

# **MOTION CONTROL OF AUTOMATED MOBILE ROBOTS IN DYNAMIC ENVIRONMENT**

A THESIS SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENTS FOR THE DEGREE OF

**Master of Technology**

*In*

**Industrial Design**

*By*

**Mihir Kanta Rath**

(Roll: 213ID1368)



**Department of Industrial Design  
National Institute of Technology  
Rourkela-769008, Odisha, India  
June 2015**

# **MOTION CONTROL OF AUTOMATED MOBILE ROBOTS IN DYNAMIC ENVIRONMENT**

A THESIS SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENTS FOR THE DEGREE OF

**Master of Technology**

*In*

**Industrial Design**

*By*

**Mihir Kanta Rath**

Under the supervision of

**Prof. B.B.V.L Deepak**



**Department of Industrial Design  
National Institute of Technology  
Rourkela-769008, Odisha, India  
June 2015**



**National Institute of Technology  
Rourkela -769008, Odisha, India**

## **CERTIFICATE**

This is to certify that the work in the thesis entitled, “**Motion Control Of Automated Mobile Robots in Dynamic Environment**” submitted by **Mr. Mihir Kanta Rath** in partial fulfilment of the requirements for the award of **Master of Technology Degree** in the Department of Industrial Design, National Institute of Technology, Rourkela is an authentic work carried out by him under my supervision and guidance.

To the best of my knowledge, the work reported in this thesis is original and has not been submitted to any other Institution or University for the award of any degree or diploma.

He bears a good moral character to the best of my knowledge and belief.

Place: NIT Rourkela

Date:

**Prof. (Dr.) B.B.V.L Deepak**  
Assistant Professor  
Department of Industrial Design  
National Institute of Technology, Rourkela

## **ACKNOWLEDGEMENT**

I feel privileged to express my heartiest thanks and gratitude to **Prof. B.B.V.L Deepak, Asst. Professor, Department of Industrial Design**, for his precise and inspiring guidance, valuable and generous suggestions throughout this project. He contributed to a measure in surmounting all hardships; I faced during the project work. His tolerance in dealing the problems and encouragement were constant source of inspiration.

I am profusely thankful to **Prof. B.B. Biswal, Head of Department, Industrial Design**, for guiding me and helping me in my research work.

I am grateful to **Prof. M. R. Khan, Department of Industrial Design** for supporting and helping me in my researching work.

Last but not the least; I would like to thank to all the staffs of NIT ROURKELA and my friends those who have directly or indirectly helped me to successfully complete the project.

**Mihir Kanta Rath**

# **Abstract**

Autonomous mobile robot navigation had been a challenge for the researches from decades. Control and navigation of mobile robot in unknown environment is burning topic in the field of robotics. Several researchers have done lot of contribution in robot navigation problem. A general robot navigation problem includes features like obstacle detection and avoidance, smooth travel and reaching of a particular goal position. Among these aspects the obstacle avoidance part is of paramount importance in robot navigation problem. The robot will avoid the collision with objects if it has the ability to sense obstacle, take decision and move away from obstacle that means the robot should be intelligent and Intelligence can be achieved through programming. Here the main goal is to design and develop multiple intelligent mobile robots for autonomous navigation in unknown dynamic environment. Deployment of multiple mobile robots in unknown environment is worthwhile compared to single mobile robot. At the same time it will add more complexity and difficulty in controlling all the mobile robots. Multi-robot cooperation has lot of implications like target seeking, search and rescue, and disaster control. The obstacle avoidance issue of multiple mobile robots in unknown dynamic environment is addressed in this paper. For better motion control and obstacle avoidance PSO algorithm is used. In future goal seeking task of the mobile robots will be performed.

# **CONTENTS**

<b>TITLE</b>	<b>PAGE</b>
<i>Acknowledgement</i>	
Abstract	i
List of Figures	v
List of Tables	viii
<b>1: INTRODUCTION</b>	
1.1: Origin of the work	4
1.2: Problem Statement	7
1.3: Objectives	7
1.4: Thesis Overview	7
<b>2: LITRETURE SURVEY</b>	
2.1: Overview	8
2.2: Implementation of Particle Swarm Optimization (PSO) in multi-robot cooperation	8
2.3: Motion planning of mobile robot using PSO and its variants	12
2.4: Path Planning of Mobile Robot Using Genetic Algorithm	16
2.5: Mobile Robot Navigation Using Reactive Algorithms	17
2.6: Summary	18

### **3: MECHANICAL SYSTEM ARCHITECTURE FOR MOBILE ROBOT**

3.1: Overview	19
3.2: Mobile Robot Positioning	19
3.3: Range Sensor module specification and Working	20
3.4: Actuators for locomotion	25
3.5: Controller for Mobile Robot	29
3.6: Summary	31

### **4: PSO BASED SYSTEM ARCHITECTURE FOR MOBILE ROBOT**

4.1: Overview	32
4.2: Mathematical representation of PSO	32
4.3: Mobile Robot Architecture	33
4.4: Fitness Function Consideration	34
4.5: Summary	37

### **5: EXPERIMENTAL AND SIMULATION RESULTS**

5.1: Overview	38
5.2: Simulation Analysis	38
5.3: Experimental Analysis	41
5.4: Summary	44

### **6: CONCLUSION AND FUTURE WORK**

6.1: Conclusion	45
6.2: Future Scope	45

<b>7: REFERENCES</b>	<b>46</b>
----------------------	-----------

## LIST OF FIGURES

<b>TITLE</b>	<b>PAGE</b>
Fig. 1.1: Characteristics of Robot	1
Fig. 1.2: Multifaceted view of Robotics	2
Fig. 1.3: 6-DOF PUMA Robot [26]	5
Fig. 1.4: 6-DOF KUKA Robot [27]	5
Fig. 1.5: Mars Orbiter Mission Spacecraft of ISRO [28]	6
Fig. 1.6: Process of Mobile Robot Control	6
Fig. 2.1: Relative co-ordinate diagram of e-puck robot [1]	8
Fig. 2.2: Robots with obstacles in the arena [2]	9
Fig. 2.3: Robots approaching source [2]	9
Fig. 2.4: Multi robot navigation with irregular and structured obstacles [4]	10
Fig. 2.5: Components of Festo Robotino [6]	11
Fig. 2.6: Obstacle Matrix [21]	12
Fig. 2.7: Power Matrix [21]	12
Fig. 2.8: Comparison of paths generated by PSO and APSO [9]	13
Fig. 2.9: Mobile robot with static obstacle [19]	14
Fig. 2.10: Mobile robot with moving obstacle [19]	14
Fig. 2.11: Comparison of path lengths produced by PRM, PSO and NPSO [20]	15
Fig. 2.12: Motion planning system of mobile robot [8]	16
Fig. 2.13: Mobile robot with GPS antenna, magnetometer and LCD unit [13]	17
Fig. 3.1: Closed loop structure of mobile robot architecture	19



Fig. 3.2: Position of mobile robot from a reference plane	20
Fig. 3.3: Ultrasonic Sensor (HC-SR04)	21
Fig.3.4: Timing diagram of HC-SR04 [29]	23
Fig .3.5: Interfacing of HC-SR04 with Arduino	24
Fig. 3.6: Distance data in (inch, cm) of ultrasonic sensor module	24
Fig. 3.7: Cone pattern of Ultrasonic sensor	24
Fig. 3.8: Servomechanism of position control	25
Fig. 3.9: Servo motor	26
Fig. 3.10: Interfacing of Servo motor with Arduino	26
Fig. 3.11: Interfacing of Servo motor with ultrasonic sensor with Arduino	27
Fig. 3.12: Pin diagram of the Motor driver IC L293D	28
Fig. 3.13: Motor driver IC L293D	28
Fig. 3.14: Interfacing of DC geared motor with Arduino	29
Fig. 3.15: Interfacing of DC geared motor and ultrasonic sensor with Arduino	29
Fig. 3.16: Arduino Mega 2560 board	30
Fig. 4.1: Structure of Particle Swarm Optimization	33
Fig. 4.2: Random particles with one global best position	35
Fig. 4.3: Flowchart of the Algorithm	36
Fig. 4.4: 3D view of particles in search space	37
Fig. 5.1: Robots in absence of obstacles	38
Fig. 5.2: Robots approaching goal in the absence of obstacles	39
Fig. 5.3: Robots reached goal in the absence of obstacles	39
Fig. 5.4: Robots moving towards goal in presence of obstacles	40

Fig. 5.5: Robots approaching goal in presence of obstacles	40
Fig. 5.6(a): Avoidance of static obstacle	40
Fig. 5.6 (b): Avoidance of dynamic obstacle	40
Fig. 5.7: Robots avoiding static obstacles	41
Fig. 5. 8: Robots avoiding static obstacles and each other	43
Fig. 5.9: Robots avoiding static and mobile obstacles	44

## **LIST OF TABLES**

<b>TITLE</b>	<b>PAGE</b>
Table 3.1: Pin specification	22
Table 3.2: Electric specification	22
Table 3.3: Physical specification	22
Table 3.4: Servomotor specification	26
Table 3.5: DC geared motor specification	27
Table 3.6: Logic table for motor driver IC	28
Table 3.7: Specification of Arduino Mega 2560	30

## 1. INTRODUCTION

*“A robot may not injure humanity or through inaction, allow humanity to come to harm”.*

*“A robot may not injure a human being or through inaction, allow a human being to come to harm, unless this would violate a higher order law”.*

*“A robot must obey orders given to it by human beings, except where such orders would conflict*

*with a higher order law”.*

*“A robot must protect its own existence as long as such protection does not conflict with a higher order law”.*

**Laws of robotics by Isaac Asimov**

A robot can be defined as an electro-mechanical device or machine which can be controlled by a computer program or electronic circuitry. In a layman’s language, it is a device which performs human task thereby reducing human effort. The word robot can allude to both physical robots and virtual programming specialists, however the last are typically alluded to as bots. There is lots of definition of robot, but there are some important characteristics of the robot such as:

- i. Sensing: It should sense the environment around it with the help of various sensors like light sensors, touch sensors, sonar sensors, pressure sensors etc.
- ii. Locomotion: It should move within its workspace or environment with the help of wheeled motors, leg like mechanical arrangement or propelled by the thrusters.

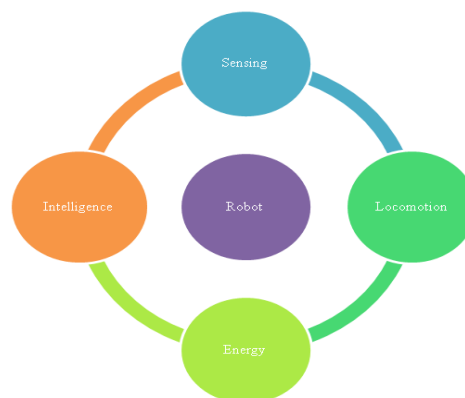


Fig 1.1 Characteristics of Robot

- iii. Energy: To perform the desired task, it should consume power. So it may be battery powered, electric powered or solar powered depending on the suitability.

- Domestic robots: Robots used in household applications like robotic vacuum cleaners, robotic pool cleaners, floor washing robots etc.
- Military robots: Robots used in the defence sector and military applications like bomb detection robotic vehicle, Unmanned Aerial Vehicle (UAV), Autonomous Underwater vehicle (AUV), reconnaissance drone etc.
- Space robots: These are special types of robots used in space shuttles, international space station, etc. Mars exploration rovers of NASA like curiosity, opportunity and sojourner are few examples of space robots.
- Entertainment robots: There are some robots used for entertainment purpose like Gupi (a robotic Pig), Aibo (robotic dog) etc.

There is another way of classification of robots based on their movement and kinematics such as:

- Stationary robots: Robots which are fixed at a particular position like industrial robots and manipulators.
- Mobile robots: Robots having ability to move around their surroundings and not altered to the physical area. Automatic Guided Vehicles (AGVs) are examples of mobile robots. Mobile robots can be classified as two types such as:
  - a. Legged robots: Robots having legs as actuators for their movement and locomotion.
  - b. Wheeled robots: Robots with wheels attached to the motors and the motors are controlled by the user through programming.

Advancement of technology made conceivable robots utilized in distinctive conditions like area, water and air. Some of the examples of it are Automatic Ground Vehicle (AGV), Unmanned Aerial Vehicle (UAV) and Autonomous Underwater Vehicle (AUV).

Classification of robots can also be done based on their method of control like

- Autonomous robots: These are the robots which work on their own without control of any external agent like human operator. This task is achieved through programming and implementation of suitable algorithm.
- Semi-autonomous robots: In case of autonomous robot all the desired tasks are performed by the robot itself and there is no need of external control. But a semi-automated robot performs some task continuously on its own while few things are

performed by the operator remotely through some mechanism. The best example is pipeline inspection robot.

- Manually controlled robots: Robots controlled by human operator through the connecting wire or by remote from a distant location.

Application of robotics is enormous which spans from agriculture to space explorations. Various fields of application of robotics include healthcare, research, agriculture, entertainment, space exploration, defence, search & rescue and radioactive material handling etc.

### **1.1 Origin of the work**

One of the first occurrences of a mechanical gadget constructed to consistently complete a specific physical assignment happened around 3000 B.C.: Egyptian water clocks utilized human dolls to strike the hour ringers. In 400 B.C., Archytus of Tarentum, designer of the pulley and the screw, additionally developed a wooden pigeon that could fly. Powerfully worked statues that could talk, gesture, and the prediction were regularly built in Hellenistic Egypt amid the second century B.C. In the first century A.D., Petronius Arbiter made a doll that could move like a person. Giovanni Torriani made a wooden robot that could get the Emperor's everyday bread from the store in 1557. Inventions in the field of robotics reached a relative peak (before the 20th century) in the 1700s; incalculable in genius, yet unrealistic, automata (i.e. robots) were made amid this time period. The 19th century was likewise loaded with new mechanical manifestations, for example, a talking doll by Edison and a steam-controlled robot by Canadians. Despite the fact that these creations, all through history may have planted the first seeds of motivation for the cutting edge robot, the exploratory advancement made in the 20th century in the field of mechanical autonomy surpass past progressions a thousand fold.

In the first century A.D., Petronius Arbiter made a doll that could move like an individual. The term "Robot" is first coined by a Czech writer Karel Capek in his book Rossum's Universal Robots in 1920. The meaning of robot in Czech is slave or servant. The earliest robots were made in the mid-1950s by George C. Devol, a creator from Louisville, Kentucky. He developed and patented a programmable manipulator called "Unimate," from "Universal Automation." For the following decade, he endeavored to offer his item in the business, however did not succeed. In the late 1960s, businessperson/engineer Joseph Engleberger

procured Devol's robot patent and had the capacity alter it into a modern robot and structure an organization called Unimation to deliver and business the robots. For his endeavors and victories, Engleberger is referred to in the business as "The Father of Robotics."

The scholarly world additionally gained much ground in the creation new robots. In 1958 at the Stanford Research Institute, Charles Rosen drove an exploration group in building up a robot called "Shakey." Shakey was much more progressed than the first Unimate, which was intended for particular, mechanical applications. Shakey could wheel around the room; watch the scene with his TV "eyes," move crosswise over new surroundings and to a certain degree, and react to his surroundings. He was given his name on account of his unbalanced and banging developments.



Fig.1.3 6-DOF PUMA Robot [26]

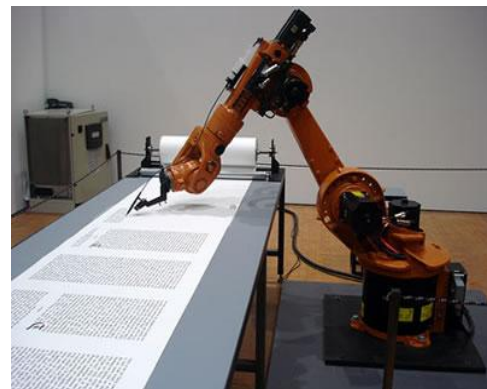


Fig.1.4 6-DOF KUKA Robot [27]

In 1978 the PUMA (Programmable Universal Machine for Assembly) is produced by Unimation with a General Motors configuration support. In 1980s the robot business enters a period of quick development. Numerous foundations present projects and courses in robotics. Robotics courses are spread across mechanical engineering, electrical engineering, and software engineering departments. From 1995 to till date developing applications in small robotics and portable robots drive a second growth of new businesses and exploration.

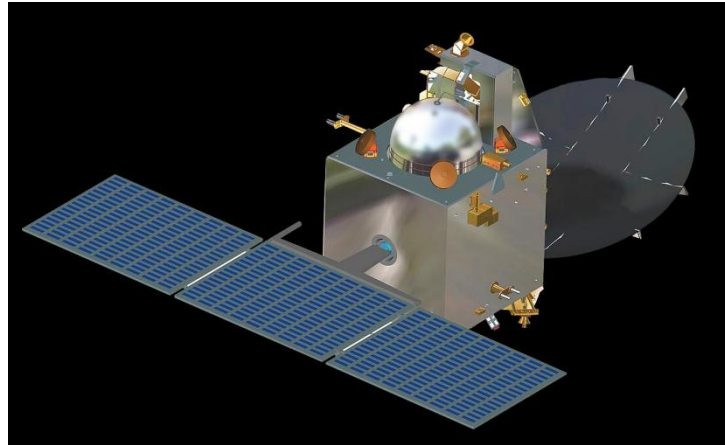


Fig.1.5 Mars Orbiter Mission Spacecraft of ISRO [28]

In 2003 launch of Mars exploration rover by NASA was a major breakthrough in the field of space exploration and robotics. After that there are lots of contributions made by different organizations all over the world in different fields of automation starting from automated stitching machine to interplanetary mission. Recently the successful launch of Mars Orbiter Mission also known as Mangalyaan by Indian Space Research Organization (ISRO) is remarkable and significant.

Over recent year's mass measure of innovative work have focused on stationary mechanical controllers as a result of their modern applications. Less exertion has been made on mobile robots.

Path planning and navigation of mobile robot is considered as one of the important aspect in the field of robotics. Smooth and accurate motion planning of mobile robot in real time environment had been a challenge for researchers from decades. Various features of a mobile robot comprises of: Sensory perception, Position determination or localization, Decision making and Motion control.



Fig.1.6 Process of Mobile Robot Control

At first, the robot will get the information about its surroundings through the mounted sensor module (ultrasonic/infrared/camera/laser). The robot will find its location in the global environment, using GPS and position encoder which is called localization. After that it will decide what task needs to be performed as per the program defined by the user and then the



actuators (motors) are controlled by the microcontroller which is nothing but the locomotion of the mobile robot.

## **1.2 Problem Statement**

Autonomous navigation and control of mobile robot is one of the control challenges in the field of robotics. Autonomous navigation involves important aspects like obstacle avoidance, accurate localization in the environment, perception of goal etc. Employment of multiple robots is more advantageous than single robot. So multi-robot cooperation is an important issue in robot navigation problem. Moreover the use of multiple robots makes this control problem more complex and challenging.

## **1.3 Objective**

The main aim of the project is to develop control strategies for cooperation of multiple robots in dynamic environment. This includes tasks like

- Interfacing of different sensors and actuators module.
- Integration and synchronization of on-board sensors and actuators.
- Formulation and development of algorithm for obstacle avoidance and control.

## **1.4 Thesis Overview**

This thesis is framed in the following way:

*Chapter 2* depicts various literatures on mobile robot path planning. It also focuses on multi-robot cooperation in different scenarios like target searching, disaster control etc. Path planning of mobile robots using various algorithms like Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and their variants also discussed.

*Chapter 3* provides the mechanical architecture of mobile robot. It describes the differential drive mobile robot platform and addresses the different component required to develop a simple mobile robot. It also provides relevant information on sensors like ultrasonic sensor (HC-SR04), actuators like servomotor (futaba S3003) and Arduino microcontroller used for controlling the mobile robot.

*Chapter 4* deals with PSO based architecture for mobile robot. It depicts the implementation of Particle Swarm Optimization (PSO) in robot path planning. Fitness function generation for mobile robot navigation is also addressed.

*Chapter 5* outlines the experimental and simulation results for mobile robot path planning. It highlights the obstacle avoidance behaviour of the mobile robot.

*Chapter 6* devotes to the conclusion and future scope of the research dissertation.

## 2. LITERATURE SURVEY

This chapter describes the research done on various methods for mobile robot navigation and control.

### 2.1 Overview

The motion planning of mobile robot is of paramount interest in the field automation and robotics. Automatic navigation of robotic vehicle saves unnecessary wastage of time and money. Immense contribution has been done by various researchers on mobile robot path planning. A lot of algorithms for path planning of mobile robot have been proposed by various researchers in the past. Some of the popular methods are like fuzzy logic, neural network, genetic algorithm, cell decomposition, visibility graph, potential field method and many more. Among these popular algorithms Particle Swarm Optimization (PSO) is one of the most popular and deserving algorithm for its simplicity and better convergence. Different robot path planning algorithms are reviewed in this chapter.

### 2.2 Implementation of Particle Swarm Optimization (PSO) in multi-robot cooperation

Search and rescue operation using autonomous robots is a crucial task in robotics field. But performing search operation by employing multiple robots is a promising and good idea. On the other hand it makes the process complex and more challenging. So multi-robot coordination is an important issue which requires a huge attention. Jim et. al. [1] did research on searching using multiple robots with Particle Swarm Optimization (PSO). They used virtual search agent for finding optimum value in multidimensional function space using PSO. They performed simulation using Webots robotic simulator. The simulation was performed using the e-puck robot.

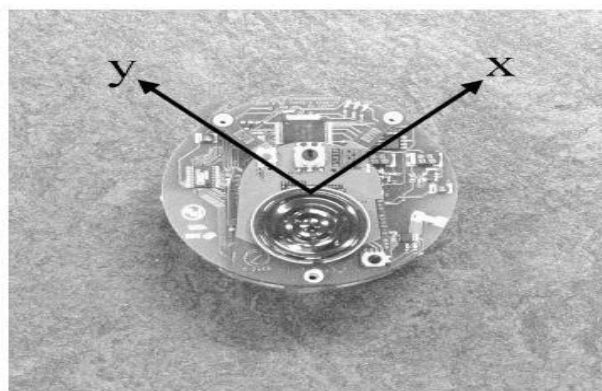


Fig.2.1 Relative co-ordinate diagram of e-puck robot [1]

For better knowledge about the environment higher value of inertia ( $w$ ) parameter is considered. Experiments is performed in two phases; in first phase they used multiple robots and GPS (Global Positioning System) to track each robots position while in the second phase they performed experiment with multiple robots without GPS by considering the relative position of robots with respect to a frame of reference.

Identifying the source of signal in dynamic cluttered environment is one of the most challenging and complex problem. The complexity becomes more due to randomness and interference of noise in the environment. Source seeking behaviour of swarm of mobile robots using Particle Swarm Optimization is described by Rui et. Al. [2]. Apart from this the obstacle avoidance nature of mobile robot is also discussed in this paper.



Fig.2.2 Robots with obstacles in the arena [2]

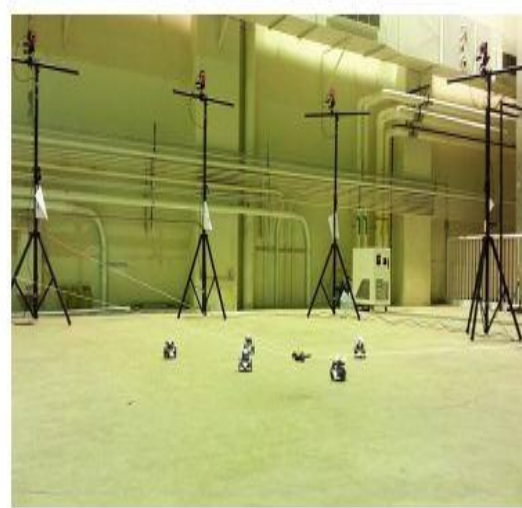


Fig.2.3 Robots approaching source [2]

Any kind of signal such as acoustic signal, thermal signal, electromagnetic signal or chemical signal may be the source but in this research article electromagnetic source signal is considered. The experiment is carried out indoor and Vicon tracking system is used for getting the information regarding position of the mobile robots. The electromagnetic source used was an XBee module.

Coordinating multiple robots for target searching in unknown environment is complex task. Yifan Cai and Simon X. Yang [4] presented a modified PSO algorithm known as Improved

Potential Field Based Particle Swarm Optimization (IPPSO) for target searching in unknown environment.

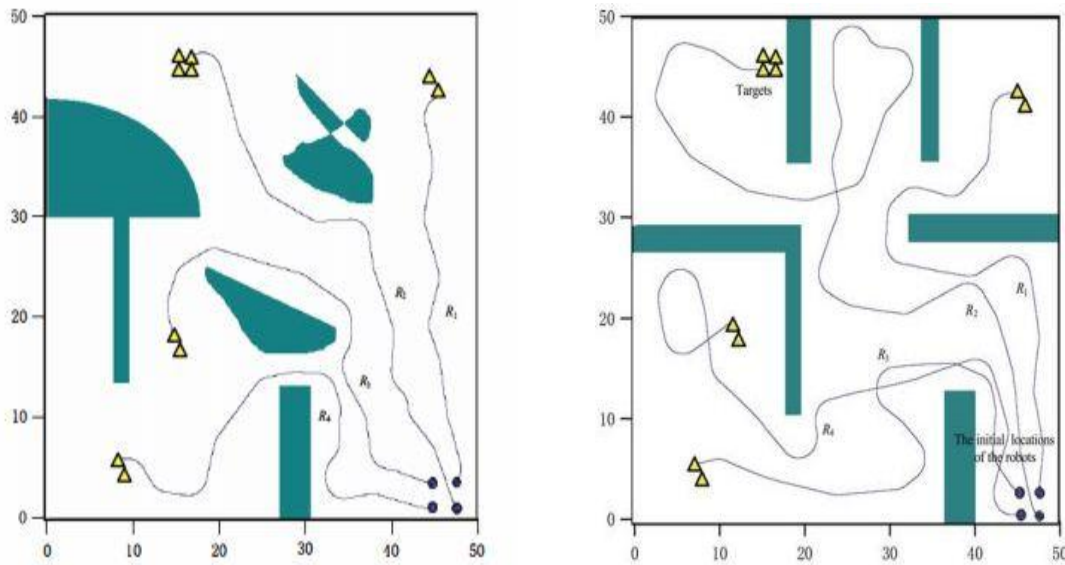


Fig.2.4 multi robot navigation with irregular and structured obstacles [4]

Establishment of coordination among all the mobile robots by varying the dynamic weights is the crux of the research article. For each robot target is different. The challenging task was dealing with obstacles of irregular shape. Simulation results were provided to validate the claims.

Target searching in unknown environment has been a major challenge in robotics. Cooperation among multiple mobile robots for target searching using potential field based particle swarm optimization (PPSO) is represented by Yifan Cai and Simon X. Yang [5]. The potential field function was considered as the fitness function for the PSO.

Tang and Eberhard [6] proposed multi robot searching using particle swarm optimization. The advantage of this method was the task is completed by the other members of the swarm if some robots were dead or struck somewhere due to environmental noise, uneven terrain, power failure or communication gap. For performing the experiment they used Festo Robotino which is an omni-directional robot capable of rotating 360 degrees. Total 20 numbers of robots were used. The different parts of robot are shown below in the figure:



Fig.2.5 Components of Festo Robotino [6]

Lisa et. Al [7] presented the PSO-CRS algorithm for target searching as well as obstacle avoidance. They discovered the obstacle avoidance and collective robot search behaviour of the algorithm. This indicates the faster convergence of PSO algorithm.

Multiple target searching using more than one mobile robot had been represented by Kurt and Milos [16]. The research was done on exploration of an unknown area by multiple mobile robots using Particle Swarm Optimization (PSO). The target locations were equipped with mobiles which transmitted the radio frequency (RF) signals. Based on received signal strength approach each mobile robot moved towards its goal position.

Gustavo et. Al [12] proposed multi robot coordination in natural disaster (forest fire) situation. The main task of the team of mobile robot was to stop the fire from being spread. Several parameters of the PSO had been changed like swarm size, inertia, weight etc. to get the desired result.

Sanjay et. Al [21] presented a novel implementation of PSO algorithm for path planning of multiple robots. The considered obstacles in this paper were static obstacles. The swarm of robots moved towards the target seeking the light (potential field) in this case. The main drawback was though light source is considered as a potential field in this case but all the laws of optics were not considered. The swarm of robots may strike at cavity points. All the obstacles were categorised as obstacle matrix which contained both obstacle and free space. Actually the whole experimental arena was divided into obstacle matrix. Similarly to move the robots towards the destination, a light source is kept at the destination.

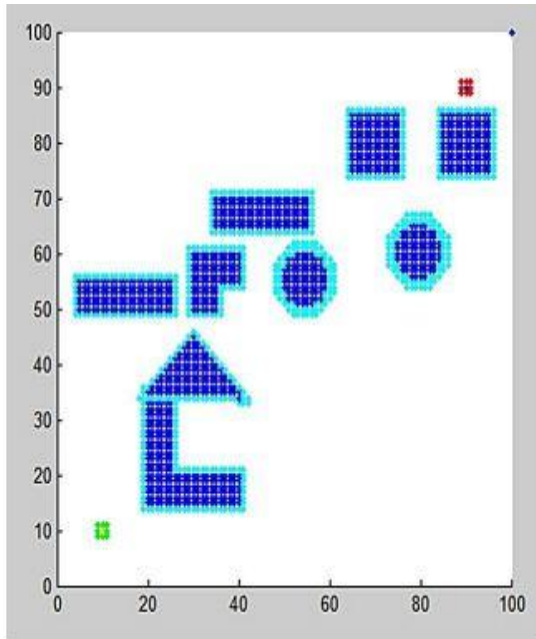


Fig. 2.12 Obstacle Matrix [21]

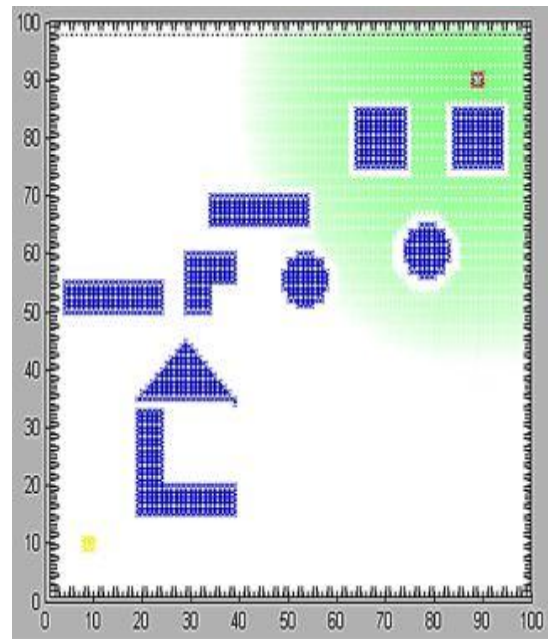


Fig. 2.13 Power Matrix [21]

A power gradient is formed to move the robot towards the destination that means all the robots would move towards increase in the intensity of power source. Since the whole area is considered as the obstacle matrix the entries of the obstacle matrix were “1” when there is obstacle and “0” when there is no obstacle. To increase the affinity of mobile robots towards the goal point the area again converted to power matrix. When it changed from obstacle matrix to power matrix the free space occupied “0” value entries were replaced by some values while the destination got the value of “1”. The simulator was tested using various configurations for static obstacles.

### 2.3 Motion planning of mobile robot using Particle Swarm Optimization (PSO) and its variants

Mobile robots can be used for surveillance and infiltration detection. Hence fast movement of mobile robot with smooth obstacle avoidance is an important topic. Abdullah et. Al [9] described a faster path planning technique for mobile robot using a modified form of Particle Swarm Optimization.

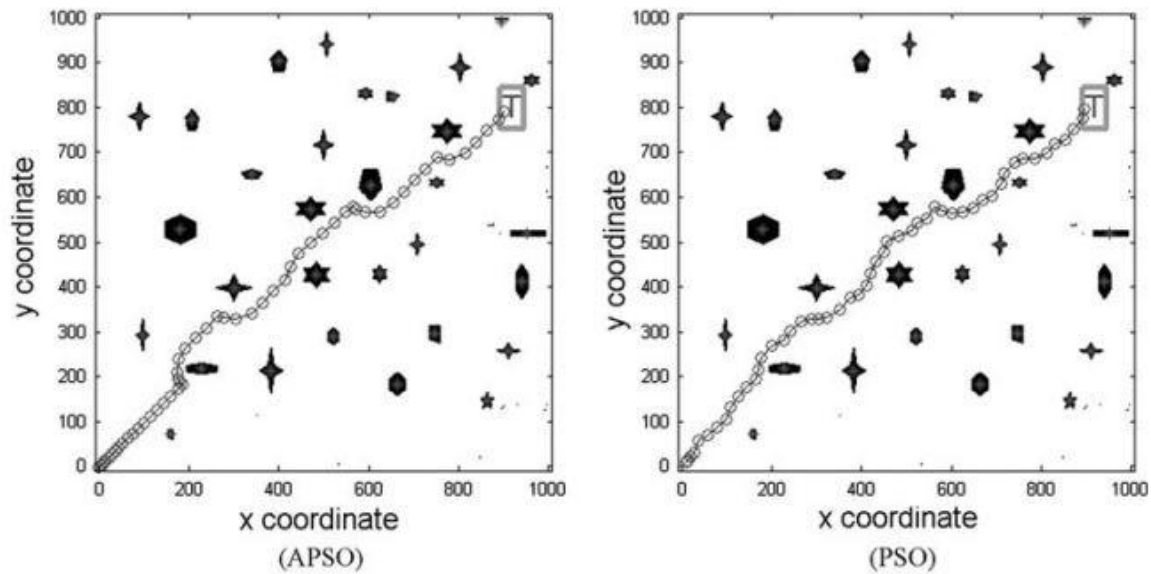


Fig.2.7 comparison of paths generated by PSO and APSO [9]

The position and velocity equation of PSO is modified and merged to a single equation to enhance the speed of the process. The process is simulated for both static and dynamic obstacles that mean the obstacles whose position changes with time. The comparison of PSO and APSO (Accelerated PSO) is performed.

Deepak et. Al [11] presented a PSO based path planner for mobile robot. Path planning of mobile robot is done using static obstacles but the environment was unknown. A novel fitness function had been generated in this research article. Smooth trajectories of robot from source to goal position were developed by proper tuning of parameters of PSO.

Nadia and Lana [10] modified the particle swarm optimization technique to improve the convergence time of the algorithm. This method was successful in making the robot to move from source to goal position. The most amazing part of the algorithm was it couldn't converge when there is no path between source and destination.

Navigation of autonomous mobile robot from source to destination position is a popular research area. Deepak et. Al [15] described a PSO based motion planner for mobile robot in unknown environment. One of the disadvantages was feasible trajectories of robot couldn't be developed for obstacles of irregular size.

Torvald and Xiaoming [17] considered the robot path planning problem an optimization problem where the smooth and safest path is the path where the cost function is minimum.

The method used virtual vehicle approach. The method was applicable for both unknown and known environments.

Robot navigation problem can be considered as a multidimensional optimization problem. Sedigheh et. Al [18] used PSO for path planning of mobile robot in unknown environment. The paper illustrates how a mobile robot moves from source to goal through a series of local goal position. The local goal positions are nothing but the intermediate global best positions of the mobile robot.

Maryam et. Al [19] used particle swarm optimization method for path planning of mobile robot in unknown environments

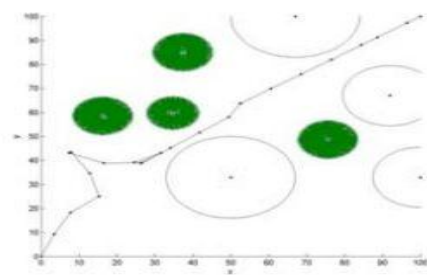
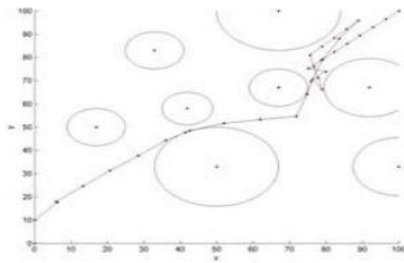


Fig.2.9 mobile robot with static obstacle [19] Fig.2.10 mobile robot with moving obstacle [19]

Both static and dynamic obstacles were considered in the research article. One important aspect of the proposed research article was the target was also movable.

Martina et. Al [22] proposed path planning of mobile robot using particle swarm optimization (PSO) with Ferguson splines. The splines were used to model the path of the mobile robot from starting point to ending point. The different parameters of the spline were controlled to get the smoothest and safest path without colliding with obstacles. PSO is used to guide the robot from starting point to destination point.

Aleksandar et. Al [25] presented motion planning of mobile robot in structured environment with static obstacles. The considered obstacles were circular in shape. The proposed approach was divided into several steps. In the first step PSO was used to generate collision free paths while in the second step interpolation (finding the smoothest path) of collision path is done. In the third part of the problem trajectory tracking issue is taken care using proportional-integral (PI) controller. An additional fuzzy controller was also included to avoid the uncertainty of the PI controller.



Ellips and Davoud [20] presented two novel Particle Swarm Optimization (PSO) based algorithms for path planning of mobile robot. The algorithms fulfilled two very important objectives such as shortest path and smoothest path. The first algorithm was a blend of PSO and PRM (Probabilistic Roadmap Method) where PSO act as global path planner while PRM works as local path planner. The second algorithm was the combination of PRM and negative-PSO (NPSO).

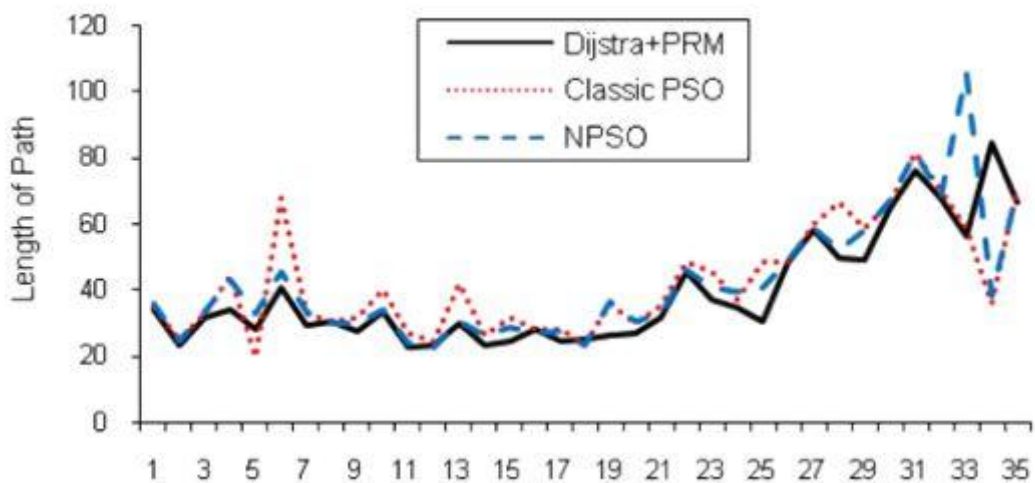


Fig. 2.11 comparison of path lengths produced by PRM, PSO and NPSO [20]

The difference between PSO and negative-PSO is that PSO determines the best position of the particle and proceeds accordingly while negative- PSO calculates the negative of worst particle position and proceeds. The path generated was smoother and shorter compared to PSO algorithm which indicated NPSO is better approach than PSO. The algorithm combined PSO, PRM and NPSO, PRM and found the best position of the particles thereby the intermediate positions of robot. This method also generated number of nodes around obstacle vertices and PRM methods are used to find the shortest path. The considered algorithms were faster than the PRM methods.

The motion planning of mobile robot in cluttered environment is tested using two different algorithms such as Particle Swarm Optimization (PSO) and Genetic Algorithm (GA). Hosam et. Al [3] performed the simulation analysis of motion of mobile robot using the above mentioned methods. For controlling the motion of the mobile robot an Advanced Artificial Potential Field (AAPF) based controlled is designed. The simulations are performed using static that is immobile obstacles. Experimental results showed that PSO was faster and better

in obstacle avoidance than GA while the latter gives better smoothness values than the earlier.

## 2.4 Path Planning Of Mobile Robot Using Genetic Algorithm

Adem et. Al [8] proposed motion planning of mobile robot using Genetic Algorithm (GA). Camera is used for locating the positions of mobile robot, target and obstacle.

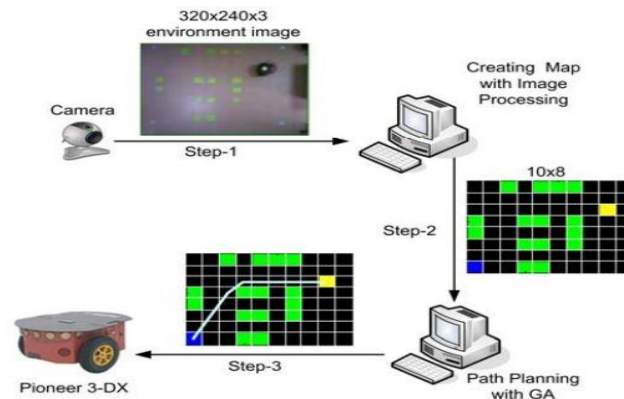


Fig.2.6 Motion planning system of mobile robot [8]

A graphic user interface (GUI) was developed for user interaction with robotic system. For performing the experiment the Pioneer P3-DX mobile robot is used.

Nadia and Lana [10] proposed path planning of mobile in static environment using modified version of Genetic Algorithm. The proposed problem was a global path planning problem where the robot has information about its surroundings. Only simulation has been done using MATLAB software. The modified genetic algorithm made the mobile robot to move from starting position to destination position without colliding any obstacle and followed the shortest path. Use of adaptive population size enhanced the execution and computational cost.

Panigrahi et. Al [13] represented three optimization methods for path planning of mobile robot. Apart From Particle Swarm Optimization they used Gravitational Search Algorithm (GSA) and Simulate Annealing (SA) method. Experiments were carried out using non-holonomic differential drive robot. Robot positions were taken care using GPS and movement of robots towards the goal was controlled using magnetometer HMC 5883L.

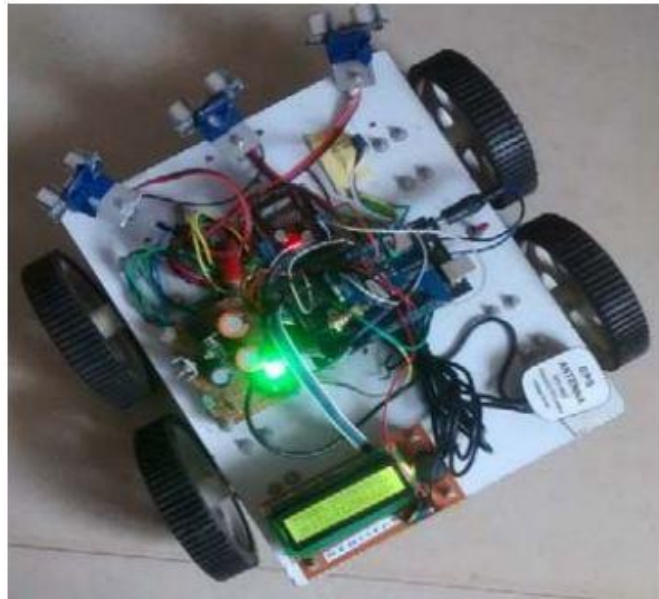


Fig.2.8 mobile robot with GPS antenna, magnetometer and LCD unit [13]

The mobile robots were developed using Arduino microcontroller. Obstacle avoidance is achieved through HC-SR04 ultrasonic sensor module. Due to less computational time and shortest path length Gravitational Search Algorithm (GSA) was inferred as the best among three methods.

### **2.5 Mobile Robot Navigation Using Reactive Algorithms**

Andrey and Chao [23] proposed a reactive algorithm for path planning of mobile robot in cluttered environment. The algorithm worked well for both static and moving obstacles. The experiment was performed by taking the integrated view of the work space. One of the important features of the research is obstacles with any shape could be avoided by the robot and no need of tracking the velocity of the obstacles. Dubins car model for the mobile robot is chosen in the experiment. The experiment was carried out by taking the Pioneer 3-Dx differential drive robot. The robot was equipped with SICK range finder and 16 ultrasonic sensors at front and rear.

Ioan et. Al [24] proposed a novel reactive algorithm called Bubble rebound algorithm for obstacle avoidance and goal seeking of mobile robot. The area of detected obstacles in this method is called “Sensitivity Bubble”. The method is that when the robot detects obstacle it rebounds (changes its direction) to a direction with less obstacle density, moves and then heads towards goal. The process continues until the robot reaches the target position. The

main positivity of the algorithm was it could be used with low cost Infrared or ultrasonic sensors. The Pioneer 3-Dx robot was used for the experiment and simulation was performed using Mobilesim.

## **2.6 Summary**

Motion control and navigation of mobile robot in various environmental conditions like static, dynamic, partially known environment were performed in the above mentioned literatures. Various researchers used different algorithms for smooth and safe path planning of mobile robot from source to destination. Multi-robot cooperation in different environmental conditions was discussed. The advantages of multi-robot operation over single robot described. Obstacle avoidance criteria for structured, random shape, static and dynamic obstacles have been discussed.

### 3. MECHANICAL SYSTEM ARCHITECTURE FOR MOBILE ROBOT

#### 3.1. Overview

System architecture for mobile robot basically refers to the integration of both software and hardware for control and navigation of robot. The software part includes the autonomous control of the mobile robot through programming whereas the hardware part includes various sensors, actuators for perception, locomotion and task performance. Moreover, we can say that the software part of the architecture is similar to human brain, which takes the decision and the hardware part is nothing but the combination of sensory organs, limbs used for sensation of surrounding and locomotion.

#### 3.2. Mobile Robot Positioning

The mobile robot is having different parts or assemblies such as a sensor module for perception, an actuator module for movement, control unit or the controller for motion control. In this research dissertation, the considered mobile robots have two wheels connected with two DC geared motors, one castor wheel, one servo motor and one ultrasonic sensor. The considered robots are differential wheeled robots which mean the manoeuvrability of these mobile robots depends on the relative motion between two wheels placed on either side of the robotic platform. Thus the robots can accordingly alter their course of movement by differing the relative rate of movement of their wheels and henceforth does not require an extra guiding movement. The whole process of motion control of mobile robot is a closed loop process which can be shown as follows:

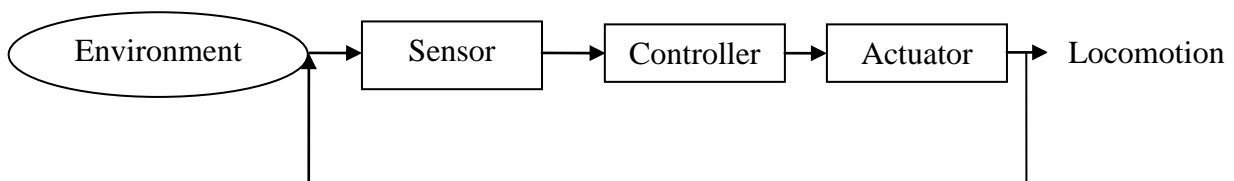


Fig. 3.1 Closed loop structure of mobile robot architecture

Let us consider a robot moving in a plane. The position of the robot can be represented by establishing the relation between the global and local frame of reference. The robot can be represented as a point with co-ordinates  $(x, y)$  in 2D. The movement of the robot from one point to another point can be represented as follows:

Translation Equation:

The translation of a point from one position  $(x, y)$  to another position  $(x', y')$  can be done as follows:

$$x' = x + \Delta x \quad (3.1)$$

$$y' = y + \Delta y \quad (3.2)$$

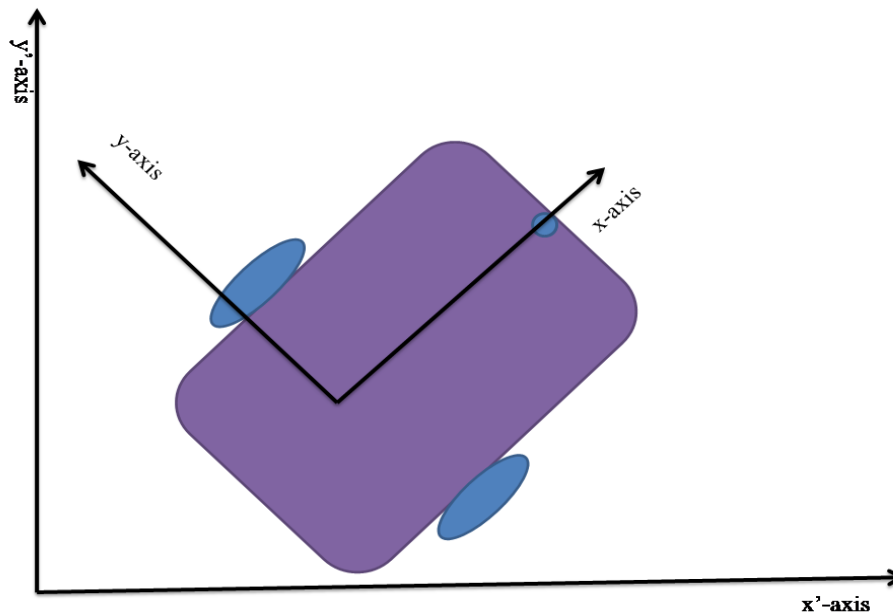


Fig. 3.2 Position of mobile robot from a reference plane

Rotation Equation:

The rotation of a point or a line from a position  $(x, y)$  to another position  $(x', y')$  by an angle  $\alpha$  can be done as follows:

$$x' = x \cdot \cos(\alpha) + y \cdot \sin(\alpha) \quad (3.3)$$

$$y' = -x \cdot \sin(\alpha) + y \cdot \cos(\alpha) \quad (3.4)$$

### 3.3. Range Sensor module specification and Working

It is the onboard sensor module which gives the information about the surroundings, thereby helps the robot in identifying objects nearer to it. There are various kinds of sensors such as a

proximity sensor, ultrasonic sensor, IR sensor and LASER sensor which can give the distance between the robot and the obstacle in front of it.

The sensor which is considered in this project is ultrasonic sensor HC-SR04. It determines the distance of the object like the way bats and dolphins do.

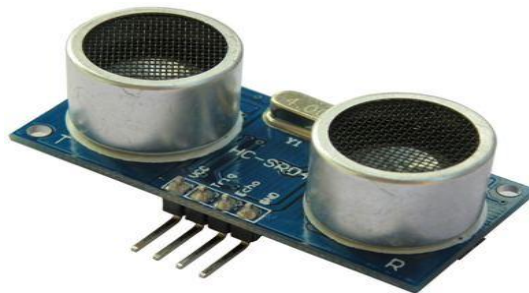


Fig. 3.3 Ultrasonic Sensor (HC-SR04)

Ultrasonic sensor works on the same principle of sonar which calculates the distance of the target by evaluating the duration of echo pulse that got reflected from the target. It offers amazing non-contact range recognition with high precision and stable readings in a simple to-utilize bundle. Its operation is not influenced by daylight or dark material like Sharp rangefinders are (although acoustical soft materials like cloth can be difficult to detect). It comes complete with ultrasonic transmitter and receiver module.

The ultrasonic sensor module can accurately measure the distance of up to 2cm-400cm. Its minimal size, higher reach and simple convenience make it a convenient sensor for separation estimation and mapping. The sensor module can be easily interfaced to any micro controller where triggering and measurement can be done using two pins. The sensor transmits an ultrasonic wave and produces an output pulse which corresponds to the time required for the echo to receive. By measuring the echo signal pulse width the distance of the target can be measured.

#### Pin Configuration:

With the sensor oriented as shown alongside, locate Pin-1 as the 1st pin (refer figure 3.3) on the left hand side.

Table 3.1 Pin specification

Pin No.	Signal
1	VCC (5V supply)
2	Trigger Pulse Input
3	Echo Pulse Output
4	GND (0V)

Electric Specification:

The electrical specification of the sensor module which is used in the research work is as follows:

Table 3.2 Electric specification

Parameter	Specification
VCC	4.8V to 5V (typical) 5.5V (Absolute maximum)
Operating Current	15mA
Temperature	0° to 60° Ambient room temperature
Trigger Input Pulse	10uS TTL pulse (5V, ± 2%)
Echo, Output Signal	TTL (0V/5V, ± 2%)

Physical specifications:

Due to the compact size it is one of suitable sensor modules which can be easily mounted on a mobile platform. The physical specifications of the sensor module can be specified as follows:

Table 3.3 Physical specification

Parameter	Specification
Dimensions	45 x 20 x 15 mm
Pin-out Pitch	2.54mm male berg
Interface	VCC, GND, SDA,SCL



### Operation:

The operation of the ultrasonic sensor can be easily understood from its timing diagram which is shown below:

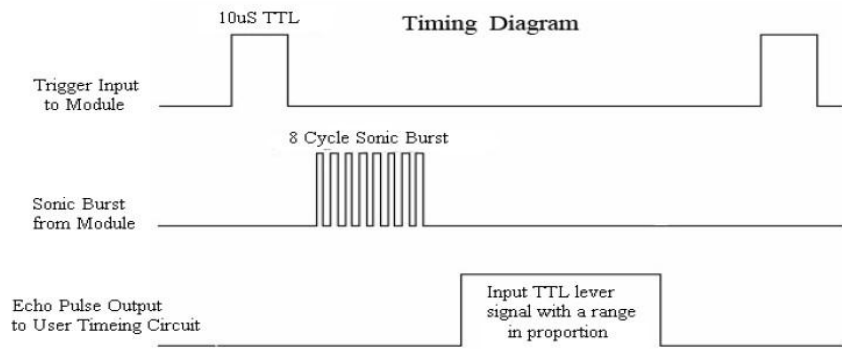


Fig.3.4 Timing diagram of HC-SR04 [29]

For accurate distance measurement, first of all the trigger pins should be provided with a pulse of 5V (HIGH) for a duration of 10 $\mu$ S. This will make the sensor to transmit 8 cycle of ultrasonic burst at 40 kHz and wait for the reflected ultrasonic pulse. When the sensor detects ultrasonic from the receiver, it sets the Echo pin to high (5V) and delay for a period (width) which proportion to distance. To obtain the distance, measure the width of Echo pin.

Time = Width of Echo pulse, in  $\mu$ S (micro second)

Distance in centimetres = Time / 58.

Distance in inches = Time / 148.

Or we can make use of the speed of sound, which is 340m/s.



Fig.3.5 Interfacing of HC-SR04 with Arduino

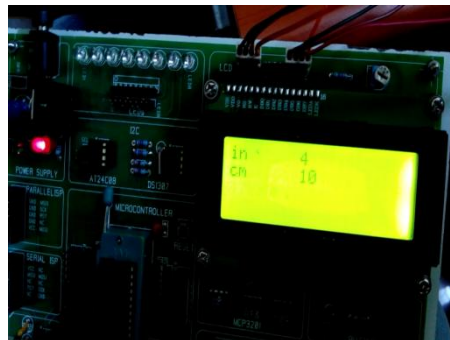


Fig.3.6 Distance data in (inch, cm) of ultrasonic sensor module

The distance sensor HC-SR04 works by sending out a pulse of ultrasonic sound and measuring the duration of time it takes for the sound to come back. This method is also known as the ‘time of flight’ method. The sound goes out in a cone pattern on either side of the centre position and detection is done. Best detection is done 15 degrees on either side of the sensor module. However, objects slight outside the cone can also be detected. The cone pattern for the ultrasonic sensor module can be shown below:

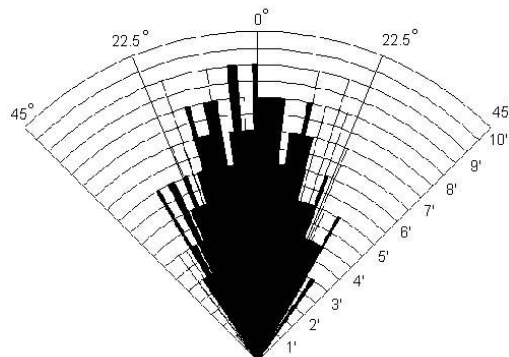


Fig.3.7 Cone pattern of Ultrasonic sensor

Since the sensor works with sound signal, it works best on objects that easily reflect sound. Flat, smooth surfaces reflect good while irregular surfaces reflect the worst.

While moving forward, the robot needs to avoid objects in all directions. Hence the ultrasonic sensor can't be fixed at a particular position. If we use more than one ultrasonic sensor then more obstacles can be detected while at the same time the complexity of the circuit increases and the robot becomes bulky. To overcome this, the ultrasonic sensor can be mounted on a servo motor which can move to a particular angle given by the user.

### 3.4. Actuators for locomotion

An actuator is a device which converts different forms of energy into motion. It is a mechanical device which takes energy as input and gives output as motion, force, etc. the various forms of energy may be hydraulic, electric or pneumatic. In the robotics field mostly motors play the role of actuators. There are various kinds of motors and accordingly it has numerous classifications. However, in robotics field three types of motors are used such as DC geared motors, servo motors and stepper motors.

Geared motors are the motors which rotate continuously. These motors have two wires, one for power and another for ground. It rotates continuously until there is supply and stops when power is removed.

Servo motors are the motors which rotate to a particular angular position. It works on the principle of closed loop servo mechanism. It is generally a combination of four things such as a DC motor, a gearing set, a controller and a position sensor. Here instead of controlling the motor by applying variable input, the device is controlled by a feedback signal generated by comparing output signal and a reference signal.

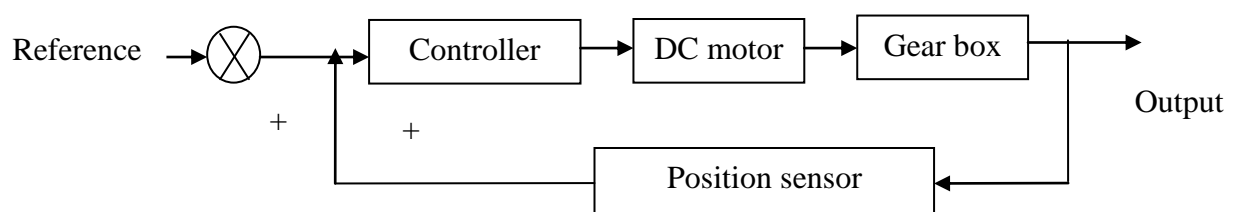


Fig.3.8 Servomechanism of position control

When input signal is given to the system the output signal is measured with a sensor and compared. The difference of the signal value is given to the controller which controls the

device in such a manner that the difference becomes zero. The feedback process continues unless the difference becomes zero.

The stepper motor is nothing but a servo motor which moves to a particular angular position through an equal number of finite steps.

To detect obstacles in front of the robot the ultrasonic sensor should be capable of detecting obstacle full 180 degrees. Hence it is mounted on a servo motor and the motor is programmed to rotate up to 180 degrees.



Fig.3.9 Servo motor

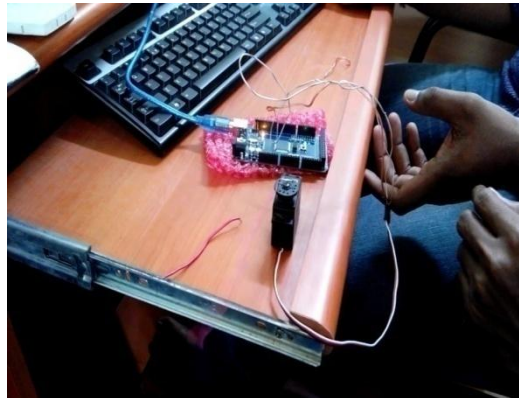


Fig.3.10 Interfacing of Servo motor with Arduino

Specification of the servo motor

Table 3.4 Servomotor specification

Parameters	Range
Operating Voltage	4.8-6.0V
PWM Input Range	Pulse Cycle $20 \pm 2$ ms, Positive Pulse 1~2ms
STD Direction	Counter Clockwise / Pulse Traveling 1500 to 1900 $\mu$ sec
Stall Torque	3Kgf.cm (41.3 Oz/in) at 4.8V, 3.2 Kgf.cm (44 Oz/in) at 6V
Operating Speed	0.2 Sec/ 60° at 4.8V, 0.18 Sec/ 60° at 6V at no load
Weight	38g (1.27 Oz)
Size	41.3*20.3*38.7*48.5*10
Plug Available	FUT, JR
Special Feature	Heavy Duty Plastic Gears, Economy Servo

For the 180 degree view of obstacle in front of the robot the ultrasonic sensor HC-SR04 is mounted on the servo motor which is programmed to rotate from 0 to 180 degrees. The mounting of ultrasonic sensor for servo motor and the interfacing of the assembly is shown below:

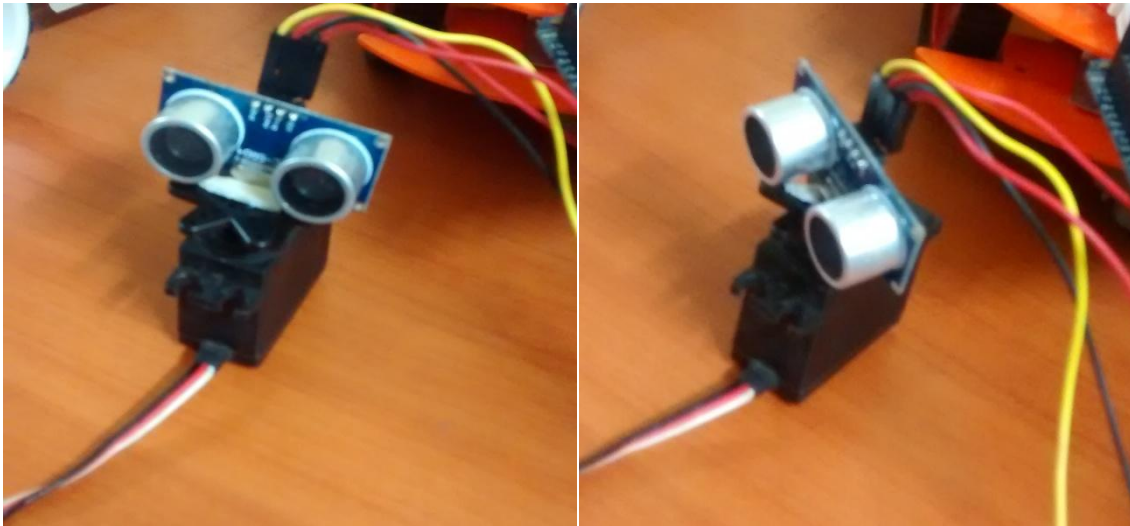


Fig.3.11 Interfacing of Servo motor with ultrasonic sensor with Arduino

For the movement of the robot two 150 RPM, 12V DC geared motors are used which are programmed with Arduino microcontroller. The specifications of the DC geared motors are as follows:

Table 3.5 DC geared motor specification

Parameters	Range and values
RPM	150
Voltage	12V
Shaft dia.	4mm
Weight	125gm
Torque	2kgcm
No-load current (max.)	60mA
Load current (max.)	300mA

The interfacing of DC geared motor with Arduino microcontroller can be done using motor driver IC (L293D) whose pin diagram is shown below:

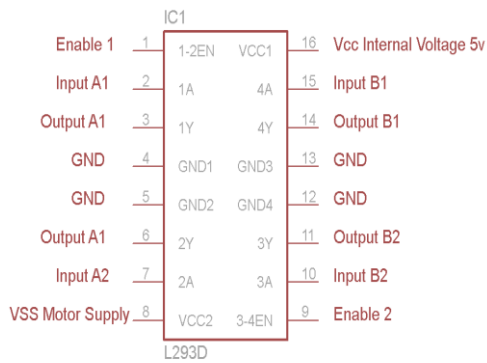


Fig.3.12 Pin diagram of the Motor driver IC L293D

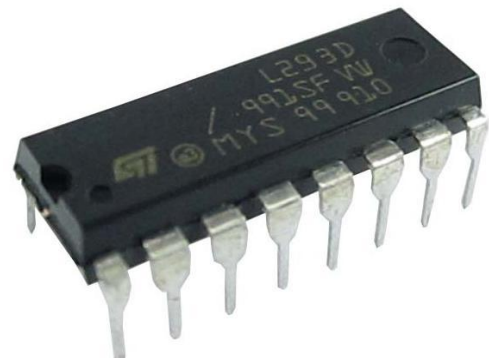


Fig.3.13 Motor driver IC L293D

When both the inputs to motor driver is of same logic value ((0,0) or (1,1)) the motor stops, whereas different input ((0,1) or (1,0)) to the motor driver makes the motor rotate in a particular direction. Similarly, by changing the polarity or the logic value makes the robot to rotate in different directions. The detailed explanation of logic control of motor in different directions is given below in the table

Table 3.6 Logic table for motor driver IC

INPUT 1	INPUT 2	ENABLE 1,2	Result
0	0	1	Stop
0	1	1	Anti-clockwise rotation
1	0	1	Clockwise rotation
1	1	1	Stop

Generally, the DC geared motors are two-wired motors, one for power and another for ground. The interfacing of DC motor with Arduino is shown below:

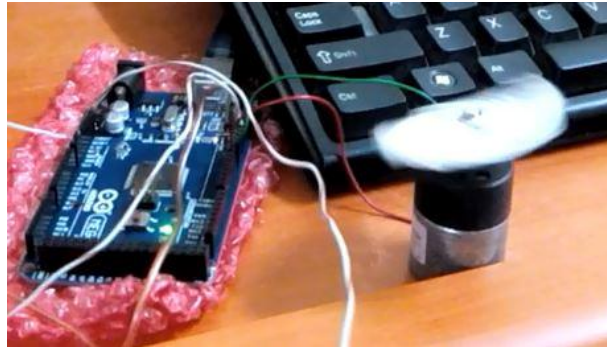


Fig.3.14 Interfacing of DC geared motor with Arduino

The controlling of movement of robot in front of obstacle is a crucial part. So initially the motor is controlled with respect to the distance value obtained by the ultrasonic sensor. The interfacing of ultrasonic sensor and motor with Arduino is shown below:

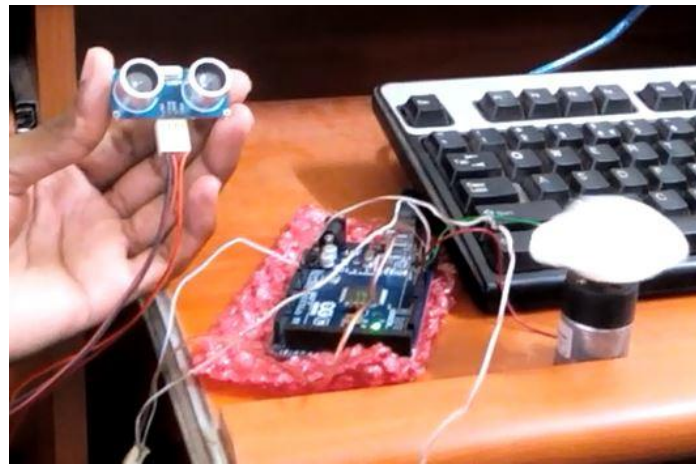


Fig.3.15 Interfacing of DC geared motor and ultrasonic sensor with Arduino

### **3.5. Controller for Mobile Robot**

The controller is the prime part which plays a vital role in any field of automation and control. Generally, all the tasks performed by the mobile robots are controlled by a microcontroller. There are different types of microcontroller available in the market. Some of the popular microcontrollers used in research are ATmega, PIC, 8051 etc. In this project, an open source hardware controller known as “Arduino” is used which uses the ATmega microcontroller.

Arduino is an open source hardware assembly which uses an onboard ATmega microcontroller and programming is done using its open source IDE. There are some features of Arduino which made it popular such as inexpensive, cross-platform, simple programming



environment, open source and extensible software, open source and extensible hardware, etc. There are different types of Arduino boards are available like nano, Mega 2560, Micro, UNO and many more. In this project the Arduino mega 2560 board is used. The schematic diagram of Arduino mega 2560 board is shown below:

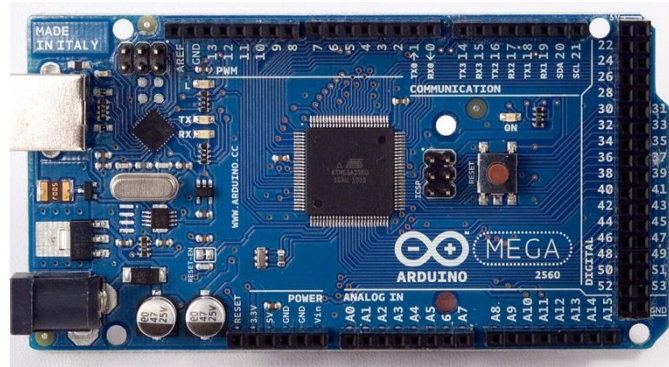


Fig.3.16 Arduino Mega 2560 board

Arduino Mega 2560 board has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. The detailed specification of Arduino Mega 2560 is shown below in the table:

Table 3.7 Specification of Arduino Mega 2560

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader



SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

### **3.6. Summary**

The interfacing of ultrasonic sensor is performed to find the distance of objects and it gave accurate results. Control of a servo motor is done using Arduino and two DC geared motors also interfaced with Arduino. Control of motor as per the data received from the ultrasonic sensor is done.

## **4. PSO BASED SYSTEM ARCHITECTURE FOR MOBILE ROBOT**

### **4.1. Overview**

Robot path planning in the unknown dynamic environment has been a challenge for the researchers from the last decades. The solution of this kind of problem plays a crucial role in the field of robotics. The main task of path planning of robot is to move the robot from its source to a goal position while avoiding obstacles and follow the optimum path. It's an optimization problem because it involves the generation of optimal collision free trajectories by the mobile robot. There are two types of path planning problem such as global path planning and local path planning. In global path planning problem the environment will be familiar with the robot while in local path planning the robot is ignorant of the surrounding. For solving the optimization problem a socially inspired popular meta-heuristic optimization algorithm is used; which is known as Particle Swarm Optimization (PSO) technique. PSO was inspired by the social behaviour of birds and fish.

### **4.2. Mathematical representation of PSO**

As mentioned earlier, the Particle Swarm Optimization technique is inspired by the social behaviour of birds and fishes. In the case of bird flocking, when birds move in a group then each bird in the flock is having its own position and velocity. This is nothing but the personal best position of each bird. The best position in the flock is gauged based on an input parameter known as objective function or fitness function. In case of bird flocking, the fitness function parameter is searching for food in less distance from the shelter for birds. So each bird is having some personal best position based on this fitness function. The position of the bird which has optimal fitness value is considered as the global best position. As the birds share information among them, all birds will tend to move towards the global best position.

As PSO is inspired by the social nature of bird flocking, each bird can be considered as one particle. So each particle has its own position and velocity. The position of the particle which satisfy optimum fitness value criterion is chosen as the global best position. Fig.1 below shows the structure of PSO.

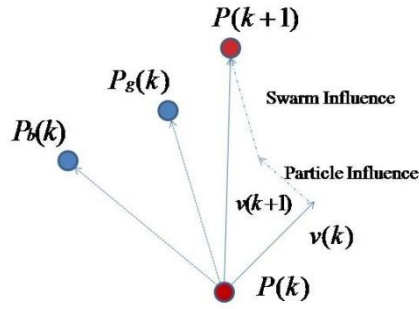


Fig.4.1 Structure of Particle Swarm Optimization

The mathematical equation which describes PSO is given below:

$$V_i(k+1) = w * V_i(k) + c_1 * r_1 (Y_{Pbest} - Y_i) + c_2 * r_2 (Y_{Gbest} - Y_i) \quad (4.1)$$

$$y_i(k+1) = y_i(k) + V_i(k+1) \quad (4.2)$$

Where  $i$  = Particle Number.

$k$  = Iteration counter.

$w$  = Inertial weight (decreasing function).

$c_1$  and  $c_2$  = Acceleration coefficients known as cognitive and social parameters.

$r_1$  and  $r_2$  = Random values in  $[0,1]$ .

$Y_{Pbest}$  = Best position of particle.

$Y_{Gbest}$  = Global best position of particle.

### 4.3. Mobile Robot Architecture

Since the robot path planning is an optimization problem, So PSO can be implemented for solving robot path planning problem.

There are certain assumptions considered in the architecture such as:

1. The robots are represented as some points in the search space. Let the search space is represented by  $Z$  and 'N' represents the number of particles.
2. The starting and ending points are also mentioned by different points.
3. The initial position of points is the personal best positions of robots and after that all the points tend to move towards a single point which is nothing but the global best position.
4. The whole process continues until robots reach the destination.

First of all, the mobile robot move towards the destination point until it senses any obstacle in front of it. If there is no obstacle in front of it, then it reaches to the destination. But when the robot senses any obstacle, it generates some random particles within its sensing range. Later the robot calculates the fitness of each particle to find out the global best position. Thus

further movement of the robot is decided towards the global best position. If suppose the global best position is the destination then it stops otherwise the process continues iteratively. For the accomplishment of a whole task fitness function plays the crucial role because based on this the positions of personal and global best position is determined.

#### 4.4. Fitness Function Consideration

The fitness function is nothing but the set of boundary conditions or constraints to fulfil the control objective. In this paper a new fitness function is generated to get the desired result. The validity of the fitness function is verified by simulation discussed later in this paper. The fitness function can be chosen like this

1. The distance between the robot (particle) and destination should be as small as possible. Hence the fitness function should be directly proportional to distance between the robot and destination.

$$F \propto \Gamma_{R-GOAL} \quad (4.3)$$

Where F = fitness function.

$\Gamma_{R-GOAL}$  = distance between the robot and goal position.

2. The distance between the robot and obstacle should as large as desired. So the fitness function is inversely proportional to distance between the robot and obstacles.

$$F \propto 1/\Gamma_{R-OBSTACLE} \quad (4.4)$$

Where  $\Gamma_{R-OBSTACLE}$  = distance between the robot and obstacles.

3. Similarly the distance between robots within its sensing range is large. Hence

$$F \propto 1/\Gamma_{R-R} \quad (4.5)$$

Where  $\Gamma_{R-R}$  = distance between robot and robot which is a time varying function.

Now combining all the conditions the complete fitness function can be written like this

$$F \propto \frac{\Gamma_{R-GOAL}}{\Gamma_{R-OBSTACLE} * \Gamma_{R-R}} \quad (4.6)$$

Now

$$\Gamma_{R-GOAL} = \sqrt{(R_{x_i} - goal_x)^2 + (R_{y_i} - goal_y)^2} \quad (4.7)$$

Where  $R_{x_i}$ ,  $R_{y_i}$  are the robots (particles) positions and  $goal_x$ ,  $goal_y$  is goal position.

$$\Gamma_{R-OBSTACLE} = \sqrt{(R_{x_i} - O_{x_i})^2 + (R_{y_i} - O_{y_i})^2} \quad (4.8)$$

Where  $O_{x_i}$ ,  $O_{y_i}$  are the obstacles in the search space.

$$\Gamma_{R-R} = \mu_1 * t * V_{R-R} + \mu_2 * \theta \quad (4.9)$$

Where  $t$  = time in seconds.

$V_{R-R}$  = relative velocity between two robots. For simplicity velocity of all the robots is taken same that is uniform but it can be taken different for different robot.

$\theta$  = angle of heading of robots.

From equation (6) we can write

$$F = \frac{\psi * \Gamma_{R-GOAL}}{\Gamma_{R-OBSTACLE} * \Gamma_{R-R}} \quad (4.10)$$

By tuning the value of  $\psi$ , different possible optimal paths for the mobile robot can be formed.

The whole process can be summarized like

1. Define robot and the goal position. Define maximum number of iterations.
2. Move the robot from its source to destination by incrementing the value of horizontal and vertical component of the position vector.
3. If there is no obstacle the robot will move to destination in the direction of slope between source and destination because the slope is the shortest path between any two points.
4. If the robot senses any obstacle in front of it generates random particles and based on minimum fitness value criteria the global best position is found.
5. The robot will move to these global best positions until it reaches the destination. When it will reach the destination the iteration stops.

The creation of random particles using MATLAB is shown below in the figure.

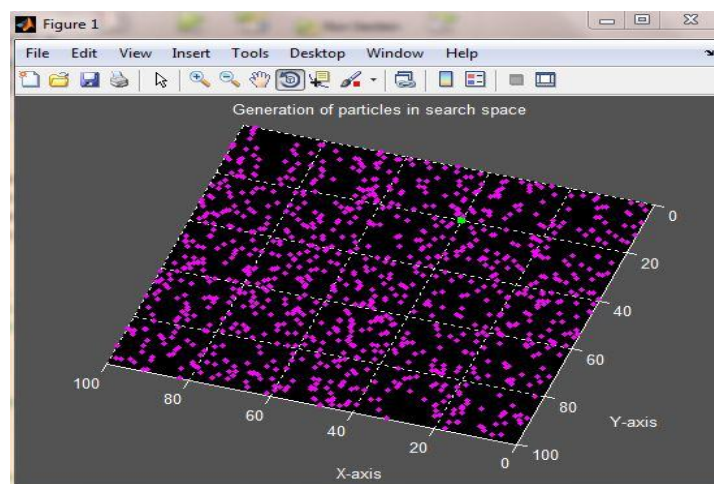


Fig.4.2 Random particles with one global best position

In the above figure the points in magenta colour represent the positions of different particles and the point in green colour represents the global best position of all the particles. The termination criteria for the algorithm are that it should stop when the robots reach to the destination or maximum iteration is reached.

The flowchart for the whole process can be illustrated like this as shown below:

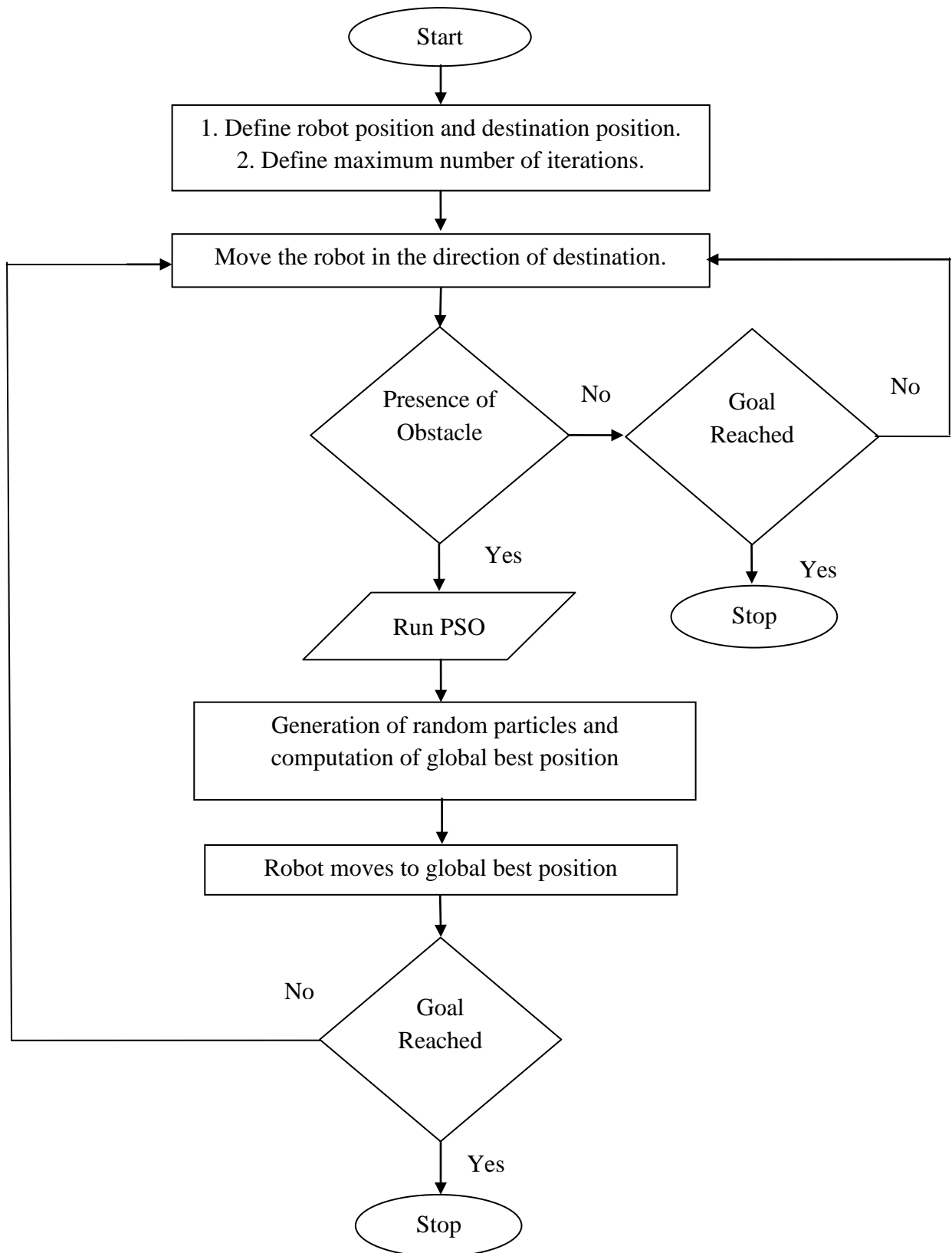


Fig.4.3 Flowchart of the Algorithm

The 3D view of particles in search space can be created and represented in MATLAB shown below in the figure 4.4.

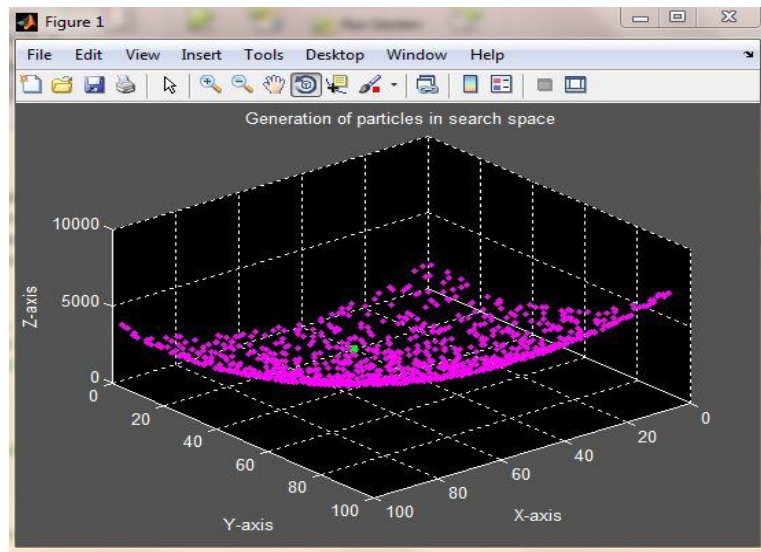


Fig.4.4 3D view of particles in search space

#### 4.5. Summary

Particle Swarm Optimization (PSO) structure and its implication in robot path planning is discussed. The main advantage of using PSO in robot path planning is its faster convergence and smoothness. The fitness function is generated as per the requirement and minimized using PSO.

## 5. EXPERIMENTAL AND SIMULATION RESULTS

### 5.1 Overview

This chapter describes the simulation analysis of the proposed PSO algorithm in robot path planning and real time implementation of the same for multiple robots. The simulation analysis is carried out using MATLAB 2013b software. The hardware design is done using Arduino microcontroller with on-board ultrasonic sensor module and actuators.

### 5.2 Simulation Analysis

The simulation analysis of goal seeking behaviour of multiple mobile robots with obstacle avoidance is shown below:

For generating different collision free trajectories the tuning parameter  $\psi$  is tuned properly. The simulation is performed by choosing  $c_1=1$  and  $c_2=1$ . However, different values for  $c_1$  and  $c_2$  can be chosen. The value of  $r_1$  and  $r_2$  is chosen as one for simplicity. After tuning the parameter, it is found that the parameter at  $\psi =0.36$  gives better results. The simulation result validating the above statements shown below:

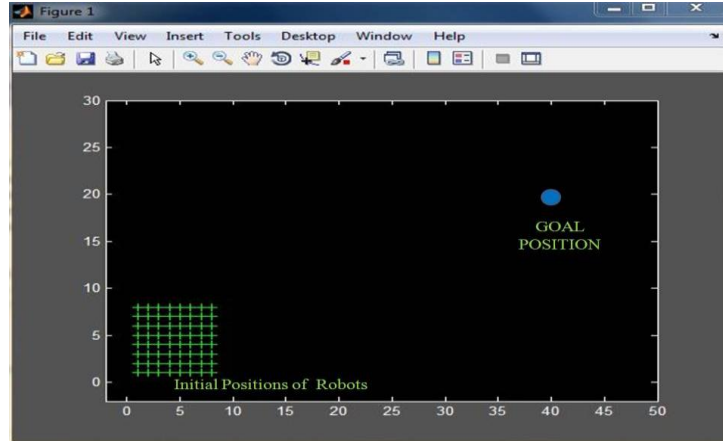


Fig.5.1 Robots in absence of obstacles

The robots start in a swarm from their starting position and approaching the destination position. At first the movement of robots is simulated without any obstacle. As per the PSO algorithm the robots moved to their intermediate goal positions before reaching the common goal position. The environment considered here 50 x 30 rectangular space. The approach of all the robots towards the final goal position is shown in the figure 5.2 below



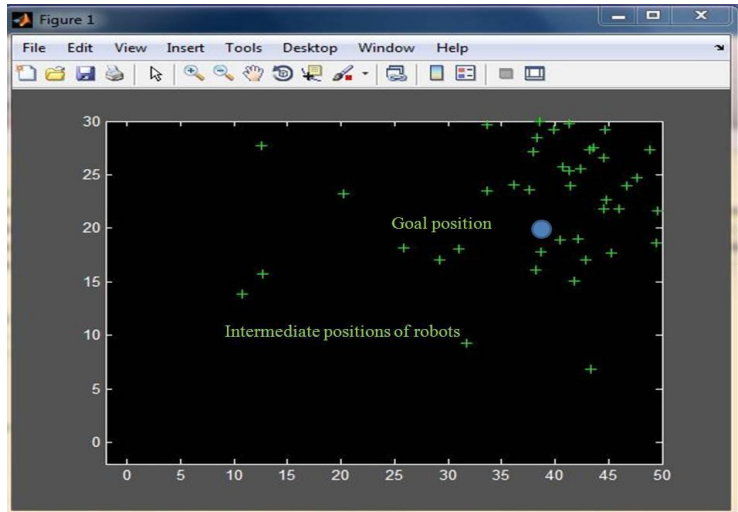


Fig.5.2 Robots approaching goal in the absence of obstacles

The convergence of robots to goal position is shown in the figure 5.3 below. Most of the robot reached to goal position. Few robots may not reach the destination due to reasons like noise in the environment, power drainage, improper tuning of the parameters of the PSO.

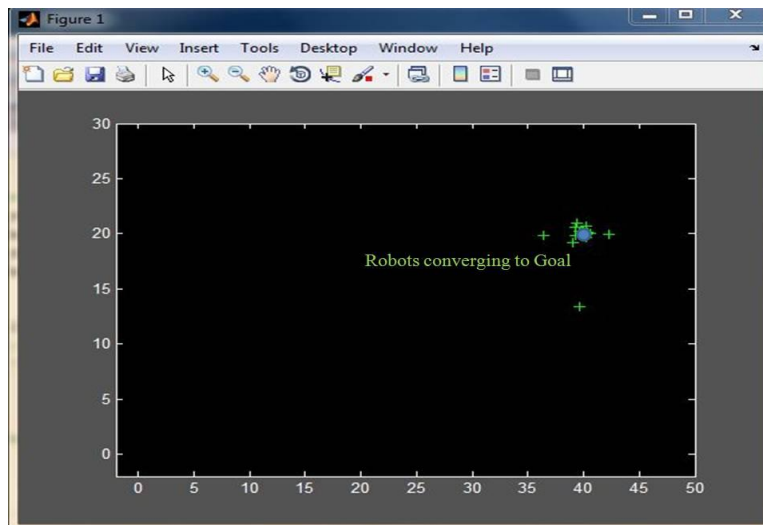


Fig.5.3 Robots reached goal in the absence of obstacles

Now, the important feature of motion planning of mobile robots using PSO is tested in the simulation environment. The simulation of mobile robot in presence of static obstacles is shown below. However the method holds good for mobile (dynamic) obstacles.

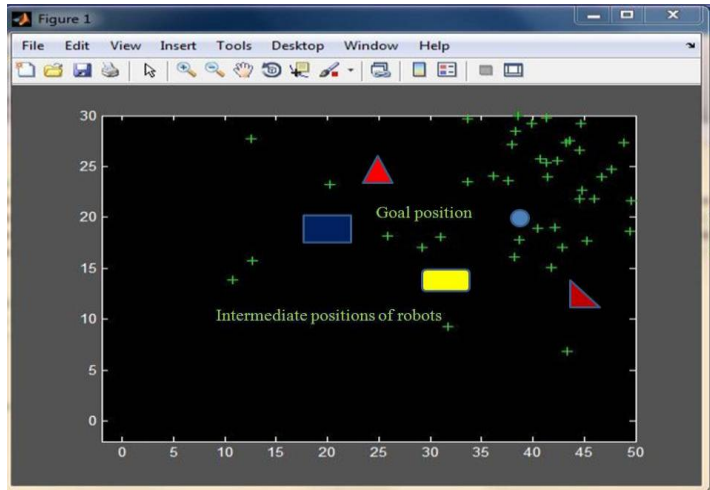


Fig.5.4 Robots moving towards goal in presence of obstacles.

The convergence of mobile Robots towards the goal in presence of obstacles is shown below in the figure 5.5.

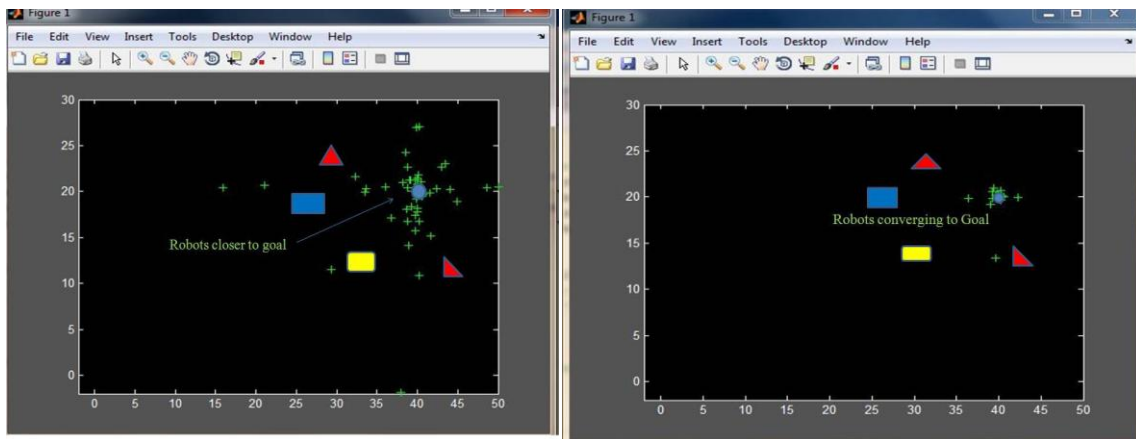


Fig.5.5 Robots approaching goal in presence of obstacles.

The obstacle avoidance of mobile robot is shown in the simulation diagram. The simulation is performed for both static and dynamic obstacles.

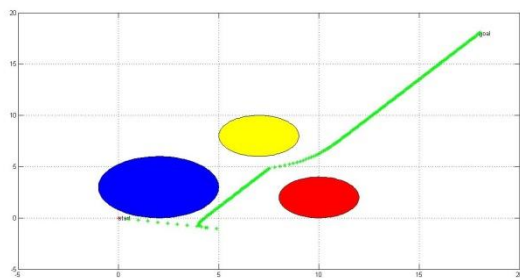


Fig 5.6 (a) avoidance of static obstacle

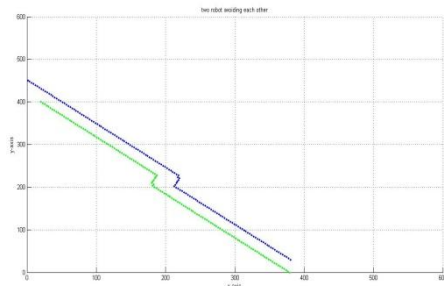


Fig.5.6(b) avoidance of dynamic obstacle

The robot is represented as a point in the diagram and the obstacles are of circular static obstacles.

### 5.3 Experimental Analysis

The experimental analysis is performed in real time environment with three mobile robots. For better sensing of obstacles the ultrasonic sensor is mounted on a 0-180 degree rotatable servo motor. The robots are capable of detecting both static and dynamic obstacles. Static obstacles are non-mobile obstacles while dynamic obstacles are movable. There are some pictures of static obstacle avoidance shown below in the figures.



Fig. 5.7 Robots avoiding static obstacles.

The pseudo code for obstacle avoidance algorithm can be illustrated as follows:

- Assign the threshold value for obstacle detection for the ultrasonic sensor. Let say threshold = 40 centimeters.
- Make the servomotor at 90 degree angle. So that the sensor will face front. Now get the front obstacle distance value from the ping sensor.

```
Myservo.write(90);
```

```
Int distance = pingsensor ( ) ;
```

- Compare distance value with threshold value.

```
If (distance > threshold )
```

```
Moveforward(); delay (1000);
```

```
Else
```

```
Allstop(); delay(1000);
```

- If distance value is less than the threshold then make robot to stop that is make the motors off. Now make the servo to move to right.

```
Myservo.write(20);
```

```
Rightdistance= pingsensor(); delay(1000);
```

- Move the servo to left by an angle of 160 degrees to check the left distance.

```
Myservo.write(160);
```

```
Leftdistance = pingsensor(); delay(1000);
```

- Now compare the leftdistance and rightdistance value; wherever the distance is more move the robot to that direction.

```
If (Leftdistance > Rightdistance && Leftdistance > threshold )
```

```
Moveleft (); delay(1000);
```

```
Else if (Leftdistance < Rightdistance && Rightdistance > threshold)
```

```
Moveright(); delay(1000);
```

```
Else
```

```
Allstop();
```

```
Myservo.write (90);
```

The detection of dynamic or mobile obstacles is major challenge compared to static obstacles. Because the position of the object changes with time in case of dynamic environment. In case of multi-robot cooperation one robot is dynamic obstacle for another robot. So better calibration of ultrasonic sensor is necessary. The pictures showing avoidance of dynamic obstacles is given below:

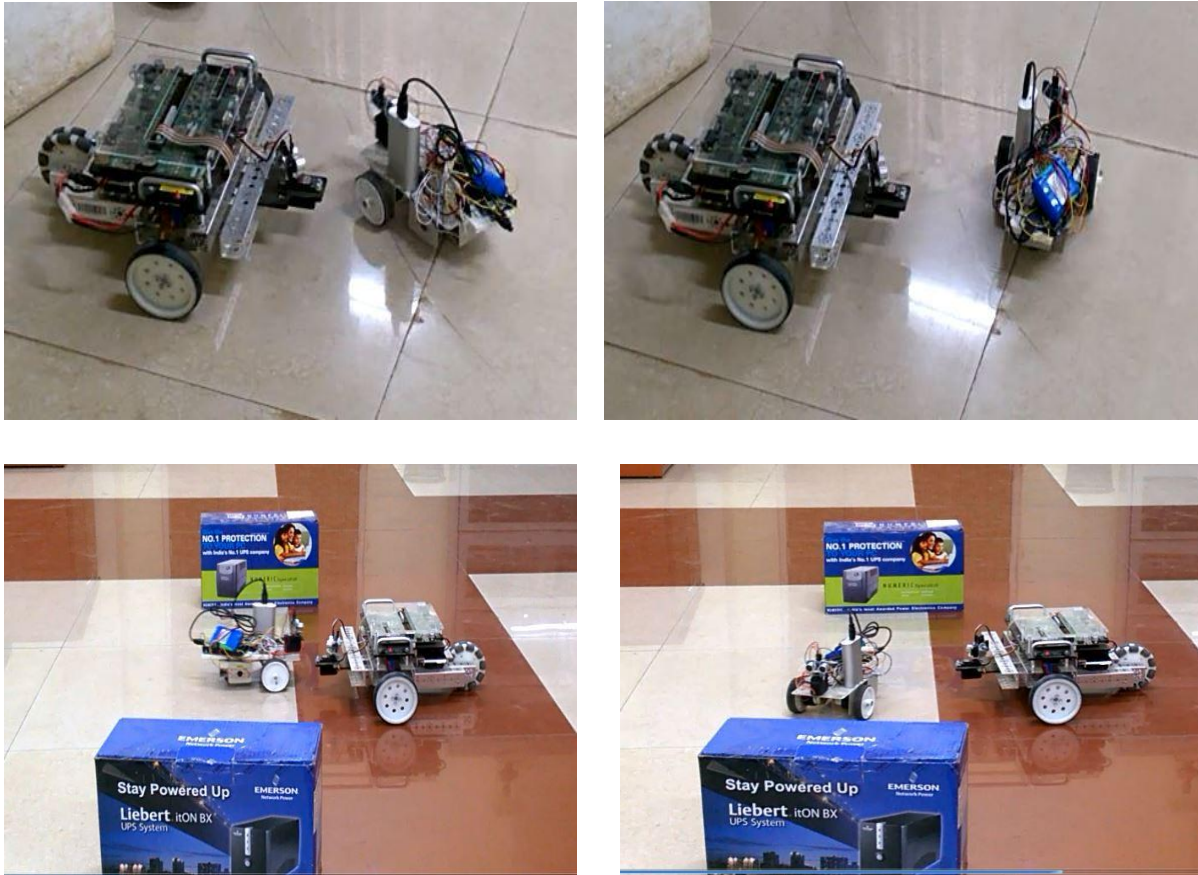


Fig.5. 8. Robots avoiding static obstacles and each other

The above diagrams represent employment of two mobile robots in real time environment. The obstacle avoidance behaviour of swarm of three mobile robots is show below in the figure. All the mobile robots are non-holonomic and differentially driven. The robots are tested in dynamic environment where both mobile and static obstacles were present. For accurate coordination among the robots proper integration of ultrasonic sensor, servo motor and geared wheels are done. Synchronization of all these sensors and actuators are the key for obstacle avoidance in real environment. The servomotor has to rotate complete 180 degrees for complete view of objects while at the same time the ultrasonic sensor has to give the correct reading of position of obstacle to the microcontroller. Now the controller will decide the direction in which the bot has to move based on ultrasonic sensor information.



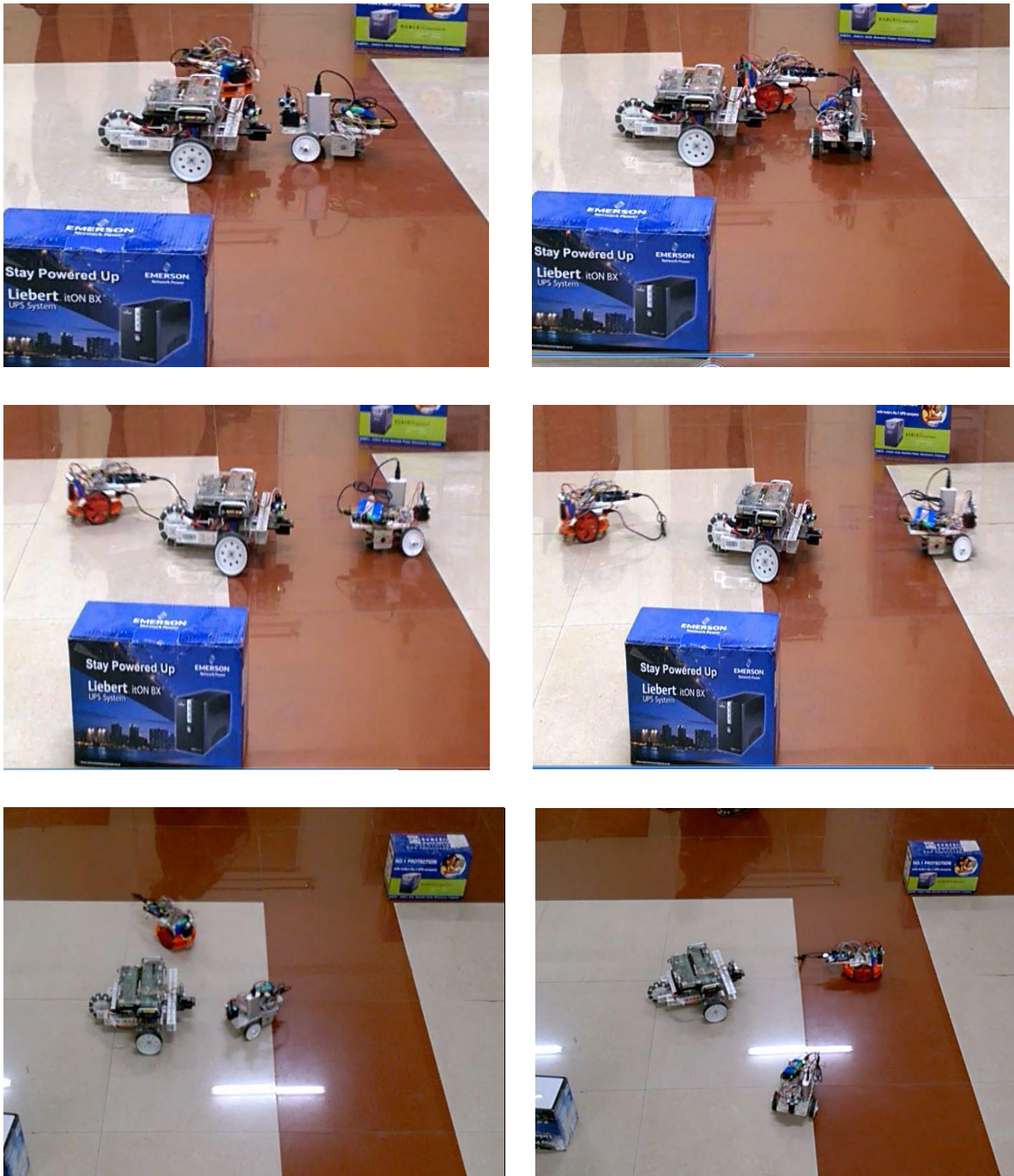


Fig. 5.9 Robots avoiding static and mobile obstacles

Three mobile robots avoiding each other and static obstacles are shown in the figure.

#### 5.4 Summary

Simulation of motion planning of multiple robots is performed in MATLAB. The obstacle avoidance issue of mobile robots is given the priority. Experimental analysis for three different robots is performed.

## **6. CONCLUSION AND FUTURE WORK**

### **6.1 Conclusion**

Multirobot cooperation is one of the flamboyant topics in the current age of robotics. The research dissertation mainly focuses on the autonomous navigation of the mobile robots. The excellent part of the project is that obstacle avoidance is performed using low cost Ultrasonic sensors and there is no use of advanced sensors like LASER, Stereo camera etc. The major problems are like smooth orientations, goal seeking of the robots etc. There are some design issues also like improper load distribution, improper differential orientation etc. The robots are tested in real time environment with static and dynamic obstacles. For complete 180 degree view of the robot in the forward movement direction the ultrasonic sensor is mounted on the servo motor. Proper control of the angle of the servo motor is necessary to move the ultrasonic sensor in the desired direction. At the same time accurate calibration of the ultrasonic sensor should be done to get the exact distance of the obstacle.

### **6.1 Future Scope**

Obstacle avoidance and target search is a nice field and it finds application in target searching, search and rescue, disaster control etc. As it is mentioned in the conclusion section that there are still some problems with the mobile robots like proper orientation and goal seeking behaviour. Some of the approaches to sort out these challenges are

- Use of position encoder for smooth and proper orientation of the robot.
- For goal seeking issue, GPS can be used to find the current coordinates and assigning the desired goal position coordinates. Proper controlling of the heading angle and orientation can be done using magnetometer (compass).
- Design issues can be addressed by suitable design of chassis, proper positioning of motors, distance sensor (ping or ultrasonic) and other sensors.

## REFERENCES

- [1] Pugh, J., & Martinoli, A. (2007, April). Inspiring and modelling multi-robot search with particle swarm optimization. In *Swarm Intelligence Symposium, 2007. SIS 2007. IEEE* (pp. 332-339). IEEE.
- [2] Zou, R., Kalivarapu, V., Winer, E., Oliver, J., & Bhattacharya, S. (2015). Particle Swarm Optimization Based Source Seeking. *arXiv preprint arXiv:1501.06622*.
- [3] Ali, H. E. I., Habashy, S. M., Saad, E. M., & Yousef, M. T. A Comparison STUDY BETWEEN USING THE PSO.
- [4] Cai, Y., & Yang, S. X. (2014, August). An improved PSO-based approach with dynamic parameter tuning for cooperative target searching of multi-robots. In *World Automation Congress (WAC), 2014* (pp. 616-621). IEEE.
- [5] Cai, Y., & Yang, S. X. (2014, June). A potential field-based PSO approach for cooperative target searching of multi-robots. In *Intelligent Control and Automation (WCICA), 2014 11th World Congress on* (pp. 1029-1034). IEEE.
- [6] Tang, Q., & Eberhard, P. (2011). Cooperative motion of swarm mobile robots based on particle swarm optimization and multibody system dynamics. *Mechanics based design of structures and machines*, 39(2), 179-193.
- [7] Smith, L. L., Venayagamoorthy, G. K., & Holloway, P. G. (2006). Obstacle avoidance in collective robotic search using particle swarm optimization. In *Swarm Intelligence Symposium*. Institute of Electrical and Electronics Engineers IEEE.
- [8] Tuncer, A., Yildirim, M., & Erkan, K. (2012). A motion planning system for mobile robots. *Adv Electr Comput Eng*, 20(1), 57-62.
- [9] Mohamed, A. Z., Lee, S. H., Hsu, H. Y., & Nath, N. (2012). A faster path planner using accelerated particle swarm optimization. *Artificial Life and Robotics*, 17(2), 233-240.
- [10] Shiltagh, N. A., & Jalal, L. D. (2013). Path Planning of Intelligent Mobile Robot Using Modified Genetic Algorithm. *International Journal of Soft Computing and Engineering (IJSCE) ISSN*, 2231-2307.
- [11] Deepak, B. B. V. L., & Parhi, D. R. (2012). PSO based path planner of an autonomous mobile robot. *Central European Journal of Computer Science*, 2(2), 152-168.
- [12] Pessin, G., Osório, F. S., & Wolf, D. F. (2010). Particle swarm optimization applied to intelligent vehicles squad coordination. *Intelligent Autonomous Vehicles*, 7(1), 401-406.
- [13] Panigrahi, P. K., Ghosh, S., & Parhi, D. R. Comparison of GSA, SA and PSO Based Intelligent Controllers for Path Planning of Mobile Robot in Unknown Environment.



- [14] Shiltagh, N. A., & Jalal, L. D. (2013). Optimal path planning for intelligent mobile robot navigation using modified particle swarm optimization. *International Journal of Engineering and Advanced Technology (IJEAT)*, 2(4), 260-267.
- [15] Deepak, B. B. V. L., Parhi, D. R., & Raju, B. M. V. A. (2014). Advance Particle Swarm Optimization-Based Navigational Controller for Mobile Robot. *Arabian Journal for Science and Engineering*, 39(8), 6477-6487.
- [16] Derr, K., & Manic, M. (2009, May). Multi-robot, multi-target particle swarm optimization search in noisy wireless environments. In *Human System Interactions, 2009. HSI'09. 2nd Conference on* (pp. 81-86). IEEE.
- [17] Ersson, T., & Hu, X. (2001). Path planning and navigation of mobile robots in unknown environments. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on* (Vol. 2, pp. 858-864). IEEE.
- [18] Ahmadzadeh, S., & Ghanavati, M. (2012). Navigation of mobile robot using the (PSO) particle swarm optimization. *Journal of Academic and Applied Studies (JAAS)*, 2(1), 32-38.
- [19] Yarmohamadi, M., Javadi, H. H. S., & Erfani, H. (2011). Improvement of robot path planning using particle swarm optimization in dynamic environments with mobile obstacles and target. *Advanced Studies in Biology*, 3(1), 43-53.
- [20] Masehian, E., & Sedighzadeh, D. (2010). Multi-objective PSO-and NPSO-based algorithms for robot path planning. *Advances in electrical and computer engineering*, 10(4), 69-76.
- [21] Jayaraj, D. (2012). Path planning in swarm robots using particle swarm optimisation on potential fields.
- [22] Saska, M., Macas, M., Preucil, L., & Lhotska, L. (2006, September). Robot path planning using particle swarm optimization of Ferguson splines. In *Emerging Technologies and Factory Automation, 2006. ETFA'06. IEEE Conference on* (pp.833-839). IEEE.
- [23] Savkin, A. V., & Wang, C. (2014). Seeking a path through the crowd: Robot navigation in unknown dynamic environments with moving obstacles based on an integrated environment representation. *Robotics and Autonomous Systems*, 62(10), 1568-1580.
- [24] Susnea, I., Minzu, V., & Vasiliu, G. (2009, December). Simple, real-time obstacle avoidance algorithm for mobile robots. In *8th WSEAS International Conference on Computational intelligence, man-machine systems and cybernetics (CIMMACS'09)* (pp. 24-29).

- [25] Čosić, A., Šušić, M., & Katić, D. (2012). Advanced algorithms for mobile robot motion planning and tracking in structured static environments using particle swarm optimization. *Serbian Journal of Electrical Engineering*, 9(1), 9-28.
- [26] <http://www.prsrobots.com/puma500.html>
- [27] [www.electrodimeron.com](http://www.electrodimeron.com)
- [28] [www.space.com](http://www.space.com)
- [29] <http://www.circuitdb.com/wp-content/uploads/2014/09/hr-sr04-timing.jpg>