JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 33, No. 2, March–April 2010

Path Planning and State Estimation for Unmanned Aerial Vehicles in Hostile Environments

Ran Dai* and John E. Cochran Jr.†
Auburn University, Auburn, Alabama 36849

DOI: 10.2514/1.46323

I. Introduction

UNMANNED aerial vehicles (UAVs) are now widely used in antiterrorism activities and intelligence gathering to enhance mission performance and maximize safety. The susceptibility of these UAVs in hostile environments raises requirements for flight path planning. Path planning strategies in hostile environments are normally composed of two phases [1,2]. The first phase is a Voronoi graph search, which will generate polygonal graphs and will optimize a safety performance index. The second is to use the virtual forces emanated from the virtual field of each surveillance radar site to refine the generated Voronoi graphs. These virtual forces provide information that can be used to reduce the vertices of the Voronoi polygonal and greatly improve the UAV performance. But the curvature continuity of the refined graphs, which plays an important role in the stability of the UAVs' turning maneuvers, often does not meet the requirements for a continuously flyable path.

Many kinds of curves have been studied and designed for UAVs to accomplish their mission [3–5]. Dubins curves, first applied in robotics path planning, are curves along which the UAV can move forward. This kind of circle–line–circle curve has a jump discontinuity in curvature at the connection points between the circle and the line that will cause a robot to stop at these connection points when traveling through the whole path. Other curves, such as the Reeds–Shepp curves [6,7], also have curvature discontinuities at their joint points. An alternative choice, the composite clothoid–line–clothoid curves, can be well designed with curvature being zero at the joint points to eliminate the discontinuities. This allows one to generate a continuous-curvature path by using different kinds of simply shaped curves, although, under most circumstances, we are expecting more flexibility in the curve shape that will allow more space for change. Shanmugavel et al. [3–5]. proposed quintic Pythagorean hodograph (PH) curves for a flyable path, with ten parameters representing each curve. The PH curves are flexible in design and their curvatures are expressed in continuous polynomials. The parameter calculation of a PH curve is an iterative process in order to satisfy different constraints. Such kinds of curves can be further simplified with fewer parameters and a more efficient optimization algorithm. All the methods discussed leave room for improvement in the area of continuous-curvature path planning.

The Cornu sprial (CS) [8,9], also known as a clothoid or Euler's spiral, has wide application in highway and railroad construction, since it can be used to design gradual and smooth transitions in highway entrances or exits. Kelly and Nagy [10] used a parametric CS model to generate real-time nonholonomic trajectories for robotics to minimize the terminal posture error. Here, we consider this CS model for use in UAV path planning and investigate how this parametric CS curve works under different constraints.

To generate a flyable and safe path with given starting and ending points for UAVs passing through areas covered, at least partially, by several radar sites, the path constraints considered here include 1) minimum accumulative exposure to all radar sites; 2) continuous curvature throughout its length, which will ensure a flyable path' 3) maximum curvature corresponding to the maximum achievable lateral turn rate; and 4) initial and final boundary constraints. Unlike other path planning problems, including that of moving objects finding the final path, which normally result in motion planning or trajectory planning with system dynamics [11], the path considered here is in a static object environment without dynamic constraints. The work in this paper is based on the developed Voronoi graph; it refines the graph by proposing a generalized CS curve along with a simplified parameter-identification procedure.

Most papers on the topic of path planning do not include the information about dynamic state variables of UAVs flying along the planned path. For control purposes, it is beneficial to estimate these state variables to construct complete information of the flight. Kalman filtering [12] has been widely used as an efficient tool in optimal filtering and prediction, especially in the field of state estimation of UAVs performing designated missions [13–17]. For example, Grillo and Vitrano [13] used an extended Kalman filter (EKF) to estimate the state variables and wind velocity for a nonlinear UAV model with Global Positioning System (GPS) measurements. Abdelkrim et al. [14] used an EKF and an $H\infty$filter to estimate the localization of UAVs for which the position, velocity, and attitude are measured by an inertial navigation system. Campbell and Ousingsawat [15] used two different estimators to provide online state and parameter estimation for path planning in uncertain environments. The state estimations in these studies have a commonality, because the UAVs in both cases have sensors or other instruments to provide useful measurement information. If all of the Voronoi points on the initial path are fixed and expected to be followed as closely as possible, they can be assumed as measurement points. The generated CS curve is then treated as the reference solution so that the state variables can be estimated by the EKF.

The following sections present the procedure for the path-information construction in three parts. The first part is the initial rough path of the Voronoi graph and the dynamic programming search algorithm. In the second part, CS curve expression and properties are introduced and different constraints and their mathematical expression are explained. This is followed by the systematic-solution nonlinear programming (NLP) solver. In the last part, the state variables are estimated based on the generated Voronoi points and the refined reference path generated in the first two parts. Simulation results are presented in each part separately.

II. Initial Path Generation

There are many developed graph-search methods that maybe used to generate the initial path that will maximize the safety of the UAVs. Two prominent ones are heuristic search [18] and probability map search [19]. The Voronoi graph search is generally used to find the rough initial path in the form of connected edges. The authors of some papers have stated the Voronoi method and the following

Received 13 July 2009; revision received 21 October 2009; accepted for publication 6 November 2009.

*Graduate Research Assistant, Department of Aerospace Engineering; currently Engineer, Dynamic Research, Inc., Torrance, CA 90501; aviator_dai@hotmail.com.

†Professor and Head, Department of Aerospace Engineering; jcochran@eng.auburn.edu. Fellow AIAA.

dynamic programming search algorithm in detail [1,2]. Here, we briefly describe them.

A. Voronoi Graph Search

A Voronoi diagram will decompose a space defined by random scattered points into separated cells in such a manner that each cell will include one point that is closer to all elements in this cell than any other points. The graph is constructed by using Delaunay triangulation and its dual-graph Voronoi diagrams. The procedure of this cell decomposition starts with a priori knowledge of the location and of number of the scattered points assumed as radar sites here. As illustrated in Fig. 1, the star points are predefined radar sites. The circle points named as Voronoi points are the centers of the circumcircle of the triangles formed by three of the Voronoi sites without including any other sites in this circumcircle. The formation of these triangles is called Delaunay triangulation. All points on edges connected by the Voronoi points have equivalent distance to two adjacent radar sites. By connecting all of the edges together, polygons are formed and these polygons construct the Voronoi graph. The UAV starting point and ending point are also illustrated as small triangles in the plot.

If UAVs are assumed to go through the radar coverage area along the edges in the formed Voronoi graph, the probability of exposure of each edge to radar sites can be determined using the distance from the points on the edge to the radar sites. Suppose each radar site emits a radiation signal with equivalent strength in all directions: the threats of exposure to this radar site is proportional to the inverse of distance to the fourth power. The whole edge threat cost is the accumulative threat calculation from the beginning of the edge to the end. Mathematically, it is expressed as [1]

$$\int_{x_1,y_1}^{x_2,y_2} \frac{\sqrt{\mathrm{d}x^2+\mathrm{d}y^2}}{[(p_x-x)^2+(p_y-y)^2]^2} \tag{1}$$

where the pair (x_1, y_1) and (x_2, y_2) are the coordinates of starting and ending points on the edge and (p_x, p_y) is the radar site coordinate. Since the edge position can be expressed by the starting and ending points as

$$y = y_1 + \alpha(x - x_1), \qquad \alpha = \frac{y_2 - y_1}{x_2 - x_1} \tag{2}$$

thus Eq. (1) can be simplified as

$$\int_{x_1}^{x_2} \frac{\sqrt{1+\alpha^2}}{[(p_x-x)^2+(p_y-y_1-\alpha(x-x_1))^2]^2}\,\mathrm{d}x \tag{3}$$

By assigning a threat cost to each edge, an optimal path with lowest probability of detection when passing through the radar area can be constructed. Determining this optimal path requires the calculation of the threat cost for all of the available paths and then selects the one with lowest cost after comparison. The dynamic programming method is applied here to solve these kinds of optimal-path selection problem.

B. Dynamic Programming

Dynamic programming is a branch of operational research. The dynamic programming procedure is to break a problem into many layers of subproblems and then solve each subproblem individually to achieve the overall optimality. At each layer, the optimal solution is saved so that when we come to this layer again, we can retrieve the saved result to avoid repeated computing. In the Voronoi graph of Fig. 1, the cost of each edge is determined and the objective is to find a path from the starting point to the ending point with lowest total cost. Counting from the ending point backward, at each Voronoi point, the lowest cost from that point to the ending is computed and saved. For example, Voronoi point 5 is set with a cost of zero and the cost from Voronoi point 4 to 5 is 8.773×10^{-5}. This process is repeated for all Voronoi points. When a point is reached that has a cost value already determined, the algorithm takes the predefined optimal path from that point to the ending without searching them again. So when counting the cost at point 2, it will take the determined route from point 3 to the ending point. The safest path is found and labeled with dark lines in Fig. 2.

III. Reference Path Generation

Obviously, the initial path generated in Fig. 2 is coarse and not flyable by UAVs. The final flight path is expected to be smooth enough everywhere so that the UAVs will not have sharp turns on the whole path. At the same time, we also expect the path to be as close as possible to the original Voronoi graph to maximize the safety factor. Since the curvature is a key factor in determining the smoothness of a curve, the problem turns out to be a curve-fitting problem with curvature constraints and some other constraints. The CS with

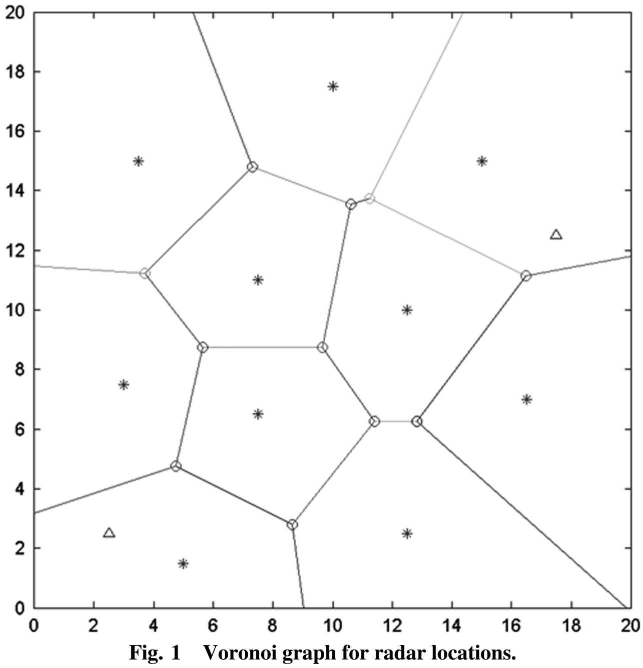

Fig. 1 Voronoi graph for radar locations.

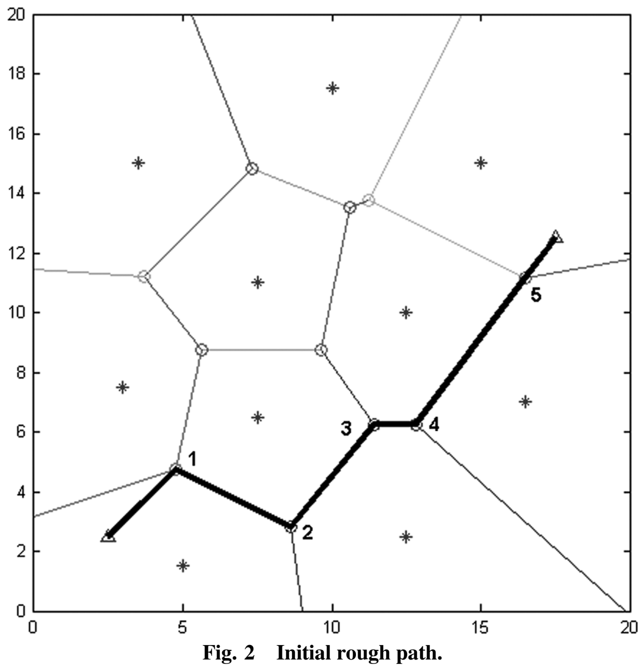

Fig. 2 Initial rough path.

advantages introduced in the following is selected as the reference path to fit the Voronoi points and satisfy the specified constraints.

A. Cornu Spiral

The CS is a well-known curve for which the curvature is defined as a polynomial function of its arc length s as

$$\kappa(s) = \sum_{i=0}^{n} \alpha_i s^i \tag{4}$$

and the curvature is also the derivative of the curve angle θ with respect to the arc length:

$$\kappa(s) = \frac{\mathrm{d}\theta}{\mathrm{d}s} \tag{5}$$

The positions of the points on this curve in Cartesian coordinates are calculated by the Fresnel integrals:

$$x(s) = \int_0^s \cos(\theta(s))\,\mathrm{d}s \qquad y(s) = \int_0^s \sin(\theta(s))\,\mathrm{d}s \tag{6}$$

One advantage of using this curve in solving the path planning problem is the simplicity of calculation of its arc length:

$$s(t) = \int_0^t \sqrt{\dot{x}(t)^2 + \dot{y}(t)^2}\,\mathrm{d}t = \int_0^t 1\,\mathrm{d}t = t \tag{7}$$

The above expressions are general, i.e., they apply to all forms of CS curves, including special cases such as a straight line ($n = 0$ and $\alpha_0 = 0$), circle ($n = 1$ and $\alpha_1 = 0$), and a unit CS ($n = 1, \alpha_0 = 0$, and $\alpha_1 = 1$). In a unit CS, the curvature equals the arc length and will therefore increase with the length. So the longer the curve, the greater the curvature: hence, the origin of the term *spiral*. The coefficients α_i affect the increase rate of the CS curve. In Fig. 3, all the CS curve curvature polynomials are order $n = 1$ with $\alpha_0 = 0$, with α_1 varied from 0.1 to 5. It is obvious that when transversing the same arc length, the larger the value of α_1, the more the curvature increases. In addition, the order in the curvature expression also affects its shape. Figure 4 shows a series of CS curves with different order numbers in the curvature polynomials. The coefficients of the highest order are all for $\alpha_n = 1$, with the others set as zero. It can be seen that curves with the higher-order polynomials “curl” faster than others.

Combing the two types of effects that determine the shape of the CSs, we can produce an expression of the curvature polynomials to define any CSs possessing as many degrees of freedom parameters as necessary to meet the required constraints. In the problem formulation stated above, the primary constraint for such curves is to meet the initial and final conditions that are classified as equality constraints. If the initial conditions $P_0 = (x_0, y_0)$ are incorporated as parameters in the CS curve, at least two additional parameters are required to meet the final boundary constraints. To have enough freedom to satisfy other inequality constraints and coordinate all the parameters to achieve the optimization purpose, one more parameter is required. Although the inclusion of more parameters in the CS expression will increase the flexibility of the curve, it will also increase the calculation burden. As a compromise between the optimality and computation time, five parameters are used in one CS curve. Three of them (a, b, and c) form the cubic polynomials of the curve-angle expression, and θ_0 is the initial heading angle. The last one is the final arc length s_f. These parameters enter the problem formulation as follows:

$$\theta(s) = \theta_0 + as + bs^2 + cs^3 \qquad \kappa(s) = \dot{\theta}(s) = a + 2bs + 3cs^2$$
$$x(s) = x_0 + \int_0^{s_f} \cos(\theta_0 + as + bs^2 + cs^3)\,\mathrm{d}s$$
$$y(s) = y_0 + \int_0^{s_f} \sin(\theta_0 + as + bs^2 + cs^3)\,\mathrm{d}s \tag{8}$$

The path planning problem has then been cast as a parameter optimization problem with equality and inequality constraints.

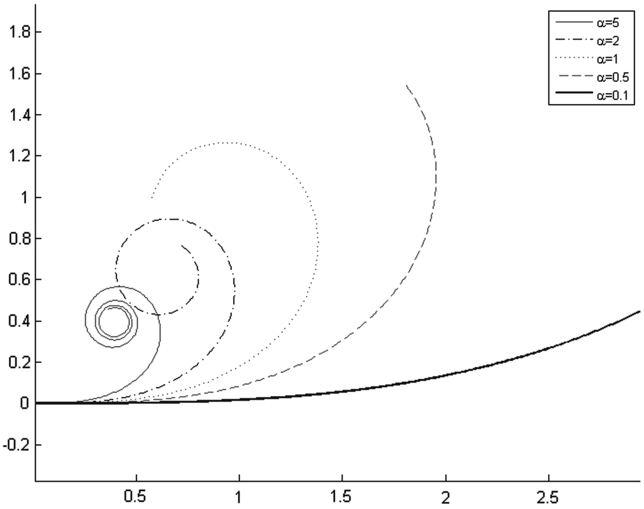

Fig. 3 Cs curves of order $n = 1$ and $\alpha_0 = 0$ with coefficients α_1 changes from 0.1 to 5.

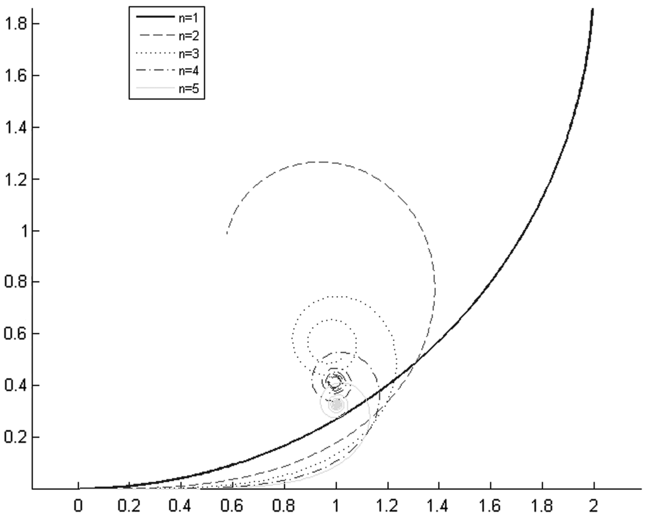

Fig. 4 Cs curves of $\alpha_n = 1$ and other coefficients set as zero, order n changes from 1 to 5.

B. Continuous-Curvature Constraints

Continuous curvature is an important factor in designing a flyable path, because the vehicle’s path must be physically continuous and the rate of change of curvature with arc length is related to the directional command, a key input in maneuvering UAVs. Discontinuities in the curvature will cause jumps in the steering angle input. Such discontinuous input is not achievable. The curvature of a planar curve defined by $x(t)$ and $y(t)$ is

$$\kappa(t) = \frac{\dot{x}(t)\ddot{y}(t) - \dot{y}(t)\ddot{x}(t)}{(\dot{x}(t)^2 + \dot{y}(t)^2)^{3/2}} \tag{9}$$

The CS curve specified in Eq. (8) with three parameters representing its polynomials can be expressed as

$$\kappa(s) = a + 2bs + 3cs^2 \tag{10}$$

Physically, the first and second derivatives of curvature correspond to the lateral velocity and acceleration of a point (the vehicle) that is moving along the curve. Therefore, the curvature must be at least twice differentiable to meet the continuous velocity and acceleration requirement. The CS curve with curvature expression in (10) has second-order polynomials, is twice differentiable, and can satisfy the continuity constraints.

C. Maximum-Curvature Constraints

In CS expression of Eq. (8), the curvature is the only shape-determining factor. By using the maximum-curvature constraint, the kinematic acceleration of UAVs can be limited. If a proper tangential velocity is maintained, then the path designed can actually be traversed. Since the curvature is a parabolic function of the arc length, its maximum value points can only occur at one of three points: the initial point, the summit or bottom point, or the final point. Mathematically, the curvature values at those three points are

$$\kappa_0 = a \qquad \kappa_m = a - \frac{b^2}{3c} \qquad \kappa_f = a + 2bs_f + 3cs_f^2 \tag{11}$$

If the curvature does not exceed the maximum value at any of these three points, then the curvature throughout the whole length will be suitably constrained.

D. Voronoi Points Projection Constraints

To fit the Voronoi points, it is necessary to minimize the distances from the Voronoi points to their projection points on the CS curve. Before the parameters of the CS curve are defined, it is difficult to calculate the orthogonal distance, but it is obvious to see that the slope of the normal line for the projection point on the CS is the negative reciprocal of its tangent line slope. Assume there are n Voronoi points. The projection of Voronoi point i $(1, \ldots, n)$ on the CS curve will satisfy the following relationship:

$$\frac{p_{y_i} - y(s_i)}{p_{x_i} - x(s_i)} \cdot \tan(\theta(s_i)) = -1 \tag{12}$$

where (p_{x_i}, p_{y_i}) are the coordinates for the Voronoi point i, and s_i is the length of the CS from the starting to the corresponding projection point with coordinates of $(x(s_i), y(s_i))$. Instead of calculating the projection points by geometric theory, they can be found by putting the constraints on s_i into Eq. (12). Then the distance from Voronoi point i to its corresponding projection point is calculated as

$$d_i = \sqrt{(p_{x_i} - x(s_i))^2 + (p_{y_i} - y(s_i))^2} \tag{13}$$

E. Boundary-Condition Constraints

The initial and final conditions are the equality constraints on the system. The path designed by the CS curve starts from the original points, so the initial condition is satisfied by setting the initial position coordinates as parameters in the CS curve:

$$x(0) = x_0 \qquad y(0) = y_0 \tag{14}$$

The final point is determined by using a Runge–Kutta numerical integration method to satisfy the specified final boundary conditions.

At this point, all the constraints have been described. Some of them can be satisfied by the properties of the CS curve itself, and others are expressed in corresponding equalities and inequalities. So the path planning problem is now a parameter optimization problem in which the parameters of the CS curvature expressions are identified while minimizing the accumulative orthogonal displacements for all Voronoi points.

F. Nonlinear Programming Solver: SNOPT 6.2

The NLP solver used to solve the NLP problem of interest here is SNOPT [20], which is based on a sequential quadrature programming algorithm and has been used as efficient tool to solve nonlinear optimal control problems with constraints [21–24]. SNOPT can be used to solve problems such as the following:

Minimize a performance index $J(x)$, subject to constraints on individual state and/or control variables,

$$x_L < x < x_U \tag{15}$$

constraints defined by linear combinations of state and/or control variables,

$$b_L < Ax < b_U \tag{16}$$

and/or constraints defined by nonlinear functions of state and/or control variables:

$$c_L < c(x) < c_U \tag{17}$$

With the above in mind, we can transfer the path planning problem into a simple form.

Let the NLP performance index J be denoted by

$$J = \sum_{i=1}^{n} [(p_{x_i} - x(s_i))^2 + (p_{y_i} - y(s_i))^2] \tag{18}$$

and the NLP variables of the CS curve as

$$x = \lfloor \theta_0, a, b, c, s_1, \ldots, s_i, \ldots, s_n, s_f \rfloor \tag{19}$$

where $i = 1, \ldots, n$, subject to the equality constraints, including the Voronoi points projection constraints and boundary conditions P_0 and P_f. These constraints will ensure that the curve-fitting will minimize the orthogonal displacement and all the coordinates on the CS curve will satisfy its differential function in the trigonometric form of Eq. (8). In addition, the inequality constraints to ensure the flyable path can be set as

$$\begin{gathered} -|\kappa_{\max}| \le a \le |\kappa_{\max}| \qquad -|\kappa_{\max}| \le a - \frac{b^2}{3c} \le |\kappa_{\max}| \\ -|\kappa_{\max}| \le a + 2bs_f + 3cs_f^2 \le |\kappa_{\max}| \quad \text{if } \left(s_0 \le -\frac{b}{3c} \le s_f \right) \end{gathered} \tag{20}$$

For the Voronoi sets in Fig. 2, a CS curve to fit those points is calculated under two conditions, one without maximum-curvature constraint and the other one with maximum-curvature constraint $|\kappa_{\max}| = 1/3$. Their path-simulation results are illustrated in Figs. 5 and 6, respectively. The corresponding curvature results with respect to curve length are illustrated in Figs. 7 and 8, respectively. From the two kinds of simulation results, the CS curve without curvature constraint fits the Voronoi points better than the one with curvature constraint by increasing the lateral kinematic acceleration to follow the path. The accumulative threats of the CS curve to the radar sites are 0.495 and 0.512 for curves without and with curvature constraint, respectively. Compared with the accumulative threat of the Voronoi

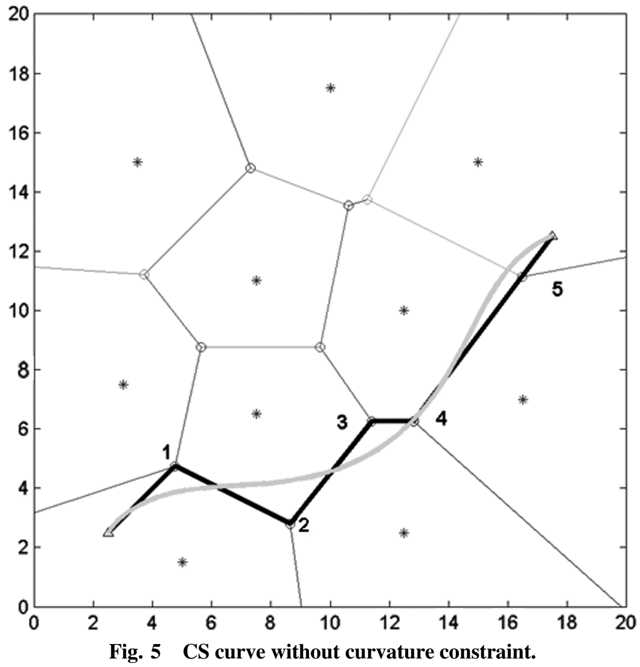

Fig. 5 CS curve without curvature constraint.

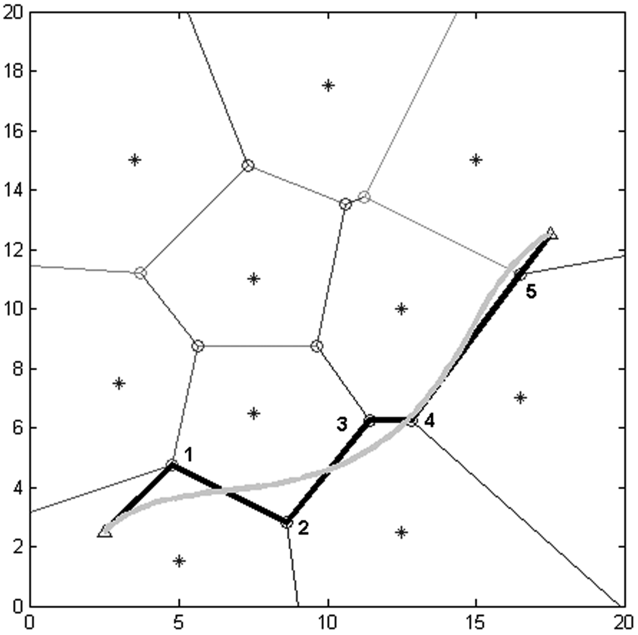

Fig. 6 CS curve with curvature constraint.

graph, 0.439, the CS curve achieved a flyable path with similar level of threat exposure to the radar sites.

IV. State Estimation

A. UAV Model

For simplicity of illustration, a two-dimensional UAV nonlinear model [25] in horizontal flight is used here, assuming all the adversarial radar sites are located in the same altitude so that the generated Voronoi graph and reference path are in the lateral plane. The three-degree-of-freedom UAV equations of motion are listed below. This model is based on the assumption that the UAV is a rigid body flying over a nonrotating and flat Earth:

$$\dot{X} = f(X,U) = \begin{bmatrix} \dot{x} = u\cos\psi - v\sin\psi \\ \dot{y} = u\sin\psi + v\cos\psi \\ \dot{u} = rv + F_x/m \\ \dot{v} = -ru + F_y/m \\ \dot{\psi} = r \\ \dot{r} = N/I_{zz} \end{bmatrix} \tag{21}$$

where (x,y) is the position, (u,v) are the lateral velocity components, ψ is the heading angle, r is the yaw rate, $[F_x, F_y, N]$ are the aerodynamic forces and moment achieved by the control actuators, m is the mass, and I_{zz} is moment of inertia about the z axis of the UAV body frame. This UAV model is built in the time domain, but the position and heading-angle variables of the reference path are built in the distance domain, and their derivatives with respect to distance are

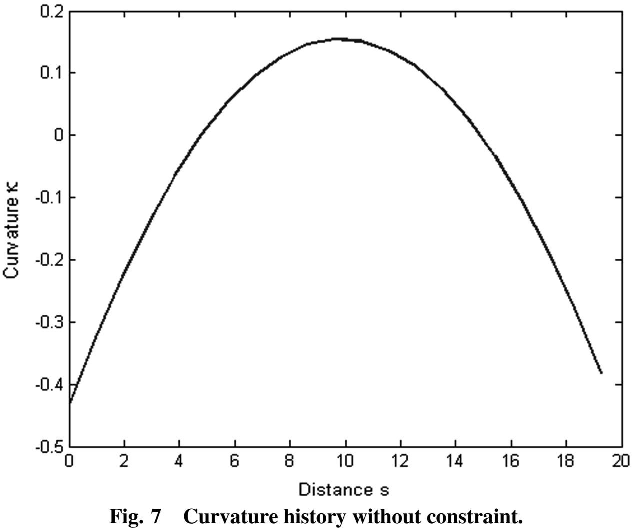

Fig. 7 Curvature history without constraint.

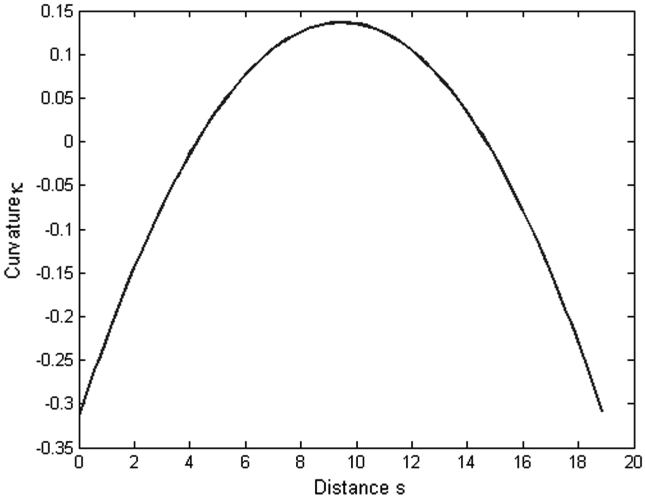

Fig. 8 Curvature history with constraint.

$$x'(s) = \cos(\theta(s)) \qquad y'(s) = \sin(\theta(s)) \qquad \theta'(s) = \kappa(s) \tag{22}$$

To estimate all variables in the time domain, it is necessary to change all the variables of the reference path from the distance domain to the time domain by calling

$$\mathrm{d}s = V(t)\,\mathrm{d}t \tag{23}$$

where $V(t)$ is the time history of the velocity magnitude. By substituting ds into Eq. (21), the equations of motion in the time domain are expressed as

$$\dot{x}(t) = V(t)\cos(\theta(t)) \qquad \dot{y}(t) = V(t)\sin(\theta(t))$$
$$\dot{\theta}(t) = V(t)\kappa(t) \tag{24}$$

By this transformation, all the state variables of the UAV model can be calculated according to the generated reference path, which will then provide a reference solution for linearization of the dynamic model in EKF.

B. Design of Extended Kalman Filter and Smoother and Simulation Results

An extended Kalman filter and smoother (EKFS) [12] can often make good estimates of the state variables when the measurements are under random disturbance. By solving the weighted-least-squares problem for the measurements, we can determine the most likely values of the state variables for continuous or discrete systems. The procedure of EKF consists of two parts: a prediction part to predict the current states from the last step and an updating part to add a correction term to the predicted states to refine them and get more accurate values. The optimal smoother is a backward-sweep procedure starting from the last time point back to the first. It will take advantage of the filtered estimates in the forward EKF and calculate the differences between them and the final smoothed values. The optimal smoothing is also a recursive computation to further refine the estimates.

Because of the precision limitation of the detection instrument, the adversarial radar sites location detected is under some degree of uncertainty, which causes the subsequent uncertainties of the Voronoi points searched in the initial graph. By implementation of the above-mentioned EKFS, the errors during the measurement can be reduced and the adjustments of the state variables are estimated and added to the nominal solution obtained in the EFKS algorithm. Since the UAV model is nonlinear, we need to derive a Jacobian matrix A by taking the derivatives of the nonlinear system equations with respect to the state variables. Its mathematical form is written as

$$A = \begin{bmatrix} 0 & 0 & \cos\psi & -\sin\psi & -u\sin\psi - v\cos\psi & 0 \\ 0 & 0 & \sin\psi & \cos\psi & u\cos\psi - v\sin\psi & 0 \\ 0 & 0 & 0 & r & 0 & v \\ 0 & 0 & -r & 0 & 0 & -u \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{25}$$

Similarly, a Jacobian matrix B derived by making the derivatives of the nonlinear system equations with respect to the control variables is

$$B = \begin{bmatrix} 0 & 0 & 1/m & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/m & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/I_{zz} \end{bmatrix}^T \tag{26}$$

According to the observation output, the observation matrix H is simply defined as

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{27}$$

These Jacobian matrices, estimated at the nominal state points X_n, are then used in the error-approximation function expressed as

$$\Delta\dot{X} = \dot{X} - \dot{X}_n = A\Delta X + B\Delta U + \text{H.O.T.} \tag{28}$$

where H.O.T. are the higher-order terms ignored in this equation.

Then the EKFS will find an optimized adjustment $\Delta\hat{X}$ to the nominal solution by going through its filter and smoother. Design of the EKFS also includes the selection of the covariance matrices of the disturbance and observation error, Q and R, respectively. If the measurement noise is carried out by the characteristics of the measurement instrument only, the elements of R can be determined from the stochastic properties of the measuring system. Noting that the initial and final positions are fixed, R elements at those points are set as extremely small numbers. The elements in Q and R directly affect the Kalman gain K: thereafter, the adjustments require considerable effort to find the best fit. In this problem, Q is a 1 by 1 identity matrix and R is set as 4 times a 2 by 2 identity matrix, except the starting and ending points, where R is 0.4 times a 2 by 2 identity matrix. We also assume that w and v are unrelated with zero mean values.

Simulations were carried out based on two kinds of reference paths generated above, one with and the other without maximum-curvature constraints. The speed of the reference paths was assumed to be constant cruise speed at maximum weight. Calculated measurements are the position of Voronoi points labeled in Fig. 1. The starting and ending points are fixed measurements with zero errors. The state-variable history of the reference solution and estimated solution are plotted in Figs. 9 and 10 for these two cases, respectively. By adding a covariance error to the reference solution, the estimated values further reduce the measurement errors.

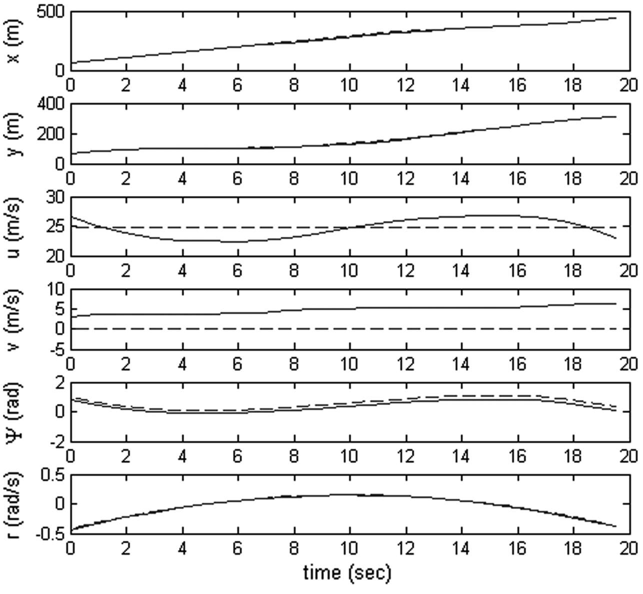

Fig. 9 States history for reference path without curvature constraint (dash line) and estimated path (solid line).

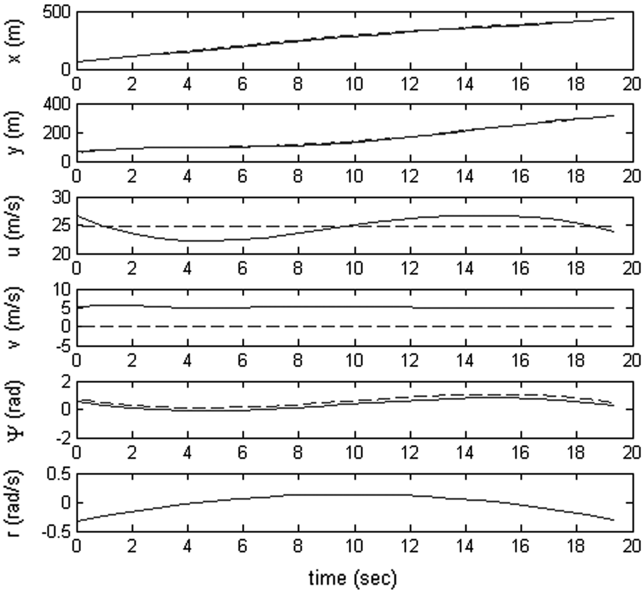

Fig. 10 States history for reference path with curvature constraint (dash line) and estimated path (solid line).

V. Conclusions

A type of UAV path with continuous curvature and constrained turning acceleration has been designed using the Voronoi graph and parameterized Cornu spirals (CSs). The state variables for such a planning path are estimated by the extended Kalman filter and smoother (EKFS). The planned paths are the suboptimal solution of the Voronoi diagram, which will maximize the safety factor. They refined the initial path by using the continuous-curvature characteristics of the CSs and trying to fit the Voronoi points with the minimum orthogonal distance. The curve-fitting algorithm includes all parameters of CSs as nonlinear programming variables in the nonlinear programming solver and gets the solution in one effort without tedious and iterative testing to adjust the coupled variables. This paper also considered disturbances in the detection and action of the hostile radar sites and estimated the state variables based on the reference path generated beforehand. Finally, the three parts of this paper (initial coarse graph, refined reference path, and state estimation) can be used to build up a complete information data set for both path planning and control purposes. Multiple UAV paths, which require cooperative control to increase simultaneous working performance, will be considered in future work.

References

[1] McLain, T. W., and Beard, R. W., “Trajectory Planning for Coordinated Rendezvous of Unmanned Air Vehicles,” AIAA Guidance, Navigation, and Control Conference, AIAA Paper 2000-4369, Aug. 2000.

[2] Bortoff, S. A., “Path Planning for UAVs,” *American Control Conference*, Inst. of Electrical and Electronics Engineers, Piscataway, NJ, June 2000, pp. 364–368.

[3] Shanmugavel, M., Tsourdos, A., Zbikowski, R., White, B. A., Rabbath, C. A., and Lechevin, N., “A Solution to Simultaneous Arrival of Multiple UAVs Using Pythagorean Hodograph Curves,” *American Control Conference*, Inst. of Electrical and Electronics Engineers, Piscataway, NJ, June 2006, pp. 2813–2818.

[4] Shanmugavel, M., Tsourdos, A., Zbikowski, R., and White, B. A., “3D Path Planning for Multiple UAVs Using Pythagorean Hodograph

Curves," AIAA Guidance, Navigation, and Control Conference, AIAA Paper 2007-6455, Aug. 2007.

[5] Shanmugavel, M., "Path Planning of Multiple Autonomous Vehicles," Ph.D. Dissertation, Department of Aerospace, Power and Sensors, Cranfield Univ., Cranfield, England, U.K., May 2007.

[6] Fraichard, T., Scheuer, A., and Desvigne, R., "From Reeds and Shepp's to Continuous-Curvature Paths," *Proceedings of the IEEE International Conference on Advanced Robotics*, Inst. of Electrical and Electronics Engineers, Piscataway, NJ, Oct. 1999, pp. 585–590.

[7] Scheuer, A., and Fraichard, T., "Continuous-Curvature Path Planning for Car-Like Vehicles," *Proceedings of the IEEE-RSJ International Conference on Intelligent Robots and Systems*, Vol. 2, Sept. 1997, pp. 997–1003.

[8] Stoer, J., "Curving Fitting with Clothoidal Splines," *Journal of Research of the National Bureau of Standards*, Vol. 87, No. 4, July–Aug. 1982, pp. 317–346.

[9] Davis, T. G., "Total Least-Squares Spiral Curve Fitting," *Journal of Surveying Engineering*, Vol. 125, No. 4, Nov. 1999, pp. 159–175. doi:10.1061/(ASCE)0733-9453(1999)125:4(159)

[10] Kelly, A., and Nagy, B., "Reactive Nonholonomic Trajectory Generation via Parametric Optimal Control," *International Journal of Robotics Research*, Vol. 22, No. 7–8, Jan. 2003, pp. 583–601. doi:10.1177/02783649030227008

[11] Hsu, D., Kindel, J. C., Latombe, J. C., and Rock, S., "Randomized Kinodynamic Motion Planning with Moving Obstacles," *Proceedings of the Workshop on Algorithmic Foundation of Robotics*, 2000.

[12] Bryson, A. E., Jr., and Ho, Y. C., *Applied Optimal Control: Optimization, Estimation, and Control*, Hemisphere, New York, 1975.

[13] Grillo, C., and Vitrano, F. P., "State Estimation of a Nonlinear Unmanned Aerial Vehicle Model Using an Extended Kalman Filter," 15th AIAA International Space Planes and Hypersonic Systems and Technologies Conference, Dayton, OH, AIAA Paper 2008-2529, April 2008.

[14] Abdelkrim, N., Aouf, N., Tsourdos, A., and White, B., "Robust Nonlinear Filtering for INS/GPS UAV Localization," *IEEE 16th Mediterranean Conference on Control and Automation*, Inst. of Electrical and Electronics Engineers, Piscataway, NJ, June 2008.

[15] Campbell, M., and Ousingsawat, J., "On-line Estimation and Path Planning for Multiple Vehicles in an Uncertain Environment," AIAA Guidance, Navigation, and Control Conference and Exhibit, Monterey, CA, AIAA Paper 2002-4466, Aug. 2002.

[16] Prevost, C. G., Desbiens, A., and Gagnon, E., "Extended Kalman Filter for State Estimation and Trajectory Prediction of a Moving Object Detected by an Unmanned Aerial Vehicle," *Proceedings of the 2007 American Control Conference*, Inst. of Electrical and Electronics Engineers, Piscataway, NJ, July 2007, pp. 1805–1810.

[17] Kallapur, A. G., and Anavatti, S. G., "UAV Linear and Nonlinear Estimation Using Extended Kalman Filter," *IEEE International Conference on Computational Intelligence for Modeling, Control, and Automation*, Inst. of Electrical and Electronics Engineers, Piscataway, NJ, Nov. 2006.

[18] Korf, R. E., "Real-Time Heuristic Search," *Artificial Intelligence*, Vol. 42, Nos. 2–3, 1990, pp. 189–211. doi:10.1016/0004-3702(90)90054-4

[19] Jun, M., and D'Andrea, R., "Path Planning for Unmanned Aerial Vehicles in Uncertain and Adversarial Environments," *Cooperative Control: Models, Applications and Algorithms*, Butenko, S., Murpheym, R., and Pardalos, P., Springer, New York, 2003.

[20] Holmstrom, K., Goran, A. O., and Edvall, M. M., "User's Guide For TOMLAB/SNOPT," TOMLAB Optimization, Inc., Pullman, WA, 2005.

[21] Hargraves, C. R., and Paris, S. W., "Direct Trajectory Optimization Using Nonlinear Programming and Collocation," *Journal of Guidance, Control, and Dynamics*, Vol. 10, No. 4, July–Aug. 1987, pp. 338–342. doi:10.2514/3.20223

[22] Kumar, R. K., Seywald, H., and Cliff, E. M., "Near-Optimal Three-Dimensional Air-to-Air Missile Guidance Against Maneuvering Target," *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 3, 1995, pp. 457–464. doi:10.2514/3.21409

[23] Betts, J. T., and Huffman, W. P., "Path-Constrained Trajectory Optimization Using Sparse Sequential Quadratic Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 16, No. 1, 1993, pp. 59–68. doi:10.2514/3.11428

[24] Fahroo, M., and Ross, I. M., "Direct Trajectory Optimization by a Chebychev Pseudospectral Method," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 1, 2002, pp. 160–166. doi:10.2514/2.4862

[25] Campbell, M. E., Lee, J. W., and Scholte, E., "Simulation and Flight Test of Autonomous Aircraft Estimation, Planning, and Control Algorithms," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 6, 2007, pp. 1597–1609. doi:10.2514/1.29719