Special Issue on the 43rd ISCIE International Symposium on Stochastic Systems Theory and Its Applications—I

Paper

Optimal Vector Smoothing Splines with Coupled Constraints*

Hiroyuki FUJIOKA† and Hiroyuki KANO‡

This paper considers the problem of designing optimal vector smoothing spline curves with equality and/or inequality constraints. The constraints are assumed to be cross-coupled among the element curves imposed at some time instant as well as over some time interval. The vector splines are constituted employing normalized uniform B-splines as the basis functions. Then various types of constraints are formulated as linear function of the so-called control points, and the problem is reduced to convex quadratic programming problem. The performance is examined by some numerical examples.

1. Introduction

Constructing curves from a given set of discrete data is an important problem with a wide range of applications – such as numerical analysis, signal and image processing, and robotics, etc. A standard approach of studying such a problem is by using interpolating and approximating spline functions (e.g. [1–3]). An advantage of using such spline functions may be the computational feasibility. In particular, using B-splines as the basis function yields simple algorithms for designing curves [3,4] .

Conventional methods of designing splines, however, are often insufficient because there are a large class of problems where we need to impose various constraints on the curves. For example, when we consider the trajectory planning problem for robots, we may have difficulties to plan the high performance trajectory without imposing constraints – such as velocity, acceleration, and jerk saturations, etc (see e.g.[5]). In particular, when via-points for the planned trajectory are sparsely distributed in some interval of interest, the difficulties may increase further. Thus, the so-called constrained splines have been studied. Martin and his group in [6,7] have developed the method of constructing smoothing spline curves with inequality constraints. By employing the control theoretic approach, they have presented that the problem of constructing such constrained smoothing splines reduces to a quadratic programming problem. Similar results are exhibited in the work by Kolter and Zg in [8], where they have used polynomial cubic splines. However, these constraints on splines can be imposed only at the isolated points.

On the other hand, Egerstedt and Martin in [9] have focused on the problem of designing the monotone smoothing splines, in which the non-negativity constraint on the derivatives of splines is imposed on some interval. In general, such a problem with constraints over interval leads to an infinite dimensional problem and not be easily solved. Therein, the constrained spline problem is formulated and solved as a dynamic programming problem. Similar results are shown in the works by Meyer in [10] and Elfving and Anderson in [11]. These methods are however specific to the cubic splines, not the splines with arbitrary degrees. The authors in [12,13] have developed a method for designing smoothing spline curves with various types of constraint – such as constraints over interval or at isolated points, constraints on derivative and integral values of splines. Then, it has been shown that B-spline approach yielded systematic treatments and solutions for the constrained spline problems using the splines with arbitrary degrees. Moreover, the recursive design algorithm of such constrained splines has recently been developed [14]. But, these methods in [12–14] have been applied only when the given data is scalar-valued, and thus we can not impose the constraints cross-coupled among the multiple curves.

In this paper, we generalize the design method of constrained smoothing splines [12] to the case of vector spline curves with equality and/or inequality constraints. The vector splines are constructed by employing normalized uniform B-splines as the basis functions. The constraints include those at isolated time instant or over some time interval, which

* Manuscript Received Date: May 16, 2012

† Department of System Management, Fukuoka Institute of Technology; 3-30-1, Wajiro-higashi, Higashi-ku, Fukuoka 811-0295, JAPAN

‡ Division of Science, School of Science and Engineering, Tokyo Denki University; Hatoyama, Hiki-gun, Saitama 350-0394, JAPAN

Key Words: B-spline, vector smoothing spline, coupled constraint, quadratic programming, trajectory planning.

usually is not easy to treat, and those involving couplings among the element curves. We will see that, in the present approach, all such constraints can be systematically incorporated to the problem of optimal splines, and the construction becomes convex quadratic programming problem, for which efficient numerical algorithms are available. Thus it provides a powerful tool for many practical applications – such as trajectory and motion planning problems in robotics (see e.g. [8]).

This paper is organized as follows. In Section **2.**, we briefly review vector B-spline curves and design method of optimal vector splines. Then in Section **3.**, we show how various types of constraints on splines can be formulated and solved by employing B-spline approach. We examine the performances of the proposed method by numerical examples in Section **4.**. Concluding remarks are given in Section **5.**.

We use the following symbols throughout the paper: $\otimes$ denotes the Kronecker product, and 'vec' the vec-function, i.e. for a matrix $A=[a_1\ a_2\cdots\ a_n]\in\mathbf{R}^{m\times n}$ with $a_i\in\mathbf{R}^m$, vec $A=[a_1^{\mathrm{T}}\ a_2^{\mathrm{T}}\ \cdots\ a_n^{\mathrm{T}}]^{\mathrm{T}}\in\mathbf{R}^{mn}$ (see e.g.[15]).

2. Optimal Vector Splines

As preliminaries, we present vector B-spline curves and the optimal design method of smoothing splines without constraints.

2.1 Vector B-Spline Curves

We construct vector spline curve $x(t)\in\mathbf{R}^p$ $(p\geq 1)$ of degree k,

$$x(t)=\left[\, x_1(t)\ x_2(t)\ \cdots\ x_p(t)\,\right]^{\mathrm{T}}, \tag{1}$$

using B-spline function $B_k(t)$ as the basis function by

$$x(t)=\sum_{i=-k}^{m-1}\tau_i B_k(\alpha(t-t_i)). \tag{2}$$

Here $\tau_i\in\mathbf{R}^p$ are the weighting coefficient vectors, called 'control points' [3], with

$$\tau_i=\left[\,\tau_{1,i}\ \tau_{2,i}\ \cdots\ \tau_{p,i}\,\right]^{\mathrm{T}}, \tag{3}$$

m is an integer, and $\alpha(>0)$ is a constant for scaling the interval between equally-spaced knot points t_i with

$$t_{i+1}-t_i=\frac{1}{\alpha}. \tag{4}$$

Note that, by an appropriate choice of τ_i, we can design arbitrary spline curves $x(t)$ of degree k on the interval $[t_0,t_m](\subset\mathbf{R})$ for t.

In (2), $B_k(t)$, $t\in(-\infty,+\infty)$ is the normalized, uniform B-spline function of degree k defined by

$$B_k(t)=\begin{cases} N_{k-j,k}(t-j), & j\leq t<j+1,\\ & j=0,1,\cdots,k\\ 0, & t<0\ \text{ or }\ t\geq k+1\end{cases} \tag{5}$$

and the basis elements $N_{j,k}(t)$ $(j=0,1,\cdots,k)$, $0\leq t\leq 1$ are obtained recursively by the following algorithm:

【Algorithm 1】 Let $N_{0,0}(t)\equiv 1$ and, for $i=1,2,\cdots,k$, compute

$$\begin{cases} N_{0,i}(t) = \dfrac{1-t}{i}N_{0,i-1}(t)\\ N_{j,i}(t) = \dfrac{i-j+t}{i}N_{j-1,i-1}(t)\\ \qquad\qquad +\dfrac{1+j-t}{i}N_{j,i-1}(t),\\ \qquad\qquad\qquad j=1,\cdots,i-1\\ N_{i,i}(t) = \dfrac{t}{i}N_{i-1,i-1}(t).\end{cases} \tag{6}$$

Thus, $B_k(t)$ is a piecewise polynomial of degree k with integer knot points and is $k-1$ times continuously differentiable. It is noted that $B_k(t)$ for $k=0,1,2,\cdots$ is normalized in the following sense,

$$\sum_{j=0}^{k}N_{j,k}(t)=1,\ 0\leq t\leq 1. \tag{7}$$

For the sake of later reference, we introduce $(k+1)$-dimensional vectors $N_k(t)$ and $h_k(t)$ as

$$N_k(t)=\left[\,N_{0,k}(t)\ N_{1,k}(t)\ \cdots\ N_{k,k}(t)\,\right]^{\mathrm{T}} \tag{8}$$

$$h_k(t)=\left[\,t^k\ t^{k-1}\ \cdots\ 1\,\right]^{\mathrm{T}}. \tag{9}$$

Then $N_k(t)$ is written as

$$N_k(t)=S_k h_k(t), \tag{10}$$

where $S_k\in\mathbf{R}^{(k+1)\times(k+1)}$ is a matrix whose i-th row consists of the coefficients of polynomial $N_{i-1,k}(t)$. When $k=3$, for example, we obtain the matrix S_3 as

$$S_3=\frac{1}{3!}\begin{bmatrix}-1 & 3 & -3 & 1\\ 3 & -6 & 0 & 4\\ -3 & 3 & 3 & 1\\ 1 & 0 & 0 & 0\end{bmatrix}. \tag{11}$$

Moreover, the l-th derivative of $N_k(t)$ is obtained as

$$N_k^{(l)}(t)=S_k C_{k,l} h_{k-l}(t) \tag{12}$$

[13], where $C_{k,l}\in\mathbf{R}^{(k+1)\times(k-(l-1))}$ is defined by

$$C_{k,l}=C_k C_{k-1}\cdots C_{k-(l-1)}, \tag{13}$$

and $C_i\in\mathbf{R}^{(i+1)\times i}$ by

$$C_i=\begin{bmatrix} i & & & \\ & i-1 & & \\ & & \ddots & \\ & & & 1\\ 0 & \cdots & \cdots & 0\end{bmatrix}. \tag{14}$$

Here the empty spaces denote zero entries.

2.2 Optimal Vector Smoothing Splines

Due to the properties of the basis functions $B_k(t)$, the function $x(t)$ in (2) describes a vectorized piecewise polynomials of degree k with the knot points t_i and is $k-1$ times continuously differentiable. In particular, depending on the choice of the control point vector $\tau_i \in \mathbf{R}^p$, a class of polynomial splines $x(t)$ of degree k can be generated on the interval $[t_0, t_m]$ for t. For such a class of splines, we consider a problem of optimally constructing the smoothing splines $x(t)$, which are not only close to the given data but also sufficiently smooth. In the sequel, we briefly review how the problem is formulated as an optimization problem on τ_i for typical cases of discrete and continuous data.

Suppose that we are given a set of data

$$\{(s_i, d_i) \,:\, s_i \in [t_0, t_m],\ d_i \in \mathbf{R}^p,\ i = 1,2,\cdots,N\}, \tag{15}$$

and let $\tau \in \mathbf{R}^{p\times M}$ $(M = m+k)$ be the control point matrix defined from (3) by

$$\tau = [\, \tau_{-k}\ \tau_{-k+1}\ \cdots\ \tau_{m-1} \,]. \tag{16}$$

Then a standard problem is to find such a τ minimizing the cost function

$$J(\tau) = \int_{t_0}^{t_m} \left|\left| x^{(2)}(t) \right|\right|_{\Lambda}^2 dt + \sum_{i=1}^{N} ||x(s_i) - d_i||_{W_i}^2, \tag{17}$$

where $||z||_S^2 = z^{\mathrm{T}} S z$, $\Lambda = \mathrm{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_p\} \in \mathbf{R}^{p\times p}$ with smoothing parameter $\lambda_i (>0)$, $\forall i$, and the weighting matrix $W_i = W_i^{\mathrm{T}} \in \mathbf{R}^{p\times p}$ satisfies $0 \le W_i \le I_p$, $\forall i$. Note that the first term in (17) governs the smoothness of $x(t)$, while the second term describes the approximation error from the given data.

Letting $\hat{\tau} \in \mathbf{R}^{pM}$ be the vec-function of $\tau \in \mathbf{R}^{p\times M}$, i.e.

$$\hat{\tau} = \mathrm{vec}\ \tau, \tag{18}$$

the cost function $J(\tau)$ can be rewritten as a quadratic function $J(\hat{\tau})$ in terms of $\hat{\tau}$,

$$J(\hat{\tau}) = \hat{\tau}^{\mathrm{T}} G_N \hat{\tau} - 2\hat{\tau}^{\mathrm{T}} g_N + \mathrm{const.} \tag{19}$$

(see Appendix 1. for derivation), where const. is the term independent of $\hat{\tau}$, and

$$G_N = Q \otimes \Lambda + \sum_{i=1}^{N} \left(b(s_i) b^{\mathrm{T}}(s_i) \right) \otimes W_i \tag{20}$$

$$g_N = \sum_{i=1}^{N} b(s_i) \otimes W_i d_i. \tag{21}$$

Here, $b(t) \in \mathbf{R}^M$ is defined by

$$b(t) = \big[\, B_k(\alpha(t - t_{-k}))\ B_k(\alpha(t - t_{-k+1})) \\ \cdots\ B_k(\alpha(t - t_{m-1})) \,\big]^{\mathrm{T}}, \tag{22}$$

and $Q \in \mathbf{R}^{M\times M}$ is a Gramian defined by

$$Q = \int_{t_0}^{t_m} \frac{d^2 b(t)}{dt^2} \frac{d^2 b^{\mathrm{T}}(t)}{dt^2} dt. \tag{23}$$

It can be shown that, given k, α and m in (2), the Gramian Q is computed explicitly by using B-splines in (5) (see, e.g. [4] for details).

Note that $G_N \ge 0$ (positive-semidefinite) in (20), since $\Lambda > 0$, $Q \ge 0$, $b(s_i) b^{\mathrm{T}}(s_i) \ge 0$ and $W_i \ge 0$. Hence $J(\hat{\tau})$ in (19) is convex in $\hat{\tau}$.

On the other hand, when the data is given as a function $f(t) \in \mathbf{R}^p$, $t \in [t_0, t_m]$ instead of (15), we employ the following cost function

$$J(\tau) = \int_{t_0}^{t_m} \left|\left| x^{(2)}(t) \right|\right|_{\Lambda}^2 dt + \int_{t_0}^{t_m} ||x(t) - f(t)||^2 dt. \tag{24}$$

This cost $J(\tau)$ can also be rewritten as a convex quadratic function in terms of $\hat{\tau}$ similarly as (19).

3. Splines with Coupled Constraints

There are various types of constraints on splines $x(t)$, $t \in [t_0, t_m]$, e.g. those for $x(t)$ and/or its derivatives, for some isolated point t or for some interval of t, for equality and/or inequality, etc. Here we first develop basic formula for expressing the constraints, which are then used to treat cross-coupled constraints among the element curves.

3.1 Basic Formula

Noting that every element of $x(t)$ in (1) is a piecewise polynomial, we examine $x(t)$ in each interval $[t_j, t_{j+1})$ for $j = 0,1,\cdots,m-1$.

By (2) and (5), the spline $x(t)$ on the interval $[t_j, t_{j+1})$ is written as

$$x(t) = \sum_{i=-k+j}^{j} \tau_i B_k(\alpha(t - t_i)). \tag{25}$$

Using (5), we then get

$$x(t) = \sum_{i=0}^{k} \tau_{j-k+i} N_{i,k}(\alpha(t - t_j)), \quad t \in [t_j, t_{j+1}), \tag{26}$$

and it depends on only the $k+1$ weight vectors τ_{j-k}, τ_{j-k+1}, $\cdots$, τ_j. Moreover, by introducing a new variable u,

$$u = \alpha(t - t_j), \tag{27}$$

the interval $[t_j, t_{j+1})$ in t is normalized to $[0,1)$ in u, and we write $x(t)$ in (26) as $\hat{x}(u)$,

$$\hat{x}(u) = \sum_{i=0}^{k} \tau_{j-k+i} N_{i,k}(u), \quad u \in [0,1). \tag{28}$$

Letting $\tau_{(j)} \in \mathbf{R}^{p\times(k+1)}$ be a sub-matrix of τ in (16) defined by

$$\tau_{(j)} = [\, \tau_{j-k}\ \tau_{j-k+1}\ \cdots\ \tau_j \,] \tag{29}$$

and using (8), we may rewrite $\hat{x}(u)$ in (28) as $\hat{x}(u) = \tau_{(j)} N_k(u)$. Hence, we have

$$x(t) = (N_k(u) \otimes I_p)^{\mathrm{T}} \hat{\tau}_{(j)} \tag{30}$$

with $\hat{\tau}_{(j)} = \mathrm{vec}\ \tau_{(j)}$, where we used the formula vec $(AXB) = (B^{\mathrm{T}} \otimes A)$vec X and $(A \otimes B)^{\mathrm{T}} = A^{\mathrm{T}} \otimes B^{\mathrm{T}}$ for Kronecker product. In general, the l-th derivative $x^{(l)}(t)$ for $t \in [t_j, t_{j+1})$ is expressed in terms of $u \in [0,1)$ in (27) by

$$x^{(l)}(t) = \alpha^l \hat{x}^{(l)}(u),\ l = 0,1,2,\cdots, \tag{31}$$

with

$$\hat{x}^{(l)}(u) = \left(N_k^{(l)}(u) \otimes I_p\right)^{\mathrm{T}} \hat{\tau}_{(j)}. \tag{32}$$

Now we are in a position to derive expressions for various types of constraints on $x(t)$.

3.2 Pointwise Constraints

From (31) and (32), we see that any linear constraint on the value of $x^{(l)}(t)$, for given $t \in [t_j, t_{j+1})$ is specified as a linear constraint on the vector $\hat{\tau} \in \mathbf{R}^{pM}$, since $\hat{\tau}_{(j)} \in \mathbf{R}^{p(k+1)}$ is a sub-vector of $\hat{\tau}$. Specifically, we consider a constraint of the following form

$$Hx^{(l)}(t) \geq c \tag{33}$$

for given constant matrix $H \in \mathbf{R}^{q \times p}$ and vector $c \in \mathbf{R}^q$.

In this case, $x^{(l)}(t)$ is written as

$$x^{(l)}(t) = (a_l \otimes I_p)^{\mathrm{T}} \hat{\tau}, \tag{34}$$

where

$$a_l^{\mathrm{T}} = \left[\ 0_j^{\mathrm{T}}\ \ \alpha^l N_k^{(l)}(u)^{\mathrm{T}}\ \ 0_{M-j-(k+1)}^{\mathrm{T}}\ \right]. \tag{35}$$

Thus, (33) is realized as the constraint on $\hat{\tau}$ by

$$H(a_l \otimes I_p)^{\mathrm{T}} \hat{\tau} \geq c. \tag{36}$$

From this arguments, we see that the constraint $'\geq'$ in (33) and (36) may readily be replaced by $'\leq'$ and by equality $'='$.

Only the point in $[t_0, t_m]$ that is not covered in the foregoing arguments is $t = t_m$. However, the values of $x^{(l)}(t)$ at $t = t_m$ for $l = 0,1,\cdots,k-1$ are obtained, by the continuity of these functions. Namely, by letting $j = m-1$ and $u = 1$, we have

$$x^{(l)}(t_m) = (a_l \otimes I_p)^{\mathrm{T}} \hat{\tau} \tag{37}$$

with

$$a_l^{\mathrm{T}} = \left[\ 0_{M-(k+1)}^{\mathrm{T}}\ \ \alpha^l N_k^{(l)}(1)^{\mathrm{T}}\ \right]. \tag{38}$$

If we need to constrain $x^{(k)}(t)$ at $t = t_m$, which is piecewise constant and is discontinuous at the knot points, we simply regard that $x^{(k)}(t_m) = \lim_{t \to t_m} x^{(k)}(t)$, implying that (38) holds also for $l = k$. Thus we have the expression of $x^{(l)}(t)$ in terms of $\hat{\tau}$ for any $l = 0,1,\cdots,k$, and $t \in [t_0, t_m]$.

As an example of pointwise constraints, consider to constrain initial conditions between the two consecutive elements of $x(t)$ as

$$x_q^{(l)}(t_0) = x_{q+1}^{(l)}(t_0) \tag{39}$$

for an l $(l = 0,1,\cdots,k)$. Then, we see that this constraints are realized by setting $H = [0_{q-1}^{\mathrm{T}}\ 1\ -1\ 0_{p-q-1}^{\mathrm{T}}]$ and $j = 0$ and $u = 0$ in (35). The conditions are expressed as

$$H(a_l \otimes I_p)^{\mathrm{T}} \hat{\tau} = 0 \tag{40}$$

with

$$a_l = [\alpha^l N_k^{(l)}(0)^{\mathrm{T}}\ 0_{M-(k+1)}^{\mathrm{T}}]^{\mathrm{T}}. \tag{41}$$

Here, $N_k^{(l)}(0)^{\mathrm{T}}$ is readily obtained from (12) with $u = 0$. For example, $N_k^{(l)}(0)^{\mathrm{T}}$ for $k = 3$, i.e. $N_3^{(l)}(0)^{\mathrm{T}}$, is obtained as

$$N_3^{(l)}(0)^{\mathrm{T}} = \begin{cases} \frac{1}{6}\left[\ 1\ 4\ 1\ 0\ \right] & l = 0 \\ \frac{1}{2}\left[\ -1\ 0\ 1\ 0\ \right] & l = 1 \\ \left[\ 1\ -2\ 1\ 0\ \right] & l = 2 \\ \left[\ -1\ 3\ -3\ 1\ \right] & l = 3 \end{cases}. \tag{42}$$

To conclude, this method can be used to specify linear equality or inequality constraint on $x^{(l)}(t)$, among the elements $x_q^{(l)}(t)$, $q = 1,2,\cdots,p$, for given $t \in [t_0, t_m]$ and $l = 0,1,\cdots,k$ as linear constraint on $\hat{\tau}$.

3.3 Constraints over Knot Point Intervals

Next we consider the cases of constraints over knot point intervals. Specifically, we consider an inequality constraint as

$$Hx(t) \geq f(t) \quad \forall t \in [t_j, t_{j+1}] \tag{43}$$

for a given matrix $H \in \mathbf{R}^{q \times p}$ and vector-valued continuous function $f(t) \in \mathbf{R}^q$,

$$f(t) = \left[\ f_1(t)\ f_2(t)\ \cdots\ f_q(t)\ \right]^{\mathrm{T}}. \tag{44}$$

Such an expression may be used to impose the constraints for each elements of $x(t)$ and coupled constraints among some elements of $x(t)$.

Now suppose that $f(t)$ is itself a spline expressed in the same form as in (2), i.e.

$$f(t) = \sum_{i=-k}^{m-1} \phi_i B_k(\alpha(t - t_i)). \tag{45}$$

Then $f(t)$ in the interval $[t_j, t_{j+1}]$ is written as

$$f(t) = \sum_{i=-k+j}^{j} \phi_i B_k(\alpha(t - t_i)) \tag{46}$$

and, similarly as (25)-(28), we get

$$f(t) = \hat{f}(u) = \sum_{i=0}^{k} \phi_{j-k+i} N_{i,k}(u),\ u \in [0,1] \tag{47}$$

with $u = \alpha(t - t_j)$.

Then, the constraint in (43) can be realized by imposing the condition

$$H\tau_i \geq \phi_i,\ i=j-k,j-k+1,\cdots,j \tag{48}$$

or

$$H\tau_{(j)} \geq \phi_{(j)}, \tag{49}$$

where $\tau_{(j)}=[\tau_{j-k}\ \tau_{j-k+1}\ \cdots\ \tau_j]$ by (29), and similarly $\phi_{(j)}=[\phi_{j-k}\ \phi_{j-k+1}\ \cdots\ \phi_j]$. This is because we then have from (26)-(28) that

$$\begin{aligned} Hx(t) &= H\hat{x}(u)=\sum_{i=0}^{k} H\tau_{j-k+i}N_{i,k}(u) \\ &\geq \sum_{i=0}^{k}\phi_{j-k+i}N_{i,k}(u)=\hat{f}(u)=f(t) \\ &\qquad \forall t\in[t_j,t_{j+1}], \end{aligned} \tag{50}$$

since $N_{i,k}(u)\geq 0\ \forall u\in[0,1]$.

We now express the condition (49) in terms of $\hat{\tau}$. Since $\tau_{(j)}$ is given as

$$\tau_{(j)}=\tau E_j \tag{51}$$

with

$$E_j^{\mathrm{T}}=\left[\, 0_{k+1,j}\ \ I_{k+1}\ \ 0_{k+1,M-j-(k+1)}\,\right], \tag{52}$$

it holds that

$$\mathrm{vec}(H\tau_{(j)})=(E_j^{\mathrm{T}}\otimes H)\hat{\tau}. \tag{53}$$

Thus the condition $H\tau_{(j)}\geq\phi_{(j)}$ is specified by

$$(E_j^{\mathrm{T}}\otimes H)\hat{\tau}\geq\hat{\phi}_{(j)}, \tag{54}$$

where

$$\hat{\phi}_{(j)}=\mathrm{vec}\ \phi_{(j)}. \tag{55}$$

A simple but useful example of $f(t)$ in (44) is a linear function in t, i.e.

$$f(t)=c_1(t-t_0)+c_2, \tag{56}$$

where $c_1,c_2\in\mathbf{R}^q$ are some constant vectors. Here we derive an expression for the control point ϕ_i in (45) yielding $f(t)$ in (56) in the interval $[t_j,t_{j+1}]$. First, the function $f(t)$ is expressed in $u=\alpha(t-t_j)$ as

$$\hat{f}(u)=\hat{c}_1u+\hat{c}_2 \tag{57}$$

with $\hat{c}_1=\dfrac{c_1}{\alpha}$ and $\hat{c}_2=j\dfrac{c_1}{\alpha}+c_2$. On the other hand, $\hat{f}(u)$ in (47) is expressed as $\hat{f}(u)=\phi_{(j)}N_k(u)$. Then we have $\phi_{(j)}N_k(u)=\hat{c}_1u+\hat{c}_2$, and (10) and (9) yield

$$\phi_{(j)}S_kh_k(u)=[0_{q,k-1}\ \hat{c}_1\ \hat{c}_2]h_k(u). \tag{58}$$

Since this holds for all $u\in[0,1]$, we get $\phi_{(j)}S_k=[0_{q,k-1}\ \hat{c}_1\ \hat{c}_2]$, and hence the control points yielding $f(t)$ in (44) in the interval $[t_j,t_{j+1}]$ are obtained by

$$\phi_{(j)}=[0_{q,k-1}\ \hat{c}_1\ \hat{c}_2]S_k^{-1}. \tag{59}$$

In the case of cubic splines, using S_3 in (11) yields

$$\begin{aligned} \phi_{(j)} &= \left[\, 0_{q,2}\ \ \hat{c}_1\ \ \hat{c}_2\,\right]S_3^{-1} \\ &= \left[\, -\hat{c}_1+\hat{c}_2\ \ \hat{c}_2\ \ \hat{c}_1+\hat{c}_2\ \ 2\hat{c}_1+\hat{c}_2\,\right], \end{aligned} \tag{60}$$

and thus

$$\phi_i=\frac{i+2}{\alpha}c_1+c_2,\ i=j-3,j-2,j-1,j. \tag{61}$$

Here we considered constraints over single knot point interval, namely $[t_j,t_{j+1}]$, but we see that the above arguments can be easily extended to larger knot point interval $[t_j,t_l]$ for any l $(>j)$ and to the entire interval $[t_0,t_m]$.

3.4 Constraints on Integral Values

We finally consider the case of an inequality constraint on the value of integral as

$$H\int_{t_0}^{t_m}x(t)dt\geq c \tag{62}$$

for given constant matrix $H\in\mathbf{R}^{q\times p}$ and vector $c\in\mathbf{R}^q$.

From (31), (32) with $l=0$ and (27), we get

$$\begin{aligned} &\int_{t_0}^{t_m}x(t)dt \\ &=\sum_{j=0}^{m-1}\int_{t_j}^{t_{j+1}}x(t)dt \\ &=\frac{1}{\alpha}\sum_{j=0}^{m-1}\int_0^1\hat{x}(u)du \\ &=\frac{1}{\alpha}\sum_{j=0}^{m-1}\left(\int_0^1 N_k(u)du\otimes I_p\right)^{\mathrm{T}}\hat{\tau}_{(j)}. \end{aligned} \tag{63}$$

Now we get $\hat{\tau}_{(j)}=(E_j^{\mathrm{T}}\otimes I_p)\hat{\tau}$ by (51), and the integral is expressed as

$$\int_{t_0}^{t_m}x(t)dt=A\hat{\tau} \tag{64}$$

where $A\in\mathbf{R}^{p\times pM}$ is given by

$$\begin{aligned} A &= \frac{1}{\alpha}\sum_{j=0}^{m-1}\left(\int_0^1 N_k^{\mathrm{T}}(u)du\otimes I_p\right)\left(E_j^{\mathrm{T}}\otimes I_p\right) \\ &= \frac{1}{\alpha}\left(\int_0^1 N_k^{\mathrm{T}}(u)du\sum_{j=0}^{m-1}E_j^{\mathrm{T}}\right)\otimes I_p \\ &= \frac{1}{\alpha}(a^{\mathrm{T}}\otimes I_p) \end{aligned} \tag{65}$$

with

$$a^{\mathrm{T}}=\int_0^1 N_k^{\mathrm{T}}(u)du\sum_{j=0}^{m-1}E_j^{\mathrm{T}}. \tag{66}$$

In (65), we used a relation $(A_1\otimes B_1)(A_2\otimes B_2)=(A_1A_2)\otimes(B_1B_2)$ for matrices with proper dimensions.

For the case of $k=3$, using (10) and (9), we get

$$\int_0^1 N_3^{\mathrm{T}}(u)du = \frac{1}{24}\left[\,1\ 11\ 11\ 1\,\right], \tag{67}$$

and by (52) we get

$$\sum_{j=0}^{m-1} E_j^{\mathrm{T}} = \begin{bmatrix} 1\ 1\ 1\ 1 \cdots 1\ 0\ 0\ 0 \\ 0\ 1\ 1\ 1 \cdots 1\ 1\ 0\ 0 \\ 0\ 0\ 1\ 1 \cdots 1\ 1\ 1\ 0 \\ 0\ 0\ 0\ 1 \cdots 1\ 1\ 1\ 1 \end{bmatrix}. \tag{68}$$

Then the vector $a \in \mathbf{R}^M$ in (66) is obtained as

$$a = \frac{1}{24}\left[\,1\ 12\ 23\ 24 \cdots 24\ 23\ 12\ 1\,\right]^{\mathrm{T}}. \tag{69}$$

In conclusion, the constraint (62) is realized by the following condition on $\hat{\tau}$,

$$HA\hat{\tau} \geq c. \tag{70}$$

3.5 Constrained Vector Splines

In Sections **3.2**–**3.4**, we showed that our B-spline approach enables us to express various types of constraints as linear function of control points $\hat{\tau}(=\mathrm{vec}\,\tau)$, including those on vector $x(t)$ and its derivatives and integrals both independently on its elements and as cross-coupled constraints among the elements. Since the optimal vector smoothing splines are obtained by minimizing the convex quadratic cost $J(\hat{\tau})$ as in (19), we then see that a general form of this constrained problem becomes a convex quadratic programming problem in $\hat{\tau}$ as follows:

$$\min_{\hat{\tau}\in\mathbf{R}^{pM}} J(\hat{\tau}) = \frac{1}{2}\hat{\tau}^{\mathrm{T}} G\hat{\tau} + g^{\mathrm{T}}\hat{\tau} \tag{71}$$

subject to the constraints of the form

$$A\hat{\tau} = d,\ f_1 \leq E\hat{\tau} \leq f_2, \tag{72}$$

for some matrices and vectors of appropriate dimensions. Although there is a variety of methods for solving this problem (see e.g. [16]), we here use the function “quadprog” in MATLAB Optimization Toolbox.

4. Numerical Example

One of major applications of the proposed method arises in the motion planning problems of the robotics field. For example, let us consider a case where an aero-robot with some range sensor is in 3-dimensional environment, where we assume the environmental map has already been built. When a set of data is measured by the range sensor and we need to plan the collision-free trajectory based on the environmental map, then the vector smoothing splines with constraints can be used effectively.

With the above case in mind, we here examine the performances of the design method by the following numerical example for $p=3$. Also, cubic splines, i.e. $k=3$, are used.

Specifically, we design smoothing splines $x(t) \in \mathbf{R}^3$,

$$x(t) = [x_1(t)\ x_2(t)\ x_3(t)]^{\mathrm{T}}, \tag{73}$$

in the time interval $[t_0, t_m] = [0, 36]$ such that

$$0 \leq x_1(t) \leq 6,\ 0 \leq x_2(t) \leq 5,\ 0 \leq x_3(t) \leq 0.5 \tag{74}$$

and satisfy the coupled inequality constraints

$$\begin{cases} \frac{1}{3}x_1(t) \leq x_2(t) \leq \frac{1}{4}x_1(t)+1, \\ \qquad\qquad\qquad\qquad \text{if}\quad t\in[0,18] \\ -\frac{3}{2}x_1(t)+8 \leq x_2(t) \leq -\frac{3}{2}x_1+11, \\ \qquad\qquad\qquad\qquad \text{if}\quad t\in(18,36]. \end{cases} \tag{75}$$

We set $\alpha=1$ and $m=36$ in (2), hence the knot points are $t_i = i,\ i=-3,-2,\cdots,\ m-1$. The initial,via point and terminal conditions are set as

$$\begin{aligned} &x(0) = [0\ 0\ 0.5]^{\mathrm{T}},\ x^{(1)}(0) = x^{(2)}(0) = 0_3, \\ &x(18) = [4\ 5\ 0]^{\mathrm{T}}, \\ &x(36) = [5\ 2\ 0.5]^{\mathrm{T}},\ x^{(1)}(36) = x^{(2)}(36) = 0_3. \end{aligned} \tag{76}$$

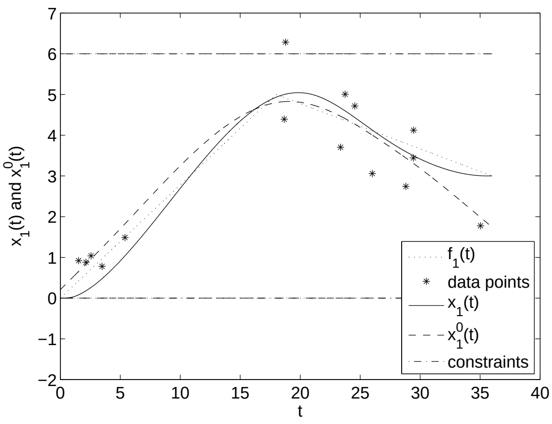
Note that the inequality constraints in (74)-(75) are those over knot point intervals and are imposed by employing the method in Section **3.3**. Also, the constraints in (76) are the pointwise constraints and the method in Section **3.2** can be used.

The coupled inequality constraints in (75) constitute a geometrical constraint on the 2-dimensional plane $o - x_1x_2$. For designing optimal smoothing splines $x(t)$ under such constraints, a natural choice of the data points $d_i \in \mathbf{R}^3$ in (15) will be the mid points from the constraint borders specified by (75). Moreover, it is natural to assume that the data is corrupted by some observation noise. Here, we generate the data d_i by sampling the function $f(t) \in \mathbf{R}^3$ corresponding to the center line and then add some Gaussian noise to d_i. Specifically, $f(t) = [f_1(t)\ f_2(t)\ f_3(t)]^{\mathrm{T}}$ is defined by

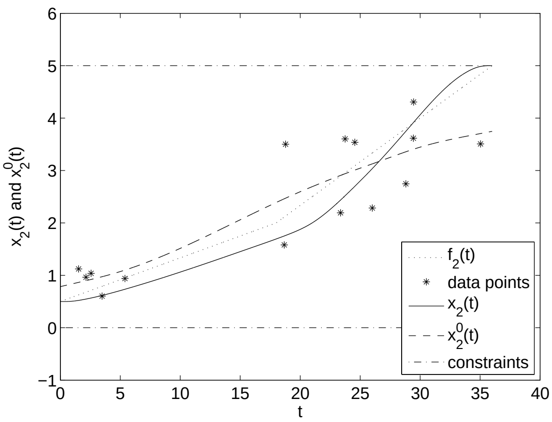
$$f(t) = \begin{cases} \left[\frac{5}{18}t\ \ \frac{1}{12}t+\frac{1}{2}\ \ \frac{1}{36}t\right]^{\mathrm{T}}, \\ \qquad\qquad\qquad\qquad \text{if}\quad t\in[0,18] \\ \left[-\frac{1}{9}t+7\ \ \frac{1}{6}t-1\ \ -\frac{1}{36}(t-18)+0.5\right]^{\mathrm{T}}, \\ \qquad\qquad\qquad\qquad \text{if}\quad t\in(18,36]. \end{cases} \tag{77}$$

The magnitude of the additive Gaussian noise in d_i is set as $\sigma = 0.55$, the number of data is set as $N=15$, and s_i's are randomly spaced in the interval $[t_0, t_m] = [0, 36]$. The smoothing parameters Λ and W_i are set as $\Lambda = 10^2 \cdot I_3$ and $W_i = I_3$ respectively. The optimal weight τ is obtained by the method in Section **3.5** for the constraints in (74)-(76), and then we compute $x(t) = [x_1(t)\ x_2(t)\ x_3(t)]^{\mathrm{T}}$ by (2).

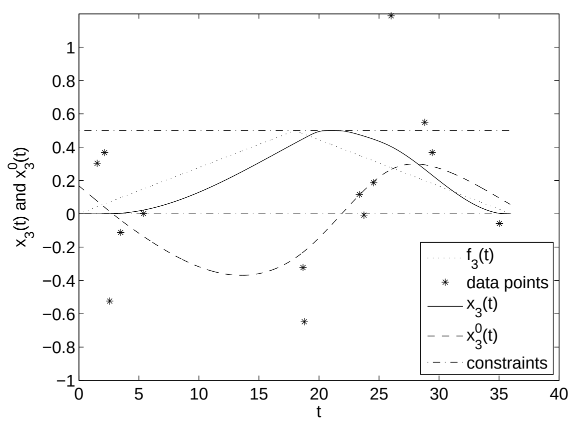
Fig. 1 shows the results for the elements of constrained vector spline $x(t)$ in solid lines, where the data points (s_i, d_i) are shown by asterisk $*$. For the sake of comparison, we plotted in the same figures the



(a) $x_1(t)$



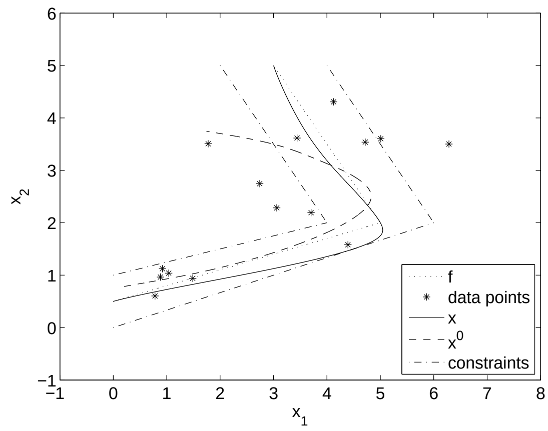
(b) $x_2(t)$



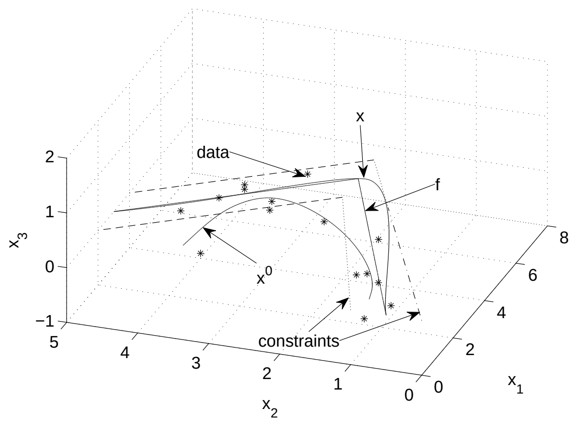
(b) $x_3(t)$

Fig. 1 Design example of constrained and unconstrained splines (resp. $x(t)=[x_1(t)\ \ x_2(t)\ \ x_3(t)]^{\mathrm{T}}$ and $x^0(t)=[x_1^0(t)\ x_2^0(t)\ x_3^0(t)]^{\mathrm{T}}$)

results for the unconstrained vector spline, denoted by $x_i^0(t)$, $i=1,2,3$, which were obtained without imposing any constraints in (74)-(76). The function $f(t)$ in (77) is plotted in dotted lines. Also, in Fig. 2, the planned motions $x(t)$ and $x^0(t)$ are plotted on 2-dimensional plane $o-x_1x_2$ and in 3-dimensional space $o-x_1x_2x_3$. From these results, we see that the pro-



(a) $x(t)$ and $x^0(t)$ on $o-x_1x_2$ plane



(b) $x(t)$ and $x^0(t)$ in $o-x_1x_2x_3$ space

Fig. 2 $x(t)$ and $x^0(t)$ on 2-dimensional plane $o-x_1x_2$ and in 3-dimensional space $o-x_1x_2x_3$.

posed method works well and the constructed spline $x(t)$ satisfies all the constraints in (74)-(76) unlike the case of $x^0(t)$ with no constraints.

5. Concluding Remarks

In this paper, we developed a method for designing optimal vector smoothing splines with equality and/or inequality constraints including cross-coupled constraints. The splines are constituted employing normalized uniform B-splines as the basis functions, and hence the central issue is to determine an optimal matrix τ of the so-called control points. Such an approach enables us to express various types of constraints as linear function of $\hat{\tau}(=\mathrm{vec}\ \tau)$, including those on the spline $x(t)$ and its elements, their derivatives and coupled constraints among the elements. The design problem becomes a convex quadratic programming problem in $\hat{\tau}$, where very efficient numerical algorithms are available. We numerically examined the performances of the design method by an example with equality and inequality constraints. The developed method is effective as well as very useful for various types of problems including the robotics applications.

Acknowledgments

This work was supported in part by Japan Ministry of Education, Culture, Sports, Science and Technology under Grant No.23560541.

Appendix

Appendix 1. Derivation of Expression of $J(\hat{\tau})$

We here show how the expression of cost $J(\hat{\tau})$ in (19) is derived from $J(\tau)$ in (17).

First, we evaluate the integral term in (17). Noting that $x(t)$ in (2) is expressed as $x(t)=\tau b(t)$ with matrix τ in (16) and vector $b(t)$ in (22), we have

$$\begin{aligned}&\int_{t_0}^{t_m}\left|\left|x^{(2)}(t)\right|\right|_{\Lambda}^{2}dt\\ &=\int_{t_0}^{t_m}\left(x^{(2)}(t)\right)^{\mathrm{T}}\Lambda x^{(2)}(t)dt\\ &=\int_{t_0}^{t_m}\left(\tau\frac{d^2b(t)}{dt^2}\right)^{\mathrm{T}}\Lambda\tau\frac{d^2b(t)}{dt^2}dt\\ &=\mathrm{tr.}\left(\tau^{\mathrm{T}}\Lambda\tau Q\right)\\ &=\hat{\tau}^{\mathrm{T}}(Q\otimes\Lambda)\hat{\tau}, \qquad\qquad \text{(A1)}\end{aligned}$$

where $\hat{\tau}\in\mathbf{R}^{pM}$ and $Q\in\mathbf{R}^{M\times M}$ denote vec-function of τ defined by (18) and Gramian defined by (23), respectively. We here used the formula $\mathrm{tr.}\left(A^{\mathrm{T}}BCD^{\mathrm{T}}\right)=(\mathrm{vec}\ A)^{\mathrm{T}}(D\otimes B)\mathrm{vec}\ C$ for matrices with proper dimensions.

Regarding the second term in (17), we similarly have

$$\begin{aligned}&\sum_{i=1}^{N}||x(s_i)-d_i||_{W_i}^2\\ &=\sum_{i=1}^{N}(x(s_i)-d_i)^{\mathrm{T}}W_i(x(s_i)-d_i)\\ &=\sum_{i=1}^{N}(\tau b(s_i)-d_i)^{\mathrm{T}}W_i(\tau b(s_i)-d_i)\\ &=\sum_{i=1}^{N}\mathrm{tr.}\left(\tau b(s_i)b^{\mathrm{T}}(s_i)\tau^{\mathrm{T}}W_i\right.\\ &\qquad\qquad \left.-2\tau b(s_i)d_i^{\mathrm{T}}W_i+d_id_i^{\mathrm{T}}W_i\right)\\ &=\sum_{i=1}^{N}\left[\hat{\tau}^{\mathrm{T}}\left(b(s_i)b^{\mathrm{T}}(s_i)\otimes W_i\right)\hat{\tau}\right.\\ &\qquad\qquad \left.-2\hat{\tau}^{\mathrm{T}}(b(s_i)\otimes W_id_i)+d_i^{\mathrm{T}}W_id_i\right]. \qquad \text{(A2)}\end{aligned}$$

Hence, letting $J(\hat{\tau})$ be the cost function $J(\tau)$ in (17) expressed in terms of $\hat{\tau}$, we get

$$\begin{aligned}J(\hat{\tau})=&\hat{\tau}^{\mathrm{T}}(Q\otimes\Lambda)\hat{\tau}+\hat{\tau}^{\mathrm{T}}\sum_{i=1}^{N}\left(b(s_i)b^{\mathrm{T}}(s_i)\otimes W_i\right)\hat{\tau}\\ &-2\hat{\tau}^{\mathrm{T}}\sum_{i=1}^{N}(b(s_i)\otimes W_id_i)+\sum_{i=1}^{N}d_i^{\mathrm{T}}W_id_i \qquad \text{(A3)}\end{aligned}$$

and (19) follows.

References

[1] G. Wahba: *Spline Models for Observational Data*, CBMS-NSF Regional Conference Series in Applied Mathematics, 59, Society for Industrial and Applied Mathematics (SIAM) (1990)

[2] Z. Zhang, J. Tomlinson and C. F. Martin: Splines and linear control theory; *Acta Appl. Math.*, Vol. 49, No. 1, pp. 1–34 (1997)

[3] C. de Boor: *A Practical Guide to Splines*, Revised Edition, Springer-Verlag (2001)

[4] 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, No. 1, pp. 96–128 (2005)

[5] L. Biagiotti and C. Melchiorri: *Trajectory Planning for Automatic Machines and Robots*, Springer-Verlag (2008)

[6] C. F. Martin, S. Sun and M. Egerstedt: Optimal control, statistics and path planning. Computation and control; *Math. Comput. Modelling*, Vol. 33, No. 1-3, pp. 237–253 (2001)

[7] M. Egerstedt and C. F. Martin: Optimal trajectory planning and smoothing splines; *Automatica*, Vol. 37, No. 7, pp. 1057–1064 (2001)

[8] J. Z. Kolter and A. Y. Ng: Task-space trajectories via cubic spline optimization; *Proc. of the 2009 Int. Conf. on Robotics and Automation*, pp. 1675–1682 (2009)

[9] M. Egerstedt and C. F. Martin: Optimal control and monotone smoothing splines; *New Trends in Nonlinear Dynamics and Control*, Lecture Notes in Control and Inform. Sci., 295, pp. 279–294, Springer (2003)

[10] M. C. Meyer: Inference using shape-restricted regression splines; *Annals of Applied Statistics*, Vol. 2, No. 3, pp. 1013–1033 (2008)

[11] T. Elfving and L. E. Anderson: An algorithm for computing constrained smoothing spline functions; *Numerische Mathematik*, Vol. 52, pp. 583–595 (1988)

[12] H. Kano, H. Fujioka and C. F. Martin: Optimal smoothing and interpolating splines with constraints; *Applied Mathematics and Computation*, Vol. 218, Issue 5, pp. 1831–1844 (2011)

[13] H. Kano, H. Fujioka and C. F. Martin: Optimal smoothing spline with constraints on its derivatives; *Proc. of the 49th IEEE Conference on Decision and Control*, pp. 6785–6790 (2010)

[14] H. Fujioka and H. Kano: Recursive construction of otimal smoothing splines with constraints; *Proc. of the 2010 American Control Conference*, pp. 1580–1585 (2010)

[15] P. Lancaster and M. Tismenetsky: *The Theory of Matrices*, Second Edition, Academic Press (1985)

[16] J. Nocedal and S. Wright: *Numerical Optimization*, Second Edition, Springer (2006)

Authors

Hiroyuki FUJIOKA (Member)

Dr. Fujioka was born in Mie, Japan, in 1975. He received the B.E., M.E. and Dr.E. degrees from Ritsumeikan University, Kyoto, Japan, in 1997, 1999, and 2004, respectively. From 2002 to 2008, he was with the Department of Information Sciences, Tokyo Denki University, Saitama, Japan, where he was an Instructor until 2005 and afterward was promoted to an Assistant Professor. In 2008, he joined as an Assistant Professor the Department of System Management, Fukuoka Institute of Technology, Fukuoka, Japan, and is currently an Associate Professor. His research interests include theory and applications of splines, and task planning and control of robots. Dr. Fujioka is a member of IEEE, RSJ, and JSME. He is a recipient of the 2004 Sunahara Prize and Young Investigators Award from the ISCIE.

Hiroyuki KANO (Member)

Dr. Kano was born in Hyogo, Japan, in 1947. He received the B.E. and M.E. degrees in Mechanical Engineering from Kyoto Institute of Technology, Japan, in 1970 and 1972 respectively, and the Ph.D. in Electrical Engineering - Systems Science from Polytechnic Institute of New York (now Polytechnic University), USA, in 1976. From 1976 to 1992, he was with the International Institute for Advanced Study of Social Information Sciences, Fujitsu Limited. In 1992, he joined as a Professor the Department of Information Sciences, Tokyo Denki University, and is currently with the Division of Science. His research interest includes control and estimation theory, robotics, and signal processings. Dr. Kano is a member of the IEEE, SICE, RSJ and JSIAM.