# CONTROL OF REAL MOBILE ROBOT USING ARTIFICIAL INTELLIGENCE TECHNIQUE

***Shubhasri Kundu***

# Control of Real Mobile Robot Using Artificial Intelligence Technique

*Thesis Submitted to the*

*Department of Mechanical Engineering*

*National Institute of Technology, Rourkela*

*For award of the degree*

*of*

**Master of Technology (Research)**

*by*

**Shubhasri Kundu**

*Under the Supervision of*

**Prof. Dayal R. Parhi**

**Department of Mechanical Engineering**

**National Institute of Technology Rourkela**

**Orissa (India)-769008**

# Declaration

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgement has been made in the text.

(*Shubhasri Kundu*)

Date:

NATIONAL INSTITUE OF TECHNOLOGY
ROURKELA -769008, ORISSA, INDIA.

## Certificate

*This is to certify that the thesis entitled,* ***"Control of Real Mobile Robot using Artificial Intelligence Techniques",*** *being submitted by Ms. Shubhasri Kundu to the Department of Mechanical Engineering, National Institute of Technology, Rourkela, for the partial fulfillment of award of the degree Master of Technology (Research), is a record of bona fide research work carried out by him under my supervision and guidance.*

*This thesis in my opinion, is worthy of consideration for award of the degree of Master of Technology (Research) in accordance with the regulation of the institute. To the best of my knowledge, the results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.*

**Supervisor**

Date: / /2011

(Dr. Dayal R. Parhi)
Professor,
Department of
Mechanical Engineering
National Institute of Technology
Rourkela, Orissa, India- 769008.

---

# Acknowledgements

My first thanks are to the Almighty God, without whose blessings, I wouldn't have been writing this “acknowledgments".

I would like to extend my heartfelt indebtedness and gratitude to Prof. Dayal R. Parhi for his kindness in providing me an opportunity to work under his supervision and guidance. During this period, without his endless efforts, immense knowledge, deep patience, invaluable guidance and answers to my numerous questions, this research would have never been possible. I am especially obliged to him for teaching me both research and writing skills, which have been proven beneficial for my current research and future career. He showed me different ways to approach a research problem and the need to be persistent to accomplish any goal. It has been a great honour and pleasure for me to do research under the supervision of Dr. Dayal R. Parhi.

I am thankful to Prof. Sunil Kumar Sarangi, Director of National Institute of Technology, for giving me an opportunity to work under the supervision of Prof. Parhi. Special thank goes to Prof. R.K. Sahoo, Head of the Department, Department of Mechanical Engineering, without his support and cooperative attitude it was not possible to reach towards a success in research work.

I extend my sincere thanks to Prof. A. K. Panda and Prof. B.D. Subudhi of Electrical Engineering Department for their kind help during course work as well as research work by providing necessary resources.

I thank all the members of the Department of Mechanical Engineering, and the Institute, who helped me in various ways towards the completion of my work.

I would like to thank all my friends and lab-mates for their encouragement and understanding. Their help and lots of lovely memory with them can never be captured in words.

Finally, I thank my parents and my entire family members for their unlimited support and strength. Without their dedication and dependability, I could not have pursued my M. Tech(R) degree at the National Institute of Technology Rourkela.

Shubhasri Kundu

# Abstract

An eventual objective of mobile robotics research is to bestow the robot with high cerebral skill, of which navigation in an unfamiliar environment can be succeeded by using on-line sensory information, which is essentially starved of humanoid intermediation. This research emphases on mechanical design of real mobile robot, its kinematic & dynamic model analysis and selection of AI technique based on perception, cognition, sensor fusion, path scheduling and analysis, which has to be implemented in robot for achieving integration of different preliminary robotic behaviors (e.g. obstacle avoidance, wall and edge following, escaping dead end and target seeking). Navigational paths as well as time taken during navigation by the mobile robot can be expressed as an optimization problem and thus can be analyzed and solved using AI techniques. The optimization of path as well as time taken is based on the kinematic stability and the intelligence of the robot controller. A set of linguistic fuzzy rules are developed to implement expert knowledge under various situations. Both of Mamdani and Takagi-Sugeno fuzzy model are employed in control algorithm for experimental purpose. Neural network has also been used to enhance and optimize the outcome of controller, e.g. by introducing a learning ability. The cohesive framework combining both fuzzy inference system and neural network enabled mobile robot to generate reasonable trajectories towards the target. An authenticity checking has been done by performing simulation as well as experimental results which showed that the mobile robot is capable of avoiding stationary obstacles, escaping traps, and reaching the goal efficiently.

***Keywords:*** Mobile Robot, Navigational Strategy, Reactive behavior, Fuzzy Logic, Fuzzy-Neural Network etc.

# Bio-Data


1. Name : Shubhasri Kundu
2. Date of Birth : 25.08.1985


3. Educational Qualification:

| Examination | Name of the Institute | Board/University | Year of passing | Division | Subject |
|---|---|---|---|---|---|
| **Madhyamik** | F.B.P.H.S.School | W.B.B.S.E | 2001 | 1st | All |
| **H.S.E** | F.B.P.H.S.School | W.B.C.H.S.E | 2003 | 1st | Science |
| **B.Tech** | Jalpaiguri Govt. Engg. College | W.B.U.T | 2007 | 1st | Electrical Engineering |
| **M.Tech (Research)** | N.I.T, Rourkela | N.I.T, Rourkela | continuing | - | Robotics |

4. Research Experience: Engaged in a Project entitled "Navigation of Multiple Robots using Artificial Intelligence Technique" sponsored by Board of Research in Nuclear Sciences, BARC as a Junior Research Fellow for 2 yrs. under the guidance of Prof. Dayal R.Parhi, Mechanical Engineering Deptt., NIT Rourkela.


5. Permanent Address: c/o H.N.Kundu, "Shymkunja", Siddheswaritala, P.O:Ranaghat,
   Dist: Nadia, Pin:741201, West Bengal.
6. Mobile No : 9800257472
7. E-mail ID : shubhasri_ee2006@yahoo.co.in

# Table of Contents

# List of Tables

# List of Figures

# List of Symbols

| | |
|---|---|
| f (.) | = Activation function |
| $\theta_{actual}$ | = Actual output of neural network |
| $\alpha$ | = Angle between local coordinate x-axis to robot reference frame |
| $\beta$ | = Angle of the wheel plane relative to the chassis or wheel orientation |
| $\theta$ | = Angular difference between the global and local reference frames |
| $\omega_1, \omega_2$ | = Angular velocity of two wheels |
| $\omega$ | = Angular tangential velocity at point C |
| G | = Centre of gravity point of the mobile robot |
| $\dot{q}$ | = Combination of Linear and Angular Velocities in Matrix form at point G |
| $J_{2f}$ | = Constant diagonal matrix $N_f$ X $N_f$ of all standard wheels radii |
| $\theta_{desired}$ | = Desired output of neural network |
| $l$ | = Distance of wheel from point P |
| d | = Distance between G and C |
| $\tau_{dR}$ | = Dynamic torque of Right-side motor |
| $\tau_{dL}$ | = Dynamic torque of Left-side motor |
| $\delta$ | = Error gradient |
| FD | =Front obstacle distance |
| $Y_3^{\{1\}}$ | = Front obstacle distance from the robot |
| $\mu$ | = Fuzzy Membership Function |
| $q_g$ | = Generalized coordinate at point G |
| $q_c$ | = Generalized coordinate at point C |
| HA | =Heading angle |

| | |
|---|---|
| lay | = Layer number |
| LD | =Left obstacle distance |
| $Y_1^{\{1\}}$ | = Left obstacle distance from the robot |
| $Y_6^{\{1\}}$ | = Left Wheel Velocity |
| $v_c$ | = Linear tangential velocity at point C |
| $vg$ | = Linear tangential velocity at point G |
| $v_R$ | = Linear velocity of right wheel |
| $v_L$ | = Linear velocity of left wheel |
| $J_{1f}$ | =Matrix ($N_f$x3) for all fixed standard wheels to their motions along their individual wheel planes |
| $C_{1f}$ | = Matrix ($N_f$x 3) which contains all sliding constraints of wheels |
| C | = Middle point of rear axle of the mobile robot |
| more neg | = More negative |
| more pos | = More positive |
| neg | = Negative |
| $x_c, y_c$ | =Positional Co-ordinates of robot in global reference frame |
| P | = Position in polar coordinates (Centred between two drive wheels) |
| pos | = Positive |
| r | = Radius of the wheel |
| RD | =Right obstacle distance |
| $Y_2^{\{1\}}$ | = Right obstacle distance from the robot |
| $Y_5^{\{1\}}$ | = Right Wheel Velocity |
| $\dot{\xi}_I$ | = Robot motion in global reference frame |
| $\dot{\xi}_R$ | = Robot motion in local reference frame |
| $\xi_I$ | = Robot Position with respect to inertial frame |
| $\xi_R$ | = Robot Position with respect to local reference frame |

| | |
|---|---|
| $\dot{\phi}_1$, $\dot{\phi}_2$ | = Spinning speed of each wheel |
| M(q) | = Symmetric, positive definite inertia matrix |
| $Y_4^{\{1\}}$ | = Target bearing |
| $N_f$ | = Total No. of fixed standard wheels |
| LV | =Velocity of left wheel |
| RV | =Velocity of right wheel |
| $W_{ji}^{\{lay\}}$ | = Weight of the connection from neuron i in layer ‘lay-1’ to neuron j in layer ‘lay’ |

# 1 Introduction

An innovative exercise, for enabling mobile robot to be explored safely in congested real world surroundings, especially, impulsively fluctuating environment and also avoiding structured or unstructured obstacles, has been conveyed in this thesis. This chapter stipulates background information and motivation pertaining to the work carried out in this thesis. It then briefly enlightens the overview of major goals of this research i.e. what type of demanding problems have been undertaken and how, which are reaffirmed later in more depth in the successive thesis chapters. Finally, thesis structure is sketched preciously.

## 1.1 Background and Motivation:

From the most primitive to the latest surmise, regarding the formation of autonomous mobile robot, it was acknowledged that irrespective of the mechanisms used to precede the robot or the means used to sense the environment; the computational principles i.e. control algorithms that govern the robot are of dominant significance. Efficient control of a robot may lead to substantial variations in the robot's inclusive behavior or action. To behave in large scale surroundings, Mobile robot is not only an assortment of algorithms for sensing real time response, augmenting possession of knowledge, rationalizing the positional error and moving about space; physical incarnations of these algorithms and ideas, which are able to conduct all of whims of the real world, are also entailed to be coupled. As such mobile robot provides an authenticity check for hypothetical concepts and algorithms.

An accurately perceptive robot needs to be able to deal with tentative, equivocal, inconsistent and noisy data by learning through its own interface with the world while achieving goal. Mechanisms, used in successful navigation of robotic agent, embrace a number of skills: from high-level capabilities such as surveying the surrounding environment, building an autonomous global map and planning a path towards an explicit goal, to the execution of rudimentary low level action like avoiding collisions with obstacles. So, over last eras, a strong motivation has been divulged to turn out self-ruling intelligent robots that are especially well-suited for tasks that reveal the subsequent features:

- An uncongenial remote environment into which sending a human being would be either very costly or very dangerous or in an utmost instance when territories are completely inaccessible to humans such as microscopic environment.

- In case of a task with a very demanding duty cycle or a very high fatigue factor.

To intermingle with the environs, animals antedate the result of their actions and envisage the behavior of other objects too. So, there is a strong contention for investigating intelligent behavior by means of positioned agents or mobile robots. Perception and action are necessitated to be tightly coupled in a closed loop to spawn navigational strategy of mobile agents. This awareness reverses the inclination of mobile robotics field towards an inherently interdisciplinary research area involving the followings:

- Mechanical Engineering for configuring particular locomotive mechanisms;
- Computer Science for representations, sensing and planning algorithms;
- Electrical Engineering for system integration, sensors and communications;
- Further, Cognitive psychology, perception and neuroscience for comprehensions on how biological organisms solve similar problems.

Over the last decades, optimization of operational capabilities and navigational tactics of mobile robot have elicited the courtesy of so many investigators due to this simultaneous application of many research disciplines in mobile robotics. Still fruition in the field of Path analysis and planning has been slower than might have been anticipated from the exhilaration and moderately hasty enhancements of the early days of research. At this perspective, this research is motivated towards real-time autonomous navigation where the robot must have the ability to:

- Sense and cope with its environmental structure.

- Interpret the sensed information to obtain the knowledge of its position and the static as well as dynamic environmental situation.
- Plan a real-time route and control motion from an initial point to target in a workspace following a path that is either a curve or a series of jointed segments.
- Avoid situations that are harmful to people, property or itself without human assistance.
- Control the robot direction and velocity to reach the desired location avoiding obstacles and dead-end positions using human perception.
- Deliver smoother motion, shorter traveling time, or more clearance from the obstacle with respect to certain performance measures.

## 1.2 Overview of Major Goals:

To survive within unforeseen situations and to amend the effects of changing environment; the power of self-government or sturdy autonomy is obligatory, which implies that the robot should be able to govern its course of action by its own perceptive process, rather than following a fixed, hardwired sequence of superficially provided instructions. This thesis is enthused to the goal of design and development of Autonomous mobile robot enriched with a distinctive control skill such that robot has the ability:

- To move in its environment,
- To perform a number of different tasks,
- To adapt the deviations in its environment,
- To learn from experience and change its behaviour accordingly,
- To build internal representation of its world that can be used for reasoning processes like navigation,
- Finally, to choose foremost suggestions adequate to human intelligence for finding a way to the consigned endpoint.

If the robot endures kinematical firmness then another contest of this research work is to model an sensible controller which may provide a universal, vigorous, collision-free and augmented path so that mobile robot navigate in real world dynamic environment. Fuzzy control concept has already proven to be worthwhile in both global and local path planning tasks (details are given in chapter 2) for autonomous mobile objects. A set of linguistic fuzzy rules are developed here to implement expert knowledge under various situations. Sensor signals are fed to the controller and the output provides motor control commands (e.g. turn left or right). Both of Mamdani and Takagi-Sugeno fuzzy model are employed in control algorithm for experimental purpose. Under the control of the proposed fuzzy logic-based model, the mobile robot can generate reasonable trajectories towards the target by integrating different preliminary robotic behaviors (e.g. obstacle avoidance, wall and edge following, escaping dead end and target seeking).

The artificial life approach to evolutionary robotics is especially designed to grow different neural structures with complex dynamical properties for path recognition of autonomous mobile robot. Neural networks are often used to enhance and optimize the outcome of fuzzy logic based system, e.g. by introducing a learning ability. This learning ability is achieved by presenting a training set of different examples to the network and using learning algorithm, which changes the weights (or the parameters of activation functions) in such a way that network will reproduce a correct output for the input values associated with nonlinearities. The difficulty is how to assure that the network is sufficiently trained or not. So, another incentive for proposed research is to provide a cohesive framework capable of using both fuzzy inference system and neural network due to some appreciable similarities and dissimilarities between them such as: Both have the ability to deal with nonlinearities along with model free modeling approaches, can follow more human like reasoning paths than conventional methods, have high fault tolerance capabilities irrespective of mathematical modeling and the main divergence between them is that FL uses heuristics knowledge to form rules but NN tunes rules based on available sample data. This research is committed to appraise the performances of fabricated controllers during navigation of mobile robot in different simulation and experimental environmental scenarios along with comparison with previous research work for endorsement.

## 1.3 Thesis Structure:

The practices as organized in this thesis are approximately divided into nine chapters.

- Succeeding the introduction, Chapter 2 puts on the literature review of foregoing investigations on kinematics and analysis of mobile robot configuration, fuzzy logic controller: both Mamdani and Takagi Sugeno approach based and fuzzy-neuro controller implemented in navigation purpose.
- Chapter 3 studies the kinematics architecture of mobile robot configuration for weighing performance of the model robot pertaining to different mechanical aspects. The stability of presented kinematic and dynamic model of robot during tracking target has also been construed in a satisfactory manner.
- Chapter 4 delineates the concept of Mamdani-based fuzzy logic and hybridization of membership functions to design a reactive behavioural controller whose performance has also been assessed.
- Chapter 5 discourses the execution and evaluation of navigational operation of Takagi-Sugeno based fuzzy controller, whose rule base and membership functions is retained same as Mamdani based one.
- Chapter 6 pronounces an assimilation of fuzzy logic and neural network algorithms towards development of more optimized mobile robot controller.
- Chapter 7 describes hardware aspect of a simple mobile robot configuration by accumulating different sub modules.
- In Chapter 8 a comprehensive description of results and discussion has been carried out.
- In Chapter 9 Contributions and Conclusions of this research and future directions for further investigation has also been conferred.

The paper published related to the thesis has been listed at the last.

# 2 Literature Review

Designing robust global navigation technique for inexpensive mobile robot has been a challenge for scientists for many years. There is an increasing number of potential applications for autonomous mobile robots in indoor environments, ranging from cleaning, to surveillance, to search and rescue operations in burning buildings or hostage situations, to assisting the handicapped or elderly around the home. To realize these applications, all difficulties and challenges in this domain must be focused. The progress made in past decades in the field of kinematics and dynamic modeling, design techniques for intelligent controller and navigational path analysis of mobile robot are briefly reviewed here regarding some exclusive contributions to this domain.

## 2.1 Introduction:

Autonomous Mobile Robot must have the ability to move in its environment, to perform a number of different tasks, to adapt the changes in environments, to learn from experience and to change behavior accordingly, last but not the least to build internal representation of its world that can be used for reasoning process like navigation.

Among many issues relevant to autonomous operation, previous research works on two main computational issues are elaborated here: Modeling of Mobile Robot and Motion Planning based on localization or Path planning and following (Navigation). Modeling of mobile robots requires a preliminary analysis of the kinematic and dynamic constraints. Navigation can be considered as a process whose inputs are the specific knowledge of the environment, description of the current position, description of the destination and the agent's observations of the environment. The produced output is the appropriate movement orders to reach the destination position, avoiding obstacles and other exception situations that can arise.

This chapter provides details survey report within important aspects of research work to seek out optimal path and track the target in the competing clutter environment on the basis of sensory data and their structural significance using fuzzy logic (Mamdani and Takagi Sugeno both) and fuzzy-neural network.

## 2.2 Modeling (kinematic and dynamic analysis) of Wheeled Mobile Robot:

The kinematic model of a mobile robot is essentially the description of the admissible instantaneous motions in respect of the constraints. On the other hand, the dynamic model accounts for the reaction forces and describes the relationship between the above motions and the generalized forces acting on the robot. These models can be expressed in a canonical form which is convenient for design of planning and control techniques.

Modeling procedure can be inspired by definition of a wheeled mobile robot according to Muri and Neuman [73] as follows “A robot capable of locomotion on a surface solely through the actuation of wheel assemblies mounted on the robot and in contact with the surface. A wheel assembly is a device that provides or allows relative motion between its mount and a surface on which it is intended to have a single point of contact.” It is desirable that the vehicle kinematic design have the appropriate degrees of freedom (mobility) so that it adapts to surface variations and the wheels roll without slip. Mobility is enhanced by the use of omnidirectional wheels instead of conventional wheels [10]. The requirement of ideal rolling without sideways slipping for wheels imposes nonholonomic (non-integrable) constraints on the motion of the wheels of mobile robot [2]. The relationship between the rigid body motion of the robot and the steering and drive rates of wheels is developed by Alexander and Maddocks [5] based on constraint as ‘rolling without sliding'. Slippage due to misalignment of the wheels is investigated here by minimization of a nonsmooth convex dissipation functional that is derived from Coulomb's Law of friction. This minimization principle is equivalent to the construction of quasi-static motions.

Three different (though related) kinematical aspects have to be considered when designing a robot: mobility, control and positioning [17, 37]. The first one deals with the possible motions that the robot may follow to reach a final configuration along with any orientation. The second aspect deals with the choice of the kinematical variables: generalized velocities or coordinates. Finally, the third aspect: positioning, considers the localization system, used to estimate the actual robot pose (position and orientation) by reducing the robot's uncertainty region based on sensor measurements necessary to achieve an autonomous operation[13].

Dynamics constraints limit the acceptable values for derivatives of an agent's position over time, while Kinematic constraints limit motion along the configuration space. Kinematic limitations apply at any speed, while dynamics constraints become steadily more important as an agent operates at higher speeds. Robot design cannot escape all agent dynamics issues, as even a holonomic robot lacking any kinematic constraints will face some form of dynamics limitations, and in particular bounds on acceleration and velocity. Thus dynamics limitations are a nearly universal issue for mobile agents.

From the control point of view, the dynamics of nonholonomic systems can be divided in two parts: external and internal dynamics. The dimension of the external dynamics of nonholonomic systems depends on the number of inputs to the system and the dimension of the internal dynamics depends on the number of independent nonholonomic constraints [24]. Yun and Yamamoto [109] have characterized internal dynamics of the mobile robot under look-ahead control using a novel Lyapunov function which stated that the internal motion of mobile robot is asymptotically stable when the reference point is commanded to move forward and unstable for backward movement.

Moon et al. [69] has shown that a wheeled mobile robot can't move along a straight line exactly, even if kinematic imperfections are corrected perfectly, and this phenomenon is attributable to acceleration constraints on motor controllers. Kinematic model of parallel wheeled mobile robot fails to meet Brockett's necessary condition for feedback stabilization. This implies that no smooth or even continuous time invariant static state feedback law exists which makes the closed loop system locally asymptotically stable. Tracking control using direct Lyapunov method [54], time variant state feedback [74] and many other primitive methods are designed on the basis of kinematic model [34]. Stabilization and control of nonholonomic systems with dynamic equations have been considered in [11], backstepping based methods are presented in several papers [28, 51, 98].

Internal error occurs from inappropriate setting up of the parameters and the time constant. External error inevitably appears while a WMR is driving; it occurs by virtue of the two driving wheels' different friction and radius. To minimize such errors, Chung et al. [20] proposes a feedback controller that has two separated feedback loops; one of which is a position feedback, and the other an orientation feedback.

A robust adaptive controller based on backstepping algorithm is proposed [46, 83] to design an auxiliary wheel velocity controller for making the tracking error as small as possible in consideration with uncertainties in the kinematics of the robot and fuzzy logic techniques are employed to learn the behaviours of the unknown dynamics of the robot and the wheel actuators. A major advantage of the proposed method is that previous knowledge of the robot kinematics and the dynamics of the robot and wheel actuators is no longer necessary. The parameters characterizing the robot dynamics are updated on-line, thus providing smaller errors and better performance in applications in which these parameters can vary, such as load transportation. The stability of the whole system is analyzed using Lyapunov theory, and the control errors are proved to be ultimately bounded [66].

A combined feedback control scheme based on Lyapunov function candidate [22] is discussed for four obstacle cases in dynamic environments considering local minima problem by Deng et al. [21]. The controller includes virtual attractive force, repulsive force and detouring force, where the potential field function used for the design of the controller considers the Euclidean distance information and the magnitude information of the relative velocity between the robot and the target [33].

A dynamic model of a two-wheeled mobile robot has been derived [81, 101] which implies the translational motion and also rotational motion with 3 degrees of freedom of the body and here, the dynamic model is reduced to the kinematic model under certain assumptions. Arvin et al. [8] presents mobile robots motion control technique based on pulse-width modulation (PWM).

Chakraborty and Ghosal [17] have modeled the wheels of mobile robot as a torus and used a passive joint allowing a lateral degree of freedom to get a slip free motion in an uneven terrain without using variable length axle (VLA) which has several limitation in application. A feedback control law [23, 78], allowing a 2-wheel differentially driven mobile robot to track a prescribed trajectory has been developed by Zhang et al. [114] using the integral backstepping method and Lyapunov function for ensuring a trajectory tracking controller with global asymptotic stability.

Using the notion of virtual vehicle [3] and the concept of flatness [29], and applying the backstepping [28] methodology Zohar et al. recently proposes control schemes for trajectory tracking of mobile robot model which includes kinematic and dynamic effects on motion [116].

The harmonic drive system for non-linear controller to compensate for kinematic error in the presence of flexibility in high-speed regulation and trajectory tracking application has been proposed by Gandhi and Ghorbel [30]. The behaviour of space robots with torque and attitude controller has been discussed by Pathak et al. [82]. A receding horizon controller is may used for tracking control of wheeled mobile robots subject to nonholonomic constraint in the environments without obstacles. The control policy is derived from the optimization of a quadratic cost function, which penalizes the tracking error and control variables in each sampling time [36, 102]. A single curvature trajectory, which has a constant and large rotation radius, is proposed by Han et al. [42] as an optimal trajectory, in order to minimize the tracking error of the differential drive mobile robot while capturing a moving object along with the pre-determined initial states (i.e., position and orientation of the mobile robot and the final states).

## 2.3 Motion Planning for Mobile Robot:

The motion planning approach depends on two important properties of the agent and its planner: global planning and local planning. The former is based on the complete knowledge of the environment and the robot either from the modeling through a prior knowledge or from the perception through a sensory system. The second class consists of local control or behavioral strategies which have been considered here. The robot motion decision is made by considering the up to date status of the robot and the relationships with its environment (sensor information). The main advantage consists in the ability to handle the changing aspect of the environment because the structural modeling of the environment is not necessary [68]. Combining both, an agent can predict the result of an action within the environment without actually executing that action.

Depending on the use of the planning and observations on the environment the navigation systems can be classified as reactive and deliberative: In general, the reactive behaviors need less time to respond to the events as they work almost always with sensor information. The deliberative part refers to the planning like the action of projecting the current situation into the future to determine a chain of actions that will take the system to the goal position. An exclusively deliberative navigation method would fail when the real environment differs from the previous knowledge or expectations since it does not have the capacity to react to unexpected events or non-modeling obstacles. Robots whose navigation system is based on purely reactive systems have a series of drawbacks: 1) Lack of flexibility: To modify their behavior usually requires the reconstruction of the whole control system. 2) They are too local since they do not plan ahead in the future and good performances are not obtained when there is not relevant local information. 3) Inefficiency since they just react to events.

A hybrid approach, combining low-level reactive behaviors with higher level deliberation and reasoning, has since then been common among researchers e.g. [7]. The hybrid systems are usually modeled as having three layers; one deliberative, one sequence layer and one reactive layer. The deliberative layer prepares activities for future by monitoring human reaction. Specially, learning techniques can be deployed in this layer that makes the system more faults tolerant. The Sequencer Layer or supervisory layer, bridges the gap between the deliberative and the reactive layers. Its basic function is to rewire the reactive layer according to a global state obtained from the deliberative layer, thus deciding which set of behaviors that should be running. The reactive layer consists of subsystem like separate behaviors running in parallel, where each behavior has one specified non-complex task. The calculations in the reactive layer should be carried out in near real-time for safety critical considerations.

In this group of architectures [15, 31, 61, 88], different tasks are implemented as behaviors that compete for the robot's control. The implementation of the behaviors varies according to the architectures. While there are behaviors to avoid obstacles, to explore, to make maps, to identify objects and to detect changes in the environment, there are behaviors to avoid obstacles, to move towards the objective and to construct distributed maps. Behaviors are distributed in a hierarchy of levels where the superior levels include the functionality of the inferior ones, these behaviors are independent. Only the inferior level is implemented using behaviors with fuzzy rules [91].

## 2.4 Fuzzy Logic for Behavioral Navigation:

Behavioral Coordination problems can be split into two main sub-problems: how to decide which behavior should be activated at each instant and how to combine the results from different behaviors into one command to be sent to the virtual agent [4, 91, 105]. In this context fuzzy logic offers useful mechanisms to address the behavior coordination problem for virtual agent navigation in virtual environments.

Fuzzy set theory was introduced by Lofti Zadeh in the mid-sixties. In 1965 Lotfi Zadeh proposed fuzzy set theory, and published a paper [110]. Fuzzy logic has been applied to diverse fields, from control theory to artificial intelligence. This section presents a variety of fuzzy logic techniques which address the challenges posed by autonomous robot navigation. Stability analysis of fuzzy systems is a very important research field in fuzzy systems practically from the pioneer work of Mamdani and Assilian [65] on fuzzy control applications. A fuzzy logic based controller (Multi-Agents System Controller (MASC)) for regulating the number of agents released to the network is presented by Olajubu et al. [77] for a two-inputs-one-output system. For a given trajectory, the parameters of Mamdani-type-Fuzzy Logic Controller can be optimized [112] by the particle swarm optimization with three different cost functions in order to compare with different controller.

The fundamental behavior of a mobile robot can be described as a dynamic process of the interaction between the robot and its local environment, and then it is modeled and controlled for the motion-planning purpose. Based on behavior dynamics, the dynamic motion-planning problem of mobile robots is transformed into a control problem of the integrated planning-and-control system which can be transformed into a conventional optimization problem in the robot's acceleration space [52]. In the case of a partially known environment, a hybrid of global and local planning or navigation strategies can be achieved by developing a control algorithm having following qualities [64]:

- Inclusion of a priori knowledge;
- Robustness with regard to the environment modifications;
- Robustness with regard to the sensors imprecision;
- Use of linguistic rules which are a priori easily transportable from one robot to another.

A behavior controller integrates basic behaviors as separate navigators so that it can control the mobile robot's steering angle and linear velocity. A key issue in behavior-based control, however, is how to coordinate conflicts and competitions among multiple reactive behaviors efficiently [61]. During reactive navigation of mobile robot in a cluttered environment, local minima problem which can be solved by Fuzzy reinforcement learning algorithm based on human intelligence [14], Fuzzy decision making accompanied by an actual–virtual target switching strategy [71], the minimum risk method [106], Local obstacle avoidance method [19] etc. A methodology to design an ordinal fuzzy logic controller with application for obstacle avoidance of Khepera mobile robot is presented by Samsudin et al. [92]. Precup and Hellendoorn [86] present a survey on recent developments of analysis and design of fuzzy control systems focused on industrial applications reported after 2000. A new framework has been designed by Vidoni et al. [104] to manage robotic agents in order to get precise, real-time information from the real world. The parking problem of non-holonomic mobile robots has already been explained by a stable switching control strategy [70, 99]. A Learning Fuzzy controller has been introduced [66, 72] to analyze the performance of the different algorithms for the design of behaviors in mobile robotics, and to extract some general rules that can help in the process to design new behaviors. The intelligent part of the algorithm, Fuzzy Decision Maker (FDM) which enables the robot to do both the guidance-based tracking algorithm and the obstacle avoidance simultaneously has also been illustrated [9, 76]. A Self tuned fuzzy controller based on on-line optimization of a zero order Takagi–Sugeno fuzzy inference system (FIS) by a back propagation-like algorithm is successfully applied by Zemalache and Maaref [113] to minimize a cost function that is made up of a quadratic error term and a weight decay term that prevents an excessive growth of parameters. Two soft computing (SC)-based approaches, namely genetic-fuzzy and genetic-neural systems and a conventional potential field method (PFM) have been developed by Hui and Pratihar [47] for a comparative study of various robot motion planning schemes. Design of a distributed coordination control algorithm for each robot in the group has been made by Zou and Pagilla [117] to achieve, and maintain, a particular formation while ensuring navigation of the group and considering constraint forces which are used in the development of the dynamics of a system of constrained particles with inertia

## 2.5 Navigation using Fuzzy-Neuro approach:

Fuzzy neural networks have several features that make them well suited to a wide range of knowledge engineering applications. These strengths include fast and accurate learning, good generalization capabilities, excellent explanation facilities in the form of semantically meaningful fuzzy rules, and the ability to accommodate both data and existing expert knowledge about the problem under consideration. Kasabov et al. [55] investigates adaptive learning, rule extraction and insertion, and neural/fuzzy reasoning for a particular model of a fuzzy neural network. A learning algorithm based on neural network techniques is developed by Zhu and Yang [115] to tune the parameters of membership functions, which smoothes the trajectory generated by the fuzzy logic system which is designed with two basic behaviors, target seeking and obstacle avoidance.

However, to fully exploit the potential of FNN structures, efficient parallel-processing implementations are highly desired. Gobi and Pedrycz [37] investigates the high potential to provide strong mechanisms for building intelligent systems and versatile neurofuzzy platform with a topology strongly influenced by theories of fuzzy modelling. The neural fuzzy controller has already been developed [25, 100] based on the Generalized Dynamic Fuzzy Neural Networks (GDFNN) learning algorithm for real-time control of an autonomous mobile robot. Not only the parameters of the controller can be optimized, but also the structure of the controller can be self-adaptive. Useful heuristic rules were combined with the fuzzy Kohonen clustering network (FKCN) by Song et al. [96] to build the desired mapping between perception and motion for getting much faster response to unexpected events and less sensitive to sensor misreading than conventional approaches.

Nefti et al. [75] introduces the adaptive navigation system (ANFIS) to mobile robot navigation in an unknown or partially unknown environment. This proposed controller based on integrated reactive-cognitive parts, learns and generates the required knowledge for achieving the desired task. Cooperative behavior of several mobile robots using online inter-communication among them has been described by Parhi et al. [80] applying rule-based and rule-based-neuro-fuzzy techniques which are analyzed for multiple mobile robots navigation in an unknown or partially known environment. A supervisory fuzzy neural network (FNN) control system is designed by Lin et al. [62] to track periodic reference inputs. A supervisory controller, which is designed to stabilize the system states around a defined bound region and an FNN sliding-mode controller, combines the advantages of the sliding-mode control with robust characteristics and the FNN with on-line learning ability. A successful way of structuring the navigation task of autonomous mobile robot in a real-world environment, avoiding structured and unstructured obstacles, especially in a crowded and unpredictably changing environment, dealing with the issues of individual robot behaviors, is discussed by Parhi and Singh [79]. In this research, action coordination of the behaviors has been addressed using fuzzy logic.

## 2.6 Sensors for Mobile Robots

Different types of sensors have been used for mobile robot navigation. They can be classified into three categories: (i) Ultrasonic Sensors, (ii) Infrared Sensors, and (iii) Other types of Sensors.

### 2.6.1 Ultrasonic Sensors for Robot Navigation:

Wu and Tsai [107] have proven that the combination of three ultrasonic transmitters and two receivers can determine both the position and the orientation (localization) of an AMR with respect to a reference frame uniquely. A method for estimating the position and heading angle of a mobile robot moving on a flat surface has been proposed by Boem and Cho [12]. Their localization method utilizes two passive beacons and a single rotating ultrasonic sensor.

The reasonable researches [26, 94, 95] have involved ultrasonic sensor-based motion planning for a single robot. They have used information from assumed sensor media as input to the motion-planning algorithm.

Kleeman and Kuc [58] have established that two transmitters and two receivers are necessary and sufficient for a mobile robot to distinguish between planes, corners and edges. Ko et al. [59] have described a method to extract acoustic landmarks for the indoor navigation of a single mobile robot using an array of ultrasonic sensors. Hong and Kleeman [45] have discussed the sensing of room boundaries for a mobile robot using an ultrasonic sensor array. They have implemented their algorithm with an extended Kalman Filter.

### 2.6.2 Infrared Sensors for Robot Navigation:

Everett and Flynn [27] have described a programmable near-infra-red amplitude detection sensor for navigation in an unstructured environment. Yu and Malik [108] have discussed the navigation of a mobile robot using an infrared sensor to avoid collision with obstacles. Kube and Zhang [60] have also used infrared sensors for obstacle avoidance. During navigation, their robot's infrared sensors can detect obstacles within a range of 1.5 m. Vandorpe et al. [102, 103] have designed an autonomous mobile robot using an infrared sensor for avoiding obstacles. Their infrared imaging sensor gives a complete panoramic image of the environment.

### 2.6.3 Other Sensors Used in Navigation:

Borenstein et al. [13] have discussed the navigation of a single mobile robot with various sensory techniques. They have shown that the magnetic compass is a very good sensor for determining the location and heading angle (x, y, and θ) for a mobile robot. However, the sensor is not appropriate for obstacle distance measurement. Gonzalez et al. [40] have presented an algorithm for efficiently estimating the position of a mobile robot based on a radially-scanning laser range finder. Their method is suitable for a single mobile robot navigating in an unknown environment.

## 2.7 Conclusion:

Firstly the kinematics and dynamic analysis of differential drive mobile robot has been addressed here, and the problem of model based constraints and trajectory tracking have been found in a number of research work. This chapter also provides a detailed review report which has been used in last decades by many researchers in the area of new intelligent control techniques like Fuzzy Logic and Fuzzy-Neural Network. Sensors used in different robotic application are also reviewed here. From the survey it has been perceived that the mobile robot navigation can be controlled successfully in a complex, unknown and dynamic environments using the above strategies.

# 3 Kinematic Architecture of Mobile Robot

When it is necessary for a mobile robot to perform operations along a specific path in a complex environment, motion planning is a critical performance feature as it needs much more treatments to allow the robot to move between its current and final configurations without any collision within the surrounding environment. To reach high control performance, the self-adaptive robot's navigation and path planning algorithm must be consistent with the kinematics of the mobile robot. The controlled model [10] takes into account the robot kinematic and dynamic constraints, leading to bounded velocities and accelerations that are compatible with those of a real mobile robot can perform.

## 3.1 Introduction:

To design appropriate mobile robot for tasks and to understand how to create control software for an instance of mobile robot hardware, the mechanical behavior of the robot has to be understood. The different aspects of designing wheeled mobile robot can be depicted as: positioning of the robot model in the environment, maneuverability analysis with respect to kinematic constraints, generalized control of developed Kinematic and Dynamic model, and design of control law after solving the trajectory tracking problem using integral backstepping algorithm based on a single Lyapunov function for mobile robot navigation.

There is no direct way to measure a mobile robot’s position instantaneously. Instead, one must integrate the motion of the robot over time. The process of understanding the motions of a robot begins with the process of describing the contribution each wheel provides for motion. By the same routine, each wheel also imposes constraints on the robot’s motion. The wheels and the ground are considered as rigid bodies and single point contact is assumed between the wheel and the ground. The equations describing the geometry of the wheel and the ground are assumed to be sufficiently smooth and continuous such that derivatives up to second-order exist. Modeling of mobile robot with differential drive wheels as control systems may be addressed with a differential geometric point of view by considering only the classical hypothesis of "rolling without slipping" [5].

As WMR has more degrees of freedom than the number of inputs under nonholonomic constraints, a Lyapunov candidate function [81] can be chosen to design a single controller that is able to achieve both trajectory tracking and stabilization for mobile robot towards goal avoiding obstacles with unknown kinematic and dynamic parameters [54].

In the following section, we introduce notation that allows expression of robot motion in a global reference frame as well as the robot’s local reference frame. Then, using this notation, simple forward kinematic models of motion describes how the robot as a whole moves as a function of its geometry and individual wheel behavior. Next, the types of wheel used in the present research work and its kinematic constraints for individual wheels are formally described. Depending on the mechanical structure, such constraints can be integrable or not; this has direct consequence on a robot’s mobility. Then, modeling of mobile robot is done by combining these kinematic constraints. The proposed controller is claimed to be robust against the changes in mass and inertia parameters of robot. The simple and clear control laws based on Lyapunov function is verified to achieve the desired performance eliminating the tracking error while seeking target with obstacle avoidance nature.

### 3.2 Position of Mobile Robot Model:

When an autonomous mobile robot performs tasks such as free-range path tracking and reactive navigation, the capability to estimate its position with respect to a reference frame is very important (localization). This is particularly important in mobile robotics because of its self-contained and mobile nature; a clear mapping between global and local frames of reference is required.

Wheels are tied together based on robot chassis geometry, and therefore their constraints combine to form constraints on the overall motion of the robot chassis. But the forces and constraints of each wheel must be expressed with respect to a clear and consistent reference frame. The robot has been considered as a rigid body on wheels and moves on a horizontal plane during analysis. The total dimensionality of this robot chassis on the plane is three, two for position in the plane and one for orientation along the vertical axis, which is orthogonal to the plane.

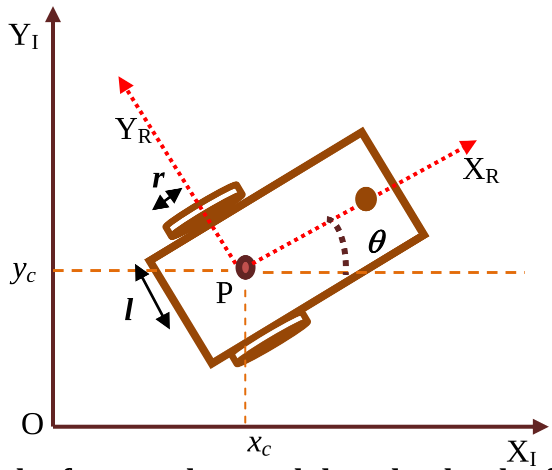



**Figure 3.1: The global reference plane and the robot local reference frame**

Let us consider an arbitrary inertial frame O: $\{X_I, Y_I\}$ on the plane as the global reference frame and P: $\{X_R, Y_R\}$ the robot's local reference frame (Fig 3.1). To specify the position of the robot, choose a reference point $P$ on the robot chassis as its position. The position of $P$ in the global reference frame is specified by coordinates $x_c$ and $y_c$, and the angular difference between the global and local reference frames is given by $\theta$.

Therefore the robot position: $\xi_I = [x_c \quad y_c \quad \theta]^T$ (3.1)

To map motion along the axes of the global reference frame to motion along the axes of the robot's local reference frame, the orthogonal rotation matrix can be used:

$$R(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.2}$$

Now, we can compute the robot's motion in the global reference frame from motion in its local reference frame:

$$\dot{\xi}_I = R(\theta)^{-1}\dot{\xi}_R \tag{3.3}$$

After defining these reference frames formally, the resulting formalism is used to annotate the kinematics of individual wheels and whole robots.

**3.3 Forward Kinematic Model:**

Deriving a model for the whole robot's motion is a bottom-up process. For the differential drive robot (Figure 3.1) has two wheels, each with diameter *r*. Given a point P centered between the two drive wheels, each wheel is a distance *l* from P. Given *r, l*, θ and the spinning speed of each wheel, $\dot{\phi}_1$ and $\dot{\phi}_2$, a forward kinematic model would predict the robot's overall speed in the global reference frame:

$$\dot{\xi}_I = [\dot{x} \quad \dot{y} \quad \dot{\theta}]^T = f(l, r, \theta, \dot{\varphi}_1, \dot{\varphi}_2) \tag{3.4}$$

The strategy will be to first compute the contribution of each of the two wheels in the local reference $\dot{\xi}_R$ . First consider the contribution of each wheel's spinning speed to the translation speed at *P* in the direction of $+X_R$. If one wheel spins while the other wheel contributes nothing and is stationary, since *P* is halfway between the two wheels, it will move instantaneously with half the speed: $\dot{x}_{r1} = \frac{1}{2} r\dot{\varphi}_1$ and $\dot{x}_{r2} = \frac{1}{2} r\dot{\varphi}_2$. In a differential drive robot, these two contributions can simply be added to calculate the $\dot{x}_R$ component of $\dot{\xi}_R$. Neither wheel can contribute to sideways motion in the robot's reference frame, so $\dot{y}_R$ is always zero.

Once again, the contributions of each wheel can be computed independently and just added for computing rotational component $\dot{\theta}_R$. Consider the right wheel (we will call this wheel

1). Forward spin of this wheel results in counterclockwise rotation at point P. If wheel 1 spins alone, the robot pivots around wheel 2. The rotation velocity $\omega_1$ at P can be computed because the wheel is instantaneously moving along the arc of a circle of radius $2l$: $\omega_1 = \frac{r\dot{\varphi}_1}{2l}$

The same calculation applies to the left wheel, with the exception that forward spin results in clockwise rotation at point P: $\omega_2 = \frac{-r\dot{\varphi}_2}{2l}$

Combining these individual formulas yields a Forward Kinematic model for the differential-drive mobile robot in reference frame: $\dot{\xi}_R = \begin{bmatrix} \frac{r\dot{\varphi}_1}{2} + \frac{r\dot{\varphi}_2}{2} \\ 0 \\ \frac{r\dot{\varphi}_1}{2l} + \frac{-r\dot{\varphi}_2}{2l} \end{bmatrix}$

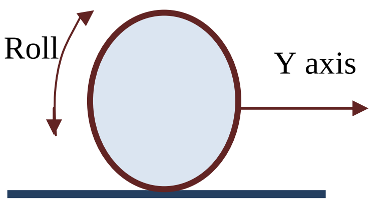
Combining these individual formulas yields a Forward Kinematic model for the differential-drive example robot:

$$\dot{\xi}_I = R^{-1}(\theta) \begin{bmatrix} \frac{r\dot{\varphi}_1}{2} + \frac{r\dot{\varphi}_2}{2} \\ 0 \\ \frac{r\dot{\varphi}_1}{2l} + \frac{-r\dot{\varphi}_2}{2l} \end{bmatrix} \quad (3.5)$$

This approach to kinematic modeling can provide information about the motion of a robot given its component wheel speeds in straightforward cases. However, we wish to determine the space of possible motions for each robot chassis design. To do this, we must go further, describing formally the constraints on robot motion imposed by each wheel.

### 3.4 Types of Wheel:

A wheeled mobile robot is a vehicle which is capable of an autonomous motion (without external human driver) because it is equipped with motors that are driven by output from on boarded microcontroller based on sensor information. We can observe two constraints for every wheel type while a wheeled robot is in motion. The first constraint enforces the concept of rolling contact as represented in Figure 3.2 (a).The second constraint enforces the concept of no lateral slippage, that the wheel must not slide orthogonal to the wheel plane as shown in Figure 3.2 (b). The initial stage of a kinematic model of the robot is to express constraints on the motions of individual wheels. Thereby we can compute the movement of the entire robot by combining the motions of individual wheels.



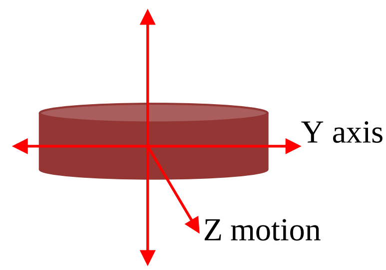




**Figure 3.2 (a): Rolling motion** **Figure 3.2 (b): Lateral slip**

Based on the geometrical constraints we can categorize the five basic wheel types: Conventional Fixed standard wheel, Steered standard wheel, Castor wheel, Swedish wheel and Spherical wheels. The castor wheel, Swedish wheel and spherical wheel impose no kinematic constraints on the robot chassis, since $\dot{\xi}_I$ can range freely in all of these cases owing to the internal wheel degrees of freedom. Only fixed standard wheels and steerable standard wheels have impact on robot chassis kinematics and therefore require consideration when computing the robot's kinematic constraints.

The fixed standard conventional wheels are the most widely used among wheel mobile robots with wheeled locomotion. These wheels are simple to construct, require less maintenance, provide smooth motion, offer high load carrying capacity and are cheap.

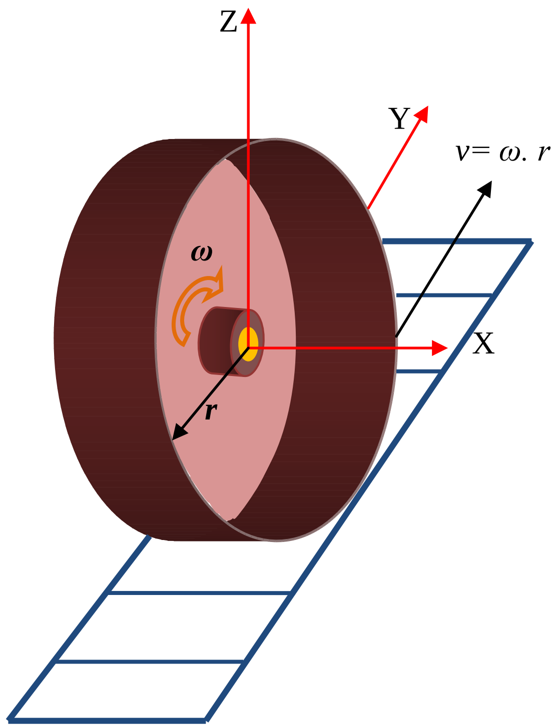



**Figure 3.3: Schematic view of conventional wheel**

The axis of rolling is orthogonal to the steering axis and the center of the wheel is at the intersection of these two axes. It allows travel along a surface in the direction of the wheel orientation, and rotation about the point-of-contact between the wheel and the floor shown in Figure 3.3. The rotational degree of freedom is slippage, since the point-of-contact is not stationary with respect to the floor surface. Even though we define the rotational slip as a degree of freedom, we do not consider slip transverse to the wheel orientation a degree of freedom, because the magnitude of force required for the transverse motion is much larger than that for rotational slip.

## 3.5 Analysis of Wheel Kinematic Constraints:

Usually, the mechanical mobile robot solution namely "two-wheel differential drive mobile robot" has three wheels minimum. Two separately controlled "drive wheels" have a common horizontal axis which is fixed (regarding its body) during robot operation. By their angular velocities, these "drive wheels" assure the mobility of the mobile robot. One free wheel, (namely "castor" wheel which is a passive one) assure the robot equilibrium, is mounted independently on a vertical axis not on a driven axis of the mobile robot body. In consequence, a castor wheel is automatically and free aligned on the route as a result of the forces developed by only the two "drive wheels" [16].

The speed difference between both two independently-driven coaxial wheels results in a rotation of the vehicle about the center of the axle while the wheels act in concert to produce motion in the forward or reverse direction. Such a robot can rotate on the spot (i.e., without moving the midpoint between the wheels), provided that the angular velocities of the two wheels are equal and opposite. Mobile robots operate at relatively low speeds and we assume vertical motion is absent.

However, several important assumptions will simplify the analysis. We assume that, the wheel always remains in vertical during the motion of a robot and there is no sliding at the single point of contact between the ground plane and the wheel. It means that the wheel is in motion under only pure rolling conditions and rotation about the vertical axis through the contact point.

Under these assumptions, we present two constraints for every wheel type. The first constraint enforces the concept of rolling contact that the wheel must roll when motion takes place in the appropriate direction. The second constraint enforces the concept of no lateral slippage that the wheel must not slide orthogonal to the wheel plane.

There is no vertical axis of rotation or steering for the fixed standard wheel. It means the angle between the chassis and the wheel axis is fixed, therefore it is limited to move the robot back and forth along the wheel plane and rotation around its contact point with the ground plane.

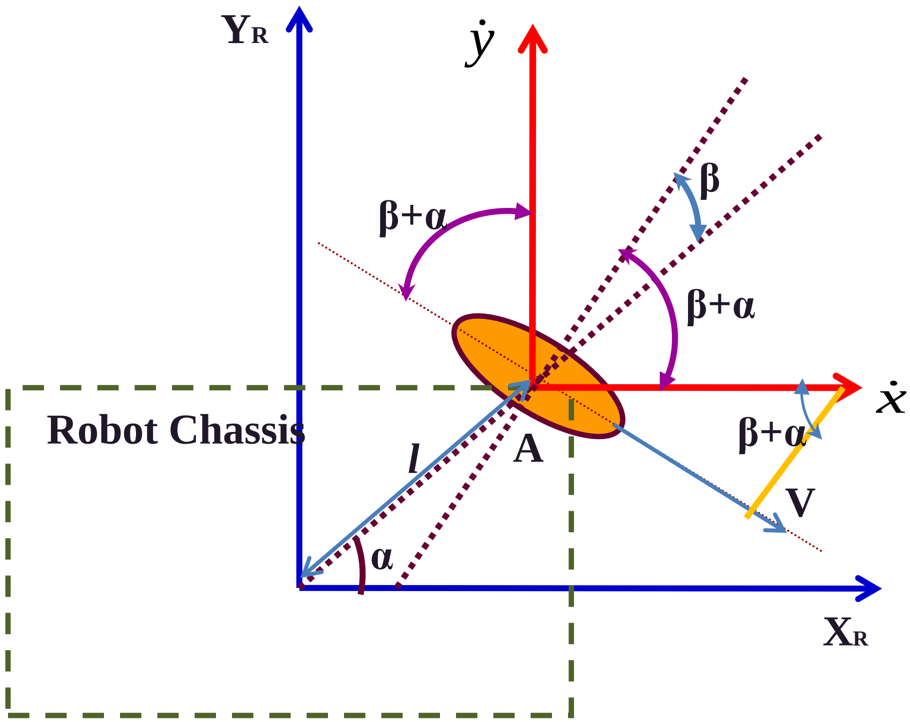



**Figure 3.4: Fixed standard wheel and its parameters**

Figure 3.4 describes a fixed standard wheel and its position relative to the robot's local reference frame. The position of the robot is then expressed in polar coordinates by distance $l$ and angle $\alpha$, $\beta$ denotes the angle of the wheel plane relative to the robot chassis. This angle is fixed since the fixed standard wheel is not steerable. Consider a wheel of radius $r$ has its rotational position around its horizontal axle is a function of time $t$: $\varphi\ (t)$.

By the adequate amount of wheel spin in order to get pure rolling at the contact point, the wheel imposes that all movement along the direction of the wheel plane:

$$\left[\sin(\alpha+\beta) \quad -\cos(\alpha+\beta) \quad -l\cos\beta\right] R(\theta)\dot{\xi}_I - r\dot{\varphi} = 0 \tag{3.6}$$

The sliding constraint for this wheel enforces the wheel's motion normal to the wheel plane must be zero: $\left[\cos(\alpha+\beta) \quad \sin(\alpha+\beta) \quad l\sin\beta\right] R(\theta)\dot{\xi}_I = 0$ (3.7)

We can now compute the kinematic constraints of the robot chassis associated with wheels. The key idea is that each wheel imposes zero or more constraints on robot motion, and so the process is simply one of appropriately combining all of the kinematic constraints arising from all of the wheels based on the placement of those wheels on the robot chassis.

Fixed standard wheels have impact on robot chassis kinematics and therefore require consideration when computing the robot's kinematic constraints. Suppose that the robot has a total of $N_f$ fixed standard wheels. ß refer to the orientation of the $N_f$ fixed standard wheels. In the case of wheel spin, the fixed wheels have rotational positions around the horizontal axle that vary as a function of time, denoted as $\varphi_f$. The castor wheel is unpowered and is free to move in any direction, so we ignore this third point of contact altogether as it does not impose any kinematic constraint.

Now it is easier to express the rolling constraints of all wheels into a unique expression as represented by equation:

$$J_{1f} R(\theta)\dot{\xi}_I - J_{2f}\dot{\varphi} = 0 \tag{3.8}$$

$$=> \dot{\xi}_I = R(\theta)^{-1} J_{1f}{}^{-1} J_{2f}\dot{\varphi} \tag{3.9}$$

Where $J_{1f}(\beta)$ denotes a matrix ($N_f$x3) for all fixed standard wheels to their motions along their individual wheel planes, and $J_{2f}$ is a constant diagonal matrix $N_f$ X $N_f$ of all standard wheels radii.

In the similar way we can formulate for the sliding constraints by combining all wheels into a single expression:

$$C_{1f} R(\theta)\dot{\xi}_I = 0 \tag{3.10}$$

Where $C_{1f}$ is of ($N_f$x 3).The above equation is a constraint over all standard wheels that their components of motion orthogonal to their wheel planes must be zero. This sliding constraint over all fixed standard wheels has the most significant impact on defining the overall maneuverability of the robot chassis.

Combining (3.8) & (3.10) in a matrix form,

$$\begin{bmatrix} J_{1f} \\ C_{1f} \end{bmatrix} R(\theta)\dot{\xi}_I = \begin{bmatrix} J_{2f} \\ 0 \end{bmatrix} \dot{\varphi} \tag{3.11}$$

To employ the fixed standard wheel's rolling constraint formula, we must first identify each wheel's values for α and ß. Suppose that the robot's local reference frame is aligned such that the robot moves forward in the direction of $+X_R$.

In the present direction of movement, for the right wheel $\alpha$=-π/2 and $ß$=π and for the left wheel $\alpha$=π/2 and $ß$=0. Note the value of $ß$ for the right wheel is necessary to ensure that positive spin causes motion in the $+X_R$ direction. Because the two fixed standard wheels are parallel, equation (3.5) results in only one independent equation. So, for the given values equation (3.11) can be written as

$$\begin{bmatrix} \begin{bmatrix} 1 & 0 & l \\ 1 & 0 & -l \end{bmatrix} \\ \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \end{bmatrix} R(\theta)\dot{\xi}_I = \begin{bmatrix} J_{2f} \\ 0 \end{bmatrix} \begin{bmatrix} \dot{\varphi}_1 \\ \dot{\varphi}_2 \end{bmatrix} \tag{3.12}$$

$$=> \dot{\xi}_I = R(\theta)^{-1} \begin{bmatrix} 1 & 0 & l \\ 1 & 0 & -l \\ 0 & 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} r & 0 \\ 0 & r \end{bmatrix} \begin{bmatrix} \dot{\varphi}_1 \\ \dot{\varphi}_2 \end{bmatrix}$$

Suppose that the robot is positioned such that $\theta = \pi/3$, $r$=1 and $l$=1. If the robot engages its wheels unevenly, with speeds $\dot{\phi}_1$=4 c.m/s and $\dot{\phi}_2$=2 c.m/s, we can compute its velocity in the global reference frame:

$$\dot{\xi}_I = \begin{bmatrix} \cos\frac{\pi}{3} & \sin\frac{\pi}{3} & 0 \\ -\sin\frac{\pi}{3} & \cos\frac{\pi}{3} & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 4 \\ 2 \end{bmatrix}$$

$$=> \dot{\xi}_I = \begin{bmatrix} 0.75 \\ 1.29 \\ 0.5 \end{bmatrix}$$

## 3.6 Mobile Robot Maneuverability:

The overall maneuverability of a robot is a combination of the mobility available based on the kinematic sliding constraints of the standard wheels, plus the additional freedom contributed by steering and spinning of the steerable standard wheels.

### 3.6.1 Degree of mobility:

The kinematic mobility of a robot chassis is its ability to directly move in the environment. The basic constraint limiting mobility is the rule that every wheel must satisfy its sliding constraint. Therefore, we can formally derive robot mobility by starting from equation (3.10) which imposes the constraint that every fixed standard wheel must avoid any lateral slip.

Robot chassis kinematics is therefore a function of the set of independent constraints arising from all standard wheels. The mathematical interpretation of independence is related to the rank of a matrix. Therefore rank of [$C_{1f}$] is the number of independent constraints. The greater the number of independent constraints, and therefore the greater the rank of [$C_{1f}$], the more constrained is the mobility of the robot.

Now we can define a robot's degree of mobility $\delta_m$ :

$$\delta_m = 3 - rank[C_{1f}] \qquad (3.13)$$

The dimensionality of the null space (*dim N*) of [$C_{1f}$] matrix is a measure of the number of degrees of freedom of the robot chassis that can be immediately manipulated through changes in wheel velocity.

In the case of the differential drive robot in figure 3.1, the two wheels are aligned along the same horizontal axis. In fact, the second wheel imposes no additional kinematic constraints on robot motion since its zero motion line is identical to that of the first wheel. Differential-drive chassis has only one independent kinematic constraint. Therefore, rank [$C_{1f}$] = 1 and $\delta_m$=2. This fits with intuition: a differential drive robot can control both the rate of its change in orientation and its forward/reverse speed, simply by manipulating wheel velocities.

### 3.6.2 Degree of Steerability:

The degree of mobility defined above quantifies the degrees of controllable freedom based on changes to wheel velocity. Steering can also have an eventual impact on a robot chassis pose$\xi$, although the impact is indirect because after changing the angle of a steerable standard wheel, the robot must move for the change in steering angle to have impact on pose.

As with mobility, we care about the number of independently controllable steering parameters when defining the degree of steerability $\delta_S$, but it deals only with steerable wheels. As we have taken the differential drive along with only the fixed standard wheels, so here $\delta_S$=0, i.e. the robot has no steerable standard wheels.

### 3.6.3 Maneuverability Measurement:

The overall Degrees of Freedom (DOF) that a robot can manipulate is called the degree of maneuverability ($\delta_M$). Thus the maneuverability comprises with the degrees of freedom that the robot changes its position directly through wheel velocity and the degrees of freedom that it indirectly manipulates by changing the steering configuration and moving.

$$\delta_M = \delta_m + \delta_s \tag{3.14}$$

Figure 3.5 represents three wheeled differential mobile robot having two fixed standard wheels and one castor wheel. For this type of robot, rank [$C_{1f}$] is one and it has no steerable standard wheels.

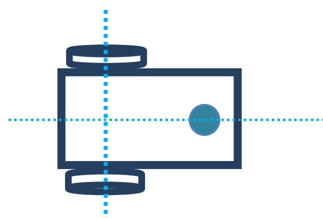


**Figure 3.5: Differential drive mobile robot with a castor wheel**

This results in the degree of mobility $\delta_m$=2 and the degree of steerability $\delta_s$=0,

∴The degree of maneuverability $\delta_M = \delta_m + \delta_s = 2$

### 3.6.4 Degrees of Freedom:

For given the kinematic constraints of the robot, its velocity space describes the independent components of robot motion that the robot can control. The number of dimensions in the velocity space of a robot is the number of independently achievable velocities. This is also called the differentiable degrees of freedom (DDOF). A robot's DDOF is always equal to its degree of mobility $\delta_m$. From the investigation of other types of wheels and drive configurations, it can be generally stated that there is an inequality relation at work: DDOF $\leq \delta_M \leq$ DOF. Just as workspace *DOF* governs the robot's ability to achieve various poses, so the robot's DDOF governs its ability to achieve various paths. For example, a two fixed standard wheeled differential drive has the following degree of maneuverability, $\delta_M$ =2. The DDOF of two wheeled differential drive is indeed 2. So, DDOF=DOF

## 3.7 Holonomicity of Mobile Robot:

In the robotics community, when describing the path space of a mobile robot, often the concept of holonomy is used. The term holonomy has broad applicability to several mathematical areas, including differential equations, functions and constraint expressions. In mobile robotics, the term refers specifically to the kinematic constraints of the robot chassis.

Holonomic vs. Nonholonomic:

- A nonholonomic kinematic constraint requires a differential relationship, such as the derivative of a position variable. Furthermore, it cannot be integrated to provide a constraint in terms of the position variables only. A holonomic kinematic constraint can be expressed as an explicit function of position variables only. For example, in the case of a mobile robot with a single fixed standard wheel, a holonomic kinematic constraint would be expressible using $\alpha$, *ß*, $l$, $r$, $\varphi$, $x$, $y$, $\theta$ only. Such a constraint may not use derivatives of these values, such as $\dot{\varphi}$ or $\dot{\xi}$.
- A nonholonomic mobile robot configuration is described by more than three coordinates. Three values are needed to describe the location and orientation of the robot, while others are needed to describe the internal geometry. However, a holonomic mobile robot can be described by three coordinates. The internal geometry does not appear in the kinematic equations of the abstract mobile robot, so it can be ignored. The robot can instantly develop a wrench or accelerate in an arbitrary combination of directions X, Y, θ.

- Nonholonomic robots are most prevalent because of their simple design and ease of control. By their nature, nonholonomic mobile robots have fewer degrees of freedom than holonomic mobile robots. These few actuated degrees of freedom in nonholonomic mobile robots are often independently controllable or mechanically decoupled, further simplifying the low-level control of the robot. Since they have fewer degrees of freedom, there are certain motions they cannot perform. This creates difficult problems for motion planning and implementation of reactive behaviors.
- Holonomicity, offers full mobility with the same number of degrees of freedom as the environment. This makes path planning easier because there aren't constraints that need to be integrated. Implementing reactive behaviors is easy because there are no constraints which limit the directions in which the robot can accelerate.
- In case of nonholonomic mobile robot, the wheels rotate in the forward direction and then backward to its previous angular position, the robot will not necessarily arrive in the same location due to slippage or any other conditions.
- In case of holonomic mobile robot, the wheels rotate in the forward direction and then backward to its previous angular position, the robot will arrive in the same location. So, holonomic robot can perform both Forward Kinematics (The angular rate difference between both wheels determines position & orientation of robot) and Inverse Kinematics (The position and orientation of a robot determines the angular rate difference between both wheels).

Considering equation (3.7), this constraint must use robot motion rather than pose because the point is to constrain robot motion perpendicular to the wheel plane to be zero. The constraint is nonintegrable, depending explicitly on robot motion. Therefore, the sliding constraint is a nonholonomic constraint and the robot is a nonholonomic one.

## 3.8 Kinematic Model of Mobile Robot:

The model of mobile robot (in Figure 3.6) consists of a vehicle chassis with two driving wheels mounted on the same axis and a front point sliding support. Both wheels have the same diameter denoted by '*2r*' and separated by distance '*2R*'. The two driving wheels are independently driven by two D.C gear motors to achieve the motion and orientation. The kinematics of the differential drive mobile robot is based on the assumptions are as follows [79]:

(1) Mobile robot moves on a plane surface.

(2) The wheel of a mobile robot rolls on the floor without translational slip.

(3) The wheel of a mobile robot makes rotational slip at the contact point between each wheel and the floor.

(4) The robot motion is slow such that the longitudinal traction & lateral force exerted on the robot's tires do not exceed the maximum static friction between tires and floor.

In Figure 3.6, Let $x_c$, $y_c$ be the Cartesian coordinates of the point C in the middle of the rear axle respectively $x_g$, $y_g$ the coordinates of the center of mass of the platform, the point G, and let $\theta$ be the angle between the heading direction and the $OX_I$-axis specifying the orientation of the local platform with respect to the inertial frame. The distance between points G and C is '*d*'. The generalized coordinates $q_g=[\ x_g\ y_g\ \theta]^T$ or $q_c=[\ x_c\ y_c\ \theta]^T$ completely specifies the position of the robot in the $X_IOY_I$ inertial Cartesian frame with a linear speed $v_c$ ($[\dot{x}_c \quad \dot{y}_c]^T$) and an angular velocity $\omega$ ($\dot{\theta}$).

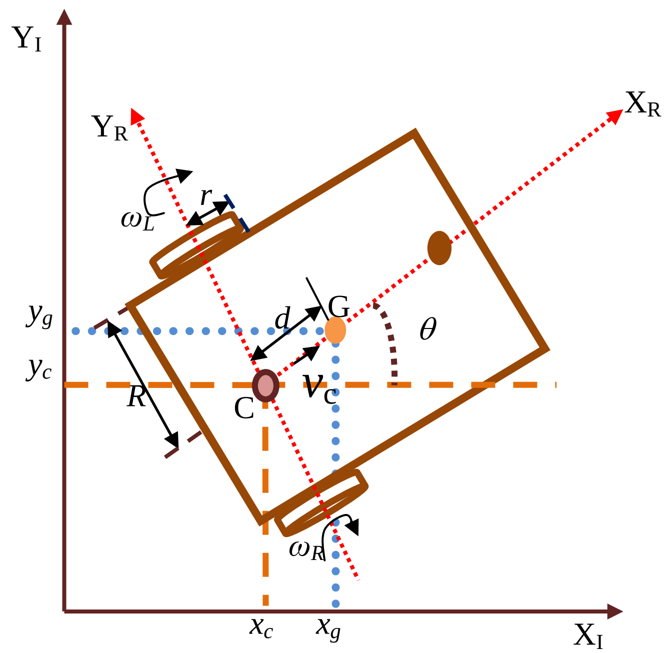



**Figure 3.6: Kinematic Analysis of Mobile Robot**

There three fundamental operations during kinematic motion [5]:

- If the angular velocities are identical ($\omega_R$=$\omega_L$), both as values and relative senses, the robot makes a linear motion. The direction on the linear motion, forward or backwards, depends of the opposite group of sense of the driven wheels angular velocities.
- If the angular velocities are identical as values but opposite as senses ($\omega_R$=-$\omega_L$), the robot make a "spin" motion. The spin motion is a rotation of the mobile robot body around its vertical axis passing through the geometrical symmetry point (or center of gravity). There is a particularity of this mechanical configuration, because only the two-wheel differential drive mobile robot can do this type of motion, very useful to escape outside from difficult obstacles.
- If the angular velocities are different as values and with the same senses, the robot makes a curve motion. Of course, the characteristics of the curve motion, i.e. the curvature coefficient *k* of the curve-segment trajectory, depend of the differences between the values of the two drive wheels. As the difference is smaller, as the curve motion tends to a linear motion.

The kinematics of the differential drive mobile robot is based on the assumption of pure rolling and there is no slip between the wheel and surface: $v_c = \frac{v_R + v_L}{2}$ and $\omega = \frac{v_R - v_L}{2R}$

Where, $v_R = r\,\omega_R$ and $v_L = r\,\omega_L$

So, in matrix form: $\begin{bmatrix} v_c \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ \frac{r}{2R} & -\frac{r}{2R} \end{bmatrix} \begin{bmatrix} \omega_R \\ \omega_L \end{bmatrix}$ (3.15)

Suffix $R$, $L$ and $t$ stand for right, left wheel and tangential (with respect to its center of gravity point of mobile robot) respectively.

From Figure 3.6, we can derive, $x_g = x_c + d\cos\theta$ and $y_g = y_c + d\sin\theta$. The linear velocity $v_c$ can be decomposed at point C in two components, as: $\dot{x}_c = v_c \cos\theta$ and $\dot{y}_c = v_c \cos\theta$ .

So, the velocity components of $v_g$ at point G,

$$\left.\begin{aligned} \dot{x}_g &= v_c \cos\theta - d\omega \sin\theta \\ \dot{y}_g &= v_c \sin\theta + d\omega \cos\theta \end{aligned}\right\} \qquad (3.16)$$

By eliminating $v_c$ from the equations, we can get a nonholonomic constraint:

$$\dot{x}_g \sin\theta - \dot{y}_g \cos\theta + d\dot{\theta} = 0$$

$$\Rightarrow \begin{bmatrix} \sin\theta & -\cos\theta & d \end{bmatrix} \begin{bmatrix} \dot{x}_g \\ \dot{y}_g \\ \dot{\theta} \end{bmatrix} = 0 \qquad (3.17)$$

This relation states that the robot can only move in the direction normal to the axis of the driving wheels as long as mobile robot satisfies the conditions of pure rolling and nonslipping. Therefore, the component of the velocity of the contact point with the ground, orthogonal to the plane of the wheel is zero.

When the center of mass of the platform, the point G, coincides with its center of rotation, the point C, then *d*=0, so nonholonomic constraint will be:

$$\dot{x}_g \sin\theta - \dot{y}_g \cos\theta = 0 \; = 0 \tag{3.18}$$

Combining linear and angular velocities at point G (from equation 3.16) can be written in matrix form,

$$\dot{q} = \begin{bmatrix} \dot{x}_g \\ \dot{y}_g \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & -d\sin\theta \\ \sin\theta & d\cos\theta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_c \\ \omega \end{bmatrix} \tag{3.19}$$

According to equations (3.15) and (3.19), the kinematic model of differential drive two wheeled mobile robot can be explicitly written as:

$$\dot{q} = \begin{bmatrix} \dot{x}_g \\ \dot{y}_g \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{r}{2}\cos\theta - \frac{rd}{2R}\sin\theta & \frac{r}{2}\cos\theta + \frac{rd}{2R}\sin\theta \\ \frac{r}{2}\sin\theta + \frac{rd}{2R}\cos\theta & \frac{r}{2}\sin\theta - \frac{rd}{2R}\cos\theta \\ \frac{r}{2R} & -\frac{r}{2R} \end{bmatrix} \begin{bmatrix} \omega_R \\ \omega_L \end{bmatrix} \tag{3.20}$$

It is easy to observe that the robot motion has three degrees-of-freedom (3DOF) while the existing number of controllable degrees-of-freedom is only 2DOF.

## 3.9 Dynamic Model of Mobile Robot:

The simplified version of the dynamic model used in for differential driven mobile robot. In this simplified model, the mass and the moment of inertia of the two wheels are considered to be negligible compared to those of the robot chassis. Assuming the total mass of mobile robot as '*m*' and the moment of inertia as '*I*' due to angular velocity '$\omega$' round the center of mass, we can derive the equation of translational kinetic energy of mobile robot as a rigid body as: $K = \frac{1}{2} m v_g^2 + \frac{1}{2} I \omega^2$ (3.21)

By Lagrangian equations of motion, we can derive the linear force applied to the robot chassis and the angular torque exerted on the mobile robot as a whole due to the dynamic torques of two motors, $\tau_{dR}$ and $\tau_{dL}$ respectively.

The Euler–Lagrange equations of motion are used to derive the dynamics of the mobile robot:

Linear Force applied on the mobile robot by the velocities and inertia of wheels,

$$u_1 = \frac{d}{dt}\left(\frac{\partial K}{\partial \dot{x}}\right) - \frac{\partial K}{\partial x} = m\dot{v}_c \tag{3.22}$$

Angular Torque exerted on the mobile robot by the velocities and inertia of wheels,

$$u_2 = \frac{d}{dt}\left(\frac{\partial K}{\partial \dot{\theta}}\right) - \frac{\partial K}{\partial \theta} = I\ddot{\theta} + md^2\ddot{\theta} \tag{3.23}$$

For non-circular orbits or trajectories, only the component of gravitational force directed orthogonal to the path is termed centripetal. Centripetal force is a force that makes a body follows a curved path: it is always directed orthogonal to the velocity of the body, toward the instantaneous center of curvature of the path. The direction of the force is toward the center of the circle in which the object is moving, or the osculating circle, the circle that best fits the local path of the object, if the path is not circular.

When a two wheeled mobile robot moves along a path, it will tend to move on a straight line, due to its inertia. However, if it comes to a curve in the path, the mobile robot has to change the velocities of wheels to follow the direction of curve. The friction between the wheels and the path create a force that is perpendicular to the direction of motion. That friction force is the centripetal force, causing the robot to go on a curved path.

The magnitude of the centripetal force on mobile robot of mass *m* moving at angular velocity ω along a path with radius of curvature d is: $F_c = md\omega^2$ towards center of mass.

The linear force applied on the robot chassis and the centripetal force are in reverse direction with respect to one another, so the magnitude of dynamic force $F_d$, responsible for forward movement, can be written as:

$$F_d = u_1 - F_c$$

$$\Rightarrow F_d = m\dot{v}_c - md\dot{\theta}^2 \qquad (3.24)$$

The Coriolis Effect exists only when one uses a rotating reference frame. Here, Local Reference frame (robot chassis frame) can rotate at an angular displacement $\theta$ with the inertial reference frame. In the rotating frame, Coriolis force behaves exactly like a real force (that is to say, it causes acceleration and has real effects) and can produce torque component also. However, it is a consequence of inertia, and is not attributable to an identifiable originating body. The Coriolis force acts in a direction perpendicular to the rotation axis and to the velocity of the body in the rotating frame. It is proportional to the object's speed in the rotating frame and rate of rotation (angular velocity). Coriolis force arises from two sources of change in velocity that result from rotation:

First is the change of the velocity of robot chassis in time: The same velocity will be seen as different velocities at different times in a rotating frame of reference. In this case, apparent acceleration is proportional to the angular velocity ($\omega$) of the reference frame (the rate at which the local reference coordinate axes change direction), and to the component of velocity ( $v_c$ ) of the object in a plane perpendicular to the axis of rotation.

The second is the change of velocity in space: Different positions in a rotating frame of reference have different velocities (as seen from an inertial frame of reference). In order for mobile robot to move in a straight line it must therefore be accelerated so that its velocity changes from point to point by the same amount as the velocities of the frame of reference. The effect is proportional to the angular velocity ($\omega$) (which determines the relative speed of two different points in the rotating local frame of reference), and to the component of the velocity ($v_c$) of the object in a plane perpendicular to the axis of rotation (which determines how quickly it moves between those points).

Both the above cases give coriolis force at center of mass of robot chassis as:

$$F_{cor} = -2mv_c\omega$$

The contribution to the total torque on robot chassis by coriolis force can be written as:

$$\tau_{corr} = -2mv_c\omega \times \frac{d}{2} = -mv_c\omega d \tag{3.25}$$

Here, the center point of wheel axis is shifted by distance '*d*' to center of mass of the whole robot structure. So, the torque developed due to the displacement from *0* to *d*. Therefore average displacement will be $\frac{d}{2}$.

The coriolis component of torque is in reverse direction with respect to the direction of angular Torque exerted on the mobile robot by the velocities and inertia of wheels, so the total dynamic torque applied on the robot to make rotational movement can be derived as:

$$\tau_d = u_2 - \tau_{corr}$$

$$=> \tau_d = (I + md^2)\ddot{\theta} + mdv_c\dot{\theta} \qquad (3.26)$$

On the other hand total force and torque are generated by the dynamic torques of the two d.c geared motors, $\tau_{dR}$ and $\tau_{dL}$ respectively:

$$\left.\begin{aligned} F_d &= \frac{1}{r}(\tau_{dR} + \tau_{dL}) \\ \tau_d &= \frac{R}{r}(\tau_{dR} - \tau_{dL}) \end{aligned}\right\} \qquad (3.27)$$

Taking into account the relations (3.24), (3.26) and (3.27) the dynamic model of WMR is represented by the matrix form:

$$M(q).\ddot{q} + C(q,\dot{q})\dot{q} = B.\tau \qquad (3.28)$$

Where, M(q) is a symmetric, positive definite inertia matrix assembled from the individual axle module inertia matrices,

$$M(q) = \begin{bmatrix} m & 0 \\ 0 & (I + md^2) \end{bmatrix}$$

$C(q,\dot{q})$ matrix is combination of Centripetal force and Coriolis component of torque,

$$C(q,\dot{q}) = \begin{bmatrix} -md\dot{\theta}^2 \\ mV_c d \end{bmatrix}$$

B is input transformation matrix, $B = \begin{bmatrix} \frac{1}{r} & \frac{1}{r} \\ \frac{R}{r} & -\frac{R}{r} \end{bmatrix}$

$\tau$ is input dynamic torque matrix, $\tau = \begin{bmatrix} \tau_{dR} & \tau_{dL} \end{bmatrix}^T$

Equation (3.28) assumes that gravitational force component is zero as the trajectory of the mobile base is constrained to horizontal plane and no surface friction presents during movement.

## 3.10 Lyapunov based Tracking Control:

The Lyapunov control is one of the design methods of a feedback controller of nonlinear systems; by setting a positive-definite function (Lyapunov function) which is minimized at the desired point and multiplying the gradient vector of the function by a symmetric positive-definite tensor, the control input is designed. When the Lyapunov control is applied to a nonholonomic system, the controlled system has equilibrium points beside the desired point and may stop at these points.

### 3.10.1 Tracking problem:

Consider model mobile robot is tracking a target at distance of D from its current position($x$, $y$, $\theta$) with velocities ($v$, $\omega$). At target point robot position will be ($x_t$, $y_t$, $\theta_t$) with velocities ($v_t$, $\omega_t$) in its global reference frame. In the local reference coordinates with respect to the body of the mobile robot, the configuration error between two position of the robot model

$P_e = (x_e, y_e, \theta_e)^T$ can be represented in terms of transformation matrix or orthogonal rotational matrix

$$\begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_t - x \\ y_t - y \\ \theta_t - \theta \end{bmatrix} \tag{3.29}$$

The time derivative of the configuration error of model mobile robot can be deduced from (3.29):

$$\begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\theta}_e \end{bmatrix} = \begin{bmatrix} \omega y_e - v + v_t \cos\theta_e \\ -\omega x_e + v_t \sin\theta_e \\ \omega_t - \omega \end{bmatrix} \tag{3.30}$$

Form the discussion above, the tracking control problem based on the kinematic model of the mobile robot can be stated as: with random initial configuration errors existing, find a bounded feedback control law for linear and angular velocities which can make system (3.30) satisfy with the condition that $(x_e, y_e, \theta_e)^T$ is bounded and $\|(x_e, y_e, \theta_e)^T\|$ limits to be zero when the time t approaches infinity.

### 3.10.2 Designing Control law for solving problem:

The trajectory tracking control model (3.30) is an under actuated nonlinear system, which can use the integral backstepping to design the control law. The integral backstepping decomposes the complex nonlinear system into several subsystems which number cannot exceed the system's order and connects the state variables of each subsystem to a virtual stable control system with a known Lyapunov function by defining some new virtual control variables such that a control law to stabilize the original controlled system is obtained.

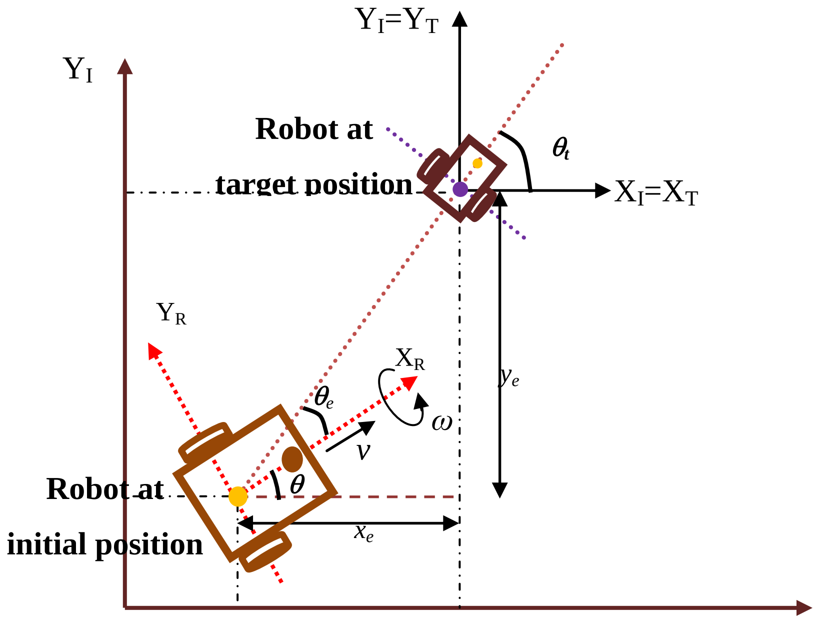



**Figure 3.7: Tracking of Mobile Robot towrads a specific target**

According to the system model (3.30), let the configuration error $x_e$ be the virtual controlled variable and a new virtual error variable can be defined as follows:

$$\bar{x}_e = x_e - asign(\omega) y_e \qquad (3.31)$$

Where $a>0$, $asign(\omega)$ is the virtual feedback. Signum Function sign (.) can be defined as

$$sign(x) = \begin{cases} 1 & if x \geq 0 \\ -1 & if x \leq 0 \end{cases}$$

If the control law for (3.31) can make the virtual controlled variable $x_e$ approach $asign(\omega) y_e$ and $\theta_e$ zero, the equation $\dot{y}_e = -a|\omega| y_e$ will be satisfied according to the kinematic model of the system. Due to the condition $a|\omega| > 0$ is always satisfied when $\omega \neq 0$, the controllable variable $y_e$ will converge exponentially to zero with $t$ approaching infinity. On condition that $y_e$ is zero, from the discussion above we can conclude that $x_e$ will also converge to be zero and consequently the configuration error $P_e$ will completely be zero. Therefore, the expected control task is fully accomplished. The trajectory tracking control task results in finding a control law to make $\overline{x}_e$=0 and $\theta_e$=0 when time approached infinity. If the condition $\omega=0$ is taken into consideration, $\dot{y}_e$=0 can only be achieved while $y_e$=0 cannot be obtained in terms of the system's model. Therefore, the design of the controller is to find the suitable controller $Q=(v,\omega)^T$, which can clearly guarantee $x_e$=0, $y_e$=0 and $\theta_e$=0 for arbitrary selected $\omega$ when time $t$ approaches infinity.

A Lyapunov function can be selected as follows:

$$V=\frac{1}{2}\overline{x}_e^{\,2}+\frac{1}{2}y_e^{\,2}+(1-\cos\theta_e) \qquad (3.32)$$

$V\geq 0$ is always satisfied and $V$=0 holds true if and only if $(\overline{x}_e, y_e, \theta_e)^T=0$ . Here, $\theta_e$ is limited in (-π, π), which can meet the various requirements in practice.

Combining (3.30) and (3.31), the derivative of (3.32) can be described as

$$\dot{V}=\overline{x}_e\dot{\overline{x}}_e+y_e\dot{y}_e+\sin\theta_e\dot{\theta}_e$$

$$=\overline{x}_e\dot{\overline{x}}_e+y_e(-\omega(\overline{x}_e+asign(\omega)y_e+v_t\sin\theta_e)+\sin\theta_e\dot{\theta}_e$$

$$=\overline{x}_e(\dot{\overline{x}}_e-\omega y_e)-a|\omega|y_e^{\,2}+\sin\theta_e(v_t y_e+\dot{\theta}_e)$$

$$=\overline{x}_e(-v+v_t\cos\theta_e+a|\omega|x_e-asign(\omega)v_t\sin\theta_e)-a|\omega|y_e^{\,2}+\sin\theta_e(v_t y_e+\omega_t-\omega) \quad (3.33)$$

Suppose $v_t$, $\omega_t$ are bounded and $v_t$ is never selected to be zero for any time. The system control law can be defined as follows:

$$\left.\begin{aligned} v&=v_t\cos\theta_e+a|\omega|x_e-asign(\omega)v_t\sin\theta_e+k_1(x_e-asign(\omega)y_e) \\ \omega&=v_t y_e+\omega_t+k_2 v_t\sin\theta_e \end{aligned}\right\} \qquad (3.34)$$

where $k_1$>0 and $k_2$>0. Then (3.32) turns into

$$\dot{V} = -k_1 \bar{x}_e^2 - a\,|\,\omega\,|\,y_e^2 - k_2 v_t \sin^2 \theta_e \tag{3.35}$$

Since $a\,|\,\omega\,|>0$, $k_1$>0 and $k_2$>0, the inequality $\dot{V} \le 0$ holds permanently. With V defined as a continuous positive definite function and meanwhile a bounded function, $\dot{V}$ is a continuous negative semi definite function, according to further analysis through Barbalet lemma [15], $\dot{V}$ will clearly confined to be zero when $t$ approaches infinity. It means that the components $\bar{x}_e^{\,2}(t), a\,|\,\omega\,|\,y_e^{\,2}(t), \sin^2 \theta_e$ in (3.35) will individually go to be zero, i.e., the equation $\lim_{t\to\infty} \bar{x}_e = \lim_{t\to\infty}(x_e - a\,|\,\omega\,|\,y_e) = 0$ and $\lim_{t\to\infty} \theta_e = 0$ are both verified. Since the variable $v_t$ will never equal zero, $\omega$ will not be zero permanently based on equation (3.34). As a result, $a\,|\,\omega\,|\,y_e^{\,2}$ will be zero, so do $y_e$ and $x_e$ when time $t$ reaches infinity. Based on the Lyapunov stabilization principle, system *m*odel under the control law (3.34) is globally asymptotically stable. Furthermore, $(x_e, y_e, \theta_e)^T$ is bounded and $\|(x_e, y_e, \theta_e)^T\|$ limits to be zero when the time $t$ approaches infinity.

Taking the dynamics of the mobile robots into consideration, if the system error is relative large, the control law for linear and angular velocities generated by (3.34) may exceed the maximum of the permitted velocities $(v_{\max}, \omega_{\max})^T$ . Slippage in robot motion occurs under large velocities, which will further the proposed control law. Therefore, we should pay attention on the control strategy to limit velocities and specify the reasonable maximums of the velocities $(v_{\max}, \omega_{\max})^T$ .

From the discussion above, the error variable $y_e$ can converge in an exponential rate. On condition that $y_e$ has been converged, $x_e$ and $\theta_e$ will also converged exponentially. As a whole, the proposed control law for can be considered as a law with approximately exponential convergence rate.

## 3.11 Conclusion:

When developing a robot it is the designer's task to analyze the terrain in which the robot will travel and what the robot has to do there. With the help of developed methodology, the robot can achieve path following as well as velocity tracking, considering both kinematic model and dynamic model of the mobile robot. According to this analysis the robots locomotion mechanism can be chosen. The trajectory tracking problem for a wheeled mobile robot is discussed by using of integral backstepping and introducing a virtual error control variable characterized by a signum function, a variable structure tracking controller is designed for the mobile robots. Lyapunov stability theory demonstrates the global asymptotic stability of the resultant closed-loop control system. The proposed controller has certain advantages over other existing tracking controller on algorithms, parameters selection, tracking performances.

# 4 Study of Reactive Behavioral Controller based on Mamdani-Fuzzy Approach

Uncertainty and ambiguity associated with reactive navigation for autonomous mobile agent in unknown or partially known chaotic surroundings, especially, unpredictably changing environment can be unraveled by making coordination and fusion of the elementary behaviors of mobile agent. The fuzzy navigation technique, which is accomplished to generate satisfactory direction and velocity maneuvers of the autonomous robot, is instigated here for the robot navigation to reach its goal safely moving on unknown static terrains. The Fuzzy logic controller (FLC), a hybrid of different membership functions, has been employed on an experimental mobile robot which uses a set of three multipurpose IR sensors and one ultrasonic sensor to perceive the environment. The fuzzy logic maps the input fuzzy sets representing the mobile robot state space determined by sensor readings to the output fuzzy sets representing mobile robot action space. Action coordination of the robotic behaviors such as following a wall, avoiding an obstacle and running towards goal, have been attained using proposed hybridized fuzzy technique which is found to be proficient and partially optimized for navigation purpose through simulation and experimental authentication.

## 4.1 Introduction

Reactive navigation technique, used for autonomous mobile robot, can be defined as a mapping between sensory data and commands in unstructured or partially unknown environment without continuous human assistance. The fuzzy logic has already been proved as effective method to map the input fuzzy sets representing the mobile robot state space determined by sensor readings to the output fuzzy sets representing mobile robot action space [93]. As uncertain and imprecise information are inherent to the perception of the environment through the robot sensors, fuzzy logic is an appropriate technique for providing a scientific formalism for reasoning and decision making [32] towards a satisfactory behavioral performance during navigation.

Recent research work has illustrated main advantage of the fuzzy based evolutionary navigation scheme over most other techniques (such as potential field method, vector field histogram and local navigation etc.) are that less local information is required for this fastest algorithm. Fuzzy logic can be used to implement individual behaviors, to coordinate the various behaviors, to select roles for each robot, and for robot perception, decision-making, and speed control [79]. Fuzzy behavior-based architecture for mobile robot navigation in unknown environments incorporates design of basic behaviors for mobile robot navigation: goal seeking behavior, obstacle avoidance behavior, following behavior, and deadlock disarming behavior [44, 87]. Each behavior is implemented by using fuzzy controller to achieve respective navigation task.

This research focuses proper evaluation of robot wheel velocities from sensory information based fuzzy logic frame work which has been implemented in model mobile robot for achieving integration of different robotic behaviors and generating reasonable trajectories towards the target under various situations. The fuzzy rules control the steering of the robot according to whether there are obstacles or targets around it and how far they are from it. The comparisons between recommended method and previously designed methods [1, 71] are resolved that the recent method can be fruitfully hired for acute navigation of mobile robot. This hybrid fuzzy controller of mobile robot for path analysis and planning has also been substantiated by experimental verification.

This chapter is systematized into six sections following the introduction; the behavioral strategy of mobile robot is described in section 4.2. Hybridized fuzzy control architecture based on Mamdani fuzzy approach has been pronounced stepwise with a small example in section 4.3. The simulation results by the present navigation technique and their comparisons with other techniques already developed by other researches are conferred in section 4.4 and in section 4.5 experimental results are verified with simulation to make evident the supremacy of the anticipated approach. Finally assessment is argued in section 4.6.

## 4.2 Reactive Behavioural Control Strategy:

Reactive (behaviour-based) navigation strategy was developed by Brooks [15]. These approaches generate control commands based on current sensory information. To take actions, the robot uses the local model of environment without planning process. Therefore, it is not necessary to build a complete model of environment. Bottom-up approach for decision making is used in the behaviour based architectures in which high level constraints are not integrated in action generation process. Reactive navigation has a quick response in the dynamic and unknown environment. Figure 4.1 represents the overall architecture of behaviour-based approaches. In first layer, robot gathers sensory information. Then a transfer function called behaviour receives particular sensory inputs perception and transforms them into the predefined response. Finally, the robot executes an action based on the output of active behaviour. In fact, complex navigation tasks are broken down into several simpler and smaller sub-level tasks which improve the total performance of the navigation system.

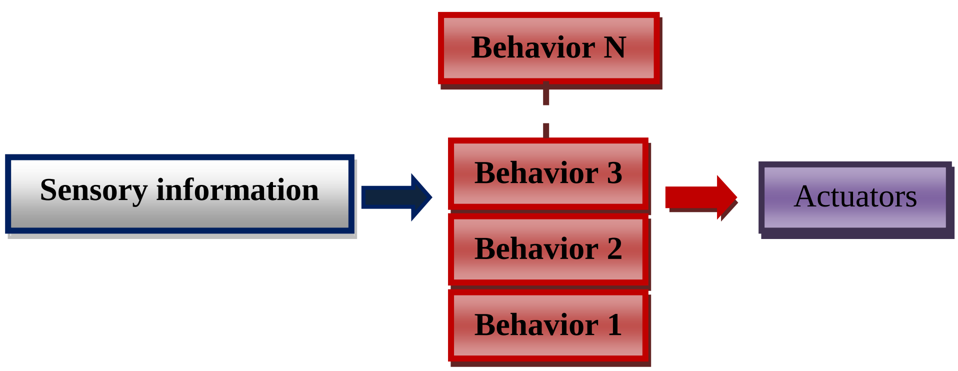



**Figure 4.1: Behavior- based overall architecture**

The particular basic behaviour-based control architecture used here is Subsumption control architecture which was introduced by brooks [15] at Massachusetts Institute of Technology *(MIT)*. It is composed of several layers of task-achieving behaviors where each behavior can receive sensory information for a given task (obstacle avoidance, wall following, target seeking, etc.). In subsumption architecture, the planning module is eliminated from the control architecture and the focus is exclusively on the sensing and acting modules. The behaviors provide a direct coupling between sensory inputs and robot's actions. As Figure 4.2 shows, in subsumption architecture, behaviors are layered and each layer receives particular sensory information. Coordination of behavior layers refers to the priority-based arbitration. Priority-based arbitration is a process of deciding which behavior to be active when multiple conflicting behaviors are triggered. Therefore, the highest active behavioral module generates the overall output of architecture.

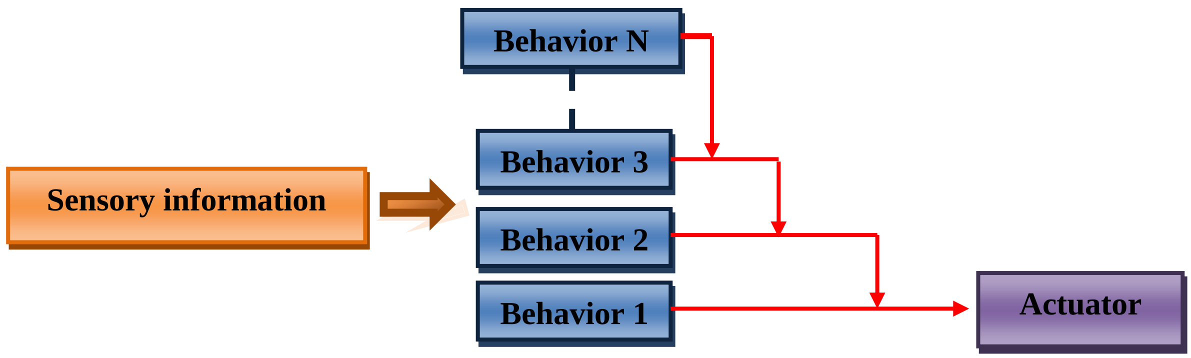



**Figure 4.2: Subsumption architecture**

The overall Advantages of behavior-based navigation systems are:

- Their ability to build a navigation system in an incremental way of layer upon layer.
- Their quick reaction to the unknown and dynamic environment.
- They do not require modeling and storing the whole model of the environment.
- There is less computation and shorter delay between perception and action.
- And they are more robust and reliable which means in case of a behavior unit failure, the other units continue the tasks.

The drawbacks of behavior-based control are as follows:

- Difficulty in coordination among the behaviors,
- The interaction between the system and environment is difficult and less predictable,

- Behaviors are low level so they do not reflect high level tasks,
- Lack of planning module could be not appropriate for some complicated tasks.

## 4.3 Hybridized Fuzzy Control Architecture:

Humans use perceptions of time, distance, speed, shape, and other attributes of physical and mental objects in day today life [111]. Perceptions are described by propositions drawn from a natural language, in which the boundaries of perceived classes are fuzzy. Fuzzy logic provides a formal methodology for representing and implementing the human expert's heuristic knowledge and perception-based actions. Using the fuzzy logic framework, the attributes of human reasoning and decision making can be formulated by a set of simple and intuitive IF (antecedent)—THEN (consequent) rules, coupled with easily understandable and natural linguistic representations [85]. Fuzzy logic expressed operational laws in linguistics terms instead of mathematical equations.

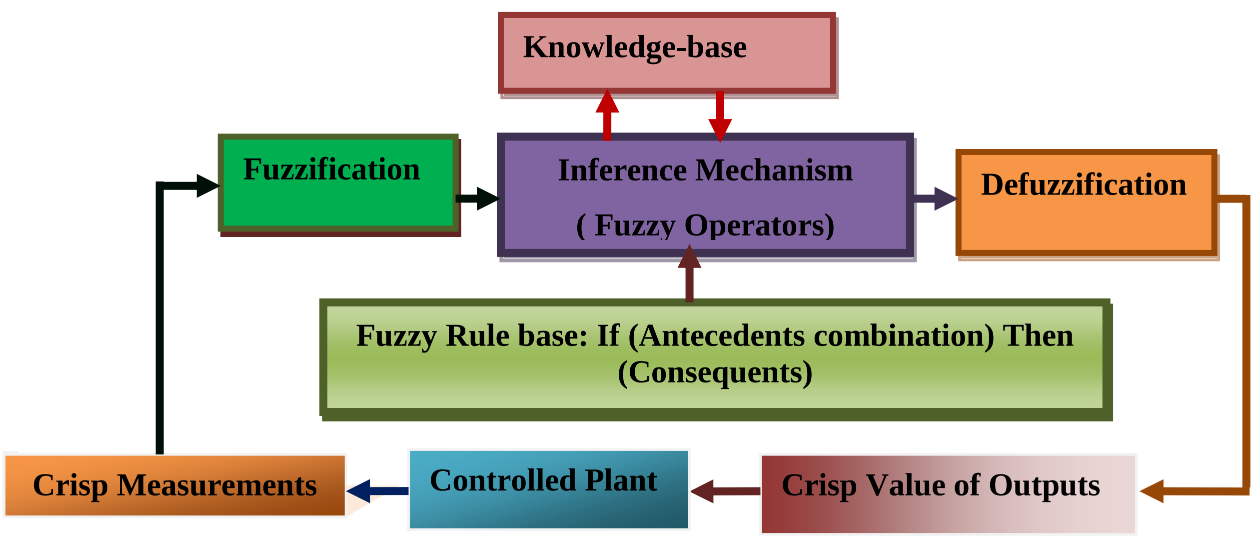



**Figure 4.3: Fuzzy Logic Controller (Mamdani Approach)**

An Intellectual Fuzzy controller for mobile robot empowers the robot to escape the obstacle and expand target seeking ability. Each robot has an array of infrared sensors for quantifying the distances of obstacles nearby it, and one ultrasonic sensor for assessing the target angle. The input signals of fuzzy controller are characterized by membership function and are labeled by linguistic variables. According to the data assimilated by the robots using their sensors, some of the fuzzy control rules are activated accordingly. The outputs of the activated rules are combined and defuzzified to get the velocities of the driving wheels of the robots.

### 4.3.1 Hybridization of fuzzy membership functions for robot controller:

The inputs to the proposed fuzzy control scheme consist of the distances between a robot and the obstacles to the left, front and right locations and heading angle of robot to the target, acquired by sensors, termed as front obstacle distance (FD), left obstacle distance (LD), right obstacle distance (RD) and detecting the bearing of target (HA). The outputs from the control scheme are commands for the speed control unit of two side wheels of the mobile robot denoted as Left wheel velocity (LV) and right wheel velocity (RV) respectively (Figure 4.4). According to the acquired range information by sensors, reactive behaviors are weighted by the fuzzy logic algorithm to control the velocities of the two driving wheels of the robot. The control system combines a repelling influence related to the distance between robots and nearby obstacles and with an attracting influence between the robots and targets [84].

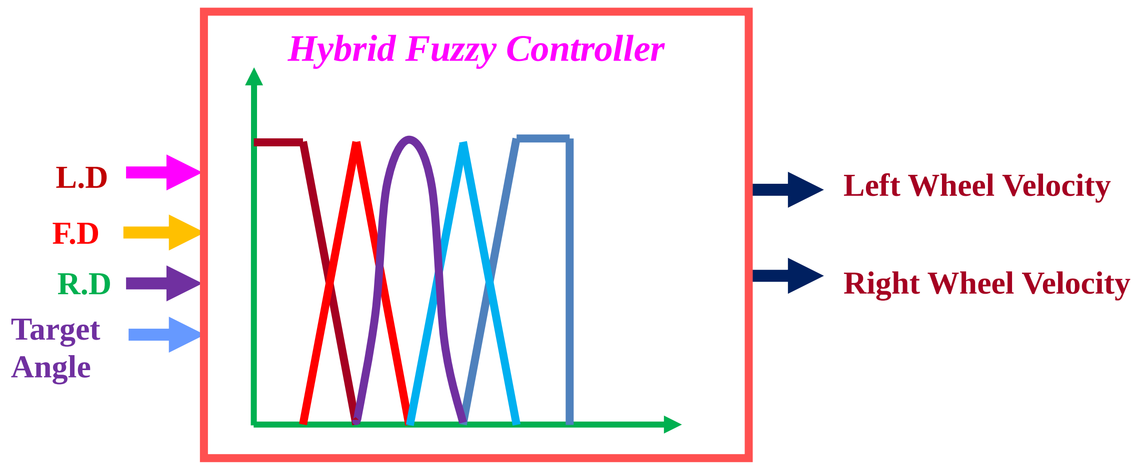



**Figure 4.4: Hybrid Fuzzy Controller embedded with Integration of Different Membership Functions for Mobile Robot Navigation**

In this research three types of membership functions (Trapezoidal, Triangular and Gaussian) are hybridized in a single controller. For five membership function, first and fifth one are taken as trapezoidal, second and fourth one are as triangular and the third one is Gaussian.

Linguistic variables like “very near”, “near”, “medium”, “far” and “very far” are considered for Left, Right and Front Obstacle distances during navigation of mobile robot. When the target is located at the left side of the mobile robot, the target angle is negative and if the target is at right side of robot, the angle defined as the term “no target consider” is used if there is no target in the environment. “more pos” (more positive),“pos” (positive),“zero”, “neg” (negative) and “more neg” (more negative) are defined for the bearing of heading angle (HA) with respect to target.

Linguistic variables like “very slow”, “slow”, “medium”, “fast” and “very fast” are considered for left wheel velocity and right wheel velocity.

The parameters defining the functions are listed in table 4.1. Values of parameters are decided empirically. The membership functions described above are shown in Figure 4.4.

**Table 4.1: Parameters of fuzzy membership functions:**

(a) Parameters for Left, Right and Front Obstacle Distance

| Variables (MF) | Parameters in meter | | |
|---|---|---|---|
| Very Near (Trapezoidal) | 0.0 | 1.2 | 2.4 |
| Near (Triangular) | 1.2 | 2.4 | 3.6 |
| Medium (Gaussian) | 2.4 | 3.6 | 4.8 |
| Far (Triangular) | 3.6 | 4.8 | 6.0 |
| Very Far (Trapezoidal) | 4.8 | 6.0 | 7.2 |

(b) Parameters for Heading Angle

| Variables (MF) | Parameters in degree | | |
|---|---|---|---|
| **More negative (Trapezoidal)** | -180 | -120 | -60 |
| **Negative (Triangular)** | -120 | -60 | 0 |
| **Zero (Gaussian)** | -60 | 0 | 60 |
| **Positive(Triangular)** | 0 | 60 | 120 |
| **More Positive (Trapezoidal)** | 60 | 120 | 180 |

(c) Parameters for Left and Right Velocity

| Variables (MF) | Parameters in meter | | |
|---|---|---|---|
| **Very Slow (Trapezoidal)** | 0.0 | 0.6 | 1.2 |
| **Slow (Triangular)** | 0.6 | 1.2 | 1.8 |
| **Medium (Gaussian)** | 1.2 | 1.8 | 2.4 |
| **Fast (Triangular)** | 1.8 | 2.4 | 3.0 |
| **Very Fast (Trapezoidal)** | 2.4 | 3.0 | 3.6 |

### 4.3.2 Fuzzy Rule Base Mechanism:

Fuzzy rules are formulated based on human perception. The fuzzy rule base is a set of linguistic rules in the form of 'if a set of conditions are satisfied, then a set of consequences are inferred'. Based on the above fuzzy subsets, the fuzzy control rules are defined in a general form for four inputs and two outputs fuzzy system as follows [79]:

If (matching degree of LD is $\mu(LD_i)$ and matching degree of FD is $\mu(FD_j)$ and matching degree of RD is $\mu(RD_k)$ and matching degree of HA is $\mu(HA_m)$, Then (matching degree of LV is $\mu(LV_{ijkm})$ and matching degree of RV is $\mu(RV_{ijkm})$. (4.1)

where i = 1 to 5, j = 1 to 5, k = 1 to 5 and m =1 to 5 because LD, FD, RD and HA have five membership functions each.

The matching degree of final output is computed by the following formula:

$$\text{Matching degree } \mu_{LV, RV}(vel_{ijkm}) = \min\{\mu(LD_i), \mu(FD_j), \mu(RD_k) \text{ and } \mu(HA_m)\} \qquad (4.2)$$

When the matching degree=1 the inferred conclusion is identical to the rule's consequent, and if it is zero no conclusion can be inferred from the rule.

Finally, the output firing area of the left and right wheel velocities can be computed by the following formula,

$$\left.\begin{aligned} \mu_{LV}(vel) &= \max\{\mu_{LV}(vel_{1111}), \ldots\ldots\mu(vel_{ijkm}), \ldots\ldots\mu(vel_{5555})\} \\ \mu_{RV}(vel) &= \max\{\mu_{RV}(vel_{1111}), \ldots\ldots\mu(vel_{ijkm}), \ldots\ldots\mu(vel_{5555})\} \end{aligned}\right\} \quad (4.3)$$

The final output (crisp value) of the fuzzy logic controller of left and right wheel velocities can be calculated by "Centre of Gravity" method [78],

$$\left.\begin{aligned} \text{Left Velocity} = LV &= \frac{\int vel \cdot \mu_{LV}(vel) \cdot d(vel)}{\int \mu_{LV}(vel) \cdot d(vel)} \\ \text{Right Velocity} = RV &= \frac{\int vel \cdot \mu_{RV}(vel) \cdot d(vel)}{\int \mu_{RV}(vel) \cdot d(vel)} \end{aligned}\right\} \qquad (4.4)$$

### 4.3.3 Inference Mechanism for Robotic Behaviors:

Behavior in a mobile robot navigation system usually signifies a concern of the robot such as seeking the target or avoid obstacle or follow a shortest route. Behavior-based inference mechanism made of fuzzy logic rules show that the robot mainly adjusts its motion direction and moves towards the target. All the Fuzzy Rules have been obtained heuristically using common sense of human intelligence. The velocities are found from those rules, based on five membership function, described in Table 4.2, 4.3 and 4.4. The rule based inference system has been decomposed into subsystems for achieving different behavioral responsibilities. To reach a specified target in a complex environment, the mobile robot needs at least the following reactive behaviors those has to be weighted to determine an appropriate control action:

#### 4.3.3.1 Obstacle Avoidance:

The distance between the robots and obstacles act as repulsive forces for avoiding the obstacles. When the robot is very close to an obstacle, the robot must change its speed and heading angle to avoid the obstacle. When the readings from any sensor are less than the minimum threshold values, the robot determines an object is close, and then obstacle avoidance behavior is activated. Collision avoidance has the highest priority, so, it can override the other behaviors.

Some rules mentioned in Table 4.2 for five-membership function, cater for extreme conditions when the obstacles have to be avoided as quickly as possible. In Table 4.2, rule 31 contains the left obstacle distance as “ Very far”, right obstacle distance as “Near”, front obstacle distance as “Very near” and no target is located around the robot, then the robot should turn to left side to avoid collision with the obstacle in front and towards right of it. For the above condition the right wheel velocity should increase very fast and left wheel velocity should decrease very slowly. The Simulation result of static obstacle avoidance has been exhibited in Figure 4.6.

#### 4.3.3.2 Wall Following:

In the absence of wall following behavior in corporation with obstacle avoidance behavior the robot is inexpert of reaching the goal position when it en-counts U shaped or dead end obstacles on their path. When the robot is moving to a specified target through a narrow channel, or escaping from a U shaped obstacle the robot should keep on heading towards the goal position, but the robot also comes closer to the obstacles. Initially, robot runs directly towards target as obstacles are sensed far away from it. But if it senses obstacles at the front it will make a left or right turn to avoid it.

**Table 4.2: List of Rules for Obstacle Avoidance based on five membership functions**

| Fuzzy Rule No. | Action | LD | FD | RD | HA | LV | RV |
|---|---|---|---|---|---|---|---|
| **1** | OA | Very Near | Very Near | Very Near | Notarget_considered | Very Slow | Very Slow |
| **2** | OA | Very Near | Very Near | Near | Notarget_considered | Slow | Very Slow |
| **3** | OA | Very Near | Very Near | Medium | Notarget_considered | Fast | Very Slow |
| **4** | OA | Very Near | Very Near | Far | Notarget_considered | Very Fast | Very Slow |
| **5** | OA | Very Near | Very Near | Very Far | Notarget_considered | Very Fast | Very Slow |
| **6** | OA | Very Near | Near | Near | Notarget_considered | Medium | Very Slow |
| **7** | OA | Very Near | Near | Medium | Notarget_considered | Fast | Very Slow |
| **8** | OA | Very Near | Medium | Far | Notarget_considered | Fast | Slow |
| **9** | OA | Very Near | Near | Very Far | Notarget_considered | Very Fast | Very Slow |
| **10** | OA | Very Near | Very Far | Far | Notarget_considered | Very Fast | Fast |
| **12** | OA | Very Near | Medium | Very Far | Notarget_considered | Very Fast | Slow |
| **13** | OA | Very Near | Near | Far | Notarget_considered | Fast | Slow |
| **14** | OA | Near | Near | Very Far | Notarget_considered | Very Fast | Slow |
| **15** | OA | Near | Near | Medium | Notarget_considered | Slow | Slow |
| **16** | OA | Near | Near | Far | Notarget_considered | Med | Med |
| **17** | OA | Near | Medium | Very Far | Notarget_considered | Very fast | Very Slow |
| **18** | OA | Near | Far | Medium | Notarget_considered | Med | Slow |
| **19** | OA | Near | Medium | Far | Notarget_considered | Fast | Med |
| **20** | OA | Medium | Near | Near | Notarget_considered | Slow | Fast |
| **21** | OA | Medium | Near | Far | Notarget_considered | Slow | Med |
| **22** | OA | Medium | Far | Near | Notarget_considered | Med | Slow |
| **23** | OA | Medium | Medium | Near | Notarget_considered | Slow | Fast |
| **24** | OA | Medium | Medium | Very Far | Notarget_considered | Very Fast | Medium |
| **25** | OA | Medium | Very Near | Far | Notarget_considered | Very Fast | Slow |
| **26** | OA | Far | Near | Near | Notarget_considered | Slow | Med |
| **27** | OA | Far | Near | Medium | Notarget_considered | Med | Fast |
| **28** | OA | Far | Medium | Near | Notarget_considered | Slow | Fast |
| **29** | OA | Far | Medium | Medium | Notarget_considered | Slow | Med |
| **30** | OA | Far | Near | Very Far | Notarget_considered | Very Fast | Slow |
| **31** | OA | Very Far | Very Near | Near | Notarget_considered | Very Slow | Very Fast |
| **32** | OA | Very Far | Medium | Far | Notarget_considered | Med | Very Fast |
| **33** | OA | Very Far | Far | Medium | Notarget_considered | Fast | Very Fast |
| **34** | OA | Very Far | Near | Far | Notarget_considered | Slow | Very Fast |
| **35** | OA | Very Far | Medium | Near | Notarget_considered | Slow | Very Fast |

If target is at right side of it, the behavior of the approaching target tries to make it turn to the right and target orientation increases gradually. Even at the right side also, robot is facing obstacle in the form of wall, then it will try to avoid it by making a left turn. Due to the nature of achieving target using shortest path, robot will again turn to the right to make itself target oriented. Thus it will be trapped in an indefinite loop. To avoid tis loop, the robot must have the wall following behavior

In Table 4.3, some fuzzy rules show that the robot shall follow wall or an edge of an obstacle when the obstacle is very close to the right or left of the robot, and the target also is located to the right or left. Wall following behavior also depends on target orientation from the current position of the robot. But these rules are not considering the target angle. Rule 40 contains the left obstacle distance as " Medium", right obstacle distance as "Medium", front obstacle distance as "Very far" and no target is located around the robot, then the robot should move quickly towards front to avoid collision with the obstacle in left and right of it. For the above condition the right wheel velocity left wheel velocity should be fast to maintain the straight forward direction. The simulation result of wall following has been shown in Figure 4.6 (a) and escaping from dead end obstacle has been shown in Figure 4.6 (b).

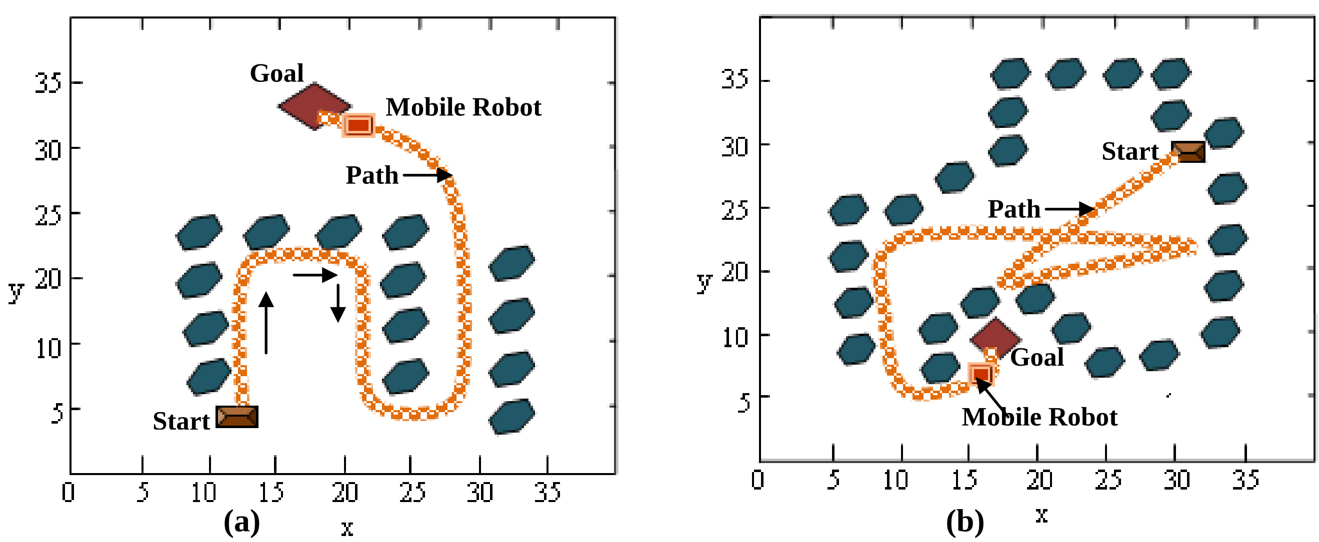



**Figure 4.5 (a): Wall Following Behavior shown by single robot.**

**(b): Escape from dead ends and find the target.**

**Table 4.3: List of Rules for Obstacle Avoidance and Wall Following based on five membership functions**

| Fuzzy Rule No. | Action | LD | FD | RD | HA | LV | RV |
|---|---|---|---|---|---|---|---|
| **36** | OA & WF | Very Near | Near | Very Near | Notarget_considered | Slow | Slow |
| **37** | OA & WF | Very Near | Medium | Very Near | Notarget_considered | Medium | Medium |
| **38** | OA & WF | Very Near | Medium | Near | Notarget_considered | Medium | Slow |
| **39** | OA & WF | Very Near | Far | Very Near | Notarget_considered | Medium | Medium |
| **40** | OA & WF | Medium | Very Far | Medium | Notarget_considered | Fast | Fast |
| **41** | OA & WF | Medium | Far | Medium | Notarget_considered | Medium | Medium |
| **42** | OA & WF | Medium | Far | Near | Notarget_considered | Fast | Fast |
| **43** | OA & WF | Medium | Very Far | Near | Notarget_considered | Very Fast | Very Fast |
| **44** | OA & WF | Near | Very Far | Near | Notarget_considered | Fast | Fast |
| **45** | OA & WF | Near | Far | Medium | Notarget_considered | Fast | Med |
| **46** | OA & WF | Near | Far | Very Near | Notarget_considered | Med | Med |

### 4.3.3.3 Target Seeking:

The attractive force between the robot and the target causes the robot seeking towards the target when the robot is very close to the target. It is used to change the direction of the robot toward the target when there are no obstacles blocking the robot.

Considering rule no 51 in Table 4.4, if the left obstacle is at “Near”, the right obstacle is at “Very Far”, the front obstacle is at “Far” and the robot detects a target located on its right side (or positive side), then the robot should turn right as soon as possible. For approaching target, the right velocity of the robot should be slow and the left velocity should be very fast.

**Table 4.4: List of Rules for Target seeking based on five membership function**

| Fuzzy Rule No. | Action | LD | FD | RD | HA | LV | RV |
|---|---|---|---|---|---|---|---|
| **47** | TS | Very Near | Near | Far | Pos | Very Fast | Slow |
| **48** | TS | Very Near | Far | Near | Zero | Fast | Fast |
| **49** | TS | Very Near | Medium | Very Far | More Pos | Very Fast | Very Slow |
| **50** | TS | Near | Very Far | Very Far | Negative | Very Slow | Medium |
| **51** | TS | Near | Far | Very Far | Pos | Very Fast | Slow |
| **52** | TS | Near | Medium | Very Far | Zero | Medium | Slow |
| **53** | TS | Medium | Far | Near | Negative | Slow | Medium |
| **54** | TS | Medium | Very Near | Far | More Pos | Fast | Very slow |
| **55** | TS | Medium | Near | Far | Negative | Very Slow | Medium |
| **56** | TS | Very Far | Very Far | Medium | More Neg | Very Slow | Very Fast |
| **57** | TS | Very Far | Very Far | Very Far | Zero | Very Fast | Very Fast |
| **58** | TS | Far | Near | Very Near | More Neg | Slow | Very Fast |
| **59** | TS | Far | Medium | Near | Zero | Medium | Fast |
| **60** | TS | Far | Very Far | Near | Neg | Medium | Very Fast |

*Note:* OA – Obstacle Avoidance, WF – Wall Following, TS – Target Seeking, Med – Medium, Pos – Positive (Right Turn), Neg – Negative (Left Turn)

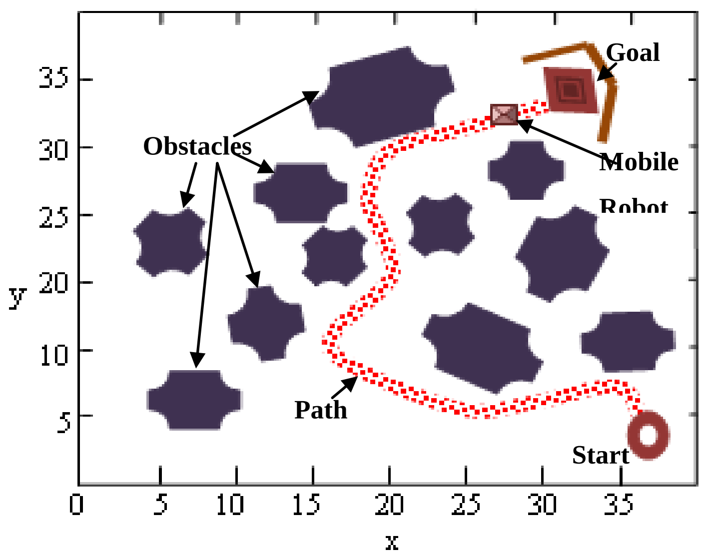



**Figure 4.6: Obstacle Avoidance and Target Seeking Behavior of mobile robot in different simulation environment**

Suppose, for a given scenario (in Figure 4.7), the robot detects left, front and right obstacle distances are 1.37m, 3.12m and 5.34m respectively. There is a target located at a positive angle of $72^0$ from the current position of robot. Therefore, heading angle can be taken as "Pos" or "More Pos". With the above-mentioned position of robot, there will be 2 X 2 X 2 X 2 = 16 fuzzy rules activated to control the left wheel velocity and right wheel velocity of the robot. For this environment the fuzzy rules, which are applicable, are given below.

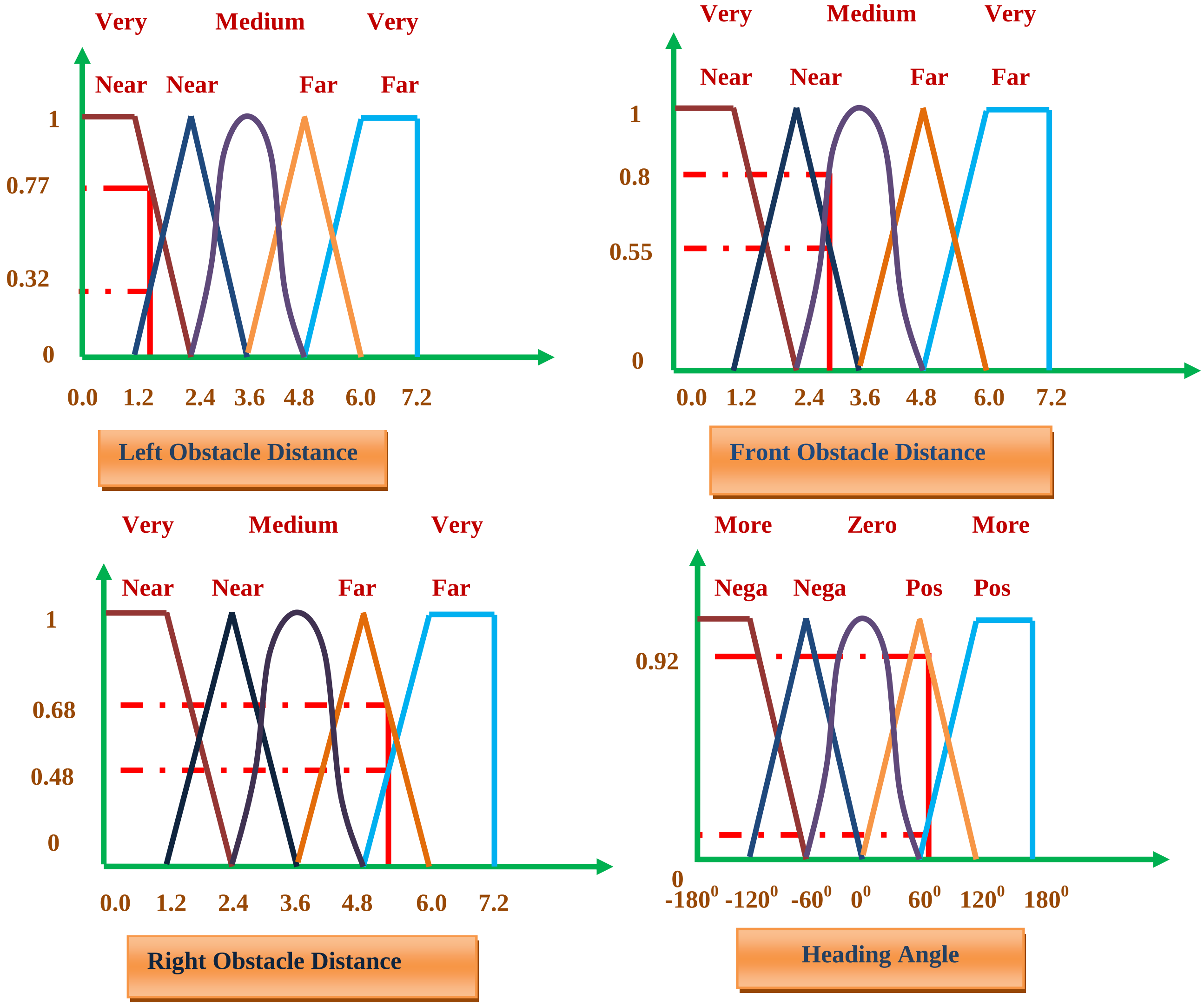



**Figure 4.7: Left, Front, Right Obstacles Distances and Heading angle at the given position of mobile robot**

Degree of membership for Left wheel and Right wheel velocities for each of sixteen rules can be computed by minimum operation between degrees of membership of four inputs for each rule respectively. Again all the output (degree of membership) from each rule can be combined by maximum operation to get the resultant Left wheel velocity and Right wheel velocity (shown in Figure 4.8 (a)) respectively for the given position of the obstacles and target around the mobile robot. The centre of gravity method has been used to get the crisp value of wheel velocities from fuzzified value, which has also been verified with the result derived in MATLAB (Figure 4.8 (b)).

**Rules activated for the present situation of the mobile robot:**

1. If LD is Very Near, FD is Near, RD is Far and HA is “Pos” then LV is Very Fast and RV is Medium.
2. If LD is Very Near, FD is Near, RD is Very Far and HA is “Pos” then LV is Fast and RV is Very Slow.
3. If LD is Very Near, FD is Near, RD is Far and HA is “More Pos” then LV is Very Fast and RV is Slow.
4. If LD is Very Near, FD is Near, RD is Very Far and HA is “More Pos” then LV is Very Fast and RV is Very Slow.
5. If LD is Very Near, FD is Medium, RD is Far and HA is “Pos” then LV is Fast and RV is Slow.
6. If LD is Very Near, FD is Medium, RD is Very Far and HA is “Pos” then LV is Very Fast and RV is Medium.
7. If LD is Very Near, FD is Medium, RD is Far and HA is “More Pos” then LV is Very Fast and RV is Very Slow.
8. If LD is Very Near, FD is Medium, RD is Very Far and HA is “More Pos” then LV is Very Fast and RV is Very Slow.
9. If LD is Near, FD is Near, RD is Very Far and HA is “Pos” then LV is Fast and RV is Slow.

**10.** If LD is Near, FD is Near, RD is Very Far and HA is “Morer Pos” then LV is Fast and RV is Very Slow.

**11.** If LD is Near, FD is Near, RD is Far and HA is “Pos” then LV is Very Fast and RV is Medium.

**12.** If LD is Near, FD is Near, RD is Far and HA is “More Pos” then LV is Very Fast and RV is Slow.

**13.** If LD is Near, FD is Medium, RD is Far and HA is “Pos” then LV is Fast and RV is Very Slow.

**14.** If LD is Near, FD is Medium, RD is Far and HA is “More Pos” then LV is Very Fast and RV is Slow.

**15.** If LD is Near, FD is Medium, RD is Very Far and HA is “Pos” then LV is Very Fast and RV is Medium.

**16.** If LD is Near, FD is Medium, RD is Very Far and HA is “More Pos” then LV is Very Fast and RV is Very Slow.

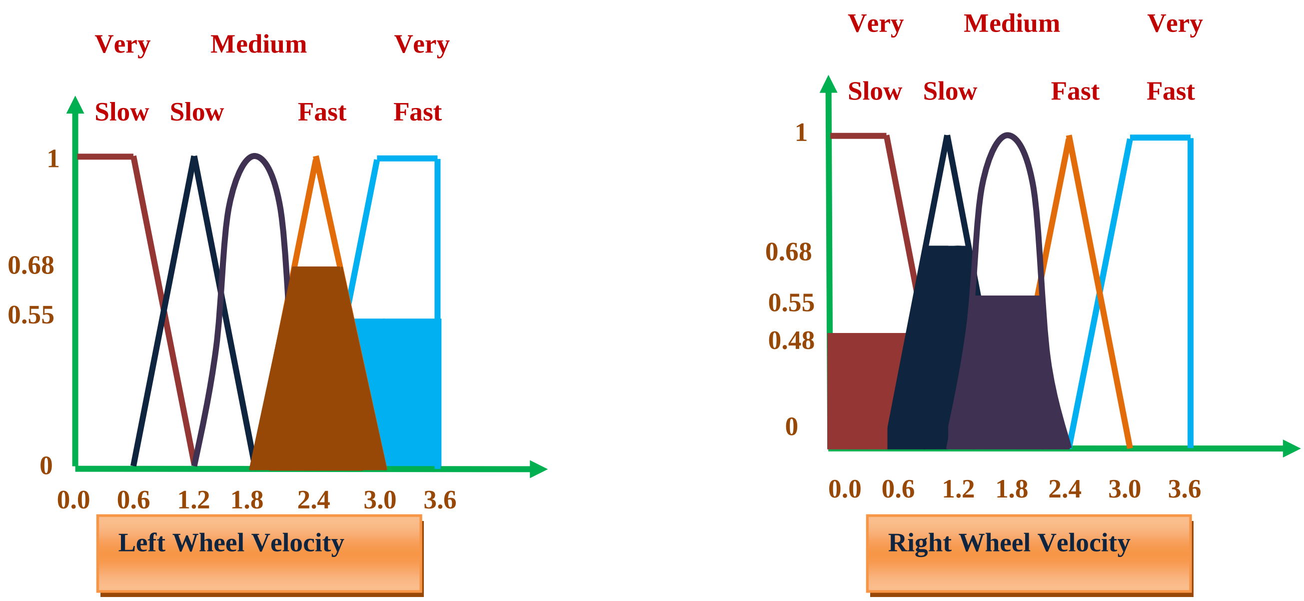



**Figure 4.8(a): Resultant Left and Right Wheel Velocity**


**Figure 4.8(b): Resultant Left Wheel and Right Wheel Velocities in Rule Viewer of MATLAB**

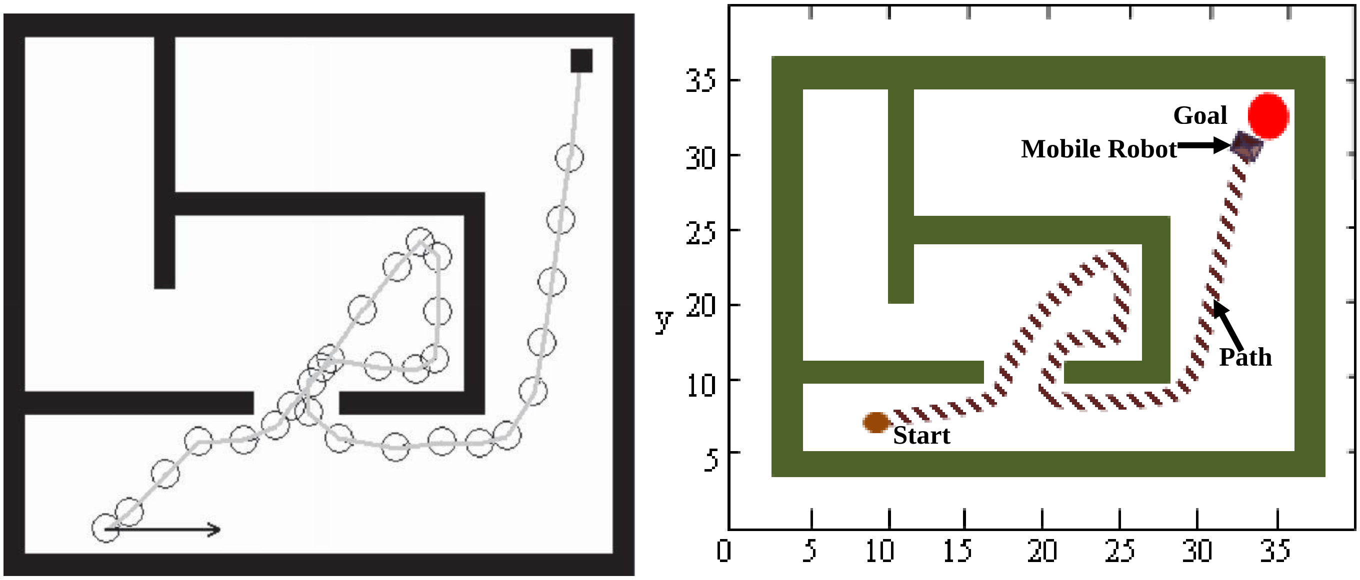
## 4.4 Simulation Results and Comparisons:

The simulation results show that the endorsed method, using information learned by infrared and ultrasonic sensors, can execute robot navigation in complex and uncertain environments. To validate the efficacy and sturdiness of the hybridized five membership function based fuzzy control algorithm, simulation results on mobile robot navigation are exhibited in various environments. Comparison of results have also been done for two different simulation environments; one of them (Figure 4.9(a)) was previously used by Motlagh et al.[71] for behavior-based mobile robot navigation using new minimum avoidance system and another one (Figure 4.10(a)) was shown by Abiyev et al. [1] using fuzzy navigation technique.

The environment generated artificially containing static obstacles as well as static target. In the entire exercises, one robot is located at the starting position, obstacles with different shapes and sizes are positioned in a cluttered manner and one target is present at a fixed point in all scenarios (Figure 4.9 (b), 4.10(b)).

Wall following behaviour of mobile robot, affinity to escape dead end condition using following edge behaviour and target seeking behaviour by maintaining proper orientation towards target throughout the navigation are illustrated in Figure 4.9 (b), so that it may locate, find and reach the specified target using nearly minimum path length. This environment is already used by Motlagh et al. [71] (Figure 4.9 (a)) for avoiding problem of limit cycles in any type of dead-ends encountered on the way to the target by executing a new fuzzy logic algorithm along with actual-virtual target switching strategy.

From the comparison, it can be quantified that the almost same enactment against dead end or local minima problem during navigation can be achieved by proposed hybridized five membership function based simple fuzzy rules, within much reduced time (shown in Table 4.5) than previously proposed new minimum avoidance system by Motlagh et al. [71] (Figure 4.9 (a)).




**(a)** **(b)**

**Figure 4.9: (a) An example of robot path planning in an environment with a dead-end subspace by Motlagh et al.[71] (b) Wall following behaviour of mobile robot during dead end situation by proposed hybridized five membership fuzzy logic algorithms**

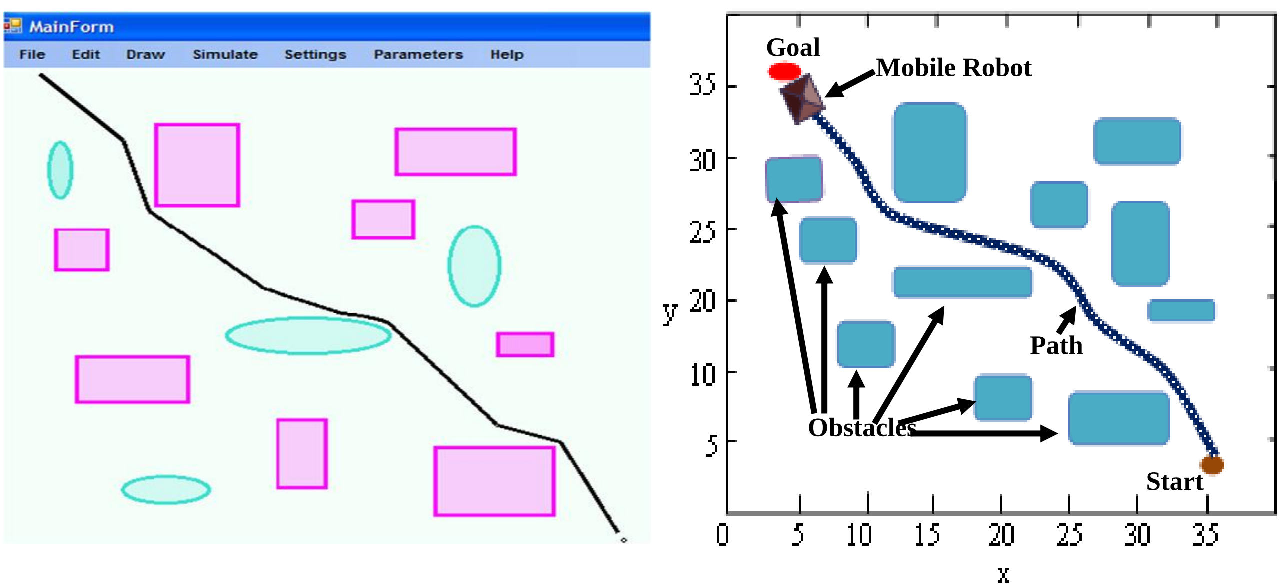



**Figure 4.10: (a) Simulation result of fuzzy navigation algorithm by Abiyev et al.[1] (b) Simulated path achieved by mobile robot applying proposed hybridized five membership fuzzy logic algorithm**

Abiyev et al. [1] has revealed that a fuzzy based evolutionary algorithm is fastest and easiest one regarding implementation in robot navigation problems than other classical algorithms e.g. potential field method, vector field histogram and local navigation. The superiority of the fuzzy algorithm is probably because of its resemblance to the intelligent human reasoning and decision making process. The simulation result by present hybridized five membership fuzzy logic algorithm (Figure 4.10 (b)) is showing healthier performance in comparison with the fuzzy based evolutionary algorithm designed by Abiyev et al. [1] (Figure 4.10 (a)) for identical obstacle arrangement along with similar starting and goal position regarding consumption of time (shown in Table 4.5) by robot for pursuing target along with its avoidance nature against obstacles of different shapes and sizes.

## 4.5 Experimental Results and Comparisons:

The assumptions employed for mechanical structure and kinematic and dynamic analysis of drive configuration used in mobile robot for experimental purpose are given in Chapter 3. The experimental results have been conducted by loading the software into the developed mobile robot in the robotics laboratory (Details will be given in Chapter 7). The simulation results are also qualified with experimental results (Figure 4.11 and Figure 4.12) and comparisons are also done with the results of previously proposed new minimum avoidance system by Motlagh et al. [71] (Figure 4.9 (a)) and fuzzy based evolutionary algorithm designed by Abiyev et al. [1] (Figure 4.10 (a)) respectively.

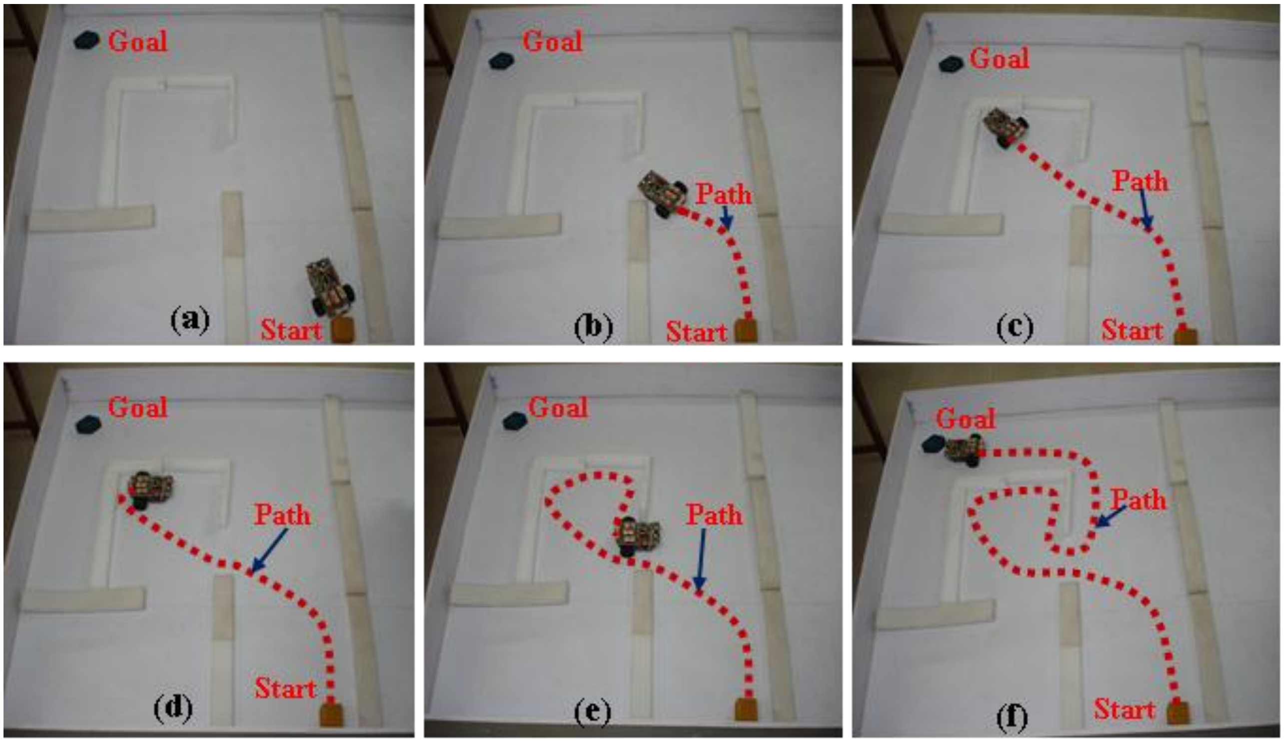



**Figure 4.11: Experimental results of mobile robot to reach the target successfully in same environment used in simulation mode (Figure 4.9)**

During experiment, paths traced by the robot are marked on the floor by a pen (fixed to the front of the robots) as they move in Figure 4.11 and 4.12. The experimentally acquired paths closely follow the paths sketched by the robots during simulation for analogous arrangement of obstacles, start and goal point. It has been acquired that the experimental path lengths and time taken are more than the simulation path lengths and time taken. This is due to presence of various errors (e.g. signal transmission error in data-cable, obstacle or target tracking error, presence of friction in rotating elements, slippage between floor and wheels, friction between supported point and floor etc.).

Table 4.5 shows an independent comparison path length used by the robot in simulation and in the experimental mode for obstacle avoidance and target seeking along with the performances of results by Motlagh et al. [71] and Abiyev et al. [1]. The path lengths are taken in average from 12 different experiments which have been performed for each of the two environmental scenarios shown in (Figure 4.11, 4.12).

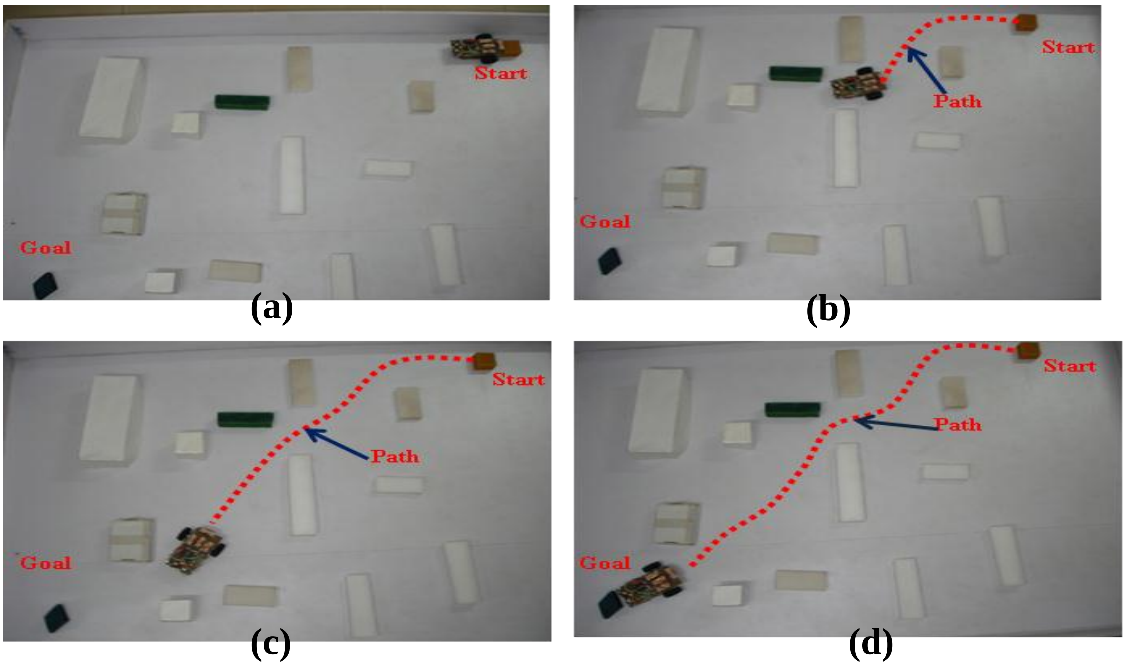



**Figure 4.12: Experimental results of mobile robot to reach the target successfully in same environment used in simulation mode (Figure 4.10)**

**Table 4.5: Path Length traced by Robot in simulation and experiment to reach target avoiding obstacles**

| Sl. No | Environmental Scenario | Path Length in simulation mode by proposed algorithm (in pixel) | Path Length in Experimental mode by proposed algorithm (in pixel) | Path Length in experiment of Previous research work (in pixel) | % of error between simulation and experimental results by proposed technique |
|---|---|---|---|---|---|
| **1** | 1st scenario (Figure 4.9 and 4.11) | 237 | 279 | 243 | 15.05% |
| **2** | 2nd scenario (Figure 4.10 and 4.12) | 181 | 213 | 190 | 15.02% |

## 4.6 Conclusion:

An assessment can be construed from the hypothetical and logical analysis strengthened by simulation and experimental results. Based on sensory information innovative fuzzy reactive controller can give a reasonable performance for static obstacle avoidance, escaping from local minima problems, and seeking target during navigation of mobile robot in a complex hazardous environment. The inference mechanism accompanied by charted fuzzy rule base gives a navigational control scheme, which indirectly addresses the demand of determining the sequence of actions such as to recognise the environment, to avoid obstacles and to achieve the goal successfully. Performance measure has been carried out through the comparison between simulation and experimental results. The comparisons with previous researches like Motlagh et al. [71] and Abiyev et al. [1] for different environmental scenarios in terms of consumed time are presenting a degree of partial optimization capability of the controller which clinches that fuzzy controller embedded with hybridized five membership functions can find the specified target using almost minimum path length and time. So the comparisons of performances certify the proficiency and malleability of hybridized five membership fuzzy logic approach. This navigation strategy can be used in a mobile robot working in hazardous conditions or unmanned space missions.

# 5 Navigational Path Analysis using Takagi-Sugeno Model based Fuzzy Controller

Appliances of autonomous wheeled mobile robotics in many unknown missions like planetary exploration and space applications prefer shorter path in optimized time for consuming less energy. Therefore an intelligent real time path planning algorithm is always required. Among classical FLC, Takagi-Sugeno model is a competent engineering tool for modelling and controlling complex systems. It can be described by simple fuzzy IF-THEN rules which can give local linear representation of a nonlinear system by decomposing the whole input space into several partial fuzzy spaces and representing each output space with a linear equation. Such a model is capable of gaining intelligence, autonomous behaviors and can be equipped with knowledge, motivation, reasoning, and planning capabilities. Obtained results depicts that the method is efficient and effective for navigation of mobile robot in dynamic cluttered environment.

## 5.1 Introduction

An important problem in autonomous navigation is the need to cope with the large amount of uncertainty, inherent to the natural environment. By utilizing human reasoning process for making decisions through linguistic rules, Fuzzy inference system (FIS) has become an adequate tool to address this problem. Generation of the knowledge base is crucial to produce high quality FIS for system modelling [18]. System modelling encounters two fundamental and rather conflicting requirements such as accuracy and transparency as far as models are concerned. In addition to this delicate balancing act, dealing with multi-variable systems, difficulties are increasing quite apparently leading towards system delay or even system breakdown [38]. Many approaches have been proposed to address the issue of automatic generation of membership functions and rules to receive more satisfactory system performance and better robustness against all the possible unpredicted inputs from the real-world environment [57].

In between two basic forms of FIS, Mamdani and Takagi Sugeno, TS rule systems are more flexible due to stronger modeling capability to solve some complex problems. The main difference between these two types of fuzzy rules lies in the fact that the consequent part of the TS rules is normally a concrete linear function of input variables instead of some fuzzy linguistic variables [51]. Therefore it is possible to use T-S model as optimization technique to find best parameters to map nonlinear dynamic systems [41]. TS algorithms are quite complicated and are mainly suitable for fuzzy rules with piecewise linear fuzzy membership functions and linear consequents. Moreover, their algorithms have difficulties in real time implementation, which has limited its application [51].

This research concentrates on appropriate assessment of robot wheel velocities from sensory information by T-S fuzzy algorithm for attaining assimilation of diverse robotic behaviors and generating real-world paths towards the target under various situations with a comparatively less computational complexity than Mamdani-Fuzzy based approach. As Mamdani-fuzzy system is suffered from lack of learning property, the system execution can be augmented by commencing T-S fuzzy rules whose coefficients of the consequent variables can be obtained by statistical estimation method. In other way, it can be argued that initially obtained membership functions and rules based on a prior knowledge are often in need of refinement towards higher accuracy. By representing the rule’s consequent as linear combination of input variables, flexibility and robustness of fuzzy systems can be enlarged to deal with unexpected unknown inputs from real-world environments.

This chapter is schematized into seven sections succeeding the introduction; generalized Takagi-Sugeno fuzzy inference system applied for robot navigation is explained in section 5.2. Here, input and output variables and fuzzy rule base are considered to be identical with chapter 4. A MATLAB Interpretation for finding out wheel velocities of robot based on T-S fuzzy approach has been pronounced along with a small example in section 5.3. Advantages and

Disadvantages of present model in navigation purpose are simplified in section 5.4. Comparisons between Mamadani and Takagi-Sugeno Approach are quantified in section 5.5. The simulation results and their comparisons with previous work are conferred in section 5.6 and in section 5.7 experimental results are substantiated with simulation to make apparent the incomparability of the anticipated approach. Finally, Conclusions are inferred in section 5.8.

## 5.2 Fuzzy Inference Process for the T-S type Fuzzy Model:

The TS fuzzy model can be defined as a fuzzy interpolation of linear modelling strategies for describing the dynamic behaviour of systems through polynomials, in which the variables are the values of input on that instant.

Using prior knowledge gathered by human experts, a knowledge base consists of fuzzified unknown inputs, membership functions and fuzzy rules can be constructed same as Mamdani-Fuzzy inference mechanism. In general, a supervised or unsupervised clustering method determines the partition of the given knowledge, and membership functions for each feature can be obtained according the resulting partition information. The output of fuzzy reasoning is given by the aggregation of the values inferred by some implications that were applied to an input. This is a simple description of the systematic mechanism of the T-S type fuzzy inference model as shown in Figure 5.1[18]

This fuzzy modelling approach modifies the rule base so that all of these relationships are represented by local linear input and output [49]. The rules can be defined, therefore, by

*If* $x_1$ *is* $X_1$ *and* $x_2$ *is* $X_2$ *and* ……$x_i$ *is* $X_i$ *then* $y = f_j(x_1, x_2, \ldots, x_i)$ (5.1)

where,

$$f_j(x_{1\ldots i}) = a_0 + a_1 x_1 + a_2 x_2 + \ldots\ldots + a_i x_i \quad (5.2)$$

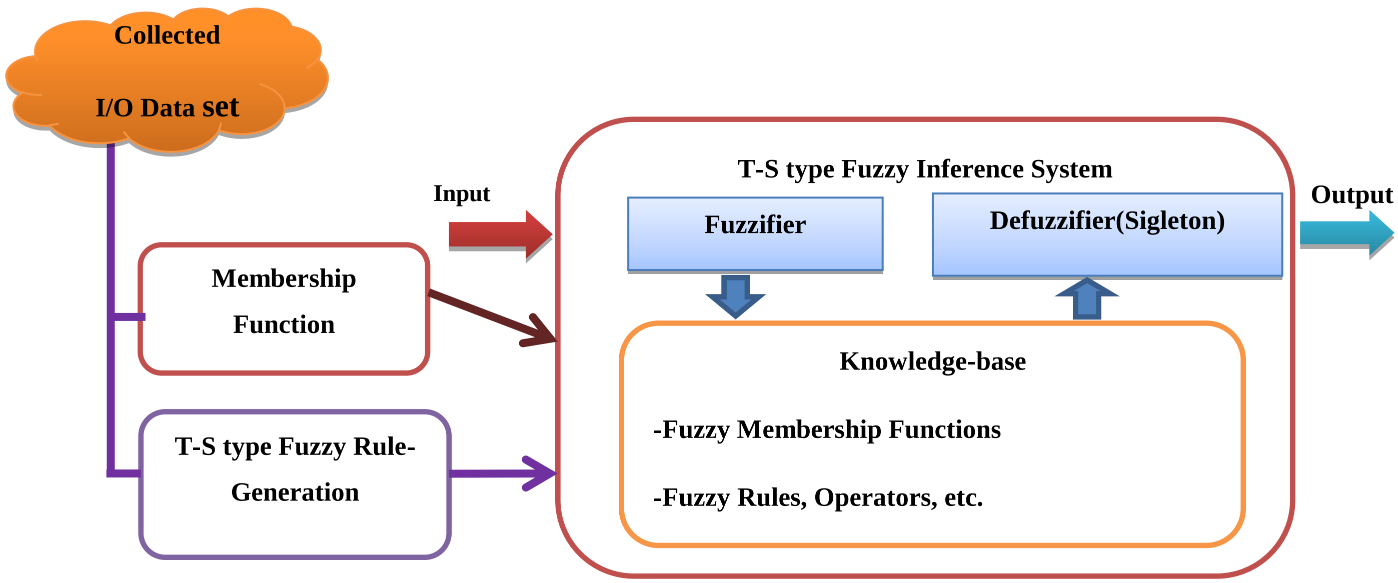



**Figure 5.1:** Fuzzy Inference Process for the T-S type fuzzy model

$x_i$ is the $i$th input variable, $f_j$ ( ) is output for jth rule and $a_{1...i}$ are the parameters of linear equation.

In the case of linear equation, the system is of "first order" and the constant type has "zero order", which can be viewed either as a special case of the Mamdani fuzzy inference system, in which each rule's consequent is specified by a fuzzy singleton (or a pre-defuzzified consequent) [48]. As a consequence, it is reasonable to affirm that, for each rule, the activation degree generated by the current state $x_{1...i}$, will be attributed to the calculated value of $f_j(x_{1...i})$ by eliminating the defuzzification step.

The T-S type fuzzy model suggested by Takagi and Sugeno [97] can be generalized in the field of mobile robot navigation by considering sensory information like Left, Front and Right obstacle distances and Heading Angle as input and Left and Right Wheel Velocities as output. Number, type, linguistic term and parametric distribution of Membership Functions for each input variables are taken same as previous chapter (Mamdani-based FLC). Single spikes, or Singletons are considered as the membership functions of the consequent of rules (Left and Right Wheel Velocities) rather than a distributed Fuzzy set. A singleton is a fuzzy set with a membership function that is unity at a single particular point on the universe of discourse and zero everywhere else. It can be imagined as a predefuzzified Fuzzy set.

So, T-S fuzzy control rules are defined in a general form for four inputs and two outputs fuzzy system as follows:

If ($LD_i$ is $\mu(LD_j)$ and $FD_i$ is $\mu(FD_k)$ and $RD_i$ is $\mu(RD_m)$ and $HA_i$ is $\mu(HA_n)$,

Then $LV_i = b_{i0} + b_{i1}\mu(LD_j) + b_{i2}\mu(FD_k) + b_{i3}\mu(RD_m) + b_{i4}\mu(HM_n)$ and

$$RV_i = c_{i0} + c_{i1}\mu(LD_j) + c_{i2}\mu(FD_k) + c_{i3}\mu(RD_m) + c_{i4}\mu(HM_n) \qquad (5.3)$$

Where,

$i$: the number of rules.

$j, k, m, n$: 1 to 5 as $LD$, $FD$, $RD$ and $HA$ have five membership functions each.

$\mu$: Matching degree of a specific input variable in a rule

$b_{i0to4}$ : Coefficients chosen for linear equation of Left Wheel Velocity

$c_{i0to4}$ : Coefficients chosen for linear equation of Right Wheel Velocity

In order to estimate the coefficients of the rule consequent, the least squares estimation has been widely used. Finally, the output firing strength for left and right wheel velocities can be computed as,

$$w_i = \min\{\mu(LD_j), \mu(FD_k), \mu(RD_m) \text{ and } \mu(HA_n)\} \qquad (5.4)$$

The final output (crisp value) of the T-S based fuzzy logic controller of left and right wheel velocities for N number of rules can be calculated as follows:

$$\left.\begin{aligned} Left\ \text{Velocity} = \text{L}V = \frac{\sum_{i=1}^{N} w_i \times LV_i}{\sum_{i=1}^{N} w_i} \\ Right\ \text{Velocity}\ = RV = \frac{\sum_{i=1}^{N} w_i \times RV_i}{\sum_{i=1}^{N} w_i} \end{aligned}\right\} \tag{5.5}$$

So, the total output, is obtained by (5.5) can be defined as a weighted summation of linear combination of inputs to represent the non-linear characteristic functions.

## 5.3 T-S Model Applied for Navigational Purpose: A MATLAB Interpretation:

In this segment of the chapter, Left and Right Wheel Velocities have been derived using the Takagi-Sugeno FIS of MATLAB toolbox. Parameters and linguistic terms of Hybridized five Membership Functions used in Input variables of Takagi-Sugeno FIS (Figure 5.2(a)) are exactly same as Mamdani FIS of Chapter 4. As Membership functions of output variables are like singleton, they are shown in Figure 5.2(b).

For a particular scenario (in Figure 4.7) same as Chapter 4, where robot has left, front and right obstacle distances as 1.37m, 3.12m and 5.34m respectively and a target is located at a positive angle of $72^0$ from the current position of robot, activated 16 fuzzy rules are applied in Sugeno-FIS (MATLAB) to find out left wheel velocity and right wheel velocity of the robot. In MATLAB, Rule View and Surface View for the present problem has been shown in Figure 5.3. It has been retrieved that assessed Left and Right Wheel Velocities from Mamdani-FIS and Sugeno-FIS for the same situation are differed by values. But velocities of Sugeno-FIS are more augmented to maintain target orientation along with obstacle avoidance than Mamdani-FIS. This assertion has been verified in simulation and experimental results followed by this section.

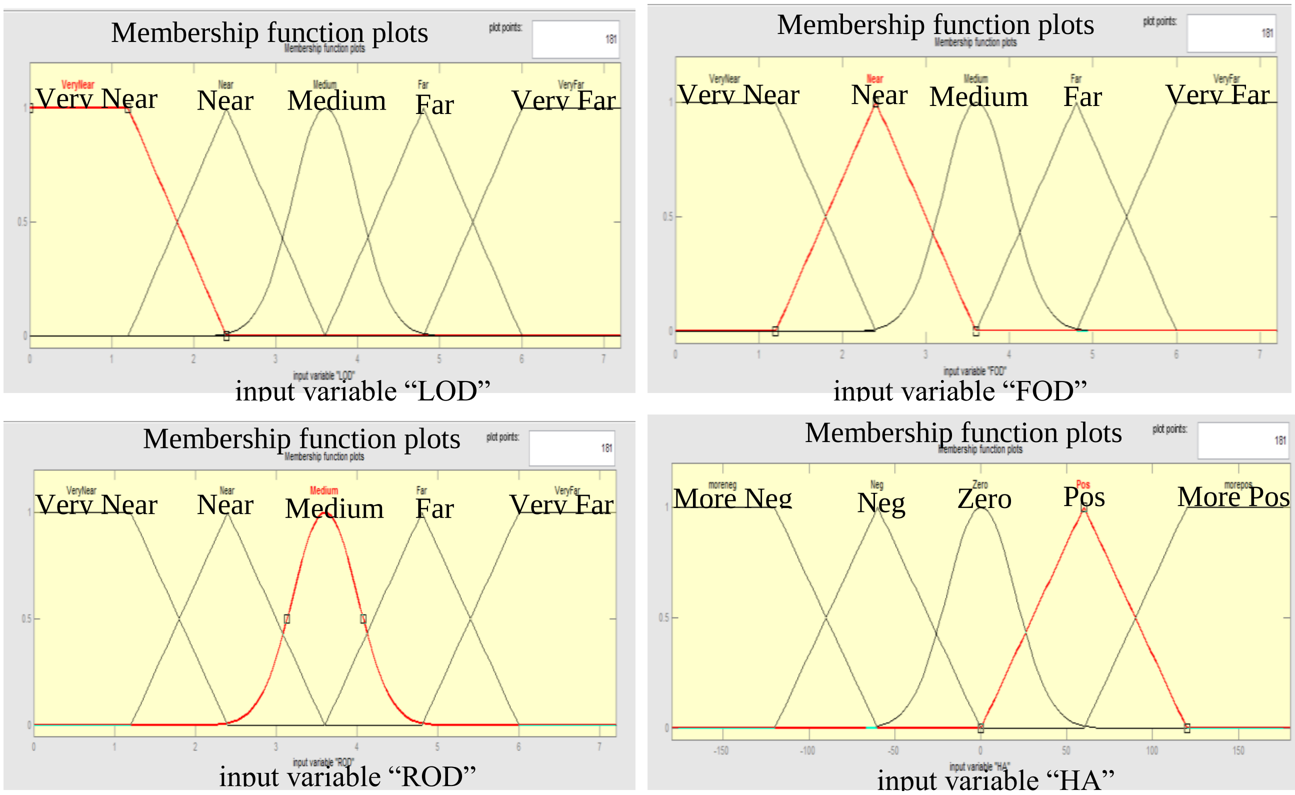



**Figure 5.1(a) Input Variables of Takagi-Sugeno FIS in MATLAB**

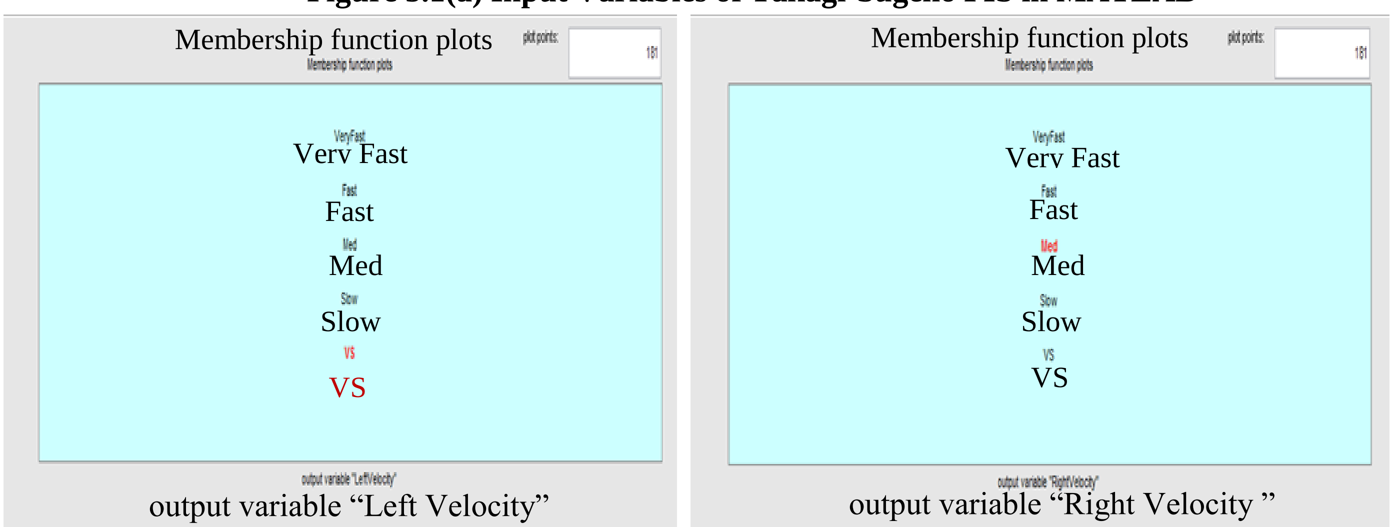



**Figure 5.2 (b) Output Variables of Takagi-Sugeno FIS in MATLAB**

**Figure 5.3: Resultant Left Wheel and Right Wheel Velocities in Rule View of Sugeno-FIS**

## 5.4 Comparisons between Mamadani and Takagi-Sugeno Approach: Simulation Result

The simulation result exposes that the anticipated T-S based FLC can perform robot navigation within complex and scattered arrangement of obstacles in more optimized manner than Mamdani Fuzzy due to its adaptive nature. From simulation result (Figure 5.4) and comparison of two controllers in terms of path length and navigational time depicted in Table 5.1, it can be resolved that T-S model has enhanced performance over Mamdani-Fuzzy not only in theoretical analysis but also in virtual environment.

To be a more compact and computationally efficient representation than a Mamdani system, the Sugeno system lends itself to the use of adaptive techniques for constructing fuzzy models. These adaptive techniques can be used to customize the membership functions so that the fuzzy system models the data accurately.

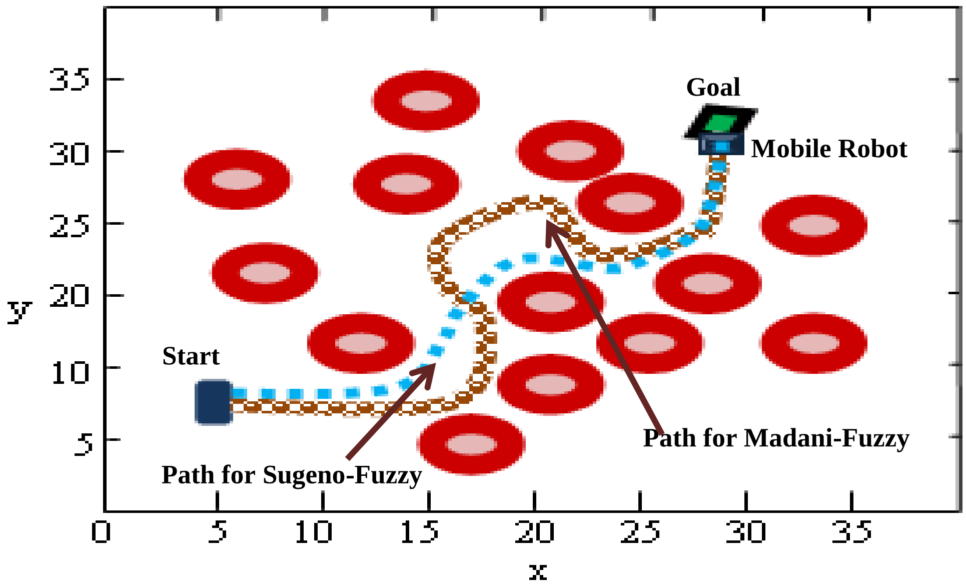



**Figure 5.4: Simulation Result for Mamdani and Sugeno based Fuzzy controller**

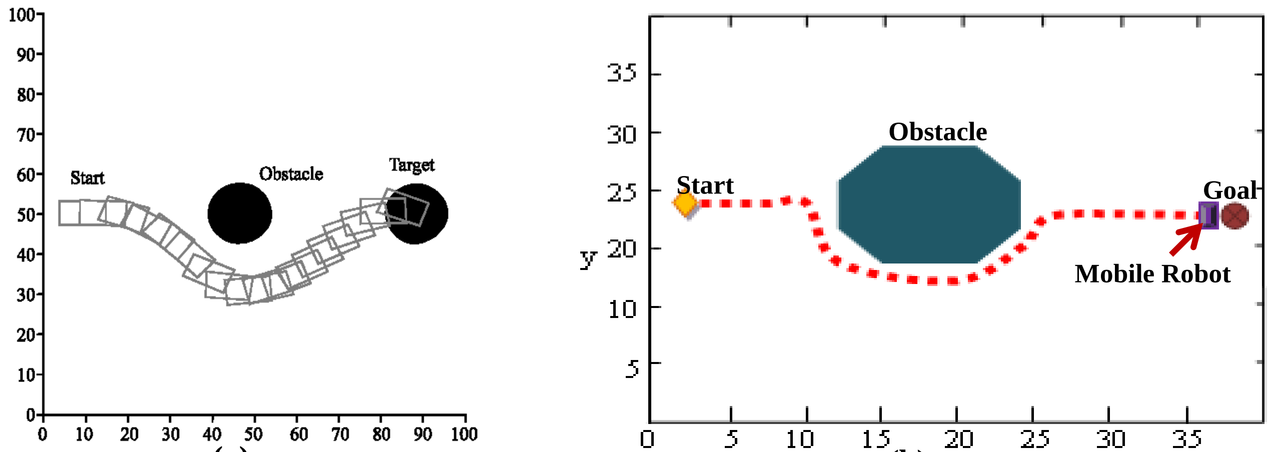
**Table 5.1: Comparison between Mamdani and Takagi-Sugeno Fuzzy Controller in terms of path length and navigational time**

| Sl.No. | Navigational Analysis for present scenario (Figure 5.4) | Path length (in pixel) | Time (in second) |
|---|---|---|---|
| **1.** | **Navigation with Mamdani based Fuzzy controller** | 175 | 18.23 |
| **2.** | **Navigation with Sugeno based Fuzzy controller** | 167 | 15.93 |

## 5.5 Simulation Result and Discussion:

To endorse the ability and robustness of the hybridized five membership function based Takagi Sugeno fuzzy control algorithm, simulation result on mobile robot navigation has been compared with previous research work on Takagi-Sugeno based FLC whose parameters are self-tuned by learning method based on gradient descent (Figure 5.5(a)) by Kermiche et al. [56]. The simulated environment is created synthetically containing obstacles as well as target at fixed positions (Figure 5.5(b)). Target searching algorithms assume that the goal state is fixed and does not change during navigation of mobile robot.

The comparison clarifies that satisfactory performance during navigation can be achieved by proposed hybridized five membership function based T-S fuzzy model, using comparatively reduced path length (shown in Table 5.2) than previously proposed self-tuned parameter based T-S Fuzzy algorithm (Figure 5.5(a)) by Kermiche et al. [56].




**(a)** **(b)**

**Figure 5.5: (a) Mobile robot reference trajectories by Kermieche et al. [56] after training (b) Path traced by Proposed T-S based Controller**

## 5.7 Experimental Result and Comparison:

During simulation and experimental result, it has been realized that mobile robot can efficiently reach the target avoiding collision with obstacles in its way. Model mobile robot traces the path in real world environment (Figure 5.6) by following similar way as performed in previous chapter. The experimentally learnt path approximately follows the path drawn by the robot during simulation for same starting and ending positions and obstacle arrangement which validate the proposed method.

Table 5.2 shows an impartial measure of path length used by the robot in simulation and in the experimental mode along with the performances of result by Kermiche et al. [56].

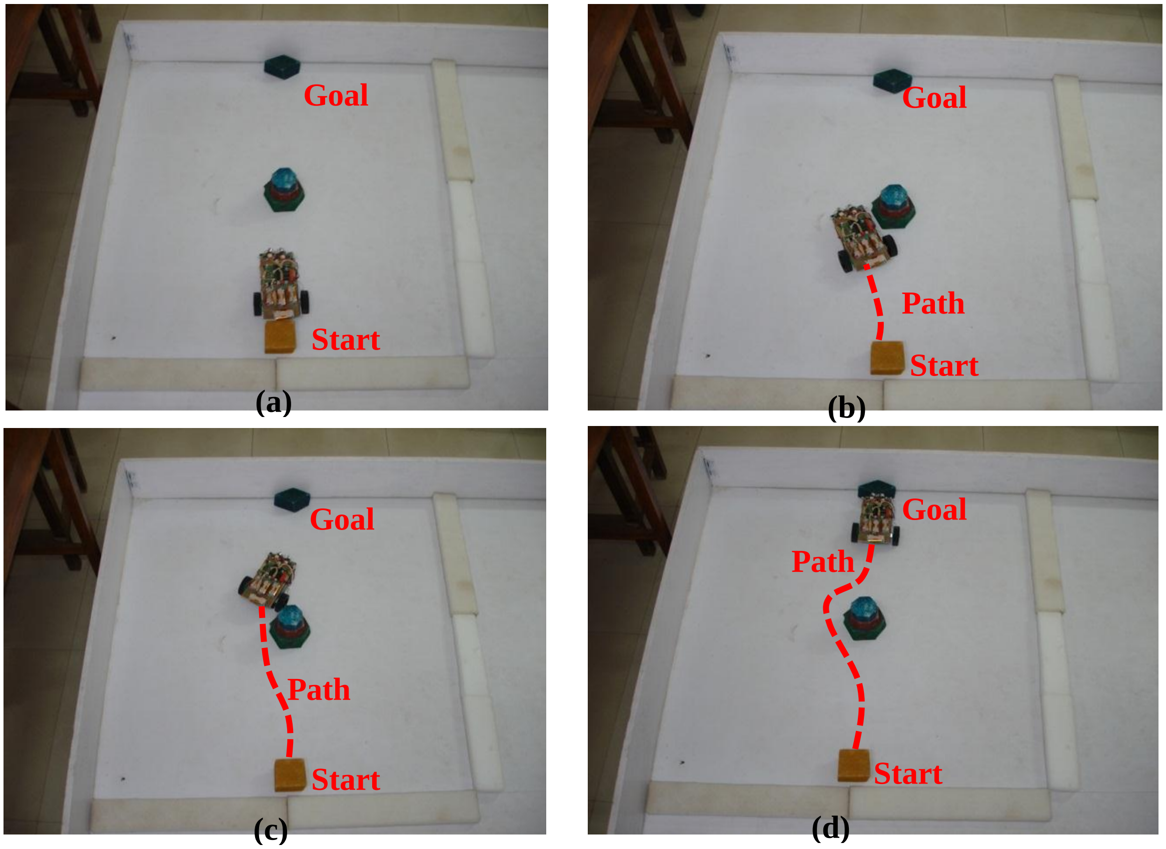



**Figure 5.6: Experimental Result by using Sugeno FIS in same environment used in simulation mode (Figure 5.5(b))**

The path lengths are measured in average from 7 different experiments which have been performed in environmental scenario (Figure 5.6). The robotics behaviours such as obstacle avoidance and target seeking have been verified in simulation and experimental modes for T-S based fuzzy T-S based FLC provides much faster navigation in an unknown environment and has less computational effort than other conventional approaches.

**Table 5.2: Path length traced by Robot in Simulation and Experiment for Sugeno Based Fuzzy Model**

| **Path Length in simulation by proposed algorithm (in pixel)** | **Path Length in Experimental mode by proposed algorithm (in pixel)** | **Path Length in simulation of Previous Research work by Kermiche et al. [56] (in pixel)** | **% of error between simulation and experimental results by proposed technique** |
|---|---|---|---|
| 186.73 | 215.54 | 191.35 | 13.664% |

## 5.8 Conclusion:

This chapter contributes to the efforts of developing practical, modular, and easy-to-implement T-S based Fuzzy navigation algorithm that is both cost and computationally effective.

1. It has been acquired that the Takagi-Sugeno FLC provides better result with less time consumption and computational complexity than previously developed Mamdani-FLC.
2. The robot rapidly maps their surroundings which provide sufficient information for path optimization during navigation.
3. It is successfully applied for navigation in static environments (Figure 5.4, 5.5(b), 5.6). The robot rapidly recognises their surroundings which provide sufficient information for path optimization during navigation.
4. Comparison of result between the current developed T-S FLC and with another T-S model by Kermiche et al. [56] is rendering an higher degree of partial optimization capability.
5. The proposed method is simple but efficient tool for mobile robot navigation, especially in a real world dynamic environment saving considerable time and effort.

Next chapter will describe an integration of fuzzy logic and neural network technique for more efficient navigation of the mobile robots by combining the benefits of each field (i.e. perception, cognition, and motion control).

# 6 Analysis of Fuzzy-Neural Network for Navigation

A new paradigm of intelligent navigation system must be enriched with some common features like: criteria for optimal performance and ways to optimize design, structure and control of robot. With the growing need for the deployment of intelligent, highly autonomous systems, it would be beneficial to flawlessly combine robust learning capabilities of artificial neural networks with a high level of knowledge interpretability provided by fuzzy-logic. Fuzzy-neural network is able to build comprehensive knowledge bases considering sensor-rich system with real time constraints by adaptive learning, rule extraction and insertion, and neural/fuzzy reasoning. The training for back propagation algorithm and its navigational performances analysis has been done in real experimental setup. As experimental result matches well with the simulation result, the realism of method is verified.

## 6.1 Introduction

Navigation of mobile robot, which can be defined as the strings of schedules required during goal achieving without any collision with static as well as dynamic obstacle, necessitates the abilities of a mobile robot to plan and execute optimized paths within its environment; it may be vague, huge and either partially or absolutely dynamic. Development of new concepts and strategies to tolerate a wide range of uncertainty [51] in the area of mobile robot navigation has attracted attentions of many researchers.

Fuzzy Logic and Neural networks both have properties for controlling inherent uncertainties and inaccuracies in the sensor data and planning of a strategic action selection mechanism.

- Fuzzy systems are able to treat uncertain and imprecise information, they make, use of knowledge in the form of easily understandable linguistic rules. In a fuzzy inference system, fuzzy logical rules can model the qualitative aspects of human knowledge and reasoning processes without employing precise quantitative analysis [53]. Their drawbacks are caused mainly by the difficulty of designing accurate membership functions and lack of a systematic procedure for the transformation of expert knowledge into the rule base.

- On the other hand, neural networks have strong learning abilities though they are weak for expressing rule-based knowledge [63]. Artificial neural network (ANN) systems offer advantages of acquiring knowledge through learning, [53] adaptation, fault-tolerance, parallelism, and generalization. Neural networks have the ability to learn but with some neural networks, knowledge representation and extraction are difficult [39].

So the idea is to combine neural networks and fuzzy systems to overcome their disadvantages, but to retain their advantages combining the versatility of neural networks and fuzzy logic replicating aspects of human thought [53]. As the former property reduces the time required to create the model, the latter increases the usefulness of the model [6], Fuzzy neural network (FNN) provides computational intelligence that come with significant learning abilities and inherent transparency (interpretability) to provide strong mechanisms for building intelligent systems that must operate in rapidly changing environments. So, it is able to learn and approximate real-world concepts, building a knowledge base that may be interpreted and modified by the user [37].

This chapter delivers a narrative attitude for design of a perceptive controller for autonomous mobile robot using multilayer feed forward neural network next to FLC, as FLC is not completely perfect to deal with the increment of variables in robotic environment. Fuzzy logic has already been used for behavior design such as obstacle avoidance, wall following and target seeking. To solve the problem of insufficient knowledge, rule-based controller is trained by a back-propagation learning algorithm that allows autonomous robot to gain more accurate steering angle than sensory information in a motive to minimize the error and to maintain a time-optimal and collision-free path in unknown and cluttered environments. Simulation and experimental results are presented to demonstrate the validity of the approach.

The framework of this chapter is as follows, succeeding the introduction, the fuzzy-neural architecture for navigation of mobile robot is depicted in section 6.2. The simulation results are discussed and to ascertain viability of the developed technique comparisons have been made with other methods [63, 106] in section 6.3. In section, 6.4 experimental results are certified with simulation to reveal the supremacy of the recommended methodology. Finally, summary has been briefed in the last section 6.5.

## 6.2 Analysis of Fuzzy-Neuro Architecture for Navigation:

To reduce travel time as well as the distance travelled, Four layer perceptron neural network has been designed by using outcomes from the FLC as well as environmental information to make navigational decisions. The first layer is used as input layer which has six neurons; four for receiving the values of the distances from obstacles in front, left and right of the robot and also for the target bearing angle and other Twos are for Left and Right wheel Velocities from FLC. Next the robot network consists of two hidden layers (shown in Figure 6.1) which adjusted the weight of neuron; as with one hidden layer it is difficult to train the network within a specified error limit. The training error is the difference between desired output and actual output. The first hidden layer has eighteen neurons and the second hidden layer has five neurons. These numbers of hidden layers were also found empirically. Then an output layer with a single neuron which provide steering angle to control the direction of movement of the robot. Back propagation method is used to minimize the error and optimize the path and time of mobile robot to reach the target [43].

The numbers of neurons are found based on the number of training patterns and the convergence of error during training to a minimum threshold error to control the direction of movement of the robot. The neural network is empirically trained to navigate by 200 training patterns representing typical scenarios, some of which are depicted in Table 6.1. For example, in training pattern no. (vii) a robot is surrounded by left, front and right obstacles at distances of 21c.m, 15c.m and 19c.m respectively. Ultrasonic sensor is giving the reading of $43^0$ for the current position of robot; i.e. target is located at an angle of $43^0$ at the right side of robot. Output of FLC for this situation as left and right wheel velocities are 5.8c.m/s and 5.1c.m/s respectively based on sensor data. In this scenario, neural network is trained to steer robot towards its right with an angle of $26^0$ with respect to goal position to maintain the shorter trajectory.

During training and during normal operation, the input patterns fed to the neural network comprise the following components:

$Y_1^{\{1\}}$ = Left obstacle distance from the robot (6.1a)

$Y_2^{\{1\}}$ = Right obstacle distance from the robot (6.1b)

$Y_3^{\{1\}}$ = Front obstacle distance from the robot (6.1c)

$Y_4^{\{1\}}$ = Target bearing (6.1d)

$Y_5^{\{1\}}$ = Left Wheel Velocity (6.1e)

$Y_6^{\{1\}}$ = Right Wheel Velocity (6.1f)

These input values are distributed to the hidden neurons which generate outputs given by

$$Y_j^{\{\mathrm{lay}\}} = \mathrm{f}(\mathrm{V}_j^{\{\mathrm{lay}\}}) \tag{6.2}$$

Where, $$\mathrm{V}_j^{\{\mathrm{lay}\}} = \sum_i W_{ji}^{\{lay\}} \mathrm{Y}_i^{\{lay-1\}} \tag{6.3}$$

lay: layer number (2 or 3)

*j*: label for jth neuron in hidden layer ‘lay’,

*i*: label for ith neuron in hidden layer ‘lay-1’

$W_{ji}^{\{lay\}}$: Weight of the connection from neuron i in layer 'lay-1' to neuron j in layer 'lay'.

f (.) : Activation function, chosen in this work as the continuous log-sigmoid function:

$$f(x) = \frac{1}{1+e^{-\beta x}} \tag{6.4}$$

Where, $\beta$ is a slope parameter.

The sigmoid has the property of being similar to the step function, but with the addition of a region of uncertainty. Sigmoid functions in this respect are very similar to the input-output relationships of biological neurons, although not exactly the same. Figure 6.2 is the graph of a sigmoid function.

During training, the network output $\theta_{actual}$ may differ from the desired output $\theta_{desired}$ as specified in the training pattern presented to the network. A measure of the performance of the network is the instantaneous sum-squared difference between $\theta_{desired}$ and $\theta_{actual}$ for the set of presented training patterns:

$$\mathrm{Err} = \sum_{all\ \text{traning pattrns}} (\theta_{desired} - \theta_{actual})^2 \tag{6.5}$$

The error back propagation method is employed to train the network. This method requires the computation of local error gradients in order to determine appropriate weight corrections to reduce Err. For the output layer, the error gradient is:

$$\delta^{\{4\}} = f'(V_1^{\{4\}})(\theta_{desired} - \theta_{actual})^2 \tag{6.6}$$

The local gradient for neurons in hidden layer {lay} is given by:

$$\delta_j^{\{lay\}} = f'(V_1^{\{lay\}})(\sum_k \delta_k^{\{lay+1\}} W_{kj}^{\{lay+1\}}) \tag{6.7}$$

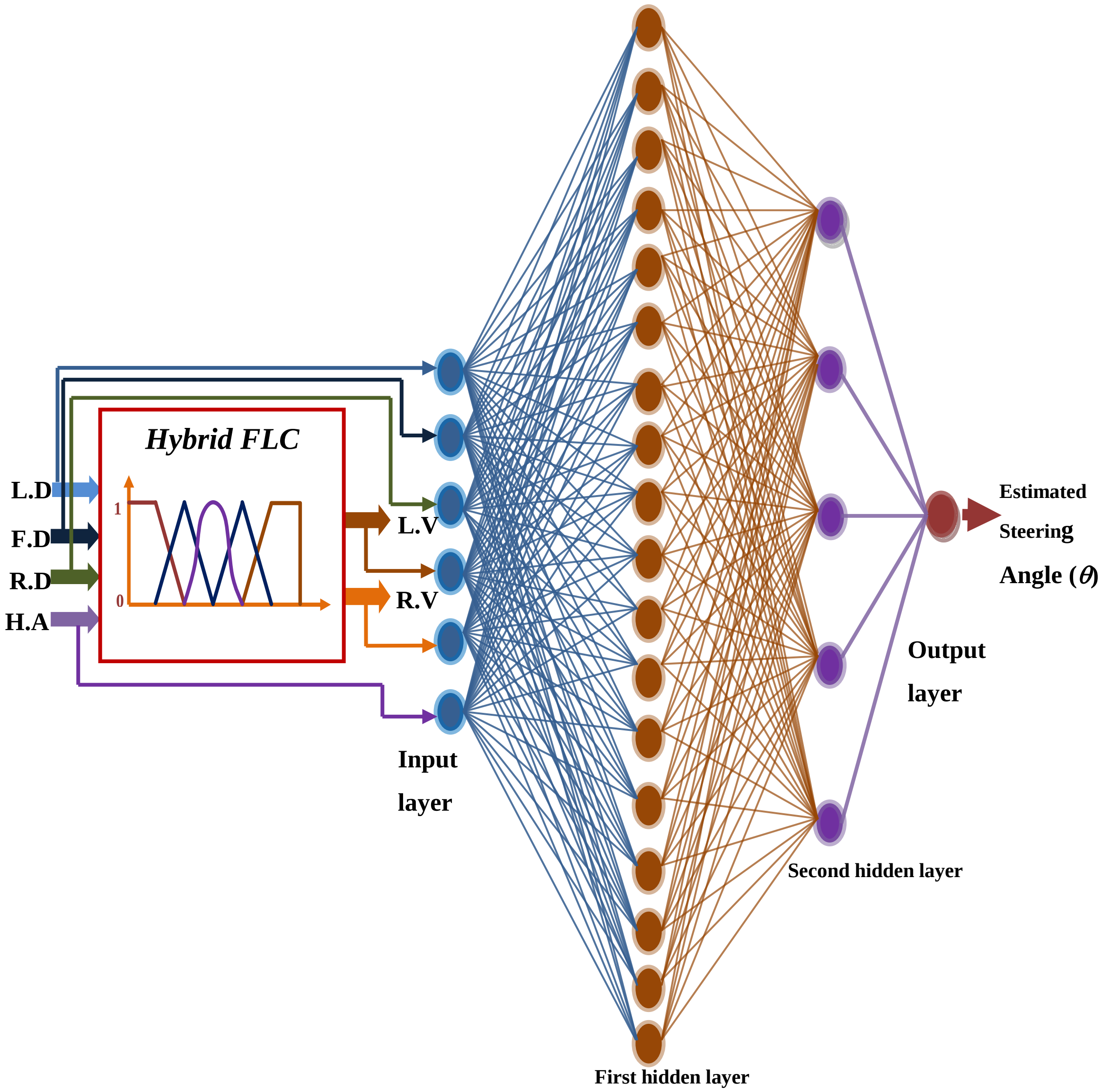



**Figure 6.1: Hybrid Fuzzy & Multilayer Neural Controller for Coordination of Robotic Behaviors**

The synaptic weights are updated according to the following expressions:

$$W_{ji}\ (t+1) = W_{ji}\ (t) + \Delta W_{ji}\ (t+1) \tag{6.8}$$

And $$W_{ji}\ (t+1) = \alpha\Delta\ W_{ji}\ (t) + \eta\delta_j^{\{lay\}} y_i^{\{lay-1\}} \tag{6.9}$$

Where, $\alpha$ = momentum coefficient (chosen empirically as 0.2 in this work)

$\eta$ = learning rate (chosen empirically as 0.35 in this work)

$t$ = iteration number, each iteration consisting of the presentation of a training pattern and correction of the weights.

The final output from the neural network is:

$$\theta_{actual} = f(V_1^{\{4\}}) \tag{6.10}$$

Where, $$V_1^{\{4\}} = \sum_i W_{1i}^{\{4\}} y_i^{\{3\}} \tag{6.11}$$

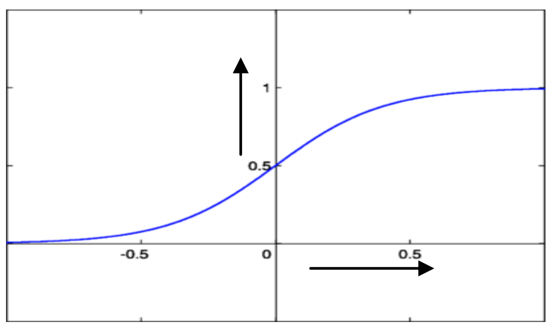



**Figure 6.2: Continuous Log-Sigmoid function used for activation function.**

It should be noted learning can take place continuously even during normal target seeking behavior. This enables the controller to adopt the changes in the robot's path while moving towards target. Mainly three behaviors (obstacle avoidance, wall following and target seeking) are required to train and to design an intelligent controller for mobile robot being used to navigate in a cluttered environment.

**Table 6.1 Some of the training pattern of Fuzzy-Neural controller**

| Sl. No. of Training Patterns | Inputs to the neural network | | | | | | Output |
|---|---|---|---|---|---|---|---|
| | Left obstacle distance (cm) | Front obstacle distance (cm) | Right obstacle distance (cm) | Target angle (degree) | Left Wheel Velocity from FLC (cm/s) (in approximate) | Right Wheel Velocity from FLC (cm/s) (in approximate) | Steering angle (degree) |
| (i) | 22 | 20 | 15 | 0 | 4.5 | 6.8 | -23 |
| (ii) | 11 | 17 | 15 | 0 | 5.2 | 4.3 | 19 |
| (iii) | 15 | 21 | 9 | 0 | 3.8 | 4.9 | -14 |
| (iv) | 8 | 13 | 15 | 27 | 6.7 | 3.9 | 23 |
| (v) | 20 | 10 | 12 | -39 | 3.2 | 7.1 | -27 |
| (vi) | 13 | 20 | 10 | 0 | 4.3 | 4.7 | -9 |
| (vii) | 21 | 15 | 19 | 43 | 5.8 | 5.1 | 26 |
| (viii) | 16 | 30 | 25 | 36 | 6.5 | 4.2 | 29 |
| (ix) | 14 | 33 | 19 | 24 | 5.23 | 4.65 | 17 |
| (x) | 27 | 35 | 16 | -54 | 4.29 | 6.7 | -41 |
| (xi) | 17 | 29 | 14 | 0 | 4.7 | 4.9 | -5 |
| (xii) | 23 | 14 | 27 | 0 | 5.9 | 4.1 | 13 |
| (xiii) | 30 | 10 | 15 | -45 | 3.4 | 6.9 | -37 |
| (xiv) | 12 | 17 | 14 | 8 | 5.1 | 3.78 | 6 |
| (xv) | 10 | 10 | 15 | 17 | 5.74 | 3.69 | 11 |
| (xvi) | 19 | 23 | 15 | 0 | 4.67 | 4.93 | -13 |
| (xvii) | 38 | 10 | 33 | -63 | 4.15 | 7.1 | -51 |
| (xviii) | 21 | 7 | 16 | -49 | 3.97 | 6.75 | -37 |
| (xix) | 35 | 18 | 20 | -37 | 4.61 | 6.98 | -29 |
| (xx) | 31 | 17 | 15 | -47 | 3.54 | 6.87 | -41 |
| (xxi) | 24 | 32 | 13 | -56 | 4.35 | 6.79 | -43 |
| (xxii) | 18 | 25 | 9 | 0 | 4.25 | 4.67 | -13 |

## 6.3 Simulation Results and Discussion:

The series of simulations test have been conducted to exhibit that the anticipated method can partially fulfill the most of the fundamental as well as critical robotic behaviors during navigation in complex and uncertain environments.

In Figure 6.3, Obstacles of different shape and size are placed in an unstructured manner. Here, mobile robot is approaching towards target; decelerating obstacle avoidance as its main reactive behavior as well as edge following behavior is also appeared here.

Wall following behavior of mobile robot and tendency to escape dead end condition using following edge behavior so that it may locate, find and reach the specified target has depicted in Figure 6.4(b). The wall following behavior mode will be adopted when the mobile robot detects an obstacle in the front while it is moving towards target, the mobile robot may turn left or right because presence of obstacle in the front. In this case, the robot tries to maintain perpendicular to the wall.

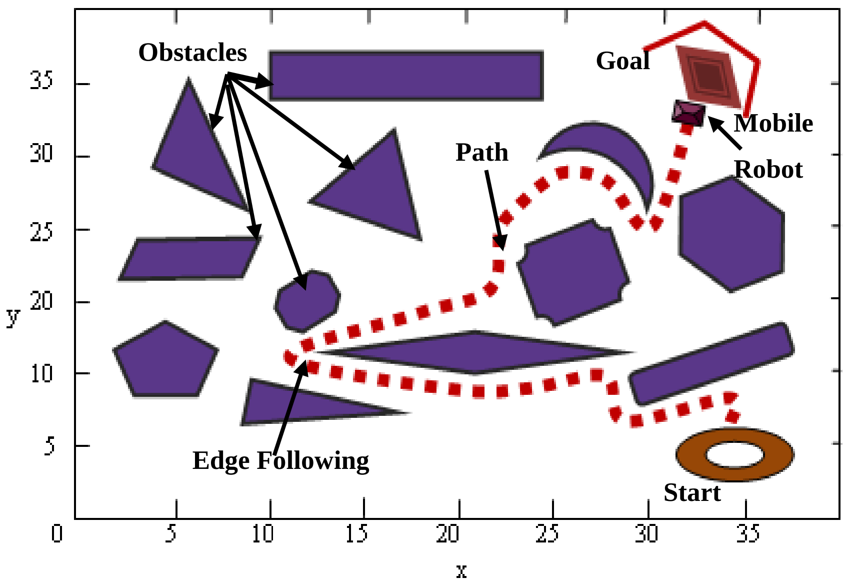



**Figure 6.3: Obstacle Avoidance and Edge following during path navigation using Fuzzy-Neural Algorithm**

When the robot moves through a large U-shaped obstacle (Figure 6.4(b)), it wants to reach directly towards the target, but it senses to be trapped in obstacle. To escape from this trap, robot has to follow the wall of obstacle; the gradual increment of target orientation has also to be checked at each point of path; and trajectory should not be repeated. Edge following behavior and a balance between wall-following and target seeking behavior successfully leads robot towards goal. This simulation result (Figure 6.4(b)) shows superior performance than (Figure 6.4(a)) by Wang and Liu [106] in terms of path length.

The fuzzy based minimum risk method [106] has been examined and compared with the proposed fuzzy-neural controller in a similar navigational environment. It has been found that the fuzzy-neural controller gives a more optimized path than the typical “trial-and-return” method based fuzzy controller (the total path length using a fuzzy controller by Wang and Liu [106] is 251 pixels in Figure 6.4(a) to reach the target, whereas the total path length using a proposed fuzzy-neural controller is 213 pixels in Figure 6.4(b)). In addition, a fuzzy-neural controller requires less computing time and computing memory than a fuzzy controller.

All simulation environments are generated artificially containing one movable robot and static obstacles as well as only one static target.

It has been perceived that the robot controlled through fuzzy-neural control has improved performance than the fuzzy controller in terms of positioning accuracy and collision avoidance and it provides optimize path to reach the goal.

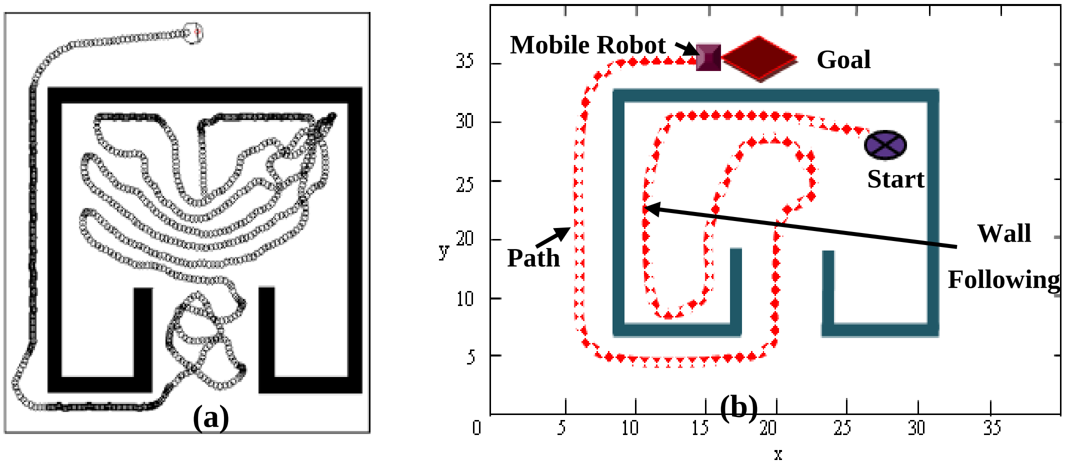



**Figure 6.4: (a) Simulation of minimum risk method in large Concave of U-shaped Environment by Wang and Liu [106]**

**(b) Wall following and Escaping Dead End in a 'U' shaped obstacle using Fuzzy-Neural Algorithm**

To justify the worth and robustness of Fuzzy-Neuro control algorithm, simulation results (Figure 6.5 (b))on mobile robot navigation has been compared with previous research work (Figure 6.5 (a)) where a new FNN is applied to a simulation mobile robot of three DOF by Ma et al. [63]. Comparison is illustrating degree of restricted optimization ability of the controller which resolves that fuzzy-neuro controller can find the definite goal using minimum path length than path traced by Ma et al. [63] in Figure 6.5 (a). So the comparison of performances has shown a good agreement and also verifies the ability and flexibility of the present approach.

**Table 6.2 Simulation result comparison between the fuzzy controllers developed by Wang and Liu [106] and the current developed fuzzy-neural approach**

| Sl.No. | Navigation by different Method | Path Length in pixel | Percentage of Deviation |
|---|---|---|---|
| 1 | Fuzzy based minimum risk method proposed by Wang and Liu [106] (Figure 6.4(a)) | 251 | 15.13% |
| 2 | Currently Proposed Fuzzy-Neural Algorithm(Figure 6.4(b)) | 213 | |

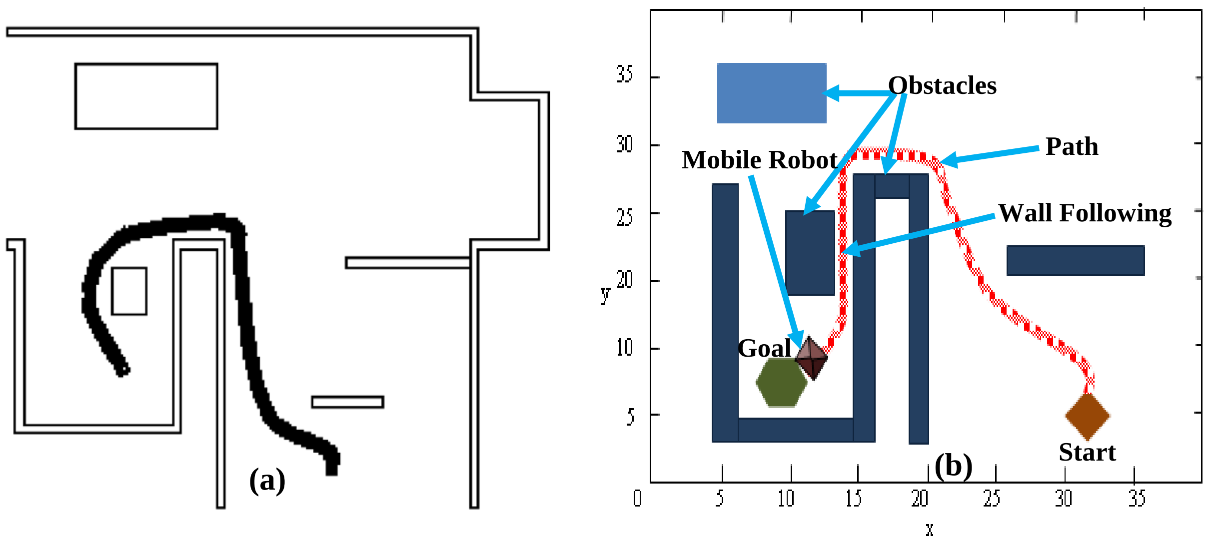



**Figure 6.5 (a) Mobile robot reference trajectories by Ma et al. [63]**
**(b) Simulation result by developed fuzzy neural controller.**

## 6.4 Experimental Result:

The experimental work has been made by loading the Fuzzy-Neuro algorithm into the developed mobile robot through software and hardware interface (Details will be given in Chapter 7). After learning and training, Fuzzy-Neuro algorithm is able to provide Left wheel and Right wheel velocities and optimized steering angle which are sufficient to avoid obstacles and achieve target in real world environment.

During experiment, paths tracing by the robot are made in same way as previously done in chapter 4 and 5. Table 6.3 shows the path length taken by the robot in simulation and in the experimental tests during finding the targets as well as the path length measured from simulation result by Ma et al. [63] (Figure 6.5(a)). Comparison shows a reasonable performance of developed algorithm.

The path lengths are taken in average from 9 different experiments which have been performed in environmental scenario as shown in Figure 6.6. Elementary as well as significant robotic behaviors have been addressed in both simulation and experimental modes employing fuzzy-neural approach.

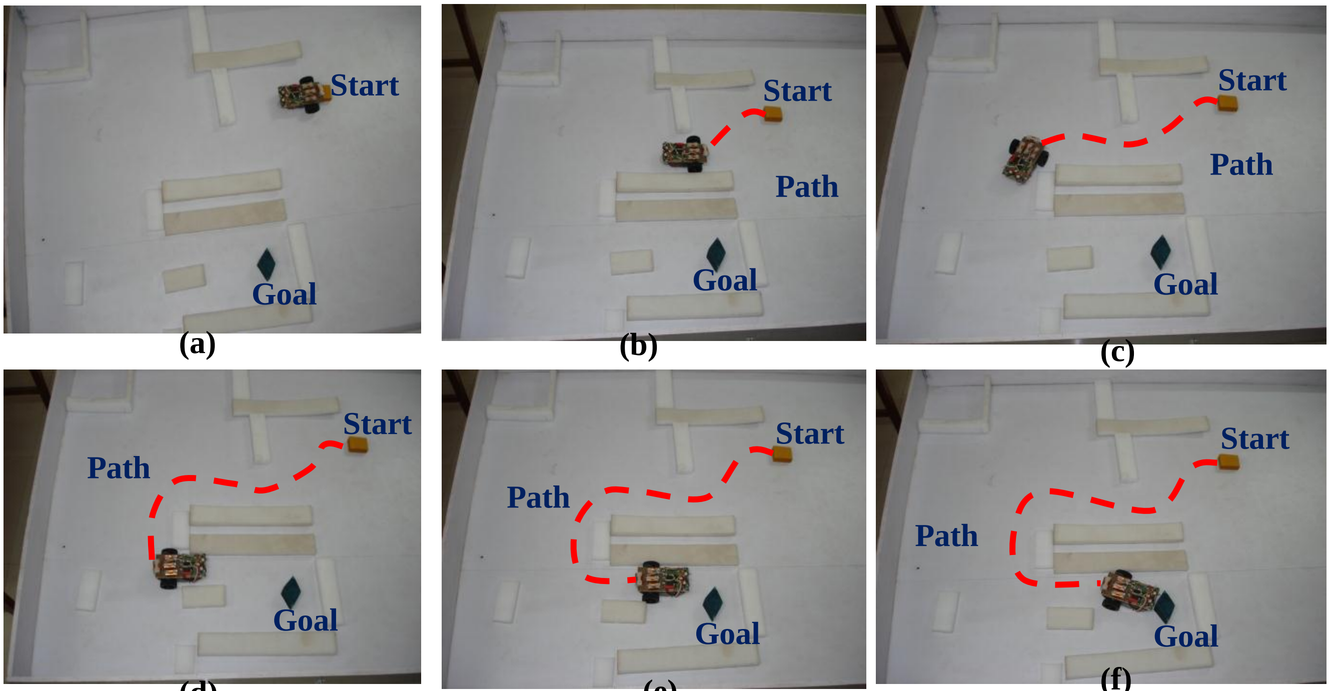



**Figure 6.6: Experimental Results Using Fuzzy-Neuro Controller in comparison with simulation result (Figure 6.4(a)) by Ma et al. [63]**

It has been found that the results obtained from experimental setup are more close to results obtained from simulation mode (shown in Figure 6.4(b)) which validate the proposed method. FLC along with Neural network affords much more rapid response in an unidentified environment and has less computational effort than other conventional approaches.

**Table 6.3: Path Length traced by Robot in Simulation and Experiment for Fuzzy-Neuro Control Algorithm**

| **Path Length in simulation by proposed algorithm (in pixel)** | **Path Length in Experimental mode by proposed algorithm (in pixel)** | **Path Length in simulation of Previous Research work by Ma et al.[63] (in pixel)** | **% of error between simulation and experimental results by proposed technique** |
|---|---|---|---|
| 193 | 221 | 197 | 12.669% |

## 6.5 Conclusion:

On the basis of hypothetical, simulation and experimental investigations, some precious features of Fuzzy-Neuro algorithm in mobile robot navigation can be briefed here:

1. It has been seen that, by using the Hybrid Fuzzy-Neuro controller the robots are able to avoid any obstacles, escape from dead ends, and find target in complex hazardous environments.
2. Various navigational control strategies (e.g. obstacle avoidance, wall and edge following and target seeking) have been addressed in simulation and experimental environments (Figure 6.3, 6.4(b), 6.5(b) and 6.6) using developed controller.
3. Training patterns of Back-Propagation Algorithm based neural network can be generated by simulation rather than by experiment, saving considerable time and effort.
4. Fuzzy-neural controller is proven to confer more optimised path than simple fuzzy controller by a comparison which has already been performed against simulation result by Wang and Liu [106] using fuzzy based minimum risk method in Figure 6.4 and Table 6.2.
5. Comparison of developed algorithm (Table 6.3) with simulation result by Ma et al. [63] employing FNN technique (Figure 6.5(a)) both in simulation (Figure 6.5(b)) and experimental (Figure 6.6) environment delivers a good performance measure concerning veracity of the method.

This hybrid approach has been trialed for achieving a resolution of reducing inaccuracy in steering angle and optimization with respect to path length and time in both simulation and experimental mode. This issue has partially been solved here.

# 7 Hardware Analyses of Mobile Robot Configuration

Robotics is about fashioning systems or can be declared as an art of science combining different engineering skills to make it intelligent and robust. By definition, robot is a mechanical device that is designed and programmed to carry out instruction and perform particular duties automatically, along with required speed and precision.

## 7.1 Introduction

In recent years, the rise of popularity of the single-chip microcomputer and the drastic reductions in size and cost of integrated circuits have opened up wide field areas of creating intelligent systems. However, assembling of robot requires more expertise where the designers should own compendium of basic skills from various fields such as mechanical engineering, electrical and electronics engineering, computer engineering, mechatronics, nanotechnology and artificial intelligence.

A mobile robot comprises of three main parts: mechanics (motion), hardware (sensors, processing and control units) and software (motion control and decision making). Every robot needs microcontroller(s), sensors and actuators (or motors) to interact with the world around it. The most often used sensors are able to measure the distance robot-target, or only detect the obstacles. The sensor-microcontroller communication is a common technical problem along with delay in sensor response time. Responses from Infrared (IR) and ultrasonic (Sonar) sensors are non-linear and depend on the reflectance characteristics of the object surface [90].

The first autonomous mobile robot was built by William Grey Walter around 1950 at the Burden eurological Institute in Bristol named ELMER and ELSIE (Electro Mechanical Robot, Light Sensitive) which is made of 8-bit microcontroller such as Microchip PIC16F690. Another great invention "GRASMOOR" was built at the University of Manchester controlled by MIT 6270 controller, which is based on Motorola 6811 microprocessor [67].

The robot’s most important specifications regarding dimension can be generalised as follows:

- Maximum and minimum dimensions: there is no maximal length, the width has to be between 200*mm* and 450*mm* and the maximum height is 500*mm*.
- Drive; only electrical power is authorized, so no combustion engines may be used.
- Locomotion; all robots need to use wheels. These wheels need to be big enough to cope with small irregularities of the floor.

## 7.2 Integration of Independent Subsystems:

An autonomous mobile robot is an assimilation of independent subsystems (in Figure 7.1), such as: Driving subsystem, Sensing subsystem, Processing unit or Brain compatible with Human Intelligence, Energy supply.

Chassis is a significant part used to support the fitting of microcontroller board, drivers i.e. motors, sensors, batteries and other relevant additional parts that work together. Chassis should be electrically insulated in nature from whole assembly. It should be capable to withstand the weight of all other components embedded on it. In this research work, rectangular chassis made of thin Tin sheet, properly insulated by plastic type material is used as a fundamental structure of the robot. The batteries, microcontroller and sensors are steadily mounted on the top of the chassis. Two motors associated with two fixed standard wheel respectively on a common axis and one castor wheel for support purpose are tightly coupled at the bottom side of the chassis so that robot can attain required motion by maintaining its center of gravity at a constant point of the whole structure.

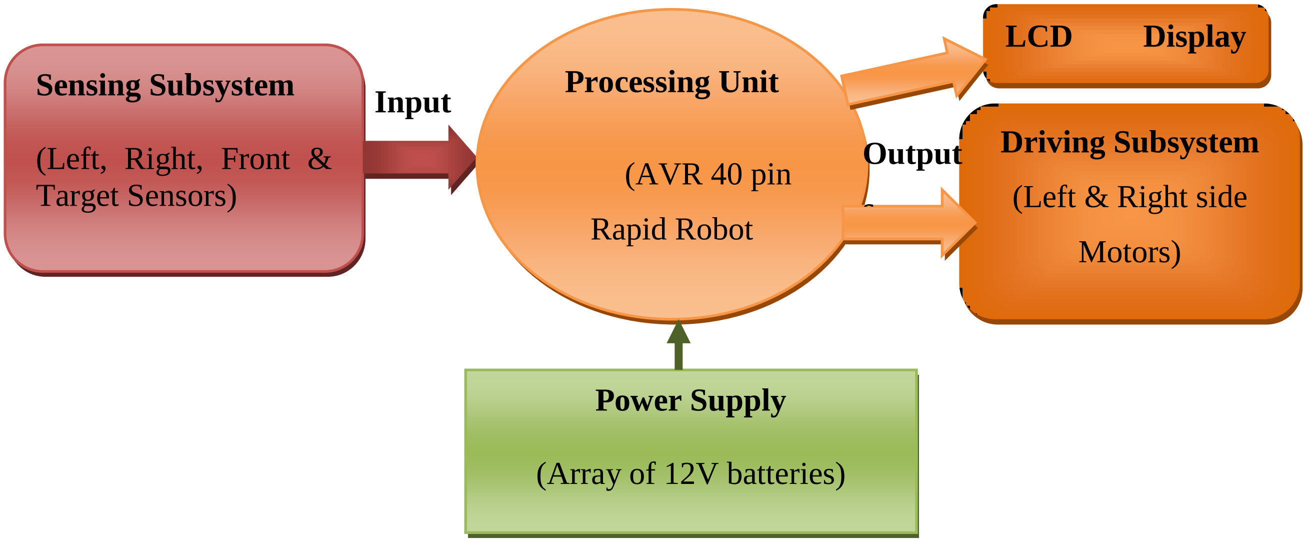



**Figure 7.1: Schematic of Combination of Different Subsystems**

### 7.2.1 Driving subsystem:

According to the requirement of present investigation, here functional driving configuration is most renowned two-wheeled differential drive mobile robot having three wheels minimum: Two "drive wheels", separately controlled by two D.C geared motors, on common horizontal axis which is fixed and One free wheel, (namely "castor" wheel which is a passive one) assure the robot equilibrium, is mounted independently on a vertical axis not on a driven axis of the mobile robot body. The two wheeled differential drive principle has some benefits with respect to other wheel configurations, in this application. The main benefits:

- Only two motors are necessary; the difference between two motors' speed is enough to change the direction of motion.
- No suspension is needed; the three wheel configuration ensures the stable ground contact of chassis.
- The robot can move to a certain position, and rotate in place to achieve a certain orientation.
- Possibility of symmetric design.

The main disadvantages:

- In three wheel configuration, actual centre of gravity of robot may be shifted from midpoint of robot chassis to the wheel axle due to unbalanced load distribution on the chassis which may lead to an unstable assembly.
- Wheel alignment has to be precisely set to guarantee straight line motion.
- Due to physical constraints and rectangular shape, this configuration cannot make turning in a very small radius, which need more floor space for vehicle turning.

All other aspects of differential drive configuration have already been discussed in chapter 3.

❖ **DC geared motor:**

Two terminals of DC motor are connected to from microcontroller pins to receive output commands. This spinning direction of motor can be simply turned to the opposite by changing the voltage polarity. DC motor usually operates at high speed and low torque and generates the highest power level when performing at middle range. Two DC geared motors (shown in Figure 7.2) of 12V independently control two wheels on a common axis.

**Salient Features of D.C geared motor:**

- 60RPM 12V DC motors with Gearbox
- 3000RPM base motor
- 6mm shaft diameter with internal hole
- 125gm weight
- Same size motor available in various rpm
- 2kgcm torque
- No-load current = 60 mA(Max)
- Load current = 300 mA(Max)

**Figure 7.2: Schematic View of 12Volt DC geared motor**

### 7.2.2 Sensing subsystem:

A robot perceives the outside world through its sensors. Using sensors it is able to acquire a map and to localize itself on the map. The most common sensors used for these tasks are range finders using sonar, laser, and infrared technology, cameras, tactile sensors, devices for dead reckoning like wheel en-coders and inertial sensors, active beacons, compasses, and Global Positioning Systems (GPS). However, all these sensors are subject to errors, often referred to as measurement noise. More importantly, most robot sensors are subject to strict range limitations. Performance of sensors can be characterized by a number of properties as follows:

- Sensitivity: ratio of change of output to change of input
- Linearity: measure for the constancy of the ratio of input to output.
- Measurement range: difference between minimum and maximum values measurable.
- Response time: time required for a change in input to be observable in the output.
- Accuracy: the difference between actual and measured values.
- Repeatability: the difference between successive measurements of the same entity.
- Resolution: smallest observable increment in input.
- Type of output.

Most sensors used in robotics are electrostatic ultrasonic sensors since this mechanism is more efficient for coupling into air. Polaroid manufactures the most common type of robotic ultrasonic transducers. It can be used to measure distances from about 0.25 m to 10 m with better precision than IR sensor. A firing pulse triggers an ultrasonic burst from the sensor and starts a counter. The counter is stopped when the sensor, now acting as a receiver, detects a signal above a pre-set threshold. The counter reading thus gives the time of flight. The measuring time with these sensors depends on the air temperature (in air by 20 $C^0$ is 340 m/s) and is relatively large with far obstacles. Poor angular resolution should be avoided.

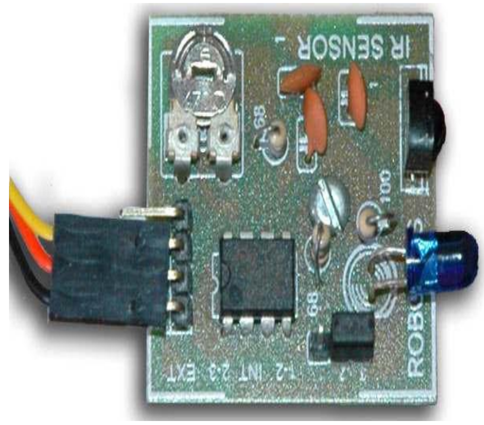
Ultrasonic sensors are safe and easily available, but sometimes the data are difficult to interpret. Infrared sensors are also safe and inexpensive in addition to being easier to use. They are suitable for moderate ranges, where transmitters of up to tens of mill watts can be employed. IR sensors have lower cost, better angular resolution and faster response time than ultrasonic (US) sensors. IR sensor's characteristics are non-linear and the reflected light quality depends on the surface quality and environment light intensity. Some technical parameters have influences on the precision of obstacle avoidance. These types of sensors are good as proximity detectors in robotics. Infrared sensors are appropriate for applications not demanding high measurement accuracies. In most cases the US and IR sensors are used together to improve the robot navigation preciseness. Based upon the study of two most suitable sensor configurations in robotics application, Infrared (IR) sensors are used for obstacle avoidance due to its quick response at the detection of obstacles and Ultrasonic sensor is used for sensing the target present in the environment in the present experiment.

40KHz Ultrasonic Transducer Transmitter and Receiver Pair (Figure 7.3) is used here [88]. These transducers are very useful in making various sensors for detecting obstacles and measuring distances. Salient Features are given below:

- Use for motion or distance sensing
- Frequency: 40kHz ±1.0kHz
- Aluminum case
- Capacitance: 2000pF ±20%
- Transmitter: bandwidth 5.0kHz/100dB,
- Sound pressure level 112dB/40 ±1.0kHz
- Receiver: bandwidth 5.0kHz @ -75dB, min.
- Sensitivity 67dB/40 ±1.0kHz (0dB vs. 1V μbar) R=3.9k.
- Lead length/spacing: 0.28"
- Case size: 0.30"H x 0.43" Dia.

**Figure 7.3: Ultrasonic Receiver & Transmitter Pair**

In this robot design, used multipurpose IR sensor can receive or transmit 38 KHz modulated IR light. Salient features of Multipurpose IR sensor [89] are given below:




- Small size : 30-33mm
- 555 timer for generating output frequency.
- Range setting potentiometer.
- Can differentiate between dark and light colours.
- IR led can be controlled externally though jumper setting (Through microcontroller or PC).
- 3 wire interface for simple obstacle and line sensor.
- 4 wire interface for IR Transreceiver.

**Figure 7.4: Multipurpose IR Sensor**

**Three wire interface of IR Sensor (Figure 7.4):**

- Black wire: GND
- Yellow wire : VCC(+5V)
- Red wire : output terminal which is input to PORTA of ATmega 32

(Logic0: when sensor receives IR modulated light i.e absence of obstacle.

Logic 1: Normally or reflected light intensity is very less i.e presence of obstacle)

**Application:** 1) obstacle detector, 2) line follower, 3) wall follower, 4) RC5 receiver, 5) RC5 transmitter

## 7.2.3 Brain compatible with Human Intelligence:

AVR microcontrollers are popular because of their Linux support and their software like AVRGCC and AVRDUDE. Microcontroller selection criteria for any application depend on its cost, response time and capability of user friendly programming.

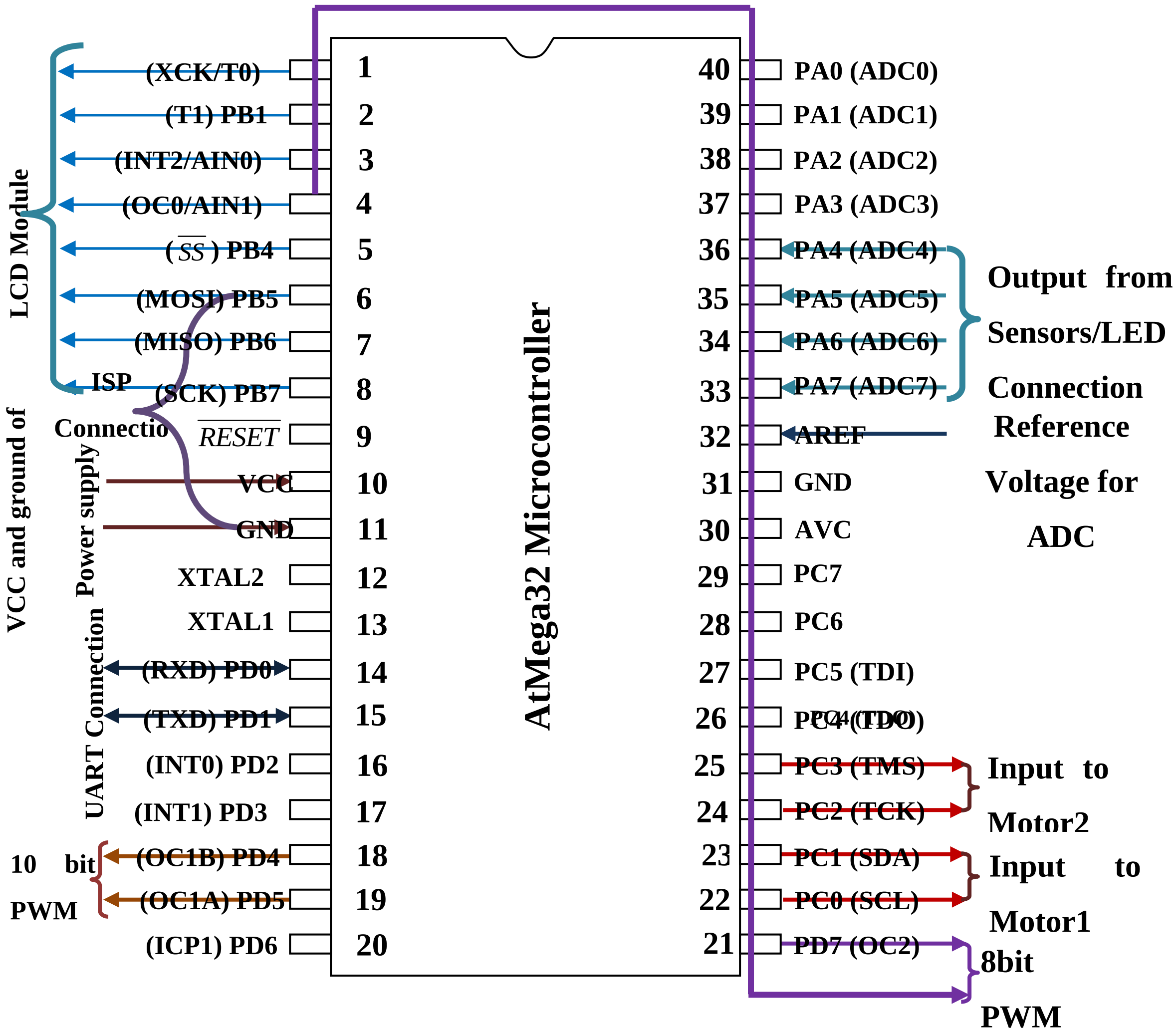



**Figure 7.5: Schematic View of Pin Configuration of AtMega32 Microcontroller**

The processing unit used is AVR 40 pin Rapid Robot Controller containing Atmel ATMega32, 8-bit microcontroller unit which is a versatile EEPROM. 32 K bytes in system programmable flash memory are available here for storage of programs. It has four I/O ports, onboard 10bit ADC and four PWM channels. It can be easily powered by an AC-DC source or battery (6-20V). It can be programmed easily with minimum hardware requirements which make it extremely popular in robotics applications.

Refer to the pin configuration of Atmega32. Each pin can have multiple functions [88]. Atmega 32 has four 8-bit (I/O) Ports. The port's function is to communicate outside the microcontroller. Each port pin consists of three register bits: DDxn, PORTxn, and PINxn. The DDxn bits are accessed at the DDRx I/O address, the PORTxn bits at the PORTx I/O address, and the PINxn bits at the PINx I/O address. The DDxn bit in the DDRx Register selects the direction of this pin. If DDxn is written logic one, Pxn is configured as an output pin. If DDxn is written logic zero, Pxn is configured as an input pin.

Except ADC operation in all other cases there exist only two levels: logic 1/logic 0. Voltage above 1.5 Volt is logic 1 and below 0.7 Volts is logic 0.

- Pin No (1-8): PORTB
- Pin No (9): Reset: On applying ground (logic0) to this port restarts the microcontroller.
- Pin No (10): VCC: This is usually 5 volts for AtMega32, but for AtMega32L it is as low as 2.7 volts.
- Pin No (11): GND: The reference /ground (total system) is connected here.
- Pin No (12-13): XTAL2-XTAL1: here an external crystal oscillator can be connected. The crystal oscillator provides clock cycle for the system to run in more speed than that what is provided by internal RF Oscillator. The maximum speed of AVRs is 16MHz.
- Pin No(14-21): PORTD
- Pin No(22-29): PORTC
- Pin No (30): AVCC: This is along with the GND forms the power supply for the ADC. Whenever we will use the ADC only then we shall give 5 Volts to this pin.
- Pin No (31): GND: This ground is connected to the other ground internally but it is advisable to externally them too, while using the ADC we must connect this GND to the system ground.
- Pin No (32): AREF: This is actually the reference voltage required for analog to digital conversion.
- Pin No (33-40): PORTA

Apart from the normal port function the other important functions are there:

- Pin No (3): INT2: This can be configured to receive interrupt.
- Pin No (4):OC0: At this pin 8bit PWM can be produced.
- Pin No (6-11): MOSI, MISO, SCK, RESET, VCC and GND: All these pins are together used to program the microcontroller by connecting to suitable pins of parallel ports. This ISP connection (6 pin male header) is used to interface between C program in AVR and AtMega 32.
- Pin No (14-15): RXD, TXD: These two pins can be configured for serial data communication with another microcontroller, a PC or any suitable device supporting serial data communication(UART connection)
- Pin No (16-17): INT0, INT1: This pin can be configured to receive interrupt.
- Pin No (18-19): OC1A, OC1B: At these pins 10-bit PWM can be produced using Jumpers PWM1 and PWM2 for speed control of motors.
- Pin No (21): OC2: At this pin 8bit PWM can be produced.
- Pin No (22-29): Motor connection: 4 DC motors/ 2 Stepper motors can be operated using Two L293D motor drivers. Two DC motors can also be controlled by PWM of AVR or at full speed by PWM1 and PWM2 selection jumpers.
- Pin No (33-40): 8 ADC Channels: If configured PORT acts as 8 channel ADC. It receives analog Voltage and converts them into 10 bit/8 bit binary numbers.

The two motors are connected to pins of PORT C to receive the outputs from AVR 40 pin Rapid Robot controller. Left side motor (Motor1) can be controlled by PC0 and PC1. If speed control is activated by PWM1 jumper then speed can be controlled by OC1A pin. Right side motor Motor2 can be controlled by PC2 and PC3. If speed control is activated by PWM2 jumper then speed can be controlled by OC1B pin.

### 7.2.4 Energy Supply:

The whole configuration needs an energy supply to be able to perform tasks autonomously. The energy is stored in accumulators or batteries. In fact there is a difference between these terms, the first ones are rechargeable, and the others are not. In this thesis the terms are used redundantly, and we assume rechargeable batteries.

The dimensions, voltage, weight and recharge method of the battery configuration form restrictions on the entire design process and determine directly the autonomy of the robot. To achieve requirements on energy supply, a symmetric battery pack can be designed to increase the stability of the robot. A battery must meet certain performance goals such as: quick discharge and recharge capability, long cycle life (the number of discharges before becoming unserviceable), low costs, recyclable, high specific energy (amount of usable energy, measured in watt hours per kilogram), high energy density (amount of energy stored per unit volume), specific power, and the ability to work in wide temperature range. In the present investigation, an arrangement of six batteries, each of an appeasement voltage of 12 volt (usually available in market) is mounted on the chassis and two terminals are connected to supply power for AVR 40 pin Rapid Robot Controller.

## 7.3 Model of Mobile Robot:

Four multipurpose IR sensors are used in sensing subsystem for measuring Front, Left and Right obstacle distance and bearing angle of robot with respect target position. The brain (AVR 40 pin Rapid Robot controller) is also fit on the robot. The power supply is typically an array of batteries.

In this research a differential drive configuration is used in rectangular shaped mobile robot in Figure 7.6. Two DC geared motors of 12V independently control two wheels on a common axis. The distance between the wheels is set as 0.18m, wheel diameter is 0.07m and wheel width is 0.03m. The sensor system consists of four IR range sensors which are equipped on the robot to sense the distance between robot and obstacles and target angle of robot according to target position. This robot has AVR 40 pin Rapid Robot controller to control the speed and orientation of mobile robot for obstacle avoidance, wall following and target seeking behavior of mobile robot.

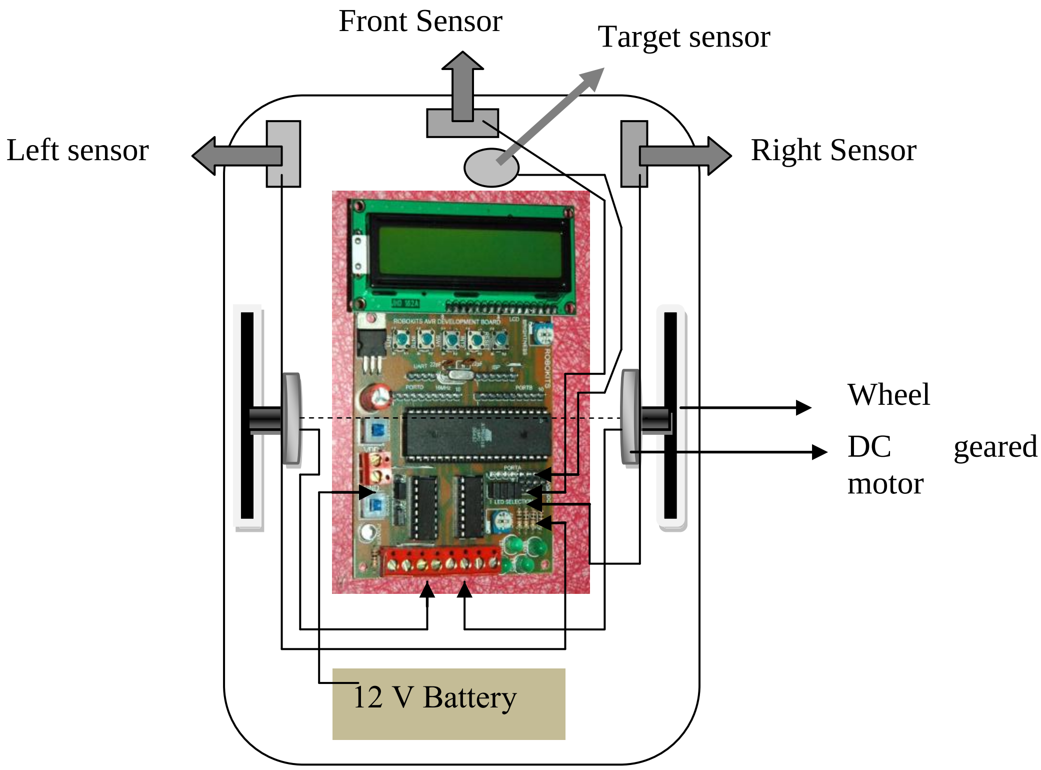



**Figure 7.6: Schematic View of Design of mobile robot with**

**AVR 40 pin Rapid Robot Controller**

In the above figure, Outputs from four multipurpose IR sensors are fed to PORT A of ATmega 32 and output from PORTC are fed into two DC geared motors to move the robot and speed control of the motors using PWM jumpers are also possible. The interfacing between ATmega 32 has done by serial ports. The code is written in C software and compiled in AVRDUDE for the Atmel ATMega32 microcontroller, which is interfaced with the sensors and motors.

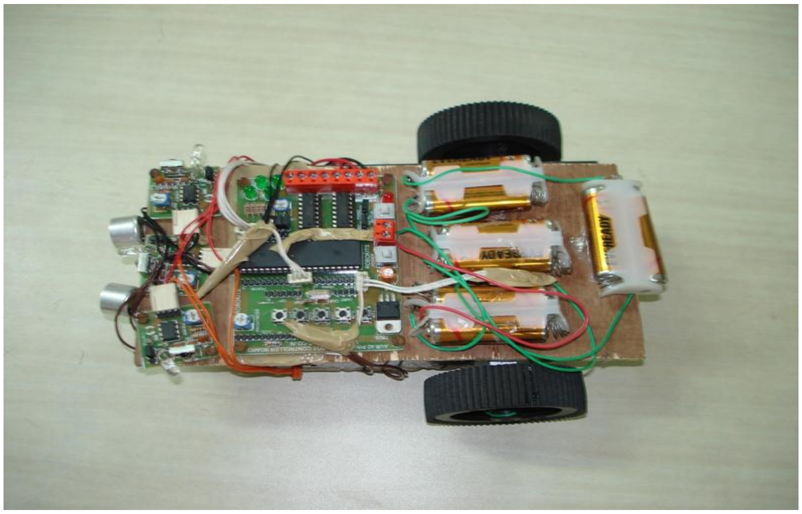


**Figure 7.7: Original View of Model Mobile Robot**

## 7.4 Conclusion

A simplified, light weighted robot configuration has already been implemented here. Study of different sub modules of integrated model mobile robot has been done sequentially in real world environment. Easy interface between hardware and software through At-mega32 microcontroller facilitates the program uploading procedure or the training of mobile robot to cope with different environmental scenarios. Accurate Hardware embodiment of model mobile robot directs towards fruitful navigational performance in experimental mode.

# 8 Results and Discussions

In this present exploration a problem related to navigational path analysis of mobile robot in frequently altering environments has been analyzed. Considering the kinematical stability, AI techniques (e.g. Fuzzy Logic, Neural Network) are used for dynamic, collision-free and optimized path so that mobile robot can reach the target by achieving integration of different preliminary robotic behaviours. This chapter, which is commended to encapsulate the performances of current work done by mentioning the analysis and results of respective chapters for endorsement, can be divided into two main parts as following:

## 8.1 Kinematic and Dynamic Modeling of Mobile Robot:

To grasp exalted expertise in performance, the self-adaptive robot's navigation and path planning algorithm must be consistent with the kinematics of the mobile robot. In another way, as sensory statistics is ineffective to provide vehicle's configuration, it becomes obligatory to obtain stable kinematic and dynamic model for the robot in global and local reference frame respectively.

In chapter three, Kinematic and Dynamic analysis of the mechanical structure of a robot has been conferred concerning the description of the motion with respect to a fixed reference Cartesian frame by ignoring the forces and moments that cause motion of the structure. Modeling of mobile robot is done by combining all kinematic constraints for individual wheels. The different levels of designing wheeled mobile robot can be portrayed as: positioning of the robot model in the environment, maneuverability analysis and holonomicity checking with respect to kinematic constraints, generalized control of developed Kinematic and Dynamic model, and design of control law after solving the trajectory tracking problem using integral backstepping algorithm based on a single Lyapunov function for mobile robot navigation.

The Maneuverability or degree of freedom deals with the possible motions that the robot may follow to reach a final configuration. Modeling of mobile robot with differential drive wheels as control systems has been addressed with a differential geometric point of view by considering only the conventional postulate of "rolling without slipping" (Figure 3.6). Such a robot can rotate on the spot (i.e., without moving the midpoint between the wheels), provided that the angular velocities of the two wheels are equal and opposite. The developed kinematic calculations of positions, velocities and accelerations have already been applied to calculate the dynamic forces and torques produced by the motion of the robot components. The dynamic model has been grown assuming zero gravitational force component as the trajectory of the mobile base is constrained to horizontal plane and no surface friction presents during movement.

## 8.2 Study of Reactive Behaviours by Employing Different Navigational Techniques:

Reactive navigational analysis of mobile agent handles more critical troubles in real world environment than the problems regarding kinematic or dynamic instability of the robot configuration. So, selection of navigational techniques has worthy significance in the research area of mobile robotics discipline. All forms of robotic behaviours; it may be elementary or complex (e.g. obstacle avoidance, wall following, target searching); depend on intelligence of the controller to get collision free navigational path. This research is devoted to assess the enactments of fabricated controllers such as Mamdani as well as Takagi-Sugeno based Fuzzy controller and hybrid Fuzzy-Neuro Controller during navigation of mobile robot in different simulation and experimental environmental scenarios along with comparison with previous research work for ratification.

In chapter four, Reactive behavioural strategy as Robotic control architecture has been delineated to construct and develop an autonomous navigation. The significance, benefits, drawbacks and effectiveness of the architecture has been conferred here.

Fuzzy navigation technique, which allows a reactive control to engender reasonable direction and velocity maneuvers of the autonomous mobile robot, is instigated here for achieving reasonable behavioural performance in static terrains. A Mamdani based Fuzzy logic controller (FLC), a hybrid of different membership functions (Figure 4.4), has been recognised to be more compatible with the reasoning process of human behaviours. Fuzzy behaviour-based architecture for mobile robot navigation in unknown environments incorporates design of Rule base considering basic behaviours for mobile robot navigation: goal seeking, obstacle avoidance, wall following and deadlock disarming etc.

The hypothetical analysis provides the requirements for the design of a suitable fuzzy rule base, in order to assure the asymptotical stability of the robot system. Simulation and experimental studies (Figures 4.5, 4.6, 4.9, 4.10, 4.11 and 4.12) on the developed Mamdani fuzzy controller of the robot system are conducted to investigate the system performance. The presented extensive experiments shows that the developed behaviour based robot is capable of achieving the target by avoiding static as well as dynamic obstacle satisfactorily. The proposed methodology has been compared with previous work presented by many researchers (in Table 4.5), which shows a noble agreement.

Chapter five confers the execution and evaluation of navigational operation of Takagi-Sugeno based fuzzy controller. Takagi-Sugeno type of FIS is analysed by using Matlab toolbox. In this analysis, rule base and membership functions are retained same as Mamdani FIS, but the defuzzification process is different. The output is taken as either linear combination of input variables or a constant value. There is no need to assign any membership functions to the output variables. Lesser numbers of rules are required. Considering gains and shortcoming of this model, it has already been compared with Mamdani-FIS regarding hypothetical as well as navigational performance in simulation mode (Figure 5.4 and Table 5.1). Simulation and experimental result and also comparison with previous research work (Figures 5.5 and 5.6 and Table 5.2) verify the effectiveness of the developed navigation algorithm.

In chapter six, a multilayer feed forward neural network using the principle of back propagation algorithm next to FLC has been employed to increase the accuracy in steering angle measurement by eliminating uncertainty presents in target sensor reading. The neural network has results of FLC as well as environmental information i.e. sensor reading as input. Mainly three behaviours (obstacle avoidance, wall following and target seeking) are addressed here. 200 training patterns (some of them are Table 6.1) are used for designing an intelligent well trained fuzzy-neuro controller for mobile robot being used to navigate in a cluttered environment.

The purpose of this exposition is to construct a generalized framework that integrates both neural networks and fuzzy inference systems Models pertaining to this framework possess both the learning capability of neural networks and the structured knowledge representation employed in fuzzy inference systems. The algorithm also produced acceptable results in simulation as well as real time experiment when tested with different kinds of static obstacles (Figure 6.3, 6.4(b), 6.5(b) and 6.6). Fuzzy-neural controller is verified to give more optimised path than simple fuzzy controller by a comparison which has already been performed against simulation result by Wang and Liu [106] using fuzzy based minimum risk method in Figure 6.4 and Table 6.2. Comparisons with previous research work using FNN (Figure 6.5 and Table 6.3) regarding performance measure are of pleasing nature.

In chapter seven, hardware aspect of a simple mobile robot configuration by accumulating different sub modules are illustrated here. Applied Sensing and Driving subsystems are discussed in detail. Interface between hardware and software through At-mega32 microcontroller has been clearly explained. Proper Hardware implementation of model mobile robot leads to the successful experimental verification of specified navigational algorithms.

Further assignment has been carried out to compare the performances of the Mamdani as well as Takagi-Sugeno based Fuzzy Logic Controller embedded with five hybridized membership functions (Trapezoidal, Triangular and Neural), and Fuzzy-neuro approach (Figure 8.1) in simulation as well as experimental mode. Only one robot, ‘U’ shaped fixed obstacle and one fixed point target are present in these scenarios. During comparison, the path length of “212 pixel”, “193 pixel” and “182pixel” are recorded for Mamdani-FLC, Takagi-Sugeno FLC and Fuzzy-Neuro controller respectively. Time taken to reach the target “23.87 second”, “20.15

second" and "18.57 second" are recorded for Mamdani-FLC, Takagi-Sugeno FLC and Fuzzy-Neuro controller respectively.

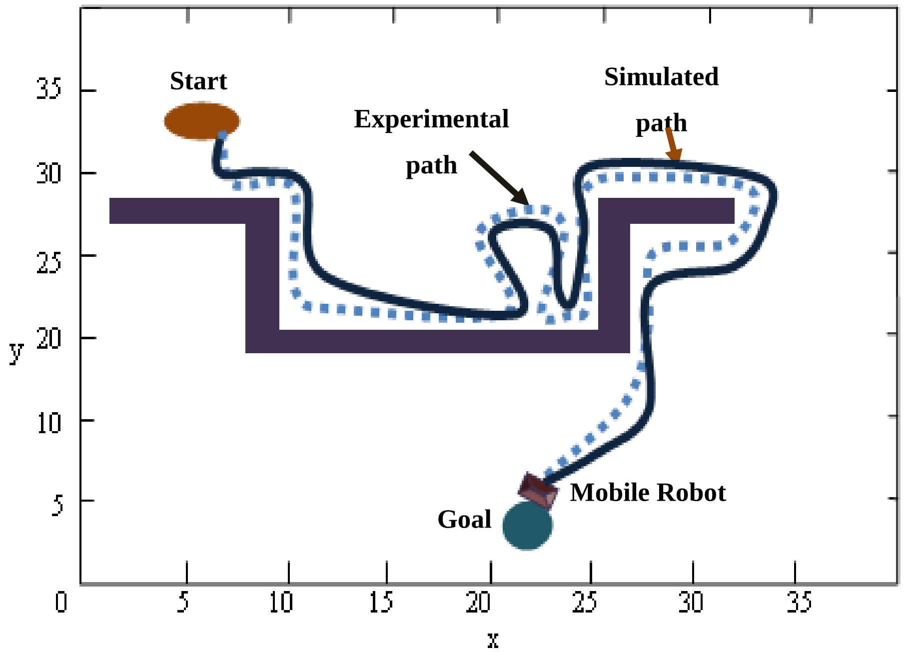



**Figure 8.1(a): Comparison in Simulation and Experimental Path Analysis using Mamdani based Fuzzy Controller**

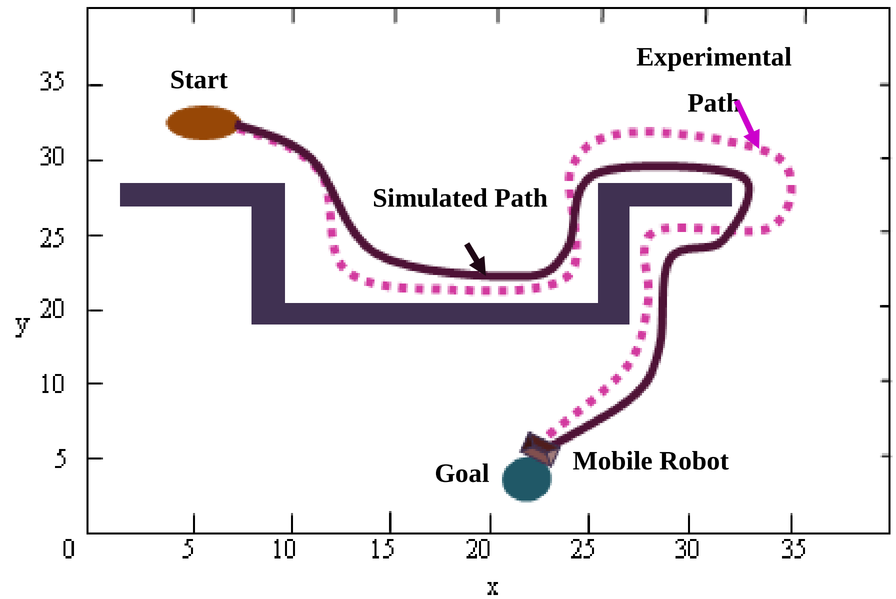



**Figure 8.1(b): Comparison in Simulation and Experimental Path Analysis using Takagi-Sugeno based Fuzzy Controller**

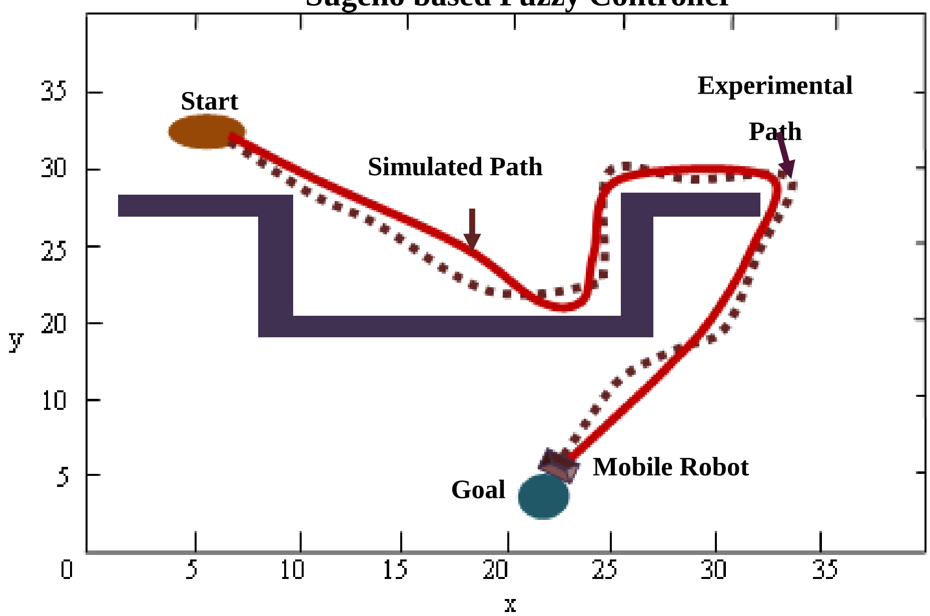



**Figure 8.1(c): Comparison in Simulation and Experimental Path Analysis using Fuzzy-Neural Controller**

**Table 8.1: Deviation of Travelled Path and Time Taken during Simulation and Experimental Mode for different control approach**

| Sl.No. | Navigational Analysis with various Controller | Percentage of Results deviations Simulation Vs Experimental mode | |
|---|---|---|---|
| | | Path | Time |
| 1. | Navigation with Mamdani based Fuzzy controller | 15.03% | 14.93% |
| 2. | Navigation with Sugeno based Fuzzy controller | 13.66% | 11.59% |
| 3. | Navigation with Fuzzy-Nerual controller | 12.67% | 13.81% |

The results obtained from various approaches are given in Table 8.1. It shows the percentage of deviation of experimental results with respect to simulation result in various controllers being used for finding the navigational path length and time taken to reach the target by mobile robot.

The solution obtained from current research leads to partially optimized navigational path analysis of mobile robot in various environments. In addition, the navigation system can be utilised in numerous applications in industrial and medical environments.

# 9 Conclusions and Future Work

The major intents of this research work have been to find out efficient control techniques for mobile robot navigation in crowded real world situations by avoiding collision with obstacles arranged in a chaotic way. This chapter recapitulates the main contributions, conclusions of the present investigations and space for additional works. This investigation anticipates for making the following contributions to the domain of navigational path analysis of mobile robots in diverse environments.

## 9.1 Contributions:

- In the kinematic analysis of mobile robot, global reference frame as well as the robot's local reference frame, has been considered for robot motion. Left wheel and Right wheel velocities of the mobile robot has also been calculated. From the wheel velocities, steering angle for the robot can be easily assessed.
- By analyzing kinematic constraints for individual wheels, robot's mobility, maneuverability and holonomicity have also been derived.
- Modeling of mobile robot is done by combining all these kinematic constraints. The dynamic model accounts for the reaction forces applied on a point of the robot and describe the relationship between the linear and angular velocities and the generalized forces and torques acting on the robot. These models are expressed in a canonical form which is convenient for design of planning and control techniques.
- In chapter four, Based on sensory information innovative mamdani-based fuzzy reactive controller has been developed for static obstacle avoidance, escaping from local minima problems, and seeking target during navigation of mobile robot in a complex hazardous environment. Simulation and Experimental results are presented and very good agreement is observed between them.
- Designed Fuzzy rule base is also applied for Takagi-Sugeno based fuzzy model to get more optimize wheel velocities of mobile robot as in this approach defuzzification

procedure is not required, crisp value is the direct output here. Comparisons between Mamdani and Sugeno based fuzzy model for navigation purpose has also been done.

- A neural network controller has been employed next to the FLC for mobile robot navigation. It is found that this hybrid approach is better than the fuzzy controller in terms of resolving uncertainty.
- Hardware Analysis of model mobile robot has also been carried out. Proper interface between navigational techniques and hardware components of autonomous mobile robot results in a successful experimental work done.

## 9.2 Conclusions:

In this research scheme, the challenge has been taken to solve a problem related to navigational path analysis of mobile robots in numerous inconsistent environments. From the current investigation clarified in this thesis the conclusion extracted are as follows:

- Sensors do not yield any data on the vehicle's configuration, it is necessary to develop stable kinematic and dynamic model for the robot in global and local reference frame respectively. The kinematic model of a mobile robot is essentially the description of the admissible instantaneous motions in respect of the constraints. The proposed controller is claimed to be robust against the changes in mass and inertia parameters of robot. The proposed dynamic controller can be used for tracking the desired velocity, which is generated by kinematic controller, without exact knowledge about the dynamic model of a mobile robot.
- The inference mechanism accompanied by charted fuzzy rule base (implemented for both Mamdani and Takagi-Sugeno Model) gives a navigational control scheme, which indirectly addresses the demand of determining the sequence of actions such as to recognise the environment, to avoid obstacles and to achieve the goal successfully. Performance measure has been carried out through the comparison between simulation and experimental results for different environmental scenarios in terms of consumed time.

- Fuzzy-Neuro approach also puts into practise for enhancing the navigational path analysis and planning performance of the robot and it gives a satisfactory consequence over simple FLC.

This research is committed to appraise the performances of fabricated controllers during navigation of mobile robot in different simulation and experimental environmental scenarios along with comparison with previous research work for endorsement.

## 9.3 Future Works:

The current effort affords a base for forthcoming growth of cohesive designing approaches of sensible controller based on artificial intelligence technique enriched with human perception. Regardless of all research that has been conducted, autonomous navigation in various environments is still an open area of research. The suggestions with several crucial and promising researches for future investigation are as follow:

- To make developed algorithms more effective in dealing with unpredictable real life situations, further development of the techniques may be required for the avoidance of moving obstacles as well as other robots present in the scenario.
- The navigational techniques developed in this research work are capable of detecting and reaching the static targets. Further modifications in these navigational techniques may be carried out so that the robots can not only detect dynamic targets but also reach them using an optimum path.
- The robot must monitor the position of the obstacle in each time interval and then predict the next position of the obstacle, based upon the trajectory of the obstacle. The robot must then avoid that position. This approach may be achieved.

Further research required to design an intelligent controller for co-operative mobile robot to carry out navigational task by avoiding static as well as moving obstacles and follow optimized path for reaching and handling a particular object.

# References


1. Abiyev R., Ibrahim D. and Erin B., “Navigation of mobile robots in the presence of obstacles”, Advances in Engineering Software, Vol. No. 41, pp. 1179–1186, 2010.
2. Ailon A. and Zohar I., “Point-to-point control and trajectory tracking in wheeled mobile robots: some further results and applications”, in: Proc. Intentional Federation Automatic Control Conf., Seoul, Korea, 2008.
3. Ailon A. and Zohar I., “Control strategies for driving a group of nonholonomic kinematic mobile robots in formation along a time-parameterized path”, to be published in IEEE/ASME Transactions on Mechatronics, (posted on IEEE Xplore) 2011.
4. Aguirre E. and Gonzalez A., "Fuzzy behaviors for mobile robot navigation: Design, coordination and fusion," International Journal of Approximate Reasoning, vol. 25, pp. 255, 2000.
5. Alexander J.C., and Maddocks J.H. “On the Kinematics of Wheeled Mobile Robots”, The International Journal of Robotics Research, 1989, Vol. 8(5), pp. 15-27.
6. Amidis I.S. and Roberts G. N., “Fuzzy Modelling & Fuzzy-Neuro Motion Control of an Autonomous Underwater Robot”, In Proc. of IEEE International Conference on AMC '98 – COIMBRA, pp. 641-646.
7. Arkin, R. C., “Integrating behavioral, perceptual, and world knowledge in reactive navigation”, in `Robotics and Autonomous Systems 6 (1990) 105-22.
8. Arvin Farshad, Samsudin Khairulmizam, and Nasseri M. Ali, “Design of a Differential-Drive Wheeled Robot Controller with Pulse-Width Modulation”, IEEE Conference on Innovative Technologies in Intelligent Systems and Industrial Applications (CITISIA 2009) Monash University, Sunway campus, Malaysia, 25th & 26th July 2009, pp.143-147.

9. Babalou A. and Seifipour N., “Application of Fuzzy Decision Making in Mobile Robot Navigation in Dynamic Environments”, FUZZ-IEEE 2009, Korea, August 20-24,2009, pp.877-881.

10. Batlle J.A. and Barjau A., “Holonomy in mobile robots” Robotics and Autonomous Systems, 57 (2009) 433–440.

11. Bloch A.M., Reyhanoglu M. and McClamorch H.,“Control and stabilization of nonholonomic dynamic systems”, IEEE Tran. on Automatic Control, 37, 11, pp. 1746-1757, (1992).

12. Boem H. R., and Cho H. S., “Mobile Robot Localisation using a Single Rotating Sonar and 2 Passive Cylindrical Beacons”, Robotica, 1995, Vol.13(3), pp. 243 - 252.

13. Borenstein J., Everett H.R., Feng L. and Wehe D., “Mobile robot positioning Sensors and techniques”, Journal of Robotics Systems 14 (4) (1997) 231–249.

14. Boubertakh Hamid, Tadjine Mohamed, Glorennec Pierre-Yves and Labiod Salim, "A Simple Goal Seeking Navigation Method for a Mobile Robot using Human Sense, Fuzzy Logic and Reinforcement Learning”, Journal of Automatic Control, VOL. 18, pages: 23-27, 2008.

15. Brooks, R. “A robust layered control system for a mobile robot”, in Proceedings of the IEEE International Conference on Robotics and Automation', (1986), Vol. RA-2, pp. 14.23.

16. Campion Guy, Bastin Georges, and Brigitte D’ AndrCa-Novel, “Structural properties and classification of kinematic and dynamic models of wheeled mobile robots, IEEE Transactions on Robotics and Automation 12 (1) (1996) 47–62.

17. Chakraborty Nilanjan and Ghosal Ashitava, “Kinematics of wheeled mobile robots on uneven terrain”, Mechanism and Machine Theory 39 (2004) 1273–1287.

18. Chang Su Lee, “A Framework of Adaptive T-S type Rough-Fuzzy Inference Systems”- A Doctoral Dissertation by, The University of Western Australia, July 2009.

19. Chaoxia Shi, Wang Yanqing and Yang Jingyu, “A local obstacle avoidance method for mobile robots in partially known environment”, Robotics and Autonomous Systems 58 (2010) 425_434.

20. Chung Yongoug, Park Chongkug and Harashima Fumio, “A Position Control Differential Drive Wheeled Mobile Robot”, IEEE Transactions on Industrial Electronics, Vol. 48, No. 4, August 2001, pp.853-863.

21. Deng M., Inoue A., Sekiguchi K. and Jiang L., “Two-wheeled mobile robot motion control in dynamic environments”, Robotics and Computer-Integrated Manufacturing 26 (2010) 268–272.

22. Dixon W. E., Jiang Z. P. and Dawson D. M., “Global exponential set point control of wheeled mobile robots: a Lyapunov approach”, Automatica 2000, Vol. 36, pp. 1741-1746.

23. Do K. D., Jiang Z. P. and Pan J., “Simultaneous Tracking and Stabilization of Mobile Robots: An Adaptive Approach”, IEEE Transactions on Automatic Control, Vol. 49, No. 7, July 2004, pp.1147-1152.

24. Eghtesad M. and D.S. Necsulescu, “Study of the internal dynamics of an autonomous mobile robot”, Robotics and Autonomous Systems, 54 (2006) 342–349.

25. Er M.J., Tan Tien Peng and Loh Sin Yee, “Control of a mobile robot using generalized dynamic fuzzy neural networks”, Microprocessors and Microsystems 28 (2004) 491–498.

26. Evans J., Krishnamurthy B., Barrows B., Skewis T., and Lumelsky V., “Handling Real-World Motion Planning, A Hospital Transport Robot”, IEEE Transactions on Control Systems 12(1992) 15 - 20.

27. Everett H., and Flynn A,. “A Programmable Near-infrared Proximity Detector for Mobile Robot Navigation”, SPIE Conference on Advances in Intelligent Robotics Systems, 1986, Vol. 727, pp. 221 - 230.

28. Fierro R. and Lewis L., “Control of a nonholonomic mobile robot: backstepping kinematics into dynamics”, proc. of the 34th IEEE Conf. on Decision & Control, New Orleans, pp. 3805-3810, (1995).

29. Fliess M., Lvine J., Martin P. and Rouchon P., “Flatness and defect of non-linear systems: introductory theory and examples”, International Journal of Control, 61 (1995) 1327–1361.

30. Gandhi P.S. and Ghorbel F., “High-speed precision tracking with harmonic drive systems using integral manifold control design,” International Journal of Control, 78(2), 2005, 112–121.

31. Gao Y., Lee C. G. and Chong T.K., “Receding horizon tracking control for wheeled mobile robots with time-delay”, Journal of Mechanical Science and Technology, 22, 2008, 2403-2416.

32. Gas Jorge and Rosetti Ailson,” Uncertainty representation for mobile robots: Perception, modeling and navigation in unknown environments”, Fuzzy Sets and Systems, Volume 107, Issue 1, 1 October 1999, Pages 1-24.

33. Ge SS and Cui YJ, “Dynamic motion planning for mobile robots using potential field method”, Autonomous Robots 2002; 13: 207–22.

34. Gholipour A., Dehghan S.M. and Ahmadabadi M. Nili. “Lyapunov based tracking control of nonholonomic mobile robot”, proc. of 10th Iranian conference on electrical engineering, Tabriz, Iran, 3, pp. 262-269, 2002.

35. Gracia L. And Tornero J., “Kinematic Control of Wheeled Mobile Robots”, Latin American Applied Research, 38 (2008) pp.7-16.

36. Gracia Luis and Tornero Josep, “Characterization of zero tracking error references in the kinematic control of wheeled mobile robots”, Robotics and Autonomous Systems 57 (2009) 565_577.

37. Gobi Adam F. and Pedrycz Witold, “The potential of fuzzy neural networks in the realization of approximate reasoning engines”, Fuzzy Sets and Systems 157 (2006) 2954 – 2973.

38. Gobi A.F. and Pedrycz W., “Fuzzy modeling through logic optimization”, International Journal of Approximate Reasoning, August 2007, Volume 45, Issue 3, pp. 488-510.

39. Godjevac J. and Steele N., “Neuro-fuzzy control of a mobile robot”, Neurocomputing, 28, 1999, 127-143.

40. Gonzalez J., Stenz A. and Ollero A. “A mobile robot iconic position estimator using a radial laser scanner”, Journal of Intelligent & Robotic Systems, 1995, Vol. 13(2), pp. 161 - 179.

41. Gupta M., Kumar T.N., Behera L., Venkatesh K.S. and Dutta A., “Environment modeling in mobile robotics through Takagi-Sugeno fuzzy model”, IET Irish Signals and Systems Conference (ISSC 2009), Dublin, Ireland, 10-11 June 2009, pp. 24-29

42. Han Soonshin, Choi ByoungSuk, Lee JangMyung, “A precise curved motion planning for a differential driving mobile robot”, Mechatronics 18 (2008) 486–494.

43. Haykin S., “Neural Networks a Comprehensive Foundation”, Second Ed., (India: Pearson prentice hall), 2006.

44. Herrero-Pérez D., Martínez-Barberá H., LeBlanc K. and Saffiotti A., “Fuzzy uncertainty modeling for grid based localization of mobile robots”, International Journal of Approximate Reasoning 51 (2010) 912–932.

45. Hong M. L., and Kleeman L., “Ultrasonic Classification and Location of 3D Room Features using Maximum Likelihood Estimation”, Robotica, 1997, Vol. 15(6), pp. 645 - 652.

46. Hou Zeng-Guang, Zou An-Min, Cheng Long, and Tan Min, “Adaptive Control of an Electrically Driven Nonholonomic Mobile Robot via Backstepping and Fuzzy Approach”, IEEE Transactions on Control Systems Technology, Vol. 17, No. 4, July 2009, pp. 803-815.

47. Hui Nirmal Baran and Pratihar Dilip Kumar, “A comparative study on some navigation schemes of a real robot tackling moving obstacles”, Robotics and Computer-Integrated Manufacturing 25 (2009) 810–828.

48. Jang J.-S.R., Sun C.T., Mizutani E., “Neuro-fuzzy and Soft Computing”, Prentice-Hall, Englewood Cliffs, NJ, 1997.

49. Jeronymo Daniel Cavalcanti, Borges Yuri Cássio Campbell and Coelho Leandro dos Santos, “A calibration approach based on Takagi Sugeno fuzzy inference system for digital electronic compasses”, Original Research Article , Expert Systems with Applications, In Press, Corrected Proof, Available online 5 May 2011.

50. Jiang Z.P. and Nijmeijer H., “Tracking control of mobile robots: A case study in Backstepping”, Automatica, 33, 7, pp. 1393-1399, (1997).

51. Jin Yaochu and Jiang Jinping, “Techniques in Neural-network-based Fuzzy System Identification and Their Application to control of complex systems”, Fuzzy Theory Systems Techniques and Applications, 1 (1999) 112-128.

52. Jing Xing-Jian,” Behavior dynamics based motion planning of mobile robots in uncertain dynamic environments”, Robotics and Autonomous Systems 53 (2005) 99–123.

53. Jolly K.G., Kumar R. Sreerama and Vijayakumar R., “Intelligent task planning and action selection of a mobile robot in a multi-agent system through a fuzzy neural network approach”, Engineering Applications of Artificial Intelligence 23 (2010) 923–933.

54. Kanayama Y., Kimura Y., Miyazaki F. and Noguchi T. “A stable tracking control scheme for an autonomous mobile robot”, proc. of IEEE Int. Conf. on Robotics and Automation, pp. 384-389, (1990).

55. Kasabov Nikola K., Kim Jaesoo, Watts Michael J.and Gray Andrew R., “FuNN/2- A Fuzzy Neural Network Architecture for Adaptive Learning and Knowledge Acquisition”, Information Sciences, Vol. 101, 1997, pp. 155-175.

56. Kermiche S., Saidi M.L., Abbassi H.A and Ghodbane H., “Takagi Sugeno Based Controller for Mobile Robot Navigation”, Journal of Applied Sciences, 2006, Vol. 6(8), pp.1838-1844.

57. Khatib Mohannad Al- and Saade Jean J, “An efficient data driven fuzzy approach to motion planning of a mobile robot”, Fuzzy Sets and Systems, 134 (2003) 65-82.

58. Kleeman L., and Kuc R. “Mobile Robot for Target Localization and Classification”, International Journal of Robotics Research, 1995, Vol 14(4), pp. 295-318.

59. Ko J. H., Kim W. J., and Chung M. I., "A Method for Acoustic Landmark Extraction for Mobile Robot Navigation", IEEE Transactions on Robotics and Automation, 1996, Vol. 12(3), pp. 478 - 485.

60. Kube C. R., and Zhang H. "Task Modelling in Collective Robotics", *Autonomous Robots,* 1997, Vol. 4, pp. 53 - 72.

61. Li Wei and Feng Xun, "Behavior fusion for robot navigation in uncertain environment using fuzzy logic ", 1994 IEEE International Conference on 'Humans, Information and Technology', Volume: 2, Page(s): 1790 – 1796.

62. Lin Faa-Jeng, Hwang Wen-Jyi and Wai Rong-Jong, "A Supervisory Fuzzy Neural Network Control System for Tracking Periodic Inputs", IEEE Transactions on Fuzzy Systems, Vol. 7, No. 1, February 1999, pp.41-52.

63. Ma Xiaowei, Li Xiaoli and Qiao Hong, "Fuzzy neural network –based real-time self-reaction of mobile robot in unknown environments", Mechatronics, 2001, Vol.11, pp.1039-1052.

64. Maaref H. and Barret C.," Sensor-based fuzzy navigation of an autonomous mobile robot in an indoor environment", Control Engineering Practice, 8 (2000), 757-768.

65. Mamdani E.H. and Assilian S., An experiment in linguistic synthesis with a fuzzy logic controller, International Journal of Man-Machine Studies, 7, 1975, 1-13.

66. Martins Felipe N., Celeste Wanderley C., Carelli Ricardo, Ma´ rio Sarcinelli-Filho and Bastos-Filho Teodiano F., "An adaptive dynamic controller for autonomous mobile robot trajectory tracking", Control Engineering Practice 16 (2008) 1354– 1363.

67. Matijevics I., "Infrared Sensors Microcontroller Interface System for Mobile Robots", 5th International Symposium on Intelligent Systems and Informatics, Subotica, Serbia, 24-25 August, 2007.

68. Montaner Marc Boumedine and Serrano Alejandro Ramirez-," Fuzzy knowledge-based controller design for autonomous robot navigation", Expert Systems with Applications, 14 (1998) 179-186.

69. Moon Jong-Woo, Park Chong-Kug and Harashima Fumio,“Kinematic Correction of a Differential Drive Mobile Robot and a Design for Velocity Trajectory with Acceleration Constraints on Motor Controllers”, In Proc. IEEE International Conference on Intelligent Robots and Systems, 1999, pp. 930-935.

70. Moustris George P. and Tzafestas Spyros G., “Switching fuzzy tracking control for mobile robots under curvature constraints”, Control Engineering Practice 19 (2011) 45–53.

71. Motlagh Omid Reza Esmaeili, Hong Tang Sai and Ismail Napsiah, “Development of a new minimum avoidance system for a behavior-based mobile robot”, Fuzzy Sets and Systems 160 (2009) 1929–1946.

72. Mucientes M., Alcalá-Fdez J., Alcalá R. and Casillas J., “A case study for learning behaviors in mobile robotics by evolutionary fuzzy systems”, Expert Systems with Applications 37 (2010) 1471–1493.

73. Muir P.F., and Neuman C.P., “Kinematic Modeling of Wheeled Mobile Robots”, Journal of Robotic Systems, 1987, 4(2), pp. 281 – 340.

74. Murray R.M., Walsh G. and Sastry S.S., “Stabilization and tracking for nonholonomic control systems using time varying state feedback”, IFAC Nonlinear Control systems design, pp. 109-114, (1992).

75. Nefti S., Oussalah M., Djouani K. and Pontnau J., “Intelligent Adaptive Mobile Robot Navigation”, Journal of Intelligent and Robotic Systems 30: 311–329, 2001.

76. Nefti Samia, Oussalah Mourad and Kaymak Uzay, “A New Fuzzy Set Merging Technique Using Inclusion-Based Fuzzy Clustering”, IEEE Transactions On Fuzzy Systems, Vol. 16, No. 1, February 2008.

77. Olajubu E.A., Ajayi O.A. and Aderounmu G.A., “A fuzzy logic based multi-agents controller”, Expert Systems with Applications 38 (2011) 4860–4865.

78. Parhi Dayal R., “Navigation of Mobile Robots Using a Fuzzy Logic Controller”, Journal of Intelligent and Robotic Systems, Volume 42, Number 3 / March, 2005.

79. Parhi D R and Singh M K, "Intelligent fuzzy interface technique for the control of an autonomous mobile robot", Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, Volume 222, Number 11 / 2008.

80. Parhi Dayal Ramakrushna, Pradhan Saroj Kumar, Panda Anup Kumar and Behera Rabindra Kumar," The stable and precise motion control for multiple mobile robots", Applied Soft Computing 9 (2009) 477–487.

81. Pavlovskii V.E., Evgrafov V.V., Pavlovskii V.V. and Petrovskaya N.V., "Mobile Robots with Two Independent Coaxial Drive Wheels: Dynamics and the Schemes of Control" published in Vestnik Moskovskogo Universiteta, Matematika. Mekhanika, 2009, Vol. 64, No. 1, pp. 44–50.

82. Pathak P.M., Mukherjee A. and Dasgupta A., "Interaction torque control by impedance control of space robots", Simulation, 85(7), 2009, 451-459.

83. Pourboghrat Farzad and Karlsson Mattias P., "Adaptive control of dynamic mobile robots with nonholonomic constraints", Computers and Electrical Engineering 28 (2002) 241–253.

84. Pradhan S.K., Parhi D.R. and Panda A.K., "Navigation of multiple mobile robots using rule-based neuro-fuzzy technique", Int. J. of Comput. Intell., Vol.3 no. 2, pp. 142-152, 2006.

85. Pradhan S.K., Parhi D.R. and Panda A.K., "Fuzzy logic techniques for navigation of several mobile robots." Appl. Soft Computing, vol. 9, pp. 290- 304, 2009.

86. Precup Radu-Emil and Hellendoorn Hans, "A survey on industrial applications of fuzzy control", Computers in Industry 62 (2011) 213-226.

87. Qing-yong BAO, Shun-ming LI, Wei-yan SHANG and Mu-jin AN, "A Fuzzy Behavior-Based Architecture for Mobile Robot Navigation in Unknown Environments", International Conference on Artificial Intelligence and Computational Intelligence, 2009, vol.2, pp.257-261.

88. Robokits India, "User Manual for AVR 40 pin Rapid", http://www.robokits.org, info@robokits.org.

89. Robokits India, "User Manual for Multipurpose IR Sensor", http://www.robokits.org, info@robokits.org.

90. Saegwart Roland and Nourbakhsh Illah R., “Introduction to Autonomous Mobile Robots”, A Bradford Book, The MIT Press, Cambridge, Massachusetts, London, England, 2004.
91. Saffiotti A., "Fuzzy logic in autonomous robotics: Behavior coordination," presented at 6th IEEE International Conference on Fuzzy Systems, Fuzzy-IEEE'97, Barcelona, Spain, 1997.
92. Samsudin Khairulmizam, Arif Ahmad Faisul and Syamsiah Mashohor, “A highly interpretable fuzzy rule base using ordinal structure for obstacle avoidance of mobile robot”, Applied Soft Computing 11 (2011) 1631–1637.
93. Selekwaa Majura F., Dunlapb Damion D., Shib Dongqing and Collins Jr. Emmanuel G., “Robot navigation in very cluttered environments by preference-based fuzzy behaviors”, Robotics and Autonomous Systems 56 (2008) 231–246.
94. Skewis T., and Lumelsky V., “Simulation of Sensor-Based Robot and Human Motion Planning”, International Journal of Robotics and Automation, 8(4) 1993 153 - 168.
95. Skewis T., and Lumelsky V., “Experiments with a Mobile Robot Operating in a Cluttered Unknown Environment”, Journal of Robotics Systems, 1994, Vol. 11(4), pp. 281 - 300.
96. Song Kai-Tai and Sheen Liang-Hwang, “Heuristic fuzzy-neuro network and its application to reactive navigation of a mobile robot”, Fuzzy Sets and Systems 110 (2000), pp. 331-340.
97. Sugeno M., and Takagi T., “Fuzzy identification of systems and its application to modelling and control”, IEEE Transactions on Systems, Man, and Cybernetics 15 (1) (1985) 116–132.
98. Tanner H.G. and Kyriakopoulos K.J., “Discontinuous backstepping for stabilization of nonholonomic mobile robots”, proc. of IEEE Conf. on Robotics and Automation, Washington DC, pp. 3948-3953, (2002).
99. Toibero Juan Marcos, Roberti Flavio, Carelli Ricardo and Fiorini Paolo, “Switching control approach for stable navigation of mobile robots in unknown environments”, Robotics and Computer-Integrated Manufacturing 27 (2011) 558–568.
100. Toledo Ana, Rafael Toledo-Moreo, José Manuel Cano-Izquierdo and Miguel Pinzolas-Prado, “Maneuver prediction for road vehicles based on a novel neuro-fuzzy dynamic architecture”, Robotics and Autonomous Systems 58 (2010) 1316–1320.

101. Tsuchiya Kazuo, Urakubo Takateru and Tsujita Katsuyoshi, “A Motion Control of a Two-Wheeled Mobile Robot”, 1999 IEEE.

102. Vandorpe J., and Van Brusell H., “A Reflexive Navigation Algorithm for an Autonomous Mobile Robot”, International Conference on Multisensor Fusion and Integration for Intelligent System, Las Vegas, NV, 1994, pp. 251 - 258.

103. Vandorpe J., Brusell V. H., and Xu H., “LIAS: A Reflexive Navigation Architecture for an Intelligent Robot System”, IEEE Transactions, Industrial Electronics, 1996, Vol. 43(3), pp. 432 - 440.

104. Vidoni Renato, Francisco García-Sánchez, Alessandro Gasparetto and Rodrigo Martínez-Béjar, “An intelligent framework to manage robotic autonomous agents”, Expert Systems with Applications 38 (2011) 7430–7439.

105. Wang M. and Liu J. N. K., "Autonomous robot navigation using fuzzy logic controller," presented at 2004 International Conference on Machine Learning and Cybernetics, Shanghai, China, 2004.

106. Wang Meng and Liu James N.K., “Fuzzy logic-based real-time robot navigation in unknown environment with dead ends”, Robotics and Autonomous Systems 56 (2008) 625–643.

107. Wu Chia-Ju and Tsai Ching-Chih, “Localization of an Autonomous Mobile Robot Based on Ultrasonic Sensory Information”, Journal of Intelligent and Robotic Systems, Vol. 30**,** 267–277, 2001.

108. Yu H. H., and Malik R., “AlMY-An Autonomous Mobile Robot Navigation in Unknown Environment with Infrared Detector System”, Journal of Intelligent and Robotic Systems, 1995, Vol. 14(2), pp. 181-197.

109. Yun Xiaoping and Yamamoto Yoshio, “Internal Dynamics of a Wheeled Mobile Robot”, In Proc. Of IEEE International Conference on Intelligent Robots and Systems at Yokohama, Japan, July, 2003, pp. 26-30.

110. Zadeh L., “Fuzzy sets”, Information and Control, 8(3), 1965, 338-353.

111. Zadeh L.A., “A new direction in AI—toward a computational theory of perceptions”, AI Magazine, 22(1), 2001, 73–84.

112. Zafer Bingul and Oguzhan Karahan, “A Fuzzy Logic Controller tuned with PSO for 2 DOF robot trajectory control”, Expert Systems with Applications 38 (2011) 1017–1031.

113. Zemalache K.M. and Maaref H., “Controlling a drone: Comparison between a based model method and a fuzzy inference system”, Applied Soft Computing 9 (2009) 553–562.

114. Zhang Huai-Xiang, Dai Guo-Jun and Zeng Hong, “A Trajectory Tracking Control Method for Nonholonomic Mobile Robots”, Proceedings of the 2007 International Conference on Wavelet Analysis and Pattern Recognition, Beijing, China, 2-4 Nov. 2007, pp. 7-11.

115. Zhu Anmin and Yang S.X.,” Neurofuzzy-Based Approach to Mobile Robot Navigation in Unknown Environments”, Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions, Volume 37, Issue 4, July 2007 pp:610 – 621.

116. Zohar Ilan, Ailon Amit Ailon and Rabinovici Raul, “Mobile robot characterized by dynamic and kinematic equations and actuator dynamics: Trajectory tracking and related application”, Robotics and Autonomous Systems, 59 (2011) 343–353.

117. Zou Yunfei and Pagilla Prabhakar R., “Distributed Constraint Force Approach for Coordination of Multiple Mobile Robots”,J Intell Robot Syst, 2008.

## Papers Published/ Communicated in International Journals

1. Dayal R.Parhi, Shubhasri Kundu, "Analysis of Fuzzy Inference System for Controlling a Mobile Robotic Agent", International Journal of Applied Artificial Intelligence in Engineering System, 2(1), 2010:65-79.

2. Dayal R. Parhi; Shubhasri Kundu; "A Hybrid Fuzzy Controller for Navigation of Real Mobile Robot",International Journal of Applied Artificial Intelligence in Engineering System 3(1), 2011: 19-33.

3. Dayal R. Parhi; Shubhasri Kundu; "Behavior-based Reactive Navigation Strategy for Mobile Robotic Agent using Hybrid Fuzzy Neuro Architecture", communicated to International Journal of Artificial Intelligence and Computational Research (IJAICR), *ISSN: 0975-3974*.

## Papers Published/ Presented in International or National Conferences

1. Shubhasri Kundu, Dayal R.Parhi, "Behavior based Navigation of Multiple Robotic Agents using Hybrid Fuzzy Controller", published in the proceedings of IEEE sponsored International Conference on Computer and Communication Technology (ICCCT-2010), Sep 17-19, MNNIT, Allahabad, pp.706-711.

2. Shubhasri Kundu; Dayal R. Parhi; "A Fuzzy Approach towards Behavioral Strategy for Navigation of Mobile Agent", published in the proceedings of IEEE & DRDO sponsored International Conference on Emerging Trends in Robotics and Communication Technologies [INTERACT-2010], Dec 3-5, Sathyabama University, Chennai, pp. 328-333.

3. Shubhasri Kundu; Dayal R. Parhi; "Fuzzy based Reactive Navigational Strategy for Mobile Agent", published in the proceedings of IEEE sponsored International Conference on Industrial Electronics, Control & Robotics 2010, NIT Rourkela, Dec 27-29,2010, pp. 37-42.

4. Dayal R. Parhi; Shubhasri Kundu; "Reactive Navigational Strategy for Mobile Agent based on Hybrid Fuzzy Logic", published in National Conference on ETME-2011, Bhubaneswar, pp. 105-118.