

2019-07-23

Restricted Orientation Dubins Path with Application to Sailboats

Vautier, Ulysse

<http://hdl.handle.net/10026.1/14737>

10.1109/Ira.2019.2930424

IEEE Robotics and Automation Letters

Institute of Electrical and Electronics Engineers (IEEE)

All content in PEARL is protected by copyright law. Author manuscripts are made available in accordance with publisher policies. Please cite only the published version using the details provided on the item record or document. In the absence of an open licence (e.g. Creative Commons), permissions for further reuse of content should be sought from the publisher or author.

Restricted Orientation Dubins Path with Application to Sailboats

Ulysse Vautier¹, Christophe Viel¹, Jian Wan¹, Luc Jaulin², Robert Hone¹, and Ming Dai¹

Abstract—This paper develops a geometrical construction of the shortest Dubins path in a discontinuous orientation-restricted environment. The method proposed here builds the shortest path from one pose to the other while avoiding a no-go zone in terms of orientation, and being constrained to move forward. Finally, an application to autonomous sailboats is then provided to validate the feasibility of the planned shortest path in a position keeping scenario.

Index Terms—Motion and Path Planning, Nonholonomic Motion Planning, Motion Control, Marine Robotics

I. INTRODUCTION

DUBINS showed that the shortest path between two poses for a Dubins vehicle consists of a subset of turns of maximum curvature and straight lines. The Dubins vehicle is a model of a vehicle capable of only moving forward at a constant velocity and turning with a constraint in curvature. The shortest path was proven to be of type *CSC* or *CCC*, representing the combination of turns (*C*) and straight lines (*S*) [1]. It was proven to be the shortest using Maximum Pontryagin's principle [2] and additional information for the synthesis of the path was given later [3]. The Dubins path is used in multiple variations for different platforms such as Aerial Unmanned Vehicles (AUVs) [4], wheeled vehicles [5] or general 2D and 3D path-planning methods [6], [7]. These studies sometimes differ from the classic Dubins vehicle model, by varying the rate of turn or the velocity of the vehicle. Other extensions to the Dubins path planning problem have also been studied such as the Reed-Sheps curves for Dubins vehicles capable of moving backward [8].

Compared to other studies of the Dubins path, this paper focuses on a classic Dubins vehicle model but adds a simple restriction on orientations at which the vehicle cannot be. This study is initialized from the problem of controlling autonomous sailboats. The Dubins vehicle matches the sailboat in short-length path-planning since it cannot brake and it is always moving forward. Furthermore, the Dubins path considered here has additional restriction since sailboats are constrained by the wind direction, and they cannot go upwind.

Manuscript received: March, 14, 2019; Revised May, 01, 2019; Accepted July, 10, 2019.

This paper was recommended for publication by Editor Jonathan Roberts upon evaluation of the Associate Editor and Reviewers' comments.

¹The authors are with the School of Engineering, University of Plymouth, UK. (e-mail: ulysse.vautier@plymouth.ac.uk; jian.wan@plymouth.ac.uk; y.dai@plymouth.ac.uk; christophe.viel@gadz.org; robert.hone@plymouth.ac.uk)

²L. Jaulin is with the ENSTA-Bretagne, France. (e-mail: lucjaulin@gmail.com)

Digital Object Identifier (DOI): see top of this page.

While this work has been studied for sailboats, the results can be generalized to other types of vehicles with the need to avoid certain ranges of orientations such as AUVs in case of emergencies or land-yachts where a certain range of orientation cannot be attained.

The Dubins' vehicles with constrained orientations have been studied. Primary work comes from [9], where the boat studied has a constrained rate of turning. The shortest path therein was proven to be of the type, or a subset of *CSCSC* in an anisotropic medium, or more specifically with a convex polar speed diagram. A method to find the shortest path was also proposed therein. Similarly to the *CSCSC* path, [10], [11] have worked on the three-points dubins path, to solve the Dubins Traveling Salesman Problem (DTSP), with or without restriction on orientation on the last two points. These work have demonstrated a method to build the shortest path, for a Dubins vehicle, between a starting pose and a final point by passing through a medium point. Other work, such as [12], [13], studied the Dubins paths in heterogeneous environments, and [14], which concerned the Dubins Interval Problem (DIP), studied paths where the start and end headings were constrained to be an interval. These work show the research done on Dubins path when the vehicle has a variable rate of turn or velocities but does not consider when the velocity of the vehicle is discontinuous.

The problem considered here is similar to [9], but the velocity considered here is binary rather than continuous: the velocity will be null in the range of the restricted orientation and full speed otherwise. While [9] gives a more generic and complete solution to constructing a Dubins path for all vehicles with a continuous convex speed polar plot, this paper proposes a simple geometric algorithm for a more specific Dubins car with a non-continuous binary speed and a constant turning radius. By restricting the domains of orientation, different problems arise such as the feasibility of the path, which are to be discussed here. This paper still uses the properties of the *CSCSC* path and develops a method to build this set of paths. One could also see similarities with [10], [11] but differs in that the second turn in the *CSCSC* path is already known. The goal of this paper is to find each of the turns to have the shortest path. The method proposed here allows integration in any embedded systems with small resources and real-time constraints. And as it will be demonstrated, working on this simple model can easily be scaled up to other complex models such as sailboats. The method of this paper is applied to station keeping for sailboats which has only a few published studies while being an important application to ocean survey [15]. Finally, this paper provides a proof of the shortest path for

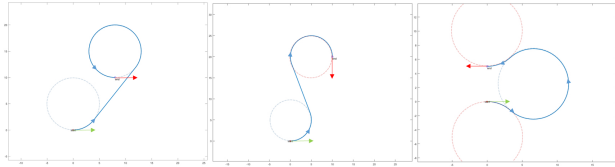


Fig. 1. Examples of classic Dubins paths in *LSL*, *LSR* and *RLR* configurations with a rate of turn $r = 5$. The starting pose is represented by the green arrow and the final pose by the red arrow. It is the shortest path for a Dubins vehicle from the start pose to the end pose.

intersecting tangent lines to two circles. This proof is used for this problem but could also be used extended to other cases.

The paper is organized as follows. First, the problem to be studied and the corresponding notations are described in Section II. The conditions for the Dubins path's attainability are discussed in Section III. The method to build the shortest route with constrained orientations is detailed in Section IV and the corresponding proof is provided in Appendix A. The application to station keeping control of sailboats using the planned shortest path is provided in Section V. Some conclusions are given in Section VI.

II. PROBLEM STATEMENT

The model of Dubins vehicle is adjusted to be as follows:

$$\begin{cases} \dot{x} &= \cos(\theta) \\ \dot{y} &= \sin(\theta) \\ \dot{\theta} &= \frac{u}{r} \\ \cos(\phi - \theta) &\leq \cos(\delta) \end{cases} \quad (1)$$

where $(x, y) \in \mathbb{R}^2$ is the vehicle's position, $\theta \in [0, 2\pi]$ is the vehicle orientation, $u \in [-1, 1]$ is the control input with $r \in \mathbb{R}^+$ the turn radius, ϕ is the restricted angle and finally $\delta \in [0, \pi]$ is the range of unacceptable orientations around ϕ . The Dubins vehicle has a velocity $v > 0$. In this case, a unit velocity $v = 1$ is chosen as it does not impact the problem. The last condition has been added from the original Dubins model for our specific problem. This condition indicates that the vehicle can only move if the orientation θ is out of the restricted range of orientation $[\phi - \delta, \phi + \delta]$.

The Dubins path is the shortest path from one pose to the other. The shortest path is a *CSC* or *CCC* set. The latter being a special case when the distance between the two poses is shorter than twice the radius of the vehicle's maximum rate of turn. The path is built using the turn circles around the starting and finishing poses and the bitangents connecting those circles. The shortest path is then constructed from the starting pose, following a turn until it leaves it by a straight line along the bitangent and finally turning to attain the final pose. In this paper, we call this Dubins path, the classic Dubins path to differentiate with the restricted path that is to be built here. Examples of the classic Dubins path in *CSC* and *CCC* configurations are given in Figure 1 with a rate of turn $r = 5$.

Bitangents, in this case, are the tangent lines between two circles. There are at most 4 bitangents to two circles. They are represented in Figure 2 in grey segments joining the red (inner tangents) and blue (outer tangents) circles. They are

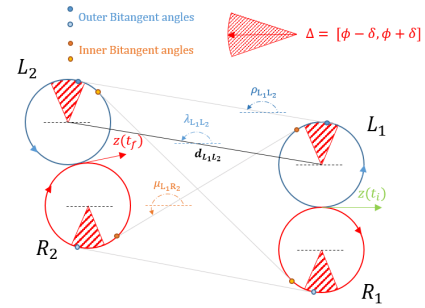


Fig. 2. Figure showing several notations used. In this specific configuration, no classic Dubins path works regarding the restricted angles.

expressed later in Eq.(2). In the Dubins path, only those that are possible, considering the directions of turn and the orientation of the vehicle, are considered. It is worth noting that when the first turn and the following one are the same, then only an outer bitangent is possible and when they are in the opposite direction, it is an inner bitangent.

The goal of this paper is to find the shortest path when the Dubins' vehicle is constrained by orientation. The vehicle is not allowed at any time to face a range of orientation. The modified model in Eq.(1) is to be used to build the path. The polar speed diagram of this vehicle is convex given the restrictions on the orientation. As such, the work that was done in [9] can be used, which means that the shortest path will be of the type, or the subset of, *CSCSC*. Similarly to [16], the Dubins' path is built upon the bitangents between the first and final turn of the path. To solve this problem, a study of the available domain of the orientations of the vehicle will be shown. Consequently, the best orientations will be chosen in that domain for which the path becomes the shortest.

The system is established in an Euclidean plane $(x, y) \in \mathbb{R}^2$. Every angle discussed in this paper is in the trigonometrical range of $T = [0, 2\pi]$, expressed in the Euclidean space, relative to the X-axis. The vehicle pose $z(t) = (x(t), y(t), \theta(t)) \in \mathbb{R}^2 \times T$ is expressed for $t \in [t_i, t_f]$ where t_i is the initial time and t_f the final time. For the examples shown in this paper, since the problem is of cylindrical symmetry, we will take $z(t_s) = (0, 0, 0)$ for the sake of simplicity. The defined variables and constants are shown in Figure 2.

- Let the times t_{mi}, t_{mf} be the medium time at which the path leaves the first turn and the medium time at which the path lands on the final turn.
- Let C be a turn and also represent the curvature's circle of the turn. It can either be a left (*L*) or right turn (*R*).
- Let C_n be the turn or the curvature's circle of the turn in an ordered configuration. As such, in a configuration $C_1SC_2SC_3$, C_1 is the first turn of the path.
- Let $\lambda_{C_1C_2}$ be the angle of inclination, relative to the X-axis, of the line passing through the center of circles C_1 and C_2 .
- Let $d_{C_1C_2}$ be the distance between the center of circles C_1 and C_2 .
- Let $u_C \in \{-1, 1\}$ be a unit of direction following the counter-clockwise (CCW) convention. Where $u_L = 1$ and $u_R = -1$. This is used to simplify the equations and

consider the direction of turns.

- Let $\rho_{C_1C_2}$ and $\mu_{C_1C_2}$ be respectively the angles, in the Euclidean frame, of the outer and inner bitangent between C_1 and C_2 . Let $\mu_{C_1C_3}^*$ be the bitangent angle in the special $C_1C_2C_3$ configuration. Let $\eta_{C_1C_2}$ be the general notation for bitangent angles.
- Let $\Delta = [\phi - \delta, \phi + \delta]$ be the no-go zone boundaries in the Euclidean frame. With $\underline{\Delta}$ its minimal boundary and $\overline{\Delta}$ its maximal boundary. Let $\underline{\Delta}_C$ express the smallest restricted angle depending on the turn and $\overline{\Delta}_C$ the highest. As such $\underline{\Delta}_C = \phi - u_C\delta$ and $\overline{\Delta}_C = \phi + u_C\delta$.

The method described in this paper uses the properties of the bitangents. Their values can be found using geometrical analysis, in the specific case where all circles are of the same radius:

$$\begin{cases} \rho_{C_1C_2} &= \lambda_{C_1C_2} \\ \rho_{C_2C_1} &= \rho_{C_1C_2} + \pi \\ \mu_{C_1C_2} &= \lambda_{C_1C_2} + u_{C_1} \arccos\left(\frac{2r}{d_{C_1C_2}}\right) \\ \mu_{C_2C_1} &= \mu_{C_1C_2} + \pi \end{cases} \quad (2)$$

In the special case when $d_{C_1C_2} < 4r$, the classic Dubins path is in a CCC configuration. In this configuration, the turns are ordered as such: $C_1C_2C_3$. The bitangent angles, in this case, can easily be found using the properties of the isosceles triangle, considering that the middle turn is at a distance of $2r$ from the center of each starting and final turn:

$$\begin{cases} \mu_{C_1C_3}^* &= \lambda_{C_1C_3} + u_{C_1} \arccos\left(\frac{d_{C_1C_3}}{4r}\right) \\ \mu_{C_3C_1}^* &= \mu_{C_1C_3}^* - u_{C_1}\pi \end{cases}$$

The first problem is to know when a classic Dubins path is not allowed considering the restriction on the orientations. This is done by using the properties of the bitangents and establishing the simple conditions in which the Dubins path orientates the vehicle towards the forbidden angles. When the classic Dubins path is not possible, another shortest path that never faces the range of unacceptable angles is built by adding a turn, as proven in [9]. The following two sections will explain the multiple conditions and the method to build the new $CSCSC$ path.

III. CONDITIONS FOR EXISTENCE OF THE RESTRICTED ORIENTATION DUBINS PATH

The restriction on orientations given to the Dubins vehicle model in Eq.(1) forces to consider in which cases any path from one pose to the other can fail. The strong restriction gives the initial conditions of this model. First of all, the initial and final poses cannot be in the no-go zone:

$$\begin{cases} \cos(\phi - \theta(t_i)) &< \cos(\delta) \\ \cos(\phi - \theta(t_f)) &< \cos(\delta) \end{cases} \quad (3)$$

Secondly, the range of restricted orientations should not stop the vehicle from ever being able to approach the goal:

$$\begin{cases} [\theta(t_i), \underline{\Delta}_{C_1}] \cap [\lambda_{C_1C_2} - \frac{\pi}{2}, \lambda_{C_1C_2} + \frac{\pi}{2}] \neq \emptyset \\ OR \\ [\overline{\Delta}_{C_2}, \theta(t_f)] \cap [\lambda_{C_2C_1} - \frac{\pi}{2}, \lambda_{C_2C_1} + \frac{\pi}{2}] \neq \emptyset \end{cases} \quad (4)$$

Now that the primary conditions are set, the conditions in which the Dubins path fails can be found. The classic Dubins path uses the bitangents to go from one turn to the other. As such, the Dubins path is not possible if the bitangent angle cannot be reached. This can be checked by looking at the intersection between the domain of the turns with the no-go zone:

$$\begin{cases} \Delta \cap [\theta(t_i), \eta_{C_1C_2}] \neq \emptyset \\ \Delta \cap [\eta_{C_2C_1}, \theta(t_f)] \neq \emptyset \end{cases} \quad (5)$$

To test these conditions in a system, the directed arclength is used to have a programmatically correct implementation. The directed arclength is defined for a turn C , between an initial angle θ_i and a final angle θ_f , as such:

$$\hat{A}(\theta_i, \theta_f)_C = \text{sawtooth}\left(u_C(\text{sawtooth}(\theta_f) - \text{sawtooth}(\theta_i))\right)$$

Where the *sawtooth* function is the sawtooth wave with a period 2π defined as $\text{sawtooth}(t) = t \pmod{2\pi} - \pi$. Using Eq.(5) and the above directed arclength, a method to check whether the Dubins path is possible can be implemented.

In the case where one of these conditions fails, a new path of type $CSCSC$ must be built. The next section will explain where to add the new turn to have the shortest path.

IV. BUILDING THE SHORTEST PATH WHEN ADDING A TURN

The path built here is of type $C_1SC_2SC_3$. Where C_1 is the starting turn and C_3 the final turn. A C_2 turn is added to be able to arrive at the final pose while avoiding the unacceptable range of orientations. The bitangents and the direction of the turns are of importance when building the path. A Dubins path will be built between C_1 and C_2 , and also between C_2 and C_3 . This means that the arriving and leaving angles on C_2 are bitangents angles. Figure 3 shows a $CSCSC$ path, in the specific case where the turn directions $C_1 = C_3 = L$, with different C_2 turns. The domain restrictions on the pair (θ_1, θ_3) , the angle on the first turn and the angle on the last turn, will be exposed here.

The first domain is the possible angles of the first and final turn considering the first and final pose and the restriction:

$$\mathbb{T}_{C_1C_3} = \{(\theta_1, \theta_3) \in T \times T \mid \begin{aligned} \theta_1 &\in [\theta(t_i), \underline{\Delta}_{C_1}], \\ \theta_3 &\in [\overline{\Delta}_{C_3}, \theta(t_f)] \end{aligned}\} \quad (6)$$

Conditions of intersection must be set for the different angles on C_1 and C_3 . The added turn can only exist if C_2 is attainable considering the directed tangent lines from the starting and finishing turn. The parametric equations of the tangent lines will be used to define their intersections. We

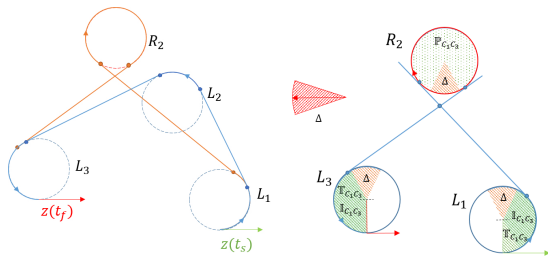


Fig. 3. (Left) Examples of LSC_2SL paths without any restriction in orientation. (Right) Domains restriction process in a particular $LSRSL$ configuration. $\mathbb{T}_{C_1C_3}$ is in green stripes, $\mathbb{P}_{C_1C_3}$ is in green dots, $\mathbb{I}_{C_1C_3}$ is the overlaying solid green color.

define O_C as the center of circle C . This equation expresses the distance to the intersection of the directed lines, tangent to the turns C_1 and C_3 :

$$\mathbb{I}_{C_1C_3} = \{(\theta_1, \theta_3) \in T \times T \mid \exists t, t' \geq 0, O_{C_1}^{\theta_1} + t \begin{bmatrix} \cos(\theta_1 + u_{C_1} \frac{\pi}{2}) \\ \sin(\theta_1 + u_{C_1} \frac{\pi}{2}) \end{bmatrix} = O_{C_3}^{\theta_3} + t' \begin{bmatrix} \cos(\theta_3 + u_{C_3} \frac{\pi}{2}) \\ \sin(\theta_3 + u_{C_3} \frac{\pi}{2}) \end{bmatrix}\} \quad (7)$$

with:

$$O_{C_n}^{\theta_k} = O_{C_n} + r \begin{bmatrix} \cos(\theta_k) \\ \sin(\theta_k) \end{bmatrix}$$

The arriving and leaving angles are bitangents, the angles on C_1 and C_3 govern the angles on C_2 . As such, the domain on C_2 must be checked to know whether they intersect with the no-go zone. Transformed to the pair $(\theta_{C_1}, \theta_{C_3})$, it gives:

$$\mathbb{P}_{C_1C_3} = \{(\theta_1, \theta_3) \in T \times T \mid \Delta \cap [\theta_{ms}, \theta_{mf}] \neq \emptyset\} \quad (8)$$

where θ_{ms} and θ_{mf} are the bitangent angles on C_2 respectively between C_1 and C_2 and between C_2 and C_3 . π is added whether the angles on C_2 are inner or outer bitangents.

$$\begin{cases} \theta_{ms} &= \theta_1 + \frac{1-u_{C_1}u_{C_2}}{2}\pi \\ \theta_{mf} &= \theta_3 + \frac{1-u_{C_3}u_{C_2}}{2}\pi \end{cases}$$

The domain of orientation of the vehicle is now set, with the domain of the bitangents on C_2 being restricted by the domains defined above Eq.(6), Eq.(7) and Eq.(8), shown in Figure 3:

$$\begin{cases} \mathbb{F}_{C_1C_3} &= \mathbb{T}_{C_1C_3} \cap \mathbb{I}_{C_1C_3} \cap \mathbb{P}_{C_1C_3} \\ (\theta([t_i, t_{mi}]), \theta([t_{mf}, t_f])) &\in \mathbb{F}_{C_1C_3} \end{cases} \quad (9)$$

Now that the domain of angles on the first and last turn is established, the angle at which the vehicle must leave the first turn and arrive on the last turn must be found. Or to put it differently, where to position the center of C_2 turn in the domain specified. The angles can be found by minimizing the length of the whole path. Another method is exposed here.

It is known that the bitangents are the shortest path possible between two circles. The hypothesis made in this paper is that the longer the distance of the middle turn is from the bitangent

lines, the longer the path becomes. This statement might be obvious but the proposition is necessary to fully understand the problem. The proposition is available in the Appendix. It demonstrates that when either the difference between the angle of the first or the final turn and the corresponding bitangent angle increases, the paths length increases. This means that to get the shortest path, θ_{C_1} and θ_{C_3} must simply be the closest to the bitangent, as much as the domains in Eq.(9) allows.

The values for the leaving and landing angles, respectively on C_1 and C_3 , are then:

$$\begin{cases} \theta(t_{mi}) &= \min(|\theta_{C_1} - \eta_{C_1C_3}|, \theta_{C_1} \in \mathbb{F}_{C_1C_3}) \\ \theta(t_{mf}) &= \min(|\theta_{C_3} - \eta_{C_3C_1}|, \theta_{C_3} \in \mathbb{F}_{C_1C_3}) \end{cases} \quad (10)$$

The singularities explained at the end of Appendix A can easily be taken care of by evolving the angles $\theta(t_{mi})$ and $\theta(t_{mf})$ from Eq. (10) in opposite ways, while staying in the available domain, until the middle turn is as close to an equidistant line.

All the information to build the path has been established so far. The leaving angle from the first turn and the arriving angle on the last turn are given. In effect, the angles on C_2 using Eq. (10) is given. Finally, to finish the path, straight lines going from C_1 to C_2 and from C_2 to C_3 and arcs on C_1 , C_3 and C_2 are constructed.

V. APPLICATION

To apply the method in an embedded system, the workflow is shown in Algorithm 1.

Data: $z(t_i), z(t_f), \phi, \delta$

Result: Path Configuration and associated angles

if Eq.(3) and Eq.(4) are satisfied **then**

foreach Pairs of initial and final turn (C_1, C_3) **do**

if Eq.(5) is satisfied **then**

// Classic Dubins

Configuration_Queue.add(C_1SC_3);

else

// Build a $C_1SC_2SC_3$ path

foreach $C_2 = \{L, R\}$ **do**

Compute the domain $\mathbb{F}_{C_1C_3}$, Eq.(9);

Compute $\theta(t_{mi})$ and $\theta(t_{mf})$, Eq.(10);

CSCSC_Queue.add($C_1SC_2SC_3$);

end

Configuration_Queue.add(min(CSCSC_Queue));

CSCSC_Queue.clear();

end

end
Configuration = min(Configuration_Queue);

end

Algorithm 1: Simplified workflow to find the best path configuration. The path can then be built using the configuration's information. Source code is available at [17].

The algorithm checks if the classic Dubins path is possible for each combination of starting and finishing turns. If not, a

new *CSCSC* path is built. Finally, the shortest path is chosen from all the different possible paths.

Multiple ways can be used to compute Eq.(9) and Eq.(10) in an embedded system. Set computation can be used if available or looping through the discretized set to compute the minimum from an array of values, while computationally expensive, can also be used.

The application here is applied to a station keeping problem. Since a sailboat cannot turn as willingly as is the case with other vehicles, a path-following control is proposed. This method will show the best path for a position keeping problem for a Dubins vehicle with restricted orientation as proven in Appendix A.

A path-following control using a well-established sailboat model inspired by [18] will be simulated. This model takes into account the wind and water current which apply forces to the sailboat at all time. As this method makes more sense in restricted path lengths, an application to station keeping will be exposed. The model of the sailboat is as followed:

$$\begin{cases} \dot{x} &= v \cos(\theta) + p_1 a_{tw} \cos(\psi_{tw}) \\ \dot{y} &= v \sin(\theta) + p_1 a_{tw} \sin(\psi_{tw}) \\ \dot{\theta} &= w \\ \dot{v} &= \frac{g_s \sin(\delta_s) - g_r p_{11} \sin(\delta_r) - p_2 v^2}{p_{10}} \\ \dot{w} &= \frac{g_s (p_6 - p_7 \cos(\delta_s)) - g_r p_8 \cos(\delta_r) - p_3 w v}{p_{10}} \end{cases} \quad (11)$$

with:

$$\begin{cases} g_s = p_4 a_{aw} \sin(\delta_s - \psi_{aw}) \\ g_r = p_5 v^2 \sin(\delta_r) \end{cases}$$

The values of the parameters for the simulation were taken from [18], Section 2.7, Table 2.2.

δ_s and δ_r are the angles of the sail and rudder respectively, that are used to control the sailboat. a is the speed of the wind and ψ its angle. tw refers to the true wind and aw to the apparent wind.

MATLAB is used to compute the simulation. The Dubins path algorithm is based upon [16]'s work, and the source code is available online [17]. The sets from Eq.(9) were discretized and used as matrices to find the minimum in Eq.(10). In terms of computational speed, on the same machine, finding the shortest path is of the same order as the classic Dubins path. Computing the shortest path and plotting it averages at 0.15s for a path with 1000 points while the classic one takes approximately 0.1s. Considering the speed and efficiency of such an algorithm, it is possible to imagine launching the algorithm in a loop to face sudden changes in wind direction. In the same wind condition, the algorithm would not change the previous path, it would simply create the same optimal sub-path *SCSC* if computed after the first turn.

For this application, a particular situation is taken where the classic Dubins path does not work. The path is then tested to see if it is plausible for a path-following algorithm. This method is applied for station keeping, or also called position keeping. In this case, the start pose and the end pose are the same. A small offset was added to the end pose for it to work, or else it would consider the optimal path to be attained

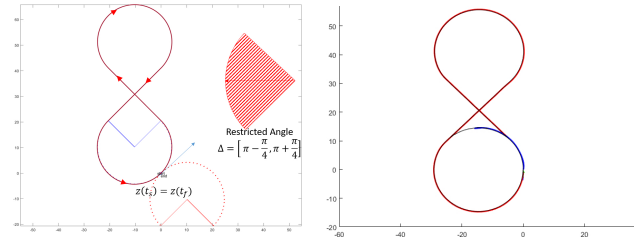


Fig. 4. Path built for a position keeping scenario where the start pose and the end pose are the same at $z(t_i) = z(t_f) = (0, 0, \frac{\pi}{4})$. On the right, simulation of the path-following control on the classic Dubins path (in blue) and on the restricted Dubins path (in red). The boat's path and the restricted path completely overlap while the classic Dubins path is not feasible - the boat had no control and was pushed away by the wind.

without moving. The start pose $[0, 0, \frac{\pi}{4}]$ is considered, the inaccessible angle $\phi = \pi$, opposite to the wind taken, $\psi = 0$ (since the sailboat cannot go against the wind) and the range around that angle $\delta = \frac{\pi}{4}$.

For the rate of turn, the minimum rate of turn is roughly calculated from the model Eq.(11) and empirically established at $\rho = 14.48\text{m}$.

The produced path is a *LSRSL* path with a turn rate of $\rho = 14.48\text{m}$ as shown in Figure 4. The overall length is of 194.86m. The only other available path was a *RSLSR* path with an overall length of 194.92m. Since the start pose and the end pose are the same, the vehicle can start at any point in the path. This makes it useful when restricted in space, such is the case for station keeping. No classic Dubins path was available because of the restriction of angle. However, for comparison purpose, the same path-following control was simulated on the classic Dubins path with the same radius. The classic Dubins path is a single full turn, marked by the circle in black outline on the bottom of the right figure in Figure 4.

The control implemented to follow the path is a simple feedback control inspired by the line following algorithm [19]. This algorithm makes the path attractive to the sailboat. The algebraic distance is calculated to the closest point to the path:

$$e = \det\left(\frac{cL}{\|cL\|}, p - c\right) \quad (12)$$

Where c is the closest point to the path, θ_c the angle of the tangent vector to the closest point on the path and p the position of the boat. It is to note that θ_c is given by the algorithm as it gives the pose at each point of the path. This error is then used in the heading control:

$$\theta^* = \theta_c - 2 \frac{\gamma_\infty}{\pi} \text{atan}\left(\frac{e}{r}\right) \quad (13)$$

where $\gamma_\infty = \frac{\pi}{4}$ is the incidence angle of the sailboat and r the cutoff distance, the distance at which the boat should not go over. This proportional gain can be improved to counter stronger currents in water and errors in motion by adding an integral component. The following controls are applied to the rudder angle δ_r and sail angle δ_s , using their maximum values, respectively, δ_{r_max} and δ_{s_max} :

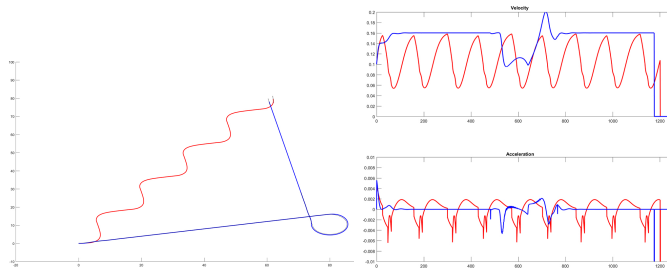


Fig. 5. Simulated path between a tacking strategy (in red) and the restricted Dubins path following control (in blue) on the left. The corridor of the tacking strategy is taken as 40m, adding more turn and slowing down the boat. In this particular scenario, the restricted Dubins path following control is faster than the tacking strategy, which is not generally the case. Measured velocity and acceleration of the corresponding simulation on the right. The acceleration is less stable in the tacking strategy case.

$$\begin{cases} \delta_r = \begin{cases} \delta_{r_max} \sin(\theta - \theta^*), & \text{if } \cos(\theta - \theta^*) > 0 \\ \delta_{r_max} \text{sign}(\sin(\theta - \theta^*)), & \text{else} \end{cases} \\ \delta_s = \delta_{s_max} \frac{\cos(\psi_{tw} - \theta^*) + 1}{2} \end{cases} \quad (14)$$

The resulting simulation is shown in Figure 4. The boat was capable of following the restricted path built, the red path shows the boat's path and in black the Dubins path built (classic and restricted). The path-following control on the classic Dubins path, which is a single full turn, could not be finished because of the wind pushing back the boat. If the wind was lower, the boat would have been able to finish the trajectory by using its inertia. Nonetheless, when the boat is against the wind, control is lost and the boat becomes uncontrollable at a lower velocity.

Further tests were done to compare this strategy to a tacking strategy, referred to as beating (zig-zagging to go upwind). Figure 5 shows the difference between the tacking strategy and the restricted Dubins path. The tacking strategy is almost always faster than the restricted Dubins path following control. But when comparing the acceleration of the boat, since the restricted Dubins path gives a smoother path, the boat seems to suffer less in mechanical disturbance and control. Also, in the tacking strategy, the boat will always face the wind at each tack, putting the sail in a flag position, which could hurt it and rip it in high winds.

To conclude on real-life application, considering errors in measurements, the path-following algorithm here could be improved to also work on more hostile environments such as high-wind and high water currents by adding integral and derivative gains to the feedback control.

VI. CONCLUSION

In this paper, a way of building a Dubins path when restricted to a range of orientation at all time was shown. Using bitangents of circles, the shortest path was built from one pose to another by adding a middle turn, making a *CSCSC* path. It was proven that, for a Dubins path with an added middle turn, whenever the difference between the bitangent angle and the angle leaving the first turn or the angle landing on the

final turn increases, the whole path length increases. By using this property, the shortest path was built by choosing angles that are closest to the bitangents, in the available domain of angles constrained by the restriction in orientation, and thus placing the middle turn correctly to have an overall shortest path. An application to a path-following algorithm using a sailboat model has shown the feasibility and the use of such a path in a position keeping scenario.

This method allows sailboats, and other vehicles, to navigate in restricted environments as efficiently as possible, distance-wise. The proof given can also be used for obstacle avoidance. Indeed, it is now known that whatever the angle is chosen on the first and final turn, the closer it is to the bitangent angle, the shorter the path will become. A vehicle could avoid an obstacle while staying as close as possible to the bitangent as possible. Different sailboat path-following controls could be used to answer to different requirements. For a normal vehicle, knowing the configurations, a control on turns and not a path-following method could be used. Future work can be done using this path in path-planning methods such as the Dubins Traveling Salesman Problem (DTSP) when restricted to certain orientations. Also merging this algorithm to the one proposed in [9] using displaced vectors could be considered to work in vehicles with both continuous and discontinuous speed.

APPENDIX

PROPOSITION OF THE SHORTEST PATH FOR A *CSCSC* PATH

This section will analyse the behaviour of the *CSCSC* path's length by analysing the distance between tangent lines of two circles. This study will show that for a *CSCSC* path, whenever the difference in angles between the first or the final turn and the corresponding bitangent angle increases, the whole paths length increases. This appendix does not take into account the restriction of angles since this study is on the more global problem of the length of a *CSCSC* path. Singularities exist near the bitangent angles which will be explained at the end of this study.

The notation will be simplified and modelled as in Figure 6. Since this work focuses on the first and last turn, the configuration order is considered as $C_1SC_mSC_2$. Configuration $C_1SC_mSC_2$ is simplified as C_1C_2 . Let θ_1 be the angle on the first circle C_1 and θ_2 the angle on the last circle C_2 of the tangents. C_1 is on the right side of C_2 on the figures. This was chosen arbitrarily, but it is of no concern for the proof.

The poses are placed so that the bitangents between C_1 and C_2 overlap the x-axis. Meaning that for a *LL* configuration, circles are placed at the same height so that the bitangent is at $\theta_1 = \theta_2 = \frac{\pi}{2}$, and in a *LR* configuration, the bitangent is placed at $\theta_1 = \frac{\pi}{2}$ and $\theta_2 = \frac{3\pi}{2}$ as shown in Figure 6. Because of the symmetric relations between the configurations, the rest of the configurations can be obtained by flipping those configurations. The radius of the turning rate is taken as a unit radius $r = 1$ and the distance on the x-axis between C_1 and C_2 are taken at 5 arbitrarily. While these last two parameters affect the results, it will be shown that the conclusion of the proposition is not affected.

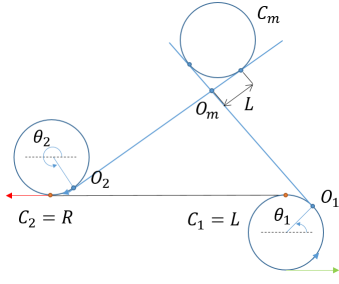


Fig. 6. Figure modelling the problem for the proof, in the specific configuration LR . Angles θ_1 and θ_2 measure the angle on the circle of the tangents.

The gradient of the length of the whole $CSCSC$ path is studied. Since the Interval Analysis tool will be used to analyse the results, an expression of the distance easily derivable must be found. As such, the parametric expressions of the tangent lines are used. The distance at which the two lines intersect must be studied. As they are algebraic, conditions are set to have the proper distance when they are positive. According to Eq.(7), t and t' must be found so that:

$$\begin{cases} O_{C_1}^{\theta_1} + t \begin{bmatrix} \cos(\tilde{\theta}_1) \\ \sin(\tilde{\theta}_1) \end{bmatrix} = O_{C_2}^{\theta_2} + t' \begin{bmatrix} \cos(\tilde{\theta}_2) \\ \sin(\tilde{\theta}_2) \end{bmatrix} \\ t \geq 0 \\ t' \geq 0 \end{cases} \quad (15)$$

where:

$$\tilde{\theta}_i = \theta_i + u_{C_i} \frac{\pi}{2}$$

This gives us multiple combinations of expressions. They are all accounted for the interval analysis study, one of which, for instance, becomes:

$$\begin{cases} t = \frac{\sin(\tilde{\theta}_1)dx - \cos(\tilde{\theta}_1)dy}{\sin(\tilde{\theta}_2 - \tilde{\theta}_1)} \\ t' = \frac{\cos(\tilde{\theta}_2)t + dx}{\cos(\tilde{\theta}_1)} \\ \cos(\tilde{\theta}_1) > 0 \\ dx = (O_{C_2} - O_{C_1}) \cdot \vec{i} \\ dy = (O_{C_2} - O_{C_1}) \cdot \vec{j} \end{cases} \quad (16)$$

Since a unit velocity is considered, the distance, or the time, at which the lines intersect is found using Eq.(16). The arc length of the added turn C_m , noted a_{C_m} , and the part of the trajectory added/removed from the intersecting lines L (shown in Figure 6) are added. The first and final arcs a_{C_1} and a_{C_2} are also added. This sum gives the whole distance of the $CSCSC$ path:

$$D(\theta_1, \theta_2) = t + t' + \tau 2L + a_{C_m} + a_{C_1} + a_{C_2} \quad (17)$$

where τ corresponds to the sign at which L is either added or removed from the trajectory:

$$\begin{cases} L & = r |\tan(\frac{\theta_2 - \theta_1}{2})| \\ a_{C_m} & = r |\theta_1 - \theta_2| \\ a_{C_1} & = r |\theta_1| \\ a_{C_2} & = r |\theta_2| \\ \tau & = \begin{cases} 1 & \text{if } |\theta_2 - \theta_1| < \pi \\ -1 & \text{else} \end{cases} \end{cases}$$

d_1 is notated as the first component of the gradient of $D(\theta_1, \theta_2)$ and d_2 the second component:

$$\vec{\nabla} D(\theta_1, \theta_2) = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = \begin{bmatrix} \frac{dD(\theta_1, \theta_2)}{d\theta_1} \\ \frac{dD(\theta_1, \theta_2)}{d\theta_2} \end{bmatrix}$$

To calculate the derivative of the added path L and a_{C_2} , it is important to note that they are of opposite sign: when L increases, a_{C_2} decreases. The absolute values are modified using the properties of the turns and the bitangents, replacing them with units of directions u_C . As such:

$$\begin{cases} \frac{dL}{d\theta_1} & = u_{C_1} u_{C_2} r \frac{1}{2} (1 + \tan^2(\frac{\theta_2 - \theta_1}{2})) \\ \frac{dL}{d\theta_2} & = -u_{C_1} u_{C_2} r \frac{1}{2} (1 + \tan^2(\frac{\theta_2 - \theta_1}{2})) \\ \frac{da_{C_m}}{d\theta_1} & = -u_{C_1} u_{C_2} \tau r \\ \frac{da_{C_m}}{d\theta_2} & = u_{C_1} u_{C_2} \tau r \\ \frac{d(a_{C_1} + a_{C_2})}{d\theta_1} & = u_{C_1} r \\ \frac{d(a_{C_1} + a_{C_2})}{d\theta_2} & = u_{C_2} r \end{cases}$$

For simplicity, the added path L and a_{C_m} will be neglected and discussed later. We call $\underline{D}(\theta_1, \theta_2)$ the distance without the added L and a_{C_m} path, with the corresponding gradient elements \underline{d}_1 and \underline{d}_2 :

$$\underline{D}(\theta_1, \theta_2) = t + t' + a_{C_1} + a_{C_2} \quad (18)$$

If only $t + t'$ was taken into account, one could see that the shortest path is not at the bitangent angle and that r might take over the distance $d_{C_1 C_2}$ and modify the overall gradient of $\underline{D}(\theta_1, \theta_2)$ and most importantly its sign. The added arcs counteract the parameter r by nulling its behaviour, making r and $d_{C_1 C_2}$ negligible towards the gradient's sign. As such arbitrarily chosen poses and parameters do not affect the conclusion of this proof.

Four different combinations of Eq.(16) are used and are derived to analyse the behaviour of $CSCSC$ path. The gradient of the expression $\underline{D}(\theta_1, \theta_2)$ from Eq.(18) is analysed using SIVIA with $\theta_1 \in [0, 2\pi]$ and $\theta_2 \in [0, 2\pi]$. The sets considered are:

$$\begin{cases} \mathbb{S}_1 = \{t < 0\} \cup \{t' < 0\} \\ \mathbb{S}_2 = \{d_1 > 0\} \cap \{d_2 > 0\} \cap \overline{\mathbb{S}_1} \\ \mathbb{S}_3 = \{d_1 < 0\} \cap \{d_2 < 0\} \cap \overline{\mathbb{S}_1} \\ \mathbb{S}_4 = \{d_1 > 0\} \cap \{d_2 < 0\} \cap \overline{\mathbb{S}_1} \\ \mathbb{S}_5 = \{d_1 < 0\} \cap \{d_2 > 0\} \cap \overline{\mathbb{S}_1} \\ \mathbb{S}_6 = \overline{\mathbb{S}_1} \cup \mathbb{S}_2 \cup \mathbb{S}_3 \cup \mathbb{S}_4 \cup \mathbb{S}_5 \end{cases} \quad (19)$$

For the results, the vehicle poses were arbitrarily taken as such to have the bitangents between C_1 and C_3 as simple as

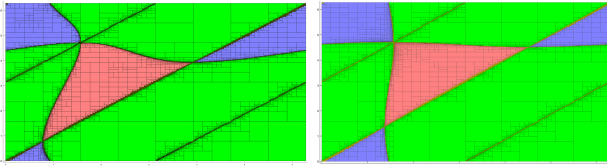


Fig. 7. The gradient of $D(\theta_1, \theta_2)$ in the LR configuration using SIVIA algorithm. In green, \mathbb{S}_1 . In pink, \mathbb{S}_4 . In light blue, \mathbb{S}_5 . In yellow, \mathbb{S}_6 i.e. uncertainties. On the left, $r = 5$ and on the right $r = 1$.

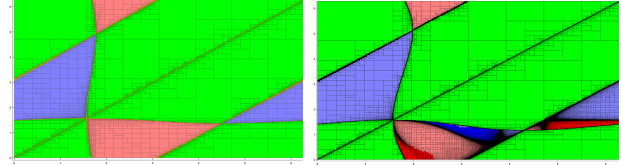


Fig. 8. Gradient of $D(\theta_1, \theta_2)$ in the LL configuration with radius $r = 1$ on the left. Gradient of the whole path length $D(\theta_1, \theta_2)$ in configurations LL with radius $r = 1$. Singularities appear near the bitangent angles.

possible. The distance between the poses and the radius are also taken arbitrarily. While these parameters affect the results, it doesn't affect the conclusion, as it will be shown later. The result for configuration LR is shown in Figure 7 for a distance between the poses of 10 and a radius of 5 with θ_1 in the x -axis and θ_2 in the y -axis. A comparison is shown in the same figure with a distance of 10 and a radius of 1.

In green, the set \mathbb{S}_1 , when the directed lines don't intersect. In pink, the set \mathbb{S}_4 . In light blue, the set \mathbb{S}_5 . Other conditions such as \mathbb{S}_2 (in red) and \mathbb{S}_3 (in dark blue) were calculated but returned empty with $D(\theta_1, \theta_2)$. Finally, in yellow, the set \mathbb{S}_6 i.e. the uncertainties.

In the configuration LR , the bitangent is at $\theta_1 = \frac{\pi}{2}$ and $\theta_2 = \frac{3\pi}{2}$ which corresponds to the top left side of the result in Figure 6 which the colour changes from pink to light blue, or from set \mathbb{S}_3 to \mathbb{S}_4 . Lines of uncertainties can be seen when $\theta_1 = \theta_2$ or with $\theta_1 = \theta_2 \pm \frac{\pi}{2}$ which represent when the two lines are parallel.

The set \mathbb{S}_4 shows that whenever θ_1 increases, the whole path' length increases while increasing θ_2 diminishes the path' length. In the particular domain of the set \mathbb{S}_4 in the LR configuration, increasing θ_1 gets it further from the bitangent angle, and increasing θ_2 gets it closer to the bitangent angle. The same observation can be made for the set \mathbb{S}_5 .

What this result shows, is that when the difference between θ_1 and θ_2 to the bitangent angle increases, $D(\theta_1, \theta_2)$ increases. This means that the global minimum is set at the bitangent. This shows that to have the shortest path, θ_1 and θ_2 must be as close to the bitangent angle as possible. Similar results can be derived from the configuration RL . Configuration LL are shown in Figure 8.

When adding L and a_{C_m} to the path and analyzing $D(\theta_1, \theta_2)$ from Eq.(17), singularities can be seen, shown in Figure 8. Whenever θ_1 or θ_2 is close to the bitangent, the place at which the middle turn has to be put should be as close to the median line between the two origins as possible to shorten the path length. This is due to the fact that the radius of the circle takes dominance over the path length. As such, these singularities happen when the intersecting lines intersect with one of the turn circles. These singularities don't disprove the proposition that when the path is further from the bitangent angles, the longer it is. It adds another information where the middle turn should be closer to the equipotential line between the two poses if possible.

The following propositions conclude this appendix:

Proposition 1: Whenever the difference between θ_1 or θ_2 to the corresponding bitangent angle increases in a $CSCC$ path, the whole path length increases.

Proposition 2: When either θ_1 or θ_2 is close to the bitangent angle and when the tangent lines of the circles intersect with one of the circles, placing the middle turn closer to the median line between the two origins of the circle shortens the whole path length.

REFERENCES

- [1] L. 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*, 1957.
- [2] H. H. Johnson, "An application of the maximum principle to the geometry of plane curves," in *Proceedings of the American Mathematical Society*, 1974.
- [3] X.-N. Bui, J.-D. Boissonnat, P. Soueres, and J.-P. Laumond, "Shortest path synthesis for dubins non-holonomic robot," in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*.
- [4] R. Penicka, J. Faigl, P. Vana, and M. Saska, "Dubins orienteering problem," *IEEE Robotics and Automation Letters*, 2017.
- [5] I. A. Hameed, "Coverage path planning software for autonomous robotic lawn mower using dubins' curve," in *2017 IEEE International Conference on Real-time Computing and Robotics (RCAR)*.
- [6] S. G. Manyam and S. Rathinam, "On tightly bounding the dubins traveling salesman's optimum," *Journal of Dynamic Systems Measurement and Control-Transactions of the Asme*, 2018.
- [7] J. Cao, J. Cao, Z. Zeng and L. Lian, "Optimal path planning of underwater glider in 3D Dubins motion with minimal energy consumption," *OCEANS 2016-Shanghai*.
- [8] J. A. Reeds and L. A. Shepp, "Optimal paths for a car that goes both forwards and backwards," *Pacific Journal of Mathematics*, 1990.
- [9] I. S. Dolinskaya and A. Maggiar, "Time-optimal trajectories with bounded curvature in anisotropic media," *International Journal of Robotics Research*, 2012.
- [10] A. Sadeghi Yengejeh, "Distributed task allocation and task sequencing for robots with motion constraints," Thesis, University of Waterloo, 2016.
- [11] X. Ma and D. Castanon, "Receding horizon planning for dubins traveling salesman problems," in *Proceedings of the 45th IEEE Conference on Decision & Control*, 2006.
- [12] H. J. Choi, "Time-optimal paths for a dubins car and dubins airplane with a unidirectional turning constraint," Diss. PhD Thesis, The University of Michigan, 2014.
- [13] B. Hérissé and R. Pepy, "Shortest paths for the dubins' vehicle in heterogeneous environments," in *2013 IEEE 52nd Annual Conference on Decision and Control*.
- [14] S. Manyam, S. Rathinam, D. Casbeer, and E. Garcia, "Shortest paths of bounded curvature for the dubins interval problem," *arXiv preprint arXiv:1507.06980*, 2015.
- [15] G. Elkaim and R. Kelbley, "Station keeping and segmented trajectory control of a wind-propelled autonomous catamaran," in *Proceedings of the 45th IEEE Conference on Decision and Control*.
- [16] A. M. Shkel and V. Lumelsky, "Classification of the dubins set," *Robotics and Autonomous Systems*, 2001.
- [17] U. Vautier, "Matlab source code for restricted dubins path," 2019. [Online]. Available: <https://github.com/plymouth-research>
- [18] J. Melin, "Modeling, control and state-estimation for an autonomous sailboat," 2015.
- [19] L. Jaulin and F. Le Bars, "A simple controller for line following of sailboats," in *Robotic Sailing 2012*, C. Sauzé and J. Finnis, Eds. Springer Berlin Heidelberg.