IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 3, 2019, accepted December 17, 2019, date of publication December 23, 2019, date of current version January 2, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2961412

Dynamic Centripetal Parameterization Method for B-Spline Curve Interpolation

CENGİZ BALTA[1], SITKI ÖZTÜRK[1], MELIH KUNCAN[2], AND ISMET KANDILLI[3]
[1]Department of Electronics and Communications Engineering, Kocaeli University, Kocaeli 41380, Turkey
[2]Department of Electrical and Electronics Engineering, Siirt University Kezer Campus, Siirt 56100, Turkey
[3]Department of Electronics and Automation, Kocaeli University, Kocaeli 41500, Turkey

Corresponding author: Melih Kuncan (melihkuncan@siirt.edu.tr)

This work was supported by the Coordination Unit of the Scientific Research Projects of Kocaeli University (BAP) under Project 2015-018.

ABSTRACT B-spline data interpolation and approximation require parameterization at the first step. For this purpose, many algorithms have been developed, such as the uniform, centripetal, chord length, Foley and universal methods. The uniform method works well if the input data points are distributed regularly. The chord length method produces large deflections if long chords exist in the data polygon. To remove this effect, the centripetal method was developed. The traditional centripetal method uses a fixed power for chord lengths for parameter distribution. In this paper, we propose an improved version of the centripetal parameterization method for B-spline data interpolation. Our experiments show that individual dynamic power calculation can be possible for each chord length. This new parameterization method produces better behavior when compared to the traditional centripetal method and is more robust against fast changes in chord lengths since it uses the natural logarithm of chord lengths to calculate the parameters.

INDEX TERMS B-spline curves, parameterization, interpolation.

I. INTRODUCTION

B-splines are widely used to model curves and surfaces in computer-aided design (CAD). They provide smooth curves and surfaces and allow local control.

If a set of points is given, B-splines can be used to fit a curve to this given data set. Here, interpolation or approximation methods can be used. In this work, we focus on parameterization methods for global interpolation and attempt to improve the centripetal (exponential) method by converting it from a static exponential form to a dynamic exponential form using some chord properties [1]–[5].

The curve that is produced from given data points using interpolation is affected by the selected parameterization method [6]–[8]. The uniform (equidistant), chord length, centripetal and Foley methods are widely used to find proper parameters from a given data set [9]. Most of them use the geometric properties of the data points, while some use the standard deviation of points and angles between consecutive data lines [10]–[14].

The centripetal method is used in CAD applications with a static exponent form. In this work, we propose a dynamic exponent form of the centripetal parameterization method.

The associate editor coordinating the review of this manuscript and approving it for publication was Qi Zhou.

Assume a set of points $\{D_k\}, k = 0, \ldots n$, are given; we want to interpolate these target data points with a pth degree B-spline curve $C(t)$ [15]–[18]. Let us define the number of knots as $m + 1$ and $m = n + p + 1$. If we assign a parameter value t_K for each target data point D_k and select a knot vector $U = \{u_0, \ldots, u_m\}$, it is possible to find necessary control points p_i and then to interpolate target data points by solving an $(n + 1) \times (n + 1)$-dimensional system of linear equations as given below [19], [20].

$$D_k = C(t_k) = \sum_{i=0}^{n} N_{i,p}(t_k) P_i \quad 0 \le k \le n \tag{1}$$

Here, $N_{i,p}$ represents the basis functions. The control points P_i are unknowns and need to be calculated. To calculate these unknowns, in the first step, we need to calculate the parameter values and knot vector using target data points represented by D_k that we want to interpolate. In this work, we use normalized parameters distributed in the range of (0,1) and clamped B-splines [19]. Furthermore, we use the averaging technique given below [19], [20], if applicable, to calculate the knot vector.

$$\begin{aligned} u_0 &= u_1 = \ldots u_p = 0 \\ u_{j+p} &= \frac{1}{p} \sum_{i=j}^{j+p-1} t_i \quad j = 1, 2, \ldots, n-p \\ u_{m-p} &= u_{m-p+1} = \ldots u_m = 1 \end{aligned} \tag{2}$$

The rest of the paper is organized as follows. Section 2 describes the most common parameterization methods. Section 3 introduces the proposed dynamic centripetal method. Section 4 shows our experimental results on multiple data sets. Finally, Section 5 presents our conclusions.

II. COMMON PARAMETERIZATION METHODS

In CAD applications, to interpolate more desirable B-spline curves on data points, many parameterization methods have been developed. The uniform, chord length, centripetal, universal and Foley methods are usually fast and sufficient to calculate these parameters. Here, we provide a short summary of each.

A. UNIFORM PARAMETERIZATION

This is the simplest method for calculating the parameter values. If the target data points are regularly distributed, this method is preferred. Assuming the use of normalized parameterization, the parameters are distributed in the range of (0,1). Given $n+1$ data points, this range is divided into n pieces linearly. Here, we set $t_0 = 0$ and $t_n = 1$. Other parameters are calculated as below [21].

$$t_i = \frac{i}{n} \quad 1 \leq i \leq n-1 \tag{3}$$

Although uniform parameterization is easy to calculate, it may produce unwanted results. If the data distribution is not linear, such as in a uniform parameter distribution, this parameterization method produces unwanted wiggles. To overcome this problem, some alternative methods have been developed.

B. CHORD LENGTH PARAMETERIZATION

If the data point distribution is not linear, chord length parameterization produces better results compared to those of uniform parameterization. Chord lengths are distances between consecutive data points.

Let us assume that $D_0, D_1, \ldots. D_n$ target data points are given. The distance between points D_{i-1} and D_i is indicated by $|D_i - D_{i-1}|$. Here, the total chord length between consecutive data points is defined as $\mathcal{L} = \sum_{i=1}^{n} |D_i - D_{i-1}|$. We set $t_0 = 0$ and $t_n = 1$. Other parameters are calculated as follows [21].

$$t_k = \frac{1}{\mathcal{L}} \sum_{i=1}^{k} |D_i - D_{i-1}| \tag{4}$$

C. CENTRIPETAL PARAMETERIZATION

This method was proposed by Lee [21]. Assume that $D_0, D_1, \ldots. D_n$ data points are given. The power factor is $e = 0.5$. The distance between two adjacent data points is defined by $|D_i - D_{i-1}|^e$. In this case, the length of the data polygon is $L = \sum_{i=1}^{n} |D_i - D_{i-1}|^e$. The first and last parameters are defined as $t_0 = 0$ and $t_n = 1$, respectively. The middle parameters are distributed in the range of (0, 1) according to the following equation.

$$t_k = \frac{1}{L} \sum_{i=1}^{k} |D_i - D_{i-1}|^e \tag{5}$$

D. UNIVERSAL METHOD

This method was proposed by Choong-Gyoo Lim in 1999 [22]. While other methods calculate the knot vector from the parameters, Lim starts with a uniformly distributed knot vector and takes maximums of the basis functions as the parameters. This method produces more natural-looking curves but may produce undesired wiggles.

Assume that we are given $n+1$ data points and use p^{th}-degree B-splines. Let us define $m = n+p+1$. In this case, the number of knots is $m+1$. The clamped knots are distributed according to the following equation [20].

$$\begin{aligned} u_0 &= u_1 = \ldots u_p = 0 \\ u_{p+i} &= \frac{i}{n-p+1} \quad i = 1, 2, \ldots, n-p \\ u_{m-p} &= u_{m-p+1} = \ldots u_m = 1 \end{aligned} \tag{6}$$

The locations on which the maximum value points of the basis functions are produced by this knot vector are chosen as the parameters.

E. FOLEY-NEILSON METHOD

In 1989, Foley and Neilson suggested a method based on the Nielson metric [23]. This method uses the angles between consecutive target data points. The chord lengths between consecutive data points are calculated using the Neilson metric.

In the case of two-dimensional data points, $m = 2$, the coefficient matrix Q, where $Q = \{q_{i,j}\}$, $i, j = 1, 2$, is defined as in 1999 by Lim.

$$\begin{aligned} q_{11} &= \frac{V_y}{g}, \quad q_{22} = \frac{V_x}{g}, \quad q_{12} = q_{21} = \frac{V_{xy}}{g} \\ g &= V_x V_y - (V_{xy})^2 \\ V_x &= \frac{\sum_{i=0}^{n} (x_i - \overline{x})^2}{n+1}, \quad V_y = \frac{\sum_{i=0}^{n} (y_i - \overline{y})^2}{n+1} \\ V_{xy} &= \frac{\sum_{i=0}^{n} (x_i - \overline{x})(y_i - \overline{y})}{n+1} \\ \overline{x} &= \frac{1}{n+1} \sum_{i=0}^{n} x_i, \quad \overline{y} = \frac{1}{n+1} \sum_{i=0}^{n} y_i \end{aligned} \tag{7}$$

In this case, the Neilson metric between points U and V can be determined from [22]

$$M[D](U, V) = \sqrt{(U-V)Q(U-V)^T} \tag{8}$$

Here, if we distribute the parameters based on the Neilson metric chord lengths, we obtain an affine-invariant parameter set.

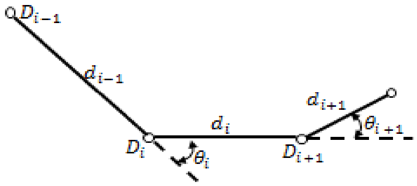

FIGURE 1. **Parameters for Foley method, from Hoschek's book [7].**

The Foley method also uses the angles between data lines. The parameter step Δt_i is defined as [9], [22], [23]

$$\begin{aligned}\Delta t_0 &= d_0\left[1+\frac{3\widehat{\theta}_1 d_1}{2(d_0+d_1)}\right]\\ \Delta t_i &= d_i\left[1+\frac{3\widehat{\theta}_i d_{i-1}}{2(d_{i-1}+d_i)}+\frac{3\widehat{\theta}_{i+1} d_{i+1}}{2(d_i+d_{i+1})}\right]\\ &\quad i=1,2....,n-2\\ \Delta t_{n-1} &= d_{n-1}\left[1+\frac{3\widehat{\theta}_{n-1} d_{n-2}}{2(d_{n-2}+d_{n-1})}\right] \end{aligned} \tag{9}$$

where $d_i = M[D](D_i, D_{i+1})$, $\widehat{\theta}_i = \min(\theta_i, \pi/2)$ and $\theta_i = \pi - \arccos\left[\frac{d_{i-1}^2+d_i^2+M^2[D](D_{i-1},D_{i+1})}{2d_i d_{i-1}}\right]$. Here, since θ_i is calculated using the Neilson metric, the angles are also affine-invariant. Fig. 1 shows the parameters used in the Foley method.

III. PROPOSED METHOD: DYNAMIC CENTRIPETAL PARAMETERIZATION

The centripetal method proposed by E.T.Y. Lee is widely used in CAD-CAM applications for B-spline data interpolation [21]. In the current design, this algorithm uses the same exponent for each chord between data points. Our experiments show that instead of using a static exponent, an individual dynamically calculated exponent related to the chord lengths between consecutive data points can be employed for the centripetal parameterization method. Distributing exponents between the maximum and minimum values with respect to the natural logarithm of the chord lengths improves the performance of the centripetal method.

Let us rearrange the centripetal parameterization with the addition of dynamic exponents. Assume that $D_0, D_1, \ldots .D_n$ data points are given and that the dynamic exponent is defined as e_i. In this case, the distance between data points D_{i-1} and D_i is calculated by $|D_i - D_{i-1}|^{e_i}$. Here, the total length of the data polygon for the proposed method is defined by

$$L = \sum_{i=1}^{n} |D_i - D_{i-1}|^{e_i} \tag{10}$$

In the case of normalized parameterization, the parameters are calculated from the following equations:

$$\begin{aligned} t_0 &= 0\\ t_k &= \frac{1}{L}\sum_{i=1}^{k} |D_i - D_{i-1}|^{e_i}\\ t_n &= 1 \end{aligned} \tag{11}$$

In the original centripetal method, the exponent value chosen is a constant, such as $e = 0.5$. Although centripetal parameterization generates better parameter distributions compared to those of the uniform and chord length methods, our experiments show that since the traditional centripetal method uses the same exponent value for long and short chords, it produces an irregular parameter distribution for a data set and needs some corrections. As a solution, we suggest an exponent distribution according to the natural logarithm of the chords. Assuming that $e < 1$ and all chord lengths are greater than 1, i.e., $|D_i - D_{i-1}| > 1, \ \forall i$, we distribute exponents between the maximum and minimum allowed values, inversely proportional to the natural logarithm of the chord lengths. For each step, the dynamic exponent value e_i that is valid for the related chord is calculated using the following definitions.

$$\begin{aligned} chord_i &: i^{th}\text{chord length, i.e., } |D_i - D_{i-1}|\\ chord_{\max} &: \text{the maximum chord length value}\\ chord_{\min} &: \text{the minimum chord length value (12)}\\ e_{\max} &: \text{maximum allowed value, default: 0.65}\\ e_{\min} &: \text{minimum allowed value, default: 0.35} \end{aligned} \tag{12}$$

In this case, the individual dynamic exponent value e_i is calculated using the equations below.

$$\frac{(e_{\max} - e_{\min})}{\log\left[\frac{chord_{\max}}{chord_{\min}}\right]} = \frac{(e_i - e_{\min})}{\log\left[\frac{chord_{\max}}{chord_i}\right]} \tag{13}$$

$$e_i = \frac{\log\left[\frac{chord_{\max}}{chord_i}\right]}{\log\left[\frac{chord_{\max}}{chord_{\min}}\right]}(e_{\max}-e_{\min})+e_{\min} \tag{14}$$

Using an individual exponent value e_i for each chord, we shorten the parameter increase for long chords in a balanced manner with other chords. This procedure produces a tightening effect for long chords. For short chords, a loosening effect is produced without disturbing the whole balance of the parameter distribution. In our experiments, we choose the following maximum and minimum exponent values: $e_{\min} = 0.35$, which is used for $chord_{\max}$, and $e_{\max} = 0.65$, which is used for $chord_{\min}$. The exponent values for other chords are distributed between $e_{\max}$ and $e_{\min}$, inversely proportional to the natural logarithm of the chord lengths. The natural logarithm of the chord lengths produces the far-sighted glasses effect [21].

IV. EXPERIMENTS ON MULTIPLE DATA SETS

Here, we use four different data sets to test our algorithm. The dynamic centripetal method usually outperforms the centripetal method, and it proves to be more powerful than the Foley method on four data sets.

As an error measurement criterion, the technique of measuring the distance of the chords connecting the data points is used in each step of the data polygon, as mentioned in the work of Fang and Hung [24]. Equation (15) shows the

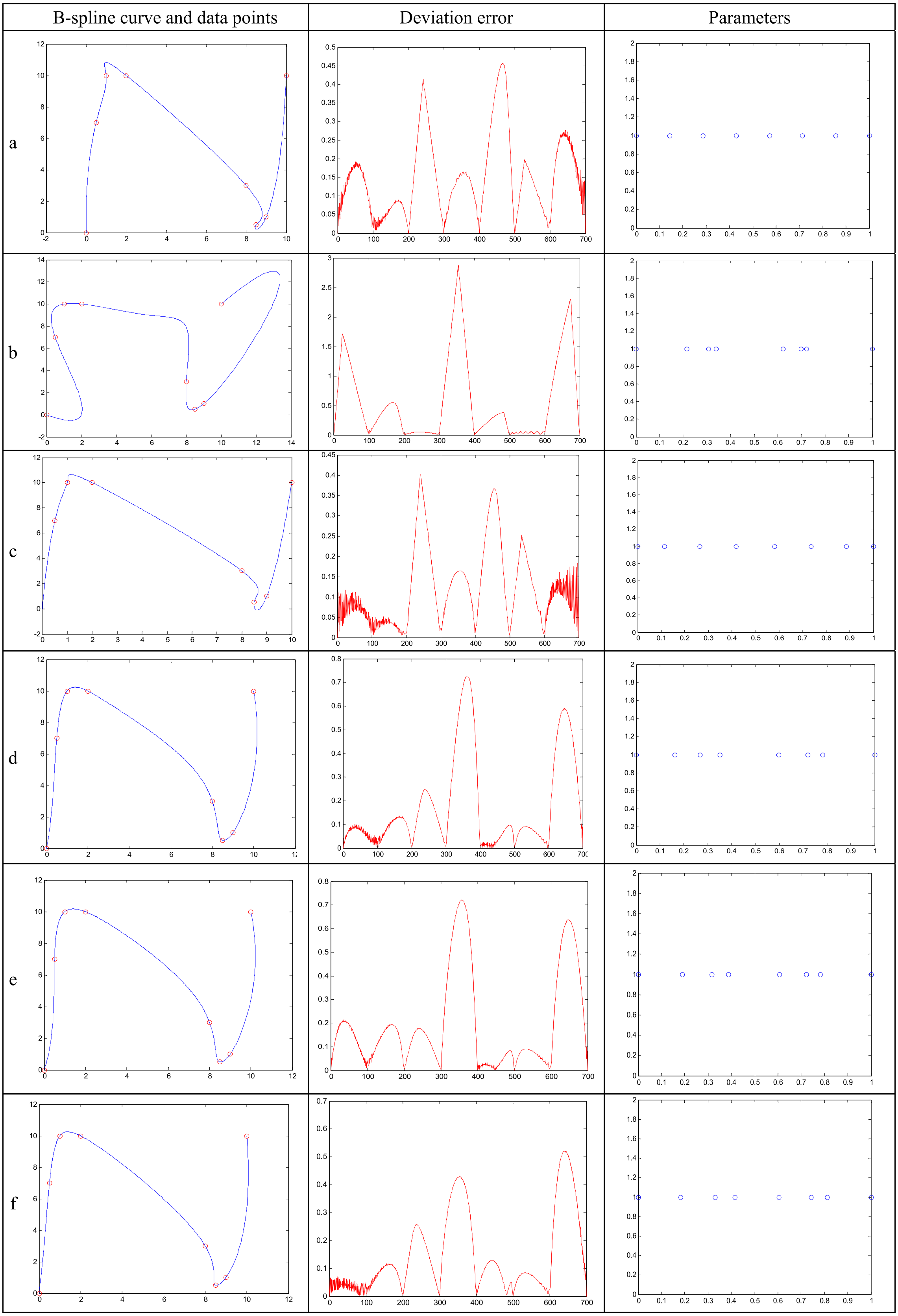

FIGURE 2. Cubic curve interpolation results for data set 1: (a) uniform, (b) chord, (c) universal, (d) Foley, (e) centripetal, and (f) proposed method.

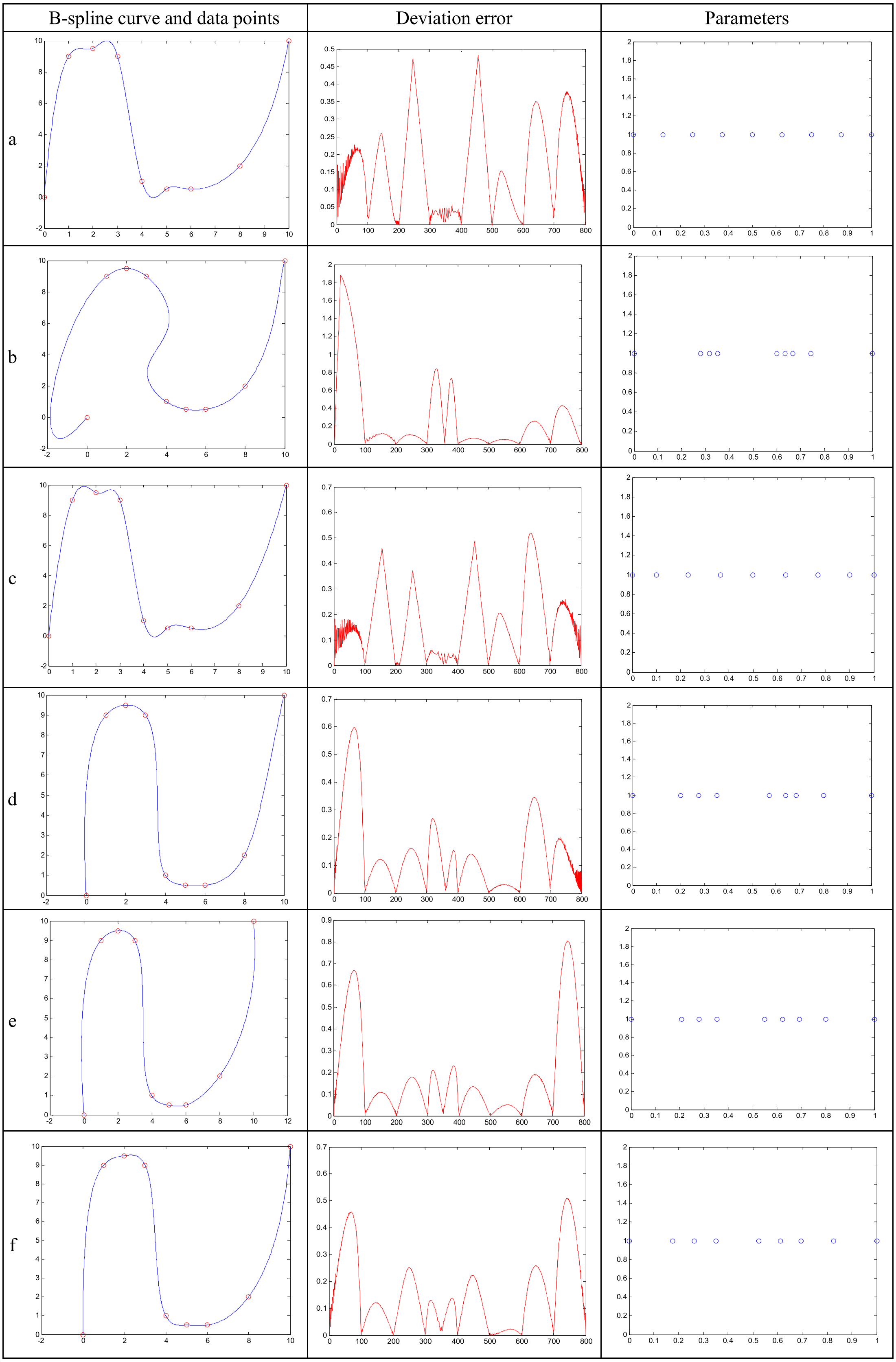

FIGURE 3. Cubic curve interpolation results for data set 2: (a) uniform, (b) chord, (c) universal, (d) Foley, (e) centripetal, and (f) proposed method.

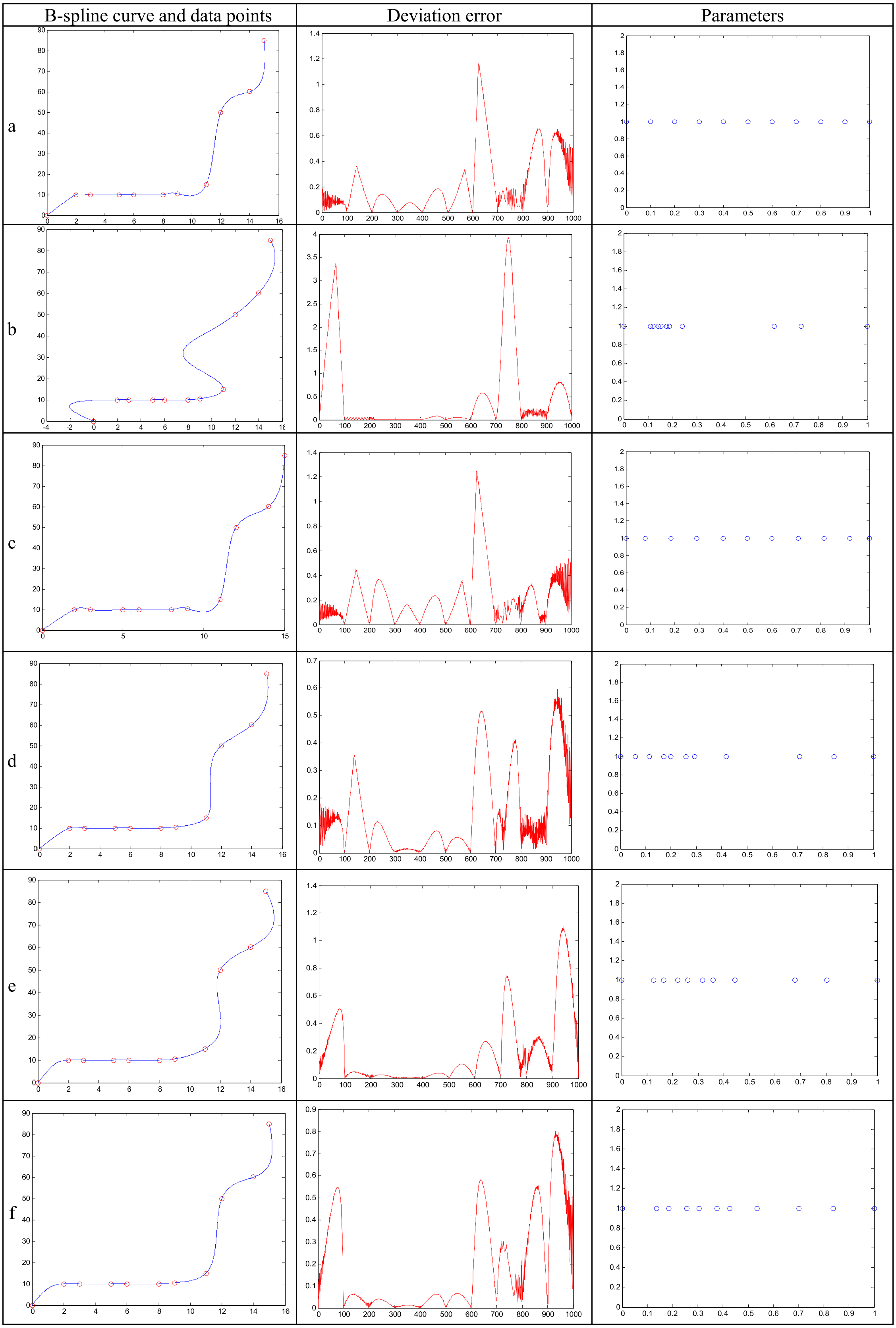

FIGURE 4. Cubic curve interpolation results for data set 3: (a) uniform, (b) chord, (c) universal, (d) Foley, (e) centripetal, and (f) proposed method.

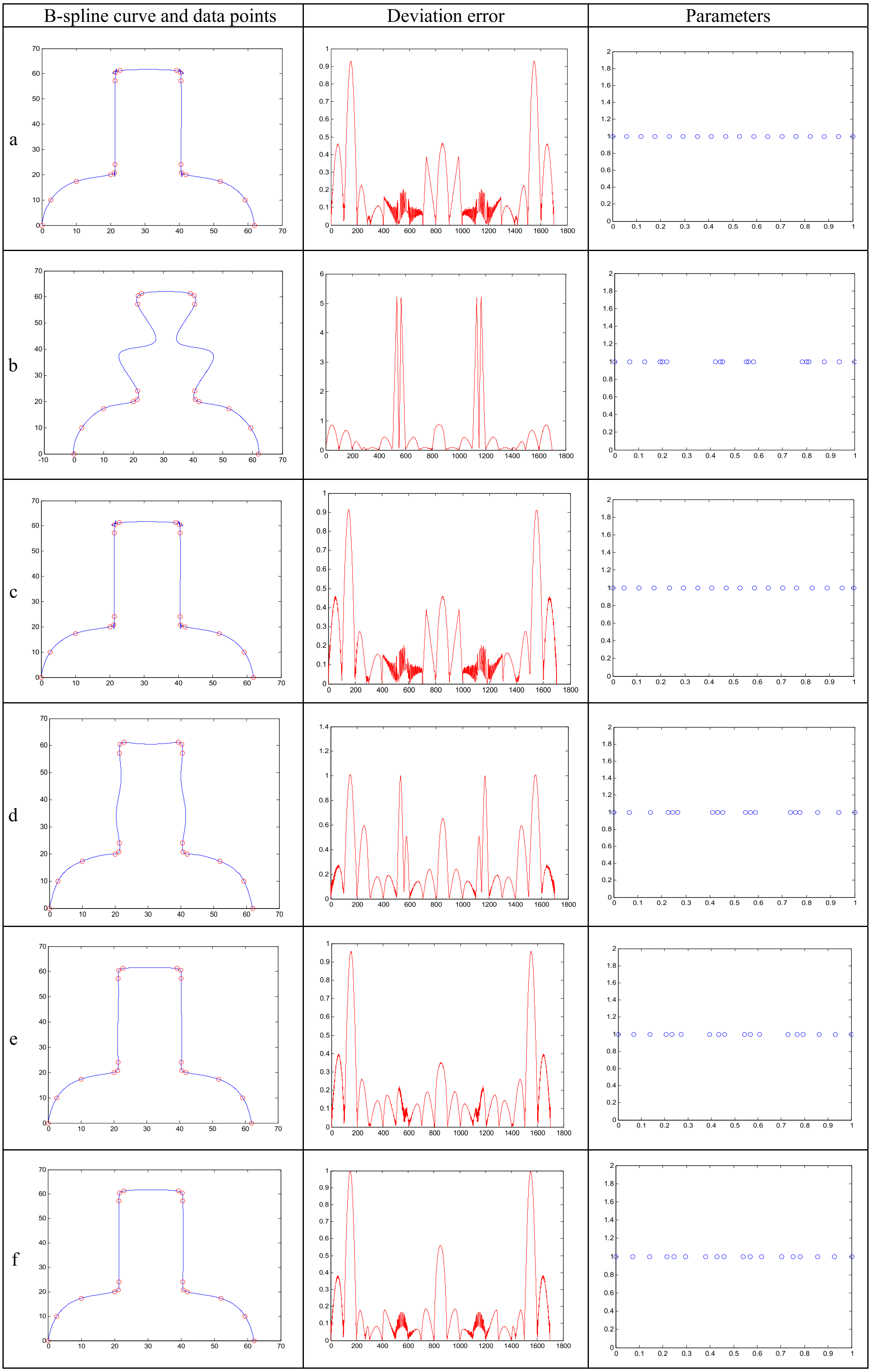

FIGURE 5. Cubic curve interpolation results for data set 4: (a) uniform, (b) chord, (c) universal, (d) Foley, (e) centripetal, and (f) proposed method.

TABLE 1. Maximum deviations from data segment for data set 1 in Figure 2.

Method	1	2	3	4	5	6	7	Average	Max Dev.
Uniform	0.1918	0.0889	0.4133	0.1647	0.4564	0.1964	0.2765	0.2550	0.4564
Chord	1.7113	0.5507	0.0524	2.8770	0.3785	0.0610	2.3040	1.1330	2.8770
Universal	0.1020	0.0470	0.4007	0.1642	0.3667	0.2513	0.1360	0.2097	0.4007
Foley	0.0988	0.1354	0.2475	0.7280	0.0976	0.0909	0.5904	0.2840	0.7280
Centripetal	0.2135	0.1952	0.1777	0.7214	0.0832	0.0906	0.6385	0.3028	0.7214
Proposed	0.0369	0.1147	0.2573	0.4283	0.1287	0.0840	0.5210	0.2244	0.5210

TABLE 2. Maximum deviations from data segment for data set 2 in Figure 3.

Method	1	2	3	4	5	6	7	8	Average	Max Dev.
Uniform	0.209	0.259	0.473	0.049	0.482	0.152	0.350	0.376	0.2938	0.4733
Chord	1.847	0.118	0.106	0.841	0.067	0.053	0.259	0.431	0.4652	1.8470
Universal	0.180	0.442	0.367	0.054	0.476	0.193	0.519	0.256	0.3109	0.5187
Foley	0.590	0.120	0.161	0.269	0.141	0.031	0.345	0.199	0.2319	0.5895
Centripetal	0.667	0.109	0.177	0.231	0.134	0.051	0.191	0.802	0.2952	0.8024
Proposed	0.460	0.121	0.251	0.140	0.223	0.023	0.258	0.509	0.2480	0.5094

TABLE 3. Maximum deviations from data segment for data set 3 in Figure 4.

Method	1	2	3	4	5	6	7	8	9	10	Average	Max. Dev.
Uniform	0.123	0.359	0.142	0.078	0.185	0.337	1.168	0.190	0.652	0.651	0.3885	1.168
Chord	3.347	0.057	0.011	0.008	0.082	0.053	0.583	3.915	0.238	0.806	0.9100	3.915
Universal	0.169	0.447	0.366	0.162	0.234	0.361	1.246	0.212	0.324	0.469	0.3990	1.246
Foley	0.164	0.354	0.111	0.131	0.079	0.057	0.516	0.412	0.130	0.597	0.2550	0.597
Centripetal	0.505	0.045	0.027	0.001	0.040	0.105	0.268	0.741	0.305	1.096	0.3132	1.096
Proposed	0.575	0.063	0.040	0.013	0.063	0.065	0.576	0.282	0.551	0.796	0.3022	0.796

mathematical expression of the error measurement.

$$D_i = \max_{\bar{u}_i < u < \bar{u}_{i+1}} dist\,[C(t),\ L_i] \tag{15}$$

Here, the value of the expression $dist\,[C(t),\ L_i]$ indicates the distance to the chord L at the instant t.

Figs. 2 − 5 show the cubic B-spline curves and error functions generated by the various methods for different data sets, as well as a comparison of the parameters. The deviation errors are shown in Tables 1 − 4. The first data set interpolated in Fig. 2 is taken from Irvine *et al.* [25]. This data set can be problematic for interpolation methods due to sudden directional changes and sparse data points. In the case of long-distance chords, a tension effect is produced on the curve by lowering the chord exponent e_i value dynamically. The curves produced by the uniform, chord length and universal methods seem to produce unwanted oscillations in the curves. In Table 1, the maximum deviations of the curves generated for the first data set from the data polygons are shown as error criteria. If we consider the average of the maximums, the proposed method produces the lowest error between successful curves. Although the mean error of the universal method does not seem low, it does not produce a successful curve. The mean error of the centripetal method has been succeeded by that of the Foley method with the dynamic power development of the proposed method.

In Fig. 3, the deviation error functions of the cubic B-spline curves and curve segments generated by the various parameterization methods for the second data set are shown together with the selected parameters. This second data set [25] was taken from the same work as that of the first data set. Similarly, sudden changes in direction and infrequent data difficulty exist here.

It is shown that the performances of the curve interpolation of the centripetal and the proposed methods are higher than those of the other methods. In Table 2, the maximum deviations of the curves generated for the second data set from the data polygons are shown as the error criteria. Taking into account the average error, the proposed method produces an error near the performance of the Foley method. The maximum deviation in the long chords is reduced by the proposed method. The third data set in Fig. 4 is taken

TABLE 4. Maximum deviations from data segment for data set 4 in Figure 5.

Method	Uniform	Chord	Universal	Foley	Centripetal	Proposed
1	0.4593	0.8607	0.4602	0.2704	0.3944	0.3685
2	0.9257	0.6698	0.9127	1.0110	0.9574	0.9965
3	0.2250	0.2848	0.2747	0.5958	0.2614	0.1802
4	0.1090	0.8897	0.1558	0.1798	0.1449	0.0820
5	0.1584	0.4455	0.1461	0.1909	0.1756	0.1794
6	0.1879	5.2240	0.2024	0.9990	0.2223	0.1116
7	0.0998	0.4396	0.1009	0.1437	0.1264	0.0649
8	0.3779	0.0840	0.3907	0.2422	0.1928	0.1859
9	0.4538	0.8721	0.4598	0.6513	0.3511	0.5599
10	0.3885	0.0839	0.3903	0.2422	0.1928	0.1859
11	0.0669	0.4396	0.0920	0.1441	0.1278	0.0666
12	0.1419	5.2240	0.2029	0.9916	0.2105	0.1313
13	0.1518	0.4433	0.1539	0.1937	0.1756	0.1794
14	0.1074	0.0775	0.1617	0.1655	0.1449	0.0823
15	0.2171	0.2919	0.2754	0.5958	0.2614	0.1797
16	0.9257	0.6717	0.9088	1.0110	0.9574	0.9965
17	0.4570	0.8607	0.4524	0.2782	0.3944	0.3798
Average	0.2830	1.0508	0.3377	0.4651	0.2718	0.2900
Max. Dev.	0.9257	5.2240	0.9127	1.0110	0.9574	0.9965

from Fritsch and Carlson [26]. Sudden distance changes are tested here. For this data set, the Foley and proposed methods produce better curve interpolation performances than do the other methods. Table 3 shows the maximum deviations of the curves produced by the third data set from the data polygon. With the proposed method, the mean deviation of the centripetal method is reduced, and unwanted oscillations in the long chords are decreased. The fourth data set in Fig. 5 was taken from Lee [21]. Here, there are more data points near the corners. Lee suggests that, in industry, corner points require more attention and should be represented with frequent data points. Table 4 shows the maximum deviations of the curves produced for the fourth data set from the data polygon. According to Table 4, the proposed method had a similar performance to that of the centripetal method. With the proposed method, the centripetal method improves, and in the case of long chords, the curve approaches the current data segment.

V. CONCLUSION

The centripetal parameterization method can be improved using the dynamic exponent presented in this work. Our experiments show that the centripetal method becomes more resistant to sudden chord length changes with the support of dynamic exponents.

Here, we use natural logarithms of the chord lengths of the data polygon, in an inverse proportion, to calculate the individual exponents of each chord. This method can be expanded to more artificial decision methods with the support of neural and fuzzy artificial intelligence algorithms. Hence, the calculation of better individual exponents can be achieved.

By using dynamic exponents, the centripetal methods become more robust against sudden changes in chord length. This makes the centripetal method more data-aware, and the method produces more desirable curves. In possible future works, this dynamic exponent method can be hybridized using angle aware methods. Furthermore, more intelligent dynamic exponent methods can be obtained by using artificial intelligence methods.

REFERENCES

[1] X. Dai and M. Bikdash, "Trend analysis of fragmented time series for mHealth apps: Hypothesis testing based adaptive spline filtering method with importance weighting," *IEEE Access*, vol. 5, pp. 27767–27776, 2017.

[2] M. Elbanhawi, M. Simic, and R. N. Jazar, "Continuous path smoothing for car-like robots using B-spline curves," *J. Intell. Robot. Syst.*, vol. 80, no. 1, pp. 23–56, Dec. 2015.

[3] M. Elbanhawi, M. Simic, and R. N. Jazar, "Randomized bidirectional B-spline parameterization motion planning," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 2, pp. 406–419, Feb. 2016.

[4] H. Ni, J. Yuan, S. Ji, C. Zhang, and T. Hu, "Feedrate scheduling of NURBS interpolation based on a novel jerk-continuous ACC/DEC algorithm," *IEEE Access*, vol. 6, pp. 66403–66417, 2018, doi: 10.1109/ACCESS.2018.2813334.

[5] B. Sun, H. Feng, K. Chen, and X. Zhu, "A deep learning framework of quantized compressed sensing for wireless neural recording," *IEEE Access*, vol. 4, pp. 5169–5178, 2016.

[6] X. D. Chen, W. Ma, G. Xu, and J. C. Paul, "Computing the Hausdorff distance between two B-spline curves," *Comput.-Aided Des.*, vol. 42, no. 12, pp. 1197–1206, Dec. 2010.

[7] S. I. Gofuku, S. Tamura, and T. Maekawa, "Point-tangent/point-normal B-spline curve interpolation by geometric algorithms," *Comput.-Aided Des.*, vol. 41, no. 6, pp. 412–422, Jun. 2009.

[8] L. Greco and M. Cuomo, "B-spline interpolation of Kirchhoff-Love space rods," *Comput. Methods Appl. Mech. Eng.*, vol. 256, pp. 251–269, Apr. 2013.

[9] J. Hoschek, D. Lasser, and L. L. Schumaker, *Fundamentals of Computer Aided Geometric Design*. Natick, MA, USA: A. K. Peters, 1993.

[10] A. Hashemian and S. F. Hosseini, "An integrated fitting and fairing approach for object reconstruction using smooth NURBS curves and surfaces," *Comput. Math. Appl.*, vol. 76, no. 7, pp. 1555–1575, Oct. 2018.

[11] L. Hu, H. Shou, and S. Fang, "A PIA optimization algorithm for non-uniform B-spline curves and surfaces," *Int. J. Model. Simul.*, vol. 37, no. 3, pp. 167–177, Apr. 2017.

[12] W. Zhong, X. Luo, W. Chang, F. Ding, and Y. Cai, "A real-time interpolator for parametric curves," *Int. J. Mach. Tools Manuf.*, vol. 125, pp. 133–145, Feb. 2018.

[13] B. Louhichi, G. N. Abenhaim, and A. S. Tahan, "CAD/CAE integration: Updating the CAD model after a FEM analysis," *The Int. J. Adv. Manuf. Technol.*, vol. 76, nos. 1–4, pp. 391–400, Jan. 2015.
[14] S. F. Hosseini, B. Moetakef-Imani, S. Hadidi-Moud, and B. Hassani, "The effect of parameterization on isogeometric analysis of free-form curved beams," *Acta Mechanica*, vol. 227, no. 7, pp. 1983–1998, Apr. 2016.
[15] L. P. Perez, F. Perez, and O. Durán, "Correction of meshless FPM interpolation sub-domains using genetic algorithms," *Stud. Inform. Control*, vol. 21, no. 2, pp. 191–200, Jun. 2012.
[16] T. Tjahjowidodo, V. T. Dung, and M. L. Han, "A fast non-uniform knots placement method for B-spline fitting," in *Proc. IEEE Int. Conf. Adv. Intell. Mechatronics (AIM)*, Busan, South Korea, Jul. 2015, pp. 1490–1495.
[17] E. Ülker and A. Arslan, "Automatic knot adjustment using an artificial immune system for B-spline curve approximation," *Inf. Sci.*, vol. 179, no. 10, pp. 1483–1494, Apr. 2009.
[18] X. Zhao, C. Zhang, B. Yang, and P. Li, "Adaptive knot placement using a GMM-based continuous optimization algorithm in B-spline curve approximation," *Comput.-Aided Des.*, vol. 43, no. 6, pp. 598–604, Jun. 2011.
[19] L. A. Piegl and W. Tiller, *The NURBS Book* (Monographs in Visual Communication Series). Berlin, Germany: Springer-Verlag, 1997.
[20] C. K. Shene, *Introduction to Computing With Geometry* (Lecture Notes). Houghton, MI, USA: Michigan Technological Univ., 2010. [Online]. Available: https://pages.mtu.edu/~shene/COURSES/cs3621/NOTES/
[21] E. T. Y. Lee, "Choosing nodes in parametric curve interpolation," *Comput.-Aided Des.*, vol. 21, no. 6, pp. 363–370, Jul./Aug. 1989.
[22] C. G. Lim, "A universal parametrization in B-spline curve and surface interpolation," *Comput. Aided Geometric Des.*, vol. 16, no. 5, pp. 407–422, Jun. 1999.
[23] T. A. Foley and G. M. Nielson, "Knot selection for parametric spline interpolation," in *Mathematical Methods in Computer Aided Geometric Design*. New York, NY, USA: Academic, 1989, pp. 261–264.
[24] J. J. Fang and C. L. Hung, "An improved parameterization method for B-spline curve and surface interpolation," *Computer-Aided Des.*, vol. 45, no. 6, pp. 1005–1028, Jan. 2013.
[25] L. D. Irvine, S. P. Marin, and P. W. Smith, "Constrained interpolation and smoothing," *Constructive Approximation*, vol. 2, no. 1, pp. 129–151, Dec. 1986.
[26] F. N. Fritsch and R. E. Carlson, "Monotone piecewise cubic interpolation," *SIAM J. Numer. Anal.*, vol. 17, no. 2, pp. 238–246, 1980.

CENGİZ BALTA received the B.S. degree in computer engineering from Bilkent University, Ankara, Turkey, in 1997, the M.S. degree in mechatronics engineering, and the Ph.D. degree in electronics and communications engineering from Kocaeli University, Kocaeli, Turkey, in 2006 and 2016, respectively.

SITKI ÖZTÜRK received the B.S. degree in electronics and communication engineering and the M.S. and Ph.D. degrees in electronics engineering from Yildiz Technical University, Istanbul, Turkey, in 1982, 1987, and 1995, respectively. He is currently an Assistant Professor with the Department of Electronics and Communication Engineering, Kocaeli University.

MELIH KUNCAN received the B.S., M.S., and Ph.D. degrees in mechatronics engineering from Kocaeli University, Kocaeli, Turkey, in 2010, 2013, and 2017, respectively. He is currently an Assistant Professor with the Department of Electrical and Electronics Engineering, Siirt University. His areas of research include control systems, instrumentation and measurement, signal processing, renewable energy, and power electronics.

ISMET KANDILLI received the B.S. degree in electronics and communication engineering from Yildiz Technical University, Istanbul, Turkey, in 1990, the M.S. degree in electronics engineering from the University of Istanbul, Istanbul, in 1995, and the Ph.D. degree in electronics engineering from Sakarya University, Sakarya, Turkey, in 2003. He is currently an Assistant Professor with the Department of Electronics and Automation, Kocaeli University. His areas of research include circuits and systems, instrumentation and measurement, signal processing, embedded systems, and tool condition monitoring.