ISSN (Print) : 0974-6846
Indian Journal of Science and Technology, Vol 9(41), DOI: 10.17485/ijst/2016/v9i41/93816, November 2016 ISSN (Online) : 0974-5645

Trajectory Planning of a Mobile Robot using Enhanced A-Star Algorithm

Priyanka Sudhakara and Velappa Ganapathy*

School of Computing, SRM University, Kancheepuram – 603203, Tamil Nadu, India; priyanka.k@ktr.srmuniv.ac.in, ganapathy.v@ktr.srmuniv.ac.in

Abstract

Objectives: This paper presents an optimization technique for a robot to identify the best path for traversing from a start state to a goal state without colliding with any obstacles. **Methods**: The existing A-star algorithm is modified to make the robot to move in an unknown scenario in which static obstacles are located. The robot is expected to travel to the goal position without colliding with any of the obstacles present. The Enhanced A* algorithm guides the robot to reach the target following an optimal path. By adding a new parameter namely number of turnings (p(n)) the robot makes during its traverse, currently existing A* algorithm is improved. This is done in order to improve the algorithm for optimal motion of the robot. **Findings**: The proposed Enhanced A* algorithm will be shown to remove certain drawbacks, which are found in the currently existing algorithms, for optimal travel of the robot. Simulation experiments are carried out to compare the performances of the proposed Enhanced A* algorithm and the original A* algorithm. The outcome of the simulation experiments is expected to provide better performance as compared to the A* algorithm concerning Elapsed time of travel. It is expected that the Enhanced A* algorithm is able to provide an effective contrivance for exploring path planning in robotics education and path control of robot. **Improvements:** The Future Enhancement is to make the algorithm work in an unknown environment with dynamic obstacles present and implement the same in real time using an autonomous mobile robot.

Keywords: A* Algorithm, Mobile Robot, Machine Learning, Path Planning, Static Obstacles, Unknown Environment

1. Introduction

Since 1980, a lot of research works were done to solve the issues relating to the path planning of mobile robots. There are two ways in which the path planning problems can be tackled. One of the methods adapted is using the global scenario, obstacle information and characteristics of robots. Another way is to collect the local information through sensors and solve the path traversing problem. Nowadays, efforts are being made to strike a trade-off between the reactive approach to environmental events and developing effective trajectory planning using optimization techniques. Several changes have been made in the traditional techniques to have an effective path planning using sensors. An effective algorithm namely Enhanced A* Algorithm for path planning by introducing an additional parameter in the existing A* Algorithm is proposed. In our work, the performances of different versions of existing A* algorithm are compared with Enhanced A* algorithm. Planning for mobile robot navigation involves the identification of obstacles and estimating an optimal path to reach the goal and also to avoid any obstacles. The main objective of motion planning of a mobile robot is to explore an effective and efficient path so that a robot moves in a prescribed route within a specified framework.

The authors[1] have considered the task of robot navigation planning in frameworks in which non-rigid obstacles exist. They have combined simulation of physical objects and probabilistic roadmaps and evaluated an optimal path which has become a trade-off between the distance

*Author for correspondence

to be travelled and the cost of deformation. Autonomous robots path planning techniques[2] are analysed for optimal motion. A local path planning algorithm based on Bug algorithm is proposed. Using sparse information, detected by the sensors which are closest to the obstacles is used by the Bug algorithm for the robot to travel to the destination. The obstacle borders are identified when the robot circumvents the obstacles so as to satisfy some conditions specified in the algorithm so that the robot safely avoids the obstacles and travel towards its destination. A Point Bug algorithm for robot navigation is also introduced by them in which the use of outer perimeter of an obstacle is minimized by looking at only certain important points on the obstacles' outer perimeter and hence the robot avoids the obstacle and turns towards the goal and finally generates a whole path from source to destination.

Multi-agent trajectory planning with obstacles present in an environment was tackled[3]. A unique optimal multi-agent trajectory planning is described using a model of the environment which is represented graphically. The environment which is represented as a graphical model is changed dynamically avoiding obstacle collisions and also correcting the path accordingly. Changes in the robot paths are introduced in the new algorithm which makes the robot to move away so that collisions are avoided. An optimal path between a source and target points in a specified grid environment[4] is found using genetic algorithm so as to control the mobile robot to reach the target safely. An algorithm which uses search spaces that are complicated and having low complexities and allowing the robot move to four-neighbouring states has been developed.

A trajectory planning algorithm[5] either global or local path planning using sensors is able to assimilate the complete updated information about obstacles. The Voronoi Diagram is used to obtain safe areas available in the environment. Then the safe areas are collected by Fast Marching Method to get the traversing path. To provide a safe and reliable robot movement, map-based and sensor-based planning operations are clubbed. A path planning algorithm for traversing a mobile robot is developed with a webcam[6]. This technique is used to find minimal path length for the robot to travel and reach the goal, at the same time avoiding collision with any of the obstacles present along the way. The obstacles' locations are traced by the images captured by the webcam. Then the shortest distance to the goal is estimated using Voronoi Diagram. The authors[7] have proposed an online path planning algorithm, based on the so-called network simplex method. They have assumed that the sensing range of the robot is short compared to the individual path lengths that it plans and hence the scenario is modelled as a graph comprised of nodes and arcs. The network simplex method is adapted to execute the re-planning issue. The utility of the method is demonstrated by combining it with a navigation control strategy.

The D* Lite Algorithm[8] has been extended and the new algorithm has been named as Enhanced D* Lite Algorithm. It prevents mobile robot from traversing across obstacle's sharp corners in between two obstacles and also avoids irregular obstacles. Virtual walls are created to avoid traversing to the same place again and again. It is capable of re-planning quickly to traverse and also remembering the path. The authors[9] have explored the robot's environment using Artificial Neural Networks and Fuzzy Logic techniques and aid the robot to reach its goal following a shortest path. Path Remembering Robot Algorithm is proposed and this will assist the robot to avoid acute obstacles. Virtual Wall building method is also proposed to prevent the robot traversing the same acute obstacles again and again. The Classical Q-Learning[10] algorithm (CQL) has been extended and the new algorithm named as Improved Q-Learning (IQL) algorithm. The CQL utilizes a Q-table to store the Q(S, a) where 'S' are states and 'a' are actions. Thus, it needs an array of size $(n \times m)$. In the IQL, it is advantages to store Q-values only at a specific state S. 'n' number of Q-values are there for 'n' states. Apart from storing Q-values to represent the current status of a specific state, 'n' Boolean Lock Variables are required. The Q-value at a particular state is not to be updated further if the Lock variable is at state 1. A Parallel Elite Genetic Algorithm (PEGA)[11] is applied for generating a global trajectory planning for a mobile robot which is to navigate in a well organized environment. The PEGA consists of two types and having a migration operator. This operator helps preserving good population diversity, keeping parallelism and preventing premature convergence as compared to that of conventional Genetic Algorithms. Cubic B-spline is used to smoothen the initially created feasible trajectory using PEGA so that a near optimal collision free trajectory can be constructed. One Field-Programmable Gate Array (FPGA) chip using the concept of system on a programmable chip technology and pipelined hardware implementation scheme both used to improve the computational speed, is utilised for trajectory planning and smoothing.

To improve the search ability, an adaptive-velocity-mutated operation[12] technique, integrated with the trajectory planning system, has been proposed. PSO utilises a framework incorporating crossover and mutation operations and the new framework is named as Evolutionary-Group-Based Particle-Swarm-Optimization (EGPSO) for trajectory planning. For selecting parents in crossover operations, particle updates and replacements, different groups are formed using EGPSO. Fuzzy Controller designed by EGPSO is utilised to navigate a mobile robot in an unknown environment. For merging target seeking and boundary following navigations, a behaviour supervisor method is presented. Dead cycles existing in the system are clearly identified and accounted. Fermet points, Improved Dijkstra Algorithm and Delaunay Triangulation methods are utilised to construct a reduced visibility graph with a minimal path length[13].

A composite global method of optimal selection of trajectory and efficient dynamic avoidance of obstacles is presented[14]. For global path selection and dynamic obstacle avoidance the authors have used Distance Transform and GJK (Gilebert-Johnson-Keerthi) algorithm. Whereas the authors[15] have used Tabu list technique for dynamic obstacle avoidance and Honey Bee Mating Optimization (HBMO) algorithm for global optimal path selection. Their simulation results obtained are found to be much better than Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO) algorithms. A robot trajectory planning using a wavelet network control scheme is proposed[16]. The optimal Wavelet Neural Network parameters (WNN) and the Proportional Integral Derivative (PID) controller parameters are determined by the PSO. The PSO method and PID controller are utilised for controlling the non-holonomic mobile robot that involves path tracing using two optimized WNN controllers one to control the speed and another to control the azimuth angle. Matlab Simulink is used to model the robot and the PSO algorithm is written in MATLAB codes. Grid method[17] has been used for scenario modelling and to boost up searching speed, ACO algorithm with bipartite parallel searching approach is utilised to boost up searching speed. Trajectory planning of a mobile robot in a static known environment is explained[18] and the planning procedure is split up into simpler tasks. The first task is to create a trajectory from starting point to target point which is free of collisions and PSO is used to create the trajectory. Second task involves using a radial basis function for trajectory generation on the collision free path obtained. The third and final task is to use a PID controller to properly plan the robot path. Without using an optimal path to move from a random starting state to a final target state for trajectory planning, Q-learning can be used to reduce the convergence time[19]. Q-value for the best possible action at a state is stored by the proposed algorithm and hence significant storage is effective. Their proposed algorithm is then compared with classical Q-learning and is studied using Khepera-II robot. Trajectory planning of a mobile robot is attempted by combining cloud theory and rough set[20]. Their existing evolutionary algorithm's shortcoming is precocity and in order to overcome this issue they have used an algorithm which has considerably increased the speed and accuracy of robot movement.

Genetic algorithm has been used for robot path planning in a 3-D static environment to find a collision free optimal path[21]. The whole 3-D space is split into layers of 2-D space and a shortest path is found to reach the goal avoiding obstacles. A cost effective and indigenously developed educational framework to explain concepts involved in robotics and mechatronics areas has been presented[22]. In this framework, the robot is able to change its shape from a wheeled mobile robot to a humanoid thus facilitating conduct of large number of experiments. General Purpose Operation System (GPOS) performance with Real Time Operating System (RTOS) using a mobile robot as a real time application is analysed[23]. The mobile robot module was implemented on a single board computer having ARM11 as its core. The method used to calculate response is real-feel method which uses dedicated timer and interrupt to calculate the response. A review is carried out bringing out all types of architectures[24] now. Dynamic modelling for motion planning system and geometric modelling for path planning system are developed using a hybrid technology to accommodate the design complexity. Agent based path planning system and motion planning system are used for the path planner control which plays a critical role in controlling the movement of a non-holonomic robot. A model based on machine learning method has been used to achieve early detection of App-DDOS[25] by multitude request flood. The following metrics "Request chain length, request chain context, ratio of packet types, ratio of packet count, route context, router chain context and ratio of request intervals" are defined in the model ARTP.

Considering all the above works, with a view to improve the efficiency of robot navigation, we have proposed some changes in the existing A* Algorithm by introducing a new parameter to the algorithm, namely the number of turnings (p(n)) that the robot makes during its traverse. To reduce the elapsed time of travel by improving the optimal motion flow of the robot the number of turnings is used as the parameters.

2. System Overview

We propose a modified version of the existing A* algorithm for planning the trajectory for the robot to move in an unknown environment. Compared to other GA based algorithms with much specified constrains, A* algorithm performs better. We venture upon using a modified A* algorithm for planning the trajectory for autonomous mobile robot instead of using GA based approach. This ensures the mobile robot to optimise energy; minimise distance travelled and to reduce execution time. To make a fair comparison of the newly introduced Enhanced A* algorithm, the well known A* algorithm is also implemented for a mobile robot to traverse in an environment containing obstacles and the shortest path to reach the goal is evaluated. A grid map is designed to represent an unknown environment containing unknown static obstacles and identifying the quadrant in which the goal is situated. Since the goal is present in any one of the four quadrants only, we identify the quadrant and direct the movement of the robot to that quadrant and hence the robots path could be optimized. It is always advantages to minimise the time of travel, consumed energy and distance covered by making the robot to travel in an optimised path by effectively utilising the Enhanced A*algorithm. The cost function in our study consists of distance and time metrics which automatically implies that energy saving is also considered.

2.1 Problem Formulation and Outline of the Solution

The formulation of the path planning strategy is given: That is, given the start and end points, an optimal path between these two specified locations in the prescribed scenario is to be planned. The proposed optimal path should satisfy certain optimisation criteria like collision avoidance, shortest path, etc.

Because of the complexities, the path planning problem is grouped under NP-hard. The two factors [static or dynamic environments and global or local trajectory planning] which decide various methods to tackle the path planning issues were put forth by designers. An area which is devoid of any moving objects other than the navigating robot is defined as static environment and an area which contains moving robots, machines and human beings is defined as dynamic environment. With static terrain and a complete knowledge about the terrain are required for designing a global path planning algorithm. For producing a new path in response to immediate environmental changes, a local path planning problem is introduced for the robot to move.

The solution would be that in our proposed system, we are adding an another parameter with the existing parameters of A* Algorithm, which is to compute the number of turns that the robot makes before reaching the target after estimating the quadrant in which the target is located. This will reduce the time of travel as the robot restricts its movement towards the goal direction and hence takes the shortest path.

2.2 Framework

2.2.1 Robot Navigation: Block Diagram

In Figure 1, the Architectural Block Diagram of Robot Navigation is given. After initializing the robot, it starts finding the target location and tracks the path and the robot will make movements according to the proposed Enhanced A* algorithm through the destined path and reaches the target.

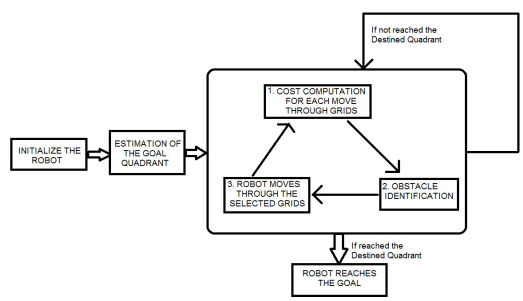

Figure 1. Architectural block diagram of the robot navigation

2.2.2 Module Split-Ups

2.2.2.1 Goal tracking

The goal is tracked by computing the co-ordinates, i.e., the difference between start coordinates and the goal coordinates as per the proposed Enhanced A* Algorithm.

2.2.2.2 Cost computation

The distance cost computation is the estimation of distance travelled from source to destination.

2.2.2.3 Path planning

A continuous motion, connecting a starting position S and the goal position G and avoiding any collision with obstacles present, is considered to be a general motion planning task. Two and three dimensional geometries are considered for describing the robot and obstacle geometries. The robots trajectory path may be described in higher dimensional configurations space also.

2.2.2.4 Obstacle detection

Obstacles avoidance in robotics is defined as the task in which many control objectives which are subjected to non collision and non intersection constraints are satisfied. Obstacle avoidance is one of the great challenges faced by Unmanned Air Vehicles (UAV) in the applications of military, agricultural, forest fire, etc. The trajectory planning and obstacle avoidance tasks are distinct in the sense trajectory planning is considered to be a reactive control while obstacle avoidance is based on pre-computation of an obstacle free path.

2.2.3 Workspace

Workspace considered for robot path planning in this paper is of size 10 x 10 grid space where the robot, the goal and the obstacles positions are defined by the grid coordinate values. The environment we work is unknown and static obstacles are present and the robot is guided to track the optimal path to travel to the goal.

3. Optimal Path Planning using the Proposed Enhanced A* Algorithm

The environment, in which the grid map is constructed, consisted of a set of states for the robot to make its transition. In this specified environment, the robot has to travel in an optimal path from the starting point to the end point. In the chosen environment, the autonomous mobile robot is capable of traversing in any one direction at a time to reach the target. In the modification that we have proposed in our Enhanced A* algorithm is to identify the quadrant in which the goal is present and then start moving towards that quadrant skipping the other three quadrants.

The robot moves within the grid map in any one of the four neighboring directions i.e., vertically (up/down) or horizontally (left/right) to reach the goal. In our modified algorithm, before reaching the goal, the robot will identify in which quadrant the goal is present relative to the robot's position and move towards the quadrant where the goal is present. Once the goal's quadrant is identified, the robot skips the movements to the other three quadrants and this state of affair is repeated in its entire movement from the start position to the goal position. Since the robot has to move either up, down, left or right, one of the neighboring points on the axes (either x or y axis) towards which the robot will move depends on the robot's current movement towards a quadrant. This process will give a reduced computational time to find the path and to reach the destination as the robots movement has been strictly restricted to move to one of the four neighboring quadrants at each step.

The quadrant where the robot is positioned currently is determined using the following study. The goal's coordinates are defined as (T_x, T_y) and the robot's initial coordinates points as (M_x, M_y). By subtracting the goal coordinate points from the initial coordinate points, the robot determines the quadrant in which the goal is present.

$D_x = (T_x - M_x)$ and $D_y = (T_y - M_y)$ where (D_x, D_y) is the coordinate point which determines the location of the goal's quadrant contained in 2-D x-y space. Further when

$$\begin{aligned} &D_x \geq 0 \text{ and } D_y \geq 0 \Rightarrow \text{goal is in first quadrant,} \\ &D_x \leq 0 \text{ and } D_y \geq 0 \Rightarrow \text{goal is in second quadrant,} \\ &D_x \leq 0 \text{ and } D_y \leq 0 \Rightarrow \text{goal is in third quadrant and} \\ &D_x \geq 0 \text{ and } D_y \leq 0 \Rightarrow \text{goal is in fourth quadrant,} \end{aligned} \quad (1)$$

The next position of the robot movement will be decided by applying Equation 1.

The cost function f(n) related to each position is given by:

$$f(n) = g(n) + h(n) + p(n) \quad (2)$$

Where

g(n) is the generation cost or movement cost for the robot to move from the starting position to the immediate next position in the grid,

p(n) is the number of turns required for the robot before it turns to the direction of movement from current position to immediate next position and h(n) is called as heuristic cost.

Equation 2 defines the main function associated with Enhanced A* Algorithm. In our algorithm, heuristic cost is nothing but the Euclidean distance between the next possible position of the robot and the target position. While the robot is navigating and if there is an obstacle, the robot now has two choices of movement, i.e., turning either to the left or to the right from its current position. A* algorithm is mainly designed to identify an optimal path from a given initial position to a given goal position and it combines uniform-cost search (Dijkstra) and greedy search algorithms. In the proposed Enhanced A* algorithm, we use greedy search algorithm and the heuristic h(n) to estimate the cost to reach the goal which considerably reduces the search cost. But the fact is this search is neither optimal nor complete. Again uniform cost search cuts the cost of the path traverse, i.e., elapsed cost g(n) and this leads to near optimal and complete solution but it may be very inefficient. The two evaluation functions discussed above can be combined to get the advantages of both the methods: uniform-cost search (Dijkstra) and greedy search.

We have modified the A* Algorithm using the function given in Equation 2. The Euclidean distance is considered to be the cost in the heuristic function. Enhanced A* algorithm can also boost up the search speed of the depth-first search algorithm used in A*. Also it uses less memory space as compared to Dijkstra algorithm. To evaluate the complexity of the algorithm, the following quantities are assumed.

b = average out degree of one node and

d = search depth from source to destination.

The complexity of Enhanced A* algorithm is represented as O(bd). Enhanced A* maintains a OPEN list for all possible subsequent steps. The next step is chosen which most likely leads to the goal in a minimal time. In order to achieve the minimal time, we need to have a heuristics which determines the near optimal path to reach the goal. Once the above step is complete, the algorithm leads to CLOSED list. Figure 2 shows the flowchart of the proposed Enhanced A* algorithm.

3.1 Features of the Proposed Algorithm

- The most efficient path to reach the target can be found.
- Searching ability can be improved and
- The Algorithm can be implemented with increased robot speed.

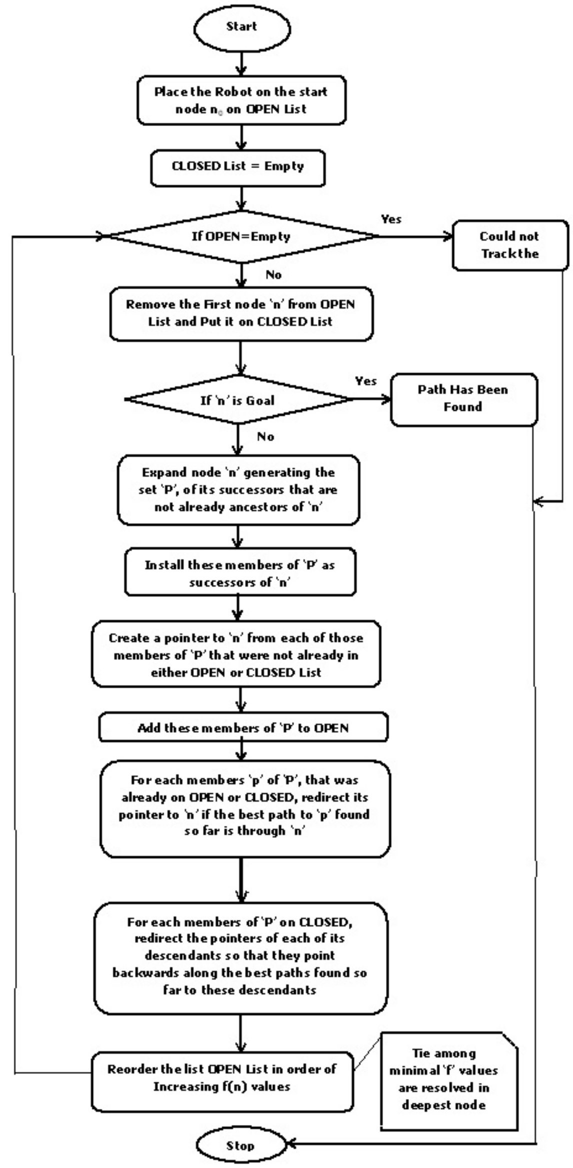

Figure 2. Flowchart depicting enhanced A* algorithm.

4. Performances Analyses

Experimental Analyses were performed using both existing A* Algorithm and the proposed Enhanced A* Algorithm. Both the algorithms are executed through simulation.

4.1 Experimental Setup-1

The Experimental Setup-1 shows one of the tests of both A* Algorithm and the proposed Enhanced A* Algorithm. Figure 3 depicts the trajectory planning for the robot using A* algorithm within a 10 x10 grid environment.

In the environment, the positions of the target and the obstacles are fixed and the obstacles are assumed to be static. The robot is initially positioned at coordinate (0, 0), and the obstacle-1 starting at (2, 4) and extending upto (4, 2). Similarly obstacle-2 starts at (2, 5) and extends upto (6, 5) and the target is located at (7, 6). Once the robot starts moving, it traverses through the optimal path determined by the algorithm and reaches the target. The time traversed by the robot to go from the start to goal position in the Experimental Setup is measured to be 1.922 seconds and the elapsed time is 0.522 seconds. Figure 4 shows the contour diagram of the path followed in Experimental Setup-1 using the A* Algorithm, whereas Figure 5 shows the trajectory planning of the robot using Enhanced A* algorithm within 10 x 10 grid environment. The robot, obstacle-1, obstacle-2 and target are positioned at the same coordinates as used to test the A* Algorithm. The time traversed by the robot to move to the target using Enhanced A* algorithm is 0.505 seconds and the elapsed time is 0.154 seconds.

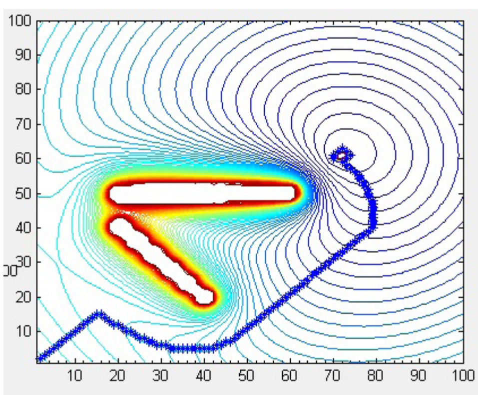

Figure 3. Path traversed using A* Algorithm (Experimental Setup-1).

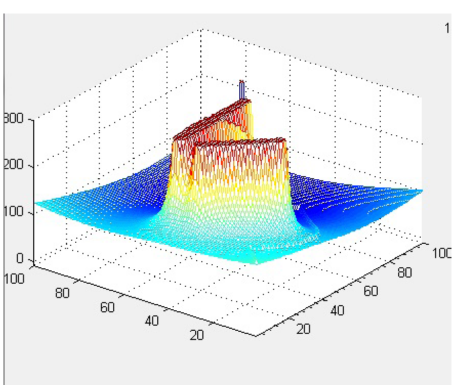

Figure 4. Contour Diagram of the Path traversed by the robot using the A* Algorithm (Experimental Setup-1).

4.2 Experimental Setup-2

In Experimental Setup-2, Figure 6 shows the trajectory traversed using the A* algorithm in 10 x 10 grid environment. In this Setup, the positions of the target and the obstacles are fixed and the obstacles are assumed to be static. The robot is positioned at coordinate (0, 0), obstacle-1 is placed starting at (2, 1) and ending at (2, 5), obstacle-2 at (2, 1) and ending at (5, 1) and the target is located at (4, 3). Once the robot starts moving, it traverses through the optimal path found by the A* Algorithm and reaches the target. The time traversed by the robot is 3.334 sec and the elapsed time is 0.906 seconds. Figure 7 shows the contour diagram of the path followed in Experimental Setup-2 and Figure 8 shows the path traversed by the robot using the proposed Enhanced A* Algorithm within the 10 x 10 grid environment. The robot, obstacle-1, obstacle-2 and the target are positioned at the same coordinate points used for testing the robot implementing the A* Algorithm. The time traversed to reach the target

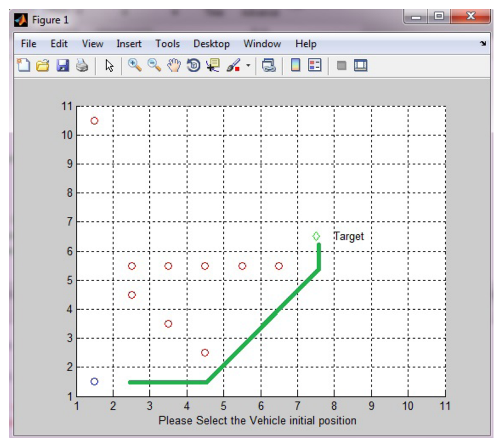

Figure 5. Path traversed by using proposed Enhanced A* Algorithm (Experimental Setup-1).

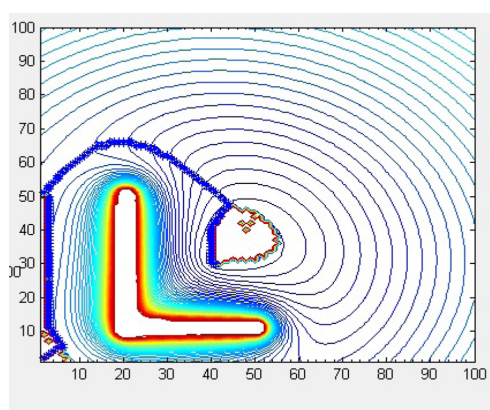

Figure 6. Path traversed using the A* Algorithm (Experimental Setup-2).

using Enhanced A* algorithm is 1.205 seconds and the elapsed time is 0.302 seconds

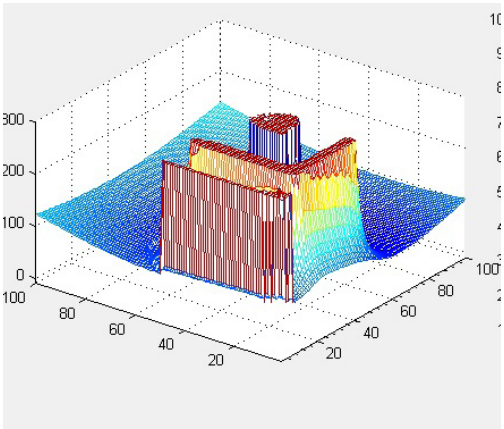

Figure 7. Contour Diagram of the Path traversed by the robot using the A* Algorithm (Experimental Setup-2).

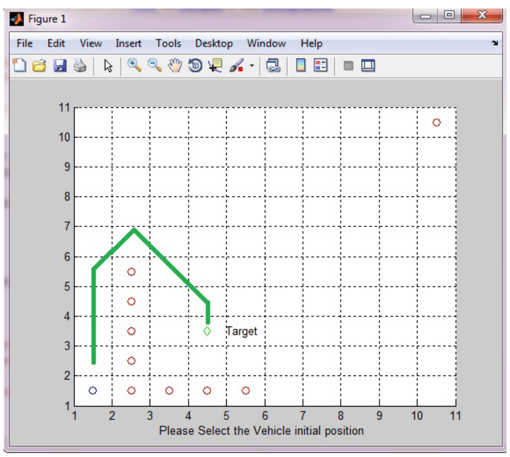

Figure 8. Path traversed using the proposed Enhanced A* Algorithm (Experimental Setup-2).

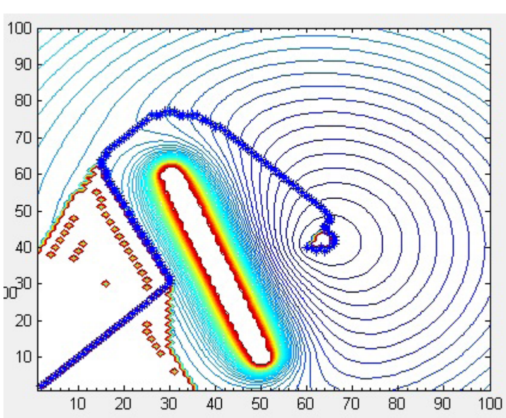

Figure 9. Path traversed using the A* Algorithm (Experimental Setup-3)

4.3 Experimental Setup-3

In Experimental Setup-3, Figure 9 shows the trajectory traversed using the A* algorithm within 10 x 10 grid environment. In this Setup, the positions of the target and the obstacles are fixed and the obstacles are assumed to be static. The robot is positioned at coordinate (0, 0), obstacle-1 at (3, 6) to (5, 1), and the target at (6, 4). The robot starts traversing through the optimal path calculated by the A* Algorithm and reaches the target. The time traversed to reach the goal is 4.671 sec and the elapsed time is 1.671 seconds. Figure 10 shows the contour diagram of the path followed by the robot through the environment and reach the target. Figure 11 shows the path traversed by the robot using the proposed Enhanced A* algorithm within the 10 x 10 grid environment. The robot, obstacle-1, obstacle-2 and the target are positioned at the same coordinates as used to test the A* Algorithm in this setup.

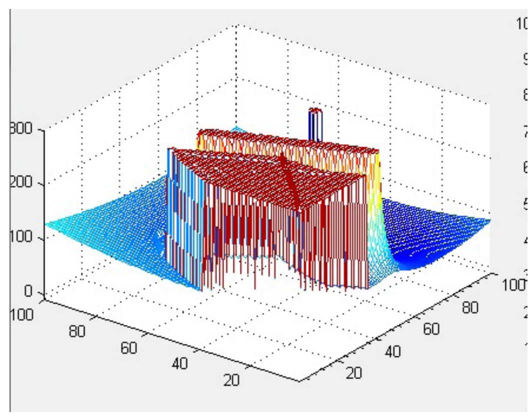

Figure 10. Contour Diagram of the Path traversed by the robot using the A* Algorithm (Experimental Setup-3).

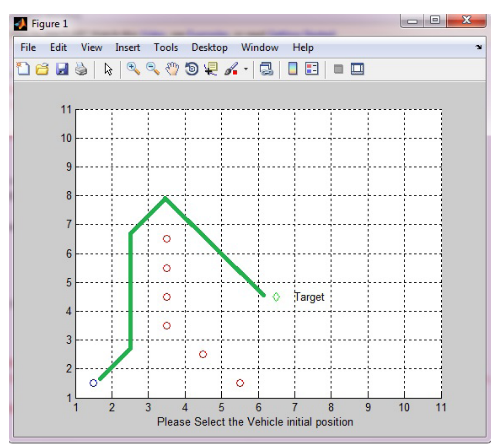

Figure 11. Path traversed using proposed Enhanced A* Algorithm (Experimental Setup-3)

Table 1. Time taken and elapsed time traversed by the robot using the existing A* algorithm and the proposed enhanced A* algorithm to reach the target (experimental setups-1, 2 and 3)

	Time Taken(in sec)		Elapsed Time(in sec)	
	Using Existing Method	Using Proposed Method	Using Existing Method	Using Proposed Method
Experimental Setup-1	1.922	0.505	0.522	0.154
Experimental Setup-2	3.334	1.205	0.906	0.302
Experimental Setup-3	4.671	1.368	1.671	0.512

The time traversed to reach the target using Enhanced A* Algorithm is 1.368 seconds and the elapsed time is 0.512 seconds.

5. Discussion

The mobile robot path planning Experimental Setups-1, 2 and 3 are described and the results are discussed. In the Experimental Setup-1, the obstacles are static and the target is fixed. Once the simulation for the Experimental Setup-1 starts, the robot traverses from its starting point through the optimal path determined by both the existing (A*) and the proposed (Enhanced A*) Algorithms and reach the target. The time taken and the elapsed time to traverse from its starting position and to reach the goal are recorded and given in Table 1. Similarly the scenario for the Experimental Setups-2 and 3 are established with static obstacles and fixed targets. As before, the robot is allowed to traverse and the respective time taken and the elapsed time for the robot to traverse the trajectory and reach target in both the Setups-2 and 3 respectively are found and tabulated. The contour diagrams for all the three Setups for the robot to traverse the path using both the A* and the Enhanced A* Algorithms are given.

Table 1 shows the time taken and the elapsed time traversed from starting position to the goal in Experimental Setups-1, 2 and 3 using the existing A* Algorithm and the proposed Enhanced A* Algorithm. From the simulation results of the Experimental Setups-1, 2 and 3, it is evident to note that the time for the robot to travel to the target using the Enhanced A* Algorithm is drastically reduced as compared to that of the existing A* Algorithm. This is achieved because of introducing a new parameter to the proposed Enhanced A* Algorithm. The new parameter (p(n)) is the number of turns the mobile robot makes in traversing the trajectory to the target. The fact is that the mobile robot turns to right or left and selects the minimum deviated turn from its starting point so as to avoid collision with all the obstacles and to traverse minimal path length. This has caused the algorithm to be more optimal as compared to the existing A* Algorithm as the time traversed is considerably minimised.

6. Conclusion

An Enhanced trajectory planning of an autonomous robot in an environment which is in an unknown terrain is proposed. The terrain contains obstacles which are static and the algorithm is simulated under MATLAB scenario. The trajectory planning and navigation are carried out with the proposed Enhanced A* Algorithm. Time elapsed by the robot to traverse from the initial position to the target using Enhanced A* Algorithm is compared with time elapsed to traverse from the same source to the target using existing A* Algorithm. It is evident that the results obtained using Enhanced A* Algorithm are near optimal than the results obtained using the existing A* Algorithm. From Table 1, the tabulated results show that the elapsed time to traverse from the source location to destination using proposed Enhanced A* Algorithm is much less as compared to the time taken by the robot using the existing A* Algorithm to trace the same path. The modern day real-time requirements of an autonomous mobile robot can be easily met using the newly introduced Enhanced A* algorithm. The proposed algorithm provides a much shorter time of travel as compared to that of the existing A* Algorithm. Hence the Enhanced A* Algorithm can accommodate a provision for generating a better control of the robot and pave the way for expediting an effective contrivance for exploring path planning in robotics education. The Future Enhancement is to make the algorithm work in an unknown environment with dynamic obstacles present and implement the same in real time.

7. References

1. Frank B, Becker M, Stachniss C, Burgard W, Teschner M. Efficient path planning for mobile robots in environments

with deformable objects. IEEE International Conference on Robotics and Automation; Pasadena, CA. 2008. p. 3737-42.

2. Buniyamin N, Ngah WAJW, Sariff NZ, Mohamad A. Simple local path planning algorithm for autonomous mobile robots. International Journal of Systems Applications, Engineering and Development. 2011; 5(2):151-9.
3. Kolushev FA, Bogdanov AA. Multi-agent optimal path planning for mobile robots in environment with obstacles. LNCS: PSI'99; Berlin Heidelberg: Springer-Verlag; 2000; 17(55):503-10.
4. Taharwa IA, Sheta A, Weshah MA. A mobile robot path planning using genetic algorithm in static environment. Journal of Computer Science. 2008; 4(1):341-4.
5. Garrido S, Moreno L, Blanco D, Jurewicz P. Path planning for mobile robot navigation using voronoi diagram and fast marching. IJRA. 2011:42-64.
6. Shojaeipour S, Haris SM, Gholami E, Shojaeipour A. Webcam-based mobile robot path planning using voronoi diagrams and image processing. WSEAS International Conference on Applications of Electrical Engineering; Malaysia. 2010. p. 151-6.
7. Ersson T, Hu X. Path planning and navigation of mobile robots in unknown environments. IEEE/RSJ International Conference onIntelligent Robots and Systems; Maui, HI. 2001. p. 785-864.
8. Ganapathy V, Yun SC, Chien TW. Enhanced D star lite algorithm for autonomous mobile robot. International Journal of Applied Science and Technology. 2011:58-73.
9. Ganapathy V, Yun SCJNG. Fuzzy and neural controllers for acute obstacle avoidance in mobile robot navigation. IEEE/ASME International Conference on Advanced Intelligent Mechatronics Suntec Convention and Exhibition Center; Singapore. 2009. p. 14-7.
10. Konar A, Chakraborty IG, Singh SJ, Jain LC, Nagar AK. A deterministic improved q-learning for path planning of a mobile robot. IEEE Transactions on Systems, Man and Cybernetics: Systems. 2013; 43(5):1141–53.
11. Tsai CC, Huang HC, Chan CK. Parallel elite genetic algorithm and its application to global path planning for autonomous robot navigation. IEEE Transactions on Industrial Electronics. 2011; 58(10):4813–21.
12. Juang CF, Chang YC. Evolutionary-group-based particle-swarm-optimized fuzzy controller with application to mobile-robot navigation in unknown environments. IEEE Transactions on Fuzzy Systems. 2011; 19(2):379–92.
13. Jan GE, Sun CC, Tsai WC, Lin TH. An O(n log n) shortest path algorithm based on delaunay triangulation. IEEE/ASME Transactions on Mechatronics. 2014; 19(2):660- 6.
14. Tamilselvi D, Shalinie SM, Hariharasudan M. Hybrid approach for global path selection and dynamic obstacle avoidance for mobile robot navigation. Advances in Robot Navigation. Mexico: INTECH Open Access Publisher; 2011.
15. Shoushtary MA, Nasab HH, Fakhrzad. Team robot motion planning in dynamics environments using a new hybrid algorithm (honey bee mating optimization-tabu list). Chinese Journal of Engineering. 2014: 1-8.
16. Abdallan TY, Hamzah MI. Trajectory tracking control for mobile robot using wavelet network. International Journal of Computer Applications. 2013; 74(3):32-7.
17. Zhao J, Fu X. Improved ant colony optimization algorithm and its application on path planning of mobile robot. Journal of Computers. 2012; 7(2):2055-62.
18. Cosic A, Susic M, Katic D. Advanced algorithms for mobile robot motion planning and tracking in structured static environments using particle swarm optimization. Serbian Journal of Electrical Engineering. 2012; 9(1):9-28.
19. Das PK, Mandhata SC, Behera HS, Patro SN. An improved Q-learning algorithm for path-planning of a mobile robot. International Journal of Computer Applications. 2012; 51(9):40-6.
20. Ma Z, Ning X. The path planning algorithm and simulation for mobile robot. Journal of Theoretical and Applied Information Technology. 2013; 50(3):601-5.
21. Suresh KS, Vaithiyanathan V, Venugopal S. Layered approach for three dimensional collision free robot path planning using genetic algorithm. Indian Journal of Science and Technology. 2015 Dec; 8(35). Doi no:10.17485/ijst/2015/v8i35/86639
22. Arif A H, Waqas M, Rahman UU, Anwar S, Malik A, Iqbal J. A hybrid humanoid-wheeled mobile robotic educational platform – design and prototyping. Indian Journal of Science and Technology. 2014 Jan; 7(12). Doi no:10.17485/ijst/2014/v7i12/51053
23. Murikipudi A, Prakash V, Vigneswaran T. Performance analysis of real time operating system with general purpose operating system for mobile robotic system. Indian Journal of Science and Technology. 2015 Aug; 8(18). Doi no:10.17485/ijst/2015/v8i19/77017
24. Subramanian MB, Sudhagar K, Rajarajeswari G. Design of navigation control architecture for an autonomous mobile robot agent. Indian Journal of Science and Technology. 2016 Mar; 9(10). Doi no: 10.17485/ijst/2016/v9i10/85769
25. Prasad KM, Reddy ARM, Rao KV. Anomaly based Real Time Prevention of under rated App-DDOS attacks on web: An experiential metrics based machine learning approach. Indian Journal of Science and Technology. 2016 Jul; 9(27). Doi no: 10.17485/ijst/2016/v9i27/87872