I J C T A, 9(37) 2016, pp. 489-502

Path Planning of a Mobile Robot using Amended A-Star Algorithm

Priyanka Sudhakara[*] and Velappa Ganapathy[**]

Abstract: This paper presents an amended new algorithm for path planning of a mobile robot for collision avoidance using optimization techniques. A modified version of the existing A-star algorithm is used to guide the robot in a complex unknown environment with static obstacles to reach the target without any collision. And this amended A-star algorithm is created for an optimal path planning of a mobile robot to reach the target. Improving the existing A-star algorithm is done by adding a new parameter namely the number of turnings ($p(n)$) the robot makes during its traverse. This is done in order to improve the algorithm for optimal motion of the robot. This algorithm will be shown to remove some of the drawbacks of the currently available path planning algorithms. The simulation outcomes of the amended A- star algorithm are compared with the simulation outcomes of the existing A-star algorithm and the results tabulated. From the results, it is shown that the amended A-star algorithm is found to perform much better as compared to the A-star algorithm in terms of Elapsed time of travel.

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

1. INTRODUCTION

Since 1980, mobile robot motion planning problems have become an important research topic that has attracted the attention of many researchers who have worked extensively to obtain efficient methods to solve path planning problems. This issue has been dealt within two general ways: One approach has concentrated on solving motion planning problems using a previously known global environment or obstacle information and the robot characteristics, whilst the second approach has concentrated on planning motion using local sensor information and the robot characteristics. In recent years, sensor-based path planning has emerged as a compromise between the need for reacting to unexpected environmental events and the need for having efficient optimized trajectories. Different variations of classical methods have been proposed to achieve an operative sensor-based path planning. In this paper we have proposed an efficient algorithm namely Amended A* Algorithm for path planning by introducing an additional parameter in the existing A* Algorithm. It will be shown that the proposed Amended A* Algorithm performs much better than the various versions of the existing A* Algorithm.

2. RELATED WORKS

Path planning "enables" mobile robots to see the obstacles and generate an optimum path so as to avoid the obstacles. The general problem of path planning for mobile robots is defined as the search for a path which a robot (with a specified geometry) has to follow in a described environment. The authors in [1], have considered the problem of path planning in environments with non-rigid obstacles such as curtains or plants. They have presented an approach that has combined probabilistic roadmaps with a physical simulation of object deformations to determine a path that optimizes the trade-off between the de-formation cost and the distance to be traveled. An overview of path planning algorithms for autonomous robots is presented in [2]. The authors have focused on the bug algorithm family which is a local path planning algorithm. Bug algorithm uses sensors to detect the nearest obstacle as the mobile robot moves towards a

[*] Research Scholar, School of Computing, SRM University, Kancheepuram, Tamil Nadu, India. *Email: priyanka.k@ktr.srmuniv.ac.in*
[**] Professor, School of Computing, SRM University, Kancheepuram, Tamil Nadu, India. *Email: ganapathy.v@ktr.srmuniv.ac.in*

target with limited information about the environment. The algorithm uses obstacle border as guidance towards the target as the robot circumnavigates the obstacle till it finds certain conditions to fulfill the algorithm criteria to leave the obstacle and traverses towards target point. In addition, they have introduced a novel approach utilizing a new algorithm called Point Bug. This algorithm attempts to minimize the use of outer perimeter of an obstacle (obstacle border) by looking for a few important points only on the outer perimeter of obstacle area so that the robot avoids the obstacle and turns towards the target and finally generates a complete path from source to target. A problem of multi-agent path planning in an environment with obstacles is discussed in [3]. A Novel approach to multi-agent optimal path planning, using graph representation of environment models is described. When planning the path of each robot, the graph model of environment is dynamically changed for path correction and collision avoidance. New algorithm applies changes of robot's paths and moves away so as to avoid collisions in multi-agent environment. In [4], the authors have studied an initial idea for using genetic algorithms to help a controllable mobile robot to find an optimal path between a starting point and ending point in a grid environment. Their proposed controlling algorithm allows four-neighbour movements, so that path-planning can adapt to complicated search spaces with low complexities. A new sensor-based Path Planner algorithm [5], which results in a fast local or global motion planning, which is able to incorporate the new obstacle information. In the first step the safest areas in the environment are extracted by means of a Voronoi Diagram. In the second step, the Fast Marching Method is applied to the Voronoi Diagram and areas extracted in order to obtain the path. The method combines map-based and sensor-based planning operations to provide a reliable motion plan, while it operates at the sensor frequency. A method to navigate a mobile robot using a webcam is discussed in [6]. Their method determines the shortest path for the robot to traverse to its target location, while avoiding obstacles along the way. They have provided a framework for mobile robot navigation using a robot mounted webcam. Using images captured by the webcam, the location of obstacles are identified. Then, using the Voronoi Diagram's technique, the shortest path to the target destination is determined. In [7], the authors 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 environment is modeled as a graph consisting of nodes and arcs. The re-planning problem is solved using the network simplex method. The applicability of the planner is demonstrated by integrating it with a navigation control strategy. The D* Lite Algorithm has been extended in [8] and is named the new algorithm 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 same place again and again. It is capable of re-planning quickly to traverse and also remembering the path. Fuzzy logic and Artificial Neural Network are used in [9], to assist the robot to learn the environment and reach the goal. 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. In [10], the Classical Q-Learning algorithm (CQL) has been extended and the new algorithm named as Improved Q-Learning (IQL). The CQL employs a Q-table to store the Q(S, a) where 'S' are states and 'a' are actions. Thus, it requires an array of $(n \times m)$ size. In the IQL, it is required to store only the Q-values at a state S for the best action. Thus, for n states, it needs to store n number of Q-values. Aside from the Q-storage, in addition, CQL requires 'n' Boolean Lock variables for all states depicting the current status of the particular state. If the Lock variable at a state is 1, then the Q-value at that state need not be updated further. A parallel elite genetic algorithm (PEGA) is presented in [11] and it is applied to global path planning for autonomous mobile robots navigating in structured environments. This PEGA, consisting of two types of PEGAs along with a migration operator, takes advantage of maintaining better population diversity, inhibiting premature convergence, and keeping

parallelism in comparison with conventional Genetic Algorithms (GAs). This initial feasible path generated from the PEGA planner is then smoothed using the cubic B-spline technique in order to construct a near optimal collision-free continuous path. Both global path planner and smoother are implemented in one field-programmable gate array chip utilizing the system-on-a-programmable-chip technology and the pipelined hardware implementation scheme, thus significantly expediting computational speed. In [12], an adaptive-velocity-mutated operation (AVMO) is incorporated to improve search ability. Evolutionary-Group-Based Particle-Swarm-Optimization (EGPSO) uses a group based framework to incorporate crossover and mutation operations in PSO. It dynamically forms different groups to select parents in crossover operations, particle updates and replacements. EGPSO designed Fuzzy Controller is applied to mobile robot in an unknown environment. A behaviour supervisor is proposed to combine the boundary following and the target seeking for navigation. The problem of dead cycles are also considered and accounted. A method that uses the concepts of Delaunay Triangulation, Improved Dijkstra Algorithm and Fermet points to construct a reduced visibility graph with minimal path length has been proposed in [13]. A hybrid method for global optimal path selection and dynamic obstacle avoidance is proposed in [14]. The authors of [14] have used Distance Transform method for global path selection and GJK (Gilebert-Johnson-Keerthi) algorithm for dynamic obstacle avoidance. Whereas in [15], the authors have used honey bee mating optimization (HBMO) algorithm for global optimal path selection and Tabu list technique for dynamic obstacle avoidance. Their simulation outcomes have shown better performance than Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO) algorithms. A new wavelet network control scheme robot path tracking is presented in [16]. PSO method has been used for determining the optimal wavelet neural network (WNN) parameters and the proportional integral derivative (PID) controller parameters for the control of non-holonomic mobile robot that involves trajectory tracking using two optimized WNN controllers one for speed control and the other for azimuth control. The robot is modelled using Matlab Simulink and the PSO algorithm was implemented using MATLAB. The authors have used grid method [17] with coding tactic based on effective vertexes of barriers as the method of environment modelling and ant colony optimization algorithm with two-way parallel searching strategy is adopted to accelerate searching speed. In [18], an approach to robot motion planning in known and static environments, is presented. This problem has been divided into several simpler problems. The first is to generate a collision free path from starting to end point, which is solved using a particle swarm optimization algorithm. The second is interpolation of the obtained collision-free path, which was solved using a radial basis function neural network, and trajectory generation, based on the interpolated path. The last was a trajectory tracking problem, which is solved using a proportional-integral controller. An alternative approach to Q-learning to reduce the convergence time without using the optimal path from a random starting state to a final goal state, when the Q-table is used for path planning of a mobile robot is discussed in [19]. Further, their proposed algorithm stores the Q-value for the best possible action at a state, and thus saves significant storage. Their proposed algorithm is then compared with classical Q-learning and is studied using Khepera-II robot. Cloud theory is combined with rough set in [20] to use in path planning of mobile robot. Their existing evolutionary algorithm's shortcoming is precocity and in order to overcome the shortcoming they have improved the speed and accuracy of the robot navigation using their algorithm.

Considering all the above works, with a view to improve the efficiency of robot navigation, we have proposed some changes in the A* Algorithm by introducing a new parameter to the algorithm, namely the number of turnings ($p(n)$) that the robot makes during its traverse in order to improve the optimal motion of the robot.

3. SYSTEM OVERVIEW

The path planning problem of a mobile robot is solved using a modified version of the real time A* algorithm in an unknown environment. Since the A* algorithm is generally considered as superior to other GA based solutions with specified constraints, we have modified the real time A* algorithm to solve the path planning problem instead of using GA based approach to optimize energy, execution time and distance travelled. In this paper, the well-known heuristic real time A* algorithm is also implemented to make the mobile robot navigate through static obstacles and find the shortest path from an initial position to the target position by avoiding the obstacles. The proposed Amended A* Algorithm for path finding strategy is designed in a grid-map form of an unknown environment with unknown static obstacles based on the quadrant concept in which the goal is present. This algorithm optimizes the path since the goal is present in any one of the four quadrants and we have restricted the movement of the robot to that quadrant. When the path planning algorithm is executed, it is necessary to plan an optimal or feasible path for itself avoiding obstructions in its way and minimizing the time, energy, and distance. In our study, we have considered the distance and time metrics as the cost function which automatically implies that energy saving is also under way.

A. Problem Formulation and Outline of the Solution

The mobile robot path planning problem is typically formulated as follows: given a mobile robot and a description of an environment, we need to plan a path between two specified locations, the start and end points. The path should be free of collision and it should satisfy certain optimization criteria (i.e., shortest path, collision avoidance, etc.). According to this definition, path planning problem can be categorized under NP-hard optimization problem.

Researchers distinguish between various methods used to solve the path planning problem according to two factors:

1. The type of environment (i.e., static or dynamic), and
2. The path planning algorithm (i.e., global or local).

The static environment is defined as the environment which does not contain any moving objects other than a navigating robot; while the dynamic environment is the environment which has moving objects (i.e., human beings, moving machines, moving robots, etc.).

The global path planning algorithms require a complete knowledge about the search environment and that all terrains should be static. On the other hand, local path planning means that path planning is being implemented while the robot is moving; in other words, the algorithm is capable of producing a new path in response to the environmental changes. 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 takes the shortest path.

B. Framework

1. Block Diagram of the System

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

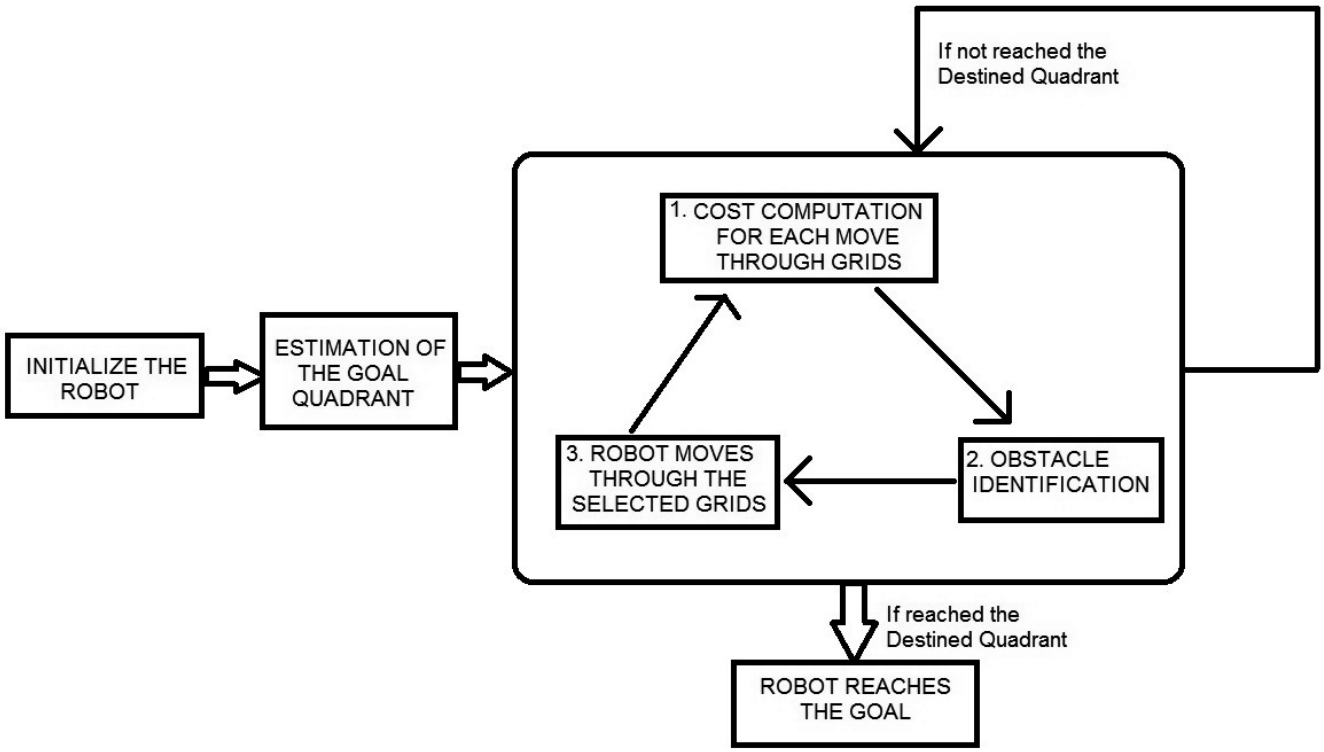

Figure 1: Architectural Block Diagram of the Proposed Robot Navigation System

2. Module Split-Ups

Goal Tracking: The goal is tracked by computing the co-ordinates, the difference between start coordinates and the goal coordinates as per the proposed Amended A* Algorithm.

Cost Computation: The distance cost computation is the estimation of distance travelled from start point to the end point.

Path Planning: A basic motion planning problem is to produce a continuous motion that connects a start configuration S and a goal configuration G, while avoiding collisions with known obstacles. The robot and obstacle geometries are described in both 2D and 3D workspaces, while the motion is represented as a path in (possibly higher-dimensional) configuration space.

Obstacle Detection: In the field of robotics, obstacle avoidance is the task of satisfying some control objectives, subject to non-intersection or non-collision position constraints. In unmanned air vehicles, this is a hot topic. What is critical about obstacle avoidance concept in this area is the growing need of usage of unmanned aerial vehicles in urban areas especially in military applications where it can be very useful for logistics of man, machines etc. Normally obstacle avoidance is considered to be distinct from path planning in that one is usually implemented as a reactive control law while the other involves the pre-computation of an obstacle-free path through which a controller will guide the robot along.

C. Workspace

The workspace considered for robot path planning in this paper is a size of 10x10 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 reach the target.

4. OPTIMAL PATH PLANNING USING THE PROPOSED AMENDED A* SEARCH ALGORITHM

The environment, where the grid map is constructed, consists of a set of states and the grid map is built for the navigation of robot. In a grid map the mobile robot has to find the path from its given initial position to the goal position (which is known to the robot) so that the robot moves in an optimal way. Generally in the grid map, the robot is capable of moving in any one direction at a time to reach the goal. We have modified the algorithm such that the robot will first check, in which quadrant the goal is present and then start moving towards that quadrant skipping the other three quadrants.

Generally in the grid map the robot moves in any one of the four neighboring directions i.e., vertically (up/down) or horizontally (left/right) to reach the goal. To find the path to the goal, we have modified the algorithm such that the robot will first check in which quadrant the goal is present. Hence it will skip the movement of the robot to the other three quadrants at each step of movement and it considers only one quadrant every time for finding the path to the goal. By this way, we have to consider only two possible neighboring points of (x, y) on the axes. This process will give a reduced computational time to find the path and to reach the goal since we have restricted the movement of the robot at each step to only one quadrant.

The following equation is used to find the position of the goal i.e., in which quadrant the goal is present now. Let us consider the coordinate point of the goal to be (T_x, T_y) and the initial coordinate point of the robot (M_x, M_y).

The robot computes the quadrant in which the goal is by subtracting the goal coordinate point and initial coordinate point of the robot. Let $D_x = (T_x - M_x)$ and $D_y = (T_y - M_y)$ where (D_x, D_y) is the coordinate point of the current position of the robot.

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

Now our algorithm will decide the next movement of the robot position based on the above Equation 1. Each position has an associated cost function as shown below:

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

Equation 2 defines the Amended A* Algorithm, where g(n) is the generation cost or movement cost from the starting position to next position in the grid, p(n) is the number of turns required for the robot to rotate pointing to the direction of movement from current position to next position and $h(n)$ is the estimated movement cost from the neighbour position of the robot to the target position. $h(n)$ is called as heuristic cost. There are many different ways of defining the heuristic cost. In our implementation, the heuristic cost is the Euclidean distance between the next possible position of the robot and the target position. The robot selects the next position having minimum $f(n)$ value from the two choices, if there is no obstacle. If there is an obstacle, then the robot moves in the other direction. A* is a graph search algorithm that finds a path from a given initial node to a given goal node. The A* algorithm stands by combining the greedy search and uniform-cost search (Dijkstra) algorithms. Greedy search minimizes the estimated cost to reach the goal (that is the heuristic, $h(n)$), and thereby cuts the search cost considerably; but it is neither optimal nor complete. On the other hand, uniform-cost search minimizes the cost of the path so far (that is elapsed cost, $g(n)$); it is near optimal and complete, but can be very inefficient. These two methods can be combined by simply summing the two evaluation functions to get advantages of both the methods.

We have modified the A* Algorithm and given in (2) by considering the heading direction as the cost in the heuristic function. Amended A* search algorithm can enhance the search speed of the depth-first search algorithm. It uses less memory space than Dijkstra algorithm, the average out degree of one node is marked as "*b*", the search depth of the shortest path from start point to end point noted as "*d*", then the time complexity of Amended A* search algorithm is represented as O(bd). In the path planning problem for finding the optimal path, the algorithm starts from a point, START (source), to another point, GOAL (destination). Amended A* keeps a list of all possible subsequent steps, called the OPEN list. It then chooses the next step that is most likely to lead us to the goal in the minimum time. In order to accomplish this we need to have a heuristic that will determine the near optimal path to reach the goal. Once that step has been chosen, it is moved to the CLOSED list. Figure 2 shows the flowchart of the proposed amended A* algorithm.

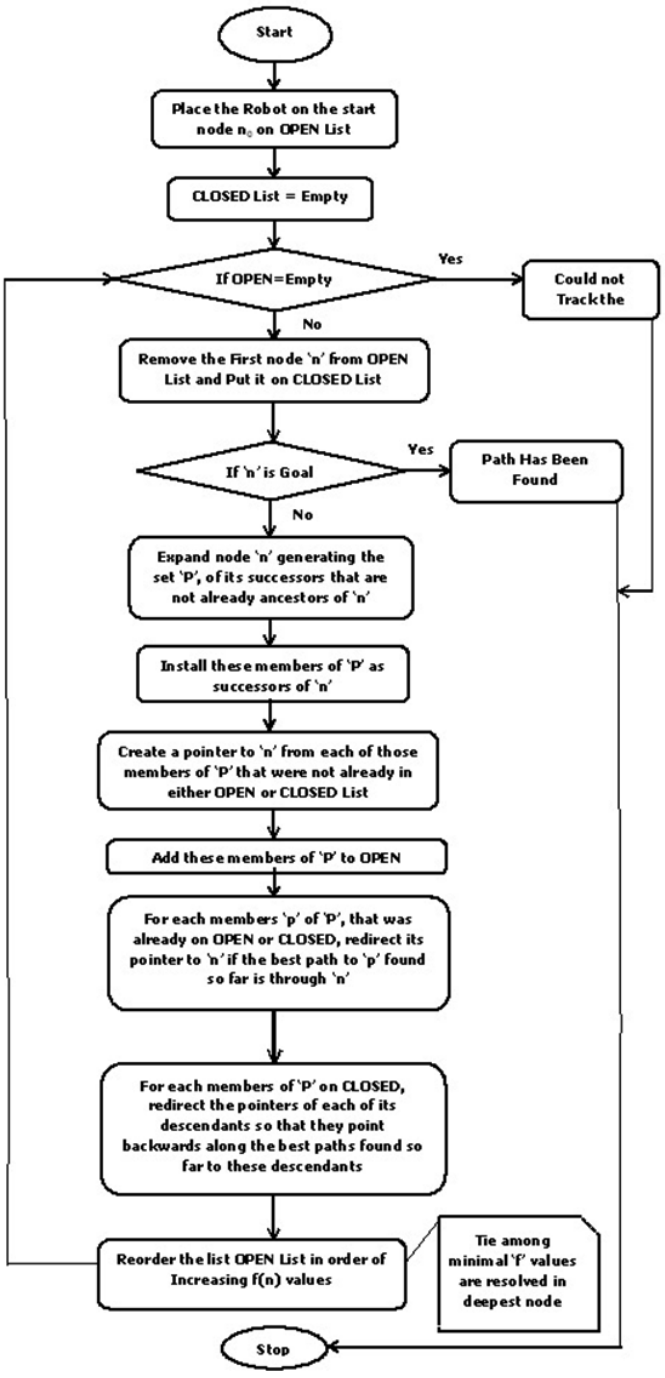

Figure 2: Flowchart of the Proposed Amended A* Algorithm

Features of the Proposed Algorithm:

1. The most efficient path to reach the target can be found.
2. Searching ability can be improved.
3. The Algorithm can be implemented with increased Robot speed.

5. EXPERIMENTAL RESULTS AND ANALYSIS

The Experimental Analyses were carried out using both the existing A* Algorithm and the proposed Amended A* Algorithm. Both the algorithms are executed through simulation.

A. Experimental Setup 1

The Experimental Setup 1 shows one of the tests of both A* Algorithm and the proposed Amended A* Algorithm. Figure 3 shows the path planning of the robot using the A* algorithm within a 10 × 10 grid environment. Here the obstacles are static and the positions of the obstacles and the target are fixed. The robot is initially positioned at cordinate (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, it traverses through the optimal path determined by the algorithm and reaches the target. The time taken for the robot to start and reach the target 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 path planning of the robot using the proposed algorithm within the 10 × 10 grid environment. The robot, obstacle-1, obstacle-2 and target are positioned at the same cordinates as used to test the A* Algorithm. The time taken for the robot to reach the target using proposed algorithm is 0.505 seconds and the elapsed time is 0.054 seconds.

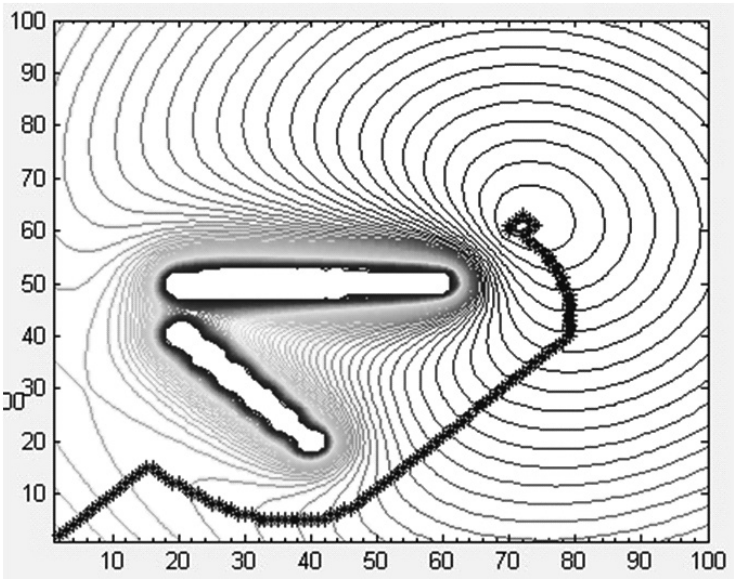

Figure 3: Path traversed by the robot to reach the target using the A* Algorithm (Experimental Setup 1)

B. Experimental Setup 2

In Experimental Setup 2, Figure 6 shows the path traversed by the robot using the A* algorithm within the 10 × 10 grid environment. Here the obstacles are static and the positions of the obstacles and the target

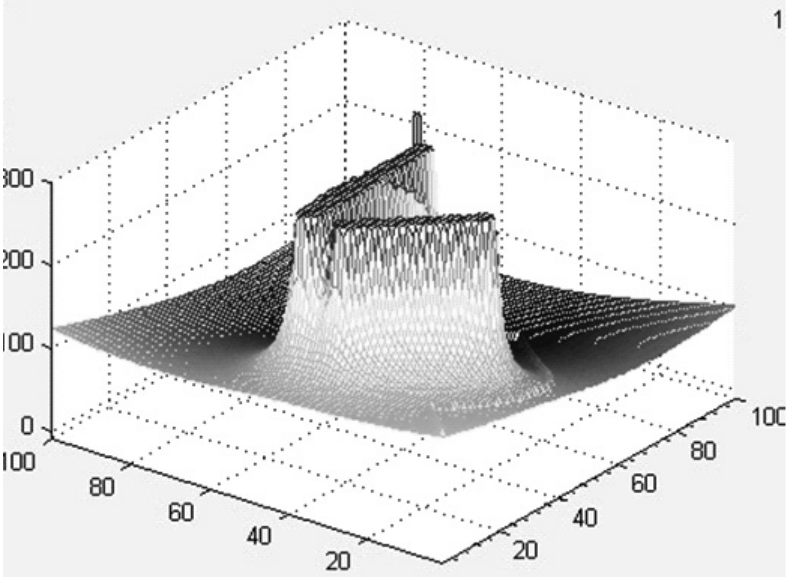

Figure 4: Contour Diagram of the Path traversed by the robot using the A* Algorithm (Experimental Setup 1)

are fixed. The robot is positioned at cordinate (0, 0), obstacle-1 is placed starting at (2, 1) and ending at (2, 5), obstacle-2 at (2.2, 1) and ending at (5, 1) and the target is located at (4, 3). Once the robot starts, it traverses through the optimal path found by the A* Algorithm and reaches the target. The time taken for the robot to reach the target is 3.334 seconds 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 Amended A* Algorithm within the 10 × 10 grid environment. The robot, obstacle-1, obstacle-2 and the target are positioned at the same cordinate points used for testing the robot implementing the A* Algorithm. The time taken for the robot to reach the target using proposed algorithm is 1.205 seconds and the elapsed time is 0.002 seconds.

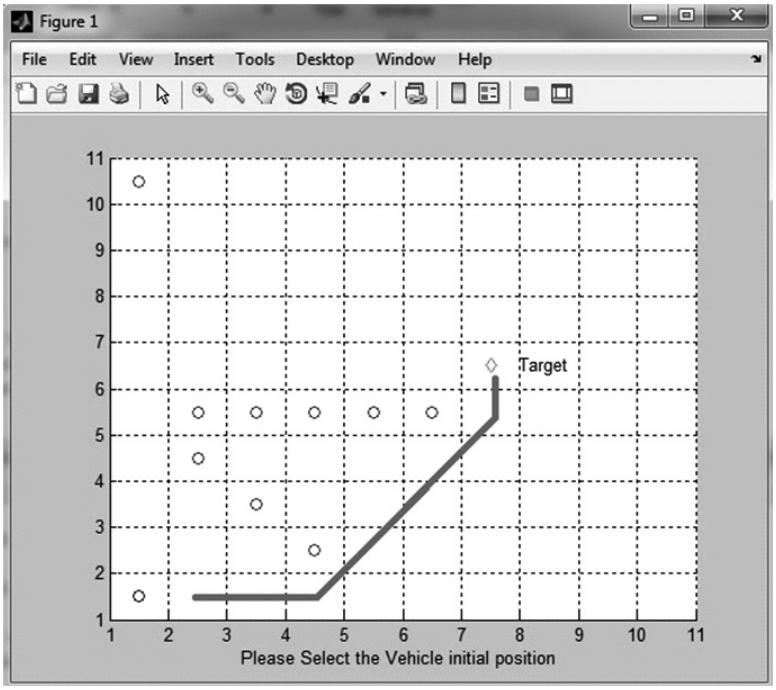

Figure 5: Path traversed by the robot to reach the target using the proposed Amended A* Algorithm (Experimental Setup 1)

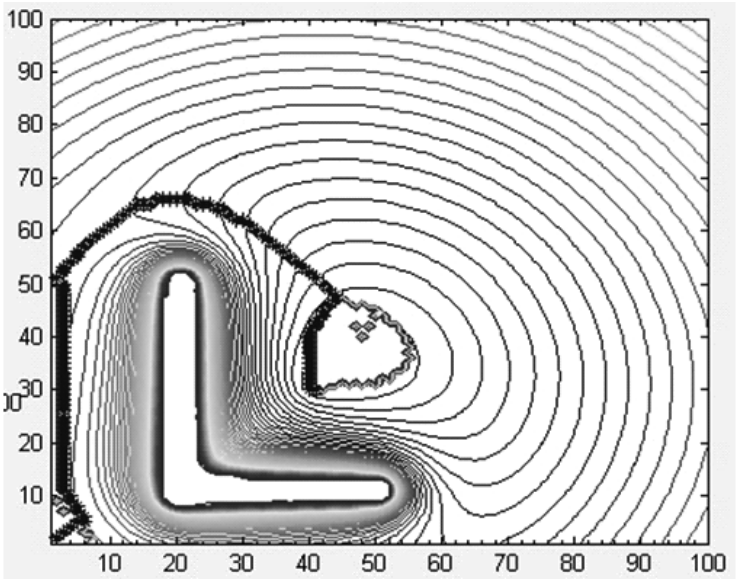

Figure 6: Path traversed by the robot to reach the target using the A* Algorithm (Experimental Setup 2)

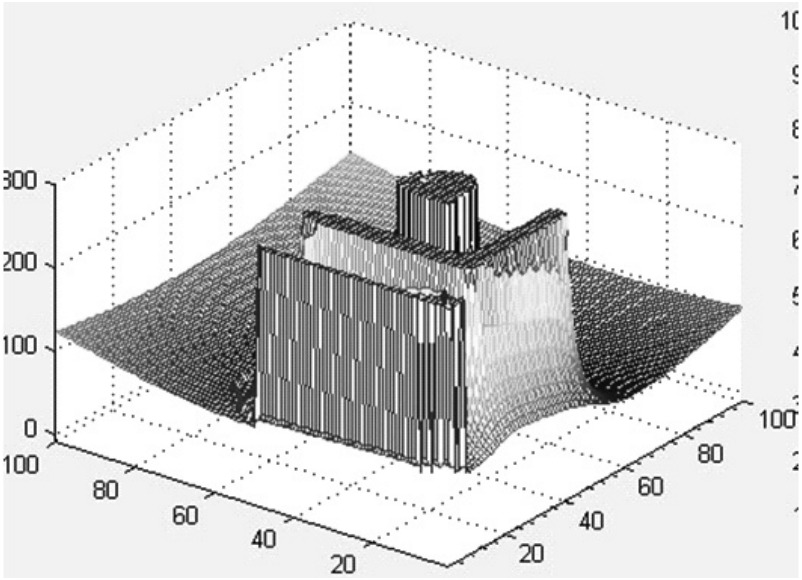

Figure 7: Contour Diagram of the Path traversed by the robot using the A* Algorithm (Experimental Setup 2)

C. Experimental Setup 3

In Experimental Setup 3, Figure 9 shows the path traversed by the robot using the A* Algorithm within the same 10 × 10 grid environment. Here the obstacles are static and the positions of the obstacles and the target are fixed. In this setup, the robot is positioned at cordinate (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 taken for the robot to reach the target is 4.671 seconds 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 Amended A* algorithm within the 10 × 10 grid environment. The robot, obstacle-1, obstacle-2 and the target are positioned at the same cordinates as used to test the A* Algorithm in this setup. The time taken for the robot to reach the target using proposed Amended A* Algorithm is 0.368 seconds and the elapsed time is 0.012 seconds.

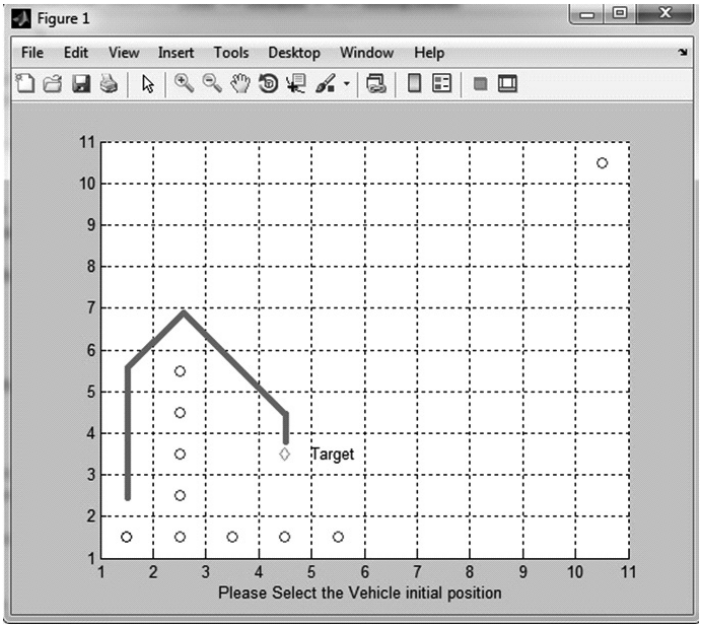

Figure 8: Path traversed by the robot to reach the target using the proposed Amended A* Algorithm (Experimental Setup 2)

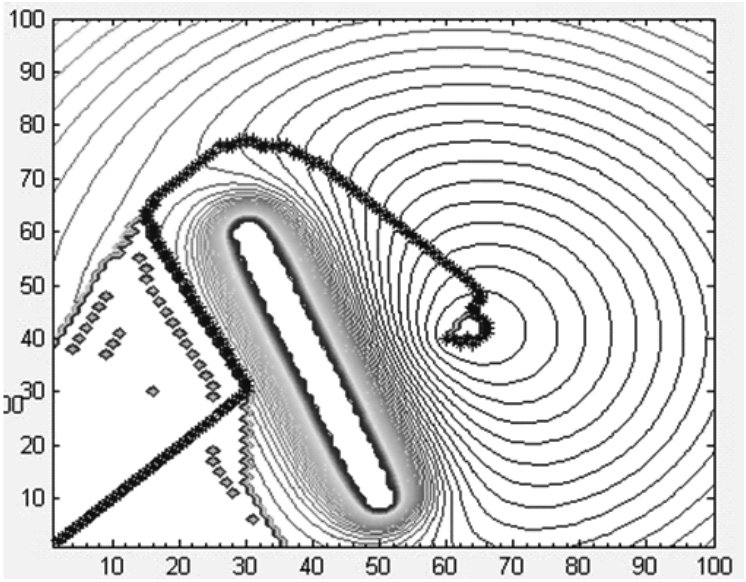

Figure 9: Path traversed by the robot to reach the target using the A* Algorithm (Experimental Setup 3)

6. 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 from its starting point traverses through the optimal path determined by both the existing (A*) and the proposed (Amended A*) Algorithms and reach the target. The time taken and the elapsed time for the robot to start from its initial position and to reach the target are measured and given in the Table I. 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 from starting point to the target and the respective time taken and the elapsed time for the mobile robot to traverse the path and reach the target for both the Setups 2 and 3 respectively are found and tabulated. The contour diagram for all the three setups for the robot to traverse the path using both the A* and the Amended A* Algorithms.

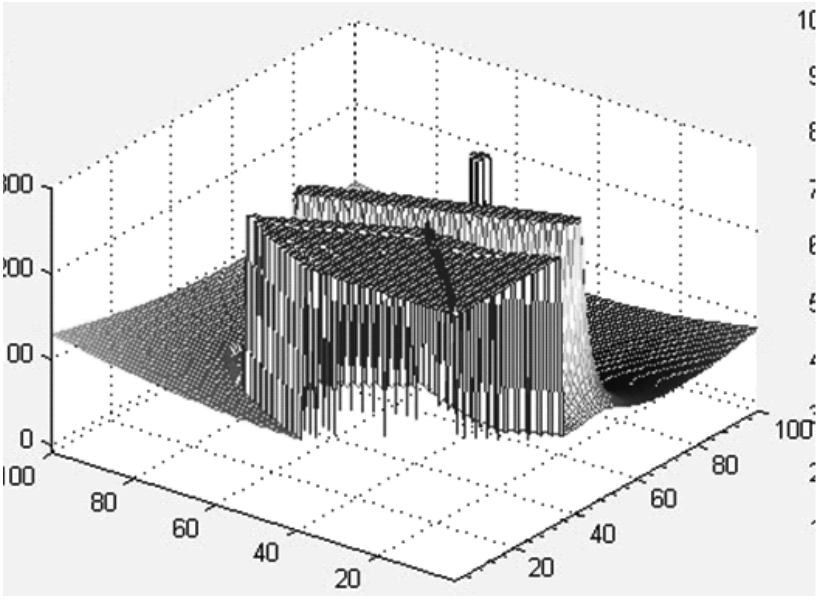

Figure 10: Contour Diagram of the Path traversed by the robot using the A* Algorithm (Experimental Setup 3)

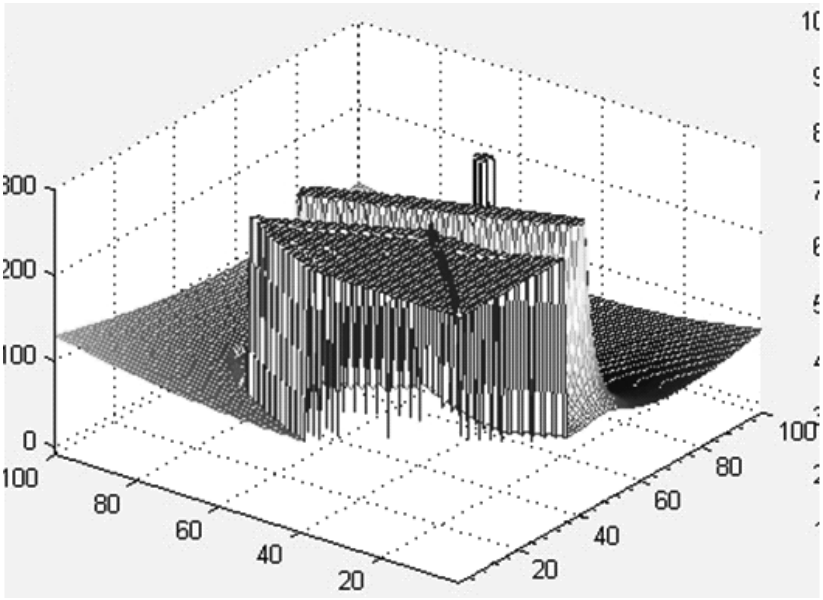

Figure 11: Path traversed by the robot to reach the target using the proposed Amended A* Algorithm (Experimental Setup 3):

The Table I shows the time taken and the elapsed time traversed by the robot from starting point to the target in the Experimental Setups 1, 2 and 3 using the existing A* Algorithm and the proposed Amended A* Algorithm. From the simulation results of the Experimental Setups 1, 2 and 3, we have found that the time of travel of the robot to reach the target using the Amended A* Algorithm is drastically reduced as compared to that of the existing A* Algorithm. This is achieved by introducing a new parameter to the proposed Amended A* Algorithm. The new parameter ($p(n)$) is the number of turns the mobile robot makes in traversing to the goal. It is observed that the robot turns to its right and 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 taken of the robot to reach the target is considerably minimised.

Table 1
Time Taken And Elapsed Time Traversed By The Robot Using The Existing A* And The Proposed Amended A* Algorithms 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	0.102
Experimental Setup 3	4.671	1.368	1.671	0.512

7. CONCLUSION

In this paper, an amended path planning of a mobile robot within an unknown environment containing static obstacles, is presented and simulated under MATLAB environment. The path planning and navigation are performed using the proposed Amended A* Algorithm. The time elapsed for the robot to travel from the starting point to the target using the Amended A* Algorithm has been compared with the time elapsed for the robot to travel from the same starting point to the target using the existing A* Algorithm. It is evident that the results obtained using Amended A* Algorithm are near optimal than the results obtained using the existing A* Algorithm. From Table I, the tabulated results show that the elapsed time for the mobile robot to travel from the starting location to the destination using the proposed Amended 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 use of the Amended A* Algorithm can meet the rapid and real-time requirements of the modern day robots' path planning problems. The proposed algorithm provides a much shorter time of travel as compared to that of the existing A* Algorithm. Hence the Amended A* Algorithm provides a platform for developing a better robot control and also provides an effective tool 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 using an autonomous mobile robot.

References

1. B. Frank, M. Becker, C. Stachniss, W. Burgard and M. Teschner, "*Efficient path planning for mobile robots in environments with deformable objects*," IEEE International Conference on Robotics and Automation, pp. 3737-3742, May 2008.
2. N. Buniyamin, W. A. J. Wan Ngah, N. Sariff and Z. Mohamad, "*A simple local path planning algorithm for autonomous mobile robots*," International Journal of Systems Applications, Engineering & Development, Vol. 5, pp. 151-159, 2011.
3. F. A. Kolushev and A. A. Bogdanov, "*Multi-agent optimal path planning for mobile robots in environment with obstacles*," PSI'99, LNCS 1755, pp. 503-510, Springer-Verlag Berlin Heidelberg 2000.
4. I. A. Taharwa, A. Sheta and M. A. Weshah, "*A mobile robot path planning using genetic algorithm in static environment*," Journal of Computer Science, Vol. 4, pp. 341-344, 2008.
5. S. Garrido, L. Moreno, D. Blanco and P. Jurewicz, "*Path planning for mobile robot navigation using voronoi diagram and fast marching*," International Journal of Robotics and Automation (IJRA), Vol. 2, pp. 42-64, 2011.
6. S. Shojaeipour, S. M. Haris, E. Gholami and A. Shojaeipour, "*Webcam-based mobile robot path planning using voronoi diagrams and image processing*", WSEAS International Conference on Applications of Electrical Engineering, pp. 151-156, 2010.
7. T. Ersson and X. Hu, "*Path planning and navigation of mobile robots in unknown environments*", IEEE/RSJ International Conference on Intelligent Robots and Systems, Vol. 2, pp. 858 – 864, 2001.
8. V. Ganapathy, S. C. Yun, and T. W. Chien, "*Enhanced D* lite algorithm for autonomous mobile robot*", International Journal of Applied Science and Technology, Vol. 1, pp. 58-73, March 2011.
9. V. Ganapathy, S. C. Yun, and J. Ng, "F*uzzy 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, July 14-17, 2009.

10. A. Konar, I. G. Chakraborty, S. J. Singh, L. C. Jain, and A. K. Nagar, "*A deterministic improved q-learning for path planning of a mobile robot*", IEEE Transactions On Systems, Man, And Cybernetics: Systems, Vol. 43, pp. 1141 – 1153, September 2013.

11. C. C. Tsai, H. C. Huang and C.K. Chan, "*Parallel elite genetic algorithm and its application to global path planning for autonomous robot navigation*", IEEE Transactions On Industrial Electronics, Vol. 58, pp. 4813 – 4821, October 2011.

12. C. F. Juang and Y. C. Chang, "*Evolutionary-group-based particle-swarm-optimized fuzzy controller with application to mobile-robot navigation in unknown environments*", IEEE Transactions On Fuzzy Systems, Vol. 19, pp. 379 – 392, April 2011.

13. G. E. Jan, C. C. Sun, W. C. Tsai and T. H. Lin, "*An O(n log n) shortest path algorithm based on delaunay triangulation*", IEEE/ASME Transactions On Mechatronics, Vol. 19, pp. 660- 666, April 2014.

14. Tamilselvi, S. Mercy Shalinie, and M. Hariharasudan, "*Hybrid approach for global path selection and dynamic obstacle avoidance for mobile robot navigation*" in INTECH Open Access Publisher "*Advances in Robot Navigation*", Mexico.

15. M. A. Shoushtary, H. H. Nasab, and M. B. Fakhrzad, "*Team robot motion planning in dynamics environments using a new hybrid algorithm (honey bee mating optimization-tabu list)*", Hindawi Publishing Corporation, Chinese Journal of Engineering, Vol. 2014, May 2014

16. T. Y. Abdallan and M. I. Hamzah, "*Trajectory tracking control for mobile robot using wavelet network*", International Journal of Computer Applications, Vol. 74, pp. 32-37, July 2013.

17. J. Zhao and X. Fu, "*Improved Ant Colony Optimization Algorithm and Its Application on Path Planning of Mobile Robot*", Journal Of Computers, Vol. 7, pp. 2055-2062, August 2012.

18. A. Cosic, M. Susic and D. Katic, "*Advanced algorithms for mobile robot motion planning and tracking in structured static environments using particle swarm optimization*", Serbian Journal Of Electrical Engineering, Vol. 9, pp. 9-28, February 2012.

19. P. K. Das, S. C. Mandhata, H. S. Behera and S. N. Patro, "*An improved Q-learning algorithm for path-planning of a mobile robot*", International Journal of Computer Applications, Vol. 51, pp. 40-46, August 2012.

20. Z. Ma and X. Ning, "*The path planning algorithm and simulation for mobile robot*", Journal of Theoretical and Applied Information Technology, Vol. 50, pp. 601-605, April 2013.