2011 American Control Conference
on O'Farrell Street, San Francisco, CA, USA
June 29 - July 01, 2011

A Quadratic Programming Approach to Path Smoothing

Yiming Zhao and Panagiotis Tsiotras

Abstract— **This paper presents a method for smoothing a path in an environment with obstacles. Some characteristic nodes of the path are updated in each iteration by solving a quadratic program, which is formulated based on the smoothness constraints and the local environment information. The generated path satisfies the prescribed smoothness constraints, such as bounds on the curvature, and avoids any collision with obstacles. The proposed method is easy to implement and computationally efficient.**

I. Introduction

Let $\mathbf{r}(s) = \{(x(s), y(s)) : 0 \le s \le s_f\} \in \mathbb{R}^2$ represent a parameterized path to be followed by a vehicle, where s is the arc length coordinate. While obstacles pose constraints on the image of $\mathbf{r}$, vehicle dynamics place constraints on its higher order derivatives. The challenge with the smooth path planning problem lies in the coordination between these two types of constraints.

The most commonly used high order path constraint is the curvature constraint. Although the Dubins vehicle path addresses curvature constraints, the result is optimal only for a vehicle with constant speed [1]. For more realistic vehicles with acceleration/decceleration capability, curvature has greater influence on both the optimality and feasibility of the path. For example, the traveling time along a longer path with small maximum curvature can be shorter than that along a shorter path with large maximum curvature [2]. Besides, a path may be infeasible due to a "minor" violation of the curvature constraint such that the feasibility can be recovered by a small variation of the path. Hence, smoothing a path via local curvature regulation may lead to an improvement in terms of feasibility and optimality.

A discontinuity in the curvature profile implies an instantaneous change of the steering wheel angle for a car-like vehicle or the bank angle or angle of attack for a fixed-wing aircraft, both of which require (theoretically) infinite control force. Therefore, the curvature of the path should be at least continuous for practical applications. For this reason clothoid arcs have been used for continuous-curvature path planning based on the Dubins' path prototype [3], [4], [5]. Reference [6] used analytical splines and heuristics for smooth path generation. Reference [7] proposed a path planning algorithm which generates a smooth path by smoothing out the corners of a linear path prototype using Bézier curves based on analytic expressions. Although all these methods can generate paths with continuous curvature, obstacle avoidance is not guaranteed by these methods per se, and can only be done in an ad hoc manner.

Y. Zhao is a Ph.D. candidate with the School of Aerospace Engineering, Georgia Institute of Technology, 270, Ferst Drive, Atlanta, GA 30332, USA, E-mail: yzhao7@gatech.edu

P. Tsiotras is a Professor with the School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA, Email: tsiotras@gatech.edu

One approach for smooth path planning in the presence of obstacles is to use a "channel" or "corridor," which is selected a priori, such that it does not intrude any of the obstacles. A smooth path is then found within the channel such that it is collision-free. For instance, [8] introduced a method for generating curvature-bounded paths in rectangular channels; reference [9] proposed a method for constructing bounded curvature paths traversing a constant width region in the plane, called corridors, and reference [10] introduced a method for generating smooth two-dimensional paths within two-dimensional bounding envelops using B-spline curves. A nonlinear optimization scheme is used to design collision-free and curvature-continuous paths in [11].

In this paper we follow a quadratic optimization approach for smooth path generation subject to curvature and obstacle clearance constraints. The proposed method minimizes the weighted L_2 norm of the curvature along the path, which is analogous to the strain energy stored in a deflected elastic beam. During the optimization process, a sequence of obstacle-free perturbations are generated along the normal direction of the path, which is based on the perturbation technique proposed in [12] for eliminating noise in GPS measurement data. When combined with other path planning algorithms that provide the initial collision-free path prototype, the proposed method generates collision-free paths under length and local curvature constraints.

II. Curve Representation and Variation

Instead of dealing with a curve (path) in the infinite dimensional space, we reduce the dimensionality of the problem by considering a finite number of characteristic nodes on the curve, and represent the path using a piecewise Bézier curve passing through those nodes.

To this end, suppose that the path is defined in parametric form as $\mathbf{r}(s) = [x(s), y(s)]^{\mathrm{T}}$, parameterized by its arc length s. The curve passes through N characteristic nodes $\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_N \in \mathbb{R}^2$ at $s_1, s_2, \ldots, s_N$, respectively, i.e., $\mathbf{r}(s_i) = \mathbf{r}_i$, $i = 1, 2, \ldots, N$, where $s_1 = 0$ and $s_N = s_f$. It is required that the path must have continuous derivatives at least to the second order. Within this context, the smoothing of the path is equivalent to the deployment of the $N-2$ characteristic nodes subject to certain smoothness criteria.

A. Continuous Curvature Path Representation

1) Approximation of path derivatives using Lagrange interpolation: The second order Lagrange interpolating polynomial curves to approximate the first and second order derivatives of $\mathbf{r}$. Each curve passes through the i^{th} characteristic node $\mathbf{r}_i$ and its two neighboring nodes $\mathbf{r}_{i-1}$ and

978-1-4577-0079-8/11/$26.00 ©2011 AACC

$\mathbf{r}_{i+1}$, $i = 2, \ldots, N-1$. Then we have

$$\mathbf{r}_i' = \alpha_{i,-1}\mathbf{r}_{i-1} + \alpha_i \mathbf{r}_i + \alpha_{i,1}\mathbf{r}_{i+1}, \quad (1)$$
$$\mathbf{r}_i'' = \beta_{i,-1}\mathbf{r}_{i-1} + \beta_i \mathbf{r}_i + \beta_{i,1}\mathbf{r}_{i+1}, \quad (2)$$

where, $\mathbf{r}_i' = \mathbf{r}'(s_i)$ and $\mathbf{r}_i'' = \mathbf{r}''(s_i)$. The coefficients $\alpha_{i,-1}$, α_i, $\alpha_{i,1}$, $\beta_{i,-1}$, β_i, $\beta_{i,1}$ are determined by Lagrange interpolation. For a planar smooth curve, the tangent vector $\mathbf{t}(s)$ is the derivative of $\mathbf{r}(s)$ with respect to the arc length, i.e., $\mathbf{t}(s) = \mathbf{r}'(s)$. The normal vector, which is perpendicular to $\mathbf{t}$, can be obtained by $\mathbf{n}(\mathbf{s}) = \mathbf{A}\mathbf{t}(\mathbf{s}) = \mathbf{A}\mathbf{r}'(\mathbf{s})$, where

$$\mathbf{A} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}.$$

Since $\mathbf{r}''(s) = \mathbf{t}'(s)$, according to the Frénet formula $\mathbf{t}'(s) = \kappa(s)\mathbf{n}(s)$, where the signed curvature κ is given by

$$\kappa(s) = \mathbf{n}^{\mathrm{T}}(s)\mathbf{r}''(s) = \langle \mathbf{A}\mathbf{r}'(s), \mathbf{r}''(s) \rangle, \quad (3)$$

where $\langle \cdot, \cdot \rangle$ denotes the standard inner product in $\mathbb{R}^2$.

2) Bézier curve interpolation: In order to generate a smooth path, a piecewise Bézier curve is used to interpolate the characteristic nodes. Specifically, between each pair of adjacent characteristic nodes, a Bézier interpolating curve is constructed to match the previously introduced first and second path derivative approximations. To this end, the fifth order Bézier curve is a natural choice. The reader may refer to [13] for more details about Bézier curves. The overall path, as the concatenation of Bézier curves between those neighboring nodes, provides continuous curvature by construction.

B. Path Variation

Consider a specific variation of the path $\mathbf{r}(s)$ along its "normal direction" $\mathbf{n}(s)$ only, i.e., the perturbed path $\tilde{\mathbf{r}}(s)$ is given by

$$\tilde{\mathbf{r}}(s) = \mathbf{r}(s) + \delta(s)\mathbf{n}(s), \quad (4)$$

where $\delta(s)$ is the variation function, $s \in [0, s_f]$.

The signed curvature of the perturbed curve $\tilde{\mathbf{r}}$ at each characteristic node for $i = 2, 3, \ldots, N-1$ is given by equation (5), which is a quadratic function in terms of the perturbations δ_{i-1}, δ_i and δ_{i+1} at three neighboring nodes. The curvature at the first and last nodes are determined by the boundary conditions, which are discussed later in this paper. Assuming that the variation is small enough, the local curvature $\tilde{\kappa}_i$ of the perturbed path at the i^{th} characteristic node can be approximated, by neglecting the quadratic terms in equation (5), as follows:

$$\begin{aligned}
&\tilde{\kappa}_i \approx \langle \mathbf{A}\mathbf{r}'_i, \mathbf{r}''_i \rangle \\
&+ (\langle \mathbf{A}\mathbf{r}'_i, \beta_{i,-1}\mathbf{n}_{i-1} \rangle + \langle \mathbf{r}''_i, \alpha_{i,-1}\mathbf{A}\mathbf{n}_{i-1} \rangle)\delta_{i-1} \\
&+ (\beta_i + \langle \mathbf{r}''_i, \alpha_i \mathbf{A}\mathbf{n}_i \rangle)\delta_i \\
&+ (\langle \mathbf{A}\mathbf{r}'_i, \beta_{i,1}\mathbf{n}_{i+1} \rangle + \langle \mathbf{r}''_i, \alpha_{i,1}\mathbf{A}\mathbf{n}_{i+1} \rangle)\,\delta_{i+1} \\
&= \kappa_i + \chi_{i-1,i}\delta_{i-1} + \chi_i \delta_i + \chi_{i+1,i}\delta_{i+1},
\end{aligned} \quad (6)$$

where $\chi_{i-1,i}$, χ_i and $\chi_{i+1,i}$ are constants introduced for the convenience of notation.

III. Quadratic Programming Formulation for the Path Smoothing Problem

In this section we formulate the path smoothing problem as a quadratic program, which approximately minimizes the L_2 norm of the curvature profile, while maintaining the path length and local curvature constraints, boundary conditions and collision-avoidance.

Definition 3.1: The problem

$$\min J(x), \quad x \in \mathcal{D} \subseteq \mathbb{R}^n$$

is a *linear-quadratic mathematical programming problem* (or a *quadratic program* for short), if J is a linear-quadratic function, that is,

$$J(x) = \frac{1}{2}x^{\mathrm{T}}Hx + F^{\mathrm{T}}x + c, \quad (7)$$

where $H = H^{\mathrm{T}} \in \mathbb{R}^{n \times n}$, $F \in \mathbb{R}^n$, and $c \in \mathbb{R}$, and $\mathcal{D}$ is a convex polyhedron, namely $\mathcal{D} = \{x \in \mathbb{R}^n : Ax \leq b\}$, where $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$.

Note that $\mathcal{D}$ is a convex set. A linear quadratic programming problem is a special case of a convex optimization problem when H is a positive semi-definite matrix. Both can be solved very efficiently using numerical methods.

A. Quadratic Cost Function

The weighted L_2 norm of the signed curvature function of the perturbed path is defined by

$$\|\tilde{\kappa}\|_2 \triangleq \left(\int_{s_0}^{s_f} w(s)\tilde{\kappa}^2(s)\mathrm{d}s \right)^{\frac{1}{2}}, \quad (8)$$

where $w(s)$ is a positive definite weighting function. Next, we propose an appropriate discretization of this cost function.

Let K be the vector of the signed curvature of the original path evaluated at the characteristic nodes, i.e., $K = [\kappa(s_1), \kappa(s_2), \ldots, \kappa(s_N)]^{\mathrm{T}}$ and, similarly, let $\tilde{K}$ be the vector of the curvature of the perturbed path evaluated at the same nodes. We can then write equation (6) for all characteristic nodes in a matrix form as $\tilde{K} \approx K + CX$, where $X = [\delta_1, \cdots, \delta_N]^{\mathrm{T}}$, and C is a full-rank $N \times N$ matrix with its entries determined by equation (6) except for χ_{12} and $\chi_{N,N-1}$, which are computed from the boundary conditions, i.e., the tangent directions at the start and end point of the path.

Let W be a diagonal matrix with positive diagonal elements $w_1, w_2, \ldots, w_N$ such that $w_i = w(s_i)$, $i = 1, 2, \ldots, N$. Then the integral in (8) can be approximated using, say, the trapezoidal rule, as follows:

$$\begin{aligned}
&\int_0^{s_f} w(s)\tilde{\kappa}^2(s)\mathrm{d}s \approx \frac{1}{2}\tilde{\kappa}_1^2 w_1 (s_2 - s_1) + \\
&\frac{1}{2}\sum_{i=2}^{N-1} \tilde{\kappa}_i^2 w_i (s_{i+1} - s_{i-1}) + \frac{1}{2}\tilde{\kappa}_N^2 w_N (s_N - s_{N-1}) \\
&= \tilde{K}^{\mathrm{T}} W \Delta_s \tilde{K} \approx (K + CX)^{\mathrm{T}} W \Delta_s (K + CX),
\end{aligned}$$

where $\Delta_s = \frac{1}{2}\mathrm{diag}([s_2 - s_1, s_3 - s_1, s_4 - s_2, \ldots, s_{N-1} - s_{N-3}, s_N - s_{N-2}, s_N - s_{N-1}])$. In order to regulate the

$$
\begin{aligned}
\tilde{\kappa}_i &= \left\langle \mathbf{A}\tilde{\mathbf{r}}_i', \tilde{\mathbf{r}}_i'' \right\rangle = \left\langle \alpha_{i,-1}\mathbf{A}\tilde{\mathbf{r}}_{i-1} + \alpha_i\mathbf{A}\tilde{\mathbf{r}}_i + \alpha_{i,1}\mathbf{A}\tilde{\mathbf{r}}_{i+1}, \beta_{i,-1}\tilde{\mathbf{r}}_{i-1} + \beta_i\tilde{\mathbf{r}}_i + \beta_{i,1}\tilde{\mathbf{r}}_{i+1} \right\rangle \\
&= \langle \alpha_{i,-1}\mathbf{A}\mathbf{r}_{i-1} + \alpha_i\mathbf{A}\mathbf{r}_i + \alpha_{i,1}\mathbf{A}\mathbf{r}_{i+1} + \alpha_{i,-1}\mathbf{A}\mathbf{n}_{i-1}\delta_{i-1} + \alpha_i\mathbf{A}\mathbf{n}_i\delta_i + \alpha_{i,1}\mathbf{A}\mathbf{n}_{i+1}\delta_{i+1}, \\
&\quad\ \beta_{i,-1}\mathbf{r}_{i-1} + \beta_i\mathbf{r}_i + \beta_{i,1}\mathbf{r}_{i+1} + \beta_{i,-1}\mathbf{n}_{i-1}\delta_{i-1} + \beta_i\mathbf{n}_i\delta_i + \beta_{i,1}\mathbf{n}_{i+1}\delta_{i+1} \rangle \\
&= \left\langle \mathbf{A}\mathbf{r}_i' + \alpha_{i,-1}\mathbf{A}\mathbf{n}_{i-1}\delta_{i-1} + \alpha_i\mathbf{A}\mathbf{n}_i\delta_i + \alpha_{i,1}\mathbf{A}\mathbf{n}_{i+1}\delta_{i+1}, \mathbf{r}_i'' + \beta_{i,-1}\mathbf{n}_{i-1}\delta_{i-1} + \beta_i\mathbf{n}_i\delta_i + \beta_{i,1}\mathbf{n}_{i+1}\delta_{i+1} \right\rangle \\
&= \left\langle \mathbf{A}\mathbf{r}_i', \mathbf{r}_i'' \right\rangle + \left\langle \mathbf{A}\mathbf{r}_i', \beta_{i,-1}\mathbf{n}_{i-1}\delta_{i-1} + \beta_i\mathbf{n}_i\delta_i + \beta_{i,1}\mathbf{n}_{i+1}\delta_{i+1} \right\rangle \\
&\quad + \langle \mathbf{r}_i'', \alpha_{i,-1}\mathbf{A}\mathbf{n}_{i-1}\delta_{i-1} + \alpha_i\mathbf{A}\mathbf{n}_i\delta_i + \alpha_{i,1}\mathbf{A}\mathbf{n}_{i+1}\delta_{i+1} \rangle \\
&\quad + \langle \alpha_{i,-1}\mathbf{A}\mathbf{n}_{i-1}\delta_{i-1} + \alpha_i\mathbf{A}\mathbf{n}_i\delta_i + \alpha_{i,1}\mathbf{A}\mathbf{n}_{i+1}\delta_{i+1}, \beta_{i,-1}\mathbf{n}_{i-1}\delta_{i-1} + \beta_i\mathbf{n}_i\delta_i + \beta_{i,1}\mathbf{n}_{i+1}\delta_{i+1} \rangle ,
\end{aligned} \tag{5}
$$

curvature profile, we therefore use the following cost function, which approximates the square of the L_2 norm of the curvature:

$$J(X) = (K + CX)^{\mathrm{T}} W\Delta_s (K + CX). \tag{9}$$

It is easily seen that J is a convex linear-quadratic function because the matrix $C^{\mathrm{T}}W\Delta_s C$ is positive definite.

B. Path Length Constraint

Because the length of the path affects the traveling time, it is desirable to have a constraint on the total length of the path. When a path is perturbed at each node along the normal direction, the total length of the path is not necessarily preserved—it could either increase or decrease depending on the perturbation scenario. Therefore, it is necessary to characterize the relationship between the perturbation and the change of the total length of the curve, and implement certain bounds on the latter.

When the spacing between adjacent characteristic nodes is small enough, the total length of the curve can be approximated by the total length of the line segments connecting each pair of the adjacent nodes. Let D_i denote the change of the length of the line segment between nodes $\mathbf{r}_i$ and $\mathbf{r}_{i+1}$ induced by the perturbation δ. The new positions of the nodes after the perturbation are given by $\tilde{\mathbf{r}}_i = \mathbf{r}_i + \delta_i\mathbf{n}_i$ and $\tilde{\mathbf{r}}_{i+1} = \mathbf{r}_{i+1} + \delta_{i+1}\mathbf{n}_{i+1}$. Then $\|\tilde{\mathbf{r}}_{i+1} - \tilde{\mathbf{r}}_i\|$ is the length of the corresponding line segment of the perturbed path. We assume that the variations δ_i and δ_{i+1} are small enough and $\delta_i,\ \delta_{i+1} \ll \|\mathbf{r}_{i+1} - \mathbf{r}_i\|$. The length of the line segment of the perturbed path between nodes s_i and s_{i+1} is

$$
\begin{aligned}
\|\tilde{\mathbf{r}}_{i+1} - \tilde{\mathbf{r}}_i\| &= \|\mathbf{r}_{i+1} + \delta_{i+1}\mathbf{n}_{i+1} - \mathbf{r}_i - \delta_i\mathbf{n}_i\| \\
&= \sqrt{\|(\mathbf{r}_{i+1} - \mathbf{r}_i) + (\delta_{i+1}\mathbf{n}_{i+1} - \delta_i\mathbf{n}_i)\|^2}.
\end{aligned}
$$

By the polarization identity for the Euclidean inner product,

$$
\begin{aligned}
\|\tilde{\mathbf{r}}_{i+1} - \tilde{\mathbf{r}}_i\| =& \big(\|\mathbf{r}_{i+1} - \mathbf{r}_i\|^2 + \|\delta_{i+1}\mathbf{n}_{i+1} - \delta_i\mathbf{n}_i\|^2 \\
&+ 2\left\langle \delta_{i+1}\mathbf{n}_{i+1} - \delta_i\mathbf{n}_i, \mathbf{r}_{i+1} - \mathbf{r}_i \right\rangle \big)^{\frac{1}{2}}.
\end{aligned}
$$

Then the segment length D_i can be written as in (10). By the small variation assumption, and dropping the square terms, expression (10) yields the following approximation for D_i

$$D_i \approx \left\langle \frac{\mathbf{r}_{i+1} - \mathbf{r}_i}{\|\mathbf{r}_{i+1} - \mathbf{r}_i\|}, \delta_{i+1}\mathbf{n}_{i+1} \right\rangle - \left\langle \frac{\mathbf{r}_{i+1} - \mathbf{r}_i}{\|\mathbf{r}_{i+1} - \mathbf{r}_i\|}, \delta_i\mathbf{n}_i \right\rangle. \tag{12}$$

The summation of all D_i's over all line segments approximates the change of the total length of the curve owing to the variation X.

In order to write equation (12) in a more compact form, let $B = \mathrm{diag}([1/\|\mathbf{r}_2 - \mathbf{r}_1\|, \ldots, 1/\|\mathbf{r}_N - \mathbf{r}_{N-1}\|])$, and define the matrix G as in (11). Also, let $\mathbf{1}_{N-1}$ denote the $N-1$ dimensional column vector with all elements equal to one. Let $\Delta_L(X)$ denote the change of the total length of the path induced by the variation X. Then Δ_L can be approximated by $\Delta_L(X) \approx \mathbf{1}_{N-1}^{\mathrm{T}}BGX$, which is a linear function of X. The constraint on the total length of the path is given by the following linear inequality constraint on X:

$$L_{\min} - L \leq \Delta_L(X) \leq L_{\max} - L, \tag{13}$$

where L is the length of the path before perturbation, and $L_{\max}$ and $L_{\min}$ are the upper and lower bounds of the path length, respectively. These inequalities are enforced elementwise. Alternatively, if the length of the path is fixed, then the linear equality constraint $\Delta_L(X) = 0$ is applied ($L_{\min} = L = L_{\max}$):

C. Curvature Constraints

Local curvature constraints are important for practical path planning. For example, a ground vehicle requires a larger turning radius when moving on a slippery surface compared with the same operation on normal ground. Let $K_{\max,i}$ and $K_{\min,i}$ be the maximum and minimum curvature constraints allowed in a neighborhood of $\mathbf{r_i}$ $(i = 1, 2, \ldots, N)$ which are determined by the vehicle dynamics and the local environment. Let $K_{\max} = [K_{\max,1}, K_{\max,2}, \ldots, K_{\max,N}]^{\mathrm{T}}$ and $K_{\min} = [K_{\min,1}, K_{\min,2}, \ldots, K_{\min,N}]^{\mathrm{T}}$. The curvature of the perturbed path then need to satisfy the linear inequality constraint $K_{\min} - K \leq CX \leq K_{\max} - K$.

D. Bounds on the Variation and Collision Avoidance

Certain approximations in the formulation of the quadratic programming problem impose limits on the allowable magnitude of variation: because the second order terms in equation (5) are neglected during the approximation of the curvature, it is required that the variation is small enough such that this approximation is valid. The small variation is also required by the approximation used in the path length constraint. On the other hand, the magnitude of the variation is also limited by the requirement of collision-avoidance, since a large variation of the path in a neighborhood of an obstacle may lead to a collision. In order to determine the bounds of the perturbation at the i^{th} node $(i = 2, 3, \ldots, N-$

$$
\begin{aligned}
D_i &= \|\tilde{\mathbf{r}}_{i+1} - \tilde{\mathbf{r}}_i\| - \|\mathbf{r_{i+1}} - \mathbf{r_i}\| \\
&= - \|\mathbf{r}_{i+1} - \mathbf{r}_i\| + \sqrt{\|\mathbf{r}_{i+1} - \mathbf{r}_i\|^2 + \|\delta_{i+1}\mathbf{n}_{i+1} - \delta_i \mathbf{n}_i\|^2 + 2\left\langle \delta_{i+1}\mathbf{n}_{i+1} - \delta_i\mathbf{n}_i, \mathbf{r}_{i+1} - \mathbf{r}_i\right\rangle} \\
&= \frac{1}{\|\mathbf{r}_{i+1} - \mathbf{r}_i\|} \frac{\|\delta_{i+1}\mathbf{n}_{i+1} - \delta_i\mathbf{n}_i\|^2 + 2\left\langle \delta_{i+1}\mathbf{n}_{i+1} - \delta_i\mathbf{n}_i, \mathbf{r}_{i+1} - \mathbf{r}_i\right\rangle}{1 + \sqrt{1 + \frac{\|\delta_{i+1}\mathbf{n}_{i+1} - \delta_i\mathbf{n}_i\|^2}{\|\mathbf{r}_{i+1} - \mathbf{r}_i\|^2} + 2\left\langle \frac{\delta_{i+1}\mathbf{n}_{i+1} - \delta_i\mathbf{n}_i}{\|\mathbf{r}_{i+1} - \mathbf{r}_i\|}, \frac{\mathbf{r}_{i+1} - \mathbf{r}_i}{\|\mathbf{r}_{i+1} - \mathbf{r}_i\|}\right\rangle}}.
\end{aligned} \tag{10}
$$

$$
G = \begin{bmatrix}
-\langle \mathbf{r}_2 - \mathbf{r}_1, \mathbf{n}_1\rangle & \langle \mathbf{r}_2 - \mathbf{r}_1, \mathbf{n}_2\rangle & & & 0 \\
& -\langle \mathbf{r}_3 - \mathbf{r}_2, \mathbf{n}_2\rangle & \langle \mathbf{r}_3 - \mathbf{r}_2, \mathbf{n}_3\rangle & & \\
& & \ddots & \ddots & \\
0 & & & -\langle \mathbf{r}_N - \mathbf{r}_{N-1}, \mathbf{n}_{N-1}\rangle & \langle \mathbf{r}_N - \mathbf{r}_{N-1}, \mathbf{n}_N\rangle
\end{bmatrix}. \tag{11}
$$

1), we consider the path segment between the $i-1^{\text{th}}$ and i^{th} nodes, and the segment between the i^{th} and $i+1^{\text{th}}$ nodes, respectively. For the former segment, initially choose $\delta_i = \delta_{i-1} = \delta_{\max}$, where $\delta_{\max}$ is a predetermined small positive number. If this segment is still collision-free after the variation, then let $Y_{u,i} = \delta_{\max}$, otherwise decrease δ_i while keeping $\delta_i = \delta_{i-1}$ until the perturbed segment is collision-free, and let $Y_{u,i} = \delta_i$. Similarly, the variation lower bound $Y_{l,i}$ of the same segment is determined by initially choosing $Y_{l,i} = -\delta_{\max} = \delta_{i-1} = \delta_i$. If collision occurs, we gradually increase δ_i while keeping $\delta_{i-1} = \delta_i$ until the perturbed path is collision-free, and let $Y_{l,i} = \delta_i$. In this way, we also find the variation lower and upper bounds of the path segment between the i^{th} and $i+1^{\text{th}}$ nodes, which are given by $Z_{l,i}$ and $Z_{u,i}$, respectively. Define $X_{\max,i} = \min\{Y_{u,i}, Z_{u,i}\}$, $X_{\min,i} = \max\{Y_{l,i}, Z_{l,i}\}$, $X_{\max} = [X_{\max,1}, X_{\max,2}, \ldots, X_{\max,N}]^{\mathrm{T}}$ and $X_{\min} = [X_{\min,1}, X_{\min,2}, \ldots, X_{\min,N}]^{\mathrm{T}}$. Note that because the path is required to pass through the start and target positions, the variation δ must be zero at these two points, which can be achieved by setting the bounds as $X_{\min,1} = X_{\max,1} = 0$, $X_{\min,N} = X_{\max,N} = 0$. Then the perturbed path would be collision-free as long as the variation X satisfies $X_{\min} \leq X \leq X_{\max}$. Collision is checked at each node and a certain number of interpolating points in each segment of path, as shown in Fig. 1. Because collisions are not checked everywhere along the path, it is possible that the perturbed path slightly intrudes some obstacle, but this can be avoided by slightly expanding the boundary of the obstacles when performing the collision checking.

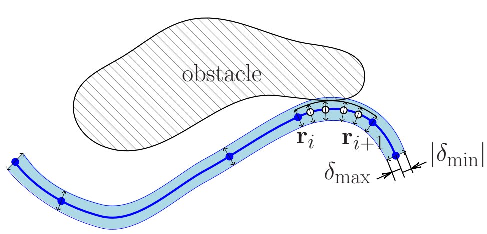

Fig. 1. Bounds of variation.

E. Initial and Final Condition

If no constraint exists on the tangent direction of the path at the start and target points, then the constraints at those two points are similar to hinge joints, i.e., the displacement δ_2 or δ_{N-1} does not alter the path curvature at node 1 or N, hence $\chi_{12} = \chi_{N-1,N} = 0$. On the contrary, if the tangent direction of the path is fixed at the boundary with heading angles $\psi_1, \psi_N \in \mathbb{R}$, then the tangent vectors at those two points are

$$
\mathbf{t}_1 = [\cos\psi_1, \sin\psi_1], \qquad \mathbf{t}_N = [\cos\psi_N, \sin\psi_N],
$$

and the corresponding normal vectors are $\mathbf{n}_1 = \mathbf{A}\mathbf{t}_1$ and $\mathbf{n}_N = \mathbf{A}\mathbf{t}_N$, respectively. This is analogous to the fixed end constraint for a beam. The tangent directional constraints on δ_2 and δ_{N-1} are

$$
\begin{aligned}
\langle \delta_2\mathbf{n}_2 + \mathbf{r}_2 - \mathbf{r}_1, \mathbf{n}_1\rangle &= 0, \\
\langle \delta_{N-1}\mathbf{n}_{N-1} + \mathbf{r}_{N-1} - \mathbf{r}_N, \mathbf{n}_N\rangle &= 0,
\end{aligned}
$$

which uniquely determine the values of δ_1 and δ_N.

In order to compute the curvature of the path at the start and end points when the tangent directions are fixed, we introduce two extra points using finite differences as follows: $\mathbf{r}_0 = \mathbf{r}_1 - (s_2 - s_1)\,\mathbf{t}_1$ and $\mathbf{r}_{N+1} = \mathbf{r}_N + (s_N - s_{N-1})\,\mathbf{t}_N$. Then $\chi_{1,2}$ and $\chi_{N,N-1}$ can be computed using equation (6).

F. Connection to Beam Theory

Consider a classical beam subject to pure bending. The bending moment and the local curvature satisfy:

$$
\kappa(s) = \frac{M(s)}{EI(s)},
$$

where $\kappa(s)$ is the local curvature of the neutral surface of the beam, $M(s)$ is the bending moment at the cross section at s, and $I(s)$ is the second moment of area of the cross section about its neutral surface, and E is the Young's modulus of the beam material. The product EI is often referred to as the *flexural rigidity* or the *bending stiffness* of the beam.

The total strain energy U of the bending beam can be written as:

$$
U = \int_0^{s_f} \frac{M^2(s)}{2EI(s)}\mathrm{d}s = \frac{1}{2}\int_0^{s_f} EI(s)\kappa^2(s)\mathrm{d}s,
$$

which is exactly the square of the weighted L_2 norm of the curvature function. Hence, the result of the quadratic program essentially corresponds to a minimum bending energy configuration in a neighborhood of the original path. It is also observed that the weight function $w(s)$ in (8) corresponds to the flexural rigidity $EI(s)$.

IV. Path Smoothing Algorithm

A. Discrete Evolution and the Path Smoothing Algorithm

Consider a family of smooth paths $\mathcal{P}(s,j)$, where s is the path coordinate parameterizing the path and j is the index parameterizing the family. The path evolves among the family $\mathcal{P}(s,j)$ at the representative nodes according to the evolution equation

$$\begin{aligned} \mathcal{P}\left(s_i, j+1\right) &= \mathcal{P}\left(s_i, j\right) + X_i^* \mathbf{n}\left(\mathbf{s}_i, j\right), \\ \mathcal{P}\left(s, 0\right) &= \mathcal{P}^{(0)}\left(s\right), \end{aligned} \tag{14}$$

where X_i^* is the i^{th} component of the solution to the quadratic program with initial path $\mathcal{P}(s,j)$. For each j, the path is given by the piecewise fifth order Bézier curve interpolation between the characteristic nodes as described in Section II-A.

The proposed path smoothing algorithm is designed based on the evolution equation (14):

1) Let j be the count of iterations, starting from $j=1$,
2) Discretize the path with N nodes, say, $s_1=0, s_2, s_3, \ldots, s_N = s_f$.
3) Determine the bounds of variation, and solve the quadratic programming problem. Interpolate the result with a piecewise fifth order Bézier curve to generate the new path,
4) Compute the difference between the new and the old path by

$$\xi_j = \int_0^{s_f} \left\| \mathcal{P}(s,j) - \mathcal{P}(s,j-1) \right\|^2 \mathrm{d}s.$$

Stop the iteration if ξ_j is smaller than some predetermined threshold, or if j reaches the maximum number of iterations. Otherwise increase j by one and go to Step 2).

B. Reconciling Conflicts Between Variation Bounds and Constraints

Due to the bounds on the allowed variation, the domain of optimization in each step of the proposed algorithm is relatively small, and sometimes the variation bounds are in conflict with the boundary conditions and curvature constraints, in the sense that the prescribed boundary conditions and curvature constraints cannot be satisfied by any variation within the bounds during a single iteration.

To resolve such conflicts, the curvature constraints and the boundary conditions are enforced progressively during the iterations when necessary, rather than being enforced explicitly in each iteration. For example, suppose the path needs to satisfy the curvature constraints $K_{\min} \le K \le K_{\max}$. Then for each iteration j, the following relaxed curvature bounds are used

$$K_{\min} - c_1 e^{-\beta_1 j} \le K_j \le K_{\max} + c_2 e^{-\beta_2 j},$$

where $c_1, c_2, \beta_1, \beta_2 > 0$. It is seen that the left and right hand sides in the above inequalities initially provide relaxed curvature bounds when $j=0$, yet approach the prescribed bounds $K_{\min}$ and $K_{\max}$ asymptotically as j increases. A similar technique is applied for the enforcement of the tangent directional constraints at the start and end points. Specifically, if the initial and final tangent directional constraint can not be satisfied in one iteration, then the following constraints are used:

$$\begin{aligned} \left| \left\langle \delta_2 \mathbf{n}_2 + \mathbf{r}_2 - \mathbf{r}_1, \mathbf{n}_1 \right\rangle \right| &< c_3 e^{-\beta_3 j}, \\ \left| \left\langle \delta_{N-1} \mathbf{n}_{N-1} + \mathbf{r}_{N-1} - \mathbf{r}_N, \mathbf{n}_N \right\rangle \right| &< c_4 e^{-\beta_4 j}, \end{aligned}$$

with $c_3, c_4, \beta_3, \beta_4 > 0$.

V. Numerical Examples

A. Fixed Length Path Smoothing with Collision Avoidance

We consider an example in which a UAV flies from point A to point B. The obstacles are represented by the gray regions in Fig. 2. The original path is generated using the A^* algorithm for minimum length and smoothed using a fourth order spline curve. This initial path is shown in blue in Fig. 2. The initial and final tangents of the path are fixed. The length of the path is fixed during the path smoothing process. The path smoothing algorithm finishes in 0.39 sec after 15 iterations. The curvature profiles for the original and smoothed paths are compared in Fig. 3. The L_2 norm of the curvature function with respect to the path coordinate decreased by 73% after smoothing, while the L_∞ norm was reduced by 70%. In Fig. 4, the optimal speed profiles of the original and smoothed paths are compared. It is clear that the smoothed path provides a shorter travel time. The optimal speed profiles are computed using the time-optimal parameterization method introduced in [15] with free final speed at point B.

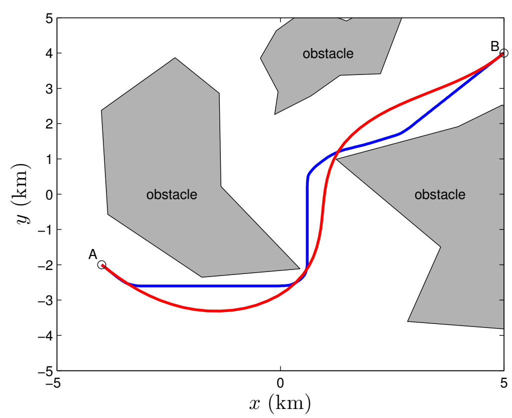

Fig. 2. Path smoothing in the presence of obstacles.

B. Path Smoothing with Localized Curvature Bounds

In this example, a ground vehicle starts from point A at one side of a frozen river, avoids the obstacle, crosses the river while passing through point B, and finally reaches the target at point C at the other side of the river. Due to the small coefficient of friction of the icy river surface, it is

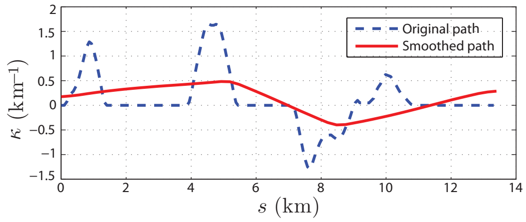

Fig. 3. Curvature profile comparison.

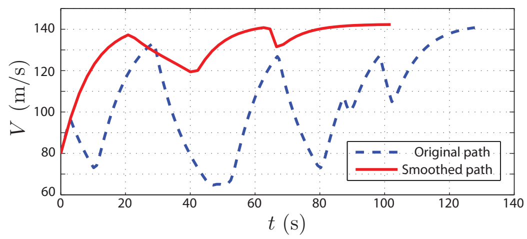

Fig. 4. Optimal speed profile.

required that the segment of the path on the ice surface must have zero curvature (no turning allowed). The initial path consists of three line segments. During the smoothing process, the constraint on the total length of the path is relaxed. Furthermore, there exists no directional constraint at the start and the end of the path. In order to ensure that the path passes through point B, a node is added to the path at point B, and the variation at this node is set to be zero during the smoothing process. The result from smoothing is shown in Fig. 5. It is clear that the ground vehicle does not need to perform any turning maneuver on the ice surface.

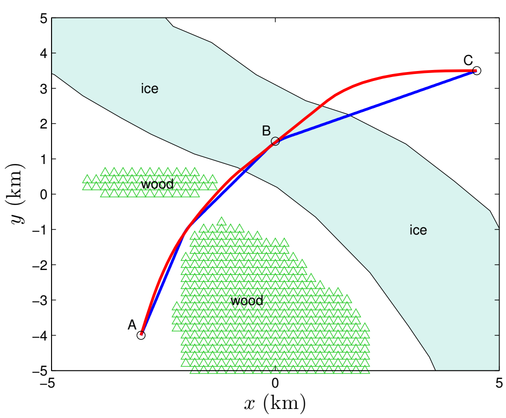

Fig. 5. Smoothed path with local curvature constraint.

VI. Conclusions

In this paper, we considered the problem of two dimensional path smoothing with obstacles and local curvature constraints. The problem is formulated as a quadratic program, which minimizes the weighted L_2 norm of the curvature along the path. By incorporating additional linear constraints into the quadratic programming problem, extra constraints on the tangent of the path, path length, and local curvature can also be accommodated. The proposed path smoothing algorithm has been applied to several examples, and its efficiency and effectiveness have been validated. Future work will focus on extending the current path smoothing method to the three-dimensional space.

Acknowledgement: This work has been supported by NASA (award no. NNX08AB94A).

References

[1] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of Mathematics*, vol. 79, no. 3, pp. 497–516, 1957.

[2] R. V. Cowlagi and P. Tsiotras, "Shortest distance problems in graphs using history-dependent transition costs with application to kinodynamic path planning," in *American Control Conference*, (St. Louis, MO), pp. 414–419, 2000.

[3] A. Scheuer and T. Fraichard, "Continuous-curvature path planning for car-like vehicles," in *IEEE International Conference on Intelligent Robots and Systems*, pp. 997–1003, 1997.

[4] T. Fraichard and A. Scheuer, "From reeds and shepps to continuous-curvature paths," *IEEE Transaction on Robotics*, vol. 20, no. 6, pp. 1025–1035, 2004.

[5] E. Bakolas and P. Tsiotras, "On the generation of nearly optimal, planar paths of bounded curvature and bounded curvature gradient," in *Proceedings of the 2009 conference on American Control Conference*, (St. Louis, MO), pp. 385–390, 2009.

[6] A. Piazzi, C. G. L. Bianco, and M. Romano, "η^3-splines for the smooth path generation of wheeled mobile robots," *IEEE Transaction on Robotics*, vol. 23, no. 5, pp. 1089–1095, 2007.

[7] K. Yang and S. Sukkarieh, "An analytical continuous-curvature path-smoothing algorithm," *IEEE Transactions on Robotics*, vol. 26, pp. 561–568, Jun 2010.

[8] E. Bakolas and P. Tsiotras, "On-line, kinodynamic trajectory generation through rectangular channels using path and motion primitives," in *47th IEEE Conference on Decision and Control*, (Cancun, Mexico), pp. 3725–3730, 2008.

[9] S. Bereg and D. Kirkpatrick, "Curvature-bounded traversals of narrow corridors," in *SCG '05: Proceedings of the twenty-first annual symposium on Computational geometry*, (Pisa, Italy), pp. 278–287, ACM, 2005.

[10] D. Jung and P. Tsiotras, "On-line path generation for small unmanned aerial vehicles using b-spline path templates," in *AIAA Guidance, Navigation, and Control Conference*, AIAA-2008-7135, (Honolulu, HI), 2008.

[11] T. Maekawa, T. Noda, S. Tamura, T. Ozaki, and K. Machida, "Curvature continuous path generation for autonomous vehicle using b-spline curves," *Computer-Aided Design*, vol. 42, no. 4, pp. 350–359, 2010.

[12] R. F. Gilimyanov, A. V. Pesterev, and L. B. Rapoport, "Smoothing curvature of trajectories constructed by noisy measurements in path planning problems for wheeled robots," *Journal of Computer and Systems Sciences International*, vol. 47, no. 5, pp. 812–819, 2008.

[13] D. Salomon, *Curves and Surfaces for Computer Graphics*. Springer-Verlag, 2006.

[14] J. A. Sethian, "Curavature and the evolution of fronts," *Communications in Mathematical Physics*, vol. 101, no. 4, pp. 487–499, 1985.

[15] Y. Zhao and P. Tsiotras, "Time-optimal parameterization of geometric path for fixed-wing aircraft," in *Infotech at Aerospace Conference*, AIAA-2010-3352, (Atlanta, GA), 2010.