# ONLINE PATH PLANNING OF MOBILE ROBOT FOR OBSTACLE AVOIDANCE USING ARTIFICIAL POTENTIAL FIELD METHOD WITH THE HELP OF ULTRASONIC RANGE SENSOR

By

**PRASENJIT PAL**

B. Tech (Mechanical Engineering), 2013
KALYANI GOVERNMENT ENGINEERING COLLEGE

**EXAMINATION ROLL NO. – M4PRD1603**

**THESIS**
**SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF MASTER OF PRODUCTION ENGINEERING IN THE FACULTY OF ENGINEERING & TECHNOLOGY,**
**JADAVPUR UNIVERSITY**
**2016**

**DEPARTMENT OF PRODUCTION ENGINEERING**
**JADAVPUR UNIVERSITY**
**KOLKATA-700032**
**INDIA**

# JADAVPUR UNIVERSITY
# FACULTY OF ENGINEERING AND TECHNOLOGY

## CERTIFICATE OF RECOMMENDATION

I/WE HEREBY RECOMMEND THAT THE THESIS ENTITLED **"ONLINE PATH PLANNING OF MOBILE ROBOT FOR OBSTACLE AVOIDANCE USING ARTIFICIAL POTENTIAL FIELD METHOD WITH THE HELP OF ULTRASONIC RANGE SENSOR"** CARRIED OUT UNDER MY/OUR SUPERVISION AND GUIDANCE BY **MR. PRASENJIT PAL** MAY BE ACCEPTED IN THE PARTIAL FULLFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF **"MASTER OF PRODUCTION ENGINEERING".**

**------------------------------------**
**(Subir Kumar Debnath)**
**Thesis Advisor**
**Dept. of Production Engineering**
**JADAVPUR UNIVERSITY**
**Kolkata-700032**

**------------------------------------**
**(Dr. Debamalya Banerjee)**
**HEAD, Dept. of Production Engineering**
**JADAVPUR UNIVERSITY**
**Kolkata-700032**

**------------------------------------**
**Dean, F.E.T.**
**JADAVPUR UNIVERSITY**
**Kolkata - 700032**

## JADAVPUR UNIVERSITY
## FACULTY OF ENGINEERING AND TECHNOLOGY

## CERTIFICATE OF APPROVAL*

The foregoing thesis is hereby approved as a creditable study of an engineering subject carried out and presented in a manner of satisfactory to warrant its acceptance as a pre-requisite to the degree for which it has been submitted. It is understood that by this approval, the undersigned do not necessarily endorse or approve any statement made, opinion expressed and conclusion drawn therein but approve the thesis only for the purpose for which it has been submitted.

**COMMITTEE ON**
**FINAL EXAMINATION**
**FOR EVALUATION OF**
**THE THESIS**

______________________________

______________________________

***** Only in case the recommendation is concurred in

# *ACKNOWLEDGEMENT*

*I owe a deep debt of gratitude to my project supervisor* ***Mr. Subir Kumar Debnath****, Department of Production Engineering, Jadavpur University for his unreserved guidance, beneficial suggestions, thought provoking suggestions and unshrinking inspiration in fostering this research work. It has been a blessing for me to spend many auspicious moments under the guidance of the perfectionist at the pinnacle of the professionalism. This present work is a manifestation to his keenness and inspiration, taken by him during the course of this thesis work in the present form.*

*I am also thankful to* ***Dr. Ajoy Kumar Dutta*** *for his encouragement and kind support during the research work.*

*I also take this opportunity to express my gratitude to all the faculty members of Production Engineering Department, specially* ***Dr. Debamalya Banerjee****, Head of the Department, for their mental support, immense help and co-operation during the course of this thesis work.*

*It is a pleasure to acknowledge the support and help provided by all of my friends and research scholars of this department.*

*I would like to express my respect to all the faculty members and non-teaching staffs of Production Engineering Department, Jadavpur University for their cordial assistance during this thesis work.*

*I thankfully acknowledge the support of all those people, who some time or the other directly or indirectly rendered their help at different stages of this work.*

*I cannot close these prefatory remarks without expressing my heartfelt gratitude to my parents for their blessings and endeavor to keep my aspiration fixed throughout the period of my research work.*

(*Prasenjit pal*)

# TABLE OF CONTENTS

# CHAPTER 1

# INTRODUCTION

## 1.1 INTRODUCTION TO MOBILE ROBOT

Mobile robot is an automatic machine which has the capability to move around in an environment and is not fixed to one physical location. Mobile robots can be "autonomous" (AMR - autonomous mobile robot) which means they are capable of navigating in unstructured environment without the need for physical or electro-mechanical guiding devices. Alternatively, mobile robots can rely on guiding devices that allow them to travel on a pre-defined navigation route in relatively controlled space (AGV - autonomous guided vehicle). By contrast, industrial robots are usually more-or-less stationary, consisting of a jointed arm (multi-linked manipulator) and gripper assembly (or end effector), attached to a fixed surface.

A lot of advanced equipment has been introduced in everyday life for the world that is rapidly developing. There has been use of an autonomous vehicle or robot in many of situation to facilitate a job such as unmanned vehicle which is a vehicle without a human on board. In others words unmanned vehicles can be autonomous vehicles which are capable of sensing their environment on their own or can either be remote controlled or remote guided vehicles.

Robots could be the only viable alternative for construction and manipulation tasks in environments that are hazardous or inaccessible for humans, e.g., disaster areas, extraterrestrial surfaces, underground mines, or undersea. However, the employment of autonomous robots in these environments is still very challenging, and demands more research. Nature is one of the sources of inspiration that can help us in this regard. We can see, by observing nature, how simple agents employ adaptive and robust solutions to construct in dynamic and unstructured environments. Examples of such constructions include beaver dams, termite mounds, caddis fly cases, bee hives, social weaver nests, spider webs, and anthill structures.

Autonomous robots which work without human operators are required in robotic fields. In order to achieve its goal, autonomous robots have to be intelligent and should decide their own action. When an autonomous robot moves from a point to a target point in its given environment, it is necessary to plan an optimal and feasible path avoiding obstacles in its way. So, path planning is one of the important factors to be considered while developing an autonomous robot. Path planning gives the path to be followed by the robot in order to reach the target.

Mobile robots have become more commonplace in commercial and industrial settings. Hospitals have been using autonomous mobile robots to move materials for many years. Warehouses have installed mobile robotic systems to efficiently move materials from stocking shelves to order fulfillment zones. Mobile robots are also a major focus of current research and almost every major university has one or more labs that focus on mobile robot research. Mobile robots are also found in industrial, military and security settings. Domestic robots are consumer products, including entertainment robots and those that perform certain household tasks such as vacuuming or gardening.

## 1.2 CLASSIFICATION OF MOBILE ROBOT

Mobile robots may be classified by:

The environment in which they travel:

- Land or home robots are usually referred to as Unmanned Ground Vehicles (UGVs). They are most commonly wheeled or tracked, but also include legged robots with two or more legs (humanoid, or resembling animals or insects).
- Delivery & Transportation robots can move materials and supplies through a work environment
- Aerial robots are usually referred to as Unmanned Aerial Vehicles (UAVs)
- Underwater robots are usually called autonomous underwater vehicles (AUVs)
- Polar robots, designed to navigate icy, crevasse filled environments

The device they use to move, mainly:

- Legged robot : human-like legs (i.e. an android) or animal-like legs.
- Wheeled robot.
- Tracks

## 1.3 IMPORTANCE OF MOBILE ROBOT

We are living in a continuously changing field of industrial, technical, medical and research oriented activity. Sometimes in the above field a situation comes to us where we can not reach our goal for some difficulty or inability. Today mobile robots are continuously gaining in importance in particular under considerations of reliability (uninterrupted and reliable execution of monotonous tasks such as surveillance), accessibility (inspection of sites that are inaccessible to humans, e.g. tight spaces, hazardous environments or remote sites) or cost (transportation systems based on autonomous mobile robots can be cheaper than standard track-bound systems). Mobile robots are already widely used for surveillance, inspection and transportation tasks. A further emerging market with enormous potential is that of mobile entertainment robots. Applications of mobile robot are:

- Support to medical services – service robots:
  Transportation of food, medication, medical exams, Automation of pharmacy service.
- Automatic cleaning of (large) areas Supermarkets, airports, industrail sites, Glass cleaning, Domestic vacuum-cleaner
- Client support :
  Museum tours, exhibitions guides
- Agricultural:
  Fruit and vegetable picking, fertilization, planting
- Forests :
  Cleaning, fire preventing, tree cutting
- Material Handling:
  AGVs, SGVs, LGVs
- Safety:
  Surveillance of large areas, buildings, airports, car parking lots
- Civil Transportation:
  Inspection of airplanes, trains, etc
- Elderly and Handicapped :
  Assistance to handicapped or elderly people, helping in transportation, healthcare,
- Entertainment:
  RobotDog,Aibo – Robot dog from Sony Telepresence

## 1.4 RANGE SENSOR IN MOBILE ROBOT

In case of mobile robot, for autonomous control, a lot of information about the environment in which it work is required. Sensor in mobile robot is a component which is used to collect data from the environment. Then all the information collected from sensors are sent to the controller. Then according to the information, controller takes the correct action. Information may consist of the position, shape, size and color of the obstacles. There are many sensors which are used in mobile robots, like ultrasonic sensor, infrared sensor, laser sensor, vision sensor, tactile sensor etc.

Among these, range sensors take a major role in mobile robot. It is extensively used in mobile robot. When the mobile robot is in unknown environment, it gets distance information by using range sensor. From range sensor mobile robot can also get the size and position of the obstacles with respect to the current position of the mobile robot.

## 1.5 LITERATURE SURVEY

Extensive research work has been carried out in the field of mobile robot path planning using range sensor and artificial potential field algorithm. Some of those have been reported below.

Lim *et.al.* [1] presents deployment strategies of ultrasonic sensors and a way of using Smartphone sensors to help mobile robot navigation in indoor environments. There are critical needs for cost-effective, reliable, and fairly accurate solutions to meet the demands of indoor robotic applications. Ultrasonic sensors have been popular in detecting simple objects due to the low-cost and simplicity despite their limitations. they proposed an efficient way of deployment of ultrasonic sensors for low-cost mobile robots. A Smartphone has many high performance sensors that can be utilized to navigate and localize mobile robots. The sensors include a camera, a gyroscope, and an accelerometer. We analyzed the use of orientation sensor of a Smartphone and compared its performance to a conventional approach. The comparison results were promising. The combination of the efficient way of the sensor deployment and the use of Smartphone sensors shows a possibility of developing a low-cost indoor mobile robotics platform for college education and robotics research laboratories.

Borenstein and koren [1988] [2] developed a mobile robot system, capable of performing various tasks for the physically disabled. To avoid collision with unexpected obstacles, the mobile robot uses ultrasonic range finders for detection and mapping. The obstacle avoidance strategy used for this robot is described. Since this strategy depends heavily on the performance of the ultrasonic range finders, these sensors and the effect of their limitations on the obstacle avoidance algorithm are discussed in detail.

Pears and Probert [1996] [3] described The mechatronic design of an eye-safe laser rangefinder, based on the lateral-effect photodiode (LEP) and synchronised scanning. The sensor acquires two-dimensional range data, which has been found sufficient to guide the local manoeuvres of a mobile robot in most environments. An analysis of LEP operation shows that image position measurement repeatability, normalized with respect to the detector half length, is equal to the signal current to noise current ratio. This result suggests a method for estimating the noise density of the measurement process and, along with a geometric model of the ranging process, allows accurate estimation of the variance of individual range measurements, making the sensor particularly amenable to statistically based range feature detection, tracking, and data fusion algorithms. The sensor is active, in the sense that it can change the orientation of its field of view, in order to track useful and stable range features. Range data acquisition, range feature extraction, and control of the active head behaviour are all implemented on a local network of six transputers. This parallel structure is described and it is shown how the sensor constitutes an intelligent agent in a balanced sensor suite for the guidance of close range mobile robot manoeuvres.

Nagahara *et. al* . [2010] [4] proposed a mobile robot control method which based on distance information obtained by scanning laser range finder. First, the mobile robot gets distance information. This distance information is between the mobile robot and objects. The scanning laser range sensor is put on the front of the mobile robot. The mobile robot decides whether the obstacles are present based on distance information of each angles. At this time, the angle where the distance information below the threshold was detected is judged as an obstacle. The mobile robot calculates sub-goals to the angle where the obstacles do not exist. The reference point for obstacles avoidance is calculated by composing calculated sub-goals. We make the robot track the reference point which is calculated and reach the goal point without colliding obstacles. Finally, we experiment in a real environment, and verify the utility of the propose method.

Pfister *et. al.* [2012] [5] introduces a "weighted" matching algorithm to estimate a robot's planar displacement by matching two dimensional range scans. The influence of each scan point on the overall matching error is weighted according to its uncertainty; they developed uncertainty models that account for effects such as measurement noise, sensor incidence angle, and correspondence error. Based on models of expected sensor uncertainty, our algorithm computes the appropriate weighting for each measurement *so* as to optimally estimate the displacement between two consecutive poses. By explicitly modeling the various noise sources, we can also calculate the actual covariance of the displacement estimates instead of a statistical approximation of it. A realistic covariance estimate is necessary for further combining the pose displacement estimates with additional odometric and/or inertial measurements within a localization framework .

Experiments using a Nomad 200 mobile robot and a Sick LMS-200 laser range finder illustrate that the method is more accurate than prior techniques.

Nelson *et. al.* [2013] [6] described a customizable mobile robot platform capable of automatic mapping with well-known algorithms was built. This robot has robust sensors, contains unambiguous code, and uses familiar microcontrollers so as to be readily useable by another team. The robot is also capable of localization and mapping so that future teams lacking excessive knowledge of mapping algorithms, sensors, or electronics can branch into software based computer vision projects with ease.

Li *et. al.* [7] present an improved artificial potential field based regression search (Improved APF-based RS) method which can obtain a global sub-optimal/optimal path efficiently without local minima and oscillations in complete known environment information. We redefine potential functions to eliminate non-reachable and local minima problems, and utilize virtual local target for robot to escape oscillations. Due to the planned path by improved APF is not the shortest/approximate shortest trajectory, they develop a regression search (RS) method to optimize the planned path. The optimization path is calculated by connecting the sequential points which produced by improved APF. Amount of simulations demonstrate that the improved APF method very easily escape from local minima and oscillatory movements. Moreover, the simulation results confirm that our proposed path planning approach could always calculate a more global optimal/near optimal, collision-free and safety path to its destination compare with general APF. That proves our improved APF-based RS method very feasibility and efficiency to solve path planning which is a NP-hard problem for autonomous mobile robot.

Zhang and Ferrari [8] present a novel artificial potential function is proposed for planning the path of a robotic sensor in a partially observed environment containing multiple obstacles and multiple targets. The sensor planning problem considered in this paper consists of planning the motion of a robot with an on-board sensor that is deployed in order to support a sensing objective, such as, target detection and classification, by gathering sensor measurements over time. An adaptive potential function approach is presented such that the sensor path accounts for prior information on the target geometry and information profit, by traveling a minimum distance.

Seki *et. al.* [9] discuss about the obstacle avoidance for a nonholonomic vehicle (mobile robot) like an autonomous wheelchair . It has two independently driven wheels and a body with a certain shape. If a vehicle can be treated as an omnidirectional movable point, numerous methods have been proposed and applied for it. Collision free path can be easily found by artificial potential field (Khatib, 1986; Rimon & Koditsuchek, 1992), graph theory (Ulrich & Borenstein, 2000), sensor based method and so on. The problem for a nonholonomic vehicle with two independently driven wheels can come down to that for an omnidirectional point by approximating vehicle's shape to a circle with the center at the midpoint of two wheels. Obstacles should be expanded by the radius of the vehicle's circle and the vehicle should be contracted to a point. However, it isn't reasonable to regard the rectangular body like a wheelchair as a circle and its circle sometimes can't pass through the narrow place where the original body can do.

Bornstein and Koren [10] had presented a new real-time obstacle avoidance approach for mobile robots. This approach permits the detection of unknown obstacles simultaneously with the steering of the mobile robot to avoid collisions and advancing toward the target. The novelty of this approach, entitled the virtual force field, lies in the integration of two known concepts: Certainty grids for obstacle representation, and potential fields for navigation. This combination is especially suitable for the accommodation of inaccurate sensor data as well as for sensor fusion, and enables continuous motion of the robot without stopping in front of obstacles. This navigation algorithm also takes into account the dynamic behavior of a fast mobile robot and solves the local minimum trap problem. Experimental results from a mobile robot running at a maximum speed of 0.78 m/s demonstrate the power of the proposed algorithm.

Xie *et. al.* [11] proposed a new method using virtual water-flow is proposed to escape local minima occurred in local path planning, which integrates virtual water-flow with a potential-field-based method to guide a mobile robot in an unknown or unstructured environment. The potential-field method coupled with virtual water-flow can navigate a mobile robot in real time. Simulations and experiments show this algorithm possesses good performance, and can overcome the problem cause by local minimum.

Vadakkepat *et. al.* [12] discusses the application of the evolutionary artificial potential field (EAPF) in mobile robot path planning. The parameters of the evolutionary artificial potential field are optimized with the multiobjective evolutionary algorithm. The EAPF is utilized in a robot soccer System.

Hwang and Ahuja [1988] [13] presents an approach to two-dimensional as well **as** threedimensional findpath problem that divides the problem into two steps. First, rough paths are found based only on topological information. This is accomplished by assigning to each obstacle an artificial potential similar to electrostatic potential to prevent the moving object from colliding with the obstacles, and then locating minimum potential valleys. Second, the paths defined by the minimum potential valleys are modified to obtain an optimal collision-free path and orientations of the moving object along the path. Three algorithms are given to accomplish this second step.These three algorithms based on potential fields are nearly complete in scope, and solve a large variety of problems.

Warren [1989] [14] describes a path planning technique for robotic manipulators and mobile robots in the presence of stationary obstacles. The planning consists of applying potential fields around *C-Space* obstacles and using this field to select a safe path for the robot to follow. The advantage of using potential fields in path planning is that they offer a relatively fast and effective way to solve for safe paths around obstacles. In the method used to accomplish path planning presented here, a trial path is chose and then modified under the influence of the potential field until an appropriate path is found. By considering the entire path, the problem of being trapped in a local minimum is greatly reduced, allowing the method to be used for global planning. The algorithm was tried with success on many different realistic planning problems. The examples in this paper illustrate the algorithm applied to a two dimensional revolute manipulator, a mobile robot capable of translation only, and a mobile robot capable of both translation and rotation.

Tilove [1990] [15] presents an overview of the method, describes the common variations in a unified framework, compares the performance of the different algorithms, and corrects some common misunderstandings. We show that force control algorithms with generalieed potentials (which depend on velocity as well as position) offer advantages over other alternatives. We also discuss the important but largely overlooked problems of stability and convergence.

Koren and Borenstein [1991] [16] presents a systematic criticism of the inherent problems. The heart of this analysis is a differential equation that combines the robot and the environment into a unified system. The identified problems are discussed in qualitative and theoretical terms and documented with experimental results from actual mobile robot runs.

Ge and Cui [2000] [17] describes the problem of goals nonreachable with obstacles nearby when using potential field methods for mobile robot path planning. Then, new repulsive potential functions are presented by taking the relative distance between the robot and the goal into consideration, which ensures that the goal position is the global minimum of the total potential.

Mbede *et. al.* [2000] [18] focuses on autonomous motion planning of manipulators in known environments and with unknown dynamic obstacles. The navigation technique of robot control using artificial potential functions is based on fuzzy logic and stability is guaranteed by Lyapunov theory. A fuzzy logic system or fuzzy system is a universal approximator which provides a rule-based mapping between the input and the output space, while classical approaches make use of analytic harmonic functions to solve the navigation problem. In this particular application, the fuzzy system proposed is used to approximate the gradient of the harmonic functions.

Park and Lee [2003] [19] present a new concept using a virtual obstacle is proposed to escape local minimums occurred in local path planning. A virtual obstacle is located around local minimums to repel a mobile robot from local minimums. A sensor based discrete modeling method is also proposed for modeling of the mobile robot with range sensors. This modeling method is adaptable for a real-time path planning because it provides lower complexity

Kowalczuk and Duzinkiewicz [2007] [20] introduce a multi-stage trajectory planning algorithm for mobile robots. *Unmanned Ground Vehicles* (UGVs) and

Unmanned Aerial Vehicles (UAVs) became very popular in the last few years. Many military and civil applications can be found in real life. One of the most challenging issues regarding UGVs and UAVs concern their autonomous control in environment with obstacles. The proposed algorithm is based on an isoline method, modification of the artificial potential field method, designed to efficiently escape local minima of the artificial potential function.

Safadi [2007] [21] discuss about implementation of a local path planning algorithm based on virtual potential field described in. The algorithm uses virtual forces to avoid being trapped in a local minimum. Simulation and experiments are performed, and compared to the results presented in the paper. They show good performance and ability to avoid the local minimum problem in most of the cases.

Liu and Bharadwaj [2011] [22] developed and implemented a hybrid Artificial Potential Field Genetic Algorithm approach for mobile robot path planning in dynamic environments. The hybrid approach first uses Grid Method where the mobile robot environment is represented by orderly numbered grids, each of which represents a location in the environment. Then, it applies Genetic Algorithm (GA), a global planner, to find an optimal path according to the current environment. The GA proposed here uses an evolutionary population initialization and genetic operators, which make the evolutionary process, converge very efficiently. Finally, a new Artificial Potential Field method, a local planner, is applied to follow the path obtained by GA from one intermediate node to next intermediate node avoiding the obstacles. Experimental results clearly illustrate that the proposed hybrid approach works well on large scale dynamic environments.

Xu *et. al.* [2012] [23] discussed The well-known potential field method for obstacle avoidance in the scope of mobile robot . Particular attention is on the car-like mobile robots, which impose practical limitations on the application of potential field method due to its limited speed and curvature in motion. Along with the review of some recent studies on this topic, we point out the necessity of implementing a nonholonomic motion planner and propose some extensions to other potential-field-related methods to deal with the constraints of car-like robots. Two exemplary scenarios based on our extensions are simuated to prove their feasibility in application.

Zhang *et.al***.** [2012] [24] proposed a new obstacle avoidance method. In this method, used the grid method to describe the information of obstacles environment, utilized the evolutionary artificial potential field method to optimize obstacle avoidance path. The simulation results show that the proposed method is feasible and effective.

Lee *et. al*. [2012] [25] describes new problem of symmetrically aligned robot-obstacle-goal (SAROG) when using potential field methods for mobile robot path planning. In addition, we consider constant robot speed for practical use. The SAROG and the constant speed involve two potential risks: robot-obstacle collision and local minima trap.For dealing with the two potential risks, we analyze the conditions of the collision and the local minima trap, and propose new potential functions and random force based algorithms. For the algorithm verification, we use WiRobot X80 with three ultrasonic range sensor modules.

Chauhan and Bajpai [2012] [26] present an optimum path without colliding with the obstacle or obstacles for a mobile robot to reach a predefined goal in an environment containing stationary or dynamic obstacles. Artificial potential field method has been employed for the purpose in this paper. However, there are certain critical situations in which this method may fail. These situations have been analyzed and a simple but robust algorithm has been suggested to overcome these situations. In this paper an algorithm for path planning is used to get optimum path without colliding with obstacle. The proposed path planning must make the robot to avoid obstacle to get optimum path towards its goal.

Li *et. al.*[2013] [27] presents a new way for mobile robots' path planning which is based on the Evolutionary Artificial Potential Fields(EAPF) approach. The APF theory is a traditional method to plan path for a robot. The Evolutionary APF aims at helping a robot jump out of the local minimum point. Using a virtual goal to produce extra force and fixing the direction of the repulsive force are combined to prompt the robot to escape from the obstacle in different situations. The simulation result shows that the evolutionary method is effective for solving the local minimum problem.

Malakar and Sinha [2013] [28] present an improved artificial potential fiel based regression search method, by analysing the shortcoming of the artificial potential field methods for robot path planning, we propose an obstacle avoidance method based on gravity chain, which can obtain a global suboptimal/optimal path efficiently without local minima and oscillations incompletely known environment information.

Wyrąbkiewicz *et. al.* [2014] [29] present a paper on artificial potential fields method applied to autonomous mobile robot mars rover. It is assumed that Mars rover operates in an unknown environment. In order to visualize the robot's path in environment Matlab software is used. The object can be inserted by graphic data input interface in top view mode. The method of artificial potential fields is extended by an additional algorithm to avoid a local minimum. The proposed algorithm is implemented as a state machine. In this paper simulations results of the developed algorithm are presented. Extended algorithm is used because in the environment may be located complex obstacles.

Min *et. al.* [2014] [30] proposed a new method to help the mobile robot to avoid many kinds of collisions effectively, which combined past experience with modified artificial potential field method. In the process of the actual global obstacle avoidance, system will invoke case-based reasoning algorithm using its past experience to achieve obstacle avoidance when obstacles are recognized as known type; otherwise, it will invoke the modified artificial potential field method to solve the current problem and the new case will also be retained into the case base. In case-based reasoning, we innovatively consider that all the complex obstacles are retrieved by two kinds of basic build-in obstacle models (linear obstacle and angle-type obstacle). Our proposed experience mixing with modified artificial potential field method algorithm has been simulated in MATLAB and implemented on actual mobile robot platform successfully. The result shows that the proposed method is applicable to the dynamic real-time obstacle avoidance under unknown and unstructured environment and greatly improved the performances of robot path planning not only to reduce the time consumption but also to shorten the moving distance.

Weerakoon *et. al*. [2015] [31] present a deadlock free APF based path planning algorithm for mobile robot navigation. The Proposed-APF (P-APF) algorithm searches the goal point in unknown 2D environments. This method is capable of escaping from deadlock and non-reach ability problems of mobile robot navigation. In this method, the effective front-face obstacle information associated with the velocity direction is used to modify the Traditional APF (T-APF) algorithm. This modification solves the deadlock problem that the T-APF algorithm often converges to local minima. The proposed algorithm is explained in details and to show the effectiveness of the proposed approach, the simulation experiments were carried out in the MATLAB environment. Furthermore, the numerical analysis of the proposed approach is given to prove a deadlock free motion of the mobile robot.

Chen *et. al.* [2015] [32] present a paper to solve the problem of local minima and unreachable destination of the traditional artificial potential field method in mobile robot path planning, chaos optimization is introduced to improve the artificial potential field method. The potential field function was adopted as a target function of chaos optimization, and a kind of “two-stage” chaos optimization was used. The corresponding movement step and direction of the robot were achieved by chaos search. Comparison of the improved method proposed in this paper and the traditional artificial potential field method is performed by simulation. The simulation results show that the improved method gets rid of the drawbacks, such as local minima and unreachable goal. Furthermore, the improved method is also verified by building up a physical platform based on “Future Star” robot. The success of the physical experiment indicates that the improved algorithm is feasible and efficient for mobile robot path planning

Zhou and Li [2015] [33] presents an improved artificial potential field method to intelligent vehicle collision-avoidance path planning in simulated outdoor scene. Artificial potential field method has been widely applied to indoor robot collision-avoidance-path planning. However, it’s not much used for outdoor scene. Whether perception model or environment model, the outdoor intelligent vehicle collision-avoidance is much more complex than indoor robot. Therefore, we have improved the method: Firstly, we have improved the perception model of the method. To simulate real-world scene, the intelligent vehicle sensing range is limited to 150◦ fan-shaped area.

Secondly, We have improved the potential function components, that is, adding the road constraints. The intelligent vehicle not only need to achieve collision-avoidance, but also can’t exceed the road boundary. Meanwhile, we effectively solve the most common problem of local minima. The simulation results show: the improved algorithm can accomplish the intelligent vehicle collision-avoidance-path planning well in a realistic simulation scene.

Ahmed *et. al.* [2015] [34] present a paper deals with the navigation of a mobile robot in an unknown environment by using artificial potential field (APF) method. The aim is to develop a method for path planning of mobile robot from start point to the goal point while avoiding obstacles on robot’s path. Artificial potential field method will be modified and optimized by using particle swarm optimization (PSO) algorithm to solve the drawbacks such as local minima and improve the quality of the trajectory of mobile robot.

Montiel *et. al.* [2015] [35] introduce the concept of Parallel Evolutionary Artificial Potential Field (PEAPF) as a new method for path planning in mobile robot navigation. The main contribution of this proposal is that it makes possible controllability in complex real-world sceneries with dynamic obstacles if a reachable configuration set exists. The PEAPF outperforms the Evolutionary Artificial Potential Field (EAPF) proposal, which can also obtain optimal solutions but its processing times might be prohibitive in complex real-world situations. Contrary to the original Artificial Potential Field (APF) method, which cannot guarantee controllability in dynamic environments, this innovative proposal integrates the original APF, evolutionary computation and parallel computation for taking advantages of novel processors architectures, to obtain a flexible path planning navigation method that takes all the advantages of using the APF and the EAPF, strongly reducing their disadvantages. We show comparative experiments of the PEAPF against the APF and the EAPF original methods. The results demonstrate that this proposal overcomes both methods of implementation; making the PEAPF suitable to be used in real-time applications.

Li *et. al.* [2016] [36] described a potential-based car-following model using the concept of the artificial potential field, which aims for the precise and fast interactive operations in an evolving environment. Spacing headway is divided into two parts according to the potential influence region. The variation rate of the spacing headway generates the control force within the potential influence region, while the difference of desired and current velocity takes control out of the influence range. Calibration and validation of the simplified model are conducted using NGSIM data. Statistical tests show that the proposed model can reproduce the car following process very well.

## 1.6 OBJECTIVE AND SCOPE OF THE PRESENT WORK

Aim of the present project is the development of necessary algorithm and software to navigate a mobile robot in presence of obstacles from a starting point to a destination using ultrasonic range sensors. A Parallax mobile robot (Boe-Bot), existing in the Robotics laboratory of Production Engineering Department of Jadavpur University, has been used for the present project. It is then necessary to obtain a path avoiding obstacles present in static environments in the workspace. After determination of location of obstacles on the workspace by ultrasonic sensor, the Boe-Bot robot has been programmed to move toward the destination from the starting position following the path using artificial potential field method, a simple local path planning algorithm.

The algorithm determines the next point to move for the robot forward target from any current point, which is the starting point at the beginning. The next point is always determined from the output of the range sensor, that gives the position and radius of the obstacle and artificial potential field method algorithm is the driving force of the mobile robot. The driving force is the resultant of the repulsive forces due to obstacles and attractive force due to goal.

Hence the main objectives of the present work are as follows:

(a) To set up an arrangement for the workspace consisting of a Boe-Bot mobile robot with Ping ultrasonic range sensor and obstacles on a suitable worktable, having a marked boundary.

(b) To mount the Ping ultrasonic range sensor on a bread board fixed on the Boe-Bot mobile robot system, and make necessary hardware connection to connect it to the Basic Stamp microcontroller through its input-output port (pins).

(c) To develop a program in PBASIC to determine the size and position of the obstacles.

(d) To develop a local path planning algorithm using artificial potential field method in the presence of multiple unknown obstacle.

(e) To develop a program in PBASIC language for producing necessary movements of the Boe-Bot mobile robot in presence of static obstacles for moving from a starting point to a target point using artificial potential field Algorithm.

(f) To run the program for different layout of workspace for testing the algorithm.

Further scope of the present project includes development of path planning algorithms using other sensors and other potential function in different environment to obtain an optimized path from a source position to a goal point.

# CHAPTER 2

# PATH PLANNING OF MOBILE ROBOT

## 2.1 OVERVIEW OF PATH PLANNING

Obstacle avoidance is the back bone of autonomous control as it makes robot able to reach to destination without collision. Path planning is involved to generate the optimum path from source to destination on the basis of sensorial information of environment.

Path planning and obstacle avoidance for mobile robot navigation are challenging topics; and efficient and simple but precise algorithm is important in path planning. Both preventing any of the robot from collision with obstacle and guaranteeing that the robot reaches the goal are important facts when getting the robot to seek the goal.

Robot path planning is one of the most fundamental functions for mobile robot. There are two types of planning for autonomous mobile robot based on how much information is known about static environment: global path planning when the environment is clearly known and sensory based local path planning when partial or no information about the environment is known in advance. In the global path planning, the environment surrounding the robot and the position of obstacles are well known in advance, and the robot is required to navigate to its destination with avoiding obstacles. The complete robot path in such applications can be calculated from the prior knowledge of the coordinates of the starting point, the destination point, and obstacles. On the other hand, the local motion planning dynamically guides the robot according to the locally sensed obstacles, which requires less prior knowledge about the environment. Therefore, the local path planning methods are more suitable and practical for mobile robot, since the environment is too complicated to be known precisely and may also be time-varying .While, the local path planning faces significant challenges.

## 2.2 CLASSIFICATION OF MOBILE ROBOT PATH PLANNING

Depending on the condition of obstacle surrounding the mobile robot path planning is divided into two categories

**Static path planning**

Path planning in static environment means moving a robot from start to goal position where the obstacles are stationary. In static environment, mobile robots reach to the destination by sensing the obstacles coming across, to get an optimal solution with minimum cost.

**Dynamic path planning**

Dynamic path planning is a combination of knowledge-based and sensor-based path planning. The knowledge-based path planning could plan a path with limited or inaccurate information. If during the execution of this path, a sensor detects any obstacle, the knowledge-based path planner is informed, and it updates its world or environment model, and then finds a new path. The dynamic Planning is characterized by separate path planning and path execution modules, in which the execution module may give feedback to the planning module.

Depending on the environmental information available to the mobile robot path planning is classified as global and local path planning

**Global path planning**

Global path planning is a path planning that requires robot to move with priori information of environment. The information about the environment first loaded into the robot path planning program before determining the path to take from starting point to a target point. In this approach the algorithm generates a complete path from the start point to the destination point before the robot starts its motion. Global path planning is the process of deliberatively deciding the best way to move the robot from a start location to a goal location. Thus for global path planning, the decision of moving robot from a starting point to a goal is already made and then robot is released into the specified environment

### Local path planning

Local path planning requires robot to move in unknown environment or dynamic environment where the algorithm, used for the path planning, will response to the obstacle and the change of environment. Local path planning also can be defined as real time obstacle avoidance by using sensory based information

These path planning algorithms can be categorized into two based on the two aspects of completeness and the scope. From the completeness point of view, algorithms can be categorized as classical or heuristic. Classical algorithms aim to find an optimal path if exists or prove that there is no solution. Heuristic algorithms try to find a better path in a short time but do not guarantee to find a solution always. However, the most of the classical methods are computationally expensive and heuristic methods can fail in complex environments.

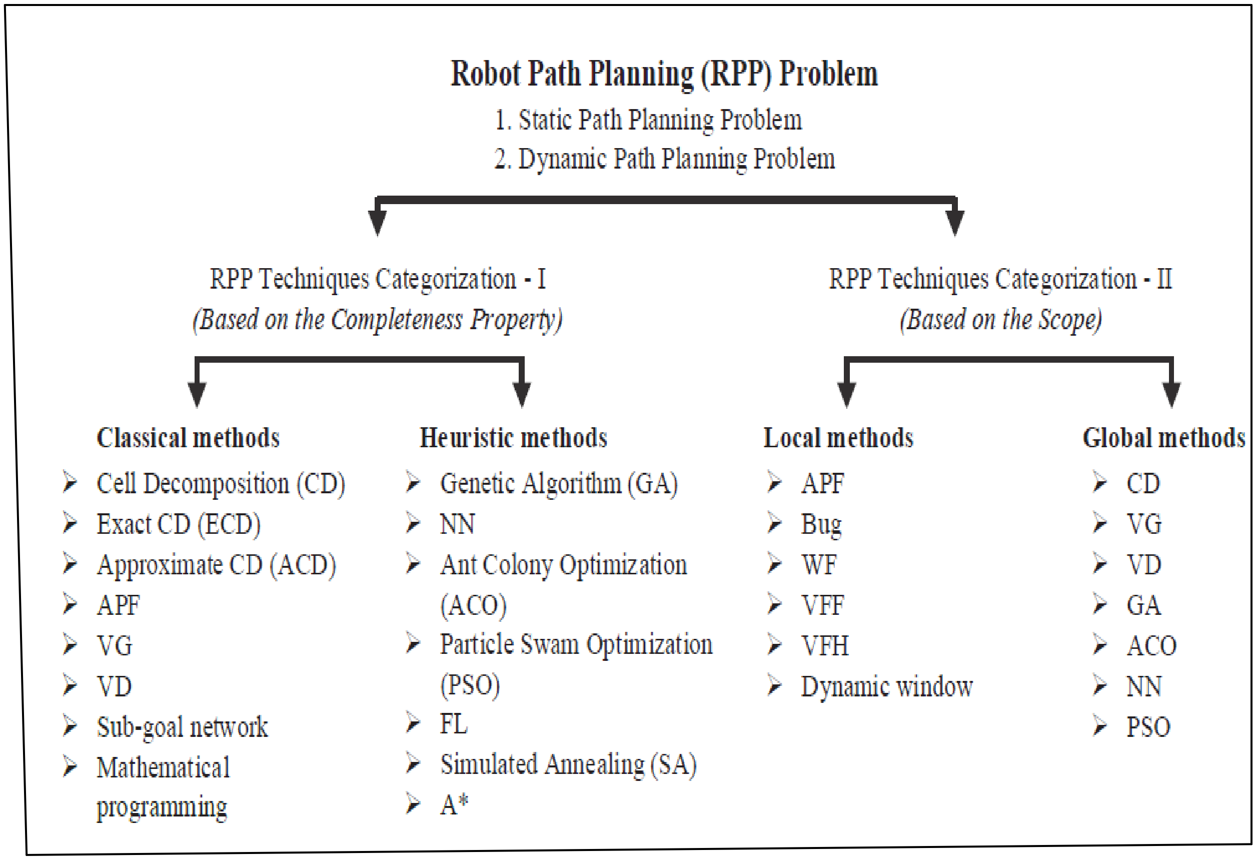

**Fig.2.1.** Taxonomy of mobile robot path planning algorithms

## 2.3 IMPORTANCE OF MOBILE ROBOT PATH PLANNING

Path planning and obstacle avoidance for mobile robot navigation are challenging topics because depending on environmental condition we have to choose different algorithm due to advantage and limitation of the algorithm. You have to decide which algorithm is suitable for this particular work. An accurate path planning algorithm is important for many reasons

i. To get a collision free path for a mobile robot path planning of mobile robot is necessary.
ii. A corrected and efficient path gives the shortest possible path that means it reduce the total functional time.
iii. A well defined path planning reduces idle time.
iv. For a cost effective mobile robot navigation an accurate path planning is necessary.
v. An appropriate path planning increase safety.
vi. For a specific work position and orientation of mobile robot is necessary , for which path planning is required.

# CHAPTER 3

# PATH PLANNING ALGORITHM USING ARTIFICIAL POTENTIAL FIELD METHOD

## 3.1 INTRODUCTION

The goal of the path planning method is to determine a sequence of configurations for the robot to move around obstacles and avoid collisions while reaching a desired goal .The artificial potential field (APF) method is widely used for planning the path of mobile robot. In the artificial potential field method, a mobile robot is considered to be subjected to an artificial potential force. The potential force has two forces: first one is attractive force and second one is repulsive force. In the artificial potential field method, we can imagine that all obstacles can generate repulsive force to the robot that is inversely proportional to the distance from the robot to obstacles and is pointing away from obstacles, while the destination or goal has attractive force that attracts robot to the goal. The combination of these two forces will generate a total force with magnitude and direction, the mobile robot should follow that direction to avoid obstacles and reach to the target in a safe path .Actually the artificial potential field method uses a scalar function called the potential function .This function has two values, a minimum value, when the robot is at the goal point and a high value on obstacles. The function slopes down towards the target point, so that the robot can reach the target by following the negative gradient of the total potential field.

The artificial potential field method proposed by Andrews, Hogan and Khatib has gained increased popularity in the field of mobile robot path planning. In these approaches, the target exerts an imaginary attractive force on the robot, while the obstacles apply repulsive forces to the robot. The total resultant force determines the subsequent direction and speed of travel. With the characteristic of simplicity, the traditional potential field method can be implemented conveniently due to its high efficiency.

## 3.2 ALGORITHM

Assume that the robot is of point mass and position of the robot moves in a two-dimensional (2-D) workspace. The position of mobile robot in the workspace is denoted by (x, y) position of obstacle = ($x_{0,}$ $y_0$) and position of goal = ($x_G$, $y_G$).

There are two virtual force acting on the mobile robot one is attractive force ,which attract the robot to the goal and other is repulsive force which repulses the robot to the obstacle for avoiding obstacles.[36]

Robot will move by the resultant of the two forces. If $F_{att}$ is the attractive force and $F_{rep}$ is the repulsive force , then the driving force

$F = F_{att} + F_{rep}$

Here attractive potential field creates attractive potential field and repulsive potential field creates repulsive potential field

### 3.2.1 Attractive Potential Field

The most commonly used form of potential field functions proposed by Kathib is defined as a parabolic form. Attractive potential field grows quadratically with the distance to the goal.

The output of the Seek Goal behavior is a vector that points the robot toward the goal. If I took the robot on a pretend journey through every point in a two-dimensional world and recorded the output vector at each point, the collection of these vectors would look something like the diagram illustrated in Figure 3.1. This collection of vectors is called a potential field because it represents synthetic energy potentials that the robot will follow. The potential field associated with the Seek Goal behavior is an example of an attractive potential because the field causes the robot to be attracted to the goal (i.e., all vectors point to the goal).

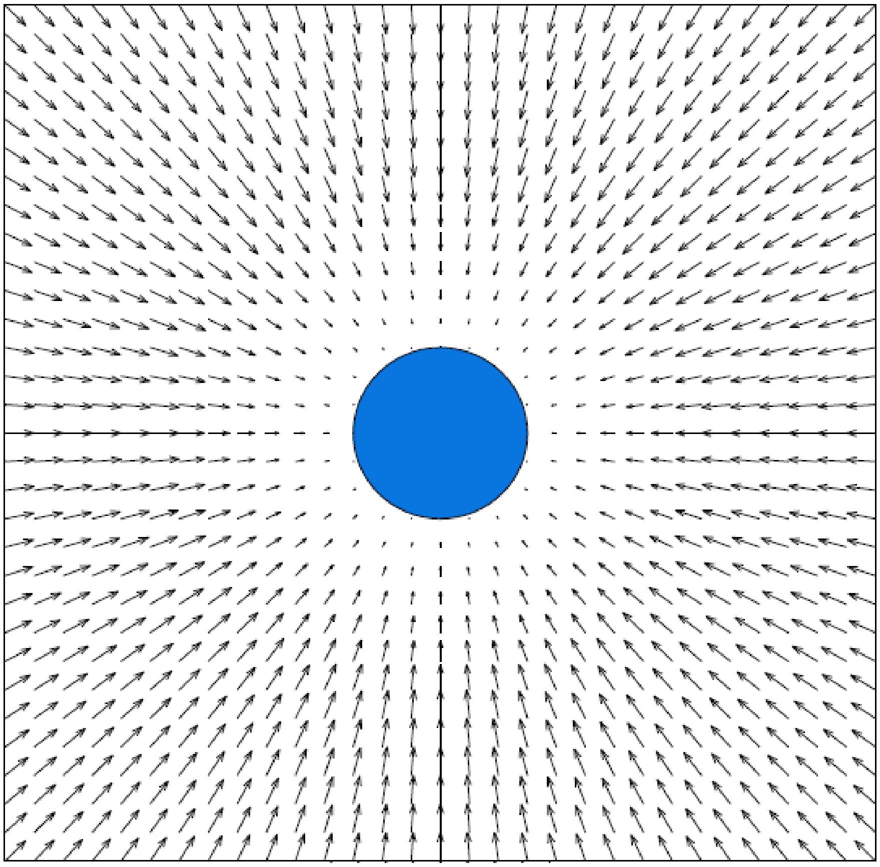

**Fig. 3.1:** An attractive potential field, corresponding to the Seek Goal behavior

Let $R_G$ denote the radius of the goal and d is the distance between goal and robot then

$$d = \sqrt{(x_G - x)^2 + (y - y_G)^2}.$$

And angle between x-axis and line connected to mobile robot with the goal

$$\theta = \tan^{-1}\left(\frac{y_G - y}{x_G - x}\right)$$

Set $\Delta x$ and $\Delta y$ according to the following:

| | |
|---|---|
| If $d < R_G$ | $\Delta x = \Delta y = 0$ |
| If $R_G \leq d \leq s + R_G$ | $\Delta x = a\,(d - R_G)\cos\Theta$ and $\Delta y = a\,(d - R_G)\sin\Theta$ |
| If $d > s + R_G$ | $\Delta x = as\,(d - R_G)\cos\Theta$ and $\Delta y = as\,(d - R_G)\sin\Theta$ |

This sets up a goal as a circle with radius $R_G$. When the agent reaches the goal no forces from the goal act upon it, whence when d < RG $_{\text{both}}$ $\Delta$x and $\Delta$y are set to zero. The field has a spread of S and the agent reaches the extent of this field when $d = S + R_G$ . Outside of this circle of extent, the vector magnitude is set to the maximum possible value. Within this circle of extent but outside of the goal's radius, the vector magnitude is set proportional to the distance between the agent and the goal. I include the constant $a > 0$ so that the strength of the field can be easily scaled.

### 3.2.2 Repulsive Potential Field

The repulsive force is inversely proportional to the distance from the obstacle. The repulsive potential results from the combination of the repulsive forces of all the obstacles.

$U_{rep} = \sum_i U_{rep}$

Where $U_{rep}$ represents the repulsive potential generated by obstacle i, that influence the environment of robot

Let $R_O$ denote the radius of the obstacle and $d_O$ is the distance between goal and robot then

$d_O = \sqrt{(\ (x_G - x_O)^2 + (y_G - y_O)^2)}$ and angle between x-axis and line connected to mobile robot with the obstacle is $\alpha$

Set $\Delta$x and $\Delta$ y according to the following:

If $d_O < R_O$ $\Delta x = -\text{sign}(\cos\alpha)\infty$ and $\Delta y = -\text{sign}(\sin\alpha)\infty$

If $R_O \leq d_O \leq S + R_O$ $\Delta x = -\text{sign}\ b\,(s + R_O - d_O)\cos\alpha$ and $\Delta y = b(S + R_O - d_O)\sin\alpha$

If $d_O > S + R_O$ $\Delta x = \Delta y = 0$

Within the obstacle, the repulsive potential field is infinite and points out from the center of the obstacle (the sign function returns the sign of an argument). Outside of the circle of influence, the repulsive potential field is zero. Within the circle of influence but outside the radius of the obstacle, the magnitude of the vector grows from zero (when $b(s + R_O - d_O) = 0$ when $d_O = s + R_O$ corresponding to the agent being on the edge of the circle of influence) to b(s).The constant $b > 0$ is given to allow the agent to scale the strength of this field. Notice that the vector points away from the obstacle; this done by introducing the negative sign into the definitions of $\Delta$ x and $\Delta$ y.

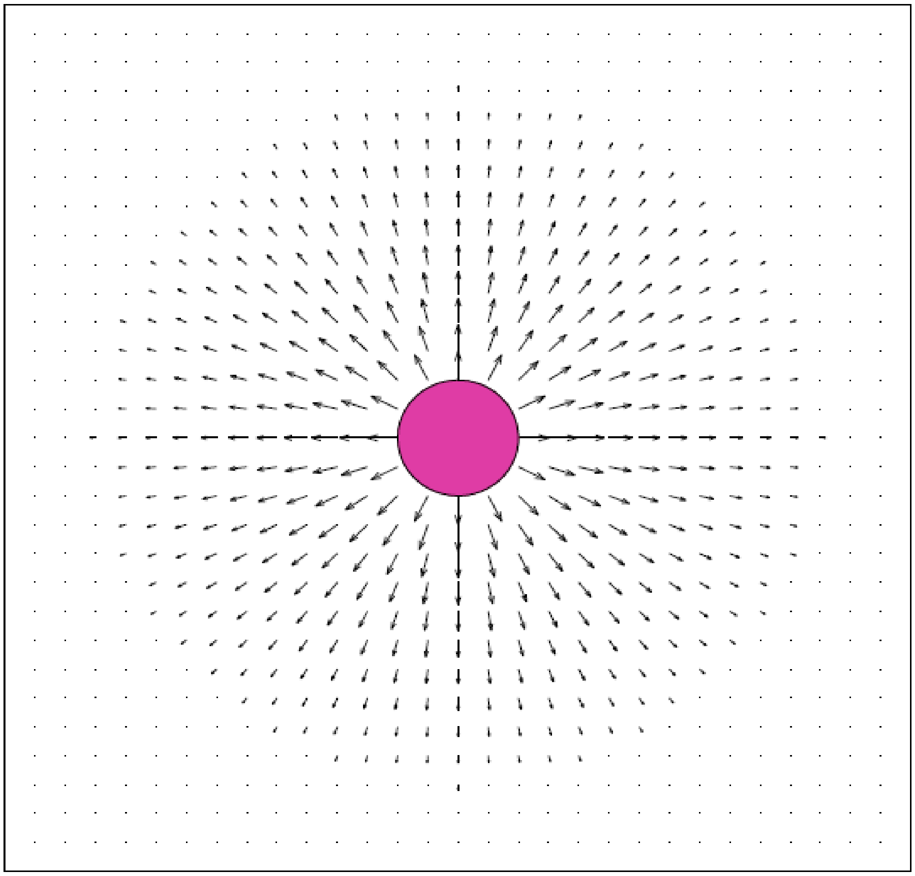

**Fig. 3.2:** A reject potential field, corresponding to the Avoid Obstacle behavior.

### 3.2.3 Total potential field

Total potential field is $U_{\text{att}} + U_{rep.}$

Since many problems will have both goals and obstacles , we have to figure out how to combine these potential fields. We combine multiple potential fields by adding them together. Doing this for our two-behavior robot in a world with both an obstacle and a goal gives the new potential field diagrammed in Figure .This field was generated by first finding $\Delta_G x$ and $\Delta_G y$ the vector generated by the attractive goal, second finding $\Delta_0 x$ and $\Delta_0 y$ , the vector generated by the repulsive obstacle, and third adding these vectors together to find

$\Delta x = \Delta_G x + \Delta_0 y$ and $\Delta y = \Delta_G y + \Delta_0 y$

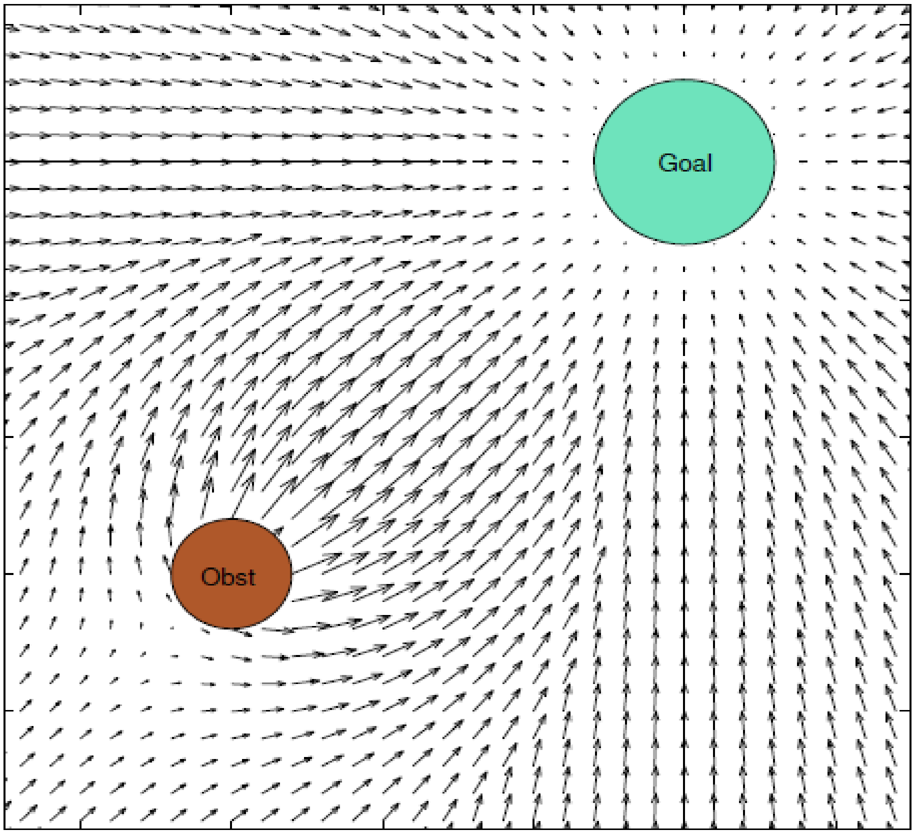

**Fig.3.3:** The combined potential field when there is a goal and an obstacle.

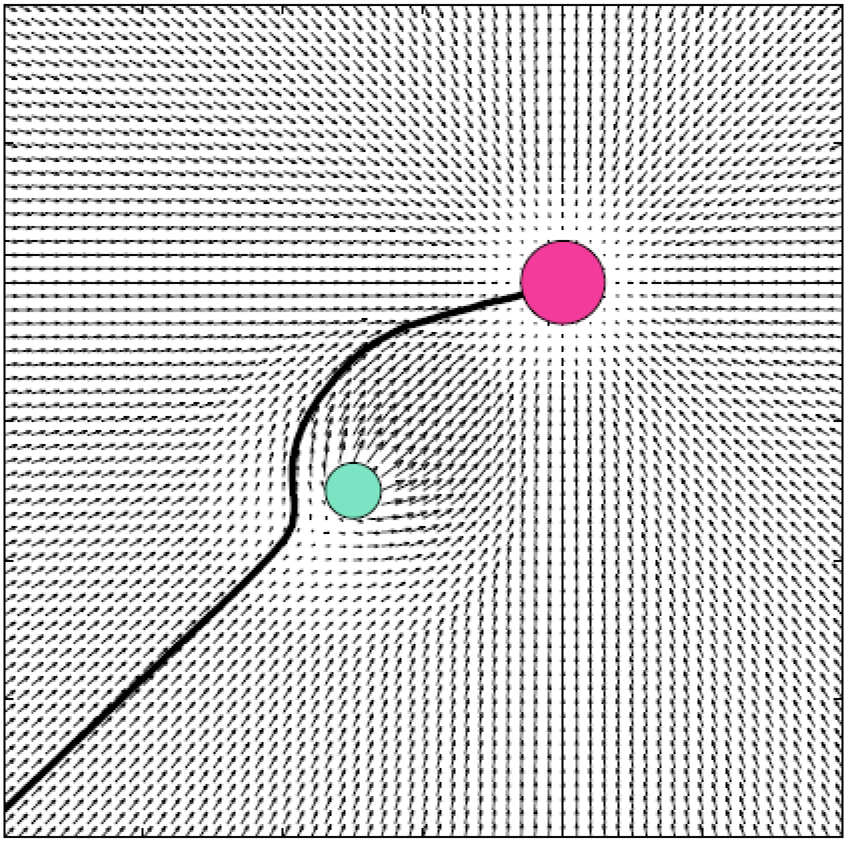

**Fig.3.4** : The trajectory experienced by our two-behavior robot when there is a goal and an obstacle.

## 3.3 ADVANTAGE AND LIMITATIONS

APF method has some advantages, it is very simple so easy to understand and you can improved it easily. It is real time based algorithm that means you can use this algorithm in an environment where obstacles are unknown. It is reliable, means you will get proper output.

In spite of several advantages, there are limitations of APF method,[16] they are-

1. Trap situations due to local minima (cyclic behavior).
2. No passage between closely spaced obstacles.
3. Oscillations in the presence of obstacles.
4. Oscillations in narrow passages.

### 3.3.1. Trap situations due to local minima (cyclic behavior)

The best-known and most often-cited problem with APF is the problem of *local minima* or *trap-situations*. It is the formation of local minima that can trap the robot before reaching its goal. A *trap-situation* may occur when the robot runs into a dead end (e.g., inside a U-shaped obstacle).This occurs when the total force acting on the agent is summed up to zero although robot has not reached its goal position yet. Traps can be created by a variety of different obstacle configurations, and different types of traps can be distinguished. The local minimum problem is sometimes inevitable in the local path planning because the robot only can detect local information's of obstacles. In other words, the robot cannot predict local minimums before experiencing the environments. The avoidance of local minimum has been an active research topic in the APF based path planning. However, trap-situations can be resolved by heuristic or global recovery.

### 3.3.2. No passage between closely spaced obstacles

During the attempt of a mobile robot to pass among two closely spaced obstacles (e.g., passing through a door frame), the repulsive forces from the two obstacles are combined into the two *lumped repulsive forces*, respectively. The sum of all repulsive forces points straight away from the opening between the two obstacles. Depending on the relative magnitude of the target-directed force, the robot will either approach the opening further, or it will turn away.

### 3.3.3. Oscillations in the presence of obstacles

Total force acting on the robot is calculated by summing up all the force components of the environmental effects. However this situation causes some oscillations in the motion of the robot. This can be handled by assigning some distance criteria for the obstacles which may indicate attraction distance. However more intelligent solution is checking the visibility of those obstacles while agent traversing in its own path.

### 3.3.4. Oscillations in narrow passages

A similar yet more severe problem with artificial potential field methods occurs when the robot travels in narrow corridors, in which the robot experiences repulsive forces simultaneously from opposite sides.

Inappropriate definitions of the potential field equations will produce local minima of potential fields. As a result, the robot might be trapped into local situations. For example, it oscillates in the presence of obstacles and swings in narrow passages. Thus, the artificial potential field method requires to be associated with some other artificial intelligence optimization algorithms, such a genetic algorithms, fuzzy and artificial neural networks, etc.

# CHAPTER 4

# HARDWARE AND SOFTWARE OF THE PROJECT

### 4.1 PARALLAX BOE-BOT MOBILE ROBOT KIT

Parallax Boe–Bot (short for Board of Education Robot) consists of a main circuit board (the Board of Education), a plug–in microcontroller, two servo motors to drive the wheels, a bread board and an aluminum chassis that the parts bolt onto.

The green detachable main circuit, mounted on the top of the robot is called the Board of Education. The microcontroller which plugs into a socket on the green circuit board is called the BASIC Stamp . The BASIC Stamp is programmed in PBASIC. The rear wheel is a drilled polyethylene ball held in place with a cotter pin. Wheels are machined to fit on the servo spline and held in place with a screw. These servomotors are connected with the P14 and P15 pins on the Board of Education carrier board. To load the program onto microcontroller, bread board is connected to debug terminal using serial interface. The BASIC Stamp is easy to program. The parallax Boe–Bot is small, approximately four inches wide, and runs on four AA batteries. At the front of the chassis, there is a gripper installed for gripping the objects. Separate servomotor is used to operate the gripper. Pin P13 is connected with this servomotor.[39]

The Boe–Bot can be adjusted to walk on six legs, sense objects, and also can attach With the PING ultrasonic range sensor.

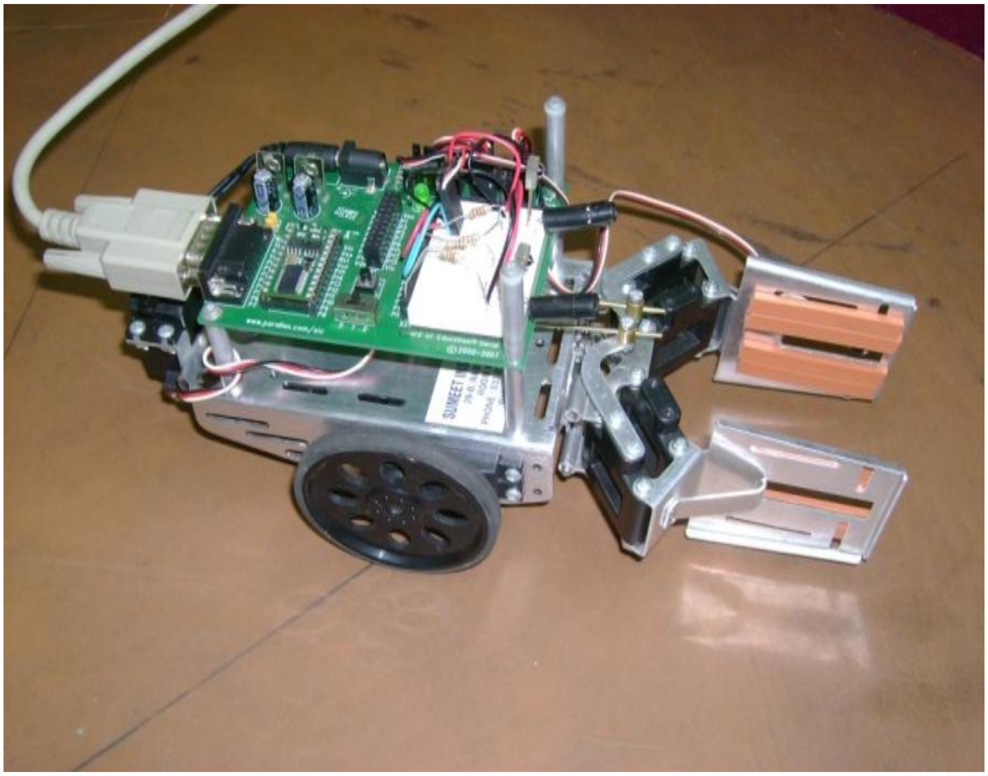

**Fig.4.1**: photographic view of boe-bot with gripper

## 4.2 BOARD OF EDUCATION CARRIER BOARD

The Board of Education carrier board along with the BASIC stamp 2(microcontroller) and other electronic components is shown in fig.. It is the main board that contains all the electronic components including the microcontroller, bread board and necessary parts for connecting servomotors and various sensors. Board of Education carrier board power requirement is 6 to 9 VDC and operating temperature is +32 to +185 °F (0 to +70 °C) .

The BASIC Stamp module plugs into the Board of Education carrier board. The Board of Education makes it easy to connect a power supply and serial cable to the BASIC Stamp module. It also makes it easy to build circuits and connect them to the BASIC Stamp.

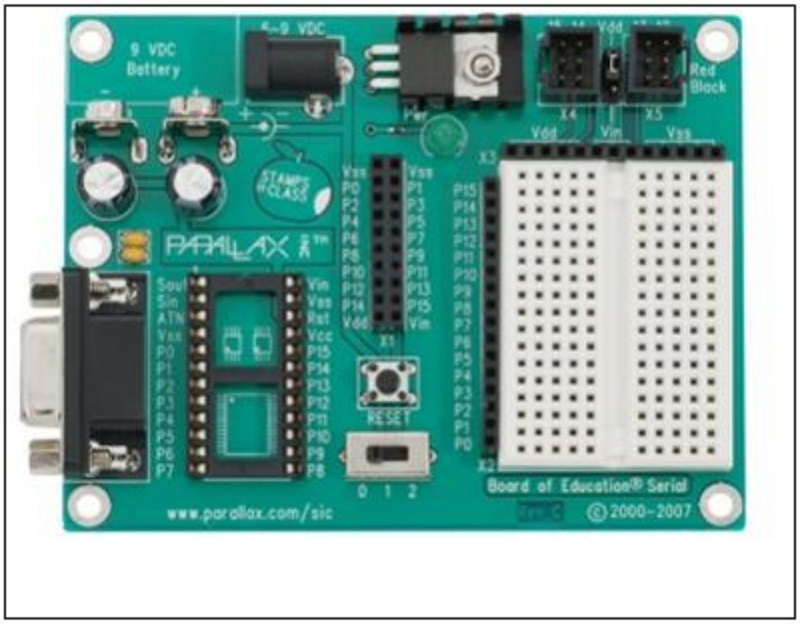

**Fig.4.2:** board of education carrier board

## 4.3 BASIC STAMP 2 MICROCONTROLLER MODULE

The BASIC Stamp 2 microcontroller module serves as the brains inside of electronics projects and applications that require a programmable microcontroller. It is able to control and monitor timers, keypads, motors, sensors, switches, relays, lights, and more. Programming is performed in an easy-to-learn language called PBASIC. Its small form factor requires very little space and non-volatile memory holds up to 500 instructions even without power.

The specifications of BASIC stamp 2 are shown below:

| | |
|---|---|
| Microcontroller | Microchip PIC16C57c |
| Processor Speed | 20 MHz |
| Program Execution Speed | 4000 instructions/sec (app) |
| RAM Size | 32 Bytes (6 I/O, 26 Variables) |
| EEPROM (Program) Size | 2K Bytes, ~500 instructions |
| Number of I/O Pins | 16 + 2 Dedicated Serial |
| Voltage Requirements | 5- 15 V (DC) |
| Current Draw@5 Volts | 3 mA Run, 50 µA Sleep |
| PC Interface | Serial (9600 baud) |

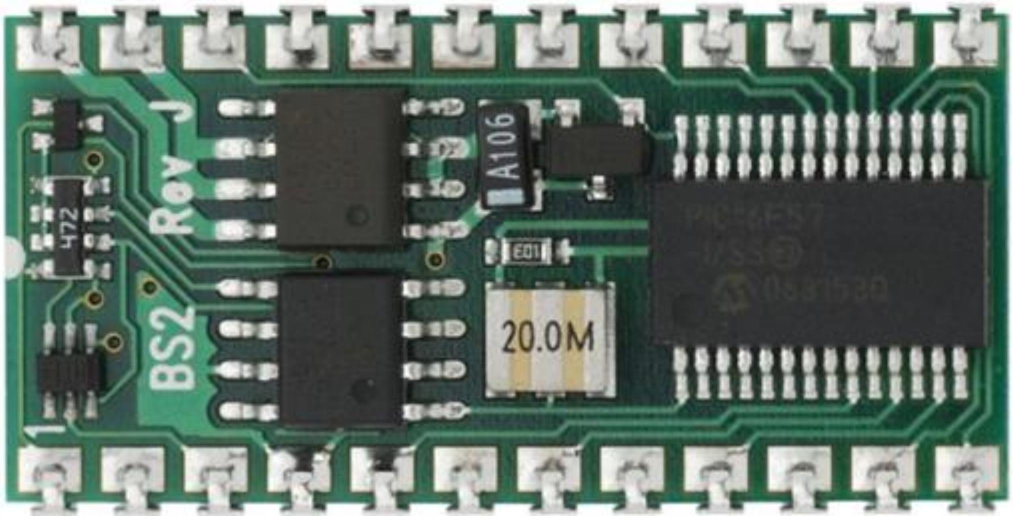

**Fig.4.3** : microcontroller (basic stamp 2)

## 4.4 PARALLAX SERVO MOTOR

The Parallax continuous rotation servo motor shown in fig 3.3 is ideal for robotic products that need a geared wheel drive or a 360 degree rotation geared motor. The Parallax continuous rotation servo output gear shaft is a standard Futaba configuration. Continuous rotation servos receive the same electronic signals, but instead of holding certain positions, they turn at certain speeds and directions. BASIC stamp send the same message over and over again to control the servo motor speed and direction. The servo motors for the wheels are connected to pin numbers 14 and 15 (P14 and P15) and a separate servo motor used to actuate the gripper is connected to pin number 13 (P13).

The specifications of Parallax servo motor are shown below:

- Power Requirements: 4 to 6 VDC
- Communication: Pulse-width modulation
- Dimensions: 2.2 x 0.8 x 1.6 in (55.8x 19 x 406 mm) excluding servo horn
- Operating temp range: +14 to +144 °F (-10 to +50 °C)
- Torque: 38 oz-in @ 6 V

Servos are also centred by running a of program run which sends a signal instructing them stay still. . The instruction consists of a series of 1.5 ms pulses with 20 ms pauses between each pulse. As the servos are not pre-adjusted at the factory, they will instead start turning. A screw driver is used to adjust them so they stay still. In BOE-BOT, after the servo has been properly adjusted, centre signal instruct it to stay still. This process is called centering the servos. Once the centring of the servo is done, a pulse width of 1.5ms causes the servos to stand still. This is done using a PULSOUT command with duration of 750. It’s best to only centre one servo at a time, because that way we can hear when the motor stops as we are adjusting it.

The Parallax continuous rotation servo rotates full speed clockwise, when a 1.3 ms pulse sent to it, as shown in fig 3.5. Full speed range from 50-60 rpm. Now a 1.3 ms pulse requires a PULSOUT command Duration argument of 650. All pulse widths less than 1.5 ms, and therefore PULSOUT Duration arguments less than 750, will cause the servo to rotate clockwise.

For rotating the servo full speed counter clockwise, when a 1.3 ms pulse sent to it. Now a 1.7 ms pulse requires a PULSOUT command Duration argument of 850. All pulse widths greater than 1.5 ms, and therefore PULSOUT Duration arguments greater than 750, will cause the servo to rotate counter clockwise.

**Fig.4.4** : parallax servo motor

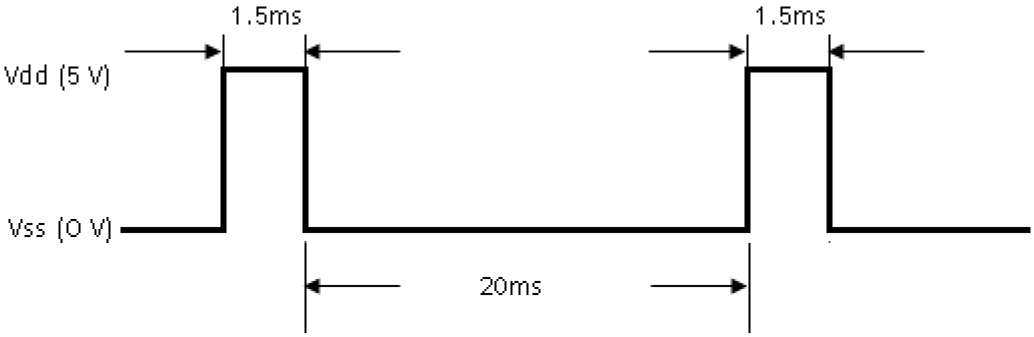

**Fig. 4.5:** pulse train for making the servo motor stand still

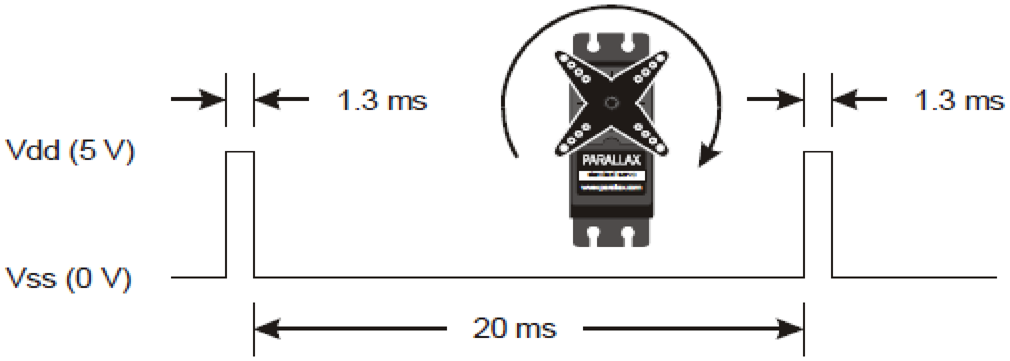

**Fig. 4.6:** pulse train for turning the servo motor full speed clockwise

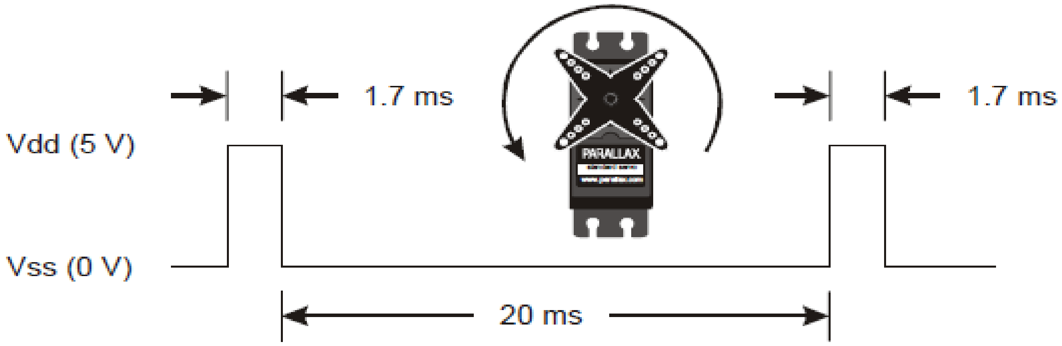

**Fig. 4.7:** pulse train for turning the servo motor full speed counterclockwise

### 4.5 BOE-BOT SYSTEM SOFTWARE

The System software that has been used for development and execution of the necessary application program in PBASIC to run on the microcontroller of the Boe-Bot system is BASIC Stamp Editor (version 2.0). This Editor program has been run on a separate computer system (PC or laptop), which is connected to the microcontroller through serial interface.

#### 4.5.1 BASIC Stamp Windows Editor (version 2.0):

It is the robot control software package. This software allows us to write programs in PBASIC on computer and download them into the Boe–Bot’s BASIC stamp brain (microcontroller). BASIC Stamp Editor is free software; it can be downloaded from the internet or from the Parallax CD. A program entered in the BASIC stamp Editor Window, some lines of the program is made automatically by clicking buttons on the toolbar. Other lines are made by typing them in the keyboard. After saving the program with a file name, if it is run, a download program window will appear briefly as the program is transmitted from the PC to BASIC stamp. The Debug terminal will appear on the computer monitor when the download is completed. Debug terminal shows the message which is sent by the BASIC stamp to PC. Every time by pressing and releasing the Reset button on Board of education, the program will rerun and every time a message will be displayed in the Debug terminal.

#### 4.5.2 PBASIC Version 2.5:

It is the programming language which is used to accomplish different tasks performed by the BASIC stamp and Boe-Bot.

PBASIC stands for –

- Parallax – Company that invented and makes BASIC stamp microcontrollers.
- Beginners – Made for beginners to use learn how to program computer.
- All purpose – Powerful and useful for solving many different kinds of problems.
- Symbolic – Using symbols ( terms that resemble English word / phrases )
- Instruction – To instruct a computer how to solve problem.

Code – In terms that one and one’s computer understand.

### 4.5.3 Types of Variables and Defining Variables

Variables are used in PBASIC programs to store values. Variables are defined by its name and specify its size. This is called declaring a variable. The most important thing about being able to store values is that the program can use them to count. When the program can count, it can both control and keep track of the number of times something happens. One can declare sizes of variable in PBASIC as follows:[38]

Syntax: **variable-name VAR size**

There are four different sizes of variables in PBASIC:

| Size | Stores |
|---|---|
| Bit | 0 to 1 |
| Nib | 0 to 15 |
| Byte | 0 to 255 |
| Word | 0 to 65535 or -32768 to +32767 |

**Description of main commands used in the project:**

**1. DEBUG**

Syntax: **DEBUG Output Data**

- Function: Display information on the PC screen within the BASIC Stamp editor program. This command can be used to display text or numbers in various formats on the PC screen in order to follow program flow (called debugging).

**2. DEBUGIN**

Syntax: **DEBUGIN Input Data**

- Function: Accept information from the user via the Debug Terminal within the BASIC Stamp Editor program.

**3. PULSOUT**

Syntax: **PULSOUT Pin, Duration**

- Function: Generates a pulse on Pin with a width of duration.
- Pin is a variable/ constant / expression (0-15) that specifies which I/O pin to set low. The pin will be placed into output mode.
- Duration is a variable/ constant/expression (0-65535) that specifies the duration of the pulse. The unit of time for duration is two microsecond.

This command has been used to rotate the servomotors of wheels and gripper in the required direction for necessary time.

**4. PULSIN**

Syntax**: PULSIN Pin, State, Variable**

- Function: Measure the width of a pulse on Pin described by State and stores the result in Variable.

**5. PAUSE**

Syntax: **PAUSE Duration**

- Function: Pause the program (do nothing) for the specified duration
- .Duration is a variable/ constant/expression that specifies the duration of the pause. The unit of time for duration is one millisecond.

**6. DO-----LOOP**

Syntax: **DO (WHILE/ UNTIL conditions) Statement(s)**
**LOOP (WHILE /UNTIL conditions)**

- Function: Create a repeating loop that executes the program lines between DO and Loop, optionally testing before or after the loop statements
- Condition is an optional variable / constant / expression (0 – 65535) which determines whether the loop will run or terminate.
- Statement is any valid PBASIC statement.

**7. FOR----- NEXT**

Syntax: **FOR Counter = Start value To End value {STEP step value} ---- NEXT**

- Function: Create a repeating loop that executes the program lines between FOR and NEXT, incrementing or decrementing counter according to step value until the value of the counter variable passes the End value.
- Counter is a variable (usually a byte or a word) used as a counter.
- Start value is a variable/ constant/expression (0-65535) that specifies the initial value of the variable (counter).

**8. IF…THEN**

Syntax: IF (condition) THEN…{ELSEIF condition)}…{ELSE}…ENDIF

- Function: In order to make decisions if there are different combinations of conditions.

**9. GOTO**

Syntax: **GOTO Address**

- Function: Go to the point in the program specified by Address. Address is a label that specifies where to go.

**10. GOSUB**

Syntax: **GOSUB Address**

- Function: Store the address of the next instruction after GOSUB, then go to the point in the program specified by address, with the instruction of returning to the stored address. Address is a label that specifies where to go.

**11. END**

Syntax: **END**

- Function: Sleep until the power cycles or the PC connects. Power consumption is reduced to approximately 50 mA.

**12. SIN, COS and ATN FUNCTIONS**

- The Sine operator (SIN) returns the two's complement, 16-bit Sine of an angle specified as an 8-bit (0 to 255) value. The BASIC Stamp SIN operator breaks the circle into 0 to 255 units instead of 0 to 359 degrees. This unit is a binary radian or brad. Each brad is equivalent to 1.406 degrees. And instead of a unit circle, which results in fractional Sine values between 0 and 1, PBASIC Stamp SIN is based on a 127-unit circle. So, at the origin, SIN is 0. At 45 degrees (32 brads), Sine is 90. At 90 degrees (64 brads), Sine is 127. At 180 degrees (128 brads), Sine is 0 again. At 270 degrees (192 brads), Sine is - 127. This is illustrated in fig 4.6.
- The Cosine operator (COS) returns the two's complement, 16-bit Cosine of an angle specified as an 8-bit (0 to 255) value, similar as sine operator.
- The Arctangent operator (ATN) returns the angle to the vector specified by X and Y coordinates values. In the PBASIC Stamp, the angle is returned in binary radians (0 to 255) instead of degrees (0 to 359). Coordinate input values are limited to -127 to 127. This is also illustrated in fig 4.6.

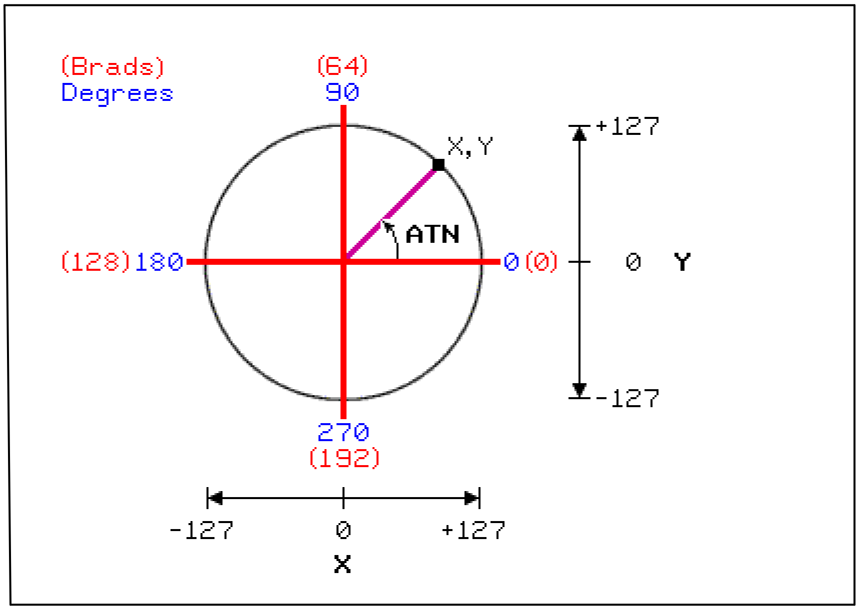

**Fig.4.8 :** sin, cos and atn functions in PBASIC

## 4.6 BOE-BOT NAVIGATION SYSTEM

Forward, backward, rotate left, rotate right and pivoting turns are the basic maneuvers of Boe-Bot which depend upon the direction of rotation of two servomotors. As has been explained earlier a signal with a pulse width of 1.5 ms keeps the servos to stay still. It can be done using a PULSOUT command with duration argument of 750.

### 4.6.1 Moving forward and backward

To move the Boe-Bot forward, it‘s left wheel (driven by a servo motor connected to pin 15) will have to turn counter clockwise and right wheel (driven by a servo motor connected to pin 14) will have to turn clockwise. So following command is used to perform the forward motion at the maximum speed.

```
FOR counter = 1 TO n
PULSOUT 15, 850
PULSOUT 14, 650
PAUSE 20
NEXT
```

To move the Boe-Bot backward, it‘s left wheel will have to turn clockwise and right wheel will have to turn counter clockwise. So following command is used to perform this motion at maximum speed.

```
FOR counter = 1 TO n
PULSOUT 15, 650
PULSOUT 14, 850
PAUSE 20
NEXT
```

In both cases the value of the variable n (i.e. the number of execution of the loop) will determine the time of running the motors and hence the distance traveled by the robot. This has been explained in the system hardware section of the servo motor.

### 4.6.2 Adjusting distance and speed

Distance covered by the Boe-Bot depends upon, how much time, the servo will run. Start value and End value of FOR---- NEXT loop command has been used to control the number of pulses that has been delivered. The End value argument also controls the time the servo will run as each pulse takes the same amount of time. PULSOUT duration argument is changed to control the speed of the servo motors. Arguments of 650 and 850 rotate the servos at maximum speed. To slow down the speed of Boe-Bot, each of the PULSOUT Duration argument has been closer to stay-still value of 750.

The distances traversed for different number of pulses have been measured at maximum speed, and are shown in table 4.1.

**Table 4.1:** Distance traversed against number of pulses sent

| OBSERVATION NO. | DURATUON ARGUMENT | | NO. OF PULSES | DISTANCE (cm) |
|---|---|---|---|---|
| | P15 (left) | P14 right) | | |
| 1 | 850 | 650 | 10 | 5.3 |
| 2 | 850 | 650 | 20 | 8.8 |
| 3 | 850 | 650 | 30 | 12.5 |
| 4 | 850 | 650 | 40 | 16.6 |
| 5 | 850 | 650 | 50 | 20.4 |
| 6 | 850 | 650 | 60 | 24.1 |
| 7 | 850 | 650 | 70 | 28.1 |
| 8 | 850 | 650 | 80 | 32.2 |
| 9 | 850 | 650 | 90 | 36.5 |

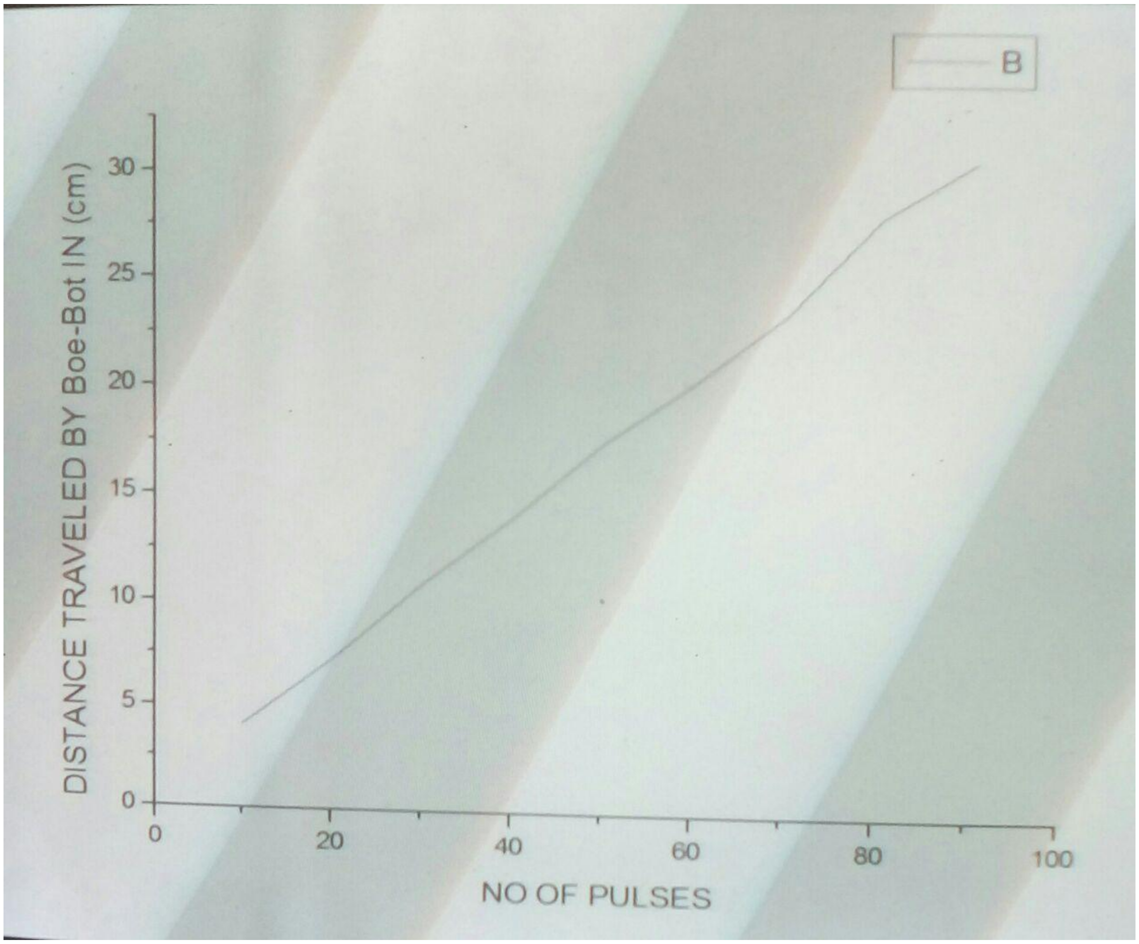

**Fig . 4.9:** Graph between Boe-Bot travel distance and no of pulses

### 4.6.3 Rotation about the centre of mobile robot

For Boe-Bot‘s counter clockwise rotation i.e. for left turn, Boe- Bot‘s right wheel will have to rotate clockwise and left wheel will have to rotate also clockwise. For right turn or clockwise rotation, both wheels will be reversed. So following commands will be used to perform left turn and right turn operations at maximum speed by rotating the robot about its centre.

For left turn (or counter clockwise rotation)

```
FOR counter = 1 TO n
PULSOUT 15, 650
PULSOUT 14, 650
PAUSE 20
NEXT
```

For right turn (or clockwise rotation)

```
FOR counter = 1 TO n
PULSOUT 15, 850
PULSOUT 14,850
PAUSE 20
NEXT
```

In both cases the value of the variable n (i.e. the number of execution of the loop) will determine the time of running the motors, and hence the angle rotated by the robot about its centre. The angles of rotation at different number of pulses have also been measured at maximum speed, and are shown in table 4.2.

**Table 4.2:** angle of rotation against number of pulses sent

| OBSERVATION NO. | DURATUON ARGUMENT | | NO. OF PULSES | ANGLE OF ROTATION (CCW) (Degree) |
|---|---|---|---|---|
| | P15 (left) | P14 (right) | | |
| 1 | 650 | 650 | 10 | 41 |
| 2 | 650 | 650 | 15 | 55 |
| 3 | 650 | 650 | 20 | 80 |
| 4 | 650 | 650 | 25 | 96 |
| 5 | 650 | 650 | 30 | 122 |
| 6 | 650 | 650 | 35 | 146 |

.

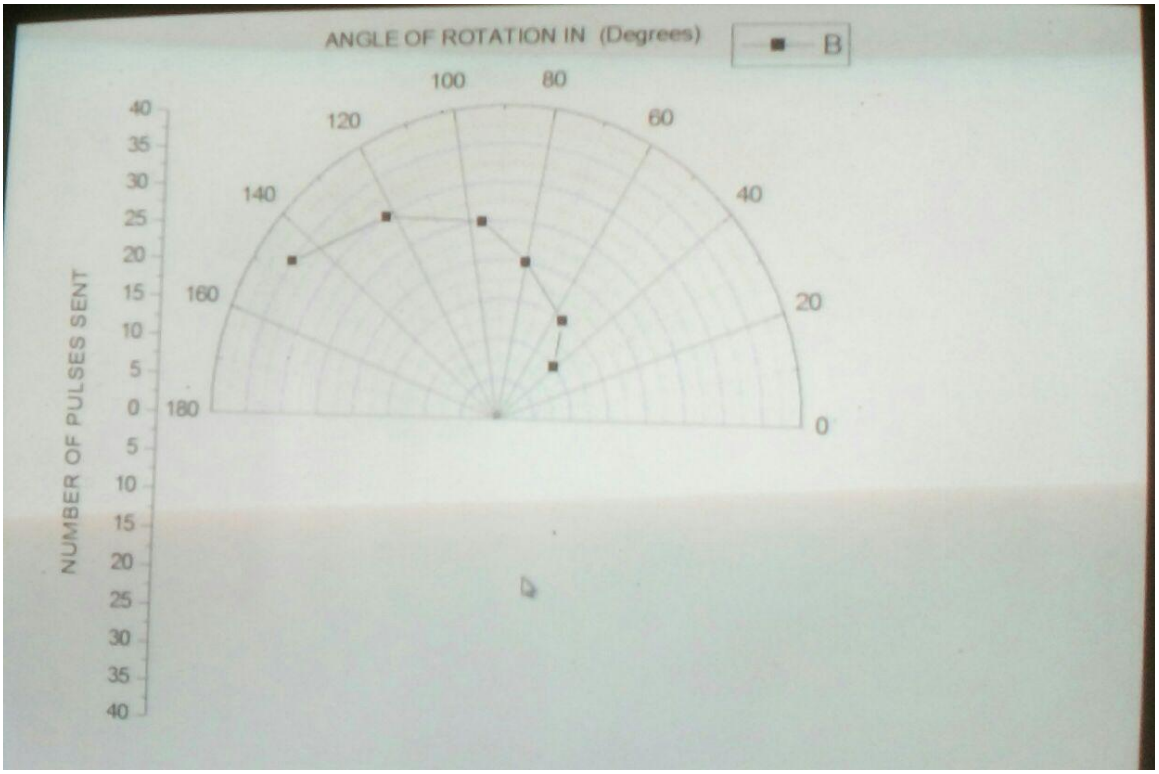

**Fig. 4.10:** Graph between no. of pulses and angle of rotation.

## 4.7 PARALLAX PING ULTRASONIC RANGE SENSOR

The Ping sensor is a device which can be used with the BASIC Stamp to measure how far away an object is. Its range is 3 centimeters to 3.3 meter. It's also remarkably accurate, easily detecting an object's distance down to the half centimeter.

The Ping sensor sends a brief chirp with its ultrasonic speaker (transmitter) and makes it possible for the BASIC Stamp to measure the time it takes the echo to return to its ultrasonic microphone (receiver). The BASIC Stamp starts by sending the Ping sensor a pulse to start the measurement. Then the Ping sensor waits long enough for the BASIC Stamp program to start a PULSIN command. At the same time the Ping sensor chirps its 40 kHz tone, it sends a high signal to the BASIC Stamp. When the Ping sensor detects the echo with its ultrasonic microphone, it changes that high signal back to low. The BASIC Stamp's PULSIN command stores how long the high signal from the Ping sensor lasted in a variable. The measurement is how long it took sound to travel to the object back. The program can calculated the objects distance in centimeter, inch, feet etc. from this time measurement using the speed of sound in air to calculate the object's distance in centimeters, inches, feet, etc.

**Fig.4.11:** ping range sensor

# CHAPTER-5

# IMPLEMENTATION OF PATH PLANNING ALGORITHM WITH THE HELP OF ULTRASONIC RANGE SENSOR

## 5.1 PROGRAM DEVELOPMENT FOR PATH PLANNING OF BOE-BOT MOBILE ROBOT WITH ULTRASONIC RANGE SENSOR USING APF METHOD

In the present project a program has been developed in PBASIC language for path planning of the Boe-Bot mobile robot using Artificial Potential Field Method with the Ping ultrasonic range sensor mounted on the system carrier board. Here artificial potential field method is the main driving force of the mobile robot. For this artificial potential field method information like number of obstacle position and size of the obstacle are required. Here the ultrasonic range sensor gives all the information with respect to the current position of the mobile robot. The PBASIC program has been developed to detect obstacles, to determine the obstacle position and size and to instruct the mobile robot to navigate from a start point to a goal point in presence of obstacles. In the current project a rectangular workspace is considered and the start point has been assumed to be the origin of a co-ordinate system. At the beginning the Boe-Bot mobile robot is assumed to be parallel to the x-axis. The co-ordinates of the goal point are entered as variables during execution of the program.

At first, the mobile robot along with the range sensor scans the whole workspace for obstacles through 90 degree anticlockwise rotation and stores all data regarding position and size of all obstacles using the first part of the program. The rest part of the program instructs the mobile robot to navigate from the start point to the goal point avoiding obstacles.

The ultrasonic range sensor detects obstacle within a wide angle from normal direction. This angular spread has been made narrow by attaching two papers in the form of hollow cylinders to both the ultrasonic transmitter and receiver. Still it detects object with in a small angular spread from normal direction.

### 5.1.1Detection of obstacles by range sensor

Obstacles number, their position and size are unknown and to be determined by range sensor. Fig.5.1 shows workspace consisting obstacle and goal.

**Measurement of number of obstacle**

Number of obstacle are measured by calculating the change of distance from the robot start position to obstacle edges point(points A and B) in the fig. 5.2. Initially robot rotates in anticlockwise direction through $90^0$ and distances corresponding to the obstacle are measured from current position of mobile robot. For a same object distances variation is small. When it crosses the obstacle, variation of distance is large compared to the variation for the same obstacle. Thus numbers of obstacle in the working field are measured.

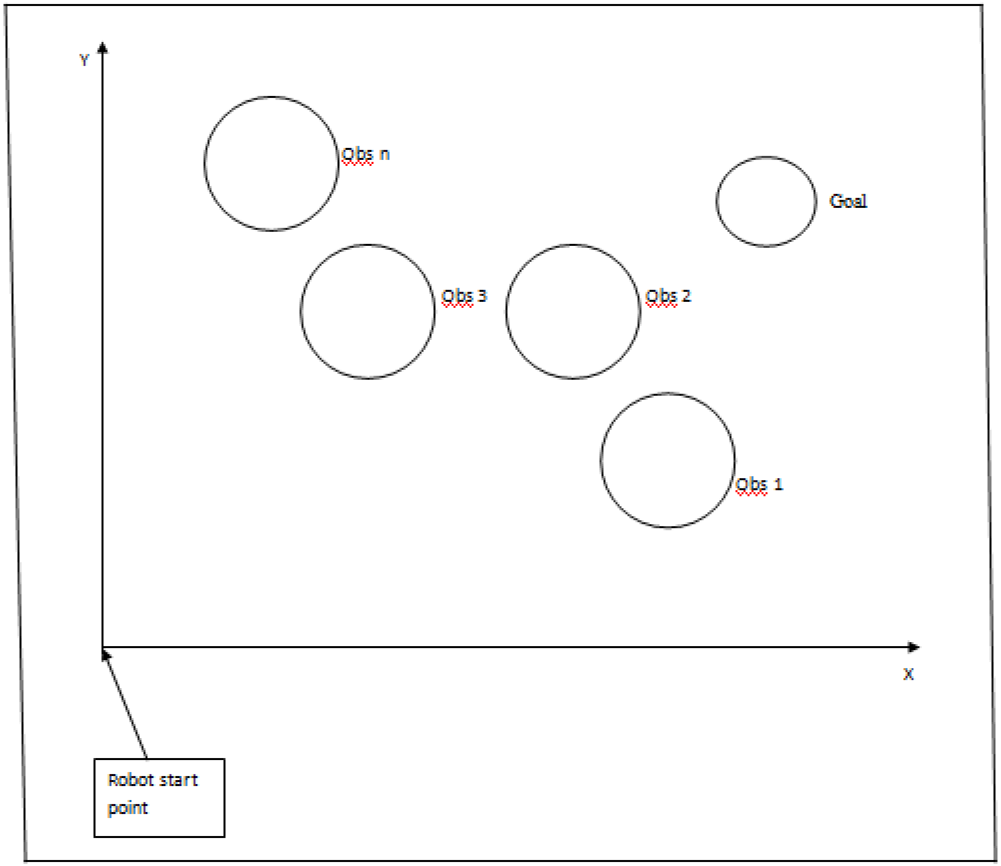

**Fig.5.1:** workspace consisting obstacle and goal.

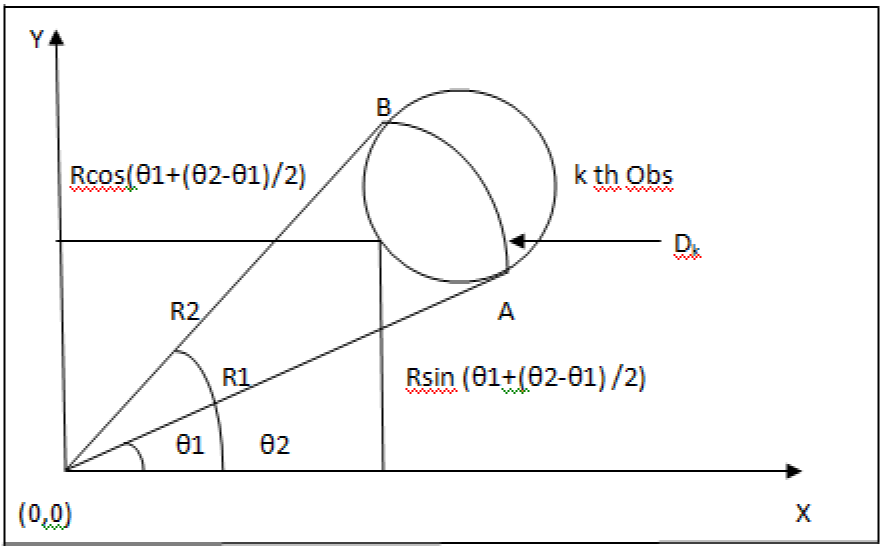

**Fig.5.2**: k th obstacle with co-ordinates

**Determination of position of obstacle**

Position of the k th obstacles is determined by finding approximate co-ordinate of the centre point of obstacle. This is done by the following process.

Let, θ1 be the first edge angle of the k th obstacle and θ2 be the last edge angle of the k th obstacle detected by the range sensor.

Let, R1 be the distance of the first point detected from the mobile robot and R2 be the distance of the last point detected from the mobile robot.

Assumption is that for a obstacle difference between R1 and R2 is very small. So $R1 \approx R2=R_k$

So, co-ordinate of the obstacle

$X_k = R\cos(\theta1+(\theta2-\theta1)/2)$

$Y_k = R\sin(\theta1+(\theta2-\theta1)/2)$

**Determination of the size of the obstacle**

Considering $R_k >> D_k$

Effective Size (approximate dia.) of the obstacle $D_k = R*(\theta2-\theta1)$

Where θ1 and θ2 are in radian.

The complete PBASIC program developed for the path planning of Boe-Bot mobile robot having ultrasonic range sensor has been listed in Table 5.1.

**Table 5.1 PROGRAM FOR OBSTACLE DETECTION AND PATH PLANNING OF MOBILE ROBOT USING ULTRASONIC RANGE SENSOR USING APF METHOD**

```
'  {$STAMP BS2}
'  {$PBASIC 2.5}
' ==============================================================================
 'Program for obstacle detection and path planning of mobile robot
 ‘With the help of ultrasonic range sensor using APF method

 time     VAR Word
 dg       VAR Word
 dist     VAR Word
 xs       VAR Word              ‘start point
 ys       VAR Word
 xg       VAR Word              ‘goal point
 yg       VAR Word
 rg       VAR Byte
 i        VAR Byte
 k        VAR Nib
 th1      VAR Byte(5)
 th2      VAR Byte(5)
 th       VAR Byte(5)
 odist    VAR Byte(5)
 ang      VAR Byte
 Present   VAR Bit
 xo       VAR Byte(5)           ‘obstacle pt
 yo       VAR Byte(5)
```

```
 ro      VAR Byte(5)
xc       VAR Word               ‘current point
yc       VAR Word
angg     VAR Byte               ‘current angle with goal
ango     VAR Byte               ‘current angle with obstaclel
angr     VAR Byte               ‘angle reqd for rotation
a        VAR Byte
b        VAR Byte
s        VAR Byte
ddo      VAR Word
dxg      VAR Word
dyg      VAR Word
dxo      VAR Word
dyo      VAR Word
d        VAR Word
m        VAR Byte
n        VAR Byte
 xs=0
 ys=0
 DEBUG "Enter goal point (xg,yg)",CR
 DEBUGIN DEC xg
 DEBUGIN DEC yg
 DEBUGIN DEC rg
 dg=SQR(((xg-xs)*(xg-xs))+((yg-ys)*(yg-ys)))
 Present=0
 k=1
‘Scan for obstacles & determination of its position & size
FOR i = 1 TO 20                 ‘for 90 degree or 64 Brad
PULSOUT 4,5
PULSIN 4,1,time
dist=time**2251
ang = i*(64/20)
```

```
IF (dist<dg)AND(Present=0) THEN
DEBUG "obstacle start point",CR
Present=1
th1(k)=ang
odist(k)=dist
DEBUG "Start angle=",DEC TH1(k),CR,"Distance=",DEC ODIST(k),CR
PAUSE 1000
 ENDIF
IF (dist>dg)AND (Present=1) THEN
DEBUG "obstacle end point",CR
Present=0
th2(k)=ang
DEBUG "End angle=",DEC TH2(k),CR
PAUSE 1000
th(k)=(th1(k)+th2(k))/2
k=k+1
ENDIF
PAUSE 100
GOSUB  ccw
NEXT
k=k-1
DEBUG "No.of obstacle=",DEC k, CR
FOR i=1 TO k
xo(i)=odist(i)*COS(th(i))/127
yo(i)=odist(i)*SIN(th(i))/127
ro(i)=(odist(i)*(th2(i)-th1(i))*22)/(128*7*2)         'converting the angle from Brad to radian
DEBUG "xo= ",DEC xo(i),"  yo= ",DEC yo(i),"   radius= ",DEC ro(i),CR
NEXT
'Move to initial orientation
FOR  i=1 to 20
GOSUB  cw
```

```
NEXT

‘Calculation for a single step movementx
J=0
xc=xs
yc=ys
angr=0
a=0.05
b=0.5
s-30
LI:
dg=SQR(((xg-xc)*(xg-xc))+((yg-yc)*(yg-yc)))
if (dg<=rg) THEN
GOTO L3
ENDIF
angg=(xg-xc) ATN (yg-yc)                    ‘angg in Brad in all 4 quadrants
dxg=(a*(dg-rg) + 5) * COS(angg)/127
dyg=(a*(dg-rg) + 5) * SIN(angg)/127
dxo=0
dyo=0
FOR  i= TO k
ddo=SQR(((xo(i)-xc)*(xo(i)-xc))+((yo(i)-yc)*(yo(i)-yc)))
IF (ddo>ro(i)+s) THEN
GOTO  L2
ENDIF
IF (ddo<ro(i)) THEN
GOTO L4
ENDIF
ango=(xo(i)-xc) ATN (yo(i)-yc)              ‘angg in Brad in all 4 quadrants
dxo=dxo - (b*(s+ ro(i)-ddo) * COS(ango)/127)
dyo=dyo - (b*(s+ ro(i)-ddo) * SIN(ango)/127)
L2:
```

```
NEXT

dx=dxg+dxo
dy=dyg+dyo
ang = dx ATN dy
angr = ang – angr
IF (angr<0) THEN
angr=angr+256
ENDIF
d=SQR((dx*dx)=(dy*dy))
m=d*65/22                  ‘convert distance into no of pulses
n=angr*20/64               ‘convert angle into Brad
FOR  i=1 to n
GOSUB  ccw
NEXT
FOR  i=1 to m
GOSUB  fm
NEXT
Xc=x+dx
Yc=y+dy
J=j+1
IF (j<=200) THEN
GOTO  L1
ENDIF
DEBUG  “Not Reaching  Goal……….Change a, b”
GOTO  L5
L3:
DEBUG  “Reached  Goal”
GOTO  L5
L4:
DEBUG  “Collision with obstacle……….Change a, b”
```

```
L5:
END
ccw:
'Subroutine for ccw rotation
PULSOUT 15,650
PULSOUT 14,650
PAUSE 20
RETURN
fm:
'Subroutine for forward movements
PULSOUT 15,850
PULSOUT 14,650
PAUSE 20
RETURN
cw:
'Subroutine for cw rotation
PULSOUT 15,850
PULSOUT 14,850
PAUSE 20
RETURN
```

# CHAPTER 6

# RESULTS AND DISCUSSIONS

## RESULTS AND DISCUSSIONS

The part of the developed program in PBASIC has been run successfully for detecting obstacles and for determining the obstacle position and size for different layouts of obstacles and goal points. The photographic view of one such layout of workspace with the robot at the initial position and orientation has been shown in fig. 6.1.

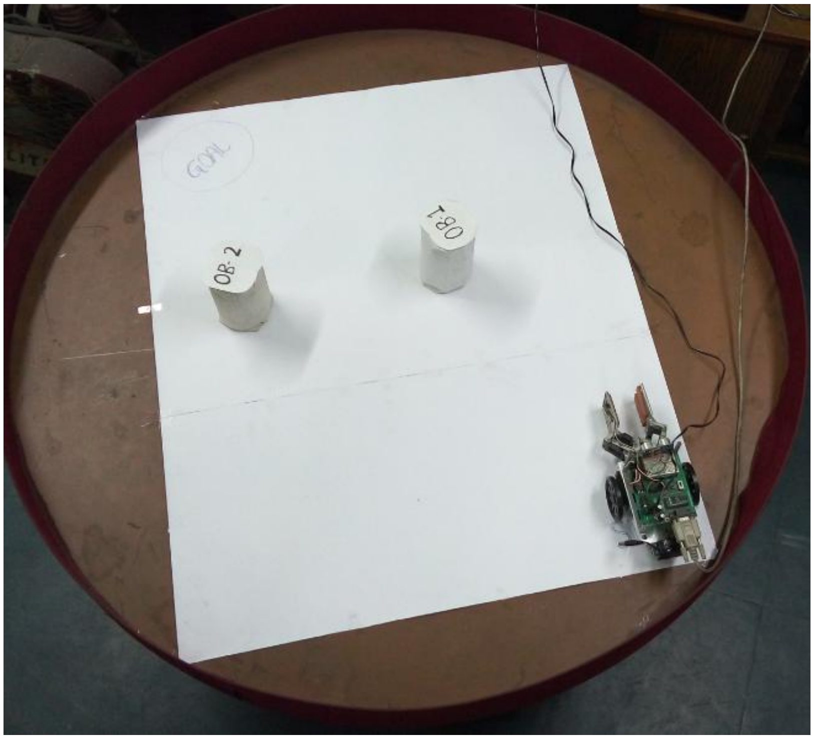

**Fig 6.1:** Starting position of robot

Photographic views of different orientations of the robot along with the range sensors at the instant of edge point detection of different obstacles while scanning the obstacles are shown in the fig. 6.2 to fig 6.6.

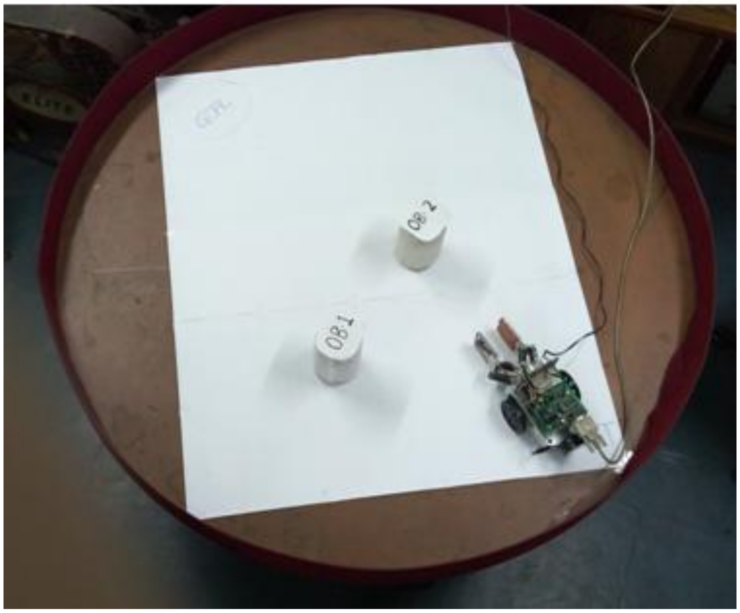

**Fig 6.2:** Detection of first edge of obstacle 1

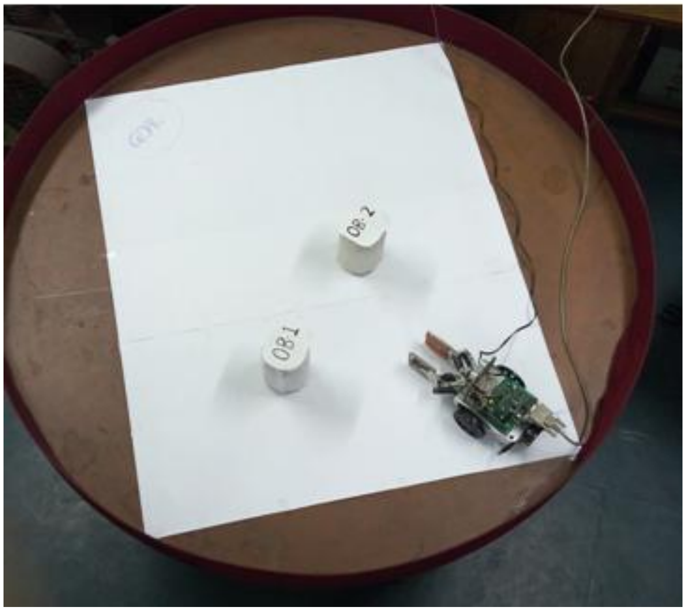

**Fig 6.3:** Detection of last edge of obstacle 1

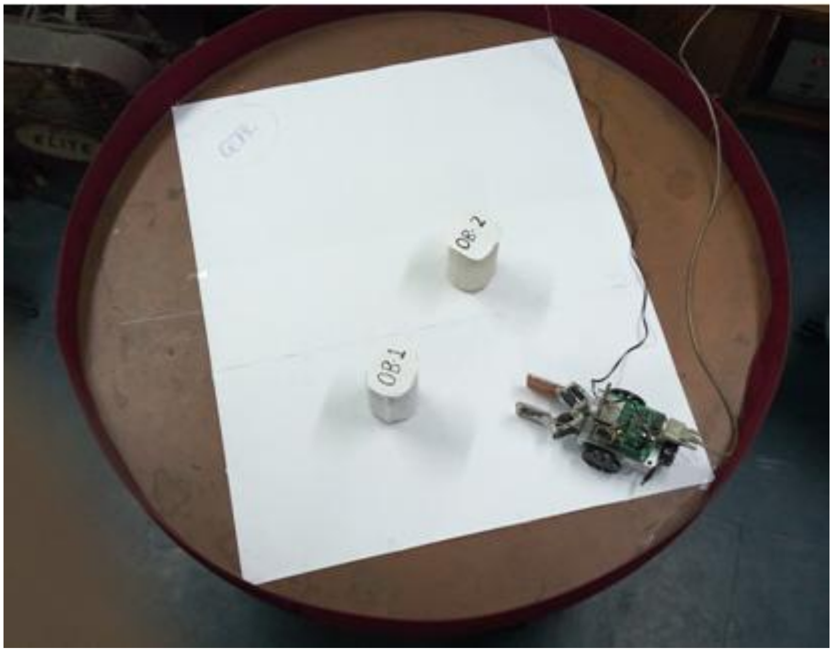

**Fig 6.4:** Detection of first edge of obstacle 2

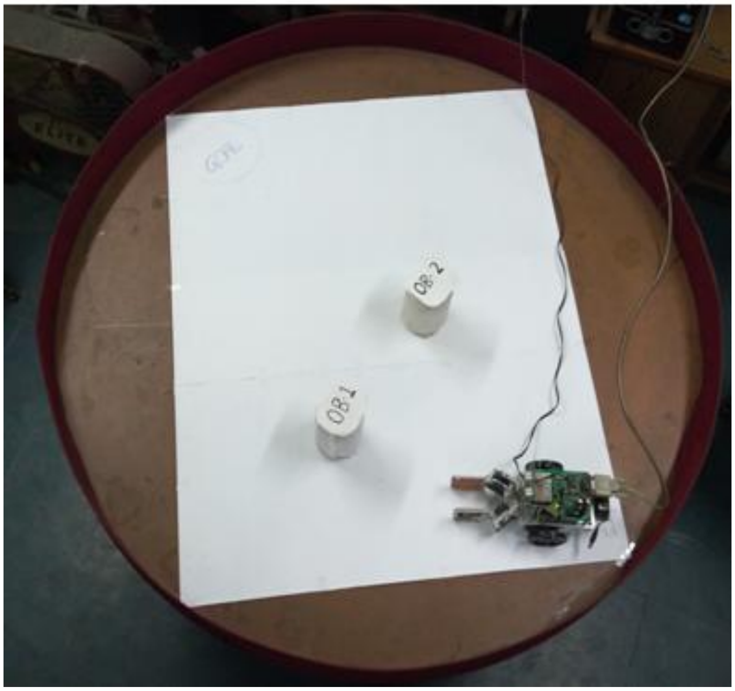

**Fig 6.5:** Detection of last edge of obstacle 2

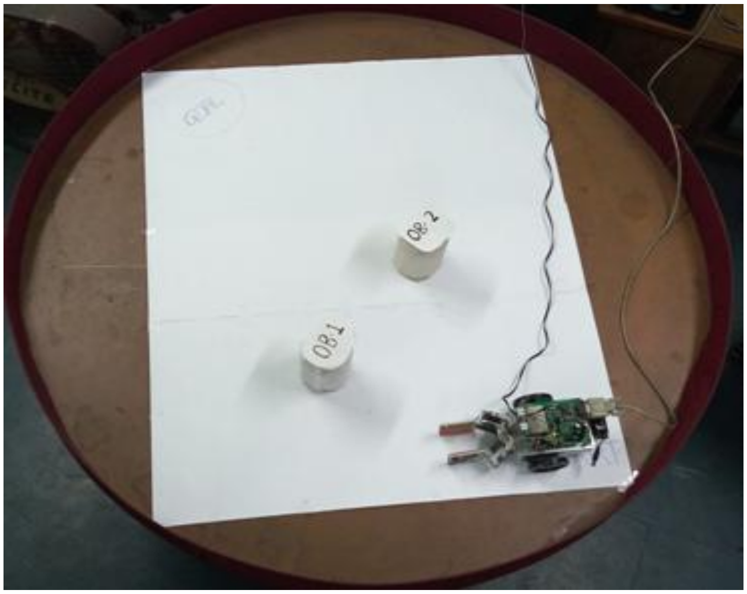

**Fig .6.6:** Completion of $90^0$ rotations

Output from the ultrasonic has been found better by reducing the angular divergence of the receiver of ultrasonic sensor by attaching two papers in the form of hollow cylinders to both the ultrasonic transmitter and receiver, as shown in fig. 6.7.

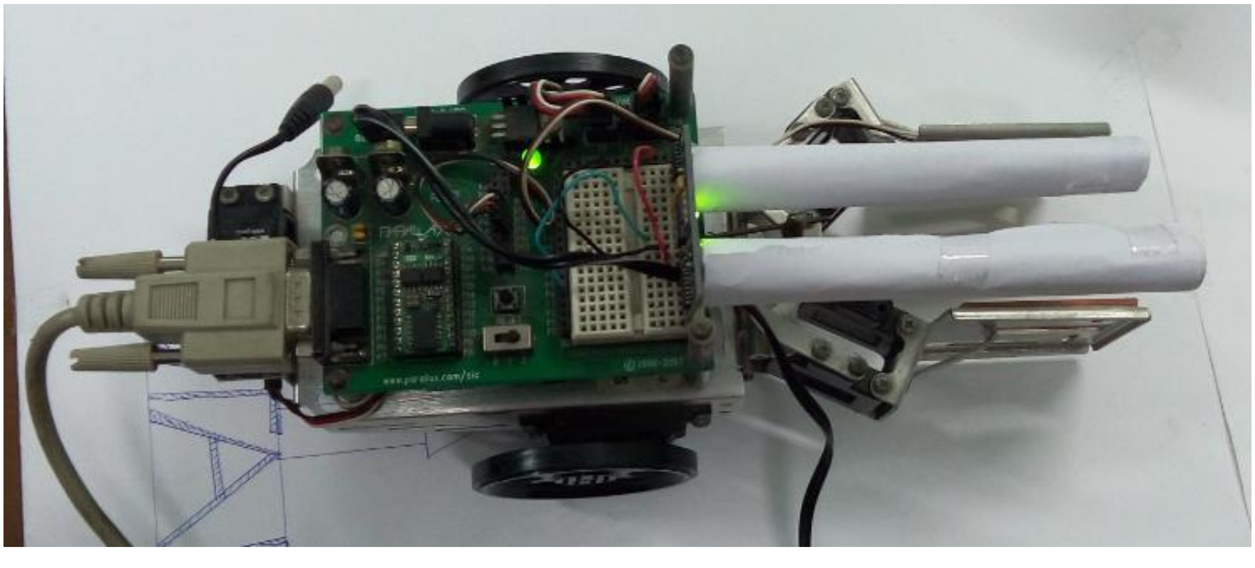

**Fig.6.7:** Ultrasonic sensor attached with two hollow paper cylinders.

However, the remaining part of the PBASIC program could not be executed, as the current version of the BASIC Stamp2 microcontroller available in the Robotics Laboratory does not have sufficient memory for storing values of all the variables used in the developed PBASIC program for path planning using APF method.

# CHAPTER 7

# CONCLUSION

## CONCLUSION

Based on the foregoing analysis, study, algorithm and program development and result on path planning of mobile robot in presence of multiple obstacles using artificial potential field method with the help of ultrasonic range sensors for the present project, the following general conclusion may be drawn:

1) Various path planning techniques and algorithms including APF method have been studied.

2) An arrangement has been set up for a workspace consisting of a Parallax Boe-Bot mobile robot with a Parallax Ping ultrasonic range sensor and multiple obstacles on a suitable work-table in the Robotics Laboratory of Production Engineering Department.

3) The Ping ultrasonic sensor has been suitably mounted on a breadboard fixed on the Boe-Bot mobile robot system, and necessary hardware connection has been made to connect it to the BASIC Stamp microcontroller which runs the mobile robot.

4) Program for detection of position and distance of obstacles from the mobile robot has been developed.

5) Program has been further modified for online path planning using APF method after detection and determination of size and location of all the obstacles.

6) Some assumption has been taken about the obstacle that distance from two edge point of the obstacle to robot position are almost same.

7) For getting better result other types of sensors like laser sensor, vision camera may be used.

Further scope of the present project includes:

1) Detection of obstacles for $360^0$ rotation of the mobile robot and find out the corresponding data for the obstacle.

2) Development of program for movement of mobile robot from any start position to goal position using APF algorithm.

# CHAPTER 8

# REFERENCES

## REFERENCES

[1] Lim,J., Lee, S.,Tewolde,G., and Kwon,J., "*Ultrasonic- Sensor Deployment Strategies and Use of Smartphone Sensors for Mobile Robot Navigation in Indoor Environment"*

[2] Borenstein,J.,and Korean,Y., "*Obstacle Avoidance with Ultrasonic Sensors*" IEEE journal of robotics and automation,Vol. 4, no. 2, april 1988

[3] Pears,N.E., and Probert, P.J., "*An intelligent active range sensor for mobile robot guidance*" Mechatronics ,Vol. 6, No. 7, pp. 733-759, 1996

[4] Nagahara, D.,and Takahashi,S., "*Mobile robot control based on information of the scanning laser range sensor",* The 11th IEEE International Workshop on Advanced Motion Control,March 21-24, 2010.

[5] Pfister,S.T., Kriechbaum, K. L., Roumelioti, S. I.and Burdick, J.W., "*Weighted Range Sensor Matching Algorithms for Mobile Robot Displacement Estimation*" international Conference on Robotics & AutomationWashington,1667-1674,may 2002.

[6] Nelson,E.,Willis,S.,and Lucero, N., "*Automatic Site Reconstruction with a Mobile Robot and Scanning Laser Proximity Sensor*" June 13, 2013

[7] Li,G., Yamashita, A., Asama, H., and Tamura Y., "*An Efficient Improved Artificial Potential Field Based Regression Search Method for Robot Path Planning*"

[8] Zhang,G., and Ferrari, S., "*An Adaptive Artificial Potential Function Approach for Geometric Sensing"*

[9] Seki, H., Kamiya, Y., Hikizu, M., *"Real-time Obstacle Avoidance Using Potential Field for a Nonholonomic Vehicle*", pp. 523-542

[10] Koren, Y., and Borenstein,J., "*Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigatio*n", Proceedings of the IEEE Conference on Robotics and Automation, pp. 1398-1404, April 7-12, 1991

[11] Xie, L., Chen, H., Xie, G., "*Artificial Potential Field Based Path Planning for Mobile Robots Using Virtual Water-Flow Method*", Communication in Computer and Information Science, pp. 1-4

[12] Vadakkepat, P., Lee, T. H., Xin, L., "*Evolutionary Artificial Potential Field — Applications to Mobile Robot Path Planning"*, Autonomous Robotic System, Vol 116, pp. 217-232

[13] Hwang, Y. K. and Ahuja, N., "*Path Planning Using a Potential Field Representation*", IEEE Conference on Robotics and Automation, pp. 648-649, 1988

[14] Warren, C. W., "*Global Path Planning Using Artificial Potential Fields*", IEEE Conference on Robotics and Automation, pp. 316-321, 1989

[15] Tilove, R. B., "*Local Obstacle Avoidance for Mobile Robots Based on the Method of Artificial Potentials", IEEE* Conference on Robotics and Automation, pp. 566-571, 1990

[16] Koren, Y. and Borenstein, J., "*Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation*", IEEE Conference on Robotics and Automation, Sacramento, California, pp. 1398-1404, April, 1991

[17] Ge, S. S. and Cui, Y. J., "*New Potential Functions for Mobile Robot Path Planning*", IEEE Transactions on Robotics and Automation, Vol. 16, No. 5, pp. 615-620, October 2000

[18] Mbede, J. B., Huang, X., Wang, M., "*Fuzzy motion planning among dynamic obstacles using artificial potential fields for robot manipulators*", Robotics and Autonomous Systems, Vol. 32, pp. 61–72, 2000

[19] Park,M.,G., Lee,M.,C., "*Artificial Potential Field Based Path Planning for Mobile robot using a virtual obstacle concept*", IEEE/ASME International Conference onAdvanced Intelligent Mechatronics, 2003

[20] Kowalczuk, Z. and Duzinkiewicz, K., "*Planning Robot Trajectories Using a Modified Artificial Potential Method*", pp. 1-4, 2007

[21] Safadi, H., "Local Path Planning Using Virtual Potential Field", April, 2007

[22] Liu, Y. and Bharadwaj, K. K., "*A Hybrid Artificial Potential Field: Genetic Algorithm Approach to Mobile Robot Path Planning in Dynamic Environments*", Computer Science and Convergence, Vol. 114, pp 325-333, December 2011

[23] Xu, Z, Hess, R., Schilling, K., *"Constraints of Potential Field for Obstacle Avoidance on Car-like Mobile Robots*", Conference on Embedded Systems, Computational Intelligence and Telematics in Control, pp. 169-175, April 2012

[24] Zhang, G. and Ferrari, S. "*An Adaptive Artificial Potential Function Approach for Geometric Sensing*" 2012

[25] Lee J., Nam Y., Hong S., Cho W. *"New Potential Functions with Random Force Algorithms Using Potential Field Method"*, J Intell Robot Syst, Vol. 66, pp 303-319, 2012

[26] Chauhan, S. S., Bajpai, A. K. "*Collision Free Autonomous Robot Path Planning*", International Journal of Engineering Research & Technology, Vol. 1, No. 8, PP. 2278-0181, October, 2012

[27] Li F., Wang, Y., Tan, Y. Ge, G. "*Mobile Robots Path Planning Based on Evolutionary Artificial Potential Fields Approach*", International Conference on Computer Science and Electronics Engineering, 2013

[28] Malakar, M. and Sinha, N., "*A Particle Swarm Optimized Potential Field Method for Obstacle Avoidance and Path Planning of Mobile Robot*", International Journal of Digital Application & Contemporary research, Vol. 2, No. 5,pp. 2319-4863, December 2013

[29] Wyrąbkiewicz, K.,Tarczewski, T., "*Artifical potential fields with extended Bug algorithmfor Mars rover path planning in an unknown environment*" Computer Applications in Electrical Engineering, Vol. 12.,2014

[30] Min, H., Lin, Y., Wang, S., Wu, F., Shen, X., "*Path planning of mobile robot by mixing experience with modified artificial potential field method*", Advances in Mechanical Engineering, Vol. 7, No. 12, December, 2015

[31] Weerakoon,T.,Ishii , K.,and Nassiraei,A.,L.,F., *"an artificial potential field based mobile robot navigation method to prevent from deadlock"* JAISCR,Vol. 5, No. 3,pp. 189-215,2003

[32] Chen,W.,Wu, X., Lu, Y., "*An Improved Path Planning Method Based on Artificial Potential Field for a Mobile Robot*", cybernetics and information technologies, vol.15, No. 2, 2015

[33] Zhou, L., Li,W., "*A Potential Field Method Approach to Intelligent Vehicle Path Planning*", Journal of Computational Information Systems,2015

[34] Ahmed, A. A., Abdalla, T. Y., Abed, A. A., "*Path Planning of Mobile Robot by using Modified Optimized Potential Field Method*", International Journal of Computer Applications, Vol. 113, No. 4, pp. 6-10, March, 2015

[35] Montiel, O., Ulveda, R. S., Rosas, U. O., "*Optimal Path Planning Generation for Mobile Robots using Parallel Evolutionary Artificial Potential Field*", J Intell Robot Syst, Vol. 79, pp. 237–257, 2015

[36] Goodrich,M.,A., "Potential Fields Tutorial"

[37] Li, C. Jiang, X., Wang, W., Cheng, Q., Shen, Y., "*A Simplified Car-Following Model Based on the Artificial Potential Field*", Procedia Engineering, Vol. 137, pp. 13 – 20, 2016

[38] BASIC Stamp Syntax and Reference Manual 2.2 , PARALLAX INC.

[39] Robotics with Boe-Bot, student Guide Version 2.2, PARALLAX INC.