Analysis and Development of Computational Intelligence based Navigational Controllers for Multiple Mobile Robots

Prases Kumar Mohanty

Analysis and Development of Computational Intelligence based Navigational Controllers for Multiple Mobile Robots

Thesis Submitted to the

Department of Mechanical Engineering

National Institute of Technology, Rourkela

For award of the degree

of

Doctor of Philosophy

by

Prases Kumar Mohanty

under the guidance of

Prof. Dayal R. Parhi

Department of Mechanical Engineering
National Institute of Technology Rourkela
Orissa (India)-769008

November 2015

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.

(*Prases Kumar Mohanty*)

Date:

Certificate

This is to certify that the thesis entitled, ***"Analysis and Development of Computational Intelligence based Navigational Controllers for Multiple Mobile Robots",*** *being submitted by Mr. Prases Kumar Mohanty to the Department of Mechanical Engineering, National Institute of Technology, Rourkela, for the partial fulfillment of award of the degree Doctor of Philosophy, 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 Doctor of Philosophy 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.

Date: **Supervisor**

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

Acknowledgements

After a so great experience, I obviously have many people to thank. I owe my gratitude to all those people who have made this dissertation possible.

First of all, I am thankful to my principal supervisor Prof. Dayal R. Parhi. I have been amazingly fortunate to have an advisor who gave me the freedom to explore on my own and at the same time the guidance to recover when my steps faltered. His suggestions and new ideas have always inspired and motivated me.

I am thankful to Prof. Sunil Kumar Sarangi, Director of National Institute of Technology, for giving me an opportunity to be a part of this institute of national importance and to work under the supervision of Prof. Dayal R. Parhi. I am thankful to Prof.S.S.Mahapatra, Head of the Department, Department of Mechanical Engineering, for his moral support and valuable suggestions during the research work.

I express my gratitude to Prof. K.P. Maity, Chairman DSC and DSC members for their indebted help and valuable support for accomplishment of dissertation.

I am extremely thankful to Mr. Sandeep Kumar for the extensive help rendered by him during the construction of the robot. Also, I would like to thank Mr.Maheshwar Das for his help in this same regard.

My heart-felt thanks to my labmates, Mr. Shakti P. Jena, Mr. Chinmaya Sahu, Mr. Priyadarshi Biplab Kumar, Mr. Anish Pandey, Mr. Adik R. Yado, Mr. Animesh Chhtoray, Mr. Krishana K. Pandey, Mr. Alok Jha, Mr. Irshad A. Khan, Mrs. Shubhasri Kundu, Mrs. Sasmita Sahu, and Mr. Devendra Ghodki for their support and co-operation which is difficult to express in words.

I express special thanks to Mr. Animesh Chhtoray, Mr. Krishana K. Pandey, Mr. Shakti P. Jena, Mr. Priyadarshi Biplab Kumar and Mr. Devendra Ghodki to have made my staying in the robotics laboratory to late night. The time spent with them will remain in my memory for years to come.

Apart from working advisors and colleagues, I express my heart-felt gratitude to my family members for their constant source of love and support. I would like to mention a special thanks to my wife, Priyadarsini for her continuous encouragement during my days

of Ph.D. I thank my little angel Swayam and nephew Sailesh, for their understanding, patient and moral support during my research.

Last, but not the least, I praise God, the Almighty for giving me the strength during the course of this research work.

Prases Kumar Mohanty

Synopsis

Navigational path planning problems of the mobile robots have received considerable attention over the past few decades. The navigation problem of mobile robots are consisting of following three aspects i.e. locomotion, path planning and map building. Based on these three aspects path planning algorithm for a mobile robot is formulated, which is capable of finding an optimal collision free path from the start point to the target point in a given environment.

The main objective of the dissertation is to investigate the advanced methodologies for both single and multiple mobile robots navigation in highly cluttered environments using computational intelligence approach.

Firstly, three different standalone computational intelligence approaches based on the Adaptive Neuro-Fuzzy Inference System (ANFIS), Cuckoo Search (CS) algorithm and Invasive Weed Optimization (IWO) are presented to address the problem of path planning in unknown environments. Next two different hybrid approaches are developed using CS-ANFIS and IWO-ANFIS to solve the mobile robot navigation problems. The performance of each intelligent navigational controller is demonstrated through simulation results using MATLAB. Experimental results are conducted in the laboratory, using real mobile robots to validate the versatility and effectiveness of the proposed navigation techniques. Comparison studies show, that there are good agreement between them. During the analysis of results, it is noticed that CS-ANFIS and IWO-ANFIS hybrid navigational controllers perform better compared to other discussed navigational controllers. The results obtained from the proposed navigation techniques are validated by comparison with the results from other intelligent techniques such as Fuzzy logic, Neural Network, Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO) and other hybrid algorithms. By investigating the results, finally it is concluded that the proposed navigational methodologies are efficient and robust in the sense, that they can be effectively implemented to solve the path optimization problems of mobile robot in any complex environment.

Keywords: Mobile robot, Navigation, ANFIS, Cuckoo Search, Invasive weed Optimization.

Table of Contents

List of Figures

List of Tables

Abbreviations

ANFIS	-	Adaptive Network Based Fuzzy Inference system
BF	-	Barrier Following
CI	-	Computational Intelligence
CS	-	Cuckoo Search
FOD	-	Front Obstacle Distance
HA	-	Heading Angle
IWO	-	Invasive Weed Optimization
LOD	-	Left Obstacle Distance
LWV	-	Left Wheel Velocity
LSE	-	Least Square Estimation
MANFIS	-	Multiple Adaptive Network Based Fuzzy Inference system
Med	-	Medium
OA	-	Obstacle Avoidance
OB	-	Obstacle
ROD	-	Right Obstacle Distance
RWV	-	Right Wheel Velocity
RAM	-	Random Access Memory
RMSE	-	Root Mean Square Error
SA	-	Steering Angle
TS	-	Target Seeking
WMR	-	Wheeled Mobile Robot

List of Symbols	
x_{OB}	x-coordinates of the obstacles
y_{OB}	y-coordinates of the obstacles
x_{ROB}	x-coordinate of the Robot
y_{ROB}	y-coordinates of the Robot
x_G	x-coordinate of the goal
y_G	y-coordinate of the goal
C_1 and C_2	Controlling parameters in Cuckoo Search algorithm
w	Wheel base or distance between the two wheels
n	Non-linear modulation index
ξ	Posture of the robot
δ_m	Degree of mobility
δ_s	Degree of steerability
δ_M	Degree of maneuverability
θ	Heading angle
θ'	Steering angle
f_i	Objective function
$\sigma_{initial}$	Initial standard deviation
σ_{final}	Final standard deviation
S_{max}	Maximum number of seeds
S_{min}	Minimum number of seeds
α and β	Controlling or weight parameters in IWO algorithm

CHAPTER-1

1. INTRODUCTION

This chapter presents the motivation and overview of the research work carried out in the dissertation. The back ground and motivation of the research area have been depicted in the first part. Second part of the chapter contains the aim and objectives of the research work. The novelty of the research work has been presented in the third section. Finally, the last part of the chapter gives an outline of each chapter of the dissertation for the current research.

1.1 Background and Motivation

Nowadays robotic systems have the capability to behave autonomously and to improve the system performance by interacting with environments. Mobile robots are the kind of robots that are able to rove, sense, and respond in a given environment and are able to perform assignments and explore without human intervention. Path planning and control of a mobile robot in unrecognized environments are one of the most challenging task in the robotics field. The mobile robots have several applications in industrial environments, medical services, military reconnaissance, space exploration, cleaning, and agricultural sectors. As the robotic technology advances very fast, it has been envisioned that in future, the mobile robotic system will infiltrate into each aspect of the human lives and have an increasing number of real time implementations. So, the mobile robots must be able to perform various tasks autonomously without collision with obstacles and other robots.

The autonomous navigation problem of mobile robot is consisting of the following additives:

- Localization
- Path planning
- Map building

Firstly, as the environment is partly known or totally unknown to the mobile robot and an adequate representation of the environment must be framed (Map building). Secondly, the

representation of the environment obtained from imperfect sensors with vulnerabilities must adapt to inaccurate localization. At last, the autonomous mobile robot must be able to simultaneously plan its motion based on the partly known environment information while extracting environmental information online. Based on these above three fundamental aspects, the path planning algorithm for a mobile robot is formulated, which is capable of finding an optimal collision free path from the source point to the target point in a given environment. The environment can be classified as known or structured environment, the semi-structured environment, and unknown environment. The different aspects of the mobile robot motion control scheme are shown in Figure1.1.

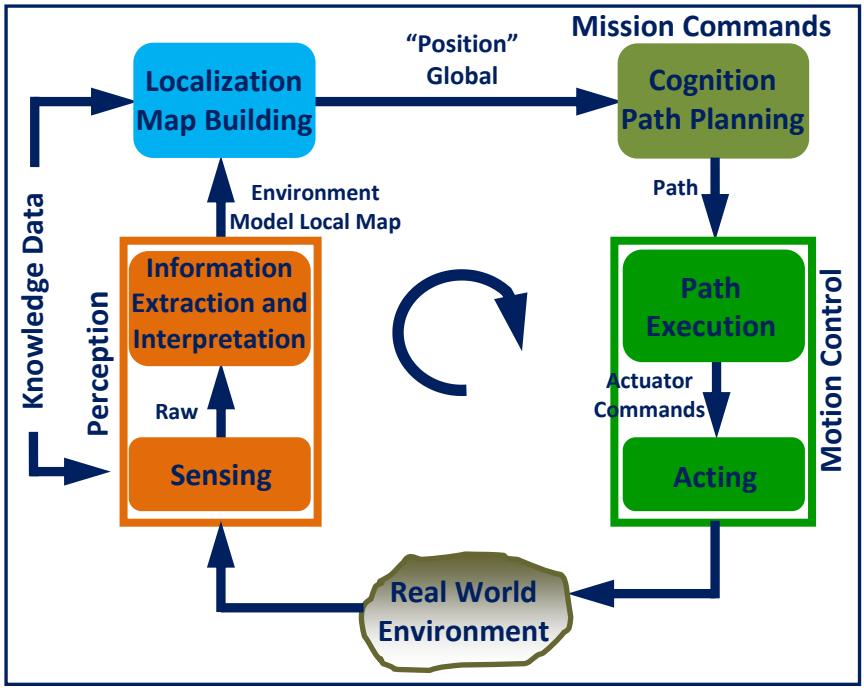


Figure 1.1 A block scheme of the navigation system.

The path planning or navigation problem of the mobile robot has been extensively studied by many researchers over the past two decades. The main difficulty of the path planning problem depends on whether the environment and obstacle positions are known in advance or not. The path planner has to search optimal route if the environment and the position of the obstacles are known in advance. These types of path planning methods are known as global or off line path planning. These methods are computationally more expensive and rarely implemented for the real time condition. If the environment is partially known or completely unknown, the estimation has to be determined on-line

under the real time condition. These types of path planner are known as local or reactive path planner.

Recently, there have been many interesting research works proposed by the researchers in the area of the robot path planning. But the presence of intricacy and uncertainty in the robot path planning problems, traditional path planning algorithms such as Voronoi diagram, Visibility graph, Grid, Cell decomposition algorithm, Road map approaches, and Artificial Potential Field (APF) method are not feasible for online applications. But the above indicated classical methods suffer from many draw backs, such as high cost of computation and may trap in the local minima situation, which make them inefficient in the method. The potential field method can be implemented effectively and can provide acceptable results for robot path planning. But the main problem is that the robot is trapped due to local minima, or when there is no passage between closely spaced obstacles.

Therefore many computational intelligence techniques such as Fuzzy logic, Artificial neural network, Evolutionary algorithms, and Swarm intelligence based techniques and hybrid approaches have been widely adopted in mobile robot navigation problems.

Despite the relative success of the above discussed methodologies, there is still stringent requirement of developing more advanced and effective navigation strategies in order to achieve the robust navigation in unknown environment with uncertainties by solving the three additives (Localization, Path planning, Map building) simultaneously.

1.2 Aim and Objectives of the Research

This research work aims at the investigating of more advanced and effective path planning algorithms for single and multiple mobile robots using computational intelligence techniques. In this dissertation, Adaptive Neuro-Fuzzy Inference System (ANFIS), Cuckoo Search (CS) algorithm, Invasive Weed Optimization (IWO) algorithm, and hybridization of CS-ANFIS and IWO-ANFIS algorithms have been implemented to solve the navigational problem of mobile robots.

The main objectives of the research work presented in the dissertation are as follows:

- To carry out kinematic analysis of a differential wheeled mobile robot.

- To develop an adaptive neuro-fuzzy inference system (ANFIS) based navigational controller for mobile robots.
- To build a navigational path planner based on the Cuckoo Search (CS) algorithm.
- To formulate a navigational controller based on the Invasive Weed Optimization (IWO) algorithm.
- To develop a hybrid navigational controller based on the CS-ANFIS approach for multiple mobile robots navigation.
- To develop a hybrid motion planner based on the IWO-ANFIS approach for multiple mobile robots navigation.
- Simulation and experimental verifications of the above discussed methods are to be carried out.

1.3 Novelty of the Research Work

In the recent decades, various computational intelligence methods have been deployed by many researchers to solve the path optimization problems for mobile robots.

In the current research work, a systematic effort is given to design and development of intelligent navigational controllers using Adaptive Neuro-Fuzzy Inference System (ANFIS), Cuckoo Search (CS), Invasive Weed Optimization (IWO), and hybridization of CS-ANFIS and IWO-ANFIS for solving the path planning problems of single and multiple mobile robots. But to the author's knowledge CS, IWO, CS-ANFIS and IWO-ANFIS algorithms are not yet implemented in the field of mobile robots navigation.

Further the developed intelligent path planners have the capability to solve the navigational tasks for single and multiple mobile robots.

1.4 Outline of the Thesis

Following this current chapter, the remainder of the dissertation is organized as follows:

- **Chapter-2** presents the literature survey on kinematic models of differential wheeled robots and navigational methodologies used for mobile robots.
- **Chapter-3** discusses the modeling and analysis of wheeled robot kinematics.
- The first part of the **Chapter-4** presents the implementation of Adaptive Neuro-Fuzzy Inference System (ANFIS) and the second part deals with Multiple

Adaptive Neuro-Fuzzy Inference System (MANFIS) for navigation of mobile robots in an unknown environment.

- **Chapter-5** describes the Cuckoo Search (CS) algorithm for single mobile robot navigation in unknown environments.
- **Chapter-6** presents the Invasive Weed Optimization (IWO) algorithm for single mobile robot navigation in unknown environments.
- **Chapter-7** discusses the CS-ANFIS hybrid algorithm for multiple mobile robots navigation.
- **Chapter-8** presents the IWO-ANFIS hybrid methodology for multiple mobile robots navigation in unknown environments.
- **Chapter-9** provides a comprehensive review of results obtained from all the discussed techniques adopted in the current research work.
- **Chapter-10** addresses the conclusion drawn from the research work carried out in the dissertation and gives the possible extension directions in the same domain.

CHAPTER-2

2. LITERATURE REVIEW

The current chapter highlights the work related to development of path planning and control strategies for mobile robot navigation in various environments. The main objective is to survey the developments made by researchers during the past few decades on the mobile robot navigation. This chapter provides ample confidence to find an appropriate literature gap or methodological weaknesses present in the existing study area to solve the research problem.

2.1 Introduction

The path planning control system for an autonomous mobile robot must perform many information processing tasks in real time condition. A considerable amount of research has been carried out on the mobile robot path planning (RPP). In the general concept, the path planning means to produce collision free paths from the start point to the target point in a given work space. Several conventional [1] and computational intelligence techniques (CI) or reactive approaches [2-4] have been studied to represent environments and plan the routes for the mobile robots. Due to the NP-hardness (Non-deterministic polynomial time) of the path planning problem, computational intelligence techniques have outperformed the conventional approaches and have received wide popularity. The collision free paths are constructed by the path planning algorithms, and robot moves along the constructed paths to reach the target. The path planning system for the mobile robots is decomposed into a series of functional units, as shown in Figure 2.1 by continuous vertical slices. After deciding the computational requirements for a robot, the path planning system is decomposed into a series of horizontal functional units to achieve the desire task behaviour required for the robot. This is illustrated in Figure 2.2. After, surveying many research articles in the robot path planning field, a number of existing research works for each technique is identified and categorized.

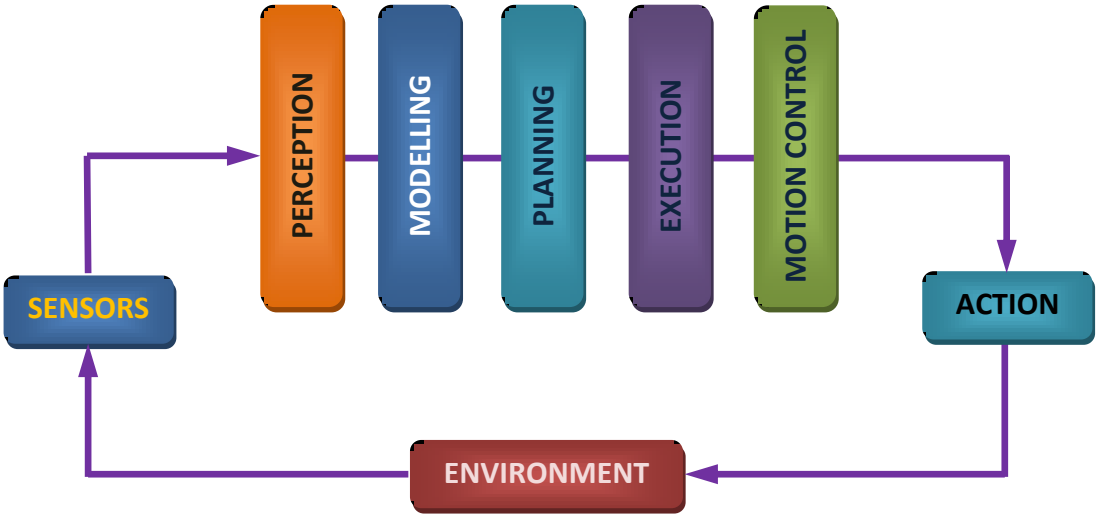


Figure 2.1 Path planning architecture of the mobile robot [2].

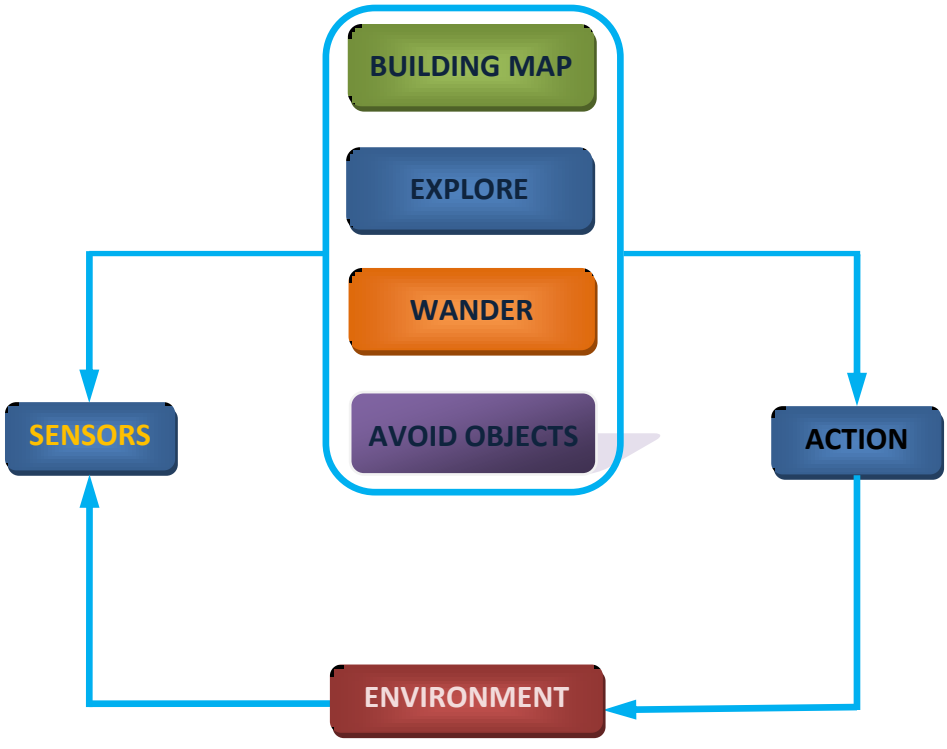


Figure 2.2 Flow diagram of task achieving behaviour of a mobile robot [2].

2.2 Kinematic Analysis of Wheeled Mobile Robots

2.2.1 Wheel Locomotion

A mobile robot can have several locomotion mechanisms [5-8] such as normal walking, sliding, running, jumping etc. These mechanisms are mostly biologically inspired from human beings, snakes, four legged animals, animals like kangaroo respectively. The reason for choosing biological activities as locomotion mechanisms is that those activities are very much successful in several environments. But artificial making of biological mechanisms suffers from various difficulties such as mechanical complexity, use of biological energy storage systems used by animals and difficulty in mathematical analysis. Due to these difficulties, the simplest biological system with less number of articulated legs are widely used. In legged locomotion, there is involvement of more mechanical complexity, as there are more degrees of freedom. To avoid this difficulty, active powered wheels are preferred. Use of active wheels makes the analysis simple, and it is suitable for flat ground types. Literature reveals about many well-known algorithms those are proposed [9-10] in the past for controlling the motion of the mobile robots. Considering the mobile robot as a planar body, the relation between the chassis of the robot and the wheels that are attached to it can be found out in [11]. The wheels of a mobile robot are categorized in five sections considering the geometrical constraints in moving condition of the wheel. After classification of the wheels, kinematic and dynamic models can be proposed [12].

There is absence of vertical axis of rotation for steering in the fixed standard wheels. The angle to chassis in this type of wheel is fixed, and there is flexibility for back and forth movement along the plane of the wheel and it can rotate around the contact point with the ground. Maneuverability, stability, and controllability are the three fundamental characteristics those are required for a mobile robot. To attain static stability, at least three wheels need to be attached to the mobile robot. Attachment of three independent standard steered wheels makes the mobile robot omnidirectional [13]. Standard steered wheels are preferred over other wheels because of their simplicity in design and reliability. Cariou et al. [14] have discussed the slipping phenomena in both angular and lateral terms and designed an automated path for navigation of a four-wheeled mobile robot. Marcovitz and Kelly [15] have derived a method to represent slip in all degrees of freedom and implemented them on wheels with skid steering mechanism.

The mobility limitations of the standard steered wheels have led to the design of caster wheels making them suitable for use in hospitals, offices, homes, etc. In [16], the kinematics has been discussed for a standard caster wheel having double-wheel-type active caster. Chung et al. [17] have focused on modeling of holonomic and omnidirectional robots having wheels that are offset steerable. A dynamic model has been proposed introducing contact stability condition for power caster wheels in [18]. Kim and Lee [19] have proposed a condition of isotropy based on the kinematic model of a mobile robot equipped with three active caster wheels those are able to move in all directions.

In a swedish wheel, the rollers are mounted on the perimeter. Due to the presence of the rollers, the robot can move in any direction and turn anywhere, i.e. the presence of rollers make the robot omnidirectional. Indiveri [20] has analyzed the kinematic model of a mobile robot using n number of Swedish wheels. A path control law has been discussed in [21] for a Swedish wheel equipped mobile robot. Doroftei et al. [22] have designed a robot with Macanum wheel. In their robot, they used four Swedish wheels to avoid any use of steering system for the robot.

The use of spherical wheels offers greater stability and mobility to the mobile robots. The controlling mechanism and planning of motion for a spherical wheel is much more complex.

In [23], a new spherical wheel has been introduced to make the robot omnidirectional. This wheel is able to climb the stairs easily. A combination of spherical and Omni wheels are discussed in [24]. The operation of this type of wheel is based on a simple spherical wheel driven by two omni-wheels at the perpendicular position. Two different algorithms have been proposed in [25] for “ball-plate problem” that deals with motion planning for a spherical mobile robot. One algorithm is based on Gauss-Bonet theorem achieving configuration through maneuvers of the spherical triangle, and another algorithm is based on standard kinematic model achieving reconfiguration through circular arcs and straight line segments. Lauwers et al. [26] have proposed a standard overall design of a robot equipped with spherical wheels that can move in any direction.

The use of a back stepping-like feedback mechanism for the tracking control of a differential drive robot has been proposed in [27]. In [28], a nonlinear feedback path controller has been proposed. Implementation of fuzzy logic for controlling the motion of mobile robots has been discussed in [29]. Menn et al. [30] have proposed a kinematic model using a genetic algorithm for motion control strategy. The use of a model-reference

adaptive motion controller is discussed in [31] and is implemented in a differential drive mobile robot.

2.2.2 Motion Control Analysis for Differential Mobile Robot

Over the past few years, the motion control of nonholonomic mobile robots has drawn attention of the researchers. Several algorithms have been proposed to solve the motion control problem.

In [32], a novel adaptive trajectory controller has been discussed for the nonholonomic mobile robot. Slusny et al. [33] have proposed an evolutionary algorithm with two motion control algorithms based on localization of mobile robot. Ali [34] has proposed a method for the development and implementation of a robotic platform. The testing of the semi-autonomous platform helps for educational and research purposes. To interface various sensors and motor drivers to the ATMEL (AVR ATmega 32) microcontroller chip, a modular hardware design has been proposed. A known path following control law has been discussed in [35].

2.3 Navigation Techniques used for Mobile Robots

Since last few decades, the researchers have emphasized on various navigational techniques for control of mobile robots. The different navigational methods used for the mobile robots are summarized below.

2.3.1 Classical Approaches

Many classical methods are used by researchers to solve the path planning problem of a mobile robot. The reviews based on the classical methods are depicted below.

2.3.1.1 Roadmap Approaches

Path planning of mobile robots using roadmap approaches, such as the visibility graph, voronoi diagram, silhouette and the sub goal network, may have some disadvantages for a long path, sharp turns or collisions with obstacles. In these techniques, the robots may have unnecessary turn around crossing points which leads to longer paths and prevent smooth motion. The roadmap method is a class of topological maps representing the distinctly feasible space in environments. Review on various well known roadmap approaches are discussed below.

The visibility graph builds the paths through connecting every pair of vertices of the obstacles by a straight line without navigating in the interior of the obstacles [36]. These set of routes are called as the roadmap. If a continuous route is found in the free space, the starting and the goal point is then joined to reach the final solution. There are more than one continuous paths found, Dijkstra's shortest path method [37] is often applied to search the best path. The generalized voronoi diagram is well known mapping approach that provides maximum clearance between the obstacles [38-40]. The diagram is framed by the paths maintaining an equal distance between the two obstacles. This method provides a safe path, which avoid the obstacles as much as possible. Some researchers have attempted to improve the efficiency of the voronoi diagram and proposed some path planning methods that make-up the draw backs of the voronoi diagram such as sharp turns and long loop [41]. A hybrid algorithm consisting of the voronoi diagram, visibility graph, and potential field method has been suggested by Masehian and Amin-Naseri [42]. They observed that the hybrid algorithm is not producing smooth path, and also the architecture of the algorithm is too complex. Yang and Hong [43] have proposed a new roadmap method for robot path planning using skeleton maps. They used crossing polygonal around crossing points to improve the efficiency of the skeleton maps. Wein et al. [44] have introduced a new type of hybrid diagram called the VV diagram (the visibility-voronoi diagram), which provides the shortest routes with preferred clearance value for a mobile robot in a planer environment. Figures 2.3 and 2.4 have shown the path generated using visibility graph and voronoi diagram respectively. Another popular methodology using Probabilistic Roadmaps (PRMs) for solving motion planning problems has been addressed by Kavraki et al. [45]. The paths obtained from the proposed methodology are piecewise linear but far from the shortest possible path. Sanchez and Latombe [46] have introduced an improvement over PRM by providing a lazy-in-collision-detection technique. Canny [47] has developed a roadmap method called silhouette method to solve the basic motion planning problem of the robot. Generally it consists of generating the silhouette in the robot’s work space, and constructing the roadmap by linking these silhouette curves together.

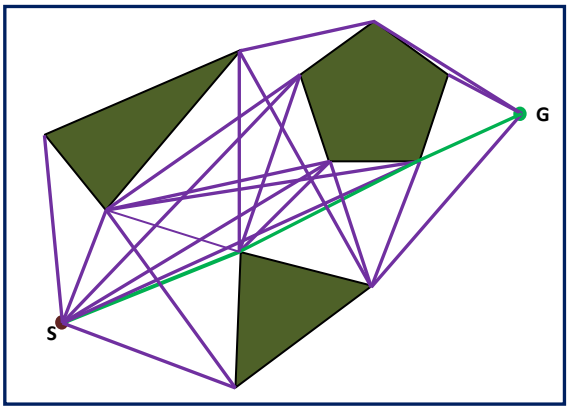


Figure 2.3 Visibility Graph [36].

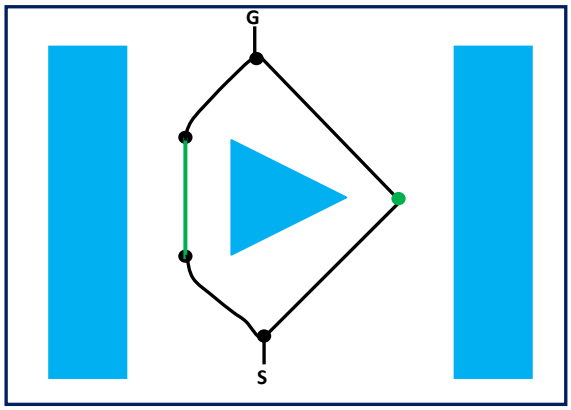


Figure 2.4 Voronoi Diagram [36].

The silhouette and the connecting curves form the roadmap from the start configuration to the target configuration. Bhattacharyya et al. [48] have tried to make-up the drawbacks of the silhouette method for robot path planning. They used Dijkstra shortest algorithm for finding the route between the two given points out of the connecting graphs. In the subgoal-network path planning method, subgoals act as key configurations expected to be useful for searching collision free routes. A network of subgoals are generated first and maintained by a global planer, and a simple local operator is adopted to reach among subgoals. These two stage path planning methods have been first introduced by Faverjon and Tournassound [49].

2.3.1.2 Cell Decomposition Methods

The idea of dividing large learning space into smaller ones resulting in the simplicity of exploration has been discussed by many researchers [50]. Motion planning methods for the robot based on the cell decomposition algorithm [51] have been extensively studied so far. In this algorithm, the robot's free space is decomposed into a set of simple cells, and the adjacency relations between the cells are generated. A collision free route has been constructed between the start configuration to the goal configuration of robot by searching the two free cells containing the start and the goal and then joining them with a sequence of connected free cells.

Cell decomposition algorithm can be categorized into two classes:

1. Exact cell decomposition
2. Approximate cell decomposition (Quadtree decomposition)

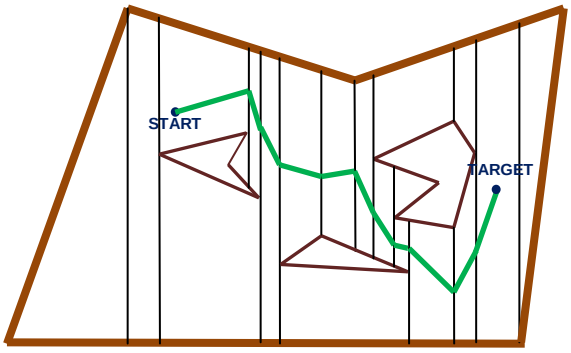


Figure 2.5 Exact Cell Decomposition [52].

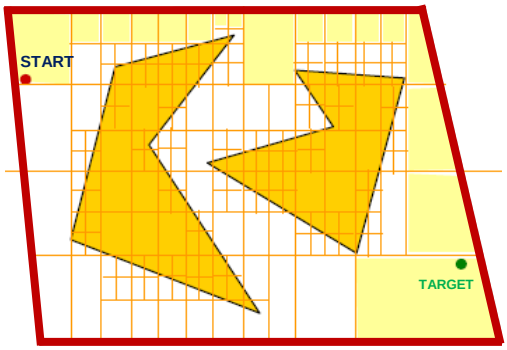


Figure 2.6 Approximate Cell Decomposition [54].

In the exact cell decomposition method [52-53], the robot's free space decomposes into trapezoidal and triangular cells shown in Figure 2.5. In the second step, a connectivity graph is generated between the adjacency relation cells and searches the graph (sequence of consecutive cells) for a path. Finally, the sequence of connectivity cells is converted into a free path for the robot. In the approximate cell decomposition method [54-55], robot's free space decomposes into smaller rectangles shown in Figure 2.6. This type of decomposition is known as "Quadtree". At the resolution, the cells whose interiors lie completely in the free space are utilized to generate the connectivity graph. In the end, the search algorithm finds a collision free path for the robot.

2.3.1.3 Artificial Potential Field Approach

Artificial Potential Field (APF) approach to the field of mobile robot navigation is first introduced by Khatib [56] around 1986. In this strategy, obstacles and targets are considered as charged surfaces and the net potential create a force on the robot. These forces pull the robot towards the goal while pushing away from the obstacles as shown in Figure 2.7. So, the robot follows the negative gradient to avoid the collision caused by the obstacles and reach the target points. The main drawbacks of the method are;

a) Local minima may occur in locations rather than the goal point and moving the robot in the wrong direction.
b) There may be situations, where it is difficult to determine the potential field due to the shape of obstacles.

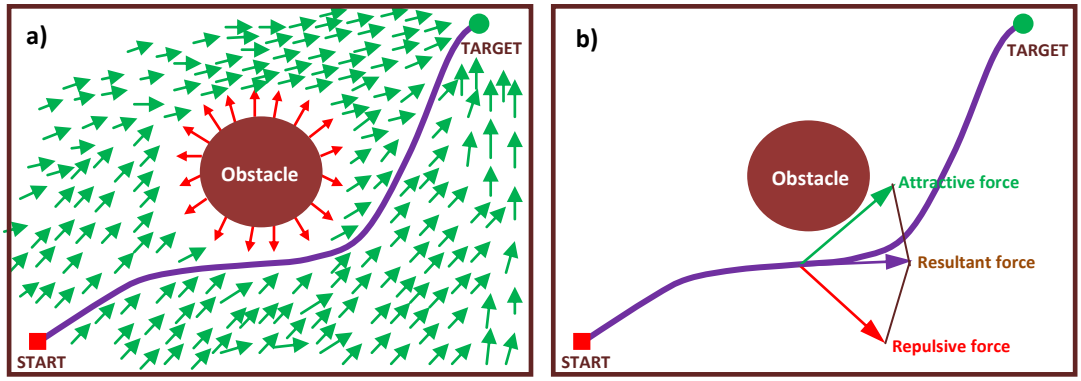


Figure 2.7(a-b) Example of Potential Field Control Approach for Mobile Robot Path Planning.

Many researchers have applied potential field method for mobile robot path planning and tried to make-up the disadvantages raised by the method. The following highlighted literature on the potential field methods are drafted below.

Local navigation of a mobile robot in a telerobotics context using potential field approach has been studied by Garibotto and Masciangelo [57]. Kim and Khosala [58] have developed a new potential field strategy to the obstacle avoidance problem for a mobile robot in a known environment. They used harmonic functions to eliminate the local minimum problem in a cluttered environment. Mobile robot path planning in dynamic environments using potential field method has been discussed by Ge and Cui [59]. They framed a new potential function with respect to both relative position and velocity between the robot, obstacles and the target. Collision free path planning for mobile robots using an electrostatic potential field (EPF) method has been studied by Valavanis et al. [60]. The laws of electrostatic are used to derive the potential function and generate an approximately optimal route in a real time condition. Velocity controlling of a mobile robot based on the potential field method has been discussed by Huang [61]. The proposed method has utilized both position and velocity information of the robot, the obstacles, and the goal in order to track the moving goal, while avoiding the obstacles along the route. Path planning method for a mobile robot using improved potential field function has been presented by Shi and Zhao [62]. The proposed navigation strategy includes both improved attractive potential function and improved repulsive potential function and can guarantee the goal point is the global minimum point. Navigation of a mobile robot in the unknown environments using improved potential field approach has been studied by Sfeir et al. [63]. In this proposed method, a new form of potential

function has been introduced to reduce oscillations and avoid conflicts, when the goal is nearer to obstacles. A rotational force is also incorporated into the system for producing a smoother trajectory around the obstacles. Controlling the motion of multiple mobile robots using potential field method has been solved by Hui [64]. The developed control system for multiple mobile robots navigation based on the same principle is discussed in [63]. Pimenta et al. [65] have discussed the problem of mobile robot navigation using potential field method. They transformed the navigation problem into electrostatic problem and solved by the finite element methods. Pradhan et al. [66] have presented a potential field method to navigate the multiple mobile robots in unknown environments. The simulation results are presented using ROBPATH tool to show the performance of the proposed approach. Borenstein et al. [67] have presented a real time obstacle avoidance method for a mobile robot. The navigation method considers the dynamic behaviour of a mobile robot and solves the local minima situation. To solve the oscillation problems near the obstacles, Biswas and Kar [68] have developed a powerful navigation system for a mobile robot using potential field method. They compared the results between the traditional method and Levenberg-Marquardt algorithm. The results demonstrated that the Levenberg-Marquardt method improves the oscillations problem and generates the smooth trajectory for the robot.

2.3.2 Computational Intelligence (CI) Approaches

Knowledge based or computational intelligence techniques are widely used as alternatives to classical approaches to model the navigation system for mobile robots. In this section, some of them and their application with examples are discussed below.

2.3.2.1 Fuzzy Inference Methods

The sensor-based navigation technique for a mobile robot using fuzzy controller has been discussed by Ishikawa [69]. In this technique, he used two functions, one function is used for tracing a planned path and the other is used to avoid stationary and moving obstacles. The effectiveness of the developed method has been discussed with simulation results. Li and Feng [70] have presented a fuzzy logic technique for robot path planning in uncertain environments. They designed a fuzzy controller whose inputs comprise of a heading angle and the separation distances between the obstacles. The outputs are the real wheel velocities of the robot. Fuzzy logic techniques for mobile robot obstacle avoidance have been discussed by Reignier [71]. In this article, he has fuzzy logic controller to solve

various reactive behaviours for the mobile robot. Wu [72] has discussed the navigation of a mobile robot in the 2-dimensional unknown environment using a sensor based fuzzy algorithm. He used an optimal learning algorithm in the fuzzy system to minimize total covering distance of the robot. Beom and Cho [73] have introduced a sensor-based path planning strategy for a mobile robot in unknown environments using fuzzy logic and reinforcement learning. The viability of the developed algorithm has been confirmed by the simulation results. Intelligent collision avoidance by the automated guided vehicle (AGV) using a fuzzy logic approach has been described by Lin and Wang [74]. In this paper, they have mainly emphasized on the sensor modeling and trap recovering situations. Maaref and Barret [75] have presented a new sensor-based fuzzy navigation method of a mobile robot in an indoor environment. The proposed navigation strategy consolidates two sorts of obstacle avoidance behaivour, one for the concave types and other is for the convex. A new navigation strategy for mobile robots in challenging environments, using a fuzzy logic method has been discussed by Seraji and Howard [76]. They introduced a new traverse-terrain behavior that uses the regional traversability index to control the robot to the safest and the most traversable terrain. Navigation for a non-holonomic mobile robot using fuzzy logic approach has been discussed by Abdessemed et al. [77]. They used an evolutionary algorithm to extract the optimized IF-THEN rules and a new fuzzy image concept is presented to avoid any collision with surrounding environment. Navigation of multiple mobile robots using fuzzy logic controller has been described by Parhi [78]. In this model (Figure 2.8), the fuzzy rules are embedded in the robot controller in order to avoid obstacles in cluttered environments. A set of collision prevention rules are designed using Petri-Net model to avoid collision among one another.

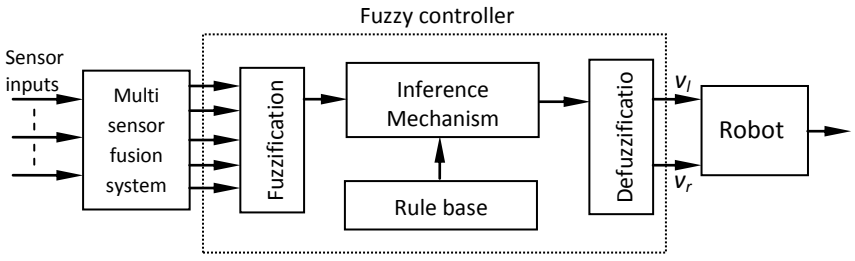


Figure 2.8 Fuzzy logic approach for Mobile robot navigation proposed by Parhi [78].

A behaviour based robotic control using fuzzy discrete event system (FDES) has been employed by Huq et al. [79]. The developed method exploits the features of fuzzy logic and discrete event system to frame the activity behaviour using fuzzy vector. Wang and Liu [80] have discussed fuzzy logic based real time robot navigation in unknown environments with dead end conditions. Navigation control of robotic vehicle based on the fuzzy behaviour system using multi valued logic framework has been addressed by Selekwa et al. [81]. The mobile robot navigation in local environment using a new fuzzy logic algorithm has been presented by Motlagh et al. [82]. The main task of the developed fuzzy controller is to perform obstacle avoidance and target seeking behaviour. Navigation of multiple mobile robots using fuzzy logic approach has been discussed by Pradhan et al. [83]. Fuzzy logic based navigational controller for Khepera-II mobile robot has been addressed by Obe and Dumitrache [84]. Samsudin et al. [85] have outlined an ordinal fuzzy logic controller for obstacle avoidance of Khepera-II portable mobile robot. It is easier to construct high elucidate rules over the conventional controller and genetic algorithm is incorporated to optimize the structure of the ordinal fuzzy controller. The reactive navigation of mobile robots using a new fuzzy control system has been introduced by Motlagh et al. [86]. They used a causal inference mechanism of the fuzzy cognitive map (FCM) to coordinate various motion concepts using input and output factors, and a genetic algorithm is introduced to tune the fuzzy inference system. Mo et al. [87] have presented a new behaviour based fuzzy control method for mobile robot path planning. They developed a fuzzy controller in which angular velocity of driving wheels is used as output of different behaviours. Abdessemed et al. [88] have presented a new navigation model for a mobile robot using fuzzy logic and stereo vision strategy. They found that some of the fuzzy rules are not triggered in the critical situation for which the stereo vision system can be used to navigate the robot successfully. Abdelkrim et al. [89] have presented a fuzzy algorithm using Kinect sensor for mobile robot path planning. The Kinect sensor is used to detect and localize the static and moving the object, and fuzzy controllers are incorporated for target seeking and obstacle avoidance.

2.3.2.2 Neural Network Methods

Koh et al. [91] have presented a neural network based navigation system for a mobile robot in indoor environments. They mainly dealt with the three issues, which are the determination of the current position and heading angle, path control in order to follow the desired path and local path planning for uncertain environments. The fuzzy logic and

reinforcement learning are used for obstacle avoidance and vision system, and the neural network is used for path tracking control. Explanation based neural network learning algorithm for solving the navigational problem of a mobile robot in an indoor environment has been addressed by Thrun [92]. Mobile robot navigation using feed forward neural network has been addressed by Pal and Kar [93]. They have trained the neural network with sonar range inputs to obtain steering angle for the robot. Obstacle avoidance and path planning of mobile robot using an analog neural network dynamics have been proposed by Glasius et al. [94]. The neural network receives inputs, then the neuron in the controller starts to change their activity towards a specific value. The path is framed using neural activity gradient by adjusting the direction of the motor response. An efficient neural network navigational controller for real time motion planning of a mobile robot has been proposed by Yang and Meng [95]. The proposed controller is planned through the biologically inspired neural network without any prior information about the environment and without any learning process. Path planning and obstacle avoidance for a mobile robot using artificial neural network has been studied by Zarate et al. [96]. They proposed a path planner, which is based on the neural memorization of a path previously planned and it allows to move the robot from its start position to target by avoiding obstacles. Yao et al. [97] have presented the RAM based neural controller in mobile robotics to detect and avoid obstacles in real time. Janglova [98] has presented a new path planning strategy for an autonomous mobile robot in partially unstructured environments using neural networks. She used two neural networks; one is used to construct the free space using ultrasonic sensors, and other is used to find the safe direction of the robot while avoiding the nearest obstacle. Navigation strategy for a mobile robot using a probabilistic neural network (PNN) has been addressed by Castro et al. [99]. They implemented the proposed methodology in a real prototype robot, and the result obtained validates its feasibility. Wahab [100] has developed a new navigational system for a mobile robot based on the artificial neural network. He used two neural networks, one neural network for creating free space to avoid obstacles and the other is for navigating the robot towards the target. Behaviour control of a mobile robot using artificial neural network has been proposed by Leon et al. [101]. They designed several behavior modules for a mobile robot based on the neural network paradigms. The developed neuro-controller has been validated through simulation results and tested on a Khepera robot. Mobile robot navigation in dynamic environments using heuristic rule based neural

networks has been addressed by Parhi and Singh [102]. They hybridized the heuristic rules with a neural network to create the required mapping between perception and motion. Singh and Parhi [103] have developed a path optimization algorithm for a mobile robot based on the neural network. They used four layer neural networks with back propagation algorithm to solve the path optimization problem of mobile robots. Motion and path planning of a mobile robot using neural network approach has been proposed by Engedy and Horvath [104]. They used back propagation through time (BPTT) training approach to train the neural network. Mobile robot path planning using neural network has been discussed by Mahmud et al. [105]. They applied the Kohonen type concurrent self-organizing map (CSOM) to determine the correct steering direction of the robot. Mobile robot navigation using the recurrent neural network (RNN) has been addressed by Brahmi et al. [106]. They used two RNN connected in series to control the motion of the mobile robot. Local path planning of a mobile robot using neural network strategy has been discussed by Motlagh et al. [107] shown in Figure 2.9. In this study, they utilized the neural network and reinforcement learning to enable the robot to learn environments on its own.

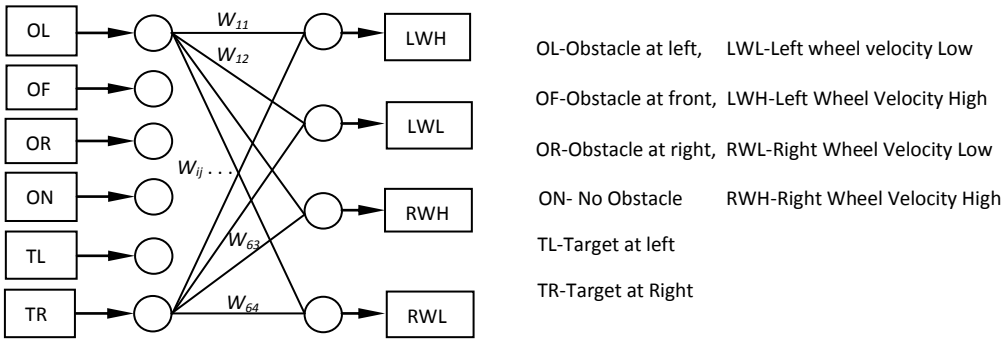


Figure 2.9 Neural Network approach for Mobile robot navigation proposed by Motlagh et al. [107].

Cao et al. [108] have presented a spiking neural network based path planner for the autonomous mobile robot. They used CCD cameras, encoders and ultrasonic sensors into spike train, which are embedded in the three layered spiking neural network to give motion to each motor.

2.3.2.3 Fuzzy-Neural Network Methods

Sensor based path planning for a mobile robot using fuzzy logic and neural network techniques have been addressed by Beom and Cho [109]. In this hybrid method, they used the neural network to preprocess the sensors' information and situation of the robot at the present instant and fuzzy rules are used to make the decision on the action of the mobile robot. Song and Sheen [110] have developed a new local path planning method for a mobile robot using Fuzzy Kohonen Clustering Network (FKCN). In this study, they combined the heuristic rules of fuzzy logic with neural network for planning the required mapping between perception and motion. Kubota et al. [111] have presented a multi-objective behaviour coordinate for a mobile robot using the fuzzy neural network. In this article, they updated the weights of fuzzy rules by a neural network according to the perceptual information. The simulation results demonstrate that the robot can control the multi-objective behaviour using the proposed hybrid technique. Ma et al. [112] have discussed a hybrid intelligent motion planning controller for a mobile robot including fuzzy logic and neural network. Er et al. [113] have developed a navigation system for real time control of a mobile robot based on the generalized dynamic fuzzy neural networks (GDFNN). Using this developed technique, not only fuzzy parameters can be optimized, but also the navigation system can be self-adaptive. Implementation of the fuzzy-neural network to improve the efficiency of the artificial potential field (APF) method has been discussed by Su et al. [114]. In this study, they used the fuzzy neural network to control the virtual forces in APF and adjust the weights associated with the each virtual force. A fuzzy neural based path planner for a mobile robot to avoid obstacles has been presented by He et al. [115]. Using this proposed hybrid system; the mobile robot can detect obstacles, take its action and then reach the target. A multisensor integration based fuzzy-neural method for obstacle avoidance of mobile robots has been discussed by Shi et al. [116]. Using this proposed strategy, the robot can recognize the obstacles and develop a collision free path. A fuzzy-neural network based intelligent motion planning strategy for a mobile robot has been proposed by Jolly et al. [117]. In this study, they designed a two dimensional fuzzification of the robot soccer field for the construction of proposed navigation system. The proposed strategy is flexible to accommodate all possible field configurations. Navigation of autonomous vehicle based on the fuzzy art map neural networks (FAMNNS) has been discussed by Chohra and Azouaoui [118]. Integration of fuzzy logic and neural network techniques for mobile

robot navigation has been suggested by Jeffril and Sariff [119]. They used fuzzy logic to extract the environment information and fed into the neural network for the training process. The performance of the proposed method has been validated through E-Puck mobile robot.

2.3.2.4 Neural-Fuzzy Techniques

A neuro-fuzzy system (shown in Figure 2.10) based mobile robot navigation has been studied by Li. [120]. In this study, he used a neural network to determine reference motion direction, and fuzzy logic is used to coordinate conflicts among the multiple types of reactive behaviours. Ng et al. [121] have implemented a neural integrated fuzzy controller for controlling the motion of multiple mobile robots. They used only nine fuzzy rules to vary the speed and direction of the robot. The proposed navigation system shown in Figure 2.11. Navigation of a mobile robot in dynamic environments using neuro-fuzzy technique has been discussed by Tsoukalas et al. [122]. They developed an idealized mobile robot called MITOS, to study the path planning in unknown environments.

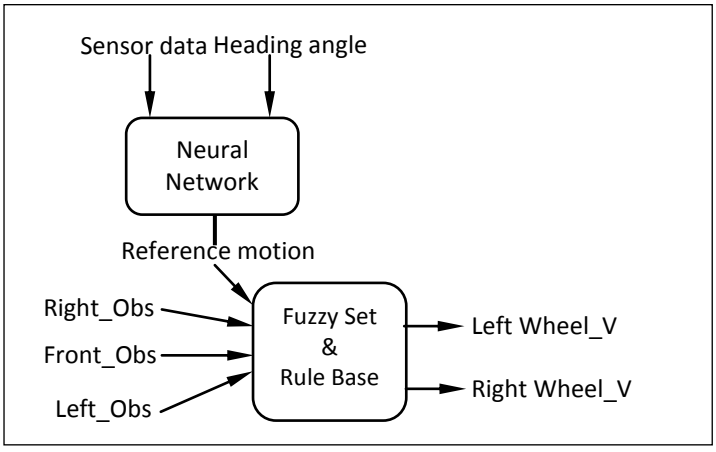


Figure 2.10 Neuro-Fuzzy approach for Mobile robot navigation proposed by Li. [120].

Godjevac and Steele [123] have presented a navigation strategy of a mobile robot using neuro-fuzzy controller. The proposed hybrid method has been successfully applied to obstacle avoidance and wall following behaviour of a mobile robot.

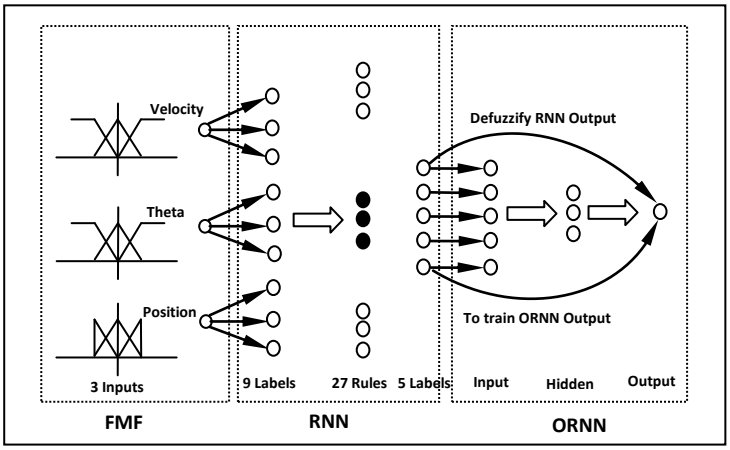


Figure 2.11 Neuro-Fuzzy approaches for Mobile robot navigation proposed by Ng and Trivedi [121].

Adaptive neuro-fuzzy system based navigation strategy of a mobile robot in unknown environments has been discussed by Nefti et al. [124]. They used mainly three sub-modules such as following a wall, avoiding obstacles and target seeking to perform the various navigation tasks. Each sub-module acts as a T-S fuzzy controller and output correspond to the orientation of the robot. Marichal et al. [125] have presented a neuro-fuzzy approach to guide a mobile robot in a maze environment. They used neuro-fuzzy system to extract the fuzzy rules and membership functions. Krishna and Kalra [126] have discussed a real time mobile robot navigation using a combination of Kohonen's self-organizing map and fuzzy art network. Wei et al. [127] have presented a new motion control method for a mobile robot based on the neuro-fuzzy system. They designed the execution part of the controller using adaptive neuro-fuzzy inference system. Ye et al. [128] have proposed a neural fuzzy system with mixed learning algorithm for obstacles avoidance of a mobile robot. They used a mixed learning algorithm, where supervised learning technique is applied for selecting the input and output membership functions, and reinforcement learning algorithm is deployed to tune the o/p membership functions of the proposed navigation system. The behaviour based mobile robot controller for indoor environments using ANFIS has been studied by Rusu et al. [129]. Wang et al. [130] have presented a neuro-fuzzy controller for mobile robot path planning. They processed the distance information through the neuro-fuzzy controller to adjust the velocity of differential drive robot. Obstacle avoidance of a mobile robot in the real world environment using neuro-fuzzy controller has been discussed by Meng and Deng [131]. The proposed neuro-fuzzy system is capable of re-adapting in a new environment.

Pradhan et al. [132] have discussed the multiple mobile robots navigation using neuro-fuzzy technique. The developed neuro-fuzzy navigation system comprises a neural network, which acts as a preprocessor for a fuzzy system. They used Petri-Net model to coordinate the multiple mobile robots. Navigation of a car-like mobile robot in a dynamic environment using neuro-fuzzy approach has been presented by Hui et al. [133]. They used different neuro-fuzzy approaches to improve the performance of the fuzzy controller. They noticed that GA optimized Mamdani controller and GA optimized neuro-fuzzy controller (T-S model) give the comparable results. Zhu and Yang [134] have proposed a neuro-fuzzy strategy to mobile robot navigation in unknown environments. They developed a fuzzy logic system with two basic behaviours, obstacle avoidance and target seeking. Two learning algorithms are designed to tune the membership function parameters and suppress the redundant rules in the rule base. Real time mobile robot navigation in a dynamic environment using ANFIS has been discussed by Parhi and Singh [135]. Demril and Khoshnejad [136] have presented a neuro-fuzzy strategy for autonomous parallel parking of a car-like mobile robot. In this navigation system, they used three sonar sensors information to decide the turning angle of the car. Park et al. [137] have developed a neuro-fuzzy rule generation strategy for backing up navigation of a car-like mobile robot. The proposed navigation method is based on the conditional fuzzy C-means (CFCM) and fuzzy equalization (FE) methods. Using these methods, they obtained a compact size of fuzzy rules, which satisfy the given target condition. Dynamic path planning of a mobile robot in the 3-D environment using ANFIS has been discussed by Woo and Polisetty [138]. Joshi and Zaveri [139] have developed a neuro-fuzzy system based motion controller for navigation of a mobile robot. In this navigation study, they used sensors information as inputs to yield wheel velocities. Erdem [140] has proposed an application of ANFIS for controlling the motion of sumo robot. In this proposed method, he used fuzzy logic, which relates sensor output signals to motor control signals and neural network based learning algorithm is used for extraction of rules and tuning of membership functions. Mutib and Matter [141] have discussed a neuro-fuzzy based navigation strategy for a mobile robot. They integrated a transputer embedded real-time controller with the robot controller board to meet various intelligence requirements for collision-free navigation. Rouabah et al. [142] have developed a navigational controller for mobile robot based on the fuzzy logic and neuro-fuzzy approaches. Mayyahi et al. [143] have discussed an adaptive neuro-fuzzy inference system for ground vehicle

navigation. In this proposed navigation study, they used four ANFIS controllers; two of which are used for controlling the wheel velocities to reach the goal position and other two controllers are used to adjust the heading direction in order to avoid the obstacles. Baturone et al. [144] have optimized an FPGA (Field-Programmable Gate Array) embedded controller for a car-like robot path planning using the neuro-fuzzy technique. Navigation and obstacle avoidance of a mobile robot using ANFIS have been discussed by Algabri et al. [145]. They performed the simulation results using Khepera simulator (Kiks) within MATLAB environment. Melingui et al. [146] have developed a new modified Type-2 fuzzy-neural network for navigation of a mobile robot in unstructured and dynamic environments. The proposed modified navigation system includes a Type-2 fuzzy process as the premise part and a two layer neural network as in the consequent part. Pothal and Parhi [147] have proposed an ANFIS navigational algorithm for multiple mobile robots navigation in unknown environments. They incorporated the ANFIS with ROBNAV software to generate the collision-free path for multiple mobile robots.

2.3.2.5 Genetic Algorithm

Pratihar et al. [151] have proposed a new, efficient path finder for the mobile robot based on a fuzzy-genetic algorithm. In this path planner, fuzzy rule base is used to find obstacles free direction locally, and GA is applied as an optimizer to search optimal obstacle free paths for the robot. Nearchou [152] has presented a genetic-based algorithm for solving the navigational problem of a robot. He assumed a known environment of the robot consisting of knot points, which are given in a graph vertices format; the GA is used to find an optimal way between the start and goal position. Perception based GA path planner (PerGA) for a mobile robot in dynamic environments has been discussed by Kubota et al. [153]. The simulation results using the proposed method demonstrate that PerGA can be able to maintain various reactive behaviours according to environmental changes. Mobile robot path planning using GA-NN has been discussed by Navarro et al. [154]. In this proposed navigation system shown in Figure 2.12, the GA is used to train the neural network.

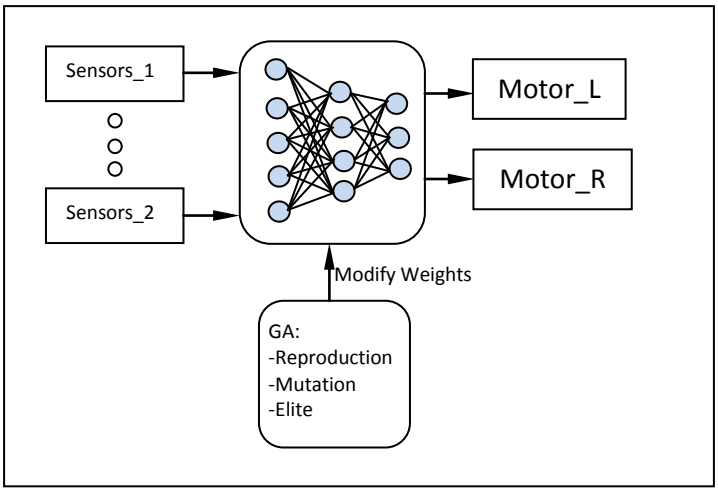


Figure 2.12 GA-NN approach for Mobile robot navigation proposed by Navarro et al. [154].

Genetic-Fuzzy based optimal navigational controller for a mobile robot has been presented by Liu et al. [155]. In this proposed navigation method, they used a genetic algorithm to optimize the fuzzy rule table and membership functions to obtain the shortest route for the robot. Castillo et al. [156] have discussed the multi-objective genetic algorithm (MOGA) for path optimization of a mobile robot. They compared the performance results using both conventional GA and MOGA. It has been noted that both types of GA controllers are efficient for solving the navigational problem of a robot. GA tuned fuzzy logic method for mobile robot path planning has been presented by Hassanzadeh and Sadigh [157]. They applied GA to optimize the I/P and O/P membership functions of the fuzzy controller. Kiks-II simulation tool is used to demonstrate the results of proposed navigation system. Taharwa et al. [158] have presented a new idea using GA for solving the path planning problem of a mobile robot in static environments. They framed a simplified fitness function that utilizes the path length as optimization criteria. Liu et al. [159] have discussed the path planning problem of a mobile robot using adaptive GA. In this adaptive GA, they used the specialized genetic operators and parameters of GA are adjusted adaptively to solve the path planning problem of a robot. Path planning strategy for multiple mobile robots using Petri-GA model has been presented by Mohanta et al. [160].The proposed algorithm is used to determine the suitable heading angle of the robots to find targets. The Petri-Net theory is implemented to avoid the inter-collision of the robots more effectively than the standalone GA. Global path planning for a mobile robot using A* and genetic algorithm have been proposed by Zhang et al. [161]. They used A* algorithm to find the shortest

path in the grid environment and genetic algorithm is applied to obtain the globally optimized path for the robot. Nagib and Gharieb [162] have presented a new methodology for global path planning of a mobile robot using genetic algorithm. In this proposed algorithm, each via point in the 2D workspace is a gene, which is represented by a binary code. The number of genes in one chromosome is a function of the number of obstacles in the map. Jiang et al. [163] have discussed the motion planning problem of a mobile robot using improved genetic algorithm. The proposed algorithm uses artificial potential field method to establish the initial population and increases the weight values in the fitness function to improve the smoothness of the robot's path. Global path planning of a mobile robot using memetic algorithm has been presented by Zhu et al. [164]. The proposed navigation strategy combined the effect of genetic algorithm based global path planning and a local path refinement. In each GA evolution, the local path refinement is used to the individual GA for improving the paths encoded.

2.3.2.6 Particle Swarm Optimization

A modified PSO named stochastic PSO has been proposed by Chen and Li [166] for mobile robot navigation. They developed S-PSO with high exploration ability so that a small swarm size can generate the smooth paths for the robot. Lu and Gong [167] have described a new fitness function for mobile robot navigation using PSO. The developed fitness function of a particle is based on the position of the obstacles and targets in the environment. The optimal path is formed with this proposed algorithm. Gong et al. [168] have developed a new path planning strategy in uncertain environments using PSO. They framed a global optimal path by using PSO in an uncertain environment and finally a local optimal strategy is implemented to handle the unknown information detected by the robot in real time situation. Masehian and Sedighizadeh [169] have discussed a multi-objective based PSO for the robot path planning. They deployed the PSO for global path planning while the probabilistic road map method (PRM) is applied for obstacle avoidance (local path planning). A specialized PSO algorithm for mobile robot navigation has been proposed by Li et al. [170]. The concept of ‘active region’ is introduced for particle in proposed algorithm to narrow the search space and increase the search rate. The invalid particles are replaced directly by a global optimum solution to reduce the iteration time. Path planning approach for a mobile robot using multi-objective PSO has been addressed by Gong et al. [171]. They introduced a self-adaptive mutation operation based on the degree of a path blocked by obstacles to improve the feasibility of a new

path. An intelligent motion planning and trajectory tracking of a mobile robot in a known environment has been presented by Susic et al. [172]. The PSO is used to generate the collision free paths from the start point to the goal point and the interpolation of collision-free paths are solved by radial basis neural network and trajectory is generated based on the interpolation path. Cooperative motion planning of swarm mobile robots using PSO and multi-dynamic systems has been discussed by Tang and Eberhard [173]. The entire swarm mobile robots are guided by the PSO and multi-dynamic systems are also included considering the physical properties of the robot. Several simulation results are presented to verify the proposed motion planning strategy. Mohamed et al. [174] have described a new path planner for a mobile robot using simultaneous localization and mapping (SLAM) and accelerated PSO. In this proposed PSO algorithm, they carried out many experiments to obtain the best parameters settings in PSO to achieve the required results. A new random PSO algorithm for local path planning of mobile robots in unknown dynamic environments has been discussed by Mohajer et al. [175]. The criterion for optimal selection depends on the particles distance to target, and Gaussian cost function is assigned to detect obstacles. Some simulation results of the proposed algorithm are performed and compared by the artificial potential field method. An improved multi-objective PSO algorithm for mobile robot path planning has been successfully implemented by Zhang et al. [176]. Firstly, they defined a fuzzy membership function to calculate the risk degree of the path. After considering the two performance merits; the risk degree and the distance of path, the path planning problem is defined as a bi-objective optimization problem with uncertain coefficients. Then PSO algorithm is applied to tackle this problem. Finally, simulation results show the effectiveness of the proposed method. Cai and Yang [177] have applied a novel potential field based PSO algorithm approach for cooperative target searching of multiple mobile robots in completely unknown environments. In this proposed algorithm, the potential function is used as the fitness function for PSO, which evaluate the exploration priority of the unknown environment. The cooperation rules are defined to lead the multi-robot system to explore the unknown environment. Deepak et al. [178] have developed an intelligent navigational controller for the mobile robot based on the PSO algorithm. They designed a fitness function that is based on the distance between each particle of the swarm and target and the distance between each particle of the swarm and the nearest obstacle. Depending on the fitness function value, the global best position is selected, and the robot

reaches these positions in sequence to reach the target. A new global path planning for a mobile robot in a static environment by combining BBO (Biogeography-based Optimization), PSO, and approximate Voronoi boundary network (AVBN) have been presented by Mo and Xu [179]. In this proposed navigation strategy, the PSO is implemented to increase the diversity of population in BBO and then the obtained Biogeography particle swarm optimization algorithm (BPSO) is used to optimize the paths in the path network obtained by AVBN modeling.

2.3.2.7 Ant Colony Optimization and Artificial Immune System

A method of collision-free trajectory for a cooperative robot team using a combination of Cellular Automata (CA) and ACO techniques have been discussed by Ioanndis et al. [180]. The proposed algorithms have been tested in a system using real world simulation environment called Webots. A fast two stage path planning for mobile robots using ACO algorithm has been addressed by Chen et al. [181]. The path planning algorithm is splitting into two stages; in the preprocessing stage the scent information is broadcasted to the whole map and then ants do path planning under the direction of scent information in the second stage. The proposed algorithm is demonstrated on various complex maps and compared with different algorithms. Implementation of ACO algorithm for multiple mobile robots navigation in cluttered environments has been presented by Parhi and Pothal [182]. Path planning for mobile robots using improved ACO algorithm has been addressed by Hsu et al. [183]. They improved the ACO algorithm by considering two aspects; continuous tuning of setting parameters and a new mechanism is established for updating pheromone. Yuan et al. [184] have proposed a novel method for robot path planning based on the ACO and AIS. The mechanism of stimulation and suppression between antigen and antibody are used to search the path, that solves complex environment modeling of ant colony algorithm and enhance the path planning efficiency. The simulation results indicated that the ACO-AIS performs better compared to standalone ACO and AIS.

Adaptive immune based motion planner for mobile robot navigation in unknown cluttered environments has been introduced by Deepak and Parhi [186]. An improved path planning strategy for mobile robot based on the Artificial Immune System (AIS) has been presented by Yuan et al. [187]. They choose a new antibody choice operator that improved the search rationality of immune network. Application of artificial immune

system based motion planner for mobile robot path planning has been discussed in [188-191].

2.3.2.8 Cuckoo-Search Algorithm

A new metaheuristic algorithm called Cuckoo Search (CS) has been formulated by Yang and Deb [192]. This proposed algorithm based on the obligate brood parasitic behavior of some cuckoo species in combination with the levy flight behavior of some birds and fruit flies. Yang and Deb [193] have applied the CS algorithm to solve the engineering design problems, including the design of spring and welded beam structures. They observed that CS algorithm provides better solution compared to PSO algorithm. To improve the accuracy and convergent rate of CS algorithm, an improved CS algorithm has been proposed by Valian et al. [194]. They used a proper strategy for tuning the fixed parameters (P_a, α) in the CS algorithm. Then the modified CS algorithm is implemented for training of feedforward neural network. A modification of the CS algorithm has been investigated by Walton et al. [195]. They made two modifications to the original CS algorithm; the first modification is made to levy flight step size (α) and the second is to add information exchange between the eggs in an attempt to speed the convergence to a minimum. Training of spiking neural network using CS algorithm has been described by Vazquez [196]. The accuracy of the algorithm has been tested using several pattern recognition problems. A new hybrid metaheuristic algorithm based on the DE and CS has been proposed by Wang et al. [197] to solve the path planning problem of UCAV (uninhabited combat air vehicle). They have applied the DE to optimize the process of selecting cuckoo of the CS algorithm during the process of cuckoo in nest updating. Jati et al. [198] have implemented a discrete CS algorithm for solving the travelling salesman problem. In this discrete CS algorithm, the step size is defined as the distance between the cuckoo and best cuckoo at its generation. Training of neural network using CS algorithm has been addressed in [199]. Chandrasekaran and Simon [200] have proposed a hybrid CS algorithm integrated with the fuzzy system for solving the multi-objective scheduling problem. The fuzzy set is used to build the fuzzy membership search domain, where it contains all possible compromise solutions, and CS algorithm is applied to explore the best solution within the fuzzy search domain. A comparison studies of CS, PSO, DE (Differential evolution), and Artificial Bees Colony (ABC) algorithm have been discussed by Civicioglu and Besdok [201]. They noticed that the CS and DE algorithms are performed better than the PSO and ABC algorithms. Yang and Deb [202] have

formulated a new CS algorithm to solve the multi-objective optimization problems. Trajectory optimization of robotic arm using different metaheuristic algorithms such as ABC, BBO (Biogeography-based optimization), GSA (Gravitational Search Algorithm), CS, FA (Firefly Algorithm)), BA (Bat Algorithm), and TLBO (teaching–learning-based optimization) have been studied by Savani et al. [203]. They observed that the TLBO, ABC, and CS algorithms are performed well for the considered trajectory optimization problem for the searching of best solutions, convergence and workspace with obstacle avoidance. Application of Bio-inspired search algorithms to solve robotic assembly line balancing problems have been studied by Nilakantan et al. [204]. PSO and CS-PSO hybrid optimization algorithms have been applied to balance the robotic assembly line with objective of minimizing the cycle time. Recent advances and applications of CS algorithm have been discussed by Yang and Deb [205]. CS algorithm in other fields has been given in [206-208].

2.3.2.9 Invasive-Weed Optimization

Invasive Weed Optimization (IWO) algorithm is a new biologically inspired algorithm that has been introduced by Mehrabian and Lucas in 2009[209]. The metaheuristic algorithm mimics the robustness, adaptation and randomness process of weed colonization behaviour of weeds. They compared results of the proposed algorithm with other evolutionary based algorithms such as GA, MA (Memetic Algorithm), PSO and SFL (Shuffled Frog-Leaping) through the benchmark multi-dimensional functions. It has been observed that IWO performs better than the results of other models. Hybridization of IWO and PSO for fast and global optimization has been proposed by Hajimirsadegli and Lucas [210]. The efficiency of the proposed hybrid algorithm has been tested and compared with IWO, PSO, and some other algorithms through common benchmark functions. Comparative studies of five evolutionary algorithms have been discussed by Krishnananda et al. [211]. The evolutionary algorithms (GA, PSO, ABC, IWO, and AIS) are tested on the multi variable benchmark functions. They noted that AIS and IWO algorithms performed well for high dimensional problems compared to other algorithms. Implementation of discrete invasive weed optimization to cooperative multiple task assignments of UAVs have been presented by Ghalenoei et al. [212]. The modification in the spatial dispersal step of IWO has shown the satisfactory performance. Efficient motion planning by a robot arm without obstacle collision using IWO algorithm has been proposed by Sengupta et al. [213]. They formulated a suitable cost function to

track the optimal path between an initial and final configuration of the robot arm joints. Multi-objective optimization with IWO algorithm has been discussed by Kundu et al. [214]. The results of the proposed algorithm have performed better than the existing evolutionary algorithms namely NSGAIILS, DECMOSA-SQP, MOEP and MOEADGM. Yin et al. [215] have proposed an improved IWO algorithm based on hybridization with a genetic algorithm. The developed hybrid algorithm improves the performance of the weeds and reduces the chance of getting trapped in local point. Basak et al. [216] have proposed a hybrid system that is combining the effect of DE and IWO for solving the real parameter and single-objective optimization problem. They noted that DE-IWO has a greater explorative power than standalone DE and IWO. IWO algorithm has been applied in various fields of engineering systems [217-220].

2.4 Discussions

The literature review on the mobile robot navigation is classified according to different approaches used by the researchers. Figure 2.13 shows a bar graph with the percentage of classical and computational Intelligence (CI) approaches employed by the authors. It has been noticed that, in the recent decade about 80% of the studied literature uses CI techniques to solve the path planning problem of mobile robots, whereas classical approaches have solved only 20% of path planning problems. However, the researchers have concentrated towards the CI approaches in order to reduce the computational complexity and to obtain the nearest optimal solution in a reasonable time.

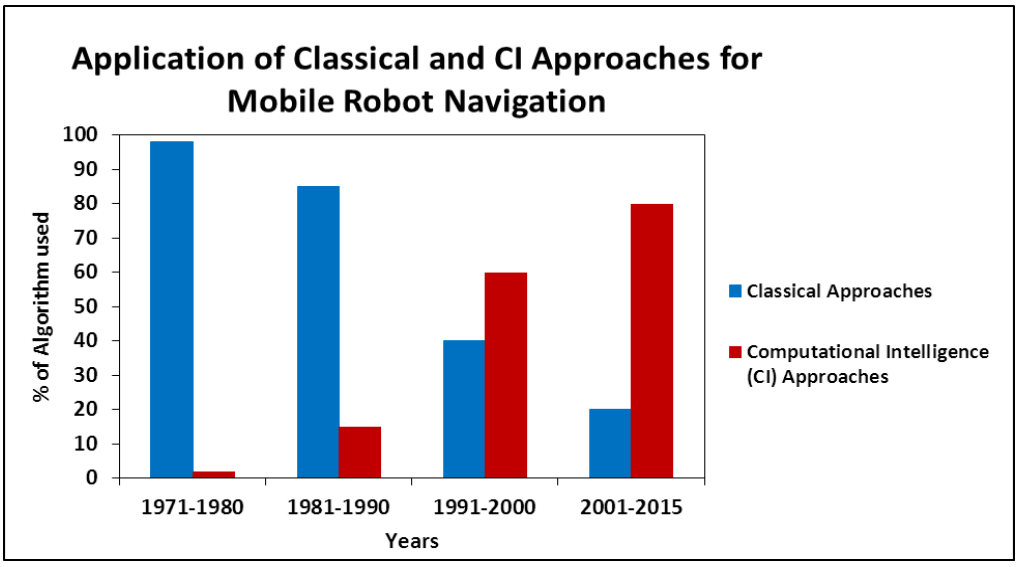


Figure 2.13 Application of classical and CI approaches for Mobile Robot navigation.

Figure 2.14 presents a pie chart to provide the percentage of paper reviewed based on the CI methods to solve the path planning problem of mobile robots. It has been illustrated that about 60% of the studied literature uses fuzzy logic, neural network and genetic algorithm to solve the navigational problem, whereas only 40% of the literature employs nature-inspired algorithms to solve the same problem. However, path planning approaches using hybrid techniques are limited in the literature.

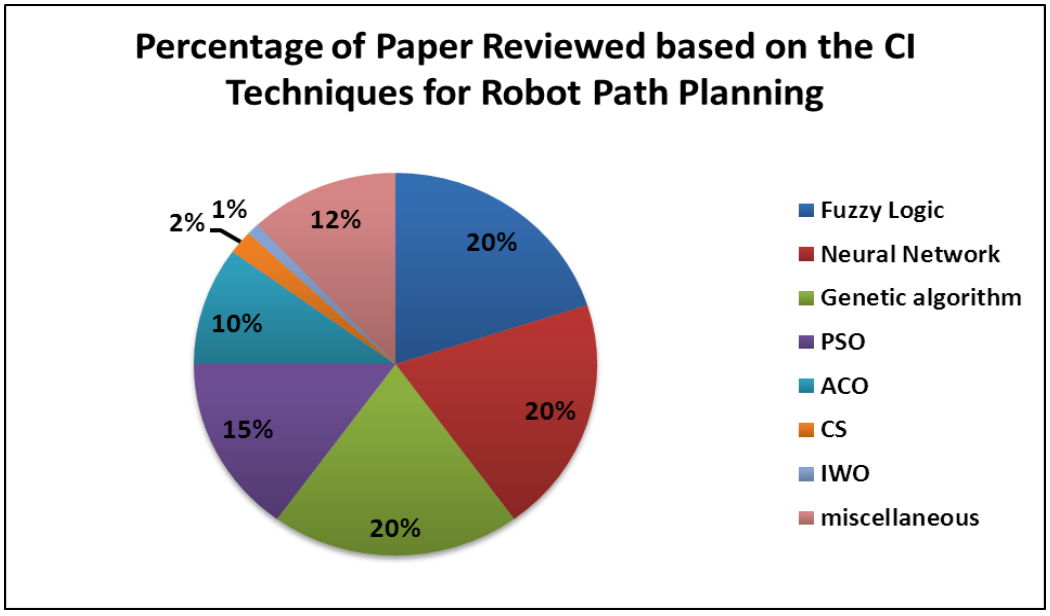


Figure 2.14 Percentage of paper reviewed based on the CI techniques for Robot path planning.

2.5 Summary

After an exhaustive study made in the literature, the following conclusions are drafted below:

- The various navigational techniques have been developed by the researchers so far for controlling the motion of mobile robots in unknown or partially known environments.
- It has been observed that the CI techniques are widely used for mobile robot path planning compared to classical techniques.
- Most of the articles are focused on efficient heuristic algorithms that can be employed for solving the navigational problem of the mobile robots. However, it has been observed that these algorithms don't always perform well as per expectation.
- A limited number of research has been carried out based on the natured-inspired algorithms to solve the path planning problem of mobile robots in unknown or partially known environments.

The present research work proposes new navigational methods i.e. ANFIS (Adaptive Neuro-Fuzzy Inference System), CS (cuckoo Search) algorithm, IWO (Invasive Weed Optimization), CS-ANFIS and IWO-ANFIS hybrid techniques for solving the path planning problem for single and multiple mobile robots in unknown or partially known environments. As per the author's knowledge theses are algorithms are not yet implemented for robot path planning problems.

CHAPTER-3

3. MODELING AND ANALYSIS OF WHEELED ROBOT KINEMATICS

Wheeled mobile robots (WMR) have been an interesting research area in the field of robotics over the past few decades. The study of kinematics deals with the motion of points, bodies and systems of bodies without reference to the forces which cause the motion. In mobile robotics, it is necessary to study the kinematics and dynamics behavior of the robot in order to develop suitable mobile robots for performing various assignments and to understand how to frame motion controlling software for a robot. This chapter describes the details of the kinematic analysis of a wheeled mobile robot.

3.1 Introduction

The wheeled mobile robots consist of a class of mechanical systems characterized by both kinematics and dynamics constraints and, therefore, cannot be ignored from the model equations. The significance is that the navigation techniques designed for the mobile robot without considering constraints are no more feasible. Therefore, kinematic modeling of a wheeled mobile robot is still a relevant issue [10-13]. Many researchers have designed methodologies for kinematic and dynamic analysis of wheeled mobile robot. Before understanding the motions of a robot, it is necessary to describe the involvement of each wheel contributes to motion. Each wheel of a robot has a crucial role to move the whole robot, and each wheel generates various constraints on the robot's movement; for example to avoid slip laterally. In the current chapter, we described the expression of robot motion in a global or base reference plane as well as the robot's local reference plane. Then, using this expression the forward kinematics for the robot has been derived. Next, kinematic constraints for each wheel have been derived and these constraints are combined to express the whole robot's kinematic constraints. With the help of this geometry and wheel behavior, we can calculate the paths and the trajectory that define the robot's maneuverability.

3.2 Representing the Mobile Robot Position

Let us consider the kinematic model of a wheeled mobile robot (WMR) and the position of the robot in the plane as shown in Figure 3.1.

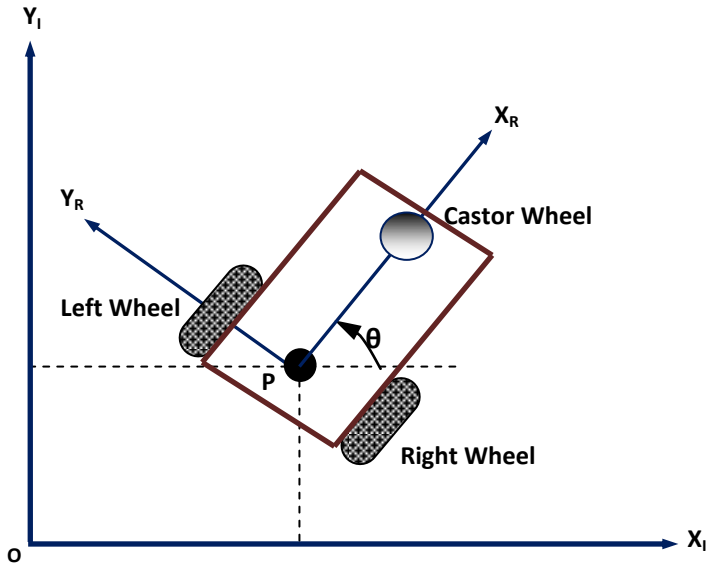


Figure 3.1 Schematic diagram of the Robot position in horizontal Plane.

The global or base reference frame (OX_IY_I) is fixed in the plane of motion, and the moving frame (OX_RY_R) or local reference frame attached to the robot [223]. In this work, we have considered that the mobile robot is a rigid body connected with wheels, and it is travelling in a horizontal plane surface. The posture of the robot ξ in the reference plane is then completely specified by three variables x, y, θ in the vector form;

$$\xi = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \tag{3.1}$$

In order to find out the robot position, it will require mapping the motion along the axes of the base reference frame to the motion along the axes of the robot's moving frame or local reference frame. The orthogonal rotation matrix expressing the orientation of the base reference frame (OX_IY_I) with respect to the robot moving frame (OX_RY_R) is given by

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

The above rotation matrix can be used to map the motion in the base reference frame *(OX_IY_I)* to motion in terms of the moving frame or the robot frame *(OX_RY_R)*. This expression is denoted by $R(\theta)\dot{\xi}_I$ because the calculation of this equation depends on the value θ;

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

3.3 Kinematic Wheels Description

The wheeled mobile robots are widely used in many areas due to their advantages of locomotion. In general, wheeled mobile robots move faster and more energy efficient than other locomotion systems (e.g., legged robots or tracked vehicles). From the control point of view, less control effort is needed, due to their simple mechanisms and reduced stability issues. Although it is difficult to overcome rough terrain or uneven environment conditions, wheeled mobile robots meet for a large class of target environments in practical implementations.

Before we derive the expressions of kinematic constraints for various types of wheels, different important assumptions are taken into consideration and shown in Figure 3.2.

a) The plane of the wheel always remains vertical, and the wheel's motions are purely rolling leading to a zero velocity at the contact point.
b) No slipping, skidding, sliding or friction for rotation around the contact point.

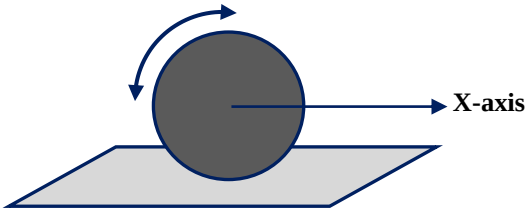


Figure 3.2(a) Rolling Motion.

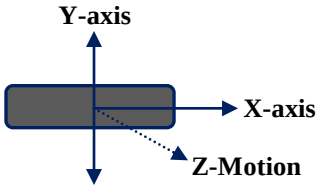


Figure 3.2(b) Lateral Slip.

3.3.1 Fixed Standard Wheel

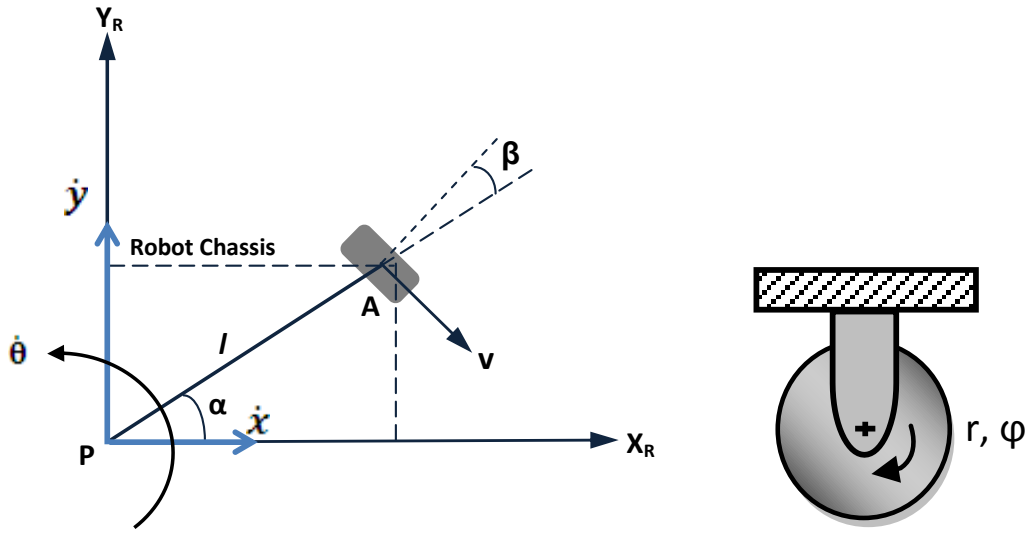


Figure 3.3 Geometric parameters of fixed standard wheel.

The schematic diagram of fixed standard wheel is shown in Figure 3.3. Where 'A' is denoted the center of the fixed wheel and also is a fixed point of the robot frame (Fig.3.3). The position of the 'A' is defined using polar coordinates by distance PA=l and the angle α. The orientation of the plane of the wheel w.r.t PA is represented by the constant angle β.The rotation angle of the fixed wheels around its (horizontal) axle is denoted φ (t) and radius of the wheel is 'r'. So the position of the fixed wheel is thus defined by four parameters α, β, l, r and its motion by time varying angle ϕ(t). When the components of the velocity of the contact point are projected on the fixed wheel plane, we can derive two following constraints:

- Pure rolling condition:

$$\left[-sin(\alpha+\beta)\ \ cos(\alpha+\beta)\ lcos\beta\right]R(\theta)\dot{\xi}+r\dot{\varphi}=0 \tag{3.4}$$

- Nonslip condition:

$$\left[cos(\alpha+\beta)\ sin(\alpha+\beta)\ lsin\beta\right]R(\theta)\dot{\xi}=0 \tag{3.5}$$

3.3.2 Steered Standard Wheel (Centered orientable wheel)

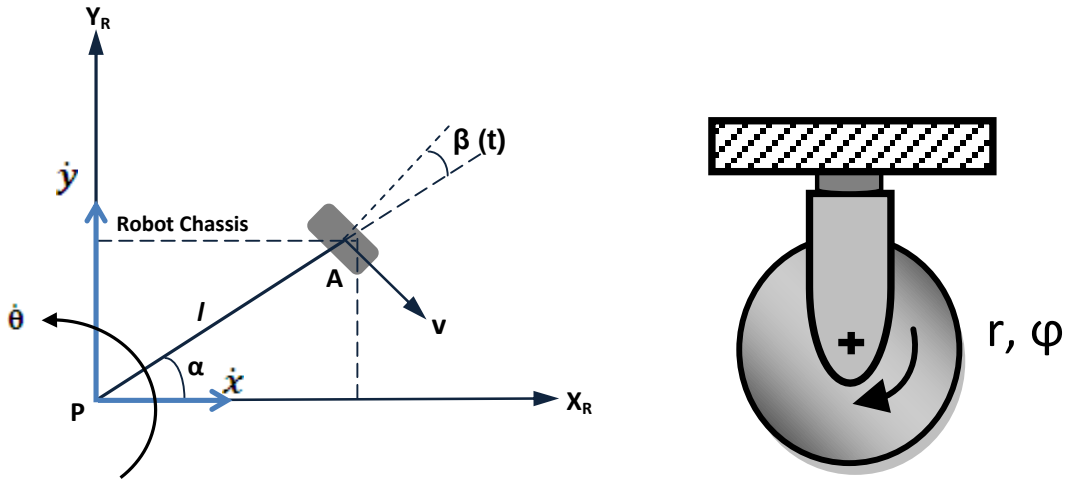


Figure 3.4 Geometric parameters of steered standard wheel.

A steered standard wheel is such that the motion of the wheel plane w.r.t the frame is a rotation around a vertical axis passing through the center of the wheel (Figure 3.4). The expression is same as for a fixed standard wheel; the only difference is that now the angle β(t) is time varying. So the position of the wheel is characterized by three constant parameters l, α, r and its motion w.r.t the frame by two time-varying angles β(t) and φ(t). The expressions for various conditions are as follows,

- Pure rolling condition:

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

- Nonslip condition:

$$\left[cos(\alpha+\beta) \quad sin(\alpha+\beta) \quad lsin\beta\right]R(\theta)\left(\dot{\xi}\right)=0 \tag{3.7}$$

3.3.3 Castor Wheel (Off centered orientable wheel)

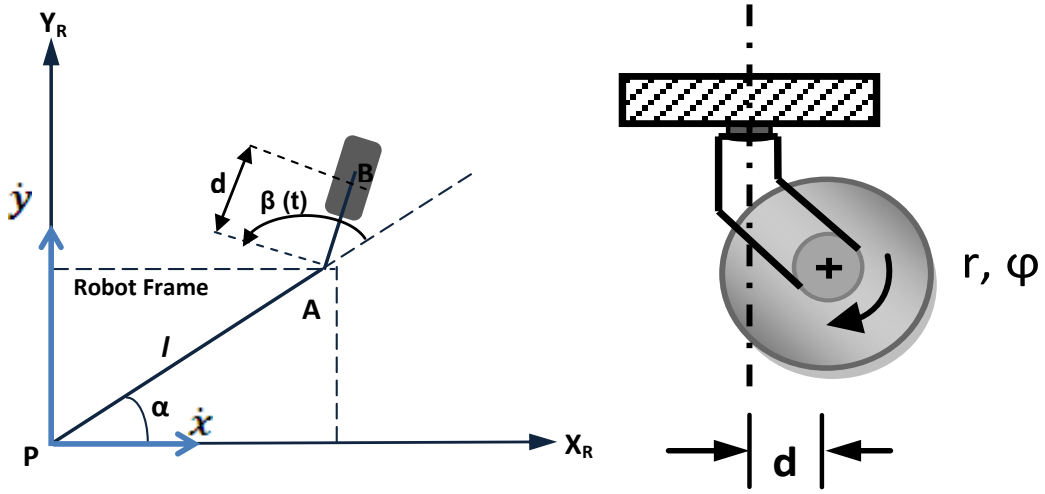


Figure 3.5 Geometric parameters of Castor wheel.

In this type of wheel, the rotation of the wheel plane is around a vertical axis which doesn't pass through the center of the wheel (Figure 3.5). 'B' is the center of the wheel and is connected to the frame by a rigid bar AB of length'd' which can be rotated around a fixed vertical axis at point 'A'. The position of the wheel is defined by four constants α, l, r, d and its motion by two varying angles β(t) and φ(t). For this wheel, constrains are in following form:

- Pure rolling condition:

$$\left[-sin(\alpha+\beta) \quad cos(\alpha+\beta) \quad lcos\beta\right]R(\theta)\left(\dot{\xi}\right)+r\dot{\varphi}=0 \tag{3.8}$$

- Nonslip condition:

$$\left[cos(\alpha+\beta) \; sin(\alpha+\beta) \; d+lsin\beta\right]R(\theta)\dot{\xi}+d\dot{\beta}=0 \tag{3.9}$$

3.3.4 Swedish Wheel

The position of the Swedish wheel w.r.t to the frame is characterized, as for fixed standard wheels, by the three constant parameters α, β and l (Figure 3.6). An additional parameter is necessary to describe the direction w.r.t to the plane of the zero component of the velocity of the contact point denoted by the angle ϒ.

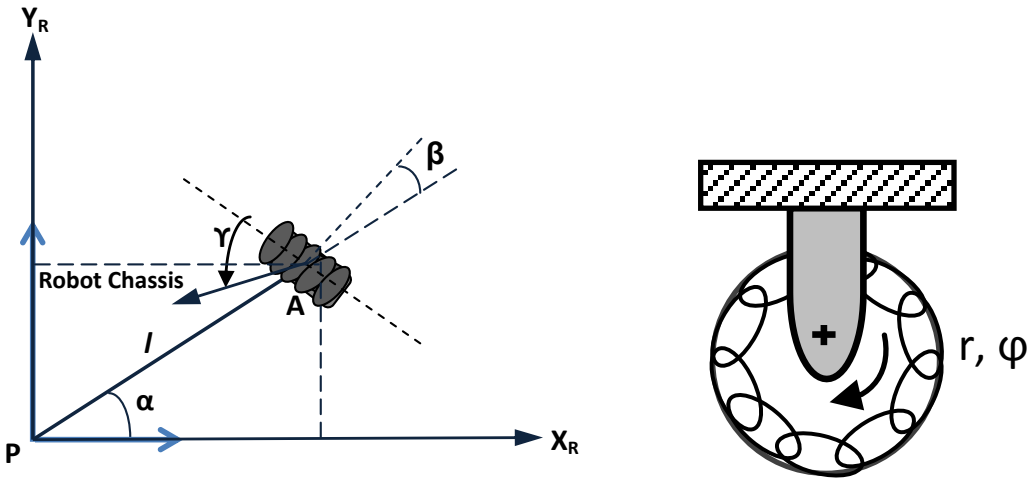


Figure 3.6 Geometric parameters of Swedish wheel.

For this wheel, the constraints are expressed as;

- Pure rolling condition:

$$\left[\sin(\alpha+\beta+\gamma)\ -\cos(\alpha+\beta+\gamma)-l\cos(\beta+\gamma)\right]R(\theta)\left(\dot{\xi}\right)-r\cos\gamma\,\varphi=0 \qquad (3.10)$$

- Nonslip criteria is not constrained because of the free rotation $\dot{\varphi}_{sw}$ of the small rollers:

$$\left[\cos(\alpha+\beta+\gamma)\ \ \sin(\alpha+\beta+\gamma)\ l\sin(\beta+\gamma)\right]R(\theta)\left(\dot{\xi}\right)-r\sin\gamma\,\varphi-r_{sw}\dot{\varphi}_{sw}=0 \qquad (3.11)$$

3.3.5 Spherical Wheel

The wheel radius 'r' being the radius of the sphere, and its rotation angle is represented by φ(t) (Figure 3.7). The spherical wheel is clearly omnidirectional and places no constraints on the robot chassis kinematics. So the equation simply describes the roll rate of the sphere in the direction of motion V_A of point 'A' of the mobile robot.

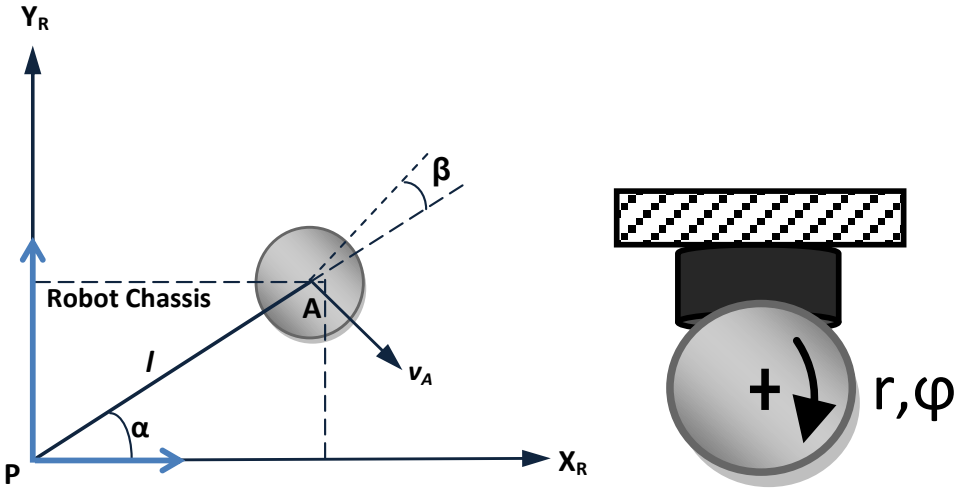


Figure 3.7 Geometric parameters of Spherical wheel.

- Pure rolling condition:

$$\left[-sin(\alpha+\beta)\ \ cos(\alpha+\beta)\ lcos\beta\right]R(\theta)\dot{\xi}+r\dot{\varphi}=0 \tag{3.12}$$

- No-slip condition:

$$\left[cos(\alpha+\beta)\ sin(\alpha+\beta)\ lsin\beta\right]R(\theta)\dot{\xi}=0 \tag{3.13}$$

3.4 Restriction on Robot Mobility

Let us consider a general wheel robot, consisting of N wheels of the four above described category. So, the total number of wheels in the is robot as follows:

$$N = N_f + N_c + N_{ca} + N_{sw} \tag{3.14}$$

The number of wheels of each type is represented by:

N_f = No. of fixed standard wheels

N_c = No. of centered orientable wheel

N_{ca} = No. of castor wheels

N_{sw} = No. Swedish wheels

The position of the wheel robot is represented by the following vector form.

- Posture coordinates: $\xi(t) = \begin{bmatrix} x(t) \\ y(t) \\ \theta(t) \end{bmatrix}$ the position of the coordinates in the plane.

- Rotation coordinates: $\varphi(t) = \begin{bmatrix} \varphi_f(t) \\ \varphi_c(t) \\ \varphi_{ca}(t) \\ \varphi_{sw}(t) \end{bmatrix}$ for the rotation angles of the wheels around their horizontal axis of the rotation.

a) The rolling constraints of all the wheels can be written under general matrix form:

$$J_1(\beta_s) R(\theta) \dot{\xi} - J_2 \dot{\varphi} = 0 \tag{3.15}$$

In this expression $J_1(\beta_s)$ represented a matrix with projection of all wheels to their motion along their wheel planes:

$$J_1(\beta_s) = \begin{bmatrix} J_{1f} \\ J_{1c}(\beta_c) \\ J_{1ca}(\beta_{ca}) \\ J_{1sw} \end{bmatrix} \tag{3.16}$$

where, J_{1f}, J_{1c}, J_{1ca} and J_{1sw} are $[N_f \times 3], [N_c \times 3], [N_{ca} \times 3]$ and $[N_{sw} \times 3]$ matrices whose forms arise from the constraint equations (3.4), (3.6), (3.8) and (3.10) respectively. J_{1f} and J_{1sw} are constant while J_{1c} and J_{1ca} are time varying respectively through $\beta_c(t)$ and $\beta_{ca}(t)$. J_2 is a constant diagonal [N x N] matrix whose diagonal values are radii r of all standard wheels.

b) Sliding constraints of all wheels can be written in the matrix form as follows;

$$C_1(\beta_s) R(\theta) \dot{\xi} = 0 \tag{3.17}$$

$$C_1(\beta_s) = \begin{bmatrix} C_{1f} \\ C_{1c}(\beta_c) \\ C_{1ca}(\beta_{ca}) \end{bmatrix} \tag{3.18}$$

where C_{1f}, C_{1c} and C_{1ca} are $[N_f \times 3], [N_c \times 3] and [N_{ca} \times 3]$ matrices whose forms arise from the constraint equations (3.5), (3.7), (3.9) and (3.11). C_{1f} is the constant while C_{1c} and C_{1ca} are time varying.

3.4.1 Degree of Mobility

The mobility of a robot frame is defined as its ability to move directly in the environment. The basic constraint rule is that every wheel, must satisfy its sliding kinematic constraint. Now consider the $N = N_f + N_c$ constraints from the equation (3.17) and written as

$$C_{1f} R(\theta) \dot{\xi} = 0 \tag{3.19 a}$$

$$C_{1c}(\beta_c) R(\theta) \dot{\xi} = 0 \tag{3.19 b}$$

For both of these constraints to be satisfied, when the motion vector $R(\theta)\dot{\xi} = 0$ must belong to the null space of the projection matrix $C_1(\beta_s)$:

$$C_1(\beta_s) = \begin{bmatrix} C_{1f} \\ C_{1c}(\beta_c) \end{bmatrix} \tag{3.20}$$

From the above investigation, we observed that the robot frame kinematics is a function of the set of independent constraints arising from all the standard wheels. Equation (3.17) defines all sliding constraints imposed by the wheels of the mobile robot. Therefore, the rank $[C_1(\beta_s)]$ denotes the number of independent constraints. Before that it is significant to notice that equations (3.19 a) and (3.19 b) have a significant geometrical interpretation. At each time, the motion of the robot can be noticed as an instantaneous rotation around the instantaneous center of rotation (ICR) whose position with respect to the chassis can be time varying. Consequently, at each instant the velocity vector of any point of the frame is orthogonal to the straight line joining this points and the ICR. In fact this is true for the centers of the fixed standard wheel and steerable standard wheels. This indicates that at each time instant, the horizontal rotation axles of all the fixed standard wheels and steerable standard wheels are concurrent at the ICR. This observation is shown in Figure 3.8.

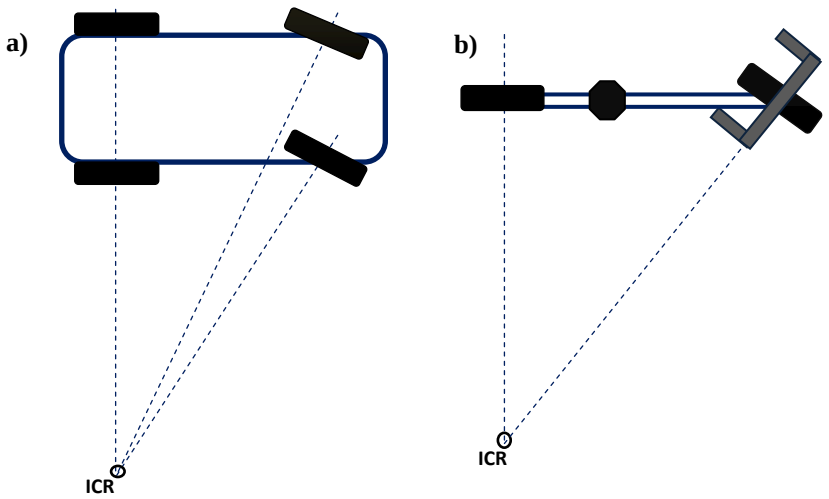


Figure 3.8 a. Four-Wheel vehicle with Ackermann steering configuration

Figure 3.8 b. Bicycle

The rank of the matrix $\left[C_1(\beta_s)\right]$ depends on the construction of the mobile robot. We express the degree of mobility δ_m of a mobile robot as

$$\delta_m = dim N\left[C_1(\beta_s)\right] = 3 - rank\left[C_1(\beta_s)\right] \tag{3.21}$$

Hence, the degree of mobility satisfies the possible range of rank values of any robot as $0 \le rank\left[C_1(\beta_s)\right] \le 3$.

Case -1:

If the $rank\left[C_1(\beta_s)\right] = 0$

This is only possible if there are zero independent constraints in $\left[C_1(\beta_s)\right]$. In this condition, there are neither fixed nor centered orientable wheel attached to the robot frame $\left(N_f = N_c = 0\right)$.

Case -2:

If the $rank\left[C_1(\beta_s)\right] = 3$

Then the mobile robot is completely constrained in all possible directions and so, any motion on the plane is impossible. Therefore, it is concluded that δ_m must lies between 0 and 3.

3.4.2 Degree of Steerability

The degree of steerability is defined as the number of centered orientable wheel that can be oriented independently in order to steer the mobile robot. The degree of steerability is expressed as follows δ_s :

$$\delta_s = rank[C_{1c}(\beta_c)] \tag{3.22}$$

The possible range of δ_s can be specified as: $0 \le \delta_s \le 2$. The case $\delta_s = 0,$ implies that the wheeled mobile robot has no steerable standard wheels, $N_s = 0$, When the $\delta_s = 1,$ is most common, the mobile robot frame consists of one or more centered orientable wheels.

3.4.3 Robot Maneuverability

Maneuvrability plays a very vital issue for a mobile robot to solve its assignments. The total degrees of freedom that a mobile robot can manipulate, known as the degree of maneuverability δ_M, and it can be expressed in terms of degree of steerability and mobility:

$$\delta_M = \delta_m + \delta_s \tag{3.23}$$

Hence, there are five types of wheeled mobile robots, corresponding to the five pairs of values of δ_{m} and δ_s according to following Table-3.1:

Table-3.1 (Five Types of Robot's Maneuverability)

δ_m	3	2	2	1	1
δ_s	0	0	1	1	2
δ_M	3	2	3	2	3

The design features of each type of wheeled robot are now explained briefly.

Type-I

$\delta_m = 3, \delta_s = 0, \delta_M = 3$

These types of robots have no fixed standard wheels $\left(N_f = 0\right)$ and no centered orientable wheels. Such types of robots are called omnidirectional because they have a full mobility and can be moved at each instant in any direction without any reorientation.

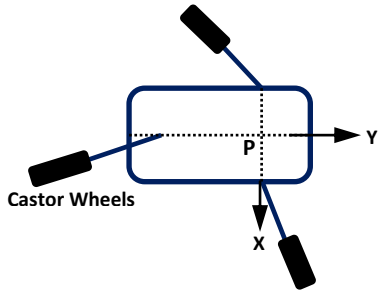


Figure 3.9 Omnidirectional type (Three castor wheels*)

Type-II

$\delta_m = 2, \delta_s = 0, \delta_M = 2$

These robots have no steered standard wheels $\left(N_c = 0\right)$

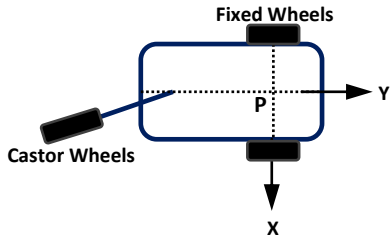


Figure 3.10 Differential type (Two fixed wheel and one castor wheel*)

Type-III

$\delta_m = 2, \delta_s = 1, \delta_M = 3$

These categories of robots have no fixed wheel $(N_f = 0)$ and at least one centered orientable wheels $(N_c \geq 1)$ If there are more than one centered orientable wheel, their orientation must be coordinated in such a way that rank $C_{1c}\left(\beta_c\right) = \delta_s = 1$.

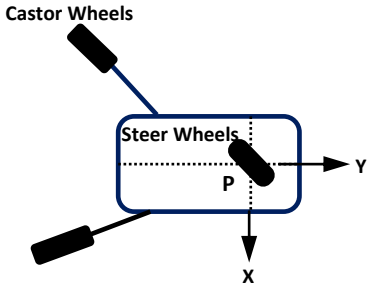


Figure 3.11.Omni-steer (one steer wheel and two castor wheel*)

Type-IV

$\delta_m = 1, \delta_s = 1, \delta_M = 2$

These types' robots have one or several fixed standard wheels with a single common axle. They have also one or several centered orientable wheels with the condition that the center of one of them is not positioned on the axle of the fixed standard and that their orientation are synchronized in such way that rank $C_{1c}(\beta_c) = \delta_s = 1$.

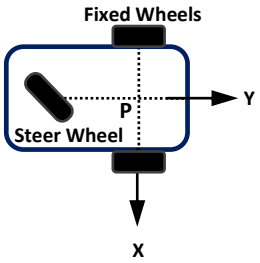


Figure 3.12.Tricycle (Two fixed wheels and one steer wheel).

Type-V

$\delta_m = 1, \delta_s = 2, \delta_M = 3$

These robots have no fixed standard wheels. They have at least two centered orientable wheels $(N_c \geq 2)$. If there are more than two centered orientable wheels, their orientation is such that rank $C_{1c}(\beta_c) = \delta_s = 2$.

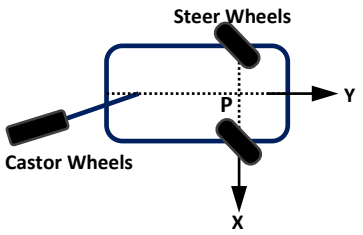


Figure 3.13.Two-steer type (one castor wheel* and two steer wheels)

(* The castor wheels can be replaced by Spherical or Swedish wheels without affecting the maneuverability. Because all the three types of wheels are omnidirectional.)

3.5 Kinematic System Description of Differential Wheel Drive

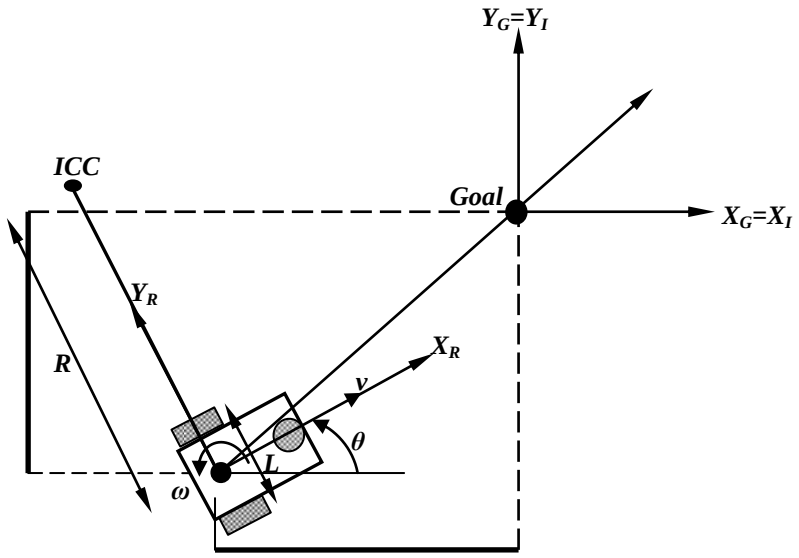


Figure 3.18 Kinematic posture of the differential wheeled robot.

In the above schematic diagram shown in Figure 3.18, the position vector $\begin{bmatrix} x & y & \theta \end{bmatrix}^T$ is represented in the inertial frame, and the goal is at the origin of the inertial frame.

Consider the kinematics of a differential drive robot derived in the inertial frame $\{X_I, \quad Y_I, \quad \theta\}$ is given by;

$$^I\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \cos\theta.v \\ \sin\theta.v \\ \omega \end{bmatrix} \tag{3.24 a}$$

$$= \begin{bmatrix} \frac{1}{2}\cos\theta.(v_r + v_l) \\ \frac{1}{2}\sin\theta.(v_r + v_l) \\ \frac{(v_r - v_l)}{L} \end{bmatrix} \tag{3.24 b}$$

$$= \begin{bmatrix} \frac{1}{2}\cos\theta & \frac{1}{2}\cos\theta \\ \frac{1}{2}\sin\theta & \frac{1}{2}\sin\theta \\ -\frac{1}{L} & \frac{1}{L} \end{bmatrix} \begin{bmatrix} v_l \\ v_r \end{bmatrix} \tag{3.24 c}$$

where, $\dot{x}$ and $\dot{y}$ represent the linear velocities of the inertial frame in the direction of X_I and Y_I respectively. Similarly v_l and v_r represent the linear velocities of the left and right wheel of the robot and L is the distance from the centre of the wheels.

Particular conditions:

1. $v_l(t) = v_r(t)$ i.e straight line trajectory

$$v(t) = v_r(t) = v_l(t)$$

$$\omega(t) = 0 \Rightarrow \dot{\theta}(t) = 0 \Rightarrow \theta(t) = \text{Constant}$$

2. $v_l(t) = -v_r(t)$ i.e circular path with ICC (Instantaneous Center of Curvature) on the midpoint between drive wheels.

$$v(t) = 0$$

$$\omega(t) = \frac{2}{L} v_t(t)$$

3.6 Summary

The following summary has been drawn from this current chapter:

- The description of the kinematic posture of a wheel mobile robot has been discussed, and it is important to compute the robot position in the global reference with respect to the robot's moving frame.
- The kinematic constraints for various types of wheels have been derived and the degree of mobility, steerability and maneuverability are also discussed for different wheeled mobile robots.
- A wheel Jacobian matrix is framed in order to map the motions of individual wheels to the motion of the robot frame. It has been noticed that according to the restriction to the mobility generated by the various constraints, the wheeled mobile robots can be categorized into five types.
- The kinematic description of the differential mobile robot is sufficient to explain the global motion of the robot in the target seeking environment.

CHAPTER-4

4. ANALYSIS OF ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM (ANFIS) FOR MOBILE ROBOT NAVIGATION

Continuous efforts have been given by the researchers to develop new path planning strategies related to mobile robot navigation. Mobile robots must have the ability to navigate autonomously in an unknown environment, to accomplish various tasks, where a variety of obstacles may endanger the safety of the robot. So, the primary aim of the robot is to frame a navigational algorithm, which consists of the planning and implementation of collision free motion in an unknown environment. This current chapter describes the path planning and navigation of mobile robot using Adaptive neuro-fuzzy inference system (ANFIS) in maze environments.

4.1 Introduction

Over the last few years, artificial neural networks and fuzzy inference systems have shown their reputation to build an optimization model. However, there are certain disadvantages of each individual model. For example, the performance of the neural network based models depends upon the systematic training of its adjustable parameters (synaptic weights and bias parameters), the optimal selection of which is a very difficult task. The fuzzy inference system gives good decisions, but much more effort is required to generate the number of fuzzy rules with corresponding membership functions. It also needs an effective training to minimize the system output errors. The Adaptive neuro-fuzzy inference system is a neural network based fuzzy inference system which combines the adaptability capability of the neural network and provide the human like reasoning (in the form of IF-THEN rules) of fuzzy systems. In this present chapter a newly developed ANFIS navigational controller is deployed to solve the navigational problem of a mobile robot in unknown and cluttered terrains.

4.2 Description of Adaptive Neuro-Fuzzy Inference System (ANFIS)

In recent times, intelligent soft computing tools such as fuzzy logic and artificial neural networks have been verified to be efficient and suitable for establishing intelligent systems. A fuzzy inference system depends mainly upon the selection and distribution of the membership functions, and its if-then rules are used to acquire knowledge from human experts to handle imprecise and vague systems. A neural network has the ability to learn from its environment and adapt in an interactive way. For these reasons, a neuro-fuzzy system has been evolved, which takes the advantages of both fuzzy logic and artificial neural network. Takagi and Hayashi [149] have proposed the details of the neuro-fuzzy system. Originally developed by Jang [150] in 1993, the Adaptive neuro-fuzzy inference system has become a well-known hybrid intelligent neuro-fuzzy system and functioning under the Takagi-Sugeno-type fuzzy inference system. ANFIS uses either a combination of least-squares estimation (LSE) method with the back propagation gradient descent method (BPGDM) or only back propagation gradient descent method to adjust premise and consequent parameters.

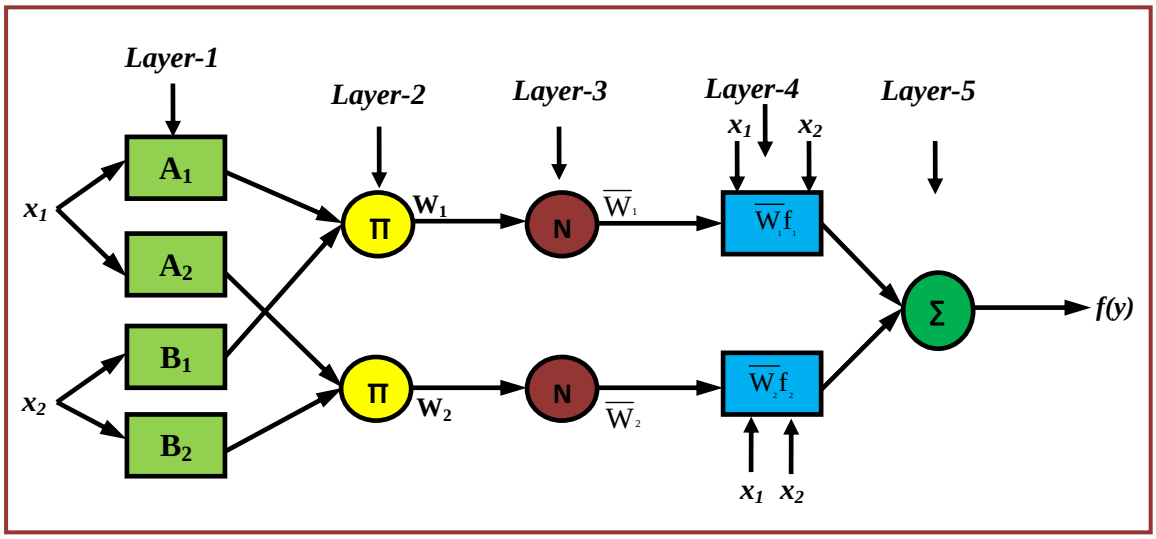


Figure 4.1 An ANFIS with five layers and two inputs.

For the present simple architecture of ANFIS, we assume that the fuzzy inference system under consideration has two inputs x_1 and x_2 and one output y, as shown in Figure 4.1. Suppose the fuzzy inference system contains two IF-THEN rules of the Takagi-Sugeno type [150] as shown in Figure 4.2 ;

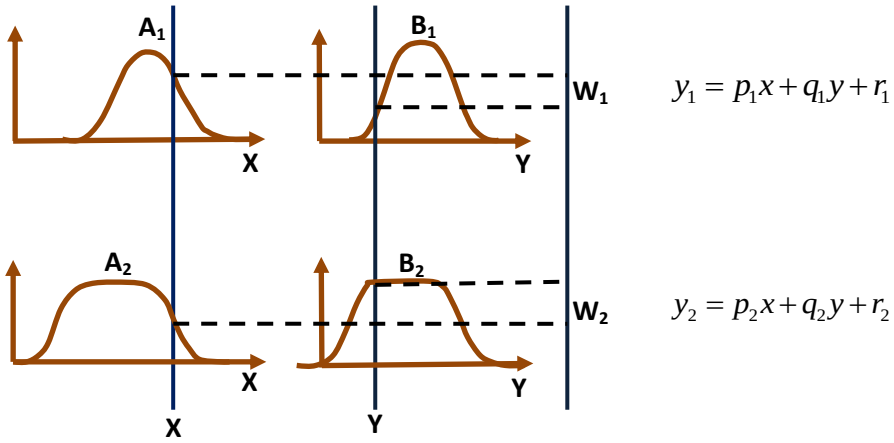


Figure 4.2 Takagi-Sugeno type fuzzy reasoning.

$$\Rightarrow f(y) = \frac{W_1 f_1 + W_2 f_2}{W_1 + W_2} = \bar{W}_1 f_1 + \bar{W}_2 f_2 \tag{4.1}$$

Rule 1: IF x_1 is A_1 AND x_2 is B_1 THEN

$$y_1 = p_1x_1 + q_1x_2 + r_1, \tag{4.2 a}$$

Rule 2: IF x_1 is A_2 AND x_2 is B_2 THEN

$$y_2 = p_2x_1 + q_2x_2 + r_2, \tag{4.2 b}$$

Where, A_1, A_2, B_1, and B_2 are the linguistic description of inputs x_1 and x_2 respectively. $\{p_1, q_1, r_1\}$ and $\{p_2, q_2, r_2\}$ are the consequent parameters, and y_1 and y_2 are the estimated output of the system.

4.3 Control Architecture of the Adaptive Neuro-Fuzzy Inference System (ANFIS) for Mobile Robot Path Planning

The objective of the proposed ANFIS navigational controller is to predict the steering angle of the mobile robot. Here, we consider the adaptive neuro-fuzzy controller (Figure 4.3) under the consideration of four input parameters, i.e. Front obstacle distance (FOD) (X_1), Right obstacle distance (ROD) (X_2), Left obstacle distance (LOD)(X_3) and Heading angle (HA)(X_4).This information is collected by an array of sensors which are mounted around the periphery of the robot.

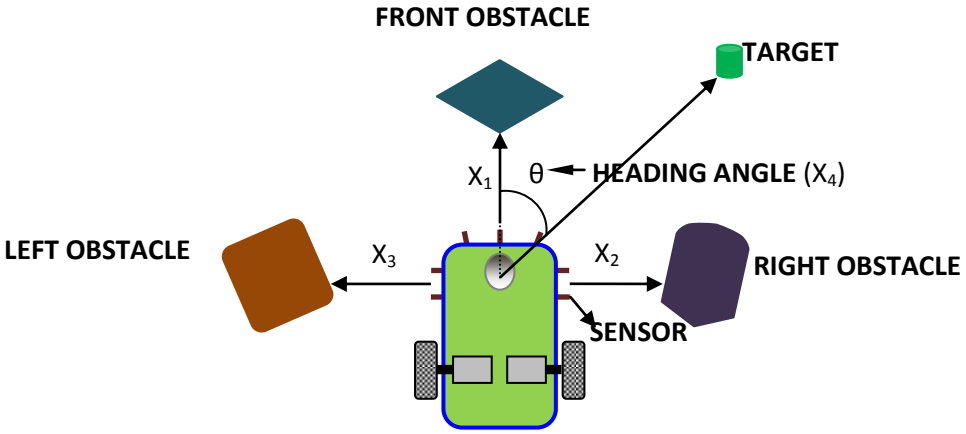


Figure 4.3 Robot initial position in environment.

In this work Bell-shaped type of membership functions are considered and each input variable has three bell membership functions (MF) or linguistic variables A_1(Near), A_2(Medium) and A_3(Far), B_1(Near), B_2(Medium) and B_3(Far), C_1(Near), C_2(Medium) and C_3(Far), D_1(Negative), D_2(Zero) and D_3 respectively; then a Takagi-Sugeno-type fuzzy inference system IF-THEN rules are constructed as follows;

IF(x_1 *is* A_i *and* x_2 *is* B_i *and* x_3 *is* C_i *and* x_4 *is* D_i)

$$THEN\ (steering\ angle\ (f_n) = p_n x_1 + q_n x_2 + r_n x_3 + s_n x_4 + u_n) \quad (4.3)$$

A, *B*, *C* and *D* are the fuzzy membership sets for the input variables x_1, x_2, x_3 and x_4 respectively.

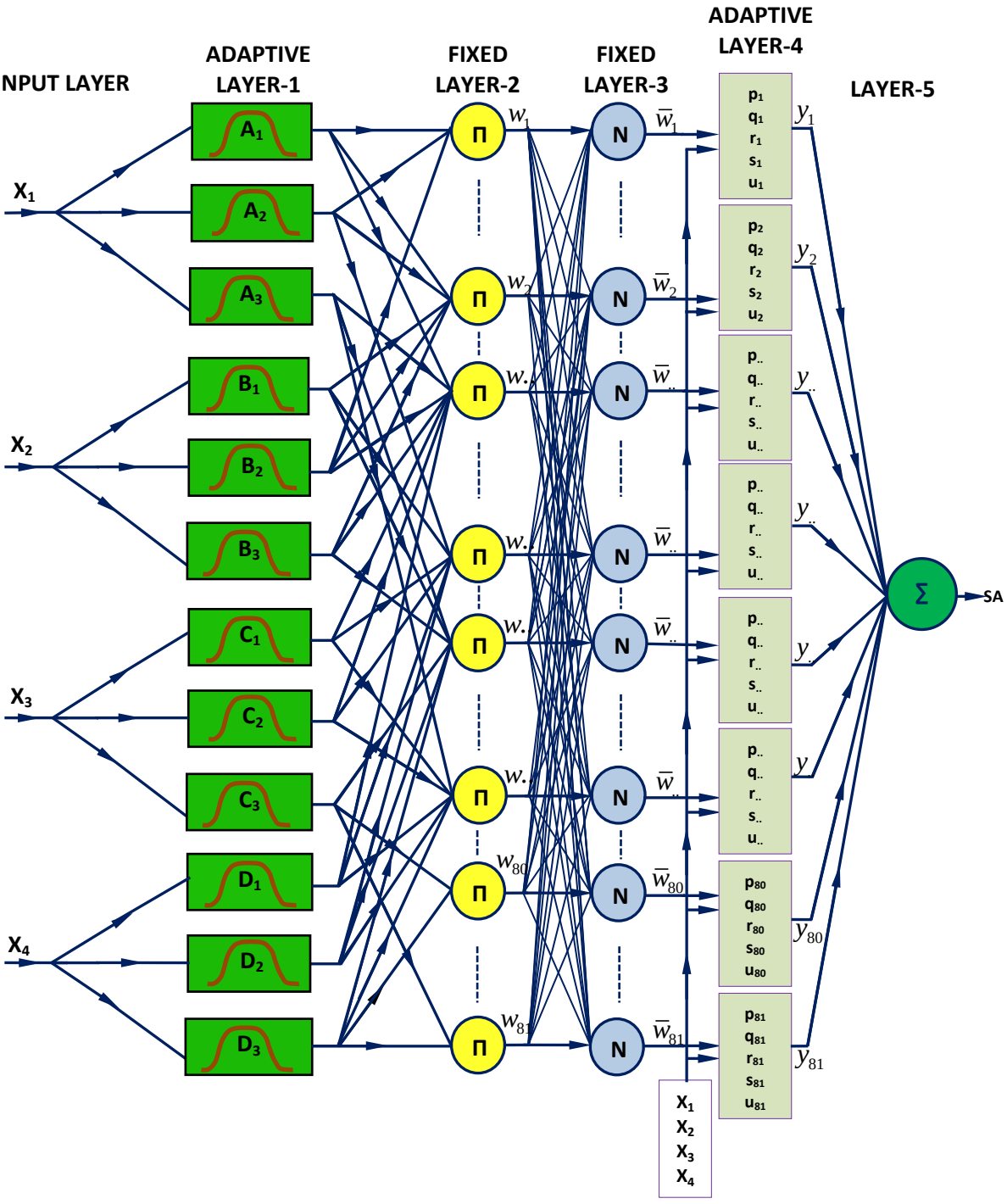


Figure 4.4 An ANFIS structure for current analysis.

Where, $i=1,2,3$ and p_n, q_n, r_n, s_n, and u_n are the linear or consequent parameters of function f_n and changing these parameters we can adjust the output of the ANFIS controller.

The function of each layer in the ANFIS structure (Figure 4.4) is discussed as follows:

Input Layer: In this layer the nodes receive signals from an array of sensors (X_1, X_2, X_3 and X_4) which specify the position of the obstacles and the target. That is defined as follows

$$\left.\begin{aligned} O_{0,FOD} &= X_1(input-1) \\ O_{0,ROD} &= X_2(input-2) \\ O_{0,LOD} &= X_3(input-3) \\ O_{0,\mathrm{HA}} &= X_4(input-4) \end{aligned}\right] \tag{4.4}$$

First Layer: This layer is the adaptive fuzzy layer. The neurons in this layer complete the fuzzification process. Every node in this stage is an adaptive node (square node) and calculates the membership function value in the fuzzy set. For the four inputs, the outputs from the nodes in this layer are presented as

$$\left.\begin{aligned} O_{1,i} &= \mu_{FOD}(X_1) \\ O_{1,i} &= \mu_{ROD}(X_2) \\ O_{1,i} &= \mu_{LOD}(X_3) \\ O_{1,i} &= \mu_{HA}(X_4) \end{aligned}\right] \tag{4.5}$$

Here $O_{1,i}$ is the Bell-shaped membership grade of a fuzzy set S (A_i, B_i, C_i and D_i) and it specifies the degree to which the given inputs (X_1, X_2, X_3 and X_4) satisfy the quantifier S. The membership functions for A, B, C and D are the bell shape functions and which is shown in Figure 4.5.

$$\mu_{A_i}(x_1) = \frac{1}{1+\left|\left(\frac{x_1-c_i}{a_i}\right)^2\right|^b} \tag{4.5 i}$$

$$\mu_{B_i}(x_2) = \frac{1}{1+\left|\left(\frac{x_2-c_i}{a_i}\right)^2\right|^b} \tag{4.5 ii}$$

$$\mu_{C_i}(x_3) = \frac{1}{1+\left|\left(\frac{x_3 - c_i}{a_i}\right)^2\right|^b} \qquad \text{(4.5 iii)}$$

$$\mu_{D_i}(x_4) = \frac{1}{1+\left|\left(\frac{x_4 - c_i}{a_i}\right)^2\right|^b} \qquad \text{(4.5 iv)}$$

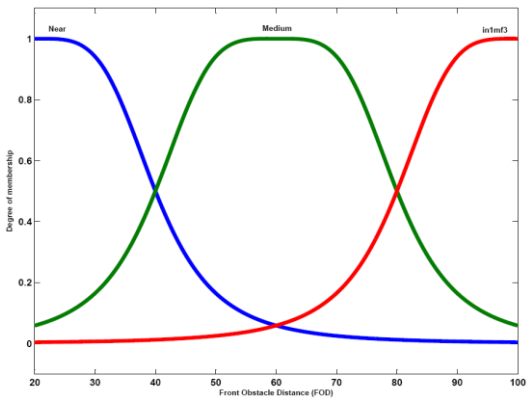


Figure 4.5 a

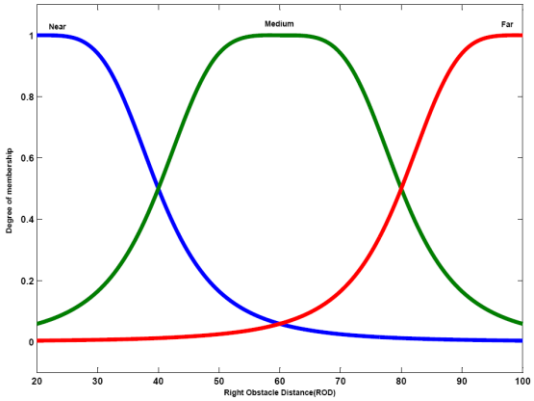


Figure 4.5 b

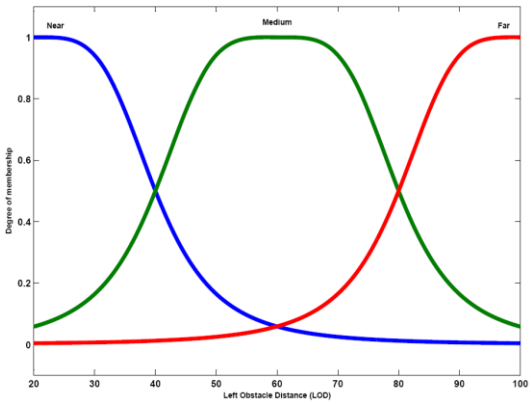


Figure 4.5 c

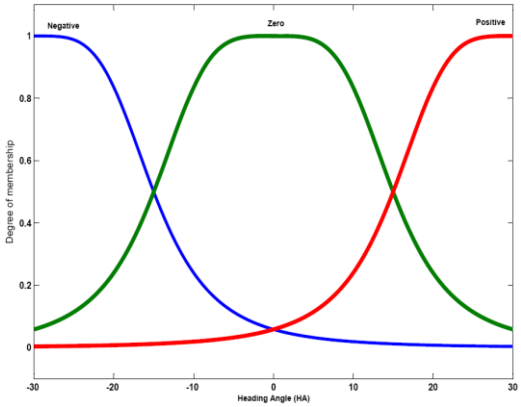


Figure 4.5 d

Figure 4.5 (a-d) Membership functions for input parameters.

a_i, b_i and c_i are parameters that control the centre, width and slope of the Bell-shaped function of node 'i' respectively. They are also known as the antecedent or premise parameters. Changing these parameters will give the various contours of a bell shaped function as required in accordance with the data set for the problem defined. The fuzzy membership function can be chosen in any form, such as triangular, Gaussian,

trapezoidal, bell-shaped, etc. Usually, we choose the bell-shaped one with the maximum value equal to 1 and minimum value equal to 0.

Second Layer: It is also known as the fuzzy rule layer. Every node in this layer is a fixed node (circular) and labeled as π_n. Every node in this stage corresponds to a single Sugeno-Takagi fuzzy IF-THEN rule. Each rule point receives inputs from the respective points of layer-2 and calculates the firing strength of each fuzzy rule. The output from each node is the product of all incoming signals.

$$O_{2,n} = W_n = \mu_{Ai}(X_1).\mu_{B_i}(X_2).\mu_{C_i}(X_3).\mu_{D_i}(X_4) \tag{4.6}$$

Where W_n represents the firing strength or the truth value, of the n^{th} rule and *n=1, 2, 3…81* is the number of Sugeno-Takagi fuzzy rules.

Third Layer: It is the normalization layer. Every node in this layer is a fixed node (circular) and labeled as N_n. Each point in this layer receives inputs from all points in the adaptive fuzzy rule layer and calculates the normalized firing strength of a given rule. The normalized firing strength of the n^{th} point of the n^{th} rule's firing strength to the sum of all rules' firing strength.

$$O_{3,n} = \overline{W_n} = \frac{W_n}{\sum_{n=1}^{81} W_n} \tag{4.7}$$

$\bar{w}_n$ represents the normalized firing strength of a given rule.

Fourth layer: Every node in this layer is an adaptive node (square node). Each node in this layer is connected to the corresponding normalization node, and also receives the initial inputs X_1, X_2, X_3 and X_4. A defuzzification node determines the weighted consequent value of a given rule presented as,

$$O_{4,n} = \overline{W}_n f_n = \overline{W}_n \left[p_n(X_1) + q_n(X_2) + r_n(X_3) + s_n(X_4) + u_n \right] \tag{4.8}$$

Where, $\overline{W}_n$ is a normalized firing strength from layer-3 and p_n, q_n, r_n, s_n and u_n are the parameters set by this node. These parameters are also called consequent parameters.

Fifth layer: It is represented by a single summation node (circular node). This single point is a fixed point and is labeled as $\sum$. This point determines the sum of the outputs of all defuzzification points and gives the overall model output, that is the steering angle.

$$O_{5,n} = \sum_{n=1}^{81} \overline{W_n} f_n = \frac{\sum_{n=1}^{81} W_n f_n}{\sum_{n=1}^{81} W_n} \tag{4.9}$$

Figure 4.4 shows the ANFIS navigational controller with its inputs and output signals. The sign convention for ANFIS in terms of heading angle (HA) with respect to obstacle is shown in Figure 4.6. The proposed ANFIS controller has been trained with about 800 patterns of training behaviors; some of them are depicted in Figure 4.7. For example, Figure 4.7 (c) shows when a robot is moving towards an obstacle, another obstacle is being on its left hand side, and no target is sensed by the robot. The proposed ANFIS navigator is trained to steer towards the right hand side. The Table-4.1 depicts some examples of training patterns of the proposed path planner. The parameters setting for training variables of ANFIS model is shown in Table-4.2. In this navigational model mainly three reactive behaviours are used by the ANFIS and demonstrated in Table-4.3. The reactive behaviors communicate with the sensory data and steer the robot accordingly for successful navigation.

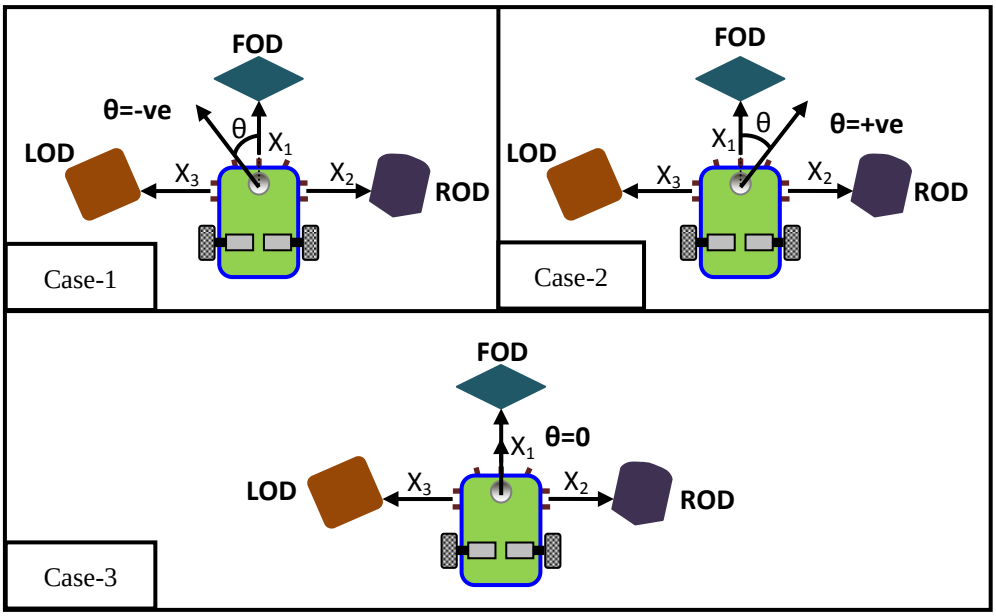


Figure 4.6 Sign convention used in ANFIS in terms of heading angle (HA) with respect to obstacle position.

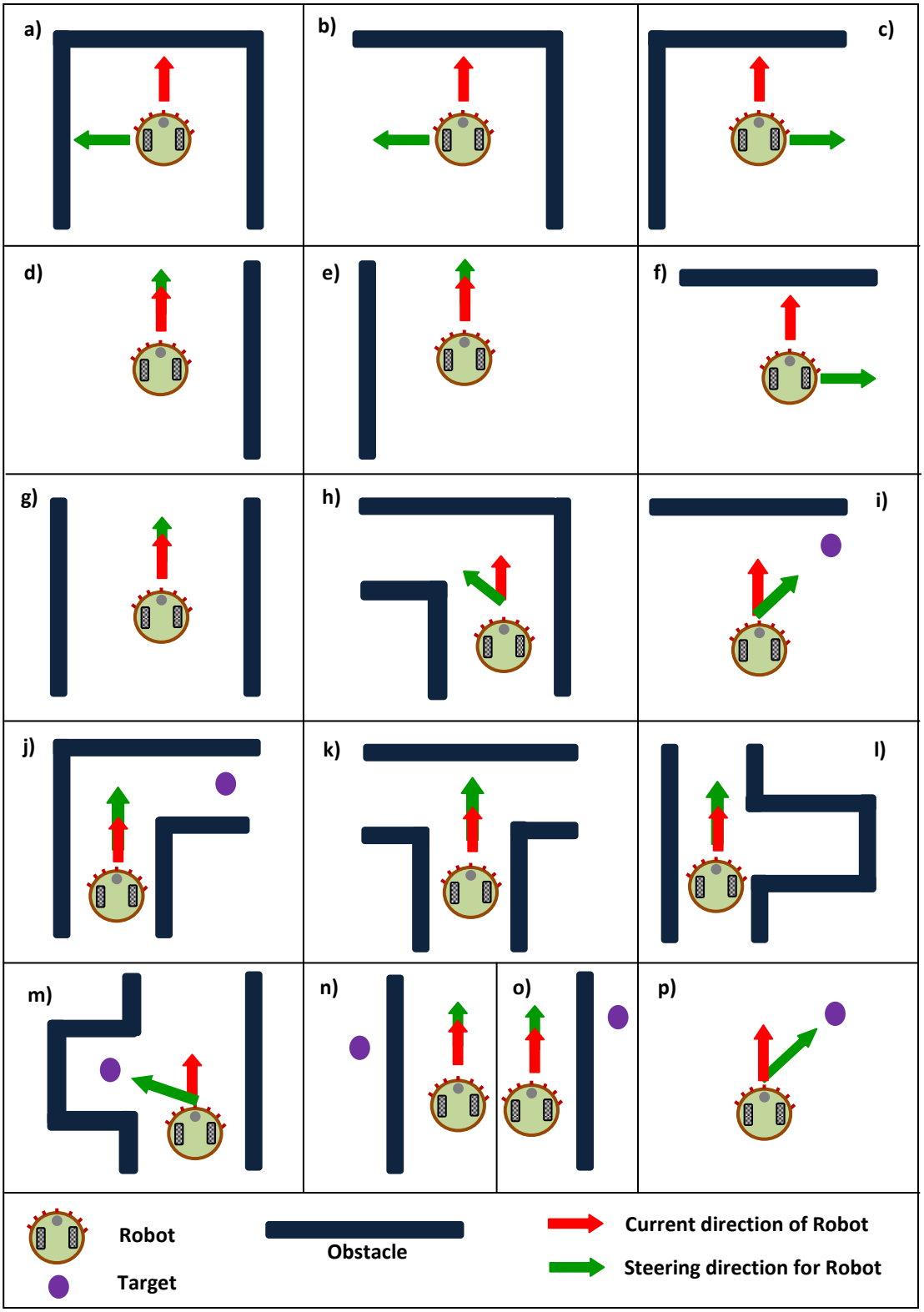


Figure 4.7 Examples of various reactive behaviors for ANFIS navigational controller.

Table-4.1 Examples of training pattern for current navigational controller.

SL.No.	ACTION	LOD	FOD	ROD	HA	SA
1	OA	N	N	N	NO GOAL CONSIDERED	POSITIVE
2	OA	F	N	N	NO GOAL CONSIDERED	POSITIVE
3	OA	N	N	F	NO GOAL CONSIDERED	NEGATIVE
4	OA	F	F	N	NO GOAL CONSIDERED	ZERO
5	OA	N	F	F	NO GOAL CONSIDERED	ZERO
6	OA	F	N	F	NO GOAL CONSIDERED	NEGATIVE
7	OA	N	F	N	NO GOAL CONSIDERED	ZERO
8	OA	N	F	N	NO GOAL CONSIDERED	POSITIVE
9	TS	F	N	F	NEGATIVE	NEGATIVE
10	TS	N	F	N	NEGATIVE	ZERO
11	TS	N	F	N	POSITIVE	ZERO
12	TS	F	F	N	NEGATIVE	ZERO
13	TS	F	F	F	POSITIVE	POSITIVE
14	OA	N	N	M	NO GOAL CONSIDERED	NEGATIVE
15	TS	N	F	M	POSITIVE	ZERO
16	TS	N	M	F	POSITIVE	POSITIVE
17	TS	N	M	N	NEGATIVE	ZERO
18	TS	F	N	M	NEGATIVE	NEGATIVE
19	TS	M	F	N	POSITIVE	POSITIVE
20	TS	F	M	N	POSITIVE	POSITIVE
21	TS	F	M	N	NEGATIVE	ZERO

[OA: Obstacle avoidance, M: Medium, TS: Target seeking, HA: Heading Angle, SA: Steering Angle, Positive: left turn, Negative: right turn, zero: straight F: Far, N: Near, M: Medium]

Table 4.2 Parameters setting for training variables.

Left Obstacle Distance **(LOD)**	2cm to 10cm
Front Obstacle Distance **(FOD)**	
Right Obstacle Distance **(ROD)**	
Heading Angle **(HA)**	-30° to 30°
Steering Angle **(SA)**	-90° to 90°

Table 4.3 Brief description of various reactive behaviours adopted by the Robot.

Types of reactive behaviours	**Definition of the reactive behaviours**	**Robot in action**
Obstacle Avoidance (OA)	When a mobile robot is sensing any obstacle in front, left or right side. This behaviour is required to avoid hitting with an obstacle.	The mobile robot set the steering angle accordingly by adjusting the speed.
Target Seeking (TS)	When a robot is directly sensing a target, and no obstacles are present around the robot, its main behaviour is to seek for the target. This behaviour is required to trace the target.	The mobile robot adjusts its steering angle and moves quickly towards the target.
Barrier Following (BF)	When a mobile robot is searching a target, and it meets an obstacle (long Barrier) in front of it or having a barrier to the left or right hand side. In this condition, the robot has to follow the barrier to get the target successfully.	The mobile robot controls its speed and fixes a constant heading angle towards the barrier. The robot moves parallely with the barrier to trace the target successfully.

4.4 Demonstrations of the ANFIS Navigational Controller

The proposed navigation technique has been deployed in both simulation as well as real time experimental scenario in different environments. The simulation results are carried out with the help of MATLAB [228]. The size dimension of simulation platform in MATLAB is considered as 30x30 units and each unit is equal to 2mm. A real time experimental environment has been set-up in the laboratory containing static obstacles as well as a static target to validate the simulation results. Finally, the effectiveness of ANFIS navigational controller is analyzed, discussed and compared for the path optimization problem.

4.4.1 Simulation Results

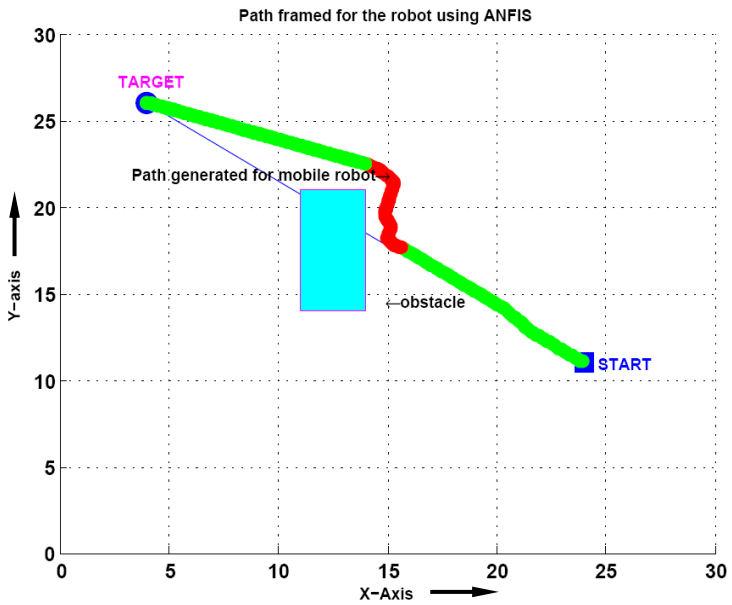


Figure 4.8 Obstacle avoidance behaviour shown by the robot using ANFIS.

The navigation problem for the mobile robots is generally decomposed into three main behaviors (sub-problems); target seeking, obstacle avoidance and barrier following.

The analysis of simulation results of the proposed navigation strategy are shown in Figure 4.8 to Figure 4.13. The average of 20 runs has been taken into consideration for each simulation and the best results have been shown in the graphical mode. When the sensors detect obstacles in the front, this means that any sensor readings are less than the threshold value , then the obstacle avoidance behaviour is activated for the mobile robot

to avoid collisions with obstacles as shown in Figure 4.8. The obstacle avoidance behaviour has given high priority compared to the other main behaviours.

The barrier following behaviour is usually introduced when a robot detects a barrier in front of it and the target is presented opposite site of the barrier as shown in Figure 4.9. In case of a U-shaped obstacle or dead end situation, the goal is presented to the back side of the obstacle. In this condition, using barrier following behaviour the robot is moving along the barrier to escape the U-shaped obstacle and to reach the goal successfully. When the robot doesn't find any obstacles around it, the main behaviour is to go for target seeking. The proposed ANFIS path planner mainly controls the robot motion direction and helps approaching towards to the goal if there is no obstacle present around the robot as shown in Figure 4.10. The robot must have multiple reactive behaviors, such as avoiding obstacles, following barriers, and seeking for the target and so on, according to the acquired sensor informations, until it reaches a definite target. In the proposed navigation strategy, the reactive behaviours are designed and trained by ANFIS.

The navigation path of mobile robots using ANFIS methodology and escaping from the narrow passage and the dead end are shown in Figures 4.11, and 4.12 respectively. Figure 4.13 shows a single robot navigating inside a maze environment.

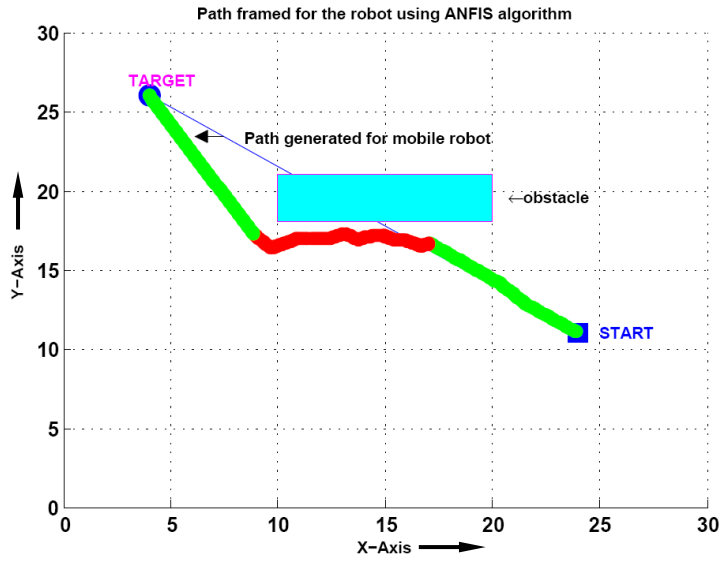


Figure 4.9 Barrier following behaviour shown by the robot using ANFIS.

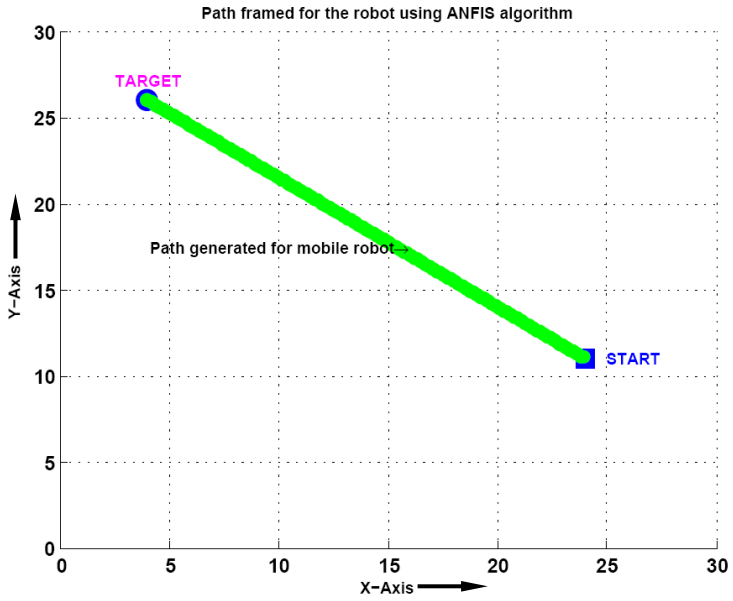


Figure 4.10 Target seeking behaviour shown by the robot using ANFIS.

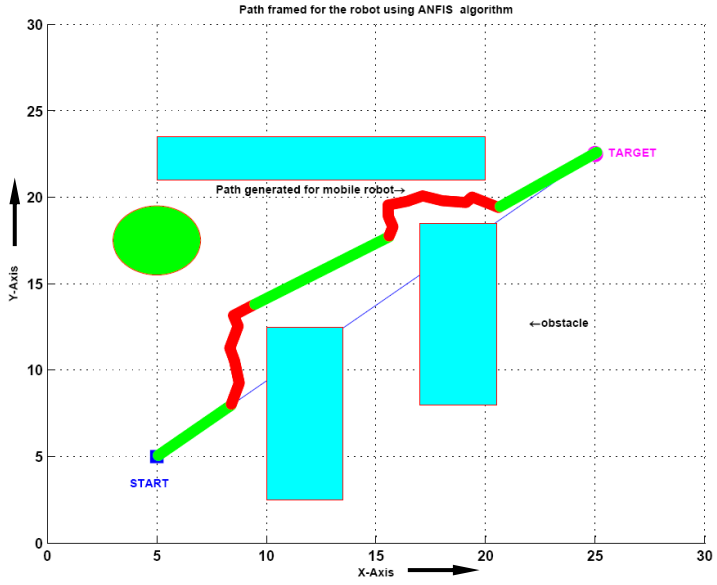


Figure 4.11 Single robot escaping from a narrow passage using ANFIS.

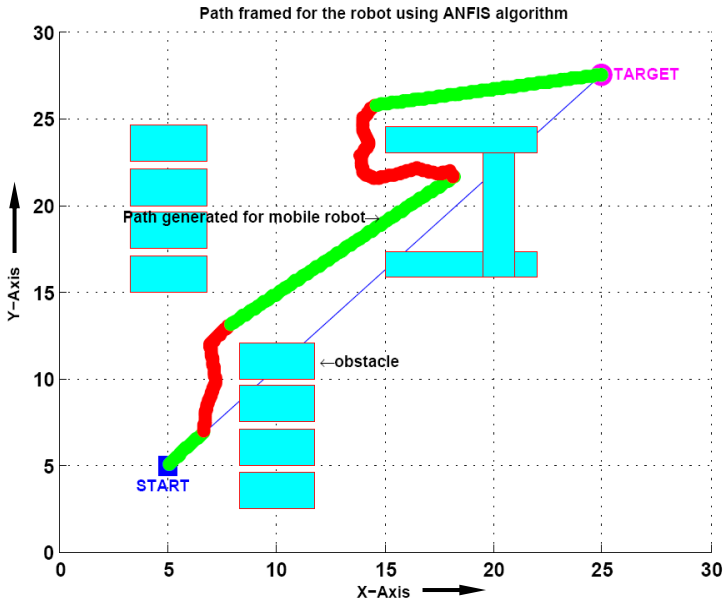


Figure 4.12 Single robot escaping from a trap situation to reach target using ANFIS.

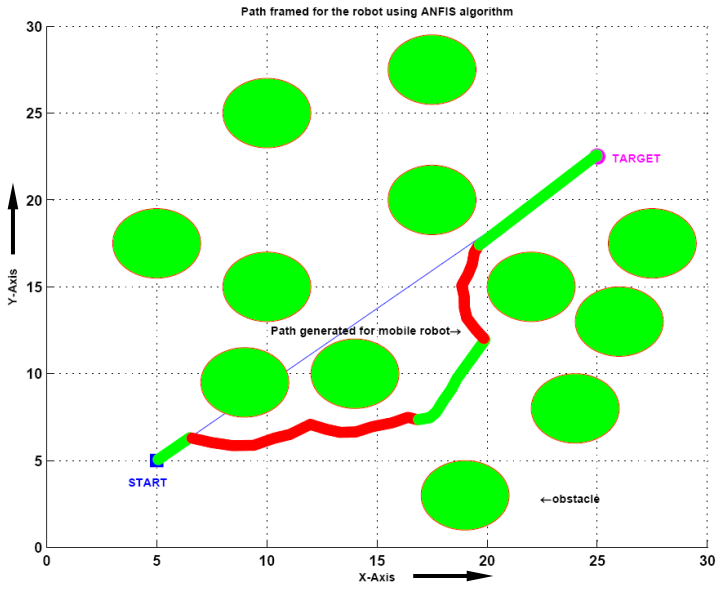


Figure 4.13 Single robot navigating inside a maze environment to reach target using ANFIS.

4.4.2 Experimental Validation with Real Mobile Robot

In order to demonstrate the effectiveness of the ANFIS navigational controller in a static environment, the simulation results are verified with real time experimental results. We have implemented the ANFIS navigation algorithm on an actual mobile robot. The mobile robot has the dimension (30cmx25cmx10cm) with two fixed standard wheels, and has differential wheel drives developed in our laboratory. The real time experiments are carried out on a platform of size 225cmx175cm.The mobile robot is equipped with a ring of three infrared sensors, three ultrasonic sensors and wheel encoders. A brief description of the mobile robot developed in the laboratory is given in Appendix-A. The mobile robot is tested through extensive experiments in various environments with a fixed start point and target position. Real time experiments are conducted on the robot to demonstrate the effectiveness of the proposed ANFIS navigation system. It can be clearly seen that the mobile robot always tries to go towards the target position if there are no close obstacles. If it faces any obstacle, then slows down its motion and adjusts steering direction as per the ANFIS strategy to perform the obstacle avoidance behaviour. Figure 4.14(a-f) shows the experimental results obtained for the similar simulation environment depicted in Figure 4.11. Similarly, Figure 4.15(a-f) shows experimental results carried out in a similar simulation result as shown in Figure 4.13. It is clearly observed that the proposed algorithm endows the mobile robot with ability to response quickly the static environment. Table-4.4 and Table-4.5 show a comparison between the path covers and time taken by the robot in simulation and practical modes of obstacle avoidance and target seeking. During practical tests, it has been observed that the path covers and time taken by the robot to reach the target are more than the simulation path lengths and time taken. This happens due to of various errors (e.g. obstacle/ target tracking error, presence of friction in rotating elements, signal transmission error in data-cable, slippage between floor and wheels, friction between supported point and floor etc.). Figures given in the table are the averages of five practical tests on each environmental scenario, and best results are shown in the experimental Figures 4.14 and 4.15.

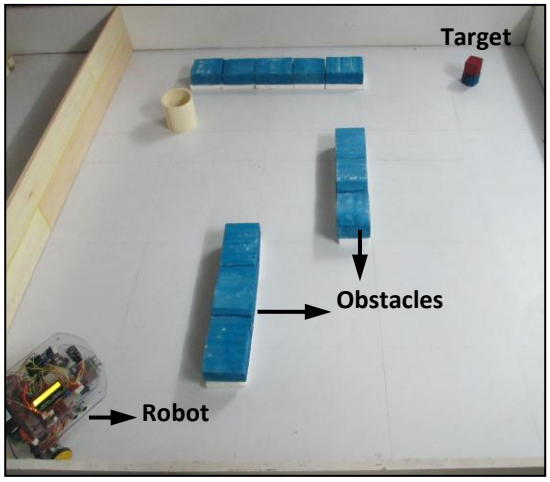


Figure 4.14 (a)

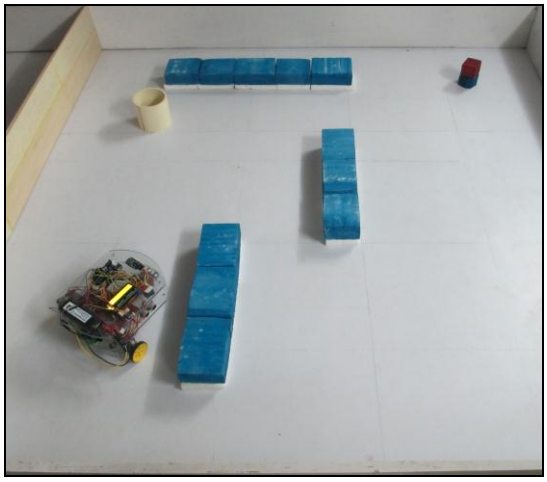


Figure4.14 (b)

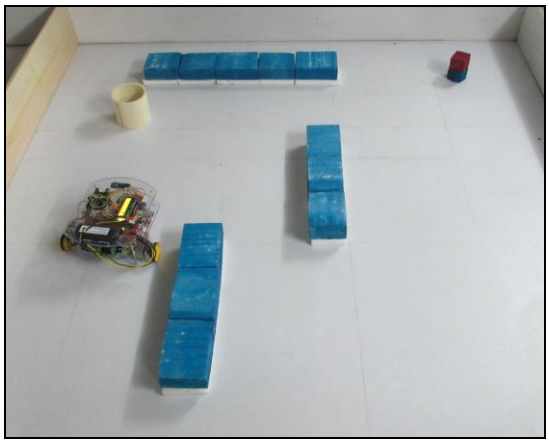


Figure 4.14 (c)

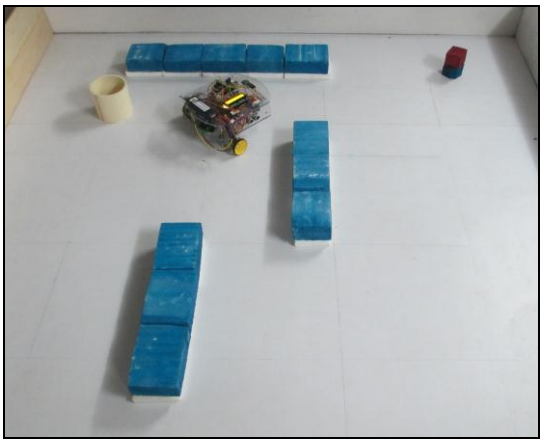


Figure4.14 (d)

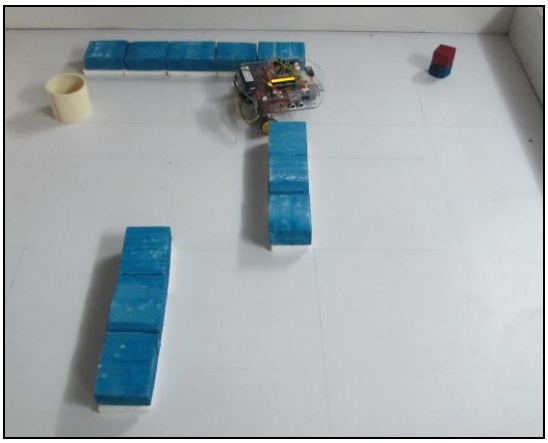


Figure 4.14 (a)

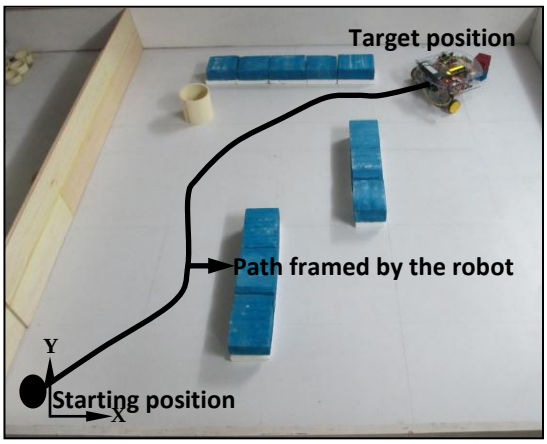


Figure4.14 (b)

Figure 4.14 (a-f) Experimental results for navigation of mobile robot in the environment shown in Figure 4.11.

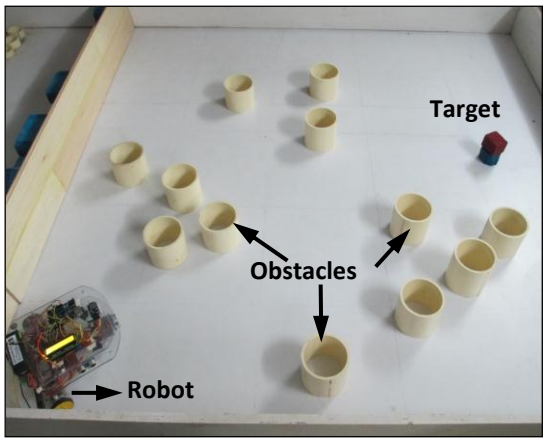


Figure 4.15 (a)

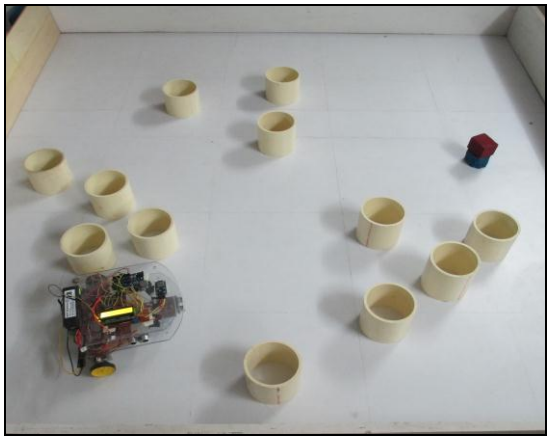


Figure4.15 (b)

Figure 4.15 (c)

Figure4.15 (d)

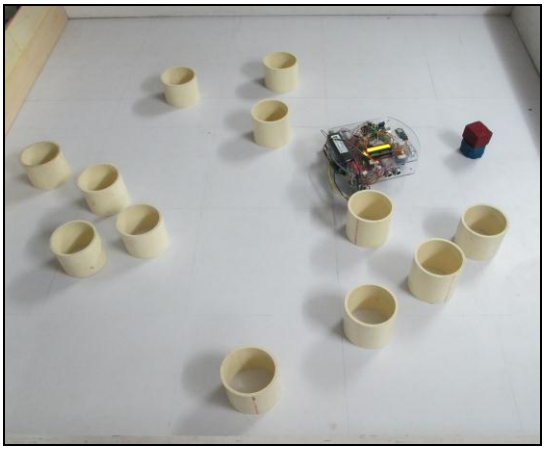


Figure 4.15 (e)

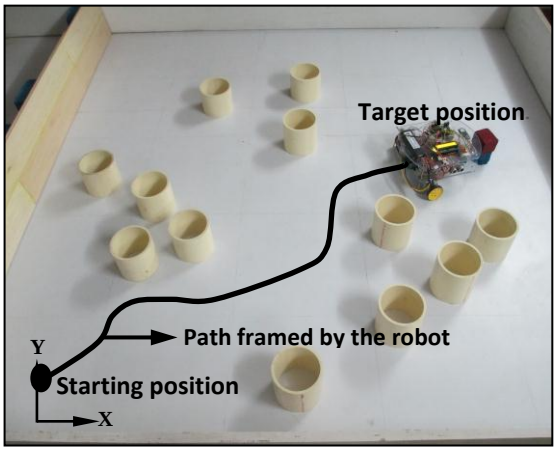


Figure 4.15 (f)

Figure 4.15 (a-f) Experimental results for navigation of mobile robot in the environment shown in Figure 4.13.

Table 4.4 Path length covered by the robot in simulation and experimental test to reach the target.

Sl No.	Path length in simulation (‘cm’)	Path length in real time test (in ‘cm’)	% of error
Scenario-1	165.08 (Fig. 4.11)	175.29 (Fig. 4.14 f)	6.18
Scenario-2	159.12 (Fig. 4.13)	169.19 (Fig. 4.15 f)	6.33

Table 4.5 Time taken by the robot in simulation and experimental test to reach the target.

Sl No.	Time taken in simulation (in ‘sec’)	Time taken in real time test (in ‘sec’)	% of error
Scenario-1	16.90(Fig. 4.11)	17.96(Fig. 4.14 f)	6.27
Scenario-2	15.15(Fig. 4.13)	16.02(Fig. 4.15 f)	5.74

4.4.3 Comparison of the Developed ANFIS Navigational Controller with other Models

In this section a comparative study has been carried out for the developed ANFIS controller in simulation mode with Obe et al. [84] and Zhang et al. [148] for the single robot. We have replicated the environment as demonstrated by the authors. In this replicated environments, the robot moves from the starting position to the target position with the help of ANFIS controller. After successful completion of the navigational task from starting point to the target point, it has been compared with the path as illustrated by the authors. The effectiveness of the comparative study is mainly measured on the basis of the path length. The comparative simulation results are presented in Figures 4.16 and 4.17 respectively. The size dimension of simulation platforms are considered as no. of units and each unit is in millimeter (mm).

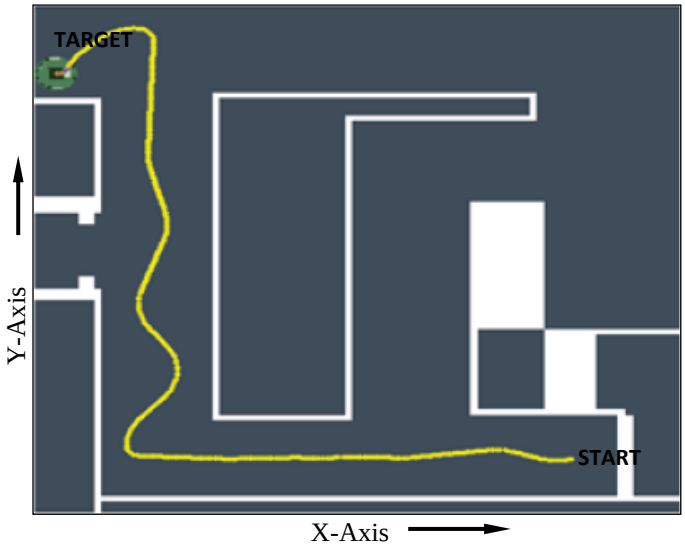


Figure 4.16 (a) Navigation path framed for single mobile robot to reach target by Obe et al.[84].

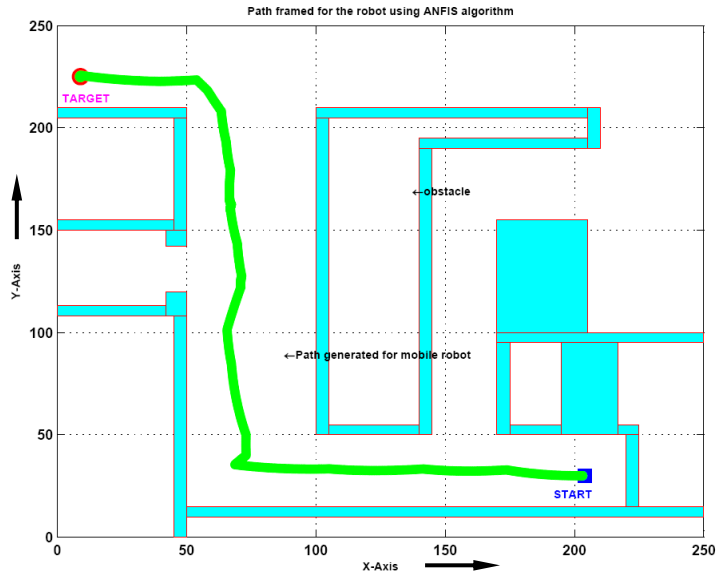


Figure 4.16 (b) Navigation path framed for single mobile robot to reach target using developed ANFIS technique.

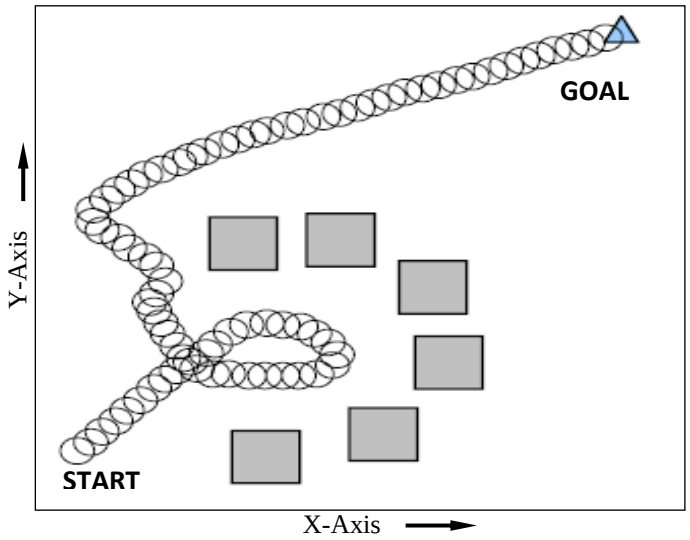


Figure 4.17 (a) Path framed for single mobile robot to reach target by Zhang et al. [148].

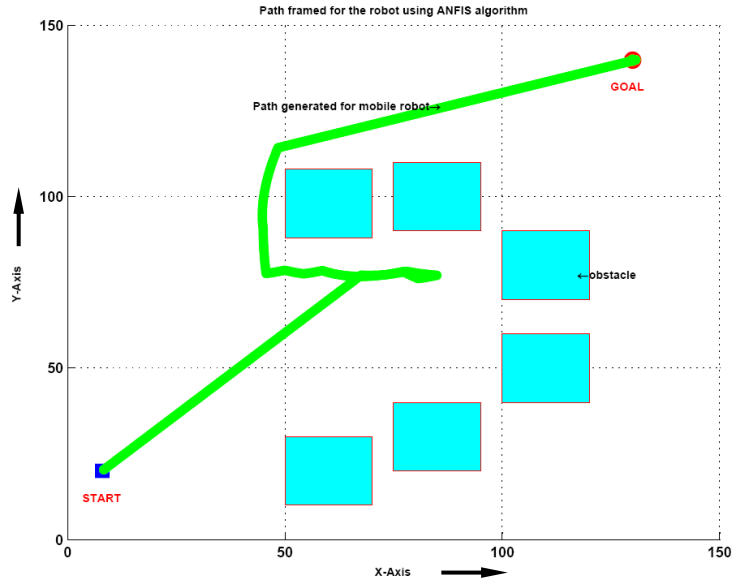


Figure 4.17(b) Path developed for single mobile robot to reach target using current investigation.

In the first comparison study both the path planners are employed in a complex environment. In this condition, the robot cannot sense the target directly due to the presence of the long barrier between them. From the simulation results shown in Figure 4.16 (a), it can be observed that the navigational controller developed by Obe et al. [84] has a greater robot steering angle and prescribed a zigzag like the path for the similar environment. This difficulty has been taken care in the current investigation and presented in Figure 4.16 (b).

In the second case both the motion controllers are deployed in a situation like a dead end problem. The path proposed by Zang et al. [148] as shown in Figure 4.17 (a), the robot got trapped in the U-shaped scenario, and then it has relieved from the trap condition by taking a long loop and reaches the target. The path framed by the robot using the current method is shown in Figure 4.17 (b). Here the robot is initially heading towards the dead end and due to the additional sensor extracted information provided by the proposed path planner, the robot can capable of handling the situation. Finally, it reaches the target by maintaining a safe distance from the obstacle. The performance of the comparison study is mainly measured in terms of path length and tabulated in Table 4.6.

Table 4.6 Comparison of simulation results in terms of path length.

Sl No.	**Environment types**	**Path length from current system (in 'cm')**	**Path length of reference model (in 'cm')**	**% of deviation**
1	Complex environment with different obstacles Figures 4.16(a) and (b)	12.3	14.1	12.77
2	U-Shaped environment Figures 4.17(a) and (b)	15.3	17.2	11.05

4.5 Architecture of Multiple Adaptive Neuro-Fuzzy Inference System (MANFIS) for Mobile Robot Path Planning

A new navigation model based on the neuro-fuzzy technique has been proposed in this section of the thesis. There are four inputs (Front obstacle distance (FOD), Right obstacle distance (ROD), Left obstacle distance (LOD) and Heading angle (HA)) and two outputs (Left wheel velocity (LWV) and Right wheel velocity (RWV)). This model consists of five layers, i.e. fuzzification layer, product layer, normalized layer, rule layer, and output layer as shown in Figure 4.18, which are already being discussed in the ANFIS navigational controller.

The objective of the MANFIS path planner is to predict the left and right wheel velocity of the mobile robot. At first, we develop an adaptive fuzzy controller with four input parameters i.e Front obstacle distance (FOD) (X_1), Right obstacle distance (ROD) (X_2), Left obstacle distance (LOD) (X_3) and Heading angle (HA) (X_4) and two output parameters i.e. Left wheel velocity (LWV) f_{LWV} and Right wheel velocity (RWV) f_{RWV}. Here we consider each input variable have three bell membership functions (MF) or linguistic variables A_1(Near), A_2(Medium) and A_3(Far), B_1(Near), B_2(Medium) and B_3(Far), C_1(Near), C_2(Medium) and C_3(Far), D_1(Negative), D_2(Zero) and D_3 respectively;

The Takagi-Sugeno-type fuzzy inference system IF-THEN rules are defined for left wheel velocity and right wheel velocity as follows:

IF(x_1 is A_i and x_2 is B_i and x_3 is C_i and x_4 is D_i)

THEN (wheel velocity)(f_n) $= p_n x_1 + q_n x_2 + r_n x_3 + s_n x_4 + u_n$ (4.10)

A, B, C and *D* are the fuzzy membership sets of the input variables x_1, x_2, x_3 and x_4 respectively.

Where, $i=1,2,3$ and p_n, q_n, r_n, s_n, and u_n are the linear or consequent parameters of function f_n and changing these parameters we can control the output of the ANFIS controller.

Here, in the fifth layer first node is the first output for left wheel velocity (LWV) and the second node is the second output for right wheel velocity (RWV). From the wheel velocity the steering angle of the mobile robot can be computed by the following equation,

$$\text{Steering angle } \left(\theta^{'}\right) = \frac{f_{LWV} - f_{RWV}}{w} \quad (4.11)$$

where f_{LWV} and f_{RWV} are the left and right wheel velocities and w is the wheel base or distance between the two wheels. If $f_{LWV} > f_{RWV}$, the steering angle is in a clockwise direction and if $f_{LWV} < f_{RWV}$, the steering angle is in a counterclockwise direction and if $f_{LWV} = f_{RWV,}$ then there will be straight motion. The computed steering angle is used for driving the robot. Examples of training pattern for proposed MANFIS controller are given in Table-4.7. The parametric values of training data are presented in Table-4.8.

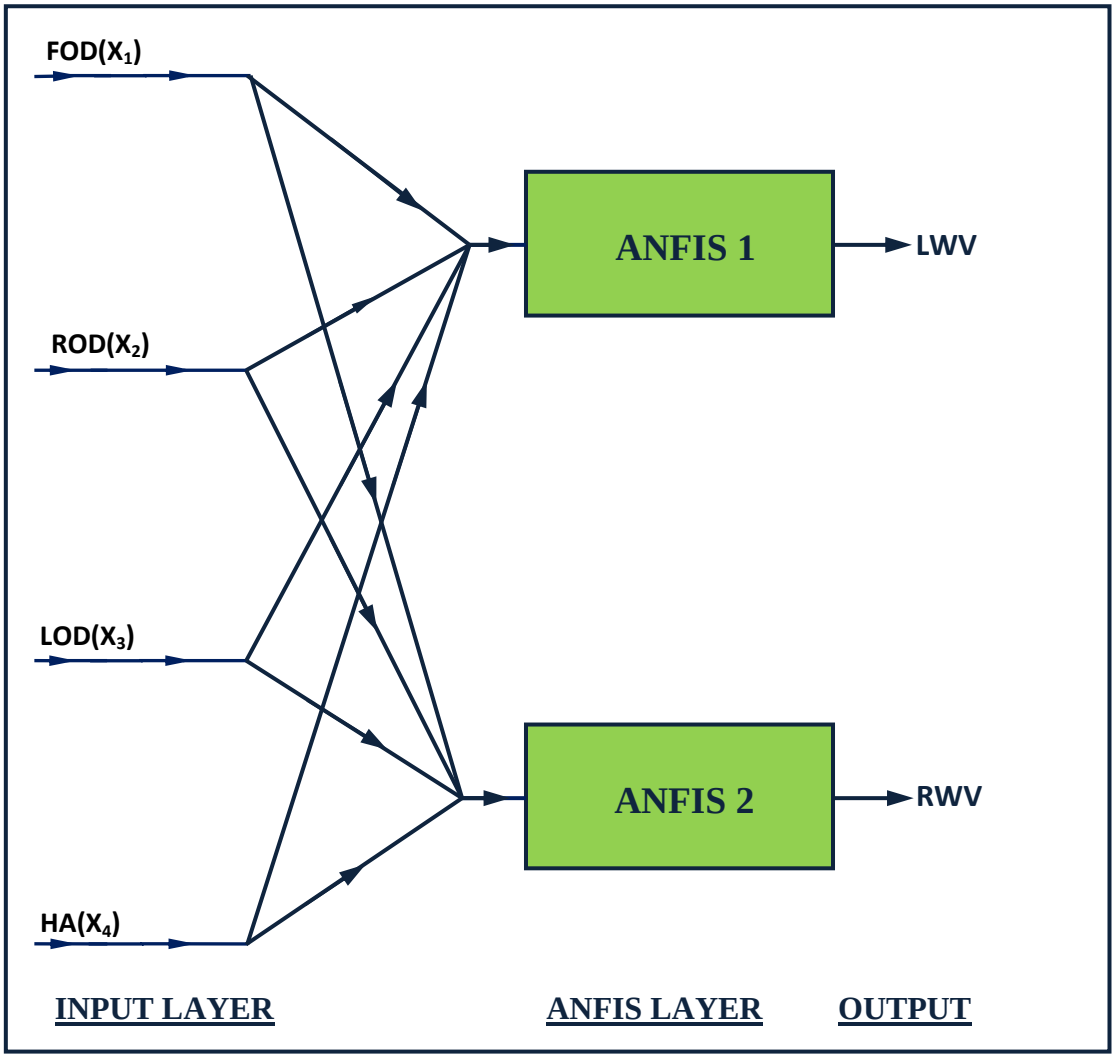


Figure 4.18 Proposed MANFIS navigational controller for current investigation.

Table-4.7 Examples of training pattern for MANFIS navigational controller.

SL. No.	BEHAVIOUR	LOD	FOD	ROD	HA	LWV	RWV
1	OA	Near	Near	Near	Zero	Slow	Slow
2	OA	Near	Near	Med	Negative	Slow	Med
3	OA	Near	Near	Far	Zero	Fast	Slow
4	OA	Near	Med	Near	Negative	Slow	Slow
5	OA	Near	Med	Med	Positive	Fast	Med
6	OA	Near	Med	Far	Positive	Med	Slow
7	OA	Near	Far	Near	Zero	Slow	Slow
8	OA	Near	Far	Med	Negative	Fast	Med
9	OA	Near	Far	Far	Positive	Fast	Slow
10	OA	Med	Near	Near	Negative	Med	Fast
11	BF	Near	Med	Med	Negative	Slow	Slow
12	BF	Far	Near	Med	Zero	Med	Slow
13	TS	Far	Far	Far	Zero	Fast	Fast
14	TS	Far	Far	Med	Positive	Slow	Med
15	TS	Far	Med	Far	Negative	Med	Fast
16	TS	Med	Far	Far	Zero	Fast	Fast
17	TS	Far	Far	Far	Positive	Fast	Med
18	TS	Far	Far	Med	Negative	Med	Fast

[OA: Obstacle avoidance, BF: Barrier following, Med: Medium, TS: Target seeking, HA: Heading Angle, LWV: Left wheel velocity, RWV: Right wheel velocity]

Table-4.8 Parameters setting for training variables.

Left Obstacle Distance **(LOD)**	2cm to 10cm
Front Obstacle Distance **(FOD)**	
Right Obstacle Distance **(ROD)**	
Heading Angle **(HA)**	-30° to 30°
Left Wheel Velocity **(LWV)**	0 to 0.3 m/s
Right Wheel Velocity **(RWV)**	0 to 0.3 m/s

4.6 Demonstrations of the MANFIS Path Controller

The proposed navigation model has been tested in simulation with different environmental scenarios. The experimental verification has been carried out with a real robot to verify the efficacy of the navigation model. Finally, the results obtained from the MANFIS path planner have been compared with other intelligent techniques.

4.6.1 Simulation Results and Discussion

In the following simulation investigations shown in Figures 4.19 to 4.22, the mobile robot's task is to travel from the source position to the target position in an unknown environment using MANFIS navigational algorithm. If there is no obstacle present in the target path, that means the path towards the goal is clear, then the robot moves at its maximum velocity to reach the goal. The speed of the robot is reduced using MANFIS strategy, only where the robot is heading towards the obstacle. The barrier following behaviour is shown by the robot in Figure 4.19. Figure 4.20 shows, how the robot reaches the target by avoiding obstacles. The simulation results presented in Figures 4.21 and 4.22, show the effectiveness of the new hybrid navigation model with dealing with various scenarios such as corridor following and maze like environments.

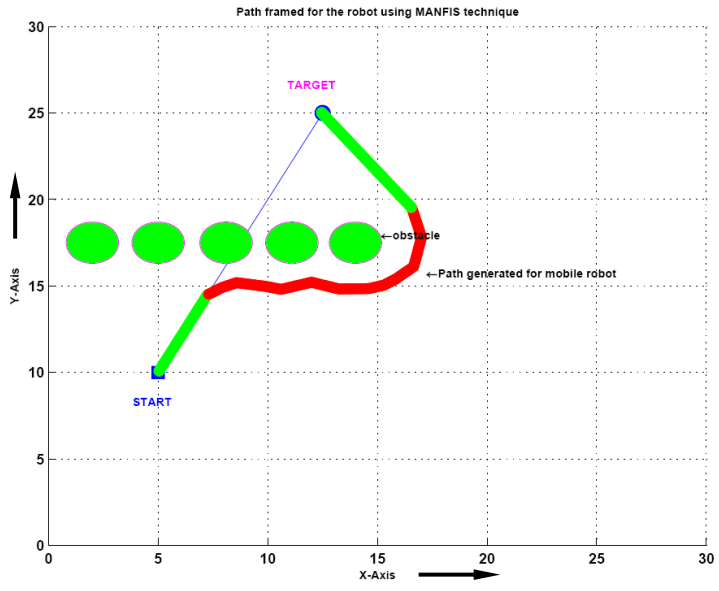


Figure 4.19 Obstacle avoidance and Barrier following behaviour shown by the single robot using MANFIS technique.

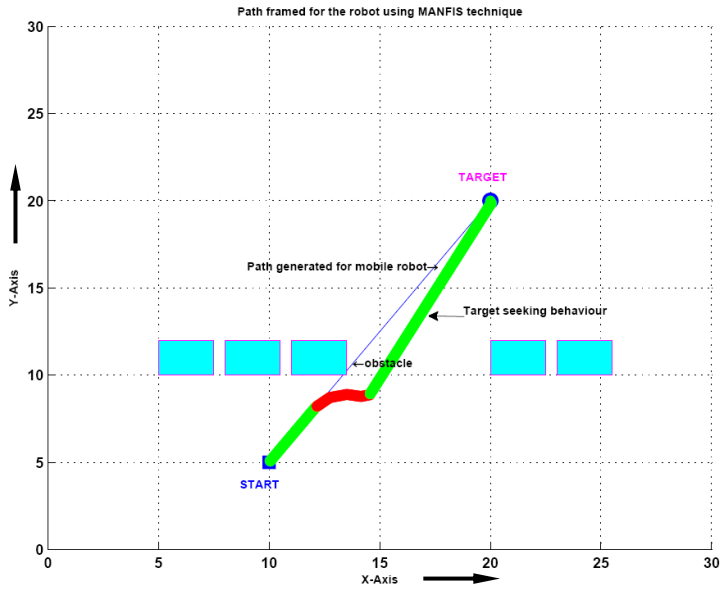


Figure 4.20 Target seeking behaviour shown by the single robot using MANFIS technique.

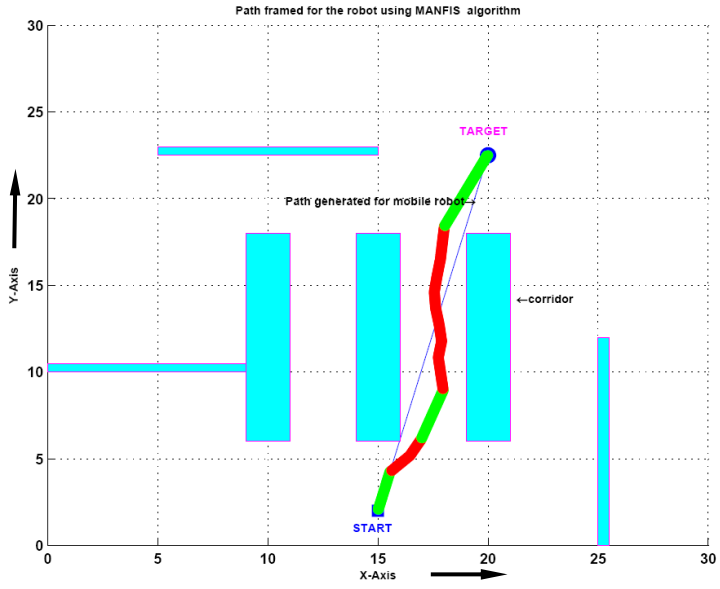


Figure 4.21 Single robot following a corridor to reach at target using MANFIS technique.

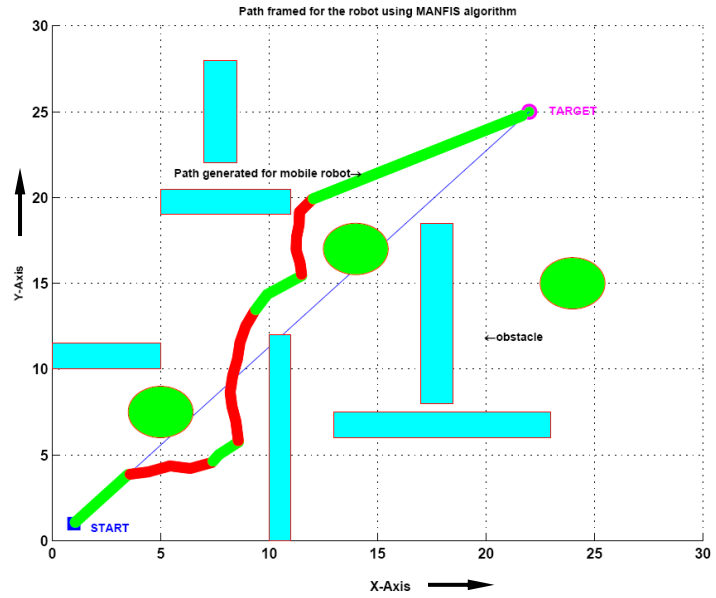


Figure 4.22 Single robot navigating inside a dense environment to reach at target using MANFIS technique.

4.6.2 Experimental validation with Real Mobile robot

After satisfying the simulation results, the proposed navigation system has been implemented and tested in our laboratory experimental platform of dimension 225cm x175cm. The details regarding experimental analysis have been discussed in the previous chapter (in section 4.4.2). To show the effectiveness and feasibility of the developed hybrid system, the simulation results are validated against experimental results and illustrated in Figures 4.23 and 4.24 respectively. The performance of the validation has been measured in terms of path length and time taken by the robot to reach the target, and it is tabulated in the Tables 4.9 and 4.10 respectively.

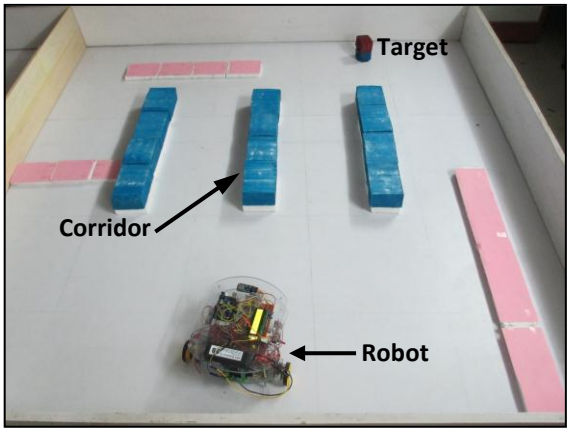


Figure 4.23(a)

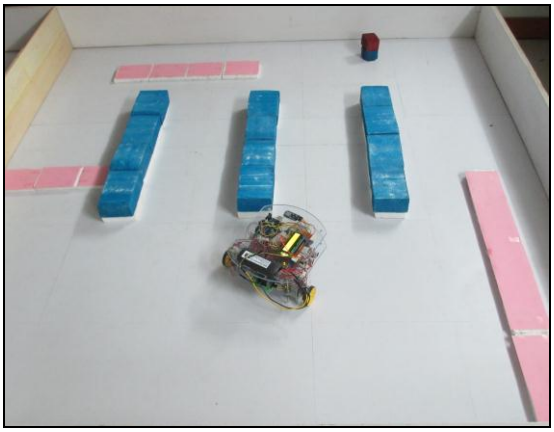


Figure 4.23(b)

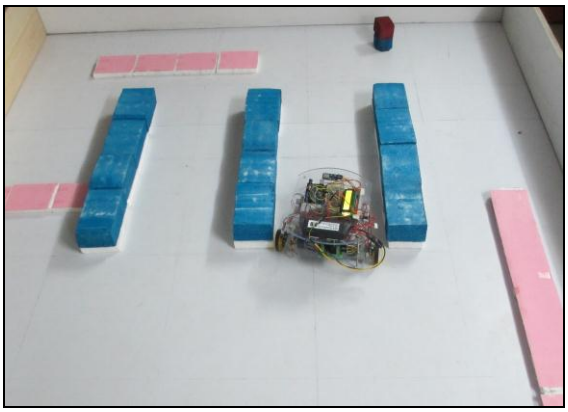


Figure 4.23(c)

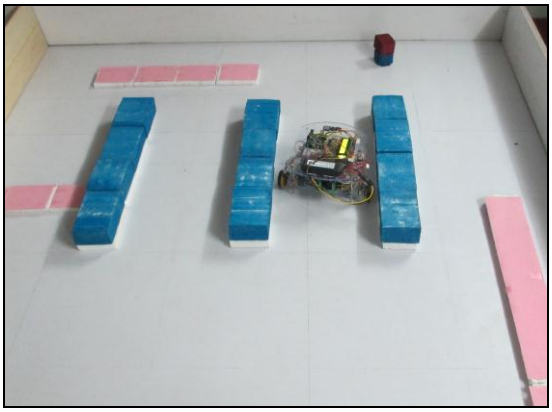


Figure 4.23(d)

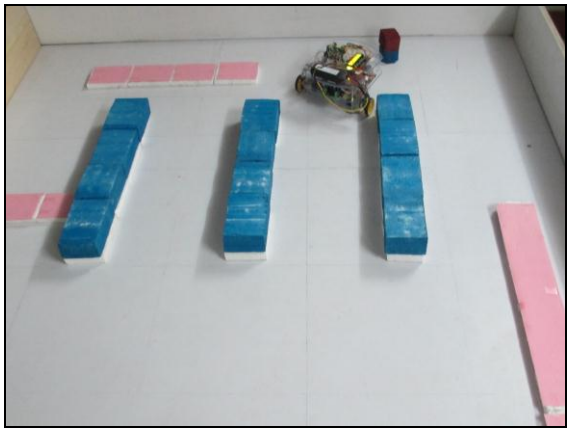


Figure 4.23(e)

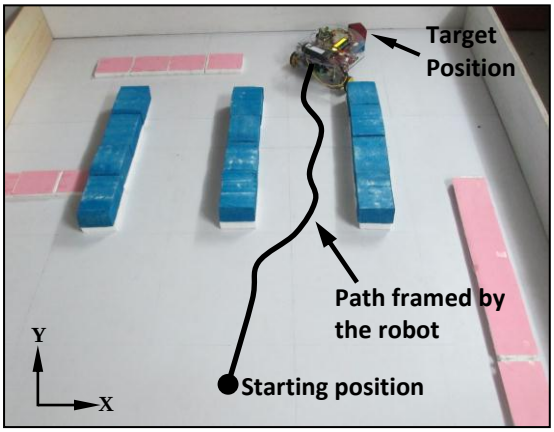


Figure 4.23(f)

Figure 4.23 (a-f) Experimental results for navigation of mobile robot in the environment shown in Figure 4.21.

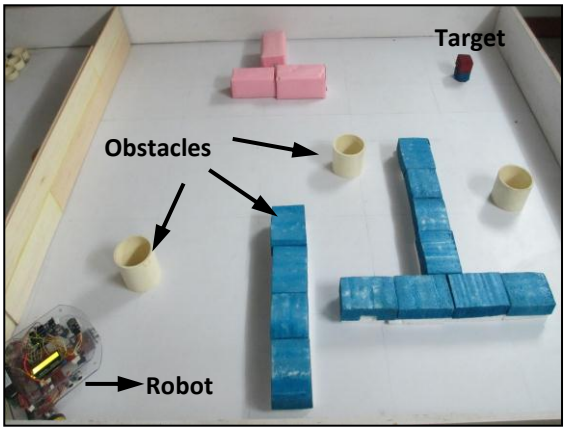


Figure 4.24(a)

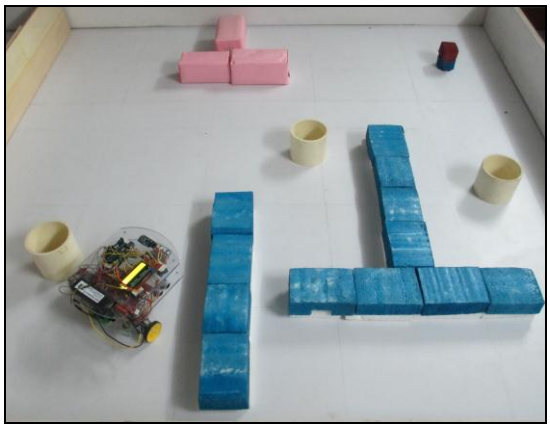


Figure 4.24(b)

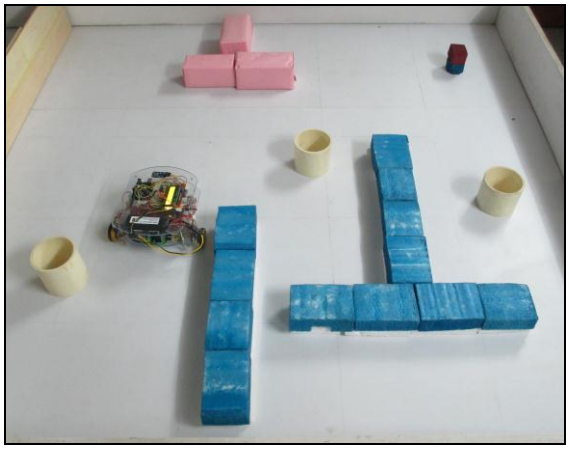


Figure 4.24(c)

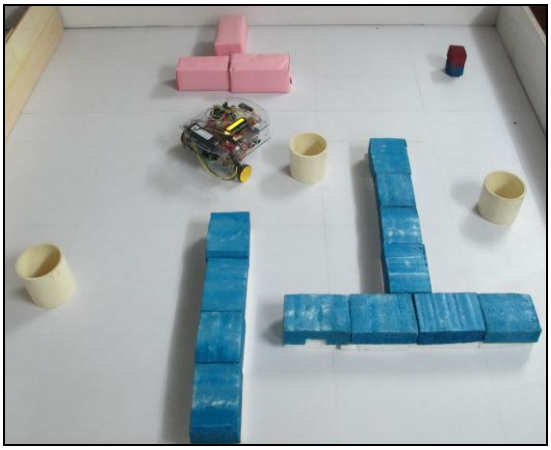


Figure 4.24(d)

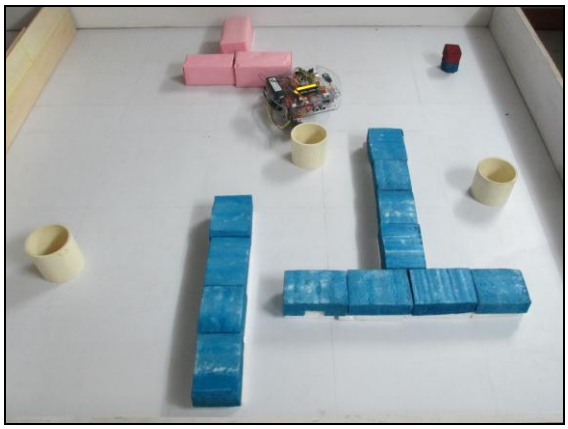


Figure 4.24(e)

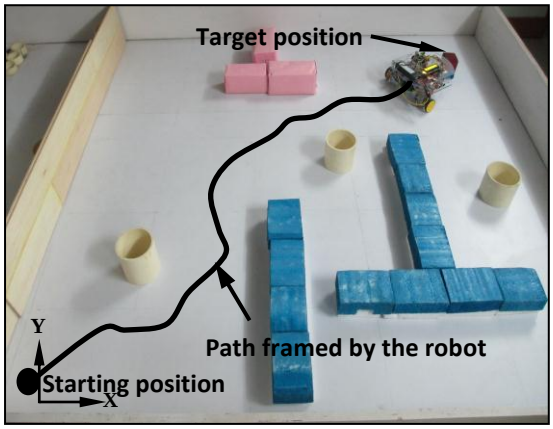


Figure 4.24(f)

Figure 4.24 (a-f) Experimental results for navigation of mobile robot in the environment shown in Figure 4.22.

Table 4.9 Path length covers by the robot in simulation and experimental test to reach the target.

Sl No.	Path length in simulation ('cm')	Path length in real time test (in 'cm')	% of error
Scenario-1	108.36 (Fig. 4.21)	115.33 (Fig. 4.23 f)	6.43
Scenario-2	182.98 (Fig. 4.22)	195.02 (Fig. 4.24 f)	6.58

Table 4.10 Time taken by the robot in simulation and experimental test to reach the target.

Sl No.	Time taken in simulation (in 'sec')	Time taken in real time test (in 'sec')	% of error
Scenario-1	11.23(Fig. 4.21)	12.01 (Fig. 4.23 f)	6.94
Scenario-2	17.43(Fig. 4.22)	18.65 (Fig. 4.24 f)	6.99

4.6.3 Comparison of the Design MANFIS Navigational Controller with other Models

In this part, a comparative study has been carried out from the developed MANFIS approach with results proposed from the other techniques. The simulation study has been performed to show the performance of the proposed hybrid system. The size dimension of simulation platforms are considered as no. of units and each unit is in millimeter (mm).

In the first comparative study, the proposed MANFIS navigational controller has been implemented in an environment shown in Figure 4.25(a). It has been noticed that using the developed method the robot covers a shorter path compared to the result proposed by Shi et al.[116]. However, it can be seen from the Figure 4.25(a) that the fuzzy-neural method proposed by author produces a smoother path to reach the goal, but it doesn't guarantee a shorter compared to the MANFIS technique results shown in Figure 4.25(b).

In the second comparative study, the developed hybrid path controller has been applied in a similar environment as stated by the Mo et al.[87]. They used the fuzzy behaviour based navigational controller to navigate the robot safely in very small gaps as illustrated in Figure 4.26(a). From the simulation result, it is clearly seen that where the robot is making turns, there are some instances where the mobile robot closer to an obstacle than the required safe distance. This problem has been taken care of by the

proposed hybrid algorithm and depicted in the Figure 4.26(b). The effectiveness of both comparative studies has been measured on the basis of path length and tabulated in Table-4.11. A comparative study has been carried out between ANFIS and MANFIS navigation systems in terms of path lengths to demonstrate the effectiveness of the path planner and tabulated in Table-4.12.

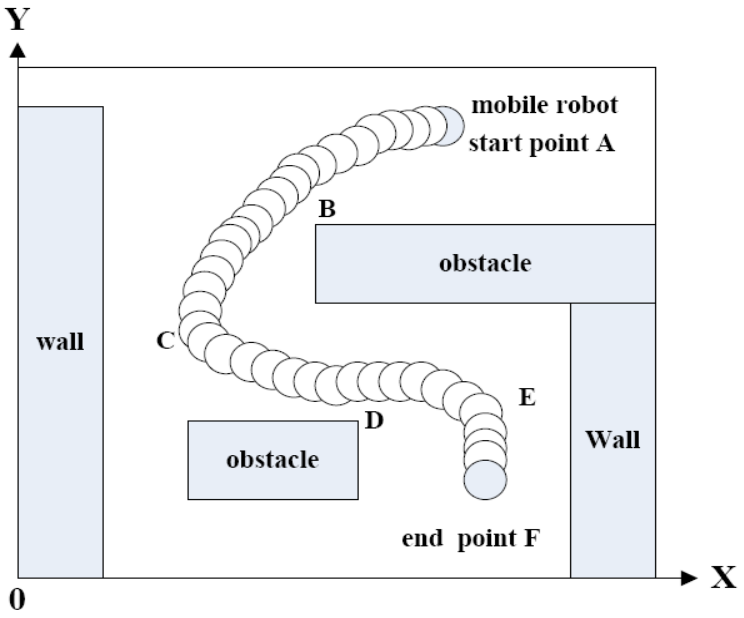


Figure 4.25 (a) Navigation path framed for a single mobile robot to reach target by Shi et al. [116].

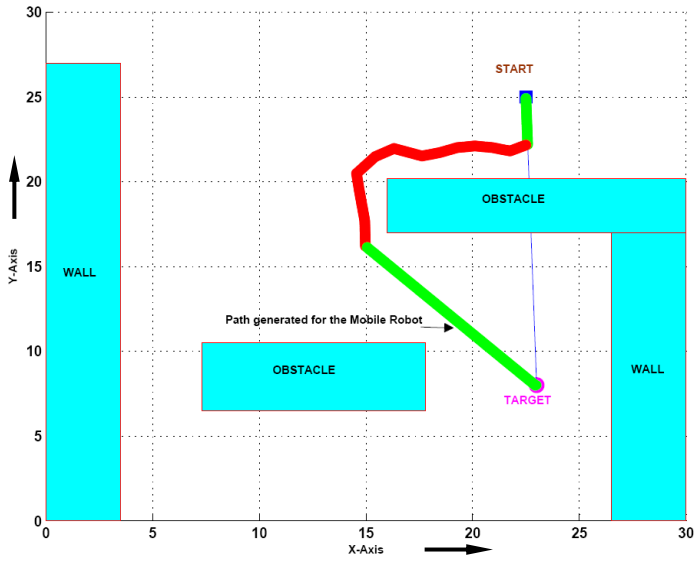


Figure 4.25 (b) Navigation path framed for a single mobile robot to reach target using developed MANFIS method.

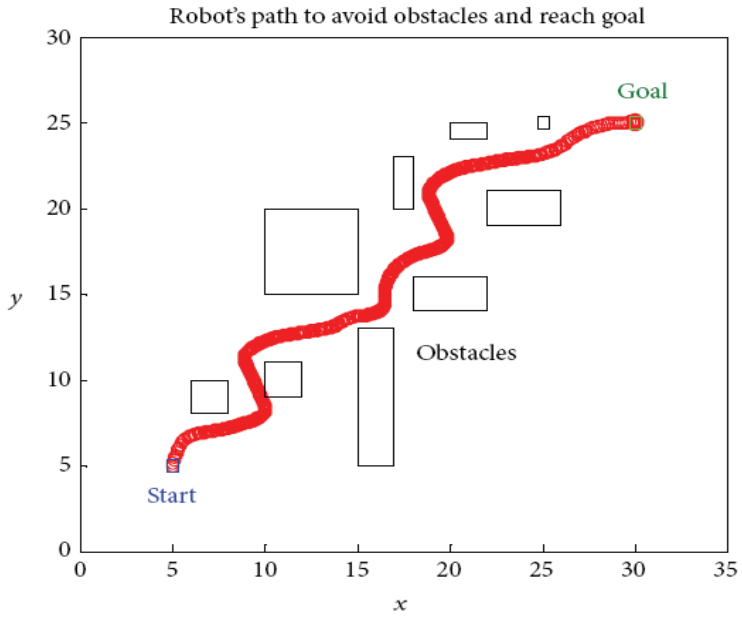


Figure 4.26 (a) Navigation path framed for a single mobile robot to reach target by Mo et al. [87].

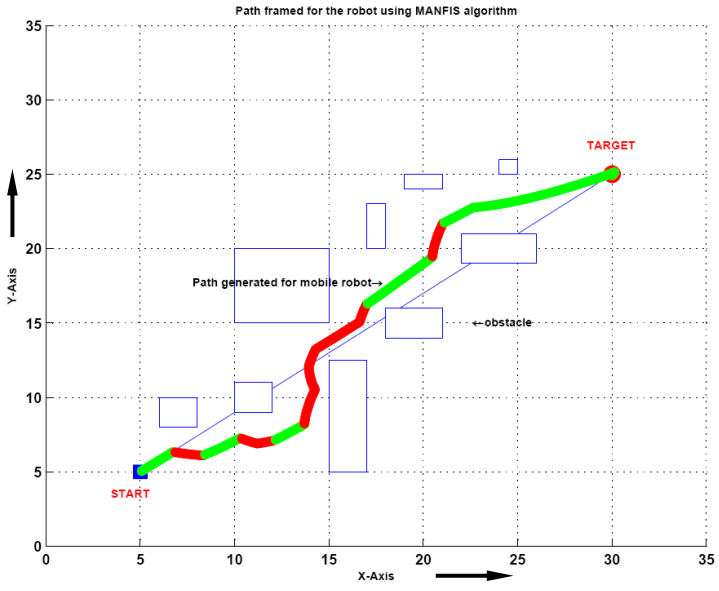


Figure 4.26(b) Navigation path framed for a single mobile robot to reach goal using developed MANFIS method.

Table 4.11 Comparison of results in terms of path length.

Sl No.	Environment types	Path length from current system (in 'cm')	Path length of reference model (in'cm')	% of deviation
1	Complex environment with long obstacles Figures 4.25(a) and 4.25(b)	7.6	8.8	13.64
2	Maze environment Figures 4.26(a) and 4.26(b)	7.9	9.3	15.05

Table 4.12 Comparison of ANFIS and MANFIS results in terms of path length.

Sl No.	Environment types	Path length using ANFIS system (in 'cm')	Path length using MANFIS model (in'cm')	%of deviation
1	Scenario-1	6.34	6.61	4.08
2	Scenario-2	8.30	8.59	3.37
3	Scenario-3	7.78	8.10	5.18
4	Scenario-4	10.25	10.49	2.28

4.7 Summary

This chapter has described the implementation of ANFIS and MANFIS techniques for the mobile robot navigation.

The following salient features are drawn based on the simulation and experimental results using ANFIS technique.

- The developed novel ANFIS navigational controller has been successfully used to control the robot in a highly cluttered environment.
- With the help of this proposed navigation system, the robots are able to perceive the environment condition and reach the goal successfully.
- In the simulation results, it clearly observed that various reactive behaviours such as obstacle avoidance, barrier following and target seeking have been performed by the proposed navigational controller.
- A series of practical tests have been carried out with a real developed robot to show the efficacy and effectiveness of the proposed navigation algorithm. They are found to be in good agreement. The percentages of errors are found to be within 7% for both path length and time taken for the robot to reach the target and tabulated in Tables 4.4 and 4.5 respectively.
- A comparative study is carried out between the performance of the proposed navigation and those of obtained by authors [87, 116] in simulation mode. It has been observed that the proposed navigation method provides better results compared to other techniques. The performance of the comparison study is mainly measured in terms of path length and tabulated in Table 4.6.

The following conclusions are drawn on the basis of simulation and experimental results using MANFIS methodology.

- The proposed methodology has been successfully implemented for solving the navigational problem of a mobile robot in an unknown or partially known static environment.
- The efficacy of the proposed navigation technique has been demonstrated through various exercises. During experimentation, it has been observed that the proposed path planner has the ability to avoid obstacles in a cluttered environment.
- The real time experimental tests have been performed to validate the developed navigation system. The percentages of errors are found to be within 7% for both path

length and time taken for the robot to reach the target and tabulated in Tables 4.9 and 4.10 respectively.

- A comparative graphical study has been demonstrated to find out the effectiveness of the proposed navigational controller and percentage of deviation is shown in the Table-4.11.
- Finally, it has been inferred that the proposed path planning system produce closer results to the ANFIS navigation system.

In the consequent chapters Cuckoo search algorithm and Invasive weed optimization algorithm have been investigated and examined as standalone methods. These algorithms are then combined and hybridized with ANFIS to produce better navigational controller for mobile robots.

CHAPTER-5

5. ANALYSIS OF CUCKOO SEARCH ALGORITHM FOR NAVIGATION OF MOBILE ROBOTS

This chapter presents the analysis of a navigational controller for solving the path planning problem of the mobile robot based on the Cuckoo search (CS) algorithm. The new novel technique is implemented in path optimization problem of the mobile robot. This nature inspired metaheuristic algorithm is analyzed and applied to the real robot in order to get the effective path planning.

5.1 Introduction

Nowadays, there are so many nature inspired metahuiristic algorithms used in various areas of science and engineering. Metahuiristic approaches such as Genetic algorithm, Evolution strategies and Differential evolution are usually bio-inspired and don't expose the drawbacks of the classical techniques (Linear programming and Non-linear programming). In recent times, most popular nature-inspired meta-heuristic algorithms such as Genetic Algorithm (GA), Simulated Annealing (SA), Ant Colony Optimization (ACO), Firefly Algorithm (FA) and Particle Swarm Optimization (PSO) have been applied for path planning of mobile robots. Most of these meta-heuristic algorithms are inspired and mimicked from successful features of the biological, physical or sociological systems. These methods have gained acceptance because of their ability and efficiency of searching for an optimal solution in a defined problem space. The analysis and development of navigational algorithms for a mobile robot in an unknown environment densely cluttered with obstacles is one of the major issues in the robotics field. In this current investigation, CS algorithm has been implemented for the mobile robot path optimization problem. An effective navigational algorithm has been designed using CS algorithm, which will steer the mobile robot safely from the source point to the destination point with covering the minimum path and shortest possible time. The distance between the robot position to obstacles and target are commonly adopted criterion for the proposed algorithm. The CS algorithm can optimize the above criterion for the robot during navigation.

5.2 Overview of Cuckoo Search Algorithm

In 2009, Yang and Deb [193] introduced an efficient Cuckoo search (CS) algorithm, and it has been seen that CS is more effective than the other existing metahuiristic algorithm including Particle swarm optimization (PSO). The CS algorithm is inspired by the brood parasitism of some cuckoo species. In nature, cuckoos use an aggressive reproduction strategy that includes the female laying their eggs in the nests of other birds. If the eggs are discovered by the host birds, they may abandon the nest completely or throw away the alien eggs. This phenomenon of evolving to best lay parasitic eggs is the essence of the cuckoo search algorithm.

For the implementation of CS method Yang and Deb [193] have used following three idealized rules [192]:

- Each cuckoo lays one egg at a time, and dumps it in a randomly chosen nest.
- The best nests with the high quality of eggs (solutions) will carry over to the next generations.
- The number of available host nests is fixed, and a host can discover an alien egg with a probability $P_a \in [0,1]$. In this case, the host bird can either throw the egg away or abandon the nest so as to build a completely new nest in a new location.

For simplicity, this last rule can be approximated by a fraction P_a of 'n' nests being replaced by the new nests (with new random solutions at new locations). Pseudo code of CS algorithm is presented in Figure 5.1.

When generating new solutions $X^{(t+1)}$ for, say cuckoo *I*, a levy flight random walk is performed. A levy flight [226] is used for performing effectively both local and global searches in the solution space and the steps are distributed in terms of step lengths according to a heavy tailed probability distribution. In levy flight, the random walk is distributed according to the power law that is

$$y = x^{-\beta} \tag{5.1}$$

where $1<\beta<3$ and therefore has an infinite variance.

For the present problem, we used the simplest approach where each nest has only a single egg. The following equation number (5.2) is used to produce a new solution $X^{(t+1)}$, for a cuckoo *I*, by a levy flight:

$$X_i^{(t+1)} = X_i^{(t)} + \alpha \oplus levy(\lambda) \tag{5.2}$$

where α>0 is the step size which should be related to the scales of the problem of interest. In most of the cases, we can use α= 1. The product $\oplus$ means entry wise multiplication. This entry wise product is similar to those used in PSO, but here the random walk via Levy flight is more efficient in exploring the search space, as its step length is much longer in the long run.

Objective function $f(x),\ x=(x_1,x_2,.....x_d)^T$

Generate initial population of n host nests $x_i(i=1,2,.....\text{n})$

while (t<Max Generation) or (Stop criterion) do

Get a cuckoo randomly by Levy flights

Evaluate its quality/fitness F_i

Choose a nest among n (say j) randomly

If $(F_i > F_j)$ **then**

replace j by the new solution

end

A fraction (p_a) of worse nests are abandoned and new ones- are built

Keep the best solutions (or nests with quality solutions)

Rank the solutions and find the current best

End while

Post process results and visualization

End

Figure 5.1 Algorithm of Cuckoo search.

5.3 Problem Formulation for Robot Path Planning with CS Algorithm

The shortest/optimal path planning is essential for efficient operation of autonomous vehicles. In this chapter a new nature inspired meta-heuristic algorithm has been applied for mobile robot path planning in an unknown or partially known environment populated by variety of static obstacles.

The path planning problem is one of the most fundamental issues in the robot navigation field. CS algorithm is a new optimization concept which broadly falls under meta-heuristic evolutionary computation techniques. The objective of the problem considered in the section is robot path planning in a partially or totally unknown environment populated by a variety of static obstacles. If the robot moves in an environment with unknown obstacles, it is essential to detect and avoid obstacles as the robot moves towards the target. Based on the sensory information about the target and obstacles, we adopt the CS algorithm to solve it. First, we have transformed the navigation problem into a minimization problem and formulated an objective function equation based on the position of the goal and the obstacles present in the unknown environment. The objective function satisfies the conditions of obstacle avoidance and target seeking behavior of robot present in the terrain. Depending on the objective function value of each nest (cuckoo) in the swarm, the robot avoids the obstacles and proceeds towards the target. Here, we have implemented the CS algorithm to solve the above path optimization problem. During this process of result and visualization, the locations of the globally best nest (cuckoo) in each iteration are chosen and robot moves to these locations in the series. When the robot does not detect any obstacles in its target path, then it will travel directly towards its destination. Then it is not necessary to implement any intelligence computing technique to travel the robot within its environment. The developed flow chart diagram for the current analysis is given in Figure 5.2.

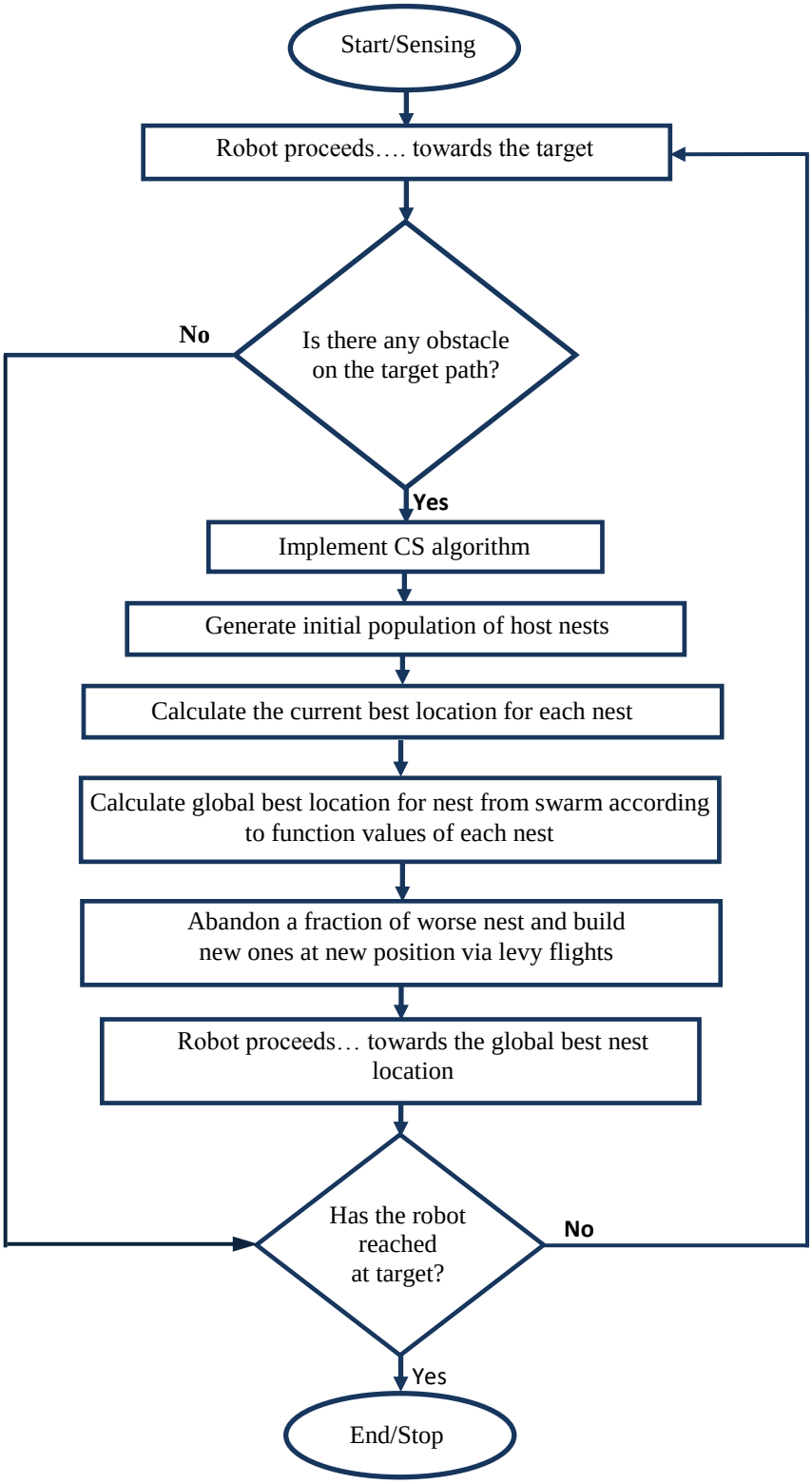


Figure 5.2 Flow chart represented for the proposed navigation system using CS algorithm.

5.3.1 Formulation of the Objective Function using the CS Algorithm

Let the robot is at start point (A) and moves to the goal point (B) in the environment as shown in Figure 5.3. The line A to B is the desired route for the robot to reach the target with the optimal path length. The robot has to reach the target while avoiding obstacles present in the environment. When the robot reaches near the obstacles, its sensors detects obstacles. Then the CS algorithm will be activated to avoid the obstacles present in the desired target path. Note that whenever an obstacle falls within the periphery of the robot sensing range, the CS algorithm will be initiated to find the best nest position for the robot (according to objective function value) to avoid collisions with obstacles present in the target path. Each solution in the problem space is associated with a numeric value. In the CS algorithm, a nest egg of the best quality (chance of producing cuckoo chick) will lead to a new generation. Therefore, the quality of a cuckoo's egg (new cuckoo) is related to the optimized path length of the robot. As explained in the above meta-heuristic optimization problem, each step to be moved by the robot is calculated based on the distance between the positions of the nests (cuckoo) to the goal and the obstacles present in the unknown environment.

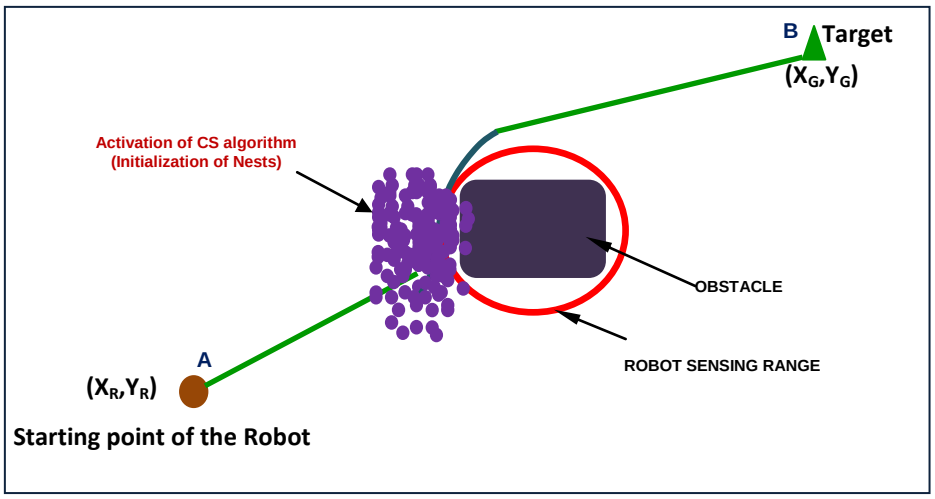


Figure 5.3 Activation of CS Algorithm.

In general, the robot path planning mainly depends upon the following two behaviors,

a) Obstacle avoidance behavior

The obstacle avoidance behavior is used to avoid hitting with obstacles (such as walls) present in the environment. The position of the best nest (cuckoo) should have kept up the maximum safe distance from the nearest obstacle. The Euclidean distance between the best nest and nearest obstacle is calculated by the following equation in terms of objective function as follows:

$$(Dist.)_{N-OB} = \sqrt{\left(x_{OB} - x_{N_i}\right)^2 + \left(y_{OB} - y_{N_i}\right)^2} \qquad 5.3\ (a)$$

where, x_{Ni} and y_{Ni} are the nest (cuckoo) position at *x* and *y* coordinates. $(Dist.)_{N\text{-}OB}$ is the Euclidean distance from the nest (cuckoo) to the nearest obstacle.

Note: The nearest obstacle to the robot can be calculated by the following expression:

$$(Dist.)_{R-OB} = \sqrt{\left(x_{OB_n} - x_{ROB}\right)^2 + \left(y_{OB_n} - y_{ROB}\right)^2} \qquad 5.3(b)$$

b) Target seeking behavior

In its simplest form, the move to target behavior can express the desire to move to a specific robot position. The position of the best nest (cuckoo) should have kept up the minimum distance from the goal. The Euclidean distance between the nest and goal is calculated by the following equation in terms of objective function as follows:

$$(Dist.)_{N-G} = \sqrt{\left(x_G - x_{N_i}\right)^2 + \left(y_G - y_{N_i}\right)^2} \qquad 5.3(c)$$

where, x_G and y_G are the goal position at *x* and *y* coordinates. $(Dist.)_{N\text{-}G}$ is the minimum Euclidean distance from the nest (cuckoo) position to the robot.

Based on the above two important behaviors of the robot, the objective function of the each nest for the path optimization problem can be expressed as follows:

$$Objective\ function(f_i) = C_1.\frac{1}{\min\limits_{OB_j \in OB_d} \left\| Dist_{N\text{-}OB_d} \right\|} + C_2.\left\| Dist_{N\text{-}G} \right\| \qquad (5.4)$$

Based on the above-discussed objective function, here we assumed that the 'n' number of obstacles are present in the environment, and we represent them as OB_1, OB_2, OB_3 ... OB_n, their center coordinates are (x_{OB1}, y_{OB1}), (x_{OB2}, y_{OB2}), (x_{OB3}, y_{OB3})... (x_{OBn}, y_{OBn}). Due to threshold of the robot sensor, in each move it can recognize a number of obstacles present in the environment and number of obstacles being recognized by the robot sensor in some stage are represented as $OB_d \in \{OB_1, OB_2, OB_3, \ldots OB_n\}$. It can be seen from the objective

function that when N_i is nearer to the goal, the objective function value of $\|Dist_{N-G}\|$ will be reduced and when N_i is away from the obstacles, the objective function value of $\min_{OB_j \dots OB_d} \left\|Dist_{N-OB_j}\right\|$ will be large. So from the above discussions, we have concluded that path planning problem for mobile robot solved by CS algorithm is a minimization one.

The objective of the CS algorithm is to minimize the objective function (f_i). When a robot falls in an obstacle sensing range, obstacle avoidance behavior based on the CS algorithm is activated to find the best position for the nest (Cuckoo). From equation no. (5.4), it can be clearly observed that the nest or cuckoo having minimum objective function value can be the best nest (cuckoo) in the swarm and the corresponding nest is maintaining the maximum distance from the nearest obstacle and minimum distance from the target. The selection of best nest (cuckoo) will continue for several iterations until the robot avoids obstacles or reaches its target.

From the objective function equation, it can be clearly understood that the controlling or fitting parameters C_1 and C_2 have influence on the robot trajectory. When C_1 is large, the robot will be far away from the obstacles if it is less there will be a chance to hit with them. Other side, when C_2 is large, the robot has a great potential to move to the goal, resulting in the path length being short, otherwise it will be large. The proper selection of the controlling parameters may result in faster convergence of the objective function and elevation of the local minima point. In this work, we have chosen the controlling parameters by the trial and error methods.

Steps of CS algorithm for Mobile Robot Navigation:

Step 1: Initialize the start and goal position of the robot.

Step 2: Proceeds the robot towards the goal until it will be stuck by an obstacle.

Step 3: When the target path is obstructing by an obstacle, implement the CS algorithm.

Step 4: Generate the initial population of host nests, each representing one trial solution of the proposed optimization problem.

Step 5: Calculate the current best nest and then find the global best nest in the swarm by equation no. 5.4.

Step 6: Abandon the worse nest and built the new nest by Levy flight (equation no. (5.2).

Step 7: Proceed the robot towards the best nest position.

Step 8: Repeat the steps 2-7 until the robot avoids obstacles or reaches its goal.

5.4 Demonstrations of the CS based Path Planner

The proposed CS based path planner has been employed in both simulations as well as real time experimental scenario with partially or totally unknown environments. The simulation results are conducted using MATLAB [229] processing under Windows XP. All simulation results are conducted on PC with Intel core2 Duo processor running on 3.0GHz, 4GB RAM and a hard disk of 160GB. A real time experimental environment has been set-up in the laboratory containing static obstacles as well as a static target to validate the simulation results. Finally, the effectiveness of the CS based path planner is analyzed, discussed and compared for the path optimization problem.

5.4.1 Simulation Result and Discussions

The navigation workspace for a collision free motion and obstacle avoidance of a single robot with the single target has been shown in Figure 5.4. This exercise demonstrates that the robot reaches the target without colliding with obstacles and by following the shortest trajectory. It can be observed that the robot moves in a smooth trajectory from its start point to end point by maintaining a safe distance from the obstacles and finds the destination efficiently. In simulation graph, the red color path represents the path generated by using CS algorithm.

The simulation result shown in Figure 5.5 involves a single robot escaping from a narrow passage using CS algorithm. It has been observed that the robot reached the target in an efficient manner without any collision with obstacles present in the unknown environment using CS technique. The CS path planner has been demonstrated for other cases of simulations shown in Figures 5.6 and 5.7 respectively. The parameter values considered for the proposed algorithm are given in Table 5.1.

Table-5.1 (Parameters used in CS algorithm.)

Parameters	**Values**
No. of Nests	10-50
No. of Iterations	100-150
P_a	0.05-0.3
C_1	0.1 to 1
C_2	0.01 to 0.0001

In the simulation experiments, we have tried to vary the no. of nests (population/nest size) n=10, 20, 30, 40, 50, 60 and 70 and mutation probability factor P_a = 0.1, 0.15, 0.2, 0.25 and 0.3. From our simulation results analysis, it is clearly observed that at n=25-30 and P_a=0.25, good results are provided compared to the other conditions. The simulation results in different choice parameters in the CS algorithm (N and P_a) will affect the shape of the trajectory and are illustrated in Figures 5.8-5.10. The path length covered by the robots during simulation has been presented in Tables 5.2 to 5.4 respectively. In the simulation graph length of X and Y axes are considered as 40x40 units and each unit is equal to 2mm.

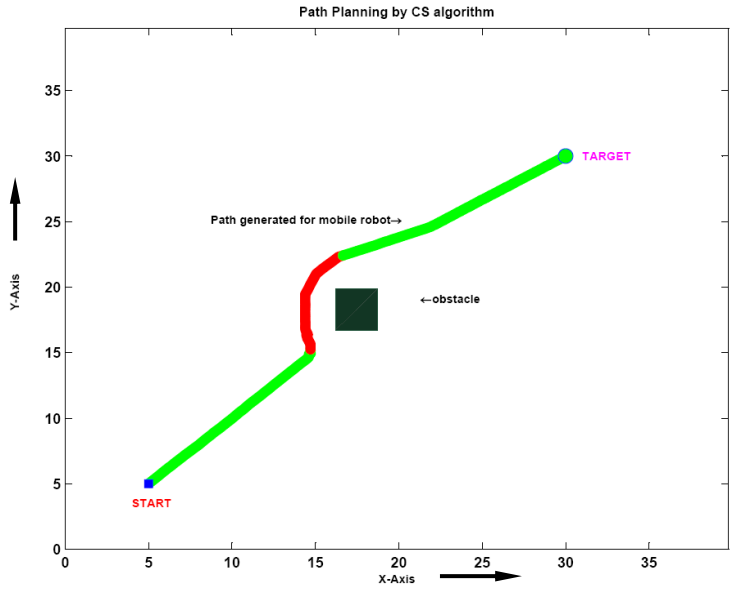


Figure 5.4 Obstacle avoidance behaviour by a single robot using CS algorithm.

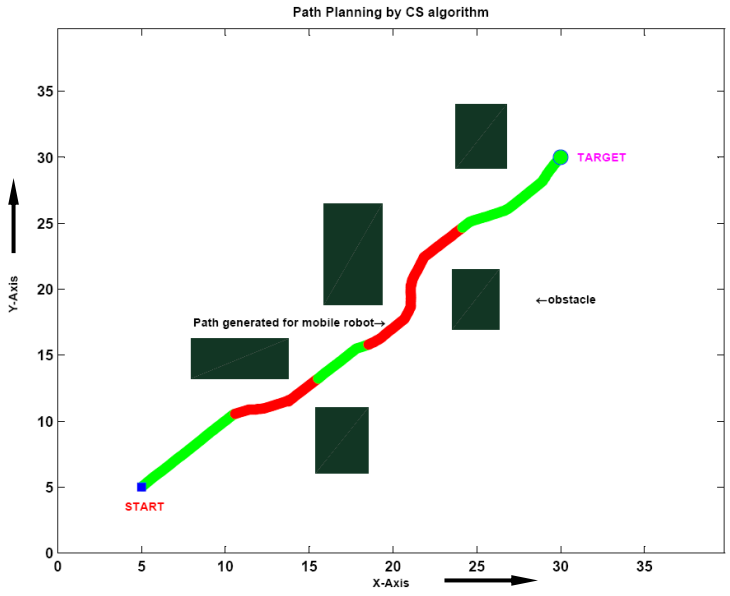


Figure 5.5 Single robot escaping from narrow end using CS algorithm.

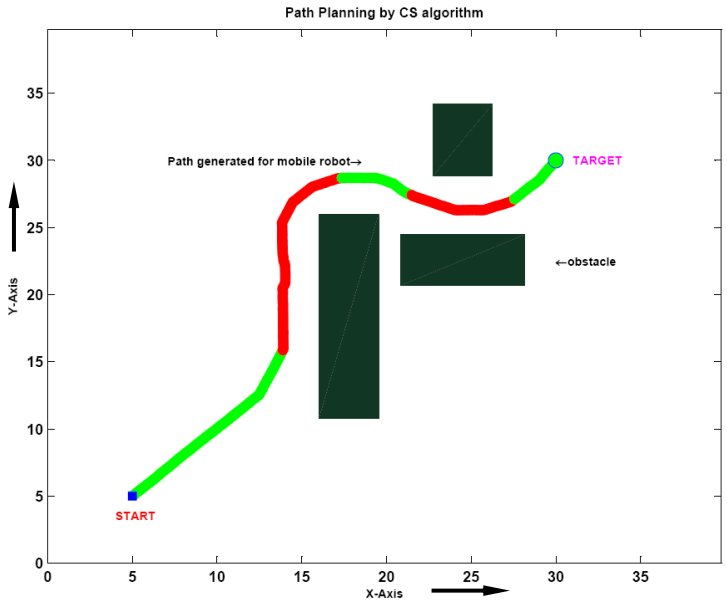


Figure 5.6 Single robot escaping from trap condition using CS algorithm.

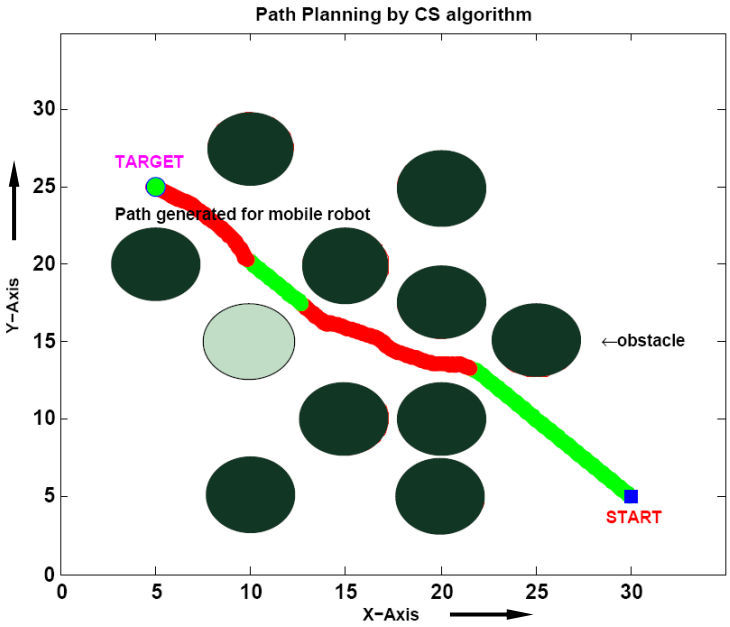


Figure 5.7 Single robot navigating in a highly cluttered environment to reach the target using CS algorithm.

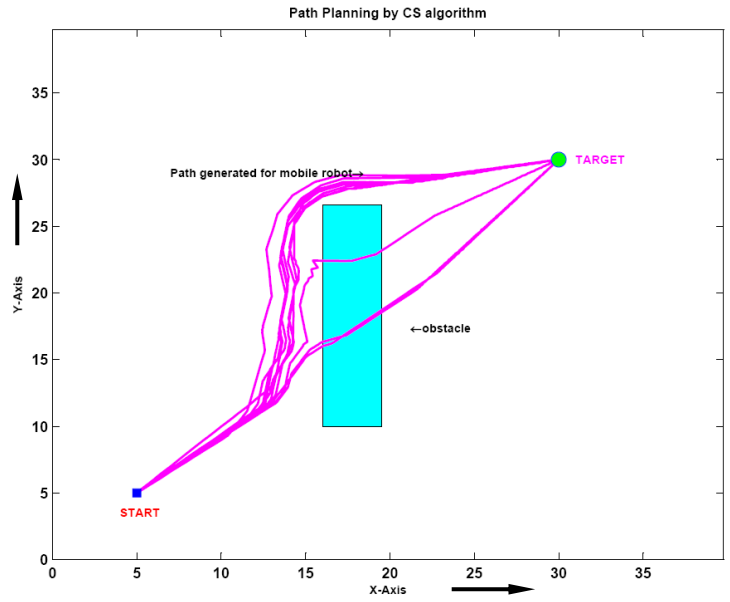


Figure 5.8 Path generated for the mobile robot to avoid a wall at different values of N and P_a of CS algorithm.

Table-5.2 Path length covered by the robot during simulation by considering different values of P_a and N. (shown in Figure 5.8)

Sl. No.	P_a	N	Path length covered by the robot (in 'cm')	Avoid Collision (Yes/No)
1	0.05	10	6.88	No
2	0.10	15	7.14	No
3	0.30	25	7.80	Yes
4	***0.25***	***26***	***7.74 (optimal)***	***Yes***
5	0.25	30	7.96	Yes
6	0.25	33	8.01	Yes
7	0.28	35	8.12	Yes
8	0.27	37	8.09	Yes
9	0.30	40	7.99	Yes
10	0.30	45	8.44	Yes

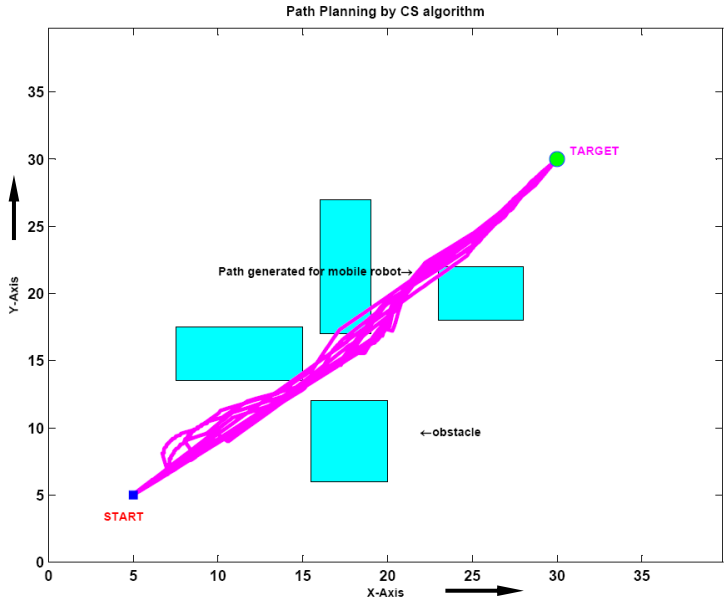


Figure 5.9 Path generated for the mobile robot to escape narrow condition at different values of N and P_a of CS algorithm.

Table-5.3 Path length covered by the robot during simulation by considering different values of P_a and N. (shown in Figure 5.9)

Sl. No.	P_a	N	Path length covered by the robot (in pixels)	Avoid Collision (Yes/No)
1	0.05	15	6.89	No
2	0.15	18	6.86	No
3	***0.26***	***25***	***6.73 (optimal)***	***Yes***
4	0.29	25	6.65	No
5	0.25	30	6.78	Yes
6	0.25	33	6.80	No
7	0.28	35	6.58	No
8	0.26	36	6.84	Yes
9	0.30	41	6.82	No
10	0.30	45	7.01	No

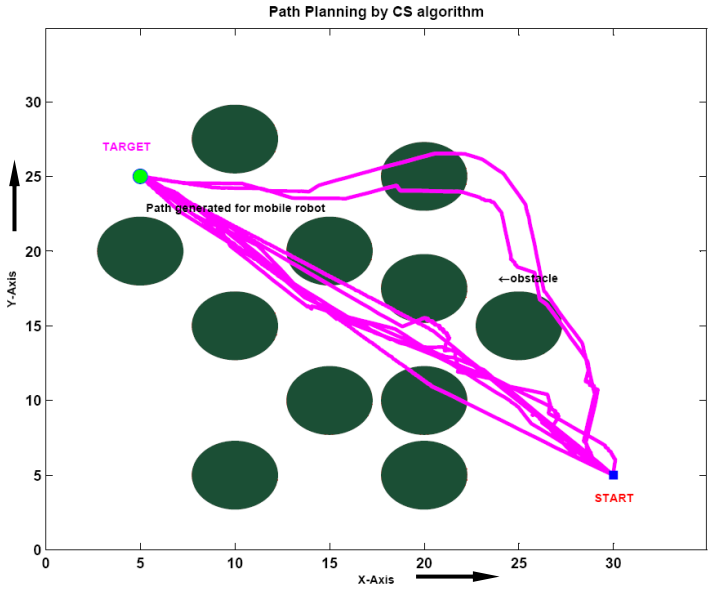


Figure 5.10 Path generated for the mobile robot in a maze environment at different values of N and P_a of CS algorithm.

Table-5.4 Path length covered by the robot during simulation by considering different values of P_a and N. (shown in Figure 5.10)

Sl. No.	**P_a**	**N**	**Path length covered by the robot (in pixels)**	**Avoid Collision (Yes/No)**
1	0.05	10	8.02	No
2	0.15	15	7.92	No
3	0.29	25	6.67	Yes
4	***0.25***	***26***	***6.60 (optimal)***	***Yes***
5	0.30	27	6.71	Yes
6	0.25	30	6.64	Yes
7	0.30	32	6.84	No
8	0.25	33	6.69	Yes
9	0.26	34	6.89	No
10	0.28	35	6.95	No
11	0.30	45	6.92	No

5.4.2 Experiments with Real Mobile Robot

To show the effectiveness of the proposed algorithm, a variety of real time experiments have been conducted using Khepra-III mobile robot and programming languages are used in MATLAB [228] and C++. MATLAB code is used to execute the path planning algorithm in simulation mode, and the C++ code is used to get the executable program during experimental analysis with a real robot. The experimental investigation has been carried out using an environment containing a variety of obstacles of different shapes and sizes. The mobile robot has 10 infrared sensors, and 5 ultrasonic sensors mounted around the front periphery in order to extract the information about the position of the obstacles and the target. The specification details of the Khepera-III mobile robot are given in Appendix-A. The proposed navigation algorithm is authenticated at following scenarios as shown in Figures 5.11 and 5.12 to show the effectiveness of the developed path planner.

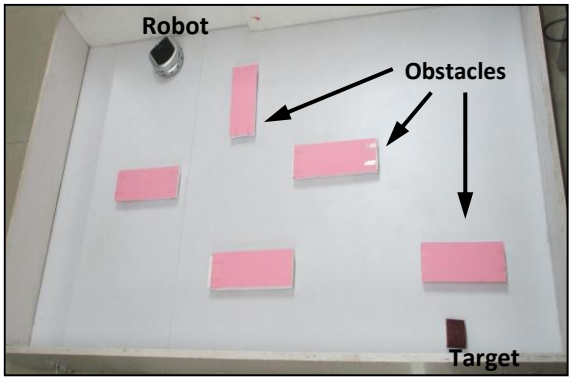


Figure 5.11 (a)

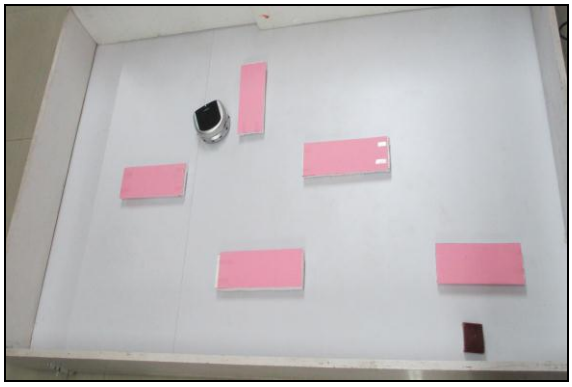


Figure 5.11 (b)

Figure 5.11 (c)

Figure 5.11 (d)

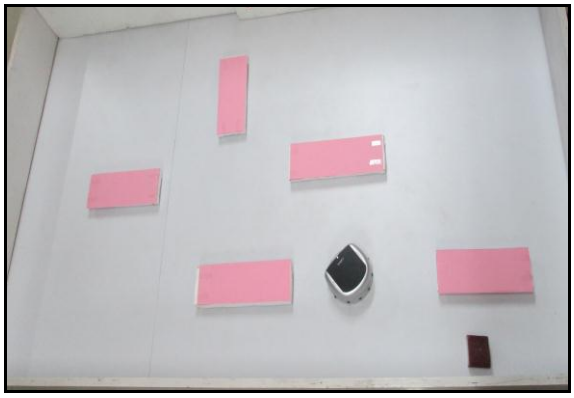


Figure 5.11 (e)

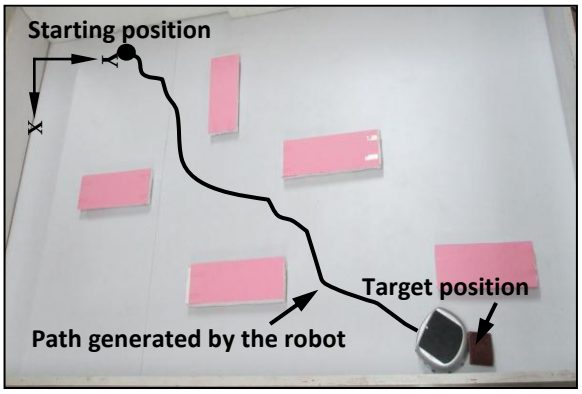


Figure 5.11 (f)

Figure 5.11 Experimental results for navigation of mobile robot in the environment shown in Figure 5.5.

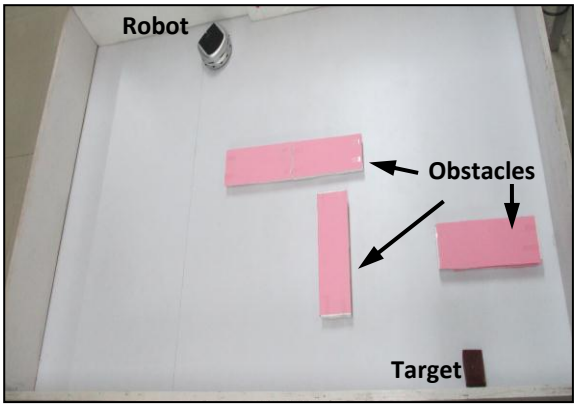


Figure 5.12 (a)

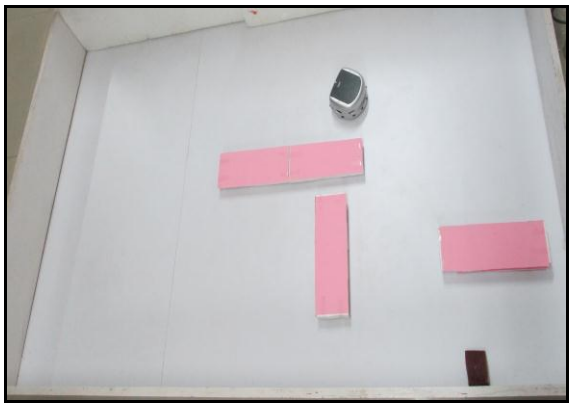


Figure 5.12 (b)

Figure 12 (c)

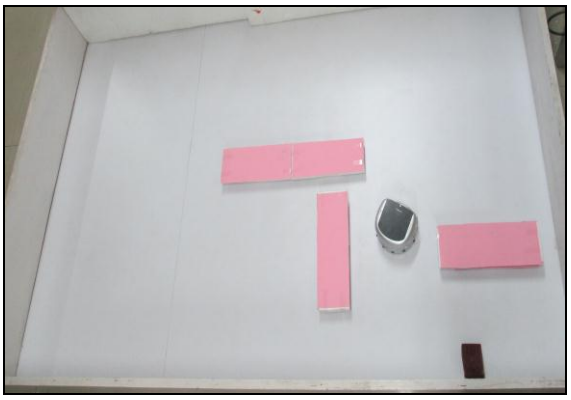


Figure 12 (d)

Figure 12 (e)

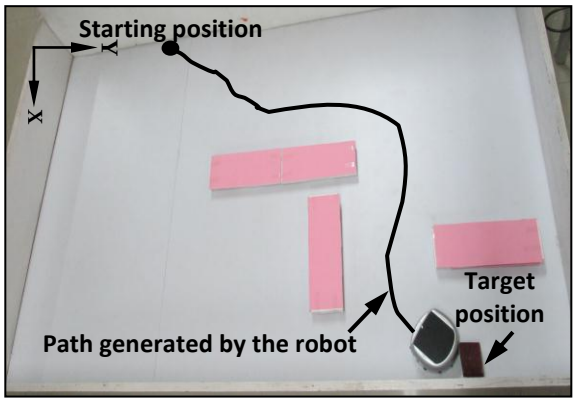


Figure 12 (f)

Figure 5.12 Experimental results for navigation of mobile robot in the environment shown in Figure 5.6.

Two different environmental conditions (Figures 5.5 and 5.6) which are already verified in simulation mode have been verified experimentally (Figures 5.11 and 5.12) to show the effectiveness and feasibility of the proposed navigational controller. The autonomous mobile robot decides its path in the shortest trajectory to reach the desired target by getting the optimized objective function values obtained from meta-heuristic based cuckoo search algorithm. The paths obtained from experiments follow closely to those traced by the robots during simulation mode. From the above experimental study, it can be clearly seen that the robot can indeed avoid obstacles and reach the destination successfully. It has been concluded by comparing the results of the simulation as well as experimental study that, the robot can follow the route by using the developed navigational algorithm for successfully reaching the destination without colliding with any obstacle present in the environment. The real time experimental analysis shows the effectiveness of the proposed navigational method. The performance of the current navigation system using CS algorithm has been checked on the basis of average path length (in‘cm’) and time (in‘sec’) taken by the robot to reach the target. For calculation of the path length and time taken, about 20 runs are considered for the proposed navigation system, which are presented in Tables 5.5-5.8 and the best results are presented in the simulation and experimental graphs. It has been noted that the average errors are found to be within 6% for both path length and time taken by the robots using proposed algorithm.

Table 5.5: Comparison of the path length during simulation and experimental for a single robot shown in Figures 5.5 and 5.11 using CS navigational algorithm.

No. of Runs	**Path length covered during simulation (in 'cm')**	**Path length covered during experiment (in 'cm')**	**% of error**
1	164.32	173.50	5.58
2	164.22	174.99	6.56
3	164.89	174.26	5.68
4	164.94	174.15	5.58
5	163.88	175.25	6.94
6	164.41	173.36	5.44
7	164.33	174.77	6.35
8	164.68	174.76	6.12
9	164.79	173.64	5.37
10	164.87	174.20	5.66
11	164.03	172.73	5.30
12	164.74	171.50	4.99
13	164.70	174.09	5.70
14	163.83	174.84	6.71
15	163.76	175.30	7.04
16	164.42	172.89	5.15
17	165.23	174.21	5.43
18	164.15	173.91	5.94
19	164.57	172.54	4.83
20	163.94	173.51	5.83
Average path length covered	**164.43**	**173.92**	**5.76**

Table 5.6: Comparison of path length during simulation and experimental for a single robot (Shown in Figures 5.6 and 5.12) using CS navigational algorithm.

No. of Runs	**Path length covered during simulation (in 'cm')**	**Path length covered during the experiment (in 'cm')**	**% of error**
1	197.55	207.90	5.24
2	195.30	208.45	6.73
3	199.76	208.38	4.82
4	199.29	212.54	6.65
5	201.02	211.26	5.10
6	199.63	210.11	5.25
7	197.92	209.90	6.05
8	194.25	205.83	5.96
9	198.34	208.99	5.37
10	197.32	208.92	5.88
11	196.57	204.59	4.98
12	199.29	210.13	5.44
13	200.33	212.15	5.90
14	197.19	211.36	7.19
15	201.36	213.12	5.84
16	198.15	213.20	7.60
17	194.77	206.95	6.25
18	193.26	203.86	5.48
19	198.54	208.77	5.15
20	197.26	208.08	5.49
Average path length covered	**197.86**	**209.22**	**5.75**

Table 5.7: Comparison of time taken by a single robot during simulation and experimental (Shown in Figures 5.5 and 5.11) using CS navigational algorithm.

No. of Runs	**Time taken by the robot during simulation (in 'sec')**	**Time taken by the robot during experiment (in 'sec')**	**% of error**
1	15.49	16.39	5.81
2	15.32	16.24	5.99
3	14.91	15.67	5.13
4	15.33	16.10	4.99
5	14.89	15.78	5.94
6	15.03	15.84	5.43
7	14.72	15.56	5.66
8	15.05	15.97	6.12
9	15.56	16.40	5.37
10	15.28	16.54	8.25
11	14.99	15.79	5.30
12	14.82	15.76	6.35
13	14.77	15.69	6.21
14	14.72	15.56	5.70
15	15.35	16.51	7.53
16	15.42	16.44	6.62
17	15.22	16.00	5.14
18	15.18	16.12	6.16
19	14.81	15.66	5.74
20	14.75	15.61	5.84
Average time taken by the robot	**15.08**	**15.98**	**5.96**

Table 5.8: Comparison of time taken by a single robot during simulation and experimental (Shown in Figures 5.6 and 5.12) using CS navigational algorithm.

No. of Runs	**Time taken by the robot during simulation (in 'sec')**	**Time taken by the robot during experiment (in 'sec')**	**% of error**
1	18.34	19.31	5.28
2	18.88	19.97	5.74
3	17.36	18.49	6.55
4	19.21	20.30	5.68
5	17.53	18.55	5.79
6	17.83	18.77	5.32
7	17.31	18.19	5.07
8	17.42	18.29	5.00
9	18.23	19.19	5.31
10	18.23	19.14	4.98
11	17.60	18.91	7.49
12	17.56	18.66	6.26
13	17.41	18.56	6.64
14	17.25	18.21	5.56
15	18.46	19.38	4.95
16	19.21	20.46	6.49
17	17.08	17.97	5.23
18	16.97	17.89	5.46
19	17.51	18.55	5.96
20	17.41	18.35	5.41
Average time taken by the robot	**17.84**	**18.86**	**5.71**

5.4.3 Comparison of the Developed CS Controller with other Navigational Controllers

In this section, a comparative study has been made between the proposed navigational controller and other intelligent controllers in simulation mode. The performances of the navigational controllers are discussed in the graphical mode. We have replicated the environment as stated by the authors. The size dimension of simulation platforms are considered as no. of units and each unit is in millimeter (mm).The proposed controller has been applied to the replicated environments as shown in Figures 5.13 and 5.14 respectively.

In the first case Wang et al. [165] and Mohamed et al. [174] have developed their navigation system based on the GA (Genetic algorithm) and PSO (Particle Swarm Optimization) techniques. The simulation results obtained from the discussed navigation system have been given in Figures 5.13(a) and 5.13(b) respectively. The result obtained from using current investigation is shown in Figure 5.13 (c).

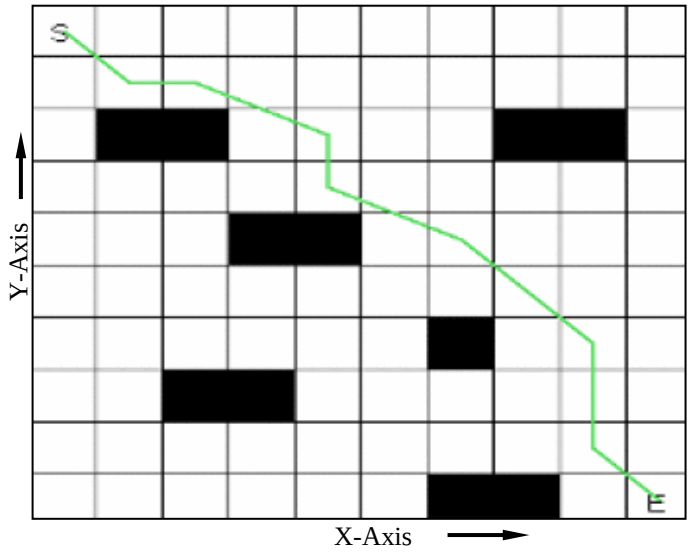


Figure 5.13(a) Simulation results from Genetic algorithm (GA) (Wang et al. [165]).

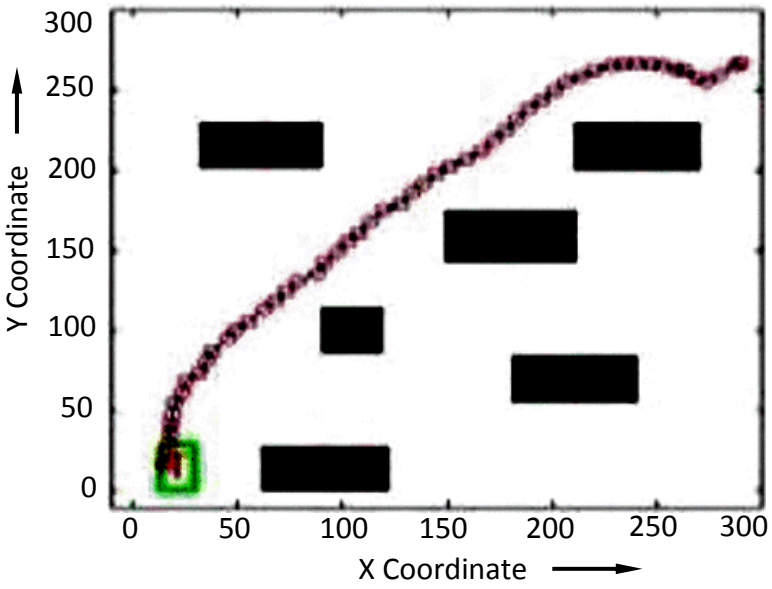


Figure 5.13(b) Simulation results from Particle swarm optimization (PSO) (Mohamed et al. [174]).

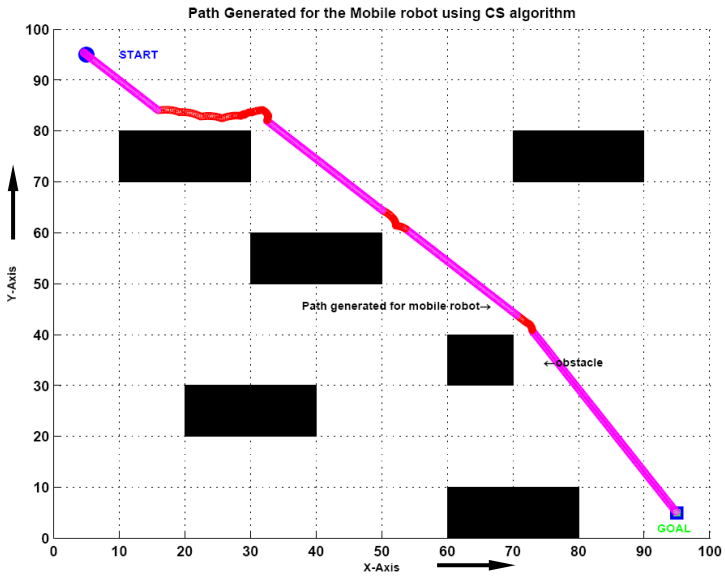


Figure 5.13(c) Simulation results obtained from current investigation.

Similarly, in the second case the Figure 5.14 (a) shows the simulation results obtained from Mohamed et al. [174] using PSO algorithm. Results obtained using the current investigation is shown in the Figure 5.14(b).

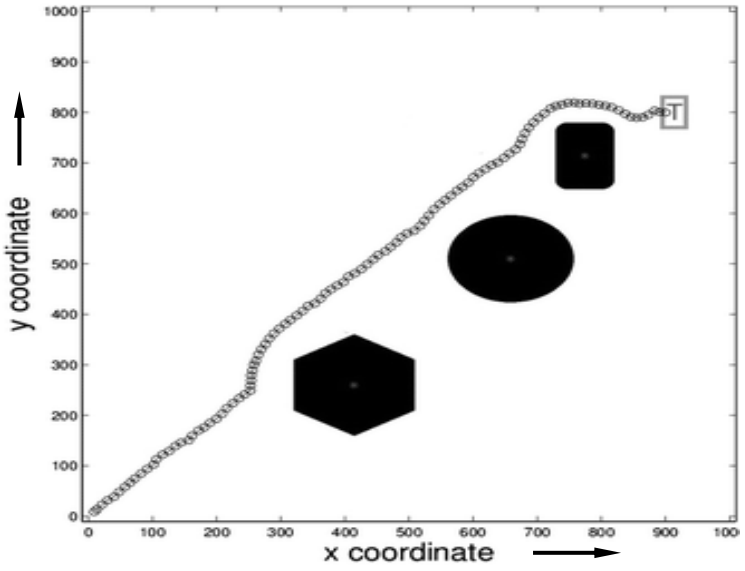


Figure 5.14(a) Simulation results from Particle Swarm optimization (PSO) (Mohamed et al. [174]).

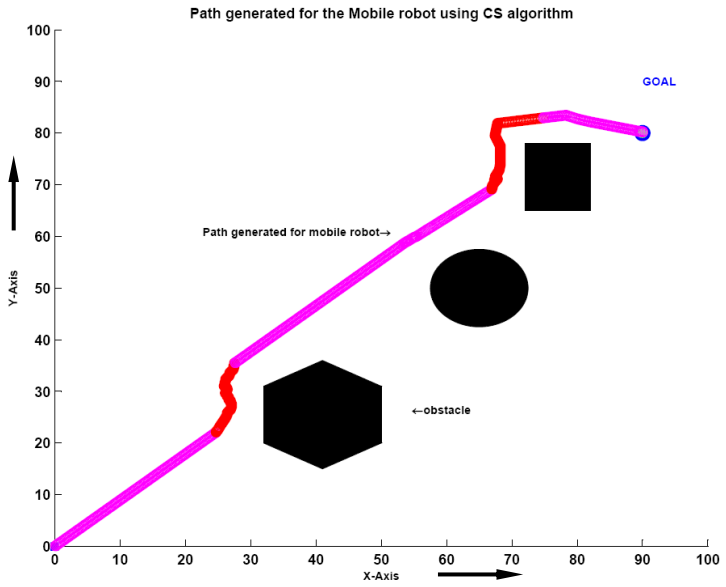


Figure 5.14(b) Simulation results obtained using current investigation.

Table 5.9: Comparison of simulation results in terms of path length.

SL. No.	Environment types	Path length using current navigation system (‘in ‘cm')	Path length from the reference model (‘in ‘cm')	Percentage of deviation
1	Obstacle avoidance behavior by the robot Figures 5.13(a) and 5.13(c)	6.39	7.21	11.37
2	Obstacle avoidance behavior by the robot Figures 5.13(b) and 5.13(c)	6.39	6.66	4.05
3	Obstacle avoidance behavior by the robot 5.14(a) and 5.14 (b)	6.44	6.63	2.86

From the above comparative simulation study, it has been clearly observed that the proposed navigational path planner provides better path trajectory in terms of smoothness compared to other navigational methods, and also it can be efficiently implemented to drive a mobile robot in a cluttered terrain. The comparison of simulation results in terms of path length (in cm) covered by the robot to reach target is presented in the Table-5.9.

5.5 Summary

This chapter has described a novel methodology for a fully autonomous mobile robot navigating in an unknown or partially known environment containing static obstacles. The following salient features are concluded based on the simulation and experimental results.

- A new objective function has been formulated between the robot and the target and obstacles, which satisfied the conditions of obstacle avoidance and target-seeking behaviour of robot present in the terrain.
- Depending upon the objective function values in the swarm, the robot avoids obstacles and reaches the target successfully.
- The developed path planning strategy has been validated through simulations as well as experiments, which demonstrate the ability of the proposed navigation system. The results are in good agreement. The average percentages of errors are found to be

within 6% for both path length and time taken by the robot to reach target and tabulated in Tables 5.5-5.8 respectively.

- The results from proposed navigation system have been compared with other navigation techniques such as GA (Genetic algorithm) and PSO (Particle Swarm Optimization). From the comparative simulation study, it has been clearly seen that the proposed path planner has the ability to navigate the robot successfully in unknown environments with optimized path length. The percentage of deviation has been depicted in the Table 5.9.
- Finally, it is concluded that by using this nature inspired algorithm the navigation performance in unknown environments can be greatly improved.

In the next chapter a new biologically inspired algorithm based on colonizing property of weeds has been applied for the path optimization problem of a mobile robot.

CHAPTER-6

6. ANALYSIS OF INVASIVE WEED OPTIMIZATION ALGORITHM FOR MOBILE ROBOT NAVIGATION

This chapter presents a population based meta-heuristic algorithm for solving the navigational problem of a mobile robot in unknown or partially known environments containing static obstacles. This meta-heuristic algorithm named Invasive weed optimization (IWO) has been successfully implemented in the path analysis problem of autonomous mobile robot.

6.1 Introduction

Recently more researchers have attracted towards the swarm intelligence based algorithms such as: Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Artificial Bee Colony (ABC), Harmony search, Cuckoo Search (CS) etc. These research techniques are important branches of Artificial Intelligence (AI) and successfully applied in many areas of engineering. Path planning for mobile robots in unknown cluttered environments is one of the important issues in the robotics field. The biologically nature inspired optimization techniques such as Genetic Algorithm (GA), ACO, Artificial Immune System (AIS), Bacteria Foraging Optimization (BFO), and PSO have been employed for path planning of mobile robots as discussed in (Chapter-2). Most of these metaheuristic algorithms are inspired from the successful features of the underlying biological, physical or sociological systems. These algorithms have gained acceptance because of their ability and efficiency of searching an optimal solution in a specified problem area. However, to some extent, theses algorithms have flaws in large calculations, and complicated coding. The feasibility and robustness of ACO are restricted to the grid division. In the current investigation a new biologically inspired algorithm [209] based on the colonizing property of weeds has been applied for the path optimization problem of a mobile robot. The path optimization strategy is regarded as an optimization problem to search an optimal position of the mobile robot step by step. Here the optimization concept considers the distances between the robot to obstacles and target in order to find the suitable position and for avoiding collisions. This algorithm has also shown successful results in various fields of engineering [216-220]. Due to its broad

range of applications, we are motivated to employ this algorithm for solving the robot navigational problem.

6.2 Outline of Invasive Weed Optimization Algorithm

Invasive weed optimization (IWO) algorithm was first introduced by Mehrabian and Lucas in 2006 [209]. This novel stochastic optimization algorithm is based on the colonizing property of weeds. It has been observed that following the properties of the invasive weeds, leads to an effective optimization method.

The IWO algorithm is summarized as follows:

Initialize a population

A finite number of seeds are being randomly dispread over the defined search space. Each random seed's position is considered as an initial solution to the optimization problem.

Reproduction

In this stage, each individual seed produces flowering plants. Before the plant produces new seeds, they are ranked according to their fitness values. In another sentence, the number of seeds produced by each flowering plant depends on its fitness function values and increases linearly from the minimum possible seeds production to its maximum. The plant will produce seeds based on the colony's lowest and highest fitness to make sure the increase is linear. Figure 6.1 shows the reproduction procedure. This stage adds an important property to the weed algorithm by allowing all of the plants to participate in the reproduction process.

Spatial dispersal

The produced seeds in the previous step are being randomly scattered over the define search space by a normal distribution with mean equal to zero (or location of the producing plants) and varying standard deviation. In simulation, the standard deviation can be expressed by

$$\sigma_{iter} = \left(\frac{iter_{\max} - iter}{iter_{\max}} \right)^{n} \left(\sigma_{initial} - \sigma_{final} \right) + \sigma_{final} \tag{6.1}$$

where $iter_{max}$ is the maximum number iterations. $\sigma_{initial}$ and σ_{final} are defined initial and final standard deviations respectively, and n is the non-linear modulation index.

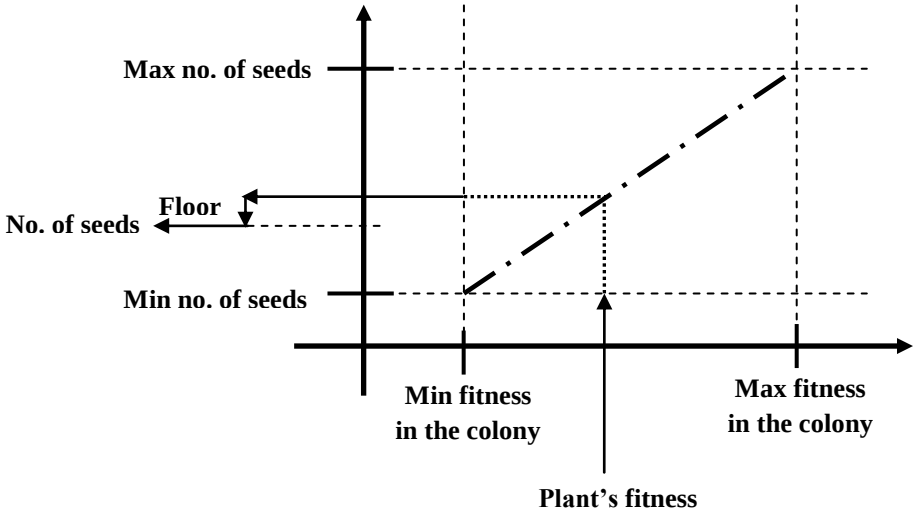


Figure 6.1 Seed production procedure in a colony of weeds [209].

Competitive exclusion

In the above step, all seeds have found their positions in the defined search area. The newly generated seeds grow to produce plants, and they are ranked together with their parents on the basis of fitness values. The flowering plants having lower fitness values in the colony are discarded to reach the maximum number of allowable populations (plants) in the colony P_{max}. The process is repeated on the reproduction stage until the maximum iteration is reached, or fitness criterion is met.

Begin;

Generate an initial random population of N solutions;

for iter=1 to the maximum number of iterations;

Determine the fitness function value of each individual;

Calculate maximum and minimum fitness in the colony:

For each individual weed ($w \in N$);

Determine number of seeds of w, corresponding to its fitness;

Randomly distribute the generated seeds over the problem search space with normal distribution around the parent plant (w);

Add the produced seeds to the initial solution set, N;

end;

If $N > N_{max}$;

Arrange the population N in descending order of their fitness;

Sort the population of weeds with smaller fitness until $N = N_{max}$;

end If;

Next iter

end;

Figure 6.2 Pseudo code for IWO algorithm.

6.3 Problem Formulation for Robot Path Planning using IWO Algorithm

The online path planning strategy of the mobile robot using Invasive weed optimization (IWO) algorithm is discussed in the present study. The proposed algorithm is based on seeds randomly distributed in front of a robot. The path optimization strategy is regarded as an optimization problem to search an optimal position of the mobile robot step by step. Here the optimization concept considers the distances between the robot to obstacles and target in order to find out the suitable position and to avoid collisions. The development of navigation system for mobile robot is one of the most important issues in the robotic research field. Recently nature inspired metahuristic algorithms have received considerable attention for solving the navigational problem for the mobile robot. Here we implement the IWO algorithm to find the optimal/shortest path for the mobile robot in highly cluttered environments. Firstly, we introduce an objective function equation based on the distances between the robot to obstacles and target, which satisfies the criteria of avoid collisions with obstacles and reaches the target behaviour of the robot. Based on the individual objective function value of each plant in the colony, the robot avoids collisions with obstacles and finally reaches the target. The plant having minimum objective function value is selected to be the next optimal position for the robot. This process will be continued until the robot reaches the target. When the robot does not detect any hurdles in its target path, then it will travel towards the designed target. Then it is not necessary to employ any navigational optimization techniques to travel the robot within its environment. The concept of the robot path planning using proposed strategy is given in Figure 6.3.

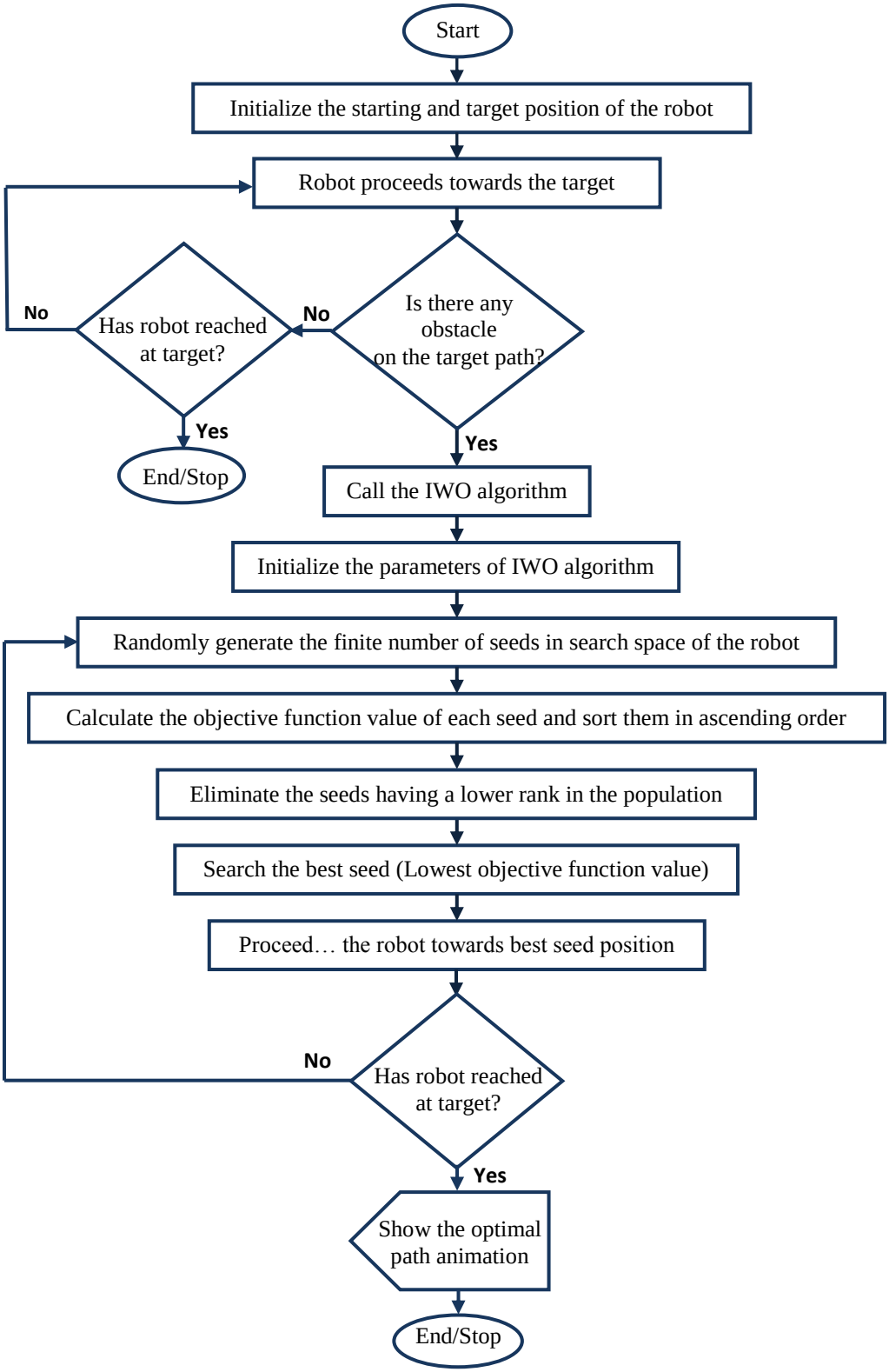


Figure 6.3 Flowchart of the proposed IWO algorithm for robot path planning.

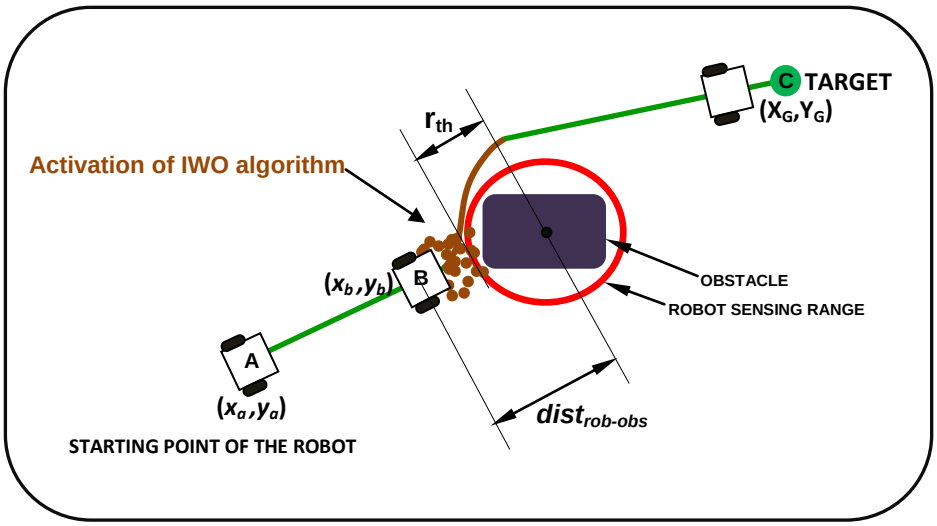


Figure 6.4 Illustration of the IWO algorithm for searching of optimal point.

Let the robot is at start point 'A' and moves to the target point 'C' in the environment shown in Figure 6.4. The robot aim is to reach the target with avoiding obstacles present in the environment. Here a robot searches its optimal position by using IWO algorithm. The current position of the robot is marked as (x_a, y_a) and the robot freely moves towards to the position B (x_b, y_b) without implementing any navigational technique. When the sensor detects an obstacle at position 'B' the robot uses its own IWO algorithm to search the next optimal position, which maintains the maximum safe range of the obstacles and minimum distance from the target. In the IWO algorithm, each seed/agent represents the one possible optimal position of the mobile robot.

The current position of the robot is marked by (x_b, y_b) and the possible positions of the robot are denoted by (x_s, y_s), $S=1,2,\ldots n,$ where n is the selected number of virtual position i.e., population number. Then the robot searches its optimal position from the possible locations in the searching area using IWO algorithm. From the population, the seed/weed having a minimum objective function value (the optimal location) is treated as the next optimal position for the robot.

6.3.1 Formulation of the Objective Function using the IWO Algorithm

The objective function for the robot path planning depends mainly on two rules:

- Avoid collision with obstacles
- Reach to target behaviour

a) Objective function for avoiding collision with obstacles

The objective function for obstacle avoidance is used to avoid collision with obstacles (such as obstacles, walls, etc.). The objective function is formulated based on the distance information between the robot and obstacles, and is defined as

$$f_{obstacle} = \begin{cases} 0, & \left\| S_i - obs_j \right\| > r_{th} \qquad j = 1,2,...p \\ \dfrac{1}{\left\| S_i - obs_j \right\|} - \dfrac{1}{r_{th}} & \left\| S_i - obs_j \right\| \le r_{th} \qquad j = 1,2,...p \end{cases} \tag{6.2}$$

where S_i represents the position of the i^{th} individual seed (or the possible optimal position for the robot), obs_j represents the center of the j^{th} obstacles, r^{th} represents the threshold value between the robot to obstacles and p represents the number of obstacles. Here our aim is to minimize the objective function value $f_{obstacle}$. In order to avoid collision with obstacles, the distance between the robot to obstacles should be more or equal to the threshold value (r_{th}). When the robot is very nearer to the obstacles j, the objective function results in a non-zero value, otherwise a zero objective value is obtained.

Note: Here we consider the nearest obstacle to the robot. The nearest obstacle to the robot can be calculated by the following expression:

$$dist_{rob-obs_j} = \min \left\| rob_b - obs_j \right\| \qquad j = 1,2,...p \ .$$

where, rob_b represents the robot position at point 'B'. The position of the robot will be varied according to the environmental scenario.

b) Objective function for reach to target behaviour

The objective function for the target is used to reach the target with minimum path length. The objective function is designed based on the distance information between robot to target, defined as,

$$f_{target} = \left\| S_i - T \right\| \tag{6.3}$$

where S_i represents the position of the i^{th} individual seed (or the possible optimal position for the robot), the T represents the target position. Here our objective is to minimize the target function (f_{target}), which implies to minimize the distance between the possible optimal positions of the robot to target.

So the total objective function for optimizing the robot's path planning strategy can be written as,

$$f_n = \alpha f_{obstacle} + \beta f_{target} \tag{6.4}$$

where α and β denote the controlling or weight parameters that influence the robot trajectory and f_n is the fitness value of the N number population. Here we choose the controlling parameters by trial and error methods.

When an obstacle is not detected in the robot's sensing range, $f_{obstacle}$ is zero and in this situation the robot directly moves towards the target. When the robot detects an obstacle, $f_{obstacle}$ gets a nonzero value. In this condition after performing the IWO algorithm, all the seeds are sorted in ascending order of the objective function value and the seed having lowest objective function value is treated as the new position of the robot. Thus, the same process is repeated for several iterations until it satisfies the convergence criteria or the robot avoids collision with obstacles and reaches its target effectively.

Steps of IWO algorithm for path planning strategy of mobile robot:

Step-1: Initialize the starting and target position of the robot.

Step-2: Proceed the robot towards the target until it will be stuck by an obstacle.

Step-3: When an obstacle is obstructing the target path call IWO algorithm.

Step-4: Initialize, the population of seeds in the search space of the robot, each of which represents one trial possible optimal position for the robot.

Step-5: Evaluate the fitness (f_n) of the population, sort the population in ascending order, this process continues until the maximum number of plants is produced. Eliminate the plants having a lower rank in the population.

Step-6: Sort the population according to the objective function value, the seed/agent having minimum objective function value is selected as the new optimal position for the robot.

Step-7: Move the robot towards the new optimal position.

Step-8: Repeat the step-4 to step-7 until the robot avoids all obstacles and reaches its target.

6.4 Demonstrations of the IWO based Navigational Controller

In this section, the results are illustrated, and the efficiency of the proposed algorithm is analysed. The simulation results are validated with experimental results to show the effectiveness of the developed path planner, and they are in good agreement. The proposed navigation system is also compared with the other path planning algorithms. Finally, the overall efficacy of the IWO path planning system is analysed based on the path length, the smoothness of the path, the total travel time and the planning success rate.

6.4.1 Simulation Result and Discussions

Several simulation results are illustrated to demonstrate the performance and effectiveness of the proposed approach. In the simulation environment, a Gaussian cost function is virtually represented by each obstacle and a 2D sphere function is assigned to the target. Four different scenarios have been framed by arranging the obstacles in different positions for IWO based navigational controller to show the ability of proposed control technique. The simulation results obtained using the proposed algorithm has been shown in Figures 6.5 to 6.8. The size dimension of simulation platforms are considered as no. of units and each unit is in millimeter (mm).The best parametric values of proposed algorithm have been selected after performing a series of simulation experiments on partially or totally unknown environments, and it is tabulated in the Table-6.1. It has been observed that some parameters in the proposed algorithm are almost insensitive to select them, whereas other parametric values affect the performance of the proposed navigation system. It has been observed that there are mainly three parameters affecting the convergence of the proposed algorithm, the initial standard deviation ($\sigma_{initial}$), the final standard deviation (σ_{final}) and the non-linear modulation index (n). The simulation results considering different choice parameters are illustrated in Figures 6.9 and 6.10 respectively. The path lengths covered by the robot during simulation have been presented in the Table-6.2 and 6.3 respectively.

Table-6.1 Details of IWO parameter values used for the robot path planning optimization problem.

Symbol	**Description**	**Value**
N	Number of initial population	20
$iter_{max}$	Maximum number of iterations	300
p_{max}	Maximum number of plant population	20-30
S_{max}	Maximum number of seeds	5-10
S_{min}	Minimum number of seeds	0
n	Non-linear modular index	3
$\sigma_{Initial}$	Initial value of standard deviation	3-4
σ_{Final}	Final value of standard deviation	0-0.001
α	Controlling parameter-1	1
β	Controlling parameter-2	$1x10^{-06}$

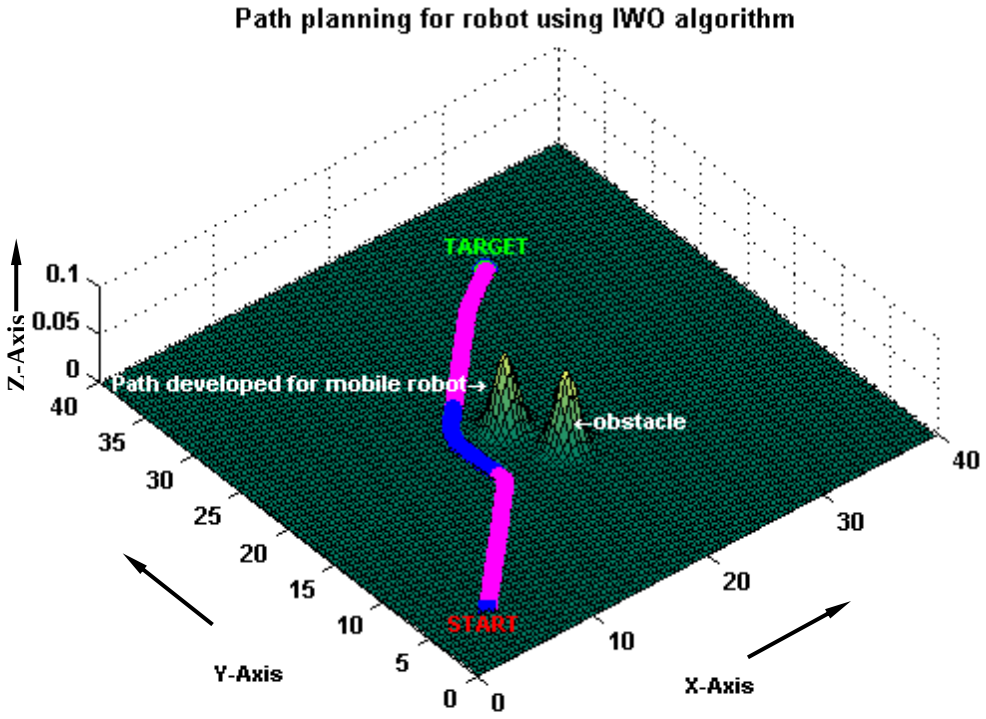


Figure 6.5 Single robot avoiding obstacles to reach target using IWO algorithm.

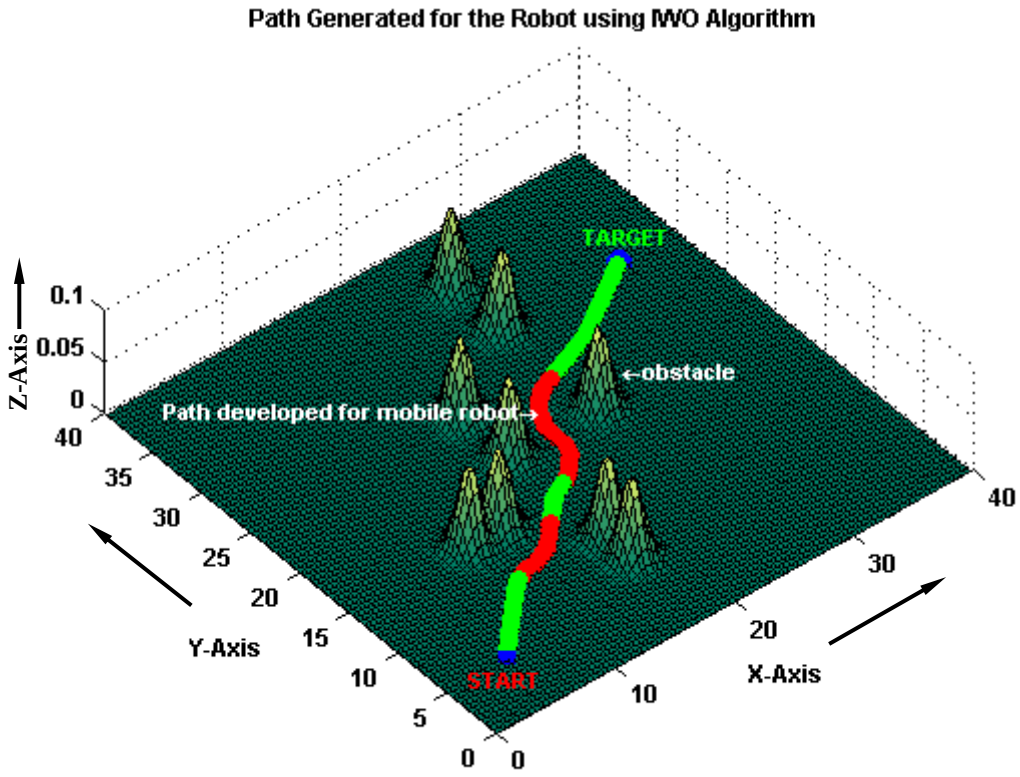


Figure 6.6 Single robot escaping from a trap condition to reach target using IWO algorithm.

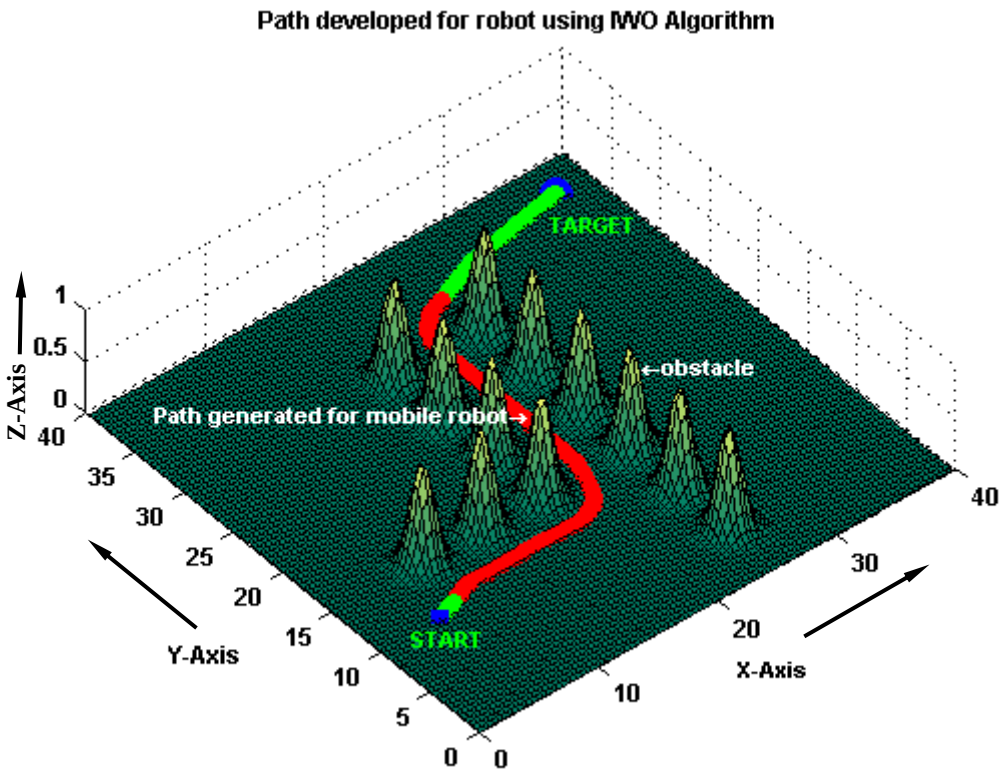


Figure 6.7 Single robot passing through a narrow corridor to reach target using IWO algorithm.

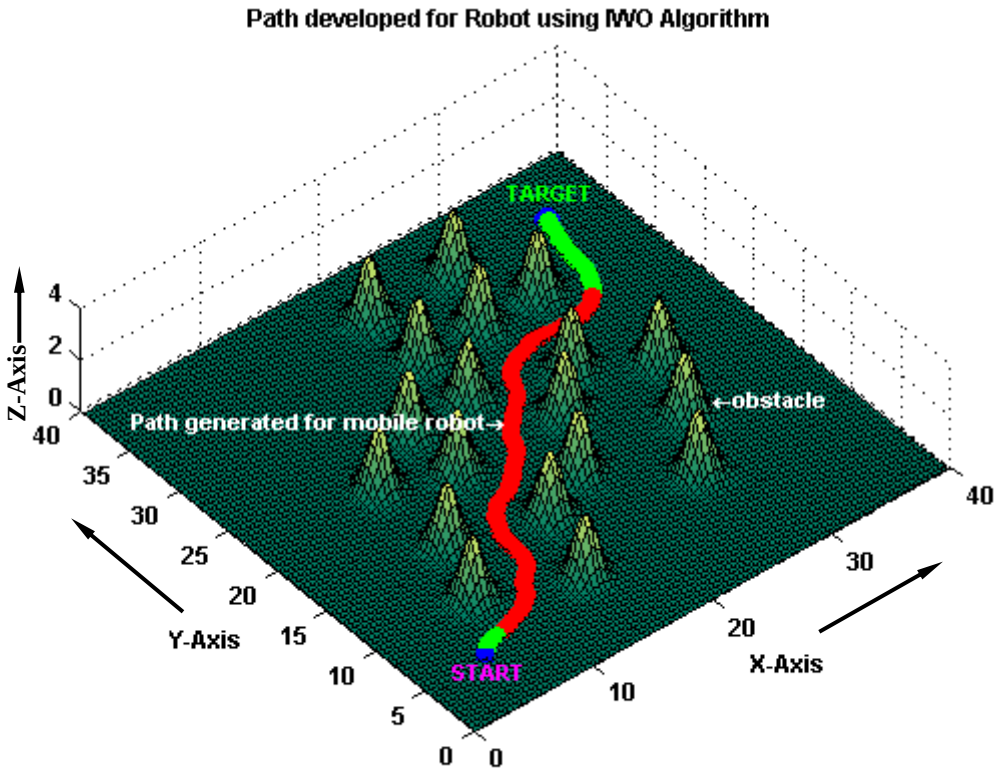


Figure 6.8 Single robot navigating inside a maze environment to reach target using IWO algorithm.

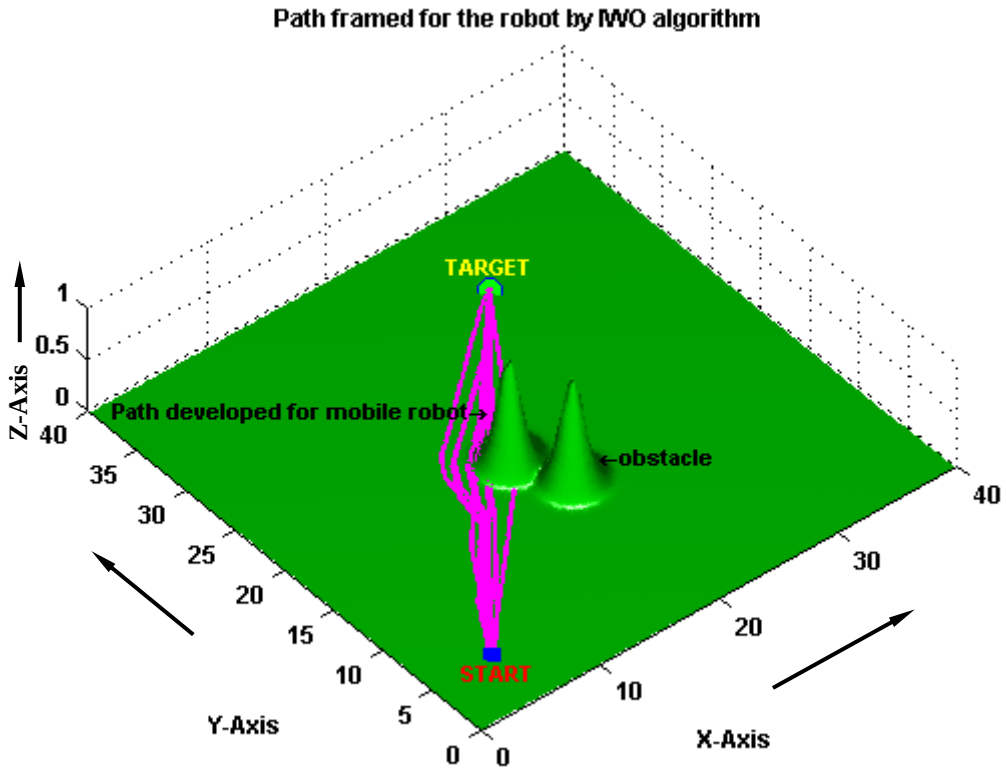


Figure 6.9 Path generated for the mobile robot at different choice parameters of IWO.

Table-6.2 Path covered by the robot during simulation by considering different values of $\sigma_{initial}$ and σ_{final} of IWO algorithm. (Shown in Figure 6.9)

Sl. No.	$\sigma_{initial}$	σ_{final}	n	Path length covered by the robot (in 'cm')	Avoid Collision (Yes/No)
1	3.5	0.005	3	6.39	Yes
2	3.6	0.006	3	6.29	Yes
3	3.7	0.007	3	6.21	Yes
4	3.8	0.008	3	6.18	Yes
5	***3.9***	***0.009***	***3***	***6.16 (optimal)***	***Yes***
6	4	0.01	3	6.13	No
7	4.1	0.02	3	6.12	No
8	4.2	0.03	3	6.04	No
9	4.3	0.04	3	6.00	No
10	4.4	0.05	3	5.96	No

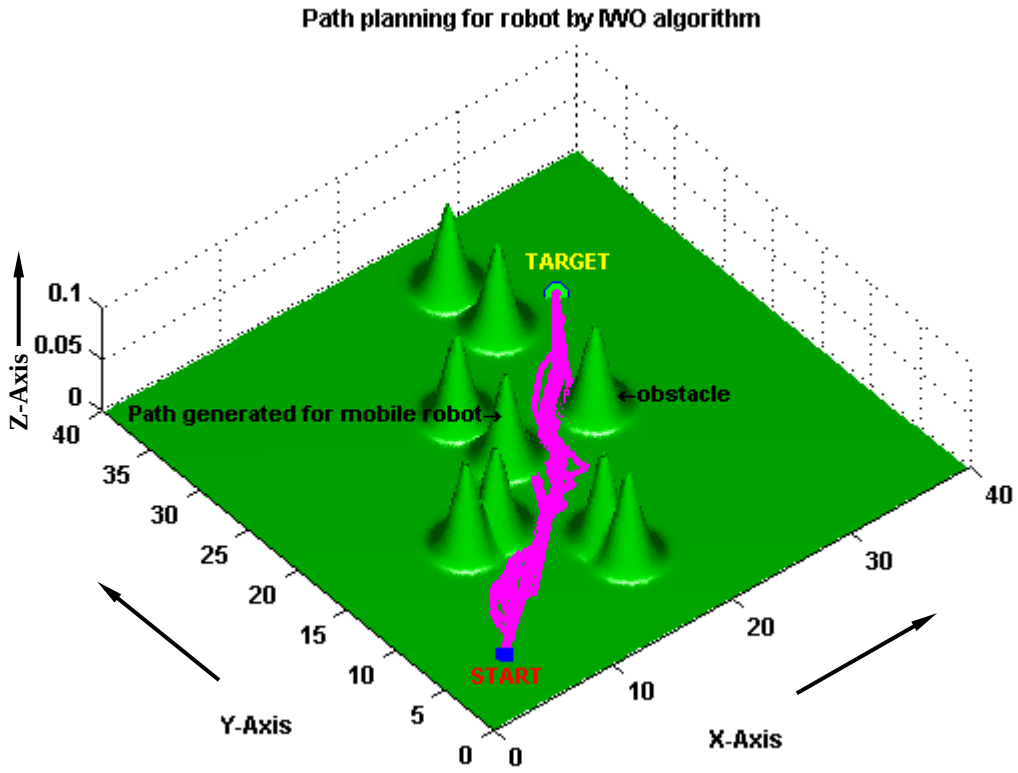


Figure 6.10 Path generated for mobile robot at different choice parameters of IWO.

Table-6.3 Path covered by the robot during simulation by considering different values of $\sigma_{initial}$ and σ_{final} of IWO algorithm. (Shown in Figure 6.10)

Sl. No.	**$\sigma_{initial}$**	**σ_{final}**	**n**	**Path length covered by the robot (in 'cm')**	**Avoid Collision (Yes/No)**
1	3.5	0.005	3	6.07	No
2	3.6	0.006	3	6.08	No
3	3.7	0.007	3	6.09	No
4	3.8	0.008	3	6.11	No
5	3.9	0.009	3	6.12	No
6	***4***	***0.01***	***3***	***6.13 (optimal)***	***Yes***
7	4.1	0.02	3	6.17	Yes
8	4.2	0.03	3	6.21	Yes
9	4.3	0.04	3	6.20	Yes
10	4.4	0.05	3	6.23	Yes

6.4.2 Comparison of the Developed IWO Path Controller with other Navigational Controllers

In this part, a comparative study has been carried out between the proposed navigational controller and other intelligent navigational controllers in simulation mode. First, we have replicated the environment as stated by other authors and implemented the developed IWO algorithm to the replicated environments to demonstrate the efficacy of the proposed algorithm. The X and Y axes of the simulation graph are considered as a sized of 30x30 units and each unit is equal to 20mm.

1. Liang et al. [227] have discussed the path planning strategy of a mobile robot using Self Adaptive Bacteria Foraging Optimization Algorithm (SABFO). In this proposed navigational system, a robot mimics the behavior of bacteria to find an optimal collision free route between the start point and a goal point in the environment surrounded by obstacles at various locations. The simulation results obtained using SABFO navigation system is demonstrated in Figure 6.11 (a). The results obtained using current investigation is presented in the Figure 6.11(b).

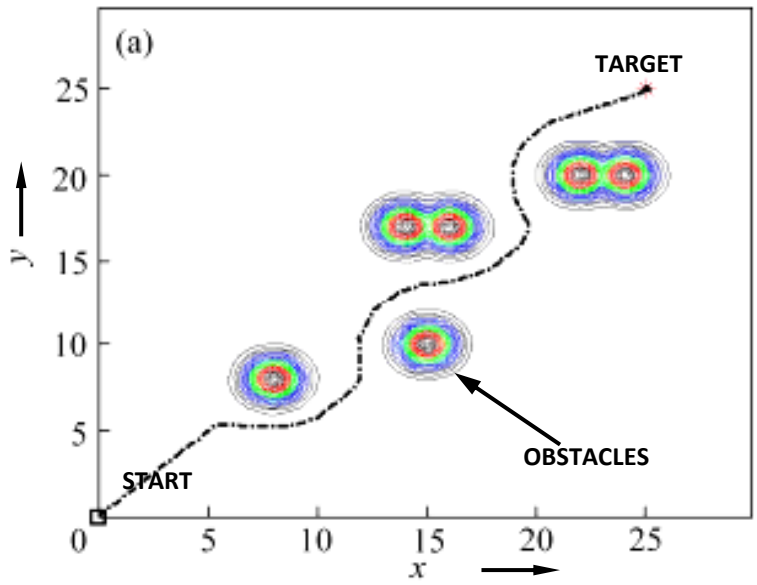


Figure 6.11 (a) Path generated for the robot using SABFO algorithm (Liang et al. [228]).

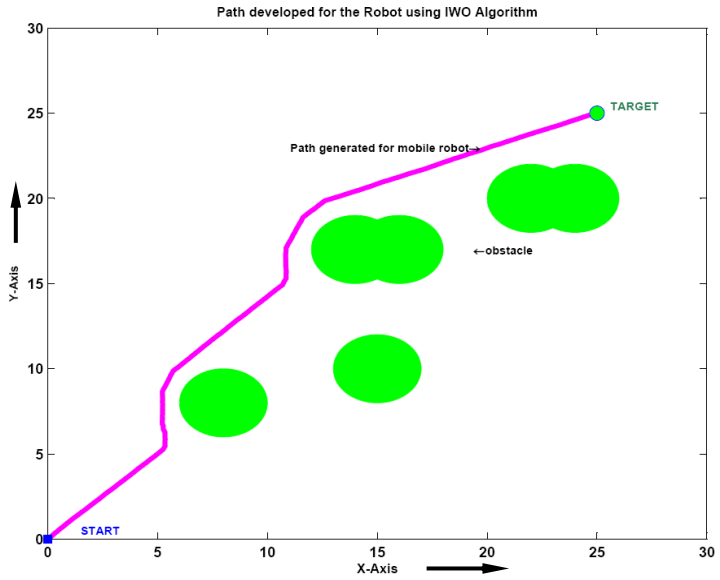


Figure 6.11 (b) Simulation path obtained using current investigation.

2. Mobile robot navigation based on Ant Colony Optimization (ACO) approach has been presented by Cen et al. [181]. The information of workspace constraints and route length are integrated into the fitness function which is calculated by neural network, the path nodes are viewed as an ant, so using an ACO algorithm, and best path has been selected. The simulation results using ACO algorithm is shown in Figure 6.12(a) and also compared with the Particle Swarm Optimization (PSO) algorithm to show the feasibility of the proposed navigation system. The simulation results obtained using current investigation is shown in Figure 6.12(b).

3. Route planning for a mobile robot using an adaptive genetic algorithm has been discussed by Wang et al. [165]. This algorithm focuses on the automatic tuning of crossover probability and mutation probability with variable environmental constraints. The simulation results obtained using adaptive genetic algorithm is presented in Figure 6.13(a). The simulation graph obtained using IWO based navigational controller is shown in Figure 6.13(b).

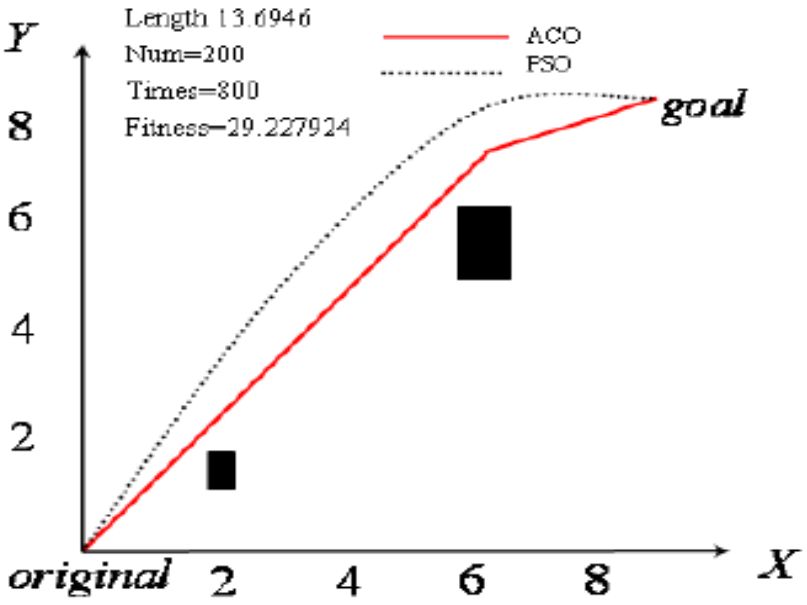


Figure 6.12(a) Path generated for single robot using ACO algorithm (Chen et al. [181]).

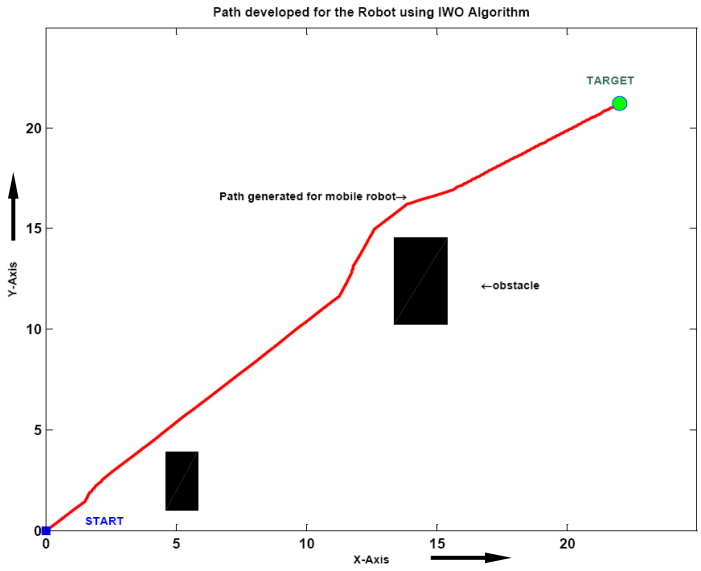


Figure 6.12(b) Path generated for single robot using IWO.

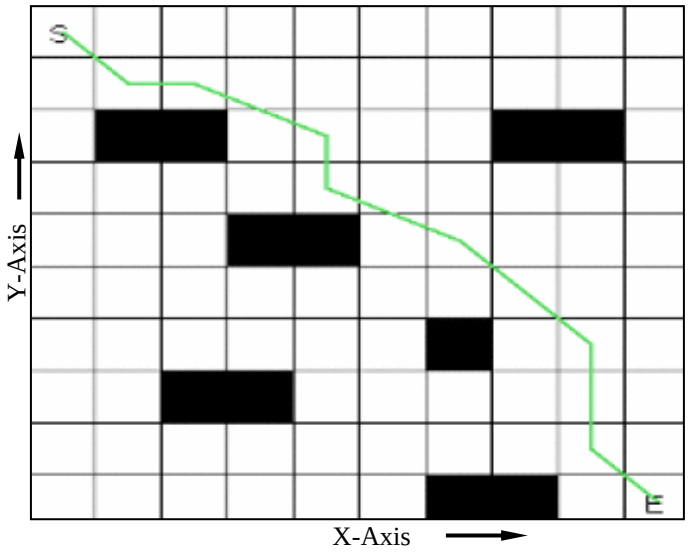


Figure 6.13(a) Path generated for single robot using Adaptive GA (Wang et al. [165]).

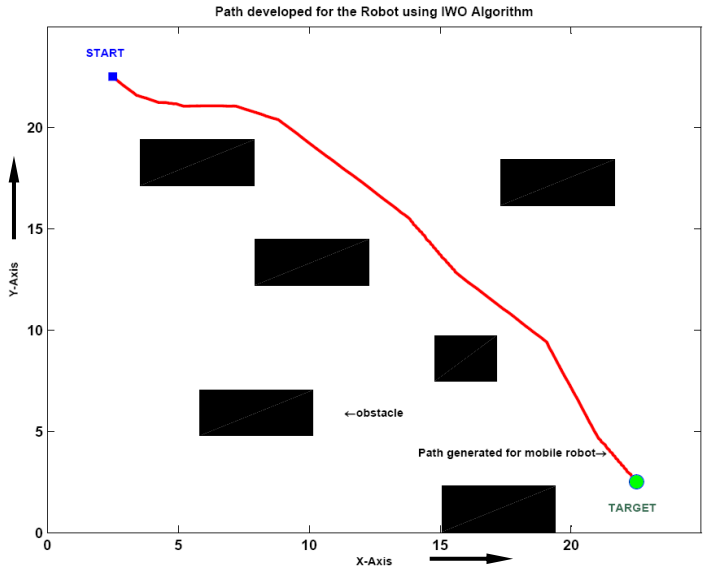


Figure 6.13(b) Path generated for single robot using current investigation.

From the comparative study, it has been clearly noticed that the proposed navigational algorithm delivers better path trajectory in terms of smoothness compared to the other discussed navigational systems and also it can be efficiently implemented to drive a mobile robot in a hurdles environments.

Table-6.4 (Comparison of simulation results in terms of path length)

SL. No.	**Environment types**	**Path length using current navigation system (‘in ‘cm')**	**Path length from the reference model (‘in ‘cm')**	**Percentage of deviation**
1	Robot inside a maze environment, Figures 6.11(a) and 6.11(b)	7.10	7.53	5.71
2	Obstacle avoidance behavior by the robot Figures 6.12(a) and 6.12(b)	6.68	7.04	5.11
3	Obstacle avoidance behavior by the robot 6.13(a) and 6.13 (b)	6.47	7.21	10.26

6.4.3 Experiments with Real Mobile Robots

Experimental validation of simulation results of the developed IWO algorithm has been performed using Khepra-III mobile robot. The specification of the robot is given in Appendix-A. The experimental results are conducted in the robotic platform of sized 225cmx175cm in the laboratory. The programming languages for the robot are used as MATLAB and C++. Three different comparative studies are discussed in the above section, which are already verified by the simulation, and have been verified experimentally (Figures 6.14-6.16) to prove the effectiveness of the proposed approach.

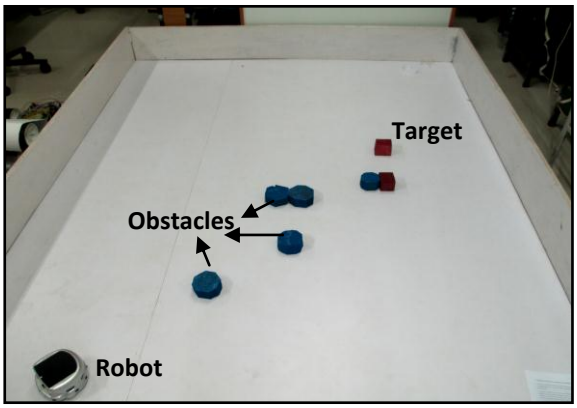


Figure 6.14(a)

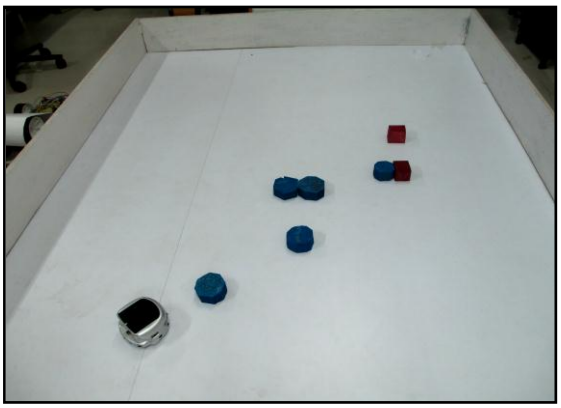


Figure 6.14(b)

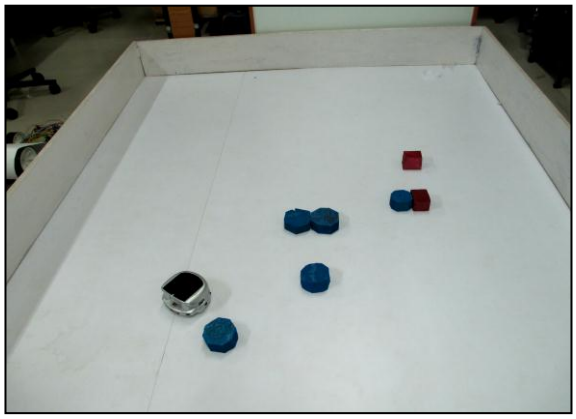


Figure 6.14(c)

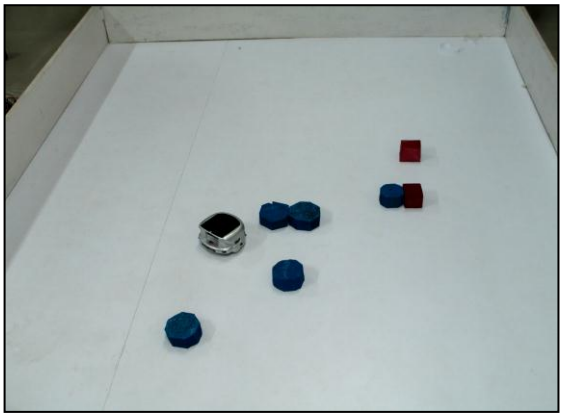


Figure 6.14(d)

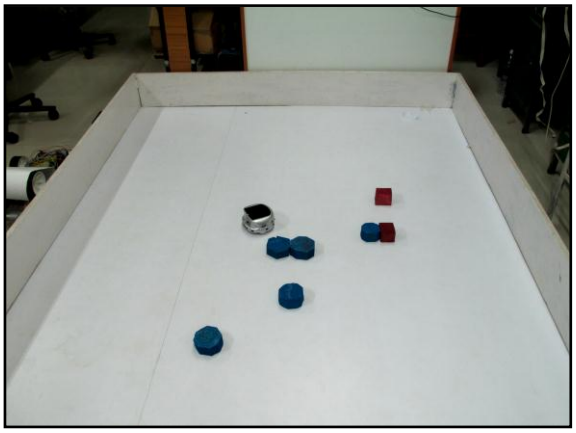


Figure 6.14(e)

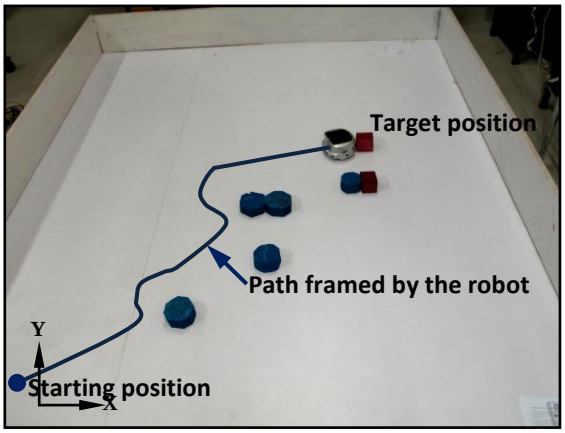


Figure 6.14(f)

Figure 6.14 Experimental results for navigation of mobile robot in the environment shown in Figure 6.11(b).

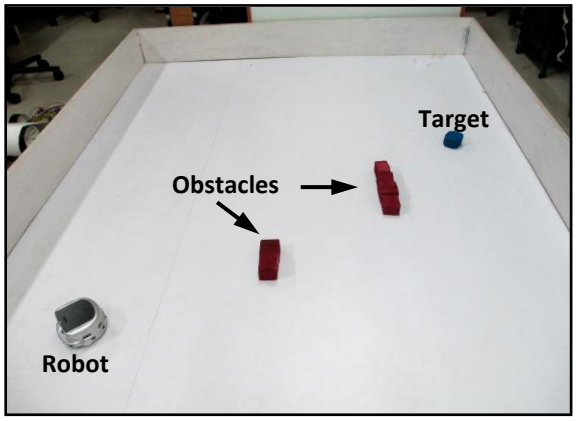


Figure 6.15(a)

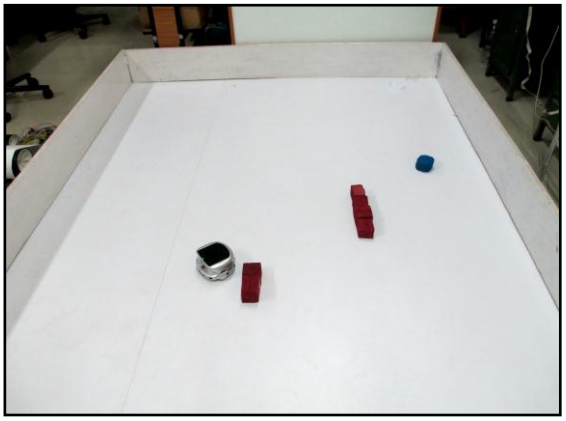


Figure 6.15(b)

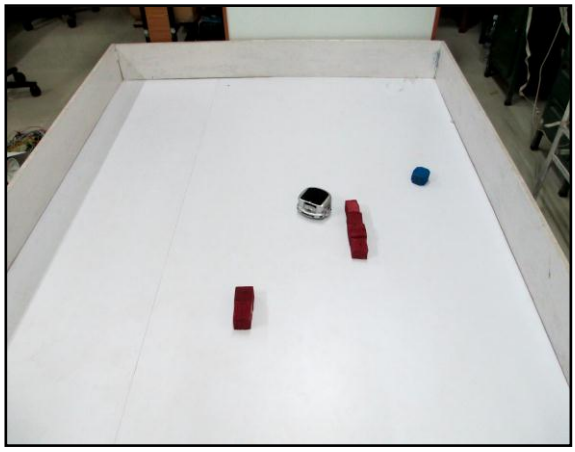


Figure 6.15(c)

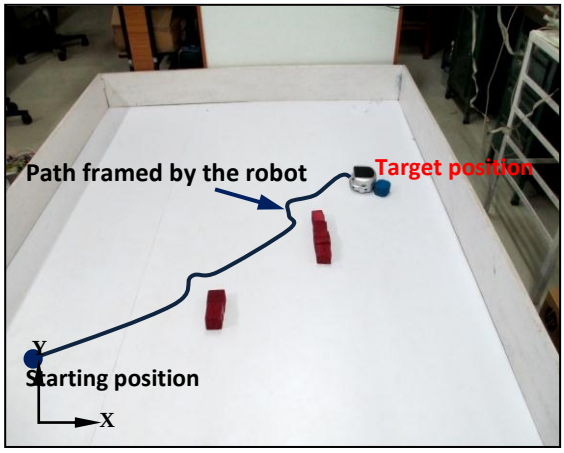


Figure 6.15(d)

Figure 6.15 Experimental results for navigation of mobile robot in the environment shown in Figure 6.12 (b).

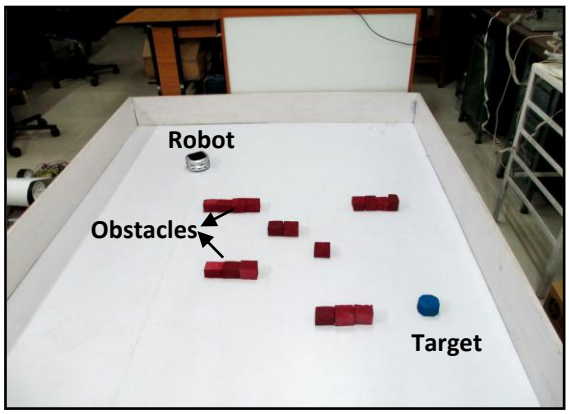


Figure 6.16(a)

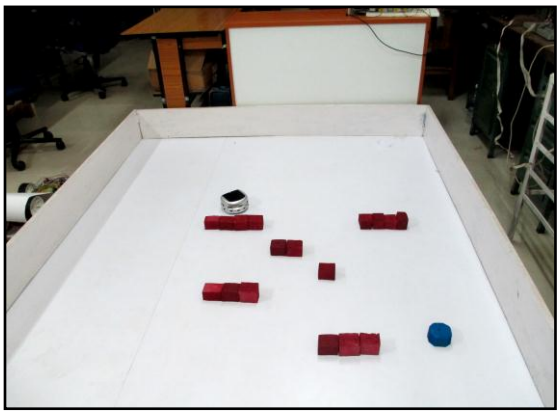


Figure 6.16(b)

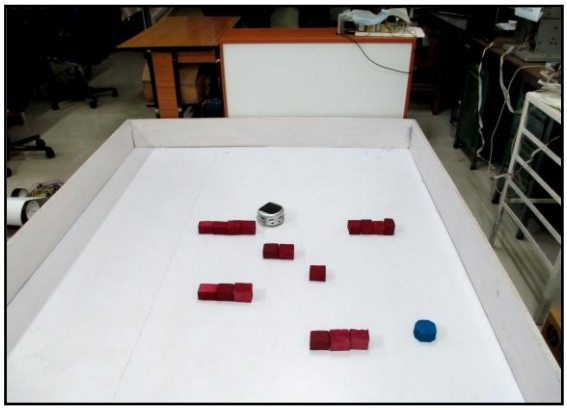


Figure 6.16(c)

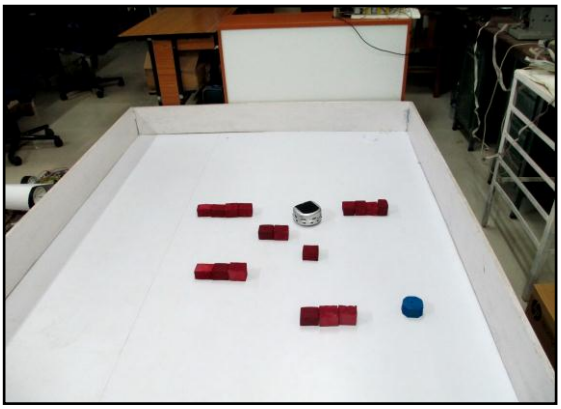


Figure 6.16(d)

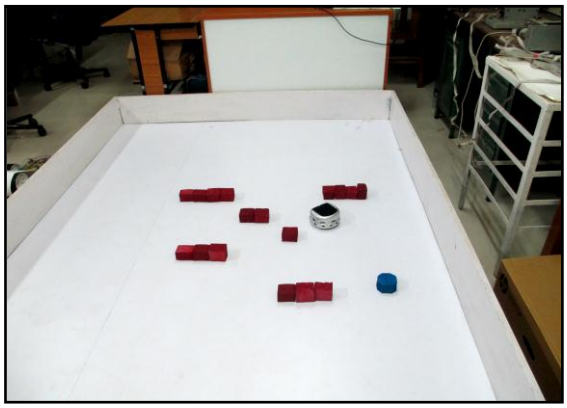


Figure 6.16(e)

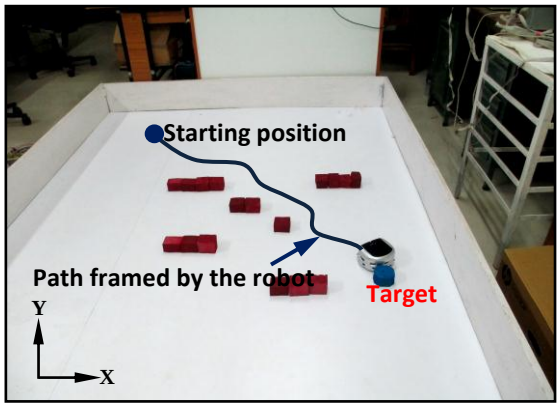


Figure 6.16(f)

Figure 6.16 Experimental results for navigation of mobile robot in the environment shown in Figure 6.13(b).

From the experimental study, it can be clearly observed that the mobile robot reaches their destination successfully without colliding with obstacles present in the environment. The optimal trajectory is generated with this developed algorithm and the path obtained from experiment follows closely those followed by the robot during simulation study.

Three different conditions (Figures 6.11-6.13) which are already verified in simulation mode have been verified experimentally (Figures 6.14-6.16) to show the effectiveness of the proposed navigational system. The mobile robot decides its path in the shortest trajectory to reach the desired goal by getting the optimized objective function values found from IWO algorithm. It has been concluded by comparing the results of the simulation as well as experimental study that, the path followed by the robot using the developed path planning algorithm can successfully reach the destination without hitting any obstacles present in the environment. The real time experimental investigation shows the effectiveness of the proposed navigational algorithm. The performance of the current navigation system has been checked on the basis of average path length (in‘cm’) and time (in‘sec’) taken by the robot to reach the goal. For calculation of path length and time taken, about 20 runs have been taken into consideration for proposed navigation system, which is presented in Tables-6.5 to 6.10 respectively and best results are presented in the simulation and experimental graph.

Table 6.5: Comparison of the path length during simulation and experimental results for a single robot shown in Figures (6.11(b) and 6.14) using IWO navigational algorithm.

No. of Runs	**Path length covered during simulation (in 'cm')**	**Path length covered during the experiment (in 'cm')**	**% of error**
1	168.98	180.70	6.94
2	170.46	179.65	5.39
3	169.70	177.69	4.71
4	169.59	178.72	5.39
5	170.72	178.06	4.90
6	168.77	178.81	5.95
7	170.23	179.74	5.59
8	170.22	180.25	5.90
9	169.07	179.85	6.38
10	169.64	181.27	6.85
11	168.12	180.20	7.18
12	168.06	178.58	6.26
13	169.53	180.90	6.71
14	170.30	182.78	7.33
15	170.78	179.60	5.17
16	171.38	181.13	5.69
17	169.65	178.86	5.43
18	169.34	181.10	6.95
19	169.99	180.21	6.01
20	168.93	177.56	5.11
Average path length covered	**169.67**	**179.78**	**5.96**

Table 6.6: Comparison of the path length during simulation and experimental results for a single robot shown in Figures (6.12(b) and 6.15) using IWO navigational algorithm.

No. of Runs	**Path length covered during simulation (in 'cm')**	**Path length covered during the experiment (in 'cm')**	**% of error**
1	159.12	169.21	6.34
2	161.98	172.22	6.32
3	161.00	170.32	5.79
4	160.92	172.05	6.92
5	162.62	171.61	5.53
6	161.44	172.04	6.56
7	161.47	171.29	6.08
8	162.44	172.13	5.97
9	162.22	172.51	6.34
10	161.31	172.11	6.70
11	159.73	171.21	7.19
12	159.96	170.03	6.30
13	161.55	170.33	5.44
14	159.11	170.34	7.05
15	160.96	172.02	6.87
16	159.67	169.32	6.04
17	162.91	171.07	5.01
18	161.85	170.77	5.51
19	161.00	169.22	5.10
20	160.88	171.02	6.30
Average path length covered	**161.37**	**171.04**	**5.99**

Table 6.7: Comparison of the path length during simulation and experimental results for a single robot shown in Figures (6.13(b) and 6.16) using IWO navigational algorithm.

No. of Runs	**Path length covered during simulation (in 'cm')**	**Path length covered during the experiment (in 'cm')**	**% of error**
1	153.15	163.08	6.48
2	156.72	165.33	5.49
3	156.50	164.97	5.41
4	156.40	166.05	6.17
5	157.52	167.80	6.52
6	156.05	165.96	6.35
7	156.09	166.22	6.49
8	157.30	166.98	6.16
9	157.03	165.74	5.55
10	155.88	165.88	6.41
11	153.91	162.33	5.47
12	154.20	163.21	5.84
13	157.43	166.55	5.79
14	153.81	163.55	6.33
15	155.45	163.99	5.49
16	153.84	164.08	6.66
17	157.89	166.89	5.70
18	156.56	166.67	6.46
19	155.50	164.99	6.10
20	156.36	166.22	6.31
Average path length covered	**155.94**	**165.32**	**5.92**

Table 6.8: Comparison of time taken by a single robot during simulation and experimental results (shown in Figures 6.11(b) and 6.14) using IWO navigational algorithm.

No. of Runs	**Time taken by the robot during simulation (in 'sec')**	**Time taken by the robot during experiment (in 'sec')**	**% of error**
1	16.71	17.70	5.91
2	16.65	17.56	5.49
3	15.89	16.72	5.27
4	16.22	17.19	5.97
5	17.02	18.19	6.88
6	15.99	17.13	7.08
7	17.27	18.35	6.29
8	16.81	17.89	6.40
9	17.07	18.24	6.90
10	15.84	16.76	5.77
11	17.03	18.24	7.10
12	17.56	18.63	6.09
13	16.65	17.92	7.60
14	17.36	18.62	7.28
15	16.62	17.76	6.88
16	17.35	18.62	7.29
17	16.15	17.07	5.68
18	15.97	17.16	7.42
19	17.38	18.65	7.31
20	15.91	16.88	6.10
Average time taken by the robot	**16.67**	**17.76**	**6.54**

Table 6.9: Comparison of time taken by a single robot during simulation and experimental results shown in Figures (6.12(b) and 6.15) using IWO navigational algorithm.

No. of Runs	**Time taken by the robot during simulation (in 'sec')**	**Time taken by the robot during experiment (in 'sec')**	**% of error**
1	13.88	14.71	5.98
2	14.58	15.59	6.94
3	14.49	15.33	5.79
4	13.44	14.22	5.80
5	14.64	15.45	5.53
6	13.59	14.53	6.92
7	14.61	15.60	6.75
8	14.62	15.66	7.12
9	14.60	15.53	6.34
10	13.58	14.59	7.44
11	13.21	14.17	7.27
12	14.40	15.30	6.30
13	14.63	15.44	5.54
14	13.55	14.31	5.61
15	14.49	15.51	7.06
16	14.37	15.41	7.24
17	14.66	15.40	5.01
18	14.57	15.37	5.51
19	14.49	15.39	6.22
20	13.62	14.48	6.31
Average time taken by the robot	**14.20**	**15.10**	**6.33**

Table 6.10: Comparison of time taken by a single robot during simulation and experimental results shown in Figures (6.13(b) and 6.16) using IWO navigational algorithm.

No. of Runs	**Time taken by the robot during simulation (in 'sec')**	**Time taken by the robot during experiment (in 'sec')**	**% of error**
1	12.65	13.46	6.36
2	13.30	14.14	6.35
3	13.01	13.77	5.79
4	14.60	15.64	7.12
5	13.33	14.03	5.27
6	12.74	13.58	6.57
7	13.12	14.14	7.83
8	12.82	13.64	6.34
9	12.52	13.37	6.81
10	14.27	15.09	5.74
11	13.91	14.63	5.14
12	12.32	13.21	7.21
13	12.93	13.85	7.12
14	12.60	13.23	5.02
15	13.39	14.33	6.99
16	12.60	13.44	6.69
17	13.30	14.04	5.57
18	13.24	14.13	6.72
19	12.71	13.57	6.80
20	13.98	15.12	7.84
Average time taken by the robot	**13.17**	**14.02**	**6.46**

Tables 6.5 to 6.10 show the performance of the proposed navigational controller using IWO algorithm. The results obtained with IWO algorithm implies, no improvement over the CS navigational algorithm. The average errors are found to be about 6% for both path length and time taken by the robot to reach target.

6.5 Summary

This chapter provides an extensive study of IWO algorithm implemented for a mobile robot path planning problem. The path planning problem has been applied to an unknown or partially known environment with static obstacles. The followings outcomes from proposed algorithm have been extracted based on the simulation and experimental results.

- The developed navigational algorithm is effective in avoiding obstacles and the robot reaches the goal successfully.
- A new objective function has been generated based on the position of the obstacles and goal. The generated objective function has been successfully implemented to find nearest optimal paths for the robot in unknown environments.
- Based on the objective function values the robot is roving towards the goal while avoiding obstacles.
- A large number of simulation experiments are done using MATLAB to demonstrate the performance of the proposed navigation system.
- The developed path planner has been validated through simulation as well as real time experiment, and they are in good agreement. The average percentages of errors are found to be within 5% for path length and 6% for time taken by the robot to reach target and tabulated in Tables 6.5-6.10 respectively.
- The proposed IWO based navigation system has been compared to other computational techniques such as; Ant colony optimization (ACO), Particle swarm optimization (PSO) etc. By analyzing results, it has been seen that the IWO based navigational controller performs better than other navigation techniques. The percentage of deviation has been given in the Table 6.4.
- Finally, it is concluded that the proposed IWO path planner has been successfully generated collision free paths for mobile robots in unknown environments with shortest possible path length.

In the next section, CS-ANFIS hybrid technique has been implemented for single as well as multiple mobile robots navigation.

CHAPTER-7

7. IMPLEMENTATION OF CS-ANFIS HYBRID ALGORITHM FOR MULTIPLE MOBILE ROBOTS NAVIGATION

7.1 Introduction

Recently, hybrid algorithms are getting more popular and may be implemented in the majority of engineering optimization problems. The concept of hybrid algorithm method is based on hybridization of two or more algorithms. Several machine learning algorithms might be good to solve the problems to find the required solutions. There are many difficulties, where a direct machine learning algorithm may fail to get a convenient (optimal solution) solution. To overcome these difficulties, we need hybridization of machine learning algorithms with other optimization algorithms. There are some possible advantages of hybridization algorithm, which are as follows:

- Improve the performance of the machine learning algorithm.
- Precisely handle the problem with large data, uncertainty, and vagueness.
- Improve the convergence rate.

In this part, a new hybrid algorithm, (i.e hybridization of Cuckoo Search (CS) and Adaptive Neuro-Fuzzy Inference System (ANFIS) techniques) has been applied to solve the multiple mobile robots navigational problems. A brief description of the Cuckoo Search (CS) and Adaptive Neuro-Fuzzy Inference System (ANFIS) techniques are discussed in the Chapter-4 and Chapter-5 respectively. In this new hybrid navigational methodology, Cuckoo search (CS) algorithm is used for training the premise part and least square estimation (LSE) method is used for training the consequent parameters of the adaptive neuro-fuzzy inference system (ANFIS). The proposed hybrid optimal path planner is developed based on a reference motion, direction, distances between the robot and the obstacles, and distance between the robot and the target, to calculate the suitable steering angle of the mobile robot. In order to avoid collision against one another, a set of collision prevention rules are embedded into each robot controller, using Petri-Net model.

The effectiveness of the algorithm has been demonstrated through a series of simulation experiments. The experimental investigations are conducted in the laboratory, using a real mobile robot to validate the versatility and effectiveness of the proposed hybrid technique. A comparison of the simulation and experimental results showed that there is good agreement between them. The results obtained from the proposed hybrid technique are validated by comparison with the results from other intelligent techniques. Finally, it is concluded that the proposed hybrid methodology is efficient and robust in the sense, that it can be implemented on the robot for navigation in any complex terrain.

Many different neuro-fuzzy models for mobile robot path planning have been proposed in [141-146]. Among these techniques, the adaptive neuro-fuzzy inference system (ANFIS) is a neural network based on the fuzzy approach, which combines the parallel computation, and capability of adaptation of the neural network and human-like knowledge representation of fuzzy systems [149]. The training and updating of the premise and consequent parameters in ANFIS is one of the major problems. The training part for ANFIS is mostly based on the gradient descent algorithms. The main disadvantage of this algorithm is the calculation of gradients in each step, and it may also cause the local minimum and affects the accuracy of the system.

In the gradient descent method, the selection of the best learning parameters is very difficult, and the convergence rate of parameters is also slow. To overcome these difficulties, we implemented a nature inspired meta-heuristic algorithm to train the antecedent parameters, and the least square estimation (LSE) method to optimize the consequent parameters of the ANFIS navigational controller, for reducing training and testing errors. Cuckoo search (CS) is an optimization metaheuristic search algorithm, developed by Yang and Deb in 2009 [192-193], which was inspired by the obligate parasitic behavior of some cuckoos, who lay their eggs in the nest of other host birds. The Cuckoo search algorithm does not require any learning rate and provides a faster convergence rate of parameters in ANFIS. An advantage of the CS algorithm is that the number of parameters to be tuned is less than those of the GA and PSO [221] and thus it is potentially more generic for application in a wider class of optimization problems, and also it has proved that the cuckoo search algorithm satisfies the global convergence necessities, and thus, has guaranteed global convergence properties. Due to its broad range of implementations in various engineering problems [198-200,206-208], we are

motivated to apply this nature inspired cuckoo search algorithm for optimizing the ANFIS parameters and solving the navigational problem for multiple mobile robots.

7.2 Architecture of CS-ANFIS Hybrid Controller for Multiple Mobile Robots Navigation

The main objective of the research work is to predict the steering angle (SA) for mobile robots to navigate safely in an environment using CS based ANFIS controller. The initial position of the robot is shown in Figure 7.1.

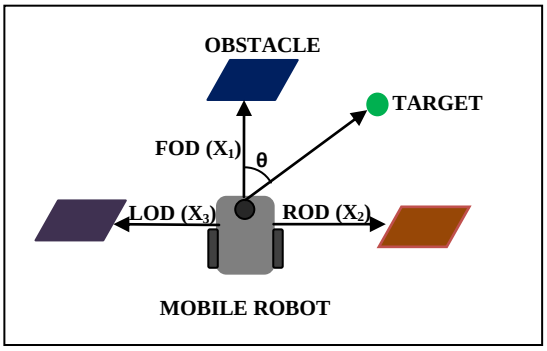


Figure 7.1 Initial position of the Robot in real environment.

Here we assume, the cuckoo search based adaptive neuro-fuzzy navigational controller (shown in Figure 7.2) under consideration. It has four sensor based input parameters, such as Front obstacle distance(FOD) (x_1), Right obstacle distance(ROD) (x_2), Left obstacle distance(LOD)(x_3), Heading angle(HA)(x_4), and each input variable has three bell membership functions(MF) A_1(Near), A_2(Medium) and A_3(Far), B_1(Near), B_2(Medium) and B_3(Far), C_1(Near), C_2(Medium), and C_3(Far), D_1(Negative), D_2(Zero) and D_3(positive) respectively.

Then the Takagi-Sugeno-type fuzzy inference system IF-THEN rules are constructed as follows;

IF(x_1 is A_i and x_2 is B_i and x_3 is C_i and x_4 is D_i)

THEN (steering angle (f_n) = $p_n x_1 + q_n x_2 + r_n x_3 + s_n x_4 + u_n$)

A_i, B_i, C_i, and D_i, are the fuzzy membership sets of the input variables x_1, x_2, x_3, and x_4 respectively.

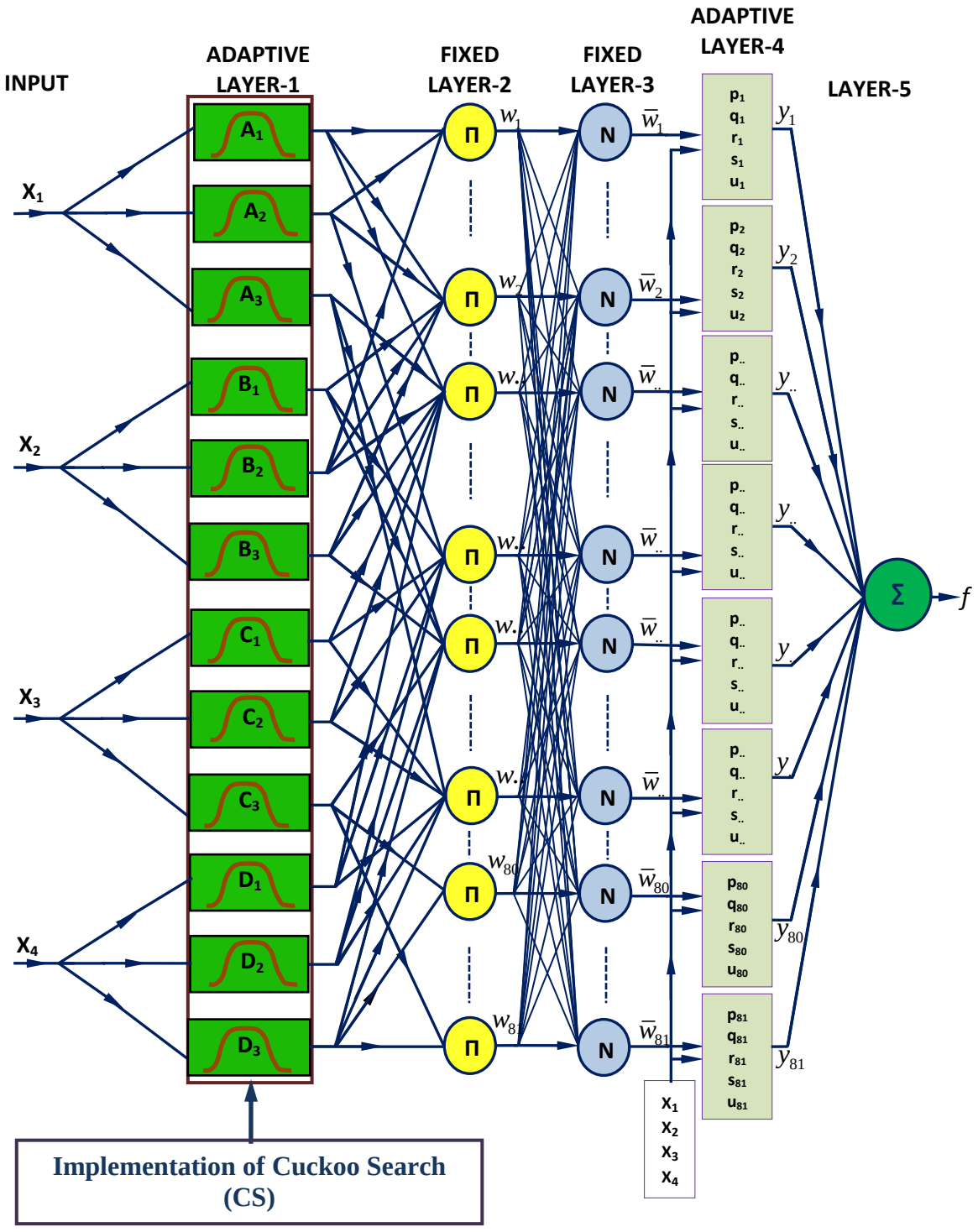


Figure 7.2 CS-ANFIS hybrid controller for Mobile robot navigation.

Where i=1, 2, 3 and n=1, 2…81, p_n, q_n, r_n, s_n, and u_n are the linear parameters of output f_n, also known as consequent parameters, and changing these parameters we can adjust the output of the CS-ANFIS controller. The fuzzy membership function can be chosen in any form such as Triangular, Gaussian, Trapezoidal, Bell-Shaped etc. Usually, we choose the Bell-Shaped one with the maximum value equal to 1 and minimum value equal to 0. Here, the membership functions for all input parameters are considered as the Bell-Shaped function (shown in Figure 7.3).

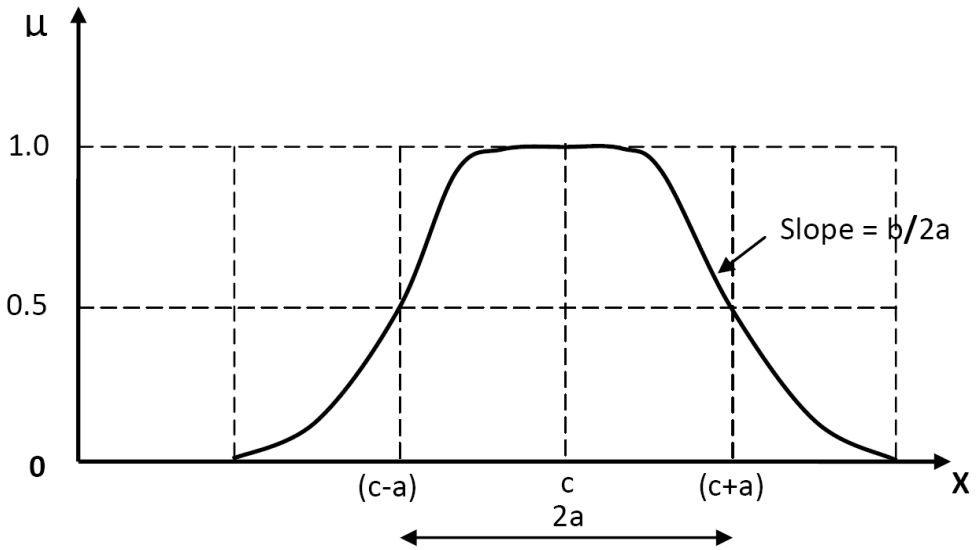


Figure 7.3 Parameters in Bell Membership function.

Where a, b, and c are the bell-fuzzy numbers. They are also known as the antecedent or premise parameters.

In our proposed CS-ANFIS model total 12 sets of premise parameters or non-linear parameters $\{a_i, b_i, c_i\}$ for inputs are presented in the proposed ANFIS navigational controller. The total number of premise parameters to be tuned is 12x3=36numbers. The number of fuzzy rules formulated is 3x3x3x3=81 numbers and there are 81x5=405 numbers of consequent parameters or linear parameters $\{p_n, q_n, r_n, s_n, u_n\}$ to be trained in ANFIS, to get the system output. In our proposed hybrid learning algorithm, the CS and least square estimation (LSE) methods are used in ANFIS for training and adjusting the premise parameters and consequent parameters. The ANFIS controller is used to determine the objective function value of the steering angle for a mobile robot generated by the CS algorithm. Finally, the error between the system output and actual training data can reach a minimum value through the iteration of the CS algorithm. Here, the objective

function of the CS optimizer is defined as the root mean square error (RMSE) of the proposed ANFIS model.

$$\text{RMSE}= \sqrt{\frac{1}{p}\sum_{i=1}^{p}\left(\theta_i - \theta_i^{'}\right)^2} \tag{7.1}$$

Where θ_i is the actual value of the steering angle, $\theta_i^{'}$ is the predicted value of the mobile robot obtained from the ANFIS based CS model, and p is the total number of observations. The iterations will stop when the RMSE value remains unchanged, or in other words, the CS algorithm cannot find any better parameters to reduce the error value. It has been observed that the CS-ANFIS hybrid algorithm produces a lower error rate than the conventional ANFIS. The coding of the premise parameters for ANFIS in each nest is shown in Figure 7.4.

a_{i1}	b_{i1}	c_{i1}	a_{i2}	b_{i2}	c_{i2}	a_{i3}	b_{i3}	c_{i3}	a_{i4}	b_{i4}	c_{i4}	.....	a_{nj}	b_{nj}	c_{nj}

a_{ij}, b_{ij}, and c_{ij} are the bell membership function parameters.

i=1,2,3,……..n. (Total number of membership functions for input parameters)

j=1….4. (Number of input parameters)

Figure 7.4 Nests contain all information of membership function parameters for all input.

Therefore, the number of parameters present in each nest will specify a real value of all possible premise parameters of FIS (Fuzzy Inference System). An excel file has been prepared that contains all the 36 numbers of antecedent parameters of ANFIS, and it can be retrieved by the MATLAB during coding. Initially, 25 number of nests are randomly generated, each represented by a vector having dimension [36x1]. The detailed flow chart of the proposed hybrid algorithm has been presented in Figure 7.5. The membership function for the fuzzy part was chosen to be the bell function, with the maximum value equal to one and the minimum value equal to zero. The proposed CS-ANFIS hybrid algorithm can be used to reduce the probability of being trapped in local minima situation and also enhance the accuracy and global search ability of ANFIS training. It has been noted that the new hybrid algorithm can save time in adjusting all parameters of ANFIS navigational controller.

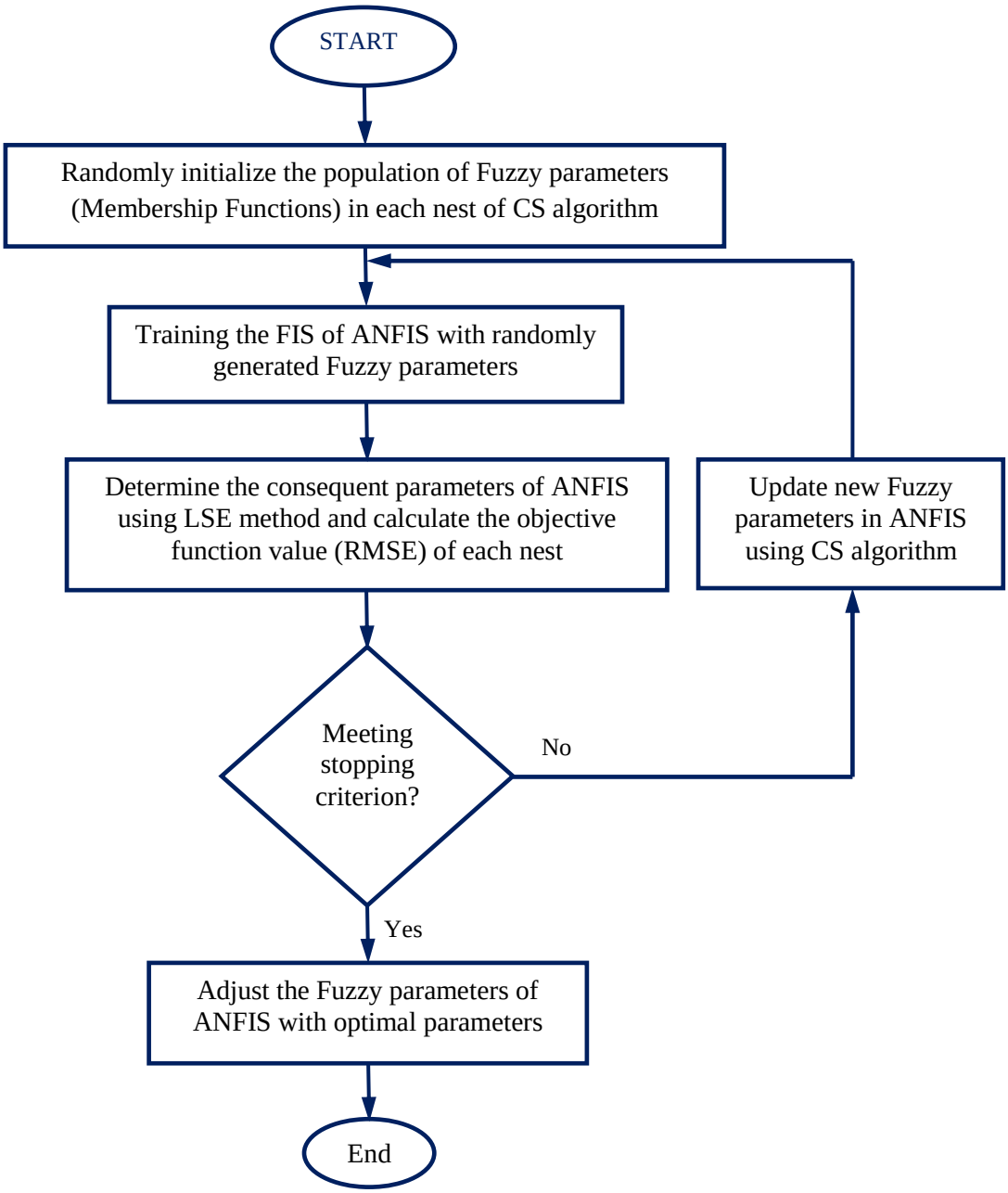


Figure 7.5 Flow chart diagram for the proposed hybrid algorithm.

Steps of CS-ANFIS hybrid algorithm for mobile robot path planning:

Step-1: Randomly generate the initial population of host nests in the search space, each representing a set of 36 numbers of antecedent parameters of ANFIS in sequence.

Step-2: Randomly a nest (sequence of antecedent parameters) is generated; its objective function (RMSE) is determined as per the equation no. 7.1 and compared with that of the randomly chosen nest, if the nest is better, it replaces the chosen nest, otherwise it leaves intact.

Step-3: The worst nests (as per the value of p_a) are removed and replaced with new nests by the Levy flight technique as well as by random generation.

Step-4: Solutions (sequence of antecedent parameters) are ranked and then the nest is updated again via Levy flight method and steps 2 to 4 are repeated until the CS algorithm finds any better parameters to reduce the error value.

Step-5: Train the optimal fuzzy parameters to ANFIS, in order to calculate the suitable steering angle.

The parameters of the navigational controller obtained from the CS algorithm are used to train the ANFIS in order to determine the suitable steering angle for any reactive condition. The various reactive behaviours and training patterns of ANFIS for mobile robot path planning are discussed in the Chapter-4.

7.3 Inter-collision Avoidance among the Robots using the Petri-Net Controller

The Petri-Net theory was developed by Petri [224-225] and is widely used to represent the dynamic systems. As per the robot navigation system, condition (place) represents the state of motion and is denoted by a circle. A state of transition (event) represents the change in robot motion, and it is shown by a bar symbol. The location of the token represents the holding of robot motion (current position) and is marked with a red symbol. The schematic diagram of this proposed model to avoid inter-collision and obstacle avoidance during navigation is shown below in Figure 7.6. The proposed Petri-Net controller consists of 6 states (or places) which are described briefly.

Initially, it is assumed that the robots are in a maze environment, without any prior information about one another and the position of the obstacles and targets. This implies

that each robot is in state “Place-I” (Wait for the start command to search for the goal) or the token is at location “Place-I”.

Once the robots have received a command to start searching for goals, they will try to trace them while avoiding obstacles and one another. Thus, the robots are in “Place-II (“Navigating, avoiding obstacles, and searching for goals”).

While roving in a cluttered environment, if the pathway to the goal is blocked by another robot, then a conflict situation is found (“Place-III”: “Detecting Conflicts”). When two robots are in a conflict situation, they will negotiate with each other to decide which one has the priority.

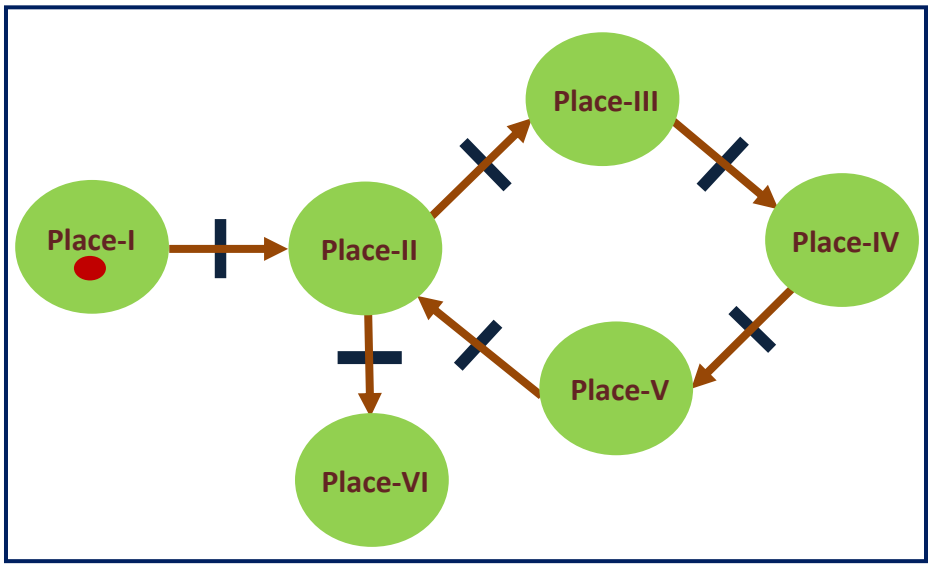


Figure 7.6 Developed Petri-Net controller for Multiple Robots Navigation.

The lower-priority robot will be considered as a static obstacle and the higher priority robot as a proper mobile system (“Place-IV”: “Negotiating”). As soon as a conflict problem is resolved, the robots will look for other conflicts and if there are none they will execute their movements. (“Place-V”: “Checking for conflict and executing movements”).

If a robot meets two other robots already in a conflict situation, then its priority will be lower and it will be considered as a static obstacle, (“Place-VI”: “Waiting”) until the conflict problem is resolved. When this is finished, the robot will re-enter the state “Place-II”. Finally, combining these conditions, the navigation of multiple mobile robots in an unstructured environment can be made possible.

The simulation and experimental results obtained based on CS-ANFIS controller while navigation of mobile robots are demonstrated and discussed below.

7.4 Simulation Results and Discussion

In this section, various simulation exercises have been performed with single and multiple mobile robots to show the capability of the proposed hybrid controller. The simulation space is a rectangular environment with different types of obstacles at various locations, and X and Y axes are considered as 30x30 units, each unit is equal to 2mm. The best parametric values for the CS algorithm have been selected after performing a series of simulation exercises on partially or totally unknown environments and are given in Table-7.1. It has been seen that the parametric values affect the performance of the proposed navigation system.

Table-7.1 (Parameters used in CS algorithm.)

Parameters	Values
No. of Nests(n)	10-40
No. of Iterations	100-200
p_a	0.05-0.3

From our simulation exercises, it has been clearly noticed that n=25-30 and p_a=0.23-0.26, provide best results compared with other parametric values of CS algorithm.

7.4.1 Simulation results for a Single Robot

In the current motion planning system, we have framed three main reactive behaviors: one for target seeking, one for wall following, and the last for avoiding obstacles. The descriptions of various behaviours are discussed in Chapter-4.

Let us consider that every obstacle is far away from the robot and only then, the "reach target behavior" is activated. On the other side when a robot is close to an obstacle, it must change its steering angle, to avoid the obstacle present on the path. Various reactive behaviors are activated, depending upon the situation between the robot and obstacles. The simulation results are obtained by placing the obstacles in random positions and a random heading angle to verify the various reactive behaviors developed by the current hybrid navigational controller. A comparative simulation study has been made between the ANFIS and CS-ANFIS hybrid techniques as shown in Figures 7.7-7.12 respectively. The importance of these two techniques is mainly measured, based on the path length

covered by the robot and is given in Tables 7.2 respectively. It can be noticed that the CS-ANFIS hybrid algorithm yields better results compared to the ANFIS technique and also provides smoothness to the robot trajectory.

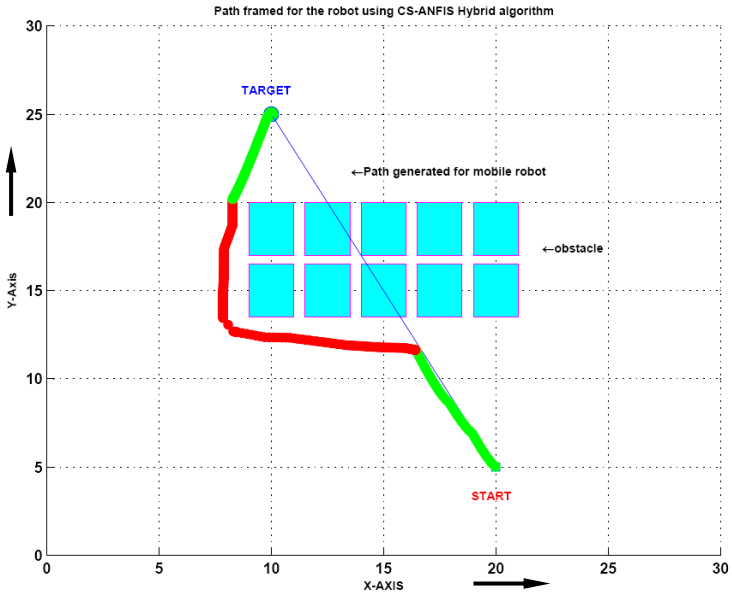


Figure 7.7 Wall following behavior by a single robot using CS-ANFIS Hybrid technique.

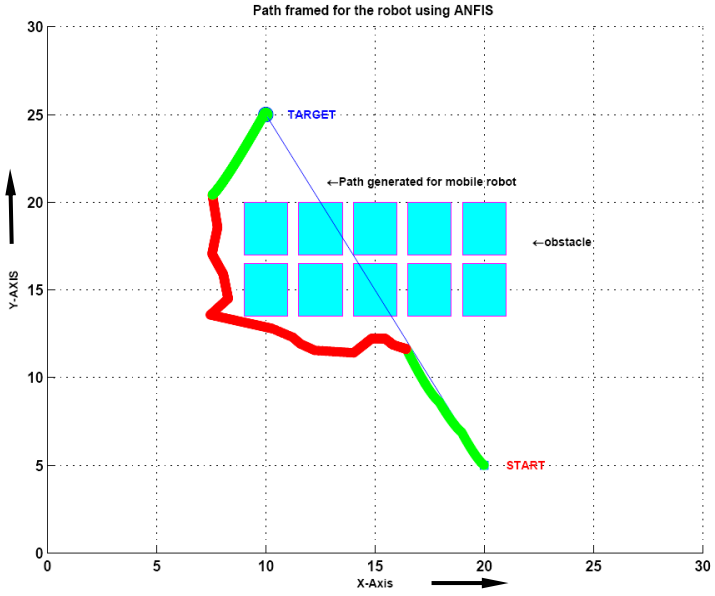


Figure 7.8 Wall following behavior by a single robot using ANFIS technique.

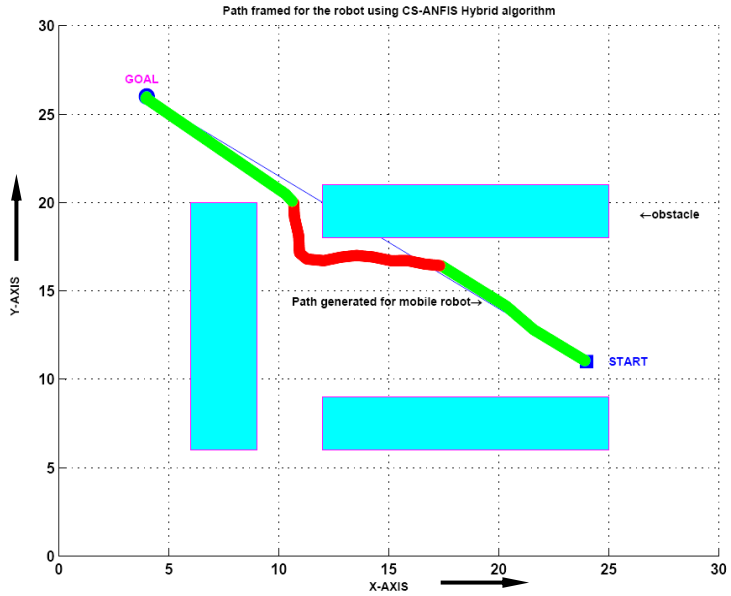


Figure 7.9 Escaping from a narrow end by a single robot using CS-ANFIS Hybrid technique.

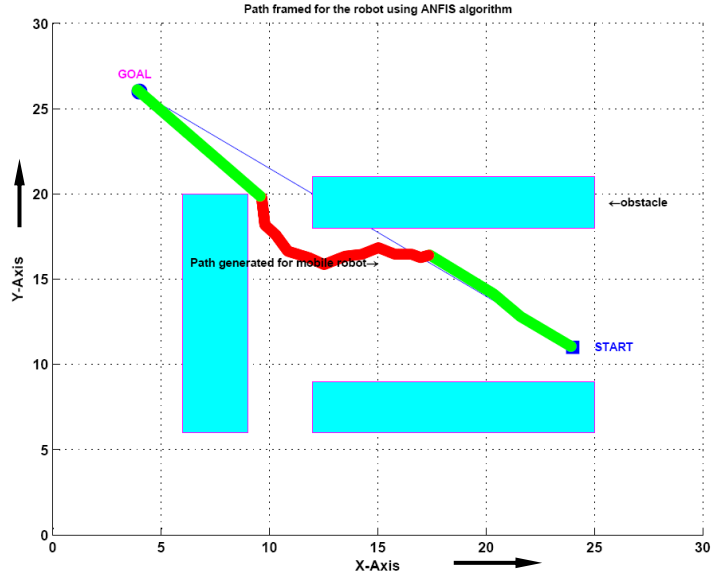


Figure 7.10 Escaping from a narrow end by a single robot using ANFIS technique.

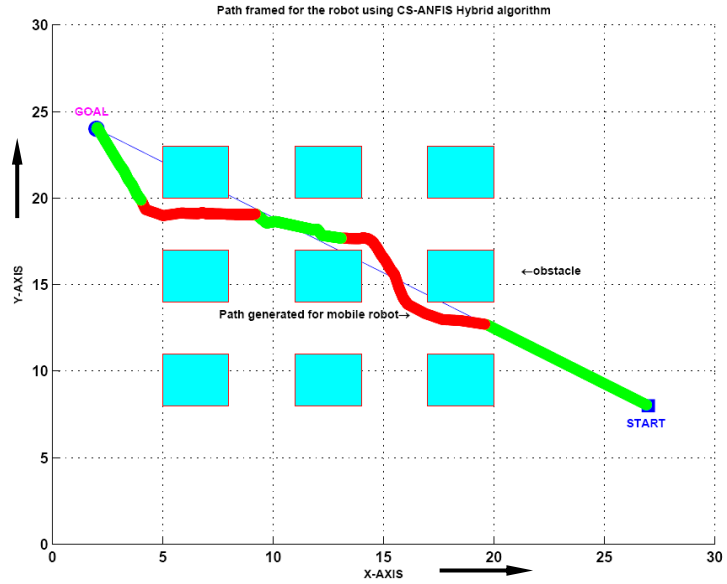


Figure 7.11 Navigating inside a maze environment by a single robot using CS-ANFIS Hybrid technique.

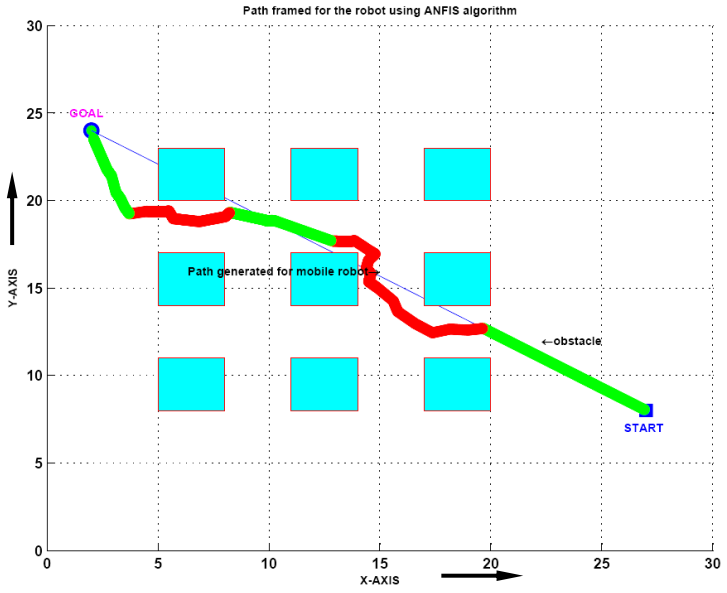


Figure 7.12 Navigating inside a maze environment by a single robot using ANFIS technique.

Table-7.2 (Comparison of CS-ANFIS and ANFIS results in terms of path length)

SL. No.	Path covers by the Robot using CS-ANFIS (in 'cm')	Path covers by the Robot using ANFIS (in 'cm')	Percentage of deviation
1	6.31 (Fig. 7.7)	6.68 (Fig. 7.8)	5.53
2	6.20 (Fig. 7.9)	6.55 (Fig. 7.10)	5.34
3	7.60 (Fig. 7.11)	7.80 (Fig. 7.12)	3.18

7.4.2 Simulation results for Multiple Robots

- *Obstacle avoidance, Wall following and Goal seeking by Multiple Robots:*

This simulation experiment shown in Figure 7.13 involves four robots with four goals at different positions of the environment. The walls present between the robots and targets acts as obstacles. As the robots navigate towards their goals, they find the walls in their way along which they continue to travel by implementing the wall following rules. Finally, the robots are able to reach their goals successfully.

- *Obstacle avoidance and collision-free motion of Multiple Robots in a maze environment:*

The obstacle avoidance and collision-free motion of multiple mobile robots in two different maze environments have been shown in Figures 7.14 and 7.15 respectively. In this simulation study, only four robots are employed for easy visualization. It can be noticed that the robots have effectively reached their goals without any collision with one another and the obstacles.

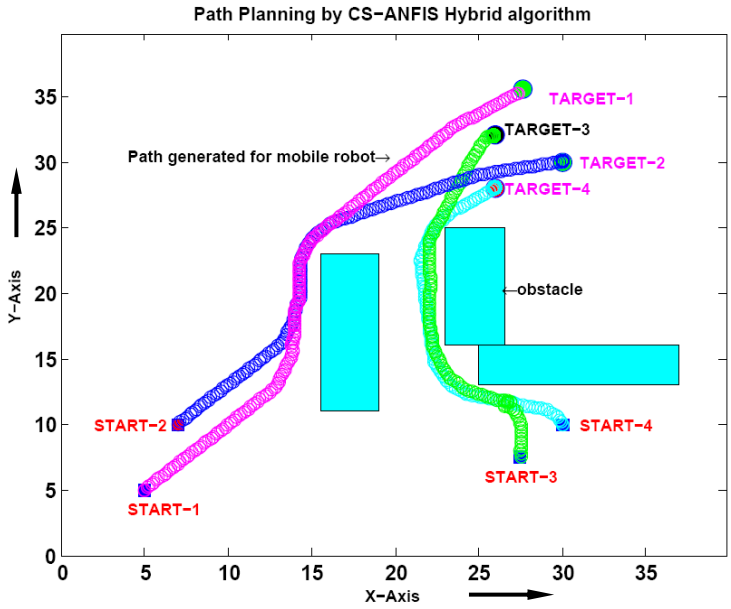


Figure 7.13 Obstacle avoidance and wall following behavior by multiple mobile robots using CS-ANFIS Hybrid technique.

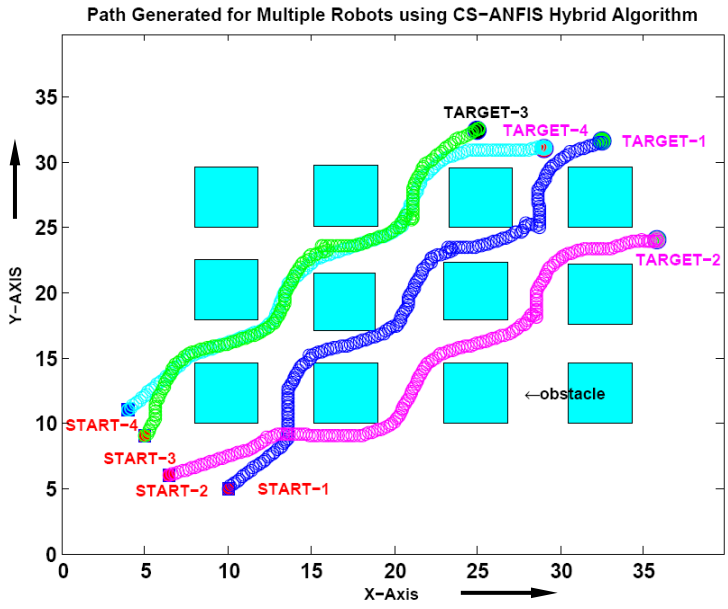


Figure 7.14 Multiple mobile robots navigating in a maze environment using CS-ANFIS Hybrid technique.

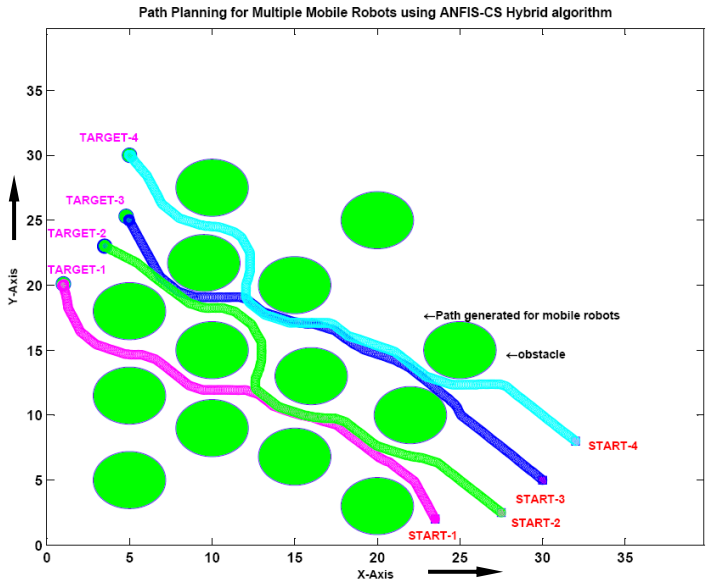


Figure 7.15 Path framed by multiple mobile robots in a maze environment using CS-ANFIS hybrid technique.

7.5 Experimental Validation with the Simulation Results

The proposed navigation method has been validated through a series of real-time experiments using Khepera-II mobile robot to show its effectiveness. The specification of the Khepera-II mobile robot has been given in the Appendix-A. The experimental investigation has been conducted in an environment containing a variety of obstacles of different shapes and sizes and located in different locations. There is total eight infra-red proximity and ambient light sensors with up to 100mm range are mounted around the periphery of the mobile robot in order to extract the information about the distances between the robots and the obstacles. The results shown in Figures 7.16-7.19, are already performed in simulation mode, and have been validated experimentally to demonstrate the efficiency of the developed path planner.

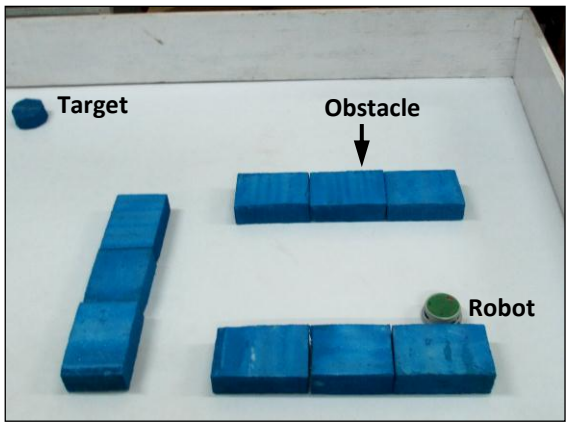


Figure 7.16 (a)

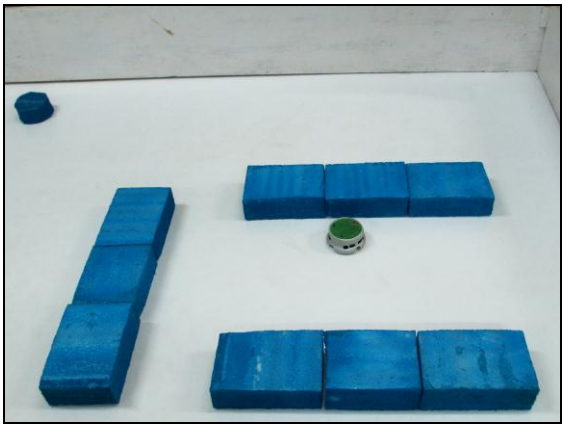


Figure 7.16 (b)

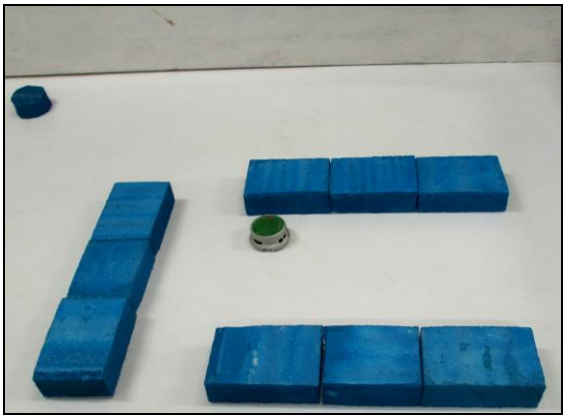


Figure 7.16 (c)

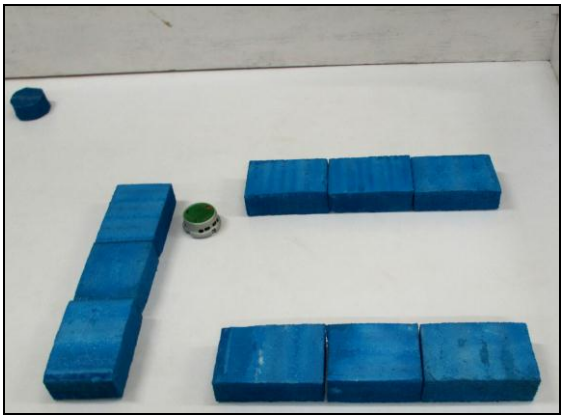


Figure 7.16 (d)

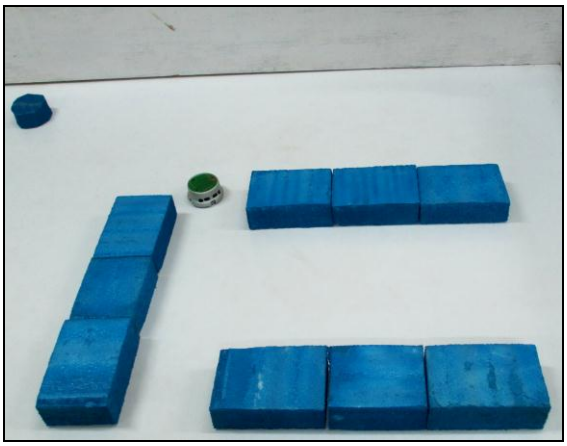


Figure 7.16 (e)

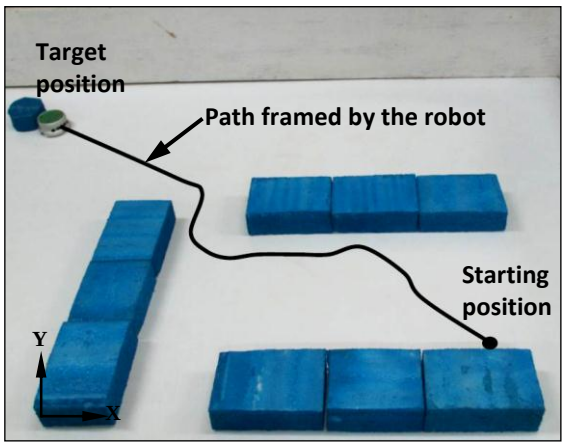


Figure 7.16 (f)

Figure 7.16 (a-f) Experimental results for navigation of mobile robot in the environment shown in Figure 5.26.

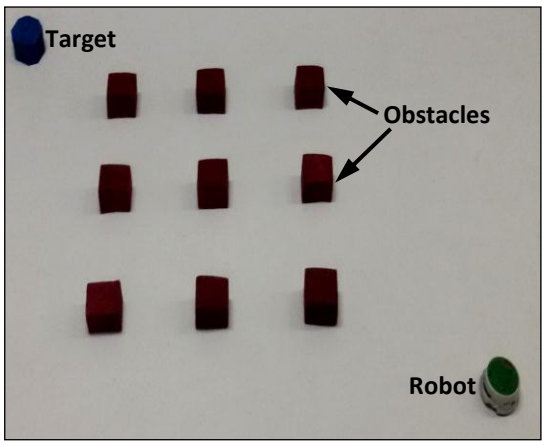


Figure 7.17 (a)

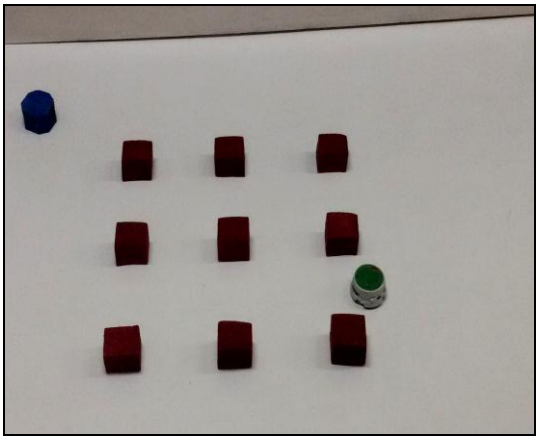


Figure 7.17 (b)

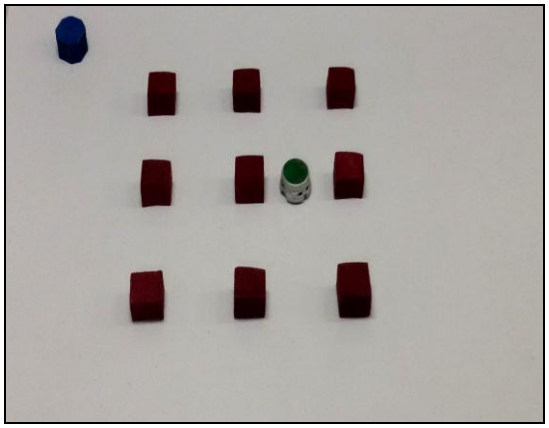


Figure 7.17 (c)

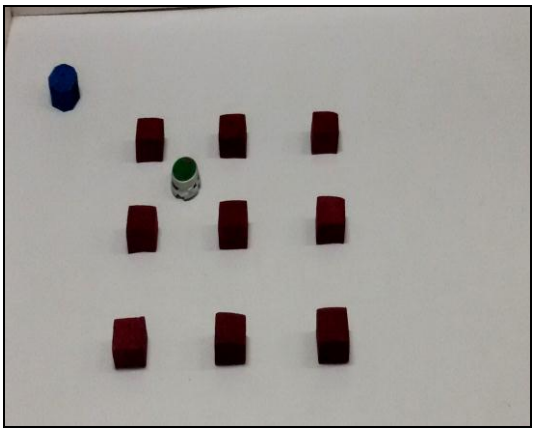


Figure 7.17 (d)

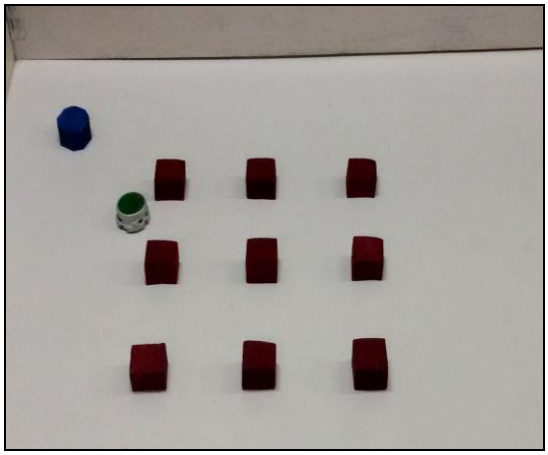


Figure 7.17 (e)

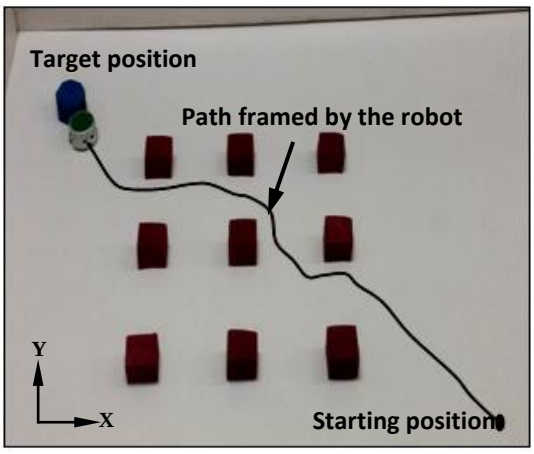


Figure 7.17 (f)

Figure 7.17 (a-f) Experimental results for navigation of mobile robot in the environment shown in Figure 5.28.

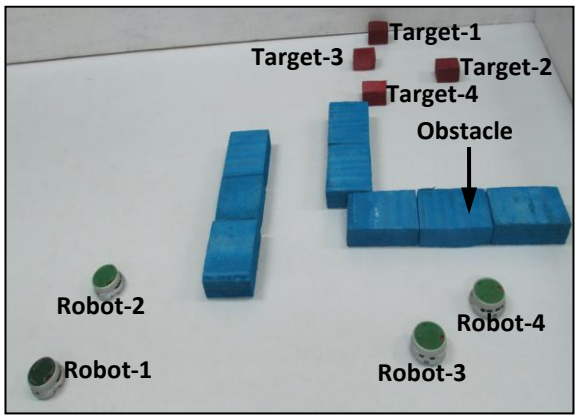


Figure 7.18 (a)

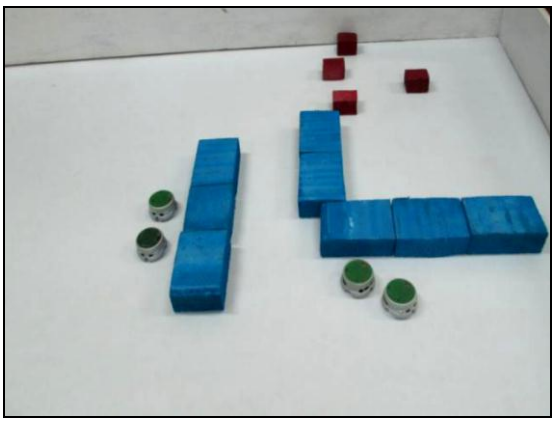


Figure 7.18 (b)

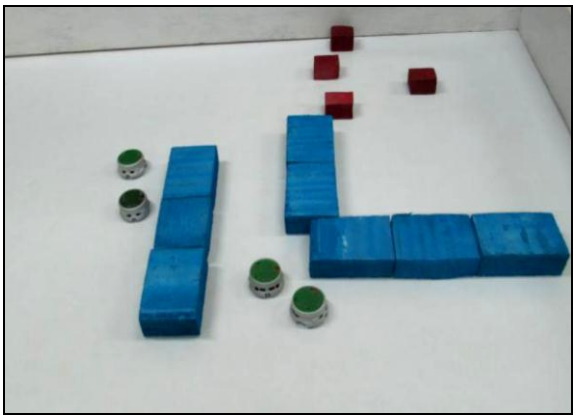


Figure 7.18 (c)

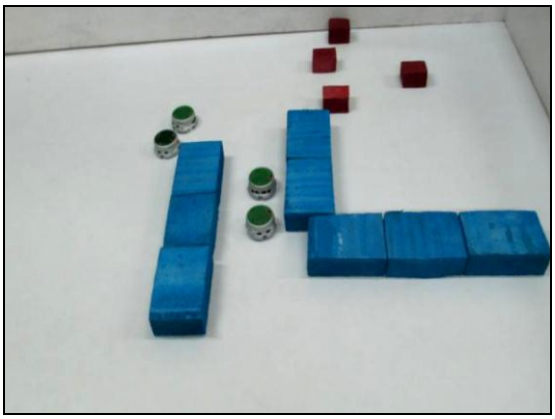


Figure 7.18 (d)

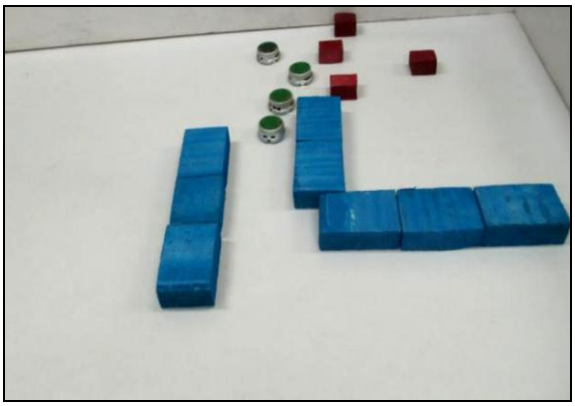


Figure 7.18 (e)

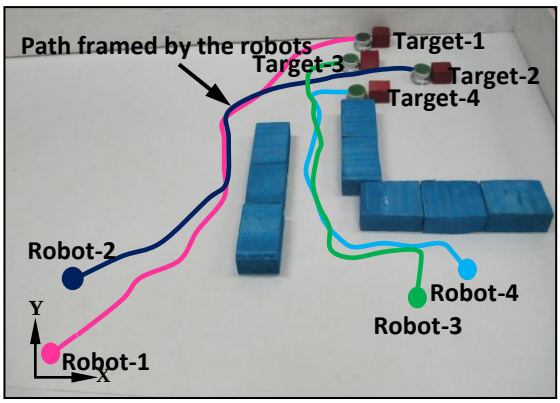


Figure 7.18 (f)

Figure 7.18 (a-f) Experimental results for navigation of multiple mobile robots in the environment shown in Figure 5.30.

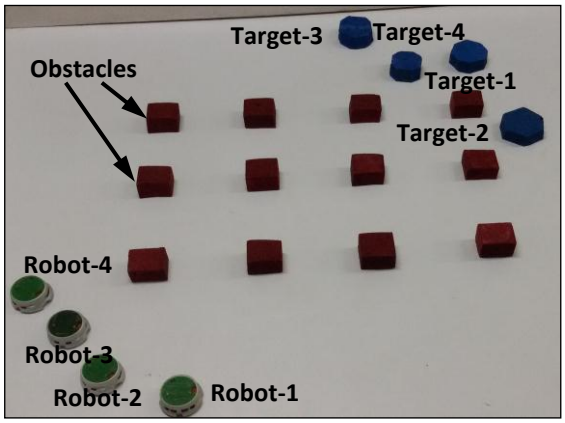


Figure 7.19 (a)

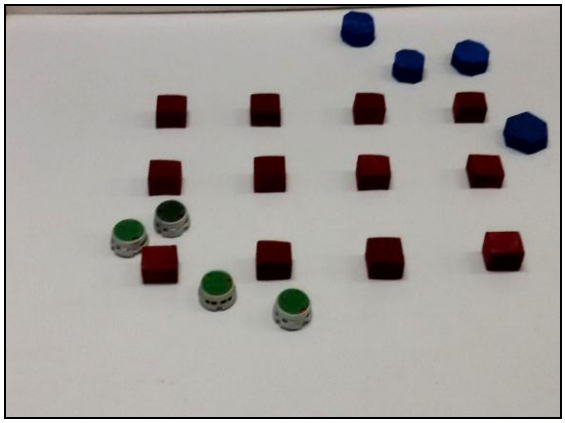


Figure 7.19 (b)

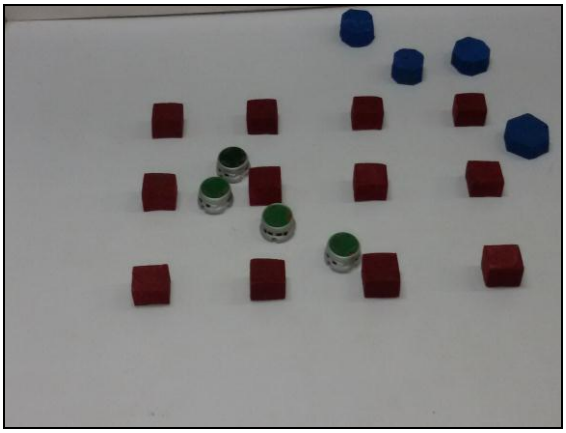


Figure 7.19 (c)

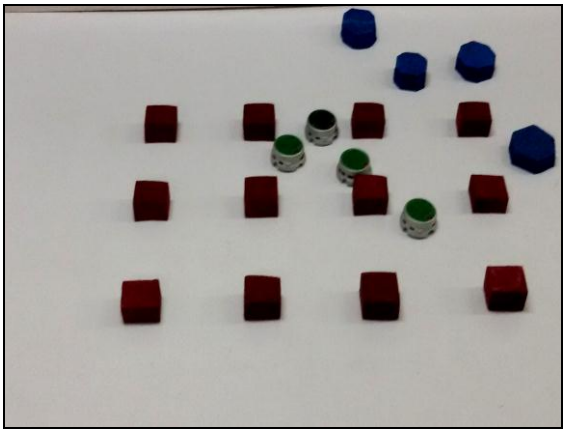


Figure 7.19 (d)

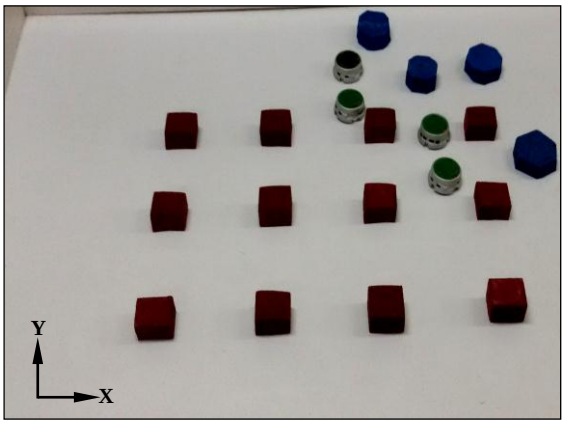


Figure 7.19 (e)

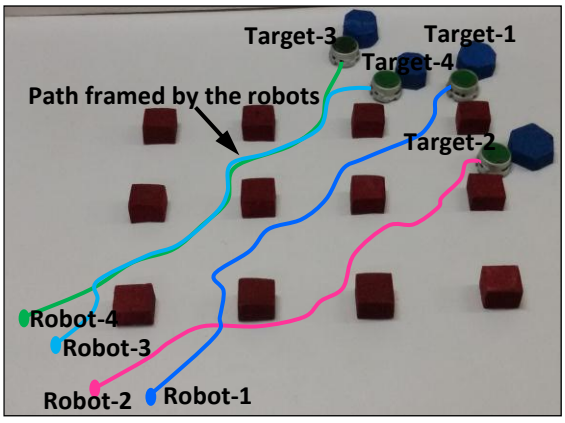


Figure 7.19 (f)

Figure 7.19 (a-f) Experimental results for navigation of multiple mobile robot in the environment shown in Figure 5.31.

Table-7.3 The Path travelled by the single robot during simulation and experimental analysis by the proposed hybrid navigation system shown in Figures 7.9 and 7.16 respectively.

No. of Runs	**Path length covered during simulation (in 'cm')**	**Path length covered during the experiment (in 'cm')**	**% of error**
1	163.17	171.84	5.31
2	163.04	170.18	4.38
3	163.96	171.02	4.31
4	164.03	171.67	4.66
5	162.57	170.33	4.77
6	163.30	172.13	5.41
7	163.19	171.16	4.88
8	163.67	169.79	3.74
9	163.82	169.82	3.66
10	163.93	170.19	3.82
11	162.78	172.01	5.67
12	163.75	170.27	3.98
13	163.69	172.06	5.11
14	162.51	170.14	4.69
15	162.41	171.34	5.50
16	163.32	170.50	4.40
17	164.42	170.89	3.93
18	162.94	170.16	4.44
19	163.52	171.39	4.81
20	162.66	170.95	5.10
Average path length covered	**163.33**	**170.89**	**4.63**

Table-7.4 The Path travelled by the single robot during simulation and experimental analysis by the proposed hybrid navigation system shown in Figures 7.11 and 7.17 respectively.

No. of Runs	**Path length covered during simulation (in 'cm')**	**Path length covered during the experiment (in 'cm')**	**% of error**
1	186.15	195.89	5.23
2	186.52	194.35	4.20
3	183.36	193.50	5.53
4	186.55	195.47	4.78
5	185.41	196.02	5.72
6	183.25	193.50	5.60
7	183.98	194.39	5.66
8	185.06	194.39	5.04
9	186.73	195.89	4.90
10	186.76	194.65	4.23
11	185.22	195.17	5.37
12	186.15	193.93	4.18
13	186.73	194.73	4.29
14	184.82	192.91	4.38
15	186.09	195.98	5.31
16	183.42	193.78	5.65
17	184.56	193.95	5.09
18	186.56	194.56	4.29
19	186.06	196.57	5.65
20	186.74	195.36	4.62
Average path length covered	**185.51**	**194.75**	**4.99**

Table-7.5 Time taken by the single robot during experimental and simulation analysis by the proposed navigation system shown in Figures 7.9 and 7.16.

No. of Runs	**Time taken by the robot during simulation (in 'sec')**	**Time taken by the robot during experiment (in 'sec')**	**% of error**
1	13.49	14.14	4.83
2	13.14	13.82	5.21
3	13.04	13.66	4.74
4	13.43	14.09	4.89
5	12.01	12.65	5.31
6	13.63	14.38	5.49
7	13.41	14.10	5.12
8	13.71	14.34	4.58
9	13.28	13.98	5.23
10	13.43	14.05	4.61
11	13.24	13.88	4.80
12	13.43	14.08	4.77
13	12.98	13.58	4.67
14	13.83	14.46	4.54
15	13.33	14.05	5.40
16	13.62	14.25	4.65
17	12.92	13.59	5.19
18	14.03	14.78	5.34
19	13.42	14.11	5.14
20	12.96	13.59	4.87
Average time taken by the robot	**13.32**	**13.98**	**4.97**

Table-7.6 Time taken by the single robot during experimental and simulation analysis by the proposed navigation system shown in Figures 7.11 and 7.17

No. of Runs	**Time taken by the robot during simulation (in 'sec')**	**Time taken by the robot during experiment (in 'sec')**	**% of error**
1	16.91	17.87	5.65
2	17.25	18.11	5.01
3	16.52	17.26	4.51
4	17.12	18.06	5.51
5	16.63	17.35	4.31
6	16.21	16.94	4.52
7	16.48	17.28	4.85
8	15.94	16.65	4.42
9	15.63	16.49	5.52
10	15.67	16.56	5.69
11	16.46	17.35	5.42
12	16.98	17.73	4.44
13	16.35	17.06	4.36
14	16.98	17.90	5.42
15	16.54	17.42	5.32
16	16.42	17.21	4.79
17	15.76	16.52	4.85
18	16.78	17.62	5.01
19	16.51	17.27	4.58
20	16.59	17.31	4.31
Average time taken by the robot	**16.49**	**17.30**	**4.92**

Table-7.7 The Path travelled by the robots during simulation and experimental analysis by the proposed hybrid navigation system shown in Figures 7.13 and 7.18 respectively.

No. of Runs	Robot No.	Path covered by the robots during simulation analysis (in 'cm')	Path covered by the robots during experimental analysis (in 'cm')	% of error
1	Robot-1 (Start-1)	176.47	185.50	5.12
	Robot-2 (Start-2)	168.88	176.55	4.54
	Robot-3 (Start-3)	139.89	146.24	4.54
	Robot-4 (Start-4)	122.27	127.46	4.24
2	Robot-1 (Start-1)	176.31	186.01	5.50
	Robot-2 (Start-2)	167.04	176.23	5.50
	Robot-3 (Start-3)	140.71	148.38	5.45
	Robot-4 (Start-4)	124.68	129.24	3.66
3	Robot-1 (Start-1)	177.42	186.22	4.96
	Robot-2 (Start-2)	167.97	175.72	4.61
	Robot-3 (Start-3)	143.18	149.48	4.40
	Robot-4 (Start-4)	123.97	128.95	4.01
4	Robot-1 (Start-1)	177.51	185.17	4.32
	Robot-2 (Start-2)	168.69	176.27	4.50
	Robot-3 (Start-3)	140.63	148.05	5.27
	Robot-4 (Start-4)	125.23	129.84	3.69
5	Robot-1 (Start-1)	175.74	185.05	5.29
	Robot-2 (Start-2)	168.23	175.23	4.16
	Robot-3 (Start-3)	143.77	149.97	4.31
	Robot-4 (Start-4)	124.95	130.10	4.12

6	Robot-1 (Start-1)	176.92	186.20	5.25
	Robot-2 (Start-2)	169.65	176.65	4.13
	Robot-3 (Start-3)	142.86	149.07	4.35
	Robot-4 (Start-4)	125.78	131.52	4.57
7	Robot-1 (Start-1)	176.49	185.03	4.84
	Robot-2 (Start-2)	168.12	177.10	5.34
	Robot-3 (Start-3)	141.75	148.34	4.65
	Robot-4 (Start-4)	125.72	130.57	3.86
8	Robot-1 (Start-1)	177.07	185.72	4.88
	Robot-2 (Start-2)	166.61	176.01	5.64
	Robot-3 (Start-3)	142.16	148.61	4.54
	Robot-4 (Start-4)	123.27	128.86	4.53
9	Robot-1 (Start-1)	177.26	186.40	5.16
	Robot-2 (Start-2)	166.65	175.86	5.53
	Robot-3 (Start-3)	140.59	146.61	4.28
	Robot-4 (Start-4)	124.87	130.01	4.11
10	Robot-1 (Start-1)	177.39	185.35	4.49
	Robot-2 (Start-2)	167.05	175.10	4.82
	Robot-3 (Start-3)	141.61	148.77	5.05
	Robot-4 (Start-4)	122.67	128.36	4.64
Average path length covered	Robot-1 (Start-1)	**176.86**	**185.66**	**4.98**
	Robot-2 (Start-2)	**167.89**	**176.07**	**4.88**
	Robot-3 (Start-3)	**141.72**	**148.35**	**4.68**
	Robot-4 (Start-4)	**124.34**	**129.49**	**4.14**

Table-7.8 The Path travelled by the robots during simulation and experimental analysis by the proposed hybrid navigation system shown in Figures 7.14 and 7.19 respectively.

No. of Runs	Robot No.	Path covered by the robots during simulation analysis (in 'cm')	Path covered by the robots during experimental analysis (in 'cm')	% of error
1	Robot-1 (Start-1)	172.22	179.88	4.45
	Robot-2 (Start-2)	169.44	176.41	4.11
	Robot-3 (Start-3)	161.75	170.18	5.21
	Robot-4 (Start-4)	154.55	160.92	4.12
2	Robot-1 (Start-1)	174.46	181.35	3.95
	Robot-2 (Start-2)	167.22	173.01	3.46
	Robot-3 (Start-3)	165.60	174.32	5.27
	Robot-4 (Start-4)	153.64	160.70	4.59
3	Robot-1 (Start-1)	173.44	180.59	4.12
	Robot-2 (Start-2)	165.15	172.22	4.28
	Robot-3 (Start-3)	167.93	175.91	4.75
	Robot-4 (Start-4)	156.58	163.71	4.56
4	Robot-1 (Start-1)	172.13	179.73	4.42
	Robot-2 (Start-2)	166.24	173.19	4.18
	Robot-3 (Start-3)	166.38	174.15	4.67
	Robot-4 (Start-4)	154.49	160.94	4.18
5	Robot-1 (Start-1)	172.90	181.25	4.83
	Robot-2 (Start-2)	165.54	173.67	4.91
	Robot-3 (Start-3)	162.90	171.59	5.34
	Robot-4 (Start-4)	155.46	162.21	4.34

6	Robot-1 (Start-1)	173.99	180.84	3.94
	Robot-2 (Start-2)	166.33	173.91	4.56
	Robot-3 (Start-3)	164.20	172.83	5.26
	Robot-4 (Start-4)	153.23	158.77	3.62
7	Robot-1 (Start-1)	172.29	180.98	5.05
	Robot-2 (Start-2)	167.33	175.49	4.88
	Robot-3 (Start-3)	164.87	173.08	4.98
	Robot-4 (Start-4)	153.09	159.63	4.27
8	Robot-1 (Start-1)	174.53	182.89	4.79
	Robot-2 (Start-2)	167.87	173.97	3.63
	Robot-3 (Start-3)	168.66	175.31	3.94
	Robot-4 (Start-4)	155.62	162.35	4.33
9	Robot-1 (Start-1)	172.50	180.88	4.86
	Robot-2 (Start-2)	167.43	175.09	4.57
	Robot-3 (Start-3)	162.65	169.43	4.17
	Robot-4 (Start-4)	150.84	158.73	5.23
10	Robot-1 (Start-1)	175.26	181.90	3.79
	Robot-2 (Start-2)	170.04	176.66	3.89
	Robot-3 (Start-3)	167.74	174.87	4.25
	Robot-4 (Start-4)	151.16	158.13	4.61
Average path length covered	Robot-1 (Start-1)	**173.37**	**181.03**	**4.42**
	Robot-2 (Start-2)	**167.26**	**174.36**	**4.25**
	Robot-3 (Start-3)	**165.27**	**173.17**	**4.78**
	Robot-4 (Start-4)	**153.87**	**160.61**	**4.38**

Table-7.9 (Time taken by the robots during experimental and simulation analysis by the proposed navigation system shown in Figures 7.13 and 7.18)

No. of Runs	Robot No.	Time taken by the robots during simulation analysis (in 'sec')	Time taken by the robots during experimental analysis (in 'sec')	% of error
1	Robot-1 (Start-1)	16.24	16.93	4.22
	Robot-2 (Start-2)	21.70	22.71	4.62
	Robot-3 (Start-3)	19.37	20.25	4.56
	Robot-4 (Start-4)	16.99	17.85	5.06
2	Robot-1 (Start-1)	15.36	16.22	5.60
	Robot-2 (Start-2)	23.14	24.35	5.25
	Robot-3 (Start-3)	19.69	20.79	5.61
	Robot-4 (Start-4)	18.44	19.32	4.77
3	Robot-1 (Start-1)	15.62	16.31	4.43
	Robot-2 (Start-2)	22.90	24.13	5.37
	Robot-3 (Start-3)	20.30	21.36	5.22
	Robot-4 (Start-4)	17.60	18.52	5.26
4	Robot-1 (Start-1)	16.35	17.05	4.30
	Robot-2 (Start-2)	23.15	24.23	4.65
	Robot-3 (Start-3)	19.56	20.44	4.48
	Robot-4 (Start-4)	18.46	19.27	4.39
5	Robot-1 (Start-1)	16.39	17.26	5.34
	Robot-2 (Start-2)	22.87	23.93	4.63
	Robot-3 (Start-3)	20.26	21.22	4.73
	Robot-4 (Start-4)	18.04	18.82	4.30

6	Robot-1 (Start-1)	15.45	16.26	5.25
	Robot-2 (Start-2)	22.54	23.53	4.37
	Robot-3 (Start-3)	20.80	21.88	5.23
	Robot-4 (Start-4)	19.10	20.09	5.19
7	Robot-1 (Start-1)	15.52	16.38	5.57
	Robot-2 (Start-2)	21.17	22.14	4.56
	Robot-3 (Start-3)	20.38	21.29	4.50
	Robot-4 (Start-4)	17.98	19.01	5.74
8	Robot-1 (Start-1)	16.16	16.86	4.31
	Robot-2 (Start-2)	22.11	23.32	5.49
	Robot-3 (Start-3)	20.38	21.22	4.11
	Robot-4 (Start-4)	17.72	18.51	4.43
9	Robot-1 (Start-1)	15.25	15.92	4.39
	Robot-2 (Start-2)	22.70	24.01	5.77
	Robot-3 (Start-3)	19.76	20.85	5.49
	Robot-4 (Start-4)	17.10	17.96	5.05
10	Robot-1 (Start-1)	16.34	17.07	4.49
	Robot-2 (Start-2)	21.96	23.09	5.11
	Robot-3 (Start-3)	19.67	20.52	4.34
	Robot-4 (Start-4)	17.39	18.29	5.15
Average time taken by the robots	Robot-1 (Start-1)	**15.87**	**16.63**	**4.79**
	Robot-2 (Start-2)	**22.42**	**23.54**	**4.98**
	Robot-3 (Start-3)	**20.02**	**20.98**	**4.83**
	Robot-4 (Start-4)	**17.88**	**18.76**	**4.93**

Table-7.10 (Time taken by the robots during experimental and simulation analysis by the proposed navigation system shown in Figures 7.14 and 7.19)

No. of Runs	Robot No.	Time taken by the robots during simulation analysis (in 'sec')	Time taken by the robots during experimental analysis (in 'sec')	% of error
1	Robot-1 (Start-1)	23.09	24.15	4.58
	Robot-2 (Start-2)	20.20	21.14	4.68
	Robot-3 (Start-3)	17.38	18.23	4.86
	Robot-4 (Start-4)	16.98	17.83	4.97
2	Robot-1 (Start-1)	24.18	25.18	4.15
	Robot-2 (Start-2)	21.27	22.05	3.69
	Robot-3 (Start-3)	18.79	19.76	5.18
	Robot-4 (Start-4)	17.96	18.59	3.49
3	Robot-1 (Start-1)	23.79	24.78	4.18
	Robot-2 (Start-2)	19.80	20.65	4.26
	Robot-3 (Start-3)	17.42	18.35	5.31
	Robot-4 (Start-4)	17.27	18.13	4.95
4	Robot-1 (Start-1)	24.15	25.42	5.25
	Robot-2 (Start-2)	23.23	24.05	3.54
	Robot-3 (Start-3)	18.46	19.42	5.21
	Robot-4 (Start-4)	17.73	18.40	3.78
5	Robot-1 (Start-1)	23.31	24.40	4.70
	Robot-2 (Start-2)	23.06	24.05	4.30
	Robot-3 (Start-3)	19.03	19.89	4.51
	Robot-4 (Start-4)	18.01	18.74	4.13

6	Robot-1 (Start-1)	22.58	23.72	5.03
	Robot-2 (Start-2)	21.63	22.58	4.40
	Robot-3 (Start-3)	17.18	18.01	4.82
	Robot-4 (Start-4)	17.58	18.46	4.98
7	Robot-1 (Start-1)	22.62	23.72	4.86
	Robot-2 (Start-2)	19.87	20.62	3.79
	Robot-3 (Start-3)	19.10	20.15	5.51
	Robot-4 (Start-4)	17.61	18.35	4.19
8	Robot-1 (Start-1)	24.02	25.10	4.49
	Robot-2 (Start-2)	20.34	21.23	4.38
	Robot-3 (Start-3)	18.89	19.78	4.69
	Robot-4 (Start-4)	17.22	17.98	4.38
9	Robot-1 (Start-1)	21.60	22.76	5.37
	Robot-2 (Start-2)	20.72	21.57	4.11
	Robot-3 (Start-3)	17.68	18.48	4.54
	Robot-4 (Start-4)	17.78	18.43	3.67
10	Robot-1 (Start-1)	21.98	23.11	5.13
	Robot-2 (Start-2)	21.96	22.96	4.56
	Robot-3 (Start-3)	17.18	18.12	5.48
	Robot-4 (Start-4)	17.43	18.26	4.78
Average time taken by the robots	Robot-1 (Start-1)	**23.13**	**24.13**	**4.77**
	Robot-2 (Start-2)	**21.21**	**22.09**	**4.17**
	Robot-3 (Start-3)	**18.11**	**19.02**	**4.96**
	Robot-4 (Start-4)	**17.56**	**18.32**	**4.33**

The average of 20 runs is considered for each simulation and experimental result. The best results are presented in simulation and experimental graph. The performance of the experimental results have been verified by the simulation results in terms of average path lengths and time taken by single and multiple mobile robots, presented in Tables7.3-7.10 respectively. From the experimental analysis, it has been clearly seen that the paths traced experimentally followed closely to that traced by robots during simulation mode. The maximum percentage of deviation between experimental and simulation results for single and multiple robots in terms of path lengths are 4.99% and 4.98% respectively. Similarly, the maximum percentage of deviation between experimental and simulation results in terms of time taken by single and multiple robots are 4.97% and 4.96% respectively.

7.6 Comparison of the Developed CS-ANFIS Hybrid Controller with other Navigational Controllers

In this part, a comparative study has been done between the proposed navigational algorithm and other intelligent navigational controllers in simulation mode. Firstly, we have replicated the environment as illustrated by the researchers [148,90,139]. In this environment, the robot moves from the source position to the target position with the help of CS-ANFIS navigational controller. The CS-ANFIS path planner is embedded in the controller of the mobile robot. After successful completion of the navigational task, it has been compared with the path as illustrated by the authors [148,90, 139]. During exercises, utmost care has been taken to replicate the environment as that of the environment represented in the Figures 7.20, 7.21 and 7.22. The size dimension of simulation platforms are considered as no. of units and each unit is in millimeter (mm).

a) A neuro-fuzzy technique for mobile robot navigation has been proposed by Zhang et al.[148]. In this method, the result obtained from fuzzy logic is used by the neural network to make navigation decisions. Here, both the navigational controllers are implemented in an environment, where the robot cannot see the goal directly due to the presence of obstacles between them. The obstacle avoidance behavior rules (shown in the Table-4.3) are embedded to cope up with the situations in order to avoid obstacles successfully. From the simulation results, it has been observed that in both the cases the robot is successfully avoiding obstacles while seeking for a target and in the meantime, it can be noted that the path length (in ‘cm’) covered by the robot as shown in Figure

7.20(a) is more compared to the current navigation technique. Figures 7.20(a) and (b) show the path generated by the robot using the neuro-fuzzy technique [148] and CS-ANFIS hybrid technique respectively.

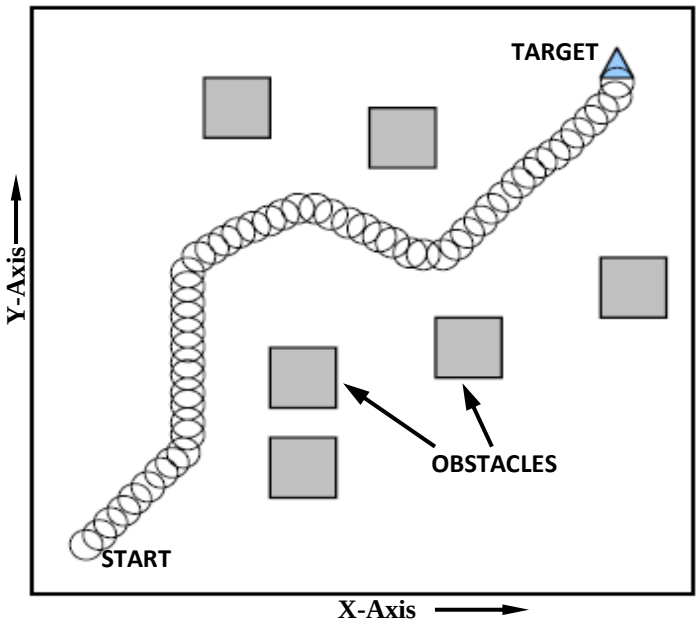


Figure 7.20(a) Path framed for the robot using Neuro-Fuzzy technique [148].

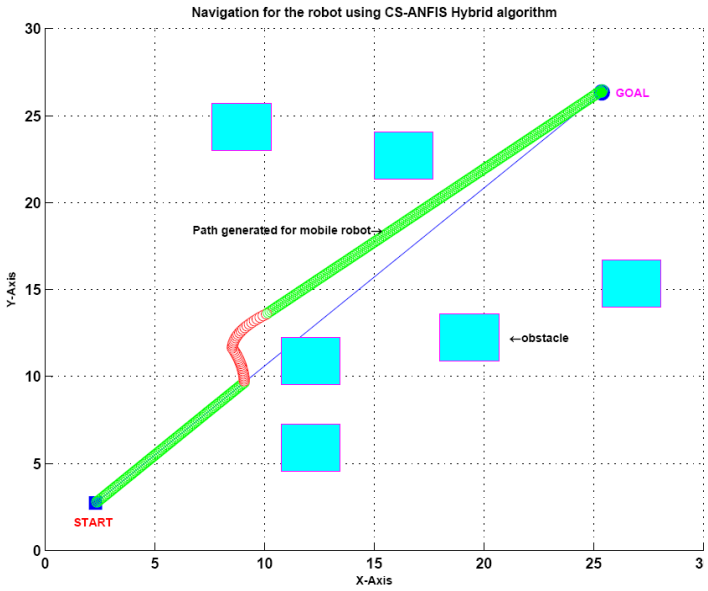


Figure 7.20(b) Path framed for the robot using CS-ANFIS Hybrid technique.

b) Path planning for the mobile robot using fuzzy logic has been addressed by Cherroun and Boumehraz [90]. In this technique, various reactive behaviors are designed using a fuzzy logic approach. The simulation results demonstrated that the proposed fuzzy algorithm can effectively control the robot movement from its start position to the goal position without hitting with the obstacles. In this problem, both the path planners are applied to a maze environment, where the obstacles are randomly placed. The simulation results using fuzzy logic [90] and current navigational algorithm are depicted in Figures 7.21(a) and 7.21(b) respectively.

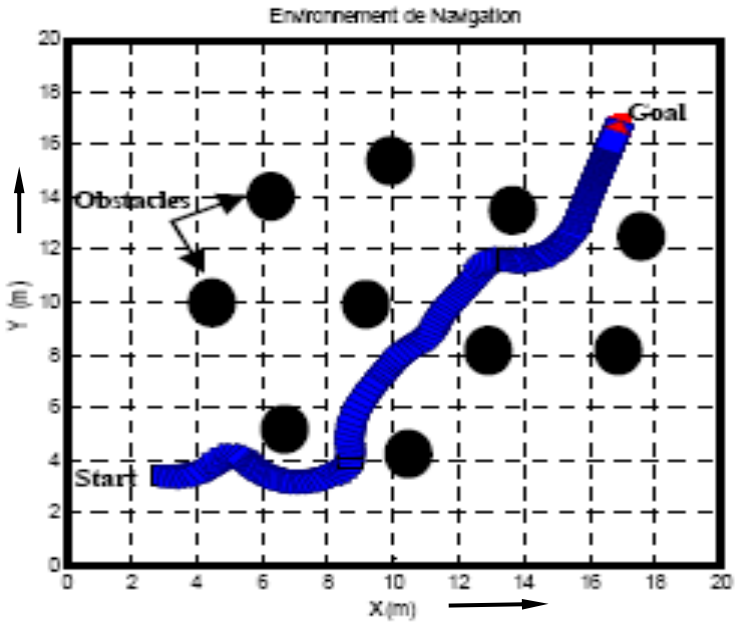


Figure 7.21(a) Path framed for the robot using Neuro-Fuzzy technique [90].

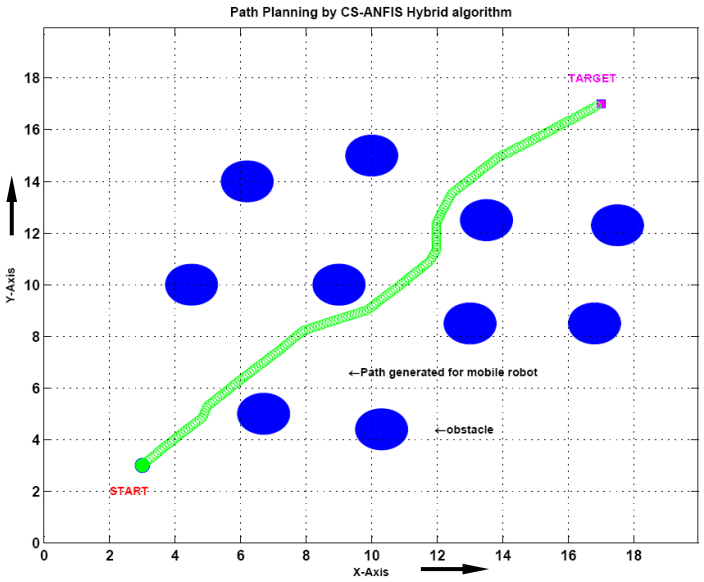


Figure 7.21(b) Path framed for the robot using CS-ANFIS Hybrid technique.

c) The mobile robot navigation based on the neuro-fuzzy approach has been presented by Joshi and Zaveri [139]. The performance of the navigation technique has been compared with neural network and fuzzy logic approaches. During the navigation, it has been noticed that the path generated by Joshi and Zaveri takes a long loop compared to results obtained using the current investigation. Figures 7.22(a) and 7.22(b) show the results obtained from [139] and current hybrid algorithm.

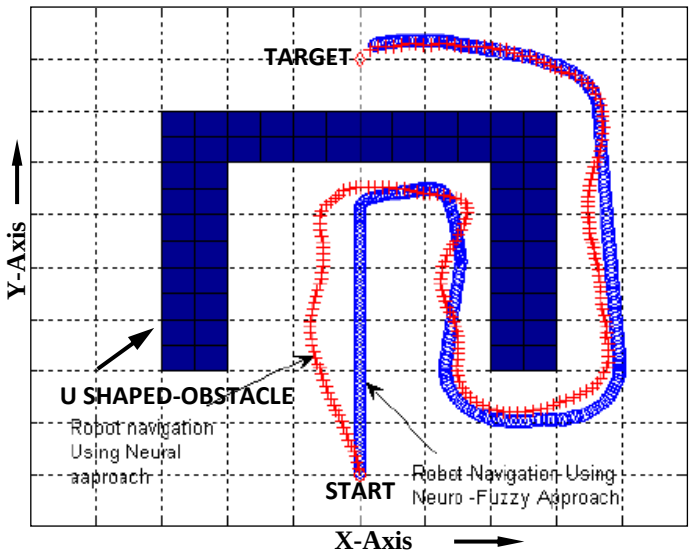


Figure 7.22(a) Path framed for the robot using Neuro-Fuzzy technique [139].

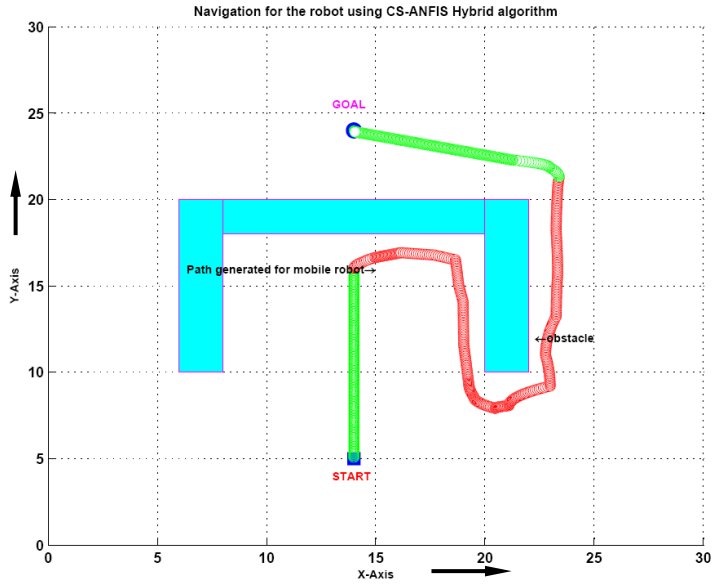


Figure 7.22(b) Path framed for the robot using CS-ANFIS technique.

In this case, both the motion planners are imposed to a U-shaped situation that would create the existence of the ‘dead-end’ concept. In this condition, the robot cannot see the goal directly due to the presence of long U-shaped wall between them. In the initial stage, the robot is heading towards the target. The wall following rules (shown in Table-ANFIS) will be implemented on the robot, until it is escaped from the trap situation and reaches the target with a smoother motion.

Table-7.11 (Comparison of simulation results between current investigation and other intelligent techniques)

SL. No.	**Environment types**	**Path length using current navigation system (‘in ‘cm')**	**Path length from the reference model (‘in ‘cm')**	**Percentage of deviation**
1	Obstacle avoidance behaviours by the robot, Figures 7.20(a) and 7.20(b)	8.12	9.41	13.7
2	Robot inside a cluttered environment Figures 7.21(a) and 7.21 (b)	6.21	6.83	9.07
3	Robot inside a U-shaped obstacle 7.22(a) and 7.22(b)	12.64	13.40	5.67

The performance of the comparative study is mainly measured on the basis of path length (in ‘cm’) and smoothness of the trajectory, which is presented in Table-7.11. From the above simulation analysis, it is clearly seen that the developed hybrid methodology can efficiently control the robot in any environment.

7.7 Summary

The following salient features are presented on the basis of the simulation and experimental results.

- The proposed CS-ANFIS navigational algorithm has been successfully implemented for single and multiple mobile robots in static environments.

- The Cuckoo search (CS) algorithm and least square estimation (LSE) method have been applied to optimize the premise and consequent parameters in ANFIS and to calculate the suitable steering angle of the mobile robots.
- In order to avoid collision problems between the mobile robots, collision prevention rules have been formulated and effectively embedded in the robot controller, using the Petri-Net model.
- The effectiveness, feasibility, and robustness of the proposed path planning algorithm have been tested in both the simulation and experimental modes, and they are in a good agreement.
- The maximum percentage of deviation between experimental and simulation results for single and multiple robots in terms of path lengths are 4.99% and 4.98% respectively. Similarly, the maximum percentage of deviation between experimental and simulation results in terms of time taken by single and multiple robots are 4.97% and 4.96% respectively.
- The current navigational methodology has been compared with other models of different authors, and it has been seen that the developed methodology yields better results compared to other techniques. The percentage of deviation is depicted in the Table 7.12.

In the next chapter IWO-ANFIS hybrid algorithm has been implemented to the path optimization problem of multiple mobile robots.

CHAPTER-8

8. ANALYSIS OF IWO-ANFIS HYBRID ALGORITHM FOR MULTIPLE MOBILE ROBOTS NAVIGATION

8.1 Introduction

In this present chapter, a new hybrid intelligent motion planning approach to mobile robot navigation has been described. In this new hybrid methodology the Invasive Weed Optimization (IWO) algorithm is used for training the premise parameters and the Least Square Estimation (LSE) method is used for training the consequent part of the Adaptive Neuro-Fuzzy Inference System (ANFIS). A brief description of the Invasive Weed Optimization and Adaptive Neuro-Fuzzy Inference System (ANFIS) techniques are discussed in the Chapter-4 and Chapter-6 respectively. In this proposed navigational model, different kinds of sensor extracted informations, such as front obstacle distance (FOD), right obstacle distance (ROD), left obstacle distance (LOD) and heading angle (HA) are given as input to the hybrid controller, for calculating the suitable steering angle (SA) for the robot. In order to avoid collision against one another, a set of collision prevention rules are also incorporated into each robot controller, using the Petri-Net model. The simulation results for single and multiple robots are verified by the real-time experimental results, using the Khepera-II and Khepera-III mobile robots to show the versatility and effectiveness of the proposed hybrid navigational algorithm. The results obtained using the proposed hybrid algorithm is validated by comparison with the results from other intelligent algorithms. Finally, it is proved that the proposed hybrid navigational controller can be implemented in the robot for navigation in any complex environment.

Intelligent soft computing techniques such as artificial neural network and fuzzy logic approaches are verified to be efficient and appropriate when implemented for variety of systems. Recent years both techniques have been rising interest and as a result, neuro-fuzzy techniques have been developed. There are many neuro-fuzzy models for the mobile robot path planning discussed by many researchers [123,124,129,133]. The ANFIS is a hybrid machine learning intelligent system, which takes the advantages of

both artificial neural network and fuzzy inference system [149-150]. The training and updating of the premise and conclusion parameters in ANFIS is one of the major problems. Gradient descent algorithm usually determines the fuzzy membership function's antecedent parameters. However, the calculation of the gradients in each step becomes difficult and may lead to the local minimum and due to this the adaptive system, affects the accuracy.

To overcome these problems, in this study an approach benefiting from the combination of adaptive network based fuzzy inference system (ANFIS) and Invasive Weed Optimization (IWO) algorithm is proposed to solve the navigational problem of a mobile robot. In this proposed hybrid algorithm, the IWO technique is introduced to enhance the performance of ANFIS by tuning the membership function parameters and subsequently minimizing the root mean square error (RMSE). The Invasive Weed Optimization (IWO) is a metaheuristic search algorithm proposed by Mehrabiana and Lucas in 2006 [209]. This optimization algorithm is based on the colonizing behaviour of weeds. An important feature of the IWO algorithm over GA and PSO is that, by tuning the important parameters in the IWO, the initial and final standard deviation, a high-level control in the convergence rate and accuracy of the algorithm are obtained.. However, to our knowledge, the IWO-ANFIS hybrid algorithm has not been implemented for mobile robot navigation till date.

8.2 Architecture of IWO-ANFIS Hybrid Controller for Multiple Mobile Robots Navigation

As specified earlier an ANFIS and IWO based obstacle avoidance scheme are used for path planning of mobile robot in cluttered environments. The main objective of this chapter is to predict the optimized steering angle for the robot using IWO based ANFIS navigational controller. In this proposed navigational model, different kinds of sensor extracted informations, such as front obstacle distance (FOD), right obstacle distance (ROD), left obstacle distance (LOD) and heading angle (HA) are given as input to the hybrid controller, in order to calculate the suitable steering angle (SA) for the robot. The detail navigation architecture for the ANFIS model is previously discussed in the Chapter-4. Figure 8.1 shows the schematic architecture for IWO-ANFIS.

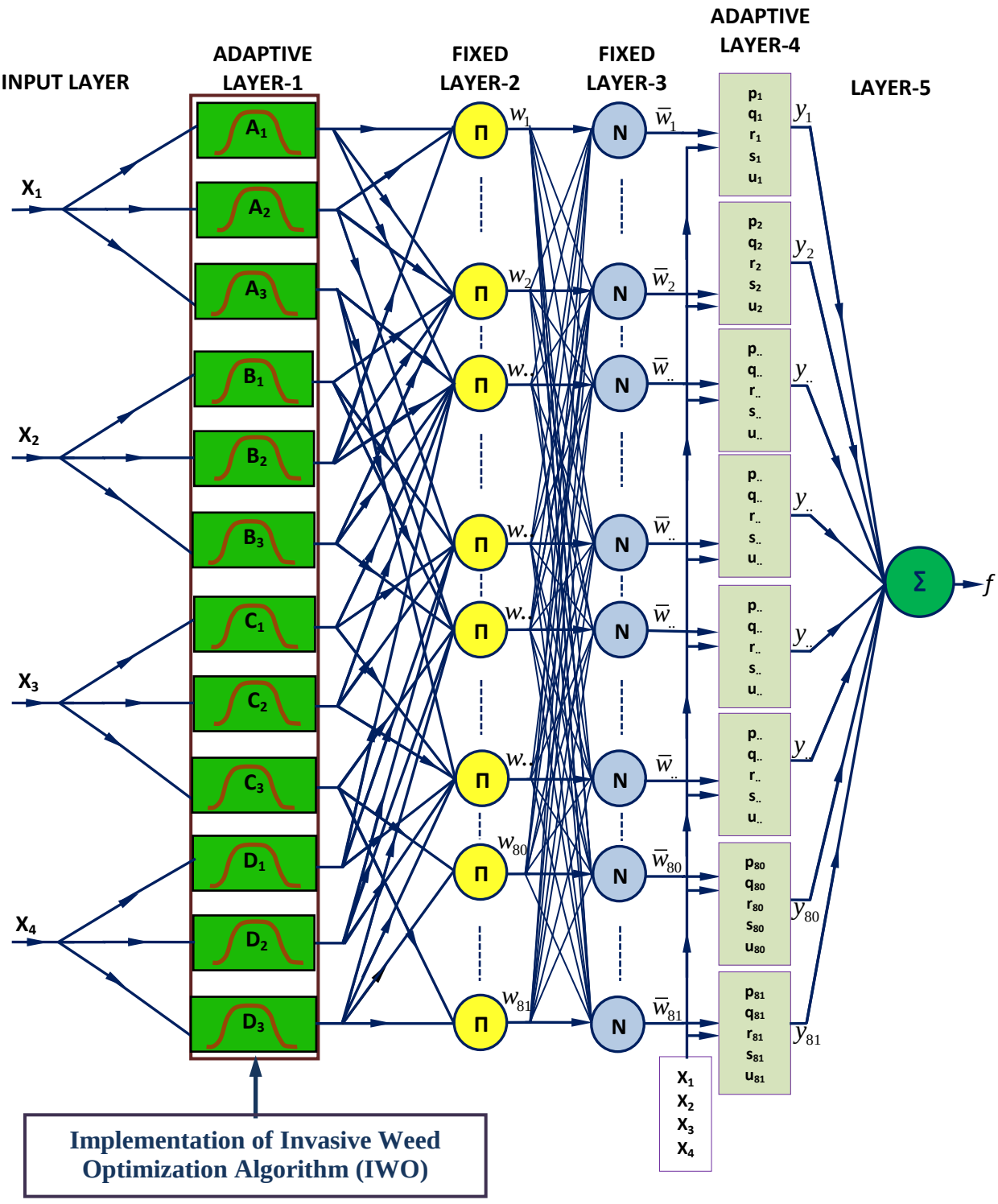


Figure 8.1 IWO-ANFIS hybrid controller for Mobile robot navigation.

Implementation of the IWO algorithm for training ANFIS

The detailed membership function parameters of ANFIS are discussed in previous chapter.

The coding structure of a seed is shown as following:

a_{i1}	b_{i1}	c_{i1}	a_{i2}	b_{i2}	c_{i2}	a_{i3}	b_{i3}	c_{i3}	a_{i4}	b_{i4}	c_{i4}	.....	a_{nj}	b_{nj}	c_{nj}

Where the parameters a_{ij}, b_{ij}, and c_{ij} control the centre, width, and slope of the Bell-shaped membership function.

i=1,2,3,...n. (Total number of membership functions for input parameters)

j=1....4. (Number of input parameters)

The initial values of the seeds are randomly generated in the first generation. The important step in implementing IWO algorithm is to select the suitable objective function, which is used to determine the fitness of each seed. Here, we have used same the objective function (Equation no. 7.1) as previously used in CS-ANFIS hybrid algorithm.

Here, the number of population is set to 20 for considering excessive running time. As each seed has 36 parameters, the colony size is [20X36] elements. The number of membership function parameters present in each seed will specify a real-value of all possible premise parameters of the fuzzy system. In the IWO algorithm the objective function is determined for each seed and sorted the population in the ascending order, this process continues until the maximum number of plants produced. Eliminate the plants having a lower rank in the population. Sort the population according to the objective function values, the seed having minimum objective function value is selected as the optimal point. The IWO algorithm runs until the maximum number of iterations, or minimum value of the objective function criteria is attainable. It is noticed that the new hybrid model can save the computation time in adjusting all parameters of ANFIS system. The flow chart of the sequential combination of IWO and ANFIS is given in Figure 8.2.

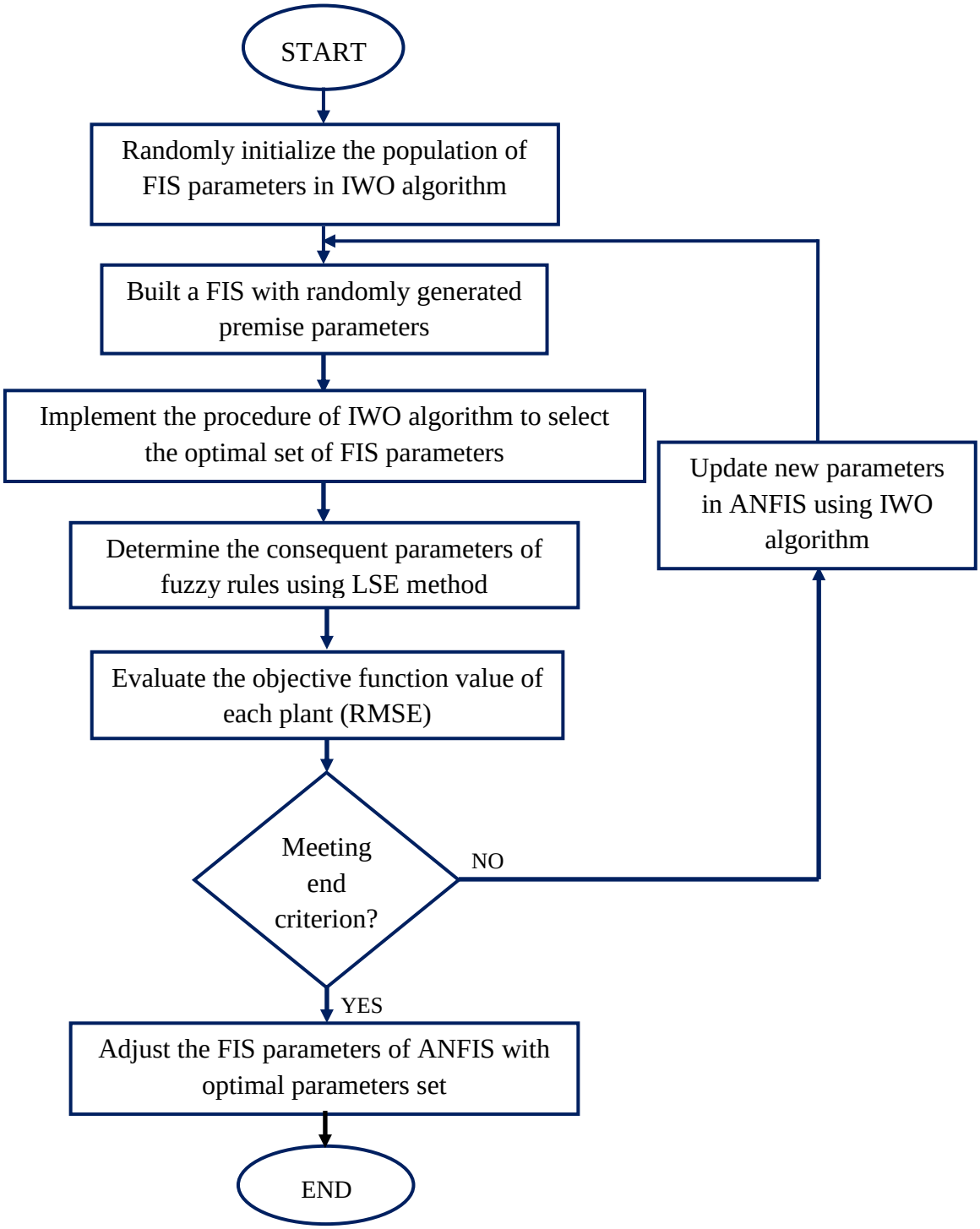


Figure 8.2 Flow chart diagram of the proposed hybrid algorithm.

Steps of IWO-ANFIS hybrid algorithm:

Step-1: Generate the initial population of host seeds in the search space, each representing a set of antecedent parameters of ANFIS in sequence.

Step-2: Determine the membership function value, the firing strength, and the normalized firing strength, using equations 4.5, 4.6, and 4.7 (Chapter-4) respectively.

Step-3: The consequent parameters are calculated using the LSE method and these are used to determine the overall ANFIS output, using equation 4.9 of Chapter-4.

Step-4: In this step the root mean square error (RMSE) between the predicted output and actual training data for each seed (plant) is calculated, which is also the objective function value of the n^{th} seed.

Step-5: The RMSE values of each seed (plant) are compared; the plant that has the smallest RMSE is selected as the best one.

Step-6: Eliminate the plants with lower fitness to reach the maximum number of plants.

Step-7: The iteration ends when the maximum numbers of iteration criterions are satisfied, and the best values of the antecedent parameters with consequent parameters are identified.

The optimized antecedent parameters of the fuzzy system obtained from the IWO algorithm are used to train the ANFIS navigational controller in order to determine the suitable steering angle of a mobile robot for any reactive condition. The various reactive behaviours and training patterns of ANFIS model for mobile robot path planning have been discussed in the Chapter-4.

8.3 Simulation Results and Discussion

To validate the performance of the current hybrid navigational controller, we have demonstrated the simulation exercises for both single and multiple robots to verify the robot trajectory in partially or completely unknown environments. The best parametric value of the IWO algorithm has been selected after performing a series of simulation experiments and is given in the Table-6.1.

The simulated robot path planning algorithm has been developed by putting the robot, goal and obstacles in different positions in the environment. In the current navigation model, we have mainly considered the three reactive behaviors: target seeking, wall

following and obstacle avoiding. In order to avoid an obstacle and generate the trajectory of the mobile robot, the controller needs to decide the steering direction in which the robot should move to reach the target. To decide the correct reactive behavior, the proposed navigation model should consider all accounts of information. When a robot is close to an obstacle, it must change its steering angle to avoid the obstacle. If a mobile robot senses a target, it will decide whether it can reach that target, i.e. it will judge whether there are any obstacles that will obstruct its path or not. If the path leading to the goal is clear, the robot will turn and proceed towards the destination. Wall following behavior will be activated when the robot is sensed an obstacle in front of it and the target is present on the opposite side of the obstacle; then the robot will travel towards the goal by following the left or right side of the wall to reach the target.

8.3.1 Simulation results for a Single Mobile Robot

A comparative simulation study for single mobile robot considering various reactive behaviors have been made between the IWO-ANFIS and ANFIS techniques as shown in Figures 8.3-8.8. The performance of the techniques is mainly measured, based on the path length given in Table-8.1. It can be noticed that the IWO-ANFIS hybrid algorithm yields better results in terms of smoothness, compared to the ANFIS technique. In the simulation graph X and Y axes are taken as 300 units and each unit is equal to 0.2mm.

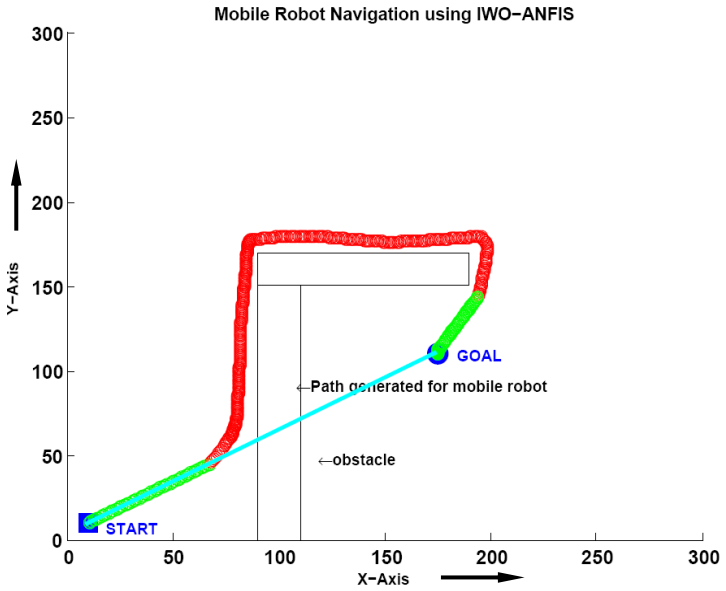


Figure 8.3 Wall following behavior by a single robot using IWO-ANFIS Hybrid technique.

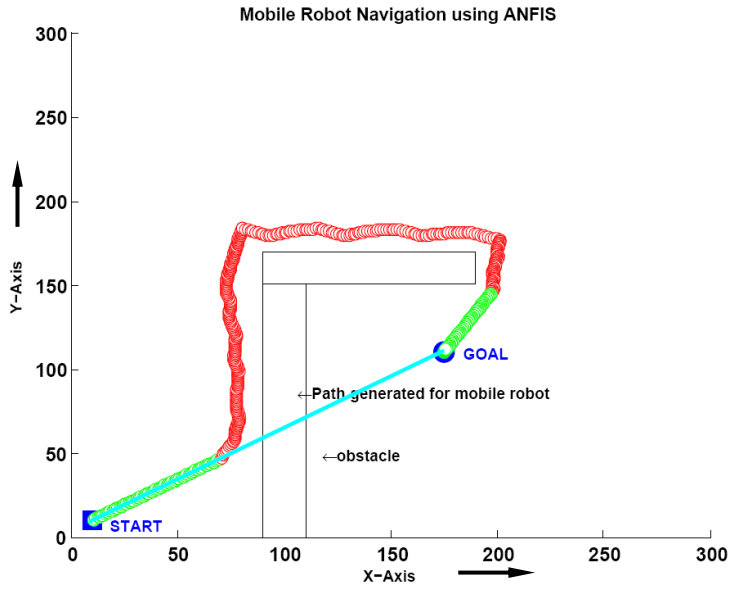


Figure 8.4 Wall following behavior by a single robot using ANFIS technique.

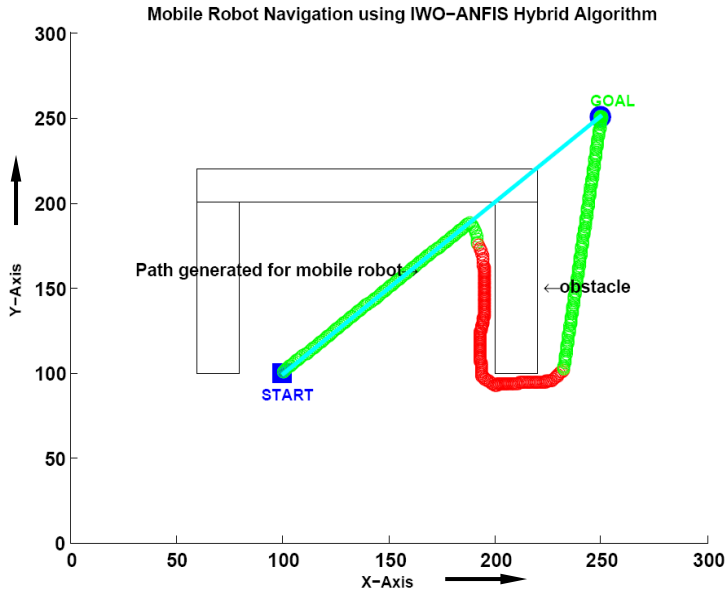


Figure 8.5 Single robot escaping from a trap condition using IWO-ANFIS hybrid technique.

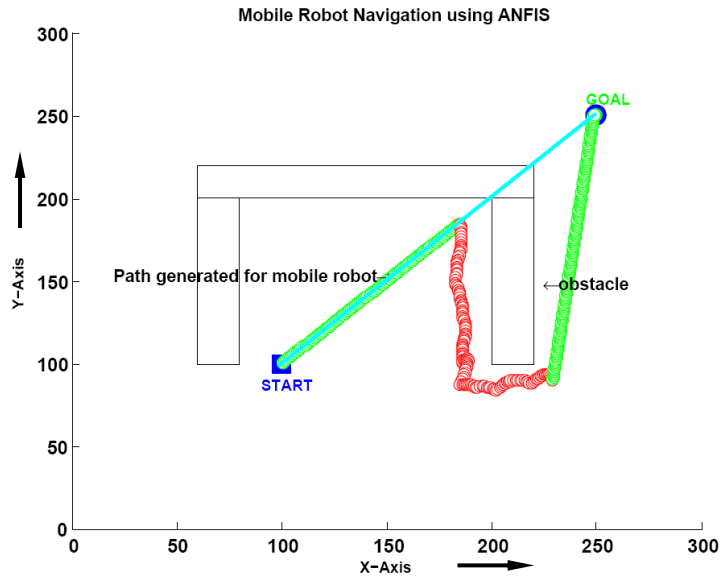


Figure 8.6 Single robot escaping from a trap condition using ANFIS hybrid technique.

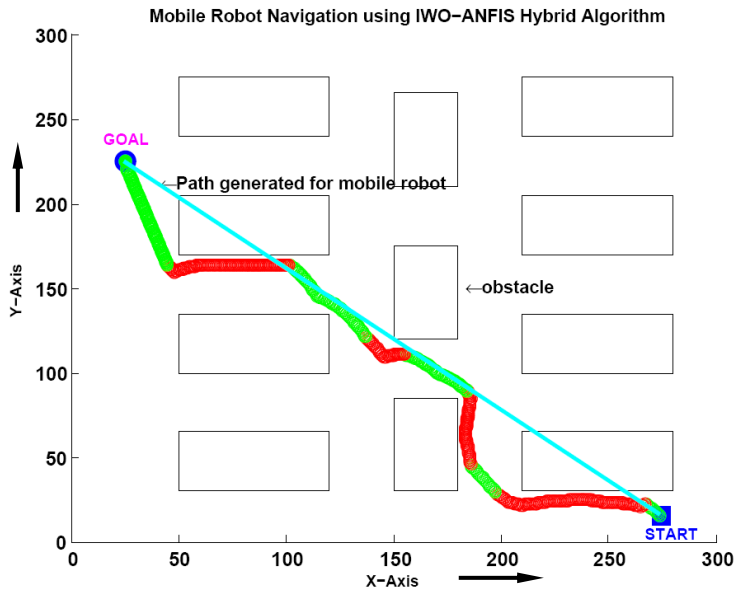


Figure 8.7 Single robot navigating inside a cluttered environment using IWO-ANFIS hybrid technique.

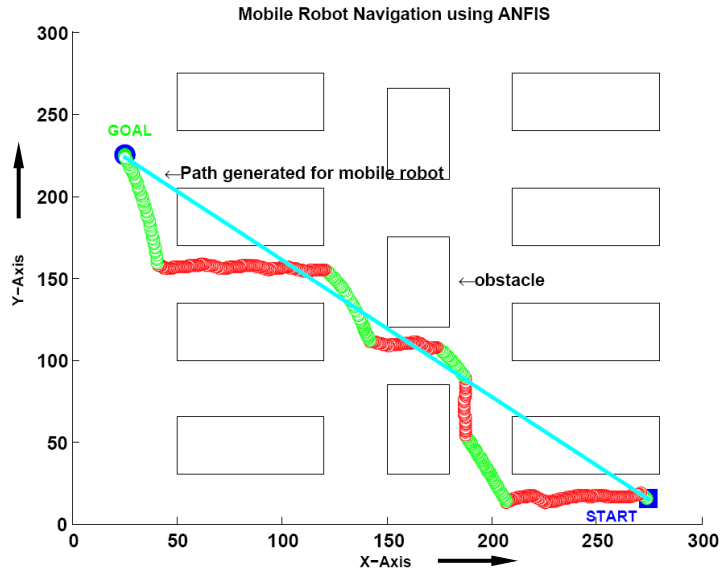


Figure 8.8 Single robot navigating inside a cluttered environment using ANFIS hybrid technique.

Table-8.1 (Comparison of IWO-ANFIS and ANFIS results in terms of path length)

SL. No.	Path covered by the Robot using IWO-ANFIS (in 'cm')	Path covered by the Robot using ANFIS (in 'cm')	% of deviation
1	9.30 (Fig. 8.3)	10.12 (Fig. 8.4)	8.10
2	9.80 (Fig. 8.5)	10.30 (Fig. 8.6)	4.85
3	10.20 (Fig. 8.7)	11.02 (Fig. 8.8)	7.44

8.3.2 Simulation Results for Multiple Mobile Robots

The simulation results for multiple mobile robots or inter-collision avoidance among the robots have been performed using the Petri-Net theory. The details of Petri-Net theory for multiple mobile robots navigation has been discussed in the Chapter-7. The simulation experiments involve four robots with four goals at different positions of the environment. The simulation results for multiple robots navigation in various environments are presented in the Figures 8.9 to 8.11. In the results, various reactive behaviors are demonstrated by the robots. It can be observed that the robots have effectively reached their targets without any collision with one another and the obstacles. In this simulation graph X and Y axes are taken as 40 units and each unit is equal to 20mm.

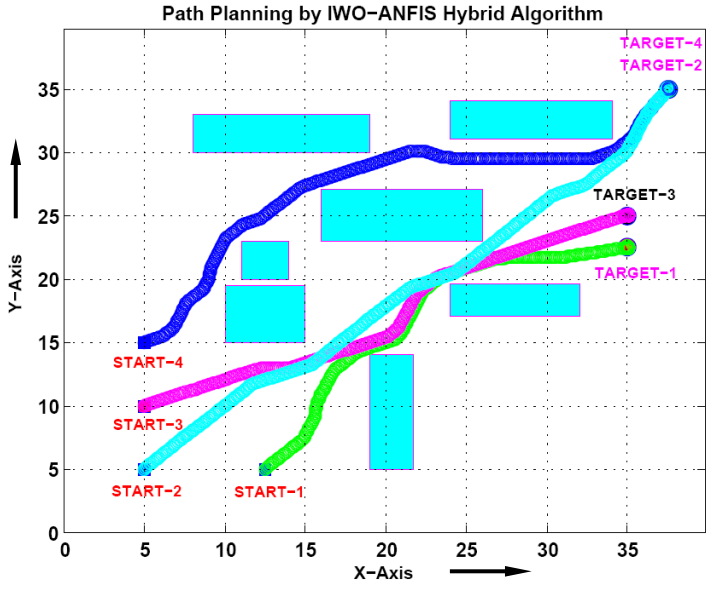


Figure 8.9 Obstacle avoidance and wall following behavior by multiple mobile robots using IWO-ANFIS hybrid technique.

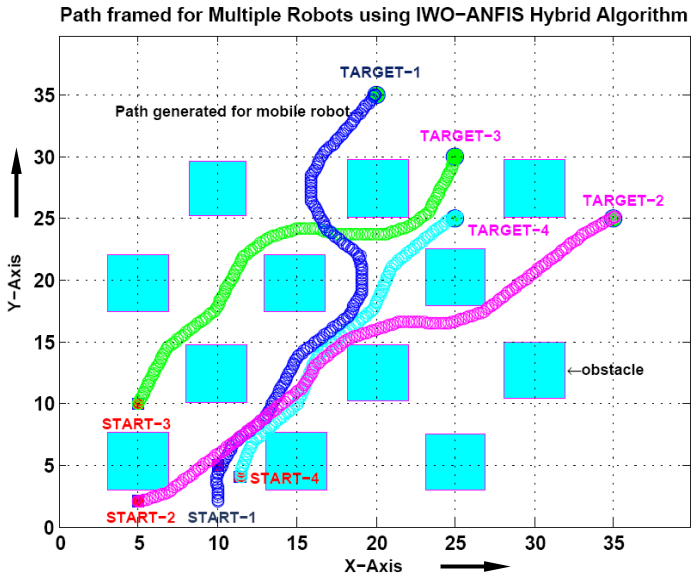


Figure 8.10 Multiple mobile robots navigating in a maze environment using IWO-ANFIS hybrid technique.

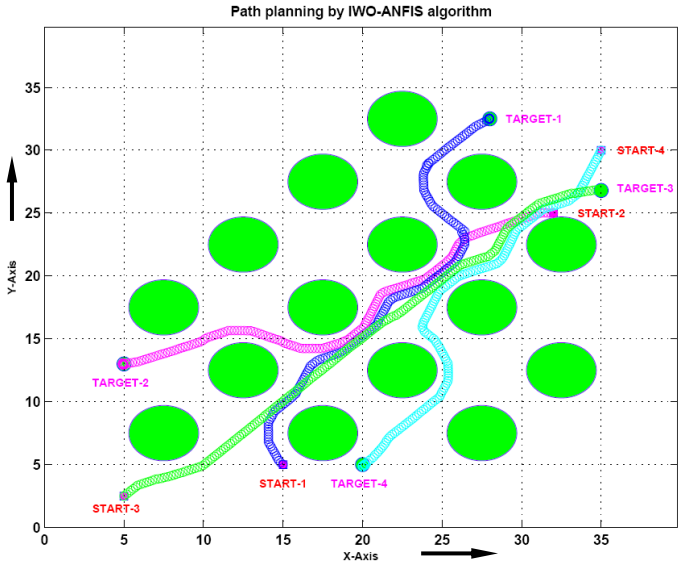


Figure 8.11 Path planning of multiple mobile robots in a maze environment using IWO-ANFIS hybrid technique.

8.4 Experimental Validation with the Simulation Results

The effectiveness of the proposed algorithm has been verified through a series of real time test on Khepera-II and Khepera-III mobile robots. The experimental results are conducted for both single and multiple mobile robots. The details specification of the robots are given in the Appendix-A. During the experiment, the robots perform the various reactive behaviours like wall following, obstacle avoidance, and target seeking as per the training of ANFIS data. The results for single and multiple robots are shown in Figures 8.12-8.16, which is already verified in the simulation mode, and has been verified experimentally to show the efficacy of the developed path planner. During experimentation, paths followed by the robot to reach the target are traced. The average of 20runs is taken into consideration for each simulation and experimental results. The best results are shown in the simulation and experimental graph. The path length and time taken by the robots to reach the target have been recorded and presented in Tables 8.2-8.11. It has been observed that the experimental results are closer to the simulation results, which validate the proposed navigation methods. The maximum percentage of deviation between experimental and simulation results for single and multiple robots in terms of path length are 4.93% and 4.97% respectively. Similarly, the maximum percentage of deviation between experimental and simulation results for single and multiple robots in terms of time taken are 5.10% and 5.17% respectively.

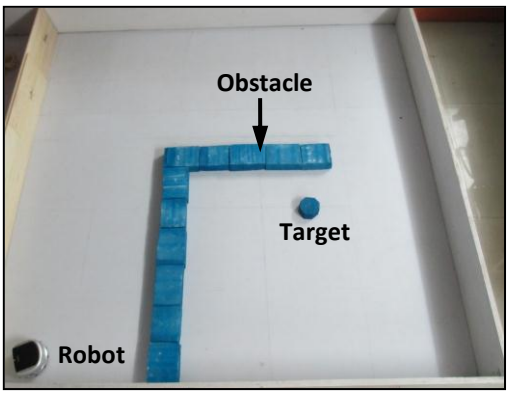


Figure 8.12 (a)

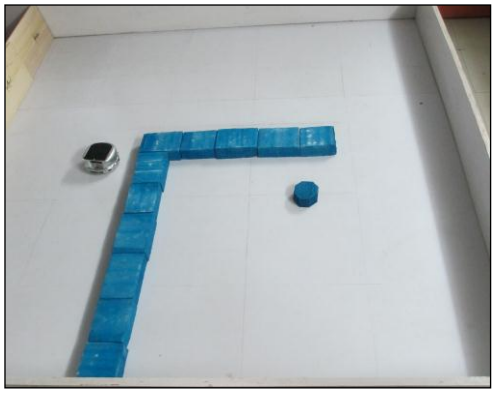


Figure 8.12 (b)

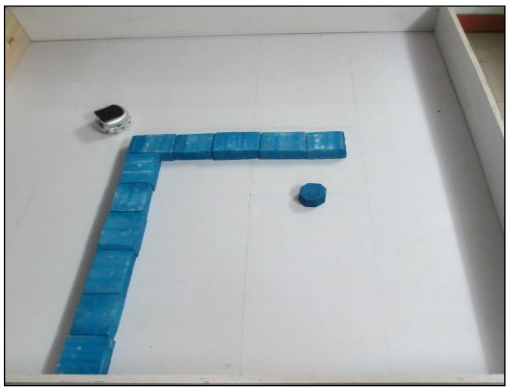


Figure 8.12 (c)

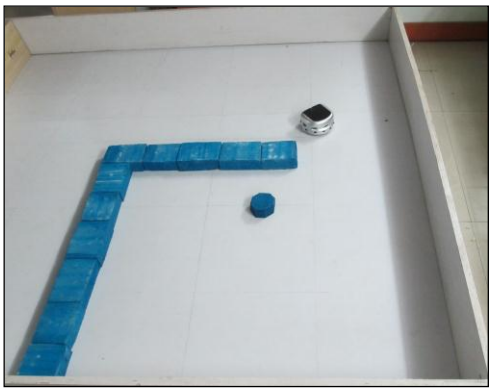


Figure 8.12 (d)

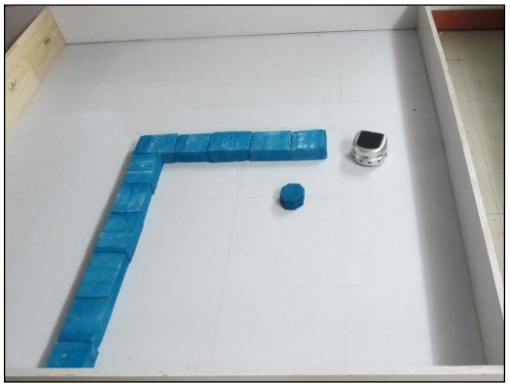


Figure 8.12 (e)

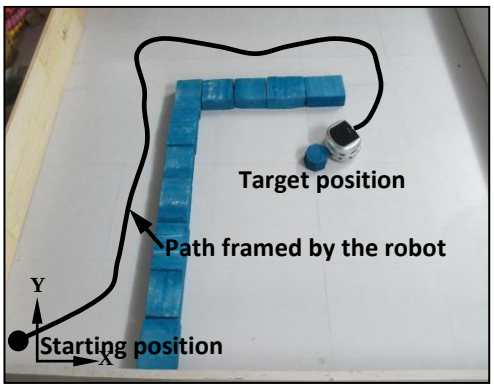


Figure 8.12 (f)

Figure 8.12 Experimental results for navigation of mobile robot in the environment shown in Figure 8.3.

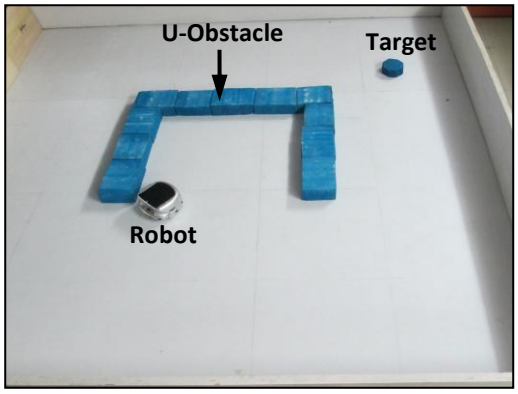


Figure 8.13 (a)

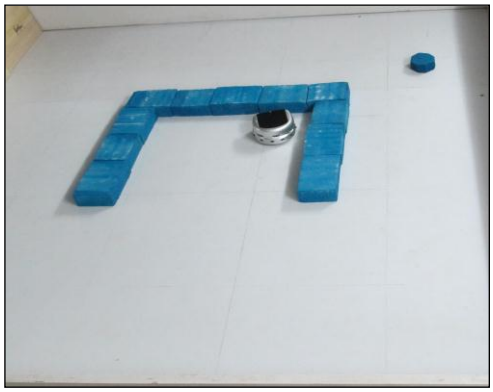


Figure 8.13 (b)

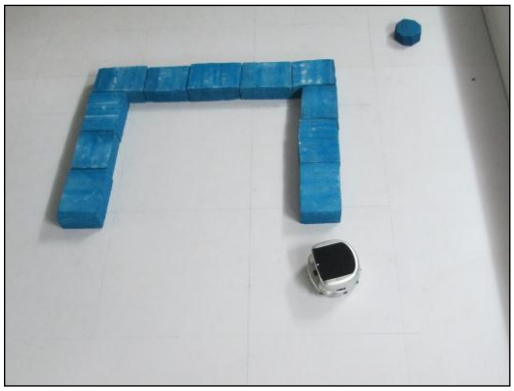


Figure 8.13 (c)

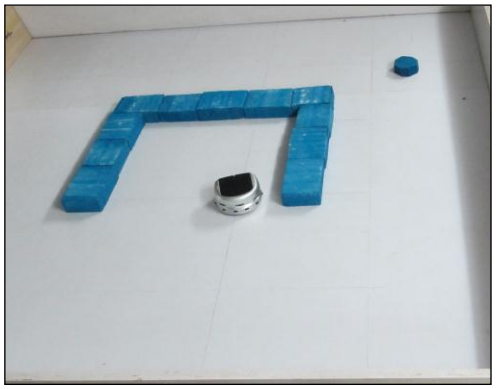


Figure 8.13 (d)

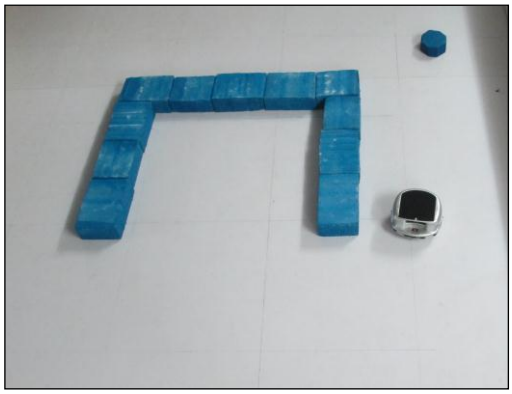


Figure 8.13 (e)

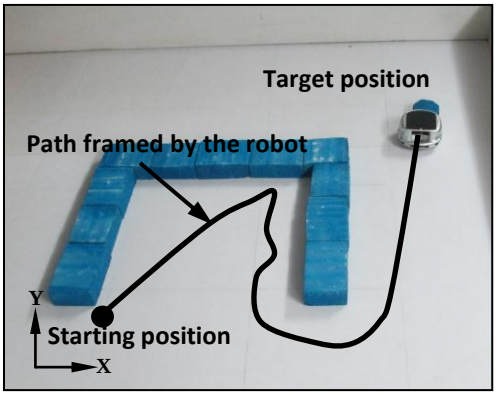


Figure 8.13 (f)

Figure 8.13 Experimental results for navigation of mobile robot in the environment shown in Figure 8.5.

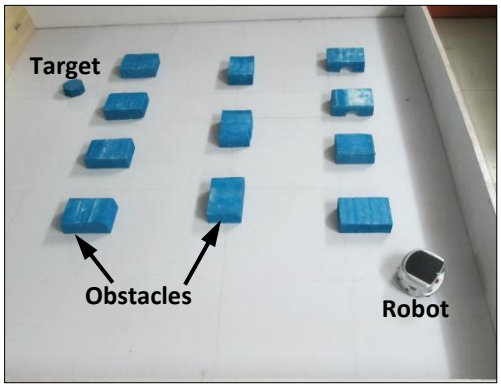


Figure 8.14 (a)

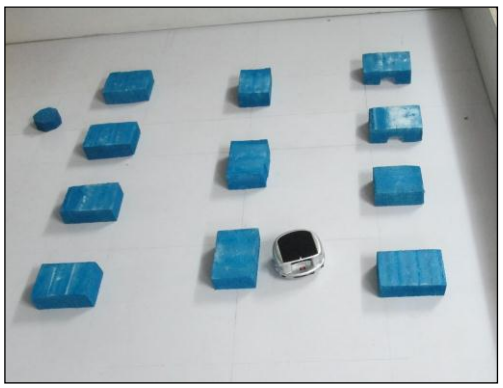


Figure 8.14 (b)

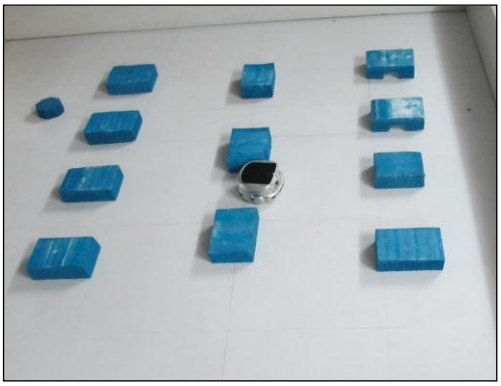


Figure 8.14 (c)

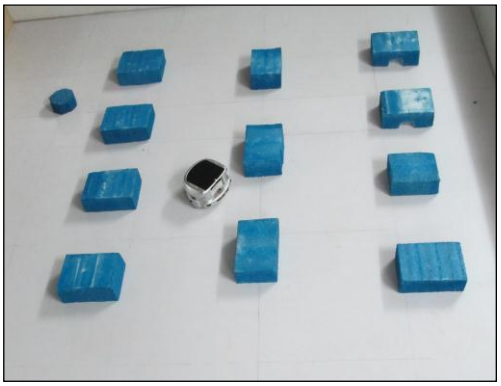


Figure 8.14 (d)

Figure 8.14 (e)

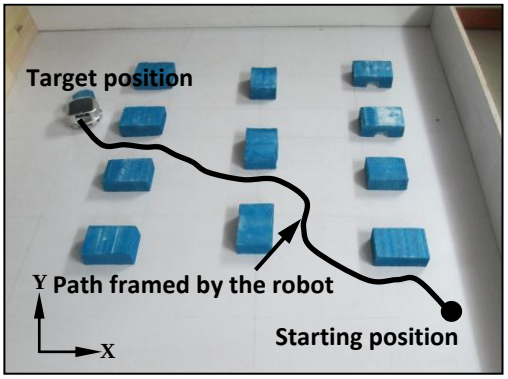


Figure 8.14 (f)

Figure 8.14 Experimental results for navigation of mobile robot in the environment shown in Figure 8.7.

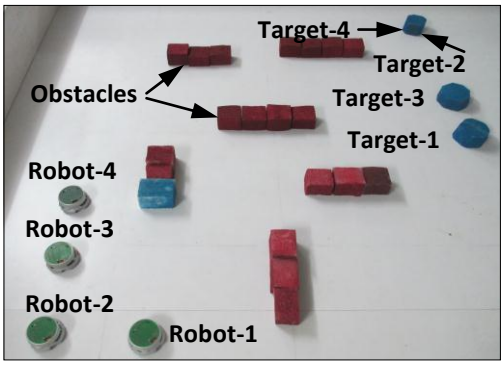


Figure 8.15 (a)

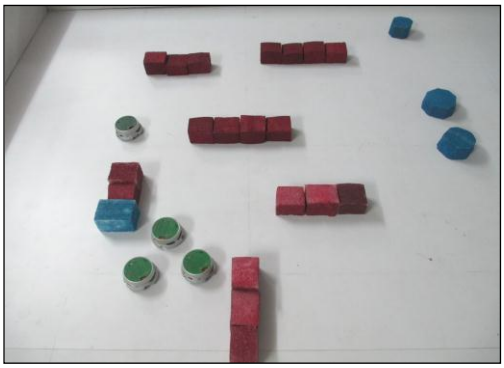


Figure 8.15 (b)

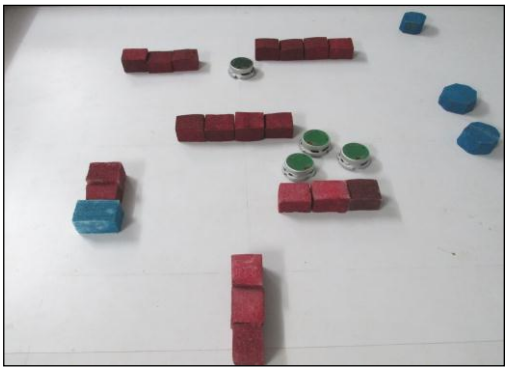


Figure 8.15 (c)

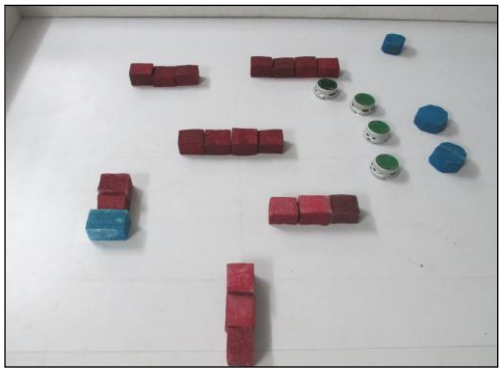


Figure 8.15 (d)

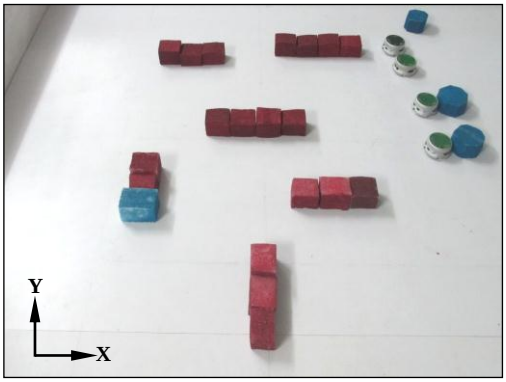


Figure 8.15 (e)

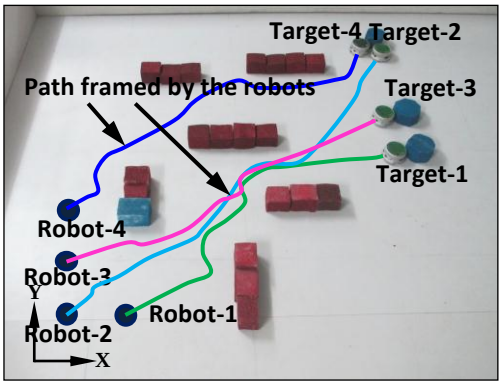


Figure 8.15 (f)

Figure 8.15 Experimental results for navigation of mobile robots in the environment shown in Figure 8.9.

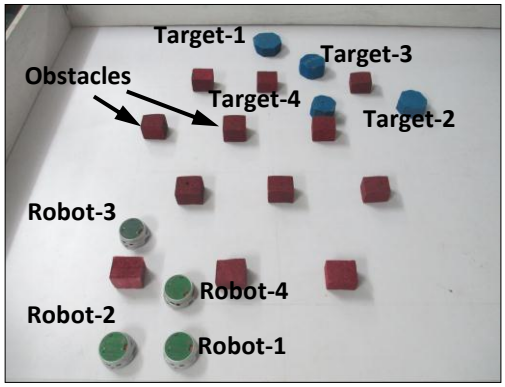


Figure 8.16 (a)

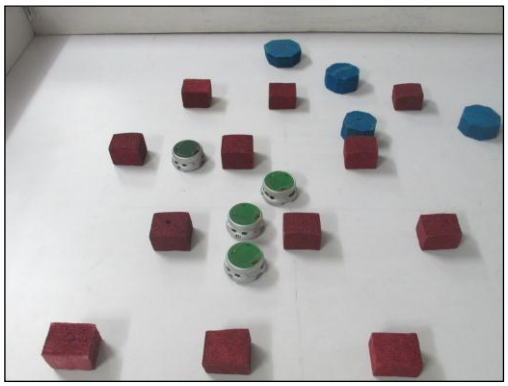


Figure 8.16 (b)

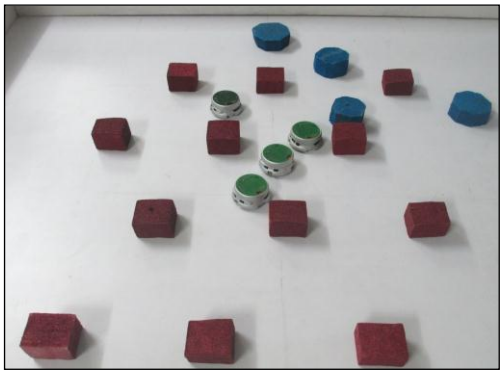


Figure 8.16 (c)

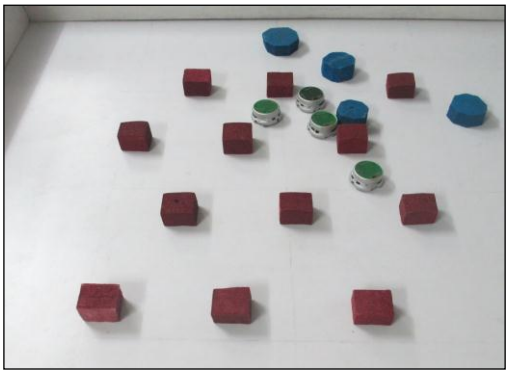


Figure 8.16 (d)

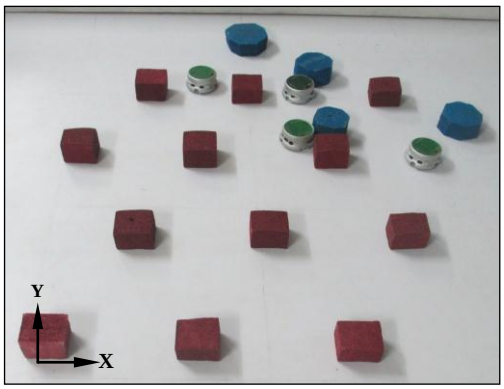


Figure 8.16 (e)

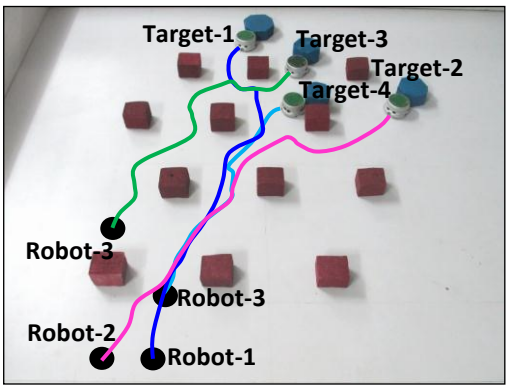


Figure 8.16 (f)

Figure 8.16 Experimental results for navigation of mobile robots in the environment shown in Figure 8.10.

Table-8.2 The Path travelled by the single robot during simulation and experimental analysis by the proposed hybrid navigation system shown in Figures 8.3 and 8.12 respectively.

No. of Runs	**Path length covered during simulation (in 'cm')**	**Path length covered during the experiment (in 'cm')**	**% of error**
1	171.80	179.44	4.45
2	172.23	180.34	4.71
3	169.74	179.82	5.94
4	173.52	181.64	4.68
5	170.92	179.25	4.87
6	168.34	177.56	5.47
7	174.22	182.02	4.48
8	170.51	179.67	5.38
9	174.98	183.48	4.86
10	172.52	181.61	5.27
11	172.38	179.96	4.39
12	168.80	177.59	5.21
13	174.98	183.56	4.90
14	170.21	179.03	5.18
15	172.98	181.47	4.91
16	171.06	180.64	5.60
17	169.90	177.51	4.47
18	173.53	182.82	5.35
19	171.69	181.17	5.52
20	174.99	182.85	4.49
Average path length covered	**172.11**	**180.57**	**4.91**

Table-8.3 The Path travelled by the single robot during simulation and experimental analysis by the proposed hybrid navigation system shown in Figures 8.5 and 8.13 respectively.

No. of Runs	**Path length covered during simulation (in 'cm')**	**Path length covered during the experiment (in 'cm')**	**% of error**
1	173.01	180.39	4.27
2	174.71	183.55	5.06
3	176.14	184.60	4.80
4	173.62	182.31	5.01
5	175.74	185.10	5.33
6	176.91	184.72	4.41
7	174.68	183.69	5.16
8	176.80	185.19	4.74
9	174.60	183.73	5.23
10	175.64	183.86	4.68
11	173.60	182.02	4.85
12	176.05	185.38	5.30
13	175.93	185.24	5.29
14	174.57	182.39	4.48
15	175.85	184.25	4.78
16	176.41	184.68	4.69
17	173.37	182.45	5.23
18	175.34	183.40	4.60
19	172.90	182.01	5.27
20	174.82	184.17	5.35
Average path length covered	**175.03**	**183.66**	**4.93**

Table-8.4 The Path travelled by the single robot during simulation and experimental analysis by the proposed hybrid navigation system shown in Figures 8.7 and 8.14 respectively.

No. of Runs	**Path length covered during simulation (in 'cm')**	**Path length covered during the experiment (in 'cm')**	**% of error**
1	161.07	169.73	5.38
2	161.20	168.59	4.58
3	159.53	166.35	4.27
4	161.20	168.89	4.77
5	161.99	170.14	5.03
6	162.15	170.75	5.30
7	159.77	167.37	4.75
8	160.47	169.15	5.41
9	160.31	167.38	4.42
10	161.94	169.24	4.51
11	163.28	170.17	4.22
12	159.81	167.73	4.96
13	160.07	168.63	5.35
14	160.32	168.71	5.24
15	159.96	168.11	5.10
16	159.92	166.79	4.30
17	159.69	167.55	4.93
18	161.27	168.63	4.56
19	159.19	167.61	5.29
20	159.96	166.75	4.25
Average path length covered	**160.66**	**168.41**	**4.83**

Table-8.5 Time taken by the single robot during simulation and experimental analysis by the proposed hybrid navigation system shown in Figures 8.3 and 8.12 respectively.

No. of Runs	**Time taken by the robot during simulation (in 'sec')**	**Time taken by the robot during experiment (in 'sec')**	**% of error**
1	18.75	19.75	5.32
2	19.44	20.45	5.20
3	18.77	19.69	4.90
4	19.66	20.56	4.58
5	20.01	21.02	5.05
6	18.75	19.55	4.27
7	18.59	19.60	5.46
8	19.12	20.11	5.18
9	18.53	19.40	4.70
10	19.66	20.56	4.58
11	20.01	20.88	4.33
12	18.99	19.99	5.27
13	19.25	20.25	5.19
14	19.32	20.39	5.54
15	18.70	19.55	4.55
16	19.74	20.71	4.91
17	19.23	20.14	4.73
18	20.10	21.07	4.83
19	18.82	19.78	5.12
20	19.43	20.50	5.51
Average time taken by the robot	**19.24**	**20.20**	**4.96**

Table-8.6 Time taken by the single robot during simulation and experimental analysis by the proposed hybrid navigation system shown in Figures 8.5 and 8.13 respectively.

No. of Runs	**Time taken by the robot during simulation (in 'sec')**	**Time taken by the robot during experiment (in 'sec')**	**% of error**
1	20.20	21.30	5.45
2	22.24	23.23	4.44
3	20.22	21.23	5.02
4	21.50	22.66	5.39
5	22.06	23.25	5.37
6	19.87	20.96	5.49
7	20.02	21.14	5.60
8	21.23	22.31	5.08
9	21.01	22.20	5.66
10	21.76	22.77	4.65
11	21.52	22.46	4.40
12	21.49	22.70	5.63
13	22.55	23.78	5.46
14	20.79	21.76	4.66
15	19.10	20.04	4.95
16	21.23	22.13	4.25
17	22.48	23.51	4.56
18	22.02	23.21	5.39
19	20.26	21.30	5.10
20	21.96	23.14	5.41
Average time taken by the robot	**21.18**	**22.25**	**5.10**

Table-8.7 Time taken by the single robot during simulation and experimental analysis by the proposed hybrid navigation system shown in Figures 8.7 and 8.14 respectively.

No. of Runs	**Time taken by the robot during simulation (in 'sec')**	**Time taken by the robot during experiment (in 'sec')**	**% of error**
1	23.95	25.09	4.75
2	22.86	24.03	5.12
3	22.75	23.89	4.99
4	23.84	25.13	5.42
5	23.76	25.09	5.58
6	23.18	24.41	5.31
7	23.06	24.20	4.96
8	22.60	23.59	4.36
9	22.63	23.88	5.52
10	23.35	24.43	4.63
11	22.82	24.08	5.54
12	22.86	23.99	4.94
13	23.35	24.41	4.54
14	22.19	23.47	5.77
15	24.10	25.31	5.02
16	22.78	24.01	5.38
17	23.15	24.34	5.14
18	22.85	23.87	4.46
19	23.66	24.79	4.78
20	23.34	24.42	4.63
Average time taken by the robot	**23.15**	**24.32**	**5.04**

Table-8.8 The Path travelled by the robots during simulation and experimental analysis by the proposed hybrid navigation system shown in Figures 8.9 and 8.15 respectively.

No. of Runs	Robot No.	Path covered by the robots during simulation analysis (in 'cm')	Path covered by the robots during experimental analysis (in 'cm')	% error
1	Robot-1 (Start-1)	146.72	153.21	4.42
	Robot-2 (Start-2)	185.30	195.03	5.25
	Robot-3 (Start-3)	172.26	179.85	4.41
	Robot-4 (Start-4)	180.96	189.10	4.50
2	Robot-1 (Start-1)	149.61	156.78	4.79
	Robot-2 (Start-2)	185.12	194.98	5.33
	Robot-3 (Start-3)	170.38	179.14	5.14
	Robot-4 (Start-4)	184.52	192.84	4.51
3	Robot-1 (Start-1)	148.77	155.23	4.34
	Robot-2 (Start-2)	186.29	195.89	5.15
	Robot-3 (Start-3)	171.33	180.02	5.07
	Robot-4 (Start-4)	183.48	192.98	5.18
4	Robot-1 (Start-1)	150.27	158.01	5.15
	Robot-2 (Start-2)	186.38	194.01	4.09
	Robot-3 (Start-3)	172.06	179.69	4.43
	Robot-4 (Start-4)	185.34	194.86	5.14
5	Robot-1 (Start-1)	149.94	156.89	4.64
	Robot-2 (Start-2)	184.53	193.98	5.12
	Robot-3 (Start-3)	171.59	179.22	4.44
	Robot-4 (Start-4)	184.92	193.54	4.66

6	Robot-1 (Start-1)	150.93	157.69	4.48
	Robot-2 (Start-2)	185.77	195.31	5.14
	Robot-3 (Start-3)	173.04	180.59	4.36
	Robot-4 (Start-4)	186.15	195.98	5.28
7	Robot-1 (Start-1)	150.86	158.58	5.11
	Robot-2 (Start-2)	185.32	194.18	4.78
	Robot-3 (Start-3)	171.49	180.62	5.33
	Robot-4 (Start-4)	186.07	194.79	4.69
8	Robot-1 (Start-1)	147.92	156.21	5.60
	Robot-2 (Start-2)	185.93	194.83	4.79
	Robot-3 (Start-3)	169.95	179.98	5.90
	Robot-4 (Start-4)	182.44	192.55	5.54
9	Robot-1 (Start-1)	149.85	157.05	4.81
	Robot-2 (Start-2)	186.12	195.51	5.05
	Robot-3 (Start-3)	169.99	178.33	4.91
	Robot-4 (Start-4)	184.81	193.63	4.77
10	Robot-1 (Start-1)	147.20	154.26	4.79
	Robot-2 (Start-2)	186.26	194.49	4.42
	Robot-3 (Start-3)	170.39	177.77	4.33
	Robot-4 (Start-4)	181.55	191.44	5.45
Average path length covered	Robot-1 (Start-1)	**149.21**	**156.39**	**4.81**
	Robot-2 (Start-2)	**185.70**	**194.82**	**4.91**
	Robot-3 (Start-3)	**171.25**	**179.52**	**4.83**
	Robot-4 (Start-4)	**184.02**	**193.17**	**4.97**

Table-8.9 The Path travelled by the robots during simulation and experimental analysis by the proposed hybrid navigation system shown in Figures 8.10 and 8.16 respectively.

No. of Runs	Robot No.	Path covered by the robots during simulation analysis (in 'cm')	Path covered by the robots during experimental analysis (in 'cm')	% error
1	Robot-1 (Start-1)	189.52	199.12	5.07
	Robot-2 (Start-2)	187.06	196.03	4.79
	Robot-3 (Start-3)	174.79	182.25	4.27
	Robot-4 (Start-4)	152.84	160.25	4.85
2	Robot-1 (Start-1)	193.25	202.30	4.68
	Robot-2 (Start-2)	186.88	195.65	4.69
	Robot-3 (Start-3)	172.89	181.88	5.20
	Robot-4 (Start-4)	155.85	162.85	4.49
3	Robot-1 (Start-1)	192.16	201.32	4.77
	Robot-2 (Start-2)	188.07	197.59	5.06
	Robot-3 (Start-3)	173.85	180.95	4.08
	Robot-4 (Start-4)	154.97	163.09	5.24
4	Robot-1 (Start-1)	194.10	203.01	4.59
	Robot-2 (Start-2)	188.16	197.02	4.71
	Robot-3 (Start-3)	174.59	183.69	5.21
	Robot-4 (Start-4)	156.53	164.68	5.20
5	Robot-1 (Start-1)	193.67	202.33	4.47
	Robot-2 (Start-2)	186.29	194.88	4.61
	Robot-3 (Start-3)	174.12	182.32	4.71
	Robot-4 (Start-4)	156.18	163.25	4.53

6	Robot-1 (Start-1)	194.95	204.15	4.72
	Robot-2 (Start-2)	187.54	196.58	4.82
	Robot-3 (Start-3)	175.59	183.59	4.56
	Robot-4 (Start-4)	157.22	165.33	5.16
7	Robot-1 (Start-1)	194.87	204.55	4.97
	Robot-2 (Start-2)	187.08	196.22	4.88
	Robot-3 (Start-3)	174.01	182.82	5.06
	Robot-4 (Start-4)	157.15	165.11	5.07
8	Robot-1 (Start-1)	191.07	199.44	4.38
	Robot-2 (Start-2)	187.70	197.05	4.98
	Robot-3 (Start-3)	172.44	179.98	4.37
	Robot-4 (Start-4)	154.09	161.55	4.84
9	Robot-1 (Start-1)	193.56	202.65	4.70
	Robot-2 (Start-2)	187.89	196.99	4.84
	Robot-3 (Start-3)	172.49	180.22	4.48
	Robot-4 (Start-4)	156.07	163.80	4.94
10	Robot-1 (Start-1)	190.14	198.69	4.50
	Robot-2 (Start-2)	188.03	197.04	4.79
	Robot-3 (Start-3)	172.90	181.96	5.24
	Robot-4 (Start-4)	153.34	160.59	4.73
Average path length covered	Robot-1 (Start-1)	**192.73**	**201.76**	**4.68**
	Robot-2 (Start-2)	**187.47**	**196.51**	**4.82**
	Robot-3 (Start-3)	**173.77**	**181.97**	**4.72**
	Robot-4 (Start-4)	**155.43**	**163.05**	**4.90**

Table-8.10 Time taken by the robots during simulation and experimental analysis by the proposed hybrid navigation system shown in Figures 8.9 and 8.15 respectively.

No. of Runs	Robot No.	Time taken by the robots during simulation analysis (in 'cm')	Time taken by the robots during experimental analysis (in 'cm')	% of error
1	Robot-1 (Start-1)	20.39	21.48	5.35
	Robot-2 (Start-2)	25.02	26.34	5.27
	Robot-3 (Start-3)	22.18	23.29	5.01
	Robot-4 (Start-4)	22.94	24.11	5.11
2	Robot-1 (Start-1)	22.12	23.25	5.09
	Robot-2 (Start-2)	26.88	28.25	5.08
	Robot-3 (Start-3)	22.60	23.91	5.79
	Robot-4 (Start-4)	24.89	26.15	5.07
3	Robot-1 (Start-1)	21.12	22.21	5.18
	Robot-2 (Start-2)	26.59	27.99	5.25
	Robot-3 (Start-3)	23.50	24.57	4.54
	Robot-4 (Start-4)	23.76	24.96	5.07
4	Robot-1 (Start-1)	22.15	23.14	4.48
	Robot-2 (Start-2)	26.83	28.10	4.74
	Robot-3 (Start-3)	22.48	23.51	4.56
	Robot-4 (Start-4)	24.92	26.13	4.87
5	Robot-1 (Start-1)	21.65	22.68	4.75
	Robot-2 (Start-2)	26.50	27.76	4.75
	Robot-3 (Start-3)	23.27	24.40	4.87
	Robot-4 (Start-4)	24.36	25.51	4.72

6	Robot-1 (Start-1)	22.92	24.18	5.50
	Robot-2 (Start-2)	26.08	27.29	4.65
	Robot-3 (Start-3)	23.82	25.17	5.65
	Robot-4 (Start-4)	25.78	27.11	5.14
7	Robot-1 (Start-1)	21.57	22.79	5.64
	Robot-2 (Start-2)	24.60	25.68	4.40
	Robot-3 (Start-3)	23.34	24.49	4.92
	Robot-4 (Start-4)	24.27	25.45	4.86
8	Robot-1 (Start-1)	21.27	22.41	5.37
	Robot-2 (Start-2)	25.69	27.05	5.30
	Robot-3 (Start-3)	23.14	24.40	5.46
	Robot-4 (Start-4)	23.93	25.12	4.99
9	Robot-1 (Start-1)	20.52	21.51	4.85
	Robot-2 (Start-2)	26.44	27.85	5.34
	Robot-3 (Start-3)	22.70	23.98	5.63
	Robot-4 (Start-4)	23.08	24.30	5.28
10	Robot-1 (Start-1)	20.87	21.85	4.71
	Robot-2 (Start-2)	25.58	26.78	4.69
	Robot-3 (Start-3)	22.42	23.60	5.26
	Robot-4 (Start-4)	23.48	24.58	4.70
Average path length covered	Robot-1 (Start-1)	**21.46**	**22.55**	**5.09**
	Robot-2 (Start-2)	**26.02**	**27.31**	**4.95**
	Robot-3 (Start-3)	**22.95**	**24.13**	**5.17**
	Robot-4 (Start-4)	**24.14**	**25.34**	**4.98**

Table-8.11 Time taken by the robots during simulation and experimental analysis by the proposed hybrid navigation system shown in Figures 8.10 and 8.16 respectively.

Run No.	**Robot No.**	**Time taken by the robots during simulation analysis (in 'cm')**	**Time taken by the robots during experimental analysis (in 'cm')**	**% of error**
1	Robot-1 (Start-1)	21.24	22.28	4.90
	Robot-2 (Start-2)	24.96	26.23	5.08
	Robot-3 (Start-3)	20.15	21.06	4.56
	Robot-4 (Start-4)	18.69	19.71	5.46
2	Robot-1 (Start-1)	23.05	24.16	4.85
	Robot-2 (Start-2)	26.61	28.01	5.28
	Robot-3 (Start-3)	20.47	21.52	5.11
	Robot-4 (Start-4)	20.25	21.25	4.94
3	Robot-1 (Start-1)	22.00	23.11	5.07
	Robot-2 (Start-2)	26.33	27.66	5.05
	Robot-3 (Start-3)	21.11	22.22	5.22
	Robot-4 (Start-4)	19.36	20.37	5.26
4	Robot-1 (Start-1)	23.07	24.19	4.86
	Robot-2 (Start-2)	26.62	27.98	5.10
	Robot-3 (Start-3)	20.35	21.26	4.48
	Robot-4 (Start-4)	20.30	21.29	4.87
5	Robot-1 (Start-1)	22.55	23.62	4.72
	Robot-2 (Start-2)	26.30	27.67	5.20
	Robot-3 (Start-3)	21.07	22.07	4.73
	Robot-4 (Start-4)	19.85	20.75	4.54

6	Robot-1 (Start-1)	23.87	25.08	5.05
	Robot-2 (Start-2)	25.92	27.17	4.82
	Robot-3 (Start-3)	21.63	22.76	5.23
	Robot-4 (Start-4)	20.74	21.71	4.70
7	Robot-1 (Start-1)	22.47	23.58	4.93
	Robot-2 (Start-2)	24.35	25.57	5.02
	Robot-3 (Start-3)	21.19	22.15	4.50
	Robot-4 (Start-4)	19.78	20.88	5.58
8	Robot-1 (Start-1)	22.15	23.19	4.68
	Robot-2 (Start-2)	25.42	26.55	4.43
	Robot-3 (Start-3)	21.20	22.07	4.11
	Robot-4 (Start-4)	19.49	20.36	4.43
9	Robot-1 (Start-1)	21.37	22.49	5.24
	Robot-2 (Start-2)	26.11	27.45	5.15
	Robot-3 (Start-3)	20.56	21.62	5.18
	Robot-4 (Start-4)	18.81	19.76	5.05
10	Robot-1 (Start-1)	21.74	22.80	4.89
	Robot-2 (Start-2)	25.26	26.54	5.07
	Robot-3 (Start-3)	20.45	21.34	4.34
	Robot-4 (Start-4)	19.13	20.05	4.82
Average path length covered	Robot-1 (Start-1)	**22.35**	**23.45**	**4.92**
	Robot-2 (Start-2)	**25.79**	**27.08**	**5.02**
	Robot-3 (Start-3)	**20.82**	**21.81**	**4.75**
	Robot-4 (Start-4)	**19.64**	**20.61**	**4.97**

8.5 Comparison of the Developed IWO-ANFIS Hybrid Controller with other Navigational Controllers

Comparisons of the developed IWO-ANFIS path planner with other intelligent path planners have been carried out in simulation mode. The developed IWO-ANFIS path planner has been applied to similar environments as stated by the researchers. Figures 8.17 and 8.18 depicted the path traced for the robot using developed hybrid technique and other intelligent techniques respectively. The size dimension of simulation platforms are considered as no. of units and each unit is in millimeter (mm). It has been found that the proposed path planner is more efficient compared to other discussed path planners.

The first comparative study has been carried out with Mo et al. [87] for single robot path planning. In this study both the path planners have implemented to the similar environments. The developed IWO-ANFIS hybrid technique is found to be more effective compared to the technique applied by Mo et al. [87].

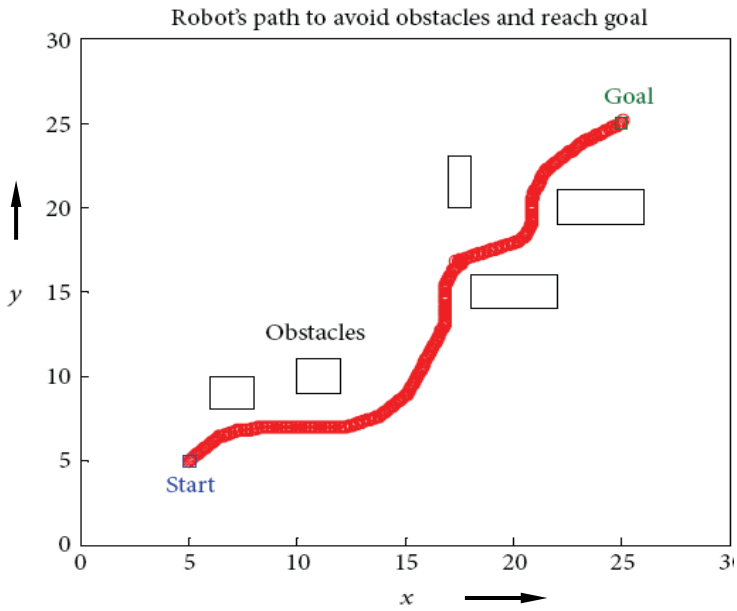


Figure 8.17 (a) Navigation path for mobile robot to reach target using Mo et al. [87] developed technique.

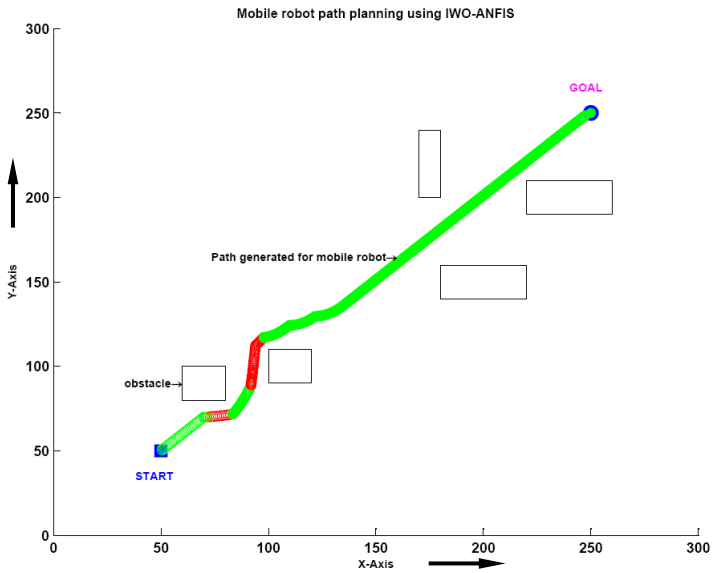


Figure 8.17 (b) Navigation path for mobile robot to reach target using current developed hybrid technique.

Similarly the second comparative study has been carried out with He et al. [115] for single robot navigation. From the graphical analysis, it has been clearly observed that the developed hybrid technique performs more efficiently compared to the technique implemented by the He et al. [115].

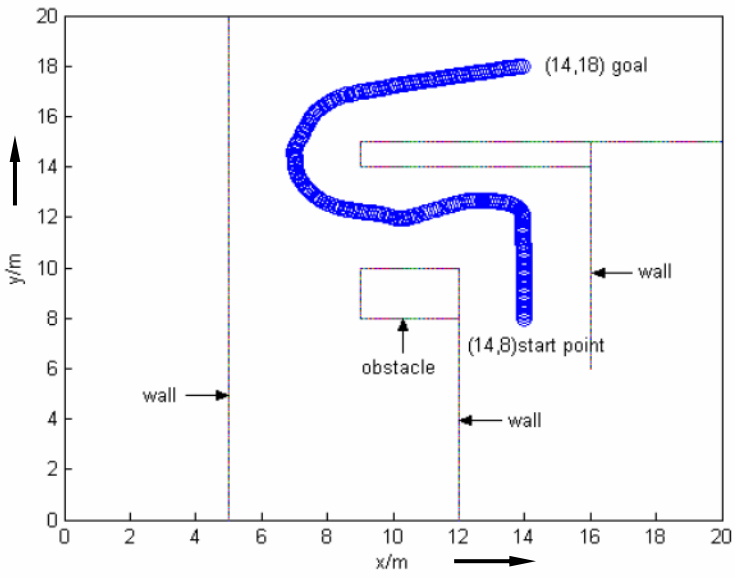


Figure 8.18 (a) Navigation path for mobile robot to reach target using He et al. [115] developed technique.

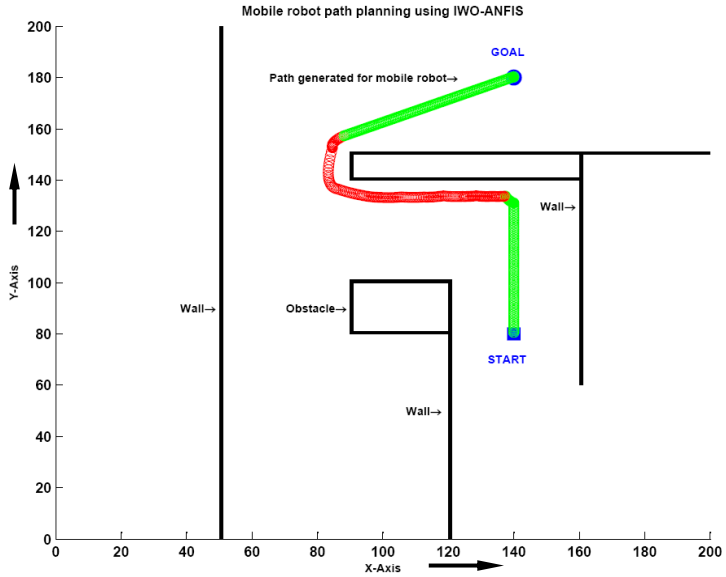


Figure 8.18 (b) Navigation path for mobile robot to reach target using current developed hybrid technique.

The comparison performances among different methods have been recorded in terms of path length (in 'cm') and presented in the Table-8.12. From the above simulation analysis, it has been clearly seen that the developed hybrid path planner can efficiently handle the robot navigation in any complex environments.

Table-8.12 Comparison of simulation results between the current investigation and other intelligent techniques.

SL. No.	**Environment types**	**Path length from the current system ('in 'cm')**	**Path length from the reference model ('in 'cm')**	**% of deviation**
1	Maze environment with narrow passages, Figures 8.17(a) and 8.17(b)	7.4	8.7	14.94
2	Long barrier is presented between the robot and target, Figures 8.18(a) and 8.18(b)	7.1	8.1	12.34

8.6 Summary

In this section, the following conclusions are obtained from the IWO based ANFIS navigational controller.

- A novel approach for controlling the navigation of multiple mobile robots, using the IWO based adaptive neuro-fuzzy inference system (ANFIS) has been effectively applied in highly cluttered environments.
- The IWO algorithm has been deployed to optimize the membership function parameters and least square estimation (LSE) method has been used to identify the conclusion parameters in ANFIS, and to calculate the suitable steering angle of the mobile robot.
- The developed navigational methodology uses sensor extracted information, in order to determine the change required in the steering angle for obstacle avoidance.
- In order to avoid collision problems between the mobile robots, collision prevention rules have been designed and effectively loaded in the robot controller, using the Petri-Net model.
- The real-time experimental tests using Khepera-II and Khepera-III mobile robots, have been validated with the simulation results, showing that the IWO based ANFIS controller performs better to navigate the mobile robot safely in a terrain populated by a variety of obstacles.
- The maximum percentage of deviation between experimental and simulation results for single and multiple robots in terms of path length are 4.93% and 4.97% respectively. Similarly, the maximum percentage of deviation between experimental and simulation results for single and multiple robots in terms of time taken are 5.10% and 5.17% respectively.
- The current navigational technique has been compared with other intelligent motion controllers, and it has been observed that the developed methodology yields better results than the other techniques. The percentage of deviation has been shown in the Table-8.12.

CHAPTER-9

9. RESULTS AND DISCUSSION

9.1 Introduction

The feasibility and effectiveness of the several navigational methods for mobile robots have been have been carried out in the current chapter of the dissertation The performance of the each developed intelligent controller has been presented systematically through both simulation and experimental validation. Different navigational techniques have been implemented in the current research work for the path planning of mobile robots in various environments and they are Adaptive Neuro-Fuzzy Inference System (ANFIS) and Multiple Adaptive Neuro-Fuzzy Inference System (MANFIS), Cuckoo Search (CS) algorithm and Cuckoo Search-Adaptive Neuro-Fuzzy Inference System hybrid algorithm (CS-ANFIS), and Invasive Weed optimization (IWO) and Invasive Weed optimization-Adaptive Neuro-Fuzzy Inference System (ANFIS) hybrid algorithm (IWO- ANFIS).

9.2 Analysis of Simulation Results

Two different simulation exercises have been carried out to compare the performances of the developed navigational controllers for single mobile robot. The obstacle avoidance, target seeking, barrier following and collision free path depend on the intelligence of the path planner. In the first simulation exercise, the robot is presented inside three long rectangular obstacles, and a target is situated in the environment. Figure 9.1(a) shows the initial condition of the scenario for all the navigational techniques. Figures 9.1(b) to 9.1(g) represent the simulation results obtained for the robot using ANFIS, MANFIS, CS, CS-ANFIS, IWO, and IWO-ANFIS respectively. For each navigational controller, the average of 20 trials has been taken into consideration to show the best simulation paths. The total path lengths covered by the robot using proposed navigation techniques are measured in ‘cm’ for the single robot. Similarly, time taken for the robot to reach the goal using proposed navigation techniques is recorded in ‘sec’ for the single robot. The path length covered by the robot and time taken to reach the target are considered as an objective to compare the performance of different navigational methods. It has been

observed that CS-ANFIS and IWO-ANFIS hybrid algorithms perform the best among the other discussed navigational methods. In the second simulation exercise shown in Figure 9.2, the robot is navigating in a cluttered environment using the proposed navigational methods. It has been noted that the CS-ANFIS and IWO-ANFIS hybrid algorithms perform better compared to the other discussed navigational techniques. Similarly, a comparative study has been performed for multiple mobile robots using CS-ANFIS and IWO-ANFIS hybrid navigational strategies. The simulation results are presented in Figure 9.3(a) and Figure 9.3(b) respectively. It can be found that the CS-ANFIS and IWO-ANFIS hybrid algorithms perform better and successfully used for solving navigational problems of multiple mobile robots. The CS-ANFIS hybrid algorithm also produces a smooth trajectory for the robots compared to the IWO-ANFIS hybrid algorithm. In simulation graph the length of X and Y axes are taken as 30 units each and one unit is equal to 2mm.

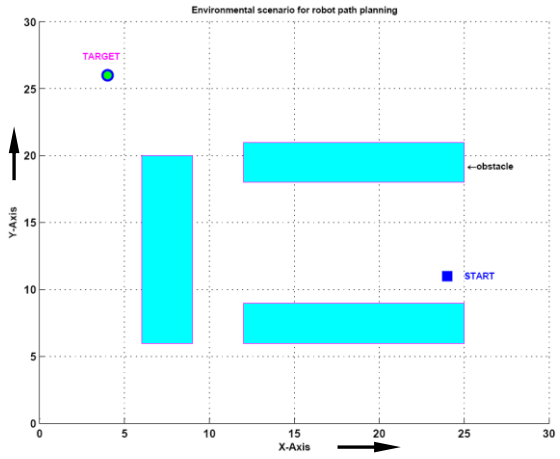


Figure 9.1 (a)

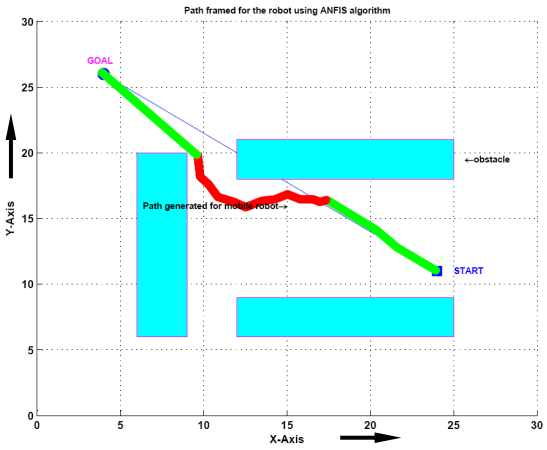


Figure 9.1 (b)

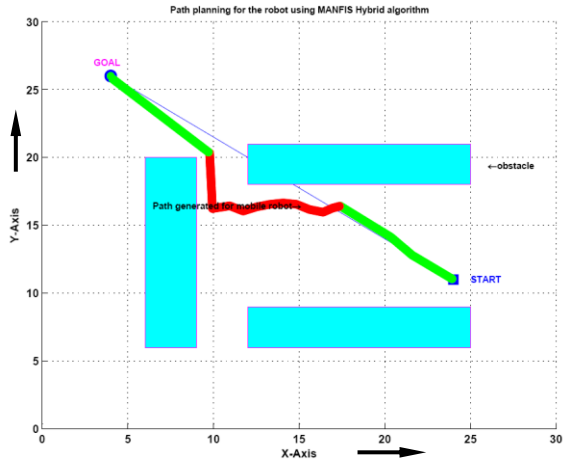


Figure 9.1 (c)

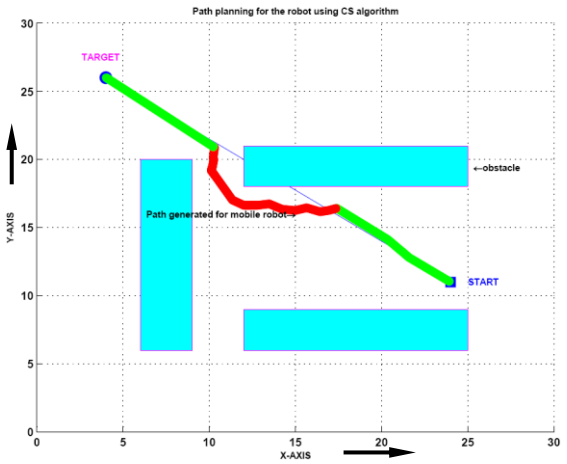


Figure 9.1 (d)

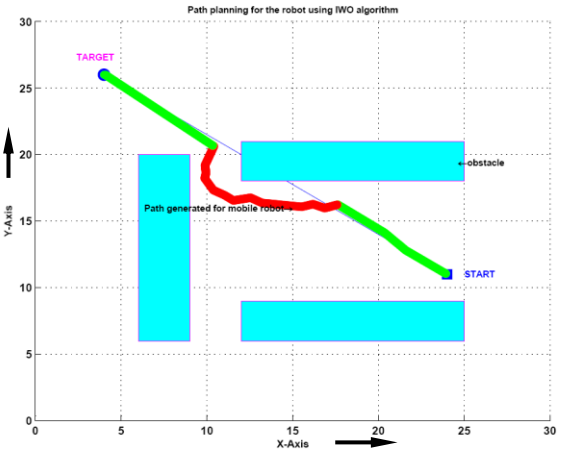


Figure 9.1 (e)

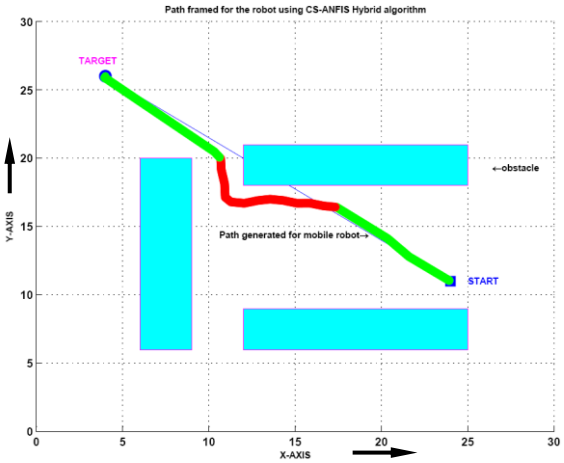


Figure 9.1 (f)

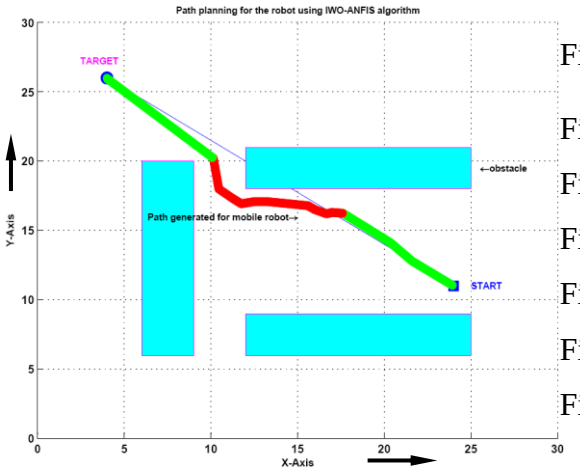


Figure 9.1 (g)

Figure 9.1 (a) Environmental scenario for navigation.

Figure 9.1 (b) Navigation using ANFIS

Figure 9.1 (c) Navigation using MANFIS

Figure 9.1 (d) Navigation using CS

Figure 9.1 (e) Navigation using IWO

Figure 9.1 (f) Navigation using CS-ANFIS

Figure 9.1 (g) Navigation using IWO-ANFIS

Figure 9.1 Comparison of path of single robot during simulation using different techniques.

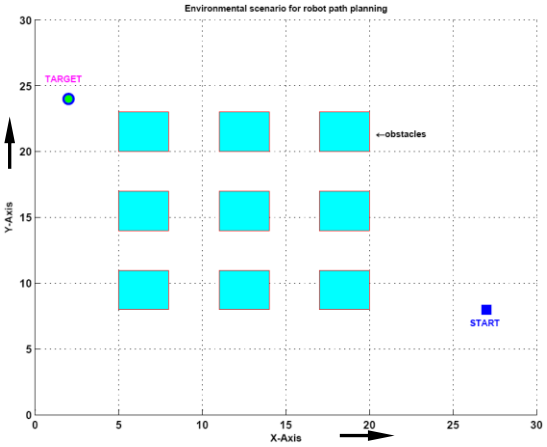


Figure 9.2 (a)

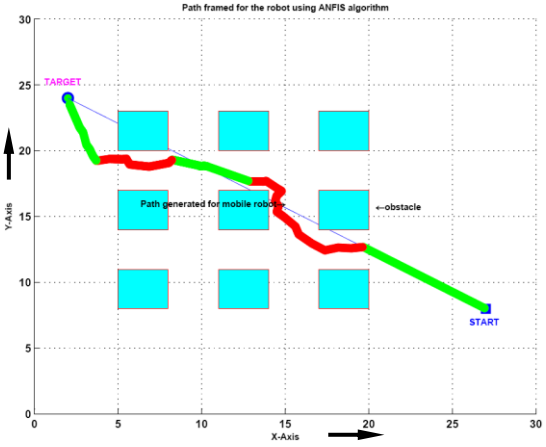


Figure 9.2 (b)

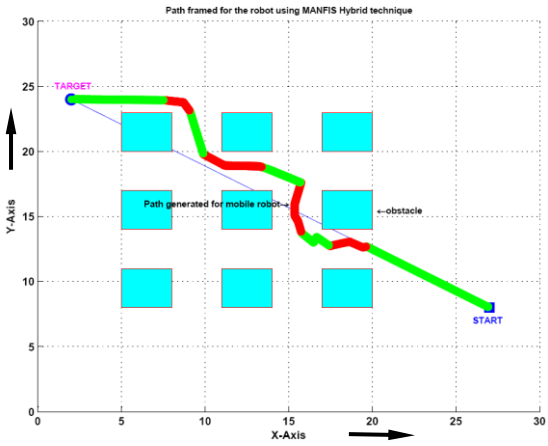


Figure 9.2 (c)

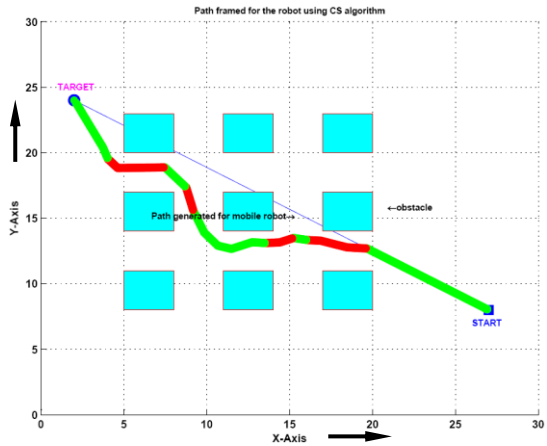


Figure 9.2 (d)

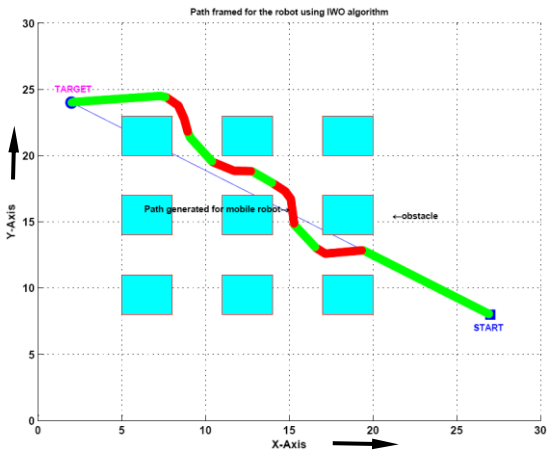


Figure 9.2 (e)

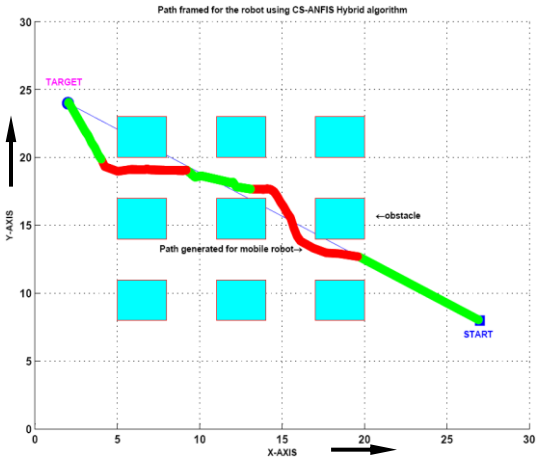


Figure 9.2 (f)

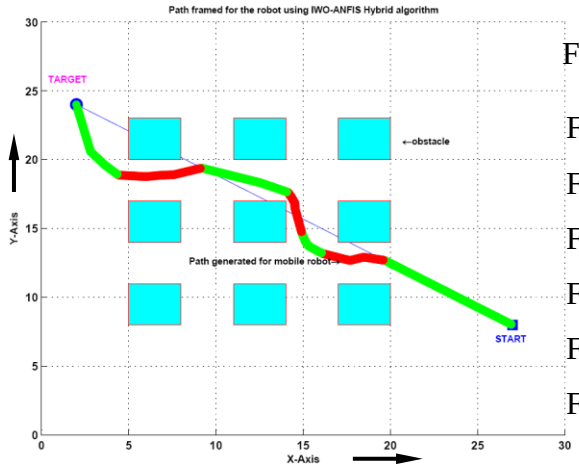


Figure 9.2 (a) Environmental scenario for navigation

Figure 9.2 (b) Navigation using ANFIS

Figure 9.2 (c) Navigation using MANFIS

Figure 9.2 (d) Navigation using CS

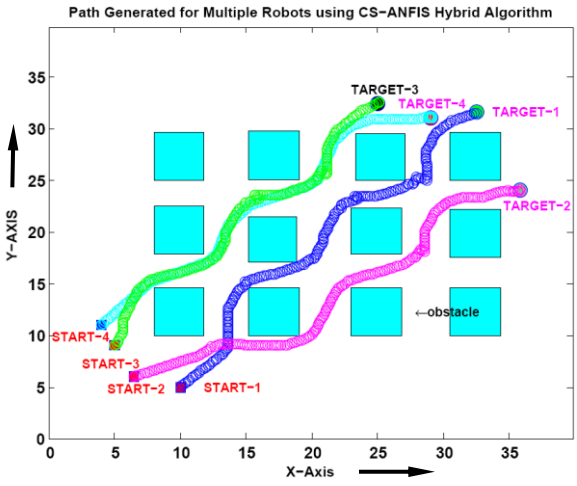
Figure 9.2 (e) Navigation using IWO

Figure 9.2 (f) Navigation using CS-ANFIS

Figure9.2 (g) Navigation using IWO-ANFIS

Figure 9.2 (g)

Figure 9.2 Comparison of path of single robot during simulation using different techniques.



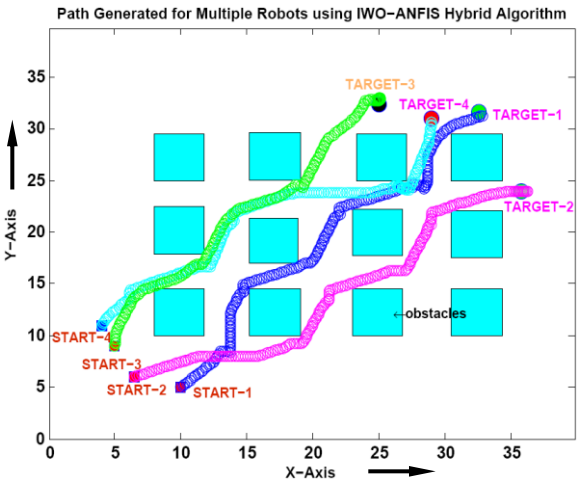


Figure 9.3 (a)

Figure 9.3 (b)

Figure 9.3 Comparison of path of multiple robots during simulation using CS-ANFIS and IWO-ANFIS techniques.

9.3 Experimental Validation with Simulation Results

The experimental analysis has been conducted in our laboratory platform (225cmx175cm) using a real robot (shown in Appendix-I) to demonstrate the effectiveness of the developed navigational methods. The experiments have been performed to validate simulation results obtained from the different path planners. The details of experimental procedures have been discussed in the previous chapter. Figures 9.4-9.7 present the experimental results obtained for ANFIS, MANFIS, CS, IWO, CS-ANFIS, and IWO-ANFIS techniques for a single robot similar to the environment as shown in Figures 9.1 and 9.2 respectively. It has been noted that the path obtained experimentally follows closely those traced by the robot during simulation results. From the experimental figures, it can be clearly seen that the robots can successfully avoid obstacles and reach the goal. The proposed hybrid navigational strategies (CS-ANFIS and IWO-ANFIS) have been implemented to solve the multiple mobile robots tasks shown in Figures 9.8-9.10. It has been seen that the robots closely follow the simulation path using the developed motion planning approach and able to rove in a highly cluttered environment. The final comparison between simulation and experimental results in terms of path lengths and time taken using different navigational strategies have been depicted in Table-9.1-9.3. It has been found that the CS-ANFIS and IWO-ANFIS hybrid motion planning approaches perform best among the other discussed navigational approaches. The error between simulation and experimental results are found to be within 5% for path length and 6% for time taken by the robot to reach the target.

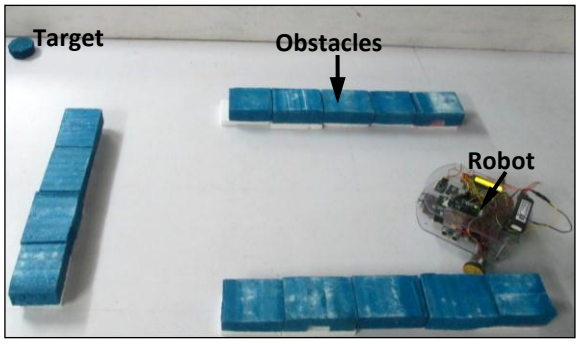


Figure 9.4 (a)

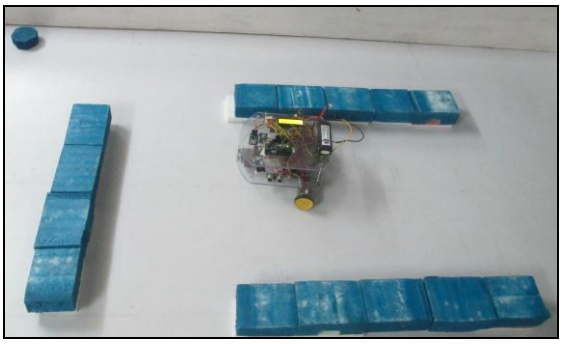


Figure 9.4 (b)

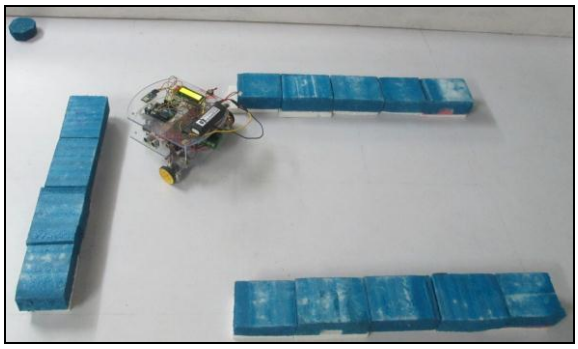


Figure 9.4 (c)

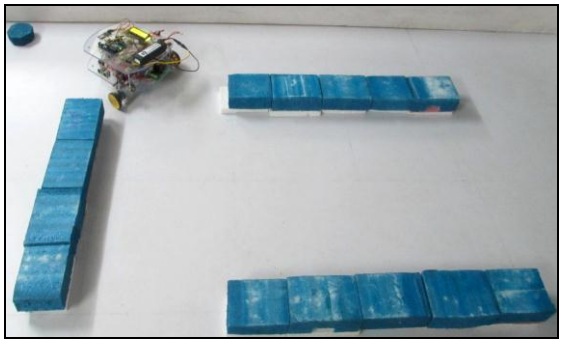


Figure 9.4 (d)

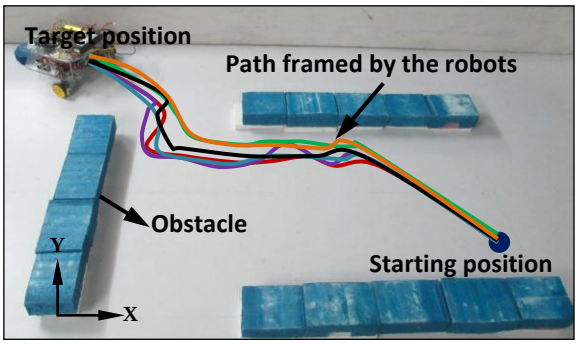


Figure 9.4 (e)

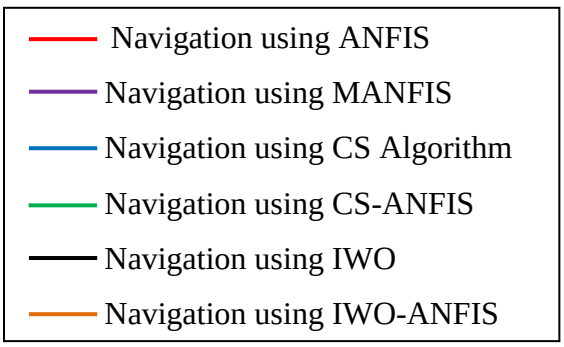


Figure 9.4 (f)

Figure 9.4 Comparison of experimental path by single robot using different techniques shown in Figure 9.1.

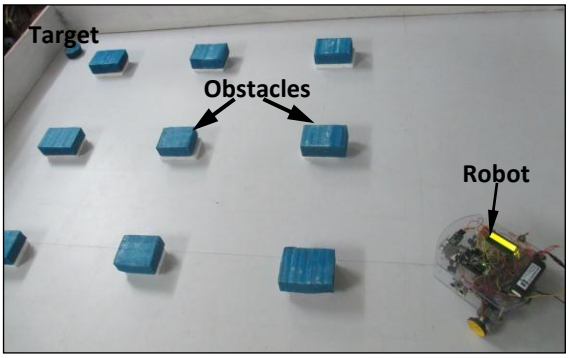


Figure 9.5(a)

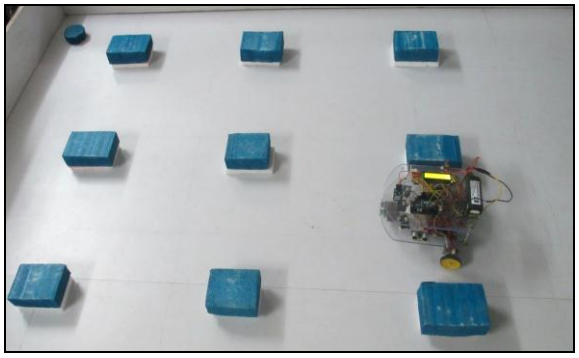


Figure 9.5(b)

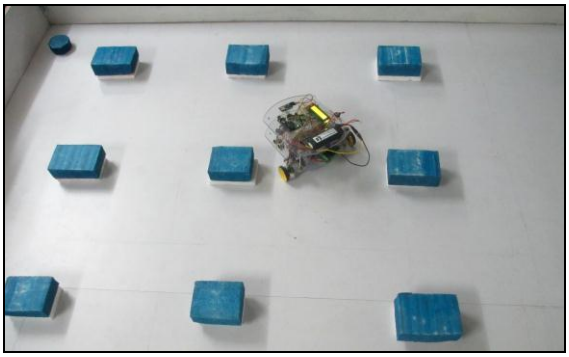


Figure 9.5(c)

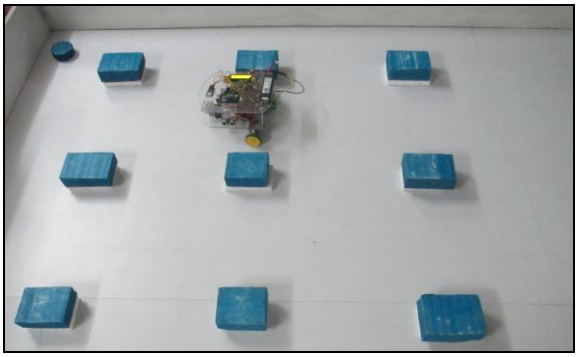


Figure 9.5(d)

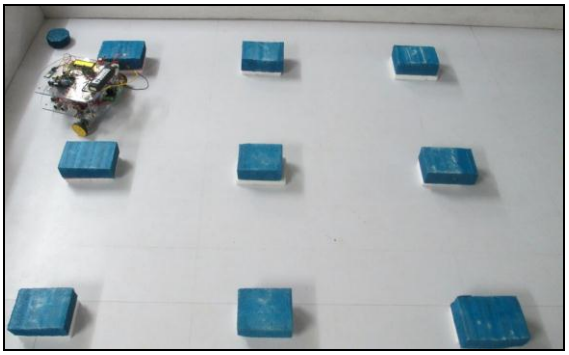


Figure 9.5(e)

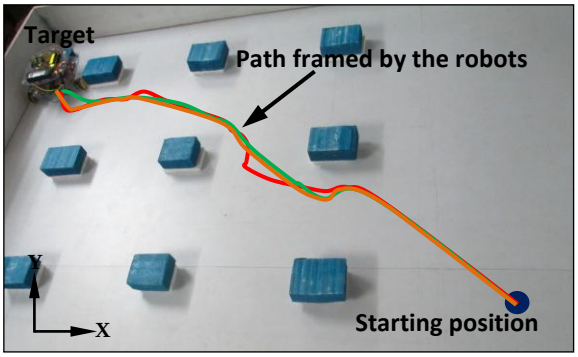


Figure 9.5(f)

Figure 9.5 Comparison of experimental path by single robot using different techniques shown in Figures 9.2(b), 9.2(f) and 9.2(g).

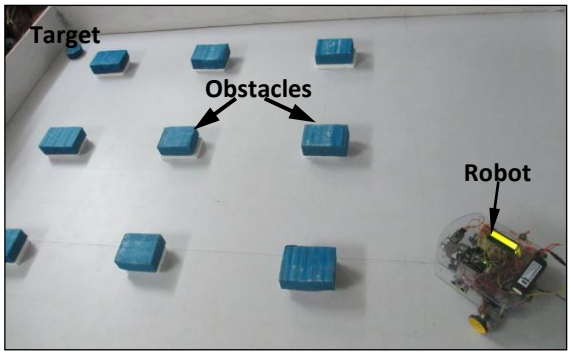


Figure 9.6(a)

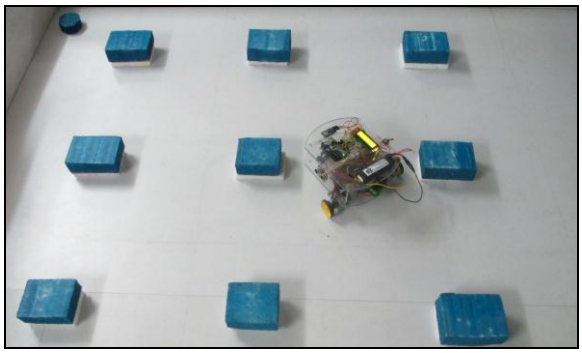


Figure 9.6(b)

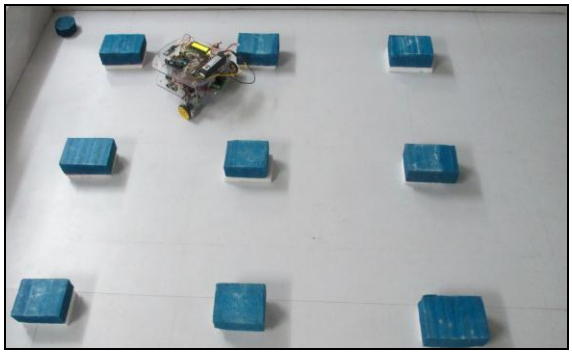


Figure 9.6(c)

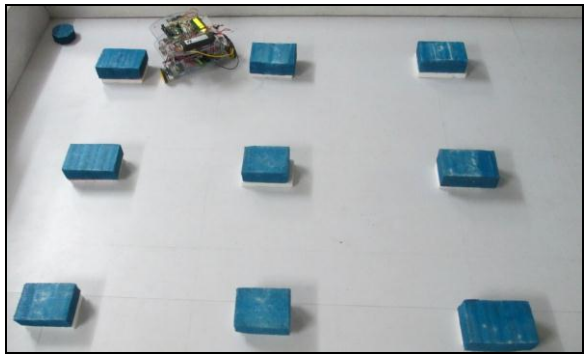
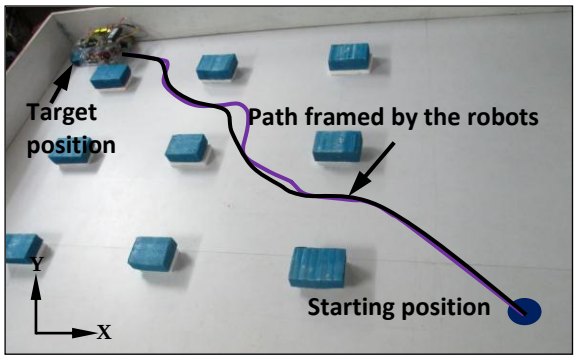


Figure 9.6(d)



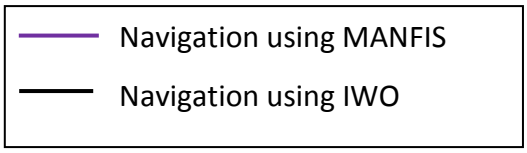


Figure 9.6(e)

Figure 9.6 Comparison of experimental path by single robot using MANFIS and IWO techniques shown in Figures 9.2 (c) and 9.2 (e).

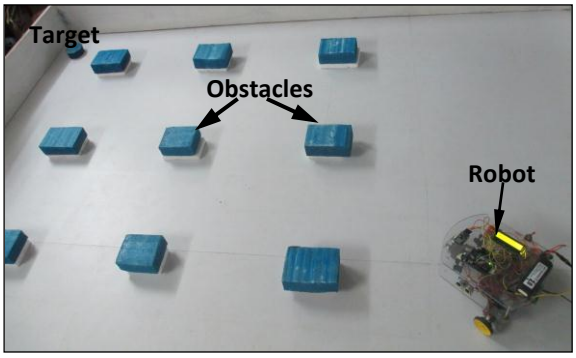


Figure 9.7(a)

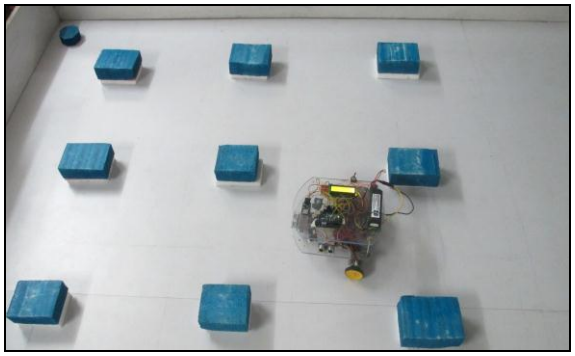


Figure 9.7(b)

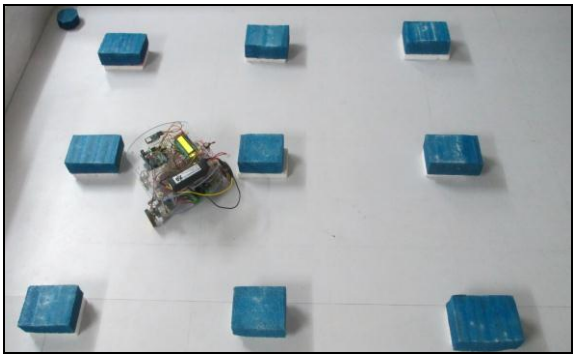


Figure 9.7(c)

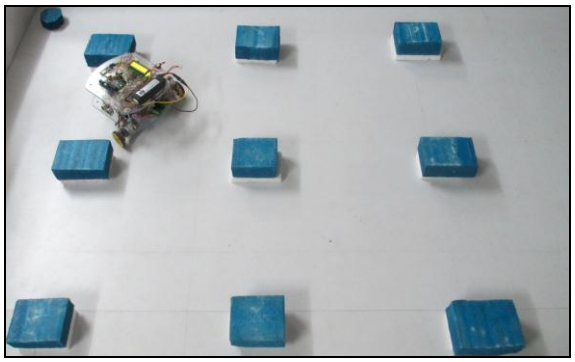


Figure 9.7(d)

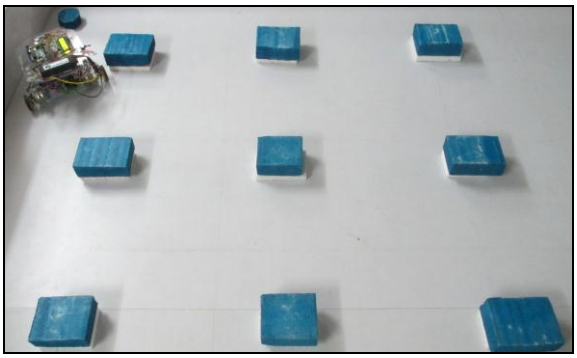


Figure 9.7(e)

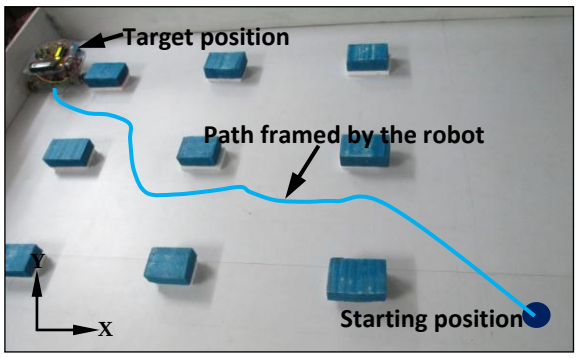


Figure 9.7(f)

Figure 9.7 Comparison of experimental path by single robot using CS algorithm shown in Figure 9.2 (d).

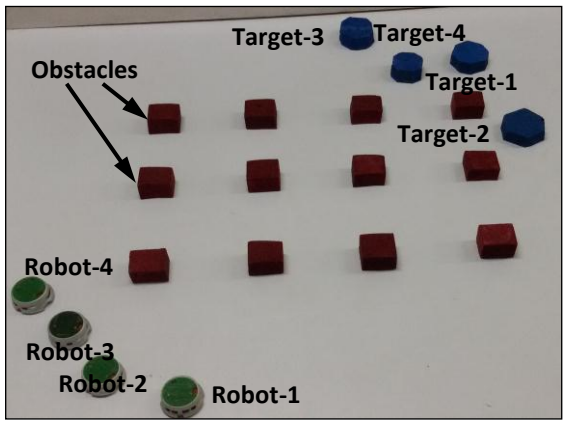


Figure 9.8 Real-time scenario for Mobile robot path planning.

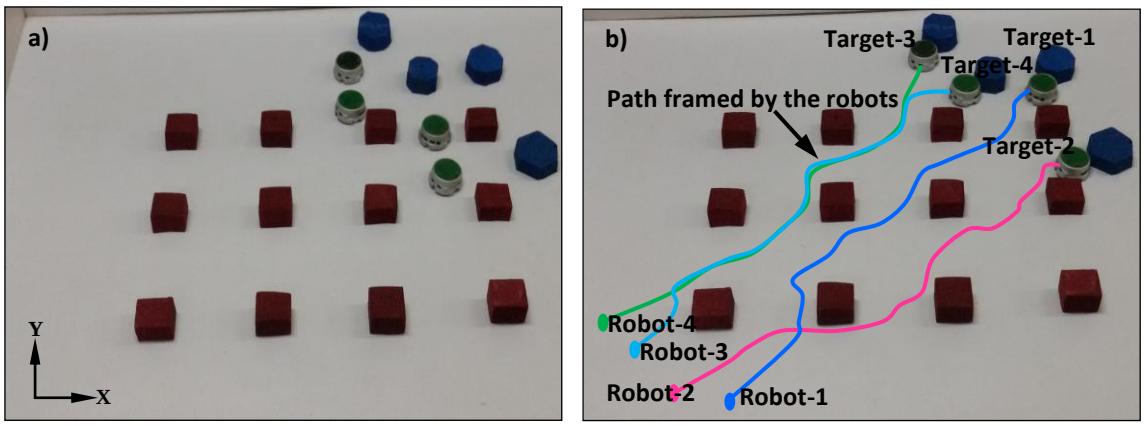


Figure 9.9 (a-b) Comparison of experimental path by multiple robots using CS-ANFIS hybrid algorithm shown in Figure 9.3 (a).

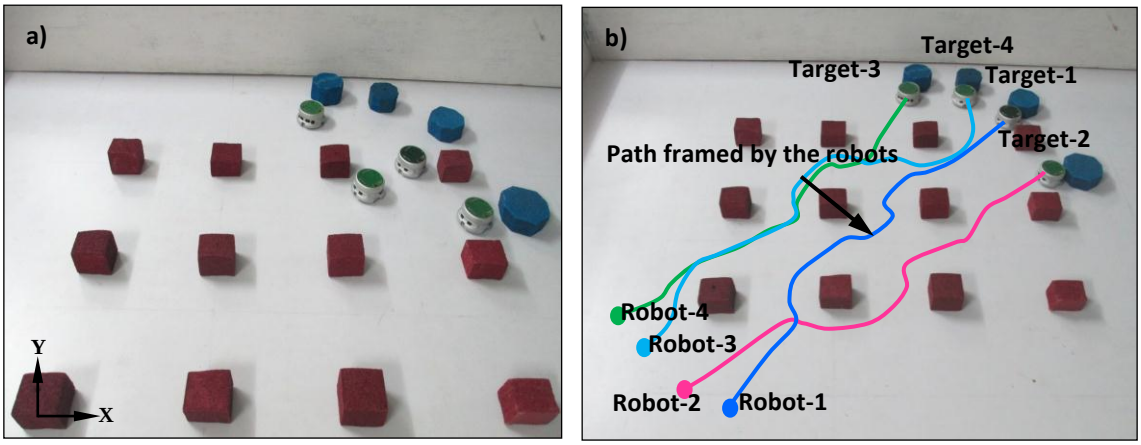


Figure 9.10 (a-b) Comparison of experimental path by multiple robots using IWO-ANFIS hybrid algorithm shown in Figure 9.3 (b).

Table 9.1 Comparison of path lengths and time taken by the robots for Scenario-1 (Figure 9.1 and Figure 9.4).

Different navigation techniques	**Average Path lengths in simulation (in 'cm')**	**Average Path lengths in experiment (in 'cm')**	**% of Average error**	**Average Time taken in simulation (in 'sec')**	**Average Time taken in experiment (in 'sec')**	**% of Average error**
ANFIS	164.62	174.34	5.91	21.11	22.5	6.58
MANFIS	165.01	175.00	6.05	21.72	23.22	6.91
CS	163.14	172.41	5.68	20.6	22.02	6.89
IWO	161.73	170.82	5.62	20.55	21.92	6.67
CS-ANFIS	***159.69***	***166.71***	***4.40***	***20.11***	***21.22***	***5.52***
IWO-ANFIS	***157.89***	***165.37***	***4.74***	***19.98***	***21.06***	***5.41***

Bold color shows the best results compared to other discussed techniques.

Table 9.2 Comparison of path lengths and time taken by the robots for Scenario-2 (Figure 9.2 and Figures 9.5-9.7).

Different navigation techniques	**Average Path lengths in simulation (in 'cm')**	**Average Path lengths in experiment (in 'cm')**	**% of Average error**	**Average Time taken in simulation (in 'sec')**	**Average Time taken in experiment (in 'sec')**	**% of Average error**
ANFIS	179.80	190.89	6.17	22.12	23.64	6.87
MANFIS	181.52	192.67	6.14	22.98	24.46	6.44
CS	181.26	191.42	5.61	22.46	24.02	6.95
IWO	179.58	190.32	5.98	22.38	23.9	6.79
CS-ANFIS	***176.33***	***185.09***	***4.97***	***21.18***	***22.42***	***5.85***
IWO-ANFIS	***179.12***	***187.55***	***4.71***	***20.94***	***22.18***	***5.92***

Bold color shows the best results compared to other discussed techniques.

Table 9.3 Comparison of path lengths and time taken by the robots for Scenario-3 (Figure 9.3, 9.9(b) and 9.10(b)).

Different hybrid navigation techniques		**Average Path lengths in simulation (in 'cm')**	**Average Path lengths in experiment (in 'cm')**	**% of Average error**	**Average Time taken in simulation (in 'sec')**	**Average Time taken in experiment (in 'sec')**	**% of Average error**
CS-ANFIS	Robot-1	173.37	181.03	4.42	23.13	24.13	4.77
	Robot-2	167.26	174.36	4.25	21.21	22.09	4.17
	Robot-3	165.27	173.17	4.78	18.11	19.02	4.99
	Robot-4	153.87	160.61	4.38	17.56	18.32	4.33
IWO-ANFIS	Robot-1	172.96	180.14	4.15	22.87	23.89	4.45
	Robot-2	169.33	176.74	4.38	21.57	22.56	4.59
	Robot-3	165.45	172.96	4.53	19.69	20.66	4.93
	Robot-4	157.91	165.33	4.70	18.33	19.21	4.80

9.4 Summary

In the present chapter, six types of navigational controllers such as ANFIS, MANFIS, CS, IWO, CS-ANFIS, and IWO-ANFIS have been discussed. To show the efficacy of the navigational algorithms for single mobile robot, two different types of scenarios (shown in Figures 9.1 and 9.2) have been taken into consideration. In each case, the average of 20 runs has been taken into account. The best simulation graph has been presented in the results. The experimental verifications also have been done for the different hybrid navigational algorithms to corroborate with their respective developed working model in reality. They are found to be in good agreement. The path length covered and time taken by the single robot to reach the target using different techniques have been depicted in the Table-9.1 and Table-9.2 respectively. During the analysis of the results, it has been noticed that the hybrid navigational controllers such as CS-ANFIS and IWO-ANFIS performs better than the other discussed algorithms. The average percentage of errors for path length and time taken are found to be within 5% and 6% respectively. By investigating the results obtained from the proposed CS-ANFIS and IWO-ANFIS methodologies, it has been observed that the developed hybrid navigational controllers can be effectively implemented to solve the path planning optimization problems of mobile robots.

The best hybrid navigational path planners (CS-ANFIS and IWO-ANFIS) have been implemented for multiple mobile robots navigation. Similarly the path length and time taken by the robots to reach the targets are depicted in Table-9.3. During experimentation, it has been found that the robot closely follow the simulation path while avoiding obstacles. By analyzing the results, it has been observed that the CS-ANFIS and IWO-ANFIS hybrid algorithms provide smoothness and best path trajectory compared to the other developed intelligent navigational controllers. It has been noticed that the average percentage of errors are found to be within 5% for both path length and time taken by the robots to reach targets.

Finally, the CS-ANFIS and IWO-ANFIS hybrid path planners are found to be suitable for solving the navigational problem of multiple mobile robots.

CHAPTER-10

10. CONCLUSION AND SUGGESTIONS FOR FUTURE RESEARCH

The work presented in this emphasizes in the analysis and development of effective and efficient intelligent navigational controllers for multiple mobile robots navigation in partially or completely unknown environments. In this chapter, the results of intelligent navigational controllers discussed in the previous chapters are summarized. The main contribution to the research work, and scope for future research directions based on the work carried out in this dissertation are also presented.

10.1 Important Contributions

The following are important contributions of the dissertation:

- The description of the kinematic posture of a wheeled mobile robot has been discussed, and it is important to calculate the wheel velocities of the mobile robot. From the wheel velocities, the steering angle of the robot is determined. The kinematic description of the mobile robot is sufficient to explain the global motion of the robot in the target seeking environment.
- A novel straight forward Adaptive Neuro-Fuzzy Inference System (ANFIS) with three bell shaped membership functions have been considered for mobile robot navigation. The developed navigational strategy has been taken the advantages of both fuzzy logic and neural network. A Multiple Adaptive Neuro-Fuzzy Inference System (MANFIS) based navigational controller has been developed to control the motion of the robot in unknown or partially known environments. It has been observed that using the above navigational techniques the robots are able to avoid obstacles and find the goals effectively.
- The optimization algorithm Cuckoo Search (CS) has been successfully implemented for the mobile robot navigation. This optimization algorithm has been modified to produce a hybrid CS-ANFIS technique for enhancing the navigational performance of the mobile robots. Using this hybrid algorithm the mobile robots has been successfully avoiding the inter-collisions problem. A new Petri-Net model has been

developed and embedded in the CS-ANFIS hybrid controller for multiple mobile robots navigation.

- The Invasive Weed Optimization (IWO) algorithm has been applied for mobile robot path planning. The developed optimization algorithm has been hybridized with ANFIS for enhancing the navigational performance of the robots. In order to avoid collision problems between the mobile robots, collision prevention rules have been designed and effectively loaded in the robot controller, using the Petri-Net theory.
- Experimental set-ups have been developed to validate the simulation results using above discussed navigational strategies.

10.2 Conclusions

The conclusions, drawn on the basis of simulation and experimental results obtained from various navigational systems as discussed above are depicted below.

- The kinematic analysis of wheeled mobile robot has been discussed. From the analysis, it has been confirmed that wheel velocities of the robot are used to determine the suitable steering of the robot. The kinematic description of the mobile robot is sufficient to explain the global motion of the robot in the target seeking environment.
- In chapter four, an effective adaptive neuro-fuzzy system has been developed for mobile robot navigation in the presence of static obstacles. From the simulation and experimental results, it has been observed that the developed hybrid navigational controller can effectively control the mobile robot in unknown or partially known environments. Similarly, the developed MANFIS path planner provides closer results to the ANFIS controller.
- The implementation of the Cuckoo Search (CS) algorithm for mobile robot navigation has been discussed in chapter five. This algorithm has increased the navigational performance of the robots compared to ANFIS technique and also produced smoothness in the robot trajectory. Similarly, in chapter seven CS-ANFIS hybrid technique has been successfully implemented for robot path planning, which provides better results compared to both standalone CS and ANFIS techniques. It has been also concluded that the CS-ANFIS hybrid technique can be successfully applied for multiple mobile robots navigation in unknown cluttered environments.

- In chapter six, a weed colonization based Invasive Weed Optimization (IWO) algorithm has been introduced to generate the optimal trajectory for the mobile robot. The path analysis result shows that IWO based navigational controller gives better results compared to ANFIS controller. Similarly, the IWO-ANFIS hybrid algorithm has been used for multiple mobile robots navigation. It has been observed that the results obtained using IWO-ANFIS are closer to the CS-ANFIS hybrid technique.
- The best performing methods are based on CS-ANFIS and IWO-ANFIS techniques, which provide robust navigation results in unknown or partially known environments. These two hybrid (CS-ANFIS and IWO-ANFIS) techniques also give better results than the other discussed techniques.

10.3 Suggestions for Future Research

After the review of the research work in this dissertation, this part lists several suggestions recommended for future directions which are further to investigate for extending the results developed in this dissertation.

- In the current work, different navigational architectures have been developed for multiple mobile robots navigation to avoid the collision among each other and static obstacles. However, further development may be needed for the avoidance of moving obstacles (Man, moving machine, animals, etc.) other than the robots.
- Multi-robot coordination may be required for the co-operative task with static as well as dynamic obstacles.
- The navigational methods have been developed in this thesis for detecting the static targets. Further modification may be required to detect the dynamic targets and also to reach them.
- Furthermore, it may be a good idea to implement the active vision technology so that the robot can actively percept the environment conditions and avoid the robots and obstacles in an intelligent manner.
- More robust hybrid intelligent navigation systems may be developed to solve the path planning optimization problem of multiple mobile robots.

REFERENCES

1. Latombe J. C., *Robot motion planning*, 124, Springer Science and Business Media, 2012.
2. Brooks R. A., A robust layered control system for a mobile robot, *IEEE Journal of Robotics and Automation*, 2(1), 1986, 14-23.
3. Mitchell Joseph.S.B., An algorithm approach to some problems in terrain navigation, *Artificial Intelligence*, 37(1-3), 1988, 171-201.
4. Masehian E. and Sedighizadeh D., Classic and heuristic approaches in robot motion planning-a chronological review, *World Academy of Science, Engineering and Technology*, 23, 2007, 101-106.
5. Stoeter S.A. and Papanikolopoulos N., Kinematic motion model for jumping scout robots**,** *IEEE Transactions on Robotics*, 22(2), 2006, 398-403.
6. Hosoda K., Takuma T., Nakamoto A. and Hayashi S., Biped robot design powered by antagonistic pneumatic actuators for multi-modal locomotion, *Robotics and Autonomous systems*, 56(1), 2008, 46–53.
7. Wu X. and Ma S., CPG-based control of serpentine locomotion of a snake-like robot, *Mechatronics*, 20(2), 2010, 326–334.
8. Bayraktaroglu Z.Y., Snake-like locomotion: Experimentations with a biologically inspired wheel-less snake robot, *Mechanism and Machine theory*, 44(3), 2009, 591–602.
9. Guzman J.L., Berenguel M., Rodriguez F. and Dormido S., An interactive tool for mobile robot motion planning, *Robotics and Autonomous systems*, 56(5), 2008, 396–409.
10. Chakraborty N. and Ghosal A., Kinematics of wheeled mobile robots on uneven terrain, *Mechanism and Machine theory*, 39(12), 2004, 1273–1287.
11. Alexander J.C. and Maddocks J.H., On the kinematics and control of wheeled mobile robots, *SRC TR 87-196*, October 1987, 1-15.
12. Campion G., Bastin G., and AndrCa-Novel B.D., Structural properties and classification of kinematic and dynamic models of wheeled mobile robots, *IEEE Transactions on Robotics and Automation*, 12(1), 1996, 47-62.

13. Kim D.S., Kwon W.H. and Park H.S., Geometric kinematics and applications of a mobile robot, *International Journal of Control, Automation, and Systems*,1(3), 2003, 376-384.
14. Cariou C., Lenain R., Thuilot B. and Martinet P., Adaptive control of four-wheel-steering off-road mobile robots: Application to path tracking and heading control in presence of sliding, IEEE International conference *on Intelligent Robots and Systems*, France, September 22-26, 2008, 1759-1764.
15. Marcovitz F.R. and Kelly A., On-line mobile robot model identification using integrated perturbative dynamics, *Experimental Robotics*, Springer Berlin Heidelberg, January, 2014, 417-431.
16. Lee J.H., Kim B.K., Tanikawa T. and Ohba K., Kinematic analysis on omni-directional mobile robot with double-wheel-type active casters, IEEE International conference *on Control, Automation and Systems*, Seoul, Korea, October 17-20, 2007,1217-1221.
17. Chung W., Moon C., Jung C. and Jin J., Design of the dual offset active caster wheel for holonomic omni-directional mobile robots, *International Journal of Advanced Robotic Systems*, 7(4), 2010, 105-110.
18. Li Y.P., Zielinska T., Ang Jr. M.H. and Lin W., Vehicle Dynamics of Redundant Mobile Robots with Powered Caster Wheels, *Springer Vienna*, 221-228.
19. Kim S. and Lee S., Local and global isotropy analysis of mobile robots with three active caster wheels, *Advances in service robotics*, ISBN 978-953-7619-02-2, 2008,117-126.
20. Indiveri G., Swedish wheeled omnidirectional mobile robots: kinematics analysis and control, *IEEE Transactions on Robotics*, 25(1), 2009, 164-171.
21. Indiveri G., Paulus J. and Ploger P.G., Motion control of swedish wheeled mobile robots in the presence of actuator saturation, *LNCS 4434*, Springer Berlin Heidelberg, 2007, 35-46.
22. Doroftei I., Grosu V. and Spinu V., Omnidirectional mobile robot-design and implementation, *Bio-inspiration and robotics: walking and climbing robots*, ISBN 978-3-902613-15-8, In-Tech publishers, 2007, 511-528.
23. Tadakuma K. and Tadakuma R., Mechanical design of "Omni-Ball": spherical wheel for holonomic omnidirectional motion, 3rd Annual IEEE conference *on Automation Science and Engineering*, Scottsdale, USA, September 22-25, 2007, 788-794.

24. Wu C.W. and Hwang C.K., A novel spherical wheel driven by omni wheels, 7th IEEE International conference on *Machine Learning and Cybernetics*, Kunming, China, July 12-15, 2008, 3800-3803.
25. Mukherjee R., Minor M.A., and Pukrushpan J.T., Motion planning for a spherical mobile robot: revisiting the classical ball-plate problem, *Transactions of the ASME - Journal of Dynamic Systems, Measurement, and Control*, 124, 2002, 502-511.
26. Lauwers T. B., Kantor G. A., and Hollis R. L., A dynamically stable single-wheeled mobile robot with inverse mouse-ball drive, IEEE International conference *on Robotics and Automation*, Orlando, Florida, May 15-19, 2006, 2884-2889.
27. Chwa D., Tracking control of differential-drive wheeled mobile robots using a back stepping-like feedback linearization, IEEE Transactions on Systems, Man, and Cybernetics-part a: Systems and Humans, 40(6), 2010, 1285-1295.
28. Petrov P. and Dimitrov L., Nonlinear path control for a differential drive mobile robot, RECENT, 11(1), 2010, 41-45.
29. Rashid R., Elamvazuthi I., Begam M. and Arrofiq M., Fuzzy-based navigation and control of a non-holonomic mobile robot, *Journal of Computing*, 2(3), 2010, 130-137.
30. Menn F.L, Bidaud P. and Amar F.B., Generic differential kinematic modeling of articulated multi-monocycle mobile robots, IEEE International conference *on Robotics and Automation*, Orlando, Florida, May 15-19, 2006, 1505-1510.
31. Feng L., Koren Y. and Borenstein J., A Model-reference adaptive motion controller for a differential-drive mobile robot, IEEE International Conference *on Robotics and Automation*, San Diego, CA, May 8-13, 1994, 3091-3096.
32. Dhaouadi R. and Hatab A.A., Dynamic modelling of differential-drive mobile robots using lagrange and newton-euler methodologies: a unified framework, *Advances in Robotics and Automation*, 2(2), 2013, 2-7.
33. Slusny S., Neruda R. and Vidnerova P., Comparison of behavior-based and planning techniques on the small robot maze exploration problem, *Neural Networks*, 23(4), 2010, 560-567.
34. Ali W.G., A semi-autonomous mobile robot for education and research, *Journal of King Saud University-Engineering Sciences*, 23(2), 2011, 131–138.
35. Indiveri G., Nuchter A. and Lingemann K., High speed differential drive mobile robot path following control with bounded wheel speed commands, IEEE International conference *on Robotics and Automation*, Roma, Italy, April 10-14, 2007, 2202-2207.

36. Asano T., Asano T., Guibas L., Hershberger J. and Imai H., Visibility-polygon search and Euclidean shortest paths, 26th IEEE Symposium on *Foundations of Computer Science*, Portland, Oregon, October 21-23, 1985, 155-164.
37. Leiserson C. E., Rivest R. L. and Stein C., *Introduction to Algorithms*. T. H. Cormen (Ed.). MIT press, 2001.
38. Choset H., Lynch K. M., Hutchinson S., Kantor G., Burgard W., Kavraki L. E. and Thrun S., *Principles of Robot Motion*, MIT Press, Cambridge, MA, 2005.
39. Takahashi O. and Schilling R.J., Motion planning in a plane using generalized Voronoi diagrams, *IEEE Transactions on Robotics and Automation*, 5(2), 1989, 143-150.
40. Dunlaing C. O. and Yap C. K., A retraction method for planning the motion of a disc, *Journal of Algorithms*, 6, 1985, 104–111.
41. Bhattacharya P., and Gavrilova M. L., Roadmap-based path planning-Using the Voronoi diagram for a clearance-based shortest path. *IEEE Robotics and Automation Magazine*, 15(2), 2008, 58-66.
42. Masehian E. and Amin-Naseri M. R., A Voronoi diagram–visibility graph–potential field compound algorithm for robot path planning, *Journal of Robotic System*, 21, 2004, 275-300.
43. Yang D. H. and Hong S. K., A roadmap construction algorithm for mobile robot path planning using skeleton maps, *Advanced Robotics*, 21(1-2), 2007, 51-63.
44. Wein R., Van Den Berg J. P. and Halperin D., The visibility-voronoi complex and its applications, *Computational Geometry*, 36, 2007, 66–87.
45. Kavraki L. E., Svestka P., Latombe J.C. and Overmars M. H., Probabilistic roadmaps for path planning in high dimensional configuration spaces, *IEEE Transaction on Robotics and Automation*, 12, 1996, 566-580.
46. Sanchez G. and Latombe J., A Single-query Bidirectional Probabilistic Roadmap Planner with Lazy Collision Checking, *Springer Tracts in Advanced Robotics*, 6, 2001, 403-417.
47. Canny J. F., *The complexity of robot motion planning*, MIT Press Cambridge, MA, USA, 1988.
48. Bhattacharyya A., Singla E., and Dasgupta B., Robot path planning using silhouette method, 13th National Conference on *Mechanisms and Machines* (NaCoMM07), Bangalore, India, December 12-13, 2007.

49. Faverjon B. and Tournassoud P., A local approach for path planning of manipulators with a high number of degrees of freedom, IEEE International Conference on *Robotics and Automation*, Raleigh, North Carolina, March 30-April 03, 4, 1987, 1152-1159.
50. Fujii T., Arai Y., Asama H. and Endo I., Multilayered reinforcement learning for complicated collision avoidance problems, IEEE International Conference *on Robotics and Automation,* Leuven, Belgium, May 16-20, 3, 1998, 2186-2191.
51. Park K. H., Kim Y. J. and Kim J. H., Modular Q-learning based multi-agent cooperation for robot soccer, *Robotics and Autonomous Systems*, 35(2), 2001, 109-122.
52. Schwartz J. T. and Sharir M., On the "piano movers'" problem I. The case of a two-dimensional rigid polygonal body moving amidst polygonal barriers, *Communications on Pure and Applied Mathematics*, 36(3), 1983, 345-398.
53. Weigl M., Siemiaatkkowska B., Sikorski K.A. and Borkowski A., Grid- based mapping for autonomous mobile robot, *Robotics and Autonomous Systems*, 11(1), 1993, 13-21.
54. Zhu D. and Latombe J. C., New heuristic algorithms for efficient hierarchical path planning, *IEEE Transactions on Robotics and Automation*, 7(1), 1991, 9-20.
55. Conte G. and Zulli R., Hierarchical path planning in a multi-robot environment with a simple navigation function, *IEEE Transactions on Systems, Man and Cybernetics*, 25(4), 1995, 651-654.
56. Khatib O., Real time Obstacle Avoidance for manipulators and Mobile Robots, IEEE International Conference on *Robotics and Automation*, Missouri, March 25-28 1985, 500-505.
57. Garibotto G. and Masciangelo S., Path planning using the potential field approach for navigation, Fifth International Conference on *Advanced Robotics*, June 19-22, Pisa, Italy, 1991, 1679-1682.
58. Kim J. O. and Khosla P. K., Real-time obstacle avoidance using harmonic potential functions, *IEEE Transactions on Robotics and Automation*, 8(3), 1992, 338-349.
59. Ge S. S. and Cui Y. J., Dynamic motion planning for mobile robots using potential field method, *Autonomous Robots*, 13(3), 2002, 207-222.
60. Valavanis K. P., Hebert T., Kolluru R. and Tsourveloudis N., Mobile robot navigation in 2-D dynamic environments using an electrostatic potential field, *IEEE*

Transactions on Systems Man and Cybernetics, Part A: Systems and Humans, 30(2), 2000, 187-196.

61. Huang L., Velocity planning for a mobile robot to track a moving target-a potential field approach, *Robotics and Autonomous Systems*, 57(1), 2009, 55-63.
62. Shi P. and Zhao Y., An efficient path planning algorithm for mobile robot using improved potential field, IEEE International Conference on *Robotics and Biomimetics*, Guilin, China, December 19-23, 2009, 1704-1708.
63. Sfeir J., Saad M. and Saliah-Hassane H., An improved artificial potential field approaches to real-time mobile robot path planning in an unknown environment, IEEE International Symposium on *Robotic and Sensors Environments*, Montreal, Canada, September 17-18, 2011, 208-213.
64. Hui N. B., Coordinated motion planning of multiple mobile robots using potential field method, IEEE International Conference on *Industrial Electronics, Control and Robotics*, Orissa, December 27-29, 2010, 6-11.
65. Pimenta L. C., Fonseca A. R., Pereira G. A., Mesquita R. C., Silva E. J., Caminhas W. M. and Campos M. F., Robot navigation based on electrostatic field computation, *IEEE Transactions on Magnetics*, 42(4), 2006, 1459-1462.
66. Pradhan S. K., Parhi D. R., Panda A. K. and Behera R. K., Potential field method to navigate several mobile robots, *Applied Intelligence*, 25(3), 2006, 321-333.
67. Borenstein J. and Koren Y., Real-time obstacle avoidance for fast mobile robots, *IEEE Transactions on Systems, Man and Cybernetics*, 19(5), 1989, 1179-1187.
68. Biswas K. and Kar I., On reduction of oscillations in target tracking by artificial potential field method, 9th IEEE International Conference on *Industrial and Information Systems* Gwalior, India, December 15-17, 2014, 1-6.
69. Ishikawa S., A method of indoor mobile robot navigation by using fuzzy control, IEEE International Workshop on *Intelligent Robots and Systems*, Osaka, Japan, November 3-5, 2, 1991, 1013-1018.
70. Li Wei and Xun Feng, Behavior fusion for robot navigation in uncertain environments using fuzzy logic, IEEE International Conference on *Systems, Man, and Cybernetics*, San Antonio, Texas, October 2-5, 2, 1994, 1790-1796.
71. Reignier P., Fuzzy logic techniques for mobile robot obstacle avoidance, *Robotics and Autonomous Systems*, 12(3), 1994, 143-153.

72. Wu C. J., A learning fuzzy algorithm for motion planning of mobile robots, *Journal of Intelligent and Robotic Systems*, 11(3), 1994, 209-221.
73. Beom H. R. and Koh K. C., A sensor-based navigation for a mobile robot using fuzzy logic and reinforcement learning, *IEEE Transactions on systems, Man and Cybernetics*, 25(3), 1995, 464-477.
74. Lin C. H. and Wang L. L., Intelligent collision avoidance by fuzzy logic control, Robotics *and Autonomous Systems*, 20(1), 1997, 61-83.
75. Maaref H. and Barret C., Sensor-based fuzzy navigation of an autonomous mobile robot in an indoor environment, *Control Engineering Practice*, 8(7), 2000, 757-768.
76. Seraji H. and Howard A., Behavior-based robot navigation on challenging terrain: A fuzzy logic approach, *IEEE Transactions on Robotics and Automation,* 18(3), 2002, 308-321.
77. Abdessemed F., Benmahammed K. and Monacelli E., A fuzzy-based reactive controller for a non-holonomic mobile robot, *Robotics and Autonomous Systems*, 47(1), 2004, 31-46.
78. Parhi D. R., Navigation of mobile robots using a fuzzy logic controller, *Journal of intelligent and robotic systems*, 42(3), 2005, 253-273.
79. Huq R., Mann G. K. and Gosine R. G., Behavior-modulation technique in mobile robotics using fuzzy discrete event system, *IEEE Transactions on Robotics*, 22(5), 2006, 903-916.
80. Wang M. and Liu J. N., Fuzzy logic-based real-time robot navigation in unknown environment with dead ends. *Robotics and Autonomous Systems*, *56*(7), 2008, 625-643.
81. Selekwa M. F., Dunlap D. D., Shi D. and Collins E. G., Robot navigation in very cluttered environments by preference-based fuzzy behaviors, *Robotics and Autonomous Systems*, 56(3), 2008, 231-246.
82. Motlagh O. R. E., Hong T. S. and Ismail N., Development of a new minimum avoidance system for a behavior-based mobile robot, *Fuzzy Sets and Systems*, 160(13), 2009, 1929-1946.
83. Pradhan S. K., Parhi D. R. and Panda A. K., Fuzzy logic techniques for navigation of several mobile robots, *Applied soft computing*, 9(1), 2009, 290-304.

84. Obe O. and Dumitrache I., Fuzzy control of autonomous mobile robot, *University" Politehnica" of Bucharest Scientific Bulletin, Series C: Electrical Engineering*, 72(3), 2010, 173-186.
85. Samsudin K., Ahmad F. A. and Mashohor S., A highly interpretable fuzzy rule base using ordinal structure for obstacle avoidance of mobile robot, *Applied Soft Computing*, 11(2), 2011, 1631-1637.
86. Motlagh O., Tang S. H., Ismail N. and Ramli A. R., An expert fuzzy cognitive map for reactive navigation of mobile robots, *Fuzzy Sets and Systems*, 201, 2012, 105-121.
87. Mo H., Tang Q. and Meng L., Behavior-Based Fuzzy Control for Mobile Robot Navigation, *Mathematical Problems in Engineering*, DOI.org/10.1155/2013/561451, 2013.
88. Abdessemed F., Faisal M., Emmadeddine M., Hedjar R., Al-Mutib K., Alsulaiman M. and Mathkour H., A Hierarchical Fuzzy Control Design for Indoor Mobile Robot, *International Journal of Advanced Robot System*, 11, 2014, 33.
89. Abdelkrim N., Issam K., Lyes K. and Khaoula C., Fuzzy logic controllers for Mobile robot navigation in unknown environment using Kinect sensor, IEEE International Conference *on Systems, Signals and Image Processing*, Dubrovnik, Croatia, May 12-15, 2014, 75-78.
90. Cherroun L. and Boumehraz M., Fuzzy behavior based navigation approach for mobile robot in unknown environment, *Journal of Electrical Engineering*, 13, 2013, 1-8.
91. Koh K. C., Beom H. R., Kim J. S. and Cho H. S., A neural network-based navigation system for mobile robots, IEEE World Congress *on Computational Intelligence*, Orlando, Florida, June 27-July 02, 4, 1994, 2709-2714.
92. Thrun S., An approach to learning mobile robot navigation, *Robotics and Autonomous systems*, 15(4), 1995, 301-319.
93. Pal P. K. and Kar A., Mobile robot navigation using a neural net, IEEE International Conference *on Robotics and Automation*, Nagoya, Japan, May 21-27, 2, 1995, 1503-1508.
94. Glasius R., Komoda A. and Gielen S. C., Neural network dynamics for path planning and obstacle avoidance, *Neural Networks*, 8(1), 1995, 125-133.

95. Yang S. X. and Meng M., An efficient neural network method for real-time motion planning with safety consideration, *Robotics and Autonomous Systems*, 32(2), 200, 115-128.
96. Zarate L. E., Becker M., Garrido B. D. M. and Rocha H. S. C., An artificial neural network structure able to obstacle avoidance behavior used in mobile robots, 28th IEEE Annual Conference *on Industrial Electronics Society*, Sevilla, Spain, November 5-8, 3, 2002, 2457-2461.
97. Yao Q., Beetner D., Wunsch D. C. and Osterloh B., A RAM-based neural network for collision avoidance in a mobile robot, IEEE International Joint Conference *on Neural Networks*, Oregon, Portland, July 20-24, 4, 2003, 3157-3160.
98. Janglova D., Neural networks in mobile robot motion, *Cutting Edge Robotics*, 1(1), 2005, 243.
99. Castro V., Neira J. P., Rueda C. L., Villamizar J. C. and Angel L., Autonomous navigation strategies for mobile robots using a probabilistic neural network (PNN), 33rd IEEE Annual Conference *on Industrial Electronics Society*, Taipei, Taiwan, November 5-8, 2007, 2795-2800.
100. Wahab W., Autonomous mobile robot navigation using a dual artificial neural network, IEEE Region 10 Conference *TENCON*, Singapore, January 23-26, 2009, 1-6.
101. Fernandez-Leon J. A., Acosta G. G. and Mayosky M. A., Behavioral control through evolutionary neuro-controllers for autonomous mobile robot navigation, *Robotics and Autonomous Systems*, 57(4), 2009, 411-419.
102. Parhi D. R. and Singh M. K., Heuristic-rule-based hybrid neural network for navigation of a mobile robot, *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 224(7), 2010, 1103-1118.
103. Singh, M. K. and Parhi D. R., Path optimisation of a mobile robot using an artificial neural network controller, *International Journal of Systems Science*, 42(1), 2009, 107-120.
104. Engedy I. and Horvath G., Artificial neural network based local motion planning of a wheeled mobile robot, IEEE International Symposium *on Computational Intelligence and Informatics*, Budapest, Hungary, November 18-20, 2010, 213-218.

105. Mahmud F., Arafat A. and Zuhori S. T., Intelligent autonomous vehicle navigated by using artificial neural network, IEEE International Conference *on Electrical and Computer Engineering*, Dhaka, Bangladesh, December 20-22, 2012, 105-108.
106. Brahmi H., Ammar B. and Alimi A. M., Intelligent path planning algorithm for autonomous robot based on recurrent neural networks, IEEE International Conference *on Advanced Logistics and Transport*, Sousse, Tunisia, May 29-31, 2013, 199-204.
107. Motlagh O., Nakhaeinia D., Tang S. H., Karasfi B. and Khaksar W., Automatic navigation of mobile robots in unknown environments, *Neural Computing and Applications*, 24(7-8), 2014, 1569-1581.
108. Cao Z., Cheng L., Zhou C., Gu N., Wang X. and Tan M., Spiking neural network-based target tracking control for autonomous mobile robots, *Neural Computing and Applications*, 2015, DOI: 10.1007/s00521-015-1848-5.
109. Beom H. R. and Cho H. S. (1992). A sensor-based obstacle avoidance controller for a mobile robot using fuzzy logic and neural network, IEEE International Conference on *Intelligent Robots and Systems*, North Carolina, USA, July 7-10, 2, 1992, 1470-1475.
110. Song K. T. and Sheen L. H., Fuzzy-neuro control design for obstacle avoidance of a mobile robot, Fourth IEEE International Conference *on Fuzzy Systems and The Second International Fuzzy Engineering Symposium*, Yokohama, Tokyo, March 20-24, 1, 1995, 71-76.
111. Kubota N., Nojima Y., Kojima F. and Fukuda T., Multi-objective behavior coordinate for a mobile robot with fuzzy neural networks, IEEE International Joint Conference on *Neural Networks*, Como, Italy, 6, 2000, 311-316.
112. Ma X., L, X. and Qiao H., Fuzzy neural network-based real-time self-reaction of mobile robot in unknown environments, *Mechatronics*, 11(8), 2008, 1039-1052.
113. Er M. J., Tan T. P. and Loh S. Y., Control of a mobile robot using generalized dynamic fuzzy neural networks, *Microprocessors and Microsystems*, 28(9), 2004, 491-498.
114. Su Z., Zeng B., Liu G., Ye F. and Xu M., Application of fuzzy neural network in parameter optimization of mobile robot path planning using potential field, IEEE International Symposium *on Industrial Electronics*, Vigo, Spain, June 4-7, 2007, 2125-2128.

115. He K., Gao Y. and Sun H., A Fuzzy Neural Network Based on TS Model for Mobile Robots to Avoid Obstacles, *Intelligent Robotics and Applications*, LNCS, 5314, 2008, 1127-1134.
116. Shi W., Wang K. and Yang S. X., A fuzzy-neural network approach to multisensor integration for obstacle avoidance of a mobile robot, *Intelligent Automation and Soft Computing*, 15(2), 2009, 289-301.
117. Jolly K. G., Kumar R. S. 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(6), 2010, 923-933.
118. Chohra, A. and Azouaoui, O., Navigation behaviors based on fuzzy ArtMap neural networks for intelligent autonomous vehicles, *Advances in Artificial Neural Systems*, Article ID 523094, 2011, DOI:10.1155/2011/523094.
119. Jeffril M. A. and Sariff N., The integration of fuzzy logic and artificial neural network methods for mobile robot obstacle avoidance in a static environment, IEEE 3rd International Conference *on System Engineering and Technology*, Shah Alam, Malaysia, August 19-20, 2013, 325-330.
120. Li W., Neuro-fuzzy systems for intelligent robot navigation and control under uncertainty, IEEE International Joint Conference *on Fuzzy Systems and The Second International Fuzzy Engineering Symposium*, Yokohama, Japan, March 20-24, 4, 1995, 1747-1754.
121. Ng K. C. and Trivedi M. M., A neuro-fuzzy controller for mobile robot navigation and multirobot convoying., *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 28(6), 2008, 829-840.
122. Tsoukalas L. H., Houstis E. N. and Jones G. V., Neuro-fuzzy motion planners for intelligent robots, *Journal of Intelligent and Robotic Systems*, 19(3), 1997, 339-356.
123. Godjevac J., and Steele N., Neuro-fuzzy control of a mobile robot, *Neuro-Computing*, 28(1), 1999, 127-143.
124. Nefti S., Oussalah M., Djouani K. and Pontnau J., Intelligent adaptive mobile robot navigation, *Journal of Intelligent and Robotic Systems*, 30(4), 2001, 311-329.
125. Marichal G. N., Acosta L., Moreno L., Méndez J. A., Rodrigo J. J. and Sigut, M., Obstacle avoidance for a mobile robot: A neuro-fuzzy approach, *Fuzzy Sets and Systems*, 124(2), 2001, 171-179.

126. Krishna K. M. and Kalra P. K., Perception and remembrance of the environment during real-time navigation of a mobile robot, *Robotics and Autonomous Systems*, 37(1), 2001, 25-51.
127. Wei W., Mbede J. B. and Zhang Y., Neuro-fuzzy motion control for the mobile robot, IEEE International Joint Conference *on Neural Networks*, Honolulu, Hawaii, May 12-17, 1, 2002, 507-512.
128. Ye C., Yung N. H. and Wang D., A fuzzy controller with supervised learning assisted reinforcement learning algorithm for obstacle avoidance, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 33(1), 2003, 17-27.
129. Rusu P., Petriu E. M., Whalen T. E., Cornell A. and Spoelder H. J., Behavior-based neuro-fuzzy controller for mobile robot navigation, *IEEE Transactions on Instrumentation and Measurement*, 52(4), 1335-1340.
130. Wang X., Yang S. X. and Meng M. H., Intelligent obstacle avoidance for an autonomous mobile robot, IEEE Fifth World Congress *on Intelligent Control and Automation*, Hangzhou, China, June 15-19, 5, 2004, 4656-4660.
131. Er M. J. and Deng C., Obstacle avoidance of a mobile robot using hybrid learning approach, *IEEE Transactions on Industrial Electronics*, 52(3), 2005, 898-905.
132. Pradhan S. K., Parhi D. R. and Panda A. K., Neuro-fuzzy technique for navigation of multiple mobile robots, *Fuzzy Optimization and Decision Making*, 5(3), 2006, 255-288.
133. Hui N. B., Mahendar V. and Pratihar D. K., Time-optimal, collision-free navigation of a car-like mobile robot using neuro-fuzzy approaches, *Fuzzy Sets and systems*, 157(16), 2006, 2171-2204.
134. Zhu A. and Yang S. X., Neuro-fuzzy-based approach to mobile robot navigation in unknown environments, *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 37(4), 2007, 610-621.
135. Parhi D. R. and Singh M. K., Navigational path analysis of mobile robots using an adaptive neuro-fuzzy inference system controller in a dynamic environment, *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 224(6), 2010, 1369-1381.
136. Demirli K. and Khoshnejad M., Autonomous parallel parking of a car-like mobile robot by a neuro-fuzzy sensor-based controller, *Fuzzy Sets and Systems*, 160(19), 2009, 2876-2891.

137. Park J. I., Cho J. H., Chun M. G. and Song C. K., Neuro-fuzzy rule generation for backing up navigation of car-like mobile robots, *International Journal of Fuzzy Systems*, 11(3), 192-201.
138. Woo P. Y. and Polisetty V., ANFIS generated dynamic path planning for a mobile robot to track a randomly moving target in a 3-D space with obstacle avoidance, IEEE International Conference *on Fuzzy Systems*, Barcelona, Spain, July 18-23, 2010, 1-8.
139. Joshi M. M. and Zaveri M. A., Neuro-fuzzy based autonomous mobile robot navigation system, IEEE International Conference *on Control Automation Robotics and Vision*, Singapore, December 7-10, 2010, 384-389.
140. Erdem H., Application of neuro-fuzzy controller for sumo robot control, *Expert Systems with Applications*, 38(8), 2011, 9752-9760.
141. Al Mutib K. and Mattar E., Neuro-fuzzy controlled autonomous mobile robotics system, IEEE 13th International Conference *on Computer Modelling and Simulation*, Cambridge UK, March 30-April 01, 2011, 1-7.
142. Rouabah H., Abdelmoula C. and Masmoudi M., Behavior control of a mobile robot based on Fuzzy logic and Neuro-fuzzy approaches for monitoring wall, IEEE 7th International Conference on *Design and Technology of Integrated Systems in Nanoscale Era*, Gammarth, Tunisia, May 16-18, 2012,1-6.
143. Al-Mayyahi A., Wang W. and Birch P., Adaptive Neuro-Fuzzy Technique for Autonomous Ground Vehicle Navigation, *Robotics*, 3(4), 2014, 349-370.
144. Baturone I., Gersnoviez A. and Barriga Á., Neuro-fuzzy techniques to optimize an FPGA embedded controller for robot navigation, *Applied Soft Computing*, 21, 2014, 95-106.
145. Algabri M., Mathkour H. and Ramdane H., Mobile Robot Navigation and Obstacle-avoidance using ANFIS in Unknown Environment, *International Journal of Computer Applications*, 91(14), 2014, 36-41.
146. Melingui A., Merzouki R. and Mbede J. B., Neuro-fuzzy controller for autonomous navigation of mobile robots, IEEE Conference on *Control Applications*, Juan Les Antibes, France, October 8-10, 2014, 1052-1057.
147. Pothal J. K. and Parhi D. R., Navigation of multiple mobile robots in a highly clutter terrains using adaptive neuro-fuzzy inference system, *Robotics and Autonomous Systems*, 2015, http://dx.doi.org/10.1016/j.robot.2015.04.007-0921-8890.

148. Zhang N, Beetner D.,Wunsch D.C., Hemmelman B., Hasan A., An embedded real-time neuro-fuzzy controller for mobile robot navigation, IEEE International conference *on Fuzzy Systems*, May 25-25, Reno, Neveda, 2005, 319–324.
149. Takagi H. and Hayashi I., NN-driven fuzzy reasoning, *International Journal of Approximate Reasoning*, 5(3), 1991, 191-212.
150. Jang J.S.R., ANFIS: Adaptive network-based fuzzy inference system, *IEEE Transaction on System, Man and Cybernetics-Part B*, 23(3), 1993, 665-685.
151. Pratihar D. K., Deb K. and Ghosh A., Fuzzy-genetic algorithms and time-optimal obstacle-free path generation for mobile robots, *Engineering Optimization*, 32(1), 1999, 117-142.
152. Nearchou A. C., Path planning of a mobile robot using genetic heuristics, *Robotica*, 16(05), 1998, 575-588.
153. Kubota N., Morioka T., Kojima F. and Fukuda T., Learning of mobile robots using perception-based genetic algorithm, *Measurement*, 29(3), 2011, 237-248.
154. Navarro N., Muñoz C., Freund W. and Arredondo V. T., Acquiring Adaptive Behaviors of Mobile Robots Using Genetic Algorithms and Artificial Neural Networks, IEEE Conference *on Electronics, Robotics and Automotive Mechanics*, Morelos, Mexico, September 26–29, 1, 2006, 87-91.
155. Liu Q., Lu Y. G. and Xie C. X., Optimal genetic fuzzy obstacle avoidance controller of autonomous mobile robot based on ultrasonic sensors, IEEE International Conference on *Robotics and Biomimetics*, Kunming, China, December 17-20, 2006,125-129.
156. Castillo O., Trujillo L. and Melin P., Multiple objective genetic algorithms for path-planning optimization in autonomous mobile robots, *Soft Computing*, 11(3), 2007, 269-279.
157. Hassanzadeh I. and Sadigh S. M., Path planning for a mobile robot using fuzzy logic controller tuned by GA, 6th IEEE International Symposium *on Mechatronics and its Applications*, Sharjah, UAE, March 23-26, 2009, 1-5.
158. Ismail A. T., Sheta A. and Al-Weshah M., A mobile robot path planning using genetic algorithm in static environment, *Journal of Computer Science*, 4(4), 2008, 341-344.
159. Liu C., Liu H. and Yang J., A path planning method based on adaptive genetic algorithm for mobile robot, *Journal of Information and Computational Science*, 8(5), 2011, 808-814.

160. Mohanta J. C., Parhi D. R. and Patel S. K., Path planning strategy for autonomous mobile robot navigation using Petri-GA optimization, *Computers and Electrical Engineering*, 37(6), 2011, 1058-1070.
161. Zhang L., Min H., Wei H. and Huang H., Global path planning for mobile robot based on A∗ algorithm and genetic algorithm, IEEE International Conference *on Robotics and Biomimetics*, Guangzhou, Hongkong, December 11-14, 2012, 1795-1799.
162. Nagib G. and Gharieb W., Path planning for a mobile robot using genetic algorithms, ICEEC International Conference *on Electrical, Electronic and Computer Engineering*, Cairo, Egypt, September 5-7, 2004, 185-189.
163. Jiang M., Fan X., Pei Z., Jiang J., Hu Y. and Wang, Q., Robot Path Planning Method Based on Improved Genetic Algorithm, *Sensors and Transducers*,166(3), 2014, 255-260.
164. Zhu Z., Wang F., He, S. and Sun Y., Global path planning of mobile robots using a memetic algorithm, *International Journal of Systems Science*, 46(11), 2015, 1982-1993.
165. Wang J., Zhang Y. and Xia L., Adaptive genetic algorithm enhancements for path planning of mobile robots, International conference *on Measuring Technology and Mechatronics Automation*, Changsha city, China, March 13-14, 3, 2010, 416–419.
166. Chen X., and Li Y., Smooth path planning of a mobile robot using stochastic particle swarm optimization, IEEE International Conference *on Mechatronics and Automation*, Luoyang, Henan, June 25-28, 2006, 1722-1727.
167. Lu L. and Gong D., Robot path planning in unknown environments using particle swarm optimization, IEEE Fourth International Conference *on Natural Computation*, Jinan, China, October 18-20, 4, 2008, 422-426.
168. Gong D., Lu L. and Li M., Robot path planning in uncertain environments based on particle swarm optimization, IEEE Congress *on Evolutionary Computation*, Trondheim, Norway, May 18-21, 2009, 2127-2134.
169. Masehian E. and Sedighizadeh D., A multi-objective PSO-based algorithm for robot path planning, IEEE International Conference *on Industrial Technology*, Vi a del Mar, Chile, March 14-17, 2010, 465-470.
170. Li Q., Tang Y., Wang L., Zhang C. and Yin Y., A Specialized Particle Swarm Optimization for global path planning of mobile robots, IEEE Third International

Workshop *on Advanced Computational Intelligence*, Suzhou, Jiangsu, August 25-27, 2010, 271-276.

171. Gong D. W., Zhang J. H. and Zhang Y., Multi-objective particle swarm optimization for robot path planning in environment with danger sources, *Journal of Computers*, 6(8), 2011, 1554-1561.
172. Šuaić M., Ćosić A., Ribić A. and Katić D., An approach for intelligent mobile robot motion planning and trajectory tracking in structured static environments, IEEE 9th International Symposium *on Intelligent Systems and Informatics*, Subotica, Serbia, September 8-10, 2011, 17-22.
173. Tang Q. and Eberhard P., Cooperative motion of swarm mobile robots based on particle swarm optimization and multibody system dynamics, *Mechanics based design of Structures and Machines*, 39(2), 2011, 179-193.
174. Mohamed A. Z., Lee S. H., Hsu H. Y. and Nath, N., A faster path planner using accelerated particle swarm optimization, *Artificial Life and Robotics*, 17(2), 2012, 233-240.
175. Mohajer B., Kiani K., Samiei E. and Sharifi M., A new online random particles optimization algorithm for mobile robot path planning in dynamic environments, *Mathematical Problems in Engineering*, 2013, http://dx.doi.org/10.1155/2013/491346.
176. Zhang Y., Gong D. W. and Zhang J. H., Robot path planning in uncertain environment using multi-objective particle swarm optimization, *Neuro-computing*, 103, 2013, 172-185.
177. Cai Y. and Yang S. X., An improved PSO-based approach with dynamic parameter tuning for cooperative multi-robot target searching in complex unknown environments, *International Journal of Control*, 86(10), 2013, 1720-1732.
178. Deepak B. B. V. L., Parhi D. R. and Raju B. M. V. A., Advance particle swarm optimization-based navigational controller for mobile robot, *Arabian Journal for Science and Engineering*, 39(8), 2014, 6477-6487.
179. Mo H. and Xu L., Research of biogeography particle swarm optimization for robot path planning, *Neuro-computing*, 148, 2015, 91-99.
180. Ioannidis K., Sirakoulis G. C. and Andreadis I., Cellular ants: a method to create collision-free trajectories for a cooperative robot team, *Robotics and Autonomous Systems*, 59(2), 113-127. year

181. Chen X., Kong Y., Fang X. and Wu Q., A fast two-stage ACO algorithm for robotic path planning, *Neural Computing and Applications*, 22(2), 2013, 313-319.
182. Parhi D. R. and Pothal J. K., Intelligent navigation of multiple mobile robots using an ant colony optimization technique in a highly cluttered environment, *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 225(1), 2011, 225-232.
183. Hsu C. C., Hou R. Y. and Wang W. Y., Path planning for mobile robots based on improved ant colony optimization, IEEE International Conference on *Systems, Man, and Cybernetics*, Manchester, UK, October 13-16, 2013, 2777-2782.
184. Yuan M., Wang S. and Li P, A model of ant colony and immune network and its application in path planning, Singapore, June 3-5, 3rd IEEE Conference on *Industrial Electronics and Applications*, 2008, 102-107.
185. Cen Y., Song C. and Xie N., Path planning method for mobile robot based on ant colony optimization algorithm, IEEE Conference *on Industrial Electronics and Applications*, Singapore, 2008, June 3-5, 289-301.
186. Deepak B. B. V. L. and Parhi D, Intelligent adaptive immune-based motion planner of a mobile robot in cluttered environment, *Intelligent Service Robotics*, 6(3), 2013, 155-162.
187. Yuan M., Wang S. A., Wu C. and Chen N., A novel immune network strategy for robot path planning in complicated environments, *Journal of Intelligent and Robotic Systems*, 60(1), 2010, 111-131.
188. Hu X. and Xu Q., Robot path planning based on artificial immune network, IEEE International Conference *on Robotics and Biomimetics*, Sanya, China, December 15-18, 2007, 1053-1057.
189. Luh G. C. and Liu W. W., An immunological approach to mobile robot reactive navigation, *Applied Soft Computing*, *8*(1), 2008, 30-45.
190. Whitbrook A. M., Aickelin U. and Garibaldi J. M., Real-world transfer of evolved artificial immune system behaviours between small and large scale robotic platforms, *Evolutionary Intelligence*, 3(3-4), 2010, 123-136.
191. Wang Y. N., Hsu H. H. and Lin C. C., Artificial immune algorithm based obstacle avoiding path planning of mobile robots, First International Conference on *Advances in Natural Computation*, Changsha, China, August 27-29, 2005, 859-862.

192. Yang X. S. and Deb S., Cuckoo search via Lévy flights, World Congress *on Nature and Biologically Inspired Computing*, December 9-11, Coimbatore, India, 2009, 210-214.
193. Yang, X. S. and Deb, S., Engineering optimisation by cuckoo search, International *Journal of Mathematical Modelling and Numerical Optimisation*, 1(4), 330-343.
194. Valian E., Mohanna S. and Tavakoli S., Improved cuckoo search algorithm for feedforward neural network training, *International Journal of Artificial Intelligence and Applications*, *2*(3), 2011, 36-43.
195. Walton S., Hassan O., Morgan K. and Brown M. R., Modified cuckoo search: a new gradient free optimisation algorithm, *Chaos, Solitons and Fractals*, 44(9), 2011, 710-718.
196. Vazquez R., Training spiking neural models using cuckoo search algorithm, IEEE Congress *on Evolutionary Computation*, New Orleans, LA, June 6-8, 2011, 679-686.
197. Wang G., Guo L., Duan H., Liu L., Wang H. and Wang B., A hybrid meta-heuristic DE/CS algorithm for UCAV path planning, *Journal of Information and Computational Science*, 5(16), 2012, 4811-4818.
198. Jati G. K., Manurung H. M. and Suyanto S., Discrete cuckoo search for traveling salesman problem, IEEE International Conference *on Computing and Convergence Technology*, Seoul, South Korea, December 3-5, 2012, 993-997.
199. Chaowanawatee K. and Heednacram A., Implementation of cuckoo search in RBF neural network for flood forecasting, IEEE Fourth International Conference *on Computational Intelligence, Communication Systems and Networks*, Phuket, Thailand, July 24-26, 2012, 22-26.
200. Chandrasekaran K. and Simon S. P., Multi-objective scheduling problem: hybrid approach using fuzzy assisted cuckoo search algorithm, *Swarm and Evolutionary Computation*, 5, 2012, 1-16.
201. Civicioglu P. and Besdok E., A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms, *Artificial Intelligence Review*, 39(4), 2013, 315-346.
202. Yang, X. S. and Deb S., Multiobjective cuckoo search for design optimization, *Computers and Operations Research*, 40(6), 2013, 1616-1624.

203. Savsani P., Jhala R. L. and Savsani V. J., Comparative study of different metaheuristics for the trajectory planning of a robotic arm, *IEEE system Journal*, 2014, DOI: 10.1109/JSYST.2014.2342292.
204. Nilakantan J. M., Ponnambalam S. G., Jawahar N. and Kanagaraj G., Bio-inspired search algorithms to solve robotic assembly line balancing problems, *Neural Computing and Applications*, 2015, DOI 10.1007/s00521-014-1811-x.
205. Yang X. S. and Deb S., Cuckoo search: recent advances and applications, *Neural Computing and Applications*, 24(1), 2014, 169-174.
206. Suja K. R. and Raglend J., Cuckoo search (CS)-NFC-based UPQC for compensating voltage sag of nonlinear load, *Journal of Experimental and Theoretical Artificial Intelligence*, 2015, DOI:10.1080/0952813X.2014.971466.
207. Ouaarab A., Ahiod B. and Yang X. S., Discrete cuckoo search algorithm for the travelling salesman problem, *Neural Computing and Applications*, 24(7-8), 2014, 1659-1669.
208. Burnwal S. and Deb S., Scheduling optimization of flexible manufacturing system using cuckoo search-based approach, *The International Journal of Advanced Manufacturing Technology*, 64(5-8), 2013, 951-959.
209. Mehrabian A. R. and Lucas C., A novel numerical optimization algorithm inspired from weed colonization, *Ecological informatics*, 1(4), 2006, 355-366.
210. Hajimirsadeghi H. and Lucas C., A hybrid IWO/PSO algorithm for fast and global optimization, St.-Petersburg, Russia, IEEE EUROCON, 2009, 1964-1971.
211. Krishnanand K. R., Nayak S. K., Panigrahi B. K. and Rout P. K., Comparative study of five bio-inspired evolutionary optimization techniques, World Congress *on Nature and Biologically Inspired Computing*, December 9-11, Coimbatore, India, 2009, 1231-1236.
212. Ghalenoei M., Hajimirsadeghi H. and Lucas C., Discrete invasive weed optimization algorithm: application to cooperative multiple task assignment of UAVs, IEEE 28th Chinese Control Conference *on Decision and Control*, Shanghai, China, December 15-18, 2009, 1665-1670.
213. Sengupta A., Chakraborti T., Konar A. and Nagar A., Energy efficient trajectory planning by a robot arm using invasive weed optimization technique, IEEE 3rd World Congress *on Nature and Biologically Inspired Computing*, Salamanca, Spain, October 19-21, 2011, 311-316.

214. Kundu D., Suresh K., Ghosh S., Das S., Panigrahi B. K. and Das S., Multi-objective optimization with artificial weed colonies, *Information Sciences*, 181(12), 2011, 2441-2454.
215. Yin Z., Wen M. and Ye C., Improved invasive weed optimization based on hybrid genetic algorithm, *Journal of Computational Information Systems*, 8(8), 2012, 3437-3444.
216. Basak A., Maity D. and Das S., A differential invasive weed optimization algorithm for improved global numerical optimization, *Applied Mathematics and Computation*, 219(12), 2013, 6645-6668.
217. Rani D. S., Subrahmanyam N. and Sydulu M., Multi-Objective Invasive Weed Optimization–An application to optimal network reconfiguration in radial distribution systems, *International Journal of Electrical Power & Energy Systems*, 73, 2015, 932-942.
218. Nikoofard A. H., Hajimirsadeghi H., Rahimi-Kian A. and Lucas C., Multiobjective invasive weed optimization: Application to analysis of Pareto improvement models in electricity markets, *Applied Soft Computing*, 12(1), 2012, 100-112.
219. Karimkashi S. and Kishk, A., Invasive weed optimization and its features in electromagnetics, *IEEE Transactions on Antennas and Propagation*, 58(4), 2010, 1269-1278.
220. Mallahzadeh A. R. R., Oraizi H. and Davoodi-Rad Z., Application of the invasive weed optimization technique for antenna configurations, *Progress in Electromagnetics Research*, 79, 2008, 137-150.
221. Shoorehdeli M. A., Teshnehlab M., Sedigh A. K. and Khanesar M. A., Identification using ANFIS with intelligent hybrid stable learning algorithm approaches and stability analysis of training methods, *Applied Soft Computing*, 9(2), 2009, 833-850.
222. Jiang H. M., Kwong C. K., Ip W. H. and Wong T. C., Modeling customer satisfaction for new product development using a PSO-based ANFIS approach, *Applied Soft Computing*, *12*(2), 2012, 726-734.
223. Siegwart R., Nourbakhsh I. R., and Scaramuzza D., *Introduction to autonomous mobile robots*, MIT press, 2011.
224. Peterson J. L., *Petri Net Theory and the Modeling of Systems*, 1981, Prentice Hall.
225. Kashima H. and Masuda R., Cooperative Control of Mobile Robots Based on Petri-Net, 10th International Conference on Advanced Robotics, 2001, 339-404.

226. Reynolds A. M. and Rhodes C. J., The Levy flight paradigm: Random search patterns and Mechanisms, *Ecology*, 90, 2009, 877–887.
227. Liang X. D.,Wu J. G., Li L.Y. and Chen H. N., Mobile robot path planning based on adaptive bacterial foraging algorithm, *Journal of Central South University*, 20(12), 2013, 3391-3400.
228. MATLAB Toolbox Release 2012b, The Math Works, Inc., Natick, Massachusetts, United States.

LIST OF PUBLICATIONS

PAPER PUBLISED IN INTERNATIONAL JOURNALS:

1. Dayal R. Parhi and Prases K. Mohanty, 2015. "IWO Based Adaptive Neuro-Fuzzy Controller for Mobile Robot Navigation in Cluttered Environments", *International Journal of Advanced Manufacturing Technology,* Springer-Verlag Berlin Heidelberg, DOI: 10.1007/s00170-015-7512-5.
2. Prases K.Mohanty and Dayal R.Parhi, 2015. "A New Hybrid Optimization Algorithm for Multiple Mobile Robots Navigation Based on the CS-ANFIS Approach", *Memetic Computing*, Springer-Verlag Berlin Heidelberg, DOI:10.1007/s12293-015-0160-3.
3. Prases K. Mohanty and Dayal R. Parhi, 2014. "Optimal path planning for a mobile robot using cuckoo search algorithm", *Journal of Experimental and Theoretical Artificial Intelligence*, Taylor & Francis, DOI: 10.1080/0952813X.2014.971442.
4. Prases K.Mohanty and Dayal R.Parhi, 2014. "Navigation of Autonomous Mobile Robot using Adaptive Network based Fuzzy Inference System", *Journal of Mechanical Science and Technology*, Volume-28 (7), pp. 2861-2868, Springer-Verlag Berlin Heidelberg.
5. Prases K.Mohanty and Dayal R.Parhi, 2013. "A New Intelligent Motion Planning for Mobile Robot Navigation using Multiple Adaptive Neuro-Fuzzy Inference System", *International Journal of Applied Mathematics and Information Science*, Volume 8, No. 5, pp.2527-2535.
6. Prases K. Mohanty and Dayal R. Parhi, 2014. "A New Efficient Optimal Path Planner for Mobile Robot Based on Invasive Weed Optimization Algorithm", *Frontiers of Mechanical Engineering*, Springer-Verlag Berlin Heidelberg, Volume 9(4), pp.317-330.
7. Prases K. Mohanty and Dayal R. Parhi, 2014. "A New Hybrid Intelligent Path Planner for Mobile Robot Navigation Based on Adaptive Neuro-Fuzzy Inference System", *Australian Journal of Mechanical Engineering*, Taylor & Francis (Accepted).
8. Dayal R.Parhi and Prases K.Mohanty, 2012. "Kinematic Modeling of an Autonomous Wheeled Robot", *International Journal of Applied Artificial Intelligence in Engineering System*, Vol.4 No.2, pp.63-73.
9. Prases K.Mohanty and Dayal R.Parhi, 2013. "Controlling the Motion of an Autonomous Mobile Robot Using Various Techniques: a Review", Journal of

Advance Mechanical Engineering, Columbia International Publishing, Vol.1, pp.24-39.

SPRINGER BOOK CHAPTERS:

1. Prases K.Mohanty, Dayal R.Parhi, 2012. "Path Generation and Obstacle Avoidance of an Autonomous Mobile Robot Using Intelligent Hybrid Controller", *Lectures Notes in Computer Science (LNCS) -7677*, Springer-Verlag Berlin Heidelberg, pp. 240–247.
2. Prases K.Mohanty, Dayal R.Parhi, 2013. "Navigation of Autonomous Mobile Robot using Adaptive Neuro-Fuzzy Controller", *Advances in Intelligent Systems and Computing (AISC)-243*, Springer-International Publishing, Switzerland, pp. 521-530.
3. Prases K.Mohanty, Dayal R.Parhi, 2013. "A New Intelligent Approach for Mobile Robot Navigation", *Lectures Notes in Computer Science (LNCS)-8251*, Springer-Verlag Berlin Heidelberg.
4. Prases K.Mohanty, Dayal R.Parhi, 2013. "Path Planning Strategy for Mobile Robot Navigation using MANFIS Controller", *Advances in Intelligent Systems and Computing (AISC)-247*, Springer-International Publishing, Switzerland, pp. 353-361.
5. Prases K.Mohanty, Krishna K. Pandey, Dayal R.Parhi, 2013. "MANFIS Approach for Path Planning and Obstacle Avoidance for Mobile Robot Navigation", *Advances in Intelligent Systems and Computing (AISC)-248*, Springer-International Publishing Switzerland, pp. 361-270.
6. Prases K.Mohanty, Dayal R.Parhi, 2013. "Cuckoo Search Algorithm for the Mobile Robot Navigation", *Lectures Notes in Computer Science (LNCS)-8297*, Springer-Verlag Berlin, Heidelberg, pp. 240–247.
7. Prases K.Mohanty, Sandeep kumar, Dayal R.Parhi, 2014. "A New Ecologically Inspired Algorithm for Mobile Robot Navigation", *Advances in Intelligent Systems and Computing (AISC)-327*, Springer-International Publishing, Switzerland, pp.755-762.

PAPER PRESENTED IN INTERNATIONAL/NATIONAL CONFERENCES:

1. Prases K.Mohanty and Dayal R.Parhi, 2012. "Navigation of an Autonomous Mobile Robot Using Intelligent Hybrid Technique", 2012 ***IEEE*** *International Conference on Advanced Communication Control and Computing Technologies* (ICACCCT), 23-25 August, Ramanathapuram, Tamil Naidu, pp.136-140.
2. Prases K.Mohanty and Dayal R.Parhi, 2013. "Kinematic Analysis of Autonomous Wheeled Robot", ***IEEE*** *International Conference on Research and Development Prospects on Engineering and Technology* (ICRDPET), 29-30 March, Nagapattinam, Tamil Naidu, Vol.1, pp. 18-23.
3. Prases K.Mohanty and Dayal R.Parhi, Alok K.Jha, Anish Pandey, 2013. "Path Planning of an Autonomous Mobile Robot using Adaptive Network based Fuzzy Controller", *3rd* ***IEEE*** *International Advance Computing Conference* (IACC-2013), 22-23 February, New Delhi, pp. 651-656.
4. Prases K.Mohanty and Dayal R.Parhi, 2013. "Navigation Strategy of Autonomous Mobile Robot using MANFIS", ***IEEE*** International Conference on Computing, Cybernetics and Intelligent Information System 21-23 November 2013, Vellore, India.
5. Krishna K. Pandey, Prases K. Mohanty and Dayal R. Parhi, 2014. "Real time navigation strategies for Webots using Fuzzy controller", ***IEEE*** 8th International Conference on Intelligent Systems and control, 10-11 January, Coimbatore, India.
6. Prases K.Mohanty and Dayal R.Parhi, 2014. "A New Real Time Path Planning for Mobile Robot Navigation using Invasive Weed Optimization Algorithm", ***ASME*** Gas Turbine India Conference, 15-17, December, New Delhi.
7. Mahesh S. Pol, Prases K. Mohanty and Dayal R. Parhi, 2014. "A Review Paper on Navigation of Autonomous Mobile Robot Using Various AI (Artificial Intelligence) Techniques", International Conference on Design, Manufacturing and Mechatronics 2014 (ICDMM2014), 9-10 January, Pune, Maharashtra, India.
8. Prases K.Mohanty and Dayal R.Parhi, 2014. "Path Planning Optimization for Mobile Robot Based on Cuckoo Search Algorithm", International Conference on Innovation in Design, Manufacturing and Concurrent Engineering (IDMC 2014), 01st-03rd, March 2014, NIT, Rourkela.

APPENDIX-A

Details description of Experimental Mobile Robots

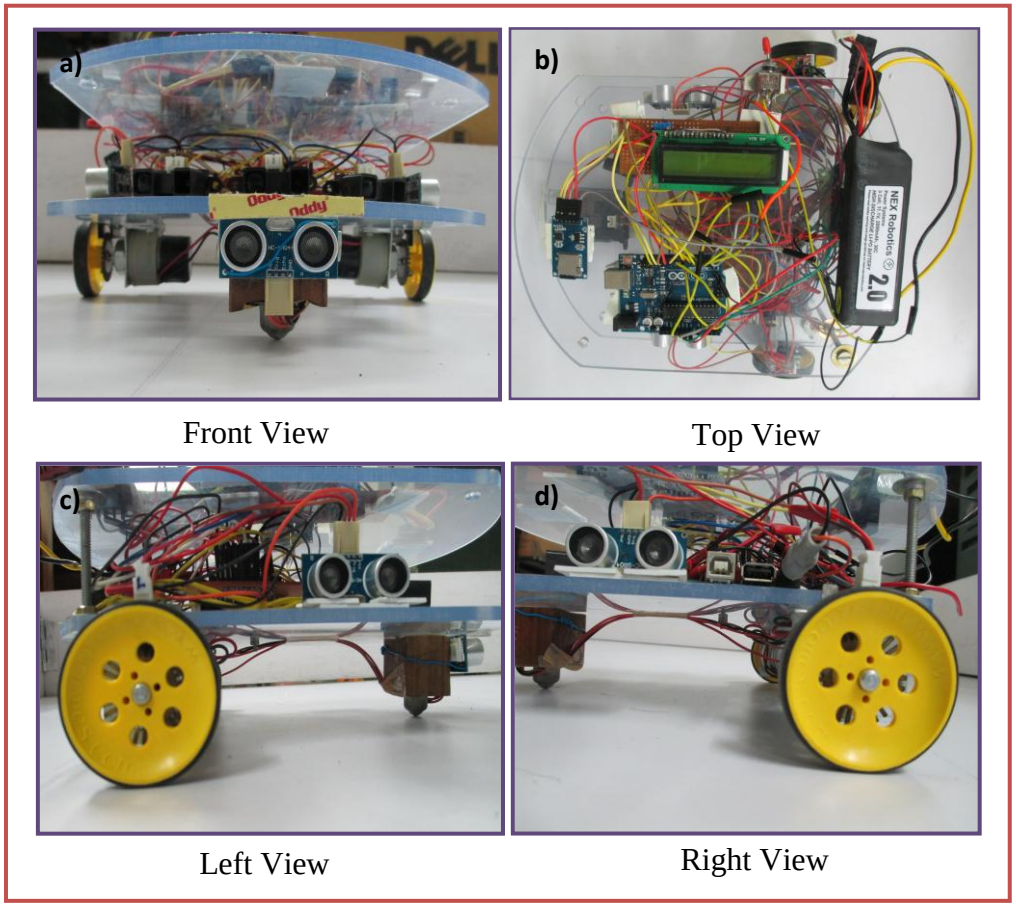


Figure A.1 (a-d) Differential Mobile Robot with different views.

Table A.1 Technical specification of Differential Mobile robot

	Elements	Technical specification
1	Processor	ATmega2560 (Arduino Mega 2560, Arduino UNO)
2	RAM	8KB, EEROM-4KB
3	Flash	256 KB (8KB used for boot loader)
4	Motors	2-DC gear motors with incremental encoders
5	Sensors	3 Infrared sensors with up to 150cm range. 3 Ultrasonic sensors with up to 400cm range.
6	Speed	Max: 0.47m/s, Min:0.03m/s
7	Power	Power adapter or Rechargeable NiMH Battery (2000mAh)
8	Communication	USB connection to the computer
9	Size	Length: 26cm, Width: 20cm, Height: 12 cm
10	Weight	Approx. 1300 g
11	Payload	Approx. 4200g
12	Remote control Software via USB cable	C/C++ ® (on PC, MAC) MATLAB ® (on PC, MAC, Linux)

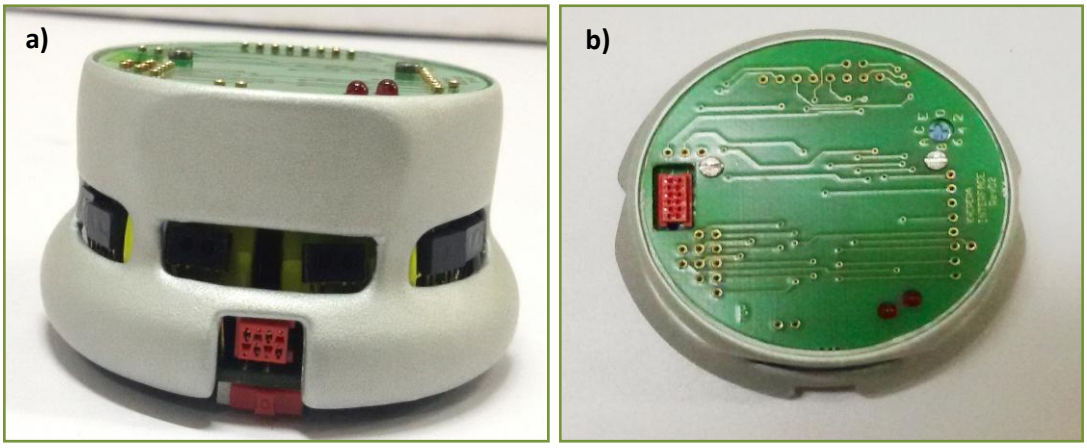


Figure A.2 (a-b) Khepera II Robot with front and top view.

Table A.2 Technical specification of Khepera II Mobile robot

	Elements	Technical specification
1	Processor	Motorola 68331 CPU, 25MHz
2	RAM	512KB
3	Flash	512 KB
4	Motors	2-DC brushed Servo motors with incremental encoders
5	Sensors	8 Infrared proximity and ambient light sensors with up to 100mm range.
6	Speed	Max: 0.5m/s, Min:0.02m/s
7	Power	Power adapter or Rechargeable NiMH Batteries
8	Communication	Standard Serial Port, up to 115KB/S
9	Size	Diameter: 70 mm , Height: 30 mm
10	Weight	Approx. 80 g
11	Payload	Approx. 250g
12	Remote control Software via tether or	LabVIEW ® (on PC, MAC or SUN) using RS232

radio	MATLAB ® (on PC, MAC, Linux or SUN) using RS232 SysQuake ® (on PC, MAC, Linux or SUN) using RS232 Freeware Any other software capable of RS232 communication

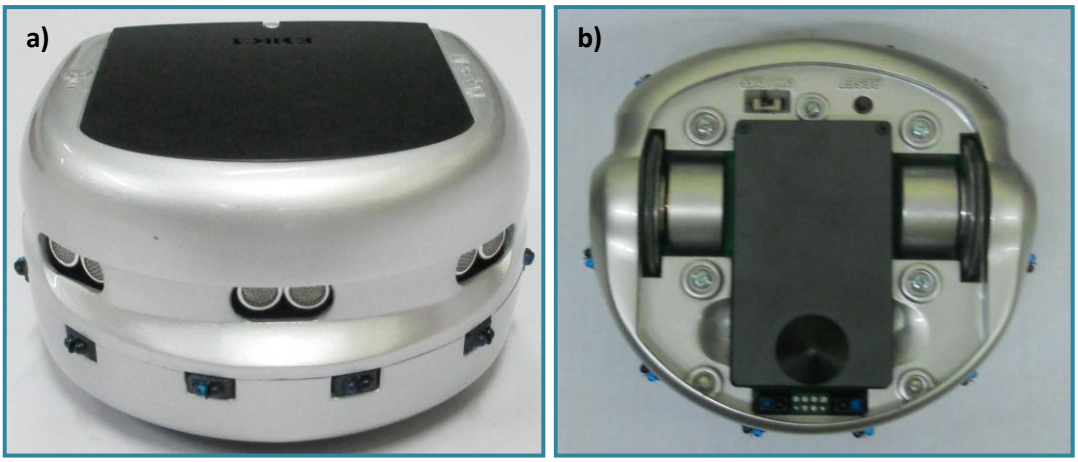


Figure A.3 (a-b) Khepera III Robot with front and bottom view.

Table A.3 Technical specification of Khepera III Mobile robot

	Elements	Technical specification
1	Processor	DsPIC 30F5011 at 60MHz
2	RAM	4 KB on DsPIC
3	Flash	66 KB on DsPIC
4	Motors	2-DC brushed Servo motors with incremental encoders
5	Sensors	8 Infra-red proximity and ambient light sensors with up to 30cm range, 2 Infra-red ground proximity sensors for line following applications, 5 Ultrasonic sensors with range 20cm to 4 meters.
6	Speed	Max: 0.5m/s, Min:0.02m/s
7	Power	Power Adapater or Swapable Lithium-Polymer battery pack (1350 mAh)
8	Communication	Standard Serial Port, up to 115kbps USB communication
9	Size	Diameter: 130 mm, Height: 70 mm.
10	Weight	Approx. 690 g

11	Payload	Approx. 2000g
12	Remote control Software via tether or radio	LabVIEW ® (on PC, MAC or SUN) using RS232 MATLAB ® (on PC, MAC, Linux or SUN) using RS232 SysQuake ® (on PC, MAC, Linux or SUN) using RS232 Freeware Any other software capable of RS232 communication