Title:
Planar Curves based on Explicit Bézier Curvature Functions

Authors:
Norimasa Yoshida, norimasa@acm.org, Nihon University
Takafumi Saito, txsaito@cc.tuat.ac.jp, Tokyo University of Agriculture and Technology

Keywords:
Aesthetic Curves, Curvature Functions, Explicit Bézier Curves

DOI: 10.14733/cadconfP.2019.323-327

Introduction:

For controlling curvature variation as much as possible with given G^1 or G^2 Hermite conditions, we propose intrinsically defined planar curves based on explicit Bézier curvature functions. In the proposed curve, the curvature variation is specified by an explicit Bézier curve. To perform G^1 or G^2 Hermite interpolation, some of *control curvatures* of the explicit Bézier curve is modified to fit the given condition. The designer can directly change the curvature variation of the curve by modifying some of control curvatures satisfying given G^1 or G^2 Hermite conditions. We clarify how the viable regions for G^2 Hermite interpolation changes depending on the degree of explicit polynomial Bézier curves. We have implemented the method in C++ and confirmed that curves can be generated fully interactively. Applications of the proposed curves include the design of aesthetic curves for aesthetic surfaces as well as 2D illustrations.

Related Work:

Freeform curves, such as Bézier curves and NURBS curves, are widely used in many applications including CAD system. In freeform curves, the curve shape in terms of a polynomial or rational form is determined by control points, either explicitly or implicitly. We are able to know the +

curvature variation after the curve shape is completely determined by computing the derivatives. In this research, we represent the curvature function in terms of an explicit (polynomial or rational) Bézier curves and the curve is generated by integrating the curvature function. Our approach is most closely related to [4], but our approach is different in that we use explicit Bézier curves and clarify some characteristics including experimental viable regions. Our approach is more efficient than [4] because we show that arc length can be determined by G^1 or G^2 Hermite conditions. Therefore, arc length is not included in optimization parameters. Wu et al.'s work [5], where the curvature radius function is represented by cubic polynomials, is also related. Although no numerical integration is required in their approach, inflection points cannot be represented. Numerical integration is required once to generate a curve in our approach, we confirmed that the curve can be generated fully interactively. Log-aesthetic curves [6] are high quality curves whose curvature functions are relatively simple functions with monotonically varying curvature. However, when performing G^2 Hermite interpolation, they cannot match a wide variety of G^2 Hermite conditions. This work can be considered as a generalization of **curvature functions** in terms of explicit polynomial and rational Bézier curves.

Curves based on Explicit Bézier Curvature Functions:

Let s_t be the length of a curve segment. Let n be the degree of an explicit Bézier curve and $\kappa_i (i = 0,1,\cdots,n)$ be the *control curvatures*. An explicit polynomial Bézier curvature function $\kappa(s)$ in terms of arc length s is defined by

$$\kappa(s) = s_t\, K(\tau)\ \ (s \in [0, s_t]),\quad K(\tau) = \sum_{i=0}^{n} B_i^n(\tau)\ \kappa_i\ \ (\tau \in [0,1]), \tag{1}$$

where $B_i^n(\tau)$ is the Bernstein polynomial. An explicit rational Bézier curvature function $\kappa_R(s)$ is given by

$$\kappa_R(s) = s_t\, K_R(\tau)\ \ (s \in [0, s_t]),\quad \mathrm{K}_R(\tau) = \frac{\sum_{i=0}^{n} B_i^n(\tau)\ w_i \kappa_i}{\sum_{i=0}^{n} B_i^n(\tau)\ w_i}\ \ (\tau \in [0,1]). \tag{2}$$

We use $\kappa_G(s)$ to mean either $\kappa(s)$ or $\kappa_R(s)$. We also use $\mathrm{K}_G(\tau)$ to mean $K(\tau)$ or $\mathrm{K}_R(\tau)$. Tangential angle $\theta\,(s)$ can be computed by

$$\theta\,(s) = \int_0^s \kappa_G(t)\mathrm{d}t. \tag{3}$$

Note that the integration of Eqn. (3) can be computed in closed form if $\kappa_G(s)$ is an explicit polynomial Bézier curvature function. The curve position $\mathbf{P}(s)$ in the standard form, where $\mathbf{P}(s)$ is at the origin if $s = 0$ and its tangent vector is $[1\quad 0]^{\mathrm{T}}$, is computed by

$$\mathbf{P}(s) = \begin{bmatrix} \int_0^s \cos\theta\,(s)\mathrm{d}t \\ \int_0^s \sin\theta\,(s)\mathrm{d}t \end{bmatrix}. \tag{4}$$

The curve in general position can be obtained by performing an appropriate similarity transformation to curve generated by Eqn. (4).

$\boldsymbol{G^1}$ or $\boldsymbol{G^2}$ Hermite interpolation Method:

In G^1 Hermite Interpolation in the standard form, two points $\mathbf{P}_s, \mathbf{P}_e$ and their tangents $\mathbf{t}_s, \mathbf{t}_e$ are given. θ_d is the angle between $\mathbf{t}_s$ and $\mathbf{t}_e$. See Fig. 1. If all the control curvatures κ_i are given, the arc length of the curve segment s_t can be computed by

$$s_t = \frac{\theta_d}{\int_0^1 K_G(t)\mathrm{d}t}. \tag{5}$$

Note that in case if $K_G(t)$ is an explicit polynomial Bézier function, s_t can be simply computed by

$$s_t = \frac{\theta_d(n+1)}{\sum_{i=0}^{n} \kappa_i}. \tag{6}$$

If $K_G(t)$ is a rational function, numerical integration is required to compute s_t.

Since Eq. (4) is in standard form, the positional constraint $\mathbf{P}(0) = \mathbf{P}_s$ and the tangential constraint $\left.\frac{\mathrm{d}\mathbf{P}(s)}{\mathrm{d}s}\right|_{s=0} = \mathbf{t}_s$ at the start point $s = 0$ are automatically satisfied. The endpoint tangential condition $\left.\frac{\mathrm{d}\mathbf{P}(s)}{\mathrm{d}s}\right|_{s=s_t} = \mathbf{t}_e$ is satisfied by using arc length s_t computed using Eqn. (5). The remaining condition for satisfying G^1 Hermite condition is $\mathbf{P}(s_t)$ to be equal to $\mathbf{P}_e$. We satisfy this condition by an optimization using two of κ_i, typically κ_0 and κ_n, as optimization parameters. Other control curvatures are either user-specified or interpolated using κ_0 and κ_n. If we use a linear interpolation to compute $\kappa_1,\cdots,\kappa_{n-1}$, the generated curve will be the Clothoid curve.

In G^2 Hermite interpolation, curvature κ_s, κ_e of start and end points are specified in addition to G^1 Hermite interpolation conditions. G^2 Hermite interpolation is performed in a similar manner by setting $\kappa_0 = \kappa_s$ and $\kappa_n = \kappa_e$. κ_1 and κ_{n-1} are typically used as optimization parameters and other control curvatures are either user-specified or interpolated using κ_0 and κ_n. Note that any G^2 (and G^1) Hermite condition can be converted to the standard form shown in Fig. 1 by an appropriate similarly transformation. If two endpoints are uniformly scaled by a factor σ, the curvature at both endpoints must be scaled by a factor $\frac{1}{\sigma}$.

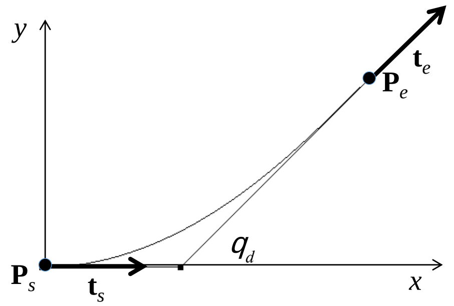


Fig. 1: G^1 Hermite Interpolation in the standard form.

Inflection Points and Curvature Monotonicity:

The curve generated by the proposed method may include an inflection point and the curvature may not to be monotonically varying. The existence of an inflection point can be checked by applying Bézier clipping [3] to $\kappa_G(\tau)$ where $\tau \in [0,1]$. If the degree of $\kappa(\tau)$ is low, we can directly compute τ such that $\kappa(\tau) = 0$. If $\kappa_G(\tau)$ becomes 0 and $\tau \in [0,1]$, the curve includes an inflection point within the curve segment.

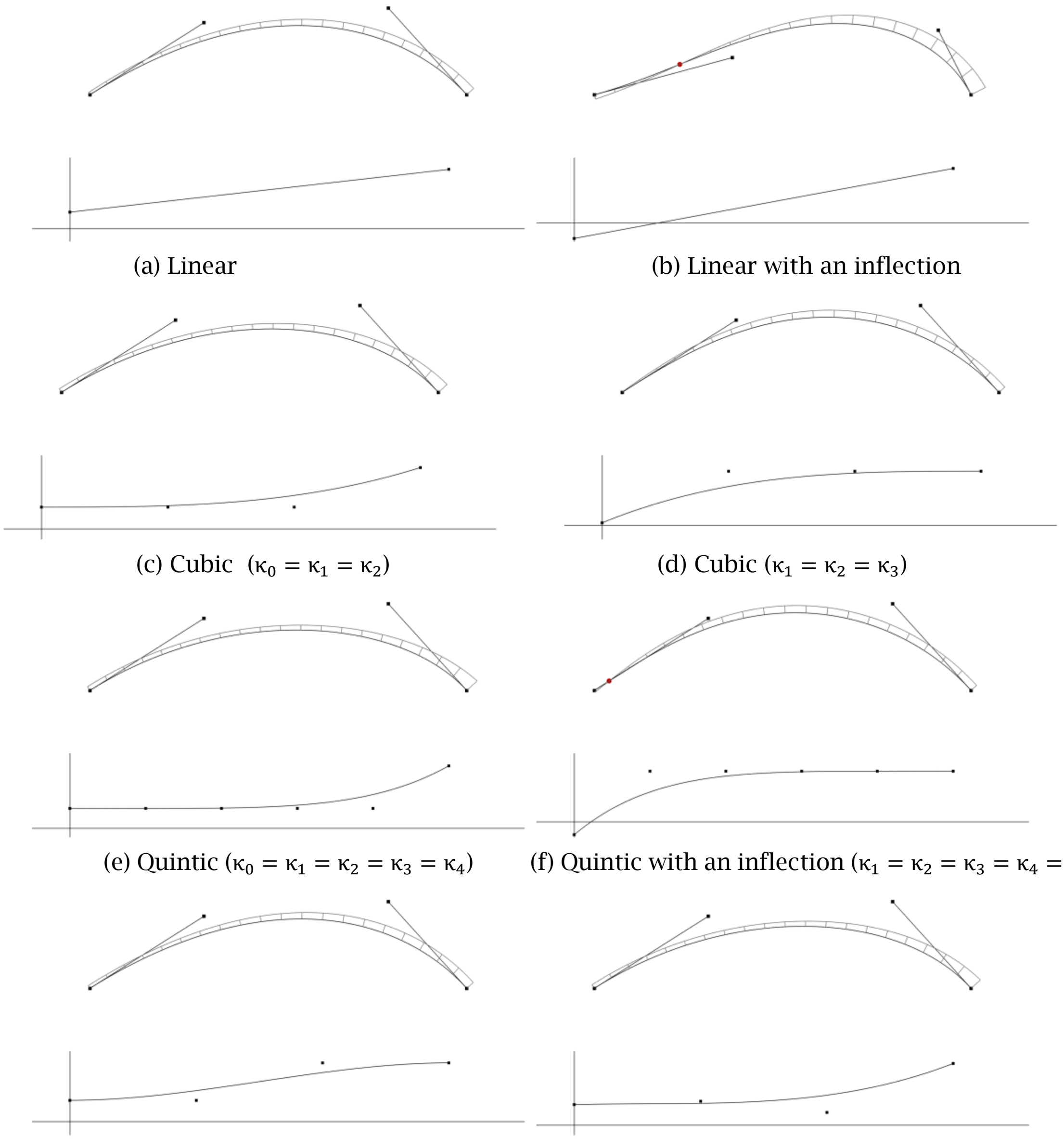
The curvature of the curve is monotonically varying if the first derivative of $\kappa_G(\tau)$ does not change its sign within $\tau \in [0,1]$. The monotonicity of curvature can be similarly checked by applying Bézier clipping to see if there is a sign change within $\tau \in [0,1]$. If there is a sign change, the curvature of the curves is not monotonically varying.

In case that an inflection point is not preferable, a user can move control points and/or control curvatures. Similarly, if a user wants the curvature to be monotonically varying and if the curve is not, the user can move control points and or control curvatures so that the curvature to be monotonically varying. In case of using a rational curvature function, the user can also change the weights of the rational function.

Results:

Fig. 2 shows various planar curves based on explicit polynomial Bézier curvature functions. Fig. 2 (a) is an example of linear curvature function. The generated curve is the Clothoid. Fig. 2 (b) is an example where the curve has an inflection point. Fig. 2 (c) and (d) are examples of using cubic Bézier curvature functions. The constraint of $\kappa_0 = \kappa_1 = \kappa_2$ is used in (c) , whereas in (d) the constraint of $\kappa_1 = \kappa_2 = \kappa_3$ is used. Fig. 2 (e) and (f) are similar examples but quantic Bézier curvature functions are used. The constraint of $\kappa_0 = \kappa_1 = \kappa_2 = \kappa_3 = \kappa_4$ is used in (e), whereas in (f) the constraint of $\kappa_1 = \kappa_2 = \kappa_3 = \kappa_4 = \kappa_5$ is used. Fig. 2 (g) is an example of using cubic Bézier curvature function where the constraint of $\kappa_0 = \kappa_1, \kappa_2 = \kappa_3$ is given. Thus in (g), $\frac{d\kappa}{ds} = 0$ at both endpoints. Harada et al. have pointed out that in many aesthetically pleasing connection between curve segments, the first derivative of curvature become 0 [2]. Thus the example of (g) may be important for aesthetically pleasing connection between segments. Note that in Fig. 2 (a), (c-g), the same G^1 Hermite condition is used. See various kinds of curvature variation can be generated for the same G^1 Hermite condition. Fig. 2 (h) shows an example of G^2 Hermite interpolation using cubic curvature function.

For G^2 Hermite interpolation, there is no guarantee that the curvature of generated curve is monotonically varying as shown in Fig. 3 (a). As shown in Fig. 3 (b) by appropriately modifying weights, we can generate a curve with monotonically varying curvature with the same G^2 Hermite condition. Thus by using rational functions, the generate curves can much more variety of G^2 conditions than using polynomial functions if the degree is the same.



(a) Linear

(b) Linear with an inflection

(c) Cubic ($\kappa_0 = \kappa_1 = \kappa_2$)

(d) Cubic ($\kappa_1 = \kappa_2 = \kappa_3$)

(e) Quintic ($\kappa_0 = \kappa_1 = \kappa_2 = \kappa_3 = \kappa_4$)

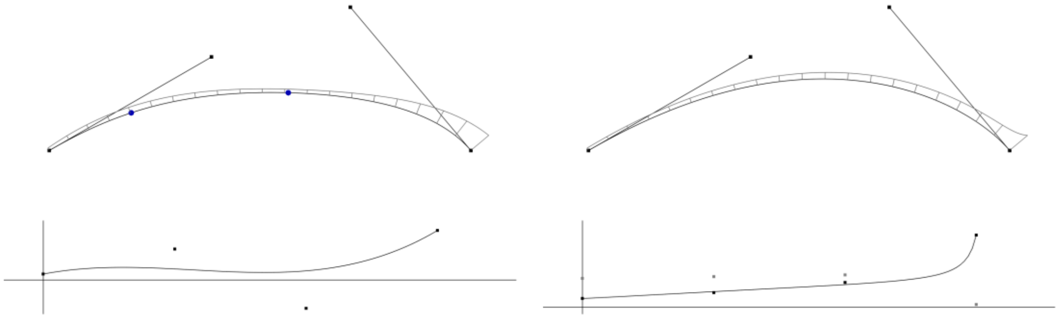
(f) Quintic with an inflection ($\kappa_1 = \kappa_2 = \kappa_3 = \kappa_4 = \kappa_5$)

(g) Cubic ($\kappa_0 = \kappa_1, \kappa_2 = \kappa_3$)

(h) Cubic, G^2 Hermite ($\kappa_s = 0.5, \kappa_e = 1.5$)

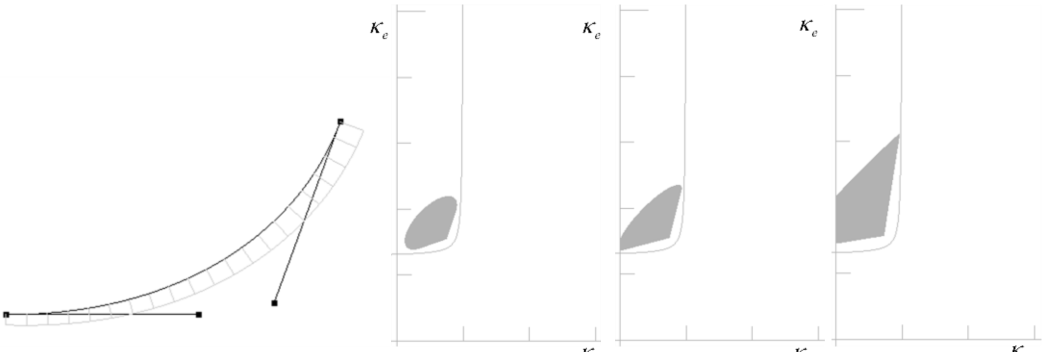
Fig. 2: Generated curves based on explicit Bézier curvature functions.

For curves based on explicit polynomial Bézier curvature functions of degree 3, 5 and 10, Fig. 4 (b), (c), (d) shows experimentally generated G^2 Hermite region of κ_s, κ_e where curves with monotonically varying curvature can be generated for the given G^1 Hermite condition shown in Fig. 4 (a). κ_1, κ_{n-1} are used as optimization parameters and $\kappa_{2,} \cdots, \kappa_{n-2}$ are linearly interpolated using κ_1 and κ_{n-1}. The hyperbolas shown in (b), (c), (d) shows the theoretically viable region where curves with monotonically varying curvature exists [1]. As the degree of the explicit Bézier curvature function gets higher, the viable region also gets larger.



(a) G^2 Hermite using polynomial Bézier (b) G^2 Hermite using rational Bézier

Fig. 3: G^2 Hermite Interpolation using polynomial and rational Bézier curves ($\kappa_s = 0.5, \kappa_s = 2.0$) .



(a) G^1 Hermite condition (b) $n = 3$ (c) $n = 5$ (d) $n = 10$

Fig. 4: G^2 Hermite region for curves based on explicit polynomial Bézier curvature functions.

Conclusions:

This paper proposed planar curves based on explicit polynomial or rational Bézier curvature functions. Future work includes clarifying G^2 Hermite regions for rational curvature functions.

Acknowledgements:

The authors are grateful for the funding provided by JSPS KAKENHI grant (No. 17K00172 and 16H02824).

References:

[1] Guggenheimer, H. W.: Differential Geometry, Dover, 1997.

[2] Harada, T.; Kono, M.: Reconstruction of rough sketch by connecting visual languages aesthetically, Journal of the Science of Design, 55(5), 55-64, 2008 (in Japanese). https://doi.org/10.11247/jssdj.55.5_55

[3] Sederberg, T. W.; Nishita, T.: Curve intersection using Bézier clipping, Computer-Aided Design, 22(9), 538-549, 1990. https://doi.org/10.1016/0010-4485(90)90039-F

[4] Watanabe, Y.; Saito, T.; Kuroda, M.: A method for generation of curves from specified curvature profile (in japanese), in: Graphics and CAD Work-shop of Information Processing Society of Japan, 7-12, 1997.

[5] Wu, W.; Yang, X.: Geometric Hermite interpolation by a family of intrinsically defined planar curves. Computer Aided Design, 77, 86-97, 2016. https://doi.org/10.1016/j.cad.2016.04.001

[6] Yoshida, N.; Saito. T.: Interactive aesthetic curve segments, Visual Computer, 22(9–11), 896–905, Aug. 2006. https://doi.org/10.1007/s00371-006-0076-5