Prasanna Marudhu Balasubramanian **Project Proposal** ENPM661

Chayan Kumar Patodi

Optimal Path Planning for Autonomous Mobile Delivery Robots

Introduction:

The delivery of packages, parcel and products has seen amazing growth all around the globe, because of the exponential growth of e-commerce, but with pros, comes it cons and the cons brings a wide array of challenges. Courier Express and Postal organizations face tough decisions daily, such as how can they continue to meet the world's changing needs for e-commerce, and it is crucial for them to adapt and evolve to meet the growing demand from customers to reduce prices and improve services.

Autonomous mobile robots (AMRs) use sensors and navigation technologies to move around their environment without human intervention. They can potentially help the Postal Service cut costs, increase efficiency, and enable new services. According to an online survey the delivery robots' market is expected to grow from USD 12 billion to USD 34 billion by 2024.

In robotics Path planning is well known and it plays an important role in the navigation of these autonomous mobile robots (artificial intelligent mobile robots). Currently, the path planning problem is one of the most researched topics in the field of autonomous robotics and that is why finding a safe path in a cluttered environment for a mobile robot is an important requirement for the success of any such mobile robot project. These tasks are challenging since robots must avoid static and dynamic obstacles, and mainly robots must perform path planning avoiding obstructions in its way and minimizing a cost such as time, energy, and distance. Most mobile robots today operate at such low speeds that dynamics are rarely considered during path planning, further simplifying the mobile robot instantiation of the problem. In most practical situations, the mobile robot cannot take the simplest path from the start to the goal point. So, path planning techniques must be used in this situation.

The three general problems of navigation are localization, path planning and motion control. Between these three problems, it can be argued that path planning is one of the most important issues in the navigation process. Path planning enables the selection and identification of a suitable path for the robot to traverse in the workspace area.

The Autonomous robots like Segway Loomo Delivery, Anybotics and Continental, Postmates Serve, NURO, KiwiBot, Starship Robots and the SameDay (FedEx) Bot are aiming to help the companies like AutoZone, Lowe's, Pizza Hut, Target, Walgreens, and Walmart to assess their need for quicker and hazzle free robotic delivery of products. The determination of a collision free path for a robot between initial and end positions through hindrances present in a workspace is important to the design of an autonomous robot path planning. Our project aims in analyzing different types of algorithm, comparing them and then generate an optimal path for the robot to navigate in an environment.

NURO BOT

Pic credit: https://www.wired.com/story/nuro-grocery-delivery-robot/

There are different algorithms that may complete the same task in less or more time, space, effort etc. Each type of algorithm has its own strength and weakness which is compared and discussed to obtain the optimal path planning algorithm for the mobile delivery robots.

For this project we will assume that Robot will have the radar, Lidar and other sensors with which the robot can sense the obstacles (including trees, buildings and other miscellaneous hindrances) in its environment and it is also assumed that the location of the customer (co-ordinates) or delivery location are known and fed to the algorithm for detecting the optimal path.

Method:

As per the assumptions, we know the start and the goal point for the robot to traverse between them. Various algorithms and methods have been developed over the decade to create real time path planning system for autonomous robots. These are the following things or activities that must be followed or carried out by an autonomous robotic system to enable the execution of the task of robot navigation: mapping and modeling the environment, path planning and driving systems. The selection of an appropriate algorithm in every stage of the path planning process is very important to ensure that the navigation process will run smoothly.

The algorithms that will be analyzed for obtaining the optimal path for the robot includes RRT, Visibility Graph, A-Star, RRT-Star, Best First Search (BFS) and Dijkstra algorithms and in each algorithm the start and goal point are known to obtain best route for the robot to traverse.

Concept of Mapping and Localization *that we will adapt in our project:*

- In Planning, a graph is denoted by G = (V, E, c), where V is the set of vertices, E is the set of edges and c: E → R is a function that assigns a cost to each edge in E. In a directed graph, each edge is an ordered pair of vertices (vi, vj) and is assigned a direction from vi to vj. A partitioned graph, G, is a graph with a partition of its vertex set into ` mutually exclusive sets (V1, . . ., V`) where ∪ ` i=1Vi = V. A route over a graph is a sequence of vertices P = (v1, . . ., vk) linked by edges (vi , vi+1), i = 1, . . . , k − 1. A walk is a route such that no edge is traversed more than once. A path is a route where vi 6= vj for all i, j ∈ {1, . . . , k − 1}. A closed route is a route of any type (e.g. route, walk, path) where v1 = vk. A tour is closed path that visits all vertices in V exactly once. Given a complete graph G = (V, E, c), the Travelling Salesman Problem (TSP) computes a minimum cost tour of G.

- Assume that path planning is considered in a square terrain and a path between two locations is approximated with a sequence of adjacent cells in the grid corresponding to the terrain. The length A (α,) from cell "α " to its adjacent cell "β " is defined by the Euclid distance from the center cell "α" of one cell to the center cell "β" of another cell. Each cell in this grid is assigned of three states: free, occupied, or unknown otherwise. A cell is free if it is known to contain no obstacles, occupied if it is known to contain one or more obstacles. All other cells are marked unknown. In the grid, any cell that can be seen by these three states and ensure the visibility constraint in space navigation. We denote that the configuration grid is a representation of the configuration space. In the configuration grid starting from any location to attend another one, cells are thus belonging to reachable or unreachable path.

- Suppose that a robot R at time (i) has a map M(i) and an initial belief state b(i). The robot's goal is to reach a position (p) while satisfying certain some temporal constraints. Thus, the robot must be at location (p) at or before timestep (n). Although the goal of the robot is distinctly physical, the robot can only really sense its belief state, not its physical location, and therefore we map the goal of reaching location (p) to reaching a belief state (b(g)), corresponding to the belief that location of R = p. With this formulation, a plan q is nothing more than one or more trajectory. In other words, plan q will cause the robot's belief state to transition from b(i) to b(g), if the plan is executed from a world state consistent with both b and M.

Algorithms under consideration:

- Visibility graph, as the name suggests is a graph of inter visible locations, typically for a set of points and obstacles in the Euclidean plane. Each node shows a point location, and every edge is a visible connection between the nodes. To make it clearer, imagine if the line segment connecting two locations does not pass through any hurdle, an edge is drawn between them in the graph.

- The concept of Dijkstra's algorithm is very simple. We begin with the initial vertex where the path should start, and it marks all adjoining neighbors of the initial vertex and the cost to reach there. It then moves forward from the vertex with the *lowest* cost to all its adjacent vertices and marks them with the cost to get to them via itself.

- In A-Star (A*) algorithm, we give priority to nodes that have a lower estimated distance to the goal than others. In this scenario , we mark each and every node with the estimated cost by calculating the Euclidean distance or the Manhattan distance between the current and the final vertex. A-Star might be much faster than Dijkstra's in certain environments.

- A* is computationally expensive when the search space is huge, e.g. due to a fine-grain resolution required for the task, or the dimensions of the search problem are high & thus we move towards Rapidly-Exploring Random Trees (RRT). RRT uses a randomized approach that aims at quickly exploring a large area of the search space with iterative refinement. It selects a random point in the environment and connects it to the initial vertex. Subsequent random points are then connected to the closest vertex in the following graph. RRT can be used for path-planning by maintaining the cost-to-start on each added point and biasing the selection of points to occasionally falling close to the goal.

- Once the environment has been converted into a graph, we can employ other algorithms. For example, floor coverage can be achieved by performing a depth-first search (DFS) or a breadth-first-search (BFS) algorithm on a graph where each vertex has the size of the coverage tool of the robot. Doing a DFS or a BFS algorithm are far from optimal as many vertices might be visited twice.

In summary, a generic path planning algorithm needs to meet several requirements as listed below:

1) “The resulting paths should have the lowest possible cost to prevent any indirection”

2) “It should be fast and correct to not thwart the simulation process, for example it should be robust if there no collisions occurred during the navigation process”

3) “The algorithm should be generic with respect to different maps where its means that it should not be fully optimized for a specific map type”

Assumptions:

Following are the assumptions we shall make while working on our project:

1) The Environment in which the robot moves is defined already
2) The robot is provided with LIDAR and other sensors to detect the obstacles and inaccessible paths.
3) The robot has enough battery power for traversing.
4) The robot is versatile and robust to handle rough terrains.
5) Robot have enough load carrying capacity

6) Remote controllability should be enabled in the robot in order to guide remotely during difficult situation.

7) The robot is designed according to the application desired (it has an arm to open the gates or has a 5^{th} wheel to climb up the stairs.)

8) The robot is equipped with required sensors and devices for the application (GPS and Perception Cameras.)

9) The robot can handle all the uncertainty.

10) When the algorithm is not able to determine the optimal path, it should alert for remote control of the robot.

Goal:

Our aim is to find out the best and optimal path planning algorithm, that can be implemented for the delivery robots. When the robot moves along the path, the shortest path from the current robot coordinates to the goal coordinates is calculated initially and when the obstacle is detected this helps the algorithm to re-plan the route faster than planning from scratch since it will modify only the previous search results locally. Approximately time-optimal trajectory where a start to a goal configuration is computed, such that the robot does not collide with any moving obstacle.

The aim of the developed strategy would be to solve the problem when the robot is located between two obstacles such as the following: to look for the node close enough to reach the target without collision and how to avoid obstacles and move between two obstacles in the shortest path.

An overview of the path planning algorithms for autonomous robot, their strengths and their weakness will be analyzed and presented as the outcome for the autonomous delivery robot application.

Simulation results will be used to compare the performance of the presented algorithms

Software:

MATLAB will be used for developing and simulating the search algorithms for this project.

References:

1. R.A Brooks and T.Lorenzo-Perez. "A subdivision algorithm in configuration space for find path with rotation," in *IEEE Transactions on Systems, Man and Cybernetics*, Mar./Apr.1985, SMC-15(2), pp 225-233.

2. Kuo-Chin Fan and Po-Chang Lui, "Solving the find-path problem in mapped environments using modified A* search algorithm," in *IEEE Transactions on Systems, Man, and Cybernetics,* vol.24, no.9, September 1994, pp 1390-1396.

3. Charles W. Warren, “Fast path planning method using modified A*method,” in. *IEEE International Conference on Robotics and Automation, pp* 662-667.

4. O. Khatib, “Real time obstacle avoidance for manipulators and mobile robots,” in *International Journal of Robotics Research*, 5(10):90-98, Spring 1986.

5. L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, “Probabilistic roadmaps for path planning in high dimensional configuration spaces,” *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, 1996.

6. T. Lozano-Perez, “Spatial planning: A configuration space approach,” in *IEEE Transactions on Computers*. 1983, vol. C-32, pp. 108–120.

7. O. Takahashi and J. Schilling, “Motion planning in a plane using generalized Voronoi diagrams,” *IEEE Transactions on Robotics and Automation*, vol. 5, no. 2, pp. 143–150, 1989.

8. Bruce R. Donald, “A search algorithm for motion planning with six degrees of freedom,” *Artificial Intelligence*, vol. 31, no. 3, pp. 295–353, 1987.

9. Christoph Niederbergerejan Radovie, and Markus Gross, “Generic path planning for real time application,” in *Proceedings of the Computer Graphics International (CGI 04),* 2004, pp 1-8.

10. Pearsall, J., ed. *Concise Oxford Dictionary*. Tenth Edition, Revised ed. 2001, Oxford University Press.

11. O.Hachour , Path planning of Autonomous Mobile robot , *International journal of systems applications, engineering & development issue 4, volume 2, 2008.*

12. Neil Mathew, Stephen L. Smith, and Steven L. Waslander , *Optimal Path Planning in Cooperative Heterogeneous Multi-robot Delivery Systems* , University of Waterloo, Waterloo, Ontario, N2L3G1, Canada.

13. *Introduction to Autonomous Mobile Robots, Chapter 6,* Book by Illah Reza Nourbakhsh and Roland Siegwart.