Project description—Time-Efficient Curvature Constrained Planning in the Presence of Obstacles

1 Introduction

In this project, we are interested in computing *curvature constrained* paths for a robot moving amidst obstacles in the plane. Namely, we have a robot that has a minimum turning radius that it can follow. This constraint allows to model a wide range of systems such as UAVs and steerable needles and has attracted increasing attention in the planning community [1, 5, 6, 7]. Arguably, the best-known model for such systems is the *Dubins model* [2, 4]. Here, the robot is assumed to be a planar system that can only move forward with a constant velocity and a minimum turning radius. The state of the system is defined by its position (the location of a predefined reference point on the robot) and orientation (the heading of the robot). In the absence of obstacles, the minimal-length path can be computed analytically [2]. Unfortunately, when the environment contains obstacles, the problem is known to be NP-hard [8].

Here, we are interested in a slightly-more complicated cost function—minimizing travel time (and not path length). In many settings, taking longer manoeuvres at a higher speed is favourable (with respect to travel time) when compared to short, but slow, manoeuvres. When there are no obstacles in the environment, several approaches exist for computing time-optimal or time-efficient manoeuvres. Wolek et al. [11] developed a list of possible cases that are solved by non-linear optimization to compute time-efficient manoeuvres. However, the solver needs proper initialization and thus cannot guarantee that the optimum is found.

An alternative approach was recently presented by Kučerová et al [3] who proposed a heuristic approach that utilizes a Dubins path [2, 4] with two turn segments and a central straight segment, for which both turning radii are optimized to get the fastest trajectory possible. The travel time of the trajectory is computed based on a speed profile that takes into account both speed and forward acceleration limits. Their approach exploits computationally efficient closed-form solutions for Dubins path [2] which can be determined in microseconds [10]. For a visualization, see Fig. 1. Unfortunately, their approach only considers the setting where the start and target of a path are sufficiently far apart.

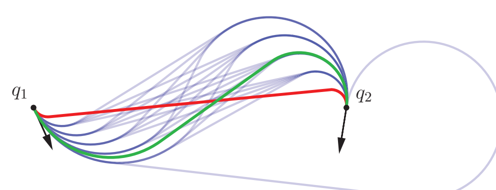

Figure 1: Example of generated Dubins trajectories for various combinations of the initial and final turning radii (blue). The fastest trajectory is in green, and the shortest one in red. Figure adapted from [3].

When the environment contains obstacles, one can discretize the environment and use search-based algorithms such as A* [9]. This builds upon the underlying assumption that an optimal path can be constructed by multiple locally-optimal short paths. However, this contradicts the assumption taken by Kučerová et al. that the paths are sufficiently far apart. For a visualization, see Fig. 2.

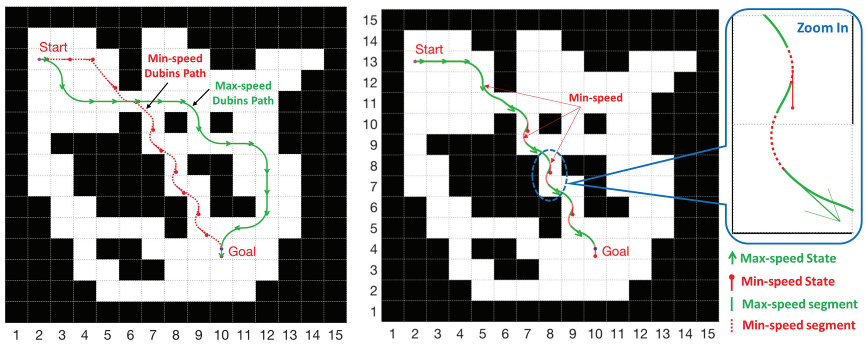

Figure 2: The Dubins paths vs. time-optimal path in an obstacle-rich environment. Left: Dubins paths using minimal (red) and maximal speed (green) of cost 55.95sec and 35.99sec, respectively. Right: Time-optimal path of cost 34.51. Figure adapted from [9].

2 Problem definition

In this project we will explore how the approach of Kučerová et al. can be used as a *post-processing* step to improve the quality of paths computed using a search-based algorithm.

We are given a curvature-constrained point robot operating in a planar environment amidst a set of polygonal obstacles (an exact description of the robot's motion model can be found in [3]). We are given start and target configurations q_s, q_t such that $q_s = (x_s, y_s, \theta_s)$ where (x_s, y_s) and θ_s describe the robot's position and orientation, respectively (similarly for q_t).

Given an environment cluttered with obstacles, we start by running an A*-like algorithm to compute a candidate path connecting q_s and q_t. The A*-like algorithm will use as motion primitives Dubins paths[1]. Let $\pi = (q_1, \ldots, q_n)$ be such a path where each q_i is a robot configuration and the Dubins path between any pair of consecutive configurations q_i and q_{i+1} is collision free.

We can take any pair of configurations q_i and q_j such that $i < j$ and q_i and q_j are sufficiently far apart, and use the approach of Kučerová et al. to compute the high-quality path $\pi^*_{i,j}$ connecting them. This path may collide with the obstacles thus it needs to be validated. If it is valid, we replace the subpath connecting q_i and q_j in π with $\pi^*_{i,j}$. If $\pi^*_{i,j}$ is invalid, we discard it and repeat the process.

[1] *ADVANCED:* An alternative approach would be to take time-optimal paths with no limits on the robot's acceleration. In contrast to using Dubins paths which give us an upper bound on the quality of a solution, this will provide a lower bound on the quality of a solution. What are the implications of using this approach?

3 Project description

The project will start with an implementation of the A*-like algorithm to compute a candidate path connecting q_s and q_t. For details on such a search algorithm (with more complicated local connections) see [9].[2] The second step would be to implement the approach of Kučerová et al. to compute the high-quality path connecting two configurations.

Once these two building blocks are implemented, we need a post-processing strategy: How do we choose which pairs of configurations to take? For how many iterations should we run this approach? How many radii should we use in Kučerová algorithm (there is an inherent trade-off there between speed of computation and quality of the result). Implement several post-processing strategies and run a series of benchmarks to evaluate the quality of the approach.

BONUS: How would the project change if we take the alternative approach mentioned in footnote (1)? Implement it and report on the results.

References

[1] Yongbo Chen, Jianqiao Yu, Yuesong Mei, Siyu Zhang, Xiaolin Ai, and Zhenyue Jia. Trajectory optimization of multiple quad-rotor UAVs in collaborative assembling task. *Chinese Journal of Aeronautics*, 29(1):184–201, February 2016.

[2] Lester E Dubins. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of mathematics*, 79(3):497–516, 1957.

[3] Kristýna Kučerová, Petr Váň, and Jan Faigl. On finding time-efficient trajectories for fixed-wing aircraft using dubins paths with multiple radii. In *ACM Symposium on Applied Computing*, page 829–831, 2020.

[4] Steven M. LaValle. *Planning Algorithms*. Cambridge University Press, Cambridge, U.K., 2006.

[5] Yuanchang Liu and Richard Bucknall. A survey of formation control and motion planning of multiple unmanned vehicles. *Robotica*, pages 1–29, 2018.

[6] Douglas G. Macharet and Mario F. M. Campos. A survey on routing problems and robotic systems. *Robotica*, page 1–23, 2018.

[2]Here we need to choose if the A*-like planner use the Dubins path with the minimum or maximum radius (computed based on speed limits)? There are pros and cons to both of them. If we choose to use a minimum radius (which implies minimum speed) then the speed profile (acceleration on straight segments) should be computed directly in the search to prefer faster trajectories with shorter turns. Post-optimization may increase the radius to increase speed. If we choose to use a maximum radius (which implies maximum speed) then the travel time of a trajectory computed using the search is easily obtained but some solutions may be missed due to the larger radius. Post-optimization may decrease the radius to shorten the trajectory.

[7] Alena Otto, Niels Agatz, James Campbell, Bruce Golden, and Erwin Pesch. Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: A survey. *Networks*, 2018.

[8] John Reif and Hongyan Wang. The complexity of the two dimensional curvature-constrained shortest-path problem. In *Workshop on the Algorithmic Foundations of Robotics (WAFR)*, pages 49–57, 1998.

[9] Junnan Song, Shalabh Gupta, and Thomas A Wettergren. T*: Time-optimal risk-aware motion planning for curvature-constrained vehicles. *IEEE Robot. Autom. Lett.*, 4(1):33–40, 2018.

[10] Petr Vána and Jan Faigl. Optimal solution of the generalized dubins interval problem. In *Robotics: Science and Systems (RSS)*, 2018.

[11] Artur Wolek, Eugene M Cliff, and Craig A Woolsey. Time-optimal path planning for a kinematic car with variable speed. *Journal of Guidance, Control, and Dynamics*, 39(10):2374–2390, 2016.