Navigation of Automatic Vehicle using AI Techniques

A thesis submitted towards partial fulfilment of the requirements for the degree of

Master of Technology

in

Machine Design and Analysis
(Mechanical Engineering Department)

by

Rakesh Kumar Sonkar
(211ME1164)

National Institute of Technology,
Rourkela-769008 Odisha, India
2013

Navigation of Automatic Vehicle using AI Techniques

A thesis submitted towards partial fulfilment of the requirements for the degree of

Master of Technology

in

Machine Design and Analysis
(Mechanical Engineering Department)

by

Rakesh Kumar Sonkar
(211ME1164)

Under the supervision of

Prof. Dayal R. Parhi
(Mechanical Engineering Department)

National Institute of Technology,
Rourkela-769008 Odisha, India
2013

National Institute of Technology, Rourkela

CERTIFICATE

This is to certify that the project entitled, **Navigation of Automatic Vehicle using AI Technique** submitted by **Rakesh Kumar Sonkar** is an authentic work carried out by him under my supervision and guidance for the partial fulfilment of the requirements for the award of **Master of Technology Degree** in **Machine Design And Analysis (Mechanical Engineering Department)** at **National Institute of Technology, Rourkela**.

To the best of my knowledge, the matter embodied in the project has not been submitted to any other University or Institute for the award of any Degree or Diploma.

Date **(Prof. Dayal R. Parhi)**

Rourkela **Mechanical Engineering Department**

ACKNOWLEDGEMENT

I express my profound gratitude and indebtedness to **Prof. Dayal R. Parhi**, Department of Mechanical Engineering, NIT, Rourkela for introducing the present topic and for his inspiring intellectual guidance, constructive criticism and valuable suggestion throughout the project work.

I express my sincere thanks to **Prof. K.P. Maithy**, HOD, Department of Mechanical Engineering, and also the other staff members of Department of Mechanical Engineering, NIT Rourkela for providing me the necessary facilities that is required to conduct the experiment and complete my thesis.

I am also thankful to **Mr. Alok Kumar Jha, Mr. Anish Pandey, Mr. Krishna Kant Pandey and Mr. Prases Mohanty,** and other member of Robotics Lab for their valuable time towards my entire project work.

I am deeply indebted to my parents who always believed in me and gave all their support for all the choices that I have made. Finally, I humbly bow my head with utmost gratitude before the God Almighty who always showed me a path to go and without whom I could not have done any of these.

Date

Rourkela **Rakesh Kumar Sonkar**

ABSTRACT

In the field of mobile robot navigation have been studied as important task for the new generation of mobile robot i.e. Corobot. For this mobile robot navigation has been viewed for unknown environment. We consider the 4-wheeled vehicle (Corobot) for Path Planning, an autonomous robot and an obstacle and collision avoidance to be used in sensor based robot. We propose that the predefined distance from the robot to target and make the robot follow the target at this distance and improve the trajectory tracking characteristics. The robot will then navigate among these obstacles without hitting them and reach the specified goal point. For these goal achieving we use different techniques radial basis function and back-propagation algorithm under the study of neural network. In this Corobot a robotic arm are assembled and the kinematic analyses of Corobot arm and help of Phidget Control Panel a wheeled to be moved in both forward and reverse direction by 2-motor controller have to be done. Under kinematic analysis propose the relationships between the positions and orientation of the links of a manipulator. In these studies an artificial techniques and their control strategy are shown with potential applications in the fields of industry, security, defense, investigation, and others. Here finally, the simulation result using the webot neural network has been done and this result is compared with experimental data for different training pattern.

CONTENTS

List of Figures

List of Table

CHAPTER 1
INTRODUCTION

1. Introduction

1.1 Background

Mobile robots have high potential in several applications. These include automatic driving with guidance for the blind and disabled, explorations of dangerous or unknown regions. Current research and development of mobile robot have attracted the attention of researchers in the areas of engineering, biology, mining, industrial and others. Autonomous mobile robots are intelligent agents which can perform desired tasks in various (known and unknown) environments without continuous human control. Many kinds of robots are autonomous to some step. One important area of robotics research is to enable the robot to cope with its environment whether this is on land, water, in the air or in space. Autonomous mobile robotics is a challenging research topic for several reasons. First, a mobile robot should able to identify features, detect obstacles, patterns and target, learn from experience, find a path and build maps, and navigate. These abilities of mobile robot require the simultaneous application of many research disciplines such as Engineering and their application.

Secondly, autonomous mobile robots are the closest approximation of intelligent agents. To satisfy this goal mobile robotics research has increasingly incorporated artificial intelligence enabling the machines to mimic living beings. Path analysis and planning is another exciting challenge in building autonomous mobile robots. An autonomous robot must be able to learn its environment and programming itself without assistance. It consists on finding a route from the origin of the robot to its target destination. Path analysis and planning becomes more difficult when some static as well as dynamic obstacles are added to the environment. Thirdly, and we use an autonomous mobile robot i.e. Corobot for navigation and use the AI techniques to evaluate the path of robot. Also we show the webot software simulation and Radial basis function for this Corobot.

1.2 Aim and Objectives

The overall aim of this research is to explore the application of artificial intelligence techniques to navigate mobile robots. In this thesis, neural network, field navigation and simulated techniques have been used to solve mobile robots navigation problems. The development of techniques for autonomous mobile robot operation constitutes one of the major trends in the current research and practice in modern robotics. The goal of autonomous mobile robotics is to build and control physical systems which can move purposefully and without human intervention in real-world environments which have not been specifically engineered for the robot. This includes the kinematics, perception, cognition, sensor fusion, path analysis, path planning and navigation. The following behaviors are required during navigation of mobile robots.

1. Obstacle avoidance behavior, so that the mobile robot able to avoid collisions with both static obstacles and dynamic obstacles in various environments.
2. Wall following behavior, so that mobile robot cannot trap in loop as the mobile robot detects an obstacle in the front while the target tracking control mode is on operation.
3. Target searching behavior, so that mobile robot quickly moves towards the target if there are no obstacles around the robot.

The design of autonomous mobile robots capable of intelligent motion and action involves the integration of many different bodies of knowledge. A 4 WMR **uses AI** for guidance, obstacle avoidance, kinematic analysis, simulation using the Webot and define the neural network for navigation of mobile robot has to be main objectives. For a **mobile** robot to follow a virtual target vehicle that is moved exactly along the path with specified velocity. Also a vehicle system, aimed at communicating with the on board PC, supporting different sensors, camera etc.

1.3 Outline of the Work

The outlined in this thesis are broadly divided into eight chapters. First the introduction, Chapter 1, after background and objective of research describes the robot overview and a short introduction about Navigation, Autonomous Mobile Robot and Artificial Intelligence techniques.

Chapter 2 provides a state of the review of navigation, mobile robot control, path planning and obstacle avoidance, AI techniques and neural network of mobile robot.

In Chapter 3 analyses the kinematics analysis of mobile robots. A wheeled mobile robot is considered for the kinematic analysis of robotic arm.

In Chapter 4 defines the concept of the neural network, Webot and Corobot. In these methodology of NN used to RBFN and back propagation algorithm to which enables the mobile robot to navigate successfully in real world environment. At the part of webot study we analyses the simulation of Corobot using webot. And finally the last part of these chapter we introduce the different view of Corobot, their initial setup, specification and installation.

In Chapter 5 a detailed report of results and discussion has been given. This chapter summarizes the findings of all chapters discussed before.

Finally in Chapter 6 conclusions of this research and future ways for further investigation has been discussed. The paper published lists are listed at the end of the thesis.

CHAPTER 2
LITERATURE REVIEW

2. Literature Review

2.1 Introduction

In a new proposal to solve the problem of path planning and obstacle avoidance for mobile robots and the study in the field of navigation of mobile robot gained an extensive interest among the researchers and scientists since last few decades. Nowadays, robotics is an important part in manufacturing practices. About mobile robots, autonomous navigation involves a great task. A mobile robot (MR) can be very beneficial in different circumstances where humans could be in risk or when they are not able to reach positive goals because of terrain environments. Like Examples of everyday tasks of driving in city traffic, parking a car, and house cleaning. In accomplishment such familiar tasks, humans use observations of time, distance, speed, shape, and other aspects of physical and mental things. For any autonomous robot obstacle avoidance is the primary requirement. Many sensors and actuators are requires for integration and coordination for designing a robot.

A mobile robot is an automatic machine that is accomplished of drive in a given environment. Mobile robots have the ability to move around in their environment and are not fixed to one physical situation. In compare, industrial robots usually consist of a jointed arm and gripper assembly that is attached to a fixed surface. Navigation is a field of study that efforts on the process of monitoring and controlling the movement of a vehicle from one place to another. This navigation field includes different general categories i.e. marine navigation, land navigation, aeronautic navigation, and space navigation. It is also the term to use for the specialized knowledge used by navigators to carry out navigation tasks. To locating the navigator position compared to known locations involves different navigational techniques.

In physics and mathematics is a system in Non-holonomic environment whose state depends on the path taken to achieve it. Such a system is refer to by a set of parameters subject to degree of difference constraints, such that when the system go forward along a path in its parameter space (the factors varying continuously in values) but finally returns to the original fixed of values at the start of the path, the system itself may not have back to its original state.

Fuzzy logic is a form of many-valued logic or probabilistic logic; it deals with reasoning that is fairly accurate rather than fixed and exact. Fuzzy logic has been extended to handle the concept of truth, where the truth value may range between completely true and entirely false. Still, when linguistic variables are used, these degrees may be able to specific functions. Robotics is the branch of technology that deals with the design, fabrication, operation and application of robots and computer systems for their control, sensory response, and information giving out. These technologies compact with automated machines that can take the place of humans, in harmful or manufacturing processes, or simply just are like humans. The mechanical structure of a robot must be controlled to perform jobs. It involves three distinct phases for control of a robot – perception, processing, and action. Sensors give facts about the environment or the robot itself (e.g. the position of its joints or its end effector).

During the past few years, wheel-based mobile robots have attracted considerable attention in various industrial and service applications. For example, room cleaning, factory automation, transportation, etc. These applications require mobile robots to have the ability to track specified path stably. In general, Non-holonomic behaviour in robotic systems is particularly interesting because most mobile robots are Non-holonomic wheeled mechanical systems. Control problems of mobile robot caused by the motion of wheels that has three degrees of freedom, while control of the mobile robot is done using only two control signals under Non-holonomic kinematics constraints.

Matveeva & Teimoori [1] consider the difficulty of navigation and guidance of a wheeled mobile robot towards a target based on the measurements concerning only the distance from the robot to the target. We propose a controller that drives the robot to the predefined distance from the target and makes the robot follow the target at that distance. Mohareri et al. [2] presented the design and operation of an adaptive path tracking controller for a wheeled mobile robot (WMR) with unknown parameters and uncertain dynamics. The learning ability of neural networks is used to design a robust adaptive back stepping controller that does not require the knowledge of the robot dynamics. Systems are found in different applications ranging from unicycles and car-like vehicles, possibly equipped with trailers, to systems like rolling spheres, snake-like robots, snake boards, roller racers, and wheel-chairs. A wheeled mobile robot (WMR) is one of the well-known systems with nonholonomic constraints, and has addressed its tracking control problem. Kanda et al. [3] shows the various mechanisms have recently been developed that combine linkage tools and wheels. In exact, the combination of passive linkage mechanisms and small wheels is a main research trend because standard wheeled mobile mechanisms finds it difficult to move on rough terrain. We propose an environment recognition system for a wheeled mobile robot that consists of multiple organization analyses to make the robot more adaptive to various environments by selecting a suitable system such as decision making, navigation and controller using the effect of the environment recognition system. In environment recognition system, image data, laser scanner data, GPS and Inertial Navigation System (INS) are often used for self-localization and mapping.

Sekmen et al. [5] dealt with the advances in technologies to create ever more sophisticated robots is outpacing our understanding of how such robots and humans successfully interact to complete specific responsibilities. In specific, it is now possible for humans to control the navigation of certain classes of robots via the Internet. The World Wide Web (WWW) provides an inexpensive and widely accessible means for teleoperation. Both computer and Internet technology are improving with amazing speed and some robotics researches have begun exploring tele-presence applications. In fact, the only information the human operator may have about the ‘‘terrain’’ is the information available through the robot control interface. Sebastian Thrun [6] defined autonomous robots must be able to learn and maintain models of their environments. To efficiently carry out complex missions in indoor environments, autonomous mobile robots must be capable to secure and maintain of their environments. Deepak et al. [7, 8] have presented mobile robots which are widely used in various fields such as domestic fields, industries, security environments etc. because of their movement 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 interference. The main objective of a navigational controller of an autonomous mobile robot is to generate collision free trajectories within its workspace. For mobile, autonomous robots the capability to purpose in, and interact with, a dynamic, changing environment is of key importance. Ming et al. [9] have used incorporation of an integration procedure is becoming an increasing necessity for autonomous robotic vehicles capable of moving along in the industrial environment. This is due to a change in the kind of application required from robotics. Usually these new applications have the same basic features: a mobile robot that moves in a partially unknown industrial environment, trying to reach a target, and an articulated arm joined to the vehicle which is devoted to carrying out the required tasks.

Dautenhahn Kerstin [10] defined autonomous robots are integrated into human people, interacting and cooperative both with humans and with each other. This goal is to suggest that these ideas should also find their way into the sciences of the artificial. Andrzej & Skrzypczynski [11] had studied the autonomy of mobile robots directly depends upon the availability of the adequate model of the environment, which can be used to back up the robot tasks at hand. In the case of robots operating in industrial environments the map could be provided in advance.

Ulrich et al. [12, 13] have defined the following three properties are foundations of robust robot navigation: (i) the use of landmarks (ii) the use of undisputed paths, and (iii) the use of topological rather than geometrical maps. Navigation is a key for any mobile robot in its most predictable form; the navigation problem can be stated as follows: given that a robot is at a unknown location in a unknown environment, how does it go about success a goal location. Danny & Phillip [14] shows navigation an outdoor robot, one of the problems of navigating robots is that many electronic sensing systems do not produce this rich sensor data. A second problem is that the sensors that yield rich information are not yet able to reliably perceive objects. This limitation of robot sensing and perception places severe constraints on the ability of a mobile robot to navigate.

Fua et al. [15] shows for miniaturized mobile robots that aim at travelling unknown environments, contact 3D sensing of basic geometrical features of the surrounding environment is one of the most important capabilities for survival and the mission. Range sensors are usually used for non-contact exploration of the surrounding environment and for the purpose of robot dynamic navigation, and they include passive sensors and active sensors. Compared with passive sensors, active sensors, which include infrared sensors, ultrasonic sensors, and laser sensors, offer more reliable range sensing. Infrared sensors are simple, small and cheap; however, they have rather shorter sensing range and less range resolution compared to other optical sensors. In addition, because an infrared sensor only measures a range distance from the sensor coordinate frame to the target point at a time, so it needs to be scanned two dimensionally for 3D sensing. Thrun Sebastian [18] states that, a mobile robot, equipped with optical, ultrasonic and laser sensors, learns to servo to a designated target object. In less time operation, the robot is able to navigate to a marked target object in an unknown environment. Mohd Nazri & Saad [4] states Person detection and tracking systems are important capabilities for applications as a service robot in different environment. This work presents a simple method that able to visually detect and track specific person using a single camera based on hybridization method of image information. This method is applied to estimate the position and orientation of a moving target person in crowded environment. The range between the target person and the mobile robot can be computed in real-time using a set of markers so that the robot can control its speed and direction to follow the target person as closely as possible.

Sebastian Thrun [6] deals Sensors are not capable of directly measuring the mass of interest. Example, like cameras measure colour, brightness and capacity of light, whereas for navigation one might be interested in statements such as “there is a door in front of the robot." One of the robots (AMELIA) is similarly fitted out with a laser range finder, which measures closeness of nearby objects with higher 3-D resolution. Wenfeng & Weiming [16] states wireless sensor network (WSN) nodes can closely sense their surroundings in a convenient and distributed way so that they can be considered as nerve terminals connected to a network such as the Internet. Recently, due to their great application potential, a trend has emerged that combines wireless sensor networks (WSN) and multi-mobile robots (MMR). Interesting applications can be found in disaster emergency response, military, communication, transport, and plant automation.

Nippun & Sudarshan [17] states that, video capturing is one more approach is used to teach the robot. In robot will follow a human demonstrator and simultaneously gathers information of the environment. Initially, a robot is controlled by a human operator who manually guides the robot through a desired path. Weckesser & Dillmann [19] defined a Robots are supposed to operate in dynamic and changing environments together with human beings and other static or dynamics objects. Sensors that are capable of providing the quality of information that is required for the described scenario are optical sensors like digital cameras and laser scanners. A multi-sensor system supports the vehicle with odometric, sonar, and visual and laser scanner information. The goal of this work is making robot navigation safer, faster, more reliable and more stable under changing environmental.

Danny & McKerrow [14] show Titan is a mobile robot built for outdoor navigation research. Titan measures its location relative to the path edge with a continuous transmission frequency modulated (CTFM) ultrasonic sensor and steers to follow a trajectory relative to the edge. Miikkulainen et al. [20] dealt about path-planning, it decides continuous from discretized places and describes procedures applicable when the implementation of a plan fails. It maintains for an integrated conception of such procedures, which must be tightly designer to the specific robot that is used, notably to the abilities and limitations of its sensory-motor tools. Path-planning, which is the process of selecting a course of actions to reach a aim, given the current location? Garcia et al. [21] defined in the Motion Planning investigation field and a method has demonstrated to outperform classical approaches gaining popularity in the last 35 years. This presents a proposal to solve the problem of path planning for mobile robots based on Simple Ant Colony Optimization Meta-Heuristic (SACO-MH). The new method is called SACOdm, where d stands for distance and m for memory. The path planner application has two operating modes, one is for effective environments, and the second one works with an actual mobile robot using wireless communication. Both operating modes are overall planners for plain terrain and support static and dynamic obstacle avoidance.

Marsland et al. [22] shows in landmark-based navigation systems for mobile robots, sensory perceptions (e.g., laser or sonar scans) are used to identify the robot's current location or to construct internal demonstrations, plans, of the robot's environment. Presence based on an outdoor structure of reference landmark-based robot navigation systems are now widely used in mobile robot applications. The problem that has concerned most attention to date in landmark-based navigation research is the question of how to deal with perceptual aliasing, i.e., perceptual uncertainties. In difference, what constitutes a good landmark, or how to select landmarks? The usual method of landmark selection is to map observation sat regular intervals, which has the problem of being inefficient and possibly disappeared 'good' landmarks that lie between sampling points. Fua et al. [15] defined based on different working principles laser sensors can be categorized into time-of-flight (TOF) and triangulation. The TOF laser scanners have the advantages of a wide measuring range and high relative accuracy at a long distance; however, they are expensive, high power consumption and heavy. Commercial TOF laser scanners such as the HOKUYO URG-04 LX and the Swiss Ranger SR4000 are still too large to be used on centimeter-scale miniature mobile robots.

Boubaker & Tarek [23] states the advantage of redundant serial robot manipulators in real applications is their dexterity and ability to avoid obstacles. This is due to the fact that they have more degrees of freedom than required to achieve desired tasks by their end-effector. However, redundancy generates complexity of the dynamic controller synthesis which can be considered as a challenge in automatic control, mainly if the robot is in the presence of mechanical constraints and mobile obstacles. A new controller approach applied to fire robot manipulators forced by mobile obstacles. The controller is constructed in task space by using optimization strategy, in order to achieve a good trajectory tracing of the end effector even if the obstacles are fixed or mobile. Abiyev et al. [24] states obstacle avoidance is the primary requirement for any autonomous robot. Designing a robot requires the integration and coordination of many sensors and actuators. In general, the robot acquires information about its surrounding through various sensors fixed on the robot. Generally, multiple sensors, such as infrared sensor, ultrasonic sensor, laser range finder, touch sensor and camera can be used to detect the presence of obstacles.

Guoqiang & Arianna [15] dealt in addition, because an infrared sensor only measures a range distance from the sensor coordinate frame to the target point at a time, so it needs to be scanned two dimensionally for 3D sensing. Ultrasonic sensors have the advantages of simple implementation and fast obstacle detection, but they are not accurate and reliable when detected obstacles have a complicated3D shape, thus they are normally used for object detection and avoidance. Ming et al. [9] deals the control system designed allow a mobile semiautonomous robot to avoid unexpected obstacles in a partially unknown environment. In order for the robot to perform these tasks correctly, it must be able with some sensory system. There exist many areas of industrial application that can benefit from automated obstacle avoidance technology, for example, mobile robots that can roam freely and safely in a factory environment. Jiann-Der Lee [25] shows an intelligent approach to robot navigation by landmark tracking using computer visualization is proposed. This approach is based on the concept that a human can reach the destination by tracking the specific landmark in an earlier environment. Only a monocular image of the landmark taken by the robot is required.

Noorani [26] describes the equipment and procedures used to modify a toy robot arm such that it can be interfaced with a computer. The motion of the arm is transmitted by plastic gears which are attached to stepper motors. These motor are interfaced with the computer where the movement of the robot arm is controlled by using the computer keyboard to input the angle and direction of motion. Paik et al. [27] proposes the construction of a humanoid-applicable anthropomorphic 7-DOF arm complete with an 8-DOF hand. Here, a humanoid robot that resembles a human in appearance and movement is built using powerful actuators paired with gear trains, joint mechanisms, and motor drivers that are all encased in a package no larger than that of the human physique. Kadir et al. [28] talk about robot to bridge the gap of the normal perception of “robots are for the industries only”, internet will be use. They present the development of an internet controlled robotic arm. Yusoff et al. [29] describe with the increases usage of wireless application, the demand for a system that could easily connect devices for transfer of data over a long distance – without cables, grew stronger. It can be moved, reverse, turn right and left for a specific distance according to the controller specification.

Yamaguchi [30] shows the research effort of Non-holonomy mobile robot, is a central issue on controlling a mobile robot in a single operation or multiple mobile robots in a collective operation. There have been significant methodologies for asymptotically/exponentially stabilizing Non-holonomic mechanical systems, including unicycle-type mobile robots, car-like mobile robots, and mobile robots towing trailers, in single operations by smooth or non-smooth time-varying feedback control laws. Defoort et al. [31] worked in multi-robot system relies on the fact that multiple robots have the possibility to solve problems more efficiently than a single one. Cooperative control gives rise to significant theoretical challenges and has various engineering applications including manufacturing, observation and space exploration.

Daniel and Wolfram [32], state non-parametric geometric representations of the environment play an important role in mobile robotics since they support various fundamental tasks such as path planning or localization. One of the most popular approaches is occupancy grids, which provide a discrete probabilistic representation of the environment. Other frequently used approaches are representations based on geometric primitives that are typically found in the environment. In this context, lines play a major role, since many man-made buildings are composed of linear structures like walls, for example. Although these techniques have been successfully applied in the past, they have certain disadvantages coming from discretization errors or because of missing features in the environment.

Ahmad et al. [33], presents a trajectory tracking controller for a Non-holonomic mobile robot using an optimization algorithm based predictive feedback control and an adaptive posture identifier model while following a continuous and a non-continuous path. The posture identifier model is a modified Elman neural network that describes the kinematics and dynamics of the mobile robot model. The feed forward neural controller is trained off-line and its adaptive weights are adapted on-line to find the orientation torques, which controls the stable-state outputs of the mobile robot system. Tsai & Song [34], presents a robust visual tracking control design for a Non-holonomic mobile robot fitted out with a tilt camera. The aim to this design allows the mobile robot to keep track of a dynamic moving target in the camera's field-of-view; even though the target is temporarily fully stop up. To achieve this, a control system consisting of a visual tracking controller (VTC) and a visual state estimator (VSE) is offered. A novel visual interface model is derived to assist the design of VTC and VSE. The VSE is in control for estimating the optimal target state and target image velocity in the image space.

Luigi & Giuseppe [35], deals a novel vision-based scheme is presented for driving a Non-holonomic mobile robot to cut off a moving object. A two-level structure method is proposed. On the lower level, the pan–tilt platform carrying the on-board camera is controlled so as to keep the target as close as possible to the centre of the image plane. And on the higher level, the relative position of the target is recovered from its image coordinates and the camera pan–tilt angles over and done with simple geometry, and used to work out a control law which drives the robot to the target. With the problem of intercepting a moving target via a Non-holonomic mobile robot through visual feedback interception (approaching a moving object until collision) and tracking (approaching a moving object while matching its location and velocity) are important tasks in a number of applications, going from robotic games to automated surveillance. In addition, the development of effective methods for performing these tasks represents a challenging tested for the integration of various techniques involving image processing, sifting, control theory and artificial intelligence (AI) strategies.

Jun [36], proposes an improved learning algorithm of compound orthogonal networks and a novel tracking control approach for Non-holonomic mobile robots by integrating the neural network into the back stepping technique. The adaptive control is derived from continuously tuning parameters using the neural network in the back-stepping control law. Markus [37], proposed a tracking controller for Non-holonomic dynamic system which allows global tracking of arbitrary reference trajectories and renders the closed loop system robust with respect to bounded disturbances. The controller is based on Sliding-mode tracking control of Non-holonomic wheeled mobile robots in polar coordinates. The control law for tracking of general Non-holonomic systems using inverse kinematic models (IKM) and sliding surfaces is stated.

Pourboghrat & Karlsson [38], presents adaptive control rules, for the Non-holonomic mobile robots with unknown parameters. For mobile robots, derived Adaptive controls for tracking of a reference path and stabilization, using back stepping techniques. For the following problem, the controller guarantees the asymptotic coming together of the tracking error to zero. For maintenance, the problem is converted to a same tracking problem, using a time varying feedback error, in the past the tracking control is applied. The designed controller makes certain the asymptotic zeroing of the stabilization error. The offered control laws include a velocity/acceleration limiter that avoids the robots wheels from slipping.

Farzad [39], death the problem of point-to-point control design for differentially steered Non-holonomic mobile robots is considered. The control variables are derived using Lyapunovs stability technique. The proposed control law guarantees the exponential stability of the closed-loop system and ensures the convergence of the position and the orientation of the robot to their desired fixed values. Toibero et al. [40], presents a stable switching control strategy for the parking problem of Non-holonomic mobile robots. For parking problem first, it is proposed a positioning-orientation switching controller. With this strategy robot backwards motions are avoided and the robot heading is always in the direction of the goal point facilitating the obstacle behavior. Second, the avoidance of sudden obstacles is considered in a reactive way by following the shape of the obstacles. Next, Showing stability under reasonable conditions, the stability of the switching parking/ obstacle-avoider controller is analyzed.

Mohareri et al. [41], presents Non-holonomic mobile robot (WMR) design and implementation of a novel adaptive trajectory tracking controller with unknown parameters and uncertain dynamics. The ability of neural networks is used to design a robust adaptive back stepping controller that does not require the data of the robot dynamics. The kinematic controller gains are set on-line to reduce the velocity error and improve the trajectory tracking characteristics. Shuzhi [42], shows a Non-holonomic mobile robot to achieve smooth path planning in an unknown environment, which are subject to various robot constraints. A hybrid approach is proposed for smooth path planning with global convergence for differential drive Non-holonomic robots. Then, a hybrid path planning approach is presented to guide the robot to move forward along the boundary of an obstacle of arbitrary shape, by causing a proper “Instant Goal” and planning reactively when needed using a fuzzy controller for wall following. Moustris & Tzafestas [43], presented a switching fuzzy logic controller for mobile robots with a bounded curvature constraint. The controller tracks piece-wise linear paths, which are an approximation of the likely smooth reference path. The controller is created through the use of a map, which make over the problem to a simpler one; namely the tracking of straight lines. This agrees to the use of an in effect fuzzy tracker set up in a previous work, and its interpretation leading to a 70% rule reduction.

Muniandy & Muthusamy [44] shows a major drawback with the popular differential drive wheeled mobile robot (WMR) when autonomously navigating on smooth indoor surfaces is its lack of ability to continuously maintain straight-line trajectories. The characteristic of its kinematic design leads to this severe dead reckoning error that inevitably accumulates over the distance moved. The mobile robot then depends on high resolution wheel encoders and rapid feedback control data processing ability that must continuously struggle to minimize this unproductive systematic odometry error. This proposes an innovative and robust drive train mechanical design called dual planetary drive (DPD) that will both drive a non-holonomic wheeled robot in straight lines effectively and more essentially, minimize systematic odometry error without the need for complex feedback control systems.

Sanhoury et al. [45], describe a new synchronization control method is developed for multiple Non-holonomic wheeled mobile robot path tracking while maintaining time-varying formations. Every robot is controlled to track its desired path while its movement is synchronized with nearby robots to maintain the desired time-varying formation. A new derivation for dynamic model of the Non-holonomic wheeled mobile robot (WMR) is proposed based on the Lagrange methods. The robot model is divided into translational and rotational models, such that, each model will be controlled individually. Furthermore, synchronous controller for each robot’s translation is developed to guarantee the asymptotic stability of both position and synchronization errors.

Chattgerjee & Matsuno [46], deals in an effort to mimic the process of obstacle avoidance behavior of human locomotion (or that of automobiles driven by human response), a mixture of basic reflex activities and higher level logical decisions is implemented. It is given away that for reflective navigation of autonomous mobile robots; the capability to reflectively avoid obstacles on one side (left or right) only is adequate for avoidance of obstacle on both sides. When joint with a free-target-approach behavior, the robot can be made capable of navigating through environments with unknown obstacles towards a desired target. In fuzzy logic based implementation of the single-sided reflex is considered. The use of perception symmetry allows perception–action mapping with reduced sensor space dimensions.

Tzafestas [47], presents autonomous mobile robot a System on Chip (SoC) for the path following task. The SoC consists of a parameterized Digital Fuzzy Logic Controller (DFLC) core and a flow control algorithm that runs under the Xilinx Micro blaze soft processor. The fuzzy controller supports a fuzzy path tracking algorithm.

M.Er et al. [48] presents the design and implementation of a neural fuzzy controller suitable for real-time control of an autonomous mobile robot. The neural fuzzy controller is developed based on the Generalized Dynamic Fuzzy Neural Networks (GDFNN) learning algorithm. The parameters of the controller only cannot be optimized, but also the structure of the controller can be self-adaptive. Motlagh et al. [49], describe for reactive navigation mobile robot control technique. The problems of large number of rules, and ineffective definition of contributing factors, e.g., robot wheel slippage, are set on. Causal inference mechanism of the fuzzy cognitive map (FCM) is taking on for deriving the required control values from the FCM's motion concepts and their causal interfaces. The FCM-based control is confirmed to be advantageous over rule-based techniques.

Yaonan et al. [50] investigates the possibility of using transferable belief model (TBM) as a promising alternative for the problem of path planning of Non-holonomic mobile robot equipped with ultrasonic sensors in an unknown dynamic environment, where the workspace is cluttered with static obstacles and moving obstacles. The concept of the transferable belief model is introduced and used to design a fusion of ultrasonic sensor data. A new strategy for path planning of mobile robot is proposed based on transferable belief model. A major advantage of the proposed method is that, with detection of the robot's trapped state by ultrasonic sensor, the navigation law can determine which obstacle is dynamic or static without any previous knowledge, and then select the relevant obstacles for corresponding robot avoidance motion. Simulation is used to illustrate collision detection and path planning.

MacFetridge & Ibrahim [51], stated the Agoraphilic algorithm is an optimistic approach to reactive path planning for mobile robot place. The technique uses essential attractive forces derived from the surrounding free space. Fuzzy logic is making use of to limit the 'free-space' force so as to promote the movement towards the goal. The algorithm was intended to be a robust technique for reactive navigation that could be implemented without the fuss of tuning the sensitive parameters required for other classical navigation routines.

Abdessemed et al. [52], using a fuzzy logic controller situated for which the vehicle tries to reach the endpoint. An evolutional algorithm problem of extracting the optimized IF–THEN rule base is solved. A new approach based on fuzzy concepts is presented in this to avoid any collision with the surrounding environment when this latter becomes relatively complex. Zohar et al. [53], establishes control strategies for Non-holonomic constraints wheeled mobile robots which are subjected to include the kinematic motion and the actuator changing aspects. Using of virtual vehicle and the concept of applying the back-stepping methodology to control schemes for trajectory tracking for the considered augmented model of the mobile robot. Blazic [54], shows a novel kinematic model is proposed where the transformation between the robot posture and the system. A construction of nonlinear control law in the Lyapunov stability analysis framework is presented. Based on the usual requirement this control law achieves a global asymptotic stability for position velocities. The control law is extensively analyses and compared to some existing, globally steady control laws.

Das et al. [55], present, a simple neuron-based adaptive controller for trajectory tracking is developed for Non-holonomic mobile robots without velocity measurements. The proposed controller is robust not only to structured uncertainty such as mass variation but also to unstructured one such as strife. The real-time control of mobile robot is achieved through the online learning. The system stability and the boundless of tracking errors are shown using Lyapunov stability theory.

2.2 Discussions

Navigation in mobile robotic is a methodology that allows guiding an MR to accomplish a mission through an unknown environment with obstacles in a good and safe way. The two basic responsibilities involved in navigation are the environment observation and path following. The navigation problem of an MR can be shared in different sub-problems:

- **Path planning** uses the structures to create an ordered sequence of objective points that the robot must reach.

- **Path generation** is the goal to obtain a path through the sequence of objective points.
- **Path tracking** it is in duty of controlling that the MR follows a path.
- **Obstacle avoidance** when the robot is very close to the target, the striking force between the robot and the target bases the robot seeking towards the target. Similarly when the robot is very close to an obstacle, because of disgusting force developed between the robot and the obstacle the robot must change its speed and angle to avoid the obstacle.
- **Localization strategies** Here we consider that a complete plan of the environment is delivered to the robot. We will be giving different localization strategies and the way they incorporate the information.
- **Sensing** The robot must define its orientation (position and heading) as well as information regarding the environment nearby it. Sensors often are not accomplished directly computing the quantity of interest. Like example, cameras measure colour, brightness and capacity of light, whereas for navigation one might be interested in statements such as "there is a door in front of the robot."

Autonomous robotics is defined as providing robots with some level of intelligence and ability to perform desired tasks without continuous human guidance. Traditionally, the problem of wheeled mobile robots (WMRs) has been used extensively in various industrial and service applications. The applications include security, transportation, inspection and planetary exploration etc. Non-holonomic behavior in robotic systems is particularly interesting, because mechanism can be completely controlled with reduced number of actuators. The non-holonomic mechanical system tools for analyzing and controlling system is based on known mathematical model are presented. Using Lagrange formalism and differential geometry, a general dynamical model can be derived for mobile robots with Non-holonomic constraints. There are also some approaches that tackle both problems simultaneously. We believe that the tracking control approach is somewhat more appropriate, since the Non-holonomic constraints and other control goals (obstacle avoidance, minimum travel time, and minimum fuel consumption) are implicitly included in the path-planning procedure.

Application of fuzzy logic controller is to a variety of industrial systems to an autonomous mobile robot in an unknown environment. The transferable belief model (TBM) is a model for the quantified representation of epistemic uncertainty and which can be an agent, an intelligent sensor, etc., and provides a highly flexible model to manage the uncertainty encountered in the multi-sensor data fusion problems. Application of the transferable belief model (TBM) too many areas has been presented in including classification and target identification during recent times. And we feel it appealing when using navigation of mobile robots.

The importance of cooperative control research of a multi-agent system has been raised in latest decades. This is motivated by the technological advancements and the growth of affordable communication, computation and sensing apparatuses. In order to operate efficiently and fulfill good execution, multi-robot system has to be correctly organized. Formation control has received a lot of attention from the researcher for its many applications such as surveillance, search and rescue, transportation, formation, etc. One of the important cooperative tasks is multi-robot formation control, where a team of robots can maintain the desired formation shape along their path or change the formation shape when required. Several control approaches have been proposed to solve the formation control problem.
With conventional WMRs such as the differential drive robot, dead reckoning is accomplished by monitoring the driven wheel revolutions from a designated start point using incremental optical encoders. Most commercially available WMRs are actuated using a kinematic configuration known as the differential drive. The design involves a pair of diametrically opposed driving wheels that are mounted parallel to each other on a common axis. Individual DC motors actuate the two wheels separately.

Mobile Robot

Mobile Robots produces autonomous robots which are the de-facto standard robotic platforms for researchers around the globe. Besides designing the good hardware, autonomous navigation technology is the core competency of Mobile Robots. The company has developed a suite of technologies including both software and sensors which enable the mobile robots to reliably navigate and explore various environments. The software platform demonstrates great performance when it comes to autonomous navigation. The API of the platform is called ARIA (Advanced Robotics Interface Application). The core software package is called Autonomous Robotic Navigation and Localization (ARNL). It enables the robots to navigate both indoor and outdoor environments with high precision. The different software package of robot makes heavy use of laser range scanners, encoders on wheels, and inertial measurement units, video cameras, and GPS sensors. The software allows wireless control. We click on a map to send a robot to a particular destination.

Over the last decade, mobile robots have been widely used to carry out manifold tasks such as military/industrial applications, planetary exploration, rescue operation and home/medical services. According to a mechanism to achieve the desired mobility, mobile robots may be split into following categories: leg-type, track-type and wheel-type mobile robots. Nonholonomic systems have velocities subject to non-holonomic constraints. Such constraints come up in locomotion applications like wheeled mobile robots and cannot be transformed to pure constraints on the configuration. Compared to holonomic ones, non-holonomic systems are much more difficult to control.

Autonomous mobile robots are intelligent agents which can perform desired tasks in various (known and unknown) environments without continuous human assistance. Many kinds of robots are autonomous to some degree.

A fully autonomous robot in the ability to:

- Gain info about the environment.
- Travel from one position to another, lacking human navigation assistance.
- Avoid circumstances that are harmful to people, property or itself.
- Repair itself without any assistance.

Navigation

Navigation is a field of study that focuses on the process of monitoring and controlling the movement of vehicle from one place to another place. Navigation field includes four general categories: land, marine, aeronautics, and space navigation. It is similarly the term of art used for the specialized knowledge used by navigators to perform navigation jobs. All navigational techniques consist of locating the navigator's position compared to known locations or patterns. Navigation, in a wide-ranging intelligence, can refer to any skill or study that includes the determination of position and direction. In this intelligence, navigation includes orienteering and pedestrian navigation.

For any mobile device, the ability to navigate in its environment is significant. Avoiding dangerous situations such as collisions and unsafe conditions (temperature, radiation, exposure to conditions, etc.) comes first, but if the robot has a purpose that relates to specific places in the robot surroundings, it must find those spaces. In the next, we will present an overview of the skill of navigation and try to detect the basic blocks of a robot navigation system, types of navigation systems, and nearby aspect at its related building constituents.

Robot navigation means the robot's ability to determine its own position in its frame of reference and then to plan a path towards some goal position. In order to navigate in its surroundings, the robot or any other mobility method has need of representation, i.e. a map of the surroundings and the ability to take to mean that representation.

Navigation can be definite combination of the three fundamental competences:

1. Self-localization
2. Path planning
3. Map-building and map interpretation

Map in this context denotes any one-to-one mapping of the world onto an internal representation. Robot localization denotes the robot's ability to establish its own position and orientation. Path planning is effectively an addition of localisation, in that it has need of the determination of the robot's current position and a position of a goal position, both within the same frame of reference or coordinates. Building robot map can be in the shape of a metric map or any representation describing locations in the robot frame of reference.

Vision-Based Navigation

In Vision based navigation uses optical sensors include laser-based range finder and photometric cameras to extract the visual features required to the localization in the surrounding atmosphere. On the other hand, there are a range of techniques for navigation and localization using vision information, the main constituents of each technique are:

- Representations of the environment.
- Sensing models.
- Localization algorithms.

To give a general idea of vision-based navigation and its techniques, we classify these techniques under indoor navigation and outdoor navigation.

Principle of Navigation strategies in an unknown environment:-

In a totally unknown environment, navigation is completely done in a reactive manner. Indeed, a classical method such as artificial potential field can be used. Though, it is known that this method suffers from local minima problems that principal to obstructive conditions. A solution has been based on the automatic tuning of attractive and repulsive forces coefficient using fuzzy rules. But, some alternation problems remain in particularly fine environment pathways, which are very constraining for dedicated indoor utility robotics.

Local navigation method:-

A local navigation method (LNM) with obstacle avoidance is considered for mobile robots in which the dynamics of the robot are taken into account. The goal is known but the geometry and the location of the obstacles are unknown. The mobile robot position is represented by the Cartesian coordinates and can move in three directions, forward, left or right. The starting point and goal points of robot are given. Using these points the directional angle of robot is determined. There may be obstacles in the plane of motion and the objective is to navigate the robot to the goal avoiding the obstacles.

Landmark Finding Strategy:-

The overall structure of the proposed navigation method consists mainly of three components.

(1) Landmark Finding.

(2) Landmark Localization.

(3) Landmark Understanding.

The Landmark Finding detects each landmark in a mobile robot's workstation. Further specifically, for a monocular image taken by the robot, image pre-processing such as edge detection, thinning, and boundary tracing are used to extract the contour of the landmark and its content (symbols depicting the information to the destination). If there are multiple landmarks in the grabbed image, image segmentation techniques must first be performed to separate each landmark. The Landmark Localization determines the relative location of the detected landmark with respect to the robot and provides the necessary geometrical information for inverse perspective transformation which is required for symbol recognition. Usually, in a typical office building, door plates containing office numbers and/or name are common observed landmarks.

The obstacles can be situated at any place on the plan except at the starting point and at the destination point.

Distance and position estimation:-

The robot orientation and the distance from robot to the targeted object are often used as feature parameters for tracking. This method has also suggested measured the distance from the target object to robot by using laser which can track the target continuously when there are no obstacles in the tracking path. Also calculate the distance from the robot to a target object by using two un-calibrated independently moving cameras. Stereo cameras are also popular for tracking, but this method requires complex computation.

Servo Motor Analysis:-

A servo motor has three wire output. Two of them are for power and ground and another one is lead feeds a position control signal to the motor. The position of the servo will be controlled by using PS2 wireless controller. In case of internet control robotic arm the Arduino Uno is the controller of the entire system. Arduino Uno will interface to the internet via an Arduino Ethernet shield, Arduino Ethernet shield will enable the Arduino Uno to interconnect to the internet via LAN cable. Then, any computer that has internet connection can access and control the robotic arm.

Mobile Robot Control

1. **Non-linear Control**

This presents a switching controller approach to address the parking problem in Cartesian coordinates. This strategy is based on the consideration of asymptotically stable subsystems that solve specific navigation actions, namely the positioning control and the heading control, and then designing a simple switching controller including both subsystems, presenting a solution for the parking problem and discussing stability at the switching times. Given the robot at an initial position, it must arrive to the destination posture avoiding the obstacles in between. The uncertainty about obstacles' shapes and positions on the scenario generally leads to the apparition of deadlocks and local minima.

2. **Adaptive predictive control**

The approach to control the mobile robot depends on the available information of the unknown nonlinear system can be known by the input–output data only and the control objectives. The predictive optimization algorithm is used to determine the torque control signal for minimum torque effort. The torque control signal will minimize the cost function in order to minimize the tracking error as well as reduce the torque control effort in the presence of external disturbance. The proposed structure of the adaptive neural predictive controller can be consists of: (a) Position and orientation neural network identifier; (b) Feed forward neural controller; (c) Feedback neural controller.

3. **Tracking control**

In this section, the tracking controller is stated. Conditions are given under which tracking is achieved, i.e., m configuration variables of the system track their reference values asymptotically, while the errors of the remaining k variables are bounded. Robustness with respect to bounded disturbances is shown. Tracking of this robot is using fuzzy logic, using flatness-based control and in complete exact linearization. Like for modelling, the robot is assumed to consist of three rigid bodies: chassis, front axle and rear axle.

4. Visual tracking controller

A visual tracking control law based on the proposed dual-Jacobian visual interaction model for tracking a target of interest in the image plane is derived exploiting feedback linearization and pole placement approaches. In order to control the system state variables from an initial state to the desired state, an error state model is formed to facilitate the tracking controller design.

5. Back-stepping Control

The kinematic based controller or the so called back-stepping controller is a stable tracking control rule for a Non-holonomic mobile robot. The main advantage of the back-stepping kinematic controller is its simplicity and practical application since it relies only on the kinematic model. Other techniques such as state feedback linearization, sliding-mode control, or conventional back-stepping control, require knowledge of the dynamic model and their hardware implementations are not straight forward. Moreover, the performance of the typical back-stepping kinematic controller with constant gains is poor and requires careful gain tuning for each reference trajectory and it does not give zero trajectory error with smooth tracking.

6. Fuzzy Control

In this application the use and simplification of a fuzzy path tracker developed. Fuzzy logic is widely used in machine control. The term itself inspires definite skepticism, sounding equivalent to "half-baked logic" or "bogus logic", but the "fuzzy" part does not discuss to a lack of rigor in the method, relatively to the fact that the logic involved can deal with concepts that cannot be expressed as "true" or "false" but rather as "partially true". Even though genetic algorithms and neural networks can carry out just as well as fuzzy logic in many cases, fuzzy logic has the improvement that the solution to the problem can be cast in terms that human operators can recognize, so that their experience can be used in the design of the controller. This types it easier to mechanize tasks that are already successfully performed by humans.

7. Lyapunov-based control

A first step at controlling the proposed impact system, example of a planar robot colliding with an un-actuated mass-spring system is used to represent a broader class of such systems. The control development is based on the assumption of exact model knowledge of the system. The control objective is to thorough knowledge a robot to collide with an un-actuated mass-spring system and regulate the spring-mass to a desired state. Lyapunov-based methods are used to develop a continuous controller that yields global asymptotic regulation of the spring-mass and robot links. The design of nonlinear controller for mechanical systems has been an extremely active area of research in the last two decades. Often, Lyapunov-based techniques are utilized as the mechanism for developing different nonlinear control structures for mechanical systems. These techniques can often be used to analyze the stability of the closed loop system by using energy like function as the Lyapunov function. The synthesis of controllers basically involves two steps: the design step and the analysis step.

Artificial Intelligence

The word AI was coined by John McCarthy three decades ago. One representative definition is pivoted around the comparison of intelligence of computing machines with human existences. Another way AI is concerned with the performance of machines which "historically have been judged to lie within the domain of intelligence". No one of these definitions or the like has been situated universally accepted, feasibly because of their references to the word "intelligence". A better definition of AI, therefore, calls for formalization of the term "intelligence". The view of Psychologist and Cognitive theorists is of the intelligence helps in identifying the right piece of knowledge at the appropriate instances of decision making. The phrase "AI" thus defined as the simulation of human intelligence on a machine, so as to create the machine efficient to identify and use the right part of "Knowledge" at a given step of solving a problem.

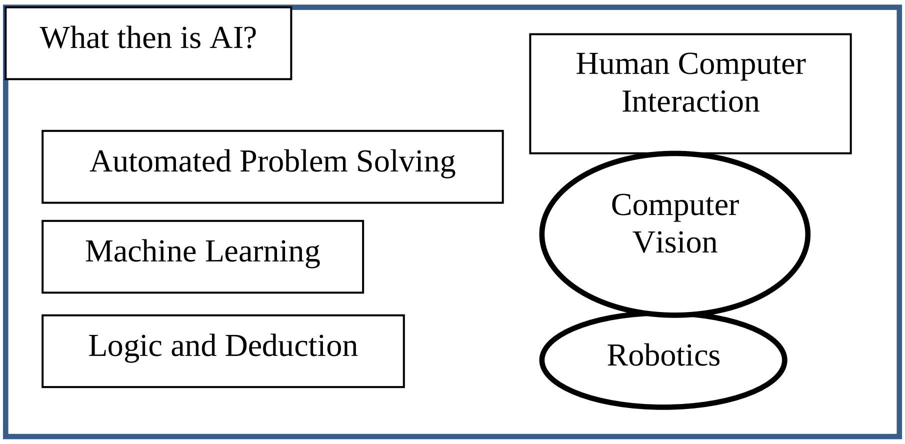


Figure 1- AI Techniques

The way in computers can be made to copy the human think. In this way, we study mental facilities through the use of computational models.

Artificial – not genuine or natural but made by people

Intelligence – The ability to understand study and think

Humankind has given itself the scientific name **homosapiens**—mans the wise—because our mental capabilities are so important to our everyday lives and our sense. In the field of Artificial **intelligence,** or AI, attempts to recognize intelligent objects. Therefore, one reason to study it is to learn more about ourselves.

Philosophy and psychology, which are also concerned with intelligence, AI strives to build intelligent objects as well as understand them. Another goal to study AI is that these constructed intelligent objects are interesting and useful in their own right. AI has produced many important and impressive products even at this early stage in its development.

Some Meaning of Artificial Intelligence:

"The exciting new effort to make computers think . . . machines with minds, in the full and literal intelligence" - Haugeland, 1985

"The automation of activities that we associate with human philosophy, activities such as decision-making, problem solving, learning ..." - Bellman, 1978

"The study of mental faculties through the use of computational models" - Charniak and McDermott, 1985

"The study of the computations that makes it possible to identify, reason, and act" - Winston, 1992

"The art of creating machines that performs functions that require intelligence when performed by people" - Kurzweil, 1990

"The study of how to make computers do things at which, at the instant, people are better" - Rich and Knight, 1991

"A field of study that pursues to explain and compete with intelligent behaviour in terms of computational processes" - Schalkoff, 1 990

"The branch of science that is concerned with the robotics of intelligent behaviour" -Luger and Stubblefield, 1993

Approach of Artificial Intelligence

Many approaches in artificial life to groups of physical robots, which take into consideration interactions between robots, prefer the simulation of social insect societies which are anonymously organized societies without individual relationships. The individuals problem which cannot be solved by a single agent interact only for cooperation, tackling. Other approaches take other agents only into consideration as moving obstacles or as competitors for limited resources. Interactions also dominate approaches in distributed artificial intelligence and multi-agent systems on collective agents.

Another way is to make computational models of human thought processes. This is a stronger and more constrained view of what the enterprise is. It is not enough to make a program that seems to behave the way humans do; you want to make a program that does it the way humans do it. A lot of people have worked on this in cognitive science and in an area called cognitive neuroscience. The research strategy is to affiliate with someone who does experiments that reveal something about what goes on inside people's heads and then build computational models that mirror those kinds of processes. Conventional approaches represent complex systems in a reductionist manner by specifying well-defined components and their individual interactions. We will investigate the question why these approaches are of limited use in artificial intelligence and cognitive science.

CHAPTER 3
KINEMATIC ANALYSIS OF MOBILE ROBOT

Kinematics is the most basic study of how mechanical systems behave and it plays greater role to follow a desired trajectory. In mobile robotics, it is necessary to understand the mechanical behavior of the robot both in order to design appropriate mobile robots for tasks and to understand how to create control software of mobile robot hardware. This chapter provides a detailed kinematic analysis of mobile robot and used to DH representation to kinematic analysis of Corobot Arm. The numerous aspects of designing a wheeled mobile robot can be depicted as: positioning of the robot in its environment, maneuverability analysis according to its kinematic constraints, generalized control of the developed Kinematic and Dynamic model. In this way 4-DOF Robotic arm kinematic analysis in the positions and orientation to different angles of a manipulator has to be presented. Here a Phidgets to be used for controlling the robotic arm with a servo motor also with the help of Phidget Control Panel an wheeled to be moved in both forward & reverse direction by high current 2-motor controller.

3.1 Introduction

Robots are mainly concerned with causing specific motion of the robot joints, at the same time allowing tooling or sensors to perform certain purposes, either when the arm is moving or at specific working configurations. The arm and fond of tooling may perform the operations themselves (such as painting) or carry parts to other devices which perform the operations [56].

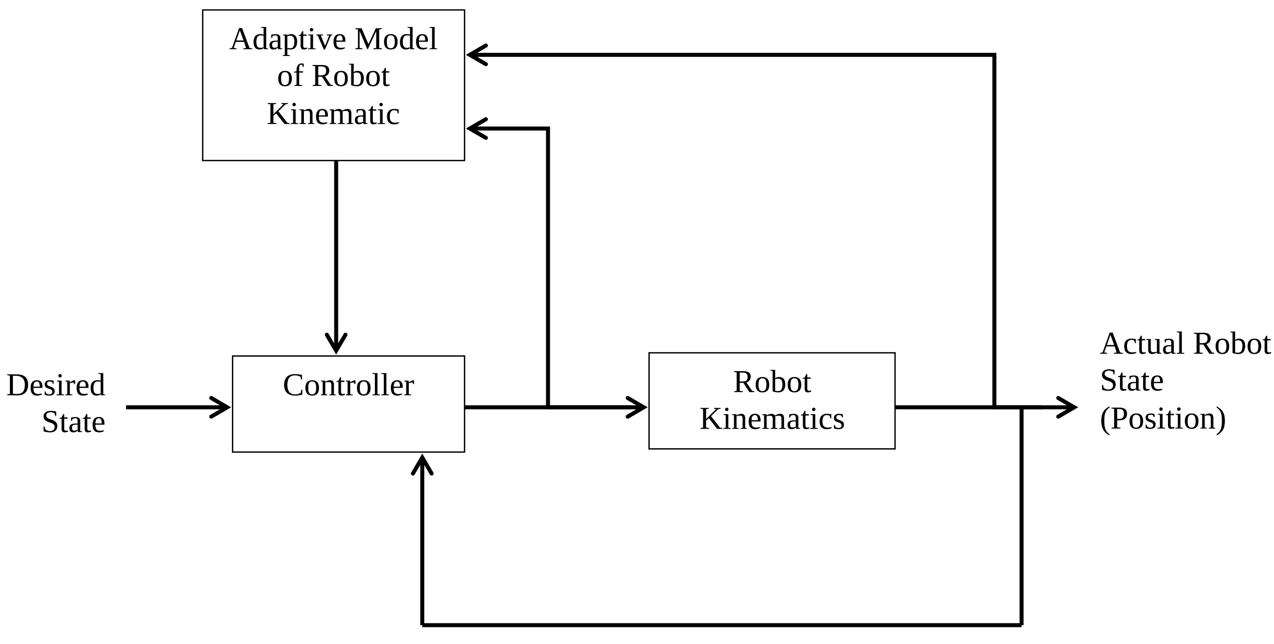


Figure 2- Robot Kinematics States

Newer technologies are concerned with robot interactions with parts such that interaction forces and torques can be well-ordered. This tools will permit more robot applications in bring together, which promises to be a growing application for robotics [57-58].

Robot kinematics is the study of the motion (kinematics) of robotics device. In a kinematic analysis the position and orientation of all the links are calculated without considering the forces that basis this motion. The relationship between motion, and the related forces and torques is studied in robot dynamics [60]. A mobile robot that carries out navigation tasks in an office-like environment, navigation from one position to another, also needs to reasons out its relationship with the environment in order to compute the best path between the current and the goal position [59]. Robots are more efficient where the parts to be handled are hot or extremely heavy. Therefore, there is a place for robots in industry and increasing applications are being explored for these programmable machines [61].

3.2 Degrees of Freedom (DOF)

The degrees of freedom, or DOF, are a very important term to know for each degree of freedom is a joint on the arm, a place where it can twist or rotate or translate.

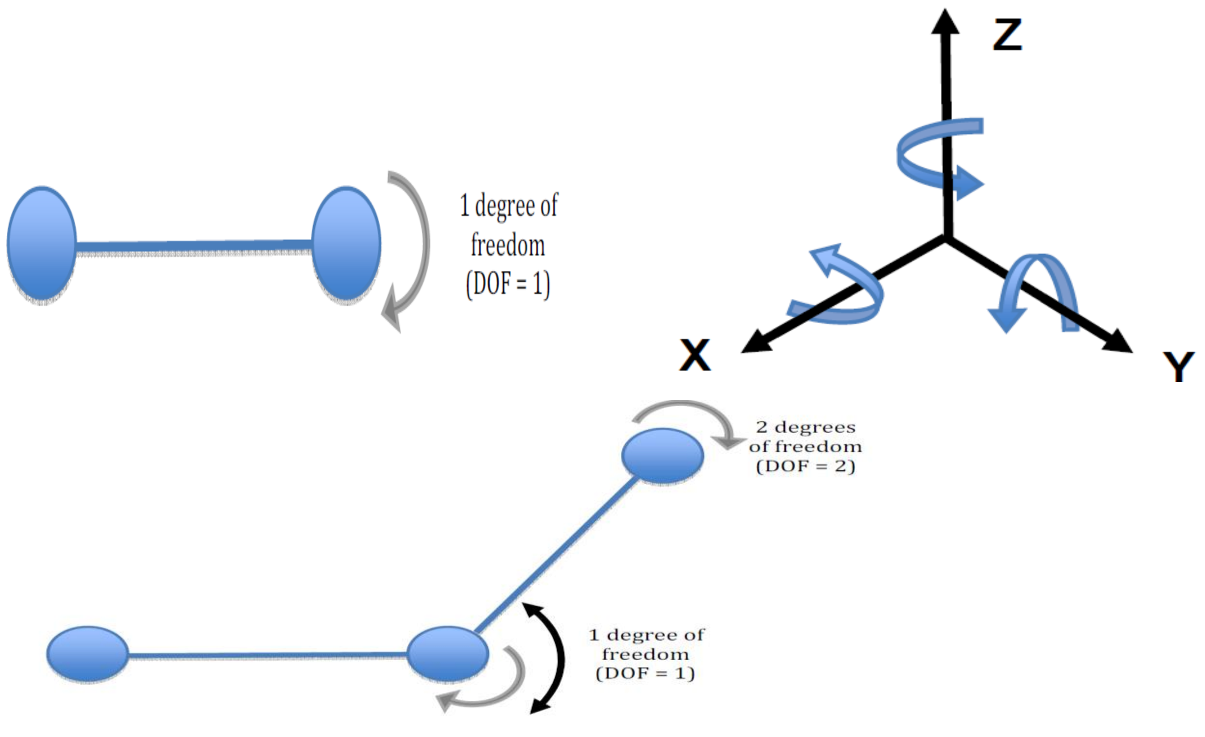


Figure 3- DOF a. Single DOF b. Different Coordinate with translation and rotation c. Links with single and double DOF

We can usually identify the number of degrees of freedom by the number of actuators on the robot arm. Now this is very important - when building a robot arm we want as few degrees of freedom allowed for our application, why? Because of each degree requires a motor, often an encoder, and exponentially complex algorithms and cost.

1. Introduce the concept of **degrees of freedom**. In terms of motion, a degree of freedom is a plane of movement. Like for example when shaking someone's hand up and down, the arm travels vertical direction. That is one degree of freedom. Each joint in a limb corresponds to one plane of freedom.

2. The concept of degrees of freedom is central to **kinematics**, the study of motion without reference to the forces that cause motion.

3. The lettered axes, X, Y and Z represent translational motion, whereas the blue arrows represent rotational motion.

3.3 Forward and Inverse Kinematics

Forward Kinematics - Forward kinematics specifies the joint parameters and computes the configuration of the series links. For sequential manipulators this is achieved by direct substitution of the joint parameters into the forward kinematics equations for the sequential chain. For parallel manipulators change of the joint parameters into the kinematics equations requires solution of the set of constraints to determine the set of possible end-effector locations. Forward kinematics is the method for determining the orientation and position of the end effector, for particular joint angles and link lengths of the robot arm.

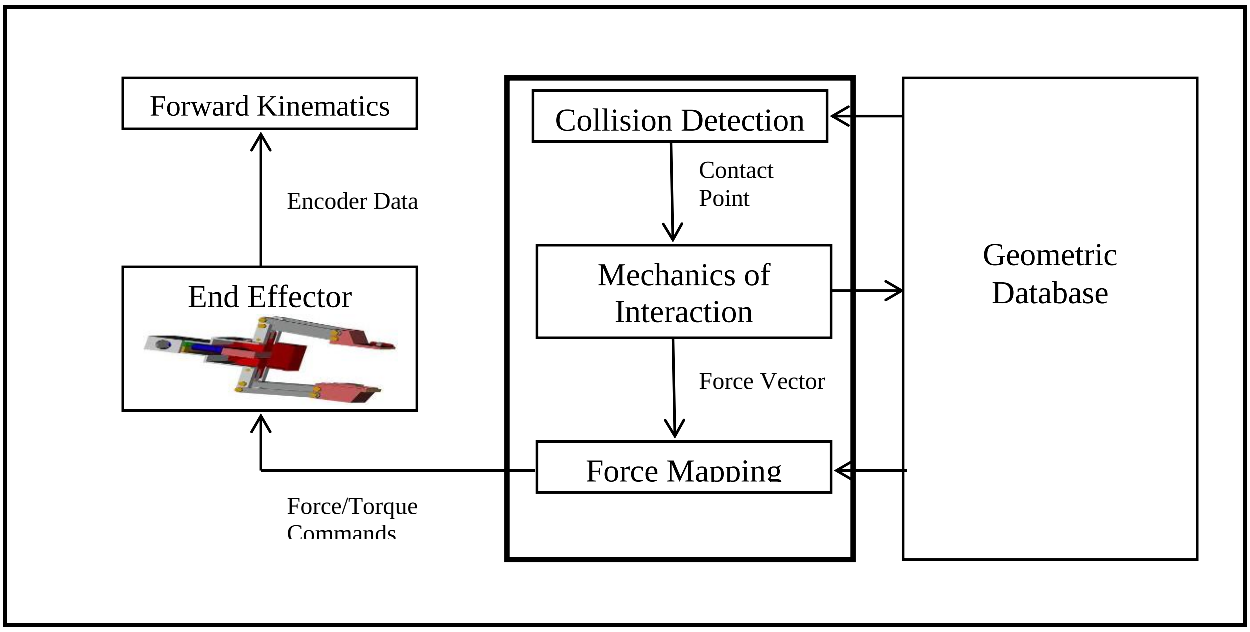


Figure 4- Forward Kinematics Procedure

The two methods for a forward kinematic analysis are as follows:

- Using straightforward geometry
- Using transformation matrices

Inverse Kinematics - Inverse kinematics specifies the end-effector location and computes the related joint angles. For sequential manipulators this requires way of a set of polynomials obtained from the kinematics equations and yields multiple configurations for the chain structure. For parallel manipulators, the specification of the end-effector location simplifies the kinematics calculations, which yields formulas for the joint parameters.

Inverse kinematics is the opposite of forward kinematics. This is when we have a desired end effector position, but essential to know the joint angles required to attain it. The robot sees a kitten and requirements to take it, what angles should each joint go to? Although for more useful than forward kinematics, inverse calculation is much more complicated.

3.4 Denavit-Hartenberg (DH) Convention

The Denavit-Hartenberg (DH) representation is the conventional method of drawing robot arms in FBD's. There are only two motions translate and rotate a joint could make. There are only three axes x, y, and z (out of plane) this could be happen. Below we show a robot arms, and then draw a FBD next to it, to show the DOF relationships. Here Note that we did not count the DOF on the gripper (otherwise known as the end effector). The gripper is a lot complex with multiple DOF, for simplicity it is treated as separate in basic robot arm design.

In each DOF there is a linkage of some specific length. Sometimes a joint can have multiple DOF in the same position. Like example would be the human shoulder. The shoulder actually has three equivalents DOF. If mathematically represent this, we would just say link length = 0. Also note that a DOF has its limits, known as the configuration space. Not all joints can rotate 360 degrees; a joint has some maximum angle restraint. Like example, no human joint can rotate more than about 200 degrees. Restrictions could be from wire wrapping, actuator abilities, servo max angle, etc. It is a good idea to make each link length and joint max angle on the FBD.

Now let’s assume that all joints rotate a maximum of 180 degrees, for the reason that most servo motors cannot exceed that amount. To define the workspace, trace all locations that the end effector can reach as in the image below. Now rotating that by the base joints another 180 degrees to acquire 3D, we have this workspace appearance. Remember that because it uses servos, all joints are some degree of to a max of 180 degrees. Since there are many possible configurations for robot arm, from now on we will only talk about the one shown below.

Figure 5- Human Arm

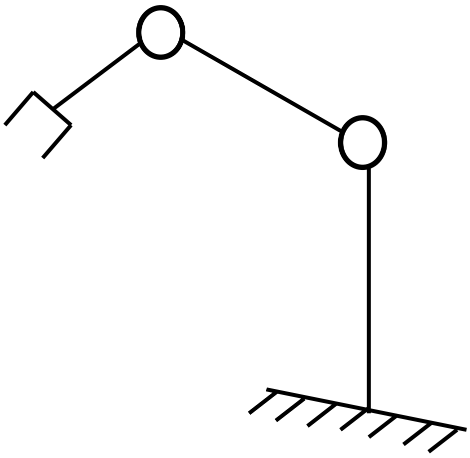


Figure 6- Robotic Arm same as Human Arm

3.5 Kinematic Analysis of Manipulator

Robot kinematics is the study of the motion of manipulator of robots. In a kinematic analysis the position, of all the links are calculated without considering the forces that cause this motion. Motion relationship associated with forces and torques is studied in robot dynamics. While dealing with the kinematics used in the robots we deal each parts of the robot by assigning a frame of reference to it and hence a robot with many parts may have many individual frames allocated to each movable parts. For easiness we deal with the single manipulator arm of the robot. In kinematic analysis of manipulator position, by two separate problems direct kinematics, and inverse kinematics to solve. In Direct kinematics problem we solve the forward transformation equation to find the location of the arm in terms of the angles and displacements between the links. And in Inverse kinematics involves solving the inverse transformation equation to find the relationships between the links of the manipulator from the location of the arm in space. A robot arm is known manipulator. It is composed of a set of joints separated in space by the arm links. The joints are where the motion in the arm occurs. In basic, a robot arm consists of the parts: base, joints, links, and a gripper. The base is the basic part over the arm; it may be fix or active. The joint is flexible and joins two separated links. The link is fixing and supports the gripper. The last part is a gripper. The gripper is used to hold and move the objects.

Each frame are named analytically with numbers, like example the fixed base part of the manipulator is numbered 0, the first link joined to the base is 1, and the next link is 2 and similarly till n for the last nth link.

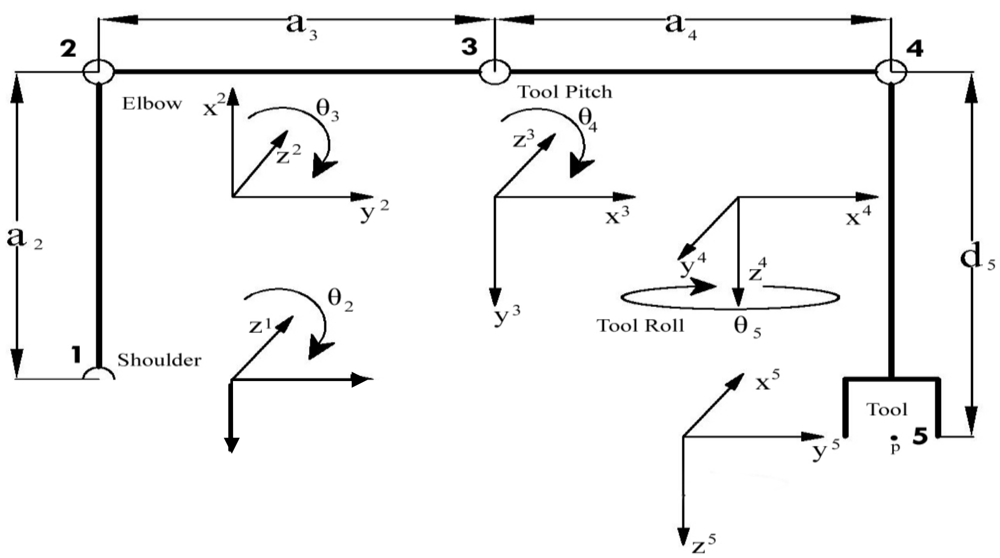


Figure 7- Link Coordinate-Frame of the Manipulator

Using Denavit - Hartenberg (DH) convention, coordinate frames for the manipulator are assigned as shown in the Fig 7. 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 1, where θi is the rotation about the Z-axis, αi rotation about the X-axis, di transition along the Z-axis, and ai transition along the X-axis.

Table-1

KINEMATIC PARAMETER OF THE ARM				
Axis	Ɵ	d (mm)	a (mm)	α
1	Ɵ1	0	0	180
2	Ɵ2	0	a2	0
3	Ɵ3	0	a3	0
4	Ɵ4	0	a4	0

The set of link coordinates assigned using DH convention is then transformed from coordinate frame (ki) to (k$i-1$), where k is the joints, using a homogeneous coordinate transformation matrix given in eq. (1).

$$A_i = Rot(z,\theta_i) * Trans(0,0,d_i) * Trans(a_i,0,0) * Rot(x,\alpha_i)$$
$$= \begin{bmatrix} Cos\theta_i & -Sin\theta_i * Cos\alpha_i & Sin\theta_i * Sin\alpha_i & \alpha_i * Cos\theta_i \\ Cos\theta_i & Cos\theta_i * Cos\alpha_i & -Cos\theta_i * Sin\alpha_i & \alpha_i * Sin\theta_i \\ 0 & Sin\alpha_i & Cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{……..(1)}$$

On substituting the DH parameters in Table 1 into eq. (1), we get individual transformation matrices T01toT45, and a global matrix of transformation T05 as in eq. (2):

$$T_0^5 = T_0^1 * T_1^2 * T_2^3 * T_3^4$$
$$= \begin{bmatrix} m_x & n_x & o_x & p_x \\ m_y & n_y & o_y & p_y \\ m_z & n_z & o_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{..…….(2)}$$

Where (p_x,p_y,p_z) represents the position and ({m_x,m_y,m_z}, {n_x,n_y,n_z}, {o_x,o_y,o_z}) the orientation of the end-effector.

$m_x = C_1C_{234}C_5 + S_1S_5$ ..…….(3)

$m_y = S_1C_{234}C_5 - C_1S_5$ ..…….(4)

$m_z = -S_{23}4C_5$ ..…….(5)

$n_x = -C_1C_{234}S_5 + S_1C_5$ ..…….(6)

$n_y = -S_1C_{234}S_5 - C_1C_5$ ..…….(7)

$n_z = S_{23}4S_5$ ..…….(9)

$o_x = -C_1S_{234}$ …….(10)

$o_y = -S_1S_{234}$ …….(11)

$o_z = -C_{234}$ …….(12)

$p_x = C_1 (a_2C_2 + a_3C_{23} + a4C_{234} - d_5S_{234})$ …….(13)

$p_y = S_1(a_2C_2 + a_3C_{23} + a_4C_{234} - d_5S_{234})$ …….(14)

$p_z = d_1 - a_2S_2 - a_3S_{23} - a_4S_{234} - d_5C_{234}$ …….(15)

Here Ci=cos(θi), Si=sin(θi), Cij =cos($\theta i + \theta j$), Sij =sin($\theta i + \theta j$), $Cijl$ =cos($\theta i + \theta j + \theta l$), $Sijl$ =sin($\theta i + \theta j + \theta l$).

3.6 Experimental Result of Manipulator

The Corobot arm has four servos motors (0, 1, 2 & 3) position which are controlled through the use of phidgets control panel. The arm is very user friendly because of the computer interface developed by us. They could lift objects up to weight of 8 oz. The robotic arm can be designed to perform any desired task such as welding, gripping, spinning etc. depending on the application. The robot arms can be autonomous or manually controlled and can be used to perform a variety of tasks with great accuracy. The robotic arm can be fixed or mobile (i.e. wheeled) and can be designed for industrial or home applications. Robotic hands often have built-in pressure sensors that express the computer how hard the robot is gripping particular objects. Under Phidget Servo, Phidget Moter controller and a Phidgets servo controller are to be studied.

Figure 8- Corobot View in Lab

Figure 9- Arm Position & Orientation

Control of the robot arm may be separated into shoulder; elbow, wrist and gripper are shown in the figure 9. In arm 4 motors are there in phidgets control panel these motor are numbered by 0, 1, 2 & 3 with respect to shoulder, elbow, wrist and gripper respectively are shown in the figure 10 & 11.

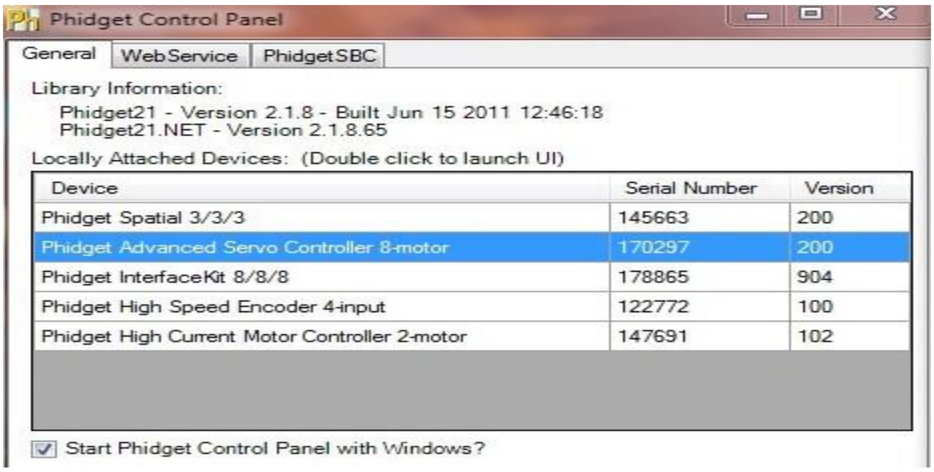


Figure 10- Phidget Control Panal Screen

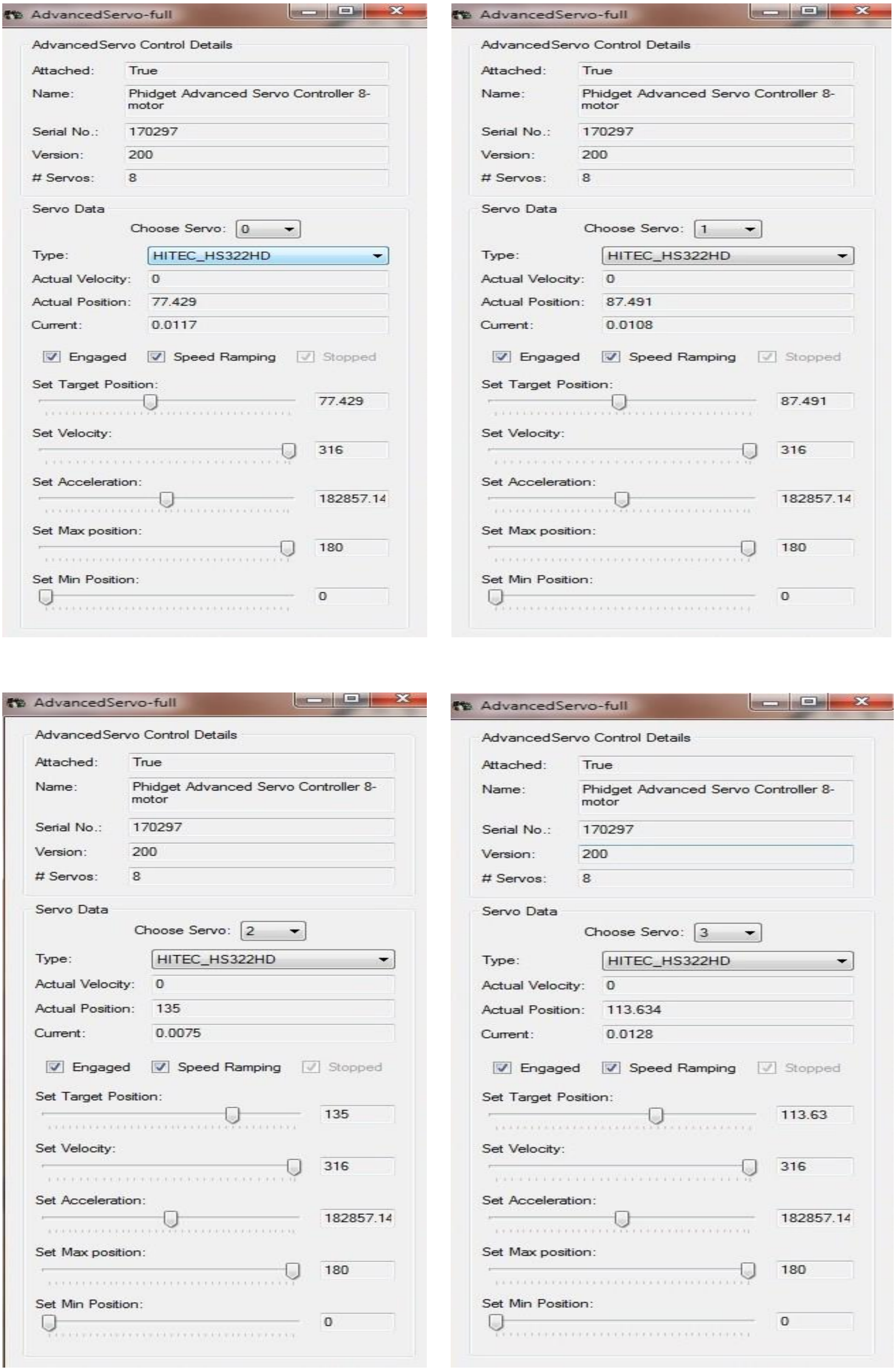


Figure 11- Phidget Control Panal Servo-Motor (0, 1, 2 &3) Screen

In this study first set the servo motor number (like 0, 1, 2 or 3) then target position are to be set. Also velocity & acceleration are maintained though the required position and motion speed then select the engaged option so arm has to move. Shoulders and elbow moves in the up and down direction but wrist are rotate in either clockwise or anticlockwise direction and also gripper is open or closed.

Table 2:- Experiment Result of Manipulator

Servo Motor No.	Minimum Position (degree)	Maximum Position (degree)	Set Target Position (degree)
0	0	180	77.429
1	0	180	87.491
2	0	180	135
3	0	180	113.63

Figure 12– Target Position of Corobot Manipulator using Phidget Control

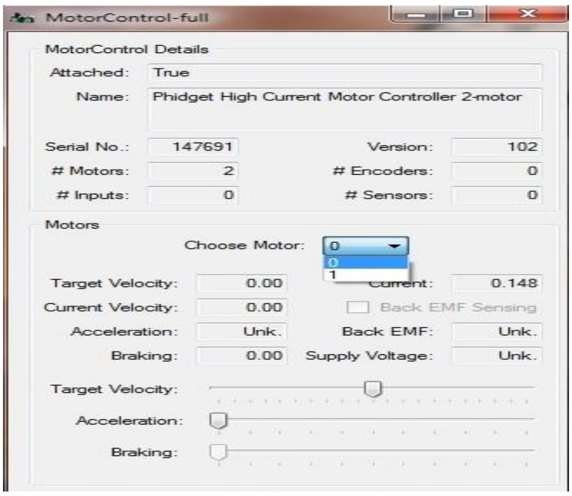


Figure 13– Phidget Motor Control

Also in these Corobot used to Phidgets Control Panel high current motor controller 2-motor four wheeled are moved in both forward and reverse direction to maximum 100 target velocity. In these 2-motor are given by a numbered 0 & 1 where servo 0 is left wheeled motor and servo 1 is right side wheel of the motor are shown in the figure below.

CHAPTER 4
NEURAL NETWORK / WEBOT / COROBOT

This chapter provides an approach for detail analysis and simulation of the 4-wheeled vehicle i.e. Corobot study for path navigation using a technique Neural Network (NN) and with the help of Webot. NN methods works on the principle of the neurons present in a normal human brain which are the basically data transfer element. In this neural network, we propose a methodology for training a new model of artificial neural network called the generalized radial basis function (GRBF) neural network. The inputs to the proposed neural controller consist of left, right, and front obstacle distance with respect to its position and target angle. The output of the neural network is steering angle. A four layer neural networks has been designed to solve the path and time optimization problem of mobile robots that deals with the cognitive tasks such as learning, adaptation, generalization and optimization. Back propagation algorithm is used to train the network. The training of the neural nets and the control performances analysis of the neural network has been done in a real experimental setup. This approach can be used very intelligently in autonomous robotic control in industrial, mining or exploration, military operation, space exploration, search and rescue in unstructured environments, rapid construction of arbitrary tools under space, etc.

Before starting the main parts of this chapter first we introduce about robot overview because this vehicle is also a robot.

Robotics Overview

Robots are primarily concerned with generating specific motion of the robot links, simultaneously allowing tooling or sensors to perform definite functions, either when the arm is moving or at precise operational configurations. The arm and involved tooling may perform the operations themselves or carry parts to other devices which perform the operations. Newer technologies are concerned with robot interactions with parts such that interaction forces and torques can be controlled. This tools will permit more robot applications in assemble, which promises to be a growing application for robotics.

Figure 14- Explorer

Robot Definition

There's no precise definition, but by general agreement a robot is a programmable machine that imitates the actions or presence of an intelligent being-usually a human. To be suitable as a robot, a machine has to be able to do two things: 1. Get information from its environments and 2. Do to some degree physical-such as move or manipulate objects.

"A robot is a reprogrammable, multi-functional manipulator designed to move physical parts, tools, or specialized devices through variable programmed motions for the performance of a variety of tasks."

Where Used & Applied?

Robots are used in almost any industry where repetitive jobs are involved, or the job is difficult manually, or hazardous, such as

- Welding, painting, or surface finishing in the aerospace or automotive industries
- Electronics and customer products assembly and inspection
- Inspection of quantities by robot assisted sensors
- Underwater and space exploration
- Hazardous waste remediation

4.1 Neural Network

4.1.1 Introduction

Artificial neural networks (ANN) are largely used in applications involving classification or task estimate. Among them, we find radial basis function neural networks (RBFNNs), multi-layer perceptions or back-propagation algorithm. All are multi-layered networks and can be considered as connectionist models. RBFNNs use, in general, hyper-ellipsoids to split the pattern space. This is different from build their classifications on pseudo-hyper-planes, defined by a weighted sum [62]. A new learning algorithm called the Extreme Learning Machine (ELM) has recently been proposed for single hidden layer feed forward neural network. This novel procedure, unlike conventional implementations of gradient-based learning algorithms, chooses randomly hidden nodes and analytically determines the output weights of the network. This algorithm provides good generalization performances at extremely fast learning speeds and in theory the universal approximate property has proved to hold true [63].

A new error-driven active learning approach to self-growing radial basis function networks for early robot learning. There are several mappings that need to be set up for an autonomous robot system for sensorimotor coordination and transformation of sensory information from one modality to another, and these mappings are usually highly nonlinear [64]. In recent years, computer vision has been widely used on industrial environments, allowing robots to perform important tasks like quality control, inspection and recognition. Vision systems are typically used to determine the position and orientation of objects in the workstation, enabling them to be transported and assembled by a robotic cell (e.g. industrial manipulator). These systems commonly resort to Cameras fixed and located in a particular work area or attached directly [65]. The RBFN and Back-propagation (BP) algorithm is one of the most powerful tools to quickly obtain a NN that gives an input-output relation close to the training data sets. In case of constructing a robot autonomously behaving in environment, however, it has been difficult to think of getting the "training data", because autonomous behavior should have no template and only be obtained by gradual improvement in the environment [66, 67].

4.1.2 Analysis of Neural Network for Navigation

Artificial neural networks consist of a set of simple, densely interconnected handling units. These units transform motions in a non-linear way. A non-parametric Neural networks are estimators, which can fit smooth functions based on input-output examples. The neural network used is a three-layer. The numbers of layers are found empirically to smooth training pattern. The input layer has four neurons, three for receiving the values of the distances from obstacles in front and to the left and right of the robot and one for the target achieving. If no target is detected, the input to the fourth neuron is set to zero. The output layer has a single neuron, which produces the steering angle to control the direction of movement of the robot. The numbers of neurons are found based on the number of training patterns and the convergence of error during training to a minimum verge error. One hidden layers are used, as with training the neural network within a specified error limit. The training error is the difference between desired output and actual output.

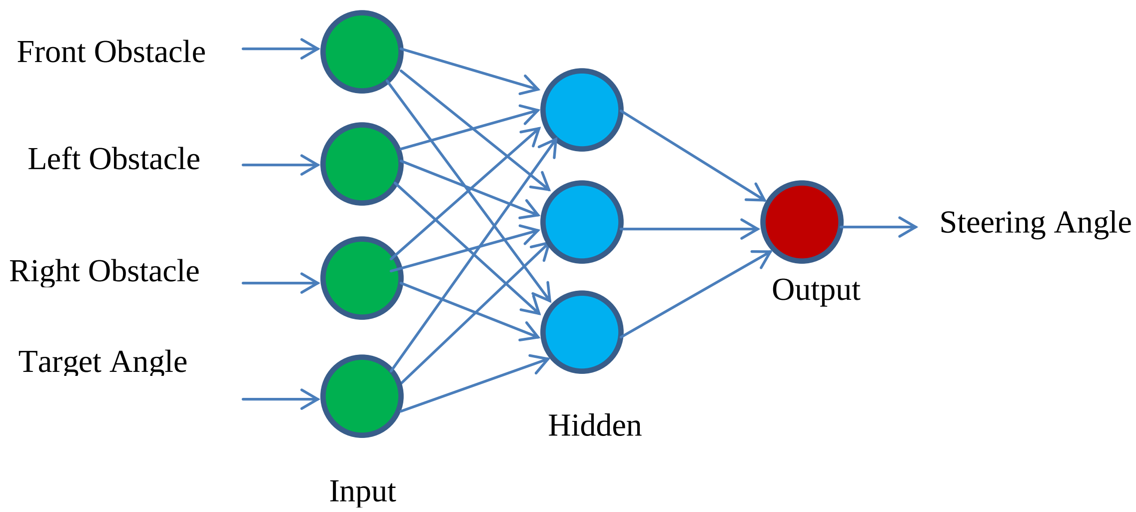


Figure 15- Simple Neural Network

Fig. shows the neural network with its input and output signals. The neural network trained to navigate by presenting it with 20 patterns representing typical scenarios, some of which are depicted in Fig. 12. For example, For example a robot is advancing towards an obstacle, another obstacle being on its right hand side. There are no obstacles to the left of the robot and no target within sight. The neural network is trained to output a command from the robot to steer towards its left. Table 2 shows the list of empirical training patterns based on Fig. 12.

During training and during normal operation, the input patterns fed to the neural network comprise the following components:

x_1 = Left obstacle distance from the robot

x_2 = Front obstacle distance from the robot

x_3 = Right obstacle distance from the robot

x_4 = Target bearing of the robot

These input values are distributed to the hidden neurons that generate outputs given by

$x_2 = f(V_j^{\{layer\}})$

Where $V_j^{\{layer\}} = \sum_i W_{ji}^{\{layer\}} * V_i^{\{layer-1\}}$

layer = layer number

j = label for j^{th} neuron in hidden layer 'layer'

i = label for i^{th} neuron in hidden layer 'layer-1'

$W_{ji}^{\{layer\}}$ = weight of the connection from neuron i in layer 'layer-1' to neuron j in layer

It should be noted learning can take place continuously even during normal target seeking behavior. This enables the controller to adopt the changes in the robot's path while moving towards target. Mainly three behaviors (obstacle avoidance, wall following and target seeking) are required to train and to design an intelligent controller for mobile robot being used to navigate in an environment. Table depicts the used behavior being trained by network.

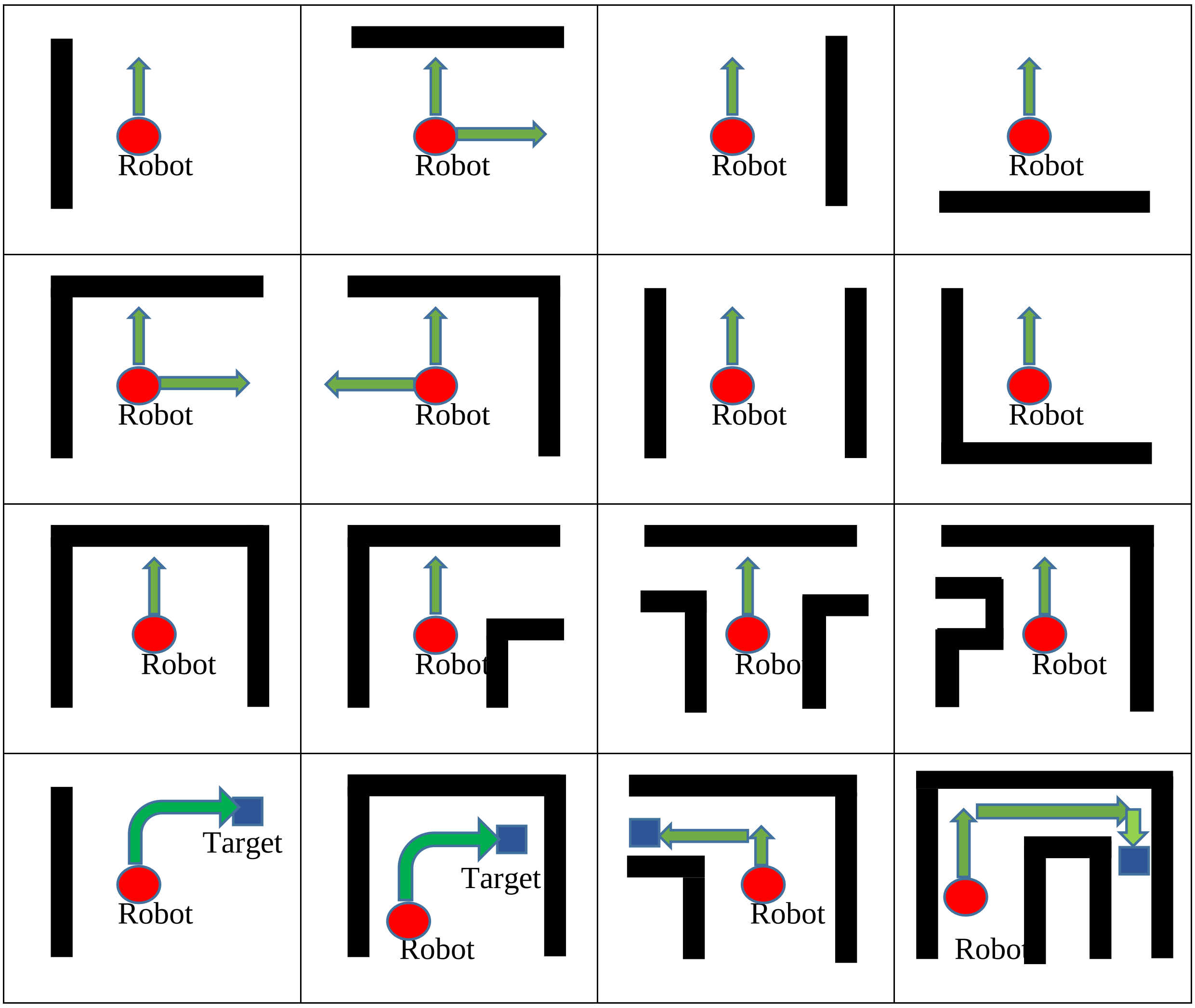


Figure 16- Example of training pattern for obstacle avoidance, wall following, Target achieving

In above example of training pattern we use following behavior.

Obstacle avoidance: - Mobile robot detects (by sensory information) any obstacle in front, left or right side. This behavior required to avoid collision with obstacle. The robot reduces the speed and set the steering angle accordingly. When information is given from the sensors shows the presence of obstacles to the front, left and right side of the robot. The robot reverses its movement. The robot stopped and takes counter clockwise rotation both left and right wheel in same speed (i.e. reverse direction).

Wall following: -Mobile robot detects an obstacle in the front while it is moving towards target and also having wall to the left or right side. The robot has to follow the wall to reach the target. The robot adjust the speed and set the heading angle with wall so that it align with wall and moves along the wall. The robot automatically makes turns to align itself along the wall and move parallels with the wall to reach the target.

Target Achieving:-When the acquired information from the sensors shows that there are no obstacles around the robot, its main reactive behavior is to seek the target. This behavior requires locating the target. The robot mainly adjusts its motion direction and quickly moves towards the target.

Table3: Some Training Pattern Data

S. No.	Inputs of the Network				Output
	Front Obstacle Distance (cm)	Right Obstacle Distance (cm)	Left Obstacle Distance (cm)	Target Angle (degree)	Steering Angle (degree)
1	60	15	60	0	0
2	20	60	60	0	90
3	60	60	15	0	0
4	60	20	60	0	-90
5	25	20	60	0	90
6	25	20	60	0	-90
7	60	20	20	0	0
8	20	60	20	0	0
9	20	15	20	0	-180
10	30	15	15	0	-30
11	30	20	20	0	15
12	60	15	20	25	20
13	60	20	25	15	10
14	60	20	15	0	0
15	60	20	60	-20	-15
16	75	15	25	30	20
17	75	15	60	30	20
18	20	75	10	-45	-35
19	20	60	25	45	35
20	30	60	25	0	5

4.1.3 Radial Basis Function

We chose the radial basis function (RBF) networks as the main framework for our study of robot behavior learning since it possesses the property of best approximation and gives a smoother fit to the training data. Object recognition is a stimulating task for many applications, especially in the field of robotics where the interaction between robots and their environment becomes a very challenging problem. Moreover, it is possible to train the RBF network in two stages, with the basic functions first being determined by unsupervised learning and then the second layer weights being determined by supervised learning. In the system, a self-organizing map is first used to cluster the robot's sensory information into prototypes according to a topographic mapping, and then the RBF is used to implement the non-linear mapping from sensory space to motor action space.

This proposes the use of radial basis function neural networks approach to the solution of a mobile robot orientation adjustment using strengthening knowledge. In order to control the orientation of the mobile robot, a neural network control system has been constructed and executed. Neural controller has been charged to improve the control system by adding some degrees. Making use of the potential of neural networks to learn the relationships, the desired location orientation and the position of the mobile robot are used in training. This is becoming an increasingly popular neural network with different applications and is probably the main challenging to the multi-layered perceptron. Much of the motivation for RBF networks has come from traditional pattern techniques. A function is radial basis if its output depends on the distance of the input from a given stored point. RBF represent local receptors, as illustrated below, where each green point is a put away point used in one RBF.

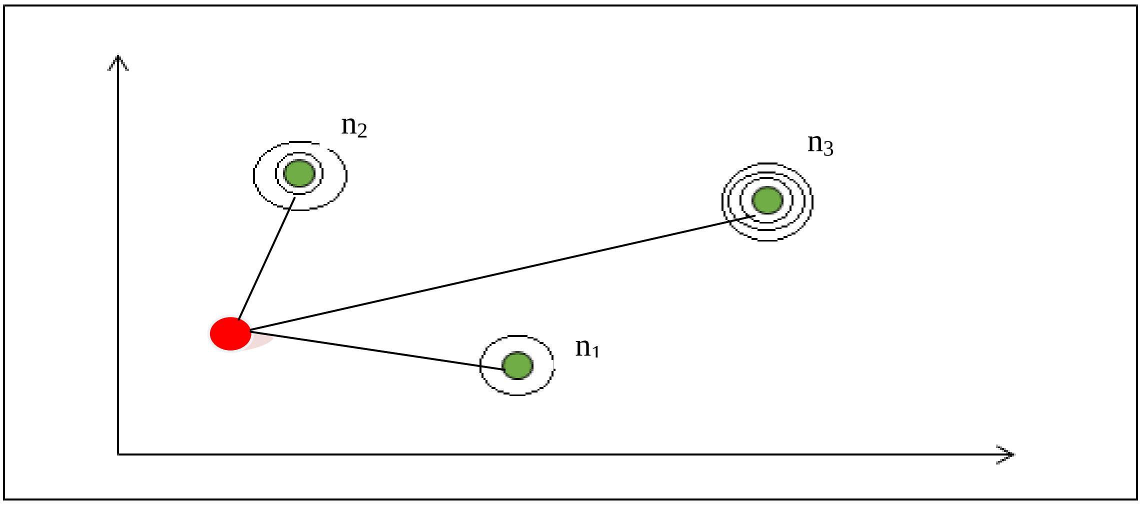


Figure 17- Simple RBFN based Neuron

The output of the red point is interpolated using the three green points, where each point gives a contribution that depends on its weight and its distance from the red point. In this figure we have, $n_1 < n_2 < n_3$.

We investigated the following four behaviors on the mobile robot:

- Obstacle Avoidance: avoids obstacles while robot moving forwards.
- Wall Following: the robot moves along the wall by keeping a constant distance to the wall.
- Path Learning: the robot moves along the trajectory of a shown path.
- Forward-Backward Moving: the robot repeatedly moves straight forwards until it encounters an obstacle, turns 180 degrees, and then travels forward again, turning 180 degrees when an obstacle is encountered in the opposite direction.

Radial Basis Function networks have been employed for several purposes from function estimation to control. RBF networks are composed of 3 layers of neurons; input layer, hidden (RBF) layer and output layer.

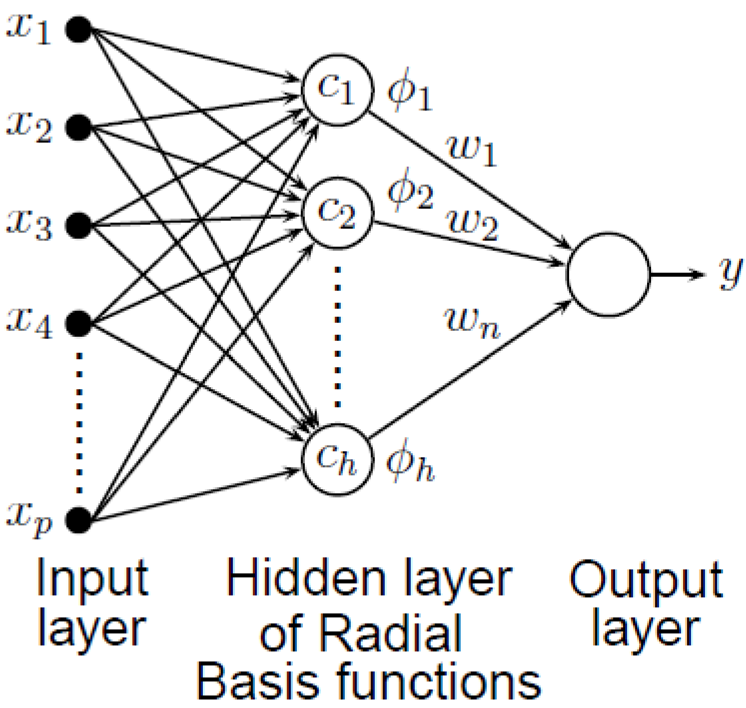


Figure 18- RBFN Network

The hidden units provide a set of functions that constitute an arbitrary basis for the input patterns.

- Hidden units are known as radial centres and represented by the vector $c_1, c_2 \ldots c_3$.
- Transformation from input to hidden unit is nonlinear whereas transformation from hidden unit space to output space is linear.
- Dimension of each centre for a p input network is p×1.

The radial basis functions in the hidden layer produce a significant non-zero response only when the input falls within a small localized region of the input space. Each hidden unit has its own receptive field in input space.

An input vector xi which lies in the receptive filed for centre c_j, would activate c_j and by proper choice of weights the target output is got. The output is given as

$$y = \sum \phi_j w_j \qquad \phi_j = \phi\left(\left\| x - c_j \right\|\right)$$

w_j: weight of j^{th} centre, ϕ: some radial function

Some different radial functions are given as follows.

Gaussian radial function $\phi(z) = e^{-z2/2\sigma2}$

Thin plate spline $\phi(z) = z^2 \log z$

Quadratic $\phi(z) = (z^2 + r^2)^{1/2}$

Inverse quadratic $\phi(z) = 1/(z^2+r^2)^{1/2}$

Here, $z = \|x - c_j\|$

The most popular radial function is Gaussian activation function.

Training of RBFN requires optimal selection of the parameters vectors c_i and w_j. Both layers are optimized using different techniques and in different time scales. Following techniques are used to update the weights and centres of a RBFN.

- Pseudo-Inverse Technique (Off line)
- Gradient Descent Learning (On line)
- Hybrid Learning (On line)
- This is a least square problem. Assume a fixed radial basis functions e.g. Gaussian functions.
- The centres are chosen randomly. The function is normalized i.e. for any x, P i = 1.
- The standard deviation (width) of the radial function is determined by an adhoc choice.

A neural network training pattern to perform some task, we must change the weights of each unit in such a manner. Radial functions are simply a class of functions. They could be employed in any sort of model linear or nonlinear and any sort of network single layer or multi-layer. An RBF network is nonlinear if the basic functions can move or change size or if there is more than one hidden layer. Here we emphasis on single layer networks with functions which are fixed in position and size.

4.1.4 Back Propagation Algorithm in Neural Network

Back-propagation, an abbreviation for "backward propagation of errors", is a common method of artificial neural networks for training different training pattern. From a desired output, the network learns from many inputs, similar to the way a child learns to identify a dog from examples of dogs. It is a supervised learning method, and is a simplification of the rule. It requires a dataset of the desired output for many inputs, making up the training set. It is most useful for feed-forward networks. Back-propagation have need of that the activation function used by the artificial neurons (or "nodes") be differentiable.

A back-propagation algorithm has been used to calculate the gradient of the error of the network with respect to the network's modifiable weights. A simple stochastic gradient is almost always used in descent algorithm to find weights that minimize the error. Back-propagation usually allows quick convergence on satisfactory local minima for error in the kind of networks to which it is suited. The number of hidden nodes depends upon the number of training patterns.

For better understanding, the back-propagation learning algorithm can be divided into two phases: propagation and weight update.

Phase 1: Propagation

It involves the following steps:

1. Forward propagation of a training pattern's input through the neural network in order to generate the propagation's output activations.
2. Backward propagation of the propagation's output activations through the neural network using the training pattern target in order to generate the deltas of all output and hidden neurons.

Phase 2: Weight update

For each weight follow the following steps:

1. Multiply its output delta and input activation to get the gradient of the weight.
2. Bring the weight in the opposite direction of the gradient by subtracting a ratio of it from the weight.

This ratio influences the speed and quality of learning; it is called the learning rate. The sign of the gradient of a weight indicates where the error is increasing; this is why the weight must be updated in the opposite direction.

Repeat phase 1 and 2 until the show the performance of the network is satisfactory.

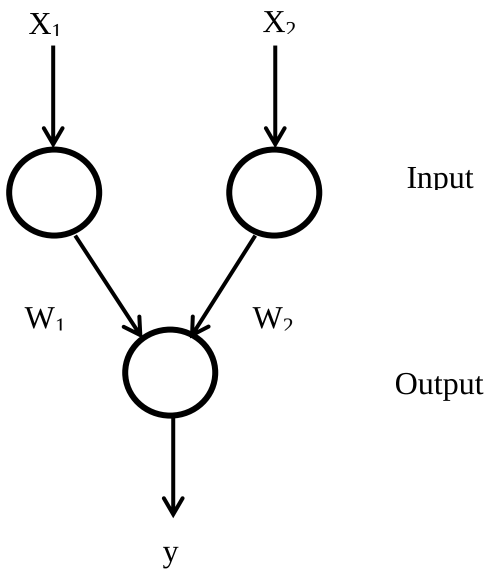


Figure 19- A Neural Network with two input and one output unit

Initially, before training, the weights will be set to random. Then learning of the neuron from training examples, which in this case contains a set of data (x_1, x_2, t) where x_1 and x_2 are the inputs to the network and t is the correct output (the output the network should eventually produce given the identical inputs). The network given x_1 and x_2 will calculate an output y which very likely differs from t (since the weights are at first random). For measuring the discrepancy a common method between the expected output t and the actual output y is using the squared error measure:

$$E = (t - y)^2,$$

Where E is the discrepancy or error.

As an example, consider the network on a single training data: (1, 1, 0), thus the input x_1 and x_2 are 1 and 1 respectively and the correct output,t is 0.

So, the problem of mapping inputs to outputs can be reduced to a problem of finding a function that will produce the minimal error.

Though, the output of a neuron depends on the weighted sum of all its inputs:

$$y = x_1 w_1 + x_2 w_2,$$

Where w_1 and w_2 are the weights on the network from the input units to the output unit. Thus, the error also depends on the incoming weights to the neuron, which is finally what needs to be changed in the network to enable learning.

On application of neural networks in control engineering the key properties of neural networks used for control are:

- Nonlinear dynamics,

- Natural complexity (multiple inputs/outputs and complex internal structure),
- Adaptability and learning ability.

Also in Target tracking is important for vision-based robots to implement tasks of grasping and avoiding obstacles. The purpose of a target tracking system is to detect a target and then to estimate the position of the target. The targets' positions are generally described by various coordinate systems for different purposes. This study focuses on the problem of coordinate transformation on mobile robots and employs the techniques of Back-Propagation Neural Networks to find out the estimate models. With such estimate models, coordinate transformation can be done with less processing time. The techniques have been implemented and integrated with a four-wheeled vision-based security robot and has been verified in real environments.

The main components of the robot used in this study include 4 wheels, one 4-DOF (degree of freedom) arm, and a camera, as shown in Figure. The robot captures images continuously and analyses these snapshots every second. For identifying objects, the snapshots are processed through generic image processing techniques for identifying targets.

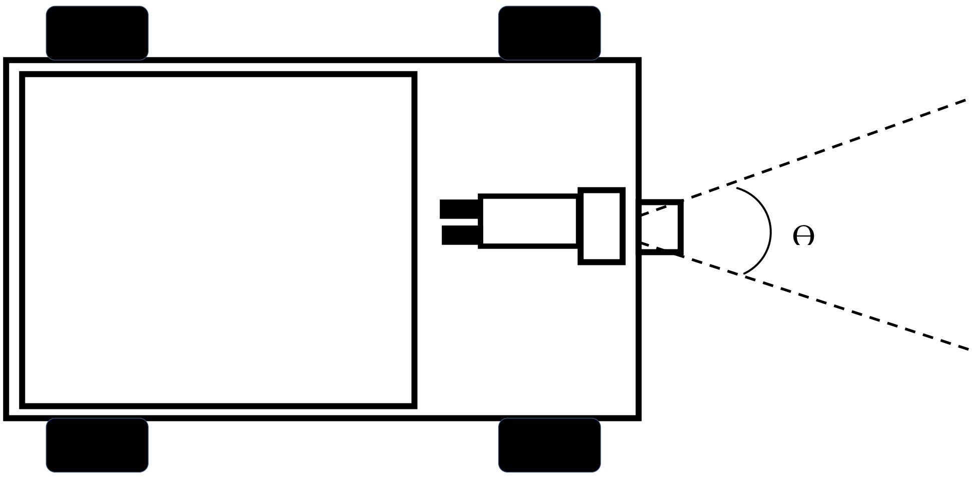
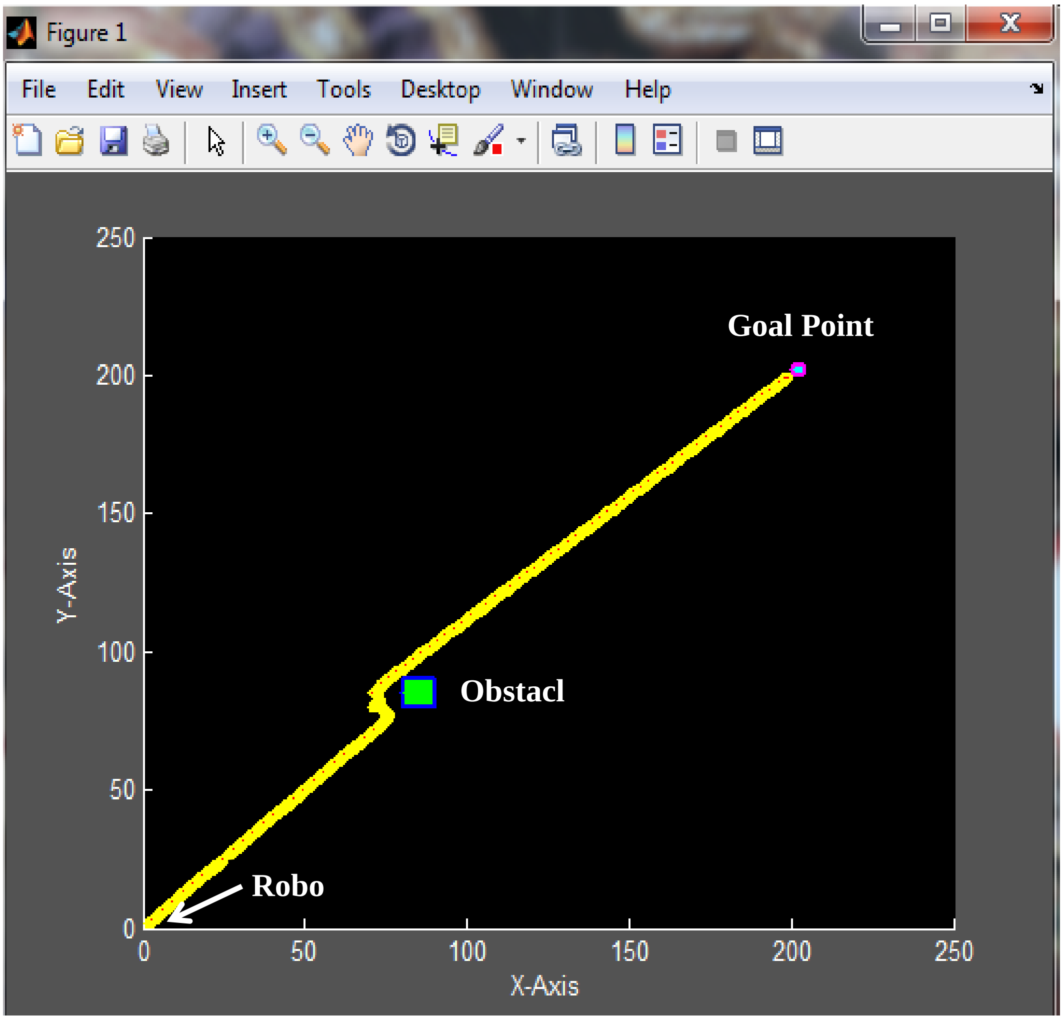


Figure 20- Robot Architecture

The two coordinate systems are applied in different situations. For target visible in the camera, the target is described by the V-frame in above figure through angle Θ. As the robot is required to pick-up all target, even ones that are currently unseen a map is constructed by recording the position of all target. Artificial Neural Network has been used successfully in prediction and classification of image and data.

4.1.5 Simulation Result

The simulation results present the effectiveness of novel approach that evolves neural network controller. The series of simulations test have been conducted with the MATLAB software. To demonstrate the effectiveness and the robustness of the proposed method, simulation results on mobile robot navigation in various environments are exhibited. An important part of robot behavior is avoidance of obstacles. Examples of static obstacles include walls, poles, fences, trees etc. as well as other moving obstacle like vehicles, people, animals etc. The wall following behavior is required to move from one room to another room. The wall following behavior mode will be adopted when the mobile robot detects an obstacle in the front while it is moving towards target, the mobile robot may turn left or right because presence of obstacle in the front. In this case, the robot tries to maintain perpendicular to the wall.



Robot (0, 0)

Head Angle (39°)

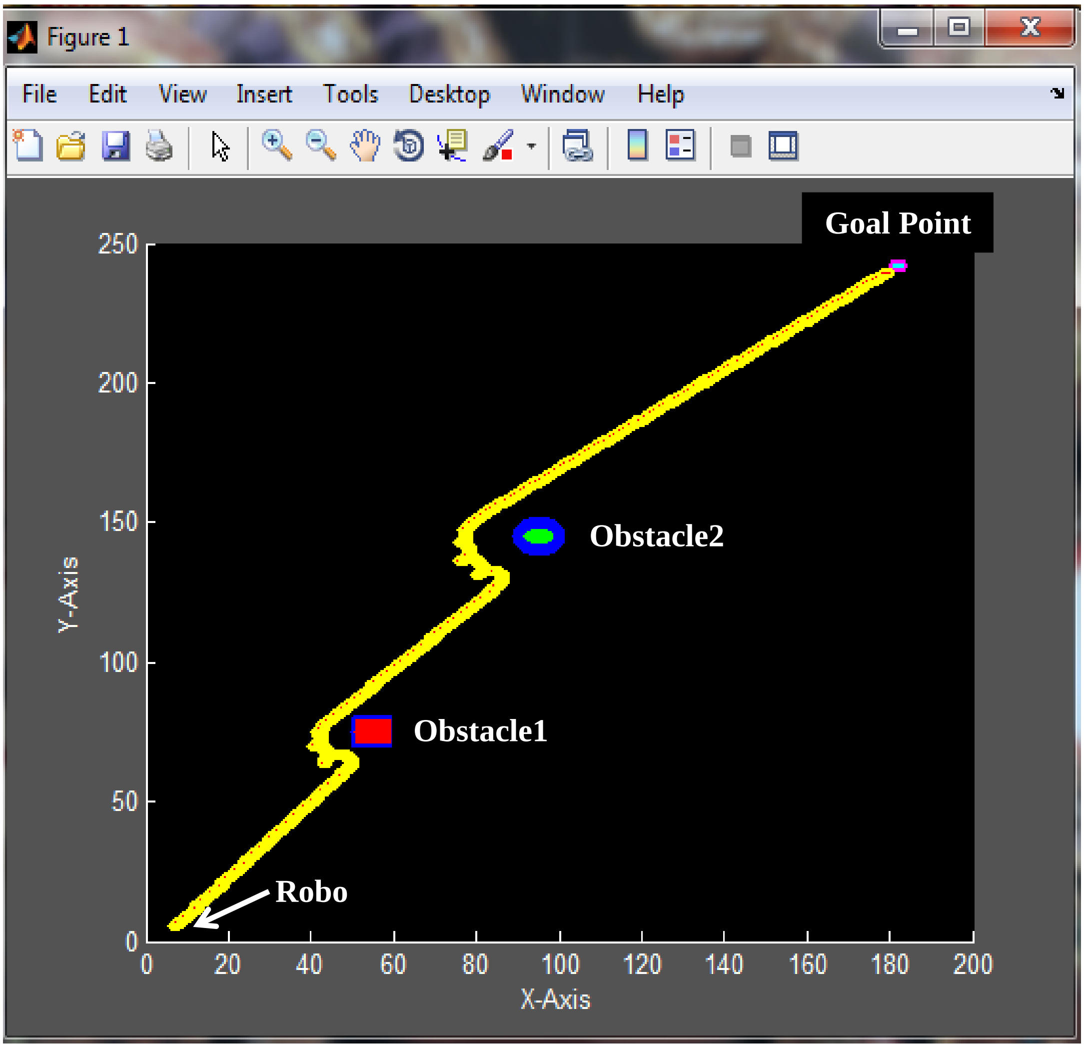
Goal (200, 200)

Obs (80, 80)

Time (34 sec.)

Figure 21- Trajectory of Mobile Robot using one obstacle

Target searching algorithms assume that the goal state is fixed and does not change during navigation of mobile robot. For example, in the problem of moving from the current location to a desired goal location along a network of paths, it is assumed that the target location is fixed and does not change during the navigation. The series of simulations test have been conducted to exhibit that the anticipated method can partially fulfill the most of the fundamental as well as critical robotic behaviors during navigation in complex and uncertain environments. The MATLAB simulation results have to present in figure 21, 22 & 23. In these figure we achieve the target using single obstacle, double obstacle then four obstacles.



Robot (5, 3)
Head Angle (45°)
Goal (180, 240)
Obs1 (50, 70)
Obs2 (90, 140)
Time (38 sec.)

Figure 22- Trajectory of Mobile Robot using two obstacle

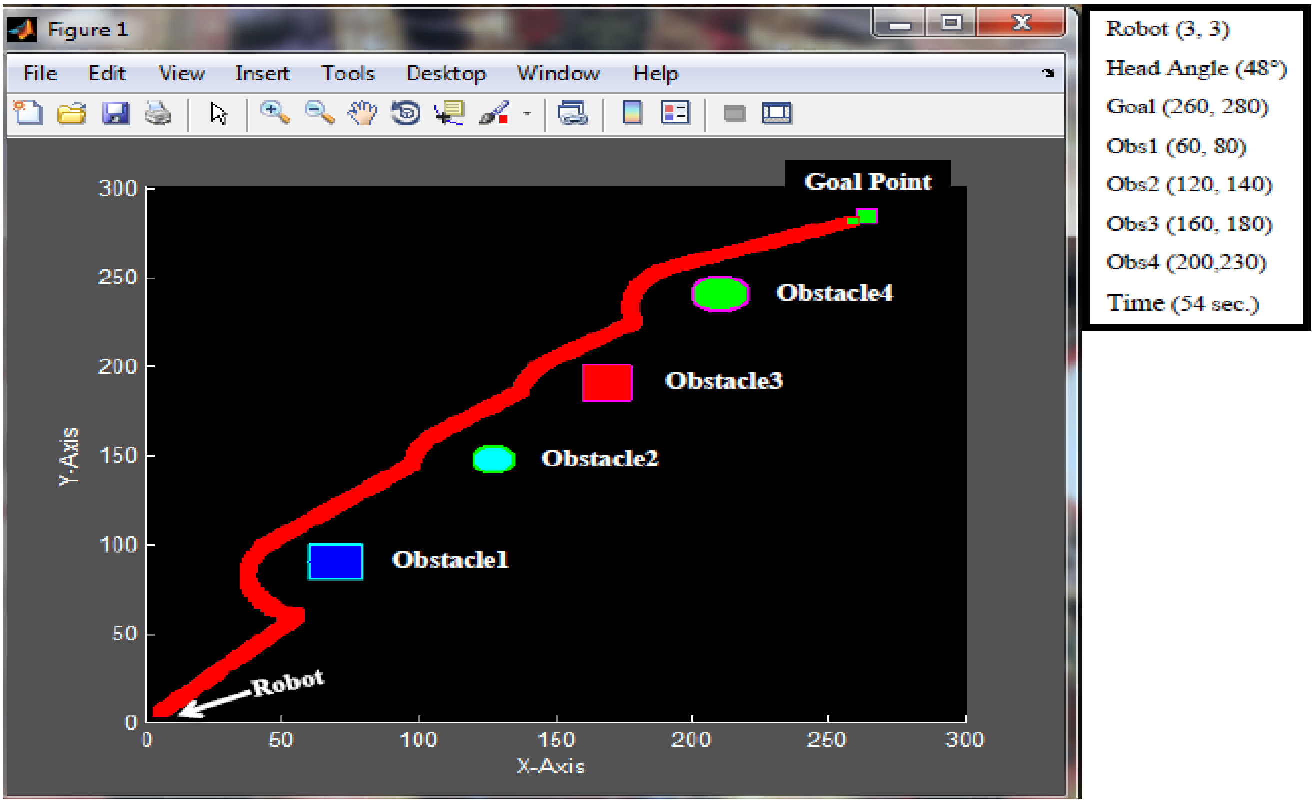


Figure 23- Trajectory of Mobile Robot using 4 obstacles

4.1.6 Application

1. Aerospace: High performance aircraft autopilot, flight path simulation, aircraft control systems, autopilot enhancements, aircraft component simulation, aircraft component fault detection
2. Automotive: Automobile automatic guidance system, warranty activity analysis
3. Banking : Check and other document reading, credit application evaluation
4. Credit Card Activity Checking: Neural networks are used to spot unusual credit card activity that might possibly be associated with loss of a credit card
5. Defense: Weapon steering, target tracking, object discrimination, facial recognition, new kinds of sensors, sonar, radar and image signal processing including data compression, feature extraction and noise suppression, signal/image identification
6. Electronics: Code sequence prediction, integrated circuit chip layout, process control, chip failure analysis, machine vision, voice synthesis, nonlinear modelling
7. Entertainment: Animation, special effects, market forecasting
8. Industrial: Neural networks are being trained to predict the output gasses of furnaces and other industrial processes. They then replace complex and costly equipment used for this purpose in the past.
9. Manufacturing: Manufacturing process control, product design and analysis, process and machine diagnosis, real-time particle identification, visual quality inspection systems, beer testing, welding quality analysis, paper quality prediction, computer-chip quality analysis, analysis of grinding operations, chemical product design

analysis, machine maintenance analysis, project bidding, planning and management, dynamic modeling of chemical process system.

10. Medical: Breast cancer cell analysis, EEG and ECG analysis, prosthesis design, optimization of transplant times, hospital expense reduction, hospital quality improvement, emergency-room test advisement
11. Oil and Gas: Exploration
12. **Robotics**: Trajectory control, forklift robot, manipulator controllers, vision systems, obstacle avoidance
13. Securities: Market analysis, automatic bond rating, stock trading advisory systems
14. Telecommunications: Image and data compression, automated information services, real-time translation of spoken language, customer payment processing systems
15. Transportation: Truck brake diagnosis systems, vehicle scheduling, routing systems

4.2 Webot

4.2.1 Introduction

A professional robot simulator Webot widely used for educational purposes. The project started in 1996, initially settled by Dr. Olivier Michel at the Swiss Federal Institute of Technology (EPFL) in Lausanne, Switzerland. Webots uses the ODE (Open Dynamics Engine) for detecting of collisions and simulating rigid body. The ODE library allows one to accurately simulate physical properties of objects such as velocity, inertia etc.

Webots is professional mobile robot simulation software. It offers a quick prototyping environment that allows the user to create 3D virtual worlds with physics properties such as mass, joints, friction coefficients, etc. Here we can add simple passive objects or active objects called mobile robots. These robots can have different arrangements like wheeled robots, legged robots, or flying robots etc. Furthermore, they may be equipped with a number of sensor and actuator devices, such as distance sensors, ultrasonic sensor, drive wheels, cameras, servos, touch sensors, receivers, etc. Finally, the user can program each robot individually to show the desired behavior. Webots also contains a number of boundaries to real mobile robots, so that simulated robot behaves as expected, transfer its control program to a real robot like Corobot, e-puck, Nao, etc. Adding new interfaces is possible through the related system.

A development environment of webot is used to model, program and simulate mobile robots. Webots are design complex robotics scenarios, including possibly several different robots, interacting in a common environment. The goods of each object, such as shape, color, texture, mass or friction, are individually adjustable. A large choice of simulated sensors and actuators is available to prepare each robot. The robot controllers can be programmed with the built-in IDE or any third party development environment. The behavior of the robots is run in physically worlds. The controller programs can optionally be transferred to existing real robots.

What can I do with Webots?

Webots is well suited for research and educational related work to mobile robotics. Many mobile robotics have relied on Webots for years in the following areas:

- Mobile robot prototyping (automotive industry, aeronautics, the vacuum cleaner industry, the toy industry, hobbyists, etc.)
- Robot locomotion (legged, humanoids, quadruped’s robots, etc.)
- Multi-agent research (swarm intelligence, collaborative mobile robots groups, etc.)
- Adaptive behavior research (genetic algorithm, neural networks, AI, etc.)

- Teaching robotics (robotics lectures, C/C++/Java/Python programming lectures, etc.)

Webots is physics-based general-purpose mobile robotics simulation software. The user can choose from a library of robot models and modify them or construct own models. Robot can be equipped with a large number of available sensors and actuators. For each object, a number of properties can be defined, such as shape, color, texture, mass, friction, etc. The user can then program the robots using an arbitrary development environment, simulate them, and optionally transfer the resulting programs onto a real robot.

4.2.2 Overview

Webots allows you to perform 4 basic stages in the development of a robotic project as depicted on the figure. The first stage is the modelling stage. It consists in designing the physical body of the robots, as well as their sensors and actuators and also the physical model of the environment of the robots. It is a bit like a virtual Corobot set where you can assemble building wheels, base and configure them by changing their properties (color, shape, technical properties of sensors and actuators, etc.).

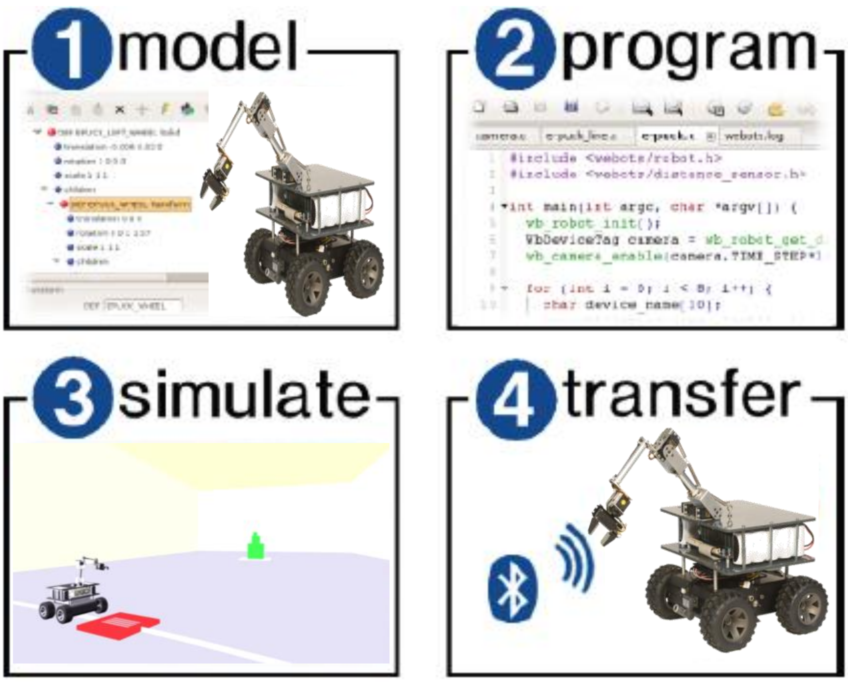


Figure 24- Webots development stages

This approach, any kind of robot can be generated, including wheeled robots, four legged robots, humanoid robots, swimming and flying robots etc. The environment of the robots is formed the same way, by populating the space with objects like walls, wheels, roof, steps, balls, obstacles, etc. The object physical parameters can be defined, like the mass distribution, the bounding objects, the friction, the bounce parameters, etc. so that the simulation in Webots can simulate their dynamics. The figure with the simulation illustrates the model of a Corobot exploring an environment. Once the virtual robots and virtual environment are created, then we can move on to the second stage.

The second stage is the programming stage. In order to achieve this, different programming tools are available. They include graphical programming tools which are easy to use for beginners and programming languages (like C, C++ etc.) which are more powerful and enable the development of more complex behaviors. The program controlling a robot is generally an endless loop which is divided into three parts: (1) read the values measured by the sensors of the robot, (2) compute what should be the next action(s) of the robot and (3) send actuators commands to perform these actions. The easiest parts are parts (1) and (3). The most difficult one is part (2) as this is here that lie all the Artificial Intelligence. Part (2) can be divided into sub-parts such as sensor data processing, learning, motor pattern generation, etc.

The third stage is the simulation stage. It allows to test the program behaves correctly or not. By running the simulation, we will see robot executing program. Then user will be able to play interactively with robot, by moving obstacles using the mouse, moving the robot itself, etc. User will also be able to visualize the values measured by the sensors, the results of the processing of program, etc. It is likely to return several times to the second stage to fix or improve the program and test it again in the simulation stage.

Finally, the fourth stage is the transfer to a real robot. Control program will be transferred into the real robot running in the real world. Then we see if control program behaves the same as in simulation. If the simulation model of robot was performed carefully and was calibrated against its real corresponding item, the real robot should behave roughly the same as the simulated robot. If the real robot doesn't work the same, then it is necessary to come back to the first stage and refine the model of the robot, so that the simulated robot will act like the real one. In this case, we have to go through the second and third stages again, but commonly for some slight tuning, rather than redesigning the program. The figure with two windows shows the Corbot control window allowing the transfer from the simulation to the real robot.

4.2.3 Webot User Interface

Sensor: - A large choice of sensors can be plugged into your robot model:

- Distance sensors (IR, US and laser)
- Cameras (1 D, 2D, spherical)
- Range finders
- Light sensors
- Touch (pressure or bumper) sensors
- Global Positioning Sensors (GPS)
- Receivers (inter-robot communication)
- Position and force sensors for servos
- Incremental wheel encoders
- Inertial Units (3D),
- Accelerometers (3D)
- Gyroscopes (3D)
- Digital compasses (3D)

Sensor parameters may be tuned individually: range, noise, response, field of view, etc.

Actuator: - Similarly, a number of actuators can be added to robot model:

- Differential wheel motor units
- Servo motors: arms, wheels, etc.
- Linear motors (pistons)
- LEDs
- Emitters (inter-robot communication)
- Grippers
- Displays (LCD)
- Pens (drawing)

User Interaction: - The graphical user interface of Webots allows to easily interacting with the robots and their environment. Robots and other objects can be translated, rotated and resized with the mouse, very intuitively and while the simulation is running. Similarly it is possible to apply forces and torques interactively to test the robustness of a mechanical system.

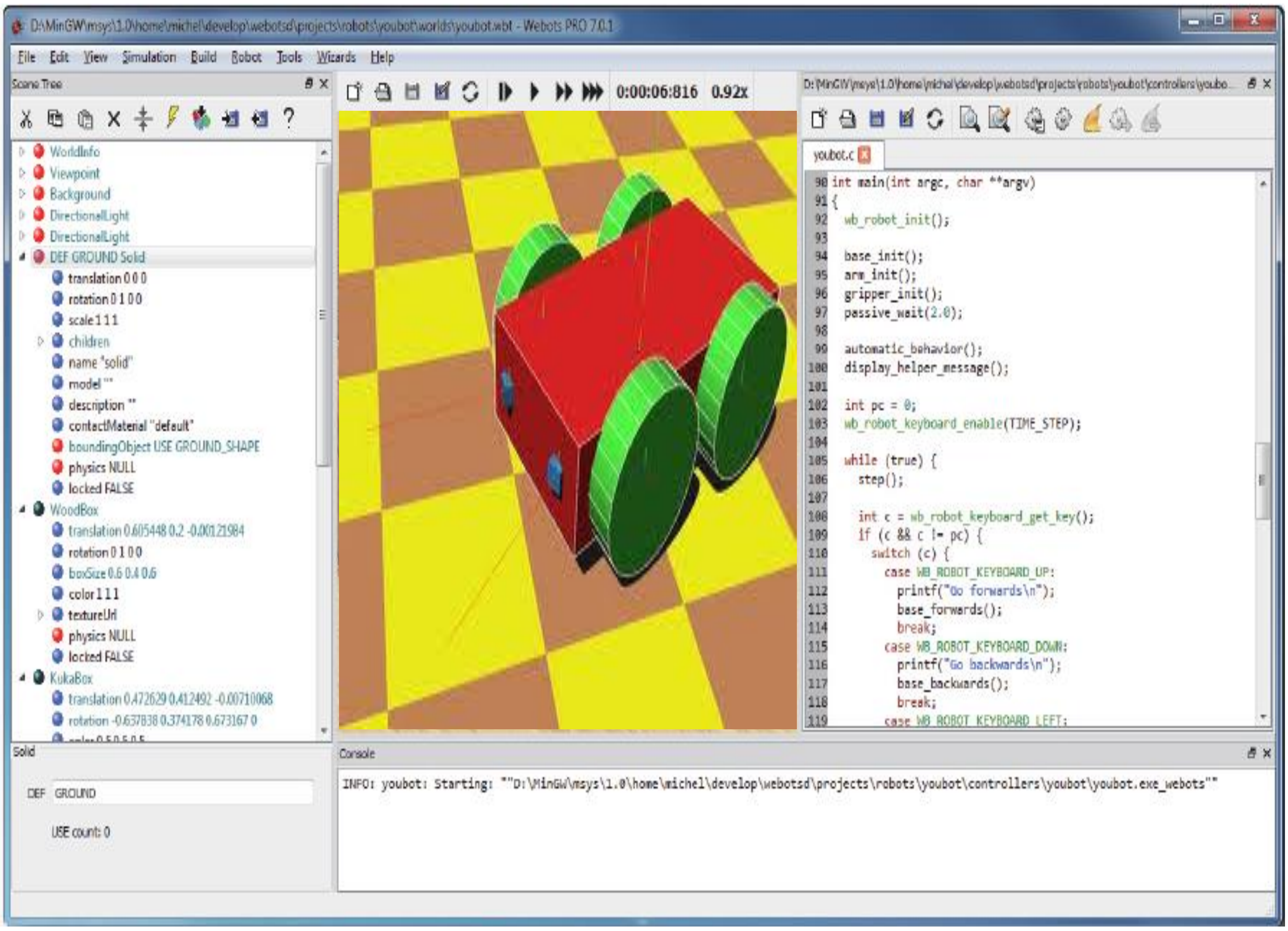


Figure 25- Webot User Interface

Robot Window: - The built-in robot windows continuously display sensor and actuator information so that the status of each device can be monitored while the simulation is running. Actuators may also be controlled from the same robot window. Such windows can be extended by the user to design custom graphical user interfaces, using the library.

Programming Interface: - Programming your robot using the C language is as simple as this:

```
#include <webots/robot.h>
#include <webots/differential_wheels.h>
#include <webots/distance_sensor.h>

int main() {
  wb_robot_init(); // initialization
  // make distance measurements each 32 ms
  WbDeviceTag ir = wb_robot_get_device("ps0");
  wb_distance_sensor_enable(ir,32);
  // perform 32 ms simulation steps
  while(wb_robot_step(32)!=-1) {
    if (wb_distance_sensor_get_value(ir) > 100)
      wb_differential_wheels_set_speed(0,0);
    else
      wb_differential_wheels_set_speed(10,10);
  }
}
```

4.2.4 Webot Simulation

A Webots simulation is composed a Webots *world* files (.wbt) that defines one or several robots and their environment. The .wbt file does sometime depend on external prototypes files (.proto) and textures. And other one or several controller programs for the above robots (in C/C++/Java/Python/Matlab).

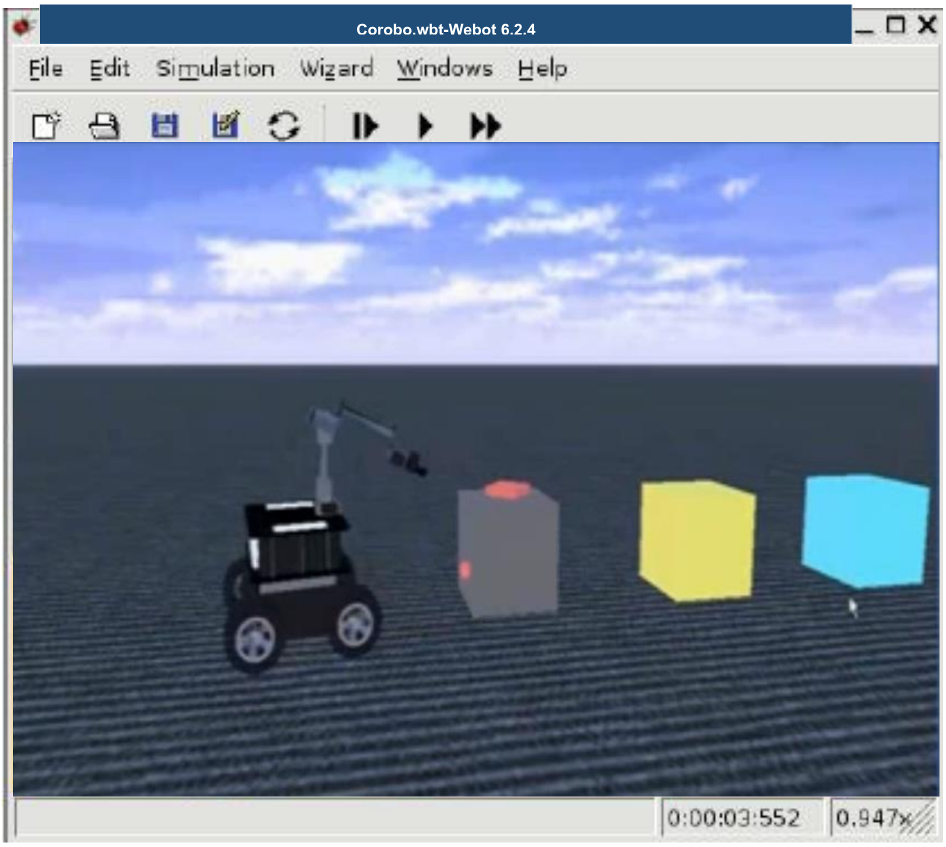


Figure 26- Simulation of a Corobot in Webots

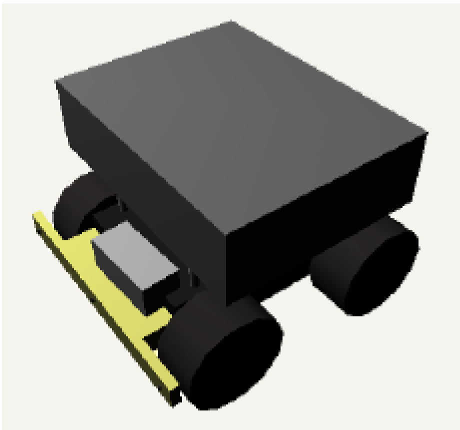



Figure 27- Corobot and Corobot Model in Webot

Research experiments often need to interact automatically with the simulation. The capability allows writing a program responsible for supervising an experiment. For example, a program can move objects, change their properties, send messages to robots, record robot trajectories, take a snapshot or record a video of the simulation. The capability can be used for optimization algorithms where a large number of robot configurations or control parameters have to be evaluated, as in genetic programming, neural networks and machine learning, etc.

4.2.5 Application

Webot has been used for many multiple robotic application projects in these fields:

- Mobile robot design and prototyping
- Fast prototyping of wheeled and legged robots
- Research on robot locomotion
- Swarm intelligence (Multi-robot simulations)
- Artificial life and evolutionary robotics
- Simulation of adaptive behavior
- Self-Reconfiguring Modular Robotics
- Experimental environment for computer vision
- Teaching and robot programming contests
- Machine vision
- Surgical robotics

4.3 Corobot

4.3.1 What is Corobot

Corobot is Coroware based company of Kirkland WA. The corobot and explorer both is mobile robot, which are four wheeled robotic expansion platforms.

“The corobot is a 4 wheeled robot with an on-board PC. It has wheel encoders, a webcam, IR range sensors and an optional 4 degree of freedom arm with gripper. It is a standard platform robot which is meant to be customized for research application.”

Corobot was generated to minimize the complexity of robot development. Through combining a powerful PC-class platform with a robust, object-oriented development system, the Corobot empowers to rapidly deploy and develop robotics results. The corobot also supports the hardware developer with additional physical mounting space, ports, sensors and communication devices.

4.3.2 Overview & Initial Setup

Carrying the Corobot:-

The Corobot arrives fully assembled and after a few initial setup steps, the Corobot is ready for use. To hold the Corobot securely with two hands placed between the wheels holding the upper and lower decks.

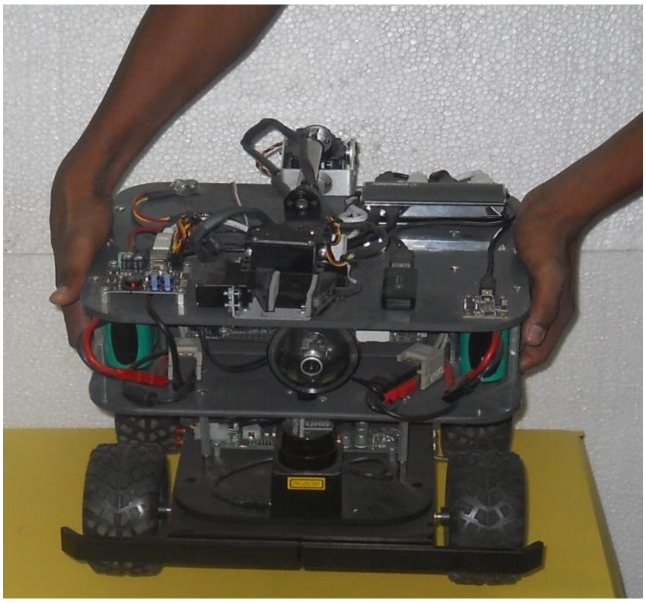


Figure 28- Carrying the Corobot

Environmental Concerns:-

The Corobot is designed for easy entrance, disassembly, and reassembly for the adding and changing of hardware and parts. Due to the uncluttered design, the Corobot is sensitive to debris. Light outdoor operation should pose no problems to the Corobot; on the other hand, water and dirt will cause damage. The unit is not water proof. Most indoor environment poses no problems for the Corobot.

Front of Corobot:-

The front of the Corobot has the primary sensor (a laser range finder, pan-tilt camera, and/or fixed camera) and the robotic arm.

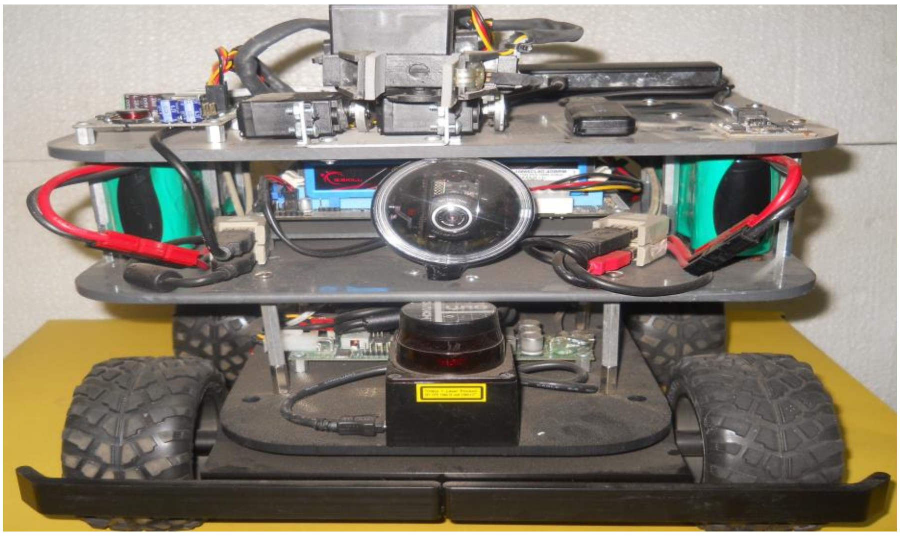


Figure 29- Front View of Corobot

Bumper Sensors:-

The Corobot comes standard with a front bumper sensor. If selected a 4-wheel drive Corobot, then have received a rear bumper. This sensor detects positive pressure collisions with obstacles using two feelers on each bumper.

Figure 30- View of Bumper Sensor

Sides of Corobot:-

The Corobot has a power switch and jacks for battery charging and tether on one side, and push buttons for turning on and resetting the on-board CPU on the other. The 2 wheel, differential drive base has the power switch and jacks on the left side of the robot and the CPU power and reset buttons on the right. The 4 wheel, skid steer base has the power switch and jacks on the right side of the robot and the CPU power and reset buttons on the left.

Figure 31- Left Side of Corobot View

For both cases, the power Selector Switch is the three position switch in the centre.

- Up: Operate off tethered power and enable battery charging
- Centre: Off and disable battery charging
- Bottom: Run off battery power (no charge)

Figure 32- Right Side of Corobot View

The other side of the Corobot contains two buttons.

1. Power button – Press this button to turn on the robot's on-board CPU. Once on, pressing the button once will signal the OS to start shutting down cleanly. Pressing the button continuously for 5 seconds will power the CPU immediately off.
2. Reset button – Press this button to reset the robot's on-board CPU.

Rear of Corobot:-

2 wheel differential drive Corobots have a caster at the rear of the robot for balance. The 4 wheel skid steer base is steadied by all four wheels, and so does not need a caster for balance. Rear connection ports are located on the upper deck in between the batteries.

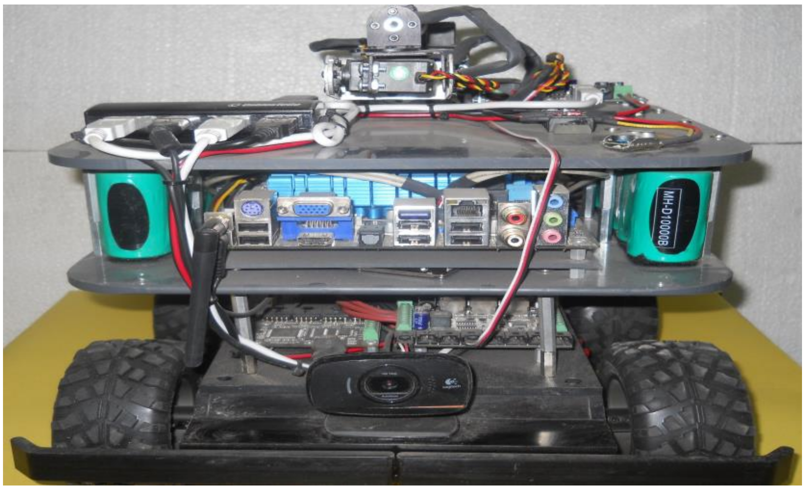


Figure 33 – Rear View of Corobot

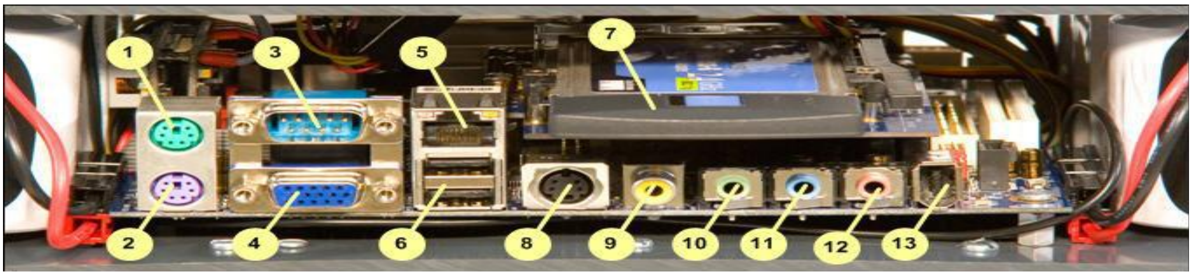


Figure 34- Rear Port Connection

Rear Connection Ports

1. Mouse (green)
2. Keyboard (purple)
3. Serial connection (top)
4. Monitor connection (bottom)
5. Ethernet jack (top)
6. USB board (bottom)
7. PCMCIA slot (Wireless Card Inserted)
8. S-video output (black)
9. RCA-video output (yellow)
10. Line out (green)
11. Line in (blue)
12. Microphone (pink)
13. Firewire

Robotic Arm:-

The Corobot can be purchased with or without the robotic arm. The Corobot arm arrives fully assembled.

Figure 35- Arm View

Overview of Arm-

- 4 DOF
- Touch sensor gripper
- Parallel jaw gripper
- Gripper opens 1.3"
- Lift capacity is 8 ounces

Applications of Robotic Arms-

- Agriculture
- Defence
- Material Handling
- Medical
- Other Industrial Applications
- Welding

Description

A robotic arm is a robotic manipulator, usually with similar functions to a human arm. Servo motor is used for joint motion & rotation. It has about same number of degree of freedom as in human arm. In order for a robot or a robotic arm to pick up or move something, someone has to tell it to perform several actions in a particular order from moving the arm, to rotating the wrist to opening and closing the gripper. So, we can control each joint through computer interface.

What are Servo Motors?

Servo brings to an error sensing feedback control which is used to precise the performance of a system. Servo or Servo Motors are DC motors equipped with a servo mechanism for precise control of angular position. The servo motors usually have a rotation limit from 90° to 180°. Their rotation is restricted in between the fixed angles.

Where are Servos used?

For precision positioning the servo are used. They are used in robotic arms and legs, sensor scanners and in toys like helicopter, airplanes and cars.

Servo Control-

A servo motor mainly consists of a DC motor, gear system, a sensor which is mostly a potentiometer, and control electronics system. The DC motors is connected with a gear mechanism which provides feedback to a position sensor which is mostly a potentiometer. From the gear box, the motor output is delivered via servo spline to the servo arm.

Phidget-

Phidgets are a system of low-cost electronic components and sensors that are controlled by a personal computer. Using the universal Serial Bus (USB) as the basis for all pidgets, the complexity is managed behind an Application programming Interface (API). Their usage is primarily focused to allow exploration of different physical computer interaction systems, but have most especially been adopted by robotic supporters as they greatly simplify PC-Robot interaction systems, but have most notably been adopted by robotic enthusiasts as they greatly simplify PC-Robot interaction.

Example of Phidgets-

Servo – Allows control of up to 4 servo motors. Each servo can be addressed individually where it can have its position read and set.

Phidget Accelerometer – The accelerometer senses acceleration in 2 and 3 dimensions.

Interface Kit – Allows input/output interface to analog and digital sensors and switches.

4.3.3 Corobot Specification

Table-4

Dimensions	**12 x 13 x 10 in (16in w/ arm)**
CPU	**1.5 Ghz**
RAM	**1 GB**
Disk Space/ROM	**80 GB**
Battery/Life	**10 AH / 2.5 Hours**
Base Type/Seteering	**4 WD skid steer**
Max. Speed	**1.5 ft. per second**
Wheel Encoders	**Yes**
Camera	**640x480 Color , 2 Megapixel**
Digital Inputs	**8**
Digital Outputs	**8**
Bumper Sensors	**Front (standard) and Back (optional)**
Voltage Sensor	**Yes**
Arm Size	**14 in long**
Arm DOF	**4**
Gripper Span	**1.3 in**
Gripper Sensor	**Yes**
Arm Payload Capacity	**8 oz**
Base Payload Capacity	**5 lbs**
Windows	**XP, Supporting C-Language API**
Linux	**Ubuntu, supporting C-language API and Player**
Optional Pan/Tilt Camera	**Yes**
Optional AMD Processor	**Yes**

4.3.4 Corobot Installation

Corobot Windows Installation:-

When powering up for first use, the user will be prompted to complete the last few steps to install windows. The default windows installation is recommended. In this Corobot a standard PC located on the lower deck. During the installation process, the user will be required to attach the monitor, keyboard, mouse, and feasibly the Ethernet connection.

Complete the following steps to install Windows:

1. Ensure that the Corobot is elevated on blocks such that the wheels do not come into contact with the ground surface.
2. Plug the tethered power card into the electrical outlet and then into the Corobot.
3. Attach the battery charger unit to the Corobot.
4. Attach the monitor, keyboard, mouse and Ethernet connection.
5. Press the power button and wait for the Corbot to fully power up.

6. The Welcome to Microsoft Windows screen will be displayed.
7. Click Next.
8. Read End User License Agreement then accepts the agreement, click Next.
9. Enter the Windows Product Key.
10. At the Help protect PC screen, we will not enable the firewall right now. Click Next when done.
11. Choose a computer name for the CoroBot's on-board computer. Click Next when
12. Select an Administrator password. Enter the password and click Next.
13. Set user accounts. These accounts will not be used by the robot, however, Microsoft Windows requires at least one account to be created. Click Next. The Welcome to Microsoft Windows screen will be displayed.
14. Windows will complete the boot-up process. At the log-on screen, select the "robot" account.
15. Click the start button, and select "Run". In the dialog box enter "control userpasswords2"
16. A dialog box prompting for the account automatically logging then Click the OK button.
17. Shut down Windows.
18. When the computer has powered down, press the Corobot power button again to reboot the computer.
19. Verify that the Corobot automatically logs into the "robot" account, and runs its services.
20. Shut down Windows
21. Disconnect the keyboard, monitor, mouse and Ethernet

4.3.5 Experiment Result

To perform the experimental test of mobile robot the developed software by Phidget Control panel is loaded into the Corobot. To control the functionalities of the robot (motors, sensors etc.), a set of command are implemented in the control protocol.

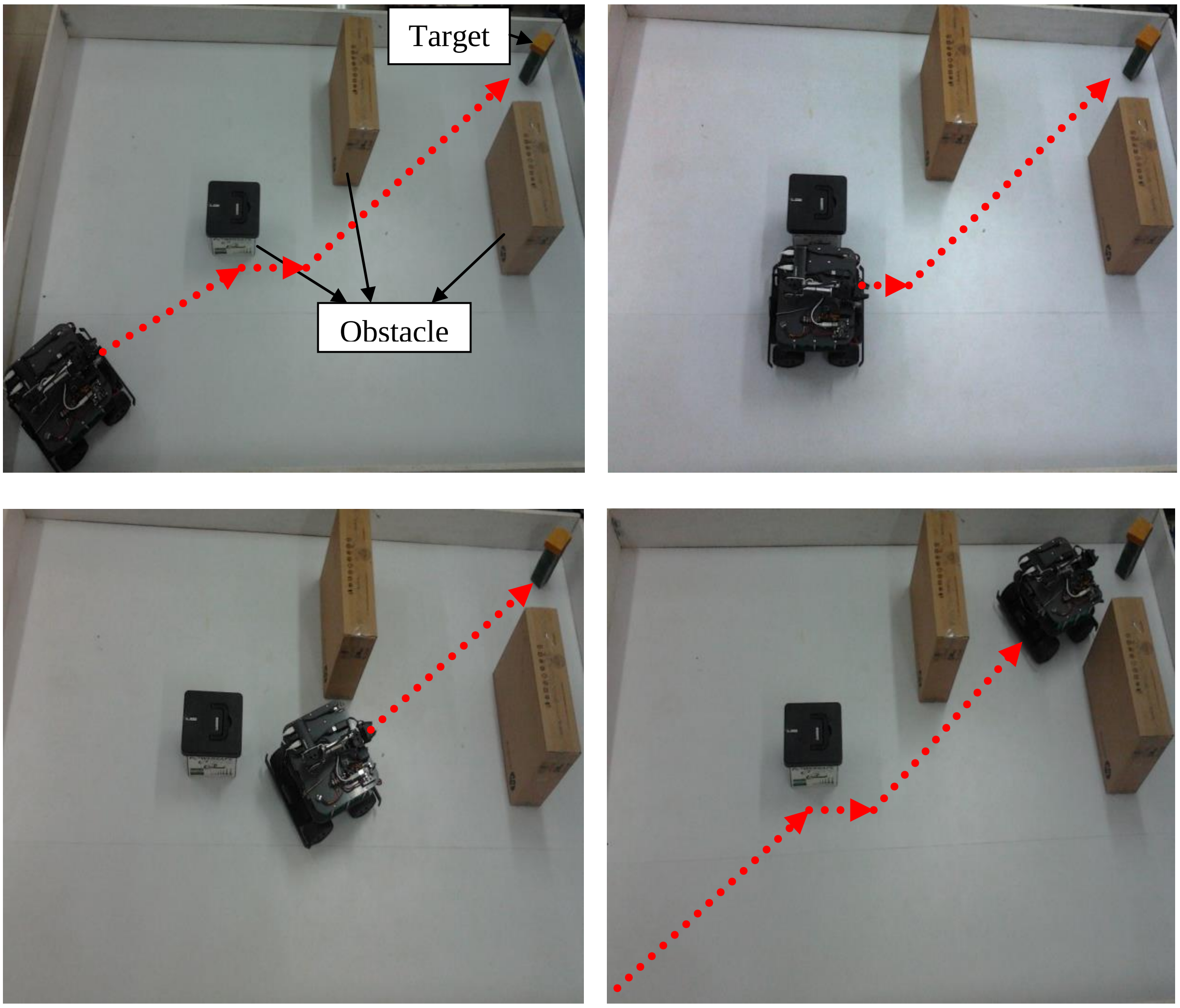


Figure 36- Experiment Setup of Corobot with avoiding obstacle to seeking the target

The assumptions about the mechanical structure and the motion of a mobile robot to which our proposed method is applied as mobile robot moves on lab specified floor area and the wheel of a mobile robot rolls on the floor without any translational slip. For this wheel two motor are used one are in left side and other are right side of Corobot. And wheel is moves in both forward and reverse direction at specified given velocity and acceleration. This Corobot achieves the target in around 20 sec to 5-8 m distance.

CHAPTER 5
DISCUSSION OF RESULT

This chapter discusses the development of autonomous navigation and obstacle avoidance systems for differential drive mobile robots operating in real environments. In this chapter the performance of developed intelligent techniques are summarized and their results are outlined.

5.1 Discussion of Result

The project can be implemented in different area like kinematic analysis of robotic arm, run the wheel encoders, etc. and neural network, webot simulation of Corobot. The first part of the result was done by used the servos to move the robot arm. To interface a robot arm with a computer and set into operation and to control the motion of robot arm by computer keyboard input. This goal was achieved by Phidgets and Servo-Motor Controller. The arm parts operate to move the shoulders, elbow, wrist and gripper joint of the robot with 4-DOF in either up, down or to be rotate.

The kinematic design is a basic part of the mobile robot system. Improved mechanical designs and mobility control systems will enable the mobile robot to navigate in no marked paths and for autonomous operation. A kinematic methodology is the first step towards achieving these goals.

In addition to avoiding static obstacles, path planning we must include under neural network method radial basis function and back-propagation algorithm. Also we discuss the result of Corobot with webot simulation and their experimental data. To illustrate this, we consider the case in which the robot can reach the target in this way. At the beginning of the simulation robot moves along a substantial path and cause a rapid change of neural activities and the direction.

In this study, we proposed the BP learning for the development of autonomous robots with proximal and distal evaluation scores of behavior, and described results of its application to a real mobile robot Corobot. The training data set was obtained in free movement in an environment, and selected in accordance with an evaluation function which represented the consequence of behavior in the near and distant future. Corobot learned to navigate in an environment with obstacles faster than that with one evaluation score, and also faster than the conventional evolution.

The results of the proposed trajectory planner are discussed through simulations on Corobot using Webots robotics simulator. The Webots simulator is based on ODE (open dynamic engine), open source physics. The presented results prove that the proposed method can produce accurate collision free paths by using simple and cheap sensors. Furthermore, the robustness and the effectiveness of the method are established. We observe that both the simulator and the real robot achieve the same final position and that, moreover, the trajectory described by both, from the initial position to the final position, is practically identical.

Table5: Comparison Simulation Result Vs. Experimental

S. No	Path Length of Robot during Simulation using NN (m)	Path Length of Robot during Experiment using NN (m)	Time Taken during Simulation (Sec)	Time Taken during Experiment (Sec)	Deviation (%)	
					Length	Time
1	4.5	5	17	19	10.00	10.52
2	6.4	7	22.5	25	09.28	10.00
3	8.2	9	28.2	31	08.89	09.03
4	10	11	34.5	38	09.09	09.21

Deviation (%) = [(Experimental – Simulation)/ Experimental] × 100

CHAPTER 6
CONCLUSION & FUTURE WORK

This chapter summarizes the conclusions of the research and proposes idea for future work.

6.1 Conclusion

In this thesis report, from the literature review we conclude that present a new mobile robot navigation strategy based on the WMR with unknown environment towards the proposed controller drives robot ultimately target at the required distance with the given speed. The aim was to obtain robust robot control suitable for real-time requirements and a laser scanner on a movable part actuated by a motor for navigation of mobile robots. Obstacle avoidance and gateway detection can be implemented using proper navigation strategies. The method suggested can be applied to the robot manipulators with a mobile obstacle obstructing the motion of the mobile robot, show the effectiveness of the proposed approach.

It is also observed that in order to navigate between two known locations seem to prefer well defined and constant paths, even if this means longer travel distances. A method of detection and tracking a specific object based on colour camera has been implemented use of combining colour features and shape information of objects. This system points out several different modelling services, and enhances a lot the robot autonomy and efficiency. Finally, we must improve the robot speed by enhancements on the global system performance. To solve the problems of landmark tracking and understanding a techniques of image processing and pattern recognition are integrated. Some robotic applications require a wide spread coverage algorithm to agreement that the robot’s path covers the whole obstacle-free part. Validation of theoretical and work has been done by simulation and a real world tests by a COROBOT simulation has been developed.

6.2 Future Work

There are a number of interesting directions to pursue as future work. In future the Corobot comes with a C-language API. This C-API code allows to runs the Corobot autonomously. And recommendation, this robot can be equipped with a view and display at the monitor screen by achieves a target by both local & global navigation. And further development of the techniques may be required for the avoidance of moving obstacles other than the robots. In these navigational techniques may be carried out so that the robots can not only detect dynamic targets but also reach them using an optimum path. There is in future good indication that uses local landmarks such as projecting trees, rocks, forest edges, etc., as well as situation landmarks such as sun, stars and magnetic senses to navigate effectively.

References

[1] S. Matveeva Alexey, Hamid Teimoori, Andrey V. Savkin, Range-only measurements based target following for wheeled mobile robot, Automatica 47 (2011) 177–184.

[2] Omid Mohareri, Rached Dhaouadi, Ahmad B.Rad, Indirect adaptive tracking control of a Non-holonomic mobile robot via neural networks, Neurocomputing 88 (2012) 54–66

[3] Atsushi Kanda, Masanori Sato, Kazuo Ishii, Environment Recognition system Based on Multiple Classification Analyses for Mobile Robots, Journal of Bionic Engineering Suppl. (2008) 113–120

[4] Nazri Abu Bakar, Abdul RahmanMohd. Saad, A Monocular Vision-based Specific Person Detection System for Mobile Robot Applications, International Symposium on Robotics and Intelligent Sensors 2012 (IRIS 2012), Procedia Engineering 41(2012)22–31

[5] Ali SafakSekmen, Mitch Wilkes, Susan R. Goldman, Saleh Zein-Sabatto, Exploring importance of location and prior knowledge of environment on mobile robot control ,Int. J. Human-Computer Studies 58 (2003) 5–20

[6] Sebastian Thrun, Learning metric-topological maps for indoor mobile robot navigation, Artificial Intelligence 99 (1998) 21-71

[7] B B V L Deepak, Dayal R. Parhi, Shubhasri Kundu, Innate immune based path planner of an Autonomous Mobile Robot , Nit Rourkela Mechanical Engineering (2011)

[8] Magnus Egerstedt , Xiaoming Hu , A hybrid control approach to action coordination for mobile robots, Automatica 38 (2002) 125-130

[9] Ming Lei, Guan Zailin & Yang Shuzi, Mobile robot fuzzy control optimization using genetic algorithm, Artificial Intelligence in Engineering 10 (1996) 293-298

[10] Kerstin Dautenhahn, Getting to know each other artificial social intelligence for autonomous robots, Robotics and Autonomous Systems 16 (1995) 333-356

[11] Andrzej Kasin ski, Piotr Skrzypczynski, Perception network for the team of indoor mobile robots: concept, architecture, implementation, Engineering Applications of Artificial Intelligence 14 (2001) 125–137

[12] Ulrich Nehmzow, Animal and robot navigation, Robotics and Autonomous Systems, 15 (1995) 71-81

[13] Andrew Howard, Les Kitchen, Navigation using natural landmarks, Robotics and Autonomous Systems 26 (1999) 99-115

[14] Danny Ratner, Phillip McKerrow, Navigating an outdoor robot along continuous landmarks with ultrasonic sensing, Robotics and Autonomous Systems 45 (2003) 73–82

[15] Guoqiang Fua, Arianna Menciassi, Paolo Dario, Development of a low-cost active 3D triangulation laser scanner for indoor navigation of miniature mobile robots, Robotics and Autonomous Systems 60 (2012) 1317–1326

[16] Li Wenfeng, Weiming Shen, Swarm behavior control of mobile multi-robot switch wireless sensor networks, Journal of Network and Computer Applications 34 (2011) 1398–1407

[17] A.A. NippunKumaar, Sudarshan TSB, Sensor Counter Approach for a Mobile Robot to Navigate a Path Using Programming by Demonstration, Procedia Engineering 30 (2012) 554 – 561

[18] Sebastian Thrun, An approach to learning mobile robot navigation, Robotics and Autonomous Systems 15 (1995) 301-319

[19] P. Weckesser, R. Dillmann, Modeling unknown environments with a mobile robot, Robotics and Autonomous Systems 23 (1998) 293-300

[20] Risto Miikkulainen, Jean-Arcady Meyera, David Filliatb, Map-based navigation in mobile robots: II. A review of map-learning and path-planning strategies, Cognitive Systems Research 4 (2003) 243–282

[21] M.A. Porta Garcia, Oscar Montiel, Oscar Castillo, Roberto Sepulveda,Patricia Melin, Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation , Applied Soft Computing 9 (2009) 1102–1110

[22] Stephen Marsland, Ulrich Nehmzow, Tom Duckett , Learning to select distinctive Land marks for mobile robot navigation , Robotics and Autonomous Systems 37 (2001) 241–260

[23] Boubaker Daachi, Tarek Madani , Abdelaziz Benallegue , Adaptive neural controller for redundant robot manipulators and collision avoidance with mobile obstacles, Neuro-computing 79 (2012) 50–60

[24] R. Abiyev, D. Ibrahim, B. Erin, Navigation of mobile robots in the presence of obstacles, Advances in Engineering Software 41 (2010) 1179–1186

[25] Jiann-Der Lee, Indoor Robot Navigation by Landmark Tracking, Math. Computer Modelling Vol. 26, No. 4, pp. 79-89, 1997

[26] I. Noorani Rafigul, Microcontroller based robot arm control, Mathl computer Modeling, Vol.14, pp. 450-455, 1990

[27] K. Paik Jamie, Bu Hyun Shin, Young-bong Bang, Young-Bo Shim, Development of an Anthropomorphic Robotic Arm and Hand for Interactive Humanoids, Journal of Bionic Engineering 9 (2012) 133-142

[28] Wan Muhamad, Hanif Wan Kadir, Reza Ezuan Samin, Babul Salam Kader Ibrahim, Internet Controlled Robotic Arm, Procedia Engineering 41 (2012) 1065 – 1071

[29] Mohd Ashiq Kamaril Yusoff, Reza Ezuan Samin, Babul Salam Kadar Ibrahim, Wireless Mobile Robotic Arm, Procedia Engineering 41 (2012) 1072 – 1078

[30] Hiroaki Yamaguchi, A distributed motion co-ordination strategy for multiple Non-holonomic mobile robots in cooperative hunting operations, Robotics and Autonomous Systems 43 (2002) 257-282

[31] Michael Defoort, Annemarie Kokosy, Thierry Floquet, Wilfrid Perruquetti, Jorge Palos,Motion planning for cooperative unicycle-type mobile robots with limited sensing ranges: A distributed receding horizon approach, Robotics and Autonomous Systems 57 (2009) 1094-1106

[32] Daniel Meyer-Delius, Wolfram Burgard, Maximum-likelihood sample-based maps for mobile robots, Robotics and Autonomous Systems 58 (2010) 133-139

[33] Ahmed S.Al-Araji, Maysam F. Abbod, Hamed S. Al-Raweshidy, Applying posture identifier in designing and adaptive nonlinear predictive controller for Non-holonomic mobile robot, Neurocomputing 99 (2013) 543-554

[34] Chi-Yi Tsai, Kai-Tai Song, Dynamic visual tracking control of a mobile robot with image noise and occlusion robustness, Image and Vision Computing 27 (2009) 1007-1022

[35] Luigi Freda, Giuseppe Oriolo, Vision-based interception of a moving target with a Non-holonomic mobile robot, Robotics and Autonomous Systems 55 (2007) 419-432

[36] Jun Ye, Tracking control for Non-holonomic mobile robots: Integrating the analog neural network into the back-stepping technique, Neurocomputing 71 (2008) 3373-3378

[37] Markus Mauder, Robust tracking control of Non-holonomic dynamic systems with application to the bi-steerable mobile robot, Automatica 44 (2008) 2588-2592

[38] Farzad Pourboghrat, Mattias P. Karlsson, Adaptive control of dynamic mobile robots with Non-holonomic constrints, Computers and Electrical Engineering 28 (2002) 241-253

[39] Farzad Pourboghrat, Exponential stabilization of Non-holonomic mobile robots, Computer and Electrical Engineering 28 (2002) 349-359

[40] Juan Marcos Toibero, Flavio Roberti, Ricardo Carelli, Paolo Fiorini, Switching control approach for stable navigation of mobile robots in unknown environments, Robotics and Computer-Integrated Manufacturing 27 (2011) 558-568

[41] Omid Mohareri, Rached Dhaouadi, Ahmad B. Rad, Indirect adaptive tracking control of a Non-holonomic mobile robot via neural networks, Neurocomputing 88 (2012) 54-66

[42] Shuzhi Sam Ge, Xue-Cheng Lai, Abdullah Al Mamun, Sensor-based path planning for non-holonomic mobile robots subject ot dynamic constaints, Robotics and Autonomous Systems 55 (2007) 513-526

[43] P.Moustris, Spyros G. Tzafestas, Switching Fuzzy tracking control for mobile robots under curvature constraints, Control Engineering 19 (2011) 45-53

[44] Murelitharan Muniandy, Kanesan Muthusamy, An Innovative Design to Improve systematic Odometry Error in Non-holonomic Wheeled Mobile Robots, International Symposium on Robotics and Intelligent Sensors 2012 (IRIS 2012), Procedia Engineering 41 (2012) 436-442

[45] Ibrahim M.H. Sanhoury, Shamsudin H.M. Amin, Abdul Rashid Husain, Multiple Non-holonomic Wheeled Mobile Robots Trajectory Tracking While Maintaining Time-Varying Formation via Synchronous Controller, International Symposium on Robotics and Intelligent Sensors (IRIS 2012), Procedia Engineering 41 (2012) 1044-1050

[46] Ranajit Chattgerjee, Fumitoshi Matsuno, Use of single side reflex for autonomous navigation of mobile robots in unknown environments, Robotics and Autonomous Systems 35 (2001) 77-96

[47] S.G. Tzafestas, K.M. Deliparaschos, G.P. Moustris, Fuzzy logic path tracking control for autonomous Non-holonomic mobile robots:Design of System on a Chip, Robotics and Autonomous Systems 58 (2010) 1017-1027

[48] M.J. Er, Tien Peng Tan, Sin Yee Loh, Control of a mobile robot using generalized dynamic fuzzy neural networks, Microprocessors and Microsystems 28 (2004) 491-498

[49] O. Motlagh, S.H.Tang, N. Ismail, A.R. Ramli, An expert fuzzy cognitive map for reactive navigation of mobile robots, Fuzzy Sets and Systems 201 (2012) 105-121

[50] Wang Yaonan, Yang Yimin, Yuan Xiaofang, Zuo Yi, Zhou Yuanli, Yin Feng, Tan Lei, Autonomous mobile robot navigation system designed in dynamic environment based on transferabel belief model, Measurement 44 (2011) 1389-1405

[51] L. McFetridge, M.Y. Ibrahim, A new methodology of mobile robot navigation: The agoraphilic algorithm, Robotics and Computer-Integrated Manufacturing 25 (2009) 545-551

[52] Foudil Abdessemed, Khier Benmahammed, Eric Monacelli, A fuzzy-based reactive controller for a non-holonomic mobile robot, Robotic and Autonomous Systems 47 (2004) 31-46

[53] Ilan Zohar, Amit Ailon, Raul Rabinovici, Mobile robot characterized by dynamic and kinematic equations and actuator dynamics: Trajectory tracking and related application, Robotics and Autonomous Systems 59 (2011) 343-353

[54] Saso Blazic, A novel trajectory-tracking control law for wheeled mobile robots, Robotics and Autonomous Systems 59 (2011) 1001-1007

[55] Tamoghna Das, I.N. Kar, S. Chaudhury, Simple neuron-based adaptive controller for a Non-holonomic mobile robot including actuator dynamics, neurocomputing 69 (2006) 2140-2151

[56] Edward Red, Robotics Overview, Professor Manufacturing & Mechanical Engineering 2001

[57] Mohd Seafiq B. Saipollah, Design and Simulation and of Robot Spray Painting, Faculty of Manufacturing Engineering, May 2008

[58] John J. Craig, Introduction to Robotics, Mechanics and Control, Addison Wesley, 1989.

[59] Eugenio Aguirre, Antonio Gonz_alez, Fuzzy behaviors for mobile robot navigation: design, coordination and fusion, International Journal of Approximate Reasoning 25 (2000) 255-289

[60] Sirma C. Yavuz, Kinematic Analysis for Robot Arm, Yildiz Technical University, Electrical and Electronics Faculty, Istanbul, 2009

[61] Rafigul I. Noorani, Micro-computer based Robot Arm Control, Department of Mechanical Engineering, Loyola Marymount University, and Math1 Cornput. Modelling, Vol.1 4,pp.4 50-455,1990

[62] Francisco Fernandez Navarro, Cesar Hervas Martinez, Javier Sanchez Monedero, Pedro Antonio Gutierrez, MELM-GRBF: A modified version of the extreme learning machine for generalized radial basis function neural networks, Neurocomputing 74 (2011) 2502–2510

[63] Rommel G. Regis, Christine A. Shoemaker, Parallel radial basis function mehods for the global optimization of expensive functions, European Journal of Operational Research 182 (2007) 514–535

[64] Qinggang Meng, Mark Lee, Error-driven active learning in growing radial basis function networks for early robot learning, Neurocomputing 71 (2008) 1449–1461

[65] Maykol Pinto, Luis F. Rocha, A. Paulo Moreira, Object recognition using laser range finder and machine learning techniques, Robotics and Computer-Integrated Manufacturing 29 (2013) 12–22

[66] hirotaka Akita, Md. Monirl Islam, Tooru matumoto and Kazuyuki Murase, Back-Propagation Learning in Real Autonomous Mobile Robot using Proximal and Disrtal Evaluation of Behaviour, Department of Human and Artificial Intelligence System, Fukui University, 2000 (IEEE)

[67] L.Mesbhahi, A. Benyettou, F.Hendel, N.E.Berrached, Reachtive navigation of mobile robot by temporal radial basis function approach, 2004 (IEEE) 484-487

List of Publication

1. Published **International** Journals on **"Different Methodologies of Navigation of Autonomous Mobile Robot for Unknown Environment"**, Dayal R. Parhi and **Rakesh Kumar Sonkar**, IJAICR4(2), 2012, pp 79-85
2. "**A Review of a fuzzy based Non-holonomic Mobile Robot Navigation using different Control Techniques**", **Rakesh Kumar Sonkar**, Dayal R. Parhi, Anish Pandey (Paper Communicated)
3. "**Kinematic Analysis of Corobot with 4-DOF Arm**", **Rakesh Kumar Sonkar**, Anish Pandey, Dayal R. Parhi (Paper Communicated)
4. "**A Review of Mobile Robot Navigation using a Fuzzy Logic**", Anish Pandey, Dayal R. Parhi, **Rakesh Kumar Sonkar** (Paper Communicated)

Conferences: -

1. **"Navigation of Mobile Robot with Obstacles avoidance using Fuzzy Logic"**, Anish Pandey, **Rakesh Kumar Sonkar**, Dayal R. Parhi, **International Conference** on Advance Mechanical Engineering, 29th - 31st May 2013, College of Engineering Pune, India
2. **"Path Planning Navigation of Mobile Robot with Obstacles Avoidance using Fuzzy Logic Controller",** Anish Pandey, **Rakesh Kumar Sonkar**, Dayal R. Parhi, **1st International** & **16th National Conferences** on Machines and Mechanics (iNaCoMM 2013) 18-20 DEC 2013, Mechanical & Industrial Engineering Department, **IIT Roorkee**, India (Paper Accepted)