A Study on Compact B-spline Curve Design using Dynamic Programming Approach

By

BD17103

Rachanart Soontornvorn

Supervisor

Prof. Dr. Hiroyuki Fujioka

A thesis submitted in partial fulfilment
of the requirements for the degree of
Doctor of Engineering

Intelligent Information System Engineering
Fukuoka Institute of Technology
Fukuoka, Japan

Acknowledgements

I would never have been able to finish my dissertation without the guidance of my advisor, help from friends, and support from my family.

Firstly, I would like to express my gratitude to my supervisor Professor Hiroyuki Fujioka of the Department of System Management Engineering at Fukuoka Institute of Technology. His expertise, understanding, and patience, added considerably to my graduate experience. I appreciate his vast knowledge and skill in many areas and for his continuous support of my doctor's degree study and related research. He provided me with an excellent atmosphere for doing research. His guidance helped me all the time with researching and writing this thesis. For six years from my master's degree until my doctor's degree, he always advised me not only with academic problems, but also whenever I had terrible situation. He always pushed me to be better. I am so proud to be the part of Fujioka Laboratory as his student.

I would like to thank Professor Takuya Tajima, Professor Yu Song from the Department of System Management Engineering and Professor Hitoshi Kino from the Department of Intelligent Mechanical Engineering for giving valuable advices to write this thesis. Also, I would like to thank Assistant Professor Vanvisa Chutchavong from Department of Computer Engineering, Faculty of Engineering at King Mongkut's Institute of Technology Ladkrabang Bangkok, Thailand, and Associate Professor Kanok Janchitrapongvej, Faculty of Science and Technology Southeast Bangkok College (SBC), Bangkok, Thailand, for giving me precious opportunities to discuss on this work. I would like to thank every member of Fujioka Laboratory. They always helped me and give me advices many times. My research would not have been possible without their help. Moreover, I would like to thank all of the staff of International Affairs at Fukuoka Institute of Technology. In particular, Ms. Fumie Yuki who teach Japanese language and always supported me in all of my Japanese life. Without her support, I could not have accomplished this thesis.

Finally, I would like to thank my family, Mr. Chupong Soontornvorn, Mrs. Krongkaeo Soontornvorn and Ms. Chayada Ukarawongchai for supporting me spiritually throughout writing this thesis and my life in general.

Rachanart Soontornvorn
Intelligent Information System Engineering
BD17103

Abstract

Fitting a curve to a set of data points is a key problem in many applications of science and engineering such as numerical analysis, robotics and image processing, etc. For solving such a curve-fitting problem, a natural approach is one with the so-called “spline function” which is a special function defined by piecewise polynomials. Such a spline approach enables us to yield the simplicity of their construction, their ease and accuracy of evaluation, and their capacity to approximate complex shapes through curve-fitting and interactive curve design. In particular, a natural choice from various types of splines is ‘B-splines’ developed by Schoenberg, in which a spline function has minimal support with respect to a given degree, smoothness, and domain partition.

This thesis considers a problem of designing curves using B-spline approach. Therein, the curves are constituted by employing the normalized, uniform B-splines as basis functions. That is, their knot points are equally-spaced. Then, a sequence of control points of B-splines is called as ‘control polygon’, which represents the geometrical outline of curves. Such a treatment on the control polygon is very powerful in the interactive curve design. With respect to the design using such a B-spline approach, we however see that the design usability may depend on the number of control points on the curves. For improving such a design usability, a natural way is to represent some given planar curves as more compact B-spline curves by using only the dominant control points, in which the desired approximation accuracy is achieved.

Main purpose of thesis is to develop a method for designing such compact B-spline curves by using only the dominant control points. In particular, such a method is developed for typical two types of B-spline curves, i.e., “planar B-spline curves” and “periodic B-spline curves”. Then, a central issue is how we optimally select a dominant control points. For solving such a problem, we here introduce an optimization approach using dynamic programming (DP) method. That is, the selection problem is formulated as a graph problem and is solved by dynamic programming. Thus, the method does not lead to huge amounts of computation time unlike the ordinary approaches - such as trial-and-error approach, etc. In addition, it is shown that representation using the selected control points can be realized using NURBS (Non-Uniform Rational B-splines). The proposed methods for the planar and periodic splines can be applied to the character design using the so-called dynamic font method and the contour modeling problem for the deformable objects, respectively. The performances are demonstrated by experimental studies.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Curve-fitting is the problem of constructing a curve (i.e., some mathematical function) that has the best fit to a series of data points, possibly subject to constraints. Such a curve fitting involves either interpolation or smoothing. Here, the former indicates that an exact fit to the data is required, On the other hand, the later indicates that a “smooth” function is constructed that approximately fits the data. The curve-fitting problem to a given set of data may arise in many applications of science and engineering such as statistics [1], [2], numerical analysis [3]-[5], image processing [6]-[8], robotics [9]-[12], etc., For example, in [13], the motion planning problem for mechanical systems has been treated as the curve-fitting problem. For solving such a curve-fitting problem, a natural approach is one with the so-called “spline function” which is a special function defined by piecewise polynomials. Such a spline approach enables us to yield the simplicity of their construction, their ease, and accuracy of evaluation, and their capacity to approximate complex shapes through curve fitting and interactive curve design. In particular, a natural choice from various types of splines is ‘B-splines’ developed by Schoenberg [14], and [15], in which a spline function has minimal support with respect to a given degree, smoothness, and domain partition. Fujioka and Kano have also studied the optimal design and properties of curves and surfaces by using the B-spline approach (e.g., [16]). Therein, they have analyzed the convergence properties of interpolating and smoothing splines as the number of data increases to the infinity [17]. Also, various types of splines - such as planar splines, periodic splines, and constrained splines, etc. have been developed. In this thesis, we restrict ourselves to the problem on the planar spline curves and periodic spline curves.

The planar spline curves are curves that can be represented in Cartesian coordinates by a parametric equation of the form $(x, y) = (x(t), y(t))$ for spline curves $x(t)$ and $y(t)$ [18] and [19]. For example, Bo's work in [20] has exhibited the fundamental planar curve-fitting problem to unorganized data points by introducing B-splines. As for its applications, the planar spline curves have been applied to the geometric modeling problem. Another typical example is described in Morgand's work [21], in which planar spline curves are used to predict the shape of specularities by introducing some planar B-splines. Similar work has been exhibited in Kim's work [22]. Therein, the planar spline curves have been used to plan the robotic motion in the geometric environment using some algebraic algorithms. Also, there is another application which we are very interested. The application is a design of character font based on the so-called "dynamic font" developed by Takayama and Kano in [23]. The dynamic font method has been developed by mimicking the writing process by humans, in which characters are generated by moving a writing device on a writing plane continuously in both time and space. This method is powerful, particularly when we want to generate and manipulate characters in Japanese calligraphy where the thickness of the stroke is important. Such a motion is generated by using normalized uniform B-spline as a basis function. The sequence of control points of B-splines is called as 'control polygon', which represents the geometrical outline of curves. Such a property leads many advantages to manipulate characters. For example, the operations on characters - such as resize, translation, rotation, and concatenation, etc. can be defined as operations on control polygon [24].

On the other hand, the periodic spline curves are curves that repeat its values in regular intervals or periods using splines [25]. A typical mathematical example of using such periodic spline curves may be to interpolate or approximate the trigonometric functions, which repeat over intervals of 2π radians. In addition, such periodic spline curves may be used throughout science to describe oscillations, waves, and other phenomena that exhibit periodicity [26]. Also, Fujioka and Kano's group has studied the optimal design of such periodic spline curves by using B-spline approach and control-theoretic approach and their result has been applied to the contour modeling of wet material objects - such as jellyfish and red blood cell, etc. [17]. Similar works on periodic spline curves have been appeared in the field of computer vision such as the idea of matching systems for contour the human face and head based on periodic curvature in [27].

This thesis considers the problem of designing curves using the B-spline approach. Therein, the curves are constituted by employing the normalized, uniform B-splines as basis functions. That is, their knot points are equally-spaced. Then, a sequence of control points of B-splines is called as ‘control polygon’, which represents the geometrical outline of curves. Such a treatment on the control polygon is very powerful in the curve design problem. As mentioned above, in the case of dynamic font using planar spline curves, manipulations on characters can be defined as some operations on the control polygon. Also, in the case of contour modeling using periodic spline curves, such a treatment yield an efficient algorithm to understand the deformation motion, etc. More specifically, the computation on the area and shape complexity can be given by simple functions in terms of control point vector (i.e., control polygon). With respect to the design using such a B-spline approach, we, however, see that the design usability depends on the number of control points on the curves. For improving such design usability and computation speed in the algorithm, a natural way is to represent a given spline-curves as more compact B-spline curves by using only the dominant control points, in which the desired approximation accuracy is achieved.

The main purpose of this thesis is to develop a method for designing such compact B-spline curves by using only the dominant control points. In particular, such a method is developed for typical two types of B-spline curves i.e., “planar B-spline curves” and “periodic B-spline curves”. Then, a central issue is how we optimally select dominant control points. For such an optimal dominant control point selection, the most typical approach may be the trial-and-error approach exhibited in Lyche and Mørkens work [28]. Therein, a larger number of control points is initially defined, and then, certain knot points are greedily removed by employing some heuristic criterion. After removing the knot points, recomputing the control points is required in order to achieve the desired approximation accuracy. Similar work is exhibited in Park and Lee’s work [29]. However, it is well-known that the trial-and-error approaches lead to huge amounts of computation time. On the other hand, Tjahjowidodo et al. [30] have recently developed a fast algorithm for optimally finding knot points using the so-called half-split method, where it is limited to the case of cubic splines. The basic idea is to approximate a subset of the second derivative sample points as a set of piecewise linear functions. This may be applicable to cases where we want to select dominant control points from original ones. But, it may be impossible to apply this method when we want to specify the number of dominant control points, as in the following discussion of this study.

For solving such a selection problem, we here introduce an optimization approach using a dynamic programming (DP) method. That is, the selection problem is formulated as a graph problem and is solved by dynamic programming. Thus, the method does not lead to huge amounts of computation time unlike the ordinary approaches - such as the trial-and-error approach [30]. Then, we have only to represent our compact B-spline curves. Since the intervals between their knot points may be non-uniform, representing dominant control point to our compact B-spline curves is achieved by using Non-Uniform Rational B-Splines (NURBS). The methods for the planar and periodic splines can be applied to the character design using the so-called dynamic font method [23] and the contour modeling problem for the deformable objects [31], respectively. The performances are demonstrated by experimental studies.

The outline of this thesis is as follows. In Chapter 1, we give the background and the purpose of this thesis as an introduction. In Chapter 2, we present the spline curve basics including B-spline as well as Non-Uniform B-splines (NURBS), which will be frequently used throughout this thesis. In Chapter 3, we present the dynamic programming including a graph theory. In Chapter 4, we develop the method for designing compact planer B-spline curves and conduct some experimental studies in order to demonstrate the usefulness and effectiveness of our proposed method. The similar work for the case of periodic splines are discussed in Chapter 5. In Chapter 6, concluding remarks of this thesis are given.

Chapter 2

Spline Curve Basics

2.1 Introduction

In this chapter, we present the spline curve basics. In Section 2.2, we present the power basis form of a curve, which is a common method for expressing polynomial functions. Then, the Bézier curves are introduced in Section 2.3. In Section 2.4 and 2.5, we present the B-spline curves and NURBS (Non-Uniform Rational B-spline), which will be used throughout in this thesis.

2.2 Power Basis Form of a Curves

We first present the power basis form for a curves, which is a common method for expressing polynomial functions.

Let $x(t)$ be n-th degree power basis curves, then we have

$$x(t) = \sum_{i=0}^{n} a_i t^i. \tag{2.1}$$

Also, this can be written in matrix form as

$$x(t) = [a_0 \, a_1 \, \cdots \, a_n] \begin{bmatrix} 1 \\ t \\ \vdots \\ t^i \end{bmatrix} = [a_i]^T[u^i]. \tag{2.2}$$

Differentiating (2.1) in terms of t may give as

$$a_i = \frac{x^i(t)|_{t=0}}{i!}, \tag{2.3}$$

where $x^i(t)|_{t=0}$ is the i-th derivative of $x(t)$ at $t = 0$. We here call the $n + 1$ function t's as the basis functions. Also, a_i's are called as coefficients of the power basis presentation.

It can be shown that the point $x(t)$ on power basis curves can be computed efficiently by introducing Horner's method as follows,

$$\begin{aligned} \text{for} \quad n = 1, &\quad x(t_0) = a_1 t_0 + a_0 \\ n = 2, &\quad x(t_0) = (a_2 t_0 + a_1)t_0 + a_0 \\ &\quad \vdots \\ n = \bar{n}, &\quad x(t_0) = ((\cdots(\bar{a}_n t_0 + \bar{a}_{n-1})t_0 + \bar{a}_{n-2})t_0 + \cdots + a_0 \end{aligned} \quad . \tag{2.4}$$

2.3 Bézier Curves

We next present Bézier curves, which are parametric polynomial curves. Since the power basis form uses polynomials for the representation of curves, thus the Bézier form is mathematically equivalent to the power basis form. That is, any curves that can be represented in the Bézier form can be represented in the power basis forms. However, the Bézier curves have been frequently used in some software - such as the modeling and drawing tools due to the drawbacks of the power basis forms (i.e., huge computational cost, etc.).

Now, let Bézier curves of $x(t)$ be the n-th degree is given by

$$x(t) = \sum_{i=0}^{n} p_i B_{i,n}(t) \qquad 0 \le t \le 1, \tag{2.5}$$

where $B_{i,n}(\cdot)$ denotes the n-th degree Bernstein polynomials given by

$$B_{i,n}(t) = \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i} \qquad t \in [0,1]. \tag{2.6}$$

The coefficients p_i are called "control points" that approximate the shape of the curve. Figures 2.1 and 2.2 illustrate design examples of quadratic and cubic Bézier curves, where we note that $x(t)$ is defined as $x(t) = [X(t)\ Y(t)]^T \in \mathbf{R}^2$.

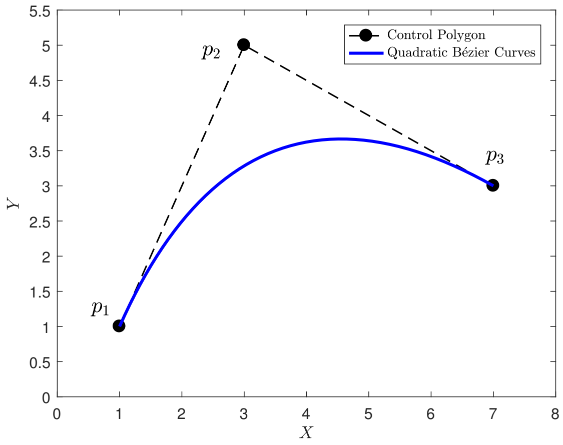


Figure 2.1: Quadratic Bézier Curves.

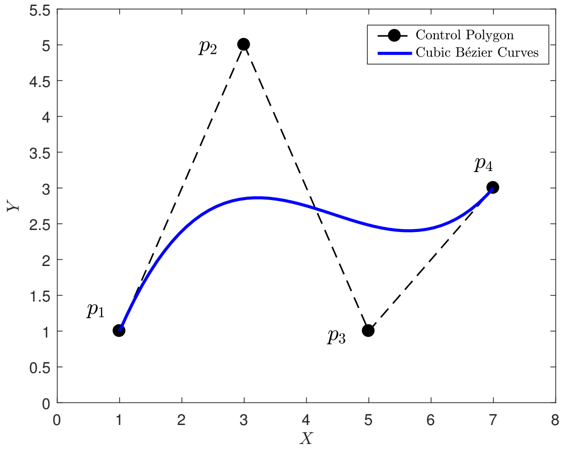


Figure 2.2: Cubic Bézier Curves.

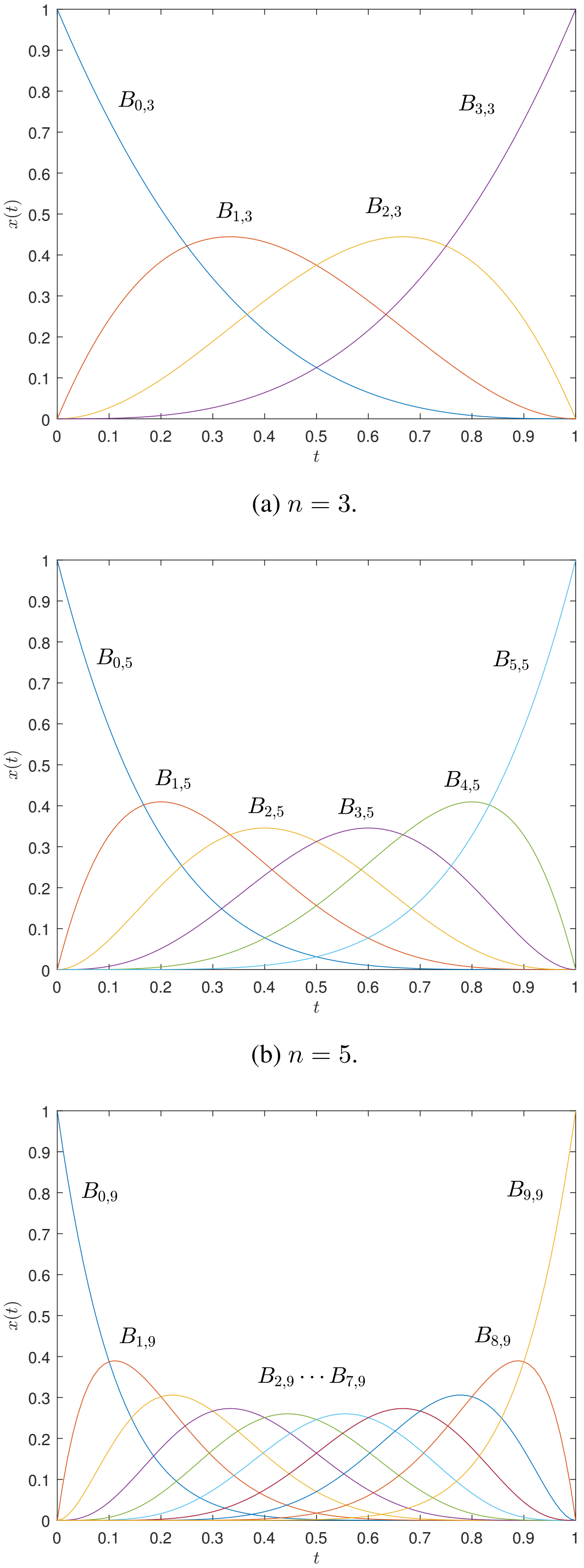
The Bézier curves are invariant under the usual transformations such as rotations and translations, and others by applies the transformation it to the control polygon. Normally, the choice of basis function determines the geometric characteristic of the scheme. The properties of basis function $B_{i,n}(t)$ degree n (i.e., degree $n = 3, 5$ and 9 in Figures 2.3) can be summarized as follows,

- Non-negativity : $B_{i,n}(t) \geq 0, \quad 0 \leq t \leq 1.$
- Partition of unity : $\sum_{i=0}^{n} B_{i,n}(t) = 1, \quad 0 \leq t \leq 1.$
- $B_{0,n}(0) = B_{n,n}(1) = 1.$
- $B_{i,n}(t)$ attains exactly one maximum on the interval [0,1], that is, at $t = \frac{i}{n}$.
- Symmetry : for any n, the set of polynomial $B_{i,n}(t)$ is symmetric with respect to $t = \frac{1}{2}$.
- Recursive definition : $B_{i,n}(t) = (1-t)B_{i,n-1}(t) + tB_{i-1,n-1}(t)$; we define $B_{i,n}(t) \equiv 0, \quad \text{if } i < 0 \text{ or } i > n.$
- Derivative : $B'_{i,n}(t) = \frac{dB_{i,n}(t)}{dt} = n(B_{i-1,n-1}(t) - B_{i,n-1}(t))$ with $B_{-1,n-1}(t) \equiv B_{n,n-1}(t) \equiv 0.$

For example, we can calculate $B_{i,n}(t)$ for the cubic case (i.e., $n = 3$) as

$$\begin{aligned}
B_{0,1}(t) &= (1-t)B_{0,0}(t) + tB_{-1,0}(t) = 1-t \\
B_{1,1}(t) &= (1-t)B_{1,0}(t) + tB_{0,0}(t) = t \\
B_{0,2}(t) &= (1-t)B_{0,1}(t) + tB_{-1,1}(t) = (1-t)^2 \\
B_{1,2}(t) &= (1-t)B_{1,1}(t) + tB_{0,1}(t) = 2t(1-t) \\
B_{2,2}(t) &= (1-t)B_{2,1}(t) + tB_{1,1}(t) = t^2 \qquad\qquad . \\
B_{0,3}(t) &= (1-t)B_{0,2}(t) + tB_{-1,2}(t) = (1-t)^3 \\
B_{1,3}(t) &= (1-t)B_{1,2}(t) + tB_{0,2}(t) = 3t(1-t)^2 \\
B_{2,3}(t) &= (1-t)B_{2,2}(t) + tB_{1,2}(t)3t^2(1-t) \\
B_{3,3}(t) &= (1-t)B_{3,2}(t) + tB_{2,2}(t)t^3
\end{aligned} \tag{2.7}$$

Using the property of recursive definition and derivative, the general expression on the derivative of a Bézier curves can be obtained as



(a) $n = 3$.

(b) $n = 5$.

(c) $n = 9$.

Figure 2.3: The basis function $B_{i,n}(t)$ for $n = 3, 5$, and 9.

$$\begin{aligned} x'(t) &= \sum_{i=0}^{n} p_i B'_{i,n}(t) \\ &= \sum_{i=0}^{n} p_i n(B_{i-1,n-1}(t) - B_{i,n-1}(t)) \\ &= n \sum_{i=0}^{n-1} (p_{i+1} - p_i) B_{i,n-1}(t). \end{aligned} \tag{2.8}$$

Thus, we obtain formulas for the end derivatives of a Bézier curve as

$$\begin{aligned} x'(0) &= n(p_1 - p_0), \qquad & x''(0) &= n(n-1)(p_0 - 2p_1 + p_2), \\ x'(1) &= n(p_n - p_{n-1}), \qquad & x''(1) &= n(n-1)(p_n - 2p_{n-1} + p_{n-2}). \end{aligned} \tag{2.9}$$

Note here that from (2.8) and (2.9), the derivative of any n-th degree Bézier curve is an $(n-1)$-th degree and the expression for the end of derivative at $t = 0, 1$ are symmetric.

2.3.1 Rational Bézier Curves

We briefly introduce rational Bézier curves. We here note that this is a particular case of rational B-spline curves which will be presented in Section 2.4. The Bézier form consists of polynomial, thus it offers many advantages. However, there exist a number of curve types which can not represented well by using Bézier form.

Then, a natural way to solve such an issue is to introduce the rational function. The rational Bézier curves of degree n-th can be defined as

$$x(t) = \frac{\sum_{i=0}^{n} w_i p_i B_{i,n}(t)}{\sum_{i=0}^{n} w_i B_{i,n}(t)} \qquad 0 \leq t \leq 1, \tag{2.10}$$

where w_i is weight coefficient. We here assume that $w_i \geq 0, \forall i$ holds. Then, we can written $x(t)$ in (2.10) as

$$x(t) = \sum_{i=0}^{n} p_i R_{i,n}(t) \qquad 0 \leq t \leq 1, \tag{2.11}$$

with

$$R_{i,n}(t) = \frac{w_i B_{i,n}(t)}{\sum_{j=0}^{n} w_j B_{j,n}(t)}. \tag{2.12}$$

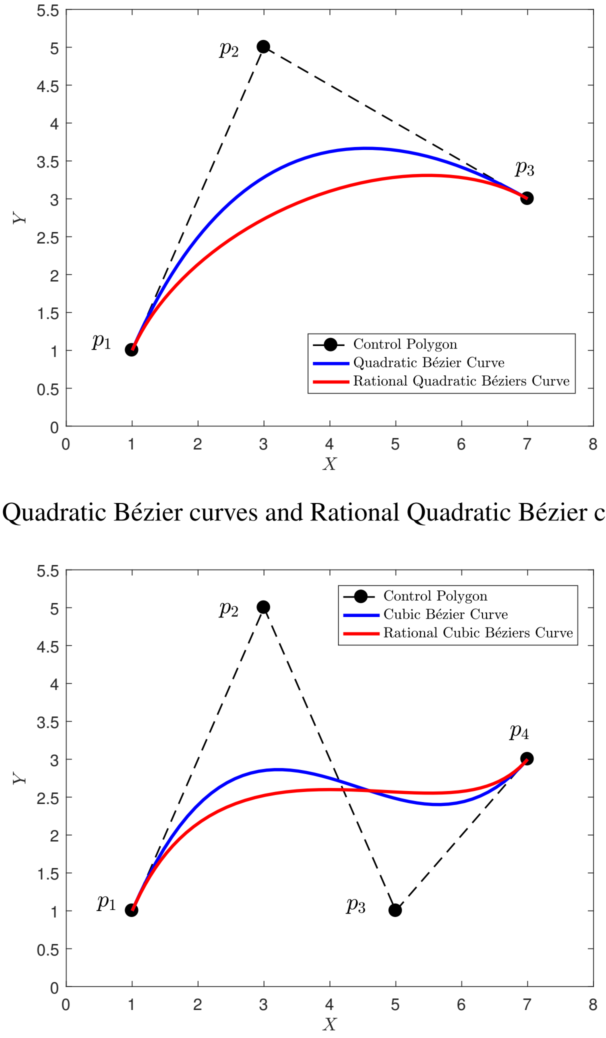
Here, $R_{i,n}(t)$ are the rational basis function for the curve form and have the properties of $B_{i,n}(t)$ as follows,

- The non-negativity property : $R_{i,n}(t) \geq 0, \quad 0 \leq t \leq 1$ for all of i, n.
- Partition of unity property : $\sum_{i=0}^{n} R_{i,n}(t) = 1, \quad 0 \leq t \leq 1$.
- $R_{0,n}(0) = R_{n,n}(1) = 1$.
- $R_{i,n}(t)$ attains exactly one maximum on the interval $[0, 1]$.
- When, $w_i = 1, \forall i$ holds, then we have $R_{i,n}(t) = B_{i,n}(t), \forall i$.
- The convex hull property : the curves are contained in the convex hulls of their defining control points p_i.
- The transformation invariance property : rotation and scalings are applied to the curve by applying them to the control points p_i.
- The end point of interpolation : $x(0) = p_0$ and $x(1) = p_n$.
- Spacial case of rational Bézier curves are the polynomial Bézier curves.

Figure 2.4 illustrates two examples of rational Bézier curves $x(t) = [X(t)\ Y(t)]^T \in \mathbf{R}^2$ for the case of $n = 2$ and $n = 3$ in $O - XY$ plane. For the sake of comparison, Bézier curves in Section 2.3.1 are also plotted.

2.4 B-spline Curves

We often face the difficulty that curves constituted by just one polynomial or rational segment are inadequate due to their drawbacks that a high degree may be required in order to satisfy a large number of constraints. For example, ($n - 1$)-degree is needed to pass a Bézier curve through a given set of n data points. In general, such a high degree curves are numerically unstable. For solving such



(a) Quadratic Bézier curves and Rational Quadratic Bézier curves.

(b) Cubic Bézier curves and Rational Cubic Bézier curves.

Figure 2.4: Design examples of Bézier curves and Rational Bézier curves.

an issue, a natural way is to introduce the so-called "piecewise polynomial". As one of the typical piecewise polynomials, we here present B-spline basis as well as B-spline curves using such a basis functions.

2.4.1 B-spline Basis Functions

We here define the B-spline basis which is given by a recurrence formula developed by de Boor and Cox [32, 33]. Let $T = t_{-k}, \cdots, t_m$ be a nondecreasing sequence of real number as $t_i \leq t_{i+1}, i = -k, \cdots, m-1$. Here, t_i are called knot. The i-th of B-spline basis function of degree k (order $k+1$), denoted by $N_{i,k}$ and can be defined as

$$N_{i,0}(t) = \begin{cases} 1 & if \quad t_i \leq t \leq t_{i+1} \\ 0 & otherwise. \end{cases}, \tag{2.13}$$

$$N_{i,k}(t) = \frac{t - t_i}{t_{i+k} - t_i} B_{i,k-1}(t) + \frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}} B_{i+1,k-1}(t). \tag{2.14}$$

Here, the critical properties of the B-spline basis functions can be summarized as follows and sample for B-spline basis functions were shown in Figure 2.5.

- Local support property : $N_{i,k}(t) = 0$ if t is outside the interval $[t_i, t_{i+k+1})$.
- In any given knot interval $[t_j, t_{j+1})$, at most $k + 1$ of the $N_{i,k}(t)$ (i.e., $N_{j-k,k}, \cdots, N_{j,k}$) are nonzero.
- Non-negativity : $N_{i,k}(t) \geq 0, \forall i, k$, and t.
- Partition of unity : it holds that $\sum_{j=i-k}^{i} N_{j,k}(t) = 1, \forall t, t \in [t_i, t_{i+1})$.
- Derivatives: all derivatives of $N_{i,k}(t)$ exist in the interior of a knot interval. Also a knot $N_{i,k}(t)$ is $k - p$ times continuously differentiable where p is the multiplicity of the knot. Thus, increasing degree increases continuity and increasing knot multiplicity decreases continuity.
- Except for the case of $k = 0$, $N_{i,k}(t)$ attains exactly one maximum value.

Next, we represent the derivative of basis function $N_{i,k}(t)$. The derivative of a basis function is given by

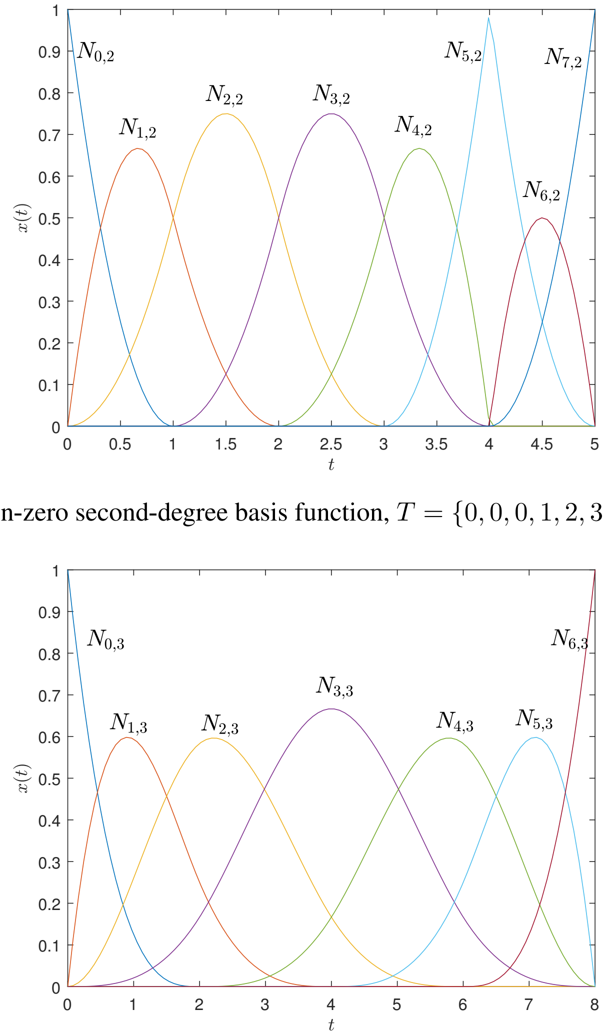
$$N'_{i,k}(t) = \frac{k}{t_{i+k} - t_i} N_{i,k-1}(t) - \frac{k}{t_{i+k+1} - t_{i+1}} N_{i+1,k-1}(t). \tag{2.15}$$

Using the product rule $(fg)' = f'g + fg'$ in to the basis function in (2.14) as

$$N_{i,k}(t) = \frac{t - t_i}{t_{i+k} - t_i} N_{i,k-1}(t) - \frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}} N_{i+1,k-1}(t),$$

and then equation we have

$$\begin{aligned} N'_{i,k}(t) = {} & \frac{1}{t_{i+k} - t_i} N_{i,k-1}(t) + \frac{t - t_i}{t_{i+k} - t_i} N'_{i,k-1}(t) \\ & - \frac{1}{t_{i+k+1} - t_{i+1}} N_{i+1,k-1} + \frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}} N'_{i+1,k-1}(t). \end{aligned} \tag{2.16}$$



(a) The non-zero second-degree basis function, $T = \{0, 0, 0, 1, 2, 3, 4, 4, 5, 5, 5\}$.

(b) Cubic B-spline basis function.

Figure 2.5: An example of B-spline basis functions.

Substituting the equation (2.15) in to (2.16) for $N'_{i,k-1}(t)$ and $N'_{i+1,k-1}(t)$ yields

$$\begin{aligned}
N'_{i,k}(t) &= \frac{1}{t_{i+k}-t_i}N_{i,k-1}(t) - \frac{1}{t_{i+k+1}-t_{i+1}}N_{i+1,k-1}(t) \\
&\quad + \frac{t-t_i}{t_{i+k}-t_i}\left(\frac{k-1}{t_{i+k-1}-t_i}N_{i,k-2}(t) - \frac{k-1}{t_{i+k}-t_{i+1}}N_{i+1,k-2}(t)\right) \\
&\quad + \frac{t_{i+k+1}-t}{t_{i+k+1}-t_{i+1}}\left(\frac{k-1}{t_{i+k}-t_{i+1}}N_{i+1,k-2}(t) - \frac{k-1}{t_{i+k+1}-t_{i+2}}N_{i+2,k-2}(t)\right) \\
&= \frac{1}{t_{i+k}-t_i}N_{i,k-1}(t) - \frac{1}{t_{i+k+1}-t_{i+1}}N_{i+1,k-1}(t) + \frac{k-1}{t_{i+k}-t_i}\frac{t-t_i}{t_{i+k+1}-t_i}N_{i,k-2}(t) \\
&\quad + \frac{k-1}{t_{i+k}-t_{i+1}}\left(\frac{t_{i+k+1}-t}{t_{i+k+1}-t_{i+1}} - \frac{t-t_i}{t_{i+k}-t_i}\right)N_{i+1,k-2}(t) \\
&\quad - \frac{k-1}{t_{i+k+1}-t_{i+1}}\frac{t_{i+k+1}-t}{t_{i+k+1}-t_{i+2}}N_{i+2,k-2}(t).
\end{aligned} \tag{2.17}$$

Here, noting that

$$\begin{aligned}
\frac{t_{i+k+1}-t}{t_{i+k+1}-t_{i+1}} - \frac{t-t_i}{t_{i+k}-t_i} &= -1 + \frac{t_{i+k+1}-t}{t_{i+k+1}-t_{i+1}} + 1 - \frac{t-t_i}{t_{i+k}-t_i} \\
&= -\frac{t_{i+k+1}-t_{i+1}}{t_{i+k+1}-t_{i+1}} + \frac{t_{i+k+1}-t}{t_{i+k+1}-t_{i+1}} + \frac{t_{i+k}-t_i}{t_{i+k}-t_i} - \frac{t-t_i}{t_{i+k}-t_i} \\
&= \frac{t_{i+k}-t}{t_{i+k}-t_i} - \frac{t-t_{i+1}}{t_{i+k+1}-t_{i+1}},
\end{aligned} \tag{2.18}$$

we can get the following expression of $N'_{i,k}(t)$ by using the deBoor and Cox formula in (2.14).

$$\begin{aligned}
N'_{i,k}(t) &= \frac{1}{t_{i+k}-t_i}N_{i,k-1}(t) - \frac{1}{t_{i+k+1}-t_{i+1}}N_{i+1,k-1}(t) \\
&+ \frac{k-1}{t_{i+k}-t_i}N_{i,k-1}(t) - \frac{k-1}{t_{i+k+1}-t_{i+1}}N_{i+1,k-1}(t) \\
&= \frac{k}{t_{i+k}-t_i}N_{i,k-1}(t) - \frac{k}{t_{i+k+1}-t_{i+1}}N_{i+1,k-1}(t).
\end{aligned} \tag{2.19}$$

In addition, letting $N^{(l)}_{i,k}(t)$ be the l-th derivative of $N_{i,k}(t)$, then it can be shown that $N^{(l)}_{i,k}(t)$ is obtained as

$$\begin{aligned}
N^{(l)}_{i,k}(t) &= k(\frac{N^{(l-1)}_{i,k-1}(t)}{t_{i+k}-t_i} - \frac{N^{(l-1)}_{i+1,k-1}(t)}{t_{i+k+1}-t_{i+1}}) \\
&= \frac{k}{k-l}(\frac{t-t_i}{t_{i+k}-t_i}N^{(l)}_{i,k-1}(t) + \frac{t_{i+k+1}-t}{t_{i+k+1}-t_{i+1}}N^{(l)}_{i+1,k-1}(t)),
\end{aligned} \tag{2.20}$$

for $l = 0, \cdots, l-1$.

2.4.2 B-spline Curves

A k-th degree of B-spline curves $x(t)$ can be define by

$$x(t) = \sum_{i=-k}^{m-1} p_i N_{i,k}(t) \qquad t_0 \le t \le t_m, \tag{2.21}$$

where p_i denotes the control points and $N_{i,k}(t)$ are the k-th degree of B-spline basis functions as in (2.14) defined on the non-uniform knot vector T as $m+k+1$ knots as

$$T = \{t_{-k}, t_{-k+1}, t_{-k+2} \cdots, t_m\}. \tag{2.22}$$

For the sake of simplicity, we generally assume $t_0 = 0$ and $t_m = 1$ in (2.22). The polygon formed as a sequence of control points p_i is called as "control polygon". The method for computing a point on B-spline curves are needed to find the knot interval. In addition, we need to compute the nonzero basis functions, and multiply the values of the nonzero basis functions with corresponding control points. The properties of B-spline curves in (2.21) can be listed as follows,

- If $n = k$ and $T = \{0, \cdots, 0, 1, \cdots, 1\}$ holds, the curves $x(t)$ is equivalent to Bézier curves.
- The curves $x(t)$ is a piecewise polynomial curves since $N_{i,k}(t)$ are piecewise polynomial.
- It holds that $x(0) = p_{-1}$ and $x(1) = p_{m-1}$, if it hold that $p_{-k} = \cdots = p_{-1}$ and $p_{m-k} = \cdots = p_{m-1}$.
- Affine invariance : An affine transformation to the curves can be defined as one to control points.
- Strong convex hull property : the curve $x(t)$ is contained in the convex hell of its control polygon.
- Local modification scheme : moving p_i changes $x(t)$ only in the interval $[t_i, t_{i+k+1})$ since $N_{i,k}(t) = 0$ for $t \notin [t_i, t_{i+k+1})$.
- The control polygon represents geometrical approximation to the curves.
- Variation diminishing property : no plane has more intersections with the curves than control polygon.
- The continuity and differentiability of $x(t)$ follow from the $N_{i,k}(t)$ since $x(t)$ is a linear combination of the $N_{i,k}(t)$. Thus, it is shown that $x(t)$ is infinitely differentiable in the interior of knot interval. Also, it is at least $k - l$ times continuously differentiable at a knot of multiplicity l. This is simply a consequence of the fact that discontinuous functions can be combined when the result is continuous.
- Multiple (coincident) control points : This follows from property that $x(t)$ is in the convex hull. More than that, since the knot has multiplicity the curve must be continuous and it has a cusp (visual discontinuity).

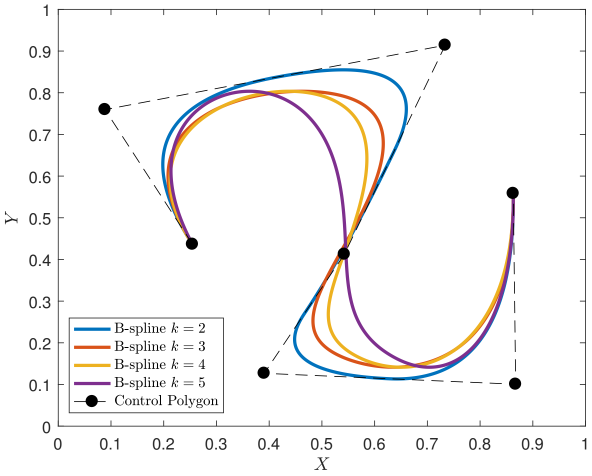


Figure 2.6: B-spline curves of degree $k = 2, 3, 4$ and 5 with the control polygons.

Figure 2.6 illustrates design examples of B-spline curves $x(t) = [X(t) \;\; Y(t)]^T \in \mathbf{R}^2$ of degree $k = 2, 3, 4$ and 5 with control polygons in $O - XY$ plane.

Let $x^{(l)}(t)$ denote the $order$-th derivative of $x(t)$ for $l = 0, \cdots, k-1$. Then, we have

$$x^{(l)}(t) = \sum_{i=-k}^{m-1} p_i N_{i,k}^{(l)}(t). \tag{2.23}$$

Then, from the equation (2.15) and (2.24), we can get as

$$\begin{aligned}
x'(t) &= \sum_{i=-k}^{m-1} p_i N'_{i,k}(t) \\
&= \sum_{i=-k}^{m-1} p_i \Big(\frac{k}{t_{i+k} - t_i} N_{i,k-1}(t) - \frac{k}{t_{i+k+1} - t_{i+1}} N_{i+1,k-1}(t)\Big) \\
&= \Big(k \sum_{i=-k-1}^{m-2} \frac{p_{i+1}}{t_{i+k+1} - t_{i+1}} N_{i+1,k-1}(t)\Big) - \Big(k \sum_{i=-k}^{m-1} \frac{p_i}{t_{i+k+1} - t_{i+1}} N_{i+1,k-1}(t)\Big) \\
&= k \frac{p_{-k} N_{-k,k-1}(t)}{t_k - t_0} + k \sum_{i=-k}^{m-2} \frac{p_{i+1} - p_i}{t_{i+k+1} - t_{i+1}} N_{i+1,k-1}(t) - k \frac{p_n N_{n+1,k-1}(t)}{t_{n+k+1} - t_{n+1}} \\
&= k \sum_{i=-k}^{m-2} \frac{p_{i+1} - p_i}{t_{i+k+1} - t_{i+1}} N_{i+1,k-1}(t) \\
&= \sum_{i=-k}^{m-2} Q_i N_{i+1,k-1}(t)
\end{aligned}, \tag{2.24}$$

where

$$Q_i = \frac{p_{i+1} - p_i}{t_{i+k+1} - t_{i+1}}. \tag{2.25}$$

Then, let T' be the knot vector obtained by dropping the first and last knots for T as

$$T' = \{0, \cdots, 0, t_0, \cdots, t_{m-k-1}, 1, \cdots, 1\}. \tag{2.26}$$

We note that in the (2.26) T' has $m-1$ knots and the functions $N_{i+1,k-1}(t)$ that competed on T will equal to $N_{i,k-1}(t)$ computed on T'. Thus

$$x'(t) = \sum_{i=-k}^{m-2} Q_i N_{i,k-1}(t). \tag{2.27}$$

Q_i is in (2.25), the $N_{i,k-1}(t)$ are computed on T' and $x'(t)$ is $(k-1)$-th degree B-spline curve. Also, the first derivatives at the end point of a B-spline curve are given by

$$\begin{aligned} x'(0) &= Q_0 = \frac{k}{t_{k+1}}(p_1 - p_0) \\ x'(1) &= Q_{n-1} = \frac{k}{1 - t_{m-k-1}}(p_n - p_{n-1}) \end{aligned}. \tag{2.28}$$

Also, let $x^{(l)}(t)$ be the k-th derivative of $x(t)$. Then, we have

$$x^{(l)}(t) = \sum_{i=-k}^{m-l-1} p_i^{(l)} N_{i,k-l}(t), \tag{2.29}$$

where $p_i^{(l)}, i = -k, \cdots, m-l-1$ is given as

$$p_i^{(l)} = \begin{cases} p_i & \text{if} \quad l = 0 \\ \frac{k-l+1}{t_{i+k+1} - t_{i+l}}(p_{i+1}^{(l-1)} - p_i^{(l-1)}) & \text{if} \quad l > 0 \end{cases}. \tag{2.30}$$

2.5 NURBS

We here present Non-Uniform Rational B-Spline curves. A k-th degree Non-Uniform Rational B-spline (NURBS) curves is defined as

$$x(t) = \frac{\sum_{i=-k}^{m-1} w_i p_i N_{i,k}(t)}{\sum_{i=-k}^{m-1} w_i N_{i,k}(t)} \quad t_0 \le t \le t_m, \tag{2.31}$$

where p_i denotes the control points (forming a control polygon) and w_i are weights. Also, $N_{i,k}(t)$ are k-th degree of B-spline basis function in (2.14) defined on non uniform knot vector in (2.22), i.e.,

$$T = \{t_{-k}, t_{-k+1}, \cdots, t_{m-1}\}. \tag{2.32}$$

Here, we assume that $t_0 = 0, t_m = 1$ and $w_i > 0, \forall i$. Let $R_{i,k}(t)$ be

$$R_{i,k}(t) = \frac{w_i N_{i,k}(t)}{\sum_{j=-k}^{m-1} w_j N_{j,k}(t)}. \tag{2.33}$$

Then we rewrite (2.30) as

$$x(t) = \sum_{i=-k}^{m-1} p_i R_{i,k}(t). \tag{2.34}$$

Here, $R_{i,k}(t)$ denotes the rational basis functions. Thus, they are piecewise rational function on $t \in [0,1]$.

The $R_{i,k}(t)$ have the properties derived from the equation (2.33) as following list.

- Non-negativity : $R_{i,k}(t) \geq 0, \forall i, k, t \in [0,1]$.
- Partition of unity : $\sum_{i=0}^{n} R_{i,k}(t) = 1, \forall t \in [0,1]$.
- $R_{0,k}(0) = R_{n,k}(t) = 1$.
- For $k > 0$, all $R_{i,k}(t)$ attain exactly one maximum on the interval $t \in [0,1]$.
- Local support : $R_{i,k}(t) = 0$ when $t \notin [0,1]$. Moreover, in any given knot span $k+1$ of the $R_{i,k}(t)$ are nonzero (in general $R_{i-k,k}(t), \cdots, R_{i,k}(t)$ are nonzero in $[t_i, t_{i+1})$).
- Derivatives of $R_{i,k}(t)$ exist in the knot interval. And, it is a rational function with nonzero denominator. At a knot, $R_{i,k}(t)$ is $k-p$ times continuously differentiable, where p is the multiplicity of the knot.
- If it holds that $w_i = 1, \forall i$, then $R_{i,k}(t) = N_{i,k}(t), \forall i$. Namely, $N_{i,k}(t)$ are special case of the $R_{i,k}(t)$. Also, for $\forall a \neq 0$, if it holds that $w_i = a$ for $\forall i$, then we have $R_{i,k}(t) = N_{i,k}(t), \forall i$.
- $x(0) = p_{-1}$ and $x(1) = p_{m-1}$, if it holds that $p_{-k} = \cdots = p_{-1}$ and $p_{m-k} = \cdots = p_{m-1}$.
- Affine invariance : An affine transformation can be defined as one on the control points.

- Strong convex hull property : if $t \in [t_i, t_{i+1})$, the curves $x(t)$ within the convex hull of the control points $p_{i-k}, \cdots, p_i$ for $t \in [t_i, t_{i+1})$.
- $x(t)$ is infinitely differentiable on the knot interval. Also, it is $k - l$ times differentiable at a knot of multiplicity l.
- Variation diminishing property : There is no plane which has more intersections with the curve than with the control polygon.
- A NURBS curve with no interior knot is equivalent to a rational Bézier curves, since the $N_{i,k}(t)$ reduce to the $B_{i,k}(t)$. This means that NURBS curves contain nonrational B-spline and rational and nonrational Bézier curves as special cases.
- Local approximation : If the control points p_i is moved or the weight w_i is changed, it affects only that portion of the curve on the interval $t \in [t_i, t_{i+k+1})$.

Property of local approximation is important for interactive shape design. Using NURBS curves, both control point movement and weight modification can be utilized to attain local shape control. Figures 2.7 and 2.8 illustrate the design example of rational cubic B-splines in $O - XY$ plane and their basis functions, respectively.

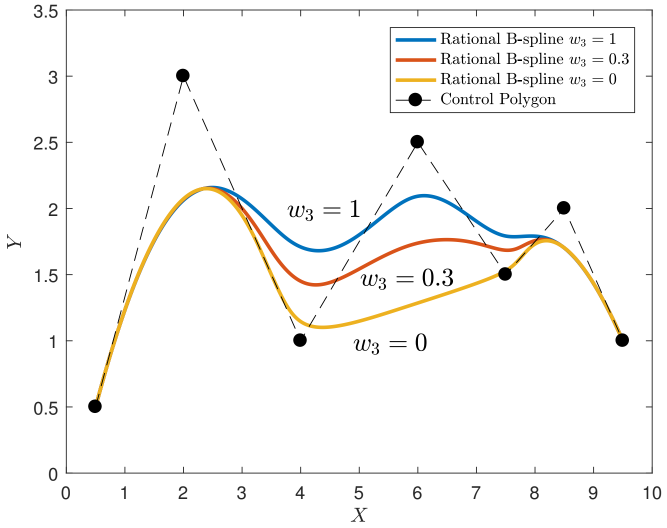
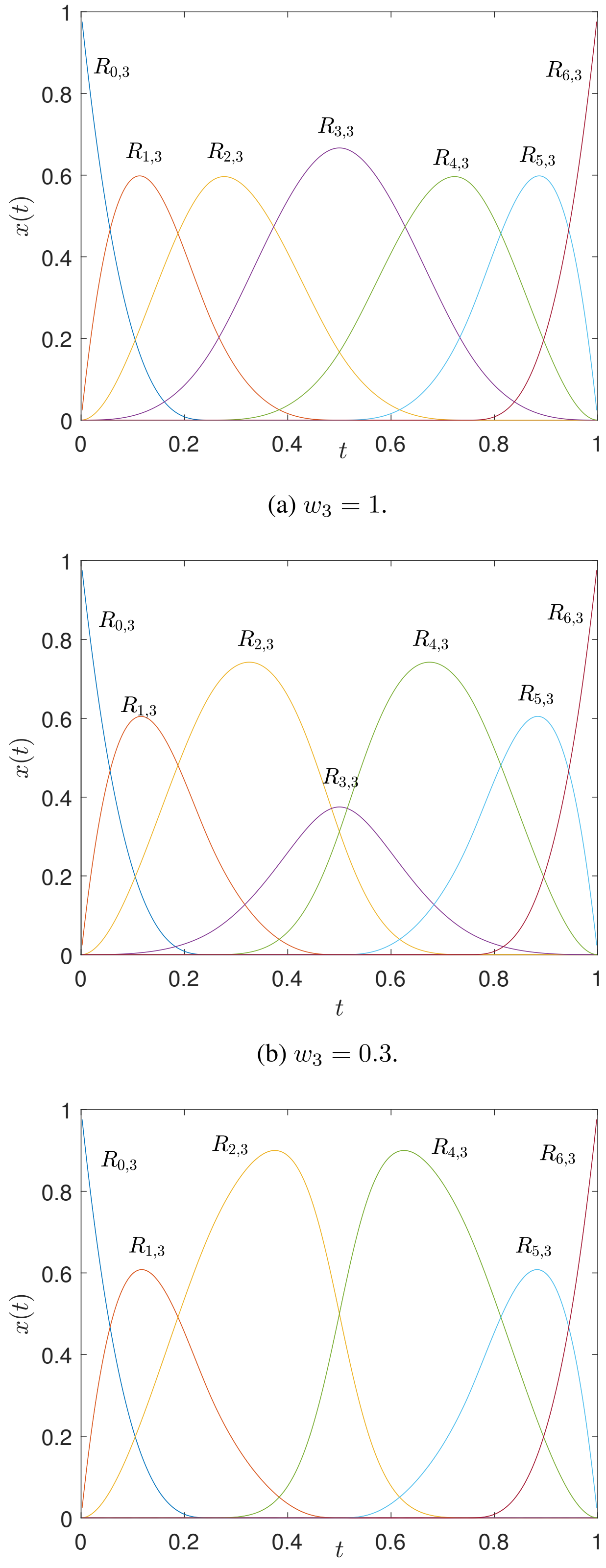


Figure 2.7: Rational cubic B-spline curves with w_3 varying.



(a) $w_3 = 1$.

(b) $w_3 = 0.3$.

(c) $w_3 = 0$.

Figure 2.8: The cubic basis functions for the curves in Figure 2.7 with w_3 varying.

Chapter 3

Dynamic Programming Approach

3.1 Introduction

In this chapter, we present graph theory [34] and dynamic programming [35], which will be used for approximate data points throughout in this thesis. In Section 3.2, we briefly present the basic of graph theory, in which the directed acyclic graph (DAG) is included. Then, the dynamic programming is introduced in Section 3.3, which will be employed in order to select the dominant control points.

3.2 Graph

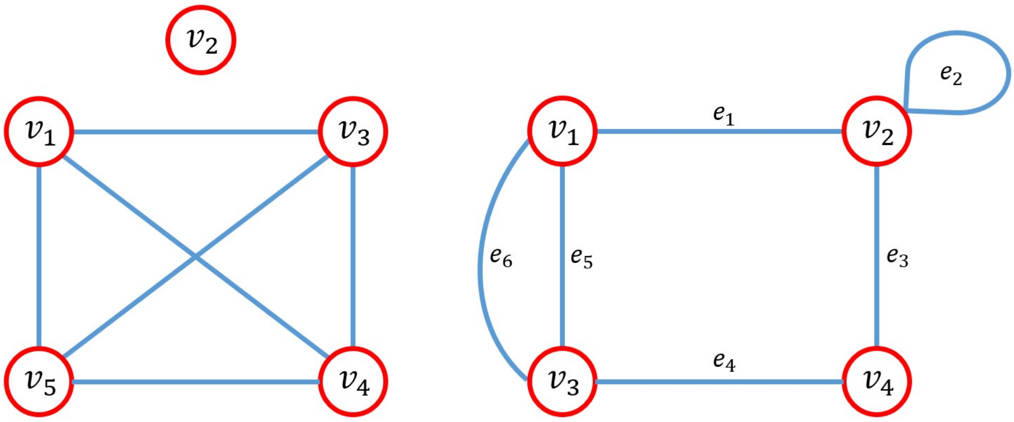
3.2.1 General Graph

Graph theory [34] is a study of graph, which has mathematical structures used in order to model pairwise relations between objects. Such a model involves the ways in which sets of vertices can be connected by edges.

Let G be a graph which consists of two finite sets V and E. Here, $V = \{v_i\}$ is the set of vertex, which is a non-empty set of elements. Also, $E = \{e_i\}$ is the edge set which is a possibly empty set of elements, where each edge e_i is assigned as an unordered pair of vertices (v_i, v_j). Letting $|V|$ and $|E|$ be order and size of the graph, then they are given as $|V| = n$ and $|E| = m$ respectively for proper n and m.

We here introduce some terms on the graph:

- Self-loop edge: An edge e_i with the same vertex as both of its end vertices (e.g., e_2 in Figure 3.1(b)).
- Parallel edge: An edge e_i which the more than one edge $e_j (j \neq i)$ is associated with a given pair of vertices (e.g., e_5 and e_6 in Figure 3.1(b)).
- Simple graph: A graph that has neither self-loops nor parallel edges (Figure 3.1(a)).
- Multi-graph: An ordered pair $G = (V, E)$ with V a set of vertices or nodes and E a multi-set of unordered pairs of vertices called edges (Figure 3.1(b)).



(a) Simple graph.

(b) Multi graph.

Figure 3.1: An example of a simple graph.

- Finte graph: A graph with finite number of vertices and finite number of edges. Otherwise, the graph is called as ‘infinite graph’.

3.2.2 Directed Graph

We briefly present the ‘directed graph’ (digraph). A directed graph is a set of vertices which are connected with the order pair of vertices. That is, a directed edge points are connected from the first vertex to second vertex. In particular, a directed graph or digraph is formed by vertices connected by directed edges (arcs).

Suppose that we are given a digraph as (V, E), where V is a set of vertices and E is a vertex pairs’ set same as in common graph. The difference is that every elements of E are order pairs, and that arcs from vertex v_i to the vertex v_j can be written as (v_i, v_j). The other pair (v_j, v_i) is the opposite direction arc. Also, we should keep track of the multiplicity of the arcs.

Here, we summarize some notations and properties for the directed graph.

- Let u_i and v_j be the initial and terminal vertex of the arc (v_i, v_j). Then, the arc is incident out of u_i and incident into v_j.
- The out-degree of the vertex v_j is the number of arcs out of it and the in-degree of v_j the number of arcs going into it. They are denoted $d^+(v_j)$ and $d^-(v_j)$, respectively.
- In the case of directed walk (trail, path or circuit (i.e., $v_{i_0}, e_{j_1}, v_{i_1}, e_{j_2}, \cdots, e_{j_k}, v_{i_k}$), then v_{i_l} is the initial vertex and $v_{i_{l-1}}$ the terminal vertex of the arc e_{j_l}.
- Vertices v_i and v_j are strongly connected if there is a directed v_i-v_j path and also a directed v_i-v_j path in G.
- Digraph G is strongly connected if every pair of vertices is strongly connected.
- A strongly connected component H of the digraph G is a directed subgraph of G such that H is strongly connected. However, if we add any vertices or arcs to it, then it is not strongly connected anymore.

The degree of v_j is $d(v_j) = d^+(v_j) + d^-(v_j)$. Then, by Handshaking Lemma in [36], the number of edge $|E_G|$ is given as

$$\sum_{v_j \in G} d^+(v_j) = |E_G| = \sum_{v_j \in G} d^-(v_j). \tag{3.1}$$

For example, the graph G in Figure 3.2 has

$$\begin{aligned} d^+(v_1) &= 2, & d^-(v_1) &= 2, \\ d^+(v_2) &= 2, & d^-(v_2) &= 2, \\ d^+(v_3) &= 2, & d^-(v_3) &= 2. \end{aligned} \tag{3.2}$$

Then, the number of edges of G is given as

$$\sum_{v_j \in G} d^+(v_j) = \sum_{v_j \in G} d^-(v_j) = |E_G| = 6. \tag{3.3}$$

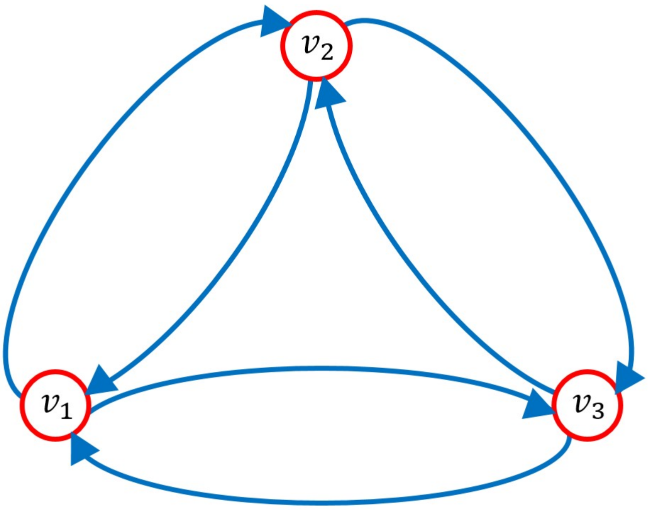


Figure 3.2: An example directed graph or digraph.

3.2.3 Directed Acyclic Graph

A directed acyclic graph (DAG) is a finite directed graph with no cycles. Normally, each directed edge between vertex to another vertex are no way to start at any vertex and follow a consistently-directed sequence of edges. Also, DAG is a directed graph that has a topological ordering sequence of the vertices such that every edge is directed from earlier to later in the sequence. Also, DAG can model many different kinds of information and can also be used as a compact representation of sequence data, such as the directed acyclic word graph representation of a collection of strings, or the binary decision diagram representation of sequences of binary choices.

A similar concept for undirected graph is an undirected graph without cycles. However, there are many other kinds of directed acyclic graph that are not formed by orienting the edges of an undirected acyclic graph. Moreover, every undirected graph has an acyclic orientation, an assignment of a direction for its edges that makes it into a directed acyclic graph. The DAG are not the same thing as directed versions of undirected acyclic graph, and some authors call them directed acyclic graph or acyclic digraph (e.g., [36]), as an example in Figure 3.3.

In an acyclic digraph, there exists at least one source (a vertex whose in-degree is zero) and at least one sink (a vertex whose out-degree is zero). For proofing, we let G be an acyclic digraph and G has no arcs. Otherwise, we consider the directed path $v_0, e_1, v_1, e_2, \cdots, e_k, v_k$, which has the maximum path length k. Since G is acyclic, $v_0 \neq v_k$. If (v, v_0) is an arc, then one of the following is true.

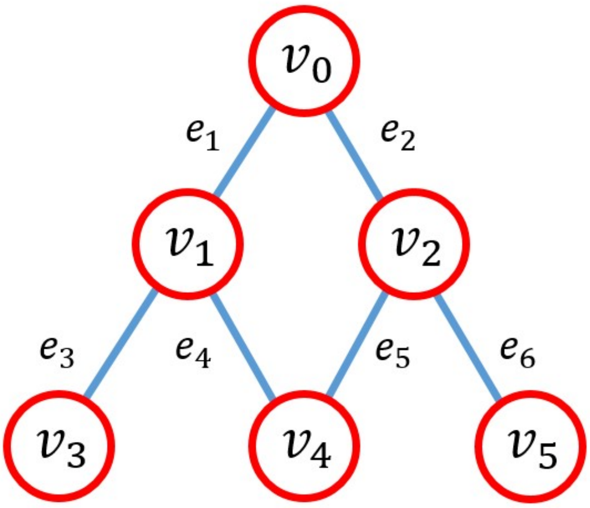


Figure 3.3: An example of directed acyclic graph (DAG).

- $v \neq v_t$ for every value of $t = 0, \cdots, k$ as $v, (v, v_0), v_0, e_1, v_1, e_2, \cdots, e_k, v_k$ is a directed path with length $k + 1$.

- $v = v_t$ for some value of t and choose the smallest t. Then, $t > 0$ because there are no loops in G and $v_0, e_1, v_1, e_2, \cdots, e_k, v_k, (v, v_0), v_0$ is a directed circuit.

Hence, $d^-(v_0) = 0$ and using a similar technique, we can show that $d^+(v_k) = 0$ as well.

3.3 Dynamic Programming

We briefly present the 'Dynamic Programming' used in the development of optimal control point selection in Chapters 4 and 5.

Dynamic programming (DP) [35] is a powerful method for solving optimization problems. As in the computer application, it used to solve many kind of the problems such as curve detection [37], and deformable object matching [38], etc. The main idea of DP is to decompose a problem from the main problem into sub-problems, such that we can directly given a solution to the sub-problems. Then, we can obtain the solution for the main problem by only solving sub-problems, recursively.

3.3.1 Dynamic Programming Algorithms

Now, let S be a solution of the optimization problem for all of x element and it can be written as $x \in S$ and E denoted the cost function. We then consider m-dimensional search space of the form $S = L^m$ where L is an arbitrary finite set and we refer to L as a set of labels. Then, the solution of the optimization problem $x \in L^m$ will be written as $(x_1, \cdots, x_m)$ for $x_i \in L$.

Suppose that we have a sequence of m elements. Then, we assign a label from L to each element in the sequence. Let D_i represents the minimum total weight of the path from the source x_1 to the x_i and $\gamma(x_i, x_{i+1})$ be a weight of each edge between x_i and x_{i+1}. A particularly common class of the cost functions in computer vision can be written as follows,

$$E(x_1, \cdots, x_m) = \sum_{i=1}^{m} D_i(x_i) + \sum_{i=1}^{m-1} \gamma(x_i, x_{i+1}). \tag{3.4}$$

This function calculates the cost setting in the label to each element between x_i and x_{i+1}.

Normally, L is a subset of $\mathbf{R}$ and the pairwise cost can be defined as follows,

$$\gamma(x_i, x_j) = ||x_i - x_j||^p \qquad p > 1. \tag{3.5}$$

In this thesis, we develop dynamic programming algorithms based on the observation, for approximation and remove some of control points gives approximation. Our algorithms is to find an acceptable approximation by selecting a points from original ones by using the method of DAG. The detail will be described in Chapter 4.

Chapter 4

A Design of Compact Planar B-spline Curves

4.1 Introduction

We here particularly focus on the design of characters based on the so-called dynamic font [24] as an application using the planar B-spline curves. Therein, the characters constituted as a result of planar trajectory curves using the normalized uniform B-splines as basis functions [41]. That is, their knot points are equally-spaced. Then, a sequence of control points of B-splines called ‘control polygon’, which represents the geometrical outline of curves, say characters. Such a treatment on the control polygon is compelling for manipulating characters. We may readily see that the design may depend on the number of control points on the characters. If characters can represent as a trajectory curves using only the dominant control points of original ones, then the design may considerably increase.

For achieving such a design, one of the issues may be to represent a given planar curve as more compact B-spline curves by using only the dominant control points, in which the desired approximation accuracy is achieved. Therefore, we here develop a method for designing such compact planar B-spline curves by introducing an idea on optimal control point selection. For such an optimal control point selection, the most typical approach may be a try-and-error approach exhibited in Lyche’s work [28]. Therein, the larger number of control points are initially defined, and then certain knot points are removed by evaluating the error of fitting curves. Similar work has been exhibited in Park’s work [29] and [42]. However, it is well known that these try-and-error approaches lead to huge computational time.

On the other hand, Tjahjowidodo et al. [30] have recently developed a fast algorithm for optimally finding knot points using the so-called half split method, where it is limited to cubic splines. The basic idea is to approximate a subset of the second derivative sample points as a set of piecewise linear functions. This may be applicable to cases where we want to select dominant control points from original ones. But, it may be impossible to apply this method when we want to specify the number of dominant control points, as in the following discussion of this study. The main purpose of this study is to develop a method for designing compact planar B-spline curves [43], where the results of the cubic case can be readily extended to ones of arbitrary degrees [44]. In particular, we here develop a method for selecting the dominant control points by introducing the dynamic programming (DP) approach [45] and a new idea for knot points selection based on a multi-level error function, where the term multi-level means that not only function values of a given curve are considered but also the derivatives. Moreover, it is shown that representation using the selected control points can be realized using nonuniform rational B-splines (NURBS). We demonstrate the approachs performance with some experimental studies using handwriting data.

4.2 Problem Statement

We first give a problem statement on the design of compact planar B-spline curves. Now suppose that we are given a planar curve $x(t) = [X(t)\ Y(t)]^T \in \mathbf{R}^2, t \in [t_0, t_m]$ using normalized uniform B-spline functions of degree three as the basis functions in Chapter 2,

$$x(t) = \sum_{i=-k}^{m-1} \tau_i B_3(\alpha(t - t_i)), \tag{4.1}$$

subject to initial and terminal conditions:

$$\begin{aligned} x(t_0) = p, \quad x^{(1)}(t_0) = x^{(2)}(t_0) = 0_2 \\ x(t_m) = q, \quad x^{(1)}(t_m) = x^{(2)}(t_m) = 0_2 \end{aligned}, \tag{4.2}$$

where $x^{(l)}(\cdot)$ denotes the l-th derivatives of $x(\cdot), p, q \in \mathbf{R}^2$ some constant vectors, and $0_2 = [0\ 0]^T \in \mathbf{R}^2$. On the other hand, the initial and terminal conditions in (4.2) are imposed by taking the first and last three control points to be hold, i.e.,

$$\begin{aligned} \tau_{-3} = \tau_{-2} = \tau_{-1} (= p) \\ \tau_{m-3} = \tau_{m-2} = \tau_{m-1} (= q) \end{aligned} . \tag{4.3}$$

Since $x(t)$ in B-spline functions is a piecewise polynomial, then it can be shown that $x(t)$ at each knot point $t_i, i = 0, \cdots, m$ (i.e., $x(t_i), i = 0, \cdots, m$) and its first and second derivatives (i.e., $x^{(1)}(t_i)$ and $x^{(2)}(t_i)$) are expressed in the case of degree is 3 as follows,

$$\begin{aligned} x(t_i) &= \frac{1}{6}(\tau_{i-1} + 4\tau_{i-2} + \tau_{i-3}) \\ x^{(1)}(t_i) &= \frac{\alpha}{2}(\tau_{i-1} - \tau_{i-3}) \\ x^{(2)}(t_i) &= \alpha^2(\tau_{i-1} - 2\tau_{i-2} + \tau_{i-3}) \end{aligned} . \tag{4.4}$$

Now, let $\tau \in \mathbf{R}^{2\times M} \quad (M = m + 3)$ be a matrix consisting of control point $\tau_i \in \mathbf{R}^2 \quad (i = -3, -2, \cdots, m-1)$ as

$$\tau = [\tau_{-3} \quad \tau_{-2} \quad \cdots \quad \tau_{m-1}]. \tag{4.5}$$

Then, a polygonal line in $\mathbf{R}$ formed from the sequence of control points τ_i's in (4.5) is called as "control polygon" $\mathcal{M}$, which defined as

$$\mathcal{M} = \tau_{-3} \quad \tau_{-2} \quad \cdots \quad \tau_{m-1}. \tag{4.6}$$

The control polygon $\mathcal{M}$ represents a geometrical outline of the B-spline curves $x(t)$. Such a property is very convenient when we want to manipulate the planar curves $x(t)$ on rotation, translation, and resizing [47]. However, the manipulations might depend on the size of τ (i.e., M). For example, the above method may cause that M becomes large as we set a significant value to α. Such a case may be appropriate when we want to edit the shape of characters locally. On the other hand, when we want to globally manipulate the curves, representing the curves with the small size of τ as possible may be desirable.

For improving such a problem on global manipulation, a natural idea may be to find some control points of M, which are dominant on the path of curves $x(t)$. We then consider the following problem.

Problem 1 : Suppose that a control polygon $\mathcal{M}$, or the matrix τ in (4.5), corresponding to the planar curves $x(t)$ is given. Then, find a set of a specified number of control points (i.e., dominant control points) from $\mathcal{M}$ such that the desired approximation accuracy is achieved.

The curves yielded from such dominant control points obtained from Problem 1 will be referred to compact planar B-spline curves in the sequel.

4.3 Reconstructing Compact Planar B-spline Curves using DP Control Point Selection

Our main task for solving Problem 1 in the previous section is to develop a method for optimally selecting a set of dominant control points from the control polygon $\mathcal{M}$ on $x(t)$ in (4.1).

In Section 4.3.1, we first present a problem that is based on Hu and Watt's work in [45]. Therein, Problem 1 is formulated as graph problem. We then present the algorithm for solving such a graph problem by employing the dynamic programming. Here, we mainly introduce the so-called multi-level error functions, where the term multi-level means that not only function values but also its derivatives will consider. In section 4.3.2, we show how to represent the compact B-spline curves from the selected control points by introducing Non-Uniform Rational B-splines (NURBS).

4.3.1 Optimal Control Point Selection using Dynamic Programming for Compact Planar B-spline Curves

We first formulate Problem 1 as a graph problem that is based on Hu and Watt's work in [45].

Let G be a weighted directed acyclic graph (DAG) in Chapter 3. We then construct G from original control polygon $\hat{\mathcal{M}} \in \mathbf{R}^{M-4}$ defined by

$$\hat{\mathcal{M}} = \tau_{-1}\ \tau_0\ \cdots\ \tau_{m-3}, \tag{4.7}$$

consisting of $M-4$ points, which we regard as the main part of the original polygon $\mathcal{M}$ as

$$\mathcal{M} =_{initial} \hat{\mathcal{M}}_{final}, \tag{4.8}$$

with

$$\begin{array}{c} initial = \tau_{-3}\ \tau_{-2} \\ final = \tau_{m-2}\ \tau_{m-1} \end{array}, \tag{4.9}$$

where we note that $\hat{\mathcal{M}}$ instead of $\mathcal{M}$ is used in order to hold the initial and terminal conditions in (4.3) in the reconstructed compact planar B-splines. Then, letting $V(G)$ and $E(G)$ be the set of vertex and the set of edge respectively, they are given as

$$\begin{cases} V(G) = \{v_i \mid -1 \leq i \leq m-3\} \\ E(G) = \{v_i, v_j) \mid -1 \leq i < j \leq m-3\} \end{cases}. \tag{4.10}$$

Here, v_i of $V(G)$ corresponds to the i-th control point, i.e., v_i for $i = 0, \cdots, m-3$, and hence (v_i, v_j) of $E(G)$ denotes a polygonal line from τ_i to τ_j in $\hat{\mathcal{M}}$. The DAG will be constructed to have a unique source (a vertex with no inbound edge) and a unique sink (a vertex with no outbound edge). The source corresponds to the $initial$ point, and the sink corresponds to the $final$ point.

Also, letting $\gamma(v_i, v_j)(\geq 0)$ be a weight of each edge between τ_i and τ_j, we define $\gamma(v_i, v_j)$ by introducing an idea so-called multi-level error functions as

$$\gamma(v_i, v_j) = \sum_{l=0}^{l_{\max}} \|(t_i, x^{(l)}(t_i))^T - (t_j, x^{(l)}(t_j))^T\|_{\Lambda_l}^2 \quad , \tag{4.11}$$

where $||z||_S^2 = z^T S z$ for row vector z, and $\Lambda_l = \mathrm{diag}\{\lambda_1^l, \lambda_2^l\} \in \mathbf{R}^{2\times 2}$ with $\lambda_1^l, \lambda_2^l(\geq 0)$ for $l = 0, 1, \cdots, l_{max}(< 3)$ is a weight matrix to control the balance among the approximations on the each level of $x^{(l)}(t), l = 0, 1, \cdots, l_{max}$.

Noting that the multi-level error functions $\gamma(v_i, v_j)$ in (4.11) can be expressed in terms of control points τ_i using first and second derivatives (i.e., $x^{(1)}(t_i)$ and $x^{(2)}(t_i)$) in (4.4). Then, we readily see that the error functions can be computed in a straightforward manner.

As is well-known, the DAG G is constructed to have a unique source and a unique sink corresponding to τ_{-1} and τ_{m-3} respectively. Thus, our task is to find a path on G from the source τ_{-1} to the sink τ_{m-3} consisting of K vertices (i.e., dominant control points) with minimum total weight of $\gamma(v_i, v_j)$ in (4.11), where K is preset. It can be shown that such a path exists and can be found by using the dynamic programming (DP) as follows.

Now, let $D_{K,j}$ be a minimum total weight of the path from the source v_0 to vertex v_j including K

vertices. Then, we initially set for the case of $K = 2$ as

$$D_{2,j} = \begin{cases} \infty & if \;\; j = -1 \\ \gamma(v_{-1}, v_j) & if \;\; -1 < j \leq m-3 \end{cases} . \tag{4.12}$$

For the case of $K \geq 3$, we can iteratively compute $D_{K,j}$ as

$$D_{K,j} = \begin{cases} min_{K-2 \leq i < j}\{D_{K-1,i} + \gamma(v_i, v_j)\} & if \;\; j \geq K-1 \\ \infty & \text{otherwise} \end{cases} . \tag{4.13}$$

In order to optimally select K-dominant control points, we thus have only to compute $D_{K,m-3}$, in which the minimum cumulative error of multi-level error functions with K-dominant control points is found. The complete algorithm to select the K-dominant control points is shown in Algorithm 1.

We have presented an algorithm to select a subset of points to optimally approximate compact B-spline curves. In particular, it is able to find an approximation with a specified number of points and providing the minimum cumulative error. The algorithm is based on dynamic programming, and it is independent of the choice of the compatible error function.

Then, letting $\hat{\tau} \in \mathbf{R}^{2\times K}$ be a matrix corresponding to the path computed by the above method using DP, we get the compressed data as a matrix $\bar{\tau} \in \mathbf{R}^{2\times \bar{M}}$ with $\bar{M} = k + 4 \;\; (< M)$:

$$\bar{\tau} = \left[\tau_{-3} \; \tau_{-2} \; \hat{\tau} \; \tau_{m-2} \; \tau_{m-1}\right]. \tag{4.14}$$

In the sequel, the control polygon corresponding to $\bar{\tau}$ will be refereed as $\bar{\mathcal{M}}$.

4.3.2 Representing Compact Planar B-spline Curves using Non-Uniform Rational B-splines (NURBS)

We are now in the position to develop a method for representing compact planar B-spline curves using the selected control polygon $\bar{\mathcal{M}}$, equivalently control point matrix $\bar{\tau}$ in (4.14).

Letting $\bar{x}(t)$ be compact planar B-spline curves defined as

$$\bar{x}(t) = [\bar{X}(t) \; \bar{Y}(t)]^T \in \mathbf{R}^2, \quad t \in [t_0, t_m], \tag{4.15}$$

Algorithm 1 Selecting the K-dominant control points for compact planar B-spline curve

Input : The control points from given data $n(\geq 2)$
Input : The specified number of $K(2 \leq K \leq n)$
Output : The indices of the K points

```
begin
    // The indices of the $K$ points
    $S$ matrix
    // The minimum weight table
    $D \leftarrow (K+1) \times n$ matrix
    // Path
    $P \leftarrow (K+1) \times n$ matrix
    // Initialization
    for $j \leftarrow 1$ to $n-1$ do
        $D_{2,j} \leftarrow \gamma(v_{-1}, v_j)$
        $P_{2,j} \leftarrow 0$
    end for
    // Compute the rest of $D$
    for $m \leftarrow 3$ to $K$ do
        for $j \leftarrow m-1$ to $n-1$ do
            $min\ weight \leftarrow \infty$
            for $i \leftarrow m-2$ to $j-1$ do
                $weight \leftarrow D_{m-1,i} + \gamma(v_i, v_j)$
                $prior\ vertex\ index \leftarrow 0$
                if $weight < min\ weight$ then
                    $min\ weight \leftarrow weight$
                    $prior\ vertex\ index \leftarrow i$
                end if
            end for
            $D_{m,j} \leftarrow min\ weight$
            $P_{m,j} \leftarrow prior\ vertex\ index$
        end for
    end for
    // Restore the Path
    $vertex\ index \leftarrow n-1$
    for $i \leftarrow 0$ to $K-1$ do
        $S \leftarrow S \cup \{vertex\ index\}$
        $vertex\ index \leftarrow P_{K-i,vertex\ index}$
    end for
    return $S$
end
```

we then consider to represent $\bar{x}(t)$ from the selected control points in (4.14) such that

$$\bar{x}(t) \approx x(t),\ \forall t \in [t_0, t_m]. \tag{4.16}$$

For the sake of simplicity, we here represent each element of $\bar{x}(t)$ in (4.15), independently. Now, let $\bar{t}_i, i = -3, -2, \cdots, \bar{m}-1$ be a knot point corresponding to each control point of $\bar{\tau} \in \mathbf{R}^{\bar{M}}$, where $\bar{m} = \bar{M} - 3$. Hence, the intervals between such knot points may be non-uniform, we then consider to represent $\bar{\tau}$ as curves $\bar{x}(t), t \in [t_0, t_m]$ by employing cubic NURBS (see e.g., [49] and [50]) as

$$\bar{x}(t) = \frac{\sum_{i=-3}^{m-1} w_i \bar{\tau}_i B_3(\alpha_i(t - \bar{t}_i))}{\sum_{i=-3}^{m-1} w_i B_3(\alpha_i(t - \bar{t}_i))}. \tag{4.17}$$

Here, $B_{i,k}(\cdot)$ denotes B-splines functions, and it can be computed by algorithm in Chapter 2. Also, $w_i(\geq 0)$ is a weight coefficient with

$$\sum_{i=-3}^{m-1} w_i = 1. \tag{4.18}$$

Thus, letting $w \in \mathbf{R}^{\bar{M}}$ be a vector of weight $w_i, i = -3, -2, \cdots, \bar{m}-1$ as

$$w = [w_{-3}\ w_{-2}\ \cdots\ w_{\bar{m}-1}]^T, \tag{4.19}$$

our task is only to compute a vector w such that $\bar{x}(t) \approx x(t)$. We then consider the following Problem 2.

Problem 2 : Find $w \in \mathbf{R}^{\bar{M}}$ such that

$$\min_{w \in \mathbf{R}^{\bar{M}}} \sum_{i=1}^{N} ||x(s_i) - \bar{x}(s_i)||^2, \tag{4.20}$$

subject to B-spline functions and $w_i \geq 0,\ \forall i$.

Here, $s_i \in [t_0, t_m],\ i = 1, 2, \cdots, N$ denotes a equally spaced sampled point defined as

$$s_i = (i-1)\Delta s, \tag{4.21}$$

with $\Delta s > 0$. It can be shown that this problem is written as non-linear programming problem with equality and inequality linear constraints in terms of w. For solving Problem 2, we here used Matlab function ‘fmincon’.

4.4 Experimental Study

We demonstrate the performance of our proposed method by experimental studies. In particular, we here apply our method to the design of characters based on the dynamic font method.

Now we suppose that we are given planar cubic B-spline curves $x(t)$ and the corresponding control polygons $\mathcal{M}$ for three characters “hello”, “fukuoka” and “welcome” as shown in the Figure 4.1, respectively. Here, the red dashed lines with circles in the Figure 4.1 (i.e., Figure 4.1 (a)-(c)) were the control polygon $\mathcal{M}$. These curves $x(t)$ and control polygon $\mathcal{M}$ are obtained based on the design method of a dynamic font for the handwriting (see e.g., [51]). Specifically, a set of handwriting data (green asterisks in Figure 4.1) is measured by the pen-tablet device and stored in a PC. We then construct the control polygon $\mathcal{M}$ using the theory of smoothing splines [41]. Here, α is set as $\alpha = 10$. Hence, the total number of control points $\mathcal{M}$ and the time interval $[t_0 \ \ t_m]$ are given as Table 4.1.

Using the proposed method in this Chapter, we design compact planar B-spline curves $\bar{x}(t)$ from the given $x(t)$. In Problem 2, Δs is set as $\Delta s = 0.02$, and hence N is set as $N = 751, 1083, 707$ for the cases of “hello”, “fukuoka” and “welcome”, respectively. Their design examples are illustrated in Figures 4.2-4.19. Here, $l_{\max}$ in (4.11) is set as $l_{\max} = 0, 1$ and 2 respectively with $\Lambda_0 = I_2, \Lambda_1 = 10I_2$ and $\Lambda_2 = 10^2 I_2$, where $I_2 \in \mathbf{R}^{2\times 2}$ denotes an identity matrix.

Also, in each result of Figures 4.2-4.19 are respectively results for the cases where the selected control point K is set as Table 4.2. In Figures 4.2(a)-4.19(a), the red and blue lines denote the given curves $x(t)$ and the compact planar B-spline curves $\bar{x}(t)$. The corresponding control polygons $\mathcal{M}$ (red dashed lines with circle marks) and $\bar{\mathcal{M}}$ (blue dashed lines with cross marks) are plotted in Figures 4.2(b)-4.19(b). Also, in Figures 4.2(c)-4.19(c), the weights w_i computed in Problem 2 are plotted for X and Y-axes in red and blue lines with square marks, respectively.

From these results, we see that our proposed method relatively works well for the case of $l_{max} = 2$. In the case of $l_{max} = 0$, noting that the control polygon $\mathcal{M}$ represents the geometrical outline of curves $x(t)$ in B-spline functions, we see that the control points are selected such that there is a discrepancy between $\mathcal{M}$ and $\hat{\mathcal{M}}$ from using the dynamic programming. Such a selection strategy may often cause that a linear sequence of control points is unselected intensively (see e.g., Figures 4.2(a), 4.5(a), 4.8(a), 4.11(a), 4.14(a) and 4.17(a)). Then, the knot point interval corresponding to the unselected control points becomes wider. Hence, representing a compact planar B-spline curve in a class of cubic NURBS may become difficult only by adjusting the weights w_i shown (see e.g., Figures 4.2(c), 4.5(c), 4.8(c), 4.11(c), 4.4(c) and 4.17(c)). As l_{max} in (4.11) becomes large, the unselected control points become scattered as shown in Figures 4.3-4.4, 4.6-4.7 for the case of "hello", Figures 4.9-4.10, 4.12-4.13 for the case of "fukuoka" and Figures 4.15-4.16, 4.18-4.19 for the case of "welcome". We then may observe that the approximation gets better.

On the other hand, as the number of selected control points K (or $\bar{\mathcal{M}}$) becomes small, we obviously see that the approximation may get worse. We then compute the root mean squared error (RMSE) defined as

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\|x(s_i) - \bar{x}(s_i)\|^2}, \tag{4.22}$$

for all the case of K, i.e. $0 \leq K \leq M - 4$ and their results for $l_{max} = 0, 1, 2$ are plotted in Figure 4.20. From these results, we observe that the approximations for the case of $l_{max} = 2$ are better than the cases of $l_{max} = 0$ and 1, but we cannot avoid that the approximation gets worse as the number of selected control points K (or $\bar{\mathcal{M}}$) becomes small.

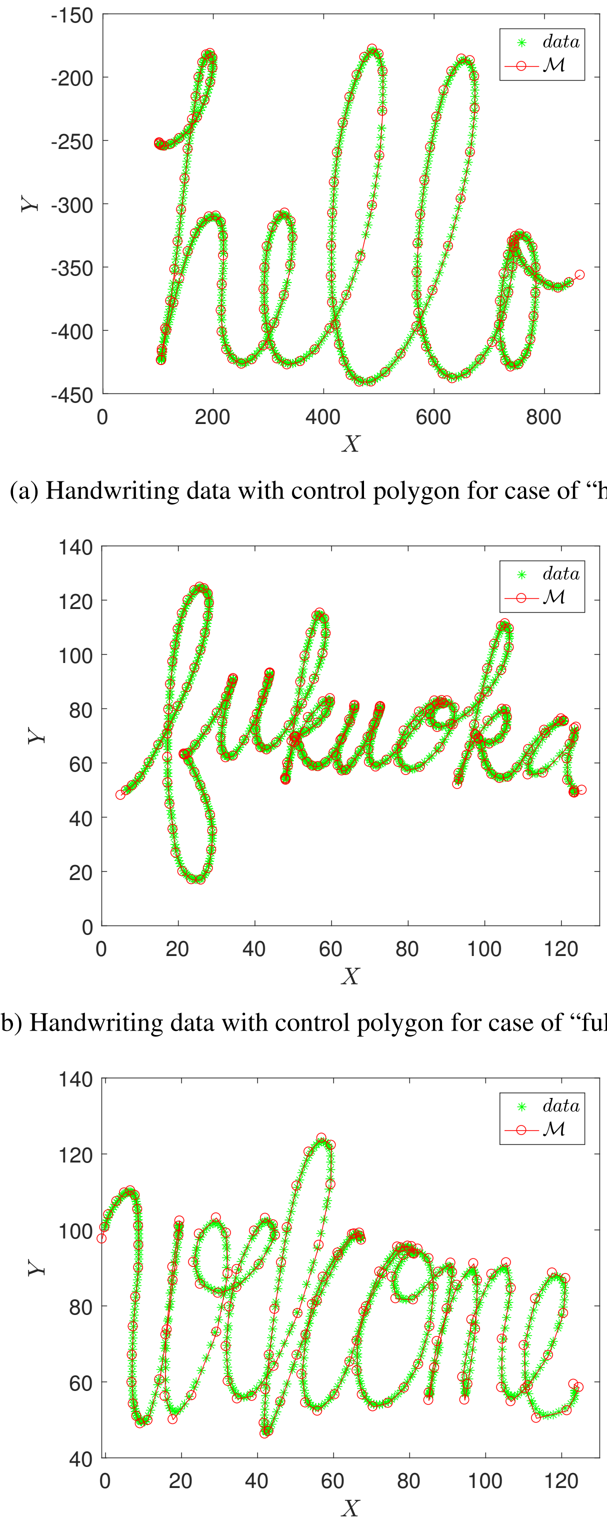
In addition, we here compare the above results with the case where the control points with a specified number of K are selected randomly. Setting K as $K = 138, 196$ and 129 for the cases of "hello", "fukuoka" and "welcome" respectively, we repeatedly construct the planar B-spline curves $\bar{x}(t)$ 100 times and their results are plotted in Figure 4.21. We see that these results indicate that randomly selecting the dominant control points often leads to the unstable reconstruction of compact planar curves. Comparing these results with the results in Figures 4.4, 4.10, and 4.16, it is obvious that our proposed method is more effective than the cases of random control point selection.

Table 4.1: The total number of control points M and the time interval $[t_0, t_m]$ for the cases of “hello”, “fukuoka” and “welcome”.

Character	M	$[t_0, t_m]$
hello	153	$[0, 15]$
fukuoka	218	$[0, 21.5]$
welcome	142	$[0, 14]$

Table 4.2: Setup on the number of selected control points K ($\bar{M}$) for the results of “hello”, “fukuoka” and “welcome” in a result for Figures 4.2-4.19.

Character	Figure Number	K ($\bar{M}$)
hello	2–4	K=139 ($\bar{M}$=143)
	5–7	K=121 ($\bar{M}$=125)
fukuoka	8–10	K=197 ($\bar{M}$=201)
	11–13	K=171 ($\bar{M}$=175)
welcome	14–16	K=130 ($\bar{M}$=134)
	17–19	K=113 ($\bar{M}$=117)

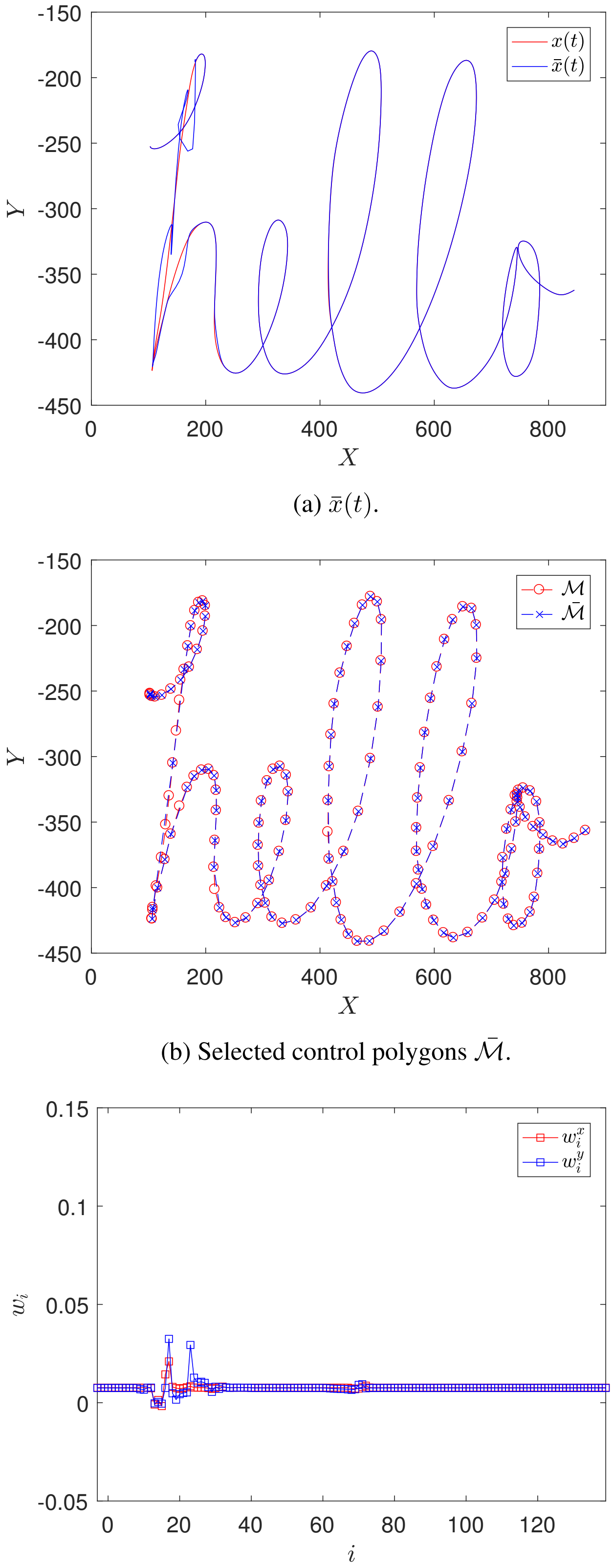


(a) Handwriting data with control polygon for case of “hello”.

(b) Handwriting data with control polygon for case of “fukuoka”.

(c) Handwriting data with control polygon for case of “welcome”.

Figure 4.1: Given planar data curves represented on $O - XY$ plane and its control polygon $\mathcal{M}$ with a set of handwriting.

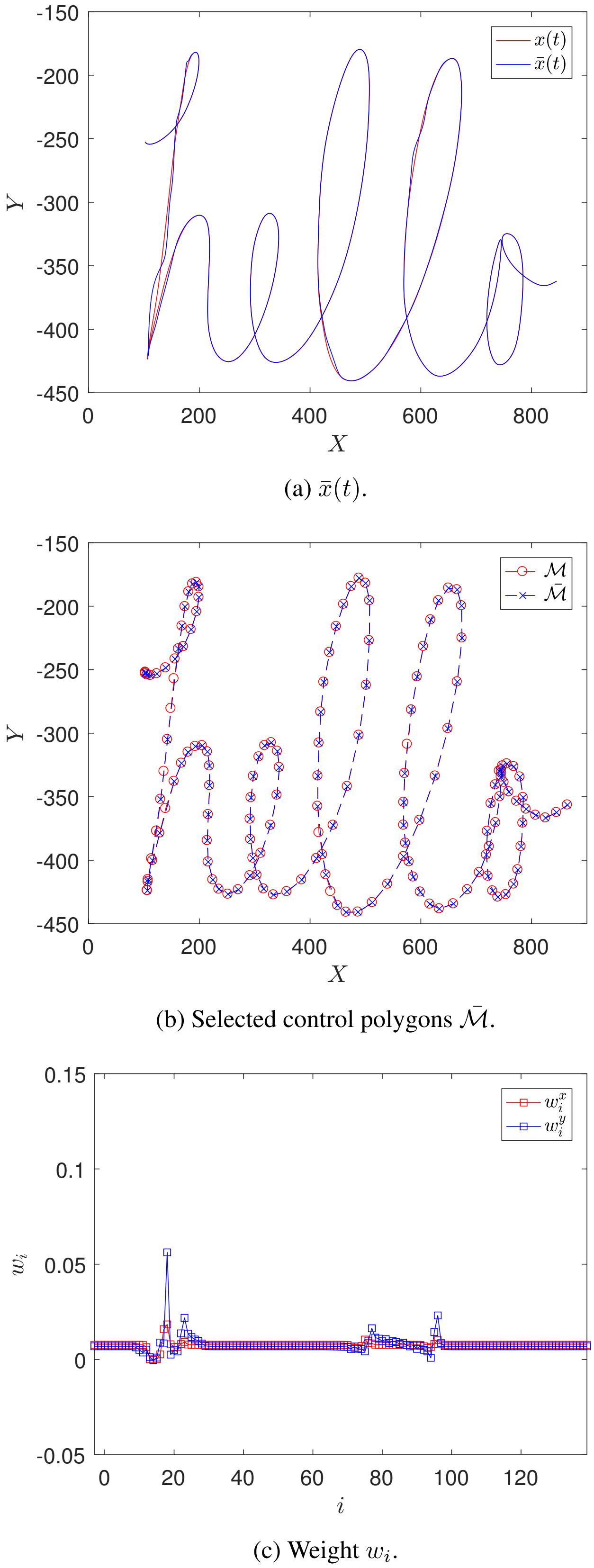


(a) $\bar{x}(t)$.

(b) Selected control polygons $\bar{\mathcal{M}}$.

(c) Weight w_i.

Figure 4.2: Compact planar B-spline curves for the case of "hello" with $l_{max} = 0, K = 139$ ($\bar{M} = 143$).

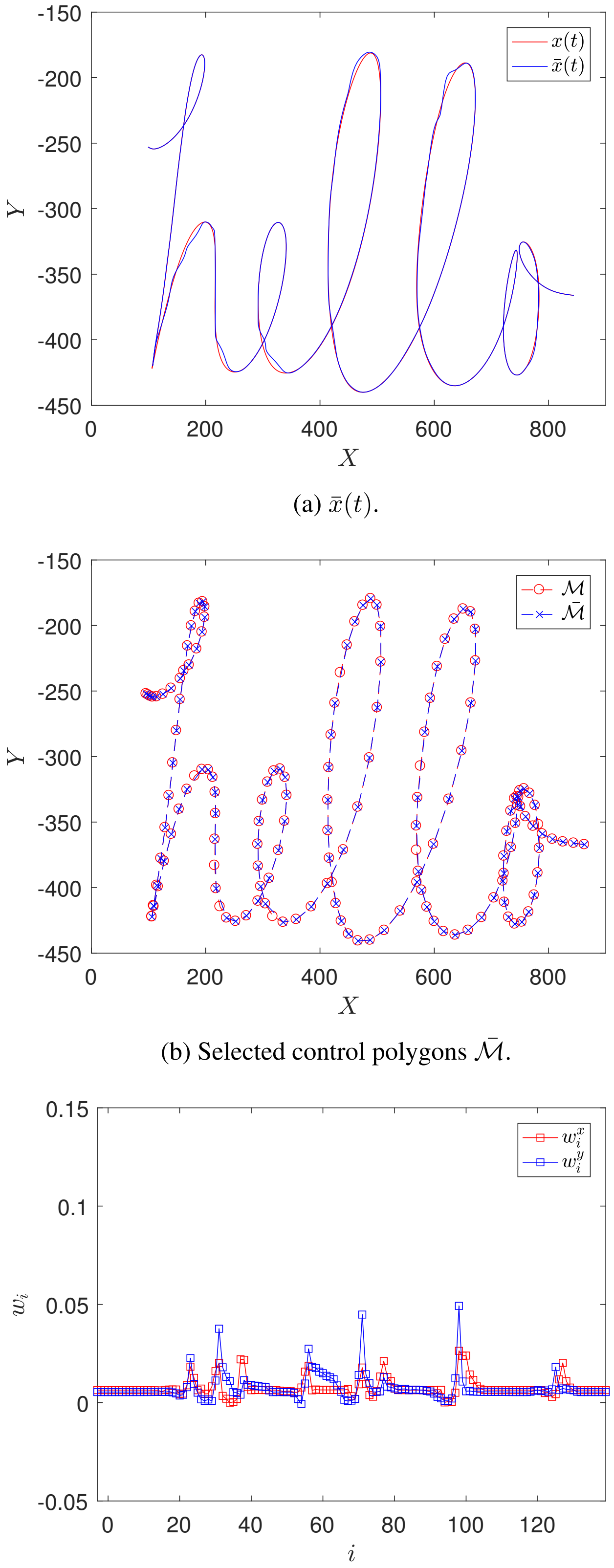


(a) $\bar{x}(t)$.

(b) Selected control polygons $\bar{\mathcal{M}}$.

(c) Weight w_i.

Figure 4.3: Compact planar B-spline curves for the case of “hello” with $l_{max} = 1, K = 139$ ($\bar{M} = 143$).

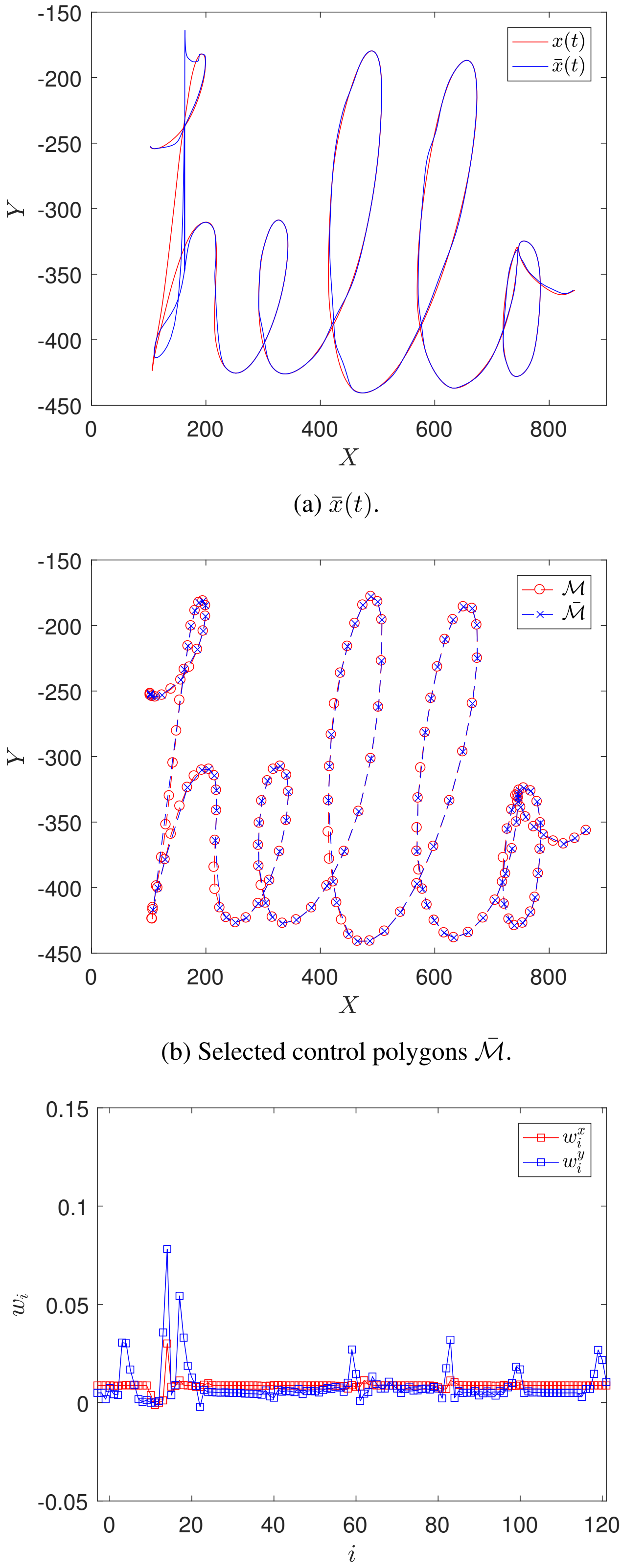


(a) $\bar{x}(t)$.

(b) Selected control polygons $\bar{\mathcal{M}}$.

(c) Weight w_i.

Figure 4.4: Compact planar B-spline curves for the case of "hello" with $l_{max} = 2$, $K = 139$ ($\bar{M} = 143$).

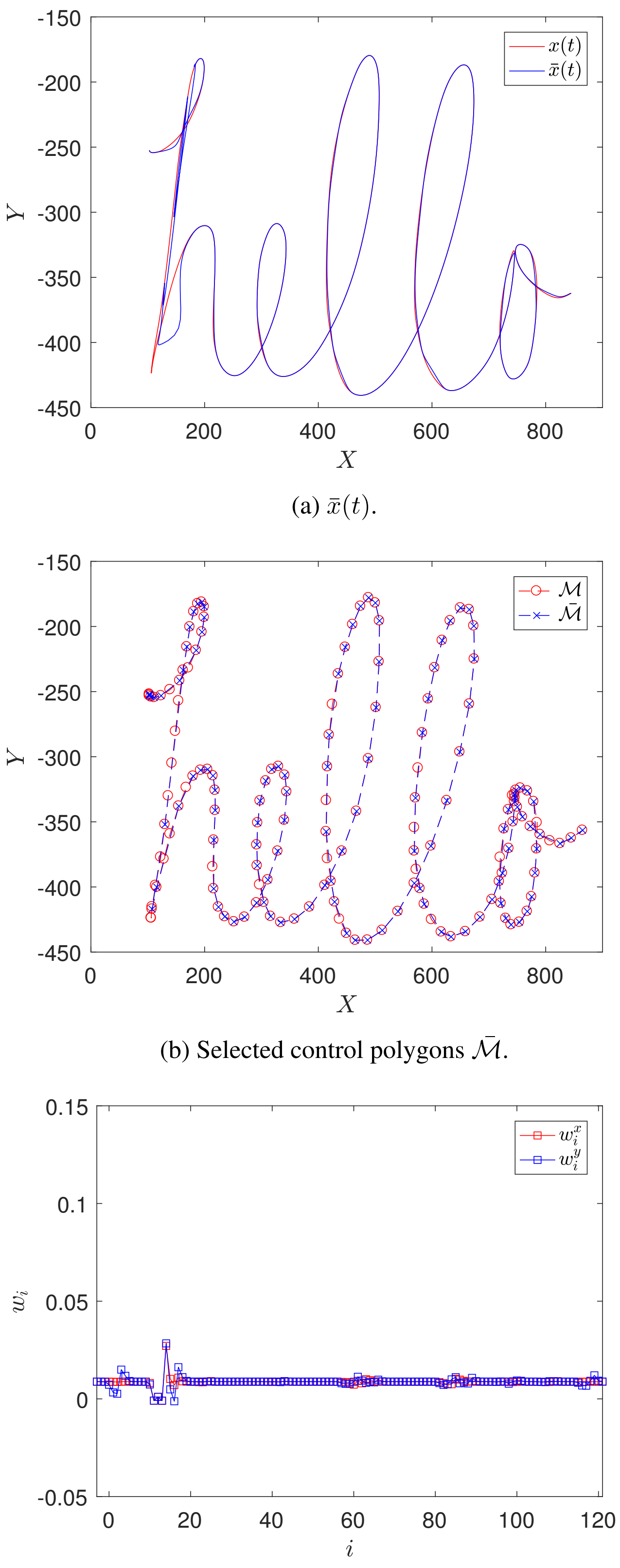


(a) $\bar{x}(t)$.

(b) Selected control polygons $\bar{\mathcal{M}}$.

(c) Weight w_i.

Figure 4.5: Compact planar B-spline curves for the case of "hello" with $l_{max} = 0, K = 121$ $(\bar{M} = 125)$.

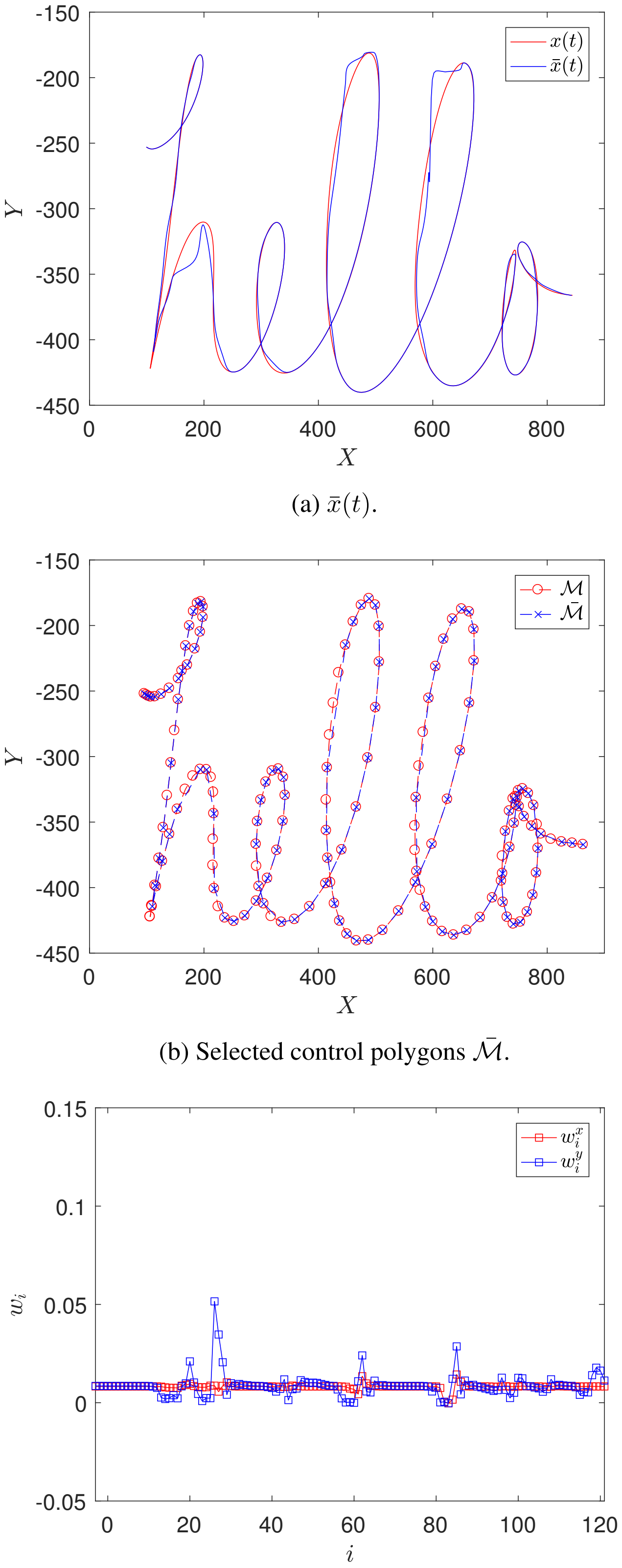


(a) $\bar{x}(t)$.

(b) Selected control polygons $\bar{\mathcal{M}}$.

(c) Weight w_i.

Figure 4.6: Compact planar B-spline curves for the case of "hello" with $l_{max} = 1, K = 121$ ($\bar{M} = 125$).

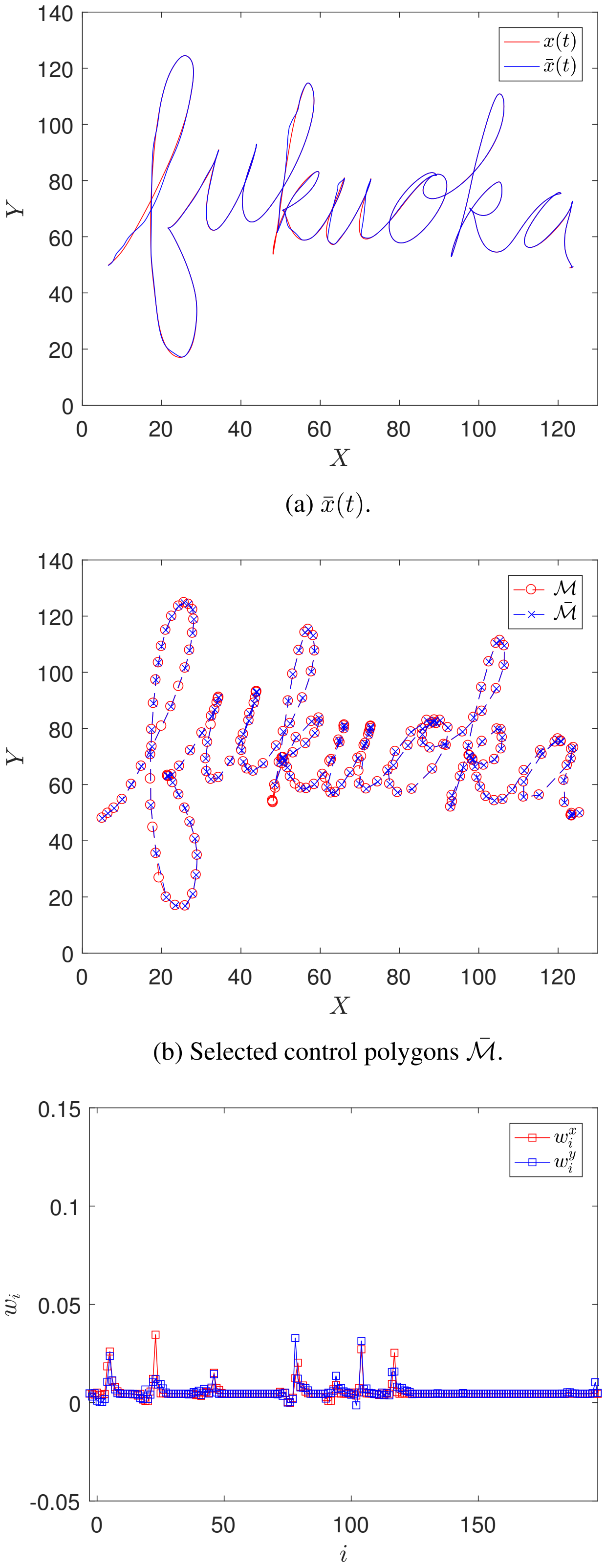


(a) $\bar{x}(t)$.

(b) Selected control polygons $\bar{\mathcal{M}}$.

(c) Weight w_i.

Figure 4.7: Compact planar B-spline curves for the case of "hello" with $l_{max} = 2, K = 121$ ($\bar{M} = 125$).

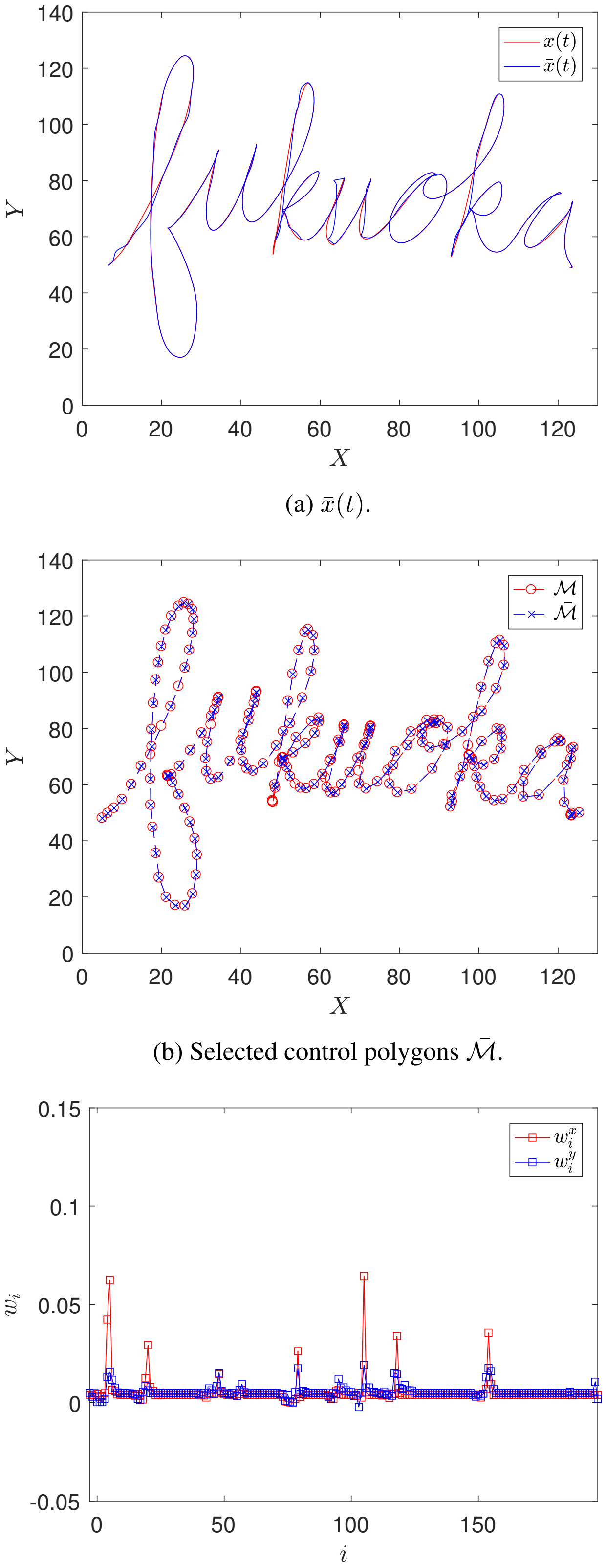


(a) $\bar{x}(t)$.

(b) Selected control polygons $\bar{\mathcal{M}}$.

(c) Weight w_i.

Figure 4.8: Compact planar B-spline curves for the case of "fukuoka" with $l_{max} = 0, K = 197$ ($\bar{M} = 201$).

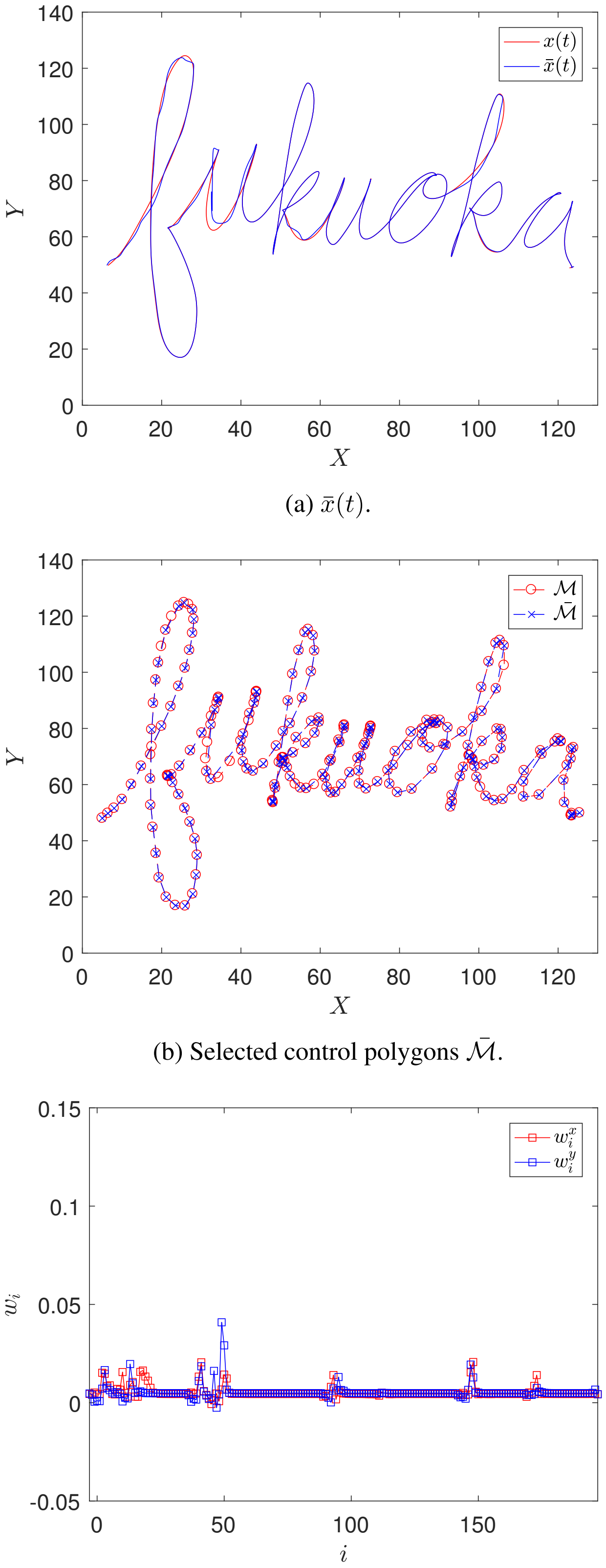


(a) $\bar{x}(t)$.

(b) Selected control polygons $\bar{\mathcal{M}}$.

(c) Weight w_i.

Figure 4.9: Compact planar B-spline curves for the case of "fukuoka" with $l_{max} = 1, K = 197$ ($\bar{M} = 201$).

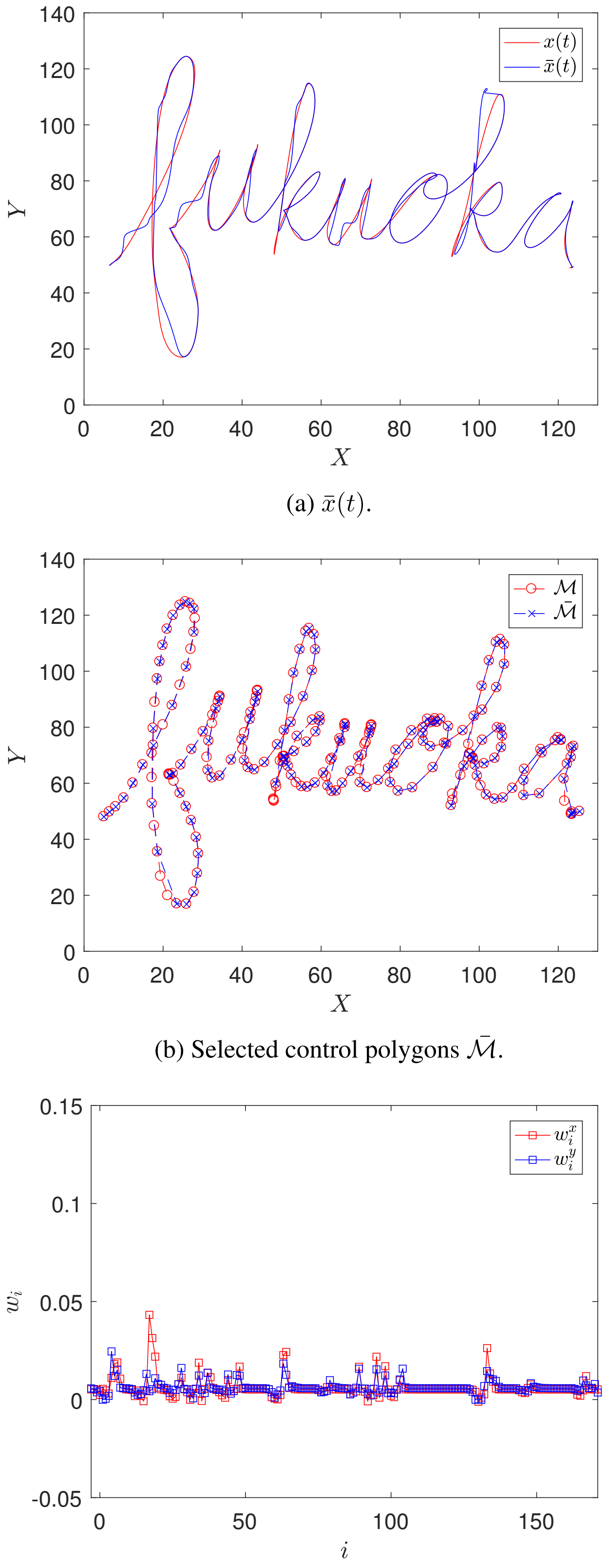


(a) $\bar{x}(t)$.

(b) Selected control polygons $\bar{\mathcal{M}}$.

(c) Weight w_i.

Figure 4.10: Compact planar B-spline curves for the case of "fukuoka" with $l_{max} = 2, K = 197$ ($\bar{M} = 201$).



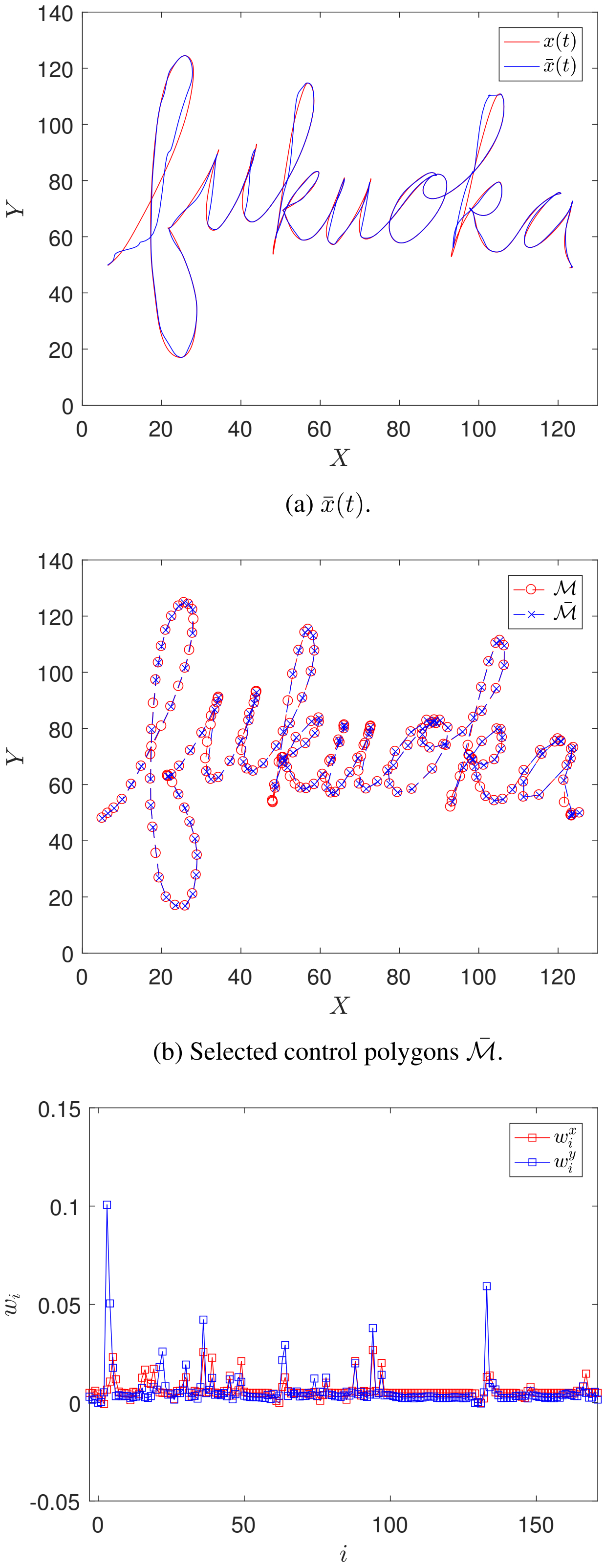
(a) $\bar{x}(t)$.

(b) Selected control polygons $\bar{\mathcal{M}}$.

(c) Weight w_i.

Figure 4.11: Compact planar B-spline curves for the case of “fukuoka” with $l_{max} = 0, K = 171$ ($\bar{M} = 175$)

.

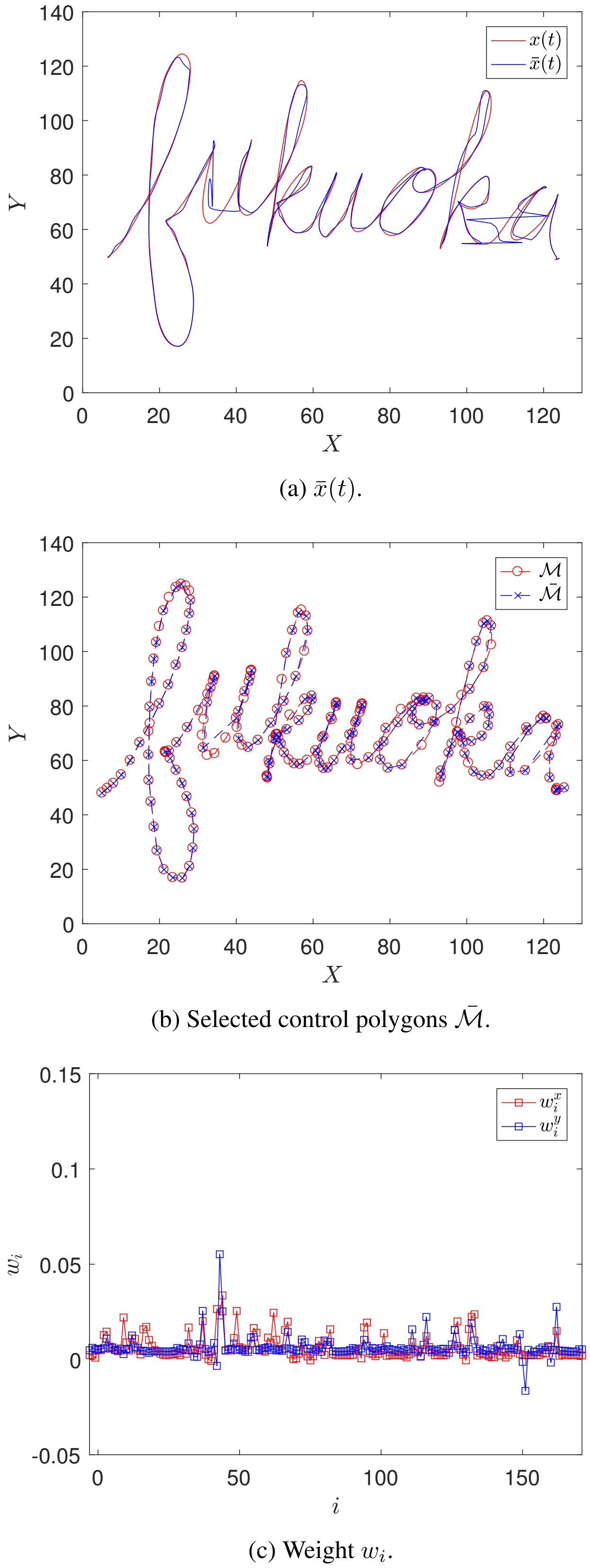


(a) $\bar{x}(t)$.

(b) Selected control polygons $\bar{\mathcal{M}}$.

(c) Weight w_i.

Figure 4.12: Compact planar B-spline curves for the case of "fukuoka" with $l_{max} = 1, K = 171$ ($\bar{M} = 175$).

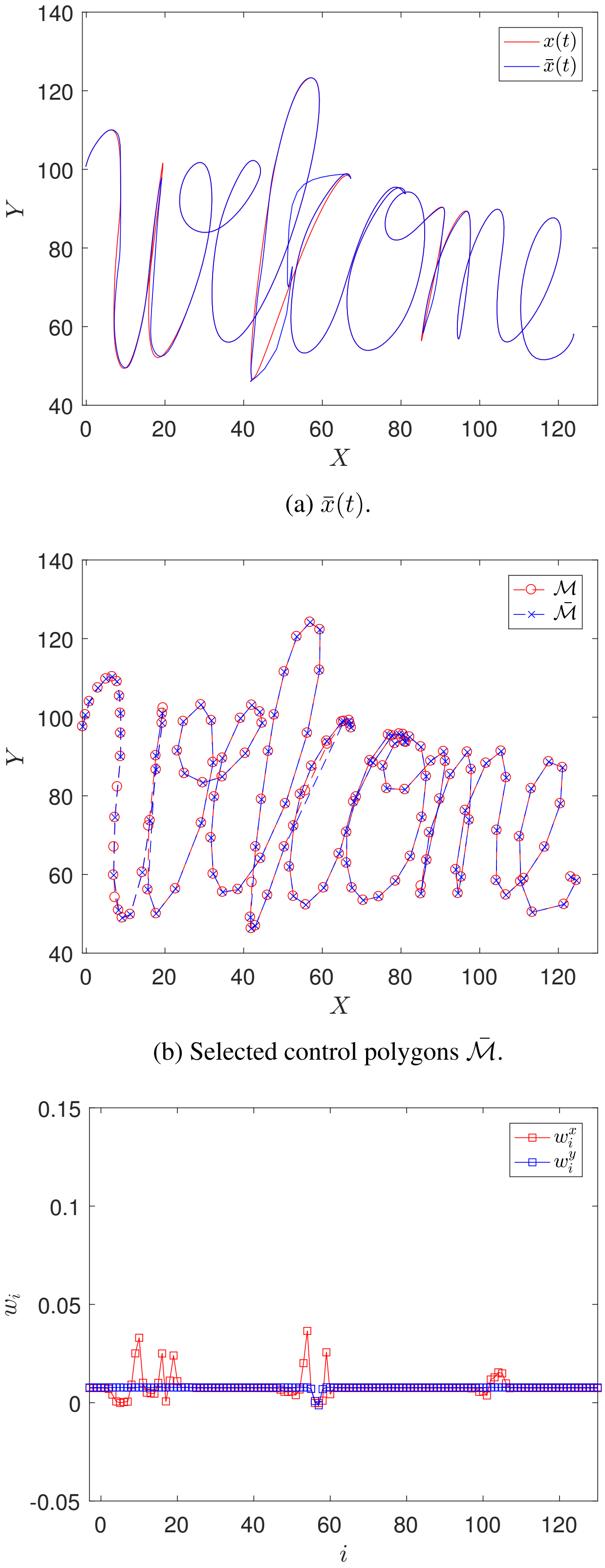


(a) $\bar{x}(t)$.

(b) Selected control polygons $\bar{\mathcal{M}}$.

(c) Weight w_i.

Figure 4.13: Compact planar B-spline curves for the case of "fukuoka" with $l_{max} = 2, K = 171$ ($\bar{M} = 175$).

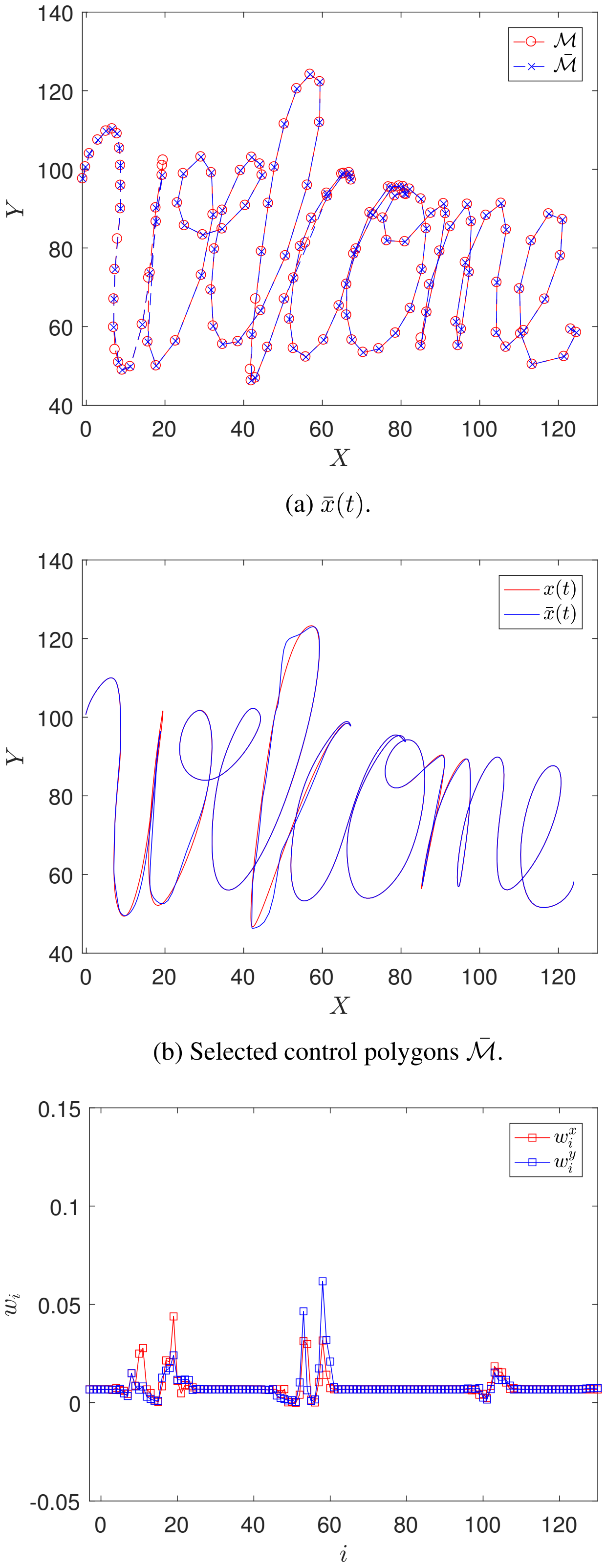


(a) $\bar{x}(t)$.

(b) Selected control polygons $\bar{\mathcal{M}}$.

(c) Weight w_i.

Figure 4.14: Compact planar B-spline curves for the case of "welcome" with $l_{max} = 0, K = 130$ ($\bar{M} = 134$).

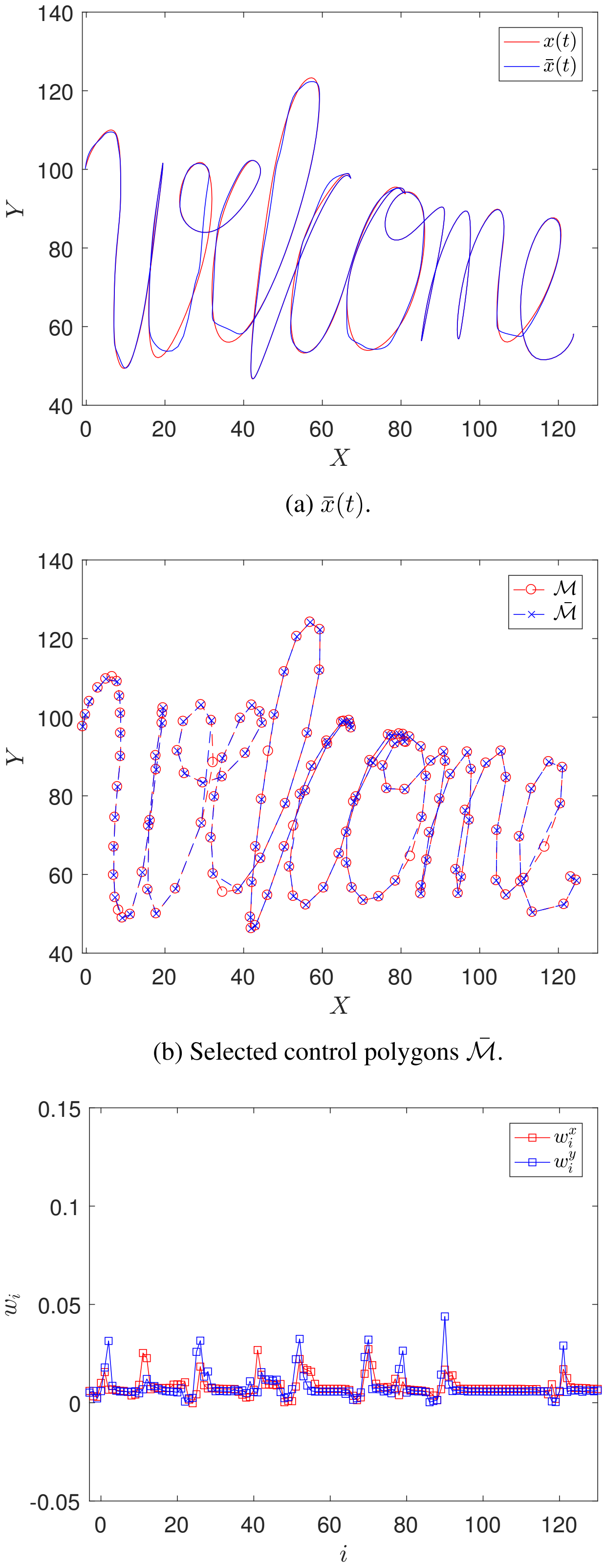


(a) $\bar{x}(t)$.

(b) Selected control polygons $\bar{\mathcal{M}}$.

(c) Weight w_i.

Figure 4.15: Compact planar B-spline curves for the case of "welcome" with $l_{max} = 1, K = 130$ ($\bar{M} = 134$).

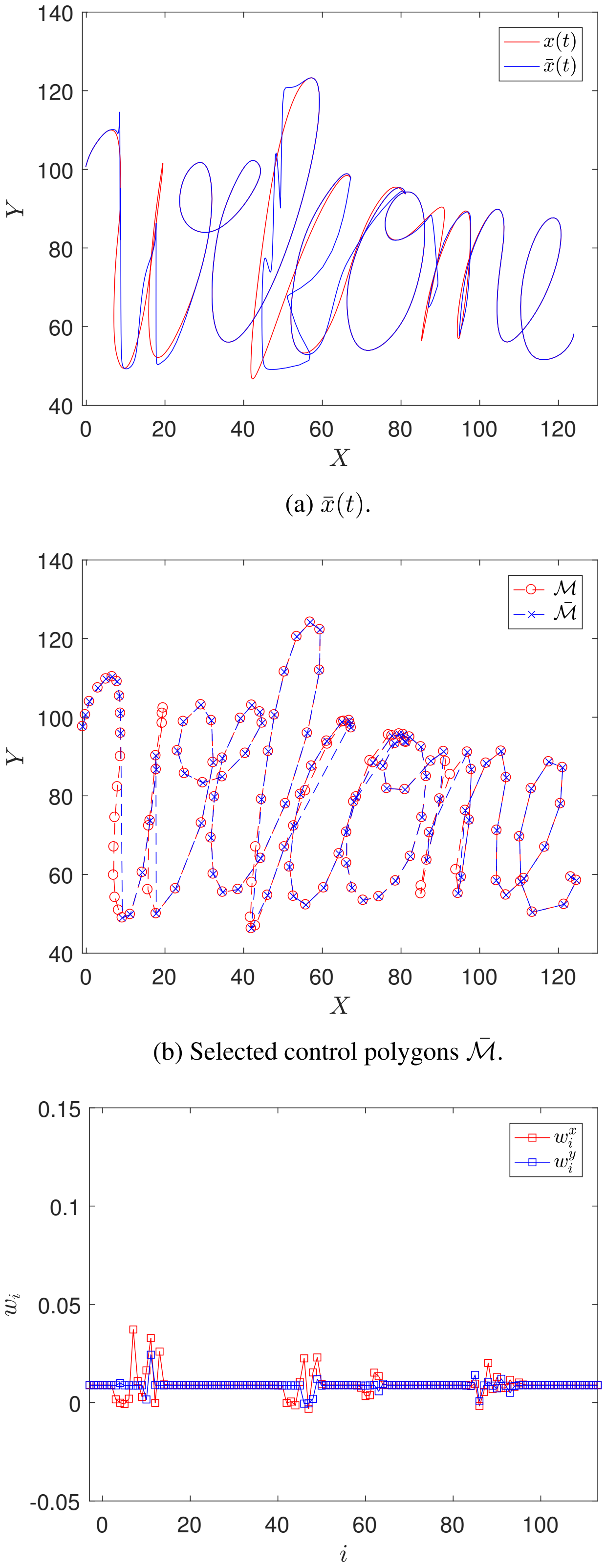


(a) $\bar{x}(t)$.

(b) Selected control polygons $\bar{\mathcal{M}}$.

(c) Weight w_i.

Figure 4.16: Compact planar B-spline curves for the case of “welcome” with $l_{max} = 2, K = 130$ ($\bar{M} = 134$).

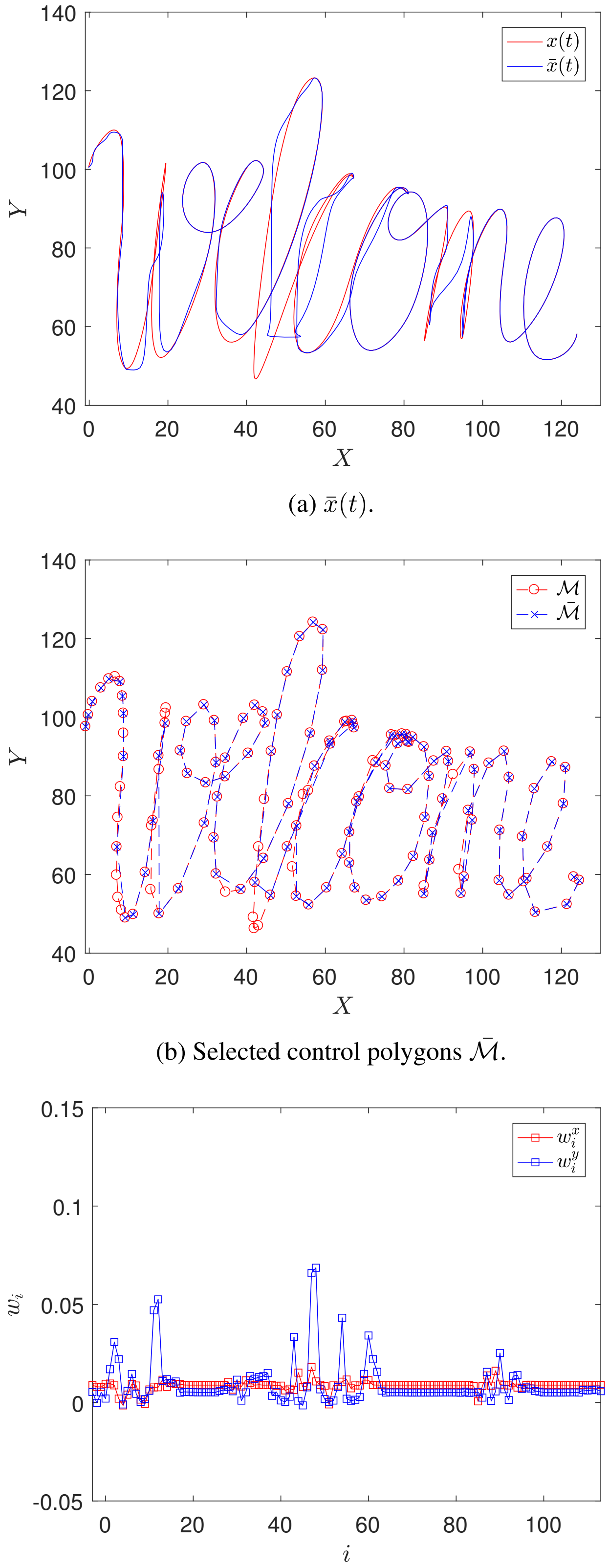


(a) $\bar{x}(t)$.

(b) Selected control polygons $\bar{\mathcal{M}}$.

(c) Weight w_i.

Figure 4.17: Compact planar B-spline curves for the case of “welcome” with $l_{max} = 0, K = 113$ ($\bar{M} = 117$).

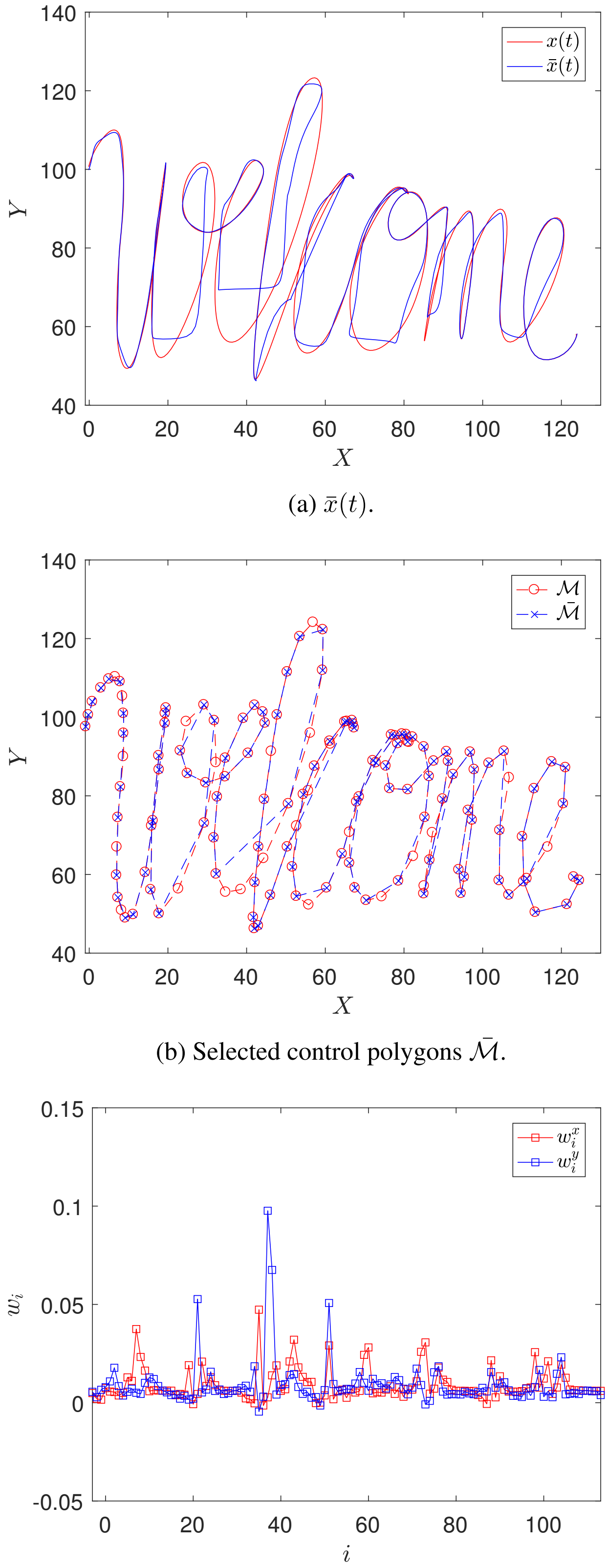


(a) $\bar{x}(t)$.

(b) Selected control polygons $\bar{\mathcal{M}}$.

(c) Weight w_i.

Figure 4.18: Compact planar B-spline curves for the case of "welcome" with $l_{max} = 1, K = 113$ ($\bar{M} = 117$).

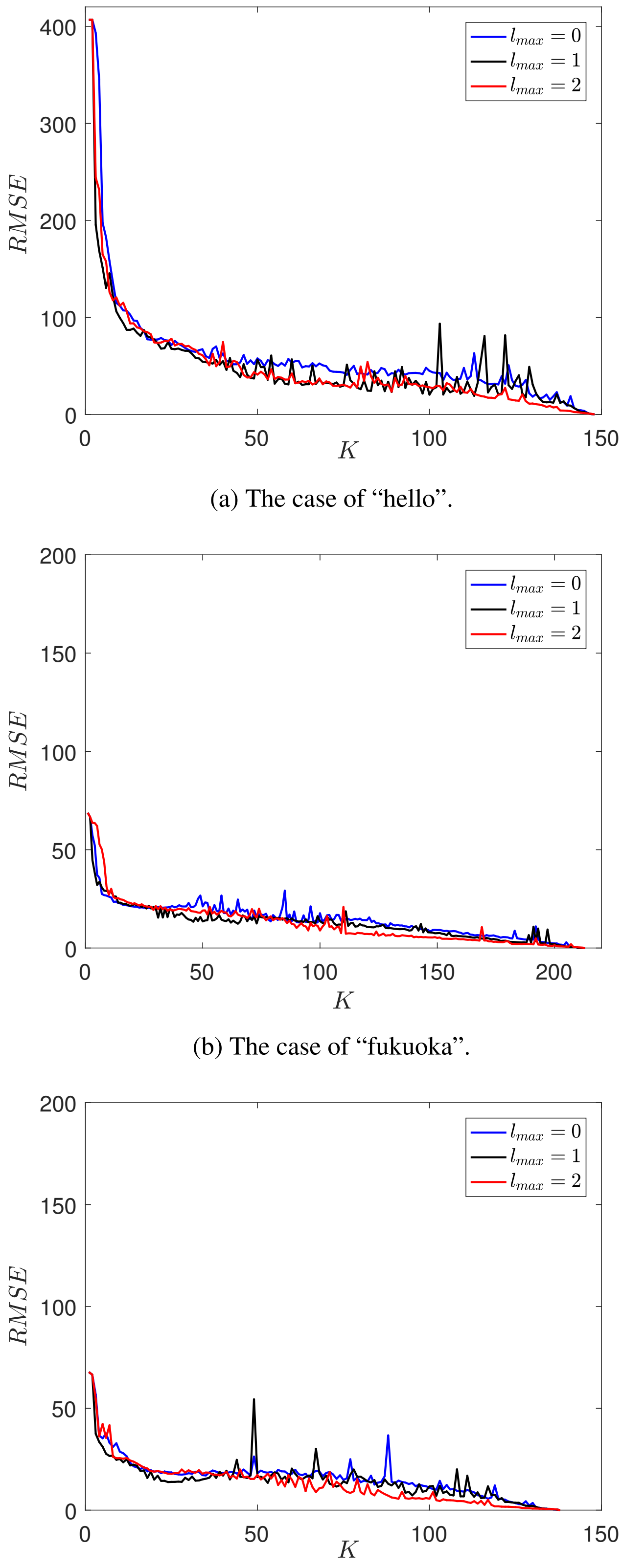


(a) $\bar{x}(t)$.

(b) Selected control polygons $\bar{\mathcal{M}}$.

(c) Weight w_i.

Figure 4.19: Compact planar B-spline curves for the case of “welcome” with $l_{max} = 2, K = 113$ ($\bar{M} = 117$).

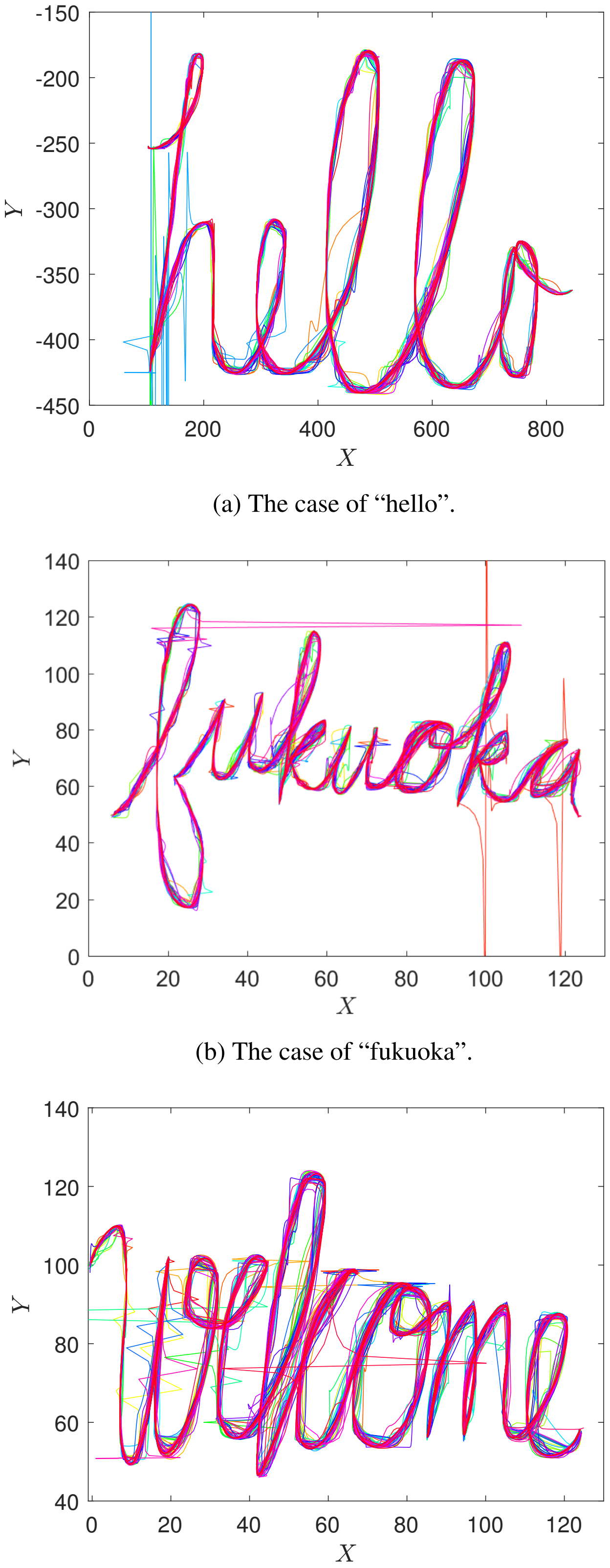


(a) The case of “hello”.

(b) The case of “fukuoka”.

(c) The case of “welcome”.

Figure 4.20: RMSE for all the case of $K, l_{max} = 0, 1$ and 2.



(a) The case of “hello”.

(b) The case of “fukuoka”.

(c) The case of “welcome”.

Figure 4.21: Compact planar B-spline curves $\bar{x}(t)$ for the cases of “hello”, “fukuoka”, and “welcome”, where the control points are randomly selected.

Chapter 5

A Design of Compact Periodic B-splines Curves

5.1 Introduction

We here focus on the problem of object contour modeling as an application of periodic B-spline curves. The problem on contour modeling of objects may arise in many fields - such as image processing, robotics, and others. Hence, they have been widely studied (e.g., [25], [52]). In their studies, spline functions have been used frequently, and various techniques-such as snakes and active contour model, have been developed [53]. Fujioka and Kano have also studied such contour modeling problems using the B-spline approach [26]. Therein, the contour is modeled as a result of periodic smoothing spline curves using normalized uniform B-splines as basis functions. Then, the knot points are equally spaced. Also, a sequence of control points of B-splines is called a control polygon, which represents the geometrical outline of curves (i.e., object contour). In addition, it has been shown that the computations on area and complexity of contour model are formulated as quadratic functions in terms of the vector of control points (i.e., control polygon). Thus, such a representation using a control polygon is very powerful for analyzing deformation of the contour model [31], [17]. In order to develop faster deformation analysis method on the contour model, our concern is to yield more compact periodic splines by using only the dominant control points, in which the desired approximation accuracy is achieved. We here develop a dynamic programming (DP)-based method for optimally selecting the dominant control points [44] with the idea of multi-level error functions. We then demonstrate the performance of a problem on object contour modeling.

5.2 Periodic B-spline Curves

Now, we consider to fit a periodic curve $x(t) \in \mathbf{R},\ t \in [t_0, t_m]$ to a set of data sampled from some function $f(t) \in \mathbf{R},\ t \in [t_0, t_m]$. Such a periodic curve $x(t)$ is constituted by employing normalized uniform cubic B-splines $B_3(\cdot)$ as the basis functions,

$$x(t) = \sum_{i=-k}^{m-1} \tau_i B_3(\alpha(t - t_i)), \tag{5.1}$$

subject to the periodic continuity constraints

$$x^{(l)}(t_0) = x^{(l)}(t_m),\ l = 0, 1, 2. \tag{5.2}$$

Letting $\tau \in \mathbf{R}^M$ $(M = m+3)$ be a vector consisting of "control point" $\tau_i \in \mathbf{R}(i = -3, -2, \cdots, m-1)$ as

$$\tau = [\tau_{-3}\ \tau_{-2}\ \cdots\ \tau_{m-1}]^T, \tag{5.3}$$

the periodic curve-fitting problem reduces to find an appropriate vector τ subject to the constraints (5.3) which are represented using B-splines as

$$\tau_{-3} = \tau_{m-3},\ \tau_{-2} = \tau_{m-2},\ \tau_{-1} = \tau_{m-1}. \tag{5.4}$$

It can been shown that such a τ can be obtained by employing a theory of smoothing splines (see e.g., [26] for the detail).

Meanwhile, the sequence of the obtained control points τ_i's forms a polygonal line, say control polygon $\mathcal{M}$, defined as

$$\mathcal{M} = \tau_{-3}\ \tau_{-2}\ \cdots\ \tau_{m-1}. \tag{5.5}$$

The control polygon $\mathcal{M}$ represents a geometrical outline of the periodic curve $x(t)$.

5.3 Reconstructing Compact Periodic B-spline Curves using DP Control Point Selection

As mentioned in Section 5.1, we here develop a method for reconstructing compact periodic B-spline curves. As with the case of planar B-spline curves, our main task is to represent a method for optimally selecting a set of dominant control points from the control polygon $\mathcal{M}$ on $x(t)$ in B-spline functions. We thus develop a method for optimally selecting a set of dominant control points from the control polygon $\mathcal{M}$ in (5.5). Next, we represent the compact periodic B-spline curves from the selected control points by introducing Non-Uniform Rational B-splines (NURBS). Consequently, we see that the same scheme for designing compact planar B-spline curves in Chapter 4 can be applied to this case with constraints (i.e., periodic constrains).

5.3.1 Optimal Control Point Selection using Dynamic Programming for Compact Periodic B-spline Curves

For yielding more compact periodic B-spline curves, we here optimally select $\bar{\mathcal{M}}$ dominant control points of $\mathcal{M}$. The control point selection is formulated as a graph problem and be solved by DP as follows,

Let G be a weighted Directed Acyclic Graphs (DAG). Then, the graph G is constructed from the control polygon $\hat{\mathcal{M}}$ as

$$\hat{\mathcal{M}} = \tau_{-1}\ \tau_0\ \cdots\ \tau_{m-3}, \tag{5.6}$$

which consists of $M-4$ points. Also, we regard $\hat{\mathcal{M}}$ as the main part of the original polygon $\mathcal{M}$ as

$$\mathcal{M} =_{initial} \hat{\mathcal{M}}_{final}, \tag{5.7}$$

with

$$\begin{array}{c} initial = \tau_{-3}\ , \tau_{-2} \\ final = \tau_{m-2}\ , \tau_{m-1} \end{array}. \tag{5.8}$$

Also, let $V(G)$ and $E(G)$ be a set of vertices and edges of graph G as follow in (4.10). Here, v_i of $V(G)$ corresponds to the i-th control point, i.e., τ_i for $i = -1, 0, \cdots, m-3$, and hence an element (v_i, v_j) of $E(G)$ denotes a polygonal line from τ_i to τ_j in $\mathcal{M}$. Also, letting $\gamma(v_i, v_j)(\geq\ 0)$ be a weight of each edge between τ_i and τ_j, we here use $\gamma(v_i, v_j)$ by introducing the same multi-level error function in (4.11), i.e.,

$$\gamma(v_i, v_j) = \sum_{l=0}^{l_{\max}} \|(t_i, x^{(l)}(t_i))^T - (t_j, x^{(l)}(t_j))^T\|_{\Lambda_l}^2.$$

Since the DAG G is constructed to have a unique source and a unique sink corresponding to τ_{-1} and τ_{m-3} respectively, then we have only to find a path on G from the source τ_{-1} to the sink τ_{m-3} consisting of K vertices (i.e., dominant control points) with minimum total weight of $\gamma(v_i, v_j)$, where K is preset. It can be shown that such a path exists. Thus, the dominant control points can be optimally selected by using Algorithm 1 in Chapter 4.

Then, letting $\hat{\tau} \in \mathbf{R}^K$ be a vector corresponding to the path computed by DP, we get the control point vector $\bar{\tau} \in \mathbf{R}^{\bar{M}}(\bar{M} = K + 4 \quad (< M))$ corresponding to compact B-spline curves as

$$\bar{\tau} = \left[\tau_{-3}\ \tau_{-2}\ \hat{\tau}\ \tau_{m-2}\ \tau_{m-1}\right]^T. \tag{5.9}$$

In the sequel, the control polygon corresponding to $\bar{\tau}$ will be refereed as $\bar{\mathcal{M}}$.

5.3.2 Representing Compact Periodic B-spline Curves using Non-Uniform Rational B-splines (NURBS)

Now, we have only to represent compact periodic B-spline curves $\bar{x}(t) \in \mathbf{R},\ t \in [t_0, t_m]$ by using $\bar{\tau}$ in (5.9) such that $\bar{x}(t) \approx x(t),\ \forall t \in [t_0, t_m]$. Since the intervals between their knot points may be non-uniform, representing $\bar{\tau}$ to curves $\bar{x}(t),\ t \in [t_0, t_m]$ can be achieved by using cubic Non-Uniform Rational B-Splines (NURBS).

We then consider to represent $\bar{x}(t)$ from the selected control points in (5.9) such that

$$\bar{x}(t) \approx x(t),\ \forall t \in [t_0, t_m]. \tag{5.10}$$

Let $\bar{t}_i, i = -3, -2, \cdots, \bar{m} - 1$ be knot points corresponding to each control point of $\bar{\tau} \in \mathbf{R}^{\bar{M}}$, where $\bar{m} = \bar{M} - 3$. Since the intervals between such knot points may be non-uniform, we then consider to represent $\bar{\tau}$ as curves $\bar{x}(t), t \in [t_0, t_m]$ by employing NURBS in Chapter 2 as

$$\bar{x}(t) = \frac{\sum_{i=-3}^{m-1} w_i \bar{\tau}_i B_k(\alpha_i(t - \bar{t}_i))}{\sum_{i=-3}^{m-1} w_i B_k(\alpha_i(t - \bar{t}_i))}. \tag{5.11}$$

Here, $\bar{t}_i$ denotes the interval of knot point after using dynamic programming (DP) selected dominant control points from original. We note that this is a general formula, which is for the case where the knot points are equally spaced. Also, $w_i (\geq 0)$ is a weight coefficient with

$$\sum_{i=-3}^{m-1} w_i = 1. \tag{5.12}$$

Thus, letting $w \in \mathbf{R}^{\bar{M}}$ be a vector of weight $w_i, i = -3, -2, \cdots, \bar{m} - 1$ defined as

$$w = [w_{-3} \ w_{-3+1} \ \cdots \ w_{\bar{m}-1}]^T, \tag{5.13}$$

our task is only to compute a vector w such that $\bar{x}(t) \approx x(t)$. We then consider to find $w \in \mathbf{R}^{\bar{M}}$ such that

$$\min_{w \in \mathbf{R}^{\bar{M}}} \sum_{i=1}^{N} ||x(s_i) - \bar{x}(s_i)||^2. \tag{5.14}$$

Here, $s_i \in [t_0, t_m], \ i = 1, 2, \cdots, N$ denotes a equally spaced sampled point defined as

$$s_i = (i - 1)\Delta s, \tag{5.15}$$

with a constant $\Delta s (> 0)$. It can be shown that this problem is written as non-linear programming problem with equality and inequality linear constraints in terms of w. We here used Matlab function 'fmincon' for solving this problem.

5.4 Experimental Study

We demonstrate the performance of our proposed method by experimental studies. In particular, we here apply the proposed method to a problem on object contour modeling. Now, suppose that we are given a periodic B-spline curve $x(t)$ and the corresponding control polygons $\mathcal{M}$ for a set of data sampled from the following periodic function $f(t)$,

$$f(t) = \sqrt{p^2(t) + q^2(t)}, \tag{5.16}$$

with

$$\begin{aligned} p(t) &= 3 + r(t)\cos(2\pi t/36), \\ q(t) &= 3 + r(t)\sin(2\pi t/36), \end{aligned} \tag{5.17}$$

and

$$r(t) = 2 + \sin(10\pi t/36). \tag{5.18}$$

Figure 5.1 illustrates the designed periodic curves $x(t)$. Here, the green dashed line with the black asterisk denotes $f(t)$ with the data. The blue line is the designed periodic B-spline curve $x(t)$. We then construct a sequence of the original control points for periodic spline curves $x(t)$ by using the theory of smoothing splines [41]. Here, α is set as α=10. Hence, the total number of control points M is set as $M = 39$ and the time interval $[t_0, t_m]$ are $[0, 36]$. Using the our proposed method in this

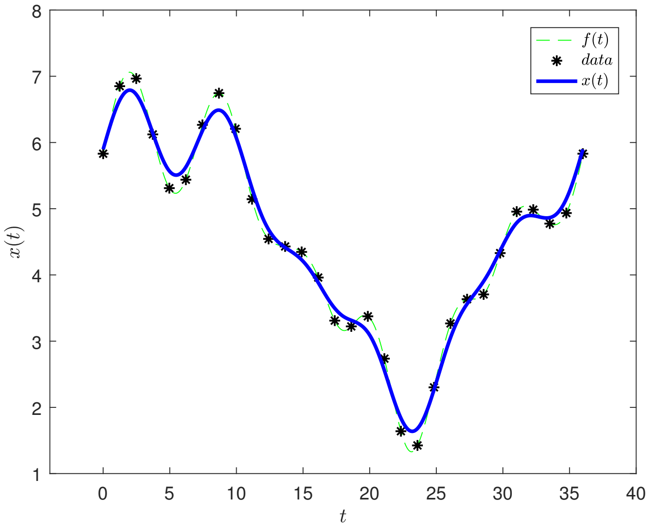


Figure 5.1: Designed periodic B-spline curve $x(t)$.

chapter, we design the compact periodic B-spline curves $\bar{x}(t)$ from the given curve $x(t)$. Also, Δs is set as $\Delta s = 0.02$, and hence N is a set of data. The design examples are illustrated in Figures 5.2-5.10. Here, $l_{\max}$ in (5.11) is set as $l_{\max} = 0, 1$ and 2 with $\Lambda_0 = I_2, \Lambda_1 = 10I_2$ and $\Lambda_2 = 10^2 I_2$, where $I_2 \in \mathbf{R}^{2\times 2}$ denotes an identity matrix. Also, for each case of Figures 5.2-5.10 were the result setting as $K = 31$ $(\bar{M} = 35)$, $K = 29$ $(\bar{M} = 33)$, and $K = 25$ $(\bar{M} = 29)$ from original control points. In

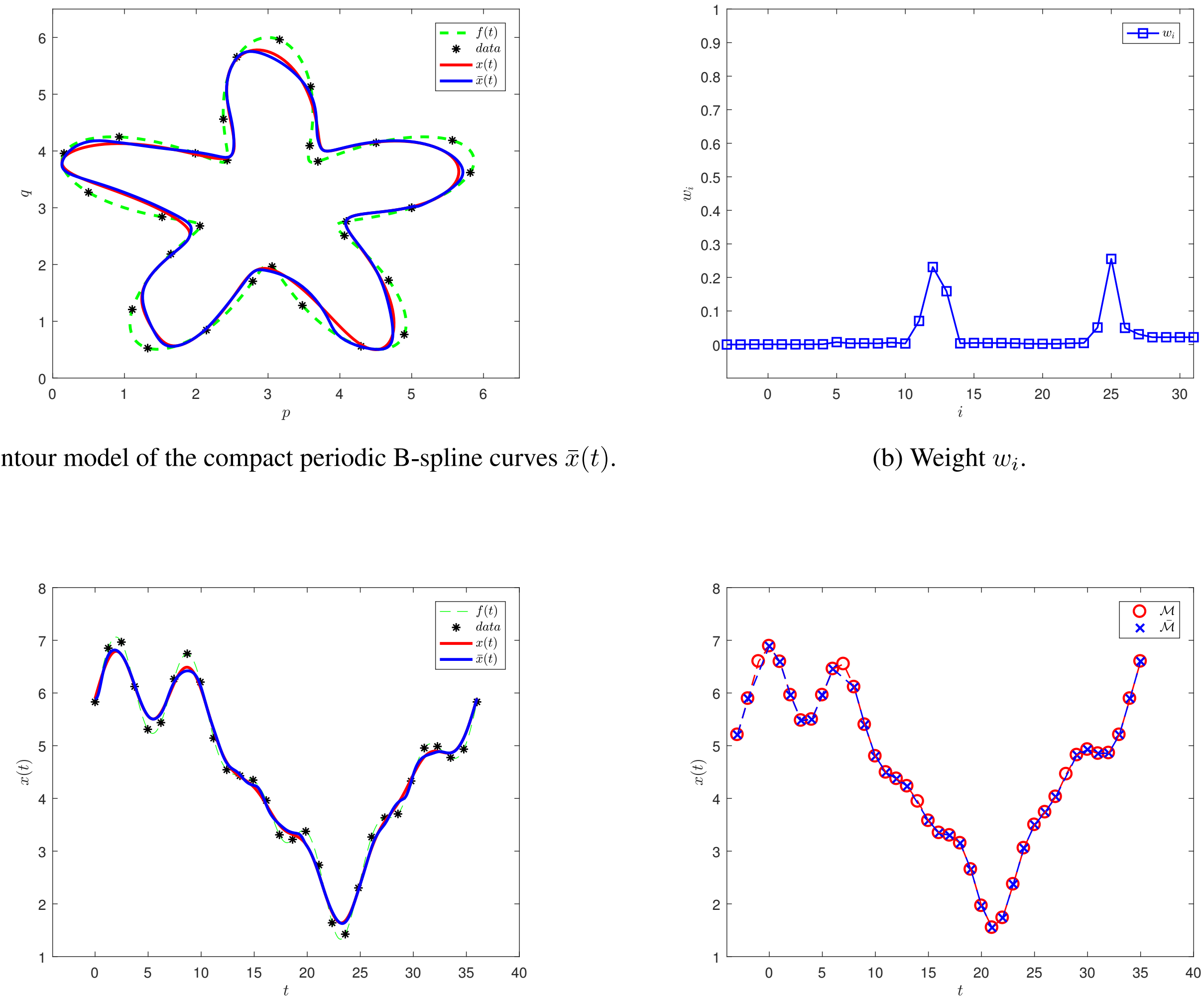
Figures 5.2(a)-5.10(a), the given $x(t)$ and the reconstructed $\bar{x}(t)$ are plotted in the polar coordinate system $O-pq$. Here, the red and the blue lines denote the given curves $x(t)$ and the reconstructed compact periodic B-spline curves $\bar{x}(t)$. Also, green dashed lines with the black asterisk denote $f(t)$ with data. Also, in Figures 5.2(b)-5.10(b), the weights w_i obtained by solving in (5.14) are plotted in the blue line with square marks. In Figures 5.2(c)-5.10(c) we show that the result of given curve $x(t)$ and the reconstructed curve $\bar{x}(t)$. Then, the corresponding control polygons $\mathcal{M}$ (red dashed lines with circle marks) and $\bar{\mathcal{M}}$ (blue dashed lines with cross marks) are plotted in Figures 5.2(d)-5.10(d), respectively.

From these results, we see that our proposed method relatively works well for the case of $l_{max} = 2$. In the case of $l_{max} = 0$, noting that the control polygon $\mathcal{M}$ represent the geometrical outline of curves $x(t)$, we may see that the control points are selected such that a discrepancy between $\mathcal{M}$ and $\hat{\mathcal{M}}$ by the dynamic programming. Such a selection strategy may often cause that a linear sequence of control points is unselected intensively (see e.g., Figures 5.2, 5.5, and 5.8). Then, the knot point interval corresponding to the unselected control points becomes wider. Hence, representing compact periodic B-spline curves in a class of cubic NURBS may become difficult only by adjusting the weights w_i. As l_{max} in (4.11) becomes large, the unselected control points become scattered as shown in Figures 5.3-5.4, 5.6-5.7 and 5.9-5.10. We then may observe that the approximation gets better.

From these results, we observe that the approximations for the case of $l_{max} = 2$ are better than the cases of $l_{max} = 0$ and 1, but we cannot avoid that the approximation gets worse as the number of selected control points K (or $\bar{\mathcal{M}}$) becomes small.

Also, we here compare the above results with the case where the control points with a specified number of K are selected randomly. Setting K as $K = 29$ $(\bar{M} = 33)$ for the cases of contour model in pq-plane, we repeatedly construct the periodic B-spline curves $\bar{x}(t)$ using the method in this Chapter one hundred times and their results are plotted in Figure 5.11. We may see that these results indicate that randomly selecting the dominant control points often leads to the unstable reconstruction of compact periodic curves. Comparing these results with the results in Figure 5.7, it is obvious that our proposed method is more effective than the cases of this random control point selection.

The most remarkable thing is that the same scheme as the case of planar B-spline curves in Chapter 4 works well even for the case with periodic constraints.



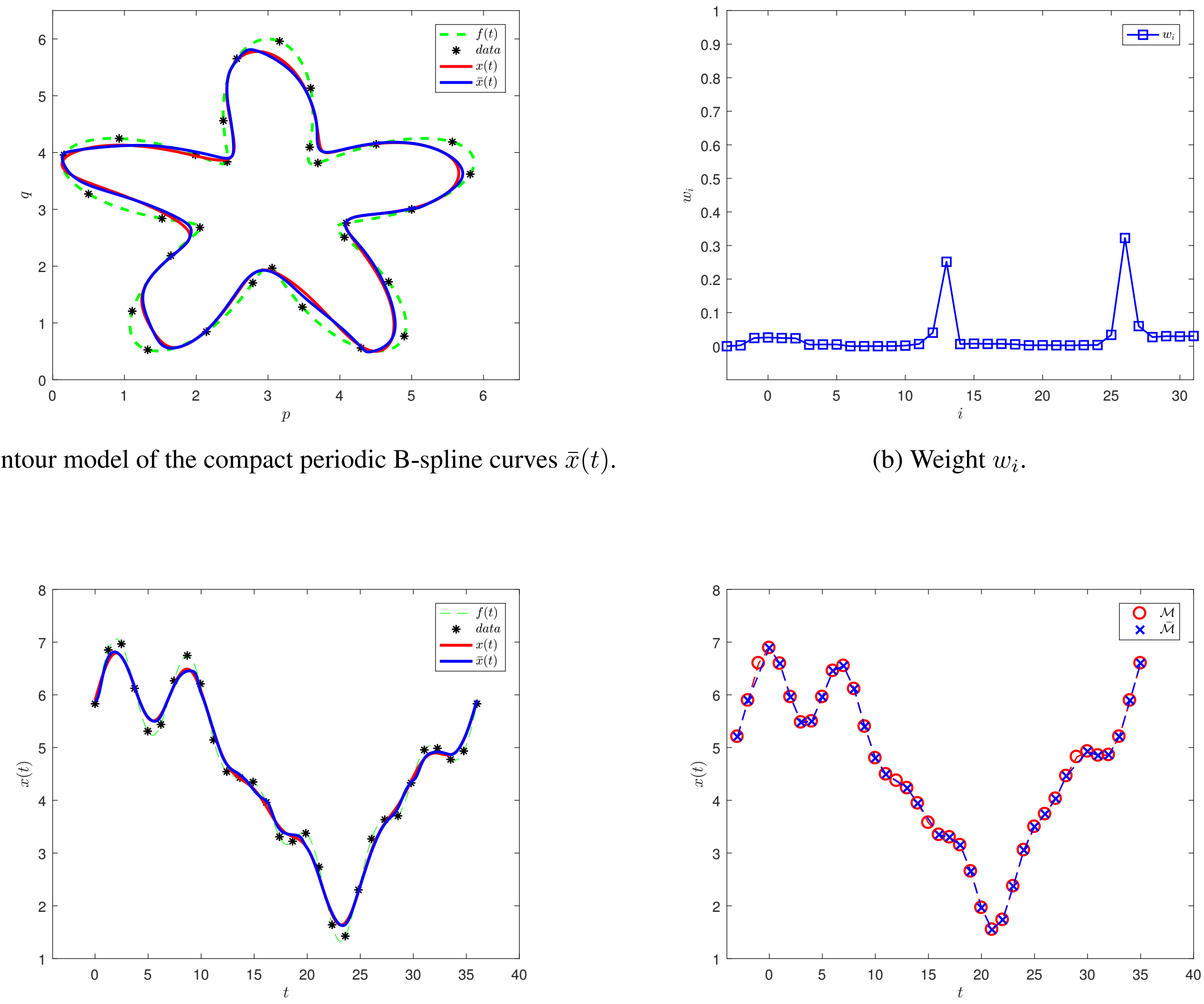
(a) Contour model of the compact periodic B-spline curves $\bar{x}(t)$.

(b) Weight w_i.

(c) Compact periodic B-spline curve $\bar{x}(t)$.

(d) Selected control polygons $\bar{\mathcal{M}}$.

Figure 5.2: Result of Compact periodic B-spline curves when $l_{max} = 0, K = 31$ ($\bar{M} = 35$).



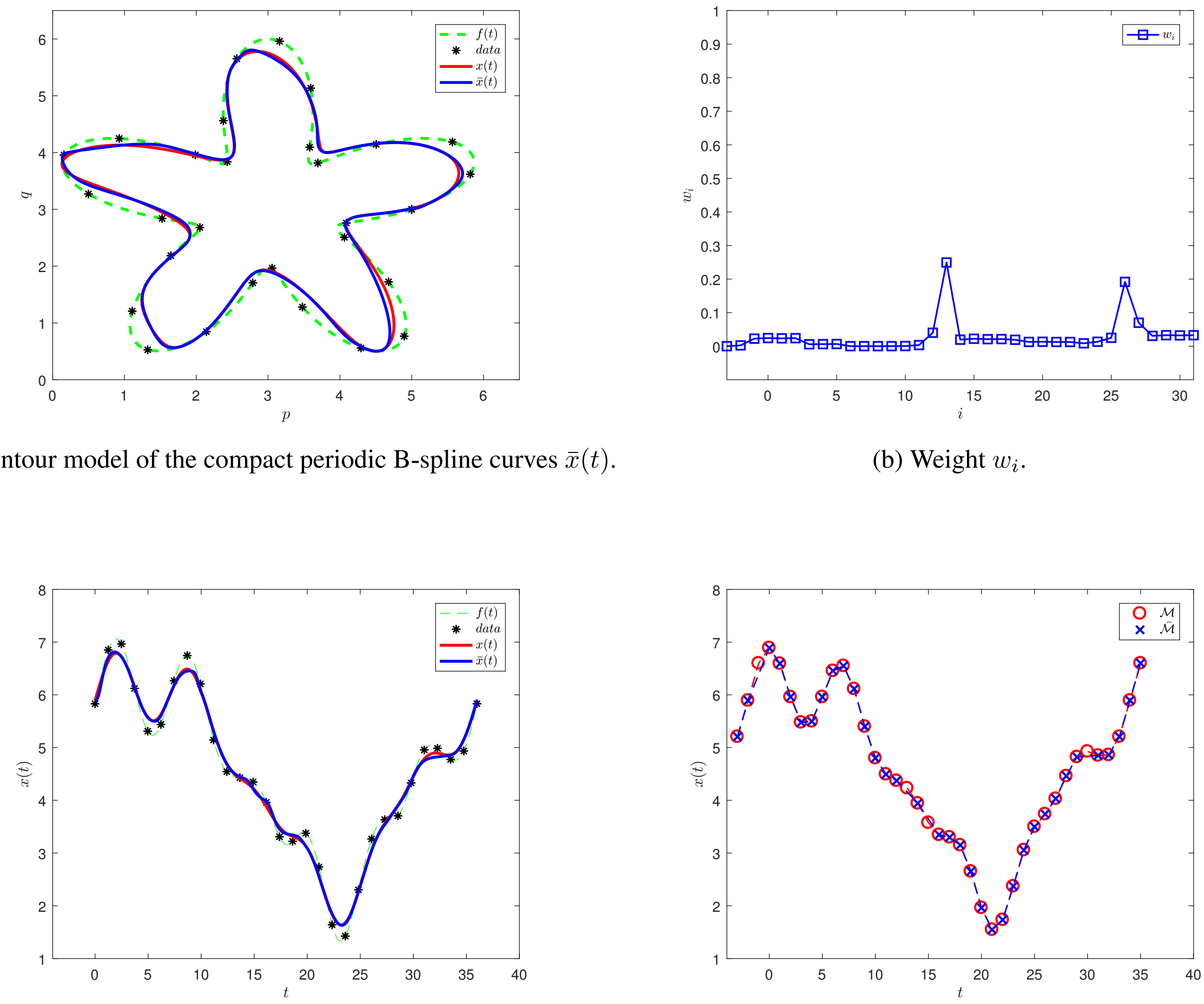
(a) Contour model of the compact periodic B-spline curves $\bar{x}(t)$.

(b) Weight w_i.

(c) Compact periodic B-spline curve $\bar{x}(t)$.

(d) Selected control polygons $\bar{\mathcal{M}}$.

Figure 5.3: Result of Compact periodic B-spline curves when $l_{max} = 1, K = 31$ ($\bar{M} = 35$).



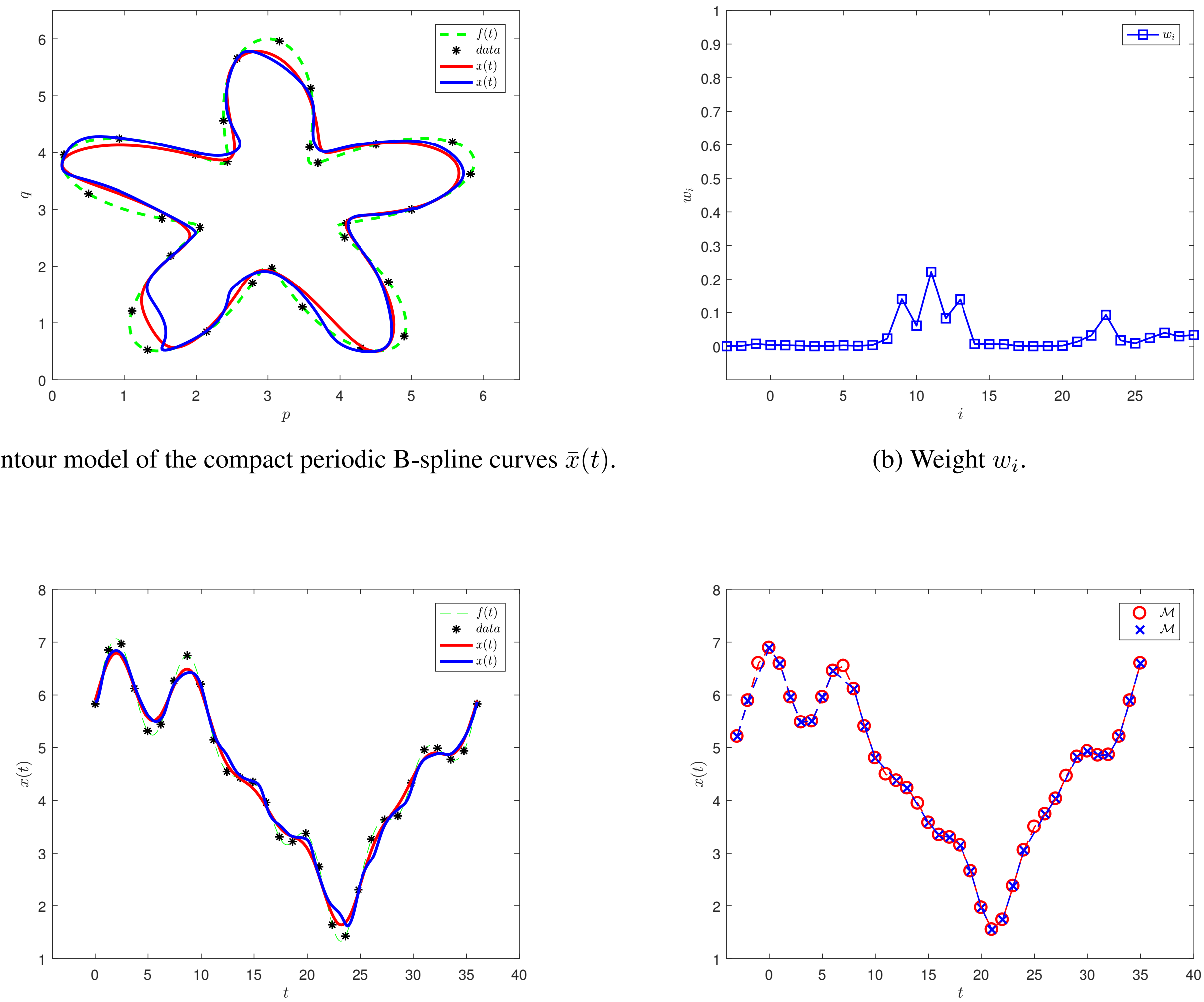
(a) Contour model of the compact periodic B-spline curves $\bar{x}(t)$.

(b) Weight w_i.

(c) Compact periodic B-spline curve $\bar{x}(t)$.

(d) Selected control polygons $\bar{\mathcal{M}}$.

Figure 5.4: Result of Compact periodic B-spline curves when $l_{max} = 2, K = 31$ ($\bar{M} = 35$).



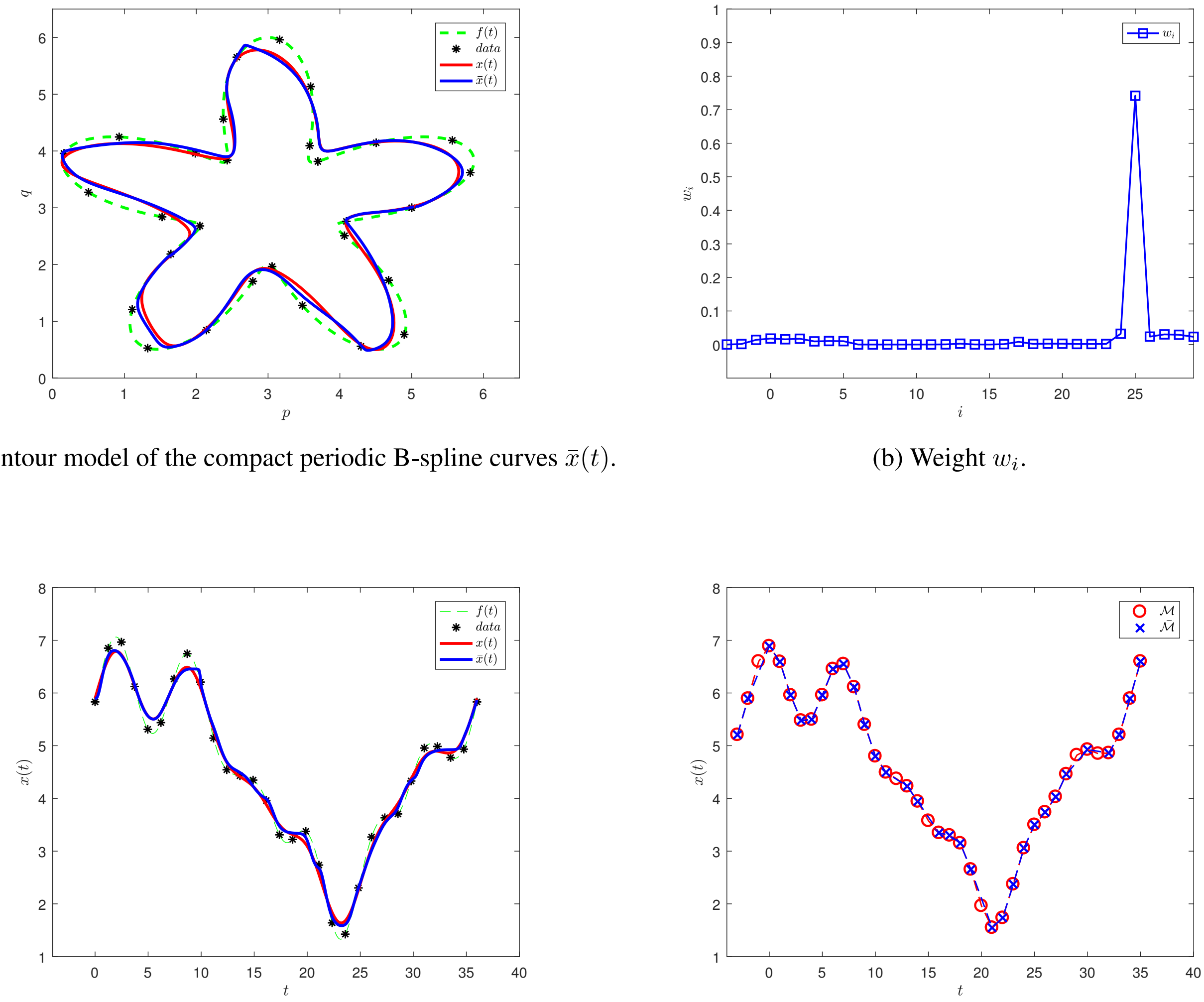
(a) Contour model of the compact periodic B-spline curves $\bar{x}(t)$.

(b) Weight w_i.

(c) Compact periodic B-spline curve $\bar{x}(t)$.

(d) Selected control polygons $\bar{\mathcal{M}}$.

Figure 5.5: Result of Compact periodic B-spline curves when $l_{max} = 0, K = 29$ ($\bar{M} = 33$).



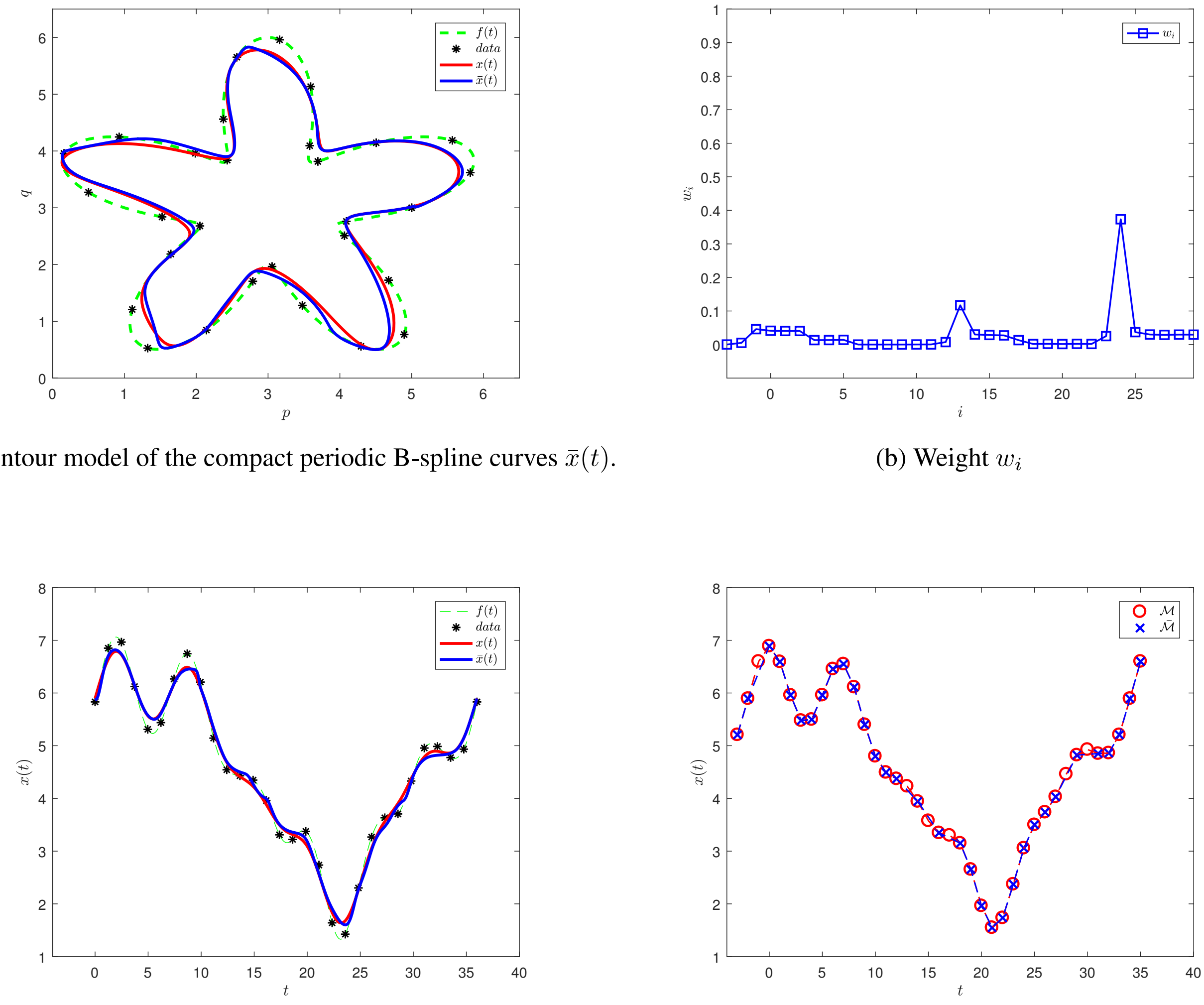
(a) Contour model of the compact periodic B-spline curves $\bar{x}(t)$.

(b) Weight w_i.

(c) Compact periodic B-spline curve $\bar{x}(t)$.

(d) Selected control polygons $\bar{\mathcal{M}}$.

Figure 5.6: Result of Compact periodic B-spline curves when $l_{max} = 1, K = 29$ ($\bar{M} = 33$).



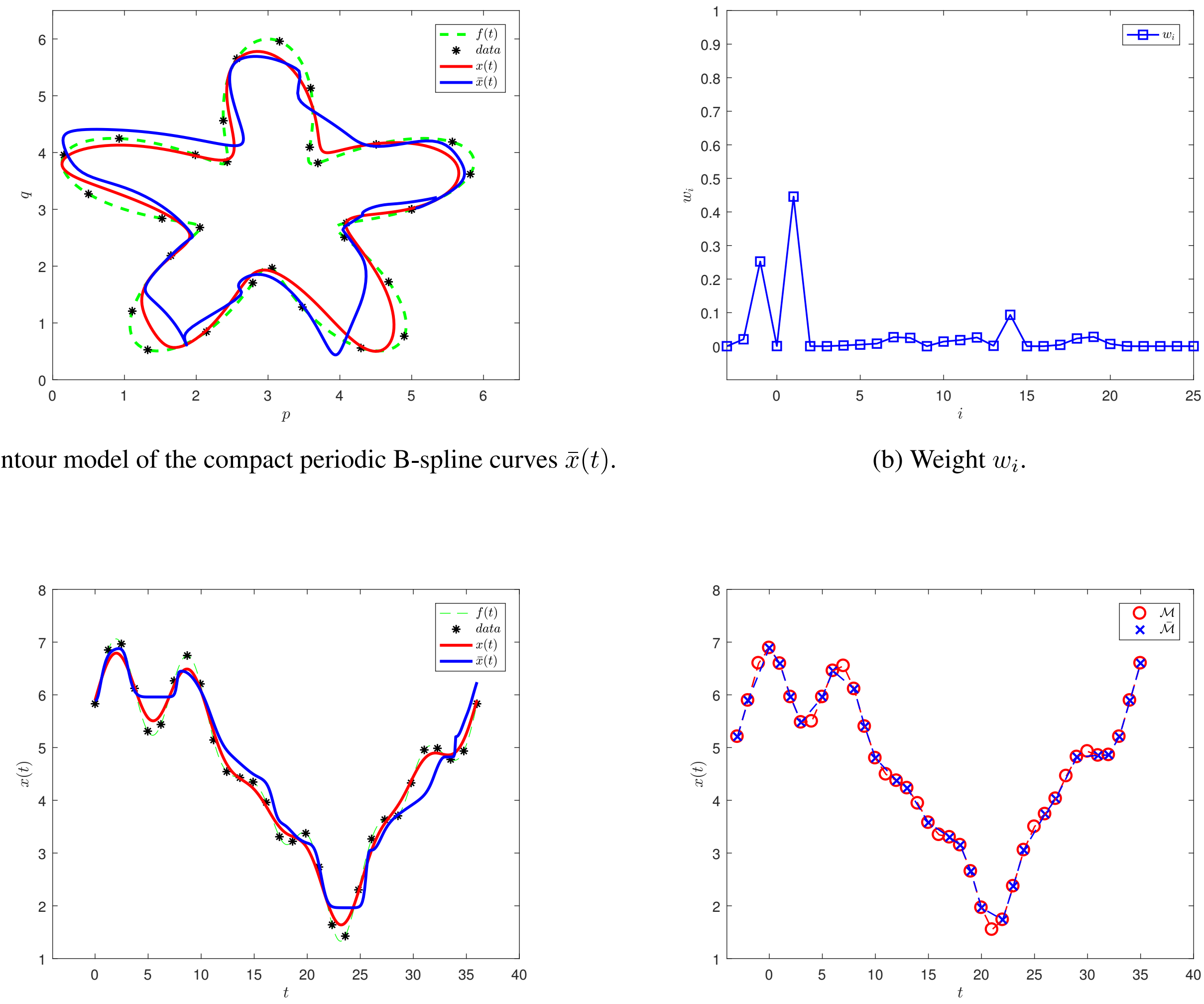
(a) Contour model of the compact periodic B-spline curves $\bar{x}(t)$.

(b) Weight w_i

(c) Compact periodic B-spline curve $\bar{x}(t)$.

(d) Selected control polygons $\bar{\mathcal{M}}$.

Figure 5.7: Result of Compact periodic B-spline curves when $l_{max} = 2, K = 29$ ($\bar{M} = 33$).



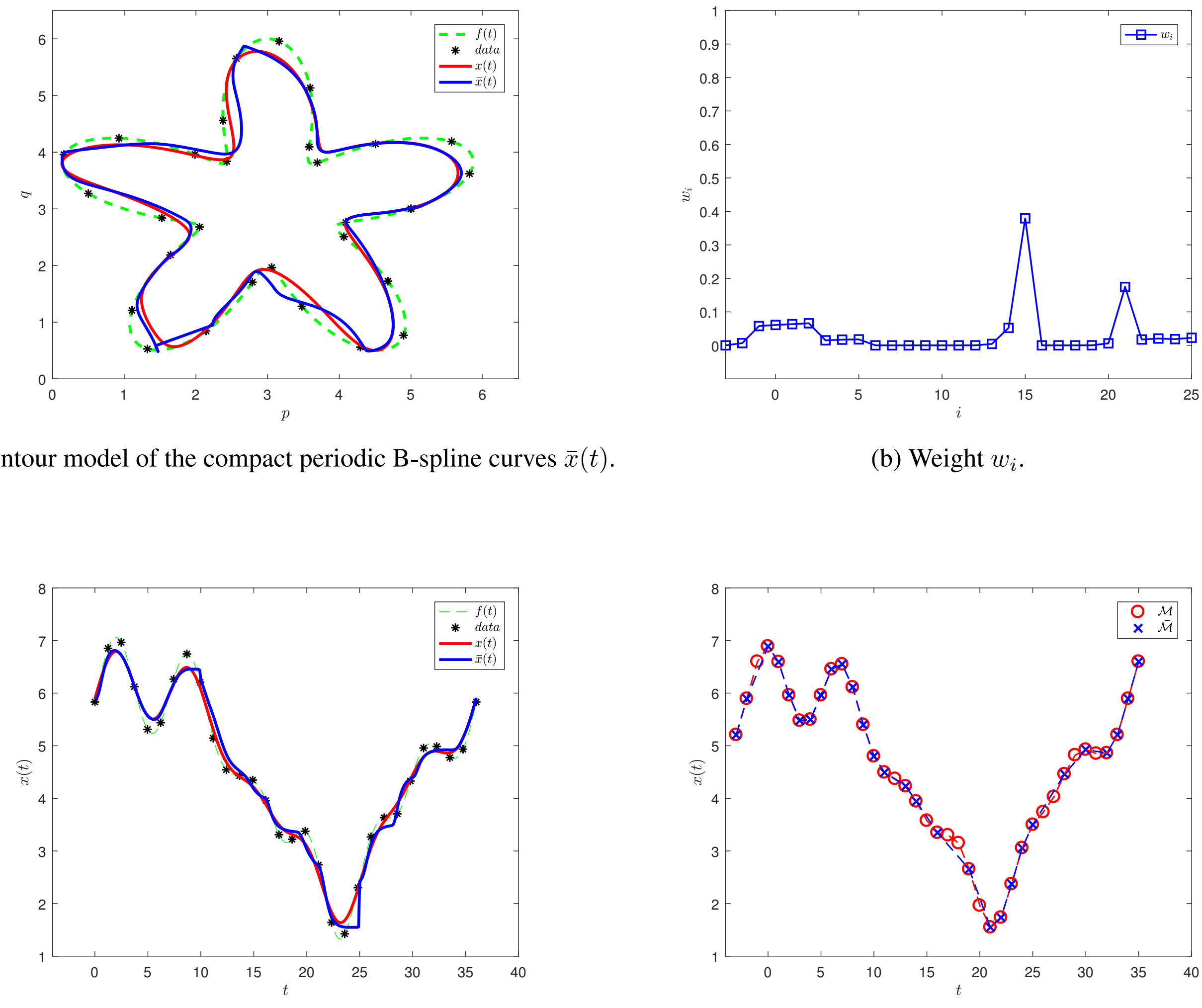
(a) Contour model of the compact periodic B-spline curves $\bar{x}(t)$.

(b) Weight w_i.

(c) Compact periodic B-spline curve $\bar{x}(t)$.

(d) Selected control polygons $\bar{\mathcal{M}}$.

Figure 5.8: Result of Compact periodic B-spline curves when $l_{max} = 0, K = 25$ ($\bar{M} = 29$).



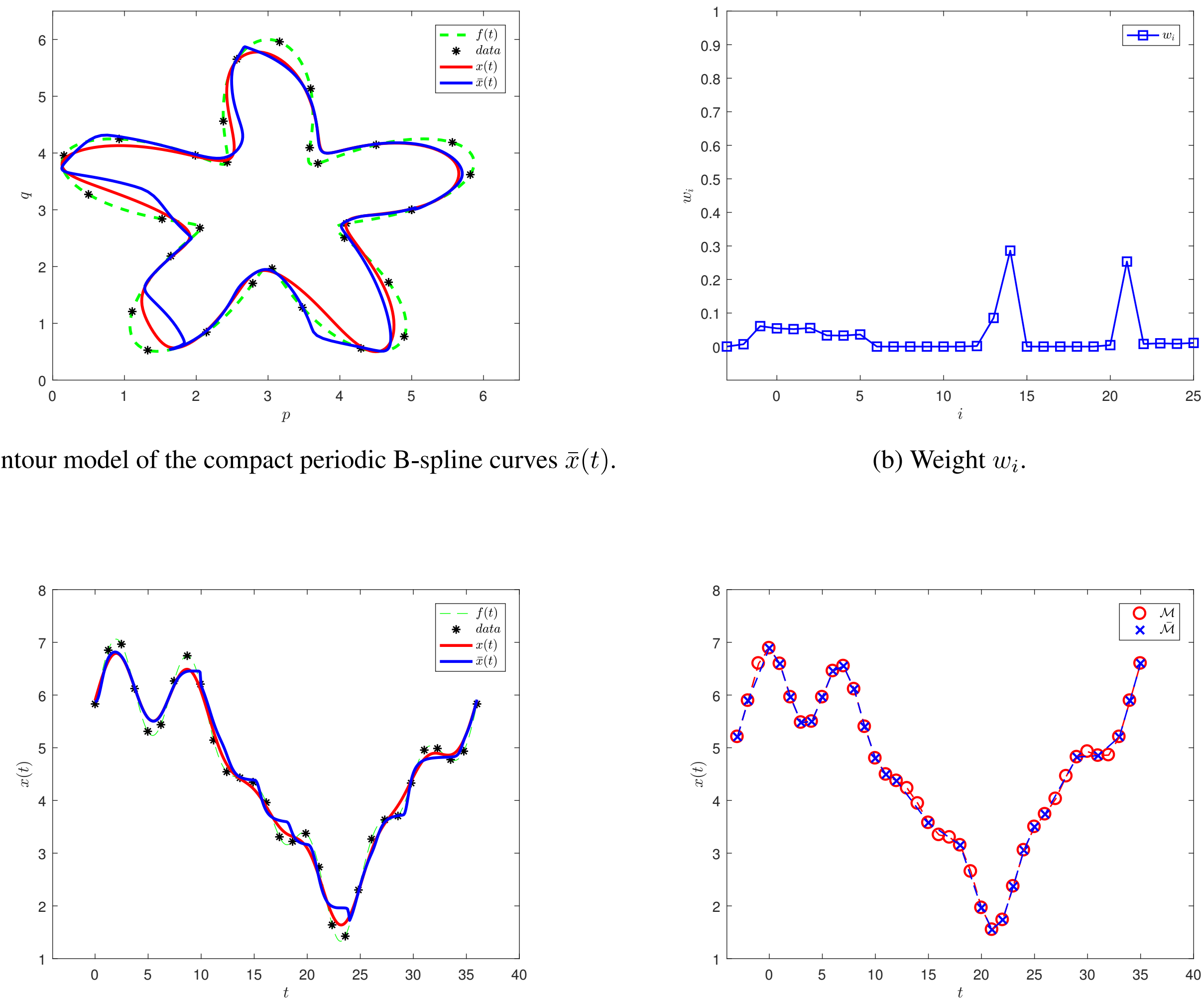
(a) Contour model of the compact periodic B-spline curves $\bar{x}(t)$.

(b) Weight w_i.

(c) Compact periodic B-spline curve $\bar{x}(t)$.

(d) Selected control polygons $\bar{\mathcal{M}}$.

Figure 5.9: Result of Compact periodic B-spline curves when $l_{max} = 1, K = 25$ ($\bar{M} = 29$).



(a) Contour model of the compact periodic B-spline curves $\bar{x}(t)$.

(b) Weight w_i.

(c) Compact periodic B-spline curve $\bar{x}(t)$.

(d) Selected control polygons $\bar{\mathcal{M}}$.

Figure 5.10: Result of Compact periodic B-spline curves when $l_{max} = 2, K = 25$ ($\bar{M} = 29$).

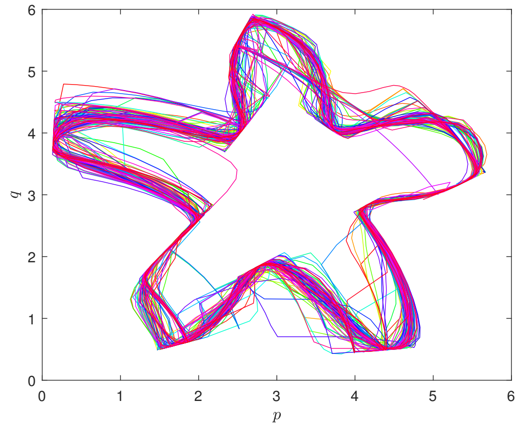


Figure 5.11: Compact periodic B-spline curves $\bar{x}(t)$ for the case where the control points are randomly selected.

Chapter 6

Concluding Remarks

In this thesis, we considered the problem of designing the compact B-spline curves by using only the dominant control points. In particular, we here developed such a design method for the typical two types of B-spline curves “planar B-spline curve” and “periodic B-spline curves”. The central issue was how to optimally select dominant control points. For such an optimal dominant control point selection, we here introduced an optimization approach using a dynamic programming (DP) method. Namely, it was shown that the selection problem is formulated as a graph problem and is solved by DP. The merit of using the DP approach is to enable us to reduce the amount of computation time unlike the ordinary approaches- such as the trial-and-error approach exhibited in Lyche and Mørken’s work [28]. In addition, what made the paper noteworthy was that a new idea of knots selections based on multi-level error functions is introduced, where the term ‘multi-level’ means that not only function values of a given curve but also its derivatives are considered. Also, we showed that representation of compact B-spline curves using the selected control points can be realized using NURBS (Non-Uniform Rational B-splines) since the allocation of their knot points are generally non-uniform. The methods for the planar and periodic splines were applied for the character design using the so-called dynamic font method and the contour modeling problem for the deformable objects, respectively. The performances were demonstrated by experimental studies.

In Chapter 1, we introduced the background including some related works and the purpose of this thesis as an introduction.

In Chapter 2, we presented the fundamental theory of spline curves. Therein, the polynomial curves and Bezier curves, and their properties were presented. Then, we presented B-spline as well as Non-

Uniform B-splines (NURBS), which were used throughout this thesis.

In Chapter 3, we presented the dynamic programming including basics of graph theory, which are mainly used in order to solve the problem of selecting dominant control points.

In Chapters 4 and 5, we considered the problem of designing the compact B-spline curves for the two cases of the planar spline curves and periodic spline curves, respectively. The planar spline curves in Chapter 4 are curves that are represented as a parametric equation of the form $(x, y) = (x(t), y(t))$ for curves $x(t)$ and $y(t)$. Then, our main task was to design compact B-spline curves independently for X and Y directions. In addition, we confirmed such the task was basically similar to the case of periodic spline curves where we need to impose the periodic constraints. For achieving such tasks, we developed the method for selecting the dominant control points by employing the DP approach. Specifically, the directed acyclic graph (DAG) was constituted from the sequence of control points. Then, the optimal path was found by DP so that some cost function is minimized, and then the dominant control points were obtained as the optimal path corresponding to the control polygon consisting of dominant control points. Regarding such the cost function, we here introduced the so-called "multi-level function" that not only function values of a given curve but also its derivatives are considered. In addition, what is remarkable about the proposed selection method is that the computation complexity is quite low because of the DP-based algorithm. In fact, the complexity to find the K-dominant control points from $\bar{M}$ is $O(K\bar{M}^2)$ [45]. Then, we showed that the representation of compact planar B-spline curves is realizable by introducing NURBS. We demonstrated the performance with some experimental studies. Then, we observed that introducing multi-level error will result in better approximations of compact planar B-spline curves, although it is unavoidable that the approximations get worse as the number of selected control points becomes small. Also, we observed through some experimental studies that random selection of the dominant control points often leads to an unstable reconstruction of compact spline curves. Thus, it was obvious that our proposed method is more effective than cases of random control point selection.

For improving the issues on approximation errors, it remains to develop some method for relocating the knot points corresponding to the selected control points, which is left for a future work. Solving such the knot points' relocation is generally not easy, and some strategies have been proposed by many researchers. In particular, the strategy which we have to develop here may be based on the idea of "free knots" spline since we could effectively select the dominant control points by our proposed method. Such the relocation method has been developed by Schwetlick, et al. [56]. Therein, the method consists of minimizing some global criterion over both the position of the knots and the coefficients of the splines. However, what we want to develop may be a method for relocating only the knot points since we want to hold the outline of the control polygon from the viewpoint of user-friendliness on curve design. On the other hand, the advantage of using B-splines is the dimensional extendability. That is, we may extend the results of this thesis to the higher dimensional cases-such as surface and three-dimensional objects, etc. This will be another future work.

Bibliography

[1] C. F. Martin, S. Sun and M. Egerstedt, Optimal Control, Statistics and Path Planning, *Mathematical and Computer Modeling*, Vol.33, No.1–3, pp.237–253, 2001.

[2] H. Ni, Z. Li and H. Song, Moving Least Square Curve and Surface Fitting with Interpolation Conditions, *Proceedings of 2010 International Conference on Computer Application and System Modeling (ICCASM 2010)*, pp.300–304, Taiyuan, China, Oct.22-24, 2010.

[3] B. W. Silverman, Some Aspects of The Spline Smoothing Approach to Nonparametric Regression Curve Fitting with Discussion, *Journal of the Royal Statistical Society*, Vol.47, No.1, pp.1–52, 1985.

[4] R. Kress, *Numerical Analysis*, New York : Springer–Verlag, 1998.

[5] X. Zhang and K. K. Mei, Time-Domain Calculation of Microstrip Components and The Curve-Fitting of Numerical Results, *Proceedings of IEEE MTT-S International Microwave Symposium Digest*, pp.313–316, Long Beach, CA, USA, Jun.13-15, 1989.

[6] L. J. Wang, W. S. Hsieh, T. K. Truong, I. S. Reed, T. C. Cheng, A Fast Efficient Computation of Cubic-Spline Interpolation in Image Codec, *IEEE Trans. Signal Process.*, Vol.49, No.6, pp.1189–1197, 2001.

[7] O. Faugeras, *Three-Dimensional Computer Vision –A Geometric Viewpoint–*, Cambridge, MA : MIT Press, 1996.

[8] N. Ray, A Concave Cost Formulation for Parametric Curve Fitting: Detection of Leukocytes from Intravital Microscopy Images, *Proceedings of 2010 IEEE 17th International Conference on Image Processing*, pp.53–56, Hong Kong, China, Sep.26-29, 2010.

[9] K. Takayama and H. Kano, A New Approach to Synthesizing Free Motions of Robotic Manipulators Based on a Concept of Unit Motions, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol.25, No.3, pp.453–463, 1995.

[10] Q. Huang, K. Yokoi, S. Kajita, K. Kaneko, H. Arai, N. Koyachi, K. Tanie, Planning Walking Patterns for a Biped Robot, *IEEE Transactions on Robotics and Automation*, Vol.17, No.3, pp.280–289, 2001.

[11] W. Khalil and E. Dombre, *Modeling, Identification and Control of Robots*, London, U.K. : Hermes Penton Ltd., 2002.

[12] J. J. Craig, *Introduction to Robotics : Mechanics and Control*, MA : Addison-Wesley, 1989.

[13] W. L. Xu, B. L. Ma and S. K. Tso, Curve Fitting Approach to Motion Planning of Nonholonomic Chained Systems, *Proceedings of 1999 IEEE International Conference on Robotics and Automation*, pp.811–816, Detroit, MI, USA, May.10-15, 1999.

[14] I. J. Schoenberg, Contributions to The Problem of Approximation of Equidistant Data by Analytic Functions. Part A–On the Problem of Smoothing or Graduation. A First Class of Analytic Approximation formulae, *Quarterly of Applied Mathematics*, Vol.4, No.2, pp.45-99, 1946.

[15] I. J. Schoenberg, Contributions to The Problem of Approximation of Equidistant Data by Analytic Function: Part B–On The Problem of Osculatory Interpolation. A Second Class of Analytic Approximation Formulae, *Quarterly of Applied Mathematics*, Vol.4, No.2, pp.112–141, 1946.

[16] H. Fujioka and H. Kano, Optimal Smoothing Splines for Detecting Extrema from Observational Data, *International Journal of Statistics and Systems*, Vol.1, No.2, pp.111–132, 2006.

[17] H. Fujioka and H. Kano, Periodic Smoothing Spline Surface and Its Application to Dynamic Contour Modeling of Wet Material Objects, *IEEE Trans. Systems, Man and Cybernetics, Part A*, Vol.39, No.1, pp.251-261, 2009.

[18] J. G. Dunham, Optimum Uniform Piecewise Linear Approximation of Planar Curves, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, Issue 1, pp.67–75, 1986.

[19] Y. H. Gu, T. Tjahjadi, Efficient Planar Object Tracking and Parameter Estimation Using Compactly Represented Cubic B-spline Curves, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol.29, No.4, pp.358–367, 1999.

[20] P. Bo, G. Luo and K. Wang, A Graph-Based Method for Fitting Planar B-spline Curves with Intersections, *Journal of Computational Design and Engineering*, Vol.3, Issue 1, pp.14–23, 2016.

[21] A. Morgand, M. Tamaazousti and A. Bartoli, A Geometric Model for Specularity Prediction on Planar Surfaces with Multiple Light Sources, *IEEE Transactions on Visualization and Computer Graphics*, Vol.24, Issue 5, pp.1691–1704, 2018.

[22] K. M. Soo, M. S. Ryong and L. K. Hee Motion Planning with Planar Geometric Models Curves, *Proceedings 1991 IEEE International Conference on Robotics and Automation*, pp.1015–1020, Sacramento, CA, USA, Apr.9–11, 1991.

[23] K. Takayama and H. Kano, Dynamic Font : A New Representational Technology, *FSTJ*, Vol.32, No.2, pp.192–202, 1996

[24] H. Nakata and H. Kano, Generation of Japanese Cursive Sentences using Optimal Smoothing Splines, *J.Inform. Process. Soc. Japan*, Vol.44, No.1, pp.134–142, 2003.

[25] N. Y. Graham, Smoothing with Periodic Cubic Splines, *The Bell System Technical Journal*, Vol.62, Issue 1, pp.101–110, 1983.

[26] H. Kano, H. Fujioka, M. Egestedt and C. F. Martin, Optimal Smoothing Spline Curves and Contour Synthesis, *Proc. the 16th IFAC World Congress*, pp.297-302, Prague, Czech Republic, July 4-8, 2005.

[27] W. XingCe, Z. MingQuan and L. XinYu, An Automatic Matching Method of Object Contour Curves Based on Periodic Curvature Function, *2006 IEEE International Conference on Robotics and Biomimetics*, pp.1146–1150, Kunming, China, Dec.7–20, 2006

[28] T. Lyche and K. Mørken, Knot Removal for Parametric B-spline Curves and Surfaces, *Computer Aided Geometric Design*, Vol.4, Issue 3, pp.217–230, 1987.

[29] H. Park and J. H. Lee, B-spline Curve Fitting Using Dominant Point, *Proceedings of International Conference on Computational Science Computational Science–ICCS 2006, Lecture Notes in Computer Science*, Vol. 3992, pp.362–366, May.28-31, 2006.

[30] T. Tjahjowidodo, V. T. Dung, and M. L. Han, A Fast Non-Uniform Knots Placement Method for B-Spline Fitting, *Proceedings of 2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pp.1490–1495, Busan, South Korea, Jul.7-11, 2015.

[31] H. Kano, H. Fujioka, M. Egerstedt and C. F. Martin. Optimal Smoothing Spline Curves and Contour Synthesis. *2005 IFAC. 16th Triennial World Congress, Prague, Czech Republic*, Vol.38, Issue 1, pp.297–302, 2005.

[32] C. de Boor, *A Practical Guide to Splines*, New York : Springer–Verlag, 1978.

[33] C. de Boor, *I. J. Schoenberg Selected Papers*, New York : Springer–Verlag, 1988.

[34] K. Ruohonen, *Graph Theory*, Tampere University of Technology, 2013.

[35] R. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, NJ, 1957.

[36] S. S. Ray, *Graph Theory with Algorithms and its Applications*, India : Springer–Verlag, 2013.

[37] A. Amini, T. Weymouth and R. Jain, Using Dynamic Programming for Solving Variational Problems in Vision, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.2, No.9, pp.855–867, 1990.

[38] R. Basri, L. Costa, D. Geiger, and D. Jacobs, Determining The Similarity of Deformable Shapes, *Vision Research*, Vol.38, Issues 15–16, pp.2365–2385, 1998.

[39] P. F. Felzenszwalb and D. P. Huttenlocher, Distance Transforms of Sampled Functions, *Theory of Computing (ToC)*, Vol.8, No.19, pp.415–428, 2004.

[40] T. C. Hu and J. D. Morgenthaler, Dynamic Programming and Graph Optimization Problems, *Computers Math. Applic.*, Vol.27, No.9–10, pp.53–58, 1994.

[41] H. Kano, H. Nakata and C. F. Martin, Optimal Curve Fitting and Smoothing Using Normalized Uniform B-splines : A Tool for Studying Complex Systems, *Applied Mathematics and Computation*, Vol.169, Issue 1, pp.96–128, 2005.

[42] H. Park and J. H. Lee, B-spline Curve Fitting based on Adaptive Refinement using Dominant points, *Computer Aided-Design*, Vol.39, pp.439–451, 2007.

[43] R. Soontornvorn and H. Fujioka, Design of Compact Planar B-spline Curves Using DP Control Point Selection, *Proceedings of TENCON 2018 IEEE Region 10 Conference*, pp.1476–1479, Jeju, Korea, Oct.28–31, 2018.

[44] R. Soontornvorn and H. Fujioka, Design of Compact Planar B-spline Curves using DP Control Point Selection with Multi–level Error Functions- Towards Usability Improvement in Design

of Dynamic Font-based Characters, *IEIE Trans. Smart Processing and Computing*, Vol.8, No.2, pp.108–120, 2019.

[45] R. Hu and S. M. Watt, Optimization of Point Selection on Digital Ink Curves, *Proceedings of 13th International Conference on Frontiers in Handwriting Recognition*, pp.525-530, Bari, Italy, Sept.18-20, 2012.

[46] P. Lancaster and M. Tismenetsky, *The Theory of Matrices*, Academic Press, 1985.

[47] H. Fujioka, H. Kano, H. Nakata and H. Shinoda, Constructing and Reconstructing Characters, Words and Sentences by Synthesizing Writing Motions, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol.36, No.4, pp.661–670, 2006.

[48] M. Hosaka, *Modeling of Curves and Surfaces in CAD/CAM*, New York : Springer–Verlag, 1992.

[49] L. Piegl and W. Tiller, *The NURBS Book*, Heidelberg : Springer–Verlag, 1995.

[50] D. F. Rogers, *Introduction to NURBS: with Historical Perspective*, Academic Press, 2001.

[51] Y. Mieno, H. Fujioka and H. Kano, Data Compression of Digital-Ink with Pen-Slips Using Multi-level L1 Smoothing Splines, *Proceedings of The 2015 IEEE International Conference on Systems, Man, and Cybernetics*, pp.1787–1792, Hong Kong, Oct. 9-12, 2015.

[52] G. Guo, T. Jiang, Y. Wang and W. Gao, Recovering Missing Contours for Occluded Object Detection, *IEEE Signal Processing Letters*, Vol.19, Issue 8, pp.463–466, 2012.

[53] A. Blake and M. Isard, *Active Contours*, London : Springer–Verlag, 1998.

[54] G. Wahba, *Spline Models for Observational Data*, Philadelphia, PA : Soc. Ind. SIAM, 1990.

[55] O. Valenzuela, M. Pasadas, I. Rojas, A. Guillen and H. Pomares, Automatic Knot Adjustment for B-spline Smoothing Approximation Using Improved Clustering Algorithm, *2013 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, Hyderabad, India, July 7-10, 2013.

[56] H. Schwetlick and T. Schutze, Least Squares Approximation by Splines with Free Knots, *BIT Numerical Mathematics*, Vol.35, No.1, pp.361–384, 1995.

Publications

Award

1. 2017 President Award Fukuoka Institute of Technology, October, 2017.

2. 2016 Excellent Presentation Award of *the IEEE Fukuoka Section Award Winners*, March, 2017.

Journal

1. R. Soontornvorn and H. Fujioka, Design of Compact Planar B-spline Curves Using DP Control Point Selection with Multi-Level Error Functions - Towards Usability Improvement in Design of Dynamic Font-based Characters, *IEIE Transactions on Smart Processing & Computing*, Vol.8, No.2, pp.108-120, 2019.

International Conferences

1. R. Soontornvorn, H. Fujioka and H. Kano, DP-based Control Point Selection of Periodic Splines and Its Application to Object Contour Modeling, *Extended Abstracts of the 51th ISCIE International Symposium on Stochastic Systems Theory and Its Applications (SSS'19)*, pp.23-24, Aizu-Wakamatsu, Fukushima, Japan, Nov. 1-2, 2019.

2. H. Fujioka, R. Soontornvorn and H. Kano, Constructing Compact Cubic B-spline Curves Using DP-based Dominant Control Point Selection, *Extended Abstracts of the 50th ISCIE International Symposium on Stochastic Systems Theory and Its Applications (SSS '18)*, pp.129-130, Kyoto, Japan, Nov. 1-2, 2018.

3. R. Soontornvorn and H. Fujioka, Design of Compact Planar B-spline Curves Using DP Control Point Selection-Towards Usability Improvement in Design of Dynamic Font-based Characters, *Proc. the 2018 IEEE Region 10 Conference (TENCON2018)*, pp.1470-1473, Jeju, Korean, Oct. 28-31, 2018.

4. J. Sawangphol, R. Soontornvorn, H. Fujioka, S. Anraku, N. Miyamoto, T. Kato, H. Kino, A. Hidaka and H. Kano, Toward an Understanding of Nanosheet Object Motion from Noisy Microscopy Images Using Deep-Learning Approach (invited paper), *Abstracts of the 5th International Conference on Nanomechanics and Nanocomposites (ICNN5)*, pp.89, Fukuoka, Japan, Aug. 22-25, 2018.

5. H. Fujioka, R. Soontornvorn and H. Kano, Design of Compact B-spline Curves Using Optimal Control Point Selection, *Extended Abstracts of the 49th ISCIE International Symposium on Stochastic Systems Theory and Its Applications (SSS '17)*, pp.43-44, Hiroshima, Japan, Nov. 3-4, 2017.

6. R. Soontornvorn, H. Fujioka, V. Chutchavong and K. Janchitrapongvej, Modeling ECG waveform using optimal smoothing Bezier-Bernstein curves, *Proc. the 2017 IEEE Region 10 Conference (TENCON2017)*, pp.1235-1238, Penang, Malaysia, Nov. 5-8, 2017.

Other Conferences

1. R. Soontornvorn and H. Fujioka, DP-based Control Point Selection of Periodic Splines for Compact Contour Modeling, *The 72th Joint Conference of Electrical, Electronics and Information Engineers in Kyushu*, pp.117, Fukuoka, Japan, Sept. 27-28, 2019.

2. R. Soontornvorn and H. Fujioka, Constructing Compact Planar B-spline Curves Using Dynamic Programming-based Control Point Selection, *The 71th Joint Conference of Electrical, Electronics and Information Engineers in Kyushu*, pp.253, Oita, Japan, Sept. 27-28, 2018.

3. J. Sawangphol, R. Soontornvorn and H. Fujioka, Design of Hairy Brush Characters Using Convolutional Encoder-Decoder Network, *The 71th Joint Conference of Electrical, Electronics and Information Engineers in Kyushu*, pp.252, Oita, Japan, Sept. 27-28, 2018.

4. R. Soontornvorn and H. Fujioka, Optimal Modeling of ECG Waveform using Smoothing Bezier-Bernstein Curves, *The 70th Joint Conference of Electrical, Electronics and Information Engineers in Kyushu*, pp.198, Okinawa, Japan, Sept. 27-28, 2017.