STATUS OF THESIS

Title of thesis	HYBRID FUZZY CONTROL AND ANT COLONY OPTIMIZATION BASED PATH PLANNING FOR WHEEL MOBILE ROBOT NAVIGATION

I MOHD RAZIF BIN MOHAMAD RASHID

hereby allow my thesis to be placed at the Information Resource Center (IRC) of Universiti Teknologi PETRONAS (UTP) with the following conditions:

1. The thesis becomes the property of UTP.
2. The IRC of UTP may make copies of the thesis for academic purposes only.
3. This thesis is classified as

☐ Confidential

☑ Non-confidential

If this thesis is confidential, please state the reason:

____

The contents of the thesis will remain confidential for ___________ years.

Remarks on disclosure:

____

Endorsed by

____________________	____________________
Signature of Author	Signature of Supervisor

Permanent address:
190 Kampung Seberang Pintasan,
23050 Dungun,Terengganu
Terengganu Darul Iman

Date: ____________________ Date: ____________________

UNIVERSITI TEKNOLOGI PETRONAS

"HYBRID FUZZY CONTROL AND ANT COLONY OPTIMIZATION BASED PATH PLANNING FOR WHEEL MOBILE ROBOT NAVIGATION"

by

MOHD RAZIF BIN MOHAMAD RASHID

The undersigned certify that they have read, and recommend to The Postgraduate Studies Programme for acceptance this thesis for the fulfillment of the requirements for the degree of Master of Science in Electrical and Electronics Engineering.

Signature: ____________________

Main Supervisor: Assoc. Prof. Dr. Irraivan Elamvazuthi

Signature: ____________________

Head of Department: Dr. Rosdiazli bin Ibrahim

Date: ____________________

UNIVERSITI TEKNOLOGI PETRONAS

HYBRID FUZZY CONTROL AND ANT COLONY OPTIMIZATION BASED PATH PLANNING FOR WHEEL MOBILE ROBOT NAVIGATION

by

MOHD RAZIF BIN MOHAMAD RASHID

A Thesis

Submitted to the Postgraduate Studies Programme

as a Requirement for the Degree of

MASTER OF SCIENCE

ELECTRICAL & ELECTRONICS ENGINEERING

UNIVERSITI TEKNOLOGI PETRONAS

BANDAR SRI ISKANDAR

PERAK

MARCH 2013

DECLARATION OF THESIS

Title of thesis	HYBRID FUZZY CONTROL AND ANT COLONY OPTIMIZATION BASED PATH PLANNING FOR WHEEL MOBILE ROBOT NAVIGATION

I MOHD RAZIF BIN MOHAMAD RASHID

hereby declare that the thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at UTP or other institutions.

Witnessed by

______________________________	______________________________
Signature of Author	Signature of Supervisor
Permanent address: 190 Kampung Seberang Pintasan, 23050 Dungun,Terengganu Terengganu Darul Iman	
Date: ______________________	Date: ___________________________

Dedicated with love to my parents and family, friends and teachers, who were behind me until today

ACKNOWLEDGEMENT

First and foremost, I would like to thank Allah the Almighty for the innumerable gifts that He has granted me, for guiding me along in completing this work and for giving me an opportunity to undergo higher education.

I would like to express my sincere gratitude to my supervisor Dr Irraivan Elamvazuthi for his guidance and counseling. He has shown me a tremendous patience and care in guiding me through. Very special thanks to my co-supervisor, Dr. Mumtaj Begam for his invaluable time and guidance on this thesis throughout these years. He has shared with me a lot of insights and experiences, all of which I appreciate.

I would like to express my utmost appreciation to my dearest parent, brothers and sisters for their encouragement throughout my whole educational life. I could have not completed my degree without continuous and immeasurable support. I also would like to thank all my friends for their help and friendship.

No one book, or any worthwhile project, for that matter can be completed let alone be successful without the help of many talented individuals. Heartfelt appreciations go to UTP lecturers and staffs, who have spent their time to provide additional support and advice throughout the duration of my Masters Degree Program here in UTP

Finally I would like to dedicate this work to my parents, Mohd Rashid b Hj Mohd Zain and Khatijah bt Hj Mohd Manan, whose without them, I won’t be here today.

Mohd Razif Bin Mohamad Rashid

March 2013

ABSTRACT

Wheeled Mobile Robot (WMR) is extremely important for active target tracking control and reactive obstacle avoidance in an unstructured environment. A WMR needs the best control performance an automatic path planning to maintain a very high level of accuracy. Therefore, the development of control strategies and path planning is very significant.

Hence, research was carried out to investigate the control and path planning issues of WMR in dynamic environment. Several controllers such as conventional controller Proportional (P), Integral (I), Derivative (D) and Fuzzy Logic controller were investigated. A Hybrid Controller for differential WMR was proposed. Various aspects of the research on WMR such as kinematics model, conventional controller, fuzzy controller and hybrid controller were discussed. Overall it was found that on average the Hybrid Controller gives the best performance with 5.5s, 5.4s and 11s for target of 10x 10y, 30x10y and 60x20y respectively.

In addition, the ant colony optimization (ACO) technique is proposed to solve the mobile robot path planning (MRPP) problem. Several maps of varying complexity used by an earlier researcher is used for evaluation. Each map consists of static obstacles in different arrangements, with a starting point and destination points. The ants (representing the mobile robot) are placed at the starting point. The ants find their shortest distance possible towards the destination whilst avoiding any obstacle. The results showed that the path length decreased as the obstacles were increased and the CPU time decreased as well when the obstacles were increased. This can be attributed to the fact that the path planning method based on ACO can find a near optimal path and avoid obstacles timely in different environments. The reason for this is that ACO exploit the characteristic of solution space in path planning.

Overall, the results demonstrate the effectiveness of the proposed approach for controller and path planning.

ABSTRAK

Kajian ini adalah mengenai Robot beroda mudah alih (WMR) yang telah dilaksanakan bagi mengesan kawalan sasaran yang aktif sekaligus mengelak halangan yang wujud di dalam situasi tertentu. WMR memerlukan kawalan prestasi rancangan laluan automatik yang terbaik untuk mengekalkan tahap ketepatan kedudukannya.Bagi mencapai matlamat ini, beberapa retak kawalan dan laluan telah dicadangkan bagi memperolehi kawalan yang lebih strategik dan bermutu. .

Oleh itu, satu kajian telah dilakukan untuk menyisat kawalan dan perancangan laluan WMR dalam persikitaran yang tidak menentu.Dalam kajian ini, satu pendedahan pengawalan dalam bentuk campuran telah dicadangkan dan ianya memberikan prestasi yang terbaik menggantikan sistem pengawalan yang sedia ada seperti kawalan konvensional dan kawalan Fuzzy Logic.

Bagi mencapai objektif yang seterusnya, satu kaedah diperkenalkan iaitu pengoptimuman koloni semut (ACO) bagi menyelesaikan masalah berkaitan dengan perancangan laluan untuk pergerakan Robot beroda. Terdapat beberapa laluan telah diuji yang berlainan tahap kepayahannya dimana pelbagai susunan halangan yang statik telah dibuat berdasarkan penyelidikan sebelum ini dengan mengambil kira titik permulaan hingga ke titik akhir. Semut yang mewakili robot mudah alih ini telah ditempatkan pada titik permulaan. Ianya akan mencari laluan yang paling singkat dan dalam masa yang sama mengelak halangan untuk sampai ke destinasi. Dalam kajian ini, kaedah ACO adalah yang terbaik dalam menentukan laluan yang paling optimum dan menjauhi halangan yang ada di dalam situasi yang berlainan. Berdasarkan keputusan yang diperolehi, laluan perjalanan dan masa pemprosesan computer akan semakin berkurangan jika halangan semakin bertambah.

Secara keseluruhannya, kajian ini membuktikan bahawa penggunaan kawalan campuran sebagai pengawal adalah yang terbaik dalam mengawal Robot beroda dan kaedah pengoptimuman koloni semut (ACO) adalah satu keadah yang efektif yang mempunyai ciri-ciri penyelesaian dalam mengawal perancangan laluan.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

WMR	Wheel Mobile Robot
MR	Mobile Robot
WMRs	Wheel Mobile Robots
ACO	Ant Colony Optimization
F3M	Fuzzy Three Membersip Function
F5M	Fuzzy Five Membersip Function
F7M	Fuzzy Seven Membersip Function
FPID	Fuzzy and Proportional Integral Derivative
FPI	Fuzzy and Proportional Integral
FP	Fuzzy and Proportional
GA	Genetic Algorithms
Vsn	Very small negative
Sn	Small negative
Ne	Negative
Ze	Zero
Po	Positive
Bp	Big Positive
Vbp	Very big positive
Vbfr	Very big far
Vfr	Very far
Fr	Far
Mi	Medium
Nr	Near
Vnz	Very near zero
Zo	Zero
Vf2	Very fast 2
Vf1	Very fast 1
Ft	Fast

Me	Medium
Sw	Slow
Vs1	Very slow 1
Vs2	Very slow 2

NOMENCLATURE

$V_R(t)$	linear velocity of right wheel
$V_L(t)$	linear velocity of left wheel
$\omega_R(t)$	angular velocity of right wheel
$\omega_L(t)$	angular velocity of left wheel
r	nominal radius of each wheel
L	distance between the two wheels
R	instantaneous curvature radius of the robot trajectory, relative to the mid-point of the wheel axis
ICC	Instantaneous Center of Curvature
$X(t)$	X-position respect to time
$Y(t)$	Y-position respect to time
$\theta(t)$	Angle position respect to time
V_a	armature voltage (V)
V_f	Field voltage (v)
R_a	armature resistance (Ω)
L_a.	armature inductance (H)
i_a	armature current (A)
i_f	Field current (A)
V_b	back emf (V)
ω	angular speed (rad/s)
T_m	motor torque (N m)
θ	angular position of rotor shaft (rad)
J_m	rotor inertia (kgm^2)
B_m	viscous friction coefficient (Nm s/rad)
K_m	motor torque constant (Nm/A)
K_b	back emf constant (V s/rad)

CHAPTER 1

INTRODUCTION

This chapter begins by proving a background of wheeled mobile robot, motivation, research objectives, contribution of the thesis and finally, the outline of the thesis.

1.1 Introduction

The term 'Robot' has its root in Czech word 'Robota' which means "servitude" or "forced labour" [1]. Historically, robots can be classified into two categories; i.e., robot manipulators and mobile robots. A definition of a mobile robot can be derived from that of the classical robots as "A robot vehicle capable of self propulsion and (re)programmed locomotion under automatic control in order to perform a certain task"[2]. Nowadays, robots have major influence on human lives where they are widely used in manufacturing industry, security and consumer market. The service robot market is expected to exceed the market for industrial robots within the next few years as shown in Figure 1.1[3-4]

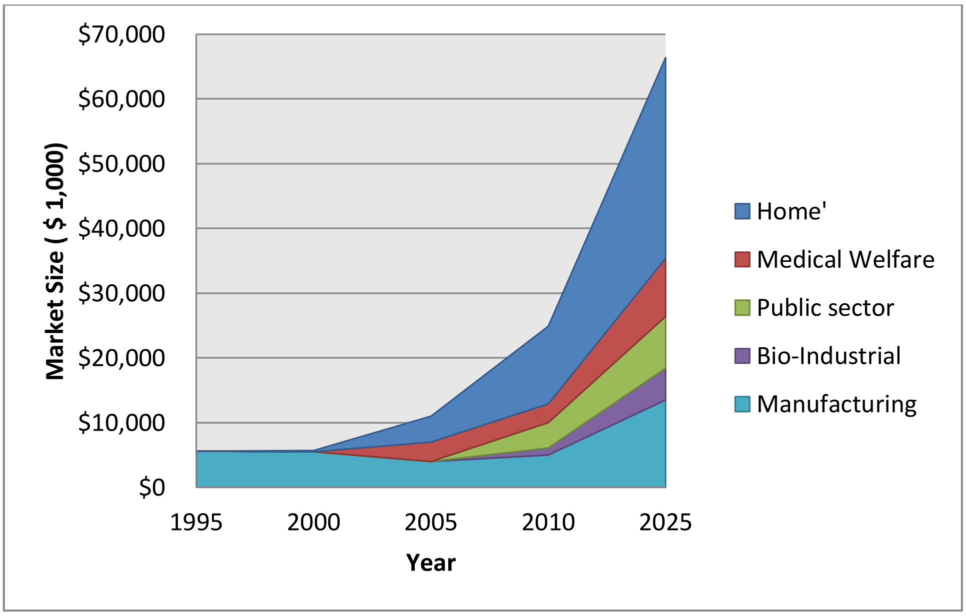

Figure 1.1: Predicted Growth in Robotics Markets [3-4]

In the last three decades, mobile robots have become a subject of significant interest due to broad range of possible applications. The research devoted to mobile robots started in three different domains; the first was the need for more flexible automated guided vehicles in manufacturing plants, the second domain to stimulate research in mobile robots was that of planetary rovers and the third domain was the research into 'artificial intelligence' where mobile robots were found to be an appealing test bed for various artificial intelligence techniques [3].

Mobile robots can be divided into land-based, air-based and water-based. The land-based robot in turn can be categorized as wheeled, tracked or legged. Land-based Wheeled Mobile Robots (WMRs) are increasingly present in industrial and service robotics and are considered to be the most popular amongst the researchers. WMRs is classified into five generic types through WMR mobility degree m (number of WMR velocities that can be assigned instantaneously and independently) and WMR steeribility degree s (number of orient able wheels that are steered independently) [5]. The WMR of type 1 are constructed with no fixed or orientable wheels and are called omnidirectional. The WMR of type 2 have one independent fixed wheel and other possibly omnidirectional wheels. An example of this type is the differential-drive WMR. The WMR of type 3 have one independent orientable wheel and other possibly omnidirectional wheels. An example is the syncro-drive WMR. The WMR of type 4 have one independent orient able wheel and another independent fixed wheel. Examples of this type are the tricycle WMR, the bicycle WMR, and the car-like WMR. The WMR of type 5 are characterized by two independent orientable wheels. Note that, the mobility degree represents the number of degrees of freedom that can be instantaneously used, without reorientation of the orientable wheels. Thus it is also defined maneuverability degree g as the sum of the mobility and steeribility degrees, and represents the total degrees of freedom, with reorientation of the orientables wheels. Therefore the WMR of types 1, 3, and 5 have full maneuverability (g = 3) and the WMR of types 2 and 4 have restricted maneuverability (g = 2) [5].

Beyond the relevance in applications, the problem of navigation of WMRs has attracted the interest of researchers in view of its theoretical challenges. The basic problem of a WMR is that of navigation: moving from one place to another by a coordination of path planning and control. In any navigation scheme the desire is to reach a destination without getting lost or crashing into anything. Put simply the navigation problem is to find a path from start (S) to target (T) and traverse it without collision. The planned path is usually decomposed into line segments between ordered sub-goals or way points. In the navigation phase, the robot follows those line segments towards the target. Path planning can be classified as either local or global [5]. Global path planning takes into account all the information in the environment when ending the optimum path between the starting and target locations. Local planning algorithms are designed to avoid obstacles within a close vicinity of the robot. Therefore, only information about the nearby obstacles is used. For more efficient and effective WMR control, it is important to have optimal or near-optimal parameters for the controller [6]. This thesis mainly studies the control and path planning issues of WMR in dynamic environments.

1.2 Motivation of the Research

A WMR needs the best control performance to maintain a very high level of accuracy. The velocities of the robot's wheels need to be controlled to a level where the speeds demanded by the motion planner would be the actual speeds of the drive wheels. The motor controller is critical to the WMR's performance. To generate a smooth path for the WMR navigation, controlling of the two wheel accelerations of the mobile robot is important and good choice. An effective controller will result in a smooth mobile robot motion without suffering speed jumps. The developed controller should be able to generate a smooth path for the mobile robot to avoid both static and moving obstacles, and to reach the target. Therefore, the literature review has shown that although several controllers have been developed for WMR, but they are unable to produce the required performance as outlined above. Thererfore, it is essential that an effective controller is developed for WMR in a dynamic environment.

Although several methods have been proposed for global planning such as voronoi planning, cell decomposition and randomized planning, however, the global approach is very time consuming in the pre-computation step. The potential field-based method for local path planning is the most known method in which the configuration space is divided into a one regular grid and then the optimum collision-free path is searched for. Different potentials are assigned to the cells of the grid. The attractive potentials are given to the cells that are close to the mobile robot's goal, while the repulsive potentials are assigned to the obstacles. The planned path is constructed along the most promising direction. Although the methods are fast, they can be trapped in local minima of the potential function [6]. To avoid local minima and at the same time to get the best optimal path, an automatic planning approach is needed.

1.3 Research Objectives

In view of the foregoing problems, the main objective of this research is to investigate the control and path planning issues of WMR in dynamic environment. Specifically, the objectives of this research are as the following:

1. to develop an effective controller algorithm of WMR in a dynamic environment
2. to develop an optimization algorithm to optimize the path planning of WMR with obstacle avoidance

1.4 Contribution of Thesis

The main contribution of this thesis is the development of a novel hybrid Fuzzy-PID controller and Ant Colony Optimization (ACO) Algorithm based path planning for WMR in dynamic environments. Eventhough hybrid Fuzzy-PID algorithm has been implemented in other engineering applications, however its application in WMR are still new. In terms of path planning of WMR, the use of Ant Colony Optimization (ACO) Algorithm is considered new as well.

1.5 Organization of the Thesis

The rest of the thesis is organized as the following:

Chapter 2 provides the background and literature review of the relevant topics. In this chapter, several aspects of mobile robots, such as history of robotics, functionalities of the mobile robot in terms of drive system, controller of mobile robots, path planning of mobile robots are briefly reviewed. Finally, critical analyses of the literature are provided.

Chapter 3 presents the modeling of mobile robot in terms of kinematic model of mobile robot and dc motor model. This is followed by discussion on simulation work using the controllers such as conventional controller, fuzzy logic controller and the proposed hybrid controller. Finally, it discusses the optimization of path planning mobile robots in terms of the principles of ant colony optimization technique and the ant colony optimization algorithm.

In Chapter 4, comprehensive results and discussion is provided for conventional controller, fuzzy logic controller, the proposed hybrid controller and path planning using the ant colony optimization algorithm. Simulation results demonstrate the effectiveness of the proposed approach.

Chapter 5 concludes the thesis with a general discussion and ideas for future work.

CHAPTER 2

LITERATURE REVIEW

In this chapter an overview of autonomous mobile robots, functionalities of the mobile robot in terms of drive system, controller of mobile robots, path planning of mobile robot and finally critical analyses of the literature are presented.

2.1 Overview of Autonomous Mobile Robots

A robot can be defined as ‘a mechanical device which performs automated tasks, either according to direct human supervision, a pre-defined program or, a set of general guidelines, using artificial intelligence techniques’[7-8]. The first commercial robot was developed in 1961 and used in the automotive industry by Ford. The robots were principally intended to replace humans in monotonous, heavy and hazardous processes. Nowadays, stimulated by economic reasons, industrial robots are intensively used in a very wide variety of applications. Most of the industrial robots are stationary. They operate from a fixed position and have limited operating range. The surrounding area of the robot is usually designed in function of the task of the robot and then secured from external influences. These robots efficiently complete tasks such as welding, drilling, assembling, painting and packaging.

However, in many applications it can be useful to build a robot which can operate with larger mobility. In contrast to most stationary robots, where the surrounding space is adapted to suit the robot tasks, mobile robots have to adapt their behavior to their surroundings. Instead of performing a fixed sequence of actions, mobile robots need to develop some awareness of their environment through interaction with all kind of sensors; they use on-board intelligence to determine the best action to take. The development of intelligent navigation systems on mobile robots, which ensures efficient and collision free movement, is still the centre of several research projects [7].

Mobile robots are generally those robots which can move from place to place across the ground. Mobility gives a robot a much greater flexibility to perform new, complex, exciting tasks [9]. A mobile robot needs locomotion mechanisms that enable it to move unbounded throughout its environment. There is a large variety of possible ways to move which makes the selection of a robot's approach to locomotion an important aspect of mobile robot design. Most of these locomotion mechanisms have been inspired by their biological counterparts which are adapted to different environments and purposes [10-11].

In mobile robotics the terms omnidirectional, holonomic and nonholonomic are often used, a discussion of their use will be helpful [10]. The terms 'holonomic' and 'omnidirectional' are sometimes used redundantly, often to the confusion of both. Omnidirectional is a poorly defined term which simply means the ability to move in any direction. Because of the planar nature of mobile robots, the operational space they occupy contains only three dimensions which are most commonly thought of as the x, y global position of a point on the robot and the global orientation, θ, of the robot. Whether a robot is omnidirectional is not generally agreed upon whether this is a two-dimensional direction, x, y or a three-dimensional direction, x, y, θ.

Nonholonomic robots are most prevalent because of their simple design and ease of control. By their nature, nonholonomic mobile robots have fewer degrees of freedom than holonomic mobile robots. These few actuated degrees of freedom in nonholonomic mobile robots are often independently controllable or mechanically decoupled, further simplifying the low-level control of the robot. Since they have fewer degrees of freedom, there are certain motions they cannot perform. This creates difficult problems for motion planning and implementation of reactive behaviours. On the other hand, holonomic offers full mobility with the same number of degrees of freedom as the environment. This makes path planning easier because there aren't constraints that need to be integrated.

Mobile robots pose a unique challenge to AI researchers. They are inherently autonomous and they force the researchers to deal with key issues such as uncertainty, reliability, and real-time response. Mobile robots also require the integration of sensing, signal processing and actions [12]. The milestones in the evolution of robotic systems are summarized in Table 2.1.

Table 2.1: Milestones in the evolution of robotic systems [5]

Year	Name	Features
1953	Elsie and Elmer	The first robots; robotic tortoises
1961	Unimate	The first installed industrial robot
1968	Shakey	Featured with a camera and bumpers
1975	PUMA	Programmable universal manipulation arm by Unimation
1977	Stanford Cart	Used stereo vision in order to navigate
1984	Wasebot	A piano-playing humanoid robot by Waseda University in Japan
1990	Navlab 5	Autonomous highway navigation at high speed
1997	Deep Blue	A chess-playing computer developed by IBM defeated world champion Garry Kasparov
1997	Sojourner	An autonomous robot operated on Mars
1998	ASIMO	A range of sophisticated humanoids by HONDA
1999	AIBO	An entertainment robot dog by SONY
2000	Trilobite	First robotic vacuum cleaner by Electrolux
2002	Roomba	A domestic autonomous mobile robot for floor cleaning by iRobot
2004	Robosapien	A biomorphic toy robot
2006	ASIMO	An autonomous robot that can learn to run and climb stairs, even serve the customer coffee, by HONDA
2008	Mind Control Project	A monkey in USA operates a robotic arm in Japan with mind control
2009	Adam	An intelligent agent that can carry out its own experiments, produce hypotheses, make scientific discoveries
2010	Robonaut 2	The latest generation of the astronaut helpers, launched to the space station aboard space shuttle Discovery on the STS-133 mission. It is the first humanoid robot in space, and although its primary job for now is teaching engineers how dexterous robots behave in space, the hope is that through upgrades and advancements, it could one day venture outside the station to help spacewalkers make repairs or additions to the station or perform scientific work.[6]
2011	Geminoid DK	Is an incredibly realistic android, nearly indistinguishable from a real human. •Geminoid-DK is build and designed by Kokoro Inc.,Tokyo.The design process was supervised by Henrik Scharfe [13]

2.2 Functionalities of a Mobile Robot

A mobile robot includes many hardware parts: sensors, actuators, power supplies, computing devices, signal processors, communication devices, etc. An autonomous mobile robot takes action based on its perception and reasoning. It is capable of perceiving its environment through its sensors, processing the information using its on-board processor(s), and responding through actuators. Artificial Intelligence (AI) plays a major role in performing the “Intelligent” behavior for a robot. It is desirable that robots are more intelligent and efficient. Meanwhile, robots have become more and more complex, equipped with various sensors and other devices to sense the environment. How to make use of the sensor information is a key issue in robotics.

After a mobile robot is built, how to effectively control such a complex system to perform desired actions is a challenging problem. Modelling of the mobile robot is required for design of a robot controller. It is not easy to create a model for a complex system with all the details and all the complexities. Development of truly autonomous systems is dependent on the creation of sophisticated models, which are able to address the different disturbances and unexpected situations. Due to the limitations of hardware and current techniques, it is almost impossible to develop a completely autonomous mobile robot to deal with complex tasks in unknown environments.

Once the robot model is established, a control method based on the model should be designed for the robot to complete the desired tasks. Normally, the control algorithm is implemented by a software development platform. The behaviour of a mobile robot results from three fundamental components [14].

- The program running on the robot (the software)
- The physical actions of the robot (the way its sensors and motors work, battery charge, etc.)
- The environment (the workspace of the robot)

The control methods for a mobile robot are based on the analysis and understanding of the relationship between these three fundamental components to generate reasonable behaviors.The mobile robot behavior emerges from the interaction between the mobile robot and its surroundings. This is illustrated in Figure 2.1.

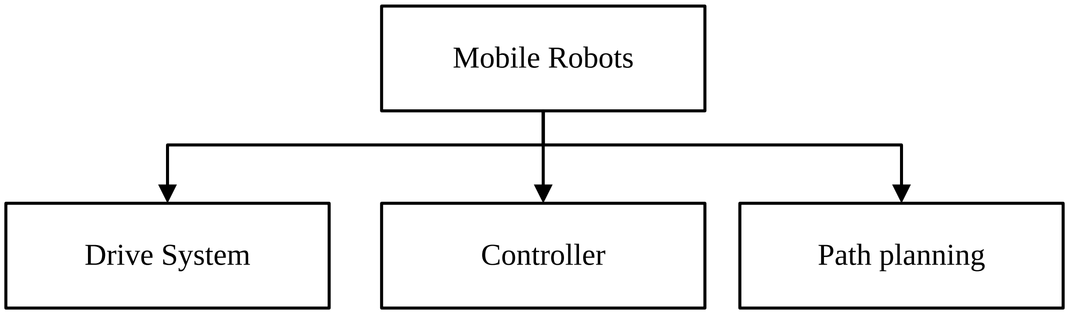

Figure 2.1:Mobile Robots system

The controller gathers and analyzes the sensory information, and produces commands to actuators for certain behaviors predefined in the control algorithm. This process repeats until a predefined goal is achieved. Thus, the robot completes the task.

2.2.1 Drive System of Wheel Mobile Robots (WMR)

Five specific drive systems are discussed in order to demonstrate concrete applications of the concepts discussed above to wheel mobile robots built for real-world activities [8][15][17]. Table 2.2 shows drive system of wheel mobile robots (WMR).

Table 2.2**:**The drive system of wheeled mobile robots [17-21]

		Differential Drive	Tricycle	Ackermann	Omini-direction	Synchronous Drive
1.	**Description**	✓ Two driving wheels (plus roller-ball for balance)	✓ Employs a single driven front wheel with two passive rearwheels.	✓ Used in automotive ✓ Designed to ensure the inside front wheel is rotated slightly sharper angle than the outside wheel when turning	✓ The dead reckoning solution for omni-directional drive is similar tothat of differential drive system. ✓ Both use shaft encoders data bits ofinformation to obtain position and velocity.	✓ Each wheel is capable of being driven and steered ✓ Three steered wheels arranged as vertices of an equilateral
2.	**Application**	✓ Indoor	✓ Outdoor and Indoor	✓ Outdoor	✓ Indoor	✓ Indoor
3.	**Advantages**	✓ simplest drive mechanism ✓ Cheap to build ✓ Easy to implement	✓ In terms of odometry, the dead-reckoning solution is similar tothat of Ackerman Steering.	✓ Simple to implement ✓ Simple 4 bar linkage controls front wheels	✓ Exceptional maneuverability. ✓ Allows complicated motions:	✓ Ability to control the orientation θ of their pose directly. ✓ Straight-line motion is guaranteed mechanically
4.	**Disadvantages**	✓ Sensitive to the relative velocity of the two wheels (small error result in different trajectories, not just speed) ✓ Difficult straight line motion	✓ When traversing up an incline, center of gravity tends to move away from the drive wheel causing traction loss. ✓ cannot turn ±90º	✓ This eliminates geometrically induced tire slippage ✓ Nonholonomic planning requiredPictures	✓ Increased wheel slippage tire wear ✓ inefficiency	✓ Complex design and implementation

2.2.2 Controller of Mobile Robots

Research and applications of the wheel mobile robots (WMR) is growing every day with many different controllers being used [9][22-27]. Many areas on mobile robots have drawn much attention [28-34]. The summary of the research carried out thus far on the controllers is outlined in Table 2.3.

Table 2.3 : Controller of mobile robots

REF	AUTHOR	TECHNIQUES	DESCRIPTION
[35]	V.M.Peri,2005	FL and P controller	The performance of P was worse than FLC because P reaches steady state later than FLC
[36]	Das T, 2006	Fuzzy Logic	In the proposed controller, only position measurements were studied
[22]	G. Narvydas 2007	Fuzzy logic and Neural network	It was found Control system with Fuzzy Logic is better because it change speed of the wheels continuously while in the State-Based control system each state has its own speed of the wheels and the jumps from one state to another cause the jumps of the wheels speed
[63]	N. Giap,,2008	Adaptive robust fuzzy controller	The proposed control scheme does not require the accurate parameter values for the actuator parameters as well as the robot parameters. The validity and robustness of the proposed control scheme are demonstrated through computer simulations
[37]	N. Liu,2009	Fuzzy Logic	As the robot's direction error and orientation error relative to oriented goal are hardly coupled, so a position-orientation alternate control method is proposed. When position error is bigger, a position controller is used to achieve the fast track, or an orientation controller is used. But during the orientation control, if orientation error is smaller, it turns back to position control. These two kinds of controllers take in turns, by which the mobile robot can be controlled to achieve path following mission effectively
[28]	D.Chwa et ,2010	Back stepping-like feedback Controller	A back stepping-like feedback control structure is proposed in the form of a cascaded kinematic and dynamic linearization to have a simpler and modular control structure. Stability analysis shows that the tracking errors of the posture (the position and heading direction angle) are globally ultimately bounded and its ultimate bound can be adjusted by the proper choice of control parameters.
[38]	H. Huang, 2011	Genetic algorithm and ant colony optimization	GA has been combined with ACO in evolving new solutions by applying crossover and mutation operators on solutions constructed by ants. These optimal parameters are used in the GA-ACO kinematic controller to obtain better performance for four-wheeled omni directional MR to achieve both trajectory tracking and stabilization.

2.2.3 Path Planning of Mobile Robot

A path planning problem is one of the important issues of the mobile robotics research. The path planning problem is to find a optimal path for which there is no collision with obstacles when the mobile robot moves from a starting point to a goal point under the given environmental information. The discussion in this thesis focuses on the global path planning problem of the planning issues. In case of this problem, the environment model is precisely defined and the planning is performed based on the information given previously. The path planning problem is one of the optimization problems. There are many algorithm to solve these, such as Genetic Algorithm (GA), Ant Colony Optimization (ACO) algorithm, Particle Swarm Optimization (PSO) algorithm and so on. However, these algorithms face a common problem which is a trade-off between the exploitation and the exploration to search the solution [39].

The path planning problem is further divided into two categories, global path planning and local path planning [40]. In global path planning, it is required that environment should be completely known and the obstacles should be static. The mobile robot generates a complete path from the starting point to the destination before it starts moving. On the other hand, local path planning enables a mobile robot to plan its path as it is moving in the environment. In other words, this means that the mobile robot will be able to produce new paths in response to any changes in the environment [41].

The summary of the research carried out thus far for path planning of mobile robot has been outlined in Table 2.4.

Table 2.4: Path planning of mobile robot

NO	AUTHOR	TECHNIQUES	DESCRIPTION
[42]	Sedighi, 2004	GA	Research was on 8 maps, each having a grid representation and an equal number of rows and columns. Besides that, each map has static obstacles and walls in different arrangements. It was based on the concept of local path planning.
[43]	Castillo and Trujillo,2005	GA	The performance of the conventional GA (one criterion; minimum path length) was compared with that of the MOGA (two criteria for a holonomic for a holonomic on a 2-dimensional grid for minimum path length and difficulty).The simulation result he was found the conventional GA and of the MOGA that show that both types of GA are effective tools for solving the point-to-point path planning problem
[44]	Lei et. Al,2006	PSO	The modified PSO algorithm needs to optimize the constrained path functions to obtain a satisfactory collision-free path between the start and the finish points.Simulation results show the fleasibility and effectiveness of the propose approach.
[45]	Li et. Al 2006	Hybrid GA	A hybrid GA to optimize the paths that have been planned by a mobile robot. The crossover and mutation operations in his GA are performed based on the self-adapting algorithm instead of the adjustment algorithm.The simulation results shows, hybrid GA has provided faster search speed compared GA.
[17]	B. A.Garro ,2007	ACO	Propose a variant of the ACO applied to optimize the path that a robot can follow to reach its target destination.
[46]	J. Lee,2008	ACO	An improved ACO algorithm was proposed to find a collision-free and optimal path from a start point to a goal point in environment of known obstacles. A ranking selection method for pheromone update was used. It was found the proposed algorithm can be applied to bigger and more complex map compared to conventional ACO algorithm has applied to only a simple and small size map.
[47]	M.A Porta,2009	SACO-MH	A Simple ACO Meta Heuristic (SACO-MH) was proposed for robot navigation find the execution time was decreased drastically.
[48]	S.H. Chia et al,2010	ACO	ACO for obstacle avoidance using grid platform, the robot has eight possible directions: north, south, east, west, north-east, north-west, southeast or south-west. The experiment was performed in a rectangle of 20X20 grids. Each node moved to was stored in the memory and hence visited once only. Obstacles of various shapes were used and optimum results were obtained in each case.
[66]	M.Brand,2010	ACO	The ACO algorithm is applied to find the shortest and collisionfree route in a grid network for robot path planningThe simulation results demonstrate that the ACO algorithm can successfully re-route the optimal path for the new network after obstacles are added.
[39]	J. Lee et al,2011	Heterogeneous ACO	HACO algorithm to solve the global path planning problem for autonomous mobile robot was proposed. These algorithms face a common problem which is a trade-off between the exploitation and the exploration to search the solution.

2.3 Discussion

A several drive systems such as differential drive, tricycle, Ackermann, omni-direction and synchronous drive was discussed in the preceding section. For the research that was carried out, the differential drive was chosen because of the advantages of it such as it can be used indoor, has simplest drive mechanism, cheap to build and easy to implement. With this drive system, it would be relatively easier to implement different controllers.

As for controllers, several controllers such as Fuzzy Logic, Proportional, Neural network, Adaptive Robust Fuzzy Controller, Back stepping-like feedback Controller, Genetic algorithm and Ant Colony Optimization have been implemented. It was found that the FLC controller was able to perform much better than the P controller, as it is seen that the robot reaches a steady state much earlier with a FLC than a P controller. In the adaptive fuzzy logic based controller, it was found that the stability and error boundness was easier to be tackled using Lyapunov stability theory. As for Fuzzy Basis Function Network (FBFN), it does not require the accurate parameter values for the actuator parameters as well as the robot parameters and the validity and robustness of the proposed control scheme were demonstrated effectively. GA has been combined with ACO in evolving new solutions by applying crossover and mutation operators on solutions constructed by ants. These optimal parameters are used in the GA-ACO kinematic controller to obtain better performance for four-wheeled omnidirectional mobile robots to achieve both trajectory tracking and stabilization. Overall, it was found that Fuzzy Logic Controller was not explored extensively for wheel mobile robot using differential drive.

As for path planning, several algorithms such as Genetic Algorithm, particle swarm optimization, Hybrid Genetic Algorithm and Ant Colony Optimization with different variants were used. GA was used in the local path planning. Hybrid GA algorithm was found be able to provide better performance compared to GA only. PSO algorithm was able to optimize the constrained path functions to obtain a satisfactory collision-free path between the start and the finish points. Recently, Ant Colony Optimization algorithm has been explored for path planning with some promising results.

Based on the literature review and discussed earlier, it was found that differential drive with Fuzzy Logic approach has scope for further investigation. It does not require heavy computation as some Neural Networks and they are able to handle imprecision and uncertainties that are often present in many real-world problems. Therefore, the Fuzzy Logic approach is adopted for developing the controller in this research. In addition, Ant Colony Optimization is easy to use and have the capability of searching for optimal or near-optimal solution in the search space. Although, Neural Networks is used for optimization through learning, but their "black box" nature does not give explicit knowledge of the model. Therefore, Ant Colony Optimization was adopted as optimization technique for the path planning in this research because it is able to provide explicit information and with less computational time.

2.4 Summary

In this chapter, an overview of wheel mobile robots was presented with detail discussions on some important aspects of wheel mobile robots such as drive system, controller and path planning. A reasonable amount of latest literature was discussed and it revealed that there is wide scope for research with Fuzzy Logic as controller and Ant Colony Optimization algorithm for path planning. In the next chapter, how these methods were used in the simulation work is discussed.

CHAPTER 3

MODELING AND SIMULATION

This chapter discusses the modeling of wheeled mobile robot (WMR) in terms of kinematic model of mobile robot and dc motor model. This is followed by discussion on simulation work using controllers such as conventional controller, fuzzy logic controller and hybrid controller. Finally, it discusses the optimization of path planning mobile robots in terms of the principles of ant colony optimization technique and the ant colony optimization algorithm.

3.1 Modeling of Mobile Robot

This section discusses the modeling of mobile robot and the DC motor that was used in the research. The Kinematic model of mobile robot in section 3.1.1 and The DC Motor model in section 3.1.2.

3.1.1 Kinematic Model of Mobile Robot

Kinematics is the most basic study of how mechanical systems behave. In a differential drive WMR, the mechanical behavior need to be studied to enable better controller design [23]. In this research, in order to analyze the system and develop controllers, a differential drive is considered for the kinematics models of autonomous mobile robot. The governing equations of the kinematic model taking nonholonomic constraints are well-known (see [35], [49-51] for detail derivation). Figure 3.1 shows the kinematic model of the WMR [23, 52].

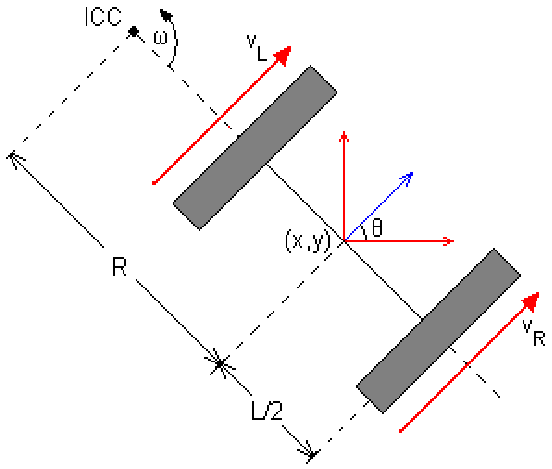

Figure 3.1: Kinematic model of the WMR [23, 52]

The the variables and equations based on the kinematic model is discussed in the following:

$V_R(t)$= linear velocity of right wheel

$V_L(t)$= linear velocity of left wheel

$\omega_R(t)$= angular velocity of right wheel

$\omega_L(t)$= angular velocity of left wheel

r = nominal radius of each wheel

L= distance between the two wheels

R= instantaneous curvature radius of the mobile robot trajectory, relative to the mid-point of the wheel axis

ICC= Instantaneous Center of Curvature

$R-(L/2)$= Curvature radius of trajectory described by left wheel

$R+(L/2)$= Curvature radius of trajectory described by right wheel

The angular velocity of the mobile robot is found with respect to ICC is given as follows:

$$\omega_R(t) = \frac{V_R(t)}{R+(L/2)} \tag{3.1}$$

$$\omega_L(t) = \frac{V_l(t)}{R+(L/2)} \tag{3.2}$$

By substitute the right and left angular velocity,the total angular velocity of the mobile robot is given as:

$$\omega(t) = \frac{V_R(t)-V_L(t)}{L} \tag{3.3}$$

The instantaneous curvature radius of the mobile robot trajectory relative to the mid-point of the wheel axis is given as:

$$R = \frac{L(V_L(t)+V_R(t))}{2(V_L(t)-V_R(t))} \tag{3.4}$$

Therefore, the linear velocity of the mobile robot is given as:

$$V(t) = \frac{V_R(t)+V_L(t)}{2} \tag{3.5}$$

The kinematics Equations of the mobile robot in the world frame can be represented as:

$$\begin{aligned} \dot{X}(t) &= V(t)Cos\,\theta(t) \\ \dot{Y}(t) &= V(t)Sin\theta(t) \\ \dot{\theta}(t) &= \omega(t) \end{aligned} \tag{3.6}$$

The kinematic Equations (3.6) can be implied as:

$$\begin{aligned} X(t) &= \int_0^t V(t)\,Cos\left(\theta(t)\right)dt \\ Y(t) &= \int_0^t V(t)\,Sin\left(\theta(t)\right)dt \\ \theta(t) &= \int_0^t \omega(t)dt \end{aligned} \tag{3.7}$$

The above Equation (3.7) can also be represented in the following form:

$$\begin{pmatrix} V_x(t) \\ V_y(t) \\ \theta(t) \end{pmatrix} = \begin{pmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} V(t) \\ \omega(t) \end{pmatrix}$$

$$= \begin{pmatrix} V(t)\cos\theta \\ V(t)\, sin\theta \\ \omega(t) \end{pmatrix} \tag{3.8}$$

Hence, substitute *ω(t)* of (3.3) *V(t) (3.5)* into Equation (3.8):

$$\begin{pmatrix} V_x(t) \\ V_y(t) \\ \theta(t) \end{pmatrix} = \begin{pmatrix} \frac{1}{2}(V_R + V_L)\cos\theta \\ \frac{1}{2}(V_R + V_L)\, sin\theta \\ (V_R - V_L)/L \end{pmatrix} \tag{3.9}$$

Therefore, the kinematic model of a WMR is represented by Equation (3.9).

3.1.2 DC Motor Model

A common actuator in control systems is the DC motor. It directly provides rotary motion and, coupled with wheels or drums and cables, can provide transitional motion. The electric circuit of the armature and the free body diagram of the rotor are shown in Figure 3.2 [53]:

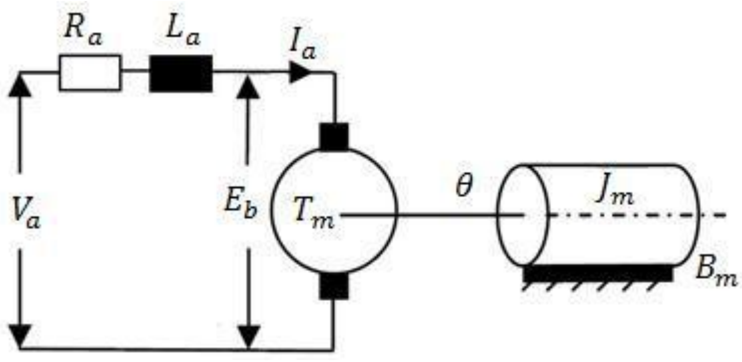

Figure 3.2: DC motor model [53]

The Figure 3.2 represents a DC motor attached to an inertial load. The voltages applied to the field and armature sides of the motor are represented by V_f and V_a .The resistances and inductances of the field and armature sides of the motor are represented by R_f, L_f,R_a and L_a.

The torque generated by the motor is proportional to$i_f$and $i_a$the currents in the field and armature sides of the motor is given as [53-54]:

$$T_m = K i_f i_a \tag{3.10}$$

In an field-current controlled motor, the armature current i_a is held constant, and the field current is controlled through the field voltage V_f In this case, the motor torque increases linearly with the field current. It can be written as:

$$T_m = K_{mf} i_f \tag{3.11}$$

By taking Laplace transforms of both sides of this equation gives the transfer function from the input current to the resulting torque. It is given as:

$$\frac{T_m(s)}{I_f(s)} = K_{mf} \tag{3.12}$$

The field side of the motor the voltage /current relationship is given as:

$$V_f = V_R + V_L \tag{3.13}$$

Hence;

$$V_f = R_f i_f + L_f(di_f/dt) \tag{3.14}$$

The transfer function is the first order system, from the input voltage to the resulting current is found by taking Laplace transforms of both sides of this equation is given as:

$$\frac{I_f(s)}{V_f(s)} = \frac{(1/L_f)}{s+(R_f/L_f)} \tag{3.15}$$

The transfer function from the input voltage to the resulting motor torque is found by combining Equations (3.12) and (3.14). It is given in first order system as:

$$\frac{I_f(s)}{V_f(s)} = \frac{T_m(s)}{I_f(s)}\frac{I_f(s)}{V_f(s)} = \frac{(K_{mf}/L_f)}{s+(R_f/L_f)} \tag{3.16}$$

Thus, a step input in field voltage results in an exponential rise in the motor torque. An equation that describes the rotational motion of the inertial load is found by summing moments as:

$$\sum M = T_m - c\omega = J\dot{\omega} \tag{3.17}$$

The counterclockwise directional of the rotational motion of the inertial load is considered as positive. The Equation 3.17 can be rewritten as:

$$J\dot{\omega} + c\omega = T_m \tag{3.18}$$

Therefore, the transfer function from the input motor torque to rotational speed changes in first order system is given as:

$$\frac{\omega(s)}{T_m(s)} = \frac{(1/J)}{s+(c/J)} \tag{3.19}$$

Equations (3.16) and (3.19) are combined and the second order system of the transfer function from the input field voltage to the resulting speed change is given as:

$$\frac{\omega(s)}{V_f(s)} = \frac{\omega(s)}{T_m(s)}\frac{T_m(s)}{V_f(s)} = \frac{(K_{mf}/L_f J)}{\left(s+\frac{c}{j}\right)(s+R_f/\,L_f)} \tag{3.20}$$

In an armature-current controlled motor, the field current i_f is held constant, and the armature current is controlled through the armature voltage V_a In this case, the motor torque increases linearly with the armature current. It can be written as:

$$T_m = K_{ma} i_a \tag{3.21}$$

The transfer function from the input armature current to the resulting motor torque is given as:

$$V_a = V_R + V_L + V_b \tag{3.22}$$

where V_b represents the "back EMF" induced by the rotation of the armature windings in a magnetic field. The back EMF V_b is proportional to the speed, i.e., $V_b(s)$= $K_b\omega(s)$.

Laplace transforms is used in Equation (3.22) gives as:

$$V_a(s) - V_b(s) = (R_a + L_a s)I_a(s) \tag{3.23}$$

Equations 3.23 can be written as:

$$V_a(s) - K_b\omega(s) = (R_a + L_a s)I_a(s) \tag{3.24}$$

Equations (3.19), (3.21) and (3.24) together can be represented by the closed loop block diagram of the DC motor armature voltage control system shown in Figure 3.3:

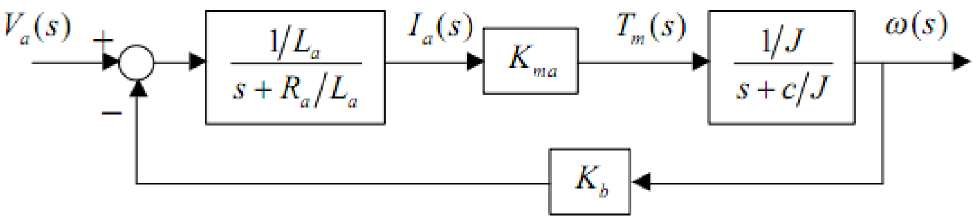

Figure 3.3: The block diagram of DC motor armature voltage control system

Block diagram reduction gives the second order system of the transfer function from the input armature voltage to the resulting speed change is given as:

$$\frac{\omega(s)}{V_a(s)} = \frac{(K_{ma}/L_a J)}{(s+R_a/L_a)(s+c/J)+(K_b K_{ma}/L_a J)} \tag{3.25}$$

From Equation (3.25) the armature inductance is very small in practices. Hence, the transfer function of DC motor speed to the input voltage can be simplified as follows:

$$\frac{\omega(s)}{V_a(s)} = \frac{K_{ma}}{(L_a s+R_a)(Js+ c)+K_b K_{ma}} \tag{3.26}$$

It also can be simplified as:

$$\frac{\omega(s)}{V_a(s)} = \frac{K_{ma}}{\tau s+1} \tag{3.27}$$

where;

$K_m = \frac{K_{ma}}{R_a c+K_b K_{ma}}$ is motor gain.

$\tau = \frac{R_a J}{R_a C+K_b K_{ma}}$ is the motor time constant.

From the Equation (3.27), the transfer function can be drawn in the DC motor system block diagram which is shown in Figure 3.4:

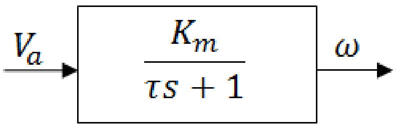

Figure 3.4: A simplified block diagram of DC motor armature voltage control system

Equations (3.9) and (3.27) that are used to build a model of the WMR. These equations were used to simulate the WMR in MATLAB Simulink. Several different controllers were tested and fine-tuned on this model, as well as compared with other controllers for optimum results.

The plant model is based on Maxon EC-4 pole 30 DC brushless motor, with R_a= 11.2 Ω, L_a= 0.1215 H , J= 0.02215 kgm^2, c=0.002953 *Nms/rad*, K_{ma}=1.28 *Nm/A* and K_b= 1.28 *Vs/rad*.The final model is as given in the following:

$$\frac{\omega(s)}{V_a(s)} = \frac{0.05}{0.00125s+1} \tag{3.28}$$

The datasheet for this motor is provided in Appendix A.

3.2 Controller

The objective of the controller is generate the velocities for both the right and the left motors of the WMR, which allows the WMR to move from one path to another path. This controller function would be based on the work carried out V.M. Peri [35] and R. Choomuang [55]. The controller used has two inputs; i.e., error in the distance and error in the angle of the wheel mobile robot. Figure 3.5 shows the mobile robot in Cartesian space and Figure 3.6 shows the input and output of controller:

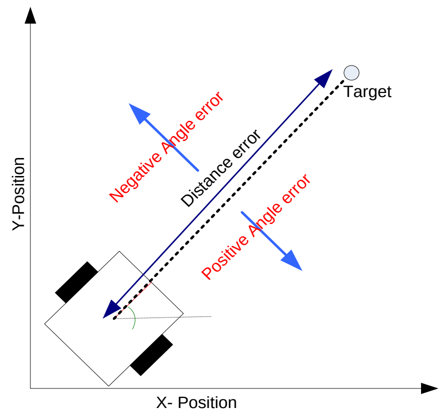

Figure 3.5: Wheel mobile robot in Cartesian space

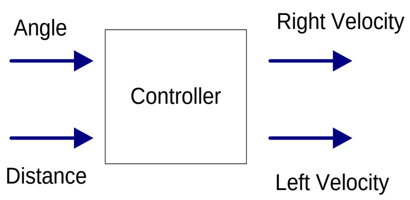

Figure 3.6:Input and Output of controller

From Figure 3.6, it can be seen that the input for the controller are the angle and the distance, whereas the output of the controller would be in the form of pulse-width-modulated signal to control the angular velocities of the left and right motors of the WMR.

Figure 3.7 shows the basic block diagram of wheel mobile robot system.

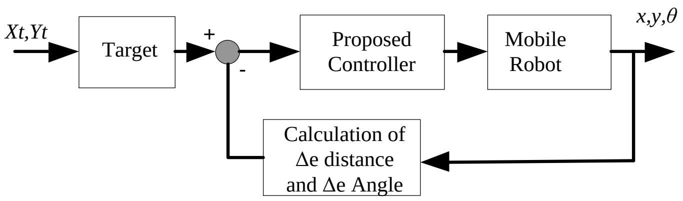

Figure 3.7: Block diagram of WMR system

3.2.1 Conventional Controller

A conventional controller is a generic control loop feedback mechanism widely used in industrial control systems. A conventional controller attempts to correct the error between a measured process variable and then outputting a corrective action that can adjust the process accordingly. The conventional controller calculation (algorithm) involves three conventional controller, Proportional controller (P), Proportional Integral controller (PI) and Proportional Integral derivative controller (PID). Figure 3.8 shows the block diagram of the conventional controller of WMR system and Figure 3.9 shows the simplified Proportional Integral Derivative controller simulation diagram.

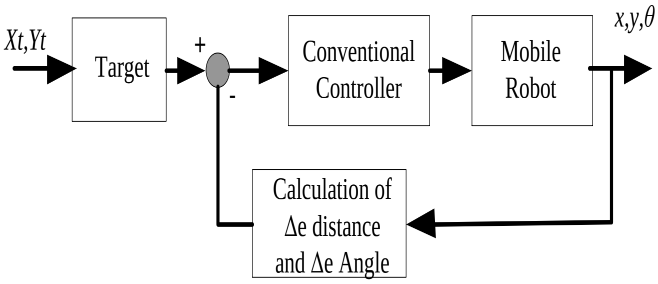

Figure 3.8:The block diagram conventional controller of WMR system.

Equation 3.29 of PID controller scheme is given as.

$$G_c(s) = K_P + \frac{K_I}{S} + K_D S = \frac{K_P S + K_I + K_D S^2}{S} = \frac{K_D(S^2 + \frac{K_P}{K_D}S + \frac{K_I}{K_D})}{S} \quad (3.29)$$

This transfer function is placed in MATLAB Simulink is shown Figure 3.9, The values of K_P, K_I and K_D are calculated using the root locus technique. The tuning method used is the Ziegler Nichols.

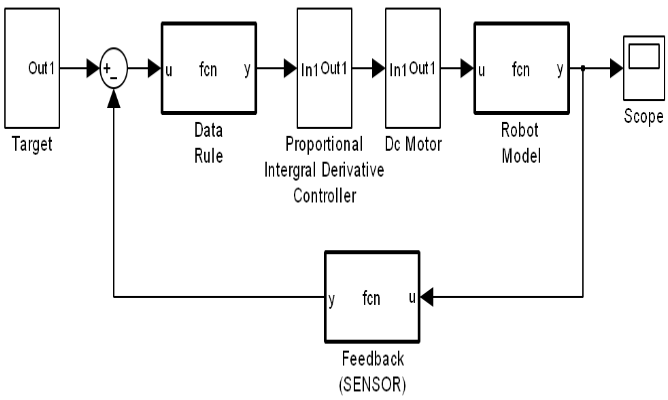

Figure 3.9**:**Simplified PID controller simulation diagram

3.2.2 Fuzzy Logic Controller

Recently, fuzzy logic controller (FLC) has become a popular tool for control applications which is originally developed by Lotfi Zadeh in the 1960's [56]. It has been used widely in applications such as servo motor, DC motor and process control. Figure 3.10 is the flow chart used in designing the FLC. In developing the design, initialization values were declared for input and output variables. Then, the mathematical model was defined so that simulation could be carried out using the previous value and the changes in the output. All parameters are then used by FLC to be simulated. FLC is a subroutine of the main simulator program designed in this project.

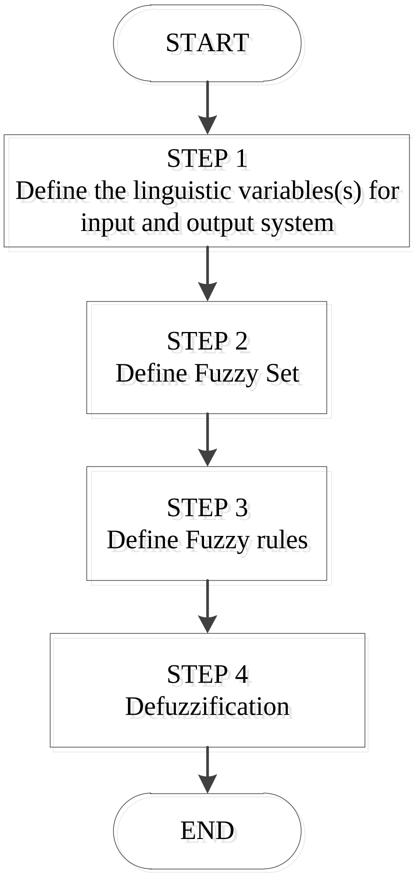

Figure 3.10: Fuzzy logic Algorithm

The fuzzy logic algorithm is as follows [55]:

Step 1: Define the linguistic variable(s) for input and output system.

- Compute two input variables: angle error (the different angle between goal angle and WMR heading) and distance error (the difference between the current position and goal position).
- Compute two output variables: the velocities of the left and right motors.

Step 2: Define fuzzy set

- The fuzzy variables is a set of overlapping values represented by triangular shape that is called the fuzzy membership function

Step 3: Define Fuzzy rules

- The operation of the system utilizes the fuzzy rule.

Step 4: Defuzzification

- The output action given by the input conditions, the centroid defuzzification computes the velocity output of the fuzzy controller.
- Centroid defuzzification or center of Area method calculated by Equation 3.30 as

$$V = \frac{\sum_{i=1}^{n} v_i X min(\mu_i^{\theta error}, \mu_i^{d error})}{\sum_{i=1}^{n} min(\mu_i^{\theta error}, \mu_i^{d error})} \tag{3.30}$$

where V is defined the fuzzy output of i^{th} rule,$\mu_i^{\theta error}$ and $\mu_i^{d error}$ are the membership value of the input fuzzy set.

Figure 3.11 shows the block diagram fuzzy controller of wheel mobile robot system and Figure 3.12 shows simplified fuzzy logic controller simulation diagram.

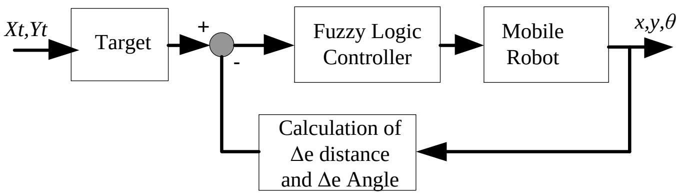

Figure 3.11: The block diagram fuzzy controller of WMR system.

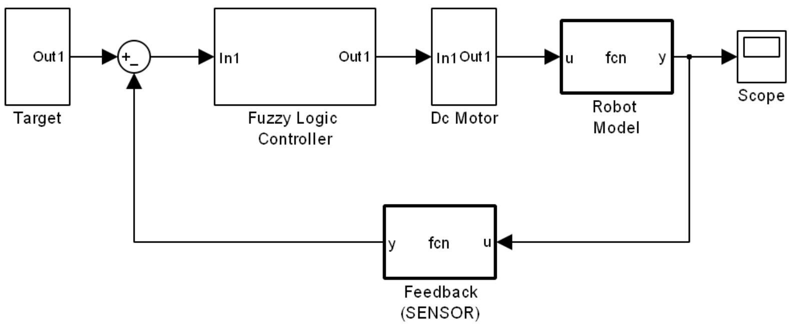

Figure 3.12: Simplified fuzzy logic controller simulation diagram

3.2.2.1   Fuzzy with Three membership function (F3M) Controller

The fuzzy with three membership function (F3M) controller is design to control the speed of motor of the wheel mobile robot (WMR) to follow a certain path. There are two input variables and two output variables for the controller. The input variables are distance error and angle error to a destination point while the output variables are the speed for the left motor and right motor respectively. Table 3.1 shows the Fuzzy rule for velocity of the right motor for fuzzy with three membership functions (F3M). Table 3.2 shows the fuzzy rule for velocity of the left motor for fuzzy with three membership functions (F3M).

Table 3.1: Fuzzy rule for velocity of the right motor

Output (u)		**Input Error distance (Δℓd)**		
		Fr	**Mi**	**Zo**
Input Error teta(Δℓd)	**Ne**	Me	Me	Sw
	Ze	Ft	Me	Sw
	Po	Ft	Ft	Ft

Table 3.2: Fuzzy rule for velocity of the left motor

Output (u)		**Input Error distance (Δℓd)**		
		Fr	**Mi**	**Zo**
Input Error teta(Δℓd)	**Ne**	Ft	Ft	Me
	Ze	Ft	Me	Sw
	Po	Me	Me	Sw

The nine Fuzzy rules for three membership of the WMR based on Tables 3.1 and 3.2 are as follows:

- Rule 1: IF ℓt is Negative angle (Ne) and ℓd is distance Far (Fr) then the velocity of the right motor is Medium (Me) and the velocity of the left motor is Fast (Ft).
- Rule 2: IF ℓt is Negative angle (Ne) and ℓd is distance Middle (Mi) then the velocity of the right motor is Medium (Me) and the velocity of the left motor is Fast (Ft).
- Rule 3: IF ℓt is Negative angle (Ne) and ℓd is distance Zero (Ze) then the velocity of the right motor is Slow (Sw) and the velocity of the left motor is Medium (Me)
- Rule 4: IF ℓt is Zero angle (Ze) and ℓd is distance Far (Fr) then the velocity of the right motor is Fast (Ft).and the velocity of the left motor is Fast (Ft).
- Rule 5: IF ℓt is Zero angle (Ze) and ℓd is distance Middle (Mi) then the velocity of the right motor is Medium (Me) and the velocity of the left motor is Medium (Me)
- Rule 6: IF ℓt is Zero angle (Ze) and ℓd is distance Zero (Zo) then the velocity of the right motor is Slow (Sw) and the velocity of the left motor is Slow (Sw)
- Rule 7: IF ℓt is Positive angle (Po) and ℓd is distance Far (Fr) then the velocity of the right motor is Fast (Ft) and the velocity of the left motor is Medium (Me)
- Rule 8: IF ℓt is Positive angle (Po) and ℓd is distance Middle (Mi) then the velocity of the right motor is Fast (Ft) and the velocity of the left motor is Medium (Me)

- Rule 9: IF ℓt is Positive angle (Po) and ℓd is distance Zero (Zo) then the velocity of the right motor is Fast (Ft) and the velocity of the left motor is Slow (Sw).

Figure 3.13 shows the representation input of the fuzzy three membership function of the distance error and angle error (theta error). The distance error was chosen to be between 0-14 because the target was set to be at 14m and the angle error was determined at +/- 1 to provide a very good accuracy. Figure 3.14 shows the representation output of the fuzzy three membership function velocity of the right motor and left motor. A value of 6 m/s was chosen for left and right motor respectively based on the speed of the Maxon EC-4 pole 30 DC brushless motor.

A similar set of range was used for F5M and F7M as well. Different membership functions were chosen to experiment with in order to determine the best set of membership functions for this particular application because the literature shows that the selection of fuzzy membership functions is application dependent.

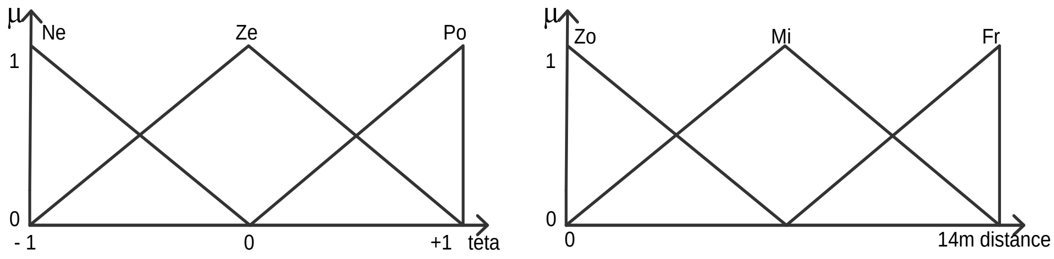

Figure 3.13: Input of the F3M function of the teta error and distance error

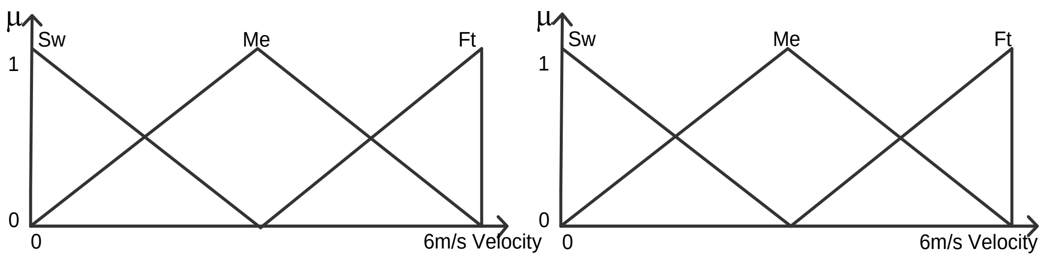

Figure 3.14: Output of the F3M function velocity of the right motor and left motor

Other membership function for F3M is given in Appendix B.

The validation of the defuzzification method can be explained through the use error in distance ($\Delta \ell d$) and the error in angle ($\Delta \ell t$). This could be applied to any number of membership function.

In this particular instance (F3M), although there are a total of nine (9) rules for the two wheels combined, at any instant only two given rules are fired thus making it easier to understand and debug the rules. Figure 3.15 shows how for a given error in distance and angle the corresponding rules are fired and the defuzzified output is arrived at. In this situation, the described error in distance ($\Delta \ell d$) is 1.81 and the error in angle ($\Delta \ell t$) is -0.495. Thus, the output using the centroid of area defuzzification, the centroid of the polygon is calculated where the crisp output for the right wheel is 2.63m/s and the crisp output of left wheel is 3.06 m/s.

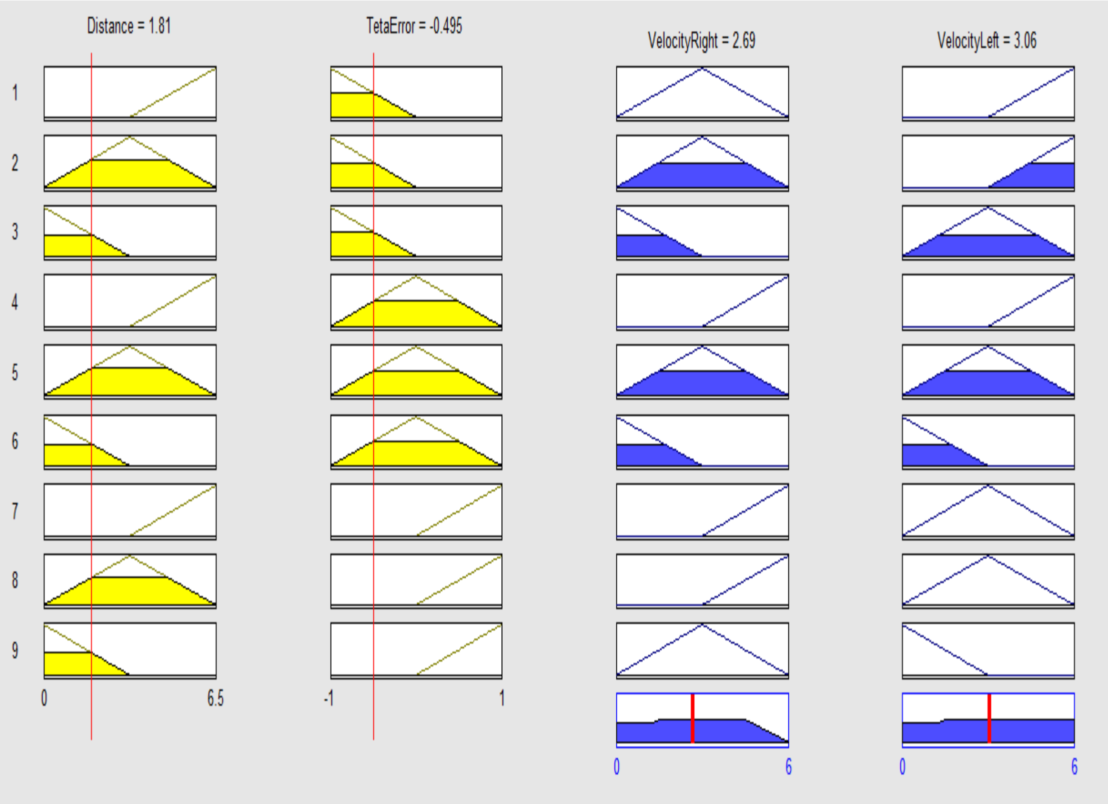

Figure 3.15: Firing F3M function rule velocity of the right and left motor

3.2.2.2 Fuzzy with Five membership function (F5M) Controller

The fuzzy logic controller is design to control the speed of motor of the WMR to follow the path. There are two input variables and two output variables for the controller. The input variables are distance error and angle error to a destination point while the output variables are the speed for the left motor and right motor respectively. Table 3.3 shows the fuzzy rule for velocity of the right motor for fuzzy with five membership function (F5M) controller. Table 3.4 shows the fuzzy rule for velocity of the left motor for fuzzy with five membership function (F5M) controller.

Table 3.3:Fuzzy rule for velocity of the right motor

Output (u)		Input Error distance (Δℓd)				
		Vfr	**Fr**	**Mi**	**Nr**	**Zo**
Input Error teta(Δℓd)	**Sn**	Me	Sw	Vs	Vs	Vs
	Ne	Ft	Me	Sw	Vs	Vs
	Ze	Vft	Ft	Me	Sw	Vs
	Po	Vft	Ft	Me	Sw	Sw
	Bp	Vft	Ft	Ft	Me	Me

Table 3.4:Fuzzy rule for velocity of the left motor

Output (u)		Error distance (Δℓd)				
		Vfr	**Fr**	**Mi**	**Nr**	**Zo**
Error teta(Δℓd)	**Sn**	Vft	Ft	Ft	Me	Me
	Ne	Vft	Ft	Me	Sw	Sw
	Ze	Vft	Ft	Me	Sw	Vs
	P	Ft	Me	Sw	Vs	Vs
	Bp	Me	Sw	Vs	Vs	Vs

The twenty five Fuzzy rules for five membership of the WMR based on Tables 3.3 and 3.4 are as follows:

- Rule 1: IF ℓt is Small Negative angle (Sn) and ℓd is distance Very Far (Vfr) then the velocity of the right motor is Medium (Me) and the velocity of the left motor is Very Fast (Vft).
- Rule 2: IF ℓt is Small Negative angle (Sn) and ℓd is distance Far (Fr) then the velocity of the right motor is Slow (Sw) and the velocity of the left motor is Fast (Ft).
- Rule 3: IF ℓt is Small Negative angle (Sn) and ℓd is distance Middle (Mi)

then the velocity of the right motor is Very Slow (Vs) and the velocity of the left motor is Fast (Ft).

- Rule 4: IF ℓt is Small Negative angle (Sn) and ℓd is distance Near (Nr) then the velocity of the right motor is Very Slow (Vs) and the velocity of the left motor is Medium (Me).
- Rule 5: IF ℓt is Small Negative angle (Sn) and ℓd is distance Zero (Zo) then the velocity of the right motor is Very Slow (Vs) and the velocity of the left motor is Medium (Me).
- Rule 6: IF ℓt is Negative angle (Ne) and ℓd is distance Very Far (Vfr) then the velocity of the right motor is Fast (Ft).and the velocity of the left motor is Very Fast (Vft).
- Rule 7: IF ℓt is Negative angle (Ne) and ℓd is distance Far (Fr) then the velocity of the right motor is Medium (Me) and the velocity of the left motor is Fast (Ft).
- Rule 8: IF ℓt is Negative angle (Ne) and ℓd is distance Middle (Mi) then the velocity of the right motor is Slow (Sw) and the velocity of the left motor is Medium (Me).
- Rule 9: IF ℓt is Negative angle (Ne) and ℓd is distance Middle (Mi) then the velocity of the right motor is Slow (Sw) and the velocity of the left motor is Medium (Me).
- Rule 10: IF ℓt is Negative angle (Ne) and ℓd is distance Zero (Zo) then the velocity of the right motor is Very Slow (Vs) and the velocity of the left motor is Slow (Sw).
- Rule 11: IF ℓt is Zero (Ze) and ℓd is distance Very Far (Vfr) then the velocity of the right motor is Very Fast (Vft).and the velocity of the left motor is Very Fast (Vft).
- Rule 12: IF ℓt is Zero (Ze) and ℓd is distance Far (Fr) then the velocity of the right motor is Fast (Ft) and the velocity of the left motor is Fast (Ft).
- Rule 13: IF ℓt is Zero (Ze) and ℓd is distance Middle (Mi) then the velocity of the right motor is Medium (Me) and the velocity of the left motor is Medium (Me).
- Rule 14: IF ℓt is Zero (Ze) and ℓd is distance Near (Nr) then the velocity of the right motor is Slow (Sw) and the velocity of the left motor is Slow (Sw).

- Rule 15: IF ℓt is Zero (Ze) and ℓd is distance Zero (Zo) then the velocity of the right motor is Very Slow (Vs) and the velocity of the left motor is Very Slow (Vs).
- Rule 16: IF ℓt is Positive (Po) and ℓd is distance Very Far (Vfr) then the velocity of the right motor is Very Fast (Vft) and the velocity of the left motor is Fast (Ft).
- Rule 17: IF ℓt is Positive (Po) and ℓd is distance Far (Fr) then the velocity of the right motor is Fast (Ft) and the velocity of the left motor is Medium (Me).
- Rule 18: IF ℓt is Positive (Po) and ℓd is distance Middle (Mi) then the velocity of the right motor is Medium (Me) and the velocity of the left motor is Slow (Sw).
- Rule 19: IF ℓt is Positive (Po) and ℓd is distance Near (Nr) then the velocity of the right motor is Slow (Sw) and the velocity of the left motor is Very Slow (Vs).
- Rule 20: IF ℓt is Positive (Po) and ℓd is distance Zero (Zo) then the velocity of the right motor is Slow (Sw) and the velocity of the left motor is Very Slow (Vs).
- Rule 21: IF ℓt is Big Positive (Bp) and ℓd is distance Very Far (Vfr) then the velocity of the right motor is Very Fast (Vft) and the velocity of the left motor is Medium (Me).
- Rule 22: IF ℓt is Big Positive (Bp) and ℓd is distance Far (Fr) then the velocity of the right motor is Fast (Ft) and the velocity of the left motor is Slow (Sw).
- Rule 23: IF ℓt is Big Positive (Bp) and ℓd is distance Middle (Mi) then the velocity of the right motor is Fast (Ft) and the velocity of the left motor is Very Slow (Vs).
- Rule 24: IF ℓt is Big Positive (Bp) and ℓd is distance Near (Nr) then the velocity of the right motor is Medium (Me) and the velocity of the left motor is Very Slow (Vs).
- Rule 25: IF ℓt is Big Positive (Bp) and ℓd is distance Zero (Zo) then the velocity of the right motor is Medium (Me) and the velocity of the left motor is Very Slow (Vs).

Figure 3.16 shows the representation input of the fuzzy five membership function of the distance error and theta error. Figure 3.17 shows the representation output of the fuzzy five membership function velocity of the right motor and left motor.

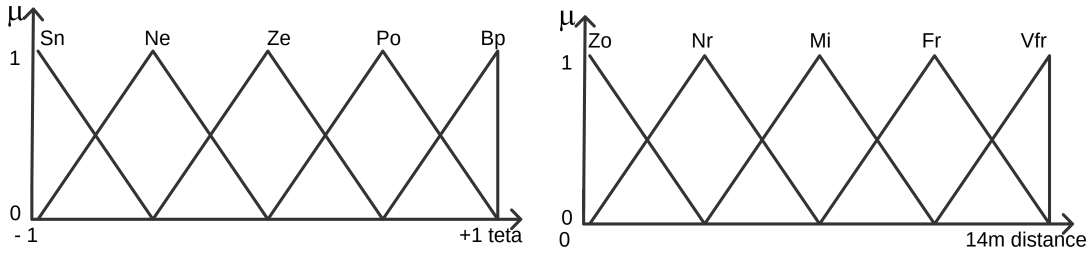

Figure 3.16: Input of the F5M function of the teta error and distance error

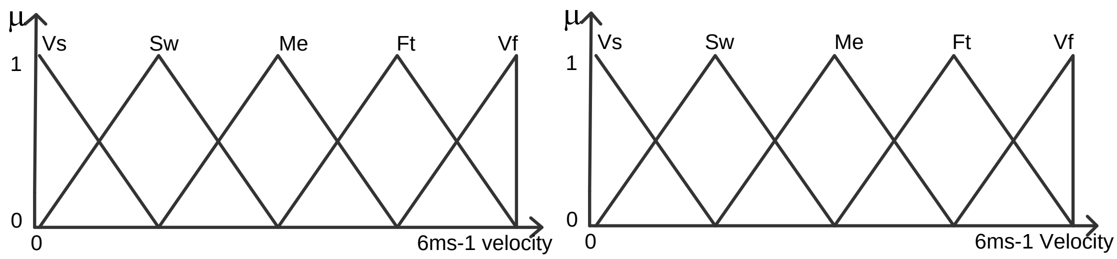

Figure 3.17: Output of the F5M function velocity of the right motor and left motor

Other membership function for F5M is given in Appendix B.

3.2.2.3 Fuzzy with Seven membership function (F7M) Controller

The fuzzy logic controller is design to control the speed of motor of the WMR to follow the path. There are two input variables and two output variables for the controller. The input variables are distance error and angle error to a destination point while the output variables are the speed for the left motor and right motor respectively. Table 3.5 has shown the Fuzzy rule for velocity of the right motor for Fuzzy with seven membership function (F7M) Controller. Table 3.6 has shown the Fuzzy rule for velocity of the left motor for Fuzzy with seven membership function (F7M) Controller.

Table 3.5 : Fuzzy rule for velocity of the right motor

Output (u)		Error distance (∆ℓd)						
		Vbfr	**Vfr**	**Fr**	**Mi**	**Nr**	**Vnz**	**Zo**
Error teta (∆ℓd)	**Vsn**	Me	Ft	Sw	Vs1	Vs1	Vs2	Vs2
	Sn	Ft	Me	Sw	Vs1	Vs1	Vs1	Vs2
	Ne	Vf1	ft	Me	Sw	Vs1	Vs1	Vs2
	Ze	Vf2	Vf1	Ft	Me	Sw	Vs1	Vs2
	Po	Vf2	Vf1	Ft	Me	Sw	Sw	Vs1
	Bp	Vf2	Vf1	Ft	Ft	Me	Me	Sw
	Vbp	Vf2	Vf2	Vf1	Ft	Me	Me	Sw

Table 3.6:Fuzzy rule for velocity of the left motor

Output (u)		Error distance (∆ℓd)						
		Vbfr	**Vfr**	**Fr**	**Mi**	**Nr**	**Vnz**	**Zo**
Error teta (∆ℓd)	**Vsn**	Vf2	Vf2	Vf1	Ft	Me	Me	Sw
	Sn	Vf2	Vf1	Ft	Ft	Me	Me	Sw
	Ne	Vf2	Vf1	Ft	Me	Sw	Sw	Vs1
	Ze	Vf2	Vf1	Ft	Me	Sw	Sw	Vs2
	Po	Vf1	Ft	Me	Sw	Vs1	Vs1	Vs2
	Bp	Ft	Me	Sw	Vs1	Vs1	Vs1	Vs2
	Vbp	Me	Ft	Sw	Vs1	Vs1	Vs1	Vs2

The forty nine fuzzy rules for seven membership of the WMR based on Tables 3.5 and 3.6 are as follows:

- Rule 1: IF ℓt is Very Small Negative angle (Vsn) and ℓd is distance Very Big Positive (Vbp) then the velocity of the right motor is Medium (Me) and the velocity of the left motor is Very Fast 2 (Vf2).

- Rule 2: IF ℓt is Very Small Negative angle (Vsn) and ℓd is distance Very Far (Vfr) then the velocity of the right motor is Fast (Ft) and the velocity of the left motor is Very Fast 2 (Vf2).
- Rule 3: IF ℓt is Very Small Negative angle (Vsn) and ℓd is distance Far (Fr) then the velocity of the right motor is Slow (Sw) and the velocity of the left motor is Very Fast 1 (Vf1).
- Rule 4: IF ℓt is Very Small Negative angle (Vsn) and ℓd is distance Far (Fr) then the velocity of the right motor is Slow (Sw) and the velocity of the left motor is Very Fast 1 (Vf1).
- Rule 5: IF ℓt is Very Small Negative angle (Vsn) and ℓd is distance Near (Nr) then the velocity of the right motor is Very Slow 1 (Vs1) and the velocity of the left motor is Medium (Me).
- Rule 6: IF ℓt is Very Small Negative angle (Vsn) and ℓd is distance Very Near Zero (Vnz) then the velocity of the right motor is Very Slow 2 (Vs2) and the velocity of the left motor is Slow (Sw).
- Rule 7: IF ℓt is Very Small Negative angle (Vsn) and ℓd is distance Zero (Ze) then the velocity of the right motor is Very Slow 2 (Vs2) and the velocity of the left motor is Slow (Sw).
- Rule 8: IF ℓt is Small Negative angle (Sn) and ℓd is distance Very Big Far (Vbf) then the velocity of the right motor is Fast (Ft) and the velocity of the left motor is Very Fast 2 (Vf2).
- Rule 9: IF ℓt is Small Negative angle (Sn) and ℓd is distance Very Far (Vfr) then the velocity of the right motor is Medium (Me) and the velocity of the left motor is Very Fast 1 (Vf1).
- Rule 10: IF ℓt is Small Negative angle (Sn) and ℓd is distance Far (Fr) then the velocity of the right motor is Slow (Sw) and the velocity of the left motor is Fast (Ft).
- Rule 11: IF ℓt is Small Negative angle (Sn) and ℓd is distance Far (Fr) then the velocity of the right motor is Slow (Sw) and the velocity of the left motor is Fast (Ft).
- Rule 12: IF ℓt is Small Negative angle (Sn) and ℓd is distance Near (Nr) then

the velocity of the right motor is Very Slow 1 (Vs1) and the velocity of the left motor is Medium (Me).

- Rule 13: IF ℓt is Small Negative angle (Sn) and ℓd is distance Very Near Zero (Vnz) then the velocity of the right motor is Very Slow 1 (Vs1) and the velocity of the left motor is Medium (Me).
- Rule 14: IF ℓt is Small Negative angle (Sn) and ℓd is distance Zero (Ze) then the velocity of the right motor is Very Slow 2 (Vs2) and the velocity of the left motor is Slow (Sw).
- Rule 15: IF ℓt is Negative angle (Ne) and ℓd is distance Very Big Far (Vbf) then the velocity of the right motor is Very Fast 1 (Vf1) and the velocity of the left motor is Very Fast 2 (Vf2).
- Rule 16: IF ℓt is Negative angle (Ne) and ℓd is distance Very Far (Vfr) then the velocity of the right motor is Fast (Ft) and the velocity of the left motor is Very Fast 1 (Vf1).
- Rule 17: IF ℓt is Negative angle (Ne) and ℓd is distance Far (Fr) then the velocity of the right motor is Medium (Me) and the velocity of the left motor is Fast (Ft).
- Rule 18: IF ℓt is Negative angle (Ne) and ℓd is distance Middle (Mi) then the velocity of the right motor is Slow (Sw) and the velocity of the left motor is Medium (Me).
- Rule 19: IF ℓt is Negative angle (Ne) and ℓd is distance Near (Nr) then the velocity of the right motor is Very Slow 1 (Vs1) and the velocity of the left motor is Slow (Sw).
- Rule 20: IF ℓt is Negative angle (Ne) and ℓd is distance Very Near Zero (Vnz) then the velocity of the right motor is Very Slow 1 (Vs1) and the velocity of the left motor is Slow (Sw).
- Rule 21: IF ℓt is Negative angle (Ne) and ℓd is distance Zero (Zo) then the velocity of the right motor is Very Slow 2 (Vs2) and the velocity of the left motor is Very Slow 1 (Vs1).
- Rule 22: IF ℓt is Zero angle (Ze) and ℓd is distance Very Big Far (Vbf) then the velocity of the right motor is Very Fast 2 (Vf2) and the velocity of the left motor is Very Fast 2 (Vf2).

- Rule 23: IF ℓt is Zero angle (Ze) and ℓd is distance Very Far (Vfr) then the velocity of the right motor is Very Fast 1 (Vf1) and the velocity of the left motor is Very Fast 1 (Vf1).
- Rule 24: IF ℓt is Zero angle (Ze) and ℓd is distance Far (Fr) then the velocity of the right motor is Fast (Ft) and the velocity of the left motor is Fast (Ft).
- Rule 25: IF ℓt is Zero angle (Ze) and ℓd is distance Middle (Mi) then the velocity of the right motor is Medium (Me) and the velocity of the left motor is Medium (Me).
- Rule 26: IF ℓt is Zero angle (Ze) and ℓd is distance Near (Nr) then the velocity of the right motor is Slow (Sw) and the velocity of the left motor is Slow (Sw).
- Rule 27: IF ℓt is Zero angle (Ze) and ℓd is distance Very Near Zero (Vnz) then the velocity of the right motor is Very Slow 1 (Vs1) and the velocity of the left motor is Very Slow 1 (Vs1).
- Rule 28: IF ℓt is Zero angle (Ze) and ℓd is distance Zero (Zo) then the velocity of the right motor is Very Slow 2 (Vs2) and the velocity of the left motor is Very Slow 2 (Vs2).
- Rule 29: IF ℓt is Positive angle (Po) and ℓd is distance Very Big Far (Vbf) then the velocity of the right motor is Very Fast 2 (Vf2) and the velocity of the left motor is Very Fast 1 (Vf1).
- Rule 30: IF ℓt is Positive angle (Po) and ℓd is distance Very Big Far (Vbf) then the velocity of the right motor is Very Fast 1 (Vf1) and the velocity of the left motor is Fast (Ft).
- Rule 31: IF ℓt is Positive angle (Po) and ℓd is distance Far (Fr) then the velocity of the right motor is Fast (Ft) and the velocity of the left motor is Medium (Me).
- Rule 32: IF ℓt is Positive angle (Po) and ℓd is distance Middle (Mi) then the velocity of the right motor is Medium (Me) and the velocity of the left motor is Slow (Sw).
- Rule 33: IF ℓt is Positive angle (Po) and ℓd is distance Near (Ne) then the velocity of the right motor is Slow (Sw) and the velocity of the left motor is Very Slow 1 (Vs1).

- Rule 34: IF ℓt is Positive angle (Po) and ℓd is distance Very Near Zero (Vnz) then the velocity of the right motor is Slow (Sw) and the velocity of the left motor is Very Slow 1 (Vs1).
- Rule 35: IF ℓt is Positive angle (Po) and ℓd is distance Zero (Zo) then the velocity of the right motor is Very Slow 1 (Vs1) and the velocity of the left motor is Very Slow 2 (Vs2).
- Rule 36: IF ℓt is Big Positive angle (Bp) and ℓd is distance Very Big Far (Vbf) then the velocity of the right motor is Very Fast 2 (Vf2) and the velocity of the left motor is Fast (Ft).
- Rule 37: IF ℓt is Big Positive angle (Bp) and ℓd is distance Very Far (Vf) then the velocity of the right motor is Very Fast 1 (Vf1) and the velocity of the left motor is Medium (Me).
- Rule 38: IF ℓt is Big Positive angle (Bp) and ℓd is distance Far (Fr) then the velocity of the right motor is Fast (Ft) and the velocity of the left motor is Slow (Sw).
- Rule 39: IF ℓt is Big Positive angle (Bp) and ℓd is distance Middle (Mi) then the velocity of the right motor is Fast (Ft) and the velocity of the left motor is Very Slow 1 (Vs1).
- Rule 40: IF ℓt is Big Positive angle (Bp) and ℓd is distance Near (Nr) then the velocity of the right motor is Medium (Me) and the velocity of the left motor is Very Slow 1 (Vs1).
- Rule 41: IF ℓt is Big Positive angle (Bp) and ℓd is distance Very Near Zero (Vnz) then the velocity of the right motor is Medium (Me) and the velocity of the left motor is Very Slow 1 (Vs1).
- Rule 42: IF ℓt is Big Positive angle (Bp) and ℓd is distance Zero (Ze) then the velocity of the right motor is Slow (Sw) and the velocity of the left motor is Very Slow 2 (Vs2).
- Rule 43: IF ℓt is Very Big Positive angle (Vbp) and ℓd is distance Very Big Far (Vbf) then the velocity of the right motor is Very Fast 2 (Vf2) and the velocity of the left motor is Medium (Me).
- Rule 44: IF ℓt is Very Big Positive angle (Vbp) and ℓd is distance Very Far (Vfr) then the velocity of the right motor is Very Fast 2 (Vf2) and the velocity of the left motor is Fast (Ft).

- Rule 45: IF ℓt is Very Big Positive angle (Vbp) and ℓd is distance Far (Fr) then the velocity of the right motor is Very Fast 1 (Vf1) and the velocity of the left motor is Slow (Sw).
- Rule 46: IF ℓt is Very Big Positive angle (Vbp) and ℓd is distance Middle (Mi) then the velocity of the right motor is Fast (Ft) and the velocity of the left motor is Very Slow 1 (Vs1).
- Rule 47: IF ℓt is Very Big Positive angle (Vbp) and ℓd is distance Near (Nr) then the velocity of the right motor is Medium (Me) and the velocity of the left motor is Very Slow 1 (Vs1).
- Rule 48: IF ℓt is Very Big Positive angle (Vbp) and ℓd is distance Very Near Zero (Vnz) then the velocity of the right motor is Medium (Me) and the velocity of the left motor is Very Slow 2 (Vs2).
- Rule 49: IF ℓt is Very Big Positive angle (Vbp) and ℓd is distance Zero (Zo) then the velocity of the right motor is Slow (Sw) and the velocity of the left motor is Very Slow 2 (Vs2).

Figure 3.18 shows the representation input of the fuzzy seven membership function of the distance error and teta error. Figure 3.19 shows the representation output of the fuzzy seven membership function velocity of the right motor and left motor.

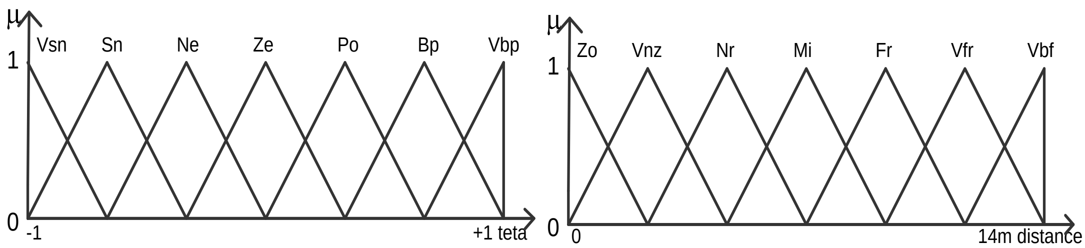

Figure 3.18: Input of the F7M function of the theta error and distance error

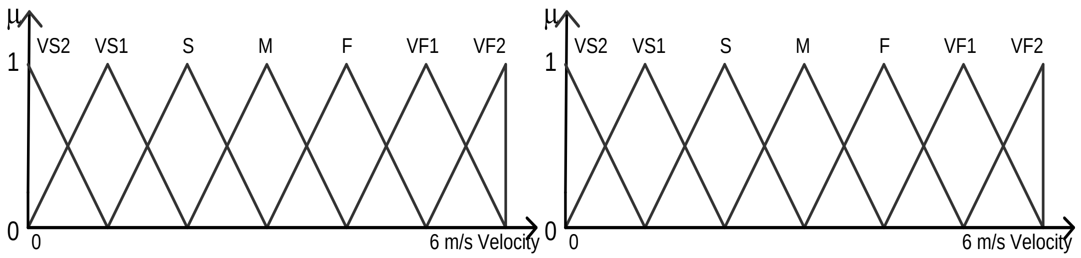

Figure 3.19: Output of the F7M function velocity of the Right motor and left motor

Other membership functions for F7M is given in Appendix B.

3.2.3 Hybrid Controller

This section discusses the hybrid controller of wheel mobile robots (WMR).The hybrid fuzzy and proportional (FP) controller in section 3.2.3.1, fuzzy with proportional integral (FPI) controller in section 3.2.3.2 and fuzzy and proportional integral derivative (FPID) controller in section 3.2.3.3

3.2.3.1 Fuzzy and Proportional (FP) Controller

Figure 3.20 shows the block diagram fuzzy and proportional controller of wheel mobile robot (WMR) system and Figure 3.21 shows the simplified fuzzy logic and proportional controller simulation diagram.

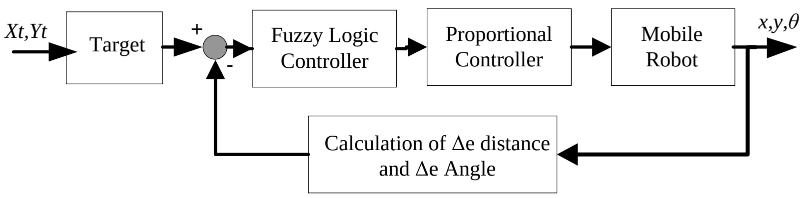

Figure 3.20: The block diagram FP controller of WMR system.

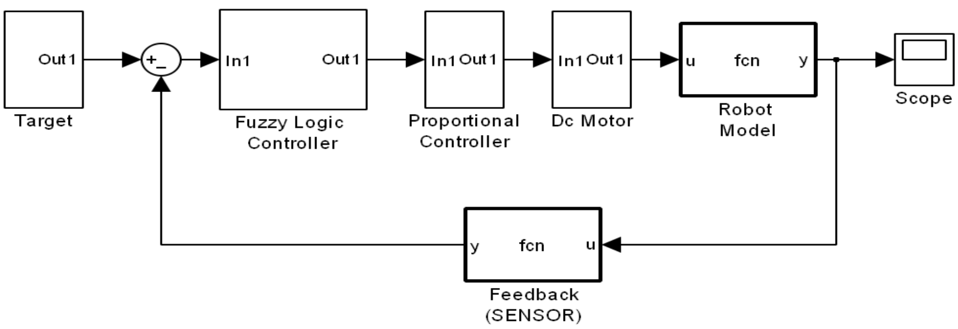

Figure 3.21: Simplified FP controller simulation diagram

3.2.3.2 Fuzzy with Proportional Integral (FPI) Controller

The output of the fuzzy control would be pulse-width- modulated signal to control are angular velocities of the left and right motors of the wheel mobile robot (WMR). Figure 3.22 shows the block diagram FPI controller of mobile robot system and Figure 3.23 shows the simplified fuzzy logic and fuzzy and proportional integral controller simulation diagram.

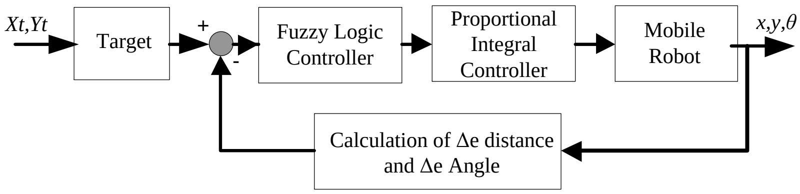

Figure 3.22: Block diagram FPI controller of WMR system.

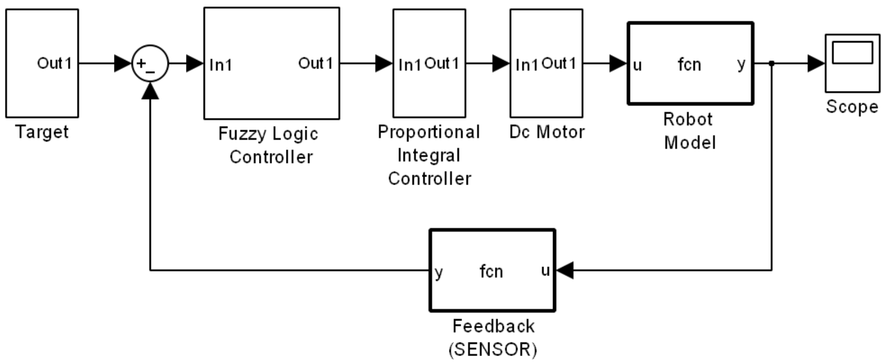

Figure 3.23: Simplified FPI controller simulation diagram

3.2.3.3 Fuzzy and Proportional Integral Derivative (FPID) Controller

The output of the fuzzy control would be pulse-width- modulated signal to control are angular velocities of the left and right motors of the wheel mobile robot (WMR). Figure 3.24 shows the block diagram fuzzy and proportional integral derivative controller of wheel mobile robot system and Figure 3.25 shows the Simplified Fuzzy and proportional integral derivative controller simulation diagram.

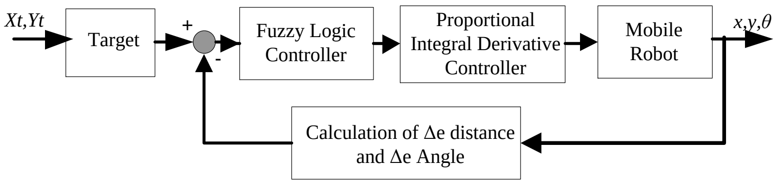

Figure 3.24: Block diagram FPID controller of WMR system

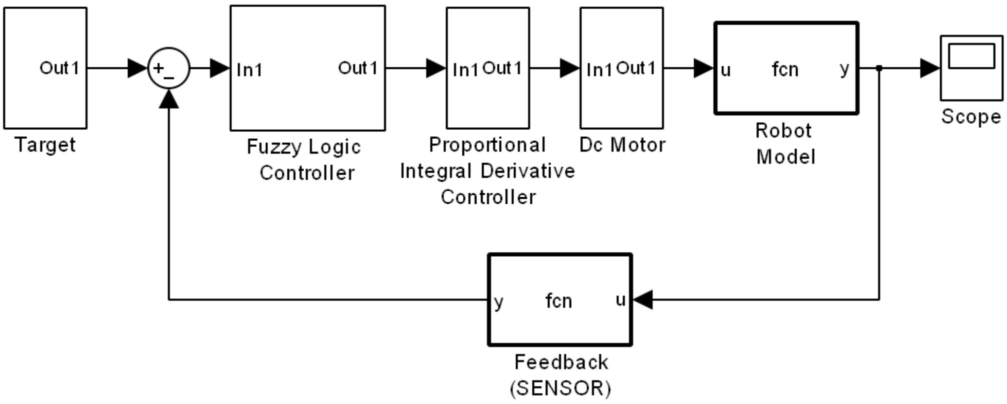

Figure 3.25: Simplified FPID controller simulation diagram

3.3 Optimization of Path Planning

This section discusses the Optimization of Path Planning of wheel mobile robots (WMR). The Principles of Ant Colony Optimization (ACO) Technique is discussed in section 3.3.1 and the description of Ant Colony Optimization Algorithm is presented in section 3.3.2.

3.3.1 Principles of Ant Colony Optimization Technique

The Travel salesman problem (TSP) is a paradigmatic optimization problem since it is used to demonstrate the original Ant System (AS) problem [57]. Since then it has often been used as a benchmark to test the new ACO concept. The proposed path-finding algorithm employs the original concept of the ACO algorithm with some modification.

The idea of the proposed algorithm is as follows: Starting from the grid point ***(0, 0)***, an ant iteratively moves from a grid point to one of its neighboring grid points. When at the ***ith*** grid point ***(x,y)***, ant ***k*** can choose the next ***jth*** grid point by choosing one of its 8 neighbor locations: [***(x-1, y+1), (x,y+1), (x+1, y+1),(x-1,y),(x,y),(x+1,y),(x-1,y-1),(x,y-1)*** and ***(x+1,y-1)***] [49-51]. The Ant Neighbouring is shown in Figure 3.26. where *N* in general is the set of all neighboring locations of the current location. The ant takes its next step randomly, based on the probability given by

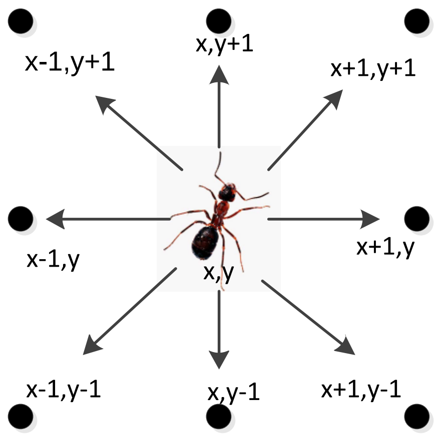

Figure 3.26: Ant neighboring

$$\varphi_{ij}^{k} = \frac{\tau_{ij}(t)}{\sum_{l \epsilon N} \tau_{il}} + \alpha^{*} \partial_{ij}(t) \qquad (3.31)$$

where $\boldsymbol{\tau_{ij}(t)}$ is accumulated pheromone on the ***jth*** grid point when the ***ith*** grid point is the ant's current location at time$(\boldsymbol{t})$. The quantity, $\boldsymbol{\tau_{ij}}$ indicates every possible ***lth*** neighbor point when the ant is in ***ith*** position. The $\boldsymbol{\partial_{ij}(t)}$is the dot product of the vector from $\boldsymbol{i}$ to $\boldsymbol{j}$ with the vector $\boldsymbol{i}$ to destination point. The first term is associated with the pheromone amount in the 8 neighboring grid points (a local pheromone). The second term is associated with a global attraction, where $\boldsymbol{\alpha}$ is a scale parameter. The global term is defined as the dot product of the agent's heading direction and the food (destination) direction. The purpose of the global term is to guide the agent in the desired direction so that a large number of ants can reach the goal point in a limited number of steps. Fig 3.27 shows an agent's current position **(x, y)** with 8 possible next positions.

(x-1,y+1)	*(x,y+1)*	*(x+1,y+1)*
(x-1,y)	*(x,y)*	*(x+1,y)*
(x-1,y-1)	*(x,y-1)*	*(x+1,y-1)*

Figure 3.27: An agent's current position (x,y) with 8 possible next positions

The solution construction ends after each ant reaches the destination or the ant took too many steps (50 or 100 steps, for example). However, the pheromone will be deposited only if the ant reaches to the destination position in less than a certain number of steps. Thus, after a group of agents finished its tour, the pheromone in the entire map will be updated by

$$\tau_{ij}(t+1) = (1-\rho)^{*}\tau_{ij}(t) + \sum_{k=1}^{m} \Delta\, \gamma_{ij}^{k}(t) \qquad (3.32)$$

Where $\mathbf{0} < \rho < 1$is the pheromone forgetting parameter. This parameter prevents the map from unlimited accumulation of the pheromone. The quantity $\Delta\boldsymbol{\gamma}_{\boldsymbol{ij}}^{\boldsymbol{k}}$is the amount of pheromone that ant k deposits on the map. It is defined as

$$\gamma_{ij}^{k}(t) = 1/L^{k}(t) \qquad (3.33)$$

Where $\boldsymbol{L^{k}(t)}$is the length of the ***kth*** ant's tour (path). When the ***kth*** ant cannot reach the destination in a certain number of steps, let the $\boldsymbol{L^{k}(t)}$ **=Infinity**.

3.3.2 Ant Colony Optimization (ACO) Algorithm

The ACO algorithm used in this experiment is the Ant System (AS) algorithm proposed by Marco Dorigo [58]. However, as shown in Equation 3.32, a new heuristic equation of the state transition rules, which is more suitable with applications of this research, was used. The evaluation of fitness function and ACO parameter setting was created based on the requirements of this research.

The design of ACO for mobile robot path planning (MRPP) was divided into three important rules which are state transition rules, local update rules and global update rules. At the beginning, ants will determine the next node to be visited by using the state transition rules based on heuristic and pheromone laid down by the ants.

An accurate value of distance by heuristic equation and the higher amount of pheromone of the visited node will be obtained by the ants that have higher probability to choose that nodes. Within these rules, ants can balance between the exploration and exploitation from the relatives' coefficient provided. During the construction of the path, the pheromone will be reduced locally by the given evaporation rate by using the formula of update local rules. After all the ants complete the path to goal, then the process of global updating is applied where ants will deposits its pheromone based on the path distance.

The amount of pheromone will continuously be updated until it attracts more ants from the next generation to follow the shorter path. Finally, the optimal mobile robot path is found by using behavior of ants' concept as shown in Fig. 3.28 [57-60].

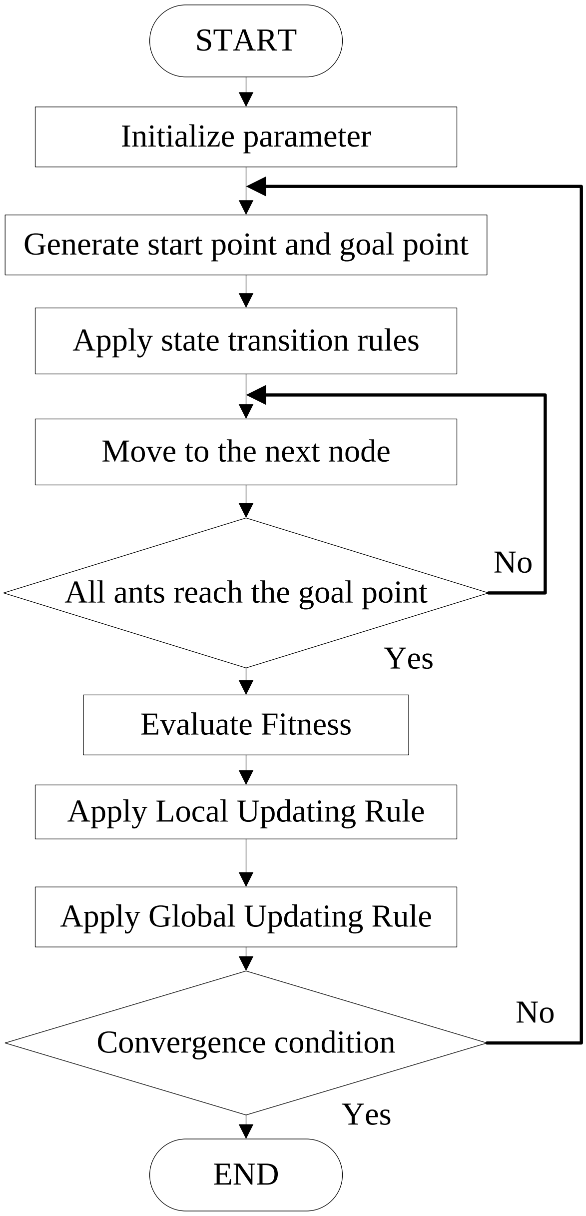

Figure 3.28: Flowchart Ant Colony Optimization Algorithm [57],[60-62] .

A program was written to produce the final output (path lengths) based on the last iteration. Upon reaching the last iteration, the program stops and outputs the final result.

3.4 Summary

This chapter has provided discussion on the modeling of mobile robot in terms of kinematic model of wheel mobile robot and dc motor model. This was followed by discussion on simulation work using controllers such as conventional controller, fuzzy logic controller and hybrid controller. Finally, it discussed the optimization of path planning wheel mobile robots in terms of the principles of ant colony optimization technique and the ant colony optimization algorithm. The findings and analysis of the simulation work are presented in Chapter 4.

CHAPTER 4

RESULT AND DISCUSSION

This chapter discusses the results of the controller design and path planning for wheel mobile robot differential drive. The first section of this chapter is the controller design with conventional controller, fuzzy logic controller and hybrid controller for wheel mobile robot differential drive. The second section discusses results on the wheel mobile robot path planning.

4.1 Controller

In this section, the controller design is divided into three categories which are conventional controller, fuzzy logic controller and hybrid controller. These controllers were chosen to carry out simulation in order to determine the best controller amongst them that can provide the best performance for WMR. The conventional controller is described in section 4.1.1, the fuzzy logic controller in section 4.1.2, then the hybrid controller in section 4.1.3.

4.1.1 Conventional Controller

This differential drive wheel mobile robots is initially tested to move from initial set point (x position= 0 metre, y position=0 metre) to the end set point (x position=10 metre y position=10) by using selected conventional controller. The plots shown in Figure 4.1, 4.2 and 4.3 are the simulation results obtained from the response of the robot with proportional (P), proportional integral (PI) and proportional integral derivation (PID) controller. Figure 4.1 shows the angle of orientation of the robot as a function of time while Figure 4.2 shows the right and left velocity of the mobile robot as a function of time. Meanwhile, Figure 4.3 shows the position of mobile robot with respect to the frame of x position and y position.

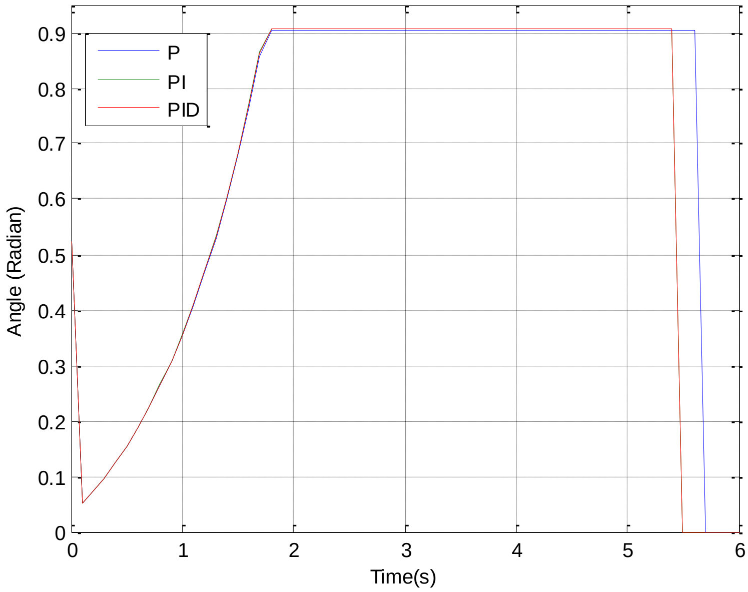

Figure 4.1: Comparison of angle of orientation of P, PI and PID controller for 10x10y

It can be seen from Figure 4.1 that the change of angle occurs until it reaches steady state at 1.8 seconds for P, PI and PID controllers. The time taken for each target point are 5.7 seconds for P and 5.6 seconds for both PI and PID controllers. Proportional integral derivation (PID) and proportional integral (PI) controller for each target point gives better response as compared to proportional (P) controller of differential drive wheel mobile robots.

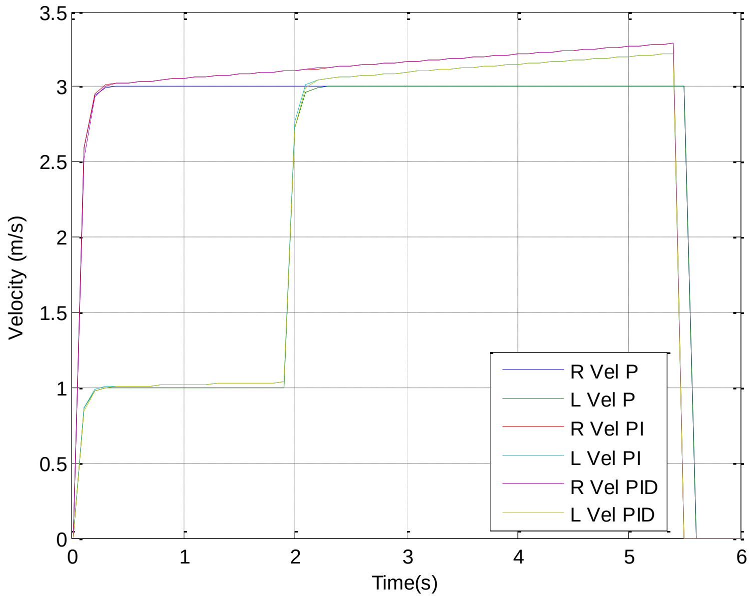

Figure 4.2:Right and left velocity of WMR conventional controller for 10x10y

From Figure 4.2, the right velocity (R Vel P) is indicated by the blue colour line while the left velocity (L Vel P) is indicated by the green colour line. These lines show the velocity of P controller. The right velocity (R Vel PI) and the left velocity (L Vel PI) of the PI controller is indicated by the red and turquoise colour line respectively. The violet and olive colour line on the other hand, shows the right velocity (R Vel PID) and left velocity (L Vel PID) of the PID controller.

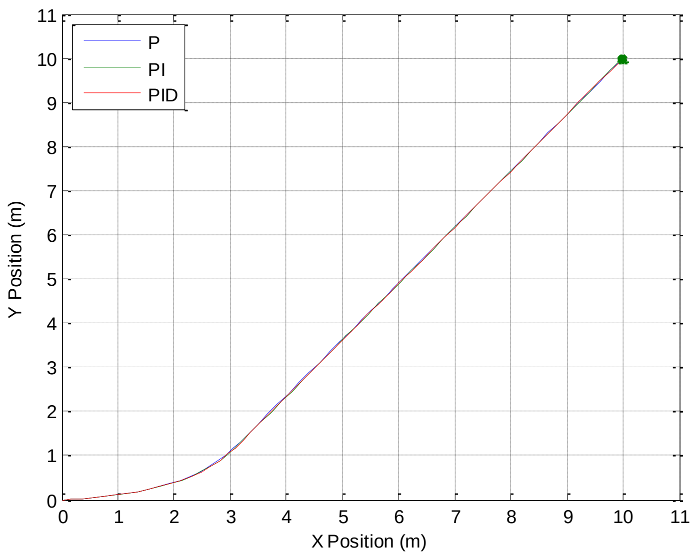

Figure 4.3: Position of WMR using convnetional controller for 10x10y

It can seen from Figure 4.3 that the position of WMR using conventional controllers for 10x10y are the same for the P, PI and PID controllers.

Table 4.1 summarizes the performance evaluation of different conventional controller of differential drive WMR. These differential drive WMRs have been initially tested to move from initial set point (x position=0 metre,y position= 0 metre) to the end set point (x position = 10 metre y position= 10 metre), (x position =30 metre y position =10 metre), (x position=60 metre y position = 20 metre).

Table 4.1 Performance evaluation of difference conventional controller of WMR

Controller Type	**Target Point**		
	10x 10y	30x10y	60x20y
P	5.7s Kp=20	9.4s Kp=30	18.2s Kp=30
PI	5.6s Kp=20,Ki=0.0375	8.9s Kp=30,Ki=0.0375	16.5s Kp=20,Ki=0.0375
PID	5.6s Kp=20,Ki=0.0375 Kd=0.009	7.8s Kp=30,Ki=0.0375 Kd=0.009	14.7s Kp=30,Ki=0.0375 Kd=0.009

Figure 4.4 shows the performance evaluation of different conventional controller of differential drive wheel mobile robots based on Table 4.1.

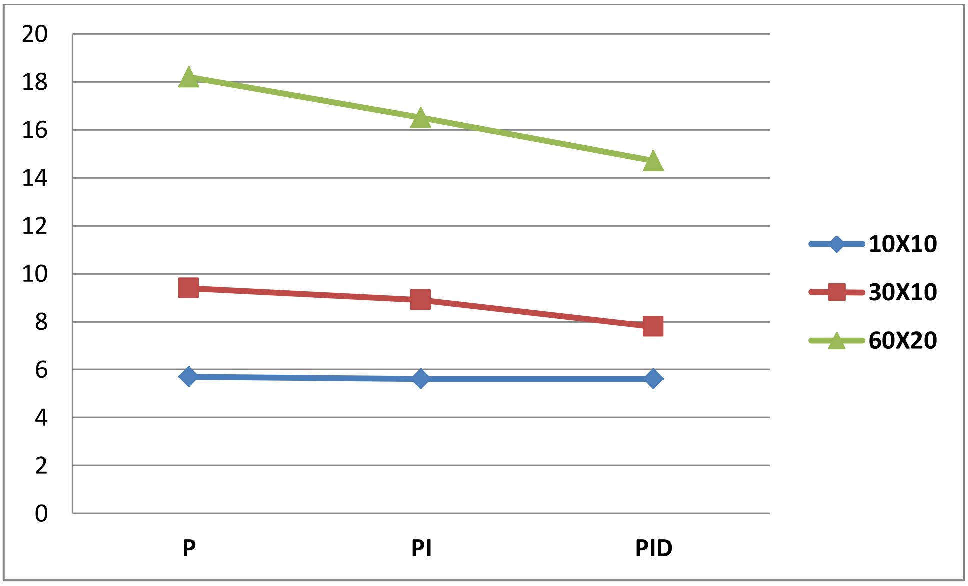

Figure 4.4:Performance evaluation of different conventional controller of WMR

It is shown that a model simulated using different type of conventional controller with proportional (P) ,proportional integral (PI) and proportional integral derivative (PID) controller and different target point for differential drive wheel mobile robots gives different values of time in seconds. Proportional integral derivation (PID) controller for each target point gives better response 5.6s, 7.8s and 14.7s for target of 10x 10y,30x10y and 60x20y respectively, as compared to proportional integral (PI) 5.6s, 8.9s and 16.5s and proportional (P) with 5.7s, 9.4s and 18.2s .

4.1.2 Fuzzy Logic Controller (F)

These robot has been initially tested to move from initial set point (x position=0 metre, y position=0 metre) to the end set point (x position=30 metre, y position=10 metre) by using selected fuzzy logic controller. The plots in Figure 4.5, 4.6 and 4.7 shows comparison of the simulation results obtained from the response of the robot with difference membership function of fuzzy logic controller. Figure 4.5 shows the angle of orientation of the robot as a function of time while Figure 4.6 shows the right and left velocity of the mobile robot as a function of time. Meanwhile, Figure 4.7 shows the position of the mobile robot with respect of the x and y positions of the frame.

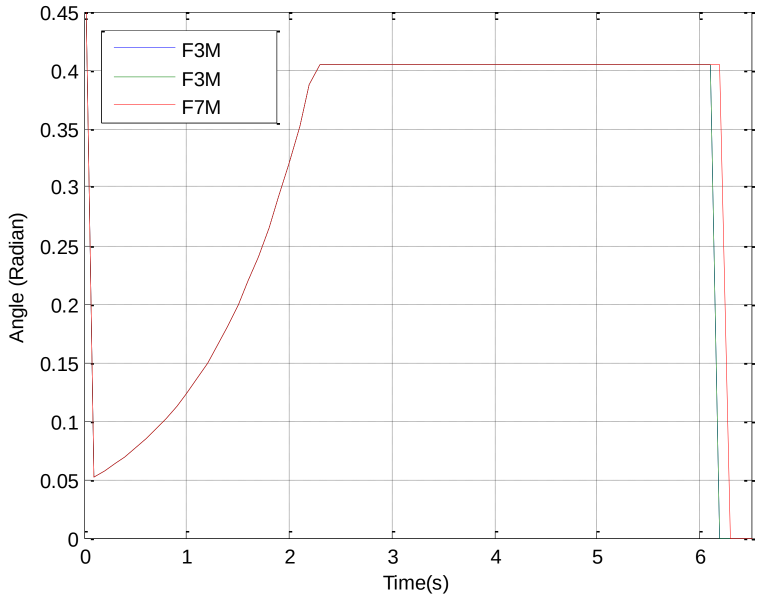

Figure 4.5: Angle of orientation fuzzy controller for 30x10y

It can be seen from Figure 4.5 that the change of angle occurs until it reaches steady state at 2.3 seconds for F3M, F5M and F7M controllers. The time taken for each target point at 6.4 seconds for F7M and 6.3 seconds for F3M and F5M controllers. Fuzzy logic with three membership function (F3M) and five membership function (F5M) controller for each target point gives better response in comparison with the fuzzy logic with seven membership function (F7M) controller of the differential drive wheel mobile robot.

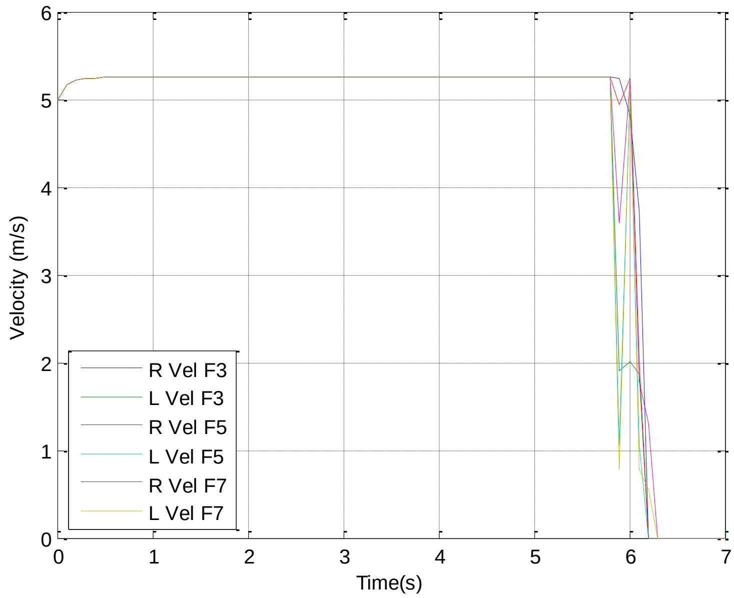

Figure 4.6: Right and Left Velocity of WMR fuzzy controller for 30x10y

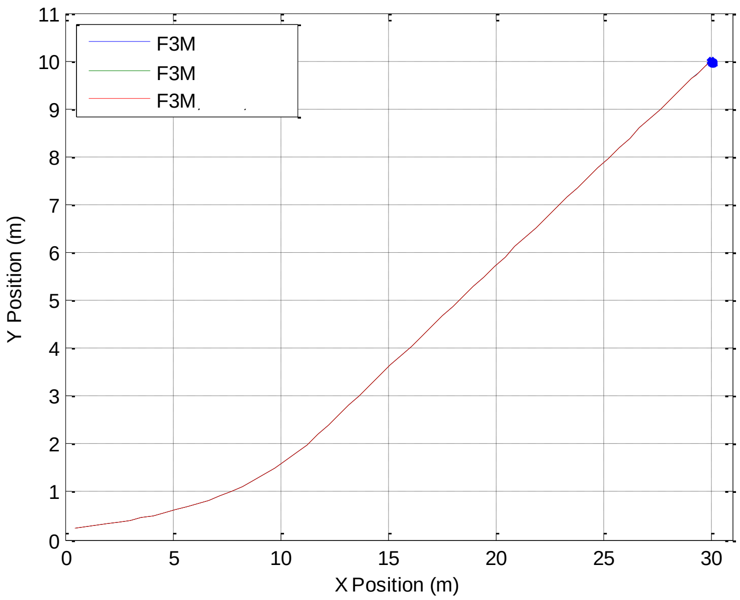

Figure 4.7:Position of WMR using fuzzy controller for 30x10y

Table 4.2 summarizes the performance evaluation of different fuzzy logic controller of differential drive wheel mobile robots. These differential drive wheel mobile robots has been initially tested to move from initial set point (x position=0 metre, y position=0 metre) to the end set point (x position=10 metre y position=10 metre), (x position=30 metre y position =10 metre) and (x position=60 metre y position=20 metre).

Table 4.2: Performance evaluation of difference Fuzzy Logiccontroller of WMR

Controller Type	**Target Point**		
	10x 10y	30x10y	60x20y
F3M	5.6s	6.3s	11.2s
F5M	5.7s	6.3s	11.2s
F7M	5.9s	6.4s	11.2s

Figure 4.8 shows the performance evaluation of different fuzzy logic controller of differential drive wheel mobile robots based on Table 4.2.

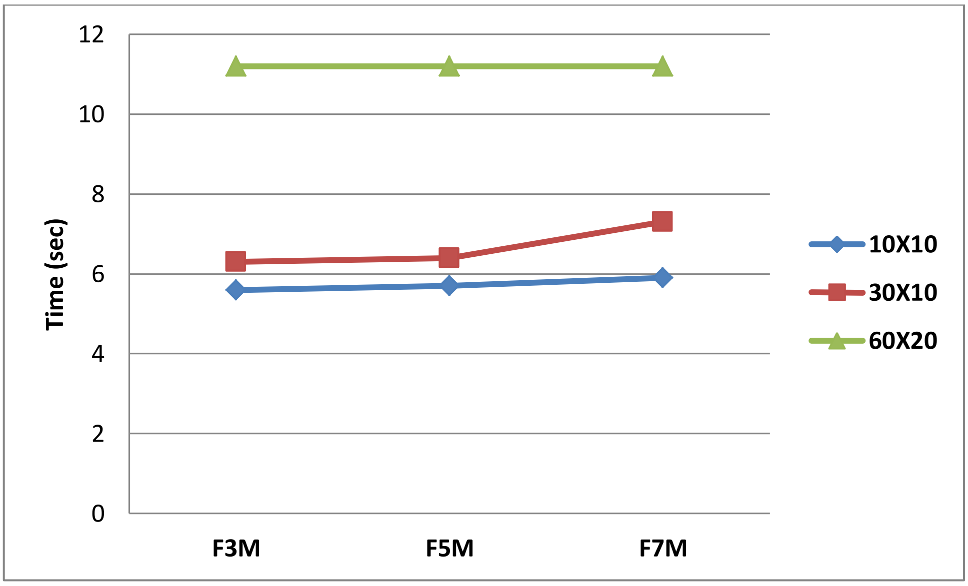

Figure 4.8:Performance evaluation of difference fuzzy controller of WMR

The resutls from Figure 4.8 shows that the Fuzzy with three membership function (F3M) controller for each target point gives better response with 5.6s, 6.3s and 11.2s for target of 10x 10y, 30x10y and 60x20y respectively, compared to fuzzy with five membership function (F5M) , 5.7s, 6.3s and 11.2s and fuzzy with seven membership function (F7M) with 5.9s, 6.4s and 11.2s .

4.1.3 Hybrid Controller

In this section, the results of hybrid controller simulation are divided into three section which are the hybrid controller fuzzy logic with proportional (P) controller in section 4.1.3.1, the hybrid controller fuzzy logic with proportional integral (PI) controller in section 4.1.3.2 and the hybrid controller Fuzzy logic with proportional integral derivative (PID) controller in section 4.1.3.3.

4.1.3.1 Fuzzy Logic with Proportional Controller (FP)

These mobile robot has been initially tested to move from initial set point (x position=0 metre, y position=0 metre) to the end set point (x position=10 metre y position=10 metre) by using hybrid controller selected fuzzy logic with proportional (FP) controller. The plot in Figure 4.9, 4.10 and 4.11shown below compares the simulation results obtained from the response of the fuzzy logic difference membership function with proportional controller of the robot. Figure 4.9 shows the angle of orientation of the robot as a function of time while Figure 4.10 shows the right and left velocity of the mobile robot as a function of time. Meanwhile, Figure 4.11 shows the position of mobile robot in respect with the x and y position of the frame.

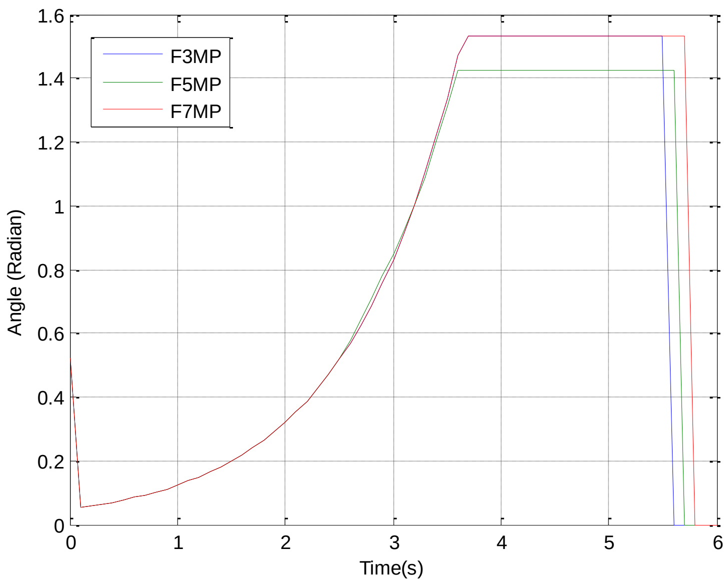

Figure 4.9:Angle of orientation FP controller for 10x10y

It can be seen from Figure 4.9 that the change of angle occurs until it reaches steady state at 3.7 seconds for F3MP, F7MP and 3.6 seconds for F7MP controller. The time taken for each target point is at 5.7 seconds for F3MP, 5.8 seconds for F5MP and 5.8 seconds for F7MP controller. Fuzzy logic three membership functions with proportional controller (F3MP) for each target point give better response compare fuzzy logic five membership functions with proportional controller (F5MP) and Fuzzy logic seven membership functions with proportional controller (F7MP) of differential drive wheel mobile robots.

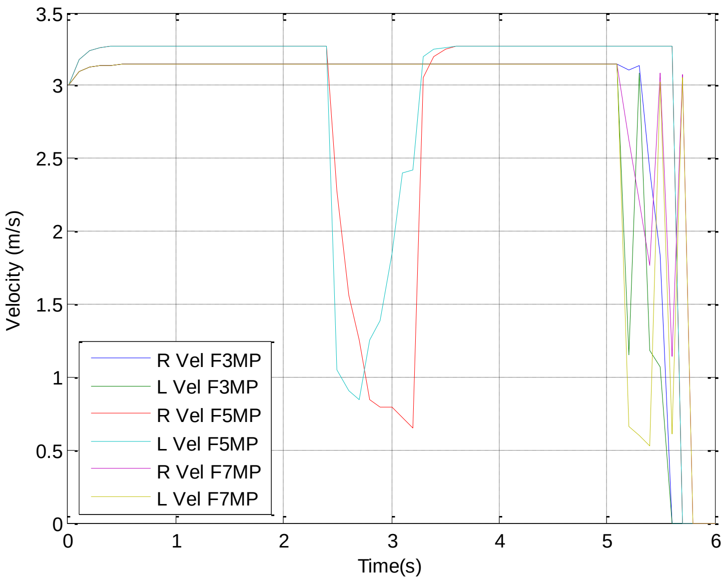

Figure 4.10:Right and Left Velocity of WMR FP controller for 10x10y

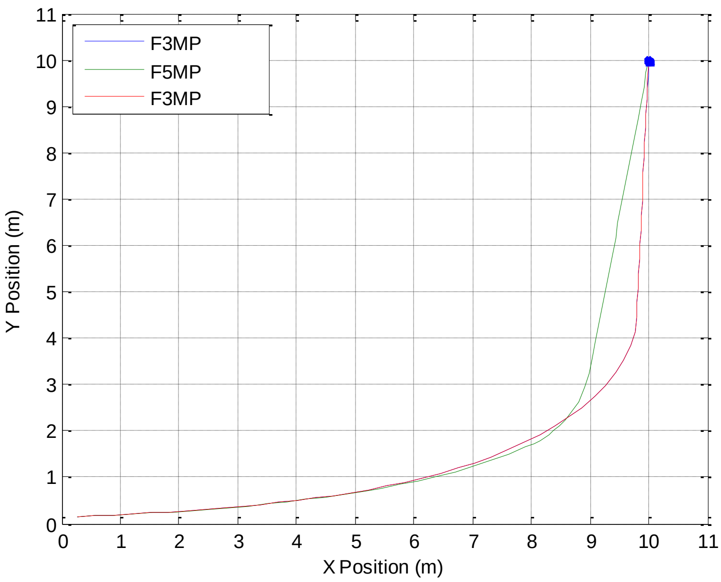

Figure 4.11:Position of WMR using FP controller for 10x10y

Table 4.3 summarizes the performance evaluation of hybrid controller with difference fuzzy logic membership function with proportional (FP) controller of differential drive wheel mobile robots (WMR).Three type of controllers are used which are fuzzy logic three membership function with proportional controller (F3MP),fuzzy logic five membership function with proportional controller (F5MP) and fuzzy logic seven membership function with proportional controller (F7MP).These differential drive wheel mobile robots (WMR) has been initially tested to move from initial set point (x position = 0 metre, y position = 0 metre) to the end set point (x position = 10 metre y position= 10 metre),(x position = 30 metre y position = 10metre) and (x position = 60 metre y position = 20 metre).

Table 4.3 : Performance evaluation of difference hybrid (FP) controller of WMR

Controller Type	**Target Point**		
	10x 10y	30x10y	60x20y
F3MP	5.7s Kp=1	5.5s Kp=5	10.8s Kp=2
F5MP	5.8s Kp=1	5.5s Kp=5	10.8s Kp=2
F7MP	5.9s Kp=1	5.7s Kp=5	10.8s Kp=2

Figure 4.12 shows the performance evaluation of difference hybrid (FP) controller of differential drive wheel mobile robots.

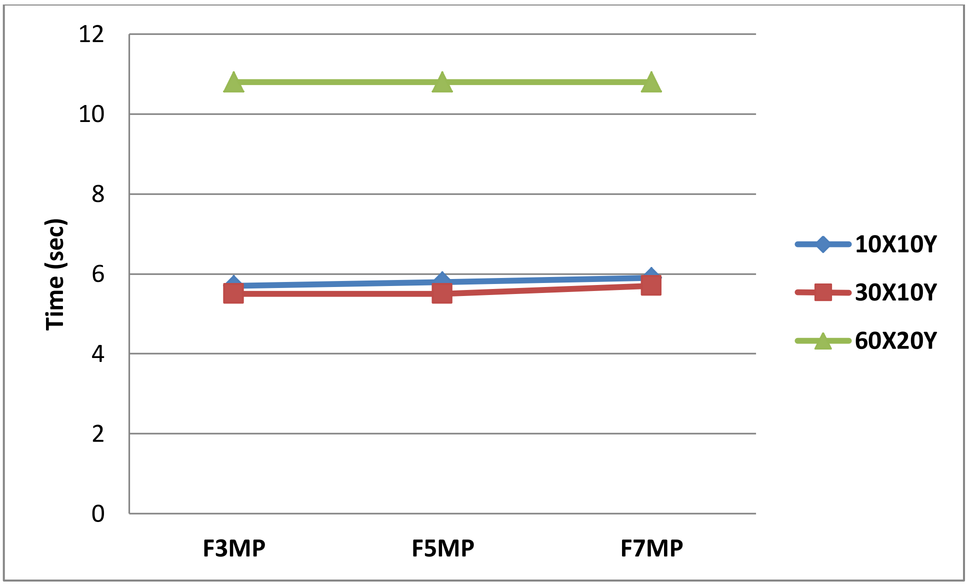

Figure 4.12:Performance evaluation of difference hybrid (FP) controller of WMR

The different type of fuzzy logic three membership function with proportional controller (F3MP) , fuzzy logic five membership function with proportional controller (F5MP) and fuzzy logic seven membership function with proportional controller (F7MP) and difference target point for differential drive wheel mobile robots give different values of time in seconds as seen from Figure 4.12. Fuzzy three membership function with proportional controller (F3MP) for each target point gives better response with 5.7s, 5.5s and 10.8s for target of 10x 10y, 30x10y and 60x20y respectively, compared to the fuzzy five membership function with proportional controller (F5MP) , 5.8s, 5.5s and 10.8s and Fuzzy seven membership function with proportional controller (F7MP) with 5.9s, 5.7s and 10.8s .

4.1.3.2 Fuzzy logic with proportional integral controller (FPI)

The robot has been initially tested to move from initial set point (x position=0 metre, y position=0 metre) to the end set point (x position=10 metre y position=10 metre) by using hybrid controller selected fuzzy logic with proportional integral (FPI) controller. The plot in Figures 4.13, 4.14 and 4.15 shown below compares the simulation results obtained from the response of the robot both with three membership function of fuzzy logic with proportional integral (FPI) controller. Figure 4.13 shows the angle of orientation of the robot while Figure 4.14 shows the right and left velocity of the mobile robot as a function of time. Meanwhile, Figure 4.15 shows the position of mobile robot with respect to the x and y positions of the frame.

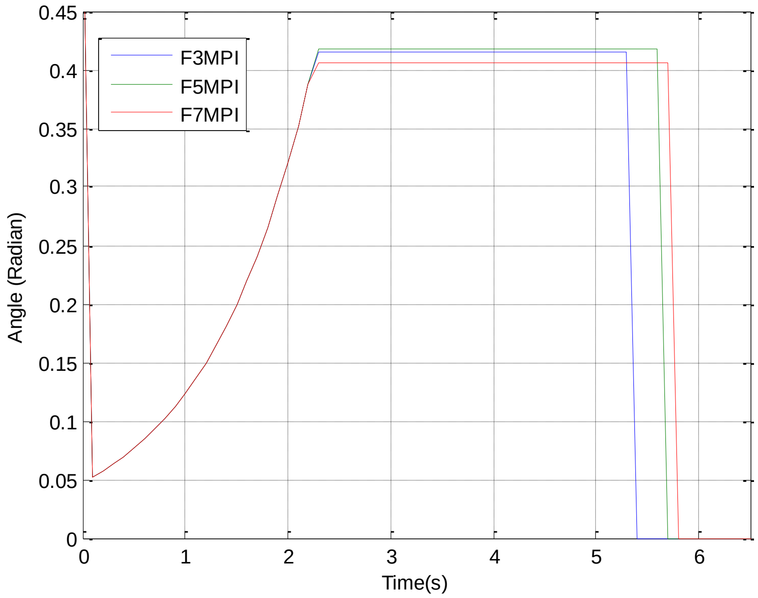

Figure 4.13:Angle of orientation FPI controller for 10x10y

From Figure 4.13, it can be seen that the change of angle occurs until it reaches steady state at 2.3 seconds for F3MPI, F5MPI and 2.25 seconds for F7MPI controller. The time taken for each target point is at 5.4 seconds for F3MPI, 5.8 seconds for F5MPI and 5.9 seconds for F7MPI controller. Fuzzy logic three membership functions with proportional integral controller (F3MPI) for each target point gives better response compared to the fuzzy logic five membership functions with proportional integral controller (F5MPI) and fuzzy logic seven membership functions with proportional integral controller (F7MPI) of differential drive wheel mobile robots.

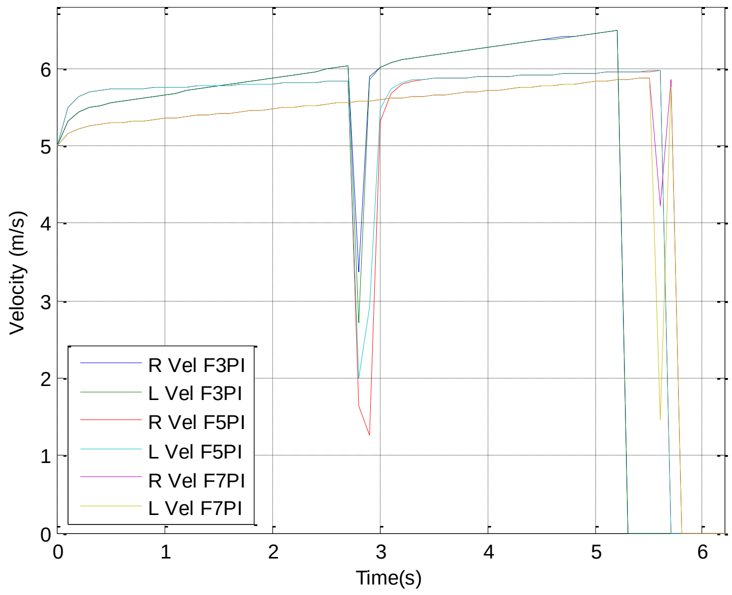

Figure 4.14:Right and left velocity of WMR FPI controller for 10x10y

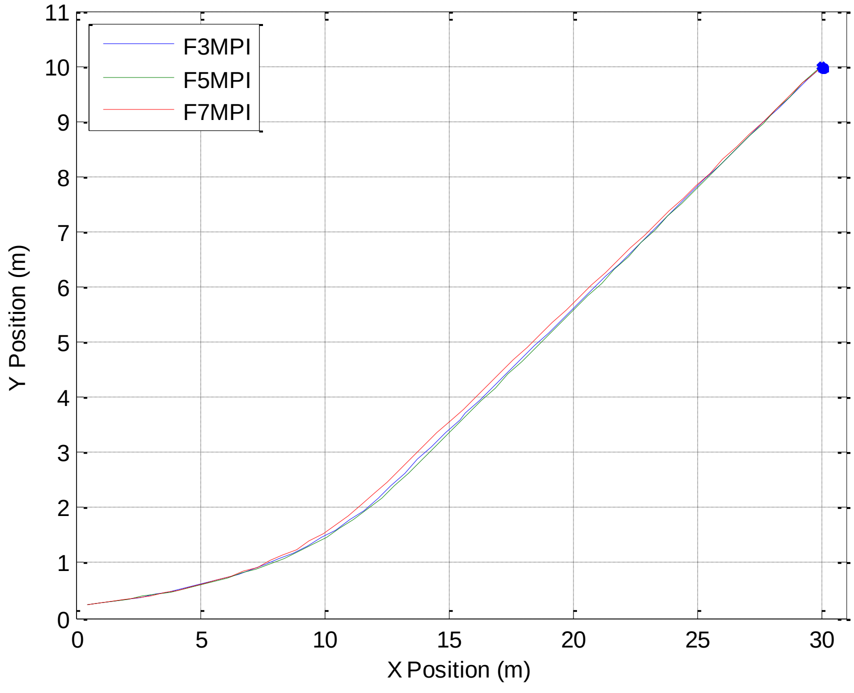

Figure 4.15:Position of WMR using FPI controller for 10x10y

Table 4.4 summarizes the performance evaluation of hybrid controller with difference fuzzy logic (FL) membership function and proportional integral (PI) controller of differential drive wheel mobile robots (WMR). These differential drive wheel mobile robots (WMR) has been initially tested to move from initial set point (x position =0 metre,y position = 0 metre) to the end set point (x position = 10 metre y position =10 metre), (x position = 30 metre y position = 10 metre) and (x position = 60 metre y position = 20 metre).

Table 4.4:Performance evaluation of difference hybrid (FPI) controller of WMR

Controller Type	Target Point		
	10x 10y	30x10y	60x20y
F3MPI	5.5s Kp=1,Ki=0.5625	5.4s Kp=2,Ki=0.5625	11.0s Kp=0.4,Ki=0.3
F5MPI	5.5s Kp=1,Ki=0.5625	5.8s Kp=3.3,Ki=0.0825	11.0s Kp=0.5,Ki=0.3
F7MPI	5.7s Kp=0.8,Ki=0.3	5.9s Kp=1,Ki=0.5625	11.12s Kp=0.5,Ki=0.3

Figure 4.16 shows the performance evaluation of different fuzzy logic and proportional Intergral controller of differential drive wheel mobile robots.

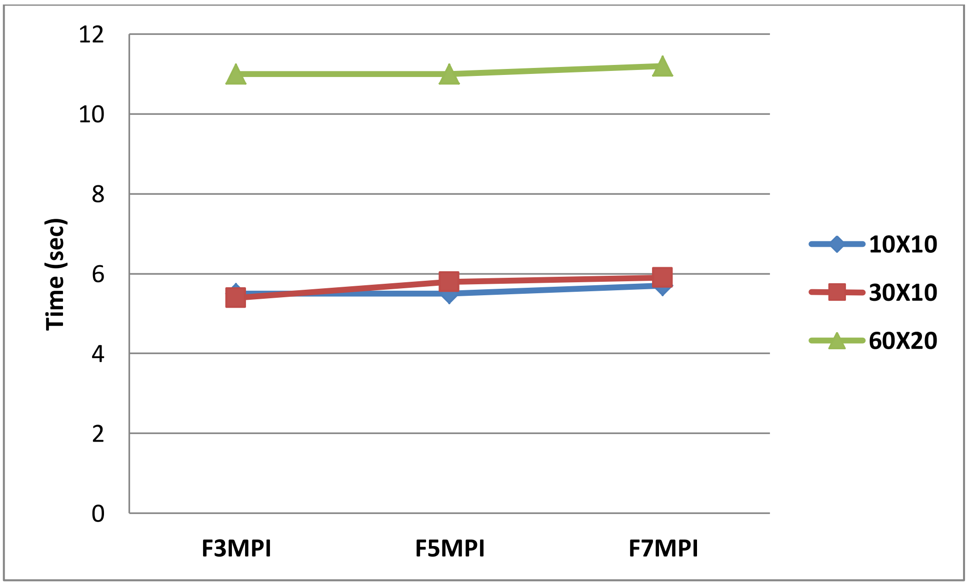

Figure 4.16:Performance evaluation of different hybrid (FPI) controller of WMR

It is also clearly shown that a model simulated using different type of fuzzy logic control with three membership function proportional integral controller (F3MPI), five membership function proportional integral controller (F5MPI) and seven membership function proportional integral controller (F7MPI) with different target point for differential drive wheel mobile robots gives different values time in seconds. Fuzzy logic control with three membership function proportional integral controller (F3MPI) for each target point gives better response with 5.5s, 5.4s and 11s for target of 10x 10y, 30x10y and 60x20y respectively, as compared to fuzzy logic control with five membership function proportional integral controller (F5MPI), 5.5s, 5.8s and 11s and fuzzy logic control with seven membership function proportional integral controller (F7MPI) with 5.7s, 5.9s and 11.12s..

4.1.3.3 Fuzzy logic with proportional integral derivative controller (FPID)

These robot has been initially tested to move from initial set point (x position=0 metre, y position=0 metre) to the end set point (x position=30 metre y position=10 metre) by using hybrid controller selected fuzzy logic and proportional integral derivative (FPID) controller. The plot in Figure 4.17, 4.18 and 4.19 shows the simulation results obtained from the response of the robot with three membership function of fuzzy logic and proportional integral derivative (F3MPID) controller. Figure 4.17 shows the angle of orientation of the robot and Figure 4.18 shows the right and left velocity of the mobile robot as a function of time. Figure 4.19 shows the position of mobile robot with respect to the x and y position of the frame.

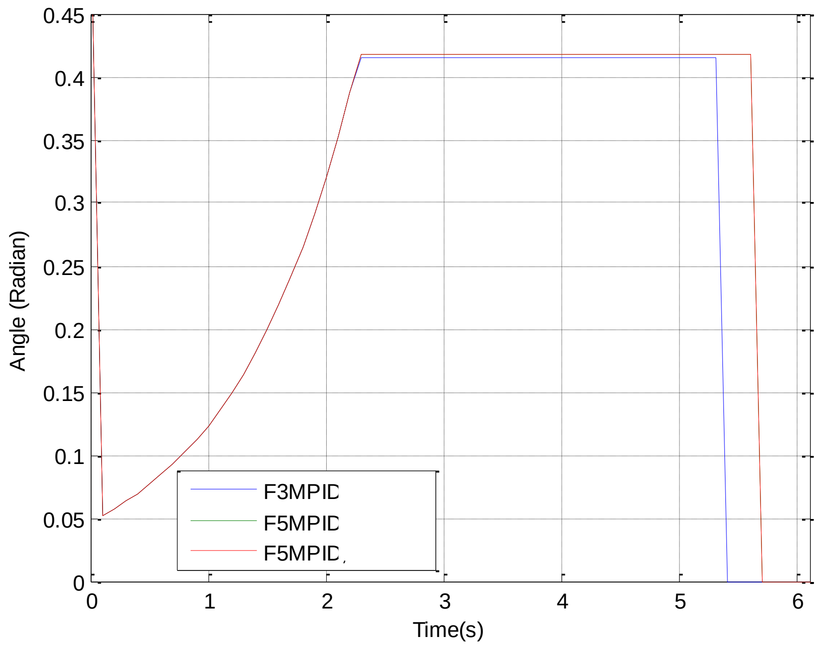

Figure 4.17:Angle of orientation FPID controller for 30x10y

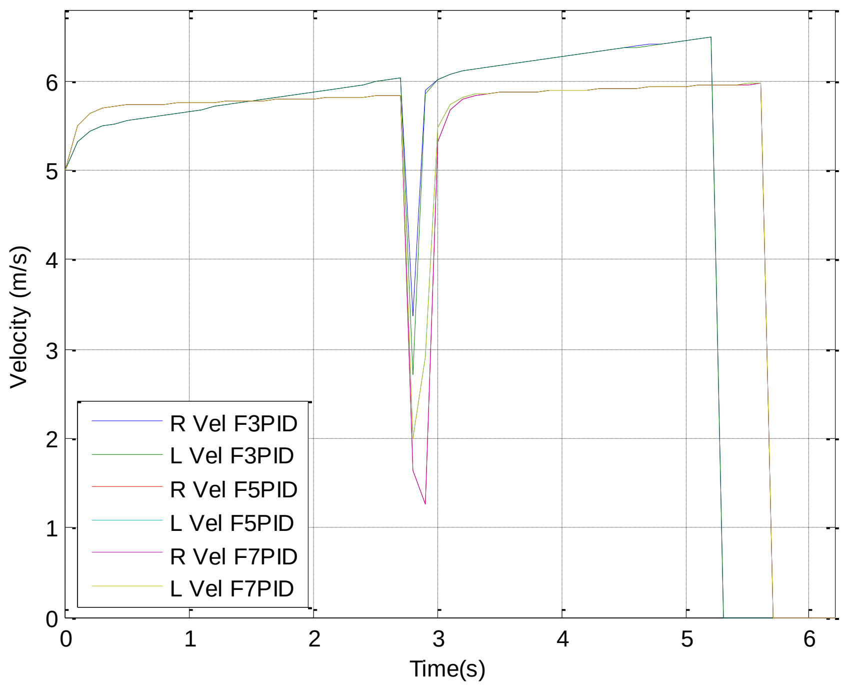

Figure 4.18:Right and left velocity of WMR FPID controller for 30x10y

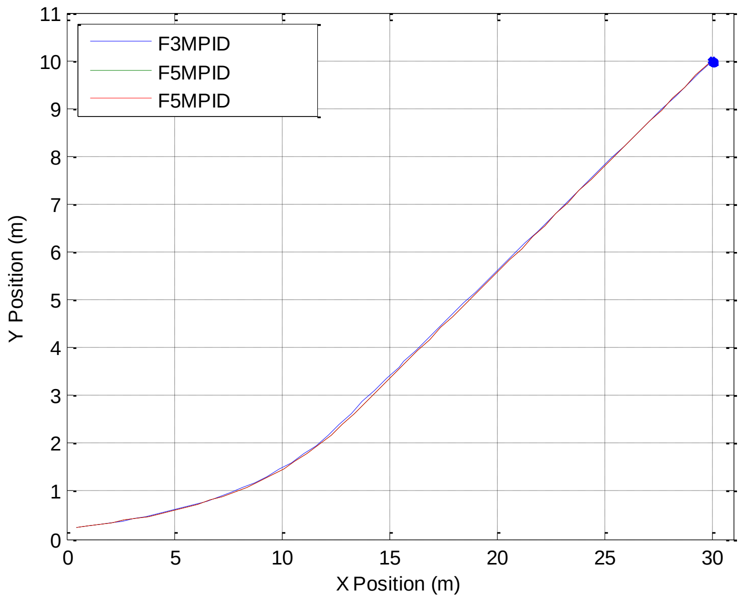

Figure 4.19:Position of WMR using FPID controller for 30x10y

Table 4.5 summarizes the performance evaluation of hybrid controller with different fuzzy logic membership function with proportional integral derivative (FPID) controller of differential drive wheel mobile robots (WMR). Figure 4.20 shows the performance evaluation of difference hybrid (fuzzy and proportional integral derivative) controller based on Table 4.5. These differential drive wheel mobile robots (WMR) has been initially tested to move from initial set point (x position=0 metre, y position=0 metre) to the end set point (x position=10 metre y position=10 metre), (x position=30 metre y position=10 metre) and (x position=60 meter y position=20 metre).

Table 4.5:Performance evaluation of different hybrid (FPID) controller

Controller Type	Target Point		
	10x 10y	30x10y	60x20y
F3MPID	5.6s Kp=1,Ki=0.1375 Kd=1.15	5.4s Kp=0.7,Ki=1 Kd=0.5	11.0s Kp=0.5,Ki=0.3 Kd=0.1
F5MPID	5.6s Kp=1,Ki=0.1 Kd=0.5	5.8s Kp=1.2,Ki=0.5375 Kd=0.5	11.0s Kp=0.5,Ki=0.3 Kd=0.1
F7MPID	5.6s Kp=1,Ki=0.1 Kd=0.5	5.8s Kp=1,Ki=0.5375 Kd=1	11.0s Kp=0.5,Ki=0.3 Kd=0.1

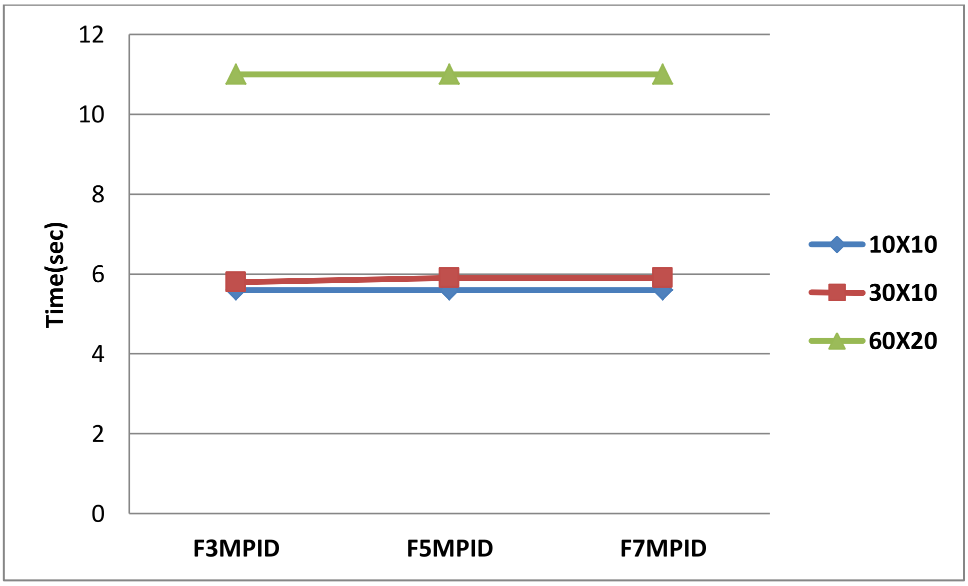

Figure 4.20:Performance evaluation of difference hybrid (FPID) controller

From Figure 4.20, the model for differential drive WMRs simulated using different types of fuzzy logic with various membership functions with proportional integral derivative controllers such as F3MPID, F5MPID and F7MPID for different target point give different values for time (in seconds). Fuzzy with three membership function proportional integral derivative controller (F3MPID) for each target point gives better responsewith 5.6s, 5.4s and 11s for target of 10x 10y, 30x10y and 60x20y respectively, compared to fuzzy logic five membership function with proportional integral derivative controller (F5MPID), 5.6s, 5.4s and 11s and fuzzy logic seven membership function with proportional integral derivative controller (F7MPID) with 5.6s, 5.4s and 11s.

The justification for the use of the kinematic and the motor models of the WMR is provided here before discussing the overall performance of the controller. The kinematic model of WMR (the final Equation 3.9) and the motor model based on Maxon EC-4 pole 30 DC brushless motor (Equation 3.28) have been used as the plant in all the simulation work carrried out throughout the research are clearly shown in all the simulation block diagrams (Figure 3.9 for conventional controller, Figure 3.12 for fuzzy logi controller and Figures 3.21, 3.23, 3.25 for the hybrid controller).

Table 4.6 shows the overall performance evaluation of different controller of differential drive wheel mobile robots.

Table 4.6: Performance evaluation of difference controller of WMR

Controller Type	Target Point		
	10x10y	30x10y	60x20y
P	5.7s	9.4s	18.2s
PI	5.6s	8.9s	16.5s
PID	5.6s	7.8s	14.7s
F3M	5.6s	6.3s	11.2s
F5M	5.7s	6.3s	11.2s
F7M	5.9s	6.4s	11.2s
F3MP	5.7s	5.5s	10.8s
F5MP	5.8s	5.5s	10.8s
F7MP	5.9s	5.7s	10.8s
F3MPI	5.5s	5.4s	11.0s
F5MPI	5.5s	5.8s	11.0s
F7MPI	5.7s	5.9s	11.12s
F3MPID	5.6s	5.4s	11.0s
F5MPID	5.6s	5.8s	11.0s
F7MPID	5.6s	5.8s	11.0s

It can be summarized from Table 4.6 that for conventional controller, the proportional integral derivation (PID) controller for each target point gives better response with

5.6s, 7.8s and 14.7s for target of 10x 10y, 30x10y and 60x20y respectively, compared to proportional integral (PI), 5.6s, 8.9s and 16.5s and proportional (P) with 5.7s, 9.4s and 18.2s .For fuzzy controller, fuzzy with three membership function (F3M) controller for each target point gives better response with 5.6s, 6.3s and 11.2s for target of 10x 10y,30x 10y and 60x 20y respectively, compared to fuzzy with five membership function (F5M) ,5.7s, 6.3s and 11.2s and fuzzy with seven membership function (F7M) with 5.9s, 6.4s and 11.2s . For hybrid controller, the fuzzy logic control with three membership function proportional integral controller (F3MPI) for each target point gives better response with 5.5s, 5.4s and 11s for target of 10x 10y,30x10y and 60x20y respectively, compared to fuzzy logic control with five membership function proportional integral controller (F5MPI), 5.5s, 5.8s and 11s and fuzzy logic control with seven membership function proportional integral controller (F7MPI) with 5.7s, 5.9s and 11.12s. Overall, on average the fuzzy logic control with three membership function proportional integral controller (F3MPI) gives the best performance.

4.2 Path planning optimization

As discussed in section 2.2.3, the path planning problem is to find a optimal path for which there is no collision with obstacles when the mobile robot moves from a starting point to a goal point under the given environmental information. It is used essentially to find the shortest path with minimal time for a different number of obstacles. In this work the ant colony optimization (ACO) algorithm was used to optimise the path planning for WMR, the starting point is (x position=1, y position=1) and the destination point is (x position =20, y position 20). Varying number of obstacles beginning from number 1 to 10 were used in this study. The obstacles can be located at any place on the map except at the starting point and at the destination point.

The experiments are done with different number of obstacles. The circular objects represent the obstacles. The experiments are conducted using MATLAB. The plots in Figures 4.21 and 4.22 show the simulation results of the path planning optimization with one obstacle the ant colony optimization (ACO) algorithm.

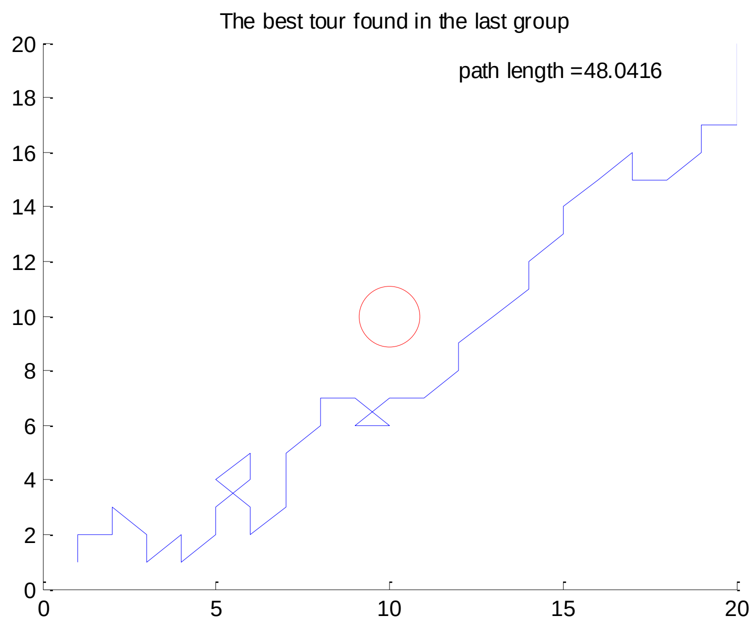

Figure 4.21:The best tour found at the 1 obstacle

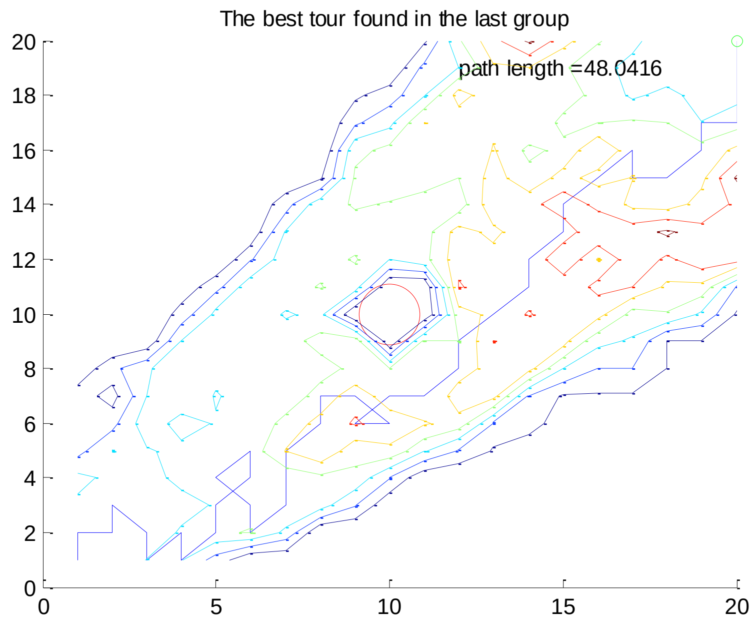

Figure 4.22:Deposited pheromone with 1 obstacles in 2D contour map

The plots in Figures 4.23 and 4.24 show the simulation results of the path planning optimization with two obstacle the ant colony optimization (ACO) algorithm.

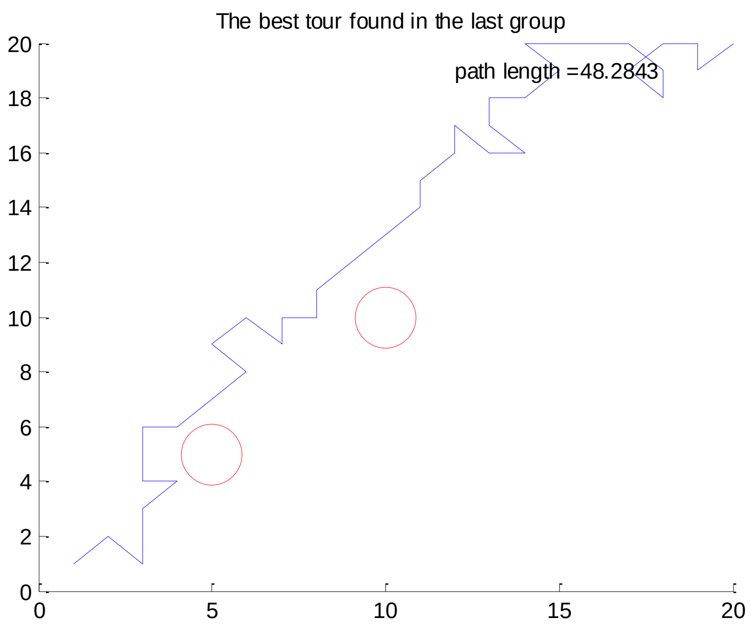

Figure 4.23:The best tour found at the 2 obstacle

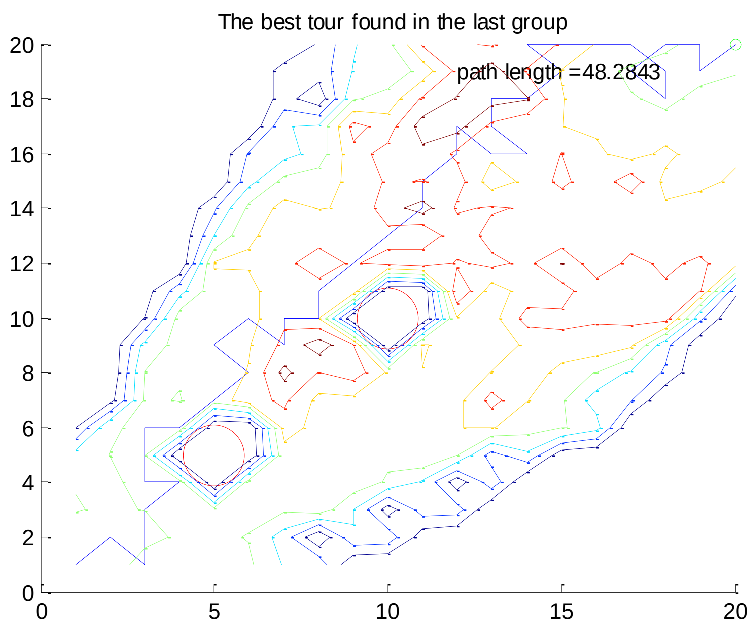

Figure 4.24:Deposited pheromone with 2 obstacles in 2D contour

The plots in Figures 4.25 and 4.26 show the simulation results of the path planning optimization with three obstacle the ant colony optimization (ACO) algorithm.

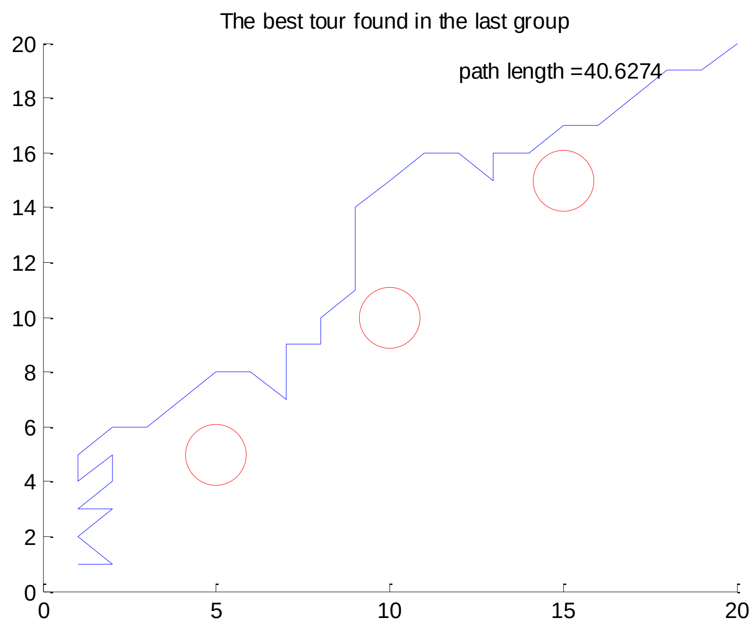

Figure 4.25:The best tour found at the 3 obstacle

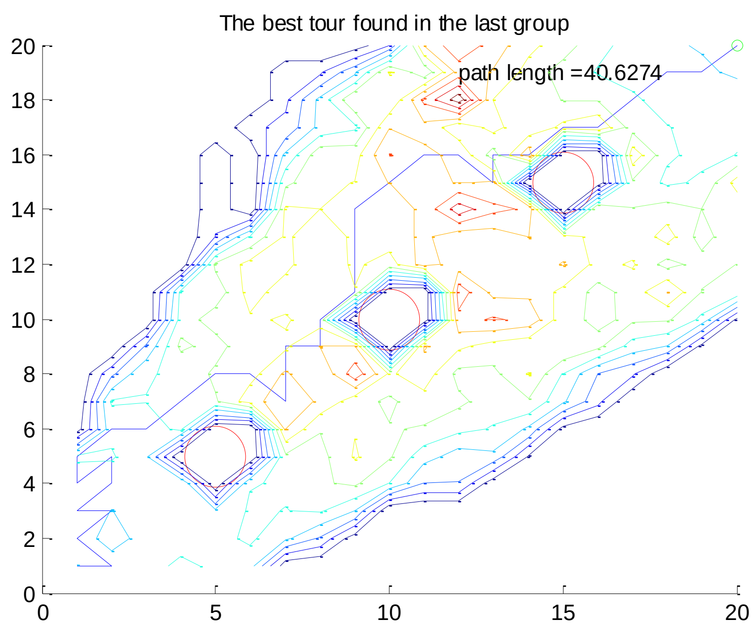

Figure 4.26:Deposited pheromone with 3 obstacles in 2D contour

The plots in Figures 4.27 and 4.28 show the simulation results of the path planning optimization with four obstacle the ant colony optimization (ACO) algorithm.

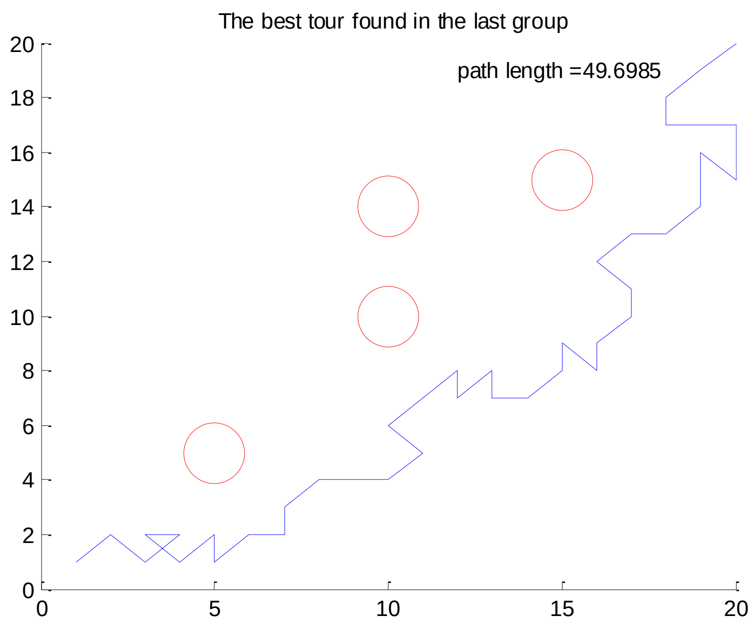

Figure 4.27:The best tour found at the 4 obstacle

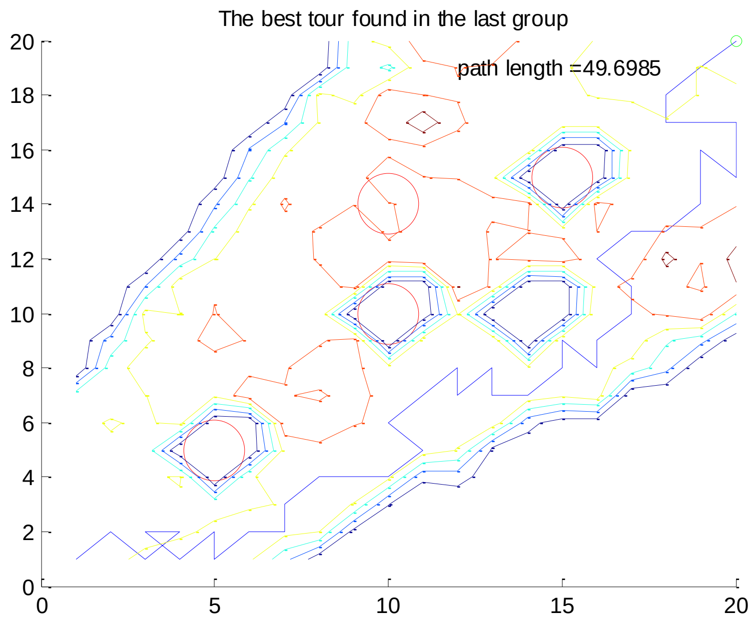

Figure 4.28:Deposited pheromone with 4 obstacles in 2D contour

The plots in Figures 4.29 and 4.30 show the simulation results of the path planning optimization with five obstacle the ant colony optimization (ACO) algorithm.

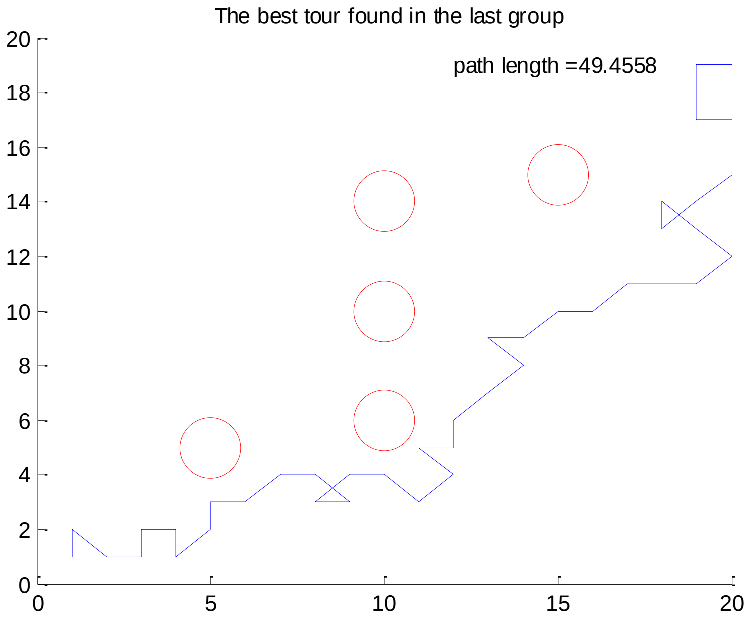

Figure 4.29:The best tour found at the 5 obstacle

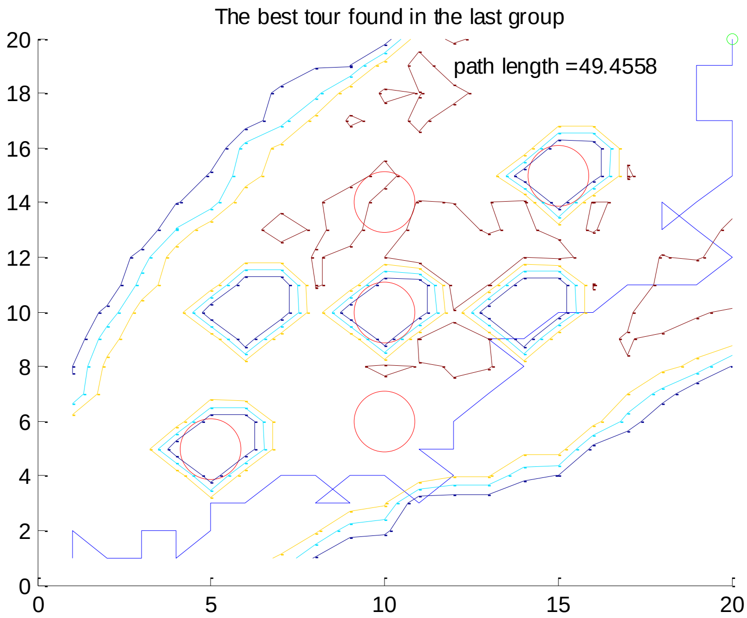

Figure 4.30:Deposited pheromone with 5 obstacles in 2D contour

The plots in Figures 4.31 and 4.32 show the simulation results of the path planning optimization with six obstacle the ant colony optimization (ACO) algorithm.

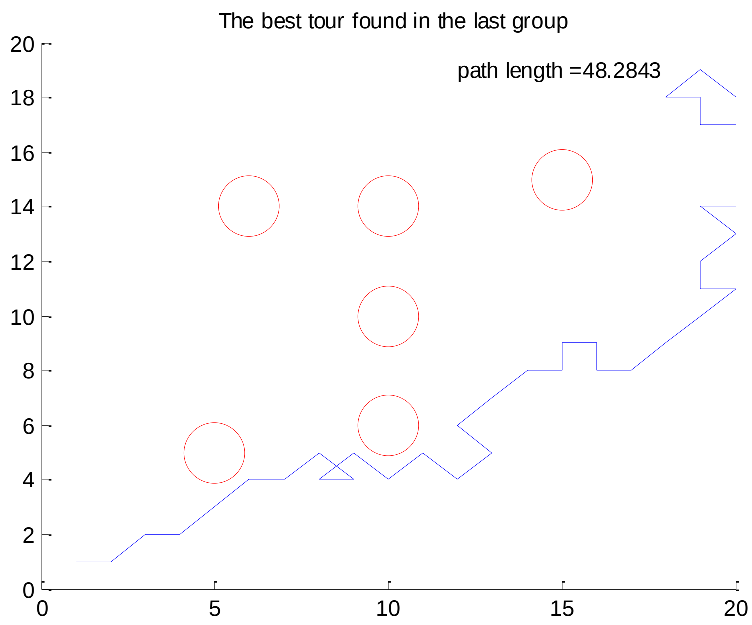

Figure 4.31:The best tour found at the 6 obstacle

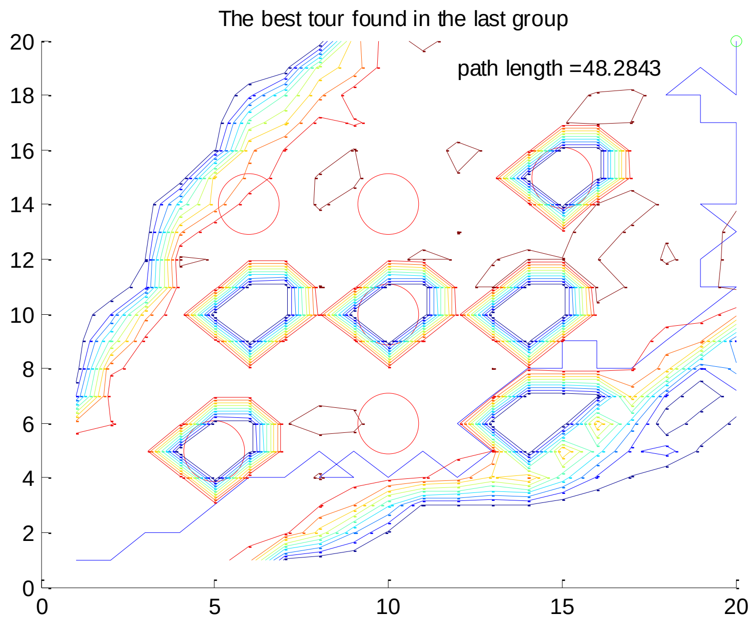

Figure 4.32:Deposited pheromone with 6 obstacles in 2D contour

The plots in Figures 4.33 and 4.34 show the simulation results of the path planning optimization with seven obstacle the ant colony optimization (ACO) algorithm.

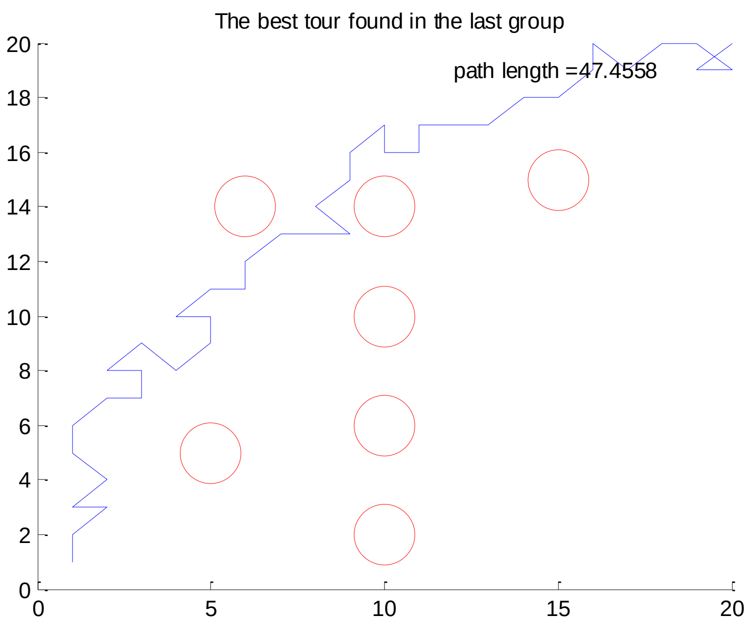

Figure 4.33:The best tour found at the 7 obstacle

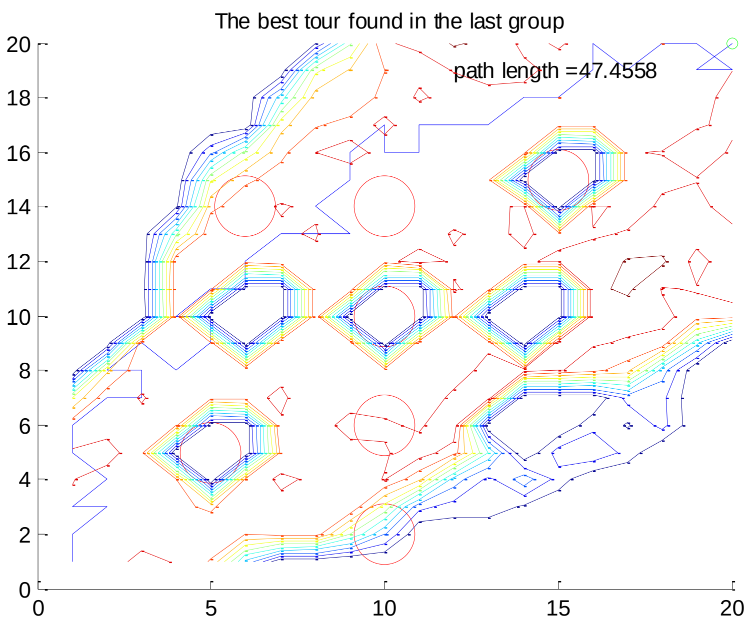

Figure 4.34:Deposited pheromone with 7 obstacles in 2D contour

The plots in Figures 4.35 and 4.36 show the simulation results of the path planning optimization with eight obstacle the ant colony optimization (ACO) algorithm.

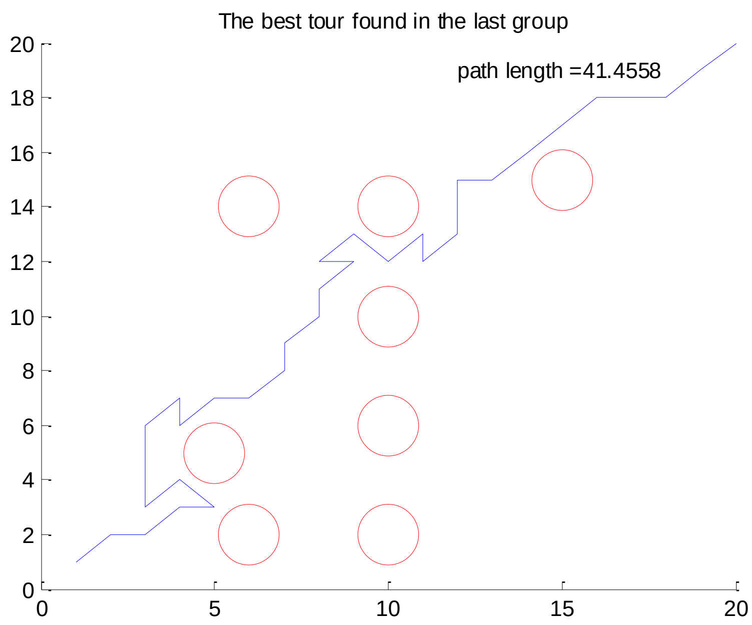

Figure 4.35:The best tour found at the 8 obstacle

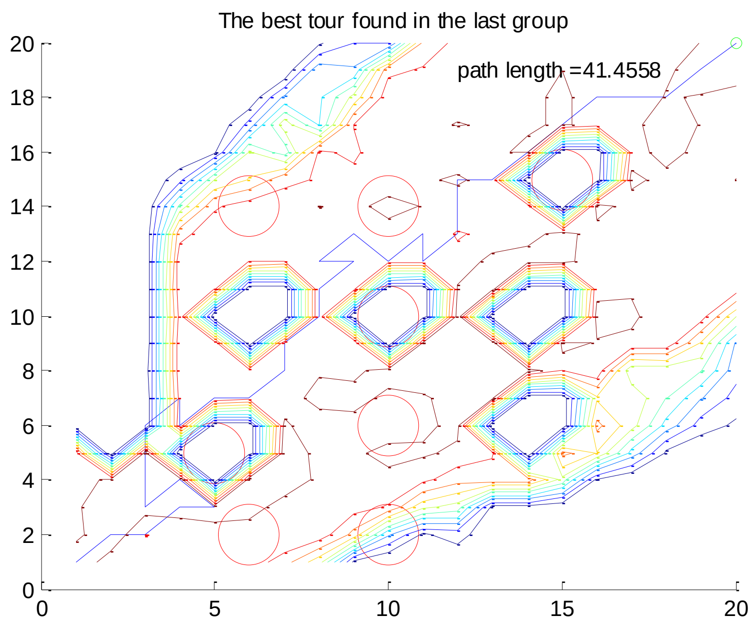

Figure 4.36:Deposited pheromone with 8 obstacles in 2D contour

The plots in Figures 4.37 and 4.38 show the simulation results of the path planning optimization with nine obstacle the ant colony optimization (ACO) algorithm.

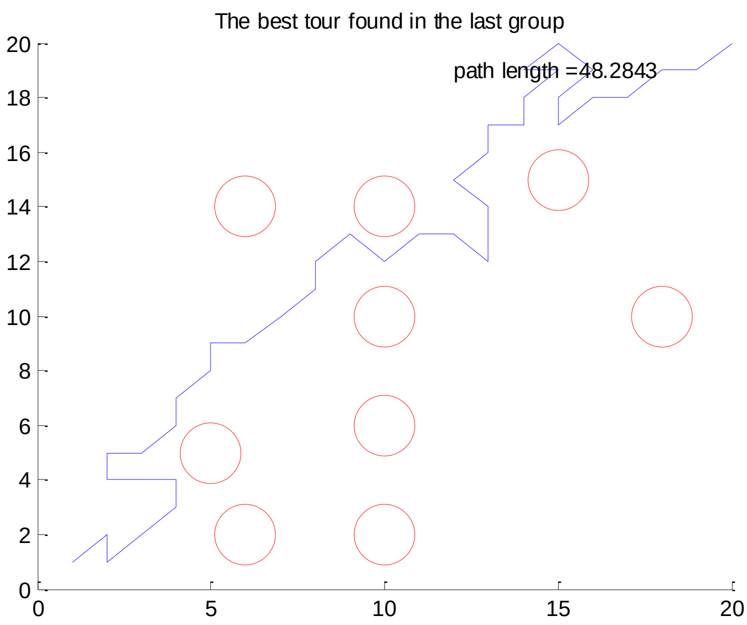

Figure 4.37:The best tour found at the 9 obstacle

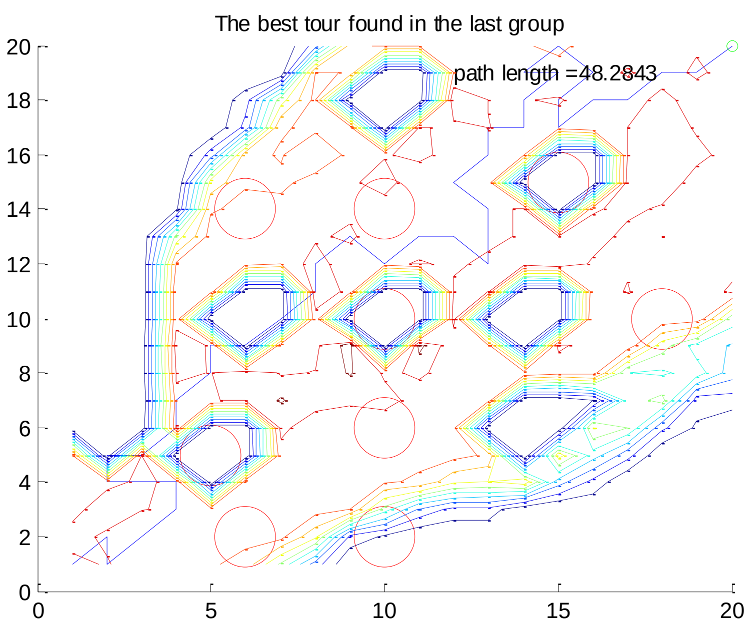

Figure 4.38:Deposited pheromone with 9 obstacles in 2D contour

The plots in Figures 4.39 and 4.40 show the simulation results of the path planning optimization with nine obstacle the ant colony optimization (ACO) algorithm.

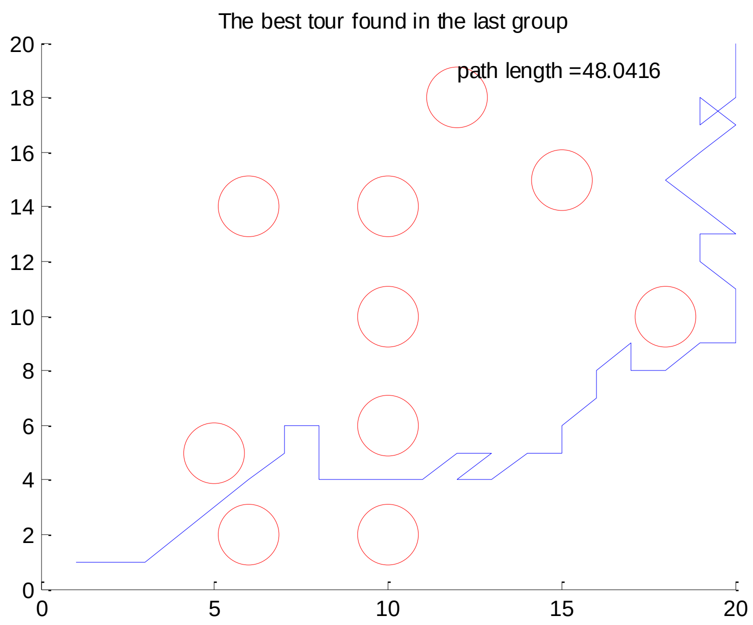

Figure 4.39:The best tour found at the 10 obstacle

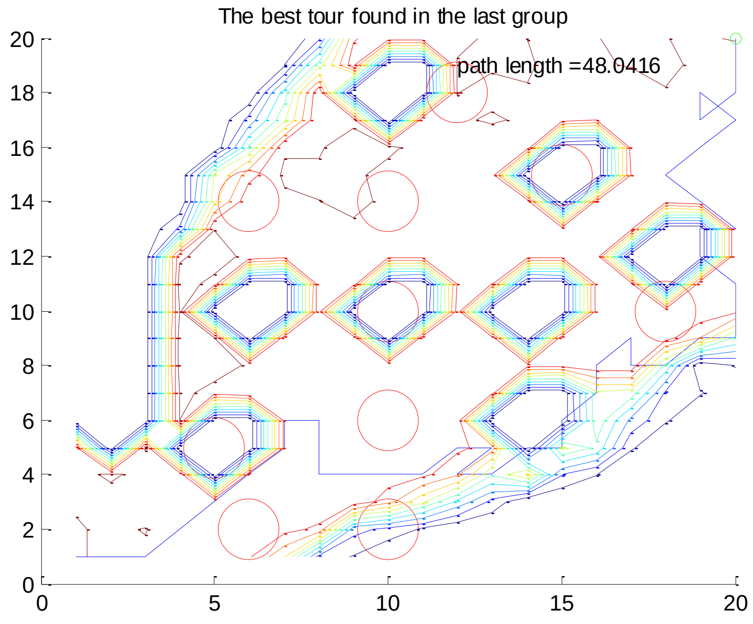

Figure 4.40:Deposited pheromone with 10 obstacles in 2D contour

The path length with five different simulations with different number of obstacles using ACO is given in Table 4.7 and the average path length of five different simulations with different number of obstacle of is given in Figure 4.41.

Table 4.7:Path length with five simulations and different number of obstacle of ACO

Number of obstacle	Path length (metre)				
	Sim1	Sim2	Sim3	Sim4	Sim5
1 obstacle	48.0416	52.2843	46.8701	53.6985	50.5269
2 obstacle	48.2843	49.1129	50.8701	46.0416	45.4558
3 obstacle	40.6274	48.0416	51.1127	50.8701	48.5269
4 obstacle	49.6985	44.2843	50.2843	48.0416	43.2132
5 obstacle	49.4558	46.0416	49.4558	47.1127	45.4558
6 obstacle	48.2843	47.6274	45.8701	46.8701	45.4559
7 obstacle	47.4558	41.4558	44.6274	47.4558	52.0416
8 obstacle	41.4558	46.6274	47.6985	41.6985	42.2843
9 obstacle	48.2843	50.6274	45.0416	44.2843	47.6985
10 obstacle	48.0416	40.0416	49.2843	45.6985	44.6274

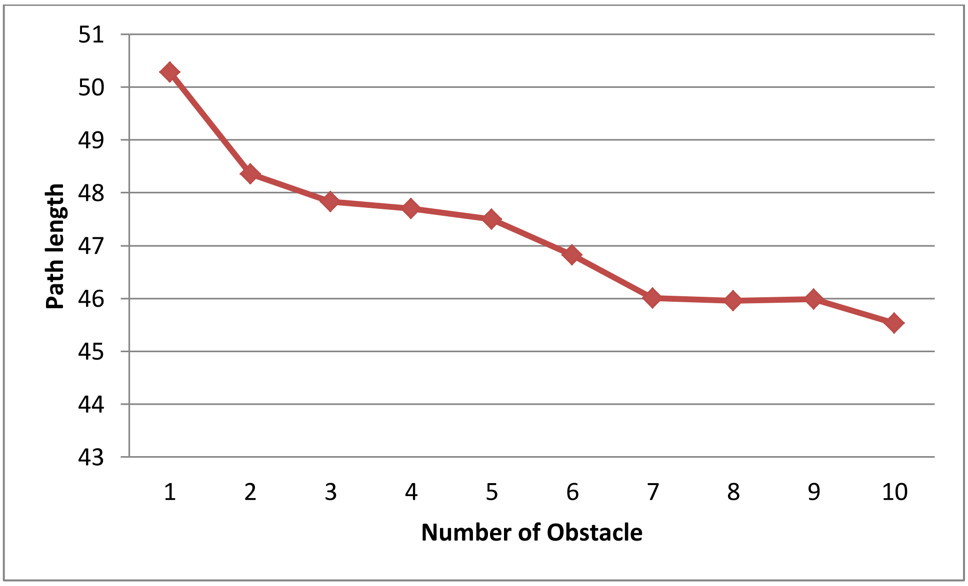

Figure 4.41:Average path length five simulations replications with different number of obstacle of ACO

The results of Table 4.7 reveal that the ACO approaches used in this research are able to generate competitive solutions for the path planning with a minimum path length of 48.0416 metre and a maximum of 49.6985 metre for the first simulation. A similar trend can be found for other four different simulations as well. As for the path length versus the number of obstacles, it displayed a random decreasing nonlinear trend.

Table 4.8 shows the actual result CPU time with five different simulations with different number of obstacles using ACO

Table 4.8:shows the actual results CPU time from the Matlab simulation.

Number of obstacle	CPU time				
	Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
1	aco_code_1ob	1	1995.266 s	248.696 s	
2	aco_code_2ob	1	1990.418 s	246.433 s	
3	aco_code_3ob	1	2007.809 s	250.175 s	
4	aco_code_4ob	1	1982.892 s	244.732 s	
5	aco_code_5ob	1	1986.088 s	249.871 s	
6	aco_code_6ob	1	1981.904 s	247.335 s	
7	aco_code_7ob	1	1982.404 s	242.895 s	
8	aco_code_8ob	1	1974.000 s	246.551 s	

	Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
9	aco_code_9ob	1	1970.841 s	243.079 s	
10	aco_code_10ob	1	1964.840 s	245.509 s	

Table 4.9 shows the summary of the CPU Time with five different simulations with different number of obstacles using ACO and average CPU time with five different simulations with different number of obstacle is given in Figure 4.42.

Table 4.9:CPU Time with five simulation reptications and different number of obstacle of ACO

Number of obstacle	CPU time (sec)				
	Sim1	Sim2	Sim3	Sim4	Sim5
1 obstacle	2017	1996	2001	1999	1999
2 obstacle	2010	1991	1989	1993	1992
3 obstacle	2009	1990	1984	2289	1980
4 obstacle	1996	1989	1983	1983	1976
5 obstacle	1996	1986	2003	1979	1980
6 obstacle	1979	1982	1978	1969	1972
7 obstacle	1982	1983	1971	1974	1972
8 obstacle	1977	1974	1970	1964	1969
9 obstacle	1979	1971	1964	1957	1957
10 obstacle	1963	1965	1970	1948	1952

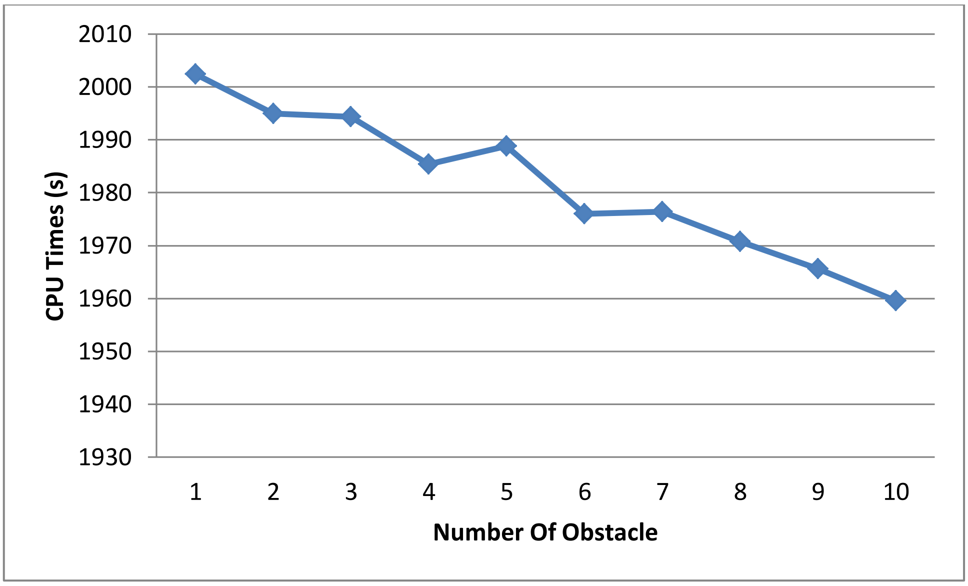

Figure 4.42:Average CPU time with five simulation replications different number of obstacle of ACO

The results of Table 4.8 shows that lowest CPU time 1996 for 1 obstacle, 1989 for 2 obstacles, 1980 for 3 obstacles, 1976 for 4 obstacles, 1979 for 5 obstacles, 1969 for 6 obstacles, 1971 for 7 obstacles, 1964 for 8 obstacles, 1957 for 9 obstacles and 1948 for 10 obstacles respectively. As for the average CPU time with five times simulation different number of obstacles, it can be observed that the CPU time decreases as the number of obstacles increase. This could be attributed to the decrease of path length as discussed earlier.

Table 4.10 shows the path length with ten times simulation and different number of obstacles using ACO and Figure 4.43 shows the average path length for ten times simulation with different number of obstacles.

Table 4.10:Path length with ten simulation replications and different number of obstacle of ACO

Path length (metre)	Number of obstacle		
	1 obstacle	5 obstacle	10 obstacle
Simulation 1	48.0416	49.4558	48.0416
Simulation 2	52.2843	46.0416	40.0416
Simulation 3	46.8701	39.4558	50.2843
Simulation 4	53.6985	47.1127	45.6985
Simulation 5	50.5269	45.4559	44.6274
Simulation 6	52.7696	45.6985	45.6985
Simulation 7	45.4558	52.5269	44.6274
Simulation 8	58.8701	46.0416	47.1127
Simulation 9	55.1127	50.5269	46.8701
Simulation 10	47.3553	45.6985	47.3553

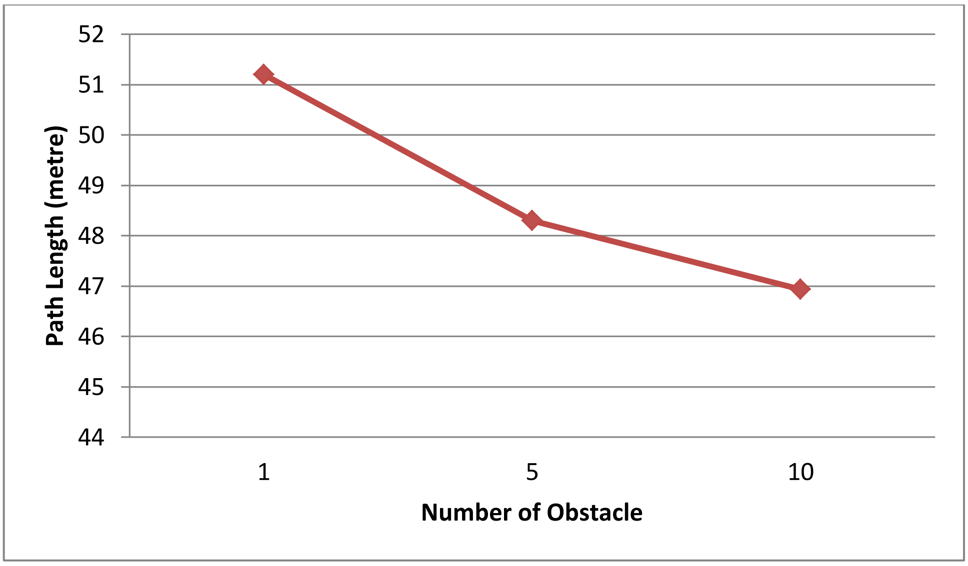

Figure 4.43:Average path length ten simulation replications with different number of obstacle of ACO

The results of Table 4.9 shows that path length of the path planning varies randomly for 1,5 and 10 obstacles with 48.0416, 49.4558 and 48.0416 respectively for simulation 1. Similar random trend was observed for simulations 2 to 10 as well. As for the path length versus the number of obstacles, the trend observed is that the path seems to be decreasing as the obstacles are increased.

Table 4.11 shows CPU time with ten simulation replications with different number of obstacle of ACO and Figure 4.44 shows the average CPU time with ten times simulation for different number of obstacles.

Table 4.11:CPU Time with ten simulation replications and different Number of Obstacle of ACO

CPU time (sec)	Number of obstacle		
	1 obstacle	5 obstacle	10 obstacle
Simulation 1	2017	1996	1963
Simulation 2	1996	1986	1965
Simulation 3	2001	2003	1970
Simulation 4	1999	1979	1948
Simulation 5	2012	1980	1952
Simulation 6	2003	1979	1953
Simulation 7	2007	1986	1967
Simulation 8	2010	1982	1962
Simulation 9	2008	1980	1964
Simulation 10	2005	1998	1964

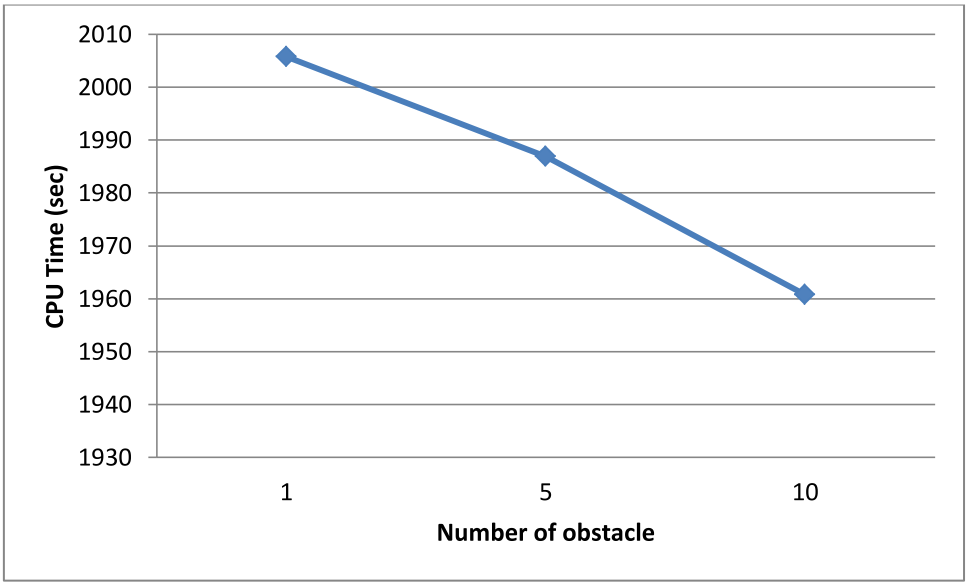

Figure 4.44: Average CPU time with ten simulation replications different number of obstacle of ACO

The results of Table 4.11 shows that CPU time of the path planning decreases for 1, 5 and 10 obstacles with 2017, 1996,1963 respectively for simulation 1. Similar random trend was observed for simulations 2 to 10 as well. As for the average CPU time versus the number of obstacles, the trend observed is that as the CPU time decreases the obstacles are increased.

From the literature review in chapter 2, it can be seen that various techniques such as Fuzzy Logic, ACO and Genetic Algorithm (GA) have been used in path planning. However, in order to validate the results of this research, two of the recent technigues, i.e., ACO and GA were chosen. Initially, a comparative analysis of the performance of ACO and GA on path planning that was carried out by Hao Mei *et. al* [65] was analysed. It was found that the performance of ACO was better than that of GA in terms of path length, time and turnings. Please refer to Appendix F for futher information about this comparison. In addition, Nohaidda Binti Sariff and Norlida Buniyamin [62] have reported that ACO out-performed GA for path planning with different nodes such as 12, 22 and 63. It was stated that this is because with the increment of length, usually GA also need to increase the population in order to get the optimal path. This will cause the process of GA to find path to become slower. In contrast with GA, for ACO, it is not necessary to increase the population because it will not affect the process. Thus, this helps ACO to minimize the time and number of iterations.

Based on the above reasons and other reported work, it was decided that it is sufficient to concentrate on the use of ACO for this particular research. Thus, further validation was carried out by comparing this research with other research work on the use of ACO only. Hence, Table 4.12 shows the comparative analysis of this work with reported results in the literature.

Table 4.12: Path length (metre) comparison with other authors

No	Author	Technique	Number of obstacles			
			1	2	5	10
1	H.M. Khung [64]	ACO	45.3570	-	-	-
2	Brand [103]	ACO	-	39.0	-	-
3	Proposed Research	ACO	45.4558	45.45	39.4558	40.0416

By refering to the Table 4.12, it shows the comparison of this research with previous researchers, H.M.Zhang [64] and Brand [66]. It shows that for one obstacle [64], the path length are very close where the difference was 0.18% only. This is due to the size of obstascles that were used in both research. It means that this research results are valid. In terms of path length with two obstacles compared with Brand [66], it was found that there was a difference of 6.45 metre. This could be attributed to the vastly different shape and size of the obstacles. However, again it can be concluded that the research results are valid due to the closeness of the results. For other obstacles such as 5 and 10, a direct comparison could not be made since the information was not found form other researchers.

From the simulation results and comparative analysis, it can be deduced that the path planning method based on ACO can find a near optimal path and avoid obstacles timely both in simple (less obstacles) and complex (many obstacles) environment. The reason for this is that ACO exploit the characteristic of solution space in path planning. In the path planning, the solutions near best solution are also good ones, which is called the smoothness of solution. ACO can reach good performance because it updates the pheromone concentration around the elements of current best solution. In this research, a new method for path planning of mobile robot is developed and tested very well. It employs ACO as path planning algorithm. The pheromone generated is exploited by ACO as global information to guide the WMR to jump to local minimum. From the simulation results, it can be seen that by using ACO algorithm, global optimal and real-time obstacle avoidance can be both satisfied.

4.3 Summary

This chapter has discussed the results of the controller design and path planning for differential drive wheel mobile robot. The first section of this chapter presented the findings of controller for conventional, fuzzy logic and hybrid controllers. The second section discussed the results of WMR path planning using Ant Colony Optimization algorithm and its validation. In chapter 5, the research undertaken is summarized with discussions on contributions made and future work is outlined.

CHAPTER 5

CONCLUSIONS

In this thesis, a comprehensive discussion on the control algorithms (conventional controller (P, PI and PID), fuzzy control and hybrid fuzzy-PID) and Ant Colony Optimisation based Path Planning for WMR Navigation have been provided. In section 5.1, a critical analysis of the main achievements of the research reported in this thesis is evaluated by discussing the research results and contributions with respect to the stated objectives. This is followed by some suggestions for future work in section 5.2, and finally it is concluded with remarks on future outlook of WMR in section 5.3.

5.1 Critical Evaluation of Achievements

This research has addressed the two critical issues for WMR in a dynamic environment. Firstly, it dealt with the development of an effective control algorithm based on PID, Fuzzy and Hybrid-Fuzzy-PID, and secondly, the design of an Ant Colony Optimisation (ACO) algorithm to optimize the path planning with obstacle avoidance.

To contextualize the research, a thorough literature review of the technologies and techniques for development of an effective control and path planning algorithms of a WMR in a dynamic environment was presented in Chapter 2 where several aspects of mobile robots, such as history of robotics, functionalities of the mobile robot in terms of drive system, controller of mobile robots, path planning of mobile robot were presented

The implementation of both the issues was addressed in Chapters 3 and 4 where initially the modeling of mobile robot in terms of kinematic model of the WMR and the mathematical model od DC motor were presented. This is followed by discussion on simulation work using the controllers such as conventional controller (P, PI, PID), fuzzy logic controllers (F3M, F5M and F7M) and the proposed hybrid controller (Fuzzy-PID). Finally, it discussed the optimization of path planning of WMR in terms of the principles of ant colony optimization technique and the ant colony optimization algorithm.

In this research, the complete kinematic and mathematical modelling of WMR has been derived. It was useful in the development of the WMR controller. The WMR controller using PID, Fuzzy Logic Controller and Hybrid Controller was obtained. It can be concluded that the Hybrid Controller produces the best results in terms of system performance as compared to the PID and Fuzzy Logic Controllers. On Overall, the Fuzzy logic control with three membership function Proportional Integral controller (F3MPI) gives the better performance as compared to the fuzzy logic controller with higher number membership functions.

In addition, optimization of path planning using ant colony optimization (ACO) technique that was carried out has shown that the path length and the CPU time would decrease when the obstacles were increased. This can be attributed to the fact that the path planning method based on ACO can find a near optimal path and avoid obstacles timely both in simple (less obstacles) and complex (many obstacles) environment. The reason for this is that ACO exploit the characteristic of solution space in path planning. Overall, the results demonstrate the effectiveness of the proposed approach.

Thus, the contributions of this research for WMR in a dynamic environment are summarised as the following:

1. development of an effective Hybrid-Fuzzy-PID control algorithm
2. design of an effective Ant Colony Optimisation (ACO) algorithm to optimize the path planning with obstacle avoidance

5.2 Suggestions for Further Work

In this section, a few suggestions for further research are given.

(i) Newer Control Algorithms

Since this work concentrated on PID, Fuzzy and Fuzzy-PID as Hybrid Controllers, further work could be explored using many other newer techniques such as Bee Colony, Fire-fly, Chaotic Algorithm, etc. to improve the system parameters in terms of overshoot, rise-time and settling time since these techniques could iteratively achieve the fitness function. This enables better tuning of controllers.

(ii) Different Techniques for Path Planning

Increasingly, swarm intelligent techniques are being used in solving many industrial problems including the robotics. Further work could be explored using other swarm intelligent techniques such as Particle Swarm Optimization, Fish Swarm and Bacteria Foraging for path planning. The fast convergence that can be achieved through these techniques could play a vital role for path planning of WMRs to reduce the computation time and path length.

(iii) WMR Prototype Development

Since this work has produce promising results in terms of simulation work, these results could be applied practically by testing them out in a WMR prototype. Therefore, a test bed should be developed for this purpose using a micro-controller to incorporate the control and path planning algorithms, sensors, differential drive mechanism, energy source and other associated components.

5.3 Concluding Remarks

This research has contributed to the development of an effective control algorithm and an improved path planning algorithm for WMR. The utilisation of these algorithms should be able to improve the performance of WMR in terms of better control and faster path planning in near future.

REFERENCES

[1] Pickett, J.P, and et al, *The American Heritage Dictionary of the English Language,*, Fourth Ed. 2000.

[2] McKerrow, P.J, and et al, *Introduction to Robotics, Addison Wesley*. 1995.

[3] J. C. Andersen, “Mobile Robot Navigation,” Technical University of Demmark, 2006

[4] J. R. Assosiation, “Summary Report on Technology Strategy for Creating a ‘ Robot Society ’ in the 21 st Century,” 2001.

[5] G. Campion, G. Bastin, and D’Andrea, “Structural properties and classification of kinematic and dynamic models of wheeled mobile robots,” in *IEEE Transactions on Robotics and Automation,12, 47-61*, 1996.

[6] J.P Laumond , “Robot motion planning and control,” in *Springer-Verlag,Telos*, 1998.

[7] Adams and M.D., “Sensor Modelling, Design and Data Processing for Autonomous Navigation,” in *World Scientific Series in Robotics and Intelligent Systems. Singapore,World Scientific Publishing Co. Ltd.*, 1999.

[8] B. Canudas de Wit, C. Siciliano, (editors), and G.Bastian, “Theory of Robot Control,” in *.New York, Spinger-Verlag*, p. 1996.

[9] S. Nolfi, D., and Floreano, *Evolutionary Robotics: The Biology,Intelligence, and Technology of Self-Organizing Machines. The MIT Press*. 2000.

[10] R. J. Carroll and D. Ruppert, “Transformation and Weighting in Regression,” in *New York, Chapman and Hall*, 1988.

[11] I. J. Cox and G. T. Wilfong, “Autonomous Robot Vehicles.,” *New York, Spinger-Verlag*, 1990.

[12] D. Kortenkamp, M. Huber, C. Cohen, U. Raschke, F. Koss, and C. Congdon, *Integrating high-speed obstacle avoidance, global path planning, and vision sensing on a mobile robot,in Mobile Robots and Artificial Intelligence*. 1998.

[13] “Http://geminoid.dk/.”,Aalborg University in Northern Denmark,2011

[14] Suo Tan, “A GA-Based Fuzzy Controller For Mobile Robots in Dynamic Enviroments with Moving Obstacles ” 2009.

[15] Adams and M.D., “Adams, M.D., Sensor Modelling, Design and Data Processing for Autonomous Navigation. World Scientific Series in Robotics and Intelligent Systems,” in *Singapore,World Scientific Publishing Co. Ltd*, 1999.

[16] Carroll, R.J., Ruppert, and D., “Transformation and Weighting in Regression 1988.,” in *New York, Chapman and Hall*, 1988.

[17] B. A. Garro, H. Sossa, R. A. Vázquez, A. Juan, D. D. Batiz, and M. D. Othon, “Evolving ant colony system for optimizing path planning in mobile robots,” pp. 444-449, CERMA 2007

[18] A. Baker, I. Mackenzie, and A. Baker, “Omnidirectional Drive Systems Kinematics and Control” 2008.

[19] M. N. Mahyuddin, “Mechatronic Design II EEM352 Drive System in Mobile Robot”University Science Malaysia,

[20] U. Nunes, “Diffrential Drive Mobile Robot Robust Motion Control.In Proc. of the 6th Portuguese Conference on Automatic Control (CONTROLO 2004), Faro , Portugal , pp. 115-120, 2004.

[21] A.Veríssimo,“Kinematic Control of the Magellan-ISR Mobile Robot,” vol.1, no.1, pp. 48-57, 2004.

[22] G. Narvydas, R. Simutis, and V. Raudonis, “Autonomous Mobile Robot Control Using Fuzzy Logic and Genetic Algorithm,” no. September, pp. 460-464, 2007.

[23] Roland Siegwart and Illah R. Nourbakhsh, “Autonomous Mobile Robots,” *Technology,* 2004.

[24] G. Narvydas, R. Simutis, and V. Raudonis, “Autonomous Mobile Robot Control Using If-Then Rules and Genetic Algorithm,” Information Technology And Control, Kaunas, Technologija, Vol. 37, No. 3, pp. 193-197, 2008.

[25`] M. John and Holland, “Designing Autonomous Mobile Robots; Inside the Mind of an Intelligent Machine,” Newnes, 2004

[26] D. Floreano and F. Mondada, “Evolution of Homing Navigation in a Real Mobile Robot,”to appear in :IEEE Transactions on system,Man,and Cybernetics-Part B vol. 26, No. 3, June 1996.

[27] David B., and Fogel, *Evolutionary computation: Toward a New Philosophy of Machine Intelligence. 2nd edition, Piscataway, NJ: IEEE Press*. 1998.

[28] D. Chwa, “Tracking Control of Differential-Drive Wheeled Mobile Robots Using a Backstepping-Like Feedback Linearization,” IEEE Press, Trans. Sys. Man Cyber. Part A Vol. 40, No. 6, pp. 1285-1295, 2010.

[29] I. Kolmanovsky and N. H. Mcclamroch, “Developments in Nonholonomic Control Problems,” in *IEEE Control Systems*, pp. 20-36,Dec 1995

[30] F. Zhang, Y. Xi, S. Member, Z. Lin, and W. Chen, “Constrained Motion Model of Mobile Robots and Its Applications,” Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions , Vol. 39, No. 3, pp. 773-787, 2009.

[31] C.-han Lin and C.-ta King, “Sensor-Deployment Strategies for Indoor Robot Navigation,” IEEE Transactions on Systems, Man, and Cybernetics,Vol. 40, No. 2, pp. 388-398, 2010.

[32] T. Dierks and S. Jagannathan, “Neural network control of mobile robot formations using RISE feedback”, IEEE Transactions on Systems, Man and Cybernetics: Part B, Vol. 39, No. 2, pp. 332-347, April 2009..

[34] M. K. Bugeja, S. G. Fabri and L. Camilleri, “Dual adaptive dynamic control of mobile robots using neural networks,” IEEE Transactions on Systems, Man and Cybernetics, Part B, Vol. 39, Issue 1, February 2009, pp. 129-141.

[34] N. Bellotto, S. Member, H. Hu, and S. Member, “Multisensor-Based Human Detection and Tracking for Mobile Service Robots,” International Journal of Social Robotics, Vol. 39, No. 1, pp. 167-181, 2009.

[35] V. M. Peri and D. Simon, "Fuzzy logic control for an autonomous robot," Fuzzy Information Processing Society, 2005. NAFIPS 2005. Annual Meeting of the North American , Vol., No., pp. 337- 342, 26-28 June 2005

[36] T. Das and I. N. Kar, “Design and Implementation of an Adaptive Fuzzy Logic-Based Controller for Wheeled Mobile Robots,” *Control*, Vol. 14, No. 3, pp. 501-510, 2006.

[37] N. Liu, "Intelligent Path Following Method for Nonholonomic Robot Using Fuzzy Control," Intelligent Networks and Intelligent Systems, 2009. ICINIS '09. Second International Conference on , Vol., No., pp.282-285, 1-3 Nov. 2009

[38] H.Chih Huang, "Design of optimal kinematic controller based on GA-ACO hybrid algorithm for four-wheeled omnidirectional mobile robots," Control Conference (ASCC), 2011 8th Asian , Vol., No., pp.902-907, 15-18 May 2011

[39] J.Woo Lee, D.Hyun Lee, and J.Jang Lee, “Global Path Planning Using Improved Ant Colony Optimization Algorithm through Bilateral Cooperative Exploration,” *Ecosystems*, Vol. 5, No. June, pp. 109-113, 2011.

[40] C. Zhang, Z. Zhen, D. Wang, and M. Li, “"UAV path planning method based on ant colony optimization," Control and Decision Conference (CCDC), 2010 Chinese , Vol., No., pp.3790-3792, 26-28 May 2010

[41] Y. Z. Cong and S. G. Ponnambalam, "Mobile robot path planning using ant colony optimization," Advanced Intelligent Mechatronics, 2009. AIM 2009. IEEE/ASME International Conference on , Vol., No., pp.851-856, 14-17 July 2009.

[42] K. H. Sedighi, K. Ashenayi, T. W. Manikas, R. L. Wainwright, and H.-ming Tai, "Autonomous local path planning for a mobile robot using a genetic algorithm," Evolutionary Computation, 2004. CEC2004. Congress on , Vol.2, No., pp. 1338- 1345 Vol.2, 19-23 June 2004

[43] O. Castillo and L. Trujillo, "Multiple Objective Optimization Genetic Algorithms for Path Planning in Autonomous Mobile Robots," *International Journal of Computers*, Vol. 6, No. 1, pp. 48-63, 2005.

[44] Y. Q. Kaiyou Lei, "A novel path planning for mobile robots using modified particle swarm optimizer," Systems and Control in Aerospace and Astronautics, 2006. ISSCAA 2006. 1st International Symposium on , Vol., No., pp.4 pp.-984, 19-21 Jan. 2006

[45] Q. Li, X. Tong, and S. Xie, "Optimum Path Planning for Mobile Robots Based on a Hybrid Genetic Algorithm," *Hybrid Intelligent Systems, 2006. HIS '06. Sixth International Conference on* , Vol., No., pp.53, Dec. 2006

[46] J.Woo Lee, J.Jung Kim, and J.Jang Lee, "Improved Ant Colony Optimization algorithm by potential field concept for optimal path planning," Humanoid Robots, 2008. Humanoids 2008. 8th IEEE-RAS International Conference on , Vol., No., pp.662-667, 1-3 Dec. 2008

[47] P. Melin, M. A. P. Garcia, O. Montiel, O. Castillo, and R. Sepu, "Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation," *Applied Soft Computing*, Vol. 9, pp. 1102-1110, 2009.

[48] S.Hiang Chia, K.Lan Su, J.Hung Guo, and C.Yun Chung, "Ant Colony System Based Mobile Robot Path Planning," Genetic and Evolutionary Computing (ICGEC), 2010 Fourth International Conference on , Vol., No., pp.210-213, 13-15 Dec. 2010

[49] Dudek and Jenkin, *Computational principles of mobile robotic,Cambridge University Publications*. 2000.

[50] M., I., and Ribeiro, *M. I. Ribeiro and P. Lima, "Kinematics Models of Mobile Robots", Av. Rovisco Pais, 11049-001 Lisboa, April, 2000*

[51] M. Oubbati, M. Schanz, P. Levi, D.- Stuttgart, and P. L. De, "Kinematic and dynamic adaptive control of a nonholonomic mobile robot using a RNN," Computational Intelligence in Robotics and Automation, 2005. CIRA 2005. Proceedings. 2005 IEEE International Symposium on , Vol., No., pp. 27- 33, 27-30 June 2005

[52] “http://www.robotix.gr/robotics/XBotPart2/.”Sunday, 6 Jan 2013 12:53

[53] Dorf and Bishop, *Modern Control Systems, 9th Ed., Prentice-Hall, Inc.* 2001.

[54] M.Namazov and O. Basturk"DC Motor Position Control Using Fuzzy Proportional-Derivative Controllers with Diffrent Defuzzification Maethods ",TJFS: Turkish Journal of Fuzzy Systems An Official Journal of Turkish Fuzzy Systems Association,” Vol. 1, pp. 36-54, 2010.

[55] R. Choomuang and N. Afzulpurkar, “Hybrid Kalman Filter / Fuzzy Logic based Position Control of Autonomous Mobile Robot,” *Advanced Robotic*, vol. 2, no. 3, pp. 197 - 208, 2005.

[56] Timothy J. Ross, *“ Fuzzy logic with Engineering Applications” international Edition*. McGraw-Hill,1997.

[57] M. Brandi, M. Masudai, N. Wehner, and X.-hua Yu, "Ant Colony Optimization algorithm for robot path planning," Computer Design and Applications (ICCDA), 2010 International Conference on , Vol.3, No., pp.V3-436-V3-440, 25-27 June 2010

[58] M. Dorigo and T. Stützle, “Ant Colony Optimization,” 2004.

[59] N. B. Sariff and N. Buniyamin, “Ant Colony System for Robot Path Planning in Global Static Environment, In Proceedings of the 9th WSEAS international conference on System science and simulation in engineering (ICOSSSE'10), Hamido Fujita and Jun Sasaki (Eds.). World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA, 192-197 ,2010

[60] A. K. Kulatunga, D. K. Liu, and G. Dissanayake, "Ant Colony Optimization based Simultaneous Task Allocation and Path Planning of Autonomous Vehicles," Cybernetics and Intelligent Systems, 2006 IEEE Conference on , Vol., No., pp.1-6, 7-9 June 2006

[61] M. Dorigo and T. Stützle. The Ant Colony Optimization Metaheuristic: Algorithms, Applications and Advances. In F. Glover and G. Kochenberger editors, Handbook of Metaheuristics, volume 57 of International Series in Operations Research & Management Science, pp 251-285. Kluwer Academic Publishers, Norwell, MA,2002.

[62] N. Binti and S. Sariff, "Comparative study of Genetic Algorithm and Ant Colony Optimization algorithm performances for robot path planning in global static environments of different complexities," Computational Intelligence in Robotics and Automation (CIRA), 2009 IEEE International Symposium on , vol., no., pp.132-137, 15-18 Dec. 2009.

[63] N.Giap,J-H Sin and W-H Kim, "Adaptive roboust fuzzy control for path tracking of a wheeled mobile robot",Artificial Life, pp 134-138,2008.

[64] H.M. Kyung ," Collision free path planning algorithm for navigation problem", Faculty of the Graduate School ,University of Missouri-Columbia,2007.

[65] H. Mei,Y.Tien, L. Zu," A Hybrid Ant Colony Optimization Algorithm for Path Planning of Robot in Dynamic Environment', International Journal of Information Technology, Vol.12, No.3, 2006.

[66] M. Brand, M. Masuda, N. Wehner, X. H. Yu; , "Ant Colony Optimization algorithm for robot path planning," Computer Design and Applications (ICCDA), 2010 International Conference on , Vol.3, No., pp.V3-436-V3-440, 25-27 June 2011

LIST OF PUBLICATIONS

[1] R. Rashid, I. Elamvazuthi, M. Begam, and M. Arrofiq, “Fuzzy-based Navigation and Control of a NonHolonomic Mobile Robot,” *Journal of Computing*, vol. 2, no. 3, 2010.

[2] R. Rashid and I. Elamvazuthi, “Differential Drive Wheeled Mobile Robot (WMR) Control using Fuzzy Logic Techniques,” in *Asia International Conference on Mathematical/Analytical Modelling and Computer Simulation*, 2010, pp. 51-55.

[3] R. Rashid, I. Elamvazuthi, and M. K. A. A. Khan, “Mamdani Fuzzy System for indoor Autonomous Mobile Robot,” in *Global Conference on Power Control and Optimization,Kuching,Malaysia*, 2010, no. December, pp. 2-4.

[4] R. Rashid, I. Elamvazuthi, K. M. Begam, M. L. M. Zain, and L. Muda, “Intelligent Controller for Differential Drive Wheeled Mobile Robot (WMR): A Comparative Analysis,” in *International Conference on Robotic Automation System (ICORAS 2011)*, pp. 3-6.

APPENDIX A

MAXON DC MOTOR

1. Motor operation range

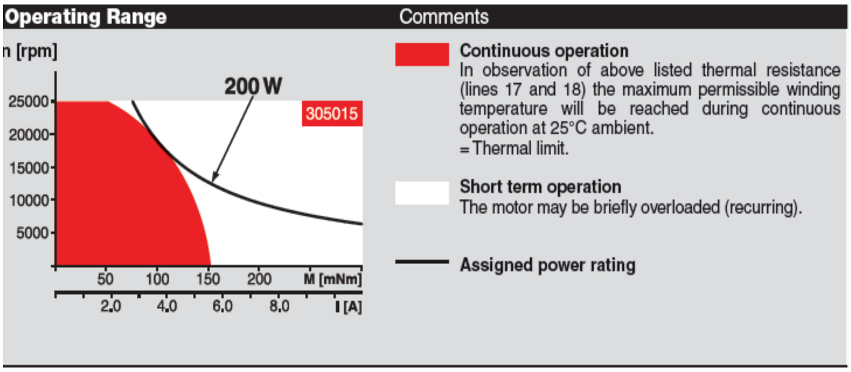

Figure A.1 Operating range Maxon EC-4 pole 30 DC brushless motor

2. Motor Data

Motor Data (provisional)			305013	305014	305015
	Values at nominal voltage				
1	Nominal voltage	V	24.0	36.0	48.0
2	No load speed	rpm	17000	17000	16500
3	No load current	mA	885	590	422
4	Nominal speed	rpm	16200	16200	15800
5	Nominal torque (max. continuous torque)	mNm	112	117	118
6	Nominal current (max. continuous current)	A	9.08	6.31	4.64
7	Stall torque	mNm	3180	3460	3430
8	Starting current	A	236	171	124
9	Max. efficiency	%	88	89	89
	Characteristics				
10	Terminal resistance phase to phase	Ω	0.102	0.210	0.386
11	Terminal inductance phase to phase	mH	0.0163	0.0368	0.0653
12	Torque constant	mNm / A	13.5	20.2	27.6
13	Speed constant	rpm / V	710	473	346
14	Speed / torque gradient	rpm / mNm	5.36	4.92	4.83
15	Mechanical time constant	ms	1.87	1.72	1.68
16	Rotor inertia	gcm²	33.3	33.3	33.3

Figure A.2 Data Maxon EC-4 pole 30 DC brushless motor

3. Motor specification

Specifications		
	Thermal data	
17	Thermal resistance housing-ambient	5.3 K / W
18	Thermal resistance winding-housing	0.209 K / W
19	Thermal time constant winding	2.11 s
20	Thermal time constant motor	848 s
21	Ambient temperature	-20 ... +100°C
22	Max. permissible winding temperature	+155°C
	Mechanical data (preloaded ball bearings)	
23	Max. permissible speed	25000 rpm
24	Axial play at axial load < 8.0 N	0 mm
	> 8.0 N	0.14 mm
25	Radial play	preloaded
26	Max. axial load (dynamic)	5.5 N
27	Max. force for press fits (static)	73 N
	(static, shaft supported)	1300 N
28	Max. radial loading, 5 mm from flange	25 N
	Other specifications	
29	Number of pole pairs	2
30	Number of phases	3
31	Weight of motor	300 g

Figure A.3 Specfication Maxon EC-4 pole 30 DC brushless motor

APPENDIX B

FUZZY MEMBERSHIP FUNCTION

1. **Fuzzy with three membership function (F3M) for target point 14 metre (10 metre x-position and 10 metre y-position)**

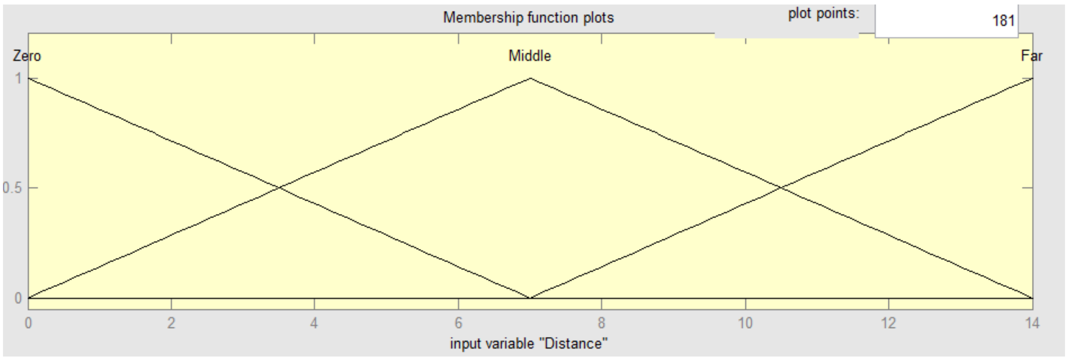

Figure B.1 Representation input of the fuzzy three membership function of the distance error for target point 14 metre

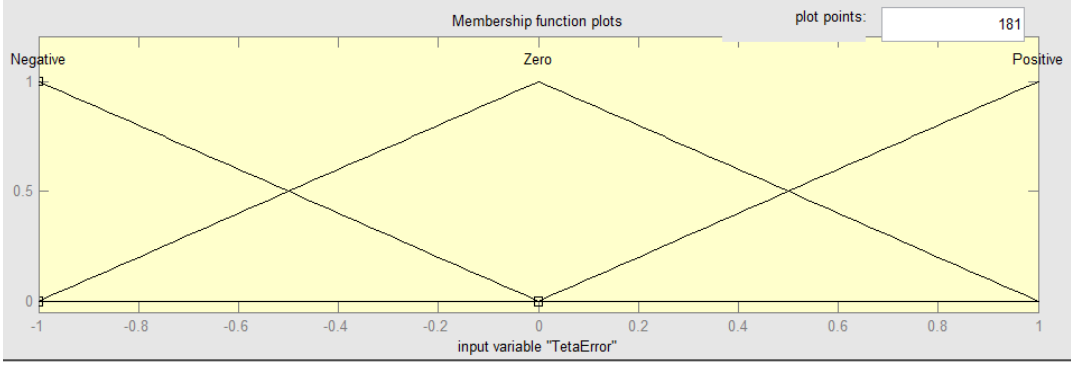

Figure B.2 Representation input of the fuzzy three membership function of the teta error for target point 14 metre

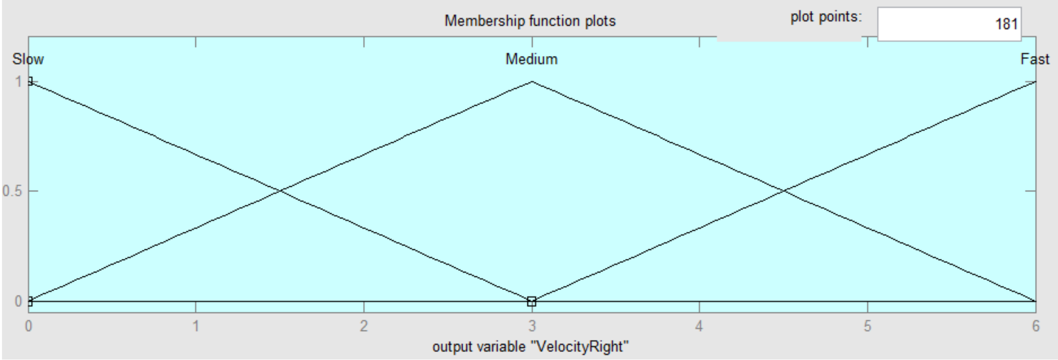

Figure B.3 Representation output of the fuzzy three membership function velocity of the Right motor for target point 14 metre

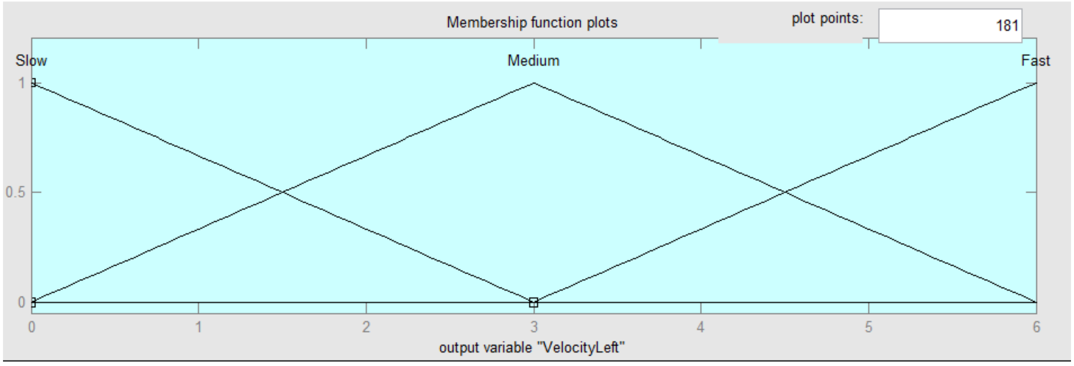

Figure B.4 Representation output of the fuzzy three membership function velocity of the Left motor for target point 14 metre

2. **Fuzzy with three membership function (F3M) for target point 32 metre (30 metre x-position and 10 metre y-position)**

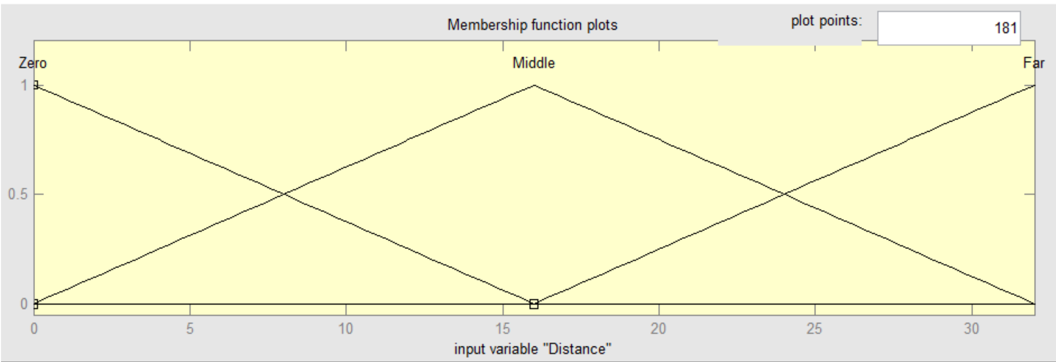

Figure B.5 Representation input of the fuzzy three membership function of the distance error for target point 32 metre

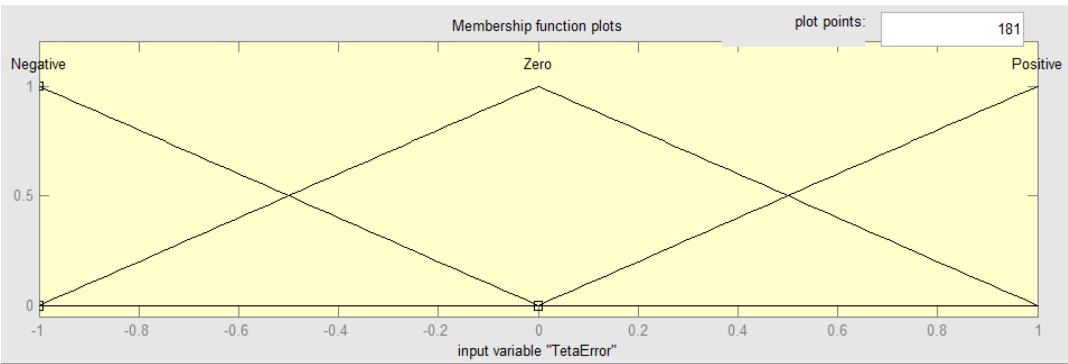

Figure B.6 Representation input of the fuzzy three membership function of the teta error for target point 32 metre

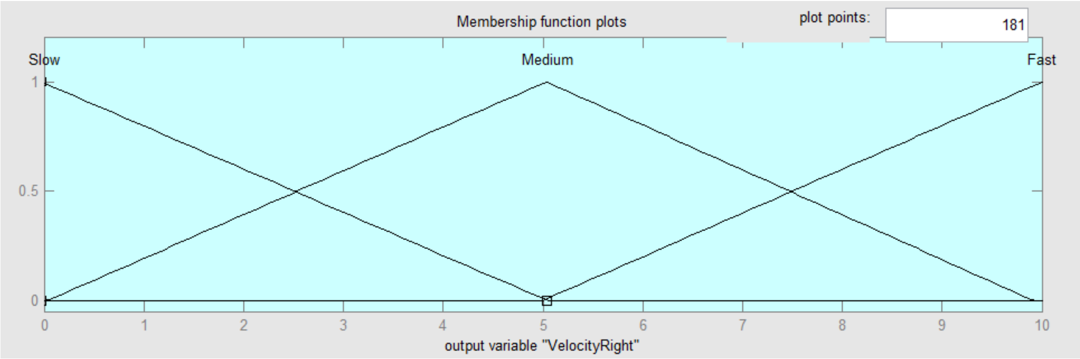

Figure B.7 Representation output of the fuzzy three membership function velocity of the Right motor for target point 32 metre

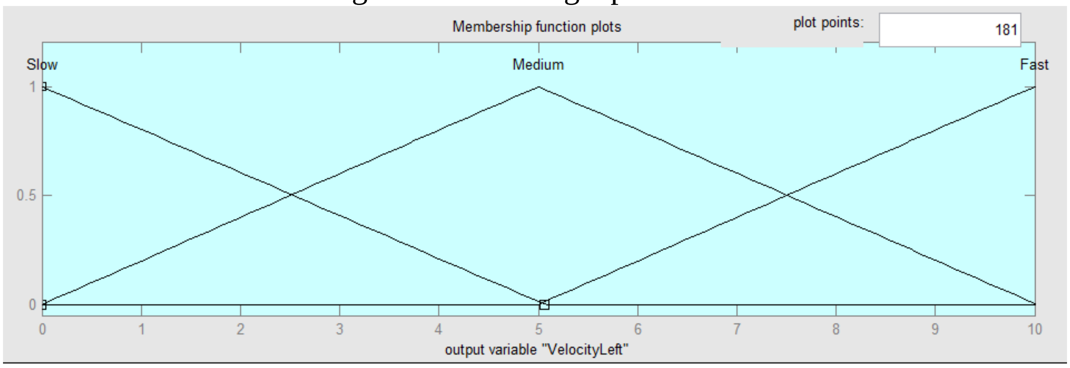

Figure B.8 Representation output of the fuzzy three membership function velocity of the Left motor for target point 32 metre

3. **Fuzzy with three membership function (F3M) for target point 63 metre (60metre x-position and 20metre y-position)**

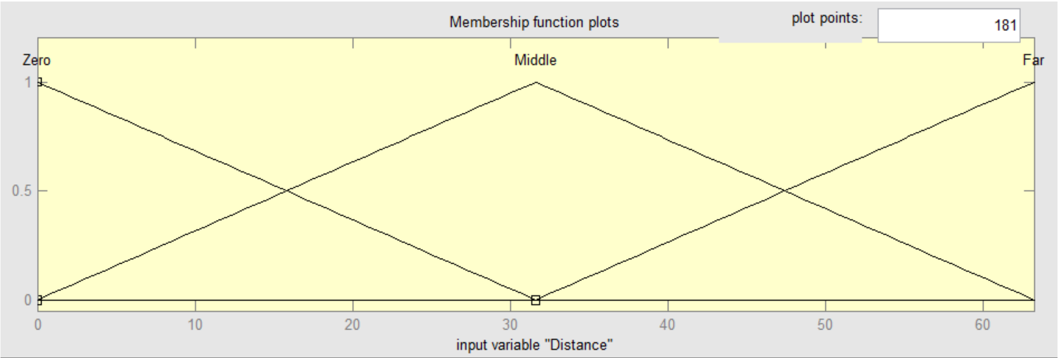

Figure B.9 Representation input of the fuzzy three membership function of the distance error for target point 63 metre

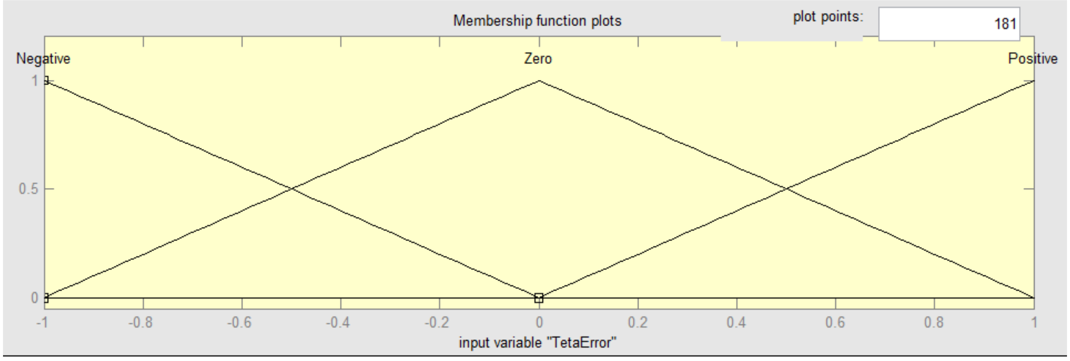

Figure B.10 Representation input of the fuzzy three membership function of the teta error for target point 63 metre

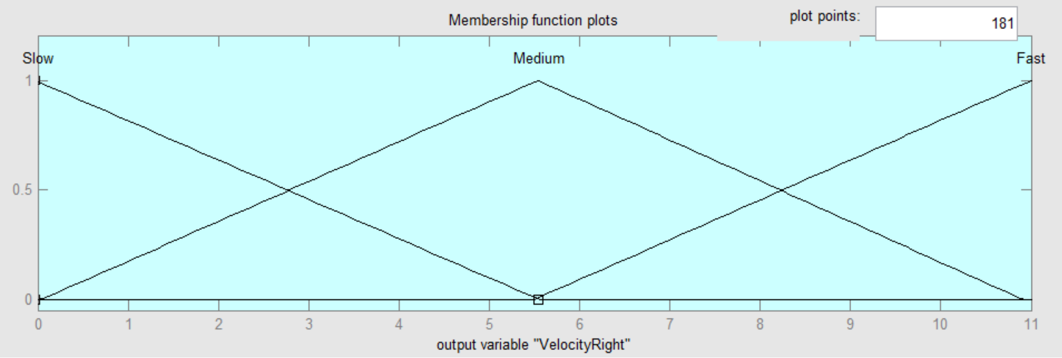

Figure B.11 Representation output of the fuzzy three membership function velocity of the Right motor for target point 63 metre

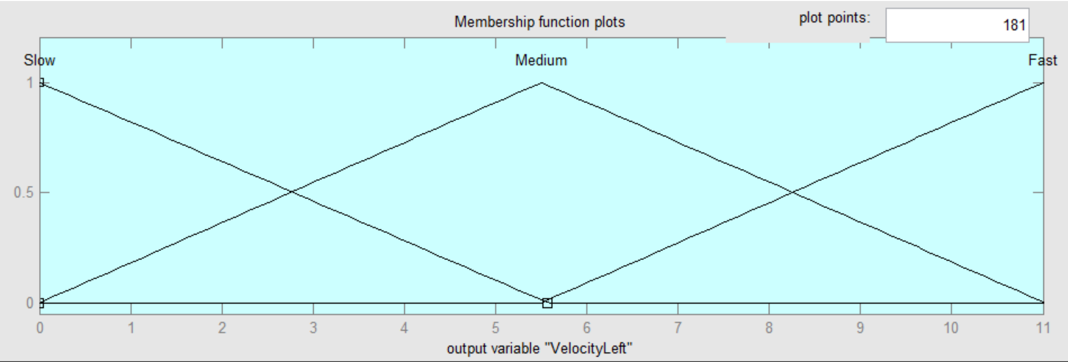

Figure B.12 Representation output of the fuzzy three membership function velocity of the Left motor for target point 63 metre

4. **Fuzzy with five membership function (F5M) for target point 14 metre (10metre x-position and 10metre y-position)**

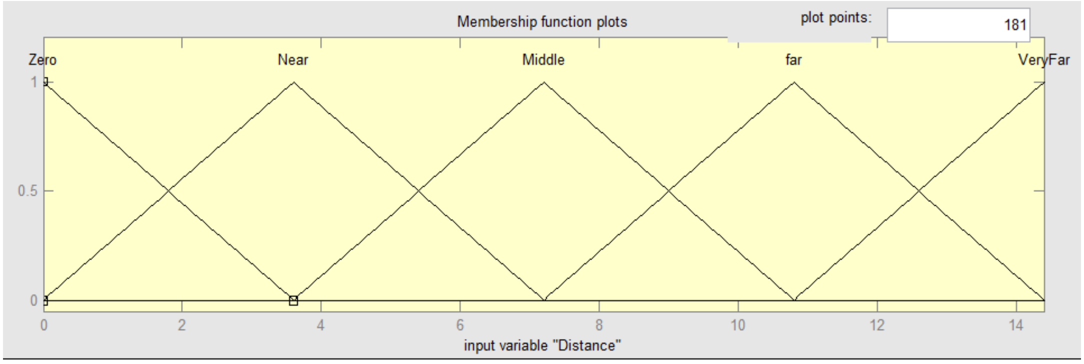

Figure B.13 Representation input of the fuzzy five membership function of the distance error for target point 14 metre

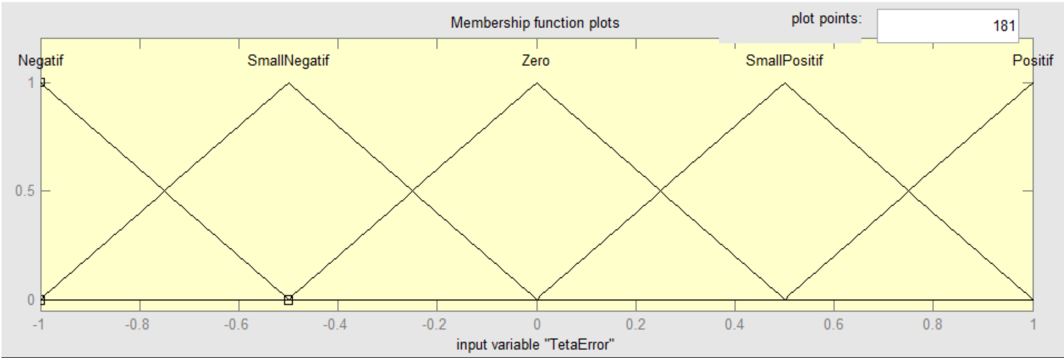

Figure B.14 Representation input of the fuzzy five membership function of the teta error for target point 14 metre

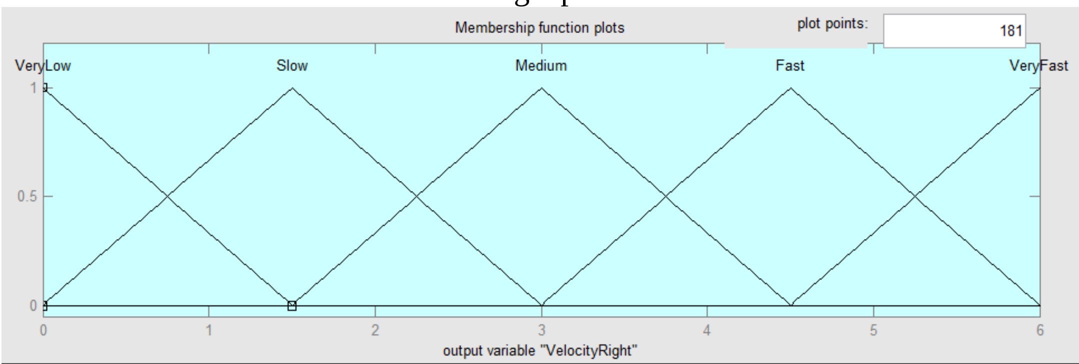

Figure B.15 Representation output of the fuzzy five membership function velocity of the Right motor for target point 14 metre

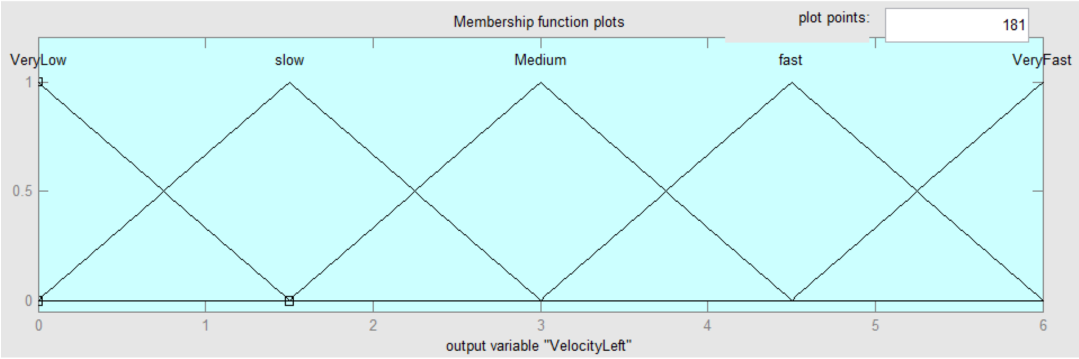

Figure B.16 Representation output of the fuzzyfive membership function velocity of the Left motor for target point 14 metre

5. **Fuzzy with five membership function (F5M) for target point 32 metre (30metre x-position and 10metre y-position)**

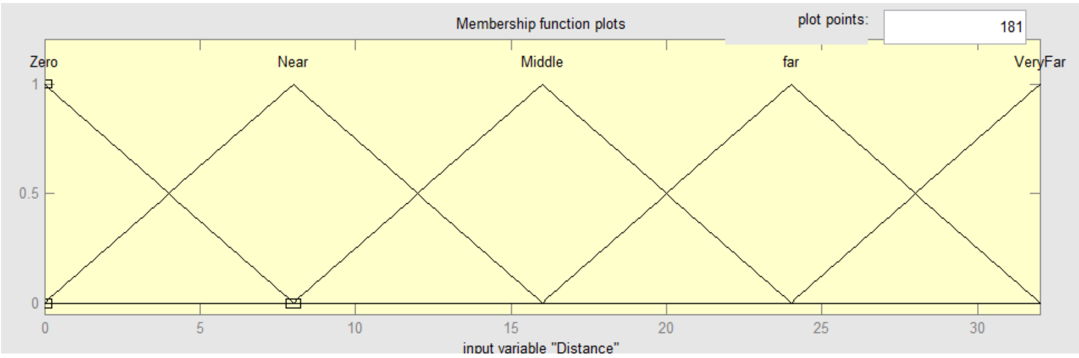

Figure B.17 Representation input of the fuzzy five membership function of the distance error for target point 32 metre

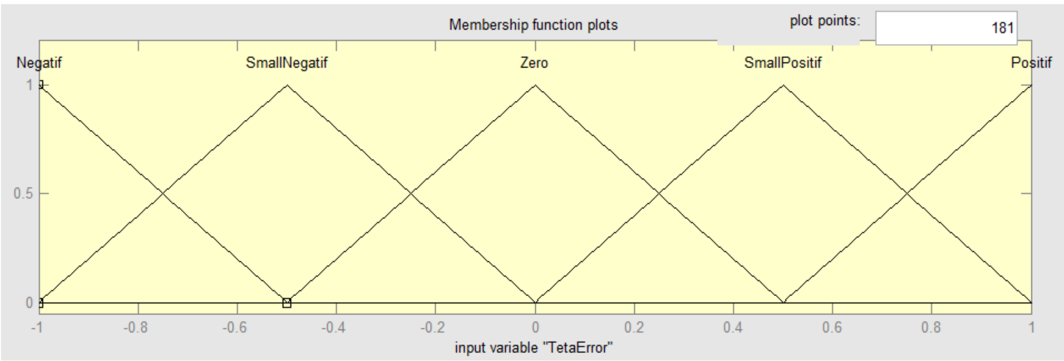

Figure B.18 Representation input of the fuzzy five membership function of the teta error for target point 32 metre

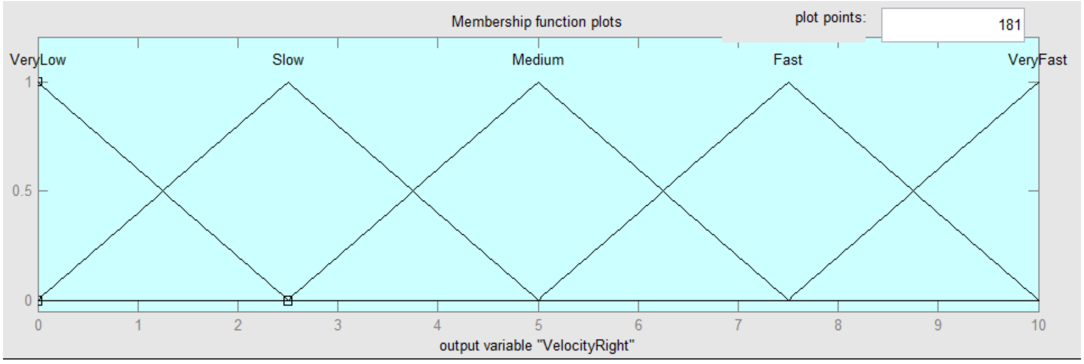

Figure B.19 Representation output of the fuzzy five membership function velocity of the Right motor for target point 32 metre

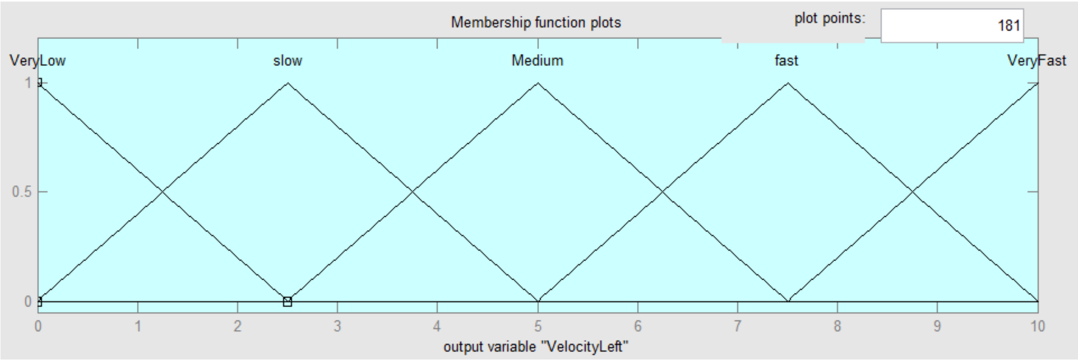

Figure B.20 Representation output of the fuzzyfive membership function velocity of the Left motor for target point 32 metre

6. **Fuzzy with five membership function (F5M) for target point 63 metre (60metre x-position and 20metre y-position)**

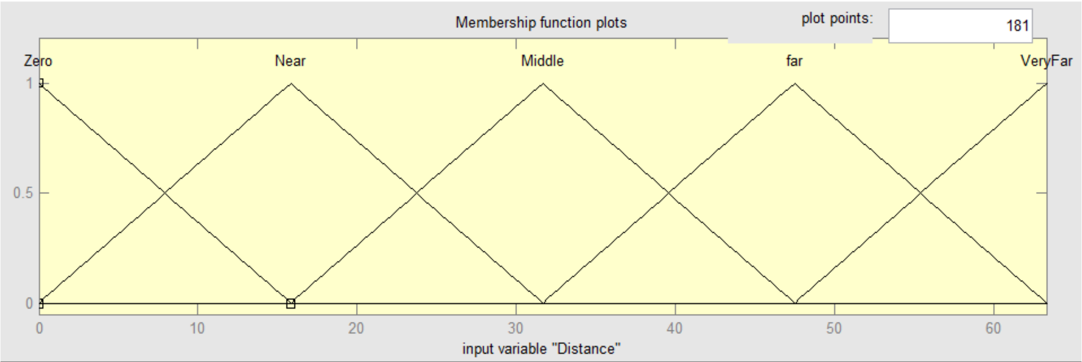

Figure B.17 Representation input of the fuzzy five membership function of the distance error for target point 63 metre

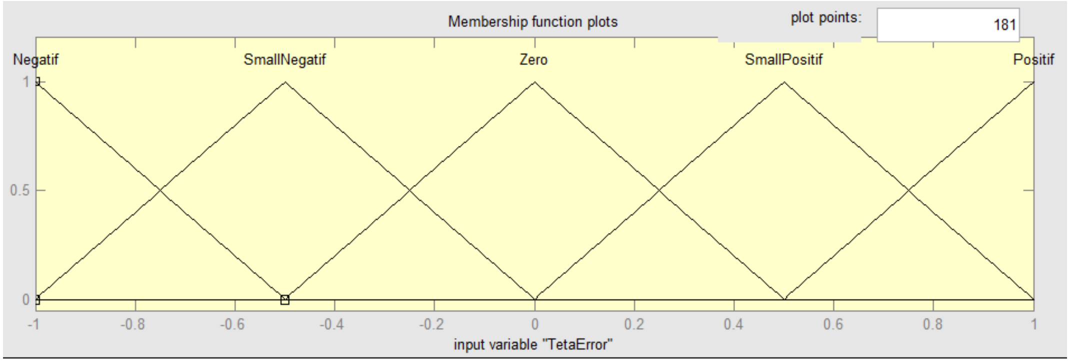

Figure B.18 Representation input of the fuzzy five membership function of the teta error for target point 63 metre

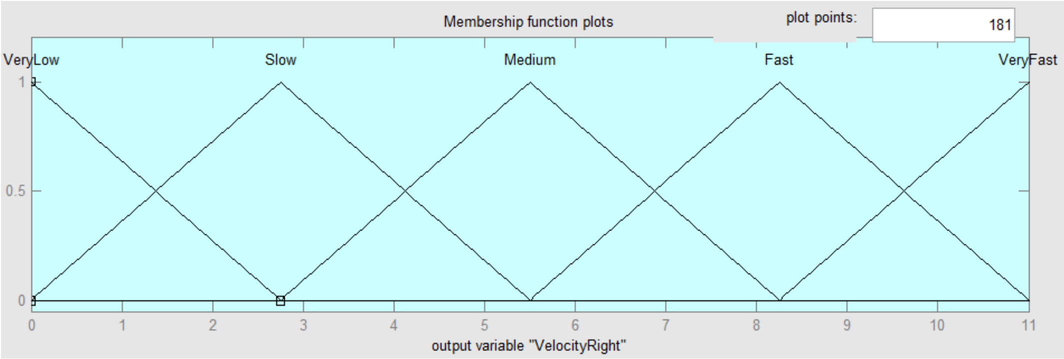

Figure B.19 Representation output of the fuzzy five membership function velocity of the Right motor for target point 63 metre

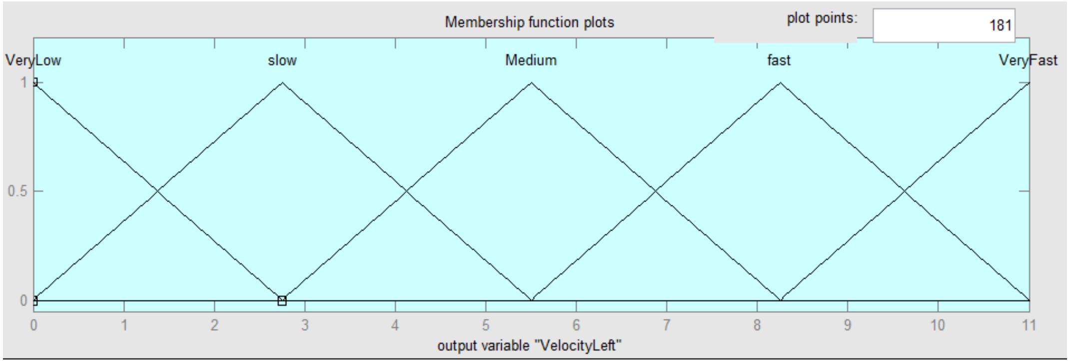

Figure B.20 Representation output of the fuzzyfive membership function velocity of the Left motor for target point 63 metre

7. Fuzzy with seven membership function (F7M) for target point 14 metre (10metre x-position and 10metre y-position)

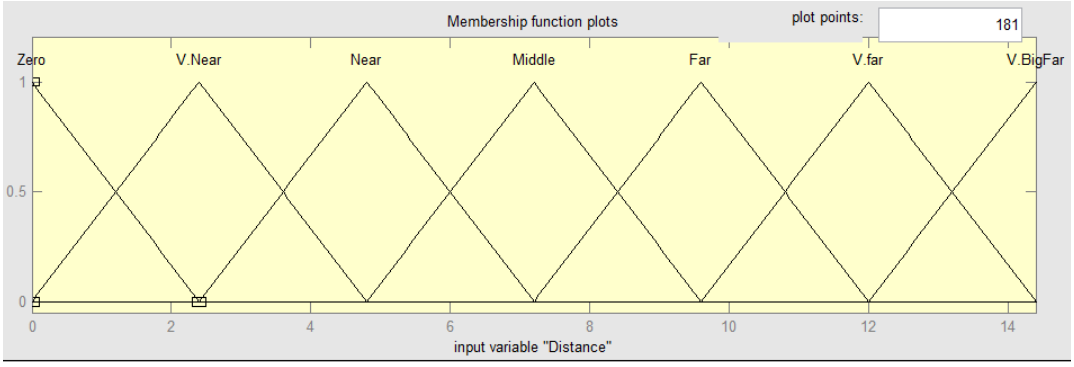

Figure B.17 Representation input of the fuzzy seven membership function of the distance error for target point 14 metre

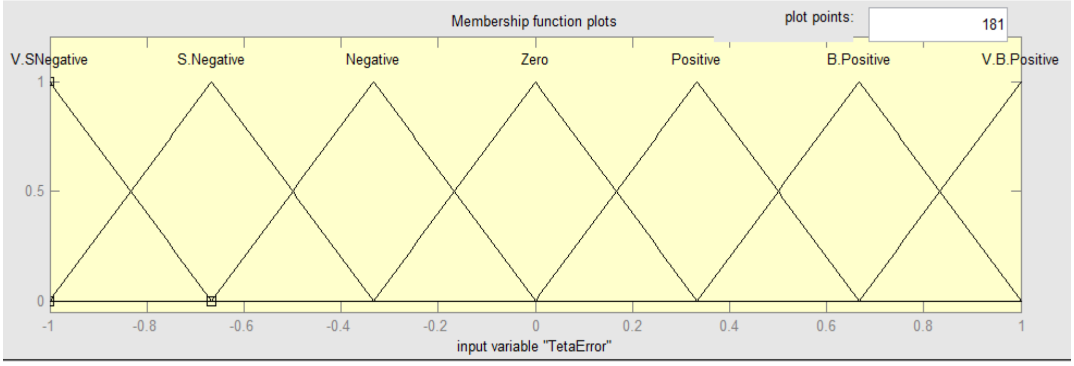

Figure B.18 Representation input of the fuzzy seven membership function of the teta error for target point 14 metre

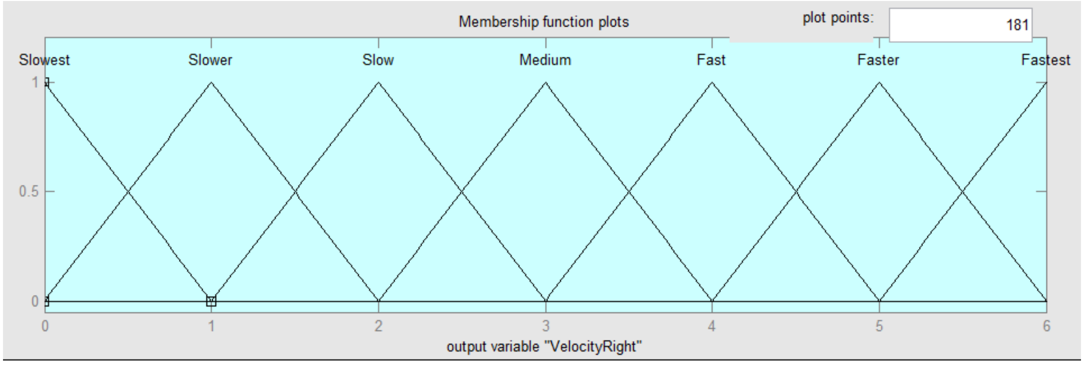

Figure B.19 Representation output of the fuzzy seven membership function velocity of the Right motor for target point 14 metre

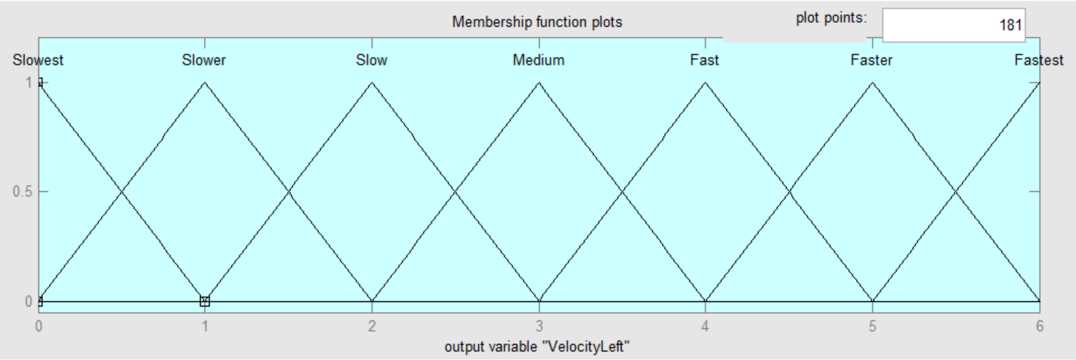

Figure B.20 Representation output of the fuzzy seven membership function velocity of the Left motor for target point 14 metre

8. **Fuzzy with seven membership function (F7M) for target point 32 metre (30 metre x-position and 10metre y-position)**

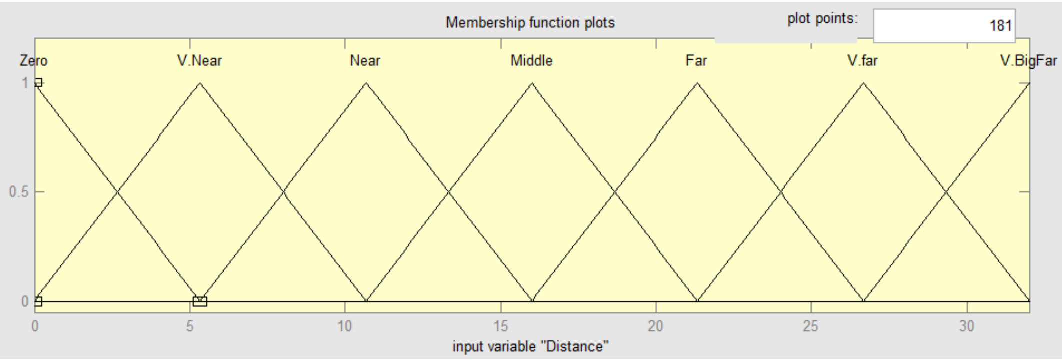

Figure B.17 Representation input of the fuzzy seven membership function of the distance error for target point 32 metre

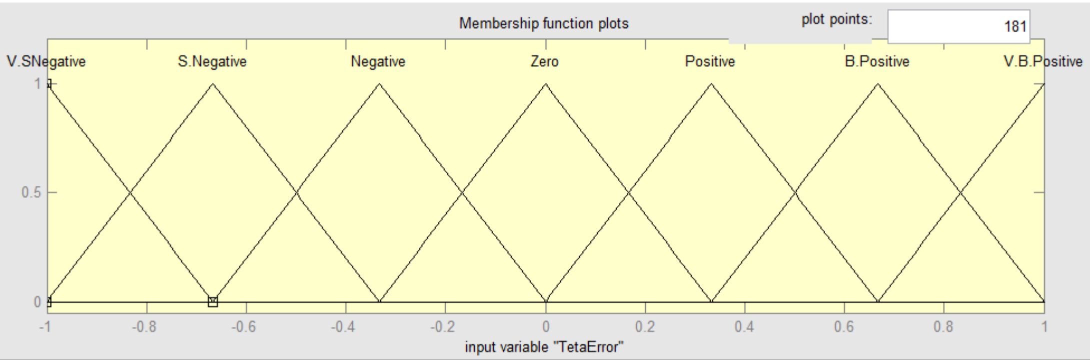

Figure B.18 Representation input of the fuzzy seven membership function of the teta error for target point 32 metre

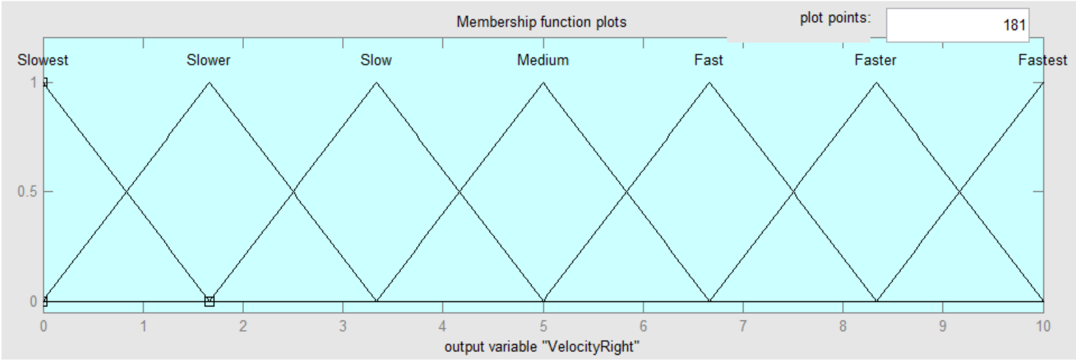

Figure B.19 Representation output of the fuzzy seven membership function velocity of the Right motor for target point 32 metre

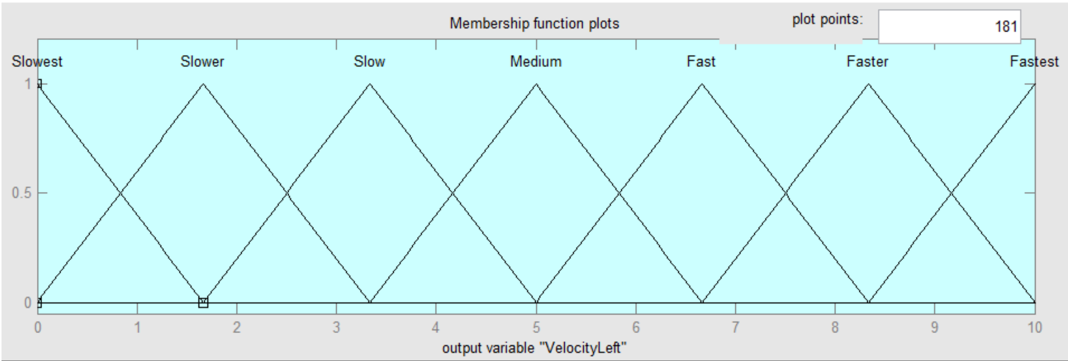

Figure B.20 Representation output of the fuzzy seven membership function velocity of the Left motor for target point 32 metre

9. Fuzzy with seven membership function (F7M) for target point 63 metre (60metre x-position and 20metre y-position)

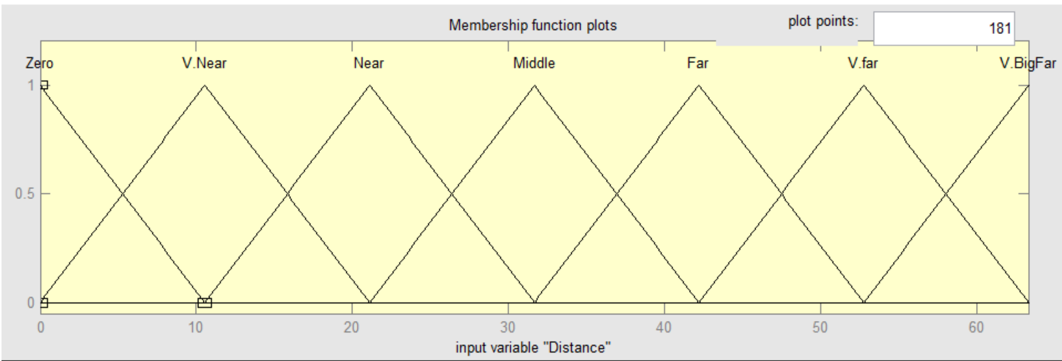

Figure B.21 Representation input of the fuzzy seven membership function of the distance error for target point 63 metre

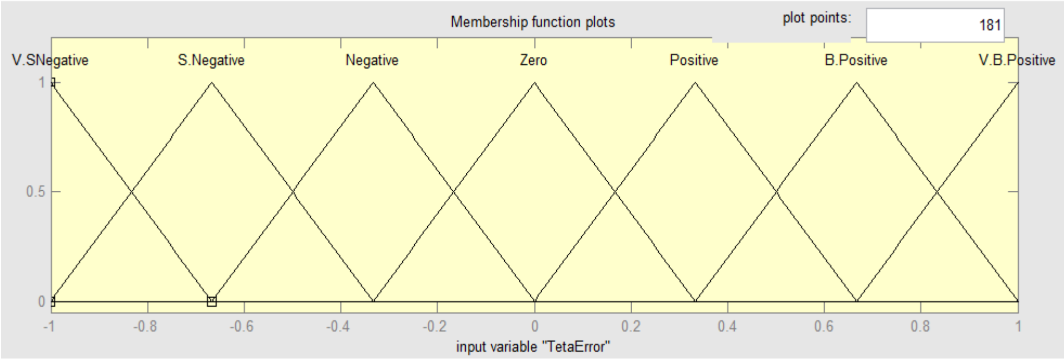

Figure B.22 Representation input of the fuzzy seven membership function of the teta error for target point 63 metre

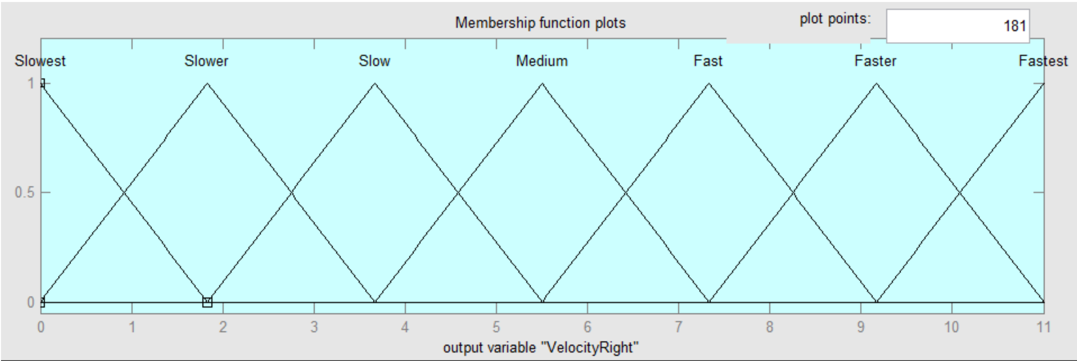

Figure B.23 Representation output of the fuzzy seven membership function velocity of the Right motor for target point 63 metre

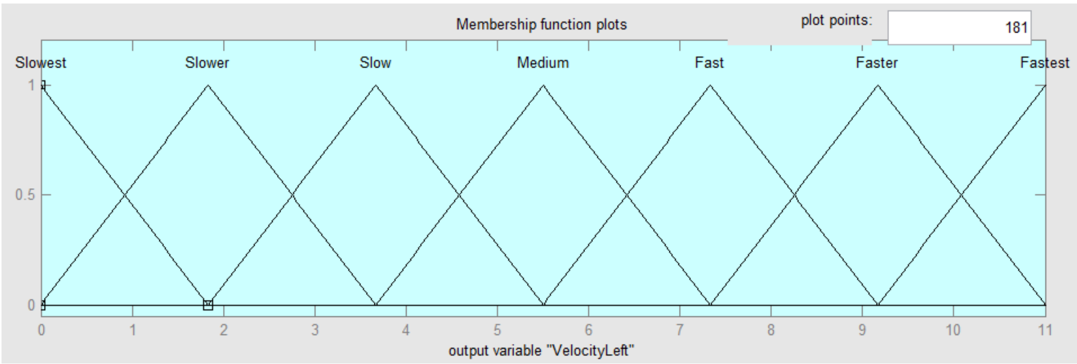

Figure B.24 Representation output of the fuzzy seven membership function velocity of the Left motor for target point 63 metre

APPENDIX C

SIMULINK MODEL WMR

1. Simulink Model Conventional Controller WMR

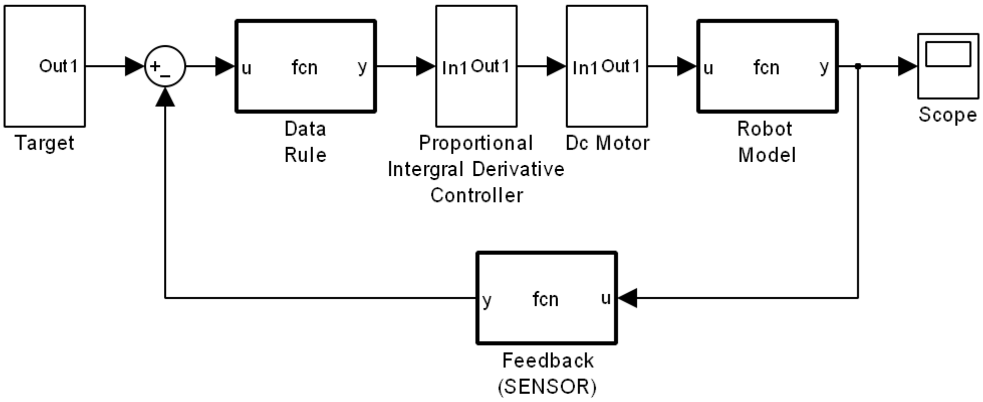

Figure **C.1:**Simplified Proportional Integral Derivative Controller simulation diagram

2. Simulink Model Fuzzy Logic Controller WMR

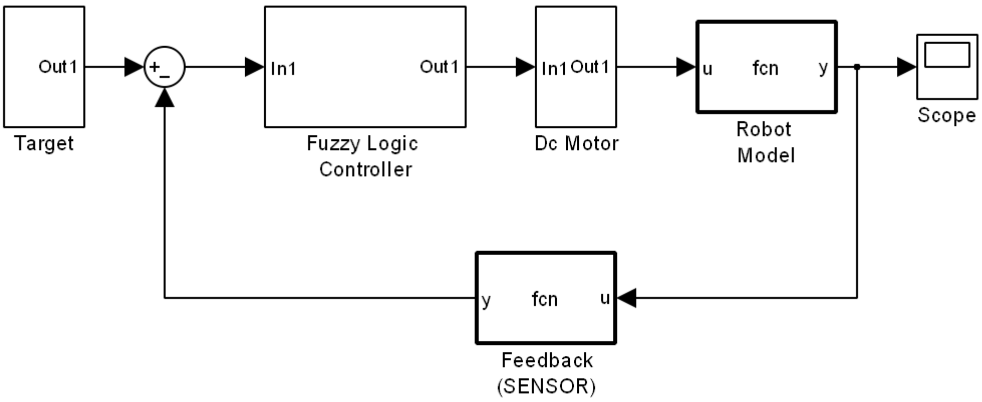

Figure C.2: Simplified Fuzzy Logic Controller simulation diagram

3. Simulink Model Fuzzy Logic and Proportional Controller WMR

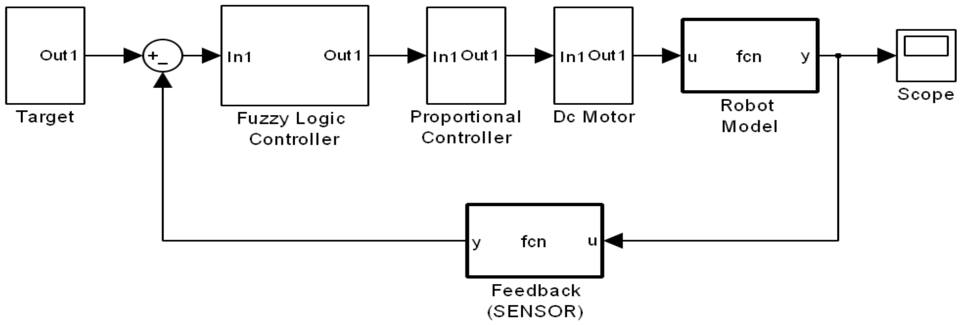

Figure C.3: Simplified Fuzzy Logic and Proportional Controller simulation diagram

4. Simulink Model Fuzzy Logic and Proportional Intergral Controller WMR

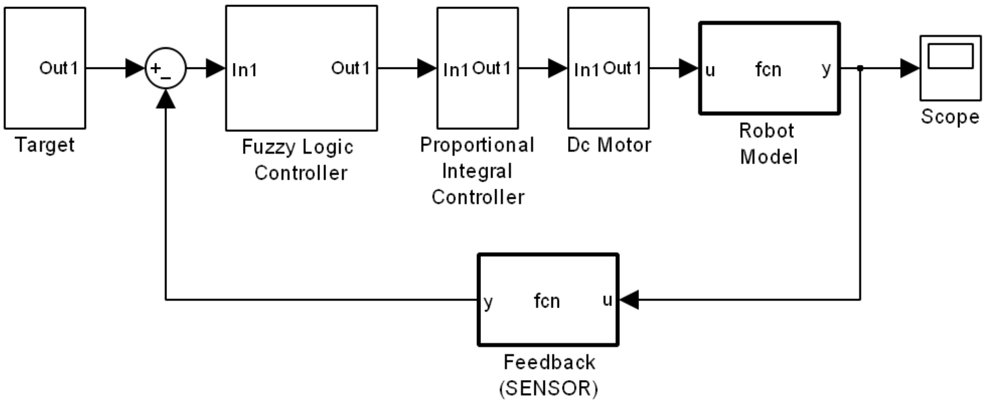

Figure C.4: Simplified Fuzzy Logic and Proportional Integral Controller simulation diagram

5. Simulink Model Fuzzy Logic and Proportional Integral Derivate Controller WMR

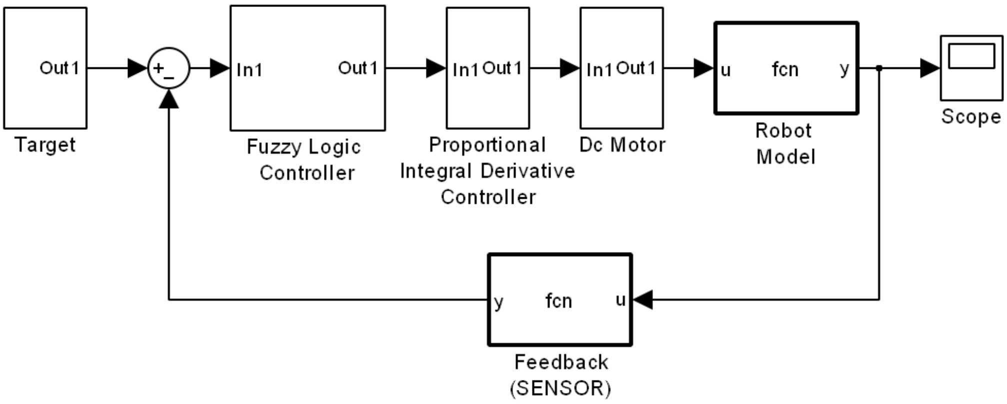

Figure C.5: Simplified Fuzzy Logic and Proportional Integral Derivative Controller simulation diagram

APPENDIX D

RESULT FOR DIFFERENT CONTROLLER

1. Proportional (P) Controller for 10x10y

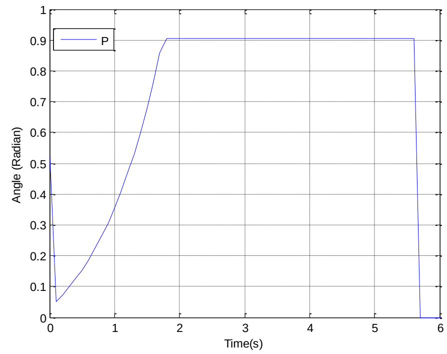

Figure D.1 Angle of orientation for P Controller

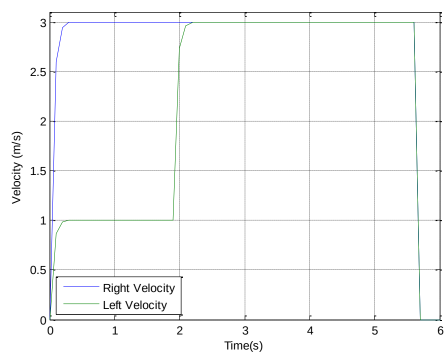

Figure D.2 Right and Left Velocity of mobile robot for P Controller

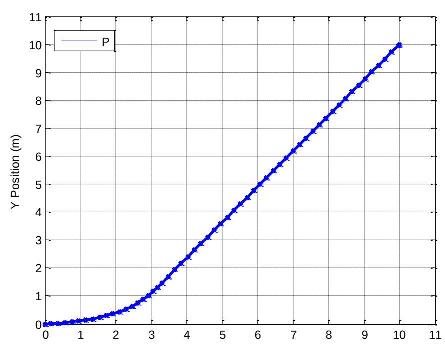

Figure D.3 Position of mobile robot using P Controller

2. Proportional (P) Controller for 30x10y

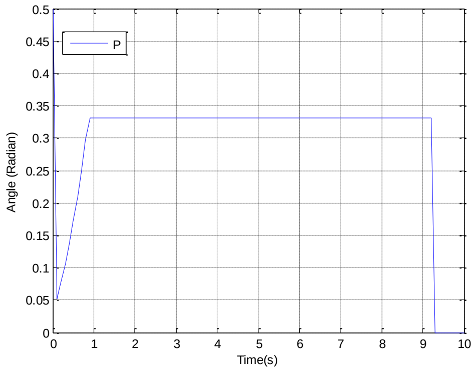

Figure D.4 Angle of orientation for P Controller

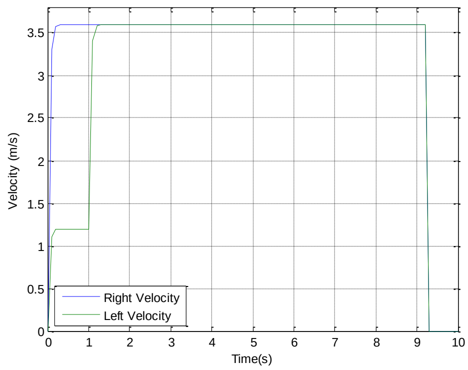

Figure D. 5 Right and Left Velocity of mobile robot for P Controller

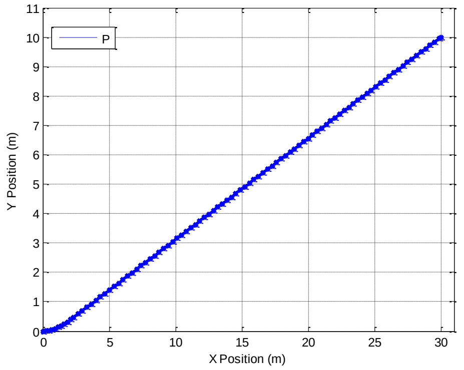

Figure D.6 Position of mobile robot using P Controller

3. Proportional (P) Controller for 60x20y

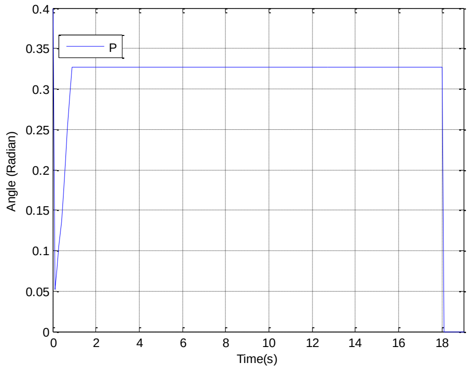

Figure D.7 Angle of orientation for P Controller

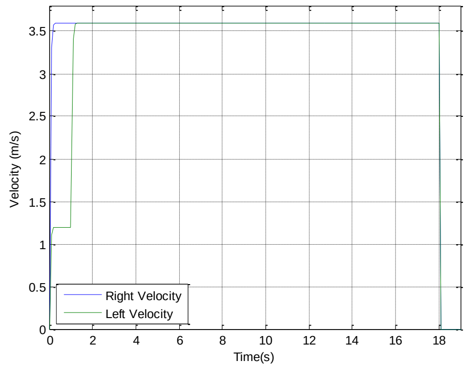

Figure D.8 Right and Left Velocity of mobile robot for P Controller

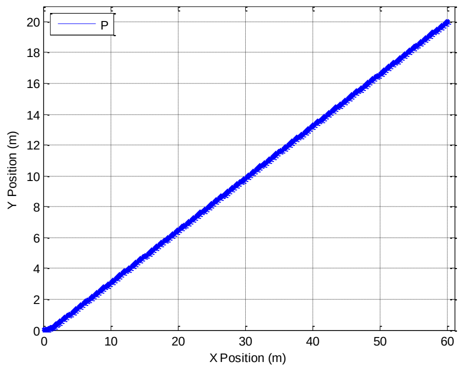

Figure D.9 Position of mobile robot using P Controller

4. Proportional Integral (PI) Controller for 10x10y

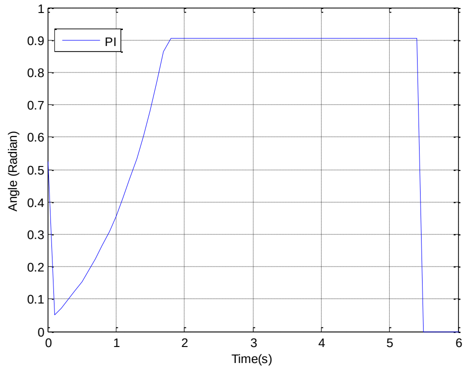

Figure D.10 Angle of orientation for PI Controller

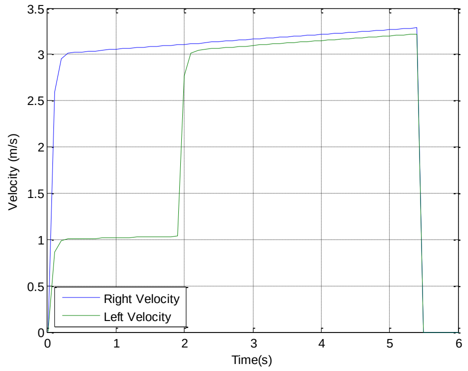

Figure D.11 Right and Left Velocity of mobile robot for PI Controller

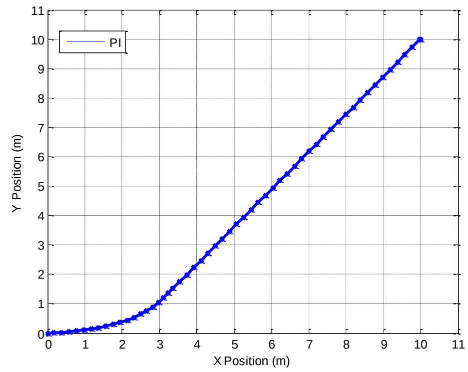

Figure D.12 Position of mobile robot using PI Controller

5. Proportional Integral (PI) Controller for 30x10y

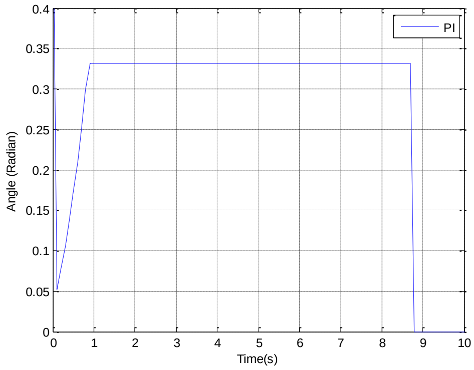

Figure D.13 Angle of orientation for PI Controller

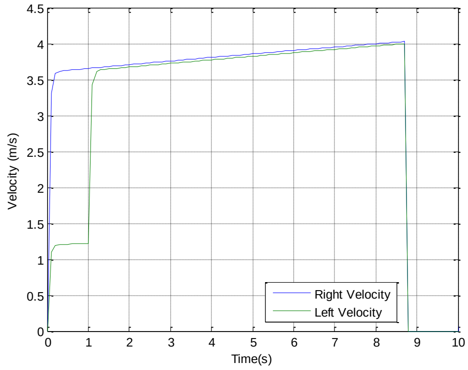

Figure D. 14 Right and Left Velocity of mobile robot for PI Controller

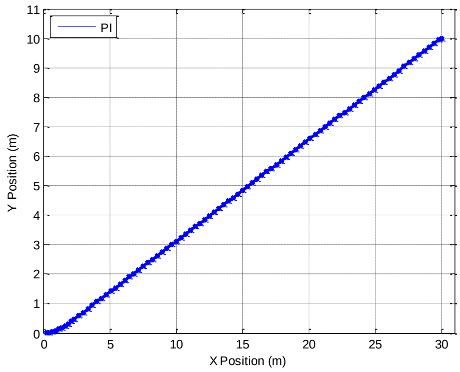

Figure D.15 Position of mobile robot using PI Controller

6. Proportional Integral (PI) Controller for 60x20y

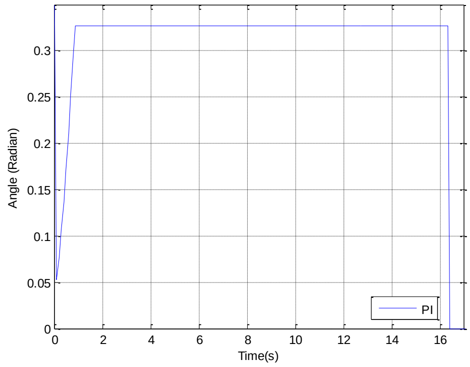

Figure D.16 Angle of orientation for PI Controller

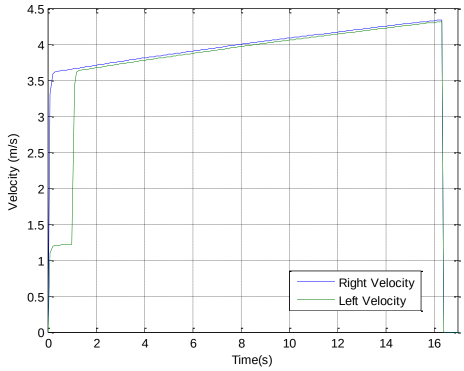

Figure D.17 Right and Left Velocity of mobile robot for PI Controller

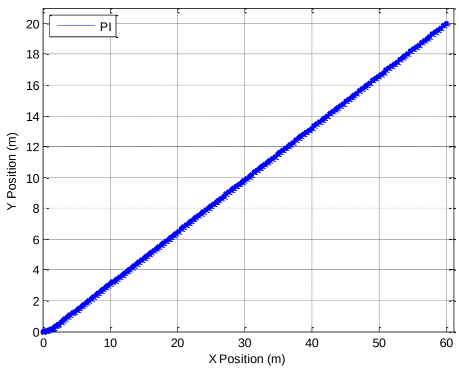

Figure D.18 Position of mobile robot using PI Controller

7. Proportional Integral Derivative (PID) Controller for 10x10y

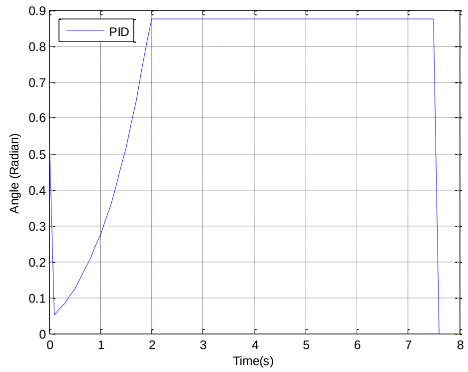

Figure D.19 Angle of orientation for PID Controller

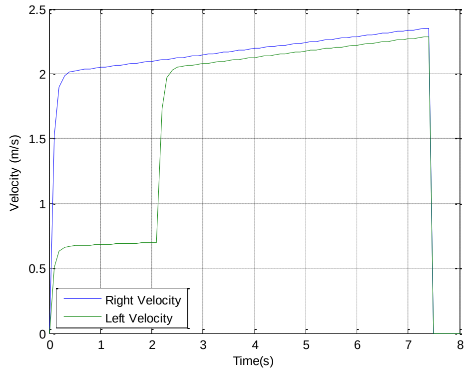

Figure D.20 Right and Left Velocity of mobile robot for PID Controller

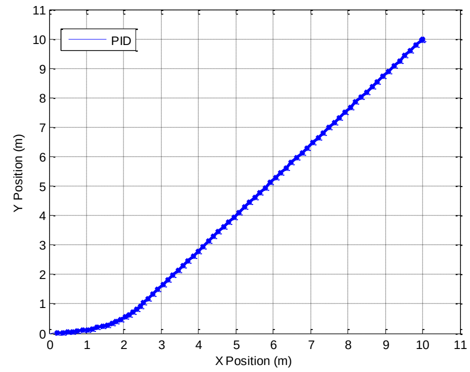

Figure D.21 Position of mobile robot using PID Controller

8. Proportional Integral Derivative (PID) Controller for 30x10y

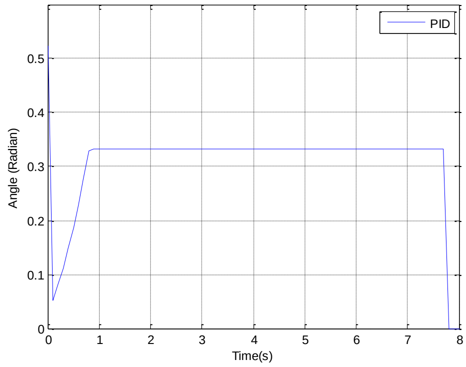

Figure D.22 Angle of orientation for PID Controller

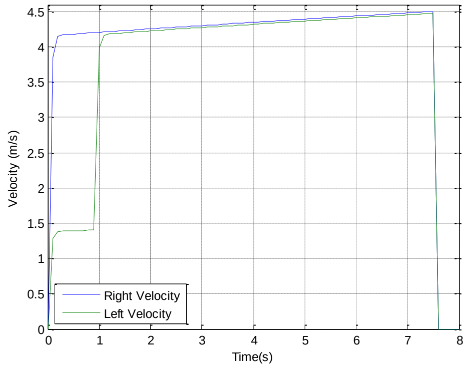

Figure D.23 Right and Left Velocity of mobile robot for PID Controller

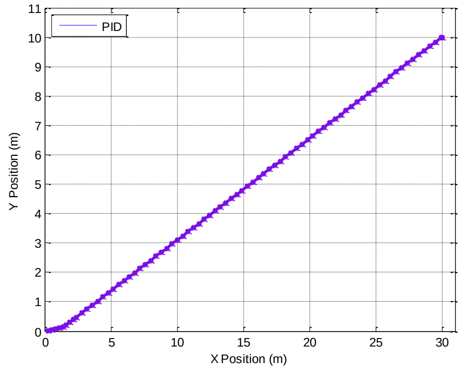

Figure D.24 Position of mobile robot using PID Controller

9. Proportional Integral Derivative (PID) Controller for 60x20y

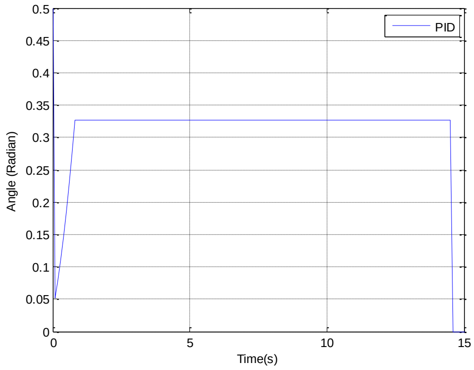

Figure D.25 Angle of orientation for PID Controller

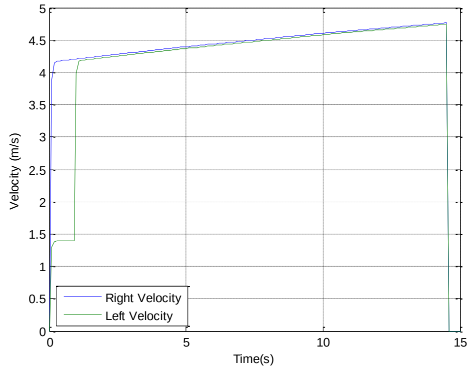

Figure D.26 Right and Left Velocity of mobile robot for PID Controller

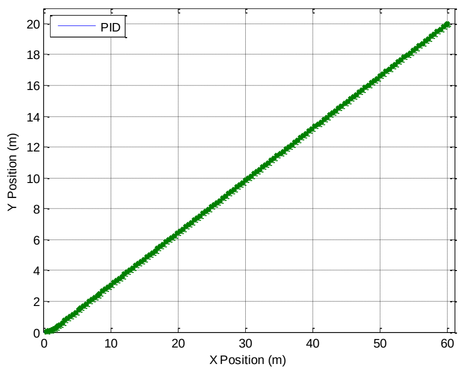

Figure D.27 Position of mobile robot using PID Controller

10. F3MP Controller for 10x10y

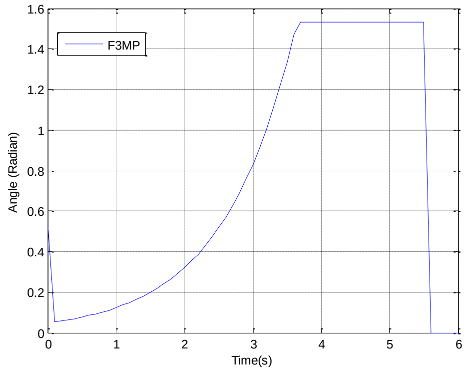

Figure D.28 Angle of orientation for F3MP Controller

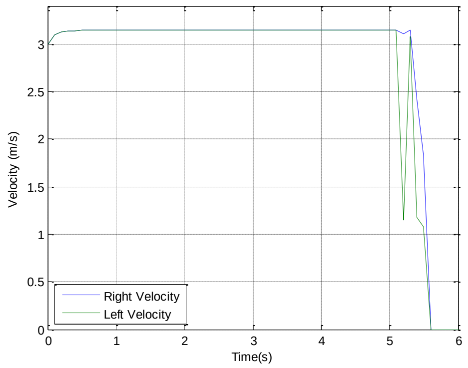

Figure D.29 Right and Left Velocity of mobile robot for F3MP Controller

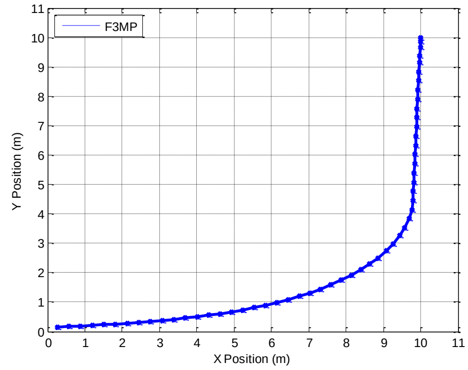

Figure D.30 Position of mobile robot using for F3M and P Controller

\

11. F3MP Controller for 30x10y

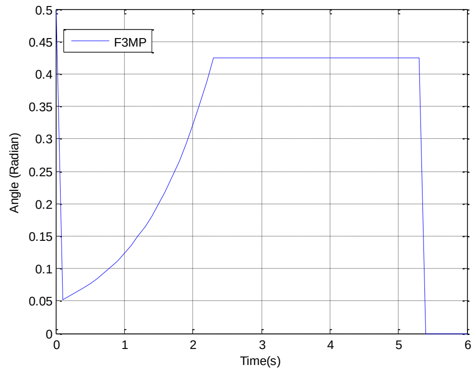

Figure D.31 Angle of orientation for F3MP Controller

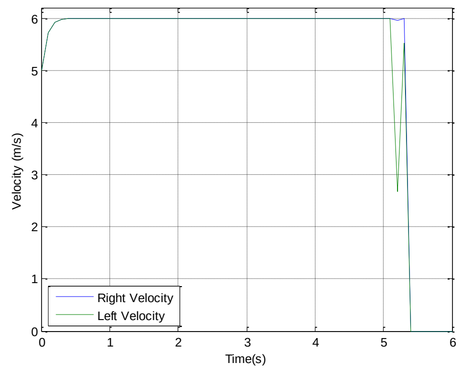

Figure D.32 Right and Left Velocity of mobile robot for F3MP Controller

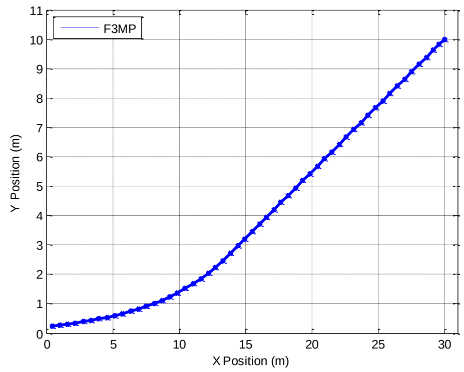

Figure D.33 Position of mobile robot using for F3MP Controller

12. F3MP Controller for 60x20y

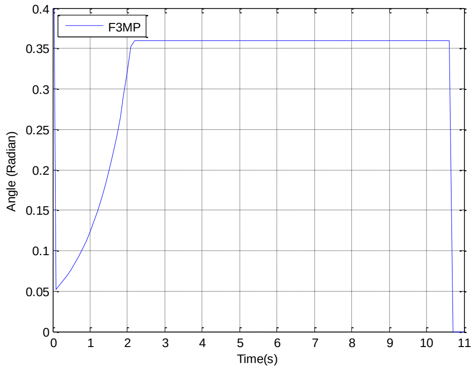

Figure D.34 Angle of orientation for F3M P Controller

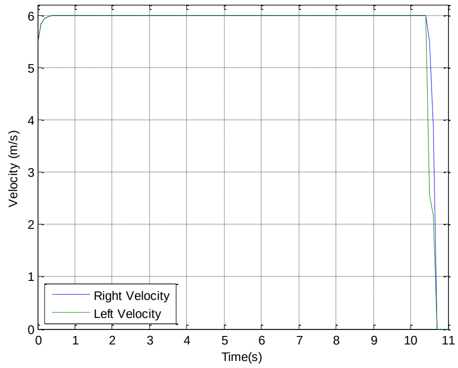

Figure D.35 Right and Left Velocity of mobile robot for F3MP Controller

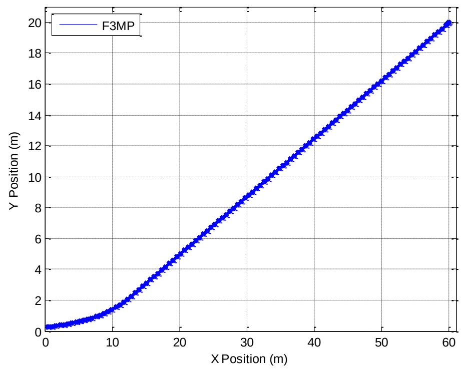

Figure D.36 Position of mobile robot using for F3MP Controller

13. F5MP Controller for 10x10y

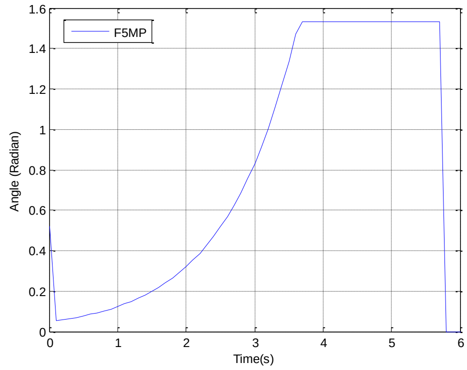

Figure D.37 Angle of orientation for F5MP Controller

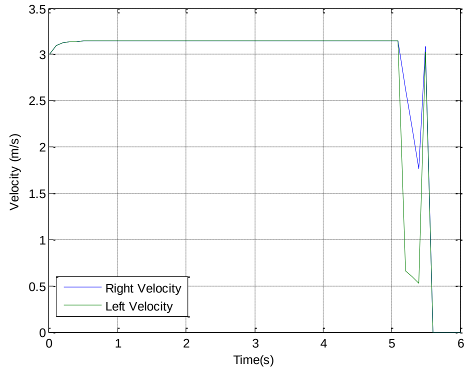

Figure D.38 Right and Left Velocity of mobile robot for F5MP Controller

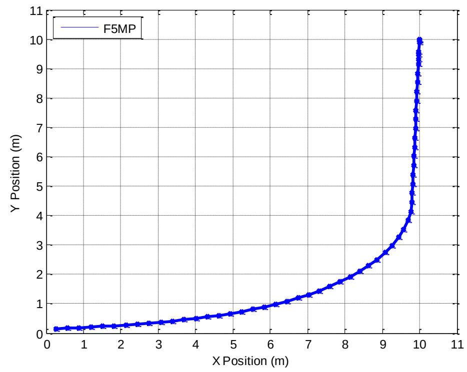

Figure D.39 Position of mobile robot using for F5M P Controller

14. F5MP Controller for 30x0y

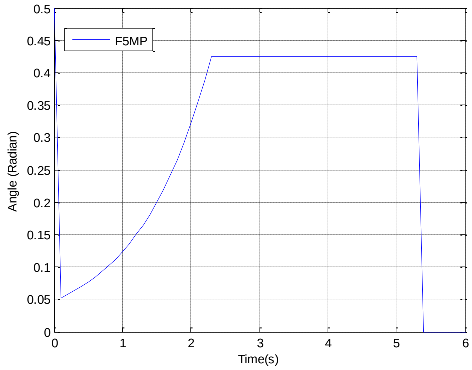

Figure D.40 Angle of orientation for F5MP Controller

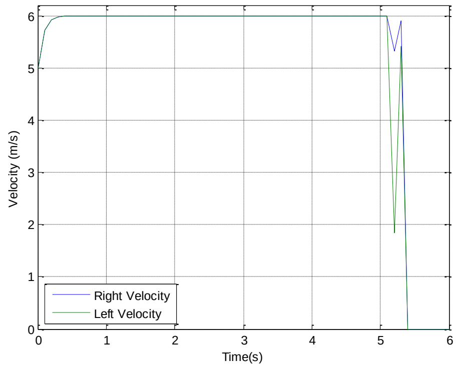

Figure D.41 Right and Left Velocity of mobile robot for F5MP Controller

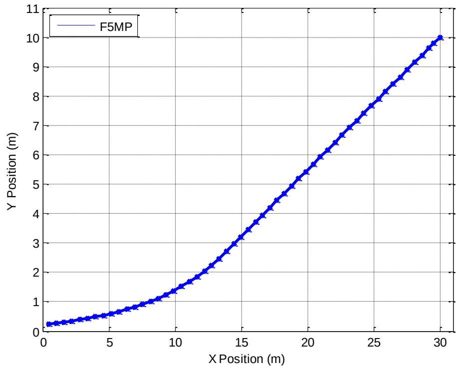

Figure D.42 Position of mobile robot using for F5MP Controller

15. F5MP Controller for 60x20y

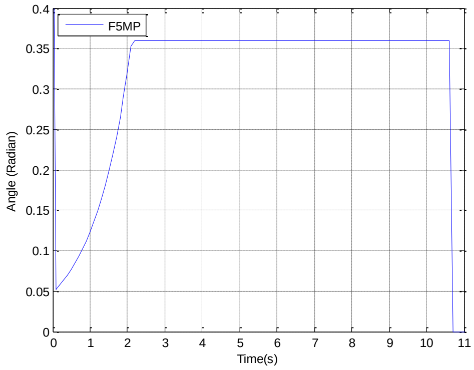

Figure D.43 Angle of orientation for F5M P Controller

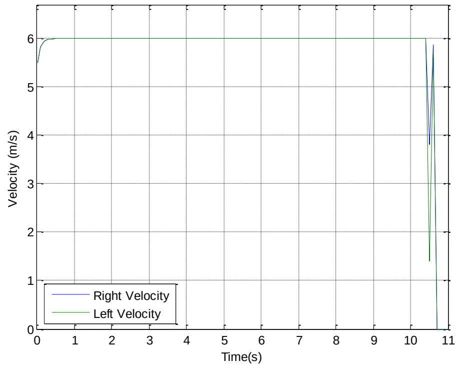

Figure D.44 Right and Left Velocity of mobile robot for F5MP Controller

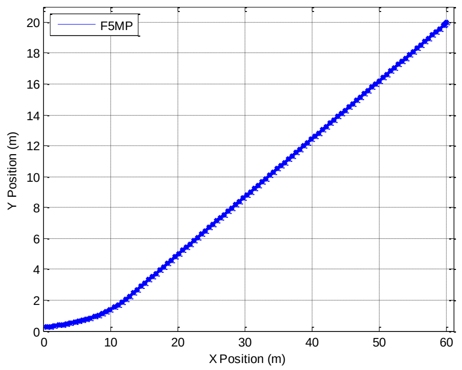

Figure D.45 Position of mobile robot using for F5M P Controller

16. F7MP Controller for 10x10y

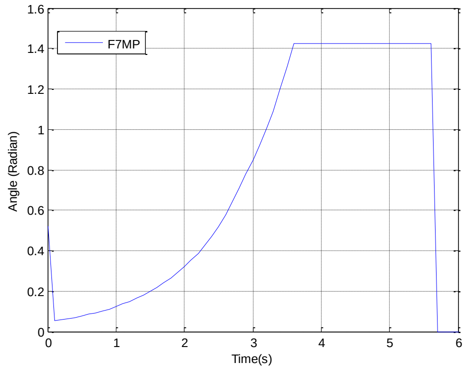

Figure D.46 Angle of orientation for F7MP Controller

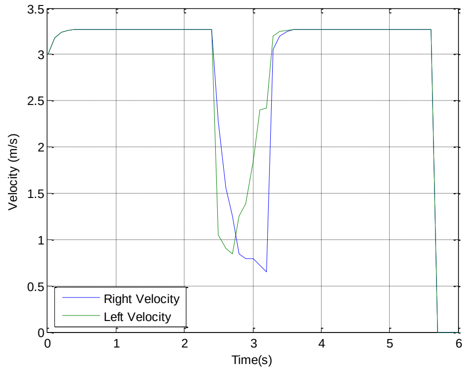

Figure D.47 Right and Left Velocity of mobile robot for F7MP Controller

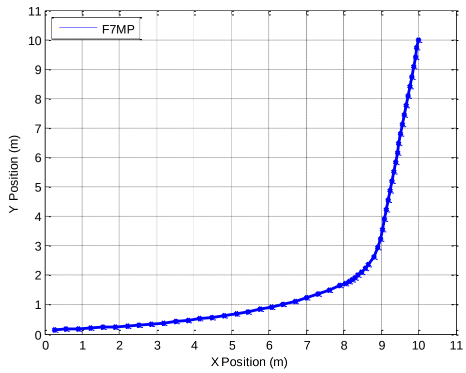

Figure D.48 Position of mobile robot using for F7MP Controller

17. F7MP Controller for 30x10y

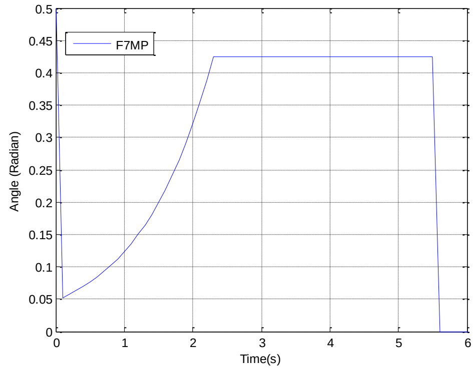

Figure D.49 Angle of orientation for F7MP Controller

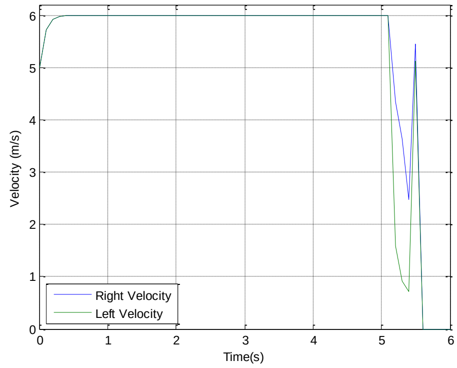

Figure D.50 Right and Left Velocity of mobile robot for F7MP Controller

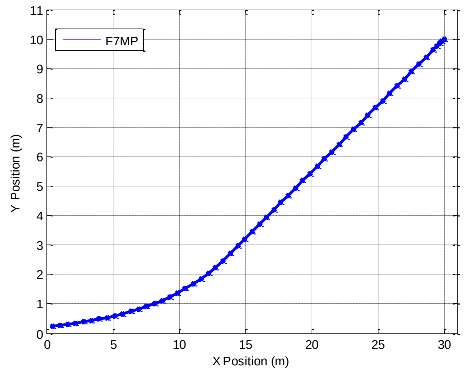

Figure D.51 Position of mobile robot using for F7MP Controller

18. F7MP Controller for 60x20y

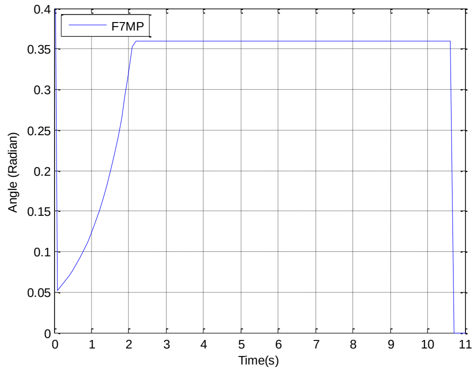

Figure D.52 Angle of orientation for F7MP Controller

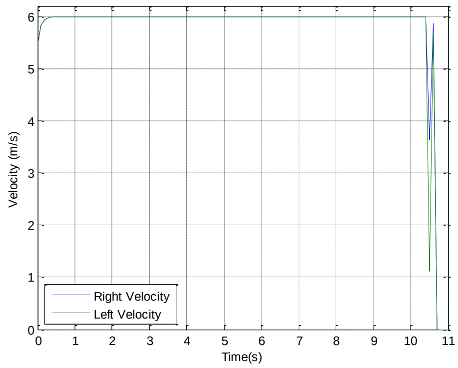

Figure D.53 Right and Left Velocity of mobile robot for F7MP Controller

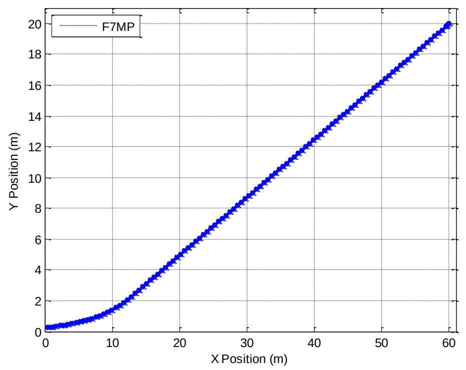

Figure D.54 Position of mobile robot using for F7MP Controller

19. F5MPI Controller for 10x10y

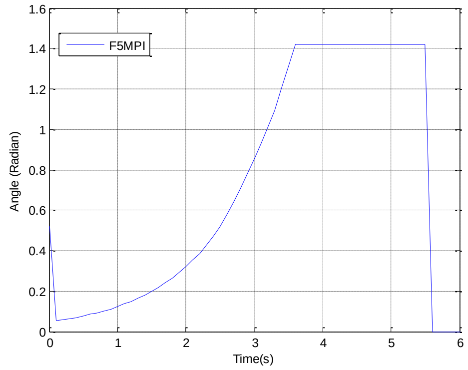

Figure D.55 Angle of orientation for F5MPI Controller

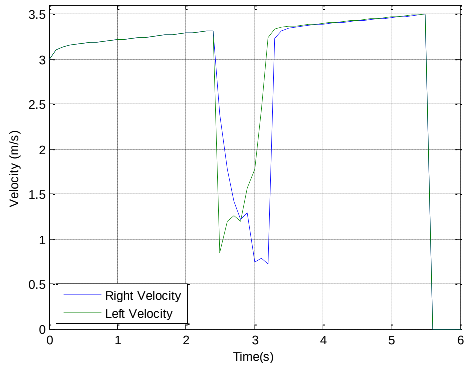

Figure D.56 Right and Left Velocity of mobile robot for F5MPI Controller

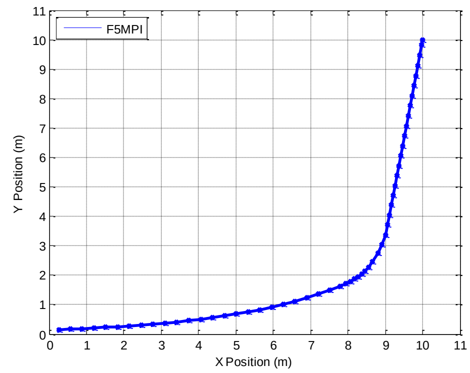

Figure D.57 Position of mobile robot using for F5MPI Controller

20. F5MPI Controller for 30x10y

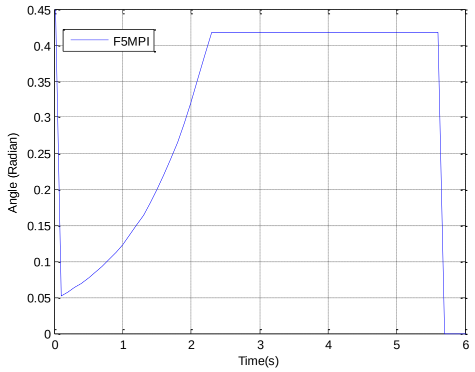

Figure D.58 Angle of orientation for F5MPI Controller

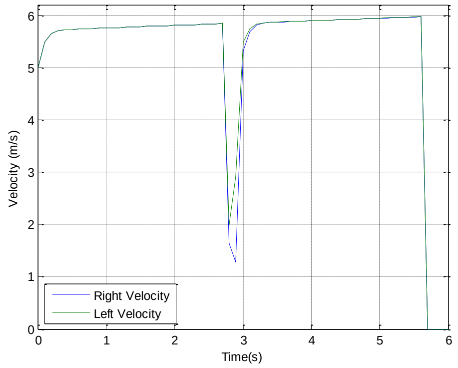

Figure D.59 Right and Left Velocity of mobile robot for F5MPI Controller

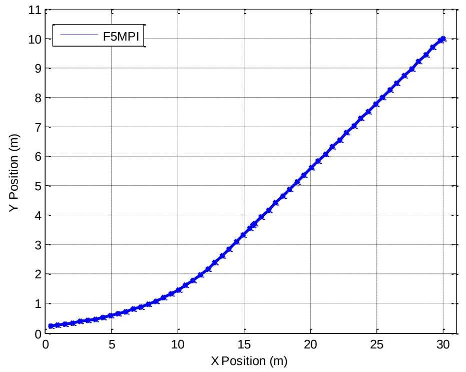

Figure D.60 Position of mobile robot using for F5MPI Controller

21. F5MPI Controller for 60x20y

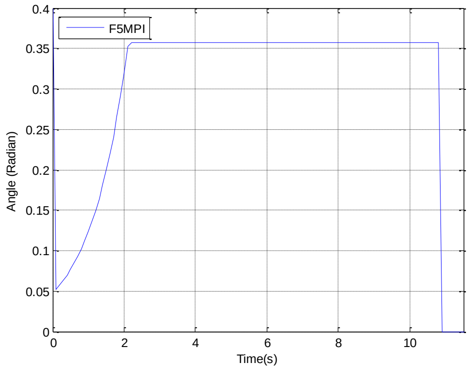

Figure D.61 Angle of orientation for F5MPI Controller

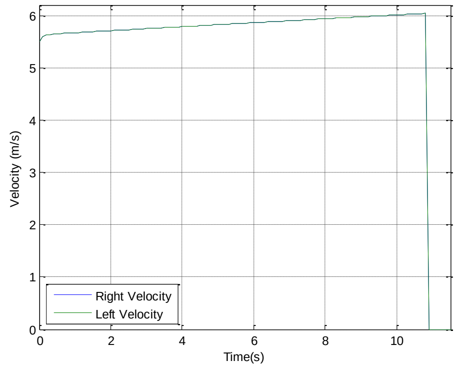

Figure D.62 Right and Left Velocity of mobile robot for F5MPI Controller

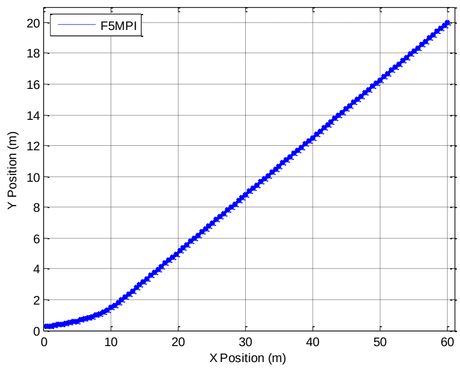

Figure D.63 Position of mobile robot using for F5MPI Controller

22. F7MPI Controller for 10x10y

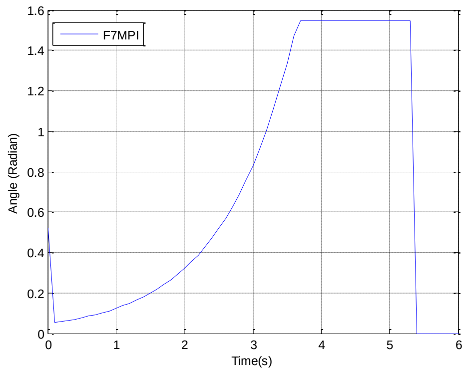

Figure D.64 Angle of orientation for F7MPI Controller

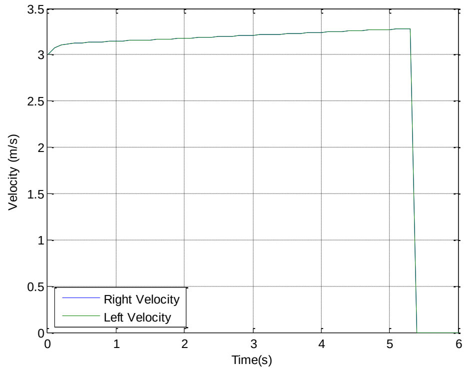

Figure D.65 Right and Left Velocity of mobile robot for F7MPI Controller

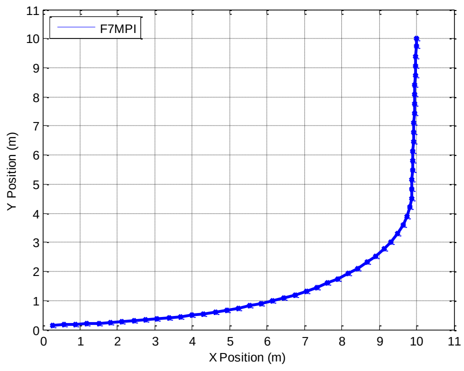

Figure D.66 Position of mobile robot using for F7MPI Controller

23. F7MPI Controller for 30x10y

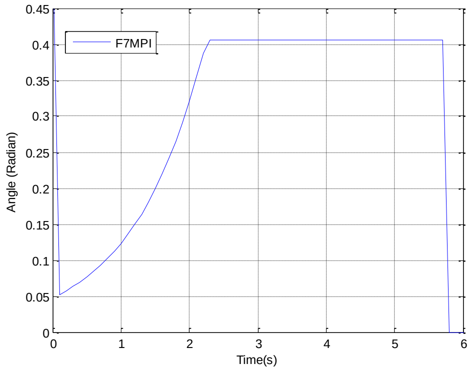

Figure D.67 Angle of orientation for F7MPI Controller

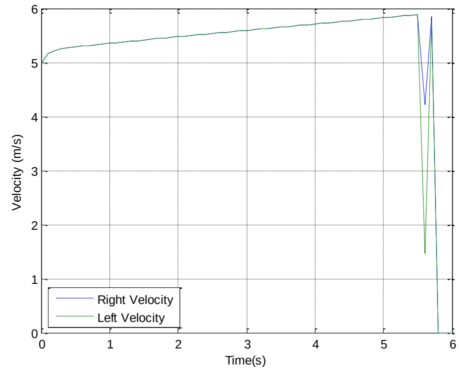

Figure D.68 Right and Left Velocity of mobile robot for F7MPI Controller

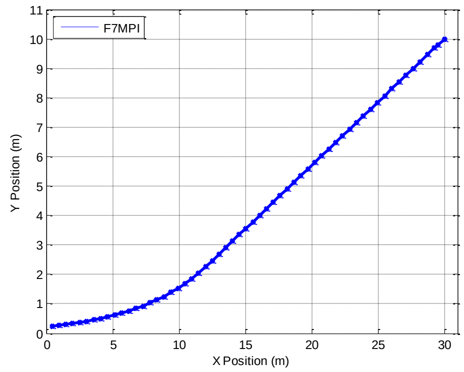

Figure D.69 Position of mobile robot using for F7MPI Controller

24. F7MPI Controller for 60x20y

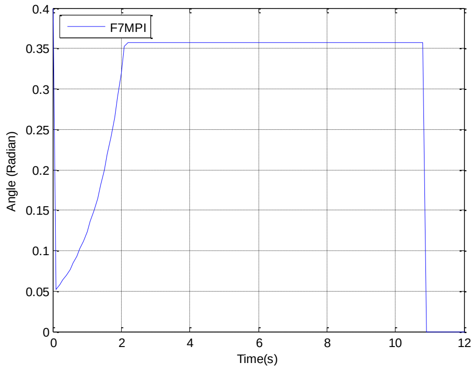

Figure D.70 Angle of orientation for F7MPI Controller

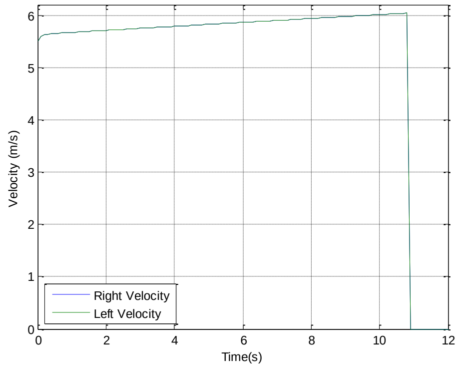

Figure D.71 Right and Left Velocity of mobile robot for F7MPI Controller

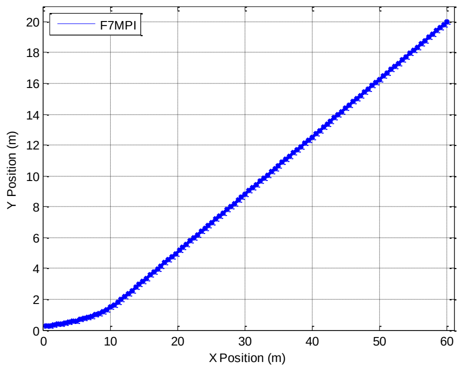

Figure D.72 Position of mobile robot using for F7MPI Controller

APPENDIX E

RESULTS ANT COLONY OPTIMIZATION PATH

1. 1 obstacle simulation 1

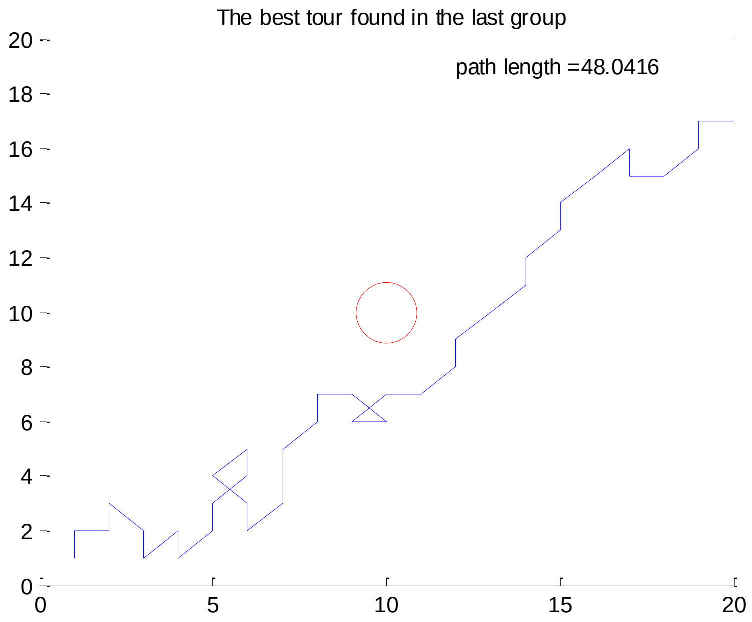

Figure E.1 :The best tour found at the 1 obstacle

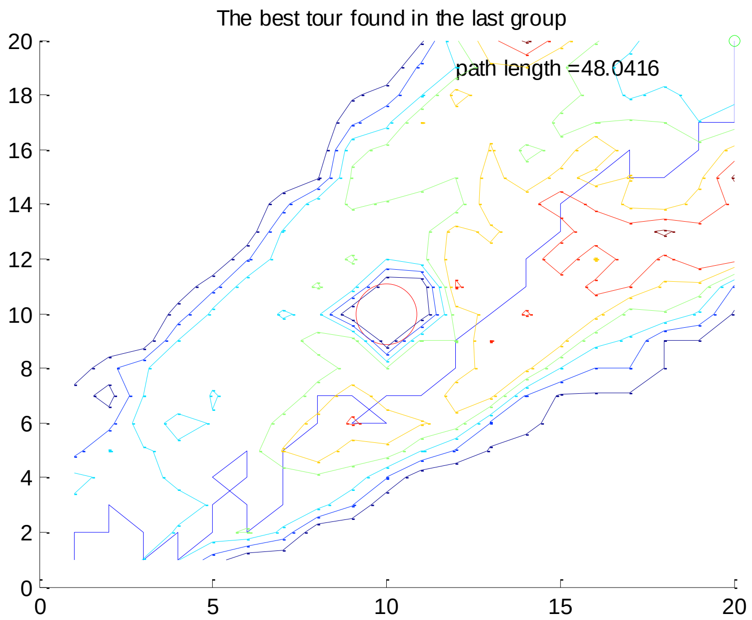

Figure E.2 Deposited pheromone in 2D contour map

2. 1 obstacle simulation 2

Figure E.3 :The best tour found at the 10 obstacle

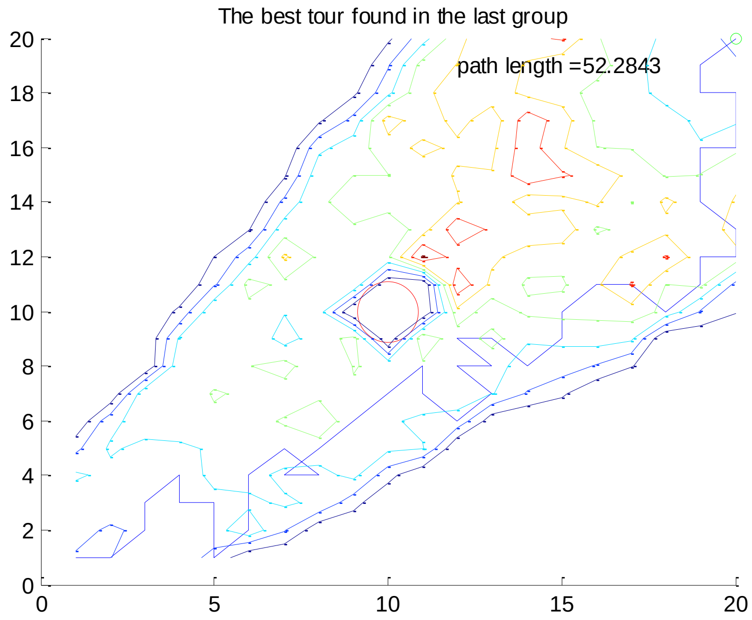

Figure E.4 Deposited pheromone in 2D contour map

3. 1 obstacle simulation 3

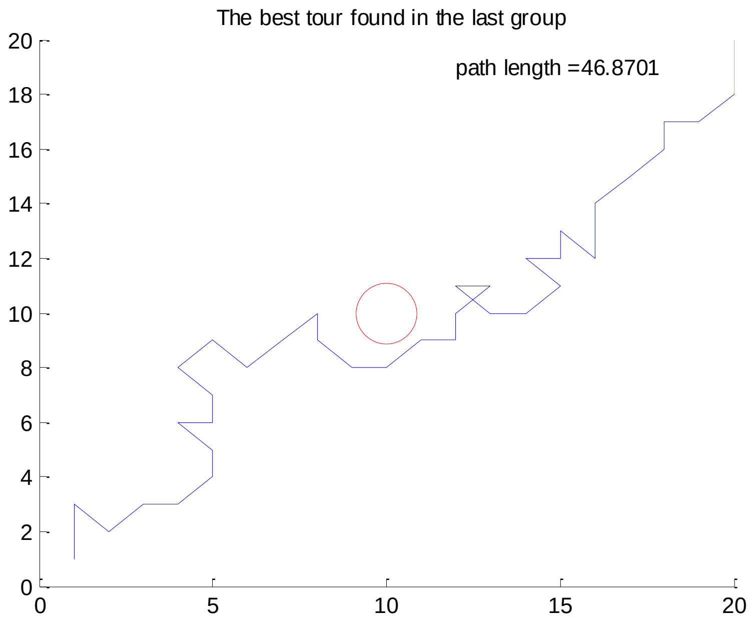

Figure E.5 :The best tour found at the 10 Obstacle

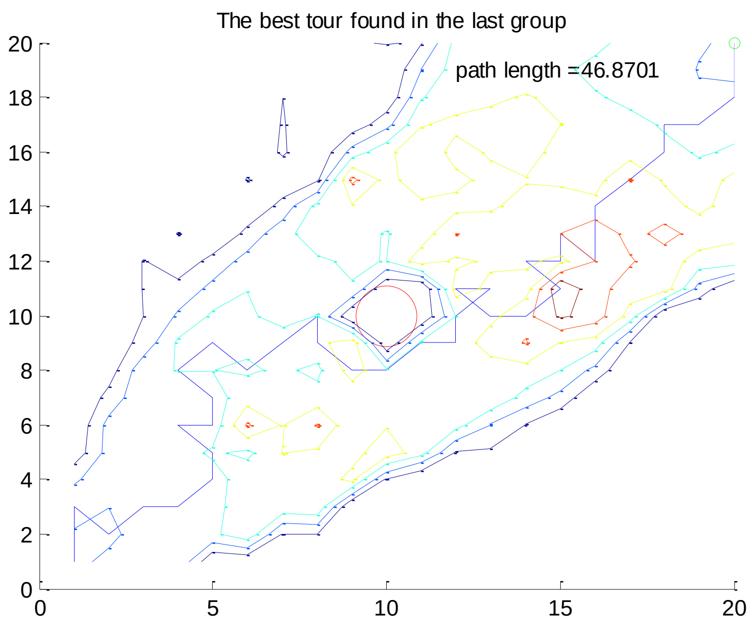

Figure E.6 Deposited pheromone in 2D contour map

4. 1 obstacle simulation 4

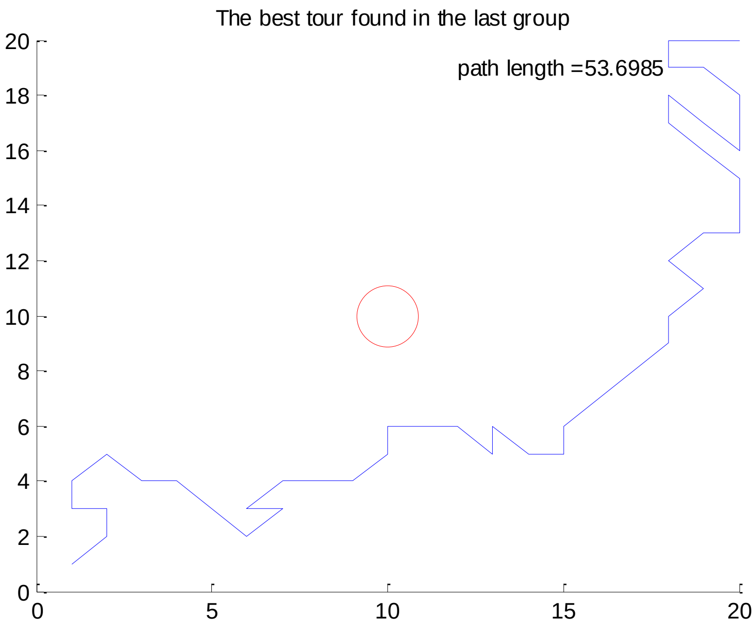

Figure E.7 :The best tour found at the 1 obstacle

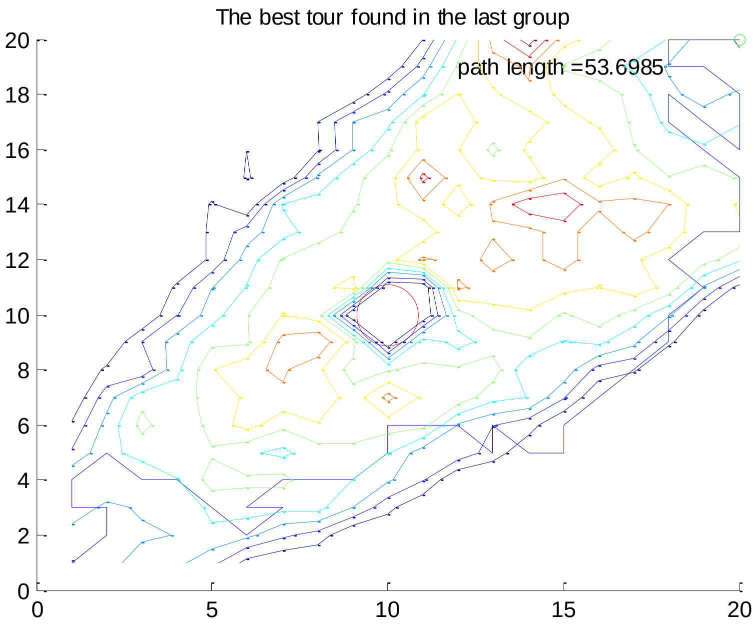

Figure E.8 Deposited pheromone in 2D contour map

5. 1 obstacle simulation 5

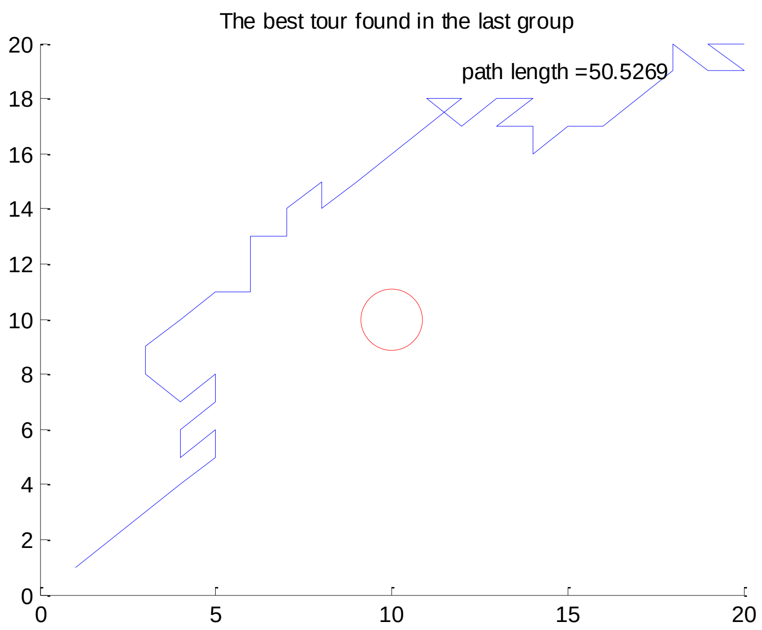

Figure E.9 :The best tour found at the 1 obstacle

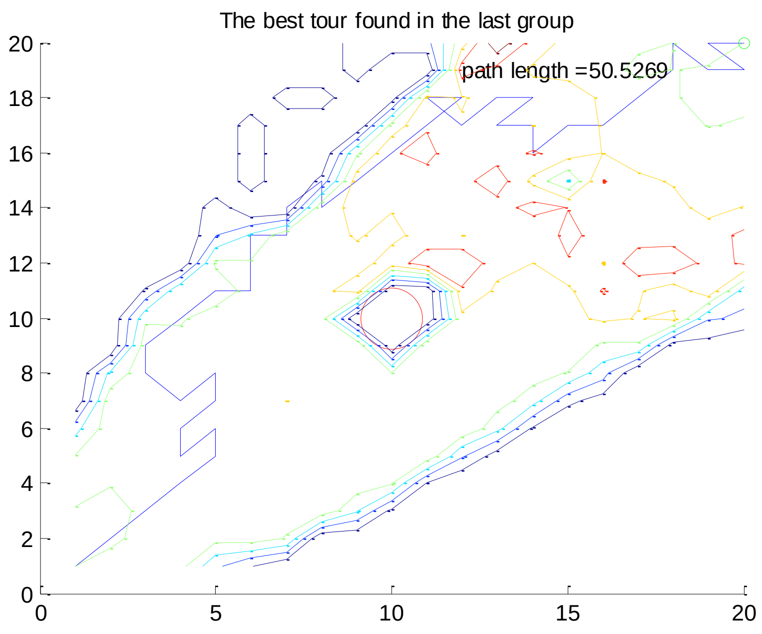

Figure E.10 Deposited pheromone in 2D contour map

6. 2 obstacle simulation 1

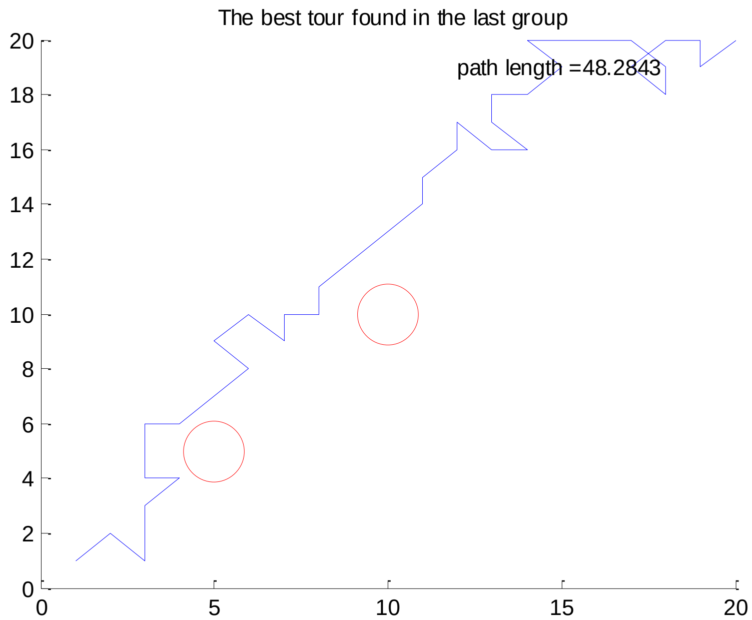

Figure E.11 :The best tour found at the 2 obstacle

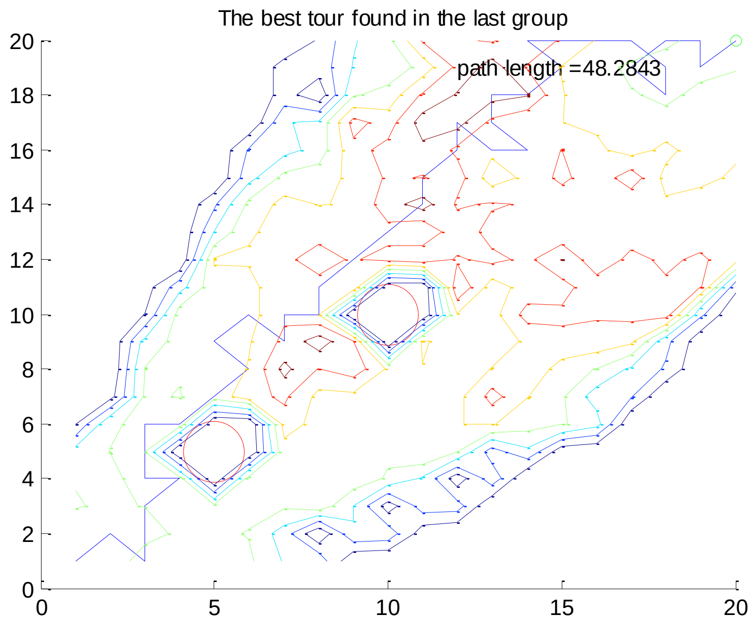

Figure E.12 Deposited pheromone in 2D contour map

7. 2 obstacle simulation 2

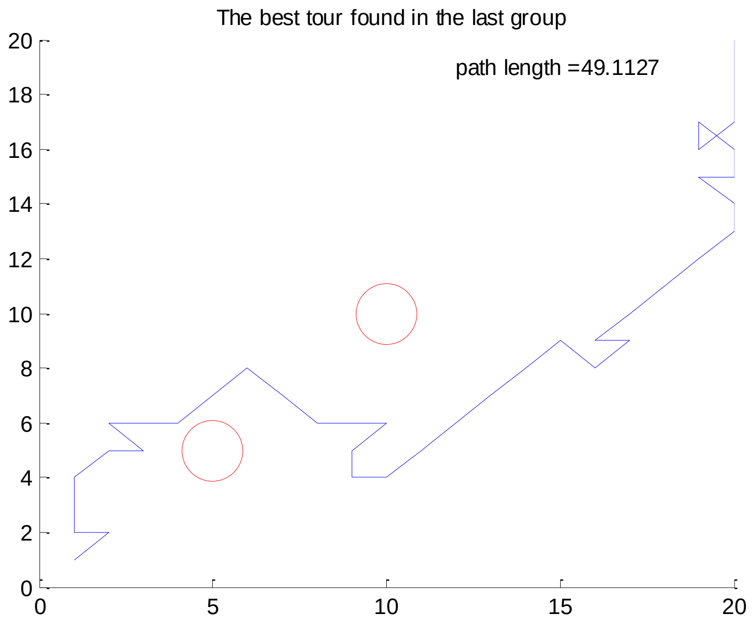

Figure E.13 :The best tour found at the 2 obstacle

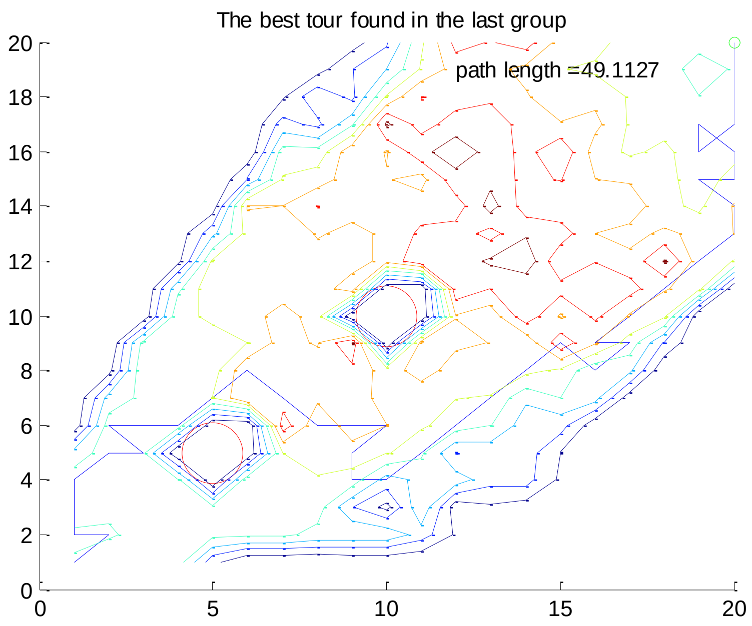

Figure E.14 Deposited pheromone in 2D contour map

8. 2 obstacle simulation 3

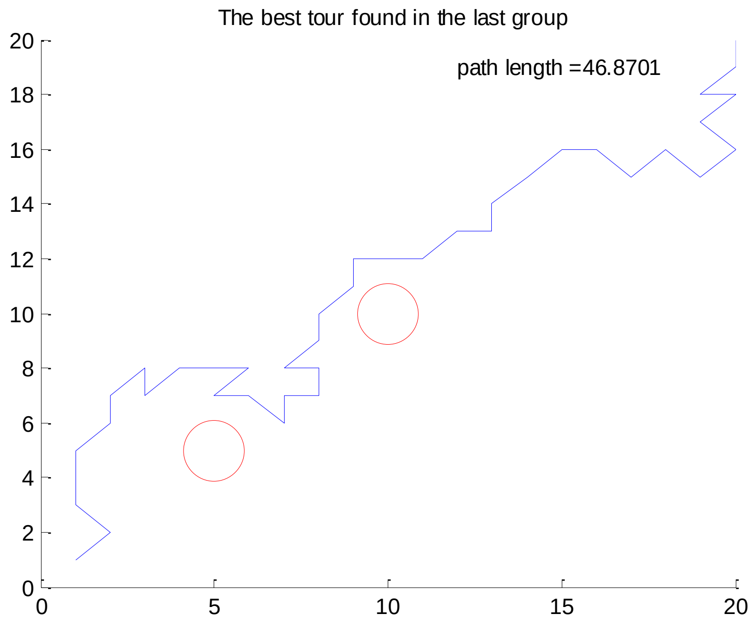

Figure E.15 :The best tour found at the 2 obstacle

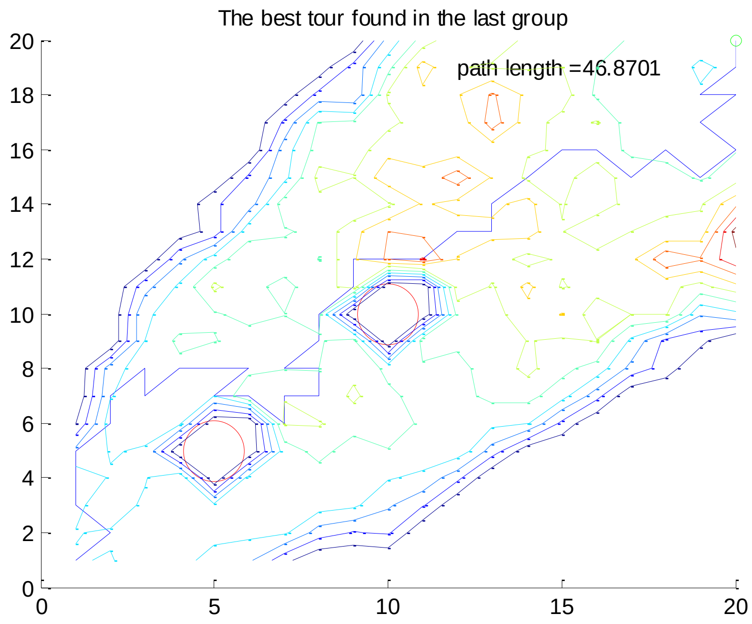

Figure E.16 Deposited pheromone in 2D contour map

9. 2 obstacle simulation 4

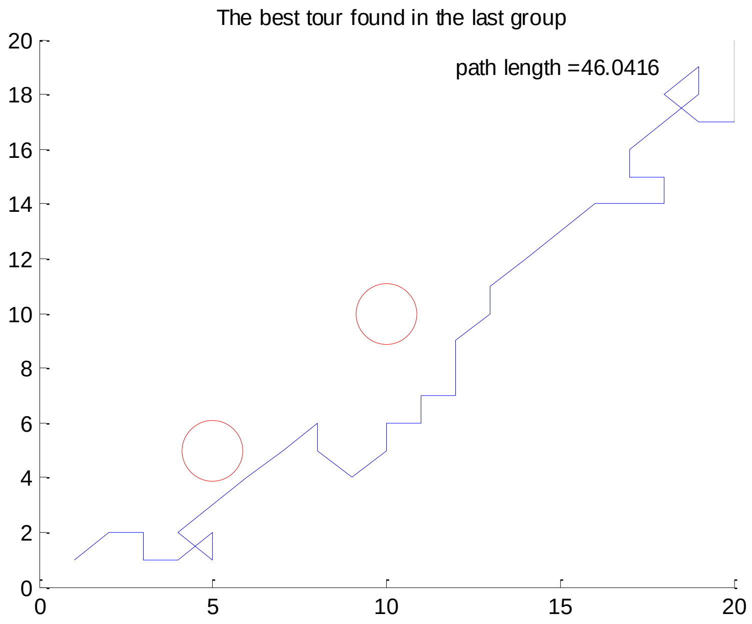

Figure E.17 :The best tour found at the 2 obstacle

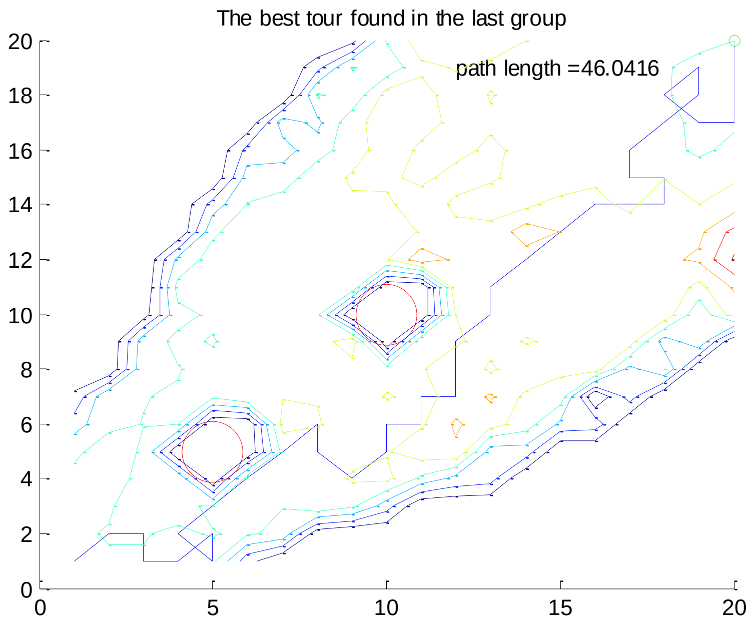

Figure E.18 Deposited pheromone in 2D contour map

10. 2 obstacle simulation 5

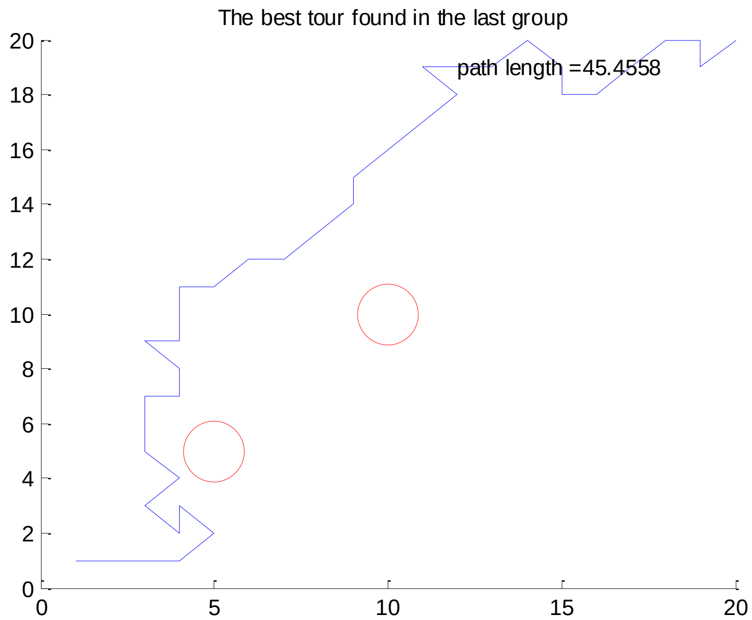

Figure E.19:The best tour found at the 2 obstacle

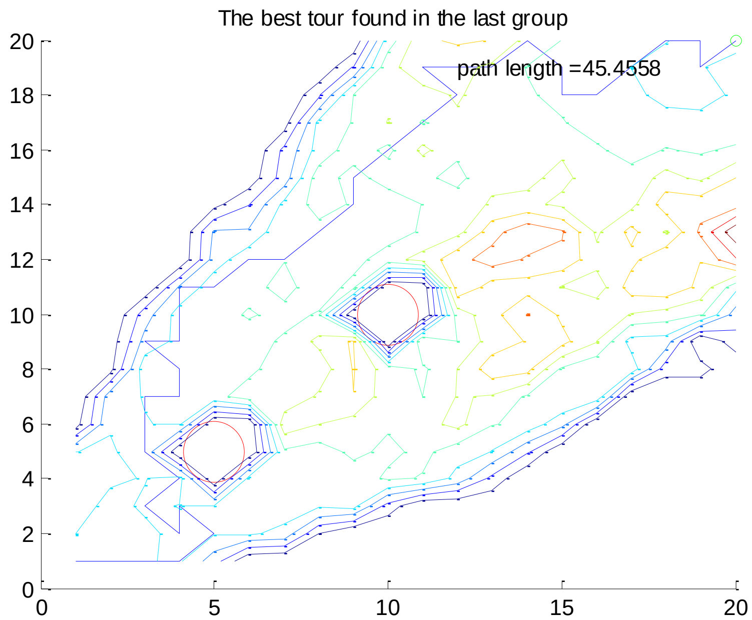

Figure E.20 Deposited pheromone in 2D contour map

11. 3 obstacle simulation 1

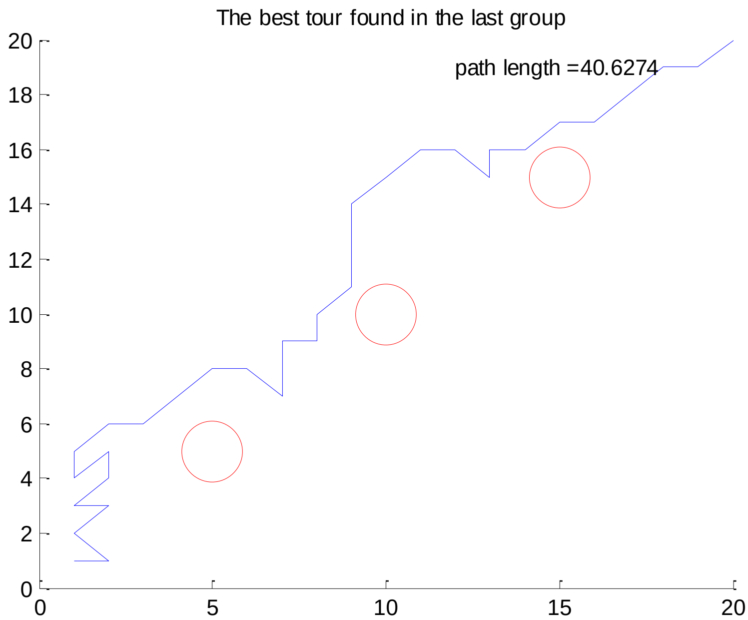

Figure E.21 :The best tour found at the 3 obstacle

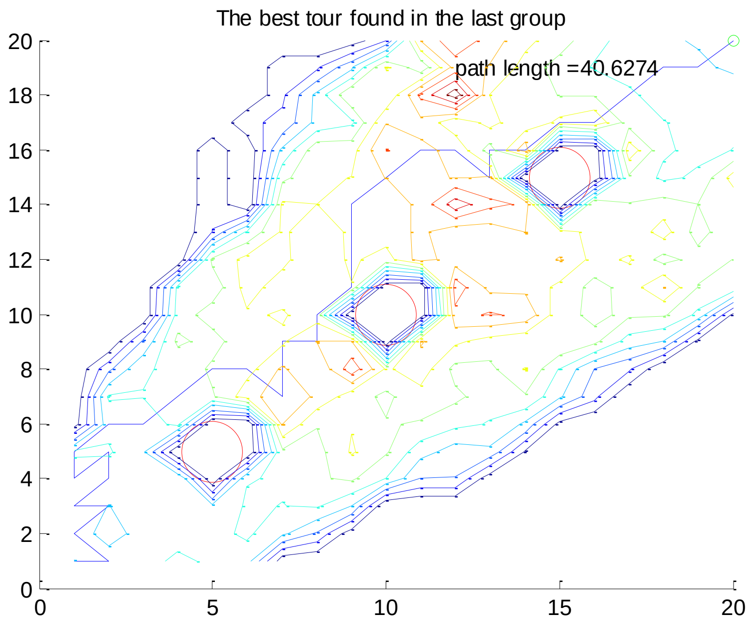

Figure E.22 Deposited pheromone in 2D contour map

12. 3 obstacle simulation 2

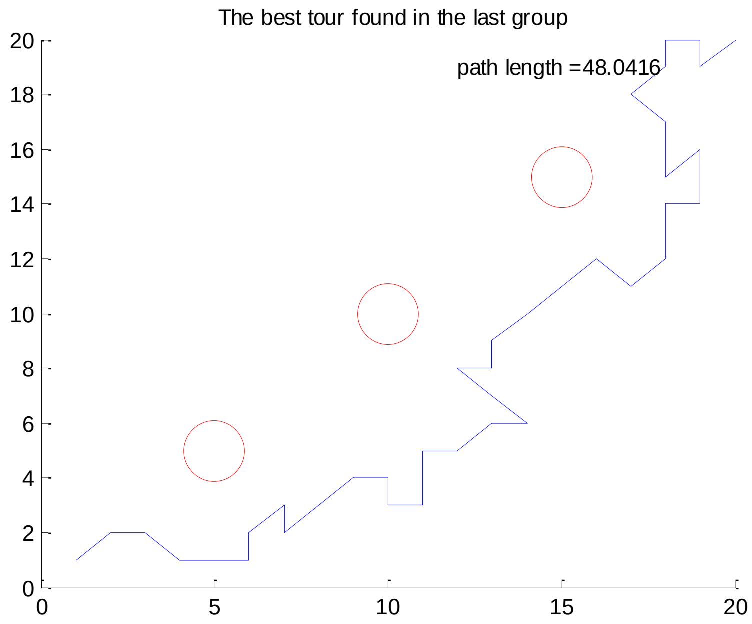

Figure E.23 :The best tour found at the 3 obstacle

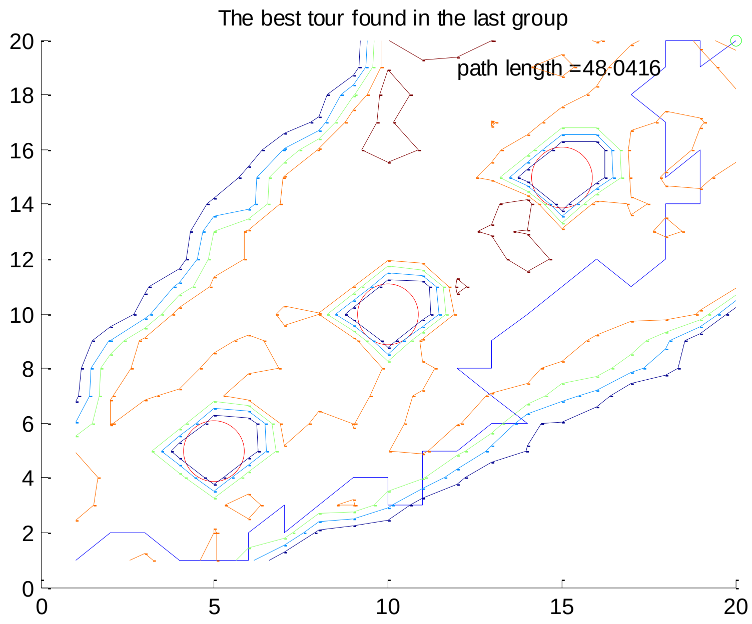

Figure E.24 Deposited pheromone in 2D contour map

13. 3 obstacle simulation 3

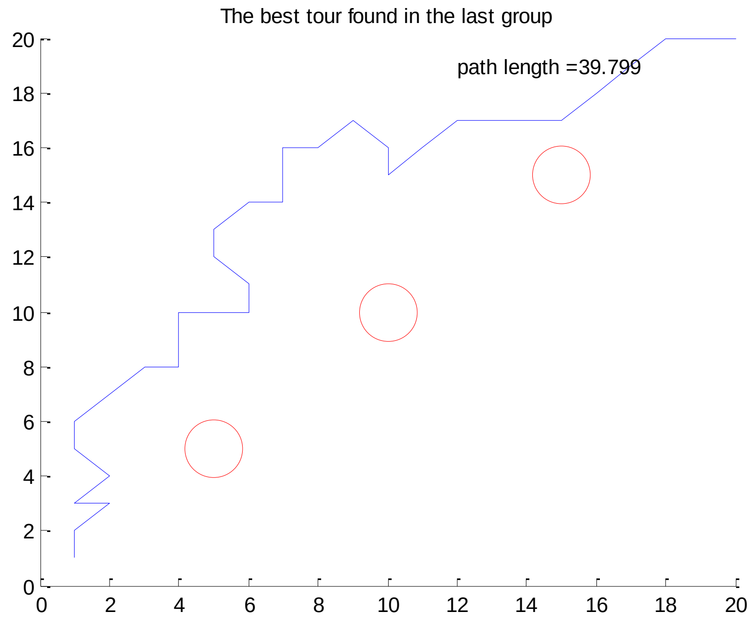

Figure E.25 :The best tour found at the 3 obstacle

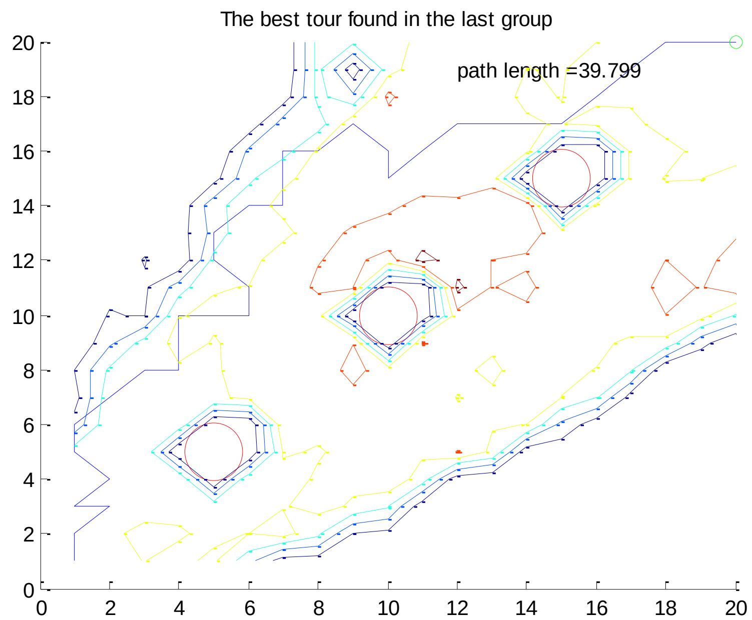

Figure E.26 Deposited pheromone in 2D contour map

14. 3 obstacle simulation 4

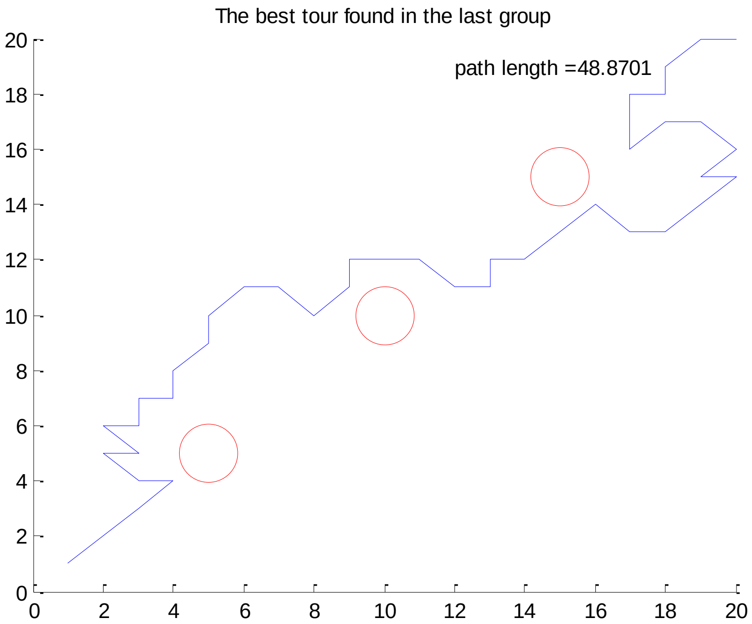

Figure E.27 :The best tour found at the 3 obstacle

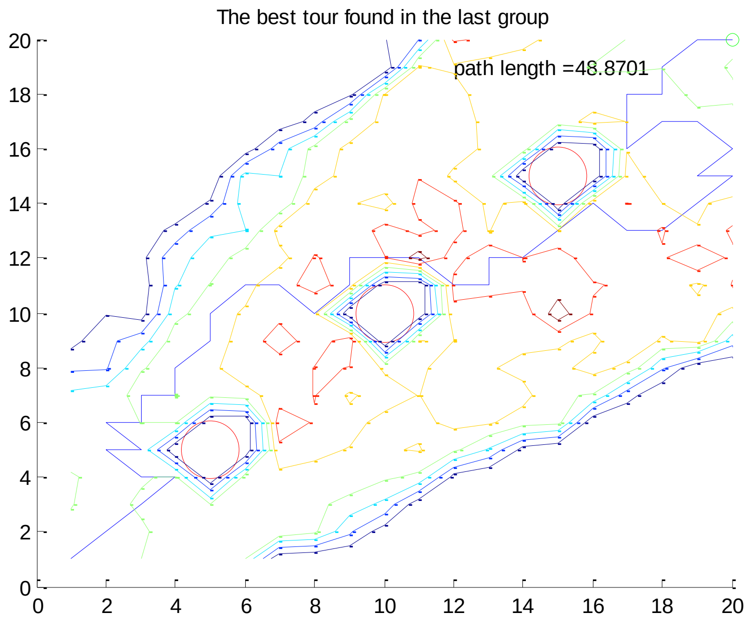

Figure E.28 Deposited pheromone in 2D contour map

15. 3 obstacle simulation 5

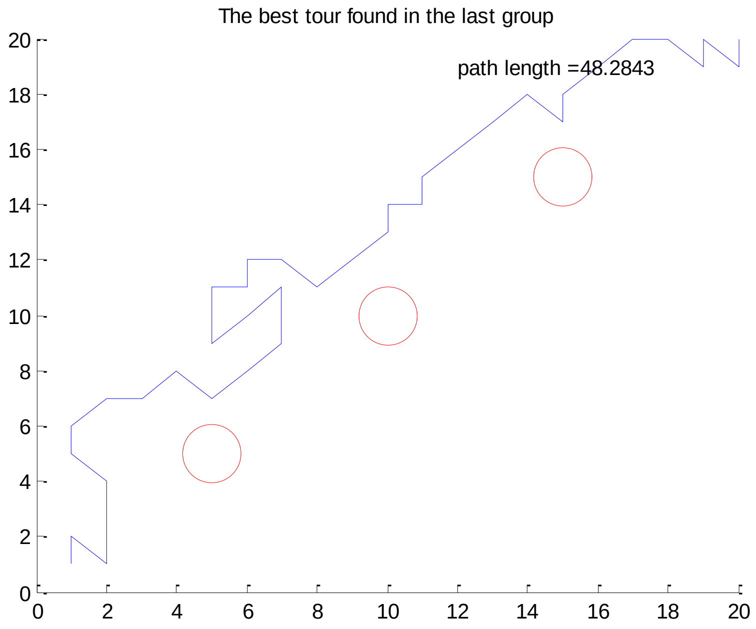

Figure E.29 :The best tour found at the 3 obstacle

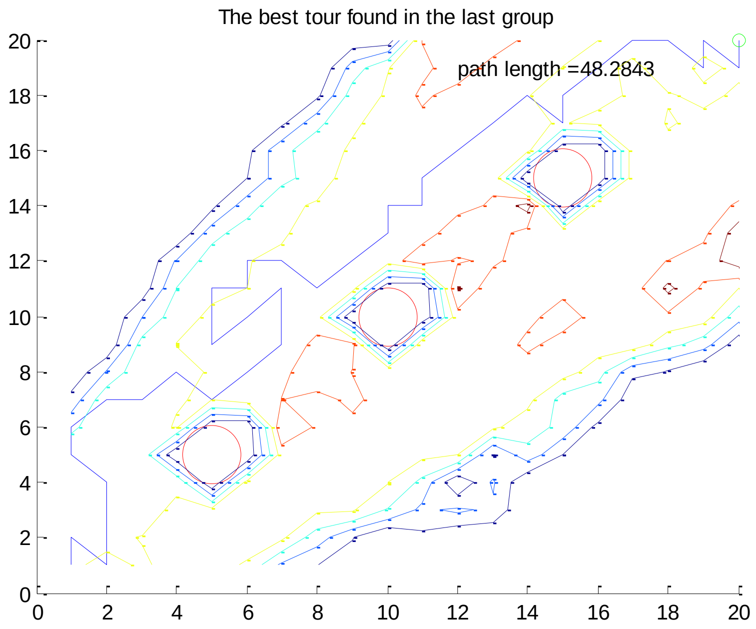

Figure E.30 Deposited pheromone in 2D contour map

16. 4 obstacle simulation 1

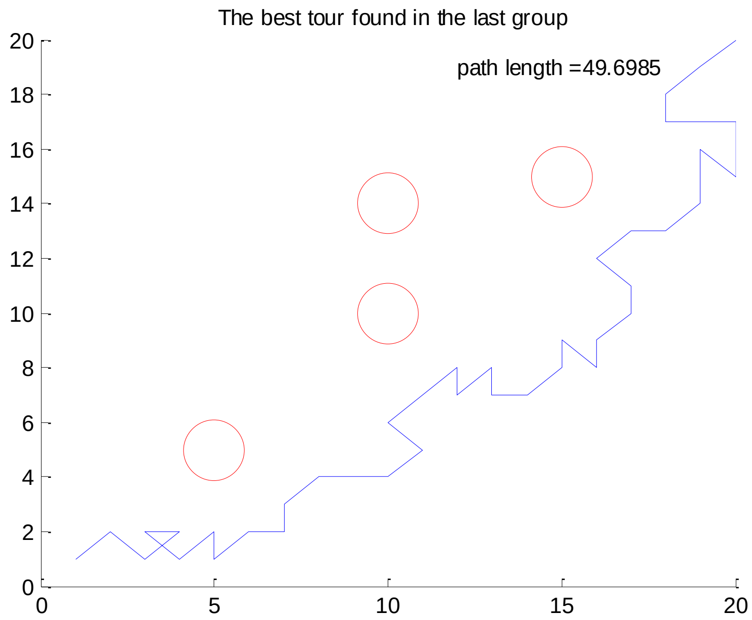

Figure E.31 :The best tour found at the 4 obstacle

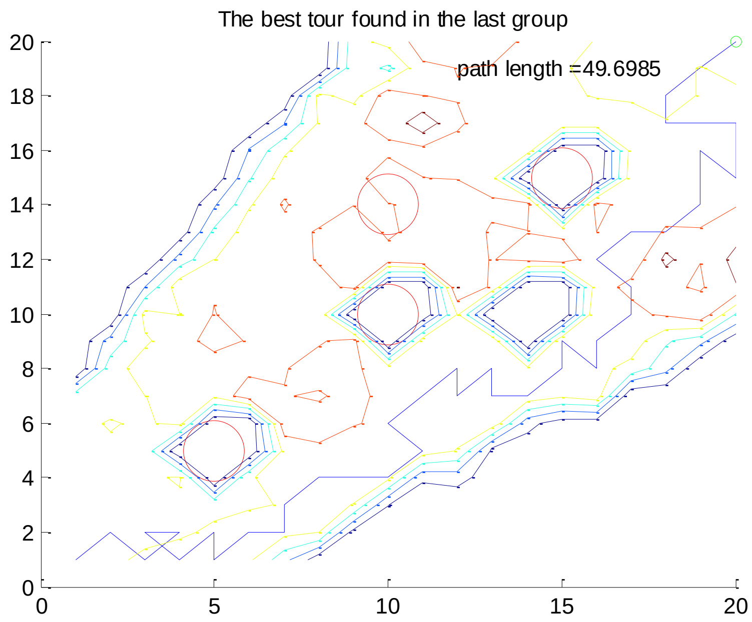

Figure E.32 Deposited pheromone in 2D contour map

17. 4 obstacle simulation 2

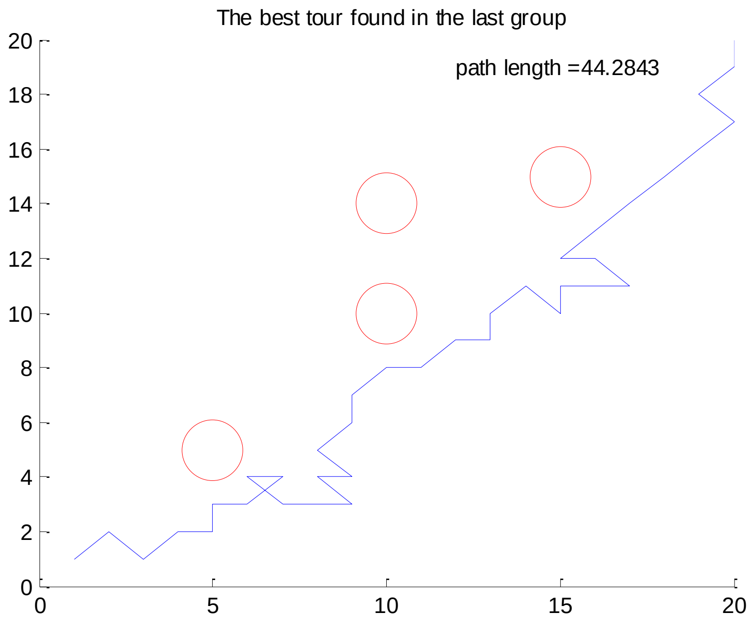

Figure E.33 :The best tour found at the 4 obstacle

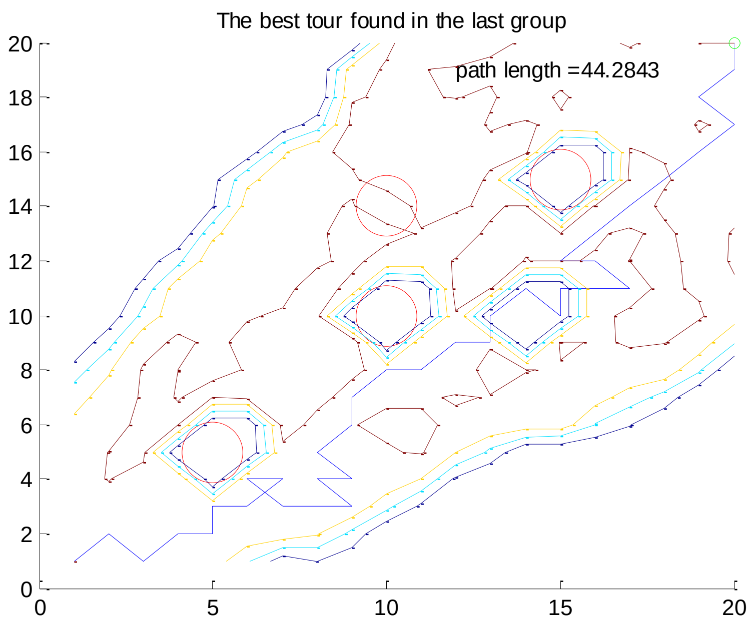

Figure E.34 Deposited pheromone in 2D contour map

18. 4 obstacle simulation 3

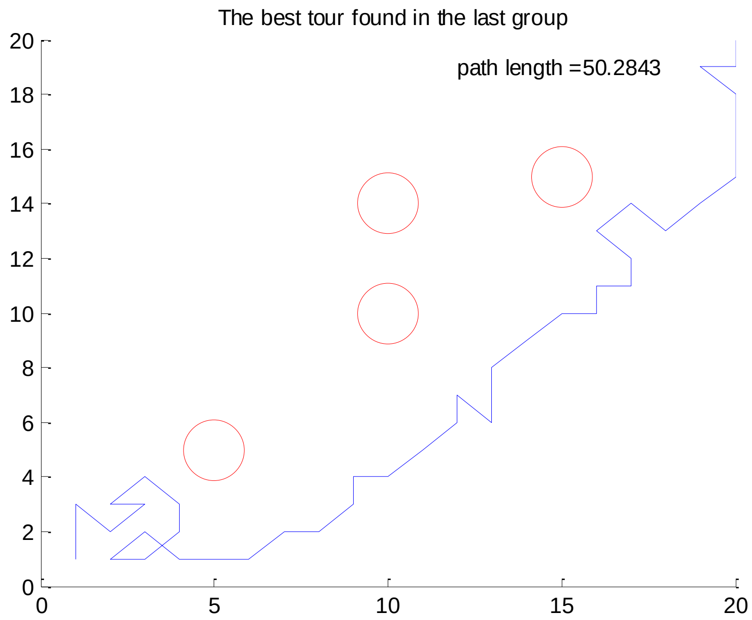

Figure E.35 :The best tour found at the 4 obstacle

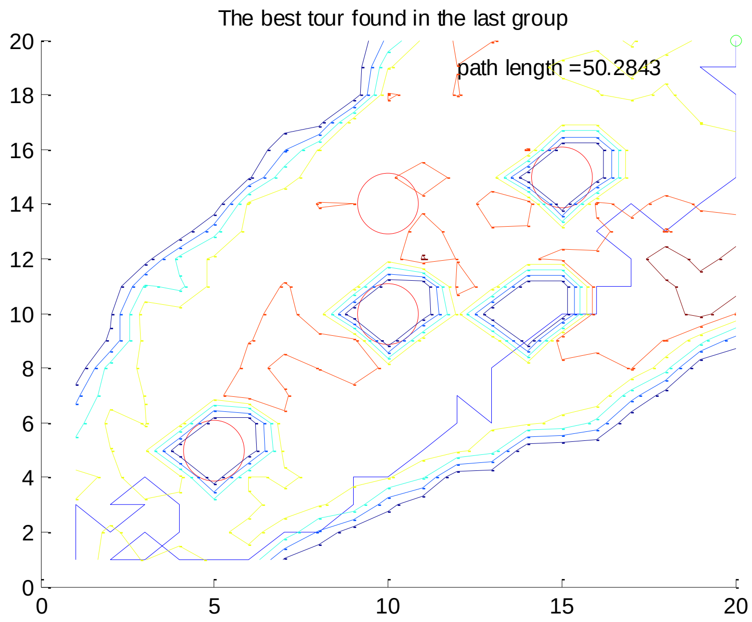

Figure E.36 Deposited pheromone in 2D contour map

19. 4 obstacle simulation 4

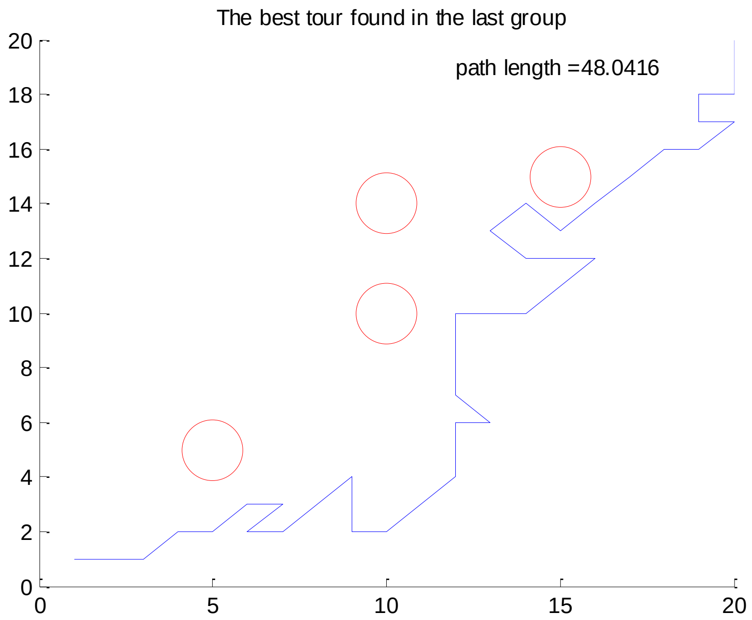

Figure E.37 :The best tour found at the 4 obstacle

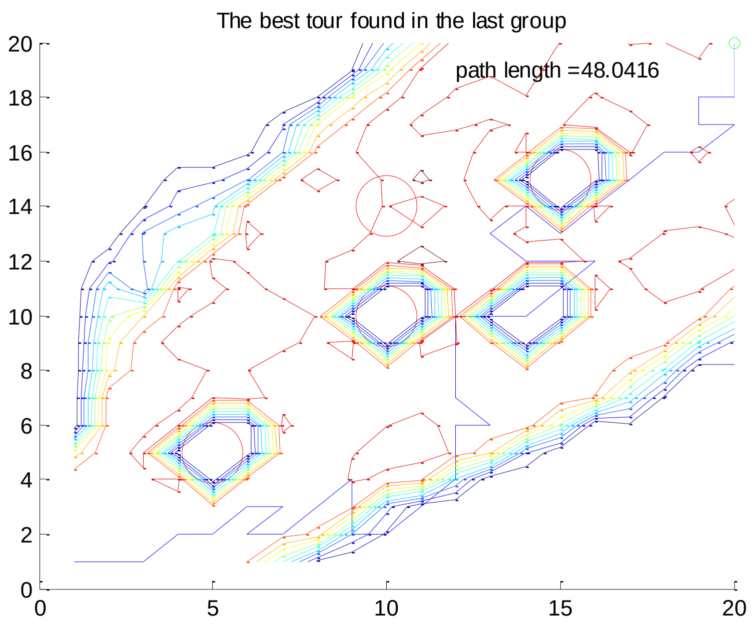

Figure E.38 Deposited pheromone in 2D contour map

20. 4 obstacle simulation 5

Figure E.39 :The best tour found at the 4 obstacle

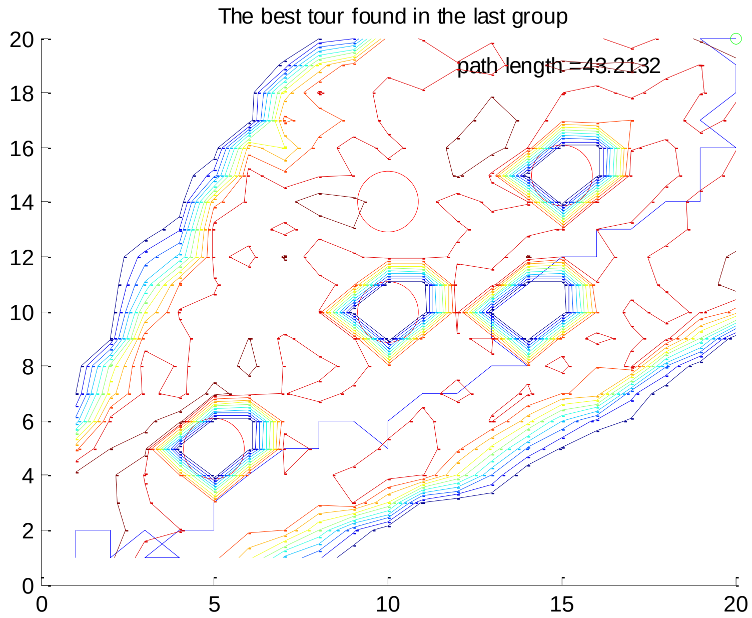

Figure E.40 Deposited pheromone in 2D contour map

21. 5 obstacle simulation 1

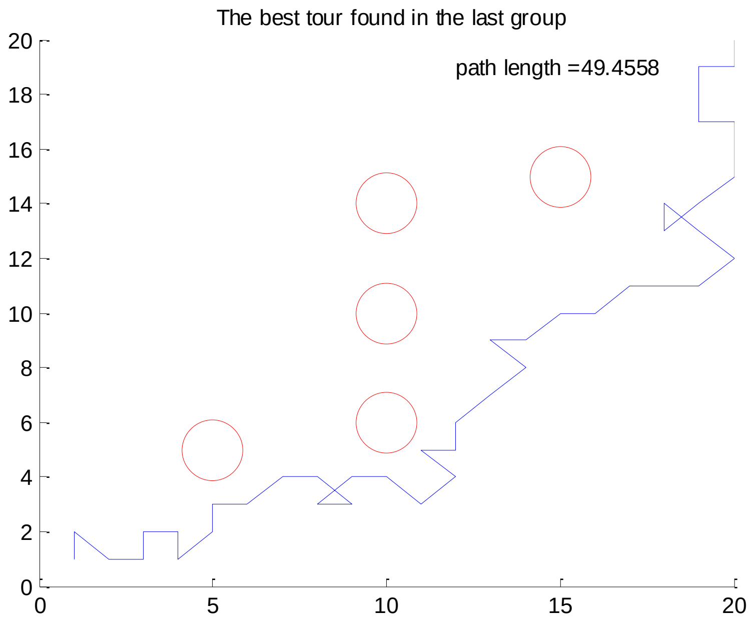

Figure E.41 :The best tour found at the 5 obstacle

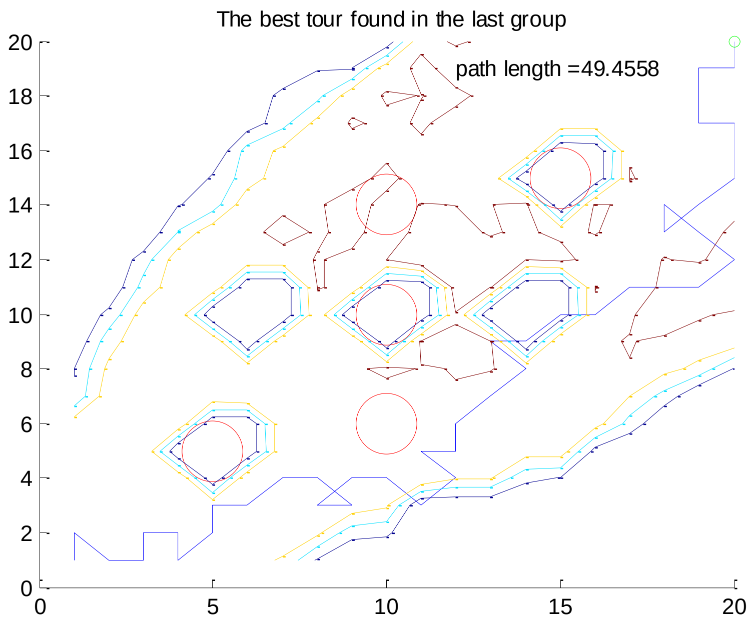

Figure E.42 Deposited pheromone in 2D contour map

22. 5 obstacle simulation 2

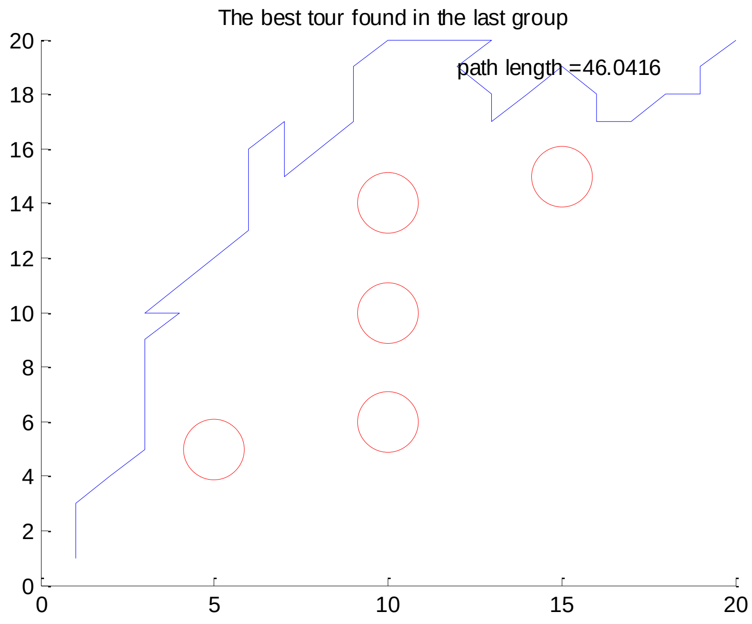

Figure E.43 :The best tour found at the 5 obstacle

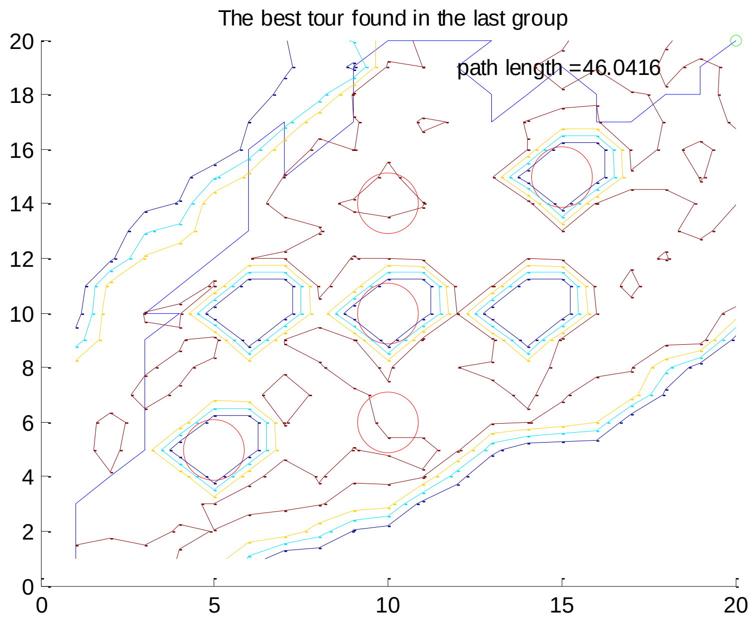

Figure E.44 Deposited pheromone in 2D contour map

23. 5 obstacle simulation 3

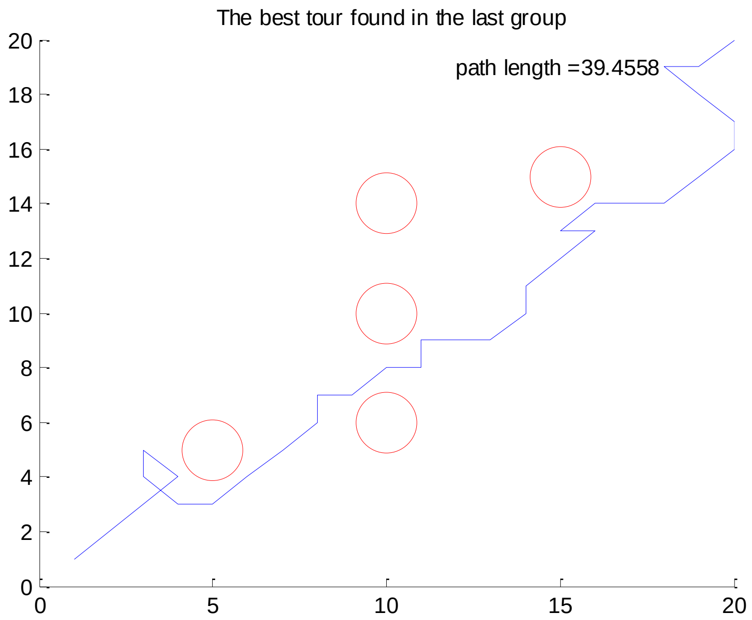

Figure E.45 :The best tour found at the 5 obstacle

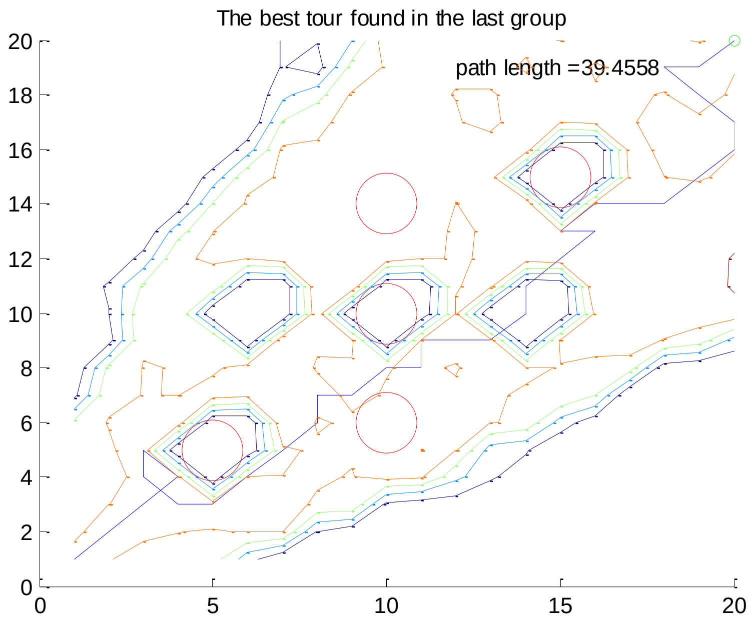

Figure E.46 Deposited pheromone in 2D contour map

24. 5 obstacle simulation 4

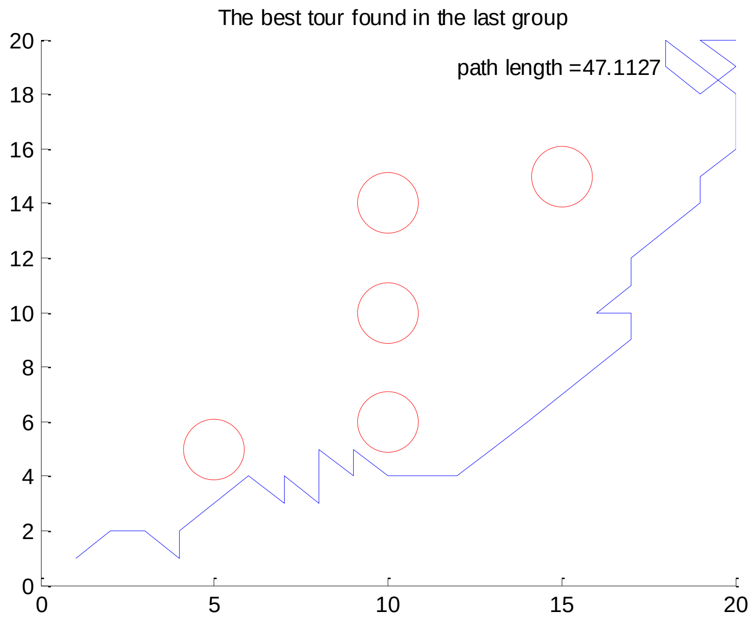

Figure E.47 :The best tour found at the 5 obstacle

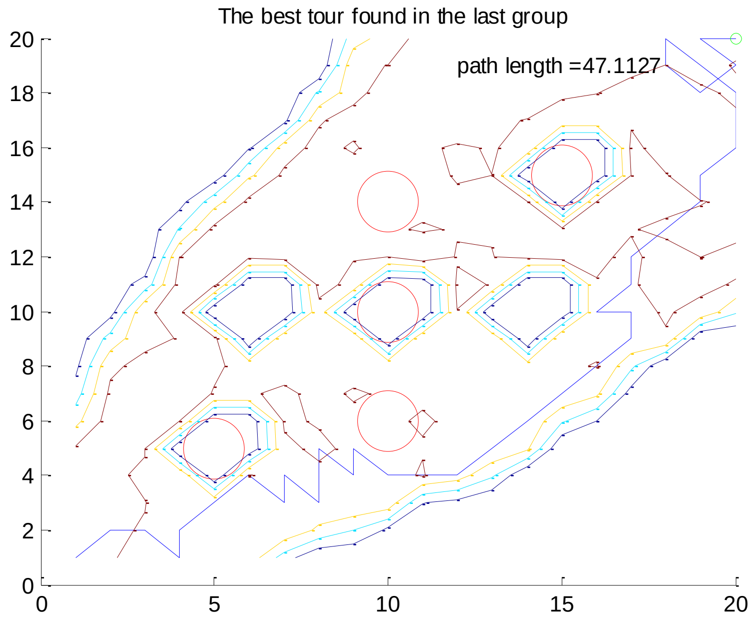

Figure E.48 Deposited pheromone in 2D contour map

25. 5 obstacle simulation 5

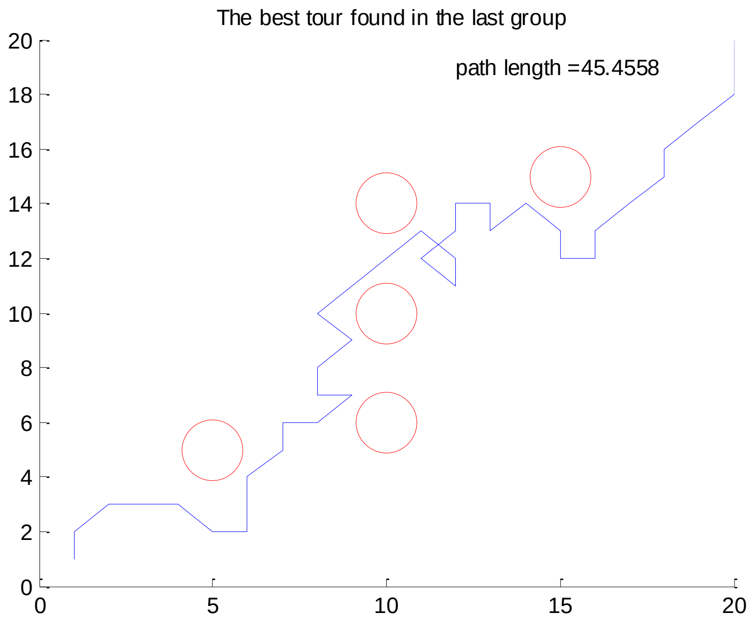

Figure E.49 :The best tour found at the 5 obstacle

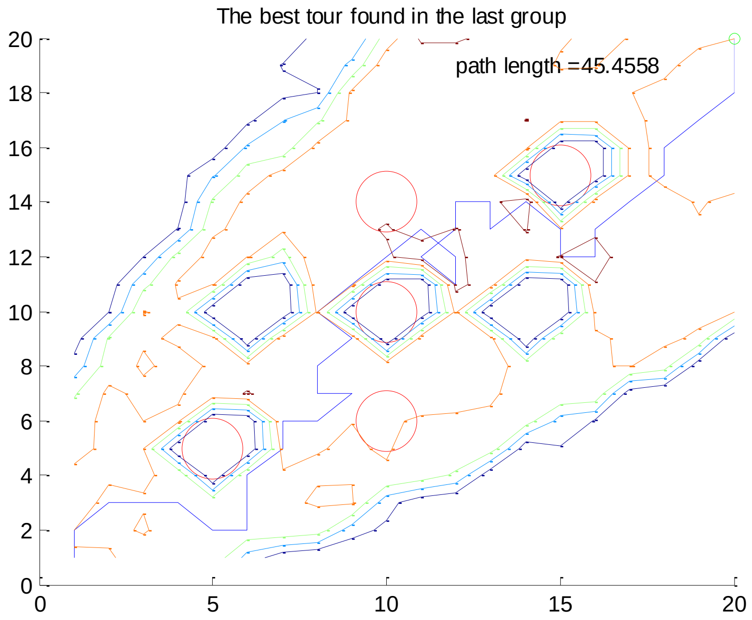

Figure E.50 Deposited pheromone in 2D contour map

26. 6 obstacle simulation 1

Figure E.51 :The best tour found at the 6 obstacle

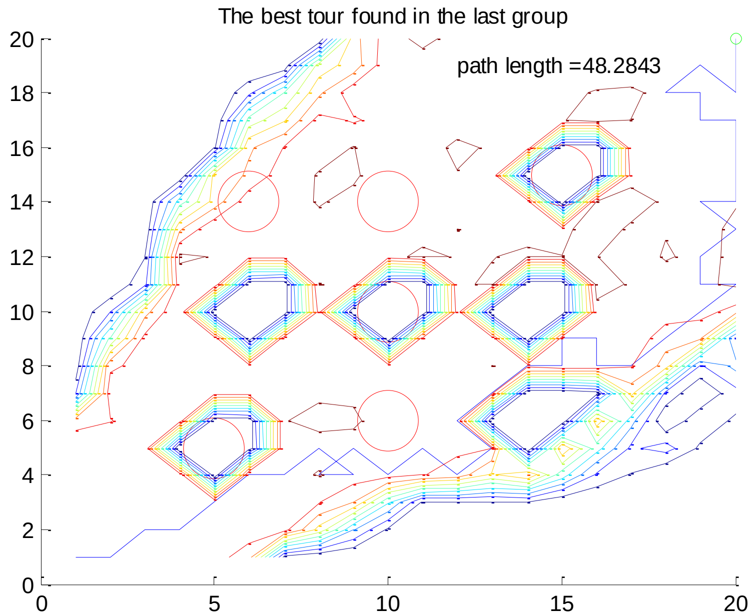

Figure E.52 Deposited pheromone in 2D contour map

27. 6 obstacle simulation 2

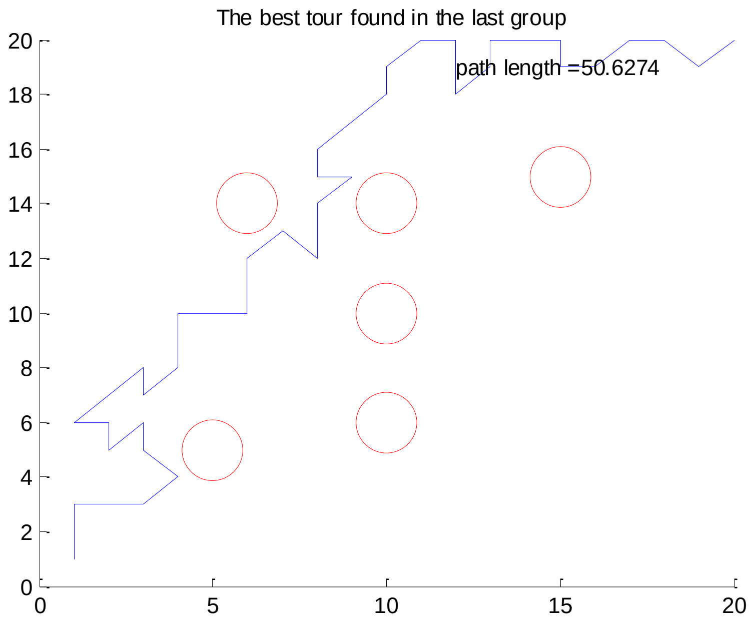

Figure E.53 :The best tour found at the 6 Obstacle

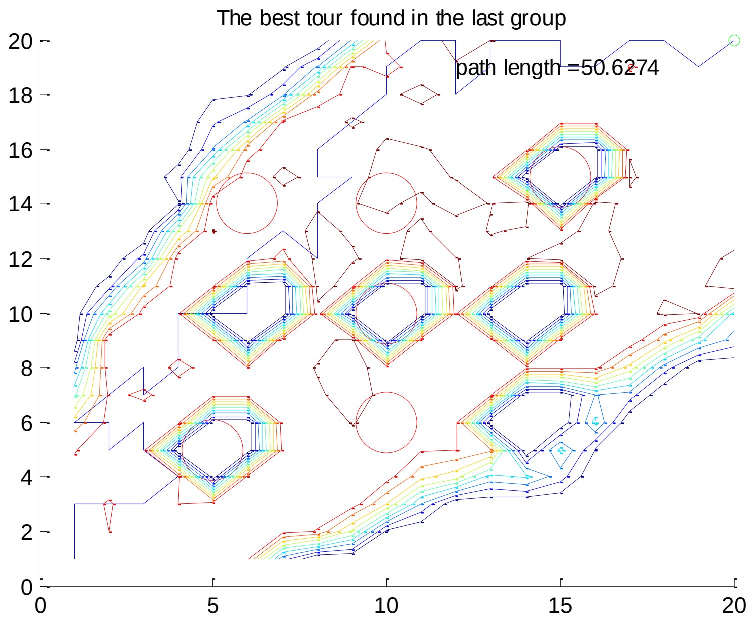

Figure E.54 Deposited pheromone in 2D contour map

28. 6 obstacle simulation 3

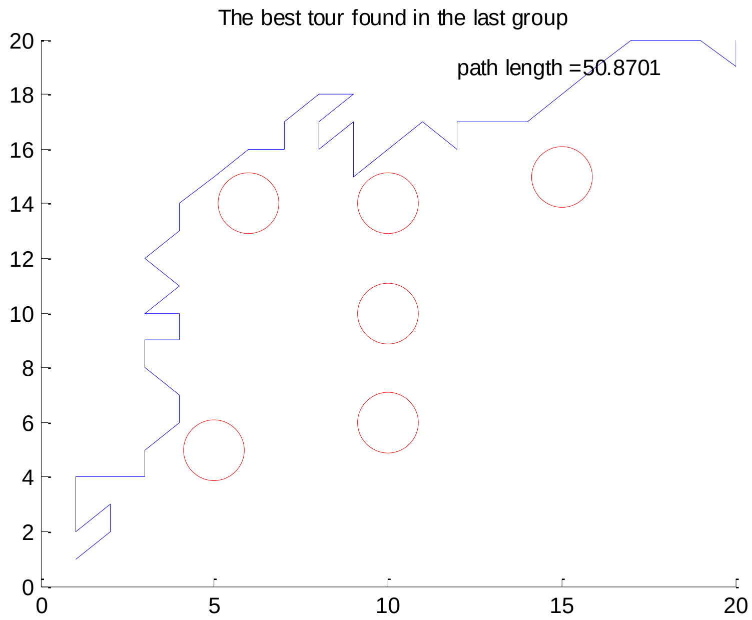

Figure E.55 :The best tour found at the 6 obstacle

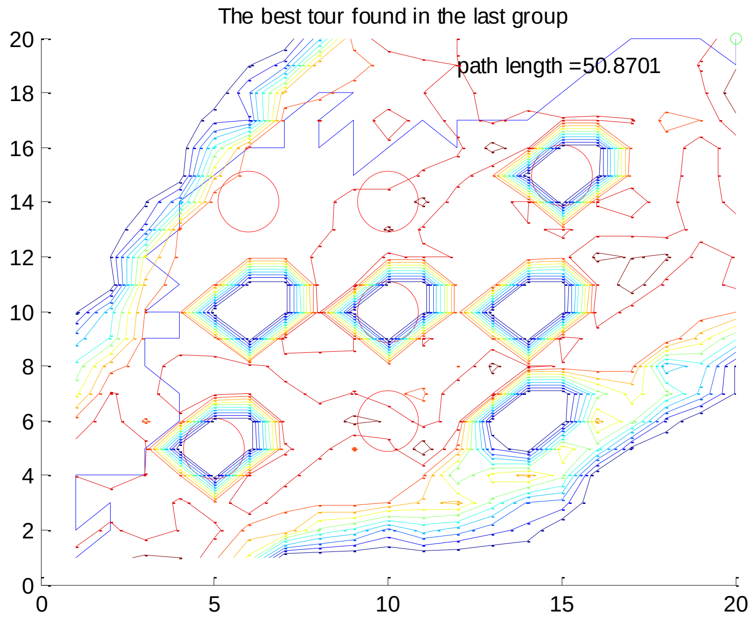

Figure E.56 Deposited pheromone in 2D contour map

29. 6 obstacle simulation4

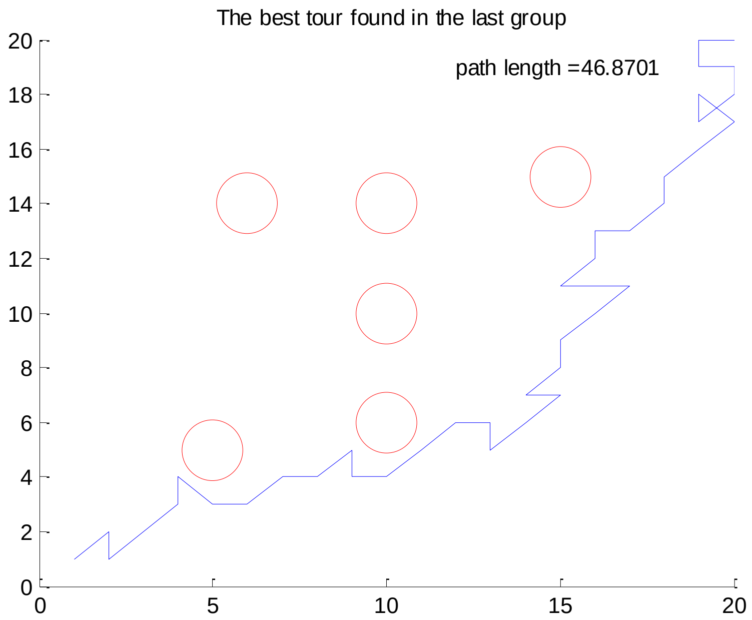

Figure E.57 :The best tour found at the 6 obstacle

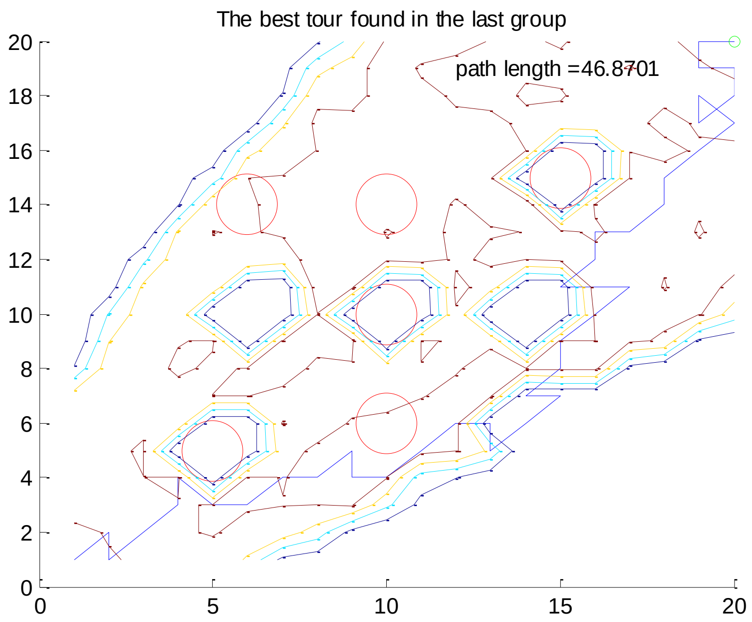

Figure E.58 Deposited pheromone in 2D contour map

30. 6 obstacle simulation 5

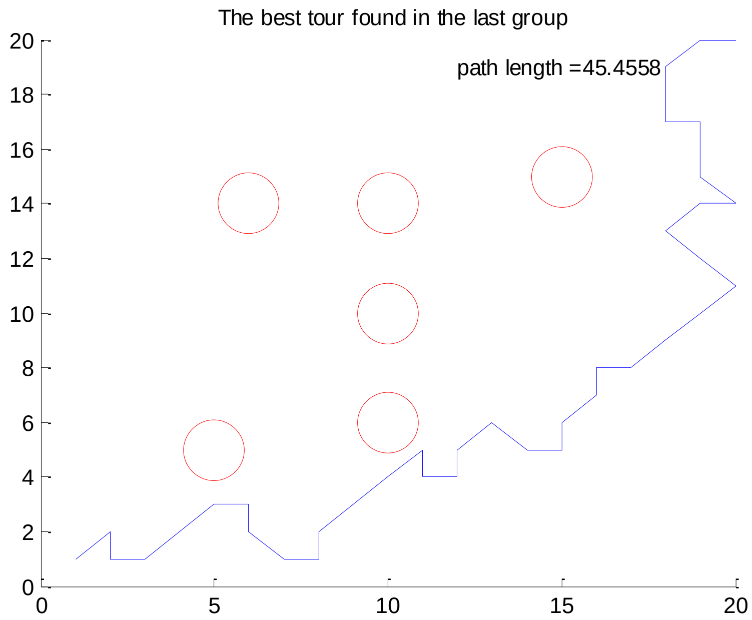

Figure E.59 :The best tour found at the 6 obstacle

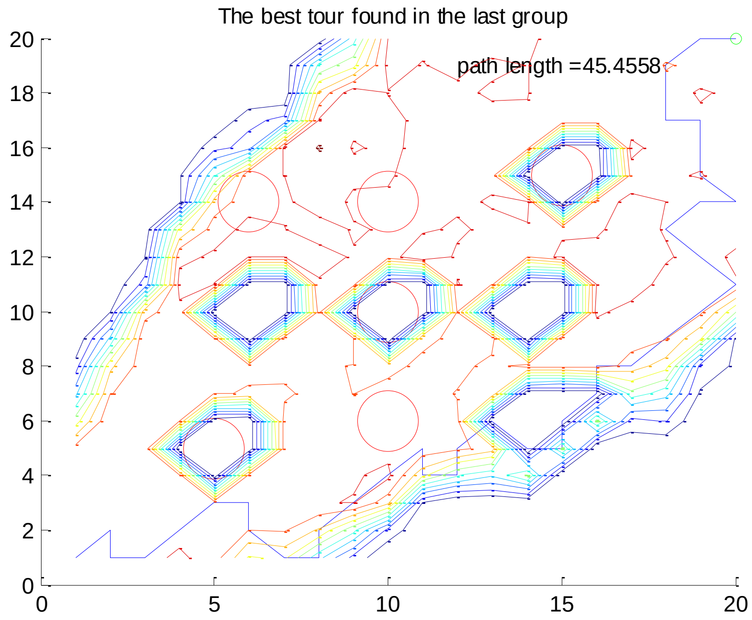

Figure E.60 Deposited pheromone in 2D contour map

31. 7 obstacle simulation 1

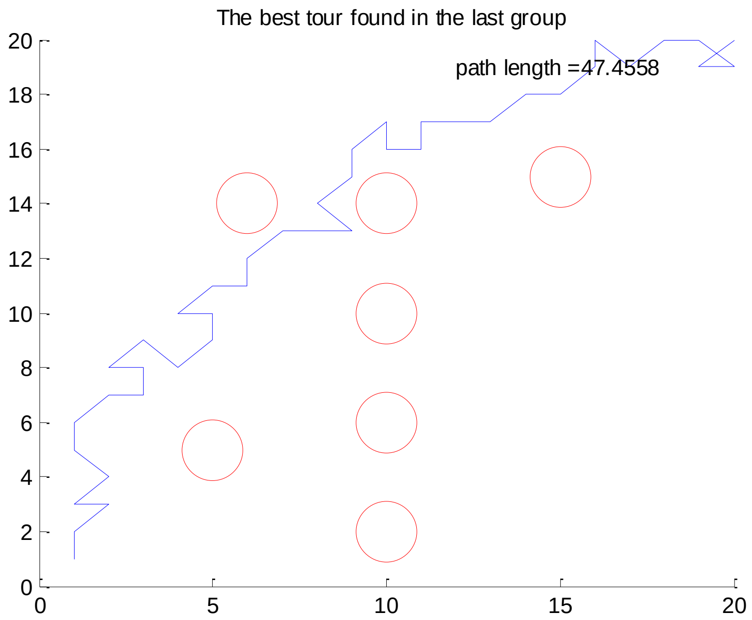

Figure E.61 :The best tour found at the 7 obstacle

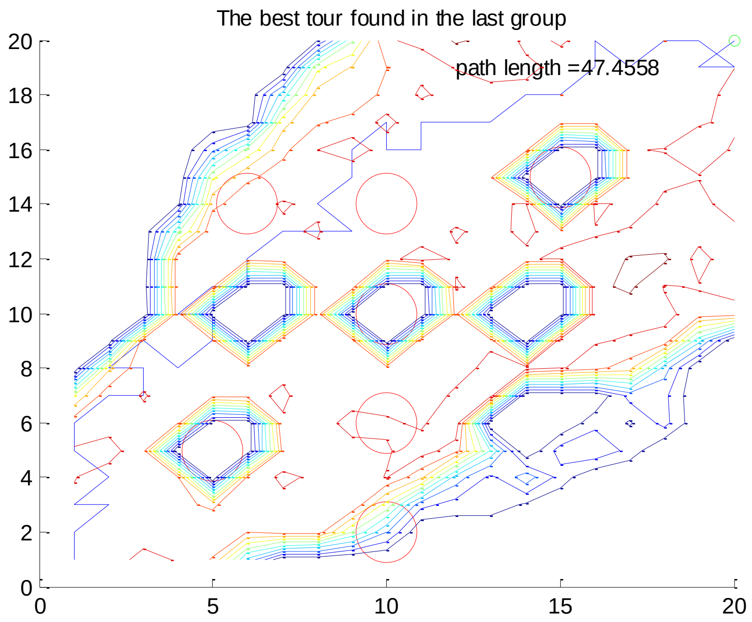

Figure E.62 Deposited pheromone in 2D contour map

32. 7 obstacle simulation 2

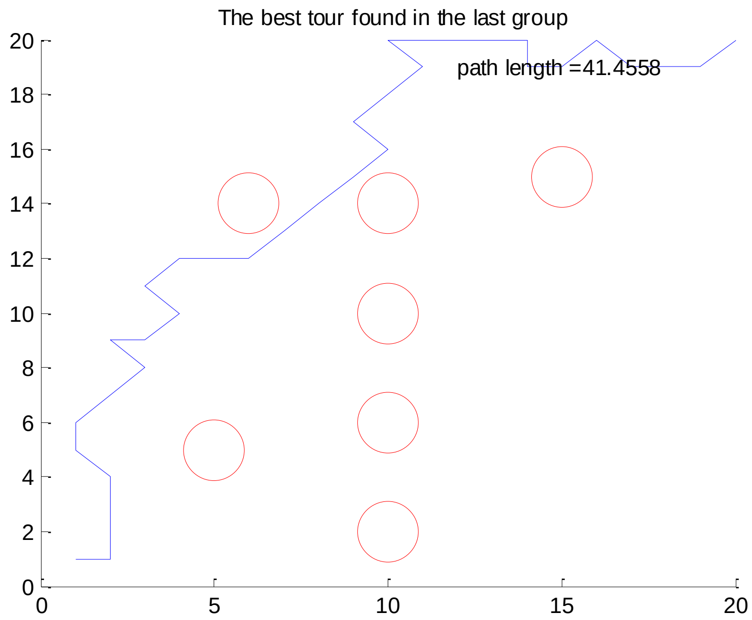

Figure E.63 :The best tour found at the 7 obstacle

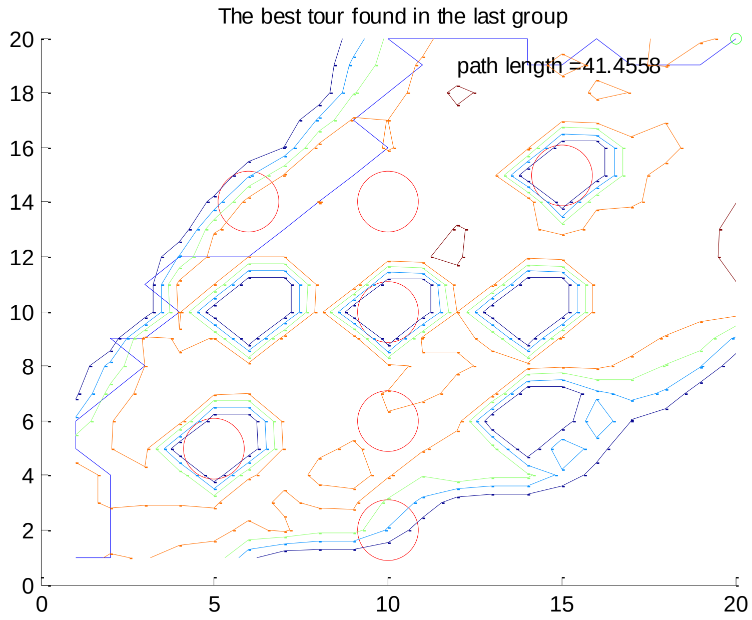

Figure E.64 Deposited pheromone in 2D contour map

33. 7 obstacle simulation 3

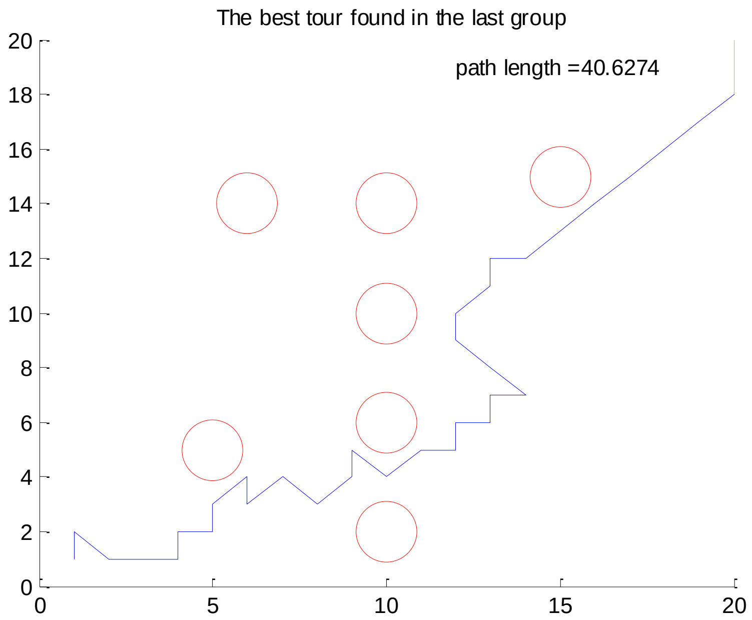

Figure E.65 :The best tour found at the 7 obstacle

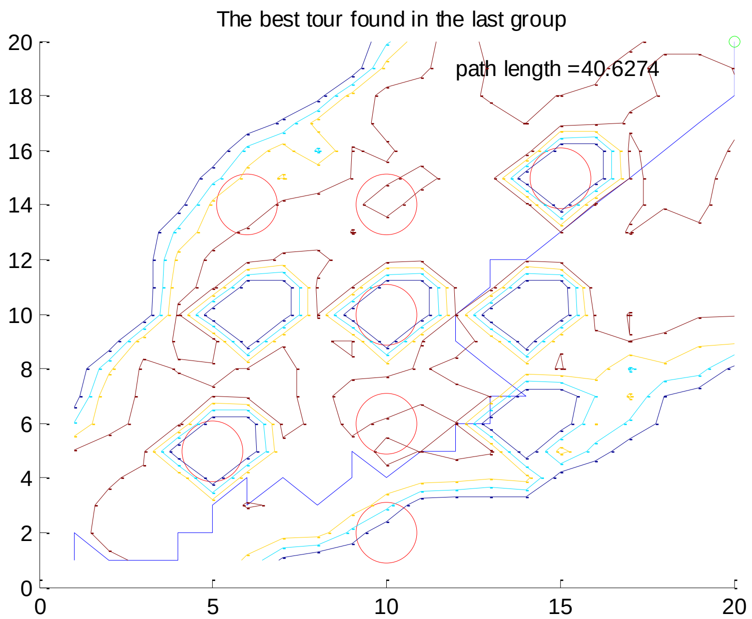

Figure E.66 Deposited pheromone in 2D contour map

34. 7 obstacle simulation 4

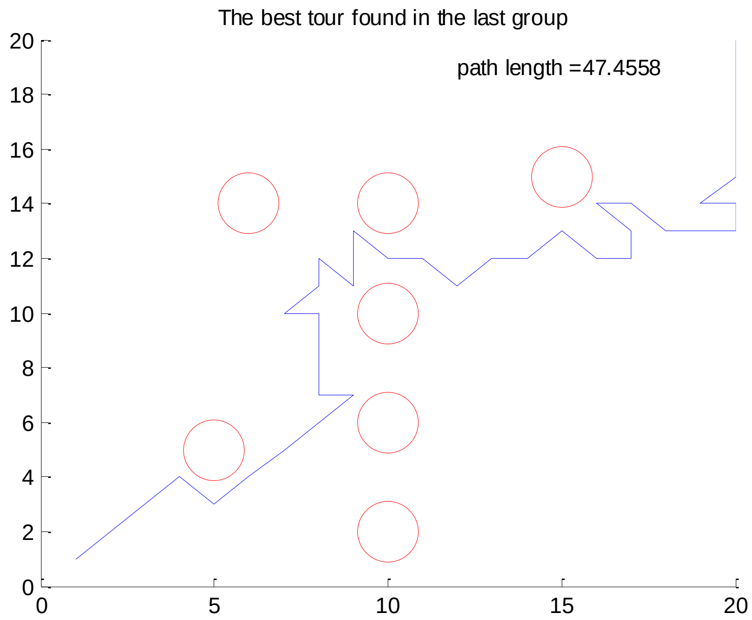

Figure E.67 :The best tour found at the 7 obstacle

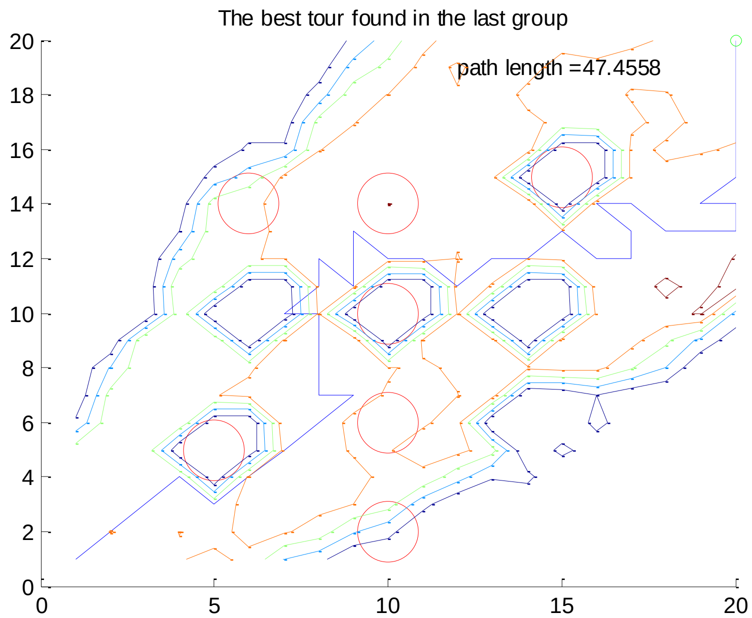

Figure E.68 Deposited pheromone in 2D contour map

35. 7 obstacle simulation 5

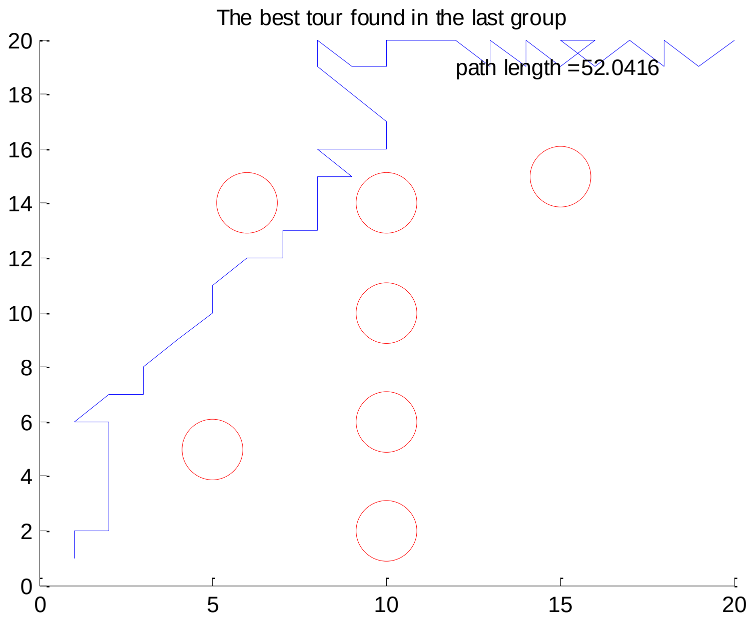

Figure E.69 :The best tour found at the 7 obstacle

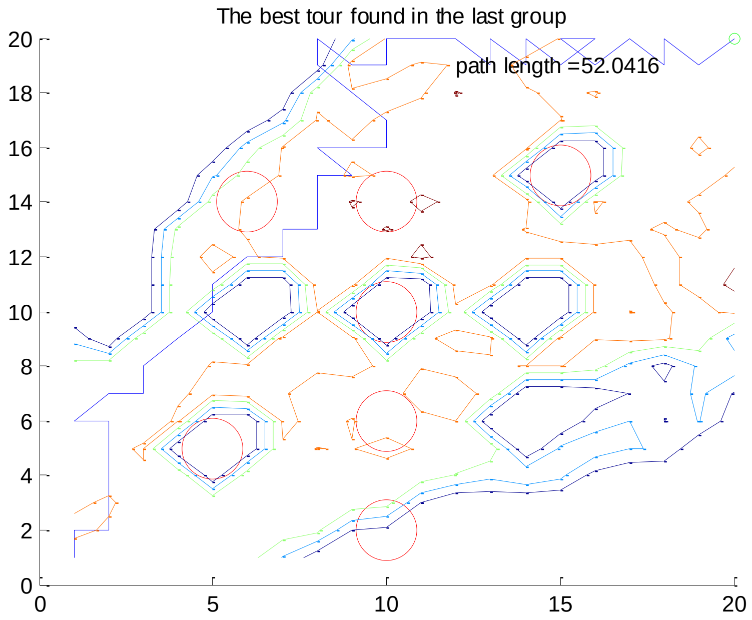

Figure E.70 Deposited pheromone in 2D contour map

36. 8 obstacle simulation 1

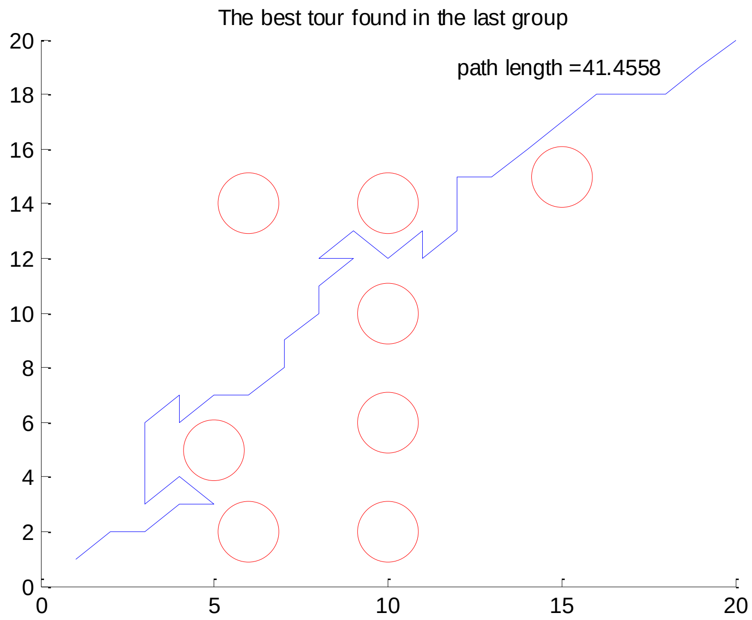

Figure E.71 :The best tour found at the 8 obstacle

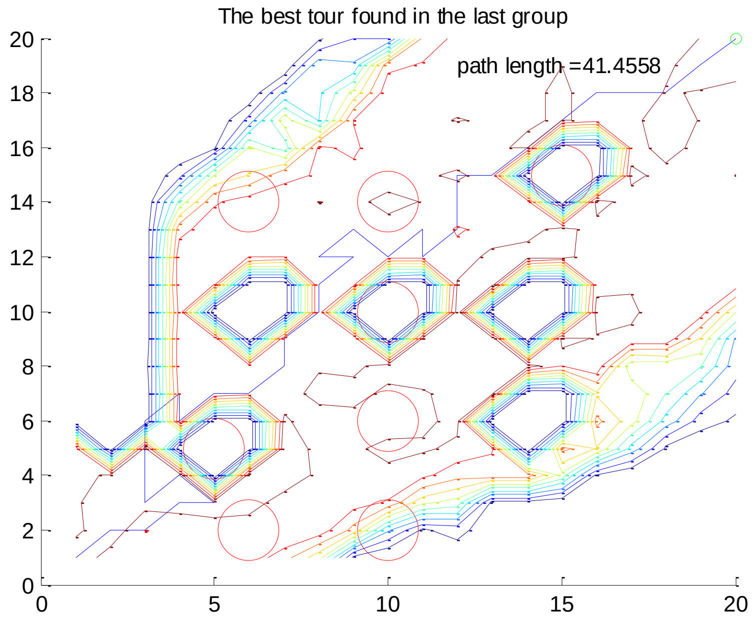

Figure E.72 Deposited pheromone in 2D contour map

37. 8 obstacle simulation 2

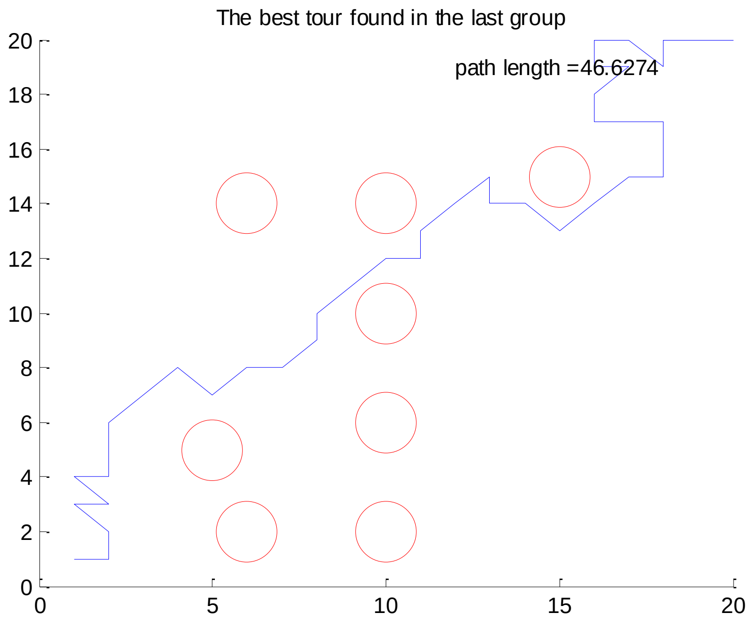

Figure E.73 :The best tour found at the 8 obstacle

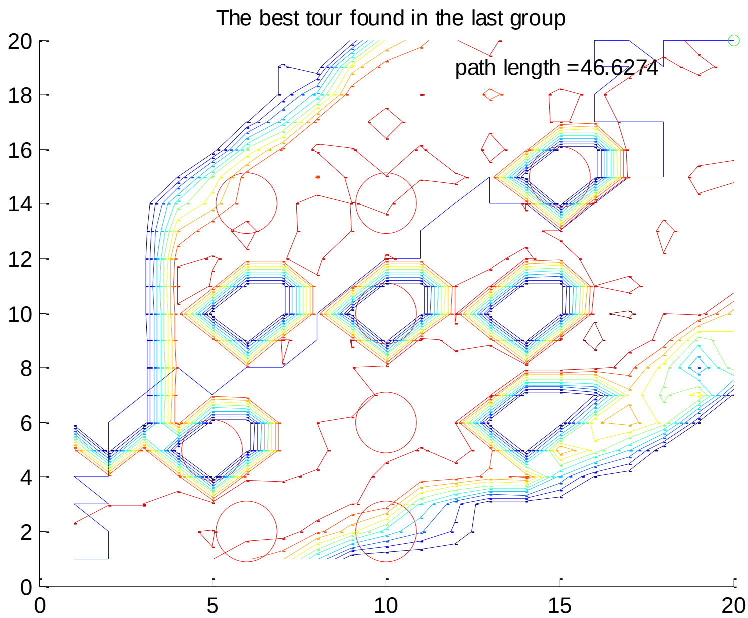

Figure E.74 Deposited pheromone in 2D contour map

38. 8 obstacle simulation 3

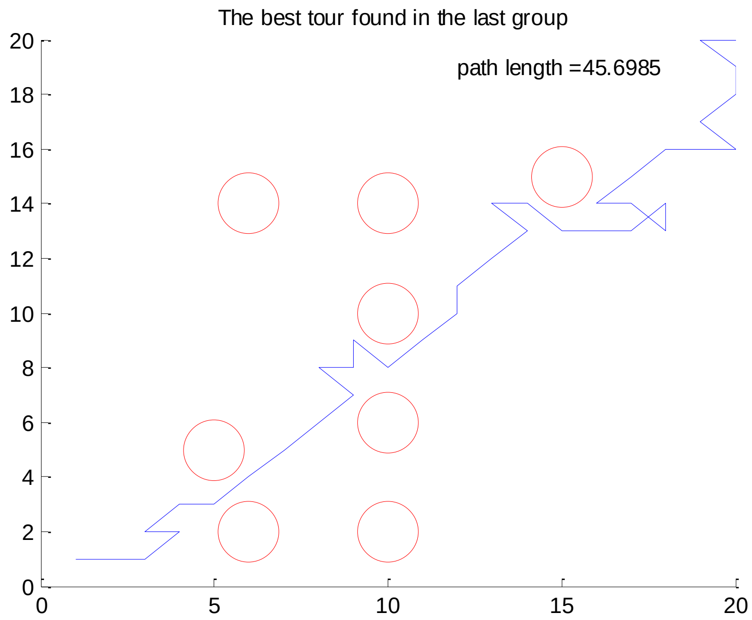

Figure E.75 :The best tour found at the 8 obstacle

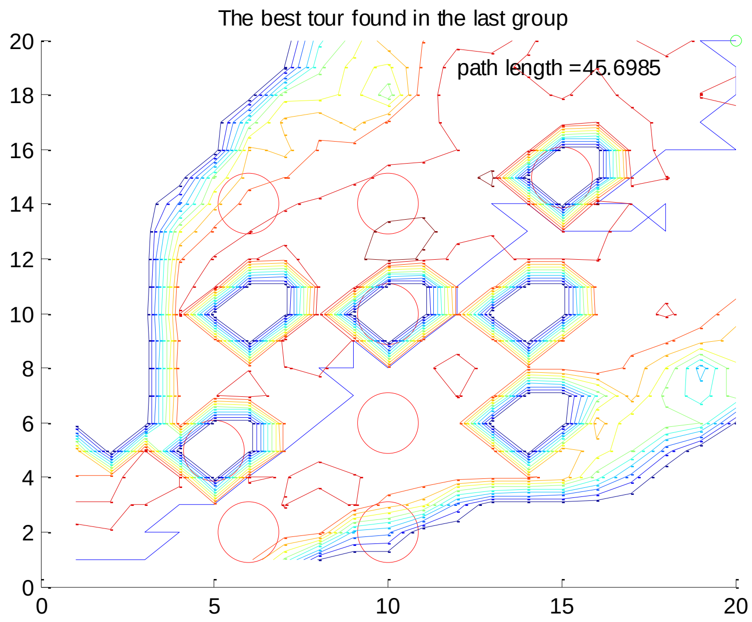

Figure E.76 Deposited pheromone in 2D contour map

39. 8 obstacle simulation 4

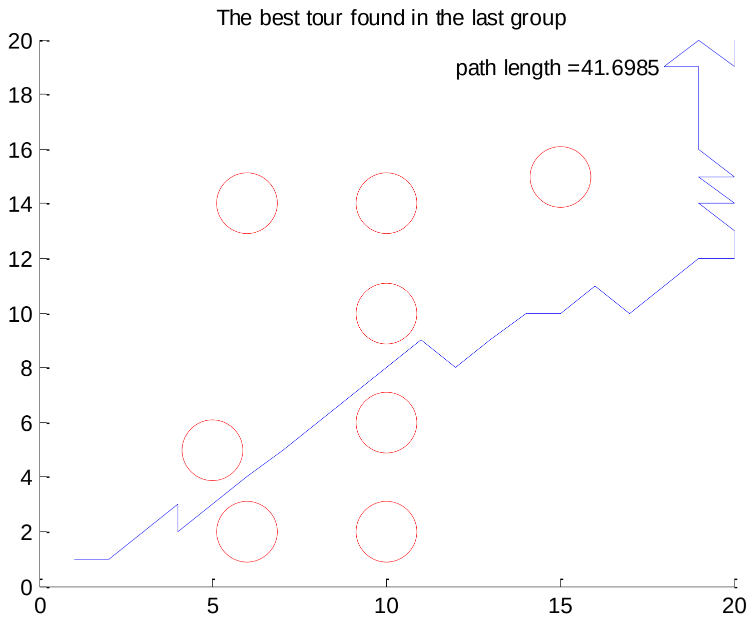

Figure E.77 :The best tour found at the 8 obstacle

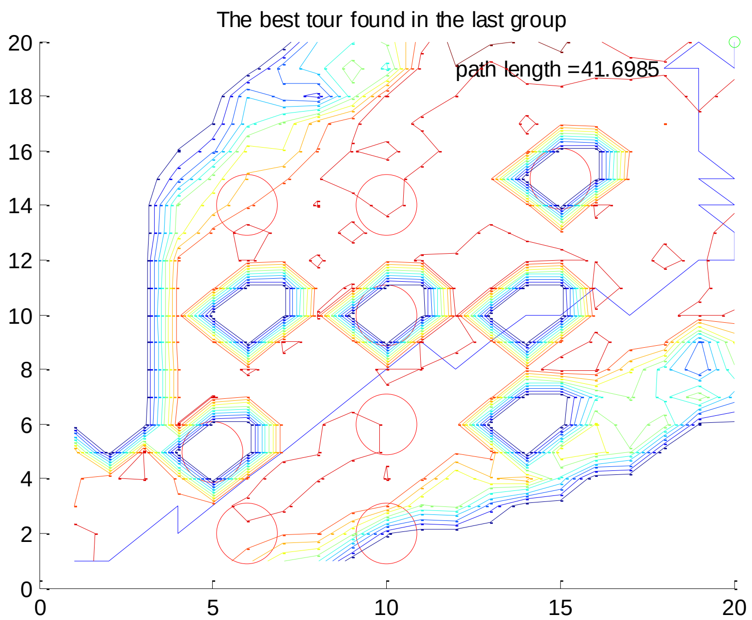

Figure E.78 Deposited pheromone in 2D contour map

40. 8 obstacle simulation 5

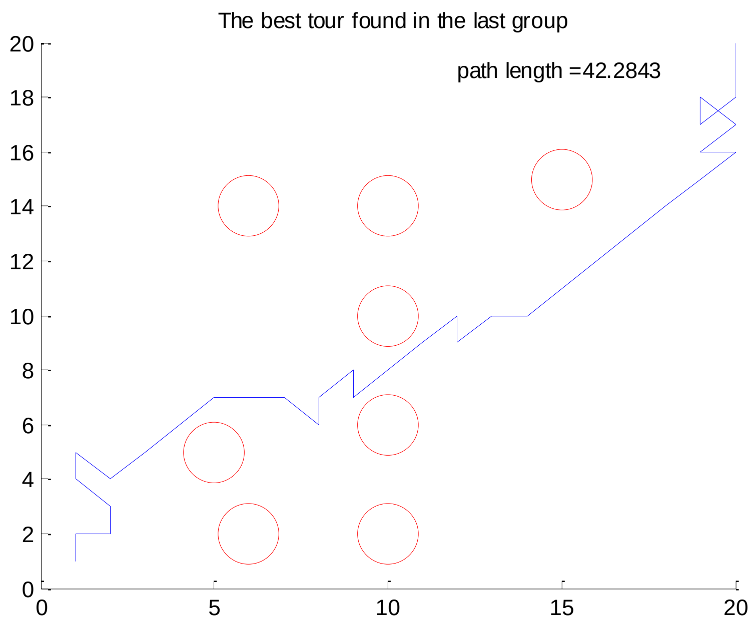

Figure E.79 :The best tour found at the 8 obstacle

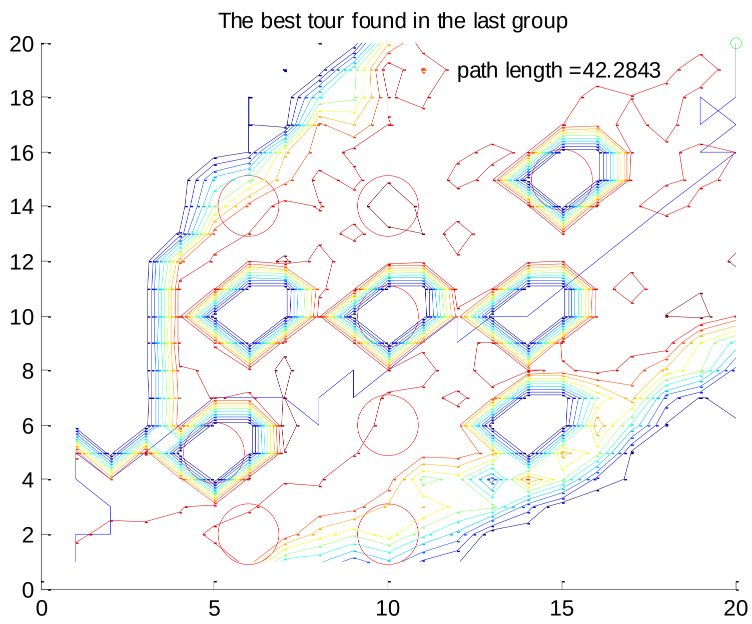

Figure E.80 Deposited pheromone in 2D contour map

41. 9 obstacle simulation 1

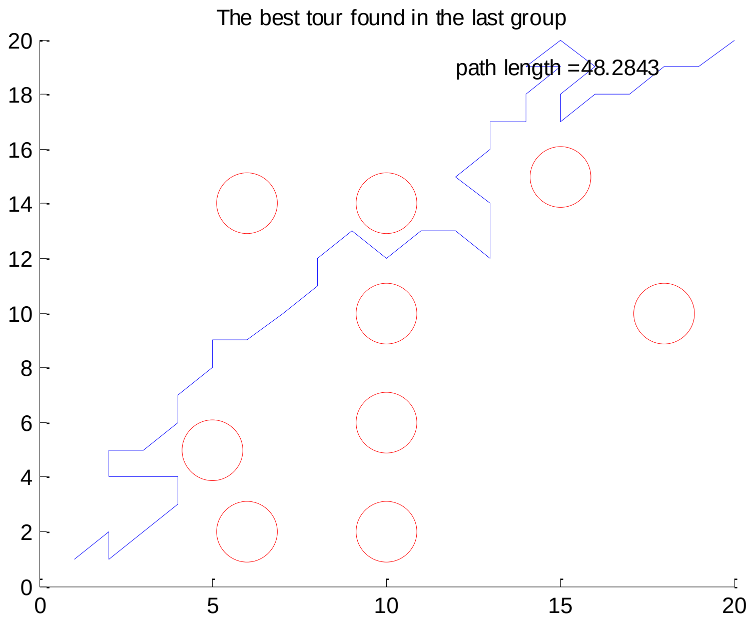

Figure E.81 :The best tour found at the 9 obstacle

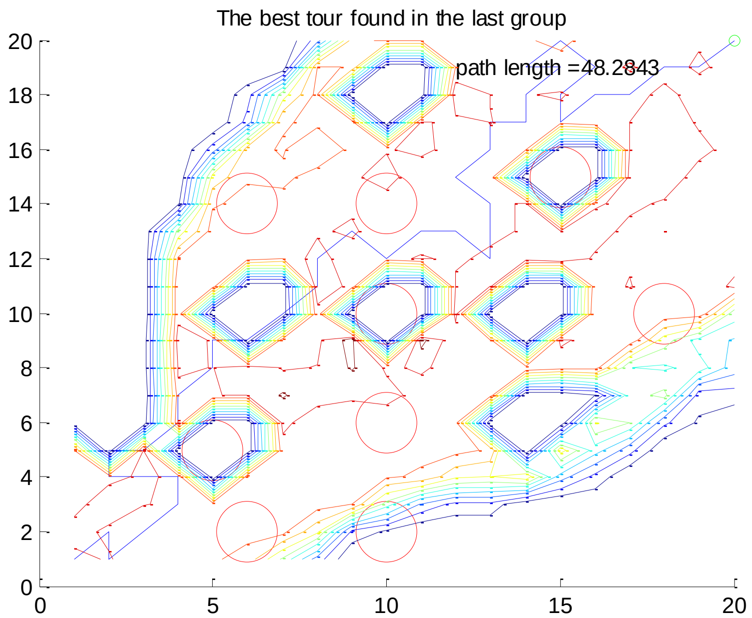

Figure E.82 Deposited pheromone in 2D contour map

42. 9 obstacle simulation 2

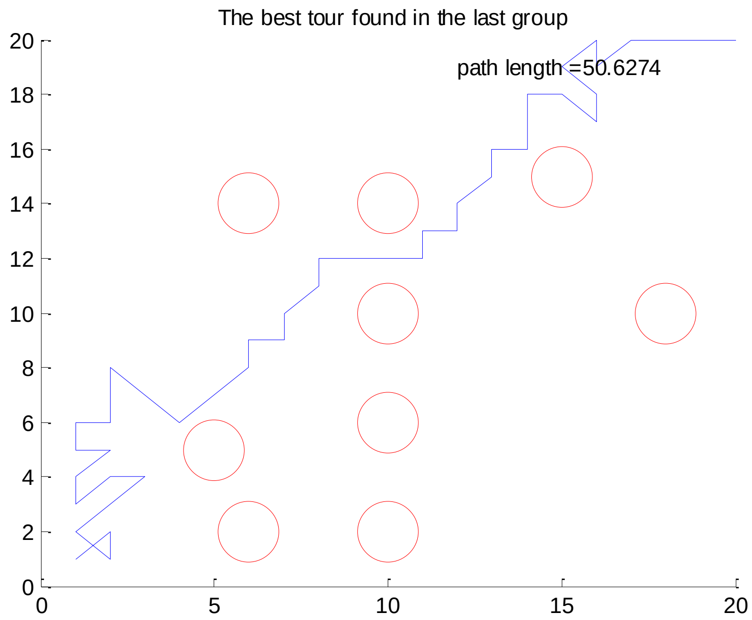

Figure E.83 :The best tour found at the 9 obstacle

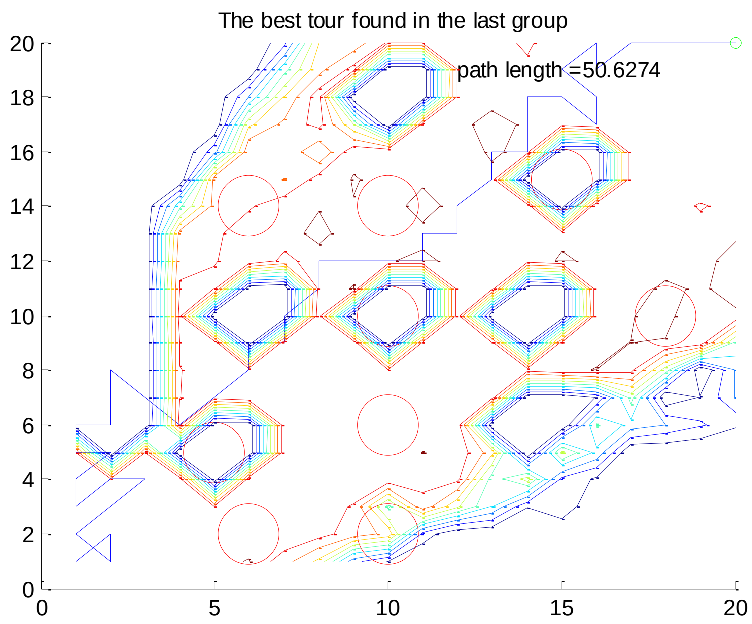

Figure E.84 Deposited pheromone in 2D contour map

43. 9 obstacle simulation 3

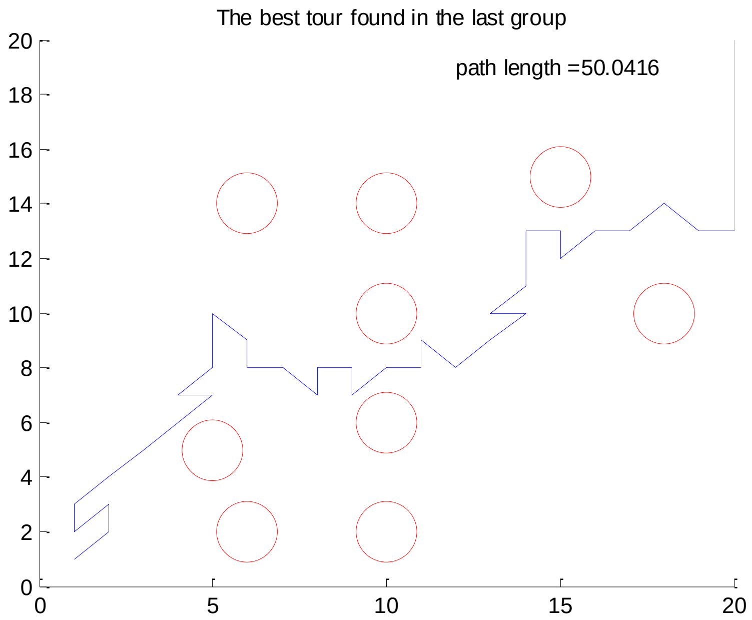

Figure E.85 :The best tour found at the 9 obstacle

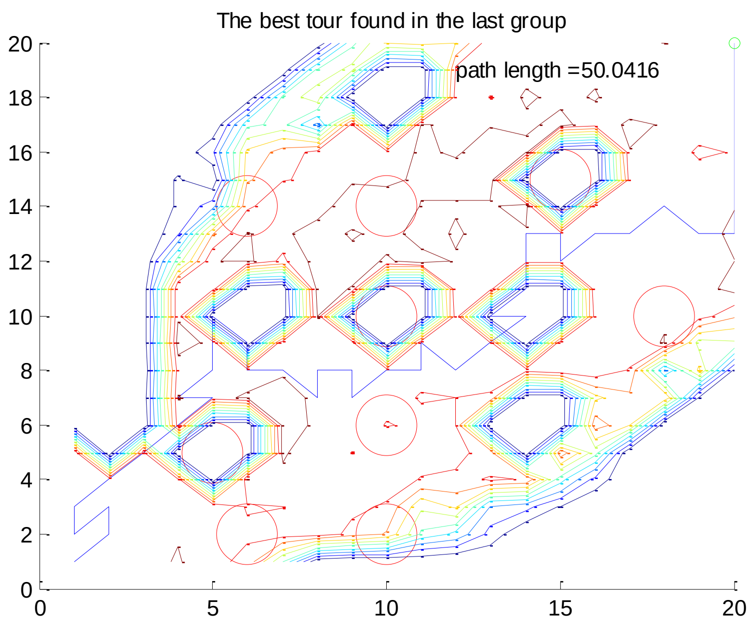

Figure E.86 Deposited pheromone in 2D contour map

44. 9 obstacle simulation 4

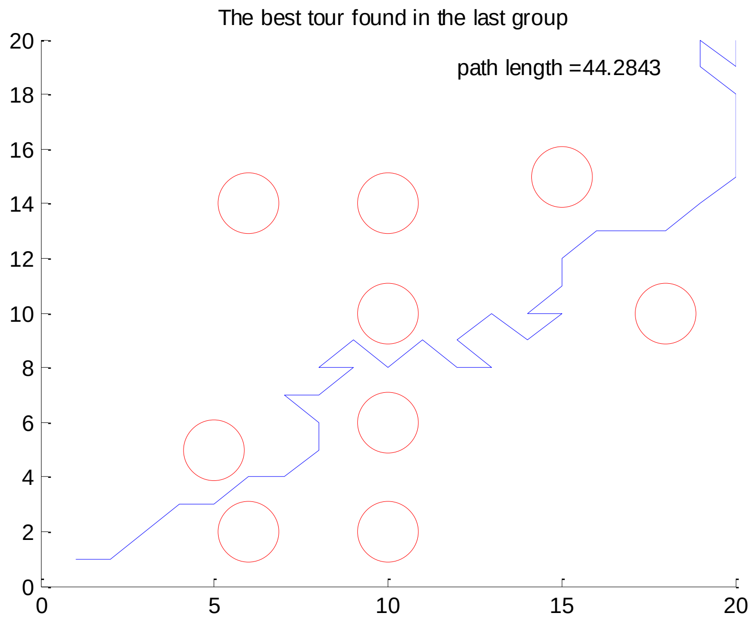

Figure E.87 :The best tour found at the 9 obstacle

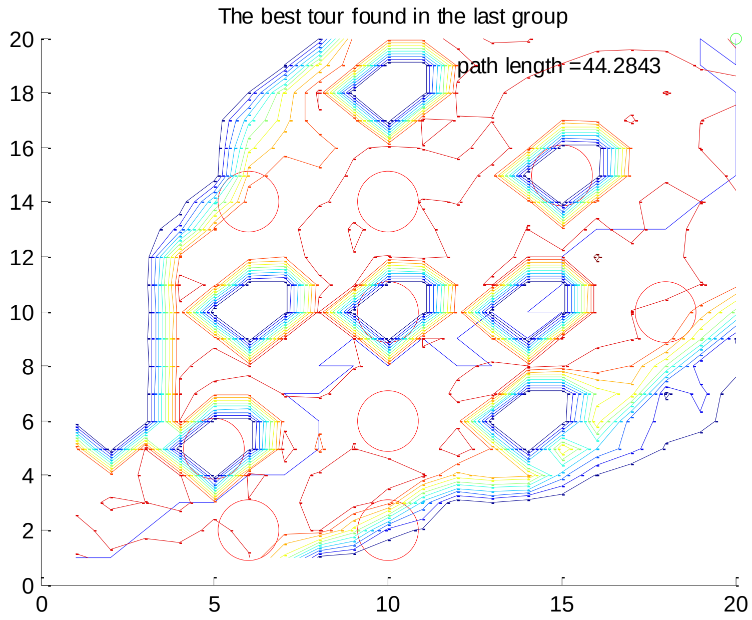

Figure E.88 Deposited pheromone in 2D contour map

45. 9 obstacle simulation 5

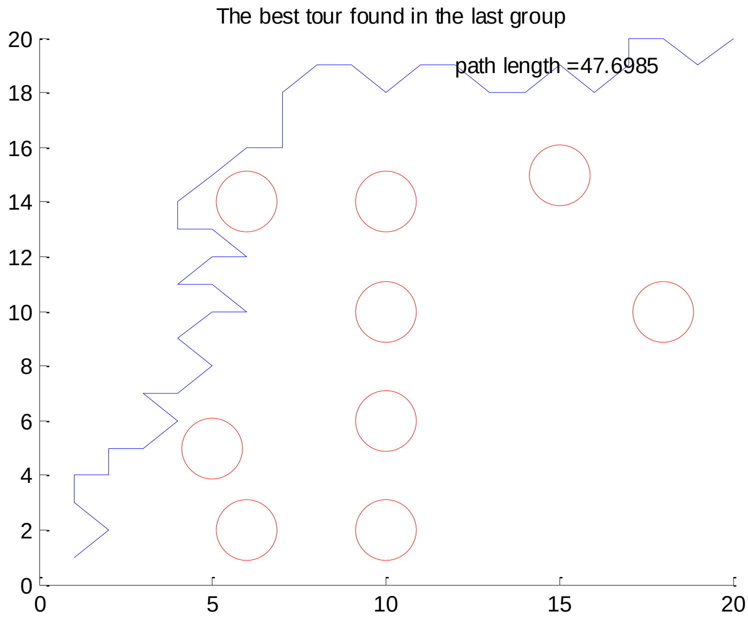

Figure E.89 :The best tour found at the 9 obstacle

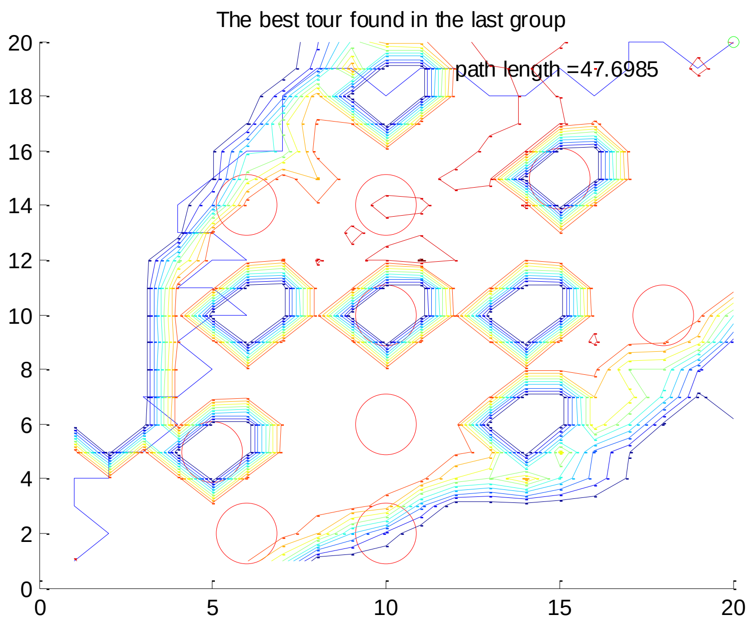

Figure E.90 Deposited pheromone in 2D contour map

46. 10 obstacle simulation 1

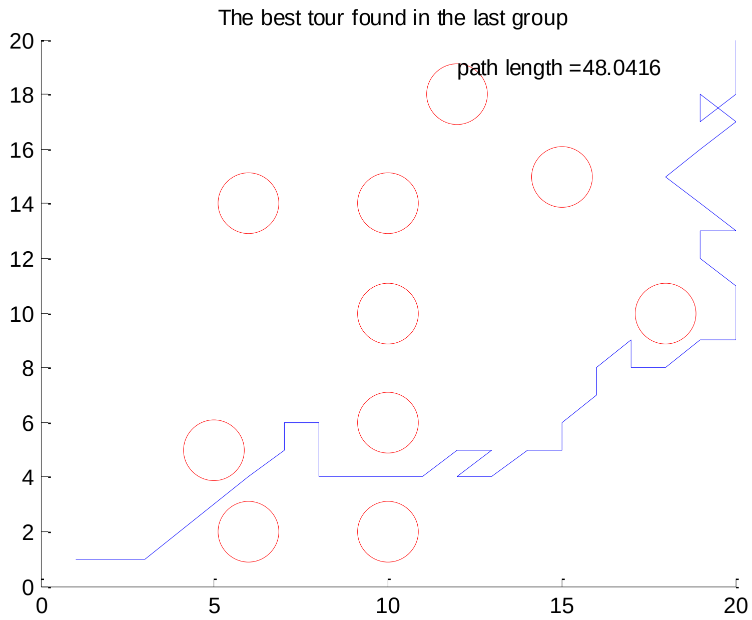

Figure E.91 :The best tour found at the 10 obstacle

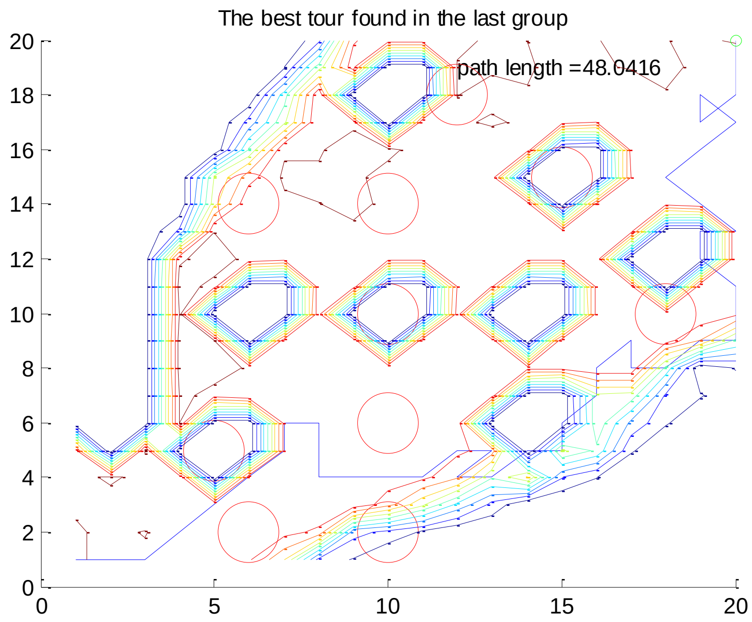

Figure E.92 Deposited pheromone in 2D contour map

47. 10 obstacle simulation 2

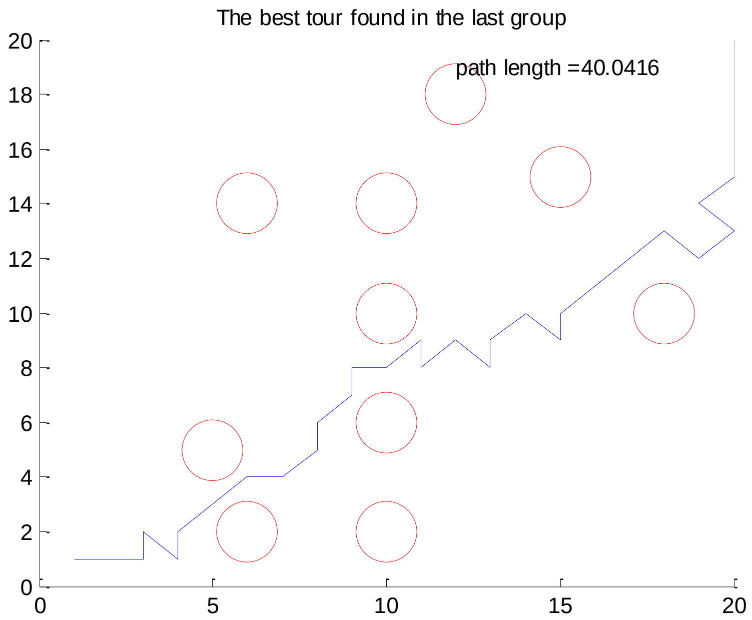

Figure E.93 :The best tour found at the 10 obstacle

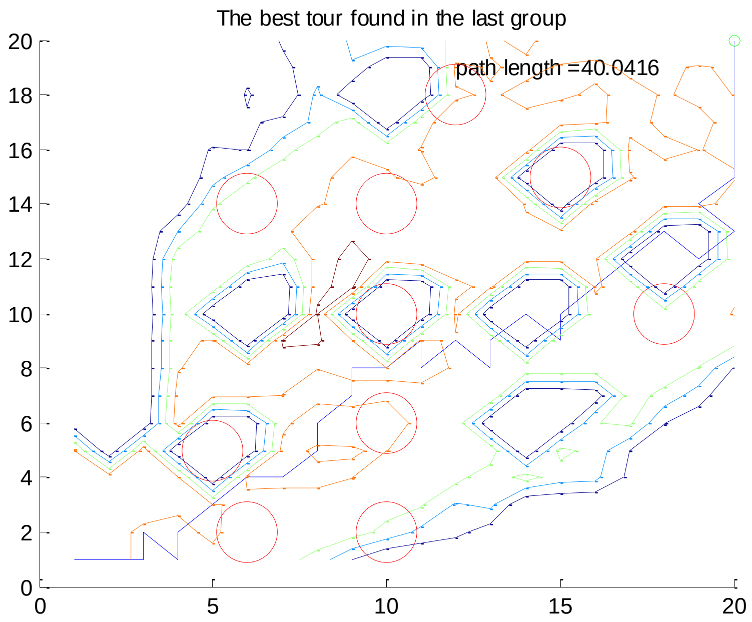

Figure E.94 Deposited pheromone in 2D contour map

48. 10 obstacle simulation 3

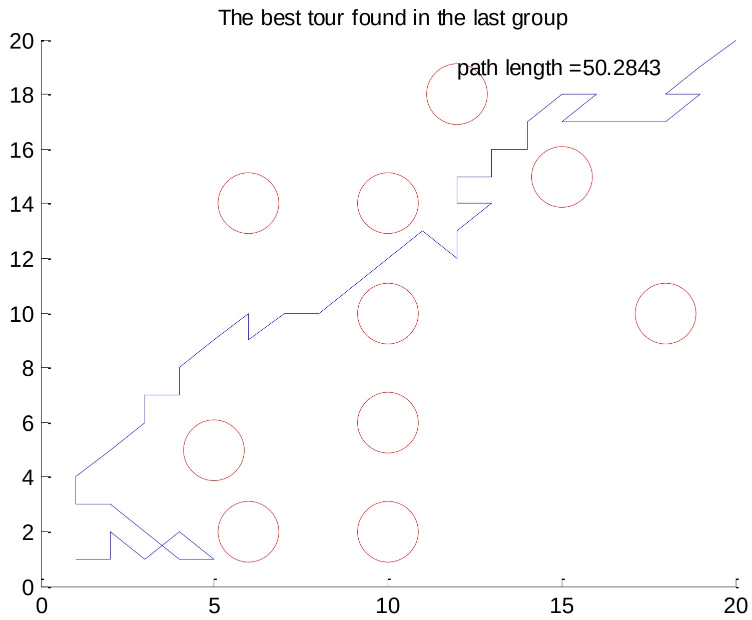

Figure E.95 :The best tour found at the 10 obstacle

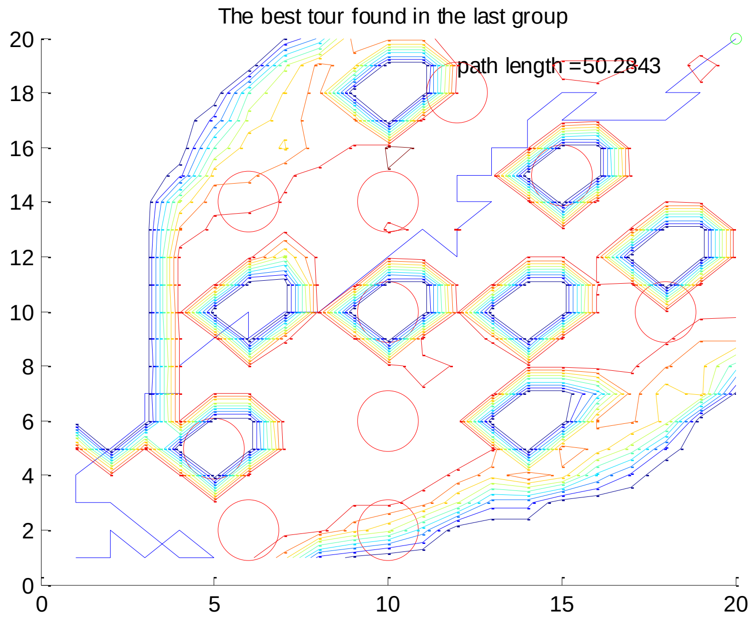

Figure E.96: Deposited pheromone in 2D contour map

49. 10 obstacle simulation 4

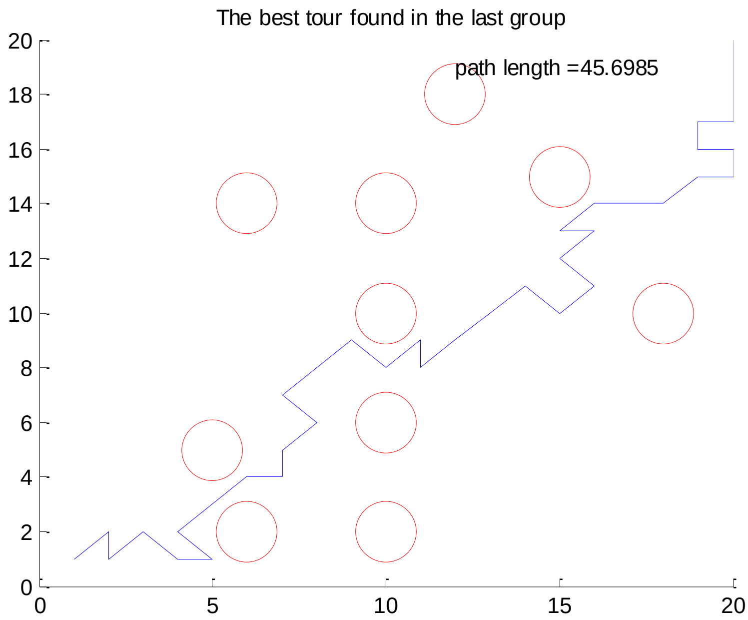

Figure E.97 :The best tour found at the 10 obstacle

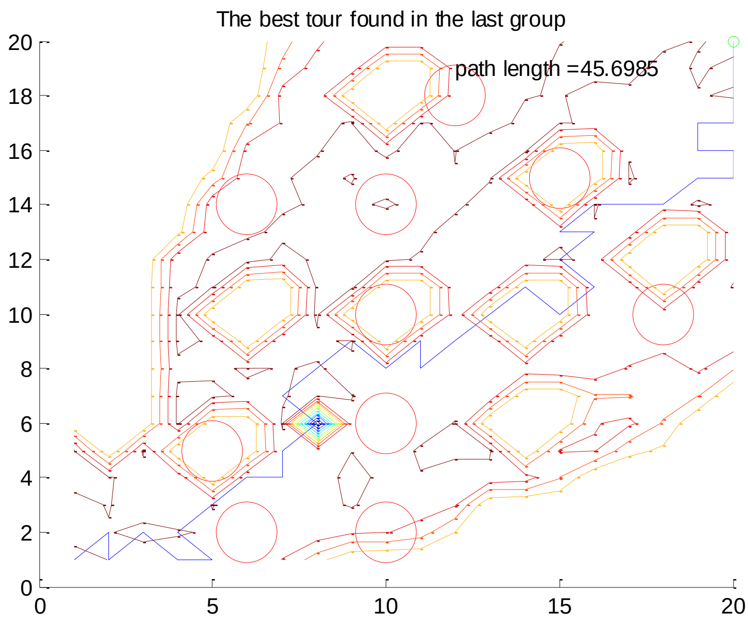

Figure E.98 : Deposited pheromone in 2D contour map

50. 10 obstacle simulation 5

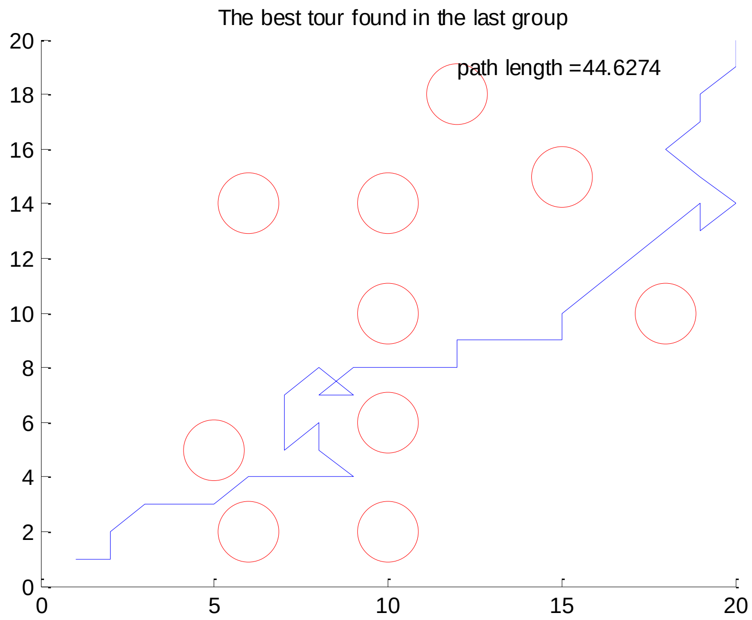

Figure E.99 :The best tour found at the 10 obstacle

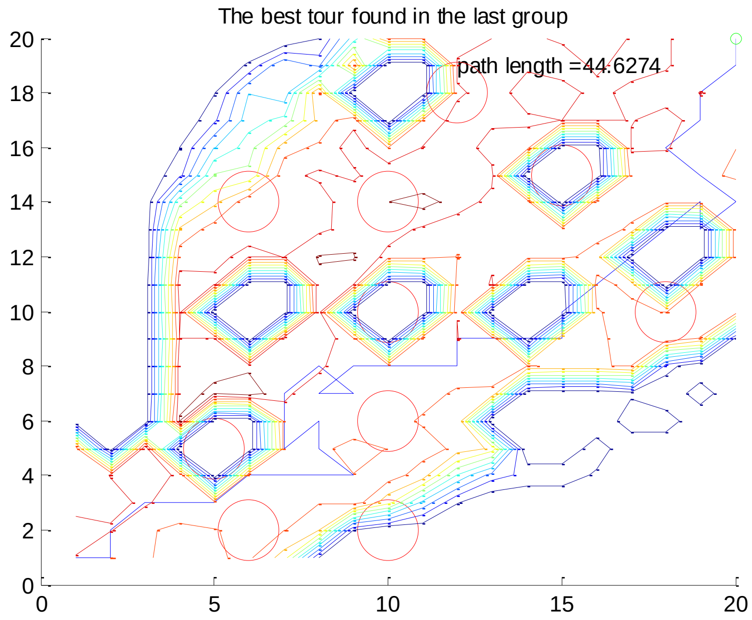

Figure E.100 : Deposited pheromone in 2D contour map

APPENDIX F

Table F1: Comparison of ACO with GA [65]

		Figure 3	Figure 4	Figure
GA	Length	46	59	72
	Time (ms)	301	340	1001
	Turnings	18	34	44
	Performance	529	687	1437
ACO	Length	48	46	60
	Time (ms)	83	246	560
	Turnings	23	17	30
	Performance	342	469	890