INTELLIGENT CONTROL AND PATH PLANNING OF MULTIPLE MOBILE ROBOTS USING HYBRID AI TECHNIQUES

Thesis Submitted to the

Department of Mechanical Engineering

National Institute of Technology, Rourkela

For award of the degree

of

Doctor of Philosophy

by

Alok Kumar Jha

Under the Supervision of

Prof. Dayal R. Parhi

Department of Mechanical Engineering

National Institute of Technology Rourkela
Odisha (India)-769008

This Thesis is dedicated

To

My Father

Late Shri Krishna Kumar Jha

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.

(*Alok Kumar Jha*)

Date:

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

Certificate

*This is to certify that the thesis entitled, **"INTELLIGENT CONTROL AND PATH PLANNING OF MULTIPLE MOBILE ROBOTS USING HYBRID AI TECHNIQUES"**, being submitted by shri Alok Kumar Jha 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.

Supervisor

Date: / /

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

Acknowledgement

Setting an endeavor may not always be an easy task, obstacles are bound to come in this way & when this happens, help is welcome & needless to say without help of the those people whom I am going to address here, this endeavor could not have been successful & I owe my deep sense of gratitude & warm regards to my supervisor Dr. Dayal R Parhi, Professor, Mechanical Engineering Department, NIT Rourkela for his in depth supervision and guidance, constant encouragement and co-operative attitude for bringing out this thesis work.

I am thankful to Prof. S K Sarangi, Director, NIT Rourkela for giving me an opportunity to carry out my research work at NIT Rourkela.

I extend my sincere thanks to Dr. S. S. Mahapatra, Professor and Head of the Department, Mechanical Engineering Department, NIT Rourkela for his valuable suggestions and support.

I further cordially present my gratitude to Prof. S K Sahoo, chairman DSC and other DSC members for their indebted help and valuable suggestion for accomplishment of my dissertation work. Special thanks to the Department of Mechanical Engineering, NIT Rourkela for giving me the opportunity to carry out my doctoral research.

PhD students often talk about loneliness during the course of their study but this is something which I never experienced at NIT Rourkela and particularly in Robotics Lab. A heartfelt thanks to all my mates who were really supportive and dynamic to make the experience feel something special.

I am also indebted to my friends not only for all their useful suggestions but also for being there to listen when I needed an ear.

I would like to thank my parents & elders without their blessings & moral enrichment I could not have landed with this outcome. Finally, I would like to acknowledge my life partner for her understanding and love.

Bio-Data of the Candidate

Name of the Candidate	:	Alok Kumar Jha
Father's Name	:	K K Jha
Permanent Address	:	Plot No. 11/3, Deen Dayal Puram, New Khursipar, Bhilai, Distt.- Durg, 490011, Chhattisgarh.
Email	:	alokjha.30@gmail.com

Academic Qualification

Examination	**Board/University**	**Year of passing**	**Division**	**Subject**
Ph.D.	Pursuing at NIT Rourkela	-	-	Mechanical Engineering
M.Tech	N.I.T, Rourkela	2009	1st	Mechanical Engineering
B.E.	P R S S U Raipur	2007	1st	Mechanical Engineering
HSSC	CG State Board	2002	1st	Science
HSC	MP State Board	2000	1st	All

Abstract

This work reports the problem of intelligent control and path planning of multiple mobile robots. Soft computing methods, based on three main approaches i.e. 1) Bacterial Foraging Optimization Algorithm, 2) Radial Basis Function Network and 3) Bees Algorithm are presented. Initially, Bacterial foraging Optimization Algorithm (BFOA) with constant step size is analyzed for the navigation of mobile robots. Then the step size has been made adaptive to develop an Adaptive Bacterial Foraging Optimization (ABFO) controller. Further, another controller using radial basis function neural network has been developed for the mobile robot navigation. Number of training patterns are intended to train the RBFN controller for different conditions arises during the navigation. Moreover, Bees Algorithm has been used for the path planning of the mobile robots in unknown environments. A new fitness function has been used to perform the essential navigational tasks effectively and efficiently.

In addition to the selected standalone approaches, hybrid models are also proposed to improve the ability of independent navigation. Five hybrid models have been presented and analyzed for navigation of one, two and four mobile robots in various scenarios. Comparisons have been made for the distance travelled and time taken by the robots in simulation and real time. Further, all the proposed approaches are found capable of solving the basic issues of path planning for mobile robots while doing navigation. The controllers have been designed, developed and analyzed for various situations analogous to possible applications of the robots in indoor environments. Computer simulations are presented for all cases with single and multiple mobile robots in different environments to show the effectiveness of the proposed controllers. Furthermore, various exercises have been performed, analyzed and compared in physical environments to exhibit the effectiveness of the developed controllers.

Keywords: mobile robot, AI, path planning, mobile robot navigation.

Contents

List of Figures

List of Tables

Nomenclature

ABFO	Adaptive Bacterial Foraging Optimization Algorithm
ACO	Ant Colony Algorithm
ANN	Artificial Neural Network
BA	Bees Algorithm
BFOA	Bacterial Foraging Optimization Algorithm
E	Experimental
FISA	First Intermediate Steering Angle
FOD	Front Obstacle Distance
FSA	Final Steering Angle
GA	Genetic Algorithm
GPS	Global Positioning System
IR	Infra-Red
LOD	Left Obstacle Distance
OA	Obstacle Avoidance
RBF	Radial Basis Function
RBFN	Radial Basis Function Network
RFID	Radio Frequency Identification Technology
ROD	Right Obstacle Distance
S	Simulation
SA	Steering Angle
SISA	Second Intermediate Steering Angle
TA	Target Angle
WMR	Wheeled Mobile Robots

Chapter 1

INTRODUCTION

The present research work deals with the intelligent control and path planning of mobile robots. This chapter has been divided into three sections. The first part of the chapter discusses about the background and motivation behind this research. In the second part, objectives and scope of the work carried out have been discussed. The structure of the thesis has been explained at the end of this chapter.

1.1 Background and Motivation

In a physical environment, mobile robots can travel in their surroundings to accomplish a range of tasks. Due to their greater ability of traversing, mobile robots find their application in a number of places like hospitals, industries, museum, military, automobile sector, railway station, and even in the homes for entertainment and surveillance.

Robot navigation is a fundamental problem in robotics. Navigation related to mobile robot is the ability of getting a free path from its source point to the target point while avoiding the obstacles. Moreover the selected path should be optimized (i.e. having smallest possible distance and minimum number of turns) to make sure that the least amount of energy and time are used by the robot while in motion from start point to its target.

Mobile robots use different kind of mechanism for locomotion, including wheels, legs and tracks. On account of their efficiency and adaptability mobile robots equipped with wheels, are getting to be progressively imperative in the industry as a method of transport, inspection, and operation. In addition, mobile robots are useful for intervention in unfriendly environments for performing tasks such as landmine detection, to take care of nuclear waste, cleaning of nuclear reactors, surveillance, etc. Furthermore, mobile robots can serve as a test platform for conducting a range of experiments for diversified applications. The work presented in this thesis describes the design and development of intelligent controllers for path planning and navigation of multiple mobile robots in a known and unknown environment consisting of various types of obstacles.

The navigation approaches of mobile robots can be classified as global path planning and local path planning. In case of global path planning the robot has complete prior information about the shape, location, orientation, and even the arrangements of the obstacles in the environment. It uses some optimization algorithms to minimize the cost function and can work in static environment only. On the other hand, in local path planning, robot uses some reactive approaches such as artificial neural network to recognize the environment based on the data collected from the different sensors and has the ability to plan a path online.

1.2 Aims and Objectives of the Research

The prime objective of this research work is to design and develop artificial intelligence techniques for intelligent control and path planning of multiple mobile robots. In particular, the research will seek to design and analyse artificial intelligence techniques such as Radial Basis Function Network (RBFN), Bacterial Foraging Optimization Algorithm (BFOA), Bees Algorithm (BA) and hybrid techniques in order to offer answers to the fundamental problem of mobile robot navigation.

Autonomous robot should have the following skills and criteria to navigate in the real world:

1. The mobile robot should be able to perceive the environment with the help of sensory data collected from various kinds of sensors mounted on it.
2. The robot should be capable of moving from one position to another position without assistance.
3. It is desirable that the robot should not cause any harm to the environment, people or itself. Also the robot should avoid those situations by offering proper response. Moreover, the robot should be able to negotiate with the obstacles without any outside assistance.
4. The robot should have the learning ability in order to gain the information about the environment with experience and hence update itself for future search.
5. It should adapt to the situation in accordance with the common sense reasoning provided in the artificial intelligence techniques embedded in its controller.

To meet the requirement discussed above, following behavior are essential during navigation of mobile robots.

1. Goal Seeking Behavior: If there is no obstacle present around the robot then the robot can search for target and move quickly towards it.
2. Obstacle Avoidance Behavior: Whenever a robot comes across any obstacle, then the robot has to change its direction in order to avoid any collision with the obstacles.
3. Wall Following Behavior: Mobile robot may get trapped in loop while searching for target if it is surrounded by obstacles. The robot should come out of this situation by maintaining a safe distance from the wall.

Moreover, it is highly desirable that the robot should follow the optimal path while travelling from start position to target position. Fuzzy logic and neural network plays vital role in designing control systems for mobile robots. Fuzzy logic uses the remarkable reasoning capability like a human being for decision making. This reasoning rule set can be incorporated into the system with the help of fuzzy logic. Neural network is used to train the system for input output mapping required during the navigation.

A number of examples can be found in nature which inspires researchers to mimic those behavior in order to solve real world problems. Bacterial foraging optimization, Ant colony optimization and Bees algorithm are some of the nature inspired algorithms those can be used to optimize the path of the mobile robot. As a result, many researchers and scientists are now trying to incorporate different techniques to develop new hybrid techniques.

This thesis also proposes some novel hybrid approaches which can be used in the design of an intelligent controller for path planning of mobile robots. Moreover the proposed approaches have been tested in simulated environment, with single as well as multiple robots. A series of experimental results have been presented to validate the effectiveness of the proposed approaches. The objectives of the present research work are given below:

1. To develop an intelligent controller based on bacterial foraging optimization algorithm (BFOA).
2. To develop a Radial basis function network (RBFN) based controller for path planning of mobile robots.
3. To develop an intelligent controller based on bees algorithm (BA).
4. To develop hybrid techniques for the intelligent control and path planning of multiple mobile robots.
5. To validate the effectiveness of the proposed artificial intelligent techniques developed in the current research work in simulated and real environments.

1.3 Novelty of the Thesis

The present work proposes novel hybrid approaches for intelligent control and path planning of mobile robots. In the current research, three soft computing methods namely 1) Bacterial Foraging Optimization Algorithm, 2) Radial Basis Function Neural Network and 3) Bees Algorithm has been hybridized to design a robust intelligent controller for the autonomous navigation of mobile robots in known and unknown environments. Further the developed approaches are found capable of solving the basic issues of path planning for mobile robots while performing navigational task.

1.4 Outline of the Thesis

The thesis is divided into following sections:

Chapter 1 provides an introduction to mobile robot navigation and discusses about the motivation behind current research work. The objectives of the present work have been given at the end of the chapter.

Chapter 2 presents the literature review of various approaches used in the field of mobile robot navigation.

Chapter 3 provides kinematic modeling and analysis of wheeled mobile robots. A model has been presented to describe the motion of the robot equipped with different types of wheels which can be successfully applied to all the basic wheel configuration.

Chapter 4 concerns with the path planning of the mobile robots using bacterial foraging optimization algorithm. The proposed approach is inspired from the foraging behavior of bacterium. An adaptive bacterial foraging approach has been implemented to get the optimal path for a mobile robot from start point to the goal position. Simulation Results show the effectiveness of the proposed approach in different environments. Experimental results are given to validate the simulation results.

Chapter 5 discusses about the use of RBFN approach for the mobile robot path planning. Various results have been presented in a simulated environment for single and multiple robots. Experiments have been conducted in order to validate the simulated results for the proposed approach.

Chapter 6 presents the analysis of a proposed approach in the context of motion planning and control of mobile robots using Bees algorithm. Bees algorithm is also a nature inspired algorithm which mimics the behavior of honey bees. The proposed approach has been adopted for path planning by appropriately designing the fitness function. The proposed approach has been tested in both simulated and in a real environment.

Chapter 7 deals with the navigation of mobile robots using hybrid approaches. The approaches already discussed in the previous chapters are combined to develop the hybrid approaches. The hybrid controllers have been designed and implemented in the prototype robot and tested in various test environments. Simulation results are presented to show the ability of the proposed approaches in different test situations demonstrating navigation of multiple mobile robots from start to goal position.

Chapter 8 summarizes the conclusion drawn on the basis of simulation and experimental analysis of the proposed techniques developed in the current research. Moreover, scopes for future research have been suggested at the end of this chapter.

Chapter 2

LITERATURE REVIEW

In the current research, intelligent control and path planning of multiple mobile robots has been carried out. This chapter reviews the prior work related to the design and development of an intelligent controller for path planning of the mobile robot in various environmental situations including structured and cluttered one. The progress made in past decades in the field of navigation control and path planning has been discussed. This chapter provides a detailed survey report for path planning and intelligent control of mobile robots using classical as well as reactive approaches. At the end of the chapter the summary of the literature survey and the knowledge gap in the earlier investigations are presented.

2.1 Background

Navigation of mobile robots and their control technique is one of the forefront researches in the research community throughout the world. Mobile robots can be used in various industrial sectors which is a key for human advancements in terms of science and technology. Many Scientists, Researchers and Engineers, since last several years have done various work on navigational strategies of the mobile robots to find a suitable methodology for controlling the robots. In the current research undertaken, efforts have been made to find out intelligent control and path planning of multiple mobile robots using hybrid artificial intelligence techniques. The research already done, which are relevant for the present study are discussed below.

2.2 Different Approaches used for Path Planning and Navigation Control of Mobile Robots

Various approaches have been developed in the past decades to solve the navigation problem related to the mobile robot and still researchers are keen to develop the new techniques which would offer smooth and optimal collision free path during the navigation of the mobile robots. Different approaches used for the navigation control and path planning of the mobile robots are reviewed below.

2.2.1 Potential Field Method

For many years, it has been seen among the researchers that potential field method is one of the promising technique in controlling mobile robots. Andrews and Hogan [1] and Khatib [2] have suggested the scheme of imaginary forces acting on a robot in such a way that the robot is considered to be attracted by the target and repelled by the obstacles. As a result, resultant force governs the following direction and speed of travel.

2.2.1.1 Navigation using Potential Field Approach

Koren and Johann [3] have used potential field methods for navigation of the mobile robots considering the inherent limitations of the mobile robots. They have stated that the potential field method can be applied elegantly and in a simplified manner for the navigation of mobile robots. In their method they have combined the robot and the environment as a unified system. They have done a comparison between theoretical and experimental results for their proposed method. Barraquand et al. [4] have discussed about numerical potential field technique for path planning of the robots. They have claimed that their method is better in comparison to the global path planning method since a global path planner requires more computational time. They have used multi-scale pyramids of bitmap arrays for representing both robot workspace and configuration space. Their algorithm has solved a variety of path planning problems.

McFetridge and Ibrahim [5] have analyzed about hybrid potential field methods for navigation of mobile robots. Their approach is based on the artificial potential field (APF) method which is extensively used for obstacle avoidance. In their method, the variables are dependent upon both the angle and distance with respect to the robot. They have shown simulation results for their proposed technique. A new potential field method has been discussed by Wang and Gregory [6] for path planning of non-spherical single body robots. Their model simulates steady state heat transfer and with variable thermal conductivity. The optimal path in their case is same as the heat flow with the minimum thermal resistance. This optimal path has been used for the navigational path planning of the mobile robot.

Autonomous robot navigation of a mobile robot using adaptive potential field method has been discussed by Cosio and Ma [7]. They have presented a method for autonomous navigation of the mobile robot based on enhanced potential fields and a genetic algorithm. In their scheme, many supplementary attraction points have been used to allow the robot to avoid the big or closely spaced obstacles. In their method, the optimal potential field has been configured automatically using genetic algorithm.

Li and Ke-Zhong [8] have discussed about the angle field potential method for the navigation and obstacle avoidance of a mobile robot in an outdoor environment. In their method, two-dimensional obstacle information in the polar coordinate space has been converted to one dimensional angle field. They have tested their method using THMR-V mobile robot.

Hassanzadeh et al. [9] have used the potential field method and fuzzy controller approach for robot navigation. They have also done a comparison between the two methods. Huang [10] has done velocity planning of a mobile robot to follow a dynamic target using potential field method. In their method, potential field has been applied to path speed and velocity planning of the robot. They have computed relative velocities and relative positions between the robot, obstacles and targets to compute the path and velocity of the robot. They have presented simulation results to verify the effectiveness of their proposed approach.

Do [11], Takahashi et al. [12], and Jaradat et al. [13] have used potential functions for path planning of mobile robots subjected to various environmental situations. They have given various simulation results to show the effectiveness of their proposed methods.

2.2.2 Neural Network based Approaches

Many researchers have used artificial neural network (ANN) technique for navigation of the mobile robots and the approaches are discussed below.

2.2.2.1 Navigation based on Neural Network

Meng and Kak [14] have done navigation of a mobile robot using neural network method. They have developed a system called NEURO-NAV, which uses non metrical representation to show various landmarks in their environment. In their method, the individual neural network has to interpret the visual information and perform primitive navigational tasks.

Predictive neural control of a car type mobile robot has been discussed by Gu and Huosheng [15]. They have used a back propagation algorithm to model nonlinear kinematics of the robot instead of a linear regression estimator in order to cope up with different environmental situations. They have shown simulation results for modeling and control to justify their propose methodology. Oh et al. [16] have tried to control a non-holonomic robot using radial basis function (RBF) network. They have presented several simulation results for their proposed method. Moreover, they have claimed that their robot can be used in real time applications.

Mesbahi et al. [17] have discussed about reactive navigation of the mobile robot using temporal radial basis function approach to develop a reactive navigation model based on neural networks. They have used structured type environment and obstacles for robot navigation. Harb et al. [18] have used neural network approach for environmental recognition and mapping during the robot navigation. They have claimed that the mobile robot can play a significant role in hazardous places where human kind cannot reach. They have used neural network against the traditional methods to locally navigate their robot in the environment. They have used multilayer neural network to process distance measurement received from the sensors.

Engedy and Horvath [19] have described about a dynamically artificial neural network for navigation and path planning of the mobile robot. They have proposed that their robot can navigate in a flat surface from starting point to end point. They have used the potential field for obstacle avoidance.

Wahab [20] has used the dual artificial neural network for navigation of the mobile robot. Their first neural network is used to determine free space needed to avoid obstacles and their second neural network is used to find the target. They have done a simulation for their proposed technique. Markoski et al. [21] have used self-learning neural network for mobile robot navigation. Their robot consists of wheel drives, encoders and short range sensors which are integrated to their proposed algorithm. They have done experiments to show the effectiveness of their proposed method.

Fernandez-Leon et al. [22] have studied the scaling of behavior in evolutive robotics. They have used an evolutionary perspective to solve the navigational task of the robot in different environmental scenarios. Rossomando and Carelli [23] have discussed about radial basis function neural compensator for autonomous mobile robot navigation. They have used kinematic controller inverse dynamic controller for their proposed method.

They have claimed that their adaptive controller is efficient enough in terms of tracking. They have shown the validity of the results through various navigational exercises.

Mohareri et al. [24] have discussed about indirect adaptive tracking control of a non-holonomic mobile robot using neural network. The performance of their controller has been verified using simulation results. Ghommam et al. [25] have discussed about the velocity measurement of the leader robot using formation control. They have proposed a reference trajectory which the leader will generate and the follower robots will track the trajectory.

Praczyk [26] have discussed about an anti-collision system of surface vehicle using neural network. Their paper describes the architecture of the neural autonomous surface vehicle for controlling the vehicles.

2.2.3 Fuzzy Logic based Approaches

Fuzzy logic is inspired from the exceptional capacity of human being to make appropriate decision on the basis of perception based information. Fuzzy Logic is a tool for modelling uncertain systems by enabling common sense reasoning in decision-making in the lack of thorough and accurate information. It enables the arrival of a definite conclusion based on input information, which is unclear, uncertain, noisy and imprecise. Many researchers have used fuzzy logic for navigation of mobile robots. This section provides a review of prior work reported related to mobile robot navigation in the past decades.

2.2.3.1 Navigation using Fuzzy Logic based Approaches

Aguirre and Antonio [27, 28] have discussed about navigation, coordination and fusion of a mobile robot. They have stated that robot behavior can be implemented as a set of fuzzy rule which expert knowledge as per the environment. In their work they have presented the architecture on the design, coordination and a fusion of robotic behavior. They have stated that the robot achieves the control objective using their proposed fuzzy controller. Xu [29] has discussed about a virtual target approach for resolving limit cycle problem in context of fuzzy behavior based mobile robot. The robot navigates according to the virtual target setup and real environmental information depending upon the sensor data. They have verified the effectiveness of the proposed method through simulation and experiment. Kang et al. [30] have discussed about multi objective navigation of a guide mobile robot in an environment. They have proposed a fuzzy grid base type local map for navigation of mobile robots. They have given some experimental results to show the effectiveness of the proposed method.

Nanayakkara et al. [31] have analyzed an evolutionary learning of fuzzy behavior based controller for a non-holonomic mobile robot in an environment. Their multi-objective evaluation function is designed so that it can incorporate complex linguistic features that a human behavior can be implemented in the mobile robot movements. They have given simulation results for their proposed technique. Maaref and Claode [32] have used fuzzy logic based sensor oriented navigation for mobile robots. They have extracted the fuzzy rules from different experimental data. Which in turn are used for building a fuzzy controller for mobile robot navigation. Yavuz and Bradshaw [33] have used a new conceptual approach to design a hybrid controller for an autonomous mobile robot. Their new approach is a hybrid system which incorporates various architecture types using the fuzzy logic approach. In their analysis a comparison of rarely control architectures is presented in respect to mobile robot navigation.

Cuesta et al. [34] have focused on intelligent control of non-holonomic mobile robots using the fuzzy perception technique. They have discussed about the tele-operation and planned behavior for control of the mobile robot. Khatib and Jean [35] have used a data driven fuzzy approach for solving motion planning problem of a mobile robot among various types of obstacles. In their method, they have proposed that their method can be used for the navigation of mobile robots from a start position to a target point efficiently.

Lee and Chia-Ju [36] have proposed a fuzzy algorithm for mobile robot navigation in an unknown environment filled with obstacles. Their mobile robot is equipped with electronic compass, two optical encoders for dead reckoning and ultrasonic sensor module for self-localization. They have stated that the navigation procedure will be active till a collision free path is obtained. They have shown their results in the simulation mode.

Abdessemed et al. [37] have presented a fuzzy based reactive controller for non-holonomic mobile robot. They have used if then rules for solving the navigational problem. They have claimed that their fuzzy controller can be used for navigation purpose in simulation mode. Demirily and Mohammad [38] have discussed about a dynamic localization based fuzzy method for mobile robot control. They have a used matching process using the fuzzy pattern matching technique for the navigation of the mobile robot. They have implemented their method in Nomad robot.

Parhi [39] has used fuzzy logic for navigation of mobile robots. In his paper, he has stated that multi-robot can be controlled using the fuzzy logic method.

Mendez and Juan [40] have used unknown environments for navigation control for mobile robot using the fuzzy adaptive technique. They have tested a pioneer mobile robot to detect obstacles and odometric sensor for localization of the robot and goal position. They have used weighting scheme for learning algorithm in their proposed fuzzy technique. Innocenti et al. [41] have discussed about multi agent architecture for controlling a mobile robot using fuzzy logic. They have shown different results using a Pioneer robot to show the effectiveness of their proposed method.

Wang and James [42] have analyzed navigation of the real time robot using fuzzy logic in an unknown environment with dead ends. They have used grid technique which addresses the local minima problem during the goal oriented task by the robot. They have used fuzzy logic for behavior design and coordination of the robot. They have tested their results in simulation mode. Huq et al. [43] have presented a technique for robot navigation using motor schema and fuzzy context dependent behavior modulation. They have used Voronoi diagram method to find out the safe path. He et al. [44] have applied fuzzy neural based model for mobile robot obstacle avoidance task. They have carried out different exercises to validate their proposed technique.

Gueaieb and Miah [45] have presented a mobile robot navigation technique using radio frequency identification technology (RFID). They have stated that RFID signals can be used effectively for navigation purpose. Martinez et al. [46] have discussed about a tracking controller for the dynamic model of a unicycle mobile robot using fuzzy logic and genetic algorithm. They have given computer simulation for their proposed method.

Chen et al. [47] have proposed a complete control law comprising of evolutionary programming based on kinematic control and adaptive fuzzy sliding mode dynamic control for path planning of the mobile robot. They have given the computer simulations to confirm the effectiveness of their proposed result.

Motlagh et al. [48] have discussed about a fuzzy algorithm for mobile robot navigation in local environments. They have given robot trajectories by simulation work and compared their results with other methods to show the effectiveness of their proposed method.

Doitsidis et al. [49] have used fuzzy controllers for robotic vehicles. They have used various sensor data for finding out the robot distance from the target. The paper by Nurmaini et al. [50] deals with Type-2 fuzzy neural controller for navigational decision of the mobile robot. They have shown the functionality of the robot using 8 bit micro-controller with 512 bytes of RAM. Norouzi et al. [51] have stated that sensor fusion is quite essential for navigation of mobile robots. They have used the fuzzy logic controller to handle the sensor information to be used for navigational purpose.

Abiyev et al. [52] have presented a software simulation of navigation problem of a mobile robot avoiding obstacles using classical and fuzzy based algorithms. They have shown a simulation results for their robot navigation. Perez et al. [53] have discussed about fuzzy uncertainty modeling for a grid based localization of mobile robots. Their method follows a typical predict-update cycle of recursive state estimators to estimate the robot's location. They have used the fuzzy grid based system to reduce the computational complexity. Tzafestas et al. [54] have presented a system on a chip for the path planning task of autonomous non-holonomic mobile robots using fuzzy logic. The system on chip consists of a digital fuzzy logic controller and a flow control algorithm used for robot navigation. Hashemi et al. [55] have been discussed about PI-fuzzy path planner for control of the omnidirectional mobile robots. They have shown different exercises to show the effectiveness of their proposed method.

The paper by Melin et al. [56] addresses the tracking problem for dynamic model of a unicycle mobile robot. They have solved the robot motion problem by integrating a kinematic and torque controller, based on fuzzy logic theory. They have presented simulation results to validate their approach. The paper by Resende et al. [57] discusses about nonlinear trajectory tracking of mobile robots using fuzzy logic technique. They have given various exercises to validate the proposed methodology.

Kala et al. [58] have discussed about navigation of mobile robots using fuzzy logic. They have claimed that the robots can negotiate with the obstacles and reach the target using fuzzy logic.

2.2.4 GA based Approaches

Genetic algorithm is a type of evolutionary algorithm based on the principle of ‘natural selection’ and ‘survival of the fittest’. This has been widely used in different fields of science and research by different researchers. GA is a search based algorithm which tries to find the fittest individual by the evolution of the initial population. The robustness of GA lies in the ability to handle nonlinear and complex multi objective problems. The major drawbacks of the algorithm are fast convergence and requirement of a large number of evaluations to reach at a global solution. The following section addresses some of the literature analysis on genetic algorithm. The scientific community has tried to control mobile robots using genetic algorithm.

2.2.4.1 Navigation using Genetic Algorithm based Approaches

Han et al. [59] have discussed about path planning and obstacle avoidance of mobile robots using GA. They have used a fitness function which will be implemented in a genetic algorithm to evaluate a path from start position to a goal position by avoiding the obstacles.

Analysis of perception based GA has been studied by Kubota et al. [60]. They have used perceiving information to map the dynamic environment so that the robot can avoid the obstacles during navigation. They have shown various simulation results for validating their method. Moreno et al. [61] have used ultrasonic sensors and genetic algorithm localization method for navigation of the mobile robot. They have used an iterated non-linear filter which can recognize the geometric beacons and can map the environment.

Path planning of a mobile robot using genetic algorithm has been discussed by Tu and Yang [62]. They have used a grid based mapping for the robot 2D workspace. They have stated that their algorithm is capable of generating a collision free path for both static and dynamic environment during navigation. They have shown the effectiveness of their algorithm through simulation mode.

Keiji and Ishikawa [63] have discussed about optimal values of parameters required in reinforcement learning using genetic algorithm for a mobile robot.

Sedighi et al. [64] have presented a genetic algorithm based path planning for local obstacle avoidance by a mobile robot. They have tried to minimize the robot path using genetic algorithm. They have claimed that their genetic algorithm controller can handle complicated robotic environments. Mucientes et al. [65] have designed a fuzzy based genetic algorithm controller for a mobile robot. Using their method, they have tried to do the wall following behavior of a mobile robot. Yamada [66] has described about the recognition of the environment by the mobile robot using genetic algorithm. In his task the robot perceives the environment as different structures.

Yang et al. [67] have discussed about a dynamic approach using genetic algorithm for navigation of mobile robots. They have simulated the navigation of a mobile robot in a computer. They have tested the feasibility of their developed genetic algorithm for the navigation of mobile robots through various exercises. Samsudin et al. [68] have discussed about a tailored genetic algorithm approach to optimize fuzzy controller and to implement in the mobile robot navigates. They have claimed that using the genetic fuzzy controller the mobile robot can negotiate with the obstacles and can reach the target.

Tuncer and Mehmet [69] have discussed about genetic algorithm technique for dynamic path planning of mobile robots. They have applied their method to different dynamic environments and compare the results with others literatures. They have stated that their developed genetic algorithm can be effectively used for the navigation of mobile robots.

2.2.5 Nature Inspired Approaches

Various researchers have been used nature inspired methods for successful navigation of mobile robots.

2.2.5.1 Navigation using Nature Inspired Approaches

Guan-Zheng et al. [70] have discussed about a globally optimal path planning using the ant colony algorithm for navigation of mobile robots. They have stated that their proposed method is effective and can be used for real time navigation of mobile robots. Tsankova et al. [71] have discussed about immune network control systems for navigation of autonomous mobile robots. They have used stigmergic principles in the artificial immune networks for the avoidance of collision with the obstacles and for object picking and dropping behavior of the mobile robot. They have given several simulation results to authenticate their proposed method.

Garcia et al. [72] have discussed about ant colony optimization technique for navigation of the mobile robot. They have used fuzzy cost function along with the ant colony optimization to build their mobile robot controller. They have tested their results in various virtual environments to show the effectiveness of their proposed method. Cong and Ponnambalam [73] have discussed about an ant colony optimization method for path planning of a mobile robot. Their environment is consisting of static obstacles and walls in different arrangement. In their method, the mobile robot tries to negotiate with the static obstacles as well the walls using ant colony optimization technique. They have performed various exercises to prove the effectiveness of the ant colony optimization technique for the navigation of the mobile robot.

Luh and liu [74] have discussed about an immunological approach for reactive navigation of a mobile robot. They have used an adaptive virtual target method to solve the local minima problem arises during robot navigation. They have given various simulation exercises to show the effectiveness of their proposed technique.

Venayagamoorthy et al. [75] have discussed about swarm intelligence technique for navigation of mobile robots. They have tested their results against the greedy search algorithm to show the effectiveness of their proposed approach.

Chia et al. [76] have discussed about ant colony optimization technique for the navigation of mobile robots. They have shown different simulated results executing various optimized motion path for mobile robot from source position to target position by avoiding the obstacles. Chen et al. [77] have discussed about swarm optimization and wall following by the robot using swarm intelligence technique. They have used social information to train the swarms during navigation. They have given experimental results to validate the theory developed.

The Flock dynamics approach has been discussed by Espelosin et al. [78] for path planning of mobile robot. They have used a heuristic method that is a dynamic particle chain method to join the position of the robot with the target through consecutive connection among the particles of flock during robot navigation. They have compared their results with the other algorithm to show the effectiveness of their proposed method.

Mo and Xu [79] have discussed about a particle swarm optimization method for path planning of a mobile robot. They have Voronoi boundary network in a static environment, which is useful during navigation. They have shown simulation results to exhibit the efficiency of their proposed technique. Castillio et al. [80] have been discussed about fuzzy logic combined with ant colony optimization technique for the navigation of a mobile robot. They have presented various results for unicycle mobile robot to show the ability of their proposed technique for mobile robot navigation.

2.2.6 Sensor based Navigation Approaches

For years, many researchers have analyzed on control techniques of mobile robots equipped with various types of sensors.

2.2.6.1 Navigation using Sensor based Approaches

Zelinsky [81] has discussed about the mobile robot map making using sonar sensors. His robot is capable of producing high resolution maps of an outdoor environments equipped with sonar range finding sensors. His robot can recognize the obstacle distances using sonar sensors. He has given various exercises of his mobile robot equipped with sonar sensors.

Varadarajan et al. [82] have developed a technique to fuse the sonar and wheel encoder information to produce a map of an environment. In their methodology, they have used specular environment which causes the sonar to produce completely unusable readings which often occurs with smooth surfaces. Masek et al. [83] have discussed about obstacle sensing using mobile robot equipped with a ring of sonar. In their method of sonar ring design, multi-wave superposition has been taken care to attain a quasi-homogeneous sonar beam intensity pattern. They have claimed that using their sonar sensor the robot can accurately detect the obstacles during navigation.

Maris [84] has discussed about attention based mobile robot using a reconfigurable sensor. The mobile robot designed by Harper and Phillip [85] can navigate through many environments that contain plants. They have used a sensor that can recognize plants which will be very useful for navigation in those environments. They have used an ultrasonic sensor to distinguish various plants according to the structure of the plants.

Asharif et al. [86] have discussed about sensory data integration to obtain the grid map required for robot navigation. They have used neural network along with sensory data for path planning of the mobile robot.

Benet et al. [87] have used infrared (IR) sensors in their robots for obstacle avoidance. They have claimed that IR sensor can accurately measure distances with reduced response time. They have given experimental results to validate the theory developed. A sensing system using sonar ring has been used by Ming et al. [88] for obstacle detection and avoidance. To eliminate the error from the sonar ring they have used a neural filtering method to prevent cross talks between the sensors.

Carelli et al. [89] have discussed about control, navigation and wall following behavior of a mobile robot using sonar and odometric sensorial information. They have added the OA capability to the control system as a perturbation signal. The paper by Carmena and John [90] discusses about biologically inspired engineering with the use of narrowband sonar in mobile robot navigation. In their method, they have used Doppler-shifting compensation successfully for navigation of mobile robots. Lin et al. [91] has discussed about topological navigation using ID Tag and WEB camera. They have discussed about node ID and post adjustment method for getting the direction angle and heading angle of the robot. They have stated that their method can be feasible for navigation of inter mobile robots.

Batalin et al. [92] have discussed about mobile robot navigation using sonar network. They have computed the direction of navigation within the sonar network using value iteration. They have done experimental verification to validate their proposed technique. Lee et al. [93] have discussed about a practical algorithm for topological navigation in corridor mapping of the mobile robot using sonar sensor. They have used circle following algorithm to handle obstacle avoidance situations during navigation.

Kim and Nak [94] have used the radio-frequency identification (RFID) system for navigation of mobile system. They have demonstrated that using the proposed system the robot can approach to a stationary target object. They have done various exercises to show the effectiveness of their proposed approach. Trajectory linearization based controller has been used by Liu et al. [95] to control the omnidirectional mobile robot. They have used sensor fusion method to combine onboard sensor and vision data system to provide accurate and reliable robot position and orientation during the navigation.

Indoor navigation of a wheeled mobile robot has been discussed by Popa et al. [96]. The robot uses video images for navigational purpose. An odometric based OA algorithm has been implemented by them for navigation of the mobile robot. Ray et al. [97] have presented an integration of GPS and sonar based technique for the navigation of mobile robots. They have stated that, this mapping enables the robot to navigate among different GPS locations to get the longitude and latitude data for a particular location. They have implemented their technique in Pioneer robot for experimental verification.

Morkovic and Ivan [98] have studied on speaker localization and tracking with a microphone array on a mobile robot using von Mises distribution and particle filtering. They have used two most common microphone arrays to conduct the experiments in order to test their algorithms quantitatively and qualitatively.

Quintia et al. [99] have proposed a learning approach for mobile robot using reinforcement based strategy and a dynamic sensory data state mapping. They have claimed that a dynamic creation of state representation will help in mobile robot in controlling. Teimoori and Andrey [100] have discussed about navigation of a mobile robot using various sensory data. They have given mathematical analysis for path planning of mobile robots. They have shown computer simulation and experimental results using Pioneer robots.

Sanada et al. [101] have discussed about self-localization of an omnidirectional mobile robot using optical flow sensor. In their experimental method, the accuracy of self-location by dead reckoning and optical flow methods are evaluated using the motion capture methodology. They have claimed that the correct position can be achieved by optical flow sensor method rather than by the dead reckoning method. Delgado-Mata et al. [102] have discussed about the navigation of ethology inspired mobile robots. They have used ultrasound sensors for the navigation of mobile robots.

Espinace et al. [103] have discussed about indoor navigation of mobile robots using the adaptive object detection method. They have used various sensors to map the environment required during robot navigation. Siagian et al. [104] have discussed about hierarchical map representation for mobile robot localization and navigation. They have used several visual perception module such as place recognition, landmark recognition, and road lane detection for mapping the environment required for the robot navigation.

Xu et al. [105] have discussed about sensor based robot navigation in complex environments. In their method, the motion detection of the robot is determined by using the key obstacle function segment sets. They have shown various simulation results to show the effectiveness of their proposed method. Pozna et al. [106] have discussed about pose estimation algorithm for the robot navigation. They have used various sensory data for navigation of the robot. They have shown the effectiveness of the proposed method using simulation results.

2.2.7 Vision based Approaches

Control of mobile robots using vision based localization has been discussed by Adorni et al. [107]. In their paper, they have discussed about three approaches for vision based self-localization. Barnes and Liu [108] have discussed about vision guided mobile robots using embodied categorization. They have stated that during the navigation, categorization helps robots to make decision more efficiently. Salinas et al. [109] have discussed about a genetic fuzzy technique for navigation of mobile robots. They have used the image sensor to map the environment during navigation. They have shown that the robot has the ability to detect rectangular doors and the robot can be used in real time applications. A robust visual tracking controller design has been discussed by Tsai and Kai-Tai [110] for a non-holonomic robot. Their robot is equipped with a tilt camera to map the environment during navigation. The work by Singh et al. [111] depicts representation and implementation of real life in the scheme of path planning of a mobile robot. They have given various exercises to demonstrate effectiveness of their proposed technique.

Ohnishi and Imiya [112] have discussed about an algorithm using the visual potential field method for the robot navigation. Their robot can dynamically select a local pathway from start position to a goal position in a robot workspace.

2.2.8 Landmark based Approaches

Wijk and Henrik [113] have presented a technique using natural point landmarks for indoor navigation of mobile robots. They have given various experimental results to authenticate their proposed methodology. An automatic landmark selection algorithm has been deployed in a mobile robot for navigation purpose by Marsland et al. [114]. They have been stated that their algorithm can map the environment more effectively and reliably.

Hayet et al. [115] have discussed about visual based navigation using various landmarks. They have given experimental verification for the navigation of an indoor mobile robot. Nunez et al. [116] have proposed a geometrical feature detection system using two dimensional laser finders for mobile robot navigation. They have divided the laser scan into line and curve segments for easy visualization of various landmarks during navigation, which is very much useful for structured and semi structured environments.

A localization method has been discussed by Loevsky and Shimshoni [117] for indoor robot navigation. They have tested their localization system on a mobile robot which can perform various types of computational tasks in parallel during navigation.

2.2.9 Kinematic Analysis of Mobile Robots

Various researchers have done the kinematic model analysis required during robot navigation.

2.2.9.1 Kinematic Analysis of Mobile Robots

Mular and Neuman [118] have discussed about kinematic modeling of mobile robots. They have used sensing characterization tree to delineate the robot motion producible by the wheel actuators using the wheel sensors. They have done the kinematic analysis of a mobile robot. Larson et al. [119] have used Kalman filter method to evaluate a dual decoder system for robot navigation. Moreover, they have been done an evaluation of kinematic and odometric approach required during robot navigation.

Moon et al. [120] have discussed about kinematic correction of differential drive mobile robots taking into account the various constraints of motor controllers during robot navigation. They have implemented their method on a wheeled mobile robot with two degrees of freedom. Li and Yugang [121] have discussed about kinematic and dynamic analysis of mobile robots. They have been used direct differential method to control their mobile robot.

Kang et al. [122] have discussed about the kinematic path tracking of mobile robots using iterative learning control. They have used iterative learning control technique for navigation of mobile robots. They have shown simulation results to show the effectiveness of the proposed method. Chang et al. [123] have discussed about six wheel mobile robots along with their kinematic analysis. They have used the Jacobian matrices for kinematic analysis. They have shown various simulation results to validate the motion of mobile robots. Duan et al. [124] have discussed about navigation of small mobile robots in multi locomotion modes. They have also discussed about kinematic modeling of mobile robots. Minguez et al. [125] have discussed about kinematic constraints arises during obstacle avoidance of mobile robot navigation. They have discussed about the kinematic constraints using mobile robots during the obstacle avoidance situation. Lee et al. [126] have discussed about kinematic analysis of omnidirectional mobile robots having caster wheels. They have studied closed form of the kinematics of omnidirectional mobile robot consisting of double wheel type active casters.

Wang et al. [127] have discussed about obstacle avoidance, self-localization of a mobile robot during navigation. They have also done the kinematic analysis of their proposed mobile robot. Gao et al. [128] have discussed about the rise and tracking control of mobile robots using the time delay method. They have shown the performance of control scheme by computer simulation with several predefined trajectories. Kinematic modeling of wheeled mobile robots has been discussed by Gracia and Tornero [129]. Their proposed kinematic control is applied to an industrial forklift in simulation mode.

Kinematic modeling of a mobile robot in rough terrain has been discussed by Jinxia et al. [130]. They have shown various simulation results in different terrain to show the effectiveness of their proposed method. Moosavian et al. [131] have discussed about hybrid serial-parallel mobile robot for navigation of a mobile robot. They have done the kinematic analysis of the mobile robot.

Kinematic modeling and analysis of skid-steered mobile robot have been done by Yi et al. [132] Skid-steer mobile robots are widely used because of their simple mechanism and high reliability. They have used the Kalman filter to take care of the kinematic constraints to enhance the accuracy during navigation. Kane et al. [133] have discussed about the control of mobile robot on arbitrary surfaces. They have done the kinematic analysis of the mobile robot movement on a two dimensional flat surface. They have tested their results in a medical surgical robot. Li et al. [134] have done the kinematic analysis of a transformable wheel-track robot using self-adaptive mobile mechanism. They have done the kinematic analysis of the movement of Amoeba-III mobile robot. They have given experimental results for validity of their mobile platform. On and Ahmet [135] have considered the kinematic constraints of a mobile robot during path planning. Using the kinematic constraints, the robot can negotiate with the sharp corners with the help of a smooth trajectory. They have conducted experiments to show the effectiveness of their proposed approach.

Yun et al. [136] have done the simulation analysis of a four wheel independent mobile robot. In their analysis, they have also done the kinematic analysis of four wheeled mobile robots. They have used nonlinear control method for tracking test of the mobile robot. Moosavian et al. [137] have done the kinematic analysis of spatial parallel robot. They have designed parallel mechanism to get the proper maneuverability of the mobile robot. They have studied the robot navigation numerically. They have discussed about the various results on hybrid serial mobile robot system.

Shammas and Daniel [138] have discussed about active angular swivel steering mechanism for mobile bases. During the kinematic analysis of the proposed swivel steering mechanism, they have explained about the best configuration of the robot which can be used to negotiate rough terrain during the navigation. Maulana et al. [139] have studied about the inverse kinematics of a two wheeled autonomous mobile robot. They have calculated the speed of the rear wheel, according to the kinematic analysis and in their robot the front wheels work as free wheels.

From the exhaustive literature survey following conclusion can be drawn:

- Various approaches have been developed in the past for the path planning and control of mobile robots. These approaches can be classified as global approaches and local approaches or offline path planning and online path planning.
- In recent years, paradigm has been shifted towards the development of reactive approaches which can be used for online path planning of the mobile robots for performing specific tasks.
- From intensive review on various approaches it have been found that though most of the methods have been employed for single mobile robot in simulated environment, comparatively less work has been reported on control of multiple mobile robots.
- Various techniques have been used for the control of the mobile robots including classical approaches. But very few works have been reported particularly using bacterial foraging optimization algorithm and bees algorithm. Moreover, use of hybrid techniques for control of mobile robots are rare.

2.3 Differences and Merits of Selected Approaches

In the present work, various controllers have been designed for the intelligent control and path planning of multiple mobile robots using bacterial foraging optimization algorithm, bees algorithm, and radial basis function neural network. Moreover, hybrid approaches have been proposed by hybridizing the developed controllers using the standalone techniques. In this section differences and merits of the selected approaches have been discussed.

(i) **Bacteria Foraging Optimization Algorithm:** There are several optimization algorithms for solving the mobile robot navigation problem. They are classified as follows:

Potential Field Method

Artificial Neural Networks

Fuzzy Logic Control

Genetic Algorithms and

Bacterial Foraging Optimization Algorithm

Among the above stated optimization algorithms, Bacterial Foraging Optimization Algorithm (BFOA) is proposed in this work to find the near optimal path for mobile robots from its initial position to goal position. This algorithm has merits and differences as follows:

Merits:

- Less computational problem,
- Global convergence, (i.e. convergence near global optima)
- Less computational time requirement and
- Can handle more number of objective functions when compared to the other evolutionary algorithms.

Differences:

- Handling and tuning of more number of parameters are required during optimization

Radial Basis Function Network: This method is becoming popular due to its diverse applications in function approximation, regularization theory, time series estimation and system control. As the name implies, this network makes use of radial functions to represent an input in terms of radial centers. However, this network is very attractive for intelligent control applications since the response of this network is linear in terms of weights. Thus, weight update rules for such networks within intelligent control paradigm become easy to derive. This algorithm also has merits and differences as follows.

Merits:

- It has simple structure with single hidden layer.
- It is easy and takes less time to train the network.
- It gives linear response in terms of weight.

Differences:

- To reach same level of accuracy as MLP, RBFN requires more nodes in the hidden layer and hence becomes complex.

(ii) Bees Algorithm:

In nature, honey bees have several complicated behaviors such as mating, breeding and foraging. These behaviors have been imitated for several honey bee based optimization algorithms. The following algorithms are motivated from foraging behavior of honey bees;

Bee System (BS),
Bee Colony Optimization (BCO),
Artificial Bee Colony (ABC) and
The Bees Algorithm (BA).

The Bees Algorithm is very alike to the ABC in the sense of having local search and global search processes. However there is a difference between both algorithms during the neighborhood search process. As mentioned above, ABC has a probabilistic approach during the neighborhood stage; however the Bees Algorithm does not use any probability approach, but instead uses fitness evaluation to drive the search. The BA has both local and global search capability utilizing exploitation and exploration strategies, respectively. The BA also has advantages and disadvantages compared to the other algorithms and are listed below.

Merits:

- The algorithm has local search and global search ability. It can be implemented with several optimization problems.
- It is easy to use and available for hybridization with other algorithms.

Differences:

- The algorithm starts with random initialization.
- The algorithm has several parameters which need to be tuned.

(iii) Hybrid Approaches:

The current analysis deals with the navigation of mobile robots using five different hybrid techniques. They are as follows: 1) ABFO-RBFN Controller based technique, 2) ABFO-Bees Algorithm based technique, 3) RBFN-BA based technique, 4) BA-RBFN based technique, and 5) RBFN-BA-ABFO based technique.

These algorithms have merits and differences as follows.

Merits: Hybrid approaches permits the incorporation of multiple features of different approaches in a single controller. Therefore, hybrid controllers can be very effective in complex and dynamic environment with multiple robots.

Differences: Their design and architecture are complicated and complex.

2.4 Summary

This chapter provides an intensive review on various approaches used for intelligent control and path planning of the mobile robots. The progress made in the past decades have been studied and discussed. In particular, potential field method, neural network method, fuzzy logic based approaches, genetic algorithm, sensor based approaches, and algorithm inspired from nature have been studied. The requirements for proposed approaches (i.e. Bacterial Foraging Optimization Algorithm, Radial Basis Function Neural Network, and Bees Algorithm) have been discussed. The next chapter discusses about the kinematic modeling and analysis of the mobile robots.

Chapter 3

LOCOMOTION, KINEMATIC MODELING AND ANALYSIS OF WHEELED MOBILE ROBOTS

Mobile robots need some kind of locomotion mechanism to travel boundless in its environment. However, since different options to move are available therefore selection of locomotion mechanism is an important aspect of robot design. Wheels are the most popular pick in the mobile robotics because of their high efficiency and simple mechanical execution. Selection of the type of wheel or combination of wheels, their arrangements and actuating system plays important role in the design of a mobile robot. These parameters decide the mobility, stability and controllability of the mobile robot. The assessment of these parameters is very much required for the kinematic analysis of mobile robot.

Kinematic analysis is the most fundamental study associated with the operation of mechanical system. In mobile robotics, kinematic analysis is related to the mechanical behaviour of the robot without considering the forces that affect the motion. While navigating, the robot has to move in a specific direction by following a trajectory, planned by the control system of the robot.

Therefore the accurate estimation of the current position of the robot throughout its travel is an area under discussion. Finding the exact location of the moving robot by direct way is still a demanding task. For the wheeled mobile robot, the study of all the constraints subjected to wheels on which the robot is placed is needed. In the present work, fixed standard wheels have been used in the design of mobile robots. Moreover differential drive mechanism has been used for the locomotion.

3.1 Representation of Position of Mobile Robot

Let us consider a global reference frame and a local or robot reference plane as OX_IY_I and OX_RY_R respectively as shown in figure 3.1. Also consider that ξ represents the robot posture in the global coordinate and is defined as;

$$\xi_I = [x, y, \theta]^T \tag{3.1}$$

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

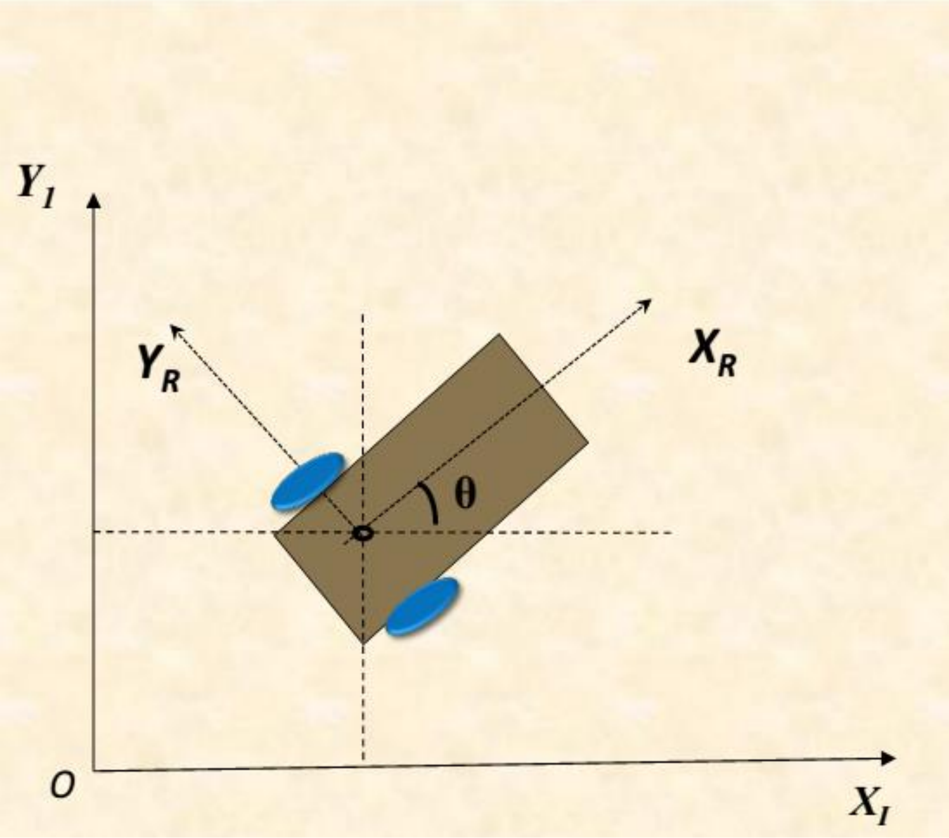


Figure 3.1 Global and Local Reference Frame

Above equations can be used to assess the robot's motion in the global reference frame from motion in its local reference frame by the operation;

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

3.2 Wheel Kinematic Constraints

To develop the kinematic model of the robot, it is required to express the various constraints of individual wheel on the motion of the robot. The motion of each individual wheel can be combined in order to calculate the motion of the robot as a whole. The following assumptions have been made in the analysis:

- The wheels are not deformable and there are point contacts between the wheels and the ground plane.
- Wheel's motions are pure rolling leading to a null velocity at the contact point.
- The wheels do not undergo slipping, skidding, sliding or friction for rotation around the contact point.
- The steering axes are orthogonal to a plane surface of the ground.
- The wheels are connected by rigid bodies to a rigid chassis of the robot.
- During the motion, the plane of each wheel remains vertical and the wheel rotates around its (horizontal) axle whose orientation with respect to the frame can be fixed or varying.

We have considered two constraints for each wheel type while the robot is in motion. The first constraint imposes the theory of pure rolling and the second one imposes the concept of no lateral slippage of the wheels in motion as shown in figure 3.2. Also based on geometric conditions we can classify the wheels into five basic types:

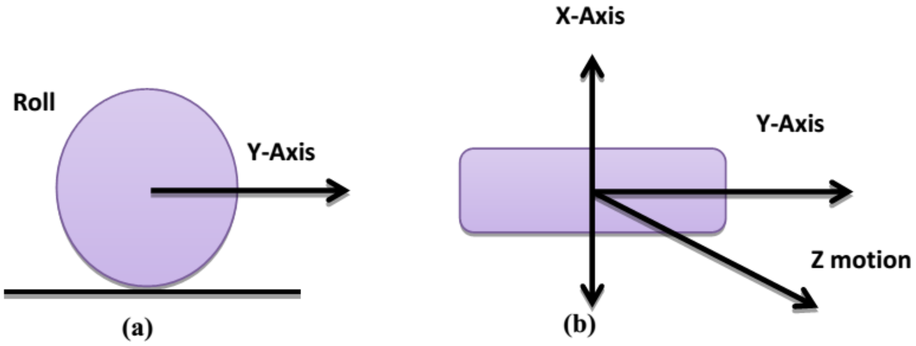


Figure 3.2 Wheel Kinematic Constraints (a) Pure Rolling and (b) Lateral Slip

1. Conventional or fixed standard wheel
2. Steered standard wheel
3. Castor wheel
4. Swedish wheel
5. Spherical wheel

3.3 Wheel Description

3.3.1 Fixed Standard Wheel

The conventional fixed wheel or fixed standard wheel as shown in figure 3.3 has no vertical axis of rotation, thus its angle to the robot chassis is fixed and therefore limited to have only backward and forward motion in the horizontal plane and rotation around its point of contact with the ground plane.

For pure rolling at the contact point;

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

The sliding constraint for this wheel enforces condition that the wheel’s motion normal to the wheel plane must be zero;

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

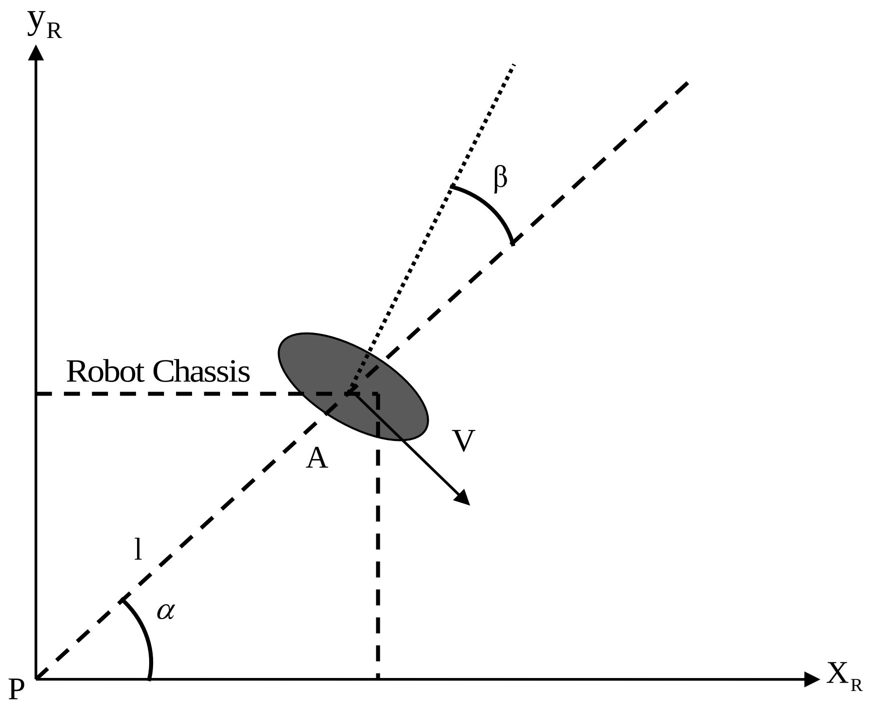


Figure 3.3 Fixed standard wheel and its parameters

3.3.2 Steered Standard Wheel

The steered standard wheel possesses an additional degree of freedom as compared to fixed standard wheel, i.e. The wheel may spin around a vertical axis passing through the wheel centre and the point of contact between the wheel and the ground. Thus the orientation of the wheel to the robot chassis now becomes variable as a function of time β (t) instead of a particular fixed value β as represented in figure 3.4. The rolling and sliding constraints are, along the wheel plane;

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

Orthogonal to the wheel plane;

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

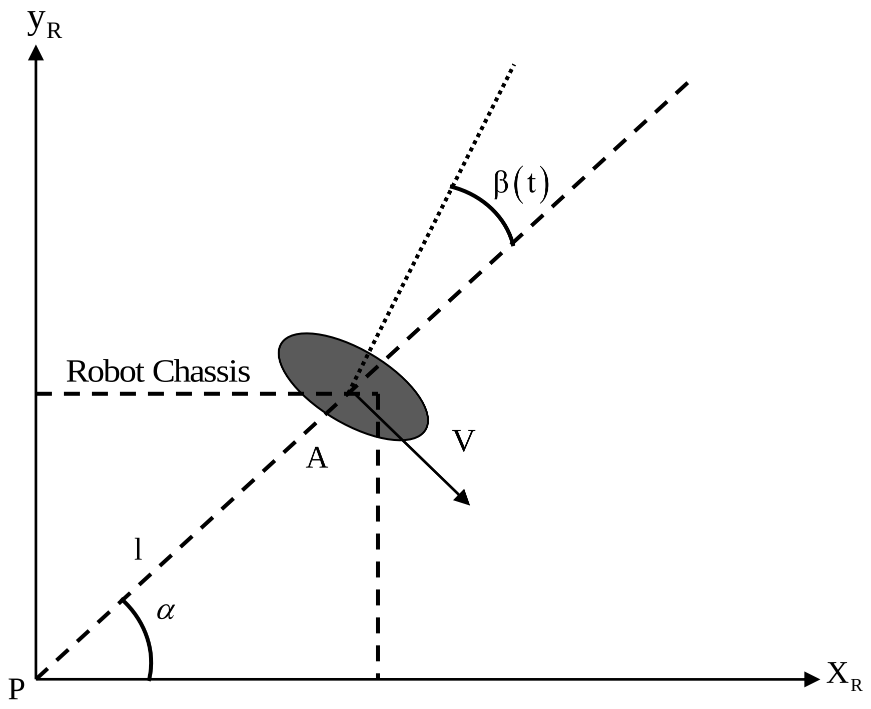


Figure 3.4 Steered standard wheel and its parameters

3.3.3 Caster Wheel

Similar to steered standard wheel, the caster wheel can also rotate around its vertical axis. However, in case of caster wheel the vertical axis of rotation is offset to the centre of the wheel and does not pass through the ground contact point.

Thus to specify the position of the caster wheel, one additional parameter (fixed length of rigid rod 'd') linked to wheel, as shown in figure 3.5.

Along the wheel plane;

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

Orthogonal to the wheel plane;

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

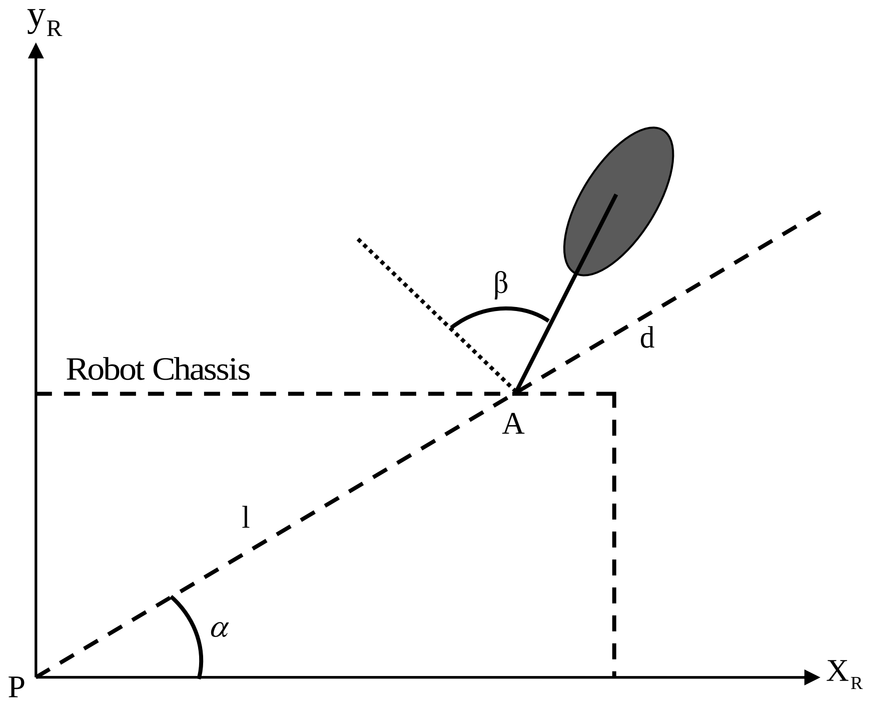


Figure 3.5 Caster wheel and its parameters

3.3.4 Swedish Wheel

Unlike the fixed standard wheel and the steered standard wheel, Swedish wheel does not comprise of the vertical axis of rotation, yet be able to move in all directions (i.e. Omni directionally). Swedish wheels consist of at least three rollers whose axes are tangent to the wheel perimeter and free about rotation. Swedish wheel can be designed similar to a fixed standard wheel with rollers connected to the wheel perimeter with axes that are antiparallel to the main axis of the fixed wheel component. The exact angle between the roller axes and the main axis (γ) can vary, as shown in figure 3.6. The rolling and sliding constraints are;

Along the wheel plane;

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

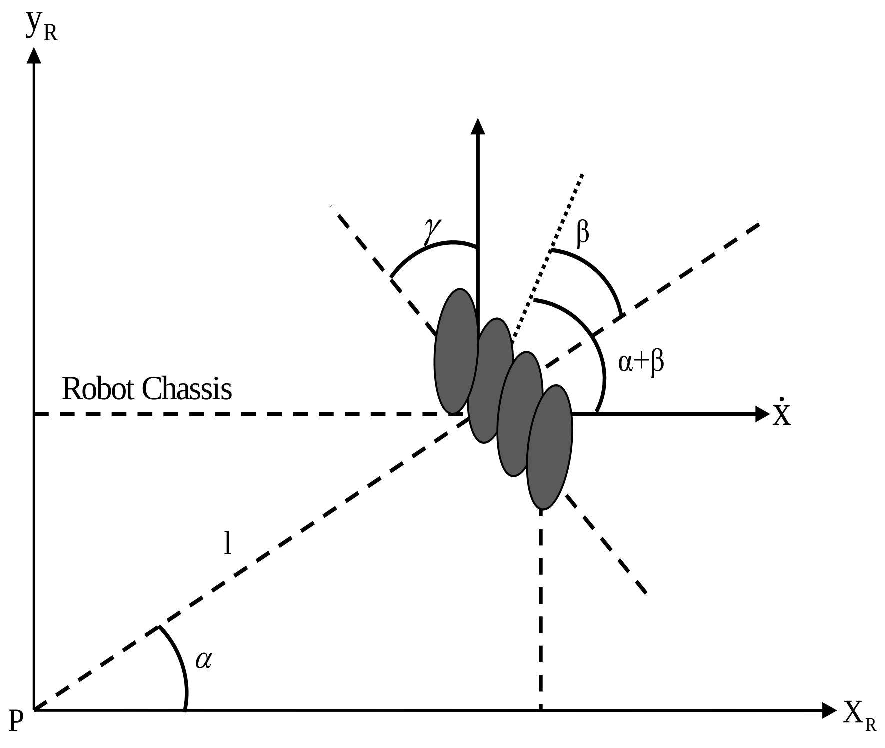


Figure 3.6 Swedish wheel and its parameters

Orthogonal to the wheel plane;

$$\left[\sin(\alpha+\beta+\gamma) \quad \cos(\alpha+\beta+\gamma) \quad l\sin(\beta+\lambda)\right]R(\theta)\dot{\xi}_I - r\dot{\phi}\sin\gamma - r_{sw}\dot{\phi} = 0 \tag{3.11}$$

3.3.5 Spherical Wheel

The spherical or ball wheel places no direct constraints on motion. As with castor wheels and Swedish wheels, the spherical wheel is omnidirectional and places no constraints on the robot chassis kinematics. Therefore, equation simply describes the roll rate of the ball in the direction of motion VA of point A on the robot, as shown in figure 3.7.

Along the wheel plane;

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

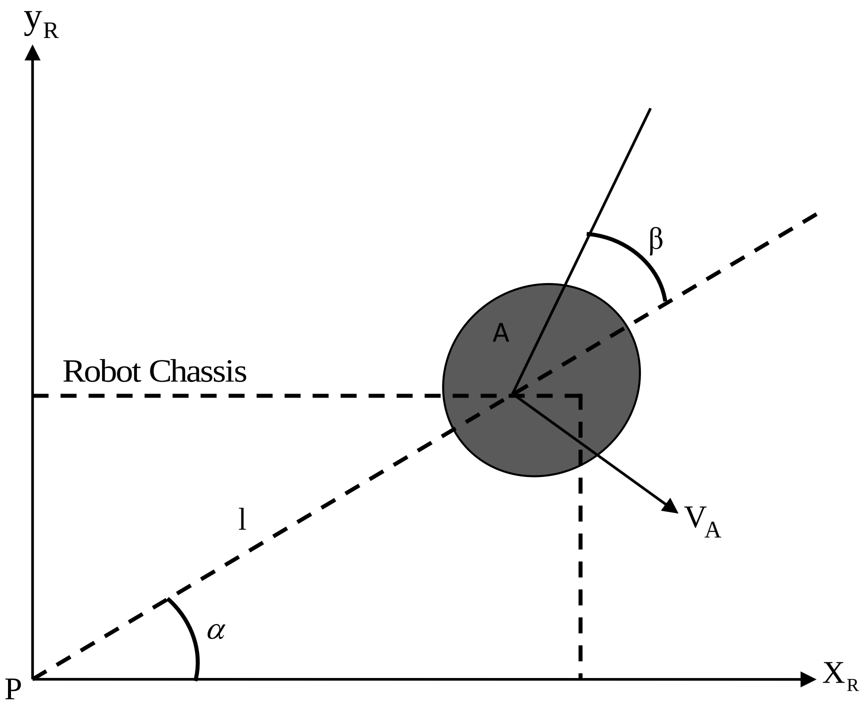


Figure 3.7 Spherical wheel and its parameters

Orthogonal to the wheel plane;

$$\begin{bmatrix}\cos(\alpha+\beta) & \sin(\alpha+\beta) & l\sin\beta\end{bmatrix} R(\theta)\dot{\xi}_I = 0 \tag{3.13}$$

3.4 Robot Kinematic Constraints

Let us consider a wheeled mobile robot having N wheels containing N_f fixed standard wheels, N_s steered standard wheels, N_c caster wheels, N_{sw} swedish wheels and N_{sph} spherical wheels. It is clear from the wheel kinematic constraints equations that the castor wheel, swedish wheel, and spherical wheel do not enforces any kinematic constraints on the robot chassis. Therefore only fixed standard wheels and steerable standard wheels have impact on robot chassis kinematics and therefore require consideration when computing the robot's kinematic constraints. The rolling constraints of all wheels can now be collected in a single expression:

$$J_1(\beta)R(\theta)\dot{\xi}_1 - J_2\dot{\phi} = 0 \tag{3.14}$$

Where J_1 represents a matrix for all wheels to their motions along their specific wheel planes, and J_2 is a constant diagonal matrix N×N of all standard wheels radii.

$$J_1(\beta) = \begin{bmatrix} J_1(\beta) & J_{1s}(\beta_s) & J_{1c}(\beta_c) & J_{1sw} & J_{1sph} \end{bmatrix}^T \tag{3.15}$$

Where J_{1f}, J_{1s}, J_{1c}, J_{1sw}, J_{1sph} are the matrices of size (N_f × 3), (N_s × 3), (N_c × 3), (N_{sw} × 3) and (N_{sph} × 3).

J2 is a constant (N × N) matrix whose diagonal elements are the radii of the wheels, apart from for the radii of the swedish wheels which are multiplied by cosγ.

In the same way we can frame the sliding constraints of all wheels in a single expression;

$$C_1(\beta)R(\theta)\dot{\xi}_1 = 0 \tag{3.16}$$

Where,

$$C_1(\beta) = \begin{bmatrix} C_1(\beta) & C_{1s}(\beta_s) & C_{1c}(\beta_c) & C_{1sw} & C_{1sph} \end{bmatrix}^T \tag{3.17}$$

3.5 Mobile Robot Maneuverability

The kinematic mobility of a robot chassis is its capability to travel in the environment. The basic constraint limiting mobility is the law that each wheel must satisfy its sliding constraint. Along with instantaneous kinematic motion, a mobile robot is capable to further manipulate its position, over time, by steering steerable wheels. The overall maneuverability of a robot is thus a combination of the existing mobility based on the kinematic sliding constraints of the standard wheels, plus the additional freedom contributed by steering and spinning the steerable standard wheels.

3.5.1 Degree of Mobility:

It has been clear from the above expressions that the caster wheel, swedish wheel, and spherical wheel not impose any kinematic constraints on the robot chassis. As a result only fixed standard wheels and steered standard wheels have been considered for calculating the robot kinematic constraints. Let us now consider

(N_f + N_s) wheels, to avoid lateral slip;

$$C_{1f} R(\theta) \dot{\xi}_1 = 0 \tag{3.18}$$

$$C_1(\beta_s) R(\theta) \dot{\xi}_1 = 0 \tag{3.19}$$

$$C_1(\beta_s) = \left[C_{1f} \quad C_{1s}(\beta_s) \right]^T \tag{3.20}$$

Mathematically, the null space of $C_1(\beta_s)$ is the space N such that for any vector n in N,

$$C_1(\beta_s).n = 0 \tag{3.21}$$

In practice, a wheeled mobile robot will have zero or more fixed standard wheels and zero or more steerable standard wheels. We can therefore identify the possible range of rank values for any robot: $0 \leq \left[C_{1s}(\beta_s) \right] \leq 3$

Consider the case rank $\left[C_{1s}(\beta_s) \right]$= 0, this is only possible if there are zero independent kinematic constraints in$C_{1s}(\beta_s)$. In this case there are neither fixed nor steerable standard wheels attached to the robot frame: N_f = N_s = 0.

Now consider another case when $\left[C_{1s}(\beta_s) \right]$= 3, then the robot is completely constrained in all directions and is, therefore, degenerate since motion in the plane is totally impossible.

Now we can define a robot's degree of mobility δ_m:

$$\delta_m = \dim N \left[C_{1s}(\beta_s) \right] = 3 - \text{rank} \left[C_{1s}(\beta_s) \right] \tag{3.22}$$

3.5.2 Degree of Steerability:

A wheeled mobile robot can be steered freely with the number of centered orientable wheels to it. Therefore degree of steerability can be calculated as:

$$\delta_s = \text{rank} \left[C_1(\beta_s) \right] \tag{3.23}$$

The range of δ_S can be specified: $0 \le \delta_s \le 2$.

3.5.3 Robot Maneuverability

Robot maneuverability (δ_M) is the overall degrees of freedom that a robot can handle.

$$\delta_M = \delta_m + \delta_s \tag{3.24}$$

Robot maneuverability for five basic types of three wheel configuration is given below in Table 3.1.

Table 3.1 Robot maneuverability (δ_M) for five basic types of three wheel robots

Wheel Configuration	δ_m	δ_s	δ_M
Omnidirectional (Three Spherical wheels)	3	0	3
Differential(Two Fixed standard wheels and one Spherical wheel)	2	0	2
Omni-steer(Two spherical wheels and one Steered standard wheel)	2	1	3
Tricycle(Two Fixed standard wheel and one Steered standard wheel)	1	1	2
Two Steer(two steered standard wheels and one spherical wheel)	1	2	3

In the above wheel configuration spherical wheels can be replaced by the caster wheels or Swedish wheels, without affecting the maneuverability of the robot.

3.6 Kinematic Analysis of the Differential Drive Wheeled Mobile Robot

Let us consider a differential drive robot, for which different parameters are shown in table 3.2 below.

Table 3.2 Parameters of the kinematic model of the mobile robot

Sl. No.	Symbol	Parameter
1	r	Radius of each wheel.
2	l	Distance between the two driving wheels along the y-axis of robot.
3	V_R	Linear velocity of the right wheel.
4	V_L	Linear velocity of the left wheel.
5	V_ω	Angular velocity of the robot.
6	V_l	Linear velocity of the robot along x-axis of the robot.
7	c	Centre of the axis of the rear wheels.
8	R	Radius of curvature for the robot trajectory.

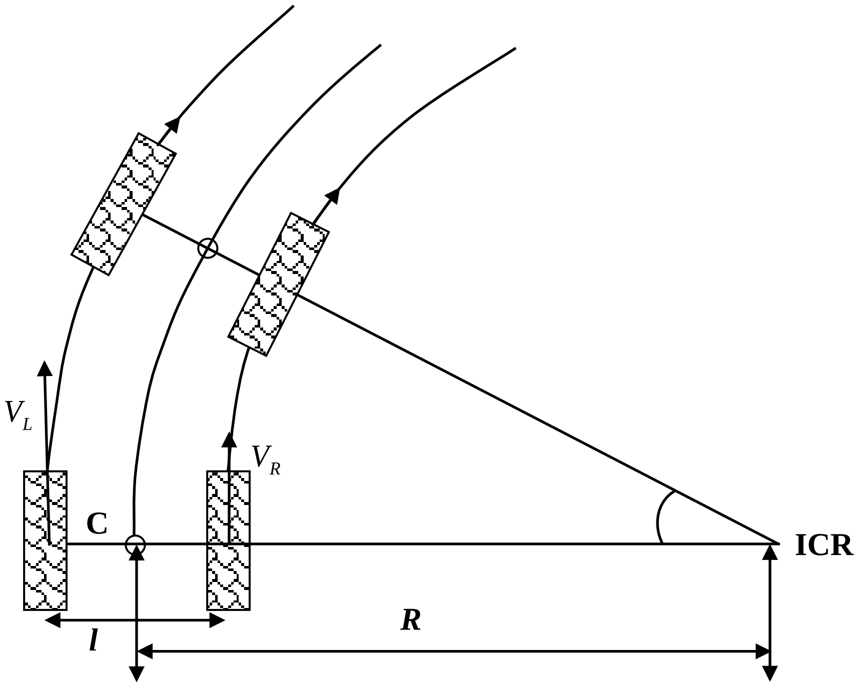


Figure 3.8 Instantaneous Centre of rotation (ICR)

Let us consider, at any instance of time ‘t’ the robot follows a path shown in figure 3.8. The robot turns along a curve around the instantaneous centre of rotation (ICR). Angular velocity will be given by;

$$V_{\omega}(t)=\frac{d\theta(t)}{dt}$$

(3.25)

Hence linear velocity of the robot;

$$V_l(t)=V_{\omega}(t)R(t) \tag{3.26}$$

Also

Left and right wheel velocities can be written as;

$$V_L(t)=V_{\omega}(t)(R(t)+\frac{l}{2}) \tag{3.27 a}$$

$$V_R(t)=V_{\omega}(t)(R(t)-\frac{l}{2}) \tag{3.27 b}$$

On solving the above equations, radius of curvature of the robot trajectory will be given by;

$$R(t)=\frac{l(V_L(t)+V_R(t))}{2(V_L(t)-V_R(t))} \tag{3.28}$$

Now angular velocity and linear velocity of the robot can be rewritten as;

$$V_{\omega}(t)=\frac{V_L(t)-V_R(t)}{l} \tag{3.29 a}$$

$$V_l(t)=\frac{V_L(t)+V_R(t)}{2} \tag{3.29 b}$$

It is clear from the above equations that, by changing the velocities of the two driving wheels of the robot different trajectories will be followed by the robot. In differential drive robot, small variations in the velocities of the two wheels results in significant change in the trajectory. Therefore errors due to slippage should take into consideration during the trajectory planning of a mobile robot.

The kinematic equations in the world frame can be written as;

$$\dot{x}(t) = V_l(t)\cos\theta(t) \tag{3.30 a}$$

$$\dot{y}(t) = V_l(t)\sin\theta(t) \tag{3.30 b}$$

$$\dot{\theta}(t) = V_\omega(t) \tag{3.30 c}$$

On integrating the above equations, we get;

$$x(t) = \int_0^t V_l(\tau)\cos\theta(\tau)d\tau + x_0 \tag{3.31 a}$$

$$y(t) = \int_0^t V_l(\tau)\sin\theta(\tau)d\tau + y_0 \tag{3.31 b}$$

$$\theta(t) = \int_0^t V_\omega(\tau)d\tau + \theta_0 \tag{3.31 c}$$

Where $(x_0, \mathrm{y}_0, \theta_0)$ represents the initial position of the robot. Above equations can be used for the robot capable of travelling towards a particular direction $\theta(t)$ with a velocity of $V_l(t)$

3.7 Summary

In this chapter, kinematic model for five basic type of wheel has been presented. The analysis is based on the constraints on mobility of the robot associated with the different wheel constraints. A model has been presented to describe the motion of the robot having numbers of wheels which can be successfully applied to the all basic wheel configurations.

Chapter 4

ANALYSIS OF BACTERIAL FORAGING OPTIMIZATION ALGORITHM FOR NAVIGATION OF MOBILE ROBOTS

4.1 Introduction

The evolution of the Artificial Intelligence (AI) techniques for navigation of mobile robots in the real-world environments has got remarkable attention by the researchers in the field of mobile robotics. In particular, the paradigm has shifted towards the application of the nature-inspired algorithms in the autonomous system. In this chapter, bacterial foraging optimization algorithm based approach has been used for the navigation of mobile robots. The focus has been made to design a robust controller for path planning of mobile robots in known and unknown environment.

4.1.1 Overview

Bacterial Foraging Optimization Algorithm (BFOA), proposed by Passino [140], is a kind of nature-inspired optimization algorithm. The algorithm is inspired from the foraging behavior of a swarm of E. coli [140] bacterial in multiple directions. Swarm of bacteria looks for nutrients by making best use of energy achieved per unit time. Each bacterium in the swarm corresponds with others by transmitting the appropriate signals. The movement of the bacterium during the foraging is known as chemotaxis. BFOA has gained the interests of scientists and researchers from diverse area of discipline mainly because of its biological inspiration [141-143]. In recent years, research community has observed a significant development in the classical BFOA algorithm.

4.1.2 Cause and Problem Formulation for Considering BFOA

There are many optimization algorithms for solving the mobile robot navigation problem. They are categorized as follows:

Potential Field Method

Artificial Neural Networks

Fuzzy Logic Control

Genetic Algorithms and

Bacterial Foraging Optimization Algorithm

Among the above stated optimization algorithms, Bacterial Foraging Optimization Algorithm (BFOA) is proposed in this work to find the near optimal path for mobile robots from its initial position to goal position. The logic behind selecting Bacterial Foraging Algorithm is that this algorithm is not mostly affected by the size and non-linearity of the problem. Also this algorithm has met to the near optimal solution in many problems where the most analytical methods have failed to converge. This algorithm also has advantages such as,

Advantages:

- Less computational problem,
- global convergence,
- less computational time requirement, and
- It can handle more number of objective functions when compared to the other evolutionary algorithms.

In the current research, robot has to navigate in an unknown environment by following a collision free path. For this objective function is to be used therefore BFOA is taken into consideration in the current research.

The prime objective of the robots is to reach some pre-defined target location by following collision free shortest trajectory in minimum possible time. Furthermore, the energy used by the robot per unit time should be minimum. In order to acquire essential information of the environment such as location of the obstacles and the goal, each robot is equipped with a number of sensors. The data received from the sensors is used to detect the obstacles presents in the robots sensing range and to perceive the position of target.

4.2 Bacterial Foraging Optimization Algorithm (BFOA)

After studying the group behavior of the swarm of E. coli bacteria, Passino [140**]** presented bacterial foraging optimization algorithm by considering four principle events, namely chemotaxis, swarming, reproduction and elimination and dispersal. Since then, BFOA has been applied in various optimization problems associated with the real-world and therefore already gained the attention of the researchers in the domain.

The group of bacteria *S* behaves as follows

Consider a group of bacteria arbitrarily spread in the map of nutrients.

- Bacterium forages for nutrient region and travel towards high-nutrient regions in the map. Those successfully found their food in sufficient, gets longer and will split into two equal parts. (i.e. reproduce). On the other hand, those sited low nutrient regions will disperse and those who to be found in noxious region will die.
- Bacterium located in the nutrient region attracts the others by generating chemical attractant and those neighbouring to noxious region warns others via chemical signals.
- Now the swarm of bacteria situated in the highly nutrient place on the map.
- Bacteria are dispersed in the map as they look for new region of nutrients.

Bacterial foraging behavior includes four main steps (1) Chemotaxis (tumble and swimming), (2) swarming (3) reproduction and (4) elimination- dispersal.

4.2.1 Chemotaxis

Chemotaxis is the very first step of the BFOA. During the chemotaxis, swimming and tumbling actions of the bacterium has been simulated. During the foraging, if the bacterium finds a promising region, it will move in the same direction or we can say that it will swim for a period of time. On the other hand, if the bacterium falls in the noxious region then it will try to explore its search action by following a tumble in random directions. Thus explore search for new promising region.

In chemotaxis step, the movement of the E.coli during swimming and tumbling through flagella is simulated. Biologically the movement of the E. coli bacterium is limited in two different ways, either it can swim for a phase of time in the same direction as prior step or it may tumble for the entire lifetime. In the classical BFOA, a unit step in an arbitrary direction represents a 'tumble' and a unit step in the same direction in the last step indicates a 'run'. Suppose $\theta^i(j,k,l)$ represents the bacterium at j^{th} chemotactic, k^{th} reproductive, and l^{th} elimination-dispersal step. $C(i)$ is the chemotaxis step size during each tumble or run. Then the movement of the i^{th} bacterium can be modeled as;

$$\theta^i(j+1,k,l)=\theta^i(j,k,l)+C(i)\phi(j) \tag{4.1}$$

where, $C(i)(i=1,2,3,...,S)$ is the size of the step taken in random direction specified by the tumble. $J(i,j,k,l)$ is the fitness, which also represents the cost at the location of the i^{th} bacterium $\theta^i(j,k,l)\in\Re^P$. If at $\theta^i(j+1,k,l)$ the cost $J(i,j,k,l)$ is smaller (better) than at $\theta^i(j,k,l)$, then another step of size $C(i)$ in this same direction will be taken. Otherwise, the bacteria will tumble via taking another step of size $C(i)$ in random directions in order to look for better nutrient environment. This process leads to gathering of bacteria into the nutrient-rich areas.

4.2.2 Swarming

An Interesting group behavior has been observed in several motile species of bacteria, including E.coli and S.typhimurium. A cell-to-cell communication mechanism is established to simulate the biological behavior of bacteria swarming. During the motion each bacterium releases chemical attractant to indicate other bacteria to swarm towards it. Meanwhile, each bacterium releases chemical repellent to notify other bacteria to keep a safe distance from it. BFOA simulates this social behavior by representing the combined cell-to-cell attraction and repelling effect. Now Combined cell-to-cell attraction and repelling effects can be modelled as;

$$\begin{aligned} J_{cc}(\theta, P(j,k,l)) &= \sum_{i=1}^{S} J_{cc}^{i}(\theta, \theta^{i}(j,k,l)) \\ &= \sum_{i=1}^{S}[-d_{attract}\exp(-w_{attract}\sum_{m=1}^{p}(\theta_m - \theta_m^i)^2)] + \sum_{i=1}^{S}[h_{repellant}\exp(-w_{repellant}\exp\sum_{m=1}^{p}(\theta_m - \theta_m^i)^2)] \end{aligned} \tag{4.2}$$

Where, $d_{attract}$ and $w_{attract}$ are the depth and width of the attractant released by the cell respectively. Likewise $h_{repellant}$ and $w_{repellant}$ are the height and measure of the width of the repellent effect. The values of attractants and repellents coefficients should be chosen properly.

4.2.3 Reproduction

For bacteria, a reproduction step takes place after all chemotactic steps. Now the fitness of i^{th} bacterium is given below:

$$J_{fitness}^{i} = \sum_{j=1}^{N_c+1} J(i,j,k,l) \tag{4.3}$$

For reproduction the population is arranged in ascending order of fitness of the bacterium. Then in reproduction step the 50 % least fit bacteria finally die and while the others having good health will reproduce (split) to keep the population constant, which is convenient in coding the algorithm.

4.2.4 Elimination- Dispersal

It is expected that the local environment where a population of bacteria live changes either gradually (e.g., via consumption of nutrients) or suddenly due to some other influence. This event may result in the sudden death of the population of bacteria (e.g. sudden increase in heat) or dispersal of a group of bacteria (e.g. sudden water flow) to a new location. In BFO, the dispersion event happens after a certain number of reproduction processes and some bacteria are chosen depending upon the probability to be killed and moved to another position within the environment.

The pseudo code for the bacteria foraging optimization algorithm (BFOA) suggested by [141] is presented below:

```
Initialize the parameters viz. $p, S, N_c, N_s, N_{re}, N_{ed}, p_{ed}$, and $C(i)(i = 1, 2, 3, ..., S)$.
1) Elimination-dispersal loop: $l = l + 1$
2) Reproduction loop: $k = k + 1$
3) Chemotaxis loop: $j = j + 1$
        a) For $i = 1, 2, ..., S$, take a chemotactic step for bacterium $i$ as follows.
        b) Compute $J(i, j, k, l)$. Let $J(i, j, k, l) = J(i, j, k, l) + J_{cc}(\theta^i(j, k, l), P(j, k, l))$ (4.4)
        c) Let $J_{last} = J(i, j, k, l)$ to save this value since we may find a better cost via a run.
        d) Tumble: Generate a random vector $\Delta(i \in \Re^p)$ with each element $\Delta_m(i), m = 1, 2, ..., p,$ random number on [-1, 1].
        e) Move: let
```

$$\theta^i(j+1,k,l)=\theta^i(j,k,l)+C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \tag{4.5}$$

This results in a step size $C(i)$ in the direction of the tumble for bacterium i.

f) Compute $J(i,j+1,k,l)$, and then let

$$J(i,j+1,k,l)=J(i,j,k,l)+J_{cc}(\theta^i(j+1,k,l),P(j+1,k,l)) \tag{4.6}$$

g) Swim

i) Let $m = 0$ (counter for swim length).

ii) While $m < N_s$ (if have not climbed down too long)

Let $m = m+1$.

If $J(i,j+1,k,l) <$ J_{last} (if doing better), let $J_{last}=J(i,j+1,k,l)$ and let

$$\theta^i(j+1,k,l)=\theta^i(j+1,k,l)+C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \tag{4.7}$$

And use this $\theta^i(j+1,k,l)$ to compute the new $J(i,j+1,k,l)$ as we did in f).

Else, let $m = N_s$. This is the end of the while statement.

h) Go to next bacterium (i+1) if $i \neq S$ (i.e. go to b) to process the next bacterium).

4) If $j < N_c$, go to step 3. In this case, continue chemotaxis, for the life of the bacteria is not over.

5) Reproduction:

a) For the given k and l and for each i= 1, 2…., S, let

$J^i_{health}=\sum_{j=1}^{N_c+1} J(i,j,k,l)$ be the health of bacterium i (a measure of how many nutrients it got over its lifetime and how successful it was at avoiding noxious substances). Sort bacteria and chemotaxis parameters $C(i)$ in order of ascending cost J_{health} (higher cost means lower health).

b) The S_r Bacteria with the highest J_{health} values die and the others S_r bacteria with the best values split (and the copies that are made placed at the same location as their parent).

6) If $k < N_{re}$, go to step 2. In this case, we have not reached the number of specified reproduction steps, so we start the next generation in the chemotactic loop.

7) Elimination-dispersal: For i=1, 2…., S, with probability p_{ed}, eliminate and disperse each bacterium (this keeps the number of bacteria in the population constant). To do this, if you eliminate a bacterium, simply disperse one to a random location on the optimization domain.

8) If $l < N_{ed}$, then go to step 1; otherwise end.

According to this algorithm, i^{th} bacterium at any point takes a chemotaxis step with a constant step size in the random direction and evaluates the cost function at each step. Now if the cost function of the next point is smaller (better) than the last point then a new step in the same direction will be taken. This process will be continued for the defined swim length. Then after each chemotaxis step the least healthy bacteria as specified by the cost function are switched by the copies of the fit ones. This process is called reproduction step and it followed by the elimination-dispersal event. For this event, each bacterium in the group is exposed to elimination-dispersal with a pre-defined probability.

4.3 Mobile Robot Path Planning using Bacterial Foraging Optimization Algorithm (BFOA)

In this section, an algorithm motivated from bacterial foraging algorithm is used for the path planning of a mobile robot. Navigation is the methodology that permits directing a mobile robot to attain pre-defined target in an environment with obstacles in a simple and safe way. Navigation can be divided into two basic tasks as perception and path planning.

Perception is to gather the information of the environment consisting of obstacles with the help of different sensors. Global path planning can be used with local path planning for navigation in indoor applications since the environment is partially known whereas for outdoor environment local path planning methods are preferred over the global path planning approaches.

4.3.1 Decision Scheme for Finding the Optimal Bacterium

In the proposed approach, if robot detects any obstacle a group of bacterium is generated randomly around it in a circle of radius equal to step size. Then position of each bacterium can be defined as $\delta_s(\mathrm{t})$ and the next position can be calculated as;

$$\delta_s(\mathrm{t}+\Delta\,\mathrm{t}) = \delta_s(\mathrm{t}) + (\mathrm{C(t)} \times \phi(\mathrm{j})) \tag{4.8}$$

Where, $\phi(\mathrm{j})$ is the unit length random vector which is used to express the bearing of bacterium in each time. During the time $(\mathrm{t}+\Delta\,\mathrm{t})$, the group of bacterium search for the optimal path towards the goal, while avoiding the obstacles. The bacterium which has found the best path for the next position is chosen and the mobile robot goes to that particular position defined by the best bacterium and it will continue over and over again till the goal is reached by the robot. For deciding the best bacterium, two factors are considered and combined in order to get the optimum result. First the distance between the goal and the mobile robot and second, the distance of the nearest obstacle from the robot’s current position. When we combine these two factors it results in an attractant-repellent profile which directs the robot towards the goal position which has the global minimum.

When the robot senses any obstacle, the obstacle function $J_{obstacle}$ virtually assign repulsive Gaussian cost function to that obstacle, which is shown by the expression below:

$$J_{obstacle} = h_{obstacle} * \left(\frac{1}{\exp(w_{obstacle} * (\|\delta_i(t) - \mathrm{P}_{obstacle}(t)\|^2)} \right) \tag{4.9}$$

Where $h_{obstacle}$ and $w_{obstacle}$ are constant values corresponding to the height and the width of the repellent respectively. $\mathrm{P}_{obstacle}$ represents the position of the obstacle sensed by the robot through sensors mounted on it. $(\| \delta_i(t) - \mathrm{P}_{obstacle}(t) \|$ represents the Euclidean distance between the robot current position with the obstacle nearby. It should be noticed that, $J_{obstacle}$ has a value only when the robot detects any obstacle in its sensing range. Hence we can define the cost function $J_{obstacle}$ by equation (4.10).

$$J_{obstacle} = \begin{cases} \left\{ h_{obstacle} * \left(\frac{1}{\exp(w_{obstacle} * (\|\delta_i(t) - \mathrm{P}_{obstacle}(t)\|^2)} \right), \text{when} \|\delta_i(t) - \mathrm{P}_{obstacle}(t)\|^2 \leq \beta_{robot} \right. \\ 0, \text{when} \|\delta_i(t) - \mathrm{P}_{obstacle}(t)\|^2 \geq \beta_{robot} \end{cases} \tag{4.10}$$

Where, β_{robot} is the sensing range of the mobile robot. In a similar way, an attractant Gaussian cost function has been assigned to the pre- defined goal position. Equation 4.11 represents the goal cost function.

$$J_{goal} = h_{goal} * \left(\frac{1}{\exp(w_{goal} * (\|\delta_i(t) - \mathrm{P}_{goal}\|^2)} \right) \tag{4.11}$$

Where, h_{goal} and w_{goal} are the height and width of the attractant respectively. $\| \delta_i(t) - \mathrm{P}_{goal} \|^2$ is the Euclidean distance between goal the robot’s current position.

Obstacle avoidance and goal seeking are the two most essential behavior that need to be controlled during the path planning of the mobile robots. For this, a cost function has been proposed as given below:

$$J_{total} = J_{obstacle} + J_{goal} \tag{4.12}$$

Equation (4.12) is further used to find the cost value of each individual bacterium at each step. The bacterium are then sorted in ascending order with respect to their cost value. Further, bacterium having best fitness (lower cost value) is chosen and the robot goes to that particular location in next step. A series of such small steps directs the robot towards the goal position by avoiding the obstacles at the same time.

4.3.2 Control Law

The control scheme, designed for the path planning of mobile robots based on the proposed algorithm has two parts.

(a) High Level Controller for Decision Making

In first part, a high level controller is used to find out the best bacterium among the group of bacterium, randomly generated around the robot in small time $\Delta \mathrm{t}$. The decision of finding the best bacterium is based on the distance error of the bacterium to the goal position $e_i^{dist}(\mathrm{t}+\Delta \mathrm{t})$ and the cost function error $e_i^J(\mathrm{t}+\Delta \mathrm{t})$. Suppose, $d_i(\mathrm{t}+\Delta \mathrm{t}) = \| \delta_i(\mathrm{t}+\Delta \mathrm{t}) - \mathrm{P}_{goal} \|^2$ and $d_i(\mathrm{t}) = \| \delta_i(\mathrm{t}) - \mathrm{P}_{goal} \|^2$ represents the distance of the bacterium *s* to the goal position at time t and $(\mathrm{t}+\Delta \mathrm{t})$. Therefore, the distance error and cost function error can be stated as

$$e_i^{dist}(\mathrm{t}+\Delta \mathrm{t}) = d_i(\mathrm{t}+\Delta \mathrm{t}) - d_i(\mathrm{t}) \tag{4.13}$$

$$e_i^J(\mathrm{t}) = J_{total}(\delta_i(\mathrm{t}+\Delta \mathrm{t})) - J_{total}(\delta_i(\mathrm{t})) \tag{4.14}$$

In time (t+ $\Delta \mathrm{t}$), total cost and distance from the goal position of each bacterium are calculated and arranged according value of distance error such that the bacterium with the least error is at top. Since the prime objective here is to move the robot towards the goal position, hence the bacterium that has negative value of cost error function is chosen with accordance to the minimum distance error. Moreover, the bacterium should be kept away from the obstacles as if cost function error of any bacterium $e_i^J(\mathrm{t}) > 0$ signifies that it is in the region of obstacles. Therefore the first bacterium having negative value of distance error in the order is selected as fittest bacterium and robot approaches towards position of this fittest bacterium in the next step**.**

(b) Low Level Controller for Decision Making

Since the algorithm generates a group of bacterium around the robot randomly, it is possible that some of the bacterium might be positioned near to sensed obstacles. In general these bacterium do not show better fitness value and omitted. However robot should approach towards the obstacle to find the optimal path. This correction has been made as a low level control scheme, which modifies the robot path. The control law is defined as;

$$u(\mathrm{t}) = \mathrm{C}(\mathrm{t}) \frac{(\delta_{ibest}(\mathrm{t}+\Delta\,\mathrm{t}) - \mathrm{q}(\mathrm{t}))}{\left\|(\delta_{ibest}(\mathrm{t}+\Delta\,\mathrm{t}) - \mathrm{q}(\mathrm{t}))\right\|} \tag{4.15}$$

Where, $\mathrm{q}(\mathrm{t})$ is the position of the robot at time t and $\delta_{ibest}(\mathrm{t}+\Delta\,\mathrm{t})$ is the position of a bacterium at time $(\mathrm{t}+\Delta\,\mathrm{t})$ having optimal position and to be chosen as best bacterium by the algorithm.

4.3.3 Simulation Results

In this section, various exercises have been presented to show the effectiveness of the proposed algorithm in the path planning of the mobile robots in different environment. A generalized code has been written which enables the user to generate any number of robots and target in the simulated environment. The exercises are designed to show the different capabilities of the proposed algorithm. The proposed algorithm has been tested in a simulated environment of 50 square units (1 unit = 3 cm) area for different configurations.

4.3.3.1 Demonstration of Obstacle Avoidance and Goal Seeking Behavior

In the present scenario a robot, moving in a test-platform with eight obstacles and one target is shown in figure 4.1. The environment is a 2D square having side equal to 50 units. The coordinates of the start and goal positions are (10, 45) and (35, 15) respectively. The environment is designed to show the obstacle avoidance and goal seeking capability of the robot using the proposed approach. Initial position of the robot, obstacles and goal position are as shown in figure 4.1. Figure 4.2 illustrates the representation of obstacles defined by obstacle function by equation 4.10. The goal function defined in equation 4.11, is shown in figure 4.3. Figure 4.4 shows the overall cost function as defined in equation 4.12. 2D representation of the path followed by robot from start to goal position is shown in figure 4.5. The same path is shown in a 3D workspace in figure 4.6.It is clear from the figure that the robot successfully reaches the goal position by following the collision free path. Moreover, during the course of the motion robot maintains safe distance from the obstacle.

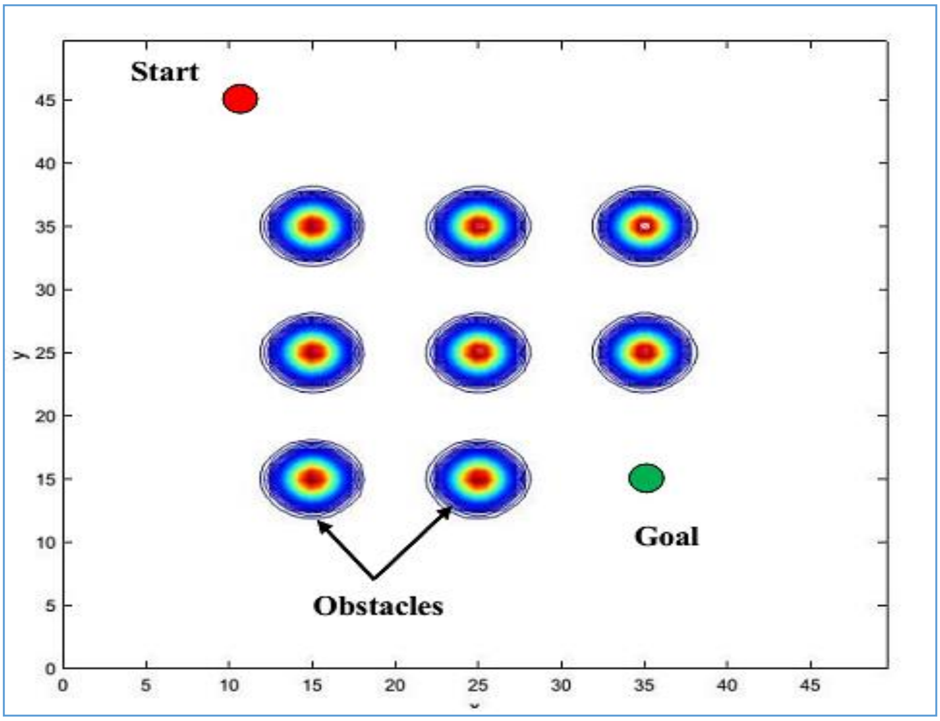


Figure 4.1 Platform for obstacle avoidance and goal seeking task

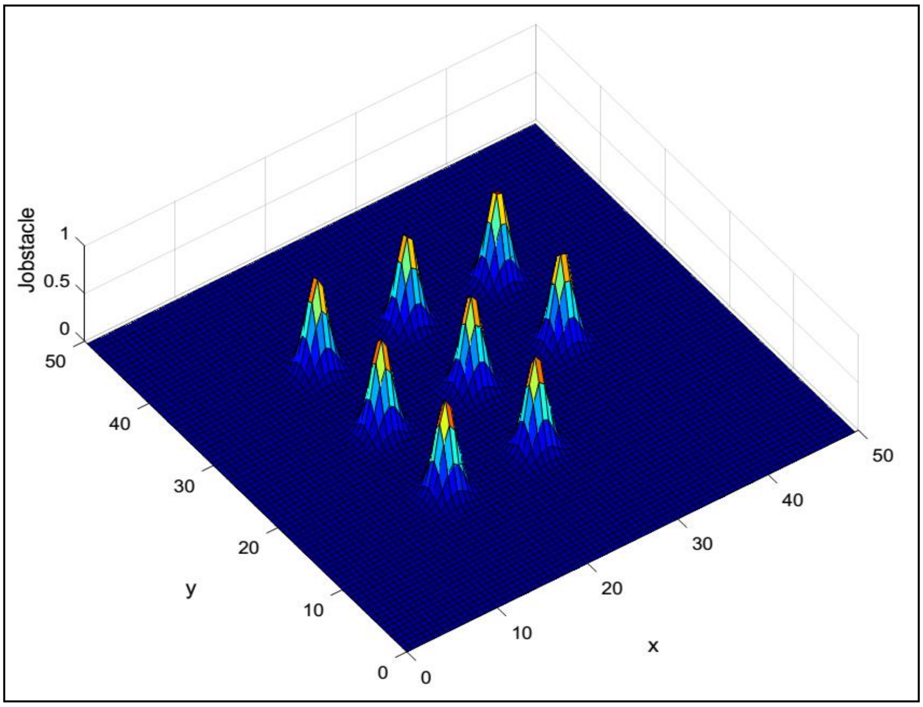


Figure 4.2 Representation of obstacle function

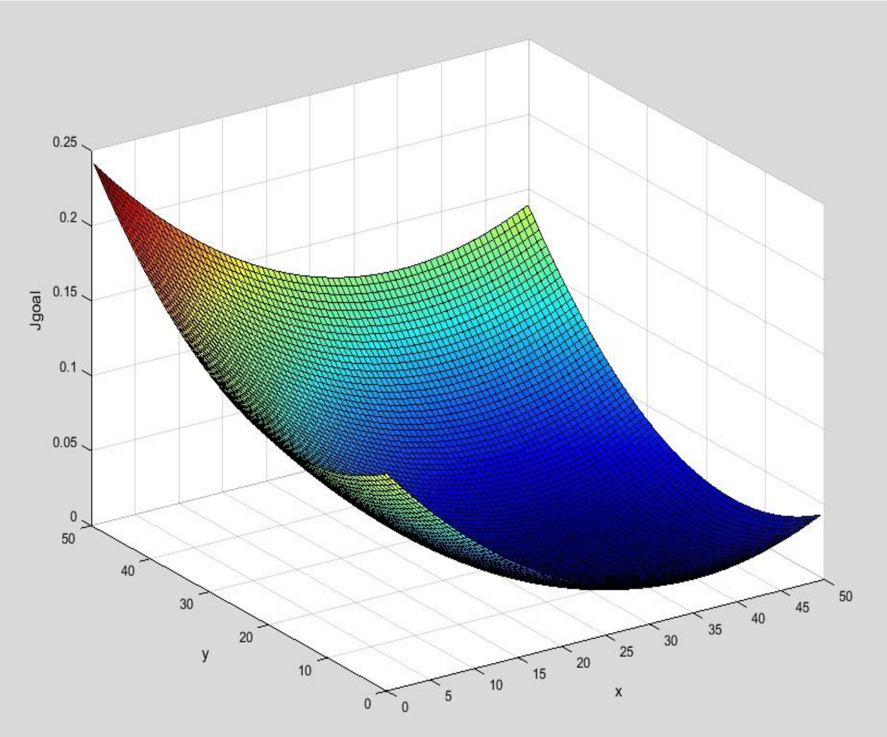


Figure 4.3 Representation of goal function

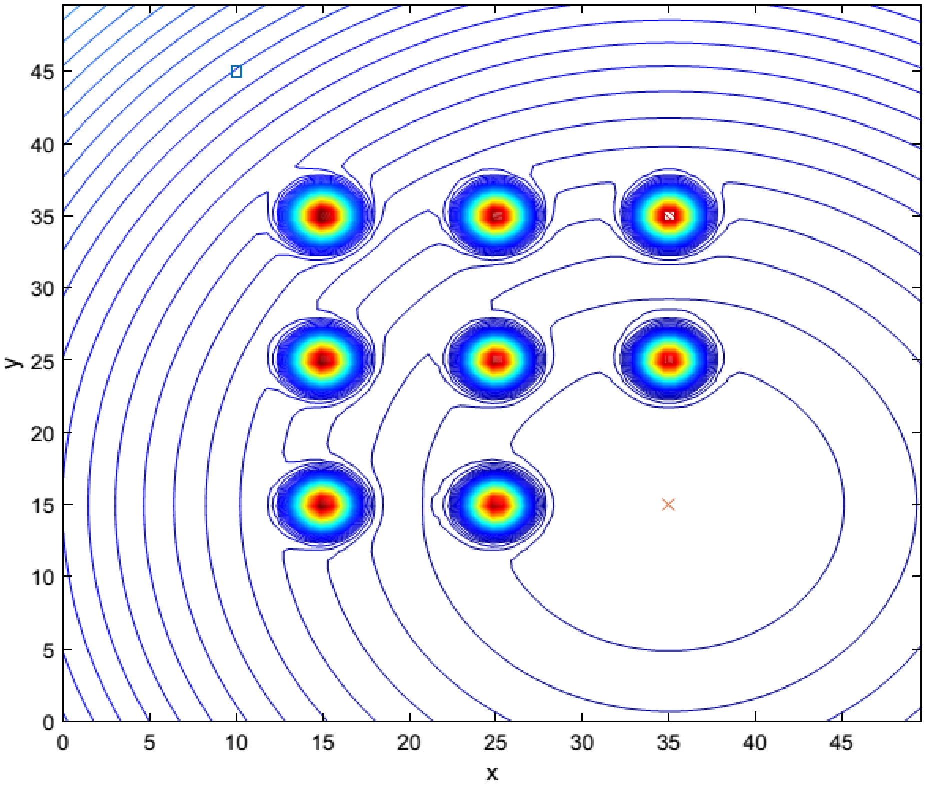


Figure 4.4 Representation of total cost function

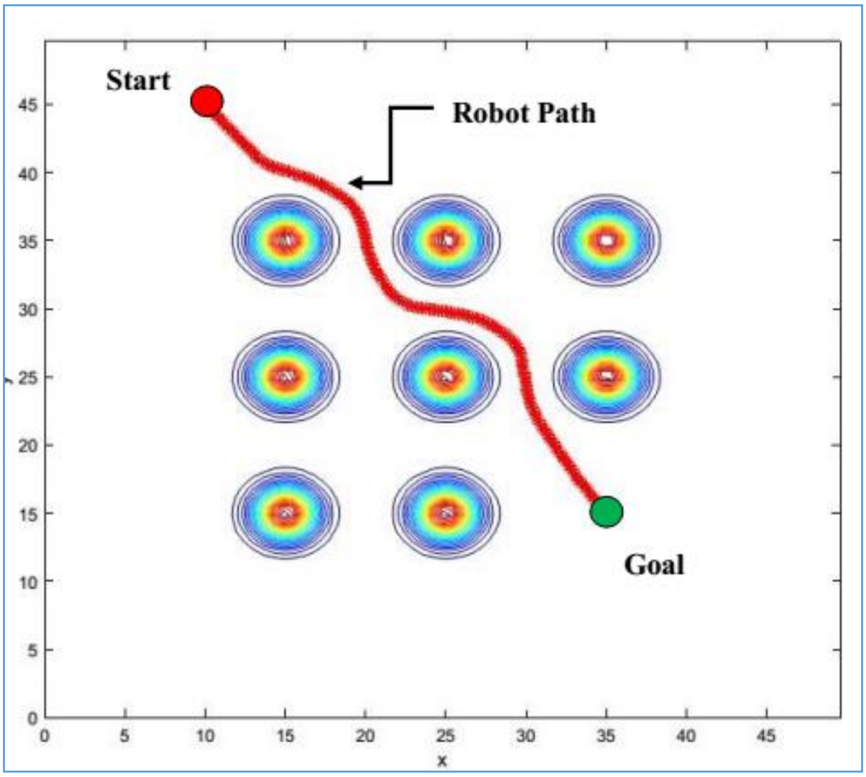


Figure 4.5 2D representation of robot path from start to goal position

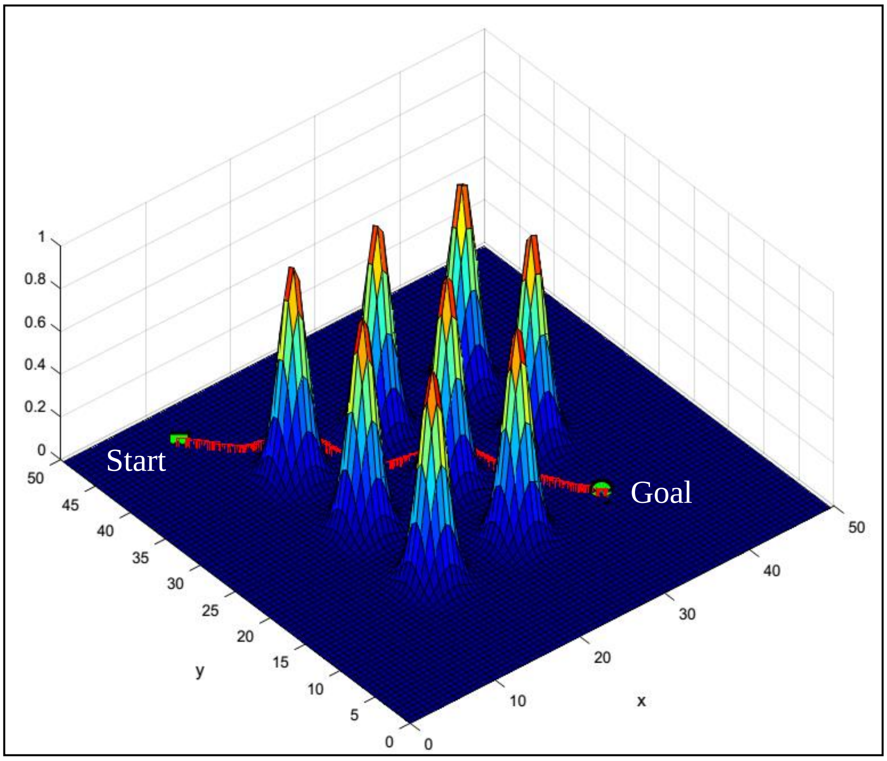


Figure 4.6 3D representation of robot path from start to goal position

4.3.3.2 Demonstration of Wall Following and Goal Seeking Behavior

Figures 4.7 and 4.8 show the robot's path demonstrating the wall following and goal seeking behavior in 2D and 3D test environment respectively. This exercise involves the wall following behavior of single mobile robot in an environment consisting of a number of obstacles. In the current situation the obstacles are organized in a specific manner so that they act like a wall between the robot and the target. Wall following behavior assists the robot to move alongside the wall by keeping the safe distance from it.

In this exercise, robot commences from its start position (5, 5) to the goal position (45, 45) by following the target angle till an obstacle is detected. As the robot meets the wall of obstacles, it changes its heading to avoid the collision and moves continuously alongside the wall till it reaches the corner.

Further, the robot again steers and proceeds towards the goal position by following a series of steering angle directed by the controller.

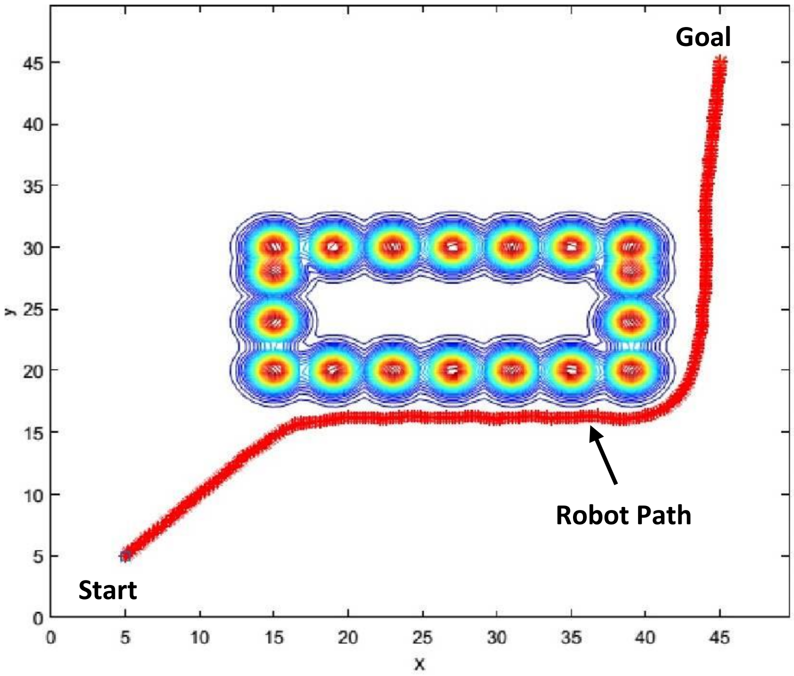


Figure 4.7 Robot path from start to goal position in 2D work space

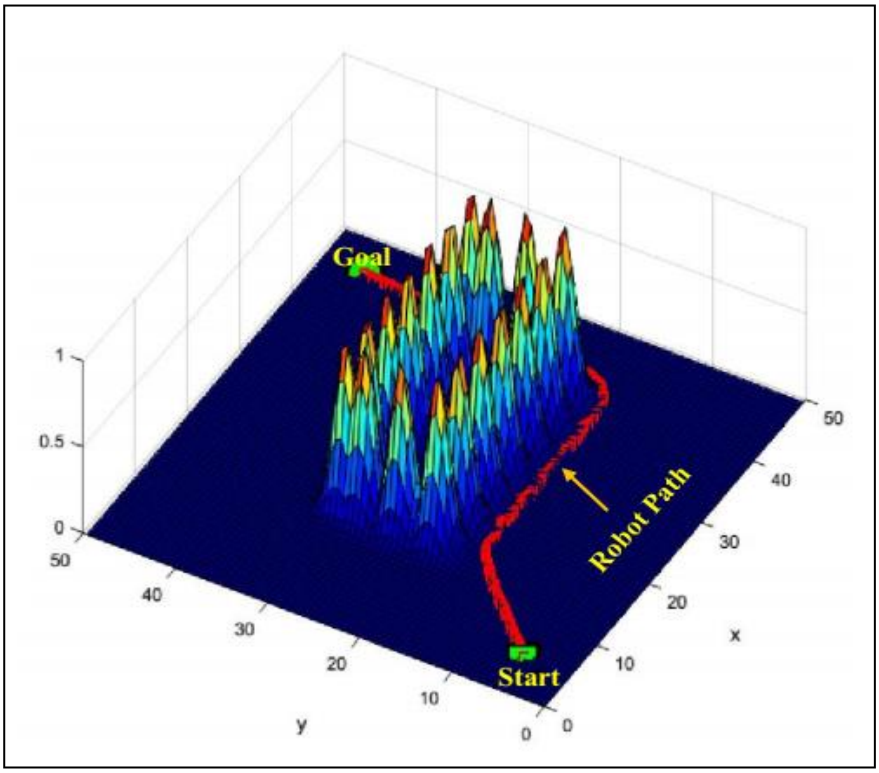


Figure 4.8 Robot’s path from start to goal position in 3D work space

4.3.3.3 Collision Free Motion with Multiple Mobile Robots

In this section, the proposed approach has been tested for two and four mobile robots in an unknown environment. Figure 4.9 demonstrates the obstacle avoidance and goal seeking behavior performed by the two mobile robots during the navigation from its start position to goal position. The paths of the robots are shown in red and blue color. It is clear from the results that both the robots successfully reach their common goal position by tracking the collision free path. Simulation result designed for obstacle avoidance and goal seeking behavior with four mobile robots has been shown in the figure 4.10.

The robots have their own start locations and a pre-defined unique goal position. Paths followed by the robots are shown in different colors for better understanding of the result. Simulation results show that all the four robots are capable of avoiding the collision with the obstacles and with each other and finally achieve their goal position.

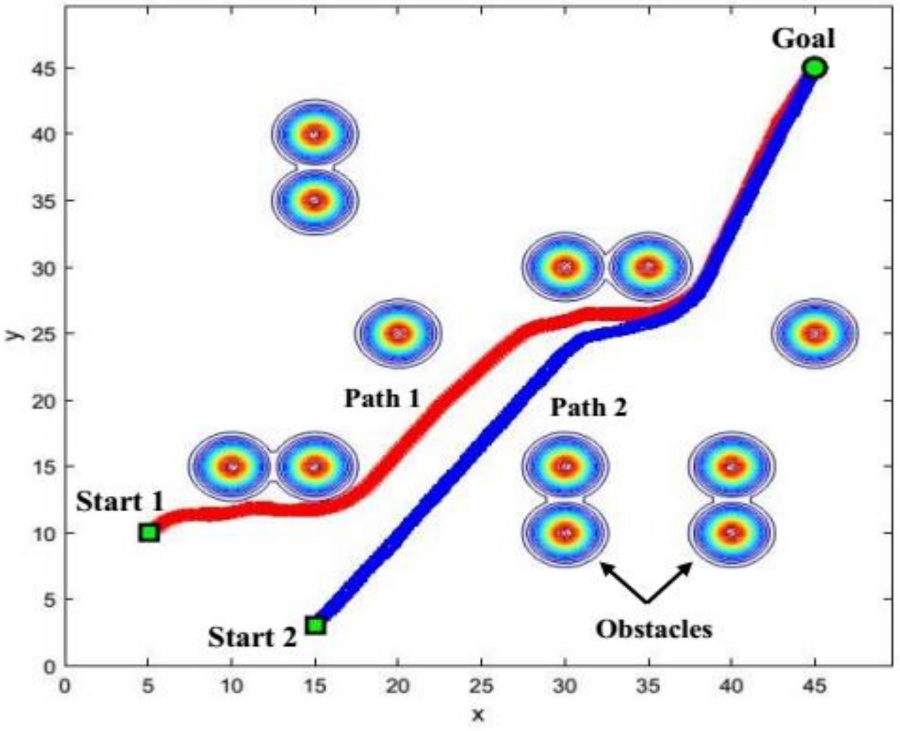


Figure 4.9 Obstacle avoidance and goal seeking by two robots in 2D work space

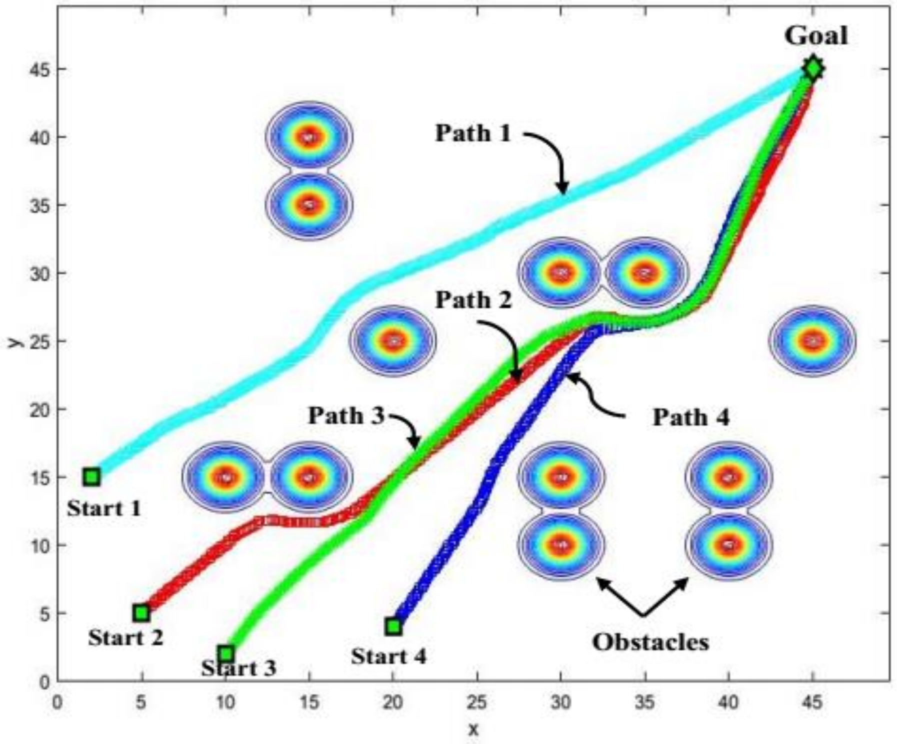


Figure 4.10 Obstacle avoidance and goal seeking by four robots in 2D work space

4.4 Mobile Robot Path Planning using Adaptive Bacterial Foraging Optimization (ABFO) Algorithm

In the classical BFOA, chemotaxis process can be termed as random search in which all the bacterium move with a constant step size in the map of nutrients. As a result, the random search may lead to delay in finding the global solution. To overcome this problem, the chemotactic step has been made adaptive to accelerate the convergence speed of the swarm of bacteria near global optima. Adaptive bacterial foraging optimization (ABFO) improves the performance of bacterial chemotaxis, which further speeds up the computation process. ABFO algorithm allows each individual bacterium to attain a fair balance between the exploration and exploitation state. In this method, each bacterium can have individual step-size and control its own search behavior by updating its current status. Thus, the step-size as well as the position of the each bacterium undergoes evolution.

In chemotaxis, each bacterium shows alternatively two distinct search states namely exploration and exploitation. In exploration state, bacterium uses large step-size to discover the unknown areas as soon as possible. On the other hand, in exploitation state bacterium uses comparatively small step-size to achieve the favorable areas gradually in its instantaneous locality. Each bacterium has to maintain a balance between these two states by changing its own step-size adaptively. The change in step-size of a bacterium depends upon its own fitness value. The conditions for selection of step size have been discussed below:

Condition 1 If the bacterium finds a new favorable region (or if fitness improves) the bacterium adapts smaller step size and bacterium’s behavior will self-adapt into exploitation state.

Condition 2 If the bacterium does not register any fitness improvement for a pre-defined iteration or it finds unfavorable region then it comes into exploration state by taking larger step size.

Now let us consider a robot which finds a collision free path between two positions by simulating the behavior of real bacterium. In the proposed approach, robot uses the bacterial chemotaxis mechanism to explore the environment and finally reaches the goal position without hitting any obstacles. During the navigation process the location of the robot can be evaluated as a multi objective cost function

$$f(\mathrm{x},\mathrm{y})_t = s_1 * f_{goal}(\mathrm{x},\mathrm{y})_t + s_2 * f_{obstacle}(\mathrm{x},\mathrm{y})_t \tag{4.16}$$

Where, $(\mathrm{x},\mathrm{y})_t$ denotes the position of the robot in the search environment at time t. $f_{goal}(\mathrm{x},\mathrm{y})_t$ is the goal function representing the distance between the goal position and the current position of the robot. Further, $f_{obstacle}(\mathrm{x},\mathrm{y})_t$ is the obstacle function which represents the distance between the current position of the robot and the nearest obstacle detected. s_1 and s_2 are the scaling parameter that specifies the relative importance of achieving obstacle avoidance and reaching the goal.

4.4.1 Simulation Results

In the present work, the proposed algorithm has been tested in a two dimensional square area of 50 square units. The goal is set as (45, 45), whereas the start point is taken as (5, 5). To define the goal function, a two dimensional sphere function has been used, which is given by;

$$\frac{1}{f_{goal}(\mathrm{x},\mathrm{y})_t} = \frac{1}{(\mathrm{x}-45)^2} + \frac{1}{(\mathrm{y}-45)^2}, \qquad \mathrm{x},\mathrm{y} \in [0,50]. \tag{4.17}$$

Therefore in each time step, the chemotaxis action leads the robot towards the goal position having the global minima. However, while moving towards the global minima, the robot may come across an obstacle. To avoid this situation, Gaussian functions have been used to define the obstacles. Thus obstacle function can be expressed as;

$$f_{obstacle}(\mathrm{x},\mathrm{y})_t = \frac{1}{\mathrm{maxexp}\{0.5(\mathrm{x}-\mathrm{x}^i_{obstacle})^2+(\mathrm{y}-\mathrm{y}^i_{obstacle})^2\}} \tag{4.18}$$

where, ($\mathrm{x}^i_{obstacle}$, $\mathrm{y}^i_{obstacle}$) is the coordinate of the *i*th obstacle. The use of maximum of all Gaussian functions confirms that each obstacle position is taken free from others. Figure 4.11 shows the robot path in 2D workspace using adaptive bacterial foraging optimization algorithm. 3D representation of the same result is shown in figure 4.12. Another simulation results has been presented below to show the effectiveness of the ABFO controller in the cluttered environment. The test platform is consist of four robots in a 2D cluttered environment consisting of 20 obstacles and two goal positions. Goal position 1 belongs to robot 1 and robot 2 whereas the robot 3 and robot 4 go to goal position 2.

The controller uses adaptive bacterial foraging optimization algorithm to generate the path from start to goal position by using the mathematical relation and criterion as discussed section 4.4.

The first and second intermediate position of all four robot during the navigation is shown in figure 4.13 and figure 4.14. The 2D and 3D representation of the final configuration of the robots as they reached their respective goal positions is shown in figure 4.15 and figure 4.16. Simulation results show that robots effectively reached their respective goal position by avoiding the obstacles in its path.

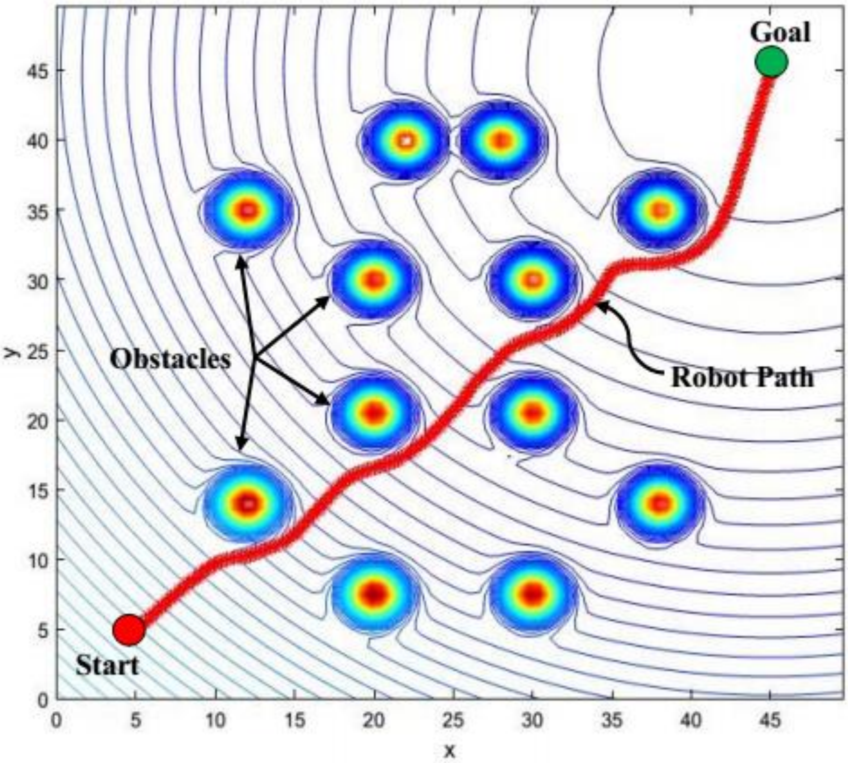


Figure 4.11 Robot path from start to goal point in 2D work space

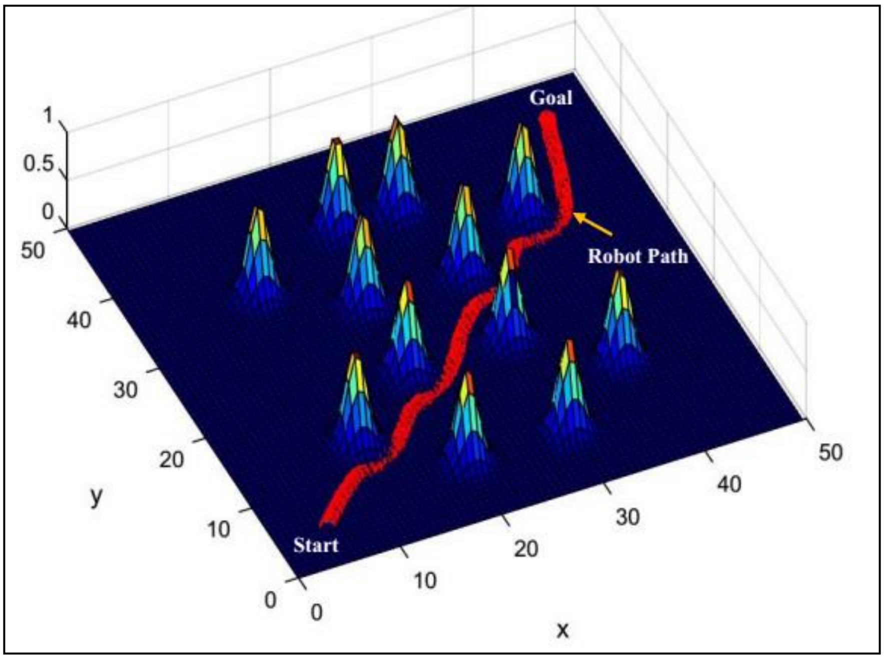


Figure 4.12 Robot path from start to goal point in 3D work space

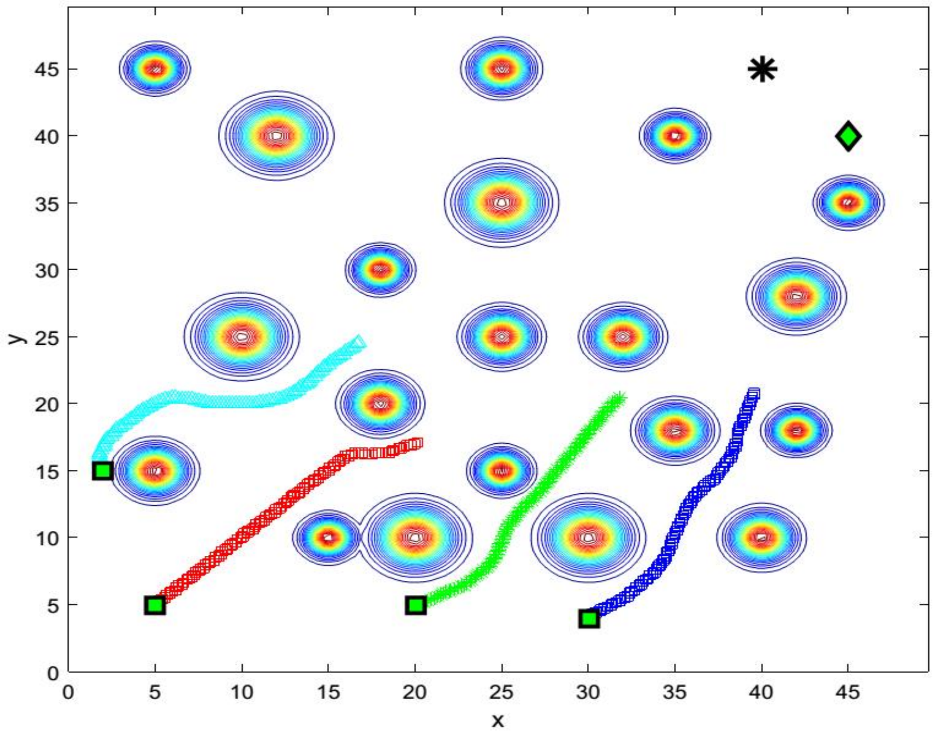


Figure 4.13 First intermediate position of the robots during the navigation

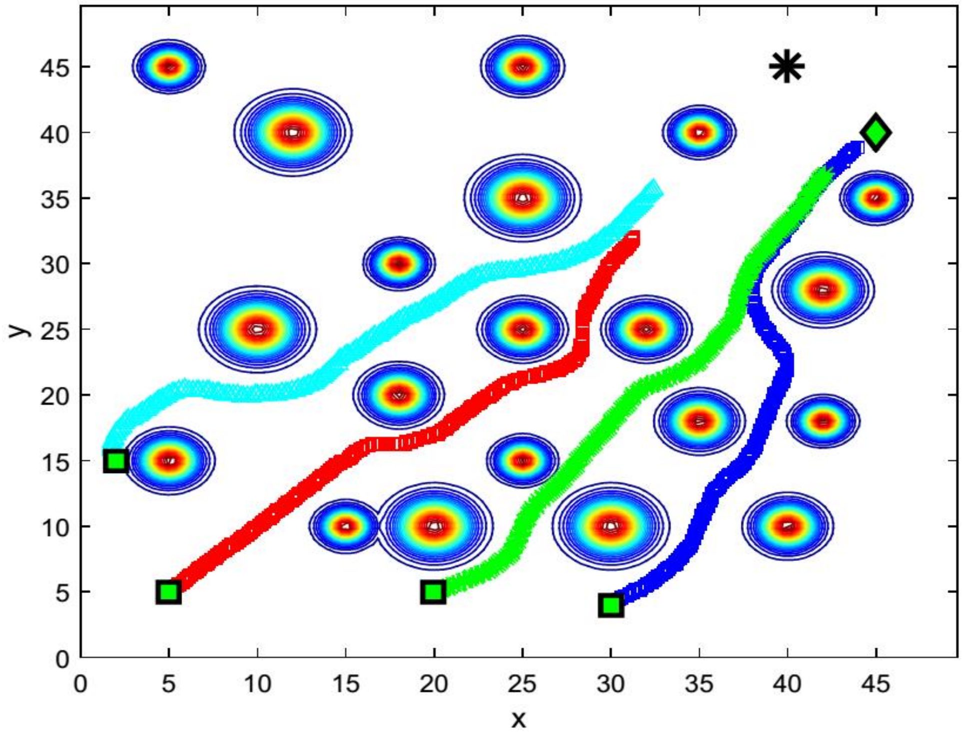


Figure 4.14 Second intermediate position of the robots during the navigation

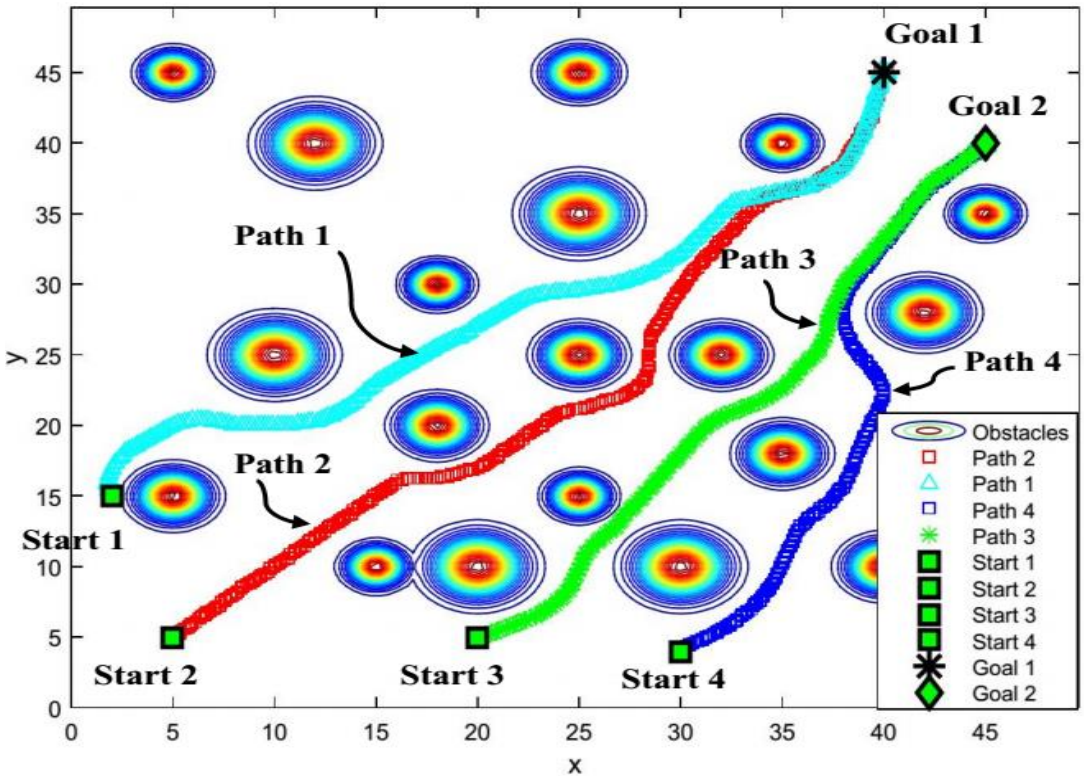


Figure 4.15 Paths followed by the four robots in 2D work space

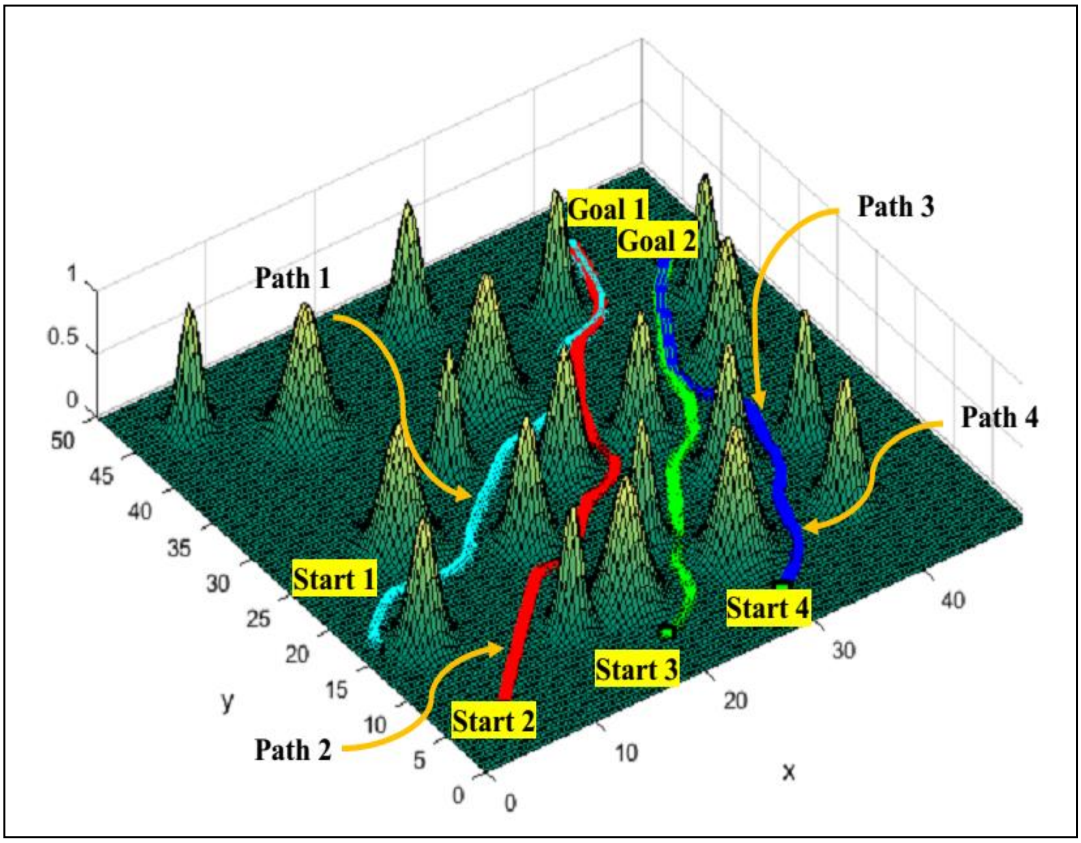


Figure 4.16 Paths followed by the four robots in 3D work space

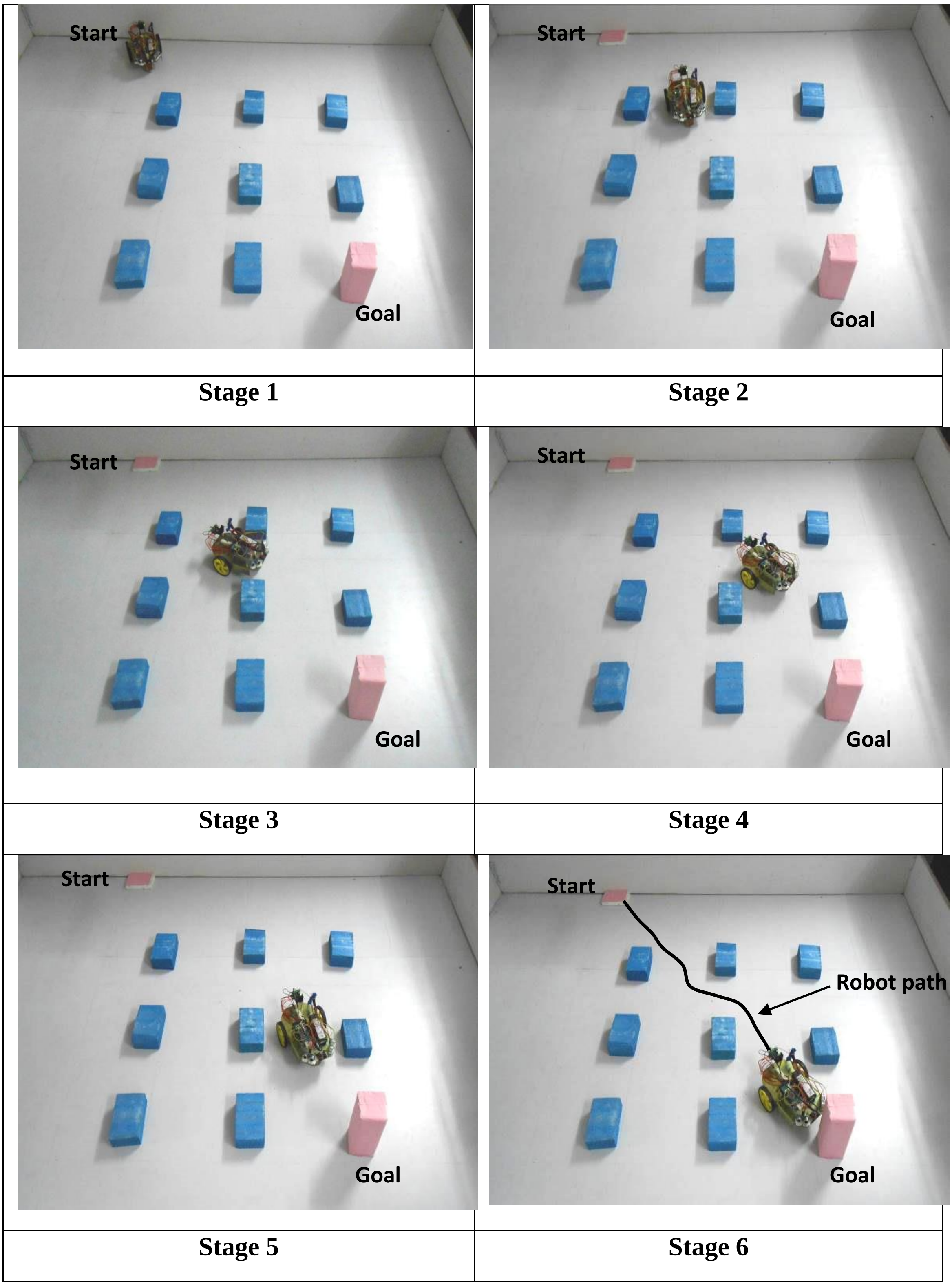
4.4.2 Experimental Results

In this section, the proposed controller has been tested in a series of experiments to illustrate its effectiveness. The experiments have been carried out using the in-house robot platform developed in our laboratory and Khepera robots. All experiments have been carried out by considering robot as a rigid body moving in a plane surface. The specifications of the robots are given in Appendix A.

Figure 4.17 shows a single robot navigating in an unknown environment using obstacle avoidance and goal seeking behavior. Figure 4.18 shows the wall following and goal seeking behavior by a single robot in an unknown environment. Further, exercises have been performed using multiple mobile robots to show the effectiveness of the proposed controller. The experimental results for two and four mobile robots in same scenario are shown in figure 4.19 and 4.20 respectively.

A comparative study has been presented in tabular form between the results obtained in simulation and real time with respect to the distance travelled and time taken by the robots to reach the goal position. Table 4.1 and Table 4.2 shows the comparison between the simulation and experimental results for distance travelled (in 'cm') and time taken (in 'sec') by a single mobile robot in ten different scenarios using BFOA and ABFO techniques. Table 4.3 and Table 4.5 presents the distance travelled by two and four mobile robots in ten different scenarios using BFOA and ABFO in simulation and real time. Similarly time taken to reach the goal by the robots are shown in table 4.4 and 4.6 respectively. The error analysis has also been presented in the respective tables to show the efficacy of the proposed navigation system.

It has been noticed that, the average percentage error is found within 6 percent for all above discussed cases in the present study.



Stage 1

Stage 2

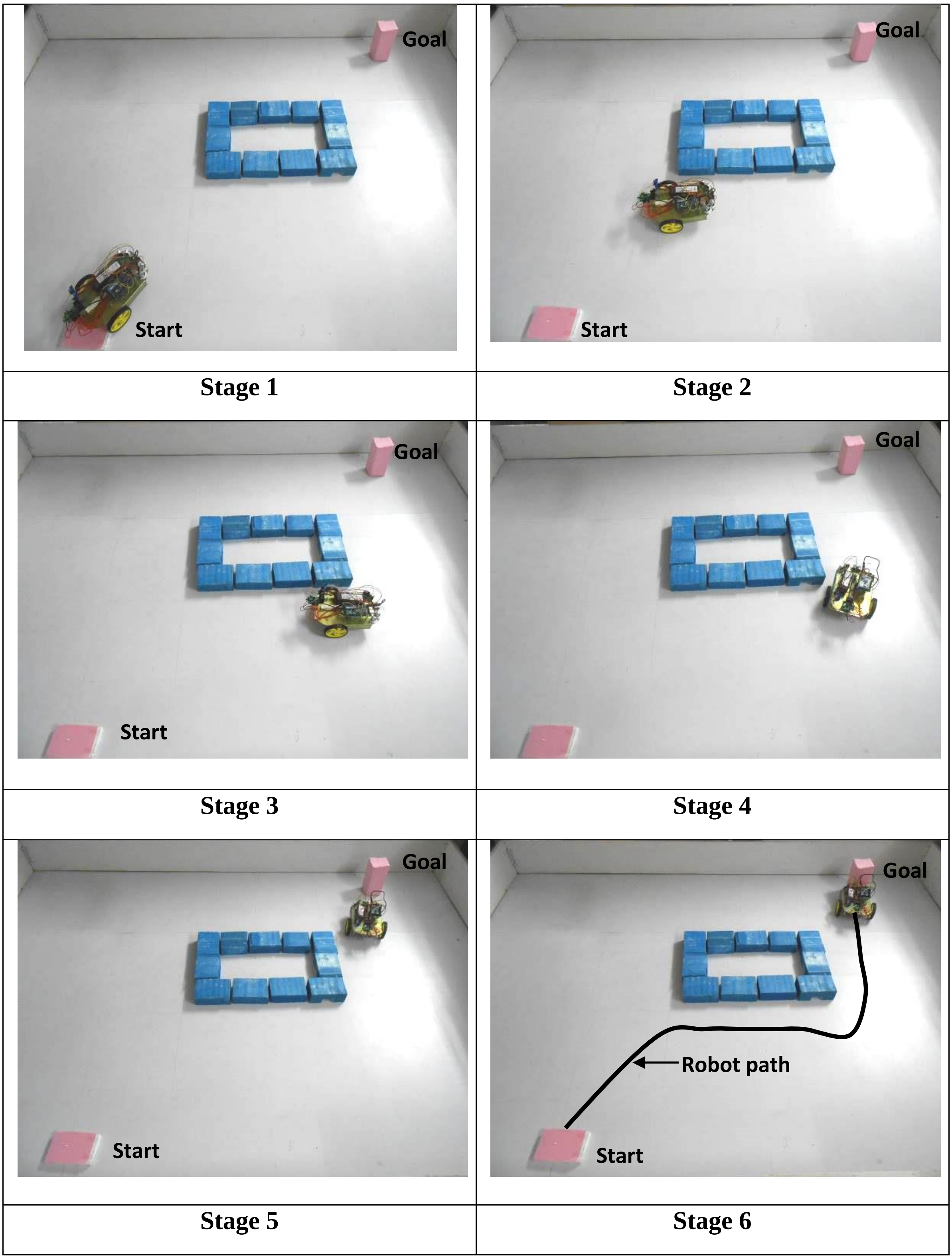
Stage 3

Stage 4

Stage 5

Stage 6

Figure 4.17 Robot path from start to goal position in real time (obstacle avoidance and goal seeking)



Stage 1

Stage 2

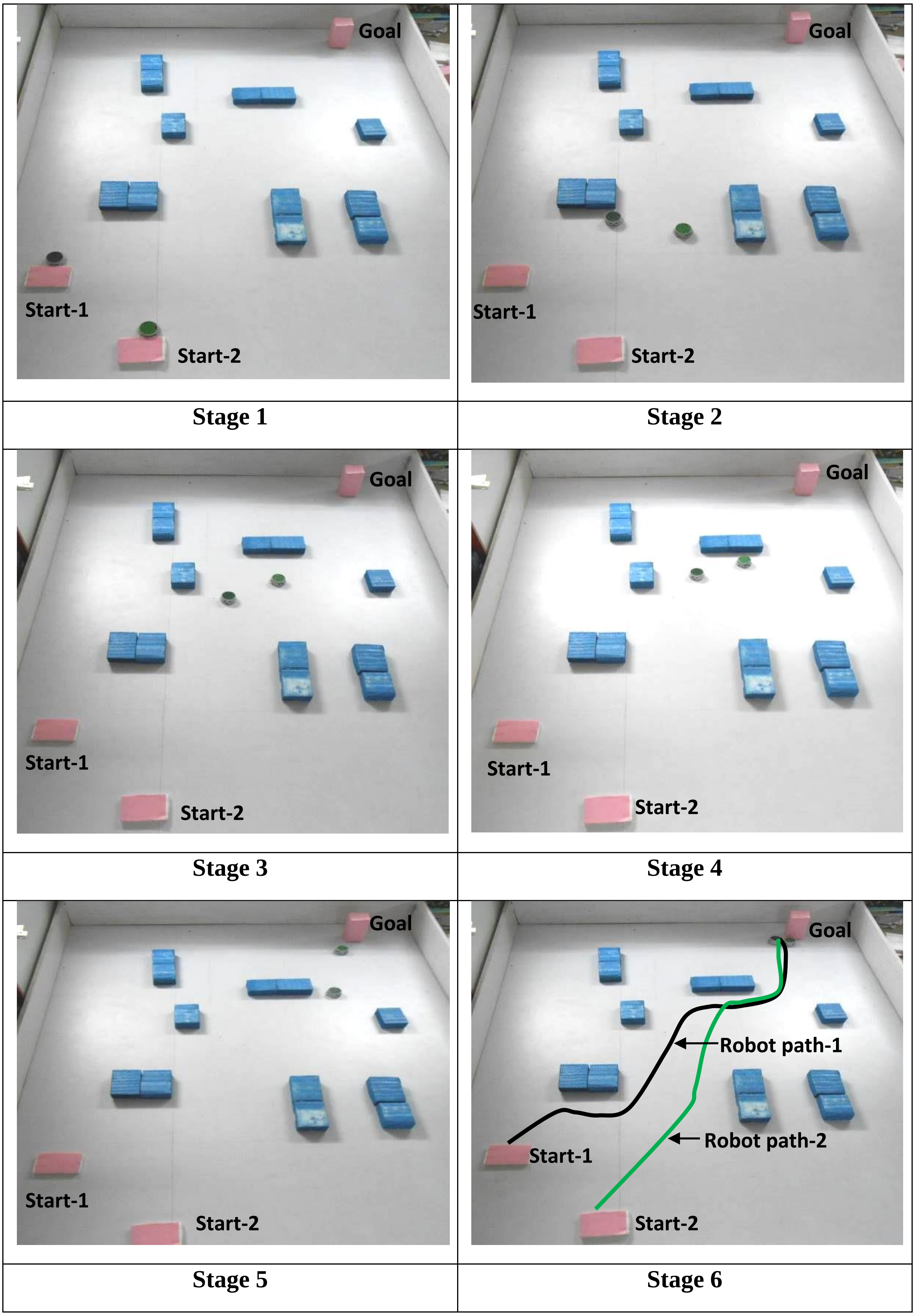
Stage 3

Stage 4

Stage 5

Stage 6

Figure 4.18 Robot path from start to goal position in real time (wall following and goal seeking)



Stage 1

Stage 2

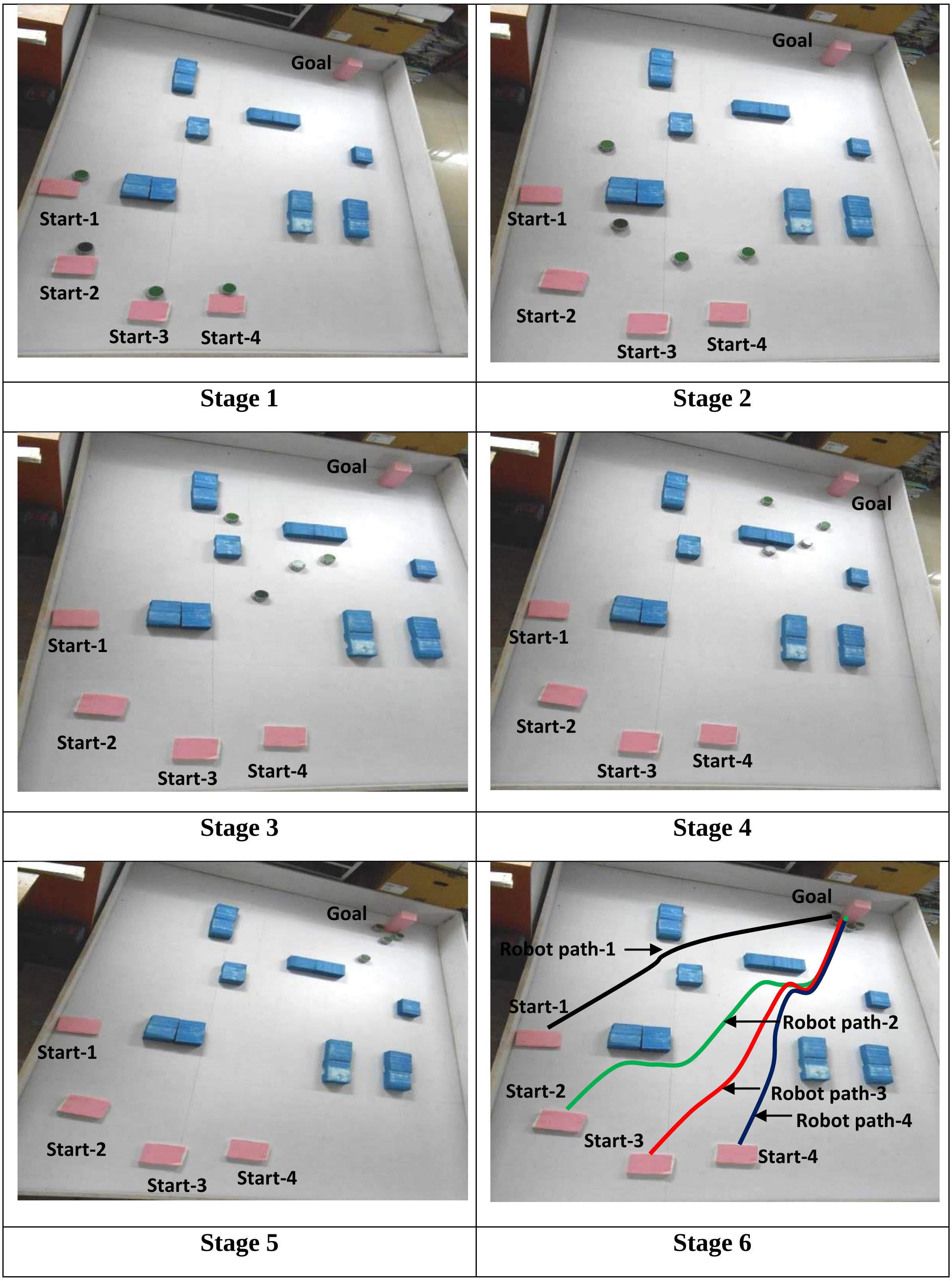
Stage 3

Stage 4

Stage 5

Stage 6

Figure 4.19 Path followed by the two robots in real time



Stage 1

Stage 2

Stage 3

Stage 4

Stage 5

Stage 6

Figure 4.20 Path followed by the four robots in real time

Table 4.1 Comparison of path length from start to goal position (in ‘cm’, one robot)

Scenario	In simulation(distance travelled in ‘cm’)		In real time (distance travelled in ‘cm’)		% Error		Avg. % Error	
	BFOA	ABFO	BFOA	ABFO	BFOA	ABFO	BFOA	ABFO
1	128.25	127.86	131.50	130.84	2.53	2.33	4.31	4.1
2	132.50	132.1	134.75	134.07	1.69	1.49		
3	164.50	164	170.15	169.29	3.43	3.23		
4	178.60	178.06	188.35	187.4	5.45	5.25		
5	228.45	227.76	237.60	236.4	4.00	3.80		
6	222.30	221.63	230.50	229.34	3.68	3.48		
7	258.50	257.72	276.20	274.81	4.77	6.63		
8	249.68	248.93	261.36	260.05	4.67	4.47		
9	195.45	194.86	207.45	206.41	6.13	5.93		
10	250.75	249.99	262.37	261.05	4.63	4.42		

Table 4.2 Comparison of time taken to reach from start to goal position (in ‘sec’, one robot)

Scenario	In simulation (time taken in ‘sec’)		In real time (time taken in ‘sec’)		% Error		Avg. % Error	
	BFOA	ABFO	BFOA	ABFO	BFOA	ABFO	BFOA	ABFO
1	14.2	13.92	14.6	14.27	2.82	2.55	4.07	3.8
2	14.7	14.41	15	14.66	2.04	1.78		
3	18.2	17.84	18.7	18.28	2.75	2.49		
4	19.8	19.40	20.5	20.04	3.54	3.27		
5	25.3	24.79	26.7	26.10	5.53	5.26		
6	24.7	24.21	25.8	25.22	4.45	4.19		
7	28.7	28.13	30.1	29.42	4.88	4.61		
8	27.8	27.24	29.2	28.54	5.04	4.77		
9	21.7	21.27	22.8	22.29	5.07	4.80		
10	26.4	25.87	27.6	26.98	4.55	4.28		

Table 4.3 Comparison of path length from start to goal position (in ‘cm’, two robots)

Scenario		In simulation (distance travelled in ‘cm’)		In real time (distance travelled in ‘cm’)		% Error	
		BFOA	ABFO	BFOA	ABFO	BFOA	ABFO
1	Robot 1	148.09	144.58	155.71	151.83	5.14	5.01
	Robot 2	121.37	116.98	128.64	123.88	5.99	5.90
2	Robot 1	144.51	141.08	152.73	148.93	5.69	5.56
	Robot 2	117.45	113.20	122.49	117.96	4.29	4.20
3	Robot 1	177.85	173.64	185.98	181.34	4.57	4.44
	Robot 2	148.29	142.92	154.38	148.67	4.11	4.02
4	Robot 1	169.31	165.29	178.23	173.79	5.27	5.14
	Robot 2	155.17	149.55	160.97	155.01	3.74	3.65
5	Robot 1	188.65	184.18	199.27	194.31	5.63	5.50
	Robot 2	163.53	157.61	172.09	165.72	5.23	5.15
6	Robot 1	210.26	205.28	218.02	212.59	3.69	3.56
	Robot 2	170.81	164.63	178.86	172.24	4.71	4.63
7	Robot 1	201.22	196.46	210.89	205.64	4.81	4.68
	Robot 2	142.94	137.77	151.25	145.65	5.81	5.73
8	Robot 1	218.81	213.62	227.03	221.38	3.76	3.63
	Robot 2	169.57	163.43	177.69	171.12	4.79	4.70
9	Robot 1	165.32	161.40	173.52	169.20	4.96	4.83
	Robot 2	134.49	129.62	140.52	135.32	4.48	4.40
10	Robot 1	171.04	166.99	179.03	174.57	4.67	4.54
	Robot 2	138.85	133.82	145.16	139.79	4.54	4.46

Table 4.4 Average error for travel time (in %, two robots)

Sl. NO.	Technique used	Average % Error	
		Robot 1	Robot 2
1	BFOA	4.82	4.77
2	ABFO	4.69	4.68

Table 4.5 Comparison of time taken to reach from start to goal position (in ‘sec’, two robots)

Scenario		In simulation (time taken in ‘sec’)		In real time (time taken in ‘sec’)		% Error	
		BFOA	ABFO	BFOA	ABFO	BFOA	ABFO
1	Robot 1	16.49	16.09	17.39	16.95	5.43	5.34
	Robot 2	13.80	13.48	14.35	13.99	3.97	3.75
2	Robot 1	16.19	15.79	17.02	16.60	5.16	5.07
	Robot 2	13.27	12.96	13.97	13.62	5.29	5.08
3	Robot 1	19.31	18.84	19.89	19.39	3.02	2.94
	Robot 2	16.51	16.13	17.33	16.90	4.98	4.76
4	Robot 1	18.18	17.73	19.09	18.61	5.02	4.93
	Robot 2	17.69	17.28	18.30	17.84	3.47	3.26
5	Robot 1	20.29	19.79	21.39	20.85	5.42	5.34
	Robot 2	18.28	17.86	19.28	18.80	5.46	5.25
6	Robot 1	22.76	22.20	23.56	22.96	3.51	3.43
	Robot 2	19.12	18.68	19.75	19.25	3.29	3.07
7	Robot 1	22.22	21.68	23.09	22.51	3.92	3.84
	Robot 2	16.31	15.93	17.08	16.65	4.73	4.52
8	Robot 1	23.72	23.14	24.59	23.98	3.70	3.62
	Robot 2	19.15	18.71	20.02	19.52	4.56	4.34
9	Robot 1	18.40	17.95	19.16	18.68	4.12	4.04
	Robot 2	15.23	14.88	15.88	15.48	4.22	4.01
10	Robot 1	18.94	18.48	19.71	19.21	4.05	3.97
	Robot 2	15.72	15.36	16.37	15.96	4.08	3.87

Table 4.6 Average error for travel time (in %, two robots, two robots)

Sl. NO.	Technique used	Average % Error	
		Robot 1	Robot 2
1	BFOA	4.81	4.40
2	ABFO	4.25	4.19

Table 4.7 Comparison of path length from start to goal position (in ‘cm’, four robots)

Scenario		In simulation (distance travelled in ‘cm’)		In real time (distance travelled in ‘cm’)		% Error	
		BFOA	ABFO	BFOA	ABFO	BFOA	ABFO
1	Robot 1	130.36	127.62	137.35	134.19	5.36	5.15
	Robot 2	138.84	134.82	145.50	140.84	4.80	4.47
	Robot 3	150.35	144.94	157.14	151.33	4.51	4.40
	Robot 4	110.85	108.48	116.41	114.20	5.02	5.27
2	Robot 1	122.61	120.04	126.98	124.06	3.56	3.35
	Robot 2	149.07	144.75	155.77	150.78	4.49	4.17
	Robot 3	144.75	139.54	151.75	146.13	4.83	4.72
	Robot 4	125.45	123.23	129.48	127.02	3.21	3.08
3	Robot 1	108.20	105.93	114.53	111.89	5.85	5.63
	Robot 2	144.74	140.54	149.12	144.34	3.03	2.71
	Robot 3	156.39	150.76	163.11	157.07	4.30	4.19
	Robot 4	117.71	115.55	122.76	120.43	4.29	4.22
4	Robot 1	112.47	110.11	117.76	115.05	4.70	4.49
	Robot 2	141.39	137.29	147.59	142.86	4.38	4.06
	Robot 3	163.86	157.96	171.06	164.73	4.40	4.29
	Robot 4	114.38	112.25	118.79	116.53	3.85	3.82
5	Robot 1	126.23	123.58	133.12	130.06	5.46	5.24
	Robot 2	144.17	139.99	151.96	147.10	5.41	5.08
	Robot 3	151.33	145.89	155.28	149.53	2.61	2.50
	Robot 4	113.15	113.15	117.57	115.34	3.91	1.93

Table 4.8 Average error for travel time (in %, four robots)

Sl. NO.	Technique used	Average % Error			
		Robot 1	Robot 2	Robot 3	Robot 4
1	BFOA	4.99	4.42	4.13	4.06
2	ABFO	4.77	4.10	4.02	3.67

Table 4.9 Comparison of time taken to reach from start to goal position (in ‘sec’, four robots)

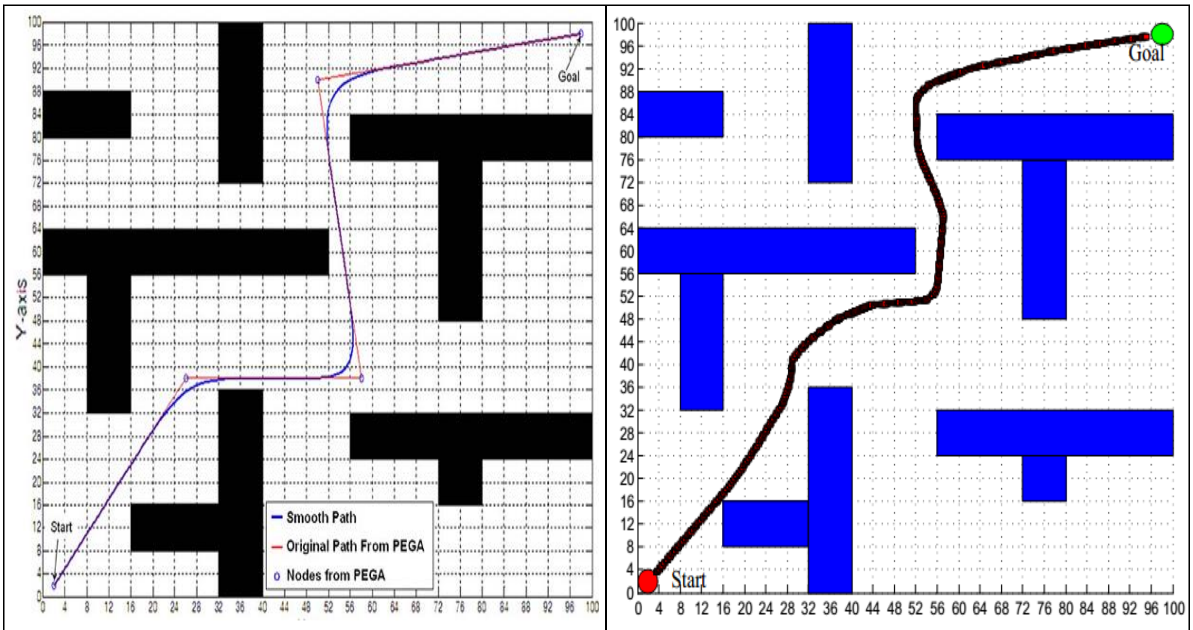
Scenario		In simulation (time taken in ‘sec’)		In real time (time taken in ‘sec’)		% Error	
		BFOA	ABFO	BFOA	ABFO	BFOA	ABFO
1	Robot 1	11.94	11.70	12.62	12.36	5.71	5.64
	Robot 2	14.13	13.84	14.85	14.54	5.14	5.10
	Robot 3	14.41	13.98	15.14	14.61	5.02	4.48
	Robot 4	12.89	12.45	13.51	13.04	4.79	4.68
2	Robot 1	13.70	13.43	14.32	14.03	4.53	4.46
	Robot 2	14.49	14.19	15.02	14.71	3.70	3.66
	Robot 3	15.55	15.08	16.02	15.46	3.06	2.53
	Robot 4	11.61	11.22	12.24	11.81	5.40	5.30
3	Robot 1	13.33	13.07	14.09	13.80	5.64	5.58
	Robot 2	14.55	14.25	15.16	14.84	4.15	4.10
	Robot 3	15.88	15.40	16.48	15.90	3.77	3.23
	Robot 4	11.61	12.34	13.31	12.84	4.12	4.01
4	Robot 1	12.91	12.65	13.53	13.25	4.80	4.74
	Robot 2	14.24	13.95	15.01	14.70	5.44	5.39
	Robot 3	15.61	15.14	16.21	15.64	3.84	3.31
	Robot 4	11.71	11.31	12.21	11.78	4.32	4.22
5	Robot 1	13.08	12.82	13.60	13.32	3.96	3.90
	Robot 2	14.47	14.17	15.27	14.95	5.55	5.51
	Robot 3	15.61	14.81	15.86	15.31	3.88	3.35
	Robot 4	12.26	11.85	12.85	12.40	4.79	4.68

Table 4.10 Average error for travel time (in %, four robots)

Sl. NO.	Technique used	Average % Error			
		Robot 1	Robot 2	Robot 3	Robot 4
1	BFOA	4.93	4.80	3.91	4.69
2	ABFO	4.86	4.75	3.38	4.58

4.5 Comparison with Other Results

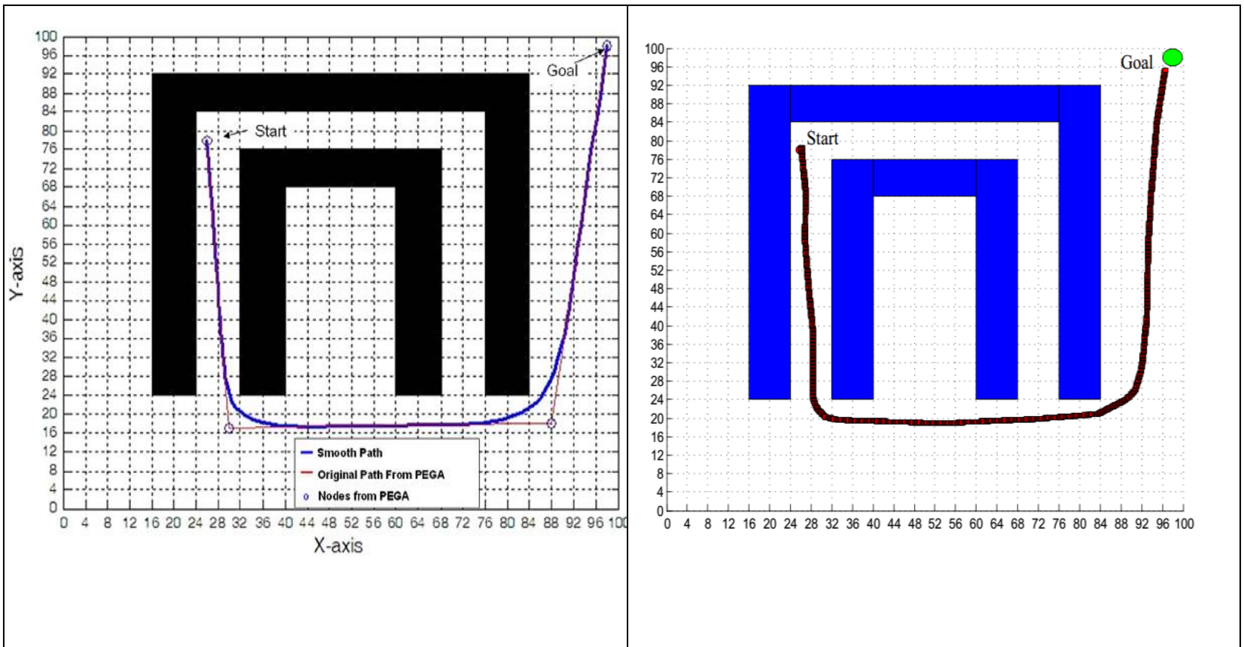
This section presents the comparison between the developed ABFO controllers in the present study with the other models presented in the literature. In particular, ABFO controller has been compared with the results given by Tsai et al. [147] and Qu et al. [148]. Figure 4.21 (a)-(b) to figure 4.25 (a)-(b) show the simulation result for ABFO controller and results found by Tsai et al. [147] and Qu et al. [148]. Comparison of path length is shown in Tables 4.11-4.15.



(a) (b)

Figure 4.21 (a) Simulation result by Tsai et al. [147]

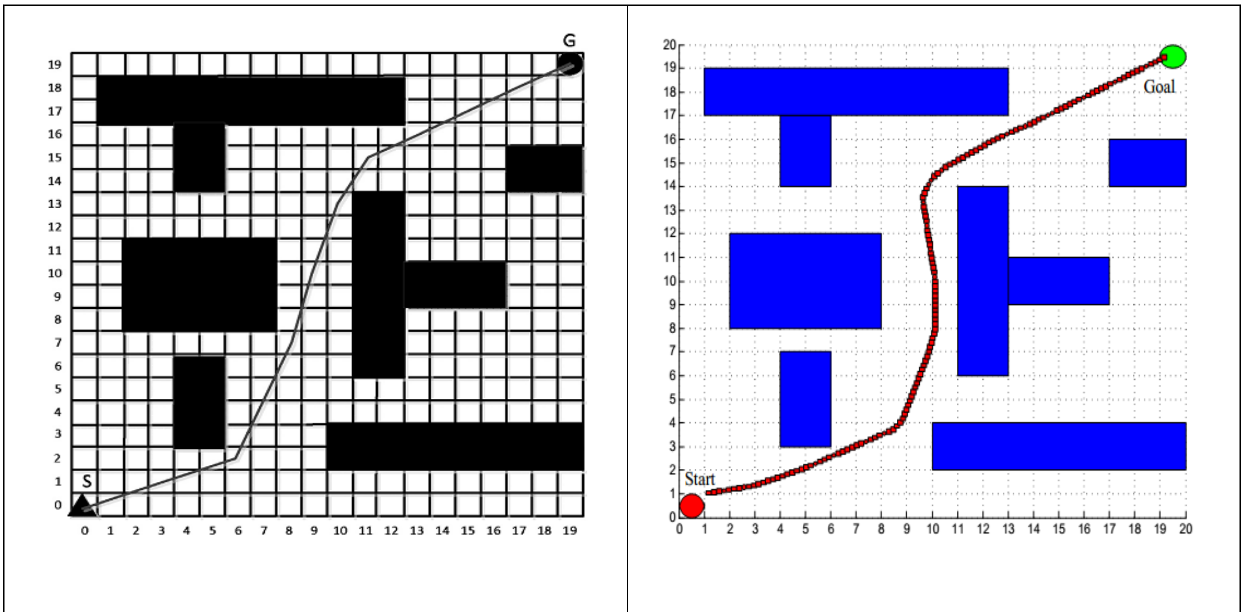
(b) Simulation result by the proposed ABFO controller



(a) (b)

Figure 4.22(a) Simulation result by Tsai et al. [147]

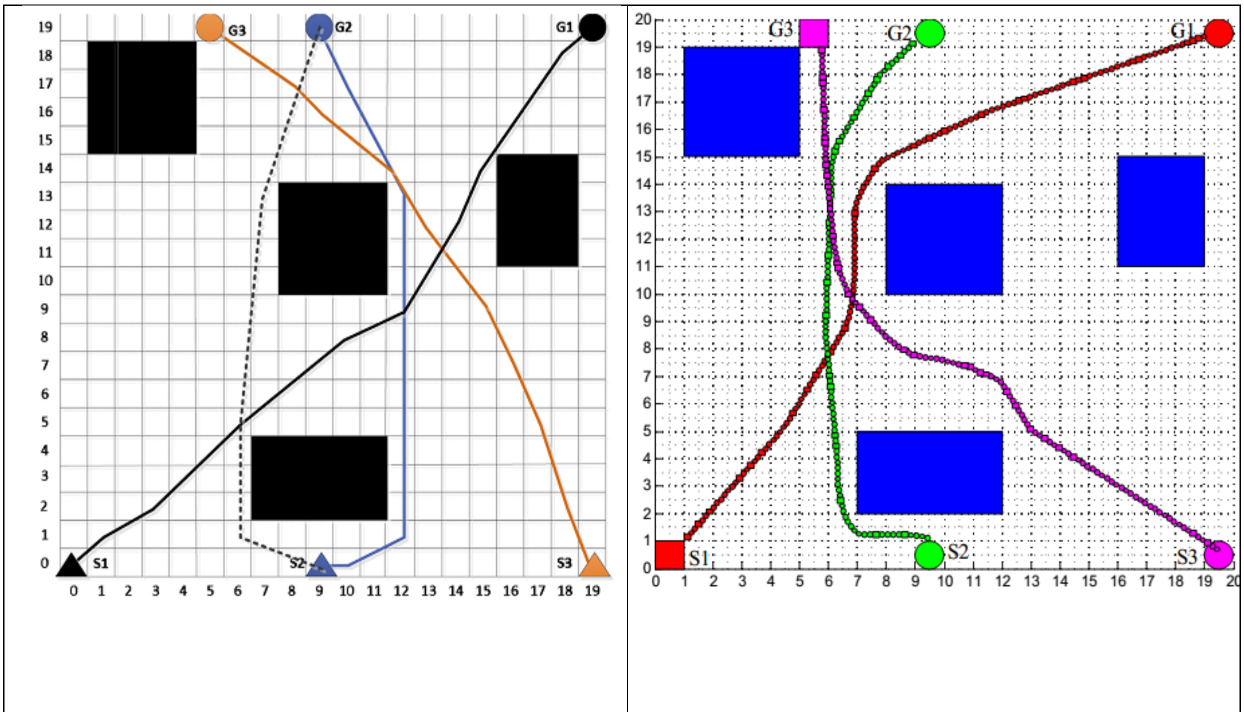
(b) Simulation result by the proposed ABFO controller



(a) (b)

Figure 4.23 (a) Simulation result by Qu et al. [148]

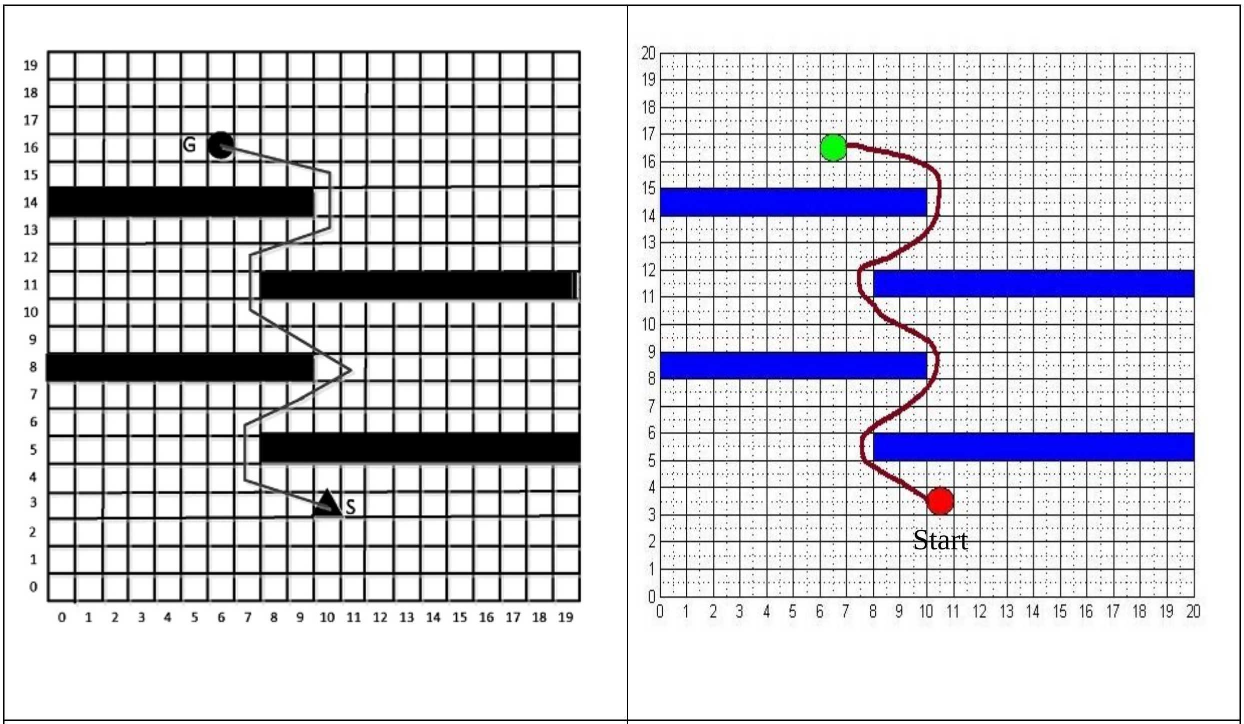
(b) Simulation result by the proposed ABFO controller



(a) (b)

Figure 4.24 (a) Simulation result by Qu et al. [148]

(b) Simulation result by the proposed ABFO controller



(a) (b)

Figure 4.25 (a) Simulation result by Qu et al. [148]

(b) Simulation result by the proposed ABFO controller

Table 4.11 Comparison of results between Tsai et al. [147] and developed ABFO controller.

Technique Used	Path length (in cm)	Deviation (in %)
Figure 4.21 (a)	23.92	1.78
Figure 4.21 (b)	23.50	

Table 4.12 Comparison of results between Tsai et al. [147] and developed ABFO controller.

Figure	Path length (in cm)	Deviation (in %)
Figure 4.22 (a)	24.06	2.82
Figure 4.22 (b)	23.40	

Table 4.13 Comparison of results between Qu et al. [148] and developed ABFO controller.

Figure	Path length (in cm)	Deviation (in %)
Figure 4.23 (a)	23.34	0.95
Figure 4.23 (b)	23.12	

Table 4.14 Comparison of results between Qu et al. [148] and developed ABFO controller.

Figure	Path length (in cm)		
	Robot1	Robot2	Robot3
Figure 4.24 (a)	20.44	15.84	17.7
Figure 4.24 (b)	20.12	15.35	17.45
Deviation (in %)	1.59	3.19	1.43

Table 4.15 Comparison of results between Qu et al. [148] and developed ABFO controller.

Figure	Path length (in cm)	Deviation (in %)
Figure 4.25 (a)	24.68	6.74
Figure 4.25 (b)	23.12	

From the results obtained, it can be noticed that the proposed controller provides smooth path from start to goal position as compared to results obtained by Tsai et al. [147] and Qu et al. [148]. Moreover, path length is found to be less in simulation results obtained by proposed ABFO controller as compared to the results by Tsai et al. [147] and Qu et al. [148].

4.6 Summary

In this chapter, an intelligent controller inspired from the bacterial foraging optimization algorithm (BFOA) has been designed, analyzed and tested for a number of situations to show the effectiveness of the proposed BFOA controller. Further, adaptive bacterial foraging optimization algorithm (ABFO) is analyzed to solve the problem of mobile robot navigation. Moreover, a comparison is made between the two approaches (i.e. BFOA and ABFO) on the basis of path length and time taken during the navigation. The ABFO algorithm is found to be more effective than BFO algorithm while used for robot navigation. The percentage of errors between the simulation and experimental results are found to be within 6 %. Further, proposed ABFO controller has been compared with other models presented in the literature. The ABFO controller provides smooth path and the path length is found to be less as compared to other models by Tsai et al. [147] and Qu et al. [148]

Chapter 5

ANALYSIS OF RBF NEURAL CONTROLLER FOR MOBILE ROBOT NAVIGATION

5.1 Introduction

In this chapter, radial basis function (RBF) neural network has been used for the navigation of mobile robots. The proposed intelligent controller enables the robot to successfully navigate in the real world environment and ensure that the robot reaches the target position by avoiding the obstacles coming across its path. Distance of the obstacles in front, left and right with respect to robot's position are used as input parameters to the RBF neural controller. Moreover, orientation of the target with respect to robot (target angle) has been given as another input to the controller. Steering angle is the only output of the proposed controller which guides the robot towards the target position. Simulation results are presented to show the effectiveness of the proposed controller. Furthermore, simulation results are compared with experimental results to validate the performance of the proposed controller in real environment.

5.2 Cause and Problem Formulation for Considering RBFN

The radial basis function (RBF) developed by Powell [144] is a feed forward neural network generally used for classification and function approximation. This network mostly uses Gaussian and Sigmoidal activation function in the hidden layer. These networks are extensively used for nonlinear function approximations, time series prediction, classification and system control. The radial basis function network shows its effectiveness over multi-layer perceptron (MLP) network as it takes lesser time for training as compared to MLP. Moreover, to reach same level of accuracy as MLP, RBFN has more nodes in the hidden layer and hence becomes complex. The popularity of a radial basis function network is due to its simple architecture with a single hidden layer, which in particular is admirable in applications requiring locally tunable (or adjustable) properties. As the name implies, this network makes use of radial functions to represent an input in terms of radial centers. However, this network is very attractive for intelligent control applications since the response of this network is linear in terms of weights. Thus, weight update rules for such networks within intelligent control paradigm become easy to derive.

As RBFN requires lesser time for training as compared to multi-layer perceptron (MLP), it can be used for training of mobile robot as the robot needs to take decision based on the sensory data received during the navigation.

In the present research large data set is required to train the network and RBF network takes lesser time to train the network therefore RBF network is used in the current research work. Moreover, RBFN can also be hybridized with the other approaches to develop hybrid controllers.

5.3 Structure of Radial Basis Function Network (RBFN)

Radial basis function neural network consists of three layers namely input layer, hidden layer and output layer with feed forward architecture. Figure 5.1 shows the general architecture of a radial basis function network. The popularity of a radial basis function network is due to its simple architecture with a single hidden layer, which in particular is admirable in applications requiring locally tunable (or adjustable) properties. As the name implies, this network makes use of radial functions to represent an input in terms of radial centers. However, this network is very attractive for intelligent control applications since the response of this network is linear in terms of weights. Thus, weight update rules for such networks within intelligent control paradigm become easy to derive.

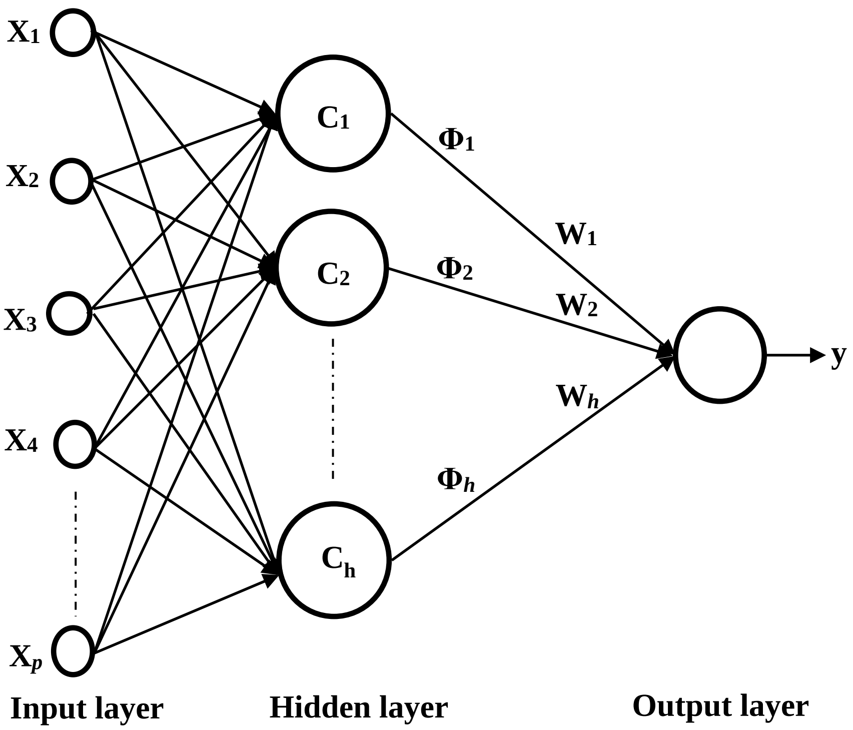


Figure 5.1 General architecture of radial basis function network

The architecture of radial basis function network is a multilayer feed forward network. The number of neurons in the input and output layer depends on the number of input and output parameters. The interconnection between the input layer and hidden layer forms hypothetical connection and between the hidden and output layer forms weighted connections. The training algorithm is used for update of weights in all the connections. The output is linear in terms of weights (w_i) and can be expressed as;

$$y = \sum_{i=1}^{h} \phi_i w_i = \sum_{i=1}^{h} w_i \phi(\|x - c_i\|) \tag{5.1}$$

Gaussian function is most commonly used activation function and is given by;

$$\phi_i(\|x - c_i\|) = e^{-\left(\frac{\|x-c_i\|}{\sigma_i}\right)^2} \tag{5.2}$$

Where, σ_i is the width factor (spread) of kernel *i*, w_i , (*i*=1,2…,*h*) are the weights and ϕ_i ,

(i=1,2…,h) are the activation functions or radial basis functions. If the activation functions are Gaussian, they yield same outputs for inputs inside a fixed circular space from the center, i.e., they are radially symmetric, this reports for the term ‘radial basis function’. Moreover, $\|x-c_i\|$ is the Euclidian distance of the input x from the center c.

5.4 Analysis of RBFN for Mobile Robot Navigation

As already specified that radial basis function network can be used in intelligent control application, a controller has been proposed and implemented for the navigation of mobile robots. Figure 5.2 shows the architecture used for the proposed RBFN controller. The control architecture has four inputs and one output. The basic inputs to the radial basis function neural controller are front obstacle distance (FOD), left obstacle distance (LOD) and right obstacle distance (ROD) and target angle (TA). Steering angle (SA) is the only output.

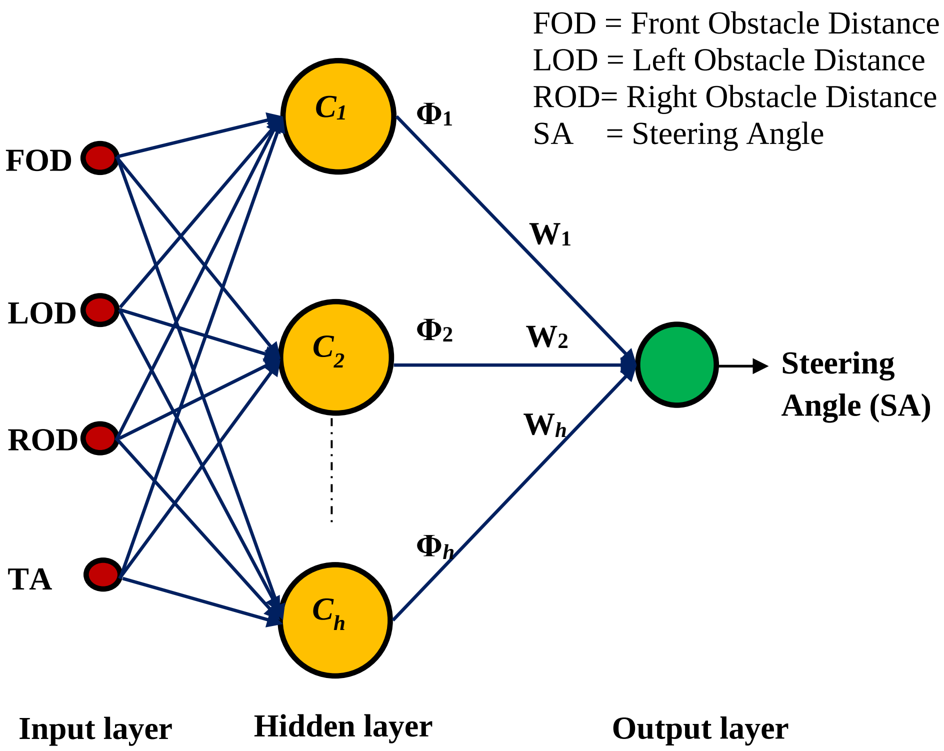


Figure 5.2 Architecture of proposed RBF neural controller

5.4.1 Training of RBF Network

Mobile robot must be trained to navigate freely without human assistance. There are certain behavior which can be required during the navigation of the mobile robot to successfully reach the destination by following a collision free path. Such behavior are avoiding the obstacles; following the wall, and seeking the target.

Training of RBF network simply means to find the value of parameters such as centers (c_i) spreads (σ_i) and weights (w_i) to get the best results. In conventional RBF training, all three parameters are computed independently. Though the method does not find the global optima with respect to all parameters but splitting makes the learning much easier than a multi-layer perceptron and is the one of the major advantages of the RBF network.

Different learning algorithm can be used for learning of RBF network parameters. The proper selection of three main parameters is very important.

1. Selection of centers of the RBF activation functions.
2. The spread of the Gaussian RBF activation functions.
3. The weights from the hidden to the output layer.

Locations of centroids are generally chosen according to the probability density of the input set. Width factors for the Gaussian activation function can be computed by minimizing a cost function measuring the overlapping between the adjacent units. Once the parameters c_i and σ_i are fixed then the weight parameter can be easily find by using pseudo-inverse or gradient decent method.

Since the objective of the present work is to design a controller which can guide the mobile robot in an unknown environment consisting of obstacles, the proposed controller has been trained for number of configurations. In particular, the robot has been trained to cope up with circumstances that may arise during the course of its motion from a start position to its goal. Figure 5.3 shows the training pattern for the navigation of a mobile robot. In order to navigate independently (without human assistance), robot has to make intelligent decisions online based on the information received by the sensors mounted on it. In order to navigate, the robot should adapt some essential behavior such as obstacle avoidance, following a wall, and goal seeking.

5.4.1.1 Obstacle Avoidance

Avoiding the obstacle is the prime necessity of any autonomous robot. As the robot starts moving its search for target and heading towards until it meets any obstacle. Now consider a situation when the robot is surrounded by the obstacles while traveling towards the target. For example if in a particular instance, robot comes across obstacles in front and left directions, then it has to steer in right side with some appropriate angle to avoid collision with the obstacles. Rules are constructed to train the controller to perform this behavior whenever required in the course of motion.

5.4.1.2 Wall Following

During navigation, if robot detects any obstacles in the front while heading towards the target along the left or right side of the wall, the robot adopts the wall following behavior. Wall following behavior is to move parallel to and maintain an offset from the obstacle arranged like a wall. The robot's goal is to stay at a given distance from the "wall" as it moves. For this the robot sets its heading direction perpendicular to the wall and maintains a safe distance from the wall.

5.4.1.3 Goal Seeking

Let us consider a situation in which a mobile robot has to travel from one start position to a pre-defined goal position. Then it should be able to discover the goal position and approach straightway towards the goal by following the goal seeking behavior.

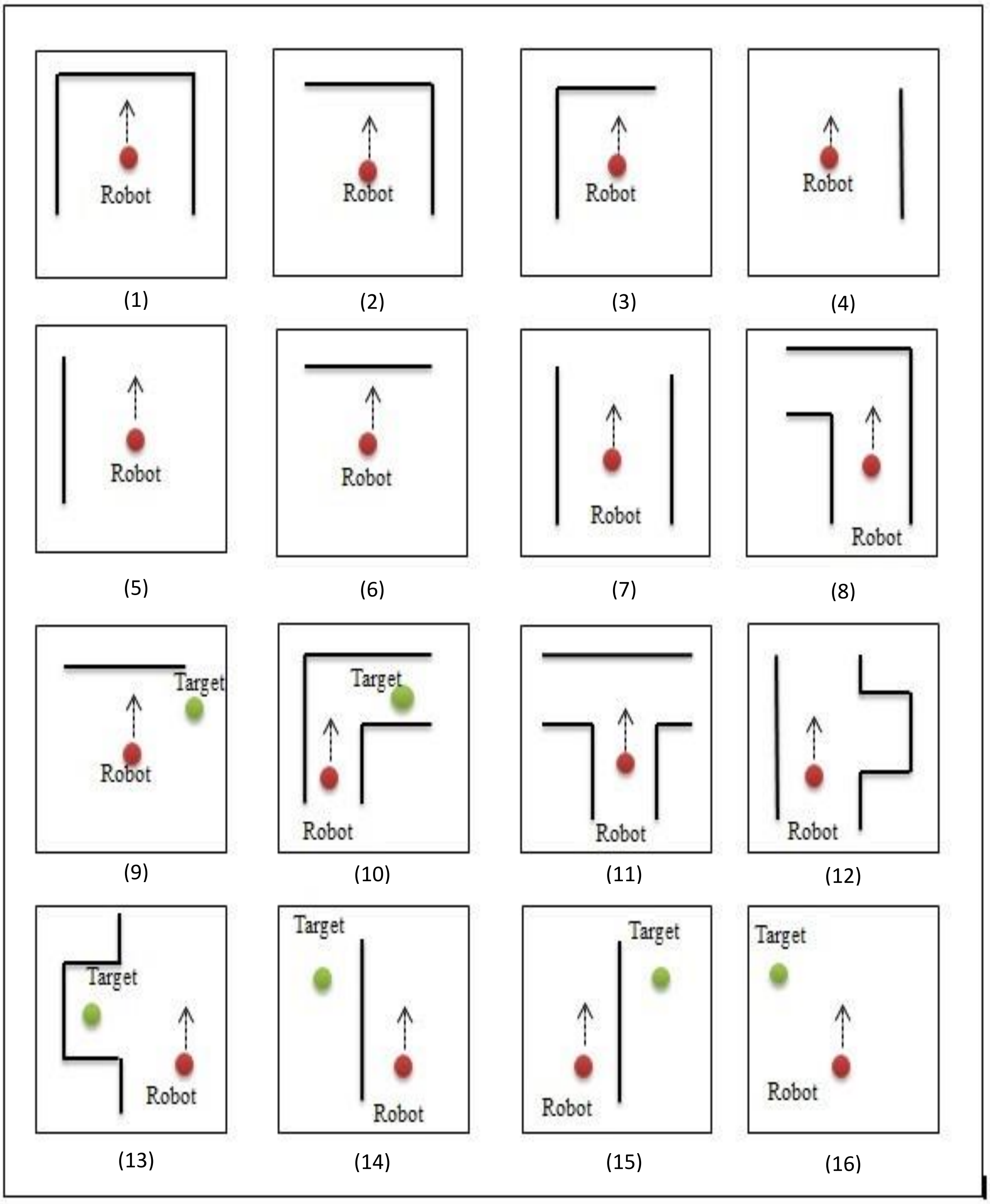


Figure 5.3 Training patterns for the proposed RBFN controller

This behavior is required for the robot to simply follow the goal position if the robot is free from the obstacles. Goal seeking behavior is opted by the robot when it is free from the obstacles within its sensing range while approaching towards the goal. Figure 5.3 shows the training pattern used for the navigation of the mobile robots. Training patterns are designed in such a manner that the robot can navigate using certain behavior as discussed above. The rules corresponding to each training pattern are given in table 5.1. As the robot moves, it acquires information from the environment in the form of distance of the goal and obstacles with respect its current position. After each iteration, the robot uses training pattern and rules to correct its steering angle to lead towards the goal position. For example, let us consider the first training pattern where the robot detects obstacles in front, right and left side of the robot. The distance of the nearest obstacles from the robot in all three directions can be seen in table 5.1.

Table 5.1 Training data for the proposed RBFN controller

S NO.	FOD	ROD	LOD	TA	SA
1	20	10	10	0	-180
2	20	10	50	0	-90
3	10	50	10	0	90
4	50	10	50	0	0
5	50	50	10	0	0
6	10	50	50	0	90
7	50	10	10	0	0
8	20	10	10	0	-20
9	20	50	50	35	30
10	20	5	10	45	20
11	20	10	10	0	5
12	50	30	5	0	0
13	50	50	20	-65	-45
14	50	50	10	-45	-10
15	50	10	50	30	5
16	50	50	50	-30	-30

Target angle is used to define the position of the goal with respect to the robot's main axis. In the above situation, as per the rule, the robot reverses its motion by 180 degrees in the positive direction. Here positive and negative sign are used to define the bearing of the robot's steering in right and left direction with respect to robot heading axis respectively.

5.5 Simulation Results

This section comprises of the simulation results of the proposed RBF neural controller used for navigation of mobile robots in various environments. The experiments are conducted in simulated environment using in-house software package. A series of experiments have been conducted to show the effectiveness of the proposed controller.

The environment considered for the simulation is a maze environment having static obstacles. Various trial runs have been performed to obtain the near optimal path with smooth trajectory. The proposed controller has been tested in different environment for both single and multiple robots. In general, the program module allows generating any number of robot and obstacles and goal within the workspace. The controller has been tested for a number of combinations such as single robot with single target, multiple robots with sole target and multiple robots with multiple target positions.

As discussed in the previous section, a number of exercises are presented to show the different behavior opted by the mobile robots during the navigation. In general, mobile robot has to follow a combination of two or more behavior subjected to the corresponding situation.

5.5.1 Obstacle Avoidance and Goal Seeking Behavior in an Unknown Environment

The test environment for obstacle avoidance and goal seeking for a mobile robot has been shown in figure 5.4.

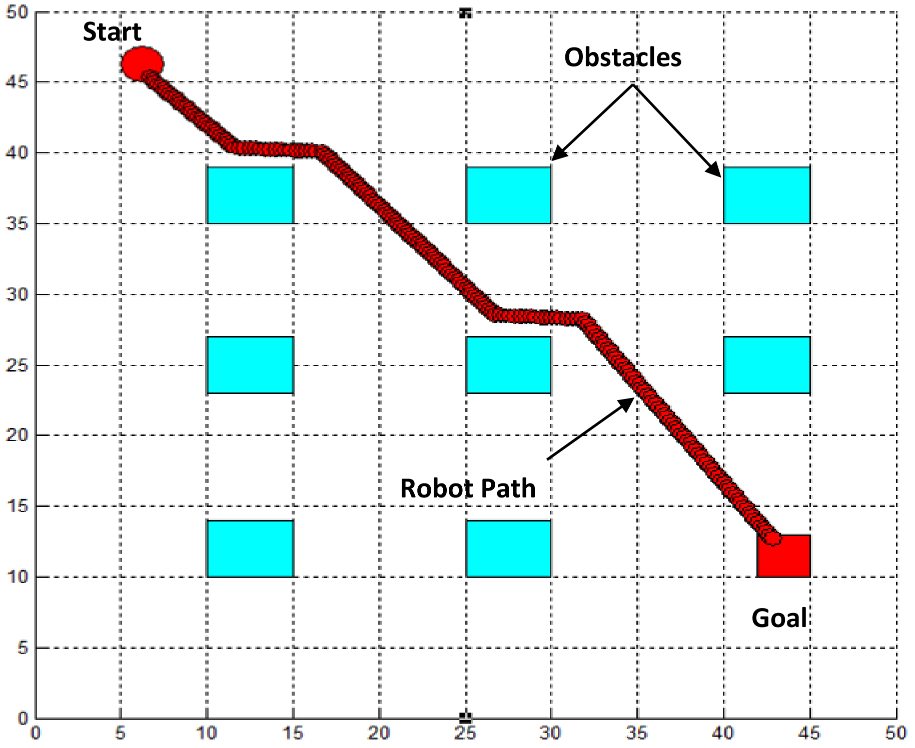


Figure 5.4 Simulation result for obstacle avoidance and goal seeking behavior

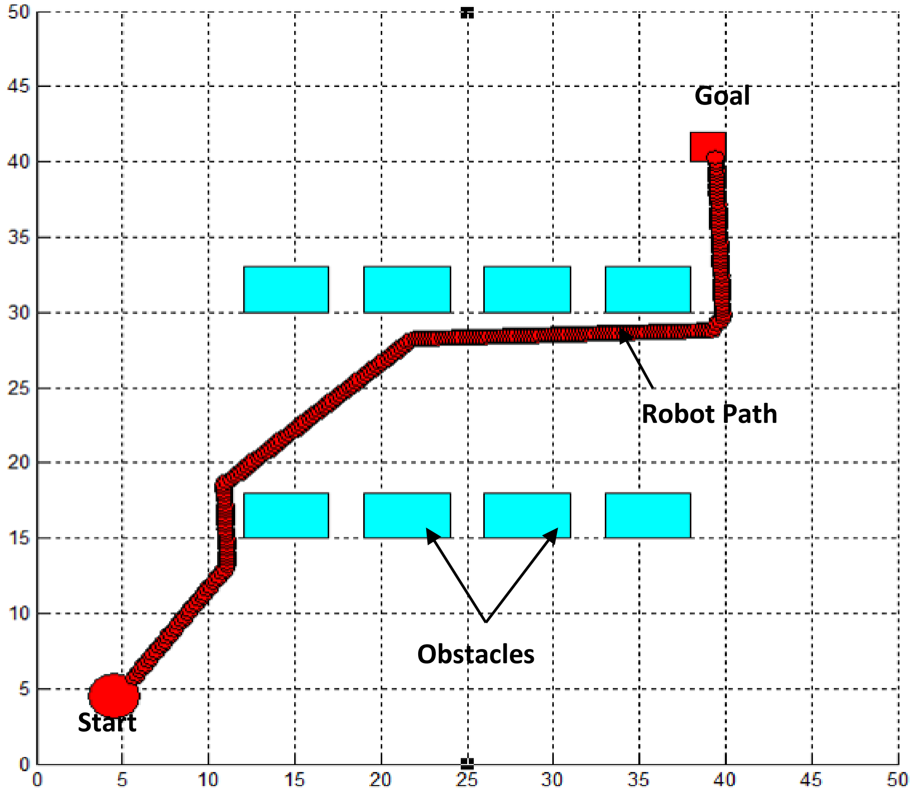


Figure 5.5 Simulation result for obstacle avoidance and wall following behavior

The environment consists of a start position, eight obstacles and one goal position. This exercise is designed to demonstrate, that the robot can successfully reach the goal point while avoiding the obstacles. Robots choose its direction and steer by its own to reach the target by following the near optimal path. It can be noted that the robots remain at a safe distance from the obstacles and move in smooth path from its start positions to end positions.

5.5.2 Wall Following and Target Seeking Behavior in an Unknown Environment

The exercise shown in figure 5.5 demonstrates the wall following and goal seeking behavior of a mobile robot in an unknown environment. In the present test environment, the obstacles are organized in a specific way so that they denote a wall between the robot and the goal. As the robots move from its start position and come across the walls then it continue to move alongside the wall by applying the wall following rules.

The next exercise demonstrates path planning of two mobile robots in a 2D test platform. The map consists of two mobile robots and one goal position. Robot path along with their start and goal position is shown in the figure 5.6. It has been found that both the robots were able to reach goal position by following the smooth and collision free trajectories.

Another test has been conducted with two mobile robots is shown in figure 5.7. In this exercise there are two start positions and two goal position associated to each mobile robot. Simulation result show the effectiveness of the RBF neural controller as both the robots successfully goes to their respective goal position by following near optimal path.

The exercise shown in figure 5.8 involves four mobile robots in a structured environment for navigation. The start and goal positions for all the robot is shown by square (green) and circle (red). For clear understanding, the paths followed by the robots are shown in different colors. In this simulation, each robot has reached their respective goal position in an efficient fashion without colliding with any obstacles and between themselves using the suggested controller.

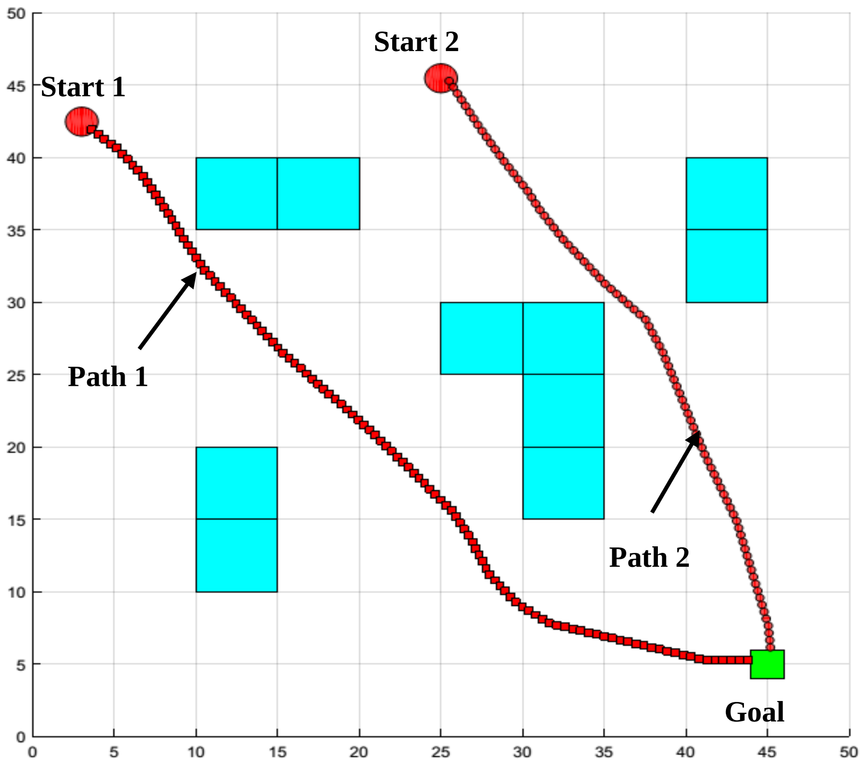


Figure 5.6 Test environment for navigation of two mobile robots (same goal position)

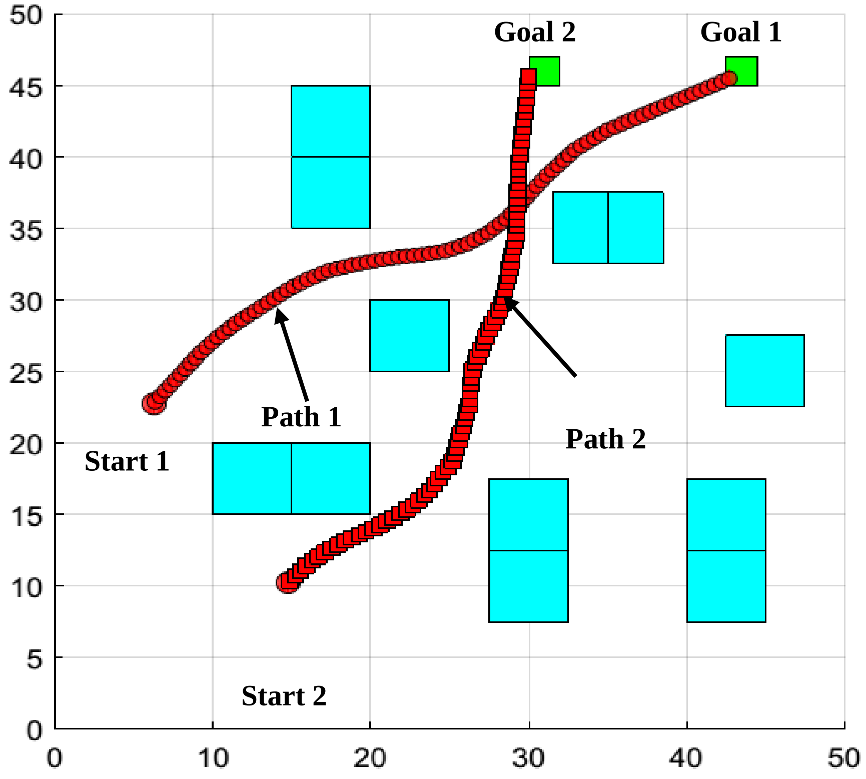


Figure 5.7 Test environment for navigation of two mobile robots (different goal position)

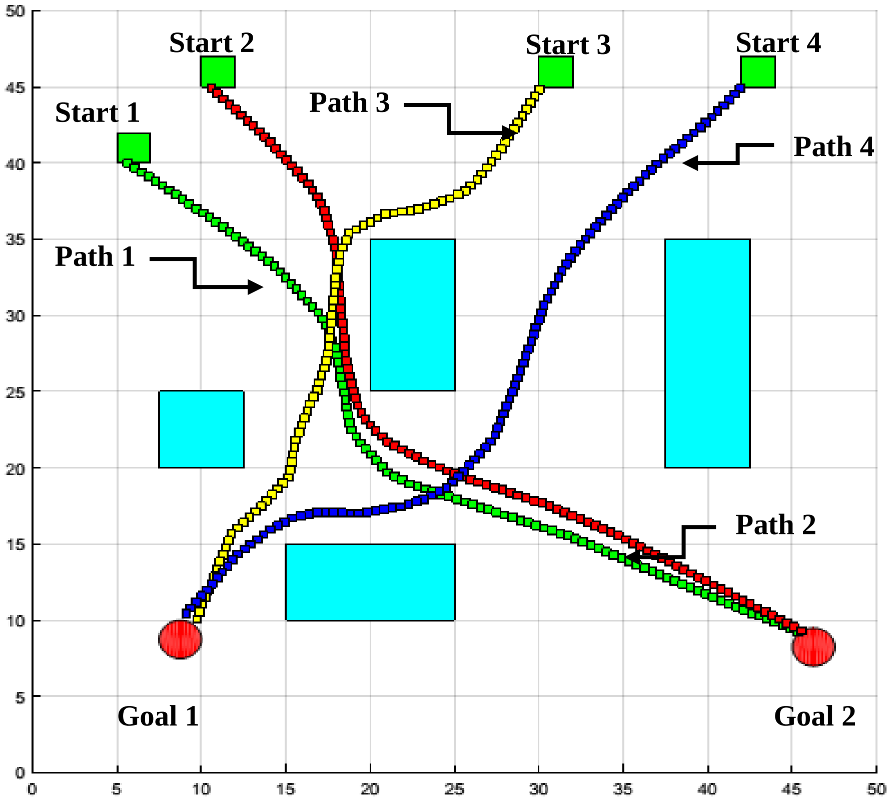


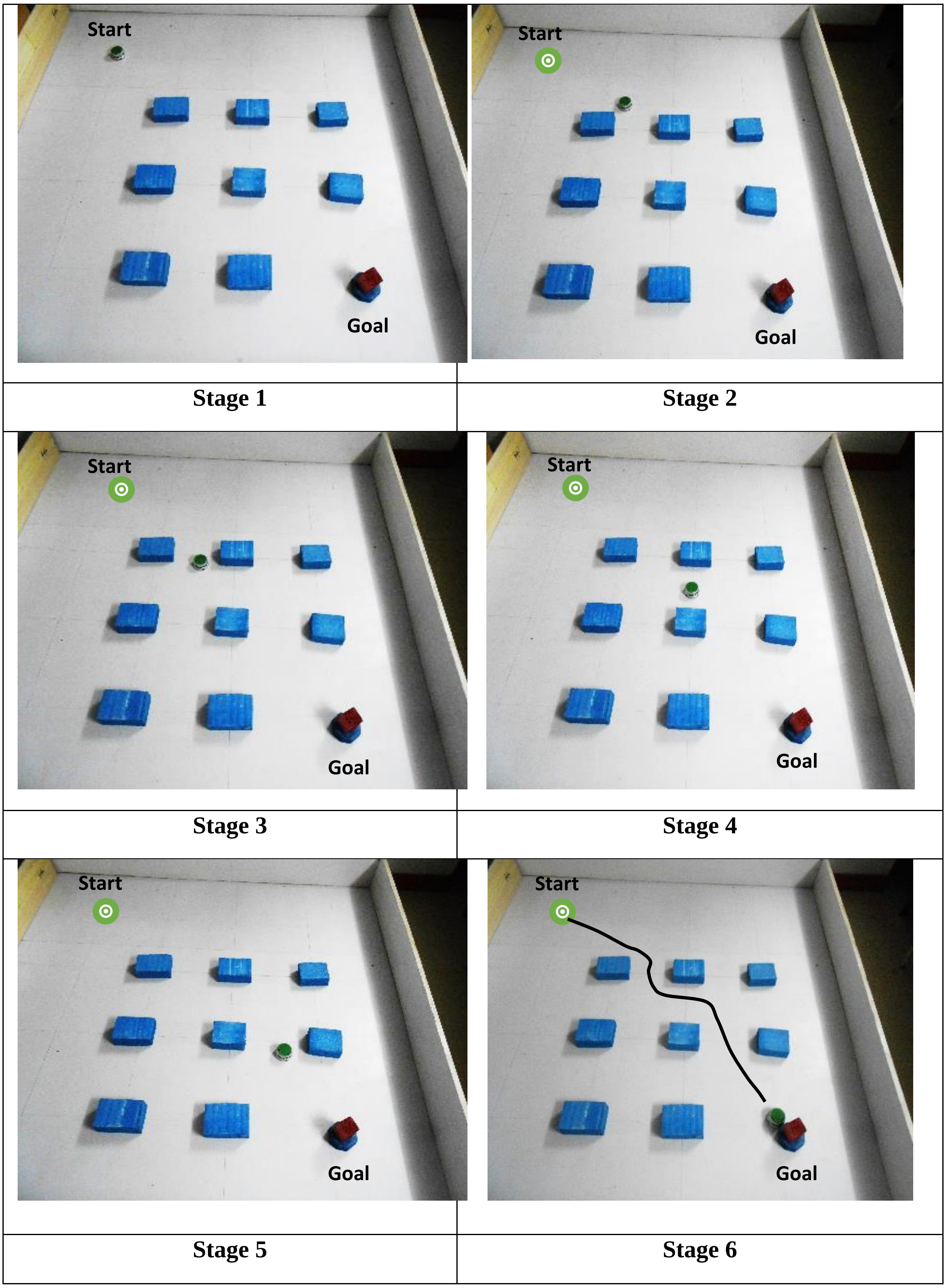
Figure 5.8 Test environment for navigation of four mobile robots

In the next section experimental results have been presented to validate the effectiveness of the proposed controller in real time. Experiments have been performed in the laboratory using an in-house mobile robot platform along with Khepera mobile robots.

5.6 Experimental Results

In this section, the proposed controller has been tested in a series of experiments to illustrate its effectiveness. The experiments have been carried out using the in-house robot platform developed in our laboratory and Khepera robots. All experiments have been carried out by considering robot as a rigid body moving in a plane surface. The specifications of the robots are given in Appendix A.

Robot is equipped with sonar and infrared sensors for measuring the distance of the nearby obstacles from the robot's current position. The neural controller uses the sensor data to perceive the environment and then provide appropriate steering angle.



Stage 1

Stage 2

Stage 3

Stage 4

Stage 5

Stage 6

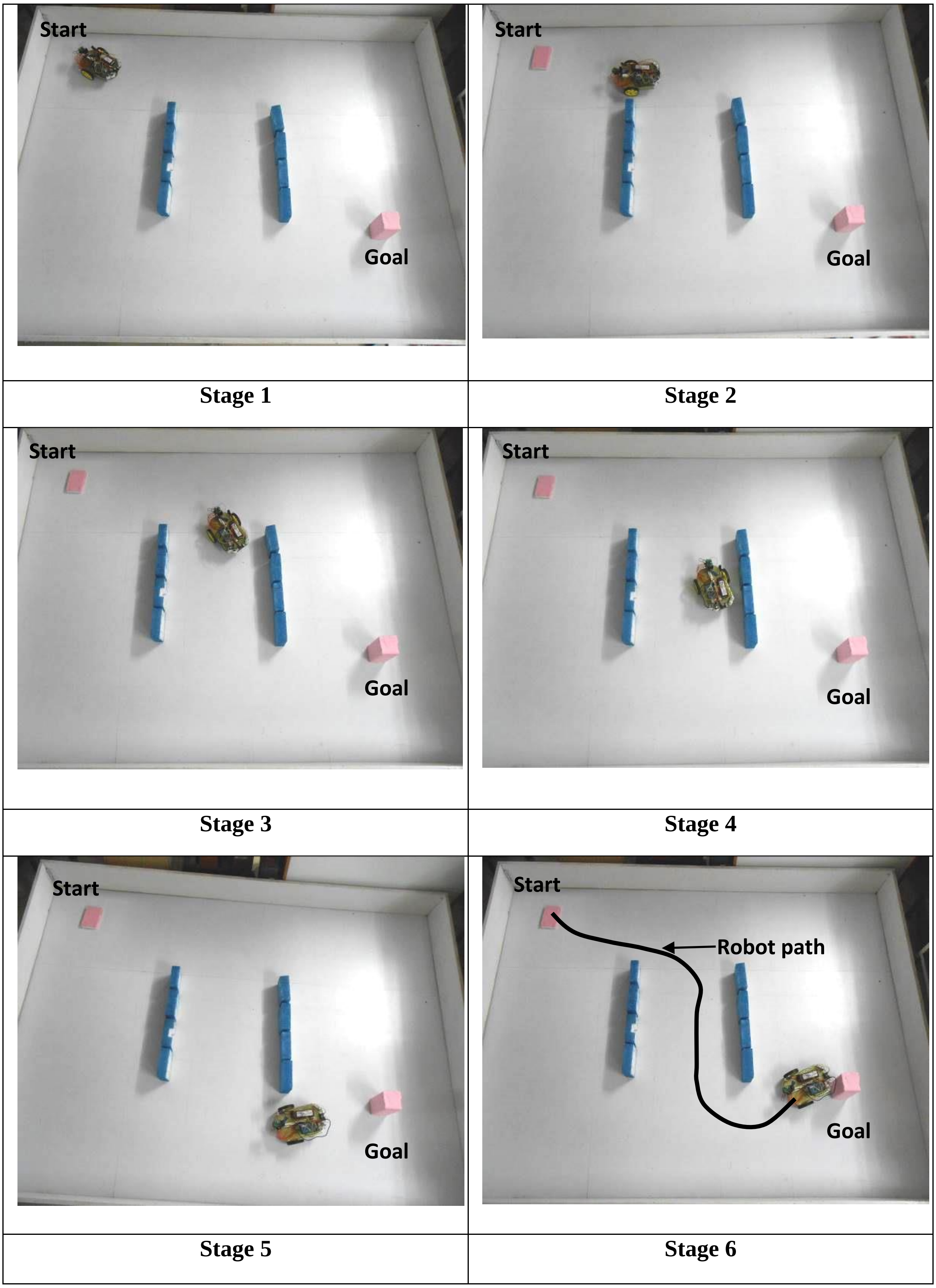
Figure 5.9 Experimental results for obstacle avoidance and goal seeking behavior (single robot)

During the course of motion, if robot comes across any obstacle then the controller offer a new steering angle corresponding to the senor data obtained from the different sensors mounted around the robot.

In this section, the proposed RBFN controller has been tested in a series of experiments to illustrate its effectiveness. The experiments have been carried out using an in-house robot platform developed in our laboratory and Khepera mobile robot. Khepera robots have been used for navigation of multiple mobile robots.

It has been observed that, by following the series of steering angle robot finally reaches the target position by following a collision free near optimal path. The time taken by the robot to reach the target position and path length for both computer simulation and real environment are given in the tables 5.1-5.2. Numerous exercises have been carried out in the laboratory for different environmental scenarios. The robotic behavior discussed above have been verified through experimental and simulation mod and the results are given in tables 5.3-5.6 for two and four mobile robots

Another exercise has been accomplished to demonstrate the wall following behavior by the robot as shown in figure 5.11. Figure 5.12 shows the collision free path generated by the two mobile robots in an unknown cluttered environment. The intermediate positions of the robots are shown in various stages.



Stage 1

Stage 2

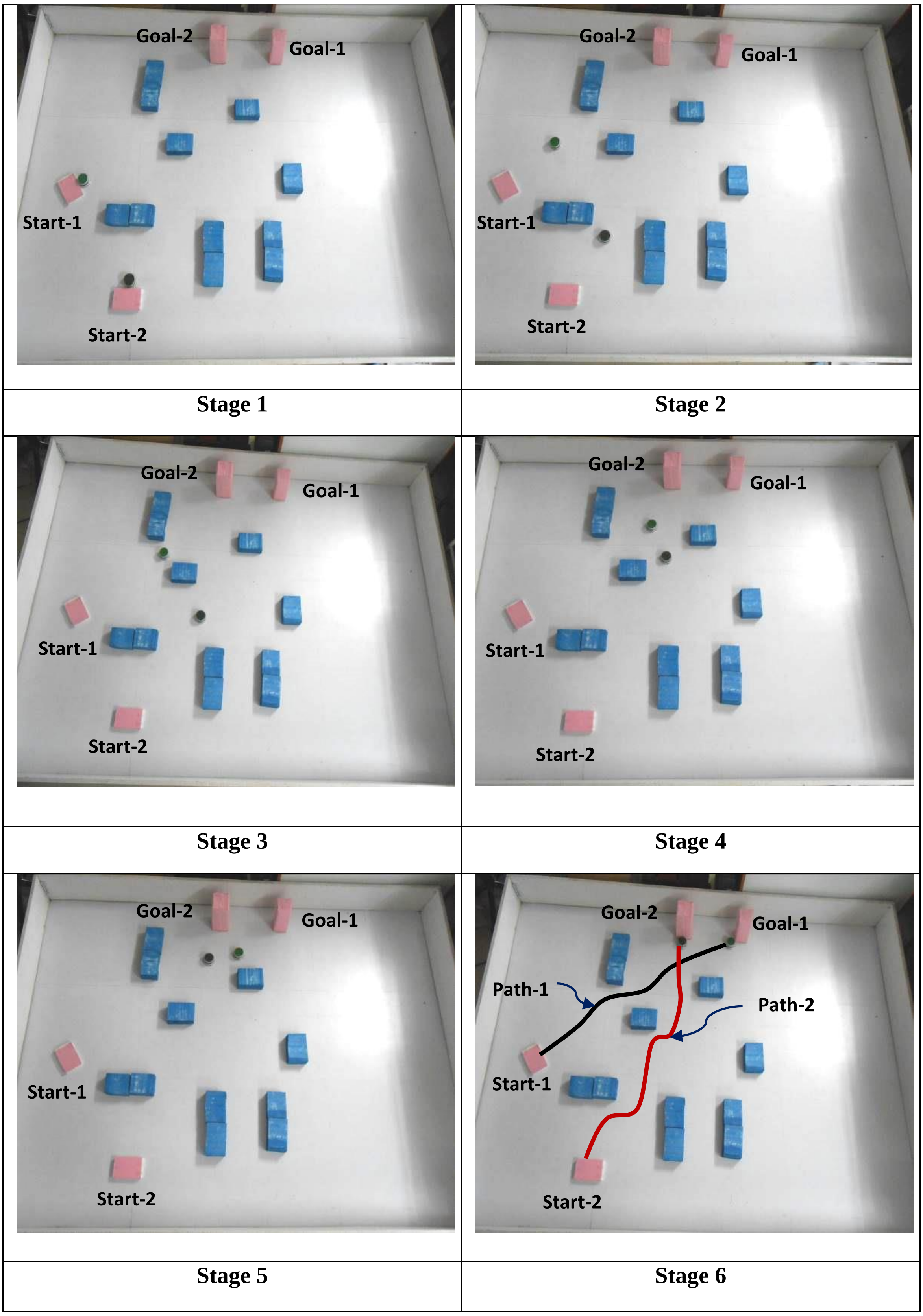
Stage 3

Stage 4

Stage 5

Stage 6

Figure 5.10 Experimental results for obstacle avoidance and wall following behavior



Stage 1

Stage 2

Stage 3

Stage 4

Stage 5

Stage 6

Figure 5.11 Experimental results for obstacle avoidance and wall following behavior

Table 5.2 Comparison between simulated and experimental result for path length (in 'cm', one robot)

Scenario	In simulation (distance travelled in 'cm')	In real time (distance travelled in 'cm')	% Error	Avg. % Error
1	147.85	153.74	3.98	4.76
2	192.53	203.15	5.51	
3	234.42	245.72	4.82	
4	162.89	171.65	5.37	
5	183.92	192.17	4.48	
6	242.63	251.39	3.61	
7	217.15	226.76	4.42	
8	198.37	209.04	5.37	
9	275.86	288.69	4.65	
10	154.29	162.53	5.34	

Table 5.3 Comparison of time taken to reach from start to goal position (in 'sec', one robot)

Scenario	In simulation (time taken in 'sec')	In real time (time taken in 'sec')	% Error	Avg. % Error
1	16.93	17.86	5.49	5.02
2	20.85	21.73	4.22	
3	25.92	27.07	4.43	
4	19.17	20.24	5.58	
5	20.64	21.62	4.75	
6	26.33	27.81	5.62	
7	23.97	25.16	4.96	
8	22.83	23.95	4.91	
9	29.78	31.48	5.72	
10	16.49	17.23	4.48	

Table 5.4 Comparison between simulated and experimental result for path length (in ‘cm’, two robots)

Scenario		In simulation (distance travelled in ‘cm’)	In real time (distance travelled in ‘cm’)	% Error	Avg. % Error	
					Robot 1	Robot 2
1	Robot 1	143.08	150.15	4.94	4.62	4.77
	Robot 2	121.37	128.64	5.98		
2	Robot 1	139.62	147.28	5.49		
	Robot 2	117.45	122.49	4.29		
3	Robot 1	171.84	179.34	4.36		
	Robot 2	148.29	154.38	4.11		
4	Robot 1	163.58	171.87	5.07		
	Robot 2	155.17	160.97	3.74		
5	Robot 1	182.27	192.16	5.43		
	Robot 2	163.53	172.09	5.23		
6	Robot 1	203.15	210.24	3.49		
	Robot 2	170.81	178.86	4.71		
7	Robot 1	194.42	203.37	4.6		
	Robot 2	142.94	151.25	5.81		
8	Robot 1	211.41	218.93	3.56		
	Robot 2	169.57	177.69	4.79		
9	Robot 1	159.73	167.33	4.76		
	Robot 2	134.49	140.52	4.48		
10	Robot 1	165.26	172.64	4.47		
	Robot 2	138.85	145.16	4.54		

Table 5.5 Comparison of time taken to reach from start to goal position (in 'sec', two robots)

Scenario		In simulation (time taken in 'sec')	In real time (time taken in 'sec')	% Error	Avg. % Error	
					Robot 1	Robot 2
1	Robot 1	16.17	17.09	5.68	4.59	4.81
	Robot 2	13.49	14.08	4.37		
2	Robot 1	15.87	16.73	5.42		
	Robot 2	12.97	13.71	5.71		
3	Robot 1	18.93	19.55	3.28		
	Robot 2	16.14	17.01	5.39		
4	Robot 1	17.82	18.76	5.27		
	Robot 2	17.29	17.96	3.88		
5	Robot 1	19.89	21.02	5.68		
	Robot 2	17.87	18.92	5.87		
6	Robot 1	22.31	23.15	3.77		
	Robot 2	18.69	19.38	3.69		
7	Robot 1	21.78	22.69	4.18		
	Robot 2	15.94	16.76	5.14		
8	Robot 1	23.25	24.17	3.96		
	Robot 2	18.72	19.65	4.97		
9	Robot 1	18.04	18.83	4.38		
	Robot 2	14.89	15.58	4.63		
10	Robot 1	18.57	19.37	4.31		
	Robot 2	15.37	16.06	4.49		

Table 5.6 Comparison of path length from start to goal position (in ‘cm’, four robots)

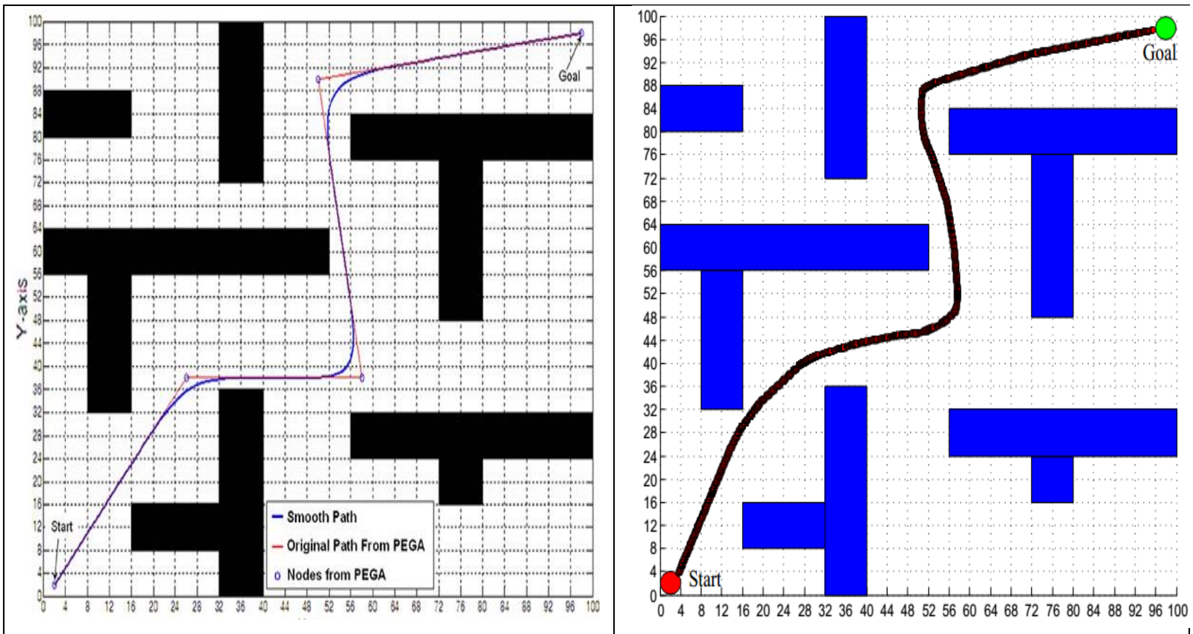
Scenario		In simulation (distance travelled in ‘cm’)	In real time (distance travelled in ‘cm’)	% Error	Avg. % Error			
					Robot 1	Robot 2	Robot 3	Robot 4
1	Robot 1	113.08	119.15	5.36	5.09	4.8	4.78	4.72
	Robot 2	133.82	140.23	4.79				
	Robot 3	143.58	151.87	5.77				
	Robot 4	107.26	111.69	4.13				
2	Robot 1	129.62	137.28	5.3				
	Robot 2	132.27	138.16	4.45				
	Robot 3	145.26	149.64	3.02				
	Robot 4	105.73	109.78	3.83				
3	Robot 1	115.56	121.59	5.22				
	Robot 2	135.84	142.34	4.78				
	Robot 3	149.74	156.29	4.37				
	Robot 4	109.59	115.56	5.44				
4	Robot 1	119.73	126.33	5.51				
	Robot 2	134.42	141.37	5.17				
	Robot 3	147.37	155.45	5.48				
	Robot 4	106.52	112.09	5.22				
5	Robot 1	125.15	130.24	4.06				
	Robot 2	137.26	143.86	4.81				
	Robot 3	144.58	152.26	5.3				
	Robot 4	108.38	113.76	4.96				

Table 5.7 Comparison of time taken to reach from start to goal position (in ‘sec’, four robots)

Scenario		In simulation (time taken in ‘sec’)	In real time (time taken in ‘sec’)	% Error	Avg. % Error			
					Robot 1	Robot 2	Robot 3	Robot 4
1	Robot 1	12.32	12.87	4.466	3.89	4.32	4.59	4.76
	Robot 2	14.74	15.27	3.59				
	Robot 3	15.86	16.72	5.42				
	Robot 4	11.83	12.34	4.31				
2	Robot 1	14.57	15.06	3.37				
	Robot 2	14.85	15.49	4.31				
	Robot 3	16.23	16.94	4.37				
	Robot 4	11.67	12.29	5.31				
3	Robot 1	12.91	13.43	4.03				
	Robot 2	15.02	15.63	4.06				
	Robot 3	16.57	17.31	4.47				
	Robot 4	12.32	12.89	4.62				
4	Robot 1	13.45	13.98	3.94				
	Robot 2	14.81	15.58	5.19				
	Robot 3	16.49	17.23	4.49				
	Robot 4	11.91	12.53	5.21				
5	Robot 1	13.84	14.35	3.68				
	Robot 2	15.37	16.05	4.42				
	Robot 3	16.19	16.87	4.2				
	Robot 4	11.97	12.49	4.344				

5.7 Comparison with Other Results

This section present the comparison between the developed RBFN controller in the present study with the other models presented in the literature. In particular, RBFN controller has been compared with the results given by Tsai et al. [147] and Qu et al. [148]. Figure 5.12 (a)-(b) to figure 5.16 (a)-(b) show the simulation result for ABFO controller and results found by Tsai et al. [147] and Qu et al. [148]. Comparison of path length is shown in Tables 5.8-5.12.



(a) (b)

Figure 5.12 (a) Simulation result by Tsai et al. [147]
(b) Simulation result by the proposed RBFN controller

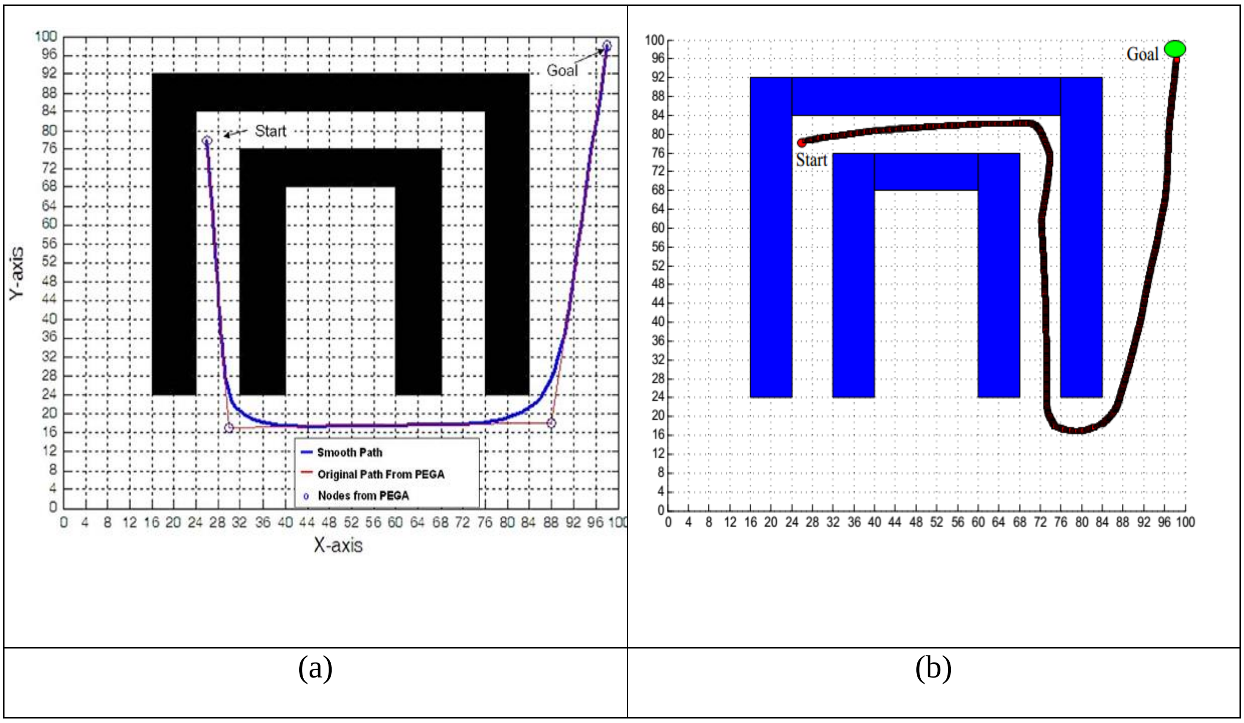


Figure 5.13 (a) Simulation result by Tsai et al. [147]
(b) Simulation result by the proposed RBFN controller

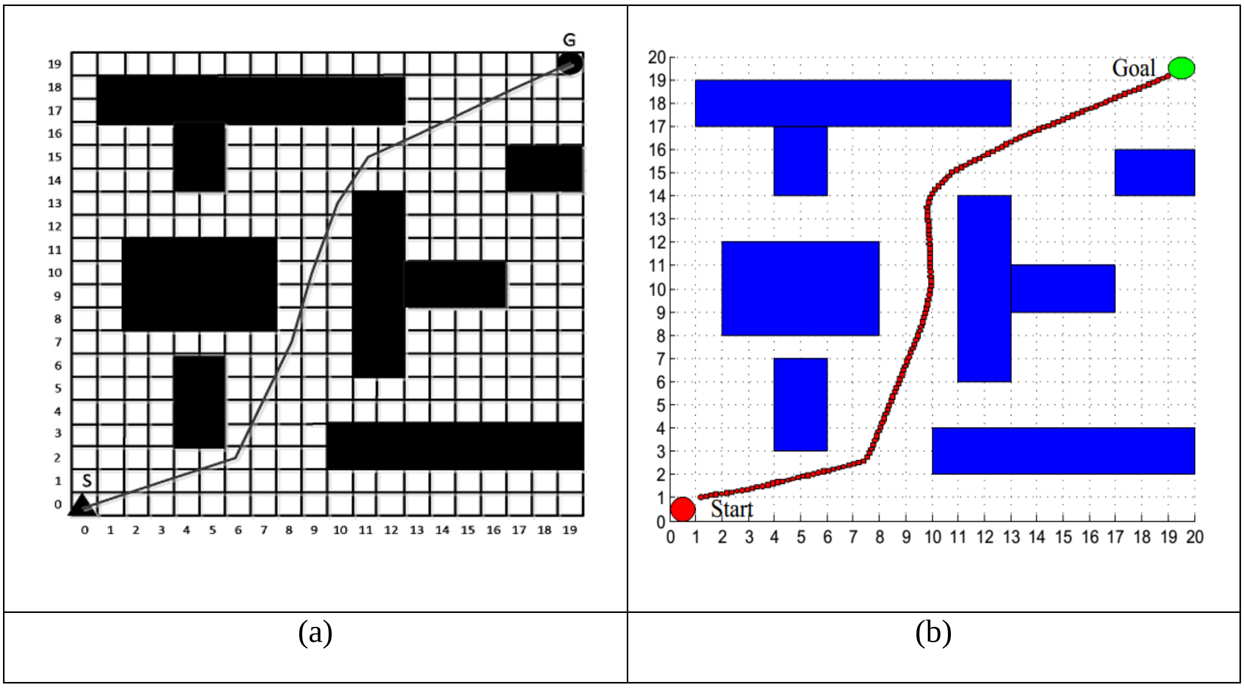
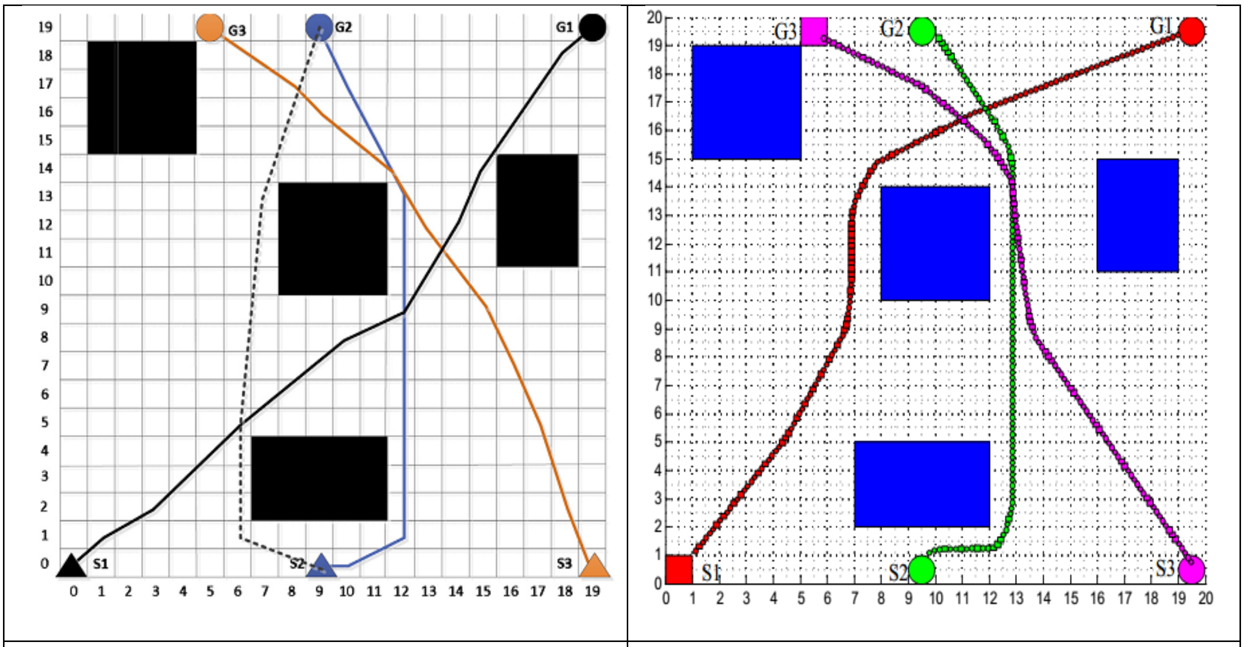


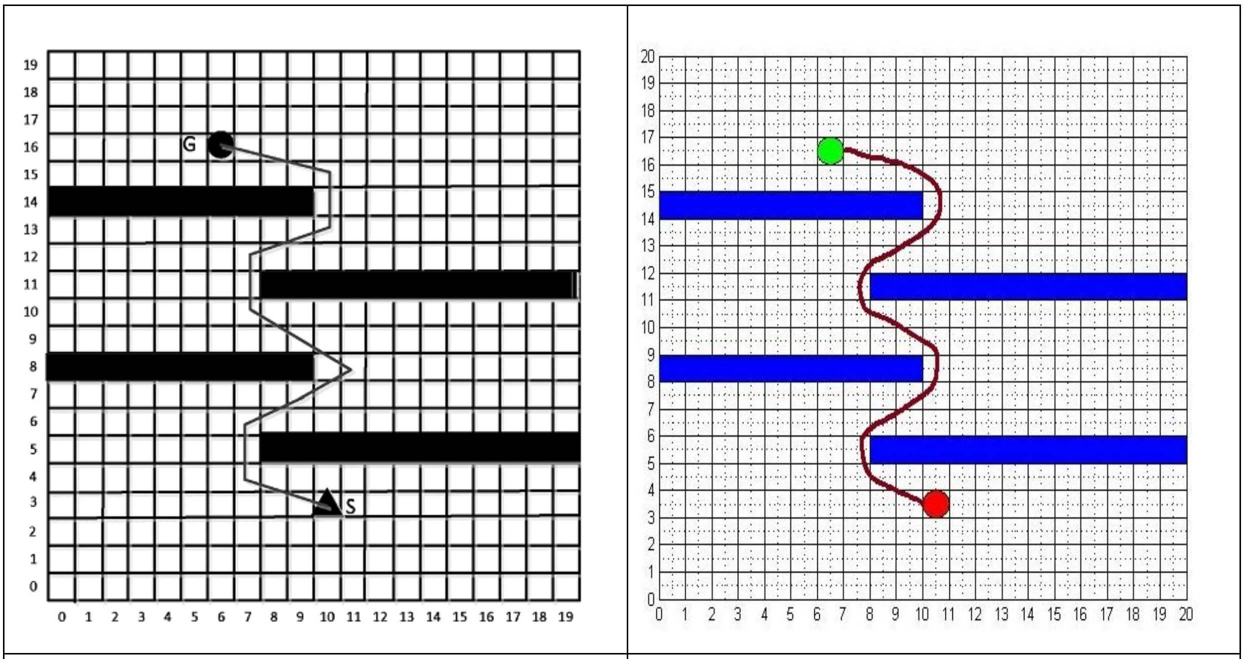
Figure 5.14 (a) Simulation result by Qu et al. [148]
(b) Simulation result by the proposed RBFN controller



(a)

(b)

Figure 5.15 (a) Simulation result by Qu et al. [148]
(b) Simulation result by the proposed RBFN controller



(a)

(b)

Figure 5.16 (a) Simulation result by Qu et al. [148]
(b) Simulation result by the proposed RBFN controller

Table 5.8 Comparison of results between Tsai et al. [147] and developed RBFN controller.

Technique Used	Path length (in cm)	Deviation (in %)
Figure 5.12 (a)	23.92	1.27
Figure 5.12 (b)	23.62	

Table 5.9 Comparison of results between Tsai et al. [147] and developed RBFN controller.

Figure	Path length (in cm)	Deviation (in %)
Figure 5.13 (a)	24.06	1.69
Figure 5.13 (b)	23.66	

Table 5.10 Comparison of results between Qu et al. [148] and developed RBFN controller.

Figure	Path length (in cm)	Deviation (in %)
Figure 5.14 (a)	23.34	0.60
Figure 5.14 (b)	23.20	

Table 5.11 Comparison of results between Qu et al. [148] and developed RBFN controller.

Figure	Path length (in cm)		
	Robot1	Robot2	Robot3
Figure 5.15 (a)	20.44	15.84	17.7
Figure 5.15 (b)	20.20	15.45	17.56
Deviation (in %)	1.18	2.52	0.79

Table 5.12 Comparison of results between Qu et al. [148] and developed RBFN controller.

Figure	Path length (in cm)	Deviation (in %)
Figure 5.16 (a)	24.68	6.15
Figure 5.16 (b)	23.25	

It can be observed from the simulation results that proposed RBFN controller offers smooth paths as compared to results obtained by Tsai et al. [147] and Qu et al. [148]. Moreover, the path length is found to be less in present research as compared to Tsai et al. [147] and Qu et al. [148]. This shows the effectiveness of the proposed controller.

5.8 Summary

Radial basis function (RBF) neural controller has been used to develop another controller for the navigation of mobile robots. The proposed RBF neural controller uses obstacles distances with respect to its current position as input along with the bearing of the goal position. The network is trained with the help of large training data set and training patterns for the prediction of a suitable value of the steering angle. Depending upon the input data set controller provides an appropriate value of steering angle to reach the target position. The percentage of errors between the simulation and experimental results are found to be within 6 %. Further, a comparison is made between the results obtained in present research and results by Tsai et al. [147] and Qu et al. [148]. It has been found that proposed controller generates smooth path and takes shorter path as compared to approaches used by Tsai et al. [147] and Qu et al. [148].

Chapter 6

ANALYSIS OF BEES ALGORITHM FOR MOBILE ROBOT NAVIGATION

In this chapter, another nature inspired technique based on Bees Algorithm (BA) has been proposed for the navigation of mobile robot. The suggested intelligent controller allows the robot to positively navigate in the real world environment and confirm that the robot reaches the predefined goal position. Simulation results are presented to show the effectiveness of the proposed controller. Moreover, simulation results are compared with experimental results with real mobile robots to validate the performance of the proposed controller in physical environment.

6.1 Introduction

The swarm based algorithms are found to be efficient to find near optimal solution to many real world optimization problems. Therefore, swarm based algorithm can be considered as intelligent optimization tool. Swarm based algorithms are inspired from nature's method to find optimum solution. The basic difference between a direct search method and a swarm based approach is that, swarm based approaches use a population of solutions for each iteration rather than a single solution.

Moreover, in swarm based approach population of solution processed in an iteration and outcome of each iteration is also a population of solutions. In this chapter, bees algorithm has been used for the analysis and optimization of path for mobile robots during the navigation.

6.2 Bees Algorithm

Bees Algorithm (BA) is a new member in the family of nature inspired algorithms. BA has been inspired by the way honey bees search for their food.

6.2.1 Bees in Nature

The colony of honey bees can spread up to 10 km in random directions (can cover an area of 10 km radius) to find more number of food sources (to explore food sources) as shown in figure 6.1. They use their members optimally as they employed more bees to those flower patches from which generous amount of nectar or pollen can be collected with less effort.

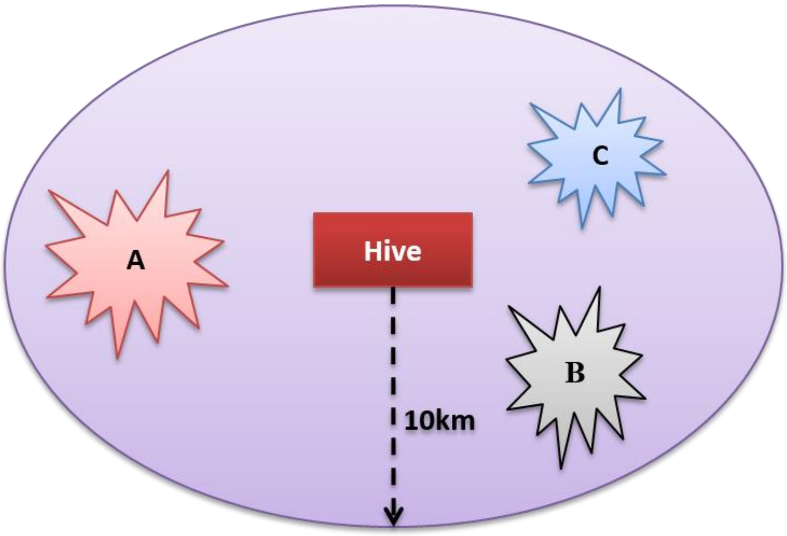


Figure 6.1 Representations of the colony of honey bee and food sources (i.e. A, B and C)

Bees transfer the following information through waggle dance which includes the following information:

1. The bearing of flower patches or food sources is same as the angle between the sun and the patch.

2. The information about the distance of the food source from the hive is communicated by duration of the dance.

3. The quality rating (fitness) of a source is defined by the frequency of the dance.

The foraging process starts with search action of scout bees. They fly arbitrarily in several directions to explore all flower patches in the surrounding. They collect the food sample and return to the hive and deliver the essential information to other bees by performing waggle dance. Scouts bees offer important information regarding a flower patch with the help of waggle dance as direction quality and distance of the flower patch from the hive. This process is essential for the survival of the colony as it helps the colony to employ flower bees to the flower patches optimally [145,146].

The pseudo code for the algorithm is defined as

1. Initialization: set random solution for the population.
2. Evaluation: calculate the fitness of the population.
3. While (termination criterion not met) create new population.
4. Site Selection: choose sites for neighborhood search.
5. Optimal allocation of bees: Assign bees for chosen sites in the previous step. (assign more bees to elite site).
6. Selection of fittest bee: select bees from each group according to fitness (i.e. bee having highest fitness).
7. Allocate remaining bees for random search and calculate their fitness.
8. End.

The algorithm begins with a defined number of scout bees positioned arbitrarily in the search environment. The fitness of the food sites visited by the scout bees are calculated in the next step (i.e. step 2). In step 4, bees having the maximum fitness are reserved as selected bees and the corresponding food sites visited by them are selected for the neighborhood search. Furthermore, in next steps (i.e. step 5 and 6) the algorithm performs neighborhood search in the selected sites by allocating the more bees to search close to the elite sites. The bees can be selected on the basis of fitness value of the sites visited. In step 6, for each food source, one bee with maximum fitness will be chosen to form the new population. In the next step (step 7), the remaining bees are then allocated arbitrarily nearby the search space seeking for new possible solutions. The above steps are repeated until a termination criterion is met.

6.2.2 Parameters of the Bees Algorithm

N_{bees}: sum of scout bees

$N_{location}$: number of locations chosen out of $\mathrm{N_{bees}}$ visited sites

N_{elite}: number of best locations out of m selected sites

N_{be}: number of bees recruited for best e sites

N_{bo}: number of bees recruited for other ($N_{location}$ - N_{elite}) selected sites

N_{ngh}: size of neighborhood search and stopping criterion.

6.3 Cause and Problem Formulation for Considering Bees Algorithm

In nature, honey bees have several complicated behaviors such as mating, breeding and foraging. These behaviors have been mimicked for several honey bee based optimization algorithms. The following algorithms are inspired from foraging behavior of honey bees;

Bee System (BS),

Bee Colony Optimization (BCO),

Artificial Bee Colony (ABC) and

The Bees Algorithm (BA).

The Bees Algorithm is very similar to the ABC in the sense of having local search and global search processes. However there is a difference between both algorithms during the neighborhood search process. As mentioned above, ABC has a probabilistic approach during the neighborhood stage; however the Bees Algorithm does not use any probability approach, but instead uses fitness evaluation to drive the search. The BA has both local and global search capability utilizing exploitation and exploration strategies, respectively. The BA also has advantages and disadvantages compared to the other algorithms

Advantages: The algorithm has local search and global search ability, Implemented with several optimization problems, Easy to use, and available for hybridization combination with other algorithms. As in the current research the robot has to reach the goal using near optimum path from start to goal. Bees algorithm can be useful as it has local and global search ability and available for hybridization with other approaches. That is why Bees Algorithm is taken into consideration in the current research.

Disadvantages: The algorithm starts with random initialization and the algorithm has several parameters which need to be tuned.

As in the current research the robot has to reach the goal using near optimum path from start to goal. Bees algorithm can be useful as it has local and global search ability and available for hybridization with other approaches. That is why Bees Algorithm is taken into consideration in the current research.

A 2D work space is considered as test environment which contains a set of static obstacles, and a mobile robot. The position of mobile robot at nay instance can be defined as the coordinate (x, y). The goal of the robot is to find a series of motion commands to drive the robot from an initial position (x_{start}, y_{start}) to a final goal position (x_{goal}, y_{goal}). Further, the path followed by the robot should be free from obstacles.

6.4 Mobile Robot Path Planning using Bees Algorithm (BA)

In this section, another algorithm inspired from the food foraging behavior of honey bees is used for the path planning of a mobile robot. The bearing of the robot is opted according to the position of fittest bee.

The motion planer initializes the food position which represents a feasible solution to the optimization problem. This step can be given as

$$x_{i,j} = l_j + rand(0,1)(u_j - l_j) \qquad (6.1)$$

where, l_j and u_j are the lower and upper limits of the parameter j respectively. *Rand* (0,1) represents the random number between 0 and 1.

Each scout bee is linked to a food position. A scout bee looks for a new food position v_i using its present food position as follows:

$$v_{i,j} = x_{i,j} + \Phi_{i,j}(x_{i,j} - x_{k,j})$$

(6.2)

where, k is a random food position index ($k \neq i$), j is a parameter chosen at random and $\Phi_{i,j}$ is a random number between [-1, 1]. The scout bee computes the nectar amount of v_i, and then x_i, the food position x_i is replaced by $v_{i,}$ and its trail counter is reset to zero, otherwise x$_i$ is retained and its trial counter increases by one.

Selection of a food source position x$_i$ depends upon the probability P_i which is defined as follows:

$$P_i = \frac{fitness_i}{\sum_{j=1}^{SN} fitness_j} \tag{6.3}$$

where, $fitness_i$ represents the quality of the food source x$_{i.}$ As soon as the bees have selected the food positions, they generate new random populations using equation 6.2. The new food positions are evaluated and the same greedy selection is applied.

When a food source position has not improve in a number of trails, (termination value), this solution is replaced by a random food position x$_i$ using equation 6.1.

6.4.1 Motion Planning using Bees Algorithm

The aim of this step is to create a collision free feasible path joining the initial and final location using the BA. The path is generated by finding a series of vertices or points from a sample of geometric configurations in the free workspace. The points to be incorporated in the selected path are found by a local search procedure using the BA. The motion planer finds the next position in the feasible path by taking into account the distances of the position from goal and obstacles. The motion planer selects the points which are close to goal position and avoids those who result in path segments in collision with the obstacles.

For this, a fitness function has been used is given below:

A food position represents an index $n \in [1, \mathrm{N}+1]$ to the next vertex $V_n(\mathrm{x}, \mathrm{y})$, where the configuration $V_{N+1}(\mathrm{x}, \mathrm{y}) = \mathrm{G}(\mathrm{x}, \mathrm{y})$.

We evaluate each index using the Euclidean distance to the goal:

$$F(\mathrm{n}) = \left\| \mathrm{V}_n(\mathrm{x}, \mathrm{y}) - \mathrm{G}(\mathrm{x}, \mathrm{y}) \right\| + f_c * p_n \tag{6.4}$$

Now the task is to find the value of n^* such that

$$\mathrm{n}^* = \mathrm{n} \leftrightarrow F(\mathrm{n}) = \underset{n \in [1, \mathrm{N}+1]}{min} F(\mathrm{n}) \tag{6.5}$$

Where, $f_c = 2\sqrt{a^2 + b^2}$ is a correction factor, a and b are the width and the height of the environment. p_n is a correction counter. The correction factor is required when:

- The portion of the path between the present position and the next position is in collision with the obstacles.
- The distance between the immediate obstacle and the segment of the present and next position is less than the robot radius.

The value of n^* minimizes the function specified by equation 6.4. In order to reach the goal position the best value is added in the current position of the robot. Further robot again search for the new best position and update its current position until it reaches its goal position. It results in local path planning of the robot from start to goal position.

Another the term $\frac{K}{OD}$ is added in the fitness function (equation 6.4) to achieve the global path planning optimization. Then the equation 6.4 becomes

$$F(\mathrm{n}) = \left\| \mathrm{V}_n(\mathrm{x}, \mathrm{y}) - \mathrm{G}(\mathrm{x}, \mathrm{y}) \right\| + f_c * p_n + \frac{K}{OD} \tag{6.6}$$

Where K is a constant. It should be noted that, the term $\frac{K}{OD}$ is added only if the distance of robot (OD) next position to the nearest obstacle is less than some threshold value. In this work, threshold value is taken as 20 units. Therefore,

$$F(\text{n}) = \left| \begin{array}{ll} F(\text{n}) = \left\| \text{V}_n(\text{x}, \text{y}) - \text{G}(\text{x}, \text{y}) \right\| + f_c * p_n & \text{if } OD \leq \text{Threshold value} \\ F(\text{n}) = \left\| \text{V}_n(\text{x}, \text{y}) - \text{G}(\text{x}, \text{y}) \right\| + f_c * p_n + \frac{K}{OD} & \text{if } OD \leq \text{Threshold value} \end{array} \right. \tag{6.7}$$

Now to compute the path length, the Euclidean distance between the intermediate points is calculated.

$$F(\text{L}_{path}) = \sum_{i=1}^{M-1} \left\| V_i(\text{x}, \text{y}) - V_{i+1}(\text{x}, \text{y}) \right\| \tag{6.8}$$

Now, to get the optimal path we need to find L_{path} such that

$$\text{L}_{path}{}^{*} = \text{L}_{path} \leftrightarrow F(\text{L}_{path}) = \underset{L_{path} \in \text{C}_0}{min} F(\text{L}_{path}) \tag{6.9}$$

where, C_0 represents the free space for path planning.

6.5 Simulation and Experimental Results

The proposed Bees Algorithm has been tested in the simulation environment. The results for one and four mobile robots during wall following and goal seeking behavior are shown in figures 6.2-6.3. Figures 6.4 and 6.5 show the experimental paths for one, and four mobile robots. Tables 6.1-6.6 demonstrate the comparison of path lengths and time taken by BA for one, two and four number of robots in simulation and real time modes.

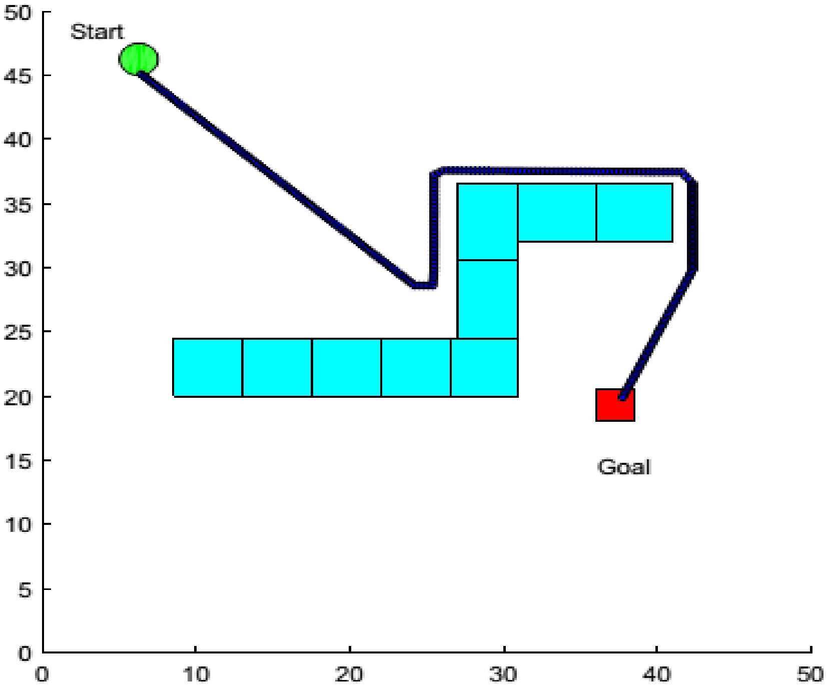


Figure 6.2 Robot path from start to goal position (in simulation)

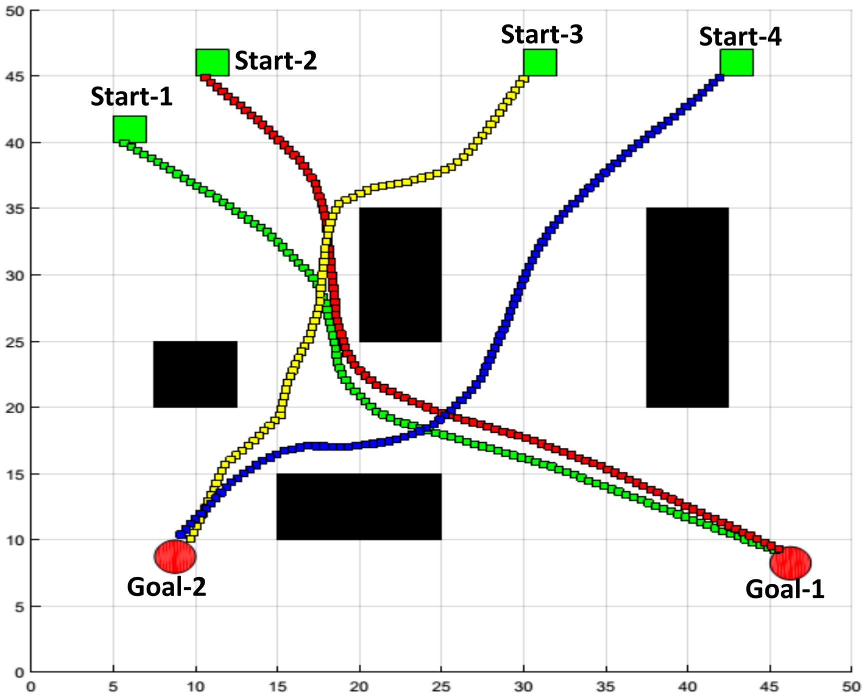
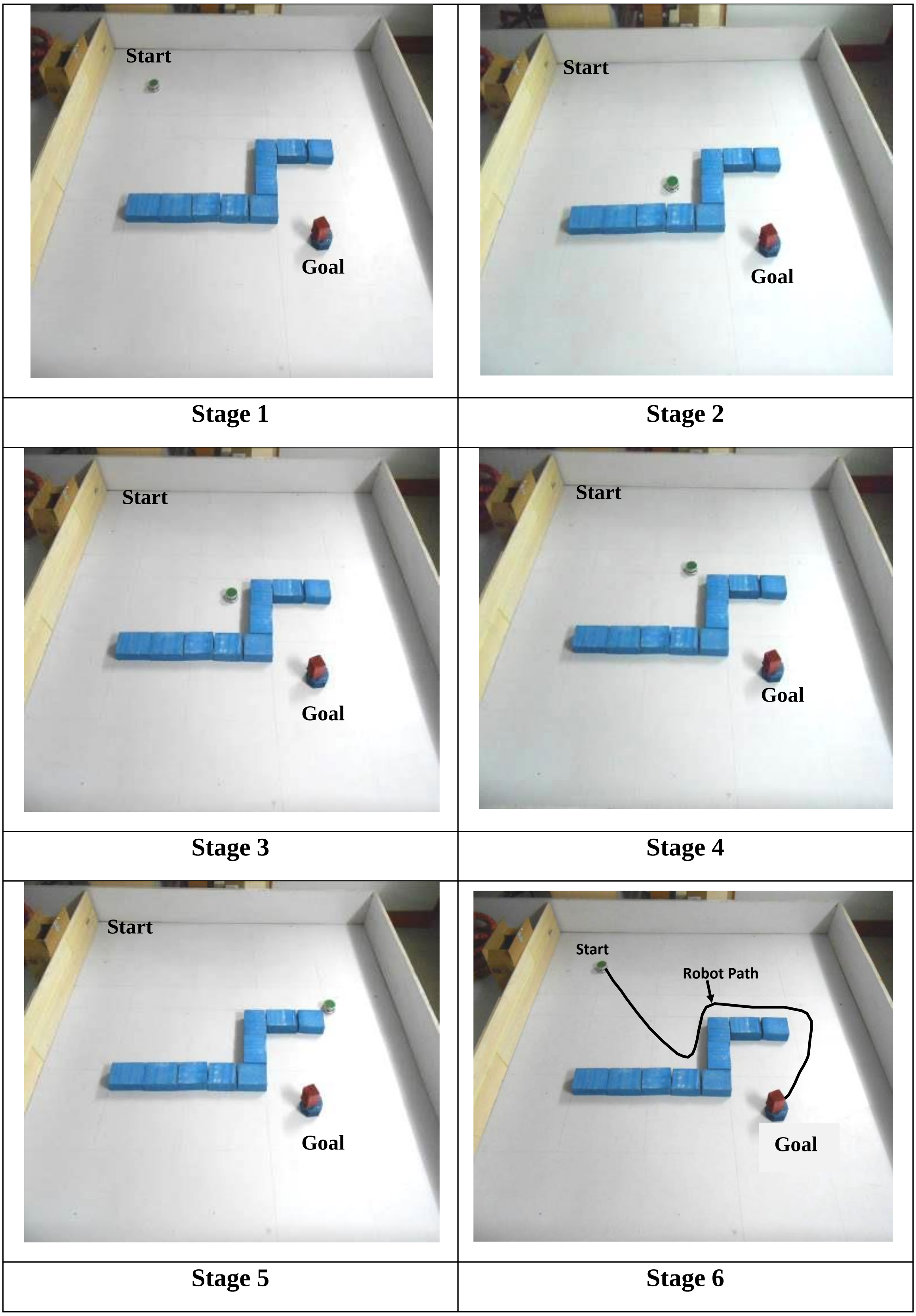


Figure 6.3 Navigation using four mobile robots and two goal positions (in simulation)



Stage 1

Stage 2

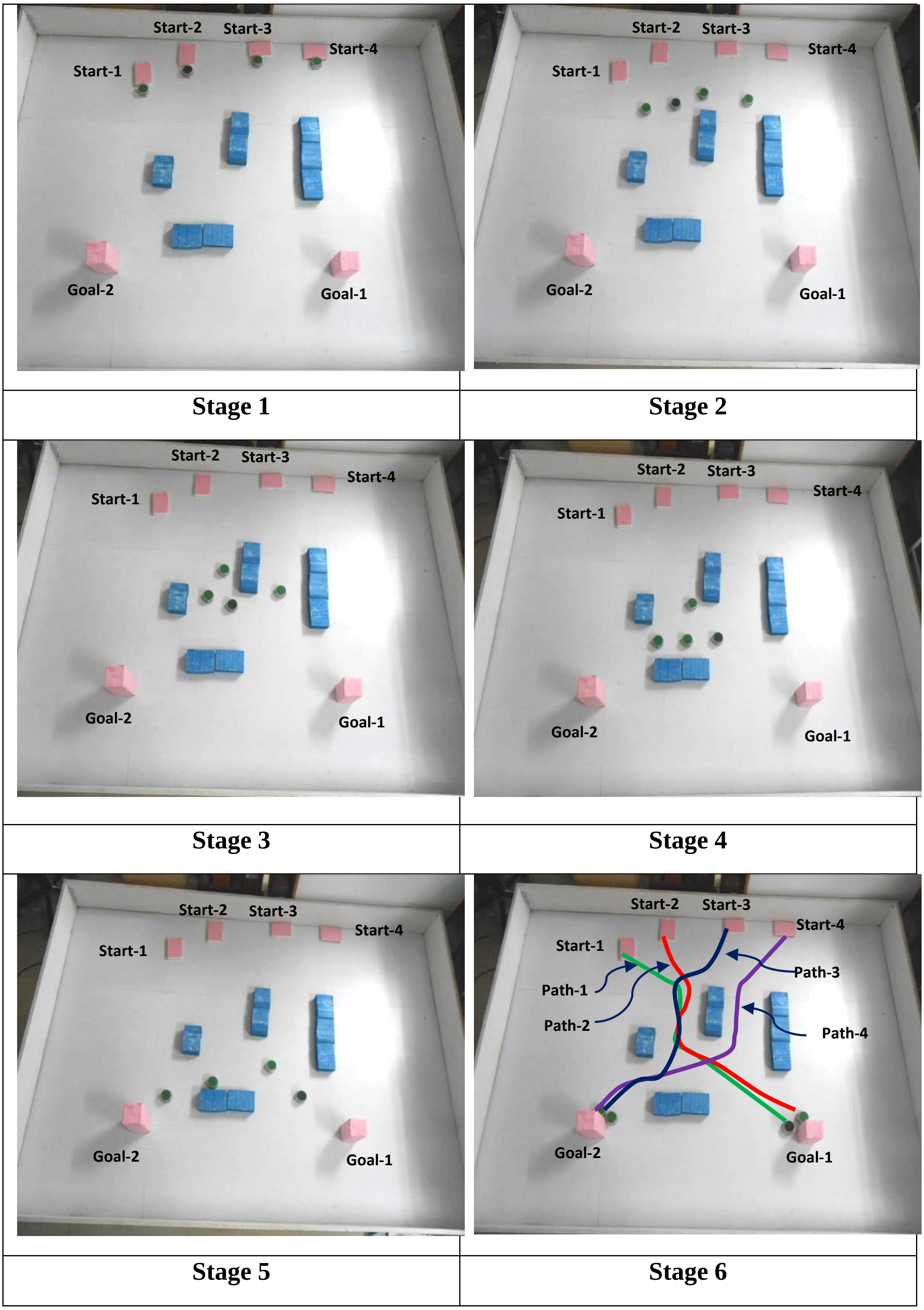
Stage 3

Stage 4

Stage 5

Stage 6

Figure 6.4 Experimental result for wall following and goal seeking behavior (single robot)



Stage 1

Stage 2

Stage 3

Stage 4

Stage 5

Stage 6

Figure 6.5 Experimental results for four mobile robot and two goal positions

Table 6.1 Distance travelled from start to goal position (in ‘cm’, one robot)

Scenario	In simulation (distance travelled in ‘cm’)	In real time (distance travelled in ‘cm’)	% Error	Avg. % Error
1	144.89	150.67	3.831	4.539
2	188.68	199.09	5.228	
3	229.73	240.81	4.599	
4	159.63	168.22	5.103	
5	180.24	188.33	4.293	
6	237.78	246.36	3.485	
7	212.81	222.22	4.238	
8	194.40	204.86	5.104	
9	270.34	282.92	4.444	
10	151.20	159.28	5.070	

Table 6.2 Time taken to reach from start to goal position (in ‘sec’, one robot)

Scenario	In simulation (time taken in ‘sec’)	In real time (time taken in ‘sec’)	% Error	Avg. % Error
1	16.63	17.54	5.207	4.775
2	20.47	21.34	4.050	
3	25.45	26.58	4.248	
4	18.82	19.88	5.287	
5	20.27	21.23	4.533	
6	25.86	27.31	5.322	
7	23.54	24.71	4.730	
8	22.42	23.52	4.676	
9	29.24	30.91	5.400	
10	16.19	16.92	4.295	

Table 6.3 Distance travelled from start to goal position (in ‘cm’, two robots)

Scenario		In simulation (distance travelled in ‘cm’)	In real time (distance travelled in ‘cm’)	% Error	Avg. % Error	
					Robot 1	Robot 2
1	Robot 1	140.36	147.15	4.611	4.311	4.452
	Robot 2	119.06	126.07	5.555		
2	Robot 1	136.97	144.33	5.104		
	Robot 2	115.22	120.04	4.017		
3	Robot 1	168.58	175.75	4.084		
	Robot 2	145.47	151.29	3.847		
4	Robot 1	160.47	168.43	4.726		
	Robot 2	152.22	157.75	3.505		
5	Robot 1	178.81	188.32	5.050		
	Robot 2	160.42	168.65	4.877		
6	Robot 1	199.29	206.04	3.274		
	Robot 2	167.56	175.28	4.403		
7	Robot 1	190.73	199.30	4.303		
	Robot 2	140.22	148.23	5.398		
8	Robot 1	207.39	214.55	3.336		
	Robot 2	166.35	174.14	4.472		
9	Robot 1	156.70	163.98	4.445		
	Robot 2	131.93	137.71	4.194		
10	Robot 1	162.12	169.19	4.177		
	Robot 2	136.21	142.26	4.249		

Table 6.4 Time taken to reach from start to goal position (in ‘sec’, two robots)

Scenario		In simulation (time taken in ‘sec’)	In real time (time taken in ‘sec’)	% Error	Avg. % Error	
					Robot 1	Robot 2
1	Robot 1	15.93	16.82	5.287	4.288	4.492
	Robot 2	13.29	13.85	4.093		
2	Robot 1	15.63	16.46	5.044		
	Robot 2	12.78	13.49	5.301		
3	Robot 1	18.65	19.24	3.073		
	Robot 2	15.90	16.74	5.018		
4	Robot 1	17.55	18.46	4.914		
	Robot 2	17.03	17.67	3.633		
5	Robot 1	19.59	20.68	5.280		
	Robot 2	17.60	18.62	5.454		
6	Robot 1	21.98	22.78	3.531		
	Robot 2	18.41	19.07	3.462		
7	Robot 1	21.45	22.33	3.913		
	Robot 2	15.70	16.49	4.796		
8	Robot 1	22.90	23.78	3.709		
	Robot 2	18.44	19.34	4.636		
9	Robot 1	17.77	18.53	4.098		
	Robot 2	14.67	15.33	4.332		
10	Robot 1	18.29	19.06	4.033		
	Robot 2	15.14	15.80	4.199		

Table 6.5 Distance travelled from start to goal position (in ‘cm’, four robots)

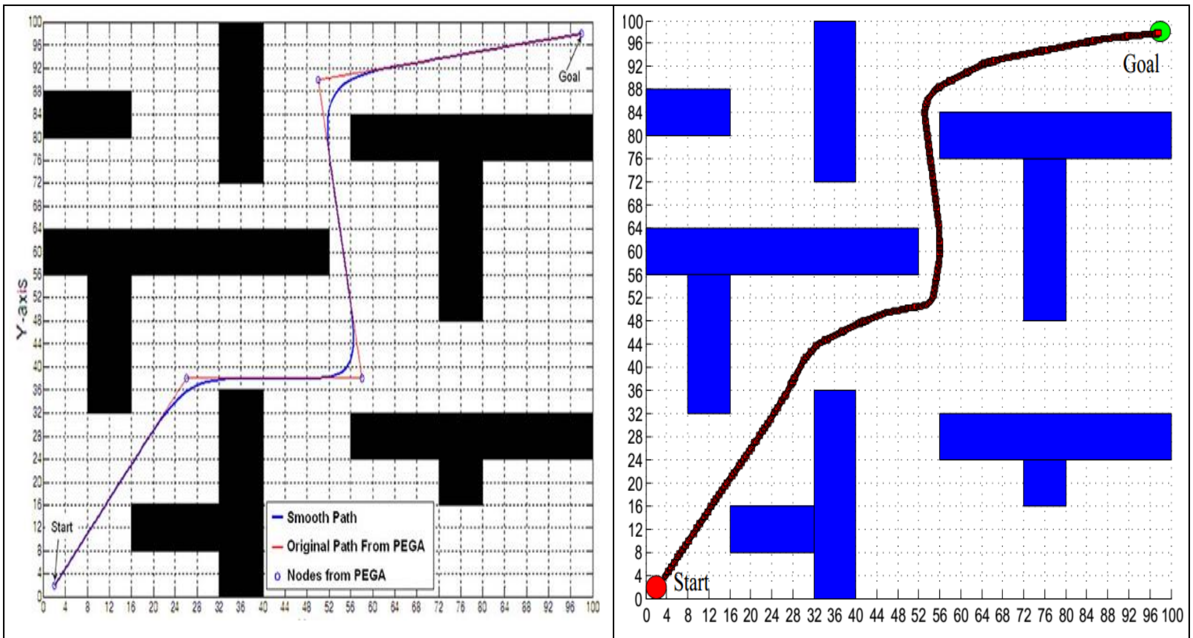
Scenario		In simulation (distance travelled in ‘cm’)	In real time (distance travelled in ‘cm’)	% Error	Avg. % Error			
					Robot 1	Robot 2	Robot 3	Robot 4
1	Robot 1	127.18	133.35	4.85	4.47	4.84	4.55	4.34
	Robot 2	141.82	149.23	5.22				
	Robot 3	153.58	161.17	4.94				
	Robot 4	102.26	107.69	5.3				
2	Robot 1	119.62	123.28	3.05				
	Robot 2	152.27	159.76	4.91				
	Robot 3	147.86	155.64	5.26				
	Robot 4	115.73	119.78	3.49				
3	Robot 1	105.56	111.19	5.33				
	Robot 2	147.84	152.94	3.44				
	Robot 3	159.74	167.29	4.72				
	Robot 4	108.59	113.56	4.57				
4	Robot 1	109.73	114.33	4.19				
	Robot 2	144.42	151.37	4.81				
	Robot 3	167.37	175.45	4.82				
	Robot 4	105.52	109.89	4.14				
5	Robot 1	123.15	129.24	4.94				
	Robot 2	147.26	155.86	5.84				
	Robot 3	154.58	159.26	3.02				
	Robot 4	104.38	108.76	4.19				

Table 6.6 Time taken to reach from start to goal position (in ‘sec’, four robots)

Scenario		In simulation (time taken in ‘sec’)	In real time (time taken in ‘sec’)	% Error	Avg. % Error			
					Robot 1	Robot 2	Robot 3	Robot 4
1	Robot 1	12.84	13.35	3.97	4.94	4.63	4.67	4.68
	Robot 2	15.23	16.05	5.38				
	Robot 3	15.74	16.47	4.63				
	Robot 4	12.11	12.69	4.78				
2	Robot 1	12.67	13.28	4.81				
	Robot 2	14.99	15.78	5.27				
	Robot 3	16.09	16.83	4.59				
	Robot 4	11.56	12.06	4.32				
3	Robot 1	13.09	13.83	5.65				
	Robot 2	15.32	15.93	3.98				
	Robot 3	16.37	17.11	4.52				
	Robot 4	12.62	13.14	4.12				
4	Robot 1	12.67	13.28	4.81				
	Robot 2	14.99	15.78	5.27				
	Robot 3	16.09	16.83	4.59				
	Robot 4	11.56	12.06	4.32				
5	Robot 1	13.09	13.83	5.65				
	Robot 2	15.32	15.93	3.98				
	Robot 3	16.37	17.11	4.52				
	Robot 4	12.62	13.14	4.12				

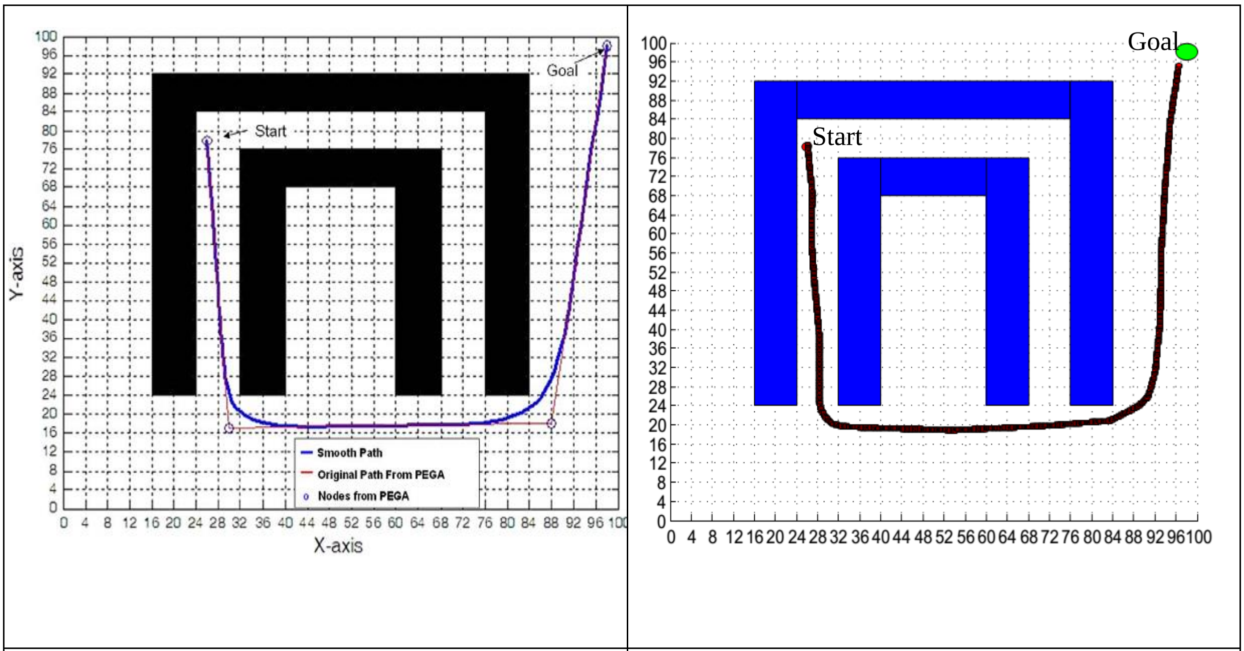
6.6 Comparison with Other Results

This section presents the comparison between the developed RBFN controller in the present study with the other models presented in the literature. In particular, RBFN controller has been compared with the results given by Tsai et al. [147] and Qu et al. [148]. Figure 6.6 (a)-(b) to figure 6.10 (a)-(b) show the simulation result for ABFO controller and results found by Tsai et al. [147] and Qu et al. [148]. Comparison of path length is shown in Tables 6.7-6.11.



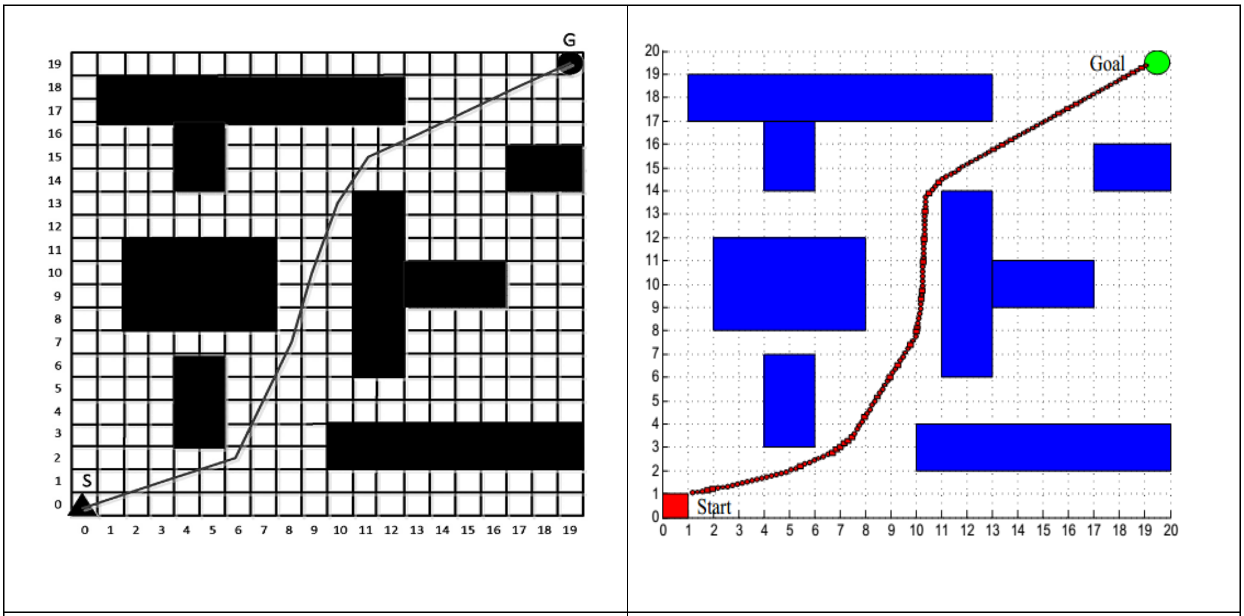
(a) (b)

Figure 6.6 (a) Simulation result by Tsai et al. [147]
(b) Simulation result by the proposed BA controller



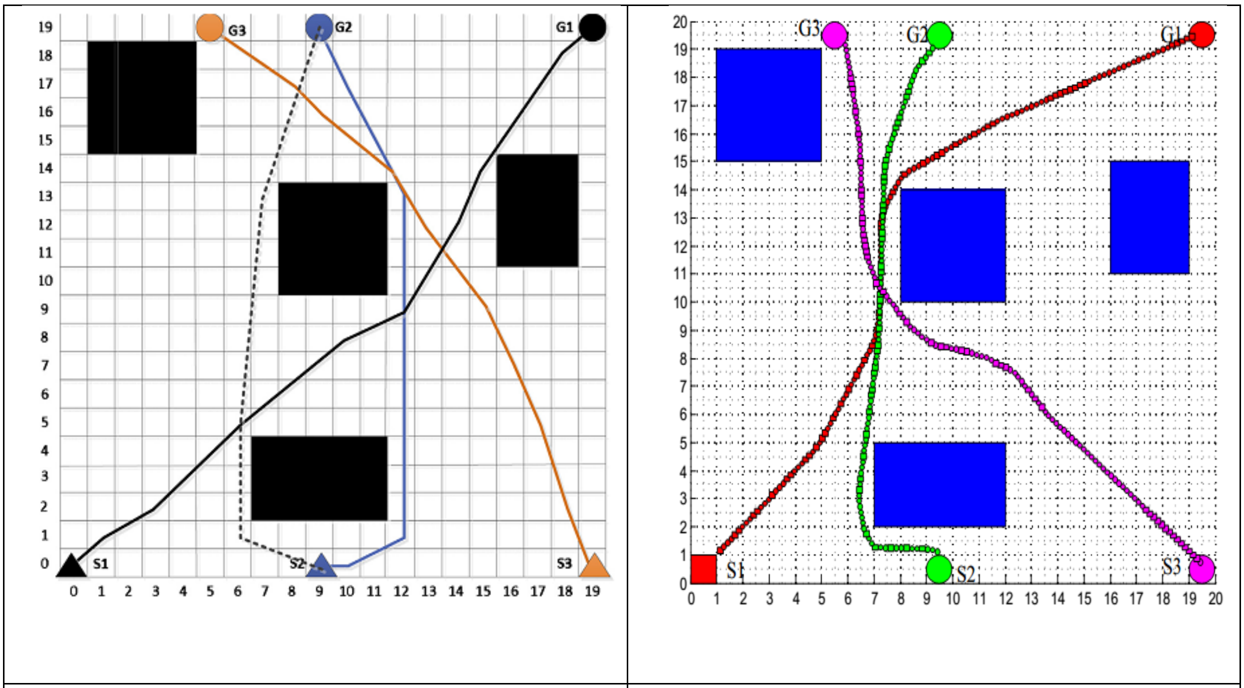
(a) (b)

Figure 6.7(a) Simulation result by Tsai et al. [147]
(b) Simulation result by the proposed BA controller



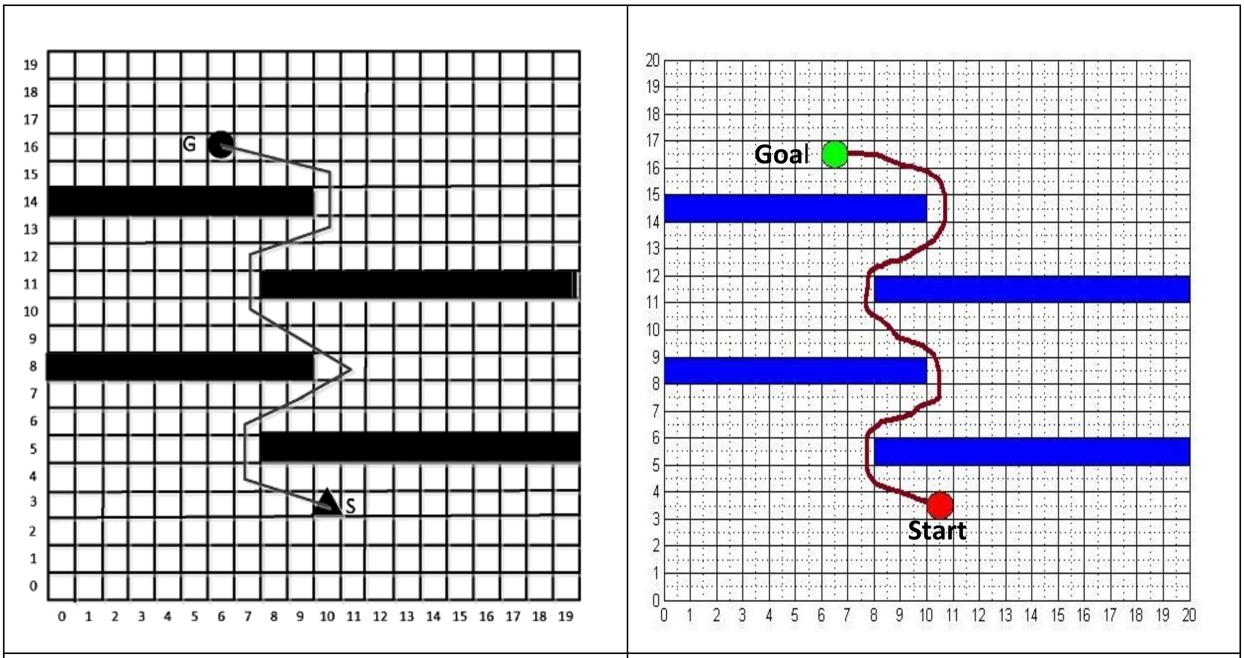
(a) (b)

Figure 6.8 (a) Simulation result by Qu et al. [148]
(b) Simulation result by the proposed BA controller



(a) (b)

Figure 6.9 (a) Simulation result by Qu et al. [148]
(b) Simulation result by the proposed BA controller



(a) (b)

Figure 6.10 (a) Simulation result by Qu et al. [148]
(b) Simulation result by the proposed BA controller

Table 6.7 Comparison of results between Tsai et al. [147] and developed BA controller.

Technique Used	Path length (in cm)	Deviation (in %)
Figure 6.6 (a)	23.92	1.14
Figure 6.6 (b)	23.65	

Table 6.8 Comparison of results between Tsai et al. [147] and developed BA controller.

Figure	Path length (in cm)	Deviation (in %)
Figure 6.7 (a)	24.06	0.75
Figure 6.7 (b)	23.88	

Table 6.9 Comparison of results between Qu et al. [148] and developed BA controller.

Figure	Path length (in cm)	Deviation (in %)
Figure 6.8 (a)	23.34	0.30
Figure 6.8 (b)	23.27	

Table 6.10 Comparison of results between Qu et al. [148] and developed BA controller.

Figure	Path length (in cm)		
	Robot1	Robot2	Robot3
Figure 6.9 (a)	20.44	15.84	17.7
Figure 6.9 (b)	20.28	15.50	17.60
Deviation (in %)	0.78	2.19	0.56

Table 6.11 Comparison of results between Qu et al. [148] and developed BA controller.

Figure	Path length (in cm)	Deviation (in %)
Figure 6.10 (a)	24.68	5.92
Figure 6.10 (b)	23.30	

It can be observed from the simulation results that proposed BA approach offers smooth paths as compared to results obtained by Tsai et al. [147] and Qu et al. [148]. Moreover, the path length is found to be less in current investigation as compared to Tsai et al. [147] and Qu et al. [148]. This shows the effectiveness of the proposed controller.

6.7 Summary

In this chapter, another nature inspired algorithm (i.e. Bess Algorithm) has been used to solve the issues of path planning of mobile robots. A new fitness function has been defined to make the appropriate decision during the navigation. Moreover controller has been tested in a number of exercises in simulation and physical environment to show the effectiveness of the developed controller. Further, the simulation and experimental results are compared and the average percentage errors are found to be within 6 %. Further, a comparison is made between the results obtained in present analysis and results by Tsai et al. [147] and Qu et al. [148]. It has been found that by proposed approach allows generation of smooth path as compared to approaches used by Tsai et al. [147] and Qu et al. [148]. Moreover, the path length is less in current study as compared to results obtained by Tsai et al. [147] and Qu et al. [148].

Chapter 7

HYBRID TECHNIQUES FOR NAVIGATION OF MOBILE ROBOTS

7.1 Introduction

This chapter deals with the navigation of mobile robots using five different hybrid techniques. They are as follows: 1) ABFO-RBFN Controller based technique, 2) ABFO-Bees Algorithm based technique, 3) RBFN-BA based technique, 4) BA-RBFN based technique, and 5) RBFN-BA-ABFO based technique. The output of the one approach is used as one of the input parameters to another approach to hybridize the two different approaches. The controller design, analysis and the execution using real mobile robots have been explained in details in the following sections of the chapter. Different exercises are presented, analyzed and compared in both simulated and the real environment to show the ability of the proposed controller in various situations.

7.1.1 Overview

In the prior chapters, three different approaches have been used alone for the navigation of mobile robots. Various simulation and experimental exercises have been presented to show the effectiveness of the proposed approaches. As a single approach does not guarantees success of the controller in posing optimal solution for all possible configurations, researchers are now aiming towards the development of hybrid controllers by combining the two or more different approaches. This hybridization permits the incorporation of multiple features of different approaches in a single controller.

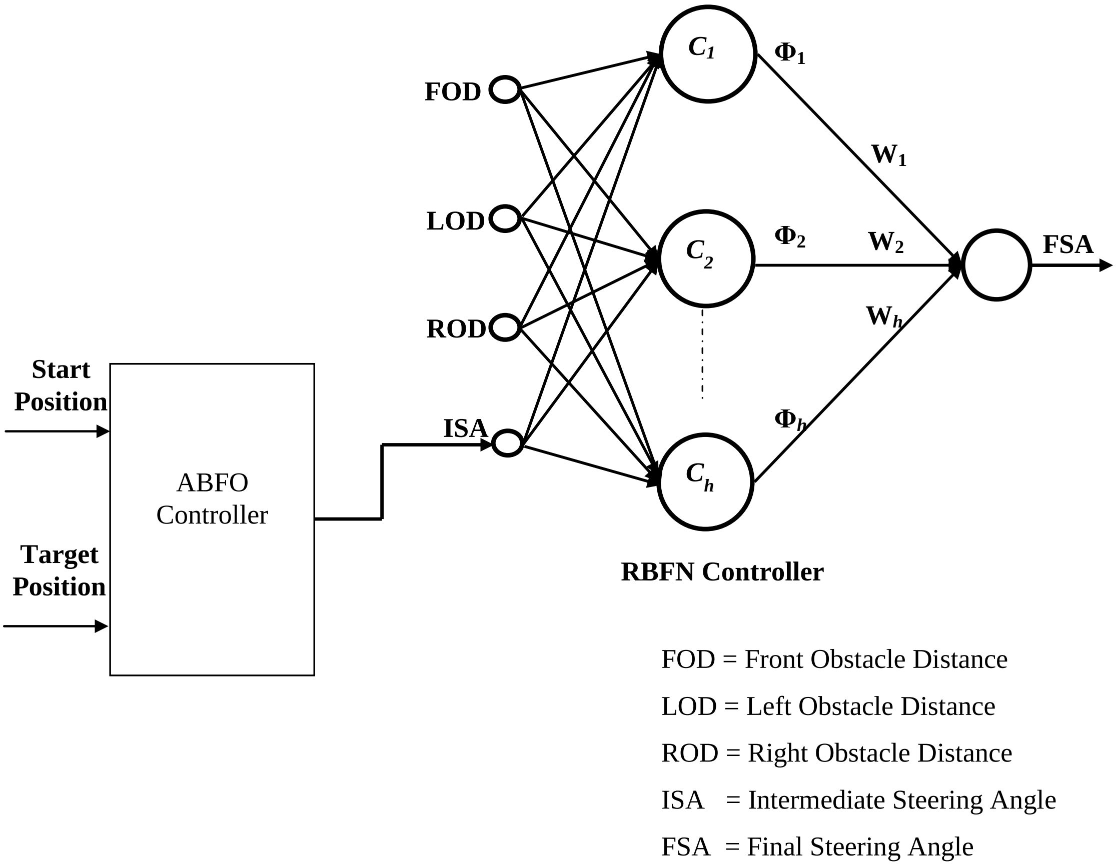
Hybrid techniques can be very effective in complex and dynamic environment with multiple robots. In case of multiple robots, inter robot collision avoidance is also required apart from target seeking and obstacle avoidance behavior. In these situations, a robust controller is required to make intelligent decisions in order to perform specific tasks. While developing hybrid techniques, initially two different techniques are selected and then hybridized in order to develop a better controller. Further, the hybrid controller is tested in different situations for optimal solution. Finally, the hybrid algorithms are embedded in the controller of the real robots to check the success of the developed approach in real environment.

7.1.2 Advantages of hybrid approaches

As a single approach does not guarantees success of the controller in posing near optimal solution for all possible configurations, researchers are now aiming towards the development of hybrid controllers by combining the two or more different approaches. This hybridization permits the incorporation of multiple features of different approaches in a single controller. This enables the success of the controller in different situations as compared to stand alone approaches. Therefore, hybrid controllers can be very effective in complex and dynamic environment with multiple robots. That is why, hybrid controllers have been taken into consideration in the present work.

7.2 Analysis of ABFO-RBFN and RBFN-ABFO Hybrid Controller for Navigation of Mobile Robots

In this section, path planning of the multiple mobile robots has been presented using two different hybrid controllers (i.e. ABFO-RBFN and RBFN-ABFO). The architecture of the proposed hybrid controllers are shown in figures 7.1 and 7.2. It can be seen that the architecture of the proposed controllers are developed by hybridizing the two standalone approaches (ABFO and RBFN) already discussed in the prior chapters. The first part of the ABFO-RBFN hybrid controller is ABFO which uses robot’s initial and goal positions as input and provides an intermediate steering angle (ISA) as the output. Further, the inputs to the second controller (i.e. RBFN) are the distance of obstacles in front, left and right direction with respect to robot’s current position and the intermediate steering angle obtained as output from the ABFO controller. An array of sensors is equipped on the robot to detect any obstacles inside the sensing range, and target position. The hybrid controller takes the control actions based on the sensory data received from different sensors mounted on the robot. To test the robot in real environment, program is embedded in the robot’s microcontroller to facilitate the free navigation.



FOD = Front Obstacle Distance

LOD = Left Obstacle Distance

ROD = Right Obstacle Distance

ISA = Intermediate Steering Angle

FSA = Final Steering Angle

Figure 7.1 Architecture of ABFO- RBFN Hybrid Controller

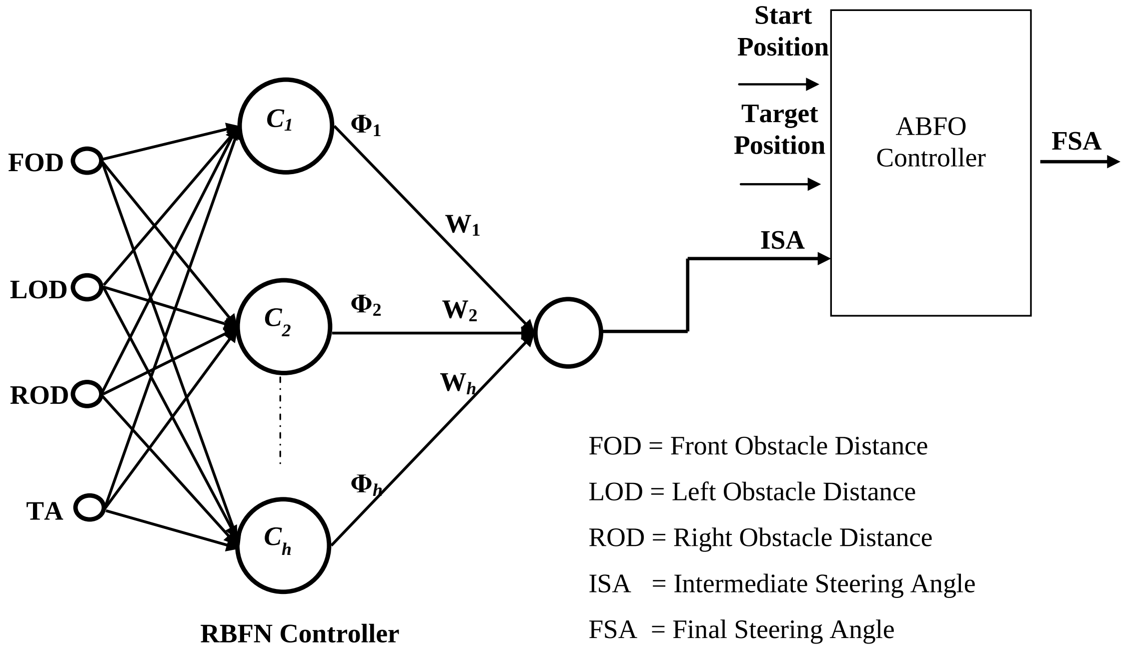

Start Position
Target Position
ISA
ABFO Controller
FSA
FOD
LOD
ROD
TA
C1
C2
Ch
Φ1
Φ2
Φh
W1
W2
Wh
RBFN Controller
FOD = Front Obstacle Distance
LOD = Left Obstacle Distance
ROD = Right Obstacle Distance
ISA = Intermediate Steering Angle
FSA = Final Steering Angle

Figure 7.2 Architecture of RBFN-ABFO Hybrid Controller

Similarly in RBFN-ABFO hybrid controller, RBFN controller gives an intermediate steering angle (ISA) as output based on the training pattern discussed in the chapter 5. Further, this intermediate steering angle will become the input to the BFO controller along with the other sensory information and control parameters. Finally, the BFO controller provides the final steering angle (FSA) to the robot for heading towards target position by following a trajectory which is smooth and free from obstacles. The simulation and experimental results for one, two and four robots for BFOA-RBFN and RBFN-BFOA hybrid controllers have been given in table 7.1-7.6.

7.3 Analysis of RBFN-BA and BA-RBFN Hybrid Controllers for Navigation of Mobile Robots

In this segment of the chapter, two hybrid controllers (i.e. RBFN-BA and BA-RBFN) have been presented for the path planning of the mobile robots. For this, RBFN controller has been hybridized with the BA and the architecture for RBFN-BA and BA-RBFN hybrid controllers are shown in figures 7.3 and 7.4 respectively. The simulation and experimental results for one, two and four robots for RBFN-BA and BA-RBFN hybrid controllers have been given in table 7.7-7.12.

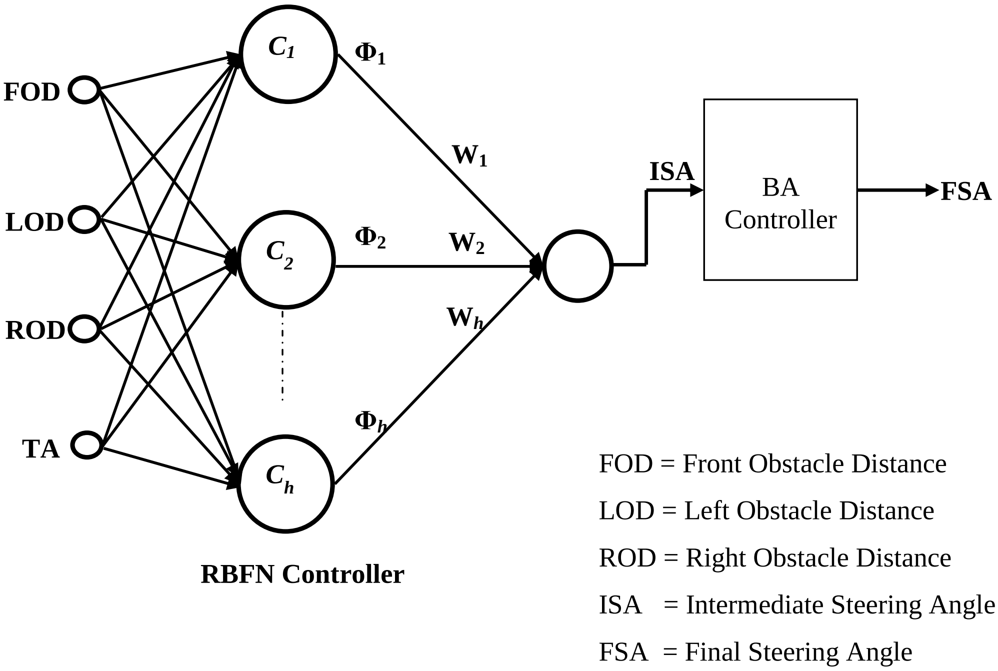

FOD
LOD
ROD
TA
C1
C2
Ch
Φ1
Φ2
Φh
W1
W2
Wh
ISA
BA Controller
FSA
RBFN Controller
FOD = Front Obstacle Distance
LOD = Left Obstacle Distance
ROD = Right Obstacle Distance
ISA = Intermediate Steering Angle
FSA = Final Steering Angle

Figure 7.3 Architecture of RBFN- BA Hybrid Controller

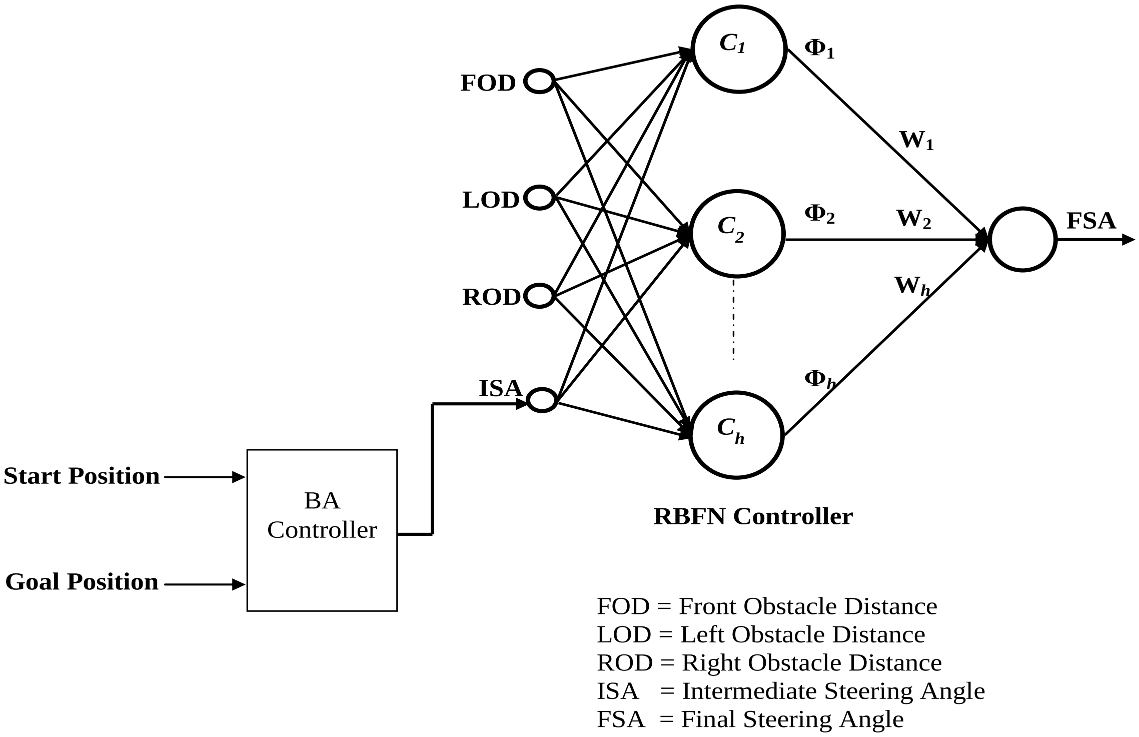

FOD
LOD
ROD
ISA
C1
C2
Ch
Φ1
Φ2
Φh
W1
W2
Wh
FSA
Start Position
Goal Position
BA Controller
RBFN Controller
FOD = Front Obstacle Distance
LOD = Left Obstacle Distance
ROD = Right Obstacle Distance
ISA = Intermediate Steering Angle
FSA = Final Steering Angle

Figure 7.4 Architecture of BA- RBFN Hybrid Controller

7.4 Analysis of RBFN-BA-ABFO Hybrid Controller for Navigation of Mobile Robots

Finally, all three approaches (i.e. ABFO, RBFN and BA) are hybridized to develop RBFN-BA-ABFO hybrid controller. The architecture of the hybrid controller is shown in figure 7.5. This work proposes a new approach for the design and development of intelligent mobile robot based on hybrid controller. It discusses the complex functional structure of such systems, providing solutions to some typical design problems. The system, on the one hand, offers some solutions for the problems related to the hybrid potential field building (like dead lock and local minima problem), and on the other hand, looks for the problem-solutions connected with self-organization of the mobile robots during navigation. It is decentralized and behavior-oriented, because the agents sharing the basic information about the positions and orientations between each other, and on the basis of these information they define the next possible position and orientation.

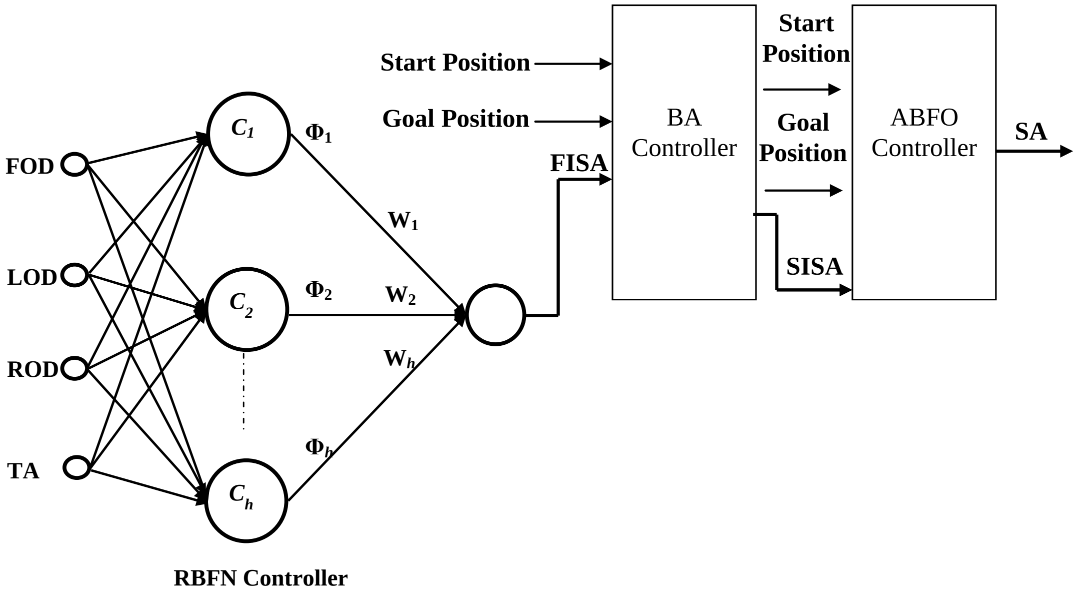


Figure 7.5 Architecture of RBFN-BA-ABFO Hybrid Controller

It is evolutionary self-organized, because the moving strategies are defined by a genetic algorithm and with the specified policies the near-optimal next possible move can be determined. To realize the controller in real sense the program is embedded in the robot.

7.5 Comparison of Results between the Standalone Approaches and Hybrid Approaches.

In this section, a comparison is made between the previously developed controllers (i.e ABFO, RBFN and BA) and hybrid controllers for the path length and time taken by the robots to reach from start to goal position. Tables 7.1, 7.3 and 7.5 show the comparison between simulation and experimental results for ABFO, RBFN, ABFO-RBFN, and RBFN-ABFO techniques with respect to distance travelled by one, two and four mobile robots while navigating from start to goal point. The time taken by the one, two and four robots in simulation and real time using different techniques (i.e. ABFO, RBFN, ABFO-RBFN, and RBFN-ABFO) are shown in Tables 7.2, 7.4 and 7.6.

In the same way, comparison between simulation and experimental results for RBFN, BA, RBFN-BA, and BA-RBFN techniques with respect to distance travelled by one, two and four mobile robots are given in tables 7.7, 7.9 and 7.11. The time taken by the one, two and four robots in simulation and real time using different techniques (i.e. RBFN, BA, RBFN-BA, and BA-RBFN) are shown in tables 7.8, 7.10, and 7.12 respectively.

Comparison between path length and time taken in simulation and experimental mode for RBFN-BA-ABFO hybrid controller and three standalone approaches (ABFO, RBFN and BA) using one, two and four mobile robots are presented in the tables 7.13- 7.18.

Table 7.1 Distance travelled by robot in simulation and real time to reach targets. (in 'cm', one robot)

Scenario	Navigation Approach							
	ABFO		RBFN		ABFO-RBFN Hybrid Controller		RBFN-ABFO Hybrid Controller	
	S	E	S	E	S	E	S	E
1	127.86	130.84	130.95	136.45	125.45	119.30	126.20	132.75
2	132.10	134.07	136.50	142.58	130.25	136.52	128.95	136.15
3	164.00	169.29	170.25	175.68	163.54	168.42	161.33	167.50
4	178.06	187.4	182.65	186.32	175.25	180.25	177.69	183.65
5	227.76	236.4	231.85	237.60	225.30	229.35	228.50	235.50

Table 7.2 Time taken by robot in simulation and real time to reach targets. (in 'sec', one robot)

Scenario	Navigation Approach							
	ABFO		RBFN		ABFO-RBFN Hybrid Controller		RBFN-ABFO Hybrid Controller	
	S	E	S	E	S	E	S	E
1	14.2	14.5	14.6	14.5	13.9	13.3	14.0	14.8
2	14.7	14.9	15.2	14.9	14.5	15.2	14.3	15.1
3	18.2	18.8	18.9	18.8	18.2	18.7	17.9	18.6
4	19.8	20.8	20.3	20.8	19.5	20.0	19.7	20.4
5	25.3	26.3	25.8	26.3	25.0	25.5	25.4	26.2

Table 7.3 Distance travelled by robot in simulation and real time to reach targets. (in 'cm', two robots)

Scenario		Navigation Approach							
		ABFO		RBFN		ABFO-RBFN Hybrid Controller		RBFN-ABFO Hybrid Controller	
		S	E	S	E	S	E	S	E
1	Robot 1	143.08	148.04	147.03	153.06	139.55	142.33	138.66	141.22
	Robot 2	121.37	126.35	124.23	130.45	118.55	122.33	116.11	118.41
2	Robot 1	139.62	144.69	143.66	150.21	137.2	140.02	138.99	141.22
	Robot 2	117.45	123.22	123.02	129.02	115.23	117.56	114.22	117.55
3	Robot 1	171.84	181.55	178.66	183.56	169.55	174.33	171.22	176.22
	Robot 2	148.29	153.33	153.66	158.22	144.55	147.66	145.99	148.55
4	Robot 1	163.58	167.85	169.2	174.21	161.25	163.55	159.22	162.39
	Robot 2	155.17	165.74	159.2	163.33	153.26	157.33	154.23	158.22
5	Robot 1	182.27	188.75	184.99	189.66	179.54	183.66	181.14	185.55
	Robot 2	163.53	168.35	167.89	171.23	162.33	166.22	164.22	167.55

Table 7.4 Time taken by robot in simulation and real time to reach targets. (in 'sec', 2 robots)

Scenario		Navigation Approach							
		ABFO		RBFN		ABFO-RBFN Hybrid Controller		RBFN-ABFO Hybrid Controller	
		S	E	S	E	S	E	S	E
1	Robot 1	15.9	16.4	16.3	17.0	15.5	15.8	15.4	15.7
	Robot 2	13.5	14.0	13.8	14.5	13.2	13.6	12.9	13.2
2	Robot 1	15.5	16.1	16.0	16.7	15.2	15.6	15.4	15.7
	Robot 2	13.1	13.7	13.7	14.3	12.8	13.1	12.7	13.1
3	Robot 1	19.1	20.2	19.9	20.4	18.8	19.4	19.0	19.6
	Robot 2	16.5	17.0	17.1	17.6	16.1	16.4	16.2	16.5
4	Robot 1	18.2	18.7	18.8	19.4	17.9	18.2	17.7	18.0
	Robot 2	17.2	18.4	17.7	18.1	17.0	17.5	17.1	17.6
5	Robot 1	20.3	21.0	20.6	21.1	19.9	20.4	20.1	20.6
	Robot 2	18.2	18.7	18.7	19.0	18.0	18.5	18.2	18.6

Table 7.5 Distance travelled by robot in simulation and real time to reach targets. (in 'cm', four robots)

Scenario		Navigation Approach							
		ABFO		RBFN		ABFO-RBFN Hybrid Controller		RBFN-ABFO Hybrid Controller	
		S	E	S	E	S	E	S	E
1	Robot 1	127.18	133.35	128.99	132.22	124.66	127.55	122.58	125.69
	Robot 2	141.82	149.23	143.66	145.27	137.55	141.21	135.88	139.55
	Robot 3	153.58	161.17	158.66	159.99	149.55	153.42	147.36	151.29
	Robot 4	102.26	107.69	104.33	107.55	99.98	103.21	97.14	101.11
2	Robot 1	119.62	123.28	121.55	126.31	117.54	120.21	115.56	118.54
	Robot 2	152.27	159.76	154.33	157.48	148.66	152.24	147.11	150.02
	Robot 3	147.86	155.64	149.47	152.44	143.55	147.54	140.89	144.21
	Robot 4	115.73	119.78	116.33	119.88	112.95	116.98	109.85	113.6
3	Robot 1	105.56	111.19	106.99	109.88	102.99	105.84	103.23	106.66
	Robot 2	147.84	152.94	149.55	152.78	145.12	148.59	147.01	150.22
	Robot 3	159.74	167.29	161.71	164.98	156.35	159.44	158.22	161.22
	Robot 4	108.59	113.56	111.23	113.65	104.66	108.99	106.11	109.99
4	Robot 1	109.73	114.33	111.25	114.22	107.33	114.99	106.44	108.85
	Robot 2	144.42	151.37	148.99	151.02	141.22	145.27	139.66	142.69
	Robot 3	167.37	175.45	171.22	175.19	165.06	168.94	163.55	168.54
	Robot 4	105.52	109.89	108.56	111.25	101.66	104.55	99.88	102.95
5	Robot 1	123.15	129.24	127.84	130.99	120.22	124.65	122.33	127.88
	Robot 2	147.26	155.86	151.69	155.99	142.35	145.87	143.59	147.65
	Robot 3	154.58	159.26	159.66	163.22	151.48	153.48	154.62	157.99
	Robot 4	104.38	108.76	108.18	112.99	101.65	105.55	103.38	106.99

Table 7.6 Time taken by robot in simulation and real time to reach targets. (in ‘sec’)

Scenario		Navigation Approach							
		ABFO		RBFN		ABFO-RBFN Hybrid Controller		RBFN-ABFO Hybrid Controller	
		S	E	S	E	S	E	S	E
1	Robot 1	14.1	14.8	14.3	14.7	13.9	14.2	13.6	14.0
	Robot 2	15.8	16.6	16.0	16.1	15.3	15.7	15.1	15.5
	Robot 3	17.1	17.9	17.6	17.8	16.6	17.0	16.4	16.8
	Robot 4	11.4	12.0	11.6	12.0	11.1	11.5	10.8	11.2
2	Robot 1	13.3	13.7	13.5	14.0	13.1	13.4	12.8	13.2
	Robot 2	16.9	17.8	17.1	17.5	16.5	16.9	16.3	16.7
	Robot 3	16.4	17.3	16.6	16.9	16.0	16.4	15.7	16.0
	Robot 4	12.9	13.3	12.9	13.3	12.6	13.0	12.2	12.6
3	Robot 1	11.7	12.4	11.9	12.2	11.4	11.8	11.5	11.9
	Robot 2	16.4	17.0	16.6	17.0	16.1	16.5	16.3	16.7
	Robot 3	17.7	18.6	18.0	18.3	17.4	17.7	17.6	17.9
	Robot 4	12.1	12.6	12.4	12.6	11.6	12.1	11.8	12.2
4	Robot 1	12.2	12.7	12.4	12.7	11.9	12.8	11.8	12.1
	Robot 2	16.0	16.8	16.6	16.8	15.7	16.1	15.5	15.9
	Robot 3	18.6	19.5	19.0	19.5	18.3	18.8	18.2	18.7
	Robot 4	11.7	12.2	12.1	12.4	11.3	11.6	11.1	11.4
5	Robot 1	13.7	14.4	14.2	14.6	13.4	13.9	13.6	14.2
	Robot 2	16.4	17.3	16.9	17.3	15.8	16.2	16.0	16.4
	Robot 3	17.2	17.7	17.7	18.1	16.8	17.1	17.2	17.6
	Robot 4	11.6	12.1	12.0	12.6	11.3	11.7	11.5	11.9

Table 7.7 Distance travelled by robot in simulation and real time to reach targets. (in ‘cm’, one robot)

Scenario	Navigation Approach							
	RBFN		BA		RBFN-BA Hybrid Controller		BA-RBFN Hybrid Controller	
	S	E	S	E	S	E	S	E
1	127.86	130.84	124.55	128.54	122.45	128.44	121.22	125.69
2	132.10	134.07	128.99	132.26	126.01	130.01	127.01	131.27
3	164.00	169.29	161.25	163.41	157.44	161.25	158.66	161.29
4	178.06	187.4	174.59	178.12	171.24	174.22	169.22	172.56
5	227.76	236.4	223.56	226.15	219.89	223.66	218.14	223.31

Table 7.8 Time taken by robot in simulation and real time to reach targets. (in ‘sec’, one robot)

Scenario	Navigation Approach							
	RBFN		BA		RBFN-BA Hybrid Controller		BA-RBFN Hybrid Controller	
	S	E	S	E	S	E	S	E
1	14.2	14.5	13.8	14.3	13.6	14.3	13.5	14.0
2	14.7	14.9	14.3	14.7	14.0	14.4	14.1	14.6
3	18.2	18.8	17.9	18.2	17.5	17.9	17.6	17.9
4	19.8	20.8	19.4	19.8	19.0	19.4	18.8	19.2
5	25.3	26.3	24.8	25.1	24.4	24.9	24.2	24.8

Table 7.9 Distance travelled by robot in simulation and real time to reach targets. (in ‘cm’, two robots)

Scenario		Navigation Approach							
		RBFN		BA		RBFN-BA Hybrid Controller		BA-RBFN Hybrid Controller	
		S	E	S	E	S	E	S	E
1	Robot 1	147.03	153.06	144.22	147.65	142.54	147.65	141.25	144.52
	Robot 2	124.23	130.45	121.33	124.15	119.65	124.51	118.57	122.65
2	Robot 1	143.66	150.21	141.25	144.27	138.95	142.31	139.54	143.55
	Robot 2	123.02	129.02	120.44	124.48	117.24	121.46	116.25	119.99
3	Robot 1	178.66	183.56	176.14	179.66	174.69	178.43	175.95	178.95
	Robot 2	153.66	158.22	150.20	154.28	146.25	150.01	145.01	148.94
4	Robot 1	169.2	174.21	167.54	170.02	163.59	169.22	161.03	165.68
	Robot 2	159.2	163.33	157.99	161.20	154.28	158.02	155.98	159.01
5	Robot 1	184.99	189.66	181.98	185.42	178.99	181.69	179.89	182.38
	Robot 2	167.89	171.23	164.92	168.25	161.56	164.29	160.35	164.25

Table 7.10 Time taken by robot in simulation and real time to reach targets. (in ‘sec’, 2 robots)

Scenario		Navigation Approach							
		RBFN		BA		RBFN-BA Hybrid Controller		BA-RBFN Hybrid Controller	
		S	E	S	E	S	E	S	E
1	Robot 1	16.3	17.0	16.0	16.4	15.8	16.4	15.7	16.1
	Robot 2	13.8	14.5	13.5	13.8	13.3	13.8	13.2	13.6
2	Robot 1	16.0	16.7	15.7	16.0	15.4	15.8	15.5	16.0
	Robot 2	13.7	14.3	13.4	13.8	13.0	13.5	12.9	13.3
3	Robot 1	19.9	20.4	19.6	20.0	19.4	19.8	19.6	19.9
	Robot 2	17.1	17.6	16.7	17.1	16.3	16.7	16.1	16.5
4	Robot 1	18.8	19.4	18.6	18.9	18.2	18.8	17.9	18.4
	Robot 2	17.7	18.1	17.6	17.9	17.1	17.6	17.3	17.7
5	Robot 1	20.6	21.1	20.2	20.6	19.9	20.2	20.0	20.3
	Robot 2	18.7	19.0	18.3	18.7	18.0	18.3	17.8	18.3

Table 7.11 Distance travelled by robot in simulation and real time to reach targets. (in 'cm', four robots)

Scenario		Navigation Approach							
		RBFN		BA		RBFN-BA Hybrid Controller		BA-RBFN Hybrid Controller	
		S	E	S	E	S	E	S	E
1	Robot 1	128.99	132.22	125.44	129.55	123.05	127.89	121.03	126.55
	Robot 2	143.66	145.27	141.58	141.02	139.65	143.29	137.88	141.02
	Robot 3	158.66	159.99	156.88	159.08	154.28	158.88	152.48	158.9
	Robot 4	104.33	107.55	102.44	106.99	101.02	107.89	99.35	105.01
2	Robot 1	121.55	126.31	119.55	124.06	117.01	122.45	118.9	122.01
	Robot 2	154.33	157.48	152.41	157.48	150.09	155.66	152.34	158.41
	Robot 3	149.47	152.44	147.25	154.99	146.02	149.54	148.01	153.19
	Robot 4	116.33	119.88	114.21	118.02	112.09	117.23	114.23	118.23
3	Robot 1	106.99	109.88	103.21	109.44	101.08	105.02	100.02	106.9
	Robot 2	149.55	152.78	147.54	152.08	145.78	149.02	143.05	148.23
	Robot 3	161.71	164.98	158.14	162.55	156.21	159.98	155.03	161.23
	Robot 4	111.23	113.65	107.95	111.25	105.06	109.54	104.09	109.02
4	Robot 1	111.25	114.22	108.01	112.58	105.14	108.69	106.99	110.23
	Robot 2	148.99	151.02	145.26	149.98	143.28	147.25	144.25	150.69
	Robot 3	171.22	175.19	168.35	172.59	165.01	169.24	166.38	172.55
	Robot 4	108.56	111.25	106.75	109.98	104.12	108.02	105.69	109.35
5	Robot 1	127.84	130.99	124.69	128.99	122.30	126.09	124.55	130.22
	Robot 2	151.69	155.99	148.59	152.69	147.01	150.02	148.51	152.69
	Robot 3	159.66	163.22	157.28	162.95	155.29	158.99	157.02	163.08
	Robot 4	108.18	112.99	105.09	110.02	103.28	108.23	104.56	110.23

Table 7.12 Time taken by robot in simulation and real time to reach targets. (in ‘sec’)

Scenario		Navigation Approach							
		RBFN		BA		RBFN-BA Hybrid Controller		BA-RBFN Hybrid Controller	
		S	E	S	E	S	E	S	E
1	Robot 1	14.3	14.7	13.9	14.4	13.7	14.2	13.4	14.1
	Robot 2	16.0	16.1	15.7	15.7	15.5	15.9	15.3	15.7
	Robot 3	17.6	17.8	17.4	17.7	17.1	17.7	16.9	17.7
	Robot 4	11.6	12.0	11.4	11.9	11.2	12.0	11.0	11.7
2	Robot 1	13.5	14.0	13.3	13.8	13.0	13.6	13.2	13.6
	Robot 2	17.1	17.5	16.9	17.5	16.7	17.3	16.9	17.6
	Robot 3	16.6	16.9	16.4	17.2	16.2	16.6	16.4	17.0
	Robot 4	12.9	13.3	12.7	13.1	12.5	13.0	12.7	13.1
3	Robot 1	11.9	12.2	11.5	12.2	11.2	11.7	11.1	11.9
	Robot 2	16.6	17.0	16.4	16.9	16.2	16.6	15.9	16.5
	Robot 3	18.0	18.3	17.6	18.1	17.4	17.8	17.2	17.9
	Robot 4	12.4	12.6	12.0	12.4	11.7	12.2	11.6	12.1
4	Robot 1	12.4	12.7	12.0	12.5	11.7	12.1	11.9	12.2
	Robot 2	16.6	16.8	16.1	16.7	15.9	16.4	16.0	16.7
	Robot 3	19.0	19.5	18.7	19.2	18.3	18.8	18.5	19.2
	Robot 4	12.1	12.4	11.9	12.2	11.6	12.0	11.7	12.2
5	Robot 1	14.2	14.6	13.9	14.3	13.6	14.0	13.8	14.5
	Robot 2	16.9	17.3	16.5	17.0	16.3	16.7	16.5	17.0
	Robot 3	17.7	18.1	17.5	18.1	17.3	17.7	17.4	18.1
	Robot 4	12.0	12.6	11.7	12.2	11.5	12.0	11.6	12.2

Table 7.13 Comparison between simulated and experimental performance with respect to path length (in 'cm', one robot)

Scenario	ABFO		RBFN		BA		RBFN-BA-ABFO Hybrid Controller	
	S	E	S	E	S	E	S	E
1	128.20	131.40	129.1	133.30	129.20	133.50	127.80	131.20
2	132.50	134.70	133.2	138.50	132.80	135.90	131.70	134.10
3	145.50	151.30	147.60	153.90	148.50	153.60	142.80	146.40
4	135.80	140.50	139.50	144.50	138.40	142.60	132.40	138.40
5	170.60	175.60	174.30	180.40	172.60	176.70	170.10	175.20

Table 7.14 Comparison between simulated and experimental performance with respect to time taken (in 'sec', one robot)

Scenario	ABFO		RBFN		BA		RBFN-BA-ABFO Hybrid Controller	
	S	E	S	E	S	E	S	E
1	14.2	14.6	14.3	14.8	14.4	14.8	14.2	14.6
2	14.7	15.0	14.8	15.4	14.8	15.1	14.6	14.9
3	16.2	16.8	16.4	17.1	16.5	17.1	15.9	16.3
4	15.1	15.6	15.5	16.1	15.4	15.8	14.7	15.4
5	19.0	19.5	19.4	20.0	19.2	19.6	18.9	19.5

Table 7.15 Comparison between simulated and experimental performance with respect to path length (in 'cm', two robots)

Scenario		ABFO		RBFN		BA		RBFN-BA-ABFO Hybrid Controller	
		S	E	S	E	S	E	S	E
1	Robot 1	148.09	155.71	150.30	156.51	149.65	155.68	147.30	152.80
	Robot 2	121.30	128.60	124.23	130.45	125.85	131.45	120.20	123.50
2	Robot 1	211.40	218.90	214.40	220.50	213.50	220.30	209.70	215.80
	Robot 2	170.50	177.60	174.80	182.50	175.60	182.40	168.60	174.90
3	Robot 1	167.20	172.60	170.60	178.60	170.50	178.50	164.80	185.30
	Robot 2	137.80	144.20	142.50	148.60	140.60	147.10	136.80	154.60

Table 7.16 Comparison between simulated and experimental performance with respect to time taken (in ‘sec’, two robots)

Scenario		ABFO		RBFN		BA		RBFN-BA-ABFO Hybrid Controller	
		S	E	S	E	S	E	S	E
1	Robot 1	16.5	17.3	16.7	17.4	16.6	17.3	16.4	17.0
	Robot 2	13.5	14.3	13.8	14.5	14.0	14.6	13.4	13.7
2	Robot 1	23.5	24.3	23.8	24.5	23.7	24.5	23.3	24.0
	Robot 2	18.9	19.7	19.4	20.3	19.5	20.3	18.7	19.4
3	Robot 1	18.6	19.2	19.0	19.8	18.9	19.8	18.3	20.6
	Robot 2	15.3	16.0	15.8	16.5	15.6	16.3	15.2	17.2

Table 7.17 Comparison between simulated and experimental performance with respect to path length (in ‘cm’, four robots)

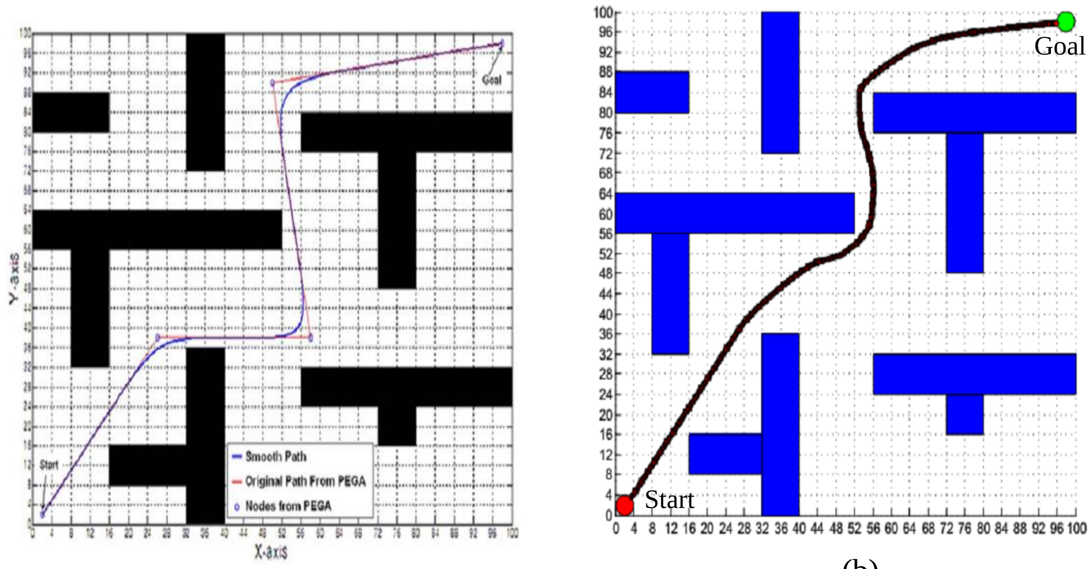
Scenario		ABFO		RBFN		BA		RBFN-BA-ABFO Hybrid Controller	
		S	E	S	E	S	E	S	E
1	Robot 1	127.62	134.19	129.45	136.12	128.50	135.09	125.25	132.20
	Robot 2	134.82	140.84	136.45	142.65	135.65	141.25	132.45	136.47
	Robot 3	144.94	151.33	147.63	153.10	146.26	152.68	142.56	145.36
	Robot 4	108.48	114.20	110.23	116.56	109.35	115.98	106.85	110.68
2	Robot 1	115.56	121.59	119.60	125.85	118.80	124.70	113.85	118.50
	Robot 2	135.84	142.34	138.85	145.68	139.75	143.58	134.20	140.52
	Robot 3	149.74	156.29	154.65	161.52	155.85	160.54	147.85	153.50
	Robot 4	109.59	115.56	114.63	120.40	116.85	123.20	109.20	114.52

Table 7.18 Comparison between simulated and experimental performance with respect to time taken (in ‘sec’, four robots)

Scenario		ABFO		RBFN		BA		RBFN-BA-ABFO Hybrid Controller	
		S	E	S	E	S	E	S	E
1	Robot 1	11.7	12.3	14.3	15.1	14.2	15.0	13.9	14.6
	Robot 2	13.8	14.5	15.1	15.8	15.0	15.6	14.7	15.1
	Robot 3	13.9	14.6	16.4	17.0	16.2	16.9	15.8	16.1
	Robot 4	12.4	13.0	12.2	12.9	12.1	12.8	11.8	12.2
2	Robot 1	12.8	13.5	13.3	14.0	13.2	13.9	12.7	13.2
	Robot 2	15.1	15.8	15.4	16.2	15.5	16.0	14.9	15.6
	Robot 3	16.6	17.4	17.2	17.9	17.3	17.8	16.4	17.1
	Robot 4	12.2	12.8	12.7	13.4	13.0	13.7	12.1	12.7

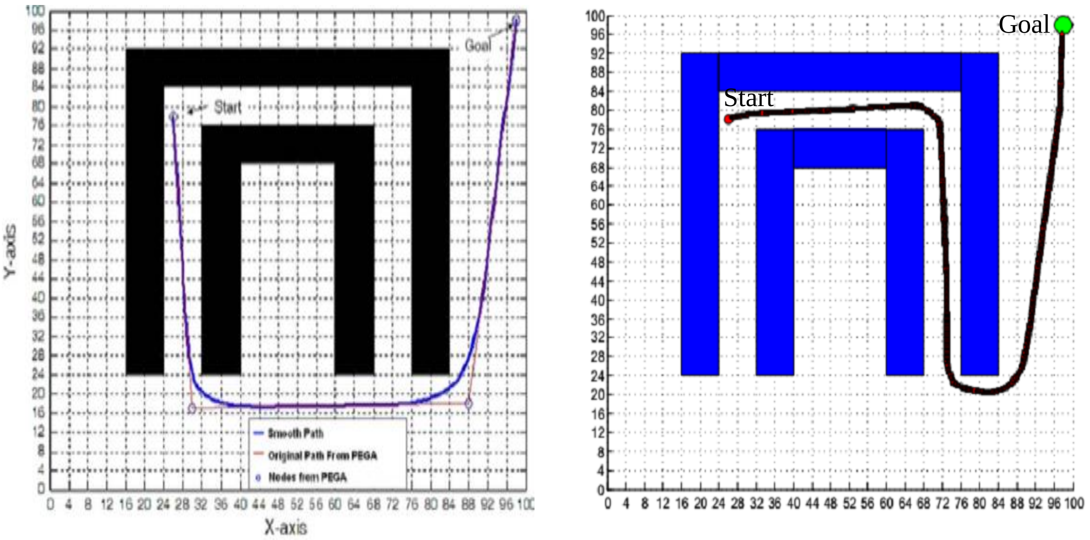
7.6 Comparison with Other Results

This section presents the comparison between the developed hybrid controllers in the present study with the other models presented in the literature. In particular, RBFN-BA-ABFO hybrid controller has been compared with the results given by Tsai et al. [147] and Qu et al. [148]. Figure 7.6 (a)-(b) to figure 7.10 (a)-(b) shows the simulation result for RBFN-BA-ABFO hybrid controller and result found by Tsai et al. [147] and Qu et al. [148]. Comparison of path length is shown in Tables 7.19-7.23.



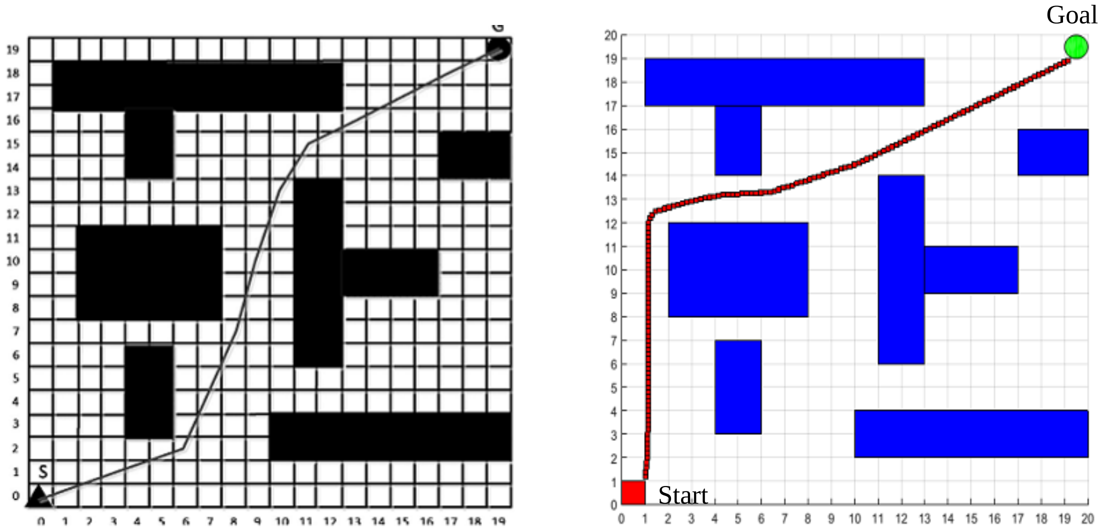
(a) (b)

Figure 7.6 (a) Simulation result by Tsai et al. [147]
(b) Simulation result by the proposed RBFN-BA-ABFO hybrid controller.



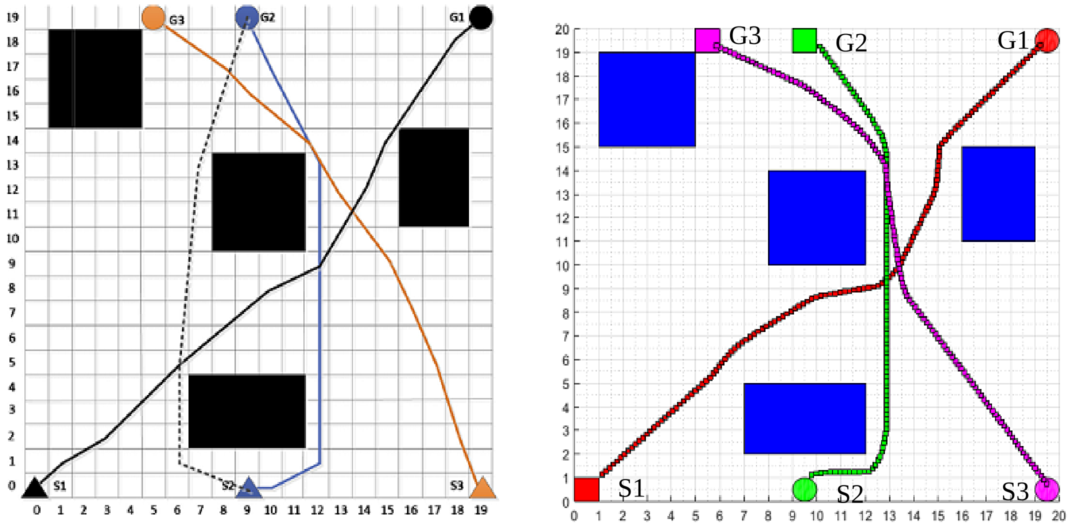
(a) (b)

Figure 7.7 (a) Simulation result by Tsai et al. [147]
(b) Simulation result by the proposed RBFN-BA-ABFO hybrid controller.



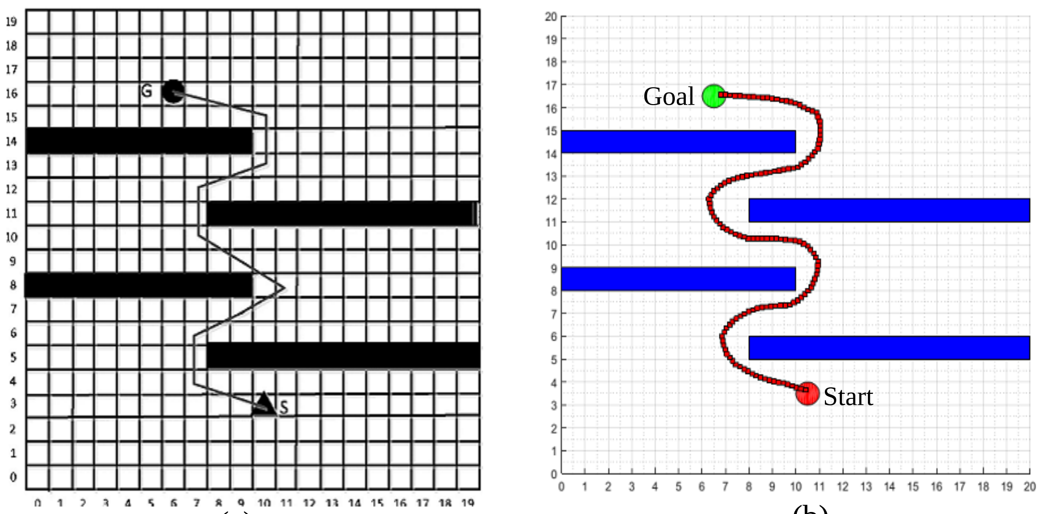
(a) (b)

Figure 7.8 (a) Simulation result by Qu et al. [148]
(b) Simulation result by the proposed RBFN-BA-ABFO hybrid controller.



(a) (b)

Figure 7.9 (a) Simulation result by Qu et al. [148]
(b) Simulation result by the proposed RBFN-BA-ABFO hybrid controller.



(a) (b)

Figure 7.10 (a) Simulation result by Qu et al. [148]
(b) Simulation result by the proposed RBFN-BA-ABFO hybrid controller.

Table 7.19 Comparison of results between Tsai et al. [147] and developed RBFN-BA-ABFO hybrid controller.

Figure	Path length (in cm)	Deviation (in %)
Figure 7.6 (a)	23.92	11.44
Figure 7.6 (b)	21.46	

Table 7.20 Comparison of results between Tsai et al. [147] and developed RBFN-BA-ABFO hybrid controller.

Figure	Path length (in cm)	Deviation (in %)
Figure 7.7 (a)	24.06	8.87
Figure 7.7 (b)	22.10	

Table 7.21 Comparison of results between Qu et al. [148] and developed RBFN-BA-ABFO hybrid controller.

Figure	Path length (in cm)	Deviation (in %)
Figure 7.8 (a)	23.34	2.05
Figure 7.8 (b)	22.79	

Table 7.22 Comparison of results between Qu et al. [148] and developed RBFN-BA-ABFO hybrid controller.

Figure	Path length (in cm)		
	Robot1	Robot2	Robot3
Figure 7.9 (a)	20.44	15.84	17.7
Figure 7.9 (b)	19.9	15.09	17.25
Deviation (in %)	2.7	4.93	2.64

Table 7.23 Comparison of results between Qu et al. [148] and developed RBFN-BA-ABFO hybrid controller.

Figure	Path length (in cm)	Deviation (in %)
Figure 7.10 (a)	24.68	11
Figure 7.10 (b)	22.23	

7.7 Summary

In the present work five different hybrid approaches (i.e. ABFO-RBFN, RBFN-ABFO, RBFN-BA, BA-RBFN and RBFN-BA-ABFO) have been analysed. All the developed hybrid approaches have been tested in different scenario for single and multiple robots. Further, all approaches are compared with each other with respect to path length and time taken. From the results, it has found that path length and time taken are comparatively less in hybrid approaches than the standalone approaches for different exercises. Results show that RBFN-BA-ABFO hybrid controller is found most effective among all the approaches in the present study. The hybrid controller is capable of generating collision free smooth trajectory from start to goal position. During comparison of simulation and experimental results the percentage of errors are found to be within 6 percent. Furthermore, comparisons of simulation results are made between the developed RBFN-BA-ABFO hybrid controller with results by Tsai et al. [146] and Qu et al. [147].

Chapter 8

CONCLUSION AND FUTURE DIRECTIONS

The prior chapters report the background, motivation, problem formulation and approaches for intelligent control and path planning of mobile robots in cluttered environments. This chapter summarizes the main contribution and conclusions of the present research work. Further, scope of future work is proposed at the end of the chapter.

8.1 Contributions of the Present Work

The prime objective of this work concerns with design and development of the controllers by using artificial intelligence techniques. In the current work, different controllers have been designed and analyzed to show their effectiveness in various environments. In order to achieve autonomous navigation, standalone AI techniques and their hybrid models have been proposed. In particular, the present work deals with the Bacterial Foraging Optimization Algorithm (BFOA), Radial Basis Function Network (RBFN) and Bees Algorithm (BA) to solve the basic issues related to the path planning and navigation control of the mobile robots. The main contributions of this research are as follows:

1. Initially, an intelligent controller inspired from the bacterial foraging optimization algorithm (BFOA) has been designed, analyzed and tested for a number of situations to show the effectiveness of the proposed BFOA controller. The proposed controller uses input data from different sensors mounted on the robot to perceive the environment, positions of the obstacles and bearing of the goal position. Based on the information, controller directs the robot to its next position via steering angle as output. The decision scheme permits the robot to choose next position from a number of positions, such that the robot reaches to the goal position without colliding with the obstacles present along the path.

Further, adaptive bacterial foraging optimization algorithm (ABFO) is analyzed to solve the problem of mobile robot navigation. The approach is same as the classical BFOA except that the step size during the chemotaxis is made adaptive. Control scheme and criteria for the selection of the proper step size are defined.

A series of exercises are presented in various simulated and physical environments. From the results obtained, it has been seen that the robots are successfully avoiding the obstacles and finding the respective goal positions. Moreover, robots follow different behaviour such as wall following and goal seeking efficiently.

2. Radial basis function (RBF) neural controller has been used to develop another controller for the navigation of mobile robots. The proposed RBF neural controller uses obstacles distances with respect to its current position as input along with the bearing of the goal position. The network is trained with the help of large training data set and training patterns for the prediction of a suitable value of the steering angle. Depending upon the input data set controller provides an appropriate value of steering angle to reach the target position.

3. A nature inspired algorithm motivated from the food foraging behaviour of the honey bees is used to solve the problem of mobile robot navigation. Fitness function has been designed for the selection of feasible path between the start and goal position.

4. At last, apart from the standalone approaches, hybrid models have been developed for the navigation of mobile robots. All three approaches are hybridized with each other to have the benefits of each standalone approach in hybrid models. The developed techniques are tested in both simulation and real time environments. Moreover, the proposed hybrid approaches are compared with the previous results to show the effectiveness of the proposed controller.

8.2 Conclusion

The results obtained from a series of simulation and experimental investigation, it has found that all the proposed approaches are able to solve the issues with the navigation of mobile robots. The proposed approaches have been tested for single and multiple robots with same or different goal positions in various environments.

- The ABFO technique has better performance than BFOA approach. BFO-RBFN and RBFN-BFO hybrid approaches are found better in term of distance travelled and time taken to reach the goal position as compared to both the standalone BFO and RBFN approach.
- Similarly the RBFN-BA and BA-RBFN approaches perform better than the simple BA or RBFN technique.
- The RBFN-BA-BFO hybrid approach is most effective in terms of path length and time taken over the other approaches (i.e. ABFO, RBFN, and BA) developed in this study. The success of hybrid controller demonstrates the advantage of combing different techniques as it facilitates the integration of multiple features of different approaches in a single controller.
- During comparison between then simulation and experimental results, the average errors are found to be within 6 % in terms of path length and time taken during navigation from start to goal point.

References

1. Andrews, J. R. and Hogan, N., Impedance Control of Manufacturing Processes and Robotic Systems, Eds. Hardt, D. E. and Book, W., ASME, Boston, 1983, 243-251.
2. Khatib, O., Real-time obstacle avoidance for manipulators and mobile robots, *The International Journal of Robotics Research*, 5(1), 1986, 90-98.
3. Koren, Y. and Johann, B., Potential field methods and their inherent limitations for mobile robot navigation, *Robotics and Automation*, Proceedings, IEEE International Conference,9-11 April 1991, Carlifonia,2, 1991,1398-1404.
4. Barraquand, J., Langlois, B. and Latombe, J.C., Numerical potential field techniques for robot path planning, *IEEE Transactions on Systems, Man and Cybernetics*, 22(2), 1992, 224-241.
5. McFetridge, L. and Yousef Ibrahim, M., New technique of mobile robot navigation using a hybrid adaptive fuzzy potential field approach, *Computers & Industrial engineering*, 35(3) ,1998, 471-474.
6. Wang, Y. and Gregory S. C., A new potential field method for robot path planning, *IEEE International Conference on Robotics and Automation*, Sanfrancio, CA, 2, 2000, 977-982.
7. Cosio, F. A. and MA P. C., Autonomous robot navigation using adaptive potential fields, *Mathematical and Computer Modeling*, 40(9), 2004, 1141-1156.
8. Li, Y.C. and Ke-Zhong H., A novel obstacle avoidance and navigation method for outdoor mobile robot based on laser radar, *Jiqiren Robot*, 28(3), 2006, 275-278.
9. Hassanzadeh, I., Hamid G., and Reza D., Design and implementation of a simple fuzzy algorithm for obstacle avoidance navigation of a mobile robot in dynamic environment, *Mechatronics and its Applications*, ISMA 2008, 5th International Symposium on. IEEE, 27-29 May, 2008, Amman, 2008,1-6.
10. 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.

11. Do, K. D., Output-feedback formation tracking control of unicycle-type mobile robots with limited sensing ranges, *Robotics and Autonomous Systems*, 57(1), 2009, 34-47.

12. Takahashi, M., Suzuki, T., Shitamoto, H., Moriguchi, T., and Yoshida, K. ,Developing a mobile robot for transport applications in the hospital domain, *Robotics and Autonomous Systems*, 58(7), 2010, 889-899.

13. Jaradat, M. A. K., Mohammad Al-R. and Lara Q., Reinforcement based mobile robot navigation in dynamic environment, *Robotics and Computer-Integrated Manufacturing*, 27(1), 2011, 135-149.

14. Meng, M. and Avinash C. K., Mobile robot navigation using neural networks and nonmetrical environmental models, *Control Systems*, IEEE 13(5), 1993, 30-39.

15. Gu, D. and Huosheng H., Neural predictive control for a car-like mobile robot, *Robotics and Autonomous Systems*, 39(2), 2002, 73-86.

16. Oh, C., Min-Soeng, K. and Ju-Jang L., Control of a nonholonomic mobile robot using an RBF network, *Artificial Life and Robotics*, 8(1), 2004, 14-19.

17. Mesbahi, L., Benyettou, A., Hendel, F., and Berrached, N. E., Reactive navigation of mobile robot by temporal radial basis function approach, TENCON 2004, 2004 IEEE Region 10 Conference, 500, 2004.

18. Harb, M., Abielmona, R., Naji, K., and Petriu, E.,Neural networks for environmental recognition and navigation of a mobile robot, *Instrumentation and Measurement Technology Conference Proceedings*, 2008. IMTC 2008. IEEE. IEEE, 2008.

19. Engedy, I. and Gábor H., Artificial neural network based mobile robot navigation, *Intelligent Signal Processing*, 2009, WISP 2009. IEEE International Symposium on. IEEE, 2009.

20. Wahab, W., Autonomous mobile robot navigation using a dual artificial neural network, TENCON 2009-2009 IEEE Region 10 Conference. IEEE, 2009.

21. Markoski, B., Vukosavljev, S., Kukolj, D., and Pletl, S., Mobile robot control using self-learning neural network, *Intelligent Systems and Informatics*, 2009. SISY'09. 7th International Symposium on. IEEE, 2009.

22. Fernandez-Leon, J.A., Gerardo, G. A. and Miguel A. M., Behavioral control through evolutionary neurocontrollers for autonomous mobile robot navigation, *Robotics and Autonomous Systems*, 57(4), 2009, 411-419.

23. Rossomando, F G., Carlos S. and Ricardo C. , Autonomous mobile robots navigation using RBF neural compensator, *Control Engineering Practice* 19(3), 2011, 215-222.

24. Mohareri, O., Rached D. and Ahmad B. Rad., Indirect adaptive tracking control of a nonholonomic mobile robot via neural networks, *Neurocomputing*, 88, 2012, 54-66.

25. Ghommam, J., Mehrjerdi, H. and Saad, M., Robust formation control without velocity measurement of the leader robot, *Control Engineering Practice*, 21(8), 2013, 1143-1156.

26. Praczyk, T., Neural anti-collision system for Autonomous Surface Vehicle, *Neurocomputing*, 149, 2015, 559-572.

27. Aguirre, E., and Antonio, G., Fuzzy behaviors for mobile robot navigation: design, coordination and fusion, *International Journal of Approximate Reasoning*, 25(3), 2000, 255-289.

28. Aguirre, E. and Antonio G., A fuzzy perceptual model for ultrasound sensors applied to intelligent navigation of mobile robots, *Applied Intelligence*, 19(3), 2003, 171-187.

29. Xu, W. L., A virtual target approach for resolving the limit cycle problem in navigation of a fuzzy behavior-based mobile robot, *Robotics and Autonomous Systems*, 30(4), 2000, 315-324.

30. Kang, D. O., Kim, S. H., Lee, H., and Bien, Z., Multiobjective navigation of a guide mobile robot for the visually impaired based on intention inference of obstacles, *Autonomous Robots*, 10(2), 2001, 213-230.

31. Nanayakkara, D. T., Watanabe, K., Kiguchi, K., and Izumi, K, Evolutionary learning of a fuzzy behavior based controller for a nonholonomic mobile robot in a class of dynamic environments, *Journal of Intelligent and Robotic Systems*, 32(3), 2001, 255-277.

32. Maaref, H. and Claude, B., Sensor-based navigation of a mobile robot in an indoor environment, *Robotics and Autonomous systems*, 38(1), 2002, 1-18.

33. Yavuz, H., and Bradshaw, A. A new conceptual approach to the design of hybrid control architecture for autonomous mobile robots, *Journal of Intelligent and Robotic Systems*, 34(1), 2002, 1-26.

34. Cuesta, F., Ollero, A., Arrue, B. C. and Braunstingl, R., Intelligent control of nonholonomic mobile robots with fuzzy perception, *Fuzzy Sets and Systems*, 134(1), 2003, 47-64.

35. Al-Khatib, M. and Jean J. S., An efficient data-driven fuzzy approach to the motion planning problem of a mobile robot, *Fuzzy Sets and Systems* 134(1), 2003, 65-82.

36. Lee, T.L. and Chia-Ju, W., Fuzzy motion planning of mobile robots in unknown environments, *Journal of Intelligent and Robotic Systems*, 37(2) 2003, 177-191.

37. Abdessemed, F., Khier B. and Eric M., A fuzzy-based reactive controller for a non-holonomic mobile robot, *Robotics and autonomous Systems*, 47(1), 2004, 31-46.

38. Demirli, K. and Mohammad M., Fuzzy dynamic localization for mobile robots, *Fuzzy Sets and Systems*, 144(2), 2004, 251-283.

39. Parhi, D. R., Navigation of mobile robots using a fuzzy logic controller, *Journal of intelligent and robotic systems*, 42(3), 2005, 253-273.

40. Mendez, M.A. O. and Juan A. F. M., Fuzzy logic user adaptive navigation control system for mobile robots in unknown environments, *Intelligent Signal Processing*, 2007. WISP 2007. IEEE International Symposium on. IEEE, 2007.

41. Innocenti, B., Beatriz L. and Joaquim S., A multi-agent architecture with cooperative fuzzy control for a mobile robot, *Robotics and Autonomous Systems,* 55(12), 2007, 881-891.

42. Wang, M. and James NK L., Fuzzy logic-based real-time robot navigation in unknown environment with dead ends, *Robotics and Autonomous Systems*, 56(7),2008, 625-643.

43. Huq, R., George KI M. and Raymond G. G., Mobile robot navigation using motor schema and fuzzy context dependent behavior modulation, *Applied soft computing*, 8(1), 2008, 422-436.

44. He, K., Hua S. and Wanjuan C., Application of fuzzy neural network based on TS model for mobile robot to avoid obstacles, *Intelligent Control and Automation*, 2008. WCICA 2008. 7th World Congress on. IEEE, 2008.

45. Gueaieb, W. and Miah, M. S., An intelligent mobile robot navigation technique using RFID technology, *IEEE Transactions on Instrumentation and Measurement*, 57(9), 2008, 1908-1917.

46. Martínez, R., Oscar C. and Luis T. A., Optimization of interval type-2 fuzzy logic controllers for a perturbed autonomous wheeled mobile robot using genetic algorithms, *Information Sciences*, 179(13), 2009, 2158-2174.

47. Chen, C.Y., Tzuu-Hseng S. L. and Ying-Chieh Y., EP-based kinematic control and adaptive fuzzy sliding-mode dynamic control for wheeled mobile robots, *Information Sciences*, 179(1), 2009, 180-195.

48. Motlagh, O. R. E., Tang S.H. and Napsiah I., Development of a new minimum avoidance system for a behavior-based mobile robot, *Fuzzy Sets and Systems*, 160(13), 2009, 1929-1946.

49. Doitsidis, L., Tsourveloudis, N. C. and Piperidis.S., Evolution of fuzzy controllers for robotic vehicles: the role of fitness function selection, *Journal of Intelligent and Robotic Systems*, 56(4), 2009, 469-484.

50. Nurmaini, S., Siti, Z. M. H. and Dayang N. A. J., Environmental recognition using RAM-network based type-2 fuzzy neural for navigation of mobile robot, *Computer and Automation Engineering*, 2009. ICCAE'09. International Conference on. IEEE, 2009.

51. Norouzi, M., Karambakhsh, A., Namazifar,M. and Savkovic. B., Object based navigation of mobile robot with obstacle avoidance using fuzzy controller, *IEEE International Conference on Control and Automation*, December 9-11, 2009,Newzland, 3,169-174, 2009.

52. Abiyev, R., Ibrahim, D. and Erin, B., Navigation of mobile robots in the presence of obstacles, *Advances in Engineering Software*, 41(10), 2010, 1179-1186.

53. Herrero-Pérez, D., Martínez-Barbera, H., LeBlanc, K. and Saffiotti, A., Fuzzy uncertainty modeling for grid based localization of mobile robots, *International Journal of Approximate Reasoning*, 51(8), 2010, 912-932.

54. Tzafestas, S. G., Deliparaschos, K. M. and Moustris, G. P., Fuzzy logic path tracking control for autonomous non-holonomic mobile robots: Design of

System on a Chip, *Robotics and Autonomous Systems*, 58(8) 2010, 1017-1027.

55. Hashemi, E., Ghaffari Jadidi, M. and Ghaffari Jadidi, N., Model-based PI–fuzzy control of four-wheeled omni-directional mobile robots, *Robotics and Autonomous Systems*, 59(11), 2011, 930-942.

56. Melin, P., Astudillo, L., Castillo, O., Valdez, F. and Garcia, M.,Optimal design of type-2 and type-1 fuzzy tracking controllers for autonomous mobile robots under perturbed torques using a new chemical optimization paradigm, *Expert Systems with Applications*, 40(8), 2013, 3185-3195.

57. Resende, C. Z., Ricardo C. and Sarcinelli-Filho, M., A nonlinear trajectory tracking controller for mobile robots with velocity limitation via fuzzy gains, *Control Engineering Practice*, 21(10), 2013, 1302-1309.

58. Kala, R., Shukla, A., & Tiwari, R., Fusion of probabilistic A* algorithm and fuzzy inference system for robotic path planning. Artificial Intelligence Review, 33(4), 2010, 307-327.

59. Han, W., Baek, S. and Kuc, T., Genetic algorithm based path planning and dynamic obstacle avoidance of mobile robots, IEEE International Conference on Systems, Man, and Cybernetics, 12-15 October 1997,Orlando, Systems, Man, and Cybernetics, *Computational Cybernetics and Simulation*, 3, 1997, 2741-2751.

60. Kubota, N., Morioka, T., Kojima, F. and Fukuda, T., Learning of mobile robots using perception-based genetic algorithm, *Measurement*, 29(3), 2001, 237-248.

61. Moreno, L., Armingol, J. M., Garrido, S., De La Escalera, A. and Salichs, M. A., A genetic algorithm for mobile robot localization using ultrasonic sensors, *Journal of Intelligent and Robotic Systems* ,34(2), 2002, 135-154.

62. Tu, J. and Yang, S.X., Genetic algorithm based path planning for a mobile robot, International Conference on Robotics and Automation,14-19 September, 2003,Taiwan, Proceeding ICRA'03, IEE Conference on,2,2003,1221-1226.

63. Keiji,K. and Ishikawa, M., Determination of the optimal values of parameters in reinforcement learning for mobile robot navigation by a genetic algorithm, *International Congress Series*, 1269, Elsevier, 2004.

64. Sedighi, K. H., Ashenayi, K., Manikas, T. W., Wainwright, R. L. and Tai, H. M., Autonomous local path planning for a mobile robot using a genetic algorithm, *Evolutionary Computation*, 2004. CEC2004. Congress on, 2. IEEE, 2004.

65. Mucientes, M., Moreno, D. L., Bugarín, A. and Barro, S., Design of a fuzzy controller in mobile robotics using genetic algorithms, *Applied Soft Computing*, 7(2), 2007, 540-546.

66. Yamada, S., Evolutionary behavior learning for action-based environment modeling by a mobile robot, *Applied Soft Computing*, 5(2), 2005, 245-257.

67. Yang, S., Fu, W., Li, D. and Wang, W., Research on application of genetic algorithm for intelligent mobile robot navigation based on dynamic approach, *Automation and Logistics*, 2007 IEEE International Conference on. IEEE, 2007.

68. Samsudin, K., Faisul A. A. and Syamsiah M., A highly interpretable fuzzy rule base using ordinal structure for obstacle avoidance of mobile robot, *Applied Soft Computing*, 11(2), 2011, 1631-1637.

69. Tuncer, A. and Mehmet Y., Dynamic path planning of mobile robots with improved genetic algorithm, *Computers & Electrical Engineering*, 38(6) 2012, 1564-1572.

70. Guan-Zheng, T. A. N., He H. and Aaron S., Ant colony system algorithm for real-time globally optimal path planning of mobile robots, *Acta Automatica Sinica*, 33(3), 2007, 279-285.

71. Tsankova, D., Georgieva, V., Zezulka, F. and Bradac, Z., Immune network control for stigmergy based foraging behaviour of autonomous mobile robots, *International Journal of Adaptive Control and Signal Processing* ,21 2007, 265-286.

72. Garcia, M. P., Montiel, O., Castillo, O., Sepúlveda, R. and Melin, P.,Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation, *Applied Soft Computing*, 9(3), 2009, 1102-1110.

73. Cong, Y. Zi. and Ponnambalam, S.G., Mobile robot path planning using ant colony optimization, *Advanced Intelligent Mechatronics*, 2009. AIM 2009. IEEE/ASME International Conference on. IEEE, 2009.

74. Luh, G.C. and Liu, W.W, An immunological approach to mobile robot reactive navigation, *Applied Soft Computing*, 8(1), 2008, 30-45.

75. Venayagamoorthy, G. K., Lisa L. G. and Sheetal D., Collective robotic search using hybrid techniques: Fuzzy logic and swarm intelligence inspired by nature, *Engineering Applications of Artificial Intelligence*, 22(3), 2009, 431-441.

76. Chia, S. H., Su, K. L., Guo Jr, H. and Chung, C. Y., Ant colony system based mobile robot path planning, *Genetic and Evolutionary Computing* (ICGEC), 2010 Fourth International Conference on. IEEE, 2010.

77. Chen, Y. L., Cheng, J., Lin, C., Wu, X., Ou, Y. and Xu, Y., Classification-based learning by particle swarm optimization for wall-following robot navigation, *Neurocomputing*, 113, 2013, 27-35.

78. Espelosín, J., Leopoldo A. and Daniel A., Path planning approach based on flock dynamics of moving particles, *Applied Soft Computing*, 13(4) 2013, 2159-2170.

79. Mo, H. and Xu., L., Research of biogeography particle swarm optimization for robot path planning, *Neurocomputing*, 148, 2015, 91-99.

80. Castillo, O., Neyoy, H., Soria, J., Melin, P. and Valdez, F. ,A new approach for dynamic fuzzy logic parameter tuning in ant colony optimization and its application in fuzzy control of a mobile robot, *Applied Soft Computing*, 28, 2015, 150-159.

81. Zelinsky, Alexander. "Mobile robot map making using sonar." *Journal of Robotic Systems*, 8(5) (1991): 557-577.

82. Varadarajan, R., Yuen, G., Bodruzzaman, M. and Malkani, M., Sensory fusion for intelligent navigation of mobile robot, *System Theory*, 1998. Proceedings of the Thirtieth Southeastern Symposium on. IEEE, 1998.

83. Masek, V., Kajitani, M., Ming, A. and Vlacic, L. V., Rapid obstacle sensing using mobile robot sonar, *Mechatronics*, 10(1), 2000, 191-213.

84. Maris, M., Attention-based navigation in mobile robots using a reconfigurable sensor, *Robotics and Autonomous Systems*, 34(1), 2001, 53-63.

85. Harper, N. and Phillip M., Recognising plants with ultrasonic sensing for mobile robot navigation, *Robotics and Autonomous Systems*, 34(2) , 2001 71-82.

86. Asharif, M. R., Behzad M. and Reza H.N., Sensor fusion by pseudo information measure: *A mobile robot application, ISA transactions*, 41(3), 2002, 283-301.

87. Benet, G., Blanes, F., Simó, J. E. and Perez, P., Using infrared sensors for distance measurement in mobile robots, *Robotics and autonomous systems*, 40(4) ,2002, 255-266.

88. Ming, A., Kajihara, K., Kajitani, M. and Shimojo, M., Development of a rapid obstacle sensing system using sonar ring for mobile robot, *Robotics and Automation,* 2002. Proceedings. ICRA'02. IEEE International Conference on. Vol. 3. IEEE, 2002.

89. Carelli, R. and Eduardo O. F., Corridor navigation and wall-following stable control for sonar-based mobile robots, *Robotics and Autonomous Systems,* 45(3), 2003, 235-247.

90. Carmena, J. M. and John CT H., The use of Doppler in Sonar-based mobile robot navigation: inspirations from biology, *Information Sciences* 161(1), 2004, 71-94.

91. Lin, W., Jia, S., Yang, F. and Takase, K., Topological navigation of mobile robot using ID tag and web camera, *Intelligent Mechatronics and Automation*, 2004. Proceedings. 2004 International Conference on. IEEE, 2004.

92. Batalin, M, Gaurav S. S. and Myron H., Mobile robot navigation using a sensor network, *Robotics and Automation*, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on. Vol. 1. IEEE, 2004.

93. Lee, K., Cho, N., Chung, W. K. and Doh, N. L., Topological navigation of mobile robot in corridor environment using sonar sensor, *Intelligent Robots and Systems*, 2006 IEEE/RSJ International Conference on. IEEE, 2006.

94. Kim, M. and Nak, Y. Chong., RFID-based mobile robot guidance to a stationary target, *Mechatronics*, 17(4), 2007, 217-229.

95. Liu, Y., Zhu, J. J., Williams, R. L. and Wu, J., Omni-directional mobile robot controller based on trajectory linearization, *Robotics and Autonomous Systems* ,56(5) ,2008, 461-479.

96. Popa, A. S., Popa, M., Silea, I. and Varlan, A., Indoor navigation of a wheeled mobile robot, *Human System Interactions,* 2008 Conference on. IEEE, 2008.

97. Ray, A.K., Behera, L. and Mo J., GPS and sonar based area mapping and navigation by mobile robots, *Industrial Informatics*, 2009. INDIN 2009. 7th IEEE International Conference on. IEEE, 2009.

98. Markovic, I. and Ivan P., Speaker localization and tracking with a microphone array on a mobile robot using von Mises distribution and particle filtering, *Robotics and Autonomous Systems*, 58(11), 2010, 1185-1196.

99. Quintia, P., Iglesias, R., Rodríguez, M. A. and Regueiro, C. V., Simultaneous learning of perception and action in mobile robots, *Robotics and autonomous systems*, 58(12) ,2010, 1306-1315.

100. Teimoori, H. and Andrey V. S., Equiangular navigation and guidance of a wheeled mobile robot based on range-only measurements, *Robotics and Autonomous Systems*, 58(2), 2010, 203-215.

101. Sanada, A., Kazuo, I. and Tetsuya Y., Self-Localization of an Omnidirectional Mobile Robot Based on an Optical Flow Sensor, *Journal of Bionic Engineering*, 7, 2010, 172-176.

102. Delgado-Mata, C., Ramiro, V. and Carlos A. G., A differential-drive mobile robot driven by an ethology inspired behavior architecture, *Procedia Technology*, 3 ,2012, 157-166.

103. Espinace, P., Kollar, T., Roy, N. and Soto, A., Indoor scene recognition by a mobile robot through adaptive object detection, *Robotics and Autonomous Systems*, 61(9), 2013, 932-947.

104. Siagian, C., Chin Kai, C. and Laurent I., Autonomous Mobile Robot Localization and Navigation Using a Hierarchical Map Representation Primarily Guided by Vision, *Journal of Field Robotics*, 31(3), 2014, 408-440.

105. Xu, W. B., Chen, X. B., Zhao, J. and Liu, X. P., Function-segment artificial moment method for sensor-based path planning of single robot in complex environments, *Information Sciences*, 280, 2014, 64-81.

106. Pozna, C., Radu-Emil, P. and Peter F., A novel pose estimation algorithm for robotic navigation, *Robotics and Autonomous Systems*, 63, 2015, 10-21.

107. Adorni, G., Stefano C., Stefan, E., Gerhard K. K., Monica M., Michael P., Marcus, R., Stefan S. and Andreas Z, Vision-based localization for mobile robots, *Robotics and Autonomous Systems*, 36(2), 2001, 103-119.

108. Barnes, N. and Zhi-Qiang, L., Embodied categorisation for vision-guided mobile robots, *Pattern recognition*, 37(2), 2004, 299-312.

109. Muñoz-Salinas, R., Eugenio, A. and Miguel G.S., Detection of doors using a genetic visual fuzzy system for mobile robots, *Autonomous Robots*, 21(2), 2006, 123-141.

110. Tsai, C.Y. and Kai-Tai, S., Dynamic visual tracking control of a mobile robot with image noise and occlusion robustness, *Image and Vision Computing*, 27(8), 2009, 1007-1022.

111. Singh, N. N., Chatterjee, A., Chatterjee, A. and Rakshit, A., A two-layered subgoal based mobile robot navigation algorithm with vision system and IR sensors, *Measurement*, 44(4), 2011, 620-641.

112. Ohnishi, N. and Atsushi, I., Appearance-based navigation and homing for autonomous mobile robot, *Image and Vision Computing*, 31(6), 2013, 511-532.

113. Wijk, O. and Henrik I. C., Localization and navigation of a mobile robot using natural point landmarks extracted from sonar data, *Robotics and Autonomous Systems*, 31(1), 2000, 31-42.

114. Marsland, S., Ulrich, N. and Tom, D., Learning to select distinctive landmarks for mobile robot navigation, *Robotics and Autonomous Systems* 37(4), 2001, 241-260.

115. Hayet, J.B., Fredrric, L. and Michel, D., A visual landmark framework for mobile robot navigation, *Image and Vision Computing* , 25(8), 2007, 1341-1351.

116. Nunez, P., Vazquez-Martin, R., Del Toro, J. C., Bandera, A. and Sandoval, F., Natural landmark extraction for mobile robot navigation based on adaptive curvature estimation, *Robotics and Autonomous Systems*, 56(3), 2008, 247-264.

117. Loevsky, I. and Ilan S., Reliable and efficient landmark-based localization for mobile robots, *Robotics and Autonomous Systems*, 58(5), 2010, 520-528.

118. Muir, P. F. and Charles P. N., Kinematic modeling of wheeled mobile robots, *Journal of robotic systems*, 4(2), 1987, 281-340.

119. Larsen, T. D., Hansen, K. L., Andersen, N. and Ravn, O., Design of Kalman filters for mobile robots; evaluation of the kinematic and odometric

approach, *Control Applications*, 1999. Proceedings of the 1999 IEEE International Conference on. IEEE, 1999.

120. Moon, J.W., Chong-Kug, P. and Fumio H., Kinematic correction of a differential drive mobile robot and a design for velocity trajectory with acceleration constraints on motor controllers, *Intelligent Robots and Systems*, 1999. IROS'99. Proceedings. 1999 IEEE/RSJ International Conference on. IEEE, 1999.

121. Li, Y. and Yugang, L., Hybrid kinematics and dynamics analysis for a mobile modular manipulator, *Electrical and Computer Engineering*, 2003. IEEE CCECE 2003. Canadian Conference on. Vol. 3. IEEE, 2003.

122. Kang, M. K., Jin, S. L. and Kyoung, L. H., Kinematic path-tracking of mobile robot using iterative learning control, *Journal of Robotic Systems*, 22(2), 2005, 111-121.

123. Chang, Y., Tan, D., Wang, H. and Ma, S., Kinematics analysis of a six-wheeled mobile robot, *Intelligent Robots and Systems*, 2006 IEEE/RSJ International Conference on. IEEE, 2006.

124. Duan, X., Huang, Q., Rahman, N., Li, J. and Du, Q., Kinematic modeling of a small mobile robot with multi-locomotion modes, *Intelligent Robots and Systems*, 2006 IEEE/RSJ International Conference on. IEEE, 2006.

125. Minguez, J., Luis, M. and Jose, S.V., Abstracting vehicle shape and kinematic constraints from obstacle avoidance methods, *Autonomous Robots*, 20(1), 2006, 43-59.

126. Lee, J. H., Kim, B. K., Tanikawa, T. and Ohba, K., Kinematic analysis on omni-directional mobile robot with double-wheel-type active casters, *Control, Automation and Systems*, 2007. ICCAS'07. International Conference on. IEEE, 2007.

127. Wang, H., Xingbin, T. and Zhen, H., Kinematic Analysis, Obstacle Avoidance and Self-localization for a Mobile Robot, *Advances in Neural Networks,* ISNN 2007,Springer Berlin Heidelberg, 2007,733-742.

128. Gao, Y., Chang, G. L. and Kil, T. C., Receding horizon tracking control for wheeled mobile robots with time-delay, *Journal of mechanical science and technology*, 22(12), 2008, 2403-2416.

129. Gracia, L. and Tornero,J., Kinematic control of wheeled mobile robots, *Latin American applied research* ,38(1), 2008, 7-16.

130. Jinxia, Y., Cai, Z. and Duan, Z., Study on kinematics modeling of mobile robot in rough terrain, *Control and Decision Conference*, 2009. CCDC'09. Chinese. IEEE, 2009.

131. Moosavian, S. Ali A., Alireza, P. and Khalil, A., Kinematics and dynamics of a hybrid serial-parallel mobile robot, *Robotics and Automation*, 2009. ICRA'09. IEEE International Conference on. IEEE, 2009.

132. Yi, J., Wang, H., Zhang, J., Song, D., Jayasuriya, S. and Liu, J., Kinematic modeling and analysis of skid-steered mobile robots with applications to low-cost inertial-measurement-unit-based motion estimation, *IEEE Transactions on Robotics*, 25(5), 2009, 1087-1097.

133. Kane, G., Boesecke, R., Raczkowsky, J. and Worn, H.,Kinematic path-following control of a mobile robot on arbitrary surface, *Robotics and Automation (*ICRA), 2010 IEEE International Conference on. IEEE, 2010, 1562 – 1567.

134. Li, Z., Ma, S., Li, B., Wang, M. and Wang, Y., Kinematics analysis of a transformable wheel-track robot with self-adaptive mobile mechanism, *Mechatronics and Automation* (ICMA), 2010 International Conference on. IEEE, 2010, 1537-1542.

135. On, S. and Ahmet Y., A comparative study of smooth path planning for a mobile robot considering kinematic constraints, *Innovations in Intelligent Systems and Applications* (INISTA), 2011 International Symposium on. IEEE, 2011, 565-569.

136. Yun, Y., Hongpeng, W., Jintai, L., Baojiong, L. and Yinhui, Y., The kinematic analysis and simulation for four-wheel independent drive mobile robot, *Control Conference* (CCC), 2011 30th Chinese. IEEE, 2011, 3958-3963.

137. Moosavian, S. Ali A., Ali, N. and Ayyub, H., Kinematics and workspace analysis of a novel 3-DOF spatial parallel robot, *Electrical Engineering (ICEE),* 2011 19th Iranian Conference on. IEEE, 2011, 1-6.

138. Shammas, E. and Daniel, A., Kinematic analysis of an active angular swivel steering mechanism for robotic mobile bases, *Robotics and Biomimetics* (ROBIO), 2012 IEEE International Conference on. IEEE, 2012, 2138-2143.

139. Maulana, E., Muhammad, A. M. and Akhmad, Z., Inverse kinematics of a two-wheeled differential drive an autonomous mobile robot, *Electrical*

Power, Electronics, Communications, Controls and Informatics Seminar (EECCIS), 2014. IEEE, 2014.

140. Passino, K. M., Biomimicry of bacterial foraging for distributed optimization and control, *Control Systems, IEEE*, 22(3), 2002, 52-67.

141. Tripathy, M. and Mishra, S., Bacteria foraging-based solution to optimize both real power loss and voltage stability limit, *IEEE Transactions on Power Systems*, 22(1), 2007,240-248.

142. Acharya, O. P., Patnaik, A. and Choudhury, B., Fault finding in antenna arrays using bacteria foraging optimization technique. In Communications (NCC), *2011 National Conference on* 1-5. IEEE, 2011.

143. Bensenouci, A., PID controllers design for a power plant using bacteria foraging algorithm. In Electronics, Communications and Photonics Conference (SIECPC), 2011 Saudi International, 1-6.

144. Powell, M. J., Radial basis functions for multivariable interpolation: a review. In Algorithms for approximation, 1987, 143-167.

145. Pham, D. T., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S. and Zaidi, M., The Bees Algorithm–A Novel Tool for Complex Optimisation. In Intelligent Production Machines and Systems-2nd I* PROMS Virtual International Conference, 2006, 454-459.

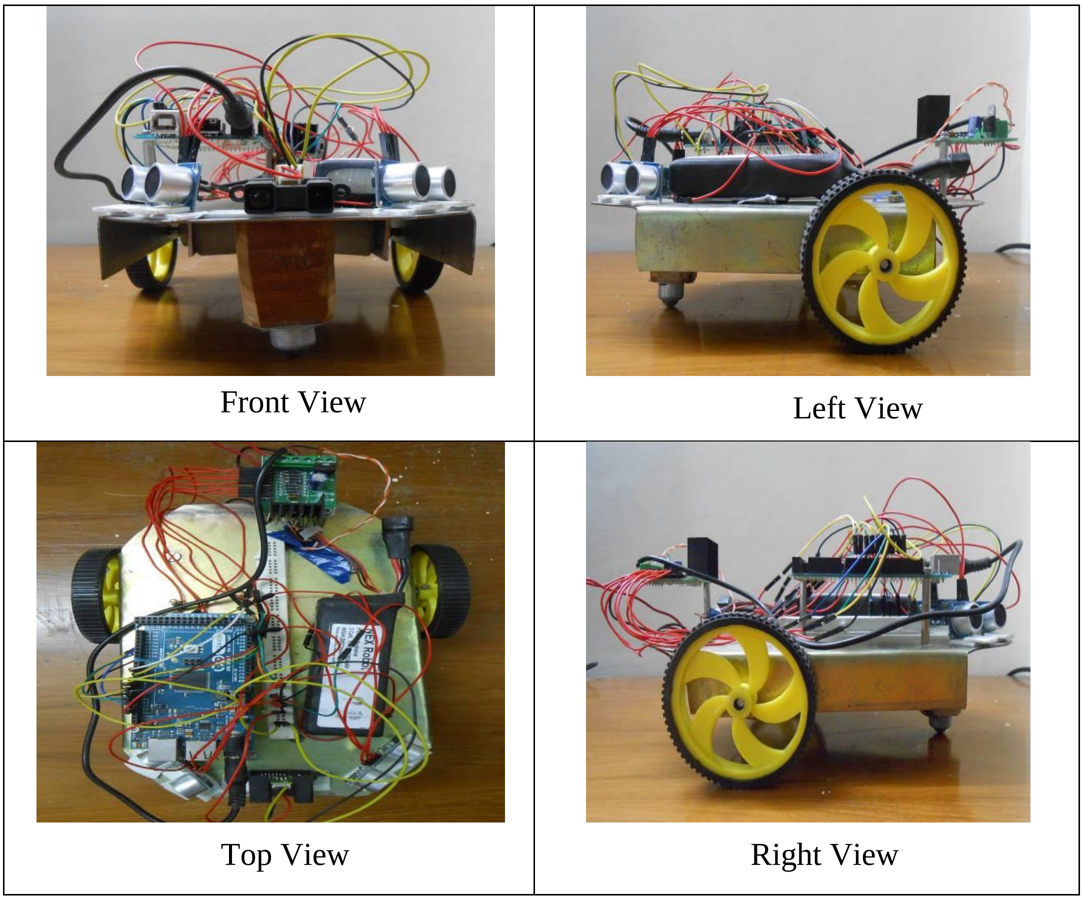
146. Karaboga, D., Basturk, B., On the performance of artificial bee colony (ABC) algorithm. *Applied soft computing*, 8(1), 2008, 687-697.

147. Tsai, C. C., Huang, H. C., & Chan, C. K., Parallel elite genetic algorithm and its application to global path planning for autonomous robot navigation. *IEEE Transactions on Industrial Electronics*, 58(10), 2011, 4813-4821.

148. Qu, H., Xing, K., & Alexander, T., An improved genetic algorithm with co-evolutionary strategy for global path planning of multiple mobile robots. *Neurocomputing*, 120, 2013 509-517.

Appendix A

Specification of the robots used in the experimental analysis.



Front View

Left View

Top View

Right View

Figure A-1 In-house differential drive robot used in the experiment.

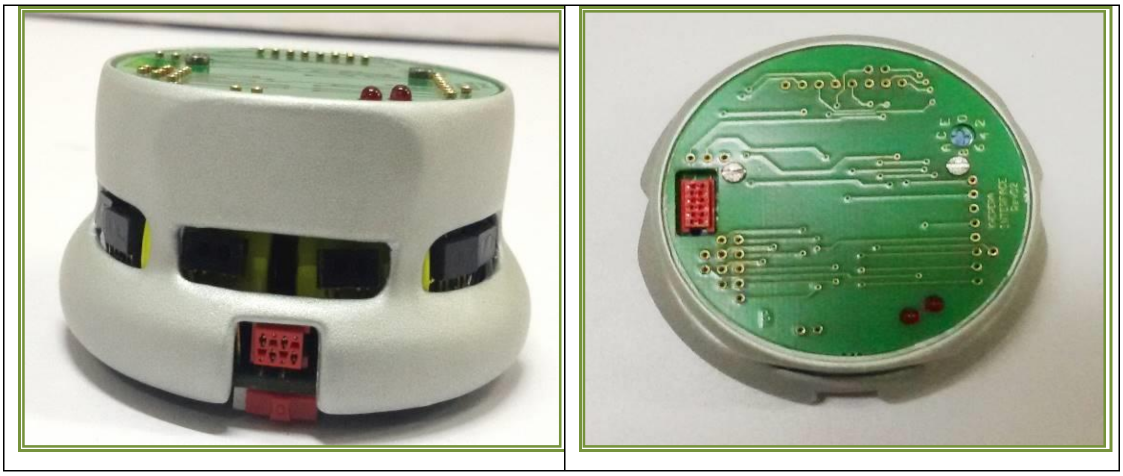


Figure A-2 Khepera II robot used in the experiment.

Table A-1 Specification of the in-house robot used in the experiment

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	Distance Sensors	a) Infrared sensors with up to 150cm range. b) 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: 25cm, Width: 19cm, Height: 12 cm
10	Weight	Approx. 1100 g
11	Payload	Approx. 4000g
12	Remote control Software via USB cable	C/C++ ® (on PC, MAC) MATLAB ® (on PC, MAC, Linux)

Table A-2 Specification of the Khepera robot used in the experiment

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 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

Different components used in the in-house robot.

1. **Ultrasonic Range Finder HC SR-04**

Specifications

Working Voltage	:	DC 5V
Working Current	:	15mA
Working Frequency	:	40Hz
Max Range	:	4m
Min Range	:	2cm
Measuring Angle	:	no more than 15 degrees
Trigger input Signal	:	10uS TTL pulse
Easy pin configuration	:	VCC, Trig, Echo, GND
Dimension	:	45*20*15mm

2. **SHARP GP2Y0A02YK0F IR Range Sensor**

Specifications

Distance measuring range	:	20 to 150cm
Output Type	:	Analog
Refresh rate	:	36ms
Supply voltage	:	4.5 to 5.5 V
Average current consumption	:	33 mA
Package size	:	29.5×13×21.6mm

3. **L298 46V, 2A Dual DC Motor Driver**

Specifications

Operating voltage	:	8V to 46V
Output current	:	2Amp per H-Bridge

Appendix B

B1. List of Papers and Patent.

Papers:

1. Jha A K, Parhi D R, "Intelligent control of mobile robots using RBFN-ABFO Hybrid Controller". (Communicated to Applied Soft Computing)
2. Jha A K, Parhi D R, "Intelligent control and path planning of multiple mobile robots using RBFN-BA-ABFO Hybrid Controller" (Communicated to Journal of Intelligent & Robotic Systems)
3. D. R. Parhi and Alok Kumar Jha, "ABFO Based Real-time navigation of mobile robots in an unknown environment", International Journal of Artificial Intelligence and Computational Research. (Accepted for publication)
4. D. R. Parhi and Alok Kumar Jha, Autonomous Mobile Robot Navigation using Bees Algorithm. "International Journal of Applied Artificial Intelligence in Engineering System (IJAAIES)" (Accepted for Publication)
5. D. R. Parhi and Alok Kumar Jha, "Path Planning of Mobile Robot using Bacteria Foraging Optimization", International Journal of Artificial Intelligence and Computational Research (IJAICR), Vol. 4, No. 1, 2012, 1-5.
6. D. R. Parhi and Alok Kumar Jha, "Review and Analysis of Different Methodologies used in Mobile Robot Navigation", International Journal of Applied Artificial Intelligence in Engineering System (IJAAIES), 4(1), 2012, 1-18.
7. BBVL Deepak Alok Kumar Jha and Dayal R Parhi, "Kinematic Model of Wheeled Mobile Robots", International Journal of Recent Trends in Engineering & Technology, 5(4), 2011, pp.5-10.
8. D. R.Parhi and Alok Jha, "Different Bio-Inspired Approaches Used In Mobile Robot Navigation - A Review", International Journal of Applied Artificial Intelligence in Engineering System (IJAAIES), 3(2), 2011, 87-93.
9. Alok Kumar Jha, Krishna Kant Pandey and Dayal R Parhi, "Fuzzy Logic based Intelligent Control of a Mobile Robot", International Conference On Advances In Mechanical Engineering, COEP, Pune, May 29-31, 2013.

Patent:

Design and Development of a Low Speed Solar Car for City Ride, (Intelligent Robot). (Application Number: 214/KOL/2013). Inventors: 1. Dr. Dayal R Parhi 2. Alok Kumar Jha (filed).

B2. Contributions of the Candidate:

1. “Intelligent control of mobile robots using RBFN-ABFO Hybrid Controller”. (Communicated to Applied Soft Computing)

The contributions are listed below:

- In this paper, a new hybrid controller has been developed by combining the features of RBFN and ABFO approaches. The hybrid controller has been designed, analysed and tested in various environments to show the effectiveness of the proposed RBFN-ABFO hybrid controller.
- The controller has two parts, the first part of hybrid controller (i.e. RBFN) gives an intermediate steering angle (ISA) as output based on the training pattern. Further, this intermediate steering angle will become the input to the ABFO controller along with the other sensory information and control parameters. Finally, the ABFO controller provides the final steering angle (FSA) to the robot for heading towards target position by following a trajectory, which is smooth and free from obstacles.
- Simulation and experimental results are presented to show the effectiveness of the proposed controller.
- The percentage of errors between the simulation and experimental results are found to be within 6 %.

The summary of contributions in this paper is as follows.

Jha A. K. has contributed in getting the simulation and experimental results, carrying out the comparison between the results (simulation and experiment), writing of the paper and contributed partially in conceiving the architecture of hybrid controller. He has also contributed greatly in developing the experimental setup using the hybrid controller. Parhi D. R. has contributed in checking the paper and looking after the layout of the paper. He has also contributed partially in designing the architecture of the hybrid controller.

2. "Intelligent control and path planning of multiple mobile robots using RBFN-BA-ABFO Hybrid Controller" (Communicated to Journal of Intelligent & Robotic Systems)

The contributions are listed below:

- In this paper, three standalone approaches (i.e. ABFO, RBFN and BA) are hybridized to develop RBFN-BA-ABFO hybrid controller.
- This work proposes a new approach for the design and development of intelligent mobile robot based on hybrid controller. It discusses the complex functional structure of such systems, providing solutions to some typical design problems.
- The system offers some solutions for the problems related to the hybrid potential field building (like dead lock and local minima problem), and on the other hand, looks for the problem-solutions connected with self-organization of the mobile robots during navigation.
- It is decentralized and behavior-oriented, because the agents sharing the basic information about the positions and orientations between each other, and on the basis of these information they define the next possible position and orientation.

The summary of contributions in the paper is as follows.

Jha A. K. has contributed in getting the simulation and experimental results, carrying out the comparison between the results (simulation and experiment), writing of the paper and contributed partially in conceiving the architecture of hybrid controller. He has also contributed to a great extent in figuring out the input output parameters at various stages so that the ABFO, RBFN and BA can be hybridized seamlessly. He has also contributed greatly in developing the experimental setup using the hybrid controller. Parhi D. R. has contributed in checking the paper and looking after the layout of the paper. He has also contributed partially in designing the architecture of the hybrid controller and development of experimental setup.

3. “ABFO Based Real-time navigation of mobile robots in an unknown environment.” (IJAICR)

The contributions are listed below:

- An intelligent controller, inspired from the bacterial foraging optimization algorithm (BFOA) has been designed, analysed and tested for a number of situations to show the effectiveness of the proposed BFOA controller.
- The step size of the bacterium in chemotaxis step has been made adaptive to speed up the convergence of swarm of the bacteria near global optima.
- ABFO algorithm allows each individual bacterium to attain a fair balance between the exploration and exploitation state.
- A new multi objective function has been defined to determine the location of the robot in the search space.
- Simulation and experimental results are presented to show the effectiveness of the proposed controller.
- The percentage of errors between the simulation and experimental results are found to be within 6 %.

The summary of contributions in this paper is as follows.

Jha A. K. has contributed in finalizing the step size of the bacterium in chemotaxis step adaptive to speed up the convergence of swarm of the bacteria near global optima and also contributed by defining the new multi objective function to determine the location of the robot in the search space. He is also responsible for getting the simulation and experimental results, carrying out the comparison between the results (simulation and experiment), writing of the paper and contributed partially in conceiving the architecture of the controller. He has also contributed greatly in developing the experimental setup using the hybrid controller. Parhi D. R. has contributed in checking the paper and looking after the layout of the paper. He has also contributed partially in designing the architecture of the controller.

4. “Autonomous Mobile Robot Navigation using Bees Algorithm.” (IJAAIES)

The contributions are listed below:

- A nature inspired algorithm (i.e. Bess Algorithm) has been used to solve the issues of path planning of mobile robots.
- A new fitness function has been defined to make the appropriate decision during the navigation.
- An additional term $\frac{K}{OD}$ is added in the fitness function to achieve the global path planning optimization. Then the equation becomes

  $$F(\mathrm{n}) = \left\| \mathrm{V}_n(\mathrm{x}, \mathrm{y}) - \mathrm{G}(\mathrm{x}, \mathrm{y}) \right\| + f_c * p_n + \frac{K}{OD}$$

  Where K is a constant. It should be noted that, the term $\frac{K}{OD}$ is added only if the distance of robot
- Moreover controller has been tested in a number of exercises in simulation and physical environment to show the effectiveness of the developed controller.
- Further, the simulation and experimental results are compared and the average percentage errors are found to be within 6 %.

The summary of contributions in this paper is as follows.

Jha A. K. has contributed in new fitness function so that the robot can make appropriate decision during the navigation. He has contributed in adding an additional term $\frac{K}{OD}$ in the fitness function to achieve the global path planning optimization. He is also responsible for getting the simulation and experimental results, carrying out the comparison between the results (simulation and experiment), writing of the paper and contributed partially in conceiving the architecture of the controller. He has also contributed greatly in developing the experimental setup using the hybrid controller and tuning of the parameters. Parhi D. R. has contributed in checking the paper and looking after the layout of the paper. He has also contributed partially in designing the architecture of the controller.

5. “Path Planning of Mobile Robot using Bacteria Foraging Optimization”, (IJAICR), Vol. 4, No. 1, 2012, 1-5.

The contributions are listed below:

- In this paper, a bio inspired approach based on the foraging behaviour of the bacteria (E-coli) has been proposed to design an optimal path for the robot.
- The present approach admires that the mobile robot is capable of reaching the goal by following the optimum path in the simulated environment.
- Simulation results have been presented for the robot motion in two different kind of environment consisting of four and six obstacles.
- The simulation results show that the robot is successfully reaching its goal by avoiding the obstacles presents along its path.

The summary of contributions in this paper is as follows.

Jha A. K. has contributed in finding out a suitable strategy for navigational control of mobile robot using Bacteria Foraging Optimization technique. He is also responsible for getting the simulation results, carrying out the comparison between the results, and writing of the paper and contributed partially in conceiving the architecture of the controller. Parhi D. R. has contributed in checking the paper and looking after the layout of the paper. He has also contributed partially in designing the architecture of the controller, and proper selection of various parameters.

6. “Review and Analysis of Different Methodologies used in Mobile Robot Navigation” (IJAAIES), 4(1), 2012, 1-18.

The contributions are listed below:

This paper provides an overview on different methodologies used for the successful navigation of mobile robot in last few decades. The paper focuses on the reactive navigation techniques based on as Neural Network, Fuzzy Logic, Neuro-Fuzzy, Fuzzy-Neuro, Genetic Algorithm, and biologically inspired techniques like Ant Colony Optimization (ACO) algorithm, Particle Swarm Optimization (PSO) Algorithm, and Bacteria Foraging Optimization Algorithm (BFOA) and Bees Algorithm (BA).

The summary of contributions in this paper is as follows.

Jha A. K. has contributed in coining of various papers as segment wise. He has also contributed in writing the critical reviews on various techniques based on as Neural Network, Fuzzy Logic, Neuro-Fuzzy, Fuzzy-Neuro, Genetic Algorithm, and biologically inspired techniques like Ant Colony Optimization (ACO) algorithm, Particle Swarm Optimization (PSO) Algorithm, and Bacteria Foraging Optimization Algorithm (BFOA) and Bees Algorithm (BA). Parhi D. R. has contributed in checking the paper and looking after the layout of the paper.

7. “Kinematic Model of Wheeled Mobile Robots” (IJRTET), 5(4), 2011, 5-10.

The contributions are listed below:

- This paper deals with the structure of the kinematic models of wheeled mobile robots (WMR).
- A wheeled mobile robot here considered as a planer rigid body that rides on an arbitrary number of wheels.
- In this paper it is shown that, for a large class of possible configurations of wheels, five types of configurations can be done namely i) fixed standard wheels, ii) steerable standard wheels, iii) castor wheels, iv) Swedish wheels, and v) spherical wheels.
- These wheels are characterized by generic structures of the model equations. Based on the geometrical constraints of these wheels a kinematical model has been proposed and the degree of mobility, steerability and maneuverability are studied.
- Finally, this analysis is applied to various wheeled mobile robots. Examples are presented to illustrate the models.

The summary of contributions in this paper is as follows.

Jha A. K. has contributed in analysing various classes and configurations of wheels such as i) fixed standard wheels, ii) steerable standard wheels, iii) castor wheels, iv) Swedish wheels, and v) spherical wheels. He has also contributed in giving the analysis applied to various wheeled mobile robots with examples. He

has also partially contributed in analysing the degree of mobility, steerability and maneuverability. He has partially contributed in writing the paper. Deepak B has contributed partially in analysing the degree of mobility, steerability and maneuverability. He has partially contributed in writing the paper. Parhi D. R. has contributed partially contributed in writing the paper, in checking the paper and looking after the layout of the paper.

8. "Different Bio-Inspired Approaches Used In Mobile Robot Navigation - A Review", (IJAAIES), 3(2), 2011, 87-93.

The contributions are listed below:

- This paper offers a brief overview on the different approaches (inspired from nature) used in mobile robot navigation.
- Selection of suitable method for any problem is depends on certain factors including the type of working environments, objectives and the constraints for the optimization problem. Therefore selection of any approach is matter of prime importance since not even a single method guaranties the optimal solution in every situation.
- The last few decades have recorded a paradigm shift towards the development of new reactive approaches inspired from nature.
- Bio-inspired approaches like particle swarm, ant colony, bacteria foraging etc. are motivated form the social and cooperative behaviour found in nature.
- Approaches based on swarm intelligence have become very popular in recent years among the researchers due to their effectiveness and ability to provide the optimal solution for several optimization problems.

The summary of contributions in this paper is as follows.

Jha A. K. has contributed in giving critical reviews on various Bio Inspired Techniques. He has also contributed in giving reviews on selection of bio inspired methodology for any problem depending upon certain factors including the type of working environments, objectives and the constraints for the optimization problem. Parhi D. R. has contributed in checking the paper and looking after the layout of the paper.

9. “Fuzzy Logic based Intelligent Control of a Mobile Robot”, International Conference on Advances in Mechanical Engineering, COEP, Pune, May 29-31, 2013.

The contributions are listed below:

- In this paper, fuzzy logic has been used to control a mobile robot in a cluttered environment.
- A mobile robot extracts information from the environment in order to perceive, map and act with the help of sensors and controller.
- An intelligent controller enables mobile robot to perform its task effectively and efficiently by execution of appropriate decision to a particular situation.
- The effectiveness of the proposed controller for navigation of robot is validated through MATLAB simulations in different environments.
- The simulation results reveal that the robot is able to avoid the obstacles and reaching towards the goal successfully by following the near optimal path.

The summary of contributions in this paper is as follows.

Jha A. K. has contributed in finding out a suitable strategy of using fuzzy logic to control a mobile robot in a cluttered environment. He has also contributed in finding a way so that the mobile robot can extracts information from the environment in order to perceive, map and act with the help of sensors and controller and also enable them to perform its task effectively and efficiently by execution of appropriate decision to a particular situation. He is also responsible for getting the simulation results, writing of the paper and contributed partially in conceiving the architecture of the controller.

Parhi D. R. has contributed in checking the paper and looking after the layout of the paper. He has also contributed partially in designing the architecture of the controller, and proper selection of various parameters.