameterized. In the following study of the ANFIS controller, the Gaussian membership function has been used [12].

A linguistic variable serves to summarize information and express it in terms of fuzzy sets instead of crisp numbers. This looks as an alternative approach to modeling human thinking. A fuzzy if-then rule assumes the form:

If x is A then y is B,

where A and B are linguistic variables. Often "x is A" is called the antecedent or premise and "y is B" is called the consequence or conclusion.

Defuzzification is referred to as the way of extracting a crisp value from a fuzzy set as a representative value. In the Takagi-Sugeno type fuzzy model, which is the type being used in the ANFIS neuro-fuzzy controller, has the form

If x is A and y is B then $z = f(x, y)$,

where A and B are fuzzy sets in the antecedent, while $z = f(x, y)$ is a crisp function in the consequent [12].

2.3 Artificial Neural Networks

An artificial neural network is a data processing system consisting of a larger number of simple, highly interconnected processing elements (neurons) in an inspired architecture. These processing elements are usually organized into a sequence of layers. Each of the connections between neurons has an adjustable weight.

Neural networks perform two major functions: learning and recall. Learning is the process of adapting the connection weights in an artificial neural network to produce the desired output. Recall is the process of accepting an input stimulus and producing an output response in accordance with the network weight structure. There are a number of training algorithms for the artificial neural networks. The feedforward neural networks operate in a simple straightforward manner. However, when there are feedback connections, either between neurons in the same layer or from the layer to an earlier layer, the process is much more complicated [25].

2.4 ANFIS Networks and Control

ANFIS is a hybrid neuro-fuzzy controller system. In other words, ANFIS belongs to a class of adaptive networks that are functionally equivalent to fuzzy inference systems.

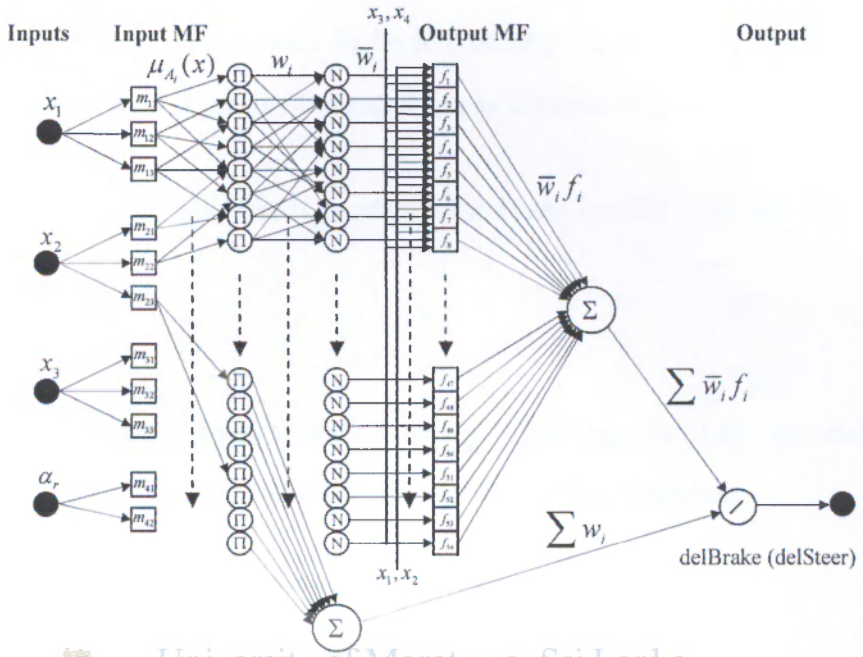


Fig. 1 ANFIS controller common structure used for the control of braking and steering where x_1, x_2, x_3, *and* x_4 are *CollisionCondition, RelativeDistance, MSSwitch,* and *SafetySpeedLimit* (for braking control)/*SteerDirection* (for steering control), respectively

The Adaptive Neuro-Fuzzy Inference System (ANFIS) architecture of linear Takagi-Sugeno type has been adopted in the main controller synthesis of this study. Two ANFIS systems are used: one for braking control and the other for steering control. ANFIS controller is a very effective neuro-fuzzy 'hybrid' scheme based on a fuzzy inference system framework with adaptive network functionality. The main advantage of using an ANFIS is that it enhances fuzzy controllers with self-learning capability for achieving the control objectives with near optimality.

ANFIS is a multilayer feed-forward network. The controller structure both common to the braking controller and the steering controller is shown in Fig. 1. There are 4 input variables and one output as characteristic of the general ANFIS.

Here, each node performs a particular function on incoming signals. That functional aspect is characterized with a set of parameters pertaining to each node.

The square nodes, which are commonly referred to as adaptive nodes, have parameters while the circular nodes (fixed nodes) have none. The collection of adaptive parameters is referred to as the parameter set of the adaptive network. To achieve the desired input-output mapping, these parameters are updated with the given training data.

Let the output of the i th node in layer l be $O_{l,i}$. The ANFIS architecture adopted here consists of 5 layers in its construction as shown in Fig. 1.

Layer 1: Every node i in this layer is an adaptive node and has the node function

$$O_{1,i} = \mu_{A_i}(x_k) \tag{3}$$

where $x_k; k = 1, \cdots, 4$ is the input to node i and A_i is the linguistic label associated with this node. We use Gaussian membership functions (MFs). Therefore

$$\mu_{A_i}(x_k) = e^{-\frac{1}{2}\left(\frac{x_k - c}{\sigma}\right)^2} \tag{4}$$

where $\{c, \sigma\}$ is the parameter set with c and σ, respectively, representing the center and the width of the MF. Parameters in this layer are referred to as the premise parameters.

Layer 2: Every node in this layer labeled as Π is a fixed node whose output is the product of all incoming signals given by,

$$O_{2,i} = w_i = \mu_{A_j}(x_1)\mu_{B_k}(x_2)\mu_{C_m}(x_3)\mu_{D_n}(x_4) \tag{5}$$

$i = 1, 2, ..., 54;\ j, k, m = 1, 2, 3;\ n = 1, 2.$

The node output represents the firing strength of the rule.

Layer 3: Every node in this layer is a fixed node labeled N. The i th node calculates the ratio of the i th rule's firing strength to the sum of all rules' firing strengths given by

$$O_{3,i} = \overline{w}_i = \frac{w_i}{\sum_i w_i}; \quad i = 1, 2, ..., 54. \tag{6}$$

The outputs of this layer are the normalized firing strengths.

Layer 4: Every node i in this layer is an adaptive node with a node function

$$O_{4,i} = \overline{w}_i f_i = \overline{w}_i (p_i x_1 + q_i x_2 + r_i x_3 + s_i x_4 + t_i) \tag{7}$$

where $\overline{w}_i$ is the normalized firing strength from layer 3, i = 1, 2, ..., 54, and $\{p_i, q_i, r_i, s_i, t_i\}$ is the parameter set of this node. Parameters in this layer are referred to as consequent parameters.

Layer 5: The single node in this layer is a fixed node labeled Σ, which computes the overall output as the sum of all incoming signals given by

$$O_{5,i} = \sum_i \overline{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i}. \tag{8}$$

Here in this study, ANFIS is used with the effective hybrid learning rule. With this learning algorithm, node outputs go forward until level 4 and the consequent parameters are identified by the least squares method. In the backward pass, the error signals propagate backwards and premise parameters are updated by gradient decent. This hybrid approach converges much faster than the original pure back-propagation method because the latter reduces the search space dimensions [12].

2.4.1 Hybrid Learning Algorithm

The hybrid learning algorithm, consisting of Least Squares Method for the forward passing and the Gradient Decent Method for the back propagation, is used to train the ANFIS. There are several ways of combining gradient decent method and the least-square method.

2.4.1.1 Least Squares Method

Using the matrix notation, the set of equations obtained from the regression function substituting training data pairs can be written as,

$$\mathbf{A}\theta = \mathbf{y}, \tag{9}$$

where, $\mathbf{A}$ is an $m \times n$ matrix, consisting of the known functions of the input data, θ being $n \times 1$ unknown parameter vector, and the $\mathbf{y}$ being the $m \times 1$ output vector.

Equations (9) can be modified incorporating an error vector $\mathbf{e}$:

$$\mathbf{A}\theta + \mathbf{e} = \mathbf{y} \tag{10}$$

The sum of squared error can be defined by,

$$E(\theta) = \sum_{i=1}^{m} (y_i - a_i^T \theta)^2 = \mathbf{e}^T \mathbf{e} = (\mathbf{y} - \mathbf{A}\theta)^T (\mathbf{y} - \mathbf{A}\theta) \tag{11}$$

The above squared error minimized for certain θ will give as an estimator. The root mean square error is straightforward to obtain from the equation (11).

2.4.1.2 Gradient Decent Method

The gradient-based optimization techniques are capable of determining search directions according to an objective function's derivative information. Here, a real-valued objective function E is defined on an n-dimensional input space, $\boldsymbol{\theta} = [\theta_1, \theta_,, \ldots, \theta_n]^T$. Finding a minimum point $\boldsymbol{\theta} = \boldsymbol{\theta}^*$ that minimizes $E(\boldsymbol{\theta})$ is of primary concern. In the iterative decent methods, the next point $\boldsymbol{\theta}_{k+1}$ is determined by a step down from the current point $\boldsymbol{\theta}_k$ in the direction vector **d:**

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \eta_k \mathbf{d}_k \quad (k=1,2,3, \ldots), \tag{12}$$

where η is the learning rate. η is determined by line search methods.

The next point should satisfy the following inequality:

$$E(\boldsymbol{\theta}_{k+1}) = E(\boldsymbol{\theta}_k + \eta \mathbf{d}) < E(\boldsymbol{\theta}_k). \tag{13}$$

The direction **d** is determined on the basis of the gradient of an objective function E.

2.5 Summary

In this chapter, the Adaptive Neuro-Fuzzy Inference System (ANFIS) is comprehensively explained in step-by-step manner. The learning algorithms of ANFIS, i.e., least squares method and the gradient decent method, have also been discussed here.

CHAPTER 3

Auxiliary Functions: Synthesis of Controller

3.1 Vehicular Controllers

There are two main vehicular onboard controllers: braking controller and steering controller. Both the controllers are based on the ANFIS and each are having four input variables and one output variable.

The primary sensory information cannot be straightaway fed into the controllers. The controllers need some secondary level information, as well. Therefore, in order to process the primary sensory information to get the secondary level information, some auxiliary functions are used. Fig. 2 gives the schematic diagram for the main controllers and the auxiliary functions.

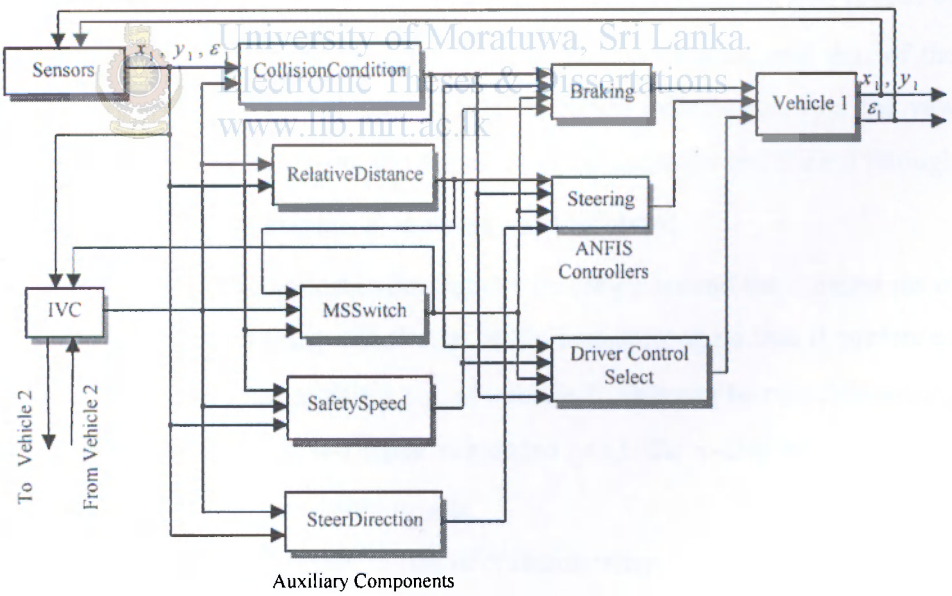


Fig. 2 Control system block diagram and information flow (vehicle-1)

3.2 Auxiliary Functions

Fig. 2 shows the auxiliary functions that are used to feed information to the two main controllers. These auxiliary functions process the primary sensor information and other related data to give the secondary level information. The following description explains in detail about each of the auxiliary function being used.

3.2.1 Collision Condition Function

This function is to detect whether the adjacent vehicle, irrespective of the distance between them, is in line of collision with the selected vehicle. The relative velocity paradigm is extensively used in determining this [17]. Here, a geometric construction as given in Fig. 3 is used for the analytical work. The points, O_1 and O_2 respectively, give the approximate centers of gravity (c.o.g.) of the vehicle-1 and vehicle-2. The Euclidean distance between O_1 and O_2 gives the relative distance between the two vehicles. (Relative Distance Function gives more details).

Let V_R be the velocity of vehicle-1 relative to vehicle-2 and the angle that V_R makes with the x-axis be η. V_R is calculated by each vehicle using the two speeds of the vehicles. Self speed is obtained using the speedometer onboard and that of the adjacent vehicle through IVC [14]. The vehicle direction information, i.e., the yaw angles, ε_1 and ε_2, are measured using the onboard digital compass and shared through IVC. When V_R, ε_1 and ε_2 are known, η can easily be calculated.

r is the radius of the smallest circle that can be drawn around the dimensions of twice the size of the participating vehicles. ψ is the half-apex angle that is subtended by the given circle at the approximate c.o.g. of vehicle-1. This can be calculated using r and the distance, O_1O_2. ϕ is the angle subtended w.r.t. the x-axis by O_1O_2. The Relative Distance Function gives more details.

The vehicles are identified as merely in line of collision when:

$$\phi + \psi > \eta \tag{14}$$

and

$$\phi - \psi < \eta \tag{15}$$

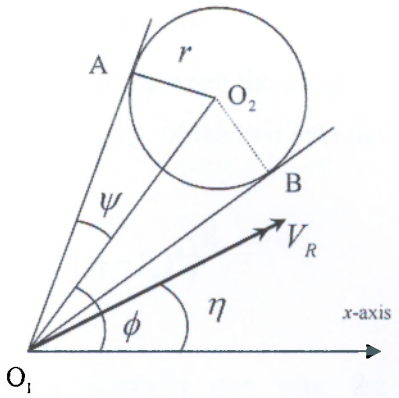


Fig. 3 Detection of collision condition, for instance, from the perspective of vehicle-1.

In other words, the above condition is satisfied when the vector V_R falls within the 'collision cone' of AO_1B. The collision condition can be explained in rigorous terms as follows. An extended collision cone is defined by adding $\pi/18$[rad] from either side of the collision cone, AO_1B. If the collision cone area is denoted as, $C_C = [\phi - \psi, \phi + \psi]$, then the extended collision cone can be described as, $C_{C+} = [\phi - \psi - \pi/18, \phi + \psi + \pi/18]$.

If $\eta \in C_C$, it can be said, the two vehicles are *'in line of collision'*. But, if $\eta \in C_{C+} \cap C'_C$, where C'_C is the complement of C_C, the situation is identified as *'in line of likely collision'*. To that end, if $\eta \notin C_{C+}$, then it will be concluded the vehicles are *'not in line of collision'* whatsoever.

These different situations are now quantified and then be fuzzified to be used by the ANFIS controllers. The situation, 'in line of collision' is assigned the value of '-3'. The situations 'in line of likely collision' and 'not in line of collision' are assigned '-1.5' and '0', respectively. These values are used to initialize the centers of the Gaussian membership functions in the Fuzzy Logic.

The Fig. 5(a) and 7(a) give the membership functions before being trained for the braking controller and the steering controller. Fig. 6(a) and 8(a) give the membership functions after being trained for the braking and steering controllers, respectively.

3.2.2 Relative Distance Function

This is the Euclidean distance between the c.o.g. s of the two vehicles as given in terms of the coordinates of the c.o.g. of the two vehicles:

$$RelativeDistance = \left((x_1 - x_2)^2 + (y_1 - y_2)^2 \right)^{1/2}. \quad (16)$$

Relative Distance, R_D , normally can take the range, $R_D = [0, \infty]$. But when $R_D < 5.0\,[\mathrm{m}]$, the two vehicles are considered to have *'collided'*. Here, the distance, r = 5.0 [m] is the minimum radius of the circle, that can be drawn around twice the size of the vehicle (Collision Condition Function gives more details).

Remark 3.1: For simplicity, the two vehicles under this study are assumed to have same dimensions.

Relative distance can easily be calculated using positional data obtained through the GPS sensors [1], [26]. The positional details of the other vehicle are received through the IVC. It can reasonably be assumed that the absolute position error of GPS is cancelled off when calculating the relative position between two GPS receivers.

3.2.3 Safety Speed Limit Function

The safety speed of the vehicle is defined in terms of the 'braking critical distance'. The speed of the vehicle is said to be safe when the relative distance between the vehicles is more than the braking critical distance.

The braking critical distance, d_{br}, is defined as follows for the three categories of collisions:

$$d_{br} = \begin{cases} \frac{1}{2\alpha}\left(v^2 + \left(v - v_{rel}\right)^2\right), & \text{for head-on collisions} \\ \frac{1}{2\alpha}\left(v^2 - \left(v - v_{rel}\right)^2\right), & \text{for rear-end collisions} \\ \frac{1}{2}\frac{v^2}{\alpha}, & \text{otherwise} \end{cases} \quad (17)$$

Where, v is the speed of the vehicle, v_{rel} is the relative velocity of the vehicle w.r.t. the other, and α is the maximum deceleration of the vehicle.

A clear criterion for correct identification of the three different collision scenarios, mentioned in (17), is important. For correct identification of head-on and rear-end collision cases, a cone area is defined using the forward longitudinal-axes directions of the two vehicles.

If $|\varepsilon_1 - \varepsilon_2| \in A_{ho} = [11\pi/12, 13\pi/12]$, the case is taken as a head-on collision, where $\varepsilon_1, \varepsilon_2$ are yaw angles of the vehicle-1 and vehicle-2, respectively.

If $|\varepsilon_1 - \varepsilon_2| \in A_{re} = [-\pi/12, \pi/12]$, the collision scenario is a rear-end collision. Otherwise, i.e., when $|\varepsilon_1 - \varepsilon_2| \notin (A_{re} \cup A_{ho})$, it is considered to be one of side-end collision situations.

The vehicle will be in the 'manual (driver controlled) mode' if the relative distance, $R_D > d_{br} + k_s$, where k_s is a constant. But if $R_D \leq 6$ [m], the mode will always be switched to *collision avoidance* mode, irrespective of the other conditions. If $R_D < d_{br} + k_s$, i.e., when the speed is 'unsafe' (while $\eta \in C_{C+}$ when $R_D \leq 25$ [m]), the vehicle is decelerated until $R_D > d_{br} + k_s$, i.e. until the 'safe' speed condition is satisfied. Once again, it can be said that if $R_D > 25$ [m], the mode of control of each vehicle will unconditionally be in the *driver control mode.*

Two Gaussian membership functions (MFs), 'safe speed' MF with '1' being the center and 'unsafe speed' MF with '0' being the center are used to fuzzify the "Safety Speed" function. This function is used only with the braking controller. Fig. 5(d) gives the membership function for the case before being trained, while Fig. 6(d) gives the membership function for the trained case.

3.2.4 Master-Slave Function

This function identifies the control hierarchy of the two vehicles. There are a number of criteria to determine which vehicle is the 'master', and thereby, which is the 'slave'. In this study, the vehicle that moves faster at the moment will be taken as the master for simplicity. For instance, one may adopt other criteria to better suit the locations of the vehicles (say, for instance, at a roundabout) based on GPS and a geographic information system (GIS). The Master-Slave block generates a signal that is fed to the ANFIS controller for decision making. Within each vehicle, the Master-Slave statuses are calculated, exchanged and agreed upon with an acknowledgement by each vehicle with 'handshake' signals through IVC.

As given earlier, the vehicles are said to be in definite collision if $R_D < d_{br} + k_s < 25$ [m], and the relative velocity of the vehicles are inside the collision cone, i.e., $\eta \notin C_{C+}$. A special case arises, when $R_D < 6.0$ [m], where the control is always switched on to *collision avoidance mode*. Whenever, the vehicle is in *collision avoidance mode*, the 'master' or 'slave' states will soon be realized by the vehicles.

Fig. 4 gives a state diagram in which different states of the vehicle and conditions for changes from one state to another are given. As discussed in the Safety Speed Function, v_1 and v_2 are the speeds of the vehicle-1 and vehicle-2, respectively, and d_{br} is the braking critical distance.

MSSwitch function has three variables, viz., master, slave, and driver control. In order to quantify these variables, 'master' is assigned '1' while the 'slave' is assigned with '0'. The value given to 'driver control' is '5'. As discussed before, these values are taken as the initial centers of the Gaussian membership functions of the corresponding fuzzy variables. The Fig. 5(c) and Fig. 7(c) give the membership functions before the training process for the braking and steering controllers,

respectively. Fig. 6(c) and 8(c) give the membership functions after they have been trained, for the two controllers, respectively.

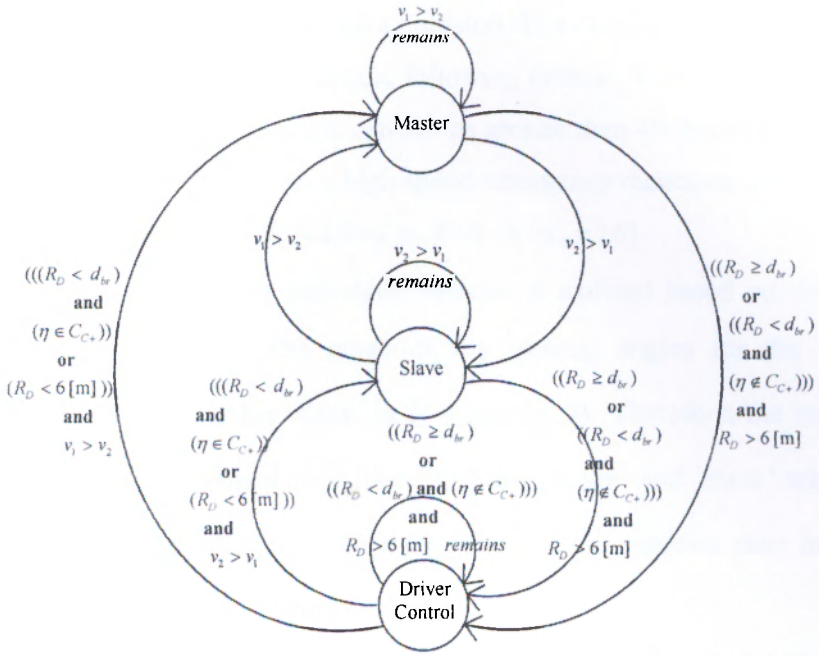


Fig. 4 State diagram for change of control for vehicle-1(vehicle-2)

3.2.5 Steering Direction Function

This function enables to clearly identify the steering directions for both 'master' and 'slave' vehicles in order to avoid a collision. The demarcated steering regions by each vehicle are identified based on the following criteria. Without loss of generality, the absolute value of steering angle can not be greater than 40 degrees [8]. In order to reduce the chances of rollover in a high speed emergency maneuver, in our study, the range for the steering angle, δ, is taken as $\delta = [-\pi/6, \pi/6]$.

The steering of the master and slave vehicles is realized based on the following criteria. If $|\varepsilon_1 - \varepsilon_2| \leq \pi/2$, the range of the steering angles for the 'master' is $\delta_m = (0, \pi/6]$ and that for the 'slave' is $\delta_s = [-\pi/6, 0)$. Therefore, the vehicles steer in opposite directions. When $|\varepsilon_1 - \varepsilon_2| > \pi/2$, both 'master' and 'slave' take the same range, i.e., $\delta_m = [0, \pi/6]$ and $\delta_s = [0, \pi/6]$. That is, the vehicles steer in the 'same direction'. It is obvious that always $\delta_s, \delta_m \subset \delta$.

Two Gaussian membership functions (MFs), 'same direction' MF with '0' as the center and 'opposite direction' MF with '1' as the center are used to fuzzify the "SteerDirection" function.

Remark 3.2: The yaw angles are always taken as 4-quadrant angles. The Collision Condition Function gives more details for obtaining yaw angles.

3.3 Controller Algorithm

The overall control procedure that includes the high level logic control parts for collision avoidance control, is elaborated as follows:

If $R_D \in (25, 40]$ [m],

Start checking IVC information. The vehicular systems alert on this condition. But no action is taken, i.e., $F^M, F^S \in \varnothing \Leftrightarrow F^D \in F_n$.

Else-If $R_D \in (6, 25]$ [m],

when, $R_D < d_{br} + k_s$ **and** $\eta \in C_{C+}$,

$F^M \in F_B$ and $F^S \in F_B$. But $F^M, F^S \notin F_S$.

when, $R_D < d_{br} + k_s$ **and** $\eta \notin C_{C+}$,

$F^M, F^S \in \varnothing \Leftrightarrow F^D \in F_n$.

when, $R_D \geq d_{br} + k_s$ **and** $\eta \in C_{C+}$,

$F^M, F^S \in \varnothing \Leftrightarrow F^D \in F_n$.

Else $R_D \in [0, 6]$[m], $\forall \eta, \forall d_{br}$,

$F^M, F^S \in F_B$ and,

$F^M, F^S \in F_S$. i.e.,

If, $|\varepsilon_1 - \varepsilon_2| < \pi/2$, steering angle of master, $\delta_m = [0, \pi/6]$ and that of slave, $\delta_s = [-\pi/6, 0]$.

Else, i.e., $|\varepsilon_1 - \varepsilon_2| > \pi/2$, for master and slave, $\delta_m = [0, \pi/6]$ and $\delta_s = [0, \pi/6]$.

Note 3.1: The set of selectable control functions in a vehicle is given by $F_n = \{F_B, F_S\}$, where F_B, F_S are the braking and steering functions, respectively. F^M and F^S give the control functions selected by the master and the slave, respectively, i.e., $F^M, F^S, F^D \subset F_n$, where F^D is the driver control function.

Note 3.2: The selected states for the collision avoidance criteria are, S_S = {Master, Slave}, (normally main states of the vehicles, S = {Master, Slave, DriverControl}, where $S_S \subset S$).

Note 3.3: $\delta_s, \delta_m \subseteq \delta = [-\pi/6, \pi/6]$, where δ is the normal range of steering angles.

3.4 Summary

In this chapter, a detailed description on the auxiliary functions, i.e., the secondary level functions which were used to process the primary sensory data, is given. At the end of the chapter, the main controller algorithm is elaborated.

CHAPTER 4

Synthesis of Controller and Training

4.1 Braking Controller

The following describe the aspects of the ANFIS Braking controller.

4.1.1 Fuzzy Input Membership Functions (Before Training)

The input variables for the ANFIS braking controller are *CollisionCondition, RelativeDistance, MSSwitch,* and *SafetySpeedLimit*. These are kind of secondary variables obtained using primary sensory outputs as explained under auxiliary functions. The output *delBrake* [m/s²] is the deceleration of the vehicle that can be converted into the braking torque. The relation between the brake pressure and the desired braking torque is rather straight forward [23].The Fig. 5(a) through Fig. 5(d) give the ANFIS membership functions before they are being trained for the braking controller.

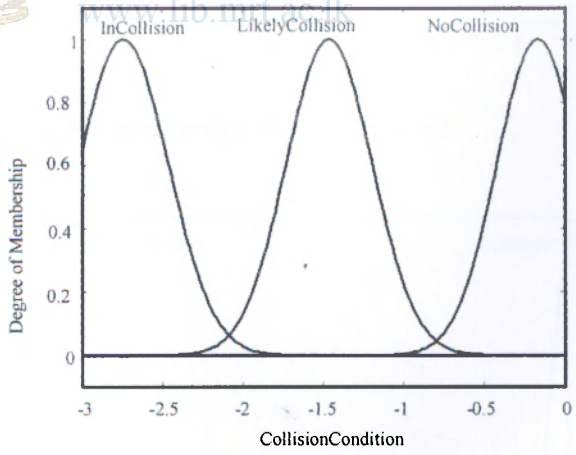


Fig. 5(a) Membership function of *CollisionCondition*

Fig. 5(a) gives the membership functions for the input variable, *CollisionCondition*, which identifies different categories in terms of collision of the vehicle. The set of linguistic variables is: {*InCollision, LikelyCollision, NoCollision*}.

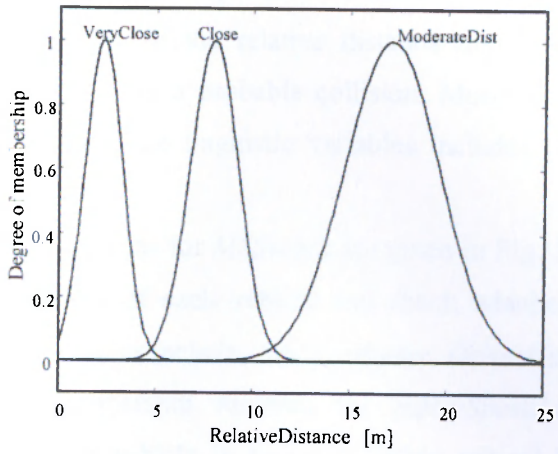


Fig. 5(b) Membership function of *RelativeDistance*.

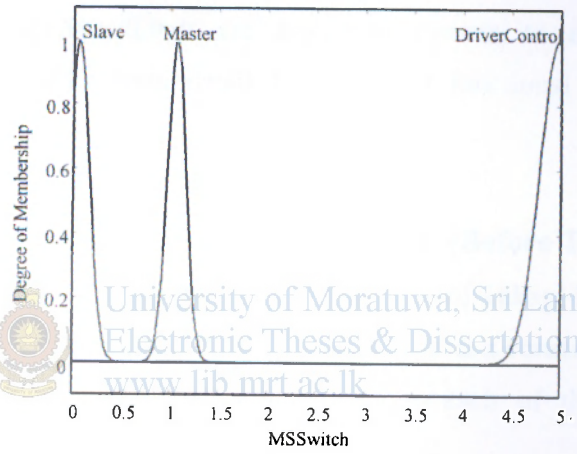


Fig. 5(c) Membership function of *MSSwitch*

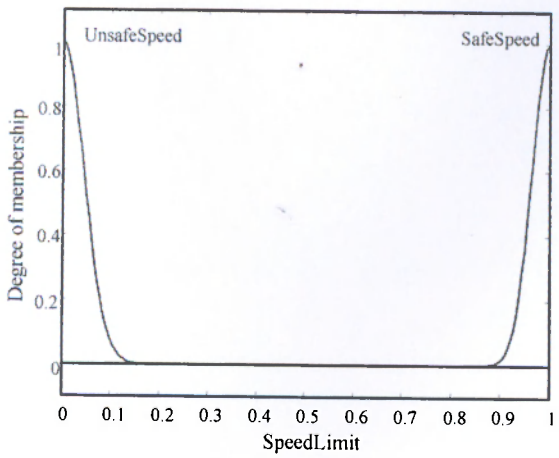


Fig. 5(d) Membership function of *SpeedLimit*

Fig. 5(b) gives the membership functions for *RelativeDistance* that identifies different categories in terms of the relative distance in meters between the two vehicles in consideration, under a probable collision. More details are given under Relative Distance Function. The linguistic variables include: {*VeryClose, Closeby, ModerateDistant*}.

The membership functions for *MSSwitch* are given in Fig. 5(c). It identifies the different hierarchical stages of each vehicle and check whether it is under driver control. The linguistic variables include: *{Slave, Master, DriverControl}*

Fig. 5(d) gives the membership function for *SafetySpeedLimit* that identifies categories of speed of the vehicle in terms of safety critical braking limits. The linguistic variables include: {*UnsafeSpeed, SafeSpeed*}.

How the universes of discourse for the variables of *CollisionCondition, MSSwitch* and *SafetySpeedLimit* are arbitrarily chosen to reflect the relative differences of each MFs were detailed in relevant functional descriptions under auxiliary functions.

4.1.2 Fuzzy Output Membership Functions (Before Training)

The coefficients of output membership functions of Takagi-Sugeno type looked like [0, 0, 0, 0, c], in generic format, before training. 'c' is a constant generated by the Matlab/Simulink environment and is different for each of the 54 membership functions.

4.1.3 Training of the Braking Controller

The training of the ANFIS braking controller was done in the Matlab/Simulink environment.

4.1.4 Fuzzy Input Membership Functions (After Training)

Fig. 6(a) through Fig. 6(d) give the ANFIS trained membership functions for the above input variables.

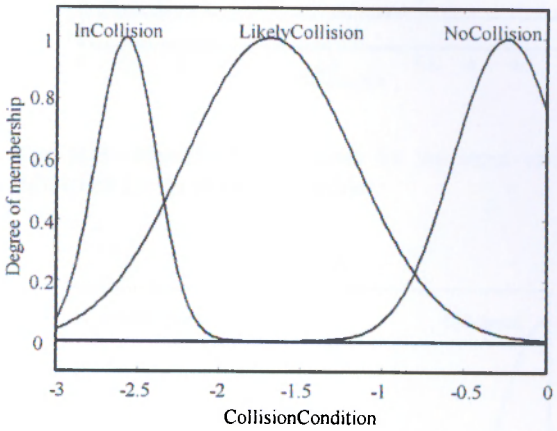


Fig. 6(a) Membership functions for the input variable *CollisionCondition* for the braking controller.

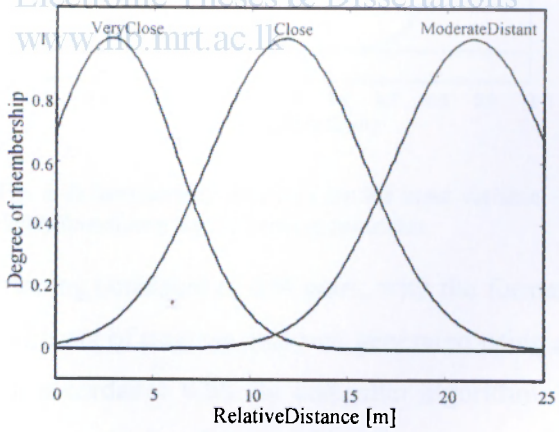


Fig. 6(b) Membership functions for the input variable *RelativeDistance* for the braking controller.

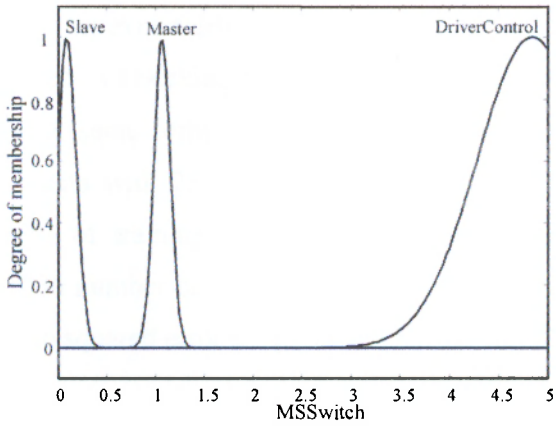


Fig. 6(c) Membership functions for the input variable *MSSwitch* for the braking controller.

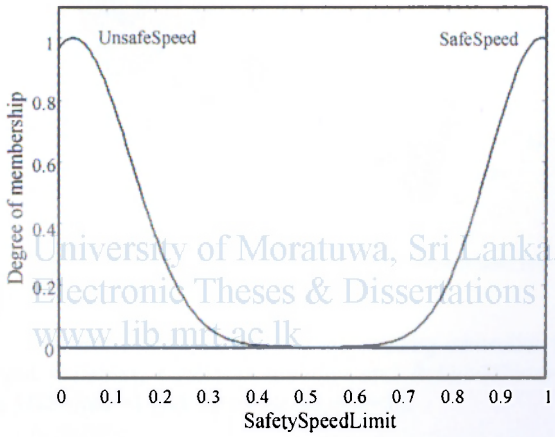


Fig. 6(d) Membership functions for the input variable *SafetySpeedLimit* for the braking controller.

The set of data for training consisted of 650 pairs, with the format of four input data plus one output data. This set of training data was generated using a program in a spreadsheet environment in accordance with the controller algorithm. The ranges of the input and output variables were taken into consideration. For each set of program selected input data, the output variable value was selected by intuition in accordance with the proposed control algorithm. For eliminating any possibility to miss data in critical boundaries in the main range of the input variable, an index of all possible combinations was used as guidance. The difference between the input data values within a critical boundary was determined by the fact of the requirement of the

number of data sets for training. The generated data set enabled to train the ANFIS so that it mimics the behaviour of an expert driver.

The model was validated by checking for model over-fitting with a formulated checking data set, having the same format. The checking data set was formed by adding 5% of the training data with 25% deviation, 10% training data with 20% deviation and the rest (85% of training data) with 10% deviation. Training and validation were done for 1000 number of epochs. The training and validation graphs with the changes in root mean squared error are given in the Fig. 6(f).

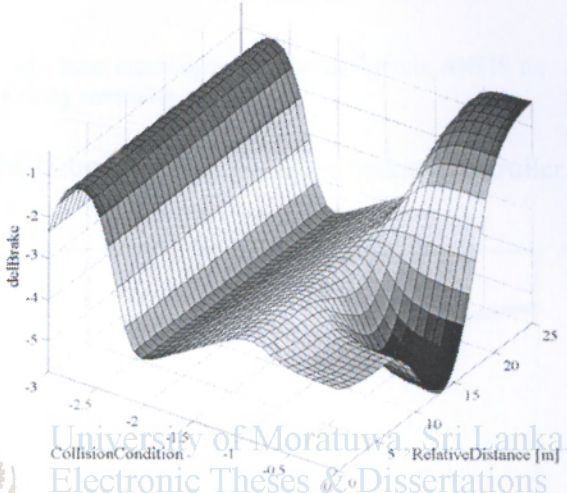


Fig. 6(e) Output control surface for the braking controller for the input variables, *CollisionCondition* and *RelativeDistance* (when *MSSwitch* =1 and *SafetySpeedLimit*= 0).

Fig.6(e) gives good continuity as is normally expected from a Takagi-Sugeno fuzzy controller.

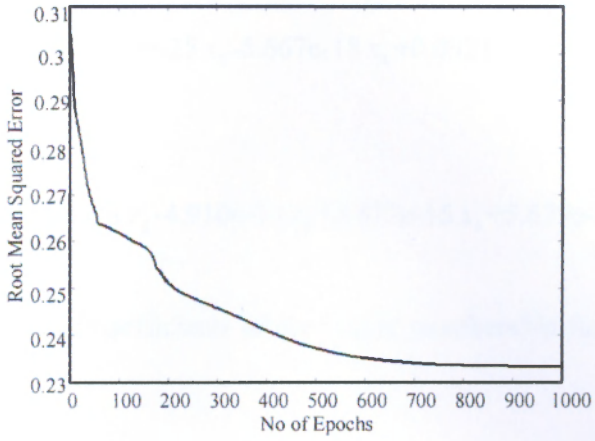


Fig. 6(f) Root mean squared error for training ANFIS for the braking controller.

Fig. 6(f) gives the ANFIS training history for the braking controller.

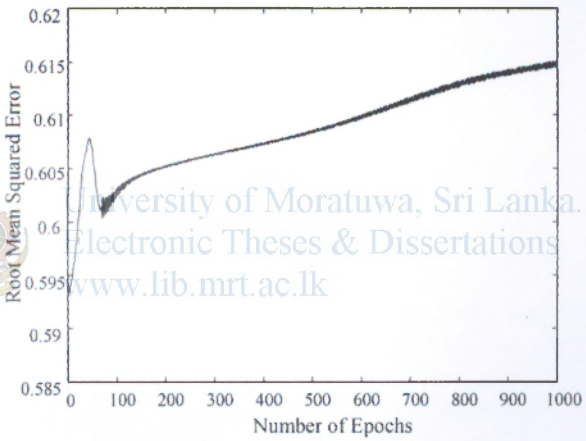


Fig. 6(g) Root mean squared error for checking ANFIS for the braking controller.

Fig. 6(g) gives the ANFIS checking and validation history for the braking controller.

4.1.5 Fuzzy Output Membership Functions (After Training)

Some selected output membership functions of Takagi-Sugeno type for the braking controller after being trained are given below:

$f_1 = 1.85\, x_1 + 1.566\text{e-}8\, x_2 - 2.79\text{e-}21\, x_3 - 2.1894\text{e-}15\, x_4 - 1.48$

$f_2 = 0.0019\,x_1 - 0.039\,x_2 - 5.655\text{e-}23\,x_3 - 4.72\text{e-}17\,x_4 - 4.225$

$f_3 = -0.311\,x_1 - 2.892\text{e-}6\,x_2 - 5.577\text{e-}25\,x_3 - 5.667\text{e-}18\,x_4 + 0.0921$

. . .

. . .

$f_{54} = -1.710\text{e-}15\,x_1 + 4.911\text{e-}18\,x_2 - 4.910\text{e-}14\,x_3 + 5.677\text{e-}15\,x_4 + 5.679\text{e-}15$

The complete list of coefficients of the output membership functions are given in the Appendix A.

4.2 Steering Controller

The following describe the aspects of the ANFIS Steering controller.

4.2.1 Fuzzy Input Membership Functions (Before Training)

The input variables for the ANFIS steering controller are: *CollisionCondition, RelativeDistance, MSSwitch and SteeringDirection*. These variables are formed as secondary variables from the primary sensory input variables before they are fed into the controller as inputs. The controller output is *delSteer* [rad], which is the steering angle command to the vehicle.

The details of ANFIS input membership functions of the input variables of the steering controller are given in Fig. 7(a) through Fig. 7(d), before training.

Fig. 7(a) gives the input membership function for *CollisionCondition* which is used to identify the collision conditions in the same way as given under the braking controller. In the same way, Fig 7(b) and 7(c) identify the *RelativeDistance* and *MSSwitch*, respectively. All these three membership functions are in the same form as in the braking controller.

Fig. 7(d) gives the membership functions for *SteeringDirection*, which takes into account the direction of steering to be adopted by the vehicle at a verge of a collision. The universe of discourse here is a discrete space arbitrarily chosen to separately identify the two regions for the steering direction. The linguistic variables are: *{SameDirect, OppDirect}*.

As given under the braking controller, for the input variables, *CollisionCondition, MSSwitch*, and *SteeringDirection,* the universes of discourse are arbitrarily chosen to reflect the relative differences of each MF in contrast to the *RelativeDistance* variable which is a straightforward distance measurement.

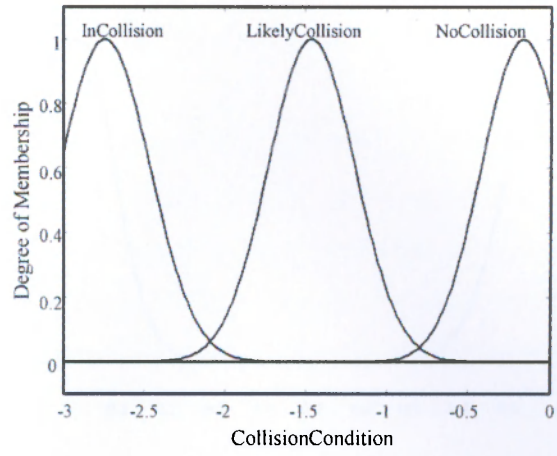


Fig. 7(a): Membership function of *CollisionCondition*

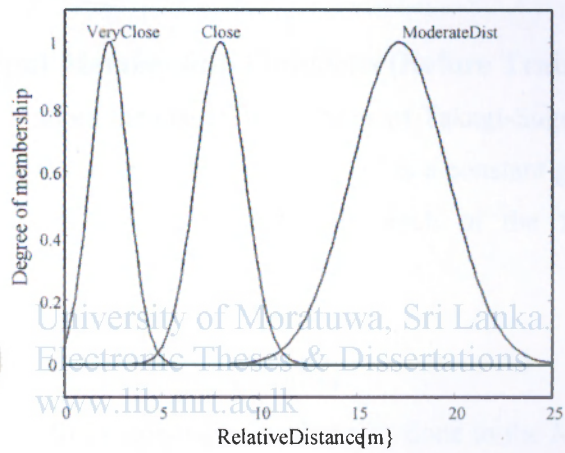


Fig. 7(b): Membership function of *RelativeDistance*.

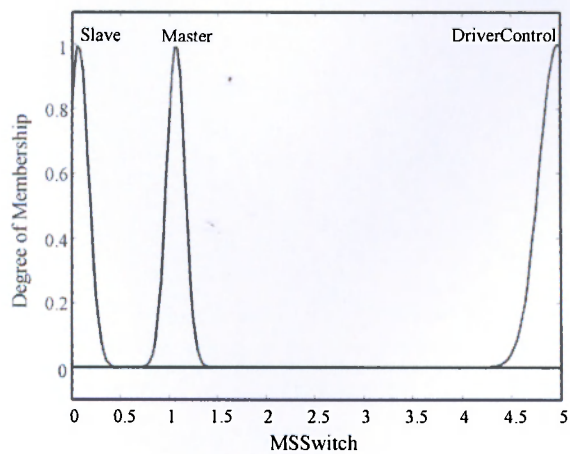


Fig.7(c): Membership function of *MSSwitch*

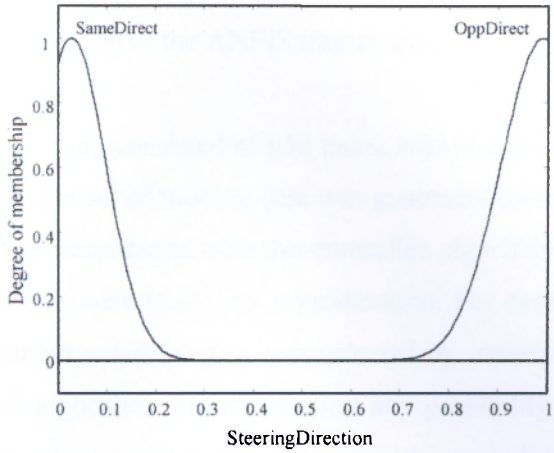


Fig. 7(d): Membership function of *SteeringDirection*

4.2.2 Fuzzy Output Membership Functions (Before Training)

The coefficients of output membership functions of Takagi-Sugeno type looked like [0, 0, 0, 0, c] in generic format, before training. 'c' is a constant generated by the Matlab/Simulink environment and is different for each of the 54 membership functions.

4.2.3 Training of the Steering Controller

The training of the ANFIS steering controller was done in the Matlab/Simulink environment.

4.2.4 Fuzzy Input Membership Functions (After Training)

Fig. 8(a) through Fig. 8(d) give the ANFIS trained membership functions for the above input variables.

The set of data for training consisted of 650 pairs, with the format of four input data plus one output data. This set of training data was generated using a program in a spreadsheet environment in accordance with the controller algorithm. The ranges of the input and output variables were taken into consideration. For each set of program selected input data, the output variable value was selected by intuition in accordance with the proposed control algorithm. For eliminating any possibility to miss data in critical boundaries in the main range of the input variable, an index of all possible combinations was used as guidance. The difference between the input data values within a critical boundary was determined by the fact of the requirement of the number of data sets for training. The generated data set enabled to train the ANFIS so that it mimics the behaviour of an expert driver.

The model was validated by checking for model over-fitting with a formulated checking data set, of the same format as the training data set. The checking data was formed by adding 5% of the training data with 25% deviation, 10% training data with 20% deviation and the rest (85% of training data) with 10% deviation. Training and validation were done for 1000 number of epochs. The training and validation graphs with the changes in root mean squared error are given in the Fig. 8(f). As such, the training of braking and steering controllers are similar in form.

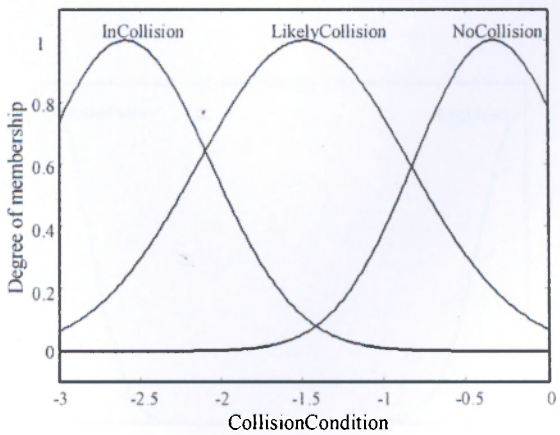


Fig. 8(a) Membership functions for the input variable *CollisionCondition* for the steering controller

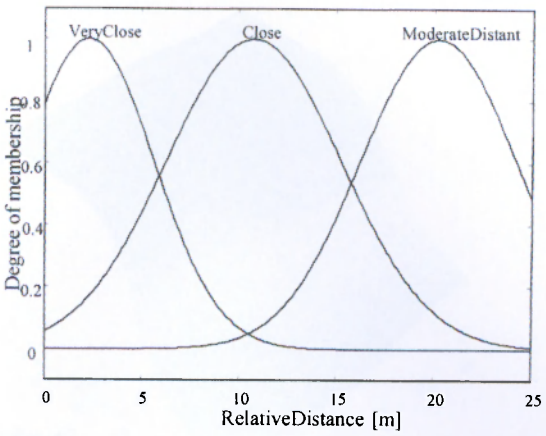


Fig. 8(b) Membership functions for the input variable *RelativeDistance* for the steering controller.

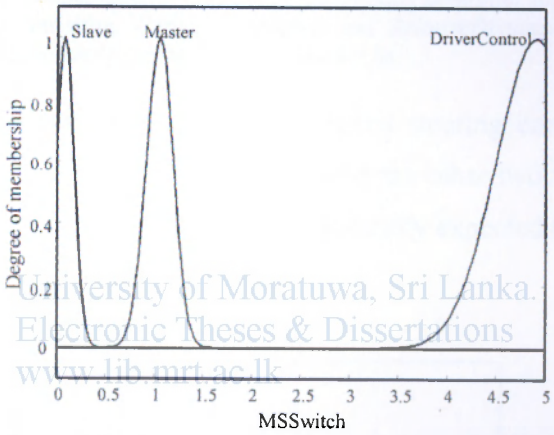


Fig. 8(c) Membership functions for the input variable *MSSwitch* for the steering controller.

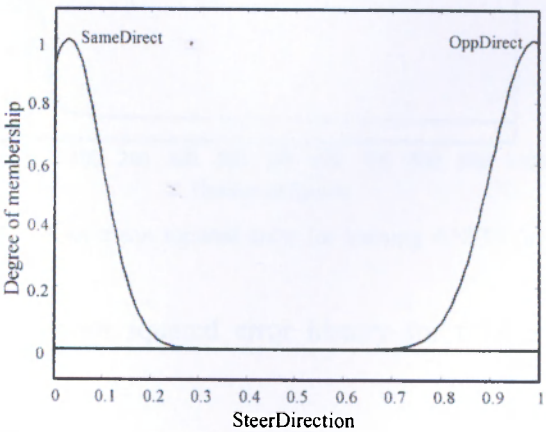


Fig. 8(d) Membership functions for the input variable *SteeringDirection* for the steering controller.

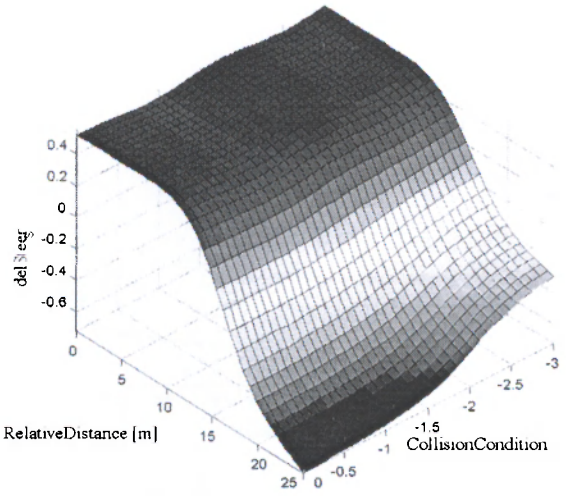


Fig. 8(e) Output control surface for the steering controller for the input variables, *CollisionCondition* and *RelativeDistance* (when *MSSwitch* =1 and *SteeringDirection* = 0)

Fig. 8(e) gives the control surface for the trained steering controller for the variables *CollisionCondition* and *RelativeDistance* with the other two variables fixed as given above. Fig.8(e) gives good continuity as is normally expected from a Takagi-Sugeno fuzzy controller.

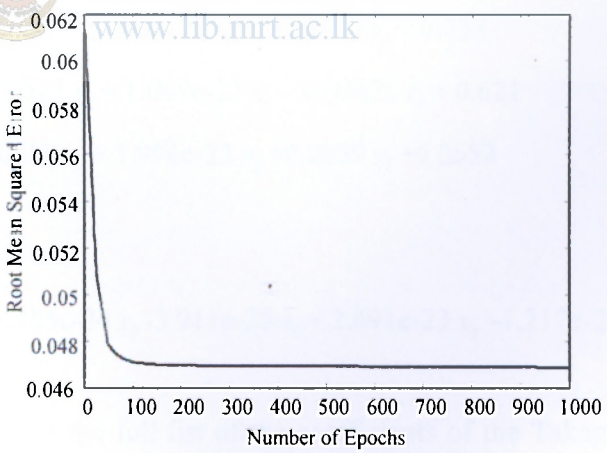


Fig. 8(f) Root mean squared error for training ANFIS for steering controller.

Fig. 8(f) gives the root mean squared error history for training of the ANFIS steering controller.

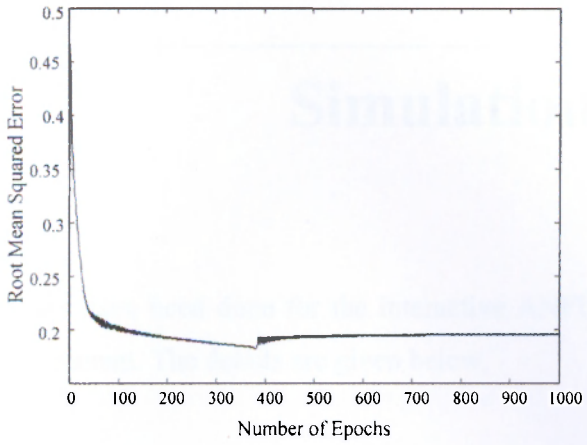


Fig. 8(g) Root mean squared error for checking ANFIS for steering controller.

Fig. 8(g) gives the root mean squared error history for checking for model validation of the ANFIS steering controller.

4.2.5 Fuzzy Output Membership Functions (After Training)

Following are some of the selected output membership functions of Takagi-Sugeno type for the steering controller after being trained:

$f_1 = -0.322\,x_1 + 0.0036\,x_2 + 5.798\text{e-}22\,x_3 + 0.00459\,x_4 + 0.233$

$f_2 = -0.0141\,x_1 + 0.00321\,x_2 + 1.069\text{e-}22\,x_3 + 0.00021\,x_4 + 0.621$

$f_3 = -0.151\,x_1 + 0.000659\,x_2 + 1.996\text{e-}23\,x_3 + 0.0039\,x_4 + 0.0652$

. . .

. . .

$f_{54} = 2.69\text{e-}24\,x_1 + 1.755\text{e-}24\,x_2 - 3.911\text{e-}25\,x_3 + 2.691\text{e-}23\,x_4 - 1.217\text{e-}23$

Appendix B gives the full list of the coefficients of the Takagi-Sugeno output membership functions for the steering controller.

4.3 Summary

In this chapter, all the Takagi-Sugeno type fuzzy membership functions for the controller are described. Even their changes after the training process have been elaborated.

CHAPTER 5

Simulation Studies

Simulation studies have been done for the interactive ANFIS controller in the Matlab/Simulink environment. The details are given below.

5.1 Vehicle Dynamics Model

Newton-Euler equations have been used to derive the below model of the vehicle. This mathematical model is to be adopted for the controller evaluation in the simulation studies. The main assumptions made in deriving the model are [23]:

1) Roll, pitch and bounce motions are negligible
2) The effect of suspension on the tire axels is negligible
3) Brake, throttle, and steering dynamics are discounted.

The first assumption is valid without appreciable loss in accuracy under typical and severe maneuvers for highway vehicles. The first two assumptions applicable to vehicle lateral motion control are assumed to be not considerably restrictive.

The Fig. 9(a) shows two reference frames, namely, world fixed frame $\{W\}$ and vehicle frame $\{V\}$. As bounce motion is not considered, all the reference frames are assumed to be in horizontal plane. The origin of the vehicle frame is defined at the center-of-gravity (c.o.g.) of the vehicle. The origin of the vehicle frame can be defined by an orthogonal basis, $\mathbf{e}_v = [i_v, j_v]^T$, with i_v taking along the longitudinal axis of the vehicle frame, and j_v along the lateral axis of the same frame.

It is possible to define the velocity of the vehicle's c.o.g. as,

$$\dot{r}^v = \dot{x}i_v + \dot{y}j_v = [\dot{x} \quad \dot{y}]\mathbf{e}_v \tag{18}$$

where $\dot{x}$ and $\dot{y}$ are the components of the vehicle's velocity along and perpendicular to the vehicle's longitudinal axis, respectively.

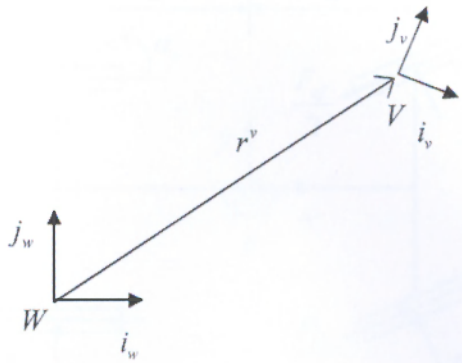


Fig. 9(a) Reference frames and position vectors

The acceleration of the vehicle is given by

$$\ddot{r}^v = [\ddot{x} \;\; \ddot{y}]\mathbf{e}_v + [\dot{x} \;\; \dot{y}]\dot{\mathbf{e}}_v \tag{19}$$

where $\dot{\mathbf{e}}_v = \boldsymbol{\Omega}\mathbf{e}_v$, with $\boldsymbol{\Omega}$ being the velocity tensor of $\{V\}$ w.r.t. $\{W\}$.At this point, it is to be noted that the speed of rotation of $\{V\}$ is equal to the yaw rate of the vehicle, $\dot{\varepsilon}$. $\boldsymbol{\Omega}$ can be approximated, for smaller angular displacements, as

$$\boldsymbol{\Omega} = \begin{bmatrix} 0 & \dot{\varepsilon} \\ -\dot{\varepsilon} & 0 \end{bmatrix}.$$

This results in

$$\ddot{r}^v = [\ddot{x} - \dot{\varepsilon}\dot{y}]i_v + [\ddot{y} + \dot{\varepsilon}\dot{x}]j_v. \tag{20}$$

The Newton-Euler equations governing the motion of the vehicle under the influence of external dynamics can be written as follows, with reference to the Fig. 9(b)

$$m(\ddot{x} - \dot{\varepsilon}\dot{y}) = F_{xr} + F_{xf}\cos\delta - F_{yf}\sin\delta - k_D\dot{x}^2 \tag{21}$$

$$m(\ddot{y} + \dot{\varepsilon}\dot{x}) = F_{yr} + F_{xf}\sin\delta + F_{yf}\cos\delta \tag{22}$$

$$I_z\ddot{\varepsilon} = aF_{xf}\sin\delta + aF_{yf}\cos\delta - bF_{yr} \tag{23}$$

where $\dot{x}$ and $\dot{y}$, respectively, are the longitudinal velocity (along vehicle's longitudinal axis) and the lateral velocity of the vehicle. And, $\dot{\varepsilon}$ is the yaw rate. The steering angle, δ, is assumed to be the same for both the front tires of the FWS (Front Wheel Steering) vehicle system. F_{xf} and F_{xr} are the longitudinal forces on the front and rear tires, and F_{yf} and F_{yr} are the tire lateral forces. The tire/road interaction forces are symmetrical about the vehicle's x-axis in magnitude. $k_D\dot{x}^2$ is the aerodynamic drag force.

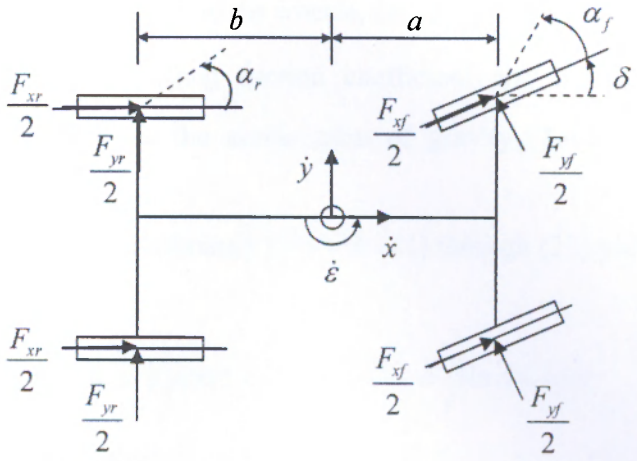


Fig. 9(b) Vehicle model

The tire lateral forces, F_{yf} and F_{yr} are due to the difference between the direction of each tire and the heading of each tire described by side slip angle , α, which is defined as

$$\alpha = \tan^{-1}\left(\frac{\dot{y}_s}{\dot{x}_g}\right) - \delta \tag{24}$$

where $\dot{y}_g$ and $\dot{x}_g = \dot{x} + \frac{c}{2}\dot{\varepsilon}$ respectively are the lateral and longitudinal velocities of the vehicle in the ground plane. Here, c is the track (length of the axle). With some mild assumptions, $\dot{x}_g \gg \dot{y}_g$ and $\dot{x} \gg \dot{\varepsilon}$, the front and rear side slip angles can be approximated as

$$\alpha_f = \frac{\dot{y} + a\dot{\varepsilon}}{\dot{x}} - \delta, \alpha_r = \frac{\dot{y} - b\dot{\varepsilon}}{\dot{x}} \tag{25}$$

respectively. Using a linear model of the tire/road interface, the tire lateral forces can be obtained as

$$F_{yf} = -C_f\alpha_f, \; F_{yr} = -C_r\alpha_r \tag{26}$$

where C_f and C_r are the front and rear cornering stiffnesses, respectively. Furthermore, the tire longitudinal forces F_{xf} and F_{xr} can be written as

$$F_{xf} = F - f\frac{b}{a+b}(mg - k_L\dot{x}^2) \tag{27}$$

$$F_{xr} = -f\frac{a}{a+b}(mg - k_L\dot{x}^2) \tag{28}$$

where F is the net force exerted on the wheels, i.e., $F_{engine} - F_{braking}$. Parameters f and k_L, respectively, are the rolling friction coefficient and the aerodynamics lift parameter. g =9.81 [m/s^2] is the acceleration of gravity. Table 1 completes the terminology.

Substituting the results in (25) through (28) into (21) through (23) yield the following form.

$$m\ddot{x} = \left(F - f\frac{b}{a+b}\left(mg - k_L\dot{x}^2\right)\right)\cos\delta + C_f\left(\frac{\dot{y}+a\dot{\varepsilon}}{\dot{x}} - \delta\right)\sin\delta + m\dot{y}\dot{\varepsilon} - f\frac{a}{a+b}\left(mg - k_L\dot{x}^2\right) - k_D\dot{x}^2 \tag{29}$$

$$m\ddot{y} = \left(F - f\frac{b}{a+b}\left(mg - k_L\dot{x}^2\right)\right)\sin\delta - C_f\left(\frac{\dot{y}+a\dot{\varepsilon}}{\dot{x}} - \delta\right)\cos\delta - m\dot{x}\dot{\varepsilon} - C_r\left(\frac{\dot{y}-b\dot{\varepsilon}}{\dot{x}}\right) \tag{30}$$

$$\frac{I_z}{a}\ddot{\varepsilon} = \left(F - f\frac{b}{a+b}\left(mg - k_L\dot{x}^2\right)\right)\sin\delta - C_f\left(\frac{\dot{y}+a\dot{\varepsilon}}{\dot{x}} - \delta\right)\cos\delta + \frac{b}{a}C_r\left(\frac{\dot{y}-b\dot{\varepsilon}}{\dot{x}}\right) \tag{31}$$

The linear velocities, $\dot{x}$ and $\dot{y}$, in (29) through (30) are the instantaneous velocities in the directions of the vehicle's longitudinal axis and the lateral axis, respectively. The following transformations, i.e., (32) and (33) are used to refer them to the fixed world coordinate frame, $\{W\}$:

$$\dot{x} = \dot{x}_w\cos\varepsilon + \dot{y}_w\sin\varepsilon \tag{32}$$

$$\dot{y} = \dot{y}_w\cos\varepsilon - \dot{x}_w\sin\varepsilon \tag{33}$$

Integrating the dynamic equations in its new form will yield vehicle's coordinate positions, $[x_w, y_w]$, in the world coordinate frame.

5.2 Simulation Setup

The simulation studies were performed in Simulink/Matlab environment. The fixed step length was taken as 0.01[s] and the external information through IVC was obtained with an added delay of 0.05 [s] in order to simulate the delays caused by information processing, transmission etc., (It was assumed that the GPS delays could be approximately compensated during each feed from the GPS sensor by a Dead Reckoning (DR) algorithm using the data from the digital compass and the speedometer). In the simulation, the ordinary differential equations type-4 (Runge-Kutta) was used. The embedded hardware (for simulation and code generation) was selected as 'unspecified (32 bit generic)'.

The model parameters are given in the Table 1 [23]. The vehicles were assumed to be identical. The maximum deceleration for both the vehicles was taken as 4 [m/s^2]. The steering angle was limited to $[-\pi/6\ ,+\pi/6]$[rad]. r is taken to be 5 [m], which is the radius of the smallest circle that can be drawn around the dimensions of twice the size of the vehicle (The Collision Condition Function provides more details). The dimensions of the vehicle are 4.45 [m] in length and 1.72 [m] in width. In simulation studies, $k_s = 3$[m].

5.3 Matlab/Simulink System Blocks for Simulation Setup

The following figures give the system block diagrams those were used to perform the simulation in the Matlab/Simulink environment.

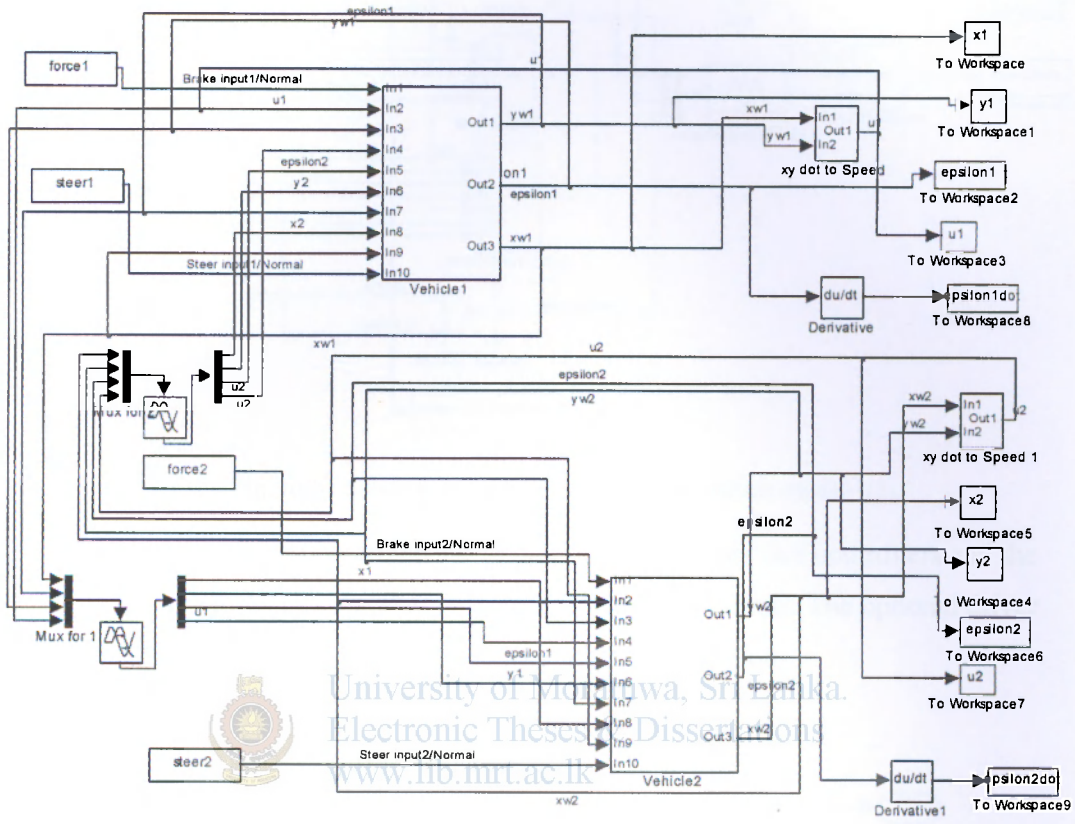


Fig.10(a) Main Simulink block model for the simulation

Fig. 10(a) gives the main system block for the simulation in the Simulink/Matlab environment. All parameter feeding and outputs are designated clearly here. Information passing between the two vehicular blocks are added with a signal delay in order to allow for the IVC communication delay.

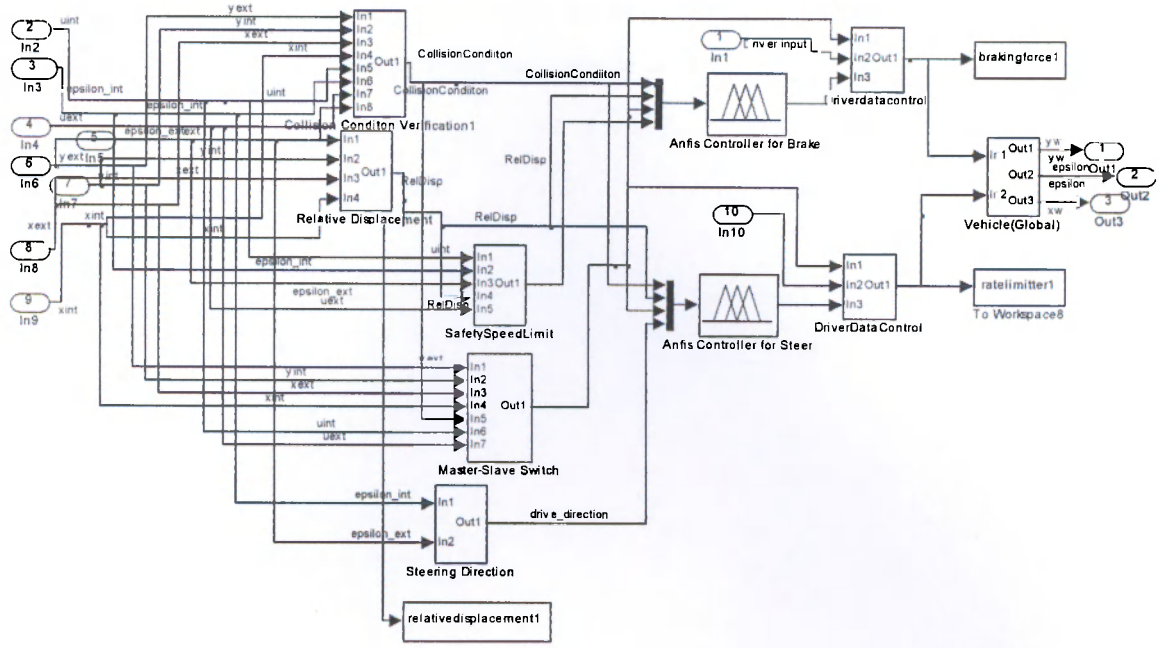


Fig. 10(b) Sub-system controller block for the vehicle model-1(2)

Fig. 10(b) gives the controller block which identifies the two controllers and the auxiliary functional blocks together with other inputs and outputs. The optional driver control blocks are fed whenever the controller is not in operation.

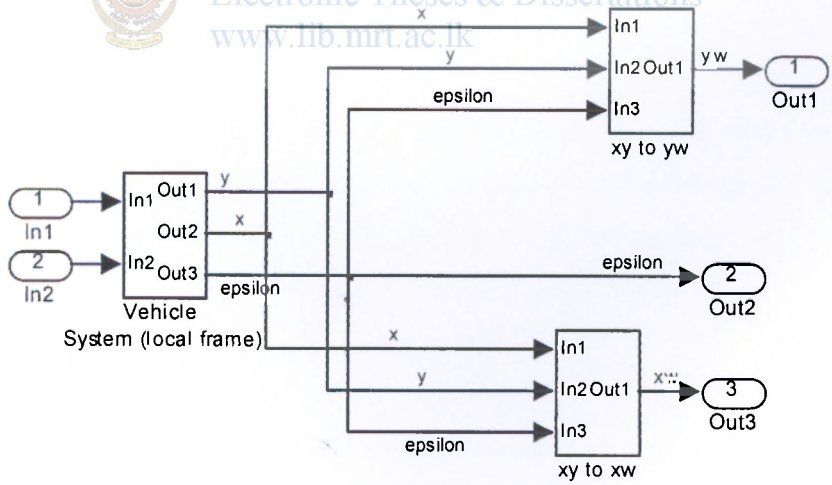


Fig. 10(c) Main vehicle model with the world-coordinate transformation blocks

Fig. 10(c) gives the model equations for the vehicle. The supporting blocks are to transform the model equations from the vehicle coordinate frame to the world coordinate frame.

5.4 Parameters for each Vehicle in Simulation

Table 1 gives the parameters for the vehicle mode used in the simulation studies. More details can be found under Vehicle Dynamics Model.

TABLE 1
TYPICAL PARAMETERS (NOMINAL) OF EACH VEHICLE SYSTEM

	Description	Value
m	Total mass of the vehicle	1640 kg
I_z	Yaw inertia	3105 kgm^2
f	Rolling friction coefficient	0.02
D	Wheel base $(a+b)$	2.78 m
a	Distance - front axle to c.o.g.	1.193 m
C_f	Cornering stiffness - front	131391 N / rad
C_r	Cornering stiffness - rear	115669 N / rad
k_l	Aerodynamics - lift parameter	0.008 Ns^2m^{-2}
k_D	Aerodynamics - drag parameter	0.49 Ns^2m^{-2}

5.5 Simulation Results

The speed range of the vehicles was taken to be 15-30 [m/s] at the verge of a collision. Following are the different vehicle positions for collision evasion scenarios. The two vehicles can be considered to have not collided as long as the relative distance is not less than 5 [m].

5.5.1 Side-End Collisions

Fig. 11(a) gives an account of an evasion of a dangerous scenario, which otherwise would definitely have led to a collision if had not been equipped with the collision avoidance controller. The crossing point of the trajectories is circled and the time each vehicle passed this point is given by t_1 and t_2 for vehicle-1 and vehicle-2, respectively. The relative distance has not reached 5 [m] value as shown in Fig. 11(b) (solid line) clearly indicating that the two vehicles have successfully avoided a probable collision. The circled point shows the instant the controller is switched on.

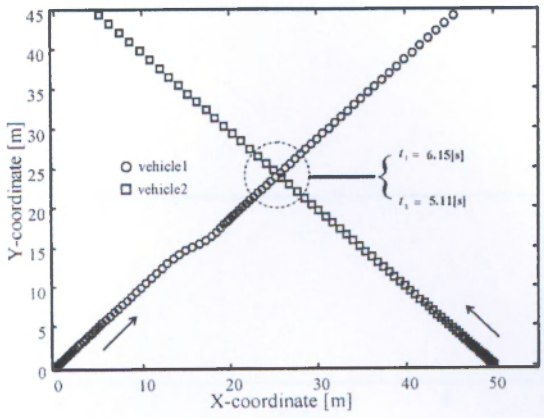


Fig. 11(a) Trajectories of the vehicles in the near side-end collision scenario and consequent evasion

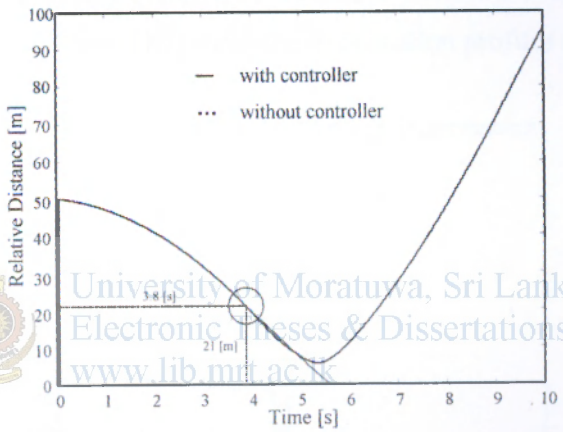


Fig. 11(b) Change of relative distance between the two vehicles (from the perspective of vehicle-1). Minimum relative distance is at 5.8 [m] (for controlled state).

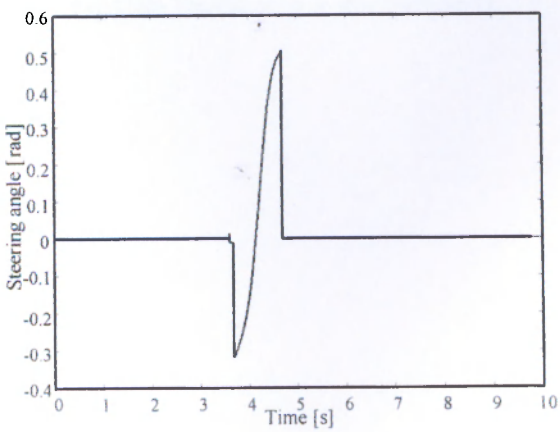


Fig. 11(c) Steering command (vehicle-1).

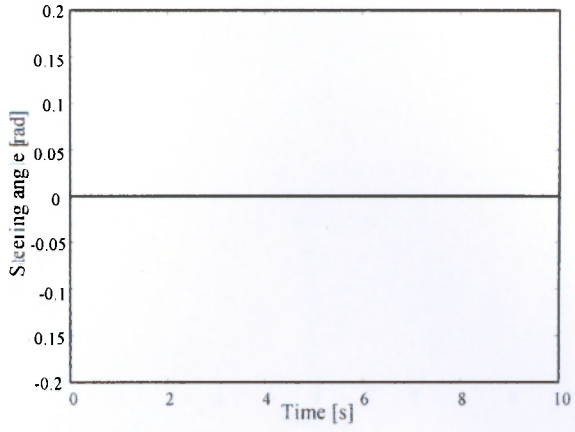


Fig. 11(d) Steering command (vehicle-2).

As given in Fig. 11(c)-11(d), only vehicle-1 performs steering maneuvers to avoid the collision. Fig. 11(e)-11(f) show the deceleration profiles of the two vehicles.

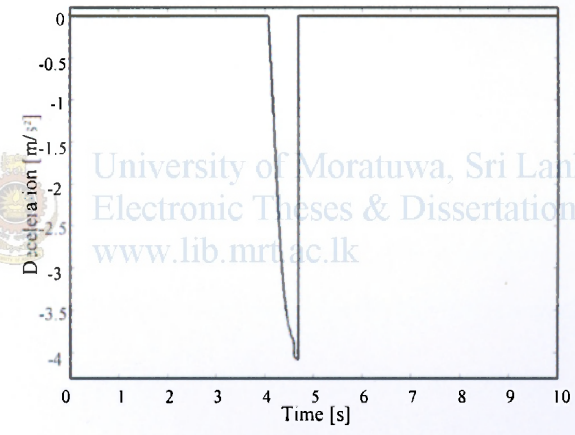


Fig.11(e) Deceleration profile (vehicle-1).

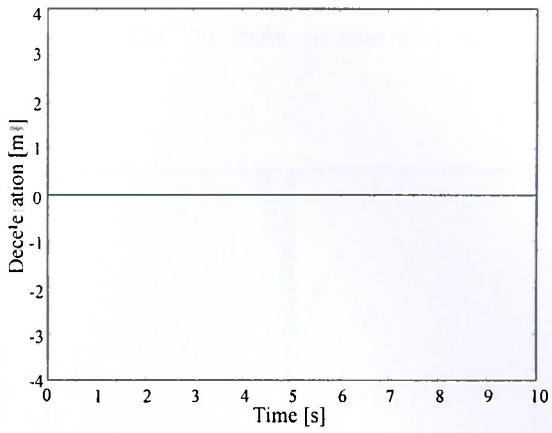


Fig. 11(f) Deceleration profile (vehicle-2).

As given in the Fig. 11(e)-11(f), the braking action is only taken by vehicle-1 in order to avoid the probable collision. Therefore, vehicle-1 has taken the total responsibility of avoiding the collision, while vehicle-2 is not disturbed. This is further visualized in Fig. 11(a).

5.5.2 Rear-End Collisions

Fig. 12(a)-12(f) elaborates on how a rear-end collision can be avoided effectively.

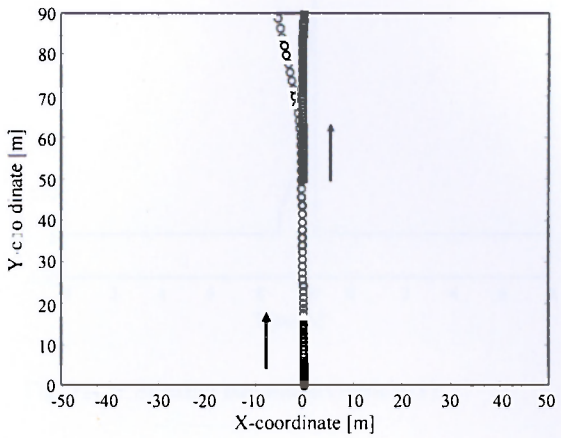


Fig. 12(a) Trajectories of the vehicles in the near rear-end collision scenario and consequent evasion.

Fig. 12(a) shows a case of a near rear-end collision. Here, vehicle-1(with a maximum speed of 30 [m/s]) is fast approaching vehicle-2 (maximum speed of 20 [m/s]).

Fig. 12(b) shows change of relative distance for the case as given in Fig. 12(a). Proper collision avoidance has been ensured by the fact that $\min(R_D) > 5\,[\mathrm{m}]$. The controller was switched on at the moment given by the circled point.

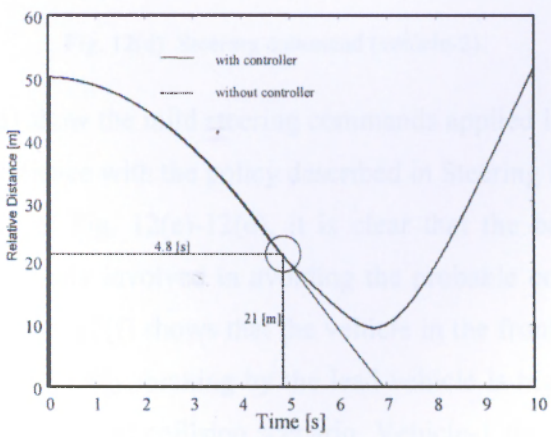


Fig. 12(b) Relative distance between the two vehicles (from the perspective of vehicle-1).

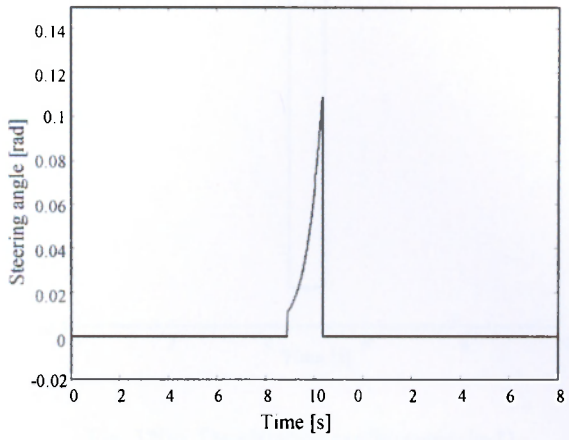


Fig. 12(c) Steering command (vehicle-1).

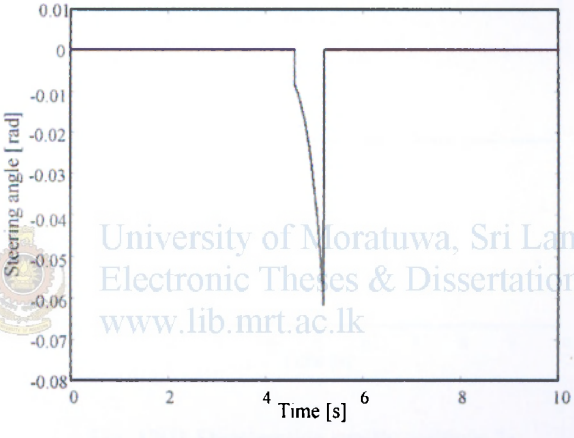


Fig. 12(d) Steering command (vehicle-2).

Fig. 12(c)-12(d) show the mild steering commands applied in this case. Steering directions are in compliance with the policy described in Steering Direction Function.

From the looks of Fig. 12(e)-12(d), it is clear that the braking controller of vehicle-1 was only actively involved in avoiding the probable collision. This is also evident in Fig. 12(a). Fig. 12(f) shows that the vehicle in the front (vehicle-2) has not put on brakes. Understandably, braking by the lead vehicle is highly unacceptable in the face of a probable rear-end collision scenario. Vehicle-1 that is behind vehicle-2 has largely depended on braking even though it also had to steer due to its high moving speed demonstrating a very natural collision avoidance performance.

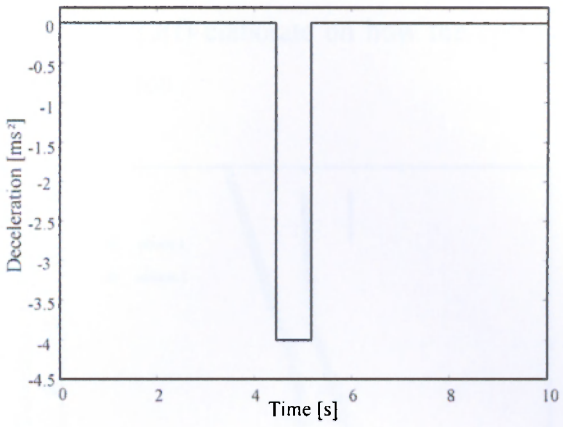


Fig. 12(e) Deceleration profile (vehicle-1).

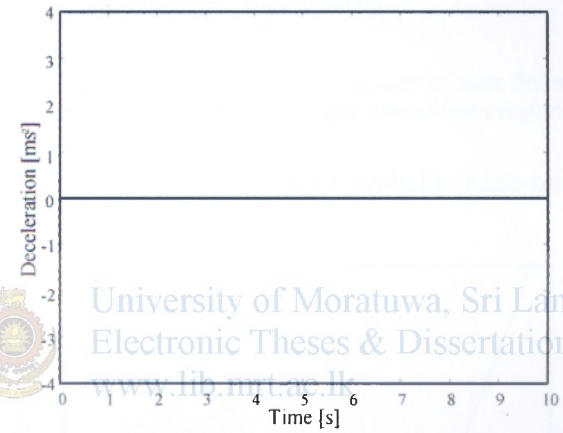


Fig. 12(f) Deceleration profile (vehicle-2).

5.5.3 Head-On Collisions

Fig. 13(a) through Fig. 13(f) elaborate on how the controller can successfully avoid a possible head-on collision.

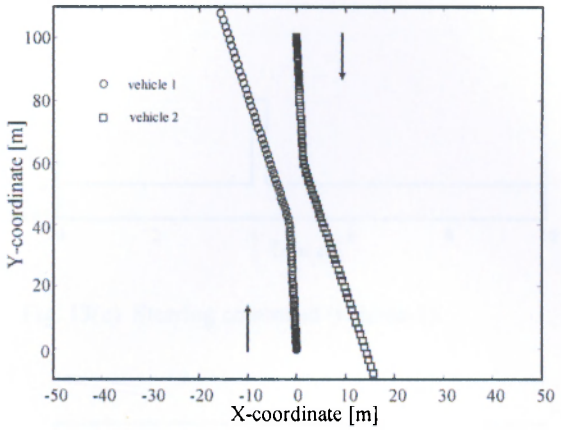


Fig. 13(a) Trajectories of vehicles in case the near head-on collision scenario and consequent evasion.

Fig. 13(a) gives the scenario of avoiding a probable head-on collision.

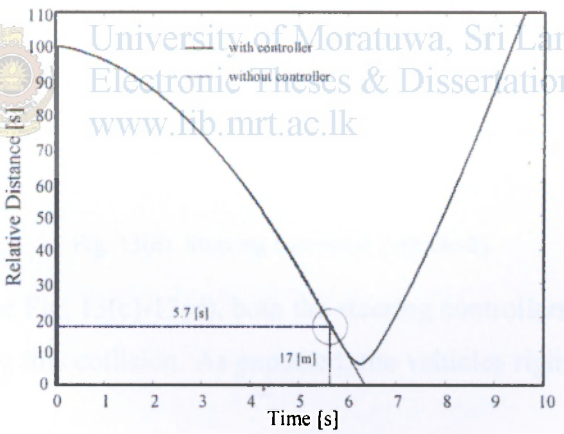


Fig. 13(b) Relative distance between the two vehicles (from the perspective of vehicle-1). Minimum relative distance is 6.6 [m].

Relative distance profile given in the Fig. 13(b) depicts successful collision avoidance with $\min(R_D) = 6.6$ [m] . The circled point shows the instant of controller switch on.

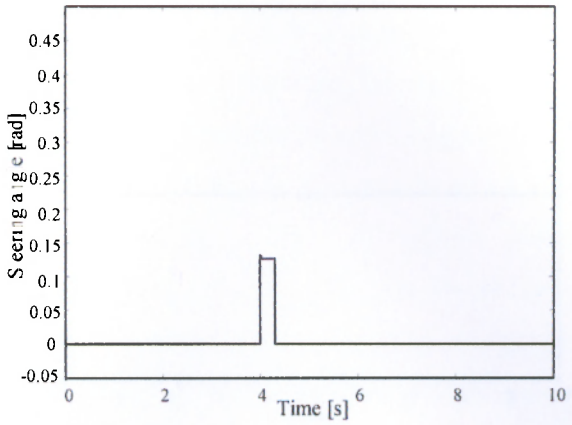


Fig. 13(c) Steering command (vehicle-1).

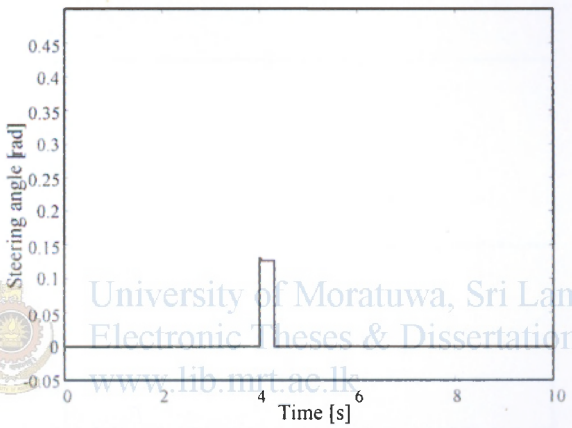


Fig. 13(d) Steering command (vehicle-2).

As given in the Fig. 13(c)-13(d), both the steering controllers have been actively involved in avoiding this collision. As expected, the vehicles rightly steer away from each other.

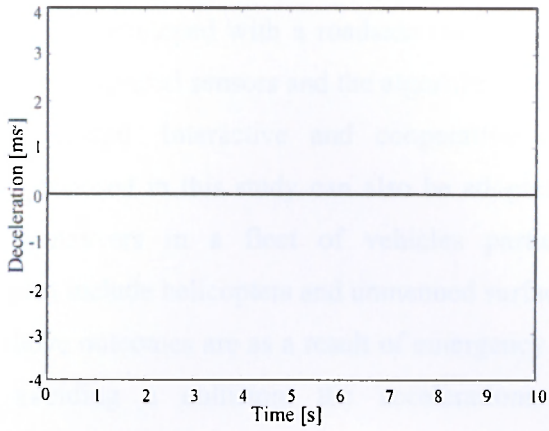


Fig. 13(e) Deceleration profile (vehicle-1).

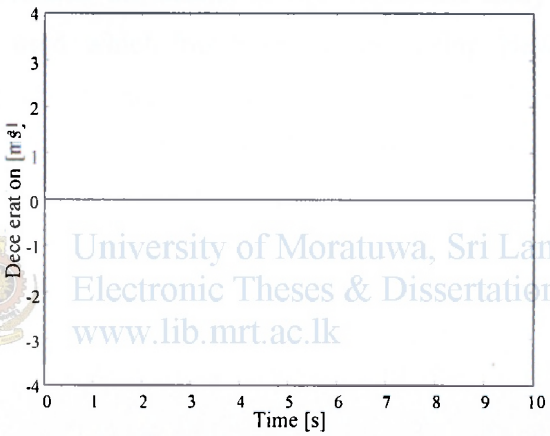


Fig.13(f) Deceleration profile (vehicle-2).

Fig. 13(e)-13(f) give the braking controller outputs. In this case the braking controllers have not played an active role in avoiding the collision.

Remark 5.1: In the simulations, it has been assumed that an emergency steering of a vehicle to avoid a collision does not build up another collision scenario or the vehicle does not get out of the road. But in practice, this may not be the case. This problem may be solved by introducing a simple logic controller that will stop the vehicle immediately after a particular collision is avoided in case it had to steer to avoid that collision. In case if a second collision scenario has already built up, the vehicle will naturally perform another collision avoidance maneuver.

***Remark 5.2*:** In case the other vehicle does not have the IVC facility or in case the collision scenario has been developed with a roadside obstacle, collision avoidance will totally depend on the peripheral sensors and the algorithm will work like a typical collision avoidance algorithm. Interactive and cooperative collision avoidance approach like the one proposed in this study can also be adopted to ensure optimal collision avoidance maneuvers in a fleet of vehicles particularly in military applications that may also include helicopters and unmanned surface vehicles (USVs).

***Remark 5.3*:** As the above outcomes are as a result of emergency intervention due to sudden actions of avoiding a collision, the accelerations that incur being comparatively high can be allowed.

5.6 Summary

This chapter contains the details of the simulation study and the results. A vehicle model was used which has been derived using Newton-Euler dynamic equations. The results of the simulation were discussed under three main categories of vehicle collisions, i.e., head-on collisions, side-end collisions and rear-end collisions.

CHAPTER 6

Prototype Realization

6.1 Component Selection

The following components have been selected for realization of two prototypes intended for studying intelligent interactive collision avoidance studies in smart vehicular systems.

6.1.1 Controller Board – Oopic R+

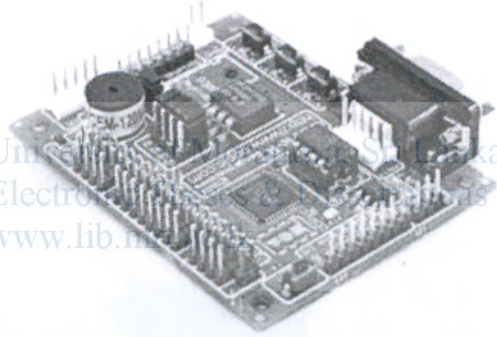


Fig. 14 Oopic R+ Controller board

Oopic R+, by Savage Innovations, Inc., is a new kind of controller board which is programmable in the object oriented style. This enables to rapidly develop or change the programs that interact with the hardware components attached to the Oopic R+ controller board. For all testing, the 'Basic' style programming was used. In Oopic R+ language, all functionality and processing are confined to the specially defined objects. This reduces the number in the instruction set making it easy to use for applications. These objects are specifically addressing some of the known functions of the hardware components, e.g. oPWM object generates a modulated

pulse as specified by the other properties of the object. In this development, the Oopic R+ firmware used is of the version B2.2+. The microcontroller used is based on the Microchip's PIC16F877 microcontroller. The 31 number of input lines can be connected to the given hardware options. There are 5 steps involved in creation of an application in Oopic R+:

1. Starting the Oopic Software development kit
2. Designing a hardware interface
3. Writing code
4. Connecting the hardware
5. Downloading and running the application

The I^2C connectivity, a versatile IC connection bus, is facilitated by the Oopic R+ board enabling effective networking with some sensors and hardware components which are featured with the same facility. This adds to easy interface made already effective by the relevant object types [15].

6.1.2 Servo Motors

Fig.15 Hitec HS-422 standard deluxe Servo Motors

The HS-422 standard deluxe servomotors, by Hitec RCD Inc., are 3-pole ferrite type attached with an in-built potentiometer. These motors are with the control system of pulse width 1500 μ s neutral type.

The operating voltage range is 4.8v to 6.0v. The speed of operation varies from 0.21sec/60° at no load to 0.16sec/60° at no load. The stall torque lies between

3.3kg.cm (45.82oz.in) to 4.1kg.cm (56.93oz.in) with the two voltage extremes, respectively [5].

6.1.3 Radio Frequency Modules (Transmitter/ Receiver)

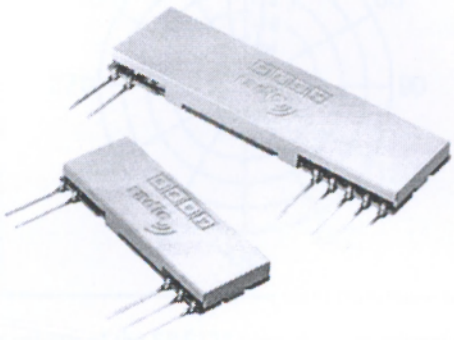


Fig. 16 RF communication Modules: Receiver and Transmitter

The EasyRadio (ER) ER400TS Transmitter, ER400RS Receiver, by LPRS Ltd., were used to give a high performance communication link between the central PC and the prototypes. These RF devices can transfer data over a range of up to 250 metres Line Of Sight (LOS), even though this much of range is not required in this application. For this kind of application, these devices can be optimized with frequency, data rate and power output. The embedded software reduces design and development time significantly, together with the vendor-supplied programming and testing platform [4].

6.1.4 Ultrasonic Sensors

These ultrasonic sensors, by Devantech Ltd., are for the purpose of clearly identifying obstacles around each vehicle prototype, while moving. The ultrasonic sensors were not meant to identify the other vehicle prototypes in close proximity (the adjacent prototypes are meant to be identified, by each other, with the RF communication between them).

With the selection of this type of ultrasonic sensor with a very low beam pattern, as indicated in Fig. 16, gives the opportunity to detect the obstacles accurately around it against detecting the floor as an obstacle with a false alarm. Mounting arrangements of these sensors were also considered with special concern [3].

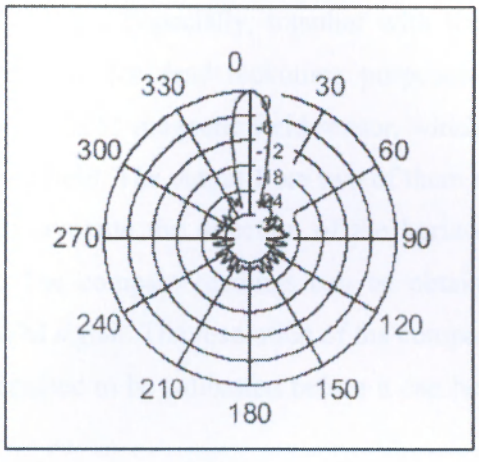


Fig.17 Beam pattern of the SRF235 'pencil beam' ultrasonic sensor

The ultrasonic sensor is with a single transducer for both transmit and receive. Therefore, there is a blanking zone out to 10cm, so the effective range is 10cm to 1.2m. Communication with the SRF235 ultrasonic rangefinder is via the I^2C bus. Therefore, this is easily connected to the Oopic R+ with its capability of I^2C. In order to connect these sensors to the Oopic R+, the address of the sensors have to be changed. The Appendix G gives more details.

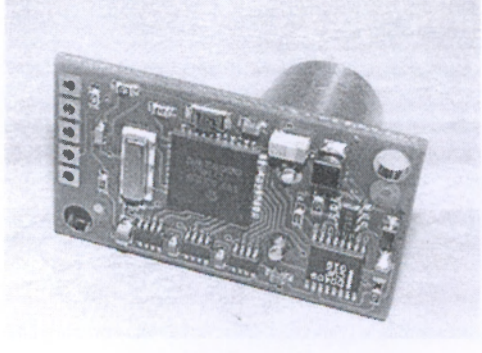


Fig 18 SRF235 'pencil beam' ultrasonic sensor

6.1.5 Digital Compass

This digital compass module, by Devantech Ltd., simply can be used as an aid to navigation in the prototypes. Especially, together with the optical encoders, this compass is meant to be used for dead reckoning purposes of the prototype. The compass uses the Philips KMZ51 magnetic field sensor, which is sensitive enough to detect the Earths magnetic field. The output from two of them mounted at right angles to each other is used to compute the direction of the horizontal component of the Earth's magnetic field. The compass readings can be obtained both from the I^2C channels as well as a PWM signal. The resolution of the compass is 0.1 degrees.

The compass is required to be calibrated before it can be used in the area being used [3].

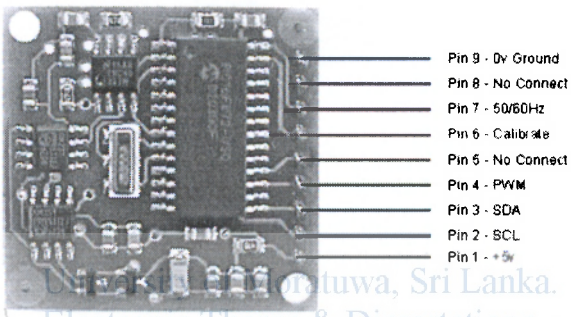


Fig. 19 Digital Compass and the external pin connections

6.1.6 Optical Encoder Modules

These optical encoder modules, by Agilent Technologies, Inc., are with three channel incremental encoders with a code wheel. The speed of the two wheels of the prototype is intended to measure using this. Optical encoder outputs together with that of the digital compass can be used for dead reckoning in navigation purposes of the prototypes.

Fig. 20 Optical Encoder and code wheel

6.2 Testing of Individual Components for Realization of Prototypes

6.2.1 RF Module Testing

The following circuit was realized for connecting to a central PC for communicating with the onboard RF modules on the two prototypes. The circuit is with the Max232 communication chip. With the 12V range operating with the RS232, the RF modules (with operating range of 5 V) cannot be directly connected without the circuit given in Fig. 20 [4].

As shown in the Fig. 20, the Transmitter/Receiver programming switches are selected i.e., either 1 or 2, in accordance with the requirement of programming of Receiver or Transmitter, respectively, while the RS232 socket is being connected to the central PC.

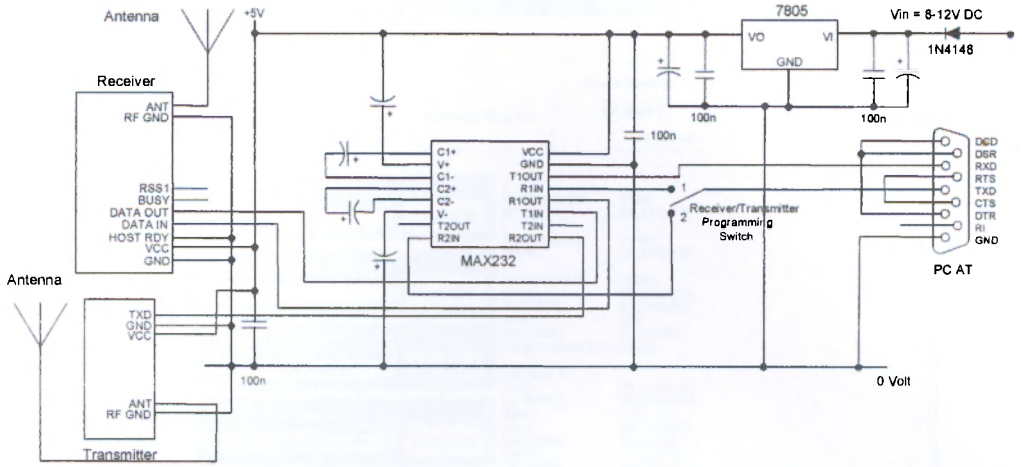


Fig. 21 Circuit diagram for RF communication between onboard RF modules

The central PC is installed with the 'Easy Radio Evaluation software v2.07'; a vendor supplied programming environment in PC platform that is required for programming the RF modules. The most important steps of programming the RF modules are given below.

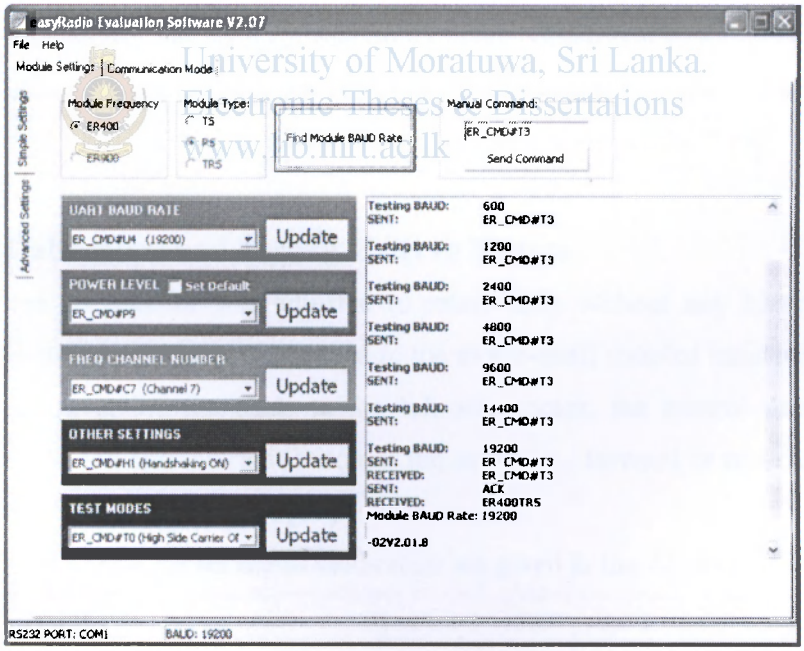


Fig. 22 Checking for existing baud rate of the receiver

The Fig.22 shows how the existing baud rate of the receiver is checked. It has been verified that it is currently at 19200 bps.

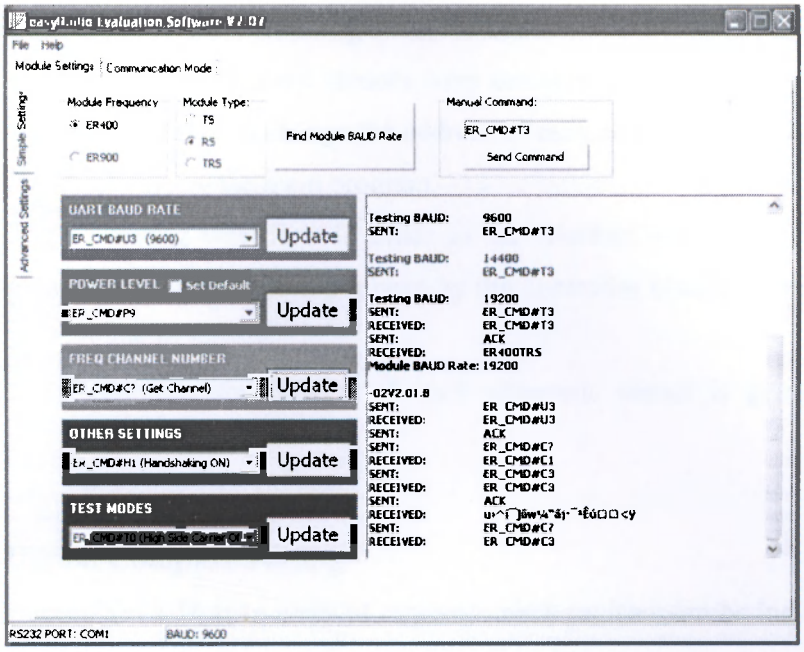


Fig. 23 After adjusting the baud rate to 9600 bps and changing the channel to #3 for the receiver

Fig 23 shows, how a particular baud rate is adjusted for the Receiver. It is shown that a baud rate of 9600 bps was adjusted in the Receiver. It also shows that the channel of the Receiver has been adjusted to #3 in the consequent step.

The main controlling program for testing communication with the prototypes that is supposed to be residing in the central PC is given in Appendix E [4].

6.2.2 Calibration and Testing of Servo Motors

First, the servomotor was adjusted to rotate fully without any hindrance by removing the mechanical stopper attached to the motor-shaft coupled main nylon gear wheel [5]. By reversing the leads to the left side motor, the control relationship between both the left and the right becomes the same, i.e., forward or reverse motion looks the same for both the motors [15].

The programs in Oopic R+ for motor calibration are given in the Appendix F.

6.2.3 Ultrasonic Sensor Testing

Initially, each of the ultrasonic sensors were connected one by one to the I^2C/ programming socket, in order to change the address of each sensor so that they could be identified individually, by the main program.

A qualitative testing was done in order to see whether the particular sensor identifies a given point with producing a beep by the controller board, exactly at the right point [3].

The program for changing the address of each ultrasonic sensor is given in the Appendix G.

6.2.4 Digital Compass Testing

The compass has a 16 byte array of registers. Each register can be individually accessed. Register 15 is used to calibrate the compass. The compass was calibrated before it was used [3].

The testing program for qualitative detection of a given bearing is given in the Appendix H.

6.2.5 Optical Encoder Testing

The HEDS-9040 is a three channel optical incremental encoder module, i.e., two channel quadrature output with a single index output with a 90 electrical degree high true pulse. Using together with the code wheel, this can detect the rotary motion and translate that into three channel digital output. The resolution of this encoder is 2000 CPR (Counts per Revolution). No signal adjustment is required for the encoder.

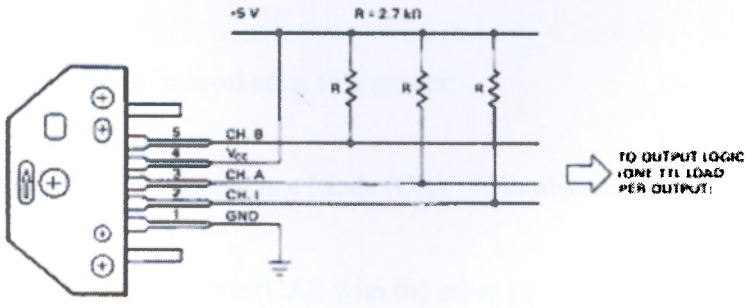


Fig. 24 Pull-up resistors on HEDS-9040 encoder module outputs.

Fig. 23 gives the pin arrangement and the pull-resistors required to connect them to the Oopic R+.

This encoder is TTL-compatible with a single 5 V supply. When the code wheel rotates in the direction of the arrow on top of the module, channel A will lead channel B. If the code wheel rotates in the opposite direction, channel B will lead channel A. Thus the reverse motion can be identified separately [16]. A qualitative testing program for the optical module is given in the Appendix I.

6.3 Development of an Algorithm for Collision Avoidance Studies with Prototypes

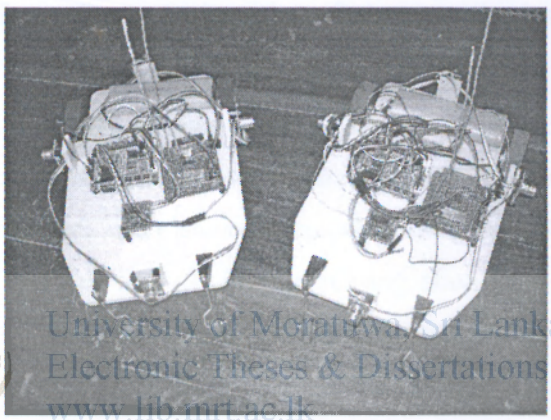


Fig. 25 Prototype platforms with assembled components

The Fig. 25 shows the assembled platforms for the two prototypes.

The following algorithm is developed for enabling the study of collision avoidance emergency intervention maneuvers for the two prototypes.

The prototype is planned to be worked in two modes:

1) Peripheral Obstacle Avoidance Mode (OA) ; with obstacles

2) Collision Avoidance Mode (CA); with the other prototype

The Obstacle Avoidance Mode is supported by four numbers of 'pencil beam' type ultrasonic sensors fixed onboard the prototype at the sides. This mode is activated whenever the prototype is in collision with an external obstacle when it detects it within 15 [cm] distance.

The Collision Avoidance Mode is supported by the RF modules and through it the positional and the heading angle information of the prototype is exchanged with the other prototype. The Collision Mode activates when the two prototypes are in line of collision when the distance between the two prototypes becomes less than or equal to 30 [cm].

6.3.1 Algorithm for the Prototype

The essential facts for an algorithm for the vehicles prototype are explained below.

6.3.1.1 Peripheral Obstacle Avoidance (OA) Module

OA mode activates when distance to the obstacle is 15 [cm] or less

(i) **If** Obstacle on Left **and** distance decreasing, **then** Turn Right for 3 [s]

(ii) **If** Obstacle on Right **and** distance decreasing, **then** Turn Left for 3 [s]

(iii) **If** Obstacle on Front & Left **and** distance decreasing, **then** Turn Right for 3 [s]

(iv) **If** Obstacle on Front & Right **and** distance decreasing, **then** Turn Left for 3 [s]

(v) **If** Obstacle on Front **then** Stop. Reverse for 2 [s]. Stop and Turn Right for 3 [s]

(vi) **If** Obstacle on Back **then** Stop (if Reversing) and Turn Right for 3 [s]

A Dead Reckoning (DR) algorithm to be used in the prototype is given below. Using the speed of the prototype, obtained from the two optical encoders (two speeds are averaged) and the heading angle, obtained from the digital compass, the following x and y positions are calculated.

$$\text{Prototype Position } (x) = \text{Last } (x) \text{ Position} + \text{Prototype's Speed} * \cos(\text{Heading Angle}) * \text{Time Step} \quad (34)$$

Prototype Position (y) = Last (y) Position + Prototype's Speed*
sin (Heading Angle)*Time Step (35)

Initially, the time step is taken as 100 [ms]. After performing a few trials, an approximate time step can be verified for the studies.

6.3.1.2 Collision Avoidance (CA) Module

CA mode applies when the distance to the other prototype is 30 [cm] or less. The description of facts for the CA module goes follows:

When the prototype detects that it is in collision situation, it stops for 3 [s]. Then it takes a 'right turn' for 3 [s]. If it still cannot move, it will turn to the 'left' for 3 [s]. As the final solution, reversing will follow for the same period of 3 [s].

According to the distance values taken, OA mode overrides the CA mode. Therefore, if the CA mode fails, the OA mode can take up and avoid any possible collision either with an obstacle or with the other prototype.

Remark 6.1: An algorithm for sensor fusion based on fuzzy logic is preferred for fusing the ultrasonic sensors.

Remark 6.2: Positions and heading angles are exchanged between the prototypes through the RF communication link via a central PC.

Remark 6.3: The speed of the prototype is assumed approximately constant for the maneuvers. Thereby, the relative velocity is approximately calculated by each prototype. Here, a 30 [deg] collision cone is always assumed for the prototypes.

Remark 6.4: Turning of the prototype is done by lowering the speed of the servo motor of the same side to which the prototype is required to be turned. By pulse width adjustments in software, approximately a 0.5 [m] radius fixed turning curve can be defined for the general turning events of the prototype.

Remark 6.5: The default turning side is to the 'right'. In all above cases, if it is not possible to turn to the right, the 'left' turn is selected. If still not possible to recover, the prototype is reversed (for 2 [s]) in order to escape from the deadlock.

Remark 6.6: The prototype detects that the other prototype is in collision terms when it is in the collision cone, while the distance between them is 30 [cm] or below.

CHAPTER 7

Conclusion and Future Directions

7.1 Conclusion

This work has presented and demonstrated the capabilities of the performance of an interactive controller based on intelligent collision avoidance. The interactive controller realizes suitable cooperative maneuvers for evading near collisions among two vehicles. In other words, some of the critical characteristics that the first vehicular system adopts in the face of a probable collision situation are supposed to be mutually agreed upon by the second system. It is obviously true, as also reflected from this study, that two or more systems trying to avoid a probable collision is more successful acting through mutual 'negotiation' rather than acting alone. Here, the collective participation yields rapid and effective solutions while bringing the synergetic effects for a better safety. To identify the hierarchy of the two vehicles, a master/slave concept was used.

In this study, an ANFIS-based control has been successfully used to synthesize the collision avoidance controllers for vehicles. ANFIS is considered to be one of the most effective neuro-fuzzy formulations to be utilized as a controller. The adaptive nature of the ANFIS allows quickly pick-up and learn efficiently. Two separate controllers, one for braking and the other for steering, have been used for each vehicle. Apart from the controllers, few auxiliary functional entities were formulated as inputs to the controllers. These functions were meant to properly identify and quantify the complex scenarios involved in the collision situation.

The simulation study proves the effectiveness of the interactive control method for intelligent collision avoidance with good results for the two vehicular systems. The results show that the controllers, braking and steering, act in different combinations to best suit the prevailing collision scenario to realize collision prevention actions that are very natural in that they resemble the actions that would be taken by an expert and alert human driver.

Even though, this study has been done for two vehicular systems, these results can be generalized even for multiple vehicles without a significant conceptual breakthrough. It may have to define different sizes for the secondary and tertiary collisions in relation to the primary collision. Additionally, an adaptive path generation for the vehicles to follow for evading collisions would be beneficial.

Preliminary testing of the components was successfully concluded for realization of prototypes with fully autonomous interactive capability for collision avoidance studies. The developed algorithms for the operational modes of the prototypes can be used to study the interactive control of intelligent collision avoidance of autonomous vehicular systems.

7.2 Suggestions for Future Directions

In this study, the controller was not tested onboard a real vehicle in a nearly real collision scenario. In the future studies, the controller needs to be taken up with the real vehicles in order to study the effectiveness of the maneuvers it generates. It is also very important to do the experiments with most of the state-of-the-art equipment and technologies to be in par with the usage of vehicle manufacturers of today. The new technologies that might include consist of CAN (Controller Area Network) based equipment, modern MEMS applications like gyroscopes, accelerometers etc., and high precision DGPS sensors for navigation, etc.

As far as other alternative directions are concerned, an adaptive-predictive controller would give a more effective solution due to its inherent characteristics in finding a relevant solution for the problem discussed. Another positive direction would be to investigate on the biologically-inspired intelligent controllers. It is hoped, this will provide some promising new solutions if coupled with adaptive-predictive realms. The potential field method is also a possible direction to be reckoned with. It can be used to get a positive solution with the consideration of dynamic constraints.

All these new directions of research will enable to find a very good solution to the problem of vehicle collisions in the future.

References

[1] Abbott, E and Powell, D, "Land-vehicle Navigation using GPS," *Proc. of the IEEE*, vol.87, no.1, pp.145-162, 1999.

[2] Bhawani Selvaretnam, K. Daniel Wong, "Handling the inter-vehicular communications Challenge- A Survey," *Proceedings of the International Conference on Computational Science (ICCS)*, pp.86-90, 2004.

[3] Devantech Ltd., Unit 2A-2B, Gilray Road, Diss, Norfolk, IP22 4EU, UK

[4] LPRS Ltd., Two Rivers Industrial Estate, Station Lane, Witney, Oxon, OX28 4BH, UK

[5] Hitec RCD USA, Inc., 12115 Paine St, Poway, CA 92064, USA.

[6] I Chicasalita, N ShahMehri, "Active support for traffic safety applications through vehicular communications," *International Workshop on Trends in Information Systems,* Romania, 2002.

[7] Horowitz, R, Varaiya, P, "Control design of an automated highway system," *Proceedings of the IEEE*, vol.88, no.7, pp.913-925, 2000.

[8] J Ackermann, *et al.*, "Linear and Nonlinear Controller Design for Robust Automatic Steering," *IEEE Trans. Contr. Sys. Tech.*, vol.3, no.1, pp.132-143, 1995.

[9] J Hancock, "Laser intensity-based obstacle detection and tracking," doctoral dissertation, tech. Report CMU-RI-TR-99-01, Robotics Institute, Carnegie Mellon University, 1999.

[10] J H. Zhao, R. Shibasaki, "Reconstructing Urban 3D Model Using Vehicle-Borne Laser Range Scanners," *3dim, Third International Conference on 3-D Digital Imaging and Modeling (3DIM '01I)*, pp.349, 2001.

[11] J M Wang and R Rajamani, "Adaptive cruise control system design and its impact on highway traffic flow," *Proc. American Control Conf.,* Anchorage, AK, pp.3690-3695, 2002.

[12] J S R. Jang, C T Sun and E Mizutani, *Neuro-fuzzy and soft computing,* Delhi: Pearson Education (Singapore) Pte. Ltd., pp.50-117,121-154,366,362-363, 2005.

[13] Jun Luo and Jean-Pierre Hubaux. "A Survey of Inter-Vehicle Communication," Technical report IC/2004/24, School of computer and Communication Sciences, EPEL, 2004.

[14] Lars Wischhof, Andre Ebner and Hermann Rohling, "Information dissemination in self-organizing intervehicle networks," *IEEE Trans. Intelligent Transp. Syst.*, vol.6, no.1, pp.1–4, 2005.

[15] Savage Innovations, 906 Bob Wallace Avenue-Suite F, Huntsville, AL 35801, USA.

[16] Agilent Technologies, Inc., 5301 Stevens Creek Blvd, Santa Clara, CA 95051, USA.

[17] Paolo Fiorini and Zvi Shiller, "Motion planning in dynamic environments using the relative velocity paradigm," *Proc. of IEEE Intl. Conf. on Robotics and Automation,* vol.1, pp.560-565, 1993.

[18] P Varaiya and S E Shladover, "Sketch of an IVHS systems architecture," Institute of Transportation Studies, University of California, Berkeley, Tech. Rep. UCB-ITS-PRR-91-3, 1991.

[19] P Varaiya, "Smart cars on smart road: Problems of control," *IEEE Transactions on Automatic Control,* vol.38, no.2, pp.195-207, 1993.

[20] Ravipriya Ranatunga, Sisil Kumarawadu, Pawan Lingras and Tsu-Tian Lee, "A new paradigm for intelligent collision avoidance via interactive and interdependent generic maneuvers," *Proc. IEEE Intl Conf. on Systems, Man & Cybernetics, Taipei,* 2006, -to be published.

[21] Richard Bishop, "Intelligent Vehicle Applications Worldwide," *IEEE Intelligent Systems,* vol.5, no.1, pp.78-81, 2000.

[22] Seiler P, Song B and Hedrick J K, "Development of a collision Avoidance System," Society of Automotive Engineers, Inc., Tech. Paper 98PC-417, 1998.

[23] Sisil Kumarawadu and Tsu-Tian Lee, "Neuroadaptive combined lateral and longitudinal control of highway vehicles using RBF networks," *IEEE Trans. on Intelligent Transp. Syst,* vol.7, no.4, pp.500-512, 2006.

[24] Taipale T and Hirai S, "A behavior-based control system applied over multi-robot system," *Proceedings of the 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems, Yokohama, Japan*, pp.1941-1943, 1993.

[25] Tsoukalas L H and Uhrig R E, *Fuzzy and Neural Approaches in Engineering*, John Wiley & Sons, Inc., NewYork, pp.191-288, 1997.

[26] T W Vaneck, "Fuzzy guidance controller for an autonomous boat," *IEEE Control Systems,"* vol.17, no.1, pp.44-47, 1997.

[27] Wang C D, and Thompson J P, "Apparatus and method for motion detection and tracking of objects in a region for collision avoidance utilizing a real-time adaptive probabilistic neural network," US Patent No.5,613,039, 1997.

[28] Wang J *et al.*, "Lane keeping based on location technology," *IEEE Transactions on Intelligent Transportation Systems,* vol.6, no.3, pp.351-356, 2005.

[29] W. Kleinhempel, "Automobile Doppler Speedometer," *Proc. IEE Vehicle Navigation & Information Systems Conf. Ottawa*, pp.509-512, 1993.

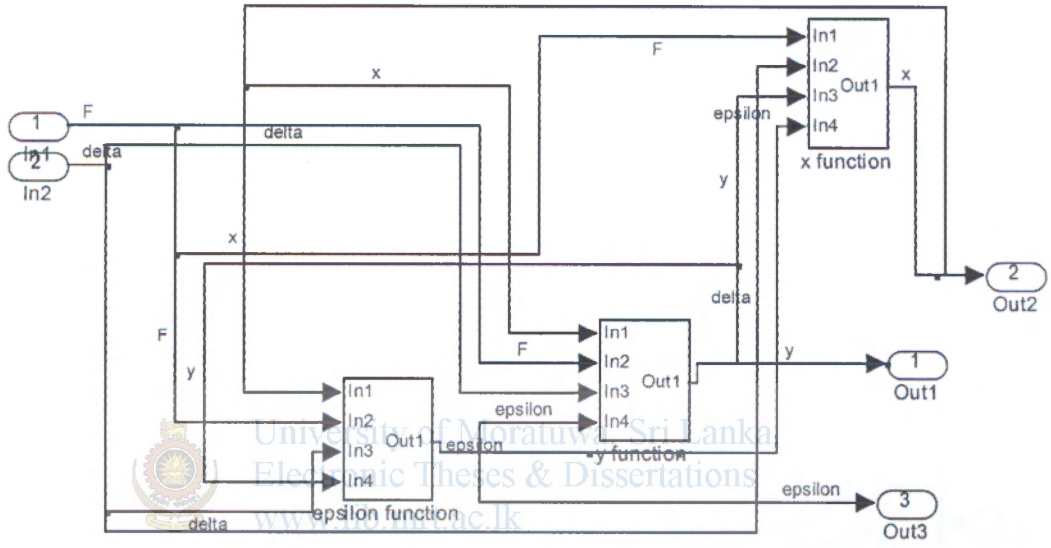
[30] Wu Zhang, "Cooperatively controlled collision avoidance," *Proceedings of the IEEE Conference on Intelligent Transportation Systems*, pp.824-829, 1997.

[31] A Bejan, R Lawrence, "Peer-to-peer cooperative driving," Proceedings of ISCIS 2002: International Symposium on Computer and Information Sciences, USA,2002, pp.259-264.

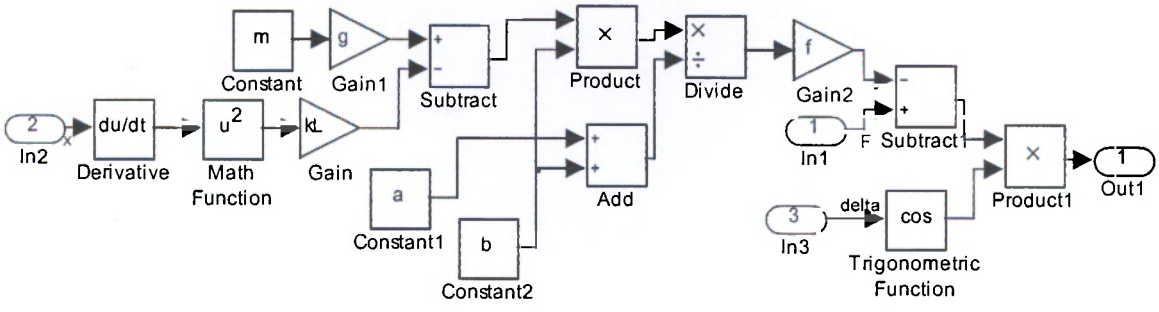
Appendix A

Overview of Simulation Sub-System Blocks

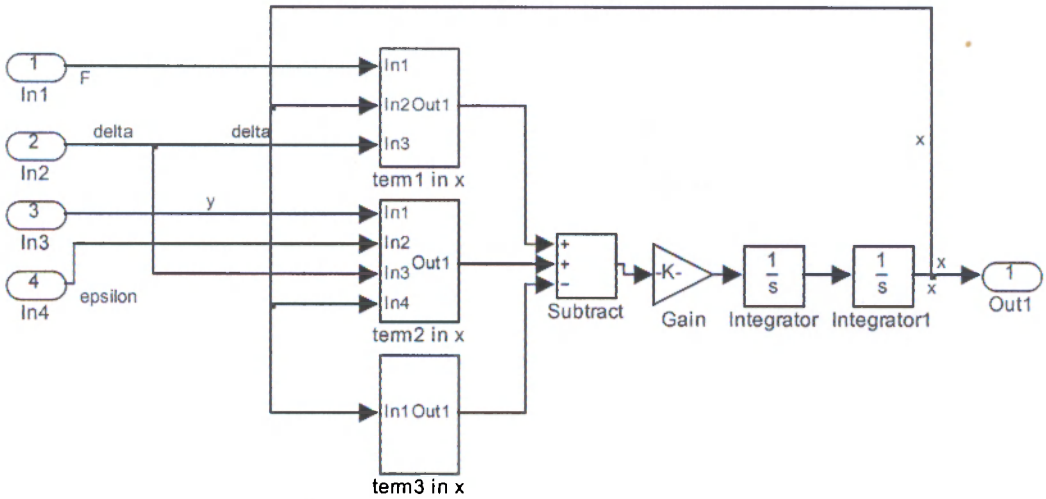
Sub-system blocks for main functional equations of vehicle model



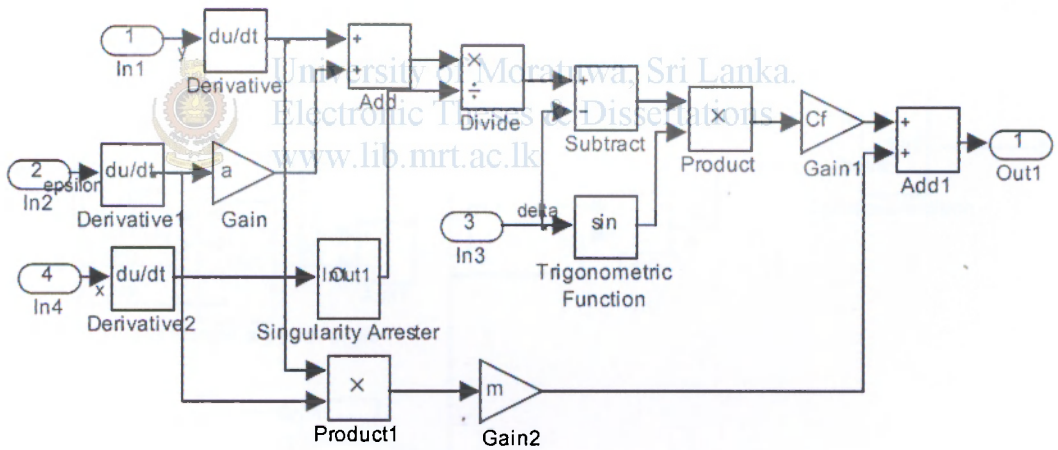
Sub-system blocks for the terms of the equation (18)



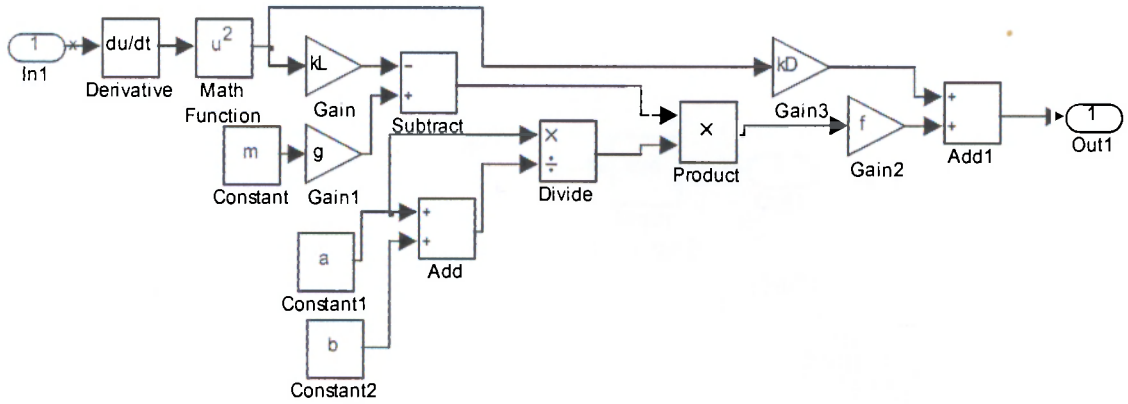
Sub-system blocks arrangement for the equation (18)



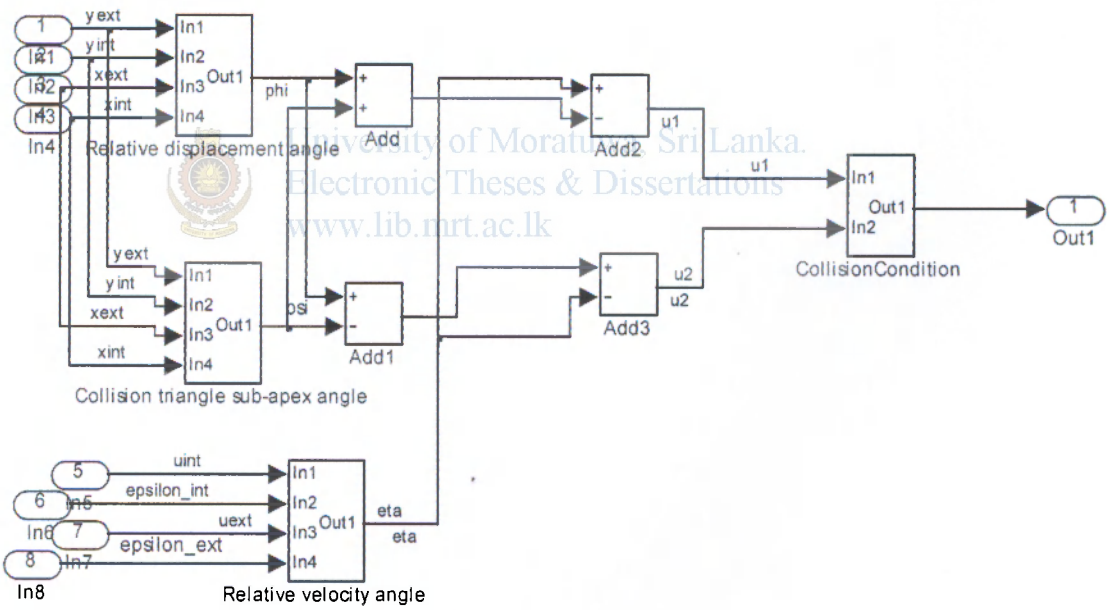
Sub-system blocks for the first term of the equation (18).



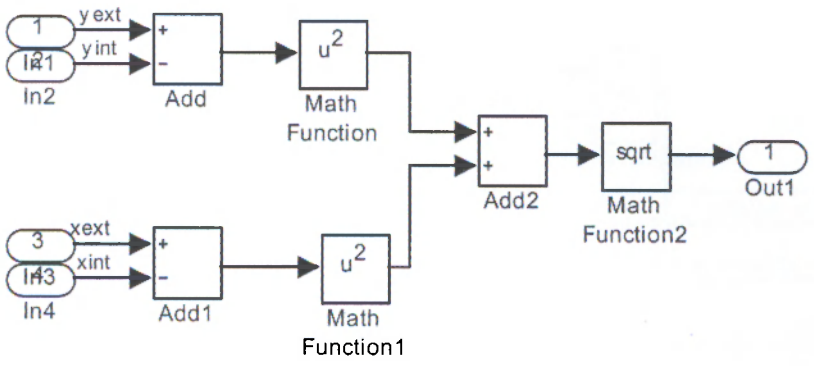
Sub-system blocks for the second term of the equation (18).



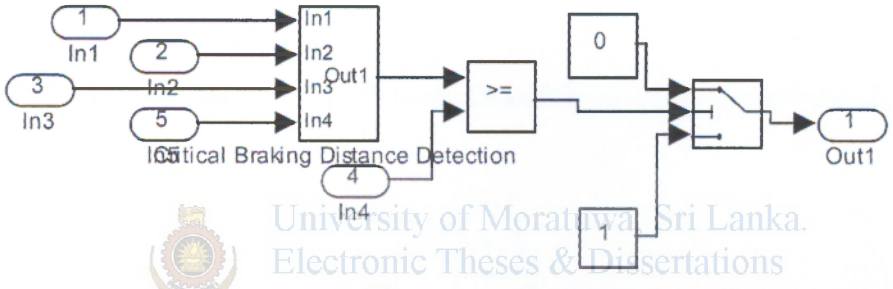
Sub-system blocks for CollisionCondition - (auxiliary function)



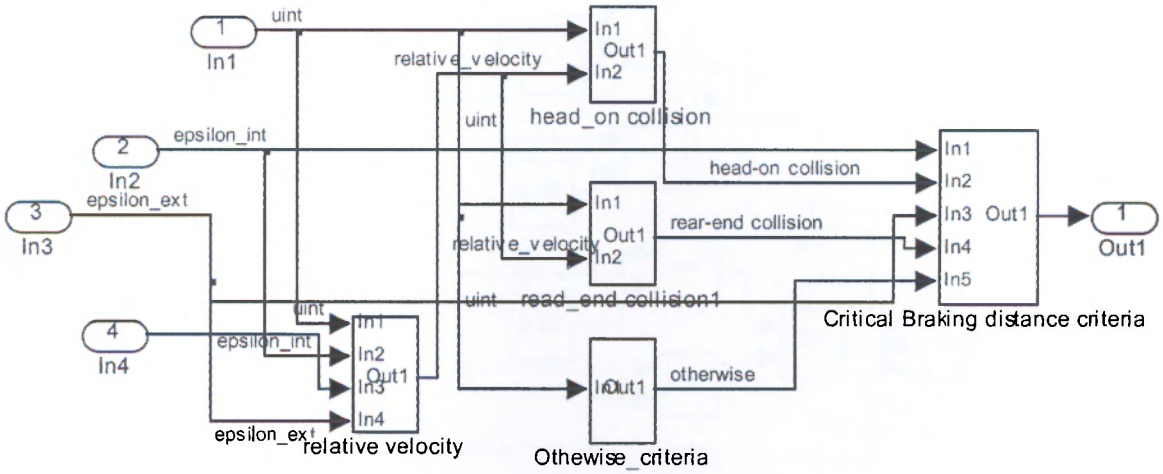
Sub-system blocks for RelativeDistance – (auxiliary functions)



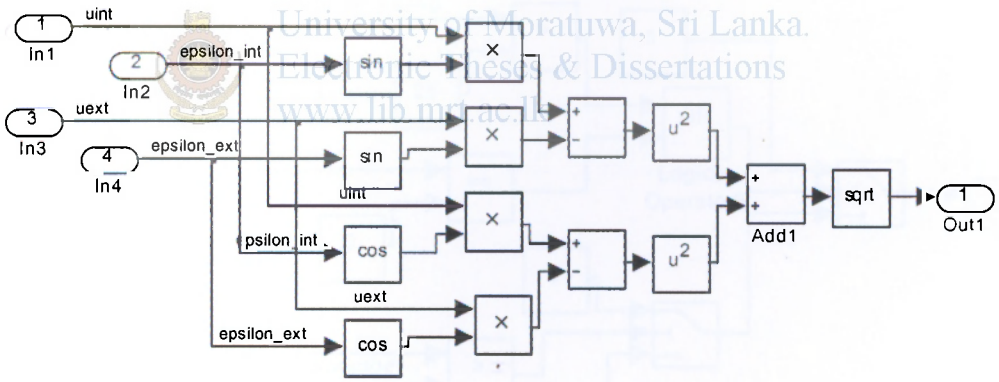
Sub-system blocks for SafetySpeedLimit – (auxiliary functions)



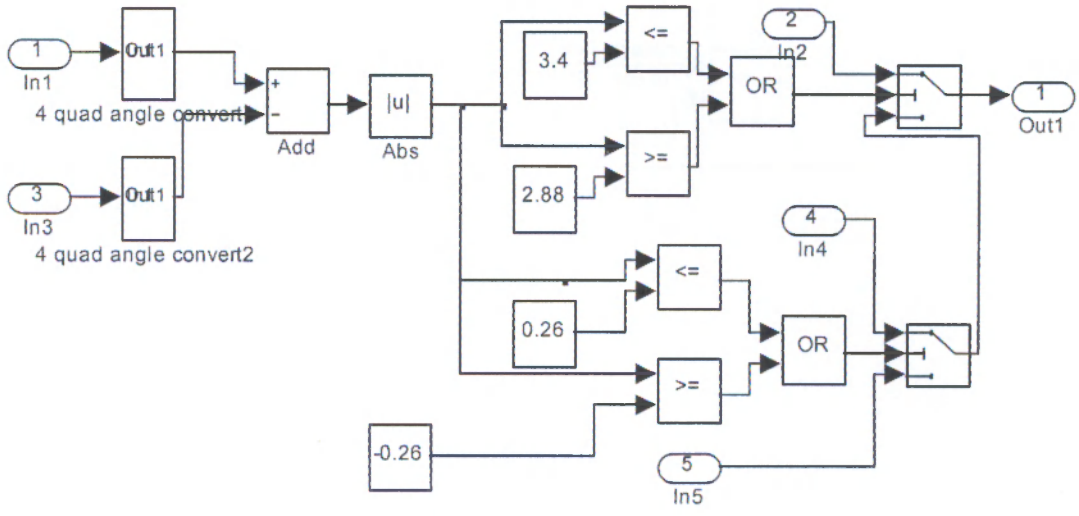
Sub-system blocks for Critical Braking Distance Detection –(in safety speed limit blocks)



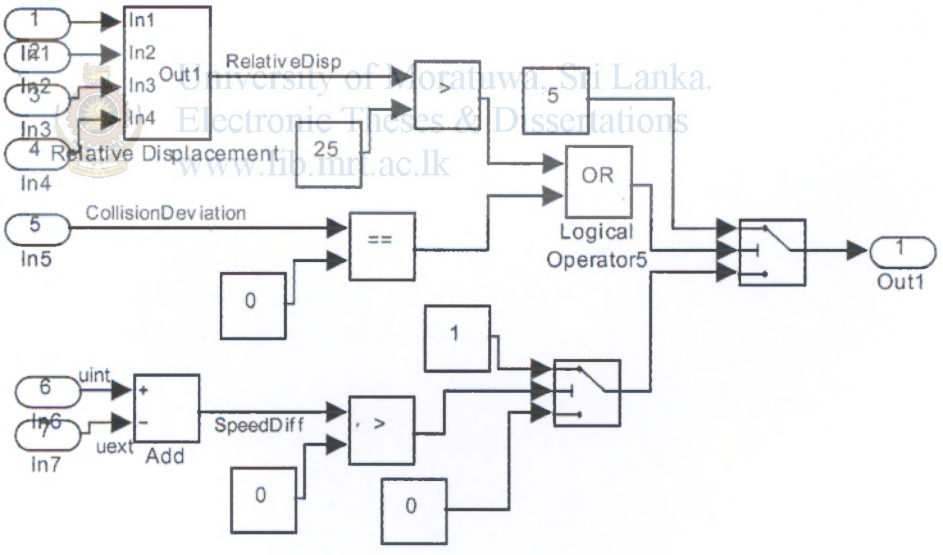
Sub-system blocks for Relative Velocity Detection-(in safety speed limit blocks)



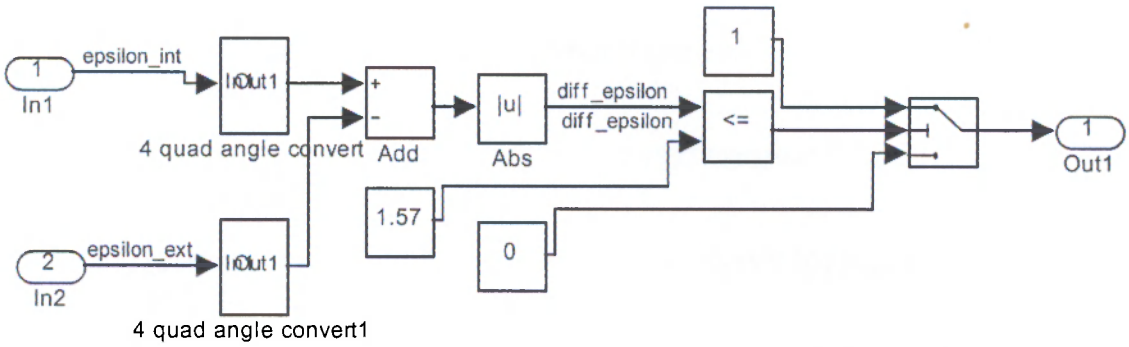
Sub system blocks for Critical Braking Distance Calculation-(in safety speed limit blocks)



Sub system blocks for MSSwitch – (auxiliary function)



Sub system blocks for SteeringDirection –(auxiliary function)



Appendix B

Coefficients of the trained Takagi-Sugeno output membership functions for the ANFIS braking controller

MF1=[1.85493287780397 1.56656347832792e-008 -2.78777090594755e-021 -2.18943002507114e-015 -1.47751322144338]

MF2=[0.00188818805114922 -0.0391046120505595 -5.655468621802166e-023 -4.71960315458807e-017 -4.25502989814296]

MF3=[-0.311206059912852 -2.89230193177666e-006 -5.57726489921515e-018 -5.66733518617125e-018 0.092168684912706]

MF4=[1.78090516036869 0.000448993461274673 -2.65215602105404e-020 -1.61378250847955e-015 -1.33107888133085]

MF5=[0.00143095653842926 -0.695511639183691 4.51766108155919e-022 5.99693924992079e-017 -0.196165342759359]

MF6=[-0.235474150671901 -0.0008022408540194 -4.02386402571529e-022 -2.53532083262607e-016 0.0784913817814176]

MF7=[1.78219754314681 -1.49971542355548e-006 -2.65247457224244e-020 2.86186326363054e-017 -1.32925671149844]

MF8=[0.00143307733892926 -0.000550923380512301 -6.02285906402626e-023 -3.18901862812465e-018 -0.00109487194865714]

MF9=[-0.236575880406084 3.51602726741409e-006 -4.03706286669063e-022 4.51208867943279e-018 0.0788586250234672]

MF10=[1.41236876156383 7.74836548921895e-005 -0.940761632951512 -1.69293387371334e-015 -0.940761632952336]

MF11=[0.000851550495225324 -0.12290699135066 -1.93407614510477 -2.90325205500156e-015 -1.93407614512668]

MF12=[0.0568838067008842 -0.000138967826958646 -0.0189612699600531 -1.4619123355947e-016 -0.0189612699595914]

MF13=[1.43067185425142 0.000322396262969765 -0.928295869708696 -1.1895977977084e-015 -0.928295869730925]

MF14=[0.000880456057858715 -0.498736800920241 0.0158125167511968 -3.76097371711935e-017 0.0158125166457378]

MF15=[0.0422524470949704 -0.00057715193090 9829 -0.0140841501258741 -3.36360450743086e-016 -0.0140841501219997]

MF16=[1.43050272287111 6.95786293633979e-006 -0.928407367059313 1.60510371436439e-017 -0.928407367292285]

MF17=[0.000880185010786668 -0.0124488847441026 -0.00210809437561626 -1.55711640963045e-018 -0.00210809437543177]

MF18=[0.0423910410731516 -1.25851479860497e-005 -0.0141303481217863 5.71738476864628e-018 -0.014130348120857]

MF19=[1.08414267218279e-010 1.67520089870481e-013 -6.92559667040256e-011 -8.94690537838035e-026 -7.16219661069277e-011]

MF20=[6.53655492616008e-014 -5.2493165112919e-016 2.42136558593993e-011 -1.65194642352065e-025 -1.13926111300394e-010]

MF21=[4.36641240708894e-012 -1.06672422411122e-014 -1.45547650880719e-012 2.87960320942741e-027 -1.45547088341401e-012]

MF22=[1.09819223879661e-010 6.80224264591896e-013 -7.12563333225284e-011 -6.15842544549948e-026 -7.12565340126381e-011]

MF23=[6.75843579401656e-014 -1.13952777928779e-014 1.22073674279561e-012 -2.24226483623394e-027 1.21515315719545e-012]

MF24=[3.24330020601853e-012 -4.43025356782097e-014 -1.08110636537503e-012 -6.21616582829639e-027 -1.08110015244337e-012]

MF25=[1.09806241205619e-010 1.68949250317951e-014 -7.12377063463645e-011 8.24981021890162e-028 -7.12596553597257e-011]

MF26=[6.75635524234489e-014 -2.25075188548078e-016 1.43244776521596e-012 -8.70791626630218e-029 1.57034768421012e-013]

MF27=[3.2539386624286e-012 -9.66043596374453e-016 -1.08465254666739e-012 1.26069877150266e-028 -1.08464630466676e-012]

MF28=[1.40954224747582 -0.018891882017138 -2.05923520450994e-020 -0.92989898455487 -0.929898984157109]

MF29=[0.000834673712229795 -0.158311478028552 -3.53143080360956e-020 -1.86035276476269 -1.86035276483816]

MF30=[0.0698185461069066 0.0246760731120892 -1.77822736795729e-021 -0.0232728497244504 -0.023272849727017]

MF31=[1.32101380978095 -0.0815101645262228 -1.4469919343987e-020 -0.687065769800061 -0.687065769595765]

MF32=[0.000481801086359453 -0.48095867278042 -4.57473839665651e-022 0.0255318897079654 0.0255318896925871]

MF33=[0.32382284571127 0.086458193655589 -4.09139005008161e-021 -0.107940949162273 -0.107940949161876]

MF34=[-0.0234682174030531 0.110522246076779 1.95240116715718e-022 0.0121843449987058 0.0121843449989397]

MF35=[-8.54744408335168e-006 -0.0105708296676607 -1.89403081617568e-023 -0.00135772046360727 -0.00135772046371755]

MF36=[-0.00576304733315131 0.0169641008091585 6.95445945046986e-023 0.00192101578831545 0.00192101578831691]

MF37=[1.21647748595465 -0.0174266273547775 -0.720764356304056 -0.720764356369119 -0.720764356382051]

MF38=[0.000573295840690735 -0.158021655897377 -1.23605572304533 -1.23605572323918 -1.2360557234049]

MF39=[0.186722171123285 0.0178399178810316 -0.0622407243688787 -0.0622407243760833 -0.0622407243835856]

MF40=[1.1437093340048 -0.101517684744655 -0.506469688538764 -0.506469688588521 -0.506469688607826]

MF41=[0.000243583665672331 -0.466613433388744 -0.0160122959535769 -0.0160122960655791 -0.016012296173864]

MF42=[0.429615050331612 0.102157517705376 -0.143205017106381 -0.143205017108502 -0.143205017110729]

MF43=[-0.0168017891915329 0.0598667351752015 0.00683370783725823 0.00683370783718899 0.00683370783702196]

MF44=[-2.42584734688325e-006 -0.00457626219347974 -0.000662940252393523 -0.000662940252340684 -0.000662940252343155]

MF45=[-0.0073025070041557 0.0213992347858687 0.00243416900437736 0.00243416900437532 0.00243416900437569]

MF46=[5.2039842657982e-011 2.80764456243822e-017 3.08486327833713e-011 -3.80913313467262e-011 -3.8091331360747e-011]

MF47=[3.80898084118402e-014 3.68989994805484e-016 2.78647409739092e-011 -7.0331400602729e-011 -7.03314005626012e-011]

MF48=[-3.67796187233628e-012 5.81675699789329e-016 2.52404612504839e-011 1.22598724670116e-012 1.22598724689592e-012]

MF49=[4.64091459092637e-011 3.95547012878422e-018 2.44106577034843e-011 -2.62194149544053e-011 -2.62194149392098e-011]

MF50=[2.09226110387692e-014 6.43631175374455e-017 1.43230483292808e-013 -9.54641239373371e-013 -9.54641247690823e-013]

MF51=[7.93957306832783e-012 1.3434346270498e-016 3.07373740224877e-011 -2.64652438369614e-012 -2.64652438697474e-012]

MF52=[-6.71888450119944e-013 9.21089033324976e-019 -3.42067170169733e-013 3.51234579552481e-013 3.51234579819694e-013]

MF53=[-2.49329672958528e-016 2.09104525578082e-017 1.81818386212095e-014 -3.70738384174794e-014 -3.70738384049999e-014]

MF54=[-1.71022249265169e-015 4.91148346166778e-018 -4.91022431025983e-014 5.66740833132776e-015 5.66940834233913e-015]

Appendix C

Coefficients of the trained Takagi-Sugeno output membership functions for the ANFIS steering controller

MF1=[-0.322805222781183 0.00362686564582176 5.79872081449726e-022 0.00459643530203612 0.233700716367951]

MF2=[-0.0141401283986403 0.00321630064447319 1.06921842723302e-022 0.000213828079351621 0.621299384797979]

MF3=[-0.151411132031514 0.000659448306702709 1.96664552202557e-023 0.00394773596868353 0.065251827280593]

MF4=[-0.205016519995446 0.00834115996663714 5.02713143430929e-022 0.00462899768246495 0.173744077047217]

MF5=[-0.0104282735824621 0.0113064529218618 1.0131801389655e-021 0.000231076288179999 0.464134594707426]

MF6=[-0.138182807882656 0.00322993071781877 1.85112013804364e-022 0.00403895685578697 0.0535539157672092]

MF7=[0.00533148927894381 -0.0346608665848006 -1.26216595392427e-023 2.69347057593369e-005 -0.00416699816015225]

MF8=[0.000271052334358062 -0.0640224703149074 -1.47264981032215e-023 1.3693391045203e-006 -0.00811108501259]

MF9=[0.00362214433976581 -0.0117758497539056 -4.9353652489758e-024 1.83029491618058e-005 -0.00140211434468265]

MF10=[-0.17800019493801 0.00133846798138546 0.138865688065683 -0.00450653677097079 0.13886568805093]

MF11=[-0.00899907153467869 0.00034423133578969 0.271422875882059 -0.000224476027383358 0.271422875947102]

MF12=[-0.131513396798413 0.00089641433958575 0.0503030385314004 -0.00403430163598855 0.0503030385339973]

MF13=[-0.164812486804915 0.00502519651146597 0.12930768053794 -0.00453805499180304 0.129307680529237]

MF14=[-0.00831734756202328 0.00289689172334492 0.260609803589193 -0.000226125063567489 0.26060980358097]

MF15=[-0.124916741168366 0.00366326905257527 0.0476144407481921 -0.00404592658589342 0.0476144407180423]

MF16=[0.00451159016967478 -0.0271449023268344 -0.00324653838842148 -4.6688127832172e-005 -0.00324653838837297]

MF17=[0.00022816960915762 9 -0.0293739470705651 -0.00378794415026845 -2.43548050652633e-006 -0.00378794415028086]

MF18=[0.00331670267250217 -0.0106724814854268 -0.00126947274150056 -1.87404839608323e-005 -0.00126947274155334]

MF19=[-9.401027073839e-021 8.03353173348444e-023 7.33446274086731e-021 -2.37838114040954e-022 7.3333548175287e-021]

MF20=[-4.75286759381198e-022 1.67205037230794e-023 1.42973758534986e-020 -1.54185987470079e-023 1.43258934246299e-020]

MF21=[-6.94510915302454e-021 4.90460316170534e-023 2.65708605956778e-021 -2.12965849731965e-022 2.65649851444954e-021]
MF22=[-8.7096618961069e-021 2.65570459024276e-022 6.84658422004529e-021 -2.37817396400502e-022 6.83199549223022e-021]

MF23=[-4.3954747014222e-022 1.51038215970592e-022 1.3731153688715e-020 -1.50273245992969e-023 1.37558444314271e-020]

MF24=[-6.59924585193978e-021 1.93422881478181e-022 2.5201885176145e-021 -2.13071808517929e-022 2.51553762230068e-021]

MF25=[2.38227030481622e-022 -1.43372353711075e-021 -1.71316406976815e-022 -2.45311050426079e-024 -1.71415195811736e-022]

MF26=[1.20481508531602e-023 -1.5456876349406e-021 -1.9922199613713e-022 -4.78202732603647e-026 -1.99868316922924e-022]

MF27=[1.75125980177255e-022 -5.63557006620458e-022 -6.70023598667599e-023 -9.86162002939691e-025 -6.70300909664824e-023]

MF28=[-0.279032466168259 0.000373882171985563 -5.06663288544759e-022 -1.07612448196234 0.163233244072163]

MF29=[-0.0144282663548714 -0.00542410818124636 5.61538479269187e-022 -0.273943098662236 -0.212108617523474]

MF30=[-0.138738023361368 3.53807734622694e-006 -1.98335729743113e-022 -1.03177695745302 0.0566076728865548]

MF31=[-0.27630345449876 0.000822403383236042 -5.11886984753064e-022 -1.08395601228554 0.166903023864185]

MF32=[-0.0142770467337974 -0.0155851027667066 4.60959076955583e-022 -0.212274261377137 -0.149832263550062]

MF33=[-0.139501704462152 -8.13787228674899e-005 -1.98861305387748e-022 -1.03466275021874 0.0567540155790804]

MF34=[-0.00114140212535761 -0.0017501178569574 -5.23914923863933e-024 -0.00616257530978748 0.00111578652920885]

MF35=[-5.86059971371137e-005 0.0350419639463808 -1.31691021775514e-023 0.00385341080411334 0.00422343695428751]

MF36=[-0.000654346082080189 0.000210910116017864 -9.53080203141531e-025 -0.00468563198771975 0.000260235537506062]

MF37=[0.220261881249638 -0.000331448744049112 -0.130323735234651 1.08744361501534 -0.130323735192707]

MF38=[0.0112978808295927 0.00310964774519205 0.144438710630962 0.20509716980539 0.144438710679008]

MF39=[0.128706387544945 -1.38115582205173e-005 -0.0510158397104172 1.03914298126503 -0.0510158397361683]

MF40=[0.218868266370653 -0.000575886714570655 -0.131667372345395 1.09461690828622 -0.131667372338044]

MF41=[0.0112197840553246 0.00986199746114663 0.118567715575698 0.179671781373333 0.118567715605922]

MF42=[0.129279738338163 3.04041103323236e-005 -0.0511510280838205 1.04214911521409 -0.0511510280792445]

MF43=[0.00159561305953783 0.00160510920473727 -0.0013476119423147 0.0112685677547858 -0.0013476119413871]

MF44=[8.36415833212739e-005 -0.0278459067217783 -0.00338735137206393 -0.00272922989350578 -0.00338735137194877]

MF45=[0.000555162964865607 -9.46905808907182e-005 -0.000245150923068885 0.00481894832724281 -0.00024515092092942]

MF46=[1.12600645392293e-020 2.59146064702886e-021 -6.50777085933625e-021 5.7462038220459e-020 -6.80715664305557e-021]

MF47=[5.76893028049699e-022 -2.29599694157831e-022 -2.00523957459086e-021 9.86730128512556e-021 5.70084994485713e-021]

MF48=[6.71996243783951e-021 4.61109769065351e-022 -2.49553404336573e-021
5.48938501700803e-020 -2.65430082252077e-021]

MF49=[9.8255859905167e-021 2.4996835525671e-023 -2.0252066689852e-021
5.82961899502198e-020 -5.96740654345533e-021]

MF50=[5.02377036586726e-022 -2.87528570197262e-024 -2.07878680622196e-021
8.65392546060255e-021 4.59320518303025e-021]

MF51=[6.07841211221842e-021 -4.68339657029822e-024 -1.13015982576662e-021
5.51898463378213e-020 -2.38693708581223e-021]

MF52=[7.98201008753218e-023 8.23977576243958e-024 -3.7794619679078e-023
5.98395920431725e-022 -6.44895791865845e-023]

MF53=[4.20135526082886e-024 5.45571151505902e-024 3.94371786562974e-023 -
1.22291094811302e-022 -1.35213205602259e-022]

MF54=[2.69414860882848e-024 1.75542892362668e-024 -3.91141831656535e-025
2.69110811973662e-023 -1.21722882141599e-023]

Appendix D

Testing program for RF modules (On PC)

The following program is written in Visual Basic language.

```
Dim Situ As Variant
Dim InputV As String
Dim InputData As Single


Private Sub Form_Load()
      MSComm1.PortOpen = True
      Situ = "A"
End Sub

Private Sub Form_Unload(Cancel As Integer)
      MSComm1.PortOpen = False
End Sub

Private Sub Joystick1_JoyMove()

      If Joystick1.XPos > 20 Then
            Situ = "R"
      End If

      If Joystick1.YPos > 20 Then
            Situ = "B"
      End If

      If Joystick1.XPos < -20 Then
            Situ = "L"
      End If

      If Joystick1.YPos < -20 Then
            Situ = "F"
      End If
```

```
    If Joystick1.XPos < 20 And Joystick1.XPos > -20 And Joystick1.YPos < 20
    And Joystick1.YPos > -20 Then
        Situ = "S"
    End If
End Sub


Private Sub MSComm1_OnComm()
    InputV = MSComm1.Input

    Label1.Caption = InputV + Label1.Caption
End Sub
Private Sub Timer1_Timer()
    If Situ = "F" Then
    MSComm1.Output = "d"
    End If

    If Situ = "L" Then
        MSComm1.Output = "e"
    End If

    If Situ = "R" Then
        MSComm1.Output = "f"
    End If

    If Situ = "B" Then
        MSComm1.Output = "g"
    End If

    If Situ = "S" Then
        MSComm1.Output = "h"
    End If
    If Situ = "A" Then
    MSComm1.Output = "6"
    End If

End Sub
```

Appendix E

Testing program for RF modules (On OOpic R+)

They are in the OOpic 'Basic' Language form.

```
'This program creates two oSerialX Objects. One is used to receive a
'Serial signal and the other are used to send a serial signal.
'An oDio1 is used to show that while 'the oSerialX object is waiting
'For incoming serial data, the 'program flow is stopped.

Dim A As New oSerialX
Dim B As New oSerialX
Dim C As New oDIO1
Dim D As New oDIO1
Dim V As New Byte

Dim L As New oPWM
Dim R As New oPWM

Sub Main()
  C.IOLine = 5
  D.IOLine = 6
  C.Direction = cvOutput
  D.Direction = cvOutput
  B.IOLineS = 25
  B.IOLineF = 18
  B.Baud = cv9600
  A.IOLineS = 26
  A.IOLineF = 16
  A.Baud = cv9600


      L.PreScale=2
      L.IOLine=17
      R.PreScale=2
      R.IOLine=18
```

```
    L.Period=79
    R.Period=79
Do

'A=B
A.Value=99'c

'A.String="B"
C.Invert
V=B
'Cruising
If V=100 Then'd
    L.Value=10
    R.Value=10
    L.Operate=1
    R.Operate=1
EndIf


'Left Turn
If V=101 Then'e
    L.Value=14
    R.Value=10
    L.Operate=1
    R.Operate=1
EndIf


'Right Turn
If V=102 Then'f
    L.Value=10
    R.Value=14
    L.Operate=1
    R.Operate=1
EndIf
```

```
    'Reverse
    If V=103 Then   'g
       L.Value=16
       R.Value=16
       L.Operate=1
       R.Operate=1
    EndIf

       ' Stop
    If V=104 Then    'h
    D.Invert
       L.Value=10
       R.Value=10
       L.Operate=0
       R.Operate=0
    EndIf

  Loop
End Sub
```

Appendix F

Testing program for servo motors

```
Dim A As new oPWM
Sub Main()
      A.Value=10
      A.PreScale=2
      A.IOline=1
      A.Period=79

      Do
            A.Operate=1
      Loop

End Sub
```


The potentiometer was adjusted until the rotation of the motor stops.

Appendix G

Program for changing addresses of ultrasonic sensors

The following program is to change the address of the first of ultrasonic sensor in prototype to change its address to xE0,

```
'Change the address of each sonar sensor
'Below are the addresses to be changed for each sonar sensor
' E0 -> 224
' E2 -> 226
' E4 -> 228
' E6 -> 230

Dim A As New oI2C

sub main ()

    A.Node=112   'address of xE0 right shifted/divided by 2
    A.NoInc= cVTrue   'or 1
    A.Mode=cv10bit
    A.width=cv8bit
    A.Location=0

    A=160     'Hex A0
    A=170     'Hex AA
    A=165     'Hex A5
    A=226     'New address to be changed to

end sub
```


When the address is changed successfully, it can be verified by applying power to the particular sonar sensor, which will indicate the new address with a series of long and short pulses of LED.

```
'Single Sonar sensor operation and measurement

Dim Ver As New oByte
Dim SR235 As New oI2C
Spk As oSpeaker


Sub main()

    SRF235.Node=112  ' address of the sonar sensor
    SRF235.Mode=cv10bit
    SRF235.NoInc=1
    SRF235.width=cv8bit
    SRF.Location=0

    Do
        SRF235=81

        Do
            Ver=SRF235
        Loop while Ver=255

        If ((Ver<25) Or (Ver>20)) Then

            Spk.Beep (60757,10, 200)

    Loop

End Sub
```

Appendix H

Testing program for digital compass

```
Dim Compass As New oI2C          ' Create the compass objects
Dim Led As New oDIO1
Dim Bearing As New oByte         ' Store the reading
Spk As oSpeaker

Sub main ()

  Setup                          ' Setup the LED and compass
  Do
    Compass.Location = 1         ' Address of single byte bearing
    Compass.Width = cv8Bit       ' Compass Data is 1-byte wide.
    Bearing = Compass.Value      ' Get the value

    If ((Bearing<5) Or (Bearing>250)) Then
        Spk.Beep(60757,100,200)
      Else                       ' Light the Led if within 5 degrees of north
        Spk.Beep(60757,10,200)
      End If
  Loop

End Sub
Sub Setup()

  Compass.Node = 96          ' Decimal of Hex address 0xC0 shifted right by 1
  Compass.Mode = cv10Bit         ' I2C mode is 10-Bit Addressing.
  Compass.NoInc = 1              ' No increase
  Led.IOLine = 30                ' Pin 28 on 40 way connector
  Led.Direction = cvOutput

End Sub
```

Appendix I

Testing program for optical encoders

```
'Checking optical encoder

Spk As oSpeaker
Dim Encoder As New oQencode
Dim Speed As New oDI08

Sub Main()

    Encoder.IOLine1=31 ' B input
    Encoder.IOLine2=30 ' A input
    Encoder.Operate=cvTrue
    Encoder.Signed=0 ' count from 0 to 65535

    Do
        ooPIC.Delay=50
        Speed=Encoder
        Beep

    Loop

End Sub

Sub Beep()
    If Encoder.Value > 3000 Then
        Spk.Beep(60757,10,200)
    End If
    Encoder.Value=0
End Sub
```