

Spline Trajectory Planning for Path with Piecewise Linear Boundaries

Hiroyuki Kano¹ Hiroyuki Fujioka²

¹Division of Science, Tokyo Denki University, Saitama 350-0394, Japan, kano@mail.dendai.ac.jp

²Department of System Management, Fukuoka Institute of Technology, Fukuoka 811-0295, Japan, fujioka@fit.ac.jp

Abstract

We consider a problem of trajectory planning for path with piecewise linear boundaries. The trajectory is constructed as smoothing splines using normalized uniform B-splines as the basis functions. The boundary constraints are treated as a collection of inequality pairs by right and left boundary lines, and are formulated as linear inequality constraints on the so-called control point vector. Smoothing splines are constructed as an approximation of a piecewise linear centerline of the given path, where the given entire time interval is divided into subintervals according to the centripetal distribution rule. Other constraints as initial and terminal conditions on the trajectory can be included easily, and the problem reduces to convex quadratic programming problem where very efficient numerical solvers are available. The effectiveness of the proposed method is confirmed by an example of fairly complex path with piecewise linear boundaries. Also an example is included to demonstrate its usefulness for trajectory planning in an environment with obstacles.

Keywords: trajectory planning, smoothing spline, B-spline, boundary constraint, quadratic programming problem

1 Introduction

Splines have been used frequently in robotics as in the problems of trajectory planning of robotic arms and mobile robots (Biagiotti and Melchiorri, 2008; Egerstedt and Martin, 2010; Khalil and Dombre, 2002). A typical problem of trajectory planning consists of constructing a function of time that satisfies initial and terminal conditions together with other requirements such as via points and obstacle avoidance.

When via points are specified, trajectories may be constructed as interpolating splines to pass the via points or as approximating splines to pass near the points (Crouch and Jackson, 1991; Egerstedt and Martin, 2001). The problems of obstacle avoidance trajectory planning are often treated by introducing a cost function consisting of distance to obstacles together with e.g. the trajectory length, which are expressed as nonlinear function of some parameters representing the trajectories. Cubic splines are frequently used to construct trajectories (Kolter and Ng,

2009; Saska et al., 2006; Piazzoli and Visioli, 2000), and an optimization problem is solved numerically for trajectory planning. Particle swarm optimization method is employed in (Saska et al., 2006).

In (Gallina and Gasparetto, 2000), representing trajectories by sums of harmonics, the trajectory planning problem is formulated as constrained nonlinear programming problem, where the obstacles are treated as inequality constraints by assuming their parametric representation as polygons and ellipses. Also, treating obstacles as linear inequality constraints in (Berglund et al., 2010), nonlinear programming problem is solved for quartic B-spline curves with minimum curvature. Only B-splines of degrees two to four are allowed.

In this paper, we consider a problem of trajectory planning for road-like path with the right and left boundaries in a 2-dimensional plane. The boundaries are assumed to be given as piecewise linear functions, and the problem is to construct a trajectory from given start point to goal point without exceeding the boundaries. Although such a problem of planning trajectories for path with boundaries naturally arises, the treatment as in this study seems novel to the authors' knowledge.

To be more specific, we construct trajectories as smoothing splines using B-splines as the basis functions (Kano et al., 2005). This approach is very suitable to the present problem, since we can construct trajectories piecewise in accordance with each piece of boundaries. Also, by dividing the entire time interval into subintervals, the problem is to construct the trajectory bounded by two lines for each time interval.

In such problems, our approach by constrained splines (Kano et al., 2011; Fujioka and Kano, 2012; Kano et al., 2014) is very effective. The studies on constrained splines include constraints at isolated time instants, those over an interval of time, constraints on function values, on time derivatives of arbitrary degrees, or on integral values, and so forth. The constraints can be equality and/or inequality, and are systematically included in the formulation by B-spline based smoothing splines.

We show that the problem is formulated as a convex QP (quadratic programming) problem. The description of the problem is easy since the boundaries can be defined by simply providing a series of pairs of right and left cor-

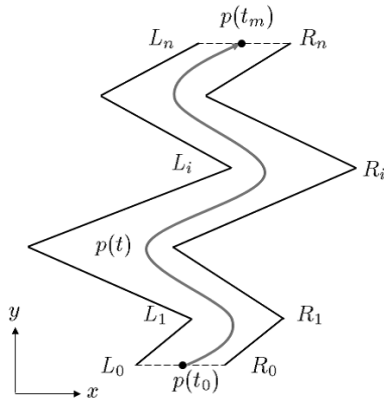


Figure 1. A path with piecewise linear boundaries.

ners, and the resulting QP problem is solved efficiently numerically by existing QP solver. The usefulness of the proposed method is confirmed by two examples: first, for relatively complex path with piecewise linear boundaries, and, second, for application to obstacle avoidance trajectory planning problem.

This paper is organized as follows. In Section 2, we present problem statement and describe 2-dimensional vector smoothing splines based on B-splines. Then in Section 3, the trajectory planning problem is formulated and solved. Two numerical examples are considered in Section 4. Concluding remarks are given in Section 5.

Throughout the paper, the symbol \otimes denotes the Kronecker product, and 'vec' the vec-function (see e.g. (Lancaster and Tismenetsky, 1985)).

2 Preliminaries

2.1 Problem Statement

As shown in Fig. 1, let a path with piecewise linear boundaries be defined by a pair of corner points (R_i, L_i) , $i = 0, 1, \dots, n$ on xy -plane. Then we consider to design a trajectory $p(t) \in \mathbf{R}^2$

$$p(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} \quad (1)$$

for given time interval $[t_0, t_m]$ and for given initial position $p(t_0)$ and final position $p(t_m)$. Our particular interest is to construct a smooth trajectory $p(t)$ that is guaranteed to stay within the path specified by the piecewise linear right boundary $R_0R_1 \dots R_n$ and the left boundary $L_0L_1 \dots L_n$. It is noted that the initial and terminal conditions can be specified as equality and/or inequality conditions on $p(t)$ and its derivatives.

We construct the trajectory $p(t)$ for $t \in [t_0, t_m]$ by splines using normalized uniform B-spline $B_k(t)$ of degree $k(\geq 1)$,

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

Table 1. $N_{j,3}(t)$ ($j = 0, 1, 2, 3$) and its derivatives

j	$3!N_{j,3}(t)$	$2!N_{j,3}^{(1)}(t)$	$N_{j,3}^{(2)}(t)$	$N_{j,3}^{(3)}(t)$
0	$(1-t)^3$	$-(1-t)^2$	$1-t$	-1
1	$4-6t^2+3t^3$	$-4t+3t^2$	$-2+3t$	3
2	$1+3t+3t^2-3t^3$	$1+2t-3t^2$	$1-3t$	-3
3	t^3	t^2	t	1

Here $\tau_i \in \mathbf{R}^2$ are weighting coefficients called control points, and $\alpha(> 0)$ is a constant for scaling the interval between equally-spaced knot points t_i with

$$t_{i+1} - t_i = \frac{1}{\alpha}. \quad (3)$$

Moreover, $B_k(t)$ is defined by

$$B_k(t) = \begin{cases} N_{k-j,k}(t-j) & j \leq t < j+1, \\ & j = 0, 1, \dots, k \\ 0 & t < 0 \text{ or } t \geq k+1, \end{cases} \quad (4)$$

where the basis elements $N_{j,k}(t)$ ($j = 0, 1, \dots, k$), $0 \leq t \leq 1$ can be derived recursively by de Boor's algorithm (de Boor, 2001) for any $k \geq 1$. Note that this basis elements satisfies $N_{j,k}(t) \geq 0 \forall t \in [0, 1]$ and

$$\sum_{j=0}^k N_{j,k}(t) = 1, \quad \forall t \in [0, 1]. \quad (5)$$

Since cubic splines are most frequently used, for reference, we show $N_{j,3}(t)$ together with its derivatives in Table 1. Smoothing splines are used to determine the control points τ_i , or the control point matrix $\tau \in \mathbf{R}^{2 \times M}$ ($M = m+k$)

$$\tau = [\tau_{-k} \quad \tau_{-k+1} \quad \dots \quad \tau_{m-1}] \quad (6)$$

as we see in the sequel.

2.2 Vector Smoothing Splines

In this particular problem, we consider smoothing splines for continuous-time data $f(t) \in \mathbf{R}^2$, where the following cost function is minimized.

$$J(\tau) = \lambda \int_{t_0}^{t_m} \left\| p^{(l)}(t) \right\|_{\Lambda}^2 dt + \int_{t_0}^{t_m} \|p(t) - f(t)\|^2 dt. \quad (7)$$

Here $\lambda(> 0)$ is a smoothing parameter, $\Lambda \in \mathbf{R}^{2 \times 2}$ is a positive-definite weight matrix, $\|u\|^2 = u^T u$, and $\|u\|_{\Lambda}^2 = u^T \Lambda u$. We take the integer l as $l = 2$ for cubic spline ($k = 3$) and $l = 3$ for quintic spline ($k = 5$). Thus the problem is to construct a smooth spline $p(t)$ that approximate the function $f(t)$, which is given, typically so as to represent the center line of the path.

It is noted that usual smoothing spline problem (Wahba, 1990) employs discrete-time set of data (s_i, f_i) in which case the second term in (7) is set as $\sum_{i=1}^N \|p(s_i) - f_i\|_{w_i}^2$.

Now, let $\hat{\tau} \in \mathbf{R}^{2M}$ be the vec-function (Lancaster and Tismenetsky, 1985) of τ , i.e.

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

Then, following the similar procedure as in (Kano et al., 2005), $J(\tau)$ in (7) is expressed as a quadratic function $J(\hat{\tau})$ in $\hat{\tau}$,

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

where $G \in \mathbf{R}^{2M \times 2M}$, $g \in \mathbf{R}^{2M}$ and $g_c \in \mathbf{R}$ are given by

$$G = Q \otimes \Lambda + Q_0 \otimes I_2 \tag{10}$$

$$g = \int_{t_0}^{t_m} b(t) \otimes f(t) dt \tag{11}$$

$$g_c = \int_{t_0}^{t_m} \|f(t)\|^2 dt. \tag{12}$$

Here $b(t) \in \mathbf{R}^M$ is a vector of shifted B-splines defined by

$$b(t) = \begin{bmatrix} B_k(\alpha(t-t_{-k})) & B_k(\alpha(t-t_{-k+1})) \\ \dots & B_k(\alpha(t-t_{m-1})) \end{bmatrix}^T, \tag{13}$$

and $Q, Q_0 \in \mathbf{R}^{M \times M}$ are Gram matrices defined by

$$Q = \int_{t_0}^{t_m} \frac{d^l b(t)}{dt^l} \frac{d^l b^T(t)}{dt^l} dt \tag{14}$$

$$Q_0 = \int_{t_0}^{t_m} b(t) b^T(t) dt \tag{15}$$

3 Trajectory Planning

Recalling that the path is constrained by piecewise linear boundaries as in Fig. 1, it is convenient to plan the trajectory $p(t)$ for each piece of the path constrained by a pair of right and left boundary line segments, $R_i R_{i+1}$ and $L_i L_{i+1}$. The construction of $p(t)$ in (2) is very suitable for this purpose since it is a piecewise polynomial with the knot points t_i .

For this purpose, we divide the time interval $[t_0, t_m]$ into n subintervals $[s_i, s_{i+1}]$, $i = 0, 1, \dots, n-1$ in accordance with n pairs of boundary segments $R_i R_{i+1}$ and $L_i L_{i+1}$. Here we take $[s_i, s_{i+1}]$ to be a knot point interval, namely each s_i is taken as one of the knot point t_j with $s_0 = t_0$ and $s_n = t_m$.

Now denote the straight line passing two points R_i and R_{i+1} by l_i , and the one for points L_i and L_{i+1} by l'_i (see Fig. 2). Then we plan the trajectory $p(t)$ so that, for each $i = 0, 1, \dots, n-1$, it lies between the two boundary lines l_i and l'_i for all t in $[s_i, s_{i+1}]$.

This condition is described in Section 3.1, a method of assigning s_i as a knot point in Section 3.2, and the spline construction procedure will be given in Section 3.3, e.g. taking initial and final conditions into account.

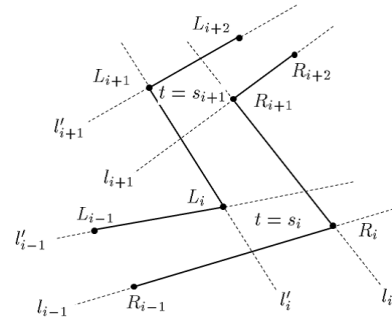


Figure 2. Corner points (R_i, L_i) and boundary lines (l_i, l'_i) .

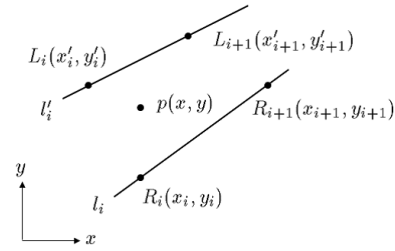


Figure 3. Right and left boundary lines l_i, l'_i and a point p .

3.1 Trajectory between Two Lines

For the present problem, it is natural to introduce the following assumptions (see Figs. 1 and 2).

- (A1) The polygon $\mathcal{P}_i = R_i R_{i+1} L_{i+1} L_i$ is a convex quadrangle for all i .
- (A2) In \mathcal{P}_i , the vertices $R_i, R_{i+1}, L_{i+1}, L_i$ are located counterclockwise.

By (A1), all the four points are distinct and the line segment $R_i R_{i+1}$ does not intersect with $L_i L_{i+1}$. By (A2), the quadrangle \mathcal{P}_i constitutes part of the path with piecewise linear boundaries, or the path is the union of the quadrangles.

Now we derive a condition such that $p(t)$ remains in a region between the two lines l_i and l'_i for all t in a knot point interval $[s_i, s_{i+1}]$, where we let $[s_i, s_{i+1}] = [t_\kappa, t_\mu]$ with $\kappa < \mu$. Here and hereafter, by the term 'region between the two lines l_i and l'_i ', we mean the region between the two lines including the quadrangle \mathcal{P}_i . Thus if lines l_i and l'_i intersect, then among the two wedge-shaped regions, the one including \mathcal{P}_i is meant.

Since the lines l_i and l'_i respectively play the roles of right and left boundaries of the path, we require the point $p(t)$ to lie to the left of l_i when viewed from R_i toward R_{i+1} along l_i , and moreover, $p(t)$ to lie to the right of l'_i when viewed from L_i toward L_{i+1} along l'_i .

In view of Fig. 3, the above conditions for a point $p(x, y)$ to lie between the two lines can be written as

$$\begin{cases} \overrightarrow{R_i p} \times \overrightarrow{R_i R_{i+1}} \leq 0 \\ \overrightarrow{L_i p} \times \overrightarrow{L_i L_{i+1}} \geq 0. \end{cases} \tag{16}$$

Here \times denotes cross product of 2-dimensional vectors, and it holds that, for $u = [u_1 \ u_2]^T$ and $v = [v_1 \ v_2]^T$

$$u \times v = \begin{vmatrix} u_1 & v_1 \\ u_2 & v_2 \end{vmatrix} = u_1 v_2 - u_2 v_1 \quad (17)$$

Applying this relation to the vectors in (16), for example $\overline{R_i p} = [x - x_i \ y - y_i]^T$, (16) can be rewritten as linear inequalities in $p = [x \ y]^T$ as follows.

$$Ap \leq d, \quad (18)$$

where

$$A = \begin{bmatrix} -y_i + y_{i+1} & x_i - x_{i+1} \\ y'_i - y'_{i+1} & -x'_i + x'_{i+1} \end{bmatrix}, \quad (19)$$

$$d = \begin{bmatrix} x_i y_{i+1} - x_{i+1} y_i \\ -x'_i y'_{i+1} + x'_{i+1} y'_i \end{bmatrix}. \quad (20)$$

Next we consider the condition such that $p(t)$ given by (2) stays between the two lines l_i and l'_i for all t in the knot point interval $[t_\kappa, t_\mu]$. By (18), it suffices to derive the condition for $Ap(t) \leq d \ \forall t \in [t_\kappa, t_\mu]$.

First note that, by the definition of $B_k(t)$ in (4), $p(t)$ is written for unit knot point interval $[t_j, t_{j+1})$ as

$$p(t) = \sum_{i=0}^k \tau_{j-k+i} N_{i,k}(\alpha(t - t_j)), \quad t \in [t_j, t_{j+1}), \quad (21)$$

and it depends on only the $k + 1$ weights $\tau_{j-k}, \tau_{j-k+1}, \dots, \tau_j$. Introducing a new variable $u = \alpha(t - t_j)$, we see that $p(t)$ is written as $\hat{p}(u)$,

$$\hat{p}(u) = \sum_{i=0}^k \tau_{j-k+i} N_{i,k}(u), \quad u \in [0, 1). \quad (22)$$

Thus if $A\tau_{j-k+i} \leq d$ holds for $i = 0, 1, \dots, k$, then using (5), we get

$$A\hat{p}(u) = \sum_{i=0}^k A\tau_{j-k+i} N_{i,k}(u) \leq d \sum_{i=0}^k N_{i,k}(u) = d \quad (23)$$

or $Ap(t) \leq d$ for all $t \in [t_j, t_{j+1})$. This interval can be readily extended to $[t_\kappa, t_\mu]$ by imposing the following condition,

$$A\tau_i \leq d, \quad i = \kappa - k, \kappa - k + 1, \dots, \mu - 1. \quad (24)$$

Namely if (24) is satisfied, then it holds that

$$Ap(t) \leq d \ \forall t \in [t_\kappa, t_\mu]. \quad (25)$$

Now the remaining task is to express the inequality condition (24) in terms of the control point vector $\hat{\tau}$ defined in (8). First rewrite (24) as

$$AT_{\kappa,\mu} \leq \mathbf{1}_{\mu-\kappa+k}^T \otimes d \quad (26)$$

with $T_{\kappa,\mu} = [\tau_{\kappa-k} \ \tau_{\kappa-k+1} \ \dots \ \tau_{\mu-1}]$ and $\mathbf{1}_i = [1 \ 1 \ \dots \ 1]^T \in \mathbf{R}^i$. Then noting that $T_{\kappa,\mu}$ is a submatrix of τ consisting of its columns from $\kappa + 1$ through $\mu + k$, it can be expressed as

$$T_{\kappa,\mu} = \tau E_{\kappa,\mu}, \quad (27)$$

where $E_{\kappa,\mu} \in \mathbf{R}^{M \times (\mu - \kappa + k)}$ is defined by

$$E_{\kappa,\mu} = [\ 0_{\mu-\kappa+k, \kappa} \quad I_{\mu-\kappa+k} \quad 0_{\mu-\kappa+k, M-\mu-k} \]^T. \quad (28)$$

Thus (26) is written in τ as

$$A\tau E_{\kappa,\mu} \leq \mathbf{1}_{\mu-\kappa+k}^T \otimes d. \quad (29)$$

Using a formula $\text{vec}(AXB) = (B^T \otimes A)\text{vec}X$ for matrices A, B, X of compatible dimensions (see e.g. (Lancaster and Tismenetsky, 1985)), and noting $\text{vec} \ \tau = \hat{\tau}$, (29) yields

$$(E_{\kappa,\mu}^T \otimes A)\hat{\tau} \leq \mathbf{1}_{\mu-\kappa+k} \otimes d, \quad (30)$$

which is the desired expression.

For convenience we summarize the above developments as follows.

Proposition 1 *The trajectory $p(t)$ lies for all $t \in [s_i, s_{i+1}] = [t_\kappa, t_\mu]$ between the two lines l_i and l'_i if the control point vector $\hat{\tau}$ satisfies the following inequality.*

$$F_i \hat{\tau} \leq h_i \quad (31)$$

where $F_i \in \mathbf{R}^{2(\mu-\kappa+k) \times 2M}$ and $h_i \in \mathbf{R}^{2(\mu-\kappa+k)}$ are given by $F_i = E_{\kappa,\mu}^T \otimes A$ and $h_i = \mathbf{1}_{\mu-\kappa+k} \otimes d$.

3.2 Centerline and Intermediate Time Instants

Recall that it is necessary to allocate the given time interval $[t_0, t_m]$ to n intervals $[s_i, s_{i+1}]$ with s_i being some knot point t_j for each $i = 0, 1, \dots, n - 1$. A natural way is to take the length of center line $C_i C_{i+1}$ (see Fig. 4) into account.

Let $C_i, i = 0, 1, \dots, n$ be defined by

$$C_i = \gamma_i R_i + (1 - \gamma_i) L_i \quad (32)$$

for some $\gamma_i (0 \leq \gamma_i \leq 1)$. Then for determining the time instants s_i in $[t_0, t_m]$, we employ the so-called centripetal distribution (Biagiotti and Melchiorri, 2008). Namely each s_i is determined so that the whole interval $[t_0, t_m]$ is divided into subintervals $[s_i, s_{i+1}]$ in proportion to the following value ζ_i ,

$$\zeta_i = \|C_{i+1} - C_i\|^v \quad (33)$$

for $i = 0, 1, \dots, n - 1$, where $v (0 < v < 1)$ is usually taken as $v = 1/2$. Actually, each s_i is determined as a knot point t_j based on ζ_i . Also it is noted that the centripetal distribution method in above requires less accelerations than other method e.g. cord length distribution, distributed proportionally to $\|C_{i+1} - C_i\|$.

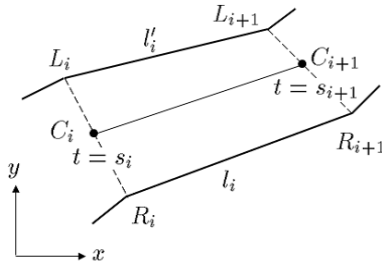


Figure 4. Centerline in the i -th quadrangle.

3.3 Smoothing Spline Trajectory

Obviously we require that, for all $i = 0, 1, \dots, n - 1$, the trajectory $p(t)$ lies between the two lines l_i and l'_i for all $t \in [s_i, s_{i+1}] = [t_k, t_{\mu}]$. Thus by Proposition 1, we impose

$$F_i \hat{\tau} \leq h_i, \quad i = 0, 1, \dots, n - 1. \quad (34)$$

Usually an initial and terminal conditions of $p(t)$ are given, and typical examples are

$$p(t_0) = p_0, \quad p^{(1)}(t_0) = 0, \quad p^{(2)}(t_0) = 0 \quad (35)$$

$$p(t_m) = p_m, \quad p^{(1)}(t_m) = 0, \quad p^{(2)}(t_m) = 0 \quad (36)$$

with p_0 and p_m lying on the line segments R_0L_0 and R_nL_n respectively. Noting that $p(t)$ in (2) is written as $p(t) = \tau b(t)$ and hence $p(t) = \text{vec } p(t) = \text{vec } (\tau b(t)) = (b(t)^T \otimes I_2) \hat{\tau}$, the above initial and final conditions can be written in terms of $\hat{\tau}$ as follows.

$$H(t_0) \hat{\tau} = \bar{h}_0, \quad H(t_m) \hat{\tau} = \bar{h}_m \quad (37)$$

where $H(t) \in \mathbf{R}^{6 \times 2M}$, $\bar{h}_0 \in \mathbf{R}^6$ and $\bar{h}_m \in \mathbf{R}^6$ are

$$H(t) = \begin{bmatrix} b(t)^T \otimes I_2 \\ b^{(1)}(t)^T \otimes I_2 \\ b^{(2)}(t)^T \otimes I_2 \end{bmatrix}, \quad \bar{h}_0 = \begin{bmatrix} p_0 \\ 0_2 \\ 0_2 \end{bmatrix}, \quad \bar{h}_m = \begin{bmatrix} p_m \\ 0_2 \\ 0_2 \end{bmatrix}. \quad (38)$$

The matrices $H(t_0)$ and $H(t_m)$ can be easily set up, for example by using Table 1 when $k = 3$.

Finally in this section we consider the function $f(t)$ used for approximation of smoothing splines in (7). A natural choice will be a function constructed from the piecewise linear centerline $C_0C_1 \dots C_n$ introduced by (32). Specifically we employ a linear function of t in each section as shown in Fig. 4 as

$$f_i(t) = q_i t + r_i, \quad t \in [s_i, s_{i+1}]. \quad (39)$$

Here $q_i, r_i \in \mathbf{R}^2$ are determined so as to satisfy $f_i(s_i) = C_i$ and $f_i(s_{i+1}) = C_{i+1}$. Thus the vector g in (11) is computed from

$$g = \sum_{i=0}^{n-1} \int_{s_i}^{s_{i+1}} b(t) \otimes f_i(t) dt \quad (40)$$

Now we are ready to formulate the trajectory planning problem by constrained smoothing splines with $J(\hat{\tau})$ in

(9), the boundary inequality conditions in (34) and initial and terminal conditions in (37). Namely the problem is to minimize the cost function,

$$\min_{\hat{\tau} \in \mathbf{R}^{2M}} J(\hat{\tau}) = \frac{1}{2} \hat{\tau}^T G \hat{\tau} - g^T \hat{\tau} \quad (41)$$

subject to the constraints of the form

$$A_{eq} \hat{\tau} = d_{eq}, \quad A_{in} \hat{\tau} \leq d_{in}, \quad (42)$$

where G and g are given in (10) and (11) respectively, A_{eq} and d_{eq} are formed as the collection of equalities (37), and A_{in} and d_{in} as collection of inequalities in (34).

Note that, in the case of (37), $A_{eq} \hat{\tau} = d_{eq}$ consists of 12 equality constraints in $2M$ unknowns $\hat{\tau}$. On the other hand, the total number of inequality constraints resulting from (34) is computed as $2M + 2(n - 1)k$, with $2(n - 1)k$ more than the number of unknowns.

This is a convex quadratic programming problem and can be solved numerically by using software tool as the function 'quadprog' in MATLAB. If necessary, other constraints may be introduced such as constraints on the magnitude of velocity or acceleration as long as all the constraints are consistent.

Finally, it is noted that, by our construction, the planned trajectory satisfies the following proposition.

Proposition 2 *The planned spline trajectory $p(t)$ is guaranteed to stay, for each $i = 0, 1, \dots, n - 1$, between the two lines l_i and l'_i as the right and left boundaries for all time $t \in [s_i, s_{i+1}]$. In particular, $p(t)$ is in the corner quadrangle formed from the four lines $l_i, l'_i, l_{i-1}, l'_{i-1}$ at time $t = s_i$ for $i = 1, 2, \dots, n - 1$.*

4 Numerical Examples

Two examples of trajectory planning are considered, where we use cubic splines, namely we set $k = 3$ and $l = 2$ in (2) and (7). MATLAB function 'quadprog' is used for numerical solution of quadratic programming problems.

4.1 Path with Piecewise Linear Boundaries

We consider a path with the right and left corners R_i and L_i given as in Table 2.

Table 2. The coordinates R_i and L_i of right and left corners for $i = 0, 1, \dots, n$ with $n = 9$.

	0	1	2	3	4	5	6	7	8	9
R_i	1	2	2	8	3	8	2	13	13	14
	0	2	9	9	6	4	0	0	8	9
L_i	0	1	1	12	6	11	6	12	12	14
	0	2	10	10	6	4	1	1	8	10

For this path, we construct a smoothing spline trajectory $p(t)$ in time interval $[t_0, t_m] = [0, 10]$. The initial and final conditions are as given in (35) and (36) with $p_0 = (R_0 +$

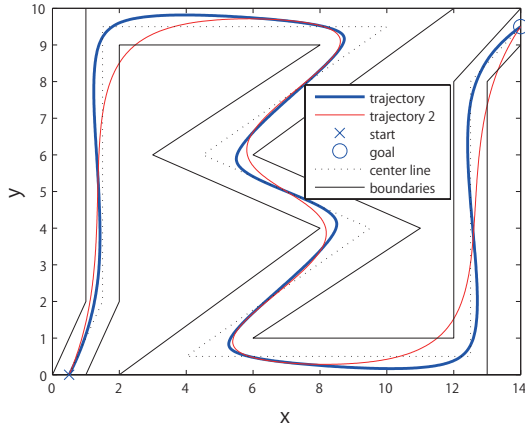


Figure 5. Constructed spline trajectories plotted in xy plane with and without the boundary constraints in (34).

$L_0)/2$ and $p_m = (R_9 + L_9)/2$. We set $\gamma_i = 1/2 \forall i$ in (32) for centerline of the path, which together with (39) is used to set $f(t)$ in (7). The number of knot points $m = 80$, and hence the knot point interval is $t_{i+1} - t_i = 10/80 = 0.125$. The smoothing parameter is set as $\lambda = 0.01$ in (7).

Fig. 5 shows the given path with the boundaries denoted by black lines and the constructed trajectory $p(t) = [x(t), y(t)]^T$ on the xy plane in thick blue line. The start and goal positions are denoted by \times and \circ respectively, and the center line is shown in dotted lines. We observe that the trajectory satisfies the boundary constraints in addition to initial and final conditions. The red line is a smoothing spline trajectory $p2(t) = [x2(t), y2(t)]^T$ constructed similarly as $p(t)$ but without the inequality boundary constraints (34). It exceeds the boundaries at three corners, and the effect of introducing the inequality constraints is apparent.

4.2 Path in Obstacle Avoidance Problem

The proposed method can be used for planning trajectories of mobile robots in an environment with obstacles. As shown in Fig. 6, let us consider an environment with three 'obstacles', denoted by yellow rectangles, and the start and goal denoted by \times and \circ respectively. If we decide to take the upper route of the table-like obstacle in the figure, which will be shorter than taking the lower route, the path with piecewise linear boundaries can be defined, for example by setting the right and left corners (R_i, L_i) as shown in Table 3. These boundaries are shown in Fig. 6 in black dotted lines with the sign \triangle and ∇ for right and left boundaries respectively.

We plan trajectories for this path with the time interval $[t_0, t_m] = [0, 10]$, initial and final conditions are as given in (35) and (36), $m = 50$ and the smoothing parameter $\lambda = 0.1$. Two cases of the parameters γ_i in (32) are considered for the centerline of the path as shown in Table 3. The first case (i) sets the line in the middle of the path, whereas the second case (ii) sets the line closer to the in-

Table 3. The coordinates R_i and L_i of right and left corners and the parameter γ_i for centerline for $i = 0, 1, \dots, n$ with $n = 5$.

	0	1	2	3	4	5
R_i	3	3	4	4	13	13
	0	8	8	7	7	5
L_i	0	0	6	6	16	16
	0	12	12	10	10	5
γ_i : Case (i)	1/2	1/2	1/2	1/2	1/2	1/2
γ_i : Case (ii)	1/2	2/3	2/3	1/3	2/3	1/2

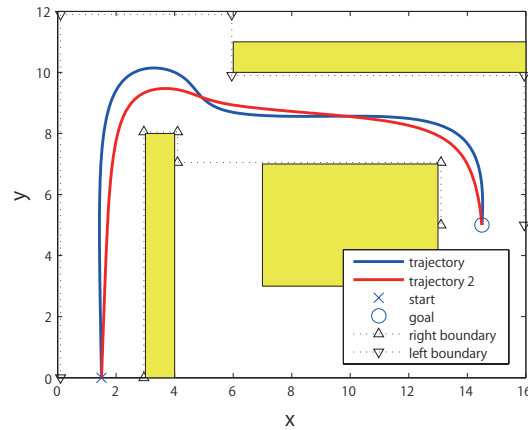


Figure 6. Constructed spline trajectories plotted in xy plane for two Cases (i) and (ii) of the centerline (see Table 3).

ner corner of the path. The planned trajectories are shown in the xy plane in thick blue line for Case (i) and thick red line for Case (ii). We see that reasonable trajectories have been generated where the second case seems more desirable, and that the parameters γ_i for the center line could be effectively used to adjust the trajectory. Note that the path is defined by only six pairs of right and left corners.

5 Concluding Remarks

We presented a method of trajectory planning for path with piecewise linear right and left boundaries. The trajectory is constructed as smoothing splines employing normalized uniform B-splines as the basis functions. Obviously, such boundaries can be described by simply providing a series of pairs of right and left corners (R_i, L_i) , and the problem can be readily defined.

The boundary constraints could be expressed as linear inequality constraints on the control point vector \hat{c} . For constructing smoothing splines, we introduced a piecewise linear centerline of the path in accordance with each pair of right and left boundaries. An appropriate time interval for each piece of the centerline is given based on the centripetal distribution rule. It is shown that the problem is formulated as convex quadratic programming problem. We confirmed the effectiveness of the proposed method by two numerical examples.

One of the future issues is to relax the assumption (A1) in Section 3.1 to allow the multiple corner points as $R_i = R_{i+1}$. Extensions of present method are also important, e.g. to the cases of higher order boundary curves and to the planning in 3-dimensional space.

References

- T. Berglund, A. Brodnik, H. Jonsson, M. Staffanson, and I. Soderkvist. Planning smooth and obstacle-avoiding b-spline paths for autonomous mining vehicles. *IEEE Trans. Automation Sci. and Eng.*, 7(1):167–172, 2010.
- L. Biagiotti and C. Melchiorri. *Trajectory Planning for Automatic Machines and Robots*. Springer, 2008.
- P. Crouch and J. Jackson. Dynamic interpolation and application to flight control. *J. of Guidance, Control and Dynamics*, 14: 814 – 822, 1991.
- C. de Boor. *A practical guide to splines, Revised Edition*. Springer-Verlag, New York, 2001.
- M. Egerstedt and C. F. Martin. Optimal trajectory planning and smoothing splines. *Automatica*, 37(7):1057–1064, 2001.
- M. Egerstedt and C.F. Martin. *Control Theoretic Splines: optimal control, statistics and path planning*. Princeton University Press, New Jersey, 2010.
- H. Fujioka and H. Kano. Optimal vector smoothing splines with coupled constraints. *Trans. Institute of Systems, Control and Information Engineers*, 25(11):299–307, 2012.
- P. Gallina and A. Gasparetto. A technique to analytically formulate and to solve the 2-dimensional constrained trajectory planning problem for a mobile robot. *J. Intelligent and Robotic Systems*, 27:237–262, 2000.
- 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*, 169(1):96–128, 2005.
- H. Kano, H. Fujioka, and C. F. Martin. Optimal smoothing and interpolating splines with constraints. *Applied Mathematics and Computation*, 218(5):1831–1844, 2011.
- H. Kano, H. Fujioka, and C. Martin. Optimal smoothing spline with constraints on its derivatives. *SICE Journal of Control, Measurement, and System Integration*, 7(2):104–111, 2014.
- W. Khalil and E. Dombre. *Modeling, Identification and Control of Robots*. Hermes Penton Ltd., 2002.
- J. Z. Kolter and A. Y. Ng. Task-space trajectories via cubic spline optimization. In *Proc. of the 2009 Int. Conf. on Robotics and Automation*, pages 1675–1682, Kobe, Japan, May 12–17, 2009.
- P. Lancaster and M. Tismenetsky. *The Theory of Matrices, Second Edition*. Academic Press, 1985.
- A. Piazzoli and A. Visioli. Global minimum-jerk trajectory planning of robot manipulators. *IEEE Trans. Industrial Electronics*, 47(1):140 – 149, 2000.
- S. Saska, M. Macas, L. Preucil, and L. Lhotska. Robot path planning using particle swarm optimization of Ferguson splines. In *Proc. IEEE Conf. on Emerging Technologies and Factory Automation (ETFA '06)*, pages 833 – 839, Prague, Sept. 20–22, 2006.
- G. Wahba. *Spline models for observational data*. CBMS-NSF Regional Conference Series in Applied Mathematics, 59, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1990.