Rendiconti Sem. Mat. Univ. Pol. Torino
Vol. 76, 2 (2018), 47 – 56

**E. Bertolazzi* - P. Bevilacqua§ - M. Frego§**

# CLOTHOIDS: A C++ LIBRARY WITH MATLAB INTERFACE FOR THE HANDLING OF CLOTHOID CURVES

**Abstract.** A C++ library with Matlab interface for the computation of clothoid curves and related algorithms, such as interpolation problems, is herein presented. Clothoids are planar curves parametrised by arclength that exhibit linear curvature. Therefore, they are useful in a variety of applications, as they naturally model many phenomena in various fields of science. However, their transcendental nature and oscillatory behaviour (because of the Fresnel integrals) render their computation a difficult task. The present library offers a numerically accurate and efficient answer to important problems of interpolation, intersection and manipulation of these curves. It is freely available at Matlab Central.

## 1. Introduction

We present an Object Oriented (OO) library for the computation of clothoid curves (and clothoid splines) also known as Euler spirals or Cornu spirals. These planar curves are also closely connected with the name of Fresnel, as they are expressed by means of the Fresnel integrals [1]. Euler was the first to introduce the clothoid, as the curve defined by the property of having the curvature linear with respect to the arc length. Over the centuries, this characteristic has turned out useful in a number of applications. The first employments were mainly in the fields of railway design and optics. Over the years, new applications emerged, i.e. in the field of highways design, where the clothoid is used to build the motorway link road, and in general in road geometry estimation. Recent applications are in computer graphics, especially in font design [2], and also in the shaping of roller coaster. In modern computer numeric control, splines of clothoids are used to generalise the sequences of line segments and circular arcs that the cutting tool follows in the milling process. In the field of autonomous driving and robotics, clothoid curves are the optimal trajectories followed by the typical vehicle models. In the last years, the clothoid has been identified as the curve that is produced by humans when they move in a free space, that is, when they have a nonholonomic behaviour, [4, 3, 6, 5]. A novel application in medicine is the use of clothoids in the reconstruction of elastin fibres in vascular models [7].
The transcendental nature of the curve implies that it is not possible to obtain closed form solutions even for just the point evaluation. We developed a number of numerically stable and efficient algorithms that solve important problems related to the practical use of clothoids, also in real time. In particular, the proposed C++ library (with Matlab interface, that can be downloaded at [8]) allows an easy computation of the $G^1$ and $G^2$ Hermite interpolation problems [9, 10]. The library provides a simple and intuitive API to handle splines of clothoids [11], as well as to perform other useful

*Dipartimento Ingegneria Industriale – Università di Trento
§Dipartimento di Ingegneria e Scienza dell'Informazione – Università di Trento

operations, e.g. determination of intersections of curves, computation of the distance of a point from a curve.

## 2. Clothoids: definition and computational issues

In this section we introduce more formally the clothoid curve and define the Hermite Interpolation Problems that is possible to solve with the proposed library. A general clothoid curve in parametric form satisfies the differential equations,

$$x'(s) = \cos\theta(s), \qquad y'(s) = \sin\theta(s), \qquad \theta'(s) = k(s),$$

with the property that the curvature is a linear function of the arclength, $k(s) = \kappa_0 + \kappa' s$. The solution is [1, 2, 14, 13, 12, 15],

$$x(s) = x_0 + \int_0^s \cos\left(\frac{\kappa'\tau^2}{2} + \kappa_0\tau + \theta_0\right) \mathrm{d}\tau, \;\; y(s) = y_0 + \int_0^s \sin\left(\frac{\kappa'\tau^2}{2} + \kappa_0\tau + \theta_0\right) \mathrm{d}\tau,$$

where $(x_0, y_0)$ is the base point where the clothoid originates (Figure 1), $\theta_0$ and $\kappa_0$ are respectively the angle and the curvature at the base point, $\kappa'$ is the curvature change rate or sharpness, $s$ is the curvilinear abscissa. Notice that $\theta(s) = \frac{1}{2}\kappa' s^2 + \kappa_0 s + \theta_0$ and $k(s) = \kappa' s + \kappa_0$ are, respectively, the angle and the curvature at the abscissa $s$. As special cases, when $\kappa' = 0$ the curve has constant curvature, i.e. is a circle and when both $\kappa_0 = \kappa' = 0$ the curve is a straight line.

In [9] it is discussed how to recast equations (2) in terms of the classic Fresnel intergrals and their momenta:

$$C_k(s) = \int_0^s \tau^k \cos\left(\frac{\pi}{2}\tau^2\right) \mathrm{d}\tau, \qquad S_k(s) = \int_0^s \tau^k \sin\left(\frac{\pi}{2}\tau^2\right) \mathrm{d}\tau.$$

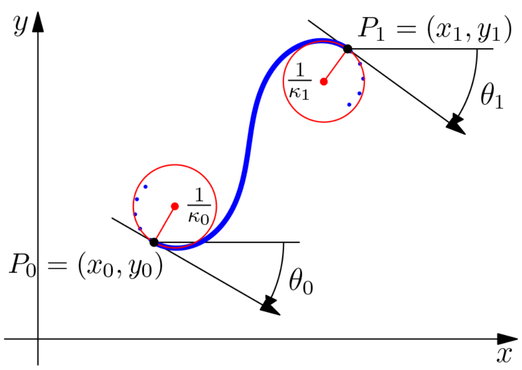



Figure 1: The general setting of a clothoid segment (in blue) joining $P_0$ to $P_1$ with initial and final angle $\theta_0$ and $\theta_1$ as well as curvatures $\kappa_0$ and $\kappa_1$. In red the osculating circles at the curve extrema, whose radius is given by the inverse of the local curvature.

These functions can be evaluated by means of recurrence relations. Unfortunately, due to the highly oscillatory nature of the Fresnel integrals, they tend to be unstable. However, for all the common problems with clothoids, only few terms of these recurrences are required. As an alternative, we propose a formulation with the Lommel function. In both cases, the computations are numerically difficult and costly (see also [16] for a discussion). Other numerical issues are represented by the possible values taken by the curvature: when $\kappa' \neq 0$ the curve is a proper clothoid, while when $\kappa' = 0$ then the curve becomes an arc of circle (for $\kappa_0 \neq 0$) or a line segment (for $\kappa_0 = 0$). The curvature parameters $\kappa_0$ and $\kappa'$ appear at the denominator of the formulas, thus another numerical difficulty is represented by the transition from a clothoid to a circle, i.e. when $\kappa' \approx 0$, and from a circle to a line, when also $\kappa_0 \approx 0$. These problems are efficiently handled by the library and are transparent to the user.

## 3. Main features of the library

In this section we briefly present the main functionalities of the proposed library. They are the $G^1$ and $G^2$ Hermite interpolation problems, the distance of a point from a clothoid, a test for the intersection of two clothoids.

### 3.1. $G^1$ and $G^2$ Hermite interpolation problems

DEFINITION 1 ($G^1$ Hermite interpolation problem). *Given two points $(x_0, y_0)$ and $(x_1, y_1)$ and two angles $\theta_0$ and $\theta_1$, the $G^1$ Hermite Interpolation Problem with clothoids asks to find the solution of the ODE (1) with boundary conditions:*

$$\begin{aligned} x'(s) &= \cos\theta(s), & x(0) &= x_0, & x(L) &= x_1, \\ y'(s) &= \sin\theta(s), & y(0) &= y_0, & y(L) &= y_1, \\ \theta'(s) &= k(s), & \theta(0) &= \theta_0, & \theta(L) &= \theta_1, \end{aligned}$$

*where $L > 0$ is the length of the curve, see Figure 1.*

This problem can be solved with a single clothoid curve, and the complete solution is discussed in [9].

DEFINITION 2 ($G^2$ Hermite interpolation problem). *Given two points $(x_0, y_0)$ and $(x_1, y_1)$, two angles $\theta_0$ and $\theta_1$ and two curvatures $\kappa_0$ and $\kappa_1$, the $G^2$ Hermite Interpolation Problem with clothoids asks to find the solution of the BVP (3) with the additional constraints on the curvature:*

$$k'(s) = u(s), \qquad k(0) = \kappa_0, \qquad k(L) = \kappa_1,$$

*where $L > 0$ is the length of the curve, $u(s)$ is a piecewise constant function to be determined that can be interpreted as a control variable, see Figure 1.*

An important difference w.r.t. the $G^1$ problem is represented by the fact that the $G^2$ problem cannot be solved with a single curve: indeed, in general, three clothoid arcs are required. There are however some special cases where solutions with one or two arcs exist. The solution algorithm is discussed in [10].

### 3.2. The distance of a point from a clothoid

A typical problem that arises in engineering applications, e.g. for the receding horizon optimal control of an autonomous vehicle that has to follow a prescribed sequence of clothoids [17, 18, 19], is to find the distance of a point $Q$ in the plane from a given curve, see Figure 2. The algorithm is discussed in [20, 21].

### 3.3. Intersection of clothoids

In principle, there can be infinitely many intersections between two clothoids, as the clothoid loops around two points at infinity. For practical applications, only a curve of finite length is considered. We use the properties of convexity/concavity of a clothoid to split the curve into pieces that are inscribed into triangles. The resulting triangles are organised into a hierarchical structure in order to determine the intersections efficiently, see Figure 3. First, each piece of curve is inscribed into a triangle constructed using the tangent lines at the extrema of the curve (Figure 3, left). The construction, as shown in [22], is well defined and uses the monotonicity of the curvature. The second step is to find the axis aligned bounding box (AABB) that encloses each triangle, see Figure 3,

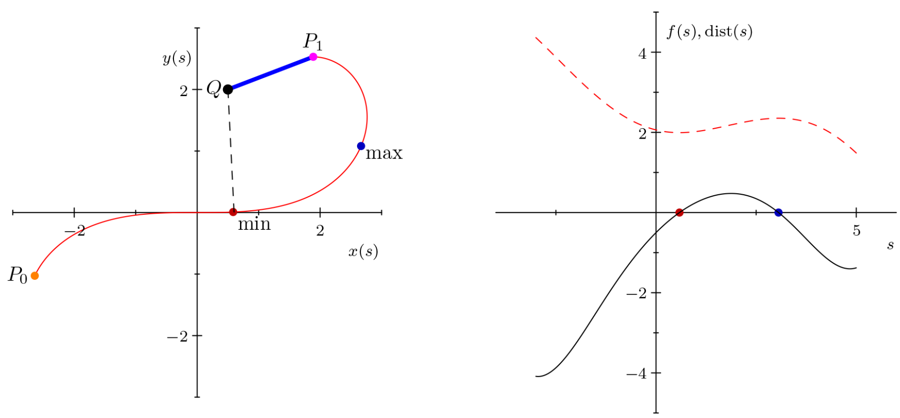



Figure 2: Left: the portion of clothoid and the point $Q$, the red dot is the critical point at minimum distance and the blue is the critical point at maximum distance. Right: the distance function is red dashed, in black the plot of of the function whose roots are the critical points of the distance. The dots are the values on the curvilinear abscissa $s$ of the points at minimum (red) and maximum (blue) distance corresponding to the plot on the left.

center. Finally, the set of boxes is organised into a hierarchical space partitioning data structure (an AABB-tree) for computational efficiency, see Figure 3, right.

## 4. Some examples of C++ library with Object Oriented MATLAB interface

The core functionalities of the library are implemented using the C++ programming language. The main reason for this choice is the computational efficiency of the compiled code. The library is composed by different classes and functions organised into a namespace called `G2lib`. Some classes represent curve primitives, and in particular line segments, circular arcs, biarcs and clothoid curves. Each of these classes exposes different methods to build the object with specific properties. For instance, a clothoid curve can be initialised both specifying its parameters or as the solution of the $G^1$ Hermite interpolation problem. Other methods allow the user to manipulate the object (rotate, reverse, scale, translate, trim, etc.) or get pointwise information about $XY$ position at a certain curvilinear abscissa, derivative, curvature, angle, and others. Moreover, each object has its own specific methods, which will be described in the sequel. These objects depend on other utility classes and functions: for example, the function that computes the Fresnel integrals is required to compute a clothoid curve; or the `Triangle2D` class is internally used to improve the efficiency of the test for intersection between two curves.

The library models the following geometric 2D objects:

- Basic elements: line segments, circular arcs, clothoid curves.
- Structured elements: biarcs (see [23]) and clothoid splines.

A biarc is the juxtaposition of two circle arcs (possibly one can be a line segment) that interpolate $G^1$ conditions, that is, point and tangent. They match at the joint point with $G^1$ continuity. The clothoid spline is a list of clothoid segments. In facts, the result of some of the problems solved by the library is not just a single clothoid segment, but a sequence of curves. The term spline is more restrictive, as it implies some kind of continuity between each curve. A clothoid list, instead, does not need to be $G^1$ continuous.

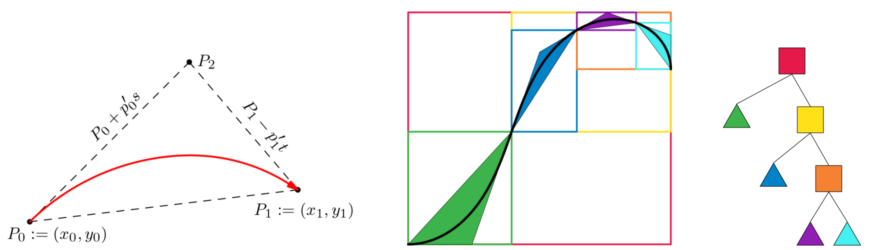



Figure 3: Left: single triangle construction with tangent lines at $P_0$ and $P_1$. Center: curve split into triangles and (hierarchical) bounding boxes. Right: corresponding hierarchical tree data structure.

```
% defining clothoid parameters
x0=0; y0=0; theta0=-3*pi/4; % initial position and angle
k0=1; dk=1; L=6;            % curvature parameters and length
% instance clothoid object
C = ClothoidCurve(); % (1)
% build clothoid curve
C.build( x0, y0, theta0, k0, dk, L); % (2)
% alternative compact version (1)+(2)
C = ClothoidCurve(x0, y0, theta0, k0, dk, L);
% plot
npts = 1000; C.plot(npts,'Linewidth',5);
```

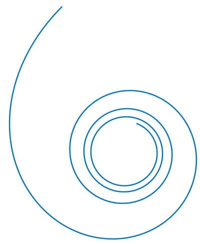


Figure 4: Example of clothoid construction using instantiation and build, or the compact instantiation with initialisation of parameters.

For each geometric object, there are the following methods: `build`, `eval` for the pointwise evaluation of the curve (possibly with offset) and its derivatives, `plot` for a graphical representation, `distance` of a point from the object, `intersect` for the intersection of two objects.
There are different ways to construct an object, for instance by defining the curve parameters or as the solution of an interpolation problem: the $G^1$ interpolation returns a clothoid curve, the $G^2$ interpolation returns a spline of one, two or three clothoid segments (depending on the nature of the solution). The next section shows some examples of the main features discussed above with snippet codes in Matlab.

## 5. Examples of usage

### 5.1. Construction of a clothoid

To define a clothoid segment the user must initialize the object and supply its parameters: the initial point $(x_0, y_0)$, the initial angle $\theta_0$, the curvature parameters $\kappa_0$ and $\kappa'$ and the length $L$, Figure 4. It is possible to specify the number of points for the plot and the usual Matlab's plotting parameters. It is possible to transform the curve with some standard functions, i.e. `translate`, `rotate`, `scale`, `reverse`. Notice that the effect is not simply the graphical transformation of the plot, but the actual update of the curve parameters according to the requested transform. An example of this is shown in Figure 5.

### 5.2. Solution of $G^1$ and $G^2$ Hermite interpolation problems

The solution of the interpolation problems has an easy syntax, but some care must be used when computing the $G^2$ interpolation: the result is not a clothoid curve, but a clothoid list (composed of up to three arcs), see Figure 6 and 7.

```
% defining clothoid parameters
...
% build clothoid
C = ClothoidCurve(x0, y0, theta0, k0, dk, L);
[xp,yp,xm,ym] = C.infinity();

C1 = ClothoidCurve(); C1.copy(C);
C2 = ClothoidCurve(); C2.copy(C);
C3 = ClothoidCurve(); C3.copy(C);

C1.translate(1,0);

C2.rotate(pi/3,xp,yp);
C2.translate(1,1);

C3.trim(2,4);
C3.translate(0,1);

% plot curve ...
```

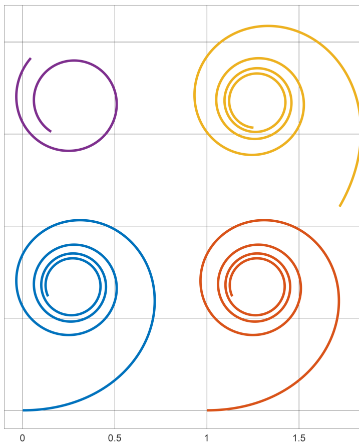



Figure 5: Library objects can be translated, rotated, scaled or trimmed.

```
x0 = 0; y0 = 0;    % initial position
x1 = 1; y1 = 0.1; % final position
theta0 = -pi/2;    % initial angle
theta1 = -pi/3;    % final angle
% instance clothoid curve
C = ClothoidCurve();
% build G1 clothoid curve
C.build_G1(x0,y0,theta0,x1,y1,theta1);
% plot ...
```

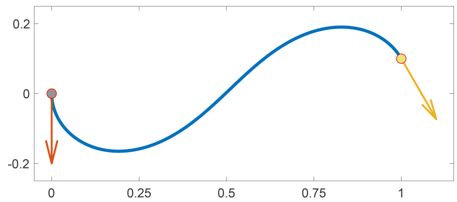



Figure 6: $G^1$ Hermite interpolation example.

```
x0 = 0; y0 = 0;    % initial position
x1 = 0.1; y1 = 1; % final position
theta0 = -pi/5;    % initial angle
theta1 = -pi/3;    % final angle
kappa0 = 7;        % initial curvature
kappa1 = -12;      % final curvature

% instance list of clothoids and build G2 interpolation with 3 arcs
CL    = ClothoidList();
iter = CL.build_3arcG2(x0,y0,theta0,kappa0,x1,y1,theta1,kappa1);
% build osculating circle
TC1 = CircleArc( x0, y0, theta0, kappa0, 2*pi/kappa0);
TC2 = CircleArc( x1, y1, theta1, kappa1, 2*pi/kappa1);
% plot
npts = 1000;
TC1.plot(npts,{'Linewidth',3,'Color',[0.7,0.7,0.7]}); hold on;
TC2.plot(npts,{'Linewidth',3,'Color',[0.7,0.7,0.7]});
CL.plot(npts);
% plot arrows...
```

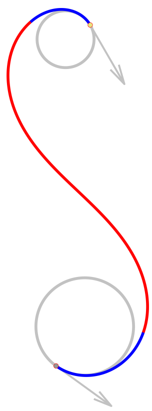


Figure 7: $G^2$ Hermite interpolation example.

```
x0=0; y0=0; theta0=pi*0.7; k0=-0.1; dk=0.05; L=30;
C1 = ClothoidCurve( x0, y0, theta0, k0, dk, L );
x0=-10; y0=0; theta0=pi/4; k0=0.1; dk=-0.05; L=30;
C2 = ClothoidCurve( x0, y0, theta0, k0, dk, L );
npts = 1000;
C1.plot(npts,'Color','red','LineWidth',5); hold on;
C2.plot(npts,'Color','black','LineWidth',5);
[s1,s2] = C1.intersect( C2 );
XY1 = C1.eval( s1 ); XY2 = C2.eval( s2 );
plot( XY1(1,:), XY1(2,:), 'ob', 'LineWidth', 2, ...
      'MarkerSize',15, 'MarkerEdgeColor','blue',...
      'MarkerFaceColor','yellow' );
```

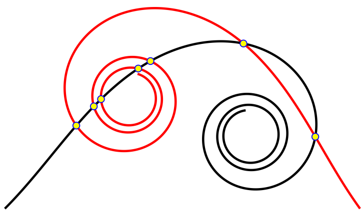


Figure 8: Example of an intersection between two clothoids. Using the OO interface, the method `intersect` of object `C1` is called to find the intersection with object `C2`.

### 5.3. Intersection of objects

Object intersection is also availabe in the library. For example, in Figure 8 it is shown how to intersect two clothoid curves. The same syntax can be used for the other objects.

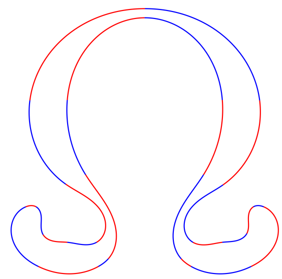
### 5.4. Spline of clothoids that minimizes a functional

The next example shows how to combine the library with the external nonlinear solver Ipopt [24] to solve a functional minimisation problem of an interpolating clothoid spline, keeping $G^2$ continuity (the algorithm is presented in [11]). The solver Ipopt is freely available, also with Matlab interface (indeed its use is transparent for the final user). The example builds a spline of clothoids from given $XY$ data, minimising the functional $P2$ that corresponds to cyclic boundary conditions. A detailed description of the list of available functionals is presented in [11], here we recall some widely used targets: $P6$ length of the spline, $P7$ energy (integral of the curvature squared), $P8$ the derivative of the curvature squared (a quantity related with the jerk). To plot the curvature and the angle of the spline there are the methods `plotCurvature` and `plotAngle`, see Figure 9.

## 6. Applications gallery and conclusions

We presented a C++ and Matlab Mex library for the manipulation of clothoid curves and splines. Related objects are circle arcs, line segments, which are particular cases of clothoids, and others. The library is suitable for real time applications (also inside Simulink) and can be compiled for embedded architectures (Beaglebone, etc.).
We used it in a number of applications regarding assistive robots [25, 26, 27] and car-like vehicles [18, 28]. Other applications of our library can be found in [29, 7, 30], the list is not comprehensive for space reasons.
Open research directions are for example the extension of the library to handle offsets (parallel curves) of the presented objects, that are of particular interest for autonomous collision avoidance and autonomous parking systems.

```
X = [2.9,2.7,2.5,1.9,1.2,1.9,2.8,0.0047,-2.8,-1.9,...
      -1.2,-1.9,-2.5,-2.7,-2.9,-2.6,-1.1,-0.9,-1.5,...
      -1.9,-0.0047,1.9,1.4,0.9,1.1,2.6,2.9];
Y = [-1.7,-1.7,-2.4,-2.6,-2.6,-1.2,0.92,3.2,0.92,...
      -1.2,-2.6,-2.6,-2.4,-1.7,-1.7,-3.2,-3.2,-3,...
      -0.92,0.93,3,0.93,-0.92,-3,-3.2,-3.2,-1.7];
CL = ClothoidSplineG2();
SL = CL.buildP2( X, Y );
% plot letter Omega
npts = 1000;
figure(); SL.plot(npts);
figure(); SL.plotCurvature(npts);
...
```



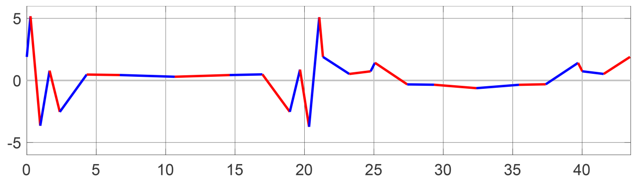



Figure 9: $G^2$ spline construction with cyclic boundary conditions (P2 problem of [11])

**References**


[1] M. Abramowitz and I. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Number 55 in National Bureau of Standards Applied Mathematics Series. U.S. Government Printing Office, Washington, D.C., 1964.

[2] R. Levien. The Euler spiral: a mathematical history. Technical Report UCB/EECS-2008-111, University of California, Berkeley, 2008.

[3] G. Arechavaleta, J.-P. Laumond, H. Hicheur, and A. Berthoz. An Optimality Principle Governing Human Walking. *IEEE Transactions on Robotics*, 24(1):5–14, Feb 2008.

[4] G. Arechavaleta, J.-P. Laumond, H. Hicheur, and A. Berthoz. On the nonholonomic nature of human locomotion. *Autonomous Robots*, 25(1-2):25–35, 2008.

[5] F. Farina, D. Fontanelli, A. Garulli, A. Giannitrapani, and D. Prattichizzo. Walking ahead: The headed social force model. *PLOS ONE*, 12(1):1–23, 01 2017.

[6] P. Bevilacqua, M. Frego, E. Bertolazzi, D. Fontanelli, L. Palopoli, and F. Biral. Path planning maximising human comfort for assistive robots. In *2016 IEEE Conference on Control Applications (CCA)*, pages 1421–1427. IEEE, 2016.

[7] T. Luo, H. Chen, and G. Kassab. 3d reconstruction of elastin fibres in coronary adventitia. *Journal of Microscopy*, 265(1):121–131, 2017.

[8] E. Bertolazzi and M. Frego. Clothoid Library. `https://github.com/ebertolazzi/Clothoids`.

[9] E. Bertolazzi and M. Frego. $G^1$ fitting with clothoids. *Mathematical Methods in the Applied Sciences*, 38(5):881–897, 2015.

[10] E. Bertolazzi and M. Frego, *On the $G^2$ Hermite interpolation problem with clothoids*, Journal of Computational and Applied Mathematics, 341 (2018), pp. 99 – 116.

[11] E. Bertolazzi and M. Frego. Interpolating clothoid splines with curvature continuity. *Mathematical Methods in the Applied Sciences*, 41(4):1723–1737, 2018.

[12] D. Connor and L. Krivodonova. Interpolation of two-dimensional curves with euler spirals. *Journal of Computational and Applied Mathematics*, 261(Supplement C):320 –332, 2014.

[13] D. Meek and D. Walton. Planar spirals that match G2 Hermite data. *Computer Aided Geometric Design*, 15(2):103 – 126, 1998.

[14] D. Meek and D. Walton. A note on finding clothoids. *Journal of Computational and Applied Mathematics*, 170(2):433 – 453, 2004.

[15] J. Stoer. Curve fitting with clothoidal splines. *J. Nat. Bureau of Standards*, 87(4):317–346, 1982.

[16] A. M. Lekkas, A. R. Dahl, M. Breivik, and T. I. Fossen. Continuous-Curvature Path Generation Using Fermat's Spiral. *Modeling, Identification and Control*, 34(4):183–198, 2013.

[17] M. Frego, E. Bertolazzi, F. Biral, D. Fontanelli, and L. Palopoli. Semi-analytical minimum time solutions for a vehicle following clothoid-based trajectory subject to velocity constraints. In *2016 European Control Conference (ECC)*, pages 2221–2227, June 2016.

[18] M. Frego, E. Bertolazzi, F. Biral, D. Fontanelli, and L. Palopoli. Semi-analytical minimum time solutions with velocity constraints for trajectory following of vehicles. *Automatica*, 86:18– 28, 2017.

[19] E. Bertolazzi and M. Frego. Semianalytical minimum-time solution for the optimal control of a vehicle subject to limited acceleration. *Optimal Control Applications and Methods*, 39(2):774–791, 2018.

[20] M. Frego, E. Bertolazzi. On the distance between a point and a clothoid curve. In *2018 European Control Conference (ECC)*, pages 1–6, June 2018.

[21] M. Frego, E. Bertolazzi. Point-clothoid distance and projection computation. 2018, submitted.

[22] M. Frego, P. Bevilacqua, E. Bertolazzi, F. Biral, D. Fontanelli, and L. Palopoli. Trajectory planning for car-like vehicles: A modular approach. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 203–209, Dec 2016.

[23] E. Bertolazzi and M. Frego. A note on robust biarc computation. Computer-Aided Design & Applications, 2019, vol. 16(5), pages 822–835.

[24] Wächter, A. and Biegler, T. L.. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Mathematical Programming, 2005, vol. 106, 1, 25–57.

[25] P. Bevilacqua, M. Frego, D. Fontanelli, and L. Palopoli. Reactive planning for assistive robots. *IEEE Robotics and Automation Letters*, 3(2):1276–1283, April 2018.

[26] V. Magnago, M. Andreetto, S. Divan, D. Fontanelli, L. Palopoli. Ruling the Control Authority of a Service Robot Based on Information Precision. *IEEE ICRA*, 2018, pages 7204–7210.

[27] P. Bevilacqua, M. Frego, D. Fontanelli, and L. Palopoli. Activity Planning for Assistive Robots, 2018, submitted.

[28] E. Bertolazzi, P. Bevilacqua, F. Biral, D. Fontanelli, M. Frego and L. Palopoli Efficient Replanning for Robotic Cars. In *2018 European Control Conference (ECC)*, pages 1068–1073, June 2018.

[29] Yun-xiao Shan, Bi-jun Li, Xiaomin Guo, Jian-Zhou, Ling-Zheng. A Considering Lane Information and Obstacle-avoidance Motion Planning Approach. 2014 IEEE 17th International Conference on Intelligent Transportation Systems (ITSC),

[30] N. Rinke, C. Schiermeyer, F. Pascucci, V. Berkhahn, B. Friedrich. A multi-layer social force approach to model interactions in shared spaces using collision prediction. Transportation Research Procedia, 25 (2017), 1249-1267.

**AMS Subject Classification: 65D17, 65D05, 65D07.**

Enrico BERTOLAZZI,
Dipartimento Ingegneria Industriale, University of Trento
via Sommarive 9, 38123 Povo (Trento), ITALY
e-mail: `enrico.bertolazzi@unitn.it`

Paolo BEVILACQUA, Marco FREGO,
Dipartimento di Ingegneria e Scienza dell'Informazione, University of Trento
via Sommarive 9, 38123 Povo (Trento), ITALY
e-mail: `{paolo.bevilacqua,marco.frego}@unitn.it`

*Lavoro pervenuto in redazione il 16-5-19.*