Real-Time Optimized Trajectory Planning for a Fixed-Wing Vehicle Flying in a Dynamic Environment

Jian Yang[1]; Zhihua Qu[2]; Jing Wang[3]; and Richard A. Hull[4]

Abstract: The problem of determining an optimal feasible trajectory, for a fixed wing flying vehicle moving in a dynamical three-dimensional space, is addressed in this paper, and an analytical solution is proposed. With explicitly considering the boundary conditions and kinematic constraints as well as by satisfying the collision avoidance criterions, trajectories are described in terms of three parameterized polynomials, and the family of feasible trajectories are found. Then, the desired near shortest trajectory is chosen from the feasible trajectories by optimizing a performance index with respect to L_2 norm. This trajectory and its associated steering controls are achieved analytically. Computer simulations validate that this approach is computationally efficient and real-time implementable.

DOI: 10.1061/(ASCE)0893-1321(2009)22:4(331)

CE Database subject headings: Optimization; Aircraft; Motion; Collision; Control.

Introduction

A lot of research efforts have been directed toward trajectory planning, and many techniques have been proposed. Among them, the method of potential field pioneered in Khatib (1986) is widely used, and its basic idea is that in planning a trajectory, potential fields are built around obstacles and pathways to expel the trajectory from obstacles and to bring the trajectory close to the final destination. Follow-up work can be found in Warren (1990), Hwang and Ahuja (1992), Kyriakopoulos et al. (1995), and Wang and Lane (1997), and these results address only the trajectory planning problem in two-dimensional environments while three-dimensional (3D) planning can be similarly done but would require much more computation capability. This problem together with the issue of local minima often existed in the potential fields makes the potential field method less than an ideal candidate for real-time path planning in the 3D space.

Also popular are the spline method (Judd and Mclain 2001; Nikolos et al. 2004) and exhaustive search methods (Fujimura 1989; Herman 1986; Chung and Saridis 1989; Bortoff 2000). In the spline method, a sequence of splines are used to generate a trajectory based on nominal paths in 3D space and no moving obstacles are presented in the environment. Liking the spline method, parametrized polynomial methodologies (Lu 1996; Lu 1997; Shen and Lu 2003) are used for getting constrained trajectories but these papers focus on trajectories generation without considering obstacles. Nonetheless, the underlining idea of parametrization and optimization is quite general and useful. On the other hand, in an application of a search-based algorithm, the space is typically divided into regions and a safe path is found for the vehicle by starting at the initial condition and successively searching adjacent regions to the goal. To avoid the need of complete prior knowledge about the environment, an improvement has been proposed in Kitamura et al. (1995), which combines a exhaustive search with the potential field method for 3D path planning in a dynamic environment. Unfortunately, the extension in Kitamura et al. (1995) does not take kinematic constraints into account, and a collision-free 3D path planned may not be feasible for a flying vehicle to follow.

There have been efforts in applying numerical iteration methods to address the presence of vehicle kinematics and dynamics in 3D trajectory planning. In these methods, obstacle avoidance criterion and kinematic model are typically converted into a set of inequality and equality constrains, and numerical iterations are used to approximate or determine the path that satisfies all the constraints. In particular, semidefinite programing is used in Frazzoli et al. (2001), nonlinear point-mass model is studied in Menon et al. (1999), and nonlinear dynamic programing is applied in Raghunathan et al. (2003). All these approaches provide a feasible and collision free trajectory, but they require perfect knowledge of the environment and their numerical iterations make them adequate only for off-line path planning.

Many of modern day applications involve vehicles that are equipped with sensors of limited range but are required to fly through a dynamically changing environment. In these applications, it is critically necessary to real-time plan a 3D trajectory that is collision-free, satisfies motion constraints and boundary conditions, and is optimized in terms of some operational requirements (if possible). To solve this problem, new methodologies are needed because of the aforementioned limitations of the existing

[1]Ph.D. Student, School of Electrical and Computer Sciences, Univ. of Central Florida, Orlando, 91324, FL; Presently, Engineer, Delta Tau Data Systems, Inc. E-mail: fish2bear@gmail.com

[2]Professor, School of Electrical and Computer Sciences, Univ. of Central Florida, Orlando, FL 32816 (corresponding author). E-mail: qu@mail.ucf.edu

[3]Research Assistant Professor, School of Electrical and Computer Sciences, Univ. of Central Florida, Orlando; presently, Assistant Professor, Computer Engineering and Computer Science, School of Science, Engineering, and Mathematics, Bethune-Cookman Univ., 640 Dr. M.M.B. Blvd., Daytona Beach, FL 32114. E-mail: wangj@cookman.edu

[4]Principal Engineer, Advanced Concepts Business Unit of Science Applications International Corporation (SAIC), 3045 Technology Pkwy, Orlando, FL 32826.

Note. This manuscript was submitted on March 23, 2007; approved on September 26, 2008; published online on September 15, 2009. Discussion period open until March 1, 2010; separate discussions must be submitted for individual papers. This paper is part of the ***Journal of Aerospace Engineering***, Vol. 22, No. 4, October 1, 2009. ©ASCE, ISSN 0893-1321/2009/4-331–341/$25.00.

methods. To the best of our knowledge, no result has been reported on analytically solving an optimized 3D trajectory in the presence of both kinematic constraints and static/moving obstacles.

The objective of this paper is to present a real-time optimized trajectory generation algorithm for flying vehicles. The algorithm is based upon polynomial parametrization and online optimization. First, by explicitly satisfying kinematic constraints and boundary conditions, a family of feasible trajectories are analytically derived by applying polynomial parameterizations. Then, the free parameter(s), representing the family of collision-free trajectories, are mapped into appropriate intervals, satisfying the collision avoidance criteria. Finally, according to a distance measure with respect to L_2 norm, the free parameter is optimized within its intervals of avoiding obstacles. The optimal solution is analytical and near shortest. Therefore, the resulting trajectory can be updated real-time as the environment changes are detected.

This paper is organized into five sections. In the "Problem Formulation" section, the problem of the real-time trajectory planning in 3D space and the kinematic model of the fixed wing flying vehicle are described. In the "Optimal Real-Time Trajectory Planning for Flying Vehicle" section, the real-time trajectories are expressed by polynomial parametrizations; then, the optimal collision-free trajectory is determined. In then "Simulation" section, simulations are performed for demonstrating effectiveness of the proposed approach. The conclusion is drawn in the "Conclusion" section.

Problem Formulation

A typical scenario of trajectory planning is that for any given fixed wing flying vehicle, its sensing range is limited and its environmental changes are to appearance/disappearance and/or motion of obstacles in the vicinity. The vehicle moves from initial configurations $\mathbf{q}_0=(x_0,y_0,h_0,v_0,\gamma_0,\psi_0)$ at time t_0 to terminal configurations $\mathbf{q}_f=(x_f,y_f,h_f,v_f,\gamma_f,\psi_f)$ at time t_f $(t_f>t_0)$, where (x,y,h) is vehicle's coordinates, v stands for its velocity, γ refers to the flight-path angle, and ψ denotes the heading angle. For the illustrative example, the 3D version is depicted in Fig. 1 in which the vehicle is represented by the sphere centered at $\mathbf{O}(t)=(x,y,h)$ and of radius r_0, its sensor range is also spherical and of radius R_s, and its velocity is $\mathbf{v}(t)\triangleq[\dot{x},\dot{y},\dot{h}]^T$. The ith obstacle $(i=1,\ldots,n)$ are represented by the sphere centered at point $\mathbf{O}_i(t)=(x_i,y_i,h_i)$ and of radius r_i. For moving objects, the origin $\mathbf{O}_i(t)$ is moving with linear velocity vector $\mathbf{v}_i(t)\triangleq[v_{i,x},v_{i,y},v_{i,h}]$.

A trajectory is said to be feasible if it satisfies the boundary conditions, the kinematic model constraint, and the collision avoidance criterion. The real-time trajectory planning problem is to find trajectory $\mathbf{q}(t)$ to satisfy kinematic model constraint, which could be described by $M(\mathbf{q},\dot{\mathbf{q}})=0$ as detailed shortly in the "Model of flying vehicle" section, to meet the boundary conditions of $\mathbf{q}_0$ and $\mathbf{q}_f$, and to avoid all the obstacles in the environment during the motion. To ensure solvability and to simplify the technical development, the following conditions are introduced:

C-1: Physical envelopes of the vehicle and all the obstacles are known. Unless stated otherwise, the envelopes are assumed to be their smallest circumscribing spheres.

C-2: Sampling period T_s used by onboard sensors and steering controls is chosen such that T_s is small, that $\bar{k}=(t_f-t_0)/T_s$ is an integer, that position $\mathbf{O}_i$ [i.e., $\mathbf{O}_i=(x_i^k,y_i^k)$ at $t=t_0+kT_s$] of all the obstacles within the sensing range are detected at the beginning of each sampling period, and that their velocities $\mathbf{v}_i^k\triangleq[v_{i,x}^k,v_{i,y}^k]^T$ are known and (approximately) constant for $t\in[t_0+kT_s,t_0+(k+1)T_s)$.

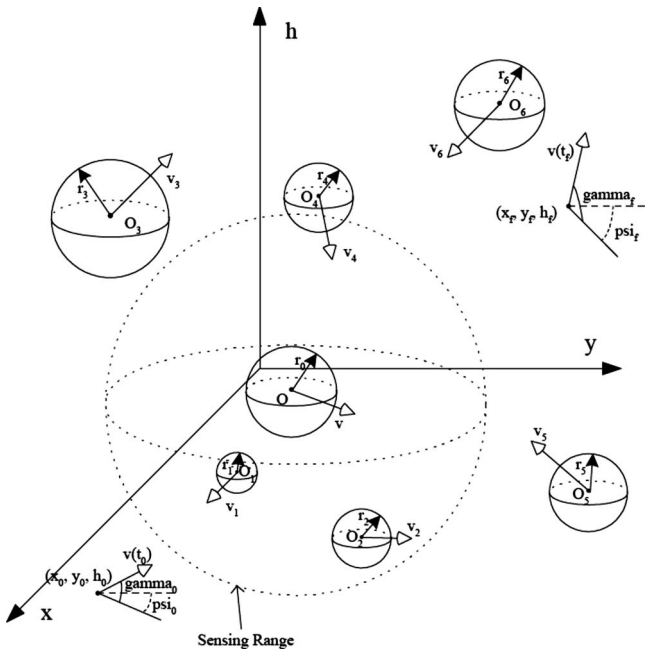

Fig. 1. Flying vehicle moving in a dynamic environment

C-3: All obstacles must be avoided. The obstacles do not form an unescapable trap, nor does any of the obstacles prevent the flying vehicle arriving at $\mathbf{q}_f$ indefinitely, and the vehicle can avoid any of the obstacles by moving faster than them. If needed, intermediate waypoints (and their configurations) can be found such that the feasible trajectory can be expressed by segments parameterized by a class of polynomial functions. Furthermore, the feasible trajectory is updated with respect to sampling period T_s in order to accommodate the environmental changes.

C-4: Boundary configurations $\mathbf{q}_0$ and $\mathbf{q}_f$ have the properties free of kinematic model's singularities or intermediate waypoints could be assigned to avoid these singularities.

C-5: For simplicity, no consideration is given to any of additional constraints such as maximum speed, minimum turning radius, etc. Under these conditions, an optimized trajectory planning algorithm can be developed as will be shown in the following sections.

In applications, Conditions C-1–C-5 can be relaxed in the following ways. Collision avoidance is achieved by imposing an appropriate minimum distance between any two objects of certain physical envelopes. If the envelopes are all spherical, the minimum distance is in terms of distance between two centers. Condition C-1 can relaxed to polygonal envelopes since they can be treated as compositions of small spheres. Though small sampling period T_s is required to detect the changes in obstacles position and velocity, the velocities in Condition C-2 may be estimated from position measurements. The intermediate waypoints required in Conditions C-3 and C-4 can be determined by applying the heuristic approach of either A* or D* search (Stentz 1994; Stentz 1995). Should the vehicle velocity be limited, it may not successfully avoid a fast-moving constant-speed obstacle unless the sensing range is properly increased. Should the vehicle be subject to certain minimum turn radius and velocity bounds, Condition C-5 can be relaxed by combining the Dubins algorithm (Dubins 1957; Sussmann and Tang 1991; Boissonnat et al. 1992).

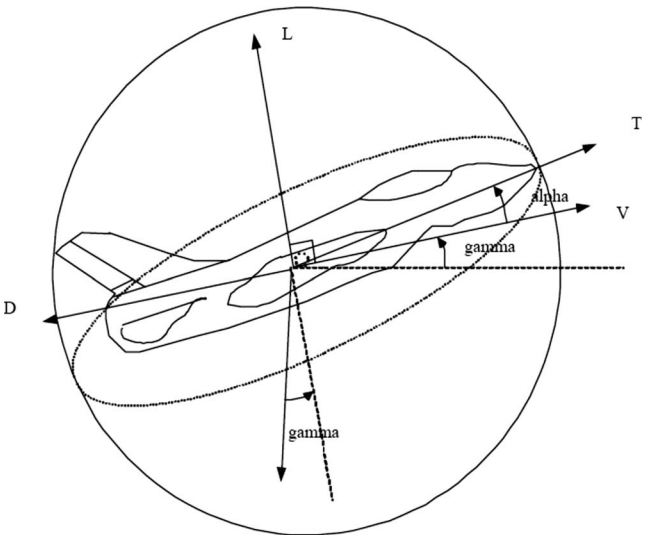

Fig. 2. Coordinate systems and forces

Graphically, vehicle operations can be represented by Fig. 1, and the above abstraction accounts for most of the issues encountered in practice. Also, sampling intervals could vary except that notations would become more complicated. A performance index $J(\mathbf{q},\dot{\mathbf{q}})$ should be chosen to satisfy some physical requirements such as the shortest path, the minimal energy, and the least time. To this end, the trajectory planning problem of this paper is mathematically formulated as the following optimization problem:

$$\min J(\mathbf{q},\dot{\mathbf{q}})$$

$$s.t.\ \mathbf{q}(t_0) = \mathbf{q}_0$$

$$\mathbf{q}(t_f) = \mathbf{q}_f$$

$$M(\mathbf{q},\dot{\mathbf{q}}) = 0$$

$$(x - x_i)^2 + (y - y_i)^2 + (h - h_i)^2 \geqslant (r_0 + r_i)^2, \quad i = 1, \ldots, n \quad (1)$$

Model of Flying Vehicle

The investigation in this paper focuses upon a fixed wing flying vehicle as shown in Fig. 2 (Vinh 1993).

Here force T is the thrust from the engine along the flying vehicle fuselage. The angle α is the angle of attack. Force L is the aerodynamic lift which is normal to the direction of velocity $\mathbf{v}$ and in the symmetric plane (lift-drag plane) of the flying vehicle fuselage. Force D is the aerodynamic drag against the direction of velocity v. The relationship of the forces is shown in the coordinate systems in Fig. 3.

Point mass M is the origin of two coordinate systems: the local-horizon system $x'-y'-z'$ and the wind-axis system $x_1-y_1-z_1$. The two rotations from $x'-y'-z'$ to $x_1-y_1-z_1$ are the heading angle ψ (about x') and flight-path angle γ (about negative z_1). The coordinate r in the figure is measured from the center of the Earth. The location of M is expressed by the topocentric (Earth surface) Cartesian coordinates (x,y,h), where x axis is parallel to y' axis; y axis is parallel to z' axis; $h=r-r_E$, where r_E is the radius of the Earth. The angle σ, which is the angle between the vector L and the (r,v) plane, is referred to as the roll, or bank angle. The resultant force F_T along the velocity vector is given by

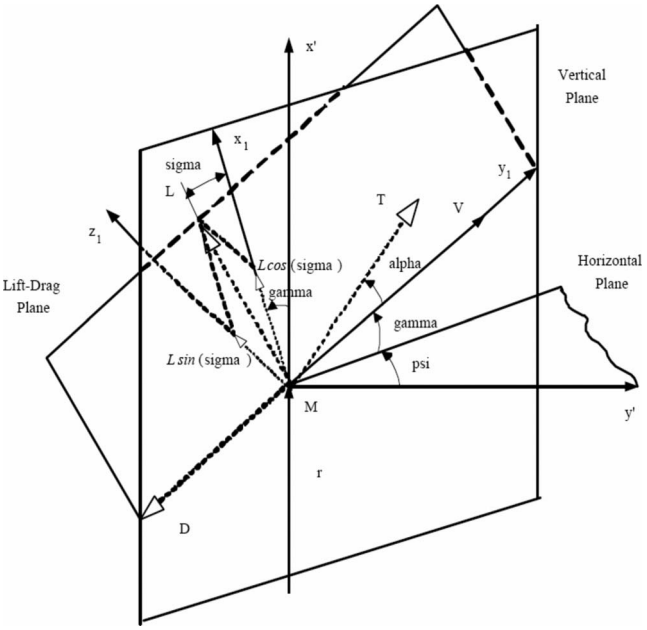

Fig. 3. Coordinate systems and forces

$$F_T = T\cos\alpha - D$$

The resultant force F_N normal to the velocity vector and in the lift-drag plane is given by

$$F_N = T\sin\alpha + L.$$

The equations of kinematics and motion are

$$\dot{x} = v\cos\gamma\cos\psi$$

$$\dot{y} = v\cos\gamma\sin\psi$$

$$\dot{h} = v\sin\gamma$$

$$\dot{v} = \frac{F_T}{M} - g\sin\gamma$$

$$\dot{\gamma} = \frac{F_N\cos\sigma}{Mv} - \frac{g\cos\gamma}{v}$$

$$\dot{\psi} = \frac{F_N\sin\sigma}{Mv\cos\gamma} \quad (2)$$

where g=gravitational acceleration; the controls are F_T, F_N, and bank angle σ. To the pilot, such controls are called "throttle," "stick pull-push," and "stick left-right."

If x and h are parametrized in terms of t, and y is expressed by x in a closed-form expression, F_T, F_N, and bank angle σ could be solved analytically. Therefore, let $\ddot{x}$ and $\ddot{h}$ denote the second order time derivatives of x and h, respectively, and assume dy/dx and d^2y/dx^2 exist. Given the doubly differential trajectory (x,y,h), the controls are solved as follows:

$$\sigma = \arctan\frac{M\dot{x}\dot{\psi}/\cos\psi}{M\dot{\gamma}\dot{x}/(\cos\psi\cos\gamma) + Mg\cos\gamma}$$

$$F_N = \frac{M\dot{x}\dot{\psi}}{\sin\sigma\cos\psi}$$

$$F_T = \frac{M(\ddot{x} + \dot{\psi}\dot{x}\tan\psi)}{\cos\gamma\cos\psi} + M(g+1)\sin\gamma + F_N\sin\gamma \tag{3}$$

where

$$\psi = \arctan\left(\frac{dy}{dx}\right)$$

$$\dot{\psi} = \frac{\dot{x}\dfrac{d^2y}{dx^2}}{1+\left(\dfrac{dy}{dx}\right)^2}$$

$$\gamma = \arctan\left(\frac{\dot{h}\cos\psi}{\dot{x}}\right)$$

$$\dot{\gamma} = \frac{\ddot{h}\dot{x}\cos^3\psi - \dot{h}\ddot{x}\cos\psi + \dot{h}\dot{x}\dot{\psi}\sin\psi}{\dot{x}^2 + \dot{h}^2\cos^2\psi}$$

Remark 0.1: The coordinate system used, local-horizon, is the one commonly used in orbital mechanics. The equations of motion actually represent a flat-earth model. Thus, they can be used for any airplane.

Remark 0.2: Kinematic model (2) has singularity at $\gamma = \pm\pi/2$, which does not occur mathematically or in practice if the range of γ is limited to $(-\pi/2, \pi/2)$; expression (3) also has singularity at $\psi = \pm\pi/2$, which can be fixed by rotating the coordinate systems.

Optimal Real-time Trajectory Planning for Flying Vehicle

In this section, the proposed method of trajectory planning is developed in three steps. First, without considering obstacles, three parametrized polynomials are employed to describe the trajectories satisfying the boundary condition and kinematic model. Then, to handle the dynamically moving obstacles, the polynomials are made piecewise and a simple collision avoidance criterion is developed. These techniques result in the family of feasible trajectories. Finally, by introducing a L_2 norm performance index, the problem is re-formulated as a constrained optimization problem. The optimal solution is analytical and near shortest so that it is efficient for real-time implementation.

Trajectory Planning without Considering Obstacles

It follows from Eq. (2) that for the flying vehicle, the kinematic constraints in horizontal plane are

$$\dot{x} = v\cos\gamma\cos\psi$$

$$\dot{y} = v\cos\gamma\sin\psi$$

$$\dot{\psi} = \frac{F_N\sin\sigma}{Mv\cos\gamma} \tag{4}$$

and the constraint on vertical axis is

$$\dot{h} = v\sin\gamma. \tag{5}$$

The state variables x, y, h, and ψ are directly affected by v, γ, and $F_{N\sin\sigma}$. Without considering kinematic constraints on v and γ, we assume $v\cos\gamma$, $v\sin\gamma$, and $F_{N\sin\sigma}$ to be surrogate controls. It is obvious that (x, y, ψ) and h could be controlled independently; in other words, they are allowed to be planned separately. The following lemma shows the property of the horizontal projection of the vehicle's trajectories.

Lemma 1: (Qu et al. 2004) Consider a flying vehicle with kinematic model (2) moves in an obstacle-free environment. Given the boundary conditions $\mathbf{q}_0$ and $\mathbf{q}_f$ for $t_f > t_0$, the projection of the flying vehicle's trajectory on horizontal plane is expressed by a function in the form of $y = F(x)$, which satisfies the boundary conditions

$$F(x_0) = y_0, \quad F(x_f) = y_f, \quad \left.\frac{dF(x)}{dx}\right|_{x_0} = \tan(\psi_0), \quad \left.\frac{dF(x)}{dx}\right|_{x_f} = \tan(\psi_f) \tag{6}$$

and the kinematic constraints (4)

In horizontal plane, boundary conditions (6) represents four constraints. As discussed in paper of the Qu et al. (2004), function $y = F(x)$ can be chosen to be a parametrized polynomial, at least third order. The third order polynomial renders a unique trajectory and gives no helps to avoid obstacles. Polynomials with higher orders (>3) generate a class of trajectories so that collision-free trajectories can be chosen from them; in other words, extra freedom is provided for avoiding obstacles. Obviously, the fourth order polynomial is the simplest choice, the framework for trajectory planning will be discussed based on it.

Given t_0, t_f, and the boundary conditions of γ and ψ, variable x is imposed by the following boundary constraints:

$$x(t_0) = x_0, \quad x(t_f) = x_f, \quad \dot{x}(t_0) = v_0\cos\gamma_0\cos\psi_0, \quad \dot{x}(t_f) = v_f\cos\gamma_f\cos\psi_f.$$

Obviously, a polynomial, no less than third order, in terms of t will match the requirements. This polynomial is shown as the following equation:

$$x(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \tag{7}$$

where

$$[a_0\ a_1\ a_2\ a_3]^T = (\mathbf{B}_1)^{-1}\mathbf{Y}_1$$

with

$$\mathbf{B}_1 = \begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 \\ 0 & 1 & 2t_0 & 3t_0^2 \\ 1 & t_f & t_f^2 & t_f^3 \\ 0 & 1 & 2t_f & 3t_f^2 \end{bmatrix}, \quad \mathbf{Y}_1 = \begin{bmatrix} x_0 \\ v_0\cos\gamma_0\cos\psi_0 \\ v_f\cos\gamma_f\cos\psi_f \end{bmatrix}.$$

Recalling Lemma 1 (Qu et al. 2004), we can express the trajectory projected on horizontal plane, that is, $y = F(x)$, as a fourth order polynomial, which provides freedom in horizontal plane to avoid obstacles. The polynomial is

$$y[x(t)] = b_0 + b_1 x + b_2 x^2 + b_3 x^3 + b_4 x^4 \tag{8}$$

where

$$[b_0\ b_1\ b_2\ b_3]^T = (\mathbf{B}_2)^{-1}(\mathbf{Y}_2 - \mathbf{A}_2 b_4)$$

with

$$\mathbf{A}_2=\begin{bmatrix}(x_0)^4\\4(x_0)^3\\(x_f)^4\\4(x_f)^3\end{bmatrix},\quad \mathbf{B}_2=\begin{bmatrix}1&x_0&(x_0)^2&(x_0)^3\\0&1&2x_0&3(x_0)^2\\1&x_f&(x_f)^2&(x_f)^3\\0&1&2x_f&3(x_f)^2\end{bmatrix},\quad \mathbf{Y}_2=\begin{bmatrix}y_0\\\tan\psi_0\\y_f\\\tan\psi_f\end{bmatrix}$$

Equations (7) and (8) are rewritten as

$$x(t)=(\mathbf{B}_1)^{-1}\mathbf{Y}_1[1\ t\ t^2\ t^3]$$

$$y[x(t)]=(\mathbf{B}_2)^{-1}(\mathbf{Y}_2-\mathbf{A}_2b_4)[1\ x\ x^2\ x^3]+b_4x^4 \tag{9}$$

Remark 0.3: Note that $t_f>t_0$, thus $\mathbf{B}_1$ is not singularly. In the case that $x_0=x_f$, $\mathbf{B}_2$ is not invertible. However, this problem is easily resolved by specifying an intermediate point $x_m\neq x_0=x_f$ and planning two subpaths. For the subpath between x_0 and x_m, the corresponding $\mathbf{B}_2$ is invertible, and so is $\mathbf{B}_2$ for the subpath from x_m to x_f. Therefore, the parametrized polynomials are still applicable.

According to Eq. (5), the variable h should satisfy boundary conditions

$$h(t_0)=h_0,\quad h(t_f)=h_f,\quad \dot{h}(t_0)=v_0\sin\gamma_0,\quad \dot{h}(t_f)=v_f\sin\gamma_f$$

To satisfy the boundary conditions and provide freedom to avoid obstacles in the vertical direction, a fourth order polynomial in terms of time is chosen as

$$h(t)=c_0+c_1t+c_2t^2+c_3t^3+c_4t^4 \tag{10}$$

The coefficients are

$$[c_0\ c_1\ c_2\ c_3]^T=(\mathbf{B}_3)^{-1}(\mathbf{Y}_3-\mathbf{A}_3c_4)$$

where

$$\mathbf{Y}_3=\begin{bmatrix}h_0\\v_0\sin\gamma_0\\h_f\\v_f\sin\gamma_f\end{bmatrix},\quad \mathbf{A}_3=\begin{bmatrix}(t_0)^4\\4(t_0)^3\\(t_f)^4\\4(t_f)^3\end{bmatrix},\quad \mathbf{B}_3=\mathbf{B}_1$$

Equation (10) is rewritten as

$$h(t)=(\mathbf{B}_3)^{-1}(\mathbf{Y}_3-\mathbf{A}_3c_4)[1\ t\ t^2\ t^3]+c_4t^4 \tag{11}$$

Thus, the trajectories are parametrized by three polynomials shown in Eqs. (9) and (11). Substituting (x,y,h) in Eqs. (9) and (11) into Eq. (3), we obtain the analytical solutions of real controls F_N, F_T, and σ. The trajectory obtained is expressed by three polynomials in terms of time t and two free parameters, b_4 and c_4. When obstacles are presented in the environment, free parameters b_4 and c_4 provide the extra freedom to avoid the potential bumping. Hence the feasible trajectory can be obtained.

Trajectory Planning with Static and Moving Obstacles

Since obstacles' velocities change dynamically and the range of sensor is limited, the planned trajectory should be replanned at once as long as the environment's information is updated. We make the polynomials in the "Trajectory Planning without Considering Obstacles" section piecewise to deal with this problem.

Let $T_m\triangleq t_f-t_0$ to be the time for the flying vehicle to complete its maneuver and T_s be the sampling period such that $\bar{k}=T_m/T_s$ is an integer, that centers of obstacles $\mathbf{O}_i$ are located at (x_i^k,y_i^k,h_i^k) at $t=t_0+kT_s$, and that these objects are all moving with known constant velocities $\mathbf{v}_i^k=[v_{i,x}^k,v_{i,y}^k,v_{i,h}^k]$ for $t\in[t_0+kT_s,t_0+(k+1)T_s]$, where $k=0,1,\ldots,\bar{k}-1$. When $k=0$, the initial condition is $\mathbf{q}_0$. For $0<k<\bar{k}$, the initial condition is given by $\mathbf{q}_k=(x_k,y_k,h_k,v_k,\gamma_k,\psi_k)$, which is the state at the beginning of the kth sampling period. The terminal condition always is $\mathbf{q}_f$. In the latter part of this paper, all the notations with superscript k or subscript k mean they are in the kth sampling period. As depicted in the "Trajectory Planning without Considering Obstacles" section, the same method can be used for real-time replanning as k increases such that, in the kth sampling period, the trajectories are described by

$$x(t)=(\mathbf{B}_1^k)^{-1}\mathbf{Y}_1^k[1\ t\ t^2\ t^3] \tag{12}$$

$$y(x(t))=(\mathbf{B}_2^k)^{-1}(\mathbf{Y}_2^k-\mathbf{A}_2^kb_4^k)[1\ x\ x^2\ x^3]+b_4^kx^4 \tag{13}$$

$$h(t)=(\mathbf{B}_3^k)^{-1}(\mathbf{Y}_3^k-\mathbf{A}_3^kc_4^k)[1\ t\ t^2\ t^3]+c_4^kt^4 \tag{14}$$

For $t\in[t_0+kT_s,t_0+(k+1)T_s]$, the collision avoidance criterion is

$$(x-x_i^k-v_{i,x}^k\tau)^2+(y-y_i^k-v_{i,y}^k\tau)^2+(h-h_i^k-v_{i,h}^k\tau)^2\geqslant(r_i+r_0)^2 \tag{15}$$

where $\tau=t-t_k$ for $t\in[t_0,t_f]$.

Except the time interval that collision may happen, it is not necessary to consider this criterion over the whole time domain. Using the relative velocity and treating ith obstacle as a "static" one, the collision avoidance criterion should only be considered for $x_i'(t)\in[\underline{x}_i',\bar{x}_i']$, with $\underline{x}_i'=x_i^k-r_i-r_0$ and $\bar{x}_i'=x_i^k+r_i+r_0$. The relative position of the flying vehicle at time t is $x_i'(t)=x(t)-v_{i,x}^k\tau$. Thus time interval $[\underline{t}_i^*,\bar{t}_i^*]\subset[t_0,t_f]$ is solved from the set $x_i^k\in[x(t)-v_{i,x}\tau-r_i-r_0,x(t)-v_{i,x}\tau+r_i+r_0]$.

Letting $c_4^k=b_4^k$ for convenience, we obtain the following Theorem 1.

Theorem 1: Consider a flying vehicle, whose kinematics model is given by Eq. (2), moving in an environment with dynamically moving and static obstacles, and its trajectories are parameterized by Eqs. (12)–(14). Under boundary conditions $\mathbf{q}_k$ and $\mathbf{q}_f$, the trajectories, avoiding n obstacles, are mapped into collision-free intervals of b_4^k, which is

$$\Omega\triangleq\{b_4^k:b_4^k\notin\cup_{i=1}^n(\underline{d}_i^k,\bar{d}_i^k)\} \tag{16}$$

where $\underline{d}_{1,i}^k$ and $\bar{d}_{1,i}^k$=upper and lower bound, incurring collision to the ith obstacle, respectively.

Proof: Substituting polynomials (12)–(14) into Eq. (15), it leads to

$$G(b_4^k,t,\tau)=\min_{i\in[\underline{t}_i^*,\bar{t}_i^*]}g_2(t,\tau)(b_4^k)^2+g_{1,i}(t,\tau)b_4^k+g_{0,i}(t,\tau)\geqslant 0 \tag{17}$$

The expressions of g_2, $g_{1,i}$ and $g_{0,i}$ are as follows:

$$g_2=(f_1)^2+(f_3)^2$$

$$g_{1,i}=2f_1(f_2-y_i^k-v_{i,y}^k\tau)+2f_3(f_4-h_i^k-v_{i,h}^k\tau)$$

$$g_{0,i}=(f_2-y_i^k-v_{i,y}^k\tau)^2+(f_4-h_i^k-v_{i,h}^k\tau)^2+f_{5,i}$$

where

$$f_1=x^4-[1\ x\ x^2\ x^3](\mathbf{B}_2^k)^{-1}\mathbf{A}_2^k,$$

$$f_2=[1\ x\ x^2\ x^3](\mathbf{B}_2^k)^{-1}\mathbf{Y}_2^k,$$

$$f_3=t^4-[1\ t\ t^2\ t^3](\mathbf{B}_3^k)^{-1}\mathbf{A}_3^k,$$

$$f_4 = [1\ t\ t^2\ t^3](\mathbf{B}_3^k)^{-1}\mathbf{Y}_3^k,$$

$$f_{5,i} = (x - x_i^k - v_{i,x}^k\tau)^2 - (r_i + r_0)^2. \tag{18}$$

It follows from Eq. (17) that $g_2 \geqslant 0$. If $g_2 > 0$, inequality (17) belongs to the family of parabolas opening upward and in terms of b_4^k. Thus, as long as $g_2 > 0$, a solution to b_4^k always exists for any one object and, for the ith object, the solution is of the form $b_4^k \notin (\underline{d}_i^k, \overline{d}_i^k)$, where $\underline{d}_i^k$ and $\overline{d}_i^k$ are the two roots of the equality version of (17). In the presence of multiple objects, the final solution is given by $\Omega \triangleq \{b_4^k : b_4^k \notin \cup_{i=1}^n (\underline{d}_i^k, \overline{d}_i^k)\}$, and it always yields at least one finite value for b_4^k unless the objects and their associated collision regions make the free space disconnected between initial condition $\mathbf{q}_0$ and terminal condition $\mathbf{q}_f$. On the other hand, if $g_2 = 0$, it follows that $f_1 = f_3 = 0$, which leads to $g_{1,i} = 0$. In this case, the choice of b_4^k plays no role in making inequality (17) valid. Studying the properties of Eqs. (13) and (14), we know that if $g_2 = 0$, the expressions of Eqs. (13) and (14) are degenerated into third order polynomials. Clearly, this only happens at the boundary conditions. For boundary points, it is not necessary to consider inequality (17) unless t_0 and/or t_f are within the time collision interval. For the case when $t_0 \in [\underline{t}_i^*, \overline{t}_i^*]$, collision has happened already, and nothing can be done. When $t_f \in [\underline{t}_i^*, \overline{t}_i^*]$, collision can be avoided by adjusting t_f. Other than these two cases, $g_{0,i} > 0$ and hence inequality (17) is valid even if $g_{1,i} = 0$.

Key properties of these trajectories are shown as follows:

- **Solvability:** Theorem 1 cannot guarantee the set Ω always be nonempty if there are to many obstacles or singularities of the kinematics model exist. In those case, intermediate waypoints required in Conditions C-3 and C-4 can be determined by applying the heuristic approach of either A* or D* search (Stentz 1994; Stentz 1995). Since A* and D* could always find the paths, unless the objects and their associated collision regions make the free space disconnected between initial condition $\mathbf{q}_0$ and terminal condition $\mathbf{q}_f$, the feasible trajectories can be obtained through Theorem 1 with appropriate intermediate points. How to choose these pints will be discuss in other paper.
- **Convergence:** In each sampling period, the trajectory is replanned and the end boundary constraints are explicitly considered in the parametrized polynomials (12)–(14) through matrices $\mathbf{Y}_1^k$, $\mathbf{Y}_2^k$, and $\mathbf{Y}_3^k$. As the solution of b_4^k can always be determined, we know that trajectories obtained are ensured to converge to the end point.
- **Complexity:** Computationally, the proposed approach requires that inequality (17) be solved for at most $N \times k$ times, where N is the number of the objects and $\bar{k}$ is the number of sampling periods. The equality version of Eq. (17) has two closed-form solutions, and its computational complexity depends only on the number of boundary conditions imposed (that is, matrix multiplications are six dimensional). Therefore, the proposed algorithm is well suited for real-time implementation. In comparison, a tree search routine would depend on a product of N, n_t, n_x, and n_y, where n_t, n_x, and n_y are the numbers of grids along the t, x, and y axes, respectively. Since it is usual that $n_t \gg \bar{k}$ and $n_x, n_y \gg 1$, a tree search algorithm is rapidly growing and, if implemented online, is, in comparison, much more computationally intensive.
- **Robustness:** The sensor range of the vehicle is limited, and motion of obstacles are not modeled or known apriori or predictable, hence the environment is dynamical. Since the collision free interval of b_4^k and all control inputs are analytical, and the solvability and convergence of trajectories are guaranteed, the ability to deal with dynamics of environment stems from the fact that the algorithm is fast and new solutions can be recomputed quickly as new information becomes available. On the other hand, disturbances, uncertainties and noises could make the vehicle deviate the obtained trajectory. However, this problem is resolved by applying a close-loop controller proposed in paper (Qu et al. 2006) or references therein. By embedding the proposed planning algorithm into such a closed loop control framework, this path planning algorithm becomes robust.

Remark 0.4: In the vicinity of but not exactly at the boundary conditions, g_2 is close to being zero. If t_0 and/or t_f are very near the interval $[\underline{t}_i^*, \overline{t}_i^*]$, one can form a proper solution not by solving inequality (17), but by adjusting the robot speed so that t_0 and/or t_f are not very close to the boundary time instants.

Remark 0.5: When moving objects change their velocities rapidly, T_s should be chosen to be sufficiently small. On the other hand, if the flying vehicle approaches the goal, smaller T_s makes matrixes $\mathbf{B}_1^k$, $\mathbf{B}_2^k$, and $\mathbf{B}_3^k$ closer to being singular. Thus, it is necessary for computational efficiency and robustness that T_s is not too small.

Remark 0.6: In our proposed method, we assume $c_4^k = b_4^k$. Thus, the trajectory planning problem in 3-D space is simplified to find a free parameter's solution for avoiding obstacles. This assumption give us convenience for finding a feasible trajectory. On the other hand, it makes the result a conservative solution. Without imposing the constraint $b_4^k = c_4^k$, the left side of inequality (17) changes from a parabola opening upward in terms of b_4^k to an ellipse in terms of b_4^k and c_4^k. Obviously, in some sense, the family of parabolas are special cases of the class of ellipses. Since the solvability for the case of parabolas is ensured, there always exists the solution for the pair of b_4^k and c_4^k. How to solve them will be investigated in another paper.

Remark 0.7: The trajectories obtained by using polynomial parameterizations are only first-differentiable continuous, thus control inputs F_N and F_T are discontinuous among sampling periods in sequence. This problem can be tackled by adding more boundary conditions, such as $\ddot{x}(t_k)$, $\dddot{x}(t_k)$, $\ddot{h}(t_k)$, and $\dddot{h}(t_f)$ for $k = 0, 1, \ldots, \bar{k}$, so that Eqs. (9) and (11) are increased to fifth-order and applying the proposed method.

Optimal Feasible Trajectory

A class of feasible trajectories is obtained in section; however, not all values of b_4^k are good choices, for instance, some associated paths have long distances or high control energies. An optimal trajectory should be chosen from them according to a certain performance index in term of physical requirements, such as near shortest distance and near minimal control energy. Also, the word "near" is needed to satisfy the real-time calculation requirement and the corresponding performance index should yield its optimal solution in closed form if possible.

Considering all these issues, a performance index is proposed as follows:

$$J_k(b_4^k) = \int_{t_k}^{t_f} [(x - x_l)^2 + (y - y_l)^2 + (h - h_l)^2]dt, \tag{19}$$

where

$$x_l = x$$

$$y_l = \frac{(y_f - y_0)(x - x_k)}{x_f - x_0} + y_0$$

$$h_l = \frac{(h_f - h_0)(x - x_k)}{x_f - x_0} + h_0$$

The point $\mathbf{P}(t) \triangleq (x_l, y_l, h_l)$ belongs to the "initial straight line," that is, the straight line between the start point and the end point. The performance index, in a form of L_2 norm, is an integration to the sum of the square of the coordinates difference, between the class of parameterized trajectories and the initial straight line. For the physical meaning, it reflects the deviation of a parametrized trajectory from the initial straight line. Graphically, it is shown in Fig. 4. The following Theorem 2 presents the optimal b_4^k generating the desired trajectory, which is optimal and feasible in 3D space for the flying vehicle.

Theorem 2: Consider the flying vehicle, whose kinematic constraint is in form (2), operating in an environment with static and dynamically moving obstacles. Its trajectories are expressed by polynomials (12), (14), and (18), for which only a free parameter b_4^k needs to be determined. Let Ω be the set of b_4^k generating collision-free trajectories, solved from inequalities (17). Then, the projection of

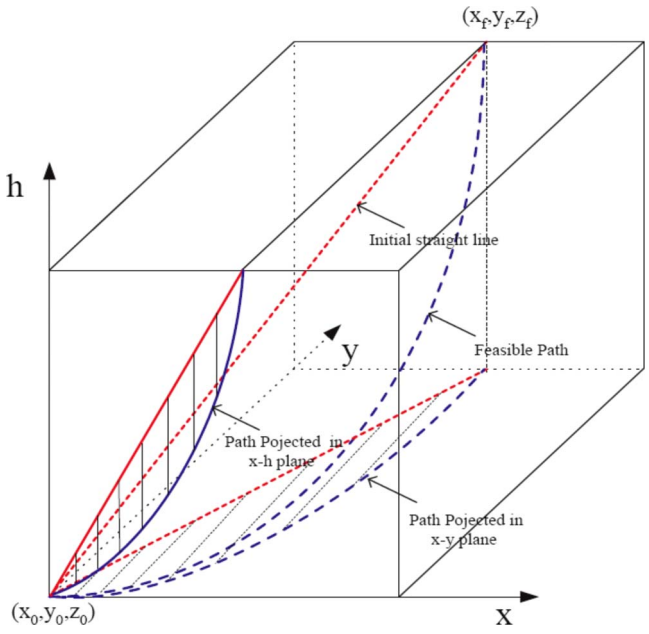

Fig. 4. Relation between the initial straight line and the parameterized trajectory

$$b_4^{k*} = \frac{\int_{t_k}^{t_f} \left[f_1 f_2 + f_3 f_4 - f_1 y_0 - f_3 h_0 - f_1 \frac{(y_f - y_0)(x - x_k)}{x_f - x_0} - f_3 \frac{(h_f - h_0)(x - x_k)}{x_f - x_0} \right] dt}{\int_{t_k}^{t_f} [(f_1)^2 + (f_3)^2] dt} \tag{20}$$

on the set Ω generates near shortest collision-free trajectory while minimizing performance index (19). The definition of the projection is

$$P_{\Omega}^{b_4^{k*}} \triangleq \{ b_4^{k*} \in \Omega : \| b_4^{k**} - b_4^{k*} \| \leqslant \| b_4^k - b_4^{k*} \|, \forall\, b_4^k \in \Omega \}$$

and the corresponding steering control commands are given by Eq. (3).

Proof: Since x_l is chosen to be x, the performance index is reduced to

$$J_k(b_4^k) = \int_{t_k}^{t_f} [(y(t) - y_l(t))^2 + (h(t) - h_l(t))^2] dt = p_1 (b_4^k)^2 + p_2 b_4^k + p_3$$

where

$$p_1 = \int_{t_k}^{t_f} q_1(t) dt = \int_{t_k}^{t_f} [(f_1)^2 + (f_3)^2] dt,$$

$$p_2 = \int_{t_k}^{t_f} q_2(t) dt = 2 \int_{t_k}^{t_f} \left[f_1 f_2 + f_3 f_4 - f_1 y_0 - f_3 h_0 - f_1 \frac{(y_f - y_0)(x - x_k)}{x_f - x_0} - f_3 \frac{(h_f - h_0)(x - x_k)}{x_f - x_0} \right] dt$$

$$p_3 = \int_{t_k}^{t_f} q_3(t) dt = \int_{t_k}^{t_f} \left\{ \left[f_2 - \frac{(y_f - y_0)(x - x_k)}{x_f - x_0} - y_0 \right]^2 + \left[f_4 - \frac{(h_f - h_0)(x - x_k)}{x_f - x_0} - h_0 \right]^2 \right\} dt$$

Since f_1, f_2, f_3, and f_4 all are polynomials in terms of t, the variables q_1, q_2, and q_3 are also polynomials in terms of time t. Thus, it is easy to get the analytical expressions of p_1, p_2, and p_3 through integration. Moreover, f_1 and f_2 are nonzero at each time instants except for t_0 and t_f, and $t_f > t_0$, thus $p_1 > 0$.

$J_k(b_4^k)$ is a second order polynomial in terms of b_4^k, without considering obstacles, its minimal value is achieved as

$$b_4^{k*} = -\frac{p_2}{p_1}$$

Then, it is routine to obtain the analytical expression of b_4^{k*} shown in Eq. (20). Note that $G_i(t, \tau, b_4^k)$ and $J_k(b_4^k)$ all are second order polynomials in terms of b_4^k. This means $J_k(b_4^k)$ is the homeomorphism of b_4^k when $b_4^k \in [b_4^{k*}, +\infty)$ or $b_4^k \in (-\infty, b_4^{k*}]$. Analyzing this property, we know the projection of b_4^{k*} on the set Ω will generate the minimal J_k with considering obstacles. This result means $P_{\Omega}^{b_4^{k*}}$ yielding the optimal solution. Obviously, the optimal solution b_4^{k**} is analytical. ◇

Remark 0.8: Instead of the initial straight line, if the trajec-

tory (x^*, y^*, h^*), generated by $b_4^k = b_4^{k*}$, is used as the reference trajectory, given the performance in the form of L_2 norm, that is,

$$J_k^*(b_4^k) = \int_{t_k}^{t_f} [(x - x^*)^2 + (y - y^*)^2 + (h - h^*)^2] dt \tag{21}$$

where

$$x^*(t) = (\mathbf{B}_1^k)^{-1} \mathbf{Y}_1^k [1 \; t \; t^2 \; t^3],$$

$$y^*(x(t)) = (\mathbf{B}_2^k)^{-1} (\mathbf{Y}_2^k - \mathbf{A}_2^k b_4^{k*}) [1 \; x \; x^2 \; x^3] + b_4^{k*} x^4,$$

$$h^*(t) = (\mathbf{B}_3^k)^{-1} (\mathbf{Y}_3^k - \mathbf{A}_3^k b_4^{k*}) [1 \; t \; t^2 \; t^3] + b_4^{k*} t^4.$$

the same solution as shown in Theorem 2 is achieved.

Remark 0.9: As the alternatives to index (19), there are other performance indices. E.g., performance index with respect to the path length is

$$J_k(b_4^k) = \int_{x_k}^{x_f} \sqrt{1 + \left(\frac{dy}{dx}\right)^2 + \left(\frac{dh}{dx}\right)^2} dx, \tag{22}$$

and the performance index associated with the control energy is

$$J_k(b_4^k) = \int_{t_k}^{t_f} T^2 dt = \int_{t_k}^{t_f} [(F_T + D)^2 + (F_N - L)^2] dt, \tag{23}$$

where $D = drag$ force and $L = lift$ force.

Intuitively, the analytical solution (20) represents a near shortest trajectory, that is, close to the shortest path with respect to performance index (22). On the other hand, given t_0 and t_f, the shorter the length of a path, the smaller the control energy likely becomes (but not necessarily optimal). Since performance indices (22) and (23) do not have analytical solutions, performance index (19) with analytical solution is the better choice for real-time implementation.

Remark 0.10: The initial straight line can be updated dynamically. In each sampling period, the straight line between point (x_k, y_k, h_k) and point (x_f, y_f, h_f) is set to be the initial straight line for this period, that is, variables x_l, y_l, and h_l are renewed as follows:

$$x_l = x$$

$$y_l = \frac{(y_f - y_k)(x - x_k)}{x_f - x_k} + y_k$$

$$h_l = \frac{(h_f - h_k)(x - x_k)}{x_f - x_k} + h_k$$

Similarly, choosing the L_2 norm as performance index, we can obtain a new optimal feasible trajectory. Compared to the one generated by b_4^{k*}, it may be smoother and its length sometime turns shorter, but a general conclusion about which one is better cannot be drawn.

Remark 0.11: If $\dot{h} = 0$, the kinematic model (2) in 3D space reduces to be a model in horizontal plane

$$\dot{x} = u_1 \cos \psi$$

$$\dot{y} = u_2 \sin \psi$$

Table 1. Parameters of Helicopters

Items	Helicopter 1	Helicopter 2	Helicopter 3
Pos.	(6.7, 5.7, 4.2)	(12.2, 5.0, 2.6)	(22.0, 14.1, 9.0)
R	2	1	0.8
Velocity (t=0–10)	(0, 0, 0)	(–0.4, 0.4, 0.2)	(0.1, 0.2, –0.3)
Velocity (t=10–20)	(0, 0, 0)	(0.2, –0.1, –0.4)	(0.2, 0.2, 0.4)
Velocity (t=20–30)	(0, 0, 0)	(1, 0.2, 0.4)	(–0.1, –0.3, –0.3)
Velocity (t=30–40)	(0, 0, 0)	(0.6, 0.2, 0.2)	(0.4, 0.4, –0.2)

$$\dot{\psi} = u_2$$

where $u_1 = v \cos \gamma$ and $u_2 = F_N \sin \sigma / Mv \cos \gamma$. This reduced form is the same as the kinematic model of a tricycle, whose trajectory planning problem is discussed extensively in literature.

Remark 0.12: The geometrical model of the obstacles does not necessarily have to be balls. Suppose that the surface of the *ith* obstacle located in 3D space are described by a closed algebraic implicit defining function in the following form $F_i(x_s, y_s, h_s) = 0$. For the flying vehicle, its most dangerous point x_D, y_D, h_D can be described as a function of its configuration-space variables. The collision avoidance criterion (15) is changed to

$$F_i(x_D, y_D, h_D) \geqslant 0$$

Given some assumptions to let the above inequality be a first or second-order Equation in terms of b_4, the proposed method still works.

Remark 0.13: Provided that all the obstacles in Remark 0.12 are static, the problem becomes planning a trajectory in an unknown terrain, which is also discussed extensively in literature.

Remark 0.14: This approach can be applied to the trajectory planning for underwater robots since the kinematics models are similar.

Simulation

This section shows the simulation results obtained from the proposed approach. Maybe this simulation looks simple but it is representative since a much more complicate case can be decomposed to simple cases with assigning intermediate points. In the scenario of simulation, all quantities conform to a given unit system, for instance, meter, m/s, etc. The sampling period is chosen to be $T_s = 1$ s for convenience. Three helicopters shown as obstacles are presented in the scenario: one is hovering and the other two are moving. The flying vehicle, shown as the fixed wing aircraft, and three helicopters (obstacles) are drawn every 10 s. The velocities of helicopters will change in each 10 s and the settings are as follows:

- Fixed wing aircraft (flying vehicle) parameters: $r_0 = 1$ and $R_s = 10$.
- Boundary conditions: $\mathbf{q}_0 = (0, 0, 0, 1, \pi/6, \pi/4)$ and $\mathbf{q}_f = (30, 20, 10, 2, \pi/10, \pi/5)$; $t_0 = 0$, and $t_f = 40$.
- Helicopters (obstacles) parameters: (shown in Table 1).

Figure 5(a) shows the whole trajectory evolutions of the fixed wing aircraft and the paths of helicopters in 3D space. Their locations are shown in every 10 s. Figures 5(b–f) are snapshots at

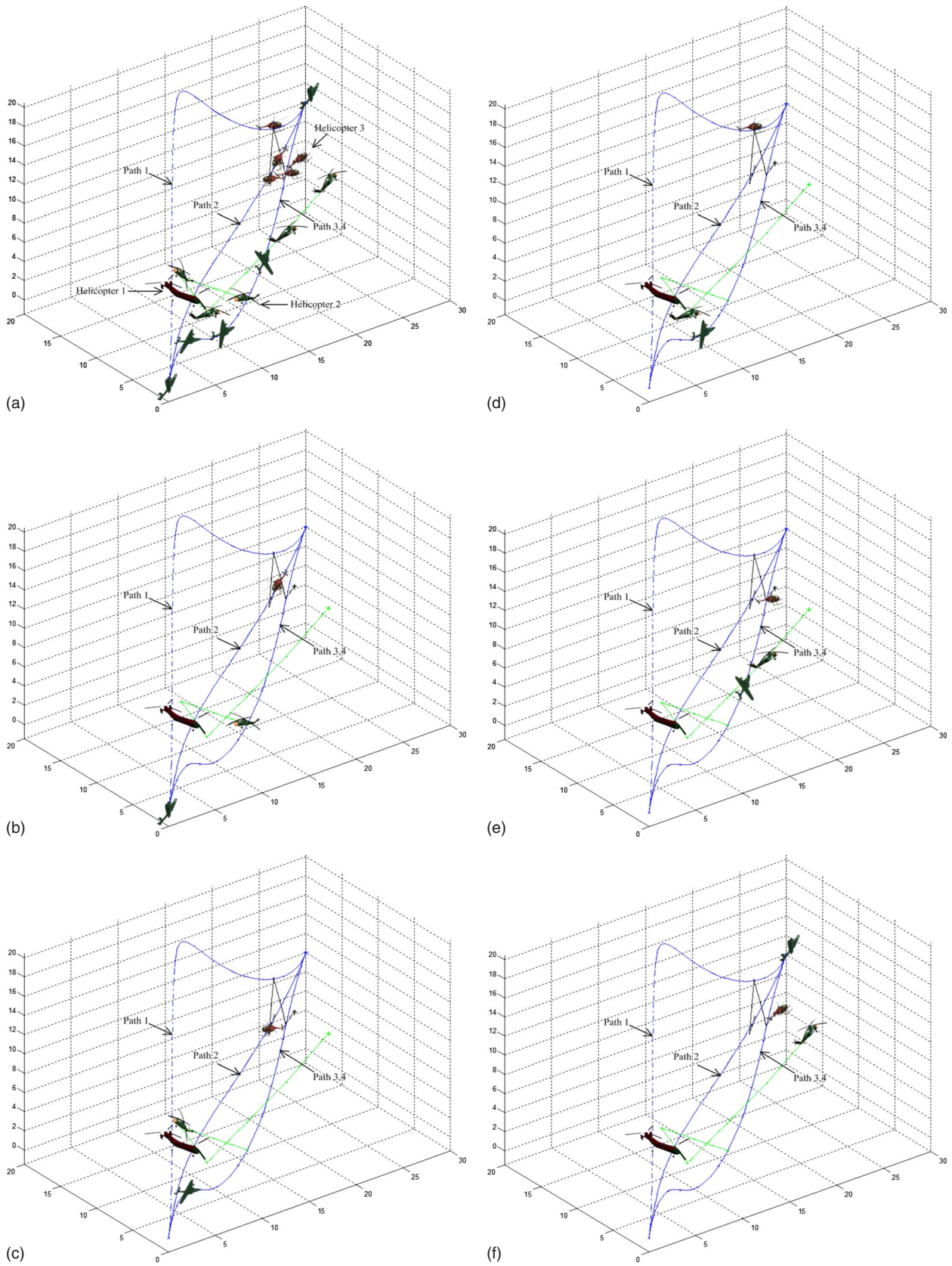

Fig. 5. Simulation results with different b_4^k: (a) trajectories comparison of the fixed wing aircraft in 3D space; (b) snapshot at $t=0$ s; (c) snapshot at $t=10$ s; (d) snapshot at $t=20$ s; (e) snapshot at $t=30$ s; and (f) snapshot at $t=40$ s

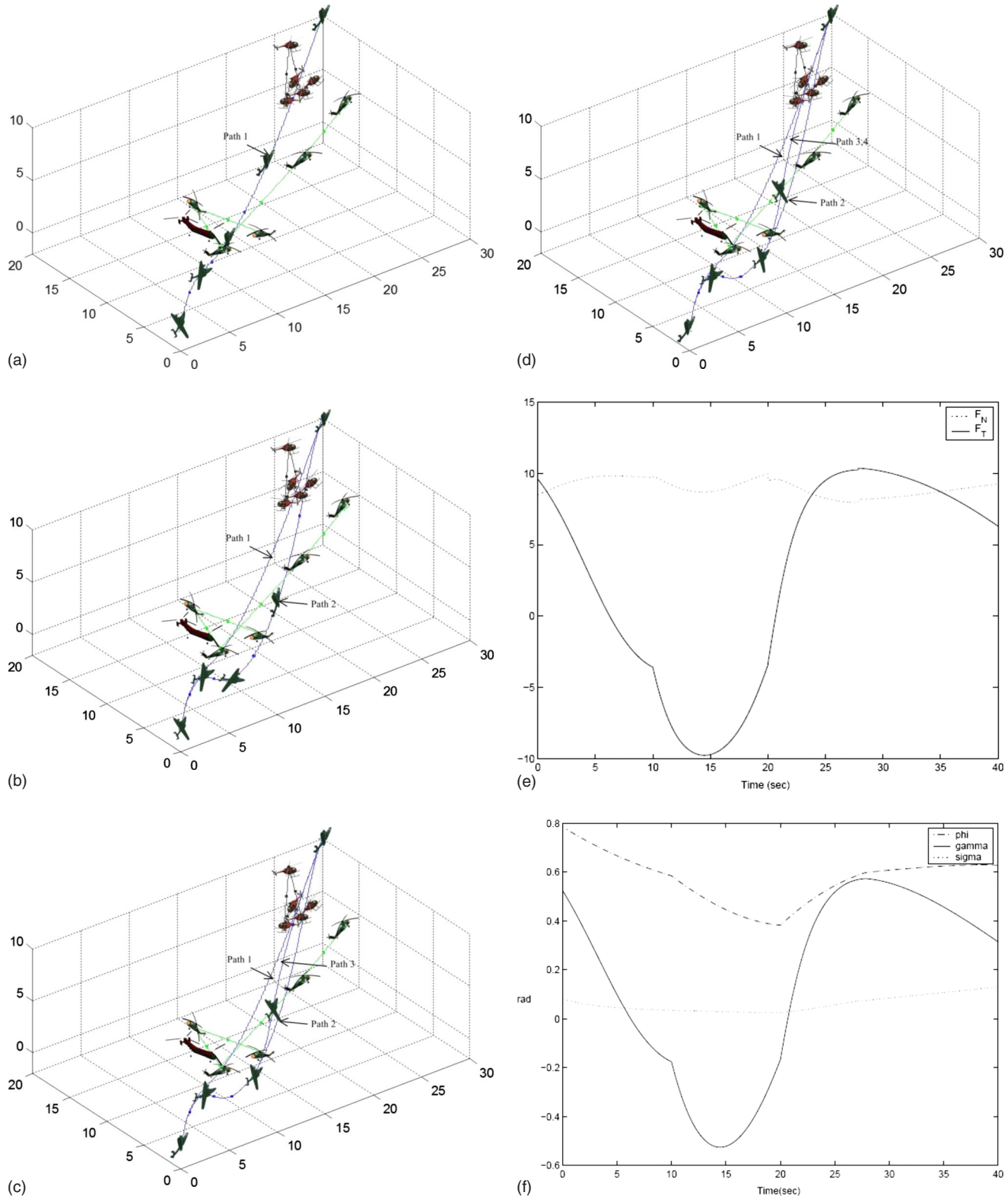

Fig. 6. Evolvement of the optimal trajectory: (a) optimal trajectory obtained at $t=0$ s; (b) optimal trajectory obtained $t=10$ s; (c) optimal trajectory obtained at $t=26$ s; (d) final optimal trajectory; (e) controls F_T and F_N of the fixed wing aircraft; and (f) angels ϕ, γ, and σ of the fixed wing aircraft

each associated instant. In these figures, Path 1 represents the trajectory that was calculated without considering the helicopters, Path 2 is the trajectory generated by arbitrarily chosen b_4^k from the feasible solution set, and Path 3 stands for the optimal trajectory taken by the fixed wing aircraft. We can see that if the fixed wing aircraft follows Path 1, it will collide with all the three helicopters. Path 2 can avoid the collision but it is a long swing. By applying our proposed optimal solution, Path 3 can successfully avoid such collision and shorten the path length effectively. The optimal solution of b_4^k is analytical such that this approach is very

attractive computationally. It is a good real-time trajectory planning algorithm for flying vehicles in 3D space.

Figures 6(a–d) show the dynamical calculation of the optimal path. In Fig. 6(a), it is clear if the fixed wing aircraft follows Path 1 planned at $t=0$ s, it can avoid Helicopter 1. However, it will collide with Helicopter 2, since it does not take Helicopter 2 into account at the first sampling period according to the limited sensor range $R_s=9$. In Fig. 6(b), the path is replanned when Helicopter 2 is detected and a potential collision exists. Path 2 is the replanned results obtained at $t=10$ s. It does not consider Helicopter 3 for the limited sensor range; thus potential collision may exist. Figure 6(c) shows a new Path 3 for shunning the potential crash with Helicopter 3. This path is obtained at $t=26$ s. The final path (Path 4) is shown in Fig. 6(d); it is almost the same as Path 3. Figures 6(e and f) show the fixed wing aircraft's control command and the flying angles, respectively.

Conclusion

A method for planning the trajectory of a flying vehicle moving in a dynamically changing environment has been developed. It is a parametrization approach, since the real-time and feasible trajectory is expressed by parametrized polynomials. The only requirement to obtain the solution of an adjustable parameter makes it possible to generate the realtime trajectory. Key features of this approach are (1) parametrized trajectories are employed to satisfy the boundary condition and kinematic equation; (2) collision-free trajectories are mapped into the associated intervals of the adjustable parameter; (3) an optimal trajectory, with performance index in the form of L_2 norm, for avoiding obstacles is analytically obtained.

This approach was illustrated in an example showing the applicability to avoid static and moving obstacles in 3D space. Also demonstrated was the effectiveness of the optimal solution. In real applications, this algorithm is successfully applied in the trajectory planning software package of L-3 Communications Corporation.

References

Boissonnat, J., Cerezo, A., and Leblond, J. (1992). "Shortest paths of bounded curvature in the plane." *Proc., Int. Conf. on Robotics and Automation*, IEEE, Nice, France, 2315–2320.

Bortoff, S. (2000). "Path planning for uavs." *Proc., American Control Conf.*, Vol. 1, Chicago, 364–368.

Chung, C., and Saridis, G. (1989). "Path planning for an intelligent robot by the extended vgraph algorithm." *Proc., IEEE Int. Symposium on Intelligent Control*, IEEE, Nice, France, 544–549.

Dubins, L. (1957). "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents." *Am. J. Math.*, 79, 497–516.

Frazzoli, E., Mao, Z. H., Oh, J. H., and Feron, E. (2001). "Resolution of conicts involving many aircraft via semidefinite programming." *J. Guid. Control Dyn.*, 24, 79–86.

Fujimura, K. (1989). "A hierarchical strategy for path planning among moving obstacles." *IEEE Trans. Rob. Autom.*, 5, 61–69.

Herman, M. (1986). "Fast, three-dimensional, collision-free motion planning." *Proc., IEEE Int. Conf. on Robotics and Automation*, AIAA, Albany, NY, 1056–1063.

Hwang, Y. K., and Ahuja, N. (1992). "A potential field approach to path planning." *IEEE Trans. Rob. Autom.*, 8, 23–32.

Judd, K., and Mclain, T. (2001). "Spline based path planning for unmanned air vehicles." *Proc., AIAA Guidance, Navigation, and Control Conference and Exhibit*, AIAA, Austin, Tex.

Khatib, O. (1986). "Real-time obstacle avoidance for manipulator and mobile robots." *Int. J. Robot. Res.*, 5, 90–98.

Kitamura, Y., Tanaka, T., Kishino, F., and Yachida, M. (1995). "3-d path planning in a dynamic environment using an octree and an artificial potential field." *Proc., Int. Conf. on Intelligent Robots and Systems*, Vol. 2, Rutgers University, New Brunswick, N.J., 474–481.

Kyriakopoulos, K., Kakambouras, P., and Krikelis, N. (1995). "Potential fields for nonholonomic vehicles." *Proc., IEEE International Symposium on Intelligent Control*, Monterey, Calif., 461–465.

Lu, P. (1996). "Nonlinear trajectory tracking guidance with application to a launch vehicle." *J. Guid. Control Dyn.*, 19, 99–106.

Lu, P. (1997). "Entry guidance and trajectory control for reusable launch vehicle." *J. Guid. Control Dyn.*, 20, 143–149.

Menon, P., Sweriduk, G., and Sridhar, B. (1999). "Optimal strategies for free-flight air traffic conflict resolution." *J. Guid. Control Dyn.*, 22, 202–211.

Nikolos, I., Valavanis, K., Tsourveloudis, N., and Kostaras, A. (2004). "Evolutionary algorithm based offline/online path planner for uav navigation." *IEEE Trans. Syst., Man, Cybern., Part B: Cybern.*, 3, 898–912.

Qu, Z., Wang, J., and Plaisted, C. E. (2004). "A new analytical solution to mobile robot trajectory generation in the presence of moving obstacles." *IEEE Trans. Rob. Autom.*, 20, 978–993.

Qu, Z., Wang, J., Plaisted, C. E., and Hull, R. A. (2006). "Global-stabilizing near-optimal control design for nonholonomic chained systems." *IEEE Trans. Autom. Control*, 51, 1440–1456.

Raghunathan, A., Gopal, V., Subramanian, D., Biegler, L. T., and Samad, T. (2003). "3d conflict resolution of multiple aircraft via dynamic optimization." *Proc., AIAA Guidance, Navigation, and Control Conf. and Exhibit*, Austin, Tex.

Shen, Z., and Lu, P. (2003). "On-board generation of three-dimensional constrained entry trajectories." *J. Guid. Control Dyn.*, 26, 111–121.

Stentz, A. (1994). "Optimal and efficient path planning for partially-knownenvironments." *Proc., 1994 IEEE Int. Conf. on Robotics and Automation*, Vol. 4, San Diego, Calif., 3310–3317.

Stentz, A. T. (1995). "The focussed d* algorithm for real-time replanning." *Proc., Int. Joint Conf. on Artificial Intelligence*, Montreal, Quebec, Canada, 1652–1659.

Sussmann, H. J., and Tang, G. (1991). "Shortest paths for the reedscshepp car: A worked out example of the use of geometric techniques in nonlinear optimal control." *Report SYCON-91-10, Rutgers Univ.*.

Vinh, N. X. (1993). *Flight mechanics of high performance aircraft*. 1st Ed., Cambridge University Press, Cambridge, U.K.

Wang, Y., and Lane, D. (1997). "Subsea vehicle path planning using nonlinear programming and constructive solid geometry." *Proc., IEE Control Theory and Applications*, Vol. 144 (2), 143–152.

Warren, C. W. (1990). "A technique for autonomous underwater vehicle route planning." *IEEE J. Ocean. Eng.*, 3, 199–204.