Real-time Approximation of Clothoids with Bounded Error for Path Planning Applications

Mišel Brezak and Ivan Petrović, *Member, IEEE*

***Abstract*—We present a method for real-time computation of clothoid coordinates that guarantees bounded approximation error over a wide range of clothoid parameters provided that clothoid's orientation change and length are bounded. It is shown that coordinates of clothoid with any parameters can be computed from those of a single clothoid (with fixed parameters), using appropriate geometrical transformations. A comprehensive analysis is given on how to determine a required set of clothoids and, based on this, how to sample a clothoid in a lookup table in order to achieve required approximation precision. The algorithm is computationally very efficient and therefore suitable for real-time path planning, as well as for other applications that benefit from fast clothoid computation.**

***Index Terms*—clothoid approximation, Fresnel integrals, path planning, motion control.**

I. Introduction

Common path-planning methods usually generate obstacle-free path, but with no or very little concern about path feasibility or optimality, so that it is usually necessary to apply some kind of transform algorithm to locally smooth such a path. Various path-smoothing algorithms are proposed in the literature: cubic splines [1], intrinsic splines [2], Bezier's curves [3], quintic Bezier splines [4] and clothoids. The main advantage of clothoids over other smoothing methods in path planning applications is linear change of their curvature, which is of great importance for transportation of people or heavy and sensitive loads since it prevents abrupt changes in the centripetal acceleration and forces experienced by a vehicle increasing driving comfort. Clothoids are very attractive in path smoothing applications as they are easy to follow due to their linearly changing curvature. In this way e.g. Shanmugavel *et al.* use Dubins paths and smooth them with clothoid arcs in order to perform co-operative path planning of multiple unmanned aerial vehicles [5].

Clothoids have also advantages over other smoothing techniques in sense of vehicles optimal motion planning—by applying the Maximum Principle from optimal control theory, for forward motion and differential drive vehicle, one can find that the necessary condition for trajectory to be time optimal yields clothoids [6]. Further, Boissonnat *et al.* [7] studied the shortest plane paths joining two given positions with given tangent angles and curvatures along which the tangent angle and the curvature are continuous and the derivative of the curvature is bounded. They showed that at a point where such a path is of class C^3, it must be locally a piece of a clothoid or a line segment. Similarly, Fraichard and Scheuer [8] showed that with requirements of continuous curvature and bounded both curvature and its derivative, shortest path consist of line segments, circular arcs and clothoids. Meidenbauer [9] found clothoids useful for path-planning, as he proved, both theoretically and experimentally, that clothoid steering model is valid for real Ackermann-steered vehicles. He obtained that the actual paths driven by the vehicle were generally a close match to the originally planned theoretical clothoid path. As roads are usually composed of clothoids, they are a natural choice for path planning of autonomous cars [10].

Manuscript received February 24, 2013. This research was supported by European Community's Seventh Framework Programme under grant agreement No. 285939 (ACROSS) and the Ministry of Science, Education and Sports of the Republic of Croatia (grant No. 036-0363078-3018).

M. Brezak and I. Petrović are with the University of Zagreb, Faculty of Electrical Engineering and Computing, Department of Control and Computer Engineering, Croatia (e-mail: {misel.brezak,ivan.petrovic}@fer.hr).

Unfortunately, clothoids are defined in terms of Fresnel integrals [11], which are transcendental functions that cannot be solved analytically. This makes clothoids difficult to use in real-time applications, so that for real-time motion planning many authors resort to curves with nonlinear curvature, that are easier to compute but whose curvature is difficult to control, such as Bezier curves [12]. However, motivated by many advantageous properties of clothoids for path planning applications, several researchers developed a number of methods that compute clothoid coordinates approximately. Hereafter, we briefly review the most advanced methods and introduce the method proposed in this paper.

A group of methods compute clothoid coordinates iteratively, e.g. using an iterative method that utilizes power series [13]. However, in this way error grows with curve length so that power series are appropriate only for small lengths. Therefore, for large lengths continued complex fractions are used instead, which are numerically involved due to complex numbers calculations. Since clothoid coordinates are computed in terms of Fresnel integrals, many existing numerical integration algorithms [13] can be utilized. Starting from some initial point, such methods evaluate numerical integration algorithm, so that apart from the final point, they also output a series of coordinates between initial and final point. Due to numerical complexity, numerical integration methods are not suitable for real-time single point approximation.

Another group of methods approximate a clothoid at defined interval with other analytical and easy to compute curves. For example Wang *et al.* [14] approximate a clothoid by Bezier curves or B-spline curves, whereas Sanchez-Reyes and Chacon [15] use s-power series. Rational function approximations of clothoids, which are very convenient in computer programs, are given by Heald [16]. Mielenz [17] uses continuous function approximation based on both Taylor expansion and a formula derived by Boersma [11]. Meek and Walton [18] use arc splines for this purpose. However, these methods, while successful in CAD applications, are not suitable for real-time path planning or smoothing applications due to high computational burden of high order approximation curves. E.g. [14] uses a 26^{th} order continuous function, which is unacceptable in most real-time systems. Montes *et al.* [19] use rational Bezier curves that are typically of 11^{th} order to approximate Fresnel integrals, and are among first authors that successfully use clothoids in real-time path planning. Nevertheless, many clothoid-based path-planning algorithms that use iterative search techniques would benefit from even faster computation of clothoid coordinates. Further, none of these methods guarantees bounded error of clothoid approximation over a broad range of clothoid parameters.

In this paper, we propose a new method for real-time approximation of clothoids, which overcomes existing methods in terms of computational simplicity and approximation accuracy. It is based on a property that coordinates of clothoid with any parameters can be computed from those of a single clothoid (with fixed parameters), using appropriate geometrical transformations. To enable real-time approximation of arbitrary clothoids we sample a single clothoid and store sampled points in a lookup table. By applying circular interpolation between points of the lookup table we ensure bounded approximation error for clothoids with any scaling or initial curvature, provided that clothoid's orientation change and length are bounded. The approximation error is of order $O(1/n^3)$, where n is the number of points in the lookup table. The accuracy of approximation can be tuned simply by resizing the lookup table, without affecting computational burden. This is not the case with the most existing algorithms, whose accuracy is increased at the cost of higher execution time. Moreover, our method is simple to implement, and we have conducted a careful error analysis that guarantees bounded approximation error over a required range of parameter values.

The proposed method could easily be integrated in any of the algorithms that plan paths based on clothoids, such as [5], [19], [20], [21] and its efficiency allows to use them in real-time. The computational simplicity and high approximation accuracy make the proposed method very suitable for high-demanding real-time path planning applications. We have already successfully applied it for fast real-time path planning and replanning in highly dynamic environments, where a path that consists of straight line segments is smoothed using clothoids [22]. This application belongs to a class of problems where computational simplicity is especially important, as it involves nonlinear equations with clothoids, which due to lack of analytic form should be solved numerically, requiring high number of clothoid evaluations. Another example where computational simplicity is of particularly high importance is for path planning algorithms that optimize the path by finding many alternative paths and choosing the best one among them [4]. Computational simplicity of the proposed method enables that all alternative paths are smoothed in real-time and that the one which provides the fastest reaching the goal position is chosen as the best one. Further, the proposed algorithm can be used for design of other time critical real-time clothoid applications, e.g. shape completion problem in computer vision [23], where a nonlinear system of equations involving Fresnel Integrals is solved, or tracking of road lanes based on clothoids [10], [24]. Application where algorithm precision is of particular importance include robot soccer, e.g. in simulator leagues [25] where a small initial robot position error may multiply by a huge factor and result in a big error at the end of ball trajectory.

The rest of the paper is organized as follows. Section II introduces the algorithm for clothoids computations and Section III discusses the interpolation algorithms. Sections IV and V describe the procedures for determination of the required set of clothoids and the lookup table parameters, respectively. Section VI presents the algorithms for computing and querying the coordinates of the clothoid. Section VII illustrates the algorithm parameter settings and gives a comprehensive analysis of the algorithm performances on a mobile robot path planning problem. The paper ends with a conclusion.

II. Approximation to a Clothoid

Parametric expressions for coordinates of a general clothoid are

$$x(s) = x_0 + \int_0^s \cos(\theta_0 + \kappa_0\xi + \frac{1}{2}c\xi^2)d\xi, \tag{1a}$$

$$y(s) = y_0 + \int_0^s \sin(\theta_0 + \kappa_0\xi + \frac{1}{2}c\xi^2)d\xi, \tag{1b}$$

where (x_0, y_0) is initial point, θ_0 is initial tangent angle, κ_0 is initial curvature, c is parameter called sharpness and $s \geq 0$ is parameter that denotes arc length. Important properties of a clothoid are:

- Curvature: $\kappa(s) = \kappa_0 + cs$;
- Tangent angle:

$$\theta(s) = \theta_0 + \kappa_0 s + \frac{1}{2}cs^2. \tag{2}$$

Instead of clothoid sharpness, *scaling factor* C is sometimes used, which is proportional with clothoid size. Its relation with clothoid sharpness is $C^2 = 1/c$.

Equations for clothoid coordinates (1) are transcendental functions. Nevertheless, if sufficient memory is available, the fastest and the most appropriate solution for online computations is to store clothoid coordinates in a lookup table.

Instead of storing coordinates of all needed clothoids, as in [26], we have investigated a more practical solution based on appropriate numerical transformations that compute coordinates of a clothoid with any parameters based on a single clothoid stored in the memory. The clothoid whose coordinates are stored in the lookup table will be called a *basic clothoid*, and any other clothoid whose points we want to compute will be called a *general clothoid*. To obtain a general clothoid, a transformation invariance property [27] of the parametric curves is used. Using this property, it is possible to compute points of any general clothoid by transforming points of the basic clothoid, where rescaling, rotation, and translation transformations are performed. Let the basic clothoid be denoted by $\mathcal{L}$. We set all initial conditions of the basic clothoid to zero and its sharpness to some constant greater than zero, i.e. $x_{\mathcal{L}0} = y_{\mathcal{L}0} = 0, \quad \theta_{\mathcal{L}0} = 0, \quad \kappa_{\mathcal{L}0} = 0, \quad c_{\mathcal{L}} > 0$. By using transformation invariance property, we obtain a relation that finds a general clothoid by using the basic clothoid coordinates

$$\begin{bmatrix} x(s) \\ y(s) \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \mathbf{R}\left(\frac{-\kappa_0^2}{2c} + \theta_0\right) \cdot \sqrt{\frac{c_{\mathcal{L}}}{|c|}} \begin{bmatrix} x_{\mathcal{L}}\left(\sqrt{\frac{|c|}{c_{\mathcal{L}}}}\left(s + \frac{\kappa_0}{c}\right)\right) - x_{\mathcal{L}}\left(\sqrt{\frac{|c|}{c_{\mathcal{L}}}}\frac{\kappa_0}{c}\right) \\ \operatorname{sgn}(c)\left[y_{\mathcal{L}}\left(\sqrt{\frac{|c|}{c_{\mathcal{L}}}}\left(s + \frac{\kappa_0}{c}\right)\right) - y_{\mathcal{L}}\left(\sqrt{\frac{|c|}{c_{\mathcal{L}}}}\frac{\kappa_0}{c}\right)\right] \end{bmatrix}, \tag{3}$$

where $(x_{\mathcal{L}}, y_{\mathcal{L}})$ are coordinates of the basic clothoid, and $\mathbf{R}(\theta)$ is a rotation matrix defined as

$$\mathbf{R}(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}.$$

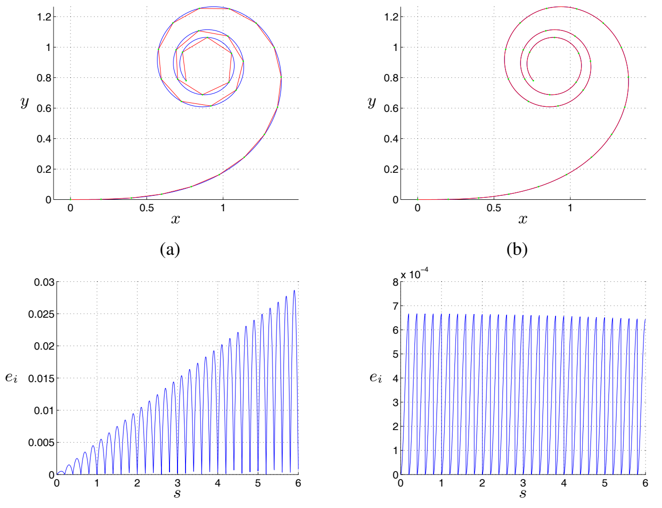
Our plan is to store coordinates of the basic clothoid in a lookup table and reuse them later to accelerate calculations with clothoids. However, it must be considered that the number of points in the table is limited. Therefore a careful analysis must be conducted in order to determine: (i) a required set of clothoids that is sufficient for intended application; and (ii) safe parameters of the lookup-table based on the required set of clothoids, which guarantee bounded error of clothoid coordinates retrieved with (3).

The following lookup-table parameters have to be determined: (i) sharpness $c_{\mathcal{L}}$ of the basic clothoid stored in the lookup table; (ii) length of the basic clothoid $s_{\mathcal{L}}$; and (iii) sampling interval $\Delta s_{\mathcal{L}}$—this is a curve length between the successive points of the basic clothoid.

Note that although relation (3) is exact, the obtained point coordinates will only be approximate. Sources of the error and possible remedies are as follows:

1) **Source:** Coordinates $x_{\mathcal{L}}$ and $y_{\mathcal{L}}$ of the basic clothoid still have to be computed numerically and therefore contain errors.
Remedy: This error can be easily bounded because coordinates can be computed offline with any desired accuracy (at least up to machine precision), and could therefore be neglected.
2) **Source:** As clothoid curve needs to be discretized in order to store it in the lookup table, a sampling error is introduced.
Remedy: To answer a query that falls between two successive points of the lookup table an interpolation is used. Approximation error will then depend on the sampling interval $\Delta s_{\mathcal{L}}$ and the quality of the interpolation.

In the sequel, firstly it is discussed how to best implement the interpolation and the interpolation error is analyzed. Then a possible procedure for determining the required set of clothoids for path-planning application is given. Based on this set, a procedure is presented for safe lookup table parameters determination for required precision.



(a) (b) (c) (d)

Fig. 1. Comparison of straight line and circular interpolation of a clothoid. The lookup table has the parameters $c_{\mathcal{L}} = 1$, $\Delta s_{\mathcal{L}} = 0.2$ and $s_{\mathcal{L}} = 6$. *(a)* Clothoid interpolated by straight-line interpolation. (Red—exact clothoid, blue—interpolated.) *(b)* Clothoid interpolated by circular interpolation. A difference between exact and interpolated clothoid is hardly visible. *(c)* Interpolation error with straight-line interpolation. *(d)* Interpolation error with circular interpolation. Note that circular interpolation error is substantially lower than linear interpolation error.

III. Interpolation

To compute clothoid coordinates that fall between two successive points of the lookup table an interpolation is used. For real-time applications it is of crucial importance that interpolation algorithm is efficient. A simple and efficient option is linear interpolation where straight line is interpolated between two neighbor points of the lookup table. Unfortunately, linear interpolation results with higher interpolation error as length of the clothoid s grows, as it is visible in Fig. 1(c) (we define interpolation error as the Euclidean distance between the exact point and the interpolated point). This is because the curvature grows with s, resulting with larger distance of the clothoid curve from the interpolation line. As clothoid is a spiral that converges to a point, the interpolation error reaches its maximum and then slowly decays to zero at large values of s. Because of high interpolation error the straight-line interpolation is also a poor choice when dealing with clothoids, but it can be used in case when high computation speed is very important.

Interpolation with circle arcs is a better choice, as a clothoid can be viewed as infinite succession of circular arc segments with linearly growing curvature. To perform circular interpolation between j-th and $(j+1)$-th point of the basic clothoid, where $j \geq 0$, we first calculate radius of curvature at the middle of the segment as

$$r_j = \kappa_{mid}^{-1} = (c_{\mathcal{L}}(j+0.5)\Delta s_{\mathcal{L}})^{-1}. \tag{4}$$

Next, using (2), we obtain tangent angle of the basic clothoid in the j-th point as

$$\theta_{\mathcal{L}}^j = \frac{1}{2} c_{\mathcal{L}} (j \Delta s_{\mathcal{L}})^2. \tag{5}$$

Then the center of the curvature in the j-th point is

$$x_c^j = x_{\mathcal{L}}^j + r_j \cos\left(\theta_{\mathcal{L}}^j + \frac{\pi}{2}\right), \quad y_c^j = y_{\mathcal{L}}^j + r_j \sin\left(\theta_{\mathcal{L}}^j + \frac{\pi}{2}\right),$$

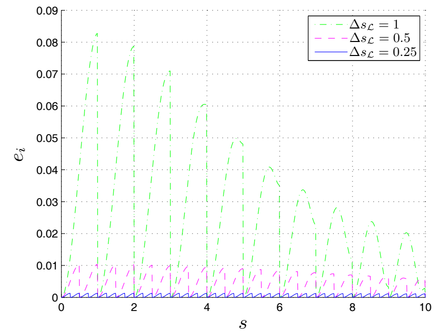
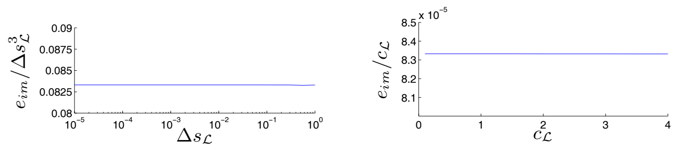


Fig. 2. Interpolation error e_i for three different values of lookup table sampling interval $\Delta s_{\mathcal{L}}$.



(a) (b)

Fig. 3. Maximum interpolation error for circular interpolation.

from where interpolated coordinates $(x_{\mathcal{L}i}(s), y_{\mathcal{L}i}(s))$ at length s are

$$\begin{aligned} x_{\mathcal{L}i}(s) &= x_{\mathcal{L}}^j + 2r_j \cos\left(\theta_{\mathcal{L}}^j + \tfrac{\Delta s}{2r_j}\right) \sin\tfrac{\Delta s}{2r_j} \\ y_{\mathcal{L}i}(s) &= y_{\mathcal{L}}^j + 2r_j \sin\left(\theta_{\mathcal{L}}^j + \tfrac{\Delta s}{2r_j}\right) \sin\tfrac{\Delta s}{2r_j}, \end{aligned} \tag{6}$$

where $\Delta s = s - j\Delta s_{\mathcal{L}}$ is distance along the basic clothoid between j-th point in the lookup table and an interpolated point.

For illustration, comparison of interpolation error for linear and circular interpolation is given in Fig. 1. It can be seen that for the circular interpolation maximum approximation error at a sampling interval decreases with clothoid length. With this kind of circular interpolation a continuity between segments is not preserved, but with dense sampling gaps are not notable in real applications.

Remark 1: Note that accuracy can be increased by using a circular arc that goes through both points therefore avoiding gaps, such as in [18]. However we found that this decreases efficiency and complicates estimation of the maximum error as in this case maximum error at a sampling interval is not monotonically decreasing with s.

To find out how the values of parameters $\Delta s_{\mathcal{L}}$ and $c_{\mathcal{L}}$ are related with the interpolation error an analysis is conducted. First, we kept constant sharpness of the basic clothoid $c_{\mathcal{L}} = 1$, and compared interpolation error for three values of sampling interval $\Delta s_{\mathcal{L}}$. The interpolation error is shown in Fig. 2. One can notice that maximum interpolation error always occurs between first two points of the basic clothoid. Further, interpolation error lowers with clothoid length $s_{\mathcal{L}}$ and converges to zero. To perform a more detailed analysis in the sequel we consider only the maximum interpolation error which we denoted by e_{im}.

Remark 2: The interpolation error converges to zero because a clothoid is a spiral that converges to a point (as Fresnel integrals converge to ½ [11]).

Next, the sharpness is kept constant at $c_{\mathcal{L}} = 1$ while sampling interval $\Delta s_{\mathcal{L}}$ is changed. In Fig. 3(a) maximum interpolation error e_{im} is displayed, so that it is visible that ratio $e_{im}/\Delta s_{\mathcal{L}}^3$ is constant for this case. This means that halving $\Delta s_{\mathcal{L}}$ results with lowering maximum interpolation error by the factor $1/2^3$ (also visible in Fig.

2). Therefore, if the lookup table has n points, then approximation error is of order $\mathrm{O}(1/n^3)$.

This can be verified as follows. The approximation error has its maximum at the end of the first interpolation interval. For this interval it is $x_{\mathcal{L}} = y_{\mathcal{L}} = 0$. From (4), as $j = 0$, we have $r_j = \frac{2}{c_{\mathcal{L}} \Delta s_{\mathcal{L}}}$. Also from (5), $\theta_{\mathcal{L}}^j = 0$. We substitute these values into (6), and considering that at the end of the first interval it is $\Delta s = \Delta s_{\mathcal{L}}$, the interpolated value for x-component becomes

$$\begin{aligned} x_{\mathcal{L}i}(\Delta s_{\mathcal{L}}) =& \frac{4}{c_{\mathcal{L}} \Delta s_{\mathcal{L}}} \cos \frac{c_{\mathcal{L}} \Delta s_{\mathcal{L}}^2}{4} \sin \frac{c_{\mathcal{L}} \Delta s_{\mathcal{L}}^2}{4} \\ =& \frac{2}{c_{\mathcal{L}} \Delta s_{\mathcal{L}}} \sin \frac{c_{\mathcal{L}} \Delta s_{\mathcal{L}}^2}{4} = \int_0^{\Delta s_{\mathcal{L}}} \cos \frac{c_{\mathcal{L}} \Delta s_{\mathcal{L}} \xi}{2} d\xi. \end{aligned}$$

Using this and (1a), and considering that the x-component of the maximum error is $x_{im} = x_{\mathcal{L}i}(\Delta s_{\mathcal{L}}) - x_{\mathcal{L}}(\Delta s_{\mathcal{L}})$, we have

$$x_{im} = \int_0^{\Delta s_{\mathcal{L}}} \left(\cos \frac{c_{\mathcal{L}} \Delta s_{\mathcal{L}} \xi}{2} - \cos \frac{c_{\mathcal{L}} \xi^2}{2} \right) d\xi \tag{7}$$

To obtain approximation error, we use the midpoint rule:

$$\int_a^b f(u) du = h f\left(\frac{a+b}{2}\right) + \mathrm{O}(h^3),$$

where $h = b - a$. Using this rule and (7), we obtain

$$x_{im} = \Delta s_{\mathcal{L}} \left(\cos \tfrac{c_{\mathcal{L}} \Delta s_{\mathcal{L}}}{4} - \cos \tfrac{c_{\mathcal{L}} \Delta s_{\mathcal{L}}}{8} \right) + \mathrm{O}(\Delta s_{\mathcal{L}}^3) = \mathrm{O}(\Delta s_{\mathcal{L}}^3).$$

From this, considering that number of points in the lookup table is $n = \mathrm{floor}(s_{\mathcal{L}} / \Delta s_{\mathcal{L}}) + 1$, we obtain $x_{im} = \mathrm{O}(1/n^3)$. The same can be obtained for the y-component of the maximum error. As we define the interpolation error as Euclidean distance, for maximum interpolation error e_{im} we also have $e_{im} = \mathrm{O}(1/n^3)$, which verifies our statement.

In Fig. 3(b) maximum interpolation error as a function of basic clothoid sharpness $c_{\mathcal{L}}$ is displayed for $\Delta s_{\mathcal{L}} = 0.1$, so that it is visible that ratio $e_{im}/c_{\mathcal{L}}$ is constant when $c_{\mathcal{L}}$ is changed. In other words maximum interpolation error is proportional with $c_{\mathcal{L}}$. Thus, halving $c_{\mathcal{L}}$ also halves e_{im}, which is expected because curvature then grows half as fast. From this analysis, an approximative relation can be derived that estimates maximum interpolation error of circular interpolation

$$e_{im} < 0.084 c_{\mathcal{L}} \Delta s_{\mathcal{L}}^3. \tag{8}$$

Remark 3: Possible improvement in the interpolation algorithm could be the use of non-constant sampling interval to better interpolate parts where error is higher, however, computational burden would then increase. One could also use other interpolation curves instead of circular arcs, e.g. s-power series [15]. However, up to now we have not found a curve that is better in terms of efficiency, implementation complexity, and accuracy, until higher order approximations are used, in which case we lose efficiency.

IV. Determination of a Required Set of Clothoids

When working with clothoids, usually a required set of clothoids, i.e. an allowed range of clothoid parameters, suitable for particular application can be found. This is important because in this way a required size and sampling interval of the lookup table can be bounded. Without loss of generality, in the following sections only clothoids in the first quadrant will be assumed, unless stated otherwise. Thus all parameters of a clothoid are nonnegative, i.e. $s \geq 0, c \geq 0, \kappa_0 \geq 0$. Clothoid initial point and tangent angle are assumed to be zero, i.e. $x_0 = y_0 = 0, \theta_0 = 0$.

A. Bounding the Clothoid Orientation Change and Length

In praxis overall orientation change $\Delta\theta$ that occurs along a single clothoid can be bounded, e.g. in path planning an orientation change greater than $\Delta\theta_{max} = \pi/2$ is rarely required [20]. Thus, assuming $\kappa_0 \geq 0$ and using (2), the length of the clothoid can be upper bounded

$$s \leq -C^2 \kappa_0 + C \sqrt{C^2 \kappa_0^2 + 2\Delta\theta_{max}}. \tag{9}$$

The maximum length of the clothoid can be additionally limited by some fixed upper-bound s_{max}, which can be determined e.g. based on the size of a vehicle. If properly chosen, this is no significant limitation for the planner, because it can avoid unnecessarily long clothoids by using lines and circular arcs instead. Using (9), the clothoid length is now upper bounded by

$$s \leq \min \left(-C^2 \kappa_0 + C \sqrt{C^2 \kappa_0^2 + 2\Delta\theta_{max}}, s_{max} \right). \tag{10}$$

In the sequel we mostly use the case when $\kappa_0 = 0$, so that it becomes

$$s \leq \min \left(C \sqrt{2\Delta\theta_{max}}, s_{max} \right). \tag{11}$$

Note that (10) is always stronger than (11) (for the first quadrant), however we use (11) because (10) is more complex and hard to use in further analysis.

B. Determination of the Minimum Scaling

When using clothoids in path planning, we usually work with some typical range of clothoid scalings. However, to be on the safe side we must also consider extreme cases and exactly predict what happens then in order not to exceed maximum allowed approximation error. One extreme case occurs when scaling is very low, and therefore the problem is that argument of the lookup table in eq. (3) can become very high, possibly higher than the length of the lookup table. To prevent this, we determine minimum scaling based on the maximum allowed approximation error. Namely, it can be shown that every clothoid is a bounded spiral so that it can be fit inside its bounding circle. As scaling C decreases, this bounding circle becomes smaller. In the limit case $C = 0$, a clothoid reduces to a point. This means that for scaling smaller than some value C_{min} clothoid can be approximated by a point without exceeding specified maximum approximation error.

We define approximation error e as the Euclidean distance between an exact clothoid point (x_e, y_e) and its approximating point (x_a, y_a)

$$e = \sqrt{(x_e - x_a)^2 + (y_e - y_a)^2}. \tag{12}$$

For the case of approximating a clothoid by its start point, considering the worst case when $\kappa_0 = 0$ and using (1), a corresponding approximation error e_{pt} is the Euclidean distance of the most distant clothoid point to the origin. It can be expressed as a function of C

$$e_{pt}(C) = \sqrt{x_d(C)^2 + y_d(C)^2},$$

so that

$$x_d(C) = \int_0^{s_d} \cos \frac{\xi^2}{2C^2} d\xi, \quad y_d(C) = \int_0^{s_d} \sin \frac{\xi^2}{2C^2} d\xi,$$

$$s_d = \arg\max_s \left(\int_0^s \cos \frac{\xi^2}{2C^2} d\xi \right)^2 + \left(\int_0^s \sin \frac{\xi^2}{2C^2} d\xi \right)^2,$$

where (x_d, y_d) is a point of the clothoid that is most distant from its start point (the origin), and s_d is the length at which this happens.

We can find the minimum scaling C_{min} by computing the scaling at which the approximation error e_{pt} is equal to maximum allowed error e_{max}, i.e. by solving nonlinear equation $e_{pt}(C_{min}) = e_{max}$ for C_{min}. This happens when the clothoid just touches, but not intersects a circle with center $(0, 0)$ and radius e_{max}. It can be shown that s_d is proportional to C, and s_d/C can be found by classical optimization methods, as well as $e_{pt}(1)$. Then, $e_{pt}(C) = Ce_{pt}(1)$ and

$$C_{min} = e_{max}/e_{pt}(1). \tag{13}$$

Finally, the queries with extremely low scaling where the argument in eq. (3) would be out of the range are treated simply by returning clothoid start point $x = x_0,\ y = y_0$, without exceeding maximum allowed error e_{max}.

C. Determination of the Maximum Scaling

Another extreme case of a clothoid occurs at very high scaling and the problem in this case is that the error could grow too high because approximation error is proportional with scaling. Therefore, scaling should be upper-bounded, too. First we will notice that when $\kappa_0 = 0$ and $C = \infty$, a clothoid does not change the tangent orientation and is reduced to a straight line. Consequently, in case that clothoid has extremely high scaling it can be approximated by a straight line.

Using (1) and (12), the error e_{ln} of approximating clothoid with zero initial curvature by a straight line at length s is

$$e_{ln}(C, s) = \sqrt{\left(s - \int_0^s \cos\frac{\xi^2}{2C^2} d\xi\right)^2 + \left(\int_0^s \sin\frac{\xi^2}{2C^2} d\xi\right)^2}. \tag{14}$$

Initially, the error grows with clothoid length, because clothoid departs more from approximating line as length grows, so that we will consider the worst case at maximum clothoid length $s = s_{max}$ (due to spiral nature, this is no more true at orientation change higher than π, however, as we here consider large scalings, we did not find this to be important in praxis). Now we find the maximum allowed scaling C_{max} by finding the scaling at which (14) is equal to specified maximum allowed error e_{max}. Therefore, we solve the following nonlinear equation for C_{max}

$$e_{ln}(C_{max}, s_{max}) = e_{max} \tag{15}$$

by using Matlab Optimization and Symbolic toolboxes [28]. We upper-bound the scaling so that $C \leq C_{max}$. If there, however, a query occurs with $C > C_{max}$, we can safely treat it as $C = \infty$ case without violating the maximum allowed error e_{max}. Such a query is answered simply by solving (1) for $C = \infty$ and $\kappa_0 = 0$, so that it reduces to the following straight-line equation

$$\begin{bmatrix} x(s) \\ y(s) \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \begin{bmatrix} \cos(\theta_0) \\ \sin(\theta_0) \end{bmatrix} s.$$

D. Determination of the Maximum Initial Curvature

In some applications we need the case when clothoid initial curvature is nonzero, e.g. when path replanning is required for a vehicle that currently executes a circular path. In this case yet another special case occurs when the clothoid has nonzero initial curvature $\kappa_0 \neq 0$ and scaling $C = \infty$. Then the clothoid is reduced to a circle with radius $1/\kappa_0$. We introduce a factor K defined as

$$K = \kappa_0 C,$$

which will serve as a measure of similarity of a clothoid to a circle. By increasing K, a clothoid becomes more similar to a circle. So instead of bounding κ_0, we will bound K because by upper bounding K we can control how much a clothoid with initial curvature κ_0 and scaling C departs from a circular arc with radius $r = 1/|\kappa_0|$. If it departs less than predefined maximum error e_{max}, the clothoid can be approximated by a circle without exceeding maximum error.

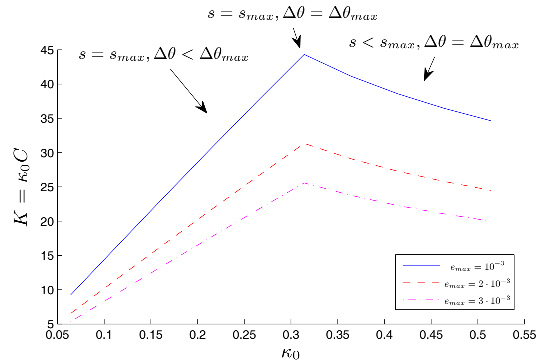


Fig. 4. Dependency of K on initial curvature κ_0, where K is computed so that the maximum error between clothoid and circular arc is exactly e_{max}, i.e. by solving (16) so that $e_{arc}(\kappa_0, C, s') = e_{max}$, where s' is upper-bounded by (17). Three values of e_{max} were evaluated. It can be seen that tighter error tolerance results with higher K and K is at its maximum when both $s = s_{max}$ and $\Delta\theta = \Delta\theta_{max}$.

To express the error, assume $\kappa_0 \neq 0$ and $C = \infty$. Then, according to (1), a clothoid reduces to a circular arc of radius $1/|\kappa_0|$, whose point (x_a, y_a) at length s is

$$x_a(s) = \frac{1}{\kappa_0}\sin(\kappa_0 s), \quad y_a(s) = \frac{1}{\kappa_0}(1 - \cos(\kappa_0 s)).$$

From here, according to error definition (12), and using (1), the error e_{arc} of approximating clothoid by a circular arc at length s is

$$\begin{aligned} e_{arc}^2(\kappa_0, C, s) = &\left(\tfrac{1}{\kappa_0}\sin(\kappa_0 s) - \int_0^s \cos(\kappa_0\xi + \tfrac{\xi^2}{2C^2}) d\xi\right)^2 \\ &+ \left(\tfrac{1}{\kappa_0}(1 - \cos(\kappa_0 s)) - \int_0^s \sin(\kappa_0\xi + \tfrac{\xi^2}{2C^2}) d\xi\right)^2. \end{aligned} \tag{16}$$

It could be proved that initially, as approximating circle departs more from the clothoid, this error becomes higher as length of the clothoid grows (due to spiral behavior, this is no more true at very large lengths, however, as allowed error is typically small, we did not find this case to be important in praxis). Thus the worst case occurs at maximum clothoid length $s = s_{max}$ or at maximum orientation change $\Delta\theta = \Delta\theta_{max}$, whichever case occurs first. Therefore, using (2), to bound error we upper bound clothoid length by

$$s \leq \min\left(\frac{\Delta\theta_{max}}{|\kappa_0|}, s_{max}\right). \tag{17}$$

To find a maximum value of K so that maximum error is not exceeded, we change the initial curvature κ_0 and search scaling C such that error (16) is exactly $e_{arc} = e_{max}$ for each particular value of κ_0, while keeping length of the clothoid bounded by (17). We found that K has its maximum at value of curvature:

$$\kappa_{0me} = \Delta\theta_{max}/s_{max},$$

At this curvature both clothoid length and orientation change are at its maximum. This is illustrated in Fig. 4. Then K is maximal because the maximum scaling (relative to initial curvature) is required in order not to exceed the maximum allowed approximation error. If we denote this worst-case scaling as C_{me}, the upper-bound of K can be expressed as

$$K_{max} = \kappa_{0me} C_{me}, \tag{18}$$

where C_{me} is found by solving nonlinear equation

$$e_{arc}(\kappa_{0me}, C_{me}, s_{max}) = e_{max}, \quad (19)$$

which can be solved using classical optimization methods [28].

The queries where it is not $K \le K_{max}$ are answered by solving (1) for $C = \infty$, which yields the following circular arc equation:

$$\begin{bmatrix} x(s) \\ y(s) \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \frac{1}{\kappa_0} \begin{bmatrix} -\sin\theta_0 + \sin(\theta_0 + \kappa_0 s) \\ \cos\theta_0 - \cos(\theta_0 + \kappa_0 s) \end{bmatrix}.$$

V. Determination of the Lookup Table Parameters

When the required range of clothoid parameters is determined, it is necessary to find a safe mapping between these parameters and parameters of the basic clothoid in the lookup table in order not to exceed the maximum approximation error.

First, to obtain required length of the basic clothoid $s_{\mathcal{L}}$, maximum possible values of the lookup table arguments, i.e. arguments of $x_{\mathcal{L}}$ and $y_{\mathcal{L}}$ in (3), must be found. To do this we rewrite the first argument in (3) as

$$\sqrt{\frac{|c|}{c_{\mathcal{L}}}} \left(s + \frac{\kappa_0}{c} \right) = C_{\mathcal{L}} \left(\frac{s}{C} + K \right), \quad (20)$$

where $C_{\mathcal{L}} = 1/\sqrt{c_{\mathcal{L}}}$ is scaling of the basic clothoid stored in the lookup table. The second argument in (3) need not to be examined because its absolute value is always less than or equal to absolute value of the first argument.

We examine the maximum value of the first term in (20). By substituting the worst case bound of the clothoid length (11) it is

$$\frac{s}{C} = \min\left(\sqrt{2\Delta\theta_{max}}, \frac{s_{max}}{C} \right),$$

whose maximum is

$$\left[\frac{s}{C} \right]_{max} = \min\left(\sqrt{2\Delta\theta_{max}}, \frac{s_{max}}{C_{min}} \right), \quad (21)$$

where C_{min} is the minimum clothoid scaling given by (13).

Regarding the second term in (20), we have already determined its maximum value by determining the maximum factor K in (18). Although both terms in (20) cannot be at their maximums instantaneously, for the simplicity we take the worst case and by using (21) write the maximum argument of the lookup table as

$$s_{\mathcal{L}max} = C_{\mathcal{L}} \left(\min\left(\sqrt{2\Delta\theta_{max}}, \frac{s_{max}}{C_{min}} \right) + K_{max} \right). \quad (22)$$

Another problem to consider is that interpolation error introduced in coordinates ($x_{\mathcal{L}}$, $y_{\mathcal{L}}$) returned by the lookup table in (3) can be increased because of scaling by factor $C/C_{\mathcal{L}}$ in (3). Error introduced in (3) will stem from m queries for basic clothoid coordinates $(x_{\mathcal{L}}, y_{\mathcal{L}})$, where $m = 1$ if initial curvature κ_0 is zero, and $m = 2$ for $\kappa_0 \neq 0$. We assume the worst case where error is at its maximum for all queries so that we have the following approximation error

$$e = m\sqrt{\frac{c_{\mathcal{L}}}{|c|}} e_{im} = m\frac{C}{C_{\mathcal{L}}} e_{im}, \quad m = 1, 2 \quad (23)$$

where e_{im} is the maximum interpolation error. If we substitute the estimated maximum circular interpolation error (8) into this, bound the maximum allowed error as $e \le e_{max}$, and consider the worst case $C = C_{max}$, the following condition on the lookup-table maximum sampling interval is obtained

$$\Delta s_{\mathcal{L}max} \approx (0.084m)^{-\frac{1}{3}} C_{\mathcal{L}} \sqrt[3]{\frac{e_{max}}{C_{max}}}. \quad (24)$$

With this choice of the maximum sampling interval the maximum error e_{max} will never be reached, even in the worst case. However, in praxis the worst case occurs very rarely, if ever. Therefore it is useful to introduce the approximation error in the typical case. We define a typical error e_{typ} as the approximation error that occurs at the typical value of the scaling C_{typ} for particular application. In this way the maximum sampling interval becomes

$$\Delta s_{\mathcal{L}max} \approx (0.084m)^{-\frac{1}{3}} C_{\mathcal{L}} \sqrt[3]{\min\left(\frac{e_{max}}{C_{max}}, \frac{e_{typ}}{C_{typ}} \right)}. \quad (25)$$

Further, by dividing the minimum required length of the lookup table (22) and the maximum sampling interval (25), we obtain minimum number of points in the lookup table. We can notice that this number is invariant to the basic clothoid scaling $C_{\mathcal{L}}$. Therefore, $C_{\mathcal{L}}$ is redundant so that any value for it can be chosen. We use $C_{\mathcal{L}} = 1$ in the sequel.

The final procedure for determination of the lookup table parameters can now be summarized in two steps:

1) Based on condition (22), choose the basic clothoid length $s_{\mathcal{L}}$ so that

$$s_{\mathcal{L}} \ge \min\left(\sqrt{2\Delta\theta_{max}}, \frac{s_{max}}{C_{min}} \right) + K_{max}. \quad (26)$$

2) Based on condition (25), choose the basic clothoid sampling interval $\Delta s_{\mathcal{L}}$ so that

$$\Delta s_{\mathcal{L}} < (0.084m)^{-\frac{1}{3}} C_{\mathcal{L}} \sqrt[3]{\min\left(\frac{e_{max}}{C_{max}}, \frac{e_{typ}}{C_{typ}} \right)}, \quad (27)$$

where $m = 1$ if for all queries in (3) initial curvature κ_0 is zero, and $m = 2$ otherwise.

VI. Algorithms for Clothoid Computation

A. Computation of Clothoid Coordinates

There still remains problem of computing the coordinates of the basic clothoid in the lookup table. Algorithms for numerical integration are appropriate for this task, since previously computed results are propagated to obtain new solutions. In this way each successive call of the integration procedure benefits from the results obtained in the previous call, as opposed to methods that compute clothoid points in a single point. For this task e.g. the Runge-Kutta numerical integration method [13] can be utilized. An alternative approach is to use a software with built-in procedures for evaluation of Fresnel integrals, such as Matlab and its Symbolic Toolbox, which can compute clothoid coordinates with specified accuracy.

Remark 4: Note that for computation of the basic clothoid the computational efficiency is not of big importance because lookup table can be computed offline in the initialization stage.

B. Querying the Clothoid Coordinates

In order to query coordinates of the basic clothoid from the lookup table, a procedure called GetBasicClothoidCoords is designed, whose pseudocode is enlisted in Alg. 1. Input of the algorithm is the clothoid length s and outputs are the basic clothoid coordinates $(x_{\mathcal{L}}, y_{\mathcal{L}})$ at this length. Values $s_{\mathcal{L}}$ and $\Delta s_{\mathcal{L}}$ that are used in the algorithm are parameters of the lookup table.

Normally it should never occur that the length s is greater than the range stored in the table as the planner uses clothoids of limited length, however if this occurs the extrapolation is added for the sake of completeness. In this case the algorithm performs an extrapolation in function Extrapolate (we use simple linear extrapolation algorithm which is not shown here).

Otherwise, the algorithm performs a circular interpolation between coordinates stored in the table according to (4), (5) and (6). As the basic clothoid is symmetrical, with the origin $(0, 0)$ as the center of

Algorithm 1: GETBASICCLOTHOIDCOORDS

Input: s
Output: $x_\mathcal{L}, y_\mathcal{L}$

$i = \text{FLOOR}(|s|/\Delta s_\mathcal{L})$
if $(i \geq nPoints - 1)$
then $(x_\mathcal{L}, y_\mathcal{L}) = \text{EXTRAPOLATE}(s)$
else $\begin{cases} s_i = i \cdot \Delta s_\mathcal{L} \\ r_i = (c_\mathcal{L} \cdot (s_i + 0.5 \cdot \Delta s_\mathcal{L}))^{-1} \\ \theta^i_\mathcal{L} = 1/2 \cdot c_\mathcal{L} \cdot s_i^2 \\ x^i_\mathcal{L} = \mathcal{L}.x(i); y^i_\mathcal{L} = \mathcal{L}.y(i) \\ ds = |s| - s_i \\ x_\mathcal{L} = x^i_\mathcal{L} + 2r_i \cos(\theta^i_\mathcal{L} + \frac{ds}{2r_i}) \sin \frac{ds}{2r_i} \\ y_\mathcal{L} = y^i_\mathcal{L} + 2r_i \sin(\theta^i_\mathcal{L} + \frac{ds}{2r_i}) \sin \frac{ds}{2r_i} \end{cases}$
if $(s < 0)$
then $\begin{cases} x_\mathcal{L} = -x_\mathcal{L} \\ y_\mathcal{L} = -y_\mathcal{L} \end{cases}$

Algorithm 2: GETGENERALCLOTHOIDPOINT

Input: $x_0, y_0, \theta_0, \kappa_0, c, s$
Output: x, y

$C = 1/(|c|)^{1/2}$
$K = \kappa_0 \cdot C$
$s_1 = C_\mathcal{L} \cdot (s/C + K)$
$s_2 = C_\mathcal{L} \cdot K$
if $(C < C_{min}$ **and** $(|s_1| > s_\mathcal{L}$ **or** $|s_2| > s_\mathcal{L}))$
then $\begin{cases} x = x_0 \\ y = y_0 \end{cases}$
else if $(\kappa_0 == 0$ **and** $C > C_{max})$
then $\begin{cases} x = x_0 + \cos(\theta_0) \cdot s \\ y = y_0 + \sin(\theta_0) \cdot s \end{cases}$
else if $(|K| > K_{max})$
then $\begin{cases} x = x_0 + 1/\kappa_0 \cdot (-\sin(\theta_0) + \sin(\theta_0 + \kappa_0 \cdot s)) \\ y = y_0 + 1/\kappa_0 \cdot (\cos(\theta_0) - \cos(\theta_0 + \kappa_0 \cdot s)) \end{cases}$
else $\begin{cases} \theta_{rot} = -\kappa_0^2/2/c + \theta_0 \\ r_{11} = r_{22} = \cos(\theta_{rot}) \\ r_{12} = -\sin(\theta_{rot}); r_{21} = -r_{12} \\ (x_1, y_1) = \text{GETBASICCLOTHOIDCOORDS}(s_1) \\ (x_2, y_2) = \text{GETBASICCLOTHOIDCOORDS}(s_2) \\ x_z = (x_1 - x_2) \cdot C/C_\mathcal{L} \\ y_z = (y_1 - y_2) \cdot C/C_\mathcal{L} \cdot \text{sgn}(c) \\ x = x_0 + r_{11} \cdot x_z + r_{12} \cdot y_z \\ y = y_0 + r_{21} \cdot x_z + r_{22} \cdot y_z \end{cases}$

symmetry, if s is negative a symmetrical point is returned in the last stage of the algorithm.

Having designed the algorithm GETBASICCLOTHOIDCOORDS for the basic clothoid coordinates retrieval, it is now easy to design a higher level procedure that calculates coordinates of a general clothoid. This procedure is called GETGENERALCLOTHOIDPOINT, whose pseudocode is enlisted in Alg. 2. Inputs of the procedure are parameters of the clothoid to be retrieved and outputs are the clothoid coordinates at length s. The procedure first checks if the parameters of a general clothoid are in the allowed range of parameters. If the answer is positive, it calls algorithm GETBASICCLOTHOIDCOORDS and uses (3) to compute the final coordinates. Otherwise, if the parameters of a general clothoid are out of range, it approximates a general clothoid by a point, a line or a circle, depending on values of the parameters (as explained in Section IV). Note that first condition could simply be $C < C_{min}$, however, this would result with loss of precision if the rest of the condition is not fullfiled, as will be shown in the numerical example in the continuation.

VII. RESULTS

An example of determining the lookup-table for a mobile robot navigating in indoor environment is given in order to illustrate parameter settings and performance of the proposed algorithms. We assume that the robot size can be approximated with a circle of 20 cm radius and that the precision of the robot position measurement is approximately 10 cm. Let the typical value of the clothoid scaling that the path planner uses be $C_{typ} = 0.5$. At this scaling we set the typical approximation error of the clothoid to $e_{typ} = 10^{-9}$ m. Regarding the size of the robot and precision of its position measurement, choosing the maximum allowed (worst case) clothoid approximation error as $e_{max} = 10^{-3}$ m should be satisfactory.

Remark 5: Lower accuracy usually suffices for path planning, however, in this example we have chosen such high accuracy (10^{-9} m) to demonstrate that our algorithm can compete with other algorithms.

Determination of a required set of clothoids: We choose maximum orientation change of a single clothoid $\Delta\theta_{max} = \pi/2$. Maximum allowed clothoid length is $s_{max} = 5$ m, which we choose based on the robot size and size of its workspace. The minimum scaling is obtained by using eq. (13) that bounds error of approximating a clothoid by a point. Using Matlab Optimization and Symbolic toolboxes it is obtained $C_{min} = 5.9447 \cdot 10^{-4}$. The maximum scaling is obtained using eq. (15) that bounds error of approximating a clothoid by a line, so that it is obtained $C_{max} = 144.34$. We suppose that we do not need the case when $\kappa_0 \neq 0$, so that $K_{max} = 0$.

Determination of the lookup-table parameters: Using condition (26) a required length of the basic clothoid that will be stored in the lookup table is determined. It is obtained $s_\mathcal{L} \geq \min(1.7725, 8410.8) = 1.7725$, and we choose $s_\mathcal{L} = 1.78$. Next, the sampling interval is obtained using (27). It is obtained $\Delta s_\mathcal{L} \leq 2.2834 \cdot \sqrt[3]{\min(6.9282 \cdot 10^{-6}, 2 \cdot 10^{-9})} = 0.0028768$, and we choose $\Delta s_\mathcal{L} = 0.00285$. It can be seen that the typical error requirement is more significant than the worst case error requirement.

The number of points in the lookup table is $n = \text{floor}(s_\mathcal{L}/\Delta s_\mathcal{L}) + 1 = 626$, which is perfectly acceptable for modern computers ($\approx$10 kB if stored in double precision). We also update the length of the basic clothoid to ensure that it is multiple of the sampling interval, and we obtain $s_\mathcal{L} = (n-1) \cdot \Delta s_\mathcal{L} = 1.78125$. The complete Matlab code needed to perform described steps and compute a lookup table is published online [28].

Execution time: The proposed algorithms are implemented in C++ programming language using double precision floating-point arithmetic and tested on a PC with 2 GHz AMD Athlon CPU and Visual C++ 2005 compiler. We also compared the obtained execution time with the RBC algorithm [19]. Both algorithms were fairly optimized, e.g. values of repeatedly used expressions are cached, and relation $\cos(x) = (1 - \sin^2(x))^{0.5}$ is utilized. Execution time was measured as an average time for 1'000'000 calls of particular algorithm. Measurement was performed five times, and a median value was taken as a final result.

On the test CPU, for retrieving coordinates of a single point of a basic clothoid with Alg. 1 execution time was 150 ns. For

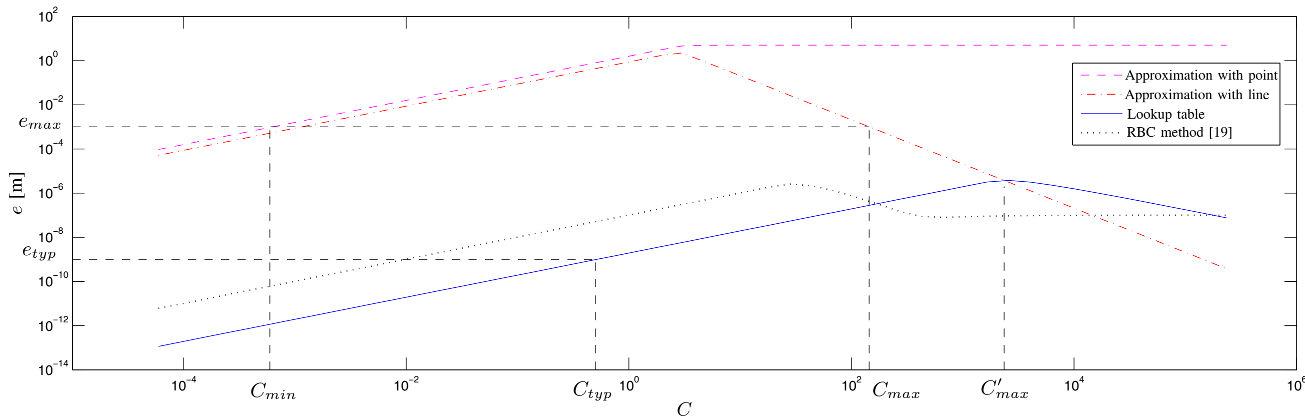


Fig. 5. Approximation error in dependency on clothoid scaling.

comparison, execution time of the RBC algorithm was 1.7 μs, which means that the proposed algorithm outperforms it for more than 11 times regarding speed, while a precision is comparable. The proposed algorithm can be further optimized by storing centers of curvature instead of clothoid points, so that there would be one trigonometric function call less.

We also measured execution time of higher level Alg. 2 that computes general clothoid. For case $\kappa_0 = 0$ the algorithm was optimized so that only one call of Alg. GETBASICCLOTHOIDCOORDS is performed, as the second call always returns $(0, 0)$ in that case. In this way execution time for retrieving a single point was 294 ns. For case $\kappa_0 \neq 0$ it was 469 ns.

If multiple points of the same clothoid are required, we optimized Alg. 2 by precomputing some parameters that remain constant, such as the rotation matrix. In this case, for $\kappa_0 = 0$, execution time for single point retrieval was 223 ns, and for $\kappa_0 \neq 0$ it was 378 ns.

Approximation error analysis: An approximation error analysis is conducted to verify that the results meet given specifications. Maximum error is measured at a broad range of scalings C, while both orientation change and the clothoid length were bounded according to specifications. The analysis is conducted in double-precision floating-point arithmetic, and values of the lookup table and the exact clothoid points are obtained using Matlab Symbolic toolbox, which can compute clothoid coordinates with maximum precision possible in double precision. The results are shown in Fig. 5. Three curves are plotted that represent approximation error in case of approximation with the initial point, the line, and the lookup table with circular interpolation (cases 1, 2 and 4 in Alg. 2, respectively).

From Fig. 5 we see that approximation with the initial point at scaling C_{min} has error exactly equal to e_{max}, as foreseen by design. In this particular example this approximation is never actually used because condition $(s_1 > s_{\mathcal{L}}$ or $s_2 > s_{\mathcal{L}})$ in Alg. 2 is never true, even when $C < C_{min}$. This means that Alg. 2 never requires values beyond the end of the lookup table. This is because at very low scalings the clothoid reaches maximum orientation change rapidly, at very short length. In Fig. 5 it can also be seen that approximation with the initial point is never the best option, as even in case $C < C_{min}$ it has the highest error.

Regarding the approximation with line, at scaling C_{max} approximation error has value e_{max}, as specified. Initially, the error is proportional with scaling, and at the scaling where clothoid reaches allowed maximum length before reaching maximum orientation change, error begins to decrease. It is important to note that even at $C > C_{max}$ the line approximation error is initially still higher compared to the lookup table based error. This is because the requirement for typical error is more stringent than maximum error requirement. However, for values $C > C'_{max}$ (see Fig. 5) the line based approximation becomes lower than the lookup table approximation error. To obtain lowest approximation error, instead of using original value C_{max} in Alg. 2, we update it so that $C_{max} = C'_{max}$.

Approximation error with the lookup table is initially proportional with the clothoid scaling. However, after the scaling value $C = \frac{s_{max}}{\Delta s_{\mathcal{L}}} = 1754.4$, the algorithm uses only the first sampling interval of the lookup table. Recalling that the approximation error is zero at the beginning of each sampling interval, and typically reaches its maximum at the end of the interval (see Fig. 2), we can explain decrease of the error at big scalings. Note also that at scaling C_{typ} the actual error is approximately equal to the required value e_{typ}.

Finally, the overall approximation error obtained with Alg. 2 will be the minimum of the three error curves in Fig. 5, in this way achieving the maximum error of about $3.7 \cdot 10^{-6}$ m near $C = 2400$. Both typical error and maximum error constraints are satisfied.

Additionally a comparison with the most up-to-date algorithm that can be found in the literature [19] is undertaken. This algorithm is based on rational Bezier curve (RBC). We used 11^{th} order RBC as in the original paper. We tested it in the same conditions (i.e. limited clothoid tangent angle and length) and the obtained approximation error is shown in Fig. 5. It can be seen that approximation error has a similar profile like our proposed algorithm. However, after approximately $C = 400$ the error begins to slightly increase, which may be due to numerical errors at high scalings.

Nonzero initial curvature: Let's now suppose that clothoids with nonzero initial curvature are required and let all other parameters remain equal as in previous example. Then the maximum K is found by solving (19) in order to bound the error of approximating a clothoid by a circular arc. It is obtained $K_{max} = 44.308$. This affects only the length of the lookup table, so that using (26) a required length of the basic clothoid is $s_{\mathcal{L}} \geq \min(1.7725, 8410.8) + 44.308 = 46.081$. The lookup table now contains 20183 points, i.e. about 315 kB in double precision, which is still acceptable for modern computers.

VIII. CONCLUSION

This paper presents a method for computation of clothoid coordinates that guarantees bounded approximation error for clothoids with any scaling or initial curvature, provided that clothoid length and orientation change are bounded. It is shown that one can compute coordinates of clothoid with any parameters from those of a single clothoid (with fixed parameters), which we store in the form of a lookup table with circular interpolation between points.

It is verified that approximation error for this interpolation is of order O($1/n^3$), where n is the number of points in a lookup table. An analysis on how to determine a required set of clothoids for path-planning application is conducted, and based on this, how to determine size and sampling of a lookup table in order not to exceed maximum approximation error. The main advantages of the method are easiness of implementation and efficiency. Compared to recent RBC method [19], our method achieves similar error and 11 times shorter computation time, so that it can be used to enable online use of existing clothoid-based algorithms, such as path planning [21], or for other time-critical real-time clothoid applications [23], [24].

References

[1] Y. J. Kanayama and B. I. Hartman, "Smooth local planning for autonomous vehicles," *The International Journal of Robotics Research*, vol. 16, no. 3, pp. 263–284, 1997.

[2] H. Delingette, M. Herbert, and K. Ikeuchi, "Trajectory generation with curvature constraint based on energy minimization," in *Proceedings of Int. Conference on Intelligent Robots and Systems (IROS '91)*, Osaka, Japan, November 1991.

[3] A. Segovia, M. Rombaut, A. Preciado, and D. Meizel, "Comparative study of the different methods of path generation for a mobile robot in a free environment," in *Proceedings of International Conference on Advanced Robotics*, Pisa, Italy, June 1991.

[4] B. Lau, C. Sprunk, and W. Burgard, "Kinodynamic motion planning for mobile robots using splines," in *IEEE/RSJ Intl. Conference on Intelligent Robots and Systems (IROS)*, St. Louis, MO, USA, 2009, pp. 2427–2433.

[5] M. Shanmugavel, A. Tsourdos, B. White, and R. Zbikowski, "Cooperative path planning of multiple UAVs using Dubins paths with clothoid arcs," *Control Engineering Practice*, vol. 18, pp. 1084–1092, 2010.

[6] S. Fleury, P. Soueres, J.-P. Laumond, and R. Chatila, "Primitives for smoothing mobile robot trajectories," *IEEE Transactions on Robotics and Automation,*, vol. 11, no. 3, pp. 441–448, June 1995.

[7] J. Boissonnat, A. Cérézo, E. Degtiariova-Kostova, V. Kostov, and J. Leblond, "Shortest plane paths with bounded derivative of the curvature," *Comptes Rendus de l'Académie des Sciences-Series I-Mathematics*, vol. 329, no. 7, pp. 613–618, 1999.

[8] T. Fraichard and A. Scheuer, "From Reeds and Shepp's to continuous-curvature paths," *IEEE Trans. Robot.*, vol. 20, no. 6, pp. 1025 – 1035, dec. 2004.

[9] K. R. Meidenbauer, "An investigation of the clothoid steering model for autonomous vehicles," Master's thesis, Virginia Polytechnic Institute and State University, Department of Mechanical Engineering, Blacksburg, Virginia, July 2007.

[10] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer *et al.*, "Autonomous driving in urban environments: Boss and the urban challenge," *Journal of Field Robotics*, vol. 25, no. 8, pp. 425–466, 2008.

[11] M. Abramowitz and I. A. Stegun, Eds., *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. NIST, 1972.

[12] J. Choi, R. Curry, and G. Elkaim, "Path planning based on Bézier curve for autonomous ground vehicles," in *Proc. of World Congress on Engineering and Computer Science*, 2008, pp. 158–166.

[13] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C — The Art of Scientific Computing*, 2nd ed. Cambridge University Press, 1992.

[14] L. Wang, K. Miura, E. Nakamae, T. Yamamoto, and T. Wang, "An approximation approach of the clothoid curve defined in the interval $[0, \pi/2]$ and its offset by free-form curves," *Computer-Aided Design*, vol. 33, pp. 1049–1058, 2001.

[15] J. Sanchez-Reyes and J. M. Chacon, "Polynomial approximation to clothoids via s-power series," *Computer-Aided Design*, vol. 35, pp. 1305–1313, 2003.

[16] M. A. Heald, "Rational approximations for the Fresnel integrals," *Math. Comp*, vol. 44, pp. 459–461, 1985.

[17] K. D. Mielenz, "Computation of Fresnel integrals II," *J.of Research of the NIST*, vol. 105, p. 589, 2000.

[18] D. S. Meek and D. J. Walton, "An arc spline approximation to a clothoid," *Journal of Computational and Applied Mathematics*, vol. 170, no. 1, pp. 59–77, September 2004.

[19] N. Montés, A. Herraez, L. Armesto, and J. Tornero, "Real-time clothoid approximation by rational Bezier curves," in *Proceedings of International Conference on Robotics and Automation*, 2008, pp. 2246–2251.

[20] Y. Kanayama and N. Miyake, "Trajectory generation for mobile robots," in *Proceedings of the International Symposium on Robotics Research*, M. Press, Ed., 1985, pp. 16–23.

[21] A. Scheuer and T. Fraichard, "Collision-free and continuous-curvature path planning for car-like robots," in *Proceedings of International Conference on Robotics and Automation (ICRA '97)*, Albuquerque, U.S.A., April 1997.

[22] M. Brezak and I. Petrović, "Path smoothing using clothoids for differential drive mobile robots," in *Proceedings of World Congress of the International Federation of Automatic Control (IFAC)*, 2011.

[23] B. Kimia, I. Frankel, and A.-M. Popescu, "Euler spiral for shape completion," *International Journal of Computer Vision*, vol. 54, pp. 157–80, 2003.

[24] C. Gackstatter, P. Heinemann, S. Thomas, and G. Klinker, "Stable road lane model based on clothoids," *Advanced Microsystems for Automotive Applications*, pp. 133–143, 2010.

[25] R. Oliveira, P. Adeodato, A. Carvalho, I. Viegas, C. Diego, and T. Ing-Ren, "A data mining approach to solve the goal scoring problem," in *Neural Networks, 2009. IJCNN 2009. International Joint Conference on*, 2009, pp. 2347–2352.

[26] D. Coombs, A. Lacaze, S. Legowik, and K. Murphy, "Driving autonomously offroad up to 35 km/h," in *Proc. of IEEE Intelligent Vehicles Symposium IVS*, Dearborn, MI., 2000.

[27] L. A. Piegl and W. Tiller, *The Nurbs Book*. Springer, 1997.

[28] M. Brezak. (2013, Feb.) Real-time approximation of clothoids with bounded error for path planning applications. [Online]. Available: http://act.rasip.fer.hr/mbrezak/ClothoidComputing.html