Project-5 Proposal

ENPM661: Robot Planning

Submitted by

Vidya Shankar Manju **Mudit Nawal Singal**

UID: 119376706 **UID: 119262689**

Table of Contents

1. Title:

RRT-A* Motion planning algorithm for non-holonomic mobile robot

2. Introduction:

2.1 Definitions

The RRT-A* motion planning algorithm for non-holonomic mobile robots is a technique that combines Rapidly-exploring Random Trees (RRT) and A* search algorithm, to generate a feasible and optimal path for a robot to navigate from its current location to the goal location.

Mobile robots are used in various applications, such as industrial automation, service robots, and search and rescue operations. Non-holonomic mobile robots have constraints on their motion, such as limited steering angles and velocities. These constraints make the planning of motion for these robots a challenging task, especially in complex environments. Rapidly-exploring Random Trees (RRT) is a popular algorithm used for motion planning of non-holonomic robots. It generates a tree structure of possible paths from the starting point to the goal point.

RRT works by sampling random points in the configuration space of the robot and connecting them to the nearest node in the tree. This process is repeated until the goal point is reached or a maximum number of nodes are generated. However, RRT does not take into account the optimality of the generated path. A* search algorithm is a popular search technique that finds the shortest path between two points in a graph. A* takes into account the cost of the path and the heuristic estimate of the remaining cost to the goal.

2.2 Background

The rise of mobile robots with their vast use cases such as house helper, industrial automation, and search and rescue operations, called for advanced planning algorithms to find a feasible path. Mobile robots with non-holonomic constraints (limited steering) have constraints on their motion, making it harder for planning algorithms to find a feasible path.

The Rapidly exploring Random Trees (RRT) algorithm made it possible to efficiently find a feasible solution using random samples from the map. Further, it is also possible to use it for generating paths for non-holonomic robots. It generates a tree structure of possible paths from the starting point to the goal point. To overcome the issue of sub-optimal path returned by RRT, a hybrid algorithm RRT-A* is developed that considers the heuristic distance of generated nodes from the goal node. This idea of combining RRT with A* leads to optimal path generation using RRT.

2.3 Literature review

The basis for the implementation of project 5 will be [1], which shows how A* is incorporated into the RRT algorithm, how non-holonomic constraints are modelled, and also the results that verify that the path generated by the RRT-A* algorithm are optimal when compared to the RRT algorithm.

When compared to [2], the path generated by goal-biased RRT is sub-optimal as more sampling is done towards the goal position. This leads to less emphasis on finding optimal path from the start node to the goal node. Ultimately, a solution is found, but the computational resources used are more and the quality of path generated is poorer than the RRT-A* algorithm.

Furthermore, techniques such as bi-directional RRT may not converge for the cases of non-holonomic robots. Thus, the RRT-A* is a good algorithm to use for efficient and optimal path generation from a start node to a goal node for non-holonomic robots (such as car-like robot).

3. Goal:

3.1 Option 1, 2 or 3

We've selected option 1: Practical implementation. Implementing RRT A* based on the above research paper.

3.2 Simulation or real-world implementation

We are going to simulate the node exploration and path planning solution on the matplot using python. Also, we will explore the MATLAB libraries for simulation. We will try for real-world implementation if time permits.

3.3 Outcome of the project

Efficient path planning, optimized path, collision free path. Learning the specifics and implementation details of RRT A* algorithm.

4. Method:

4.1 Path planning Method

The method begins by initializing the tree from starting node (random or start node). At each step the RRT-A* algorithm assigns a cost to a newly sampled node and checks if it is feasible to reach the node without colliding with obstacles. By iterating through different nodes, the algorithm attempts to connect the path from start node to the goal node. Additionally, we also have a time limit on the search algorithm.

When compared to RRT algorithm, the RRT-A* algorithm incorporates the A* into RRT node generation and search. This gives the new algorithm the ability to search for optimal paths instead of just feasible paths.

To ensure that the generated path is collision-free, the method employs a local planner that checks for obstacles along the path and generates a new path if an obstacle is detected. The local planner uses a Dubbin's path planning algorithm to generate feasible paths that the robot can follow.

4.2 Title of the paper/papers that you will implement or use (provide the pdf files)

RRT-A* Motion planning algorithm for non-holonomic mobile robot.

RRT-A_Motion_planning_algorithm_for_non-holonomic_mobile_robot.pdf

4.3 List of software and package that you are planning to use:

ROS, OMPL, Gazebo, MATLAB, Python

4.4 Hardware requirements

Four-wheel mobile robot with Ackerman steering system. A motor controller to command the motors. A microcontroller (Arduino) to command the motor controller. No sensors or encoders required for open loop execution. For closed loop control, wheel encoders track the position of the robot accurately and implement accurate velocity control. It will help in eliminating noise and un-modelled disturbances in the system.

5. Timetable:

Milestones and Timelines		
Task	**Days**	**Total Days**
Study various RRT variations and research papers	4/2/2023 - 4/7/2023	5
Selection and detailed review of final selected paper	4/7/2023 - 4/11/2023	4
Analysis of software and tools required for implementation	4/11/2023 - 4/15/2023	4
Flow chart and pseudo code preparation.	4/15/2023 - 4/18/2023	3
Implementation of RRT variation for node generation and path exploration	4/18/2023 - 4/25/2023	7

Simulation of generated path.	4/25/2023 - 4/28/2023	3
Testing with different use cases	4/28/2023 - 4/30/2023	2
Hardware implementation (if time permits)	4/30/2023 - 5/2/2023	3
Presentation and report making	5/2/2023 - 5/5/2023	3

6. References:

[1] J. Li, S. Liu, B. Zhang and X. Zhao, "**RRT-A* Motion planning algorithm for non-holonomic mobile robot**," 2014 Proceedings of the SICE Annual Conference (SICE), Sapporo, Japan, 2014, pp. 1833-1838, doi: 10.1109/SICE.2014.6935304.

[2] X. Jin, Z. Yan, H. Yang, Q. Wang and G. Yin, "**A Goal-Biased RRT Path Planning Approach for Autonomous Ground Vehicle**," 2020 4th CAA International Conference on Vehicular Control and Intelligence (CVCI), Hangzhou, China, 2020, pp. 743-746, doi: 10.1109/CVCI51460.2020.9338597.