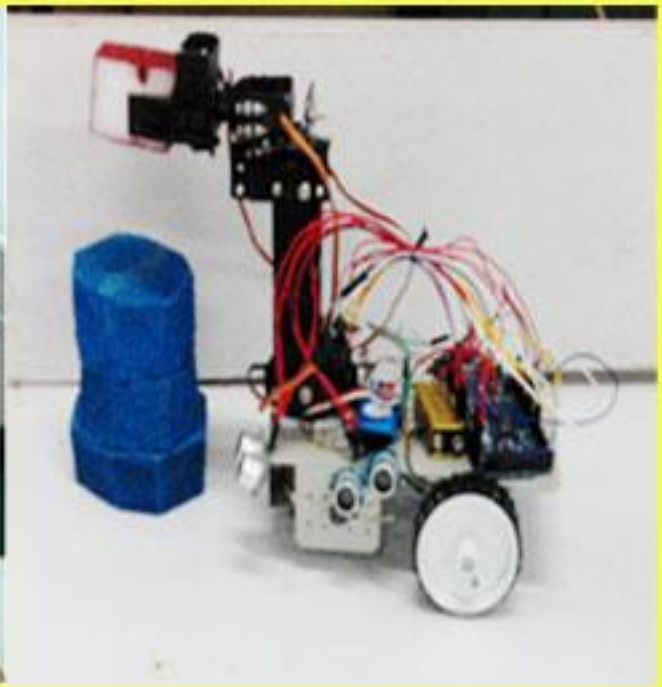
Ph.D. Thesis

Design and Development of an Automated Mobile Manipulator for Industrial Applications



B B V L DEEPAK

DESIGN AND DEVELOPMENT OF AN AUTOMATED MOBILE MANIPULATOR FOR INDUSTRIAL APPLICATIONS

A thesis submitted in fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Mechanical Engineering

by

B B V L DEEPAK

Roll No: 510ME608

Under the supervision of

Prof. Dayal R. Parhi

DEPARTMENT OF MECHANICAL ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY ROURKELA

MARCH, 2015

Declaration

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

Date: B B V L Deepak

N.I.T. Rourkela

Department of Mechanical Engineering

NATIONAL INSTITUTE OF TECHNOLOGY, ROURKELA

ORISSA, INDIA – 769 008

CERTIFICATE

This is to certify that the thesis entitled "***Design and Development of an Automated Mobile Manipulator for Industrial Applications***", being submitted by **B B V L Deepak**, Roll No. **510ME608,** to the National Institute of Technology, Rourkela for the award of the degree of ***Doctor of Philosophy*** in Mechanical Engineering, is a bona fide record of research work carried out by him under my supervision and guidance.

The candidate has fulfilled all the prescribed requirements.

The thesis, which is based on candidate's own work, has not been submitted elsewhere for the award of a degree.

In my opinion, the thesis is of the standard required for the award of Doctor of Philosophy in Mechanical Engineering.

To the best of my knowledge, he bears a good moral character and decent behaviour.

Supervisor

Dr. Dayal R. Parhi,

Professor, Department of Mechanical Engineering

NATIONAL INSTITUTE OF TECHNOLOGY

Rourkela-769 008 (INDIA)

Acknowledgement

This report is a result of my efforts as a research scholar towards my Ph. D. in Robotics Laboratory, Department of Mechanical Engineering, National Institute of Technology Rourkela. During this time, I have been supported by various people to whom I wish to express my gratitude.

I am thankful to Prof. Sunil Kumar Sarangi, (Director, NIT-Rourkela) for giving me an opportunity to work under the supervision of Prof. D. R. Parhi. I am indebted to Prof. D. R. Parhi. He offered me the possibility to work in a liberal environment and given me the freedom to carry out my research in an independent way. The charming personality of Prof. Parhi has been unified perfectly with knowledge that creates a permanent impression in my mind. His receptiveness to new and different ideas and his willingness to leave his space and time were always important sources of inspiration and motivation.

I would like to thank to all DSC members and other faculty members of the institute for their co-operation and help. I am very much thankful to Prof. S. S. Mohapatra, HOD, Department of Mechanical Engineering for his continuous encouragement. Also, I am indebted to him for providing me all official and laboratory facilities.

In addition, I take this opportunity to express my regards and obligation to my family members for encouraging me in all aspects for carrying out the research work.

Last but not the least, let me thank all of my friends for their co-operation during my stay at Robotics Lab. Also, I am thankful to all the teaching & non-teaching staffs of Mechanical Engineering and Industrial Design Departments for their kind cooperation.

Finally, I am deeply grateful for the support of many kinds I got from my colleagues and friends who are at my side during the research work.

B B V L Deepak

Abstract

This thesis presents the modeling, control and coordination of an automated mobile manipulator. A mobile manipulator in this investigation consists of a robotic manipulator and a mobile platform resulting in a hybrid mechanism that includes a mobile platform for locomotion and a manipulator arm for manipulation. The structural complexity of a mobile manipulator is the main challenging issue because it includes several problems like adapting a manipulator and a redundancy mobile platform at non-holonomic constraints. The objective of the thesis is to fabricate an automated mobile manipulator and develop control algorithms that effectively coordinate the arm manipulation and mobility of mobile platform.

The research work starts with deriving the motion equations of mobile manipulators. The derivation introduced here makes use of motion equations of robot manipulators and mobile platforms separately, and then integrated them as one entity. The kinematic analysis is performed in two ways namely forward & inverse kinematics. The motion analysis is performed for various WMPs such as, Omnidirectional WMP, Differential three WMP, Three wheeled omni-steer WMP, Tricycle WMP and Two steer WMP. From the obtained motion analysis results, Differential three WMP is chosen as the mobile platform for the developed mobile manipulator. Later motion analysis is carried out for 4-axis articulated arm. Danvit-Hartenberg representation is implemented to perform forward kinematic analysis. Because of this representation, one can easily understand the kinematic equation for a robotic arm. From the obtained arm equation, Inverse kinematic model for the 4-axis robotic manipulator is developed.

Motion planning of an intelligent mobile robot is one of the most vital issues in the field of robotics, which includes the generation of optimal collision free trajectories within its work space and finally reaches its target position. For solving this problem, two evolutionary algorithms namely Particle Swarm Optimization (PSO) and Artificial Immune System (AIS) are introduced to move the mobile platform in intelligent manner. The developed algorithms are effective in avoiding obstacles, trap situations and generating optimal paths within its unknown environments. Once the robot reaches its goal (within the work space of the manipulator), the manipulator will generate its trajectories according to task assigned by the user.

Simulation analyses are performed using MATLAB-2010 in order to validate the feasibility of the developed methodologies in various unknown environments. Additionally, experiments are carried out on an automated mobile manipulator. ATmega16 Microcontrollers are used to enable the entire robot system movement in desired trajectories by means of robot interface application program. The control program is developed in robot software (Keil) to control the mobile manipulator servomotors via a serial connection through a personal computer. To support the proposed control algorithms both simulation and experimental results are presented. Moreover, validation of the developed methodologies has been made with the ER-400 mobile platform.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

ACO	Ant Colony Optimization
AIS	Artificial immune system
AIMP	Adaptive Immune based Motion Planner
ANN	Artificial Neural Network
B	Local frame
${}^{B}r_{p}$	Point P representation in local frame B
BFOA	Bacterial Foraging Optimization Algorithm
$C_1(\beta)$	Sliding constraints of all standard wheels
C_1 & C_2	Cognitive and social parameters
DOF	Degrees of Freedom
D-H	Denavit-Hartenberg
F_i	i^{th} particle Fitness function
FLC	Fuzzy Logic Control
FMF	Fuzzy logic Membership Functions
G	Global reference frame
GA	Genetic Algorithm
${}^{G}r_{p}$	Point P representation in global frame G
${}^{G}T_{B}$	Homogeneous transformation matrix between frames B and G
IIMP	Innate Immune based Motion Planner
J(β)	Projections of all wheels motions along their individual wheel planes
$[J_M]$	Velocity Jacobean of manipulator
$[J_{MP}]$	Velocity Jacobean of Mobile Platform
$[J_{WMP}]$	Velocity Jacobean of Mobile Manipulator
K	Iteration counter.
K_1	Proportionality/controlling parameter
MRC	Mobile Robot Controller
NiF-T	Neural integrated Fuzzy controller
N_f	Number of fixed wheels equipped to mobile platform
N_s	Number of Steerable standard wheels equipped to mobile platform
N_{sw}	Number of Swedish wheels equipped to mobile platform

N_{sp}	Number of spherical wheels equipped to mobile platform
N_c	Number of Castor wheels equipped to mobile platform
Nob	Nearest obstacle
ORNN	Output-Refinement Neural Network
PSO	Particle Swarm Optimization
PFIN	Potential filed immune network
R	Rotational mapping matrix
$R(\psi)$	Orthogonal rotation matrix
RL	Reinforcement Learning
RNN	Rule Neural Network
Sob	Certain number of obstacles
T_i	Homogeneous transformation matrix of i^{th} coordinate system
TSP	Travelling Salesman Problem
WMM	wheeled mobile manipulator
W_1& W_2	First fitness function controlling parameter
$\{\vec{X}\}$	Mobile platform position representation
$\{\dot{\vec{X}}_m\}$	Mobile platform motion representation
X_{pbest}	Best position decided by each particle, called as position best
X_{gbest}	Global best position
a_i	translation along the X-axis
a_t	Adaptive score
ag	Antigen
ab	Antibody
ag_d	Dominant antigen
d_i	translation along the Z-axis
$dist_{p_iT}$	Distance between i^{th} particle and target position
$dist_{p_iNOb}$	Distance between i^{th} particle and nearest obstacle
$dist_{ROb}$	Distance between the robot and the sensed obstacle
k	weighing parameter of adaptive score
rand	Random variables
$(robotx, roboty)$	Point represented of robot
i_X, j_Y, k_Z	Unit vectors along the coordinate axes OXYZ

i_x, j_y, k_z	Unit vectors along the coordinate axes oxyz
p	position vector
$\{\dot{q}\}$	WMM parametric matrix
r	Wheel radius
$2s$	The wheel base
$v_i(k+1)$	Particle updated velocity
υ_{Rt}	Right wheel velocity
υ_{Lt}	Left wheel velocity
υ_b	maximum possible linear velocity of the robot
$x_i(k+1)$	Particle updated position
α_i	Rotation about the X-axis
$\alpha_{\wp}$	Allowance parameter
β	The angle of the wheel plane relative to the chassis
$\beta(t)$	The orientation of the wheel to chassis as a function of time
δ_s	The degree of Steerability
δ_M	Degree of manoeuvrability
γ	The exact angle between the roller axes and the main axis.
ω_{ag}	Antigenic weight
$\varphi(t)$	Rotational position of the wheel as a function of time t
ρ	Angular position of the wheel expressed in polar coordinates.
ψ	The angular difference between the global and local reference frames
Γ_r	Learning rate
$\dot{\theta}_{Rt}$	right wheel angular velocity
$\dot{\theta}_{Lt}$	Left wheel angular velocity
θ_i	Rotation about the Z-axis

Chapter 1

INTRODUCTION

Origin of the Work

Problem Statement

Objectives

Thesis Overview

1. INTRODUCTION

With an incessant need for increased productivity and the delivery of end products with uniform quality, manufacturing industries are turning more and more toward the computer-based automation. At the present time, industries are equipped with special-purpose machines to perform prearranged tasks in manufacturing process. But these machines are inflexible in performing variety of tasks and generally expensive. These limitations have led to the development of robots with capable of performing a variety of manufacturing functions in a more flexible working environment at lower production costs.

It is claimed that robots are used to perform 4A works for 4D, or 3H environments as represented in Fig.1.1. 4A performances are automation, augmentation, assistance and autonomous tasks. 4D environments are Dangerous. Dirty, Dull and Difficult. 3H Means Hot, Heavy and Hazardous environments.

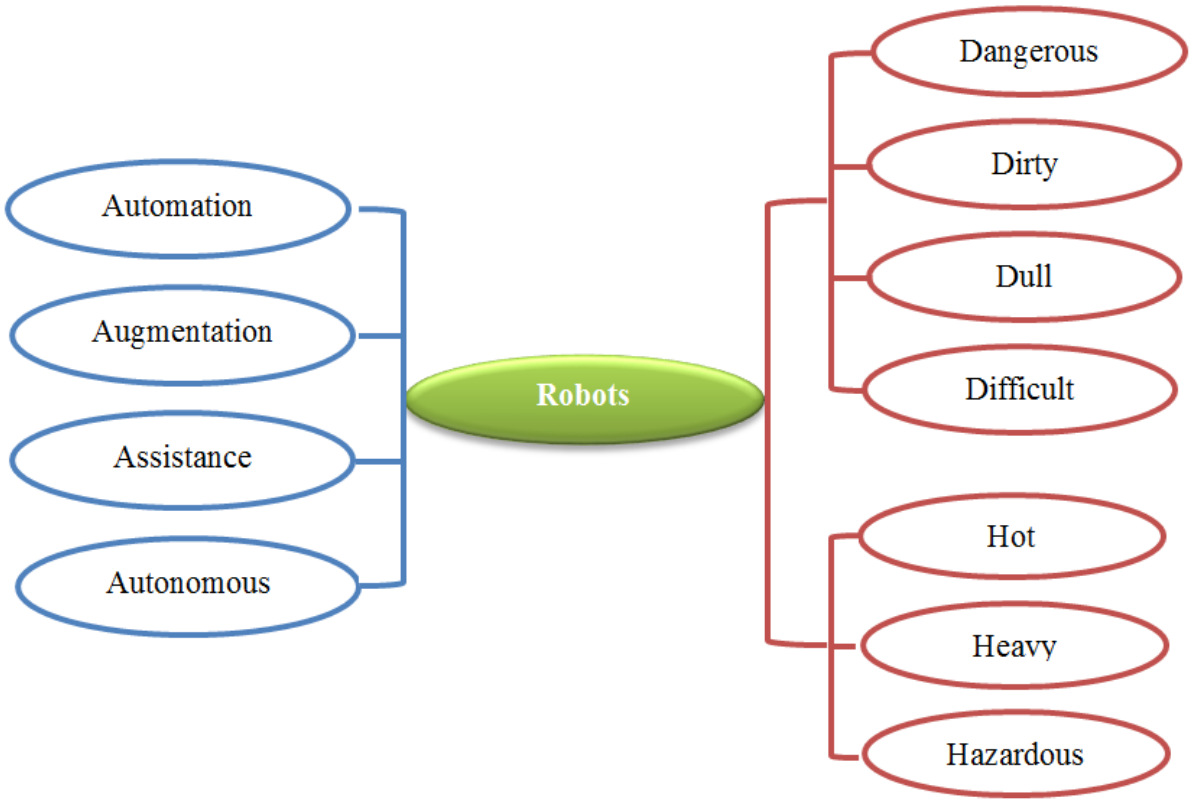


Fig.1.1 Robot applications for 4A in 3D & 3H

Robot is a multi-disciplinary engineering device that ranges in scope from the design of mechanical and electrical components to sensor technology, computer systems, and artificial intelligence as represented in Fig.1.2.

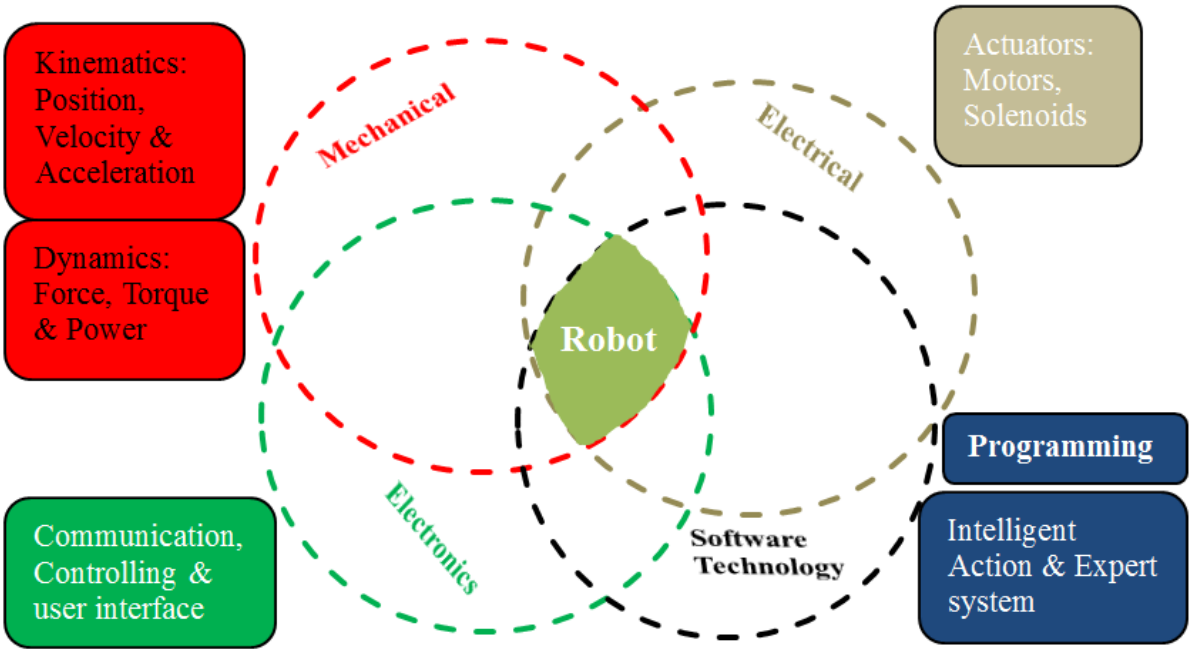


Fig.1.2 Multi-disciplinary nature of Robot

Mechanical domain deals with the design of mechanical components (links, end-effectors) with respect to its kinematics, dynamics and control analyses of robots. Electrical engineering works on powered actuators to give the motion at each joint of the robot. System design engineering deals with perception, sensing and control methods of robots. Programming/software engineering is responsible to make the robot as expert system by integrating intelligence to it.

1.1.Origin of the Work

An industrial robot is a general-purpose, computer-controlled manipulator consists of several rigid links connected by revolute or prismatic joints as represented in Fig.1.3. One end of the robot arm is attached to a fixed base, while the other end of the manipulator is an end-effector called tool to perform various tasks. The end-effector may be a gripper, welding tool, or spraying gun attached to the robot according to the desired tasks such as manipulate objects, welding, painting and assembly tasks etc. End-effector motion is decided according to the motion of the joints which results in relative motion of the links. Mechanically, an industrial robot is composed of an arm (or mainframe) and a wrist subassembly plus a tool. It is designed to reach the end-effector to a specific position located within its work volume.

Many commercially available industrial robots are widely used in manufacturing and assembly tasks, such as material handling, spot/arc welding, parts assembly, paint spraying,

loading and unloading numerically controlled machines, space and undersea exploration, prosthetic arm research, and in handling hazardous materials.

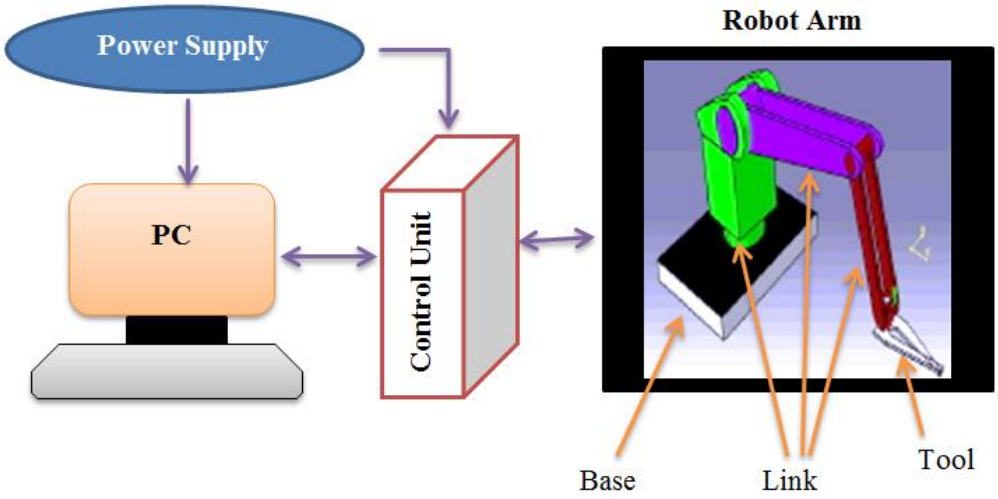


Fig.1.3 Industrial robot system architecture

Yet, for all of their successes, these industrial robots have a fundamental disadvantage: lack of mobility. A fixed manipulator has a limited work space that depends on where it is bolted down. In contrast, a mobile robot is able to travel all over the manufacturing plant and can apply its talents wherever it is most effective.

Over the past twenty years, bulk amount of research and development have concentrated on stationary robotic manipulators because of their industrial applications. Less effort has been made on mobile robots. Typically, a mobile robot's structure consists of a mobile platform that is driven with the aid of a pair of tracks, wheels/legs.

Path planning is one of the important tasks in the navigation of an autonomous mobile robot. The most important characterization of a motion planner is according to the problem it solves. It means that the effectiveness of the motion planner depends on properties of the robot solving the task. The different aspects under mobile robot motion control scheme are shown in Fig.1.4. These aspects of motion planning problem for autonomous mobile robot/robots includes:

1. Robot locomotive ability (most popular mechanism for robot motion)
2. Perception (acquiring robots environment condition)
3. Localization (determines the robot position in its environment)
4. Cognition (decision-making and execution to achieve its highest-order goals)
5. Motion Control (path execution in real world environment)

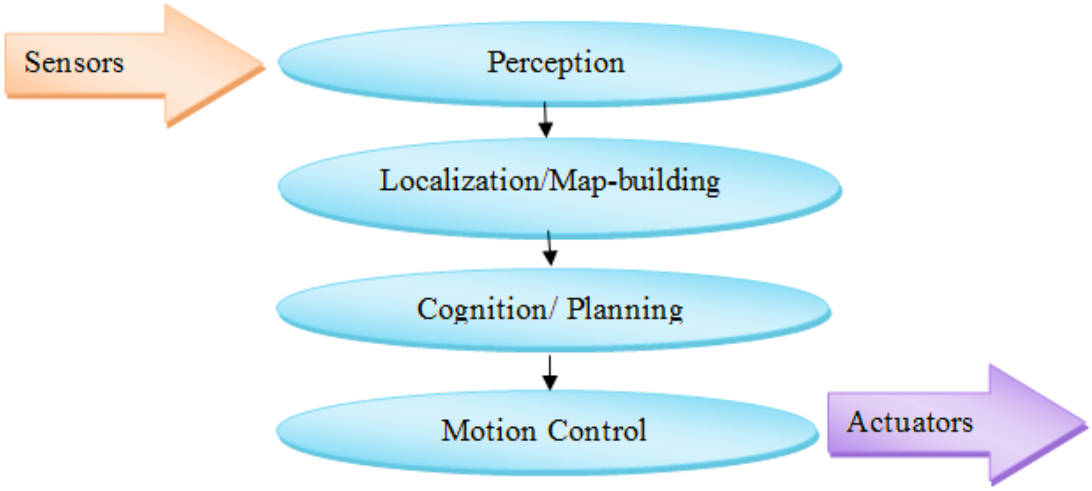


Fig.1.4 Mobile robot motion control scheme

There are varieties of possible ways to move, and so the selection of a robot's locomotion mechanism is an important aspect of mobile robot design. Most of these locomotion mechanisms are inspired by their biological counterparts such as walk, jump, run, slide, skate, swim, fly, and, of course roll. However, replicating nature in this regard is extremely difficult for several reasons. Although legged locomotion has been studied, the devastating majority of the mobile robots with wheel locomotion mechanism have been built and evaluated. In general, legged locomotion requires higher degrees of freedom and therefore greater mechanical complexity exists compared to wheeled locomotion. Wheels, in addition to being simple, are extremely well suited to flat ground and these are more energy efficient than legged/treaded robots on hard and smooth surfaces. Wheeled Mobile Robots (WMRs) find its widespread application in industry because of the hard, smooth plant floors in existing industrial environments.

1.2. Problem Statement

The new design is based on hybridization of locomotion and manipulation. In this approach, the mobile platform and manipulator arm are interchangeable in their roles so that both can support locomotion and manipulation in several modes of operation. In this system architecture, a manipulator is equipped on top of the mobile platform to provide the required manipulation capability (neutralization of bombs or landmines, manipulation of hazardous materials, etc). The development of the hybrid manipulator system covers mechanics of systems design, system dynamic modeling and simulations, design optimization, computer architecture and control system design.

1.3. Objectives

The aim of the research work is to increase significantly the robot's mobility and manipulability functionalities while increasing its reliability and reducing its complexity and cost. The objective of the current investigation is to develop a new paradigm for the design of hybrid manipulators in order to solve foremost existing problems and overcome barriers for of mobile platforms in unfamiliar terrain applications. The idea of the design paradigm is the interchangeability of the locomotion and manipulation functions, which benefits the robot's overall operation and function.

A mobile manipulator in this study is a manipulator mounted on a mobile platform. Since the motion of the arm is unknown a priori, the mobile platform has to use the joint position information of the manipulator for motion planning.

The following objectives were accordingly specified for this project work:

- Forward and inverse kinematic analyses are performed on the robotic arm and the mobile platform in order to navigate the robot and manipulate the objects in desired direction.
- Modeling of the arm is carried out in CAD software 'CATIA V6' in order to simulate each part of the manipulator.
- Intelligent planning and control algorithms are developed for the platform so that the manipulator is always positioned at the preferred configurations measured by its manipulability.
- To plan robot trajectory in terms of the kinematics of the manipulator, an intelligent algorithm is developed. The control scheme associates the mobile base activities and the manipulator activities to produce an intelligent hybrid system that performs a coordinated motion and manipulation.
- The control program is developed in robot software 'MATLAB & KIEL' to control the arm servo through a personal computer.
- Finally, experimental analysis is performed in order to validate the efficiency of the developed control strategies. Moreover, the developed methodologies are implemented to the fabricated hybrid robot and the procured ER-400 robot in various robotic environments.

1.4. Thesis Overview

This dissertation is organized as follows

- *Chapter 2* provides a background to the field of development of mobile manipulator. This includes the literature survey on kinematic models of the manipulator & mobile platform. It also introduces the existed types of design architectures to control the mobile platform.
- *Chapter 3* devotes to mechanism development and simulation analysis for the 4-axis robot arm. This study is carried out on detailed parts of robotic arm assembly. Preliminary computations will be done to build relationship among the robotic arm joints parameters (angles and lengths). These relations are used to control the position and orientation of the end effector(gripper) of the robotic arm. The trajectories generated by robotic arm in the workspace is obtained by performing numerical kinematic analysis (forward & inverse) in "MATLAB-2010. Simulation analysis of robotic arm is studied by performing mock-up in "CATIA V6".
- *Chapter 4* outlines the mechanical structure of the mobile platform to optimize the design by defining suitable operating parameters such as maneuverability and required motor torques.
- *Chapter 5* deals with the integration of the manipulation and locomotion functionalities to develop a new hybrid mobile manipulator.
- *Chapter 6* describes the control system design paradigms for the mobile platform. Two system architectures are developed those are based on Particle swarm optimization (PSO). The developed algorithms generate optimal trajectories within its search space based on the robot sensory information.
- *Chapter 7* addresses the control system design paradigms for the mobile platform. Two system architectures are developed those are based on Artificial Immune System (AIS). The developed algorithms generate optimal trajectories within its search space based on the robot sensory information.
- *Chapter 8* deals with the experimental analysis of the developed mobile manipulator to control its motion with techniques described in the previous chapters. And also this chapter presents the circuit diagram of the various parts interfacing with microcontroller & its descriptions to control the motion of the mobile manipulator.
- *Chapter 9* addresses the conclusions.

Chapter 2

LITERATURE SURVEY

Manipulator Kinematics

Mobile Platform Locomotion

Mobile Manipulator Structure

Navigation of Mobile Platform

Objectives of the Thesis

Novelty of Work

Summary

2. LITERATURE SURVEY

A mobile manipulator in this study is a manipulator mounted on a mobile platform. Since the motion of the arm is unknown a priori, the mobile platform has to use the joint position information of the manipulator for motion planning. The study of the coordination and control of mobile manipulators includes several research domains. Major issues related to this thesis include the control of a wheeled mobile platform, the path planning of mobile robots and the coordination strategy of locomotion and manipulation of the mobile manipulator. A wide literature has been studied and analysed concerning the motion control issues for locomotion and manipulation.

2.1. Manipulator Kinematics

To perform various industrial operations robots can replace humans. The robot design engineers put forth assumptions as regard to the provision of links and joints of the mechanical articulated system. The Manipulation System of a robot arm is a mechanism which form a kinematic chain composed of rigid links and are joined by kinematic joints often with one degree of mobility (rotary or prismatic joints). A prime validation [1] of this choice is of a geometrical nature. At first sight the mechanical design architecture for a robot arm appears rather simple and yet it presents a very complex basic difficulty. Mechanisms and machines theory devoted greatly to planar and spatial mechanism synthesis with several degrees of freedom [2-4].

An approach to find the mathematical solution for simple manipulators is described in [5]. Mitrouchev [6] presented an overview about the design process of an industrial robot kinematic chain. The method allows the generation of plane mechanical structures with different degrees of mobility. The generalized solution of manipulators [7] kinematics and dynamics has been presented. They used 4×4 homogenous transformation matrices to describe the rotation and translation of manipulator links. Two and three link manipulator kinematics has been presented in [8]. They also implemented soft computing techniques to solve robot kinematics. From the generalised kinematic models of a manipulator with 6 degree of freedom; an alternative design introduced based on the concept of an in-parallel actuated mechanism [9]. The offline programming of industrial robots for the use in machining applications, three possible solutions are presented in [10-13] for overcoming kinematic constraints (end-effector rotation, part translation, and part rotation).

A new formulation method to solve the kinematic problem of multi arm robot systems based on screw theory and quaternion algebra has been presented by Sariyildiz and Temeltas [10]. Using the robot kinematics equation, a kinematic error model [11] is deduced by mapping structural parameters to the joint angular parameter. Dynamic error compensation method also discussed in their work through adding control algorithm to the compensation software. Accuracy and repeatability characteristics of industrial robot are analysed by developing an error tree [12]. The accuracy and repeatability of the manipulator are investigated utilizing the Denavit-Hartenberg kinematics parameters, the homogeneous transformation matrix, and the differential transformation matrix theory, and corresponding measures are developed. A Laser Tracker system based low-cost position measurement [14] setup has been developed for industrial robot kinematic parameters estimation The theoretic derivation of this calibration approach showed that the base calibration is not necessary.

The inverse kinematic models of a serial robot manipulator map the Cartesian space to the joint space. The solution for inverse kinematics problem is difficult since the relation between the joint space and Cartesian space is non-linear and involves complicated transcendental equations with multiple solutions.

Pozna [15] has been focussed on robot manipulators with rotation joints which are reciprocally perpendicularly/parallel. The study ensured by the general form of the link which lies two successive joint and the generality of the connection type between the link and the joint. Aydın and Kucuk [16] have been presented closed form solutions of the 6-DOF industrial robot manipulators with Euler wrist using dual quaternions. The developed tool can represent rotation and translation simultaneously. Forward and inverse kinematics solutions are presented For the TR 4000 educational robot arm [17] with 5-DOF. Later a software program is interfaced to show the robot motion with respect to its mathematical analysis. Pashkevich [18] has been dealt with the inverse transformation of coordinates for robotic manipulators with high payload capacity, those are widely used in welding applications. The developed algorithms allow users to obtain solutions to the prescribed configuration, and possess good convergence, even for singular manipulator locations. Inverse kinematics and dynamics algorithms for serial robots are presented in [19]. These robots are controlled by the joint actuators (required joint torques and forces) to produce the desired trajectory by its end effector.

As the complexity of robot increases, finding the inverse kinematics solution are difficult and computationally expensive since it requires the solution of nonlinear equations having transcendental functions. Alavandar and Nigam [20] have been presented the angular

difference is deduced and the data predicted with Adaptive Neuro-Fuzzy Inference System for two and three degree of freedom planar manipulator. In [21], for three-link planar manipulator inverse kinematic models have been developed for generating desired trajectories in the Cartesian space (2D) by using a feed-forward neural network.

2.2. Mobile Platform Locomotion

This section describes the preliminaries and the previous work related to solve mobile robot navigation problem. Numerous research works have been devoted to this area within the last decades. Large number of techniques has been developed in an attempt to solve this problem but no single technique has been universally accepted. An overview of major techniques used in mobile robot navigation has been addressed here.

2.2.1. Overview of motion planning concepts

The most important characterization of a motion planner is according to problem it solves. It means the effectiveness of the motion planner depends on properties of the robot solving the task. Once the task and the robot system are defined the next step is to develop algorithms to solve the specific problem.

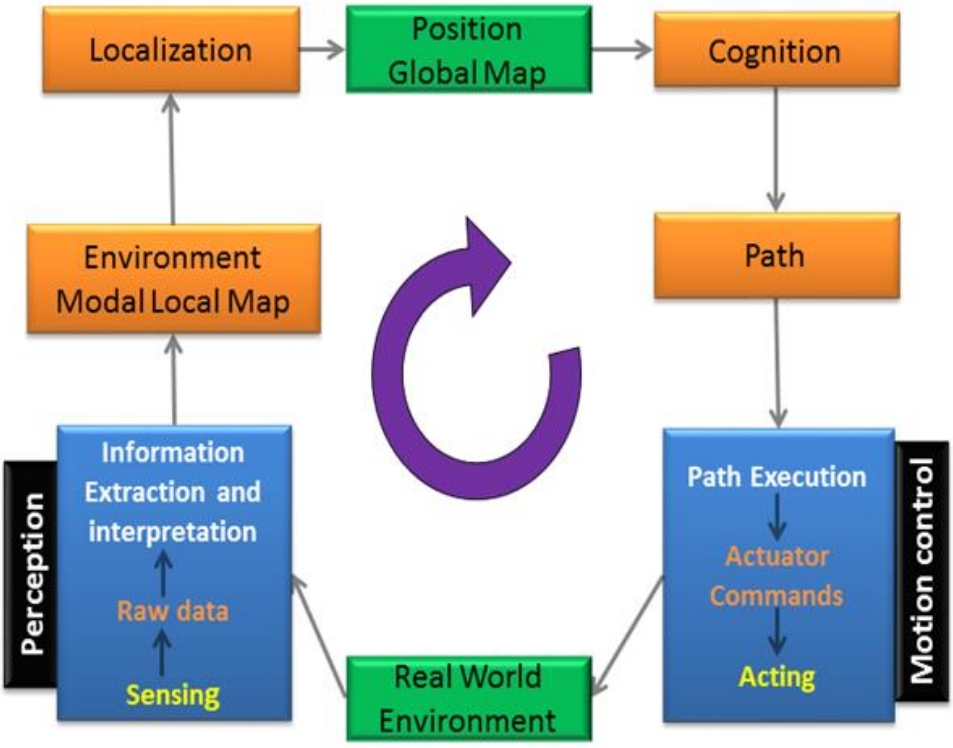


Fig.2.1 Control scheme for path execution of autonomous mobile robot

The different aspects under mobile robot motion control scheme are shown in Fig.2.1. These aspects of *motion planning* problem for autonomous mobile robot/robots includes:

→ Robot locomotive ability (most popular mechanism for robot motion)

→ Perception (acquiring robots environment condition)

→ Localization (determines the robot position in its environment)

→ Cognition (decision-making and execution to achieve its highest-order goals)

→ Motion Control (path execution in real world environment)

2.2.2. Wheel locomotion

In order to move a mobile robot from one position to another position it requires some locomotion mechanisms. So the selection of locomotion mechanism for a mobile robot design is an important aspect from a large number of possible ways to move. Most of these locomotion mechanisms [22-25] for a mobile robot, such as walk (human), slide (snake), run (4legged animal), jump (animals like kangaroo) etc. are biologically inspired. The selection of locomotion mechanism is desirable to motivate form the biological systems because they succeed in moving through various types of environments. But the manmade structural replication of biological systems is extremely difficult for certain reasons includes with:

→ Mechanical complexity (each part must be fabricated individually)

→ Biological energy storage systems used by animals (manmade hydraulic activating system)

→ Mathematical complexity (if a biological system is having large number of parts then it is difficult for motion analysis of a manmade system) etc.

Because of these limitations, the simplest biological locomotion mechanism with a small number of articulated legs are widely using.

But the mechanical complexity is more in legged locomotion because it requires higher degrees of freedom. In order to avoid this difficulty active powered wheels are equipped to the mobile robot. Wheel type locomotion mechanism is quite simple and is suitable for flat ground type environments. The only limitation with this mechanism is, when the environment (surface) becomes very soft it gives inefficient results due to rolling friction (between contact point of wheel & surface). In the past research work, several well-known algorithms and techniques have been proposed [26-28] for controlling mobile robot motion.

The relation between the robot chassis and the wheels equipped to it can be developed easily by considering the mobile robot as a planar rigid body [29]. The classification of wheels for a mobile robot can be classified into five types according to the geometrical constraints of

wheels. Then for each type of wheel, the structure of the kinematic and dynamic models has been analyzed in [30].

When a mobile robot is equipped with three independent steered standard wheels, the developed wheel architecture [31] is helpful to provide omni-directional motions. Mobile robots equipped with steered standard wheels are widely using because of their reliability and simple in mechanism. Kinematic relationships between the wheel slips of a steered wheel robot and positions of the instantaneous rotation centers have been developed in [32]. Cariou *et. al.* [33] have been considered sliding phenomena as both lateral and angular deviations and generated automatic path for a four wheeled steering vehicle. Marcovitz and Kelly [34] have been developed a method based on perturbative models to represent slip in all degrees of freedom, later they have implemented it to skid-steered vehicles.

Due to their mobility limitations of steered standard wheels on the plane, caster wheels are usually employing in most household goods utilize in home, offices and hospitals etc. The kinematics of a mobile robot having double-wheel-type active caster [35] has been investigated for moving the robot in its planar workspace. Chung *et. al.* [36] have developed kinematic models for holonomic and omni-directional mobile robot when it is equipped with offset steerable wheels. The dynamic model of a wheeled mobile robot equipped with powered caster wheels has been derived based on vehicle dynamics by introducing contact stability condition [37]. Kim and lee [38] have been derived the isotropy conditions based on the kinematic model of an omnidirectional wheeled mobile robot with three active caster wheels.

A Swedish wheel is nothing but a common wheel except that rollers are mounted on its perimeter. Because of rollers arrangement, the robot will have a full mobility the plane which means the robot can move in any direction i.e. the robot equipping with Swedish wheels is an omnidirectional mobile robot. Indiveri [39] has been introduced the issues of motion control and the kinematic models of wheeled mobile robot having N number of Swedish wheels. In [40], a path tracking control law has been addressed for wheeled mobile robots equipped with Swedish wheels. Doroftei *et. al.* [41] have been designed a Mecanum-wheeled mobile robot. Their robot is having four Swedish wheels in order to move the robot Omni directionally without any conventional steering system.

Another type of wheels is spherical wheels which offer greater mobility and stability to a mobile robot but the understanding of its controlling and motion planning is very complex. A novel spherical wheel has been introduced in [42] for an omni-directional mobile robot and is effective in step climbing capability with its hemispherical wheels. A novel combination of

Omni wheel and spherical wheel unit has been presented in [43] and its operating principle is basically a spherical wheel driven by two perpendicular pairs of Omni wheels. The motion planning problem for the rolling sphere is often referred as the ‘‘ball-plate problem,’’ and propose two different algorithms for reconfiguration as described in [44]. The first algorithm is based on standard kinematic model and invokes alternating inputs to obtain a solution comprised of circular arcs and straight line segments. The second algorithm is based on the Gauss-Bonet theorem which achieves reconfiguration through spherical triangle maneuvers. The kinematic model of a spherical wheeled mobile robot has been developed using quaternions [45] for the description of the orientation of the robot. Lauwers *et. al.* [46] have been presented the overall design of a single spherical wheel mobile robot, which can move directly in any direction.

The fixed standard wheel has no vertical axis of rotation for steering. Its angle to the chassis is thus fixed, and it is limited to motion back and forth along the wheel plane and rotation around its contact point with the ground plane. Since for a wheeled mobile robot, three fundamental characteristics are required: maneuverability, controllability, and stability; minimum two wheels are sufficient for static stability. A large amount of research work [47-52] has been focused on two-wheel differential-drive robot which can achieve static stability if the center of mass is below the wheel axle. Dhaouadi and Hatab [53] presented differential drive mobile robot dynamics, which will assist researchers in the modeling and design of suitable controllers for mobile robot navigation and trajectory tracking.

A tracking control method for differential-drive mobile robots with non-holonomic constraints by using a back stepping-like feedback linearization [47]. A nonlinear feedback path controller [48] is presented for a differential drive mobile robot. To control the differential robot kinematic model are developed, then fuzzy logic based control strategy is implemented [49]. Menn *et. al.* [51] have been presented a generic kinematic modelling approach for articulated multi-monocycle mobile robots. Later this methodology is implemented to the RobuRoc mobile robot. Feng *et. al.* [52] have been presented a model-reference adaptive motion controller implemented to a differential-drive mobile robot. This controller provides compensation for external errors.

2.2.3. Motion Control Strategy for differential mobile platform

The motion control of nonholonomic wheeled mobile robot has received considerable attention over the last few years and several tracking control algorithms have been addressed to solve this issue.

A novel adaptive trajectory tracking controller for a nonholonomic wheeled mobile robot has been presented in [53] with unknown parameters and uncertain dynamics. Sluný *et. al.* [54] have been presented two motion control architectures, evolutionary algorithm and a traditional reinforcement learning algorithm based on robot's localization. In [55], a local control strategy for the control of vehicle platoons has been addressed. In their investigation, the vehicle has the information about its own orientation, distance and azimuth of the leading vehicle. Ali [56] has been presented the development, implementation, and testing of a semi-autonomous robotic platform, which is used for educational and research purposes. A modular hardware design is employed to interface different sensors and motor drivers to the ATMEL microcontroller chip (AVR ATmega32). With reference to the kinematical model of a differential mobile robot, a known path following control law [57] is adapted to account for actuator velocity saturation. The proposed solution is verified experimentally for high speed applications.

2.3. Mobile Manipulator Structure

Mobile manipulator is a widespread term in nowadays to refer to robots that combine the capabilities of locomotion and manipulation. A mobile manipulator in this investigation consists of a robotic manipulator and a mobile platform resulting in a hybrid mechanism that includes a mobile robot platform for locomotion and a manipulator for manipulation. The tasks allotted to these robotic systems are often in terms of end effector evolution, either in point-to-point or in continuous motion execution.

Bayle *et. al.* [58] have been presented a generic approach of a kinematic controller for wheeled mobile manipulators. Their work discussed about the control of a simulated nonholonomic mobile manipulator, where the task is defined as a Cartesian trajectory of the end-effector. A simple and general kinematic model for non-holonomic mobile manipulator is derived in [59] by merging the manipulator kinematics with the permissible differential motion of the platform. A non-holonomic mobile manipulator has been considered in [60] which is built with a *n* DOF joint robotic arm and a non-holonomic mobile platform with two independently driven wheels. Viet *et. al.* [61] have been proposed a tracking control method for a three-wheeled omnidirectional manipulator system considering the effects of friction. The system is separated into two subsystems, a three-wheeled omnidirectional mobile platform and a selective compliant articulated robot for assembly type of manipulator.

Therefore, two controllers are designed; kinematic controller for the manipulator system and sliding mode controller to control the locomotion of the mobile platform.

Compared to a conventional mobile platform with two or four regular driving wheels, an omnidirectional mobile platform equipped with three DOF have the dexterous ability to move in any direction and to attain the desired orientation simultaneously [62-66]. Xu *et. al.* [67] have been proposed a neural network-based trajectory controller of the omnidirectional mobile manipulator with three castor wheels. Hung *et. al.* [68] have been addressed a control strategy for omnidirectional manipulator system to track a desired trajectory with a constant velocity and a desired posture of links. Datta *et. al.* [69] have been developed an indigenous autonomous mobile robot with a manipulator for carrying out tasks related to manufacturing. The developed system can navigate autonomously and transport jobs and tools in a manufacturing environment. An integrated motion planning and control framework for a nonholonomic wheeled mobile manipulator is presented in [70] by taking advantage of the (differential) flatness property.

Although the kinematic analysis of WMM has been treated a lot, but the concentration on dynamic analysis is few [71–75]. A novel approach [76] is presented for determining the maximum payload-carrying capacity of a coordinated mobile manipulator in an environment with presence of obstacle, based on stability. The suggested method considers the tip over stability on zero moment point criterion, when the path of the end-effector is predefined but the position of the mobile platform is free. Tchon *et. al.* [77] have been deduced a collection of inverse kinematic algorithms for mobile manipulators from a common root of inverse dynamic Jacobian. They defined the inverse dynamics with reference to the inequality that offers a sufficient condition for the convergence of the inverse kinematics algorithms. Trajectory tracking of a mobile manipulator has done by adapting a fuzzy control [78] and back stepping approach based on a dynamic model. The suggested control structure considers the dynamic interaction between the mobile platform and manipulator. Yamamoto and Yun [79-80] have been analyzed the effect of the dynamic interaction between a mobile platform and a mobile manipulator based on the task performance. Meghdari *et. al.* [81] have been studied the dynamic interaction between a one DOF manipulator and the vehicle of a mobile manipulator of a planar robotic system.

2.4. Navigation of Mobile Platform

Navigation related to mobile robots is moving towards the goal while avoiding local obstacles. Numerous researches have been devoted to this area within the last decades. Large

number of techniques has been developed in an attempt to solve this problem. No single technique has been universally accepted. Some techniques work only in ideal conditions and others solve local problems but not global problems. There are many approaches that combine both local and global navigation and seem to be the most effective ones. A large number approaches that capable of generating a valid path or trajectory through a cluttered environment. The research work presented in this thesis is an overview of techniques used in the navigation of mobile robots. Some short of obstacle avoidance is simply called as navigation of mobile robot. This problem includes:

(i) Cognitive mapping "is knowledge of a large scale environment used to find routes and determine the relative positions of places that is acquired by integrating observations."

(ii) Localization describes "where the robot is in a given environment based on the sensed information."

(iii) Path planning represents "the necessary relationships among the sub goals in order to achieve a particular target".

(iv) Motion Control "Concerned with the controlling of robot motion in order to reach its ultimate goal by path planning"

This section presents the path planning approaches of mobile robots. Path planning is classified into local path planning and global path planning. In former robot has prior knowledge about the orientation, shape and location even the movements of the obstacles in the environment. Later employs some reactive strategies to perceive the environment based on the sensory information such as distance information from either from sonar & laser sensors or visual information from cameras. Most of these approaches are basically dealt with the main problem of navigation of mobile robot.

2.4.1. Roadmap Methods

Roadmap methods basically used to reduce the dimensionality of the environment in which path planning and navigation of a mobile robot takes place. Well known methods in this type are the Voronoi diagrams and the visibility graphs.

Voronoi Diagrams & Visibility Graphs

A voronoi diagram records information about the distances between sets of points in any dimensional space. For path planning, Voronoi tends to be used in two dimensional space, where sets of points all lie within a plane. A plane is divided into cells so that each cell contains exactly one site. For every point in the cell, the Euclidean distance of the point to the site within the cell must be smaller than the distance of that point to any other site in the

plane. If this rule is followed across the entire plane, then the boundaries of the cells known as Voronoi edges will represent the point's equidistance from the nearest 2 sites. The point where multiple boundaries meet, called a Voronoi vertex, is equidistance from its 3 nearest sites. To find the generalized Voronoi diagram, it is necessary to compute the diagram exactly or use an approximation based on the simpler problem of computing the Voronoi diagram for a set of discrete points. O'Dunlaing and Yap [82] have been proposed the use of generalized voronoi diagram for motion planning in autonomous robots. Later many improvised models have been proposed. Ilhan and Howie [83] have been addressed a new sensor roadmap model called generalized Voronoi graph incremental construction, which was included with an already existing incremental construction procedure.

Vlassis *et. al.* [84] have been proposed a method which generated a Voronoi graph by the robot dynamically builds of its configuration space by applying an adaptive kk-means clustering algorithm to the robot's free space. They developed a Voronoi diagram of the robot's free space by the probabilistic clustering plan. The probabilistic growing cell structures algorithm includes i) Initialization, ii) Adaptation, iii) updating, iv) Cluster insertion and v) Cluster deletion. Later triangulation method applied to the resulting cluster centers to construct a graph. Lisien and Morales [85] have been presented a new map designed for robots operating in large free space and possibly in higher dimensions called hierarchical atlas. This hierarchical atlas has two levels: at the highest level there is a topological map that creates the free space into submaps at the lower level. The lower-level submaps are simply a collection of features. For other tasks such as navigation, obstacle avoidance, and global localization the resulting map is also useful. Lee and Howie [86] have been introduced a new roadmap which is used to guide a convex body to explore an unknown planar workspace called the convex hierarchical generalized Voronoi graph (convex-HGVG). The convex-HGVG composed of two components: 1) convex-GVG edges, which are three-way equidistant; and 2) convex-R edges, which are two-way equidistant paths that connect disconnected convex-GVG edges.

Visibility graphs

The visibility graph of a polygon is a graph whose nodes relate to their vertices of the polygon, and whose edges correspond to the edges of the polygon formed by connecting the vertices that can see each other. Visibility graphs may be used to find the shortest Euclidean paths among a set of polygonal obstacles in the environment. The shortest path between two obstacles except at the vertices of the obstacles follows straight line segments. So the Euclidean shortest path is the shortest path in a visibility graph is the start and destination

points and the vertices of the obstacles. Therefore, the Euclidean shortest path problem may be solved in the manner: constructing the visibility graph then apply a shortest path algorithm such as Dijkstra's algorithm to the graph. For planning the motion of a robot that has non-negligible size compared to the obstacles, a similar approach may be used after expanding the obstacles to compensate for this size of the robot.

Yamamoto *et. al.* [87] have proposed a near-time-optimal trajectory planning for car-like robots, where the connectivity graph is generated in a fashion very similar to that of speed optimization algorithm as proposed by Lozano-Perez & Wesley [88]. Oommen *et. al.* [89] have been presented an algorithm for navigation process composed of a number of travels and each traversal is from an arbitrary source point to an arbitrary destination point. In any stage of knowledge, a learned visibility graph is represented as the partially learned terrain model. After each traversal, this visibility graph is updated. It was proven that when the source and destination points are chosen randomly, the learned visibility graph converges to the visibility graph with probability one. Ultimately, optimized global path plan of the mobile robot was enabled by the availability of the complete visibility graph and it also eliminates the further usage of sensors. This Optimization technique includes the following theorems:

Theorem1: The procedure NAVIGATE-LOCAL always finds a path from S to D in finite time for a non-interlocking workspace.

Theorem2: If one point exists in a non-interlocking workspace, the procedure BACKTRACK leads to a solution to the navigation problem.

Theorem 3: If the LVG converges to the VG with a probability one then no point in the free space has zero probability measure of being a source or destination point or a point on a path of traversal.

Theorem4: If N is the total number of vertices of the obstacles then the number of sensor operations performed within the procedure Update-Vgraph to learn the complete VG is N^2.

The visibility graph of a simple polygon has the polygon's vertices as its point locations, and the exterior of the polygon as the only obstacle. Visibility graphs of simple polygons must be Hamiltonian graphs: the boundary of the polygon forms a Hamiltonian cycle in the visibility graph. However, the precise characterization of these graphs is unknown.

It is a major open problem in this area whether there exists a polynomial time algorithm that can take input as a graph (possibly together with a fixed Hamiltonian in the cycle corresponds to the boundary) and produce output as a polygon for the visibility graph, if such a polygon exists.

2.4.2. Geometric approaches & Cell decomposition method

To derive solutions to path planning and navigation problem these approaches use geometric methods. “Piano- Movers” problem is a famous problem in this category which deals with the ‘maneuvers’ required to move a long piano out of a room through a narrow door.

Choset *et. al.* [90] have been concerned with new method, by using conventional (Polaroid) low-resolution ultrasonic sensors mounted in a circular array on a mobile robot for improving the azimuth accuracy of range information. Then to fuse sonar data to better approximate the actual obstacle location a new method was introduced. This new method is known as the arc transversal median method because the robot find out the location of an object 1) with intersecting other arcs by one arc as shown in Fig 2.2; 2) by considering only “transversal” intersections, those which exceed a threshold in angle as shown in Fig.2.3, and 3) by taking the median of the intersections. This method can improve the azimuth accuracy of the sonar sensor by a specified amount under well-defined conditions via some simple geometric relationships.

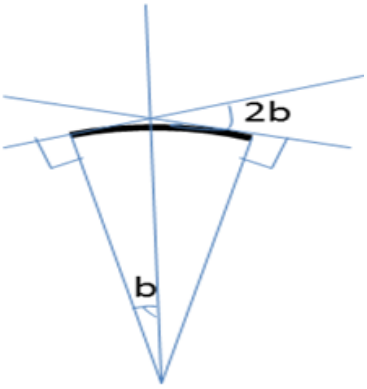


Fig.2.2. Object must be tangent to the arc, but there are an infinite number of orientation possibilities. Uncertainty is the beam width.

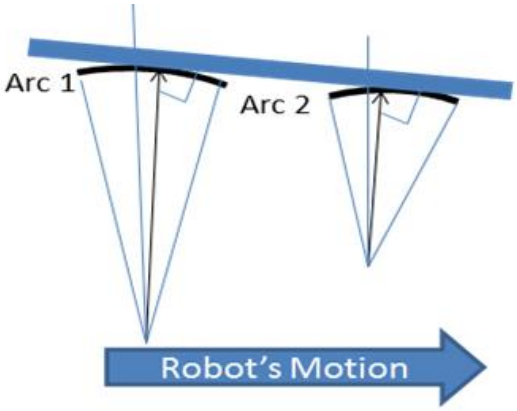


Fig.2.3 Object must be tangent to Arc1 and Arc2. Now there is only one possible orientation. Intersection points are the endpoints of the desired segment.

Miah and Gueaieb [91] have been outlined two different aspects using received signal strength of a customized radio frequency identification system for mobile robot navigation problem in indoor environments. First, a trilateration method is applied where the localization problem is solved through a geometric approach based on Cayley-Menger determinants to estimate the robot's current location. Second, a desired path along which a mobile robot is required to navigate can be found when the problem is explored by a set of points on the ground. Jan *et. al.* [92] have been introduced optimal path planning for navigating rectangular mobile robot based on a higher geometry maze routing algorithm among obstacles and weighted regions. To obtain an optimal collision free path with linear time and space complexities an 8-geomerty maze routing algorithm is applied. These methods cannot only search an optimal path among various terrains but also find the 2-D piano mover's problem with 3 DOF optimal paths. In addition to that, the algorithm can be easily extended among multiple autonomous robots or path planning problem in the 3-D space to avoid the dynamic collision.

In cell decomposition the free space is decomposed into a large number of small regions called cells. These cells define configuration where the robot can trivially navigate between them. Path construction here can be realized by an algorithm for graph search. These methods are perhaps the most well studied and have been applied widely for robot navigation.

In the cell decomposition approach [93] the free space is represented as a union of cells and a sequence of cells comprises a solution path. For efficiency, hierarchical trees, e.g. octree, are often used. The path planning algorithm proposed by Zelinsky [94] is quite reliable and combines some advantages of the above algorithms. It makes use of quad tree data structure to model the environment and uses the distance transform methodology to generate paths. The obstacles are polygonal-shaped which yields a quad tree with minimum sized leaf quadrants along the edges of the polygon, but the quad tree is large and the number of leaves in the tree is proportional to the polygon's perimeter and this makes it memory consuming.

Acar *et. al.* [95] have been defined an exact cellular decomposition where critical points of Morse functions indicate the location of cell boundaries. They introduced a general framework for coverage of tasks while varying the Morse function. The variation in Morse function effects the recognized pattern change by the robot. To construct the decomposition they described a sensor-based algorithm. Acar and Choset [96] have been achieved an exact cellular decomposition whose cells are defined in terms of critical points of Morse functions. Generically, a cell is defined by two critical points. Using a graph they encoded the topology of the Morse decomposition. While incrementally constructing a graph, the nodes in the

graph correspond to the critical points and the pairs of critical points represent the edges of the cells and simultaneously the robot covers its search space. To encounter all the critical points by the robot, they presented a complete algorithm thereby constructing the full graph, i.e., achieving complete coverage.

A planner presented in [97] that determines a robot path to reach its goal such that it does not have to heavily rely on odometry. To reach the goal the robot must follow sequence of obstacle boundaries determined by the planner. They used an exact cellular decomposition with simple structure to simplify the problem and introduced an algorithm to travel the robot between two points that does not heavily depend on dead-reckoning. Rekleitis *et. al.* [98] have been presented an algorithm on a team of mobile robots for the complete coverage of free space. Based on a single robot coverage algorithm their approach has been done. Single robot coverage is achieved by ensuring that the robot visits every cell. The new multi-robot coverage algorithm can be approached like a single robot planar cell-based decomposition. This method allows planning to occur for a team of *N* robots in a two-dimensional configuration space.

2.4.3. Virtual force field & Potential fields

In virtual force field method the robot determines its motion according to the virtual forces acting on it. In general, it is assumed that the repulsive forces are producing by obstacles whereas target position applying an attractive force on the robot. The summation of force vectors is usually determines the robot motion. The result is nothing but the robot will approach its target position due to the target attraction and simultaneously avoiding obstacles by giving repulsive from the obstacles so that the robot push away from them. The drawback of such methods is 'local minima'; in some situations where the attractive and repulsive forces are equal the robot can be trapped easily.

Olunloyo and Ayomoh [99] have been christened a new methodology hybrid virtual force field for an effective robot path planning, which is an integration of the virtual obstacle concept and the virtual goal concept in combination with the traditional virtual force field concept. The specific problem addressed for the local minima problem posed by concave shaped obstacles. Djekoune *et. al.* [100] have been addressed a novel method called DVFF combining the virtual force field (VFF) for obstacle avoidance approach and global path planning based on D* algorithm for navigation of a mobile robot in unknown environment. While the VFF local controller generates the admissible trajectories, D* generates global path information towards a goal position that ensure safe robot motion. Since it has the capability

of rapid re-planning in partially known or changing environment, the D* search algorithm is widely using with challenging terrains. By minimizing a predefined cost function D* search algorithm produces an optimal path from the start position to the target. In addition, D* algorithm can be easily combined with the real time obstacle avoidance algorithm developed because first, they both generate a direction, and second when we use local obstacle avoidance algorithm it does not necessary a C-space to enlarge the cells in the map grid according to the robot dimensions.

To determine the frontal repulsive force applied to the robot, the VFF method is applied at one critical point. By affixing six ultrasonic sensors at the front of the mobile robot (Fig.2.4) it can be measured the resultant repulsive force as the vectorial sum of the individual forces from all frontal occupied cells.

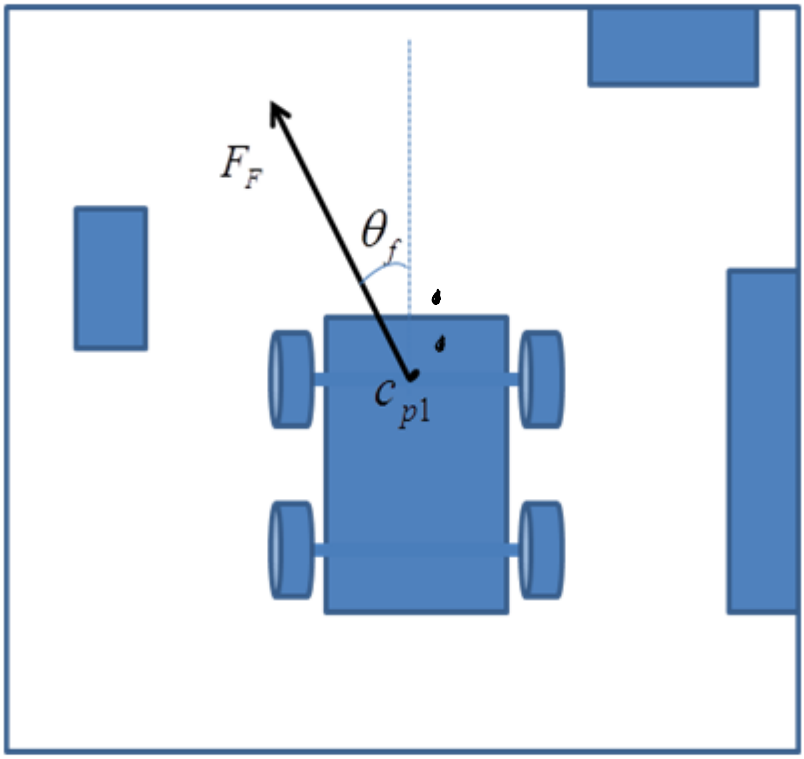


Fig.2.4 Computing the frontal repulsive force

Navigation of a mobile robot in the case of potential fields is defined as the process of following the maximum gradient of some particular quantity in the environment. This idea is used to identify target position by source of stimulus and gradient of the stimulus can be measured by the sensors attached on the robot, so that in the direction of maximum gradient of the stimulation, the robot is guided to the target. The gradient can be calculated at any position when the diffusion process takes place in all directions in the environment. Obviously, starting from any initial position a path can be constructed. This is a navigation

technique, which is a structure that "enhances the relationship between an environment and a specific target location in that environment". In some sense it includes the virtual force fields described above, in which gradient directions are determined by forces acting on mobile robot.

The potential field methods and their applications to path planning for autonomous mobile robots have been extensively studied in the past two decades [101-106].The basic concept of the potential field methods is to fill the workspace with an artificial potential field in which the robot is attracted to the goal position being at the same time repulsed away from the obstacles. Ideally, the potential field should have the following properties:

→ The magnitude of the potential field should be unbounded near to obstacle boundaries and should decrease with range.

→ The potential should have a spherical symmetry far away from the obstacle.

→ The equipotential surface near an obstacle should have a shape similar to that of the obstacle surface.

→ The potential, its gradient and their effects on the path must be spatially continuous

Conner *et. al.* [102] have been developed a method for navigation and control problem when the given system operating in a constrained environment. This method composed of simple control policies for both kinematic and simple dynamical systems, applicable over a limited region in a dynamical system's free space. This work assumed that the desired velocity vector fields are somewhat aligned with the initial velocities. An electrostatic potential field [104] through a resistor network is derived to represent the environment. It is used to solve two-dimensional collision free path planning problem for an autonomous mobile robot. Based on cell resolution they generated an approximate optimal path in a real-time frame. The path following the steepest gradient from the initial position to the goal position will be the path of least resistance. The derived potential function is a global navigation function and is free of all local minima. It includes four major modules: 1) Object Detection, 2) Localization, 3) Path Planning, and 4) Collision Avoidance.

While performing all tasks, the sensor based environment map generation and trajectory following are being inherently included in the four mentioned modules [105].

The algorithm to create the potential field follows four steps:

→ Introduce an occupancy map of the environment

→ Create the resistor network

→ Create the conductance map

→ Obtain the potential field by solving the resistor network.

Huang [105] has been applied a potential field method for a mobile robot in a dynamic environment where the target and the obstacles are moving. This method can be used for both path and the velocity planning. By relative velocities and relative positions among the robot, obstacles and targets the robot's planned velocity is determined. The proposed approach guarantees that the robot tracks the moving target while avoiding moving obstacles as shown in Fig.2.5.

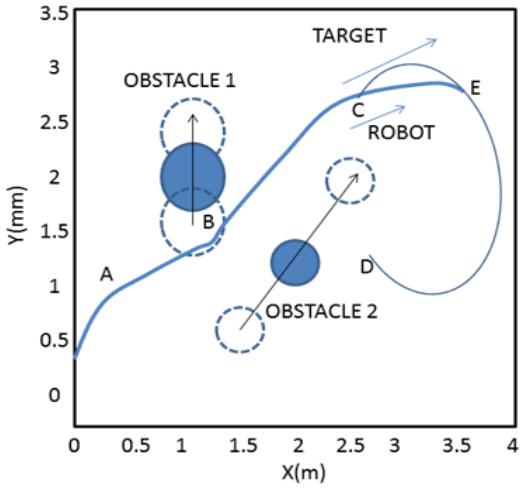


Fig.2.5 Trajectories of the robot (Solid: target, Dashed: robot) [105]

A path-planning algorithm presented by using a potential field representation [106] of obstacles for the classical mover's problem in three dimensions. A potential function is assigned to each obstacle and the topological structure of the free space is derived in the form of minimum potential valleys. In two levels Path planning is done. First, from the minimum potential valleys, a global planner selects the robot's path and its orientations. Second, a heuristic estimation is done along the path is to minimize the path length and the chance of collision avoidance. Then the path and orientations are modified by a local planner to derive the orientations and final collision-free path. A new path and orientations are selected by the global planner when the local planner fails and subsequently examined by the local planner. This algorithm runs much faster than exact algorithms and simultaneously solves much wider class of problems than other heuristic algorithms. This Algorithm summarized in the following steps:

i. The free space is represented by a graph consisting of a finite number of nodes and edges, corresponding to points and edges along MPV.
ii. Each node is assigned a cost depending on the width of free space at the node.

iii. A candidate path is found that minimizes both the path length and the chance of collisions.
iv. The local planner modifies the candidate path to derive a final collision-free path and orientations of the robot.
v. If a collision-free path cannot be found, remove the edge at which the unavoidable collision occurs, and go to (iii).
vi. Repeat (iii) to (v) until a solution is found or no candidate path exists.
vii. If a solution is found, further optimize the solution with the numerical algorithm.

2.4.4. Reactive approaches

Reactive approaches attempt to overcome the computational limitations of deliberative approaches, like the ones described already, by offering fast solutions for real-time applications. The basic idea is to actuation with respect to percept data, so that appropriate motor commands directly activate particular sensed patterns. Such couplings can be implemented as a simple structure. Although such approaches seems to be quite limited, when several behaviors are employed concurrently, the real power of reactive systems is exhibited and the overall behavior of the robotic system emerges as a result of some sort of combination of them (e.g. competition or weighted summation).

Reactive Robot Navigation:

A Purposive Approach has been used in [107] for the process of motion control based on the analysis of data gathered by visual sensors. The field of visual navigation is of particular importance mainly because of the rich perceptual input provided by vision. A new vision paradigm has attracted the interest of the computational vision research community. It is a vision based reactive navigation capability that enables a robot to navigate in indoor environments (long corridors, narrow passages), avoiding collisions with walls and obstacles. The term reactive is used to express lack of a particular destination that could be set by using maps of the environment, landmark recognition etc. Free space is defined based on the motor capabilities of the robot: since the robot moves on a plane. All those 3D structures are not belong to this plane can be potentially harmful, if the robot crashes to them. Therefore, these structures are considered as obstacles. By combining the extracted information from the two peripheral and one central camera of the proposed configuration, the following function can be computed:

$$A = C\left({}^{1}/_{Z_L} - {}^{1}/_{Z_R}\right) \quad (2.1)$$

In Eq. (1), A is a quantity that can be directly computed from the images acquired by the central and peripheral cameras. C is an unknown constant that depends on the characteristics of the body of the observer as well as its constant translational velocity. Finally, Z_L and Z_R are the depths perceived at the left and right peripheral cameras of the robot. A is equal to zero when the left and right cameras are in equal distances from the world and takes positive or negative values depending on whether the right camera is closer or farther from obstacles compared to the left camera. For the robot to move in the middle of the free space, A should be kept as close to zero as possible. Since A is a computable quantity, it can be used for controlling the rotational velocity of the robot.

The quantity A depends on the projection of the optical flow (u, v) along the intensity gradient direction (i.e. the perpendicular to the edge at that point) which is also known as normal flow. The normal flow is less informative than optical flow but can be computed robustly and efficiently from image sequences by using differentiation techniques. Moreover, in contrast to the computation of optical flow, no environmental assumptions such as smoothness are required for normal flow computation. A is the algebraic sum of several functions of normal flow. These functions are computed by selecting normal flow vectors in particular directions, which depend on specific motion parameters

However, the lack of any representation in reactive methods limits the scope of tasks they can achieve. To overcome this problem, recently several hybrid architectures have been addressed. Behavior-based approaches have been popularized within the last decade, mainly because of their success on real robots. They have provided a completely different paradigm for building 'intelligent' robots that departs from the traditional AI approaches. Some of the reactive approaches are provided below.

Bug Algorithm

Bug algorithm is a local path planning algorithm which uses sensors to detect the nearest obstacle when the mobile robot moves towards its target position with limited information about the environment. The algorithm uses obstacle border as guidance toward the target as the robot circumnavigates the obstacle till it finds certain condition to fulfil the algorithm criteria to leave the obstacle toward target point.

The first bug family algorithm has been proposed by Lumelsky *et. al.* [108], that incorporates a range sensor to calculate shortcuts relative to the path generated by Bug2 algorithm from or to m-line. Alg1 has been proposed by Noborio et.al. [109] to improve Bug2 weakness, it can trace the same path twice by storing the sequence of hit points occurring within an actual path

to the goal. These storing data are used to generate shorter paths by choosing opposite direction to follow an obstacle boundary when a hit point is encountered for the second time. The same researcher introduced the Alg2 to improve Alg1 by ignoring the m-line of Bug2 with new leaving condition. The Alg1 and Alg2 still face a reverse procedure problem where after encountering a visited point that causes loop; a mobile robot follows an uncertain obstacle by an opposite direction until it can leave the obstacle.

The reverse procedure problem solved in Alg1 andAlg2 by introducing a mixing reverse procedure [110] with alternating following method to create shorter average bound of path length and named the algorithm as Rev1 and Rev2. Other bug algorithms that also incorporate range sensors are DistBug algorithm and TangentBug Algorithm. The basic operation of mobile robot using bug algorithm can be viewed as following pseudo code

```
While Not Target
If robot rotation <= 360
        Robot rotates right of left according to position of dmin
        If sudden point
                If 180 degree rotation
                        Ignore reading /* to avoid robot return to previous point */
                Else
                        Get distance from current sudden point to next sudden point
                        Get angle of robot rotation
                        Move to new point according to distance and rotation angle
                        Record New dmin value
                        Reset rotation
                End if
        End if
        Robot Stop /* No sudden point and exit loop */
End if
While end
Robot Stop /* Robot successfully reaches target *
```

Devi and Prabakar [111] have been developed an algorithm to navigate a point robot in planar unknown environment with stationary obstacles of arbitrary shape. It determines where the next point to move toward target from a starting point. The next point is determined by output of range sensor which detects the sudden change in distance from sensor to the nearest obstacle. The sudden change of range sensor output is considered in

constant reading of distance either it is increasing or decreasing. The initial position of robot is facing straight to the target point and then the robot rotates left or right searching for sudden point. After the first sudden point is found, the rotation direction of the robot is according to position of straight line between current sudden point and target point. This algorithm attempts to minimize the use of outer perimeter of an obstacle (obstacle border) by looking for a few important points on the outer perimeter of obstacle area as a turning point to target and finally generate a complete path from source to target. The less use of outer perimeter of obstacle area produces shorter total path length taken by a mobile robot.

Ant colony Optimization, Bacteria foraging & Particle swarm optimization

Ant Colony Optimization is one of the search algorithms, which have been successfully applied to solving NP hard problems. From the behavior of colonies of real ants, ant algorithms are inspired and in particular how they forage for food. The ants can communicate with one another is one of the main ideas behind this approach. Ant colony optimization has been successfully applied to both classes of traveling salesman problem with good and often excellent results.

The ACO algorithm skeleton for this problem is as follows:

```
Set parameters, initialize pheromone trails
While (termination condition not met) do
        Construct Solutions
        Apply Local Search % optional
                Update Trails
        end
end
```

Zheng *et. al.* [112] have been proposed a novel method based on the ant colony system (ACS) algorithm for the real-time globally optimal path planning of mobile robots. This method comprised three steps: the first step is to establish the free space model of the mobile robot by utilizing the MAKLINK graph theory, the second step is to find a sub-optimal collision-free path by utilizing the Dijkstra algorithm, and the third step is to generate the globally optimal path with the help of ACO algorithm. It is needed to calculate the cost function of a path when using the Dijkstra algorithm. In general, each edge on a path will be given a weight, so the cost function can be defined as the sum of the weights of all the edges of the path. To use the MAKLINK graph theory for establishing the free space, the following assumptions made:

→ The heights of the environment and obstacles can be ignored.

→ There exist some known polygonal shape obstacles distributed in the environment.

→ In order to avoid a moving path too close to the obstacles, the boundaries of every obstacle can be expanded by an amount that is equal to half of the greater size in the length and width of the robots body plus the minimum measuring distance of the relevant sensors.

Garcia *et. al.* [113] have been presented a new method was named SACOdm, where d stands for distance and m for memory based on Simple Ant Colony Optimization Meta-Heuristic. In the proposed ACO, the existing distance between the source and target nodes influences the decision making process, moreover the ants can remember the visited nodes. The selection of the optimal path relies in the criterion of a fuzzy inference system, which is adjusted using a simple tuning algorithm. The path planner application has two operating modes, one is for virtual environments, and the second one works with a real mobile robot using wireless communication. Both operating modes are support static and dynamic obstacle avoidance and global planners for plain terrain. Since equivalent distribution of initial pheromone in ant colony optimization algorithm results more search period, pheromone gain and heuristic information are integrated to reasonably allocate initial pheromone [114]. In [115-116] ACO is used to find the shortest path of mobile robot while avoiding obstacles and reach to its target from the source position. Furthermore, a global attraction term is added which leads ant to reach the goal point.

Natural selection tends to eliminate species with poor foraging strategies and favour the propagation of genes of species with successful foraging behavior. Considering all the constraints presented by own physiology of foraging organism or animal such as sensing and cognitive capabilities of environment, the natural foraging strategy can lead to optimization for solving real-world optimization problems. Based on this conception, Passino proposed an optimization technique known as the Bacterial Foraging Optimization Algorithm (BFOA). Until date, BFOA applied successfully to practical problems such as optimal controller design, harmonic estimation, transmission loss reduction, active power filter synthesis, and learning of artificial neural networks.

The classical BFOA algorithm: The bacterial foraging system includes with four principal mechanisms, namely chemotaxis, swarming, reproduction, and elimination-dispersal. Below we describe briefly each of these processes and provide a pseudo-code of the complete algorithm.

i) Chemotaxis: Through swimming and tumbling via flagella, an E.coli cell will simulates to move. Biologically an E.coli bacterium can move in two different ways as swim and tumble and alternate between these two modes of operation for the entire lifetime.

ii) Swarming: An interesting group behavior has been observed for several motile species of bacteria including E.coli and S.typhimurium, in semisolid nutrient medium intricate and stable spatio-temporal patterns are produced. A group of E.coli cells arrange themselves in a traveling ring by moving up the nutrient gradient when placed amidst a semisolid matrix with a single nutrient chemo-effecter. The cells release an attractant aspertate when stimulated by a high level of succinate, which helps them to aggregate into groups and thus move as concentric patterns of swarms with high bacterial density.

iii) Reproduction: The least healthy bacteria eventually die while each of those yielding lower value of the objective function asexually split into two bacteria, which are then placed in the same location. This keeps the swarm size constant.

iv) Elimination and Dispersal: Due to various reasons e.g. a significant local rise of temperature may kill a group of bacteria that are currently in a region with a high concentration of nutrient gradients, in the local environment where a bacterium population lives Gradual or sudden changes may occur. Events can take place in such a way that a group is dispersed into a new location or all the bacteria in a region are killed. To simulate this phenomenon in BFOA the new replacements are randomly initialized over the search space and some bacteria are liquidated at random with a very small probability.

Dasgupta *et. al.* [117] have been presented a mathematical analysis of the chemotactic step in BFOA from the viewpoint of the classical gradient descent search. The classical BFOA pointed out that the chemotaxis analysis, usually results in sustained oscillation, especially on flat fitness landscapes, when a bacterium cell is close to the optima. Two simple schemes for adapting the chemotactic step-height have been proposed to accelerate the convergence speed of the group of bacteria near global optima. Finally they investigated an interesting application to the frequency-modulated sound wave synthesis problem, appearing in the field of communication engineering. In the proposed BFOA based path planning [118], the bacterial chemotaxis mechanism enables the robot to explore the environment and finally locates the target point without colliding any obstacles, while the self-adaptive foraging strategy save the traveling time and actuators energy of the self-navigating robot. A new method proposed in [119] by using BFO algorithm to solve mobile robot path navigation problem. In their work constraints are defied i.e. site fix penalty and route fix penalty; such that the path generated by the robot will avoid the obstacles in its environment.

Finally they proved the proposed BFO algorithm results are efficient than genetic algorithm and differential evolution.

Kennedy and Eberhart [120] have been developed an evolutionary computational method named as Particle Swarm Optimization (PSO), which was motivated by social behaviour of bird flocking or fish schooling. Because of its special features like proximity, quality, diverse response, stability and adaptability; it has been successfully implemented to solve many engineering problems. PSO is used it to solve the optimization problems. A "bird" in the search space is each single solution in PSO is called a "particle". All of the particles have fitness values which are evaluated by the fitness function to be optimized, and have velocities which direct the flying of the particles. In every iteration each particle is updated by following two "best" values. After finding the two best values (position and global bests), the particle updates its velocity and positions.

The pseudo code of the procedure is as follows

```
for each particle
   Initialize particle
end
while maximum iterations or minimum error criteria is not attained
do
   for each particle
      Calculate fitness value
      If the fitness value is better than the best fitness value (pBest) in history
         set current value as the new pBest
   end
   Choose the particle with the best fitness value of all the particles as the gBest
   for each particle
      Calculate particle velocity & Update particle position
   end
```

Hassan *et. al.* [121] have been explained, that PSO is more efficient in computational view (uses less number of function evaluations) than the Genetic Algorithm. Because of its effectiveness and faster response, various authors have been applied PSO for solving scientific problems such as unknown parameters estimation in nonlinear systems [122], Bioinformatics [123], Machine Learning [124], job-shop scheduling [125] and constrained optimization problem [126] etc..

Path planning of a mobile robot can be considered as a multi objective optimization problem because it includes generation of trajectories from its source position to destination with less distance/time traversal and avoiding obstacles within its known/unknown environments. In past, researchers have been applied PSO for solving mobile robot path planning problem. It has been proved by Venaygamoorty and Doctor [126], for optimal navigation of mobile sensors, the time taken by convergence with PSO is ten times faster than time taken by fuzzy logic. Zhang and Li [127] have been applied PSO for motion planning of a robot, when the work space is having the obstacles of generalised polygons. But their implementation may not generate optimal paths in all situations. To overcome this difficulty Qin *et. al.* [128] have been applied PSO with mutation operator. But this approach requires lot of work to adjust the controlling parameters of PSO. Maschian and Sedighizadeh [129] have recently developed a novel PSO based motion planner for an autonomous mobile robot. They have done a large amount of work for adjusting the controlling parameters of PSO and generating optimal trajectories between two successive robot positions. Even though, their work requires an adaptive algorithm to generate safest path within its environments.

The above mentioned algorithms are useful only for known environments, but their implementation can't be applied for unknown/partially known environments. Derr and Mannic [130] have been outlined a PSO based computational method for motion planning of robots in noisy environments, but this methodology increases the robot search time in finding its target. Several researchers [131, 132] have applied PSO for obstacle avoidance in collective robotic search within the robotic noisy environments. Lu and Gang [133] have proposed an algorithm using PSO, for generating optimal path of a mobile robot in unknown environment. However their implementation lacks in adjusting the controlling parameters of their developed fitness function to improve the performance of system architecture.

Fuzzy Logic Approach

Lotfi Zadeh developed a fuzzy set theory, has become a popular tool for control applications. In servomotor and process control applications, Fuzzy control has been used extensively. One of its main benefits is that without a person need, it can give a mathematical description of the problem. This system can also incorporate a human being’s expert knowledge about how to control a system. Fuzzy logic is used to keep the robot on its path, except when the danger of collision arises. In this case, a fuzzy controller for obstacle avoidance takes over. The basic idea of Fuzzy Logic Control (FLC) is to translate a sensor data into a label as performed by

human expert controllers. The general structure of a fuzzy logic control is presented in Fig.2.6.

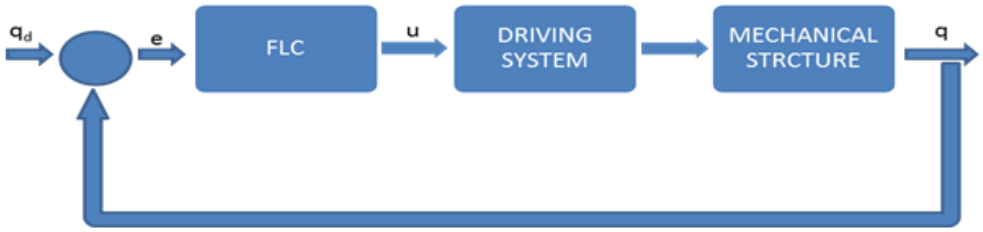


Fig.2.6. General Structure of a fuzzy logic control

In order to secure the desired performances of the system, FLC implements the control law as an error function. It contains three main components: the fuzzifier, the inference system and the defuzzifier as represented in Fig.2.7. The fuzzifier has the role to convert the measurements of the error into fuzzy data. In the inference system, linguistic and physical variables are defined. The universe of discourse, the set of linguistic variables, the membership functions and parameters are specified for the each physical variable. One option giving more resolution to the current value of the physical variable is to normalize the universe of discourse. Defuzzifier combines the reasoning process conclusions into a final control action. Different models may be applied, such as: the most significant value of the greatest membership function, the computation of the weighted average of all the concluded membership functions.

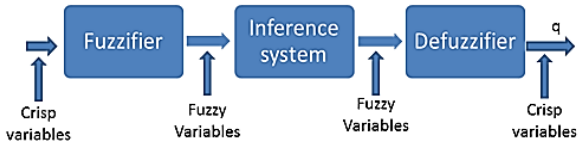


Fig.2.7 The structure of the fuzzy logic control

Bin and Wenfeng [134] have been concerned with a new method using fuzzy logic for behavior based control of mobile robots navigation in unknown environments. The basic behaviors are designed to form complex robotics system based on fuzzy control technique and are integrated and coordinated. To approximate a robot towards its target location while avoiding obstacles, the output from the target steering behavior and the obstacle avoidance behavior are combined. Narvydas *et. al.* [135] have been presented a possibility to use FUZZY logic, IF-THEN rules, and Genetic Algorithm for autonomous mobile robot control is presented. They proved that control using fuzzy logic is softer and better then control using IF-THEN rules.

In the robotics it is necessary to pay much attention on the intelligent hybrid control systems and evolutionary algorithms as well as to improve both the hardware and the software. An efficient control systems can be designed in two stages:

1) To perform a specific task, create a hybrid intelligent control system.
2) To improve the results of the first stage, use evolutionary programming methods.

The first one related to the creation of the framework of a FUZZY control system (FCS), and the second one related to the methodology using the Genetic Algorithm (GA) to improve the control system. Islam *et. al.* [136] have been introduced a designing model for autonomous Mobile Robot Controller (MRC) hardware with navigation concept using Fuzzy Logic Algorithm (FLA). Without human intervention, the MRC enables the robot navigation in unstructured environment to avoid collisions from the encountered obstacles. Obstacle avoidance can be done by turning the robot to proper angle with the help of designed hardware architecture. Noureddine *et. al.* [137] have been implemented an efficient positioning of an autonomous car-like mobile robot, respecting to its final orientation. Two FLCs, robot positioning controller and robot following controller have been developed to accomplish oriented positioning of the robot as shown in Fig. 2.8.

Pradhan *et. al.* [138] have been investigated navigation techniques using fuzzy logic for several mobile robots as many as (one thousand) robots in a totally unknown environment. First a fuzzy controller has been used with four types of input members and two types of output members with three parameters in each. Next two types of fuzzy controllers developed having same input members and output members with five parameters each. Each robot has an array of ultrasonic sensors and an infrared sensor for measuring the distances of obstacles around it and the target. Amongst the developed techniques, FLC has Gaussian membership function is found to be most efficient for mobile robots navigation. Martinez *et. al.* [139] have been developed a fuzzy logic based intelligent control strategy for handling the collision avoidance problem. A mobile robot system was tested by the fuzzy controller in an indoor environment and found to perform satisfactorily despite having crude sensors and minimal sensory feedback.

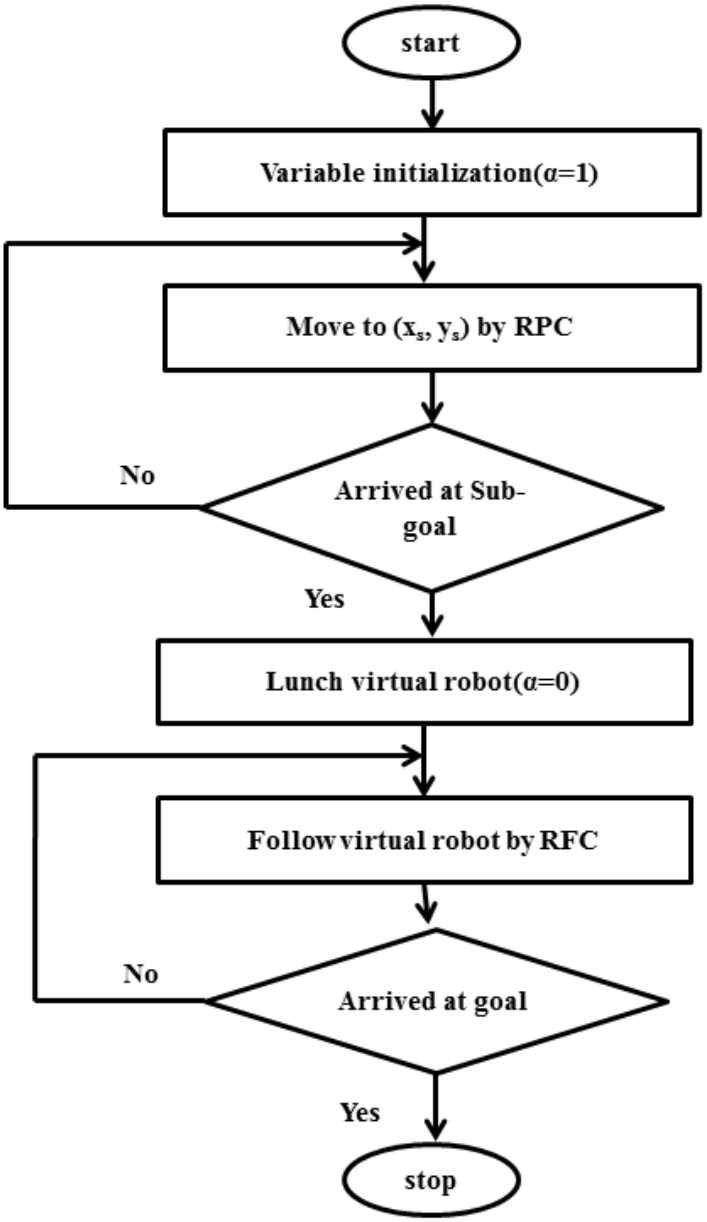


Fig.2.8 Flow chart of oriented positioning [137]

Neural network

Artificial Neural Network (ANN) is an attempt at modeling the information processing competencies of nervous systems. The nervous system of an animal has its information processing unit which comprises of sensory inputs, i.e., signals from the environment, are coded and processed to remind the suitable response.

To solve the motion-planning problem of mobile robot, ANN based intelligent control strategy has been developed in [140]. This methodology works in the environment with any number of obstacles of arbitrary shape and size; some of them are allowed to move. A dynamic artificial neural network based mobile robot motion and path planning system is

addressed in [141]. This method has the able of navigating the robot car on flat surface among static and moving obstacles, from any starting point to any endpoint. Neural based systems have been developed in [142] for mobile robot reactive navigation. The proposed system transforms sensors input to yield wheel velocities.

Neuro- Fuzzy system

In the field of artificial intelligence, combination of artificial neural networks and fuzzy logic is referred as neuro-fuzzy. Neuro-fuzzy hybridization results in a hybrid intelligent system that synergizes these two techniques by combining the human-like reasoning style of fuzzy systems with learning and connectionist structure of neural networks. Neuro-fuzzy system (the more popular term is used henceforth) incorporates the human-like reasoning style of fuzzy systems through the use of fuzzy sets and a linguistic model consisting of a set of IF-THEN fuzzy rules. The main strength of neuro-fuzzy systems is that they are universally approximate with the ability to solicit interpretable IF-THEN rules. The strength of neuro-fuzzy systems involves two contradictory requirements in fuzzy modeling: interpretability versus accuracy. In practice, one of the two properties prevails. The neuro-fuzzy in fuzzy modeling is divided into two fields: linguistic fuzzy modeling that is aimed on interpretability and precise fuzzy modeling that is aimed on accuracy.

Kim and Trivedi [143] have been developed A Neural integrated Fuzzy controller (NiF-T) for nonlinear dynamic control problems which combines the fuzzy logic representation of human knowledge with the learning capability of neural networks. The NiF-T architecture includes three distinct parts as shown in Fig.2.9: (1) Fuzzy logic Membership Functions (FMF), (2) Rule Neural Network (RNN), and (3) Output-Refinement Neural Network (ORNN). To fuzzify input parameters FMF are utilized. After defuzzification RNN interpolates the fuzzy rule set, the output is used to train ORNN. The weights of the ORNN can be adjusted on-line to fine-tune the controller. NiF-T can be applied for a wide range of sensor-driven robotics applications, which are characterized by high noise levels and nonlinear behavior, and where system models are unavailable or are unreliable. Tahboub and Munaf [144] have been presented a Neuro-fuzzy reasoning approach has the advantage of greatly reducing the number of if-then rules by introducing weighting factors for the sensor inputs, thus inferring the reflexive conclusions from each input to the system rather than putting all the possible states of all the inputs to infer a single conclusion.

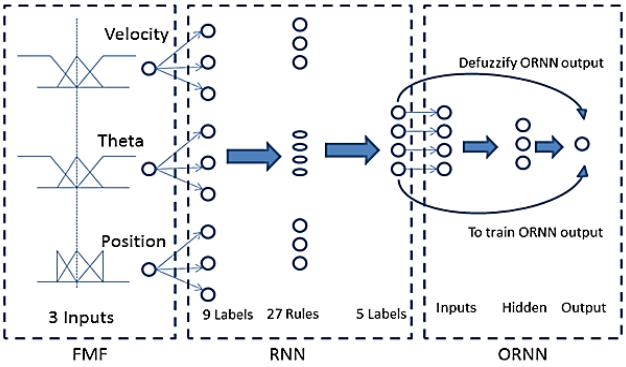


Fig.2.9 The NiF-T model and its three main modules [143].

To determine the weighting factors for the distance readings acquired by the robot's sensory system four simple neural networks are used. These weighting factors represent the degree of collision avoidance by the robot from a certain side then these values are treated as fuzzy values that are input to a defuzzifier to come up with a crisp value for the robot's steering angle and speed. For one sensor input each neural network is responsible for determining the weighting factor. The suggested system has two advantages. First, if-then rule base is replaced by a set of simple neural networks. Second, inference is on the reflexive conclusions from each input to the system [145], in which a RAM-based neural network method was combined with the fuzzy logic strategy to decrease the number of fuzzy rules and related processing. Using a simple, 8-bit microcontroller the feasibility of this neuro-fuzzy approach was demonstrated on a mobile robot. The neuro-fuzzy approach is code-efficient, fast, and easy to relate to the physical world and act as an action supervisor. Based on its observation of the environment, a RAM-based neural network chooses the reactive outputs from these fuzzy logic controllers. It classifies the current environmental conditions based on sensor inputs and then chooses the best control response from the outputs of the two fuzzy logic controllers.

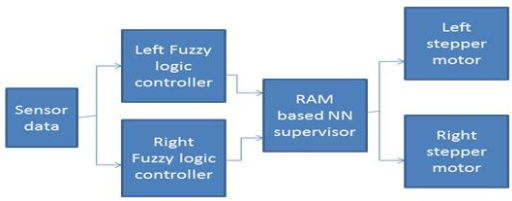


Fig.2.10. Block diagram of the neuro-fuzzy navigation system [145]

Artificial immune system(AIS)

The interest in studying the immune system is increasing over the last few years. Computer scientists, engineers, mathematicians, philosophers and other researchers are particularly interested in the capabilities of this system, whose complexity is comparable to that of the

brain. Many properties of the immune system are of great interest for computer scientists and engineers:

→ Uniqueness: each individual possesses its own immune system.

→ Recognition of Foreigners: the antigens are recognized and eliminated from body.

→ Anomaly Detection: the immune system can detect and react to pathogens that the body has never encountered before.

→ Distributed Detection: the cells are distributed all over the body and are not to subject any centralized control.

→ Noise Tolerance: the system is flexible since the recognition of the antigens is not required.

→ Reinforcement Learning and Memory: future responses to the same pathogens are faster and stronger since the immune system can "learn" the structures of pathogens.

In recent years, researchers have applied artificial immune system algorithms to autonomous mobile robot for generating collision free trajectories [146-152]. While developing immunological system architectures, they have been investigated the interactions among the various immune components.

Mamady *et. al.* [148] have developed a new immunised computational method for mobile robot navigation, if the robot environment having uniform mass and general shape objects. To get better results from the methodology as described in [151], it is necessary to evolve the immune network by the presence of much more connections, but this will increase the network complexity. An adaptive AIS mechanism has been introduced in [152] for arbitration of an autonomous mobile robot. An adaptive learning mechanism based on immune system [153, 154] has been developed for Lego robots. Later they applied their mechanism for two moving robots on their pre-defined near concentric tracks. An AIS based robot navigation has been described in [155], but their implementation is in early state and requires a lot of work to apply this mechanism to a real robot. An immune system has been presented to achieve behavior of a mobile robot learns to detect vulnerable areas of a track as well as adapts to the required speed over such environments. The test bed comprises of two Lego robots placed simultaneously on two predefined near concentric tracks. When inner one misaligns from the track, outer robot will help the inner one to keep it on predefined track. The panic-stricken robot records the conditions under which it was misaligned and learns to detect and adapt under similar conditions thereby making the overall system immune to such failures.

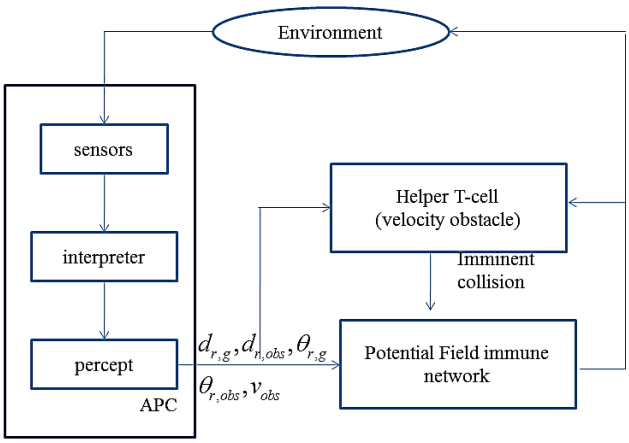


Fig.2.11 The architecture of the potential field immune network [103]

Guan-Chun and Weiwen [103] have been proposed a potential filed immune network (PFIN) as represented in Fig. 2.11 for dynamic navigation of mobile robots with moving obstacles and fixed/moving targets in an unknown environment. To determine imminent obstacle collision for a robot moving in the time varying environment the velocity obstacle method is utilized. With the aid of fuzzy system the response of the overall immune network is derived. Whitbrook *et. al.* [156] have been addressed a method for integrating an idiotypic AIS network with a Reinforcement Learning (RL) based control system. A simplified hybrid AIS-RL that implements idiotypic selection independently of derived concentration levels and a full hybrid AIS-RL scheme are examined. Huang *et. al.* [157] have been presented a metaheuristic artificial immune system (AIS) algorithm for mobile robot navigation in a structured environment with obstacles. The path generated from the AIS planner is then smoothed using the cubic B-spline technique, in order to construct a collision-free continuous path.

2.5. Objectives

Since the motion of the arm is unknown a priori, the mobile platform has to use the joint position information of the manipulator for motion planning.

The following objectives were accordingly specified for this project work:

- Forward and inverse kinematic analyses are performed on the robotic arm and the mobile platform in order to navigate the robot and manipulate the objects in desired direction.
- Modeling of the arm is carried out in CAD software ‘CATIA V6' in order to simulate each part of the manipulator.

- Intelligent planning and control algorithms are developed for the platform so that the manipulator is always positioned at the preferred configurations measured by its manipulability.
- Control scheme associates the mobile base activities and the manipulator activities to produce an intelligent hybrid system that performs a coordinated motion and manipulation.
- Finally, experimental analysis is performed in order to validate the efficiency of the developed control strategies.

2.6. Novelty of Work

The objectives of this research work are to address novel methodologies such as modified particle swarm optimisation and artificial immune system to control mobile manipulator. The main challenge in these applications is that the robots have less prior knowledge about the environment while performing their jobs. In the current research suitable probabilistic relational models based on swarm intelligence and immune system are used by robots to perform their tasks intelligently.

2.7. Summary

This section described the related work on the mechanical structure of a conventional robot arm and its control strategies according to its kinematic constraints. Later the survey is extended to mobile platforms, its configurations and the motion/velocity control laws according to its geometric constraints. Because of its limited workspace, recently researchers are concentrated hybridisation of locomotion and manipulation. Various techniques developed to control mobile manipulation system is also emphasised in the current section. Moreover, this section addressed various approaches used for navigation of mobile platform.
From this chapter, it is noticed that the mobile robot navigation can be controlled successfully in a complex environment using various techniques as described. Keeping in view of the above survey, researcher can focus for finding out efficient control strategy for navigation of mobile robots for any cooperative or individual task in multidimensional robot environment.

Chapter 3

MECHANICAL DESIGN ARCHITECTURE OF THE ROBOT ARM

Rotation Kinematics

Homogeneous Transformation

Manipulator Kinematics

D-H notation

Forward Kinematic Model

Inverse Kinematic Model

Summary

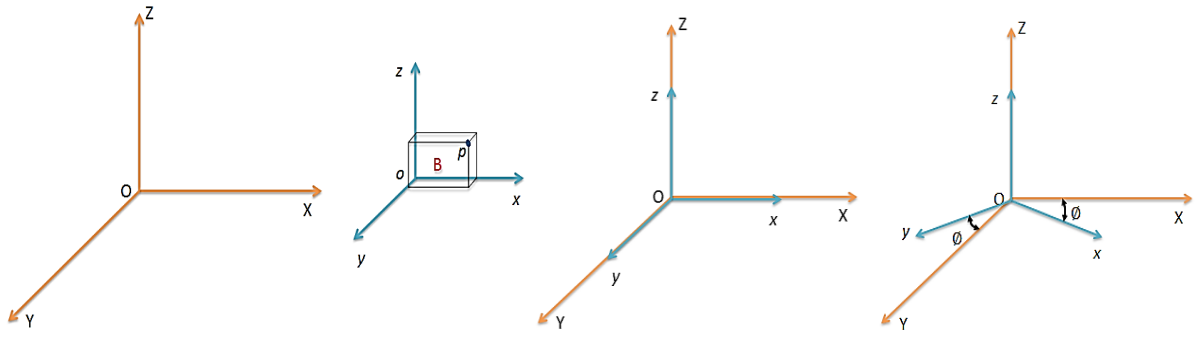
3. MECHANICAL DESIGN ARCHITECTURE OF THE ROBOT ARM

Manipulator kinematics deals with the geometric motion of a robot arm with respect to a fixed reference frame without concerning the forces/moments that cause the motion. Thus, kinematic models of robot obtain the relations between the joint-variable space and the position and orientation of the end-effector of a robot arm. Positioning is to bring the end-effector of the arm to a specific point within its workspace, whereas orientation is to move the end-effector to the required orientation at the specific position. The positioning and orientation are the jobs with respect the arm and wrist respectively. Sometime the positioning and orientation of the end-effector can decouple to simplify the kinematic analysis.

3.1. Rotation Kinematics

The links in a robotic system are modeled as rigid bodies. Therefore, the characteristics of a rigid body displacement take a vital place in robotics. Since the robot links may rotate / translate with respect to each other, it is required to find their relative configurations with respect to world reference frame. The position of one link B relative to another link A is defined by a coordinate transformation ${}^{A}T_{B}$ between reference frames attached to the link.

Consider a global coordinate frame OXYZ and a rigid body B with a local coordinate frame oxyz as shown in Fig.3.1(a). Initially the body B is fixed to the ground G and these two coordinate frames are coincident at point O as represented in Fig. 3.1(b).



(a) (b) (c)

Fig.3.1 a) Global & Local Coordinate frames, b) Initial frames position and c) Local frame rotation with respect to global frame

If the rigid body B rotates about the Z-axis of the global coordinate frame with ∅ degrees as shown in Fig. 3.1(c), then coordinates of a body point P in the local and global coordinate frames are related by the Eq. (3.1).

$$G(p) = R(Z, \emptyset)B(p) \tag{3.1}$$

Here $R(Z, \emptyset)$ is rotational mapping matrix

$$G(p) = \begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix} \text{ and } B(p) = \begin{Bmatrix} x \\ y \\ z \end{Bmatrix}$$

Let (i_X, j_Y, k_Z) and (i_x, j_y, k_z) be the unit vectors along the coordinate axes of the OXYZ and oxyz systems respectively.

$p_{XYZ} = Xi_X + Yj_Y + Zk_Z = xi_x + yj_y + zk_z = p_{xyz}$

Thus, using the definition of a scalar product the point *p* can be defined as the components of a vector as represented in Eq. (3.2).

$$\left.\begin{aligned} X &= i_X \,.\, p = i_X \,.\, i_x x + i_X \,.\, j_y y + i_X \,.\, k_z z \\ Y &= j_Y \,.\, p = j_Y \,.\, i_x x + j_Y \,.\, j_y y + j_Y \,.\, k_z z \\ Z &= k_W \,.\, p = k_W \,.\, i_x x + k_W \,.\, j_y y + k_W \,.\, k_z z \end{aligned}\right\} \tag{3.2}$$

$$\Rightarrow \begin{Bmatrix} p_X \\ p_Y \\ p_Z \end{Bmatrix} = \begin{bmatrix} i_X \,.\, i_x & i_X \,.\, j_y & i_X \,.\, k_z \\ j_Y \,.\, i_x & j_Y \,.\, j_y & j_Y \,.\, k_z \\ k_W \,.\, i_x & k_W \,.\, j_y & k_W \,.\, k_z \end{bmatrix} \begin{Bmatrix} p_x \\ p_y \\ p_Z \end{Bmatrix}$$

$$= \begin{bmatrix} cos(\emptyset) & cos(90 + \emptyset) & cos(90) \\ cos(90 - \emptyset) & cos(\emptyset) & cos(90) \\ cos(90) & cos(90) & cos(0) \end{bmatrix} \begin{Bmatrix} p_x \\ p_y \\ p_Z \end{Bmatrix}$$

$$= \begin{bmatrix} cos(\emptyset) & -sin(\emptyset) & 0 \\ \sin(\emptyset) & cos(\emptyset) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} p_x \\ p_y \\ p_Z \end{Bmatrix}$$

From the above it is noticed that the mapping matrix $R(Z, \emptyset) = \begin{bmatrix} cos(\emptyset) & -sin(\emptyset) & 0 \\ sin(\emptyset) & cos(\emptyset) & 0 \\ 0 & 0 & 1 \end{bmatrix}$.

Similarly global coordinates of point *p* can be obtained for the given rotation ∅ degrees about the X-axis or Y-axis of the global frame relative to the local frame by the following Eqs.(3.3) & (3.4).

$$R(X, \emptyset) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\emptyset) & -sin(\emptyset) \\ 0 & \sin(\emptyset) & cos(\emptyset) \end{bmatrix} \tag{3.3}$$

$$R(Y,\emptyset) = \begin{bmatrix} cos(\emptyset) & 0 & sin(\emptyset) \\ 0 & 1 & 0 \\ -sin(\emptyset) & 0 & cos(\emptyset) \end{bmatrix} \quad (3.4)$$

Basic rotation matrices are to be multiplied sequentially to represent rotations about the principal axes of the OXYZ coordinate system. Since matrix multiplications are not commute, the order of performing rotations is important.

3.2. Homogeneous Transformation

A simple *3 x 3* rotation matrix does not give any translation and scaling, a fourth component is required to a position vector $p = (p_x, p_y, p_z)^T$ in a three-dimensional space which makes it $\hat{p} = (wp_x, wp_y, wp_z, w)^T$. Using a rotation matrix plus a vector directs the use of homogeneous coordinates. Representation of the homogeneous coordinates of a point vector is as follows:

$$p_x = {wp_x}/{w} \qquad p_y = {wp_y}/{w} \qquad p_z = {wp_z}/{w}$$

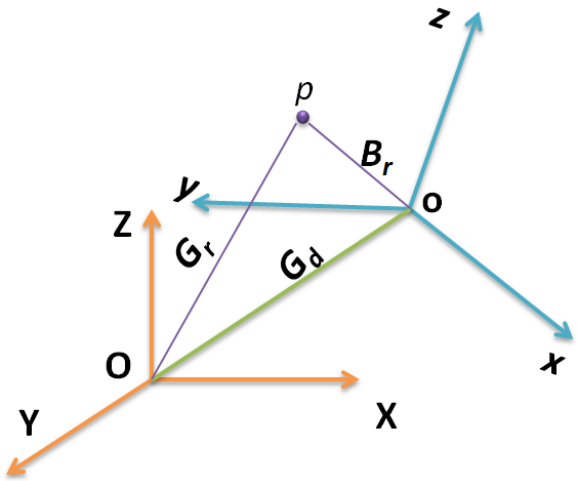


Fig.3.2 Point representation in coordinate frames B and G

Consider a rigid body with local frame *B* and global reference frame *G* as shown in Fig.3.2. A body point *p* can be represented as ${}^{B}r_p$ and ${}^{G}r_p$ in the frames *B* and *G* respectively. The vector G_d specifies the origin position of the body frame with respect to the global frame. Rigid body B (oxyz) motion in the global frame G (OXY Z) can be expressed as a combination of rotation ${}^{G}r_B$ as well as translation G_d.

$$G_{r_p} = G_{r_B} B_{r_p} + G_d \quad (3.5)$$

Where $G_{r_p} = \begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix}$, $B_{r_p} = \begin{Bmatrix} x \\ y \\ z \end{Bmatrix}$ and $G_d = \begin{Bmatrix} X_0 \\ Y_0 \\ Z_0 \end{Bmatrix}$

The homogeneous transformation matrix maps a position vector expressed in homogeneous coordinates from one coordinate system to another coordinate system and this transformation matrix is a *4x4* single matrix transformation which represents the rigid motion.

$$G_{r_p} = G_{T_B} B_{r_p} \tag{3.6}$$

Where $G_{T_B} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & X_0 \\ r_{21} & r_{22} & r_{23} & Y_0 \\ r_{31} & r_{32} & r_{33} & Z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} G_{r_B\,3x3} & G_{d\,3x1} \\ 0_{1x3} & 1_{1x1} \end{bmatrix}$

The top left *3 x 3* sub-matrix represents the rotation matrix; the top right *3 x 1* sub-matrix relates the position vector of the origin of the rotated coordinate system with respect to the global reference system; the lower left *1 x 3* sub-matrix gives the perspective transformation; and the fourth diagonal element is the global scaling factor. The homogeneous transformation matrix can be used to obtain the geometric liaison between the body attached frame oxyz and the global reference frame OXYZ.

3.3. Manipulator Kinematics

An articulated object is a set of rigid segments connected with joints; they are usually either rotational or translational. A minimal kinematic model is defined by its individual rigid segment lengths, joint degrees of freedom, their maximum and minimum joint limits, and a tree structured hierarchy of the segments and joints. Each joint maintains the rotations currently in effect at the corresponding joint. There are two major problems while developing robot arm kinematic models. The first problem brings up the direct/forward kinematics problem, while the second problem is the inverse kinematics/arm solution problem. Since the joint variables are independent variables in a robot arm, and a manipulator task is usually stated in terms of the reference coordinate frame, the inverse kinematics problem is used more commonly.

3.3.1. D-H notation [5]

Denavit and Hartenberg introduced a systematic and generalized approach to represent the spatial geometry of the robot arm links with respect to a fixed reference frame. This method uses a *4 x 4* homogeneous transformation matrix to describe the spatial relationship between two adjacent links that relates the spatial displacement of the hand coordinate frame to the reference coordinate frame.

The kinematic relations of robot components according to the D-H notation are as follows:

i. A robot arm with *n* joints will have *n+1* links. Start numbering the base link with '0' and increase sequentially up to '*n*' for the end-effector link.
ii. The z_i-axis is lined up with the *i+1* joint axis.
iii. The x_i-axis is labeled along the common normal for the z_{i-1} and z_i axes, pointing from the $z_{i-1}$to the z_i-axis.
iv. The y_i-axis is resolved by the right-hand rule, $y_i = z_i \times x_i$.
v. a_i is the kinematic link length and is the distance along the x_i-axis from z_{i-1} to z_i axes.
vi. α_i is the link twist and is measured the rotation about the x_i axis between z_{i-1}-axis and z_i-axis.
vii. d_i is the joint distance/link offset along the z_{i-1}-axis between $x_{i-1}$and x_i axes.
viii. Joint angle θ_i is the required rotation of x_{i-1}-axis about the z_{i-1}-axis to become parallel to the x_i -axis.

3.3.2. Forward kinematic model

After giving a thorough consideration of the preceding works in this field, a four degree of freedom multi-functional reprogrammable manipulator is chosen. This is a four axis articulate manipulator designed to move material like machine parts, tools, specialized devices, etc. Fig.3.3 shows the different degrees of freedom of the arm and its specification is illustrated in Table.3.1. It is driven by four servomotors and has a gripper as an end-effector.

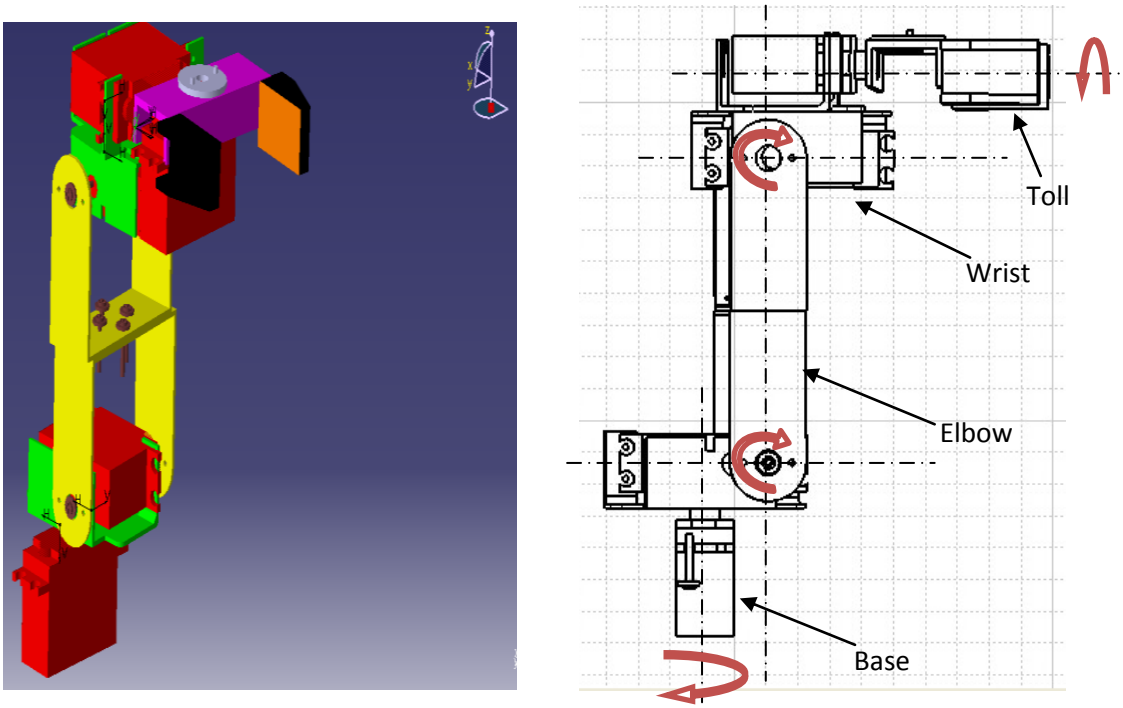


Fig.3.3 3D design of industrial arm with four degrees of freedom

Design's practical functions include:

Movement: The manipulator's workspace comprises of a 180 degree hemispherical envelop round itself throughout the arm's length as shown in Fig.3.4.

Manipulation: Servo motors are used for the movement of the arm.

Power Source: It is powered by batteries as it could be used in different environment. The manipulator can also be electrically powered when directly connected to the electric power supply with an AC/DC adaptor.

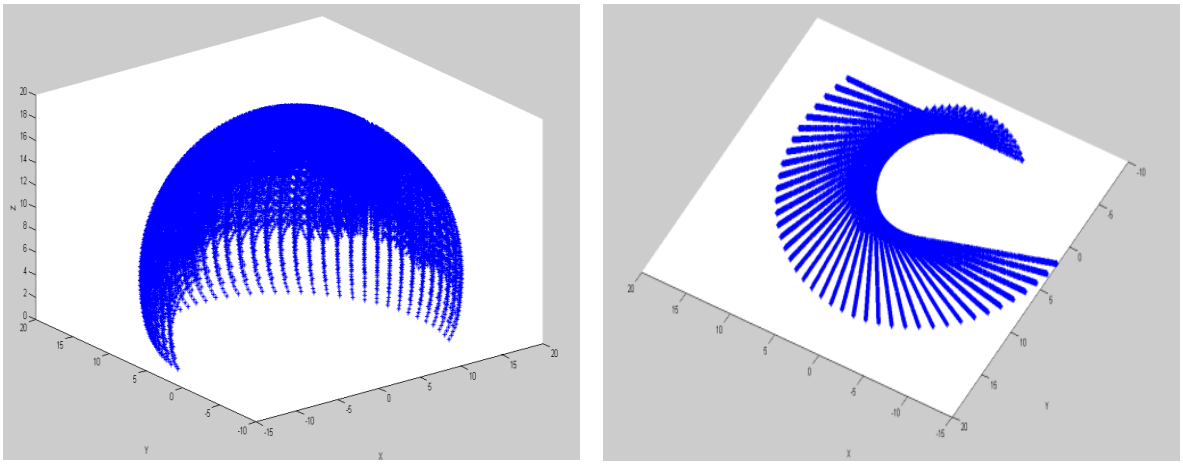


Fig.3.4 Manipulator work envelope 3D & 2D views [All dimensions are in cm.]

Table 3.1 Basic Specification of the Manipulator

Specification	**Value**	**Units**
Number of axes	4	
Horizontal reach	210	mm
Vertical reach	280	mm
Drives	5 PMDC servo motors	
Configuration	4 Axes plus gripper All axes completely independent All axes can be controlled simultaneously	
Work Envelope	Refer (Fig. 3.5) Body Rotation : 180 Elbow Rotation :- 180 Wrist Rotation : -60-120 Gripper Rotation : 90-270	degrees

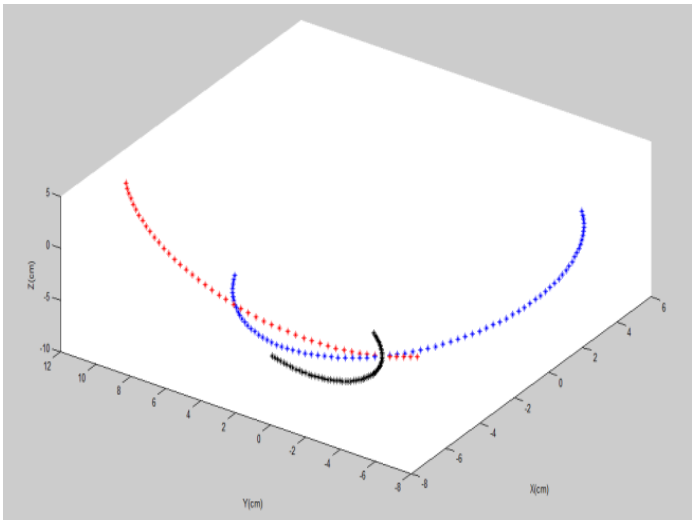


Fig.3.5 Variation of end-effector position when all joint angles are varied uniformly and other joints are at fixed angle [All dimensions are in cm.]

Kinematics of the manipulator deals with each moveable part of the robot by assigning it a frame of reference and since the manipulator has many parts, it has many individual frames. An analysis of the links at different position is methodically calculated.

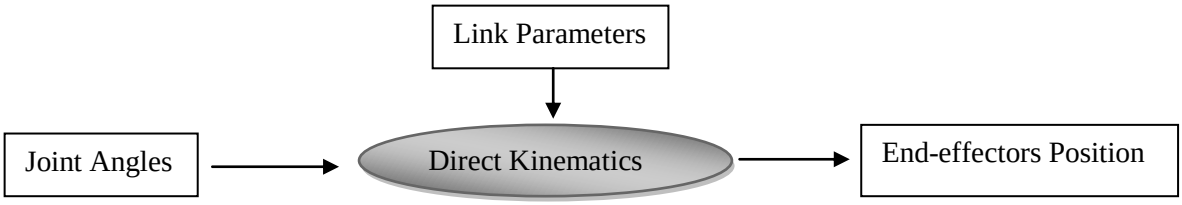


Fig.3.6 Schematic diagram of direct kinematics of a manipulator

Using D-H convention, coordinate frames for the manipulator are assigned as shown in the Fig.3.7.

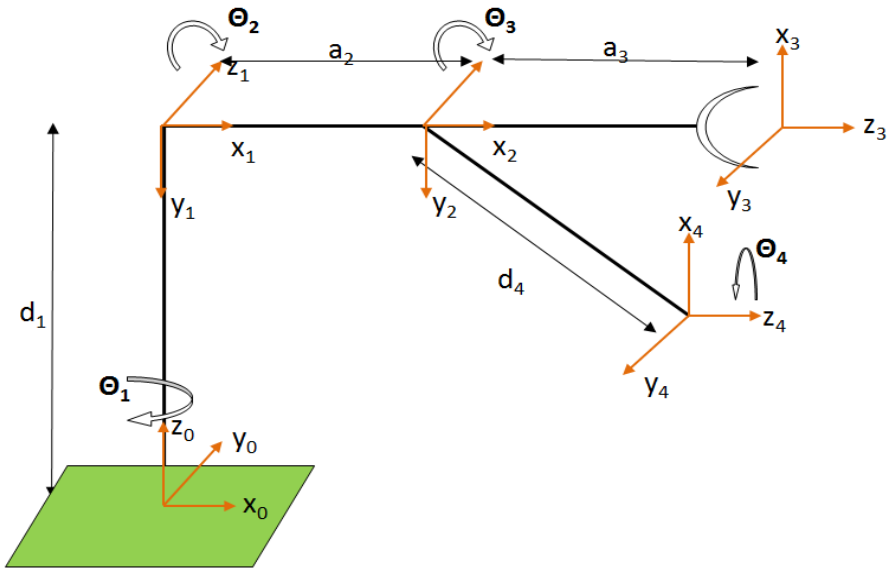


Fig.3.7 Link coordinate frame of the manipulator

The position and orientation of the end-effector in terms of given joint angles is calculated using a set of equations and this is forward kinematics. This set of equations is formed using DH parameters obtained from the link coordinate frame assignation. The parameters for the manipulator are listed in Table 3.2, where θ_i is the rotation about the Z-axis, α_i rotation about the X-axis, d_i transition along the Z-axis, and a_i translation along the X-axis.

Table 3.2 Kinematic parameters of the manipulator

Axis	***θ***	***d (mm)***	***a (mm)***	***α***
1	θ_1	$d_1 = 70$	0	$-\pi/2$
2	θ_2	0	$a_2 = 100$	0
3	θ_3	0	$a_3 = 70$	$-\pi/2$
4	θ_4	$d_2 = 45$	$a_4 = 0$	0

The set of link coordinates assigned using DH convention is then transformed from coordinate frame (N_i) to (N_{i-1}). Using a homogeneous coordinate transformation matrix, the relation between adjacent links is given in Eq.(3.7).

$$T_i = Rot(z, \theta_i) * Trans(0,0, d_i) * Trans(a_i, 0,0) * Rot(x, \alpha_i) \tag{3.7}$$

$$= \begin{bmatrix} C\theta_i & -S\theta_i C\alpha_i & S\theta_i S\alpha_i & a_i C\theta_i \\ S\theta_i & C\theta_i C\alpha_i & -C\theta_i S\alpha_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.8}$$

Here C_i=cos(θ_i), S_i=sin(θ_i)

On substituting the kinematic parameters illustrated in Table 3.2 into Eq. (3.8), individual transformation matrices T_0^1 to T_4^5 can be found and the global transformation matrix T_0^5 of the robot arm is fund according to the Eq. (3.9).

$$T_{base}^{tool} = T_{base}^{wrist} * T_{wrist}^{tool} \tag{3.9}$$

Where $T_{base}^{wrist} = T_0^1 * T_1^2$

$$T_{wrist}^{tool} = T_2^3 * T_3^4$$

$$T_{base}^{tool} = \begin{bmatrix} m_x & n_x & o_x & p_x \\ m_y & n_y & o_y & p_y \\ m_z & n_z & o_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R(\theta)_{3x3} & P_{3x1} \\ 0 & 1 \end{bmatrix} \quad (3.10)$$

Where (p_x, p_y, p_z) represents the position and $R(\theta)_{3x3}$represents the rotation matrix of the end effector. From this transformation matrix, the position (translation) of end-effector with reference to base frame as a function of the joint angles is depicted in Fig.3.5.

Tool configuration is six-dimensional because arbitrary specified by three position co-ordinates (x, y, z) and orientation co-ordinates (yaw, pitch, roll).

$$\text{Tool position} = \begin{Bmatrix} p_x \\ p_y \\ p_z \end{Bmatrix} = \begin{Bmatrix} c_1(a_2c_2 + a_3c_{23} - d_4s_{23}) \\ s_1(a_2c_2 + a_3c_{23} - d_4s_{23}) \\ d_1 - a_2s_2 - a_3s_{23} - d_4c_{23} \end{Bmatrix} \quad (3.11)$$

Tool orientation coordinates can be defined by its approach vector (tool roll angle θ_4).

$$\text{Orientation coordinates} = \begin{Bmatrix} y \\ p \\ r \end{Bmatrix} = \begin{Bmatrix} -[exp(\theta_4)/\pi]c_1s_{23} \\ -[exp(\theta_4)/\pi]s_1s_{23} \\ -[exp(\theta_4)/\pi]c_{23} \end{Bmatrix} \quad (3.12)$$

Therefore the final arm equation for the considered 4-axis manipulator is

$$X = \begin{Bmatrix} p_x \\ p_y \\ p_z \\ y \\ p \\ r \end{Bmatrix} = \begin{Bmatrix} c_1(a_2c_2 + a_3c_{23} - d_4s_{23}) \\ s_1(a_2c_2 + a_3c_{23} - d_4s_{23}) \\ d_1 - a_2s_2 - a_3s_{23} - d_4c_{23} \\ -[exp(\theta_4)/\pi]c_1s_{23} \\ -[exp(\theta_4)/\pi]s_1s_{23} \\ -[exp(\theta_4)/\pi]c_{23} \end{Bmatrix} \quad (3.13)$$

3.3.3. Inverse kinematic model

Since the independent variables in a robotic arm are the joint variables and a task is usually in terms of reference coordinate frame, inverse kinematics is used more frequently. In general, the inverse kinematics problem can be solved by various techniques such as matrix algebraic, iterative and geometric approaches.

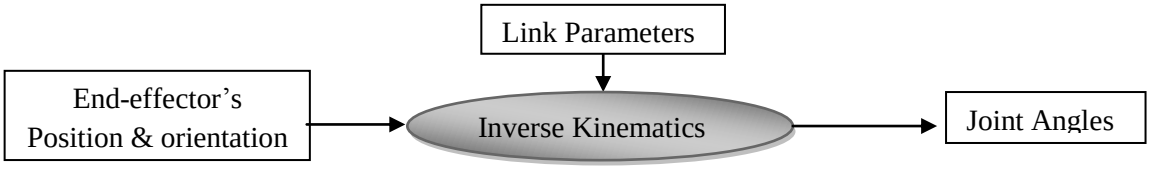


Fig.3.8 Schematic diagram of inverse kinematics of a manipulator

This section describes the development of inverse kinematic models of an arm based on its link coordinate systems.

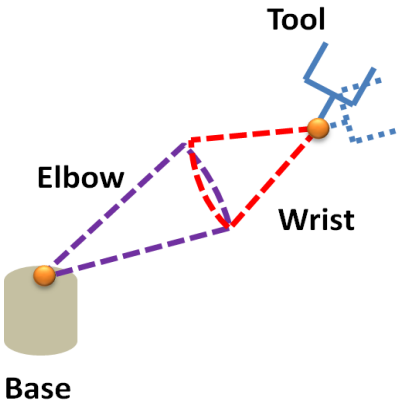


Fig.3.9 Elbow position & Wrist position for the same end-effector position

By observing the Eq. (3.16), there is a possibility of getting two wrist angles ($\pm\theta_3$) for the same tool position. Since the elbow angle (θ_2) depends on wrist angle (θ_3), there will be two elbow angles corresponds to each wrist angle as shown in the Fig.3.9.

Since there is no yaw motion, the base angle can be found easily by Eq. (3.14).

$$\text{Base angle } \theta_1 = arctan\left({}^{p_y}/_{p_x}\right) \tag{3.14}$$

Where p_x *and* p_y can be found from eq(3.13) and from the arm equation, the global pitch angle θ_{23} can be found as follows:

$$\theta_{23} = arctan\left({}^{(c_1 y + s_1 p)}/_{r}\right) \tag{3.15}$$

The wrist angle can be found as follows:

$$\theta_3 = \pm arccos\left({}^{(\|b\|^2 - {a_2}^2 - {a_3}^2)}/_{p_x}\right) \tag{3.16}$$

Where $\|b\|^2 = {b_1}^2 + {b_2}^2$; *and* $b_1 = c_1 p_x + c_2 p_y + d_4 s_{23}$ & $b_2 = d_1 - d_4 c_{23} - p_z$

Once θ_3 is known then elbow angle θ_2 can be found from the global pitch angle θ_{23}.

$$\because \theta_{23} = \theta_3 + \theta_2 \Rightarrow \theta_2 = \theta_{23} - \theta_3 \tag{3.17}$$

The final joint parameter θ_4 can be found from the arm Eq. (3.18) as follows

$$\text{Tool roll angle } \theta_4 = \pi * ln\sqrt{(y^2 + p^2 + r^2)} \tag{3.18}$$

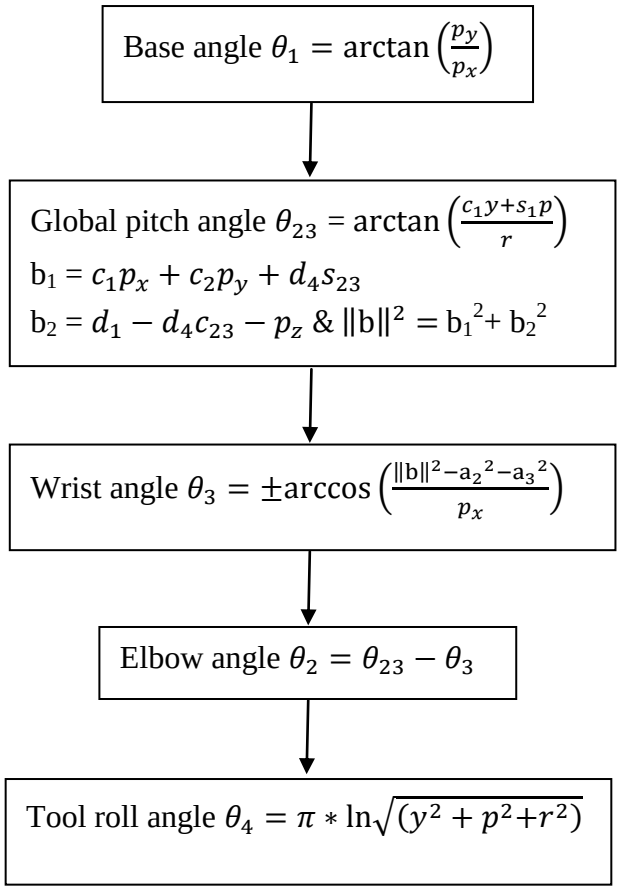


Fig.3.10 Flow chart for inverse kinematics of a 4-axis articulated robot

3.4. Summary

The purpose of the current section is to understand the motion of the robot arm structure. The motion of the structure is analyzed in two ways, forward and inverse kinematics. Forward kinematics is the determination of the every link configuration, specially the end-effector, when the joint variables are given. Inverse kinematics deals with determining the joint variables of a robot manipulator for the given position and orientation of the end-effector.

The standard Denavit-Hartenberg convention is implemented for determining coordinate frames attached to each robot's link. Based on the D-H rule, each transformation matrix can be expressed for the adjacent coordinate frames by four kinematic parameters; link length, link twist, joint distance, and joint angle. Later, arm equation is developed for the considered 4-axis manipulator using forward kinematic models. From the developed arm equation, inverse kinematic models have been developed to find out the joint parameters.

Chapter 4

MECHANICAL DESIGN ARCHITECTURE OF THE MOBILE PLATFORM

4. MECHANICAL DESIGN ARCHITECTURE OF THE MOBILE PLATFORM

A mobile manipulator is a robotic arm mounted on mobile platform. This combination permits manipulation tasks over unlimited workspace. However, since the platform and the manipulator are with independent degree of freedom, the system can reach to a specific position in the workspace in multiple configurations, resulting in a system with redundancy. The mechanical system considered in this research work is a non-holonomic wheeled mobile platform.

4.1. Mobile Platform Position Representation

A mobile platform is equipped with definite number of wheels which is capable of an autonomous motion. In order to achieve the desired motion by the mobile platform, each of the wheels is fitted with electric motors. Throughout this analysis we model the mobile platform as a rigid body on wheels, operating on a horizontal plane. The total dimensionality of this system chassis on the plane is three, two for position in the plane and one for orientation along the vertical axis, which is orthogonal to the plane.

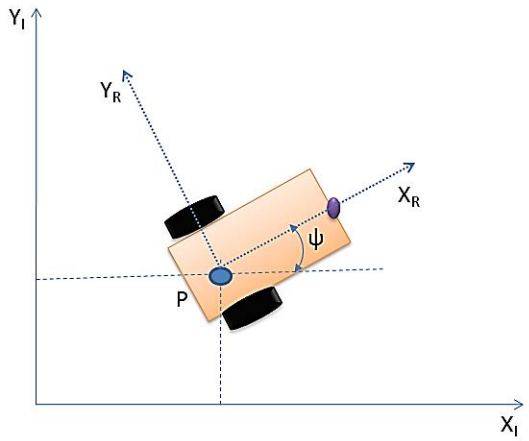


Fig.4.1 The global reference plane and the mobile platform local reference frame

Let the robot is moving on a plane, then the motion of the mobile platform can be depicted as follows. In order to specify the position of the mobile platform on the plane, it is required to establish a relationship between the global reference frame of the plane and the local reference frame of the mobile platform, as in Fig.4.1. An arbitrary inertial base frame $O{:}\{X_I, Y_I\}$ is fixed on the plane of motion, while a mobile reference frame $\mathrm{M}{:}\{X_R, Y_R\}$ is attached to the mobile robot. The position of P in the global reference frame is specified by coordinates x and y, and the angular difference between the global and local reference frames is given by ψ.

Therefore the mobile platform position:

$$\{\vec{X}\} = [x \quad y \quad \psi]^T \tag{4.1}$$

And mapping is accomplished using the orthogonal rotation matrix:

$$[R(\psi)] = \begin{bmatrix} cos\psi & sin\psi & 0 \\ -sin\psi & cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.2}$$

From Eq.(4.2) the mobile platform's motion can be computed in the global reference frame with respect to its local reference frame:

$$\{\vec{X}_m\} = [R(\psi)]\{\vec{X}\} \text{ and } \{\dot{\vec{X}}_m\} = [R(\psi)]\{\dot{\vec{X}}\} \tag{4.3}$$

$$\{\dot{\vec{X}}\} = [R(\psi)^{-1}]\{\vec{X}_m\} \tag{4.4}$$

4.2. Kinematic Constraints of Various Wheel Configurations

While a mobile platform is in movement, two feasible constraints are existing for every wheel. The first constraint enforces the concept of rolling contact as represented in Fig.4.2; wheel must roll when motion takes place in the appropriate direction. The second constraint enforces the concept of no lateral slippage. It means the wheel must not slide orthogonal to the wheel plane as shown in Fig.4.3. The first step to a kinematic model of the mobile platform is to express constraints on the motions of individual wheels. The motions of individual wheels can be combined later to compute the motion of the mobile platform as a whole.

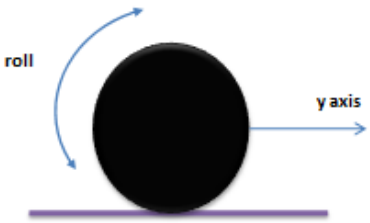


Fig.4.2 Rolling motion

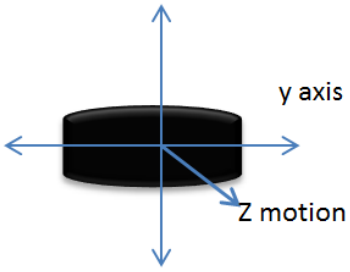


Fig.4.3 Lateral slip

The following assumptions are taken into consideration while performing the kinematic analysis:

a. Platform movement is on a horizontal plane.
b. Equipped wheels are not deformable.
c. Wheels are connected to the rigid frame (chassis).
d. Attached wheels are always remaining in vertical during the motion of mobile platform.
e. The contact between wheel and ground is a single point.

f. There is no friction for rotation around contact point.

g. There is no sliding at the single point of contact between the ground plane and the wheel. It means that the wheel is in motion under only pure rolling conditions and the rotation is about the vertical axis through the contact point.

In the present investigation the following four basic wheel types are considered:

- Fixed standard wheel
- Steered standard wheel
- Swedish wheel
- Spherical wheels
- Castor wheel

4.2.1. Fixed standard wheel

The fixed standard wheel [30] has no vertical axis of rotation for steering. Its angle to the chassis is thus fixed, and it is limited to motion back and forth along the wheel plane and rotation around its contact point with the ground plane. Fig.4.4 depicts a fixed standard wheel and indicates its position pose relative to the mobile platform's local reference frame. The position of the wheel is expressed in polar coordinates by distance l and angle α. The angle of the wheel plane relative to the chassis is denoted by *β*. The wheel, which has radius *r*, can spin over time, and so its rotational position around its horizontal axle is a function of time t: *φ(t)*.

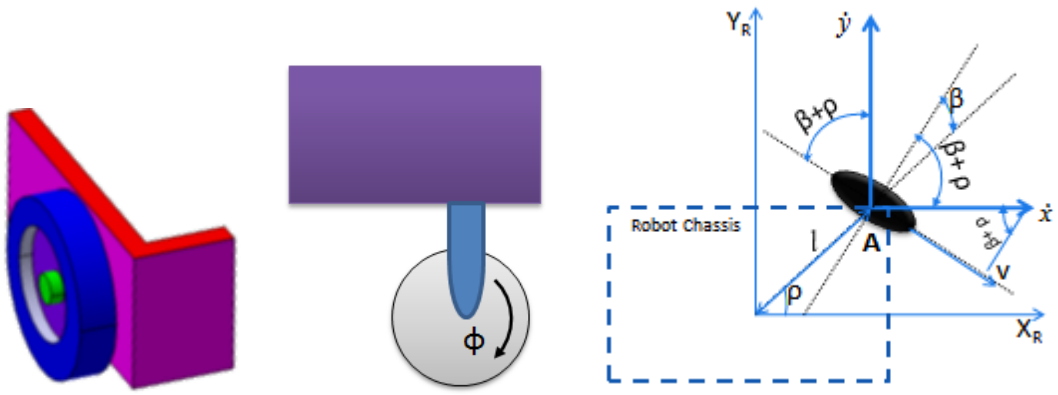


Fig.4.4 Fixed standard wheel and its parameters [30]

The rolling constraint for this wheel enforces that complete motion along the direction of the wheel plane must be accompanied by the appropriate amount of wheel spin so that there is pure rolling at the contact point:

$$[sin(\rho+\beta) \quad -cos(\rho+\beta) \quad (-l)cos\beta] * R(\psi)\dot{\xi}_I - r\dot{\varphi} = 0 \qquad (4.5)$$

The sliding constraint for this wheel enforces that the component of the wheel's motion orthogonal to the wheel plane must be zero:

$$[cos(\rho+\beta) \quad sin(\rho+\beta) \quad l\,sin\beta]R(\psi)\dot{\xi}_I = 0 \qquad (4.6)$$

4.2.2. Steered standard wheel

The steered standard wheel [30] differs from the fixed standard wheel only when there is an additional degree of freedom: the wheel may rotate around a vertical axis passing through the center of the wheel and the ground contact point. The orientation of the wheel to the mobile platform chassis is no longer a single fixed value β, but instead varies as a function of time: β(t) .

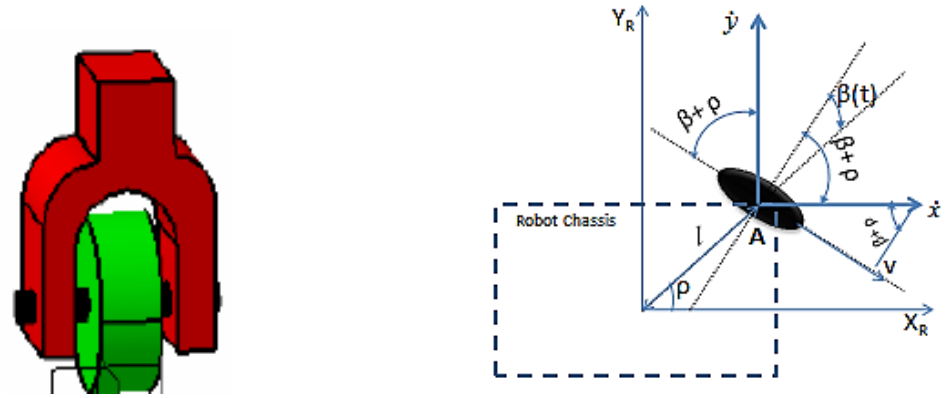


Fig.4.5 Steerable standard wheel and its parameters [30]

The rolling and sliding constraints for the steered standard wheel shown in Fig.4.5:

$$[sin(\rho+\beta) \quad -cos(\rho+\beta) \quad (-l)cos\beta]*R(\psi)\dot{\xi}_I - r\dot{\varphi} = 0 \qquad (4.7)$$

$$[cos(\rho+\beta) \quad sin(\rho+\beta) \quad lsin\beta]R(\psi)\dot{\xi}_I = 0 \qquad (4.8)$$

4.2.3. Swedish wheel

Swedish wheels [30] have no vertical axis of rotation, yet are able to move omnidirectionally like the castor wheel. This is possible by adding a degree of freedom to the fixed standard wheel. Swedish wheels consist of a fixed standard wheel with rollers attached to the wheel perimeter with axes that are antiparallel to the main axis of the fixed wheel component. The exact angle γ between the roller axes and the main axis can vary, as shown in Fig.4.6.

The motion constraint that is derived looks identical to the formula is modified by adding γ such that the effective direction along which the rolling constraint holds is along this zero component rather than along the wheel plane:

$$\left[sin(\rho+\beta+\gamma) \quad -cos(\rho+\beta+\gamma) \quad (-l)\,cos(\beta+\gamma)\right] * R(\psi)\dot{\xi}_I - r\dot{\varphi} = 0 \qquad (4.9)$$

Orthogonal to this direction the motion is not constrained because of the free rotation $\dot{\varphi}_{sw}$ of the small rollers.

$$[sin(\rho+\beta+\gamma) \quad cos(\rho+\beta+\gamma) \quad (l)sin(\beta+\gamma)] * R(\psi)\dot{\xi}_I - r\dot{\varphi}sin\gamma - r_{sw}\dot{\varphi}_{sw} = 0 \tag{4.10}$$

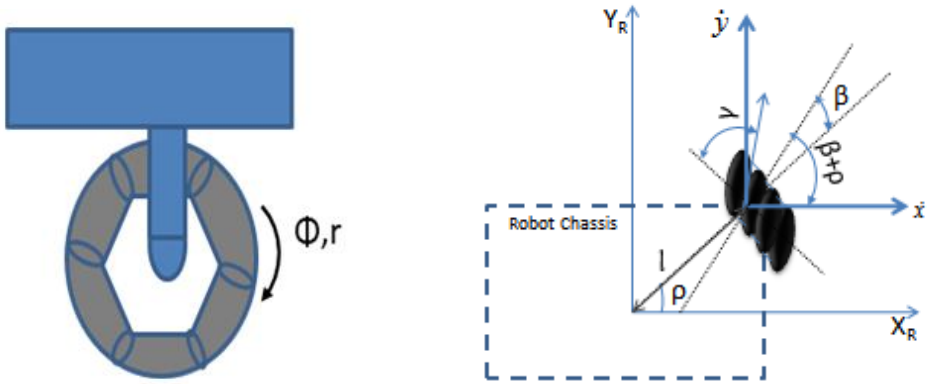


Fig.4.6 Swedish wheel and its parameters [30]

Consider $\gamma = 0$, this represents the Swedish 90-degree wheel. In this case, the zero component of velocity is in line with the wheel plane and so Eq. (4.9) reduces exactly to Eq. (4.5), the fixed standard wheel rolling constraint. But because of the rollers, there is no sliding constraint orthogonal to the wheel plane. At the other extreme $\gamma = \frac{\pi}{2}$, the rollers have axes of rotation that are parallel to the main wheel axis of rotation. For $\gamma = \frac{\pi}{2}$ in Eq. (4.9), result the fixed standard wheel sliding constraint, Eq. (4.6).

4.2.4. Spherical wheel

A ball or spherical wheel [30], places no direct constraints on motion (Fig.4.7). Such a mechanism has no principal axis of rotation, and therefore no appropriate rolling or sliding constraints exist.

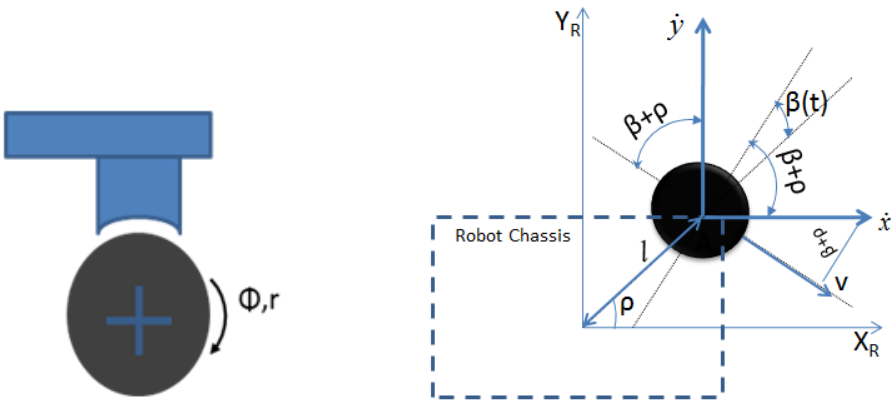


Fig.4.7 Spherical wheel and its parameters [30]

Therefore Eq. (4.11) simply describes the roll rate of the ball in the direction of motion v_A of point A of the mobile platform.

$$[sin(\rho+\beta) - cos(\rho+\beta)(-l)cos\,\beta]R(\psi)\dot{\xi}_I - r\dot{\varphi} = 0 \quad (4.11)$$

By definition the wheel rotation orthogonal to this direction is zero.

$$[sin(\rho+\beta)cos(\rho+\beta)l\,sin\beta]R(\psi)\dot{\xi}_I = 0 \quad (4.12)$$

4.2.5. Castor wheel

Castor wheels [30] are able to steer around a vertical axis. However, unlike the steered standard wheel, the vertical axis of rotation in a castor wheel does not pass through the ground contact point. Fig.4.8 depicts a castor wheel, demonstrating that formal specification of the castor wheel's position requires an additional parameter which is a rigid rod of fixed length connected to wheel.

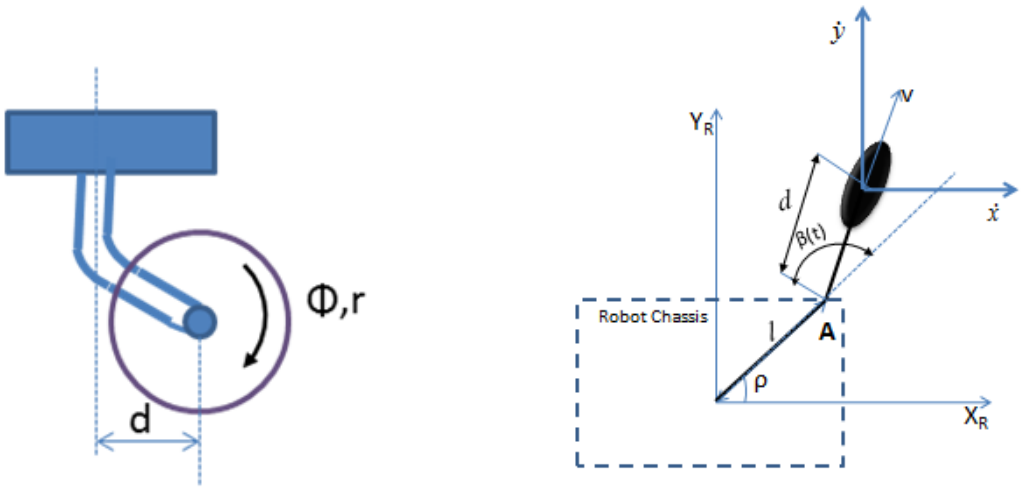


Fig.4.8 Castor wheel and its parameters [30]

For the caster wheel, the rolling constraint is identical to equation because the offset axis plays no role during motion that is aligned with the wheel plane:

$$[sin(\rho+\beta) - cos(\rho+\beta)(-l)cos\beta]R(\psi)\dot{\xi}_I - r\dot{\varphi} = 0 \quad (4.13)$$

Because of the offset ground contact point relative to A, the constraint with zero lateral movement would be wrong. Instead, the constraint is much like a rolling constraint with appropriate rotation of the vertical axis as follows:

$$[cos(\rho+\beta)\,sin(\rho+\beta)d + l\,sin\beta]R(\psi)\dot{\xi}_I + d\dot{\beta} = 0 \quad (4.14)$$

4.3.Kinematic Constraints of a Mobile Platform

Consider a general mobile platform, equipped with ***N*** wheels of the five above described categories. The five following subscripts to identify quantities relative to these classes: *f* for fixed wheels, *s* for steerable standard wheels, *sw* for Swedish wheels, *sp* for spherical wheel

and *c* for castor wheels. The numbers of wheels for each type are denoted N_f, N_s, N_c, N_{sw}, N_{sp} with $N_f + N_s + N_c + N_{sw} + N_{sp} = N$. The configuration of the mobile platform is fully described by the following vectors of coordinates

- Posture coordinates: $\xi_I = [x(t) \quad y(t) \quad \psi(t)]^T$
- Angular coordinates: β_f, $\beta_s(t)$, $\beta_c(t)$, $\beta_{sw}(t)$, and $\beta_{sp}(t)$ for the five types of wheels respectively.
- Rotational coordinates: $[\varphi_f(t) \quad \varphi_s(t) \quad \varphi_c(t) \quad \varphi_{sw}(t) \quad \varphi_{sp}(t)]^T$ for the rotation angles of the wheels around their horizontal axis of rotation.

The rolling constraints of all wheels can now be collected in a single expression:

$$J_1(\beta)R(\psi)\dot{\xi}_I - J_2\dot{\varphi} = 0 \tag{4.15}$$

This expression bears a strong resemblance to the rolling constraint of a single wheel, but substitutes matrices in lieu of single values, thus taking into account all wheels. J_2 is a constant diagonal matrix $N \times N$ whose entries are radii *r* of all standard wheels. $J_1(\beta)$ denotes a matrix with projections for all wheels to their motions along their individual wheel planes:

$$J_1(\beta) = \begin{bmatrix} J_{1f} \\ J_{1s}(\beta_s) \\ J_{1c}(\beta_c) \\ J_{1sw} \\ J_{1sp} \end{bmatrix} \tag{4.16}$$

where J_{1f} , J_{1s} , J_{1c} , J_{1sw} , and J_{1sp} are the matrices of $(N_f \times 3)$, $(N_s \times 3)$, $(N_c \times 3)$, $(N_{sw} \times 3)$ and $(N_{sp} \times 3)$, whose forms derive readily from the constraints (4.5), (4.7), (4.9), (4.11) and (4.13). J_2 is a constant $(N \times N)$ matrix whose diagonal entries are the radii of the wheels, except for the radii of the Swedish wheels which are multiplied by $\cos\gamma$.

We use the same technique to collect the sliding constraints of all standard wheels into a single expression with the same structure as Eqs. (4.15) and (4.16):

$$C_1(\beta)R(\theta)\dot{\xi}_I + C_2\dot{\beta}_S = 0 \tag{4.17}$$

Where $C_1(\beta) = \begin{bmatrix} C_{1f} \\ C_{1s}(\beta_s) \\ C_{1c}(\beta_c) \\ C_{1sw} \\ C_{1sp} \end{bmatrix}$ and $C_2 = \begin{bmatrix} 0 \\ 0 \\ C_{2c} \\ 0 \\ 0 \end{bmatrix}$

The terms C_{1f} , C_{1s} , C_{1c} , C_{1sw} , and C_{1sp} are the matrices of $(N_f \times 3)$, $(N_s \times 3)$, $(N_c \times 3)$, $(N_{sw} \times 3)$ and $(N_{sp} \times 3)$, whose forms derive readily from the constraints (4.6), (4.8), (4.10), (4.12) and (4.14). C_2 is a constant $(N \times N)$ matrix whose diagonal entries are equal to *d* for N_c of the castor wheels.

4.4. Maneuverability of a Mobile Platform

The kinematic mobility of a mobile platform chassis is its ability to directly move in the environment. The basic constraint limiting mobility is the rule that every wheel must satisfy its sliding constraint. In addition to instantaneous kinematic motion, a mobile platform is able to further manipulate its position, over time, by steering steerable wheels. The overall maneuverability of a mobile platform is thus a combination of the mobility available based on the kinematic sliding constraints of the standard wheels, plus the additional freedom contributed by steering and spinning the steerable standard wheels.

4.4.1. Degree of mobility

It is observed from the wheel kinematic constraints (Eqs. (4.9), (4.11), and (4.13)), that the Swedish wheel, spherical wheel and castor wheel impose no kinematic constraints on the mobile platform chassis because these wheels can range freely owing to the internal wheel degrees of freedom. Therefore only fixed standard wheels and steerable standard wheels have impact on mobile platform chassis kinematics and therefore require consideration when computing the mobile platform's kinematic constraints.

Consider now the $(N_f + N_S)$ wheels of fixed and steered standard wheels. To avoid any lateral slip the motion vector $R(\psi)\dot{\xi}_I$ has to satisfy the following constraints:

$$C_{1f}R(\psi)\dot{\xi}_I = 0 \tag{4.18}$$

$$C_1(\beta_s)R(\psi)\dot{\xi}_I = 0 \tag{4.19}$$

where $C_1(\beta_s)=\begin{bmatrix} C_{1f} \\ C_{1s}(\beta_s) \end{bmatrix}$

Mathematically it represents $R(\psi)\dot{\xi}_I$ must belong to the null space of the projection matrix. Null space of $C_1(\beta_s)$ is the space *N* such that for any vector *n* in *N*

$$C_1(\beta_s).n = 0 \tag{4.20}$$

Mobile platform chassis kinematics is therefore a function of the set of *independent* constraints arising from all standard wheels. The mathematical interpretation of independence is related to the *rank* of a matrix. Therefore $rank[C_{1s}(\beta_s)]$ is the number of independent constraints. In general, a mobile platform will have zero or more fixed standard wheels and zero or more steerable standard wheels. Therefore the possible range of rank values for any mobile platform is $0 \leq [C_{1s}(\beta_s)] \leq 3$.

Now we are ready to formally define a mobile platform's *degree of mobility* δ_m:

$$\delta_m = dimN[C_1(\beta_s)] = 3 - rank[C_1(\beta_s)] \tag{4.21}$$

It means for a completely constrained mobile platform $rank[C_1(\beta_s)] = 3$ and its motion in the plane is totally impossible. Fig.4.9 represents the mobile platform is completely constrained when it is equipped with three fixed wheels.

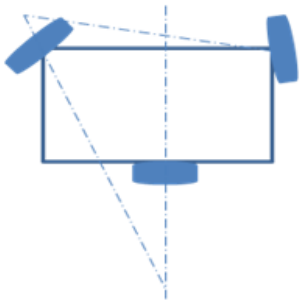


Fig.4.9 Constrained mobile platform

For this configured mobile platform the matrix $C_1(\beta_s)$ retains three independent constraints and has a rank of three. Let the specifications for the mobile platform shown in Fig.4.9 as $l_1 = l_2 = l_3 = 1,\ \beta_1 = \beta_2 = \beta_2 = {}^{\pi}/_{2}$, and $\rho_1 = \pi, \rho_2 = {}^{\pi}/_{3}, \rho_3 = {}^{-\pi}/_{4}$

$$C_1(\beta_s) = \mathrm{C_{1f}} = \begin{bmatrix} \cos\frac{3\pi}{2} & \sin\frac{3\pi}{2} & 1 \\ \cos\frac{5\pi}{6} & \sin\frac{5\pi}{6} & 1 \\ \cos\frac{\pi}{2} & \sin\frac{\pi}{2} & 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 1 \\ \frac{-\sqrt{3}}{2} & \frac{1}{2} & 1 \\ 0 & 1 & 1 \end{bmatrix} \tag{4.22}$$

By observing equation (4.22), it is obvious that the$rank[C_1(\beta_s)] = 3$ and its degree of mobility (δ_m) is 3- $rank[C_1(\beta_s)]$=0

Consider a mobile platform with two fixed standard wheels having specifications $l_1 = l_2, \beta_1 = \beta_2 = 0,$ and $\rho_2 = \rho_1 + \pi$. The configuration is like a differential-drive mobile platform as shown in Fig.4.10.

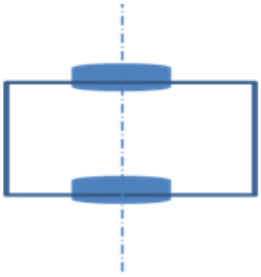


Fig.4.10 Differential-drive mobile platform

Then the matrix $C_1(\beta_s)$ has two constraints but rank one. Therefore the degree of mobility $\delta_m = 3 - rank[C_1(\beta_s)]$=2

$$C_1(\beta_s) = \mathrm{C_{1f}} = \begin{bmatrix} \cos(\rho_1) & \sin(\rho_1) & 0 \\ \cos(\rho_1 + \pi) & \sin(\rho_1 + \pi) & 0 \end{bmatrix} = \begin{bmatrix} \cos(\rho_1) & \sin(\rho_1) & 0 \\ -\cos(\rho_1) & -\sin(\rho_1) & 0 \end{bmatrix} \tag{4.23}$$

4.4.2. Degree of steerability

Platform can be equipped with the number of centered orientable wheels in order to steer the mobile platform. This impact of steering is indirect since the mobile platform must move for the change in steering angle of the steerable standard wheel.

The degree of steerability is defined:

$$\delta_s = rank\ [C_{1s}(\beta_s)] \tag{4.24}$$

An increase in the rank of $[C_{1s}(\beta_s)]$ implies more degrees of steering freedom and thus greater eventual maneuverability. The range of δ_s is given by $0 \leq \delta_s \leq 2$. Since $[C_1(\beta_s)]$includes$[C_{1s}(\beta_s)]$, a steerable standard wheel can both decrease mobility and increase steerability:

- Its particular orientation at any instant imposes a kinematic constraint.
- Its ability to change the orientation can lead to additional trajectories.

A differential drive mobile platform represented in Fig.4.11 has no centered orientable wheels. Therefore its Degree of steerability is $\delta_s = 0$.

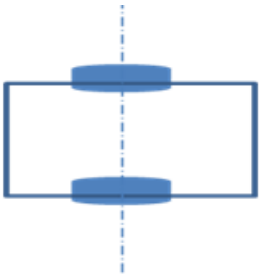


Fig.4.11 No centered orientable wheels

4.4.3. Maneuverability of wheeled platform

The overall (Degrees of Freedom) DOF that a mobile platform can manipulate is called the degree of maneuverability(δ_M). It can be defined in terms of mobility and steerability. Thus the maneuverability is the degrees of freedom that the mobile platform manipulates directly through wheel velocity and the degrees of freedom that it indirectly manipulates by changing the steering configuration.

$$\therefore \delta_M = \delta_m + \delta_s \tag{4.25}$$

4.4.4. Maneuverability of various configured wheeled platforms

Consider various wheeled platform configurations as shown in Fig.4.12 having three or four wheels with arbitrary orientations with respect to its wheel plane. The maneuverability of these platforms are illustrated in Table.4.1.

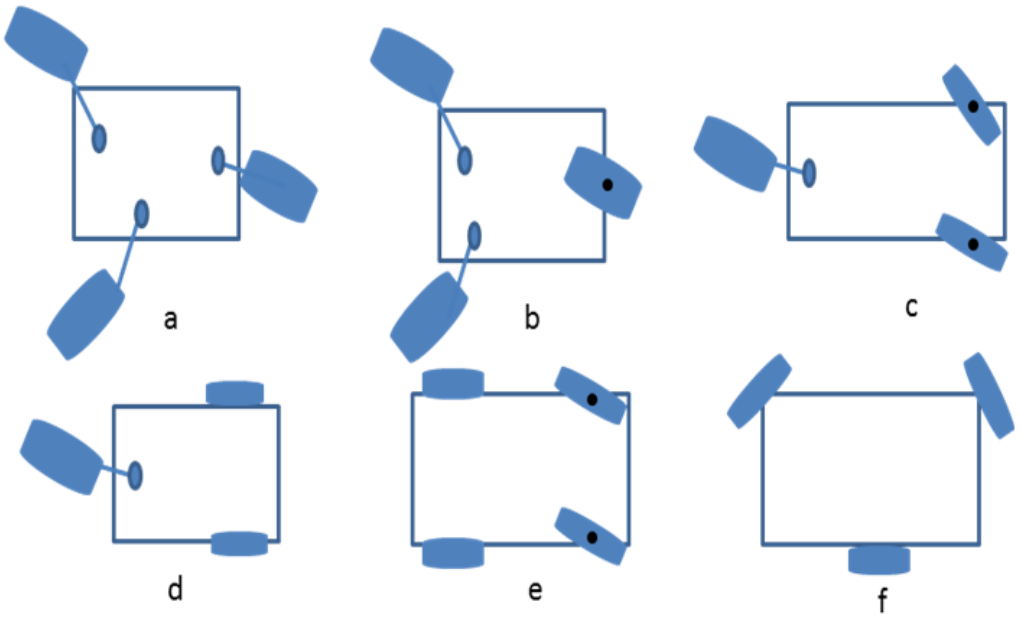


Fig.4.12 WMRs with various configured wheels

Table.4.1 Maneuverability of various configured wheels for Fig.23

Figure 4.12	$\boldsymbol{\delta_m}$	$\boldsymbol{\delta_s}$	$\boldsymbol{\delta_M}$
a (neither fixed wheels nor centered orientable wheels)	3	0	***3***
b (one steerable standard wheel & two castor wheels)	2	1	***3***
c (two independent steerable standard & one castor wheels)	1	2	***3***
d (two dependent fixed & one castor wheels)	2	0	***2***
e (two dependent fixed & centered orientable wheels)	1	1	***2***
f (three fixed wheels)	0	0	***0***

4.5.Velocity Equations of Differential Drive Wheeled Platform

For a differential robot, linear and angular velocities are depends on its wheel (left and right) velocities by following relation:

$$\left.\begin{aligned}\omega &= \frac{v_{Rt}-v_{Lt}}{2*s}\\ v &= \frac{v_{Rt}+v_{Lt}}{2}\end{aligned}\right\} \tag{4.26}$$

Where v_{Rt}& $v_{Lt}$are the right and left wheel velocities respectively and '$2s$' is the wheel base for the considered differential robot. The wheel velocities are bounded to some value v_b, which depends on the maximum possible linear velocity of the robot. The considered differential robot can move in the curved paths with the curvature:

$$\gamma = \frac{v_{Rt}-v_{Lt}}{s*(v_{Rt}+v_{Lt})} \quad where \quad \frac{-1}{s} \leq \gamma \geq \frac{1}{s} \quad and \quad 0 \leq v_{Rt}, v_{Lt} \geq v_b \tag{4.27}$$

Sharp turns should be avoided in practice so the constraints are applied to curvature and wheel velocities which will avoid the overstress on the robot. If the both wheels are moving with same velocity, $\gamma = 0$ and robot takes straight path. If $v_{Rt} = -v_{Lt}$, robot takes turn with large curvature or it takes sharp turns.

From Eqs.(4.26) & (4.27) the robot velocity coordinates are:

$$\dot{\xi}_I = \begin{Bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{Bmatrix} = \begin{bmatrix} cos\psi & 0 \\ sin\psi & 0 \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} 1/2 & 1/2 \\ -1/2s & 1/2s \end{bmatrix} * \begin{Bmatrix} v_{Rt} \\ v_{Lt} \end{Bmatrix} = \frac{1}{2s} \begin{bmatrix} s * cos\psi & s * cos\psi \\ s * sin\psi & s * sin\psi \\ -1 & 1 \end{bmatrix} * \begin{Bmatrix} v_{Rt} \\ v_{Lt} \end{Bmatrix} \quad (4.28)$$

4.6.Summary

In this section, the behavior of wheeled mobile platforms has been analyzed. These robots ride on a system of wheels and axles, some of which may be steerable or driven. For these platforms, there are many wheels and axle configuration have been used. The ultimate objective of this chapter is to investigate the complete description of the control theory of wheeled mobile robots and its maneuverability. Equations are modeled to describe the rigid body motions that arise from rolling trajectories based on the geometrical constraints of these wheels. Finally this analysis is applied to various three/four wheeled mobile robots. Moreover, this section detailed about the differential drive wheeled platform and its motion equations in terms of wheel velocities.

Chapter 5

MECHANICAL STRUCTURE OF THE MOBILE MANIPULATOR

Velocity Jacobean of Manipulator

Velocity Jacobean of Differential Platform

Velocity Jacobean of WMM

Summary

5. MECHANICAL STRUCTURE OF THE MOBILE MANIPULATOR

The important feature of a mobile manipulator is the flexible operational workspace in contrast with the bounded workspace of a fixed manipulator which is bolted to the base. This property confers a mobile manipulator with the ability of performing many tasks. Among the properties possessed by such a system, redundancy is one of the most important factor. This property enables one to use the redundant degrees of freedom to accomplish secondary tasks. In this way, coordinated motion of the manipulator and mobile platform leads to a wide range of redundancy which is a powerful tool. Due to the velocity constraints imposed on the mobile base as discussed in Section - 4, the wheeled mobile manipulator (WMM) is a non-holonomic system. So it is required to develop a kinematic controller to make the robot system follow a desired end-effector and platform trajectories in its workspace coordinates simultaneously.

A mobile manipulator in this investigation is a 4 – axis manipulator mounted on a non-holonomic differential wheeled mobile platform (as shown in Fig.5.1) which has two driving wheels and one castor wheels; the two driving wheels are independently driven by two motors and the castor wheel considered here for stability of the platform.

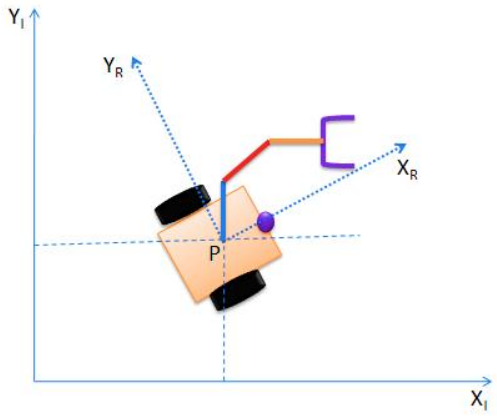


Fig.5.1 Mobile manipulator representation

5.1.Velocity Jacobean of Manipulator

Chapter 3 addressed the development of forward & inverse kinematic models of a 4-axis manipulator. The arm equation is obtained as represented in Eq.(3.13), which is function of θ_1, θ_2, θ_3 and θ_4. The arm equation consists of six elements; three corresponds to the end-effector's position and the remaining three represents the orientation of the end-effector.

The time derivative of the end-effector's position gives the linear velocity of the end-effector. The position of the end-effector $[p_x \quad p_y \quad p_z]^T$ is a function of (θ_1, θ_2, θ_3) because θ_4 indicates the orientation of the tool. According to Eq.(3.13):

$$\begin{Bmatrix} p_x \\ p_y \\ p_z \end{Bmatrix} = \begin{Bmatrix} c_1(a_2c_2 + a_3c_{23} - d_4s_{23}) \\ s_1(a_2c_2 + a_3c_{23} - d_4s_{23}) \\ d_1 - a_2s_2 - a_3s_{23} - d_4c_{23} \end{Bmatrix}$$

$$\Rightarrow \begin{Bmatrix} v_x \\ v_y \\ v_z \end{Bmatrix} = \frac{d}{dt}\begin{Bmatrix} p_x \\ p_y \\ p_z \end{Bmatrix} \tag{5.1}$$

$$\Rightarrow \begin{Bmatrix} v_x \\ v_y \\ v_z \end{Bmatrix} =$$

$$\begin{Bmatrix} -s_1\dot{\theta}_1(a_2c_2 + a_3c_{23} - d_4) + c_1\left(-a_2^2s_{23}\dot{\theta}_2 - a_3^2s_{23}^2(\dot{\theta}_2 + \dot{\theta}_3) - d_4^2c_{23}(\dot{\theta}_2 + \dot{\theta}_3)\right) \\ c_1\dot{\theta}_1(a_2c_2 + a_3c_{23} - d_4s_{23}) + s_1\left(-a_2^2s_{23}\dot{\theta}_2 - a_3^2s_{23}^2(\dot{\theta}_2 + \dot{\theta}_3) - d_4^2s_{23}(\dot{\theta}_2 + \dot{\theta}_3)\right) \\ -a_2^2c_2\dot{\theta}_2 - a_3c_{23}^2(\dot{\theta}_2 + \dot{\theta}_3) + d_4^2s_{23}(\dot{\theta}_2 + \dot{\theta}_3) \end{Bmatrix} \tag{5.2}$$

$$\Rightarrow \begin{Bmatrix} v_x \\ v_y \\ v_z \end{Bmatrix} = [J_M]_{3x3}\begin{Bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{Bmatrix} \tag{5.3}$$

Where $[J_M]_{3x3}$ is manipulator velocity Jacobean matrix and is equal to:

$$\begin{bmatrix} -s_1(a_2c_2 + a_3c_{23} - d_4s_{23}) & -a_2^2s_2c_1 - a_3^2s_{23}c_1 - d_4c_{23}c_1 & -a_3s_{23}c_1 - d_4c_{23}c_1 \\ c_1(a_2c_2 + a_3c_{23} - d_4s_{23}) & -a_2s_1s_2 - a_3s_1s_{23} - d_4s_1s_{23} & s_1(-a_3s_{23} - d_4s_{23}) \\ 0 & -a_3c_{23} + d_4s_{23} & -a_3c_{23} + d_4s_{23} \end{bmatrix} \tag{5.4}$$

5.2.Velocity Jacobean of Mobile Platform

There are three constraints for a differential platform as mentioned in chapter – 4; first one corresponds to move the platform in the direction of axis of symmetry and the remaining two are rolling constraints not allow the wheels to slip. The motion equation of a differential mobile platform is a function of left wheel and right wheel velocities as represented in the Eq.(4.28), since these two are rotating independently to impart the motion to the entire system.

From Eq.(4.28) $\dot{\xi}_I = \begin{Bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{Bmatrix} == \frac{1}{2s} \begin{bmatrix} s * cos\psi & s * cos\psi \\ s * sin\psi & s * sin\psi \\ -1 & 1 \end{bmatrix} * \begin{Bmatrix} v_{Rt} \\ v_{Lt} \end{Bmatrix}$

While moving the mobile manipulator, the following kinematic Eq.(5.5) is used which relates the linear velocity of the mobile platform reference frame to the wheel velocities.

$$\begin{Bmatrix} V_x \\ V_y \end{Bmatrix} = [J_{MP}]_{2x2} \begin{Bmatrix} \dot{\theta}_{rt} \\ \dot{\theta}_{lt} \end{Bmatrix} \tag{5.5}$$

Where $[J_{MP}]_{3x3}$ is mobile platform velocity Jacobean matrix and $\dot{\theta}_{rt}$ & $\dot{\theta}_{lt}$ are angular velocities of right and left wheels respectively

Velocity Jacobean matrix $[J_{MP}] = \frac{1}{2r} \begin{bmatrix} cos\psi & cos\psi \\ sin\psi & sin\psi \end{bmatrix}$

5.3.Velocity Jacobean of Mobile Manipulator

The differential kinematics of the mobile manipulator can be obtained by combining the kinematic Eqs. (5.4) & (5.5) of a 4-axis manipulator and the differential mobile platform. The parametric matrix shown in Eq. (5.6) represents the motion of the mobile manipulator.

$$\{\dot{q}\} = \begin{Bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_{rt} \\ \dot{\theta}_{lt} \end{Bmatrix} \tag{5.6}$$

The first parameters in the above equation relate to the manipulator and the remaining two corresponds to the differential platform. Therefore the final kinematic model of the mobile manipulator is as follows:

$$\begin{Bmatrix} v_x \\ v_y \\ v_z \\ V_x \\ V_y \end{Bmatrix} = [J_{WMP}]_{5x5} \begin{Bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_{rt} \\ \dot{\theta}_{lt} \end{Bmatrix} \tag{5.7}$$

Where $[J_{WMP}]$ is the Velocity Jacobean of Mobile Manipulator and is represented as follows:

$$[J_{WMP}] = \begin{bmatrix} [J_{WMP}]_{11} & [J_{WMP}]_{12} & [J_{WMP}]_{13} & [J_{WMP}]_{14} & [J_{WMP}]_{15} \\ [J_{WMP}]_{21} & [J_{WMP}]_{22} & [J_{WMP}]_{23} & [J_{WMP}]_{24} & [J_{WMP}]_{25} \\ [J_{WMP}]_{31} & [J_{WMP}]_{32} & [J_{WMP}]_{33} & [J_{WMP}]_{34} & [J_{WMP}]_{35} \\ [J_{WMP}]_{41} & [J_{WMP}]_{42} & [J_{WMP}]_{43} & [J_{WMP}]_{44} & [J_{WMP}]_{45} \\ [J_{WMP}]_{51} & [J_{WMP}]_{52} & [J_{WMP}]_{53} & [J_{WMP}]_{54} & [J_{WMP}]_{55} \end{bmatrix}$$

Where

$$[J_{WMP}]_{11} = -s_1(a_2c_2 + a_3c_{23} - d_4s_{23})$$

$$[J_{WMP}]_{12} = -a_2^2s_2c_1 - \dot{a_3^2}s_{23}c_1 - d_4c_{23}c_1$$

$$[J_{MP}]_{13} = -a_3s_{23}c_1 - d_4c_{23}c_1$$

$$[J_{WMP}]_{14} = [J_{MP}]_{15} = 0$$

$$[J_{WMP}]_{21} = c_1(a_2c_2 + a_3c_{23} - d_4s_{23})$$

$$[J_{WMP}]_{22} = -a_2s_1s_2 - a_3s_1s_{23} - d_4s_1s_{23}$$

$$[J_{WMP}]_{23} = s_1(-a_3s_{23} - d_4s_{23})$$

$$[J_{WMP}]_{24} = [J_{MP}]_{25} = 0$$

$$[J_{WMP}]_{31} = 0$$

$$[J_{WMP}]_{32} = -a_3c_{23} + d_4s_{23}$$

$$[J_{WMP}]_{33} = -a_3c_{23} + d_4s_{23}$$

$$[J_{WMP}]_{34} = [J_{MP}]_{35} = 0$$

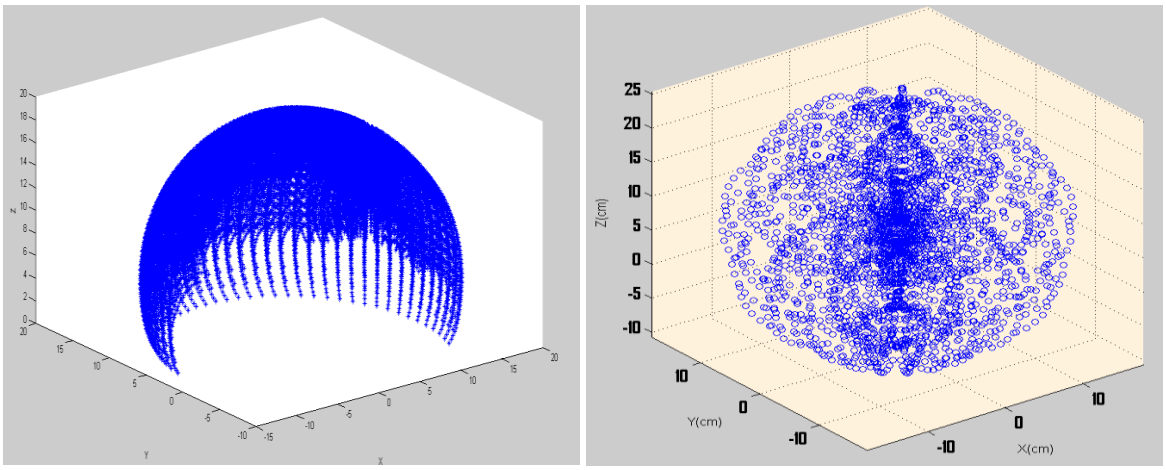
$$[J_{WMP}]_{41} = [J_{MP}]_{42} = [J_{MP}]_{43} = 0$$

$$[J_{WMP}]_{44} = [J_{MP}]_{45} = {}^{(cos\psi)}/_{2r}$$

$$[J_{WMP}]_{51} = [J_{MP}]_{52} = [J_{MP}]_{53} = 0$$

$$[J_{WMP}]_{54} = [J_{MP}]_{55} = {}^{(sin\psi)}/_{2r}$$

Fig. 5.2 represents the workspace generated by the mobile manipulator, when it is at a specific position. The developed hybridised system extends the workspace of a fixed manipulator by two times as shown in Fig.5.2.



(a) (b)

Fig.5.2 Workspace generated by (a)Manipulator and (b)WMM [All dimensions are in cm.]

5.4. Summary

This chapter addressed the kinematic structure of a wheeled mobile manipulator. The study integrates the kinematic models of a 4-axis manipulator and a differential mobile platform. The developed WMM is controlled by five parameters in which three parameters (joint velocities) gives the velocity information of the manipulator and the remaining two (left & right wheel velocities) corresponds the differential mobile platform. The final velocity Jacobean has been developed for a mobile manipulator which controls the entire robot system. Thereby, it makes the robot to follow desired trajectories by the manipulator and platform within its workspace.

Chapter 6

SWARM BASED CONTROL SYSTEM PARADIGM

6. SWARM BASED CONTROL SYSTEM PARADIGM

Navigational strategies for mobile robot in cluttered environment require serious attention by the researchers for effective path planning. Path planning generally includes the generation of optimal collision free trajectories within its work space and finally reaches its destination position. Based on this issue the path planning can be categorized into two types namely global path planning and local path planning. In first category, the robot generates the path from its initial position to final position in its known static environments. In second category, robot generates path trajectories within its unknown environments. While the robot is in motion, path planning should follow three aspects: 1) Acquire the knowledge from its environmental conditions. 2) Determine its position in the environment and 3) Decision-making and execution to achieve its highest-order goals. This chapter aims at develop an efficient particle swarm optimization (PSO) based path planner of an autonomous mobile robot.

Swarm Intelligence is an innovative distributed intelligent paradigm for solving optimization problems that originally took its inspiration from the biological examples by swarming, flocking and herding phenomena in vertebrates.

6.1. PSO Structure

PSO is a population based methodology, which was inspired by social behaviour of bird flocking or fish schooling. The population considered in PSO is called swarm and its individuals are known as particles. So a swarm in PSO can be defined as a set $S = \{P_1, P_2, P_3, \dots, P_n\}$. Where $P_1, P_2, P_3, \dots, P_n$ are 'n' number of particles in the swarm. These particles are assumed to move within the search space. While the particles are moving, their new positions can be updated with a proper position shift called velocity. Let us consider the positions of 'n' particles are: $\{x_1, x_2, x_3, \dots, x_n\}$ and their velocities are: $\{v_1, v_2, v_3, \dots, v_n\}$. The new velocity of each particle is obtained from the communicated information of particles among the swarm. It can be done in terms of memory i.e. each particle stores its best position, it has ever visited during its search. The best position decided by each particle is called position best and is indicated by X_{pbest}. So there are 'n' number of position best values for 'n' particles in the swarm.

Now the particles in the swarm are mutually communicated their experience and they will approximate to one global best position, ever visited by all particles as shown in Fig.6.1. Selection of global best position can be done by calculating the fitness values of each particle in the swarm.

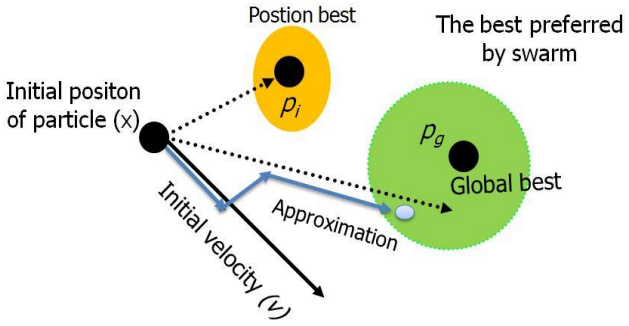


Fig.6.1 Basic structure of PSO for global best approximation

The particle which has the best fitness value can be treated as the global best position and is represented by X_{gbest}. The determination of X_{gbest} indicates the completion of one PSO-iteration. This process will be continued until maximum number of iterations has occurred or robot has reached its target. Once finding each X_{pbest} and swarm X_{gbest}, the velocity and position of each particle will be updated according to Eqs. (6.1) & (6.2).

$$v_i(k+1) = v_i(k) + C_1 * rand1 * \left(X_{pbest} - x_i\right) + C_2 * rand2 * (X_{gbest} - x_i) \quad (6.1)$$

And $x_i(k+1) = x_i + v_i(k+1)$ (6.2)

Where k= iteration counter; rand1 & rand2 are random variables and C_1 & C_2 are cognitive and social parameters.

6.2. Mobile Robot System Architecture

PSO can be applied to mobile robot navigation by defining a fitness function as well as transforming it into a minimization problem. The efficiency of a motion planner depends on the two conditions: the primary condition is, the robot has to generate trajectories by avoiding obstacles and escaping traps; and second priority (condition) is the robot has to reach its target by travelling short distance in minimum possible time. In order to find fitness of each individual, a fitness function has to be developed, which should meet the above mentioned aspects.

If the robot is not sensing any obstacles in its environment, the robot can move towards its destination. Therefore it is not necessary to apply any adaptive mechanism to move the robot within its work space. But it is very difficult task to generate trajectories for an autonomous mobile robot, when it senses obstacles in its environment.

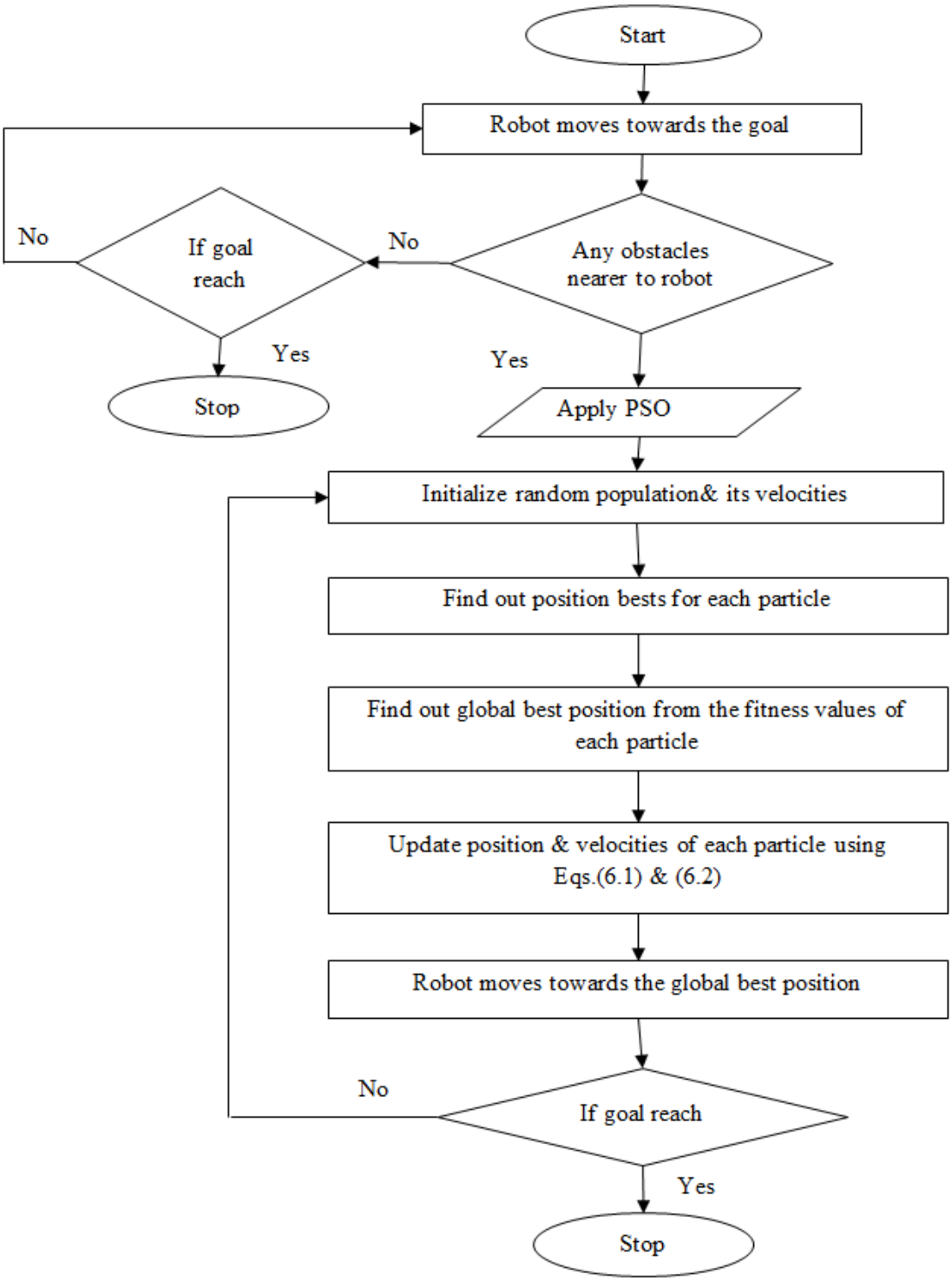


Fig.6.2 Flow chart for mobile robot navigation using PSO

The present research work analyzes PSO based system architecture for obtaining optimal path trajectories when the robot senses obstacles within its work space. In this way the developed system architecture will work for generating optimal path trajectories of an autonomous mobile robot within its unknown environments. The flow chart for this methodology is represented in Fig.6.2.

6.2.1. Fitness function development

During navigation, a mobile robot has sensed certain number of obstacles (S_{ob}) within its sensing range, then the robot can detect the nearest obstacle according to reflected radiation intensity from the sensed obstacles. The robot is represented as a point $(robotx, roboty)$ in X, Y – plane. Similarly the centres of sensed obstacles are represented as a point $(obx_i, oby_i)$for $1 \leq i \leq S_{ob}$. Then the distance between the robot and the sensed obstacles can be obtained from equation (6.3):

$$(dist_{ROb})_i = \sqrt{(robotx - obx_i)^2 + (roboty - oby_i)^2} \quad \text{for } 1 \leq i \leq S_{ob} \qquad (6.3)$$

From the calculated 'S_{ob}' number of distance values, the obstacle which is having minimum $dist_{ROb}$ can be selected as a nearest obstacle. Once the robot detects the nearest obstacle (NOb) within its sensing range, robot will generate a random population/swarm around it within the sensing range. So one fitness function (F) is required to calculate the fitness of each particle in the swarm for further robot movement.

Let the positions of particle, target and nearest obstacle be represented in XY-plane as (p_{x_i}, p_{y_i}), $(goalx, goaly)$ and (NOb_x, NOb_y), then the distance from each particle to robot's destination and nearest obstacle can be calculated from Eqs. (6.4) & (6.5).

$$dist_{p_iT} = \sqrt{\left(p_{x_i} - goalx\right)^2 + \left(p_{y_i} - goaly\right)^2} \qquad (6.4)$$

$$dist_{p_iNOb} = \sqrt{\left(p_{x_i} - NObx\right)^2 + \left(p_{y_i} - NOby\right)^2} \qquad (6.5)$$

As explained in previous section it is necessary to find the fitness value of each particle in the swarm. For this purpose a new fitness function has to model by satisfying the following conditions.

1. *First priority condition:* The fitness of particle should maintain the maximal distance from the nearest obstacle, in other words the fitness function is indirectly proportional to distance between the particle and nearest obstacle. Because of this condition, a repulsive action is generated between the particle and the obstacle.

$$\Rightarrow F_i \propto \left({}^{1}/_{dist_{P_iNOb}}\right) \qquad \text{for } 1 \leq i \leq n \tag{6.6}$$

Where $dist_{P_iNOb}$ indicates the distance between i[th] particle and nearest obstacle.

2. *Second priority condition:* The fitness of particle should maintain the minimal distance from the robot's destination, in other words the fitness function is directly proportional to distance between the particle and target. Because of this condition, an attractive action is generated between the particle and the target in order to move the robot towards its destination.

$$\Rightarrow F_i \propto \left(dist_{P_iT}\right) \qquad \text{for } 1 \leq i \leq n \tag{6.7}$$

Where $dist_{P_iT}$ indicates the distance between i[th] particle and target position.

6.2.2. Type 1 fitness function

From the above mentioned conditions shown by Eqs.(6.6) and (6.7), the required fitness function should maintain the both attractive action towards its target and repulsive action towards the nearest obstacle. So the final form of the fitness function can be generated as represented by Eqs.(6.8) and (6.9).

$$F_i = W_1 * dist_{P_iT} + W_2 * \left({}^{1}/_{dist_{P_iNOb}}\right) \tag{6.8}$$

Where $W_1$and $W_2$are the proportionality constants/controlling parameters can be varying according to the positions of particle, target and nearest obstacle.

6.2.3. Type 2 fitness function

Eqs.(6.6) and (6.7) is can be represented in another way (Eqs. 6.9)

$$F_i = K_1 * \left({}^{\lambda_{P_i-T}}/_{\lambda_{P_i-NOb}}\right) \qquad \text{for } 1 \leq i \leq p \tag{6.9}$$

Where K_1 is the proportionality/controlling parameter can be varying according to the positions of particle, target and nearest obstacle.

While considering the $min(\lambda_{P_i-T})$ and $max(\lambda_{P_i-NOb})$, there are two particles say k^{th} & l^{th} particles which maintain $min(\lambda_{P_k-T})$ & $max(\lambda_{P_l-NOb})$ respectively. Therefore the possible minimum value of the fitness using equation (6.10) is

$$F_{min} = \left({min(\lambda_{P_k-T})}/{max(\lambda_{P_l-NOb})} \right) \quad \text{for } 1 \leq k, l \leq p \tag{6.10}$$

The objective of the motion planner is to find out the particle which would maintain the similar properties of the k^{th} & l^{th} particles. Therefore the final fitness function can be transformed from equations (6.9) & (6.10):

$$(F_{final})_i = |F_{min} - F_i|$$

$$\Rightarrow (F_{final})_i = \left| \left({min(\lambda_{P_k-T})}/{max(\lambda_{P_l-NOb})} \right) - K_1 * \left({\lambda_{P_i-T}}/{\lambda_{P_i-NOb}} \right) \right| \quad \text{for } 1 \leq i, k, l \leq p \tag{6.11}$$

The selection of X_{gbest} will be continued for several cycles until the robot is away from the obstacle or it reaches to its destination.

By observing Eqs. (6.8) & (6.11), the particle which is having the minimum fitness value can be treated as X_{gbest}, because that particle (X_{gbest}) is away from nearest obstacle and close to the goal position. The selection of X_{gbest} will be continued for several cycles until the robot is away from the sensed obstacles or it reaches to its destination. The algorithm for PSO based mobile robot navigation is as follows:

Step 1: Initialize robot source and destination positions.

Step 2: Robot moves until it senses any obstacles or its target position.

Step 3: If robot senses any obstacles, apply PSO.

Step 4: Initialize positions and velocities of random population.

Step 5: Obtain each particle's X_{pbest} and swarm X_{gbest}.

Step 6: Find out new positions and velocities of each particle by using Eqs. (6.1) & (6.2)

Step 7: Repeat steps 4, 5 and 6 until the robot is away from the sensed obstacles.

Step 8: Repeat step 2 until robot reaches its destination.

Note: The velocities of particles in the swarm are here used for obtaining their position best and swarm global best position; but the particle velocities are not influencing the robot velocity.

Once the robot detects global best position among the swarm, it will start its motion towards the X_{gbest}. In this manner, iterations will be continued until the robot is away from the sensed obstacles or maximum possible number of cycles has reached.

6.3. Simulation Results

From Eqs. (6.8) & (6.11), it can be noticed that the minimum fitness value is considered as X_{gbest} from the particle among the swarm. During the analysis, population (=80) is initialized randomly by defining their positions and velocities around the robot within its sensing range (say 20 units for calculation purpose) and velocities of particles are varying from 0 to 5. Until the robot senses any obstacles within its sensing range, it will continue its motion towards the predefined destination as shown in Fig.6.3. Once the robot senses any obstacles in front it, it will form a swarm of 80 particles randomly within its sensing range as shown in Fig.6.4.

Robot path generated in green colour indicates the robot motion when it is not facing any obstacles; yellow coloured points around the mobile robot symbolize the random particle distribution within its sensing range and small red coloured circles correspond to the global best positions obtained by calculating fitness value of each individual. White coloured trajectories represent the most favourable path obtained by the proposed PSO algorithm as shown in Fig.6.4.

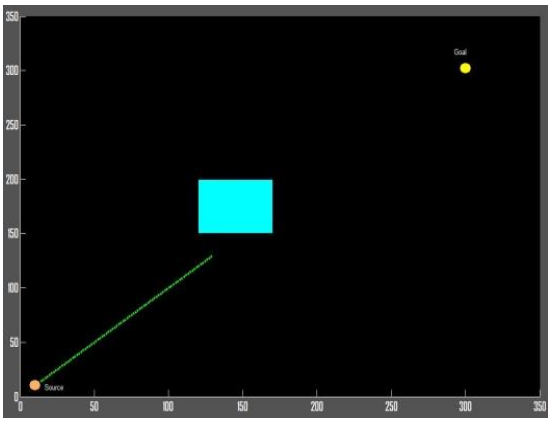


Fig.6.3 Initial robot motion towards goal

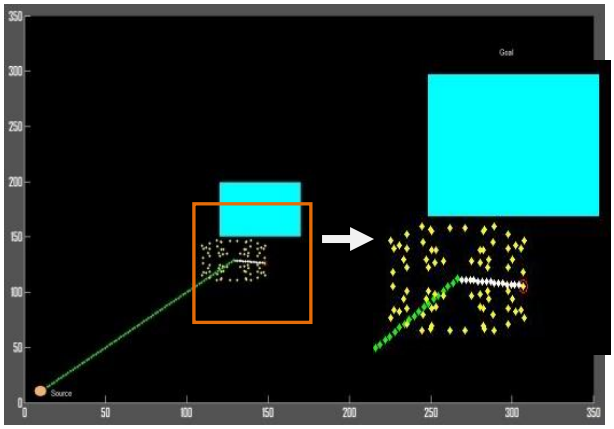


Fig.6.4 Swarm generation by robot

6.3.1. First fitness function controlling parameters (W_1&W_2)

For getting the feasible paths generated by robot, it is necessary to tune the fitness parameters as explained above. As discussed in previous section, the primary objective of the work is to obtain collision free path and target seeking is the secondary priority, therefore the obstacle avoidance parameter W_2 should convey more weight than target seeking parameter W_2. The first controlling parameter W_1 in Eq. (6.8) indicates the closeness of the particle to the robot's target and second controlling parameter W_2 indicates the particle far away from the nearest obstacle. So X_{gbest} can be obtained by minimizing the fitness function as shown in Eq.(6.8). High value of W_1 indicates the particle is very close to the target and low value of W_1 indicates the particle is far from the robot's target. Similarly high value of W_2 indicates the particle is maintaining more distance from the nearest obstacle and low value of W_2 indicates the particle is very close to the nearest obstacle. So it is required to adjust the controlling parameters of fitness function to low W_1 and high W_2 values.

Tuning of W_1&W_2*:*

Velocity of each particle depends on the parameters C_1&C_2 and $rand1$ & $rand2$. Usually the random values $rand1$ & $rand2$ are varying in the range of [0, 1], these values are influencing the particle velocity but not the robot travelled distance. For simplicity these values are adjusted to a fixed value '1'. The fitness parameters $W_1$and W_2 can be adjusted according to the mobile robot travelled distance within its work space. It means C_1&C_2 are indirectly effecting (to find X_{gbest}) while tuning the parameters $W_1$and W_2. For easy consideration, simulation experiments are conducted at the values of $C_1 = 1$ & $C_2 = 1$.

PSO parameters: $rand1 = rand2 = C_1 = C_2 = 1$

While performing the analysis, there are four possible cases as follows:

Case 1: High values of $W_1(\geq 1)$ and High values of $W_2(\geq 150)$

Table 6.1 Experimental results for Case 1

W_1	W_2	Robot travelled distance (cm)	Collision free path(Yes/No)
1	150	457.2	No
	300	460.8	No
	450	466.4	No
	600	477.6	No
	750	488	No
	900	481.6	No
2	150	453.2	No
	300	457.2	No
	450	459.6	No
	600	460.8	No
	750	466	No
	900	466.4	No
3	150	453.2	No
	300	453.2	No
	450	457.2	No
	600	460.8	No
	750	459.6	No
	900	460.8	No

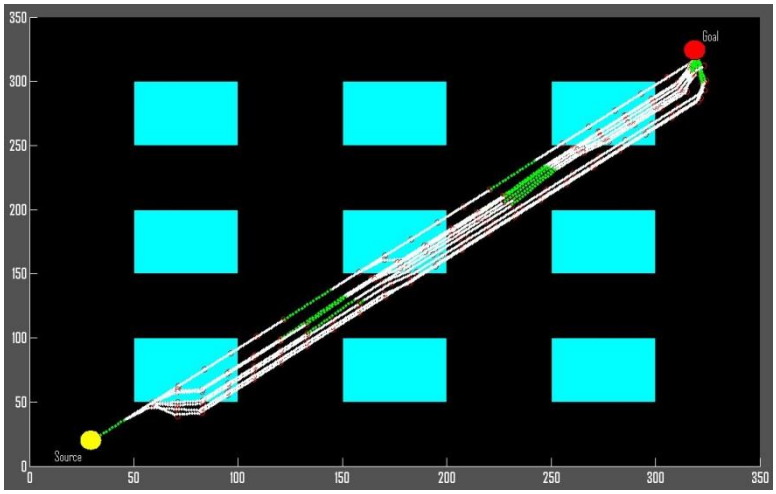


Fig.6.5 Mobile robot path for Case 1

Case 2: High values of $W_1(\geq 1)$ and Low values of $W_2(\leq 150)$

Table 6.2 Experimental results for Case 2

W_1	W_2	Robot travelled distance (cm)	Collision free path(Yes/No)
1	0	457.2	No
	50	461.6	No
	100	464.4	No
	150	465.6	No
2	0	455.6	No
	50	474	No
	100	455.6	No
	150	461.6	No
3	0	462	No
	50	455.2	No
	100	462.8	No
	150	462.8	No

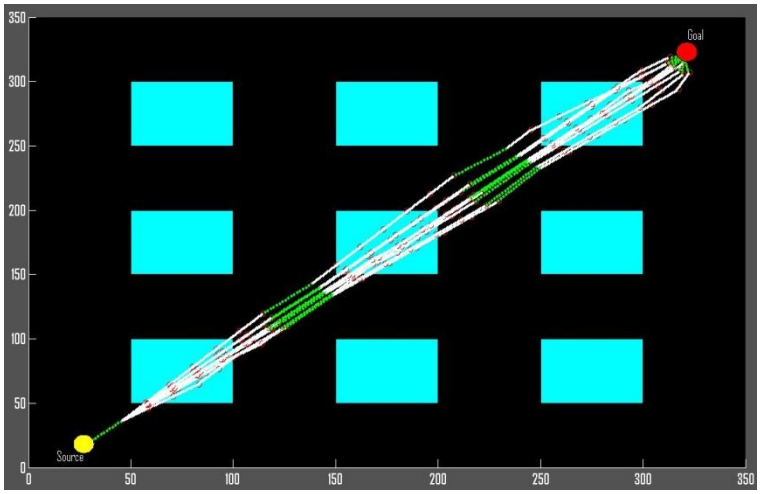


Fig.6.6 Mobile robot paths for Case 2

Case 3: Low values of $W_1(\leq 1)$ and Low values of $W_2(\leq 150)$

Table 6.3 Experimental results for Case 3

W_1	W_2	path travelled (cm)	Collision free path(Yes/No)
0	0	566.4	Robot follows zigzag motion (Yes)
	50	519.6	Robot takes more navigational time (Yes)
	100	553.2	Robot follows zigzag motion (Yes)
	150	553.2	Robot follows zigzag motion (Yes)
0.3	0	455.6	No
	50	456.4	No
	100	467.6	No
	150	471.6	No
0.6	0	458.4	No
	50	472.8	No
	100	465.6	No
	150	463.2	No
0.9	0	484	No
	50	460.8	No
	100	453.2	No
	150	462	No

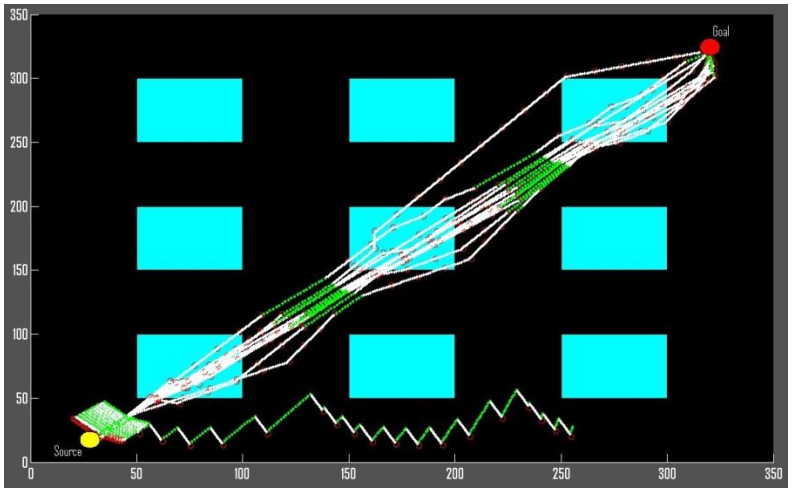


Fig.6.7 Mobile robot paths for Case 3

Case 4: Low values of $W_1(\leq 1)$ and High values of $W_2(\geq 150)$

Table 6.4 Experimental results for Case 4

W_1	W_2	Robot travelled distance (cm)	Collision free path(Yes/No)
0.1	150	545.2	Yes
	300	625.2	Yes
	450	613.2	Yes
	600	553.2	Yes
	750	553.2	Yes
	900	553.2	Yes
0.3	150	466.4	No
	300	488.4	No
	450	545.2	Yes
	600	580	Yes
	750	661.2	Yes
	900	625.2	Yes
0.5	150	460.8	No
	300	477.6	No
	450	481.6	No

	600	497.6	No
	750	545.2	Yes
	900	538	Yes (Min.)
0.7	150	460	No
	300	466.4	No
	450	481.6	No
	600	488	No
	750	484.4	No
	900	499.6	No
0.9	150	457.2	No
	300	460.8	No
	450	466.4	No
	600	481.6	No
	750	488	No
	900	488.4	No

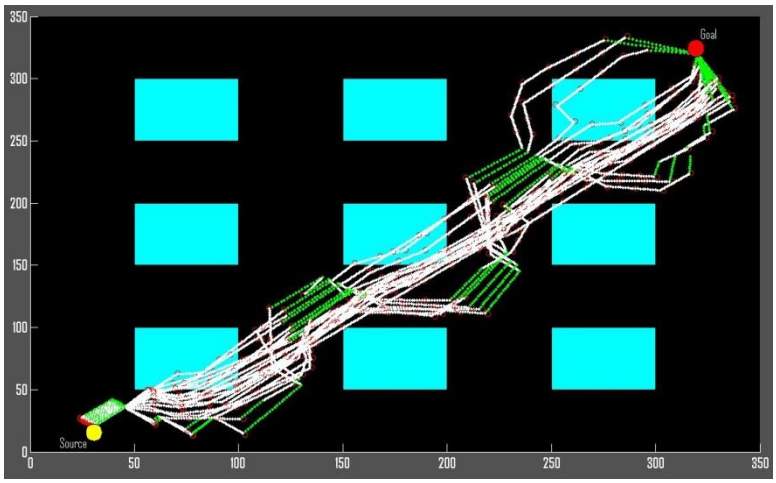


Fig.6.8 Mobile robot paths for Case 4

By observing the results from Table.6.1 to 6.3 and Figs.6.6 to 6.8, the robot can't satisfy the primary criteria i.e. avoiding obstacle in first three cases. The robot is generating collision free paths in some situations, when W_1 *and* W_2 values according to Case 4. By observing the results from first two cases, it can be noticed that the robot can't generate collision free path at larger values of W_1 (>1). From Table.6.4, it can be observed that the robot is generating

least distance path at the values of $W_1=0.5$& $W_2=900$. So it is necessary to carry out the experiments at $W_1=0.5$ to 0.65 & $W_2=750$ to 900 to find the best path travelled by the mobile robot.

Table 6.5 Experimental results for $W_1=0.5$ to 0.65 & $W_2=750$ to 900

W_1	W_2	Robot travelled distance (cm)	Collision free path(Yes/No)
0.5	750	545.2	Yes
	800	532.8	Yes (**Min**.)
	850	537.6	Yes
	900	538	Yes
0.55	750	505.2	No
	800	543.2	Yes
	850	527.6	No
	900	537.6	Yes
0.6	750	504.4	No
	800	505.2	No
	850	505.2	No
	900	545.2	Yes
0.65	750	494.8	No
	800	497.6	No
	850	505.2	No
	900	505.2	No

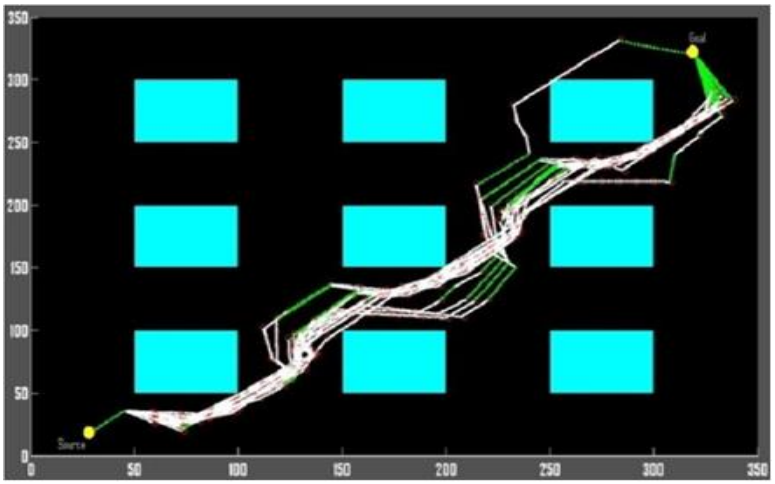


Fig.6.9. Mobile robot paths for Table 6.5 (W_1 & W_2) parameters consideration

From the above statistical results illustrated in Table 6.5 corresponds to the Fig.6.9, it is concluded that the robot is generating most favorable paths at W_1=0.5 & W_2=800.

6.3.2. Second fitness function Controlling Parameters (K_1)

The final form of the fitness function for an i[th] particle is represented with Eq. (6.11)

$$\left(F_{final}\right)_i = \left| \left({min\left(\lambda_{P_k - T}\right)} \Big/ {max\left(\lambda_{P_l - NOb}\right)} \right) - K_1 * \left({\lambda_{P_i - T}} \Big/ {\lambda_{P_i - NOb}} \right) \right|$$

and the value $\left({min\left(\lambda_{P_k - T}\right)} \Big/ {max\left(\lambda_{P_l - NOb}\right)} \right)$ is always less than $\left({\lambda_{P_i - T}} \Big/ {\lambda_{P_i - NOb}} \right)$.

So the robot path is influenced by the tuning parameter K_1 as follows:

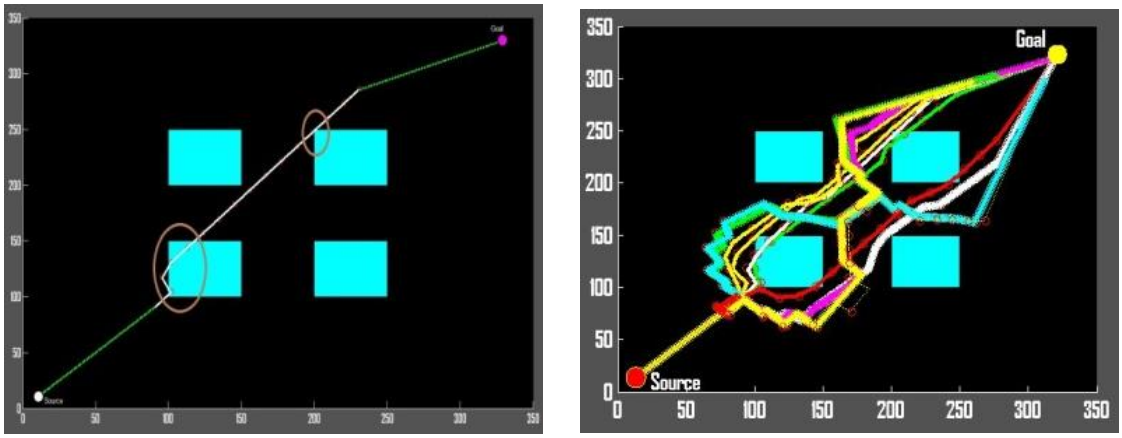
Case 1: For higher values of K_1 (>2)

While considering the higher K_1 values (>2), the particles fitness value becoming larger since $\left({min\left(\lambda_{P_k - T}\right)} \Big/ {max\left(\lambda_{P_l - NOb}\right)} \right)$ is always less than $\left({\lambda_{P_i - T}} \Big/ {\lambda_{P_i - NOb}} \right)$. A particle with higher fitness value is nothing but the particle is maintaining less $\lambda_{P_i - NOb}$ which gives obstacle collision path by the robot as shown in Fig.6.10 (a) (robot path at K_1=2). So the consideration according to the case 1 contradicts the problem statement.

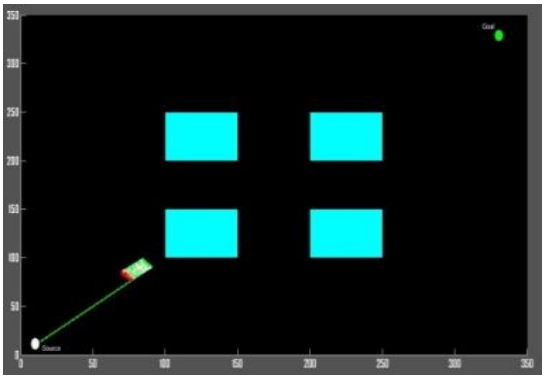
Case2: For lower values of K_1 (0< K_1<2)

While considering the lower K_1 values (<2), the particles fitness value becoming smaller and satisfying the criteria of maintaining shortest distance from the target and highest distance from the nearest obstacle. But at very less fitness values, the particle is far away from the nearest obstacle. In this situation, the developed algorithm may find the wrong X_{gbest} which causes the robot is in indefinite motion as shown in Fig.6.10 (c) (robot path for K_1=0.1).

By observing the above two cases, it is not recommended the higher (>2) and lower (<0.1) K_1 values. It means that the robot will generate feasible paths at 0.1< K_1<2. From the statistical results it is found that the robot is giving most favorable paths at K_1=1.3.



(a) (b)



(c)

Fig.6.10. Robot motion at (a) K_1= 2, (b) 2< K_1>0.1, (c) K_1= 0.1

A large amount of experiments as tabulated in Table 6.6 have been performed in order to tune the parameter K_1. Statistical results showed that the values of K_1 greater than '1.6' are not obtaining collision free paths as shown in Fig.6.10 (b) and it can be noticed that the robot is facing trap situation for small values of $K_1 \leq 0.3$.

Table 6.6 Experimental results for K_1 variation

K_1	**Robot travelled distance (cm)**	**Collision free path(Yes/No)**
0.1	560.4	Yes (Indefinite motion)
0.2	555.2	Yes (Indefinite motion)
0.3	552.4	Yes (Indefinite motion)
0.4	548.8	Yes
0.5	548.6	Yes
0.6	546.4	Yes

0.7	545.0	Yes
0.8	542.8	Yes
0.9	542.2	Yes
1.0	542.2	Yes
1.1	541.6	Yes
1.2	540.8	Yes
1.3	**538.2 (Minimum)**	Yes
1.4	538.6	Yes
1.5	538.6	Yes
1.6	535.2	No
1.7	533.8	No
1.8	530.4	No
1.9	528.6	No
2	525.2	No

Later experiments have been conducted when $K_1$is varying from 0.3 to 1.6. Results showed that the mobile robot is obtaining optimal trajectories when it sensing obstacles in front of it, at the value K_1=1.3.

6.3.3. Tuning of social parameters (C_1&C_2)

The next step is to adjust the social and cognitive parameters C_1&C_2 in order to get better results than the previous analysis. The experiments are conducted at parameter criteria; for the first fitness function, the controlling parameters at W_1=0.5 & W_2=800 and for the second fitness function K_1 = 1.3. The robot paths are shown in Fig.6.11 at different C_1 and C_2 values.

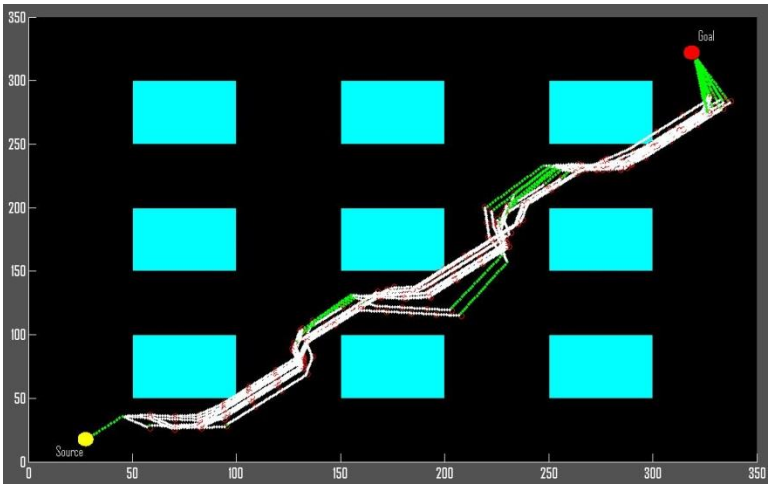


Fig.6.11 Mobile robot paths at different C_1 and C_2 values

Table 6.7 Experimental results for various C_1&C_2

C_1	C_2	Robot travelled distance (cm)		Collision free path(Yes/No)
		1st fitness function	2nd fitness function	
0.5	0.5	533.2	537.4	Yes
	1	533	537	Yes
	1.5	532.6	535.6	Yes
	2	529.6	534.8	No
1	0.5	533	537.2	Yes
	1	532.8	535.4	Yes
	1.5	532	534.6	Yes (**Min.**)
	2	530.4	532.8	No
1.5	0.5	534	536.8	Yes
	1	533.6	535	Yes
	1.5	532.8	535.6	Yes
	2	531.6	532.8	No
2	0.5	534.2	536	Yes
	1	534.2	536	Yes
	1.5	534.6	538.2	Yes
	2	534.8	538.2	Yes

From Table 6.7, the results showed that the robot is generating most favorable and shortest paths at $C_1 = 1$ & $C_2 = 1.5$.

6.3.4. Stochastic nature of the developed PSO motion planners

While implementing PSO path planner for several runs in the same environmental criteria, the path deviation is observed within the first twenty runs.

The following norms are considered for the proposed PSO motion planner:

Start criteria: when obstacle(s) are sensed.

Population: generation of 80 individuals randomly.

Stop criteria: until obstacle avoidance.

Number of runs performed: 20 runs for the same environmental criteria.

The stochastic nature of the developed PSO algorithm is represented in Fig.6.12. When the number of runs increased, threshold deviation is decreased continuously for the first 18 cycles and then it reaches to an asymptotic value of threshold 0.09%.

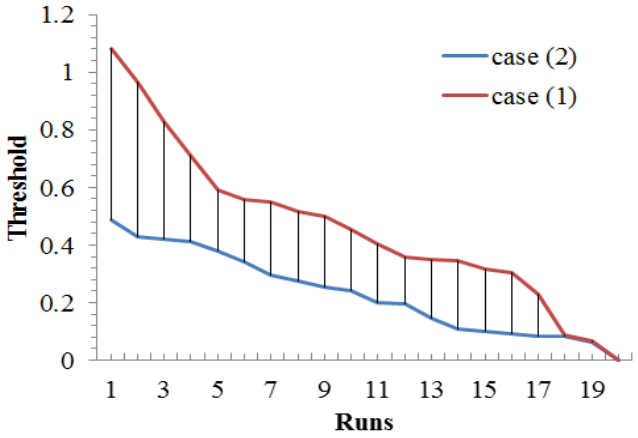


Fig.6.12 Stochastic nature of the developed PSO based path planner

For the scenarios case 1 & 2 of Fig.6.14, the shortest paths are achieved at 13th & 20th runs respectively. The threshold variation with respect to shortest path at each run is shown Fig.6.13. Path analysis results for the scenario, Fig.6.14 are illustrated in Table 6.8.

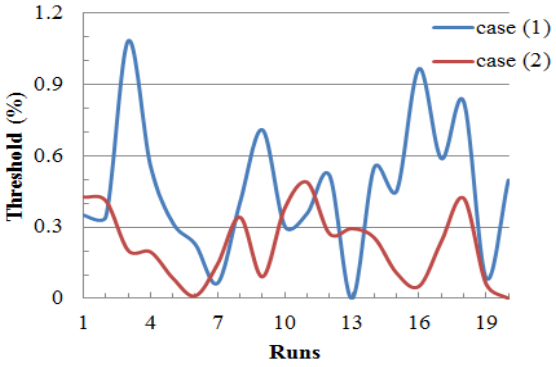
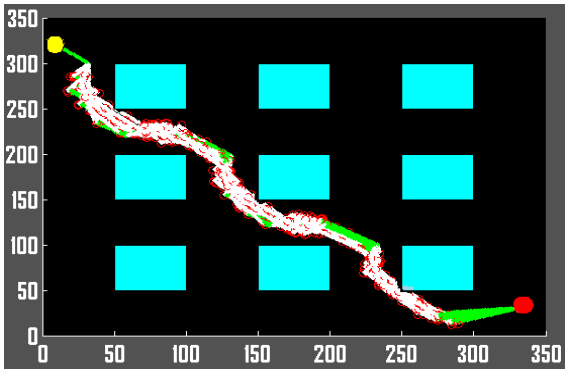
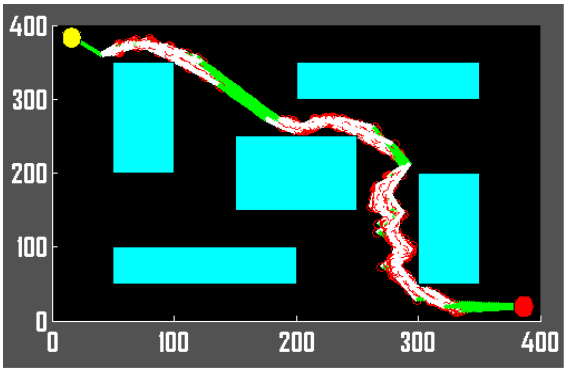


Fig.6.13 Shortest path achievement with respect to each run



Case (1)



Case (2)

Fig.6.14 Path obtained by the robot in the same environmental criteria for 20 runs

Table 6.8 Path analysis results for Fig.6.14

Case (1)					
Run index	**Distance travelled(cm) by the robot**		**% deviation with respect to shortest distance travelled**		**Stop criteria**
	1st Fitness	**2nd Fitness**	**1st Fitness**	**2nd Fitness**	
1st run	569.2	573.9	0.351	0.36	Target reached
2nd run	568.8	573.6	0.345	0.35	Target reached
3rd run	616	620.9	1.084	1.09	Target reached
4th run	581.6	586.4	0.557	0.56	Target reached
5th run	567.2	571.5	0.317	0.32	Target reached
6th run	562	566.5	0.228	0.23	Target reached
7th run	552.8	556.8	0.065	0.06	Target reached
8th run	572.4	576.9	0.405	0.41	Target reached
9th run	591.2	595.9	0.71	0.71	Target reached
10th run	566.4	570.9	0.304	0.31	Target reached
11th run	569.6	574.4	0.358	0.37	Target reached
12th run	579.2	584.2	0.518	0.53	Target reached
13th run	549.2	553.4	0	0	Target reached
14th run	581.2	585.8	0.551	0.55	Target reached
15th run	575.2	579.8	0.452	0.45	Target reached
16th run	608	613	0.967	0.97	Target reached

17th run	583.6	588.3	0.589	0.59	Target reached
18th run	598.8	603.3	0.828	0.83	Target reached
19th run	554	558.1	0.087	0.08	Target reached
20th run	578	582.8	0.498	0.5	Target reached
Case (2)					
Run index	**Distance travelled(cm)**		**% deviation with respect to shortest distance travelled**		**Stop criteria**
	1st Fitness	**2nd Fitness**	**1st Fitness**	**2nd Fitness**	
1st run	730.8	736.9	0.427	0.43	Target reached
2nd run	729.6	735.7	0.411	0.41	Target reached
3rd run	714	719.7	0.202	0.2	Target reached
4th run	713.6	719.5	0.196	0.2	Target reached
5th run	705.6	710.9	0.085	0.08	Target reached
6th run	700.4	706	0.011	0.01	Target reached
7th run	710	715.2	0.146	0.14	Target reached
8th run	724.4	730.1	0.342	0.34	Target reached
9th run	706	711.6	0.091	0.09	Target reached
10th run	727.2	732.9	0.38	0.38	Target reached
11th run	735.6	741.9	0.489	0.49	Target reached
12th run	719.2	725.4	0.273	0.28	Target reached
13th run	720.8	726.4	0.294	0.29	Target reached
14th run	718	723.7	0.256	0.25	Target reached
15th run	707.2	712.9	0.107	0.1	Target reached
16th run	703.2	709	0.051	0.05	Target reached
17th run	716.8	722.5	0.24	0.24	Target reached
18th run	730.4	735.9	0.422	0.41	Target reached
19th run	704	709.2	0.062	0.05	Target reached
20th run	699.6	705.4	0	0	Target reached

6.4. Comparison between the developed PSO motion planners

Figs.6.15 (a) - (d) show the path generation of the developed autonomous mobile robot in its unknown environments. Comparison between simulation as well as experimental results are provided to validate the capability of the robot system using the proposed PSO based system architectures.

For scenario - i of Figs.6.15 (a), the mobile manipulator is moving from its source position (50, 375) to the target position (440, 10) using type-1 fitness based motion planner.

- Fig. 6.15 (a) - (i) represents the robot motion (in simulation mode) from source to target.
- Fig. 6.15 (a) - (ii) represents the robot motion (in real mode) from source to target.
- Fig. 6.15 (a) - (iii), Fig. 6.15 (a) - (iv), Fig. 6.15 (a) - (v) represents the robot positions during its navigation from source to target.

Similarly, for scenario - i of Figs.6.15 (b), the mobile manipulator is moving from its source position (50, 375) to the target position (440, 10) using type-2 fitness based motion planner.

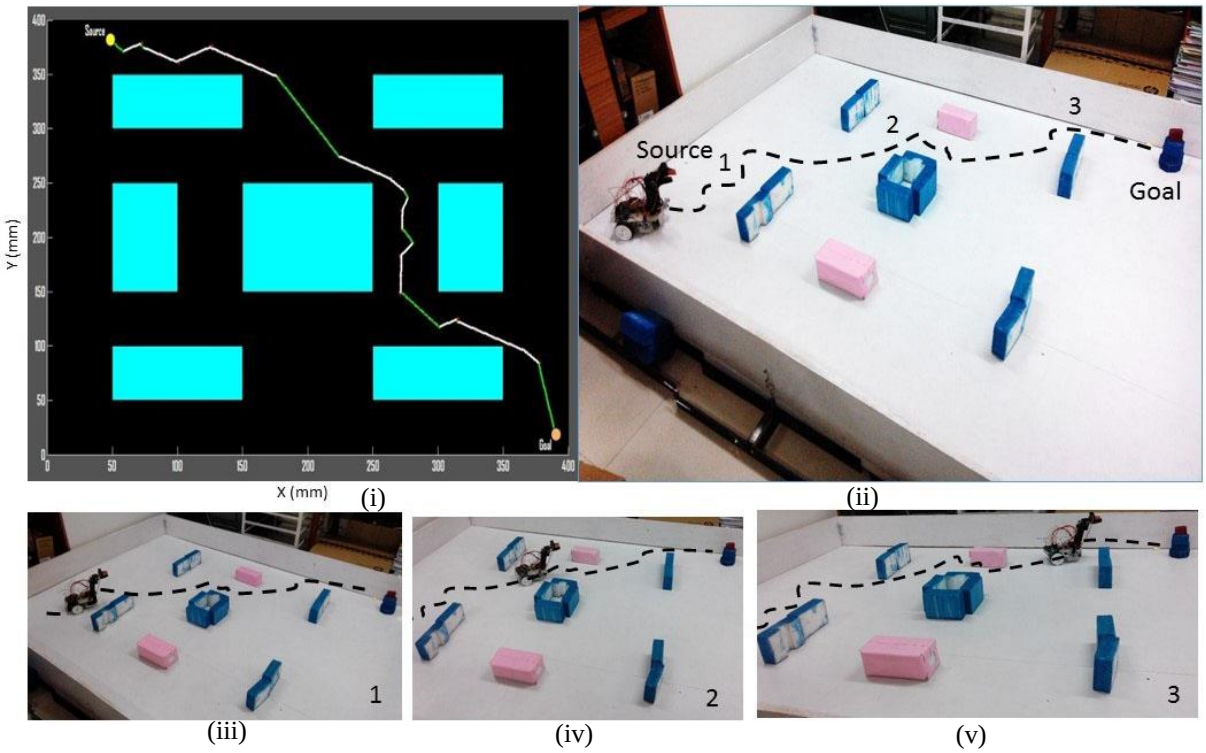
- Fig. 6.15 (b) - (i) represents the robot motion (in simulation mode) from source to target.
- Fig. 6.15 (b) - (ii) represents the robot motion (in real mode) from source to target.
- Fig. 6.15 (b) - (iii), Fig. 6.15 (b) - (iv), Fig. 6.15 (b) - (v) represents the robot positions during its navigation from source to target.

For scenario - ii of Figs.6.15 (c), the mobile manipulator is moving from its source position (10, 365) to the target position (45, 360) using type-1 fitness based motion planner.

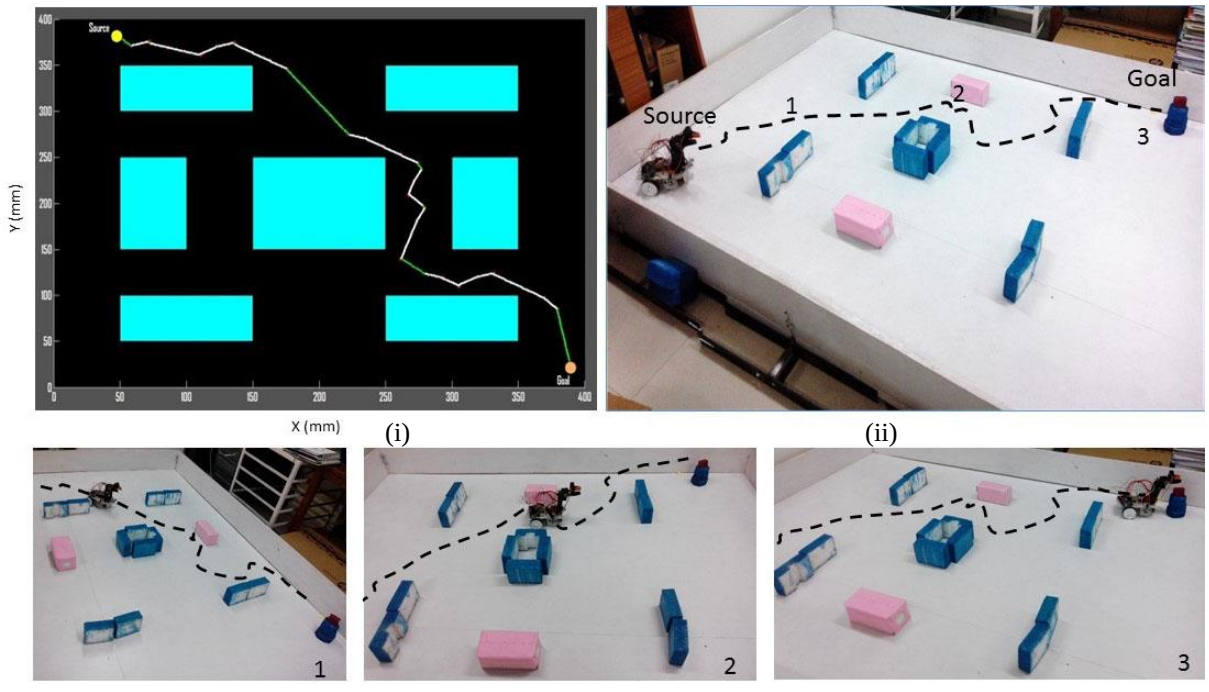
- Fig. 6.15 (c) - (i) represents the robot motion (in simulation mode) from source to target.
- Fig. 6.15 (c) - (ii) represents the robot motion (in real mode) from source to target.
- Fig. 6.15 (c) - (iii), Fig. 6.15 (c) - (iv), Fig. 6.15 (c) - (v) represents the robot positions during its navigation from source to target.

Similarly, for scenario - ii of Figs.6.15 (d), the mobile manipulator is moving from its source position (10, 365) to the target position (45, 360) using type-2 fitness based motion planner.

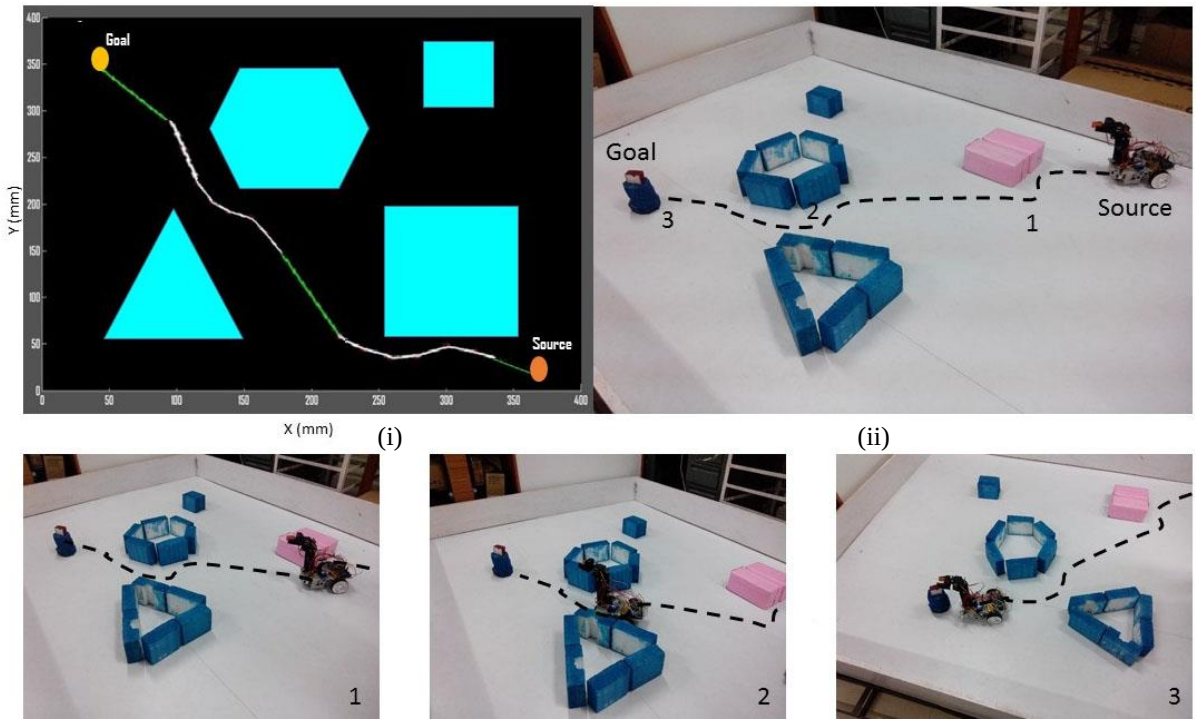
- Fig. 6.15 (d) - (i) represents the robot motion (in simulation mode) from source to target.
- Fig. 6.15 (d) - (ii) represents the robot motion (in real mode) from source to target.
- Fig. 6.15 (d) - (iii), Fig. 6.15 (d) - (iv), Fig. 6.15 (d) - (v) represents the robot positions during its navigation from source to target.



6.15 (a) Path generated by 1st fitness path planner (Scenario (i))



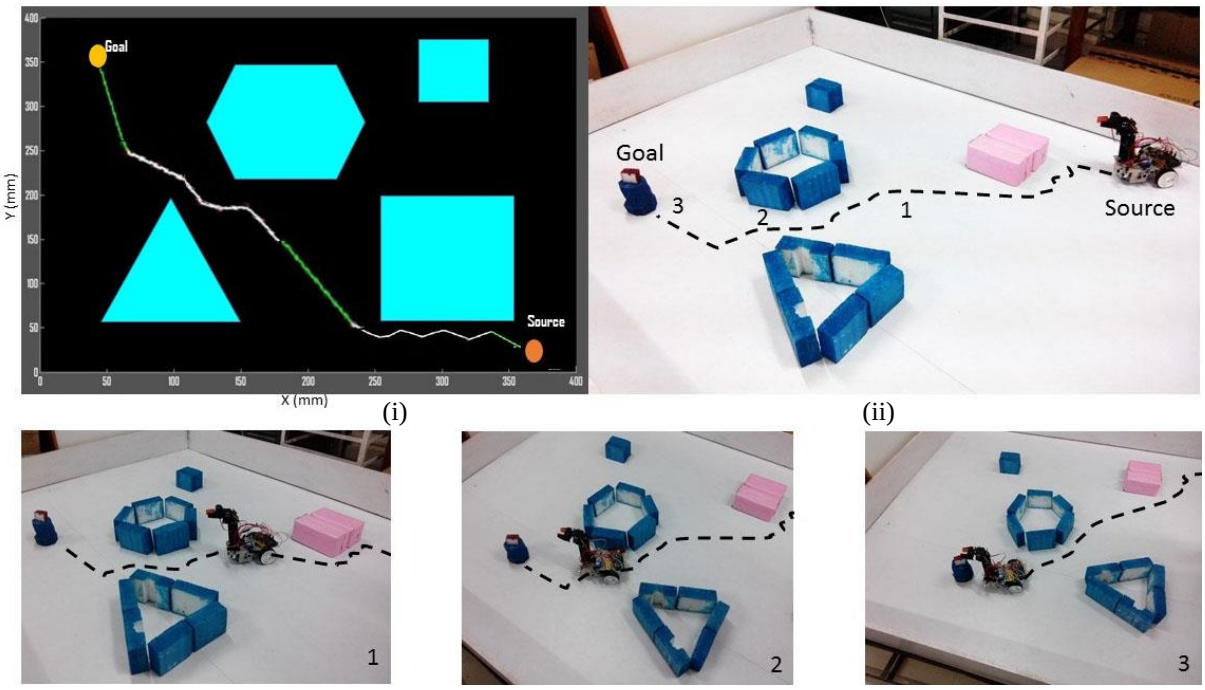
6.15 (b) Path generated by 2nd fitness path planner (Scenario (i))



(i) (ii)

(iii) (iv) (v)

6.15 (c) Path generated by 1st fitness path planner (Scenario (ii))



(i) (ii)

(iii) (iv) (v)

6.15 (d) Path generated by 2nd fitness path planner (Scenario (ii))

Fig.6.15 Mobile manipulator paths in its search space while reaching to its target

For same scenario-i, type-1fitness based PSO motion planner obtained the shortest paths as compared to type-2 fitness based PSO motion planner. From the path analysis results as illustrated in Table 6.9, for scenario-i type-2 fitness based motion planner deviated the travelled distance by 7.3% from the simulation results and 6.6% form the experimental results. Similarly, for scenario-ii, type-2 fitness based motion planner deviated the travelled distance by 2.3% from the simulation results and 2.2% form the experimental results. Path analysis results of scenarios- i & ii showed that the robot motion in simulation environments are giving good agreement (percent of error is below 5%) with the real time environments.

From the path analysis results as illustrated in Table 6.9, it is concluded that the motion planner with first fitness function is giving better results as compared to the motion planner with second fitness function.

Table 6.9 Path analysis results for Fig.6.15

Scenario	Fitness type	Simulation				Experimental				% of error
		No. of Iterations	Time Taken (sec.)	Distance travelled(cm)	% of path deviation	No. of Iterations	Time Taken (sec.)	Distance travelled(cm)	% of path deviation	
i	1	481	192.4	577.2	7.3%	495	198	594	6.6%	2.8%
	2	519	207.6	622.8		530	212	636		2.1%
ii	1	461	184.5	553.6	2.3%	449	179.6	538.8	2.2%	2.7%
	2	472	188.8	566.4		459	183.6	550.8		2.8%

6.5. Comparison with Previous work

When the robot moves from its source position to the destination position, the considerations for the developed PSO based path planner are as follows:

Population generated within sensing range: 80

No. of runs performed: 20

Distance travelled by the robot when PSO is activated: 1.2 units

Distance travelled when the robot is not sensed any obstacles: 2 units

PSO parameters at $C_1 = 1$ & $C_2 = 1.5$ and $rand1 = rand2 = 1$.

Stop criteria: until Target reached

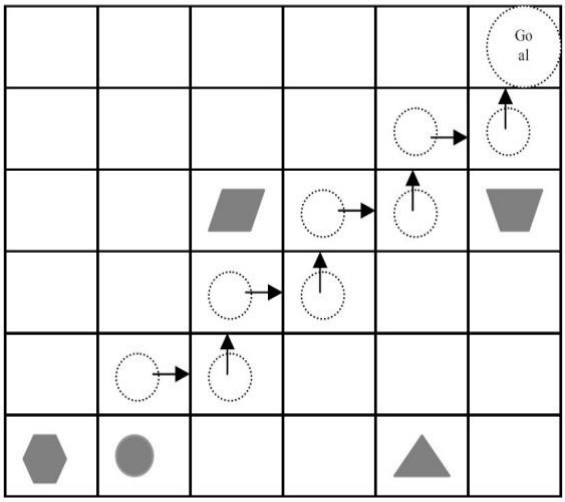
First fitness proportional parameters W_1=0.5 & W_2=800

Second fitness function proportional parameter K_1 = 1.3.

Note: The path travelled by the robot is represented in various environments by considering abscissa as the X-axis in centimeters (cm) and ordinate as the Y-axis in centimeters.

6.5.1. Comparison with respect to 1st Fitness function

Das *et. al.* [158] have implemented a well-known heuristic A* algorithm for solving mobile robot navigation in static unknown environment. In their work, they considered the cost function as the time metric of distance travelled by the mobile robot. The aim of their work is to minimize the cost function by using A* algorithm. In other words the total distance travelled by the mobile robot from its initial position to destination should be minimum.



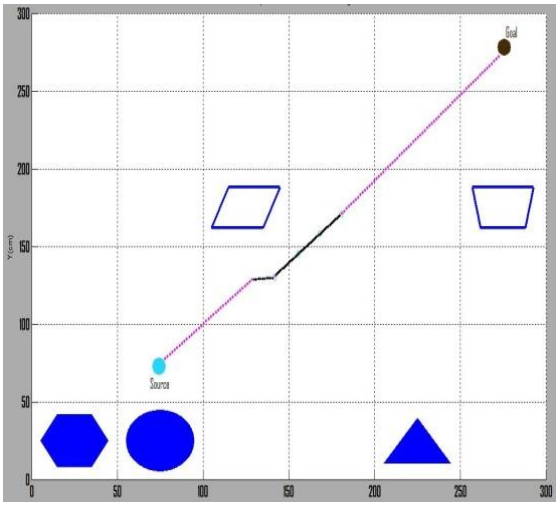
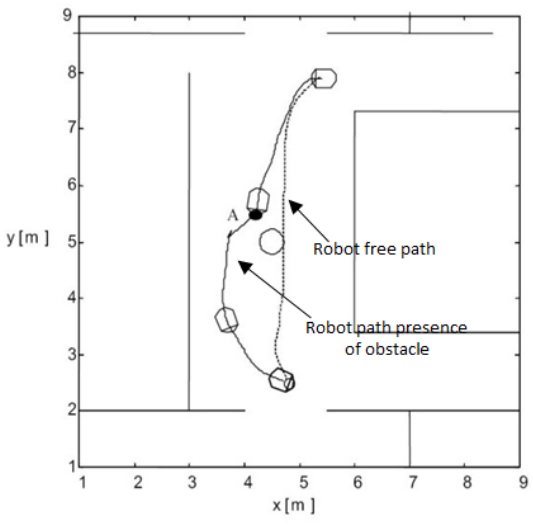


Fig.6.16 (a) Path obtained by Das *et. al.* [158] Fig.6.16 (b) Path obtained by present Motion planner



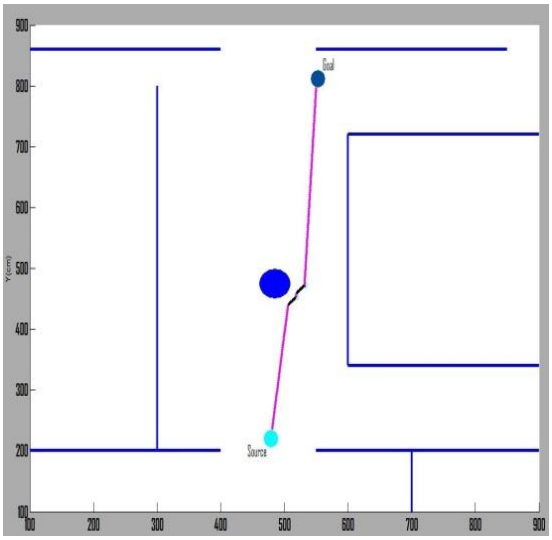


Fig.6.17 (a) Path obtained by Secchi *et. al.* [159] Fig. 6.17(b) Path obtained by present Motion planner

Secchi *et. al.* [159] have presented an effective control law for obstacle avoidance in unknown environments. The proposed control system concerns two loops namely, position control loop and impedance control loop. Impedance here referred as a function of the distance between the robot and the sensed obstacles.

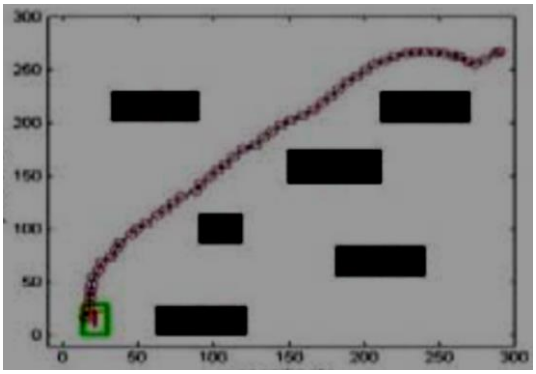


Fig.6.18 (a) Robot Path by Zawawi *et. al.* [160]

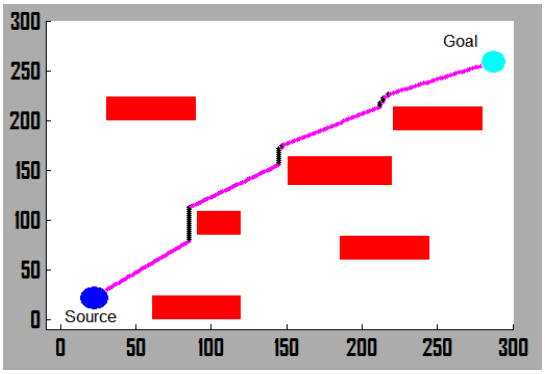


Fig.6.18 (b)Path obtained by present Motion planner

Zawawi *et. al.* [160] have described an efficient system architecture development for an autonomous mobile robot using visual simultaneous localisation & mapping, and particle swarm optimization. Their developed methodology is suitable for navigating a mobile robot in indoor environments.

Table 6.10 Path analysis results for Figs.6.16-6.18

Previous methodology	**Distance travelled(cm)**		**% of path deviation**
	Previous methodology	**Present methodology**	
Heuristic A* algorithm [158]	470.8	455.2	3.3%
Position and impedance control loops [159]	1103.6	1037.4	6%
Particle Swarm Optimization [160]	495.2	482.6	2.5%

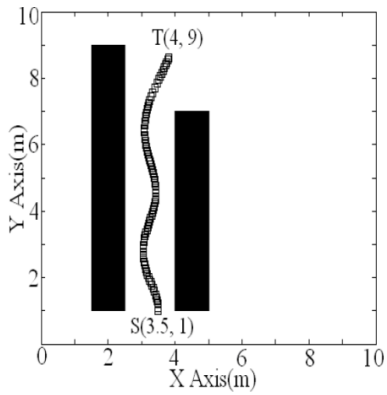
From the Table 6.10, it is noticed that the current motion planner is giving better results as compared to results obtained by Das *et. al.* [158], Secchi *et. al.* [159] and Zawawi *et. al.* [160] by robot path deviation of 3.3%, 6% and 2.5% respectively.

6.5.2. Comparison with respect to 2nd Fitness function

To solve mobile robot navigation task, various approaches have been introduced in last few decades. Fuzzy Inference System (FIS) is one of the well-known approach have been used for

solving path planning problem of an autonomous mobile robot because of its capacity to handle uncertain and imprecise information obtained from sensors using linguistic rules.

Yong *et. al.* [161] have introduced a behaviour based architecture based on fuzzy logic for solving mobile robot navigation problem in unknown environments. Fig. 6.19(a) shows the path generated by their algorithm for an autonomous mobile robot starting at the position S(3.5, 1) and its destination at T(4,9). For the same robotic environment, the path generated by current methodology is shown in Fig. 6.19(b).



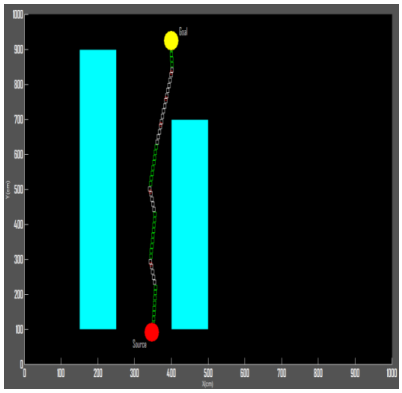
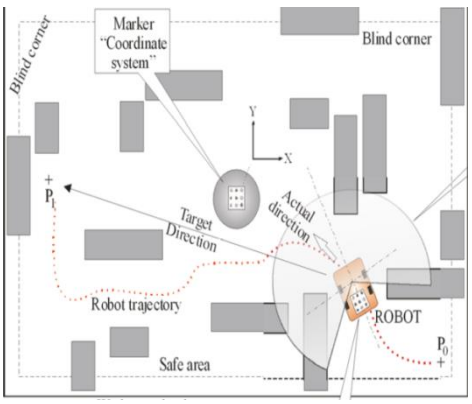


Fig.6.19 (a) Path obtained by Yong *et. al.* [161] Fig.6.19 (b) Path obtained by Current methodology

Recently, Mester and Rodic [162] have explained a sensor based intelligent mobile robot navigation in unknown environments. They used Fuzzy Inference System for generating obstacle collision free trajectories within robotic work space. A simulation result of their approach is shown in Fig. 6.20(a) regarding the goal seeking and the obstacle avoidance mobile robot paths. For the same robotic environment, the path generated by current algorithm is shown in Fig. 6.20(b).



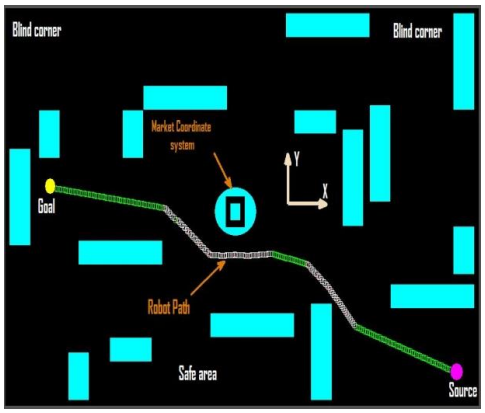


Fig. 6.20 (a) Path obtained by Mester and Rodic [162] Fig. 6.20 (b) Path obtained by Current methodology

Table 6.11 Path analysis results for Figs.6.19-6.20

Previous methodology	**Distance travelled(cm)**		**% of path deviation**
	Previous methodology	**Present methodology**	
Behavior based architecture [161]	1153.8	1107.6	4%
Fuzzy Inference System [162]	594.2	549.6	7.5%

From the Table 6.11, it is noticed that the current motion planner is generating the shortest robot path within the robotic unknown environments as compared to results obtained by Yong *et. al.* [161] and Mester and Rodic [162] by robot path deviation of 4% and 7.5% respectively.

6.6. Summary

A new computational methodology has been proposed for solving path planning problem of an intelligent mobile platform, based on Particle Swarm Optimization. The developed algorithm is effective in avoiding obstacles and generating optimal paths within its unknown environments. The trajectories generated by robot are based on the selection of global best position in each iteration. Among the swarm, the particle which has the minimum fitness is considering as the global best position. There by, the robot moves towards the global best position and this process is continued for several iterations until the robot reaches its target position. A large number of experiments have been carried out for adjusting the controlling parameters of the modelled fitness functions. Simulation results showed the capability of a mobile robot, how effectively the robot is generating trajectories with the help of developed algorithm, by avoiding obstacles, escaping traps and reaches to its goal position within its unknown maze environments. Moreover, from the developed two fitness functions, type-1 fitness is giving efficient results as compared to type-2 fitness function (Table 6.9). Deviations are found to be within 5% during comparisons between simulation and corresponding experimental results. Although the proposed methodology solves the local minima problem up to certain level than the previous researchers as addressed in Chapter 2, it requires some reinforcement learning strategy to achieve better results.

Chapter 7

IMMUNE BASED CONTROL SYSTEM PARADIGM

7. IMMUNE BASED CONTROL SYSTEM PARADIGM

Motion planning is one of the vital issues in the field of mobile robots because of their usage in various fields such as domestic fields, industries, security environments and hospitals etc. The main goal of an efficient motion planner of a mobile platform is to generate collision free trajectories from the sensory information without continuous human intervention.

In this section, two efficient immunological path planners have modelled to make the mobile platform work in intelligent way. First motion planner is inspired from the innate immune system and is focused on the special feature anomaly detection. Later an adaptive learning mechanism has been introduced on the basis of previous sensory information, to the first motion planner. The proposed adaptive methodology is simple because of very few controlling parameters in its structure. Moreover this approach is useful for solving local minima problems (avoiding obstacles & escaping traps in maze environments) and mobile platform navigation task in unknown complex environments by generating optimal collision free trajectories.

7.1. Biological Immune System

The interest in studying the immune system is increasing over the last few years. Computer scientists, engineers, mathematicians, philosophers and other researchers are particularly interested in the capabilities of this system, whose complexity is comparable to that of the brain. Many properties of the immune system are of great interest for computer scientists and engineers:

- *Uniqueness:* each individual possesses its own immune system.
- *Recognition of Foreigners:* the antigens are recognized and eliminated from body.
- *Anomaly Detection:* the immune system can detect and react to pathogens that the body has never encountered before;
- *Distributed Detection:* the cells are distributed all over the body and are not to subject any centralized control.
- *Noise Tolerance:* the system is flexible since the recognition of the antigens is not required.
- *Reinforcement Learning and Memory:* future responses to the same pathogens are faster and stronger since the immune system can “learn” the structures of pathogens.

A brief introduction to development of computational tools is followed by the presentation (Fig**.7.**1**)** of the concept of *immune engineering*, and then a more systemic view of the system is given.

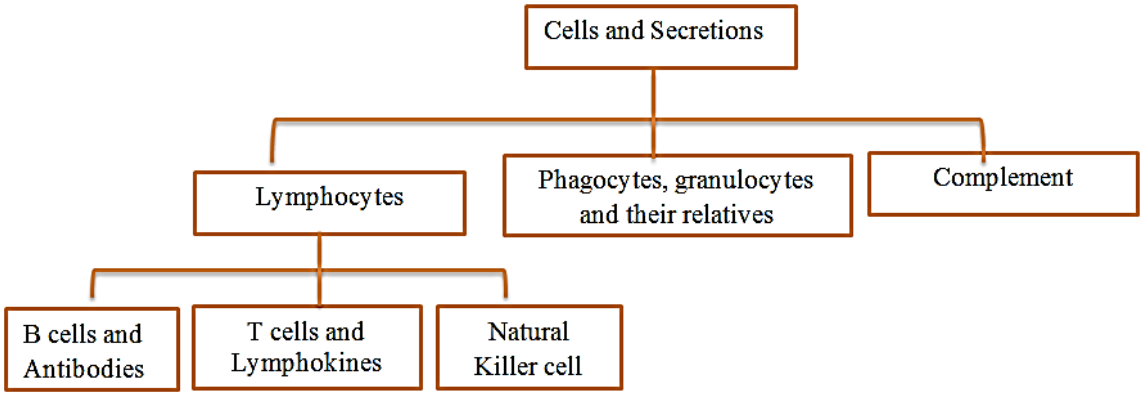


Fig**.7.**1 Structural division of the cells and secretions of the immune system

7.1.1. Basic immune models and algorithms

Because of its special features, several models have been introduced to solve various engineering problems. The following are the some of the immune system inspired algorithms.

- Bone Marrow Models
- Negative Selection Algorithms
- Clonal Selection Algorithm
- Somatic Hyper-mutation
- Immune Network Model

i. *Bone marrow models: this model represents the production of various antibodies in the* bone marrows as represented in Fig.7.2.

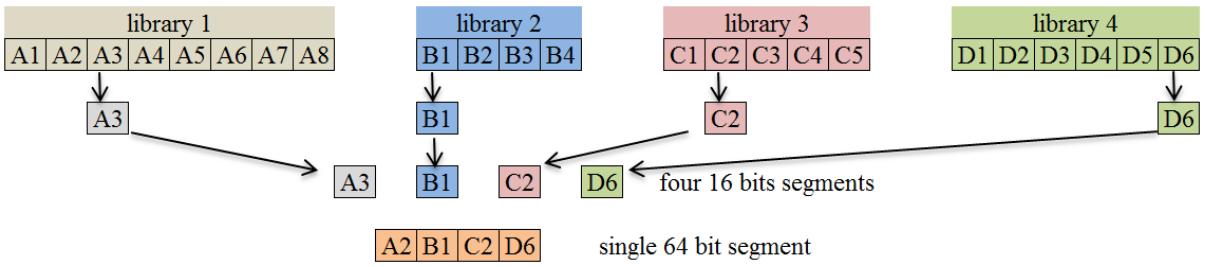


Fig.7.2 Antibody production through a random concatenation from gene libraries

ii. Negative selection algorithms: This idea taken from the negative selection of T-cells in the thymus as shown in Fig.7.3.

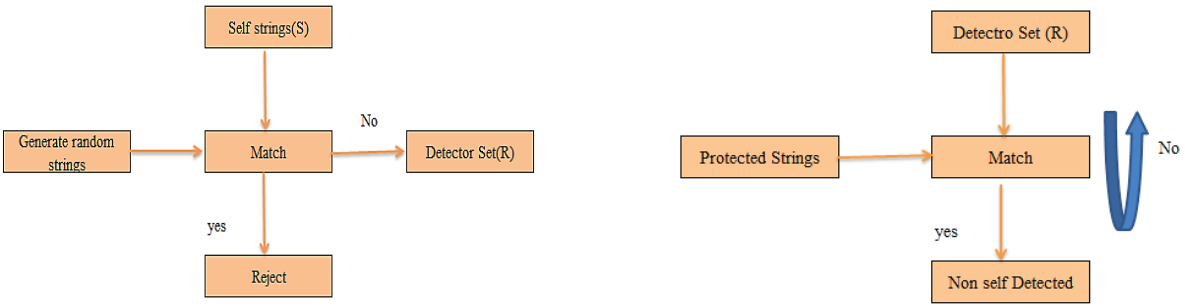


Fig.7.3 Negative selection algorithm Censoring & Monitoring

iii. Clonal selection algorithm

Randomly initialise a population (P)
For each pattern in Antigen (ag)
Determine affinity to each Antibody (ab) in P
Select *n* antibodies of highest affinity from P
Clone and mutate proportional to affinity with Ag
Add new mutants to P
End For
Select highest affinity Ab in P to form part of M and replace them
Until stopping criteria

iv. Immune network models:

Initialise the immune network (P)
For each pattern in Ag
Determine affinity to each Ab in P
Calculate network interaction
Allocate resources to the strongest members of P
Remove weakest Ab in P
End For
If termination condition met *exit*
Else Clone and mutate each Ab in P (based on a given probability) and integrate new mutants into P based on affinity
Repeat

v. Somatic hypermutation :

- Very controlled mutation in the natural immune system.
- The greater the antibody affinity the smaller its mutation rate.
- Classic trade-off between exploration and exploitation.

7.2. System Architecture

The immune system protects human body from the foreign invaders known as antigens (bacteria and virus) in two ways namely 1) innate immune system and 2) adaptive immune system. In first category, immune system produces protecting cells (antibodies) according to its special feature, anomaly detection [158]. The second type of immune system remembers the previous action i.e. when the same antigen encounters again, this immune system recognizes the antigens and it produces suitable antibodies rapidly to neutralize antigens in quick time.

7.2.1. Innate immune based motion planner

The special feature anomaly detection of the innate immune system is the recognition of antigens and produces suitable antibodies in order to vanish them. The same criteria can be implemented in the case of autonomous mobile platform navigation within its workspace. When a mobile platform is moving from one position to another position within its search space, it may sense different environmental criteria. According to the environmental situation, platform should perform its task. Fig.7.4 shows the similarities of human immune system and a behaviour based platform navigation system.

By observing Fig. 7.4, the basic immune components can be considered as follows:

- B-cell → Autonomous Mobile Platform
- Antibodies → Robot Actions (antibodies are produced from B-cell)
- Antigens → Environmental situation (B-cell has to be produced suitable antibodies according antigen structure)

Antibody Consideration: The possible robot actions and their representation in coordinate system are illustrated in Table.7.1.

Antigen consideration: The possible environmental situations are tabulated in Table 7.2

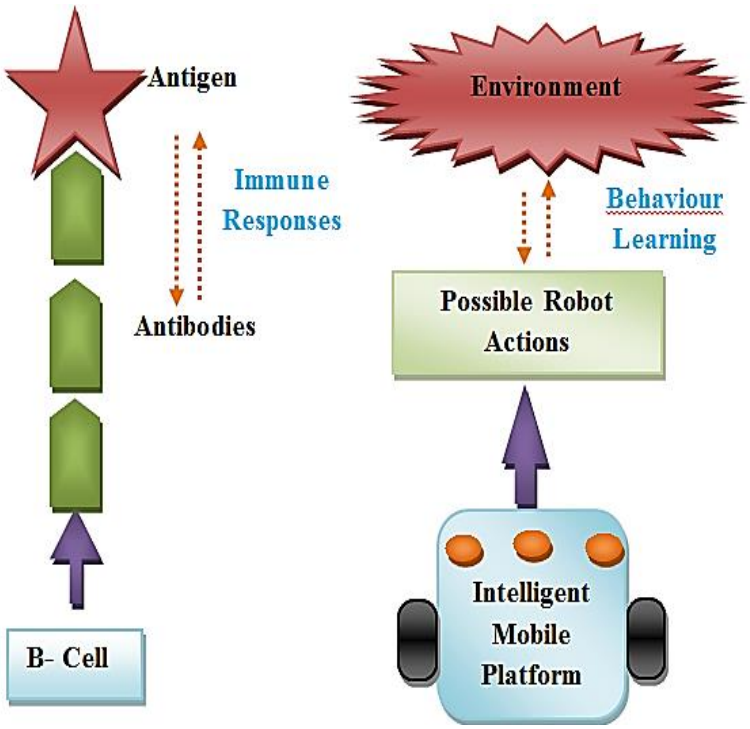


Fig.7.4 Relation between basic immune structure & mobile platform navigation systems

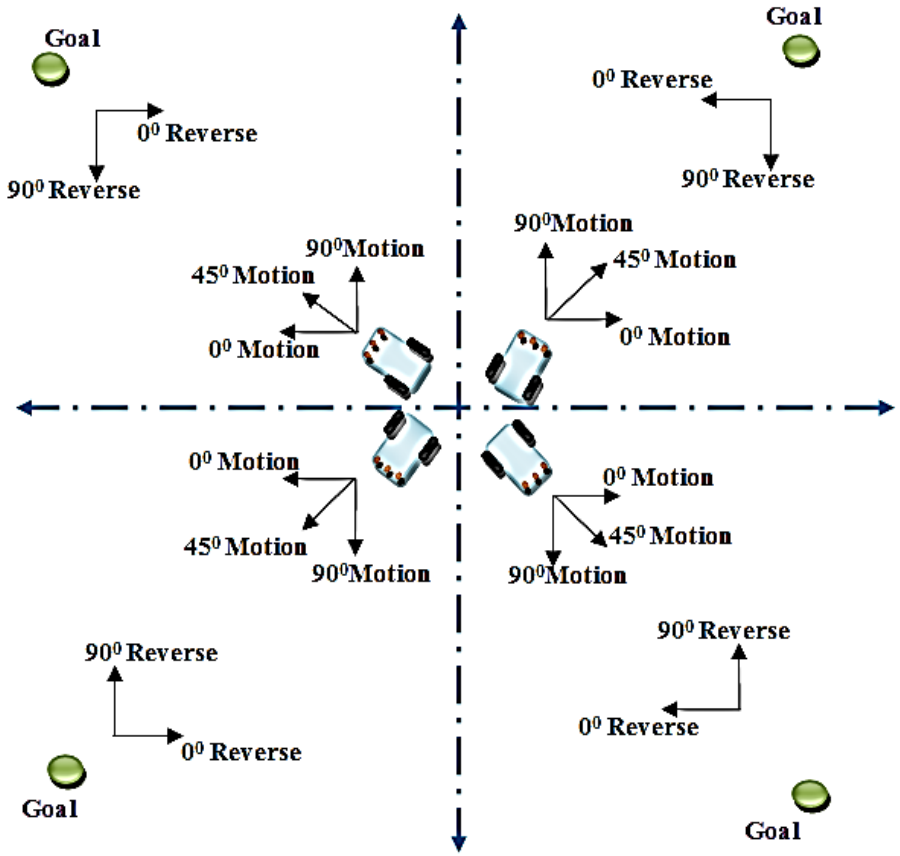


Fig.7.5 Antibody representation in mobile platform coordinate system

Table 7.1 Possible robot actions

Antibody		**Score**
Continuous Motion	Towards Goal	1
	Random	2
Left Motion	Left 10^0	3
	Left 20^0	4
	Left 30^0	5
	Left 0^0	6
Right Motion	Right 10^0	7
	Right 20^0	8
	Right 30^0	9
	Right 0^0	10
Forward Motion	90^0 robot motion	11
Reverse Motion	Reverse 0^0 (Left/Right)	12
	Reverse 10^0 (Left/Right)	13
	Reverse 20^0 (Left/Right)	14
	Reverse 30^0 (Left/Right)	15
	Reverse by $+90^0/-90^0$	16

Table 7.2 Possible robot actions

Antigen			**Score**
Goal Position	Goal Known		1
	Goal Unknown		2
Obstacle Position	No Obstacle		1
	Left	Far distance	5
		Medium distance	4
		Short distance	3
		Wall following	6
	Right	Far distance	9

		Medium distance	8
		Short distance	7
		Wall following	10
	wall following- Front		11
	Wall following (Left/Right)		12
Trap situation	Behind	Far distance	15
		Medium distance	14
		Short distance	13
	Wall following (in front/behind)		16

Note: antibody scores are assigned arbitrarily from 1 to 16 and each antigen score is assigned according to the suitable antibody score. For example consider the first antibody, robot motion towards the goal. It will occur when the robot is not sensing any obstacles and a criterion is goal known. So the scores for goal known (ag-1) is '1' and no object (ag-3) is '1'.

7.3. Behaviour Learning

In order to select suitable robot action for specific environment, a new parameter named as learning rate is introduced here and is indicated by Eq.(7.1).

$$Learning\ rate\ (\Gamma_r) = \alpha_{\wp} * \omega_{ag} \tag{7.1}$$

Allowance parameter ($\alpha_{\wp}$): $\alpha_{\wp}$ indicates the affinity strength between one antibody and one antigen. The value of $\alpha_{\wp}$ is ranging from '0' to '1' and is defined as follows.

$$\left(\alpha_{\wp}\right)_{i,j} = {min\{score(ab_i), score(ag_j)\}}\Big/{max\{score(ab_i), score(ag_j)\}} \tag{7.2}$$

Where $\left(\alpha_{\wp}\right)_{i,j}$ is allowance parameter of the ith antibody (ab_i) with respect to jth antigen (ag_j).

Antigenic weight (ω_{ag}): When the robot is in motion within its work space, there is a possibility of antigen criteria as follows:

- Sensed antigens

- Non-sensed antigens

From the sensed antigens, robot has to perform its action according to the dominant environmental situation and other sensed environmental situation. For this purpose three antigenic strengths are defined as follows:

Strength for sensed antigens → 0.5

Strength for non-sensed antigens → 0

Strength for high scored sensed antigens → 5

Once the robot gets the knowledge from its environment, it will score the antigen weights for each antigen according to Eq.(7.3)

$$\omega_{ag} = socre(ag) * strength(antigen\ type) \tag{7.3}$$

From the above analysis the learning rate of i[th] antibody with respect to j[th] antigen can be found out from Eq.(7.4).

$$(\Gamma_r)_{i,j} = \left(\alpha_{\wp}\right)_{i,j} * \omega_{ag_j} \tag{7.4}$$

$$\Longrightarrow (\Gamma_r)_{i,j} = \left(min\{score(ab_i), score(ag_j)\} \Big/ max\{score(ab_i), score(ag_j)\}\right) * socre(ag_j) * strength(ag_j) \tag{7.5}$$

Therefore the learning rate of i[th] antibody with respect to all nine antigens is as follows:

$$\overset{over\ all}{\widetilde{(\Gamma_r)_i}} = (\Gamma_r)_{i,1} + (\Gamma_r)_{i,2} + \cdots + (\Gamma_r)_{i,q} \tag{7.6}$$

$$= \left(\alpha_{\wp}\right)_{i,1} * \left(\omega_{ag}\right)_1 + \left(\alpha_{\wp}\right)_{i,2} * \left(\omega_{ag}\right)_2 + \cdots + \left(\alpha_{\wp}\right)_{i,q} * \left(\omega_{ag}\right)_{q=9}$$

$$\Rightarrow \overset{over\ all}{\widetilde{(\Gamma_r)_i}} = \sum_{j=1}^{9}\left(\alpha_{\wp}\right)_{i,j} * \left(\omega_{ag}\right)_j \tag{7.7}$$

The suitable robot action for a specific environmental condition is selected from the overall learning rate value of each antibody. The antibody which has the highest $\overset{over\ all}{\widetilde{(\Gamma_r)_i}}$ can be selected as the suitable robot action. While the robot is moving from one position to another

position within its unknown environment, the robot may face different environmental situations in each cycle. According to the extracted sensory information, the robot will calculate $\overbrace{(\Gamma_r)_i}^{over\ all}$ for each predefined robot action and it will select the suitable robot action. This process will continue until the robot reaches to its destination.

7.4. Adaptive Immune based Motion Planner

By observing the Innate Immune based Motion Planner (IIMP), the robot is not utilizing any previous sensory information when it is moving in its search space. Because of this, the robot may take more time to reach its destination. To over face this difficulty a new adaptive learning mechanism has been introduced and integrated to IIMP. The new motion planner in this investigation is the modified artificial immune system called as Adaptive Immune based Motion Planner (AIMP). AIMP utilizes the previous sensory information and makes the robot more intelligent in doing works as compared to IIMP.

7.4.1. Adaptive learning mechanism

A new adaptive learning mechanism is developed on the basis of previous robot sensory information by adding an adaptive score (a_l) to a specific allowance parameter such that:

$$\left[(\alpha_{\wp})_{i,d}\right]_t = \left[(\alpha_{\wp})_{i,d}\right]_{t-1} + a_l \qquad (7.8)$$

Where $\left[(\alpha_{\wp})_{i,d}\right]_t$ = allowance parameter of i^{th} antibody with respect to dominant antigen (high scored sensed antigen)

In order to obtain efficient results, adaptive mechanism should be developed in terms of previous sensory information. Since the IIMP works on the basis of predominant environmental situation, the adaptive score (a_l) should be the function of dominant antigen in the t^{th} and $(t\text{-}1)^{th}$ iterations. So for each cycle, allowance parameter is updating according to the robot sensed dominant antigens in the current and previous iterations. Therefore the adaptive score can be defined as follows:

$$a_l = \mathrm{k} * \left|\text{score of } ag_d\ (\mathrm{t}-1)^{\mathrm{th}} \text{ cycle} - \text{score of } ag_d\ (\mathrm{t})^{\mathrm{th}} \text{ cycle}\right| \qquad (7.9)$$

To get optimal path results, several experiments have been carried out for finding optimal 'k' value. Since $(\alpha_{\wp})_{i,j}$ should be less than '1' it means the adaptive score (a_l) is also less than '1'. Therefore the adaptive score should vary [0, 0.9]. Very less value of a_l does not affect the IIMP, so the experiments have been performed within the range [0.1, 0.9] as shown in Fig.7.6. But the paths generated (represented be several colours for various 'k' values) are not the safest paths.

Note: The path travelled by the robot is represented in various environments by considering abscissa as the X-axis in centimeters (cm) and ordinate as the Y-axis in centimeters.

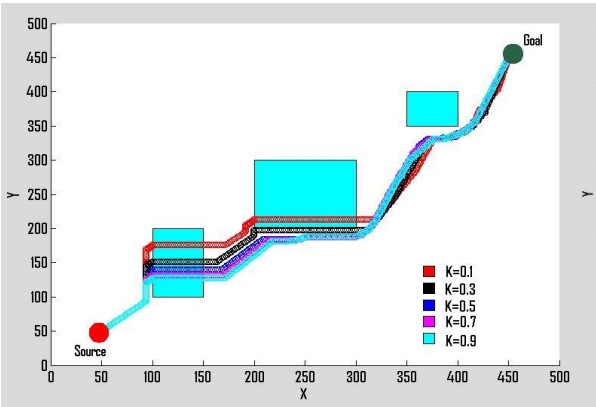


Fig.7.6 Robot paths for 'k' varying [0.1, 0.9]

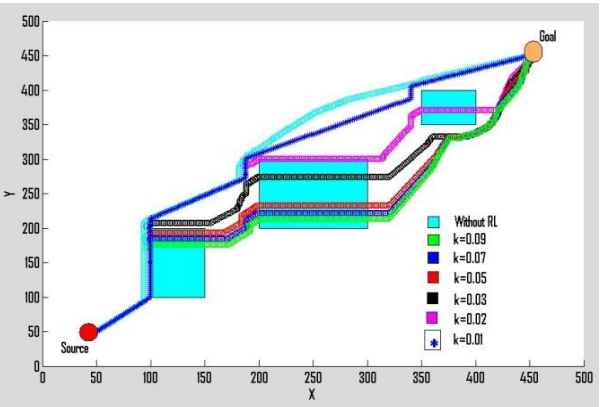


Fig.7.7 Robot paths for 'k' varying [0, 0.09]

Later experiments have been performed with the 'k' value varies from [0, 0.09] as shown in Fig.7.7. Robot path at k=0 is nothing but the path planning implementation without adaptive mechanism and the path is represented by cyan colour. The blue colored robot path is the optimal one and is generated at the 'k' value of 0.01.

7.5. Results & Discussion

From the developed motion planners, further movement of the robot is decided by finding the maximum $\overbrace{(\Gamma_r)_i}^{over\ all}$ from all considered antibodies. The mobile platform will perform its further action according to the selection of best robot action. Figs. 7.8 (a) - (d) represent the activation of suitable robot actions by different colours for various environmental situations. Figs.7.8 (a) & (b) represent the robot path when the workspace is with individual obstacles and Figs.7.8 (c) & (d) represent the robot path when it faces trap situation.

For scenario - I of Figs.7.8 (a), the mobile manipulator is moving from its source position (10, 275) to the target position (40, 300) using type-1 fitness based motion planner.

- Fig. 7.8 (a) - (i) represents the robot motion (in simulation mode) from source to target.
- Fig. 7.8 (a) - (ii) represents the robot motion (in real mode) from source to target.
- Fig. 7.8 (a) - (iii), Fig. 7.8 (a) - (iv), Fig. 7.8 (a) - (v) represents the robot positions during its navigation from source to target.

Similarly, for scenario - I of Figs.7.8 (b), the mobile manipulator is moving from its source position (10, 275) to the target position (40, 300) using type-2 fitness based motion planner.

- Fig. 7.8 (b) - (i) represents the robot motion (in simulation mode) from source to target.
- Fig. 7.8 (b) - (ii) represents the robot motion (in real mode) from source to target.
- Fig. 7.8 (b) - (iii), Fig. 7.8 (b) - (iv), Fig. 7.8 (b) - (v) represents the robot positions during its navigation from source to target.

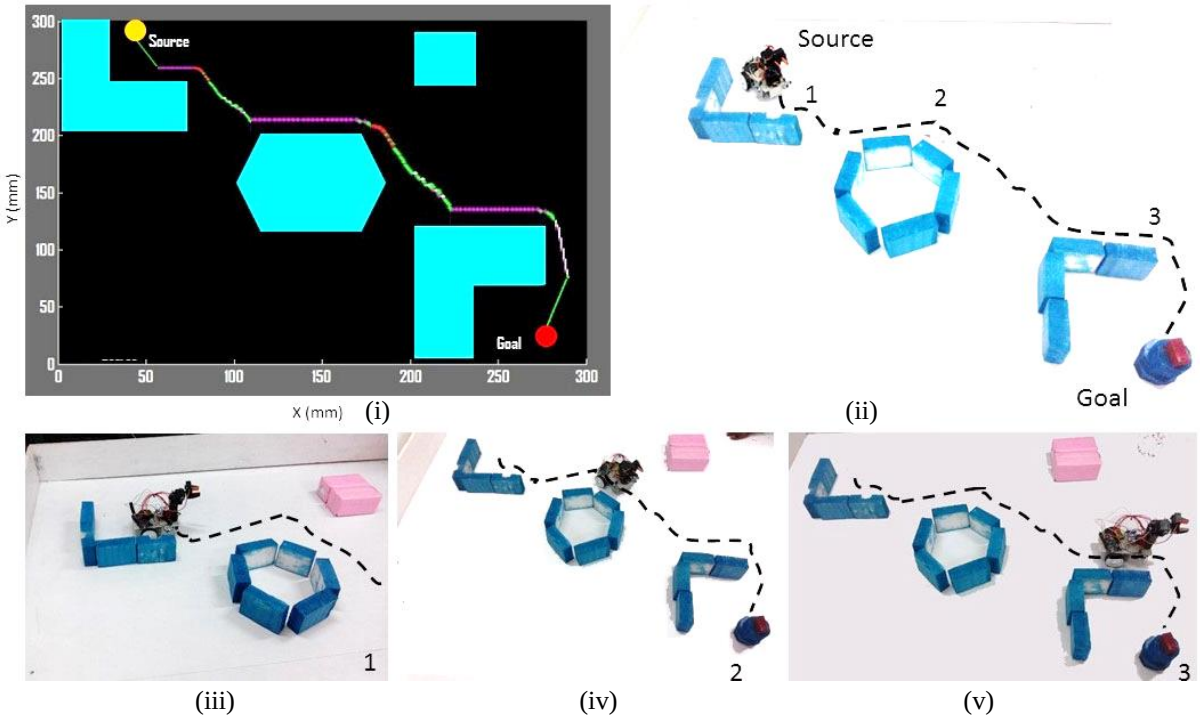
For scenario - II of Figs. 7.8 (c), the mobile manipulator is moving from its source position (200,200) to the target position (450, 200) using type-1 fitness based motion planner.

- Fig. 7.8 (c) - (i) represents the robot motion (in simulation mode) from source to target.
- Fig. 7.8 (c) - (ii) represents the robot motion (in real mode) from source to target.
- Fig. 7.8 (c) - (iii), Fig. 7.8 (c) - (iv), Fig. 7.8 (c) - (v) represents the robot positions during its navigation from source to target.

Similarly, for scenario - II of Figs. 7.8 (d), the mobile manipulator is moving from its source position (200,200) to the target position (450, 200) using type-2 fitness based motion planner.

- Fig. 7.8 (d) - (i) represents the robot motion (in simulation mode) from source to target.
- Fig. 7.8 (d) - (ii) represents the robot motion (in real mode) from source to target.
- Fig. 7.8 (d) - (iii), Fig. 7.8 (d) - (iv), Fig. 7.8 (d) - (v) represents the robot positions during its navigation from source to target.

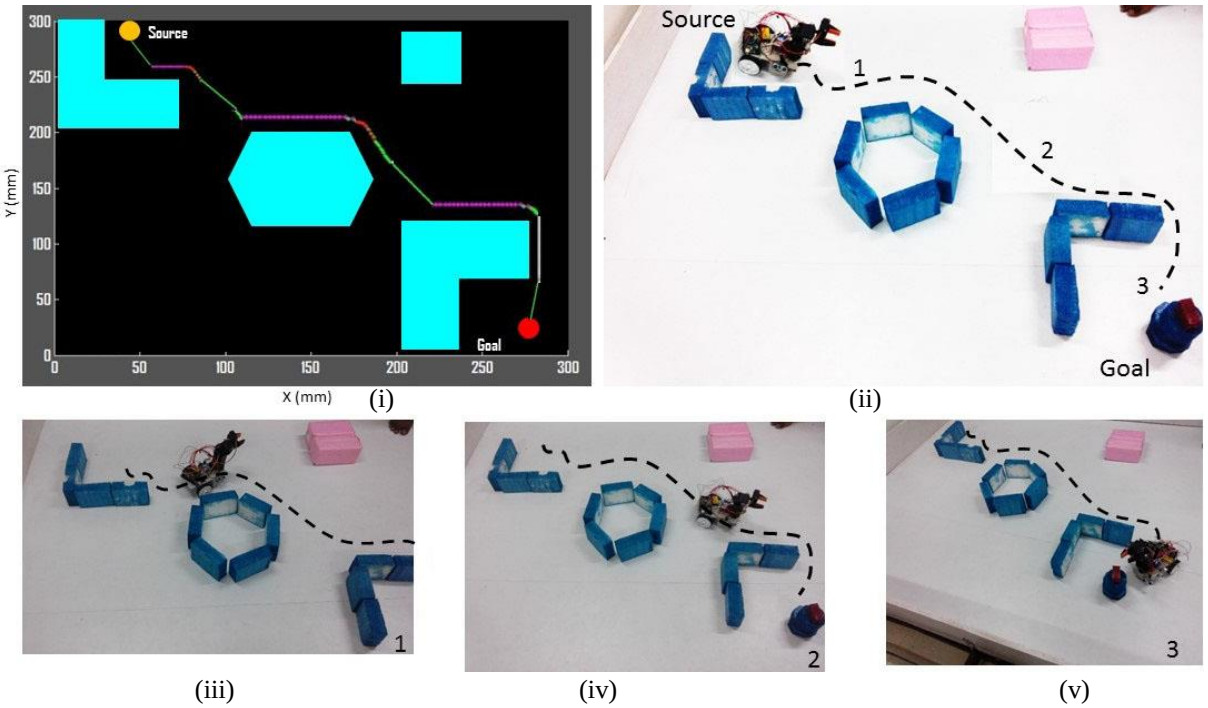
For same scenario-I & II, AIMP obtained the shortest paths as compared to IIMP. From the path analysis results as illustrated in Table 7.3, for scenario-I, AIMP maintains the travelled distance deviation of 3.8% form the simulation results and 4.5% form the experimental results. Similarly, for scenario-ii, AIMP maintains the travelled distance deviation of 4.2% form the simulation results and 4.1% form the experimental results. Path analysis results of scenarios- I & II showed that the robot motion in simulation environments are giving good agreement (% of error is below 5%) with the real time environments.



(i) (ii)

(iii) (iv) (v)

Fig. 7.8(a) path generated using IIMP (Scenario - I)



(i) (ii)

(iii) (iv) (v)

Fig. 7.8(b) path generated using AIMP (Scenario - I)

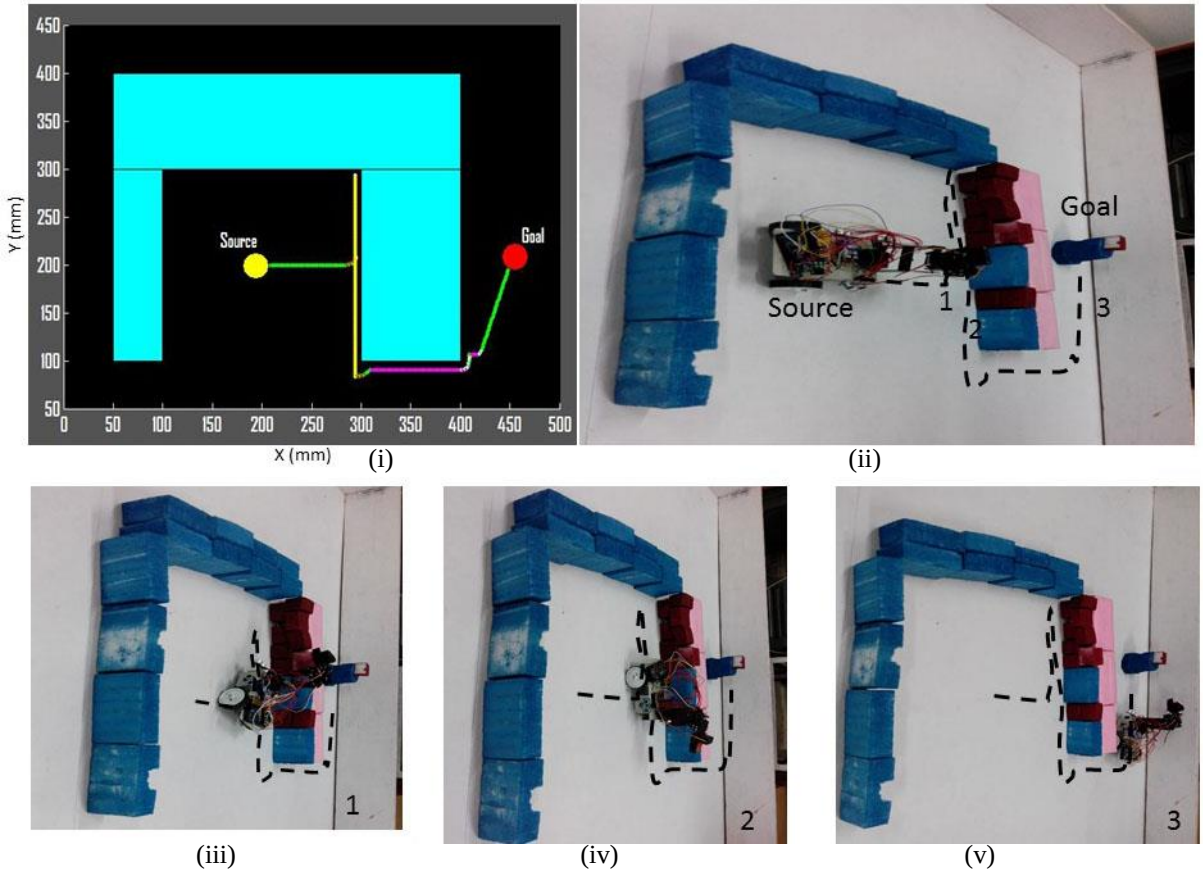


Fig. 7.8(c) Path generated using IIMP (Scenario - II)

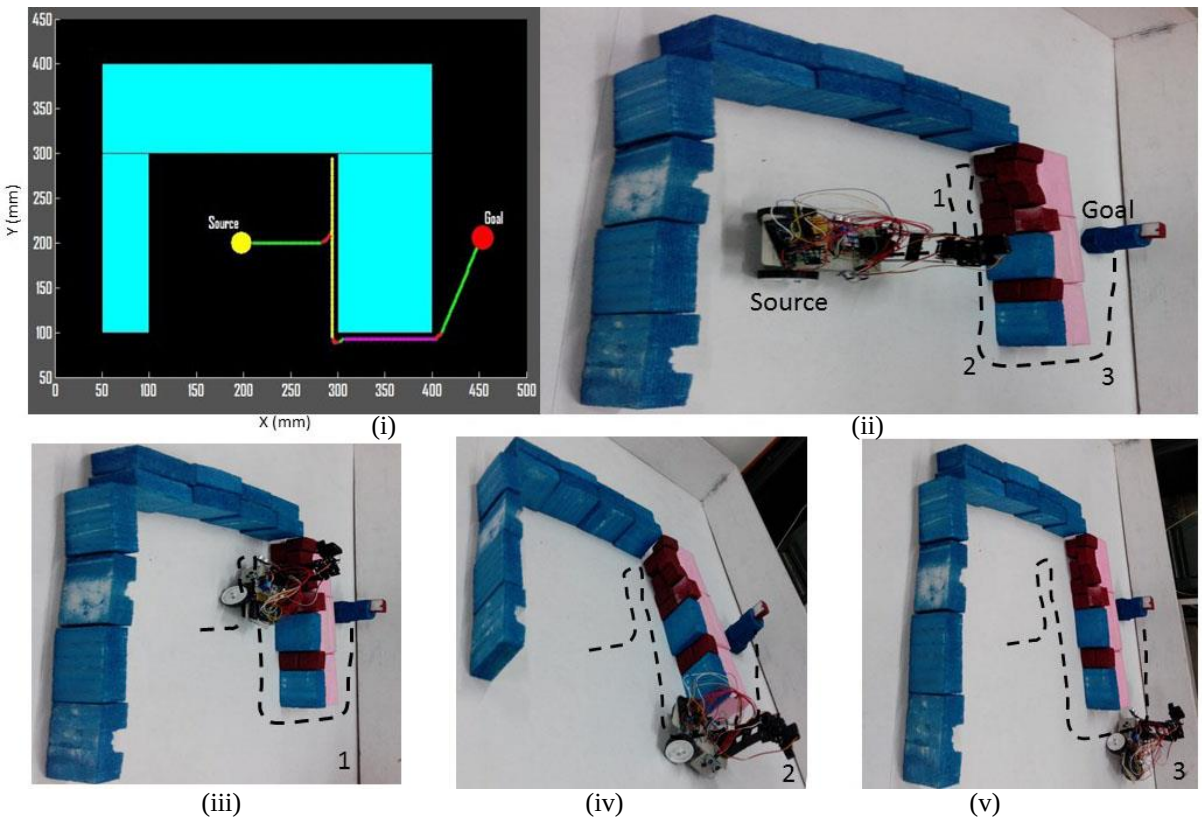


Fig. 7.8(d) Path generated using AIMP (Scenario - II)

By observing the path analysis results from Table 7.3, it is concluded that adaptive immune based motion planner is generating most favorable and optimal path as compared to path obtained by innate immune based motion planner.

Table 7.3 Path analysis results for Fig.7.8

Scenario	Motion planner	Simulation				Experimental				% of error
		No. of Iterations	Time Taken (sec.)	Distance travelled(cm)	% of path deviation	No. of Iterations	Time Taken (sec.)	Distance travelled(cm)	% of path deviation	
I	IIMP	369	147.6	442.8	3.8%	381	152.4	457.2	4.5%	3.1%
	AIMP	355	142	426		364	145.6	436.8		2.5%
II	IIMP	502	200.8	602.4	4.2%	523	183.6	627.6	4.1%	4.0%
	AIMP	481	192.4	577.2		502	199.6	602.4		4.1%

7.6. Comparison with Previous Work

Wahab [163] has dealt with neural network based intelligent control of a mobile robot to find its target within its environment. In their methodology two neural networks are used, the first neural network is used to find the free work space and the second neural network is used to navigate the robot towards its destination.

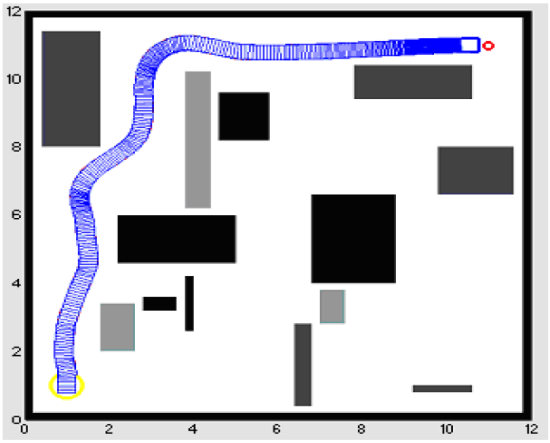


Fig.7.9(a) Path generated by Wahab [163]

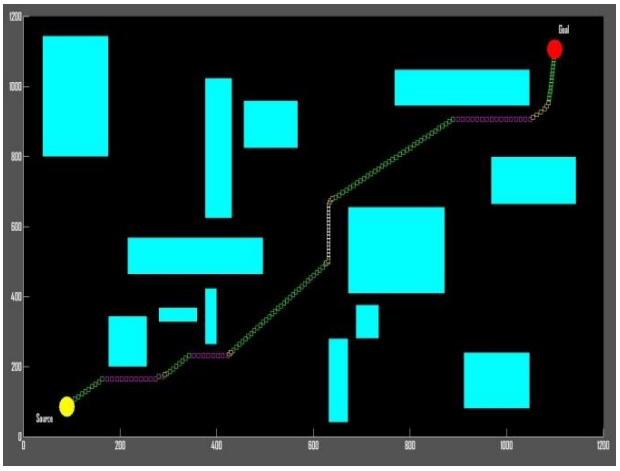
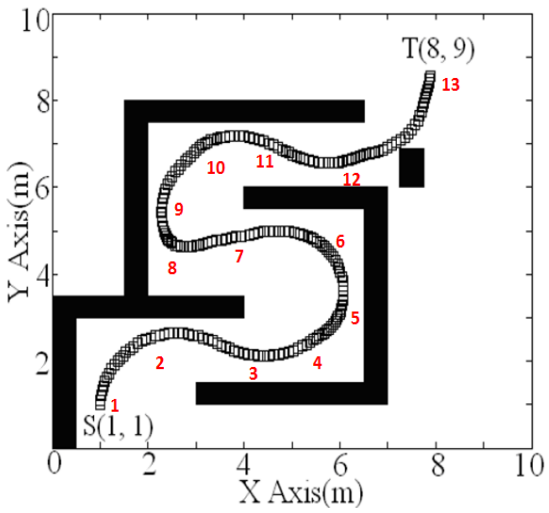


Fig.7.9(b) Path generated by AIMP

Yong *et. al.* [164] have introduced a behavior based architecture using fuzzy logic for mobile platform navigation in unknown environments. They designed four FLC based behaviors for mobile platform navigation, including target seeking, obstacle avoidance, tracking and deadlock disarming. Later they integrate these four behaviors to determine the control action of the mobile platform.

The path deviation by the mobile platform in Fig.7.10(a) at '7-8-9-10-11' is seems to be more robot travelled distance as compared to the robot travelled distance at '7-8-9' in Fig.7.10(b).



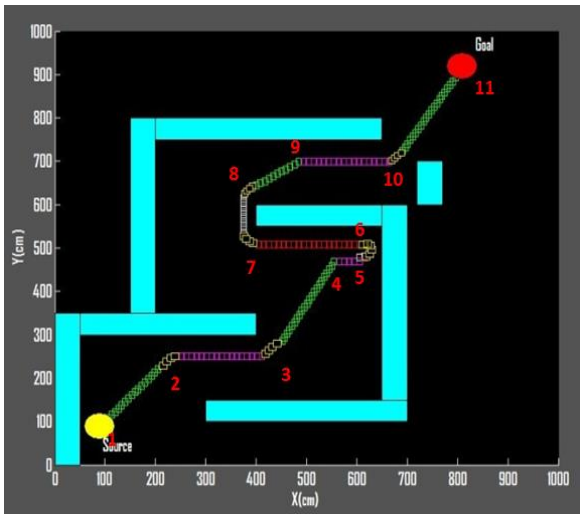
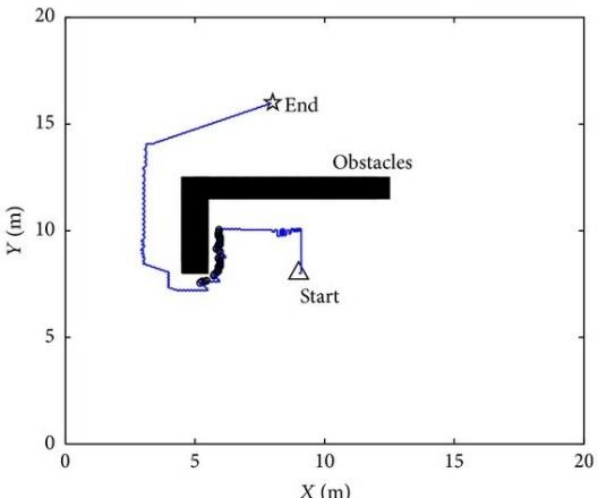


Fig.7.10(a) Robot Path by Yong *et. al.* [164]

Fig.7.10(b) Path generated by AIMP

Thus, polyclonal-based artificial immune network [165] has been proposed to solve the mobile robot navigation problem in complex unknown static environment. The developed immune based algorithm effectively overcomes the local minima problem.



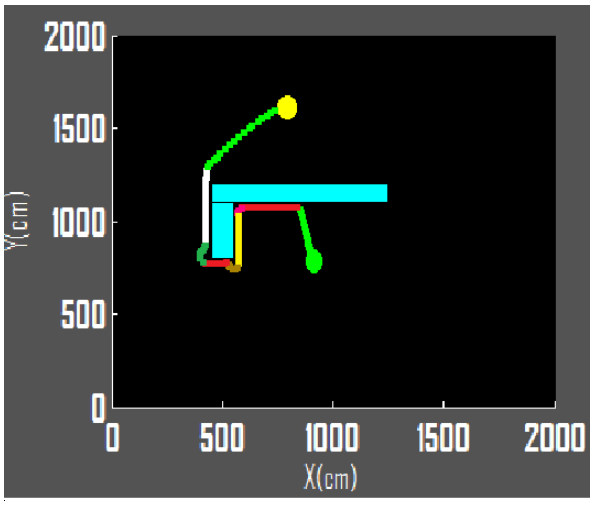


Fig.7.11 (a) Path by Deng *et. al.* [165]

Fig.7.11 (b) Path generated by AIMP

Table 7.4 Path analysis results for Fig.7.9 – 7.11

Previous methodology	**Distance travelled(m)**		**% of path deviation**
	Previous methodology	**Present methodology**	
neural network [163]	17.2	16.0	3.3%
Behavior based architecture [164]	15.7	14.8	6%
Artificial immune network [165]	29.1	25.2	2.5%

By observing the path analysis results from Figs. 7.9 (b), 7.10 (b) & 7.11 (b), the proposed motion planner is generating shortest paths as compared to path obtained by Wahab [163], Yong *et. al.*[164] and Dang *et. al.* [165] with a path deviation of 3.3%, 6% and 2.5% respectively.

7.7. Summary

Two efficient immune based motion planners have been introduced for solving mobile robot path planning problem in unknown environments. The first motion planner called innate immune based motion planner, is working only on the basis of the parameter 'Learning Rate' and is not getting any global information from the system. For the second motion planner, an efficient adaptive learning mechanism has been presented and integrated to the developed innate immune based motion planner. Path analysis results showed that, both motion planners are generating collision free paths in their robotic search space, but the second motion planner called adaptive immune based motion planner is obtaining shortest path than the path obtained by the innate immune based system architecture (Table 7.3). Deviations are found to be within 5% during comparisons between simulation and corresponding experimental results.

Chapter 8

EXPERIMENTAL ANALYSIS

8. EXPERIMENTAL ANALYSIS

This chapter deals with the experimental analysis of mobile manipulator to control its motion using the techniques described in the previous chapters. Manipulator design and its control system architecture are analyzed then various mechatronic components have been integrated to the manipulator to perform desired tasks.

8.1. Manipulator Design

In Chapter – 3, the motion of the robot arm structure is evaluated by performing forward and inverse kinematic analyses. In this investigation a 4-axis manipulator is considered and its specifications are illustrated in Table.3.1.

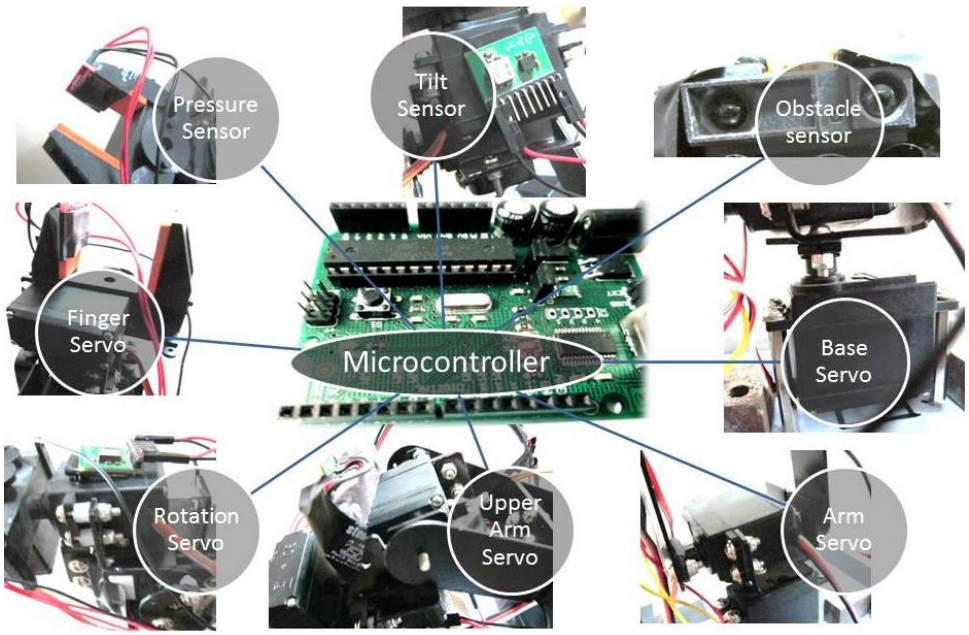


Fig.8.1 Experimental setup of the robot manipulator

To perform the various tasks, the robot arm is interfaced with several components as represented in Fig.8.1. A brief description about each component of the structure is given as follows:

→ *Microcontroller:* AT89C52 type microcontroller has been used in this investigation. A detailed description about the pin configuration of the considered microcontroller is given in Appendix – C.

→ *Servo motor:* The servo motor has three pins: VCC, GND and PWM and these are connected to 6 V DC power supply, ground and to the microcontroller respectively. The rotation direction of each servo is controlled with the help of the pulse duration of the generated PWM. If the supplied pulse duration is of 1.5ms then the servo in 90 deg position. For clock-wise rotation generated PWM in the range of 1.5ms - 2.0ms; and for

anti-clockwise rotation supplied PWM should be in the range of 1.0ms - 1.5ms. Mostly pin 0 of port 1 is used for the PWM generation. The 4-axis robotic manipulator is represented in Fig.8.2.

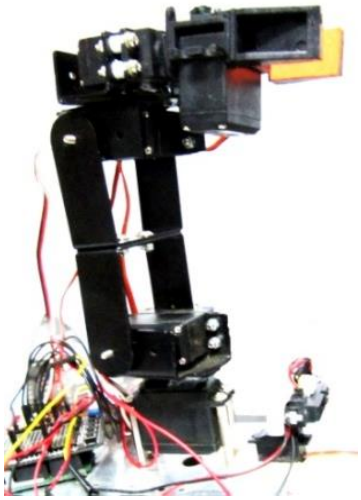


Fig.8.2 4-axis robot arm to perform experimental analysis

As per the requirement of the rotation angle, the microcontroller will execute the particular PWM generation code block and will generate the corresponding PWM. Pin 0 of port 1 should be connected to the PWM pin of the servo motor. Generated pulse from the microcontroller will control the rotation angle of the servo. By controlling each of the servo motor (at 4 different axes), we can control the motion of the end-effecter which is connected to the manipulator. Fig.8.3 shows the interfacing of arm servo motors to a microcontroller.

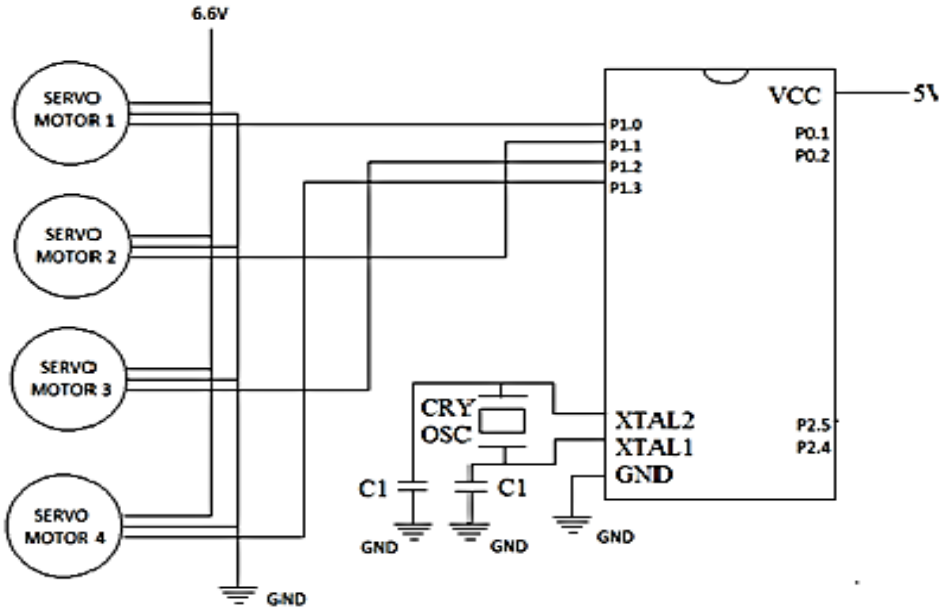


Fig.8.3 Circuit diagram of 4-axis robot arm servos interfacing with microcontroller

A pre-programmed pick & place task is performed by the developed robot arm is represented in Fig.8.4. The initial and final positions of the end-effector is given to robot and it will move towards the object according to the rotation of each servo which decided according to the inverse kinematic model as explained in section 3.3.

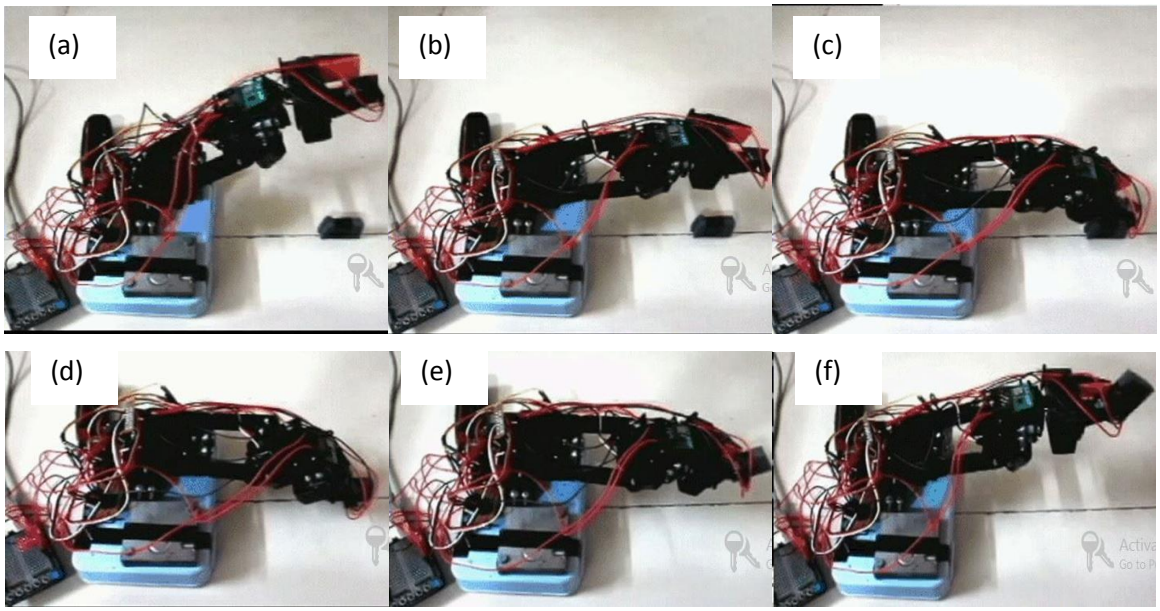


Fig.8.4 Pick & place task performing by robot

→ *Ultrasonic sensors:* It consists of an ultrasonic transmitter & receiver which are made up of piezoelectric sensors. Transmitter generates an ultrasonic wave when it receives a square pulse of 40 KHz from the microcontroller and the receiver generates an AC pulse when it receives an ultrasonic wave reflected back from the obstacle. That pulse is generated by microcontroller and feed directly to the transmitter. The square pulse is generated by switching the voltage status of the second pin of the microcontroller with time duration of 12.5 µs. After 12.5 µs time interval internal code will swap the pin status of the microcontroller. The process will go on for infinite. By using any infinite loop structure like 'while (1)'or 'for (; ;)' we can perform the task. Fig.8.7 represents the object detection while robot is in operation condition. Fig. 8.6 represents the interfacing of ultrasonic sensor to the microcontroller to detect the object within the robot workspace.

(a) (b)

Fig.8.5 (a) ultrasonic receiver & transmitter; (b) Distance measurement in volts

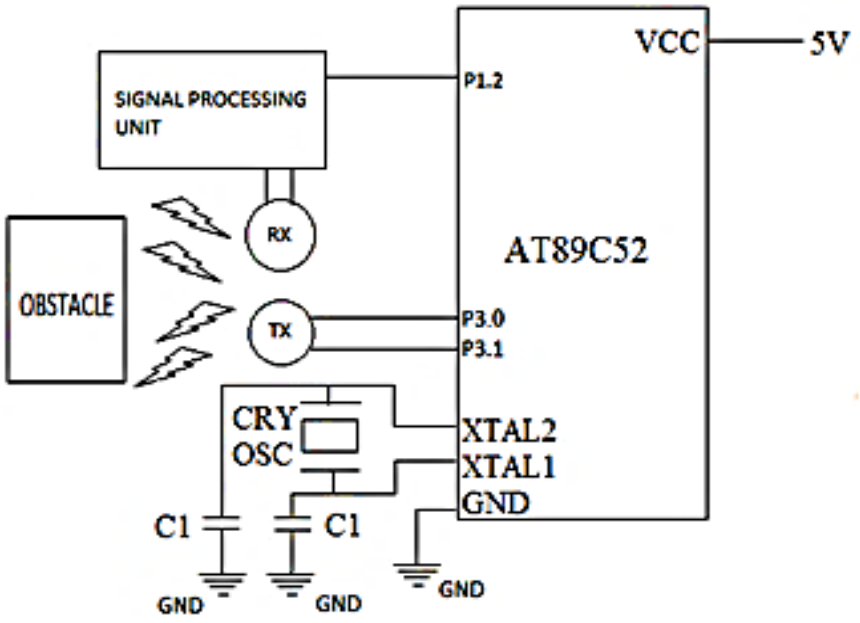


Fig.8.6 Schematic diagram of ultrasonic sensor interfacing with microcontroller

The transmitted signal will be transmitted through the air and finally reflected by the target object at the other end. Due to the noise in the air, the transmitted signal will be distorted and gets attenuated. The received signal will be amplified with the help of an op-amp (IC-LM741). After receiving the transmitted wave it will pass the signal to signal processing unit which helps in regenerating the original signal. The signal processing unit consists of 2 major blocks: amplifying unit and rectifying unit. Once the received signal gets amplified, then the signal will be rectified by a bridge rectifier.

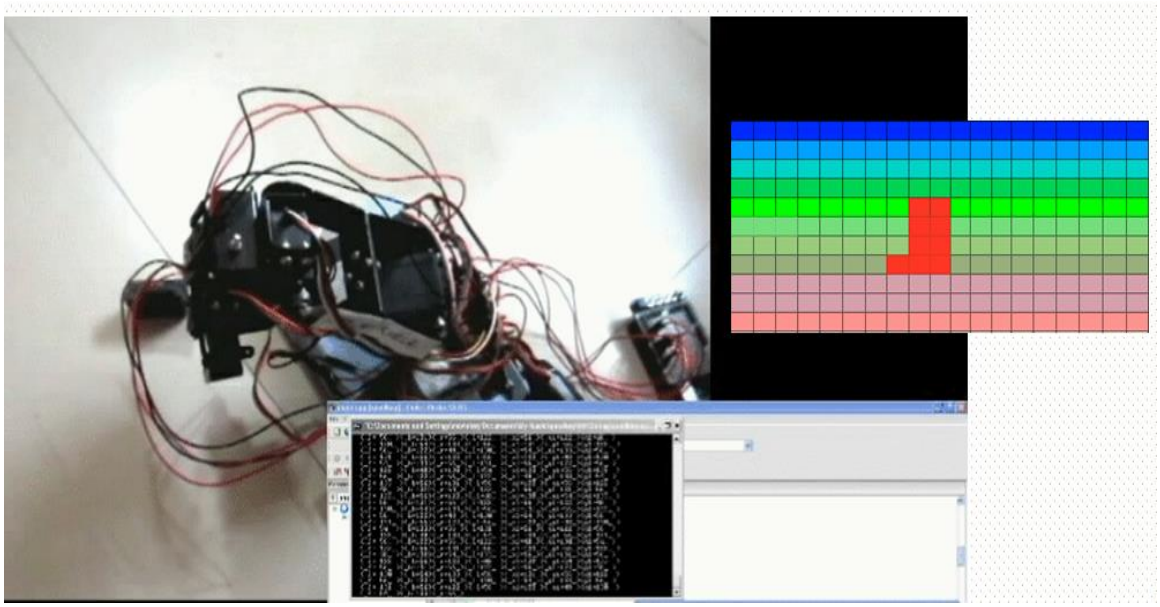


Fig.8.7 Virtual image generation by object detector mounted on end effector

→ *Tactile switch:* Tactile (LTS) Sensor is interfaced to a microcontroller (AT89S52). Terminal 1 of LTS is connected to a 5V DC supply and terminal 3 to the microcontroller (pin 0 of port 1) as shown in Fig. 8.8. If the applied pressure is not enough, the (LTS) sensor will be in open condition. After receiving per-defined pressure, it will close the internal connection. This sensor gives the information about the object grasping by the end-effector.

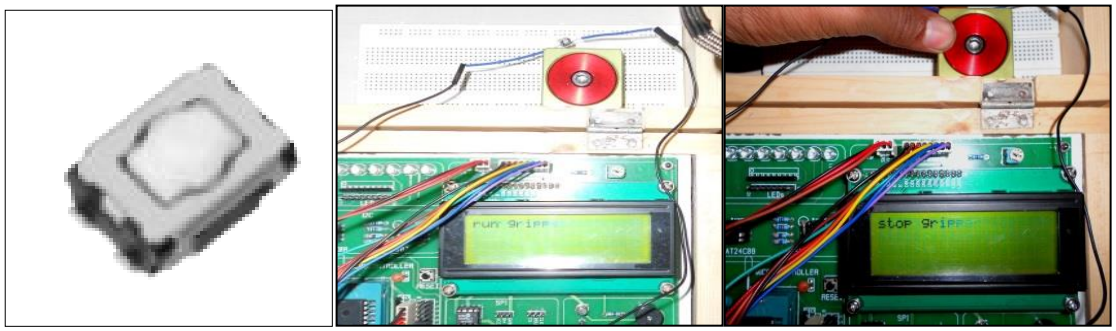


Fig.8.8 Pressure application & verification by LTS

LTS interfacing with Microcontroller is shown in Fig.8.9 and it will sense the pin status continuously. If the pin status is 0 V or at logic zero, it indicates enough pressure not yet applied and it will run the gripper for grasping object. Once the pin status will jump to logic 1 or 5 VDC, it indicates enough pressure has been applied and the microcontroller will stop the gripper to maintain its current form. The above operation can be performed by a simple logical program.

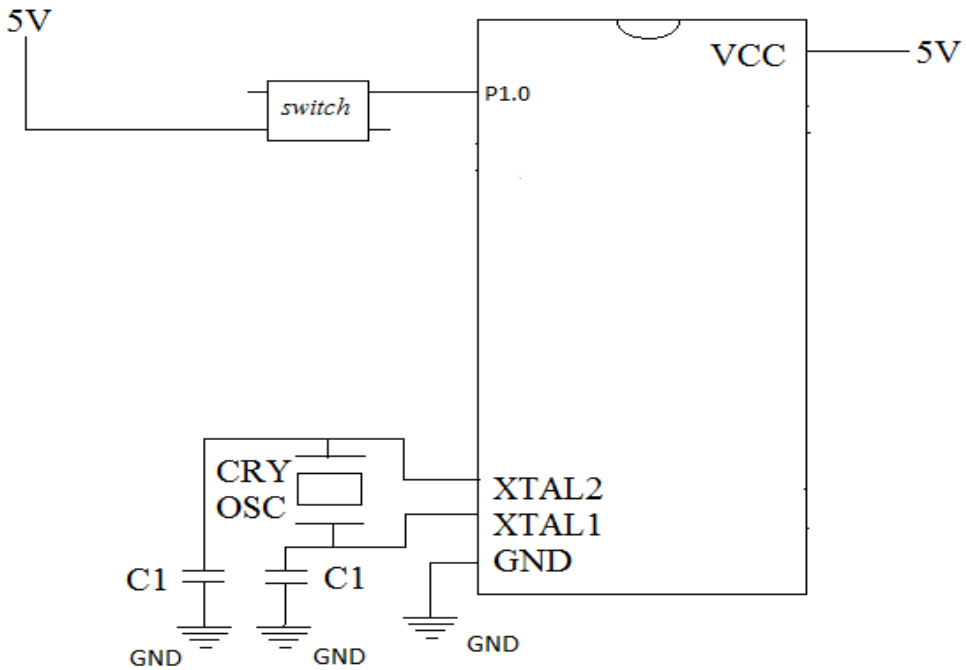


Fig.8.9 LTS interfacing with microcontroller

→ *Tilt Sensor:* When the robot is carrying hazardous materials and moving in the uneven terrain manufacturing environments, the loaded material may be unstable. In order to maintain the material in specific orientation it is required to measure the slope of the rough inclination. So, here tilt sensor is used to measure the tilting of an object which is carried by the robot's end-effector as shown in Fig.8.10.

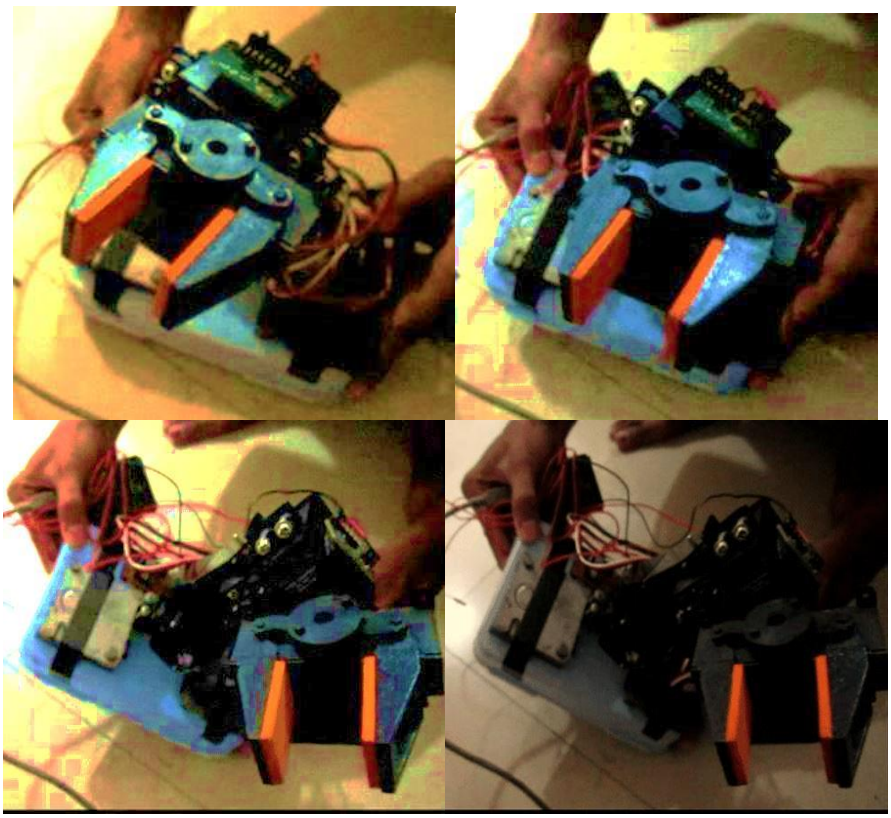


Fig.8.10 End-effector maintaining horizontal orientation at various slopes

8.2. Trajectory Generation by the End-Effector

To perform various tasks by the robot arm end effector has to move within its search space while satisfying all kinematic relations. The below example here is considered when all the joints (servos) are moving with constant speed of 100rpm. The trajectory generated by the end-effector is represented in Fig.8.11.

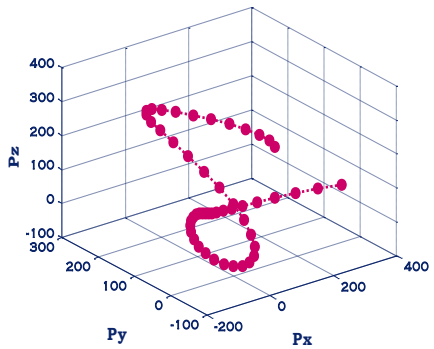


Fig.8.11 End-effector's trajectory when all the joints are rotating with uniform velocity

In order to validate the simulation result, the real robot is programmed in the way that all the servos are at the constant speed of 100rpm. Continuous trajectory generation by the robot arm is shown in Fig.8.12.

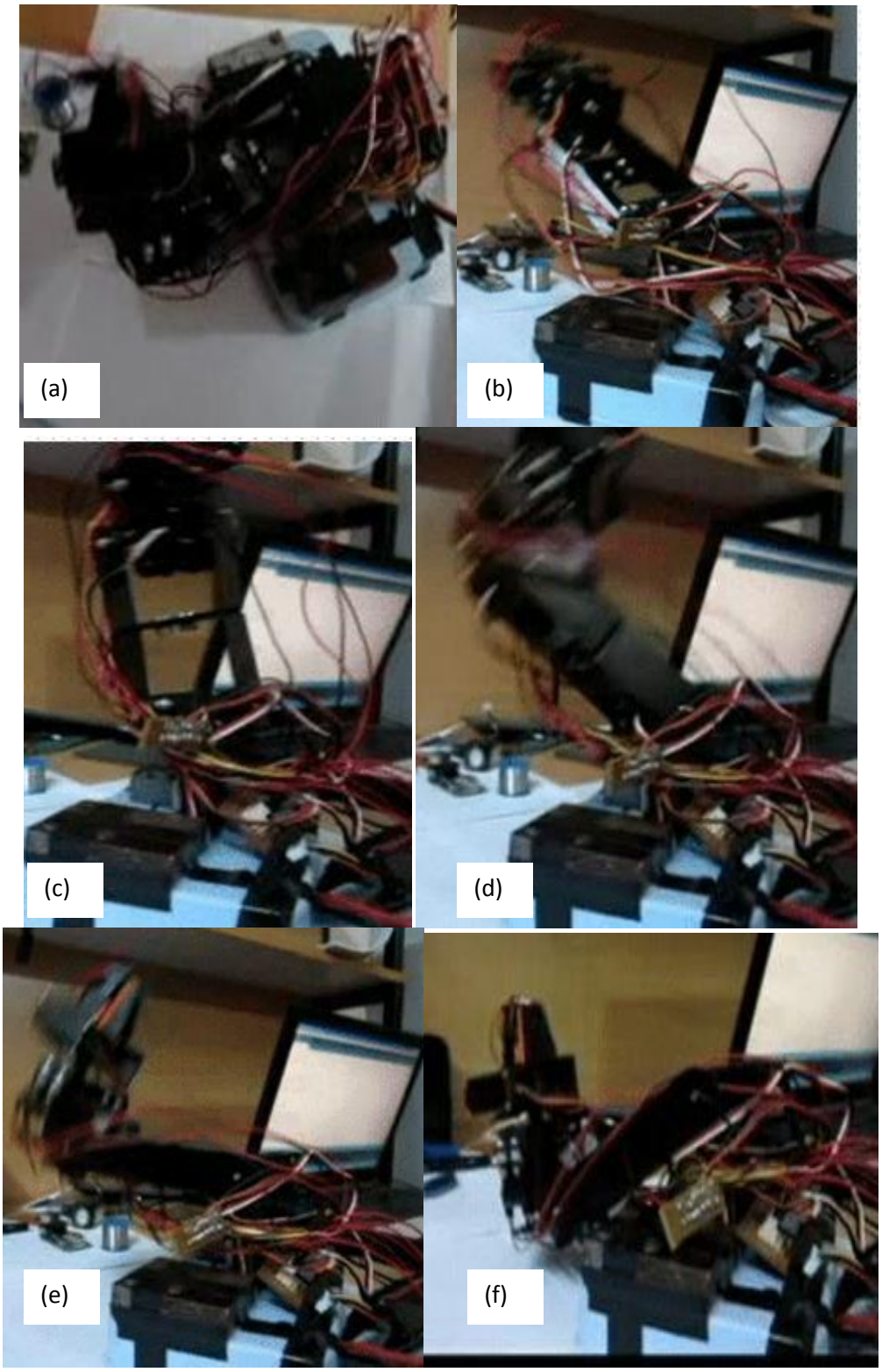


Fig.8.12 Sequential steps followed by the End-effector's with respect to Fig.8.11

8.3. Mobile Platform Design

In Chapter – 4, the locomotion of the mobile structure is evaluated and it is concluded that the differential mobile platform is suitable because of its simple structure. To give the motion to entire robot system, the mobile platform is equipped with two independent motorized fixed standard wheels and non-motorized caster wheel. The assemble structure of the mobile platform is as follows:

→ *Platform Chassis:* This is made up of Aluminum material. All the accessories components are equipped to this chassis. The arm structure is mounted on this platform.

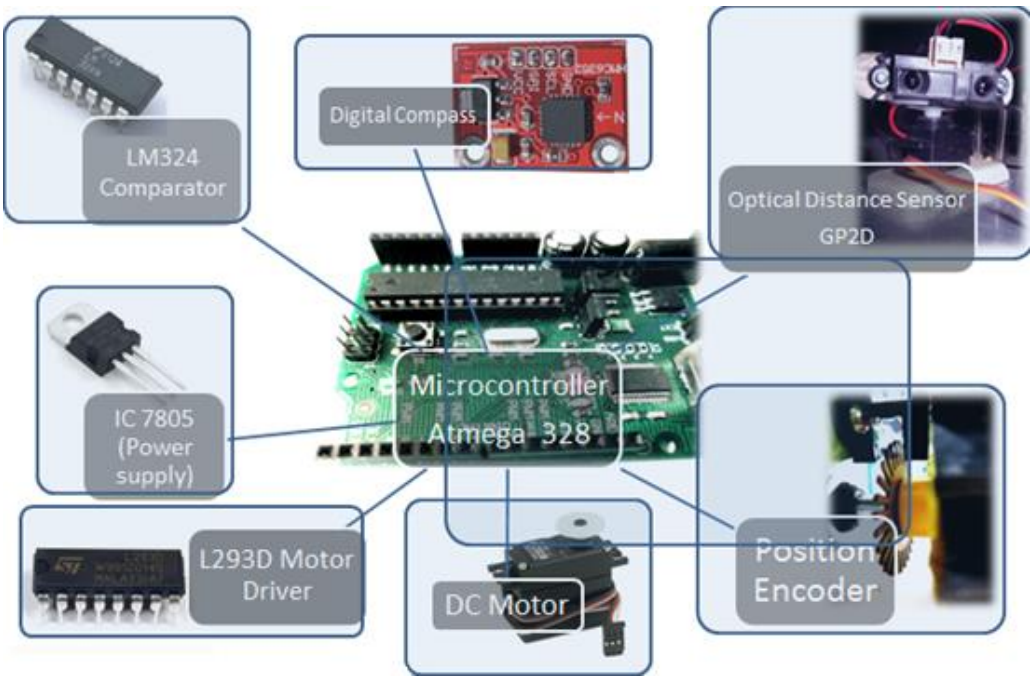


Fig.8.13 Experimental setup of the mobile platform

→ *Microcontroller:* a similar type of microcontroller as explained in the section 8.3 is used for controlling all electronic components those are equipped to the mobile platform.

→ *Ultrasonic Sensor:* a similar type of ultrasonic sensor as explained in the section 8.3 is used for finding obstacles within the robotic search space.

→ *DC motor:* DC motors are controlled by the microcontroller pins through the motor driver IC (L293D). Two pins of the DC motors are for +ve and –ve supply. By changing the polarity of the supply voltage we can change the direction of rotation. This technique is used to control the direction of the DC motors installed in the mobile platform. In order to reverse the direction of the DC motor of the mobile platform we simply swap the status of the two pins those are connected to the DC motor. The motors are not directly connected to the microcontroller pin. Current rating of the individual microcontroller pin is not enough to drive a motor. L293D is a motor driver chip is used between the

microcontroller and the DC motor. It amplifies the output pin current so that it can drive the motor attached with the microcontroller.

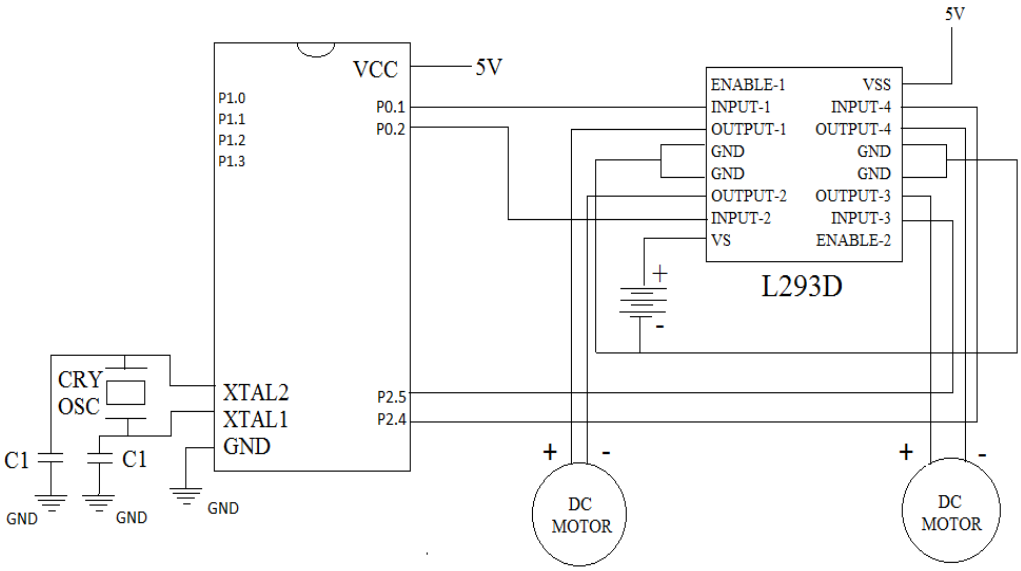


Fig.8.14 Microcontroller interfacing with two DC motors & Motor driver

→ *Position Encoder:* It is an electro-mechanical device that converts the angular position or motion of a rotary element to an analog or digital code. With the help of this device the wheel velocities can be measured as shown in Fig.8.14. In this study two position encoders are used to measure the velocities of two fixed standard wheels, thereby the velocity of the entire robot system can be controlled when it is in operation.

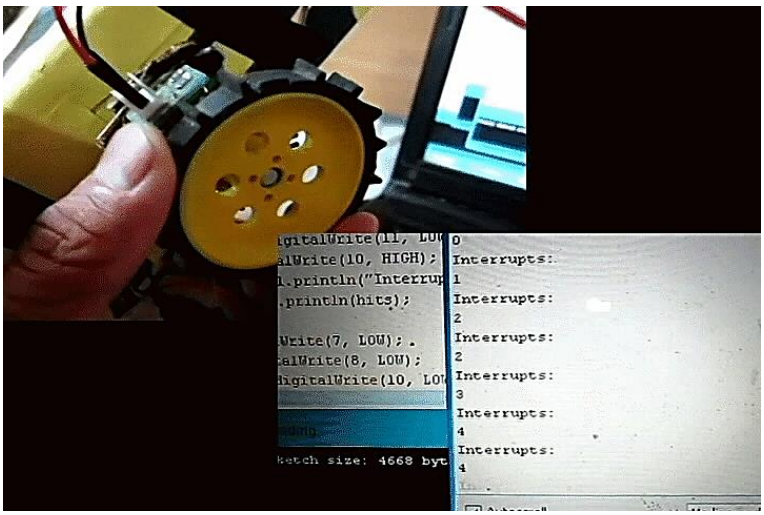


Fig.8.15 Angular velocity measurement of DC motors of the mobile platform

→ *Digital Compass:* It is used to measure the tilt angle by the robot chassis. When the robot is moving in its workspace, it directs toward its target position while avoiding obstacles. The direction oriented by the mobile platform is measured with the help of Digital compass as shown in Fig.8.15.

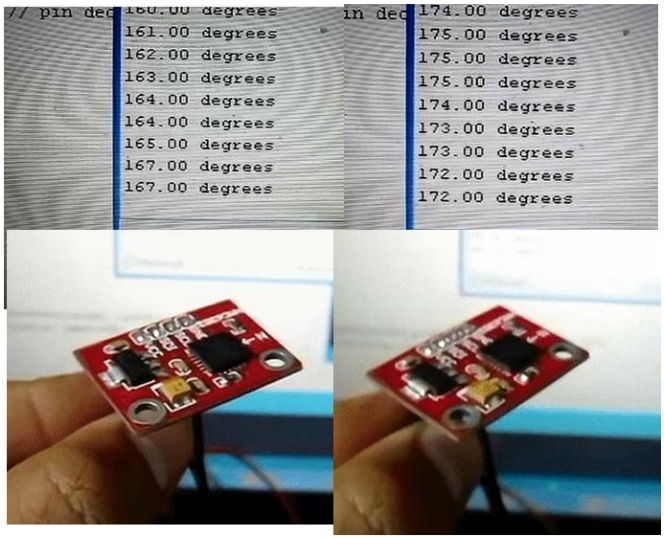


Fig.8.16 Measurement of the direction of the robot system using digital compass

8.4. Robot Motion Control

In order to validate the theoretical results, experimental analysis has been carried out for a Mobile manipulator with differential platform. The final configuration the developed mobile manipulator is shown in Fig.8.17.

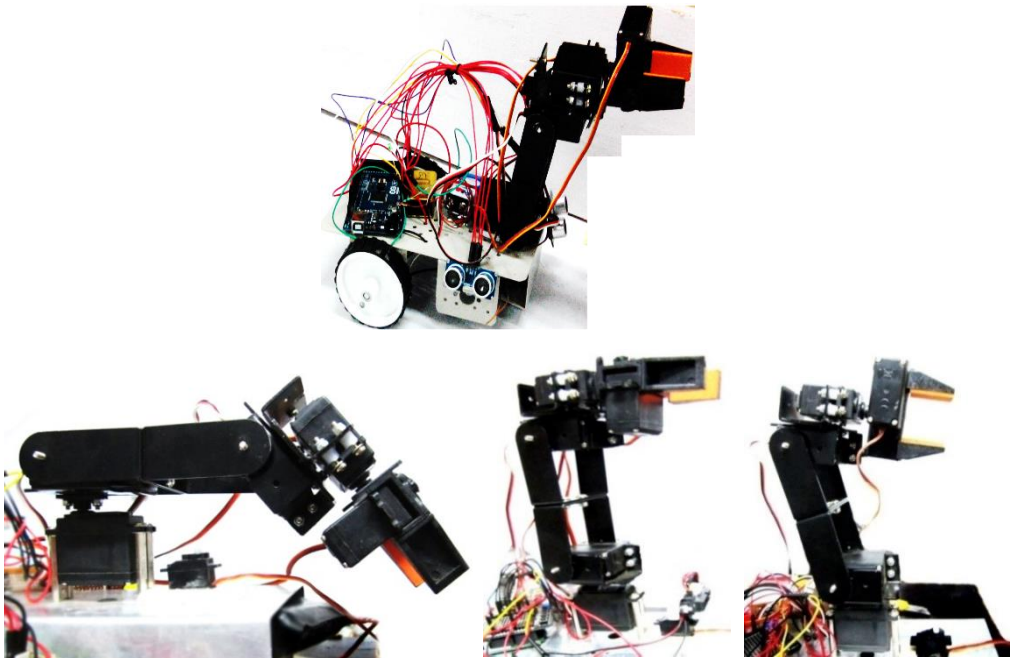


Fig. 8.17 Developed manipulator & various end effector orientations

In the current research work four path planners have been developed as discussed in chapters 6 & 7. First two are swarm intelligence based and the remaining two are immune inspired motion planners. From the two swarm based motion planners, first fitness based motion planner is giving better solution in the same way from the developed immune based motion planners AIMP is generating shortest paths as compared to IIMP.

8.4.1. Comparison between the developed methodologies

This section presents the simulation and experimental results of the AIMP as compared to the first fitness function based PSO motion planner, since the first fitness function gives better results than second fitness function.

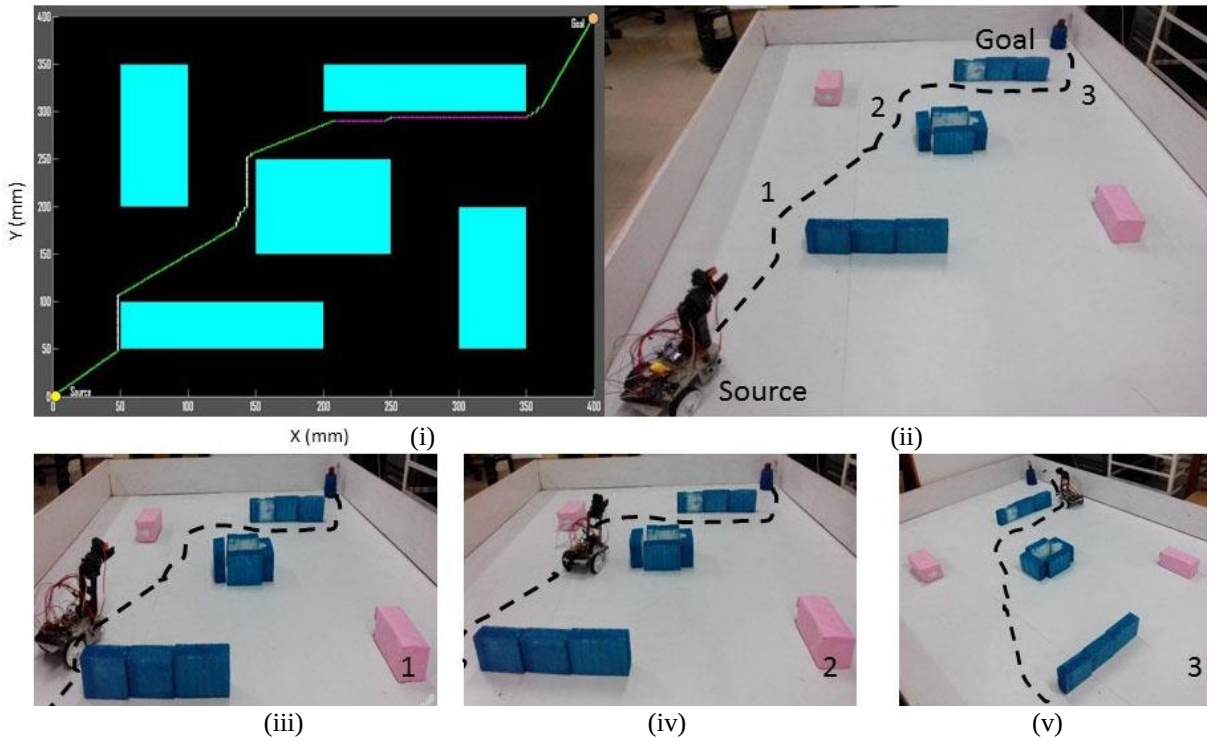


Fig. 8.18(a) Robot path by AIMP (Scenario - I)

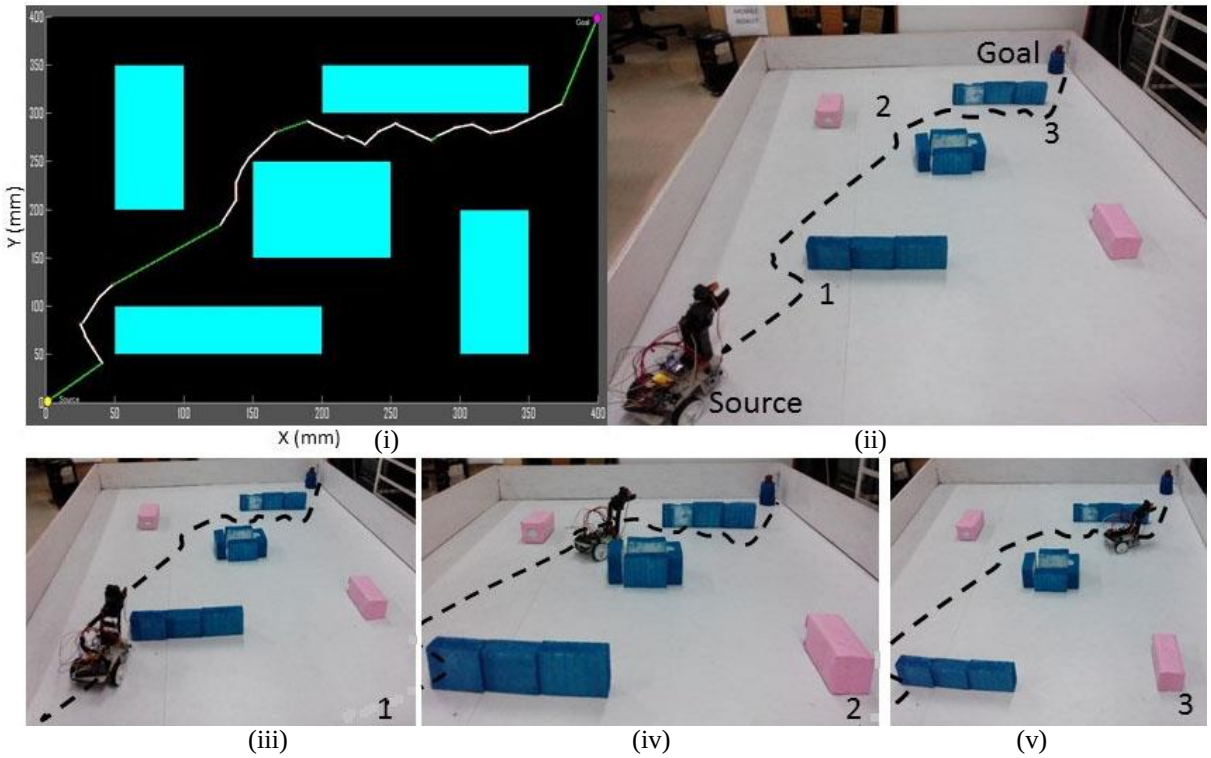


Fig. 8.18(b) Robot path by PSO motion planner (Scenario - I)

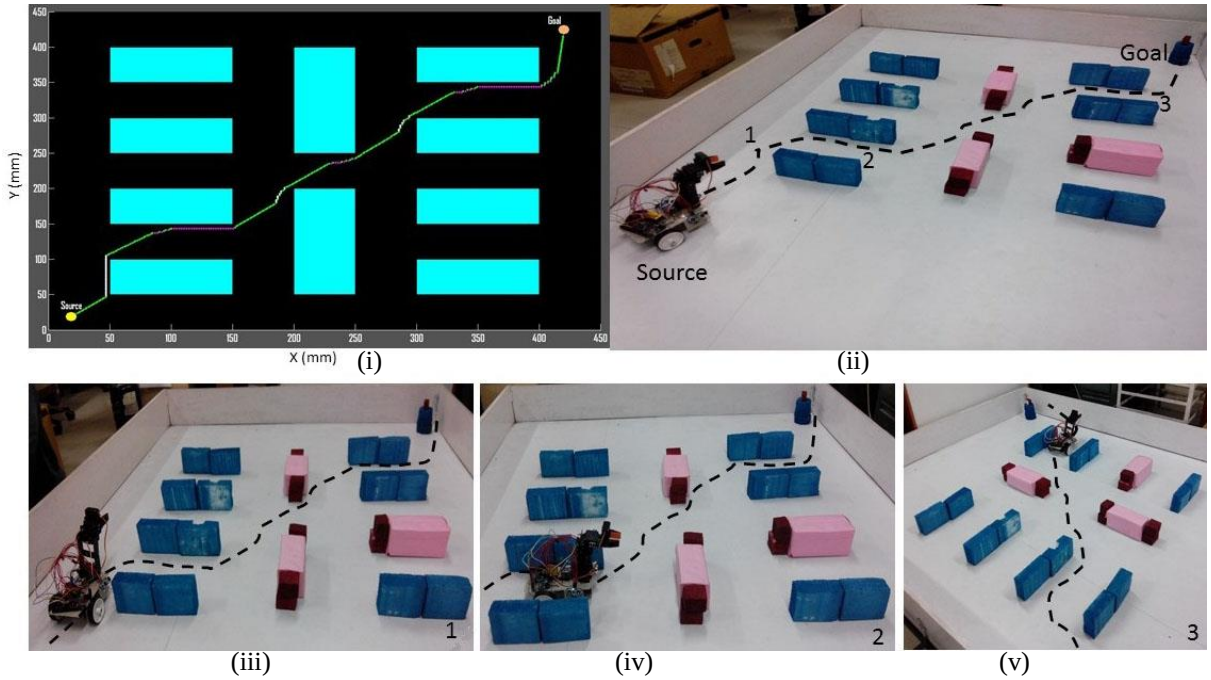


Fig. 8.19(a) Robot path by AIMP (Scenario - II)

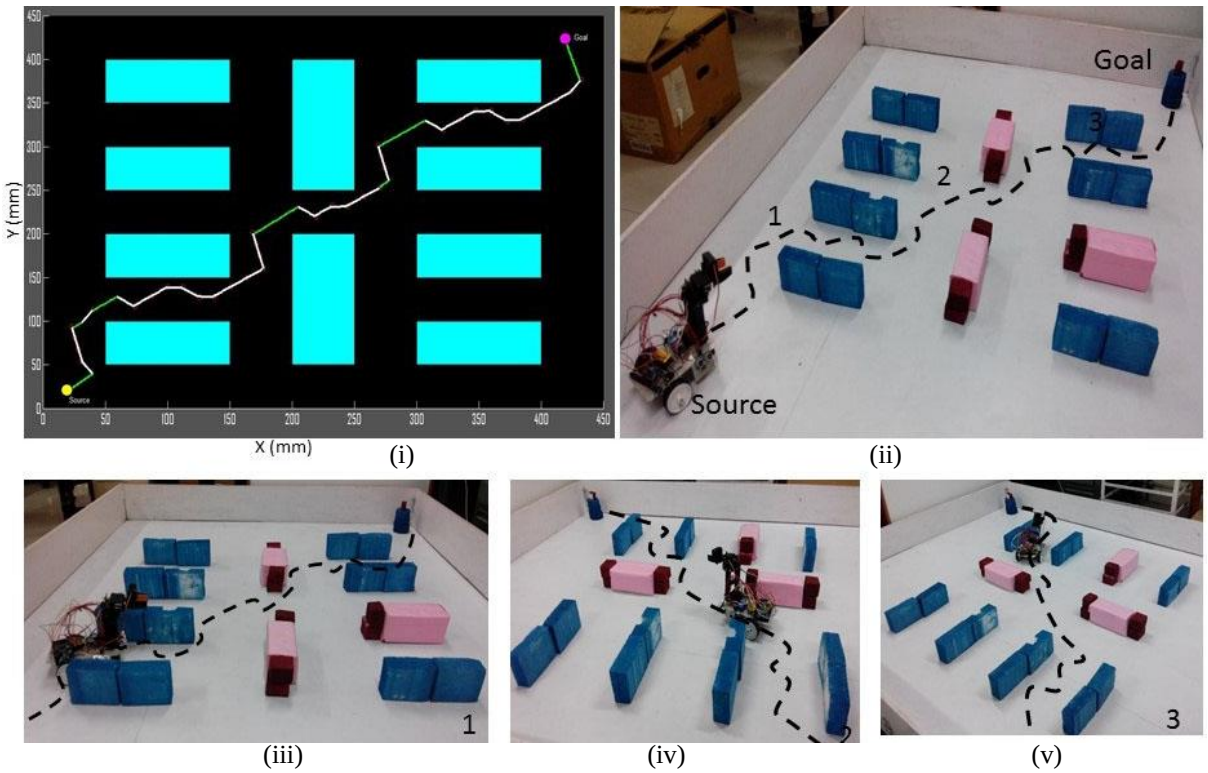
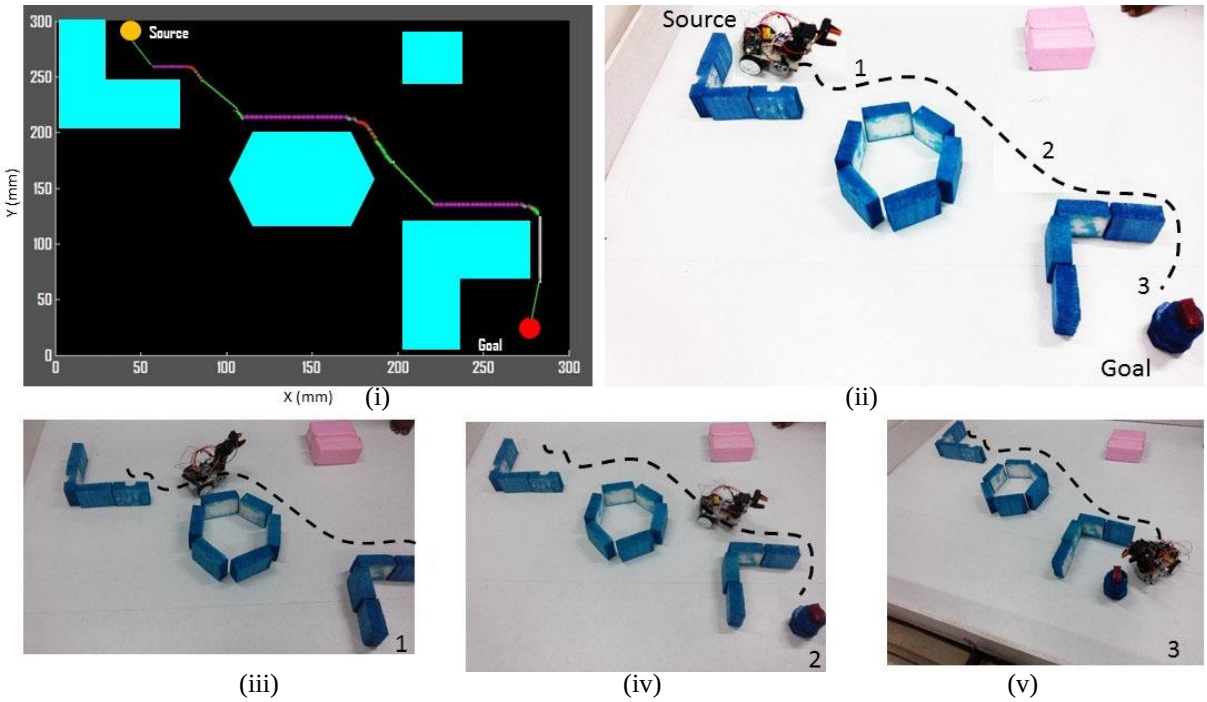


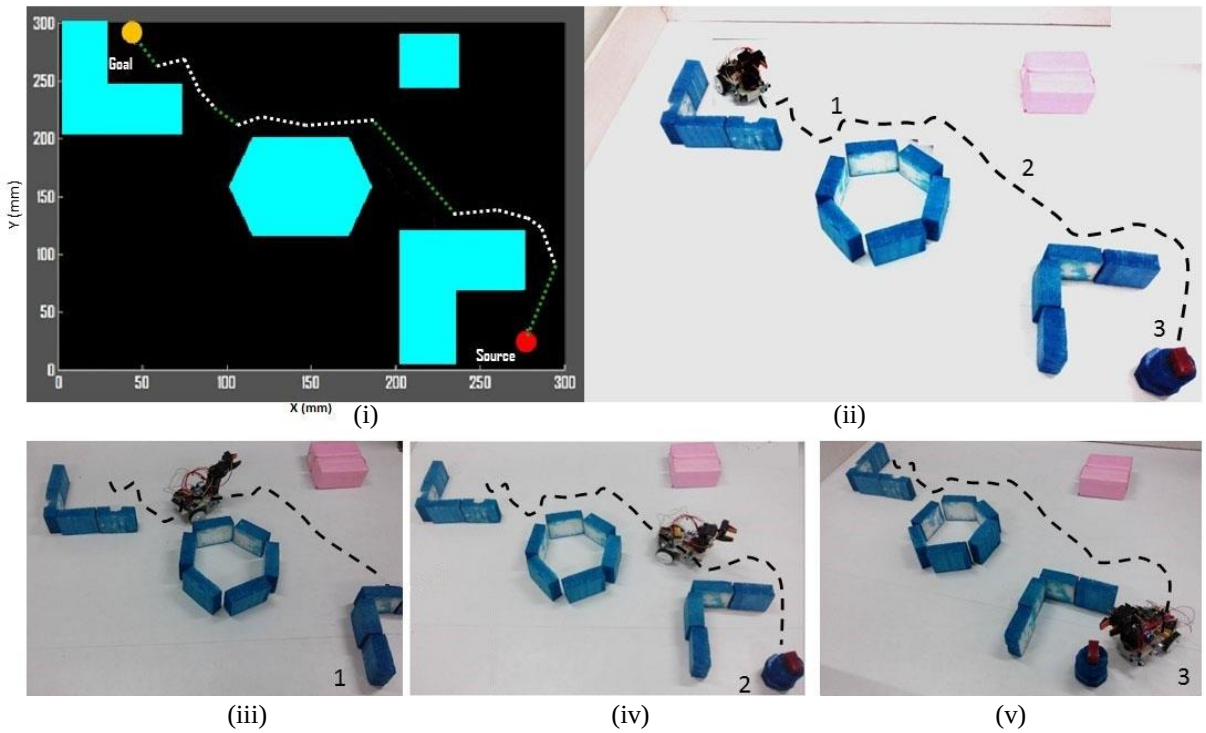
Fig. 8.19(b) Robot path by PSO motion planner (Scenario - II)



(i) (ii)

(iii) (iv) (v)

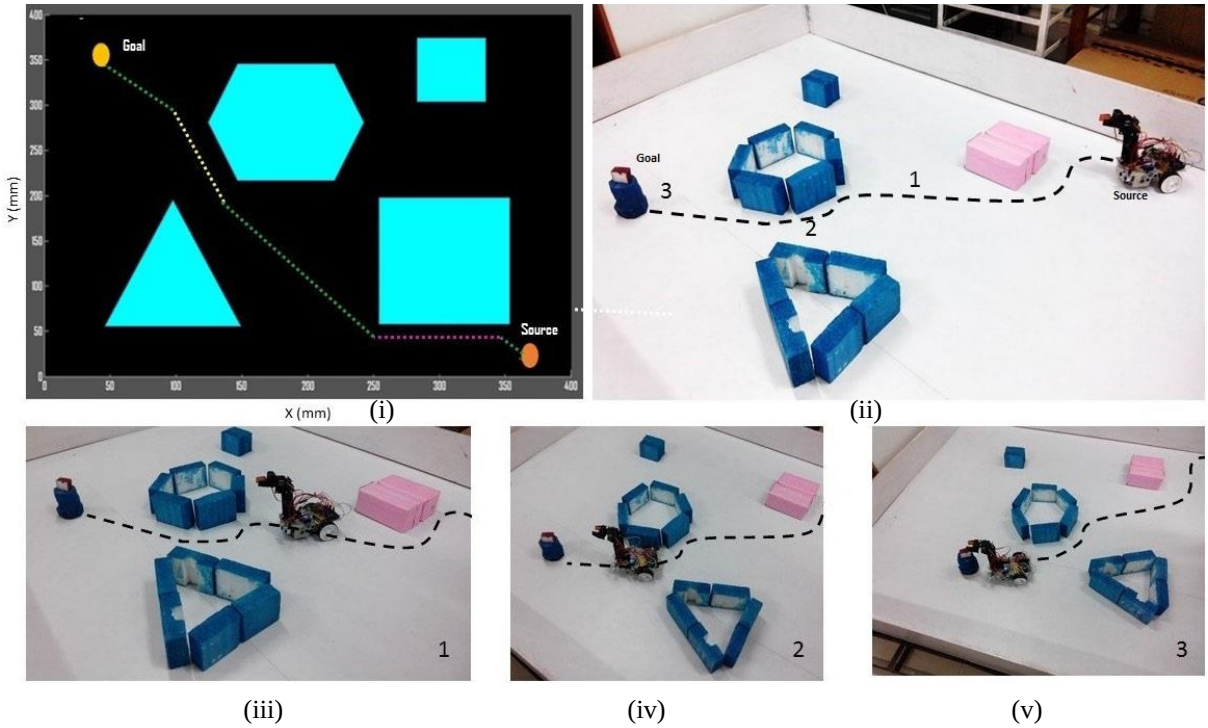
Fig. 8.20(a) Robot path by AIMP (Scenario - III)



(i) (ii)

(iii) (iv) (v)

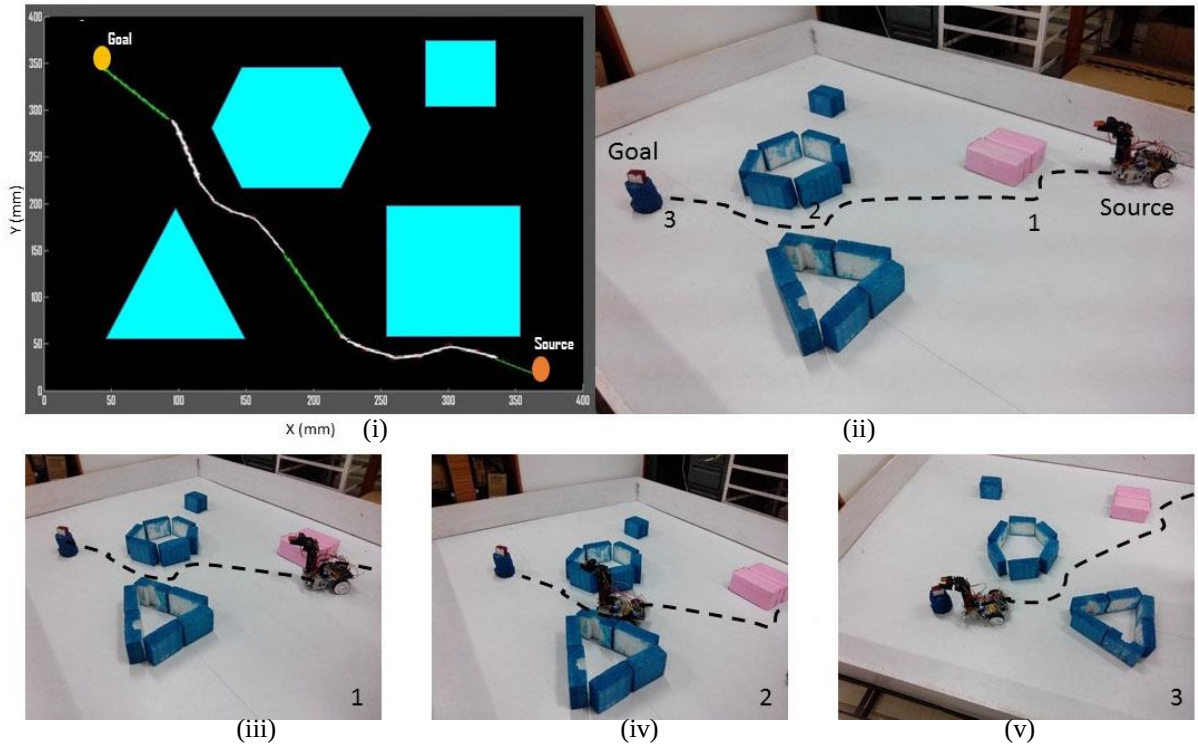
Fig. 8.20(b) Robot path by PSO motion planner (Scenario - III)



(i) (ii)

(iii) (iv) (v)

Fig. 8.21(a) Robot path by AIMP (Scenario - IV)



(i) (ii)

(iii) (iv) (v)

Fig. 8.21(b) Robot path by PSO motion planner (Scenario - IV)

For scenarios - I to IV of Figs.8.18 (a) to 8.21 (a), the mobile manipulator is moving from its source position to the target position using AIMP.

- Fig. 8.18 (a) - (I), Fig. 8.19 (a) - (I), Fig. 8.20 (a) - (I) and Fig. 8.21 (a) - (I) represent the robot motion (in simulation mode) from its source position to target position.
- Fig. 8.18 (a) - (II), Fig. 8.19 (a) - (II), Fig. 8.20 (a) - (II) and Fig. 8.21 (a) - (II) represent the robot motion (in real mode) from its source position to target position.
- Fig. 8.18 (a) - (III) to Fig. 8.21 (a) - (III), Fig. 8.19 (a) - (II), Fig. 8.20 (a) - (II) and Fig. 8.21 (a) - (II) represent the robot positions during its navigation from source to target.

For scenarios - I to IV of Figs.8.18 (b) to 8.21 (b), the mobile manipulator is moving from its source position to the target position using type-1 based PSO motion planner.

- Fig. 8.18 (b) - (I), Fig. 8.19 (b) - (I), Fig. 8.20 (b) - (I) and Fig. 8.21 (a) - (i) represent the robot motion (in simulation mode) from its source position to target position.
- Fig. 8.18 (b) - (II), Fig. 8.19 (b) - (II), Fig. 8.20 (b) - (II) and Fig. 8.21 (b) - (II) represent the robot motion (in real mode) from its source position to target position.
- Fig. 8.18 (b) - (III) to Fig. 8.21 (b) - (III), Fig. 8.19 (b) - (II), Fig. 8.20 (b) - (II) and Fig. 8.21 (b) - (II) represent the robot positions during its navigation from source to target.

The path analysis results for the scenarios Figs.8.18 to 8.21 are illustrated in Tables. 8.1. In all the scenarios, AIMP is giving better results as compared to the developed PSO motion planner. Path analysis results of scenarios- I to IV showed that the robot motion in simulation environments are giving good agreement (percent of error is below 5%) with the real time environments.

Table 8.1 Path analysis results for Figs. 8.18 to 8.21

Scenario	Motion Planner	Simulation				Experimental				% of error
		No. of Iter0ations	Time Taken (sec.)	Distance travelled (cm)	% of path deviation	No. of Iterations	Time Taken (sec.)	Distance travelled (cm)	% of path deviation	
I	AIMP	520	208	624	6.5%	529	211.6	634.8	7.4%	1.7%
	PSO	556	222.4	667.2		571	228.4	685.2		2.6%
II	AIMP	531	212.4	637.2	6.5%	539	215.6	646.8	8.1%	1.5%
	PSO	568	227.2	681.6		587	234.8	704.4		3.2%
III	AIMP	355	142	426	3.3%	364	145.6	436.8	4%	2.5%
	PSO	367	146.8	440.4		379	151.6	454.8		3.1%
IV	AIMP	439	175.6	526.8	4.8%	428	513.6	171.2	4.6%	2.6%
	PSO	461	184.5	553.6		449	179.6	538.8		2.7%

It is observed that the AIMP is the most efficient motion planner among the four developed motion planners. Experimental results showed that the designed robot generated safest paths by avoiding obstacles and escaping traps and finally reached to its destination with the help of proposed algorithms. Figs. 8.18 to 8.21 represent the path travelled by the differential platform using developed immune based and swarm based algorithms.

In PSO based motion planner, further movement of the robot is decided according the particles fitness value. If the robot moves in the environment with longer objects (like to perform wall following nature) beyond the sensing rang, it takes more search time for determining the global best position. Due to this nature, sometimes the robot may generate indefinite motions within its search space. Unlike PSO control strategy, AIMP works based on the dominant antigen in the previous iteration and performs its next movement in short span of time. The complete differentiation of AIMP from the PSO algorithm is illustrated in Table. 8.2.

Table 8.2. Comparison between AIMP & PSO motion planner

Criteria	**PSO**	**AIMP**
Number of tuning parameters	More	Less
Mathematical complexity	More	Less
Trap situation avoidance	Not as much of AIMP	Good in avoiding
Adaptive nature	No	Yes
Actuation time by robot	Takes more time	Takes less time
Shortest path achievement	Certainly not, as compared to AIMP	Yes

Once the robot system reached to its target position, the manipulator which mounted on the mobile platform will come to the actuation and will search for the object to perform pick & place operation. Fig. 7.17 represents the manipulator picked the detected object after the robot system reached to its goal position.

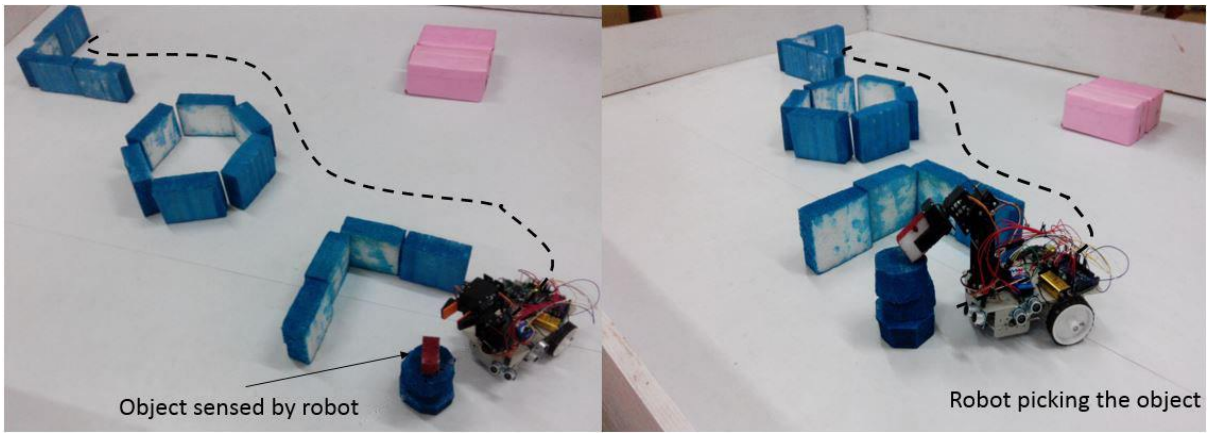


Fig. 8.22 Manipulation task by the developed mobile manipulator for the detected object

8.5. Validation with ER-400 Mobile Platform

The ER-400 robot's platform is ideal for teaching and research in a variety of fields, including artificial intelligence, control, navigation, real time programming, remote viewing & operation and materials handling.

ER 400 mobile robot (Intellitek) has been procured (Fig.7.14) and used in this investigation to validate the efficiency of the developed methodologies.

Fig.8.23 ER 400 mobile robot

SPECIFICATIONS OF ER400 MOBILE PLATFORM

A. Characteristics:

- Dimensions: Diameter: 43 cm, Height: 33 cm
- Weight: 14 kg (including 2 x 3 kg batteries)
- Payload: Up to 10 kg
- Power: 2 x 12V, 7AH sealed lead acid rechargeable batteries
- Average work time: 5 hours
- Charging time: 8 hours

- Controller: Hitachi microcontroller.
- Drive:

  2-wheel drive,
  2 amp motors with odometers,
  Rear caster with position sensors,
  Maximum speed: 40 cm/sec,
  Turn radius: 0 cm.
- Communications Ports:

  RS232 or USB serial ports
  5 IR receivers for remote control
  TCP/IP (from on board PC or PPC)

B. Actions of ER400:

- Run autonomously
- Avoid obstacles
- Carry large accessories
- Localize
- Follow a wall
- Communicate and control information via sensors
- Communicate with other robots
- Play and listen to sound and synthesize speech
- Map spaces, including rooms and buildings
- Plan paths for robot navigation
- Analyze and transmit video

C. Ports Available for User Program:

- Analog and digital I/O
- 3 PWM outputs

D. Control Software Options:

- C/C++/VB
- MATLAB

E. Sensors

- 10 ultrasonic sensors with resolution ±5mm
- 1 Optical Range Measurement (PSD) sensor
- 14 analog sensors
- 7 infrared sensors
- 18 digital sensors

F. Safety Features:

- Touch sensing bumper
- Senses contact of less than100g
- 10 ultrasonic sensors
- Stair detector (robot stops operation upon detecting a stair)
- CE approved
- CIM integration

G. Camera Specification:

- 1/4 inch color CCD sensor
- 43mm Lens Size
- Pan: Range 270º
- Tilt: Range 90º
- 4x Digital Zoom

Experiments have been conducted using real mobile robots for transporting light weight objects from one place (source) to another place (target) in Lab environment. ER-400 robot is navigating from source to destination, surrounded with a number of obstacles in a cluttered environment. The robot can detect objects and obstacles by using vision sensor (color camera) mounted on it.

The mobile platform is equipped with a 4-axis manipulator as shown in the Fig. 7.15. The specifications of the manipulator are illustrated in Table 7.5.

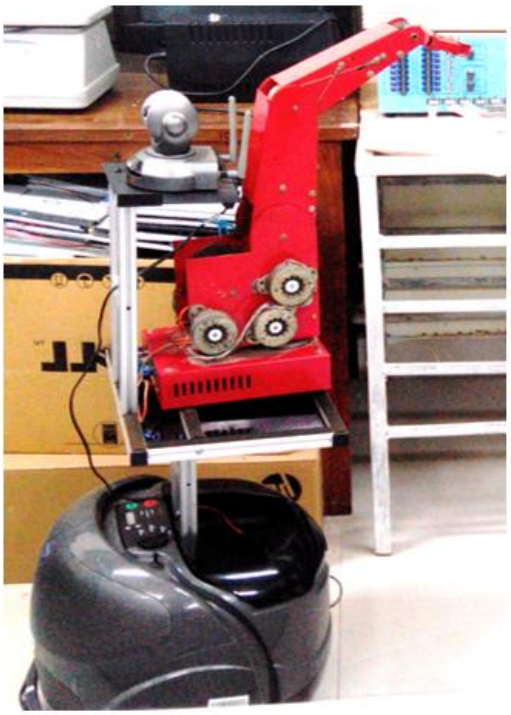


Fig. 8.24 Manipulator mounted on ER-400 mobile platform

Table 8.3 Basic Specification of the Manipulator

Specification	**Value**	**Units**
Number of axes	4	
Horizontal reach	300	mm
Vertical reach	350	mm
Drives	5 no. of 0.9^0 stepper motors	
Configuration	4 Axes plus gripper All axes completely independent All axes can be controlled simultaneously	

Here, AIMP methodology has been used as an obstacle avoidance algorithm because form the developed motion planners AIMP gives efficient results as discussed in previous sections. The Hitachi microcontroller embedded in the robot is loaded with C++ program to carry object from initial position i.e. predefined co-ordinate (x1, y1; source) to desired location (x2, y2; destination) avoiding obstacles in an unstructured and dynamic environment.

Robot is handling object during its motion towards the target position. Some of the coordinates followed by ER400 robot have been shown pictorially (Figs.8.25 &8.26). During the robot navigation (in lab environment) from its source position (100, 15), it reached to various intermediate positions (85, 315), (112, 468), (121,591), and (62, 638) as shown in Fig.8.26 (iii) - Fig.8.26 (iii).

From Table 8.3, it is observed that the developed motion planner theoretical result is giving good agreement with the experimental result (error < 4%).

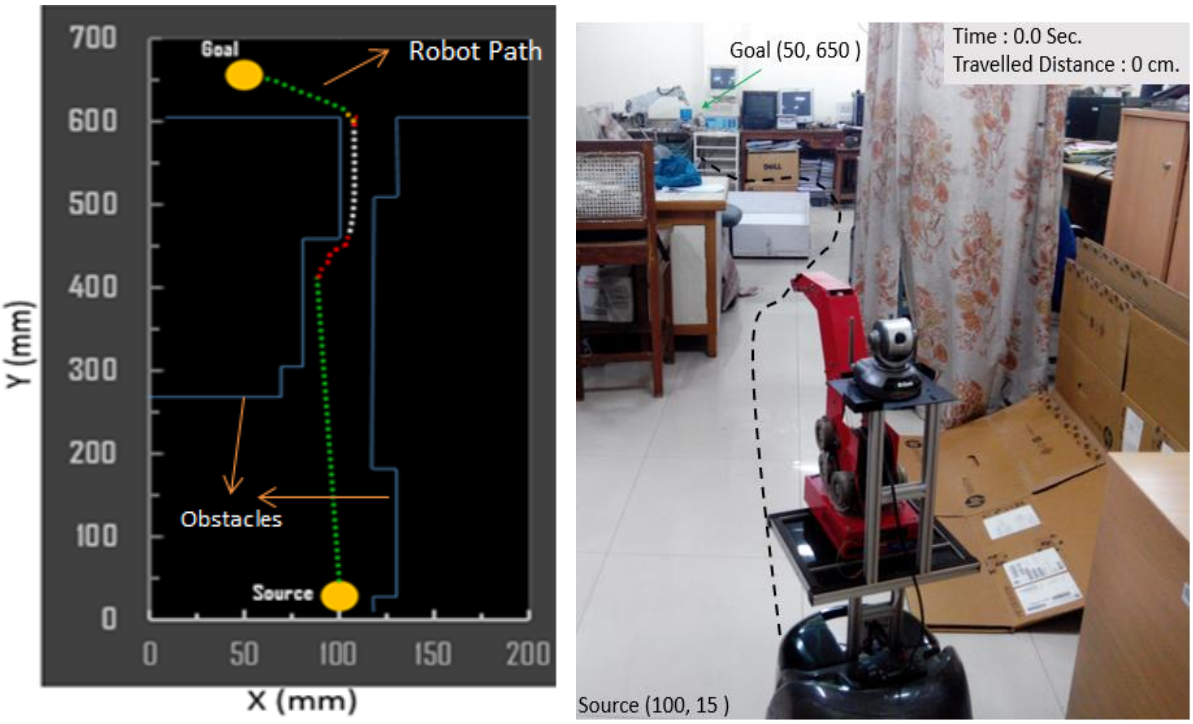
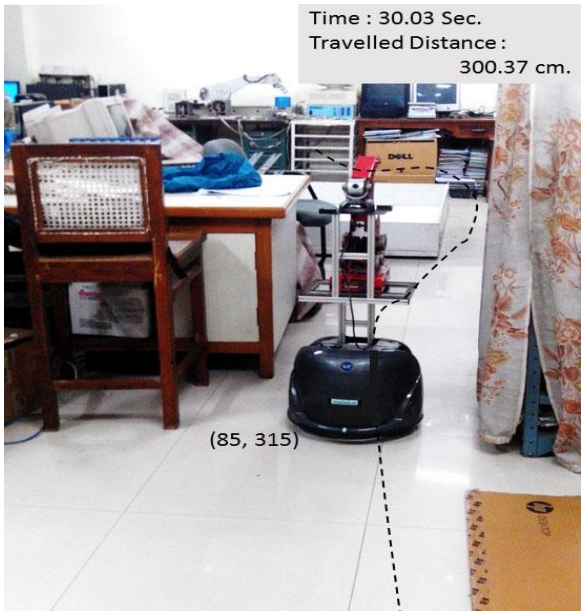
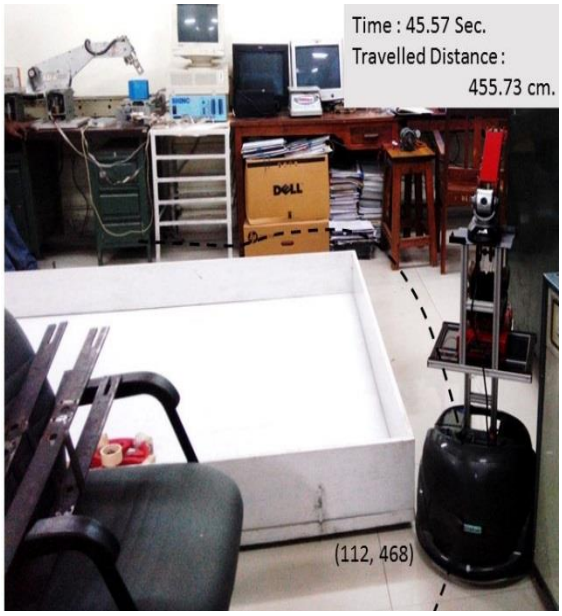


Fig. 8.25 Path generated by ER400 robot in simulation & experimental environment

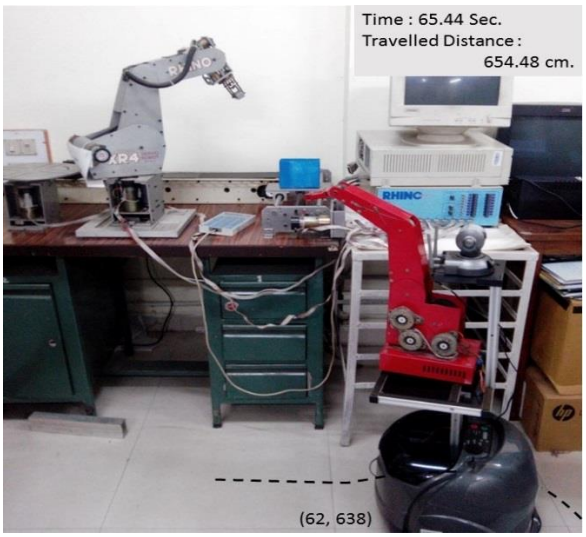


(i)



(ii)

(iii)



(iv)

Fig. 8.26 Path generated by mobile manipulator to reach its target position

Table 8.4 Path analysis results of ER-400 mobile platform

Scenario	Simulation			Experimental			% of error
	No. of Iterations	Time Taken (sec.)	Distance travelled(cm)	No. of Iterations	Time Taken (sec.)	Distance travelled(cm)	
8.25 (a)	69	68.2	682	71	70.5	704.8	3.27%

Once the robot system reached to its target position, the manipulator which mounted on the mobile platform will come to the actuation and will search for the object to perform pick & place operation. Fig. 8.26 represents the manipulator picked the detected object after the robot system reached to its goal position.

Fig. 8.27 Manipulation task by mobile manipulator for the detected object

8.6. Summary

Two experimental setups have been developed independently, (i) task performed by the manipulator and (ii) the mobile platform path generation within its environments. The manipulator in this study has 4-DOF and is used here for pick & place task. While performing the operation by the robot online monitoring is performed by embedding various electronic elements with microcontroller. In the same way the mobile platform structure is made and developed motion planners have been implemented to the designed differential mobile platform. Experimental results are deviated within 5% as compared with theoretical results.

Form the simulation as well as the experimental results, AIMP is generating safest and shortest paths as compared to the PSO based motion planners (Table 8.1). In addition, the efficient motion planner (AIMP is efficient motion planner as compared to PSO motion planner) is validated with ER-400 mobile platform.

Chapter 9

CONCLUSION & FUTURE SCOPE

Contributions

Conclusions

Future Scope

9. CONCLUSION & FUTURE SCOPE

This thesis work investigation has been carried out to generate flexible operational space using the robotic arm by integrating mobility feature to it. This feature facilitates the mobile manipulator with the ability to perform its tasks in a large workspace. The mobile manipulator considered in this study consists of a robotic manipulator mounted on a mobile platform. The development of the hybrid manipulator system covers mechanics of systems design, system dynamic modeling and simulations, design optimization, computer architecture, and control system design.

9.1. Contributions

This section summarizes the main contributions of the current research work as follows:

→ *Kinematic modeling of manipulator:* The motion of the structure is analyzed in two ways, forward and inverse kinematics. Forward kinematics is the determination of the every link configuration, specially the end-effector, when the joint variables are given. Inverse kinematics deals with determining the joint variables of a robot manipulator for the given position and orientation of the end-effector. The standard Denavit-Hartenberg convention is implemented for determining coordinate frames attached to each robot's link. Later, arm equation is developed for the considered 4-axis manipulator using forward kinematic models. From the developed arm equation, inverse kinematic models have been developed to find out the joint parameters.

→ *Kinematic modeling of mobile platform:* The behavior of various wheeled mobile platforms has been analyzed. The work aimed towards to investigate the complete description of the control theory of wheeled robots and its maneuverability. Equations are modeled to describe the rigid body motions that arise from rolling trajectories based on the geometrical constraints of these wheels. Moreover, this study explains about the differential drive wheeled platform and its motion equations in terms of wheel velocities.

→ *Coordination of manipulation & locomotion:* The prime objective of the work is to develop a hybridised robot system by integrating the kinematic models of a 4-axis manipulator and a differential mobile platform. The developed WMM is controlled by five parameters in which three parameters (joint velocities) gives the velocity information of the manipulator and the remaining two (left & right wheel velocities) corresponds the differential mobile platform. The final velocity Jacobean has been developed for a mobile

manipulator which controls the entire robot system and makes the robot to follow the desired trajectories by the manipulator and platform within its workspace.

→ *Swarm based motion control paradigm:* A new computational methodology has been proposed for solving path planning problem of an intelligent mobile platform, based on Particle Swarm Optimization. The developed algorithm is effective in avoiding obstacles and generating optimal paths within its unknown environments. The trajectories generated by robot are based on the selection of global best position in each iteration. Among the swarm, the particle which has the minimum fitness is considering as the global best position. There by, the robot moves towards the global best position and this process is continued for several iterations until the robot reaches its target position. Moreover, from the developed two fitness functions, type-1fitness is giving efficient results as compared to type-2 fitness function.

→ *Immune based motion control paradigm:* Two efficient immune based motion planners have been introduced for solving mobile robot path planning problem in unknown environments. The first motion planner called innate immune based motion planner, comprises the parameter 'Learning Rate' and is not getting any global information from the system. For the second motion planner, an efficient adaptive learning mechanism has been presented and integrated to the developed innate immune based motion planner. Path analysis results showed that, both motion planners are generating collision free paths in their robotic search space. Moreover, from the developed swarm based & immune based motion planners, AIMP gives better results as compared to PSO based motion planner.

→ *Experiments:* Two experimental setups have been developed independently, one is the task performed by the manipulator and another is for the mobile platform path generation within its environments. The manipulator in this study has 4-DOF and is used for pick & place task. Online monitoring of the entire robot system is performed while the WMM is in operation, by interfacing various electronic elements with microcontroller. In the same way the mobile platform structure is made and developed motion planners have been implemented to the designed differential mobile platform. Experimental results are in good agreement with the theoretical results. Path analysis results showed that, from the developed swarm based & immune based motion planners, AIMP gives better results as compared to PSO based motion planner.

9.2. Conclusions

The conclusions drawn from the current investigation are depicted below:

1. Kinematic models of the manipulator have been done to analyze the position of the arm during pick and place task in various situations.
2. Kinematic analysis for the wheeled mobile platform has been done to measure its maneuverability while subjected to wheel geometric constraints.
3. Co-ordination of manipulator and mobile platform has been analyzed to reach the various coordinates in cluttered work space efficiently and effectively.
4. Swarm system architecture has been introduced to generate the optimal trajectories by the mobile manipulator. With the help of sensory information, two fitness functions namely type-1 fitness and type -2 fitness have been developed to solve path planning problem. Results showed that, type-1 fitness based motion planner giving efficient results as compared to type-2 fitness function type-1 fitness based motion planner by 7%.
5. Immune based system architecture has been introduced to generate the optimal trajectories by the mobile manipulator. With the help of sensory information, two motion planners namely IIMP and AIMP have been developed to solve path planning problem. Results showed that, AIMP gives efficient results as compared to IIMP by 4%.
6. Comparison evaluations have been made for the developed type-1 fitness based motion planner (efficient swarm based architecture) and AIMP (efficient immune based architecture). . Path analysis results showed that, AIMP gives better results as compared to type-1 fitness based PSO motion planner by 6.5%.
7. Since the developed AIMP solves the path planning problem efficiently, AIMP has been validated with ER-400 mobile platform in simulation and lab environments. Experimental results are giving good agreement with theoretical results (error <4%).

9.3. Future Scope

The focus of this thesis is on modeling, control, and coordination of a single mobile manipulator which consists of a 4-axis manipulator mounted on differential mobile platform. The following is a list of interesting directions to pursue as future work by improving the state of the art immediately related to this work.

→ *Multiple mobile manipulators:* One of the techniques for coordinating multiple mobile manipulators is the leader/follower mothodology, in which one WMM is chosen as the leader and the other WMMs are designated as followers. These types of systems are used to perform simple tasks such as jointly transporting a large object. The algorithms

developed in Chapter 6 & 7 can be implemented to the followers that keep in contact with and support the object to be transported. The coordination of multiple WMMs when performing complex tasks such as mechanical parts assembly is a future study where coordination strategies other than the leader/follower has to be considered.

→ *Two manipulators on a mobile platform:* In this investigation a mobile manipulator is considered which has only one manipulator equipped on top of the mobile platform. In general, one may have two or more manipulators on the same mobile platform for performing coordination works. In that case, the motion of one manipulator will affect the motion of the mobile platform as well as the motion of the other manipulators aboard.

→ The system is to be redesigned with the effects of external forces under certain circumstances. This feature renders more applicability of the proposed coordination algorithm, while the system is in interaction with ubiquitous environments.

REFERENCES

[1]. J.P. Merlet, Workspace-oriented methodology for designing a parallel manipulator, Proceeding of IEEE international conference on robotics and automation, Nantes France, 1996, PP.777-786.

[2]. W-M. Hwang and Y-W. Hwang, Computer-aided structural synthesis of plan kinematic chains with simple joints, mechanism and machine theory, vol. 27, no.2, 1992, PP.189-199.

[3]. L-M. Herve, The mathematical group structure of the set of displacements, Mechanisms and machine theory, vol. 29, no.1, 1994, PP.73-81.

[4]. M. Karouia and J.M. Herve, Asymmetrical 3-dof spherical parallel mechanisms, European journal of mechanics (A/Solids), vol.24, no.1, 2005, PP.47-57.

[5]. J. Lloyd and V. Hayward, Kinematics of common industrial robots, North-holland robotics, vol.4, 1994, PP. 160-191.

[6]. S. Cubero, Industrial robotics-theory modelling and control, ISBN 3-86611-285-8, PP. 964, ARS/plV, Germany, December 2006, PP. 83-116.

[7]. I. Milicevic, R.Slavkovic and D.Golubovic, Industrial robot models designing and analysis with application of MATLAB software, Faculty of technical sciences NOVI SAO, may 18th 2007, 47th anniversary of the faculty, PP. 71-78.

[8]. G.Singh, Dr. V. K. Banga and J.Kaur, Robotic arm kinematics and soft computing, Proceedings of international conference on advances in electrical and electronics engineering (ICAEE'2011), PP. 189-193.

[9]. A.Srikanth, Y.Ravithej, V.Sivaraviteja and V.Sreechand, Kinematic analysis and simulation 6 D.O.F of robot for industrial applications, International journal of engineering and science, vol.3, no.8, 2013, PP. 01-04.

[10]. E. Sariyildiz and H. Temeltas, A new formulation method for solving kinematic problems of multiarm robot systems using quaternion algebra in the screw theory framework, Turk J Elec Eng & Comp Sci, vol.20, no.4, 2012, PP. 607-6228.

[11]. J.Zhang and J.Cai, Error analysis and compensation method of 6-axis industrial robot, International journal on smart sensing and intelligent systems, vol. 6, no. 4, 2013, PP.1383-1399.

[12]. P.S. Shiakolas, K.L. Conrad, and T.C. Yih, On the accuracy, repeatability, and degree of influence of kinematics parameters for industrial robots, International journal of modelling and simulation, vol. 22, no. 3, 2002, PP.1-10.

[13]. P.Krzic, F. Pusavec, J. Kopac , Kinematic constraints and offline programming in robotic machining applications, Technical Gazette, vol.20, no.1, 2013, PP.117-124.

[14]. I.C. Ha, Kinematic parameter calibration method for industrial robot manipulator using the relative position, Journal of mechanical science and technology, vol. 22, no.6, 2008, PP.1084-1090.

[15]. C.Pozna, The modular robots kinematics**,** Acta polytechnica hungarica, vol. 4, no. 2, 2007, PP. 5-18.

[16]. Y.Aydin and S.Kucuk,Quaternion based inverse kinematics for industrial robot manipulators with euler wrist, Proceeding of 3rd IEEE international conference on mechatronics, 1-4244-97, 13th April 2006, PP. 581-586.

[17]. M.J.Hayawi, Analytical inverse kinematics algorithm of a 5-DOF robot arm, Journal of education of college, vol.1, no.4, march 2011.

[18]. A**.**Pashkevich, Real-time inverse kinematics for robots with offset and reduced wrist, Control eng. practice, voi. 5, no.10, 1997, PP. 1443-1450.

[19]. P. Marothiya and S. K. Saha, Robot inverse kinematics and dynamics algorithms for windows**,** Proceedings of the Conference on Advances. and Recent Trends in Manufacturing, New Delhi, Dec., 2003, PP. 229-237.

[20]. S.Alavandar, M. J. Nigam, Neuro-fuzzy based approach for inverse kinematics solution of industrial robot manipulators, International journal of computers, communications & control, vol. 3, no. 3, 2008, PP. 224-234.

[21]. A.V.Duka, Neural network based inverse kinematics solution for trajectory tracking of a robotic arm, Proceedings of 7th International conference on inter disciplinary in engineering, Procedia technology. 12 (2014), PP. 20 – 27.

[22]. S.A.Stoeter and N.Papanikolopoulos, Kinematic motion model for jumping scout robots**,** IEEE transactions on robotics, vol. 22, no. 2, 2006, PP. 398-403.

[23]. K. Hosoda, T. Takuma, A. Nakamoto and S. Hayashi, Biped robot design powered by antagonistic pneumatic actuators for multi-modal locomotion, Robotics and autonomous systems, vol.56, no.1, 2008, PP. 46–53.

[24]. X. Wu and S. Ma, CPG-based control of serpentine locomotion of a snake-like robot, Mechatronics, vol. 20, no. 2, 2010, PP. 326–334.

[25]. Z.Y. Bayraktaroglu, Snake-like locomotion: Experimentations with a biologically inspired wheel-less snake robot, Mechanism and machine theory, vol. 44, no.3, 2009, PP.591–602.

[26]. J.L. Guzman, M. Berenguel, F. Rodriguez and S. Dormido, An interactive tool for mobile robot motion planning, Robotics and autonomous systems, vol. 56, no.5, 2008, PP.396–409.

[27]. N.Chakraborty and A.Ghosal, Kinematics of wheeled mobile robots on uneven terrain, Mechanism and machine theory, vol.39, no.12, 2004, PP.1273–1287.

[28]. J.Borenstein and Y.Koren, Motion control analysis of a mobile robot, Transactions of ASME, Journal of Dynamics, Measurement and Control, vol. 109, no. 2, 1986, PP. 73-79.

[29]. J.C.Alexander and J.H.Maddocks, The kinematics and control of wheeled mobile robots, SRC TR 87-196, October 1987, PP. 1-15.

[30]. G.Campion, G.Bastin, and B.D.AndrCa-Novel, Structural properties and classification of kinematic and dynamic models of wheeled mobile robots, IEEE transactions on robotics and automation, vol.12, no.1, 1996**,** PP.47-62.

[31]. D.S.Kim, W.H.Kwon and H.S.Park, Geometric kinematics and applications of a mobile robot**,** International Journal of Control, Automation, and Systems vol. 1, no. 3, 2003, PP.376-384.

[32]. J.Yi,H.Wang,J.Zhang, D.Song, S.Jayasuriya, and J.Liu, Kinematic modelling and analysis of skid-steered mobile robots with applications to low-cost inertial-measurement-unit-based motion estimation, IEEE transactions on robotics, vol.25, no.5, 2009, PP.1087-1097.

[33]. C.Cariou, R.Lenain, B.Thuilot and P.Martinet, Adaptive control of four-wheel-steering off-road mobile robots:Application to path tracking and heading control in presence of sliding, Proceedings of IEEE/RSJ international conference on intelligent robots and systems, France, 22-26th September, 2008, PP.1759-1764.

[34]. F.R.Marcovitz and A.Kelly, On-line Mobile Robot Model Identification using Integrated Perturbative Dynamics, Proceedings of 12th international symposium on experimental robotics, December, 2010, PP.1-15.

[35]. J.H.Lee, B.K.Kim, T.Tanikawa and K.Ohba, Kinematic analysis on omni-directional mobile robot with double-wheel-type active casters, Proceedings of international conference on control, automation and systems, October. 2007, Seoul, Korea, PP.1217-1221.

[36]. W.Chung, C.Moon, C.Jung and J.Jin, Design of the dual offset active caster wheel for holonomic omni-directional mobile robots, International journal of advanced robotic systems, vol.7, no.4, 2010, PP. 105-110.

[37]. Y.P.Li, T.Zielinska, M.H.Ang Jr. and W.Linz, Vehicle Dynamics of Redundant Mobile Robots with Powered Caster Wheels, Proceedings of the sixteenth symposium on Robot design, dynamics and control, 2006, PP.221-228.

[38]. S. Kim and S. Lee, Local and global isotropy analysis of mobile robots with three active caster wheels, Advances in service robotics, InTech publisher, ISBN 978-953-7619-02-2, 2008, PP.117-126.

[39]. G.Indiveri, Swedish wheeled omnidirectional mobile robots: kinematics analysis and control, IEEE transactions on robotics, vol.25, no.1, 2009, PP.164-171.

[40]. G.Indiveri, J.Paulus and P.G. Ploger, Motion control of swedish wheeled mobile robots in the presence of actuator saturation, G. Lakemeyer *et. al.* (Eds.): RoboCup 2006, LNAI 4434, 2007, PP.35–46.

[41]. I.Doroftei, V.Grosu and V.Spinu, Omnidirectional mobile robot – design and implementation, Bioinspiration and robotics: walking and climbing robots, ISBN 978-3-902613-15-8, In-Tech publishers, 2007, PP.511-528.

[42]. K.Tadakuma and R.Tadakuma, Mechanical design of "Omni-Ball": spherical wheel for holonomic omnidirectional motion, Proceedings of 3rd annual IEEE conference on automation science and engineering, Scottsdale, AZ, USA, 2007, PP.788-794.

[43]. C.W.Wu and C.K.Hwang, A novel spherical wheel driven by omni wheels, Proceedings of 7th International conference on machine learning and cybernetics, Kunming, 2008, PP.3800-3803.

[44]. R. Mukherjee, M.A.Minor, and J.T.Pukrushpan, Motion planning for a spherical mobile robot: revisiting the classical ball-plate problem, Transactions of the ASME - Journal of dynamic systems, measurement, and control, vol. 124, 2002, PP.502-511.

[45]. V.A.Joshi, R.N.Banavar and Rohit Hippalgaonkar, Design and analysis of a spherical mobile robot, Mechanism and Machine Theory 45 (2010), PP. 130–136.

[46]. T. B. Lauwers, G. A. Kantor, and R. L. Hollis, A dynamically stable single-wheeled mobile robot with inverse mouse-ball drive, Proceedings of IEEE International conference on robotics and automation, Orlando, FL, 2006.

[47]. D. Chwa, Tracking control of differential-drive wheeled mobile robots using a backstepping-like feedback linearization, IEEE transactions on systems, man, and cybernetics—part a: systems and humans, vol. 40, no. 6, 2010, PP. 1285-1295.

[48]. P.Petrov and L.Dimitrov, Nonlinear path control for a differential drive mobile robot**,** RECENT, vol.11, no.1, 2010, PP.41-45.

[49]. R.Rashid, I.Elamvazuthi, M.Begam and M. Arrofiq, Fuzzy-based navigation and control of a non-holonomic mobile robot, Journal of computing, vol.2, no.3, 2010, PP.130-137.

[50]. L. Mikova, R.Surovec, E.Prada and M.Kelemen, Mathemathical model of mobile robot for purpose of path tracking of the robot, journal of interdisciplinary research, vol.1, no.1, PP.146-147.

[51]. F.L.Menn, P.Bidaud and F.B.Amar, Generic differential kinematic modeling of articulated multi-monocycle mobile robots, Proceedings of IEEE International conference on robotics and automation, Orlando, Florida, 2006, PP.1505-1510.

[52]. L. Feng, Y. Koren and J. Borenstein, A Model-reference adaptive motion controller for a differential-drive mobile robot, Proceedings of IEEE International Conference on Robotics and Automation, San Diego, CA, 1994, PP.3091-3096.

[53]. R.Dhaouadi and A.A.Hatab, Dynamic modelling of differential-drive mobile robots using lagrange and newton-euler methodologies: a unified framework, Advances in robotics and automation, vol.2, no.2, PP.2-7.

[54]. S. Slusny, R.Neruda and P.Vidnerova, Comparison of behavior-based and planning techniques on the small robot maze exploration problem, Neural Networks, vol.23, no.4, 2010, PP. 560-567.

[55]. G.Klancar, D.Matko and S.Blazic, A control strategy for platoons of differential drive wheeled mobile robot, Robotics and autonomous systems, vol.59, no.2, 2011, PP.57–64.

[56]. W.G.Ali, A semi-autonomous mobile robot for education and research, Journal of king saud university – engineering sciences, vol.23, no.2, 2011, PP.131–138.

[57]. G.Indiveri, A.Nuchter and K.Lingemann, High speed differential drive mobile robot path following control with bounded wheel speed commands, Proceedings of IEEE International conference on robotics and automation, Roma, Italy, 2007, PP.2202-2207.

[58]. B.Bayle, J.Y.Fourquet, F.Lamiraux and M.Renaud, Kinematic control of wheeled mobile manipulators, Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems EPFL, Lausanne, Switzerland, October 2002, PP.1572-1577.

[59]. A.D.Luca. G.Oriolo and P.R.Giordano, Kinematic modeling and redundancy resolution for nonholonomic mobile manipulators, Proceedings of IEEE International conference on robotics and automation, Orlando, Florida - May 2006, PP.1867-873.

[60]. J. Y. Fourquet, B. Bayle and M. Renaud, Nonholonomic mobile manipulators: kinematics, velocities and redundancies, Journal of intelligent and robotic systems, vol.36, no.1, 2003, PP. 45–63.

[61]. T. D. Viet, P. T. Doan, N. Hung, H.K. Kim and S.B. Kim, Tracking control of a three-wheeled omnidirectional mobile manipulator system with disturbance and friction, Journal of mechanical science and technology, vol.26, no.7, 2012, PP.2197-2211.

[62]. F. G. Pin and S. M. Killough, A new family of omnidirectional and holonomic wheeled platforms for mobile robots, IEEE Transactions on robotics and automation, vol.10, no.4, 1994, PP.490-489.

[63]. A. Betourne and G. Campion, Dynamic modelling and control design of a class of omnidirectional mobile robots, Proceedings of IEEE International conference on robottics and automation, Minnesota (1996), pp .2810-2815.

[64]. K. Watanabe, Y. Shiraishi, S. G. Tzafestas, J. Tang and T. Fukuda, Feedback control of an omnidirectional autonomous platform for mobile service robots, Journal of Intelligent and Robotic Systems, vol.22, no.3-4, 1998, PP. 315-330.

[65]. H. C. Huang and C. C. Tsai, Adaptive robust control of an omnidirectional mobile platform for autonomous service robots in polar coordinates, Journal of Intelligent and Robotic Systems, vol.51, no.4, 2008 , PP.439-460.

[66]. H. C. Huang and C. C. Tsai, FPGA Implementation of an embedded robust adaptive controller for autonomous omnidirectional mobile platform, IEEE Transactions on Industrial Electronics, vol.56, no.5, 2009, PP.1604-1616.

[67]. D. Xu, D. Zhao, J. Yi and X.Tan, Trajectory tracking control of omnidirectional wheeled mobile manipulators robust neural network-based sliding mode approach,

IEEE Transactions on Systems, Man, and Cybernetics, vol.39, no.3, 2009, PP.788-799.

[68]. N. Hung, D. W. Kim, H. K. Kim and S. B. Kim, Tracking controller design of omnidirectional mobile manipulator system, ICROS-SICE International Joint Conference, Fukuoka, 2009, PP.539-544.

[69]. S. Datta, R. Ray and D. Banerji, Development of autonomous mobile robot with manipulator for manufacturing environment, International Journal Advanced Manufacturing Technology, vol.38, no.5-6, 2008, PP.536–542.

[70]. C.P.Tang, P. T. Miller, V. N. Krovi, J.C. Ryu and S. K. Agrawal, Kinematic control of a nonholonomic wheeled mobile manipulator – a differential flatness approach, Proceedings of ASME Dynamic Systems and Control Conference, 2008, Ann Arbor USA.

[71]. M. H. Korayem , H. Gariblu, Maximum allowable load on wheeled mobile manipulators imposing redundancy constraints, Robot autonomy system, vol. 44, no. 2, 2003, PP.151–159.

[72]. M. H. Korayem, H. Ghariblu, Maximum allowable load of mobile manipulator for two given end points of end-effector, International journal of advanced manufacturing technology, vol. 24, no. 9-10, 2004, PP. 743–751.

[73]. M. H. Korayem, H. Ghariblu, A. Basu, Dynamic load-carrying capacity of mobile-base flexible joint manipulators, International journal of advanced manufacturing technology, vol. 25, no. 1-2, 2005, PP.62–70.

[74]. M. H. Korayem, A. Nikoobi, V. Azimirad, Maximum load carrying capacity of mobile manipulators: optimal control approach, International journal of advanced manufacturing technology, vol. 46, no. 5-8, 2010, PP. 811–829.

[75]. M. H. Korayem, H. Ghariblu, A. Basu, Optimal load of elastic joint mobile manipulators imposing an overturning stability constraint, International journal of advanced manufacturing technology , vol. 26, no. 5-6, 2005, PP. 638–644.

[76]. M. H. Korayem, V. Azimirad , A. Nikoobin and Z. Boroujeni, Maximum load-carrying capacity of autonomous mobile manipulator in an environment with obstacle considering tip over stability, International journal advanced manufacturing technology, 2010,DOI 10.1007/s00170-009-2146-0, PP. 811–829.

[77]. K. Tchon, J. Jakubiak and L. Malek, Dynamic Jacobian inverses of mobile manipulator kinematics, Advances in Robot Kinematics: Motion in Man and Machine, DOI 10.1007/978-90-481-9262-5_2, 2010, PP. 15-21.

[78]. G. Zhong, Y. Kobayashi, Y. Hoshino and T. Emaru, System modeling and tracking control of mobile manipulator subjected to dynamic interaction and Uncertainty, Nonlinear Dyn, DOI 10.1007/s11071-013-0776-0, 2013, PP.167–182.

[79]. Y. Yamamoto, X. Yun, X, Effect of the dynamic interaction on coordinated control of mobile manipulators, IEEE robotics and automation society, vol. 12, no. 5, 1996, PP. 816–824.

[80]. Yamamoto, Y., Yun, X.: Modeling and compensation of the dynamic interaction of a mobile manipulator. In: Proceedings of IEEE International conference on robotics and automation, San Diego, CA, 1994, PP. 2187–2192.

[81]. M. Meghdari, M. Durali and, D. Naderi, Investigating dynamic interaction between the one d.o.f. manipulator and vehicle of a mobile manipulator, Journal of intelligent and robotic systems, vol. 28, no. 3, 2000, PP. 277–290.

[82]. C. O'Dunlaing, C. K. Yap, A retraction method for planning the motion of a disk, Journal of algorithms, vol. 6, no. 1, 1985, PP. 104–111.

[83]. JiYeong Lee and Howie Choset, Sensor-Based exploration for convex bodies: a new roadmap for a convex-shaped robot, IEEE Transactions on robotics, vol.21,no. 2, 2005 PP. 240-247.

[84]. N. A. Vlassis, G. Papakonstantinou and P. Tsanakas(1997), Learning the voronoi centers of a mobile robot's configuration space, Proc. 3rd ECPD international conference on advanced robotics, Intel automation and act systems, 1997, PP. 1-5.

[85]. B. Lisien and D.Morales, The hierarchical atlas, IEEE Transactions on robotics, vol. 21, no. 3, 2005, PP. 473-481.

[86]. J.Y. Lee and H. Choset, Sensor-Based exploration for convex bodies: a new roadmap for a convex-shaped robot, IEEE Transactions on robotics, vol.21,no. 2, 2005, PP. 240-247.

[87]. Yamamoto. M., Iwamura, M. and Mohri, A.,Near-time-optimal trajectory planning for mobile robots with two independently driven wheels considering dynamical constraints and obstacles, Journal of the robotics society of japan, vol.16, no.8, 1998, PP. 95–102.

[88]. Lozano-Perez, T. and Wesley,M.A, An algorithmfor planning collision-free paths among polyhedral obstacles, Communications of the ICM, vol. 22, no. 10, 1979, PP. 560–570.

[89]. B. J. Oommen, S. SitharamaIyengar, Nageswara S.V.R and R. L. Kashyap (1997), Robot navigation in unknown terrains using learned visibility graphs. part i: the disjoint convex obstacle case, IEEE Journal of robotics and automation, vol. RA-3, no. 6, PP 672-681.

[90]. H.Choset, K.Nagatani and N. A. Lazar, The arc-transversal median algorithm: a geometric approach to increasing ultrasonic sensor azimuth accuracy, IEEE transactions on robotics and automation, vol.19, no. 3,2003 ,pp 513-522.

[91]. M. SuruzMiah and W.Gueaieb, A stochastic approach of mobile robot navigation using customized rfid systems, International conference on signals, circuits and systems, Medenine, 2009, PP. 1-6.

[92]. G. E. Jan, K. Y. Chang, and I. Parberry, Optimal path planning for mobile robot navigation, IEEE/ASME Transactions on mechatronics, vol. 13, no. 4,2007, PP 451-460.

[93]. B. Paden, A. Mees, and Fisher, M. Path planning using a jacobian-based free space generation algorithm, Proceedings of IEEE International conference on robotics & automation, Arizona, USA. 1989, PP. 1732–1737

[94]. A. Zelinsky, A mobile robot exploration algorithm, IEEE Transactions on Robotics & Automation, vol.8, no.6,1992, PP. 707–717.

[95]. E. U. Acar and H. Choset, Sensor-based coverage of unknown environments: incremental construction of morse decompositions, The international journal of robotics research, vol.21, no.4, 2002, PP.345-366.

[96]. E. U. Acar and H. Choset, Exploiting critical points to reduce positioning error for sensor-based navigation, Proceedings of the IEEE international Conference on robotics & automation, Washington, 2002, PP.3831-3837.

[97]. E. U. Acar, H.Choset, A. A. Rizzi, P. N. Atkar and D.Hull, Morse decompositions for coverage tasks, The international journal of robotics research, vol. 21, no. 4, 2002, PP. 331-344.

[98]. I. Rekleitis, V. L.Shue, A. P. New and H. Choset, Limited communication, multi-robot team based coverage, Proceedings of the IEEE international conference on robotics & automation, New Orleans, vol. 4, 2004, PP. 3462-3468.

[99]. V.O.S. Olunloyo and M.K.O. Ayomoh, Autonomous mobile robot navigation using hybrid virtual force field concept, European journal of scientific research, vol.31 no.2, 2009, PP.204-228.

[100]. A. O. Djekoune, K. Achour and R. Toumi, A sensor based navigation algorithm for a mobile robot using the DVFF approach, International journal of advanced robotic systems, vol. 6, no. 2, 2009, PP. 97-108

[101]. S. Akishita, S. Kawamura and K. Hayashi, Laplace potential formoving obstacle avoidance and approach of a mobile robot, Japan-USA symposium on flexible automation, A pacific rim conference, 1990, PP. 139–142.

[102]. David C. Conner, Alfred A. Rizzi, and Howie Choset(2003), Composition of Local Potential Functions for Global Robot Control and Navigation, Proceedings of the 2003 IEEE/RSJ, Intl. Conference on Intelligent Robots and Systems, Las Vegas, Nevada, PP.3546-3551.

[103]. G.C. Luh and W. W. Liu, Dynamic mobile robot navigation using potential field based immune network, systemics, cybernetics and informatics, Vol 5 - No 2, 2000, pp 43-50.

[104]. K. P. Valavanis, T. Hebert, R.Kolluru, and N.Tsourveloudis, Mobile robot navigation in 2-D dynamic environments using an electrostatic potential field, IEEE transactions

on systems, man, and cybernetics-part a: systems and humans, vol. 30, no. 2, 2000, PP.187-196.

[105]. L. Huang, Velocity planning for a mobile robot to track a moving target—a potential field approach, Robotics and autonomous systems, vol. 57, no. 1, 2009, PP. 55–63.

[106]. Y. K. Hwang, and N. Ahuja, A Potential Field Approach to Path Planning, IEEE Transactions On Robotics And Automation, vol. 8, no. 1, 1992, PP 23-32.

[107]. AA. Argyros, Reactive robot navigation: a purposive approach, In Proceedings of the EU TMR Networks Conference, Graz, Austria, May 6-9, 1998.

[108]. V. Lumelsky, T. Skewis, Incorporating range sensing in the robot navigation function, transactions on systems, Man and cybernetics, vol.20, no.5, 1990, PP. 12.

[109]. H. Noborio, T. Yoshioka, An on-line and deadlock-free path planning algorithm based on world topology, IEEE/RSJ International Conference on intelligent robots and systems, Yokohama, vol. 2, 1993, PP. 1425-1430.

[110]. Y. Horiuchi and H. Nohorio, Evaluation of path length made in sensor-had path-planning with the alternative following, In Proceedings of IEEE ICRA'O1, 2001, PP. 909-916.

[111]. B. Margaret Devi, Prabakar S, Dynamic point bug algorithm for robot navigation, International journal of scientific & engineering research, vol. 4, no. 4, 2013, PP. 1276-1279

[112]. T.G.Zheng, H. Huan and S. Aaron, Ant colony system algorithm for real-time globally optimal path planning of mobile robots, Acta Automatica Sincia, vol.33, no.3, 1996, PP. 279-284.

[113]. M.A. P. Garcia, O. Montiel, O.Castillo, R.Sepulveda, and P. Melin, path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation, applied soft computing ,2009, PP. 1102–1110.

[114]. Z. Juanping, F. Xiuhui and J. Ying, An improved ant colony optimization algorithm for mobile robot path planning, 3 rd international work shop on intelligent systems and applications, Wuhan, 2011, PP. 1-4.

[115]. A. Reshamwala and D.P.Vinchurkar, Robot path planning using an ant colony optimization approach: a survey, International journal of advanced research in artificial intelligence, vol. 2, no.3, 2013, PP.65-71.

[116]. S. H. Chia, K. L. Su, Jr. H. Guo and C. Y. Chung, Ant colony system based mobile robot path planning, 4th IEEE international conference on genetic and evolutionary computing, Shenzhen, 2010, PP. 210-213.

[117]. S. Dasgupta, S. Das, A. Abraham, and A. Biswas, Adaptive computational chemotaxis in bacterial foraging optimization: an analysis, 2000, PP. 1-37.

[118]. L. X .Dan, L. L. Yu, W. J. Gang and C. H. Ning, Mobile robot path planning based on adaptive bacterial foraging algorithm, Journal of south central university, vol. 20, no. 12, 2013, PP. 3394−3400.

[119]. A.Sharma1, S. Satav, Path navigation using computational intelligence, International journal of advanced research in computer science and software engineering, vol.2, no.7, 2012, PP.395-398.

[120]. J Kennedy and J Eberhart, Particle Swarm Optimization, Proceedings of IEEE International Conference on Neural Networks, Perth, Australia, 1995, PP.1942-1948

[121]. R Hassan, B Cohanim, O D Weck and G Venter, (2005) A comparison of particle swarm optimization and the genetic algorithm. In: 1st AIAA multidisciplinary design optimization specialist conference, Austin.

[122]. A. Alireza1, PSO with adaptive mutation and inertia weight and its application in parameter estimation of dynamic systems, Acta Automatica Sinica, vol.37, no.5, 2005, PP 541-549.

[123]. Y. Zhang, J. Xuan, G. Benildo, R. Clarke and W. R. Habtom, Reverse engineering module networks by PSO-RNN hybrid modelling, International conference on bioinformatics & computational biology, Las Vegas, USA, 2009 PP. 1-18.

[124]. Q. Wu, Car assembly line fault diagnosis based on robust wavelet SVC and PSO, Expert Systems with Applications, vol.37, no.7, 2010, PP 5423-5429

[125]. D.Y. Sha, and H.H. Lin, A multi-objective PSO for job-shop scheduling problems, Expert systems with applications, vol.37, no. 2, 2010, PP. 1065-1070.

[126]. G.K. Venayagamoorthy & S. Doctor, Navigation of mobile sensors using PSO and embedded PSO in a fuzzy logic controller, Industry Applications IEEE Conference, 39th IAS Annual Meeting, vol.2, 2004, PP. 1200 – 1206.

[127]. Q Zhang and S. Li, A global path planning approach based on particle swarm optimization for a mobile robot, Proceedings of the 7th WSEAS international conference on robotics, control & manufacturing technology, Hangzhou, China, 2007, PP. 263-267.

[128]. Y.Q. Qin, D.B. Sun, M. Lii and Y.G Cen, Path planning for mobile robot using the particle swarm optimization with mutation operator, Proceedings of the 3rd international conference on machine laming and cybernetics, Shanghai, 2004, PP. 2473-2478.

[129]. E. Masehian, and D. Sedighizadeh, Multi-Objective PSO- and NPSO-based algorithms for robot path planning, Advances in electrical and computer engineering, vol.10, no.4, 2010, PP.69-76

[130]. K Derr and M Manic, Multi-robot, multi-target particle swarm optimization search in noisy wireless environments, Proceedings of HIS'09, Catania, Italy, 2009, PP. 81-87

[131]. S. Doctor, G.K. Venayagamoorthy and G. Gudise, Optimal PSO for collective robotic search applications, Proceedings of congress on evolutionary computation, vol. 2, 2004, PP.1390-1395.

[132]. L. L. Smith, G. K. Venayagamoorthy and G. H Phillip, Obstacle avoidance in collective robotic search using particle swarm optimization, Proceeding of IEEE conference on swarm intelligence symposium, Indianapolis, USA, 2006.

[133]. L. Lu and D. Gong, Robot path planning in unknown environments using particle swarm optimization, Proceedings of 4th international conference on natural computation, Jinan, vol. 4, 2008, PP. 422-423

[134]. B. Lei and W. Li, A Fuzzy Behaviours Fusion Algorithm for Mobile Robot Real-time Path Planning in Unknown Environment, Proceedings of IEEE international conference on integration technology, 2007, Shenzhen, China, PP.173-178.

[135]. G. Narvydas, R. Simutis and V.Raudonis, Autonomous mobile robot control using fuzzy logic and genetic algorithm, IEEE International workshop on intelligent data acquisition and advanced computing systems: technology and applications, Dortmund, Germany, 2007, PP 460-464.

[136]. M. S. Islam, M. Zaman, B. Madon, and M. Othman, Designing fuzzy based mobile robot controller , IEEE international conference on semiconductor electronics, Kuala lumpur, 2006, PP. 825-829.

[137]. N. Ouadah, L. Ourak and F. Boudjema, Car-Like mobile robot oriented positioning by fuzzy controllers, International journal of advanced robotic systems, vol. 5, no. 3, 2008, PP. 249-256

[138]. S. K. Pradhan, D. R. Parhi and A. K. Panda, Fuzzy logic techniques for navigation of several mobile robots, Applied soft computing, vol. 9, 2009, PP. 290–304.

[139]. A. Martinez, E. Tunstel, and M. Jamshidi (1994), Fuzzy logic based collision avoidance for a mobile robot, Third international conference on industrial fuzzy control and intelligent sytems, Houston, 1993, PP. 66-69.

[140]. D. Janglova, Neural networks in mobile robot motion, International journal of advanced robotic systems, vol.1 no.1, 2004, PP. 15-22.

[141]. I. Engedy and G. Horvath, Artificial neural network based mobile robot navigation, 6th IEEE international symposium on intelligent signal processing, Budapest, Hungary, 2009, PP.241-246.

[142]. M.M.Joshi and M.A.Zaveri, Optimally learnt neural network based autonomous mobile robot navigation system, ACEEE International journal on control system and instrumentation, vol.2, no.1, 2011, PP. 28-32.

[143]. C. Ng Kim and M. Trivedi Mohan, A neural-fuzzy controller for real-time mobile robot navigation, IEEE transactions on systems, man and cybernetics, vol. 28, no. 6, 1998, PP. 829-840.

[144]. K. K. Tahboub and M. S. N. Al-Din, A neuro- fuzzy reasoning system for mobile robot navigation, Jordan journal of mechanical and industrial engineering, vol.3, no.1, 2005, PP.77 – 88.

[145]. N. Zhang, D. Beetner, C. Donald, B. Hemmelman and A. Hasan, An embedded real-time neuro-fuzzy controller for mobile robot navigation, 14th IEEE international conference on fuzzy systems, Reno, 2005, PP. 319-324.

[146]. G.G. Acosta, J.F. Leon, and M.A. Mayosky, Artificial immune system inspired behaviour coordination for autonomous mobile robot trajectory generation, Proceedings of IEEE congress on evolutionary computation, DOI: 10.1109/CEC.2010.5586353, 2010, pp 1 – 6.

[147]. G.C. Luh and W.W. Liu, Reactive immune network based mobile robot navigation, LNCS 3239, Springer-Verlag Berlin Heidelberg, 2004, PP. 119–132.

[148]. D. Mamady, G. Tan and L.T. Mohamed, An artificial immune system based multi-agent model and its application to robot cooperation problem, Proceedings of the 7th world congress on intelligent control and automation, 2008, PP. 3033-3039.

[149]. A.V. Patrícia, L.N. Castro, R. Michelan, and F.J.V. Zuben, An immune learning classifier network for autonomous navigation, Proceedings of ICARIS 2003, LNCS 2787, Springer-Verlag Berlin Heidelberg, PP. 69–80.

[150]. D. Zeng, X. Gang, X. Cunxi, and Y. Degui, Artificial immune algorithm based robot obstacle-avoiding path planning, Proceedings of the IEEE international conference on automation and logistics, China, 2008, PP. 798-803.

[151]. R. Michelan, and F.J. Zuben, Decentralized control system for autonomous navigation based on an evolved artificial immune network. Proceedings of 2002 congress on evolutionary computation, vol. 2, PP.1021-1026.

[152]. Y. Watanabe, A. Ishiguro, Y. Shirai, and Y. Uchikawa, Emergent construction of behaviour arbitration mechanism based on the immune system. Proceedings of IEEE international conference on evolutionary computation, 1998, PP. 481–486.

[153]. BBVL Deepak, DR Parhi and S Kundu, Innate immune based path planner of an autonomous mobile robot, Elsevier procedia engineering, vol.38, 2012, PP. 2663-2671.

[154]. C.T. Singh, and S.B. Nair, An artificial immune system for a multi agent robotics system, World academy of science, engineering and technology, vol. 11, 2007, PP. 11-22.

[155]. M. Krautmacher, and W. Dilger, AIS based robot navigation in a rescue scenario. ICARIS 2004. LNCS, vol. 3239, Springer, Heidelberg, 2004, PP. 106 – 118.

[156]. A. M. Whitbrook, U. Aickelin, and J. M. Garibaldi, Idiotypic immune networks in mobile robot control, IEEE transactions on systems, man and cybernetics , vol. 37, no. 6, 2007, PP. 1581-1598.

[157]. H. C. Huang, T. F. Wu and C. Y. Huang, A Metaheuristic Artificial Immune System Algorithm for Mobile Robot Navigation, 2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), DOI: 10.1109/IIH-MSP.2014.203, PP: 803- 806

[158]. P.K. Das, A. Konar, and R. Laishram, Path planning of mobile robot in unknown environment, International journal of computer & communication technology, vol.1, no.2,3,4, 2010, PP. 26-31.

[159]. H. Secchi, R. Carelli, V. Mut, An experience on stable control of mobile robots, Latin American applied research, vol.33, no.4, 2003, PP. 379-385.

[160]. A. M. Zawawi, H. L. Sang, and Y. H. Hung, Autonomous mobile robot system concept based On PSO path planner and vSLAM, proceedings of IEEE international conference on computer science and automation engineering, July 2011, Shanghai, PP. 92 – 97.

[161]. B.Q. Yong, L.S. Ming, S.W. Yan and A.M. Jin, A fuzzy behavior-based architecture for mobile robot navigation in unknown environments, Proceedings of international conference on artificial intelligence and computational intelligence, November.2009, PP.257 - 261.

[162]. G. Mester, and A. Rodić, Sensor-Based intelligent mobile robot navigation in unknown environments, International journal of electrical and computer engineering systems, vol.1, no. 2, 2010, PP. 55-62.

[163]. W. Wahab, Autonomous mobile robot navigation using a dual artificial neural network, Proceedings of IEEE Region 10th Conference TENCON 2009, Singapore, DOI: 10.1109/TENCON.2009.5395892, ISBN: 978-1-4244-4546-2

[164]. B. Q. Yong, S. M. Li, W. Y. Shang and M. J. An, A fuzzy behavior-based architecture for mobile robot navigation in unknown environments, Proceedings of international conference on artificial intelligence and computational intelligence, vol. 2, DOI: 10.1109/AICI.2009.125, 2009, PP.257 – 261.

[165]. L. Deng, X. Ma, J. Gu and Y.Li, Mobile robot path planning using polyclonal-based artificial immune network, Journal of control science and engineering, volume 2013 (2013), Article ID 416715.

Appendix – A: Global Transformation Matrix of 4-Axis Manipulator

In chapter – 3 rotational kinematics and the homogeneous transformations have been discussed to relate the motion of one frame of reference to another frame of reference. Since the develop manipulator is of 4 DOF, four reference frames have been considered corresponds to each DOF. The end-effector motion is a function of kinematic parameters (two of link parameters and two of joint parameters) as illustrated in Table.3.2. The global transformation matrix of the manipulator is represented in Eq.(3.9) as follows:

$T_{base}^{Tool} = T_{base}^{wrist} * T_{wrist}^{tool}$

Where $T_{base}^{wrist} = T_0^1 * T_1^2$ and $T_{wrist}^{tool} = T_2^3 * T_3^4$

$$T_0^1 = \begin{bmatrix} C\theta_1 & -S\theta_1 C\alpha_1 & S\theta_1 S\alpha_1 & a_1 C\theta_1 \\ S\theta_1 & C\theta_1 C\alpha_1 & -C\theta_1 S\alpha_1 & a_1 S\theta_1 \\ 0 & S\alpha_1 & C\alpha_1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_0^2 = \begin{bmatrix} C\theta_2 & -S\theta_2 C\alpha_2 & S\theta_2 S\alpha_2 & a_2 C\theta_2 \\ S\theta_2 & C\theta_2 C\alpha_2 & -C\theta_2 S\alpha_2 & a_2 S\theta_2 \\ 0 & S\alpha_2 & C\alpha_2 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2^3 = \begin{bmatrix} C\theta_3 & -S\theta_3 C\alpha_3 & S\theta_3 S\alpha_3 & a_3 C\theta_3 \\ S\theta_3 & C\theta_3 C\alpha_3 & -C\theta_3 S\alpha_3 & a_3 S\theta_3 \\ 0 & S\alpha_3 & C\alpha_3 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_3^4 = \begin{bmatrix} C\theta_4 & -S\theta_4 C\alpha_4 & S\theta_4 S\alpha_4 & a_4 C\theta_4 \\ S\theta_4 & C\theta_4 C\alpha_4 & -C\theta_4 S\alpha_4 & a_4 S\theta_4 \\ 0 & S\alpha_4 & C\alpha_4 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Assume $[A] = [T_0^1] * [T_0^2] = \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{33} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{bmatrix}$ (A1)

And $[B] = [T_2^3] * [T_3^4] = \begin{bmatrix} B_{11} & B_{12} & B_{13} & B_{14} \\ B_{21} & B_{22} & B_{23} & B_{24} \\ B_{31} & B_{33} & B_{33} & B_{34} \\ B_{41} & B_{42} & B_{43} & B_{44} \end{bmatrix}$ (A2)

Where

$A_{11} = C\theta_1 C\theta_2 - S\theta_1 S\theta_2 C\alpha_1$

$A_{12} = C\theta_1(-S\theta_2 C\alpha_2) - S\theta_1 C\alpha_1 C\theta_2 C\alpha_2 + S\theta_1 S\alpha_1 S\alpha_2$

$A_{13} = C\theta_1 S\theta_2 S\alpha_2 + S\theta_1 C\alpha_1 C\theta_2 S\alpha_2 + S\theta_1 S\alpha_1 C\alpha_2$

$A_{14} = C\theta_1 a_2 C\theta_2 - S\theta_1 C\alpha_1 a_2 S\theta_2 + S\theta_1 S\alpha_1 d_2 + a_1 C\theta_1$

$A_{21} = S\theta_1 C\theta_2 + C\theta_1 C\alpha_1 S\theta_2$

$$A_{22} = -S\theta_1 S\theta_2 C\alpha_2 + C\theta_1 C\alpha_1 C\theta_2 C\alpha_2 - C\theta_1 S\alpha_1 S\alpha_2$$

$$A_{23} = S\theta_1 S\theta_2 S\alpha_2 - C\theta_1 C\alpha_1 C\theta_2 S\alpha_2 - C\theta_1 S\alpha_1 C\alpha_2$$

$$A_{24} = S\theta_1 a_2 C\theta_2 + C\theta_1 C\alpha_1 a_2 S\theta_2 - C\theta_1 S\alpha_1 d_2 + a_1 S\theta_1$$

$$A_{31} = S\theta_2 S\alpha_1$$

$$A_{32} = C\alpha_1 C\theta_2 S\alpha_1 + C\alpha_1 C\alpha_2$$

$$A_{33} = -S\alpha_1 C\theta_2 S\alpha_2 + C\alpha_1 C\alpha_2$$

$$A_{34} = a_2 S\alpha_1 S\theta_2 - d_2 C\alpha_1 + d_1$$

$$A_{41} = A_{42} = A_{43} = 0$$

$$A_{44} = 1$$

Matrix B:

$$B_{11} = C\theta_3 C\theta_4 - C\theta_3 S\theta_4 C\alpha_4$$

$$B_{12} = -C\theta_3 S\theta_4 C\alpha_4 - S\theta_3 C\alpha_3 C\theta_4 C\alpha_4 + S\theta_3 S\alpha_3 S\alpha_4$$

$$B_{13} = C\theta_3 S\theta_4 S\alpha_4 + S\theta_3 C\alpha_3 C\theta_4 S\alpha_4 + S\theta_3 S\alpha_3 C\alpha_4$$

$$B_{14} = a_4 C\theta_3 C\theta_4 - a_4 S\theta_3 C\alpha_3 S\theta_4 + S\theta_3 S\alpha_3 d_4 + a_3 C\theta_3$$

$$B_{21} = S\theta_3 C\theta_4 + C\theta_3 C\alpha_3 S\theta_4$$

$$B_{22} = -S\theta_3 S\theta_4 C\alpha_4 + C\theta_3 C\alpha_3 C\theta_4 C\alpha_4 - C\theta_3 S\alpha_3 S\alpha_4$$

$$B_{23} = S\theta_3 S\theta_4 S\alpha_4 - C\theta_3 C\alpha_3 C\theta_4 S\alpha_4 - C\theta_3 S\alpha_3 C\alpha_4$$

$$B_{24} = a_4 S\theta_3 C\theta_4 + a_4 C\theta_3 C\alpha_3 S\theta_4 - C\theta_3 S\alpha_3 d_4 + a_3 S\theta_3$$

$$B_{31} = S\alpha_3 S\theta_4$$

$$B_{32} = S\alpha_3 C\theta_4 C\alpha_4 + S\alpha_3 S\alpha_4$$

$$B_{33} = -S\alpha_3 C\theta_4 S\alpha_4 + C\alpha_3 C\alpha_4$$

$$B_{34} = a_4 S\alpha_3 S\theta_4 - C\alpha_3 d_4 + d_3$$

$$B_{41} = a_4 C\theta_4$$

$$B_{42} = a_4 S\theta_4$$

$$B_{43} = d_4$$

$$B_{44} = 1$$

Therefore the global transformation matrix of 4-axis manipulator is product of Eqs. (A2) & (A3) as follows:

$$T_{base}^{Tool} = T_{base}^{wrist} * T_{wrist}^{tool} = [A] * [B] \quad \text{(A3)}$$

Eq.(3.10) represents the global transformation matrix which maps from its base coordinate frame of reference to toll coordinate reference frame as below

$$T_{base}^{tool} = \begin{bmatrix} m_x & n_x & o_x & p_x \\ m_y & n_y & o_y & p_y \\ m_z & n_z & o_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R(\theta)_{3x3} & P_{3x1} \\ 0 & 1 \end{bmatrix}$$

Comparing the Eq.(3.10) with Eq.(A3) the elements of the rotation and position matrix is as follows:

$$T_{base}^{tool} = \begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{33} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{bmatrix} * \begin{bmatrix} B_{11} & B_{12} & B_{13} & B_{14} \\ B_{21} & B_{22} & B_{23} & B_{24} \\ B_{31} & B_{33} & B_{33} & B_{34} \\ B_{41} & B_{42} & B_{43} & B_{44} \end{bmatrix} \quad \text{(A4)}$$

$$m_x = A_{11}B_{11} + A_{12}B_{21} + A_{13}B_{31}$$
$$n_x = A_{11}B_{12} + A_{12}B_{22} + A_{13}B_{33}$$
$$o_x = A_{11}B_{13} + A_{12}B_{23} + A_{13}B_{33}$$
$$p_x = A_{11}B_{14} + A_{12}B_{24} + A_{13}B_{34}$$
$$m_y = A_{21}B_{11} + A_{22}B_{21} + A_{23}B_{31}$$
$$n_y = A_{21}B_{12} + A_{22}B_{22} + A_{23}B_{33}$$
$$o_y = A_{21}B_{13} + A_{22}B_{23} + A_{23}B_{33}$$
$$p_y = A_{21}B_{14} + A_{22}B_{24} + A_{23}B_{34}$$
$$m_z = A_{31}B_{11} + A_{32}B_{21} + A_{33}B_{31}$$
$$n_z = A_{31}B_{12} + A_{32}B_{22} + A_{33}B_{33}$$
$$o_z = A_{31}B_{13} + A_{32}B_{23} + A_{33}B_{33}$$
$$p_z = A_{31}B_{14} + A_{32}B_{24} + A_{33}B_{34}$$

Appendix – B: Kinematic Model of a Differential Mobile Platform

The motion of the wheeled mobile platform is discussed in chapter -3. A kinematic model of a wheeled mobile platform with corresponding parameters is represented in Fig. A.1. The two rear wheels are fixed parallel to robot body and allowed to roll or spin but not slip.

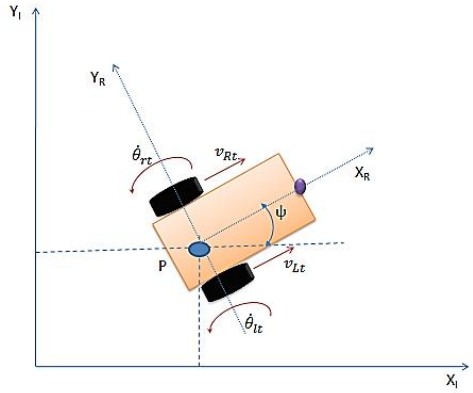


Fig.B.1 Kinematic model of a wheeled mobile platform

The configuration of the robot moving on a plane surface at every instant is defined by three parameters (x, y, ψ). The rear wheel is always tangent to the orientation of the vehicle. No slipping condition mentioned previously requires that the robot navigate in the direction of its wheels. Let $\dot{\theta}_{Rt}, v_{Rt}$ and $\dot{\theta}_{Lt}, v_{Lt}$ are the left and right wheel rotational and linear velocities respectively. By changing the velocities of the two wheels, the instantaneous center of rotation will move and different trajectories will be followed.

At each moment of time the left and right wheels follow a path that moves around its instantaneous centre of rotation with a radius of curvature.

$$R = \left((v_{Rt} + v_{Lt}) \Big/ (v_{Rt} - v_{Lt}) \right) \frac{S}{2} \qquad \text{(B1)}$$

The velocity of the centre of mass of the platform, which is the midpoint between the two wheels, can be calculated as follows:

$$\omega = (v_{Rt} - v_{Lt})/S \qquad \text{(B2)}$$

$$v = (v_{Rt} + v_{Lt})/2 \qquad \text{(B3)}$$

If $v_{Rt} = v_{Lt}$ then the radius R is infinite and the robot moves in a straight line. For various values of v_{Rt}& v_{Lt}, the mobile platform does not move in a straight line but rather follows a curved trajectory around a point located at a distance R from centre point. If $v_{Rt} = -v_{Lt}$, then the radius R is zero and the robot rotates around one wheel. For any real value of the velocity, R must be real, to get a real curved path. Thus, ψ has to lie within 0^0 and 90^0, for a non-holonomic robot.

APPENDIX – C: 8051 Microcontroller Pin Configuration

Schematic diagram of an 8051 microcontroller pin configuration is shown in Fig.C.1 and the detailed description about each pin is as follows:

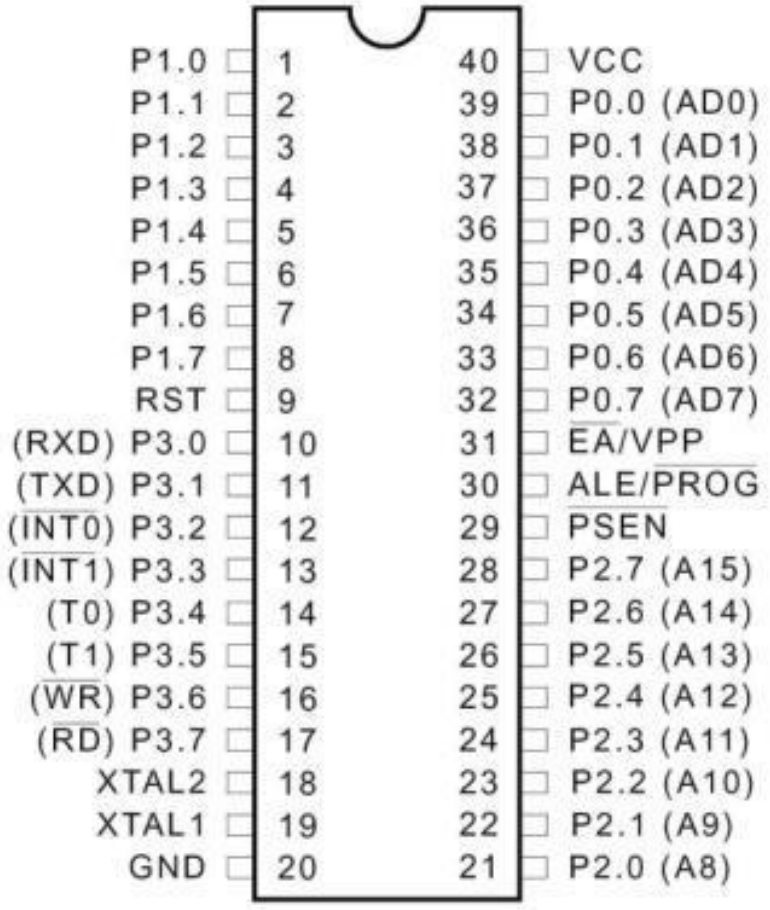


Fig.C.1 8051 microcontroller pin diagram

Description of each pin

→ Port 0 (P0.0 to P0.7): It is an 8-bit bi-directional addressable I/O port. During external memory access, it functions as multiplexed data and low-order address bus AD0-AD7.

→ Port 1 (P1.0 to P1.7): It is an 8-bit bi-directional bit/ byte addressable I/O port. When logic 1is written into port latch then it work as input mode.

→ Port 2 (P2.0 to P2.7): It is an 8-bit bi-directional addressable I/O port. During external memory access it functions as higher order address bus (A8-A15).

→ Port 3 (P3.0 to P3.7): It is an 8-bit I/O port. In which an alternating function of each pins can be used as a special function I/O pin.

→ P3.0-RxD: This I/P pin receives serial data of serial communication circuit.

→ P3.1-TxD: Through this O/P pin data is transmitted.

→ P3.2- (INT0): It is external hardware interrupt I/P pin used for interrupt generation.

→ P3.3-(INT1): It is external hardware interrupt I/P pin used for interrupt generation.

→ P3.4- T0: External clock pulses can connect to timer-0 through this I/P pin.

→ P3.5-T1: External clock pulses can connect to timer-1 through this I/P pin.

→ P3.6-(WR): It is an active low O/P control pin used during external RAM (Data memory) access. If (WR) =0, then microcontroller will perform write operation to the external RAM.

→ P3.7-(RD): It is an active low O/P control pin used during external RAM (Data memory) access. If (RD) =0, then microcontroller will perform read operation from the external RAM.

→ XTAL 1 and XTAL 2: These are two I/P line for on-chip oscillator and clock generator circuit. A resonant network as quartz crystal is connected between these two pin.

→ (EA)/VPP: It is an active low I/P pin. When (EA) =0, 8051 microcontroller access from external program memory (ROM). If EA =1, it access internal and external program memories (ROMS).

→ (PSEN): It is an active low O/P pin. It is used to enable external program memory (ROM). When (PSEN) =0, then external program memory becomes enable and microcontroller read content of external memory location. Therefore it is connected to (OE) of external ROM. It is activated twice for every external ROM memory cycle.

→ ALE-Address latch enable: It is an active high O/P pin. When it goes high, external address latch becomes enabling and lower address of external memory (RAM or ROM) latched into it. Thus it separates A0-A7 address from AD0-AD7. It provides proper timed signal to latch lower byte address. The ALE is activated twice for every machine cycle. If external RAM & ROM are not accessed, then ALE is activated at constant rate of $1/6^{th}$ oscillator frequency, which can be used as a clock pulses for driving external devices.

→ RESET: It is an active high I/P signal. It should be maintained high for at least two machine cycles in order to reset the microcontroller, i.e. it clears the internal registers.

List of Publications

Published in International Journals:

1. **BBVL Deepak**, D R Parhi and B M V A Raju, Advance particle swarm optimization based navigational controller for mobile robot, Arabian Journal for Science and Engineering, Volume 39, No. 8, pp 6477-6487, 2014.
2. Elias Eliot, **B.B.V.L. Deepak**, D.R. Parhi, J. Srinivas, Design & kinematic analysis of an articulated robotic manipulator, International Journal of Mechanical and Industrial Engineering, Vol-3, No-1, , pp. 105-108, 2013.
3. **B. B. V. L. Deepak** and Dayal Parhi, Intelligent adaptive immune-based motion planner of a mobile robot in cluttered environment, Intelligent Service Robotics, Vol. 6, No. 3, pp. 155-16, 2013.
4. D.R Parhi and **BBVL Deepak**, "Path Generation of a Differential Mobile Robot using Particle Swarm Optimization", International Journal of Artificial Intelligence and Computational Research, Vol.4 No.1, pp: 7-11, 2012.
5. **BBVL Deepak** and Dayal R. Parhi, PSO Based Path Planner of an Autonomous Mobile Robot, Central European Journal of Computer Science, Vol. 2, No.2, pp. 152-168, 2012.
6. **BBVL Deepak**, Dayal R. Parhi and and Anand Amrit, Inverse Kinematic Models for Mobile Manipulators, Caspian Journal of Applied Sciences Research, Vol. 1, No.13, pp. 151-158, 2012.
7. D.R. Parhi, **BBVL Deepak**, D. Nayak and and A. Amrit, Forward and Inverse Kinematic Models for an Articulated Robotic Manipulator, International Journal of Artificial Intelligence and Computational Research, Vol.4 No.2, pp: 103-109, 2012.
8. **BBVL Deepak** and Dayal R Parhi, "Kinematic Model of Wheeled Mobile Robots", International Journal of Recent Trends in Engineering & Technology, Vol. 05, No. 04, pp.5-10, 2011.
9. D.R Parhi and **BBVL Deepak**, "Kinematic Model of Three Wheeled Mobile Robot", Journal of Mechanical Engineering Research, Vol. 3, No.9, pp. 307-318, 2011.
10. D.R Parhi and **BBVL Deepak**, "Sugeno Fuzzy Based Navigational Controller of an Intelligent Mobile Robot", International Journal of Applied Artificial Intelligence in Engineering System, Vol. 3, No.2, pp. 103-108, 2011.

11. D.R Parhi, JC Mohanta, **BBVL Deepak** and S K Patel, "Analysis of Hybrid Genetic Technique for Navigation of Intelligent Autonomous Mobile Robots", International Journal of Applied Artificial Intelligence in Engineering System, Vol. 02, No. 02, pp 133-136, 2010.
12. D.R Parhi, JK Pothal and **BBVL Deepak**, "Navigation of Mobile Robots using Fuzzy-Ant Optimization Technique", International Journal of Applied Artificial Intelligence In Engineering System, pp 111-117, 2010.

(B) PAPERS UNDER REVISION

1. **BBVL Deepak** and D R Parhi, Control of an Automated Mobile Manipulator Using Artificial Immune System, journal of experimental and theoretical artificial intelligence, Taylor & Francis Publications.

Book Chapters Published:

1. Dayal R. Parhi, **B.B.V.L. Deepak**, Jagan Mohana, Rao Ruppa, and Meera Nayak, "Immunised Navigational Controller for Mobile Robot Navigation", S. Patnaik & Y.-M. Yang (Eds.): Soft Computing Techniques in Vision Sci., SCI 395, Springer-Verlag Berlin Heidelberg 2012, pp. 171–182, ISBN: 978-3-642-25506-9 (Print) 978-3-642-25507-6 (Online).

(C)Published in International Conferences:

1. **B.B.V.L. Deepak** and Dayal R. Parhi, Target Seeking Behaviour of an Intelligent Mobile Robot Using Advanced Particle Swarm Optimization, Proceedings of IEEE sponsored International Conference on CARE 2013, IIIT Jabalpur.
2. Elias E., **B.B.V.L. Deepak**, D.R. Parhi, and J. Srinivas, Design & Kinematic Analysis of an Articulated Robotic Manipulator, Proceedings of International Conference on Mechanical and Industrial Engineering (ICMIE-2012), Goa, pp. 261-264.
3. **BBVL Deepak**, Dayal R Parhi, and Subhasri K. "Innate immune based path planner of an Autonomous Mobile Robot", International Conference on Modelling, Optimisation and Computing (ICMOC – 2012), NI University, ELSEVIER Procedia Engineering 38(2012), pp. 2663 – 2671.

4. **BBVL Deepak,** Dayal R Parhi and Devidutta N., Development of Forward and Inverse Kinematic Models for 5-Axis Articulated Manipulator, Proceedings of 4th International and 25th All India Manufacturing Technology, Design and Research (AIMTDR) Conference, jadavpur university, 2012, pp.102-106.
5. **BBVL Deepak**, Dayal R Parhi, and J R Ruppa, "Immunised Navigational Controller for Mobile Robot Navigation", Proceedings of International Conference on Artificial Intelligence and Soft Computing (ICAISC 2011),Bhubaneswar, pages. 259-264.
6. J R Ruppa, DR Parhi and **BBVL Deepak**, "Ant Colony Optimization Algorithm for the Travelling Salesman Problem", Proceedings of International Conference on Artificial Intelligence and Soft Computing (ICAISC 2011), Bhubaneswar, pages. 282-285.
7. Kalpana S, **BBVL Deepak**, and Dayal R Parhi, "PSO Based Motion Planner of an Intelligent Mobile Robot", Advances in Modeling, Optimization and Computing (AMOC - 2011), IIT-Roorkee, pp. 187-191.

(D) Published in National Conferences:

1. **BBVL Deepak**, Alok K J and D R Parhi, "Mobile Robot Obstacle Avoidance using Particle Swarm Optimization", Proceedings of National Conference on Emerging Trends in Computing and Information Technology (NCETCIT-2011). RKGITW, Ghaziabad, pp. 40-44.
2. **BBVL Deepak**, Alok K J and D R Parhi, "Path Planning of an Autonomous Mobile Robot using Artificial Immune System", Proceedings of National Conference on Emerging Trends in Computing and Information Technology (NCETCIT-2011). RKGITW, Ghaziabad, pp. 45-48.
3. **BBVL Deepak**, A. Amrit, N. Kumar and D R Parhi, "Development of Forward Kinematic Model for an Articulated Robotic Manipulator", Proceedings of 3rd National Conference on Recent Advances in Manufacturing (RAM 2012), SVNIT, Surat, pp. 86-90.
4. **BBVL Deepak**, A. Amrit, N. Kumar and D R Parhi, "Path Generation of a Differential Mobile Robot using Fuzzy Inference System", Proceedings of 3rd National Conference on Recent Advances in Manufacturing (RAM 2012), SVNIT, Surat, pp. 86-90.

Arab J Sci Eng (2014) 39:6477–6487
DOI 10.1007/s13369-014-1154-z

RESEARCH ARTICLE - SYSTEMS ENGINEERING

Advance Particle Swarm Optimization-Based Navigational Controller For Mobile Robot

B. B. V. L. Deepak · Dayal R. Parhi · B. M. V. A. Raju

Received: 3 November 2012 / Accepted: 15 July 2013 / Published online: 3 June 2014

Abstract While the robot is in motion, path planning should follow the three aspects: (1) acquire the knowledge from its environmental conditions. (2) determine its position in the environment and (3) decision-making and execution to achieve its highest-order goals. The present research work aims to develop an efficient particle swarm optimization-based path planner of an autonomous mobile robot. In this approach, a fitness function has been introduced for converting the mobile robot navigation problem into multi objective optimization problem. The fitness of the swarm mainly depends on two parameters: (1) distance between each particle of the swarm and target, (2) distance between each particle of the swarm and the nearest obstacle. From the obtained fitness values of the swarm, the global best position of the particle is selected in each cycle. Thereby, the robot reaches the global best position in sequence. The effectiveness of the developed algorithm in various environments has been verified by simulation modes.

Keywords Mobile robot navigation · Local path planning · Obstacle avoidance · Particle swarm optimization · Swarm intelligence

B. B. V. L. Deepak (✉) · D. R. Parhi
Robotics Lab., Department of Mechanical Engineering, National Institute of Technology, Rourkela, India
e-mail: deepakjournal@gmail.com

B. M. V. A. Raju
Department of Industrial Design, National Institute of Technology, Rourkela, India

الخلاصة

يجب أن يتبع تخطيط المسار خلال حركة الروبوت ثلاثة جوانب: 1) اكتساب المعرفة من ظروفه المحيطية. 2) تحديد موضعه في البيئة المحيطة. 3) اتخاذ القرار والتنفيذ لتحقيق الأهداف الأعلى ترتيبا. يهدف هذا العمل البحثي إلى تطوير مخطط مسار يعتمد على سرب جسيمات أمثل فعال لروبوت متنقل ذاتي الحركة. وقد تم – ضمن هذا النهج – عرض اقتران ملائمة لتحويل مسألة ملاءمة الروبوت المتنقل إلى مسألة أمثلة متعددة الأهداف. وتعتمد مواءمة السرب بشكل أساسي على معلمين اثنين هما: 1) المسافة بين كل جسيم من السرب والهدف، 2) المسافة بين كل جسيم من السرب والعائق الأقرب. ومن خلال قيم ملاءمة السرب التي يتم الحصول عليها يُختار أفضل موضع كلي للجسيم في كل دورة. وبالتالي يصل الروبوت إلى أفضل موضع كلي بشكل متسلسل. وقد تم التحقق من فعالية الخوارزمية التي تم تطويرها في بيئات متنوعة باستخدام وسائط محاكاة.

List of Symbols

C_1	Cognitive parameter considering in position shift
C_2	Social parameter considering in position shift
F_i	ith particle primary fitness value
F_{final}	Final fitness function
(goalx, goaly)	Robot destination position in its work space
λ_{R-Ob}	Distance between robot and sensed obstacle
λ_{P_i-T}	Distance between ith particle and target position.
λ_{P_i-NOb}	Distance between ith particle and nearest obstacle
P	Population size/number of particles in the swarm.
NOb	Nearest obstacle within robots sensing range
(NOb_x, NOb_y)	Nearest obstacle centre in robotic environment

Intel Serv Robotics (2013) 6:155–162
DOI 10.1007/s11370-013-0131-9

ORIGINAL RESEARCH

Intelligent adaptive immune-based motion planner of a mobile robot in cluttered environment

B. B. V. L. Deepak · Dayal Parhi

Received: 9 September 2012 / Accepted: 7 May 2013 / Published online: 25 May 2013

Abstract Learning of an autonomous mobile robot for path generation includes the use of previous experience to obtain the better path within its work space. When the robot is moving in its search space for target seeking, each task requires different form of learning. Therefore, the modeling of an efficient learning mechanism is the hardest problem for an autonomous mobile robot. To solve this problem, the present research work introduced an adaptive learning-based motion planner using artificial immune system, called adaptive immune-based path planner. Later the developed adaptive mechanism has been integrated to the innate immune-based path planner in order to obtain the better results. To verify the effectiveness of the proposed adaptive immune-based motion planner, simulation results as well as experimental results are presented in various unknown environments.

Keywords Mobile robot navigation · Adaptive learning mechanism · Artificial immune system

1 Introduction

Motion planning is one of the vital issues in the field of mobile robots because of their usage in various fields such as domestic fields, industries, security environments, hospitals, etc. The main goal of an efficient motion planner of a mobile robot is to generate collision-free trajectories from the sensory information without continuous human intervention.

Electronic supplementary material The online version of this article (doi:10.1007/s11370-013-0131-9) contains supplementary material, which is available to authorized users.

B. B. V. L. Deepak (✉) · D. Parhi
National Institute of Technology, Rourkela, India
e-mail: bbv@nitrkl.ac.in

To solve mobile robot navigation problem different algorithms have been developed from the last few decades. In which, artificial potential fields [11] are widely used because of its easy implementation and simple in structure, but the robot may face local minima situation in certain conditions. Chris [3] has implemented genetic algorithm-based motion controller for an autonomous mobile robot. Ellips and Davoud [5,23] have presented a new path planner based on particle swarm optimization. Yong et al. [16] have introduced a behavior-based architecture using fuzzy logic for mobile robot navigation in unknown environments. Wahab [14] has dealt with neural network-based intelligent control of a mobile robot to find its target within its environment. Beside these, many algorithms were proposed for solving motion planning problem as described in [4]. Most of these algorithms have been implemented by adjusting the controlling parameters to optimized values in order to obtain efficient results. Local minima is one of the well-known drawback in mobile robot navigation; in this situation the robot may get trapped in its maze environment or it wanders indefinitely in a region. As explained by Luh and Liu [7], many of the developed path planners did not consider local minima problem solving.

In recent years, researchers have applied artificial immune system algorithms to autonomous mobile robot for generating collision-free trajectories [1,8,9,12,17,18]. While developing immunological system architectures, they have been investigated the interactions among the various immune components. Mamady et al. [9] have developed a new immunized computational method for mobile robot navigation, if the robot environment having uniform mass and general shape objects. To get better results from the methodology as described in [10], it is necessary to evolve the immune network by the presence of much more connections, but this will increase the network complexity. An adaptive AIS

Cent. Eur. J. Comp. Sci. • 2(2) • 2012 • 152-168
DOI: 10.2478/s13537-012-0009-5

VERSITA

Central European Journal of **Computer Science**

PSO based path planner of an autonomous mobile robot

Research Article

BBVL Deepak[1]*, Dayal R. Parhi[2]†

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

2 *Department of Mechanical Engineering, National Institute of Technology-Rourkela, Odisha-769008, India*

Received 19 October 2011; accepted 03 March 2012

Abstract: A novel approach based on particle swarm optimization has been presented in this paper for solving mobile robot navigation task. The proposed technique tries to optimize the path generated by an intelligent mobile robot from its source position to destination position in its work space. For solving this problem, a new fitness function has been modelled, which satisfies the obstacle avoidance and optimal path traversal conditions. From the obtained fitness values of each particle in the swarm, the robot moves towards the particle which is having optimal fitness value. Simulation results are provided to validate the feasibility of the developed methodology in various unknown environments.

Keywords: path planning • autonomous mobile robot • particle swarm optimization • obstacle avoidance

1. Introduction

Motion planning of an intelligent mobile robot is one of the most vital issues in the field of robotics, which includes the generation of optimal collision free trajectories within its work space and finally reaches its target position. Based on this issue the path planning can be classified into two categories: global path planning and local path planning. In the former type, the robot generates the path from its source position to goal position within its known static environments. In the latter, robot generates path trajectories within its completely unknown/ partially known environments.
Since last decades, numerous researches have been devoted to solve the mobile robot path planning problem and various techniques such as artificial potential field, visibility graphs and cell decomposition method etc. have been proposed. Potential field method is [11] widely used because of its simple structure and easy execution, but this approach may face the local minima problem, which leads to robot trap situation within its environments. Visibility graph [10] requires more control accuracy, because its search path efficiency is low as described in [12]. In the cell decomposition approach [5], the environment is divided into a number of cells which are predefined in size and shape. This method is not suitable

* *E-mail: deepakjournal@gmail.com*
† *E-mail: DRKPARHI@mitrkl.ac.in*

Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Engineering 38 (2012) 2663 – 2671

Procedia Engineering

www.elsevier.com/locate/proc

ICMOC - 2012

Innate immune based path planner of an Autonomous Mobile Robot

B B V L Deepak[a,b], Dayal R.Parhi[b], Shubhasri Kundu[b], a*

[a]*Dept. of Industrial Design, National Institute of Technology, Rourkela-769008, India*
[b]*Dept. of Mechanical Engineering, National Institute of Technology, Rourkela-769008, India*

Abstract

Path planning of mobile robot is related to generating safest trajectories within its work space by avoiding obstacles, escaping traps and finally reaches its destination within optimal period. While an autonomous mobile robot is motion, each robot task needs a different form of learning because of its environmental changes. To select suitable robotic action at different environmental situations, a new motion planner is described in the current research work. The proposed motion planner is motivated from the biological innate immune system. To actuate a suitable robotic action, one new parameter named as learning rate has been introduced, which correlates the robot sensory information and the pre-engineered robot actions. The further movement of the robot is then decided by selecting of a suitable robot predefined task, there by the robot will move in sequence until it reaches to its destination. Finally, the developed path planner is executed in various simulated robotic environments.

Key words: autonomous mobile robot; artificial immune system; mobile robot navigation; behaviour learning.

1. Introduction

Mobile robots are widely using in various fields such as domestic fields, industries, security environments and hospitals etc. because of their mobility nature. So motion planning is one of the vital issues in the field of mobile robots. In which, the robot should adapt the behavior learning from the sensory information without continuous human intervention. The main goal of a navigational controller of an autonomous mobile robot is to generate collision free trajectories within its workspace.

* Corresponding author. Tel.:+91-661-246-2514.
E-mail address: *deepak.bbvl@gmail.com*.

1877-7058 © 2012 Published by Elsevier Ltd.
doi:10.1016/j.proeng.2012.06.313

Immunised Navigational Controller for Mobile Robot Navigation

Dayal R. Parhi, B.B.V.L. Deepak, Jagan Mohana, Rao Ruppa, and Meera Nayak

Abstract. Over the last few years, the interest in studying the Artificial Immune System (AIS) is increasing because of its properties such as uniqueness, recognition of foreigners, anomaly detection, distributed detection, noise tolerance, reinforcement learning and memory. Previous research work has proved that AIS model can apply to behavior-based robotics, but implementation of idiotypic selection in these fields are very few.The present research aims to implement a simple system architecture for a mobile robot navigation problem working with artificial immune system based on the idiotypic effects among the antibodies and the antigens. In this architecture environmental conditions are modeled as antigens and the set of action strategies by the mobile robot are treated as antibodies. These antibodies are selected on the basis of providing the robot with the ability to move in a number of different directions by avoiding obstacles in its environment. Simulation results showed that the robot is capable to reach goal effectively by avoiding obstacles and escape traps in its maze environment.

Keywords: Artificial Immune System, Idiotypic effect, Immune Network, Robot Navigation.

1 Introduction

Artificial immune system (AIS) has been developed from the natural immune system to solve engineering problems efficiently. Based on these issues previous

Dayal R. Parhi · B.B.V.L. Deepak · Jagan Mohana · Rao Ruppa
Department of Mechanical Engineering,
National Institute of Technology- Rourkela
e-mail: DRKparhi@nitrkl.ac.in, deepak.bbvl@gmail.com

Meera Nayak
Lecturer, G.I.E.T., Bhubaneswar
e-mail: meeradash@gmail.com

S. Patnaik & Y.-M. Yang (Eds.): Soft Computing Techniques in Vision Sci., SCI 395, pp. 171–182.
springerlink.com

Caspian Journal of Applied Sciences Research, 1(13), pp. 151-158, 2012
Available online at http://www.cjasr.com
ISSN: 2251-9114, ©2012 CJASR

Full Length Research Paper

Inverse Kinematic Models for Mobile Manipulators

B.B.V.L.Deepak[1,2*], Dayal R. Parhi[1], Anand Amrit[2]

[1]Robotics Lab., Department of Mechanical Engineering, National Institute of Technology-Rourkela
[2]Department of Industrial Design, National Institute of Technology-Rourkela, India
*Corresponding Author: Email: bbv@nitrkl.ac.in; Tel.: 0661-246-2855

Received 25 September 2012; Accepted 14 November 2012

Proper motion planning algorithms are necessary for both robotic manipulators and mobile robots (as well their combination, i.e. mobile manipulators) in order to execute their specific tasks. To solve this problem, current research work introduces the inverse kinematic models for mobile manipulators. In general a systematic closed form solution is not available in the case of inverse kinematic problem. So the solution for inverse kinematic problem is more complex as compared to direct kinematics problem. The current research work aims to combine the functionality of a robot arm with an autonomous platform. It means development of an autonomous wheeled mobile robot on which the robot arm is mounted. The purpose of this work is to integrate both the segments (i.e. mobile manipulator & mobile platform), such that the system can perform the constrained moves of the arm in the mean while as the platform is moving.

Key words: Robotic manipulator, wheeled mobile robot, inverse kinematic models, end effector constraints, geometric wheel constraints.

1. INTRODUCTION

In the case of robotic manipulators, motion planning is a critical issue because of its end effectors path constraints. Whereas, the motion control of mobile robots or the mechanical behavior of the robot depends upon the wheel geometric constraints while the robot is in motion. To support the development and to enlarge the application prospective of industrial robotics, it is rational to combine locomotion capabilities with manipulation abilities, hereby creating mobile manipulators.

Compared to conventional industrial robots, it is easier for mobile manipulators to adapt to changing environments and perform a wide variety of manufacturing jobs (Hammer et al., 2009). Another benefit of this category of robot is that the existing industrial environments do not have to be altered or modified as in the case of Automated Guided Vehicles (AGV's), where permanent cable layouts and/or markers are required for navigation (Datta et al. 2008). In past (Edgar et al., 2010; Gracia and Tornero, 2008; Martinez et al. 2004; Thomas and Hugh, 1993), authors dealt with kinematic models of wheeled mobile robots to generate trajectory within its environments. Thomas and Hugh (1993) have derived the inverse and forward kinematics for modular wheeled mobile robots based on the theory of planar motion. Edgar et al. (2010) have dealt with the problem of trajectory control of a wheeled mobile robot which is having four skid-steerable driven wheels. Further they established an inverse kinematics formulation in the function of exert trajectory control, namely instantaneous linear velocity, angular velocity, and robot Z-turning axes. Gracia and Tornero (2008) aimed to develop a practical approach for motion planning of wheeled mobile robot based on avoiding singularities. In their work, it has been proposed a cost index that assesses the nearness to singularity of forward and inverse kinematic models. Martínez et al. (2004) proposed a kinematic approach for tracked vehicles in order to improve motion control and pose estimation based on the kinematic similarities between tracked vehicles and wheeled differential drive vehicles.

In automation industries, industrial robots are used in place of human workers due to their great speed capacity, precision and their cost-effectiveness in repetitive tasks. The manipulator performance depends on the technological, geometrical and environmental constraints as well as any other constraints inherent both to the robot design and to the nature of the task to be executed (Henten et al. 2003; Manfred et al. 2007; Ali et al. 2006; Solteiro et al. 2007; Parhi and Deepak 2011). Van Henten et al. (2003) presented an inverse kinematics algorithm that has been used in a functional model of a cucumber-harvesting robot

DESIGN & KINEMATIC ANALYSIS OF AN ARTICULATED ROBOTIC MANIPULATOR

ELIAS ELIOT[1], B.B.V.L. DEEPAK[2], D.R. PARHI[3] & J. SRINIVAS[4]

[1&2]Department of Industrial Design, National Institute of Technology-Rourkela
[3&4]Department of Mechanical Engineering, National Institute of Technology-Rourkela
E-mail : bbv@nitrkl.ac.in

Abstract-This paper describes the design, fabrication and analysis a five axes articulated robotic manipulator. The current work is undertaken by considering various commercially available robotic kits to design and fabricate a five degree of freedom (D.O.F) arm. Forward kinematic model has been presented in order to determine the end effectors position and orientation. Although this work is still in primary level, this analysis is useful for path tracking of an industrial manipulator with pick-and-place application. Based on this analysis, a researcher can develop path tracking behaviour of an end effector in complicated work space.

Keywords - *5-DOF robotic arm, 5-axes articulate robotic manipulator, kinematic analysis.*

I. INTRODUCTION

Industrial robots are not completely androids that mimic human, but are more anthropomorphic in nature, in the sense that they are designed with resemblance to a human hand; and are also incapable of self-movement.

The requirement graph for these industrial robots has always been an upward one. Faster robots with multiple functions to increase production and reduce manufacturing cost are the necessity of the day. Factors such as: better precision, accuracy and repeatability; maximum load carrying capacity and work space and versatile operating environments are being given utmost importance during the development of any industrial robot.

The history of industrial automation [1-2] is characterized by periods of rapid change in popular methods. Either as a cause or, perhaps, an effect, such periods of change in automation techniques seem closely tied to world economics. Use of the industrial robot, which became identifiable as a unique device in the 1960□s, (along with computer aided design (CAD) systems, and computer sided manufacturing (CAM) systems), characterizes the latest trends in the automation of the manufacturing process. Industrial robots were studied independently as complex manipulator arms by various authors. The kinematic modelling and analysis of a 5-axis stationary articulated robotic arm has been conducted by Manjunath [3]. Using C++ language, it was shown visually the kinematic model incorporating obstacle avoidance algorithms for the pick and place operation. Hernandez *et al.* [4] integrated two Barrett WAM arms on top of a Segway RMP mobile base by putting together power sources, computers, and distributed software systems. Instead of using locally engineered and built components, they used - commercially available components to assemble a mobile manipulator. Xu*et al.* [5] systematically analysed the forward and inverse kinematics of a five DOF manipulator and suggested an analytical solution for the manipulator to follow a given trajectory while keeping the orientation of one axis in the end-effector frame. Alpha II is a five axis articulate robot arm manufactured by Microbot [6] which has a variety of standard or specialized gripper mechanisms. It is a low-cost robot system designed specifically to help manufacturing operations management, improve productivity by automating low-level tasks that human workers find hazardous or difficult to repeat accurately for long periods of time. Rhino XR-3 [7] is also a five axis articulate robotic manipulator. This robotic manipulator has a rugged open design, which makes it very easy to study. Using this robot as a major reference all successive works have been carried out.

The present work aims to apply forward kinematics to a 5 DOF articulated manipulator. Simulation results are presented for the modelled manipulator which reprsents the path tracking of each individual link of the manipulator with respect to its base position. Although this work is still in primary level, this analysis is useful for path tracking of an industrial manipulator with 'pick-and-place' application.

II. DESIGN DETAILS

After giving a thorough consideration of all the preceding works in this field, a five degree of freedom multi-functional reprogrammable manipulator having variable programmed motions to carry out variety of tasks in diverse environments is chosen. This is a five axis articulate manipulator designed to move material like machine parts, tools,

I J A I C R, 4(2), 2012, pp. 103-109

Forward and Inverse Kinematic Models for an Articulated Robotic Manipulator

Dayal R. Parhi, BBVL Deepak, Devedutta Nayak and Anand Amrit

Dept. of Mechanical Engg., National Institute of Technology-Rourkela, Odisha, India

Kinematics of a robot arm deals with the geometry of motion of robot arm with respect to a fixed reference coordinate system without regard to the effect of forces. The above problem can be defined in two fundamental ways, namely forward or direct kinematics and the second problem is inverse kinematics. Since the independent variables in a robotic arm are the joint variables and a task is usually in terms of reference coordinate frame, inverse kinematics is used more frequently. Direct kinematics approach is generalized to describe and represent the location of the links of a robot arm with respect to the base coordinate system. Denavit and Hartenberg (D-H) introduced the matrix representation of a link in order to describe the spatial geometry of a robotic manipulator. Because of this representation, one can easily understand the kinematic equation for a robotic arm. The present research work aims to implement D-H parameters to a modeled 4-axis articulated robotic arm. Moreover, this paper describes the development of inverse kinematic models of the 4-axis robotic manipulator based on its arm equation. Finally, the results obtained from the analytical solutions are compared with the experimental results for a 4-axis articulated manipulator.

Keywords: *forward kinematic model, inverse kinematic model, D-H Parameters, 4-axis robotic manipulator, arm equation*

1. INTRODUCTION

An articulated object is a set of rigid segments connected with joints; they are usually either rotational or translational. A minimal kinematic model is defined by its individual rigid segment lengths, joint degrees of freedom, their maximum and minimum joint limits, and a tree structured hierarchy of the segments and joints. Each joint maintains the rotations currently in effect at the corresponding joint. Kinematic analysis can be done in two ways namely forward and inverse manner.

In forward kinematics, for a given joint parameters (link lengths and joint angles), end-effector's position can be known. To solve this kind of problem various algorithms have been implemented for various robotic manipulators. To solve the forward kinematic problem of a 3RPR parallel manipulator, combination of genetic algorithms and simulated annealing implemented [1]. Closed-form solutions for the forward displacement and velocity kinematics of two parallel Hunt-Primrose parallel manipulators are presented in [2]. For a bio-inspired parallel manipulator [3] with one translation along z-axis and two rotations along x- and y- axes has developed as the hybrid head mechanism of a groundhog robotic system. Accordingly, the forward kinematics addressed based on the integration of radial basis function network and inverse kinematics.

In inverse kinematics, for a given link lengths, end-effector's position and orientation, joint angles can be known. To solve this kind of problem various algorithms have been implemented in past. SZKODNY [7] presented the equations of links and actuator kinematics of the IRb-6 manipulator in matrix form. A new algorithm [4] to compute the inverse kinematics of a general 6R serial kinematic chain is presented on the basis of classical multidimensional geometry to structure the problem. Algebraically this procedure means to solve a system of seven linear equations and one resultant to arrive at the univariate 16 degree polynomial. Henten et al [5], presented an inverse kinematics algorithm that has been used in a functional model of a cucumber-harvesting robot consisting of a redundant P6R manipulator.

An adaptive learning strategy [7] using an artificial neural network has been proposed to control the motion of a 6 D.O.F manipulator robot and to overcome the singularities and uncertainties in arm configurations. A new geometrical approach [8] has introduced to solve the inverse kinematics for continuous backbone robot manipulators.

The purpose of the current investigation is to understand the development of arm equation using forward kinematics based on Danvit-Hartenberg parameters. From the developed arm equation, inverse kinematic models

*Corresponding author: *bbv@nitrkl.ac.in*

International Journal of Applied Artificial Intelligence in Engineering System 3(2), 2011: 103-108

IJAAIES

SUGENO FUZZY BASED NAVIGATIONAL CONTROLLER OF AN INTELLIGENT MOBILE ROBOT

Dayal R Parhi and BBVL Deepak

Abstract: This paper deals with the development of fuzzy inference system in order to solve mobile robot navigation problem. The developed system architecture works on the basis of Sugeno fuzzy type. To achieve the better path by an intelligent mobile robot within its work space, the proposed fuzzy model requires two inputs: (1) the distance from the robot to the obstacles in the workspace and (2) position of the target i.e. the robot heading angle towards the destination. Once the system gets the knowledge from the environment, it will obtain the suitable steering angle for an autonomous mobile robot. This process will be continued until the robot reaches its goal. Simulation results are given to verify the feasibility of the proposed methodology for an autonomous mobile robot.

Key words: fuzzy inference system, obstacle avoidance, autonomous mobile robot, mobile robot navigation

1. INTRODUCTION

The basic requirement of an autonomous mobile robot is the ability of generating path by avoiding obstacles within its environment. Behaviour-based methodologies have been introduced in order to solve this problem in the recent years as explained in [1]. A large amount of work has been dedicated to solve this problem and reviews on many approaches have been conducted by various authors as described in [2-4].

Fuzzy Logic Algorithm (FLA) have implanted in [5] for a mobile robot controller hardware to avoid obstacles by turning proper steering angle. Behaviour-based fuzzy control has been applied in [6] to sonar based obstacle avoidance for Help-mate mobile robot. Yousfi et al. [7] have developed sugeno fuzzy controller for a mobile robot and the consequences of the fuzzy system have been optimized by using gradient method. Mobile robot navigation in indoor environments has been carried out in [8] using fuzzy logic controller. In their work inputs are the desired direction of motion and the sensors readings; and the output is the accelerations of the robot wheels. Norouzi et al. [9] have introduced a navigation approach using fuzzy controller to produce paths based on extracted lines and fusing data from sensory information. A fuzzy logic-based real-time navigation controller [10] has been addressed which combines path planning and trajectory for an autonomous mobile robot. A rough-fuzzy controller for an autonomous mobile robot based on rough set and fuzzy set theory has been developed in [11]. By this approach they improved uncertainty reasoning of the mobile robot in order to obtain better wall-following behaviour results. A two-stage fuzzy inference system [12] has been introduced for sensor based mobile robot in order to navigate the robot from one position to another. The first stage of the fuzzy system gives the outputs as angular velocity based on the robot's five sonar sensor readings then these values are fed into stage of the fuzzy controller to get the required linear velocity as the output.

The present research work aims to develop efficient system architecture using fuzzy inference system. This paper arranged into six sections as follows: second section outlines the Sugeno fuzzy model; third section describes the selection of member ship functions and the required rules for mobile robot navigation are illustrated in section four. The results obtained from the developed methodology are discussed in section five and in the last section the concluding results are explained.

* Robotics Laboratory, Department of Mechanical Engineering, National Institute of Technology, Rourkela, Orissa, India, E-mail: dayaldoc@yahoo.com; deepakjournal@gmail.com

Int. J. on Recent Trends in Engineering & Technology, Vol. 05, No. 04, Mar 2011

Kinematic Model of Wheeled Mobile Robots

B B V L Deepak[1], Dayal R Parhi[1] and Alok Kumar Jha[1]
Department Of Mechanical Engineering, National Institute of Technology- Rourkela, Rourkela, India
E-mail: deepak.bbvl@gmail.com ; dayalparhi@yahoo.com

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

***Index terms*- Wheeled Mobile Robot, Kinematic Analysis, Degree of Maneuverability**

I. INTRODUCTION

A mobile robot is a self-operating machine that can move with respect to its free space. By understanding the wheel constraints placed on the robots mobility it is easier to understand the mobile robot motion. It is difficult to measure a mobile robot's position instantaneously by direct way. It is an extremely challenging task to express the exact mobile robot's position while it is in motion.

A wheeled mobile robot has been modeled as a planar rigid body that rides on an arbitrary number of wheels in order to develop a relationship between the rigid body motion [1] of the robot and the steering and drive rates of wheels. In prevous research work kinematic models [2],[3][5][6] are designed for holonomic mobile platform[2] in order to provide omni-directional motions [3]by three individually driven and steered wheels. Kinematic modeling, singularity analysis, and motion control has been outlined in [3] for a mobile robot equipped with N Swedish wheels. With the help of an interactive tool [4] it is possible to solve mobile robot motion problems by understanding several well-known algorithms and techniques. In [6] it has been modeled that the wheels as a torus and proposed the use of a passive joint allowing a lateral degree of freedom for kinematic analysis of a wheeled mobile robot (WMR) moving on uneven terrain.

Most of the previous research work dealt with the kinematic models for few wheel types [3],[7]. The present research work deals with a general wheeled mobile robot (WMR), with arbitrary number of various configured wheels. Our aim is to point out the kinematic models for various WMRs. It has been shown that the variety of possible robot constructions and wheel configurations by the concepts of degree of mobility and of degree of steeribility. The set of WMR can be partitioned in to five configurations and this analysis is carried out in later sections.

II. KINEMATICS OF VARIOUS WMR

A WMR is equipped with motors which is capable of an autonomous motion (without external human driver). These are driven by an embarked computer for its motion.

The total dimensionality of a robot chassis on the plane is three, two for position in the plane and one for orientation along the vertical axis, which is orthogonal to the plane as shown in Figure1.

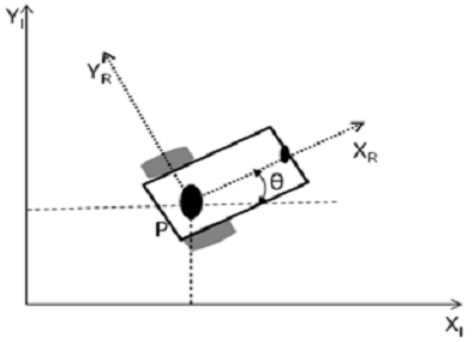


Figure1. The global and local reference planes.

The position of *P* in the global reference frame is specified by coordinates *x* and *y*, and the angular difference between the global and local reference frames is given by θ. Therefore the robot position: $\xi_I = [x \quad y \quad \theta]^T$ (1)

And mapping is accomplished using the orthogonal rotation matrix:

$$R(\theta) = \begin{bmatrix} cos\theta & sin\theta & 0 \\ -sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

From equation (2) we can compute the robot's motion in the global reference frame with respect to its local reference frame:

$$\dot{\xi}_I = R(\theta)^{-1}\dot{\xi}_R \quad (3)$$

A. Wheel Kinematic Constraints:

We can observe two constraints while a WMR is in motion. The first constraint is due to rolling contact as represented in Figure2.The second constraint is due to no lateral slippage – that the wheel may slide orthogonal to the wheel plane as shown in Figure3.

DOI: 01.IJRTET.05.04.128

Journal of Mechanical Engineering Research Vol. 3(9), pp. 307-318, 21 September, 2011
Available online at http://www.academicjournals.org/jmer
ISSN 2141 - 2383 ©2011 Academic Journals

Full Length Research Paper

Kinematic model of three wheeled mobile robot

Dayal R. Parhi* and B. B. V. L. Deepak

Robotics Laboratory, Department of Mechanical Engineering, National Institute of Technology-Rourkela, Rourkela, Orissa, India.

Accepted 11 July, 2011

In this paper, the behavior of wheeled mobile robots (WMR) has been analyzed. These robots ride on a system of wheels and axles, some of which may be steerable or driven. For WMRs, there are many wheels and axle configuration that have been used. The present work outlined for five categories of WMRs. The ultimate objective of this paper is to investigate the complete description of the control theory of such robots and its maneuverability. Equations are modeled to describe the rigid body motions that arise from rolling trajectories based on the geometrical constraints of these wheels. Finally this analysis is applied to various three wheeled mobile robots.

Key words: Wheeled mobile robot, kinematic analysis, degree of maneuverability.

INTRODUCTION

A mobile robot is a combination of various hardware and software components in order to move in its free space. It is a collection of subsystems as shown in Figure 1; *Locomotion*: it enables how the robot to move unbounded throughout its environment. *Sensing*: How the robot measures properties of itself and its environment. *Control*: How the robot generate physical actions. *Reasoning*: How the robot maps measurements into actions; *Communication*: How the robots communicate with each other or with an outside operator. But there are a large variety of possible ways (Xiaodong and Shugen, 2010) to move, and so the selection of a robot's approach to locomotion is an important aspect of mobile robot design. Understanding mobile robot motion starts with understanding constraints placed on the robots mobility. Owing these limitations, mobile robots generally locomote either using wheeled mechanisms, a well-known human technology for vehicles, or using a small number of articulated legs (Júlia and Federico, 2009), the simplest of the biological approaches to locomotion. In general, legged locomotion requires higher degrees of freedom and therefore greater mechanical complexity than wheeled locomotion. Wheels, in addition to being simple, are extremely well suited to flat ground.

Alexandery and Maddocks (1988) modeled a wheeled mobile robot as a planar rigid body that rides on an arbitrary number of wheels. And they developed a relationship between the rigid body motion of the robot and the steering and drive rates of wheels. The structure of the kinematic and dynamic models has been analyzed by Guy et al. (1992), for various wheeled mobile robots. Those wheel types, ome of the researchers have been dealt with some types of wheels. Wheel architecture has been developed by Kim et al. (2003), for the holonomic mobile platform in order to provide omni-directional motions by three individually driven and steered wheels. Jae et al. (2007) investigated the kinematics of a mobile robot with the proposed double-wheel-type active caster, which is developed as a distributed actuation module and can endow objects with mobility on the planar workspace. For a mobile robot equipped with N Swedish wheels, Giovanni (2009), has been described its kinematic modeling, singularity analysis, and motion control. Vrunda et al. (2010) have analysed on spherical wheeled mobile robot for feasible path planning and feedback control algorithms. Nilanjan and Ashitava (2004) modeled the wheels as a torus and proposed the use of a passive joint allowing a lateral degree of freedom for kinematic analysis of a wheeled mobile robot (WMR) moving on uneven terrain. A University of Minnesota's Scout is a small cylindrical robot has been modeled by Sascha et al.

*Corresponding author. E-mail: dayalparhi@yahoo.com.