Mapping-Based Navigation

James Acker, Ishmael Roger, Jherrod Thomas

Dept. of Robotics and Mechatronics Engineering, Kennesaw State University
Kennesaw, GA 30144

Email: [jacker1]@students.kennesaw.edu
Email: [irogers1]@students.kennesaw.edu
Email: [jthom896]@students.kennesaw.edu

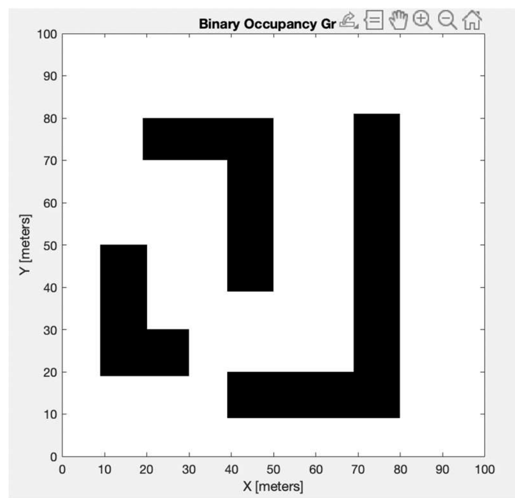
1. Project Objective

Path planning is an essential component of autonomous navigation for robots or vehicles. It allows the robot to find the shortest and most obstacle-free path from a start to goal state. Path planning needs a map of the environment, often represented in a several different ways e.g., grid-maps, state spaces, & topological roadmaps. Additionally, start and goal states are essential input for path planning. When combined with perception (or vision) and control systems, path planning is one of the three main building blocks for autonomous navigation.

2. Occupancy Grid Map

The Occupancy grid is a method to assist in mobile robot path planning An occupancy grid is a virtual map of a physical environment where values represent the presence of obstacles or cost for a particular path within the grid.. Figure 1 represents a 2D Occupancy Grid created using Peter Corke's Robotic Toolbox. The graphical grid is also transformed into a data matrix where the coordinates of the black obstacles have a value of 1 and the coordinates of the white area have a value of 0.

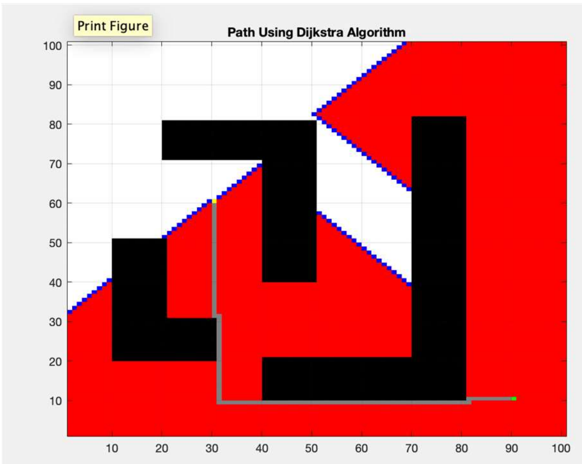
Figure 1 GRID MAP FROM PETERCORKE ROBOTICS TOOLBOX



3. Dijkstra Algorithm

Dijsktra's formula seeks to find the shortest distance or path between a starting node and goal node. The starting node is updated with distance of zero whereas all other nodes in the grid are updated with distance of infinity. Each node is visited in the grid, and its distance from the previous node is stored. Once all the nodes are visited, it works backwards using the shortest distance between each neighboring node in order to find the shortest path. A disadvantage to Dijkstra's algorithm is that the algorithm searches for the goal in all directions which can overcomplicate simple navigation maps or can be computationally heavy for complex maps. Figure 2 is example of Dijkstra’s Algorithm.

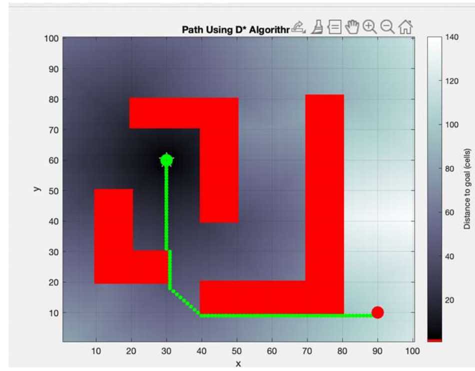
Figure 2 DIJKSTRA'S ALGORITHM



4. D*Algorithm

The D* Algorithm is an extension of the A* algorithm which introduces a cost map to the occupancy grid. This algorithm seeks to improve both Dijkstra and A* algorithms by incorporating cost analysis and remapping. The cost map determines the cost of how a cell could be traversed (e.g. in a horizontal or vertical direction). There is also cost applied for traversing cells diagonally and cost of obstacles. The D* algorithm also allows for replanning should the map costs change. A disadvantage to the D* algorithm is that it is computationally heavy for the initial planning and replanning phases. Figure 3 is an example of the D* Algorithm.

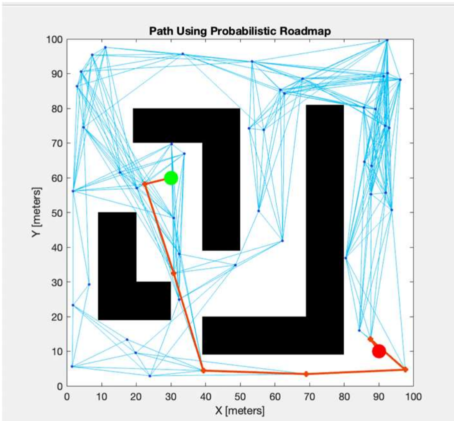
Figure 3 D ALGORITHM*



5. Probabilistic Roadmap Planning

The probabilistic roadmap planner (PRM) seeks to overcome some of the disadvantages of the D* algorithm. PRM has two stages: a planning stage and a query stage. The introduction of the planning stage allows for changing the starting point as well as changing the goal points. During planning, a graphical roadmap is constructed for navigating the environment. The first step includes a random selection followed by a nearest neighbor analysis to determine probable paths. During the query phase, the start and goal points are introduced into the path analysis from planning. A disadvantage of the PRM approach is that random sampling can result in different paths each time the planning phase is run and furthermore the randomness can lead to not always having the shortest path selected. Figure 4 is example of the PRM algorithm.

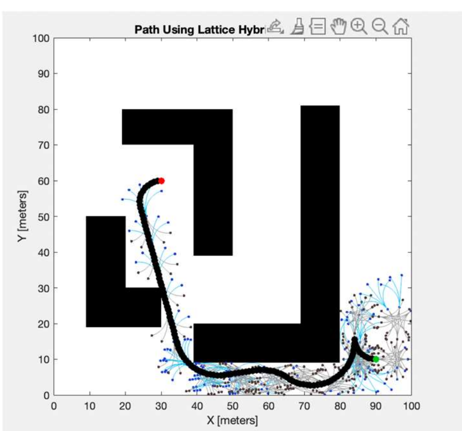
Figure 4 PROBABALISTIC ROADMAP PLANNING



6. Lattice Planning

Lattice Planning incorporates robot motion int the path planning process. The motion can happen with three points, and each path forms an arc. At the end of each branch, the algorithm adds the same set of three motions, each motion being rotated. A disadvantage of Lattice planning is a robot driving along a diagonal path as well as any turns requiring sharper steering (e.g. hard right angle) is not optimal. Figure 5 represents a lattice planning algorithm called Hybrid A*.

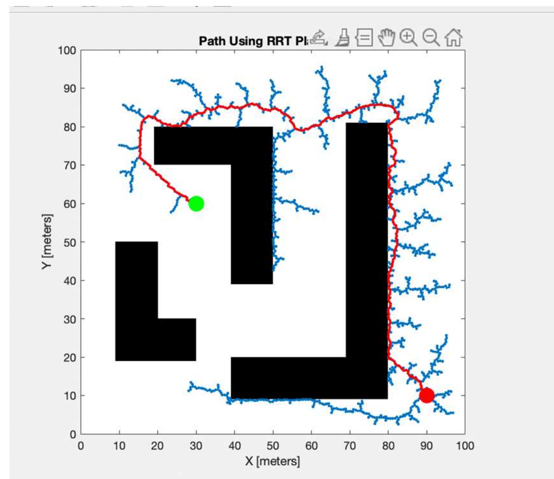
Figure 5 LATTICE PLANNING



7. RRT Algorithm

The Rapidly Exploring Random Tree algorithm (RRT) is similar to the Lattice Algorithm since it also incorporates robot motion. However, the RRT uses probabilistic methods like the PRM planner. This allows the algorithm to reduce some of the randomness pathing. A disadvantage of RRT is that it requires the correct selection of nodes and simulation time in order to get an optimal path between the start and goal points. Figure 6 is an example of an RRT algorithm.

Figure 6 RAPIDLY EXPLORING RANDOM TREE (RRT)



8. Summary

Indeed, path planning is an incredibly useful tool for managing complex tasks. It helps us to define clear goals, identify potential obstacles, and develop effective strategies to achieve success. By breaking down a task into smaller, more manageable steps, path planning can make even the most daunting task feel more achievable. Path planning is an invaluable tool that can help us reach our goals in a more efficient and organized manner.

9. References

[1] https://mathworks.com/help/