designs

MDPI

Article

On the Design of Smooth Curvature Tunable Paths for Safe Motion of Autonomous Vehicles

Gianfranco Parlangeli

Department of Engineering for Innovation, University of Salento, Via per Monteroni, 73100 Lecce, Italy; gianfranco.parlangeli@unisalento.it; Tel.: +39-0832-297301

Abstract

Navigation is an essential ability for autonomous systems, and efficient motion planning for mobile robots is a central topic for autonomous vehicle design and service robotics. Most path-planning algorithms produce reference paths with sharp or discontinuous turns, inducing several drawbacks during mission execution, such as unexpected inertial stress and strain on the mechanical structure, passenger discomfort, and unsafe and unpredictable deviation of the real trajectory with respect to the reference planned one. Oppositely, smooth and feasible trajectories are often desired in real-time navigation for nonholonomic mobile robots where the surrounding environment can have a dynamic and complex shape with obstacles. In this paper, we propose a novel technique for the generation of smooth, collision-free, and near time-optimal paths for nonholonomic mobile robots. The proposed method exploits the features of a set of tunable bump functions, with the goal of pursuing smooth reference curves with tunable features (such as curvature, or jerk) yet seeking a reasonable length minimality, thus combining the advantages of the two most adopted techniques, namely Bezier interpolation and Dubins curves. After a thorough description of the analytical methods, the paper is primarily concerned with the design and tuning methods of the path-planning algorithm. Both a graphical method and numerical investigations and examples are performed to fully exploit the algorithm potentialities and to show the efficiency of the proposed strategy.

Keywords: smooth motion planning algorithms; dubins vehicle; autonomous navigation

1. Introduction

Recent advances of modern technologies make tomorrow's world a place where humans and machines are more and more integrated in society. The topic of designing autonomous driving vehicles is a powerful example of integrated technologies and sophisticated algorithms with humans.

One key ability to reach autonomy for a mobile device is decision-making in planning its motion in the environment, namely its ability to find feasible paths in several diverse, dynamically varying environments, avoiding moving obstacles and minimizing the total travel length for achieving minimal travel time or energy consumption purposes. For this reason, the challenging problem of safe and autonomous navigation has surged as a central topic in the wide area of robotics and autonomous systems [1].

Navigation is a fundamental ability for autonomous systems. However, most path planning algorithms generate a reference path with many sharp or discontinuous turns, inducing several drawbacks during mission execution, such as unexpected inertial stress and strain on the mechanical structure, passenger discomfort, and unsafe and unpredictable

Academic Editors: Yuping He and Junnian Wang

Received: 4 February 2026
Revised: 8 March 2026
Accepted: 27 March 2026
Published: 7 April 2026

deviation of the actual trajectory. Oppositely, smooth and feasible trajectories are often desired or requested for an overall satisfactory behavior of the feedback system; most convergence results assume indeed that the reference signal is smooth [2,3].

Among the most studied models representing the vehicle motion, the 'Dubins vehicle' [1] models a basic yet effective kinematic behavior. In essence, it describes the motion of a point in the plane having fixed forward speed and limited steering angle. Its popularity is related to its simplicity and that it is well suited to describe the movement of any vehicle whenever it is not required to make hard braking, slowdown or stop operations, which are often undesired or impossible in most applications.

The adoption of a Dubins model for path planning leads to navigation solutions in many diverse fields and in air, ground and marine applications. In the relevant field of aerial applications, the judicious management of tasks such as surveillance, reconnaissance, and delivery operations require careful path planning [4]. Aircraft, especially those with limited maneuverability, may encounter a precise way point crossing during specific tasks of flight mission [5], such as air-to-air refueling or navigating through airspace restrictions. In such scenarios, a safe, precise and efficient path planning is of paramount importance. Analogous considerations hold for marine and underwater applications, and hence the Dubins vehicle is often the chosen model for path planning [6].

However, despite the popularity and ubiquity of Dubins solutions in a variety of path-planning algorithms, there are several drawbacks that make this choice critical, and in the last decades there have been more and more attempts by researchers to overcome them through more sophisticated algorithms.

One peculiarity of Dubins paths which can produce serious drawbacks in practical applications is that length-optimal solutions naturally have non-differentiable curvature. Indeed, a characteristic feature of these paths is that the curvature switches between three values, namely zero and maximum left and right turn, thus the corresponding curvature profile has a rectangular shape.

The temporal discontinuity of the curvature function is usually a negative feature for the performance of a tracking control system, which in general requires a high degree of smoothness of the reference to guarantee good tracking performance (see e.g., [2,3]). What happens in practice when discontinuous reference signals are assigned is that the tracking error shows an abrupt deviation at the discontinuity time instants that converges to zero for the stabilizing action of the control system. It is worth remarking that, in all the application examples described above and many others, the occurrence of jumps in the tracking error is negative in terms of safety (since the real trajectory deviates from the expected one) and mission goal achievement (several discontinuity time instants may produce a deviation which is not recoverable during the mission execution).

For all these reasons, in the last decades, the scientific community has tried to extend the classical path planning based on the Dubins model in several different directions, with the goal of ensuring the continuity and differentiability of the reference path, as we briefly describe in the following.

1.1. Literature Review

One first attempt to generalize the Dubins vehicle model to exclude discontinuities in the curvature shape is to extend the state of configurations including the curvature rate. This model was introduced in the seminal paper [7], and widely studied later by several authors. In this framework, clothoid curves emerge as extremal curves of length-optimal problems, but solving the optimal control problem shows that an optimal trajectory containing one linear segment must have an infinite concatenation of clothoids [7], so that it is not convenient to adopt such mathematical optimal solution in practice. Notwithstanding this negative result about optmality, clothoid-shape reference paths have been largely investigated [8]. A recent detailed paper on clothoid is [9]. In the recent paper [10], clothoid-shaped primitives are integrated with a two-way D* for real-time navigation of nonholonomic mobile robots in dynamic and complex environments while avoiding obstacles.

However, there are several applications in which clothoid-shaped reference paths are not well suited, and there are several research activities trying to generalize the reference path shape to more general and smoother profiles.

In aerial and marine applications, vehicle propulsion and motion is based on the interaction with air or water, and the inertial drag of the vehicle body allows to track trajectories with slow curvature rate [11,12]. Motion control for these vehicles is therefore exceptionally hard, and highly smooth reference paths are often a primary choice [13].

More in detail, vehicle motion in marine or aerial applications is generated through fluidodynamical interactions, so that the tracking error of a discontinuous reference signal is usually large, and it makes the vehicle susceptible to large deviation of the actual motion from the expected one. This phenomenon can be a serious drawback in terms of safety, since the avoidance of obstacles is guaranteed by the attainment of the reference path, so that large deviations can lead to unpredictable collisions. Moreover, they induce a huge transient phase that is susceptible to unexpected oscillations and possible loss of stability. As for example, considering marine applications, in the paper [2] a dynamic control law making the system converge to the origin at an exponential rate is proposed, under the assumption that a pre-defined twice differentiable reference trajectory is set. In the framework of aerial applications, in the paper [3] the tackled trajectory tracking problem and the convergence results are achieved under the assumption that a sufficiently smooth desired trajectory with bounded derivatives is provided. Under this hypothesis, a controller is designed such that all the closed-loop signals are bounded and the trajectory tracking error converges to an arbitrarily small neighborhood of the origin.

For these reasons, most path-planning algorithms in this framework are designed through Bezier curves [14,15]. Their advantages are generality, conceptual simplicity, and high continuity. However, collision, via-point crossing and curvature-constraint checking must usually be done numerically in the case of such primitives, which leads to significant computational cost. We discuss the main advantages and drawbacks of using Bezier curves in Section 2.2.

Finally, one different novel yet relevant application field pushing toward the use of path-planning algorithms producing smooth reference profiles is service robotics. Indeed, more and more often robotics is adopted for human transport and assistance, ranging from urban mobility to personal assistance through automated wheelchairs. In most of the applications within this framework, robots interact with fragile humans, or they may carry delicate or dangerous items. Reference paths with discontinuous curvature are inconvenient, as their tracking would produce discomfort and possible accidents. Wheeled mobile robots following a path with continuous curvature may also benefit on wheels slippage reduction and low odometry errors, since transitions are softer with constant curvature rates. It is worth remarking that slippage phenomena are detrimental to precise control, and they can be compensated adopting sophisticated ad hoc algorithms [16]. On the contrary, smooth trajectories are often desired for a pleasant and safe mission execution. In all the applications where human users are involved, comfort is adopted as a criterion for design. In [17] propose a tunable trajectory planner generating G3 curves for road vehicles. The path-planning algorithm in [18] produces smooth profiles for a better tracking performance of the vehicle.

1.2. Paper Contribution

Considering all the above discussion, the goal of this paper is to derive an algorithm for the construction of reference paths having smooth curvature along the way, with the opportunity to set the curvature bound together with the first and second derivatives, and ensuring the via-point crossing. The paper [13] is inspired by analogous considerations, and the proposed solution allows building smooth reference trajectories. However, the solution proposed in this paper significantly differs from the above solution in terms of presence of parameter values that allow a tuning of the shape of the curvature function to meet possible constraints on the curvature up to the second derivative. Also, the reference path construction follows a different logic.

One main purpose of the proposed research activity is the generation of reference trajectories that allow safe and comfortable behavior to a vehicle having the ability to track them. The two terms, namely safe and comfortable, related to different goals. The former refers to a vehicle traveling free from accidents and undesired deviations, and it is strictly related to the feasibility of a reference trajectory by the vehicle. The latter refers to reference trajectories whose tracking does not generate lateral accelerations and impulsive inertial forces [19]. Considering all these diverse facets of control goals, we seek a reference curvature profile with tunable bounds on the maximum curvature together with first and second derivatives, considering that such bounds have been set as a reasonable trade-off between all the above features.

2. Problem Framework

In this Section, we fully describe and state the problem tackled in this paper as an advancement in the topic of local path-planning algorithms. We start with the genuine description of the most relevant features of a path-planning algorithm, with focus on the applications described above. Then we describe the advantages and drawbacks of the two most adopted approaches, namely Bezier curves and Dubins paths, and finally we formally state the tackled problem.

2.1. Path Planning for Mobile Robots

Path planning algorithms are an essential part of a control system for a mobile device and autonomous vehicles [1]. It typically receives as inputs an a priori knowledge of the environment (e.g., maps), some knowledge provided by sensors (e.g., localization) and the desired goal to achieve (e.g., achieve a given pose in a remote location of the map), and it provides as output a reference trajectory to be tracked by the mobile device so to accomplish the assigned task [1]. To solve this problem, path-planning algorithms can be classified as *global* or *local path planning* algorithms. Global path-planning algorithms exploit pre-computed environmental data (e.g., maps) to compute the optimal path from start to destination before the mission execution, thus computing the most efficient route from start to goal. Traditional solving methods make use of graph search tools or optimization-based search approaches (e.g., particle swarm optimization, Ant Colony Optimization), while modern approaches take advantage of artificial intelligence methods (e.g., deep learning). Oppositely, local path-planning algorithms are concerned with the problem of connecting two positions exploiting real-time data from sensors to continuously update the path and distance to the target, while incorporating objectives for trajectory smoothness and minimal jerk. Most path-planning algorithms are usually organized in two stages, taking advantage of the features of each of the two approaches. One stage is focused on the selection of a set of convenient way points in the map and it is useful to build a rough route, the other stage being establishing the connections between the selected way points through convenient *primitives* or *interpolating curves* to build a viable path to be tracked by the device [1]. Besides the spatial profile of the reference path, the output of a path planner and one major goal of a navigation system is the generation of a *guidance law* for the vehicle, namely the temporal profile that should be impressed to the input of the vehicle in order to follow a reference path leading to goal achievement.

Overall, the path-planning algorithm is designed to consider some basic performances. The generated path must be *safe*, namely free from collisions with the environment. It must be *feasible* by the mobile device, so that it can be asymptotically tracked by a real robot through a suitable control system algorithm. It additionally must be reasonably *short*, namely having length minimality properties, which is a desired feature for several reasons, e.g., for fuel savings, which is important for an autonomous device to minimize the need for fuel. Finally, for recent applications, the algorithm must be sufficiently *fast*, namely the time needed for its execution should be negligible so that it can be run in real-time when the device moves in a dynamic environment, and it must eventually update the reference path due to perturbed conditions.

However, most path-planning algorithms produce reference paths made of straight lines connected through sharp turns [20]. Such paths are not a good choice for robot motion in many modern applications of practical interest, as we detail later in Section 2.2. The main reason is that a mobile device cannot make these sharp turns suddenly when moving at cruise speed, and it would be necessary to stop or slow down to track them. Alternatively, the tracking error shows deviations when discontinuity points occur.

Unfortunately, the former operational way to track the reference path, namely adjusting the cruise speed for tracking purposes, is not possible in most aerial or marine applications, or they turn out to be highly inconvenient for road or ground vehicles for inducing user discomfort or inertial accelerations that solicit undesired unmodeled dynamics which may perturb the system even to instability.

In the paper [21], continuous curvature reference paths are built to attain human comfort together with other goal achievement.

2.2. Smoothing Bezier Curves and Dubins Primitives

Before discussing the proposed approach in detail, in this Section we describe the advantages and drawbacks of the two main classical tools for generating reference paths which are mostly adopted as local solvers in the guidance algorithms for vehicles, namely the interpolation of two or more points using Bezier curves and Dubins primitives. The outcome of this investigation allows us to focus on the features that our algorithm must hold, while avoiding the main drawbacks associated to the use of traditional approaches.

One of the most popular methods to build reference paths is through the Bezier curve interpolation [22]. These parametric curves make use of 'control points' to define the shape of the generated curve. In essence, they exploit Bernstein polynomials as follows. For an assigned set of $n+1$ points $\mathbf{P}_0, \mathbf{P}_1, \ldots, \mathbf{P}_n$ called *control points*, the interpolating Bézier curve is defined as:

$$\mathbf{p}(t) = \sum_{i=0}^{n} B_i^n(t)\mathbf{P}_0, \qquad t \in [0,1], \tag{1}$$

where the blending functions $B_i^n(t)$ are the Bernstein polynomials equal to:

$$B_i^n(t) = \binom{n}{i}(1-t)^{n-i}t^i, \qquad i \in \{0,1,\ldots,n\}. \tag{2}$$

Since their introduction nearly one century ago, their use has been intense in various topics, ranging from computer graphics and animation to robotics [22].

https://doi.org/10.3390/designs10020042

Bezier curves success is undoubtedly related to its ease of use, its smoothness properties, and its flexibility and ease of shape modeling through a tuning of the relative control points.

Unfortunately, both length optimality and bounded curvature properties of a path for vehicles like mobile robots cannot be obtained with analytical methods, but heuristic algorithms are adopted [23]. In this framework, the problem is usually translated into finding the optimal placement of control points or modifying curve parameters, often using heuristic methods to find the actual maximum curvature, which isn't easily found in closed form.

Another drawback in the use of this method is the exact crossing of the via points, which cannot be directly guaranteed and requires specific attention [24].

2.3. Dubins Vehicle Model

We here briefly mathematically describe the widely adopted Dubins vehicle model, and we introduce the notation that we use along the paper.

The Dubins car model describes the kinematics of a mobile device moving forward at constant velocity with limited steering capability. Consider a particle in the $x-y$ plane moving according to the following kinematics:

$$\begin{cases} \dot{x} = \cos\theta, \\ \dot{y} = \sin\theta, \\ \dot{\theta} = k, \\ ||k|| \leq \Omega; \end{cases} \tag{3}$$

where Ω is a known constant. From a geometric perspective, Equation (3) describes a vehicle moving only forward at a constant speed equal to one and limited steering ability with a maximum turning radius Ω^{-1}. Equation (3) is referred to as the *Dubins' car* model [1]. In this framework, a path is called *feasible* for a Dubins vehicle if it can be generated through (3), so that feasible curves are essentially those plane curves with curvature lower than Ω.

Consider Figure 1. Vector $\mathbf{q} = (\boldsymbol{P}, \theta)$ is called the *pose* or *configuration* of the vehicle described in (3), $\boldsymbol{P} = (x, y)$ representing the position of the vehicle in the Euclidean plane, $\theta \in [0, 2\pi)$ its orientation (see Figure 1. In this basic framework the angular speed u is assumed to be the control signal as input of the vehicle.

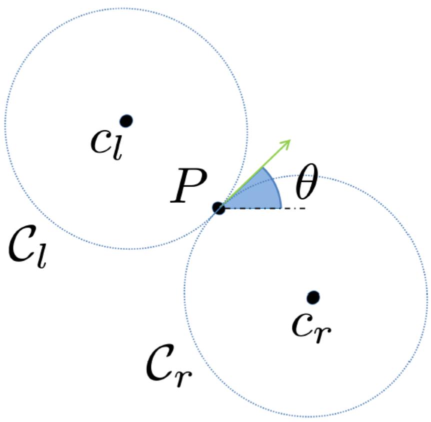

Figure 1. Graphical sketch of the left and right turning circles and their centers for an initial configuration $\mathbf{q_i} = (\boldsymbol{P}_i, \theta_i)$.

When dealing with Dubins vehicles, it is common to graphically draw two circles of radius Ω^{-1} tangent to $\mathbf{q}$ in $\boldsymbol{P}$ that we call resp. *left circle* and *right circle*, each corresponding to the sharpest turn.

The use of Equation (3) is popular in mobile robotics as it provides a simple yet effective kinematic model for a variety of mobile devices and vehicles. One main topic in path planning is the use of length-minimal curves connecting two given poses and satisfying (3).

This mathematical problem was stated by L.E. Dubins in [25], where he proved that an optimal path is made of at most three segments, having the shape of circular arcs at the given poses and either a straight line or a circular arc of maximum curvature in the middle.

In the following, a straight segment is denoted by S and a generic circular path is denoted by C, or by R or L if it is defined by the right or left turning direction, respectively.

Using this notation, it is common to state that length-optimal path is one among the six possible path types $\{LSL, RSR, RSL, LSR, RLR, LRL\}$ cases, some of them eventually of length zero.

However, if the initial and final points are sufficiently far apart, and specifically more than $4\,\Omega^{-1}$, CCC profiles are never optimal. Moreover, CCC profiles are inconvenient in most applications since small variations on the terminal configurations produce completely different solutions, and in practice, the recalculation of a reference path during a mission execution update produces a completely different reference path, with possible serious drawbacks on the mission prosecution execution. For the above reasons, in this paper we assume that each pair of points is rather far apart and CCC paths do not occur.

Despite the fact that (3) is one of the most adopted models for trajectory generation, there are possible serious drawbacks in its use and several attempts have been made over the years to overcome them. One main debated topic is the discontinuity of the curvature with respect to arc length. Indeed, a curve made of interconnections of linear segments and circular arcs has a piecewise constant curvature shape, showing jumps between the three constant values $0, \frac{1}{R}, -\frac{1}{R}$. As a result, the generated reference trajectories cannot be tracked by the mobile device when discontinuities occur, and the real motion of the mobile device may significantly differ from the profile one would expect in advance considering the prevision provided by the reference profile. This unpredictable behavior may have serious drawbacks in terms of unexpected collisions and, in general, for safety purposes.

One possible approach to impose curvature continuity is to extend the state space including the curvature itself [7,8]. In this case, the Dubins model (3) is extended to:

$$\begin{cases} \dot{x} = \cos\theta, \\ \dot{y} = \sin\theta, \\ \dot{\theta} = \kappa, \\ \dot{\kappa} = \sigma, \\ |\sigma|| \leq \Omega; \end{cases} \tag{4}$$

Unfortunately, in this framework, the solution curves of the length-minimal optimal problem do not represent an interesting option for path planning. Indeed, in the meticulous work of [7] it is shown that optimal curves including a straight segment must have a non-finite concatenation of clothoid pieces, and this mathematical feature is not convenient for path planning. Nevertheless, clothoid-shaped curves have been largely adopted and studied for path planning [8].

A possible way to seek more practical solutions for path planning is to modify the cost function including other goals together with length minimality, as for example keeping bounded accelerations for avoiding large inertial effects at the occurrence of corner points, or enhancing user comfort in service mobile robotics and human transport. In the recent papers [17,18,26] the authors adopt an even more extended model than (4), including also the curvature rate

$$\begin{cases} \dot{x} = \cos\theta, \\ \dot{y} = \sin\theta, \\ \dot{\theta} = \kappa, \\ \dot{\kappa} = \sigma, \\ \dot{\sigma} = \rho, \\ |\rho|| \leq \Omega; \end{cases} \tag{5}$$

and a cost function built as a trade-off between path length, comfort and inertial stress of the structure [17]:

$$\mathcal{C} = \int_0^{T_f} (w_a \mathcal{C}_a + w_J \mathcal{C}_J + w_y \mathcal{C}_y + w_t) dt \tag{6}$$

where $C_a = |a_N(t)|^2 + |a_T(t)|^2$ represent the squared magnitude of the acceleration, $C_J = J_N(t)^2 + J_T(t)^2$ is the squared magnitude of jerk, $C_y = \left(\frac{d\theta(t)}{dt}\right)^2$, while w_a, w_J, w_y are weights to be set considering the relative importance of each feature.

In this paper, we seek the same ambitious goal, but we adopt a different design strategy. We believe that adding higher order derivatives of the input in the model to guarantee a prescribed degree of smoothness of the input is not the best choice in view of the negative results in terms of optimal solution described in [7]. In that paper, the authors conclude that the extension of the model to the first derivative of the input leads to optimal solutions having infinite switchings among the three extremal curves (linear, circular and clothoid shapes), which is not interesting from an engineering point of view. In practice, the common choice when dealing with continuous curvature path planning is to find the best approximation of the discontinuous Dubins optimal path through the use of curvature functions with a desired temporal profile.

In this paper we follow this latter approach, namely we seek the best approximation of a discontinuous Dubins path, but we impose a smooth curvature profile having a shape which is compliant with the motion generation system constraints and the requirements stated above. Moreover, we introduce several tunable parameters in order to impose the bounds on the curvature shape according to a specific request.

The use of bump functions to build artificial potential field functions for safe navigation has been recently studied in [27].

Path Continuity

In this paper, for all the reasons explained above, we are interested in reference paths with high smoothness property. For an appropriate definition and use of these properties, we refer to the milestone work of Prof. B.Barsky and his coauthors [28]. Some basic definitions and properties on parametric curves and the related smoothness properties are reported in Appendix A. The interested reader can consider the references on $\mathbb{G}^p$ [29,30] for further insights.

2.4. Problem Statement

In this paper, we adopt the classical Dubins model (3), but we consider that the curvature cannot be set arbitrarily, but it is constrained by a function of the arc length $k(s) = \mathcal{F}(s)$, where $\mathcal{F}(\cdot)$ is a parametric function encoding the shape of the curvature, having the properties of being a smooth function with tunable maximum value of curvature together with its first and second derivatives.

The shape of function $\mathcal{F}(s)$ dictates the curvature profile, and so it is set to take in to account for several diverse features of the curvature profile. On one hand, the shape of $\mathcal{F}(s)$ allows us to manage actuator dynamics and/or fluidodynamic interactions which accomplish thrust motion. Setting a curvature profile which is compliant to these phenomena allows to produce reference paths which can be effectively tracked through bounded inputs.

Moreover, function $\mathcal{F}(s)$ can be set considering those features related to inertial effects, ranging from avoiding undesired inertail forces and accelerations to avoid unpleasant effects in terms of comfort.

It is worth mentioning that the linear-shaped curvature profiles discussed above, which allow to approximate Dubins paths through clothoids, can be considered as a special case of function $\mathcal{F}(s)$ having linear structure $\mathcal{F}(s) = ks$; however, we believe that the shape of the functions $\mathcal{F}(s)$ proposed in Section 3 has better structure, being smooth and tunable in terms of curvature together with its first and second derivative.

We are now ready to state the problems addressed in this paper.

Problem 1. *Given two points in the plane, $\mathbf{P}_1$ and $\mathbf{P}_2$, assuming that the orientations in $\mathbf{P}_1$ and $\mathbf{P}_2$ are given and that the initial curvature and curvature rate in such points are fixed to zero, find a smooth path $\mathcal{P}$ connecting them.*

Problem 2. *Given a set of points in the plane $\mathbf{P}_1, \ldots, \mathbf{P}_n$, assuming that the orientations of the first and the last points, $\mathbf{P}_1$ and $\mathbf{P}_n$ are given and that the steering rate curvature and curvature rate in such points are free, find a smooth path $\mathcal{P}$ connecting $\mathbf{P}_1$ and $\mathbf{P}_n$ and crossing through the intermediate points.*

The above two problems are similar, but they have some differences in their resolution, as discussed in Section 4. It follows that they constitute two basic problems which are both useful for adopting the proposed path-planning algorithm in a wide range of applications.

It is worth noting that our approach is analogous to that of the paper [13], where the authors adopt smooth transition functions, but these are not explicitly parametrized in terms of arc length. A first attempt to study a problem solution in the framework of this paper was performed in [31].

In the following section, we assume that the optimal path type has been selected using an algorithm for Dubins path classification, as in [32] for the connection of two poses as in Problem 1 or its generalization [33] for the additional crossing of a via point for the resolution of Problem 2. The goal of this paper is to deduce a smooth reference path connecting start and goal and satisfying prescribed values of curvature together with its first and second derivatives, thus representing a smooth approximation of the classical Dubins path.

3. Smooth Curvature Profile Design

In this section, we introduce the class of functions that we adopted as a basis for the proposed smooth approximation approach.

The goal of this paper is to build some tailored functions encoding the shape of the desired path reference profile according to the following logic. The curvature is set in relation to the arc length, $\kappa = \kappa(s)$, thus expressing its Cesaro equation. Function $\kappa(s)$ is built as a piecewise function on compact intervals where it takes either one of the constant values between $\{0, \Omega, -\Omega\}$, or a smooth connection among them that is constructed as the composition of two basic functions $f(x)$, $g(x)$, $\kappa(s) = f(g(s))$, $f(x)$ being a smooth monotone function from zero to one, and $g(x)$ set to 'shrink' the domain to a desired interval.

Considering the above structure and logic, take the following functions:

$$f(x) = a\tanh(cx) + b \tag{7}$$

$$g(x) = \tan(dx + f) \tag{8}$$

with parameters $a, b, c, d, f \in \mathbb{R}$. Function $f(x)$ smoothly and monotonically connects $b - a$ to $a + b$, so that in the following we set either $a = b = \frac{1}{2}$ in case of transition from 0 to 1, or $a = 1, b = 0$ for transition from -1 to 1. Function $g(x)$ monotonically maps the interval $\mathcal{I} = (\frac{-\pi-2f}{2d}, \frac{\pi-2f}{2d})$ to the real line. A graphical sketch of these functions is plotted in Figure 2.

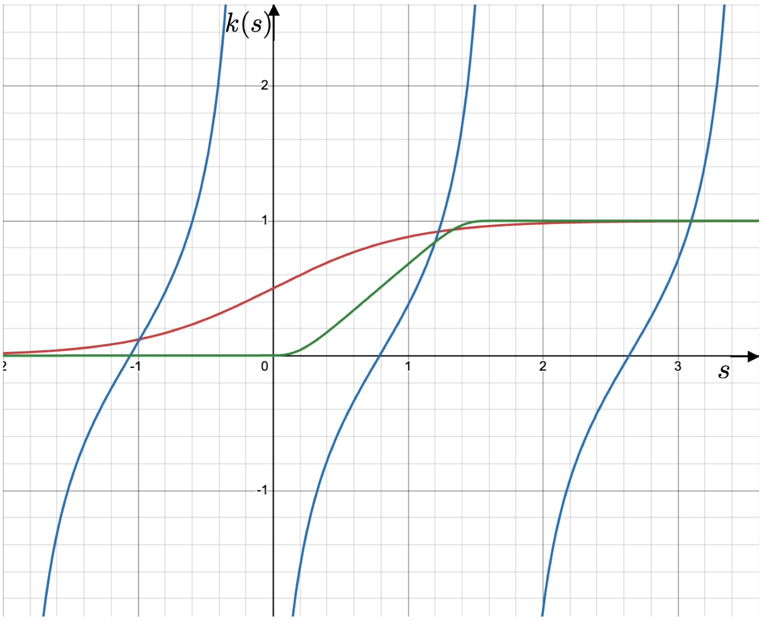

Figure 2. Graphs of function $\mathcal{F}$ in (9) with $a = b = \frac{1}{2}$ together with the constitutive functions $f(x)$ and $g(x)$. Function f is plotted in red, $g(x)$ in blue, and $\mathcal{F} = f(g(x))$ is plotted in green.

By suitably combining the above functions and tuning the related parameters, in the following we consider a curvature profile as

$$\kappa_{0-1}(s) = \begin{cases} 0 & \text{if } s \leq \dfrac{-\pi - 2f}{2d} \\ \dfrac{1}{2}(\tanh(c\tan(ds + f) + 1)) & \text{if } s \in \mathcal{I} \\ 1 & \text{if } s \geq \dfrac{\pi - 2f}{2d} \end{cases} \tag{9}$$

in the case of transition between 0 and 1, thus useful for connecting a straight line to a circle, or:

$$\kappa(s)_{1-1} = \begin{cases} -1 & \text{if } s \leq \dfrac{-\pi - 2f}{2d} \\ (\tanh(c\tan(ds + f))) & \text{if } s \in \mathcal{I} \\ 1 & \text{if } s \geq \dfrac{\pi - 2f}{2d} \end{cases} \tag{10}$$

for transitions between -1 and 1, thus useful for changing turning direction.

It is worth noting that steering operations not reaching the maximum curvature value are also possible, and the curvature profile adopted for this basic steering in the framework of this paper is:

$$\kappa_{0-a}(s) = a \cdot \kappa\left(\frac{s}{a}\right) \tag{11}$$

with $a \in (0, 1)$, which allows to reach the same maximum derivative of (9) while reducing the maximum value of a.

Functions (9)–(11), together with their mirrored versions $\kappa_{1-0}(s) = \kappa_{0-1}(-s)$, $\kappa_{0-a}(s) = \kappa_{a-0}(-s)$, are adopted in this paper as a smooth transition between constant values as described in Section 4.

Discussion on Criteria for Parameters Tuning

Function $\mathcal{F}$ can be tuned through five parameters a, b, c, d, f. However, they have a different impact on the curvature profile. Parameters a, b are set to impose the boundary values. Oppositely, parameters c, d, f are set to adjust the curvature shape with respect to arc length and hence they are the way to design the best trade-off among length, feasibility and comfort. More in details, they are set according to the following rules.

Parameters d and f are set to adapt any desired interval to the transition interval $\mathcal{I}$, namely for a prescribed transition interval $\mathcal{I} = \{s_1, s_2\}$ we set $d = \frac{\pi}{s_2 - s_1}$ and $f = -\frac{\pi}{2}\frac{s_2 + s_1}{s_2 - s_1}$. Conversely, for a desired function $\tan(dx + f)$ to be assigned, the inverse relations

$$\begin{cases} s_1 = -\dfrac{f + \pi/2}{d} \\ s_2 = -\dfrac{f - \pi/2}{d} \end{cases} \tag{12}$$

provide the values defining the $\mathcal{I}$. As a general rule, parameter f shifts both s_1 and s_2 of the same amount, while parameter d sets the distance between s_1 and s_2, being inverse proportional to $s_2 - s_1$ (the higher d, the narrower the interval $s_2 - s_1$).

Finally, parameter c magnifies the values provided by $f(x)$, and it is strictly related to the slope of $\kappa(s)$. It comes out that both c and d modify the slope of $\kappa(s)$, the difference between them is that c changes the slope of $f(g(s))$ keeping the same abscissa values, while d scales the x-axis and hence modifies the width of the transition zone.

To have a deeper insight, we study the features of the first and second derivative of the transition function (9). The analytical expression is developed in Appendix B, and it is possible to verify that $\dfrac{dk(s)}{ds}$ is a bump function. Here we discuss the structure of the functions derived in Appendix B through their graph.

Figure 3 shows the plots of these functions for $c = 0.9$, $c = 1$ and $c = 1.1$. From Equation (A4) and considering the plots in Figure 3, it is easy to see that, for $c \geq 1$, $\dfrac{dk(s)}{ds}$ has one single maximum equal to acd in the midpoint of $[s_1, s_2]$, or equivalently $\dfrac{d^2k(s)}{ds^2}$ has one only zero crossing, otherwise $\dfrac{dk(s)}{ds}$ has multiple stationary points and $\dfrac{d^2k(s)}{ds^2}$ several zero crossings. In terms of curvature shape, this implies that function $f(g(s))$ has a single inflection point at the midpoint of $[s_1, s_2]$ in the interval for $c \geq 1$, otherwise not.

Starting from the above discussion, we now describe the tuning criteria for a, b, c, d, f in order to build a function $\mathcal{F}(s)$ which is compliant with the motion generation and planning system contraints in a common case, the other cases can be treated analogously. Consider that we are interested in selecting a curvature profile as in (9) that is compliant with the constraints

$$\begin{cases} \kappa(s) \leq \bar{k}, \\ \frac{d}{ds}k(s) \leq \bar{\sigma}, \\ \frac{d^2}{ds^2}k(s) \leq \bar{\rho}, \end{cases} \tag{13}$$

which encode the geometry of a turn. Considering the first constraint and thus transitions of $k(s)$ from 0 to $\bar{k}$, we set $b - a = 0$ and $a + b = \bar{k}$, so that $a = b = \frac{\bar{k}}{2}$.

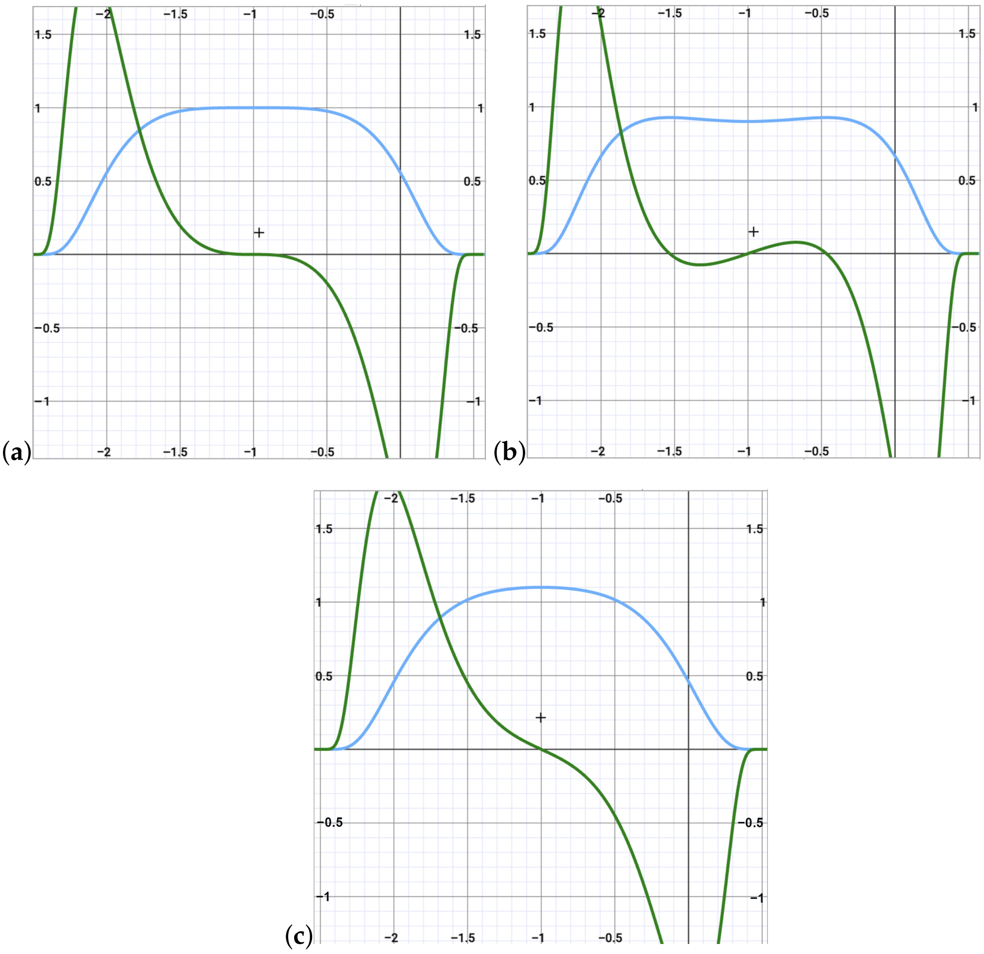

Figure 3. Graphic representation of $\frac{d\mathcal{F}}{ds}$ and $\frac{d^2\mathcal{F}}{ds^2}$ for different values of parameter c. Function $\frac{d\mathcal{F}}{ds}$ is depicted in light blue, $\frac{d^2\mathcal{F}}{ds^2}$ is in green; (**a**) c = 1, (**b**) c = 0.9 (**c**) c = 1.1.

As for $\frac{d}{ds}k(s) \leq \bar{\sigma}$, observe that, in view of the above discussion, for $c \geq 1$ it holds $\frac{d}{ds}k(s) \leq acd$, so that imposing $acd \leq \bar{\sigma}$ makes the second constraint of (13) satisfied. Overall, first and second relation are compactly expressed by $cd \leq \frac{2\bar{\sigma}}{\bar{k}}$. As for the last constraint of (13), from (A5) in Appendix B it is possible to state that $\frac{d^2}{ds^2}k(s) \leq \alpha d^2 c^2$ where α is a constant that can be handily computed, and this in turn reveals that imposing $\alpha d^2 c^2 \leq \bar{\rho}$ makes also the third constraint of (13) satisfied.

To keep all the above relations compact, define $\Xi = \max\{\frac{4\bar{\sigma}^2}{\bar{k}^2}, \frac{\bar{\rho}}{\alpha}\}$, that allows to encode both the second and the third contraint into the relation $d^2c^2 \leq \Xi$. Now, considering that it is better to impose $c \geq 1$ and that Equation (12) expresses that a larger d corresponds to a thinner transition width (which is equal to π/d), a possible set of parameters satisfying (13) is:

$$\begin{cases} a = b = \frac{\bar{k}}{2}, \\ c = 1, \\ d = \Xi^{1/2}, \\ \Xi = \max\{\frac{4\bar{\sigma}^2}{\bar{k}^2}, \frac{\bar{\rho}}{\alpha}\}. \end{cases} \tag{14}$$

The above set of parameters allows the path-planning algorithm to generate reference trajectories compliant with the set of constraints (13), and it turns to guarantee prescribed levels of comfort and inertial stress of the structure encoded by a cost function as in (6). It is worth remarking that the above setting is just one of the possible choices, and we believe that our framework provides tools for a wide set of possibilities, according to diverse control goals, and this reveals the flexibility of the proposed approach.

4. Smooth Approximation of a Dubins Path

In this section, we exploit the functions previously introduced to compute any Dubins path. As a first step, we analyzed the geometry of a steering operation associated to a curvature profile as in (9). The outcome of this analysis constitutes the basic block for the proposed smooth approximation.

Geometry of a Turn Associated to a Curvature Profile as in (9)

We start the analysis comparing the motion of a rectangular curvature shape with the proposed smooth profile (9). Figure 4a shows the curvature profile for a left turn to reach the maximum curvature starting from zero, which corresponds to the graph of (9) for $c = 1$, $d = \pi$, $f = \frac{\pi}{2}$. In Figure 4b the motion of either an ordinary Dubins vehicle is plotted when moving according to $0 - 1$ curvature (red line), and that of a vehicle following curvature profile as in Figure 4a. It is interesting to note the displacement and direction deviations, respectively equal to $(0.755, 0.327)$ with $\theta = 55.8°$ for the Dubins vehicle and $(0.958, 0.115)$ with $\theta = 26.25°$, so basically the motion corresponding to Figure 4a is flattened to the x-axis, and the angle is reduced nearly to half of the Dubins path. This shifting is more evident considering Figure 5, where the displacement of the final point at maximum curvature is plotted ($+$ symbol) together with the center of the left Dubins circle (o symbol) for several values of parameter d.

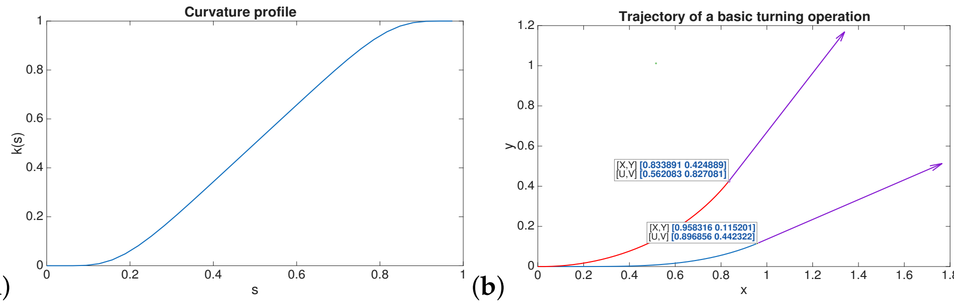

Figure 4. Behavior under a basic curvature profile connecting $k = 0$ with $k = 1$ through (9). (**a**) Graphic representation of the curvature profile (9). (**b**) Final configurations for a Dubins vehicle following a rectangular-shaped curvature (red line) and a vehicle moving as in (9) (blue line); purple vectors represent the final configuration (X,Y representing the position and U,V the orientation) after the turning operation in each case.

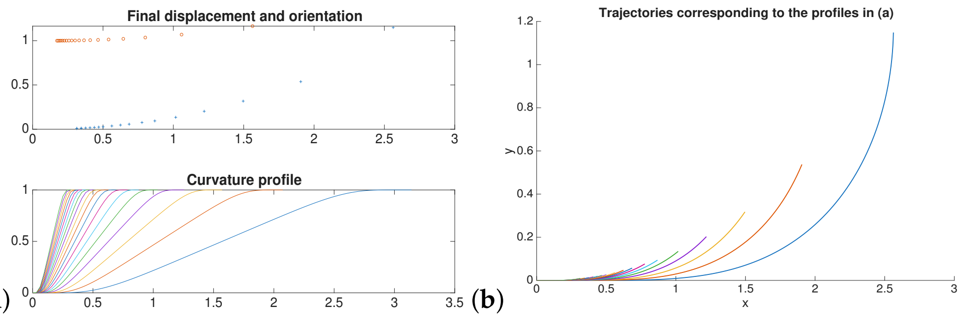

Figure 5. Behavior under a basic curvature profile $\kappa_{0-1}(s)$ connecting $k = 0$ with $k = 1$ through (9). (**a**) Curvature profile (9) for several values of parameter d (below). Final displacement and center of the left circle (above). (**b**) Trajectories corresponding to the curvature profiles in (**a**).

Figure 6 is related to the geometry of a left turn from the maximum to zero, thus following either the curvature profile $\kappa_{1-0}(s)$ depicted at the top of Figure 6a that we use for large turns, or $\kappa_{a-0}(s)$ as in Equation (11), which is adopted when dealing with small turns. Figure 6b shows the plot of a family of positions and orientations achieved for the curvature profiles Figure 6a, where it is assumed that the vehicle starts with maximum steering and which are divided between large turns, where the curvature is kept to the maximum value for a variable amount of time, and short turns, that are characterized by Equation (11) for several values of a. Red vectors denote positions and orientations relative to small turns obtained following profiles Equation (11).

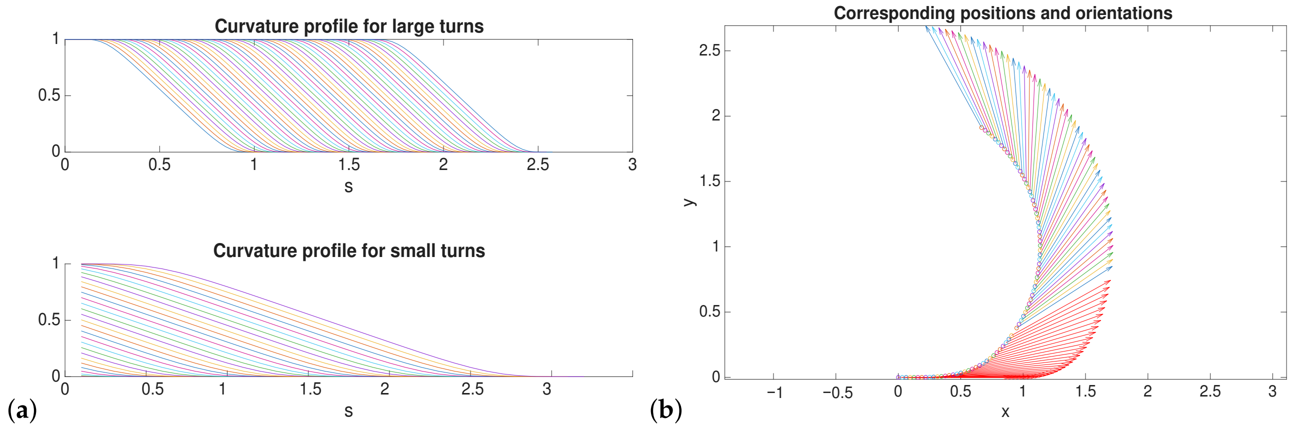

Figure 6. Behavior under a basic curvature profile $\kappa_{1-0}(s)$ connecting $k = 1$ with $k = 0$ through (9). (**a**) Graphic representation of the curvature profile (9). (**b**) Final configurations for a vehicle following the curvature profile (9). Red vectors correspond to curvature profiles for small turns.

Figure 7 shows positions and orientations of a vehicle following the Figure 4a profile and keeping the final curvature, namely a 'mirrored' curvature with respect to the profiles in Figure 6a.

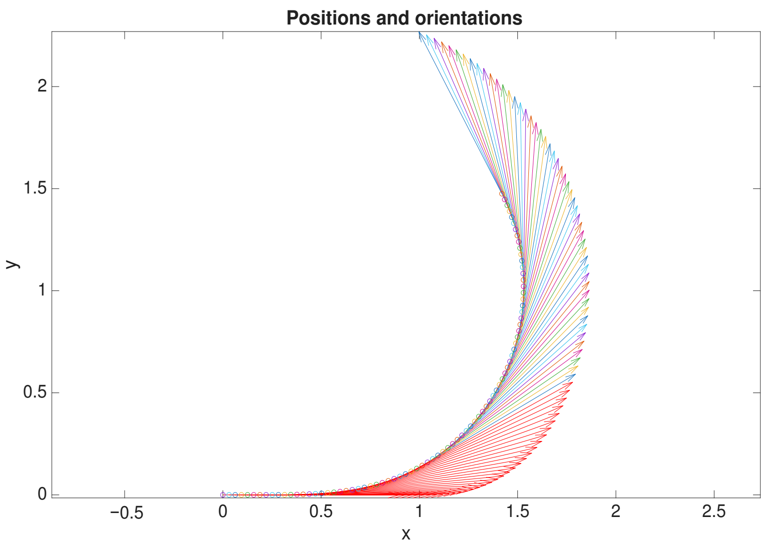

Figure 7. Positions and orientations achieved adopting the curvature profile $\kappa_{0-1}(s)$ (large turns) and $\kappa_{0-a}(s)$ (small turns). Resuls relative to small turns are colored in red.

In Section 5, it turns out that the two figures Figures 6b and 7 are a powerful graphical description of the geometry of a basic left turn respectively reaching or starting from the straight line, and we make use of them in Section 5 for the construction of reference paths.

Finally, we consider a full regular turning operation starting and returning to the zero curvature. Two types of turning profiles are possible depending on the total amount of steering angle, namely reaching and keeping the maximum turning radius, or not. Two examples of such profiles are depicted in Figure 8, where the plots on the top refer to a steering manouver for a large turning angle. In this case, the maximum curvature is reached and kept for a suitable time. In the figures below, an example of a small turn is depicted, the curvature profile in this case does not reach the maximum value.

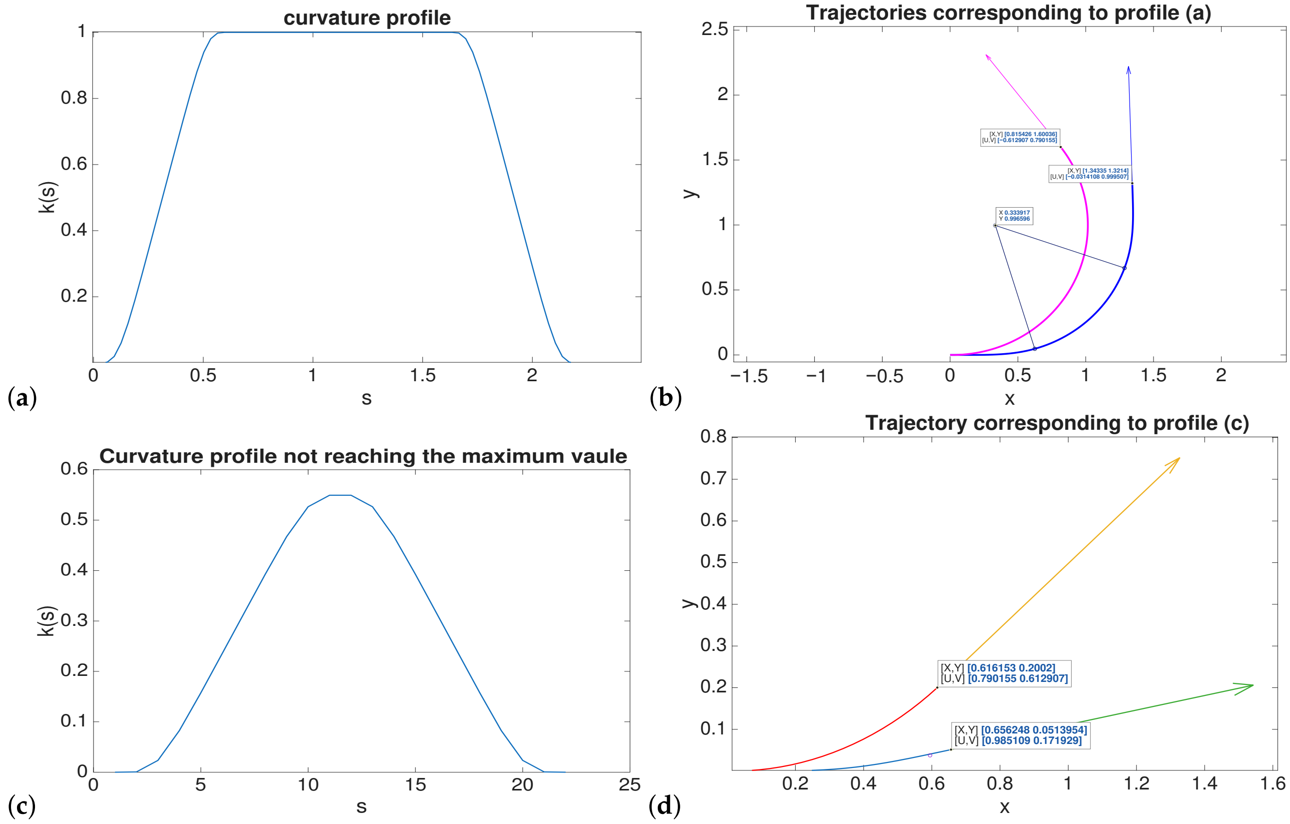

Figure 8. Behavior under curvature profiles through (9) reaching and keeping the maximum turning radius, or not. (**a**) Graphic representation of the curvature profile (9) reaching and keeping the turning radius to 1. (**b**) Final configurations for a Dubins vehicle (purple line) and a vehicle moving as in (9) (blue line); thinner vectors represent the final configuration (X,Y representing the position and U,V the orientation) after the turning operation in each case. (**c**) Graphic representation of the curvature profile (9) not reaching the turning radius. (**d**) Final configurations for a Dubins vehicle (red line) and a vehicle moving as in (9) (blue line). Arrows represent the final configuration (X,Y representing the position and U,V the orientation) after the turning operation in each case.

As a final remark, analogously to Figures 6 and 7, we graphically describe the geometry of these full regular turning operations through the computation of positions and orientations achieved for a family of curvature profiles, both for small and large turns. Those corresponding to small turns, not reaching the maximum curvature value, are depicted in red. Aside from the importance of these graphs themselves, it comes out that they constitute a fundamental tool for the investigation of Section 5.

5. Path Planning Design

Starting from the analysis developed in the previous Section, we now derive an algorithm to build the reference path profile for each of the two basic problems described in the Problem Statement.

We start from Problem 1. Here we describe how to graphically sketch the solution from the geometry of the turn described in the previous section exploiting the turning profile depicted in Figure 9b for left turns and Figure 9c for right turns, and then we describe the instructions to get a numerical solution through an appropriate algorithm.

Consider Figure 10a, where two points in the plane, $\mathbf{P}_i$ and $\mathbf{P}_f$, together with their orientations, are fixed. Assume that a preliminary analysis on the Dubins path type has been performed, with solution LSR, denoted in the figure with the red line. Under the assumption that the curvature and its derivative are zero at initial point, and considering that the first part is a left turn, positions and orientations achieved adopting the curvature profile $\kappa_{0-1}(s)$ are given by the vector sum of the initial pose and any of the vectors describing the motion relative to the curvature profile $\kappa_{0-1}(s)$ depicted in Figure 7. For this reason, we make a roto-translation of all the vectors in Figure 7 centered at $\mathbf{P}_i$, so that we graphically describe the motion starting from P_i with variable curvature profile (Figure 10b). The first part is a full left turn, positions and orientations achieved adopting the curvature profile in Figure 9a are given by the vector sum of the initial pose and any of the vectors describing the motion relative to the curvature profile depicted in Figure 9b. For this reason, we make a roto-translation of all the vectors in Figure 9b centered at $\mathbf{P}_i$, so that we graphically describe the motion starting from P_i with variable curvature profile (Figure 10b).

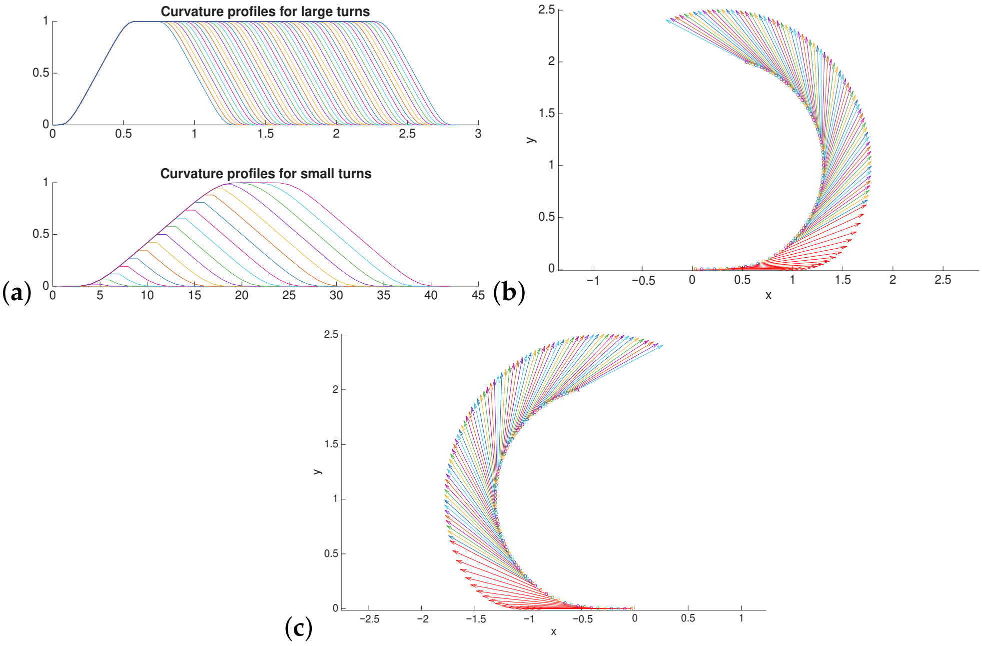

Figure 9. Behavior under a full curvature profile through (9), from $k = 0$ to $k = 0$. (**a**) Graphic representation of the curvature profile for large turns (**above**) and small turns (**below**). (**b**) Final configurations (positions and orientations) after the turning operation in each case. Resuls relative to small turns are colored in red. (**c**) Final configurations (positions and orientations) corresponding to right turn.

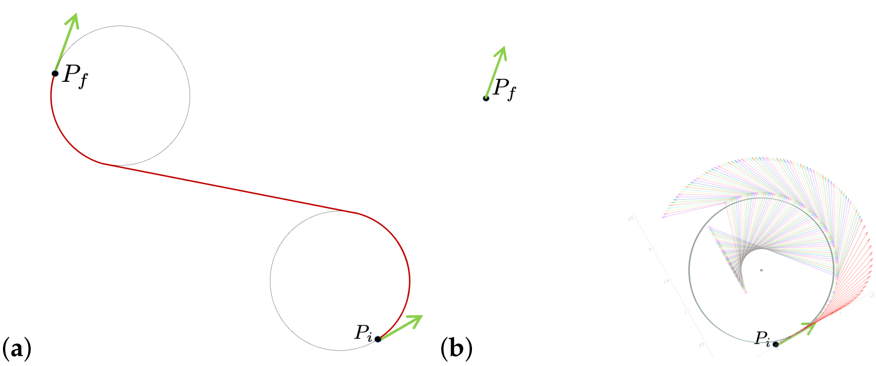

Figure 10. Graphical solution of Problem 1. (**a**) Initial and final configurations (green arrows), and the Dubins path type (LSR) to be approximated (in red). (**b**) Graphical description of the positions and orientations at the end of the first turn through Figure 9b.

As a further step, consider that the Dubins path connecting $\mathbf{P}_i$ and $\mathbf{P}_f$ with orientations φ_i and φ_f coincides with the Dubins path connecting $\mathbf{P}_f$ and $\mathbf{P}_i$ with orientations $-\varphi_f$ and $-\varphi_i$. Indeed, the same path can be travelled from $\mathbf{P}_i$ to $\mathbf{P}_f$ or backwards from $\mathbf{P}_f$ to $\mathbf{P}_i$, with opposite orientation. For this reason, in Figure 11a we make also a roto-translation of all the vectors in Figure 7 centered at $\mathbf{P}_f$ with opposite orientation (denoted as $\bar{\mathbf{P}}_f$ in Figure 11b).

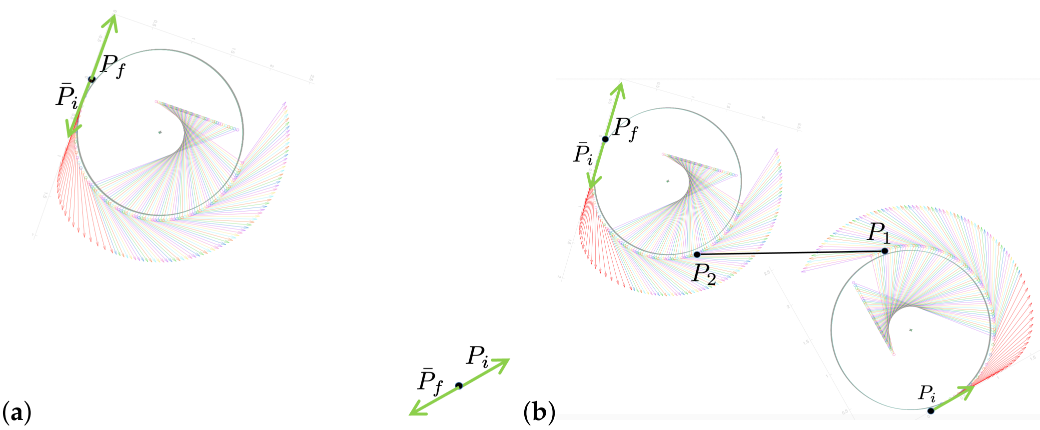

Figure 11. Graphical solution of Problem 1. (**a**) Reversed orientations referred to a backwards path (**b**) Graphical representation of condition (15) to compute the linear segment.

To get the solution, we focus on the central straight 'S' part of the Dubins path. In its smooth approximation, the central line must connect the point where the first turning operation ends (that we denote as P_1 in Figure 11b) to the point where the second turning operation starts. From a graphical perspective, the solution is obtained as follows. Take one point $\bar{P}_1$ on $\mathcal{C}_{P_i}$, another one $\bar{P}_2$ on $\mathcal{C}_{\bar{P}_f}$, select those satisfying:

$$\varphi(P_1) = -\varphi(P_2) = \theta_{12}, \tag{15}$$

where θ_{12} is the orientation of the line connecting P_1 and P_2, $\varphi(P_1)$ the orientation in P_1.

As a final parameter to get the final solution, we can deduce the angle that it is necessary to travel when the maximum curvature is reached (or which profile for small turn must be selected if the maximum curvature is not reached). Consider first the case that the graphical solution provides P_1 and P_2 which are not the red vectors, and denote with P_a the last red vector (which corresponds to the curvature touching but not keeping the maximum curvature, which separates the small turn profiles from large turns profiles). It is easy to see that the angular distance between P_1 and P_a is equal to how much the curvature must be kept equal to the maximum in the first turn, and analogously the angular distance between P_2 and P_b with respect to the second turn.

On the other hand, if the solution above allows to select either P_1 or P_2, or both or them, in the red part, the distance of P_1 from P_a (or P_2 from P_b) allows to select the appropriate curvature profile for small turns which solves the problem.

The final solution is depicted in Figure 12b.

We now describe the steps to follow to pursue the problem solution through a numerical approach. Figure 13 displays the solution sought through a numerical routine that we use in the following to describe the steps of the algorithm in general. The initial and final poses are equal to $P_i = (10, 0)$ and $P_f = (0, 10)$, with orientation respectively $\phi_i = 30°$ and $\phi_f = 60°$.

A preliminary step is the construction of a matrix $\mathcal{C}_l \in \mathbb{R}^{3\times(N+M)}$ encoding the directions and orientations for variable curvature values, as graphically depicted in Figure 9b. We assume that N is a uniform subdivision of a total angle α where the solution belongs to (usually between $\pi/2$ and π), so that column $N + i$ of $\mathcal{C}_l$ encodes position and angle for a steering of an angle equal to $i \cdot \frac{\alpha}{M}$ computed through Equations (A1)–(A3). We assume the same for small turns, where the positions and orientations are computed for a set of N curvature profiles having maximum value equal to $\frac{k}{N} \cdot i$.

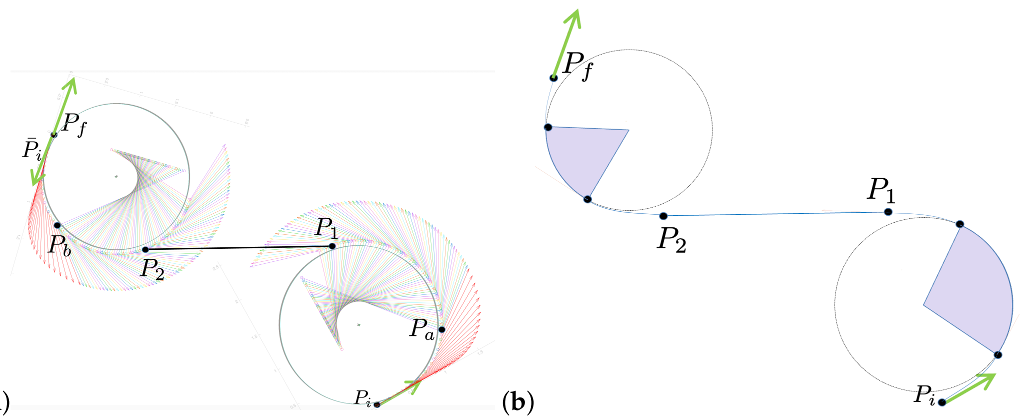

Figure 12. Graphical solution of Problem 1. (**a**) Construction of the relavant points P_a, P_b, P_1, P_2. (**b**) Final solution to Problem 1.

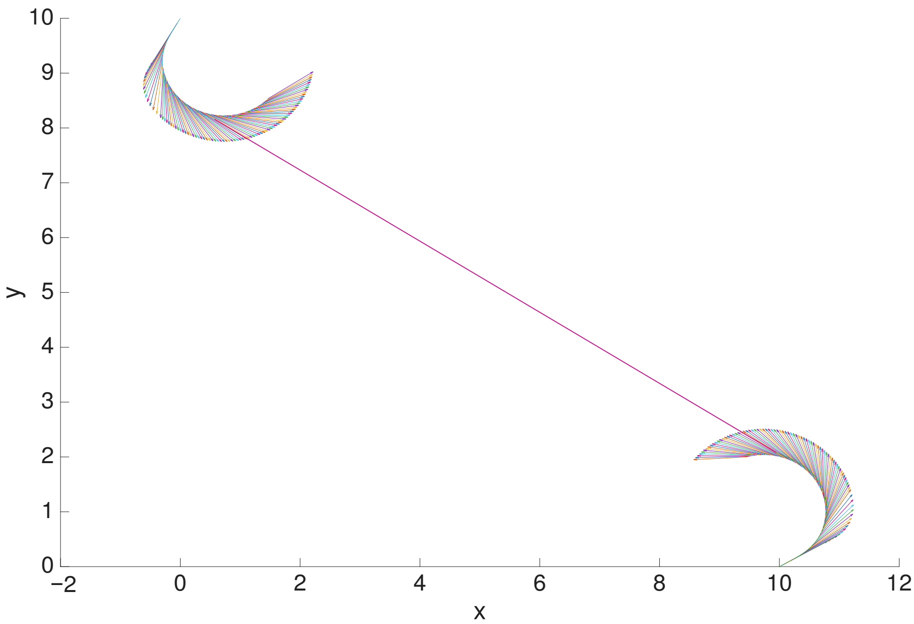

Figure 13. Numerical solution of Problem 1. $P_i = (10, 0)$, $P_f = (0, 10)$, $\phi_i = 30°$, $\phi_f = 60°$.

As a first step, we make a roto-translation of positions and orientations stored in $\mathcal{C}_l$ at $P_i = (10, 0)$, in order to center them at P_i with orientation $\phi_i = 30°$ as follows. Denoting by $\mathbf{x}_{\mathcal{C}_l}$ the first two rows of $\mathcal{C}_l$ and by $\boldsymbol{\phi}_{\mathcal{C}_l}$ its third row, representing respectively positions and orientations of Figure 9b, then $\mathbf{x}_{\mathcal{C}_i} = \mathcal{R}_l \mathbf{x}_{\mathcal{C}_l} + \mathbf{x}_{P_i}$, where $\mathbf{x}_{\mathcal{C}_i}$ are the values of positions and orientations after the roto-translation, and $\mathcal{R}_l$ is the rotation matrix defined as $\mathcal{R}_l = \begin{bmatrix} \cos(\phi_i) & -\sin(\phi_i) \\ \sin(\phi_i) & \cos(\phi_i) \end{bmatrix}$. Next, we do the same at the reversed final position, namely at P_f with orientation $\phi_i = -60°$.

Then we take one point $\bar{P}_1$ on $\mathcal{C}_{P_i}$, another one $\bar{P}_2$ on $\mathcal{C}_{\bar{P}_f}$ and we impose (15); in the example depicted in Figure 13 we found $i = 65$ for the point on $\mathcal{C}_{P_i}$, and $j = 48$ for the point on $\mathcal{C}_{P_f}$, and considering that the step was set equal to $\pi/100$, it results that the first turning angle is equal to 0.65 π and the last is 0.48 π.

Consider now Problem 2. A basic problem in this framework is to select the smooth curve approximating the Dubins path connecting P_i or P_f considering fixed orientations and free curvature and curvature derivative. The solution to this problem can be computed both graphically and numerically following the above steps that lead to the solution of Problem 1, with the only difference of using positions and orientations depicted in Figure 5b when dealing with curvature $\kappa_{1-0}(s)$ at the initial position or Figure 6 for $\kappa_{0-1}(s)$ used at the final position. Figure 14a shows the construction of the graphical solution. Denote by t the angular distance between P_a and P_1, and by q the one between P_b and P_2.

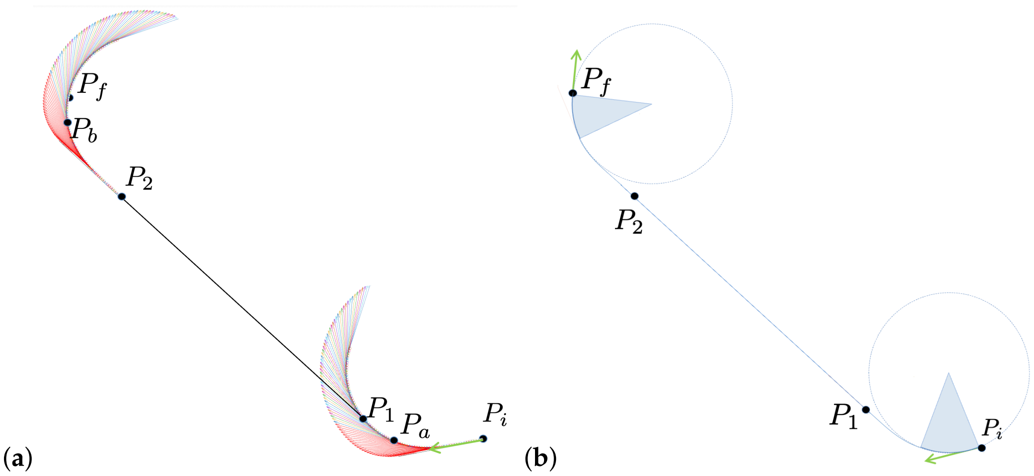

Figure 14. Graphical solution of the basic Problem 2. (**a**) Construction of the relevant points P_a, P_1, P_2, P_b. (**b**) Final solution.

The most interesting fact is that the above solution is useful to solve the general Problem 2.

Before describing the construction of the solution in the general case of n points P_i, we state one main optimality principle which is widely adopted when dealing with the n points Dubins problem, that we extend also in the case of smooth paths.

The following Statement is a corollary proved in [33]. The same principle has been assumed to hold for the general n-points Dubins problem in [34].

Property 1. *The length-optimal path has the following property. Optimal paths never have a change of curvature at the crossing of a via point. The final turning angle of the first subpath is equal to the first turning angle of the second subpath (i.e.,* $q_1 = t_2$ *in Figure 15).*

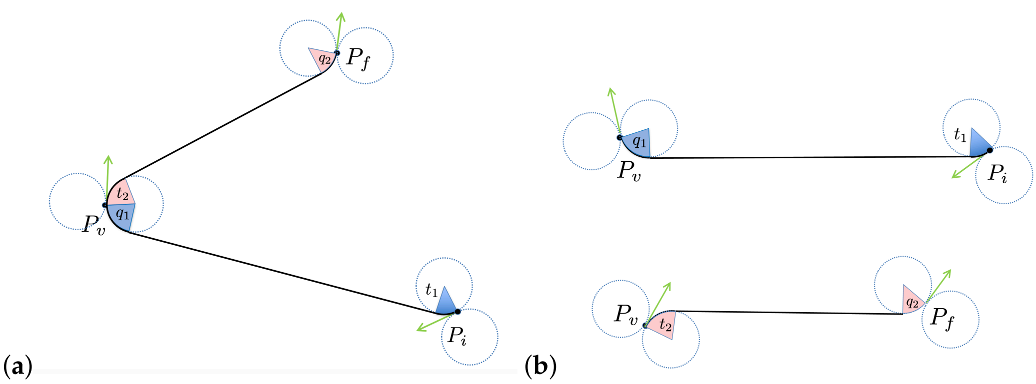

Figure 15. Graphical solution of the 3-point Dubins problem (3PDP). (**a**) Sketch of the solution. (**b**) Decomposition of 3PDP into two basic Problem 2.

Consider the problem of interconnecting three points, the middle being with free orientation (the so called 3PDP problem). Figure 15a shows the optimal solution with discontinuous curvature as deduced in [35], which is characterized by the fact that the angle $q_1 = t_2$. Considering that the crossing of a via point P_v requires a full turning operation, from a straight line to a different straight line, it is possible to divide the problem into two, one focused on the smooth path between P_i and P_v, the other between P_v and P_f, with the additional constraint that $q_1 = t_2$, as depicted in Figure 15b. Moreover, since the most reasonable turning at the via point requires that the crossing of P_v is made at the maximum curvature, each of the two splitted sub-problems have the structure of a basic Problem 2 (see Figure 16a). To get the final solution, the two sub-problems are then merged together at the via point (see Figure 16b) to finally get the final solution path as depicted in Figure 17. Moreover, this same strategy is adopted for the general n-points Dubins Problem.

In general, the n-points Dubins Problem is solved by recurring to any algorithm providing the solution for the ordinary Dubins vehicle (as the one in [34]), and then constructing the smooth approximation expoiting the strategy described above, and thus converting the original problem into a $(n-1)$ basic Problem 2.

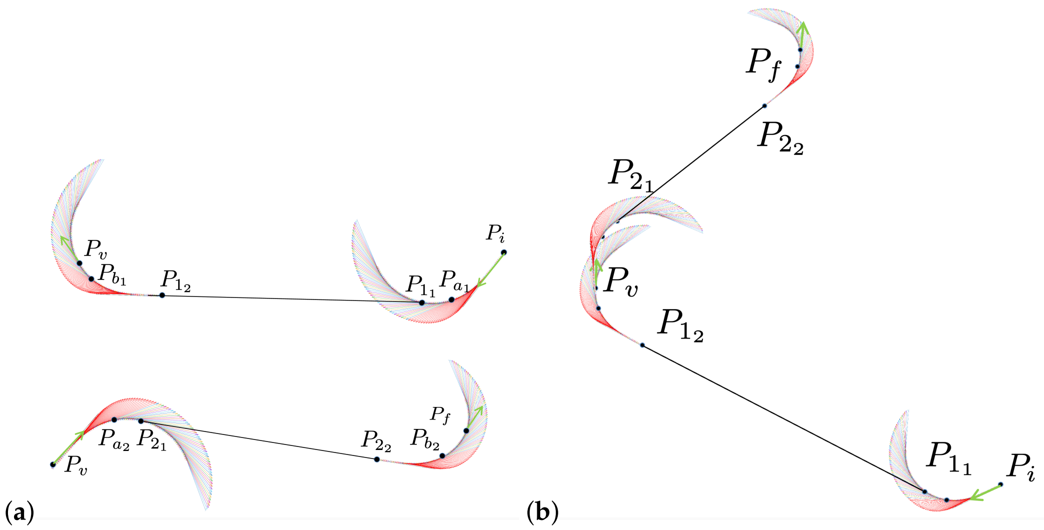

Figure 16. Graphical solution of the 3-point Dubins problem (3PDP). (**a**) Graphical solution of each subproblem. (**b**) Composition of the two partial solutions to get the approximated 3PDP path.

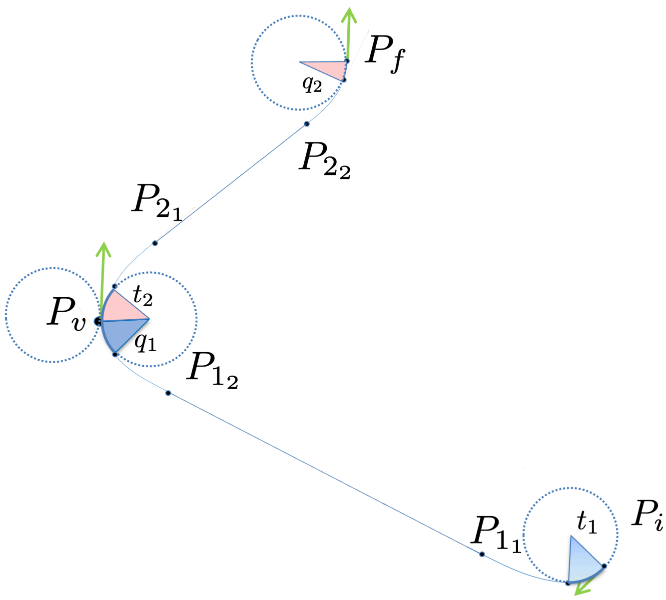

Figure 17. Final solution of the 3PDP as depicted in Figure 15.

6. Conclusions

In this paper, the use of a tunable, smooth curvature function for path planning of Dubins vehicle is thoroughly investigated. This study is motivated by the importance of a path-planning algorithm to produce curvature profiles which can be effectively tracked by the vehicle through a suitable control algorithm. After a wide discussion of the features and drawbacks of the approaches already available in the scientific literature, the problem setting is introduced, and two basic problems are stated in order to solve a wide variety of cases of interest. Next, a detailed analysis is performed regarding a smooth tunable compact function that is further exploited to generate a smooth transition between constant curvature values. The resulting methods are graphical and thus effective for real-time applications, and the structure of the algorithm is finally described to generate the reference path profile by machine computation.

Funding: This research received no external funding.

Data Availability Statement: The datasets generated for this study are available on request to the corresponding author.

Conflicts of Interest: The author declares no conflicts of interest.

Appendix A. Plane Curves: Definitions and Properties

Here we briefly report some definitions and properties of plane curves that we adopted in the paper. A *plane curve* is a locus of points $(x, y) \in \mathbb{R}^2$ satisfying a relation among them, which can be *explicit* if it is defined by an equation as $y = f(x)$, *implicit* if it is of the form $f(x, y) = 0$, or *parametric* if $x = x(t)$, $y = y(t)$ for a *parameter* $t \in \mathcal{I} \subseteq \mathbb{R}$. In this paper, we deal with parametric curves.

Parametric Curve Description

A (parametric) plane curve is a function $\mathbf{p} : [a, b] \to \mathbb{R}^2$. A parametric curve is said to be *regular* if the first derivative vector does not vanish in any point of the domain. It is well known from differential geometry that regularity is, in general, essential for the smoothness of the resulting curve. We therefore restrict the discussion to parametrizations that are regular.

A regular parametric curve $\mathbf{p} : [a, b] \to \mathbb{R}^2$ with $\mathbf{p}(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix}$ is *differentiable* if both $\frac{d}{dt}x(t)$ and $\frac{d}{dt}y(t)$ exist, and *differentiable k times* if both $\frac{d^k}{dt^k}x(t)$ and $\frac{d^k}{dt^k}y(t)$ exist. In the following, for the ease of readability, we denote $\frac{d^k}{dt^k}[\mathbf{p}(t)]$ with $\mathbf{p}^{(k)}(t)$ (whenever it exists).

In geometry, a curve parametric representation is called a *natural equation* if it specifies a curve independently of any choice of coordinates or parameterization. Basically, it describes the curve's shape, not its placement in the plane. The Cesaro equation of a plane curve is a natural equation relating the curvature κ at one point of the curve to the arc length s from the start of the curve to the given point. It may also be equivalently given as an equation relating the radius of curvature R to arc length. Indeed, these representations are equivalent since $R = \frac{1}{\kappa}$.

Euler gave an integral solution for plane curves, that we use in this paper as follows. Let ϕ be the angle between the tangent line to the curve and the x-axis the tangent angle, then:

$$\phi(s) = \int_0^s \kappa(\varsigma) d\varsigma, \tag{A1}$$

where $\kappa(s)$ is the curvature of the curve with respect to the arc length s. Then the equation $\kappa = \kappa(s)$ have the solution expressed by the curve with parametric equations:

$$x(s_1) = \int_0^{s_1} cos(\phi(s))ds, \tag{A2}$$

$$y(s_1) = \int_0^{s_1} sin(\phi(s))ds. \tag{A3}$$

We make use of this expressione to compute the path corresponding to an assigned curvature profile expressed through a Cesaro equation.

Appendix B. Bump Functions: Definitions and Properties

Roughly, bump functions are smooth functions with compact support (zero outside a region). Since its early days in mathematical physics, the use of bump functions has been widely adopted in several diverse areas of science and technology in many modern applications,. Indeed, their use is natural and successful for creating smooth approximations, partitions of unity (patching things together), mollifiers (smoothing data/functions), and defining test functions in analysis (distribution theory), appearing in differential geometry, PDEs, aerodynamics (Hicks-Henne), and as reliability models. They allow smooth handling of abrupt changes, enabling complex global structures from local ones.

Definition A1 ([36])**.** *Let f be a function $f : \mathbb{R} \to \mathbb{R}$. we define f a bump function if the two conditions:*

1. $f \in \mathcal{C}^\infty$,
2. *supp(f) is compact, where supp(f) is the support of function f, namely the set supp(f) $= \{x \in \mathbb{R} : f(x) \neq 0\}$.*

Through the use of bump functions it is possible to build smooth approximations of the step function, as in hte case of the functions adopted in this paper for building the transition to turn and back to the straight line.

The next result shows the analytical expression of the first and second derivative, whose graph in plotted in Figure 3. The importance of this result is the opportunity to bound the maximal value through a constraint on the design parameters.

Proposition A1. *Given the function $f : \mathcal{I} \to \mathbb{R}$ defined as $f(s) = a\tanh(c\tan(ds + f) + b)$, then its derivative is equal to:*

$$\frac{\mathrm{d}}{\mathrm{d}s} f(s) = \frac{acd}{\cos^2(ds+f)\cosh^2(c\tan(ds+f)+1)}, \tag{A4}$$

and second derivative equal to:

$$-\frac{2\,a\,c^2\,d^2\,\sinh(c\,\tan(d\,s+f)+b) - 2\,a\,c\,d^2\,\cos(d\,s+f)\,\sin(d\,s+f)\,\cosh(c\,\tan(d\,s+f)+b)}{\cos^4(d\,s+f)\,\cosh^3(c\,\tan(d\,s+f)+b)}. \tag{A5}$$

Proof. Here we give a sketch of the proof, which is conducted by direct inspection. Recall (1) the chain rule for the derivation of a composite function, namely $f_x(g(x)) = f_y(y)]_{y=g(x)} g_x(x)$, (2) the common derivatives $\frac{\mathrm{d}}{\mathrm{d}x}\tanh(x) = \frac{1}{\cosh^2(x)}$ and $\frac{\mathrm{d}}{\mathrm{d}x}\tan(x) = \frac{1}{\cos^2(x)}$. By the linearity of derivation one has:

$$a\frac{1}{[\cosh^2 y]_{y=c\tan(dx+f)}} \cdot \frac{\mathrm{d}}{\mathrm{d}x}(c\tan(dx+f)) = \frac{ac}{\cosh^2(c\tan(dx+f)) \cdot [\cos^2 y]_{y=dx+f}\frac{\mathrm{d}}{\mathrm{d}x}(dx+f)}$$

and hence (A4) follows. As for (A5), by applying the derivative of reciprocal function $\left(\frac{1}{v}\right)' = -\frac{(v)'}{v^2}$ and iteratively the above chain rule $f_x(g(x)) = f_y(y)]_{y=g(x)} g_x(x)$, it is matter of simple computation to get (A5). □

Unfortunately, it is not possible to compute the generic k-th derivative in a unique analytical formulation, but they can be computed for each given k.

References

1. LaValle, S.M. *Planning Algorithms*; Cambridge University Press: New York, NY, USA, 2006.
2. Del-Rio-Rivera, F.; Ramírez-Rivera, V.M.; Donaire, A.; Ferguson, J. Robust trajectory tracking control for fully actuated marine surface vehicle. *IEEE Access* **2020**, *8*, 223897–223904. [CrossRef]
3. Zuo, Z.; Liu, C.; Han, Q.L.; Song, J. Unmanned aerial vehicles: Control methods and future challenges. *IEEE/CAA J. Autom. Sin.* **2022**, *9*, 601–614. [CrossRef]
4. Hota, S.; Ghose, D. Optimal trajectory planning for unmanned aerial vehicles in three-dimensional space. *J. Aircr.* **2014**, *51*, 681–688. [CrossRef]
5. Hansen, K.D.; La Cour-Harbo, A. Waypoint planning with Dubins curves using genetic algorithms. In *Proceedings of the 2016 European Control Conference (ECC)*; IEEE: New York, NY, USA, 2016; pp. 2240–2246.
6. Cai, W.; Zhang, M.; Zheng, Y.R. Task assignment and path planning for multiple autonomous underwater vehicles using 3D dubins curves. *Sensors* **2017**, *17*, 1607. [CrossRef] [PubMed]
7. Boissonnat, J.D.; Cérézo, A.; Degtiariova-Kostova, E.V.; Kostov, V.P.; Leblond, J. Shortest plane paths with bounded derivative of the curvature. *Comptes Rendus L'Académie Des. Sci. Ser. I-Math.* **1999**, *329*, 613–618. [CrossRef]
8. Fraichard, T.; Scheuer, A. From Reeds and Shepp's to continuous-curvature paths. *IEEE Trans. Robot.* **2004**, *20*, 1025–1035. [CrossRef]
9. Zhang, S.; Wang, R.; Jian, Z.; Zhan, W.; Zheng, N.; Tomizuka, M. Controllable clothoid path generation for autonomous vehicles. *IEEE Robot. Autom. Lett.* **2023**, *8*, 5108–5115. [CrossRef]
10. Šelek, A.; Seder, M. Smooth motion planning method for mobile robots in dynamic environments. *Robot. Auton. Syst.* **2025**, *195*, 105220. [CrossRef]
11. Khanfir, S.; Hasegawa, K.; Nagarajan, V.; Shouji, K.; Lee, S.K. Manoeuvring characteristics of twin-rudder systems: Rudder-hull interaction effect on the manoeuvrability of twin-rudder ships. *J. Mar. Sci. Technol.* **2011**, *16*, 472–490. [CrossRef]
12. Khanfir, S.; Nagarajan, V.; Hasegawa, K.; Lee, S.K. Estimation of mathematical model and its coefficients of ship manoeuvrability for a twin-propeller twin-rudder ship. In Proceedings of the International Conference on Marine Simulation and Ship Maneuverability (MARSIM 2009), Panama City, Panama, 17–20 August 2009; Volume 9, pp. M159–M166.
13. Gawron, T.; Michałek, M.M. A $\mathbb{G}^3$-continuous extend procedure for path planning of mobile robots with limited motion curvature and state constraints. *Appl. Sci.* **2018**, *8*, 2127. [CrossRef]
14. Hassani, V.; Lande, S.V. Path planning for marine vehicles using Bézier curves. *IFAC-PapersOnLine* **2018**, *51*, 305–310. [CrossRef]
15. Yang, F.; Liu, J.; Li, S.; Hu, X. Minimum jerk motion planning for maritime autonomous surface ships based on Bézier curves. *J. Mar. Eng. Technol.* **2025**, *24*, 85–97.
16. De Giorgi, C.; De Palma, D.; Parlangeli, G. Online odometry calibration for differential drive mobile robots in low traction conditions with slippage. *Robotics* **2023**, *13*, 7. [CrossRef]
17. Botros, A.; Smith, S.L. Tunable trajectory planner using g 3 curves. *IEEE Trans. Intell. Veh.* **2022**, *7*, 273–285. [CrossRef]
18. Banzhaf, H.; Berinpanathan, N.; Nienhüser, D.; Zöllner, J.M. From G^2 to G^3 continuity: Continuous curvature rate steering functions for sampling-based nonholonomic motion planning. In *Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV)*; IEEE: New York, NY, USA, 2018; pp. 326–333.
19. Mondek, M.; Hromčík, M. Linear analysis of lateral vehicle dynamics. In *Proceedings of the 2017 21st International Conference on Process Control (PC)*; IEEE: New York, NY, USA, 2017; pp. 240–246.
20. Ravankar, A.; Ravankar, A.A.; Kobayashi, Y.; Hoshino, Y.; Peng, C.C. Path smoothing techniques in robot navigation: State-of-the-art, current and future challenges. *Sensors* **2018**, *18*, 3170. [CrossRef]
21. Bevilacqua, P.; Frego, M.; Bertolazzi, E.; Fontanelli, D.; Palopoli, L.; Biral, F. Path planning maximising human comfort for assistive robots. In *Proceedings of the 2016 IEEE Conference on Control Applications (CCA)*; IEEE: New York, NY, USA, 2016; pp. 1421–1427.
22. Farouki, R.T. The Bernstein polynomial basis: A centennial retrospective. *Comput. Aided Geom. Des.* **2012**, *29*, 379–419. [CrossRef]
23. Montés, N.; Herraez, A.; Armesto, L.; Tornero, J. Real-time clothoid approximation by Rational Bezier curves. In *Proceedings of the 2008 IEEE International Conference on Robotics and Automation*; IEEE: New York, NY, USA, 2008; pp. 2246–2251.
24. Kawabata, K.; Ma, L.; Xue, J.; Zhu, C.; Zheng, N. A path generation for automated vehicle based on Bezier curve and via-points. *Robot. Auton. Syst.* **2015**, *74*, 243–252. [CrossRef]
25. Dubins, L.E. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *Am. J. Math.* **1957**, *79*, 497–516. [CrossRef]
26. Kiss, D. A Continuous Curvature Rate Path Planner for Autonomous Cars in Narrow Environments. In *Proceedings of the 2025 29th International Conference on Methods and Models in Automation and Robotics (MMAR)*; IEEE: New York, NY, USA, 2025; pp. 408–413.

27. Zheng, A.; Narayanan, S.S.; Vaidya, U. Safe navigation using density functions. *IEEE Robot. Autom. Lett.* **2023**, *8*, 8502–8509. [CrossRef]
28. Barsky, B.A.; DeRose, T.D. Geometric continuity of parametric curves: Three equivalent characterizations. *IEEE Comput. Graph. Appl.* **2002**, *9*, 60–69.
29. Coetzee, M.; Baart, M. Review of geometric continuity, with application to the construction of G2 continuous curves. *Quaest. Math.* **1992**, *15*, 279–298. [CrossRef]
30. Gregory, J.A. Geometric continuity. In *Mathematical Methods in Computer Aided Geometric Design*; Elsevier: Amsterdam, The Netherlands, 1989; pp. 353–371.
31. Parlangeli, G.; Indiveri, G. Dubins inspired 2D smooth paths with bounded curvature and curvature derivative. *IFAC Proc. Vol.* **2010**, *43*, 252–257. [CrossRef]
32. Shkel, A.M.; Lumelsky, V. Classification of the Dubins set. *Robot. Auton. Syst.* **2001**, *34*, 179–202. [CrossRef]
33. Parlangeli, G. Shortest paths for Dubins vehicles in presence of via points. *IFAC-PapersOnLine* **2019**, *52*, 295–300. [CrossRef]
34. Goaoc, X.; Kim, H.S.; Lazard, S. Bounded-curvature shortest paths through a sequence of points using convex optimization. *SIAM J. Comput. Soc. Ind. Appl. Math.* **2013**, *42*, 662–684. [CrossRef]
35. Parlangeli, G.; De Palma, D.; Attanasi, R. A novel approach for 3PDP and real-time via point path planning of Dubins' vehicles in marine applications. *Control Eng. Pract.* **2024**, *144*, 105814. [CrossRef]
36. Nestruev, J. *Smooth Manifolds and Observables*; Springer: Berlin/Heidelberg, Germany, 2003.