Preprints of the 18th IFAC World Congress
Milano (Italy) August 28 - September 2, 2011

Recursive Construction of Optimal Smoothing Spline Surfaces with Constraints

Hiroyuki Fujioka * **Hiroyuki Kano** **

* *Department of System Management, Fukuoka Institute of Technology, Fukuoka 811-0295, JAPAN (e-mail: fujioka@fit.ac.jp).*
** *School of Science and Engineering, Tokyo Denki University, Saitama 350-0394, JAPAN (e-mail: kano@mail.dendai.ac.jp)*

Abstract: In this paper, we consider the problem of recursively constructing smoothing spline surfaces with equality and/or inequality constraints each time when a new set of data is observed. The splines are constituted by using 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 quadratic programming problem. Based on the results, we develop the recursive design method for constructing such constrained smoothing spline surfaces. The performance is examined by some numerical examples.

Keywords: Splines, Smoothing, Constraints, Recursive algorithms.

1. INTRODUCTION

Constructing curves and surfaces for a given set of discrete observational data is one of key problems in many fields of engineering and sciences – such as computer aided design, numerical analysis, image processing, robotics, data analysis, etc. In such problems, interpolating and approximating methods using spline functions have been used frequently and studied extensively (e.g. Boor [2001]).

In addition to traditional approximating or interpolating splines, there are a large class of problems where we need to impose various constraints on splines – such as monotone smoothing splines (Egerstedt [2003]), inequality constraints at isolated points (Martin [2001]), etc. By employing B-splines approach, the authors have also developed a method for designing smoothing splines with constraints over interval or at isolated points, and the construction of the splines then becomes a quadratic programming problem (Kano [2007]). Some of the results using B-spline approach have been extended to the case of surfaces (Fujioka [2009a,b]).

Most of the above design methods may face the problem that the sizes of relevant matrices and vectors become large as the number of given data increases. Furthermore, their methods may be undesirable in such a case where some sets of data are observed one after another and we would like to construct spline curves and surfaces each time when a new set is given. Such a case typically arises in some robotics applications – such as SLAM (Simultaneously Localization and Mapping), etc. Thus, the so-called recursive design methods have been studied for constructing splines (see e.g. Frezza [1995], Karasalo [2007], Piccolo [2009]), but they are only for spline curves. We have also developed similar design methods for both the cases of curves and surfaces by employing B-spline approach (Fujioka [2008, 2009c]). In particular, the design method for the case of curves has recently been extended to the case of constrained splines (Fujioka [2010]).

This paper is a continuation of our studies on the optimal design of smoothing splines employing B-spline approach. In particular, based on our studies (Fujioka [2009b,c]), we here develop a recursive design algorithm of optimal smoothing spline surfaces with equality and/or inequality constraints. The splines are constructed by employing normalized uniform B-splines as the basis functions. We then show that the equality and/or inequality constraints can be systematically added as linear functions of the so-called control points and that the construction of the spline surfaces becomes convex quadratic programming problems. Based on such formulations, a recursive design algorithm for constrained smoothing spline surfaces is developed. The algorithm enables us to use in the broad range of robotics applications. We here apply the results to the 3-dimensional contour representation using a mobile robot and the effectiveness and usefulness are examined by numerical studies.

For designing surfaces $x(s,t)$, we employ normalized, uniform B-spline function $B_k(t)$ of degree k as the basis functions,

$$x(s,t) = \sum_{i=-k}^{m_1-1} \sum_{j=-k}^{m_2-1} \tau_{i,j} B_k(\alpha(s-s_i)) B_k(\beta(t-t_j)) \quad (1)$$

on a domain $\mathcal{D} = [s_0, s_{m_1}] \times [t_0, t_{m_2}] \subset \mathbf{R}^2$. Here, $\tau_{i,j}$ are the weighting coefficients called control points, $\alpha,\ \beta(>0)$ are constants, $m_1,\ m_2(>2)$ are integers, and s_i's, t_j's are equally spaced knot points with

$$s_{i+1} - s_i = \frac{1}{\alpha}, \quad t_{j+1} - t_j = \frac{1}{\beta}. \quad (2)$$

We summarize some of the symbols that will be used throughout the paper: ∇^2 denotes the Laplacian operator, and $\otimes$ the Kronecker product. Moreover, 'vec' denotes 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 $A = [a_1^T\ a_2^T\ \cdots\ a_n^T]^T \in \mathbf{R}^{mn}$ (see e.g. (Lancaster [1985])).

2. OPTIMAL SPLINE SURFACES

As preliminaries, we briefly review B-splines and optimal smoothing spline surfaces.

2.1 Normalized Uniform B-Splines

Normalized uniform B-spline $B_k(t)$ of degree k is defined by

$$B_k(t) = \begin{cases} N_{k-j,k}(t-j) & j \le t < j+1,\ j = 0,1,\cdots,k \\ 0 & t < 0 \ \text{ or } \ t \ge k+1, \end{cases} \tag{3}$$

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

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

$$\begin{cases} N_{0,i}(t) = \dfrac{1-t}{i} N_{0,i-1}(t) \\ N_{j,i}(t) = \dfrac{i-j+t}{i} N_{j-1,i-1}(t) + \dfrac{1+j-t}{i} N_{j,i-1}(t), \\ \qquad\qquad j = 1,\cdots,i-1 \\ N_{i,i}(t) = \dfrac{t}{i} N_{i-1,i-1}(t). \end{cases} \tag{4}$$

Thus, $B_k(t)$ is a piece-wise 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 sense of $\sum_{j=0}^{k} N_{j,k}(t) = 1,\ \ 0 \le t \le 1$. Using Algorithm 1, the basis elements $N_{j,k}(t)$ can readily be computed for arbitrary degree k.

2.2 Optimal Smoothing Surfaces

The control points $\tau_{i,j}$ in (1) may be determined by the theory of smoothing splines as follows.

Suppose that we are given a set of data

$$\{(u_i, v_i; d_i) : u_i \in [s_0, s_{m_1}], v_i \in [t_0, t_{m_2}], \\ d_i \in \mathbf{R}, i = 1,2,\cdots,N\} \tag{5}$$

and let $\tau \in \mathbf{R}^{M_1\times M_2}$ be the weight matrix defined by

$$\tau = \begin{bmatrix} \tau_{-k,-k} & \tau_{-k,-k+1} & \cdots & \tau_{-k,m_2-1} \\ \tau_{-k+1,-k} & \tau_{-k+1,-k+1} & \cdots & \tau_{-k+1,m_2-1} \\ \vdots & \vdots & \cdots & \vdots \\ \tau_{m_1-1,-k} & \tau_{m_1-1,-k+1} & \cdots & \tau_{m_1-1,m_2-1} \end{bmatrix} \tag{6}$$

with $M_1 = m_1 + k$ and $M_2 = m_2 + k$. Then a standard problem is to find such a τ minimizing the cost function

$$J(\tau) = \lambda \int_{I_1}\int_{I_2} \left(\nabla^2 x(s,t)\right)^2 ds dt + \sum_{i=1}^{N} w_i (x(u_i,v_i) - d_i)^2, \tag{7}$$

where $I_1 = [s_0, s_{m_1}]$, $I_2 = [t_0, t_{m_2}]$, $\lambda (> 0)$ is a smoothing parameter, and w_i $(0 \le w_i \le 1)$ are the weights for approximation errors.

Letting $\hat\tau \in \mathbf{R}^{M_1 M_2}$ be a vec-function of τ defined as

$$\hat\tau = \text{vec } \tau, \tag{8}$$

the cost function $J(\tau)$ in (7) can be rewritten as a quadratic function in terms of $\hat\tau$ (see e.g. Fujioka [2005] for details),

$$J(\hat\tau) = \hat\tau^T G \hat\tau - 2 g^T \hat\tau + r, \tag{9}$$

with

$$G = \lambda Q + \Gamma W \Gamma^T,\ g = \Gamma W d,\ r = d^T W d. \tag{10}$$

Here, $Q \in \mathbf{R}^{M_1M_2\times M_1M_2}$ is a Gram matrix defined by

$$Q = \int_{I_1}\int_{I_2} \left(\nabla^2 (b_2(t)\otimes b_1(s))\right)\left(\nabla^2(b_2(t)\otimes b_1(s))\right)^T ds dt \tag{11}$$

with

$$b_1(s) = [B_k(\alpha(s - s_{-k}))\ B_k(\alpha(s - s_{-k+1}))\ \cdots \\ \cdots\ B_k(\alpha(s - s_{m_1-1}))]^T, \tag{12}$$

$$b_2(t) = [B_k(\beta(t - t_{-k}))\ B_k(\beta(t - t_{-k+1}))\ \cdots \\ \cdots\ B_k(\beta(t - t_{m_2-1}))]^T. \tag{13}$$

The matrix $\Gamma \in \mathbf{R}^{M_1M_2\times N}$ in (9) is defined by

$$\Gamma = [b_2(v_1)\otimes b_1(u_1)\ \cdots b_2(v_N)\otimes b_1(u_N)]. \tag{14}$$

Also, $W \in \mathbf{R}^{N\times N}$ and $d \in \mathbf{R}^N$ are given by

$$W = \text{diag}\{w_1,\ w_2,\ \cdots,\ w_N\} \\ d = [d_1,\ d_2,\ \cdots,\ d_N]^T. \tag{15}$$

It can be shown that the matrix G in (10) is positive-semidefinite (see e.g. Fujioka [2005]). Thus, the cost $J(\hat\tau)$ in (9) is a convex function in $\hat\tau$. Hence, if we design the smoothing surfaces without imposing any constraints, the optimal solution $\hat\tau$ minimizing the cost function in (7) is given as a solution of $G\hat\tau = g$.

3. OPTIMAL SPLINE SURFACES WITH CONSTRAINTS

There are various types of constraints such as pointwise constraints on $x(s,t)$ and/or its derivatives, and constraints over intervals or domains in $\mathcal{D}$, either equality or inequality. Using B-splines approach, it can be shown that such constraints are formulated as linear functions of the control points (see Fujioka [2009b] for details).

As an example, we here review only an inequality constraint over a domain,

$$x(s,t) \ge f(s,t)\ \ \forall (s,t) \in [s_\kappa, s_{\kappa+1}] \times [t_\mu, t_{\mu+1}] \tag{16}$$

for a given continuous function $f(s,t)$. Note here that the inequality '$\ge$' may readily be replaced with '$\le$' and equality '$=$' as we will see in below.

3.1 Expression of Constraints

We first present the basic formula for expressing the constraints. Noting that $x(s,t)$ is constructed as a product of two piecewise polynomials, we examine the polynomial in each knot point region $\mathcal{D}_{\kappa,\mu} = [s_\kappa, s_{\kappa+1}) \times [t_\mu, t_{\mu+1})$ for $\kappa = 0, 1, \cdots, m_1 - 1$ and $\mu = 0, 1, \cdots, m_2 - 1$. For $\mathcal{D}_{\kappa,\mu}$, the function $x(s,t)$ in (1) is written as

$$x(s,t) = \sum_{i=-k+\kappa}^{\kappa} \sum_{j=-k+\mu}^{\mu} \tau_{i,j} B_k(\alpha(s-s_i)) B_k(\beta(t-t_j)), \tag{17}$$

and, by (3), we get

$$x(s,t) = \sum_{i=0}^{k} \sum_{j=0}^{k} \tau_{\kappa-k+i,\mu-k+j} \times N_{i,k}(\alpha(s-s_\kappa)) N_{j,k}(\beta(t-t_\mu)). \tag{18}$$

Then, by introducing new variables u and v defined by

$$u = \alpha(s - s_\kappa),\ v = \beta(t - t_\mu), \tag{19}$$

the region $\mathcal{D}_{\kappa,\mu}$ is normalized to the unit region $\mathcal{E} = [0,1) \times [0,1)$ for (u,v). Now, $x(s,t)$ is expressed in terms of (u,v) as $x(s,t) = \hat{x}(u,v)$ with

$$\hat{x}(u,v) = \sum_{i=0}^{k} \sum_{j=0}^{k} \tau_{\kappa-k+i,\mu-k+j} N_{i,k}(u) N_{j,k}(v). \tag{20}$$

Using the expression in (20) and the idea of "limiting spline surface" (see e.g. Fujioka [2005]), the constraint in (16) can be imposed as follows: For the given function $f(s,t)$, we first compute the limiting spline surfaces $x^c(s,t)$ with the same form and the same degree k as in (1), i.e.

$$x^c(s,t) = \sum_{i=-k}^{m_1-1} \sum_{j=-k}^{m_2-1} \tau^c_{i,j} B_k(\alpha(s-s_i)) B_k(\beta(t-t_j)). \tag{21}$$

From our past works (e.g. see Fujioka [2005]), we have empirically confirmed that the surfaces $x^c(s,t)$ can approximate functions $f(s,t)$ fairly precisely, i.e. $x^c(s,t) \approx f(s,t)$. We thus regard the constraint $x(s,t) \geq f(s,t)$ in (16) as $x(s,t) \geq x^c(s,t)$. Then, such a constraint may be realized by imposing the condition $\tau_{i,j} \geq \tau^c_{i,j}$ for $i = \kappa-k, \kappa-k+1, \cdots, \kappa$ and $j = \mu-k, \mu-k+1, \cdots, \mu$, or in terms of vectors $\hat{\tau}$ and $\hat{\tau}^c (= \text{vec}\ \tau^c)$ as

$$E_{\kappa\mu}\hat{\tau} \geq E_{\kappa\mu}\hat{\tau}^c, \tag{22}$$

where $E_{\kappa\mu} \in \mathbf{R}^{(k+1)^2 \times M_1 M_2}$ is defined by

$$E_{\kappa\mu} = [0_{k+1,\mu}\ I_{k+1}\ 0_{k+1,M_2-\mu-k-1}] \otimes [0_{k+1,\kappa}\ I_{k+1}\ 0_{k+1,M_1-\kappa-k-1}] \tag{23}$$

In fact, if (22) holds, we have from (18)-(20) and $N_{i,k}(t) \geq 0\ \forall t \in [0,1]$,

$$x(s,t) = \hat{x}(u,v) = \sum_{i=0}^{k} \sum_{j=0}^{k} \tau_{\kappa-k+i,\mu-k+j} N_{i,k}(u) N_{j,k}(v)$$

$$\geq \sum_{i=0}^{k} \sum_{j=0}^{k} \tau^c_{\kappa-k+i,\mu-k+j} N_{i,k}(u) N_{j,k}(v)$$

$$= \hat{x}^c(u,v) = x^c(s,t) \quad \forall (s,t) \in [s_\kappa, s_{\kappa+1}) \times [t_\mu, t_{\mu+1}). \tag{24}$$

The above arguments can be readily extended to the case of broader region $[s_\kappa, s_\zeta) \times [t_\mu, t_\eta)$ for arbitrary $\zeta (> \kappa)$ and $\eta (> \mu)$. It is noted that the condition in (22) is only sufficient for $x(s,t) \geq x^c(s,t)$ to hold. A simple but useful case of the function $f(s,t)$ in (16) is a constant, i.e. $f(s,t) = c$, where $c \in \mathbf{R}$ is some constant. Then, it can be shown that such constraint is expressed without employing the idea of limiting spline surface as $E_{\kappa\mu}\hat{\tau} \geq \mathbf{c}_{(k+1)^2}$, where $\mathbf{c}_i = [c\ c\ \cdots\ c]^T \in \mathbf{R}^i$.

3.2 Constrained Spline Surfaces

Using the expression of constraints as in Section 3.1, a fairly large number of constrained spline surface problems may be treated. The formulation is quite simple and is very well fit for numerical solutions as quadratic programming problems. Namely, the optimal smoothing spline surfaces are obtained by minimizing the quadratic cost $J(\hat{\tau})$ in (9), whereas a number of constraints on the splines may be expressed as linear constraints on the vector $\hat{\tau}$, either equality or inequality or both. Then, a general form of problems can be written as quadratic programming problems as follows:

(QP1) Find $\hat{\tau}$ such that

$$\min_{\hat{\tau} \in \mathbf{R}^{M_1 M_2}} J(\hat{\tau}) = \frac{1}{2}\hat{\tau}^T G \hat{\tau} + g^T \hat{\tau} \tag{25}$$

subject to the constraints of the form

$$A\hat{\tau} = q,\ f_1 \leq E\hat{\tau} \leq f_2,\ h_1 \leq \hat{\tau} \leq h_2, \tag{26}$$

for some matrices and vectors of appropriate dimensions. A very efficient numerical algorithm is available for this purpose (e.g. Nocedal [2006]).

4. RECURSIVE DESIGN ALGORITHM OF CONSTRAINED SMOOTHING SPLINE SURFACES

Based on the foregoing development, we develop a recursive algorithm of optimal smoothing spline surfaces with constraints. Such an algorithm prevents the size of relevant matrices and vectors from keep growing due to the increasing number of given data.

Now suppose that, up to the p-th recursion, we are given a set of N_p data,

$$\{(u_i, v_i; d_i) : u_i \in [s_0, s_{m_1}], v_i \in [t_0, t_{m_2}], d_i \in \mathbf{R}, i = 1, 2, \cdots, N_p\}, \tag{27}$$

where $p = 1, 2, \cdots$. Here, we assume that the number of data given at the i-th recursion is $n_i (\geq 1),\ i = 0, 1, \cdots$, and hence $N_p = \sum_{i=0}^{p} n_i$. Then, letting $x_{[p]}(s,t)$ be the optimal smoothing spline surface at p-th recursion, we consider the following recursive spline problem of constructing $x_{[p]}(s,t)$ by minimizing

$$J_{[p]}(\tau) = \lambda \int_{I_1} \int_{I_2} \left(\nabla^2 x_{[p]}(s,t) - \nabla^2 x_{[p-1]}(s,t) \right)^2 ds dt$$

$$+\frac{1}{N_p}\sum_{i=1}^{N_p}(x_{[p]}(u_i,v_i)-d_i)^2 \tag{28}$$

subject to some constraints with the form in (26). Let $\tau_{[p]} \in \mathbf{R}^{M_1\times M_2}$ be the solution τ to this problem. Then, our task is to develop an algorithm for recursively generating the sequence of $\tau_{[p]}$, i.e. $\tau_{[0]}, \tau_{[1]}, \cdots$. Then, a sequence of the associated optimal smoothing surfaces, denoted as $x_{[0]}(s,t), x_{[1]}(s,t), \cdots$, are generated by (1).

Such an algorithm is developed as follows: We first introduce $\Delta x_{[p]}$ and $\Delta\hat{\tau}_{[p]}$ defined by

$$\Delta x_{[p]} = x_{[p]} - x_{[p-1]} \tag{29}$$

$$\Delta\hat{\tau}_{[p]} = \hat{\tau}_{[p]} - \hat{\tau}_{[p-1]}, \tag{30}$$

where $\hat{\tau}_{[p]}$ denotes the vec-function of $\tau_{[p]}$, i.e. $\hat{\tau}_{[p]} = \mathrm{vec}\,\tau_{[p]}$. Then, the cost function $J_{[p]}(\tau_{[p]})$ in (28) is written as $\tilde{J}_{[p]}(\Delta\hat{\tau}_{[p]})$,

$$\tilde{J}_{[p]}(\Delta\hat{\tau}_{[p]}) = \Delta\hat{\tau}_{[p]}^T G_p \Delta\hat{\tau}_{[p]} + 2g_p^T\Delta\hat{\tau}_{[p]} + \mathrm{const.} \tag{31}$$

Here, $G_p \in \mathbf{R}^{M_1M_2\times M_1M_2}$ and $g_p \in \mathbf{R}^{M_1M_2}$ are defined as

$$G_p = \lambda Q + \frac{1}{N_p}\bar{\Gamma}_p\bar{\Gamma}_p^T \tag{32}$$

$$g_p = \frac{1}{N_p}(\bar{\Gamma}_p\bar{\Gamma}_p^T\hat{\tau}_{[p-1]} - \bar{\Gamma}_p\bar{d}_p), \tag{33}$$

where we set $\hat{\tau}_{[0]}$ as $\hat{\tau}_{[0]} = 0_{M_1M_2}$. In (32) and (33), $\bar{\Gamma}_p \in \mathbf{R}^{M_1M_2\times N_p}$ and $\bar{d}_p \in \mathbf{R}^{N_p}$ are defined as

$$\begin{aligned}\bar{\Gamma}_p &= \left[b_2(v_1)\otimes b_1(u_1)\ \cdots\ b_2(v_{N_p})\otimes b_1(u_{N_p})]\right] \\ &= [\bar{\Gamma}_{p-1}\ \hat{\Gamma}_p]\end{aligned} \tag{34}$$

$$\begin{aligned}\bar{d}_p &= [d_1\ d_2 \cdots d_{N_p}]^T \\ &= [\bar{d}_{p-1}^T\ \hat{d}_p^T]^T,\end{aligned} \tag{35}$$

where $\hat{\Gamma}_p \in \mathbf{R}^{M_1M_2\times n_p}$ and $\hat{d}_p \in \mathbf{R}^{n_p}$ denote

$$\hat{\Gamma}_p = \left[b_2(v_{N_{p-1}+1})\otimes b_1(u_{N_{p-1}+1})\ \cdots\ b_2(v_{N_p})\otimes b_1(u_{N_p}))\right] \tag{36}$$

$$\hat{d}_p = [d_{N_{p-1}+1}\ d_{N_{p-1}+2}\cdots d_{N_p}]^T. \tag{37}$$

By the expression of $\bar{\Gamma}_p$ in (34) and $\bar{d}_p$ in (35), G_p in (32) and g_p in (33) are rewritten as

$$G_p = G_{p-1} + \left(\frac{1}{N_p} - \frac{1}{N_{p-1}}\right)\bar{\Gamma}_{p-1}\bar{\Gamma}_{p-1}^T + \frac{1}{N_p}\hat{\Gamma}_p\hat{\Gamma}_p^T \tag{38}$$

$$g_p = \frac{1}{N_p}\left(\bar{\Gamma}_p\bar{\Gamma}_p^T\hat{\tau}_{[p-1]} - \bar{\Gamma}_{p-1}\bar{d}_{p-1} - \hat{\Gamma}_p\hat{d}_p\right). \tag{39}$$

The expressions in (38) and (39) yield efficient recursive method for computing G_p and g_p.

On the other hand, by using (30), the constraints (26) on the spline surfaces $x_{[p]}(s,t)$ can be readily expressed as linear constraints in terms of $\Delta\hat{\tau}_{[p]}$. Then, the general form of this problem is identical to the quadratic programming problem (QP1) in Section 3.2:

(QP2) Find $\Delta\hat{\tau}_{[p]}$ such that

$$\min_{\Delta\hat{\tau}_{[p]}\in\mathbf{R}^{M_1M_2}} \tilde{J}_{[p]}(\Delta\hat{\tau}_{[p]}) = \frac{1}{2}\Delta\hat{\tau}_{[p]}^T G_p\Delta\hat{\tau}_{[p]} + g_p^T\Delta\hat{\tau}_{[p]} \tag{40}$$

subject to the constraints of the form

$$A\Delta\hat{\tau}_{[p]} = \bar{q},\ \bar{f}_1 \le E\Delta\hat{\tau}_{[p]} \le \bar{f}_2,\ \bar{h}_1 \le \Delta\hat{\tau}_{[p]} \le \bar{h}_2, \tag{41}$$

where $\bar{q},\ \bar{f}_l,\ \bar{h}_l$ for $l = 1, 2$ are computed by

$$\bar{q} = q - A\hat{\tau}_{[p-1]},\ \bar{f}_l = f_l - E\hat{\tau}_{[p-1]},\ \bar{h}_l = h_l - \hat{\tau}_{[p-1]}. \tag{42}$$

Thus, by solving this problem with respect to $\Delta\hat{\tau}_{[p]}$, we get the control point vector $\hat{\tau}_{[p]}$ for the p-th recursion by (30) as

$$\hat{\tau}_{[p]} = \hat{\tau}_{[p-1]} + \Delta\hat{\tau}_{[p]}. \tag{43}$$

The recursive design algorithm of constrained smoothing spline surfaces can be summarized as follows.

Algorithm 2. The recursive algorithm, after initialization steps (I-1)-(I-6), is carried out in the steps (R-1)-(R-8).

Initialization steps:

- (I-1) Let $p = 0$, and set the parameters k, α, β, λ, s_0, t_0, m_1 (or $M_1(= m_1 + k)$) and m_2 (or $M_2(= m_2 + k)$).
- (I-2) Compute $Q \in \mathbf{R}^{M_1M_2\times M_1M_2}$ in (11), $\bar{\Gamma}_0 \in \mathbf{R}^{M_1M_2\times N_0}$ in (34) and $\bar{d}_0 \in \mathbf{R}^{N_0}$ in (35).
- (I-3) Compute $G_0 \in \mathbf{R}^{M\times M}$ and $g_0 \in \mathbf{R}^M$ by (32) and (33).
- (I-4) Set the constraints in (26) as required.
- (I-5) Find $\hat{\tau}_{[0]}$ by solving (QP1).
- (I-6) Construct $x_{[0]}(s,t)$ by (1).

Recursive steps:

- (R-1) Set $p = p + 1$, and compute $\hat{\Gamma}_p \in \mathbf{R}^{M_1M_2\times n_p}$ in (36) and $\hat{d}_p \in \mathbf{R}^{n_p}$ in (37).
- (R-2) Compute G_p by (38).
- (R-3) Set $\bar{\Gamma}_p\bar{\Gamma}_p^T = \bar{\Gamma}_{p-1}\bar{\Gamma}_{p-1}^T + \hat{\Gamma}_p\hat{\Gamma}_p^T$.
- (R-4) Compute g_p by (39).
- (R-5) Set $\bar{\Gamma}_p\bar{d}_p = \bar{\Gamma}_{p-1}\bar{d}_{p-1} + \hat{\Gamma}_p\hat{d}_p$.
- (R-6) Compute $\bar{q}$, $\bar{f}_l$, $\bar{h}_l$, $l = 1, 2$ by (42) and set the constraints in (41) as required.
- (R-7) Find $\Delta\hat{\tau}_{[p]}$ by solving (QP2) and compute $\hat{\tau}_{[p]}$ by (43).
- (R-8) Construct the spline curve $x_{[p]}(s,t)$ by (1). Go to (R-1).

Compared with the ordinary design method in Section 3, the proposed method must solve the quadratic programming problems at each recursion in Step (R-7). Thus, the computational complexity of proposed method may be modestly increased. However, we can prevent the size of relevant matrices and vectors from keep growing as the total number of data N_p getting larger. It remains to prove the convergence properties of the above method. However, when we assume that the data d_i for constructing smoothing curves are obtained by sampling some curve $f(s,t)$, we may expect that the optimal solution of this problem converges to the one of minimizing

$$J_{[p]}(\tau) = \lambda\int_{I_1}\int_{I_2}\left(\nabla^2 x_{[p]}(s,t) - \nabla^2 x_{[p-1]}(s,t)\right)^2 dsdt$$

$$+\int_{I_1}\int_{I_2}(x_{[p]}(s,t)-f(s,t))^2 dsdt \quad (44)$$

subject to some constraints of the form in (41). Also, the optimal solution $\tau \in \mathbf{R}^{M_1 \times M_2}$ of the problem, denoted by $\tau^c_{[p]}$, can be computed recursively by the similar algorithm as Algorithm 2. The associated splines $x^c_{[p]}(s,t)$ are then obtained.

5. NUMERICAL EXAMPLES

A typical application of optimal smoothing spline surfaces with constraints appears in the problem of constructing and reconstructing 3-dimensional contour in the robotics fields. For example, when a set of data are measured recursively by some range sensor and we need to construct and reconstruct contour representation of an environment by using some continuous periodic surface, the method in the previous section can be used effectively. We here examine performances of recursive design algorithm in the previous section by the following numerical studies.

Let us consider a case where a mobile robot with some range sensor is in a closed environment in 3-dimensional space $o-pqt$ as shown in Figure 1. Here, the mobile robot is illustrated as green cylinder. The blue surface shows the closed environment. We here assume that the closed environment is given as a periodic surface,

$$p(s,t) = (r_c + r_m)\cos\theta(s) - r_m \cos\left(\frac{r_c + r_m}{r_m}\theta(s)\right)$$
$$q(s,t) = (r_c + r_m)\sin\theta(s) - r_m \sin\left(\frac{r_c + r_m}{r_m}\theta(s)\right)$$
$$\forall (s,t) \in [0,10]\times[0,10] \quad (45)$$

with $r_c = 10$, $r_m = 1/5$ and $\theta(s) = \frac{36\pi}{180}s$. Also, letting $f(s,t)$ be the distance between robot and environment, $f(s,t)$ is written as

$$f(s,t) = \sqrt{p^2(s,t) + q^2(s,t)}, \forall (s,t) \in [0,10]\times[0,10]. \quad (46)$$

Now suppose that the data d_i in (27) measured by a range sensor is obtained by sampling $f(s,t)$. Here the number of data is set as $n_i = 30$ at each recursion. (u_i, v_i)'s are randomly spaced in the region $\mathcal{D} = [s_0, s_{m_1}] \times [t_0, t_{m_2}] = [0,10]\times[0,10]$, and the magnitude of the additive noise in d_i is set as $\sigma = 0.01$. However, note that such a measured data may be unreliable unless they are within the measurable range of range sensors. In Figure 1, such a measurable range is plotted as a pink circle with radius $r_a = 13$.

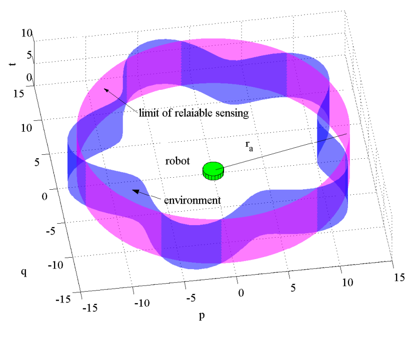


Fig. 1. Constructing contour by a mobile robot with range sensors.

For constructing the reliable contour in $o-pqt$ space, we consider constructing the smoothing spline surfaces $x_{[p]}(s,t)$ in the time domain $\mathcal{D}$ recursively such that

$$0 \le x(s,t) \le r_a (=13),\ \forall (s,t) \in [0,10]\times[0,10]. \quad (47)$$

We set $k = 3$, $\lambda = 10^{-4}$, $\alpha = \beta = 1$, $s_0 = t_0 = 0$ and $m_1 = m_2 = 10$ (i.e. $s_{m_1} = t_{m_2} = 10$). Thus the knot points s_i, t_j are taken as integers as $s_i = i$, $t_j = j$. The periodicity constraints are set as

$$\frac{\partial^l}{\partial s^l}x(s_0,t) = \frac{\partial^l}{\partial s^l}x(s_{t_{m_1}},t),\ \forall t \in [t_0, t_{m_2}] \quad (48)$$

for $l = 0,1,2$. Note that the inequality constraint in (47) is imposed by employing the method in Section 3.1. In addition, (48) is constraints over intervals and the method in the paper (Fujioka [2009a]) can be used. By Algorithm 2, optimal weight $\tau_{[p]}$ for $p = 0,1,\cdots$ are computed together with the associated spline surfaces $x_{[p]}(s,t)$.

Figure 2 shows the results $x_{[p]}(s,t)$ for (a) $p = 0$, (b) $p = 5$ and (c) $p = 30$ in colored surfaces together with data points (green squares). Also, the corresponding 3-dimensional contours plotted in $o-pqt$ space are shown in Figure 3, where we here employ the following coordinates

$$(p(s,t), q(s,t)) = (x(s,t)\cos\theta(s),\ x(s,t)\sin\theta(s)). \quad (49)$$

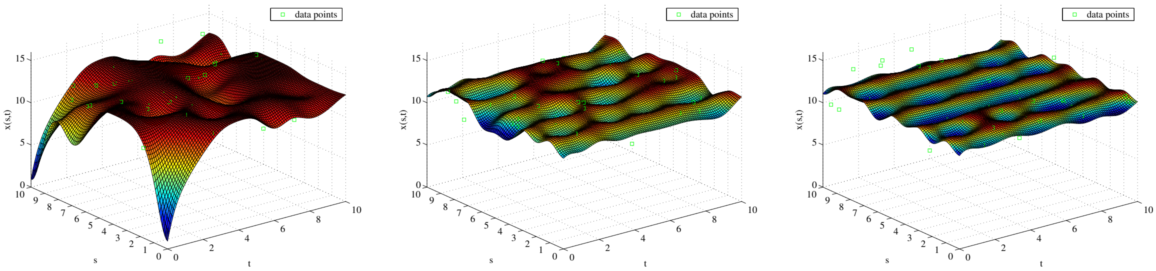
From these figures, we may see that the developed method works quite well and the surface $x_{[p]}(s,t)$ satisfies all the constraints in (47) and (48). Also, Figure 4 shows the function error $||x^c_{[30]}(s,t) - x_{[p]}(s,t)||^2_{L_2}$ and error of weight matrix $||\tau^c_{[30]} - \tau_{[p]}||_2$. Here $x^c_{[30]}(s,t)$ is constructed from $f(s,t)$ by (44), where the corresponding control point matrix is denoted as $\tau^c_{[30]}$. From these figures, we may observe that $x_{[p]}(s,t)$ converges to $x^c_{[p]}(s,t)$ as the iteration number p increases.

6. CONCLUDING REMARKS

We developed a systematic method for recursive design of optimal smoothing spline surfaces with equality and/or inequality constraints. The spline surfaces are constituted employing normalized uniform B-splines as the basis functions. Then the central issue was to determine an optimal matrix of the so-called control points. Such an approach enables us to express various types of constraints as linear function of control points. The design problem becomes a quadratic programming problem in terms of vec-function of control point matrix, and very efficient recursive algorithm was developed. We examined the performances of the design method by numerical examples for 3-dimensional contour constructing problem with equality and inequality constraints. It is concluded that the method is very effective as well as very useful for many applications in various fields including robotics.

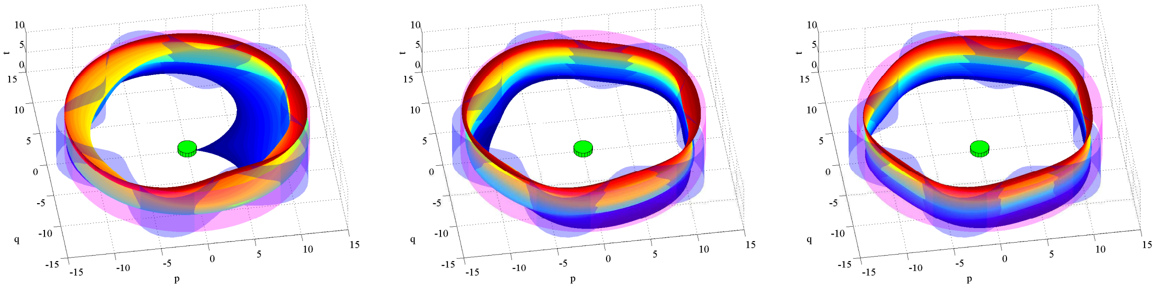
REFERENCES

C. de Boor, *A practical guide to splines*, Revised Edition, Springer-Verlag, New York, 2001.



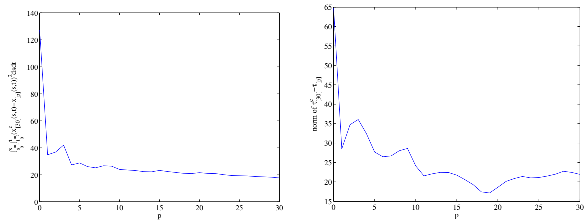
(a) $x^{[0]}(s,t)$ (b) $x^{[5]}(s,t)$ (c) $x^{[30]}(s,t)$

Fig. 2. optimal smoothing splines.



(a) $x^{[0]}(s,t)$ (b) $x^{[5]}(s,t)$ (c) $x^{[30]}(s,t)$

Fig. 3. Periodic surfaces $x(s,t)$ represented in the $o-pqt$ space.



(a) $||x^c_{[30]}(s,t) - x_{[p]}(s,t)||^2_{L_2}$ (b) $||\tau^c_{[30]} - \tau_{[p]}||_2$

Fig. 4. Function error $||x^c_{[30]}(s,t) - x_{[p]}(s,t)||^2_{L_2}$ and weight matrix error $||\tau^c_{[30]} - \tau_{[p]}||_2$.

M. Egerstedt and C. F. Martin, Optimal control and monotone smoothing splines. New trends in nonlinear dynamics and control, and their applications, 279–294, Lecture Notes in Control and Inform. Sci., 295, Springer, Berlin, 2003.

H. Fujioka, H. Kano, M. Egerstedt and C. Martin, Smoothing Spline Curves and Surfaces for Sampled Data, *Int. J. of Innovative Computing, Information and Control*, vol.1, no.3, pp.429-449, 2005.

H. Fujioka and H. Kano, Recursive Construction of Optimal Smoothing Splines, *Proc. of the 40th ISCIE Int. Symp. on Stochastic Systems Theory and Its Applications*, pp.270–275, Kyoto, Japan, Nov. 14-15, 2008.

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.

H. Fujioka and H. Kano, Constrained Smoothing and Interpolating Spline Surfaces using Normalized Uniform B-splines, *Proc. the IEEE Int. Conf. on Industrial Technology*, pp.41-46, Gippsland, Victoria, Australia, Feb. 10-13, 2009.

H. Fujioka and H. Kano, Recursive Construction of Smoothing Spline Surfaces Using Normalized Uniform B-splines, *Proc. of the 48th IEEE Conf. on Decision and Control held jointly with 2009 28th Chinese Control Conf.*, pp.5550-5555 Shanghai, China, Dec. 16-18, 2009.

H. Fujioka and H. Kano, Recursive Construction of Optimal Smoothing Splines with Constraints, *Proc. of the 2010 American Control Conf.*, pp.1580-1585, Baltimore, MD, USA, June 30-July 2, 2010.

R. Frezza and G. Picci, On Line Path Following by Recursive Spline Updating, *Proc. of the IEEE Conf. on Decision and Control*, pp.4047-4052, New Orleans, LA, USA, Dec.13-15, 1995.

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.

M. Karasalo, X. Hu and C.F. Martin, Contour Reconstruction and Matching Using Recursive Smoothing Splines, *Lecture Notes in Control and Information Sciences*, vol.364, pp.193–206, 2007.

H. Kano, H. Fujioka, and C. F. Martin, Optimal Smoothing and Interpolating Splines with Constraints, *Proc. of the 46th IEEE Conf. on Decision and Control*, pp.3011-3016, New Orleans, LA, USA, Dec. 12-14, 2007.

P. Lancaster and M. Tismenetsky, *The Theory of Matrices*, Second Edition, Academic Press, 1985.

C.F. Martin, S. Sun and M. Egerstedt, Optimal control, statistics and path planning. Computation and control, VI (Bozeman, MT, 1998). *Math. Comput. Modelling*, 33, no.1-3, pp.237–253, 2001.

J. Nocedal and S.J. Wright, *Numerical Optimization*, 2nd Edition, Springer, 2006.

G. Piccolo, M. Karasalo, D. Kragic and X. Hu, Contour Reconstruction using Recursive Smoothing Splines - Algorithms and Experimental Validation, *Robotics and Autonomous Systems*, vol.57, issue 6-7, pp.617-628, 2009.