Optimal Control and Monotone Smoothing Splines

Magnus Egerstedt[1] and Clyde Martin[2]

[1] School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA, magnus@ece.gatech.edu

[2] Department of Mathematics and Statistics, Texas Tech University, Lubbock, TX 79409, USA, martin@math.ttu.edu

Summary. The solution to the problem of generating curves by driving the output of a particular, nilpotent single-input, single-output linear control system close to given waypoints is analyzed. The curves are furthermore constrained by an infinite dimensional, non-negativity constraint on one of the derivatives of the curve. The main theorem in this paper states that the optimal curve is a piecewise polynomial of known degree, and for the two-dimensional case, this problem is completely solved when the acceleration is controlled directly. The solution is obtained by exploiting a finite reparameterization of the problem, resulting in a dynamic programming formulation that can be solved analytically.

1 Introduction

When interpolating curves through given data points, a demand that arises naturally when the data is noise contaminated, is that the curve should pass close to the interpolation points instead of demanding exact interpolation. This means that outliers will not be given too much attention, which could otherwise potentially corrupt the shape of the curve. In this paper, we investigate this type of interpolation problem from an optimal control point of view, where the interpolation task is reformulated in terms of choosing appropriate control signals in such a way that the output of a given, linear control system defines the desired interpolation curve. The curve is obtained by minimizing the energy of the control signal, and we, furthermore, deal with the outliers-problem by adding quadratic penalties to the energy cost functional. In this manner, deviations from the data points are penalized in order to produce smooth output curves [5, 14]. The fact that we minimize the energy of the control input, while driving the output of the system close to the interpolation points, tells us that the curves that we are producing belong to a class of curves that in the statistics literature is referred to as *smoothing splines* [14, 15].

However, in many cases, this type of construction is not enough since one sometimes want the curve to exhibit a certain structure, such as monotonicity or convexity. These properties correspond to non-negativity constraints on the first and second derivative of the curve respectively, and hence the non-negative derivative constraint will be the main focus of this paper. Our main theorem, Theorem 2, states that the optimal curve is a piecewise polynomial of known degree, and we will show how the corresponding infinite dimensional constraint (it has to hold for all times) can be reformulated and solved in a finite setting based on dynamic programming.

That piecewise polynomial splines are the solutions to a number of different optimal control problems is a well-known fact [11]. However, in [11], the desired characteristics of the curves were formulated as constraints, while this paper investigates how the introduction of least-square terms in the cost function affects the shape of the curve. Furthermore, in [13] only the optimality conditions (necessary as well as sufficient) were studied, but it is in general not straight forward to go from a maximum principle to a numerically tractable algorithm, which is the case in this paper. The problem of monotone interpolation has furthermore been extensively studied in the literature. In [8, 9] the problem of exact interpolation of convex or monotone data points using monotone polynomials is investigated. Questions concerning existence and convergence of such interpolating polynomials have been studied in [4, 7, 12]. Those results are in general not constructive in so far as they can not be readily implemented as a numerical algorithm. In [3], however, a numerical algorithm for producing monotone, interpolating polynomials is developed, even though no guarantees that the monotonicity constraint is respected for all times is given. What is different in this paper is first of all that we focus exclusively on producing monotone, *smoothing* curves, i.e. we do not demand exact interpolation. Secondly, we want our solution to be constructive so that it can be implemented as a numerically sound algorithm with guaranteed performance in the case when the curve is generated by a second order system. The main contribution in this paper is thus that we show how concepts, well studied in control theory, such as minimum energy control and dynamic programming, give us the proper tools for shedding some new light on the monotone smoothing splines problem. The outline of this paper is as follows: In Sections 2 and 3, we describe the problem and derive some of the properties that the optimal solution exhibits. We then, in Section 4, show how the problem can be reparameterized as a finite dimensional dynamic programming problem. In Section 5, we give the exact solution to the monotone interpolation problem when the underlying dynamics is given by a particular second order system.

2 Problem Description

Consider the problem of constructing a curve that passes close to given data points, at the same time as we want the curve to exhibit certain monotonicity properties. In other words, if $p(t)$ is our curve, we want $(p(t_i) - \alpha_i)^2$, $i = 1, \ldots, m$ to be qualitatively small. Here, $(t_1, \alpha_1), \ldots, (t_m, \alpha_m)$ are the data points, with $\alpha_i \in \Re$, $i = 1, \ldots, m$, and $0 < t_1 < t_2 < \ldots < t_m \leq T$, for some given terminal time $T > 0$. We do not only, however, want to keep the interpolation errors small. We also want the curve to vary in a smooth way, as well as

$$p^{(n)}(t) \geq 0, \ \forall t \in [0, T], \tag{1}$$

for some given, positive integer n.

Let

$$\begin{aligned} A &= \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix} \\ b &= \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \\ c_1 &= \begin{pmatrix} 1 & 0 & \cdots & 0 \end{pmatrix} \\ c_2 &= \begin{pmatrix} 0 & 0 & \cdots & 1 \end{pmatrix}, \end{aligned} \tag{2}$$

where A is an $n \times n$-matrix, b is $n \times 1$, and c_1 and c_2 are $1 \times n$. Then, by using the standard notation from control theory [2], our problem can be cast as

$$\inf_u \left\{ \frac{1}{2} \int_0^T u^2(t) dt + \frac{1}{2} \sum_{i=1}^m \tau_i (c_1 x(t_i) - \alpha_i)^2 \right\}, \tag{3}$$

subject to

$$\begin{cases} \dot{x} = Ax + bu, \ x(0) = 0 \\ u \in L^2[0, T] \\ c_2 x(t) \geq 0, \ \forall t \in [0, T], \end{cases} \tag{4}$$

where $\tau_i \geq 0$ reflects how important it is that the curve passes close to a particular $\alpha_i \in \Re$. Here, $c_1 x(t)$ takes on the role of $p(t)$, and by our particular choices of A and b in Equation 2, x is a vector of successive derivatives. It is furthermore clear that by keeping the L^2-norm of u small, we get a curve that varies in a smooth way.

Now, if $\dot{x} = Ax + bu$ then $x(t)$ is given by

$$x(t) = e^{At} x(0) + \int_0^t e^{A(t-s)} bu(s) ds,$$

which gives that $c_1x(t_i)$ can be expressed as

$$c_1x(t_i) = \int_0^{t_i} c_1e^{A(t_i-t)}bu(t)dt$$

since $x(0) = 0$. This expression can furthermore be written as

$$c_1x(t_i) = \int_0^T g_i(t)u(t)dt,$$

where we make use of the following linearly independent basis functions:

$$g_i(t) = \begin{cases} c_1e^{A(t_i-t)}b & \text{if } t \leq t_i \\ 0 & \text{if } t > t_i \end{cases} \quad i = 1, \ldots, m. \tag{5}$$

The fact that these functions are linearly independent follows directly from the observation that they vanish at different points.

Our infimization over u can thus be rewritten as

$$\inf_u \left\{ \frac{1}{2}\int_0^T u^2(t)dt + \frac{1}{2}\sum_{i=1}^m \tau_i \left(\int_0^T g_i(t)u(t)dt - \alpha_i \right)^2 \right\}, \tag{6}$$

which is an expression that only depends on u.

Since we want $c_2x(t)$ to be continuous, we let the constraint space be $C[0,T]$, i.e. the space of continuous functions. In a similar fashion as before, we can express $c_2x(t)$ as

$$c_2x(t) = \int_0^t c_2e^{A(t-s)}bu(s)ds = \int_0^t f(t,s)u(s)ds.$$

This allows us to form the associated Lagrangian [10]

$$\begin{aligned} L(u,\nu) = {} & \frac{1}{2}\int_0^T u^2(t)dt + \frac{1}{2}\sum_{i=1}^m \tau_i \left(\int_0^T g_i(t)u(t)dt - \alpha_i \right)^2 \\ & - \int_0^T \int_0^t f(t,s)u(s)dsd\nu(t), \end{aligned} \tag{7}$$

where $\nu \in BV[0,T]$ (the space of functions of bounded variations, which is the *dual space* of $C[0,T]$). The optimal solution to our original optimization problem is thus found by solving

$$\max_{0 \leq \nu \in BV[0,T]} \inf_{u \in L^2[0,T]} L(u,\nu). \tag{8}$$

3 Properties of the Solution

Lemma 1. *Given any triple* $(\tilde{A}, \tilde{b}, \tilde{c})$*, where* $\tilde{A}$ *is an* $n \times n$ *matrix,* $\tilde{b}$ *is* $n \times 1$*, and* $\tilde{c}$ *is* $1 \times n$*. If* $\dot{x} = \tilde{A}x + \tilde{b}u,\ x(0) = 0$*, then the set of controls in* $L^2[0,T]$ *that make the solution to the differential equation satisfy*

$$\tilde{c}x(t) \geq 0,\ \forall t \in [0,T],$$

is a closed, non-empty, and convex set.

Proof: We first show convexity. Given two $u_i(t) \in L^2[0,T],\ i = 1,2$, such that

$$\int_0^t \tilde{c}e^{\tilde{A}(t-s)}\tilde{b}u_i(s)ds \geq 0,\ \forall t \in [0,T],\ i = 1,2,$$

then for any $\lambda \in [0,1]$ we have

$$\int_0^t \tilde{c}e^{\tilde{A}(t-s)}\tilde{b}(\lambda u_1(s) + (1-\lambda)u_2(s))ds \geq 0,\ \forall t \in [0,T],$$

and convexity thus follows.

Now, consider a collection of controls, $\{u_i(t)\}_{i=0}^{\infty}$, where each individual control makes the solution to the differential equation satisfy $\tilde{c}x(t) \geq 0\ \forall t \in [0,T]$, and where $u_i \to \hat{u}$ as $i \to \infty$. But, due to the compactness of $[0,t]$, we have that

$$\lim_{i\to\infty} \int_0^t \tilde{c}e^{\tilde{A}(t-s)}\tilde{b}u_i(s)ds = \int_0^t \tilde{c}e^{\tilde{A}(t-s)}\tilde{b}\hat{u}(s)ds \geq 0, \forall t \in [0,T].$$

The fact that $L^2[0,T]$, with the natural norm defined on it, is a Banach space gives us that the limit, $\hat{u}$, still remains in that space. The set of admissible controls is thus closed.

Furthermore, since $x(0) = 0$, we can always let $u \equiv 0$. This gives that the set of admissible controls is non-empty, which concludes the proof. □

Lemma 2. *The cost functional in Equation 3 is convex in* u*.*

The proof of this lemma is trivial since both terms in Equation 3 are quadratic functions of u.

Lemmas 1 and 2 are desirable in any optimization problem since they are strong enough to guarantee the existence of a unique optimal solution [10], and we can thus replace *inf* in Equation 7 with *min*, which directly allows us to state the following, standard theorem about our optimal control.

Theorem 1. *There is a unique* $u_0 \in L^2[0,T]$ *that solves the optimal control problem in Equation 3.*

We omit the proof of this and refer to any textbook on optimization theory for the details. (See for example [10].)

Lemma 3. *Given the optimal solution u_0. The optimal $\nu_0 \in BV[0,T]$, $\nu_0 \geq 0$, varies only where $c_2x(t) = 0$. On intervals where $c_2x(t) > 0$, $\nu_0(T) - \nu_0(t)$ is a non-negative, real constant.*

Proof: Since $\nu_0(T) - \nu_0(t) \geq 0$, due to the positivity constraint on ν_0, we reduce the value of the Lagrangian in Equation 7 whenever ν_0 changes, except when $c_2x(t) = 0$. But, since ν_0 maximizes $L(u_0, \nu)$, we only allow ν_0 to change when $c_2x(t) = 0$, and the lemma follows. □

Now, before we can completely characterize the optimal control solution, one observation to be made is that

$$c_2x(t) = \begin{pmatrix} 0 & 0 & \cdots & 1 \end{pmatrix} x(t) = \int_0^t u(s)ds,$$

i.e. $f(t,s)$ is in fact equal to 1 in Equation 7. This allows us to rewrite the Lagrangian as

$$\begin{aligned} L(u,\nu) = & \frac{1}{2}\int_0^T u^2(t)dt + \frac{1}{2}\sum_{i=1}^m \tau_i \left(\int_0^T g_i(t)u(t)dt - \alpha_i \right)^2 \\ & - \int_0^T \int_0^t u(s)dsd\nu(t). \end{aligned} \tag{9}$$

By integrating the Stieltjes integral in Equation 9 by parts, we can furthermore reduce the Lagrangian to

$$L(u,\nu) = \frac{1}{2}\int_0^T u^2(t)dt + \frac{1}{2}\sum_{i=1}^m \tau_i (\int_0^T g_i(t)u(t)dt - \alpha_i)^2 - \int_0^T (\nu(T) - \nu(t))u(t)dt, \tag{10}$$

which is a more easily manipulated expression.

Definition 1. *Let $P\mathcal{P}^k[0,T]$ denote the set of piecewise polynomials of degree k on $[0,T]$. Let, furthermore, $\mathcal{P}^k[0,T]$ denote the set of polynomials of degree k on that interval.*

Theorem 2. *The control in $L^2[0,T]$ that minimizes the cost in Equation 3 is in $P\mathcal{P}^n[0,T]$. It furthermore changes from different polynomials of degree n only at the interpolation times, t_i, $i = 1, \ldots, m$, and at times when $c_2x(t)$ changes from $c_2x(t) > 0$ to $c_2x(t) = 0$ and vice versa.*

Proof: Due to the convexity of the problem, and the existence and uniqueness of the solution, we can obtain the optimal controller by calculating the *Fréchet differential* of L with respect to u, and setting this equal to zero for all increments $h \in L^2[0,T]$.

By letting $L_\nu(u) = L(u,\nu)$, we get that

$$\begin{aligned} & \delta L_\nu(u,h) = \lim_{\epsilon \to 0} \tfrac{1}{\epsilon}(L_\nu(u+\epsilon h) - L_\nu(u)) = \\ & \int_0^T \left(u(t) + \sum_{i=1}^m \tau_i (\int_0^T g_i(s)u(s)ds - \alpha_i) g_i(t) - (\nu(T) - \nu(t)) \right) h(t)dt. \end{aligned} \tag{11}$$

For the expression in Equation 11 to be zero for all $h \in L^2[0,T]$ we need to have that

$$u_0(t) + \sum_{i=1}^{m} \tau_i \Big(\int_0^T g_i(s) u_0(s) ds - \alpha_i \Big) g_i(t) - (\nu(T) - \nu(t)) = 0.$$

This especially has to be true for $\nu = \nu_0$, which gives that

$$u_0(t) + \sum_{i=1}^{m} \tau_i \Big(\int_0^T g_i(s) u_0(s) ds - \alpha_i \Big) g_i(t) - C_j = 0, \tag{12}$$

whenever $c_2 x_0(t) > 0$. Here C_j is a constant. The index j indicates that this constant differs on different intervals where $c_2 x_0(t) > 0$.

Now, the integral terms in Equation 12 do not depend on t, while $g_i(t)$ is in $\mathcal{P}^n[0,t_i]$ for $i = 1, \ldots, m$. This, combined with the fact that $\nu_0(T) - \nu_0(t) = C_j$ if $\dot{x}(t) > 0$, directly gives us that the optimal control, $u_0(t)$, has to be in $\mathcal{PP}^n[0,T]$. It obviously changes at the interpolation times, due to the shape of the g_i's, but it also changes if C_j changes, i.e. it changes if $c_2 x_0(t) = 0$. It should be noted that if $c_2 x_0(t) \equiv 0$ on an interval, $\nu_0(t)$ may change on the entire interval, but since $c_2 x_0(t) \equiv 0$ we also have that $u_0(t) \equiv 0$ on the interior of this interval. But a zero function is, of course, polynomial. Thus we know that our optimal control is in $\mathcal{PP}^n[0,t]$, and the theorem follows. □

Corollary 1. *If $n = 2$ then the optimal control is piecewise linear (in $\mathcal{PP}^1[0,T]$), with changes from different polynomials of degree one at the interpolation times, and at times when $c_2 x(t)$ changes from $c_2 x(t) > 0$ to $c_2 x(t) = 0$ and vice versa.*

4 Dynamic Programming

Based on the general properties of the solution, the idea now is to formulate the monotone interpolation problem as a finite-dimensional programming problem that can be dealt with efficiently. If we drive the system $\dot{x} = Ax + bu$, where A and b are defined in Equation 2, between x_i and x_{i+1} on the time interval $[t_i, t_{i+1}]$, under the constraint $c_2 x(t) \geq 0$, we see that we must at least have

$$\begin{aligned} &c_2 x_i \geq 0 \\ &c_2 x_{i+1} \geq 0 \\ &D(x_{i+1} - x_i) \geq 0, \end{aligned} \tag{13}$$

where

$$D = \begin{pmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix},$$

and the inequality in Equation 13 is taken component-wise. We denote the constraints in Equation 13 by

$$\mathcal{D}(x_i, x_{i+1}) \geq 0.$$

Since the original cost functional in Equation 3 can be divided into one interpolation part and one smoothing part, it seems natural to define the following *optimal value function* as

$$\begin{cases} \hat{S}_i(x_i) = \\ \quad \min\limits_{x_{i+1} \mid \mathcal{D}(x_i, x_{i+1}) \geq 0} \left\{ V_i(x_i, x_{i+1}) + \hat{S}_{i+1}(x_{i+1}) \right\} + \tau_i (c_1 x_i - \alpha_i)^2 \\ \\ \hat{S}_m(x_m) = \tau_m (c_1 x_m - \alpha_m)^2, \end{cases} \tag{14}$$

where $V_i(x_i, x_{i+1})$ is the cost for driving the system between x_i and x_{i+1} using a control in $P\mathcal{P}^n[t_i, t_{i+1}]$, while keeping $c_2 x(t)$ non-negative on the time interval $[t_i, t_{i+1}]$.

The optimal control problem thus becomes that of finding $\hat{S}_0(0)$, where we let $\tau_0 = 0$, while α_0 can be any arbitrary number. In light of Theorem 2, this problem is equivalent to the original problem, and if $V_i(x_i, x_{i+1})$ could be uniquely determined, it would correspond to finding the $n \times m$ variables $x_1, \ldots, x_m$, which is a *finite dimensional* reparameterization of the original, infinite dimensional programming problem.

For this dynamic programming approach to work, our next task becomes that of determining the function $V_i(x_i, x_{i+1})$. Even though that is typically not an easy problem, a software package for computing approximations of such monotone polynomials was developed in [3]. In [8, 9] this problem of exact interpolation, over piecewise polynomials, of convex or monotone data points was furthermore investigated from a theoretical point of view. It is thus our belief that showing that the original problem can formulated as a dynamic programming problem involving exact interpolation is a result that is valuable since it greatly simplifies the structure of the problem. It furthermore transforms it to a form that has been extensively studied in the literature.

In the following section, we will show how to solve this dynamic programming problem exactly for a second order system in such a way that the computational burden is kept to a minimum. This work was carried out in detail in [6], and we will, throughout the remainder of this paper, refer to that work for the proofs. Instead we will focus our attention on the different steps necessary for constructing optimal, monotone, *cubic* splines.

5 Example – Second Order Systems

If we change our notation slightly in such a way that our state variable is given by $(x, \dot{x}),\ x, \dot{x} \in \Re$, the dynamics of the system becomes

$$\ddot{x} = u.$$

The optimal value function in Equation 14 thus takes on the form

$$\begin{cases} \hat{S}_i(x_i, \dot{x}_i) = \\ \quad \min_{x_{i+1} \geq x_i, \dot{x}_{i+1} \geq 0} \left\{ V_i(x_i, \dot{x}_i, x_{i+1}, \dot{x}_{i+1}) + \hat{S}_{i+1}(x_{i+1}, \dot{x}_{i+1}) \right\} + \tau_i (x_i - \alpha_i)^2 \\ \\ \hat{S}_m(x_m, \dot{x}_m) = \tau_m (x_m - \alpha_m)^2. \end{cases} \tag{15}$$

5.1 Two-Points Interpolation

Given the times t_i and t_{i+1}, the positions x_i and x_{i+1}, and the corresponding derivatives $\dot{x}_i$ and $\dot{x}_{i+1}$, the question to be answered, as indicated by Corollary 1, is the following: *How do we drive the system between $(x_i, \dot{x}_i)$ and $(x_{i+1}, \dot{x}_{i+1})$, with a piecewise linear control input that changes between different polynomials of degree one, only when $\dot{x}(t) = 0$, in such a way that $\dot{x}(t) \geq 0$ $\forall t \in [t_i, t_{i+1}]$, while minimizing the integral over the square of the control input?* Without loss of generality, we, for notational purposes, translate the system and rename the variables so that we want to produce a curve, defined on the time interval $[0, t_F]$, between $(0, \dot{x}_0)$ and $(x_F, \dot{x}_F)$.

Assumption 1

$$\dot{x}_0, \dot{x}_F \geq 0, \;\; x_F > 0, \;\; t_F > 0.$$

It should be noted that if $x_F = 0$, and either $\dot{x}_0 > 0$ or $\dot{x}_F > 0$, then $\dot{x}(t)$ can never be continuous. This case has to be excluded since we already demanded that our constraint space was $C[0, T]$. If, furthermore, $x_F = \dot{x}_0 = \dot{x}_F = 0$ then the optimal control is obviously given by $u \equiv 0$ on the entire interval.

One first observation is that the optimal solution to this *two-points interpolation problem* is to use standard cubic splines if that is possible, i.e. if $\dot{x}(t) \geq 0$ for all $t \in [0, t_F]$. In this well-studied case [1, 13] we would simply have that

$$x(t) = \frac{1}{6} a t^3 + \frac{1}{2} b t^2 + \dot{x}_0 t, \tag{16}$$

where

$$\begin{pmatrix} a \\ b \end{pmatrix} = \frac{6}{t_F^3} \begin{pmatrix} t_F(\dot{x}_0 + \dot{x}_F) - 2x_F \\ t_F x_F - 1/3 t_F^2 (2\dot{x}_0 + \dot{x}_F) \end{pmatrix}. \tag{17}$$

This solution corresponds to having $\nu(t) = \nu(t_{i+1})$, for all $t \in [t_i, t_{i+1})$ in Equation 10, and it gives the total cost

$$\mathcal{I}_1 = \int_0^{t_F} (at + b)^2 dt = 4 \frac{(\dot{x}_0 t_F^2 - 3 x_F t_F)(\dot{x}_0 + \dot{x}_F) + 3 x_F^2 + t_F^2 \dot{x}_F^2}{t_F^3}, \tag{18}$$

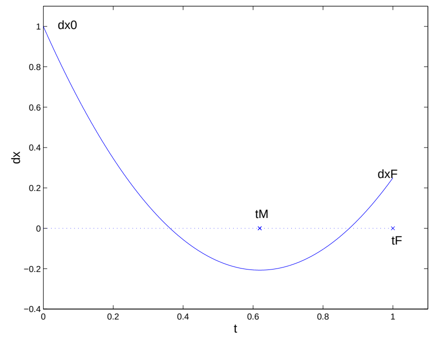

Fig. 1. The case where a cubic spline can not be used if the derivative has to be non-negative. Plotted is the derivative that clearly intersects $\dot{x} = 0$.

where the subscript 1 denotes the fact that only one polynomial of degree one was used to compose the second derivative.

However, not all curves can be produced by such a cubic spline if the curve has to be non-decreasing at all times. Given Assumption 1, the one case where we can not use a cubic spline can be seen in Figure 1, and we, from geometric considerations, get four different conditions that all need to hold for the derivative to be negative. These necessary and sufficient conditions are

$$\begin{array}{rl} (i) & a > 0 \\ (ii) & b < 0 \\ (iii) & \dot{x}(t_M) < 0 \\ (iv) & t_M < t_F, \end{array} \tag{19}$$

where a and b are defined in Equation 16, and t_M is defined in Figure 1.

We can now state the following lemma.

Lemma 4. *Given Assumption 1, a standard cubic spline can be used to produce monotonously increasing curves if and only if*

$$x_F \geq \chi(t_F, \dot{x}_0, \dot{x}_F) = \frac{t_F}{3}(\dot{x}_0 + \dot{x}_F - \sqrt{\dot{x}_0 \dot{x}_F}). \tag{20}$$

The proof of this follows from simple algebraic manipulations [6], and we now need to investigate what the optimal curve looks like in the case when we can not use standard, cubic splines.

5.2 Monotone Interpolation

Given two points such that $x_F < \chi(t_F, \dot{x}_0, \dot{x}_F)$, how should the interpolating curve be constructed so that the second derivative is piecewise linear, with switches only when $\dot{x}(t) = 0$? One first observation is that it is always possible to construct a piecewise polynomial path that consists of three polynomials of degree one that respects the interpolation constraint, and in what follows we will see that such a path also respects the monotonicity constraint.

The three interpolating polynomials are given by

$$u(t) = \begin{cases} a_1 t + b_1 & \text{if } 0 \le t < t_1 \\ 0 & \text{if } t_1 \le t < t_2 \\ a_2(t - t_2) + b_2 & \text{if } t_2 \le t \le t_F, \end{cases} \tag{21}$$

where

$$\begin{aligned} \begin{pmatrix} a_1 \\ b_1 \end{pmatrix} &= \tfrac{6}{t_1^3} \begin{pmatrix} t_1 \dot{x}_0 - 2x_1 \\ t_1 x_1 - 2/3 t_1^2 \dot{x}_0 \end{pmatrix} \\ \begin{pmatrix} a_2 \\ b_2 \end{pmatrix} &= \tfrac{6}{(t_F - t_2)^3} \begin{pmatrix} (t_F - t_2)\dot{x}_F - 2(x_F - x_1) \\ (t_F - t_2)(x_F - x_1) - 1/3(t_F - t_1)^2 \dot{x}_F \end{pmatrix}, \end{aligned} \tag{22}$$

and where $x(t_1) = x(t_2) = x_1$ that, together with t_1 and t_2, is a parameter that needs to be determined.

Assumption 2

$$\dot{x}_0, \dot{x}_F, x_F, t_F > 0.$$

We need this assumption, which is stronger than Assumption 1, in the following paragraph but it should be noted that if $\dot{x}_0 = 0$ or $\dot{x}_F = 0$ we would then just let the first or the third polynomial on the curve be zero.

We now state the possibility of such a feasible three polynomial construction.

Lemma 5. *Given $(t_F, \dot{x}_0, x_F, \dot{x}_F)$ such that $x_F < \chi(t_F, \dot{x}_0, \dot{x}_F)$, then a feasible, monotone curve will be given by Equation 21 as long as Assumption 2 holds. Furthermore, the optimal t_1, t_2, and x_1 are given by*

$$\begin{cases} t_1 = 3\frac{x_1}{\dot{x}_0}, \\ t_2 = t_F - 3\frac{x_F - x_1}{\dot{x}_F}, \\ x_1 = \frac{\dot{x}_0^{3/2}}{(\dot{x}_0^{3/2} + \dot{x}_F^{3/2})} x_F. \end{cases} \tag{23}$$

The proof is constructive and is based on showing that with the type of construction given in Equation 21, the optimal choice of t_1, t_2, x_1 gives a feasible curve. We refer the reader to [6] for the details, and we can thus construct a feasible path, as seen in Figure 2, by using three polynomials whose second derivatives are linear.

Theorem 3 (Monotone Interpolation). *Given Assumption 1, the optimal control that drives the path between $(0, \dot{x}_0)$ and $(x_F, \dot{x}_F)$ is given by Equation 16 if $x_F \ge \chi(t_F, \dot{x}_0, \dot{x}_F)$ and by Equation 21 otherwise.*

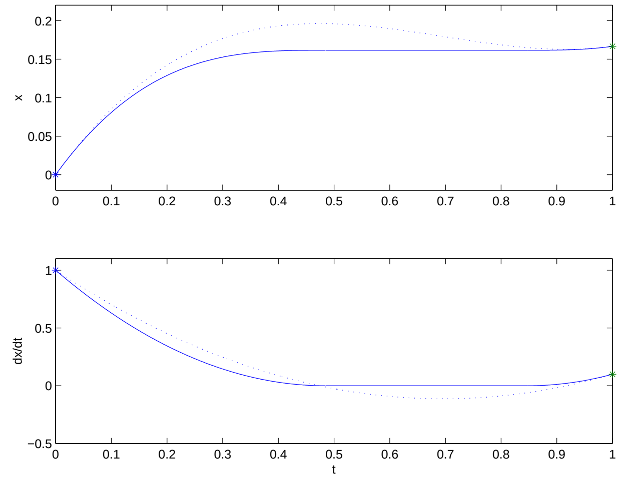

Fig. 2. The dotted line corresponds to a standard, cubic spline, while the solid line shows the three polynomial construction from Lemma 5. Depicted is the position and the velocity.

Proof. The first part of the theorem is obviously true. If we can construct a standard, cubic spline, then this is optimal. However, what we need to show is that when $x_F < \chi(t_F, \dot{x}_0, \dot{x}_F)$ the path given in Equation 21 is in fact optimal.

The cost for using a path given in Equation 21 is

$$\mathcal{I}_3 = \int_0^{t_1} (a_1 t + b_1)^2 dt + \int_{t_2}^{t_F} (a_2(t - t_2) + b_2)^2 dt = \frac{4(\dot{x}_F^{3/2} + \dot{x}_0^{3/2})^2}{9x_F},$$

where the coefficients are given in Equation 23. We now add another, arbitrary polynomial, as seen in Figure 3, to the path as

$$u(t) = \begin{cases} a_1 t + b_1 & \text{if } 0 \leq t < t_1 \\ 0 & \text{if } t_1 \leq t < t_3 \\ a_3(t - t_3) + b_3 & \text{if } t_3 \leq t < t_4 \\ 0 & \text{if } t_4 \leq t < t_2 \\ a_2(t - t_2) + b_2 & \text{if } t_2 \leq t \leq t_F, \end{cases} \tag{24}$$

where $0 < t_1 \leq t_3 \leq t_4 \leq t_2 < t_F$. Furthermore, t_3, t_4, and $x_2 = x(t_4)$ (see Figure 3) are chosen arbitrarily while the old variables, t_1, t_2 and $x_1 = x(t_1)$, are defined to be optimal with respect to the new, translated end-conditions that the extra polynomials give rise to.

After some straight forward calculations, we get that the cost for this new path is

$$\mathcal{I}_5 = \frac{4(\dot{x}_F^{3/2} + \dot{x}_0^{3/2})^2}{9(x_F - x_2)} + \frac{12(x_2 - x_1)^2}{(t_4 - t_3)^3}, \tag{25}$$

where the subscript 5 denotes the fact that we are now using five polynomials of degree one to compose our second derivate. It can be seen that we minimize $\mathcal{I}_5$ if we let $x_2 = x_1$ and make $t_4 - t_3$ as large as possible. This corresponds to letting $t_3 = t_1$ and $t_4 = t_2$, which gives us the old solution from Lemma 5, defined in Equation 21. □

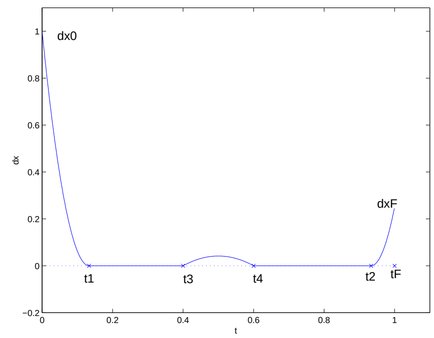

Fig. 3. Two extra polynomials are added to the produced path. Depicted is the derivative of the curve.

5.3 Monotone Smoothing Splines

We now have a way of producing the optimal, monotone path between two points, while controlling the acceleration directly. We are thus ready to formulate the transition cost function in Equation 15, $V_i(x_i, \dot{x}_i, x_{i+1}, \dot{x}_{i+1})$, that defines the cost for driving the system between $(x_i, \dot{x}_i)$ and $(x_{i+1}, \dot{x}_{i+1})$, with minimum energy, while keeping the derivative non-negative.

Based on Theorem 3 we, given Assumption 1, have that[3]

$$V_i(x_i, \dot{x}_i, x_{i+1}, \dot{x}_{i+1}) =$$

$$\begin{cases} 4\frac{\dot{x}_i(t_{i+1}-t_i)^2-3(x_{i+1}-x_i)(t_{i+1}-t_i)(\dot{x}_i+\dot{x}_{i+1})+3(x_{i+1}-x_i)^2+(t_{i+1}-t_i)^2\dot{x}_{i+1}^2}{(t_{i+1}-t_i)^3} \\ \text{if } x_{i+1}-x_i \geq \chi(t_{i+1}-t_i, \dot{x}_i, \dot{x}_{i+1}) \\ \frac{4(\dot{x}_{i+1}^{3/2}+\dot{x}_i^{3/2})^2}{9(x_{i+1}-x_i)} \\ \text{if } x_{i+1}-x_i < \chi(t_{i+1}-t_i, \dot{x}_i, \dot{x}_{i+1}), \end{cases} \tag{26}$$

[3] If $x_{i+1} - x_i = \dot{x}_i = \dot{x}_{i+1} = 0$ then the optimal control is obviously zero, meaning that $V_i(x_i, \dot{x}_i, x_{i+1}, \dot{x}_{i+1}) = 0$.

where $t_0 = x_0 = \dot{x}_0 = 0$.

If we use this cost in the dynamic programming algorithm, formulated in Equation 15, we get the results displayed in Figures 4–6, which shows that our approach does not only work in theory, but also in practice.

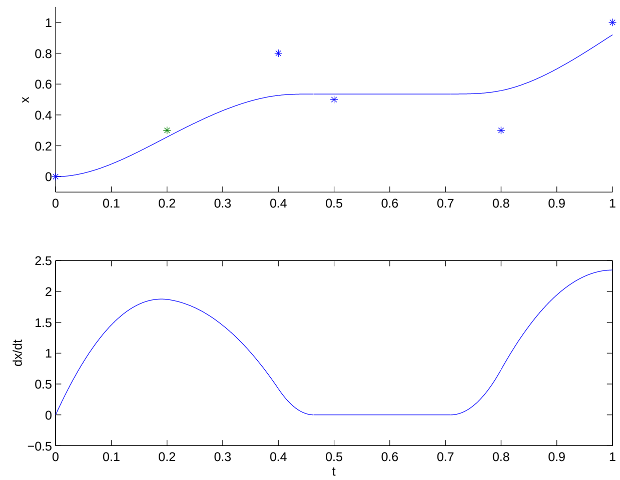

Fig. 4. Monotone smoothing splines with $\tau_i = 1000, i = 1, \ldots, 5$.

6 Conclusions

In this paper we propose and analyze the optimal solution to the problem of driving a curve close to given waypoints. This is done while the state space of the control system used for generating the curve is constrained by an infinite dimensional non-negativity constraint on one of the derivatives of the curve.

This problem is found to support a finite reparameterization, resulting in a dynamic programming formulation that can be solved analytically for the second order case. Simulation results furthermore support our claim that the proposed solution is not just theoretically sound, but also produces a numerically stable algorithm for computing monotone, smoothing splines.

References

1. Ailon A, Segev R (1988) Driving a linear constant system by piecewise constant control. International Journal of Control 47(3):815–825
2. Brockett RW (1970) Finite dimensional linear systems. John Wiley and Sons, Inc., New York

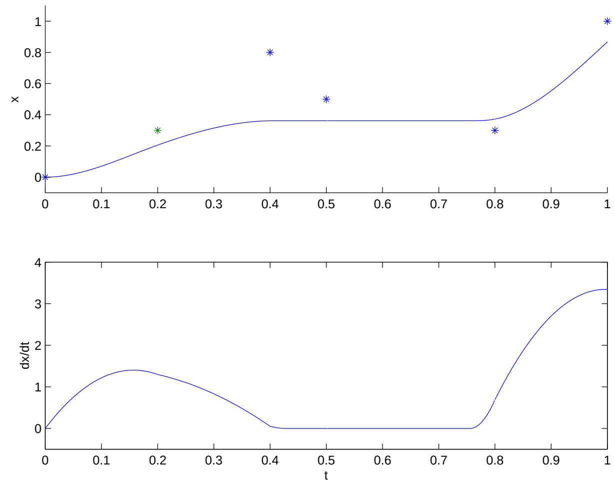

Fig. 5. Monotone smoothing splines with $\tau_4 = 10\tau_i, i \neq 4$ (with $t_4 = 0.8$), resulting in a different curve compared to that in Figure 4 where equal importance is given to all of the waypoints.

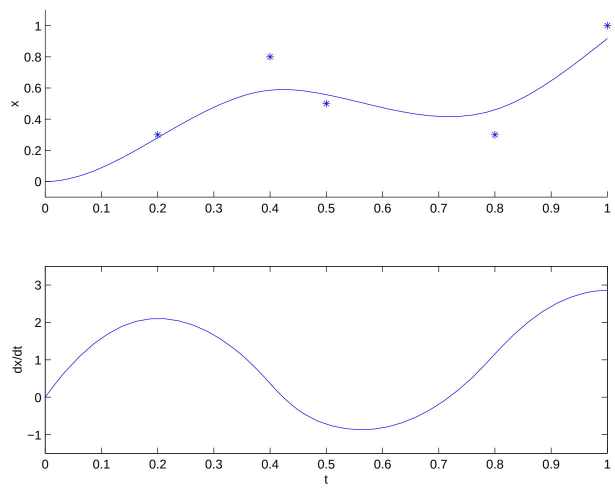

Fig. 6. Smoothing splines, without the monotonicity constraint on the derivative. The curve is produced based on the theory developed in [5, 16].

3. Constantini P (1997) Boundary-valued shape-preserving interpolating splines. ACM Transactions on Mathematical Software 23(2):229–251
4. Darst RB, Sahab S (1983) Approximation of continuous and quasi-continuous functions by monotone functions. Journal of Approximation Theory 38:9–27

5. Egerstedt M, Martin CF (1998) Trajectory planning for linear control systems with generalized splines. Proceedings of the Mathematical Theory of Networks and Systems, Padova, Italy
6. Egerstedt M, Martin CF (2000) Monotone smoothing splines. Proceedings of the Mathematical Theory of Networks and Systems, Perpignan, France
7. Ford WT (1974) On interpolation and approximation by polynomials with monotone derivatives. Journal of Approximation Theory 10:123–130
8. Hornung U (1980) Interpolation by smooth functions under restrictions on the derivatives. Journal of Approximation Theory 28:227–237
9. Iliev GL (1980) Exact estimates for monotone interpolation. Journal of Approximation Theory 28:101–112
10. Luenberger DG (1969) Optimization by vector space methods. John Wiley and Sons, Inc., New York
11. Mangasarian OL, Schumaker LL (1969) Splines via optimal control. Approximation with Special Emphasis on Spline Functions, Schoenberg IJ (Ed.), Academic Press, NY
12. Passow E, Raymon L, Rouler JA (1974) Comotone polynomial approximation. Journal of Approximation Theory 11:221–224
13. Schumaker LL (1981). Spline functions: basic theory. John Wiley and Sons, New York
14. Wahba G (1990) Spline models for observational data. Society for Industrial and Applied Mathematics
15. Wegman EJ, Wright IW (1983) Splines in statistics. Journal of the American Statistical Association 78(382)
16. Zhang Z, Tomlinson J, Martin CF (1997) Splines and linear control theory. Acta Applicandae Mathematicae 49:1–34