A Note on the Connection Between Bezier Curves and Linear Optimal Control

Magnus B. Egerstedt and Clyde F. Martin

***Abstract*—We show that Bezier curves can in fact be thought of as the solutions to linear optimal control problems, using results from Hermite interpolation in combination with traditional linear optimal control. This provides us with a computational view of Bezier curves that differs from the standard DeCasteljau Algorithm, and it furthermore points out the close relationship between Bezier curves and interpolating dynamic splines.**

***Index Terms*—Bezier curves, hermite interpolation, optimal control.**

I. Bezier Curves

Bezier curves constitute a class of approximating curves in that they are defined using control points, but do not necessarily pass through these control points. Instead the control points define the shape of the curve as

$$B(t) = \sum_{i=0}^{N} B_{N,i}(t)p_i \tag{1}$$

where $p_i \in \mathbb{R}^p, i = 0, \ldots, N$ are the control points, and

$$B_{N,i}(t) = \binom{N}{i}(1-t)^{N-i}t^i \tag{2}$$

is a Bernstein polynomial. It is immediately clear from (2) that the Bezier curves are parameterized by t, taking on values between 0 and 1.

From (2), it furthermore follows that we need $N+1$ control points in order to define a Bezier curve of degree N. Given $N+1$ such control points in $\mathbb{R}^p$, the Bezier curves can be established by an iterative algorithm, the DeCasteljau Algorithm, that produces a single point on the curve for each iteration of the algorithm. The construction is shown in Fig. 1, where the points $p_0, \ldots, p_4$ are the control points in (1). The curve is produced by letting λ sweep [0, 1] as follows. The control points are connected with lines, and new points are defined on those lines at a fraction λ of the distance between the endpoints of the individual lines. In Fig. 1, those points are p_{01}, p_{12}, p_{23}, and p_{34}. This procedure is repeated, generating the points p_{012}, p_{123}, and p_{234} in the second step, and p_{0123} and p_{1234} in the third step. The final point $p_{01\,234}$ is a point on the Bezier curve, and, in this particular case, we have that $p_{01\,234} = B(\lambda)$.

The existence of this computationally inexpensive algorithm is what makes the Bezier curves useful in a number of applications (see, for example, [1]), and they are used extensively in computer graphics, as well as in such areas as computer aided design. There is a wealth of literature associated with this topic, and the books by Farin [3], [4] are standard references for this subject. However, what will be shown in this paper is that while the DeCasteljau algorithm is elegant, the Bezier curve is in fact a fundamental object from linear control theory.

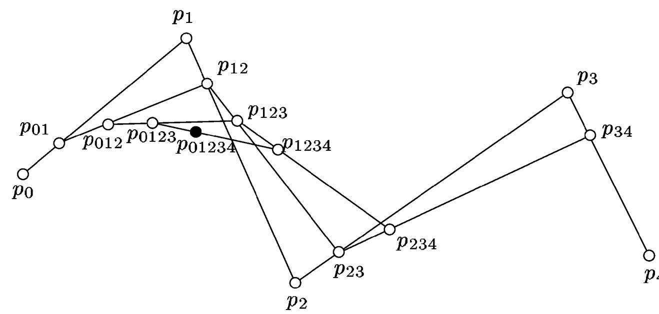

Fig. 1. Standard construction of a single point on the Bezier curve.

Now, the question has been raised of whether or not the Bezier curves are intrinsically better than the dynamic splines discussed in [5]–[10]. In this paper we will show that the Bezier curves are in fact intimately associated with a linear optimal control problem and, thus, that they can be realized as the solution to just such a problem. What this means is that before it can be claimed that Bezier curves offer better performance than other spline methods, further research is required. In producing this result, we will rely heavily on the fact that the Bezier curves can be related to certain Hermite interpolation problems, and that in turn these Hermite interpolation problems are in fact linear optimal control problems.

II. Preliminaries

From (2), it is easy to calculate the derivatives of a Bezier curve using the differential recursion for the Bernstein polynomials, given by

$$\frac{d}{dt}B_{N,i}(t) = NB_{N-1,i-1}(t) - NB_{N-1,i}(t), \quad i = 1, \ldots, N-1, \tag{3}$$

and

$$\begin{aligned}\frac{d}{dt}B_{N,0}(t) &= -NB_{N-1,0}(t)\\ \frac{d}{dt}B_{N,N}(t) &= NB_{N-1,N-1}(t).\end{aligned} \tag{4}$$

Thus, we can calculate the derivative of the Bezier curve in (1) as

$$\frac{d}{dt}B(t) = N\sum_{i=0}^{N-1} B_{N-1,i}(t)(p_{i+1} - p_i). \tag{5}$$

From (5), it follows that the derivative is in itself a scalar multiple of a Bezier curve, calculated from the differences of the original control points.

It is now straightforward to calculate all of the derivatives of the curve. As is shown in [3] and [4], these derivatives have a very nice closed form expression in terms of the forward differencing operators Δ_F^k, defined recursively as

$$\begin{aligned}\Delta_F^k p_j &= \Delta_F^{k-1}p_{j+1} - \Delta_F^{k-1}p_j, \quad k = 1, 2, \ldots\\ \Delta_F^0 p_j &= p_j.\end{aligned} \tag{6}$$

Thus, the kth derivative is given by

$$\frac{d^k}{dt^k}B(t) = \frac{N!}{(N-k)!}\sum_{i=0}^{N-k} B_{N-k,i}(t)\Delta_F^k p_i. \tag{7}$$

Manuscript received May 15, 2001; revised April 4, 2002. Recommended by Associate Editor C. Oara. The work of M. B. Egerstedt was supported in part by the U.S. Army Research Office under Grant Number DAAG 5597-1-0114, and in part by the Sweden-America Foundation 2000 Research Grant. The work of C. F. Martin was supported in part by the National Science Foundation under Grants ECS 9720357 and ECS 9705312.

M. B. Egerstedt is with the Electrical and Computer Engineering Department, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: magnus@ece.gatech.edu).

C. F. Martin is with the Department of Mathematics and Statistics, Texas Tech University, Lubbock, TX 79409 USA (e-mail: martin@math.ttu.edu).

Digital Object Identifier 10.1109/TAC.2004.835393

We now note that only one of the Bernstein polynomials in (1) is nonzero at $t = 0$, namely $B_{N,0}(t)$, with $B_{N,0}(0) = 1$. Hence

$$\begin{aligned} B(0) &= p_0 \\ \frac{d}{dt}B(0) &= N\Delta_F p_0 \\ &\vdots \\ \frac{d^k}{dt^k}B(0) &= \frac{N!}{(N-k)!}\Delta_F^k p_0. \end{aligned} \tag{8}$$

With this formulation, it is possible to calculate the derivatives of the Bezier curve at the two end points p_0 and p_N. These two points, p_0 and p_N, have a special significance in the Bezier curve construction since they are the only two points that the Bezier curve is guaranteed to interpolate. From (8), we see that the interior points determine the derivatives of the Bezier curve at the two endpoints. This suggests that the Bezier curves are associated with certain Hermitian interpolation problems. In fact, taking N derivatives gives the following two vectors:

$$\begin{pmatrix} p_0 \\ N\Delta_F p_0 \\ \vdots \\ N!\Delta_F^N p_0 \end{pmatrix} \in \mathbb{R}^{p(N+1)}, \quad \begin{pmatrix} p_N \\ N\Delta_B p_N \\ \vdots \\ N!\Delta_B^N p_N \end{pmatrix} \in \mathbb{R}^{p(N+1)} \tag{9}$$

of derivatives at the endpoints that we would like to interpolate. Here, Δ_B is the backward differencing operator, and it should be pointed out that at this point we do not know if we in fact need to take all N derivatives at the endpoints in order to generate the correct Hermite interpolation problem.

III. Linear Control Theory

We consider a linear system of the form

$$\dot{x} = \mathcal{A}x + \mathcal{B}u, \quad y = \mathcal{C}x \tag{10}$$

where $u, y \in \mathbb{R}^p$ and $x \in \mathbb{R}^{pq}$ for some q to be determined later. Furthermore, we let $\mathcal{A}$ have the form

$$\mathcal{A} = \begin{pmatrix} A & 0 & \cdots & 0 \\ 0 & A & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & & & \\ & \cdots & & A \end{pmatrix} \tag{11}$$

where A is the $q \times q$ nilpotent matrix

$$A = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & 1 \\ 0 & \cdots & \cdots & 0 & 0 \end{pmatrix}. \tag{12}$$

Furthermore, we let $\mathcal{B}$ be the matrix

$$\mathcal{B} = (e_q, e_{2q}, \ldots, e_{pq}), \tag{13}$$

where e_k is the kth unit vector in $\mathbb{R}^{pq}$, with 1 in the kth position. The matrix $\mathcal{C}$ is similarly given as

$$\mathcal{C} = \begin{pmatrix} e_1^T \\ e_{1+q}^T \\ \vdots \\ e_{1+(p-1)q}^T \end{pmatrix}. \tag{14}$$

The system in (10) thus consists of p single-input–single-output (SISO) linear systems in parallel, and hence it suffices to consider the subsystems individually. The algorithm for constructing the Bezier curve is also a coordinate-wise algorithm. Thus, we will, without loss of generality, consider the SISO linear system

$$\dot{x} = Ax + bu, \qquad y = cx, \tag{15}$$

where A is described previously, $b = e_q$ (qth unit vector in $\mathbb{R}^q$), and $c = e_1^T$ (first unit vector in $\mathbb{R}^q$).

A classical result for this type of system is given by the following.

Theorem III.1: Let $x(0), x(1) \in \mathbb{R}^q$. Then the control, $u(t)$, that minimizes

$$J(u) = \int_0^1 u^2(t)\,dt \tag{16}$$

and drives the controllable system

$$\dot{x} = Ax + bu \tag{17}$$

from $x(0)$ to $x(1)$ is given by

$$b^T e^{A^T(1-t)} \left(\int_0^1 e^{A(1-s)} bb^T e^{A^T(1-s)}\,ds \right)^{-1} (x(1) - e^{A1}x(0)). \tag{18}$$

We now recall that if A is given in (12) we have that the exponential is a polynomial matrix. In fact,

$$e^{At} = \begin{pmatrix} 1 & t & \frac{t^2}{2!} & \cdots & \frac{t^{p-1}}{(p-1)!} \\ 0 & 1 & t & \cdots & \frac{t^{p-2}}{(p-2)!} \\ \ddots & \cdots & \ddots & \ddots & \\ 0 & \cdots & & & 1 \end{pmatrix}. \tag{19}$$

Thus, the control in Theorem III.1 is polynomial, and the degree of the control is $q - 1$.

IV. Two-Point Hermite Interpolation

The classical Hermite interpolation problem is discussed in great detail in almost every elementary numerical analysis book (in particular, see [2]). We are interested in a specific form of the general problem, namely the two-point problem. At time 0 we specify k values $a_0, a_1, \ldots, a_{k-1}$, and at time 1 we likewise specify k values $b_0, b_1, \ldots, b_{k-1}$.

The problem we are interested in is to find a polynomial of minimum degree such that $p^{(i)}(0) = a_i$ and $p^{(i)}(1) = b_i, i = 0, \ldots, k-1$. It is easy to see that there exists a unique polynomial of degree less than or equal to $2k - 1$ which satisfies the requirement.

In the previous section, we saw that there exists a control law that drives a system from a point in $\mathbb{R}^q$ to another point in $\mathbb{R}^q$ in such a way that the resulting trajectory is polynomial. Using the notation of Theorem III.1, we see that since $u(t)$ is a polynomial of degree at most

$q-1$, $y(t)$ is in fact also a polynomial of degree at most $2q-1$. The output, $y(t)$, thus satisfies the constraints of the Hermite interpolation problem discussed previously, in the case when $k=q$. Thus, we can conclude that the linear optimal control problem and the special, two-point Hermite problem have the same solution.

V. Main Results

We saw previously that given $N+1$ points in $\mathbb{R}^p$, the Bezier curve is a polynomial curve of degree N. Since the Bezier algorithm operates at the level of coordinates we can restrict ourselves to the case $p=1$, which corresponds to only using the subsystem in (15) instead of the full system in (10). For the continuation of this section, we consider two cases based on the parity of N.

A. Case 1: $\mathbf{N = 2M - 1}$

If we want to produce a solution to the Hermite interpolation problem that has the same degree as the Bezier curve, we need to interpolate between points in $\mathbb{R}^k$ such that $2k-1=N$, i.e., $k=M$. In order to produce such interpolation points in $\mathbb{R}^M$, we need to compute $M-1$ derivatives of the Bezier curve.

Using the expression for the derivative of the Bezier curve in (7), we have that the derivatives of the curve at $t=0$ are given by the vector

$$x(0)=\begin{pmatrix} p_0 \\ (2M-1)\Delta_F p_0 \\ \vdots \\ \frac{(2M-1)!}{M!}\Delta_F^{M-1}p_0 \end{pmatrix} \in \mathbb{R}^M \tag{20}$$

where we have now assumed that $p_0 \in \mathbb{R}$. The corresponding first derivatives at time 1 are given by

$$x(1)=\begin{pmatrix} p_{2M-1} \\ (2m-1)\Delta_B p_{2M-1} \\ \vdots \\ \frac{(2M-1)!}{M!}\Delta_B^{M-1}p_{2M-1} \end{pmatrix} \in \mathbb{R}^M. \tag{21}$$

Now, the linear system of Theorem III.1 that drives (15) between $x(0)$ and $x(1)$, produces an output curve of degree $2M-1$, which is equal to N, i.e., the degree of the Bezier curve. However, by the Hermite problem, this curve is unique and, hence, the Bezier curve and the curve produced by the linear optimal control law are one and the same.

The case when $N=2M$ is a bit more involved.

B. Case 2: $\mathbf{N = 2M}$

As before, the degree of the Bezier curve in (1) is N, which is equal to $2M$. Let us now proceed by taking M instead of $M-1$ derivatives in order to get the two endpoints in the Hermite interpolation problem. We get

$$x(0)=\begin{pmatrix} p_0 \\ 2M\Delta_F p_0 \\ \vdots \\ \frac{(2m)!}{M!}\Delta_F^{M}p_0 \end{pmatrix} \in \mathbb{R}^{M+1} \tag{22}$$

$$x(1)=\begin{pmatrix} p_{2M} \\ 2M\Delta_B p_{2M} \\ \vdots \\ \frac{(2M)!}{M!}\Delta_B^{M}p_{2M} \end{pmatrix} \in \mathbb{R}^{M+1}. \tag{23}$$

Now, it is a well-posed problem to construct the polynomial of minimal degree that solves the Hermite interpolation problem. Since we are interpolating in $\mathbb{R}^{M+1}$, we get an upper bound of $2(M+1)-1=2M+1$ on the degree of the polynomial obtained from the Hermite interpolation. However, the Bezier curve also interpolate the same data using a polynomial of degree $2M$. Hence, in this case, the degree of the unique Hermite polynomial is actually $2M$ instead of the generic degree $2M+1$.

We can also construct a polynomial that interpolates this data using linear control theory. Again, using Theorem III.1, with the degree of the system being $M+1$, we construct a polynomial of degree $2M+1$ that interpolates the data. Again appealing to the uniqueness of the Hermite interpolation problem we conclude that the degree is actually $2M$.

C. Main Theorem

Based on the observations in the previous two subsections, we have established the following fact that we state as a theorem.

Theorem V.1: Let $\{p_i : i=0,\ldots,N\}$ be a set of $N+1$ points in $\mathbb{R}$, with the corresponding Bezier curve

$$B(t)=\sum_{i=0}^{N} B_{N,i}p_i \tag{24}$$

where

$$B_{N,i}=\binom{N}{i}(1-t)^{N-i}t^i. \tag{25}$$

Let the function $y(t)$ be given by

$$\begin{aligned}\dot{x}&=Ax+bu\\ y&=cx\end{aligned} \tag{26}$$

where A, b, c are given in (15), and where $u(t)$ solves

$$\min_u \int_0^1 u^2(t)\,dt \tag{27}$$

while interpolating

$$x(0)=\begin{pmatrix} p_0 \\ N\Delta_F p_0 \\ \vdots \\ \frac{N!}{(N-m)!}\Delta_F^{m}p_0 \end{pmatrix} \in \mathbb{R}^{m+1} \tag{28}$$

$$x(1)=\begin{pmatrix} p_N \\ N\Delta_B p_N \\ \vdots \\ \frac{N!}{(N-m)!}\Delta_B^{m}p_N \end{pmatrix} \in \mathbb{R}^{m+1}. \tag{29}$$

Then, $y(t)$ is in fact identical to the Bezier curve $B(t)$, with the choice of $m=\lfloor (N/2)\rfloor$, with $\lfloor\cdot\rfloor$ being the floor operator.

From Theorem V.1, it now follows that it is possible to use the Bezier curves for constructing interpolating splines. However, the control points must be chosen to insure continuity of the derivatives at each of the nodes, and the procedure for doing this is exactly the procedure used in [9]. Hence, the usual claim that Bezier splines are cheaper than the interpolating splines to compute is probably not true if it is desired to have a closed form for the resulting spline, and there is more than two interpolating nodes. In this paper we make no claims that dynamic splines are better than Bezier curves, but we have shown that Bezier curves are in fact a fundamental construction of linear control theory.

It should be noted that when producing dynamic smoothing splines, more than two interpolation points can be introduced without any major increase in computational burden, while the Bezier curve construction, using the DeCasteljau algorithm, becomes cumbersome as more than two points are interpolated. (By this we mean that the DeCasteljau algorithm is straightforward only when we have two endpoints. The introduction of more intermediary control points, however, does not change the nature of the problem since we have seen that the intermediary control points simply determine the endpoint derivatives.) There has furthermore been a renewed interest, in the control literature, in dynamic smoothing splines [7], [9], [10], i.e., in splines that do not interpolate the data exactly, due to noise corrupted data points. Such smoothing curves do not have a corresponding Bezier curve formulation, which furthermore somewhat restricts the generality of the Bezier curve construction.

References

[1] P. Crouch, G. Kun, and F. S. Leite, "The DeCasteljau algorithm on lie groups and spheres," *J. Dyna. Control Syst.*, vol. 5, no. 3, pp. 397–429, 1999.

[2] P. Davis, *Interpolation and Approximation*. New York: Dover, 1975.

[3] G. Farin, *Curves and Surfaces for Computer-Aided Geometric Design: A Practical Guide*, 4th ed. San Diego, CA: Academic, 1997.

[4] ——, *NURBS: From Projective Geometry to Pracitcal Use*, 2nd ed. Natick, MA: A. K. Peters, 1999.

[5] C. Martin, P. Enqvist, J. Tomlinson, and Z. Zhang, "Linear control theory, splines and interpolation," in *Computation and Control IV*, J. Lund and K. Bowers, Eds. Boston, MA: Birkhäuser, 1995, pp. 269–288.

[6] L. L. Schumaker, *Spline Functions: Basic Theory*. New York: Wiley, 1981.

[7] S. Sun, M. Egerstedt, and C. Martin, "Control theoretic smoothing splines," *IEEE Trans. Automat. Contr.*, vol. 45, pp. 2271–2279, Dec. 2000.

[8] G. Wahba, "Spline models for observational data," in *CBMS-NSF Regional Conference Series in Applied Mathematics*, vol. 59. Philadelphia, PA, 1990.

[9] Z. Zheng, J. Tomlinson, and C. Martin, "Splines and linear control theory," *Acta Applicandae Math.*, vol. 49, pp. 1–34, 1997.

[10] Y. Zhou, M. Egerstedt, and C. Martin, "Optimal approximation of functions," *Commun. Inform. Syst.*, vol. 1, no. 1, pp. 101–112, 2001.

An Improved Second-Order Sliding-Mode Control Scheme Robust Against the Measurement Noise

Giorgio Bartolini, Alessandro Pisano, and Elio Usai

***Abstract*—In this note, the second-order sliding-mode control problem is tackled by explicitly taking into account the presence of measurement error with *unknown* upper bound δ. Under sensible assumptions regarding the uncertain dynamics of a broad class of nonlinear plants, a new switching controller is proposed guaranteeing a sliding accuracy of order $O(\delta)$. Simulations highlight the robustness and good performance of the proposed approach, and confirm the expected precision order.**

***Index Terms*—Measurement noise, nonlinear systems, second-order sliding modes, uncertain systems.**

I. Introduction and Problem Formulation

Consider the following nonlinear uncertain system affine in the scalar control variable u:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t) + \mathbf{g}(\mathbf{x}, t)u, \qquad \mathbf{x} \in R^n, \ \ u \in R \tag{1}$$

where $\mathbf{x}$ is the state vector, t is time, and $\mathbf{f}$ and $\mathbf{g}$ are n-dimensional uncertain vector fields. The objective is to enforce to zero, possibly in a finite time, the measurable sliding (or constraint) variable $s = s(\mathbf{x}, t)$. Assume that the sliding variable has a globally defined relative degree two [7], which implies that the second derivative of s can be expressed as

$$\ddot{s} = \varphi(\mathbf{x}, t) + \gamma(\mathbf{x}, t)u \tag{2}$$

where the control gain function γ is always separated from zero. It is often assumed that the sign of γ is known and that the uncertainties satisfy sensible structural and/or boundedness conditions such that the effective application of Lyapunov-based or classical first-order sliding-mode control (1-SMC) design methods [6], [11] is possible. Their main drawback reveal when the relative degree r of the measurable quantity s to regulate is higher than one, as they generally require the knowledge of its derivatives up to the $(r-1)$th order. As far as the problem under investigation in concerned $(r = 2)$, the usually not-measurable $\dot{s}$ must be estimated by means of some observer (e.g., "high-gain" observer [6] or sliding differentiator [9]).

Let the measured value of s, say $\hat{s}$, differ from the actual one because of a bounded additive measurement noise n, i.e.,

$$\hat{s}(t) = s(t) + n(t), \quad |n(t)| \leq \delta. \tag{3}$$

A possible usual choice for the relative degree one signal to regulate is $\sigma = \hat{\dot{s}} + cs \ (c > 0)$, and its estimation in the presence of noise actually deteriorates the accuracy, since the error affecting the estimated derivative $\hat{\dot{s}}$ cannot be less than $O(\sqrt{\delta})$[5], [9], and might even cause instabilities if the observer/controller parameters are not properly set.

The second-order SMC (2-SMC) approach [1] solves the stabilization problem for (2) by requiring the knowledge of s and just the *sign* of $\dot{s}$. A class of 2-SMC algorithms derived from a generalization of the

Manuscript received March 5, 2003; revised January 20, 2004 and May 5, 2004. Recommended by Associate Editor M. Reyhanoglu.

The authors are with the Department of Electrical and Electronic Engineering (DIEE), University of Cagliari, Cagliari I-09132, Italy (e-mail: (giob@diee.unica.it; pisano@diee.unica.it; eusai@diee.unica.it).

Digital Object Identifier 10.1109/TAC.2004.835395