Interpolating Splines of Biarcs from a Sequence of Planar Points

Enrico Bertolazzi[1], Marco Frego[1,2] and Francesco Biral[1]

[1]Department of Industrial Engineering – University of Trento, Italy, enrico.bertolazzi@unitn.it
[2]Department of Information Engineering and Computer Science – University of Trento, marco.frego@unitn.it

Corresponding author: Enrico Bertolazzi, enrico.bertolazzi@unitn.it

Abstract. An algorithm for the numerical computation of a spline of biarcs that interpolates a given set of ordered planar points is presented. Biarcs are G^1 curves composed of two arcs of circle that may degenerate to line segments. The tangents at each point are free variables, which are optimised to minimise three different targets, namely: the total length of the spline, the integral of the absolute value of the curvature, the integral of the square of the curvature. Indeed other targets are possible. Conditions for the existence of the spline are given in terms of admissible point sequences and numerical experiments validate the proposed method.

Keywords: Biarc, Two-point G^1 Hermite interpolation, Geometric Continuity
DOI: https://doi.org/10.14733/cadaps.2021.66-85

1 INTRODUCTION

This work extends our previous study on biarc interpolation [7], where we discussed a new approach to the fitting problem in 2D, changing the point of view from purely geometrical to completely analytical, hence avoiding the need of distinguishing among C-shaped, S-shaped or J-shaped biarcs. These classic geometric methods, although being intuitive, do not blend smoothly among the three mentioned cases, in particular for small values of the curvature.

In our work, we showed how all possible cases can be handled in one shot with the analytic approach, which relies on a smooth formulation of the problem. The introduction of the sinc function, allowed us to write a numerically stable equation that considers a biarc which includes degenerated arcs, i.e. line segments. Another benefit of our solution is that the problem is solved in closed form. The crucial point is to avoid the division by the curvature that appears if one writes down the equation of a circle, a denominator that has to go to zero when the circle blends to a line. In the classic methods, this blending is not considered because each case is tackled with an exclusive condition on the curvature (exactly equal to zero, or different from zero). Clearly, in practical applications, when data come from experiments or sensors, it is rare to have perfect lines (i.e. curvatures that are exactly zero), indeed very small values can appear producing almost vanishing denominators and thus numerical instabilities.

Paper contribution. Aim of the present work is to extend our smooth algorithm in the computation of a spline of biarcs that interpolates a set of given ordered planar points, without information on the angles. Indeed, if the angles are specified, the spline is easily determined by a serial application of the algorithm in [7], because the problem is completely determined. When only the points are given and the angles are free, there are enough degrees of freedom to construct a nonlinear optimisation problem (NLP) that minimises a functional (e.g. length, curvature) and outputs a G^1 spline of biarcs.

There is a standalone version of the algorithm available at Matlab Central and a C++ version, with Matlab interface [3], via mex files[1].

Related work. Biarcs were originally studied by Bèzier [10], Bolton [11] and Sabin [30], in an industrial rather than academic setting. They appeared later also in textbooks, e.g. [21]. The Authors in [32] remark that biarcs are a simple but powerful class of curves that offers many interesting features, among them we recall that the arc-length can be evaluated in closed form, hence they are well suited for (real-time) numerical algorithms, e.g. numerically controlled machining CNC. Their offsets have also closed form parametrisations, the computation of the distance of a point from the curve is straightforward. Despite the simple geometric properties, they can be effectively used as primitives for approximations with a reasonably high level of quality. The tangent vector field is continuous and defined everywhere, the curvature is defined almost everywhere and is piecewise constant. Biarcs are employed successfully in various applications, for instance, in the approximation of higher degree curves [28, 19, 13] or spirals [20], they easily produce curves particularly used in CNC machining and milling, where the cutting devices follow the so-called G-code, i.e. a path composed of straight lines and circles. Other applications of biarcs are in Computer Aided Design or Manufacturing (CAD-CAM), where they are used to specify the path [33] or the offset of a more general curve [16], or approximating NURBS [27, 29] and data [26, 28, 34, 17]. These papers are mainly focused on a low degree approximation of a curve with biarcs within a specified tolerance. The idea is that there is complete knowledge of the curve to be approximated, such as tangents and lengths, the object of study is the sampling phase to produce a spline that is close to the curve within the tolerance. A different problem is considered in [19], where a point sequence is approximated (within a user tolerance) with the minimum number of arc segments. This approach is interesting also because it handles obstacles along the path.

We propose an algorithm that interpolates the original point sequence minimising a functional, thus the degrees of freedom relative to the angles are used as variables for the optimisation step.

2 BIARC FORMULATION

Before discussing the biarc spline, we briefly revise the construction of a biarc that connects two points in the plane with assigned initial and final angles with a pair of (possibly degenerate) circle arcs, see Figure 1. Let $\boldsymbol{p}_0 = (x_0, y_0)^T$ and $\boldsymbol{p}_1 = (x_1, y_1)^T$ be two points in the plane $\mathbb{R}^2$, ϑ_0 and ϑ_1 be the associated angles, then the biarc problem requires to find the solution of the following Boundary Value Problem (BVP), which is the solution of the following G^1 Hermite interpolation problem [7, 6, 4]:

$$
\begin{aligned}
x'(\ell) &= \cos\theta(\ell), & x(0) &= x_0, & x(L) &= x_1,\\
y'(\ell) &= \sin\theta(\ell), & y(0) &= y_0, & y(L) &= y_1,\\
\theta'(\ell) &= k(\ell), & \theta(0) &= \vartheta_0, & \theta(L) &= \vartheta_1,
\end{aligned}
\tag{1}
$$

[1] https://it.mathworks.com/matlabcentral/fileexchange/69782-biarc, https://github.com/ebertolazzi/Clothoids

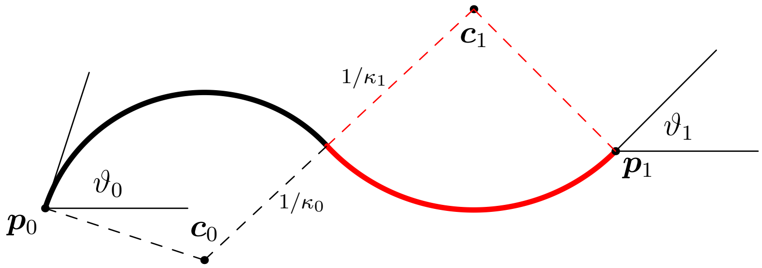

Figure 1: Standard scheme of a biarc interpolation, $\boldsymbol{p}_0 = (x_0, y_0)^T$ and $\boldsymbol{p}_1 = (x_1, y_1)^T$ are the initial and final points, ϑ_0 and ϑ_1 the associated angles. The centres of the circles of the two arcs are $\boldsymbol{c}_0$ and $\boldsymbol{c}_1$.

where the curvilinear abscissa ℓ is in the range $[0, L]$ and the curvature must be piecewise constant:

$$k(\ell) = \begin{cases} \kappa_0 & 0 \le \ell < \ell_\star, \\ \kappa_1 & \ell_\star \le \ell \le L. \end{cases} \tag{2}$$

To solve BVP (1) with (2), it is required to find the junction angle $\vartheta(\ell_\star)$ to obtain a smooth and unique solution (see [7], other choices can be found in [23]). The next lemma sets the conditions for existence and uniqueness of the solution for a biarc interpolation problem, which is useful in the discussion that follows for the spline of biarcs.

Lemma 1 (Biarc existence/uniqueness) *The solution of the BVP* (1) *with a piecewise constant curvature of the form* (2) *exists and is unique provided that*

1. *There exists ω that solves the problem*

$$\boldsymbol{p}_1 - \boldsymbol{p}_0 = \|\boldsymbol{p}_1 - \boldsymbol{p}_0\| \begin{pmatrix} \cos\omega \\ \sin\omega \end{pmatrix}, \qquad |\vartheta_0 - \omega| \le \pi, \qquad |\vartheta_1 - \omega| \le \pi.$$

2. *The junction angle $\vartheta(\ell_\star)$ satisfies $\vartheta(\ell_\star) = 2\omega - (\vartheta_0 + \vartheta_1)/2$,*
3. *The angles ϑ_1 and ϑ_0 satisfy $|\vartheta_1 - \vartheta_0| < 2\pi$.*

Proof. See Lemma 2 of reference [7] for points 1, 2 and 3. □

The next lemma shows that the solution to a biarc interpolation has a smooth dependence on the parameters $\boldsymbol{p}_0$, $\boldsymbol{p}_1$, ϑ_0 and ϑ_1 of the problem.

Lemma 2 (Biarc smoothness w.r.t. parameters) *The solution presented in the above Lemma 1 has* smooth *dependence on the parameters $\boldsymbol{p}_0$, $\boldsymbol{p}_1$, ϑ_0 and ϑ_1 provided that*

$$|\vartheta_0 - \omega| \le \pi - \varepsilon, \qquad |\vartheta_1 - \omega| \le \pi - \varepsilon, \qquad |\vartheta_1 - \vartheta_0| \le 2\pi - \varepsilon, \qquad \|\boldsymbol{p}_1 - \boldsymbol{p}_0\| \ge \varepsilon, \tag{3}$$

for an $\varepsilon > 0$. The unique solution can be explicitly written as

$$\kappa_0 = \kappa(\vartheta_0, \vartheta_1, \omega, d), \quad \kappa_1 = -\kappa(\vartheta_1, \vartheta_0, \omega, d), \quad \ell_0 = \ell(\vartheta_0, \vartheta_1, \omega, d), \quad \ell_1 = \ell(\vartheta_1, \vartheta_0, \omega, d),$$

Figure 2: Regular (left) and non regular (right) aligned points.

where $\omega = \omega(\boldsymbol{p}_0, \boldsymbol{p}_1)$ *and* $d = \|\boldsymbol{p}_1 - \boldsymbol{p}_0\|$,

$$\begin{aligned}
\omega = \omega(\boldsymbol{p}_0, \boldsymbol{p}_1) &= \operatorname{atan2}(y_1 - y_0, x_1 - x_0), \\
\kappa(\vartheta_0, \vartheta_1, \omega, d) &= \frac{2}{d}\left(\sin(\omega - \vartheta_0) + \sin\left(\omega - \frac{\vartheta_0 + \vartheta_1}{2}\right)\right), \\
\ell(\vartheta_0, \vartheta_1, \omega, d) &= \frac{d}{2}\left(\cos\left(\frac{\vartheta_1 - \vartheta_0}{4}\right)\operatorname{sinc}\left(\omega - \frac{3\vartheta_0 + \vartheta_1}{4}\right)\right)^{-1},
\end{aligned}$$

and the function $\operatorname{sinc} z$ *is defined as*

$$\operatorname{sinc} z := \frac{\sin z}{z} = 1 + \sum_{n=1}^{\infty}(-1)^n \frac{x^{2n}}{(2n+1)!}.$$

Proof. From Corollary 1 of reference [7], the definition of κ and ℓ is deduced. Lemma 2 of reference [7] gives the required existence and uniqueness of the solution. The ε in (3) is required to be away from the singular points of the smooth function κ and ℓ. □

Remark 1 *The function* $\operatorname{atan2}(y, x)$ *is the angle* ω *(argument) in the range* $(-\pi, \pi]$ *of the complex number* $x + \boldsymbol{i}\, y$ *that satisfies (see the* Arg *function section 2.1 [1])*

$$\begin{cases} x = d\cos\omega, \\ y = d\sin\omega, \end{cases} \qquad d = \left\|\begin{pmatrix} x \\ y \end{pmatrix}\right\|.$$

3 SPLINE OF BIARCS

This section is devoted to the study of a spline of biarcs, that is, a sequence of those curves that interpolate given ordered points in the plane. To construct such a sequence of biarcs it is enough to determine the angle associated with each point. However, to ensure the existence of a spline through a set of assigned points, a regularity condition must be enforced.

Definition 1 (Regular points) *A sequence of points* $\{\boldsymbol{p}_i\}_{i=0}^N$ *is regular if* $\|\boldsymbol{p}_{i-1} - \boldsymbol{p}_i\| > 0$ *for* $i = 1, 2, \ldots N$ *and the angle of three consecutive points is not* $\pm\pi$*, i.e.,*

$$(\boldsymbol{p}_{i+1} - \boldsymbol{p}_i) \cdot (\boldsymbol{p}_{i-1} - \boldsymbol{p}_i) + \|\boldsymbol{p}_{i+1} - \boldsymbol{p}_i\|\,\|\boldsymbol{p}_{i-1} - \boldsymbol{p}_i\| > 0, \qquad i = 1, 2, \ldots N - 1. \tag{4}$$

This definition implies that if three consecutive points $\boldsymbol{p}_{i-1}$, $\boldsymbol{p}_i$ and $\boldsymbol{p}_{i+1}$ are aligned, then the point $\boldsymbol{p}_i$ is between $\boldsymbol{p}_{i-1}$ and $\boldsymbol{p}_{i+1}$ (see Figure 2). The regularity condition of definition 1 does not exclude aligned points if they lie ordered (in the sense of the regularity condition 1) on the common line, see Figure 2.

Definition 2 (Spline of biarcs) *Let $\{\boldsymbol{p}_i\}_{i=0}^N$ be a sequence of regular planar points. A spline of biarcs is a sequence of biarcs that interpolate these points with C^1 continuity.*

The next theorem shows the existence of a spline of biarcs that interpolates a sequence of assigned regular points. The available degrees of freedom ensure also that the set that contains the solution is not empty and has more than a single element, that is, there is the opportunity to optimise the degrees of freedom in order to obtain a spline that minimises a functional, we propose in what follows some classic target functions. Before stating the theorem, it is necessary to present a technical proposition, required for its proof.

Proposition 1 *Let (x_0, y_0) and (x_1, y_1) be two points in the plane, then*

$$\operatorname{atan2}(y_1, x_1) - \operatorname{atan2}(y_0, x_0) = \operatorname{atan2}(y_1 x_0 - y_0 x_1, x_0 x_1 + y_0 y_1) + 2\pi k \tag{5}$$

where $k = 0$ if $\operatorname{atan2}(y_1, x_1) - \operatorname{atan2}(y_0, x_0) \in (-\pi, \pi]$ otherwise $k = \pm 1$.

Proof. The first consideration is that $\operatorname{Arg}(x + \boldsymbol{i}y) = \operatorname{atan2}(y, x)$, where Arg is the principal value of the argument for complex numbers. Then, considering the two points as two complex numbers, $z_0 = x_0 + \boldsymbol{i}y_0$, $z_1 = x_1 + \boldsymbol{i}y_1$, it is a classic result in complex analysis (see [1]) that the principal value of the argument of the quotient of two complex numbers is equal to the difference of the two arguments:

$$\operatorname{Arg}\left(\frac{z_1}{z_0}\right) = \operatorname{Arg}(z_1) - \operatorname{Arg}(z_0) \qquad \mod (-\pi, \pi].$$

The ratio at the left hand side can be simplified with standard computations as

$$\frac{z_1}{z_0} = \frac{x_0 x_1 + y_0 y_1}{x_0^2 + y_0^2} + \boldsymbol{i}\frac{y_1 x_0 - y_0 x_1}{x_0^2 + y_0^2},$$

where the denominator is a conformal scaling factor that can be neglected as it does not change the angles. Recasting the result in the correct interval yields the proof. $\square$

Theorem 1 (Biarc spline existence) *Let $\{\boldsymbol{p}_i\}_{i=0}^N$ be a sequence of regular points, let $\boldsymbol{p}_i$ and $\boldsymbol{p}_{i+1}$ be two consecutive points. Then, there exists an integer k_i, that defines the angle $\omega_{i+1/2}$ of the segment that joins $\boldsymbol{p}_i$ with $\boldsymbol{p}_{i+1}$, as:*

$$\omega_{i+1/2} := 2k_i\pi + \operatorname{atan2}(y_{i+1} - y_i, x_{i+1} - x_i). \tag{6}$$

These angles satisfy the condition $\left|\omega_{i-1/2} - \omega_{i+1/2}\right| < \pi$. The choice of any sequence of angles $\vartheta_i \in [\vartheta_{i,\min}, \vartheta_{i,\max}]$ defines a unique spline of biarcs, where the non empty intervals $[\vartheta_{i,\min}, \vartheta_{i,\max}]$, for $i = 1, 2, \ldots, N-1$, are defined as

$$\vartheta_{i,\min} := \begin{cases} \omega_{1/2} - \pi & i = 0, \\ \omega_{N-1/2} - \pi & i = N, \\ \max\left(\omega_{i-1/2}, \omega_{i+1/2}\right) - \pi & \textit{otherwise}, \end{cases} \quad \vartheta_{i,\max} := \begin{cases} \omega_{1/2} + \pi & i = 0, \\ \omega_{N-1/2} + \pi & i = N, \\ \min\left(\omega_{i-1/2}, \omega_{i+1/2}\right) + \pi & \textit{otherwise}. \end{cases}$$

Proof. Consider the identity (5) and let α be the angle between the vectors $(x_0, y_0)^T$ and $(x_1, y_1)^T$. Then, with standard properties of the scalar and vector product in 2D:

$$x_0 x_1 + y_0 y_1 = \ell_0 \ell_1 \cos\alpha, \quad x_0 y_1 - y_0 x_1 = \ell_0 \ell_1 \sin\alpha, \quad \ell_0 = \sqrt{x_0^2 + y_0^2}, \quad \ell_1 = \sqrt{x_1^2 + y_1^2}.$$

The angle α satisfies

$$\begin{aligned}\alpha &= \operatorname{atan2}(\sin\alpha,\cos\alpha) = \operatorname{atan2}(\ell_0\ell_1\sin\alpha,\ell_0\ell_1\cos\alpha) = \operatorname{atan2}(x_0y_1-y_0x_1, x_0x_1+y_0y_1)\\ &= \operatorname{atan2}(y_1,x_1)-\operatorname{atan2}(y_0,x_0)+2k\pi.\end{aligned} \tag{7}$$

Condition (4) implies that $|\alpha|<\pi$, and relation (7) for the difference $\omega_{i-1/2}-\omega_{i+1/2}$ gives that $\omega_{i-1/2}-\omega_{i+1/2}=\beta+2n\pi$ with $|\beta|<\pi$. Starting with $\omega_{1/2}=\operatorname{atan2}(y_1-y_0,x_1-x_0)$, the integers k_i are computed recursively. The condition $|\omega_{i-1/2}-\omega_{i+1/2}|<\pi$ implies that $\vartheta_{i,\min}<\vartheta_{i,\max}$, thus the intervals are not empty. □

The next lemma presents explicit formulas for the evaluation of points and angles on a spline of biarcs.

Lemma 3 (Biarc spline evaluation) *The pointwise evaluation along a biarc spline, inside the segment $[s_{i-1},s_i]$, can be written as*

$$x(s)=\begin{cases}x_{i-1}+f(s-s_{i-1},\vartheta_L,\kappa_L) & s\le s_\star,\\ x_i+f(s_i-s,\vartheta_R,\kappa_R) & s\ge s_\star,\end{cases}\qquad f(s,\vartheta,\kappa)=s\operatorname{sinc}\left(\frac{\kappa s}{2}\right)\cos\left(\vartheta+\frac{\kappa s}{2}\right)$$

$$y(s)=\begin{cases}y_{i-1}+g(s-s_{i-1},\vartheta_L,\kappa_L) & s\le s_\star,\\ y_i+g(s_i-s,\vartheta_R,\kappa_R) & s\ge s_\star,\end{cases}\qquad g(s,\vartheta,\kappa)=s\operatorname{sinc}\left(\frac{\kappa s}{2}\right)\sin\left(\vartheta+\frac{\kappa s}{2}\right)$$

$$\theta(s)=\begin{cases}\vartheta_L+(s-s_{i-1})\kappa_L & s\le s_\star,\\ \vartheta_R+(s_i-s)\kappa_R & s\ge s_\star,\end{cases}$$

where

$$\vartheta_L=\vartheta_{i-1},\quad \vartheta_R=\vartheta_i,\quad \kappa_L=\kappa(\vartheta_L,\vartheta_R,\omega_{i-1/2},d_{i-1/2}),\quad \kappa_R=-\kappa(\vartheta_R,\vartheta_L,\omega_{i-1/2},d_{i-1/2}),$$

with $\omega_{i-1/2}$ defined in (6) and $d_{i-1/2}=\|\boldsymbol{p}_{i-1}-\boldsymbol{p}_i\|$. Moreover, s is the arc length in the range $[s_{i-1},s_i]$ and the curvilinear abscissa of the junction point is $s_\star=s_{i-1}+\ell(\vartheta_L,\vartheta_R,\omega_{i-1/2},d_{i-1/2})$.

Proof. The proof is a direct application of the results presented in [7]. □

4 INTERPOLATION PROBLEMS

Theorem 1, for a regular set of points, defines a family of interpolating biarc splines. Since any choice of the angles associated with the points yields a biarc spline, there are degrees of freedom available to optimise a functional and thus to determine a unique spline. Keeping the interpolation points fixed, a target function T depends only on the chosen angles, i.e. it is possible to write $T(\vartheta_0,\vartheta_1,\ldots,\vartheta_N)$. The spline is selected by minimisation of the target. The problem can be stated as a constrained minimisation:

$$\begin{aligned}&\text{Minimize}\quad T(\vartheta_0,\vartheta_1,\ldots,\vartheta_N)\\ &\text{Subject to}\ \vartheta_{i,\min}\le\vartheta_i\le\vartheta_{i,\max}.\end{aligned} \tag{8}$$

Notice that the interpolation conditions on the points are not explicitly necessary because, by construction, all the possible biarc splines pass trough the interpolation points.

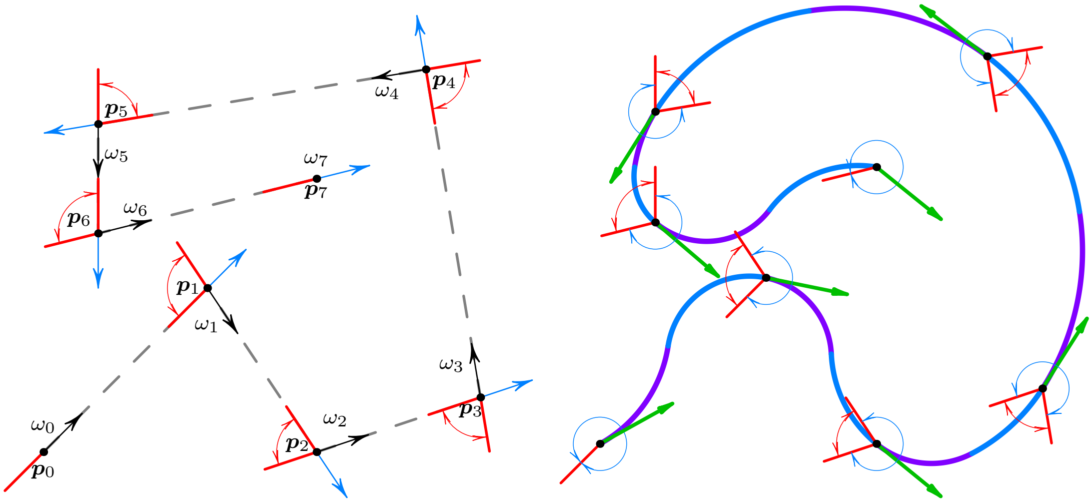

Figure 3: The standard setting for the construction of the guess (green arrow - right) starting from the piecewise linear spline (left). In blue the admissible angle intervals, in red the forbidden angle ranges.

Remark 2 *Theorem 1 ensures that the intervals $[\vartheta_{i,\min}, \vartheta_{i,\max}]$ are not empty and that the target is defined for any sequence of angles ϑ_i belonging to the associated interval. This is a Slater condition, that is, the set defined by the constraints is not empty and the target is coercive (bounded from below), which implies that the minimum optimal solution exists.*

Consider the biarc joining $\boldsymbol{p}_{i-1}$ and $\boldsymbol{p}_i$ with initial and final angles ϑ_{i-1}, ϑ_i, with $d_{i-1/2} = \|\boldsymbol{p}_i - \boldsymbol{p}_{i-1}\|$ and $\omega_{i-1/2}$ the angle of the line segment joining $\boldsymbol{p}_{i-1}$ and $\boldsymbol{p}_i$ with respect to the x-axis. For this biarc, the contribution is denoted (with a slight abuse of notation) by $T(\vartheta_{i-1}, \vartheta_i, \omega_{i-1/2}, d_{i-1/2})$ and the target becomes:

$$T(\vartheta_0, \vartheta_1, \ldots, \vartheta_N) = \sum_{i=1}^{N} T(\vartheta_{i-1}, \vartheta_i, \omega_{i-1/2}, d_{i-1/2}).$$

A good property shared by a target function is the invariance with respect to isometries, that is a composition of rotations and translations. Thus, we propose only targets that satisfy

$$T(\vartheta_L, \vartheta_R, \omega, d) = T(\vartheta_L + \alpha, \vartheta_R + \alpha, \omega + \alpha, d)$$

for any angle α. In the applications, there are special targets of interest. Among the most used in the field of (polynomial) splines, we mention the shortest length and the least curvature.
For computational reasons, it is convenient to write the target relative to a single biarc as the sum of the contributions of the two curves that build the biarc. Some classic target functions discussed next are:

P1. Minimum total length of the spline [22, 5] (target $T \equiv T_1$), which can be expressed over each biarc as

$$T_1(\vartheta_L, \vartheta_R, \omega, d) = \ell(\vartheta_L, \vartheta_R, \omega, d) + \ell(\vartheta_R, \vartheta_L, \omega, d).$$

P2. Minimum of the (absolute) curvature integral [22] (target $T \equiv T_2$), which yields

$$\begin{aligned} T_2(\vartheta_L, \vartheta_R, \omega, d) &= \int_{s_{i-1}}^{s^\star} |\kappa(s)|\, \mathrm{d}s + \int_{s^\star}^{s_i} |\kappa(s)|\, \mathrm{d}s = \left| \int_{s_{i-1}}^{s^\star} \kappa(s)\, \mathrm{d}s \right| + \left| \int_{s^\star}^{s_i} \kappa(s)\, \mathrm{d}s \right| \\ &= |\ell(\vartheta_L, \vartheta_R, \omega, d)\kappa(\vartheta_L, \vartheta_R, \omega, d)| + |\ell(\vartheta_R, \vartheta_L, \omega, d)\kappa(\vartheta_R, \vartheta_L, \omega, d)|. \end{aligned} \tag{9}$$

Target (9) can be simplified using the identity

$$2\kappa(\vartheta_L, \vartheta_R, \omega, d)\ell(\vartheta_L, \vartheta_R; \omega, d) = 4\omega - 3\vartheta_L - \vartheta_R, \tag{10}$$

hence, the contribution in (9) becomes

$$T_2(\vartheta_L, \vartheta_R, \omega, d) = \frac{1}{2}\left|4\omega - 3\vartheta_L - \vartheta_R\right| + \frac{1}{2}\left|4\omega - 3\vartheta_R - \vartheta_L\right|.$$

P3. Minimum energy [12, 15, 24, 25, 5] (integral of the curvature squared, target $T \equiv T_3$), which can be expressed over each biarc as

$$\begin{aligned} T_3(\vartheta_L, \vartheta_R, \omega, d) &= \int_{s_{i-1}}^{s^\star} \kappa(s)^2\, \mathrm{d}s + \int_{s^\star}^{s_i} \kappa(s)^2\, \mathrm{d}s \\ &= \ell(\vartheta_L, \vartheta_R, \omega, d)\kappa(\vartheta_L, \vartheta_R, \omega, d)^2 + \ell(\vartheta_R, \vartheta_L, \omega, d)\kappa(\vartheta_R, \vartheta_L, \omega, d)^2. \end{aligned} \tag{11}$$

Target (11) can be simplified using the identity (10), hence, the contribution in (11) becomes

$$T_3(\vartheta_L, \vartheta_R, \omega, d) = \frac{1}{2}(4\omega - 3\vartheta_L - \vartheta_R)\kappa(\vartheta_L, \vartheta_R, \omega, d) + \frac{1}{2}(4\omega - 3\vartheta_R - \vartheta_L)\kappa(\vartheta_R, \vartheta_L, \omega, d).$$

For target P2, a typical problem of handling the absolute value is the lack of differentiability: to overcome this problem, we implemented a regularised approximation of the modulus function, which is discussed later. The smooth approximation of the modulus, combined with Lemmas 1 and 2, guarantees the smoothness of problem (8) and its efficient solution with methods that require gradients and hessians of the target function. The selection of one specific target is application dependent, we give some examples: for the generation of a path to be followed by a wheeled (nonholonomic) vehicle, a popular choice is the shortest path, that connects an initial with a final point, thus target P1 is suitable. In this case, a spline of biarcs can be an alternative to the Markov-Dubins curves [9, 14], which also combine line segments and circle arcs. This can be effective for re-planning a short deviation (obstacle avoidance) from a nominal trajectory for high performance vehicles [2], but in other cases, where comfort of the passengers is important, target P2 or P3 may be preferred. As another application, we used target P2 to reconstruct the borders of a road, after a smoothing of the raw sensor data acquired by the car [8], where a very high accuracy is not so important compared to the speed of the computation.

We now discuss a feasible initial guess for the nonlinear programming (8) (NLP). Theorem 1 readily gives intervals where to select the optimal solution. Thus, we define the starting point of the solver as the intermediate angle (see green arrows in Figure 3 (right) inside the blue angle ranges) according to the formula:

$$\theta_0 = \omega_{1/2}, \quad \theta_N = \omega_{N-1/2}, \quad \theta_i = \left(\frac{\omega_{i-1/2}}{d_{i-1/2}} + \frac{\omega_{i+1/2}}{d_{i+1/2}} \right) \Big/ \left(\frac{1}{d_{i-1/2}} + \frac{1}{d_{i+1/2}} \right), \quad i = 1, \ldots, N-1. \tag{12}$$

The above formula weights the initial angles with the length of the line segment that connect consecutive points. Numerical evidence shows that this choice works in all the tested cases. Equation (12) is used to initialize the starting point for the NLP (8).

4.1 Derivative of Target P1

The numerical solver performance can be improved by providing the explicit expressions of the gradient and Hessian of the target function. In this section we present these formulas in a form suitable for implementation. The next quantities are given in terms of the angles and of the sinc function and its derivatives.

$$
\begin{gathered}
t_1 = \frac{\vartheta_L - \vartheta_R}{4}, \quad t_2 = \omega - \frac{3\vartheta_R + \vartheta_L}{4}, \quad t_3 = \omega - \frac{3\vartheta_L + \vartheta_R}{4}, \quad S_L = \mathrm{sinc}(t_2), \quad S_R = \mathrm{sinc}(t_3), \\
t_4 = -\frac{\mathrm{sinc}'(t_2)}{S_L}, \quad t_5 = \frac{\mathrm{sinc}'(t_3)}{S_R}, \quad t_6 = \frac{\mathrm{sinc}''(t_2)}{S_L}, \quad t_7 = \frac{\mathrm{sinc}''(t_3)}{S_R}, \quad t_8 = \tan(t_1), \quad t_9 = \cos(t_1), \\
t_{10} = \frac{t_8}{S_R} + \frac{t_8}{S_L}, \quad t_{11} = t_{10}(2t_8 + 1/t_8), \quad t_{12} = \frac{t_5(6t_5 - 2t_8) - 3t_7}{S_R}, \quad t_{13} = \frac{t_4(6t_4 - 2t_8) - 3t_6}{S_L}.
\end{gathered}
$$

The above auxiliary expressions allow us to write the target of problem $\mathrm{P1}$ as

$$
T_1(\vartheta_L, \vartheta_R; \omega) = \frac{d}{2t_9}\left(\frac{1}{S_L} + \frac{1}{S_R}\right),
$$

the gradient and the Hessian can be built with the following partial derivatives:

$$
\begin{aligned}
\frac{\partial}{\partial \vartheta_L} T_1(\vartheta_L, \vartheta_R; \omega) &= \frac{d}{8t_9}\left(t_{10} + \frac{3t_5}{S_R} - \frac{t_4}{S_L}\right) \\
\frac{\partial}{\partial \vartheta_R} T_1(\vartheta_L, \vartheta_R; \omega) &= \frac{d}{8t_9}\left(\frac{t_5}{S_R} - \frac{3t_4}{S_L} - t_{10}\right) \\
\frac{\partial^2}{\partial {\vartheta_L}^2} T_1(\vartheta_L, \vartheta_R; \omega) &= \frac{d}{32t_9}\left(t_{11} + 3t_{12} + \frac{2t_4(t_4 - t_8) - t_6}{S_L}\right) \\
\frac{\partial^2}{\partial {\vartheta_R}^2} T_1(\vartheta_L, \vartheta_R; \omega) &= \frac{d}{32t_9}\left(t_{11} + 3t_{13} + \frac{2t_5(t_5 - t_8) - t_7}{S_R}\right) \\
\frac{\partial^2}{\partial \vartheta_L \partial \vartheta_R} T_1(\vartheta_L, \vartheta_R; \omega) &= \frac{d}{32t_9}\left(t_{12} + t_{13} - t_{11}\right)
\end{aligned}
$$

4.2 Smoothing and Derivatives of Target P2

For target $\mathrm{P2}$, the absolute value is not differentiable, thus we need to use a smoothed approximation of this function. In [31], there are versions of a smoothed absolute value function, namely:

$$\begin{aligned}
\phi_1(x;\varepsilon) &= \varepsilon\left(\log\left(1+\mathrm{e}^{-x/\varepsilon}\right)+\log\left(1+\mathrm{e}^{x/\varepsilon}\right)\right) = 2\varepsilon\log\left(1+\mathrm{e}^{-|x|/\varepsilon}\right)+|x| \\
\phi_2(x;\varepsilon) &= \begin{cases} |x| & |x|\geq\varepsilon/2 \\ \frac{x^2}{\varepsilon}+\frac{\varepsilon}{4} & \text{otherwise} \end{cases} \\
\phi_3(x;\varepsilon) &= \sqrt{4\varepsilon^2+x^2} \\
\phi_4(x;\varepsilon) &= \begin{cases} |x|-\frac{\varepsilon}{2} & |x|\geq\varepsilon \\ \frac{x^2}{2\varepsilon} & \text{otherwise.} \end{cases}
\end{aligned} \tag{13}$$

Let $\phi(x;\varepsilon)=\phi_k(x;\varepsilon)$ be one of the above regularised functions, then, target $\mathrm{P2}$ becomes:

$$\int_0^L |\kappa(s)|\,\mathrm{d}s \approx \sum_{i=1}^N T\left(\vartheta_{i-1},\vartheta_i,\omega_{i-1/2},d_{i-1/2};\varepsilon\right)$$

with

$$T_2(\vartheta_L,\vartheta_R,\omega,d;\varepsilon)=\frac{1}{2}\phi\left(4\omega-3\vartheta_L-\vartheta_R;\varepsilon\right)+\frac{1}{2}\phi\left(4\omega-3\vartheta_R-\vartheta_L;\varepsilon\right).$$

We can supply the solver with the gradient and the Hessian of the target function:

$$\begin{aligned}
\nabla_\vartheta T_2(\vartheta_L,\vartheta_R,\omega,d) &= -\frac{1}{2}\phi'\left(4\omega-3\vartheta_L-\vartheta_R;\varepsilon\right)\begin{pmatrix}3\\1\end{pmatrix}-\frac{1}{2}\phi'\left(4\omega-3\vartheta_R-\vartheta_L;\varepsilon\right)\begin{pmatrix}1\\3\end{pmatrix}, \\
\nabla^2_\vartheta T_2(\vartheta_L,\vartheta_R,\omega,d)^T &= \frac{1}{2}\phi''\left(4\omega-3\vartheta_L-\vartheta_R;\varepsilon\right)\begin{pmatrix}9&3\\3&1\end{pmatrix}+\frac{1}{2}\phi''\left(4\omega-3\vartheta_R-\vartheta_L;\varepsilon\right)\begin{pmatrix}1&3\\3&9\end{pmatrix}.
\end{aligned}$$

There are no big differences in the use of the smoothing functions (13); the presented numerical tests use the simpler $\phi_3(x;\varepsilon)$. The value for ε is problem dependent, it should be as small as possible, but if it is too small, it can make the NLP hard to solve. A good choice is $\varepsilon=|\mathrm{typ}(\kappa)|\,10^{-4}$ where $\mathrm{typ}(\kappa)$ is the typical value of the curvature used in the problem.

4.3 Derivative of Target P3

The target $\mathrm{P3}$ and its derivatives can be expressed in terms of the following auxiliary functions:

$$t_1=\omega-\vartheta_L,\quad t_2=\omega-\vartheta_R,\quad t_3=4\omega-3\vartheta_L-\vartheta_R,\quad t_4=4\omega-3\vartheta_R-\vartheta_L,\quad t_5=\frac{\vartheta_L+\vartheta_R}{2}-\omega,$$

$$t_6=\cos t_1,\quad t_7=\cos t_2,\quad t_8=\sin t_1,\quad t_9=\sin t_2,\quad t_{10}=\sin t_5,\quad t_{11}=\cos t_5,$$

$$t_{12}=(t_3+t_4)t_{10},\quad t_{13}=4t_{10}-\frac{1}{2}t_{11}(t_3+t_4),\quad t_{14}=4t_{11}+\frac{1}{4}t_{10}(t_3+t_4),\quad t_{15}=t_3t_8,\quad t_{16}=t_4t_9.$$

The integral of the curvature squared is written as

$$T_3(\vartheta_L, \vartheta_R; \omega) = -\frac{1}{d}\left(t_{12} + t_{15} + t_{16}\right).$$

The derivatives of $\mathrm{P3}$ are

$$\begin{aligned}
\frac{\partial}{\partial \vartheta_L} T_3(\vartheta_L, \vartheta_R; \omega) &= \frac{1}{d}\left(t_{13} - t_3 t_6 + 3t_8 + t_9\right) \\
\frac{\partial}{\partial \vartheta_R} T_3(\vartheta_L, \vartheta_R; \omega) &= \frac{1}{d}\left(t_{13} - t_4 t_7 + t_8 + 3t_9\right) \\
\frac{\partial^2}{\partial {\vartheta_L}^2} T_3(\vartheta_L, \vartheta_R; \omega) &= \frac{1}{d}\left(t_{14} + t_{15} + 6t_6\right) \\
\frac{\partial^2}{\partial {\vartheta_R}^2} T_3(\vartheta_L, \vartheta_R; \omega) &= \frac{1}{d}\left(t_{14} + t_{16} + 6t_7\right) \\
\frac{\partial^2}{\partial \vartheta_L \partial \vartheta_R} T_3(\vartheta_L, \vartheta_R; \omega) &= \frac{1}{d}\left(t_{14} + t_6 + t_7\right).
\end{aligned}$$

5 NUMERICAL EXPERIMENTS

We validate the presented algorithm for the construction of an interpolating biarc spline over some numerical tests, most of them are taken from [18]. We report also the dataset of the considered cases, see Table 2. For each test we collected the results in Table 1.

The minimum length is interesting *per se* and also in comparison with Dubins curves, that is lines and arc segments with bounded curvature. The minimum energy is useful to obtain paths that do not contain sharp turns. Finally, the integral of the absolute curvature looks to be a good trade-off between the loose curve of minimal energy and the minimum length. This can be noticed especially in test 1 (Figure 4 - $\mathrm{P2}$), where the curve follows the intuitive path of lines and arcs. It is also interesting to compare the length of $\mathrm{P1}$ which, graphically, seems longer than $\mathrm{P2}$ but it is not. Test 2,3,4,5 represent corners: target $\mathrm{P1}$ and $\mathrm{P2}$ produce the expected shape, whereas $\mathrm{P3}$ gives a smoother path. Test 4 was taken from Figure 1 of [18], test 5 from Figure 7-8 of [18]. Test 6 represents the profile of a bottle (Figure 9 of [18]); here $\mathrm{P1}$ and $\mathrm{P2}$ show a more traditional style, whereas $\mathrm{P3}$ exhibits a more stylish (pleasing) design. Test 7 is the shape of a shoe/footprint; also in this case target $\mathrm{P2}$ is a good trade-off between $\mathrm{P1}$ and $\mathrm{P3}$. The points are sampled from Figure 2 of reference [15]. In addition, the cyclic condition $\vartheta_0 = \vartheta_N$ is added to the constraints. Finally, we present test 8 to show that the algorithm scales well with the number of points (700). The curve represents the F1 circuit track of Spa-Francorchamps (Belgium), whose data have been extracted from [3] by sampling the reference clothoid spline curves every 10 meters. The number of iterations is contained and a graphical result is shown in Figure 11. We put the picture of one target only, as the other curves are graphically indistinguishable.

6 CONCLUSION

We have presented a method to construct a spline of biarcs, which takes a sequence of ordered planar points as input. The spline has G^1 continuity and is selected via an optimisation process that minimises a target function. We proposed three possible targets, which are widely used in the construction of polynomial or clothoid splines, the minimum length, the integral of the absolute value of the curvature and the energy (curvature squared). Concretely, the algorithm consists of an NLP; for the mentioned targets we give the gradient and the Hessian to improve the performance of the numeric solver. Also an initial guess is furnished,

Table 1: Table of the results of the 8 proposed numerical tests. Each test has been conducted with the three presented targets $P1 - P2 - P3$, the results are given in terms of iterations of the nonlinear solver IPOPT and number of function evaluations. We reported the values of the targets (T) as well as the total length of the spline (L), in order to compare qualitatively and quantitatively the splines. T and L of target $\mathrm{P1}$ are of course the same.

	Target	iter.	feval.	T	L
Test 1	P1	10	11	18.0747	18.0747
	P2	22	154	18.8460	18.7084
	P3	16	38	12.1828	18.9841
Test 2	P1	9	15	13.9010	13.9010
	P2	55	270	12.6426	14.2821
	P3	14	51	12.5622	14.0510
Test 3	P1	9	15	80.8207	80.8207
	P2	55	270	3.4734	80.8617
	P3	14	51	0.4647	83.4518
Test 5	P1	9	16	72.9525	72.9525
	P2	4	45	3.0239	72.9583
	P3	8	13	0.0868	76.7610

	Target	iter.	feval.	T	L
Test 6	P1	9	12	81.9130	81.9130
	P2	13	101	3.2174	81.9139
	P3	14	26	0.4153	83.4164
Test 7	P1	17	36	82.1899	82.1899
	P2	47	248	15.6840	82.4792
	P3	33	107	5.0824	83.0096
Test 7	P1	19	249	367.61	367.61
	P2	18	298	15.241	375.12
	P3	25	286	0.263	374.48
Test 8	P1	15	34	6991.51	6991.51
	P2	4	53	61.5821	6991.52
	P3	26	45	0.76673	6991.67

which is always a feasible point. The algorithm has been validated over a test set partially present in literature and partially new. The results show that the absolute curvature is an effective cost function that is a good trade-off between the minimum length and the curvature squared.

Table 2: Table of the dataset used in the numerical experiments. For the test 8 the point sequence is too long to fit the page but is available online.

Test 1	x	0	4	4	1	1	4	4	0										
	y	0	0	1	1	2	2	3	3										
Test 2	x	0	1	2	2	2	1	0	-1	-2	-2	-2	-1	0					
	y	-2	-2	-1	0	1	2	2	2	1	0	-1	-2	-2					
Test 3	x	1	2	3	4	5	6	7	8	9	10								
	y	10	10	10	10	10	10	10.515	50	60	85								
Test 4	x	0	26	28	54														
	y	0	24	24	0														
Test 5	x	0	10	31	40.8	41	41	41	41										
	y	41	41	41	41	40.8	31	10	0										
Test 6	x	0	1.34	5	10	10.6	10.7	10.7	10.8	11.4	19.6	20.2	20.3	20.3	20.4	21	26	29.6	31
	y	0	5	8.6	10	10.4	12	28.6	30.2	30.6	30.6	30.2	28.6	12	10.4	10	8.6	5	0
Test 7	x	242	218	196	171	137	85	143	173	213	242								
	y	96	117	113	112	123	98	66	73	75	96								

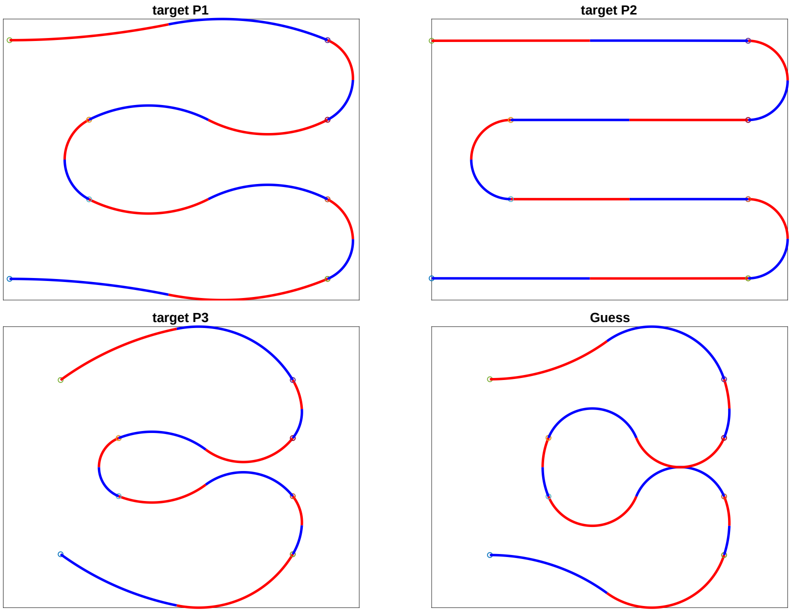

Figure 4: Test 1. Top left minimum length $(P1)$, top right integral of the absolute curvature $(P2)$, bottom left minimum energy $(P3)$, bottom right initial guess given by (12).

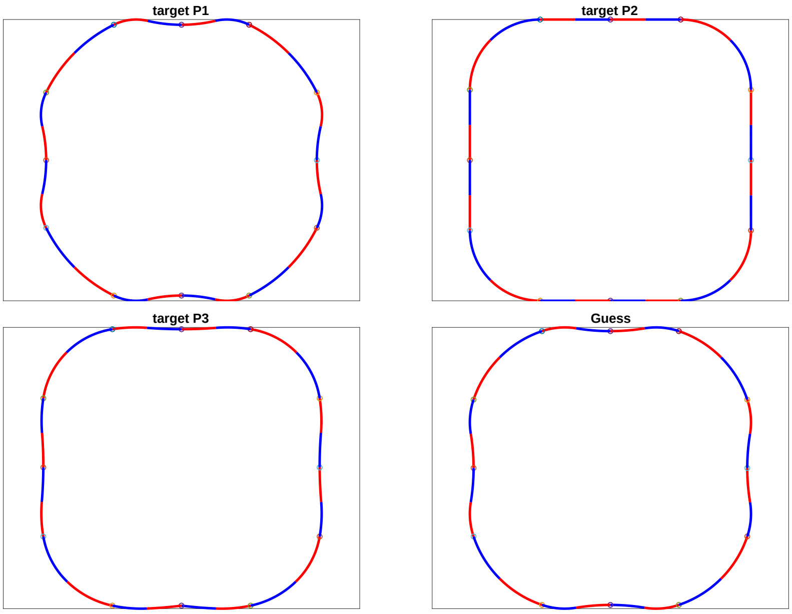

Figure 5: Test 2. Top left minimum length $(\mathrm{P1})$, top right integral of the absolute curvature $(\mathrm{P2})$, bottom left minimum energy $(\mathrm{P3})$, bottom right initial guess given by (12).

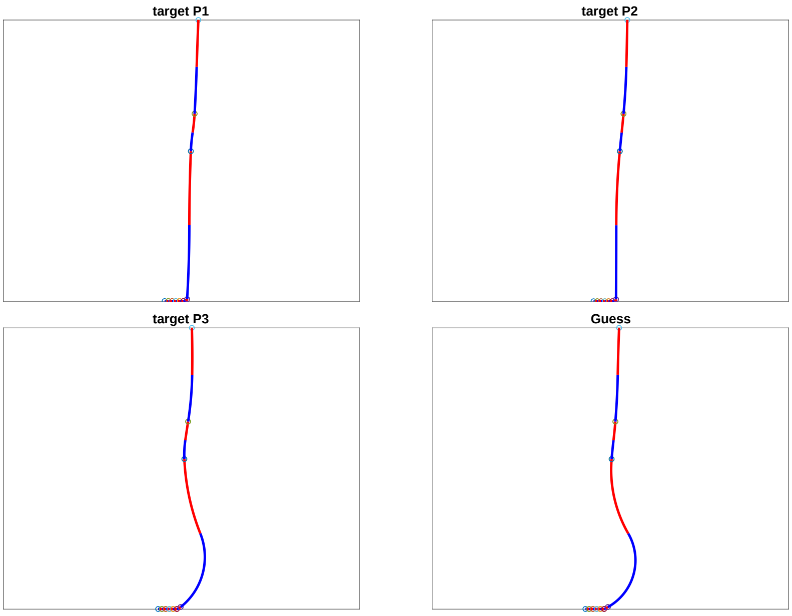

Figure 6: Test 3. Top left minimum length $(\mathrm{P1})$, top right integral of the absolute curvature $(\mathrm{P2})$, bottom left minimum energy $(\mathrm{P3})$, bottom right initial guess given by (12).

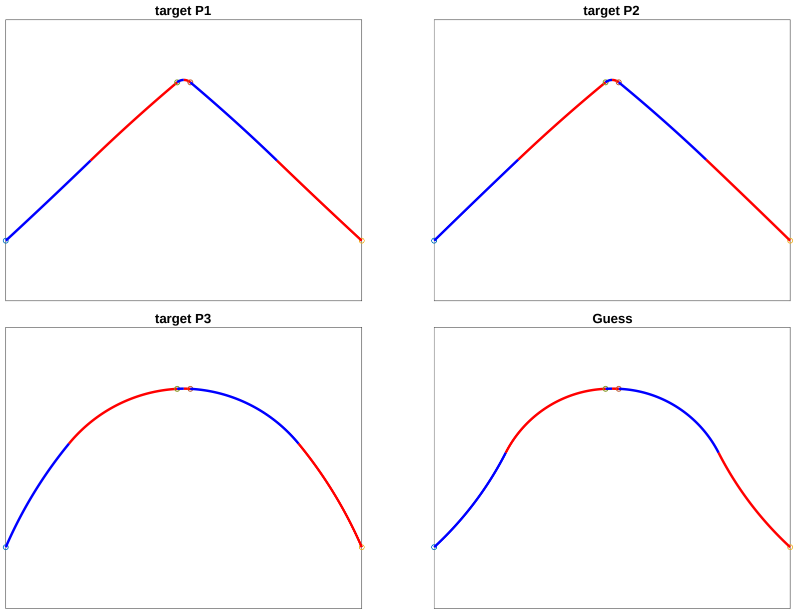

Figure 7: Test 4. Top left minimum length $(\mathrm{P1})$, top right integral of the absolute curvature $(\mathrm{P2})$, bottom left minimum energy $(\mathrm{P3})$, bottom right initial guess given by (12).

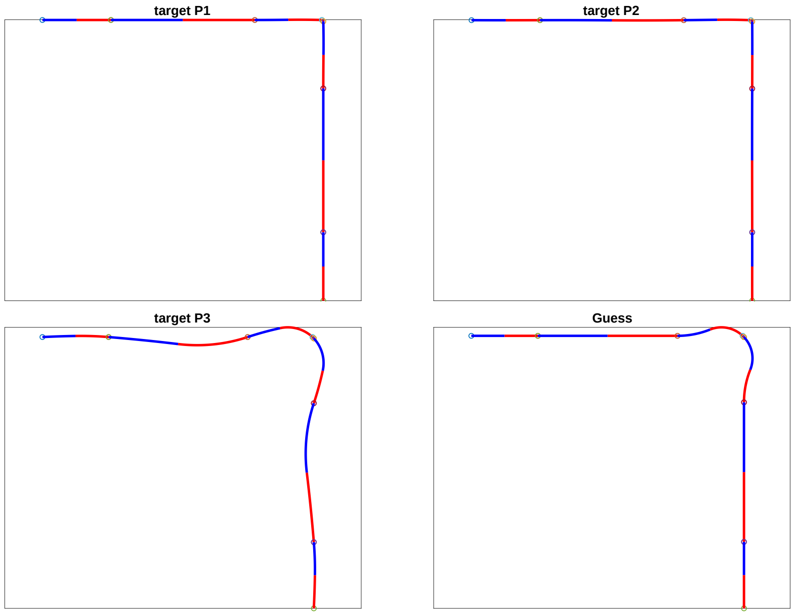

Figure 8: Test 5. Top left minimum length $(\mathrm{P1})$, top right integral of the absolute curvature $(\mathrm{P2})$, bottom left minimum energy $(\mathrm{P3})$, bottom right initial guess given by (12).

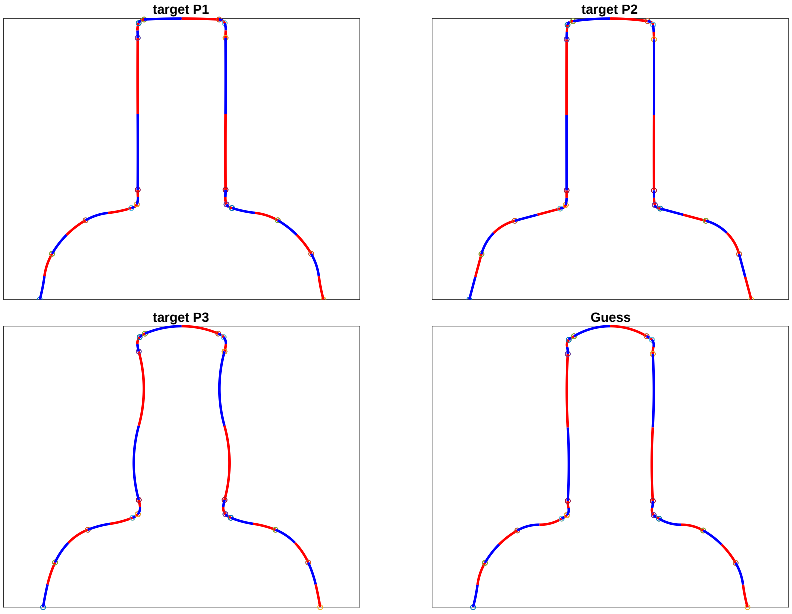

Figure 9: Test 6. Top left minimum length $(\mathrm{P1})$, top right integral of the absolute curvature $(\mathrm{P2})$, bottom left minimum energy $(\mathrm{P3})$, bottom right initial guess given by (12).

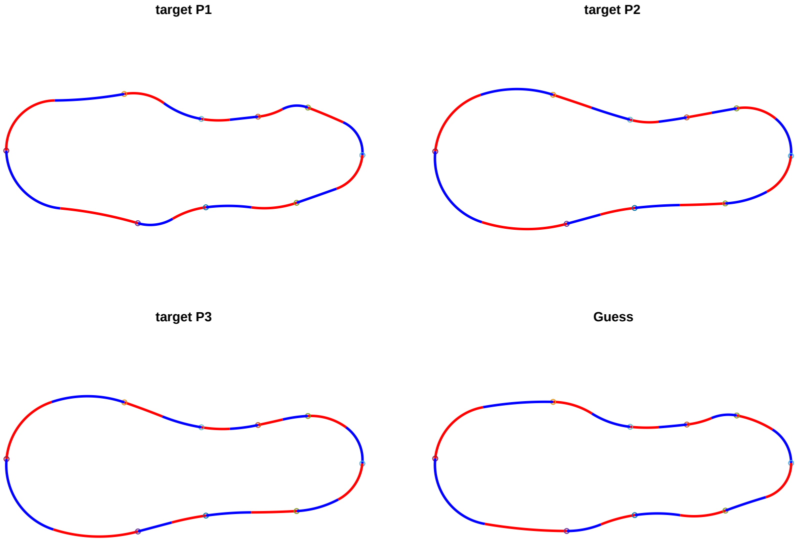

Figure 10: Test 7. Top left minimum length $(\mathrm{P1})$, top right integral of the absolute curvature $(\mathrm{P2})$, bottom left minimum energy $(\mathrm{P3})$, bottom right initial guess given by (12).

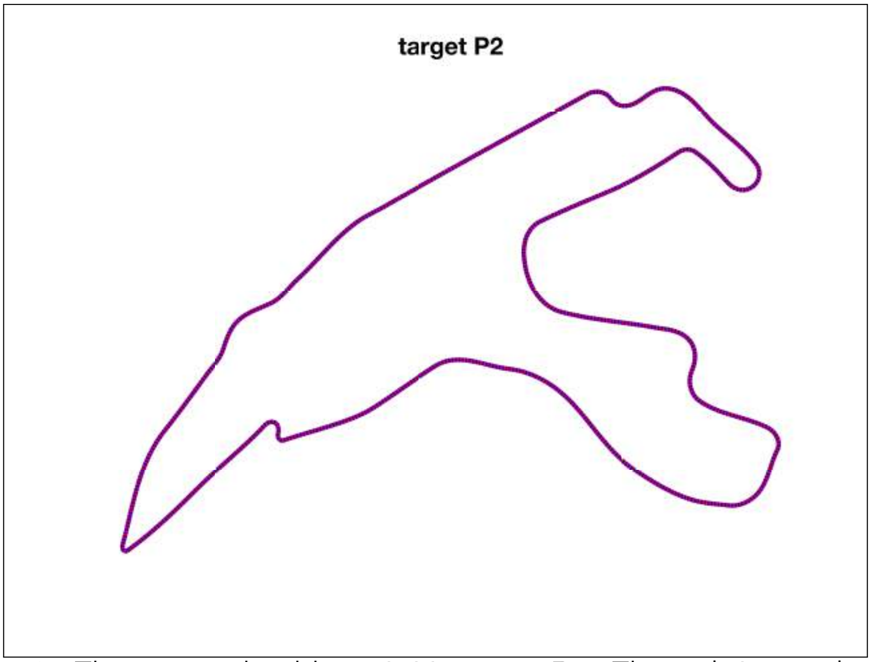

Figure 11: Test 8. The curve produced by optimising target $\mathrm{P2}$. The result is very close to the curves produced by the other targets.

Algorithm 1: Biarc solution algorithm

```
Biarc (x0, y0, ϑ0, x1, y1, ϑ1);
begin
```

$d_x \leftarrow x_1 - x_0$; $\quad d_y \leftarrow y_1 - y_0$;

$d \leftarrow \left(d_x^2 + d_y^2\right)^{1/2}$ $\quad \omega \leftarrow \mathrm{atan2}(d_y, d_x)$;

$\theta_0 \leftarrow \omega + \mathtt{Range}(\vartheta_0 - \omega)$; $\quad \theta_1 \leftarrow \omega + \mathtt{Range}(\vartheta_1 - \omega)$;

$t \leftarrow 2\cos\left(\frac{\theta_1-\theta_0}{4}\right)/d$ $\quad \theta_\star \leftarrow 2\omega - \dfrac{\theta_0+\theta_1}{2}$;

$\Delta\theta_0 \leftarrow \frac{\theta_\star-\theta_0}{2}$; $\quad \Delta\theta_1 \leftarrow \frac{\theta_\star-\theta_1}{2}$;

$\ell_0 \leftarrow 1/\left(t\,\mathrm{sinc}\left(\Delta\theta_0\right)\right)$; $\quad \ell_1 \leftarrow 1/\left(t\,\mathrm{sinc}\left(\Delta\theta_1\right)\right)$;

$\kappa_0 \leftarrow 2t\sin\left(\Delta\theta_0\right)$; $\quad \kappa_1 \leftarrow -2t\sin\left(\Delta\theta_1\right)$;

$x_\star \leftarrow x_0 + \cos\left(\frac{\theta_\star+\theta_0}{2}\right)/t$; $\quad y_\star \leftarrow y_0 + \sin\left(\frac{\theta_\star+\theta_0}{2}\right)/t$;

return *[*ℓ_0, θ_0, κ_0*]*, *[*ℓ_1, θ_1, κ_1*]*, $x_\star$, $y_\star$, $\theta_\star$;

end

`Sinc` (x) `// approximate` $(\sin x)/x$ `with error` $\leq 1.3 \cdot 10^{-20}$

begin

if $|x| < 0.002$ **then return** $1 + \dfrac{x^2}{6}\left(1 - \dfrac{x^2}{20}\right)$;

return $(\sin x)/x$

end

`Range` (θ) `// return` $\theta + 2k\pi$ `with` k `such that the angle is in` $[-\pi, -\pi]$

begin

while $\theta > +\pi$ **do** $\theta \leftarrow \theta - 2\pi$;

while $\theta < -\pi$ **do** $\theta \leftarrow \theta + 2\pi$;

return θ

end

References

[1] Ahlfors, L.: Complex analysis: an introduction to the theory of analytic functions of one complex variable. International series in pure and applied mathematics. McGraw-Hill, 1979. ISBN 9780070006577.

[2] Bertolazzi, E.; Bevilacqua, P.; Biral, F.; Fontanelli, D.; Frego, M.; Palopoli, L.: Efficient re-planning for robotic cars. In 2018 European Control Conference (ECC), 1068–1073, 2018. http://doi.org/10.23919/ECC.2018.8550215.

[3] Bertolazzi, E.; Bevilacqua, P.; Frego, M.: Clothoids: a C++ library with Matlab interface for the handling of clothoid curves. Rend. Sem. Mat. Univ. Pol. Torino, 76(2), 47–56, 2018. ISSN 0373-1243.

[4] Bertolazzi, E.; Frego, M.: G^1 fitting with clothoids. Mathematical Methods in the Applied Sciences, 38(5), 881–897, 2015. ISSN 1099-1476. http://doi.org/10.1002/mma.3114.

[5] Bertolazzi, E.; Frego, M.: Interpolating clothoid splines with curvature continuity. Mathematical Methods in the Applied Sciences, 41(4), 1723–1737, 2017. http://doi.org/10.1002/mma.4700.

[6] Bertolazzi, E.; Frego, M.: On the G^2 Hermite interpolation problem with clothoids. Journal of Computational and Applied Mathematics, 341, 99–116, 2018. ISSN 0377-0427. http://doi.org/10.1016/j.cam.2018.03.029.

[7] Bertolazzi, E.; Frego, M.: A note on robust biarc computation. Computer-Aided Design and Applications, 16(5), 822–835, 2019. http://doi.org/10.14733/cadaps.2019.822-835.

[8] Bertolazzi, E.; Frego, M.; Biral, F.: Point data reconstruction and smoothing using cubic splines and clusterization. Mathematics and Computers in Simulation, 2020. ISSN 0378-4754. http://doi.org/10.1016/j.matcom.2020.04.002.

[9] Bevilacqua, P.; Frego, M.; Fontanelli, D.; Palopoli, L.: A novel formalisation of the markov-dubins problem. Proceedings of the European Control Conference 2020, accepted(to appear), 2020.

[10] Bézier, P.: Numerical control: mathematics and applications. Wiley, 1970.

[11] Bolton, K.: Biarc curves. Computer-Aided Design, 7(2), 89–92, 1975.

[12] Brunnett, G.; Kiefer, J.: Interpolation with minimal-energy splines. Computer-Aided Design, 26(2), 137–144, 1994. ISSN 0010-4485. http://doi.org/10.1016/0010-4485(94)90034-5.

[13] Deng, C.; Ma, W.: A biarc based subdivision scheme for space curve interpolation. Computer Aided Geometric Design, 31(9), 656–673, 2014. ISSN 0167-8396. http://doi.org/10.1016/j.cagd.2014.07.003.

[14] Frego, M.; Bevilacqua, P.; Saccon, E.; Palopoli, L.; Fontanelli, D.: An iterative dynamic programming approach to the multipoint markov-dubins problem. IEEE Robotics and Automation Letters, 5(2), 2483–2490, 2020.

[15] Hobby, J.D.: Smooth, easy to compute interpolating splines. Discrete & Computational Geometry, 1(2), 123–140, 1986. ISSN 1432-0444. http://doi.org/10.1007/BF02187690.

[16] Kim, Y.J.; Lee, J.; Kim, M.S.; Elber, G.: Efficient offset trimming for planar rational curves using biarc trees. Computer Aided Geometric Design, 29(7), 555–564, 2012. ISSN 0167-8396. http://doi.org/10.1016/j.cagd.2012.03.014. Geometric Modeling and Processing 2012.

[17] Kovač, B.; Žagar, E.: Curvature approximation of circular arcs by low-degree parametric polynomials. Journal of Numerical Mathematics, 24(2), 95–104, 2016. http://doi.org/10.1515/jnma-2014-0046.

[18] Lee, E.: Choosing nodes in parametric curve interpolation. Computer-Aided Design, 21(6), 363 –370, 1989. ISSN 0010-4485. http://doi.org/10.1016/0010-4485(89)90003-1.

[19] Maier, G.: Optimal arc spline approximation. Computer Aided Geometric Design, 31(5), 211–226, 2014. ISSN 0167-8396. http://doi.org/10.1016/j.cagd.2014.02.011.

[20] Narayan, S.: Approximating cornu spirals by arc splines. Journal of Computational and Applied Mathematics, 255(C), 789–804, 2014. ISSN 0377-0427. http://doi.org/10.1016/j.cam.2013.06.038.

[21] Nutbourne, A.; Martin, R.: Differential Geometry Applied to Curve and Surface Design: Foundations. Differential Geometry Applied to Curve and Surface Design. E. Horwood, 1988. ISBN 9780470210369.

[22] Pagani, L.; Scott, P.J.: Curvature based sampling of curves and surfaces. Computer Aided Geometric Design, 59, 32–48, 2018. ISSN 0167-8396. http://doi.org/j.cagd.2017.11.004.

[23] Park, H.: Optimal single biarc fitting and its applications. Computer-Aided Design and Applications, 1(4), 187–195, 2004. http://doi.org/10.1080/16864360.2004.10738258.

[24] Parkinson, D.: Optimised biarc curves with tension. Computer Aided Geometric Design, 9(3), 207–218, 1992. ISSN 0167-8396. http://doi.org/10.1016/0167-8396(92)90018-K.

[25] Parkinson, D.; Moreton, D.: Optimal biarc-curve fitting. Computer-Aided Design, 23(6), 411–419, 1991. ISSN 0010-4485. http://doi.org/10.1016/0010-4485(91)90009-L.

[26] Piegl, L.; Tiller, W.: Data approximation using biarcs. Engineering with Computers, 18(1), 59–65, 2002. ISSN 1435-5663. http://doi.org/10.1007/s003660200005.

[27] Piegl, L.A.; Rajab, K.; Smarodzinava, V.; Valavanis, K.P.: Using a biarc filter to compute curvature

extremes of nurbs curves. Engineering with Computers, 25(4), 379, 2009. ISSN 1435-5663. http://doi.org/10.1007/s00366-009-0131-8.

[28] Piegl, L.A.; Tiller, W.: Biarc approximation of nurbs curves. Computer-Aided Design, 34(11), 807–814, 2002. ISSN 0010-4485. http://doi.org/10.1016/S0010-4485(01)00160-9.

[29] Rajab, K.; Piegl, L.: A knowledge-guided approach to line nurbs curve intersection. Computer-Aided Design and Applications, 11, 1–9, 2013. http://doi.org/10.1080/16864360.2013.829966.

[30] Sabin, M.: The Use of Piecewise Forms for the Numerical Representation of Shape. Computer & Automation Institute, Hungarian Academy of Sciences, 1976. ISBN 9789633110355.

[31] Saheya, B.; Yu, C.H.; Chen, J.S.: Numerical comparisons based on four smoothing functions for absolute value equation. Journal of Applied Mathematics and Computing, 56(1), 131–149, 2018. ISSN 1865-2085. http://doi.org/10.1007/s12190-016-1065-0.

[32] Song, X.; Aigner, M.; Chen, F.; Juttler, B.: Circular spline fitting using an evolution process. Journal of Computational and Applied Mathematics, 231(1), 423 – 433, 2009. ISSN 0377-0427. http://doi.org/10.1016/j.cam.2009.03.002.

[33] Yang, X.; Chen, Z.C.: A practicable approach to G1 biarc approximations for making accurate, smooth and non-gouged profile features in CNC contouring. Computer Aided Geometric Design, 38(11), 1205–1213, 2006. ISSN 0010-4485. http://doi.org/10.1016/j.cad.2006.07.006.

[34] Yang, X.; Wang, G.: Planar point set fairing and fitting by arc splines. Computer-Aided Design, 33(1), 35–43, 2001. ISSN 0010-4485. http://doi.org/10.1016/S0010-4485(00)00059-2.