JOURNAL OF AIRCRAFT
Vol. 47, No. 4, July–August 2010

Three-Dimensional Curvature-Constrained Trajectory Planning Based on In-Flight Waypoints

A. R. Babaei[*] and M. Mortazavi[†]
Amirkabir University of Technology, 15875-4413 Tehran, Iran

DOI: 10.2514/1.47711

This paper proposes an efficient algorithm with a novel procedure for trajectory planning of unmanned aerial vehicles in three-dimensional space. This work has been motivated by a challenge to develop a fast trajectory planning algorithm for autonomous unmanned aerial vehicles through the midcourse waypoints. The waypoints that are defined as a preflight or in-flight procedure are described in a five-dimensional configuration: the position in three dimensions plus desired crossing heading and flight-path angles. For achieving the waypoints, the Dubins path is extended to three-dimensional applications by using the geometrical concepts. In addition, the trajectory planning algorithm is represented as a set of ordinary differential equations called optimal-constrained-trajectory kinematics by applying the differential geometry concepts. Optimal-constrained-trajectory kinematic is a closed-loop guidance law that generates the guidance commands based on the waypoint configuration and minimum turning radius and is solved in a real-time manner. The proposed algorithm includes an operational framework that leads to gradually generating the smooth three-dimensional trajectory, aimed at reaching the midcourse targets and final target so that they are smoothly connected to each other. Finally, the simulation results show the capability of the algorithm in dynamic trajectory planning in low computational burden.

Nomenclature

e_v^H	=	velocity unit vector in H coordinate system
S	=	direction of turn
TM_a	=	transformation matrix from the inertial frame to the local coordinate system
TM_b	=	transformation matrix from the local coordinate system to the P coordinate system
$[T]^{\mathrm{FI}}$	=	transformation matrix from the inertial frame to the Frenet frame
t	=	flight time
V	=	total velocity vector
v	=	total velocity magnitude
$X(x, y, z)$	=	position vector
x	=	downrange
y	=	crossrange
z	=	altitude
Γ	=	angle between the T plane and the P plane
γ	=	flight-path angle
ψ	=	heading angle
Ω	=	rotational velocity vector with respect to the inertial frame
ω_j	=	jth component of rotational velocity vector with respect to the inertial frame
$1/R_{\min}$	=	maximum curvature

Subscripts

i	=	waypoint index
p	=	index for variables defined in the P coordinate system
t	=	index for variables defined in the local coordinate system
0	=	switching index from the T plane to the P plane
1	=	first switching index in the P plane
2	=	second switching index in the P plane

I. Introduction

UNMANNED aerial vehicles (UAVs) have become increasingly attractive for missions in which human presence is dangerous or difficult. Many uses of UAVs in civil, military, and commercial applications include weather and atmospheric monitoring, emergency communications, telecommunications, border patrol, battlefield deployment, etc. [1]. Recent advances in the navigation systems, the fast digital computers, and digital vision systems make it possible to develop research efforts aimed at designing autonomous systems that exhibit a high degree of reliability in their operation, in the face of an uncertain environment and various applications.

Among many open issues in the development of autonomous UAVs, the trajectory planning is a very important task that deals with the time evolution of the flight path. In the trajectory planning of UAVs, an important part is to design an optimal (or near-optimal) and feasible flight trajectory automatically to fly from one point to another. Feasible trajectories should respect the limitation of vehicle performances under specific constraint conditions. Examples of such constraints include minimum turning radius and minimum and maximum speed of the UAV. In several situations, developing practical methods for trajectory planning must be sufficiently fast and always reliable for the online use on airborne processors.

Several methods can be used to design a nominal trajectory. The current methods first use preflight defined waypoints (WPs) that are set according to the given mission [1–7]. The UAVs mission may be specified for reconnaissance or implementation of some tasks [2,3], threat avoidance [1,4,5], or obstacle avoidance [8–11]. Second, the path is further refined to make it flyable by using the curves [3–7,11–17]. Thus, the objective of the trajectory planning is not only passing from the waypoints but also satisfying the UAV performance limitations. Often, the conventional methods involve the computational load that depends on the structure of problem and its solution and capabilities of the hardware. For example, solution methods including dynamic programming [18] still require fairly powerful computers, and they may not be suitable for real-time operation. [10,11] describe the evolutionary algorithm (EA) approach to solve the path planning problem. The EA-based algorithms have disadvantages in online applications due to their slow behavior.

Received 19 October 2009; revision received 14 May 2010; accepted for publication 20 May 2010. Copyright © 2010 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved. Copies of this paper may be made for personal or internal use, on condition that the copier pay the $10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 0021-8669/10 and $10.00 in correspondence with the CCC.

[*] Ph.D. Graduate, Aerospace Engineering Department, Hafez Avenue; arbabaei@aut.ac.ir.

[†] Assistant Professor, Center of Excellence in Computational Aerospace, Aerospace Engineering Department, Hafez Avenue; mortazavi@aut.ac.ir.

In [13], it has been shown that the shortest path between two configurations satisfying the curvature constraints is composed of circles and straight-line path segments called *Dubins paths*. Thereafter, the Dubins paths were used for trajectory planning by others [14–17]. In addition, [3] has used a 2-D geometric approach with circular segments for crossing the midcourse waypoints. In [6], based on a geometric solution, tangential circular segments have been used for changing smoothly heading angle for 2-D global path that is defined by waypoints. These Dubins-path-based approaches have low computational burden and simple design procedure. But, these approaches are basically applied in 2-D applications.

In this paper, a Dubins-path-based algorithm that exploits analytical geometry and differential geometry [19] is proposed for trajectory planning of UAVs in three-dimensional space with an assumption of no wind. This work has been motivated by the challenge to develop and implement a fast trajectory planning algorithm for autonomous UAVs guidance through midcourse waypoints (as temporary targets). In general, it can be assumed that there is no prior knowledge of these waypoints, and their configuration (included 3-D position and desired crossing direction) may be computed as an in-flight procedure by onboard hardware/software systems or by ground control stations. The in-flight waypoints are generated based on the imposed missions that are not previously programmable (e.g. obstacle avoidance mission in unknown environments). In contrast, the offline uploading the waypoints is the preflight procedure. This algorithm should be able to find not only a feasible trajectory but also an optimal trajectory. The feasible trajectory is implemented to satisfy the performance constraints, and the optimality objective is to locally minimize the length of the trajectory. The trajectory planning with the shortest path can lead to less fuel consumption, longer durability, and lowest deviation from a straight line that connects the current waypoint to the next waypoint. In short, this paper provides a practical, simple, and fast procedure for designing a 3-D Dubins-path-based trajectory by following an analytical solution that leads to a closed-loop guidance method. This guidance law will generate the required commands for automatic flight control systems (AFCS) based on the WP configuration and the UAV performances.

The rest of the paper is organized as follows: Section II presents principles for the proposed trajectory planning algorithm. The definition of problem, the requirements, and important statements are presented in this section. The solution for the problem and trajectory planning algorithm are described in Sec. III. The simulation results of the trajectory planning algorithm for a defined scenario are presented in Sec. IV. Finally, conclusions are drawn in Sec. V.

II. Principles for Proposed Algorithm

In this section, principles for the proposed algorithm are discussed.

A. Problem Framework

1) After the waypoint is sent to the UAV, the trajectory planning algorithm must be able to rapidly generate the trajectory that will guide the vehicle to the waypoint configuration. In general, this waypoint can be sent to the UAV at any time of flight or preflight. It is possible that before the waypoint is passed by the UAV, it is canceled and a new waypoint is sent to the UAV to implement a new mission. In short, the proposed algorithm must be able to perform trajectory planning (based on a preflight mission) and replanning (based on an in-flight mission) in a fast manner.

2) On the way to the final target, the UAV passes through the midcourse waypoints so that the global path leads to a series of straight-line segments that performs a time-independent path. Thus, the global path is considered in terms of a list of waypoints that are described in a five-dimensional configuration $(x_i, y_i, z_i, \psi_i, \gamma_i)$: the position in three dimensions plus the desired crossing direction that is defined by the desired crossing heading and flight-path angles.

3) This path is refined into the flyable time-dependent trajectory by using curves. This task must be done by the trajectory planning algorithm. The trajectory planning algorithm must satisfy the performance constraints to produce a smooth trajectory from the start point to the target. The considered constraint includes the minimum turning radius $R_{\min}$ or the maximum curvature of the UAV flight.

4) The online trajectory planning algorithm gradually produces a smooth 3-D trajectory aimed at reaching the midcourse targets and final target. The generated trajectory consists of smaller curves (segment i) smoothly connected to each other.

5) The initial point (waypoint i) and the final point (waypoint $i+1$) configurations of each segment i can be defined via the preflight or in-flight procedure. Through the trajectory planning algorithm, we try to find not only a feasible trajectory but also an optimal trajectory from waypoint i (WP i) to waypoint $i+1$ (WP $i+1$). The objective for each segment i is to minimize the length of the trajectory. Here, the Dubins paths are extended to generate the 3-D trajectory passing through WP i and WP $i+1$. Based on the Dubins paths, the shortest trajectory between WP i and WP $i+1$ is the concatenation of circular arcs and their tangent lines. The circular arcs allow for the shortest turn possible by the UAV, satisfying the minimum turning radius constraint, and the straight line is the shortest distance between any two points. According to the results from [13,20], Dubins paths include the sufficient set of optimal paths that can be partitioned into two families: circle–circle–circle (CCC) or circle–straight-line–circle. These paths transform the configuration space into a finite number of trajectories.

Therefore, concerning points 4 and 5, the global trajectory from the start configuration to the target configuration is a continuous and piecewise C^2 path in the space $\mathbb{R}^3$. In other words, such a trajectory is formed of C^2 paths that are glued together at some commutation points, where the trajectory is C^1 [20].

In short, the list of requirements imposed by the above statements is as follows:

1) Reach the waypoint configuration.

2) Smooth the connection of all segments that are formed the global trajectory.

3) Satisfy the performance constraints by the global trajectory.

4) Consider a shortest-path problem.

5) Rapidly redesign the trajectory based on the in-flight waypoints.

B. Problem Statement

To present a simplified algorithm with aforementioned framework, the trajectory is planned with respect to a local coordinate system and then fed to the UAV through the transformation between the local coordinate system and the inertial frame. Thus, if the WP configuration is obtained with respect to the inertial frame, then it is required to transform it to the local coordinate system. The local coordinate system is defined as the flight-path coordinate system (see Fig. 1) [21] as follows:

1) The origin coincides with the initial point of each segment of trajectory (WP i).

2) The x axis points along with the velocity vector at WP i, the y axis remains in the horizontal plane (x-y plane of the inertial frame), and the z axis completes the local coordinate system.

3) The orientation is fixed with respect to the inertial frame during the each segment of the trajectory, and it is updated for new segment.

4) Two angles ψ_i and γ_i (the heading angle and flight-path angle at WP i) relate the local coordinate system to the inertial frame.

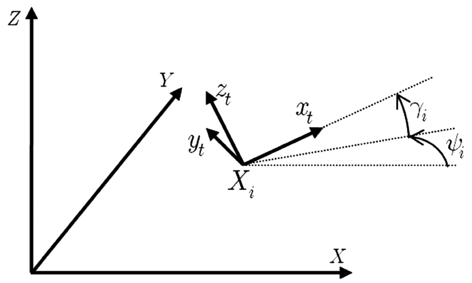

Fig. 1 Orientation of the local coordinate system with respect to the inertial frame.

The transformation between the trajectory in the local coordinate system, $X_t = (x_t, y_t, z_t)$, and the trajectory in the inertial frame, $X = (x, y, z)$, is done by the following equation:

$$X_t = \mathrm{TM}_a(X - X_i) \tag{1}$$

where $X_i = (x_i, y_i, z_i)$ is the position of the UAV at WP i, and transformation matrix TM_a is derived by two sequential transformations through the heading angle and flight-path angle at WP i (ψ_i, γ_i) as follows:

$$\begin{aligned}\mathrm{TM}_a &= T(-\gamma_i)T(\psi_i)\\ &= \begin{bmatrix} \cos\gamma_i\cos\psi_i & \cos\gamma_i\sin\psi_i & \sin\gamma_i \\ -\sin\psi_i & \cos\psi_i & 0 \\ -\sin\gamma_i\cos\psi_i & -\sin\gamma_i\sin\psi_i & \cos\gamma_i \end{bmatrix}\end{aligned} \tag{2}$$

In addition, the heading angle and the flight-path angle in the local coordinate system are derived through the following procedure. The velocity unit vector in the inertial frame and the local coordinate system are as follows:

$$\begin{aligned} e_v^I &= [\cos\gamma\cos\psi \quad \cos\gamma\sin\psi \quad \sin\gamma]^T \\ e_v^t &= [\cos\gamma_t\cos\psi_t \quad \cos\gamma_t\sin\psi_t \quad \sin\gamma_t]^T \end{aligned} \tag{3}$$

The following equation gives the relationship between these unit vectors so that we can derive heading and flight-path angles in the local coordinate system in terms of these angles in the inertial frame:

$$e_v^t = \mathrm{TM}_a e_v^I \tag{4}$$

Now, based on the local coordinate system, we have a problem with the following properties:

1) To pass from WP i and WP $i+1$, the following boundary conditions are considered:

$$x_t(0) = 0, \qquad y_t(0) = 0, \qquad z_t(0) = 0$$

$$x_t(t_{i+1}) = x_{t_{i+1}}, \qquad y_t(t_{i+1}) = y_{t_{i+1}}, \qquad z_t(t_{i+1}) = z_{t_{i+1}}$$

where the t_{i+1} parameter is the total flight time for the segment i.

2) The new segment of the trajectory must be tangentially connected with the previous segment of the trajectory. The connection point is a commutation point with C^1 continuity. Therefore, the initial heading and flight-path angles of each segment of the trajectory $[\psi_t(0)$ and $\gamma_t(0)]$ are considered equal with zero value:

$$\psi_t(0) = \gamma_t(0) = 0$$

3) Based on the desired crossing heading angle and the desired crossing flight-path angle of waypoints, the final heading and flight-path angles of each segment of the trajectory are defined as follows:

$$\psi_t(t_{i+1}) = \psi_{t_{i+1}}, \qquad \gamma_t(t_{i+1}) = \gamma_{t_{i+1}}$$

C. Problem Solution

Kinematics of the UAV in the local coordinate system is as follows:

$$\begin{cases} \dot{x}_t = v\cos\gamma_t\cos\psi_t \\ \dot{y}_t = v\cos\gamma_t\sin\psi_t \\ \dot{z}_t = v\sin\gamma_t \\ \dot{\psi}_t = u_1 \\ \dot{\gamma}_t = u_2 \end{cases} \tag{5}$$

where u_1 and u_2 are the control inputs, and v is the UAV velocity. Trajectory planning leads to the computation of u_1 and u_2 based on the minimum turning radius, WP i, and WP $i+1$. Direct optimal computations for this 3-D problem are complex and burdensome. Because of these reasons, we try to propose a three-dimensional flight strategy by enforcing UAV flight on two intersected planes called the T plane and the P plane for achieving the problem requirements (see Fig. 2). This strategy, which is not unique, has been defined based on the lowest deviation from basic Dubins paths.

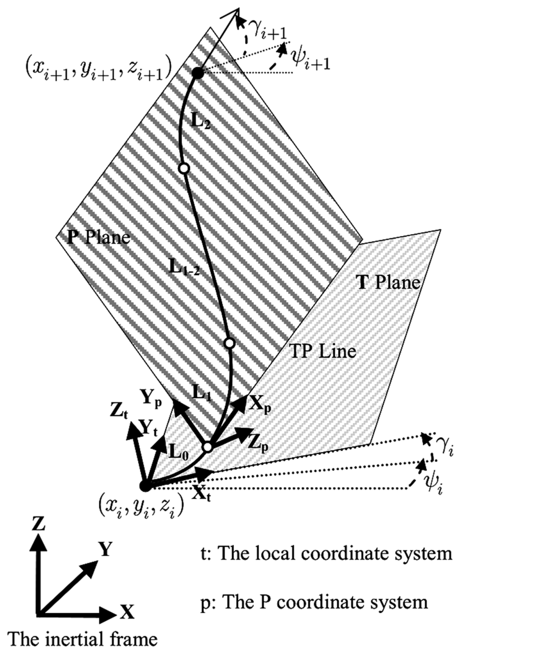

Fig. 2 Three-dimensional flight strategy is proposed by using two planes: *T* and *P*.

1) In the minor section of the trajectory, the UAV is enforced to fly in the T plane (x_t-y_t plane) by using a turning trajectory (subsegment L_0). The turning flight continues until the T plane is intersected by the P plane at time t_{t_0}. Tangentially, the P plane is intersected by the turning trajectory in the T plane (at point $X_t(t_{t_0}) = X_{t_0} = (x_{t_0}, y_{t_0}, 0)$, $\psi_t(t_{t_0}) = \psi_{t_0}$, and $\gamma_t(t_{t_0}) = 0$). It should be noted that motion (turning flight) in the T plane leads to pull-up/pull-down and turn maneuvers with respect to inertial frame. In other words, the term *turning flight* is the correct term for the local coordinate system.

2) In the major section of the trajectory, the UAV flight is done in the P plane. The trajectory in this plane is based on the Dubins paths so that it includes three subsegments: initial turning flight (subsegment L_1), straight flight (subsegment L_{1-2}), and final turning flight (subsegment L_2). The P plane is defined as follows:

Definition 1: The P plane includes points X_{t_0}, $X_{t_{i+1}}$, and $X'_{t_{i+1}}$. The point $X'_{t_{i+1}}$ is defined in the direction of the vector that is determined by the desired crossing heading and flight-path angles in the waypoints. Thus, the points X_{t_0} and $X_{t_{i+1}}$ and the desired tangent vector in WP $i+1$ are coplanar:

$$X'_{t_{i+1}} - X_{t_{i+1}} = [\cos\gamma_{t_{i+1}}\cos\psi_{t_{i+1}} \quad \cos\gamma_{t_{i+1}}\sin\psi_{t_{i+1}} \quad \sin\gamma_{t_{i+1}}]^T \tag{6}$$

Definition 2: Tangentially, the P plane is intersected by the subsegment L_0. Therefore, the P plane includes the vector directed from X_{t_0} to X'_{t_0}:

$$X'_{t_0} - X_{t_0} = [\cos\psi_{t_0} \quad \sin\psi_{t_0} \quad 0]^T \tag{7}$$

Certainly, passing the P plane from points $X_{t_{i+1}}$, $X'_{t_{i+1}}$, X_{t_0}, and X'_{t_0} specifies the successful trajectory planning.

According to Definition 1 and the definition of plane in analytical geometry, the equation of the P plane is derived as follows:

$$\begin{vmatrix} x_t - x_{t_0} & y_t - y_{t_0} & z_t \\ x_{t_{i+1}} - x_{t_0} & y_{t_{i+1}} - y_{t_0} & z_{t_{i+1}} \\ x'_{t_{i+1}} - x_{t_0} & y'_{t_{i+1}} - y_{t_0} & z'_{t_{i+1}} \end{vmatrix} = 0 \tag{8}$$

or

$$ax_t + by_t + cz_t = d \tag{9}$$

where

$$a = z'_{t_{i+1}}(y_{t_{i+1}} - y_{t_0}) - z_{t_{i+1}}(y'_{t_{i+1}} - y_{t_0})$$
$$b = z_{t_{i+1}}(x'_{t_{i+1}} - x_{t_0}) - z'_{t_{i+1}}(x_{t_{i+1}} - x_{t_0})$$
$$c = (x_{t_{i+1}} - x_{t_0})(y'_{t_{i+1}} - y_{t_0}) - (x'_{t_{i+1}} - x_{t_0})(y_{t_{i+1}} - y_{t_0})$$
$$d = ax_{t_0} + by_{t_0}$$

In addition, the angle between the T plane and the P plane (Γ) is computed by using the following equation:

$$\Gamma = \arccos\left(\frac{\mathbf{N}_T \cdot \mathbf{N}_P}{|\mathbf{N}_T||\mathbf{N}_P|}\right) \tag{10}$$

where $\mathbf{N}_T = (0, 0, 1)$ and $\mathbf{N}_P = (a, b, c)$ are the normal vectors of the T plane and the P plane.

III. Trajectory Planning Algorithm

In the previous section, the general solution of problem was presented by definition of a near-optimal flight strategy. Here, details of the proposed trajectory planning algorithm is presented through the following steps:

A. Trajectory Planning in the *T* Plane

It is required to find a point that specifies the switching position from the T plane to the P plane X_{t_0}. The UAV flight in the T plane is implemented through a relative turning trajectory. The kinematics of the UAV with respect to the local coordinate system (in the T plane) is presented in the following equations:

$$\dot{x}_t = v\cos\psi_t, \qquad \dot{y}_t = v\sin\psi_t, \qquad \dot{\psi}_t = S_0 c, \qquad c = v/R_{\min} \tag{11}$$

where the UAV is assumed to be flying at a constant velocity. The S_0 parameter that is 1 or -1 represents the direction of turn in T plane (right or left).

The UAV trajectory in the T plane (subsegment L_0) is obtained by forward integration of Eq. (11) from (0, 0, 0) to $(x_{t_0}, y_{t_0}, \psi_{t_0})$:

$$\begin{cases} x_{t_0} = S_0 R_{\min}\mathrm{Sin}\psi_{t_0} \\ y_{t_0} = S_0 R_{\min}(1 - \mathrm{Cos}\psi_{t_0}) \end{cases} \tag{12}$$

By using Eq. (9) for point $X_t = X'_{t_0}$ (concerning Definition 2) and Eqs. (6), (7), and (12), the following equation is achieved:

$$A\sin\psi_{t_0} + B\cos\psi_{t_0} = D \tag{13}$$

where

$$A = z_{t_{i+1}}\cos\gamma_{t_{i+1}}\cos\psi_{t_{i+1}} - x_{t_{i+1}}\sin\gamma_{t_{i+1}}$$
$$B = (y_{t_{i+1}} - S_0 R_{\min})\sin\gamma_{t_{i+1}} - z_{t_{i+1}}\cos\gamma_{t_{i+1}}\sin\psi_{t_{i+1}}$$
$$D = -S_0 R_{\min}\sin\gamma_{t_{i+1}}$$

The solution of Eq. (13) is as follows:

$$\psi_{t_0} = \arctan\left(\frac{AD \pm B\sqrt{A^2 + B^2 - D^2}}{BD \mp A\sqrt{A^2 + B^2 - D^2}}\right) \tag{14}$$

The switching position (x_{t_0}, y_{t_0}) is computed by using Eq. (12), and the flight time corresponding to this position is derived by the following equation:

$$t_{t_0} = |\psi_{t_0}/(S_0 c)| \tag{15}$$

B. Definition of *P* Coordinate System

Once the subsegment L_0 is traversed, the UAV starts its motion in the P plane. Therefore, before the investigation of the trajectory in this plane, the P coordinate system is defined as follows:

1) The origin is X_{t_0}.

2) The x axis (x_p) is directed along the intersection of the T plane and the P plane (TP line in Fig. 2) with direction ψ_{t_0} with respect to the x_t axis.

3) The z axis (z_p) is normal to the P plane.

4) The x_p-y_p plane is coplanar with the P plane.

By using the P coordinate system, the trajectory planning in the P plane can be reduced to the two-dimensional trajectory planning problem. The kinematics of the UAV with respect to the P coordinate system (in the P plane) is presented through the following equations:

$$\dot{x}_p = v\cos\psi_p, \qquad \dot{y}_p = v\sin\psi_p, \qquad \dot{\psi}_p = S_i c \tag{16}$$

where the parameter S_i that is 1, -1, or 0 (based on the definition of Dubins paths) represents the direction of turn in the P coordinate system, and (x_p, y_p, ψ_p) is the UAV trajectory configuration in the P coordinate system.

The transformation between the trajectory in the local coordinate system and the trajectory in the P coordinate system $X_p = (x_p, y_p, 0)$ is required and done by the following equation:

$$X_p = \mathrm{TM}_b(X_t - X_{t_0}) \tag{17}$$

where the transformation matrix TM_b is derived by the two sequential transformations through angles ψ_{t_0} and Γ as follows:

$$\mathrm{TM}_b = T(\Gamma)T(\psi_{t_0}) = \begin{bmatrix} \cos\psi_{t_0} & \sin\psi_{t_0} & 0 \\ -\sin\psi_{t_0}\cos\Gamma & \cos\psi_{t_0}\cos\Gamma & \sin\Gamma \\ -\sin\psi_{t_0}\sin\Gamma & -\cos\psi_{t_0}\sin\Gamma & \cos\Gamma \end{bmatrix} \tag{18}$$

The heading angle in the P coordinate system is derived from the following procedure (the flight-path angle in the P coordinate system is zero due to two-dimensional flight in the P plane). The velocity unit vector in the P coordinate system is as follows:

$$e_v^p = [\cos\psi_p \quad \sin\psi_p \quad 0]^T \tag{19}$$

The heading angle in the P coordinate system is derived by using the following equation:

$$e_v^p = \mathrm{TM}_b e_v^t \tag{20}$$

Now, after the determination of switching position X_{t_0}, we have a trajectory planning problem in the P coordinate system with the following properties:

1) For passing from WP i and WP $i + 1$, the following boundary conditions are considered:

$$x_p(0) = 0, \qquad y_p(0) = 0$$
$$x_p(t_{p_{i+1}}) = x_{p_{i+1}}, \qquad y_p(t_{p_{i+1}}) = y_{p_{i+1}}$$

where the $t_{p_{i+1}}$ parameter is the total flight time in the P plane.

2) The subsegment L_0 in the T plane must be tangentially connected with the subsegment L_1 in the P plane. Therefore, the heading angle in the initial point of the subsegment L_1 in the P coordinate system $[\psi_p(0)]$ is considered equal with the zero value:

$$\psi_p(0) = 0$$

3) Based on the desired crossing heading angle of waypoints, the final heading angle of each segment of the trajectory in the P coordinate system is defined as follows:

$$\psi_p(t_{p_{i+1}}) = \psi_{p_{i+1}}$$

C. Trajectory Planning in the *P* Plane

Most of the flight is considered through the P plane. The trajectory in this plane that is based on the Dubins paths is divided to three smoothly concatenated parts. Therefore, the parameter S_i in Eq. (16) can be presented as follows:

$$S_i = \begin{cases} S_1 = 1 \quad \text{or} \quad -1 & \text{if } 0 \le x_p \le x_{p_1} \\ 0 & \text{if } x_{p_1} < x_p < x_{p_2} \\ S_2 = 1 \quad \text{or} \quad -1 & \text{if } x_{p_2} \le x_p \le x_{p_{i+1}} \end{cases} \tag{21}$$

where the parameters x_{p_1} and x_{p_2} are switching positions from the initial turning flight (subsegment L_1) to the straight flight (subsegment L_{1-2}) and the straight flight to the final turning flight (subsegment L_2) in the P plane, respectively. Configuration of the trajectory in the P plane is illustrated in Fig. 3. Equation (21) is used by ignoring CCC-type paths so that the Dubins paths are reduced to four possible states. The CCC-type paths may be created for only very short trajectories compared with the maximum curvature.

The programming of the trajectory based on the downrange variable is an advantage, because we can directly use the output of the navigation system. The downrange variable x is the output of the navigation system that can be transformed to the local coordinate system (x_t) by Eq. (1) and to the P coordinate system (x_p) by Eqs. (1) and (17). The subsegments L_1, L_{1-2}, and L_2 are obtained by forward integration of Eq. (16) are as follows:

1) For subsegment L_1 ($0 \le x_p \le x_{p_1}$), the following equations are derived by using forward integration of Eq. (16) from (0, 0, 0) to ($x_{p_1}, y_{p_1}, \psi_{p_1}$):

$$\begin{cases} x_{p_1} = S_1 R_{\min} \mathrm{Sin}\psi_{p_1}(t) \\ y_{p_1} = S_1 R_{\min}(1 - \mathrm{Cos}\psi_{p_1}(t)) \end{cases} \tag{22}$$

where position (x_{p_1}, y_{p_1}) is the final point of the subsegment L_1 (first switching position in the P plane), and angle ψ_{p_1} is change in the heading angle due to the motion along the subsegment L_1.

2) Subsegment L_{1-2} ($x_{p_1} < x_p < x_{p_2}$) has length L and is aligned at heading angle ψ_{p_1}. Thus,

$$\begin{cases} x_{p_2} - x_{p_1} = L\mathrm{Cos}\psi_{p_1} \\ y_{p_2} - y_{p_1} = L\mathrm{Sin}\psi_{p_1} \end{cases} \tag{23}$$

3) For subsegment L_2 ($x_{p_2} \le x_p \le x_{p_{i+1}}$), the following equations are derived by using the forward integration of Eq. (16) from ($x_{p_2}, y_{p_2}, \psi_{p_1}$) to ($x_{p_{i+1}}, y_{p_{i+1}}, \psi_{p_{i+1}}$):

$$\begin{cases} x_{p_{i+1}} - x_{p_2} = S_2 R_{\min}(\mathrm{Sin}\psi_{p_{i+1}} - \mathrm{Sin}\psi_{p_1}) \\ y_{p_{i+1}} - y_{p_2} = -S_2 R_{\min}(\mathrm{Cos}\psi_{p_{i+1}} - \mathrm{Cos}\psi_{p_1}) \end{cases} \tag{24}$$

where ($x_{p_2}, y_{p_2}, \psi_{p_1}$) is configuration of the initial point on the subsegment L_2 (second switching configuration on the P plane).

The total displacements in the x direction ($x_{p_{i+1}}$) and the y direction ($y_{p_{i+1}}$) are computed by using the following equations:

$$\begin{cases} x_{p_{i+1}} = x_{p_1} + (x_{p_2} - x_{p_1}) + (x_{p_{i+1}} - x_{p_2}) \\ y_{p_{i+1}} = y_{p_1} + (y_{p_2} - y_{p_1}) + (y_{p_{i+1}} - y_{p_2}) \end{cases} \tag{25}$$

By using Eqs. (22–25), we can obtain the following equations:

$$\begin{cases} x_{p_{i+1}} = S_1 R_{\min}\mathrm{Sin}\psi_{p_1} + L\mathrm{Cos}\psi_{p_1} + S_2 R_{\min}(\mathrm{Sin}\psi_{p_{i+1}} - \mathrm{Sin}\psi_{p_1}) \\ y_{p_{i+1}} = S_1 R_{\min}(1 - \mathrm{Cos}\psi_{p_1}) + L\mathrm{Sin}\psi_{p_1} \\ \quad - S_2 R_{\min}(\mathrm{Cos}\psi_{p_{i+1}} - \mathrm{Cos}\psi_{p_1}) \end{cases} \tag{26}$$

or

$$\begin{cases} (S_1 - S_2)R_{\min}\mathrm{Sin}\psi_{p_1} + L\mathrm{Cos}\psi_{p_1} = f_1 \\ -(S_1 - S_2)R_{\min}\mathrm{Cos}\psi_{p_1} + L\mathrm{Sin}\psi_{p_1} = f_2 \end{cases} \tag{27}$$

where

$$\begin{cases} f_1 = x_{p_{i+1}} - S_2 R_{\min}\mathrm{Sin}\psi_{p_{i+1}} \\ f_2 = y_{p_{i+1}} + S_2 R_{\min}\mathrm{Cos}\psi_{p_{i+1}} - S_1 R_{\min} \end{cases} \tag{28}$$

The parameters f_1 and f_2 include the waypoint configuration. Therefore, to solve the optimal control problem, it is sufficient to solve the nonlinear equations (27). The set of Eqs. (27) has two unknown parameters ψ_{p_1} and L that lead to determination of the first and second switching positions by using Eqs. (22) and (23). The closed-form solution of Eq. (27) is as follows:

$$\begin{aligned} \psi_{p_1} &= \arctan\left(\frac{f_1 \pm f_2\sqrt{f_1^2 + f_2^2 - (S_1 - S_2)^2 R_{\min}}}{-f_2 \pm f_1\sqrt{f_1^2 + f_2^2 - (S_1 - S_2)^2 R_{\min}}}\right) \\ L &= \frac{f_1 + (S_1 - S_2)R_{\min}\sin\psi_{p_1}}{\cos\psi_{p_1}} \end{aligned} \tag{29}$$

Subsequently, the time t_p and the length l_p of the trajectory corresponding to positions x_{p_1}, x_{p_2}, and $x_{p_{i+1}}$ (namely, t_{p_1}, l_{p_1}, t_{p_2}, l_{p_2}, $t_{p_{i+1}}$, and $l_{p_{i+1}}$) are calculated as follows:

$$\begin{cases} \psi_p(t_p) = S_1 c t_p \Rightarrow t_{p_1} = \left|\frac{\psi_{p_1}}{S_1 c}\right|, & l_{p_1} = R_{\min}|\psi_{p_1}| \\ x_{p_2} - x_{p_1} = v\cos\psi_{p_1}(t_{p_2} - t_{p_1}) \Rightarrow t_{p_2} = t_{p_1} + \frac{L}{v}, & l_{p_2} = l_{p_1} + L \\ \psi_{p_{i+1}} = \psi_{p_1} + S_2 c(t_{p_{i+1}} - t_{p_2}) \Rightarrow t_{p_{i+1}} = t_{p_2} + \left|\frac{\psi_{p_{i+1}} - \psi_{p_1}}{S_2 c}\right|, & l_{p_{i+1}} = l_{p_2} + R_{\min}|\psi_{p_{i+1}} - \psi_{p_1}| \end{cases} \tag{30}$$

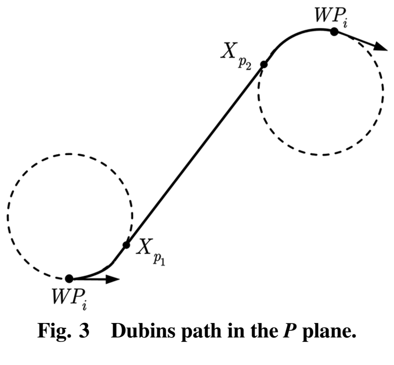

Fig. 3 Dubins path in the *P* plane.

D. Limitations of the Algorithm

According to Eq. (29), the algorithm has no solution for the following condition:

$$f_1^2 + f_2^2 - (S_1 - S_2)^2 R_{\min} < 0 \tag{31}$$

In other words, for given ($x_{p_{i+1}}, y_{p_{i+1}}$), there is an unavailable region with lower bound ψ_{p_a} and upper bound ψ_{p_b} that satisfy condition (31). In short, for given

$$(x_{p_{i+1}}, y_{p_{i+1}}) \quad \forall\ \psi_{p_{i+1}} \in [\psi_{p_a}, \psi_{p_b}]$$

the algorithm has no solution. It is clear that condition (31) never creates for Dubins paths with $S_1 = S_2$ and sufficiently distant

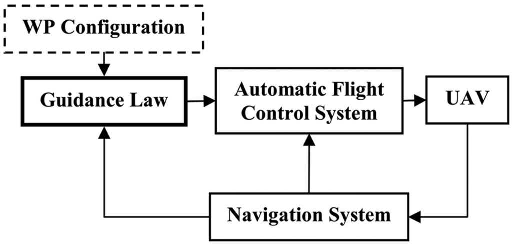

Fig. 4 Block diagram of an autonomous UAV.

waypoints compared with maximum curvature. The paths that lead to satisfying condition (31) are ignored, and the path with the other S_1 and S_2 is replaced.

E. Guidance Law

For real-time implementation of the trajectory planning algorithm, a guidance law with an operational form is required to generate the guidance commands that are compatible with AFCS. Figure 4 represents the block diagram for an autonomous UAV. As already noted, the focus of this paper is the guidance law block. The guidance law generates the commands that lead to the curvature-constrained trajectory from the start configuration to the target configuration. Subsequently, it has been attempted to represent the trajectory planning in the form of a closed-loop guidance law.

Three switching positions (X_{t_0}, X_{p_1}, and X_{p_2}) have been computed in terms of the waypoint configuration (WP $i+1$). It is required to transform these positions to the inertial frame by using the following equations:

$$\begin{cases} X_0 = \mathrm{TM}_a^T X_{t_0} + X_i \\ X_1 = \mathrm{TM}_a^T \mathrm{TM}_b^T X_{p_1} + \mathrm{TM}_a^T X_{t_0} + X_i \\ X_2 = \mathrm{TM}_a^T \mathrm{TM}_b^T X_{p_2} + \mathrm{TM}_a^T X_{t_0} + X_i \end{cases} \tag{32}$$

In addition, the rotational velocity vector of the Frenet frame corresponding to motion of the UAV along the proposed Dubins-path-based trajectory is as follows:

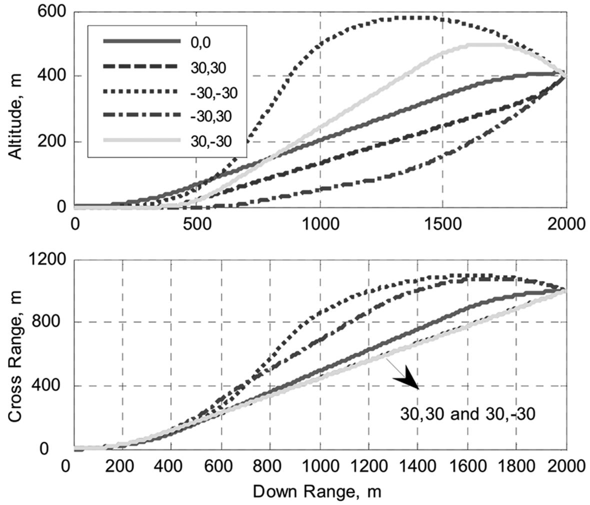

Fig. 6 Crossrange and altitude in terms of downrange for the 3-D trajectories illustrated in the Fig. 5.

$$\Omega = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{cases} S_0 \mathrm{TM}_a^T C & \text{if } x_i \le x \le x_0 \\ S_1 \mathrm{TM}_a^T \mathrm{TM}_b^T C & \text{if } x_0 \le x \le x_1 \\ 0 & \text{if } x_1 < x < x_2 \\ S_2 \mathrm{TM}_a^T \mathrm{TM}_b^T C & \text{if } x_2 \le x \le x_{i+1} \end{cases} \tag{33}$$

where $C = [0 \quad 0 \quad c]^T$ is the rotational velocity vector of the Frenet frame in the Frenet frame. A Frenet frame [19] is a moving frame for which the x axis is directed along instantaneous velocity vector, the y axis is directed along instantaneous curvature vector, and the z axis is normal to the instantaneous motion plane. Angles ψ, γ, and φ relate Frenet frame to inertial frame. The angles ψ and γ represent the orientation of the UAV velocity, and the additional angle φ is a complementary angle for the instantaneous motion plane. Therefore, transformation from inertial frame to Frenet frame is done through three sequential rotations as follows:

$$[T]^{\mathrm{FI}} \equiv T(\varphi)T(-\gamma)T(\psi) = \begin{bmatrix} \cos\gamma\cos\psi & \sin\psi\cos\gamma & \sin\gamma \\ -\cos\psi\sin\gamma\sin\varphi - \sin\psi\cos\varphi & -\sin\psi\sin\gamma\sin\varphi + \cos\psi\cos\varphi & \cos\gamma\sin\varphi \\ -\cos\psi\sin\gamma\cos\varphi + \sin\psi\sin\varphi & -\sin\psi\sin\gamma\cos\varphi - \cos\psi\sin\varphi & \cos\gamma\cos\varphi \end{bmatrix} \tag{34}$$

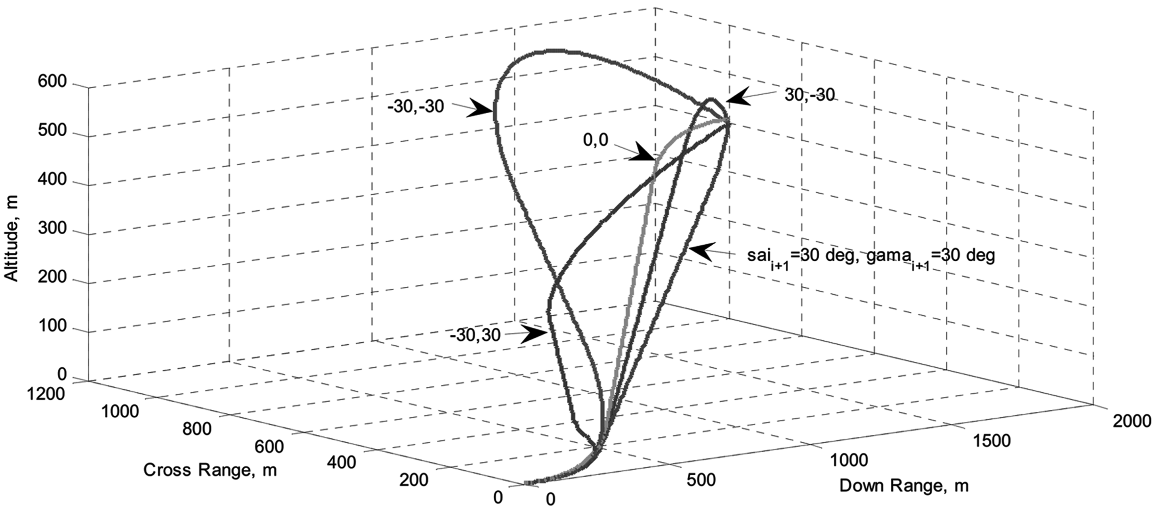

Fig. 5 Generated trajectories by the algorithm for several crossing heading and flight-path angles.

Table 1 Mission properties that can be received by the UAV in the online procedure

	X position, m	Y position, m	Z position, m	Heading angle, deg	Flight-path angle, deg
Initial WP	0	0	0	0	0
WP 1	2000	1000	500	−10	−10
WP 2	4000	1500	100	−15	0
WP 3	6000	500	300	−15	−10
WP 4	8000	1500	600	−20	−20
WP 5	10,000	0	100	0	−30

The rotational velocity vector of the Frenet frame in the Frenet frame is computed by the following relationship:

$$[T]^{\mathrm{FI}}\Omega = \begin{bmatrix} \dot{\varphi} \\ 0 \\ 0 \end{bmatrix} + T(\varphi)\begin{bmatrix} 0 \\ -\dot{\gamma} \\ 0 \end{bmatrix} + T(\varphi)T(-\gamma)\begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} \tag{35}$$

Expanding this equation leads to the following ordinary differential equations:

$$\begin{cases} \dot{\varphi} = \omega_x - (\omega_y \sin\varphi + \omega_z \cos\varphi)\tan\gamma \\ \dot{\gamma} = -\omega_y \cos\varphi + \omega_z \sin\varphi \\ \dot{\psi} = (\omega_y \sin\varphi + \omega_z \cos\varphi)\frac{1}{\cos\gamma} \end{cases} \tag{36}$$

In addition to these equations, the following complementary equations are required to have others commands such as altitude command:

$$\begin{cases} \dot{x} = v\cos\gamma\cos\psi \\ \dot{y} = v\cos\gamma\sin\psi \\ \dot{z} = v\sin\gamma \end{cases} \tag{37}$$

The ordinary differential equations (36) and (37) [as the optimal-constrained-trajectory kinematics (OCTK) of UAVs] perform a closed-loop guidance law and are solved in the real-time manner, then some variables such as z and ψ (called the command altitude and the command heading angle) can be commanded to AFCS. The real-time implementation of OCTK leads to the shortest trajectory that can guide the UAV to waypoint configuration.

F. Trajectory Planning Algorithm

The trajectory planning algorithm is summarized as follows:

1) Receive the waypoint configuration (as a temporary target that is defined with respect to the inertial frame), $(x_{i+1}, y_{i+1}, z_{i+1}, \psi_{i+1}, \gamma_{i+1})$.

2) Based on the given UAV configuration at the initial point of segment i, $(x_i, y_i, z_i, \psi_i, \gamma_i)$, transform the waypoint configuration to the local coordinate system, $(x_{t_{i+1}}, y_{t_{i+1}}, z_{t_{i+1}}, \psi_{t_{i+1}}, \gamma_{t_{i+1}})$, using Eqs. (1) and (4).

3) Calculate the switching configuration from the T plane to the P plane, $(x_{t_0}, y_{t_0}, \psi_{t_0})$, using Eqs. (12) and (14).

4) Transform the waypoint configuration to the P coordinate system, $(x_{p_{i+1}}, y_{p_{i+1}}, \psi_{p_{i+1}})$, using Eqs. (17) and (20).

5) Find ψ_{p_1}, L, S_1, and S_2 corresponding to the shortest trajectory between four possible states for S_1 and S_2 (1 and 1, 1 and −1, −1 and 1, and −1 and −1) by using Eqs. (29) and (30).

6) Calculate the switching positions in the P plane (X_{p_1} and X_{p_2}) by using Eqs. (22) and (23).

7) Transform the switching positions X_{t_0}, X_{p_1}, and X_{p_2} to the inertial frame by using Eq. (32).

8) In the real-time manner, implement the strategy presented in Eqs. (33), (36), and (37) to generate the optimal trajectory.

9) If the UAV reaches the waypoint configuration, finish the algorithm; otherwise, go to step 8.

IV. Simulation Results

To verify the performance of the flight trajectory planning methodology, the numerical simulation is performed. Here, the velocity of the UAV and the maximum load factor, respectively, are considered 120 m/s and 2. On the other hand, the minimum turning radius is 735 m.

Figures 5 and 6 illustrate the trajectory planning algorithm results for $X_{i+1}(2000, 1000, 400)$ m and several crossing heading and flight-path angles. The results show that any feasible trajectory can be generated for given waypoint configuration. These trajectories are constructed by positive turn in the T plane, positive turn in the P plane, straight flight, and negative/positive turn in the P plane.

To verify the algorithm, it is required to define some missions through several waypoints. The mission properties of the UAV are listed in Table 1. This mission may be uploaded in UAV before the flight and may be updated during the flight to implement new missions. Therefore, it is possible to change the waypoints corresponding to the new mission after the flight has begun. The

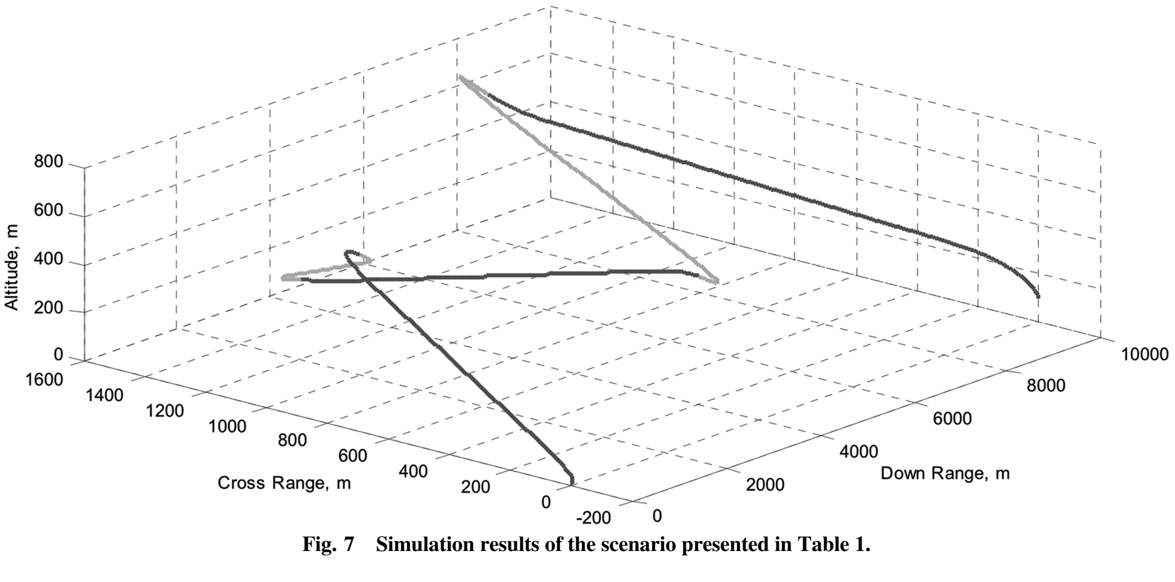

Fig. 7 Simulation results of the scenario presented in Table 1.

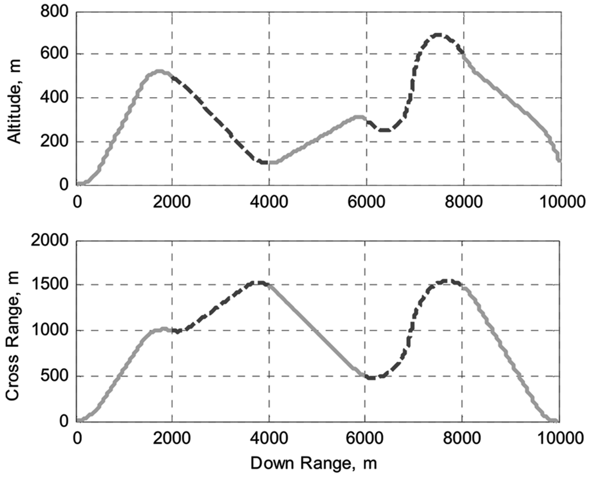

Fig. 8 Crossrange and altitude in terms of downrange for the 3-D trajectory illustrated in the Fig. 7.

simulation results are illustrated in Figs. 7 and 8. These figures show that the imposed missions can successfully be implemented. In the implementation of the missions, the flight trajectory has been constrained by the maximum curvature. The trajectory planning algorithm leads to the piecewise C^2 trajectory so that all segments 1 to 5 have been tangentially (C^1) connected. The processing time of the trajectory planning algorithm as one of the performance criteria must be investigated. This time must be low so that the online capability is not faced with difficulty. However, the trajectory planning algorithm has been run in the MATLAB, but this time is a few seconds for any segment, and so it is reliable for use by the online airborne processors.

V. Conclusions

In this paper, three-dimensional trajectory planning of unmanned aerial vehicles is implemented through the preflight or in-flight waypoints-based mission via a new algorithm. The proposed algorithm includes several important contributions:

1) The application of the Dubins paths has been extended to the 3-D space.

2) By following some geometrical concepts, a closed-form optimal solution has been derived so that the trajectory planning can be rapidly done while the vehicle is moving for previously unprogrammable missions.

3) The trajectory planning has been presented in the form of the closed-loop guidance law that generates commands for AFCS in terms of the waypoint configuration and the allowable maximum curvature.

The simulation results show the capabilities of the algorithm. Consequently, due to the simplicity and the operational structure of this algorithm, it is recommended for dynamic environments. Modification of the algorithm in the presence of wind can be considered for future research.

References

[1] Gu, D. W., Kamal, W., and Postlethwaite, I., "A UAV Waypoint Generator," AIAA 1st Intelligent Systems Technical Conference, Chicago, AIAA Paper 2004-6227, 20–22 Sept. 2004.

[2] Baralli, F., Pollini, L., and Innocenti, M., "Waypoint-Based Fuzzy guidance for Unmanned Aircraft A new Approach," AIAA Guidance, Navigation, and Control Conference and Exhibit, Monterey, CA, AIAA Paper 2002-4993, 5–8 Aug. 2002.

[3] Jorris, T. R., and Cobb, R. G., "Multiple Method 2-D Trajectory Optimization Satisfying Waypoints and No-Fly Zone Constraints," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 3, 2008, pp. 543–553.
doi:10.2514/1.32354

[4] Chandler, P. R., Rasmussen, S., and Pachter, M., "UAV Cooperative Path Planning," AIAA Guidance, Navigation, and Control Conference and Exhibit, Denver, CO, AIAA Paper 2000-4370, 14–17 Aug. 2000.

[5] Judd, K. B., and McLain, T. W., "Spline Based Path planning for Unmanned Air Vehicles," AIAA Guidance, Navigation, and Control Conference and Exhibit, Montreal, AIAA Paper 2001-4238, 6–9 Aug. 2001.

[6] Anderson, E. P., Beard, R. W., and McLain, T. W., "Real Time Dynamic Trajectory Smoothing for Unmanned Aerial Vehicles," *IEEE Transactions on Control Systems Technology*, Vol. 13, No. 3, 2005, pp. 471–477.
doi:10.1109/TCST.2004.839555

[7] Moon, G., and Kim, Y., "Optimum Flight Path Design Passing Through Waypoints for Autonomous Flight Control System," AIAA Guidance, Navigation, and Control Conference and Exhibit, Austin, TX, AIAA Paper 2003-5334, 11–14 Aug. 2003.

[8] Corbets, J. B., and Langelaan, J. W., "Parameterized Optimal Trajectory Generation for Target Localization," AIAA Guidance, Navigation, and Control Conference and Exhibit, Hilton Head, SC, AIAA Paper 2007-6750, 20–23 Aug. 2007.

[9] Frazzoli, E., Dahleh, M. A., and Feron, E., "Real-Time Motion Planning for Agile Autonomous Vehicles," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 1, 2002, pp. 116–129.
doi:10.2514/2.4856

[10] Zheng, C., Li, L., Xu, F., Sun, F., and Ding, M., "Evolutionary Rout Planner for Unmanned Air Vehicles," *IEEE Transactions on Robotics and Automation*, Vol. 21, No. 4, 2005, pp. 609–620.
doi:10.1109/TRO.2005.844684

[11] Nikolos, I. K., Valavanis, K. P., Tsourveloudis, N. C., and Kostaras, A. N., "Evolutionary Algorithm Based Offline/Online Path Planner for UAV Navigation," *IEEE Transactions on Systems, Man, and Cybernetics. Part B. Cybernetics*, Vol. 33, No. 6, 2003, pp. 898–912.
doi:10.1109/TSMCB.2002.804370

[12] Coté, K., "Complex 3-D Flight Trajectory Generation and Tracking Using Cubic Spline," AIAA 1st Technical Conference and Workshop on Unmanned Aerospace Vehicles, Portsmouth, VA, AIAA Paper 2002-3498, 20–23 May 2002.

[13] Dubins, L. E., "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, 1957, pp. 497–516.
doi:10.2307/2372560

[14] Chitsaz, H., and LaValle, S. M., "Time-Optimal Paths for a Dubins Airplane," *46th IEEE Conference on Decision and Control*, Inst. of Electrical and Electronics Engineers, Piscataway, NJ, 12–14 Dec. 2007, pp. 2379–2384.

[15] Agarwal, P. K., Biedl, T., Sylvian, L., Robbins, S., Suri, S., and Sue, W., "Curvature-Constrained Shortest Paths in a Convex Polygon," *Proceedings of the Fourteenth Annual Symposium on Computational Geometry*, ACM, New York, 1998, pp. 392–401.

[16] Shanmugavel, M., Tsourdos, A., Żbikowski, R., and White, B. A., "3-D Dubins Sets Based Coordinated Path Planning for Swarm of UAVs," AIAA Guidance, Navigation, and Control Conference and Exhibit, Keystone, CO, AIAA Paper 2006-6211, 21–24 Aug. 2006.

[17] Shanmugavel, M., Tsourdos, A., Żbikowski, R., and White, B. A., "Path Planning for Multiple UAVs Using Dubins Sets," AIAA Guidance, Navigation, and Control Conference and Exhibit, San Francisco, AIAA Paper 2005-5827, 15–18 Aug. 2005.

[18] Sun, W., and Yuan, Y. X., *Optimization Theory and Methods, Nonlinear Programming*, Springer, New York, 2006.

[19] Farin, G., "Differential Geometry I," *Curves and Surfaces for Computer Aided Geometric Design*, 4 ed., Academic Press, New York, 1997, pp. 171–179.

[20] Boissonnat, J., Cerozo, A., and Leblond, J., "Shortest Paths of Bounded Curvature in the Plane," *Proceedings of the IEEE International Conference on Robotics and Automation*, Inst. of Electrical and Electronics Engineers, Piscataway, NJ, May 1992, pp. 2315–2320.

[21] Zipfel, P. H., "Frames and Coordinate Systems," *Modeling and Simulation of Aerospace Vehicle Dynamics*, 2nd ed., AIAA, Reston, VA, 2007, pp. 55–86.