Implementation of Informed RRT* planning algorithm for optimal trajectory

1st Pritom Gomes
Maryland Applied Graduate Engineering
University of Maryland
College Park, USA
pgomes10@umd.edu

2nd Joseph Thomas
Maryland Applied Graduate Engineering
University of Maryland
College Park, USA
joseph10@umd.edu

3rd Amogh Wyawahare
Maryland Applied Graduate Engineering
University of Maryland
College Park, USA
wamogh@

***Abstract*— This paper deals with the task of running a software simulation in ROS using gazebo on a customized map. The algorithm in use was Informed Optimal Rapidly Exploring Random Tree (Informed RRT*). This planning algorithm provides better convergence and cost-effectiveness in terms of reaching the goal state defined by the user from its initial position. The Informed-RRT* algorithm utilizes a hyper-ellipsoid that is created from an initial feasible solution to guide the sampling strategy towards promising areas of the state space, resulting in faster convergence to an optimal solution. It has proven more effective than the former root variant RRT* planning algorithm.**

Keywords —algorithm, convergence, Rapidly exploring Random Tree, ROS.

I. INTRODUCTION

A. *Background Information*

Mobile robots have become prevalent across many industries, significantly improving industrial production efficiency and convenience in people's daily lives. Motion planning, which involves determining a sequence of actions required to complete a specific task, is a critical area of study in robotics. As mobile robotics research continues to evolve, motion planning is increasingly utilized in diversified fields, including industrial robotics, search and rescue robotics, and autonomous vehicles.

The RRT* is a motion planning algorithm that is widely used due to its asymptotic optimality, but it has a slow convergence speed. To address this limitation, the authors propose an improved version of RRT* called Informed-RRT*. Informed-RRT* uses a hyper-ellipsoid built from an initial feasible solution to guide sampling, which speeds up convergence to an optimal solution. The effectiveness of the improved algorithm is validated through numerical simulations, demonstrating that it efficiently finds the initial solution and achieves the same solution in less time compared to RRT*. Based on our simulation results of Informed RRT* based on both software and hardware simulation we can conclude whether we are getting the effective cost of path for the Informed RRT* planner.

B. *Dual steps of RRT**

The RRT* algorithm comprises two primary steps: Choose Parent and Rewire, differentiating it from RRT. During the Choose Parent process, RRT* searches for nodes within a specific radius hypersphere centred on the newly extended node X^{new}. The algorithm then selects the node in the hypersphere with the lowest path cost from the initial node to x new via the chosen node, as depicted in Figure 1, and assigns it as the parent node of X^{new} In the Rewire procedure; the algorithm scans the hypersphere to identify any nodes whose cost would decrease if the path were to pass through X^{new}. If such a node exists, the algorithm modifies the tree structure, replacing the node's parent with X^{new}.

II. METHODOLOGY

A. *Definition of Problem*

Let us assume $X \subseteq R^d$ be a *d*-dimensional configuration space, where the $d \in N$ and $d \geq 2$. Our obstacle map will be defined as a collection of $Xobs \subset X$: the obstacle space, and $X\ free = cl(X \setminus Xobs\ )$: the obstacle-free space, where $cl(\cdot)$ refers to the closure of the set.

Let $xinit \in X\,free$ be the initial node and $X\,goal \subset X\,free$ be the goal region, which means that the motion planning is completed once reaching this region. The continuous function σ : [0,1] α X is called a path.

A motion planning problem is to find a collision-free path σ:[0,1] α $Xfree$, whereσ(0) = $xinit$, σ(1) ∈ $Xgoal$. The sampling area then conforms to the hyper-ellipsoid, facilitating more targeted sampling in promising regions of the state space.

Informed-RRT* utilises a method of direct sampling. The algorithm functions like RRT* until the initial solution is identified. Upon locating the initial solution, Informed-RRT* is defined as

$$||x - xinit\,|| + ||xgoal - x|| \leq c\ best$$

B. *Algorithm of Informed RRT**

This algorithm was executed in Python using all known libraries apart from the navigation stack or any SLAM-oriented package. Some libraries used were rospy, matplotlib, cv2, NumPy etc. Initially,

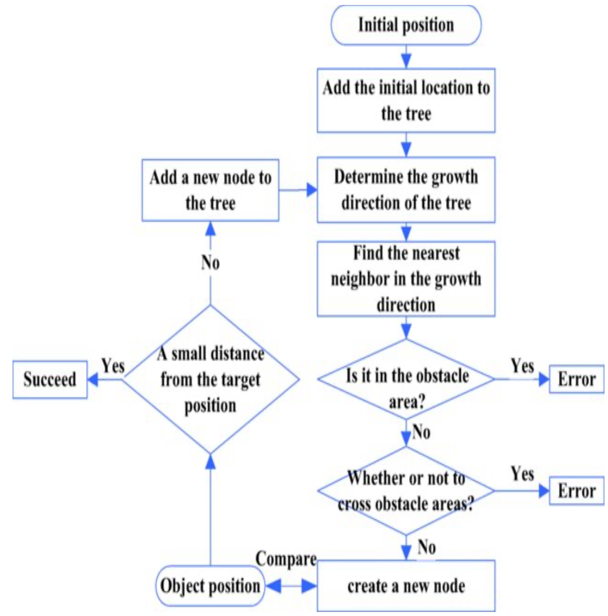

Fig 1: Informed RRT* algorithm as a flowchart

We initiated our code by giving the starting and ending coordinates in our obstacle map, followed by adjacent nodes and then the checksum entities of the obstacle as the map proceeded through the cost-effective trajectory. We also set the angular velocities of the turtle bot and select the nodes at every few centimetres to follow the linear course from node to node and approach the goal node. The planner helps both to form a hypersphere and aids it in choosing the minor cost node.

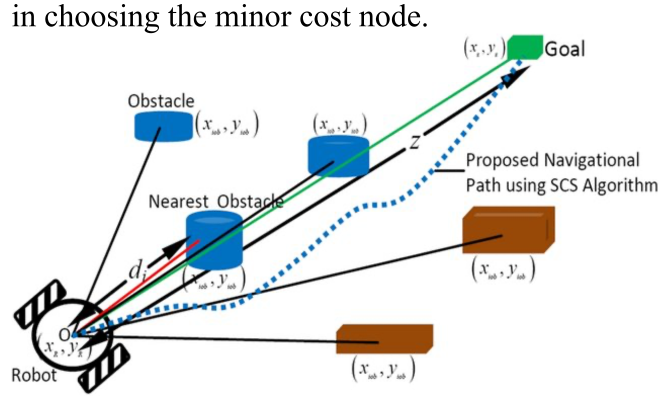

Fig 2: Turtlebot planning analysis

III. SIMULATION AND RESULTS

A. *Map used for simulation*

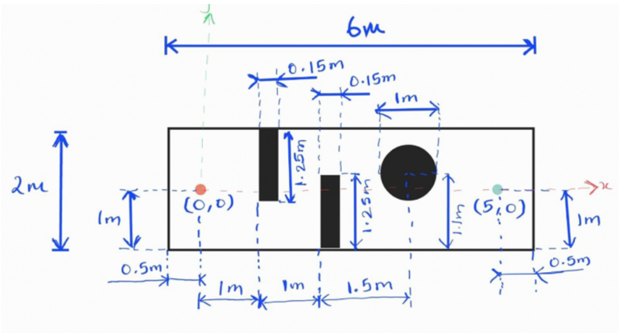

Fig 3: Obstacle Map

The map we used consists of two identical rectangles of dimensions and one circle of diameter 1m. The black shapes act as the robot's obstacle, and the white part is the free space.

B. *2D simulation*

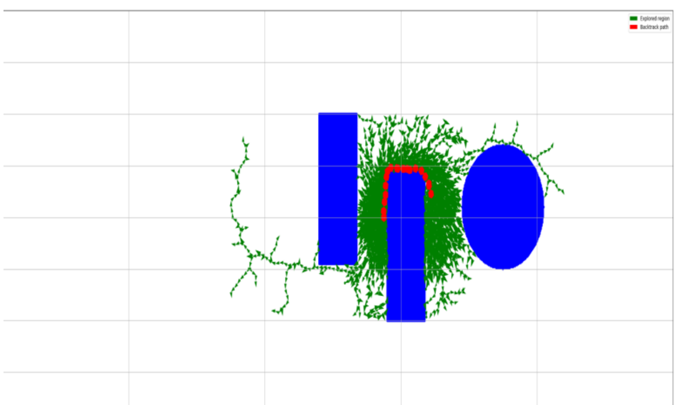

Fig 4: Informed RRT* simulation

The simulation enables the turtle bot to draw an ellipsoid around the randomly occurring nodes to reach the goal node in optimal duration. Hence, the bot successfully implants its path trajectory as per our code.

C. *3D simulation in gazebo using turtle bot*

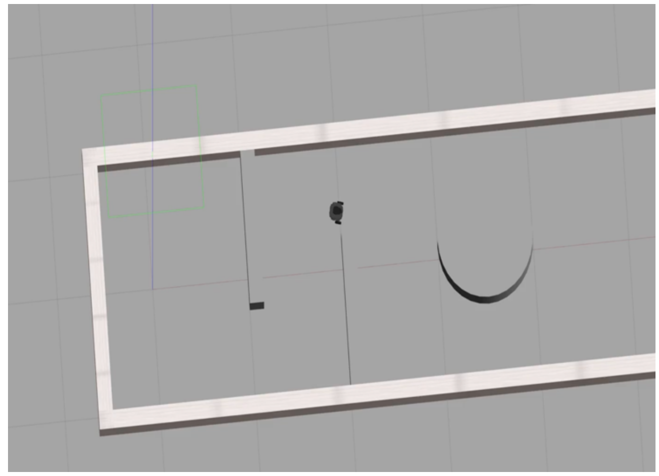

Fig 5: Obstacle map with turtlebot in gazebo world

The turtle bot follows our custom-made obstacle map in the gazebo, just like our 2D simulation. It does not use any form of LIDAR as the obstacle size has negligible height. We set the angular frequency error in our code as. $\pm 2°$

Simulation of video in the gazebo

https://drive.google.com/file/d/138pQsJUD4JEr1oZdm54xGpx_YOx1EnBO/view?usp=share_link

IV. CONCLUSION

After completing the software simulation, we can determine that the informed RRT* provides an optimal trajectory to the turtle bot from the start to the goal node. However, the hardware simulation did not go as expected as the software simulation. We had many issues with the publishing values of the actual turtle bot. The bot revolves and follows the wrong path against our code. Therefore, we have asked our instructor for an extension and will give the video link of the hardware simulation after the report.

References

.

[1] Y. Zhao, Y. Liu, H. Gao, and S. Yan, "Research on Informed-RRT* with Improved Initial Solution," 2022 4th International Conference on Intelligent Control, Measurement and Signal Processing (ICMSP), Hangzhou, China, 2022, pp.977981,doi:10.1109/ICMSP55950.2022.9859229.

[2] J. D. Gammell, S. S. Srinivasa and T. D. Barfoot, "Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic," *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Chicago, IL, USA, 2014, pp. 2997-3004, doi: 10.1109/IROS.2014.6942976.

[3] Mohanty, Prases. (2020). An intelligent navigational strategy for mobile robots in uncertain environments using an innovative cuckoo search algorithm. Journal of Ambient Intelligence and Humanized Computing. 11. 1-16. 10.1007/s12652-020-02535-5.