A Literature Review: Prioritized Trajectory Planning and Optimization for a system of Non-holonomic Mobile Robots

Abhijay Singh

UID: 118592619

Tharun Puthanveetil

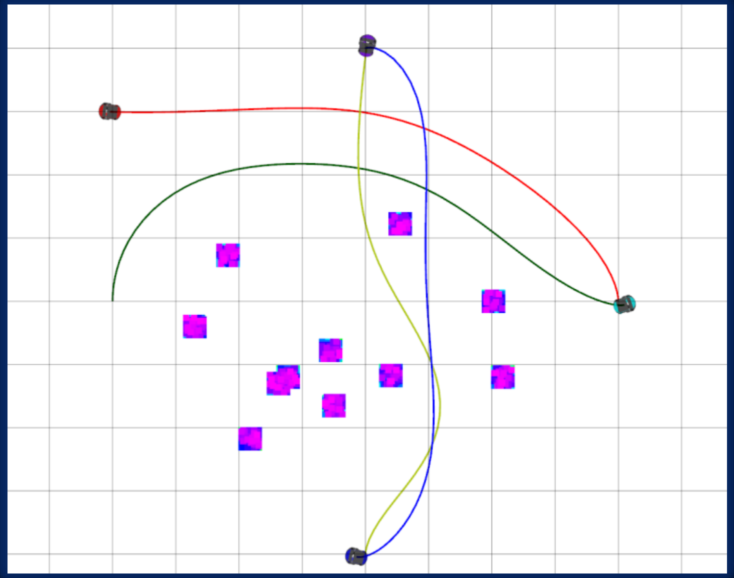
UID: 119069516

Maryland Applied Graduate Engineering, University of Maryland, College Park, Maryland, USA

(Dated: November 22, 2022)

Problem Statement

To develop an efficient collision-free multi-robot trajectory planning strategy for a large group of non-holonomic mobile robots. A priority-based trajectory optimization technique is proposed to obtain computationally efficient trajectories for multi-robot systems.



Trajectory generation for a 4-agent system

Literature Survey

- Optimal trajectory generation problems restricted to systems with small number of robots and few obstacles in the environment
- Decoupled planning methods are fast but suffer from incompleteness
- Trajectory planning problems are generally solved using a two stage pipeline (i.e) path finding and trajectory optimization where path finding stage gives the initial estimates for the optimal paths and trajectory optimization stage provides the optimal trajectories out of the initial estimates.
- Most trajectory planning solutions are designed for robots with linear dynamics.
- Distributed planning methods are computational efficient, but cannot guarantee no deadlock and are poorly suited to problems in maze-like environments.

Problem Formulation

- Multi-robot system comprising of N mobile robots whose 2D workspace is defined as $\mathcal{W} \subseteq \mathbb{R}^2$
- Obstacles in the environment are considered to be known and denoted as *O*
- Free workspace of the robot is defined by $F = W/\ O$
- Robots are assigned with circular collision model.
- Space occupied by the robot : R(x) ; x – position of the robot s.t x € $\mathbb{R}^2$
- Start position, s_i € F and end position, g_i € F where i denotes each agent

- $p = \{r^k\}_0^{N_p}$ where p is path of waypoints from 0 to $N_{P.}$

  r^k - k^{th} waypoint on path p

- Collision avoidance condition:
  - $\mathcal{R}(s_j) \cap \mathcal{R}(s_k) = \emptyset$
  - $\mathcal{R}(g_j) \cap \mathcal{R}(g_k) = \emptyset$ for all $j \neq k$
  - $\mathcal{R}(\text{pos}(r_i(t))) \cap \mathcal{R}\left(\text{pos}(r_j(t))\right) = \emptyset, \forall i \neq j$

Problem Formulation

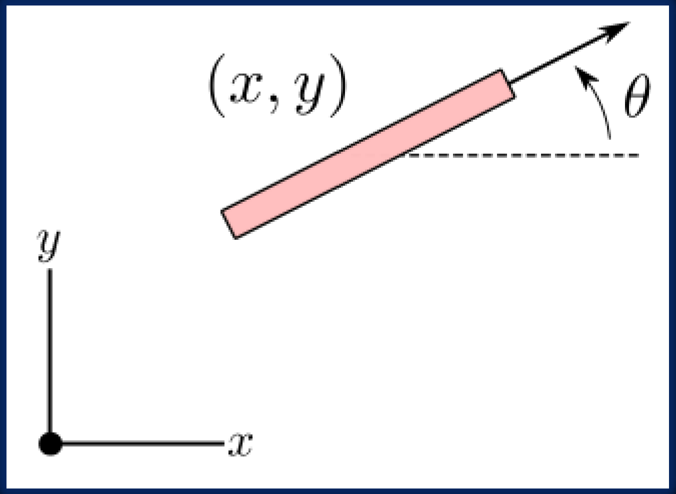
- A unicycle model is used to formulate the kinematics of the non-holonomic robots
- State of the robot given by z = $[x, y, \theta]^T$; x,y – position & θ – orientation
- Motion equations of the robot :

$$\dot{x} = v\cos(\theta)$$
$$\dot{y} = v\sin(\theta)$$
$$\dot{\theta} = \omega$$

where v – linear velocity and ω - angular velocity

- $\mathcal{U}_i = \{[v_i, \omega_i]^{\mathrm{T}} : |v_i| < v_i^{\max}, |\omega_i| < \omega_i^{\max}\}$

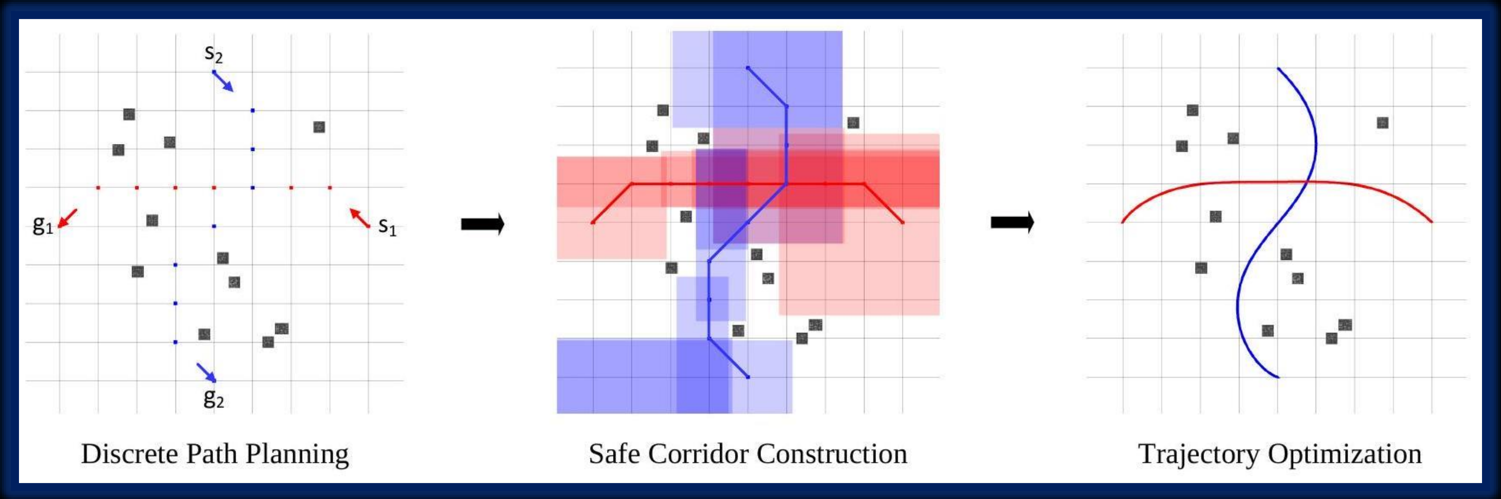
  - Max velocity constraints for each robot i



Overview

A priority-based trajectory optimization technique is proposed to obtain computationally efficient trajectories for multi-robot systems. The proposed strategy works in three stages:

- Discrete Path Planning
- Safe Corridor Construction
- Prioritized Trajectory Optimization



Discrete Path Planning

- Objective:

  Find the shortest collision free paths for the non-holonomic robots
- Workspace of the multi-robot system is represented as an undirected graph.
- Graph is created from the occupancy map of the workspace and represents both the position as well as the orientation of the robot at each vertex.
- Edges are represented as a set of predefined motion primitives – move forward, move back or turn 90degrees.

- Feasibility of motion primitives is ensured with :

$$v_i^{\max}\Delta T \geq \frac{\pi}{2} D,\ w_i^{\max}\Delta T \geq \frac{\pi}{2},\ \forall i \in \{1, \cdots, N\}.$$

Discrete Path Planning

- Vertex and edge conflicts are represented as follows:

$$\mathrm{VCon}(k) = \{\langle i, j \rangle \mid \mathcal{R}\left(\mathrm{pos}\left(r_i^k\right)\right) \cap \mathcal{R}\left(\mathrm{pos}\left(r_j^k\right)\right) \neq \emptyset\}$$

$$\mathrm{ECon}(k_1, k_2) = \{\langle i, j \rangle \mid \mathcal{R}\left(\mathrm{pos}(r_i(t))\right) \cap \mathcal{R}\left(\mathrm{pos}\left(r_j(t)\right)\right) \neq \emptyset, \forall t \in (k_1 \Delta T, k_2 \Delta T)\}$$

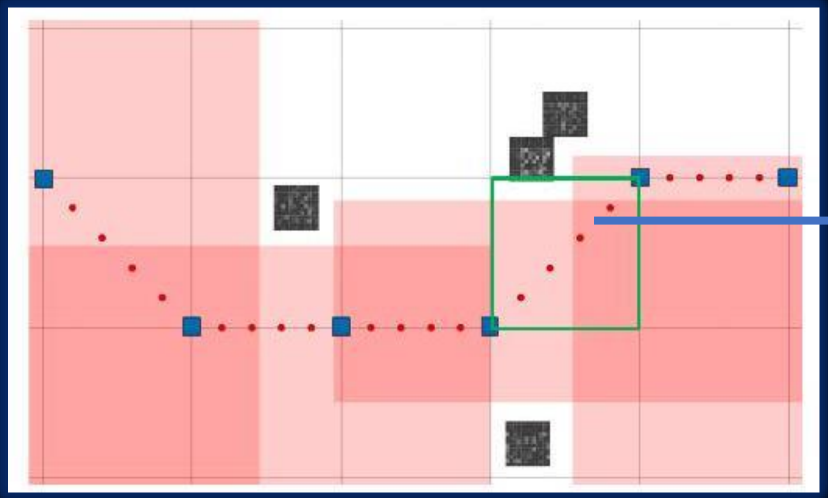
- Enhanced Conflict Based Search(ECBS) is used to solve the path-finding problem with the constraints defined in the previous slide.
- Constraint Tree (CT) with two levels – conflict resolution on high level and optimal path solution at low level is constructed.
- CT expansion continues until a no-conflicts remain and an optimal path for each robot is available.
- Discrete paths of each robot are of the same length, M.

Safe Corridor Construction

Objective:

Construct geometric safe corridors around the planned robot path to model a safe space around of the robot

- **Safe corridor** : A region of safety around the planned discrete path of each robot model from the start position to the end position.
- Construction of a convex polyhedron around each line segment in the path.
- Sequential collection of these polyhedra from start to goal makes up the safe corridor.



Diagonal line segment

Safe Corridor Construction

- Constraint to avoid obstacles in the environment is enforced on every polyhedron

$$\mathcal{R}(a) \cap \mathcal{O} = \emptyset, \forall a \in \mathcal{S}^k$$

where $\mathcal{S}^k$ is the generated polyhedron around path I^k ($r^{k-1} \longrightarrow r^k$)

- The construction algorithm is based on the **axis-search method**. Expansion of the safe corridor in the x and y axes to cover maximum area.
- To ensure diagonal line segments in the path lie inside the safe corridor, each line segment is subsampled into h parts.
- The same process is continued to expand the area around these subsampled line segments

Trajectory Optimization

Objective:

To generate smooth and feasible trajectories out of generated paths

- Discrete paths are converted to trajectories by assigning a time for each waypoints in the path.
- The objective function of the optimization problem is:

$$\text{minimize} \sum_{i=1}^{N} \left(\sum_{j=1}^{H-1} \Delta u_i^{j\,\mathrm{T}} P \Delta u_i^j + \sum_{j=1}^{H} \hat{z}_i^{j\,\mathrm{T}} Q \hat{z}_i^j \right)$$

- The constraints are:

$$z_i^0 = s_i, z_i^H = g_i \ \forall i$$

$$z_i^{j+1} = f\left(z_i^j, u_i^j\right), \ \forall i, j$$

$$\mathrm{pos}\left(z_i^j\right) \in \mathcal{S}_i^j, \ \forall i, j$$

$$u_i^j \in \mathcal{U}_i, \ \forall i, j$$

$$\left\|\mathrm{pos}\left(z_{i_1}^j\right) - \mathrm{pos}\left(z_{i_2}^j\right)\right\| \geq R_{i_1} + R_{i_2}, \forall i_1, i_2, j, i_2 \neq i_1$$

Prioritized Trajectory Optimization

Objective:

To increase probability of obtaining optimal solutions by reducing the chances for obtaining infeasible solutions.

- Priority-based optimization of the trajectories based on grouping of robots and priority assignment to these groups of robots.

- The robot team is subdivided into triples based on the inter-robot constraint

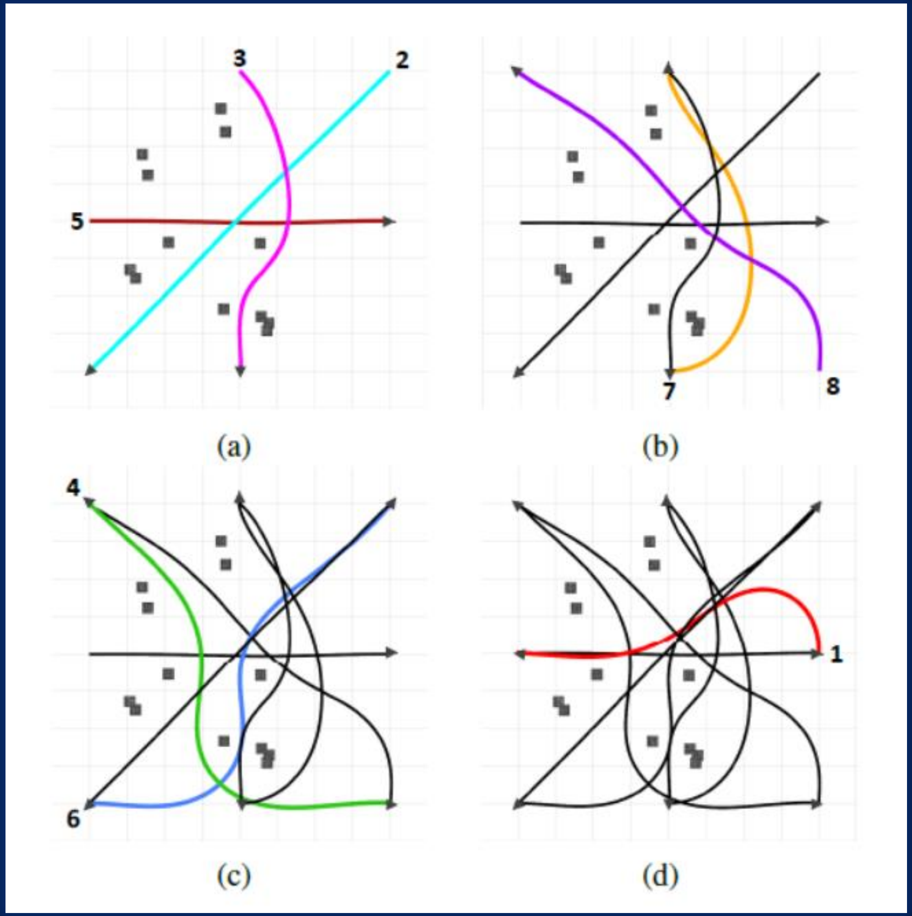
$$\|\mathrm{pos}(r_a^t) - \mathrm{pos}(r_b^t)\| \leq D_{\mathrm{th}}\ , \forall a, b \in \{i_1, i_2, i_3\}, b \neq a$$

where $D_{th} = \sqrt{2}\ D$

and added to a list.

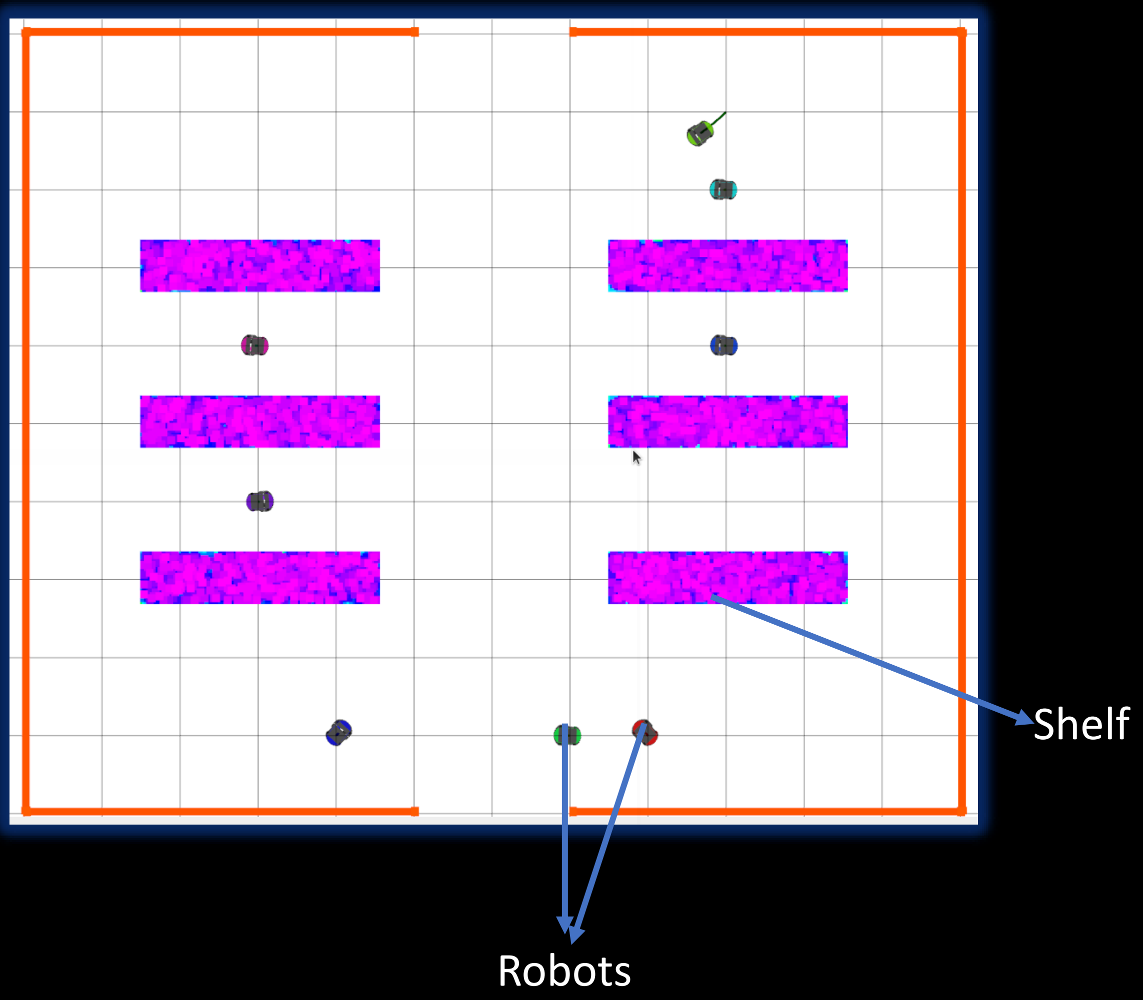
Prioritized Trajectory Optimization

- Priority assignment to groups is based on the frequency of a triple in the list.
- The group obtained after the priority assignment may contain one, two or three robots.
- Robots with low-coupled trajectories are considered as single-robot groups.
- The non-linear optimization problem is applied on these groups of robots to find the optimal trajectories.
- Optimization is performed sequentially from the high-priority groups to the low-priority groups.



Trajectory planning done by groups in the order of priority

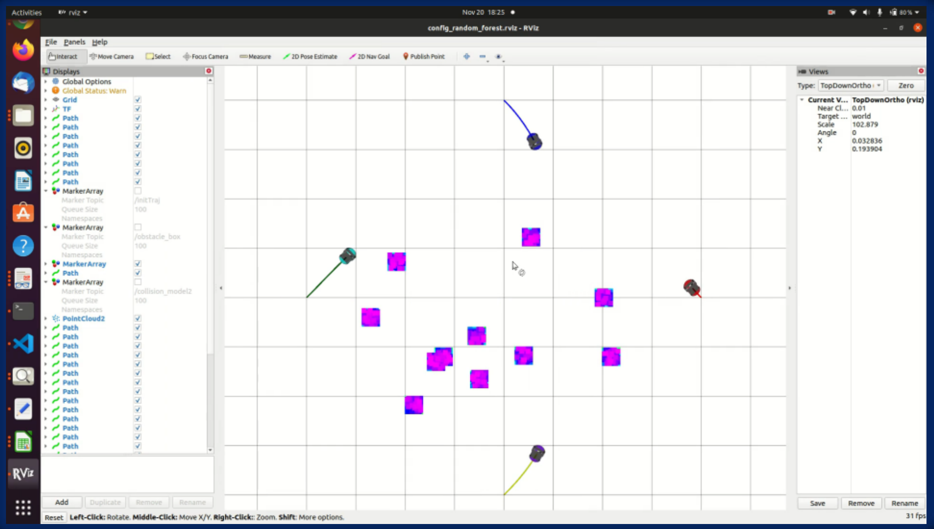
Experiment Scenario



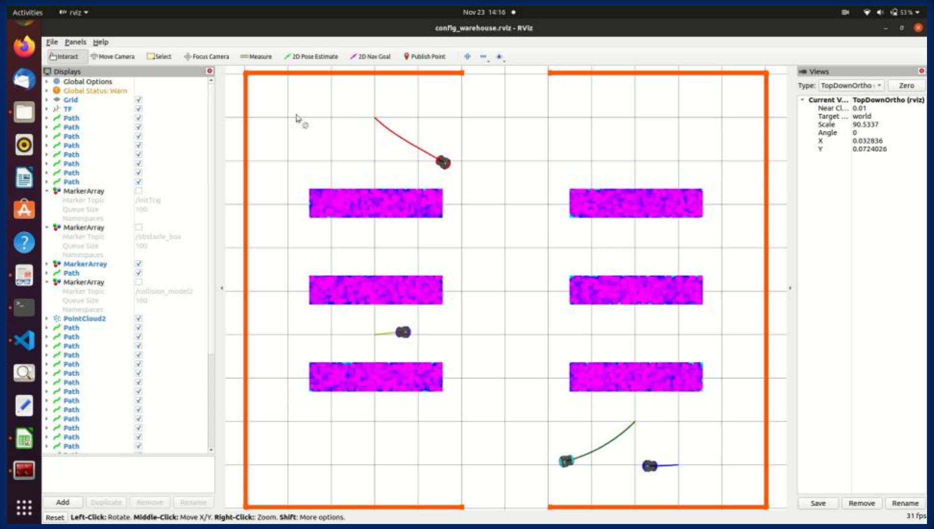
- Warehouse simulation world with dimensions 10m x 12m.
- 6 shelves of dimensions 3m x 0.6m.
- Start and goal positions of the robots are randomly selected – around edges of warehouse of near shelves.

Simulation

- System specs Intel i5-6300HQ@2.30GHz CPU and 12GB of RAM running Ubuntu 16.04 OS.
- Inter point nonlinear programming solver IPOPT is used to solve the optimization problem.
- Collision model of radius R=0.15m, v_max=1m/s and w_max=1rad/s.
- Grid size of graph $D=1m$ and *deltaT* = 1.6s
- Waypoints are divided into 5 parts (h).



Random Plan



Warehouse Plan

Hardware Experiment

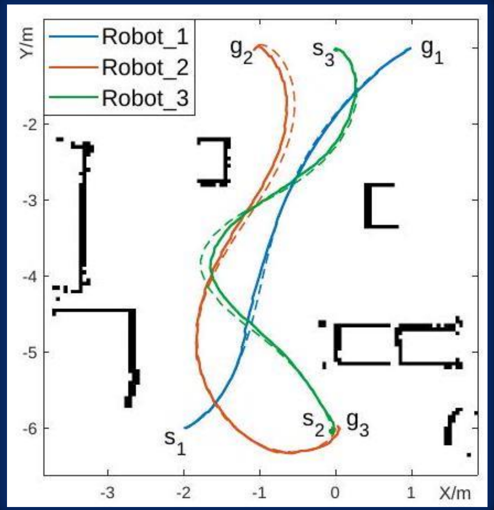
- Three Pioneer 3-AT robots were used to test the approach in a practical setting with dimensions 7m x 8m.
- Velodyne VLP-16 Lidar was mounted on one and Hokuyo 2D Lidars on the others (30m range)

Hardware experiment setup in the reference paper

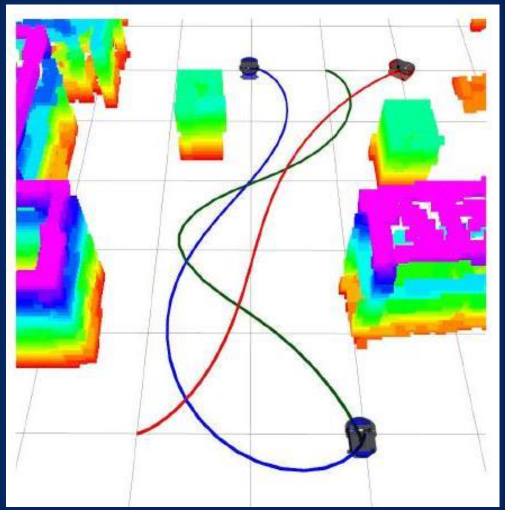
Note: The hardware experiments are not considered within the scope of this report. The experiment results of the reference paper is included for giving the reader a better overall view of the proposed solution.

Hardware Experiment

- LeGO-LOAM odometry and mapping is used to create the map of the environment
- Collision model of radius R=0.3m, v_max=0.6m/s and w_max=0.6rad/s T=2.65s.
- AMCL ROS package is used for localization.
- Model Predictive Control was used to trace the optimal trajectory for each robot.



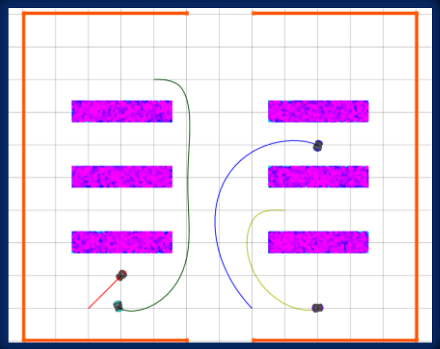
Trajectory generated in the hardware experiment



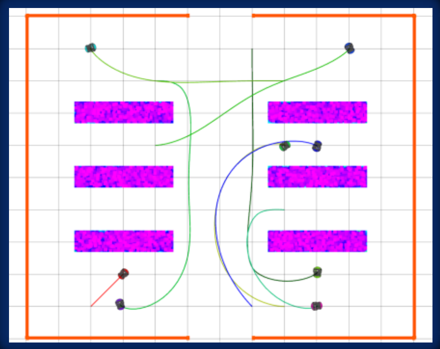
3D map of the Hardware experiment setup

Results

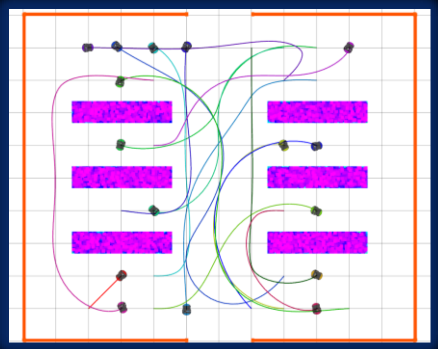
- Simulation tests were conducted on 4, 8, 16, 24 and 32 robot systems.
- Comparative study between coupled trajectory optimization and prioritized trajectory optimization was conducted
- Analysis and comparison of random and priority based grouping strategy was also performed.



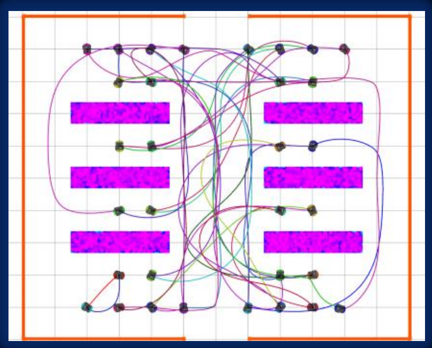
4 agents



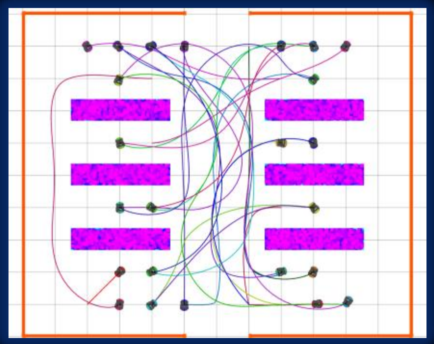
8 agents



16 agents



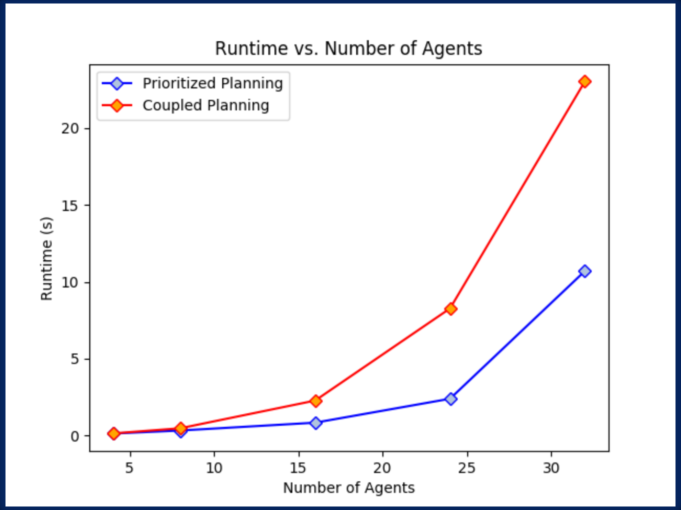
24 agents

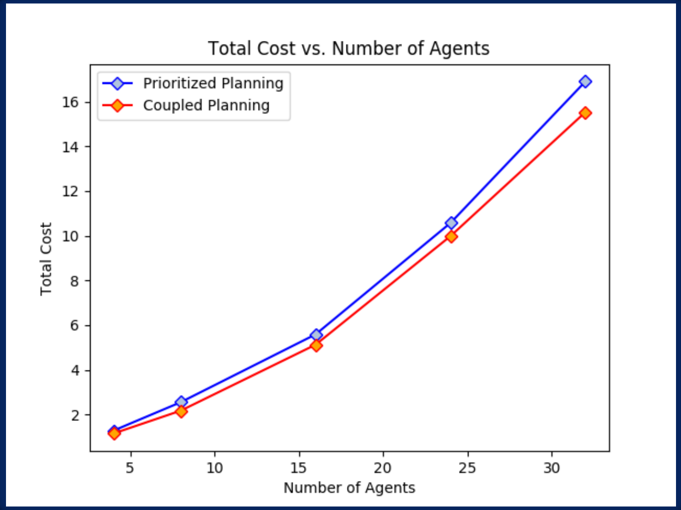


32 agents

Results

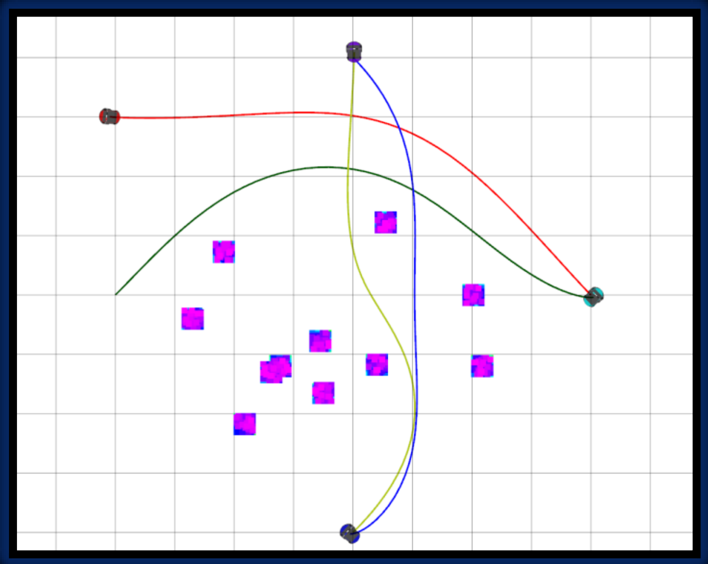
- Runtime was found to be higher for prioritized trajectory optimization in comparison to the coupled strategy.
- Cost of prioritized trajectory optimization is comparatively more than that of the coupled approach.
- Success rate of random grouping strategy optimization with the increase in the robots in the system.



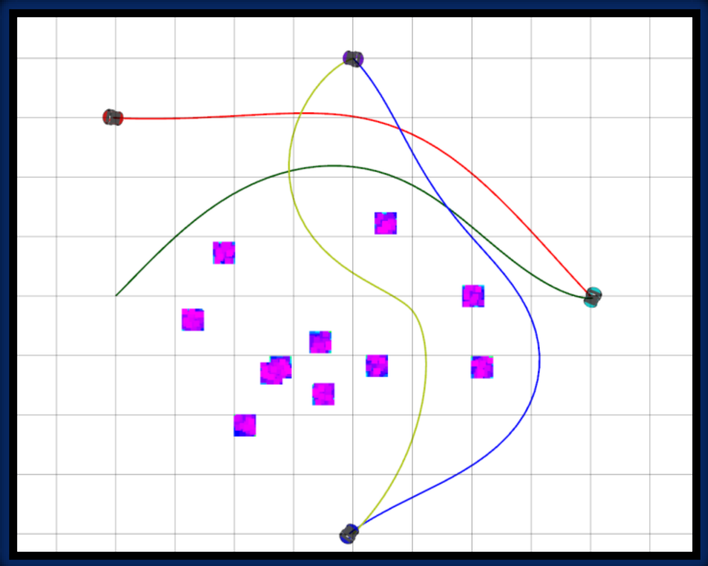


Results

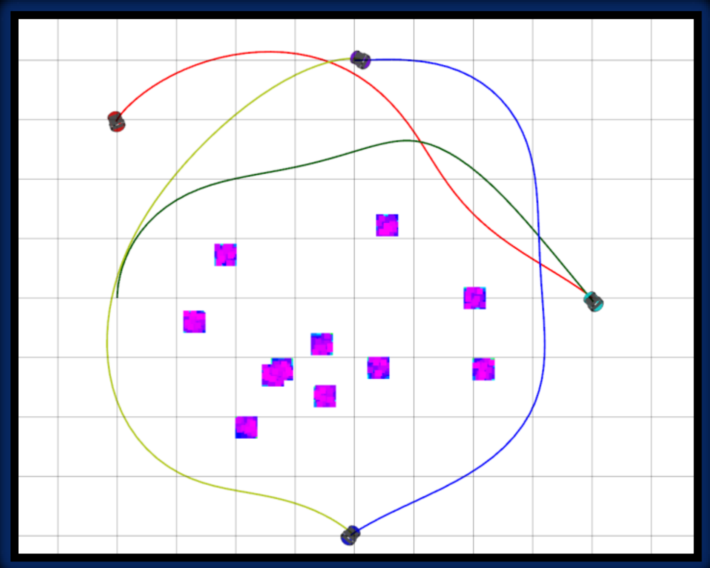
- Trajectory traced by the agents are largely dependent on the collision radius of the agents.



R=0.15



R=0.3



R=0.5

Inferences

- Runtime for prioritized trajectory optimization follows a linear trend in comparison that of the coupled approach implying which showed a steep increase with increase in the number of agents.
- The cost of prioritized optimization strategy was found to be higher due to the occurrence of infeasible solutions that have to be handled during optimization.
- Success rate of priority-based grouping was considerably higher as it was able to better resolve intermediate inter-agent conflicts.
- The collision radius assigned to each which in turn defines its physical space greatly influence the final trajectory. Larger the radius larger the chances for the algorithm to longer trajectories farther from the obstacles.

Conclusion

- Priority-based trajectory planning strategy for multiple non-holonomic robots as trajectory planning strategy is found to provide a guaranteed solution irrespective of the number of agents in the systems at a marginally higher but acceptable cost.
- The proposed strategy is efficient in handling large-scale systems.
- The cost of computation of the solution for the proposed technique marginally higher but within the admissible limits.
- Choice of hyperparameters like collision model radius influence the planned trajectory.

Future Work

- Due to time constraints and practical limitations the experiments of this review were limited to simulations however in the future, we intend to test the proposed strategies on real robot systems and analyze the results

Acknowledgement

We would like to thank Dr. Waseem Malik for the opportunity and the Teaching Assistants – Rishabh Singh and Nitesh Jha for their valuable feedback.