

Navigation and Control of Automated Guided Vehicle using Fuzzy Inference System and Neural Network Technique

A Project Report Submitted in partial fulfillment of the Requirements for the Degree of

Bachelor of Technology

in

Mechanical Engineering

by

Arpan Bandhu

Roll No. 107ME049

and

Atula Kumar Panda

Roll No. 107ME014

Under the supervision of

Dr. Dayal R Parhi

Professor

Department of Mechanical Engineering, NIT Rourkela



National Institute of Technology Rourkela

Navigation and Control of Automated Guided Vehicle using Fuzzy Inference System and Neural Network Technique

A Project Report Submitted in partial fulfillment of the Requirements for the Degree of

Bachelor of Technology

in

Mechanical Engineering

by

Arpan Bandhu

Roll No. 107ME049

and

Atula Kumar Panda

Roll No. 107ME014

Under the supervision of

Dr. Dayal R Parhi

Professor

Department of Mechanical Engineering, NIT Rourkela



National Institute of Technology Rourkela



National Institute of Technology Rourkela

CERTIFICATE

This is to certify that the work in this thesis entitled “**Navigation and Control of Automated Guided Vehicle using Fuzzy Inference System and Neural Network Technique**” by **Arpan Bandhu and Atula Kumar Panda**, has been carried out under my supervision in partial fulfillment of the requirements for the degree of **Bachelor of Technology in Mechanical Engineering** during session 2010 – 2011 in the **Department of Mechanical Engineering, National Institute of Technology, Rourkela.**

To the best of my knowledge, this work has not been submitted to any other University/Institute for the award of any degree or diploma.

Dr. Dayal R. Parhi

(Supervisor)

Professor

Department of Mechanical Engineering

National Institute of Technology Rourkela

ACKNOWLEDGEMENT

We place on record and warmly acknowledge the continuous encouragement, invaluable supervision, timely suggestions and inspired guidance offered by our guide Prof. **Dayal R. Parhi**, Department of Mechanical Engineering, National Institute of Technology, Rourkela, in bringing this report to a successful completion. We consider ourselves extremely fortunate to have had the opportunity to work under the guidance of such a knowledgeable, dynamic and generous personality.

We are also thankful to Prof. **R. K. Sahoo**, Head of Department, Department of Mechanical Engineering; National Institute of Technology Rourkela; for his constant support and encouragement.

Finally, we extend our sincere thanks to all other faculty members and friends at the Department of Mechanical Engineering, National Institute of Technology Rourkela, for their help and valuable advice in every stage for successful completion of this project.

Arpan Bandhu

Roll No.: 107ME049

Atula Kumar Panda

Roll No.: 107ME014

Contents

Abstract	1
Chapter 1 Introduction	2
Chapter 2 Literature Review	5
2.1 Review of Fuzzy Logic Techniques for Vehicle Navigation	6
2.2 Review of Neural Network Techniques for Vehicle Navigation	13
Chapter 3 Analysis of Fuzzy & Neural Technique	19
3.1 Analysis of Fuzzy Logic Technique	20
3.1.1 Fuzzy Logic	20
3.1.2 Fuzzy Set & Membership Function	20
3.1.3 Fuzzy If-Then Rules	21
3.1.4 Fuzzy inference System.....	21
3.2 Analysis of Neural Network Technique.....	24
3.2.1 Introduction to Neural Networks	24
3.2.2 Analogy between human and artificial neural nets	27
3.2.3 The Mathematical Model	28
3.2.4 Types of Neural Networks	29
3.2.4.1 Feed forward networks	29
3.2.4.2 Feedback networks	30
3.2.5 Back Propagation	30
Chapter 4 System Modeling using Fuzzy & Neural Technique	32
4.1 System Modeling using Fuzzy Inference System	33
4.2 System Modelling using Neural Network Technique	46
Chapter 5 Experimental Model	49
Chapter 6 Results and Discussion	54
Chapter 7 Conclusion and Scope for Future Work	58
References	60

List of Figures

Figure 1 Fuzzy Inference System	21
Figure 2 Biological Neural Network.....	26
Figure 3 Artificial Neuron.....	28
Figure 4 Feed Forward Neural Network	30
Figure 5 MATLAB Fuzzy Logic Toolbox.....	39
Figure 6 Triangular Membership function plot for left_obs	40
Figure 7 Triangular Membership function plot for front_obs	40
Figure 8 Triangular Membership Function Plot for Right Obstacle Distance	40
Figure 9 Triangular Membership Function Plot for Heading Angle.....	41
Figure 10 Triangular Membership Function Plot for Vehicle Speed	41
Figure 11 Triangular Membership Function Plot for Steering Angle.....	41
Figure 12 Trapezoidal Membership Function Plot for Left Obstacle Distance	42
Figure 13 Trapezoidal Membership Function Plot for Front Obstacle Distance	42
Figure 14 Trapezoidal Membership Function Plot for Right Obstacle Distance	42
Figure 15 Trapezoidal Membership Function Plot for Heading Angle	43
Figure 16 Trapezoidal Membership Function Plot for Vehicle Speed	43
Figure 17 Trapezoidal Membership Function Plot for Steering Angle	43
Figure 18 Gaussian Membership Function Plot for Left Obstacle Distance	44
Figure 19 Gaussian Membership Function Plot for Front Obstacle Distance	44
Figure 20 Gaussian Membership Function Plot for Right Obstacle Distance	44
Figure 21 Gaussian Membership Function Plot for Heading Angle.....	45
Figure 22 Gaussian Membership Function Plot for Vehicle Speed	45
Figure 23 Gaussian Membership Function Plot for Steering Angle	45
Figure 24 Modelling in MATLAB with Neural Network	47
Figure 25 Performance Plot	48
Figure 26 Training State Plot.....	48
Figure 27 Regression Plot	48
Figure 28 REVA i, the Commercial Vehicle undertaken for the project	50
Figure 29 Central Pinion attached to Steering of AGV. Diameter of Pitch Circle=32mm.....	50
Figure 30 Pinion (spur Gear) attached to the Servo motor for rotating the Central Pinion attached to the Steering of Vehicle. Diameter of pitch circle = 45mm.....	51
Figure 31 Dummy setup (not affixed to steering handle) showing driving gears mating with driven. 51	
Figure 32 Shaft carrying the Central Pinion (to be attached to Steering wheel).....	52
Figure 33 Steering Mechanism.....	53

List of Tables

Table 1 Analogy between Biological & Artificial Neural Network.....	27
Table 2 Fuzzy Rule Matrix	34
Table 3 Comparison of Outputs for a sample set of Inputs.....	56

Abstract:

Automatic motion planning and navigation is the primary task of an Automated Guided Vehicle (AGV) or mobile robot. All such navigation systems consist of a data collection system, a decision making system and a hardware control system. Artificial Intelligence based decision making systems have become increasingly more successful as they are capable of handling large complex calculations and have a good performance under unpredictable and imprecise environments.

This research focuses on developing Fuzzy Logic and Neural Network based implementations for the navigation of an AGV by using heading angle and obstacle distances as inputs to generate the velocity and steering angle as output. The Gaussian, Triangular and Trapezoidal membership functions for the Fuzzy Inference System and the Feed forward back propagation were developed, modelled and simulated on MATLAB. The research presents an evaluation of the four different decision making systems and a study has been conducted to compare their performances.

The hardware control for an AGV should be robust and precise. For practical implementation a prototype, that functions via DC servo motors and a gear systems, was constructed and installed on a commercial vehicle.

CHAPTER 1

Introduction

Introduction

One of the primary functions of Automatic Guided Vehicles (AGVs) or mobile robots is the automatic motion planning required for their autonomous navigation. Mobile robot navigation has increasingly become a field of interest over the last two decades. Initially they were used in industries for the transport of materials in manufacturing industries and have since been used in a range of industries from food and beverage, pharmaceutical to nuclear, automotive and printing.

Mobile robots have become an integral part of space exploration and military missions and in other unreachable or hazardous environments. In such environments the mobile robots are expected to not only traverse a path to a desired location using prior information and by sensing its environment from a range of electronic sensors but the mobile robot is also expected to build a map of the environment about which there is no prior knowledge.

Furthermore, mobile robots are crucial to the advancement in the area of flexible manufacturing systems because they are responsible for transporting materials to and from workstations and warehouses. A truly automated and integrated manufacturing system is made possible only by the presence of mobile robots that can navigate in the changing shop floor or industrial environment.

Also it has been envisioned that commercial vehicles would be able to navigate autonomously. There have been many advances in this area with guided or human aided vehicles initial objective could soon be realized. However, there are still challenges that arise from the efficiency of the system employed, performance in actual traffic conditions and safety and reliability of the system.

The environment in which the AGV or mobile robot navigates is uncertain and is constantly changing. The configuration and orientation of the obstacles vary along with the landscape or topography of the surroundings. The objective of autonomous mobile robot navigation is to build a mobile platform which is capable of successfully navigating in these environments without any human assistance. The outcome of the techniques which navigation methods are based upon; vary in relation to the computational time and resources available to the robot, the dynamic and static memory available to it, the accuracy and efficiency of performance expected from the robot. They are generally classified by the characteristics of the environment in which the mobile robot is to autonomously navigate. The first in this classification is a simple environment which is said to be structured or known. This is followed by a semi-structured or partially known environment with final more difficult to deal with unstructured or unknown environment.

There exist methods which suffer from a heavy reliance on sensory information gathered about the environment, pre-loaded maps, map building and updating. However, due to the inherent nature of the sensors themselves (such as noise) and uncertainties that are always present in a practical environment, these approaches have had restricted success. Fuzzy Logic based technologies and Neural Network applications for computing, categorized under soft computing techniques have come forward as suitable and effective due their inherent capability to handle large complex computations that deals with uncertain or imprecise data.

The overall aim of the research that has been described here is to develop artificial intelligence based navigation mechanisms for an AGV. The techniques explored in this research are restricted to Fuzzy Inference System and Artificial Neural Networks.

CHAPTER 2

Literature Review

Literature Review

2.1 Review of Fuzzy Logic Techniques for Vehicle Navigation

Liu et.al [1] has described a fuzzy logic controller for real-time navigation. It has been shown that path planning and trajectory following is integrated and co-ordinated into a single unit, thus becoming capable of executing manoeuvres such as docking and obstacle avoidance on-line. Only little information from a low-cost sonar system is necessary which is easily available. The controller and the autonomous vehicle can also function in a dynamically changing environment due to a tight coupling between sensor data and controller actions. The driving mechanism is spontaneous in reacting to the sensor data during the navigation of the mobile robot in the environment. Path planning is not required to be separately performed.

Simon et.al [2] has defined mobile robots as devices that can move in their environment with a certain degree of autonomy. Autonomous navigation has been said to be associated with available external sensors capturing information from the environment of the mobile robot through proximity measurements or visual images. The most common proximity sensors are ultrasonic and laser sensors. These proximity sensors are able to detect obstacles and measure the distance from the obstacles. Advanced autonomous robots are required to navigate in indoor environments such as industrial or civil buildings or offices. It is essential for them to have the ability to move through corridors, to follow paths and turn corners and enter open areas if rooms. The research goes on with the proposition of certain control algorithms that are based on artificial vision through cameras used as visual or optical sensors. These control algorithms are applied for navigation along a corridor and for following the path along a wall. In one of the algorithms proposed, using image processing, the robot is guided along the centre axis of the corridor by detection of the perspective lines in the corridor. In another algorithm, two lateral cameras are mounted on the robot for

stereoscopic vision. From the computed optical flow, apparent image velocities on both the cameras are compared for navigation of the autonomous robot. Optic flow computation and its temporal derivatives are used by a single camera to drive the robot along the corridor axis or to follow the path along a wall. For another method, a globally stable control algorithm for wall-following is developed. This is based on incremental encoders and a single sonar sensor. Here there is also the discussion of a fuzzy-based reactive controller that is applied to non-holonomic mobile robots. Also, an ultrasonic sensor has been described to steer an autonomous robot along a concrete path where the edge of the wall and the floor is used as a continuous landmark. Finally, it has been proposed to use the information from an odometric sensor for corridor navigation of a mobile robot.

Kumar, et.al [3] have described the mobile robot as a micro-controller controlled device that is a small four-wheeled mobile platform. This mobile robot is capable of sensing its environment using a range of electronic sensors. Using the information gathered by the electronic sensors the mechanical actuators of the mobile robot help to move it around. A program which is present in the micro-controller determines the behaviour of the mobile robot. The autonomous robot in this research has been designed and constructed so as to perform a range of navigation algorithms. The design of the mobile robot consisted of two main sections which are—electronic analysis of the information captured by the various robot sensors and the programming techniques or navigation algorithms that are used to interface the sensors with the microcontroller that is present on the mobile platform. The research uses IR proximity sensors for obstacle detection and IR sensors also for path guidance of the autonomous robot. The robot is to follow a predetermined path which has varying turns. The fuzzy logic reasoning present in the microcontroller determines the speed of the mobile robot. The results obtained in the research have also been experimentally proven and MATLAB was used to plot the surface viewer graph.

Castillo et.al [4] have used Fuzzy Logic for trajectory tracking and control of an autonomous unicycle robot. A back stepping approach is used in Fuzzy Logic Controllers to obtain the asymptotic stabilization of the robot's position and orientation around the desired trajectory. This also takes into account the kinematics and dynamics of the vehicle. The Mamdani Fuzzy Inference system with nice IF-THEN rules has been used to construct the controller. Input torques and velocities have been taken as linguistic variables and centroid of area method has been used as the method for defuzzification of the Fuzzy sets. The performance of the Fuzzy Logic Controller has also been illustrated by conducting a simulation study.

The environment of a mobile robot can contain modelled and un modelled obstacles that are commonly present in crowded and unpredictable changing environments. Fatmi et.al [5] addresses the challenges of autonomously navigating a mobile robot in such environments. The research says that a successful method of structuring the navigation task to deal with these challenges is to use behaviour based navigation approaches. Thus, issues of individual behavioural design and the action coordinates of the behaviours have been dealt with using Fuzzy Logic. A layered approach has been employed in the method described. Here a supervision layer based on the context is used to decide which behaviour must be processed or activated as opposed to processing all behaviours and then blending the ones found to be appropriate. This achieves reduction of time and the saving of computational resources.

There can be environments containing both static and moving obstacles about which the mobile robot has no prior information available. The research by Malhotra et.al [6] has discussed the design of a mobile robot for such an environment. A design of a Fuzzy brain for the autonomous robot, its integration with both the control system and sensor system has been used for detection of the obstacles in the workspace of the mobile robot. The artificial

potential field method has been made as the strategy for obstacle avoidance. The strategy developed has been implemented by a Fuzzy Logic based system so as to reduce the computational effort required for the implementation of the artificial potential field method. An intelligent obstacle avoidance algorithm was also proposed to integrate the fuzzy system with that main control system.

Singh et.al [7] have employed the following approach for the control of the mobile robot. Desirable traits have been expressed as quantitative preferences that have been defined over the set of all possible control actions. This has been done from the perspective of the goal with which the particular behaviour is designed to be associated with. For instance, the behaviour used to avoid obstacles can map configurations of sonar readings obtained from sensors. The mapped configurations correspond to the presence of obstacles to the left of the robot into a function that prefers actions that steer the robot to the right. So these configurations can be mapped into a function that gives a preference to actions which steer the mobile robot to the right. Results of the control are used to calculate the desirability of the control. The research documents the calculation of the desirability of a control which uses only one level of estimation. It is also proposed that an extension can be made which includes a sequence of controls so that present action can be taken by looking into more future conditions which might be encountered. This resembles game tree techniques that are used in Artificial Intelligence based computer programs. The second extension which is suggested is to take the inputs from the sensors as Fuzzy variables and model them to account for the more realistic noisy and time dependant nature of most sensors.

Ramos et.al [8] have discussed an algorithm for Fuzzy decision making that coordinates the behaviour of the mobile robot. This algorithm belongs to the arbitration class of behaviour control mechanisms. In this class only one behaviour runs at a time. But, it is also possible to

use a hierarchical decision making mechanism for hierarchical behaviours. This method does not have interference between hierarchical levels. This Fuzzy decision method is capable of representing a particular model of the world where in the robot can evolve. To achieve this an algorithm is to be developed which consisting of a defining set of behaviours and a set of world states in addition to a cost function for the behaviours including a set of goals and constraints for the same. A cost function has been computed for each behaviour and actual world state pair and the cost of each pair is evaluated from the overall goals. A Fuzzy operator is used to aggregate the goals and constraints, and the behaviour with the maximum resulting value is taken as the optimum choice. This algorithm was also successfully tested for a robot playing the role of a goalkeeper in a soccer game using practical simulations.

Bidding mechanisms can be used by a group of agents that are responsible for guiding a robot in an unfamiliar environment towards a given target. This approach to land marking has been used by Busquets et.al [9] and is based on Fuzzy Logic. The research starts off with the problem of outdoor navigation stating that it is still an open and difficult challenge in the field of mobile robotics. It has been said that many of the present approaches have made the assumption that an appropriately detailed and accurate metric map of the environment can be got by using sensors that are located on the mobile platform. However, it has been pointed out that many of these sensing methods rely on odometry sensors. These odometry sensors can be unreliable due to their imprecise nature which results in erroneous results. This problem has been attempted to be circumvented by using the collected visual information of the environment. This requires the robot to be able to recognise the important visual objects and to use them for mapping and navigation tasks. This theory incorporating agents imparts flexibility to the solution in problems dealing with dynamic and imprecise environment information. A natural way of providing the agent with the required intelligence can be attained by using computational intelligence along with agents' theory.

Figueiredo, et.al [10] describe a method to use intelligent agents for the control of a robot which has been simulated by a Khepera simulator. The intelligence of these agents is based on a Fuzzy Logic system. These systems have shown by repeated application to the area of mobile robotics that they are an effective procedure for control problems. The behaviour that is expected by each agent at each moment is defined with the help of a set of fuzzy rules. This is based on the position of the robot, its sensor values and heading angle. A path memory system was also developed so that the robot is not stuck by particular obstacles. This allows the robot to look for other alternatives on getting trapped. For a control system that must be capable of avoiding dead end situations, this method has again resulted in a successful combination of computational intelligence and the agents' theory.

Wei et.al [11] have stated in the research that by creating a trading off with low behaviour control an efficient strategy can be formulated. High level global planning for navigation of the robot can be obtained as most applications involve some prior knowledge of the environment. The global planner thus would only require to produce few subgoal positions as opposed to the exact geometric paths. It is relatively simple to remove or add these subgoals from or into the planner. This then reduces the computational time spent on global planning and is also quite flexible as it can re plan in dynamic environments. Simulations were run to check if the strategy discussed could be applied to the particular problem of navigation of mobile robots in complex or dynamic environments. These simulations were successful in showing that the strategy was applicable.

Wei Li [12] described a strategy for robot navigation that uses Fuzzy Logic and is applicable to uncertain environments. It is developed by a process of multisensory integration. Conflicts and competitions arise during multiple reactive behaviors. The study conducted focused on the efficient coordination of these differences using fuzzy sets and a rule base. To execute the

above an array of ultrasonic sensors coupled with a vision system were placed on the mobile robot platform. Distance between the robot and would be obstacles are provided by the ultrasonic sensors which is used in designing the behavior control of the mobile robot. A vision system is incorporated to identify goals which determines the right direction for motion of the mobile robot and to avoid getting trapped in a local region. On simulation the strategy discussed shows that it can successfully be applied for efficient navigation of the mobile robot in both complex as well as uncertain environments by the application of various behaviors. These behaviors range from slowing down at narrow and curved roads, maneuvering around obstacles, moving towards the destination and escaping U-shaped objects.

Tso et.al [13] have in their research have pointed out that the performance of automatic guided vehicles is measured by the errors in their position and orientation. It so happens at times the orientation errors are not measurable. And the output feedback employed is obtained using position errors. A controller that uses out-put feedback is designed for an AGV with differential-wheel-drive. Opto-sensors are present in a simple linear array of 64x1. At first a linear controller is designed whose performance is further improved by the implementation of Fuzzy Logic. Accordingly, two fuzzy-logic controllers are formed. The Hurwitz criterion, the Popov criterion, and the phase-plane method are used to demonstrate the stability of the system which is verified by experiments conducted using a laboratory AGV.

2.2 Review of Neural Network Techniques for Vehicle Navigation

Birsel Ayrulu and Billur Barshan [21] have used neural networks and investigated the processing of sonar signals for the robustly differentiating commonly encountered features that are found in indoor robot environments. This differentiation of features is significant for a range of applications for intelligent systems. Here processing of various representations of amplitude and times of flight measurement patterns are done. These patterns are in turn obtained from a real sonar system. In this study, modular and non-modular neural networks have been trained. The training method used is back-propagation with the use of generating-shrinking algorithms for to develop a learning process for identification of relations between parameters for the target primitives. The networks which are trained with this mentioned generating-shrinking algorithm have shown better generalization and interpolation capability as well as a faster convergence rate. It is found that neural networks can differentiate more targets using only a single sensor node. Also, it results in a higher correct differentiation percentage than what was achieved by using methods that existed previously that depended on multiple sensor nodes. A sensor node has been defined as a pair of transducers with fixed separation, that can rotate and scan the target to collect data. It has been said that the performance of the other methods would have been worse if the number of sensing nodes were reduced. The neural network approach is successful and it demonstrates that sonar signals can give information sufficient for differentiating all types of targets. The examples shown here were system control based on acoustic signal detection and identification, obstacle avoidance, map-building, target-tracking. navigation etc for autonomous mobile robots as well as other intelligent systems.

Franco Scarselli and Ah Chung Tsoi [22] have researched into recent works on approximation methods by feedforward neural networks. Their work has focused on mainly on the problems seen in computational aspects of the methods. In particular, the work

determines the chance of developing a feed forward neural network that can deliver a predetermined degree of accuracy of approximation. Also, the number of hidden layers which will be required for the feed forward neural network to attain the predetermined accuracy of approximation is also looked into. A unifying framework was proposed that helps to understand the existing approaches for investigating the problem of universal approximation by using feedforward neural networks. Subsequently, two algorithms for training are proposed. These training algorithms can determine (to any required level of accuracy) the corresponding weights of the given inputs in the feed forward neural network where the activation function used is the sigmoid function. The training algorithms are successful in escaping from local minima; a problem which commonly plagues several neural network training algorithms.

Soylu et.al [23] in their research have considered a special type of Automatic Guided Vehicle or AGV routing problem. For a single, free ranging AGV the problem identified is to find the shortest tour through which it can carry out multiple number of pick and deliver or P&D tasks. This specific objective is an instance of the asymmetric travelling salesman problem that is understood to be NO-complete. Based on Kohonen's self-organizing maps an Artificial Neural Network (ANN) algorithm has been introduced. For different pick and deliver tasks and for various parameter settings, the performance of the introduced algorithm is tested. These results are compared with those obtained from nearest neighbor rule and optimal solution rule. This research has positive results for sound quality and computation time involved in the artificial neural network algorithm.

Orille et.al [24] have in their research discussed the use of a multi-layer neural network that develops on an analogy with the traditional switching look up table method for induction motor Direct Torque Control (DTC) that obtains optimal switching patterns. The

MATLAB/SIMULINK program has been used extensively for the developing the system. The full simulation of the system ie the Direct Torque Control, the induction motor including the the inverter switch and firing circuit, as well as the control unit; have all been simulated using MATLAB. The appropriate neural network was chosen that represents the switching look-up table, an effective neural network configuration is designed and also tested upon. Finally, the neural network representing the switching look-up table is tested as part of the Direct Torque Control (DTC) so as to find out the stability and reliability of the system thus developed. The simulation result of the DTC that is obtained by the switching look-up table is used as a reference to evaluate the performance of the neural network in its simulation for the DTC.

Yang and Meng [25] have looked at the mobile robot navigation problem of real time and collision free motion planning from the approach of biologically inspired neural networks. This can also be applied to manipulators on a non-stationary environment. Every neuron that is present in the topologically organized neural network is provided with only local connections. The neural dynamics of these local connections are characterized by a shunting equation. Therefore now, the size of the neural network determines the computational complexity in a linear relationship. There isn't any prior knowledge of the dynamic and changing environment, no explicit search over the free workspace and the collision paths, and no learning procedures are involved. Thus the real-time motion of the mobile robot or the manipulator is planned through the dynamic activity landscape of the neural network and this method is computationally efficient. The qualitative analysis conducted and the Lyapunov stability theory confirms the global stability of the neural network. Furthermore, the simulation studies conducted give a fair idea of the effectiveness and efficiency of the approach that has been proposed.

Barbera et.al [26] have focused on the objective of developing a mobile robot that can navigate in an unknown and unstructured environment while performing delivery tasks. The mobile robot is supplied with unreliable sonar as well as infra-red sensors. A method for sensor fusion is also developed so as to cope with the unreliable nature of the sensors. The

lack of a prior model of the environment is the primary difficulty in applying classical fusion methods. The robot first has to carry out process for map building. It has been shown that the simple existing methods of sensor fusion do not address each of the specific requirements of the particular robot tasks. Thus it has been proposed to use neural networks to achieve the function of sensor fusion. This approach delivers a more reliable data set. Also, the training procedure of the neural network has been automated. The sensor fusion method thus developed has been used to validate the map building process.

Chronis and Skubic [27] have in their research worked on the difficulties in programming robots. A programming by demonstration or PbD paradigm has been discussed. This paradigm extracts robot behavior from the control actions demonstrated. The work is in an attempt to develop robot programming methods that allow the definition of the task use by domain experts for robots as semi-autonomous tools. Due to this, there was an intention of injecting into the acquired behavior the biases of the human trainer. For these reasons, the programming by demonstration paradigm was chosen as opposed to preferring an autonomous learning method. Thus the study tests the feasibility of training a neural network from demonstrated navigation actions. The results from using three different training data collection methods were compared. The three methods were a mouse driven software joystick, a novel PDA interface and a programmed control. A neural network configuration was developed that can be used in training a mobile robot for the purpose of corridor following behavior. A consistent mapping between inputs and outputs of the network from the training data set is necessary for good level of convergence of the feed forward MLP. For robust control to be provided in a range of conditions the training data must also include the complete range of possible sensor variations. Experimental results show that the most robust behaviors are produced by the PDA generated training sets.

Davis [28] developed MAMMOTH which is a Modular Architecture Multi-Modal Theory. It is a neural network paradigm. This paradigm is an architectural blueprint as well as a training system that combines the internal representations of multiple neural networks where each neural network is trained so that it can recognize different kinds of features. Functional decomposition of a task is achieved by the modules present in a MAMMOTH. Functional decomposition means that for a given input, every module completes a portion of the overall task and these results are combined by the higher levels of the MAMMOTH network, thus giving the solution. The MAMMOTH networks have been applied to several tasks that include vision for alignment to be conducted by an inspection robot, on-road navigation as well as cross-country navigation. Finally the advantage of using MAMMOTH has been seen as its ability for learning low level features separately as well as in parallel which speeds up the overall training process for the neural network without any trade off with the performance of the neural network.

Pomerleau [29] describes a learning system--ALVINN (Autonomous Land Vehicle In a Neural Network) This is applicable to the problem of vision based autonomous driving or navigation. The challenges in this domain are the constantly changing environment and the simultaneous real time processing of the information. Thus the flexibility and efficiency of the neural network is of significance. ALVINN allows driving in a range of circumstances that include single lane paved and unpaved roads, lined and unlined multilane roads. It can also deal with obstacle-ridden on-and off road environments.

Glasius et.al [30] have shown that effective path planning and obstacle avoidance can be achieved by a model of a topologically organized neural network. This is of Hopfield type and has nonlinear analog neurons. It is a deterministic system which is capable of instantaneously giving a suitable path that starts from an arbitrary starting position and can

direct to any target position. It in the process also avoids static as well as moving obstacles that can have any arbitrary shape. The model makes the assumption that an external input given, activates a target neuron that corresponds to the target position and specifies all the obstacles that can be found in the topologically ordered neural map. The path comes from the neural network dynamics and the neural activity gradient which is in the topologically ordered neural map. Computer simulations and analytical results together give a fair idea of the performance of the network.

CHAPTER 3

Analysis of Fuzzy & Neural Technique

Analysis of Fuzzy & Neural Technique

3.1 Analysis of Fuzzy Logic Technique

3.1.1 Fuzzy Logic:

Unlike Crisp logic based on binary sets which is essentially a two-valued logic, “fuzzy logic” is a form of multi-valued logic and is based on fuzzy set theory. To deal with fluid or approximate reasoning, fuzzy logic variables can take a truth value that ranges in degree between 0 and 1. Fuzzy logic is a super set of conventional logic that has been extended to handle the concept of partial truth: the truth values between completely true and completely false.

3.1.2 Fuzzy Set & Membership Function

A fuzzy set is a set without a crisp, clearly defined boundary. It can contain elements with only a partial degree of membership.

A membership function (MF) is a mapping from an input space (often referred to as the universe of discourse) to a membership value between 0 and 1. In fuzzy logic, it represents the degree of truth as an extension of valuation.

A classical set can be expressed as

$$A = \{x \mid x < 13\}$$

If X is the universe of discourse and its elements are denoted by x , then a fuzzy set A in X is defined as a set of ordered pairs.

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

$\mu_A(x)$ is the membership function of x in A .

3.1.3 Fuzzy If-Then Rules:

Fuzzy if-then rule statements are conditional statements that comprise fuzzy logic. A single fuzzy if-then rule is of the form

if x is A then y is B

Where A and B are linguistic values defined by fuzzy sets on the ranges (universes of discourse) X and Y, respectively. The if-part of the rule "x is A" is called the antecedent or premise, while the then-part of the rule "y is B" is called the consequent or conclusion.

3.1.4 Fuzzy inference System

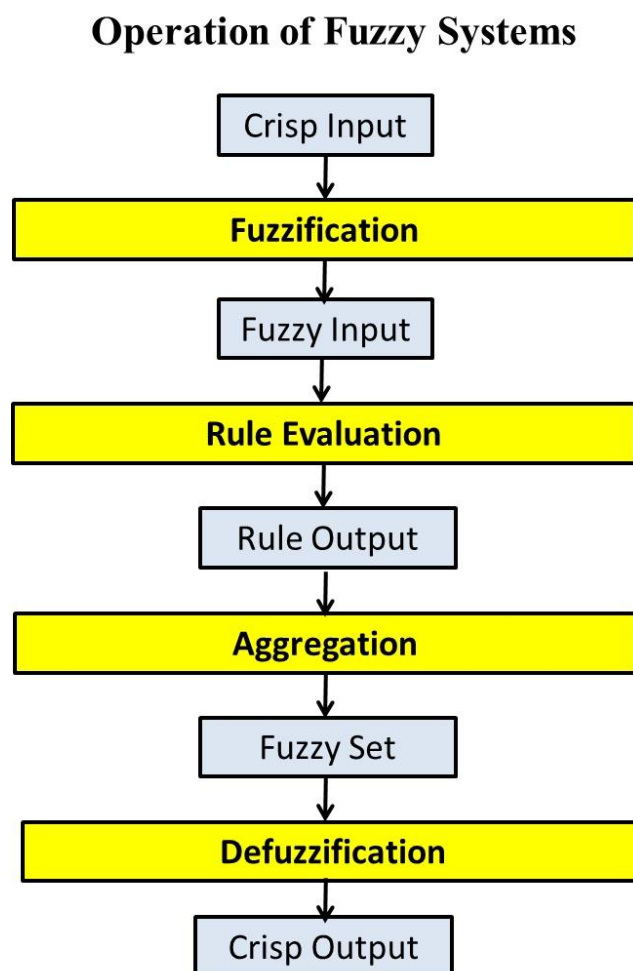


Figure 1 Fuzzy Inference System

Steps in a fuzzy inference System:

Step 1: Fuzzification

The first step in a fuzzy inference system is the fuzzification of crisp inputs. It transforms the exact logic problem into a fuzzy logic problem. Unlike crisp logic, fuzzy logic deals with linguistic variables instead of numerical variables. The process of converting numerical variables of the problem into grades of membership for linguistic terms of fuzzy sets is called fuzzification. Thus it is a mapping from a certain input space to fuzzy sets in certain input universes of discourse.

Step 2: Rule Evaluation

The next step in the fuzzy inference system is to apply the fuzzified inputs to the antecedents of the fuzzy rules. In case a given fuzzy rule has more than one antecedent, we make use of the fuzzy operator AND or OR in order to obtain a single truth value that would represent the result of the antecedent evaluation.

To evaluate the conjunction(intersection) & disjunction(union) of the rule antecedents, the fuzzy operators AND & OR are used respectively.

$$\text{AND: } \mu_{A \cap B}(x) = \min [\mu_A(x), \mu_B(x)]$$

$$\text{OR: } \mu_{A \cup B}(x) = \max [\mu_A(x), \mu_B(x)]$$

Then the result of the evaluation would be applied to the consequent membership function.

There are two main methods of doing so:

1. Clipping:

This involves cutting (alpha-cut) the consequent membership function at the level of result of the antecedent evaluation. As the top of the membership function is sliced, some information

loss is inevitable in case of clipping. Still it is often preferred as it doesn't involve too complex mathematics.

2. Scaling:

In this case, the membership functions of the rule consequent are adjusted by multiplying all its membership degrees by the truth value of the rule antecedent. There is not much loss of information.

Step 3: Aggregation of the rule outputs

This is the process of uniting the outputs of all rules that are invoked for a particular set of inputs into a single fuzzy set. The clipped or scaled consequent membership functions serve as the input to the aggregation process and the output of the process is one fuzzy set for each output variable.

Step 4: Defuzzification

It is the final step in the fuzzy inference process. It is the reverse process of Fuzzification. The Fuzzy Logic Controller (FLC) produces required output in a fuzzy set; however the final output has to be a crisp value. Defuzzification involves taking the aggregate output fuzzy set and producing a single crisp value corresponding to each of the output variables. Of various defuzzification methods, the centroid defuzzification method is most commonly used. Centroid defuzzification method involves finding a point that would represent the Centre of gravity of the aggregate fuzzy set. Mathematically this point can be expressed as:

$$COG = \frac{\int_a^b \mu_A(x)xdx}{\int_a^b \mu_A(x)dx}$$

3.2 Analysis of Neural Network Technique

3.2.1 Introduction to Neural Networks

An Artificial Neural Network (ANN) is a paradigm for information processing which has been developed from an analogy with biological nervous system and is similar in manner to how the brain processes information. It bears a strong resemblance to the axons and dendrites which are present in the nervous system and is an abstract simulation of a real nervous system that contains a collection of neuron units communicating with each other via axon connections. The central idea for this paradigm is the novel structure developed for the information processing system. It is the result of a large number of highly interconnected processing elements (neurons) which work together to solve a given problem.

An Artificial Neural Network works by example which is achieved by the activity of training the network. This feature is similar to how humans learn. An Artificial Neural Network is generally configured with the help of a learning process specifically for a particular application. These applications range from image processing, optimization, decision making , neural networks can be used for a variety of data mining tasks, among which are classification, descriptive modeling, clustering, function approximation, time series prediction etc. The complexities of biological neural networks can be better understood with the help of artificial neural networks. Some artificial intelligence problems are solved by neural networks without necessarily creating a model of a real biological system. Biological systems learn by adjusting the synaptic connections existing between neurons. Artificial Neural Networks also incorporate this feature.

The human brain consists of neurons which are highly interconnected through dendrites and axons. Information is transmitted between the neurons through activation signals that pass from one neuron to another through the axons and dendrites that connect them. This is how

intelligence is achieved in human brain. The artificial neural network can be understood as an algorithmic version of the biological neural network explained above. They too consist of interconnected neurons which communicate through activation signals. Consequently, the Artificial Neural Network can approximate a function of multiple inputs and outputs that apply to the specific application being considered. The biological neural network, the Artificial Neural Network and their analogies have been explained below.

The brain is responsible for the simultaneous processing of a multitude of information (ranging from colors to shapes and sounds) in highly variable environments. The brain is able to accomplish such tasks by means of multiple parallel processing elements. Artificial Neural Networks incorporate the same idea of parallel processing to the computer in order to simulate and thereby take advantage of the brain's computing strategies.

The neuron essentially consists of three parts, which are:

1. The neuron cell body or cyton,
2. Dendrites which are branching extensions from the cyton for receiving input, and
3. Axon which carries the neuron's output to the dendrites of other neurons.

A neuron gets input signal from other neurons through a series of *dendrites*. Synapses are the junction between neurons which are responsible the transfer of signals from one neuron to another. They are of several types and determine among other properties, the speed and information in a signal. It has been estimated that the human nervous system comprises of over 100 billion neurons and synapses.

The neuron sends out spikes of electrical signals through a long, thin stand known as the *axon* which splits into a number of branches. The synapses are located at the ends of each branch. The synapses convert this into electrical effects. These electric effects either inhibit or excite activity in the neurons to which the branch of the axon is connected. When the next neuron in

the information pathway receives excitatory input that is large enough in relation to its inhibitory input, it sends a spike of electrical activity down its axon. The network learns by adjusting the effectiveness of the synapses so that the influence of the neurons on each other changes. However, the interaction between neurons is still not fully understood. This interaction is again different for different neurons.

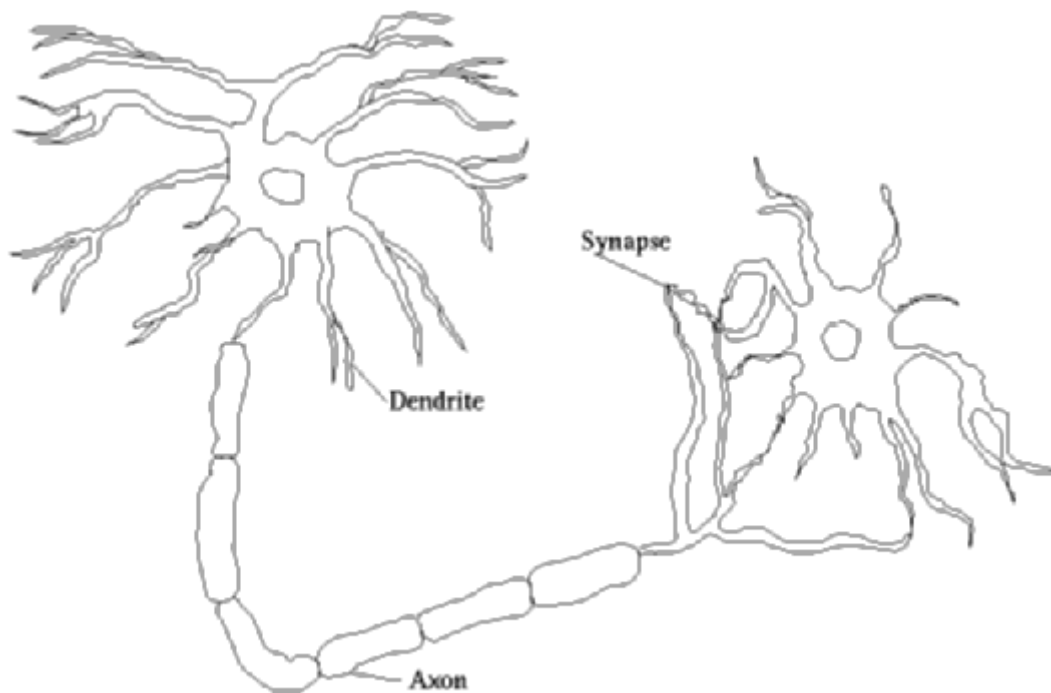


Figure 2 Biological Neural Network

The process by which the action potentials (which depend on the neuron's potential) travel down the axon is often modeled as a propagation rule represented by a net value u .

All the incoming signals at the synapses of the neuron are collected. The excitatory and inhibitory influences acting on the neuron are thus summed. This output is a nonlinear function of the inputs and the strength of the connection in the synapses can be modified by learning. When the excitatory influences become dominant, then the neuron fires and sends the message to other neurons via the outgoing synapses. When the combined signal strength

exceeds a certain threshold it causes firing. In the general case, an activation function $f(x)$ gives the value of the neuron.

The brain learns by modifying the strength and nature of the synapses. Artificial neural networks attempt to model algorithmically the very same behavior. The biological learning is simulated mathematically in artificial neural networks by the positive or negative reinforcement of connections and weights.

3.2.2 Analogy between human and artificial neural nets

An artificial neuron has a processing element which has several input connections. These input connections each have an associated weight and a transfer function to calculate the output for the given output connection. A network of these artificial neurons forms an artificial neural network. A set of given inputs and corresponding outputs, called a training set or training pattern, is fed to the network. Adjusting of the weights of the connections in the net results in training of the network. Adding an extra connection to each neuron with an input value of -1 in addition to a weight representing the threshold, adjusts the threshold values and weights. If the sum is now greater than zero then the neuron fires. A summary of the analogy between a human and artificial neural network has been tabulated below.

Table 1 Analogy between Biological & Artificial Neural Network

Human	Artificial
Neuron	Processing Element
Dendrites	Combining Function
Cell Body	Transfer Function
Axons	Element Output
Synapses	Weight

Real neurons differ in many respects in comparison to the artificial neurons developed from them. They have elaborate dendritic mechanisms and do not only function on the weighted sum of the inputs. They also do not stay *on* until the inputs change. Information is also encoded using complex pulse arrangements. Brains, containing neurons in the order of 11^{10} , are considerably more complex than artificial neural networks which are restricted to a few thousand nodes. Thus there is a difference in the degree of complexity and the resulting computational power.

3.2.3 The Mathematical Model

In the neuron of an Artificial Neural Network the weight is a number which represents a synapse from a biological neural network. When the weight is negative it denotes an inhibitory connection, whereas a positive weight denotes an excitatory connection. All inputs are summed together and are also modified by their corresponding weights. This process is referred to as a linear combination. The amplitude of the output is controlled by an activation function.

The figure describes the mathematical process.

Consider an artificial neuron having N input, namely u_1, u_2, \dots, u_N .

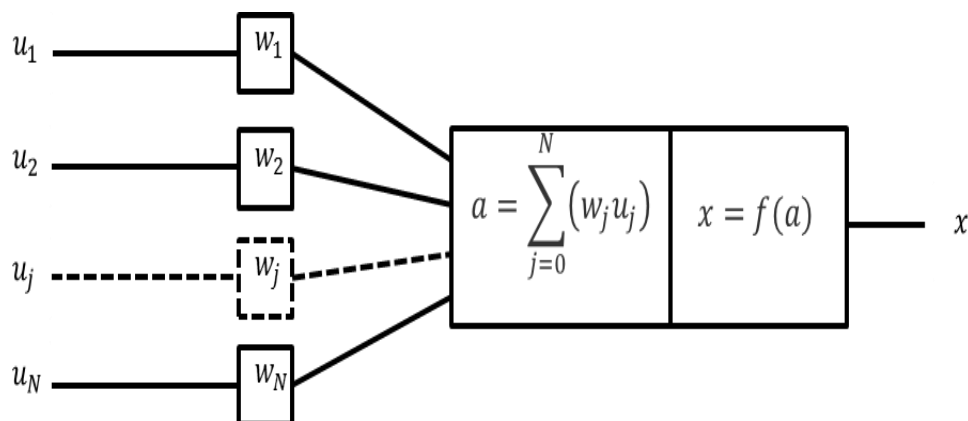


Figure 3 Artificial Neuron

Each of the line connecting these inputs to the neuron is assigned some weight, denoted by w_1, w_2, \dots, w_N respectively. For the particular graded potential the activation is determined by the formula: $w_j u_j$. This model describes the interval activity of the neuron as

$$a = \sum_{j=0}^N (w_j u_j)$$

The weighted sum value 'a' is the net input to the unit & the output of the activation function on the value of a, is the output of the neuron.

$$x = f(a)$$

The unit's activation function is the function f . In the simplest case, the unit's output is its net input and f is an identity function.

3.2.4 Types of Neural Networks

3.2.4.1 Feed forward networks:

Feed-forward networks can have signals travelling in one direction only; i.e. from input to output. These are characterized by absence of feedback loops in the network. So the output of any layer does not affect the same layer. Feed-forward ANNs are straight forward networks that associate inputs directly with outputs. They are extensively used in pattern recognition. This type of organization is also referred to as bottom-up.

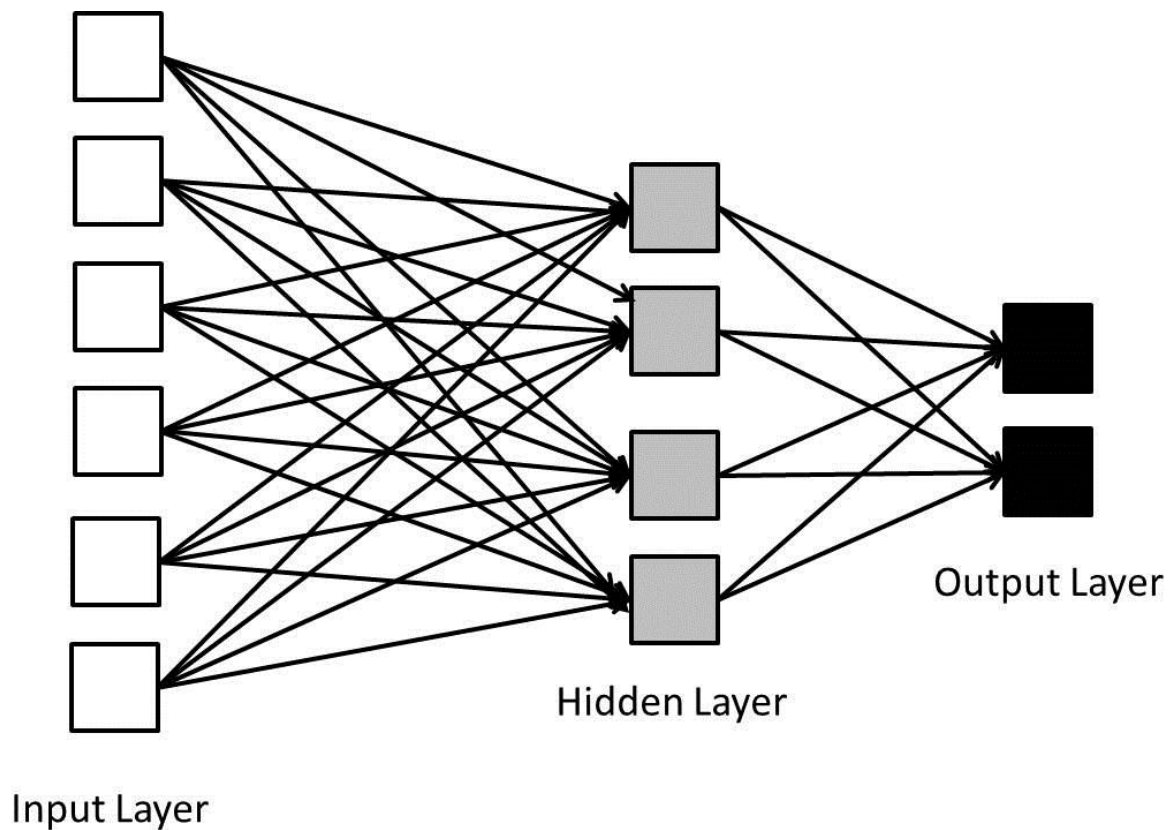


Figure 4 Feed Forward Neural Network

3.2.4.2 Feedback networks:

Feedback networks allow signals to move in both directions. This is made possible by introduction of feedback loops in the network. These are very powerful & dynamic. Their state changes continuously until they attain equilibrium & remain at the point until the input changes and some new equilibrium is found. These are also called interactive or recurrent networks.

3.2.5 Back Propagation

Back propagation is a supervised learning method of training artificial neural networks and implements the Delta rule. It is most useful for feed-forward networks.

Back propagation technique:

1. A training sample is presented to the neural network and the network's output is obtained.
2. The network's output is compared to the desired output as supplied by the training sample and the error at each output neuron is calculated.
3. For each neuron, the local error is calculated which shows how much lower or higher the output must be adjusted to match the desired output.
4. The weight of each neuron is adjusted to lower the local error.
5. Blames are assigned for the local error to neurons at the previous level, giving greater responsibility to neurons connected by stronger weights.
6. Steps 3 to 5 are repeated on the neurons at the previous level, using each one's blame as its error.

CHAPTER 4

System Modeling using Fuzzy & Neural Technique

System Modeling using Fuzzy & Neural Technique

4.1 System Modeling using Fuzzy Inference System

The vehicle used here is a commercial electric vehicle 'Reva i'. In order to acquire information about dynamic environment, the vehicle is equipped with 3 arrays of ultrasonic sensors for measuring the distances to nearest obstacles around it. one on the front & two on the two sides of the vehicle. Each of these sensor measures distance between the vehicle & obstacle in real world & provide the front obstacle distance (d_{front}), left obstacle distance (d_{left}), right obstacle distance (d_{right}). In order to locate the target, the vehicle should keep track of its position & the target location. This is achieved by using encoders, GPS navigator & heading compass.

The presence of obstacles in its path acts as repulsive forces for avoiding the obstacles, and the bearing of the target acts as an attractive force between vehicle and target.

From the sensory data we build a simple model for representation of distances between the vehicle & obstacles as follows:

Let d_{1i} , d_{2i} & d_{3i} be the distances measured by the left, front & right sensor array respectively.

($i = 1,2,3,4,5,\dots$)

$$d_{left} = \text{Min}\{ d_{1i} \}$$

$$d_{front} = \text{Min}\{ d_{2i} \}$$

$$d_{right} = \text{Min}\{ d_{3i} \}$$

The minimum values d_{left} , d_{front} , d_{right} now express the distance between the vehicle and the obstacles to the left, front & right respectively. The input to the fuzzy logic scheme are the distances between the vehicle & obstacles to the left, front & right denoted by d_{left} , d_{front} , d_{right} respectively & the heading angle ang_{head} which denotes the destination

position with respect to the vehicle. The output of the fuzzy controller is the vehicle speed & the vehicle steering angle denoted by v & ang_steer respectively.

In this research three types of membership functions are considered for each of the input and output variables: Triangular, Trapezoidal & Gaussian.

The linguistic variables such as “far”, “medium” and “near” are taken for the three membership functions of d_left , d_front , d_right . To fuzzify vehicle speed v , the linguistic variables “fast”, “med” & “slow” would be used for its three membership function. For the heading angle ang_head & vehicle steering angle ang_steer five membership functions are considered with the linguistic variables “Big Left (BL)”, “Small Left (SL)”, “Straight (S)”, “Small Right(SR)” & “Big Right (BR)” .

A set of fuzzy logic rules is used to describe the reactive behaviors. According to the information acquired by the sensors, some of the rules are fired. The outputs of those rules are weighted by fuzzy reasoning and then the velocity and steering angle are calculated. The fuzzy logic rules that we have chosen are as follows:

Table 2 Fuzzy Rule Matrix

Left_obs	Front_obs	Right_obs	Head_ang	V_Smededall	Steer_ang
Far	Far	Far	BR	Med	BR
Far	Far	Far	SR	Fast	SR
Far	Far	Far	S	Fast	S
Far	Far	Far	SL	Fast	SL
Far	Far	Far	BL	Med	BL
Far	Far	Med	BR	Med	SR
Far	Far	Med	SR	Med	SR
Far	Far	Med	S	Fast	S
Far	Far	Med	SL	Fast	SL
Far	Far	Med	BL	Med	BL

Far	Far	Near	BR	Slow	S
Far	Far	Near	SR	Slow	S
Far	Far	Near	S	Med	S
Far	Far	Near	SL	Med	SL
Far	Far	Near	BL	Med	BL
Far	Med	Far	BR	Slow	BR
Far	Med	Far	SR	Med	SR
Far	Med	Far	S	Med	S
Far	Med	Far	SL	Fast	SL
Far	Med	Far	BL	Slow	BL
Far	Med	Med	BR	Slow	SR
Far	Med	Med	SR	Slow	SR
Far	Med	Med	S	Slow	S
Far	Med	Med	SL	Slow	SL
Far	Med	Med	BL	Slow	BL
Far	Med	Near	BR	Slow	BR
Far	Med	Near	SR	Med	SR
Far	Med	Near	S	Slow	S
Far	Med	Near	SL	Med	SL
Far	Med	Near	BL	BL	S
Far	Near	Far	BR	Slow	BR
Far	Near	Far	SR	Slow	SR
Far	Near	Far	S	Slow	SR
Far	Near	Far	SL	Slow	SL
Far	Near	Far	BL	Slow	BL
Far	Near	Med	BR	Slow	BR
Far	Near	Med	SR	Slow	SR
Far	Near	Med	S	Slow	BL
Far	Near	Med	SL	Med	SL
Far	Near	Med	BL	Slow	BL
Far	Near	Near	BR	Slow	BL
Far	Near	Near	SR	Slow	BL
Far	Near	Near	S	Slow	BL
Far	Near	Near	SL	Slow	SL
Far	Near	Near	BL	Slow	BL

Med	Far	Far	BR	Slow	BR
Med	Far	Far	SR	Fast	SR
Med	Far	Far	S	Fast	S
Med	Far	Far	SL	Slow	SL
Med	Far	Far	BL	Slow	BL
Med	Far	Med	BR	Slow	BR
Med	Far	Med	SR	Fast	SR
Med	Far	Med	S	Fast	S
Med	Far	Med	SL	Fast	SL
Med	Far	Med	BL	Slow	BL
Med	Far	Near	BR	Slow	BL
Med	Far	Near	SR	Slow	BL
Med	Far	Near	S	Fast	S
Med	Far	Near	SL	Slow	SL
Med	Far	Near	BL	Slow	BL
Med	Med	Far	BR	Slow	BR
Med	Med	Far	SR	Med	SR
Med	Med	Far	S	Slow	S
Med	Med	Far	SL	Med	SL
Med	Med	Far	BL	Slow	BL
Med	Med	Med	BR	Slow	BR
Med	Med	Med	SR	Med	SR
Med	Med	Med	S	Med	S
Med	Med	Med	SL	Med	SL
Med	Med	Med	BL	Slow	BL
Med	Med	Near	BR	Slow	BL
Med	Med	Near	SR	Med	SR
Med	Med	Near	S	Med	S
Med	Med	Near	SL	Med	SL
Med	Med	Near	BL	Slow	BL
Med	Near	Far	BR	Slow	BR
Med	Near	Far	SR	Slow	SR
Med	Near	Far	S	Slow	BR
Med	Near	Far	SL	Slow	SL
Med	Near	Far	BL	Slow	BL

Med	Near	Med	BR	Slow	BR
Med	Near	Med	SR	Slow	SR
Med	Near	Med	S	Slow	BR
Med	Near	Med	SL	Slow	SL
Med	Near	Med	BL	Slow	BL
Med	Near	Near	BR	Slow	BL
Med	Near	Near	SR	Slow	BL
Med	Near	Near	S	Slow	BL
Med	Near	Near	SL	Slow	SL
Med	Near	Near	BL	Slow	BL
Near	Far	Far	BR	Slow	BR
Near	Far	Far	SR	Med	SR
Near	Far	Far	S	Fast	S
Near	Far	Far	SL	Slow	BR
Near	Far	Far	BL	Slow	BR
Near	Far	Med	BR	Slow	BR
Near	Far	Med	SR	Med	SR
Near	Far	Med	S	Fast	S
Near	Far	Med	SL	Slow	BR
Near	Far	Med	BL	Slow	BR
Near	Far	Near	BR	Med	S
Near	Far	Near	SR	Med	S
Near	Far	Near	S	Med	S
Near	Far	Near	SL	Med	S
Near	Far	Near	BL	Med	S
Near	Med	Far	BR	Slow	BR
Near	Med	Far	SR	Med	SR
Near	Med	Far	S	Med	S
Near	Med	Far	SL	Slow	BR
Near	Med	Far	BL	Slow	BR
Near	Med	Med	BR	Slow	BR
Near	Med	Med	SR	Med	SR
Near	Med	Med	S	Med	S
Near	Med	Med	SL	Slow	BR
Near	Med	Med	BL	Slow	BR

Near	Med	Near	BR	Slow	BR
Near	Med	Near	SR	Slow	SR
Near	Med	Near	S	Med	S
Near	Med	Near	SL	Slow	SL
Near	Med	Near	BL	Slow	BL
Near	Near	Far	BR	Slow	BR
Near	Near	Far	SR	Slow	SR
Near	Near	Far	S	Slow	BR
Near	Near	Far	SL	Slow	BR
Near	Near	Far	BL	Slow	BR
Near	Near	Med	BR	Slow	BR
Near	Near	Med	SR	Slow	SR
Near	Near	Med	S	Slow	BR
Near	Near	Med	SL	Slow	BR
Near	Near	Med	BL	Slow	BR
Near	Near	Near	BR	Slow	BR
Near	Near	Near	SR	Slow	BR
Near	Near	Near	S	Slow	BL
Near	Near	Near	SL	Slow	BL
Near	Near	Near	BL	Slow	BL

Modeling in MATLAB using Fuzzy Logic Toolbox:

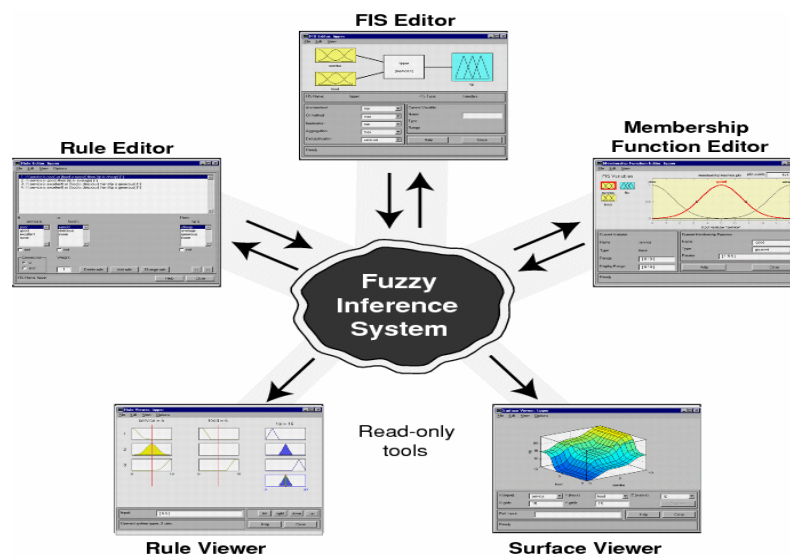


Figure 5 MATLAB Fuzzy Logic Toolbox

In MATLAB fuzzy logic toolbox, five GUI tools are provided namely The FIS Editor, The Membership Function Editor, The Rule Editor, The Rule Viewer & The Surface Viewer which can be used to build, edit, and observe fuzzy inference systems.

1. The Fuzzy Inference System or FIS Editor: The FIS Editor helps us create a fuzzy inference system & define the input & output variables.
2. The Membership Function Editor: The Membership Function Editor helps us define the type of the membership functions associated with each of the input & output variables.
3. The Rule Editor: The Rule Editor helps us create & edit fuzzy rules to define behavior of the system.
4. The Rule Viewer: The Rule Viewer helps us visualize fuzzy inference diagram. This is a read-only tool & used as a diagnostic purposes. It shows the activity status of different rules, & shows the effect of membership function shapes on results.
5. The Surface Viewer: The Surface Viewer generates an output surface map & helps us visualize how one or any of the outputs varies with any one or two of the inputs.

Modeling with Triangular Membership Functions

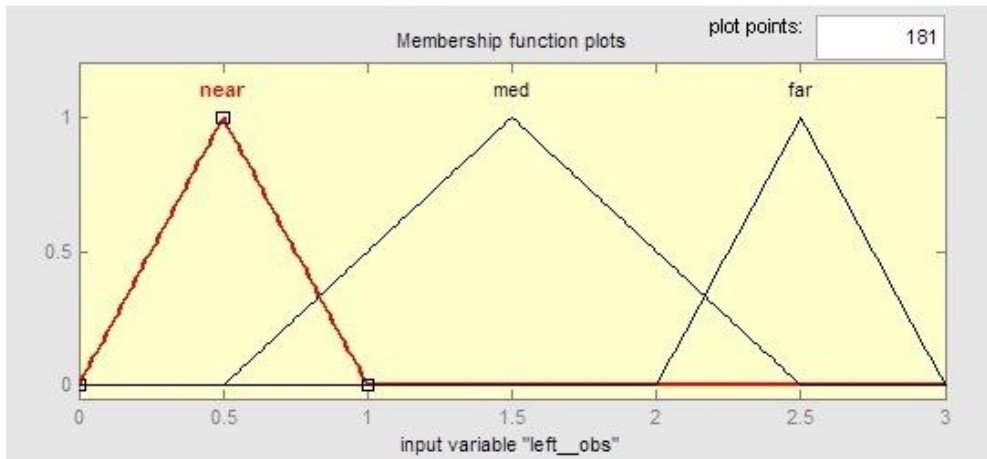


Figure 6 Triangular Membership function plot for left_obs

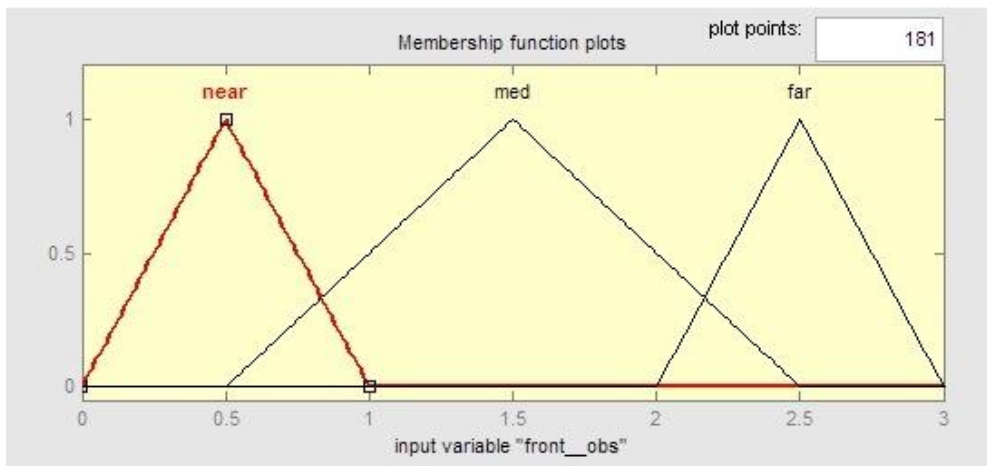


Figure 7 Triangular Membership function plot for front_obs

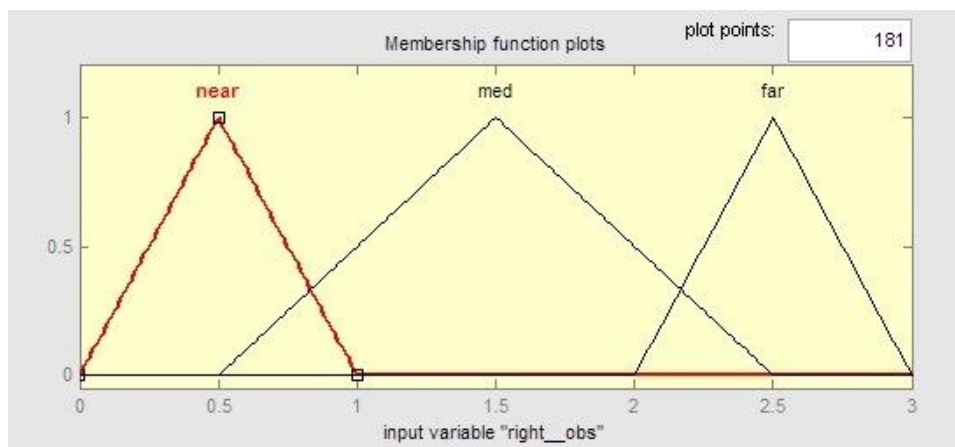


Figure 8 Triangular Membership Function Plot for Right Obstacle Distance

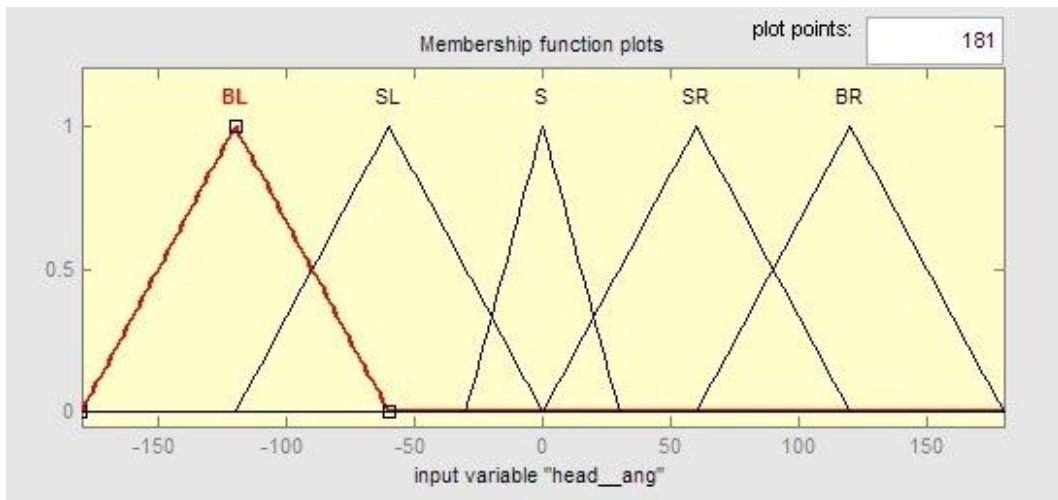


Figure 9 Triangular Membership Function Plot for Heading Angle

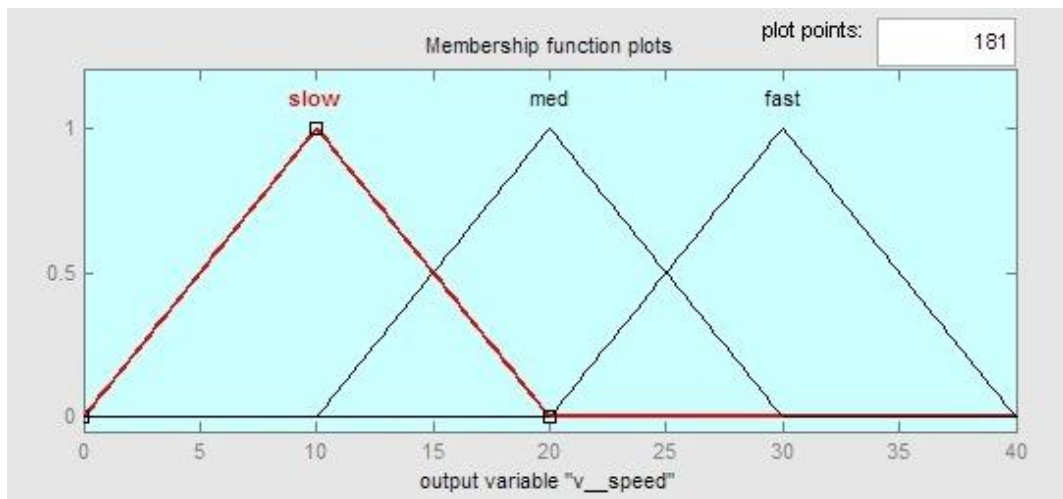


Figure 10 Triangular Membership Function Plot for Vehicle Speed

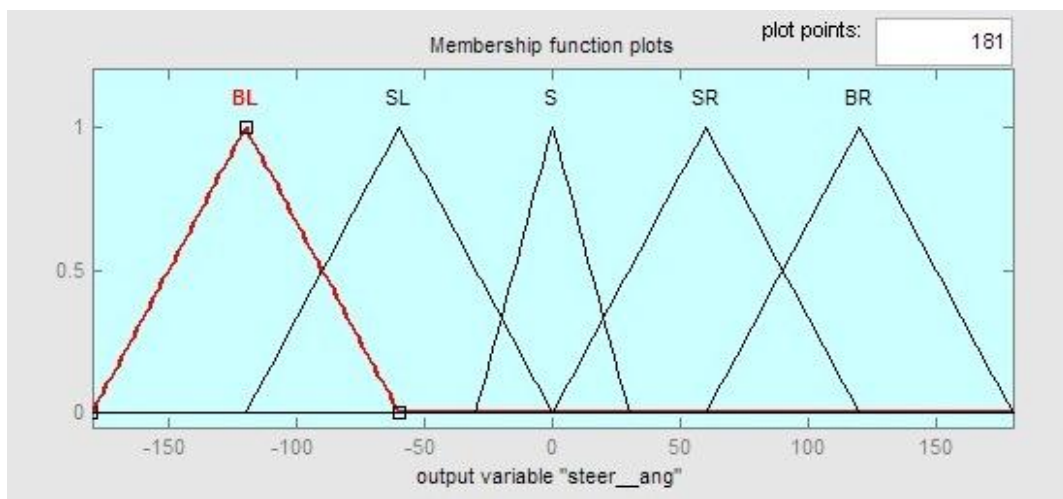


Figure 11 Triangular Membership Function Plot for Steering Angle

Modeling with Trapezoidal Membership Functions

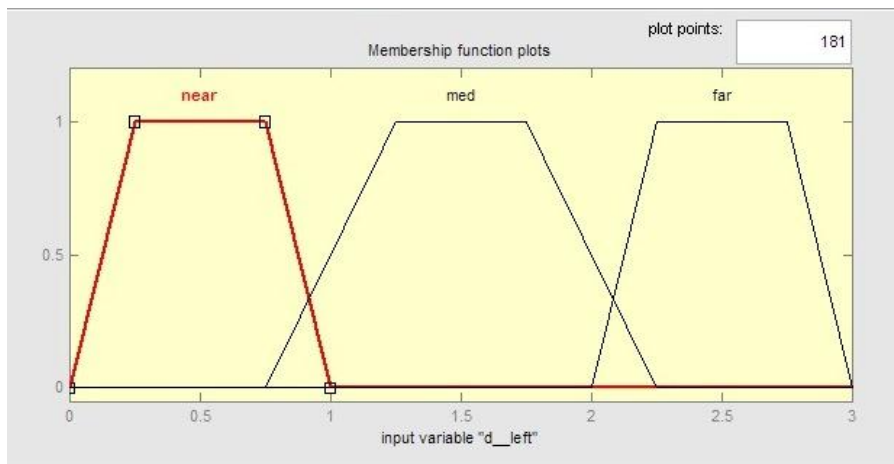


Figure 12 Trapezoidal Membership Function Plot for Left Obstacle Distance

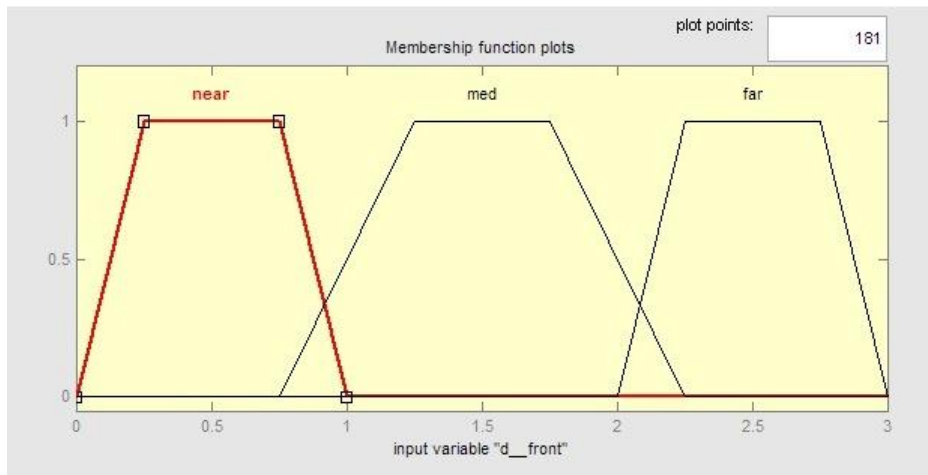


Figure 13 Trapezoidal Membership Function Plot for Front Obstacle Distance

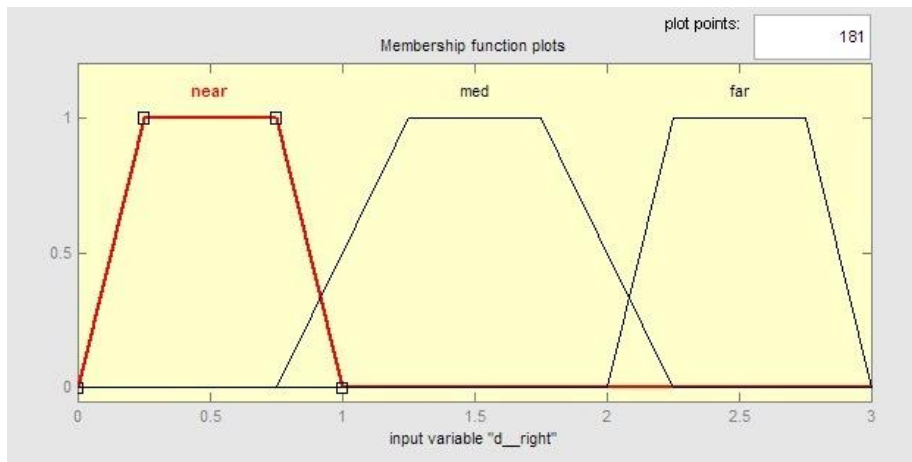


Figure 14 Trapezoidal Membership Function Plot for Right Obstacle Distance

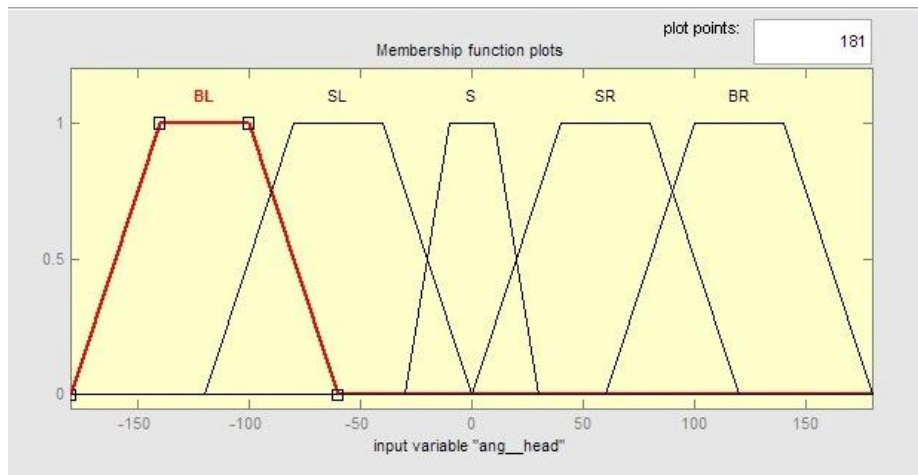


Figure 15 Trapezoidal Membership Function Plot for Heading Angle

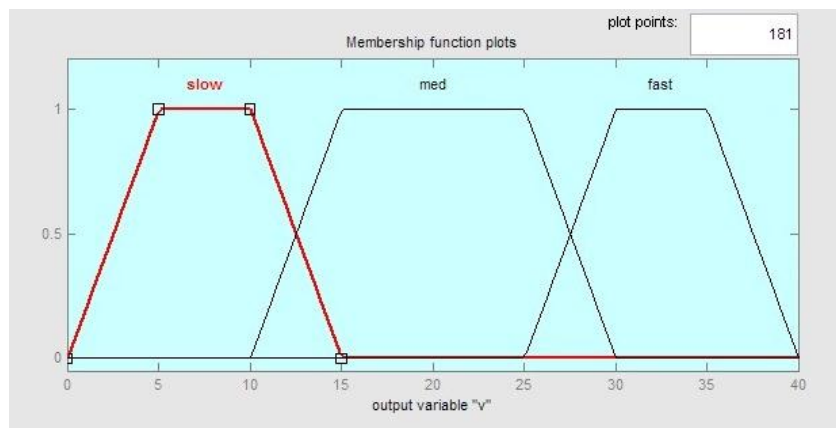


Figure 16 Trapezoidal Membership Function Plot for Vehicle Speed

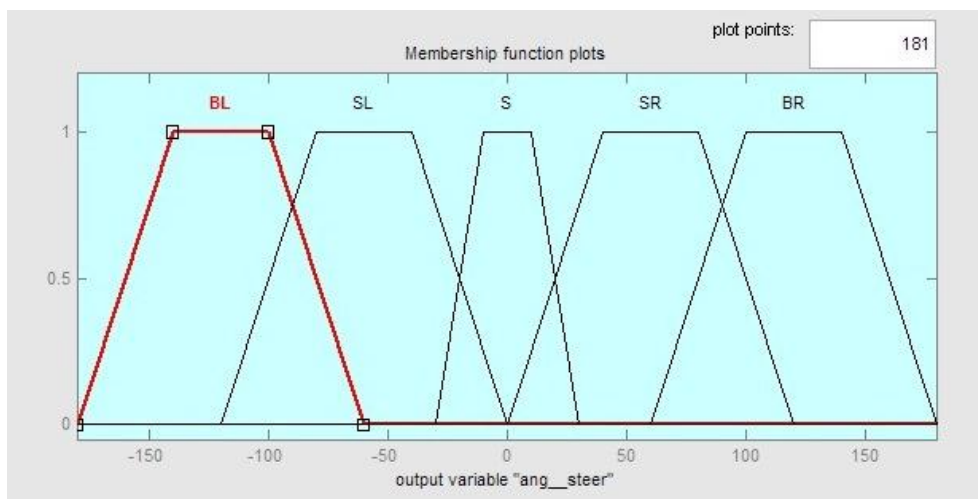


Figure 17 Trapezoidal Membership Function Plot for Steering Angle

Modeling with Gaussian Membership Functions

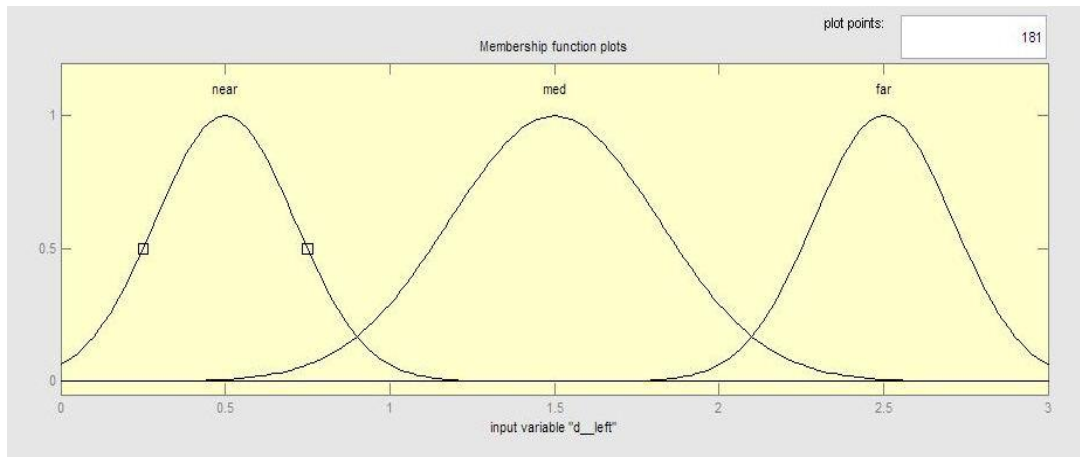


Figure 18 Gaussian Membership Function Plot for Left Obstacle Distance

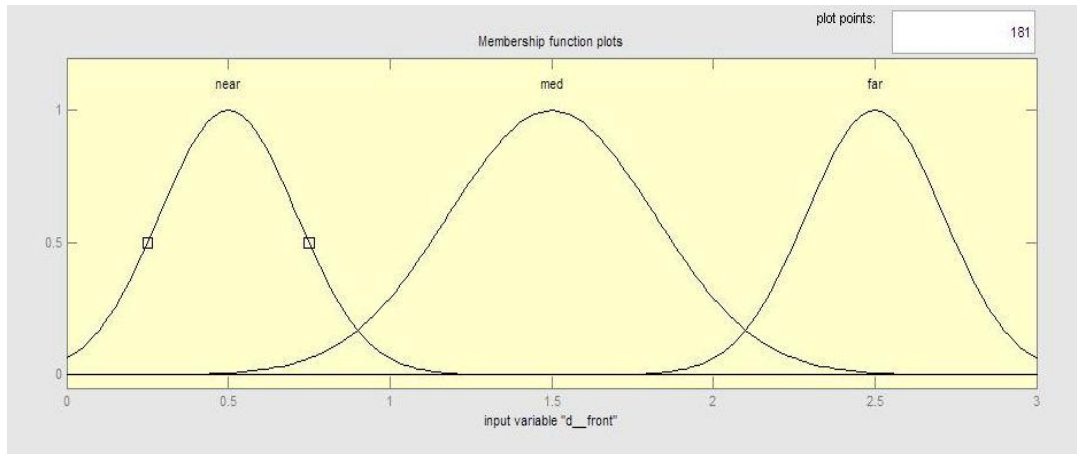


Figure 19 Gaussian Membership Function Plot for Front Obstacle Distance

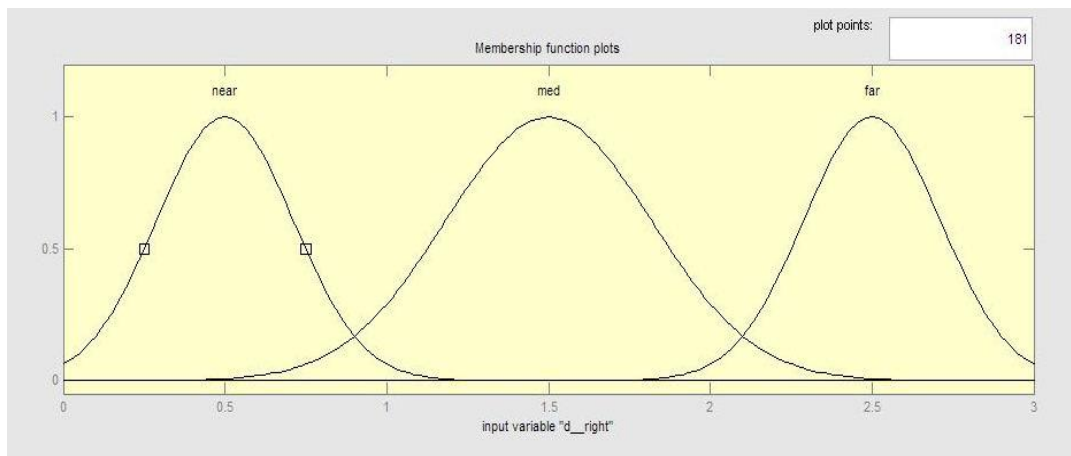


Figure 20 Gaussian Membership Function Plot for Right Obstacle Distance

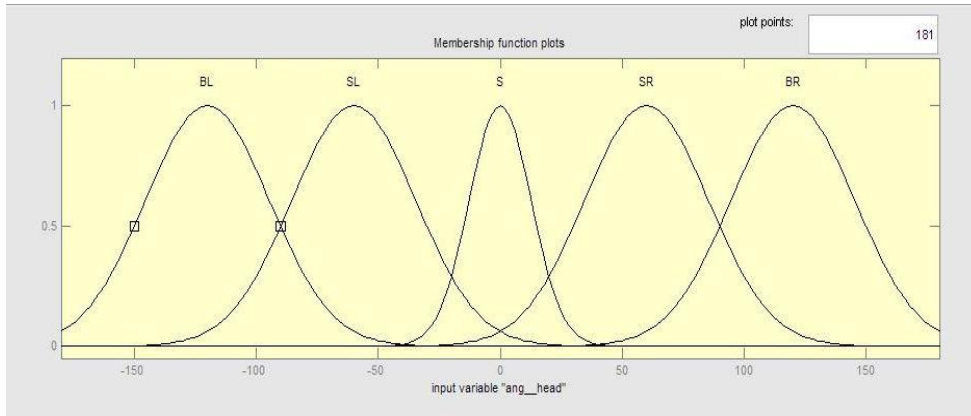


Figure 21 Gaussian Membership Function Plot for Heading Angle

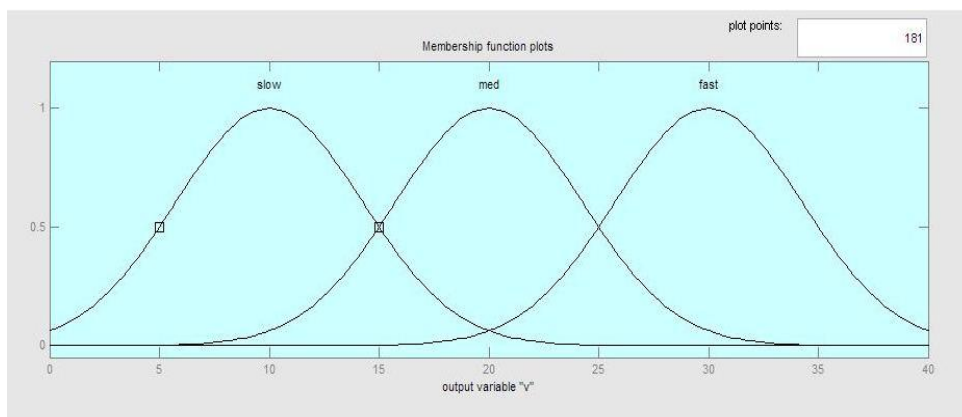


Figure 22 Gaussian Membership Function Plot for Vehicle Speed

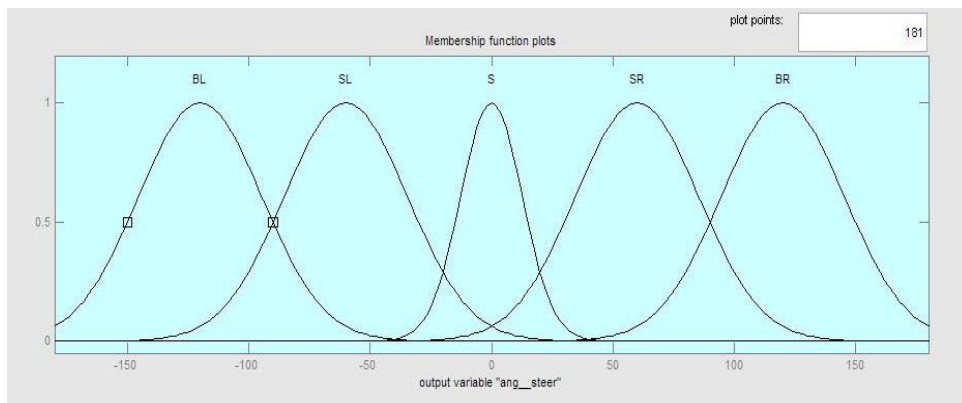


Figure 23 Gaussian Membership Function Plot for Steering Angle

4.2 System Modelling using Neural Network Technique

For modelling in neural network, we use the front obstacle distance (d_{front}), left obstacle distance (d_{left}), right obstacle distance (d_{right}) & heading angle (ang_{head}) as input parameters to the ANN. The target parameters are vehicle speed (v) & steering angle (ang_{steer}). We set the input parameters in a matrix [input] and target in matrix [target]. Once the input and the target matrix are set, we use the nntool algorithm to train and simulate the network.

Steps involved in setting up the Neural Network:

1. The network manager window was opened with the command nntool.
2. The input matrix [input] and the target matrix [target] were imported into the network manager.
3. A new network was created with the inputs and outputs of the following type:

Network type: Feed-forward backprop

Training Function: TRAINGD

Adaptation Learning Function: LEARNGDM

Performance Function: MSE

9. The number of layers was varied to get the best result (least mean square error) possible
10. The network was trained for 1000 to 10,000 epochs to the error reached our pre designated threshold.
11. The performance graph was plotted.

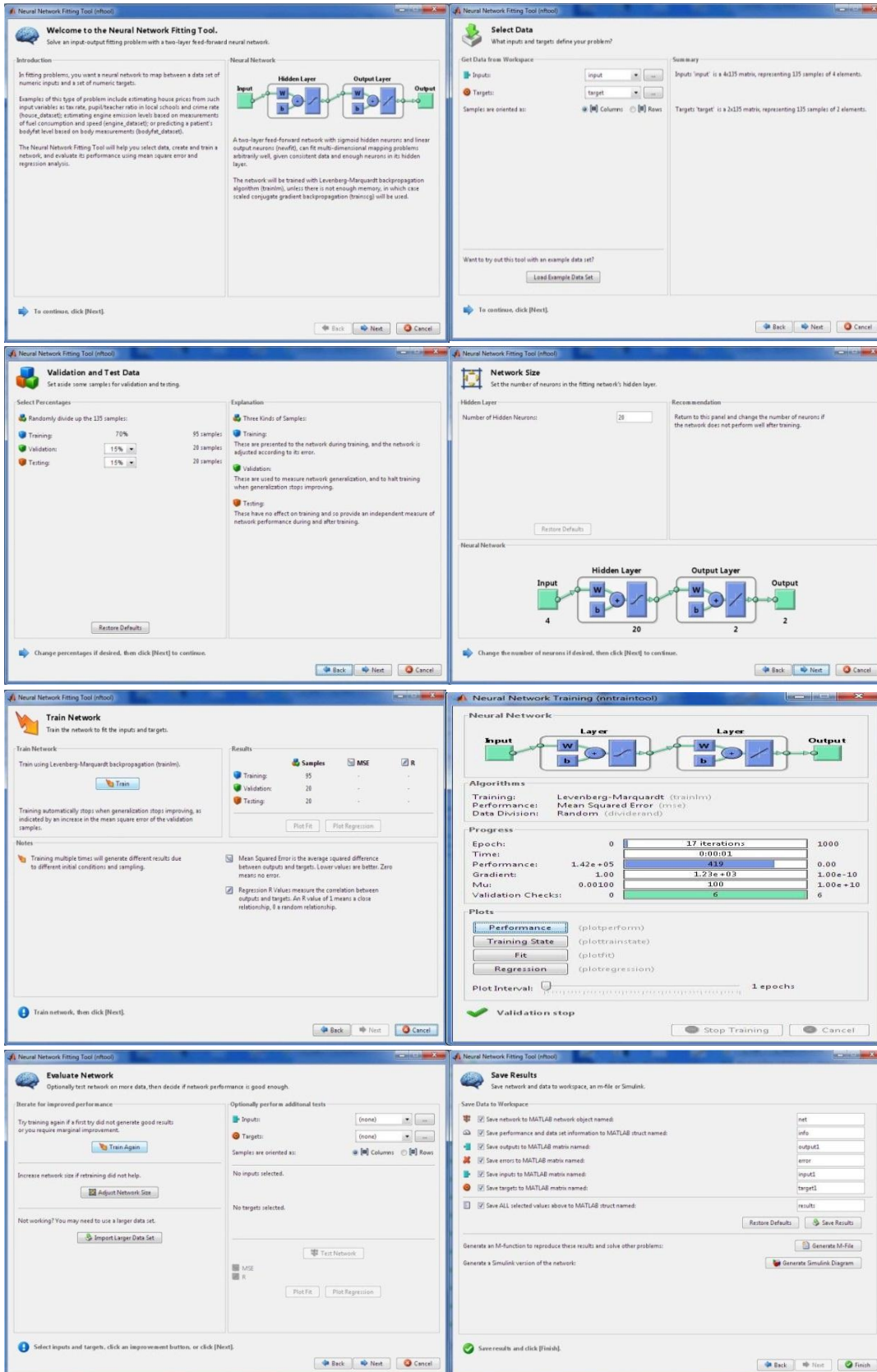


Figure 24 Modelling in MATLAB with Neural Network

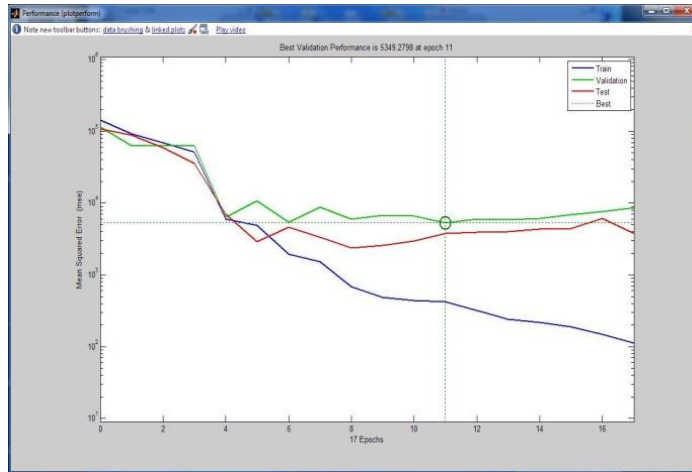


Figure 25 Performance Plot

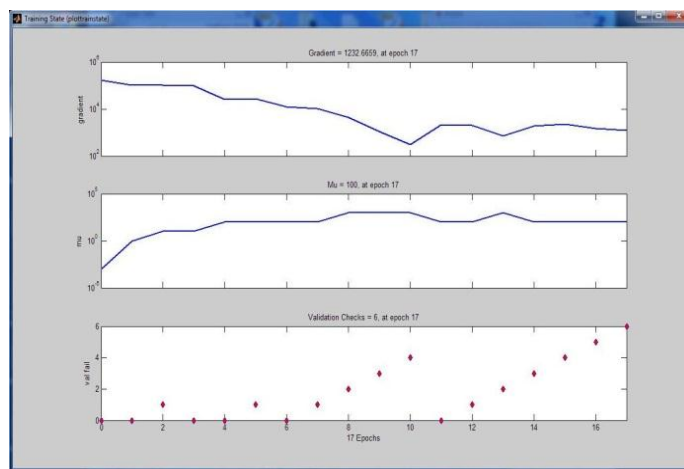


Figure 26 Training State Plot

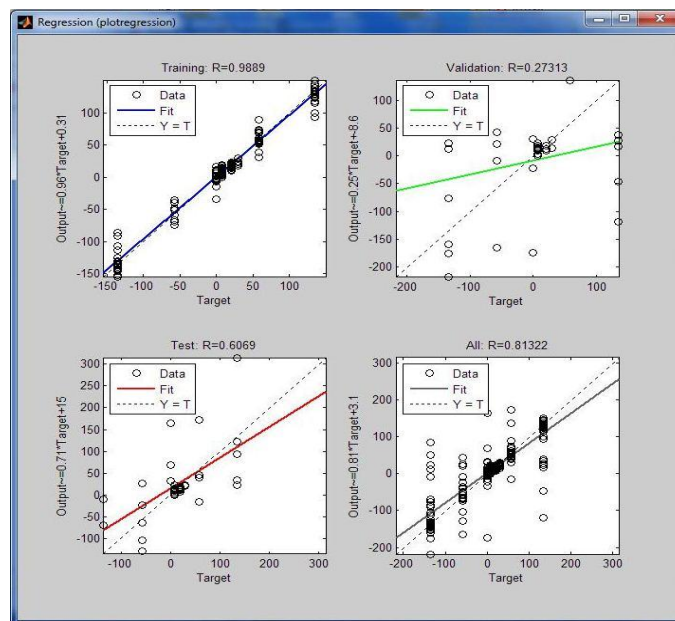


Figure 27 Regression Plot

CHAPTER 5

Experimental Model

Experimental Model



Figure 28 REVA i, the Commercial Vehicle undertaken for the project



Figure 29 Central Pinion attached to Steering of AGV. Diameter of Pitch Circle=32mm



Figure 30 Pinion (spur Gear) attached to the Servo motor for rotating the Central Pinion attached to the Steering of Vehicle. Diameter of pitch circle = 45mm



Figure 31 Dummy setup (not affixed to steering handle) showing driving gears mating with driven.

The servo motors used have the following specification:

- Quantity used: 2
- Servo Motor model: Hitec HS-805BB
- Torque: at 6 Volts; 24.7 kg/cm

To each of the servo motors a spur gear was attached with the help of an arm. Over the steering wheel a central rectangular shaft was attached. The driven gear is to be attached to this shaft. The driven gear is a central gear which meshes with the driving gears in a straight line. Additionally, the centre of the driven gear coincides with the centre of the steering wheel. Also a frame was built by arc welding of mild steel plates. The view shown for the frame with the servo motors is that which is seen from the steering towards the front seat of the vehicle. All the dimensions are as per the particular application required for the commercial vehicle chosen.



Figure 32 Shaft carrying the Central Pinion (to be attached to Steering wheel)

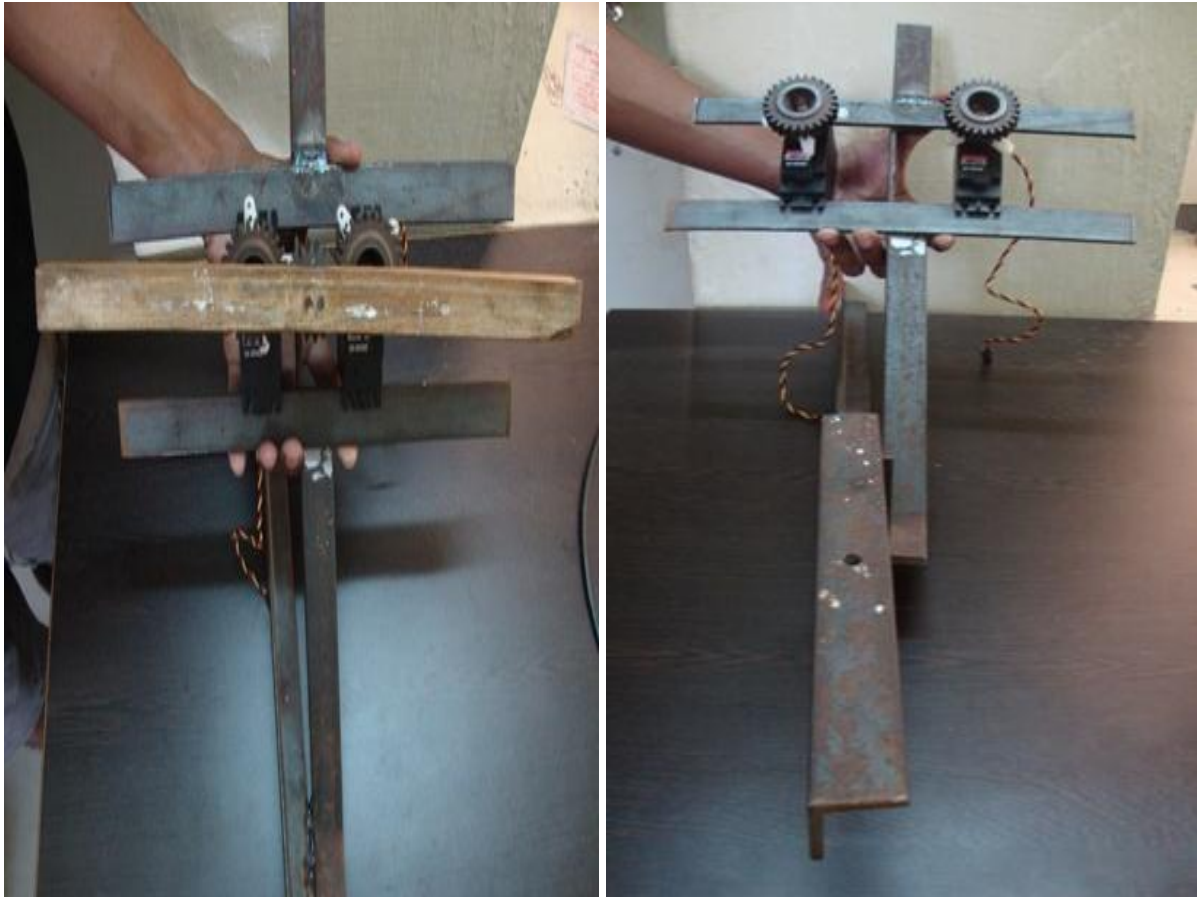


Figure 33 Steering Mechanism

For speed control a mechanism was designed consisting of control over the accelerator and brake of the vehicle. A servo motor each is used to actuate the accelerator and brake pedals. Similarly, for the steering another mechanism was designed consisting of a central spur gear attached to the steering wheel of the AGV. This spur gear is driven by a pair of driving gears which mesh on either side of it on a straight line and is powered by the servo motors (Hi-tec HS-805BB (24.7 Kg-cm torque)). For mounting on the vehicle, a steel frame has been fabricated which is secured to the floor of the AGV platform. The servo motors and driving gears are mounted onto this frame. While designing the driving and the driven gears a gear ratio of 3 was obtained to achieve gear reduction which gives better control and larger rotation of steering wheel via smaller rotation of the servo motors.

CHAPTER 6

Results & Discussion

Results and Discussion

Following the theoretical analysis of Fuzzy Inference Systems and Neural Networks; corresponding navigation mechanism for a mobile robot or AGV has been developed.

Based on the nature of the membership functions—Gaussian functions, Triangular functions and Trapezoidal functions; the Fuzzy Inference System provided three separate decision making systems. The Feed forward back propagation algorithm under the neural network paradigm was the fourth decision making system developed.

Each of the systems had four inputs: Left Distance (Distance between the AGV and the nearest obstacle to its left); Right Distance (Distance between the AGV and the nearest obstacle to its right); Front Distance (Distance between the AGV and the nearest obstacle ahead of it) and Heading Angle (position of the AGV in relation with the target destination expressed as an angle). Two outputs were generated for navigation by each of the systems: Velocity (Speed at which the AGV should travel) and Steering Angle (The angle to which the steering wheel of the AGV must be turned)

Experimental data sets were created for the specific application of the AGV available for the research. Each of the four decision making systems that were developed were successfully set-up modeled in MATLAB.

To evaluate the performance of each of these decision making systems, simulations were conducted for each decision making system, maintaining a constant set of inputs all over.

Inputs	Outputs							
	Fuzzy Inference System						Neural Network	
	Gaussian		Triangular		Trapezoidal		Feed forward Back propagation	
	Velocity	Steering angle	Velocity	Steering Angle	Velocity	Steering Angle	Velocity	Steering Angle
d_left = 2.6 d_front = 1.2 d_right = 0.8 ang_head = -38	18.20	-58.38	15.69	-60.03	18.93	-60.02	13.93	-65.47
d_left = 2.6 d_front = 2.4 d_right = 1.3 ang_head = 5	28.20	9.13	26.66	14.99	30.18	15.92	35.02	20.26
d_left = 1.4 d_front = 1.8 d_right = 2.9 ang_head = 38	19.75	59.47	20.00	60.12	20.00	60.07	15.81	54.33

Table 3 Comparison of Outputs for a sample set of Inputs

After the required training and learning is completed, it is expected that these decision making systems can by themselves without any human intervention guide a mobile robot or AGV in unknown and unstructured environment. This is supported by the accurate and precise outputs generated by the decision making systems. The outputs that are obtained are in agreement with the behavior initially desired. This desired behavior had been decided by a human for the particular situation. Thus the systems should be capable of performing autonomous navigation replacing a human and rendering human intervention unnecessary. However, it was noted that the neural network implemented was modeled with a high mean square error and this approach gave outputs that considerably deviated from the rest.

The hardware for the practical implementation of the above discussed autonomous navigation has been designed. The design incorporates a frame which is secured to the car floor and onto which the gears and servo motors are mounted.

CHAPTER 7

Conclusion and Scope for Future Work

Conclusion and Scope for Future Work

A navigation system has been developed which can be fed with information gathered from the environment using an array of sensors. Ultrasonic sensors are to provide the distance of an obstacle from the AGV whereas a GPS device and heading compass are to compute the orientation of the AGV with respect to the target destination (heading angle). The output of the proposed navigation system is the steering angle and velocity of the AGV. These outputs are then used as stimulus for producing the required mechanical behavior through the system hardware. The hardware consisting of a mounting frame, driving and driven gears and servo motors was constructed and installed in the commercial vehicle REVAi.

Firstly, the neural network approach can be further optimized to reduce the mean square error in the training. There exist a vast number of algorithms for machine learning which can be explored. Additionally, the ultrasonic sensor data collection system can be replaced by a visual sensors or stereoscopic cameras. The data collected from cameras would also require similar processing. Image processing techniques and algorithms through artificial neural networks, neuro-fuzzy systems and genetic algorithms are an area of high interest. Developments in these fields can be incorporated with those in the autonomous decision making systems for further improvements in mobile robot navigation.

References

1. Liu K. & Lewis F.L.; “Fuzzy Logic based Navigation Controller For An Autonomous Mobile Robot; IEEE, Vol. 7, pp.49-63,1989.
2. Simon Dan & Peri Vamsi Mohan; “Fuzzy Logic Control For An Autonomous Robot”; IEEE Competition, Vol. 6, No.4, pp.445-454, 1990.
3. Harisha S.K., Kumar Ramkanth P., Sharma S.C.; “Fuzzy Logic Reasoning To Control Mobile Robot On Pre-Defined Strip Path”; Proceedings of World Academy of Science, Engineering and Technology, Vol. 4, pp.539-546,2003.
4. Castillo Oscar, Aguilar Luis T., Cardinas Selene; “Fuzzy Logic Tracking Control For Unicycle Mobile Robots”; Engineering Letters, pp. 73-82, 2003.
5. Fatmi Anis, Yahmadi Amur Al, Khriji Lazhar, Masmoudi Nouri; “A Fuzzy Logic Based Navigation of a Mobile Robot”; Proceedings of World Academy of Science, Engineering and Technology, pp. 159-177, 1995.
6. Malhotra Rajiv, Sarkar Atri; “Development of a Fuzzy Logic Based Mobile Robot for Dynamic Obstacle Avoidance and Goal Acquisition in an Unstructured Environment”; Proceedings of IEEE/ASME, International Conference on Advanced Intelligent Mechatronics, pp.235-247, 2003.
7. Singh Harmeet, Arora Sanchit; “Using Fuzzy Logic for Mobile Robot Control”; IIT-D manuals, India, 2006.
8. Ramos Nelson, Lima Pedro U., Sousa Joao M.C.; “Robot Behaviour Coordination Based On Fuzzy Decision-Making”; GCAR, IDMEC, Portugal, pp. 145-172.

9. Busquets D., Sierra C., Mantaras R.Lopez De; "A Multi-Agent Approach to Fuzzy Landmark-Based Navigation"; Multi-Valued Logic & Soft Computing, pp.971-978, 1995.
10. Figueiredo K., Vellasco M., Pacheco M.A.; "Mobile robot control using intelligent agents"; Second international workshop on Intelligent systems design and application, pp.159-173, 1999.
11. Li Wei, Chen Zushun, Ma Chenyu, Fuzzy Logic based Behaviour Fusion for Navigation of an Intelligent Mobile Robot, Journal of Computer Science & Technology, pp. 125-132, 1999.
12. Li wei, Fuzzy Logic Based Robot Navigation In Uncertain Environment By Multisensor Integration; IEEE Proceeding of the 1994 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI'94); Las Vegas, Nevada; Oct 1994.
13. Tso S.K., Fung Y.H. & Cheung Y.P.; "Fuzzy-logic Control for Differential-wheel-drive AGVs Using Linear Opto-sensor Arrays"; Proceedings of the 1996 IEEE International Conference on Robotics and Automation, Minneapolis, Minnesota - April 1996.
14. Kumile Christopher M. and Tlale Nkgatho S.; "Intelligent Distributed Fuzzy Logic Control System (IDFLCS) of a Mecanum Wheeled Autonomous Guided Vehicle"; Proceedings of the IEEE International Conference on Mechatronics & Automation, Niagara Falls, Canada· July 2005.
15. Huang Min and Huang Dagui; "Embedded Fuzzy Logic Control of AGV in Path Tracking"; Proceedings of the 2010 IEEE International Conference on Mechatronics and Automation, August 4-7, 2010, Xi'an, China.
16. Wijesoma W. S., Kodagoda K.R.S. and Teoh E.K.; "Un-coupled Fuzzy Controller for Longitudinal & Lateral control of a Golf Car-like AGV".

17. Butdee S., Suebsomran A, Vignat F, Yarlagadda P.K.D.V; “Control and Path Prediction of an Automated Guided Vehicle”; Journal of Achievements in Materials & Manufacturing Engineering, Vol. 31, Issue 2, Dec. 2008.
18. Wijesoma W. S., Khaw Ping Ping, and, Teoh Eam Khwang; “Control and Navigation of an Outdoor AGV using Fuzzy Reasoning”.
19. Khaw Ping Ping, Wijesoma W. S., and, Teoh Eam Khwang, “Intelligent Control And Navigation of An Outdoor AGV”.
20. Quan YUAN, Yunzhou ZHANG, Hao W U, He WANG, “Application of Fuzzy Algorithm In the Direction of Intelligent Car”.
21. Ayrulu Birsal, Barshan Billur; “Neural networks for improved target differentiation and localization with sonar”; Neural Networks 14 (2001) 355-373.
22. Scarselli Franco, Tsoi Ah Chung; “Universal Approximation Using Feedforward Neural Networks: A Survey of Some Existing Methods, and Some New Results”; Neural Networks, Vol. 11, No. 1, pp. 15–37, 1998.
23. Soylu Mustafa, Ozdemirel Nur E., Kayaligil Sinan; “Theory and Methodology A self-organizing neural network approach for the single AGV routing problem”; European Journal of Operational Research 121 (2000) 124-137.
24. Orille A. L., Sowilam G.M.; “An Application Of Neural Networks For Direct Torque Control”; Computers & Industrial Engineering 37 (1999) 391-394.
25. Yang Simon X., Meng Max; “An efficient neural network approach to dynamic robot motion planning”; Neural Networks 13 (2000) 143–148.

26. Barberá H. Martínez, Skarmeta A. Gómez, Izquierdo M. Zamora, Blaya J. Botía; "Neural Networks for Sonar and Infrared Sensors Fusion"; In Proceedings of the Third International Conference on Information Fusion, 2000. FUSION 2000.
27. Chronis George, Skubic Marjorie; "Experiments in Programming by Demonstration: Training a Neural Network for Navigation Behaviors"; Dept. of Computer Engineering and Computer Science,; University of Missouri-Columbia.
28. Davis Ian Lane; "A Modular Neural Network Approach to Autonomous Navigation"; The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA; April 1996.
29. Pomerleau Dean A.; "Knowledge-based Training of Artificial Neural Networks for Autonomous Robot Driving".
30. Glasius Roy, Komoda Andrzej, Gielen Stan C. A. M.; "Neural Network Dynamics for Path Planning and Obstacle Avoidance".
31. MATLAB tutorials.
32. Pradhan Saroj Kumar, Parhi Dayal Ramakrushna and Panda Anup Kumar; "Navigation of Multiple Mobile Robots using Rule-based-Neuro-Fuzzy Technique"; International Journal on Computational Intelligence Volume 3 Number 2.
33. Singh Mukesh Kumar, Parhi Dayal R.; "Intelligent Neuro-Controller for Navigation of Mobile Robot".
34. Nandiganahalli Suraj; "Mobile Platform Control Using Fuzzy-Logic and Webots".
35. Pradhan Saroj Kumar; "Navigation Techniques for Control of Multiple Mobile Robots".

36. Stergiou Christos and Siganos Dimitrios; Neural Networks;

http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html

37. Rodríguez Francisco J., Neural Network Tutorial;

<http://www.gc.ssr.upm.es/inves/neural/ann1/anntutor.htm>