Dynamic Feedback Motion Planning for Car-Like Robots Using Funnel-Graph Algorithm

Iyed DERAR [(1)] *, Reda GUERNANE [(2)],
[(1) (2)] Institute of Electrical & Electronics Engineering, University of Boumerdes, Boumerdes, Algeria
*derariyed@gmail.com

Abstract: This study presents a funnel-based motion planning algorithm for a car-like robot, utilizing a dynamic model to capture the robot's motion. The funnel-based planner addresses the obstacle avoidance problem and dynamically updates the path to guide the robot to its goal. The proposed algorithm's performance is evaluated in a dynamic environment, and with a dynamic goal where re-planning capabilities are demonstrated. The results indicate that funnel planner provides robust navigation even in uncertain conditions.
Keywords: funnel, dynamic model, motion planning, car-like robot.

1. INTRODUCTION

In robotics, motion planning algorithms are essential for guiding systems through their tasks. Traditional methods like PRM and RRT [1][2][3] generate open-loop plans that are passed to online control algorithms, which then manage challenges such as localization errors, control inaccuracies, and unforeseen obstacles [4]. Feedback motion planning, such as potential field methods [5][6][7], addresses these issues by incorporating real-time environmental feedback, enabling closed-loop plans that adapt to unpredictable events during execution. Burridge [8] introduced the concept of funnels, where a series of local controllers guide the robot's state through funnel-shaped regions, eventually reaching the goal. This idea has been adapted in many studies to develop sampling-based feedback strategies [4][8]-[14]. Building on funnel graph generation from [4] and [14], our work extends this approach to support dynamic environments for a dynamic car-like robot model.

2. BACKGROUND AND THEORY

a) The Funnel-Graph:

The funnel-graph is represented as an undirected graph, where each vertex corresponds to a unique neighborhood in free state space with a center point, and edges connect vertices if their neighborhoods intersect but not enclaved within each other. The union of these neighborhoods approximates the free state space, and the sampling of centers is biased away from obstacles within the free state space to ensure maximum safe space for the robot and a minimum number of funnels. This process terminates after satisfying a coverage criteria, providing an approximate representation of the free state space that captures both its area and connectivity through the funnel-graph structure [4]. The robot is then guided through the centers of funnels and the convergence to any goal is guaranteed from any start in the environment and provided that both the start and the goal are captured withing the graph.

b) The Dynamic Car-Like Robot Model:

To capture the motion dynamics of the car-like robot the following equations from [15] were adopted:

$$\begin{aligned} \dot{x} &= \vartheta\cos\theta \\ \dot{y} &= \vartheta\sin\theta \\ \dot{\theta} &= \omega \\ \dot{\vartheta} &= \frac{F}{m} \\ \dot{\omega} &= \frac{\tau}{J} \end{aligned} \qquad (1)$$

Where (x, y) is the cartesian position of the robot, θ is the orientation, ϑ is the linear velocity, ω is the angular velocity, F is the applied force, m is the mass, J is the moment of inertia and τ is the applied torque.

In this work, we are going to use the same method used in [15] to control the position of the robot, where feedback linearization about a hand position L was used.

$$\begin{aligned} h_x &= x + L\cos\theta \\ h_y &= y + L\sin\theta \end{aligned} \qquad (2)$$

We differentiate (2) twice

$$\begin{pmatrix} \ddot{h}_x \\ \ddot{h}_y \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} \frac{F}{m} - L\omega^2 \\ \frac{\tau L}{J} + \vartheta\omega \end{pmatrix} \quad (3)$$

After some manipulations:

$$\begin{pmatrix} F \\ \tau \end{pmatrix} = \begin{pmatrix} m & 0 \\ 0 & \frac{J}{L} \end{pmatrix} \left(\begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} \ddot{h}_x \\ \ddot{h}_y \end{pmatrix} - \begin{pmatrix} -L\omega^2 \\ \vartheta\omega \end{pmatrix} \right) \quad (4)$$

Using $(\ddot{h}_x, \ddot{h}_y)$ as virtual inputs, we can then use simple state feedback to control the position (x, y). From [15] all the zero dynamics are stable.

3. The Funnel-Graph Algorithm:

The Funnel-Graph Algorithm begins by estimating the obstacle-free area in the map and initializing the graph at the goal, estimating initial coverage. It then samples a random state within the map's limits, ensuring it is collision-free and estimating the obstacle-free neighborhood around the state. An exponential sampling rejection term is applied to bias funnels toward larger areas,

Where samples are rejected if the bellow condition is satisfied minimizing the number

$$Uniform[0,1] > e^{-\frac{1}{neighbourhood\ radius}} \quad (5)$$

of funnels. The algorithm ensures that the sample is not within an existing funnel's neighborhood but intersects with at least one to guide expansion. The new node is added as a funnel, with edges based on the distance between centers of intersecting neighborhoods. A path (centers of the funnels) is set once the start state and desired coverage are reached, using Dijkstra's algorithm to navigate the graph. A dedicated function continuously estimates the obstacle-free area until the coverage criteria are met. Figure 1 and figure 2 illustrate an experiment where a coverage criterion of 96 percent is used : .

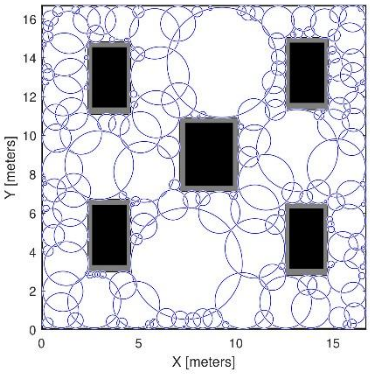

Figure 1 : Visualization of the funnel-graph Algorithm.

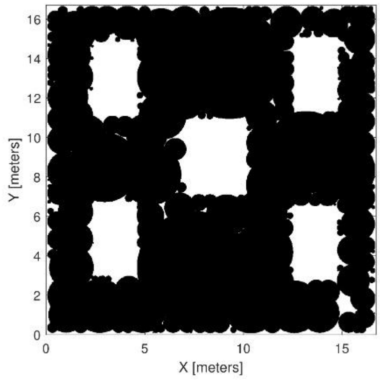

Figure 2 : Coverage estimation map.

4. Execution Phase:

a) In a static environment:

Figure 3 shows the execution phase of the algorithm where the goal is set at [15.5, 15.5] and the start at [1, 1] where the blue circles represent the path funnels.

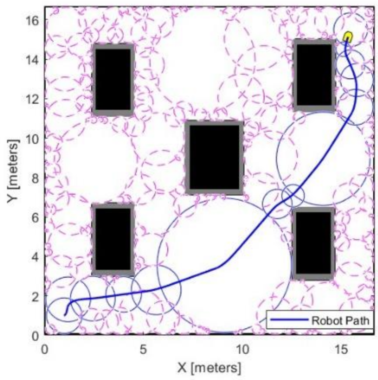

Figure 3 : Execution in a static environment

b) In a dynamic environment:

Figures 4 and 5 depict the scenario when the robot faces a not-anticipated road blockage the algorithm adapts to the event and adjusts the connectivity of the graph accordingly and generate new path funnels ensuring a safe arrival to the goal.

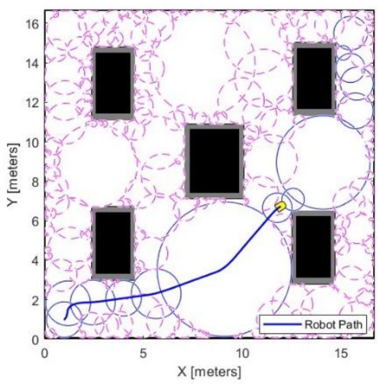

Figure 4 : Before the blockage appears.

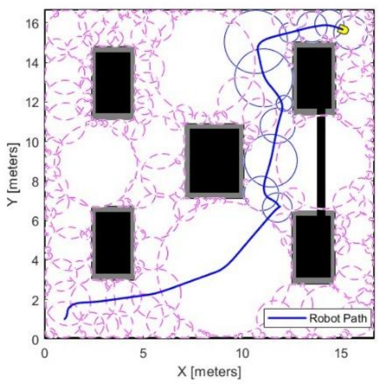

Figure 5: Adaptation after the blockage of the initial path.

c) With a dynamic goal in a dynamic environment

In this scenario, the Follower robot (yellow with a red trace) continuously adapts its position to follow the Leader robot (red with a blue trace), leveraging a graph data structure to find the shortest obstacle-free path between them. The algorithm ensures that the Follower tracks the Leader by frequently updating their positions within a predefined funnel. In a dynamic environment, where obstacles change frequently, the Follower robot must adjust in real-time, finding new paths to avoid the obstacles while maintaining proximity to the Leader. This tests the algorithm's ability to handle rapid environmental changes.

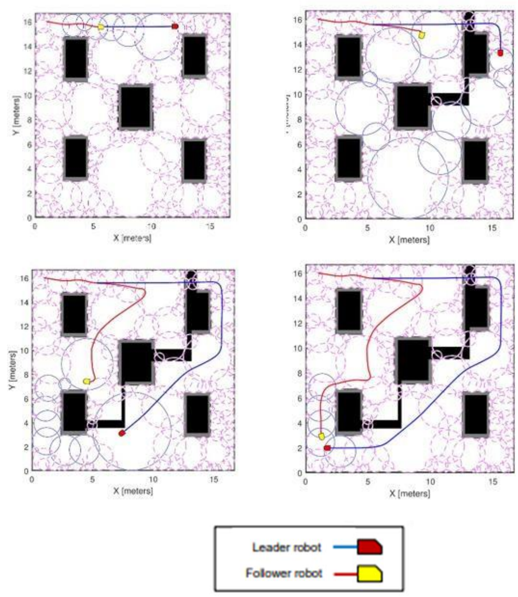

Figure 6 : Leader-Follower in a dynamic environment .

References

[1] S. M. LaValle. Rapidly-exploring random trees:
A new tool for path planning, 1998.

[2] . E. Kavraki, P. Svestka, J. -. Latombe and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," in IEEE Transactions on Robotics and Automation, vol. 12, no. 4, pp. 566-580, Aug. 1996.

[3] S. LaValle, Planning Algorithms, Cambridge, U.K.:Cambridge Univ. Press, 2006.

[4] S. M. L. Libo Yang, "The sampling-based neighborhood graph: An approach to computing and executing feedback motion strategies," IEEE Transactions on Robotics and Automation, June 2004.

[5] Hwang, Y.K. and Ahuja, N., 1992. A potential field approach to path planning. *IEEE transactions on robotics and automation*, *8*(1), pp.23-32.

[6] Bounini, F., Gingras, D., Pollart, H. and Gruyer, D., 2017, June. Modified artificial potential field method for online path planning applications. In *2017 IEEE Intelligent*

Vehicles Symposium (IV) (pp. 180-185). IEEE.

[7] Warren, C.W., 1989, January. Global path planning using artificial potential fields. In *1989 IEEE International Conference on Robotics and Automation* (pp. 316-317). IEEE Computer Society.

[8] R. R. Burridge, A. A. Rizzi, and D. E. Koditschek. Sequential composition of dynamically dexterous robot behaviors. International Journal of Robotics Research, 18(6):534555, June 1999.

[9] R. Tedrake, I. R. Manchester, M. Tobenkin, J. W. Robterts, LQRTrees: Feedback Motion Planning via Sums of Squares Verification, International Journal of Robotics Research, Vol 29, Issue 8, pp 10321052, 2010

[10] E. Ege and M. M. Ankarali, Feedback motion planning of unmanned surface vehicles via random sequential composition, Transactions of the Institute of Measurement and Control, vol. 41, no. 12, pp. 33213330, 2019.

[11] Durmaz, A., Ozdemir, O. and Ankarali, M.M., 2024. Feedback motion planning via sequential composition of random elliptical funnels. *Ocean Engineering*, *310*, p.118694.

[12] R. Tedrake, LQR-trees: Feedback motion planning on sparse randomized trees, Robotics: Science and Systems V, 2009

[13] Karagoz, O.K., Atasoy, S. and Ankarali, M.M., 2020, October. MPC-graph: Feedback motion planning using sparse sampling-based neighborhood graph. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 6797-6802). IEEE.

[14] Özcan, M. and Ankarali, M.M., 2019, October. Feedback motion planning for a dynamic car model via random sequential composition. In *2019 IEEE international conference on systems, man and cybernetics (SMC)* (pp. 4239-4244). IEEE.

[15] Lawton, J.R., Beard, R.W. and Young, B.J., 2003. A decentralized approach to formation maneuvers. *IEEE transactions on robotics and automation*, *19*(6), pp.933-9