Bioinspired Robot Path Planning using PointBug Algorithm

Nupur Choudhury
Department of Computer Science Engineering & IT
Assam Don Bosco University
Guwahati, India
nupur.choudhury@dbuniversity.ac.in

Rupesh Mandal
Department of Computer Science and Engineering
Sikkim Manipal Institute of Technology
Sikkim, India
rupeshmandal29@gmail.com

Suman Kalyan Kar
Department of Computer Science and Engineering
Sikkim Manipal Institute of Technology
Sikkim, India
sumansit@gmail.com

Abstract— **This paper is an outcome of a research implementation that was carried out on a BOEbot (Board Of Education bot) existing in the robotics laboratory of Computer Science and Engineering Department of Sikkim Manipal Institute of Technology which resulted in the development of an essential algorithm that is primarily used to navigate a mobile robot in presence of a wide range of obstacles enabling the robot to move from starting location to a target location avoiding all the obstacles in between with the help of ultrasonic sensors. In order to achieve this it is utmost necessary to obtain a path that would avoid the obstacles that might be present in the static environments which in this paper is obtained using simple local path planning algorithm known as the PointBug algorithm. Initially the algorithm determines the next point for the robot to move forward to the target from any current point which depicts the starting point in the beginning. The next point is determined by the output of the range sensors which depicts the nearest obstacle distance from the sensor which can also be termed as sudden change which can be considered as increasing or decreasing by a considerable value Δd which needs to be defined with accuracy.**

Keywords— BOEBot; bug algorithms; robot navigation; path planning; pointbug.

I. INTRODUCTION

A robot is basically a machine formed by electromechanical circuitry which can be operated via a computer program or electronic devices and is capable of performing automated tasks using artificial intelligence techniques with the aid of direct human supervision, preloaded programs or by the utilization of a general set of guidelines. The automotive industry Ford made use of the first commercially developed robot in 1961. Robots are expected to act as a replacement to humans in heavy, hazardous and monotonous activities. At present due to diversified economic factors, the industrial robotic frameworks primarily the stationary ones are utilized in wide range of applications operating from a fixed location having restricted operating range. The area that surrounds the robot is secured from external influences and is generally designed in function which corresponds to the task of robots which may include tasks like welding, drilling, assembling, painting and packaging of various items. However it is always better to construct a robot which has greater mobility and locomotion. The primary focus of these robots is to adapt their behavior towards the tasks that are being oriented with him. Along with the various action sequences that are fixed for it, the robot should also be able to take inputs from the sensors that are attached to it and gather a general perception of the various kinds of sensors that are used and utilize the on board intelligence as to what particular action to take at which moment which leads to the collision free navigation of the robot in an environment which has formed the basis of several research projects in this particular domain. A robot which is not fixed in one particular location and is capable of navigating throughout its environment can be termed as mobile robot, However, the industrial robots are those robots which has a joint arm(multi-linked manipulator) fixed to it or a gripper assembly(or end effectors) which is affixed with a surface. These robot forms a lot of current research activities present in almost all the major universities which has one or more research labs that focus on mobile robot research. Mobile robots have formed an essential part in military, security and industry environments and can carry out a lot of exciting tasks with greater flexibility. The modification for the world is not required here and the robot framework can move wherever it is required with the advantage of the deployment of fewer robots in hand and can cooperate with men by sharing workspace together. Considering this locomotion is an important aspect of robot navigation design. These locomotion designs are mostly inspired by the biological counterparts which have inherited the capability of adapting to environmental techniques such as walking, crawling, slithering and hopping.

II. PATH PLANNING OF MOBILE ROBOTS

The various researches in autonomous mobile robot initiated somewhere around 1970. The primary motive of the path planning strategy is to construct a collision-free and optimized path (i.e. shortest distance). With regards to this definition path planning problem is a categorization of an optimization problem. However the roadblocks include computational complexity, local optimum and adaptability.

978-1-4673-9939-5/16/$31.00 ©2016 IEEE

Path planning algorithm may be primarily categorized in two categories [1]: Offline and online which can also be termed as global path planning and local path planning respectively which is centered on the information perceived from the environment. Offline path planning relates to the field where the robot only needs to compute the path in the initial state and it follows the planned path to the target point provided it has detailed information about the environment. Online path planning basically focusses in utilizing information from the sensors as it navigates around in an environment. The robot essentially needs to have the capability of building a map of an area as it moves through it which is a continuous process of moving to a new position, sensing the environment and planning the next motion after updating the maps. Ultrasonic sensors or cameras mounted on the robots chassis can act as an aid to determine the obstacles in the trajectory of motion. In the present work, a simple local path planning algorithm (PointBug Algorithm) is used for finding a collision-free path for a mobile robot, which determines where the next point to move toward target from a starting point. The next point is determined by the output of an ultrasonic range sensor.

III. OBJECTIVE

The primary objective of the present work is as follows:

a. To create a workspace which would consist of a BOEbot robot (mobile) with ultrasonic range sensors (ping) and the obstacles with a marked boundary for navigation.

b. To develop a program in PBASIC language for enabling the robot to move in presence of static obstacles from a starting point to a particular target point by implementing PointBug algorithm.

c. Several test cases needs to be satisfied which would include a wide variety of varied layouts of workspace for testing the algorithm.

IV. PATH PLANNING TECHNIQUES:AN OVERVIEW

According to environment, type of sensor, robot capabilities and others, various approaches have been proposed for path planning of mobile robots which includes approaches that demonstrates better performance in terms of complexity, time and distance.

A. Path planning in static and dynamic environment

Static path planning involves navigating a robot from the initial to the goal position by avoiding the stationary obstacles that are on the way. The main target of the robot is to navigate freely from one point to another by avoiding all the obstacles that are stationary and achieve an optimal solution with minimum costs. A combination of sensor based and knowledge based path planning can be seen in case of dynamic path planning. Even though inaccurate information is generated the path planning could be done efficiently in case of the knowledge based path planning. A sensor is responsible for detecting any obstacle on the execution of the path after which the information is passed on to the planner and accordingly the world or environment model is updated. In case of dynamic planning the execution module always generates a feedback to the planning module as it is characterized by separate path planning and path execution modules.

B. Planning strategies: Local and Global path planning

The domain that involves the robot to navigate with prior information falls under Global planning where the information that is related to the environment is initially loaded into the path planning program of the robot prior to determining the path that needs to be taken from starting point to the target point which will enable the program to determine an entire path from the starting to the target point before the navigation starts. In other words it involves deliberate process of determining the best move to any location by comparing with any other location by pre-deciding on the various factors that might affect the environment. On the other hand Local path planning involves the enabling the robot to learn from its environment and autonomously move around where there are various algorithms for path planning that might be able to respond to various obstacles that might be on the way or to a change of the entire environment as a whole due to which it can also be termed as real time obstacle avoidance. Usually in local path planning the robot is guided by a single line until and unless it senses any obstacles after which it performs obstacle avoidance phase starts off and new information is updated by mentioning the new location and the current position so that the robot can reach the target location accurately. In the present work, a new local path planning algorithm, called PointBug algorithm, has been used. The algorithm is based on the Bug family of path planning algorithms.

V. AN OVERVIEW OF BUG ALGORITHMS

Local path planning can be carried out efficiently using a wide variety of the Bug algorithms which enhances the mobile robot navigation [2]. These algorithms were initialized by James Ng and Thomas Braunl and they listed around 11 different types of Bug algorithms [3]. However most common types that are used in mobile robot path planning are Bug1 and Bug2, DistBug, VisBug and TangentBug. An addition to it are Alg1 and Alg2, Class1, Rev1 and Rev2, OneBug and LeaveBug. These variations are in the form of the algorithms are in the form of the effort towards shorter path planning, timing, simpler algorithm and better performance. Bug1 and Bug 2 are almost similar in behavior, however the difference lies in Bug2 (Figure1 (a) and Figure1(b)) where it has m-line which is utilized as the hitting and the leaving point. The Bug1 is considered as overcautious which has coverage more than the entire perimeter of the obstacle and is more effective than bug 2 since n some cases the algorithm results in shorter coverage and local loops. VisBug is used implement range sensors that can calculate the shortest distances from or to the m-line generated by the Bug2 algorithms. Alg1 could improve Bug2 weakness as it could trace the same path two times by calculating and storing the sequences of the various hit points that is generated in the actual path to reach the goal. Alg2 was then developed to improve on Alg1 where new leaving condition was introduced and the m-line of Bug2 was ignored. However both Alg1 and Alg2 faces reverse procedure problem where visited points within a loop created certain problems by following uncertain obstacles. This was countered by Horiuchi's Rev1 and Rev2 algorithms by introducing a combination of reverse procedure with alternating following method which would decrease the possibility of the robot to follow a loop round the destination. DistBug and TangentBug algorithms also implements these sensors where they make use of odometry, goal position and range sensor data and calculates its positions[5]. In this algorithms, a global information dmin(T) is obtained from visual information which is not accurate at times to determine if the loop around an obstacle is completed by the mobile robot. TangentBug on the other hand build a Local Tangent Graph (LTG) which is dependent on the sensor's immediate information. A scale from 0 to infinity (a 3600 infinite orientation resolution is used in order to detect the obstacles. The entire process results in the robot making a 3600 revolution an also calculating the minimum distance to the next target in the goal line while following the boundary of the obstacle and moves towards the target as and when it could avid the obstacle[6]. However the major disadvantage this type of algorithm is that it has to make a complete circle of 3600 before it could draw any decision based on the trajectory to move to the next goal. The latest variant of this algorithm is LadyBug which could incorporate heuristics based on biological inspired methodologies that could improve the trajectory of the robot in real time resembling the Ladybug's aphids hunt considering a network of mobile robots [9].

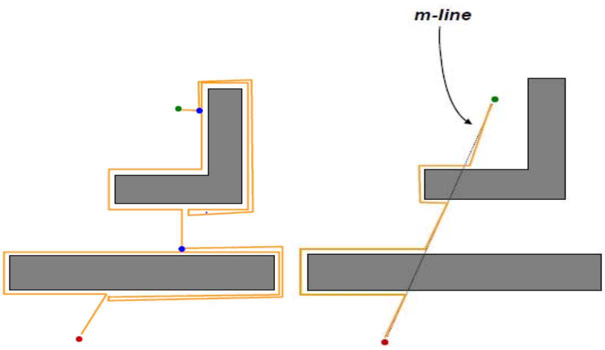

Fig 1: Path traversal by robot using Bug 1(a) and Bug 2(b) algorithm. [6]

VI. INTRODUCTION TO POINTBUG ALGORITHM

This algorithm commonly known as PointBug algorithm is used for navigating a point of a mobile robot in an unknown environment plane which has stationary obstacles of various shapes [4]. It is used to detect the successive point (which is basically determined by the output of the range sensors depicting the nearest obstacle to it) to move in the trajectory containing the starting point and the target. The sharp change in the sensor reading (increasing or decreasing) is dented as the 'sudden change' which is measured in terms of the distance from the sensor to the nearest obstacle. Δd associated to it is the difference value which can have a range from infinity to a certain value or vice versa or between 2 different values. The 'sudden point' is calculated as the range difference calculated by the range readings via sensors in Δd at a time interval tn to tn+1. The robot is capable of scanning the environment with its range sensors by a 00 to 3600 rotation. Initial position of the robot faces the target and then it can rotate in either left or right direction in search of sudden points in the trajectory in order to detect any obstacles that might be there in between. On discovering the first sudden point the direction of rotation of the robot is dependent on the position of the straight line that connects the sudden point and the target or in other terms the dmin line which is generally towards the direction of the dmin line. The value of dmin is always recorded every time the robot attains a new value or a new sudden point and is the smallest distance between the current point and the target. However the sensor reading at 1800 rotation is usually ignored by the robot in order to avoid and pre detected sudden points which would make the robot to return to the previous point encountered by it with respect to the current point. Target would be considered as unavailable if there are no sudden point detected by the robot even after making a complete rotation of 3600 and the robot will automatically reach a halting state.

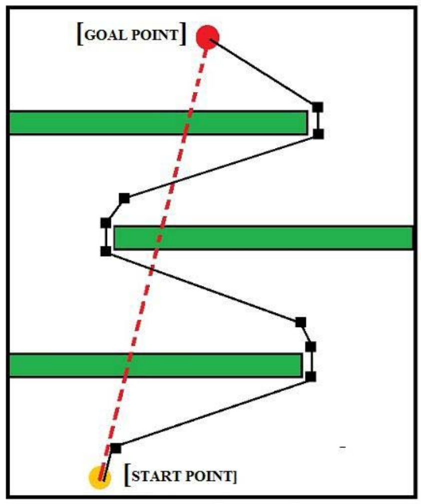

Fig 2: Sudden points and trajectory generated using PointBug Algorithm in simple maze based environment.

The pseudo code of the PointBug algorithm is as follows:

While current point is not equal to target
If degree of rotation (robot) <= 360^0
According to d_{min} value perform rotation (left or right)
If a sudden point is encountered
If rotation=180^0
Ignore detected point.
Else
Get distance from detected sudden point to next new sudden point.
Obtain the angle of rotation and navigate to the new point.
Record new d_{min} value
Reset the rotation angle
End if
End if

Else
Halt robot /* No next sudden point and exit loop */
End if
While end
Stop Robot /* Robot successfully reaches target

This algorithm is capable of operating in dynamic environments because of its ability of obtaining immediate information from the range sensor during its movement. However the performance of the algorithm is dependent on the number of detected sudden points where the lesser the number of sudden points the better is the performance [7]. Figure 3 shows a simple maze based environment where the sudden points are generated at each and every vertex of the rectangles which would enable the algorithm to generate the shortest trajectories from the initial to the final point.

VII. HARDWARE AND SOFTWARE REQUIREMENTS

The hardware and software required to perform this experiment is as follows.

Hardwares: Parallax Board of Education Robot (BOEBot) and EmbededBlue eb500-SER (Bluetooth Module)

Softwares: 1. Basic Stamp Editio (v2.4.3),2. Boe-Bot Controller (Android App).

VIII. PROGRAMMING FOR PATHPLANNING USING UTRASONIC RANGE SENSORS

A program has been developed in PBASIC language in the present project for path planning of the BOE-Bot mobile robot using PointBug algorithm using the Ping ultrasonic range sensor mounted on the system carrier board. The PBASIC program is developed to instruct the mobile robot to navigate from a start point to a goal point in presence of obstacles. The start point has been assumed to be the origin of a co-ordinate system, and the co-ordinates of the goal point are entered as variables during execution of the program. The BOE-Bot mobile robot is assumed to be parallel to the x-axis at the beginning. The mobile robot rotates towards the goal point and starts moving towards it until any obstacle is detected on its path by the range sensor, whose output is monitored continuously during movement of the robot. Only the nearest obstacle, which is within a specified range (taken as 20 cm), has been considered. If an obstacle is detected, the robot rotates both left and right searching for ‘sudden point’, where the reading of the range sensor changes sharply. A change in the value from 20 cm (the distance at which an obstacle is detected) to 25 cm or greater in the reading of the range sensor output has been considered for detecting ‘sudden point’. The robot then moves towards that ‘sudden point’ out of the two ‘sudden points’ for an obstacle, for which the rotational movement of the robot minimum. When the robot moves towards the ‘sudden point’, it also checks for the presence of any other obstacle on its path by monitoring continuously the output of the range sensor. If another obstacle is detected, the process is repeated, otherwise the robot moves through a further distance (taken as 10 cm) past the ‘sudden point’ to prevent any collision with the obstacle when it again rotates towards the goal point and moves straight towards it. The process is repeated until the mobile robot reaches the goal point, which is assured by considering the distance of the current robot position from the goal point. The mobile robot is assumed to be a point in the algorithm. Since the ultrasonic range sensor detects obstacle within some angular spread from normal distance, the ‘sudden point’ is detected at a small distance from the obstacle so that when the mobile robot moves towards ‘sudden point’, it automatically moves away by a small distance from the obstacle avoiding any collision. As stated above, the robot is also forced to move past the ‘sudden point’ by a small distance before turning towards goal point.

IX. EXPERIMENTATION AND RESULTS

The developed program has been run successfully for different layouts of obstacles and goal points. Here in this experiment BOE-Bot is used as the robot and ultrasonic range sensor (Ping sensor) is used. The photographic view of one such layout of workspace with the robot at the initial position and orientation has been shown in the following figures. Here are the images which are captured during robots movement in an environment which also have obstacle. The images are arranged in step wise manner that is the robots movement from initial position to its goal position avoiding all possible obstacles.

a

b

c

d

e f

g h

i j

Fig 2: Series of inages (a, b, …., j) of robot movement in the workspace with obstacles.

The robot is first kept on the ground which is its initial position. The user need to provide the location of the goal point. The coordinates of this goal point is the sent back to robot and the minimum distance is calculated between initial and goal point (shown in image a). After calculating the minimum distance is started traversing the minimum distance which is straight path toward the goal (shown in image b). When it reach a point which was just 20 cm away from the obstacle, it came to know that some obstacle is lying in the path which it was traversing (shown in image c). In this position the robot was searching for next sudden change point. Then it is moving toward that sudden change point (shown in image d). Once it reached the sudden change point, it again calculate the minimum distance between the sudden change point and the goal point and started traversing the path (shown in image e & f). While traversing the new path again it found obstacle (shown in image g). Again it searched for sudden change point and move toward that (shown in image h). Form that sudden change point further it calculated the minimum distance toward the goal and starts traversing (shown in image i). Finally it reached the goal as no more obstacle is there (show in image j).

X. Conclusion and Future Scope

Based on the analysis, study, algorithm and program development, experimentations and results on the Online Path planning using Ultrasonic sensors of a Mobile robot we could draw the following conclusions.

1. A local path planning algorithm was implemented using an ultrasonic sensor to attain a collision free path having unknown stationary obstacles and positions. PointBug algorithm is one of the algorithms of the well-known Bug family algorithms used for robot navigations using simple fundamentals and has minimum sensor requirements where the next point of traversal is basically determined by the output of the sensor readings.
2. The sudden change in the distance of the sensor to the nearest stationery obstacle is depicted from the range senor's output by noting the well-marked difference value of the sensor readings.
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. A program based on PointBug Algorithm has been developed in PBASIC language for producing necessary movements of the BOE-Bot mobile robot in presence of static obstacles for moving from a start point to a goal point using the output of the range sensor.
5. The obstacle is detected within an angular spread from the normal distance so that whenever a sudden point is encountered, the robot moves towards it and automatically avoids any collision by moving past the sudden point in a range of a small distance prior to the turn towards goal point.
6. The PBASIC program has been run successfully for different layouts of workspace.

Further scope of the present project includes development of path planning algorithms in presence of static as well as dynamic (moving) obstacles in the environment using camera or other sensors.

References

[1] Al-Taharwa I., Sheta A., Al-Weshah "A Mobile Robot Path Planning Using Genetic Algorithm in Static Environment", Journal of Computer Science, 4 (4): 341-344, 2008.

[2] Bug Algorithm [online]. http://www.aishack.in/tutorials/obstacle-avoidance-with-the-bug-algorithm/ , Last accessed : 25/05/2015.

[3] Implementing bug algorithms variants [online]. https://github.com/SelmaKchir/BugAlgorithms/wiki/Implementing-Bug-Algorithms-variants , Last accessed : 25/05/2015.

[4] Kumar, E.V, M. Aneja, D. Deodhare, "Solving a Path Planning Problem in a Partially Known Environment using a Swarm Algorithm", in IEEE International Symposium on Measurements and Control in Robotics. 2008. Bangalore, India.

[5] Ng J., T. Bräunl, "Performance Comparison of Bug Navigation Algorithms", Journal of Intelligent and Robotic Systems, 2007, p: 73-84.

[6] Robotic Motion Planning: Bug Algorithms [online] www.cs.cmu.edu/~motionplanning/lecture/Chap2-Bug-Alg_howie.pdf , Last accessed : 25/05/2015.

[7] Sankaranarayanan A., M. Vidyasagar, "Path Planning for Moving a Point Object Amidst Unknown Obstacles in a Plane", the universal lower bound on the worst path lengths and a classification of algorithms, IEEE International Conference on Robotics and Automation, 1991: p. 1734 - 1741.

[8] Susnea I., Viore Minzu, Grigore Vasiliu, "Simple Real-Time Obstacle Avoidance Algorithm for Mobile Robots", in 8th WSEAS International Conference on Computational intelligence, man-machine systems and cybernetics CIMMACS'09), 2009.

[9] O. Khatib, K. Yokoi, O. Brock, K. Chang, and A. Casal,Robotics Laboratory,Department of Computer Science Stanford University, Stanford, California, Robots in Human Environments, [Available online: www.redaktion.tu-berlin.de , Last Access: October 2014].

[10] Vladimir Kulyukin, Chaitanya Gharpure Nathan De Graw, Computer Science Department Utah State University Logan. Human-Robot Interaction in a Robotic Guide for the Visually Impaired,[Available online: www.aaai.org (American Association for Artificial Intelligence). Last Access: October 2014].

[11] [Hyun Keun Park, Hyun Seok Hong, Han Jo Kwon, and Myung Jin Chung, Department of Electrical Engineering & Computer Science Korea Advanced Institute of Science and Technology, A Nursing Robot System for The Elderly and The Disabled, [Available online: http://www.pdx.edu , Last Access: September 2014]