Faculty of Electrical Engineering
Department of Cybernetics

Bachelor's thesis

Time-optimal trajectory planning for vehicles with minimum turning radius and variable velocity

Kristýna Kučerová

August 2020

Supervisor: prof. Ing. Jan Faigl, Ph.D.
Supervisor specialist: Ing. Petr Váňa

BACHELOR'S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Kučerová Kristýna** Personal ID number: **474523**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Cybernetics**

Study program: **Open Informatics**

Branch of study: **Computer and Information Science**

II. Bachelor's thesis details

Bachelor's thesis title in English:

Time-Optimal Trajectory Planning for Vehicles with Minimum Turning Radius and Variable Velocity

Bachelor's thesis title in Czech:

Plánování časově optimálních trajektorií pro vozidla s omezeným poloměrem zatáčení

Guidelines:

1. Familiarize yourself with trajectory planning approaches for non-holonomic vehicles with minimal turning radius [1] and their extensions for time-optimal [2] and energy-optimal [3] variants.
2. Propose an extension of the Dubins path by considering different turning radius for each of the turns separately.
3. Consider also using multiple segments for each turn and optimization of the turning radii of the segments.
4. Utilize the proposed trajectory planning between two given configuration in multi-goal scenarios to find time-efficient trajectories, as for the Dubins Traveling Salesman Problem [4].
5. Evaluate the performance of the proposed approach(es) and compare it with the existing method(s) [2].

Bibliography / sources:

[1] Dubins, Lester E: On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. American Journal of Mathematics, vol. 79, no. 3, pp. 497–516, 1957.
[2] Wolek, Artur, Eugene M. Cliff, and Craig A. Woolsey. Time-optimal path planning for a kinematic car with variable speed. Journal of Guidance, Control, and Dynamics vol. 39, no. 10, pp. 2374-2390, 2016.
[3] Tokekar, Pratap, Nikhil Karnad, and Volkan Isler. Energy-optimal trajectory planning for car-like robots. Autonomous Robots, vol. 37, pp. 279-300, 2014.
[4] P. Oberlin, S. Rathinam and S. Darbha, Today's Traveling Salesman Problem, IEEE Robotics & Automation Magazine, vol. 17, no. 4, pp. 70-77, 2010.

Name and workplace of bachelor's thesis supervisor:

prof. Ing. Jan Faigl, Ph.D., Artificial Intelligence Center, FEE

Name and workplace of second bachelor's thesis supervisor or consultant:

Date of bachelor's thesis assignment: **10.01.2020** Deadline for bachelor thesis submission: **14.08.2020**

Assignment valid until: **30.09.2021**

prof. Ing. Jan Faigl, Ph.D.	doc. Ing. Tomáš Svoboda, Ph.D.	prof. Mgr. Petr Páta, Ph.D.
Supervisor's signature	Head of department's signature	Dean's signature

III. Assignment receipt

The student acknowledges that the bachelor's thesis is an individual work. The student must produce her thesis without the assistance of others, with the exception of provided consultations. Within the bachelor's thesis, the author must state the names of consultants and include a list of references.

. _____
Date of assignment receipt

_____
Student's signature

Declaration

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

Prague, August 14, 2020

.....
Kristýna Kučerová

Acknowledgment

I would like to thank prof. Ing. Jan Faigl, Ph.D. for supervising my thesis and bringing order to chaos. I would also like to thank my supervisor specialist Ing. Petr Váňa for his valuable insight into the given topic and never-ending optimism. Many thanks to all of my friends who made the time at CTU more enjoyable, especially Petra Štefaníková (who moved me by mentioning me in her thesis). And finally, big thank you to my family for motivating me throughout my studies, and for not killing me during the isolation.

Abstract

In the thesis, we propose a novel method for solving the problem of finding the time-optimal trajectory for vehicles with limited turning radius, variable speed, and bounded acceleration. The proposed method extends the Dubins vehicle model for which the path consists of arcs and a straight line. Unlike the Dubins path, the proposed trajectory might have different initial and final turning radius selected to minimize the total time needed to travel the path. The proposed multi-radius Dubins path for the specified Dubins vehicle model is then utilized in time-optimal multi-goal trajectory planning. The problem is defined as the time-optimal variant of the Dubins Traveling Salesman Problem (DTSP) and solved using the Variable Neighborhood Search (VNS) combinatorial metaheuristic. In comparison to the Dubins path, the reported results for the vehicle's parameters of the fixed-wing aircraft Cessna 172 show a decrease in the travel time in point-to-point trajectories about 5–20 % on average, and about 15–30 % on a trajectory over multiple locations.

Keywords: Dubins vehicle; time-optimal trajectory planning; multi-radius Dubins path; speed profile; Dubins traveling salesman problem

Abstrakt

V této práci navrhujeme novou metodu plánování časově-optimální cesty pro vozidla s omezeným poloměrem zatáčení, proměnnou rychlostí a limitovanou akcelerací. Navrhovaná metoda rozšiřuje model Dubinsova vozidla, pro který se optimální cesta skládá z částí kružnice o minimálním poloměru zatáčení a úsečky. Na rozdíl od Dubinsovy cesty, navrhovaný model uvažuje různé poloměry zatáčení na začátku a konci cesty. Vhodnou volbou poloměrů zatáčení je nalezena časově optimální trajektorie splňující dané omezení rychlosti a zrychlení. Navržená metoda je dále použita při řešení časově-optimálního plánování cesty přes více cílů, formulované jako časově-optimální varianta úlohy obchodního cestujícího pro Dubinsovo vozidlo (DTSP). Úloha je řešena kombinatorickou metaheuristikou Variable Neighborhood Search (VNS) s využitím nalezených časově-optimálních trajektorií mezi jednotlivými cíli. V práci reportované výsledky (pro model parametrizovaný dle letadla Cessna 172) indikují časové zkrácení trajektorií v průměru o 5 až 20 % v porovnání s řešením založeným na klasickém modelu Dubinsova vozidla. V případě trajektorií přes více cílů dosahuje navrhovaná metoda časového zkrácení o 15 až 30 %.

Klíčová slova: Dubinsovo vozidlo; časově-optimální plánování; Dubinsova cesta s více poloměry; rychlostní profil; problém obchodního cestujícího pro Dubinsovo vozidlo

Used Abbreviations

CTU	Czech Technical University in Prague
DTP	Dubins Touring Problem
DTSP	Dubins Traveling Salesman Problem
TSP	Traveling Salesman Problem
TTE	Travel Time Estimate
VNS	Variable Neighborhood Search

Contents

List of Figures

List of Tables

List of Algorithms

CHAPTER 1

Introduction

In this thesis, we study planning of time-optimal curvature-constrained trajectories that are feasible for vehicles with limited turning radius, variable speed, and bounded acceleration. The curvature-constrained trajectory for a vehicle with a defined minimum turning radius can be modeled using the Dubins vehicle [1], which moves only forward with a constant speed. Since the Dubins vehicle has constant forward speed, such a trajectory is called the Dubins path. The Dubins path is the shortest in a 2D plane between two configurations (locations) with a specified vehicle's heading angle and given turning radius, and has a closed-form solution. The Dubins path consists of three segments, which can be an arc with the minimum turning radius, or a straight line. The optimal closed-form solution supports constant computational complexity [2], and the Dubins path can be determined in microseconds using nowaday computational resources [3]. However, the regular Dubins vehicle model is limited by the assumption of constant forward speed and single (minimal) turning radius. Therefore, we aim to propose a generalization of the Dubins vehicle model towards time-optimal trajectories that would also be computationally effective.

The main idea of the proposed approach is considering the acceleration of the vehicle on the straight segment of the Dubins path; however, the crux of the achieved reduction of the travel time along the found trajectories is in considering a larger turning radius and different initial and terminal turning radii. Thus, the Dubins path with two arcs and a straight line segment consists of the first arc with the initial turning radius, followed by the straight line segment, and terminated by the arc of different turning radius. Generally, using a larger turning radius allows a higher forward speed of the vehicle; hence, an overall faster trajectory can be determined. By using a larger radius, the total trajectory length can be increased, and the trade-off between the length increase and time decrease needs to be found. Hence, a speed profile needs to be computed to obtain the speed throughout the trajectory and the total travel time, based on the particular turning radii. The concept of multiple radii has already been utilized for finding time-optimal [4], energy-optimal trajectories [5], and safe emergency landing trajectories [6]. However, none of the mentioned work deals with the time-optimal trajectory planning with a limited acceleration of the vehicle. Therefore, we aim to fill this niche and propose time-optimal trajectory planning for a vehicle with a limited turning radius, variable speed, and bounded acceleration.

In the first part of the thesis, we focus on trajectory computation between two configurations of the vehicle, where the configuration is given by vehicle position, forward speed, and heading angle. We propose to extend the Dubins vehicle model to take advantage of larger turning radii, which allows increasing the speed and generate faster trajectories [7]. An example of the proposed multi-radius Dubins paths is shown in Fig. 1.1. The travel time improvement of the shortest trajectory (red) with a speed increase on the straight segment in comparison to the trajectory traveled at a constant speed is 38 %. The fastest trajectory (green) reaches 54 % travel time improvement compared to the shortest trajectory with constant speed, which is an additional 16 %. Another trajectory is also proposed in the thesis, utilizing variable radii and multiple segments for time-optimal trajectory planning for the specified vehicle. The trajectory allows further time decrease in comparison with the multi-radius Dubins path, but is harder to optimize.

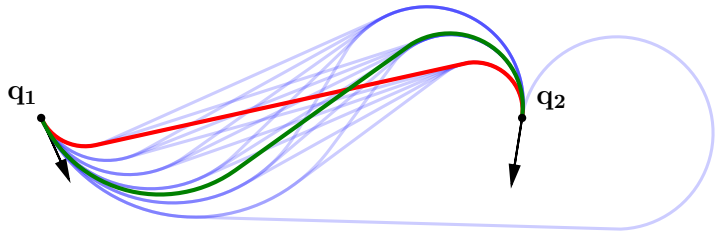


Figure 1.1: Example of extended Dubins paths with various turning radii. The original Dubins path with minimal turning radius is marked red, possible trajectories using multiple radii Dubins path are blue, and the fastest trajectory found using local optimization is marked with green color.

The second part of the bachelor thesis is dedicated to employing the proposed multi-radius Dubins path in multi-goal trajectory planning formulated as a variant of the Traveling Salesman Problem (TSP) with curvature-constrained trajectories, in particular, the Dubins TSP [8]. Both the TSP and DTSP are known to be NP-hard [9, 8], and the proposed time-optimal variant is even more challenging. Therefore, we solve the extended variant of DTSP by the combinatorial metaheuristic Variable Neighborhood Search (VNS) [10], already deployed in routing problems with Dubins vehicle [11, 12, 13]. The VNS is particularly utilized because it can improve the solution by moving it out of local optima, and allows to incorporate the proposed model into the computation easily.

The thesis is structured as follows. The addressed time-optimal trajectory planning is formally introduced in Chapter 2. The proposed methods for determining the trajectory between two configurations is described in Chapter 3. The results of the empirical evaluation of the two proposed trajectories are reported in Chapter 4. The employment of the proposed method in the solution of the multi-goal planning and its empirical evaluation is presented in Chapter 5. The conclusion is in Chapter 6.

CHAPTER 2

Problem Statement

The problem studied in the thesis is the optimization problem of finding the fastest curvature-constrained trajectory feasible for a vehicle with limited turning radius, variable speed, and maximal acceleration/deceleration. The considered vehicle model is the Dubins vehicle, extended by using variable speed. The speed on an arc segment of the Dubins path is limited, and by increasing the turning radius, the speed can be increased as well, which allows decreasing the total travel time. Here, it is worth mentioning that the vehicle's forward acceleration is bounded to make the model more accurate, which contrasts with time-optimal trajectories proposed in [4]. The rest of the chapter is organized as follows. The travel time optimization problem of finding the fastest trajectory between two configurations with the motion constraints is formulated in Section 2.1. The extended formulation for multi-goal trajectory planning is presented in Section 2.2.

2.1 Time-Optimal Trajectory Planning

The addressed problem of the time-optimal trajectory is to determine curvature-constrained trajectory between two configurations, where the configurations correspond to the vehicle state $\mathbf{q}$,

$$\mathbf{q} = \langle \mathbf{p}, \theta \rangle, \quad \mathbf{q} \in SE(2), \tag{2.1}$$

where $\mathbf{p}$ is the position in $\mathbb{R}^2$ and θ is the heading angle of the vehicle at $\mathbf{p}$ from the interval $[0, 2\pi)$. The vehicle is allowed to move forward with variable speed v along the trajectory with the curvature κ. The vehicle motion can be described as the Dubins vehicle [1]

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = v \begin{bmatrix} \cos\theta \\ \sin\theta \\ \kappa \end{bmatrix}. \tag{2.2}$$

The curvature κ represents the current turning rate, and it is a scalar value, which can be positive or negative. The sign of the curvature indicates the turn direction. The positive sign denotes counter-clockwise (left) turn, and negative is clockwise (right) turn. The curvature can be computed at any given point as

$$\kappa = \frac{\dot{x}\ddot{y} - \dot{y}\ddot{x}}{(\dot{x}^2 + \dot{y}^2)^{\frac{3}{2}}}, \tag{2.3}$$

and the turning radius can be obtained as $r = |\kappa^{-1}|$. The vehicle is limited by the minimal turning radius $r_{\min}$, and thus the curvature κ is constrained by

$$|\kappa| \leq \frac{1}{r_{\min}}. \tag{2.4}$$

The speed of the vehicle v is variable and is assumed to be limited to the interval

$$v \in [v_{\min}, v_{\max}], \quad v_{\min}, v_{\max} > 0. \tag{2.5}$$

For the aerial vehicle, the value $v_{\min}$ is larger than the vehicle's stall speed, which allows the flight in still air. The value $v_{\max}$ depends on the physical capabilities of the vehicle. The speed is further constrained based on the curvature such that

$$v \leq \sqrt{\frac{g \tan(\varphi_{\max})}{|\kappa|}}, \tag{2.6}$$

where g is the gravitational acceleration, and $\varphi_{\max}$ represents the maximal bank angle. The bank angle is the angle at which the fixed-wing aircraft can be tilted sideways, and is used to change its turn rate. The speed limitation (2.6) ensures that the aircraft can compensate for the centrifugal force and maintain a steady flight configuration with zero side-slip without exceeding the maximum bank angle.

The forward acceleration and deceleration are also restricted to make the vehicle model more accurate. The acceleration $\dot{v}$ is limited by its minimal and maximal values

$$\dot{v} \in [a_{\min}, a_{\max}], \quad a_{\min} < 0, \ a_{\max} > 0. \tag{2.7}$$

The formulas (2.1–2.7) ensure that the defined constraints for the vehicle motion are fulfilled. The problem of finding the fastest trajectory Γ between two configurations is formulated as a continuous optimization Problem 1, where the minimized variable is the terminal travel time T of the trajectory Γ.

Problem 1 (Time-optimal trajectory planning between two configurations)

$$\begin{aligned} \min_{\Gamma} \quad & T, \\ \text{s.t.} \quad & \Gamma : [0, T] \to SE(2), \\ & \Gamma(0) = \mathbf{q_1}, \quad \Gamma(T) = \mathbf{q_2}, \\ & (2.1 - 2.7) \text{ are met.} \end{aligned}$$

2.2 Multi-Goal Time-Optimal Trajectory Planning

The multi-goal time-optimal trajectory planning is an extension of Problem 1, where the time-optimal trajectory is visiting n specified locations. The problem consists of two main parts. First, we need to determine the visiting order of the locations. This is a combinatorial optimization problem that is a variant of the TSP to find a cost-efficient closed tour over the given set of locations [9]. Besides, we need to find the most suitable heading angle for each location and determine the connecting trajectories between every two subsequent locations on tour. Thus, the second part is a continuous optimization problem, where finding individual trajectories is a solution to Problem 1.

The set of n locations can be denoted $\mathcal{P}$,

$$\mathcal{P} = \langle \mathbf{p_1}, \ldots, \mathbf{p_n} \rangle, \quad \mathbf{p_i} \in \mathbb{R}^2 \quad \forall i \in \{1, \ldots, n\}. \tag{2.8}$$

The visiting order of locations is a permutation of the locations set Σ,

$$\Sigma = \langle \sigma_1, \ldots, \sigma_n \rangle, \quad \sigma_i \in \{1, \ldots, n\}, \quad \sigma_i \neq \sigma_j \text{ for } i \neq j. \tag{2.9}$$

Because the multi-goal trajectory is defined as a closed-loop, the two locations σ_1 and σ_n are adjacent, which is reflected in the equivalence

$$\sigma_j \triangleq \sigma_{j-n} \text{ for } j > n. \tag{2.10}$$

The best heading angle θ_i for each location $\mathbf{p_i}$ is in the set Θ,

$$\Theta = \langle \theta_1, \ldots, \theta_n \rangle, \quad \theta_i \in [0, 2\pi) \ \forall i \in \{1, \ldots, n\}. \tag{2.11}$$

The configurations $\mathbf{q_i}$ are indexed by the visiting order

$$\mathbf{q_i} = \langle \mathbf{p}_{\sigma_i}, \theta_{\sigma_i} \rangle, \quad i \in \{1, \ldots, n\}. \tag{2.12}$$

The problem of finding the fastest multi-goal trajectory $\hat{\Gamma}$ over a set of specified locations $\mathcal{P}$ is a combined optimization problem, where the minimized variable is the terminal travel time T of the trajectory $\hat{\Gamma}$. The trajectory $\hat{\Gamma}$ consists of n trajectories of Problem 1, and thus, the same motion constraints of the vehicle (2.1–2.7) need to be also fulfilled for $\hat{\Gamma}$. Note that the constraint (2.7) ensures that the subsequent trajectories have equal speed in the connecting location, and the speed on $\hat{\Gamma}$ is continuous.

Problem 2 (Time-optimal trajectory planning over multiple locations)

$$\min_{\Sigma, \Theta, \hat{\Gamma}} \quad T = \sum_{i=1}^{n} T_i,$$

s.t.

$$\hat{\Gamma} = \langle \Gamma_1, \ldots, \Gamma_n \rangle,$$
$$\Gamma_i = [0, T_i] \to SE(2),$$
$$\Gamma_i(0) = \mathbf{q_i}, \quad \Gamma_i(T_i) = \mathbf{q_{i+1}},$$
$$(2.1 - 2.12) \text{ are met.}$$

CHAPTER 3

Proposed Method for Time-Optimal Trajectory Generation

In this chapter, the proposed method of time-optimal trajectory planning for the extended model of Dubins vehicle is described to solve Problem 1. The motivational vehicle of the thesis is a fixed-wing aircraft with variable speed, which allows utilizing a variable turning radius. The vehicle's forward acceleration is limited, and a speed profile is used to compute the time needed to travel the determined trajectory.

The proposed method is motivated by existing approaches introduced in the literature for planning paths between two locations, like the Dubins path [1], the Bézier curves [14], segmented paths utilizing arcs with multiple radii [4, 5, 6] to name few. We consider the variable speed and turning radius, and propose a method, which decreases the travel time of the trajectory while simultaneously, we also aim to develop a computationally efficient solution. A brief overview of existing approaches follows.

The Bézier curves [14] are smooth curves computed numerically between two locations, and they are able to resemble any shape, even similar to the Dubins path [15]. They have been utilized in surveillance scenarios [16] and collision-free planning [17]. However, the minimum turning radius has to be checked in the derivation of the Bézier curve, and the travel time must be calculated numerically, which is computationally demanding.

The trajectory generation using multiple segments with variable radii can produce a wide variety of trajectories. The energy-optimal trajectory for the Dubins vehicle introduced in [5] reaches a lower energy consumption compared to the Dubins path with speed increase on the straight segment. The trajectories created from segments with varying curvature for time-optimal planning are described in [4], but the authors do not consider the bounded acceleration of the vehicle, which is a less realistic scenario. Both approaches take advantage of the variable speed and increased radius, and use a local optimization to find the best trajectory.

Closed-form expression for the shortest path of the Dubins vehicle [1] allows computationally efficient calculation of the Dubins path together with the travel time. Therefore, it has been found a suitable model for the addressed trajectory planning. We propose to combine the variable radii and Dubins path to exploit the benefit of the analytic computation and increased vehicle speed on the arcs with a large radius to decrease the time needed to travel the trajectory. On the other hand, increasing the radius elongates the entire trajectory, and

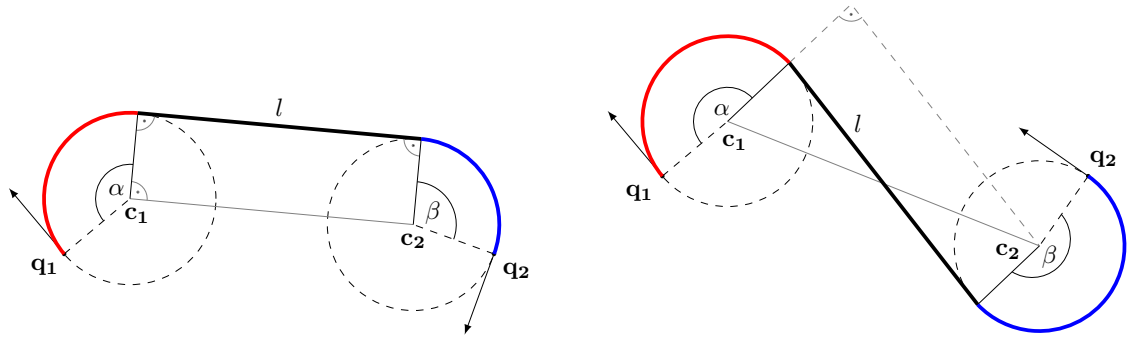
therefore, the trade-off between the trajectory length and possible vehicle speed needs to be found to minimize the travel time. Hence, we need to find a solution to optimization Problem 1, where the trajectory Γ is a variant of the Dubins path consisting of arcs and a straight segment.

The rest of the chapter is organized as follows. In Section 3.1, the Dubins path is described as a background for the reader together with the visual path representation. Section 3.2 shows the proposed trajectory with multiple radii that provides trajectories with decreased travel time compared to the Dubins path. The employed computation of the Travel Time Estimate (TTE) for the proposed three-segment trajectory is presented in Section 3.3, together with the method to compute the TTE for trajectories with an arbitrary number of segments. Finally, we further generalize the idea of the multi-radius Dubins path and propose multi-segment trajectory parametrization in Section 3.4.

3.1 Dubins Path

The problem of planning the shortest curvature-constrained path connecting two locations with the prescribed initial and final orientation of the vehicle was studied in 1957 by Dubins in [1]. The considered vehicle motion model assumes the vehicle is moving forward with a fixed speed and has ability to turn left or right with a limited turning radius. Dubins showed that the optimal path consists only of three segments of two types. Besides, the solution to the optimal path has a closed-form expression.

The first segment type, denoted as C, is an arc with the minimum turning radius $r_{\min}$. The second type is a straight line segment, denoted as S. Two types of the Dubins path are thus CSC or CCC. Furthermore, the C segment type can be distinguished by turn direction: clockwise or right and counter-clockwise or left, marked R and L, respectively. Thus, the total number of possible combinations is six: RSR, RSL, LSL, LSR, LRL, and RLR. A major benefit of the Dubins path is that it can be found analytically using closed-form expression, making it computationally efficient. The computation is described in detail in Section A.1. Example of the CSC path types are visualized in Fig. 3.1 and example of the CCC path type in Fig. 3.2. Due to the necessary connectivity of all three circles for given configurations, the CCC maneuvers are feasible only when the length between circle centers c_1 and c_2 is smaller than or equal to $4r_{\min}$, and the maximum distance between configurations is $6r_{\min}$.



(a) RSR path. **(b)** RSL path.

Figure 3.1: Example of the CSC Dubins paths.

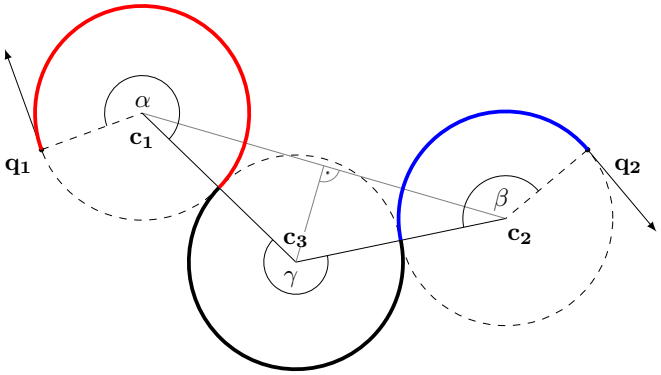
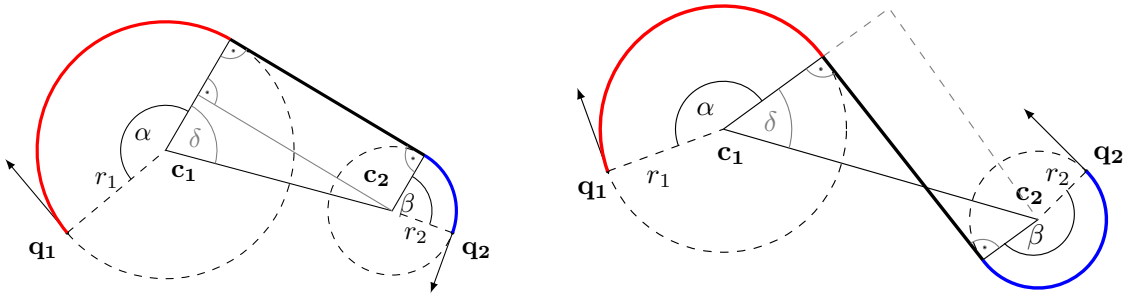


Figure 3.2: Example of the CCC (RLR) Dubins path.

3.2 Generalized Multi-Radius Dubins Path

The proposed trajectory for planning between the two configurations is the extended Dubins path with the individual radius for each arc segment of the CSC path type. The proposed method allows for changing the turning radius of each arc segment independently. For a larger radius, the speed can increase, and thus shorter travel times can be achieved despite the path being longer. The different radii combinations on both arcs need to be examined to find the shortest travel time. In the proposed multi-radius Dubins path extension, we consider only the four CSC path types. The closed-form expression for a particular first and second radius and configurations of the vehicle is presented in Section A.2. Notice that the turning radii of the fastest trajectory can be determined by examining possible combinations of radii, e.g., using discrete samples of the radii or by local optimization of the initial and final radii. Examples of the proposed trajectories are shown in Fig. 3.3.



(a) RSR path type.

(b) RSL path type.

Figure 3.3: Example of the extended Dubins paths with variable turning radii.

The CSC path types are utilized based on empirical evaluation of the CCC path benefits. The CCC paths are optimal solutions only for relatively close configurations, i.e., closer than four times the minimal turning radius, and thus they are optimal solutions less frequently than CSC paths. The ratio between the length of the Dubins paths using both CSC and CCC types to the length of the paths using only the CSC path type is depicted in Table 3.1. The results indicate that the benefit of the CCC path type on the configurations that are distant from $r_{\min}$ up to $6r_{\min}$ is overall 4 %.

Table 3.1: Dubins path length of CCC path type compared with CSC path type for distances between the configurations up to $6r_{\min}$ averaged over 1 000 000 random paths.

Path Type	Distance between $\mathbf{q_1}$ and $\mathbf{q_2}$ / $r_{\min}$											
	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0	5.5	6.0	overall
CSC only	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
CSC and CCC	0.84	0.89	0.95	0.98	0.98	0.98	1.00	1.00	1.00	1.00	1.00	0.96

In the Dubins path computation, we need to evaluate all six possible paths and select the shortest (feasible) one. In the multi-radius Dubins path, we can consider discretization of the turning radii into a finite set of k radii samples, and the number of combinations of the path types becomes $4k^2$ for the CSC paths and $2k^3$ for the CCC paths. For $k \geq 3$, the number of the CCC-based combinations becomes significantly higher than for the CSC path types. Considering the number of combinations and the expected benefit reported in Table 3.1, we decided not to use the CCC path type for the employed multi-radius Dubins paths.

3.3 Travel Time Estimation

The Travel Time Estimation (TTE) is a method for obtaining the expected time to travel the planned trajectory. The travel time is used for evaluating the trajectory cost and determining the fastest trajectory. The TTE is a value of the terminal time in the speed profile, which is suited specifically for a fixed-wing aircraft, but could be customized for other vehicles. The employed model is extended from [4], and we bound the vehicle's acceleration, which makes the model more accurate regarding the estimation of real travel time.

The speed profile is computed with respect to the speed limitations from (2.5–2.6). The maximum speed $v_{\max}$ is limited by the physical properties of the vehicle, and the speed on each segment is also limited by the curvature. Accelerating to $v_{\max}$ is possible on a straight line or an arc with a large turning radius, where the aircraft is able to maintain control and avoid a side slip. The turning radius r can be obtained from the curvature κ as $r = |\kappa^{-1}|$. The maximum speed $\hat{v}$ on the segment with the curvature κ is computed as

$$\hat{v}(\kappa) = \min\left(\sqrt{\frac{g\tan(\varphi_{\max})}{|\kappa|}}, v_{\max}\right), \tag{3.1}$$

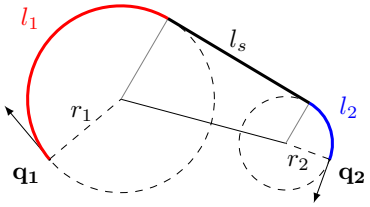
where g is the gravitational acceleration, and $\varphi_{\max}$ is the maximum bank angle of the aircraft. In the trajectory computation, we limit the turning radius of the arcs r between the minimum and maximum value $r_{\min}$ and $r_{\max}$ as

$$r \in [r_{\min}, r_{\max}], \quad r_{\min} = \frac{v_{\min}^2}{g\tan(\varphi_{\max})}, \quad r_{\max} = \frac{v_{\max}^2}{g\tan(\varphi_{\max})}. \tag{3.2}$$

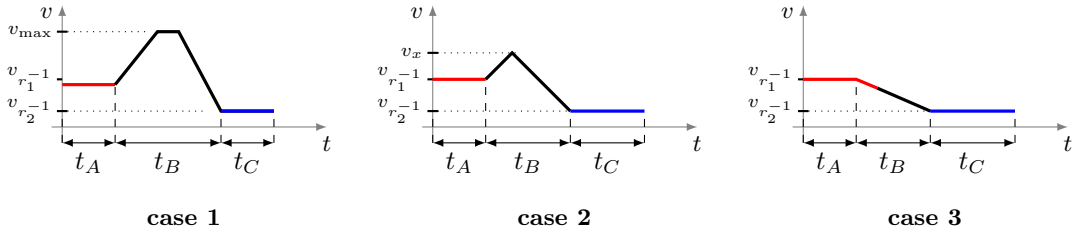
Using radius smaller than $r_{\min}$ is prohibited, as it would require unsafe flight with speed below $v_{\min}$, or with the risk of side-slip. On the other hand, a radius longer than $r_{\max}$ elongates the trajectory but does not allow increasing the speed above $v_{\max}$, and therefore it is not beneficial.

3.3.1 Travel Time Estimation for the Extended Dubins Path

The multi-radius Dubins path introduced in Section 3.2 consists of three segments, with two arcs and one straight segment in the middle. Let the radius of the initial segment be r_1, and the radius of the final segment be r_2. The segment lengths are denoted as l_1 for the initial arc, l_s for the straight segment, and l_2 for the final arc. The speed can change in each segment, but the main speed increase and decrease happen on the straight segment. On the straight segment, the vehicle accelerates towards $v_{\max}$ and decelerates to match the next segment's maximum speed to achieve the shortest possible travel time.



(a) The RSR multi-radius Dubins path.



(b) Three different speed profile cases. The colored parts in the speed profile correspond to the segments of same color in Fig. 3.4a.

Figure 3.4: Speed profile for three-segment Dubins path.

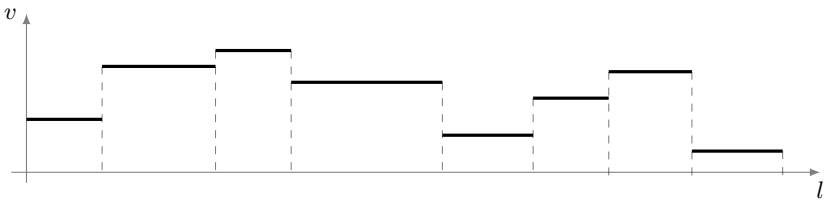
The computation of the speed profile begins with obtaining $\hat{v}(\kappa)$ for each segment, and the possible speed profile cases are shown in Fig. 3.4. The speed in the initial and final configurations is set to $\hat{v}$ for a given radius. In the next step, it is determined if the vehicle is able to accelerate from $\hat{v}(r_1^{-1})$ to $v_{\max}$ and then decelerate to $\hat{v}(r_2^{-1})$ on the straight segment of the length l_s. If so, then the speed profile is constructed as in case 1 in Fig. 3.4b. Otherwise, the length necessary to change the speed from $\hat{v}(r_1^{-1})$ to $\hat{v}(r_2^{-1})$ is calculated, and if the length is less than or equal to l_s, then the maximum reachable speed on the straight segment v_x is determined, and the speed profile is constructed as in the case 2.

If changing the speed from $\hat{v}(r_1^{-1})$ to $\hat{v}(r_2^{-1})$ is not possible on the straight segment, there are two other cases. The first one is that l_s and the length of the segment with the larger radius are sufficient to change the speed from $\hat{v}(r_1^{-1})$ to $\hat{v}(r_2^{-1})$ and the case 3 of the speed profile is constructed that occurs only if $r_1 \neq r_2$. The last case is when the maximum speed on the initial or final configurations cannot be reached concerning the motion constraints. Such a trajectory is either not time-optimal or feasible, and therefore, this case is prohibited. By prohibiting the last case, the initial and final speed matches the maximum possible for the given radius in both configurations. We also disqualify the cases, which are likely not optimal by using an inefficiently large radius.

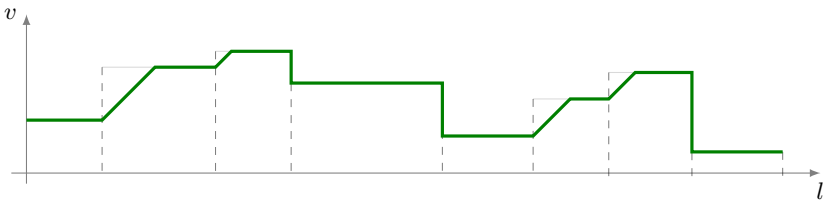
The resulting travel time is the sum of times for individual segments $t_A + t_B + t_C$. The closed-form expression exists for computing the TTE of the multi-radius Dubins path, and it is presented in Section A.3.

3.3.2 Travel Time Estimation using Two-Phase Algorithm

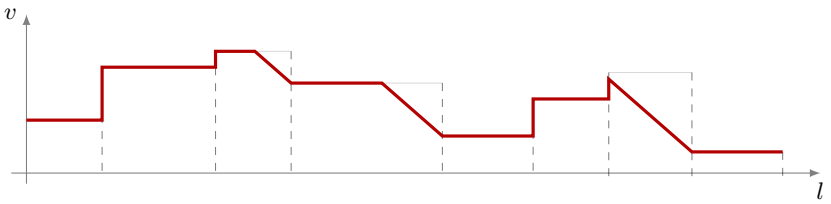
The computation of the TTE can be further generalized for paths with more than three segments. The time is computed using the uniform acceleration in a similar way as for the previous case. Then, the total travel time can be computed using a two-phase algorithm with computational complexity linearly proportional to the number of segments. The full computational example is visualized in Fig. 3.5.



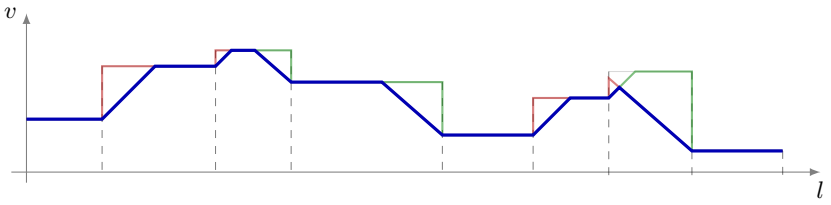
(a) Initialization: for each segment, set the maximum speed by its curvature.



(b) Forward phase: continuous acceleration or sharp speed drop, moving from the first segment to the last.



(c) Backward phase: continuous acceleration or sharp speed drop, moving in the opposite direction from the last segment to the first.



(d) Result: minimum speed from 3.5b and 3.5c at each point.

Figure 3.5: General two-phase algorithm for computing the TTE.

The initial step is the same. The maximum speed is computed on each segment, shown in Fig. 3.5a. After that, the first phase of the two-phase algorithm is the forward phase, which calculates the speed increase using $a_{\max}$ when the speed on the next part is higher than on the current part. If the next maximum speed is lower than the current speed, it is set abruptly to the next speed, e.g., as shown in Fig. 3.5b. The second step mimics the first step, but the vehicle starts at the trajectory end, and the profile is computed as if the vehicle would

move backward. The speed increase is computed using $-a_{\min}$, and the result is visualized in Fig. 3.5c. The resulting speed profile is then obtained as the minimum value in each point from the forward and backward phase; see Fig. 3.5d.

The speed v is not linear with the distance l. The lines are drawn as linear in Fig. 3.5 for intuitive visualization. During the practical computation, the minimum speed value is taken at each position of the trajectory.

3.4 Trajectory with Multiple Segments

The Dubins path with two arc segments with different turning radii can be further generalized by considering multiple segments with increasing/decreasing turning radius that allows the vehicle to accelerate/decelerate, and thus quickly achieve maximal possible speed. This is contrary to the Dubins path, where the vehicle can increase its speed only on the straight segment. The proposed approach replaces the initial and final arc of the Dubins path with multiple segments and optimizes their curvature to achieve a shorter time to traverse the whole trajectory. The construction of the multi-segment trajectory is illustrated in Fig. 3.6.

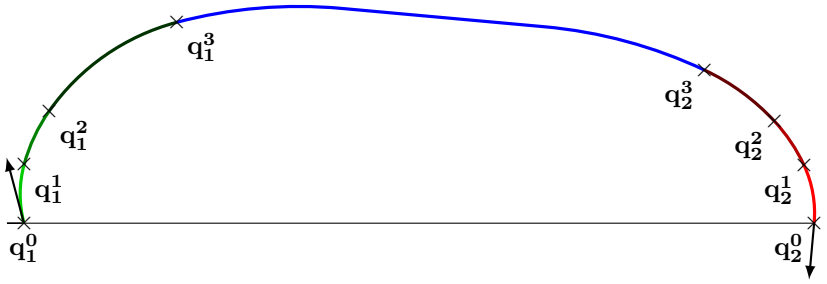


Figure 3.6: Multi-segment trajectory construction with three arcs on each side. The green segments are the initial sequence of arcs, and the red segments are the final sequence of arcs. The blue part is a Dubins path with maximum turning radius.

The multi-segment trajectory is constructed from the Dubins path with the minimum turning radius $r_{\min}$ connecting the given configurations $\mathbf{q_1}$ and $\mathbf{q_2}$. The type of the Dubins path's segments define the curvature's sign of the additional turning segments. Let denote the curvature κ_1 for the initial arc, κ_2 for the final arc, and let m be the number of additional segments on each side. Thus, the total number of segments is $2m + 3$. The set K specifies the curvature, constrained by (3.2), of all segments

$$K = \langle \kappa_1^1, \ldots, \kappa_1^m, \kappa_2^1, \ldots, \kappa_2^m \rangle, \quad |\kappa_1^i|, |\kappa_2^i| \in \left[\frac{1}{r_{\max}}, \frac{1}{r_{\min}} \right] \; \forall i \in \{1, \ldots, m\}. \tag{3.3}$$

The curvature is limited because a larger radius than $r_{\max}$ elongates the path, but a further increase of the speed above $v_{\max}$ is not possible. Similarly, a smaller turning radius than $r_{\min}$ would require flying at the unsafe speed lower than $v_{\min}$. Note that the values $\kappa_1^1, \ldots, \kappa_1^m$ have the same sign as κ_1, whereas the values $\kappa_2^1, \ldots, \kappa_2^m$ have the opposite sign of κ_2, because they are used for arcs computation in the opposite direction.

The turn angles of the arcs are organized into a set Ξ defined as

$$\Xi = \langle \xi_1^1, \ldots, \xi_1^m, \xi_2^1, \ldots, \xi_2^m \rangle, \quad \xi_1^i, \xi_2^i \in \left[0, \frac{2\pi}{m} \right] \; \forall i \in \{1, \ldots, m\}, \tag{3.4}$$

where the first arc starts at the configuration $\mathbf{q_1}$, ends at $\mathbf{q_2}$. Thus, the used notation is that the i-th arc starts at the end of the previous arc $\mathbf{q_j^{i-1}}$ and the whole sequence of arcs ends at the configuration $\mathbf{q_j^i}$, where $\mathbf{q_j^i} = \langle \mathbf{p_j^i}, \theta_j^i \rangle$. For the first sequence of arcs $\mathbf{q_1} = \mathbf{q_1^0}$. However, $\mathbf{q_2^0} = \langle \mathbf{p_2}, \theta_2 + \pi \rangle$ because arcs for $\mathbf{q_2}$ are computed in the opposite direction.

The heading angles of the particular configurations of additional arcs are computed from

$$\theta_j^i = \theta_j^{i-1} + \xi_j^i \mathrm{sgn}(\kappa_j^i), \tag{3.5}$$

where $\mathrm{sgn}(\kappa)$ is the sign of the curvature κ. The position $\mathbf{p_j^i}$ is computed as

$$\mathbf{p_j^i} = \mathbf{p_j^{i-1}} + \frac{1}{\kappa_j^i} \left(\left\langle \sin\theta_j^{i-1}, -\cos\theta_j^{i-1} \right\rangle + \left\langle -\sin\theta_j^i, \cos\theta_j^i \right\rangle \right), \quad \forall i \in \{1, \ldots, m\}, j \in \{1, 2\}. \tag{3.6}$$

Thus, the first sequence of arcs ends at the configuration $\mathbf{q_1^m}$ and the final sequence at $\mathbf{q_2^m}$. The configurations $\mathbf{q_1^m}$ and $\mathbf{q_2^m}$ are connected by the Dubins path with the maximal turning radii $r_{\max}$ to enable the fastest speed. The Dubins path simplifies the computation, as the sequence of arcs can end with an arbitrary heading angle. It is not further optimized, because the maximum speed on all three segments is equal to $v_{\max}$

Based on the proposed trajectory parametrization, we define the time-optimization of the trajectory as Problem 3, where the function $D(\mathbf{q_1^m}, \mathbf{q_2^m})$ returns the length of the Dubins path with $r_{\max}$ between configurations $\mathbf{q_1^m}$ and $\mathbf{q_2^m}$. The time to traverse the trajectory $\mathrm{TTE}(L, V, a_{\min}, a_{\max})$ is computed by the two-phase algorithm from Section 3.3.2, where L and V denote the segments' lengths and segments' speeds, respectively, with the corresponding order of the particular segments. The L and V are the result of the initialization phase for the two-phase algorithm. The speed $\hat{v}(\kappa)$ for the curvature κ is computed using (3.1).

Problem 3 (Time-optimal planning for a path with multiple segments)

$$\begin{aligned}
&\min_{K,\Xi} T, \\
&\text{s.t.} \\
&\quad l = D(\mathbf{q_1^m}, \mathbf{q_2^m}), \\
&\quad L = \left\langle \frac{\xi_1^1}{|\kappa_1^1|}, \ldots, \frac{\xi_1^k}{|\kappa_1^k|}, l, \frac{\xi_2^k}{|\kappa_2^k|}, \ldots, \frac{\xi_2^1}{|\kappa_2^1|} \right\rangle, \\
&\quad V = \left\langle v_{\kappa_1^1}, \ldots, v_{\kappa_1^k}, v_{\max}, v_{\kappa_2^k}, \ldots, v_{\kappa_2^1} \right\rangle, \\
&\quad T = \mathrm{TTE}(L, V, a_{\min}, a_{\max}).
\end{aligned}$$

3.4.1 Planning Trajectory with Multiple Segments

The optimization problem proposed in this section is solved by a local optimization to find the lengths and radii of the trajectory's segments. The optimization function accepts an initial vector $\langle K, \Xi \rangle$, and the lower and upper bounds determined as the minimum and maximum values from the limits of K and Ξ. The vector for the optimization is thus $4m$ long, which is very challenging to optimize for large values of m. Therefore, reported results on the planning trajectory with multiple segments are presented only for relatively small m.

A proper initialization is essential for the local optimization methods that can find only a local optimum. The initial Dubins path is employed for the initialization, which is computationally efficient and provides a suitable solution being improved. The arc angles $\xi_1^1, \ldots, \xi_1^m$ are

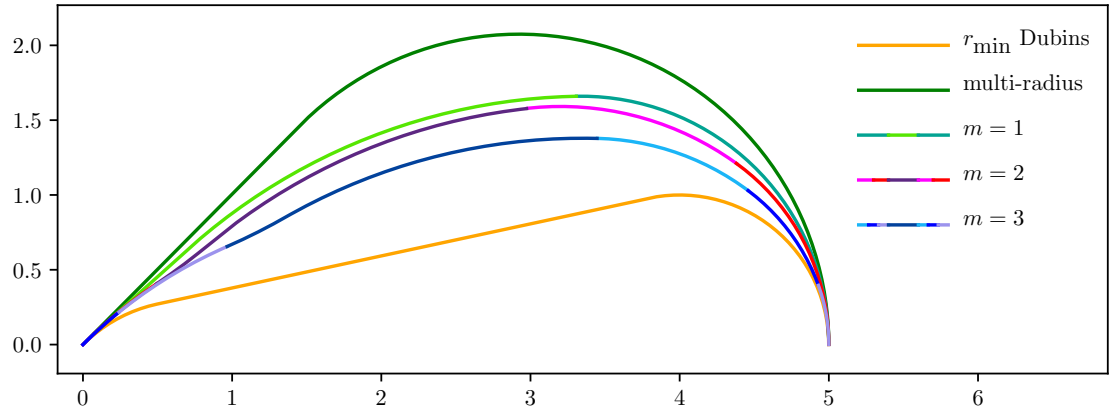


Figure 3.7: Example of proposed trajectories with multiple segments, $\theta_1 = \frac{\pi}{4}, \theta_2 = \frac{3\pi}{2}$. The orange trajectory is the initial Dubins path; the green trajectory is extended multi-radius Dubins path with optimal radii found using local optimization.

initialized from the angle of the initial segment of the Dubins path that is divided into m segments. Similarly, the initial values of the angles of the arcs $\xi_2^1, \ldots, \xi_2^m$ are determined from the final Dubins segment. The values of curvature $\kappa_1^1, \ldots, \kappa_1^m$ are initialized as $\mathrm{sgn}(\kappa_1) r_{\min}^{-1}$, and $\kappa_2^1, \ldots, \kappa_2^m$ with $-\mathrm{sgn}(\kappa_2) r_{\min}^{-1}$. Example paths are shown in Fig. 3.7 and corresponding improvements on the trajectory time are depicted in Table 3.2. Note that only increasing

Table 3.2: Travel and execution times of the multi-segment trajectories in comparison to Dubins path.

Trajectory type	**Travel Time**	**Improvement [%] vs.** Dubins $r_{\min}$	Multi-radius	T_{cpu} [s]
Dubins path with $r_{\min}$	1.31	0.0	−14.2	<0.01
Multi-radius Dubins path & Optim	1.15	12.4	0.0	0.05
Multi-segment trajectory, $m = 1$	1.11	15.4	3.4	0.49
Multi-segment trajectory, $m = 2$	1.07	18.3	6.6	2.68
Multi-segment trajectory, $m = 3$	1.07	18.3	6.7	1.55
Multi-segment trajectory, $m = 4$	1.07	18.5	7.0	4.39
Multi-segment trajectory, $m = 5$	1.06	19.4	8.0	5.15
Multi-segment trajectory, $m = 6$	1.05	19.8	8.5	10.45
Multi-segment trajectory, $m = 7$	1.07	18.5	7.0	12.60
Multi-segment trajectory, $m = 8$	1.07	18.1	6.5	28.13
Multi-segment trajectory, $m = 9$	1.12	14.7	2.6	12.05
Multi-segment trajectory, $m = 10$	1.09	17.0	5.3	33.27

The reported results are mean values over 30 runs using the computational environment as reported in Chapter 4

the number of segments does not necessarily lead to decreasing the travel time. It is because of the increased number of variables being optimized, for which the local optimization stuck in a local extreme.

CHAPTER 4

Evaluation of the Proposed Time-Optimal Trajectory Planning

In this chapter, the proposed solutions to time-optimal trajectory planning based on the multi-radius Dubins path and multi-segment trajectory are examined and compared to the existing methods. The evaluations are designed to show how the travel time is decreased by utilizing higher turning radii than the minimal. Besides, we studied the influence of different parameters on the resulting trajectories and reported the results. The trajectories also depend on the used vehicle, and for this thesis, we consider Cessna 172 aircraft [18], which is mostly used in general aviation. The used parameters are summarized in Table 4.1, where the minimal turning radius for selected speed has been computed using equation (2.6). The minimum speed $v_{\min}$, which is the maximum reachable speed on the minimum turning

Table 4.1: Values of the Cessna 172 aircraft for trajectory computation.

Parameter	Symbol	Value
Stall speed	v_{stall}	$27.0\,\text{m}\,\text{s}^{-1}$
Minimal velocity	$v_{\min}$	$30.0\,\text{m}\,\text{s}^{-1}$
Maximal velocity	$v_{\max}$	$67.0\,\text{m}\,\text{s}^{-1}$
Maximal bank angle	$\varphi_{\max}$	$60.0\,^{\circ}$
Min. turning radius for $v_{\min}$ and $\varphi_{\max}$	$r_{\min}$	$65.7\,\text{m}$
Min. turning radius for $v_{\max}$ and $\varphi_{\max}$	$r_{\max}$	$264.2\,\text{m}$
Minimal acceleration	$a_{\min}$	$-3.0\,\text{m}\,\text{s}^{-2}$
Maximal acceleration	$a_{\max}$	$2.0\,\text{m}\,\text{s}^{-2}$

radius, has been selected higher than the stall speed v_{stall} of the aircraft to ensure a safe flight. The maximal speed $v_{\max}$ is determined by the maximum power output of the vehicle.

All the proposed solutions have been implemented in Julia language, version 1.2, and were executed on a personal computer with Intel CPU i7-8550U @ up to 4.0 GHz. A library for time-optimal trajectories with unbounded acceleration [4] written in C++ was also used, and the generated trajectories are parsed by Julia.

The rest of the chapter is organized as follows. The first section is dedicated to the proposed multi-radius Dubins path, which has also been published in [7]. The multi-segment trajectory is studied in Section 4.2. The results compared to the approach Wolek et al. [4] are reported in Section 4.3.

4.1 Evaluation of the Proposed Multi-Radius Dubins Path

The benefits of the proposed multi-radius Dubins path are evaluated and compared to the Dubins path with the minimal turning radius [1]. The focus of the examination is to find the influence of the radii on the travel time with random initial and final configurations. Two solution methods for finding a multi-radius Dubins path are examined. The first is a local optimization of the radii on each arc, initialized from the Dubins path with the minimum turning radius. The second method is sampling with different sample distributions.

The evaluation is divided into three parts. The influence of the turning radii on the trajectory travel time is studied in Section 4.1.1. In Section 4.1.2, we reported on the utilization of various radii and the achieved shortening of the trajectory travel time. Different radii distributions and numbers of radii samples are compared with the local optimization approach in Section 4.1.3.

4.1.1 Influence of the Turning Radii on the TTE

First, we examine the TTE on the initial and final radii for the fixed leaving and arriving angles, and distance between the configurations. The radii are uniformly sampled with a small step, and the travel time is computed between the configurations for each radii combination. The determined trajectories are compared to the shortest Dubins path with the minimum turning radius and speed increase in the middle segment.

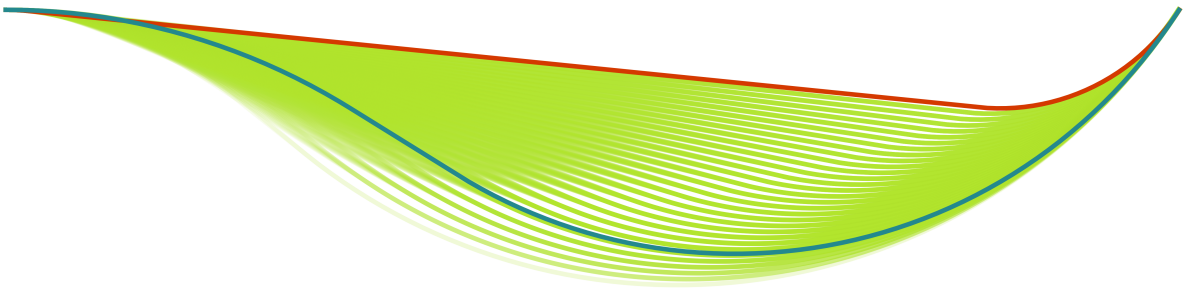


Figure 4.1: Possible trajectories for various turning radii, distance between the configurations $5.5\, r_{\min}$, $\theta_1 = 0$, $\theta_2 = 1$. The red trajectory shows the shortest Dubins path, the green trajectories are candidate trajectories, and the light blue trajectory is the fastest trajectory from all candidates.

Few candidate trajectories that are faster than the shortest trajectory between two configurations are shown in Fig. 4.1. In Fig. 4.2, the gradient of speed-up on trajectories with different initial and final radii (the XY axis) is shown. The speed-up is the ratio between the TTE of the shortest trajectory (Dubins path) and the TTE of the newly proposed trajectory with two turning radii. The white dots in Fig. 4.2 represent selected candidate trajectories shown in Fig. 4.1. The change of the path type causes a sharp change in the gradient. The gradient is continuous if the neighboring maneuver type is the same. However, when then the path type changes, the value of the travel time may change abruptly. The Fig. 4.2 shows

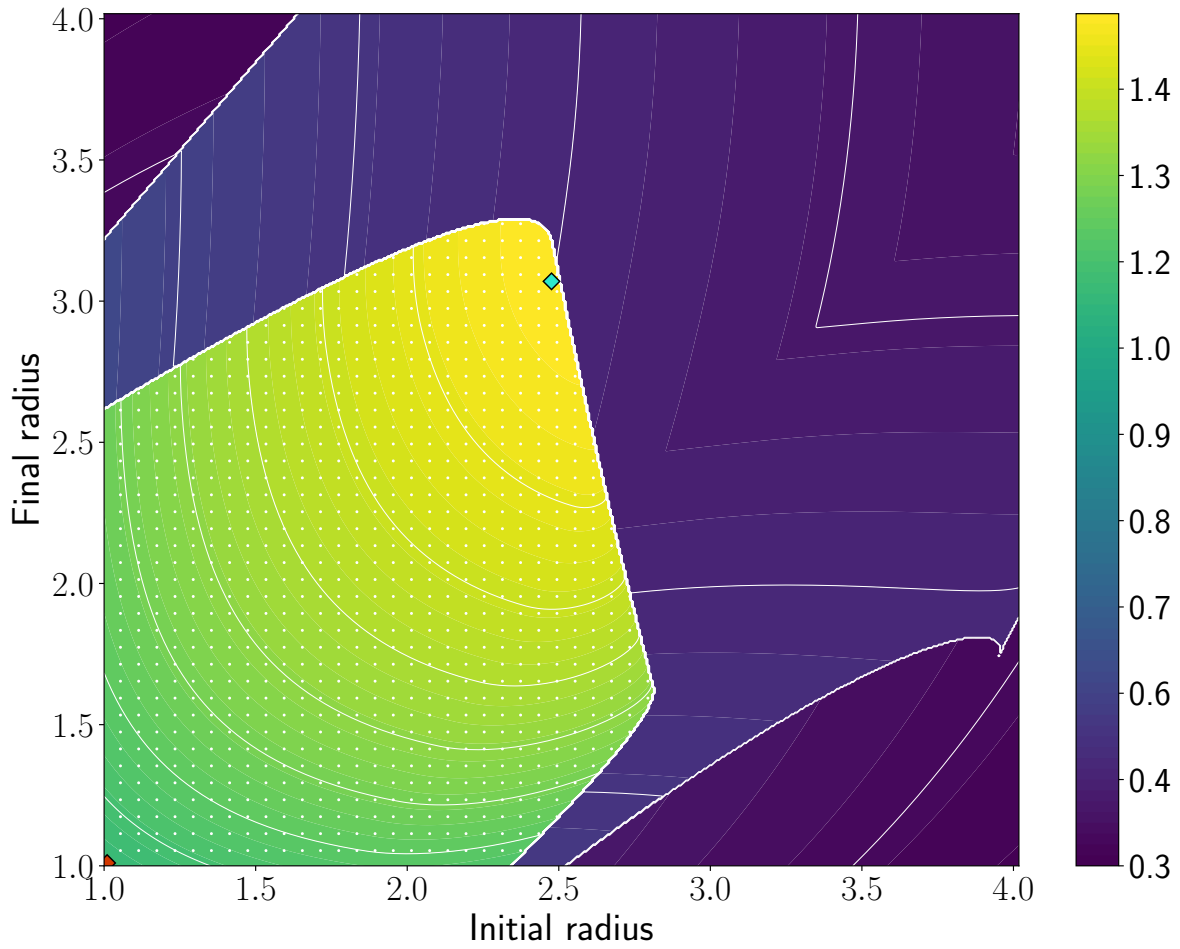


Figure 4.2: Influence of the initial/final turning radius on the TTE for the distance between the configurations $5.5\,r_{\min}$, $\theta_1 = 0$, $\theta_2 = 1$. Color scale represents the speed-up with respect to the trajectory with the minimum turning radius (speed-up = 1). The red and light blue diamonds correspond to the shortest trajectory and the fastest trajectory.

that the turning radii in the continuous part of the gradient can be easily locally optimized. However, the optimization might get stuck in a local optimum. If the value of the speed-up is smaller than one after the path type change, it means that the turning angles α or β are larger than π, because the radius is inefficiently large. The trajectory becomes too long to be compensated by the increased speed on a larger radius, which shows the importance of selecting suitable radii for particular configurations.

4.1.2 Influence of Distance Between the Configurations

Next, we examine the influence of the distance between the configurations on the radii usage. The distance is selected from an interval from one to forty times the minimum turning radius. For each distance, 500 random instances are generated, and the percentile of radii usage and speed-up improvement is computed. An example of the best multi-radius Dubins path with fixed distance and the same heading angles is shown in Fig. 4.3. Notice that the trajectories with a larger distance between configurations have a longer straight segment.

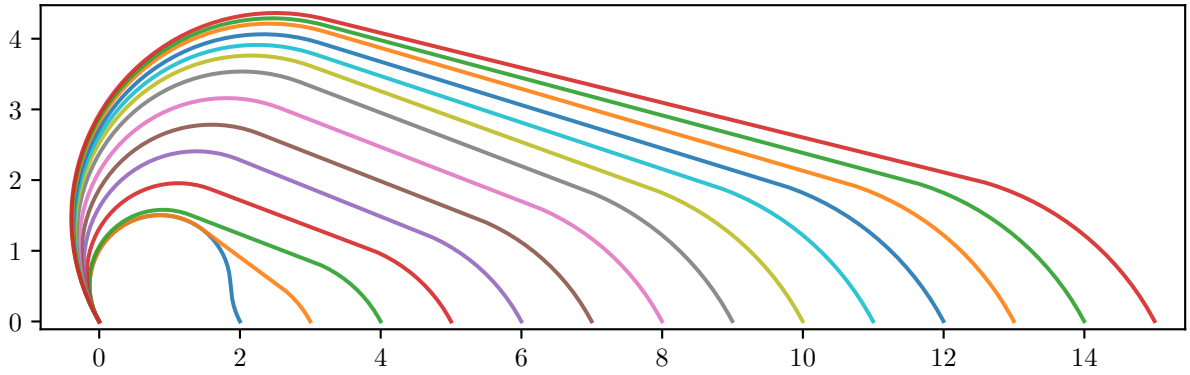
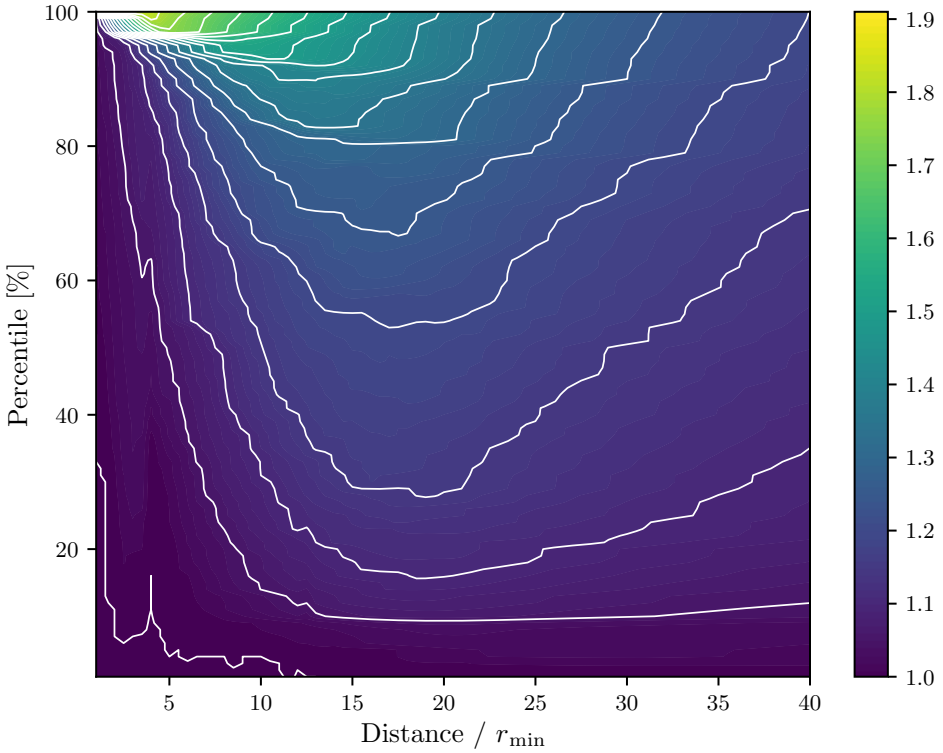


Figure 4.3: Trajectories with fixed initial and final heading angles and different distances.

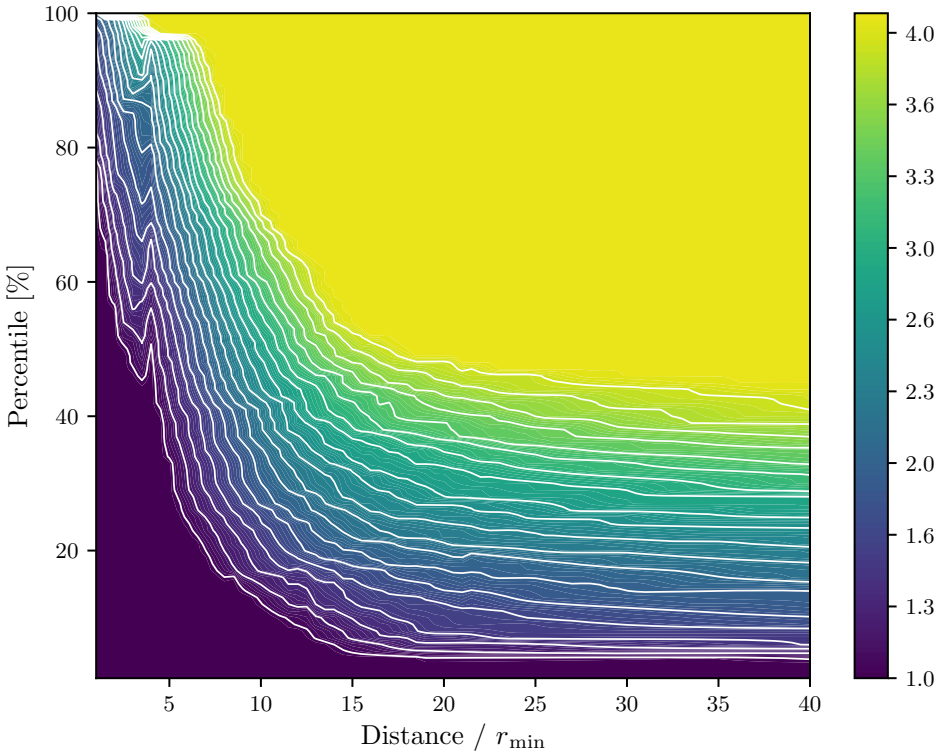
In Fig. 4.4, the percentile indicates how often the radius is used, and a certain speed-up achieved. The trajectory speed-up is compared to the shortest Dubins path with the speed increase on the straight segment. We used an optimization framework called Optim [19] with the L-BFGS [20] optimization method to find the best turning radii for minimizing the TTE of the trajectory.

The resulting percentiles of the speed-up are shown in Fig. 4.4a. The maximum value is reached approximately at a distance between two configurations four times the minimum turning radius r_{min} apart. On a larger distance, the maximum speed-up of the multi-radius Dubins path is smaller because the length of straight segment allows to reach the maximum vehicle speed for both the shortest and fastest trajectory. The radii usage based on the distance is shown in Fig. 4.4b. For short distances, the minimum turning radius is used most often, as a large turning radius increases the trajectory length disproportionately to the possible speed-up. Short trajectories usually do not have a long straight segment, which allows the highest speed increase. However, for long distances, the maximum turning radius is used in the majority of cases. The maximum radius is used in more than 50 % of cases, and usage of the minimum radius drops below 5 %. The maximum vehicle speed can be utilized on the arc with the maximum radius, and the trajectory length increase by using a larger radius is perceptually not that significant.

The maximum speed-up achieved by optimizing the turning radii is above 1.9. Hence, the reported results show how selecting the optimal value of turning radii varies based on the distance between the configurations, and why using the full range of possible turning radii can be beneficial.



(a) Percentile of the speed-up. Color bar shows time improvement with respect to the shortest trajectory with minimum turning radius.



(b) Percentile of turning maneuver usage. Color bar shows turning radii, with minimum radius normalized to one.

Figure 4.4: Influence of the distance between initial and final configuration on the overall speed-up and turning radius usage.

4.1.3 Radii Distribution

The most suitable radii are found by the optimization framework Optim [19] in the results reported in the previous section. However, the local optimization can fall in a local optimum. Besides, convergence time depends on the instance. Thus, we compare the computational time of the local optimization with the approach based on the explicit discretization of the turning radii. Two sampling schemata are considered: linear and exponential. The linear sampling uniformly distributes the radii between the minimum radius $r_{\min}$ and maximum radius $r_{\max}$. The number of samples is denoted k, and the i-th sample $s_i, i \in \{1, \ldots, k\}$ is computed as

$$s_i = r_{\min} + (i-1) \cdot \frac{r_{\max} - r_{\min}}{k-1}. \tag{4.1}$$

In the exponential sampling, the samples interval is also between $r_{\min}$ and $r_{\max}$, but the radii are more densely sampled near $r_{\min}$. The i-th sample $s_i, i \in \{1, \ldots, n\}$ is computed as

$$s_i = \exp\left[\ln r_{\min} + (i-1) \cdot \frac{\ln r_{\max} - \ln r_{\min}}{k-1}\right]. \tag{4.2}$$

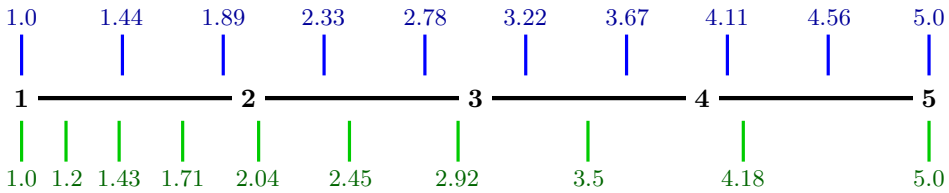


Figure 4.5: Distributions of ten samples between $r_{\min} = 1, r_{\max} = 5$. The blue values of radii are sampled uniformly, and the green exponentially.

All the combinations of samples are examined to select the fastest multi-radius Dubins path. The number of combinations is $4k^2$ because of four CSC path types. Two utilized sampling distributions are visualized in Fig. 4.5 on an interval between $r_{\min}, r_{\max}$, where $r_{\min} = 1$, $r_{\max} = 5$, and $k = 10$. The blue marks represent values of the linear radii samples computed by (4.1), and the green are exponential samples (4.2).

The speed-up values averaged over 500 iterations are shown in Fig. 4.6. The exponential sampling shows to be more beneficial than the linear sampling mainly for small distances, due to the dense sample distribution near $r_{\min}$. The difference between the distributions is most visible when using a small number of samples. On ten radii samples, the average speed-up is approximately the same in both distributions and similar to the speed-up achieved with the Optim framework. The value of speed-up computed using Optim is higher because the optimization function can change the radius more precisely. However, the local optimization is vulnerable to falling into a local optimum, and that is why the sampling-based approach provides better results for large distances. The average speed-up at distance $6r_{\min}$ is about 10 %, and about 20 % on distance $15r_{\min}$, which is shown in Fig. B.1.

The computation times are shown in Fig. 4.7. The results suggest that a relatively small number of radii samples helps to increase the speed-up competitively with the numeric optimization using the Optim framework. However, the computational time of discrete radii samples is more than three orders of magnitude shorter than for the optimization method. The average computational time of multi-radius Dubins path with the corresponding TTE

is 2.6 ms for ten radii samples on average. The same computation using optimization with Optim framework is about 1063.5 ms on average because a large number of candidate trajectories (thousands) are examined before converging.

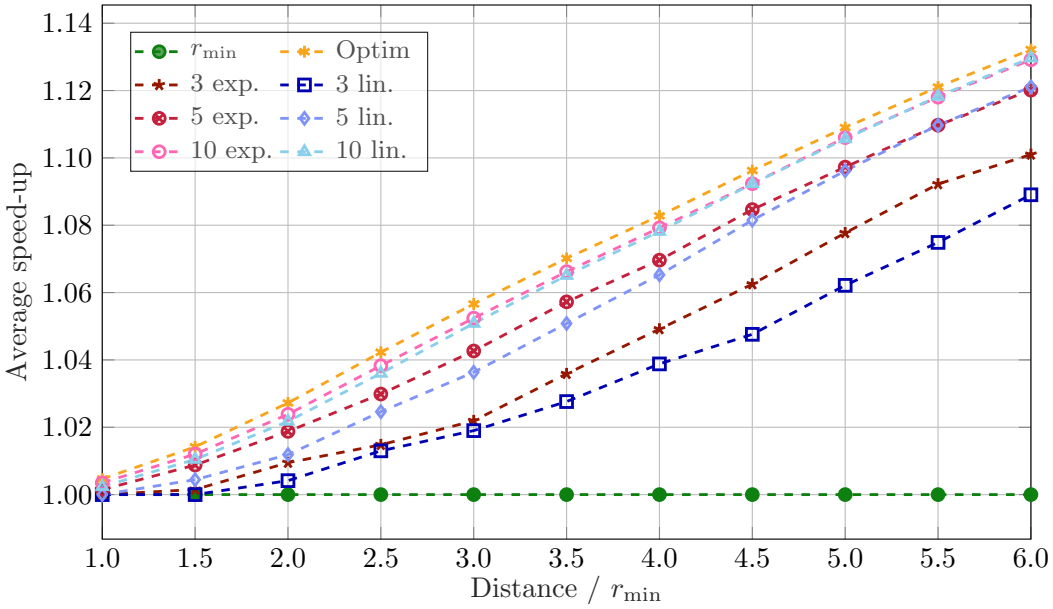


Figure 4.6: Average speed-up of TTE based on the distance between initial and final configuration. All results are averaged over 100 randomly generated instances.

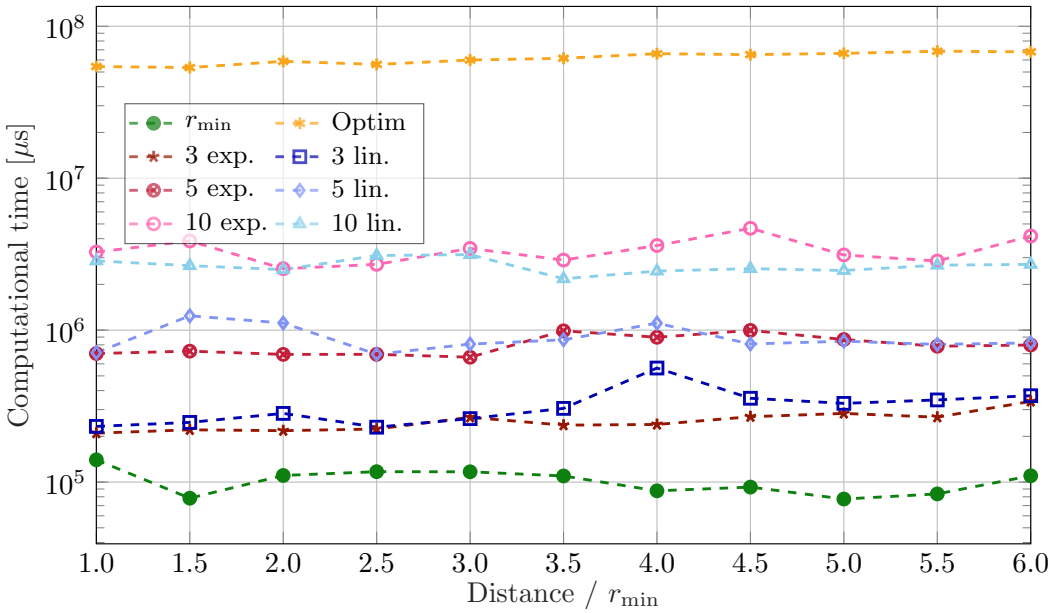


Figure 4.7: Computational time based on the distance between initial and final configuration up to distance $6r_{\min}$. All results are averaged over 100 randomly generated instances.

4.2 Examination of the Multi-Segment Trajectory

The multi-segment trajectory proposed in Section 3.4 is made of two sequences of segments (arcs), and the sequences are connected by the Dubins path with the maximum turning radius. The arc lengths and radii have to be determined to construct the trajectory. For a single segment on each side, the number of parameters is equal to four, and the discretization would result in large amount of combinations. Therefore, we decided to compute the trajectory by the local optimization. Thus, the trajectory is initialized as the Dubins path with the minimum turning radius. The turning radii of all segments are set to $r_{\min}$, and segments length are set to the length of the corresponding turn of the Dubins path, divided by the number of segments.

The multi-segment trajectory generation is examined regarding the influence of the number of segments to the required computational time, and the results are reported in Section 4.2.1. Then, we examine the initialization of the multi-segment trajectory using the multi-radii Dubins path in Section 4.2.2.

4.2.1 Computational Requirements of Multi-Segment Trajectory

We compare the multi-segment trajectory to the Dubins path with the minimum turning radius, and the multi-radius Dubins path. The multi-segment trajectory is computed using local optimization of the Optim framework [19] for test instances with the fixed distance and randomly generated initial and final configurations. The results for the number of segments in the range $\langle 1, \ldots, 10 \rangle$ are depicted in Table 4.2. The multi-radius Dubins paths are also

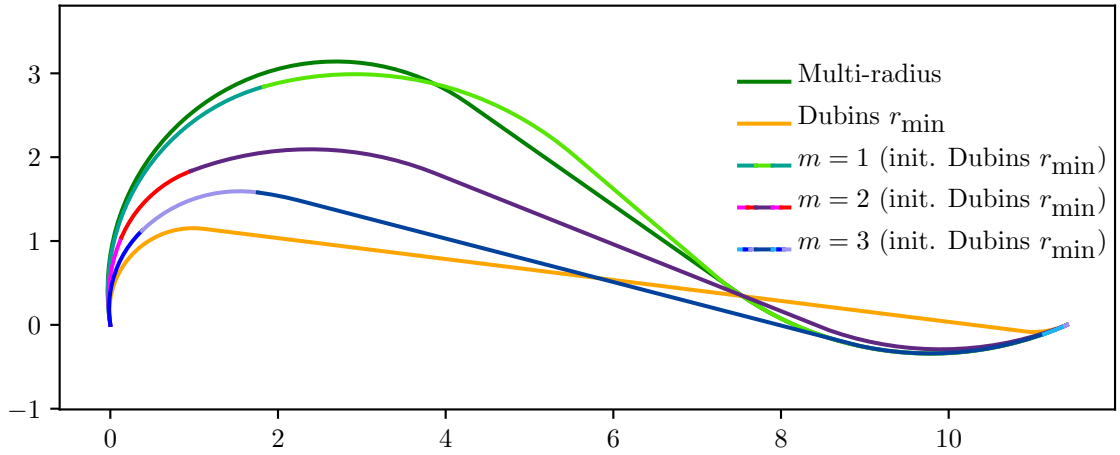
Table 4.2: Multi-segment trajectory in comparison to the original and extended Dubins path. Without optimization limits, 10 random instances, distance $5r_{\min}$ between configurations.

Trajectory type	**Travel Time**	**Improvement [%] vs.** Dubins $r_{\min}$	Multi-radius	T_{cpu} [s]
Dubins $r_{\min}$	1.85	0.00	-5.42	<0.01
Multi-radius	1.75	5.14	0.00	0.06
Multi-seg. $m = 1$	1.75	6.46	2.11	0.42
Multi-seg. $m = 2$	1.70	9.07	4.60	1.76
Multi-seg. $m = 3$	1.70	9.47	5.19	2.14
Multi-seg. $m = 4$	1.72	7.63	1.92	15.38
Multi-seg. $m = 5$	1.71	8.21	2.46	14.36
Multi-seg. $m = 6$	1.72	7.44	1.80	42.35
Multi-seg. $m = 7$	1.71	8.03	2.23	82.07
Multi-seg. $m = 8$	1.73	6.95	1.09	184.17
Multi-seg. $m = 9$	1.73	6.43	0.53	373.66
Multi-seg. $m = 10$	1.74	5.94	-0.01	355.06

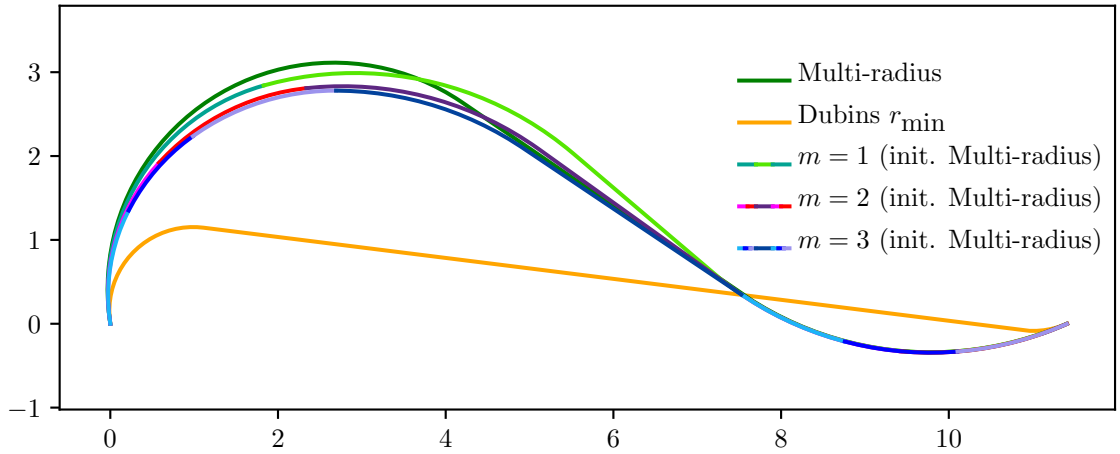
determined using local optimization. It is not surprising that the computational time is increased with an increasing number of segments. However, the final trajectory does not have major time improvement for a high number of segments. The highest speed-up is achieved for three segments, and is further elaborated for computational time limited to 120 s in the next section.

4.2.2 Influence of Initialization to Multi-Segment Trajectories

The multi-segment trajectory generation is proposed with the initialization using the Dubins path with the minimum turning radius in Section 3.4 because the Dubins path is obtained quickly. However, the multi-radius Dubins path is less demanding than a multi-segment trajectory. Therefore, we decided to use a multi-radius Dubins path for the initialization to see the influence on the travel time. The resulting trajectories are shown in Fig. 4.8.



(a) Trajectory initialized from Dubins path with $r_{\min}$, most significant improvement on $m = 1$: 21.3 % vs. Dubins path, 2.2 % vs. multi-radius Dubins path. For $m = 3$, the improvement is −3.9 % vs. multi-radius.



(b) Trajectory initialized from multi-radius Dubins path, most significant improvement on $m = 3$: 23.2 % vs. Dubins path, 4.5 % vs. multi-radius Dubins path.

Figure 4.8: Example of multi-segment trajectories with different initialization, distance between configurations $11.41 r_{\min}$, $\alpha = 1.72, \beta = 0.42$.

Performance indicators from 100 random configurations with distance $10 r_{\min}$ are depicted in Table 4.3. The trajectories initialized by the multi-radius Dubins path have almost the same travel time as the trajectories initialized by the Dubins path with $r_{\min}$ up to $m = 3$. For $m \geq 5$, the travel time is shorter when the trajectory is initialized by the multi-radius Dubins path; moreover, the computational time is decreased. The initial value of the travel time is closer to a local optimum, which leads to faster convergence of the local optimization function on average.

Table 4.3: Multi-segment trajectory for 100 random configurations $10r_{\min}$ apart with Dubins path and multi-radius Dubins path initialization.

(a) Initialization with Dubins Path with $r_{\min}$.

Trajectory type	**Travel Time**	**Improvement [%] vs.** Dubins $r_{\min}$	Multi-radius	T_{cpu} [s]
Dubins $r_{\min}$	2.76	0.00	-6.15	<0.01
Multi-radius	2.60	5.79	0.00	0.06
Multi-seg. $m = 1$	2.55	8.54	2.12	0.32
Multi-seg. $m = 2$	2.47	11.42	5.12	0.82
Multi-seg. $m = 3$	2.43	12.54	6.29	2.07
Multi-seg. $m = 4$	2.42	13.02	6.74	5.37
Multi-seg. $m = 5$	2.46	11.21	4.36	19.64
Multi-seg. $m = 6$	2.46	10.94	4.00	17.59
Multi-seg. $m = 7$	2.50	9.84	2.86	74.15
Multi-seg. $m = 8$	2.45	11.47	4.77	67.27

(b) Initialization with multi-radius Dubins Path.

Trajectory type	**Travel Time**	**Improvement [%] vs.** Dubins $r_{\min}$	Multi-radius	T_{cpu} [s]
Dubins $r_{\min}$	2.76	0.00	-6.15	<0.01
Multi-radius	2.60	5.79	0.00	0.06
Multi-seg. $m = 1$	2.55	8.50	2.07	0.34
Multi-seg. $m = 2$	2.47	11.38	5.07	1.68
Multi-seg. $m = 3$	2.43	12.58	6.33	3.97
Multi-seg. $m = 4$	2.44	12.31	6.07	15.00
Multi-seg. $m = 5$	2.42	13.11	6.91	24.65
Multi-seg. $m = 6$	2.42	13.10	6.90	48.34
Multi-seg. $m = 7$	2.43	12.96	6.77	53.95
Multi-seg. $m = 8$	2.42	13.01	6.82	42.79

4.3 Comparison of Proposed Trajectories to Time-optimal Trajectories with Unbounded Acceleration

In this section, we compare the proposed trajectories with the time-optimal path planning presented by Wolek et al. [4]. The trajectories utilize multiple segments, variable radii and speed; however, the authors do not consider limited acceleration of the vehicle. Thus, the travel time must be computed with the limited acceleration to allow comparison, for which we used the two-phase algorithm from Section 3.3.2. The trajectories [4] consist of turn denoted $\mathcal{T}$ and straight segments $\mathcal{S}$. The turns $\mathcal{T}$ further consist of three arcs $\mathcal{BCB}$: bang arc $\mathcal{B}$ has the radius $r_{\max}$ and the cornering arc $\mathcal{C}$ has the radius $r_{\min} < r_{\max}$. The segments $\mathcal{S}$, $\mathcal{B}$, and $\mathcal{C}$ have length ≥ 0. The segments' length can be computed from the turning angles α, β, γ of the turn $\mathcal{BCB}$, and from the length $\mathcal{L}$ of the straight segment $\mathcal{S}$. The possible trajectory types are $\mathcal{TT}$, $\mathcal{TST}$, $\mathcal{TTT}$, and $\mathcal{TTTT}$.

The minimum speed is denoted $v_{\min}$, and the maximum speed $v_{\max}$ is equal to 1. An example of feasible trajectories between two configurations is shown in Fig. 4.9.

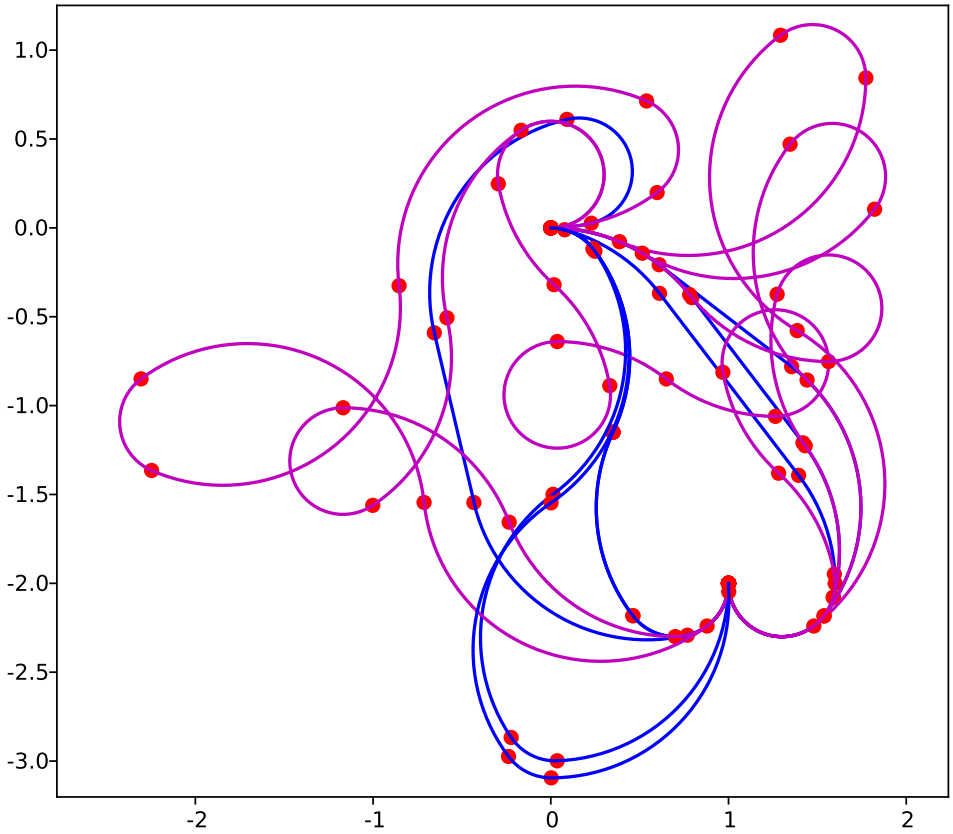


Figure 4.9: The feasible trajectories proposed by Wolek et al. [4] between configurations $\mathbf{q_1} = \langle 0, 0, 0 \rangle$, $\mathbf{q_2} = \langle 3, 0, 1.57 \rangle$ and maximum turning radius scaled to 1.

Table 4.4: Scaled values for trajectory computation with Wolek et al. library.

Parameter	Symbol	Value
Minimal velocity	$v_{\min}$	$0.54\,\mathrm{m\,s^{-1}}$
Maximal velocity	$v_{\max}$	$1.00\,\mathrm{m\,s^{-1}}$
Min. turning radius for $v_{\min}$ and $\varphi_{\max}$	$r_{\min}$	$0.30\,\mathrm{m}$
Min. turning radius for $v_{\max}$ and $\varphi_{\max}$	$r_{\max}$	$1.00\,\mathrm{m}$
Minimal acceleration	$a_{\min}$	$-0.73\,\mathrm{m\,s^{-2}}$
Maximal acceleration	$a_{\max}$	$0.49\,\mathrm{m\,s^{-2}}$

In the available library of [4], the initial configuration is fixed to $\langle 0, 0, 0 \rangle$, and the turning radii to $r_{\min} = 0.3, r_{\max} = 1$. The library does not process different initial heading angle, and for different values of $r_{\min}$ and $r_{\max}$, trajectories did not finish in the defined final configuration. Thus, we did not use the Cessna parameters to compare the proposed trajectories with Wolek et al. [4], as it could require more substantial modifications of the available source codes. The trajectories are therefore examined with the vehicle model according to Table 4.4

and the trajectory travel time is computed using the TTE for multiple segments, see Section 3.3.2. The length of the turn segments is computed from the values of α, β, γ with the corresponding turning radius. The straight segment and bang arcs have maximum speed $v_{\max}$, and the cornering arc $v_{\min}$.

Table 4.5: Comparison of the proposed trajectories with existing approaches over 100 random instances. The Multi-radius Dubins path is computed from 35 radii samples.

Trajectory type	**Travel time**				T_{cpu}
	Dist. 2	**Dist. 3**	**Dist. 5**	**Dist. 8**	**[ms]**
Dubins $r_{\min}$	3.87	4.54	6.50	9.63	0.44
Dubins $r_{\max}$	6.75	6.13	6.85	10.00	0.52
Multi-radius	3.75	4.38	6.33	9.47	86.23
Multi-seg. $m = 1$	3.55	4.20	6.14	9.28	607.66
Multi-seg. $m = 2$	3.49	4.15	6.10	9.23	1549.03
Multi-seg. $m = 3$	**3.48**	**4.14**	**6.09**	**9.22**	6177.64
Wolek et. al. [4]	4.08	5.14	7.41	10.86	2192.31

The travel time comparison of all trajectories is shown in Table 4.5. All trajectories in one picture are shown in Fig. 4.10. The shortest trajectories are provided by the proposed multi-radius Dubins path and multi-segment trajectories, which are more demanding for $m = 3$ than the approach by Wolek et al. [4]. In all the evaluation described in this chapter, the trajectories proposed in the thesis decreased the travel time in comparison to the Dubins path with $r_{\min}$. Moreover, the multi-radius Dubins path is computationally efficient. The multi-segment trajectories are able to further decrease the travel time than the multi-radius Dubins path, however, the computation requires a longer time for optimization before convergence. The tests suggest that the trajectories by Wolek et al. have longer travel time than the Dubins path with $r_{\min}$, when using the bounded acceleration. This result was expected, considering that the trajectories were optimized with unbounded acceleration.

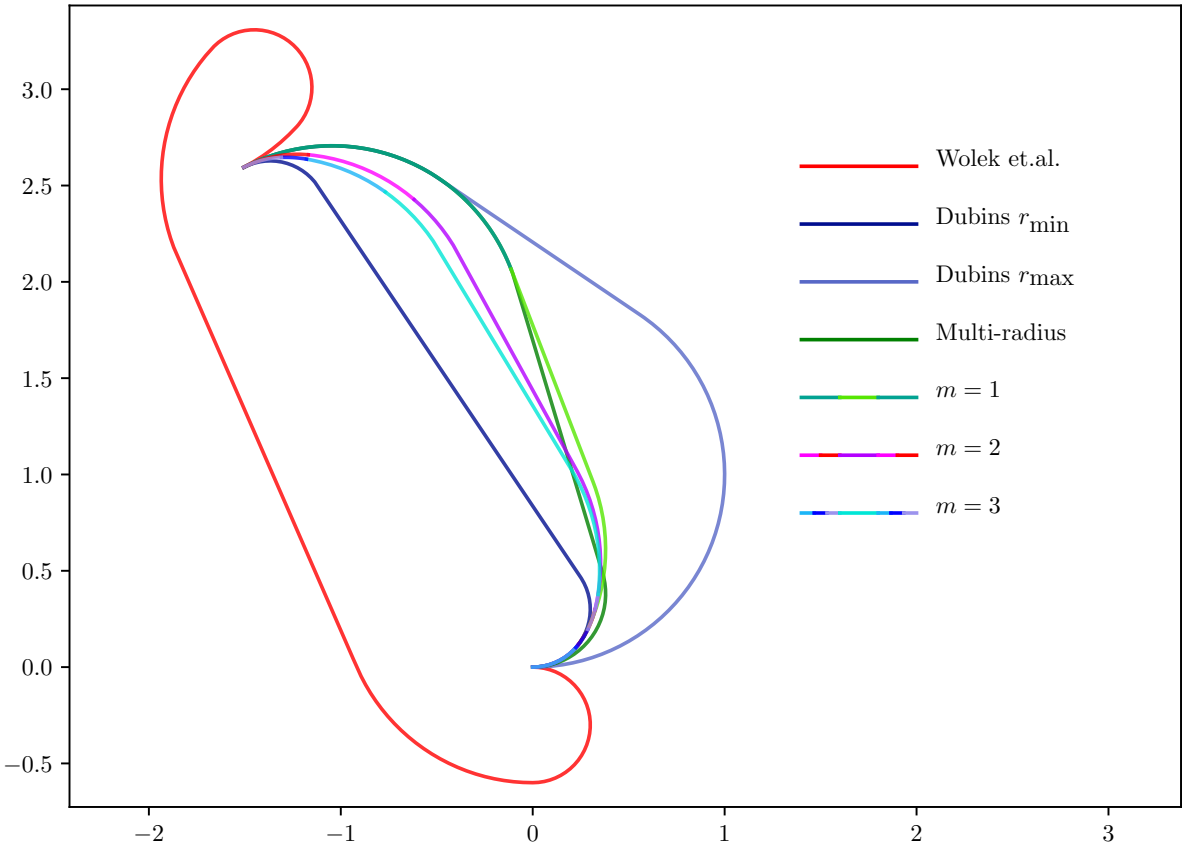


Figure 4.10: Example of Dubins path, multi-radius Dubins path, multi-segment trajectory, and trajectory by Wolek et al.

CHAPTER 5

Multi-Radius Dubins Path in Multi-Goal Scenarios

In this chapter, we describe how the proposed extended Dubins path with multiple radii is deployed in time-optimal trajectory planning over multiple locations defined as Problem 2. The problem is modeled as a variant of the Dubins Traveling Salesman Problem (DTSP) [8]. In the DTSP, the problem is to find a closed, cost-efficient tour to visit each location exactly once, and the tour has to represent the fastest feasible trajectory for the Dubins vehicle. The DTSP is a well-studied problem with many existing approaches [21, 11], and therefore, deployment of the proposed multi-radius extension of the Dubins path is relatively straightforward. The main difficulty of solving instances of the DTSP is in the combination of the

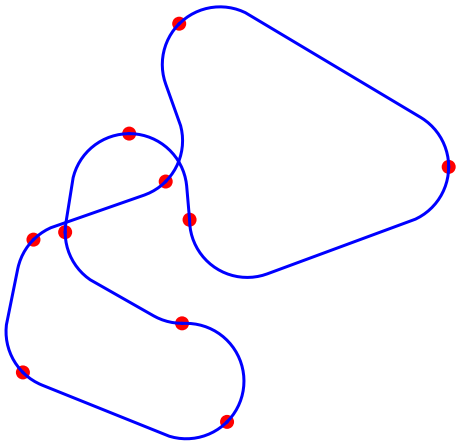


Figure 5.1: Example of DTSP with 10 random locations.

combinatorial optimization of the underlying TSP with continuous optimization of the fastest trajectory. In the Dubins vehicle case, it is necessary to determine the optimal heading angles of the vehicle for each target location from which optimal Dubins paths can be computed to get a smooth multi-goal Dubins path; see an example of the solution in Fig. 5.1.

Difficulties of solving the DTSP arising from the continuous optimization of the vehicle headings can be addressed by discretizing the possible heading angles into a finite set of sam-

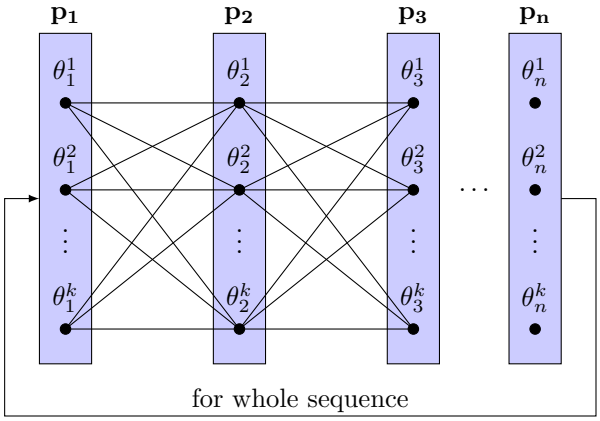


Figure 5.2: Search graph with k sampled heading angles per each of n target locations and given sequence of visits to the locations.

ples [21, 11]. Then, for a given sequence of visits to the target locations, we can construct a graph as in Fig. 5.2 and find optimal heading angles, and thus optimal solution on the given discretization and sequence of visits. For k heading samples per each target locations and a sequence of n target locations, the complexity of finding the solution can be bounded by $O(nk^3)$ [22]. Thus, depending on the number of samples k, the solution can be found relatively quickly. However, we also need to determine a sequence of visits, which is a NP-hard combinatorial optimization problem because of the underlying TSP [9].

In the solution of the DTSP, the sequence can be determined by relaxing motion constraints of the Dubins vehicle and solution of the Euclidean TSP [22]. Such a decoupled approach solves the sequencing part and finds optimal heading angles independently, and thus an unfortunate sequence can be determined. Therefore, we propose to utilize a coupled approach where the sequence is optimized on top of the discretized heading angles, which can be further formulated as the Generalized TSP (GTSP) [11]. The GTSP instance size depends on the number of samples, and since the proposed multi-radius Dubins path has two radii in addition to the heading angle, the size of the instance quickly grows. That is why we instead consider combinatorial metaheuristic called the Variable Neighborhood Search (VNS) [10] to address the generation of the DTSP with multi-radius Dubins paths.

Algorithm 1: VNS-based solver for the DTSP

Input: $\mathcal{P}$ – the set of locations coordinates
Output: Σ – the visiting sequence of the locations $\mathcal{P}$.

$\Sigma \leftarrow$ initial solution found by cheapest insertion `// Initialization`
while not terminating condition **do** `// Main step`
  $\Sigma' \leftarrow$ `shaking`(Σ)
  $\Sigma'' \leftarrow$ `localSearch`(Σ')
  if $\mathcal{T}(\Sigma'') \leq \mathcal{T}(\Sigma)$ **then**
    $\Sigma \leftarrow \Sigma''$
  end
end

The VNS is a general metaheuristic for solving combinatorial optimization problems that has been already adapted to routing problems with the Dubins vehicle [13, 23, 12, 11]. Furthermore, VNS can get out of the local optima and find the solution with a cost closer to the global optimum than the initial solution. The VNS consists of two phases. The first is the initialization, where the initial solution is created, e.g., using the cheapest insertion [24]. The second phase is iterative optimization until the terminating condition, such as the maximum number of iterations or given computational time, is met. The optimization phase runs two procedures called `shaking` and `localSearch`, which are implementation-dependent. The operation `shaking` randomly modifies the previous best solution, and `localSearch` locally optimizes the modified solution. If the modified and optimized solution is better than the previous best solution, it becomes the new best solution. The VNS-based solver to the DTSP is overviewed in Algorithm 1.

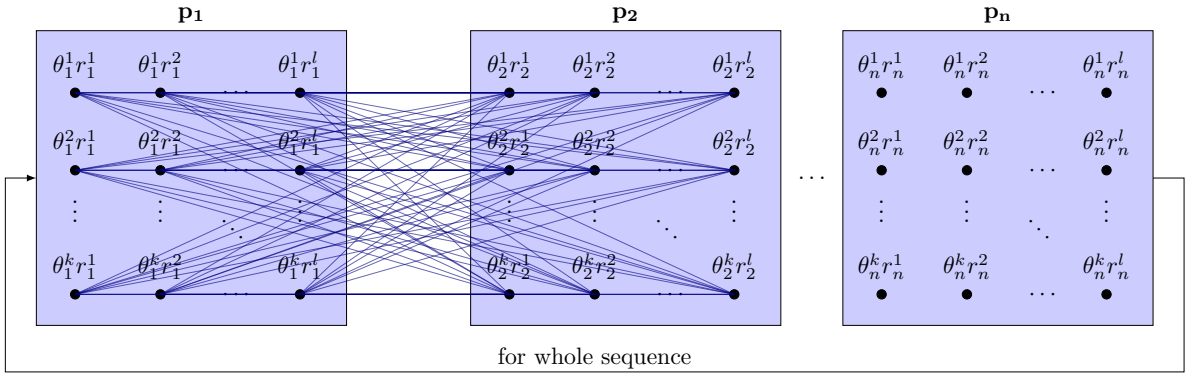


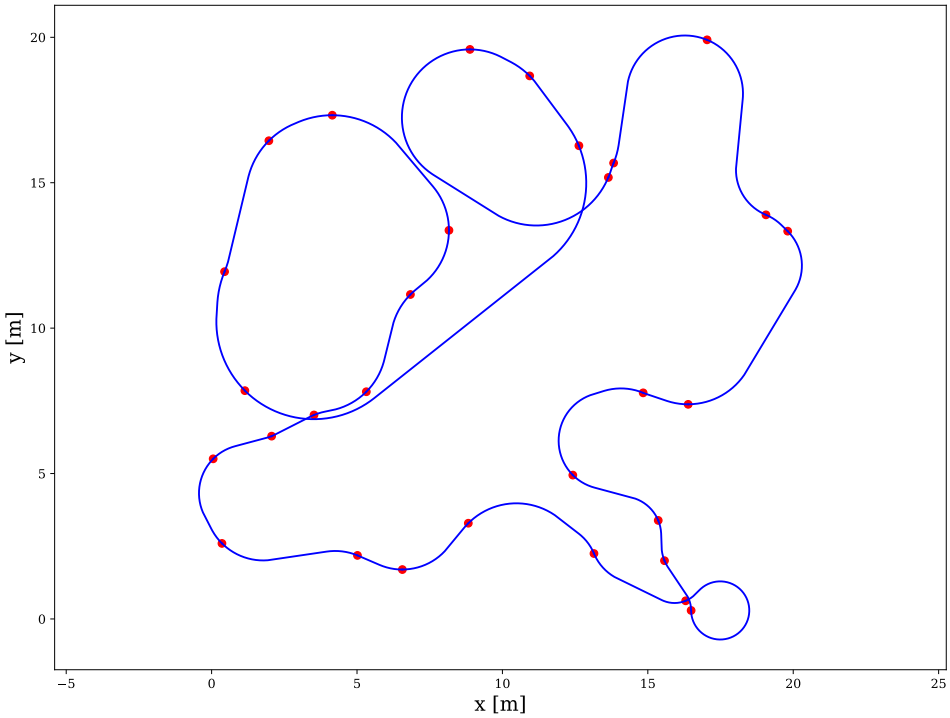
Figure 5.3: Search graph with heading angle and turning radii samples. The sequence is n locations long, the number of heading angle samples as k, and the number of turning radii samples is l.

The main problem to employ the existing VNS-based solver, in particular, we consider available implementation [12], is to determine individual trajectories connecting the target locations. The continuous optimization is addressed by discretizing the heading angles in k samples and possible turning radii into l samples. For a particular sequence, i.e., determined during the VNS-based combinatorial optimization, the multi-goal trajectory can be a similar approach as for the DTSP. However, instead of the search graph shown in Fig. 5.2, kl configurations are considered for each target locations, and the search graph looks like in Fig. 5.3.

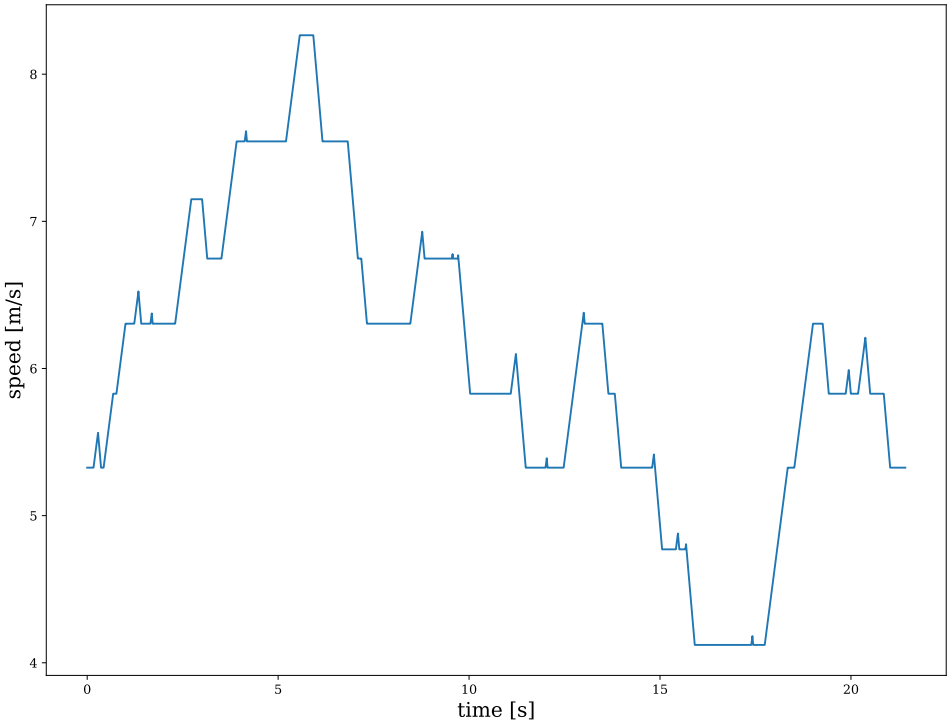
The complexity of multi-goal trajectory assessment for a particular sequence is relatively costly because of more samples and more demanding multi-radius Dubins path than for the DTSP. Therefore, we further employ windowing speed-up technique [25] based on lower bound estimation for quick rejection of unfavorable sequences [26]. Empirical results on multi-goal planning are reported in the following section.

5.1 Evaluation of the Multi-goal Trajectories

The proposed multi-radius Dubins path has been deployed in the VNS-based DTSP solver with precomputed paths between every two samples of turning radii and heading angles. We examined the influence of the sampling to the achieved TTE in random scenarios with the number of target locations ranging from 10 to 75. The target locations are generated randomly



(a) Found trajectory with 10 radii samples



(b) Speed profile

Figure 5.4: Example of the found multi-goal trajectory with 30 target locations and its corresponding speed profile.

with uniform distribution within squared bounding-box with the size 20. The heading angles are sampled uniformly in the range $[0, 2\pi)$. The turning radii are sampled with linear and exponential distribution. The vehicle motion constraints as of Table 4.1 are used with the minimum turning radius scaled to 1. An example of the found multi-goal trajectory with 30 locations is depicted in Fig. 5.4. For brevity, the results with 50 locations for the TTE examination in the rest of the section.

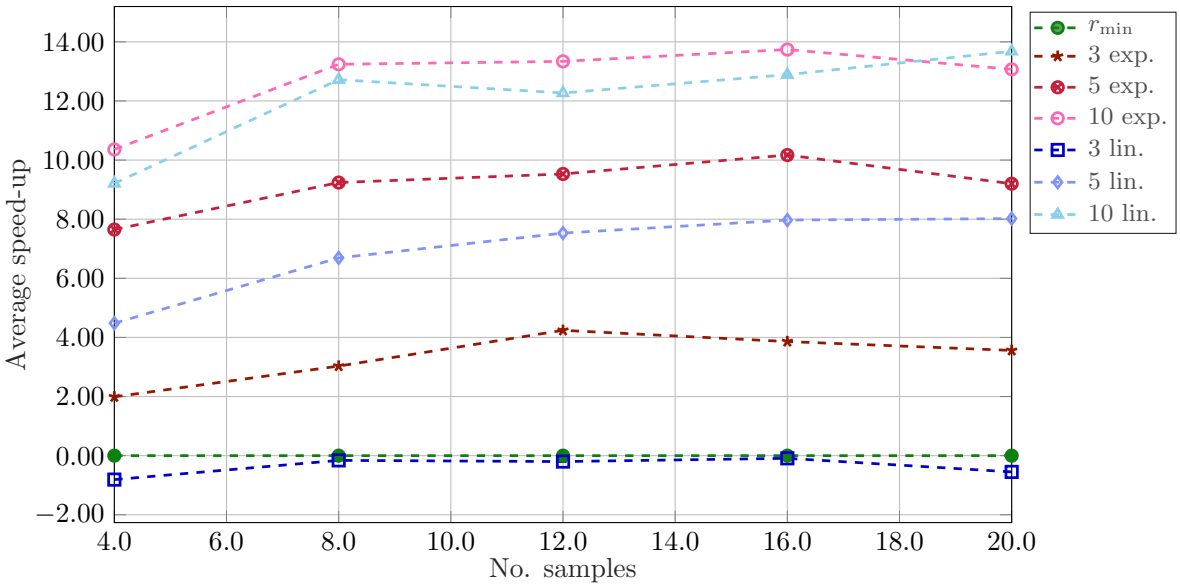


Figure 5.5: Speed-up for multi-goal trajectory with 50 randomly generated target locations for increasing number of heading samples. Average values among 50 trials. Linear and exponential sampling schemata (see Section 4.1.3) are employed with 3, 5, and 10 samples.

We examine the results of different sampling on 50 test instances with different number of locations. The linear and exponential sampling schemata from Section 4.1.3 are utilized with 3, 5, and 10 samples. The average achieved speed-up for multi-goal trajectories with 50 target locations is shown in Fig. 5.5 and detail results are listed in Table 5.1. The values of Slow time are the trajectory travel time without speed increase, and values of Fast time are with speed increase. The speed-up of the Fast time to the Slow time is computed as

$$\text{Speed-up} = 100 \left(\frac{\text{Slow time}}{\text{Fast time}} - 1 \right). \tag{5.1}$$

The Speed-up vs. $r_{\min}$ is the speed-up of the Fast time of the multi-radius Dubins path to the Fast Time of Dubins path with $r_{\min}$ with same number of heading angle samples. The data show that the exponential sampling schema achieves better results when considering the same number of angle samples. The largest speed-up in comparison to $r_{\min}$ is >13 %, but for a different number of goals in the same bounding box, the speed-up can be over 30 %, see evaluations in Section B.2. The execution time increases considerably with larger number of heading angle samples.

Table 5.1: Results for data with 50 goals, averaged.

Num. goals	Radii samples	Angle samples	Fast time	Slow time	Path length	Execute [s]	Speed-up [%]	Speed-up vs. $r_{\min}$ [%]
50	$r_{\min}$	4	41.51	43.42	178.95	10.80	4.60	0.00
50	$r_{\min}$	8	37.12	38.82	160.01	15.78	4.58	0.00
50	$r_{\min}$	12	35.42	37.01	152.55	23.00	4.50	0.00
50	$r_{\min}$	16	34.68	36.26	149.46	33.18	4.55	0.00
50	$r_{\min}$	20	34.08	35.62	146.79	45.63	4.50	0.00
50	3 exp	4	40.70	47.05	193.92	24.73	6.86	1.99
50	3 exp	8	36.03	42.94	176.99	65.61	7.90	3.03
50	3 exp	12	33.98	40.29	166.03	128.81	9.09	4.24
50	3 exp	16	33.39	40.20	165.67	220.05	8.74	3.86
50	3 exp	20	32.91	40.01	164.89	363.91	8.39	3.56
50	5 exp	4	38.56	48.31	199.09	51.56	12.73	7.65
50	5 exp	8	33.98	43.20	178.03	154.11	14.39	9.24
50	5 exp	12	32.31	41.43	170.74	361.38	14.71	9.63
50	5 exp	16	31.48	40.98	168.90	912.99	15.35	10.17
50	5 exp	20	31.21	41.39	170.57	1671.50	14.30	9.20
50	10 exp	4	37.52	49.13	202.47	158.23	15.91	10.63
50	10 exp	8	32.78	44.05	181.55	654.71	18.65	13.24
50	10 exp	12	31.25	42.44	174.92	2011.68	18.62	13.34
50	10 exp	16	30.49	41.50	171.05	4032.53	19.09	13.74
50	10 exp	20	30.14	42.01	173.12	5868.47	18.40	13.07
50	3 lin	4	41.85	47.00	193.71	24.81	3.92	-0.81
50	3 lin	8	37.18	43.68	180.01	159.78	4.57	-0.16
50	3 lin	12	35.49	43.58	179.62	224.02	4.47	-0.20
50	3 lin	16	34.71	44.04	181.51	316.86	4.59	-0.09
50	3 lin	20	34.27	43.74	180.26	710.15	4.09	-0.55
50	5 lin	4	39.73	49.36	203.43	54.39	9.43	4.48
50	5 lin	8	34.79	44.14	181.94	166.94	11.79	6.69
50	5 lin	12	32.94	43.46	179.12	354.85	12.55	7.53
50	5 lin	16	32.12	43.06	177.46	666.93	13.12	7.97
50	5 lin	20	31.55	42.57	175.44	1737.26	13.01	8.02
50	10 lin	4	38.01	49.99	206.02	156.25	14.43	9.21
50	10 lin	8	32.93	44.20	182.15	655.16	18.05	12.72
50	10 lin	12	31.55	43.18	177.97	2257.73	17.51	12.27
50	10 lin	16	30.72	43.03	177.35	3525.81	18.23	12.89
50	10 lin	20	29.98	41.96	172.92	5380.44	19.01	13.68

CHAPTER 6

Conclusion

In this thesis, two different time-optimal trajectory planning methods with the extended Dubins vehicle have been proposed. The proposed generalization of the Dubins vehicle considers the variable speed and multiple turning radii. The methods are called *multi-radius Dubins path* and *multi-segment trajectory*. Both trajectories utilize variable turning radii to find the tradeoff between the trajectory length and time needed to travel the trajectory. The multi-radius Dubins path is further utilized in multi-goal planning formulated as a time-optimal variant of the Dubins Traveling Salesman Problem (DTSP). The time-optimal multi-goal trajectory planning is addressed by VNS-based combinatorial metaheuristic. A discretized variant of the trajectory planning is utilized with sampled heading angles and turning radii.

The multi-radius Dubins path follows the CSC type of the Dubins path, and it consists of three segments, where the ending segments are arcs of various radii. For the given radii, the multi-radius Dubins path can be computed using the closed-form expressions; however, finding the suitable radii is computationally challenging. Two computational methods are proposed: local optimization and discretization. The number of samples and sampling distributions influence the solution, and linear and exponential distributions have been examined. Both distributions of 10 radii samples achieve similar time improvement as the local optimization. However, the computational time of the local optimization is, on average, a thousand times longer than using ten radii samples. The exponential sampling provides better results than linear when a smaller number of samples is used. Thus, the sampling-based approach is more suitable for practical use, and exponential sampling might be preferred when computational time is constrained.

The travel time improvement of the multi-radius Dubins path over the regular Dubins path depends on the distance between the locations. For mutually distant locations, the vehicle can accelerate on the straight segment of the CSC path type. The best achieved value of speed-up reaches up to 90 % at a distance about four times the minimum turning radius $r_{\min}$ in comparison to the Dubins path with $r_{\min}$. For a distance of the locations about $15r_{\min}$, the average improvement with ten radii samples on the trajectory travel time is about 20 %.

The multi-segment trajectory decreases the trajectory travel time even further than the multi-radius Dubins path at the cost of significantly increased computational requirements. Due to the increased number of trajectory parameters, local optimization is utilized to find multi-segment trajectories. The optimization can be initialized by the Dubins path with $r_{\min}$.

However, initialization using the proposed multi-radius Dubins path provides better results by means of decreased computational time to find faster trajectories.

Furthermore, the proposed time-optimal trajectories have been compared with the approach of Wolek et al. [4], which, however, does not consider the acceleration limits of the vehicle. Using the same vehicle model, both the proposed multi-radius Dubins path and multi-segment trajectory provides faster trajectories than the method [4].

Finally, the multi-radius Dubins path has been employed in multi-goal planning where it provides time improvement about 10–30 % in comparison to the Dubins path with $r_{\min}$, based on the number of target goals. The computational time increases significantly with a higher number of goals. For datasets with 50 locations, the computational time is ten times longer when using 3 radii samples instead of $r_{\min}$, and hundred times with 10 radii samples.

Based on the reported results, we can conclude that the proposed trajectories reduce the travel time in all examined cases. Besides, the multi-radius Dubins path is computationally effective, and a closed-form expression exists for both the trajectory and travel time computation.

Bibliography

[1] 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.

[2] XN Bui, JD Boissonnat, JP Laumond, and P Soures. *The shortest path synthesis for non-holonomic robots moving forwards, INRIA, Nice-Sophia-Antipolis*. Research Report 2153, 1994.

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

[4] 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*, pages 2374–2390, 2016.

[5] Pratap Tokekar, Nikhil Karnad, and Volkan Isler. Energy-optimal trajectory planning for car-like robots. *Autonomous Robots*, 37(3):279–300, 2014.

[6] Petr Váňa, Jakub Sláma, Jan Faigl, and Pavel Pačes. Any-time trajectory planning for safe emergency landing. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5691–5696, 2018.

[7] Kristýna Kučerová, Petr Váňa, and Jan Faigl. On finding time-efficient trajectories for fixed-wing aircraft using dubins paths with multiple radii. In *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, pages 829–831, 2020.

[8] Jerome Ny, Eric Feron, and Emilio Frazzoli. On the dubins traveling salesman problem. *IEEE Transactions on Automatic Control*, 57(1):265–270, 2011.

[9] David L Applegate, Robert E Bixby, Vasek Chvatal, and William J Cook. *The traveling salesman problem: a computational study*. Princeton university press, 2006.

[10] Nenad Mladenović and Pierre Hansen. Variable neighborhood search. *Computers & operations research*, 24(11):1097–1100, 1997.

[11] Jan Faigl, Petr Váňa, Robert Pěnička, and Martin Saska. Unsupervised learning-based flexible framework for surveillance planning with aerial vehicles. *Journal of Field Robotics*, 36(1):270–301, 2019.

[12] Petr Váňa, Jakub Sláma, and Jan Faigl. The dubins traveling salesman problem with neighborhoods in the three-dimensional space. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 374–379, 2018.

[13] Robert Pěnička, Jan Faigl, Petr Váňa, and Martin Saska. Dubins orienteering problem. *IEEE Robotics and Automation Letters*, 2(2):1210–1217, 2017.

[14] Ji-wung Choi, Renwick Curry, and Gabriel Elkaim. Path planning based on bézier curve for autonomous ground vehicles. In *Advances in Electrical and Electronics Engineering-IAENG Special Edition of the World Congress on Engineering and Computer Science 2008*, pages 158–166. IEEE, 2008.

[15] A Askari, M Mortazavi, HA Talebi, and A Motamedi. A new approach in uav path planning using bezier–dubins continuous curvature path. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 230(6):1103–1113, 2016.

[16] Jan Faigl and Petr Váňa. Surveillance planning with bézier curves. *IEEE Robotics and Automation Letters*, 3(2):750–757, 2018.

[17] Liang Yang, Dalei Song, Jizhong Xiao, Jianda Han, Liying Yang, and Yang Cao. Generation of dynamically feasible and collision free trajectory by applying six-order bezier curve and local optimal reshaping. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 643–648. IEEE, 2015.

[18] *Information Manual SKYHAWK*. Cessna Aircraft Company, 1998.

[19] Patrick Kofod Mogensen and Asbjørn Nilsen Riseth. Optim: A mathematical optimization package for Julia. *Journal of Open Source Software*, 3(24):615, 2018.

[20] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer, 2006.

[21] Paul Oberlin, Sivakumar Rathinam, and Swaroop Darbha. Today's traveling salesman problem. *IEEE robotics & automation magazine*, 17(4):70–77, 2010.

[22] Jan Faigl, Petr Váňa, Martin Saska, Tomáš Báča, and Vojtěch Spurnỳ. On solution of the dubins touring problem. In *2017 European Conference on Mobile Robots (ECMR)*, pages 1–6. IEEE, 2017.

[23] Petr Váňa, Jan Faigl, Jakub Sláma, and Robert Pěnička. Data collection planning with dubins airplane model and limited travel budget. In *European Conference on Mobile Robots (ECMR)*, pages 1–6, 2017.

[24] Daniel J Rosenkrantz, Richard E Stearns, and Philip M Lewis, II. An analysis of several heuristics for the traveling salesman problem. *SIAM journal on computing*, 6(3):563–581, 1977.

[25] Jan Drchal, Jan Faigl, and Petr Váňa. Wism: Windowing surrogate model for evaluation of curvature-constrained tours with dubins vehicle. *IEEE Transactions on Cybernetics*, 2020.

[26] Jan Faigl, Petr Váňa, and Jan Drchal. Fast sequence rejection for multi-goal planning with dubins vehicle. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020. (To appear).

APPENDIX A

Computations

A.1 Computation of the Dubins Path

Dubins path is a three-segment path, consisting of arcs (part of a circle) or a straight line. The arc is denoted as C and straight segment S, and the Dubins path can be of type CSC or CCC. The described method of computing the Dubins path is based on finding the centers of the arc segments. The required values for path construction are the turning radius, and the initial and final configurations $\mathbf{q_1}, \mathbf{q_2}$. The configurations are $\langle \mathbf{p}, \theta \rangle \in SE(2)$, where $\mathbf{p} \in \mathbb{R}^2$ is the position in the 2D space, and $\theta \in [0, 2\pi)$ is the heading orientation of the vehicle. To simplify the computation, the initial position $\mathbf{p_1}$ is transformed to $\langle 0, 0 \rangle$ and the final position $\mathbf{p_2}$ to $\langle d, 0 \rangle$, where d is the distance between the configurations $\mathbf{q_1}$ and $\mathbf{q_2}$. The computation of the CSC path type is described in Section A.1.1 and CCC in Section A.1.2

A.1.1 CSC Dubins Path Type

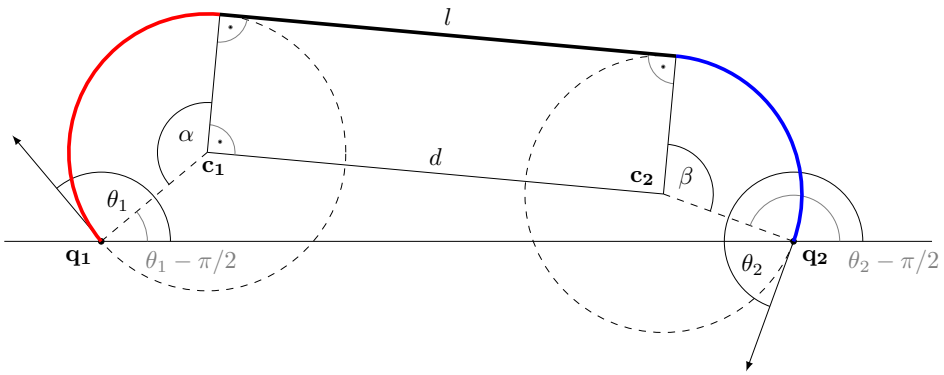


Figure A.1: RSR Dubins path type.

The RSR path type is visualized in Fig. A.1 and RSL in Fig. A.2. To construct the path, the angles of the initial and the final arc α, β and the length of the straight segment l are needed. The computation of the CSC maneuver begins with determining the center of each turn. Let the center of the initial turn be denoted $\mathbf{c_1}$ and the final one $\mathbf{c_2}$. The configuration

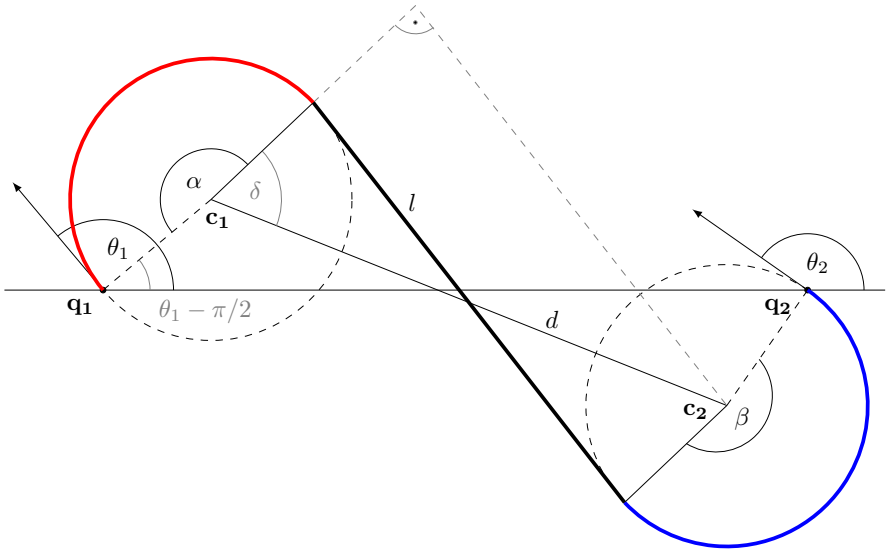


Figure A.2: RSL Dubins path type.

$\mathbf{q_i}$ lies at the tangent of the circle, and the coordinates of the vector with the length of the radius r and angle $\theta_i \pm \frac{\pi}{2}$ are obtained using the sin and cos functions. The angle correction of $\frac{\pi}{2}$ is added based on the turn directions as

$$\mathbf{c_i} = \begin{cases} \mathbf{p_i} + r\,[\sin\theta_i, -\cos\theta_i] & \text{for right turn center,} \\ \mathbf{p_i} + r\,[-\sin\theta_i, \cos\theta_i] & \text{for left turn center.} \end{cases} \tag{A.1}$$

After computing the path centers, the angle ξ is calculated, which is the direction of the vector $\mathbf{c_1c_2} = \langle x_c, y_c \rangle$ between the circle centers $\mathbf{c_1}$ and $\mathbf{c_2}$,

$$\xi = \arctan \frac{y_c}{x_c}. \tag{A.2}$$

The distance d between the circle centers is calculated as

$$d = ||\mathbf{c_2} - \mathbf{c_1}|| = \sqrt{x_c^2 + y_c^2}. \tag{A.3}$$

The turning angles α and β are computed using the following equations, where $\alpha, \beta \in [0, 2\pi)$ and can be shifted to the interval by the correction constant $2k\pi, k \in \mathbb{Z}$ if necessary,

$$\alpha = \begin{cases} +(\theta_1 + \frac{\pi}{2}) - (\xi + \delta) & \text{for RSR/RSL,} \\ -(\theta_1 - \frac{\pi}{2}) + (\xi - \delta) & \text{for LSL/LSR,} \end{cases} \tag{A.4}$$

$$\beta = \begin{cases} -(\theta_2 + \frac{\pi}{2}) + (\xi + \delta) & \text{for RSR,} \\ +(\theta_2 - \frac{\pi}{2}) - (\xi - \delta) & \text{for LSL,} \\ +(\theta_2 - \frac{\pi}{2}) - (\xi + \delta + \pi) & \text{for RSL,} \\ -(\theta_2 + \frac{\pi}{2}) + (\xi - \delta + \pi) & \text{for LSR,} \end{cases} \tag{A.5}$$

and the auxiliary angle δ is computed by

$$\delta = \begin{cases} \frac{\pi}{2} & \text{for RSR/LSL,} \\ \arccos \frac{2r}{d} & \text{for RSL/LSR.} \end{cases} \tag{A.6}$$

The total length of the Dubins path is denoted $\mathcal{L}$ and is calculated using the arc angles α, β and the length of the central segment l as

$$\mathcal{L} = \alpha r + l + \beta r. \tag{A.7}$$

The length of the center segment can be calculated as

$$l = \begin{cases} d & \text{for RSR/RSL,} \\ \sqrt{d^2 - 4r^2} & \text{for LSL/LSR.} \end{cases} \tag{A.8}$$

A.1.2 CCC Dubins Path Type

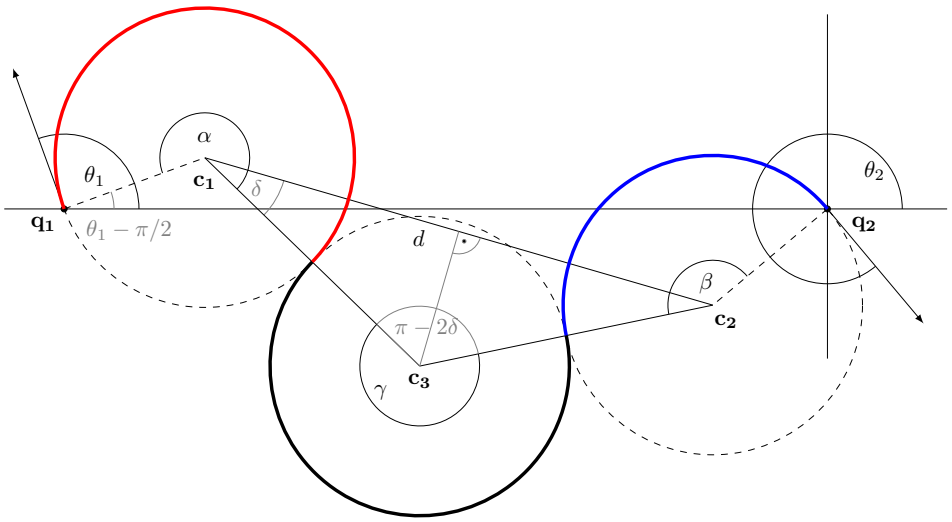


Figure A.3: RLR Dubins path type.

For the CCC paths (Fig. A.3), the middle segment is also a part of a circle, and the path computation differs slightly from the computation of the CSC maneuvers. The centers $\mathbf{c_1}$ and $\mathbf{c_2}$ are computed the same as for the CSC maneuvers by (A.1). The angle ξ (A.2) and distance d between the circle centers $\mathbf{c_1}$, $\mathbf{c_2}$ (A.3) are also needed. However, the CCC path type is only feasible if the distance d is less than or equal to $4r$, otherwise the initial and final segments cannot be connected by a part of circle with radius r. The center $\mathbf{c_3}$ is the center of the middle segment and is computed as

$$\mathbf{c_3} = \begin{cases} \mathbf{c_1} + 2r[\cos(\xi - \delta), \sin(\xi - \delta)] & \text{for RLR,} \\ \mathbf{c_1} + 2r[\cos(\xi + \delta), \sin(\xi + \delta)] & \text{for LRL.} \end{cases} \tag{A.9}$$

The angles of the curve segments α, β and γ are computed using the following equations, where $\alpha, \beta, \gamma \in [0, 2\pi)$ and can be shifted to the interval by the correction constant $2k\pi, k \in \mathbb{Z}$ if necessary,

$$\alpha = \begin{cases} +(\theta_1 + \frac{\pi}{2}) - (\xi - \delta) & \text{for RLR,} \\ -(\theta_1 - \frac{\pi}{2}) + (\xi + \delta) & \text{for LRL,} \end{cases} \tag{A.10}$$

$$\beta = \begin{cases} -(\theta_2 + \frac{\pi}{2}) + (\xi + \pi + \delta) & \text{for RLR,} \\ +(\theta_2 - \frac{\pi}{2}) - (\xi + \pi - \delta) & \text{for LRL,} \end{cases} \tag{A.11}$$

$$\gamma = \pi + 2\delta + 2k\pi \quad \text{for both RLR and LRL,} \tag{A.12}$$

and the auxiliary angle δ for both RLR and LRL maneuvers is determined as

$$\delta = \arccos \frac{d}{4r}. \tag{A.13}$$

The total length of the Dubins path for the CCC maneuver is

$$\mathcal{L} = (\alpha + \gamma + \beta)r. \tag{A.14}$$

A.2 Computation of the Multi-Radius Dubins Path

The closed-form expression exists for the Dubins path, and constructing the path is computationally efficient. The closed-form expression for the multi-radius Dubins path is presented in this thesis.

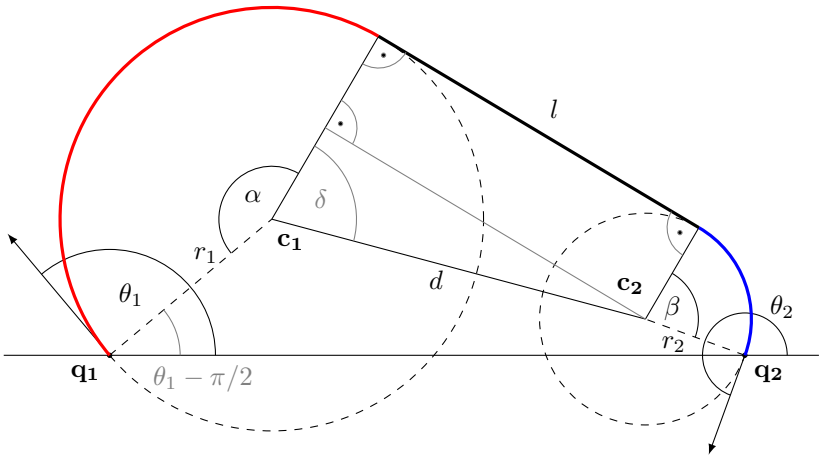


Figure A.4: Examples of multi-radius Dubins paths: RSR maneuver.

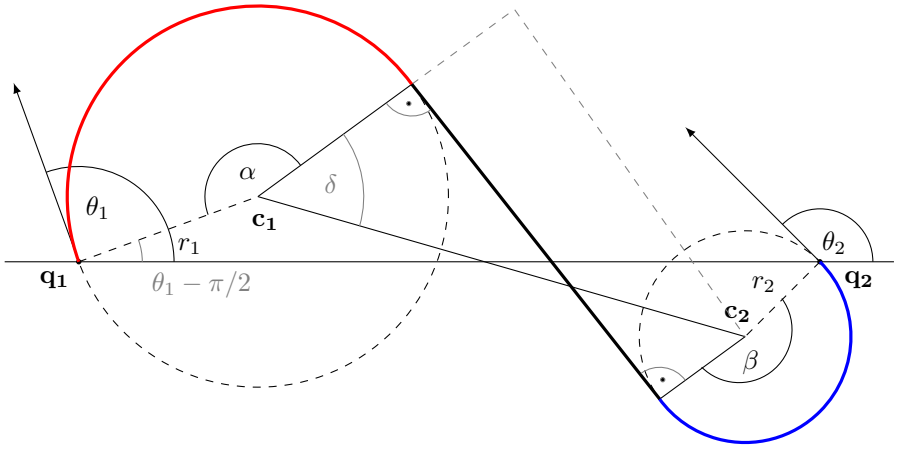


Figure A.5: Examples of multi-radius Dubins paths: RSL maneuver.

The centers of the initial and final segments are computed the same way as for the single radius Dubins with using the correct curve radius. The distance between the centers $\mathbf{c_1}$ and $\mathbf{c_2}$ is denoted as d. The angle ξ of the vector $\mathbf{c_1c_2}$ and total path length $\mathcal{L}$ are also calculated the same way, see equations (A.1-A.7).

Calculation of the length of the straight segment l is different as the straight segment is no longer perpendicular to the vector between the circle centers $\mathbf{c_1c_2}$,

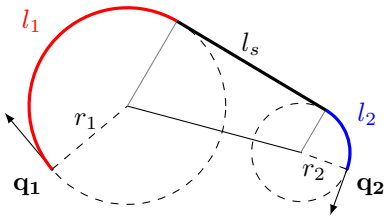
$$l = \begin{cases} \sqrt{d^2 - (r_1 - r_2)^2} & \text{for RSR/RSL,} \\ \sqrt{d^2 - (r_1 + r_2)^2} & \text{for LSL/LSR.} \end{cases} \tag{A.15}$$

The turn angles α and β are computed almost as for the single radius Dubins with the only difference in computation of the auxiliary angle δ,

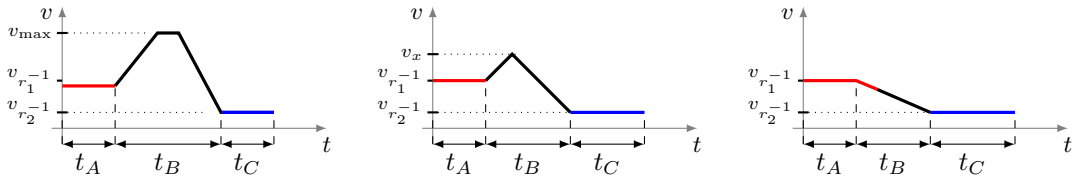
$$\delta = \begin{cases} \arccos \frac{r_1 - r_2}{d} & \text{for RSR/LSL,} \\ \arccos \frac{r_1 + r_2}{d} & \text{for RSL/LSR.} \end{cases} \tag{A.16}$$

A.3 TTE computation for the multi-radius Dubins path

In this section, the computation of each speed profile case is described. The possible cases of the speed profile are shown in Fig. A.6.



(a) The RSR multi-radius Dubins path.



(b) Three different speed profile cases.

Figure A.6: Speed profile for three-segment Dubins path.

The computation of the TTE for multi-radius Dubins path starts with determining if the vehicle is able to accelerate to maximum vehicle speed $v_{\max}$ on the straight segment. The time t_1 needed to accelerate from $\hat{v}(r_1)$ to $v_{\max}$ is computed as

$$t_1 = \frac{v_{\max} - \hat{v}(r_1)}{a_{\max}}, \tag{A.17}$$

and time t_2 to decelerate from $v_{\max}$ to $\hat{v}(r_r)$ is

$$t_2 = \frac{\hat{v}(r_1) - v_{\max}}{a_{\min}}. \tag{A.18}$$

In time t_i, the vehicle travels the length l_i, $i \in \{1, 2\}$, which is computed as

$$l_{t_i} = t_i \frac{\hat{v}(r_i) + v_{\max}}{2}. \tag{A.19}$$

If the length sum $l_{t_1} + l_{t_2}$ is shorter or equal to the length of the straight segment l_s, the speed-up to the maximum speed is possible. The total time of the path T, which is the value of TTE, is in this case computed as

$$T = \underbrace{\frac{l_1}{\hat{v}(r_1)}}_{t_A} + \underbrace{t_1 + \frac{l_s - l_{t_1} - l_{t_2}}{v_{\max}} + t_2}_{t_B} + \underbrace{\frac{l_2}{\hat{v}(r_2)}}_{t_C} \quad \text{for case 1.} \tag{A.20}$$

If $l_{t_1} + l_{t_2}$ is larger than l_s, and the vehicle can change the speed from $\hat{v}(r_1)$ to $\hat{v}(r_2)$ on distance shorter than l_s, the maximum reachable speed v_x has to be calculated as

$$v_x = \sqrt{\frac{a_{\max}\hat{v}(r_2)^2 - 2a_{\max}a_{\min}l_s - a_{\min}\hat{v}(r_1)^2}{a_{\max} - a_{\min}}}, \tag{A.21}$$

and the total time of the path T is in this case computed as

$$T = \underbrace{\frac{l_1}{\hat{v}(r_1)}}_{t_A} + \underbrace{\frac{v_x - \hat{v}(r_1)}{a_{\max}} + \frac{\hat{v}(r_2) - v_x}{a_{\min}}}_{t_B} + \underbrace{\frac{l_2}{\hat{v}(r_2)}}_{t_C} \quad \text{for case 2.} \tag{A.22}$$

The speed profile can be computed using closed-form expression, and calculation of the TTE is computationally efficient and easy to implement into the final solution. The speed profile looks differently based on the value of initial and final radius, and all the posssible speed profiles for the multi-radius Dubins are shown in Figures A.7–A.9

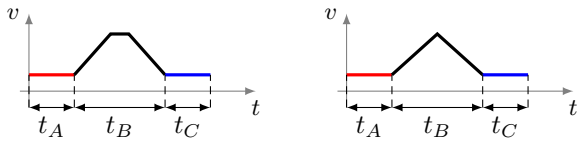


Figure A.7: Speed profile: Equal initial speed and final speed of the vehicle.

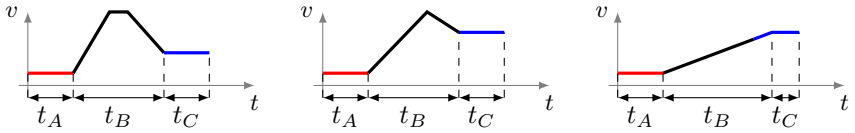


Figure A.8: Speed profile: Initial vehicle's speed smaller than final.

In the case 3 in Fig. A.8, the TTE is calculated as:

$$T = \underbrace{\frac{l_1}{\hat{v}(r_1)}}_{t_A} + \underbrace{\frac{\hat{v}(r_2) - \hat{v}(r_1)}{a_{\max}}}_{t_B} + \underbrace{\frac{l_s + l_2 - \overbrace{\frac{\hat{v}(r_2)^2 - \hat{v}(r_1)^2}{2a_{\max}}}^{l_{t_B}}}{\hat{v}(r_2)}}_{t_C}. \tag{A.23}$$

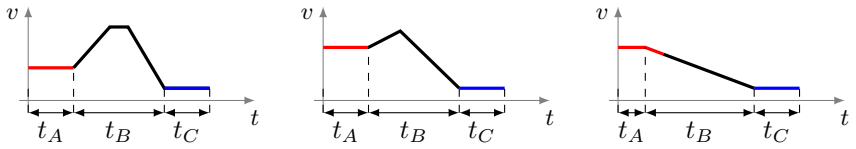


Figure A.9: Speed profile: Initial vehicle's speed larger than final.

The case 3 in Fig. A.9 is the reverse of case 3 in Fig. A.8, and the TTE is calculated as

$$T = \underbrace{\frac{l_1 + l_s - \overbrace{\frac{\hat{v}(r_2)^2 - \hat{v}(r_1)^2}{2a_{\min}}}^{l_{t_B}}}{\hat{v}(r_1)}}_{t_A} + \underbrace{\frac{\hat{v}(r_2) - \hat{v}(r_1)}{a_{\max}}}_{t_B} + \underbrace{\frac{l_2}{\hat{v}(r_2)}}_{t_C}. \tag{A.24}$$

APPENDIX B

Evaluations

B.1 Radii Distribution on higher distances

The average speed-up of the multi-radius Dubins path on larger distances is shown in Fig. B.1. The highest average speed-up is 20 % at distance approximately $15r_{\min}$. The speed-up for $r_{\min}$ is omitted in the plot for better visualization.

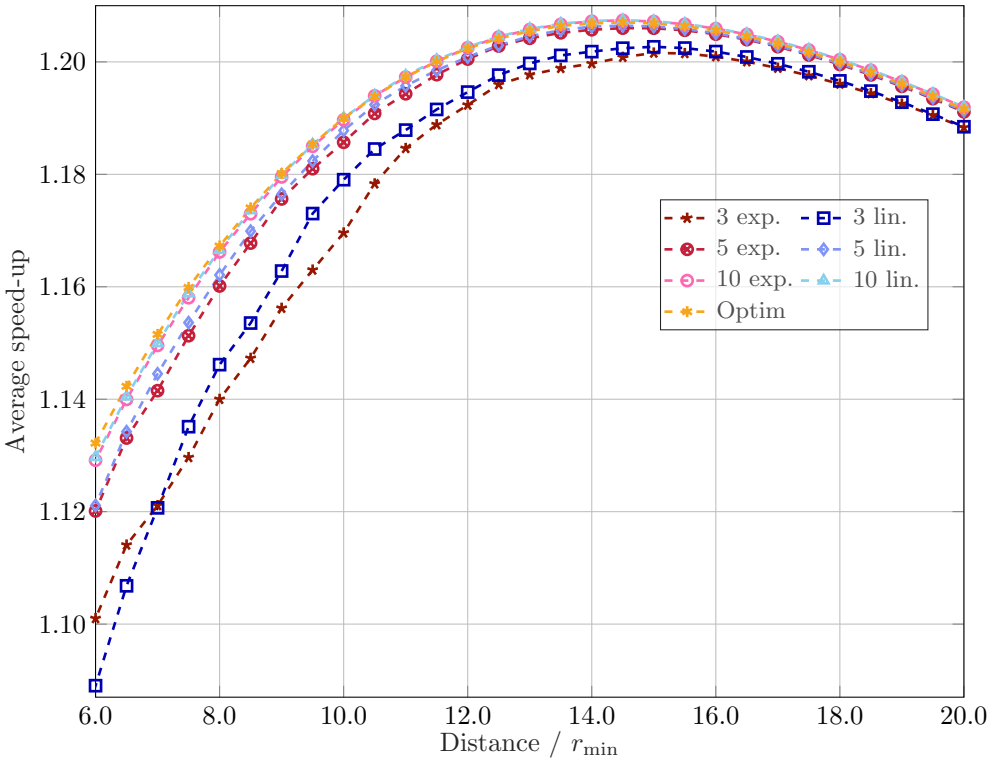


Figure B.1: Speed-up based on the distance between initial and final configuration from distance $6r_{\min}$ up to $20r_{\min}$. All results are averaged over 500 randomly generated instances.

B.2 Different Number of Locations in Multi-Goal Trajectories Evaluations

The evaluations for different number of target locations in multi-goal trajectories are shown in this section. The results show that for densely spaced goals the achieved speed-up is smaller than for sparse goals datasets. This corresponds to the evaluation of trajectories from Chapter 4. The number of locations examined are 10, 20, 30, and 75, evaluated over 50 random goals datasets.

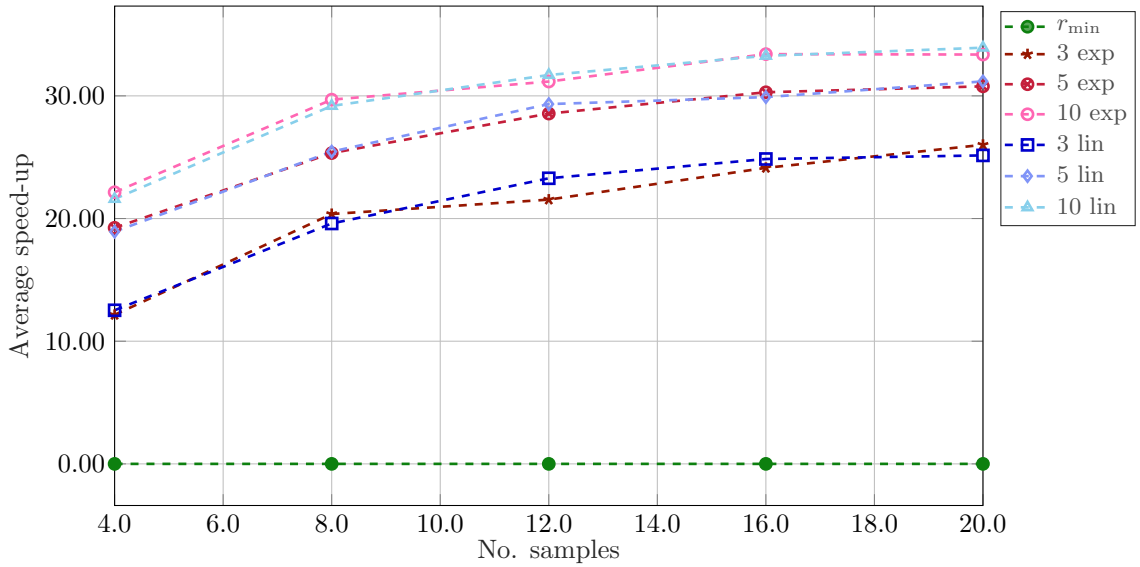


Figure B.2: Speed-up comparison with different heading angles and turning radii samples for 10 randomly generated goals in 50 datasets.

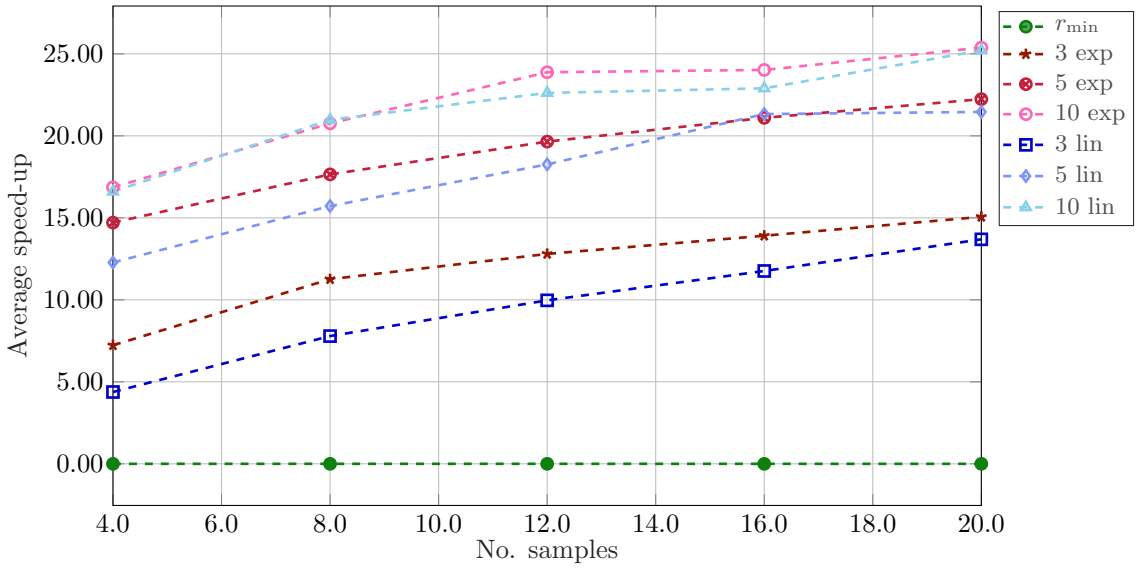


Figure B.3: Speed-up comparison with different heading angles and turning radii samples for 20 randomly generated goals in 50 datasets.

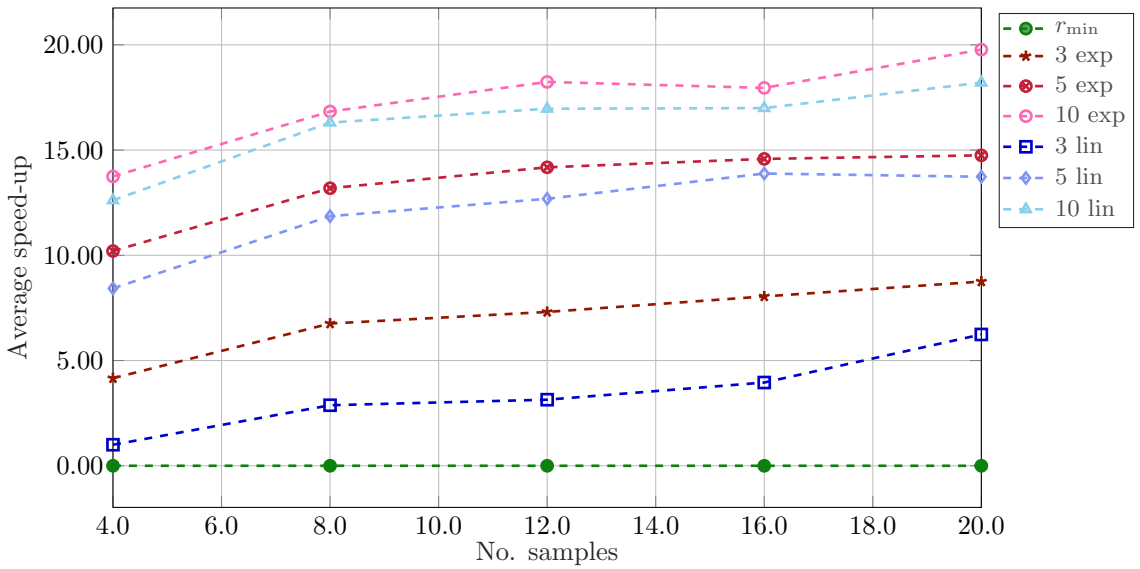


Figure B.4: Speed-up comparison with different heading angles and turning radii samples for 30 randomly generated goals in 50 datasets.

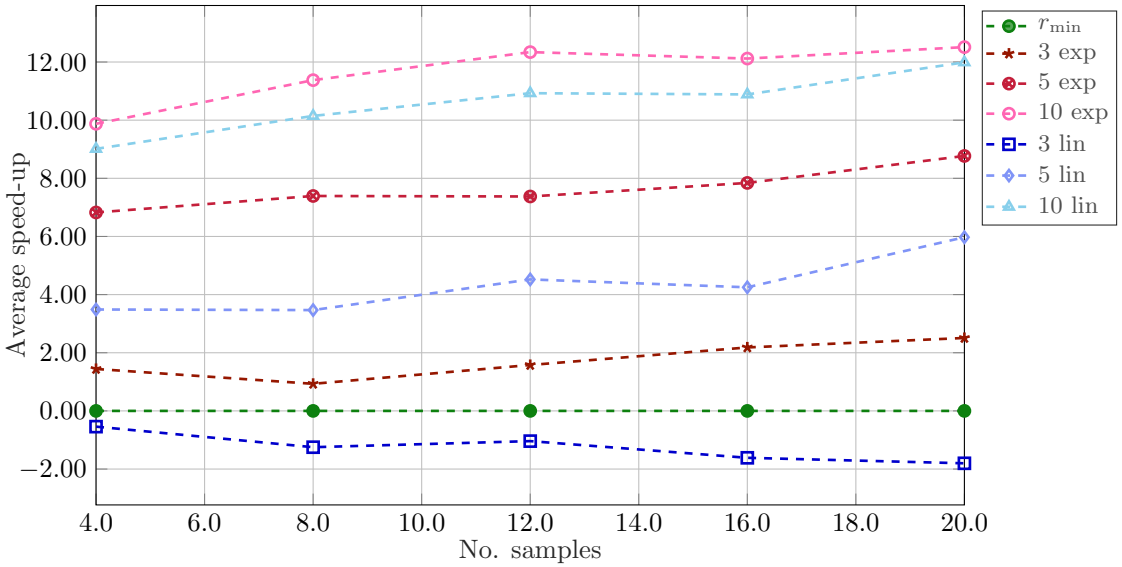


Figure B.5: Speed-up comparison with different heading angles and turning radii samples for 75 randomly generated goals in 50 datasets.

Table B.1: Results for data with 10 goals, averaged.

Num. goals	Radii samples	Angle samples	Fast time	Slow time	Path length	Execute [s]	Speed-up [%]	Speed-up vs. $r_{\min}$ [%]
10	$r_{\min}$	4.00	13.57	15.51	63.94	7.61	14.15	0.00
10	$r_{\min}$	8.00	13.06	15.02	61.92	7.68	14.91	0.00
10	$r_{\min}$	12.00	12.92	14.92	61.49	7.96	15.40	0.00
10	$r_{\min}$	16.00	12.86	14.84	61.16	8.26	15.34	0.00
10	$r_{\min}$	20.00	12.79	14.80	60.99	8.70	15.60	0.00
10	3 exp	4.00	12.10	17.45	71.91	8.01	29.40	12.15
10	3 exp	8.00	10.85	16.69	68.79	9.11	39.41	20.37
10	3 exp	12.00	10.63	16.82	69.32	11.13	41.20	21.54
10	3 exp	16.00	10.36	16.75	69.04	13.63	44.33	24.13
10	3 exp	20.00	10.15	16.81	69.28	16.90	46.78	26.01
10	5 exp	4.00	11.38	17.35	71.49	8.69	37.67	19.24
10	5 exp	8.00	10.42	16.84	69.42	11.69	45.38	25.34
10	5 exp	12.00	10.05	16.51	68.05	16.93	49.73	28.56
10	5 exp	16.00	9.87	16.58	68.35	24.07	51.67	30.29
10	5 exp	20.00	9.78	16.56	68.24	31.86	52.64	30.78
10	10 exp	4.00	11.11	17.53	72.26	11.75	41.11	22.14
10	10 exp	8.00	10.07	16.66	68.65	24.09	50.65	29.69
10	10 exp	12.00	9.85	16.60	68.40	43.53	52.89	31.17
10	10 exp	16.00	9.64	16.49	67.98	69.14	55.38	33.40
10	10 exp	20.00	9.59	16.53	68.12	102.63	55.93	33.37
10	3 lin	4.00	12.06	17.97	74.07	8.05	29.52	12.52
10	3 lin	8.00	10.92	17.70	72.96	9.15	38.40	19.60
10	3 lin	12.00	10.48	17.08	70.37	10.83	43.25	23.28
10	3 lin	16.00	10.30	16.86	69.49	13.39	45.19	24.85
10	3 lin	20.00	10.22	17.26	71.14	16.88	45.74	25.15
10	5 lin	4.00	11.41	17.52	72.20	8.67	37.38	18.93
10	5 lin	8.00	10.41	17.02	70.15	11.79	45.49	25.46
10	5 lin	12.00	9.99	16.46	67.82	16.89	50.58	29.33
10	5 lin	16.00	9.90	16.81	69.30	23.69	50.93	29.90
10	5 lin	20.00	9.75	16.57	68.28	32.31	53.13	31.18
10	10 lin	4.00	11.16	17.55	72.33	11.61	40.26	21.59
10	10 lin	8.00	10.11	17.02	70.16	23.75	49.90	29.18
10	10 lin	12.00	9.81	16.57	68.30	43.21	53.48	31.70
10	10 lin	16.00	9.65	16.56	68.27	70.01	55.21	33.26
10	10 lin	20.00	9.55	16.62	68.48	101.83	56.48	33.93

Table B.2: Results for data with 20 goals, averaged.

Num. goals	Radii samples	Angle samples	Fast time	Slow time	Path length	Execute [s]	Speed-up [%]	Speed-up vs. $r_{\min}$ [%]
20.00	$r_{\min}$	4.00	21.21	23.01	94.84	8.02	8.45	0.00
20.00	$r_{\min}$	8.00	19.66	21.39	88.14	8.65	8.72	0.00
20.00	$r_{\min}$	12.00	19.30	21.09	86.92	9.64	9.20	0.00
20.00	$r_{\min}$	16.00	19.00	20.72	85.40	11.00	8.98	0.00
20.00	$r_{\min}$	20.00	19.02	20.76	85.56	12.89	9.10	0.00
20.00	3 exp	4.00	19.78	25.15	103.63	10.08	16.62	7.23
20.00	3 exp	8.00	17.67	23.52	96.92	15.60	21.37	11.26
20.00	3 exp	12.00	17.11	24.38	100.47	23.45	23.63	12.80
20.00	3 exp	16.00	16.68	23.91	98.53	36.39	24.62	13.91
20.00	3 exp	20.00	16.53	23.75	97.87	51.33	25.94	15.06
20.00	5 exp	4.00	18.49	25.59	105.46	13.06	24.83	14.71
20.00	5 exp	8.00	16.71	24.06	99.18	28.24	28.34	17.65
20.00	5 exp	12.00	16.13	24.31	100.18	51.36	31.10	19.65
20.00	5 exp	16.00	15.69	24.06	99.17	80.28	32.59	21.10
20.00	5 exp	20.00	15.56	24.04	99.10	120.49	33.84	22.24
20.00	10 exp	4.00	18.15	26.50	109.24	27.72	27.11	16.86
20.00	10 exp	8.00	16.28	24.56	101.20	82.78	31.82	20.76
20.00	10 exp	12.00	15.58	24.13	99.44	167.04	35.81	23.88
20.00	10 exp	16.00	15.32	24.20	99.73	290.24	35.72	24.02
20.00	10 exp	20.00	15.17	24.01	98.96	502.02	37.42	25.38
20.00	3 lin	4.00	20.32	26.94	111.03	10.07	13.59	4.38
20.00	3 lin	8.00	18.24	25.86	106.57	15.49	17.52	7.79
20.00	3 lin	12.00	17.55	26.66	109.87	23.55	20.54	9.97
20.00	3 lin	16.00	17.00	26.18	107.90	36.40	22.22	11.76
20.00	3 lin	20.00	16.73	25.93	106.85	50.89	24.43	13.69
20.00	5 lin	4.00	18.89	26.52	109.29	13.17	22.18	12.28
20.00	5 lin	8.00	16.99	25.24	104.02	28.01	26.19	15.72
20.00	5 lin	12.00	16.32	24.75	102.02	50.58	29.73	18.26
20.00	5 lin	16.00	15.66	24.65	101.59	80.02	32.78	21.33
20.00	5 lin	20.00	15.66	24.55	101.19	119.49	32.98	21.46
20.00	10 lin	4.00	18.19	26.24	108.16	28.10	26.79	16.60
20.00	10 lin	8.00	16.25	24.41	100.61	81.90	31.96	20.98
20.00	10 lin	12.00	15.74	24.86	102.45	167.01	34.46	22.62
20.00	10 lin	16.00	15.46	24.57	101.27	297.23	34.53	22.90
20.00	10 lin	20.00	15.19	24.34	100.32	472.14	37.19	25.21

Table B.3: Results for data with 30 goals, averaged.

Num. goals	Radii samples	Angle samples	Fast time	Slow time	Path length	Execute [s]	Speed-up [%]	Speed-up vs. $r_{\min}$ [%]
30	$r_{\min}$	4.00	28.30	30.15	124.28	8.60	6.53	0.00
30	$r_{\min}$	8.00	25.75	27.42	113.02	10.46	6.47	0.00
30	$r_{\min}$	12.00	24.96	26.61	109.66	12.96	6.57	0.00
30	$r_{\min}$	16.00	24.44	25.99	107.12	16.36	6.31	0.00
30	$r_{\min}$	20.00	24.35	25.96	107.00	20.79	6.58	0.00
30	3 exp	4.00	27.17	32.90	135.58	13.78	11.15	4.16
30	3 exp	8.00	24.12	30.56	125.93	28.32	13.83	6.76
30	3 exp	12.00	23.26	30.20	124.47	47.88	14.46	7.31
30	3 exp	16.00	22.62	29.76	122.65	81.02	15.07	8.05
30	3 exp	20.00	22.39	29.24	120.50	113.88	16.08	8.75
30	5 exp	4.00	25.68	33.53	138.18	21.74	17.48	10.20
30	5 exp	8.00	22.75	31.03	127.91	58.34	20.71	13.19
30	5 exp	12.00	21.86	30.48	125.64	112.37	21.81	14.18
30	5 exp	16.00	21.33	30.33	125.01	188.92	21.99	14.58
30	5 exp	20.00	21.22	29.98	123.57	294.81	22.47	14.75
30	10 exp	4.00	24.88	34.14	140.72	55.27	21.37	13.75
30	10 exp	8.00	22.04	31.21	128.63	187.29	24.56	16.83
30	10 exp	12.00	21.11	30.52	125.78	458.51	26.17	18.24
30	10 exp	16.00	20.72	30.91	127.41	876.07	25.58	17.95
30	10 exp	20.00	20.33	29.96	123.47	2276.12	27.81	19.77
30	3 lin	4.00	28.02	34.36	141.61	13.47	7.79	1.00
30	3 lin	8.00	25.03	32.47	133.83	28.31	9.68	2.88
30	3 lin	12.00	24.20	33.15	136.60	48.09	10.00	3.14
30	3 lin	16.00	23.51	33.31	137.26	84.01	10.65	3.96
30	3 lin	20.00	22.92	32.90	135.60	115.62	13.45	6.24
30	5 lin	4.00	26.10	34.92	143.93	21.96	15.66	8.43
30	5 lin	8.00	23.02	31.07	128.06	61.83	19.21	11.86
30	5 lin	12.00	22.15	31.58	130.17	115.79	20.24	12.69
30	5 lin	16.00	21.46	30.34	125.04	193.46	21.23	13.89
30	5 lin	20.00	21.41	31.91	131.50	303.21	21.37	13.73
30	10 lin	4.00	25.13	34.30	141.36	58.15	20.12	12.61
30	10 lin	8.00	22.14	31.74	130.82	190.61	24.01	16.31
30	10 lin	12.00	21.34	31.51	129.87	465.82	24.82	16.96
30	10 lin	16.00	20.89	31.09	128.13	886.50	24.57	16.99
30	10 lin	20.00	20.60	31.49	129.79	1429.35	26.11	18.20

Table B.4: Results for data with 75 goals, averaged.

Num. goals	Radii samples	Angle samples	Fast time	Slow time	Path length	Execute [s]	Speed-up [%]	Speed-up vs. $r_{\min}$ [%]
75	$r_{\min}$	4.00	56.97	59.11	243.60	16.99	3.74	0.00
75	$r_{\min}$	8.00	49.83	51.57	212.54	28.73	3.50	0.00
75	$r_{\min}$	12.00	47.61	49.23	202.90	46.90	3.40	0.00
75	$r_{\min}$	16.00	46.35	47.99	197.76	69.61	3.53	0.00
75	$r_{\min}$	20.00	45.77	47.35	195.14	99.26	3.45	0.00
75	3 exp	4.00	56.16	63.12	260.15	52.61	5.31	1.44
75	3 exp	8.00	49.37	55.45	228.54	159.52	4.54	0.93
75	3 exp	12.00	46.87	53.40	220.09	407.70	5.14	1.58
75	3 exp	16.00	45.36	52.99	218.39	736.04	5.88	2.18
75	3 exp	20.00	44.65	51.99	214.27	992.89	6.17	2.51
75	5 exp	4.00	53.33	64.90	267.47	115.23	10.95	6.83
75	5 exp	8.00	46.40	56.88	234.43	421.40	11.23	7.39
75	5 exp	12.00	44.34	54.67	225.30	1017.66	11.12	7.37
75	5 exp	16.00	42.98	53.86	221.98	1959.65	11.73	7.84
75	5 exp	20.00	42.08	52.72	217.27	3026.79	12.62	8.77
75	10 exp	4.00	51.85	65.84	271.34	413.32	14.12	9.87
75	10 exp	8.00	44.74	57.73	237.92	1740.74	15.35	11.38
75	10 exp	12.00	42.38	56.19	231.57	4213.00	16.28	12.34
75	10 exp	16.00	41.34	54.58	224.93	7550.80	16.22	12.12
75	10 exp	20.00	40.68	53.79	221.68	12064.50	16.58	12.51
75	3 lin	4.00	57.28	62.26	256.59	51.64	3.29	-0.54
75	3 lin	8.00	50.46	55.14	227.26	153.04	2.31	-1.25
75	3 lin	12.00	48.11	55.01	226.71	306.51	2.48	-1.04
75	3 lin	16.00	47.11	54.05	222.74	632.30	1.97	-1.61
75	3 lin	20.00	46.61	56.42	232.53	1014.47	1.71	-1.80
75	5 lin	4.00	55.05	65.79	271.16	116.70	7.49	3.49
75	5 lin	8.00	48.16	57.59	237.33	438.51	7.18	3.47
75	5 lin	12.00	45.55	56.25	231.84	1083.51	8.19	4.52
75	5 lin	16.00	44.46	55.89	230.33	1921.22	8.03	4.25
75	5 lin	20.00	43.19	54.49	224.57	3018.99	9.73	5.97
75	10 lin	4.00	52.26	67.10	276.55	410.83	13.19	9.01
75	10 lin	8.00	45.24	58.71	241.97	1802.73	14.11	10.15
75	10 lin	12.00	42.92	56.12	231.29	4382.61	14.82	10.93
75	10 lin	16.00	41.80	56.32	232.10	7786.35	14.91	10.89
75	10 lin	20.00	40.87	55.44	228.49	12238.91	15.97	11.99