IJCTA, 8(3), 2015, pp. 1053-1061

Occupancy Grid Based Path Planning and Terrain Mapping Scheme for Autonomous Mobile Robots

C. Saranya[1], Manju Unnikrishnan[2], S. Akbar Ali[3], D. S. Sheela[4] and Dr. V.R. Lalithambika[5]

Abstract: Autonomous mobile robots operate under partially or fully unknown environments. Based on sensor information robot identifies various obstacles and plans its path to reach the target without collision. These real time systems may not support extensive approaches for Path Planning and Terrain Mapping, due to their limited processing and memory capabilities. This paper proposes a simple distance based path planning scheme to avoid obstacles and reach the target. Also a modified DP SLAM based approach is used to generate an occupancy grid based terrain map of the region explored. These algorithms were simulated in C and evaluated with different test scenarios. Results of simulation are also briefed.

Key words: Path Planning, Terrain Mapping, SLAM, Autonomous navigators.

1. INTRODUCTION

Autonomous mobile robots are destined to explore unknown terrain or to reach a set target point with partial or no prior information about obstacles. As the rover moves it senses its environment and identifies the obstacles. Based on the obtained information, rover updates its terrain map and also re-plans its path to avoid obstacles, if any.

Literature suggests several schemes for path planning. Ref 1 deals with different path planning approaches. Classical approaches include grid based algorithms like A* and D*. Ref 2 and 3, briefs about A* and D* algorithms. Potential field method and Rapidly Exploring Random Trees are other approaches to Path Planning. Ref 4 and 5, talk about these techniques. Ref 6 gives a comparison of the various path planning techniques. A* method involves Heuristics for cost computation. Also it is efficient only for static and known environments. D* algorithm efficiently computes path for dynamic and unknown environments. But whenever new obstacles are identified, cost computation is redone. This requires large memory and processing speed which increases proportionally with the area under exploration. Potential field method may fail under a trap situation and Rapidly Exploring Random Trees has large memory requirements. In this paper a simple grid based method based on Distance measurement is used for planning path. This method enables to reach target point in the shortest possible route. Also the algorithm suits implementation in real time systems with limited memory and processing capabilities.

[1] Scientist/Engineer, CGFSD, saranya_c@vssc.gov.in
[2] Deputy Division Head, CGFSD,
[3] Division Head, CGFSD,
[4] Group Director, Control & Guidance Design Group,
[5] Deputy Director, Control Guidance & Simulation Entity, Vikram Sarabhai Space Centre, India

Simultaneous Localisation and Mapping (SLAM) enable to estimate the current position of the rover while building a map of the terrain explored. Widely used SLAM techniques are Extended Kalman filter based and particle filter based. Particle filter have computational advantage over Extended Kalman filter. But the number of particles required may be more. Here, a modified version of DP SLAM particle filter based technique is used to generate occupancy grid of the region explored. This method uses less number of particles and a single occupancy grid is used to represent the terrain. Hence it can be implemented for real time computation even with limited memory and processing resources. Ref 7, 8 and 9 deals with DP SLAM and occupancy grid based SLAM method.

This paper also brings out the simulation and validation of algorithm. This paper is categorized as follows: Section II gives a description of Distance based Path Planning Algorithm and Section III describes modified DP SLAM algorithm. Section IV describes the simulation and section V briefs the results of a typical test case. Section VI presents the concluding remarks.

2. PATH PLANNING ALGORITHM

Path planning or Motion Planning algorithm aims to navigate the rover to its target point in spite of the uncertainties of its environmental conditions. In this paper, grid based technique using distance to be travelled as the cost metric is designed to plan the path of the rover.

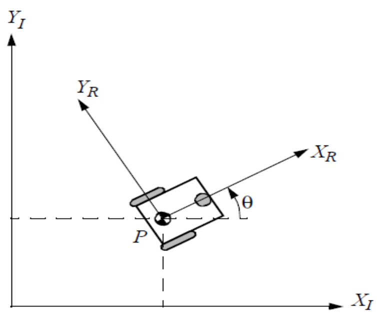

Figure 1: States of the Rover

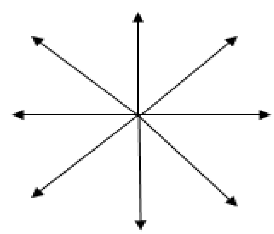

Figure 2: Eight possible orientations of Rover

Rover is aware of its starting coordinates and the target coordinates. It is assumed that the rover is completely unaware of the obstacles present in its environment. As the rover moves towards the target, it senses the adjoining regions and updates its terrain map. It is essential to estimate the position of the rover at each instant in order to keep track of its motion and also to build the terrain map. The rover has three states [X, Y, θ] as in figure 1. Its current position is given by X, Y coordinates while its orientation with the global frame of reference is given by θ. The motion of rover along its local frame of reference can be transformed to the global reference frame using the transformation matrix as in equation 1.

$$R(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

The motion planning algorithm logically splits the region of exploration into cells of size a x a. Since occupancy grid based representation is used for the generation of terrain map in this paper, grid based approach is used for path planning also. Based on the current position and the target coordinates, the distance to be travelled is computed as in equation 2.

$$D = \sqrt{(T_x - C_x)^2 + (T_y - C_y)^2} \quad (2)$$

Where Tx, Ty denotes the target coordinates while Cx Cy denotes the current coordinates of the rover. The required orientation is computed as given in equation 3.

$$\theta c = \tan^{-1} {(T_y - C_y)}/{(T_x - C_x)} \tag{3}$$

Since the region is split into grids, rover can move to only one of its adjacent eight cells at each step. Thus the rover can have one of the eight possible orientations or θ, as in figure 2. Assume that the rover is in the centre position in figure 2. From its current cell, it can move to any one of the adjacent eight cells depending on its orientation. Since movement to only adjacent eight cells are possible, orientation of the rover varies from 0 to 315 degrees, in steps of 45 degrees.

Based on the computed distance and orientation, rover moves ahead, provided there are no obstacles on its way. When obstacles are identified, rover changes its orientation in increments of 45 deg until there are no obstacles and continues to move further by avoiding the obstacle. On reaching an obstacle free zone, rover reorients itself to the target direction and proceeds further, until encountering the next obstacle. The pseudo code of the Motion Planning algorithm is given below

Until Target Reached

```
Compute Distance to target, D
Compute Required Orientation, θc
Rotate by θ – θc
Move Forward until obstructed
If Obstacle is identified
    Re-orient in steps of 45 deg wrt Current Theta, θ
    Move Forward to next cell provided no obstacle
End if
End
```

In order to successfully navigate out of a trap situation, path traced by the rover is stored. When the rover toggles between two cells, it moves back to previous cell and moves to another cell which was not chosen in the past to avoid the trap situation.

3. TERRAIN MAPPING ALGORITHM

Simultaneous Localisation And Mapping (SLAM) Techniques aims at building a terrain map while simultaneously determining the position of the rover. Extended Kalman Filter based SLAM approaches are computationally extensive. They are not suited for implementation in real time systems. Graph based techniques address the SLAM problem, but in offline mode.

Particle filter approach involves less computational time compared to Kalman Filter technique. Each Particle represents the possible trajectory of the rover. Each particle maintains its own map. Ref 10 explains the DP SLAM algorithm. The major steps of the algorithm are briefed below:

1. Based on the encoder value, position of each particle gets updated.
2. Observations are made by sensing the environment.
3. Each particle updates its map based on the sensed information
4. Particles are weighed based on its ability to predict the known observations. Particles with lower probability are deleted and those with higher probabilities are retained.

Two main representations of the map are Landmark and Occupancy grid. DP SLAM uses Occupancy grid technique. This is a simple method involving less memory requirements. Further it supports grid based path planning algorithms also.

DP SLAM uses a single occupancy grid to represent the map. But each of the cells is a tree consisting observations made by different particles. As observations are made, each particle updates the tree along with its unique ID. Further an ancestry tree is also maintained to keep track of the ancestors of each particle on re-sampling.

In this paper a simplified scheme is proposed where a single occupancy grid is build based on the information from the best available particle. As each observation is made each of the particles makes prediction and they are weighed based on how closely they could predict the observation. The particle that has the highest weight is used to build the map. By this, the occupancy grid generated is based on the best available information and additional tree structure required by DP SLAM algorithm is avoided. Memory and computational requirements are thus drastically reduced to support real time systems especially ones with limited processing and memory capability.

4. SIMULATION

Motion Planning and Terrain Mapping algorithms were simulated in C. The current states of the Rover are denoted by X, Y coordinates and its orientation. At any instant the Rover can move to one of its adjacent eight cells, which is represented by its orientation.

Servos are commanded with appropriate values to enable the rover to move in forward direction. If the rover is not capable of lateral movement, it is rotated by giving differential commands to both wheels to attain the required orientation and then moved forward.

To localise the current position of the rover, its servo readings are accumulated over time. The rover moves approximately by 17 cms for one full rotation of its wheels as per equation 4

$$\text{Dist} = 2 * \text{pi} * r \tag{4}$$

here r is the radius of the wheel of typical LEGO Mindstorms Rover, 2.75 cms

The size of each grid was chosen to be 1600sq.cms. as the rover will be contained within a cell as it moves across. Ultrasonic sensor readings were used to sense the obstacles in its environment. The rover does not have any prior knowledge about the obstacles. The rover starts from the Start cell and moves towards the target cell using the Motion Planning algorithm. As it moves across the rover identifies the obstacles from Ultrasonic readings and generates the terrain occupancy grid map using the Terrain Mapping algorithm.

5. RESULTS

The Motion Planning and Terrain Mapping algorithms were simulated and tested with different scenarios. One typical case is elaborated below. An 8x8 grid is considered as shown in figure 2 along with its XY coordinates. The rover starts from the cell marked ‘S’ and should reach the target cell marked ‘T’. Shaded region in the figure represents obstacles.

Initially the rover is not aware of the obstacles present. As it moves towards the target, it senses the obstacles. Three particles were considered in the Modified DP SLAM algorithm. Figure 3 shows the path traced by the rover. Figures 4, 5 and 6 show the propagation of each of the particles along with sensed obstacles.

At each step the estimate of the best particle is used to build the terrain map. Figure 7 gives the comparison of the three particles. The propagation of each of the particles matches closely. At each observation, particles are weighted based on how closely they could predict some known observations. The particle that has the highest weight is used to update the global terrain map. When the weight of the particles reduces below a threshold, they are reassigned with the states of the particle with highest weight. In the test scenario considered, particle 2 was having the highest weight while predicting the rover states. Whenever the weight of the other particles were reducing below a threshold, they were re-initialized with the states of particle 2. The generated output terrain map based on the information from the best available particle, is given in figure 8.

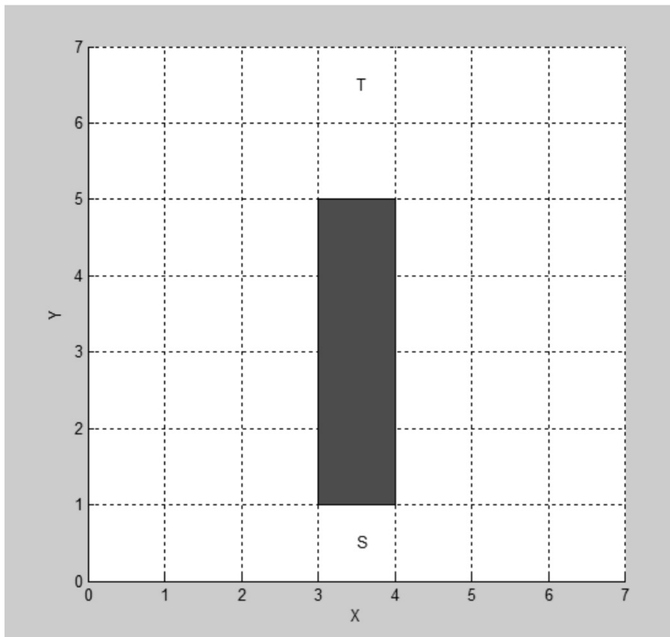

Figure 2: Grid Considered

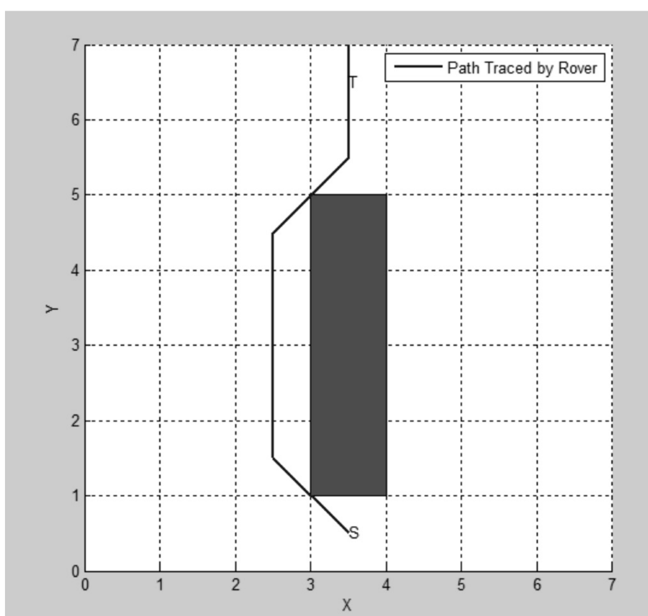

Figure 3: Path Traced by the Rover

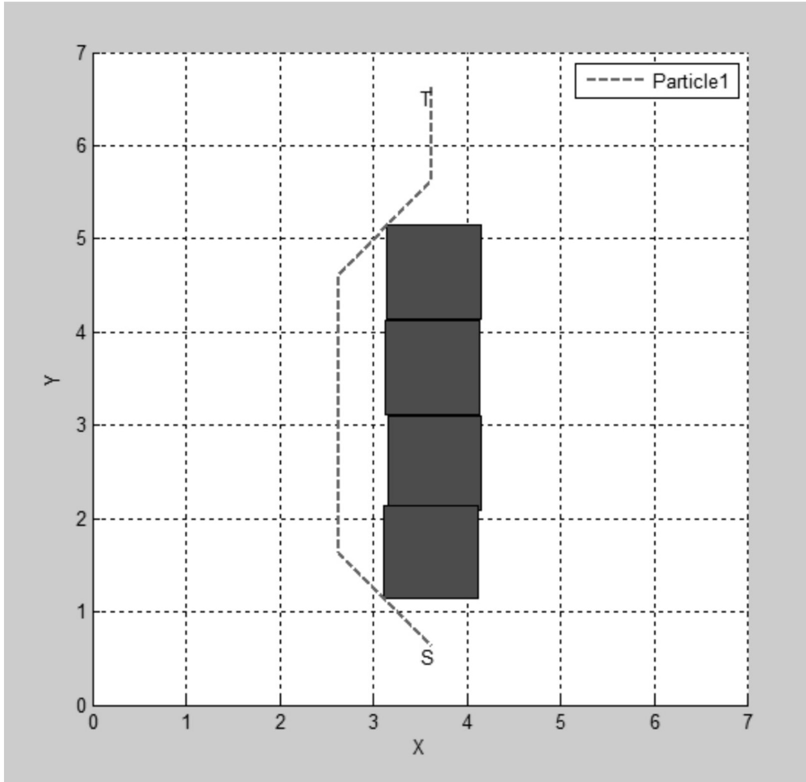

Figure 4: Propagation of Particle1

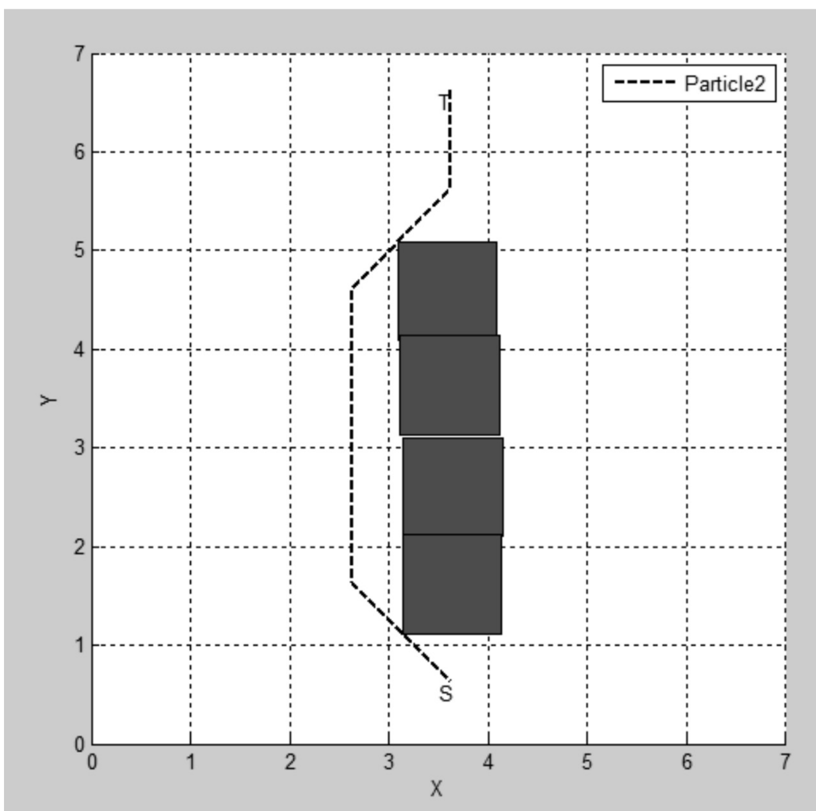

Figure 5: Propagation of Particle2

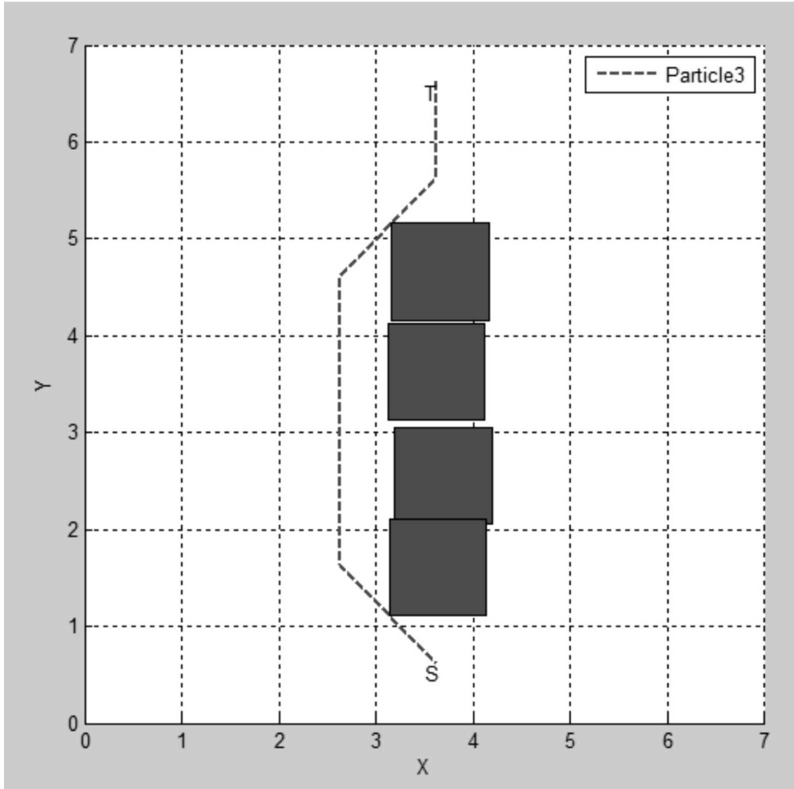

Figure 6: Propagation of Particle3

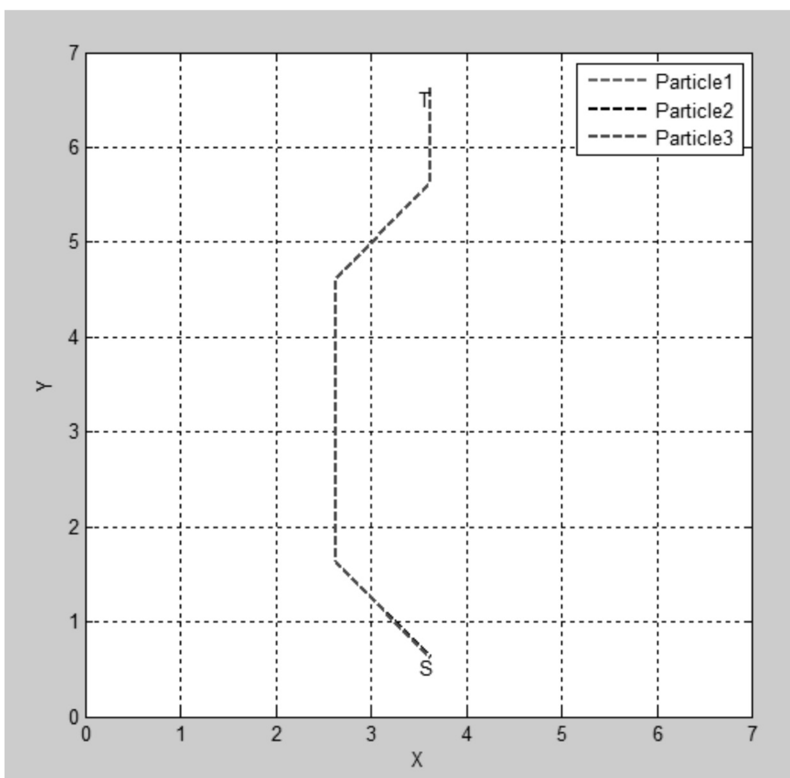

Figure 7: Comparison of Particle Propagation

While estimating the rover position white Gaussian noise was added to servo encoder values in order to account for the uncertainties such as wheel slippage. Since the processing capability and memory availability of simple autonomous rover will be limited, single occupancy grid is generated using three particles

For example LEGO Mindstorms rover has 32-bit ARM7 based processor which is supported with 256kB Flash memory and 64 KB RAM. The algorithm can be run on such platforms efficiently and used to generate the terrain map of the region explored.

The computational time taken for propagation of particle states and updating terrain map for one observation is approximately 0.166 msec. Hence the algorithm could be implemented for real time systems.

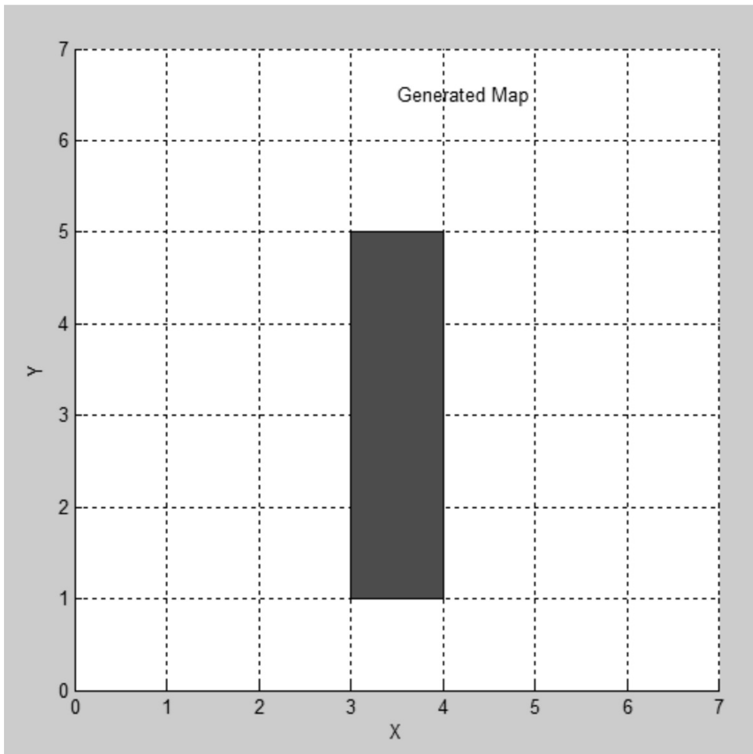

Figure 8: Generated Terrain Map

6. CONCLUSION

Autonomous Mobile Robots mandatorily need efficient techniques for Path Planning and Terrain Mapping. Based on the capabilities of the real time system simple and effective algorithms are essential. In this paper a simple distance based path planning algorithm is presented with modified DP SLAM algorithm for generating occupancy grid based terrain map of the region explored. These algorithms were simulated with various test scenarios and proved to be efficient.

References

[1] Steven M.LaValle, "Planning Algorithms", Cambridge University Press 29/5/2006.

[2] Takayuki Goto, Takeshi Kosaka, and Hiroshi Noborio, "On the Heuristics of A* or A Algorithm in ITS and Robot Path-Planning", *Proceedings of the 2003 IEEWRSJ, InU. Conference on Intelligent Robots and Systems*

[3] Anthony Stentz, "Optimal and Efficient Path Planning for partially known environment", *IEEE International Conference on Robotics and Automation,* May 1994.

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

[5] Jorge Nieto, Emanuel Slawinski, Vicente Mut and Bernardo Wagner, "Online Path Planning based on Rapidly-Exploring Random Trees", *Industrial Technology (ICIT)*, 2010 IEEE International Conference, 14-17, March 2010.

[6] GopiKrishnan S, BVS Shravan, Harishit Gole, Pratik Barve and Ravikumar L, "Path Planning Algorithms: A comparative study", *STS-2011*, Trivandrum, VSSC.

[7] Gurkan Tuna, Kayhan Gulez, V. Cagri Gungor, T. Veli Mumcu, "Evaluations of Different Simultaneous Localization And Mapping (SLAM) Algorithms", *IECON 2012 - 38th Annual Conference On IEEE Industrial Electronics Society*, 25-28 Oct. 2012

[8] Austin I. Eliazar and Ronald Parr, "DP-SLAM 2.0".

[9] Ozan Özışık, Sırma Yavuz, "An Occupancy Grid Based SLAM Method", *CIMSA 2008 – IEEE International Conference on Computational Intelligence for Measurement Systems And Applications Istanbul* – Turkey, 14-16 July 2008

[10] Jason Miller Henry de Plinval Kaijen Hsiao, "Mapping Contoured Terrain: A Comparison of SLAM Algorithms for Radio-Controlled Helicopters", *Cognitive Robotics*, Spring 2005