Control Engineering Practice 144 (2024) 105814

Contents lists available at ScienceDirect

Control Engineering Practice

journal homepage: www.elsevier.com/locate/conengprac

A novel approach for 3PDP and real-time via point path planning of Dubins' vehicles in marine applications

Gianfranco Parlangeli, Daniela De Palma *, Rossella Attanasi

Department of Engineering for Innovation, University of Salento, Lecce, Italy

ARTICLE INFO

Keywords:
Dubins vehicle
Marine vehicles
Traveling salesman problems
Path planning

ABSTRACT

This paper addresses the problem of finding the shortest Dubins path between three consecutive via-points with prescribed initial and final orientations and without a prescribed orientation at the intermediate via-point. The problem plays a crucial role for online path planning in many marine applications, as for example, it is instrumental to solve the Dubins Traveling Salesman Problem. A novel solution is proposed using simple tools borrowed from analytic geometry, and an efficient algorithm is presented as a basic routine for real-time path planning algorithms. Extensive simulations confirmed the efficiency of the proposed strategy in terms of both computational complexity and accuracy of the solution. Moreover, a comparative analysis with recent existing approaches is performed showing the effectiveness of the proposed solution.

1. Introduction

The recent advances in technology pushed an increasing use of robotic vehicles in several fields, and an impressive surge of research was devoted in the last years to exploit such an opportunity. More and more efficient and robust algorithms have been developed to face the challenging problems arising from the goal of safe and autonomous navigation, and still many improvements are expected shortly. As for example, collision-free trajectory planning for autonomous vehicles in an only partially known environment is a research topic studied for many years and it is still widely debated (Cheng et al., 2021; Galceran & Carreras, 2013).

Though autonomous vehicles are still not present in the civil society by now, the use of autonomous vehicles has become real in several significant applications, as for example exploration in harsh environments.

A special attention is devoted to the marine environment, for a twofold reason. On one hand, several underwater activities are often required in terms of sub-sea inspection, exploration and maintenance, as detailed after, and on the other hand it is a harsh and hostile environment and sea robots require a special care in terms of robustness and efficiency of hardware and algorithms.

Several activities in hazardous environments that were performed by expert human operators with high risk and effort are more and more executed through the use of autonomous vehicles, as for example ocean and environmental monitoring, wave and energy offshore infrastructures inspection, harbor patrolling, bathymetric data collection, or geophysics and geotechnical surveying (Antonelli et al., 2018; Caillau et al., 2019; Ozog et al., 2016), among the others.

In all these applications, mission success relies strongly on the vehicles ability to accomplish basic motion control problems such as path following and trajectory tracking (Karimi & Lu, 2021). In this context, the path planning problem is in fact a core component to achieve a high level of autonomy. Considering the above applications, due to the high demand for faster and more efficient algorithms needed in real-time applications, real-time path planning is a fundamental research field (Zeng et al., 2015). A kinematic model which has been constantly studied over the years in path planning applications is the so-called Dubins vehicle (Dubins, 1957; LaValle, 2006). Dubins vehicle refers to a vehicle moving only forward at a constant speed with a maximum curvature, and it is well suited to describe the motion of a vehicle without hard braking, slowdown or even stop. Sea robots like marine vessels, underwater or surface vehicles typically have strong limitations in terms of limited mission length or stringent nonholonomic motion constraints. As for example, in marine seismic acquisition, marine vessels town a set of long and fragile underwater cables. Therefore their motion is subject to a maximum curvature constraint of the planned paths, which must be small enough and without stops in order to preserve the integrity of the towed equipment during the turn (Caillau et al., 2019). Most of unmanned underwater vehicles moving at constant depth (Indiveri et al., 2016) are also conveniently treated as Dubins vehicles.

* Corresponding author.
E-mail addresses: gianfranco.parlangeli@unisalento.it (G. Parlangeli), daniela.depalma@unisalento.it (D. De Palma), rossella.attanasi@unisalento.it (R. Attanasi).

https://doi.org/10.1016/j.conengprac.2023.105814
Received 9 May 2023; Received in revised form 26 October 2023; Accepted 2 December 2023
Available online 19 December 2023

Indeed, it is quite common to specify a desired path for marine vessels, underwater or surface vehicles in terms of Dubins path (Caillau et al., 2019; Fossen et al., 2015; Wheare et al., 2019). Dubins path refers to the shortest path between two poses for Dubins vehicles. Given its importance, research on Dubins paths is a long-lasting research theme which is summarized in the following. The interested reader is referred to LaValle (2006) for details.

Since the milestone contribution by Dubins (1957) several research lines have emerged and many aspects of the problem were analyzed over the years such as path smoothness (Boissonant et al., 1994; Fraichard & Scheuer, 2001), vehicles that can also move backwards (Reeds & Shepp, 1990), routing or network optimization problems for Dubins vehicles (Mitchell, 2000; Savla et al., 2008). The first works on shortest path Dubins problems through a sequence of fixed or moving points were presented in Berdyshev (1991) and Berdyshev (2002). Other recent related works that solve generalizations of the Dubins problem through N points are Frego et al. (2020) and Kaya (2019).

The work in Váňa and Faigl (2020) analyzed the Generalized Dubins Interval Problem (GDIP) where the departure and terminal headings of Dubins vehicle are within specified angle intervals. The Dubins Traveling Salesman Problem (DTSP) is related to these latter topics and it was considered in Savla et al. (2008).

This problem has been widely afforded for its relevance in several applications. For example, in water quality monitoring applications, Unmanned Surface vessels (USV) extract water quality samples at multiple target locations in heavily polluted water, avoiding the exposure of humans to risk (Liang et al., 2020). More in general, in a large variety of missions, the vehicles are required to visit a number of predetermined via-points, and perform some tasks at the via-points locations.

Several contributions on this subject (Sadeghi & Smith, 2016) showed that when the order of via-points is fixed, the solution of the DTSP can be decomposed at each step in subproblems known as 3-Point Dubins Problem (3PDP) (Chen & Shima, 2019b).

As already highlighted in Chen and Shima (2019b), the Dubins problem between three consecutive via-points is relevant for several reasons, as for example, on one hand, the solution of 3PDP gives insightful views on the solution paths of the DTSP, as it is a natural extension of the 3PDP, and on the other hand it provides a method for a solution once that a new point is inserted into a mission with minimum additional cost, and the solution to a new 3PDP is required (Chen & Shima, 2019b).

In addition, synthesizing the solution of 3PDP gives insightful views on the solution paths of the DTSP and the CCSPP (Curvature-Constrained Shortest-Path Problem) as they are natural extensions of the 3PDP.

For these reasons, 3PDP recently became a challenging research topic whose goal is the development of ever faster and more efficient methods to solve the problem (Chen & Shima, 2019b; Sadeghi & Smith, 2016). In Sadeghi and Smith (2016) the proposed solution relies on inversive geometry. First, the authors provide a direct way to find an approximate optimal heading at the via-point ("Approximate Method"). Such a solution, although computationally efficient provides only an approximation of the heading. Then, this approximation is exploited to initialize an "Iterative Method" that converges to the optimal heading by successive corrections. This iterative method gives the exact solution of the problem at the cost of a higher computational effort.

The paper (Chen & Shima, 2019b) is a milestone paper on the 3PDP problem, and the resolution methods are based on the Pontryagin principle and optimality criteria. It provides a complete solution which covers all possible cases and the solution is based on the resolution of one-variable Polynomials equations. This approach is able to give an accurate solution. Compared with that approach, the solution achieved in this paper relies on different novel methodologies, namely some optimal properties of conic sections, and the solution can be computed either through a purely graphical way, or analytically, recurring to the matrix description of conic sections. The results and the methods of this paper can be helpful to have a better insight of the geometric features of Dubins' curves.

Nevertheless, these studies have paved the way for recent advances on relaxed versions of the problem (Chen & Shima, 2019a) where the Dubins vehicle moves through 3 consecutive via-points with prescribed orientation only at the initial via-point. In Chen (2020) the tackled problem is the shortest path of a Dubins vehicle from a position with a prescribed heading angle to a target circle with the final heading tangential to the target circle, while in Jha et al. (2020) the shortest Dubins path problem from an initial configuration (position and orientation), via the boundary of an intermediate circle, to a target configuration is investigated.

Motivated by its importance and the need of fast and efficient algorithms required in real-world scenarios, in this paper a novel method to get the solution of the shortest Dubins path between three consecutive via-points is developed using simple tools borrowed from analytic geometry. The basic concept behind this alternative approach was introduced in Parlangeli (2019b) and a preliminary attempt of implementation as an effective algorithm was presented in Parlangeli (2019a). Here the work is extended by providing a detailed proof of the proposed solution, an exhaustive simulation campaign to prove the low computational complexity and the good accuracy of the approach, and a diligent comparison with existing approaches. The ultimate goal of this work is to offer a solution to this problem that is implementable in real-time to allow any possible re-planning in dynamic environments.

The rest of this paper is organized as follows. In Section 2 the addressed 3-Point Dubins Problem is defined in a rigorous manner, after that a solution of the problem is proposed in Section 3 based on simple tools borrowed from analytic geometry. In Section 4 the whole algorithm to be implemented to completely solve the 3-Point Dubins Problem is detailed. Then in Section 5 the performances of the proposed approach are demonstrated through Monte Carlo simulations, they are quantitatively compared with results obtained using existing approaches. Finally, concluding remarks are summarized in Section 6.

2. Problem formulation

Consider a vehicle whose motion in the plane is described by a set of equations as:

$$\begin{cases} \dot{x} = \cos\theta, \\ \dot{y} = \sin\theta, \\ \dot{\theta} = u, \\ \|u\| \leq \Omega; \end{cases} \tag{1}$$

where Ω is a bound on the maximal curvature of the path. Eqs. (1) are the simplest kinematic model of a vehicle moving only forward at a constant speed (assumed, without loss of generality, equal to 1) with a maximum turning radius Ω^{-1}, and they are referred to as the *Dubins' car* (LaValle, 2006). A *feasible* path is a curve in the plane that is viable for the Dubins' vehicle, namely a curve whose maximum curvature along the path is bounded by Ω.

In the following, the state vector $\mathbf{q} = (\mathbf{P}, \theta)$ is referred as to the *pose* or the *configuration* of the vehicle, being $\mathbf{P} = (x, y)$ the position of the vehicle in the Euclidean plane, and $\theta \in [0, 2\pi)$ its orientation; moreover, u denotes the control input (given by its angular speed). Each configuration can be represented graphically using a vector of orientation θ together with two tangent circles of radius Ω^{-1}, on the right and left side of the position $\mathbf{P}$, as in Fig. 1. These circles represent the small-time inaccessible regions for the robot, namely those points out of reach from a given pose $\mathbf{q} = (\mathbf{P}, \theta)$ for any choice of u in a small amount of time, such that $\|u\| \leq \Omega$ (shadowed area in Fig. 1). Thus, each circle corresponds to the sharpest left turn or right turn. They are called (sharpest) *left circle* denoted as C_l and (sharpest) *right circle* C_r,

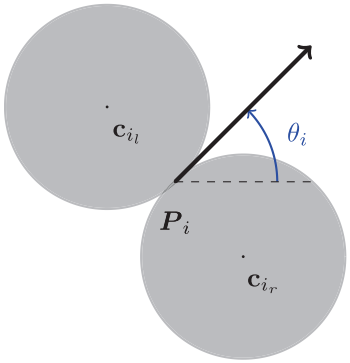

Fig. 1. Graphic interpretation of the left and right centers for an initial configuration $\mathbf{q}_i$.

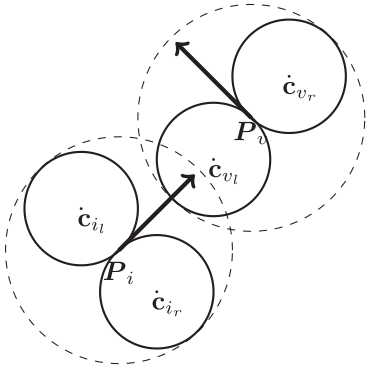

Fig. 2. A sketch of the long distance points hypothesis in Assumption 1.

and their centers are denoted with $\mathbf{c}_l$ and $\mathbf{c}_r$, respectively. An example is illustrated in Fig. 1 where an initial configuration $\mathbf{q}_i = (\mathbf{P}_i, \theta_i)$ is represented using the left circle and right circle having centers in $\mathbf{c}_{i_l}$ and $\mathbf{c}_{i_r}$, respectively.

One main topic on this model is the so-called Dubins Path Problem which consists in finding the shortest curve that connects an initial configuration $\mathbf{q}_i = (\mathbf{P}_i, \theta_i)$ to a final configuration $\mathbf{q}_f = (\mathbf{P}_f, \theta_f)$ for a *Dubins' car*. This problem was firstly solved by Dubins (1957) whose main result is summarized as follows:

Theorem 1. *An optimal path between any two configurations is of type CCC or CSC, or a sub path of a path of either of these two types (where C denotes circular arc, and S denotes straight line segment). Moreover, to be optimal, a CCC path must have its middle arc of length greater than $\pi\Omega^{-1}$ being Ω the maximum admissible curvature.*

Actually, CCC paths do not occur when the points $\mathbf{P}_i$ and $\mathbf{P}_f$ satisfy the condition of void intersection of the four circles $\{C_{i_l} \cup C_{i_r}\} \cap \{C_{f_l} \cup C_{f_r}\} = \emptyset$ (Shkel & Lumelsky, 1996), i.e. black circles in Fig. 2. A sufficient condition usually adopted in the literature (Goaoc et al., 2013; Sadeghi & Smith, 2016; Shkel & Lumelsky, 2001) ensuring that CCC paths do not occur is the void intersection of the dashed larger circles in Fig. 2 . The convenience of using this latter condition, summarized in Assumption 1, is that it is easy to check in practical applications.

Assumption 1 (*Long Distance Points Hypothesis*)**.** Given any two configurations $\mathbf{P}_i$ and $\mathbf{P}_f$, the distance between $\mathbf{P}_i$ and $\mathbf{P}_f$ is larger than $4\Omega^{-1}$.

Now it is possible to properly state the problem considered in this paper.

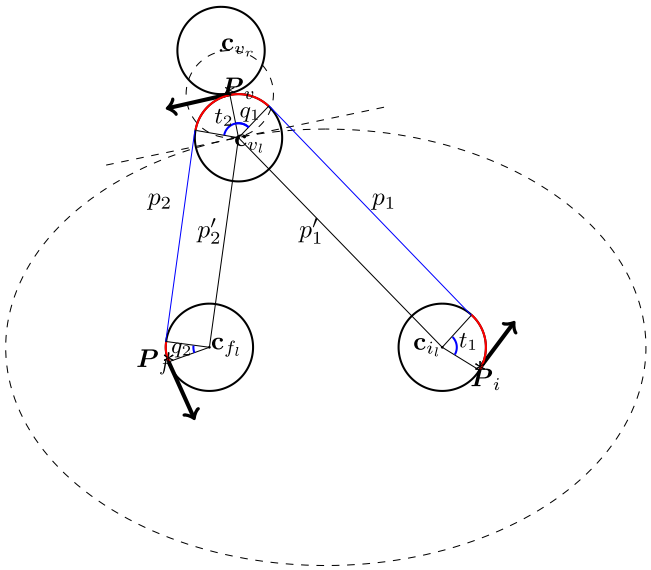

Fig. 3. An example of $LSL - LSL$ path.

Problem 1 (*3PDP – Three-Point Dubins Problem – Dubins Path in Presence of Via Points*)**.** Given the tuple $(\mathbf{q}_i, \mathbf{P}_v, \mathbf{q}_f)$, namely the initial configuration $\mathbf{q}_i = (\mathbf{P}_i, \theta_i)$, the final configuration $\mathbf{q}_f = (\mathbf{P}_f, \theta_f)$ and a pre-assigned via point $\mathbf{P}_v$, where each pair of points $\mathbf{P}_i$, $\mathbf{P}_v$, and also $\mathbf{P}_v$, $\mathbf{P}_f$ satisfies Assumption 1, find the shortest curve of (1) connecting $\mathbf{P}_i$, $\mathbf{P}_v$ and $\mathbf{P}_f$ with tangent direction θ_i at $\mathbf{P}_i$ and θ_f at $\mathbf{P}_f$ among all the curves in the plane with curvature bounded by Ω.

As already stated in the introduction, this problem has been introduced and discussed in the last years for its significance in several practical applications (Chen & Shima, 2019b; Sadeghi & Smith, 2016). It is worth remarking that the stated problem is formulated under Assumption 1, and the cases requiring "CCC" paths are out of the scope of the present paper, thus resulting in mathematically unsolved cases using the proposed approach.

Remark 1. Assumption 1 is a restriction on the choice of the via point by a planner, and indeed it can result as a limitation for some mission types (e.g. rescue, demining). However, for several types when the main goal is the coverage of an area e.g. surveillance, monitoring, sampling, it is reasonable that the planner selects via points ensuring the Assumption without any significant impact or drawback on mission objective. From a mission-oriented standpoint, close via points can be merged into a single representative via point.

According to the Bellman's principle for optimality, the solution of the optimal path for a Dubins' vehicle between three consecutive configurations can be obtained by concatenating two optimal Dubins paths, with orientation angle at the via point θ_v not specified a priori.

Under Assumption 1, each of the basic optimal Dubins paths must be CSC, namely an initial left (denoted in the following L) or right (R) turn, an intermediate straight line motion, and a final left or right turn. In the following t, p, and q denote the length of the initial turn, the straight line segment, and the final turn; the subscript 1 or 2 is used to refer resp. to the path between $\mathbf{P}_i$ to $\mathbf{P}_v$ and from $\mathbf{P}_v$ to $\mathbf{P}_f$ (see Fig. 3). Under Assumption 1, the optimal-length path through three consecutive points is made of two Dubins paths of type CSC, where any circular arc may be eventually of zero degree, hence the solution to Problem 1 must be of a type $CSC - CSC$.

According to Chen and Shima (2019b) and Sadeghi and Smith (2016), the two arcs incident to the via point must have the same turning direction, so the following fact holds.

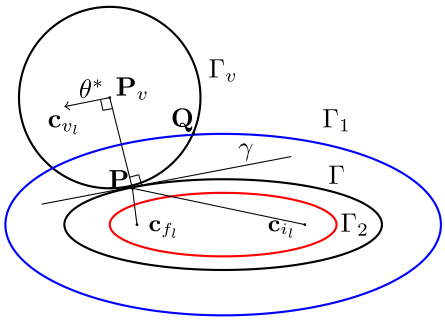

Fig. 4. Geometric interpretation of the solution of the Three-Point Dubins Problem.

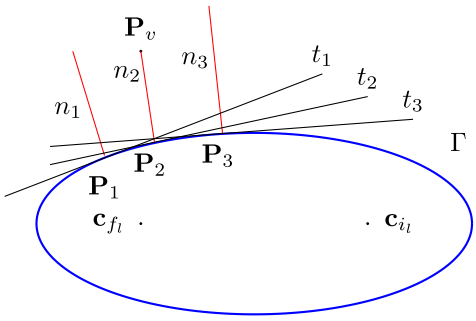

Fig. 5. Graphical representation of the orthogonality condition. Point $\mathbf{P}_2$ satisfies the condition.

Corollary 1. *The optimal solution of Problem 1 under Assumption 1 must belong to* $\mathcal{T} = \{RSR-RSR,\ LSR-RSR,\ RSR-RSL,\ LSR-RSL,\ LSL-LSL,\ RSL-LSL,\ LSL-LSR,\ RSL-LSR\}$*, where any circular arc may possibly be of zero length.*

3. Problem solution

The goal of this section is to solve the problem for each potential Dubins path type in $\mathcal{T}$. Two different subsets of $\mathcal{T}$ are considered:

1. $\mathcal{T}_1 = \{RSR-RSR, LSL-LSL\}$,
2. $\mathcal{T}_2 = \{LSR-RSR, RSR-RSL, LSR-RSL, RSL-LSL, LSL-LSR, RSL-LSR\}$,

with $\mathcal{T} = \mathcal{T}_1 \cup \mathcal{T}_2$. Again, for the sake of conciseness of notation, zero-length circular arcs are implicitly included. The reason for this choice is that such curves do not require a separate study, and they can be investigated both as elements of $\mathcal{T}_1$ and $\mathcal{T}_2$. It is remarkably different from Pontryagin-based approaches, e.g. in Chen and Shima (2019b) these curves are singular cases of the main Theorem and they are investigated separately. In the proposed approach, singular cases are different, and they related to some special geometric configurations (see next Section 3.1.1).

Path types in $\mathcal{T}_1$ are characterized by having all circular arcs in the same turning direction. For each of the Dubins path types in $\mathcal{T}_1$ an efficient approach is provided to determine exactly the optimal orientation θ_v at the via point $\mathbf{P}_v$ corresponding to the optimal path length.

Such an approach is then extended to the path types in $\mathcal{T}_2$ in the second part of the section.

This allows to calculate the length of all potential "optimal candidates" $CSC-CSC$ paths $\in \mathcal{T}$, compare them and choose the shortest one. Alternatively, one can immediately select the shortest path type using the classification approach illustrated in De Palma and Parlangeli (2022) and compute only its length and its optimal heading at the via-point using the approaches here presented.

3.1. Solution for path types in $\mathcal{T}_1$

In the following the analytical solution is derived.

The next proposition lays the theoretical foundations for the proposed algorithm for paths in $\mathcal{T}_1$.

Proposition 1. *Consider paths whose type is in* $\mathcal{T}_1$*. Let* $\mathbf{P}_v$ *be the preassigned via-point,* $\mathbf{q}_i$ *and* $\mathbf{q}_f$ *the initial and final configurations, and finally let* $\mathbf{c}_{i_l}$ $(\mathbf{c}_{i_r})$ *and* $\mathbf{c}_{f_l}$ $(\mathbf{c}_{f_r})$ *denote the center of the left (right) initial and final circles. Problem 1 can be fully solved by finding the ellipse having* $\mathbf{c}_{i_l}$ $(\mathbf{c}_{i_r})$ *and* $\mathbf{c}_{f_l}$ $(\mathbf{c}_{f_r})$ *as foci and tangent to the circle centered in* $\mathbf{P}_v$ *with radius* Ω^{-1}*. In details:*

- *s1. the angle* θ_v^* *of the optimal path crossing* $\mathbf{P}_v$ *is equal to the angle of the common tangent line between such an ellipse and the circle centered in* $\mathbf{P}_v$ *with radius* Ω^{-1}*,*
- *s2. the slope of* p_1 *is equal to the slope of the line* p_1' *connecting* $\mathbf{c}_{i_l}$ $(\mathbf{c}_{i_r})$ *to the tangent point between the ellipse and the circle,*
- *s3. the slope of* p_2 *is equal to the slope of the line* p_2' *connecting* $\mathbf{c}_{f_l}$ $(\mathbf{c}_{f_r})$ *to the tangent point between the ellipse and the circle.*

Proof. Without loss of generality, consider Fig. 3 representing a generic $LSL-LSL$ path, the problem is to find the orientation angle in $\mathbf{P}_v$ in order to have the shortest length of the path $\mathbf{P}_i$–$\mathbf{P}_v$–$\mathbf{P}_f$ or, equivalently, to find the angular position in $\mathbf{c}_{v_l}$. Notice that $\mathbf{c}_{v_l}$ lies on a circle centered in $\mathbf{P}_v$ and radius equal to the minimum curvature radius, and its angular position is $\frac{\pi}{2} + \theta_v$, where θ_v is the orientation angle in $\mathbf{P}_v$.

The Dubins path connecting $\mathbf{q}_i$ and $\mathbf{q}_f$ through $\mathbf{P}_v$ is made of two elementary Dubins paths, each depending on θ_v only, so that it is necessary to look for:

$$\theta_v^* = \underset{\theta_v \in [0,2\pi)}{\arg\min}\{L_1(\theta_v) + L_2(\theta_v)\},$$

where $L_1(\theta_v)$ and $L_2(\theta_v)$ are the lengths of each elementary path. However, it is easily verified that the length of the whole circular motion, namely $t_1+q_1+t_2+q_2$ in Fig. 3, does not depend on θ_v and it is always equal to the angular displacement between $\mathbf{q}_i$ and $\mathbf{q}_f$ (and it is always equal to either $\theta_f - \theta_i$ or $\theta_f - \theta_i + 2\pi$). This allows us to reduce the problem to:

$$\theta_v^* = \underset{\theta_v \in [0,2\pi)}{\arg\min}\{p_1 + p_2\}.$$

Moreover $p_1 = p_1'$, where p_1' is the distance of $\mathbf{c}_{i_l}$ from $\mathbf{c}_{v_l}$, and $p_2 = p_2'$, where p_2' is the distance from $\mathbf{c}_{v_l}$ to the $\mathbf{c}_{f_l}$ because they are opposite borders of a rectangle.

Considering all the previous facts, the problem can be finally rearranged as the search of the point $\mathbf{c}_{v_l}$ placed at a prescribed distance from $\mathbf{P}_v$ and with minimum total length from two (fixed) points $\mathbf{c}_{i_l}$ and $\mathbf{c}_{f_l}$. That is, with reference to Fig. 3, to find the point $\mathbf{c}_{v_l}$ lying on a circle centered in $\mathbf{P}_v$ and fixed radius which corresponds to the least distance $p_1' + p_2'$.

To solve the problem described above, consider the geometric procedure as follows. Take a point $\mathbf{Q}$ distant Ω^{-1} from $\mathbf{P}_v$, build the (unique) ellipse having $\mathbf{c}_{i_l}$ and $\mathbf{c}_{f_l}$ as foci and passing through $\mathbf{Q}$ (that is, the ellipse Γ_1 in Fig. 4), and build the circle Γ_v centered in $\mathbf{P}_v$ with radius Ω^{-1}. All the points of Γ_v inside the ellipse Γ_1 are distant Ω^{-1} from $\mathbf{P}_v$ and have lengths sum from the two foci less than $\mathbf{Q}$. The only case where such points inside the ellipse are not present is when the ellipse is tangent to the circle Γ_v (as for Γ in Fig. 4). In this situation, the only point $\mathbf{P}$ belonging to the ellipse is the one corresponding to the least distance $p_1' + p_2'$. In fact, all the ellipses corresponding to inferior values of the sum of the lengths from the two foci are distant from $\mathbf{P}_v$ strictly more than Ω^{-1} (e.g. Γ_2 in Fig. 4).

Finally, to show that the problem is completely solved once that such an ellipse is found, consider that:

s1. the orientation angle of the optimal trajectory is equal to the slope of the common tangent line (line γ in Fig. 4) because they both are orthogonal to the radius $\mathbf{P}_v - \mathbf{c}_{v_l}$,

s2. the line segment p_1 of the $LSL - LSL$ optimal path have slope equal to the line connecting $\mathbf{c}_{i_l}$ to the tangent point,

s3. the line segment p_2 of the $LSL - LSL$ optimal path have slope equal to the line connecting the tangent point to $\mathbf{c}_{f_l}$. □

The most important result of Proposition 1 is that the optimization problem is rearranged in terms of analytic geometry, and a number of classical results can be applied to get the solution in a computationally convenient form using graphical or analytical tools. Moreover, some recent advances of this problem achieved in the last years can be derived as corollaries using the above reformulation. The following result was first derived in Berdyshev (2002). It was later independently conjectured in Parlangeli et al. (2009), proved in the main result of Goaoc et al. (2013) and strongly considered in the work (Sadeghi & Smith, 2016).

Corollary 2. *The length optimal path has the following property. The circular segment of the first sub path reaching the via point has the same length of the circular segment of the second sub path leaving the via point (i.e. $q_1 = t_2$ in Fig. 3).*

Proof. This result is a straight consequence of the property of a tangent line to an ellipse (see Appendix). □

Now a solution to Problem 1 for paths in $\mathcal{T}_1$ is provided. For the ease of presentation, $\Omega = 1$, however it is no loss of generality to restate the problem using a proper scaling factor (Caillau et al., 2019). The following result provides the general solution under a technical hypothesis (i.e. $x_0 y_v - y_0 x_v \neq 0$), however the cases when $x_0 y_v - y_0 x_v = 0$ are studied in the next section.

Proposition 2 (*Analytical Solution to Problem 1*)**.** *Consider the ellipse Γ having $\mathbf{c}_{i_l}$ ($\mathbf{c}_{i_r}$) and $\mathbf{c}_{f_l}$ ($\mathbf{c}_{f_r}$) as foci and tangent to the circle Γ_v centered in $\mathbf{P}_v$ with radius Ω^{-1}. There exists a coordinate system $\mathcal{F}$ such that, denoting by (x_0, y_0) the coordinates in $\mathcal{F}$ of the tangent point $\mathbf{P}$ between Γ_v and Γ, by $2c$ the distance between the foci of Γ, and by b the length of its semi-minor axis, the solution to Problem 1 for a path type in $\mathcal{T}_1$ is given by:*

s1. the tangent line γ between the circle Γ_v and the ellipse Γ:

$$y = -\frac{b^2}{b^2 + c^2}\frac{x_0}{y_0}x + \frac{b^2}{y_0},$$

so that, the optimal direction θ_v^* at the via point $\mathbf{P}_v$ is such that:

$$\tan(\theta_v^*) = -\frac{b^2}{b^2 + c^2}\frac{x_0}{y_0},$$

s2. the line p_1':

$$y = x\frac{y_0}{x_0 - c} + \frac{y_0 c}{x_0 - c},$$

s3. the line p_2':

$$y = x\frac{y_0}{x_0 + c} + \frac{y_0 c}{x_0 + c}.$$

Proof. Assume to make a roto-translation of coordinates to put the ellipse equation in canonical form, so that the center of the initial left (right) circle $\mathbf{c}_{i_l}$ ($\mathbf{c}_{i_r}$) has coordinate $(+c, 0)$ and the center of the final left (right) circle $\mathbf{c}_{f_l}$ ($\mathbf{c}_{f_r}$) has coordinate $(-c, 0)$. In this coordinate frame $\mathcal{F}$ an ellipse curve is expressed as:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1, \tag{2}$$

with $a^2 = b^2 + c^2$, being $(c, 0)$ and $(-c, 0)$ the coordinates of the foci lying on the x-axis. The coordinates of the via point in $\mathcal{F}$ are denoted by $\mathbf{P}_v = (x_v, y_v)$, and by $\mathbf{P} = (x_0, y_0)$ the tangent point between the circle Γ_v and the ellipse Γ having foci in $\mathbf{c}_{i_l}$ ($\mathbf{c}_{i_r}$) and $\mathbf{c}_{f_l}$ ($\mathbf{c}_{f_r}$) described by Eq. (2) (see Fig. 4). In the following the values of x_0, y_0, a, b are derived, provided that c, x_v, y_v are assigned. Since $\mathbf{P}$ belongs to the ellipse and $a^2 = b^2 + c^2$, then:

$$\frac{x_0^2}{b^2 + c^2} + \frac{y_0^2}{b^2} = 1. \tag{3}$$

The tangent line at $\mathbf{P}$ is (see Appendix):

$$y = -\frac{b^2 x_0}{a^2 y_0}x + \frac{b^2}{y_0}. \tag{4}$$

A convenient way to impose the tangency condition is that the line passing through $\mathbf{P} = (x_0, y_0)$ and orthogonal with respect to the tangent in $\mathbf{P}$ with equation:

$$y = y_0 + \frac{a^2 y_0}{b^2 x_0}(x - x_0), \tag{5}$$

must pass through $\mathbf{P}_v$. This condition is graphically represented in Fig. 5, where three different points are considered, the only satisfying the orthogonality condition is $\mathbf{P}_2$ because the normal to the ellipse at this point crosses $\mathbf{P}_v$. This condition gives the following relation:

$$x_0 y_0 = \frac{b^2}{c^2}x_0 y_v - \frac{a^2}{c^2}y_0 x_v. \tag{6}$$

It is now useful to merge relation (3) and (6). Rewriting (6) as $c^2 y_0(x_0 + x_v) + b^2(x_0 y_v - y_0 x_v) = 0$ and considering the case of $x_0 y_v - y_0 x_v \neq 0$, it is possible to put it into (3) substituting the term b^2 and after some elaboration finally get:

$$\frac{x_0}{y_0 + y_v} + \frac{y_0}{x_0 + x_v} = \frac{c^2}{y_0 x_v - x_0 y_v}. \tag{7}$$

Eq. (7) represents the locus of (x_0, y_0) as tangency points of a circle centered in $\mathbf{P}_v = (x_v, y_v)$ for any value of the radius. The case $x_0 y_v - y_0 x_v = 0$ gives rise to several singular cases which are treated separately in Section 3.1.1. The last condition to impose is that the distance between $\mathbf{P}$ and $\mathbf{P}_v$ is 1:

$$(x_0 - x_v)^2 + (y_0 - y_v)^2 = 1. \tag{8}$$

From (7) and (8) it is possible to get the following equation:

$$x_0 y_0(x_v^2 + y_v^2 - c^2) - x_v y_v(x_0^2 + y_0^2 + c^2) = y_0 x_v(1 - c^2) - y_v x_0(1 + c^2). \tag{9}$$

Eq. (8)–(9) are two quadratic equations in the two unknown parameters x_0 and y_0. Thus, the coordinates of the tangent point $\mathbf{P}$ are finally found by resorting to the analytic geometry tools on conic sections (Swokowski, 1979) to solve the system of Eqs. (8)–(9). Using the matrix representation of conic sections (see Appendix), Eq. (8) can be written as a quadratic form with matrix:

$$\mathbf{A}_1 = \begin{bmatrix} 1 & 0 & -x_v \\ 0 & 1 & -y_v \\ -x_v & -y_v & x_v^2 + y_v^2 - 1 \end{bmatrix}; \tag{10}$$

and Eq. (9):

$$\mathbf{A}_2 = \begin{bmatrix} -x_v y_v & \frac{1}{2}(x_v^2 + y_v^2 - c^2) & \frac{1}{2}y_v(1 + c^2) \\ \frac{1}{2}(x_v^2 + y_v^2 - c^2) & -x_v y_v & -\frac{1}{2}x_v(1 - c^2) \\ \frac{1}{2}y_v(1 + c^2) & -\frac{1}{2}x_v(1 - c^2) & -x_v y_v c^2 \end{bmatrix}. \tag{11}$$

The intersection between the two conics $\mathbf{A}_1$ and $\mathbf{A}_2$ allows us to find x_0 and y_0 corresponding to the optimal solution. The intersection points between two conics can be at most 4, in the considered case 2 distinct real solutions are expected, corresponding to the situations with the circle ($\mathbf{A}_1$) internally or externally tangent to the ellipse ($\mathbf{A}_2$), as depicted in Fig. 6, so that point $\mathbf{P}$ can be easily found either graphically, or analytically as detailed later in Section 4.2.

Once that point $\mathbf{P} = (x_0, y_0)$ is determined, it is possible to compute the parameter b as:

$$b^2 = c^2\frac{y_0(x_v - x_0)}{x_0 y_v - y_0 x_v}. \tag{12}$$

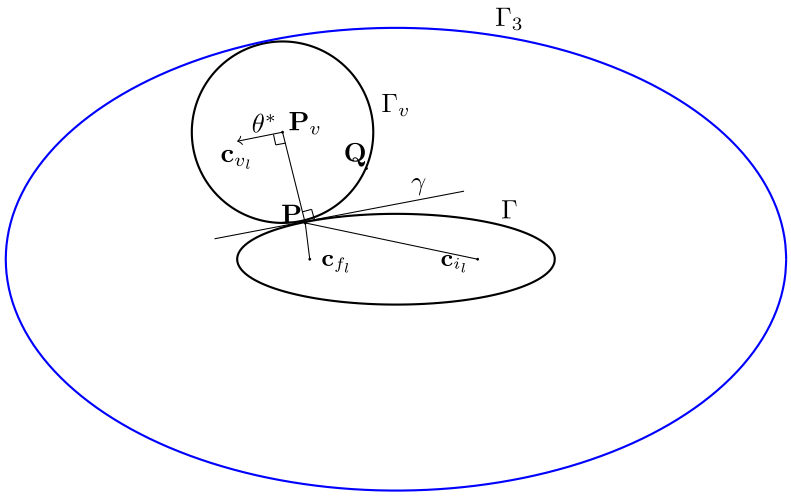

Fig. 6. The circle Γ_v internally or externally tangent to two ellipses.

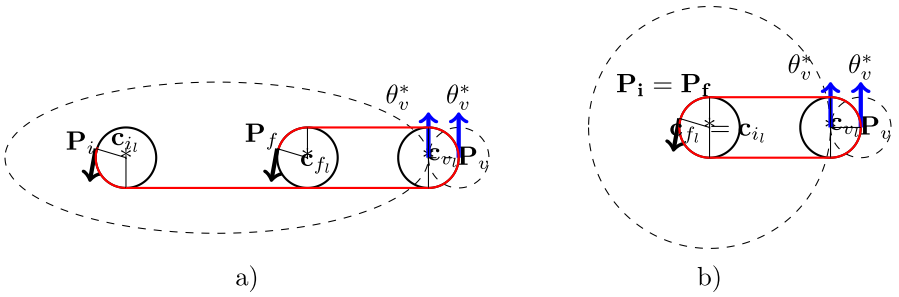

Fig. 7. Singular cases (i): (a) Centers aligned with the via point, with $\mathbf{c}_{i_l} \neq \mathbf{c}_{f_l}$. (b) Centers aligned with the via point, $\mathbf{c}_{i_l} = \mathbf{c}_{f_l}$.

Based on these results, the complete solution is found:

s1. the tangent line γ, whose slope is the optimal direction θ_v^* at the via point $\mathbf{P}_v$:

$$y = -\frac{b^2}{b^2 + c^2}\frac{x_0}{y_0}x + \frac{b^2}{y_0}, \tag{13}$$

s2. the line p_1' connecting $\mathbf{c}_{i_l}$ ($\mathbf{c}_{i_r}$) to $\mathbf{c}_{v_l}$ ($\mathbf{c}_{v_r}$):

$$y = x\frac{y_0}{x_0 - c} + \frac{y_0 c}{x_0 - c}, \tag{14}$$

s3. the line p_2' connecting $\mathbf{c}_{v_l}$ ($\mathbf{c}_{v_r}$) to $\mathbf{c}_{f_l}$ ($\mathbf{c}_{f_r}$):

$$y = x\frac{y_0}{x_0 + c} + \frac{y_0 c}{x_0 + c}. \tag{15}$$

The relations (13)–(14)–(15) complete the problem solution in the case of $LSL - LSL$ (or $RSR - RSR$) paths. It is worth remembering that the solution is expressed in the coordinate frame $\mathcal{F}$, so the inverse roto-translation is required to come back to the original coordinate frame. □

3.1.1. Singular cases (i.e. $x_0 y_v - y_0 x_v = 0$)

Now the situations related to the singular cases $x_0 y_v - y_0 x_v = 0$ are considered. There are two possible conditions leading to this singularity, namely when $\mathbf{c}_{i_l}$, $\mathbf{c}_{f_l}$ and $\mathbf{P}_v$ are aligned, and when $\mathbf{P}_v$ lies on the orthogonal line with respect to the line connecting $\mathbf{c}_{i_l}$ and $\mathbf{c}_{f_l}$.

(i) The first singular case is when $\mathbf{c}_{i_l}$, $\mathbf{c}_{f_l}$ and $\mathbf{P}_v$ are aligned. In this case the coordinates in frame $\mathcal{F}$ of $\mathbf{P}$ and $\mathbf{P}_v$ are characterized by having $y_0 = y_v = 0$. In particular, the following two situations can occur.

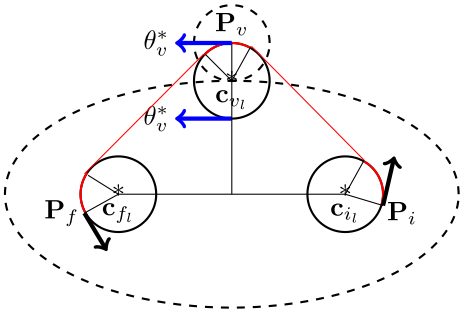

Fig. 8. Singular cases: (ii) Via point on the orthogonal line to the axis of the two centers.

(a) The situation in which $\mathbf{c}_{i_l} \neq \mathbf{c}_{f_l}$ (Fig. 7 (a)). In this cases the tangent ellipse has the semi-major axis $a = d_{\mathbf{P}_v} - \Omega^{-1}$, where $d_{\mathbf{P}_v}$ is the distance from the midpoint between $\mathbf{c}_{i_l}$ and $\mathbf{c}_{f_l}$, and $\mathbf{P}_v$. The value of θ_v^* is the orthogonal direction with respect to the line connecting $\mathbf{c}_{i_l}$, $\mathbf{c}_{f_l}$ and $\mathbf{P}_v$.

(b) The second situation is drawn in Fig. 7 (b), where $\mathbf{c}_{i_l} = \mathbf{c}_{f_l}$. It is a special case of the above example, and in such case the ellipse is a circle. This case is reported in order to make the present work complete, but in the simulations this particular case does not occur.

(ii) The last described situation is the case with $\mathbf{P}_v$ on the orthogonal line with respect to the line connecting $\mathbf{c}_{i_l}$ and $\mathbf{c}_{f_l}$, and crossing their midpoint (see Fig. 8). In this case the coordinates in frame $\mathcal{F}$ of $\mathbf{P}$ and $\mathbf{P}_v$ are characterized by having $x_0 = x_v = 0$. It is easy to see that the semi-minor axis $b = y_v - \Omega^{-1}$, so the solution is

completely determined (i.e. $\mathbf{c}_{v_l} = (x_v, y_v - \Omega^{-1})$), and θ_v^* is in the same direction of the line connecting $\mathbf{c}_{i_l}$ and $\mathbf{c}_{f_l}$.

3.2. Solution for path types in $\mathcal{T}_2$

In this section the approach for $\mathcal{T}_1$ path type to those in $\mathcal{T}_2$ is extended, where an elementary Dubins path of the 3PDP has a change in the curvature (t_1 has opposite sign w.r.t. q_1 and/or t_2 has opposite sign w.r.t q_2).

Without loss of generality, consider a $RSL-LSL$ path as depicted in Fig. 9. In order to compute the length of the RSL portion ($\mathbf{P}_i - \mathbf{P}_v$), it is possible to resort to the construction of the isometric LSL path ($\mathbf{P}'_i - \mathbf{P}_v$) adopted in Bui et al. (1994), translating the problem to a path type $LSL-LSL \in \mathcal{T}_1$, and thus falling into the cases considered in Section 3.1. The construction of isometric paths has been introduced and already used in Bui et al. (1994) to solve the synthesis problem related to elementary Dubins paths. The construction of the isometric path requires a reasonable value for θ_v and $\bar{\theta}_v$ is selected as the mid value of the interval where θ_v belongs to. Fig. 10 shows the two boundary cases of zero-length of circular arc at $\mathbf{P}_v$. In particular, the blue path is built by imposing the first arc q_1 at $\mathbf{P}_v$ of zero length (i.e. $q_1 = 0$), the corresponding value of θ_v is denoted by θ_{1_0}. The red path corresponds to zero-length of the ending circular arc at $\mathbf{P}_v$ (i.e. $t_2 = 0$), the corresponding value of θ_v is denoted by θ_{2_0}. Since the optimal path cannot have a change of curvature at the via point, the range where the optimal value θ_v belongs to, is between θ_{1_0} and θ_{2_0} (refer to Fig. 11). The values θ_{1_0} and θ_{2_0} can be easily computed either graphically or analytically. Indeed, the value θ_{1_0} corresponds to the degenerate first elementary Dubins path RSL into a path of type RS, while the value θ_{2_0} corresponds to the degenerate second elementary Dubins path LSL into a path of type SL. Therefore, θ_{1_0} is the slope of the line passing through $\mathbf{P}_v$ and tangent to the right circle C_{ir} of $\mathbf{P}_i$, while θ_{2_0} is the slope of the line passing through $\mathbf{P}_v$ and tangent to the left circle C_{fl} of $\mathbf{P}_f$. Summarizing, with reference to the $RSL-LSL$ path considered in Fig. 9, the steps to follow in order to construct the isometric $LSL-LSL$ path are:

1. Find the direction of θ_{1_0}, that is find the two lines passing through $\mathbf{P}_v$ and distant Ω^{-1} from $\mathbf{c}_{i_r}$, and select the one with the orientation θ_i at $\mathbf{P}_i$.
2. Do the same to find θ_{2_0}, considering $\mathbf{c}_{f_l}$ instead of $\mathbf{c}_{i_r}$.
3. Define:
$$\bar{\theta}_v = \frac{\theta_{1_0} + \theta_{2_0}}{\|\theta_{1_0} + \theta_{2_0}\|}, \tag{16}$$
if the orientations are expressed as unit vectors, or,
$$\bar{\theta}_v = \frac{\theta_{1_0} + \theta_{2_0}}{2}, \tag{17}$$
if the orientations are expressed as angles with respect to the horizontal line.
4. Find the left and the right circles of $\mathbf{P}_v$ with the direction of $\bar{\theta}_v$; if it is expressed as an angle $\mathbf{c}_{v_l} = \left(x_v + \cos(\bar{\theta}_v + \frac{\pi}{2}), y_v + \sin(\bar{\theta}_v + \frac{\pi}{2})\right)$, or if $\bar{\theta}_v$ is expressed as a vector it is possible to use rotation matrices.
5. Calculate the tangent line to the right circle C_{ir} related to $\mathbf{P}_i$ and to the left circle C_{vl} related to $\mathbf{P}_v$. As represented in Fig. 12, there are four common tangent lines, i.e. two cross tangents, namely the lines that intersect each other in the middle point between $\mathbf{c}_{i_r}$ and $\mathbf{c}_{v_l}$, and two external tangents, the lines parallel to the line connecting $\mathbf{c}_{i_r}$ and $\mathbf{c}_{v_l}$. However, only one of these tangents is consistent with the orientation θ_i at $\mathbf{P}_i$ and the orientation $\bar{\theta}_v$ at $\mathbf{P}_v$, as the solid red line in the example of Fig. 12.
6. Finally, calculate $\mathbf{c}'_{i_l}$ by imposing that the tangent point is the middle point between $\mathbf{c}'_{i_l}$ and $\mathbf{c}_{i_r}$.

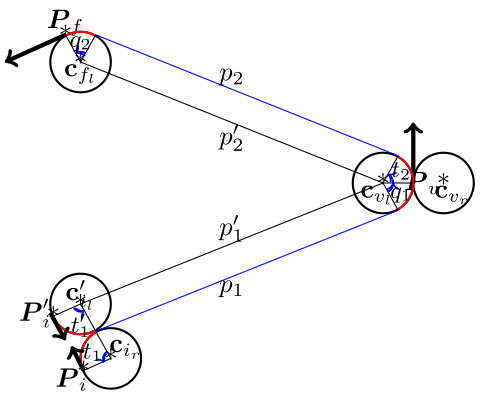

Fig. 9. An example of $RSL-LSL$ path and its isometric $LSL-LSL$ path.

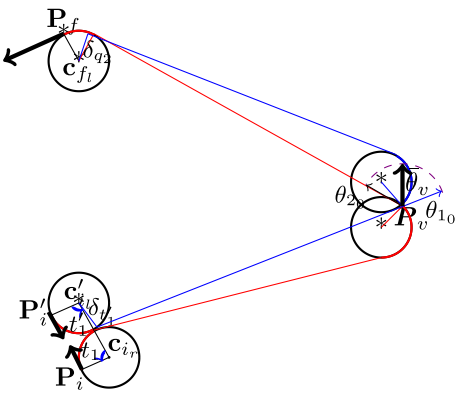

Fig. 10. Graphical representation of the admissible values of θ corresponding to the optimal path.

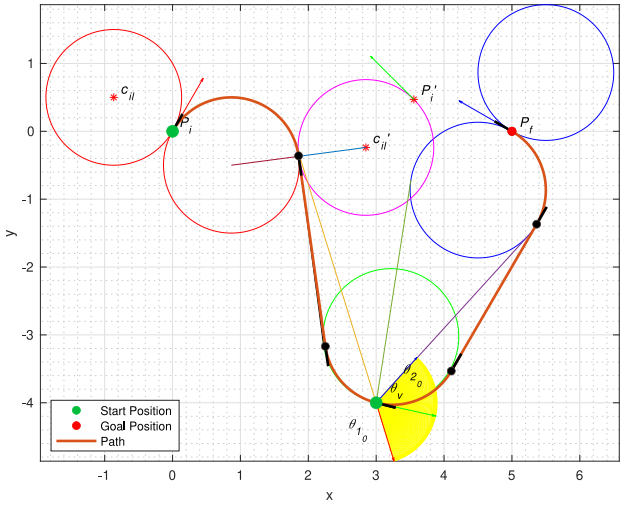

Fig. 11. In this figure it is shown a $RSL-LSL$ path choosing θ_v, the orientation in P_v, as the middle value between θ_{1_0} and θ_{2_0}, as reported in Eq. (16) or equivalently in (17).

Using this procedure, it is possible to recast each path type in $\mathcal{T}_2$ into an isometric path of a type in $\mathcal{T}_1$, and so use the approach in Section 3.1 to all paths.

Remark 2 (*Admissible Intervals in the 4-PDP*)**.** The procedure of building the admissible heading intervals where the solution belongs to, as depicted in Fig. 10, can be easily extended, both graphically and

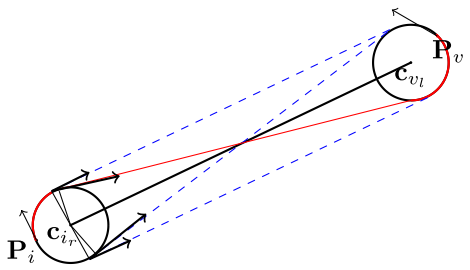

Fig. 12. Sketch of the common tangents between the circles C_{i_r} and C_{v_l}.

analytically, to a larger number of intermediate via points, as depicted in Fig. 13. This procedure is helpful to easily compute the mid value as a convenient initial guess of the optimal heading, which is of great importance in real world applications. Indeed, it is well known that the N-PDP problem has high oscillation nature and it suffers from many local minima (Frego et al., 2020; Saccon et al., 2021).

4. Algorithm implementation

In this Section the mathematical results achieved so far are translated into an algorithm. A block scheme of the algorithm is sketched in Algorithm 1, where the two procedures **ComputationTangentPoint** and **ComputationIsometricPath** are detailed in the successive subsections.

The output of the algorithm is the Dubins path type with the shortest length and the related optimal orientation at the via-point $\mathbf{P}_v$.

Algorithm 1 3PDP - Three point Dubins' problem

Require: $\mathbf{q}_i$, $\mathbf{q}_f$, $\mathbf{P}_v$
Ensure: θ_v^*
1: **for all** $p \in \mathcal{T}_1$ **do**
2: Ensure $\mathbf{q}_i$, $\mathbf{q}_f$, $\mathbf{P}_v$ are expressed in $\mathcal{F}$
3: Compute $\mathbf{c}_{i_l}$, $\mathbf{c}_{i_r}$, $\mathbf{c}_{f_l}$, $\mathbf{c}_{f_r}$
4: Compute $\mathbf{A}_1$ and $\mathbf{A}_2$ using Eq. (10)–(11)
5: $\mathbf{P} \leftarrow$ **ComputationTangentPoint**($\mathbf{P}_v$, $\mathbf{A}_1$, $\mathbf{A}_2$)
6: Compute b using Eq. (12)
7: Compute θ_v through Eq. (13)
8: Compute p_1' using Eq. (14)
9: Compute p_2' using Eq. (15)
10: Compute $L_p = L_1(\theta_v) + L_2(\theta_v)$ {▷ length of path $p \in \mathcal{T}_1$}
11: **end for**
12: **for all** $p \in \mathcal{T}_2$ **do**
13: $\mathbf{q}_i'$, $\mathbf{q}_f'$ $\leftarrow$ **ComputationIsometricPath**($\mathbf{q}_i$, $\mathbf{q}_f$, $\mathbf{P}_v$, $p \in \mathcal{T}_2$) {▷ Compute $\mathbf{q}_i'$, $\mathbf{q}_f'$ of the isometric path $p' \in \mathcal{T}_1$}
14: Repeat lines 2–10 for the isometric path $p' \in \mathcal{T}_1$
15: **end for**
16: Compute

$$p^* = \underset{(p \in \mathcal{T})}{\operatorname{argmin}} L_p \qquad \{\triangleright \text{ path with minimum length}\}$$

17: **return** θ_v^* corresponding to the path p^*

4.1. An illustrative example

In this subsection it is shown in detail how Algorithm 1 works through a numerical simulation.

The required variables $\mathbf{q}_i$ i.e. the starting configuration, the final one $\mathbf{q}_f$ and the via point $\mathbf{P}_v$ let the algorithm start following an $LSL-LSL$ path. In this example $\mathbf{q}_i = (7.24, 4.75, 0.95)$, $\mathbf{q}_f = (5.97, 0.67, 0.63)$ and $\mathbf{P}_v = (0.73, 1.99)$ and the tangent left circles to the initial and final orientation are computed in the original frame $\mathcal{O}$, as depicted in the top left corner of Fig. 14. Then $\mathbf{q}_i$, $\mathbf{q}_f$ and $\mathbf{P}_v$ are put in the new frame $\mathcal{F}$, that is the reference system in which the left centers $\mathbf{c}_{i_l}$ and $\mathbf{c}_{f_l}$ are aligned and symmetric with respect to the origin, see the top right corner in Fig. 14. The circle centered in $\mathbf{P}_v$ with radius equal to 1 is computed ($\mathbf{A}_1$ in Algorithm 1) and then the ellipse tangent to this circle with $\mathbf{c}_{i_l}$ and $\mathbf{c}_{f_l}$ as foci is depicted ($\mathbf{A}_2$ in Algorithm 1), as it is possible to see in the lower left corner of Fig. 14. In this simulation:

$$\mathbf{A}_1 = \begin{bmatrix} 1 & 0 & 2.72 \\ 0 & 1 & -4.62 \\ 2.72 & -4.62 & 27.78 \end{bmatrix}, \quad \mathbf{A}_2 = \begin{bmatrix} 12.58 & 12.4 & 11.52 \\ 12.4 & 12.58 & -4.05 \\ 11.52 & -4.05 & 50.07 \end{bmatrix}.$$

The coordinates of the tangent point are $\mathbf{P} = [-2.27, 3.73]$, $b = 7.25$, $\vartheta_v = 4.91\,\text{rad} = 281.37\,\text{deg}$, $p1' = 5.67$ and $p2 = 3.74$, as illustrated in the lower left corner of Fig. 14. Finally, it is possible to compute the path length $L = 15.37$. The solution in the frame $\mathcal{O}$ is reported in the lower right corner of Fig. 14.

4.2. Computation of the tangent point

Here a computationally efficient algorithm to locate the tangent point $\mathbf{P}$ is described by means of the intersection of the two conics $\mathbf{A}_1$ and $\mathbf{A}_2$ in Eq. (10)–(11).

As a first step, express the coordinates of the initial, final and via points in the reference frame $\mathcal{F}$, so as to have the ellipse's equations in canonical form, and the matrix representation of conics $\mathbf{A}_1$ and $\mathbf{A}_2$ expressed by the Eqs. (10) and (11), respectively (see Appendix).

Here there is an explanation of how the intersection of two conics is computed. Let us consider the set of all the linear combinations $\mu\mathbf{A}_1 + \lambda\mathbf{A}_2$, that is the set of the matrices passing through the same four intersection points as the initial matrices. Among the elements of this set, representing the bundle of conics $\{\mu\mathbf{A}_1 + \lambda\mathbf{A}_2, \mu, \lambda \in \mathbb{R}\}$, it is necessary to search for parameters μ and λ such that the matrix $\mu\mathbf{A}_1 + \lambda\mathbf{A}_2$ is degenerate. Then, the degenerate conic is split into two lines, and each line is intersected with one of the two original conics, thus reducing the problem to a pair of equations of degree two.

According to this approach, the intersection points between the conics $\mathbf{A}_1$ and $\mathbf{A}_2$ are computed using the following algorithm (see the pseudo-code in Algorithm 2).

1. The intersection points of $\mathbf{A}_1$ and $\mathbf{A}_2$ belong also to all the conics given by the matrix pencil $\mu\mathbf{A}_1 + \mathbf{A}_2$, and a degenerate conic $\mathbf{A}_0 = \mu\mathbf{A}_1 + \mathbf{A}_2$ such that $\det(A_0) = 0$ is determined. $\mathbf{A}_0$ consists of two lines or one double line.
2. The two lines r_1 and r_2 of the degenerate conic are extracted from the matrix $\mathbf{A}_0$ (through the function **decomposeDegenerateConic** in Algorithm 2).
3. $\mathbf{A}_1$ is intersected with r_1 and then with r_2, in this way four points $\mathbf{P}_1$, $\mathbf{P}_2$, $\mathbf{P}_3$, $\mathbf{P}_4$ are obtained (through the functions **intersectConicLine** in Algorithm 2).
4. The value of the searched point is initialized with $\mathbf{P}_v$; indeed the searched point has modulus strictly less than $\mathbf{P}_v$ because the ellipse that has to be found is exterior to the circle. This is the reason why among these points $\mathbf{P}$ is chosen among the real ones and such that $\|\mathbf{P}\| > \|\mathbf{P}_j\|$, for $j = 1, \dots, 4$.

This completes the description of the algorithm for the computation of the tangent point $\mathbf{P}$, and allows for the successive computation of the line p_1 and p_2 as described in the previous section, and finally solve the $LSL - LSL$ (or $RSR - RSR$) problem.

4.3. Computation of the isometric path

Here there is the description of the algorithm to derive the isometric path of a generic path $p \in \mathcal{T}_2$, namely for a given path $p \in \mathcal{T}_2$, the objective is to find a path $p' \in \mathcal{T}_1$ with the same length of p. Without loss of generality, the steps to follow for a $RSL - LSL$ path are described in Section 3.2 and summarized in Algorithm 3.

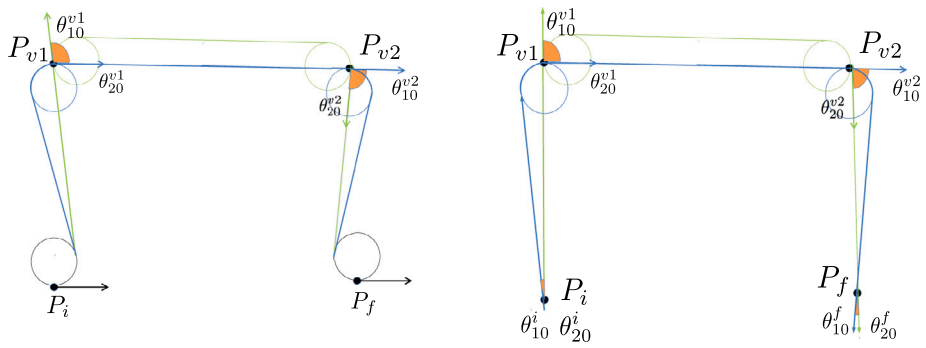

Fig. 13. Figure (a): Feasible intervals in case of 4PDP. Fixed external headings. Figure (b): Free external headings.

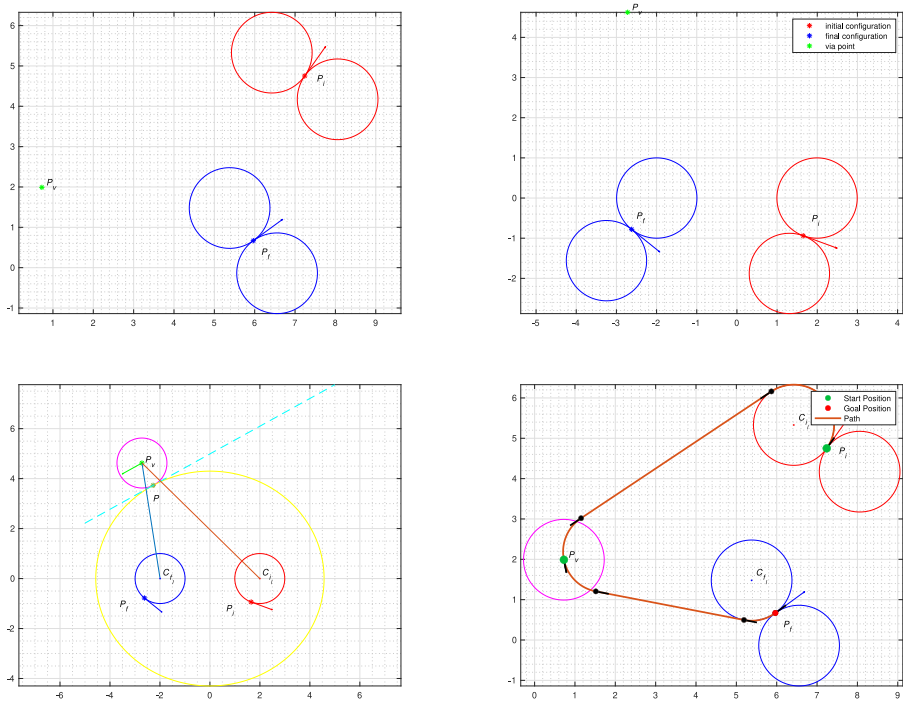

Fig. 14. Illustrative example of the steps related to Algorithm 1, a numerical simulation with $\mathbf{q}_i = (7.24, 4.75, 0.95)$, $\mathbf{q}_f = (5.97, 0.67, 0.63)$ and $\mathbf{P}_v = (0.73, 1.99)$.

Algorithm 2 Computation of the tangent point $\mathbf{P} = (x_0, y_0)$

function ComputationTangentPoint($\mathbf{P}_v$, $\mathbf{A}_1$, $\mathbf{A}_2$)
  $\mu \leftarrow det(\mu\mathbf{A}_1 + \mathbf{A}_2) = 0$
  $[r_1, r_2]$ = **decomposeDegenerateConic**($\mathbf{A}_0$)
  $[\mathbf{P}_1, \mathbf{P}_2]$ = **intersectConicLine**($\mathbf{A}_1, r_1$)
  $[\mathbf{P}_3, \mathbf{P}_4]$ = **intersectConicLine**($\mathbf{A}_1, r_2$)
  $\mathbf{P} = \mathbf{P}_v$
  for j=1:4 **do**
    if ($||\mathbf{P}|| \geq ||\mathbf{P}_j||$) & (Im($\mathbf{P}_j$)=0) **then**
      $\mathbf{P} = \mathbf{P}_j$
    end if
  end for
  return $\mathbf{P}$
end function

Algorithm 3 Computation of the isometric path $p' \in \mathcal{T}_1$

function ComputationIsometricPath($\mathbf{q}_i$, $\mathbf{q}_f$, $\mathbf{P}_v$, $RSL - LSL$)
  Compute θ_{1_0} { ▷ direction of line passing through $\mathbf{P}_v$ and distant Ω^{-1} from $\mathbf{c}_{ir}$ consistent with orientation θ_i }
  Compute θ_{2_0} { ▷ direction of line passing through $\mathbf{P}_v$ and distant Ω^{-1} from $\mathbf{c}_{fl}$ consistent with orientation θ_f }
  Compute $\bar{\theta}_v$ using (16) or (17)
  Compute $\mathbf{c}_{vl}$ associated to $\bar{\theta}_v$
  Compute tangent line between circles C_{ir} and C_{vl} consistent with orientation θ_i
  Compute $\mathbf{c}'_{il}$ { ▷ the isometric $\mathbf{q}'_i = (\mathbf{P}'_i, \theta'_i)$ s.t. $p' \in \mathcal{T}_1$ }
  return $\mathbf{q}'_i$
end function

4.4. Path planning algorithm

Once the optimal $CSC - CSC$ path p^* has been chosen, a path planning algorithm can be defined in order to produce the turning commands for the vehicle. The profile of u can be easily derived considering that $u = \pm\Omega$ along C segments and $u = 0$ on S tracts. Thus, in this subsection a guidance rule algorithm is presented.

The guide algorithm designed for Dubins' vehicle is based on comparisons between:

(i) the lengths of the linear segments traveled by the vehicle (denoted by l in Algorithm 4) and the length of the segments p_1, p_2 (see Fig. 15),
(ii) the orientation of the vehicle and the angular coefficients α and β given by the slope of the straight segments of the trajectory (p_1 and p_2), as depicted in Fig. 15.

In Algorithm 4 the pseudo-code related to a $LSL-LSL$ path is reported, so the angular velocity $u = \Omega$ is chosen for the left turn. The guidance algorithm for $RSR-RSR$ path is analogous apart from $u = -\Omega$.

It is worth remarking that the variables l and θ, namely the length of the linear segment and its actual orientation, can be eventually computed inside the code but since they are related to the status of the vehicle it is possible to consider that they are updated by the sensor units.

In words, Algorithm 4 works as follows. The vehicle starts turning to left from its initial orientation until it reaches α (numerically, until the difference between its current orientation and α is less than a certain tolerance θ_{tol}). Then, the length l of the linear segment traveled by the vehicle is compared with the length of the segment p_1 (lines from 4 to 6 in Algorithm 4). If their difference is larger than a certain tolerance l_{tol}, u is settled to 0 to allow the vehicle to travel along the segment p_1. When these two lengths are the same, the vehicle starts to rotate again and follows analogous rules along the successive circular and linear segments until the final position and orientation are reached.

Algorithm 4 Guidance rule from $\mathbf{q_i} = (\mathbf{P_i}, \theta_i)$ to $\mathbf{q}_f = (\mathbf{P}_f, \theta_f)$ through $\mathbf{q}_v = (\mathbf{P}_v, \theta_v)$, LSL-LSL path.

Require: $\mathbf{P}_i, \theta_i, \mathbf{P}_f, \theta_f, \mathbf{P}_v, \theta, l$
Ensure: u
1: **while** $|\theta - \alpha| > \theta_{tol}$ **do**
2: $u = \Omega$
3: **end while**
4: **while** $|l - p_1| > l_{tol}$ **do**
5: $u = 0$
6: **end while**
7: **while** $|\theta - \beta| > \theta_{tol}$ **do**
8: $u = \Omega$
9: **end while**
10: **while** $|l - p_2| > l_{tol}$ **do**
11: $u = 0$
12: **end while**
13: **while** $|\theta - \theta_f| > \theta_{tol}$ **do**
14: $u = \Omega$
15: **end while**

5. Simulations and comparison results

In this section the results obtained from simulations in Matlab on a Intel Corei5 @1,6 GHz processor are presented. The numerical experiments consider a Dubins' vehicle with $\Omega^{-1} = 1$. Simulations are conducted on $M = 10\,000$ random $(\mathbf{q}_i, \mathbf{P}_v, \mathbf{q}_f)$ instances, where the points are uniformly randomly selected in a 10×10 area and satisfy Fig. 2 (the long distance points hypothesis). This choice allows us to compare the simulation results with the work in Sadeghi and Smith (2016) and especially with the milestone contribution on 3PDP in Chen and Shima (2019b). In Sadeghi and Smith (2016) the authors show two methods based on inversive geometry: the Approximation Method that provides an approximate optimal heading at the via-point, and the Iterative Method, that starting from the approximate heading converges to the optimal one by iteratively corrections. In Chen and Shima (2019b) the authors propose a polynomials based method (PBM). They derive a formula that reveals the relationship, in terms of polynomials, between the unknown orientation angle at the via point and known parameters, the 3PDP is solved by finding zeros of those polynomials.

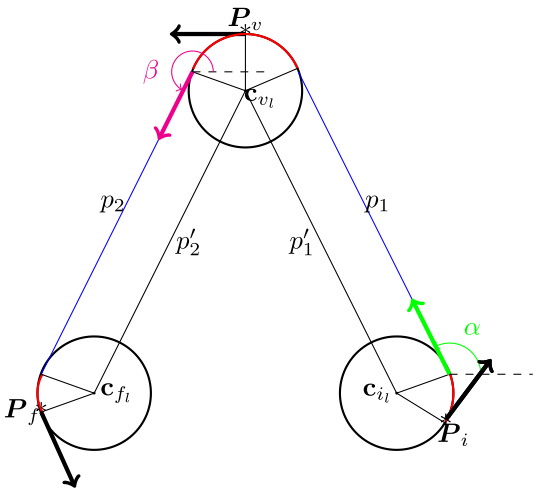

Fig. 15. Graphical sketch of the orientations α and β used in the path planning algorithm.

Conversely, the proposed method is based on analytic geometry tools. For the sake of clarity of exposition, it is denoted as Geometry Based Method (GBM) in the following. The comparison among the different approaches is made in terms of both, computational complexity and accuracy.

5.1. Computational complexity

In order to make a computational complexity comparison between the performance of the algorithm here proposed and the methods mentioned above, it is necessary to introduce a metric for performance evaluation, namely the average factor of improvement I as detailed in the following. The performances of the different methods are compared with the so-called Discretization-Based Method (DBM), as in Chen and Shima (2019b) and Sadeghi and Smith (2016). The Discretization-Based Method is an exhaustive search of the optimal heading at the via point. The rationale behind this method is to discretize the heading θ_v at $\mathbf{P}_v$ into N equally spaced headings, namely $\bar{\theta}(N) = \{\frac{2\pi i}{N} : i = 1, \dots, N\}$, and to solve N pairs of optimal Dubins paths between the configurations $(\mathbf{q}_i, \mathbf{q}_v)$ and $(\mathbf{q}_v, \mathbf{q}_f)$. The DBM selects the shortest path among the N headings, i.e.:

$$\theta_v^{DBM} = \arg \min_{\theta_v \in \bar{\theta}(N)} (L_1(\theta_v) + L_2(\theta_v)). \tag{18}$$

In the following, T_{DBM} denotes the execution time for the solution of the 3PDP with the DBM considering a discretization level $N = 360$, and with T the execution time using an alternative method. The average factor of improvement I over M runs for a given method with respect to the DBM is defined as:

$$I = \frac{1}{M} \sum_{j=1}^{M} \frac{T_{DBM}(j)}{T(j)} \tag{19}$$

In Table 1 it is shown the average factor of improvement of the proposed method in comparison with the methods proposed in Sadeghi and Smith (2016) and in Chen and Shima (2019b).

It is worth highlighting that the algorithms in Chen and Shima (2019b) and Sadeghi and Smith (2016) are able to handle the more general cases where the long distance assumption is not valid. Indeed, in Chen and Shima (2019b) and Sadeghi and Smith (2016) the improvement factor compared with the DBM (with N = 360) is presented for different values of the distance between the points. As clarified above, the proposed algorithm is only valid under the long distance assumption, so the presented comparison is restricted to the 'far case', namely cases where the distance between points is larger than 4 times

Table 1
Comparison of the average factor of improvement in runtime that the various methods give versus the Discretization-Based Method.

Method	I
GBM	152.3
Approx. heading (Sadeghi & Smith, 2016)	74.2
PBM (Chen & Shima, 2019b)	45.7
Iterative method (Sadeghi & Smith, 2016)	13.6

the maximum turning radius. The values reported in Table 1 for the methods in Chen and Shima (2019b) and Sadeghi and Smith (2016) refers only to the long distance scenario.

Overall, comparing the results presented in Table 1 for the different existing methods, it appears that the proposed GBM method is much more efficient than the other ones.

Its significant improvement factor is reasonable because the GBM method is based on different methodologies to get the solution. It uses optimality properties of conic sections curves, rather than the Pontryagin optimality principle as in the most literature. As a result, the geometric computational tools adopted in this paper are expected to have different performances. Actually, even if restricted to the long distance case, the average factor of improvement is doubled if compared to the fastest approach existing in the literature (Table 1).

Hence, the proposed GBM appears to be adequate and well-suited for real-time applications of the shortest path planning three-points Dubins' problems. The reduced computational complexity makes the problem solvable with a shorter execution time and makes the strategy suitable for on-line path planning applications.

5.2. Accuracy

In order to make an accuracy comparison between the discussed methods, the *percentage deviation* d metric in path length for a given method has to be introduced, relative to the path length computed using the discretization method with 360 equally spaced headings, as follows:

$$d = \frac{L_{DBM} - L}{L_{DBM}} \cdot 100, \tag{20}$$

where L_{DBM} denotes the path length for the solution of the 3PDP computed by DBM (with $N = 360$), and L denotes the path length computed by a given method. Such a metric was already used in Sadeghi and Smith (2016), and allows a fair comparison with the Approximation Method and the Iterative Method described therein. The positive values of *percentage deviation* d represent instances in which the proposed method outperforms the discretization based method.

As first comparison, an example pointed out by Chen and Shima (2019b) is considered. The start pose is $\mathbf{q}_i = (0, 0, \frac{\pi}{3})$, $\mathbf{P}_v = (10, 5)$ and $\mathbf{q}_f = (15, 20, \frac{\pi}{6})$. The solution obtained using GBM gives a *percentage deviation* d equal to $+1.08 \cdot 10^{-6}$. This positive value for d means GBM computes a shorter path than the DBM, thus confirming a better accuracy with respect to the exhaustive and computationally demanding search given by the DBM. In this example the optimal heading obtained with the GBM is equal to $\theta_v = 49.02\,\text{deg}$. Note that the DBM with a level of discretization $N = 360$ never reaches this angle, it provides $\theta_v = 49\,\text{deg}$ so that it finds a non-optimal path. In order to reach an optimal angle with at least 2 decimal digits $\theta_v = 49.02\,\text{deg}$ the level of discretization N should be $N \geq 36\,000$, which however results in an higher computational complexity.

A similar result for the same example was presented in Chen and Shima (2019b) for the PBM, the authors shown that the length of solution computed by PBM is slightly less than that by DBM. Hence, both methods, GBM and PBM, provide accurate results, but, as discussed in the previous section, the solution computed by the proposed GMB requires less computational complexity.

As a further comparison, the simulations related to the $M = 10\,000$ random $(\mathbf{q}_i, \mathbf{P}_v, \mathbf{q}_f)$ instances are considered. Simulations results show that percentage of paths over the $M = 10\,000$ instances in which the proposed GBM method is able to find a path shorter than that of the discretization method is 92.23% and, in the other cases the mean error is around 10^{-7}. It is important to emphasize that the used DBM is implemented with such a high level of discretization ($N = 360$) that its solution can be regarded as the exact solution rounded to the nearest integer value. Thus, the relative accuracy of 92.23% is meant as a confirm by simulation that the proposed approach is actually able to find the exact solution using a different methodology.

In Fig. 16 it is possible to see the percentage deviation of the length of paths generated by the GBM from the thick discretization method of 360 equally spaced headings, for the $M = 10\,000$ random instances. The right plot shows the zoom of the left figure to better show the presented result. The central mark indicates the median, and the bottom and top edges of the box indicate the 25th and 75th percentiles, respectively. The whiskers extend to the most extreme data points not considered outliers, and the outliers are plotted individually using the '+ ' marker symbol. Here the median is equal to the positive value $1.0674 \cdot 10^{-6}$ corroborating the accuracy of the approach.

The graphical representation in Fig. 16 can be compared with the equivalent representation reported in Sadeghi and Smith (2016) for the Approximation Method and the Iterative Method (see the fig. 8 of Sadeghi and Smith (2016) related to a distance greater than $4\Omega^{-1}$ between the points).

It can be deduced that GBM works better than the Approximation Method in terms of accuracy because the results published in Sadeghi and Smith (2016) demonstrate that the Approximation Method is always less accurate than the DBM.

On the other hand, the Iterative Method, even if more computationally demanding, is characterized by a better accuracy with respect to the DBM. Therefore, the conclusion is that the accuracy is comparable to what GBM is able to achieve.

This confirms that the proposed GBM is suitable for real-time applications ensuring an adequate level of accuracy as well as a reduced computational effort.

6. Conclusions

This paper addressed the so-called three point Dubins' problem. A novel method based on analytic geometry is proposed and rigorously analyzed to find the shortest Dubins path between three consecutive via-points with prescribed initial and final orientations. Furthermore, an efficient algorithm is derived for each type of Dubins path so as to make the problem solvable with a reduced computational effort. An in-depth comparison with recent existing approaches is discussed corroborating its performance superiority. The simulation and comparative analysis suggests that the proposed strategy is suitable to be adopted for on-line applications given its efficient performances in terms of both accuracy and computational complexity. Potential applications that can benefit from this strategy could be the mission or guidance control levels of autonomous marine or underwater vehicles where the path planning algorithms must be executed in real-time. Other examples are all dynamic situations (e.g., changing environments, …) where it is essential to obtain a real-time or fast solution for constant replanning.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: The author De Palma Daniela serves as Guest Editor of the Special Issue of Control Engineering Practice on "Advances in Navigation and Control for Marine Robotic Applications".

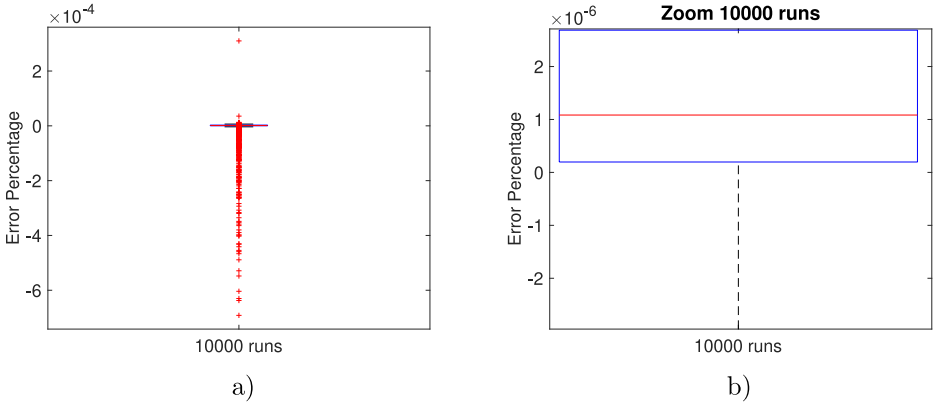

Fig. 16. Figure (a): The percentage deviation d in path length for the GBM. Figure (b): Zoom of figure (a).

Appendix. Basic properties of ellipses

The ellipse is a closed planar curve whose points have constant sum of the distances to two given points called *foci*.

To describe an ellipse through an equation it is possible to use the so called canonical equation of the ellipse, that is:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1. \tag{A.1}$$

Thanks to this equation one immediately understands if the foci lie on the x-axis, i.e. if $a^2 > b^2$, or, otherwise, they lie on the y-axis. In the case $a^2 > b^2$ it is possible to write the coordinates of the foci as $F_1 = (c, 0)$, $F_2 = (-c, 0)$, with $a^2 = b^2 + c^2$. Let P be a point of the ellipse described by Eq. (A.1), then by definition, the following equation holds:

$$d(P, F_1) + d(P, F_2) = 2a, \tag{A.2}$$

where $d(P_1, P_2)$ is the distance between two points P_1 and P_2.

The Euclidean plane is divided by any ellipse into two regions, each of which has an analytic description, derived from (A.1). The *interior* region is enclosed by the ellipse curve and its analytical description is: $\frac{x^2}{a^2} + \frac{y^2}{b^2} < 1$; the *exterior* part is described by the equation $\frac{x^2}{a^2} + \frac{y^2}{b^2} > 1$. The tangent line at one point $P_0 = (x_0, y_0)$ of the ellipse is given by

$$\frac{x_0}{a^2}x + \frac{y_0}{b^2}y = 1. \tag{A.3}$$

Property 1 (*Property of the Tangent Line*). *The angle between the tangent line and the segment connecting the tangent point to one focus is equal to the angle to the other focus. Considering the complementary angles, the normal line at one point P of the ellipse bisects the angle between the lines PF_1 and PF_2.*

An ellipse curve is one type of conic section that is formed by the intersection of a plane and a cone. In the Cartesian coordinate system, all conic sections can be graphed by a quadratic equation in two variables, with the most general equation taking the form $ax^2 + bxy + cy^2 + dx + ey + f = 0$, where a, b, and c are not all zero, and all coefficients are real numbers. This equation can be represented as a matrix equation:

$$\begin{pmatrix} x & y & 1 \end{pmatrix} \underbrace{\begin{pmatrix} a & b/2 & d/2 \\ b/2 & c & e/2 \\ d/2 & e/2 & f \end{pmatrix}}_{\mathcal{M}} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = 0.$$

Matrix representation of conics makes possible their classification in *degenerate* and *non degenerate* conics. In details, given a conic $\mathbf{M}$, when $\det(\mathbf{M}) \neq 0$ the conic is non degenerate and it is an ellipse, a parabola or a hyperbola, otherwise when $\det(\mathbf{M}) = 0$ the conic is degenerate and it consists of two lines. As far as a degenerate conic is concerned it is always possible to extract the two lines' equations from the degenerate conic by solving a quadratic expression in the variable x or y.

References

Antonelli, G., Arrichiello, F., Caiti, A., Casalino, G., De Palma, D., Indiveri, G., Razzanelli, M., Pollini, L., & Simetti, E. (2018). ISME activity on the use of autonomous surface and underwater vehicles for acoustic surveys at sea. *ACTA IMEKO*, [ISSN: 0237028X] *7*(2), 24–31. http://dx.doi.org/10.21014/acta_imeko.v7i2.539.

Berdyshev, Y. (1991). Time-optimal control of a nonlinear system in the problem of visiting a group of points. *Cybernetics and Systems*, *27*, 949–952. http://dx.doi.org/10.1007/BF01246532.

Berdyshev, Y. (2002). A problem of the sequential approach to a group of moving points by a third-order non-linear control system. *Journal of Applied Mathematics and Mechanics*, *66*(5), 709–718. http://dx.doi.org/10.1016/S0021-8928(02)90001-8, URL: https://www.sciencedirect.com/science/article/pii/S0021892802900018.

Boissonant, J., Cerezo, A., & Leblond, J. (1994). *A note on shortest paths in the plane subject to a constraint on the derivative of the curvature*: *Tech. Rep. 2160*, INRIA.

Bui, X.-N., Boissonnat, J.-D., Soueres, P., & Laumond, J.-P. (1994). Shortest path synthesis for Dubins non-holonomic robot. In *Robotics and automation, 1994. Proceedings., 1994 IEEE international conference on* (pp. 2–7). IEEE, http://dx.doi.org/10.1109/ROBOT.1994.351019.

Caillau, J. B., Maslovskaya, S., Mensch, T., Moulinier, T., & Pomet, J. B. (2019). Zermelo-Markov-Dubins problem and extensions in marine navigation. In *2019 IEEE 58th conference on decision and control* (pp. 517–522). http://dx.doi.org/10.1109/CDC40024.2019.9029293.

Chen, Z. (2020). On Dubins paths to a circle. *Automatica*, *117*, Article 108996. http://dx.doi.org/10.1016/j.automatica.2020.108996, URL: https://www.sciencedirect.com/science/article/pii/S0005109820301941.

Chen, Z., & Shima, T. (2019a). Relaxed Dubins problems through three points. In *2019 27th Mediterranean conference on control and automation* (pp. 501–506). http://dx.doi.org/10.1109/MED.2019.8798584.

Chen, Z., & Shima, T. (2019b). Shortest Dubins paths through three points. *Automatica*, 368–375. http://dx.doi.org/10.1016/j.automatica.2019.04.007.

Cheng, C., Sha, Q., He, B., & Li, G. (2021). Path planning and obstacle avoidance for AUV: A review. *Ocean Engineering*, *235*, Article 109355. http://dx.doi.org/10.1016/j.oceaneng.2021.109355, URL: https://www.sciencedirect.com/science/article/pii/S002980182100771X.

De Palma, D., & Parlangeli, G. (2022). Shortest path type classification for real-time three-points dubins problems. In *2022 30th Mediterranean conference on control and automation* (pp. 520–525). http://dx.doi.org/10.1109/MED54222.2022.9837205.

Dubins, L. E. (1957). 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. http://dx.doi.org/10.2307/2372560.

Fossen, T. I., Pettersen, K. Y., & Galeazzi, R. (2015). Line-of-sight path following for dubins paths with adaptive sideslip compensation of drift forces. *IEEE Transactions on Control Systems Technology*, *23*(2), 820–827. http://dx.doi.org/10.1109/TCST.2014.2338354.

Fraichard, T., & Scheuer, A. (2001). From reeds and sheepp's to continuous-curvature paths. *IEEE Transactions on Robotics*, *20*, 1025–1035. http://dx.doi.org/10.1109/TRO.2004.833789.

Frego, M., Bevilacqua, P., Saccon, E., Palopoli, L., & Fontanelli, D. (2020). An iterative dynamic programming approach to the multipoint markov-dubins problem. *IEEE Robotics and Automation Letters*, *5*(2), 2483–2490. http://dx.doi.org/10.1109/LRA.2020.2972787.

Galceran, E., & Carreras, M. (2013). A survey on coverage path planning for robotics. *Robotics and Autonomous Systems*, *61*(12), 1258–1276. http://dx.doi.org/10.1016/j.robot.2013.09.004.

Goaoc, X., Kim, H. S., & Lazard, S. (2013). Bounded-curvature shortest paths through a sequence of points using convex optimization. *SIAM Journal on Computing*, *42*(2), 662–684. http://dx.doi.org/10.1137/100816079.

Indiveri, G., De Palma, D., & Parlangeli, G. (2016). Single range localization in 3-D: Observability and robustness issues. *IEEE Transactions on Control Systems Technology*, *24*(5), 1853–1860. http://dx.doi.org/10.1109/TCST.2015.2512879.

Jha, B., Chen, Z., & Shima, T. (2020). On shortest Dubins path via a circular boundary. *Automatica*, *121*, Article 109192. http://dx.doi.org/10.1016/j.automatica.2020.109192.

Karimi, H. R., & Lu, Y. (2021). Guidance and control methodologies for marine vehicles: A survey. *Control Engineering Practice*, *111*, Article 104785. http://dx.doi.org/10.1016/j.conengprac.2021.104785, URL: https://www.sciencedirect.com/science/article/pii/S0967066121000629.

Kaya, C. (2019). Markov-Dubins interpolating curves. *Computational Optimization and Applications*, *73*, 647–677. http://dx.doi.org/10.1007/s10589-019-00076-y.

LaValle, S. M. (2006). *Planning algorithms*. USA: Cambridge University Press.

Liang, X., Jiang, P., & Zhu, H. (2020). Path planning for unmanned surface vehicle with Dubins curve based on GA. In *2020 Chinese automation congress* (pp. 5149–5154). http://dx.doi.org/10.1109/CAC51589.2020.9327163.

Mitchell, J. S. (2000). Geometric shortest paths and network optimization. In *Handbook of computational geometry, vol. 334* (pp. 633–702). Citeseer.

Ozog, P., Carlevaris-Bianco, N., Kim, A., & Eustice, R. M. (2016). Long-term mapping techniques for ship hull inspection and surveillance using an autonomous underwater vehicle. *Journal of Field Robotics*, *33*(3), 265–289. http://dx.doi.org/10.1002/rob.21582, URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21582.

Parlangeli, G. (2019a). A low-complexity algorithm for shortest Dubins paths with intermediate via points. In *27th Mediterranean conference on control and automation* (pp. 495–500). http://dx.doi.org/10.1109/MED.2019.8798576.

Parlangeli, G. (2019b). Shortest paths for Dubins vehicles in presence of via points. In *IFAC-PapersOnLine: 10th IFAC symposium on intelligent autonomous vehicles IAV 2019, vol. 52, no. 8* (pp. 295–300). http://dx.doi.org/10.1016/j.ifacol.2019.08.086.

Parlangeli, G., Ostuni, L., Mancarella, L., & Indiveri, G. (2009). A motion planning algorithm for smooth paths of bounded curvature and curvature derivative. In *Mediterranean conference on control and automation* (pp. 73–78). Thessaloniki, Greece: IEEE, http://dx.doi.org/10.1109/MED.2009.5164517.

Reeds, J., & Shepp, L. (1990). Optimal paths for a car that goes both forwards and backwards. *Pacific Journal of Mathematics*, *145*(2), 367–393. http://dx.doi.org/10.2140/PJM.1990.145.367.

Saccon, E., Bevilacqua, P., Fontanelli, D., Frego, M., Palopoli, L., & Passerone, R. (2021). Robot motion planning: Can GPUs be a game changer? In *2021 IEEE 45th annual computers, software, and applications conference* (pp. 21–30). IEEE, http://dx.doi.org/10.1109/COMPSAC51774.2021.00015.

Sadeghi, A., & Smith, S. L. (2016). On efficient computation of shortest dubins paths through three consecutive points. In *Decision and control (CDC), 2016 IEEE 55th conference on* (pp. 6010–6015). IEEE, http://dx.doi.org/10.1109/CDC.2016.7799192.

Savla, K., Frazzoli, E., & Bullo, F. (2008). Traveling salesperson problems for the dubins vehicle. *IEEE Transactions on Automatic Control*, *53*(6), 1378–1391. http://dx.doi.org/10.1109/TAC.2008.925814.

Shkel, A., & Lumelsky, V. (1996). On calculation of optimal paths with constrained curvature: the case of long paths. In *Proceedings of IEEE international conference on robotics and automation, vol. 4* (pp. 3578–3583). http://dx.doi.org/10.1109/ROBOT.1996.509258.

Shkel, A. M., & Lumelsky, V. (2001). Classification of the dubins set. *Robotics and Autonomous Systems*, *34*, 179–202. http://dx.doi.org/10.1016/S0921-8890(00)00127-5.

Swokowski, E. W. (1979). *Calculus with analytic geometry*. Taylor & Francis.

Váňa, P., & Faigl, J. (2020). Optimal solution of the Generalized Dubins Interval Problem: finding the shortest curvature-constrained path through a set of regions. *Autonomous Robots*, *44*(7), 1359–1376. http://dx.doi.org/10.1007/s10514-020-09932-x.

Wheare, J., Lammas, A., & Sammut, K. (2019). Toward the generation of mission plans for operation of autonomous marine vehicles in confined areas. *IEEE Journal of Oceanic Engineering*, *44*(2), 320–330. http://dx.doi.org/10.1109/JOE.2018.2884859.

Zeng, Z., Lian, L., Sammut, K., He, F., Tang, Y., & Lammas, A. (2015). A survey on path planning for persistent autonomy of autonomous underwater vehicles. *Ocean Engineering*, *110*, 303–313. http://dx.doi.org/10.1016/j.oceaneng.2015.10.007, URL: https://www.sciencedirect.com/science/article/pii/S0029801815005442.