Path Following for Unmanned Combat Aerial Vehicles Using Three-Dimensional Nonlinear Guidance

Zian Wang, Zheng Gong, Jinfa Xu, Jin Wu, and Ming Liu, *Senior Member, IEEE*

***Abstract*—An effective path-following guidance algorithm for unmanned combat aerial vehicles by pursuing a look-ahead target-point along the desired path is devised. The concept of look-ahead pursuit for 3-D path-following was extended from the planar $\mathcal{L}_1$ guidance, which has been widely adopted in the application of lateral guidance of fixed-wing drone. Decoupled implementation of the planar guidance action on each of the longitudinal (vertical) and lateral (horizontal) planes supplies longitudinal-lateral acceleration commands in two perpendicular planes which is well accepted by aviation engineers. The computation of the look-ahead point utilizes a finite and iterative numerical search method by introducing an auxiliary path-related parameter. Furthermore, additional implementation-specific details such as the modification of adaptive $\mathcal{L}_1$ length and the protection strategies ensuring wide applicability of the algorithm indeed guarantee certain necessity to be considered in real implementation of the guidance law. The improved schemes eliminate complicated coordinate transformations and are built on the path-independent assumption, thus making them easier to be adjusted between various types of paths, including discrete optimal trajectories. Monte Carlo simulations and real-world flight tests have confirmed the effectiveness of the nonlinear guidance algorithm.**

***Index Terms*—Flight tests, look-ahead pursuit guidance law, 3-D path-following, unmanned combat aerial vehicles.**

Manuscript received March 22, 2021; revised July 20, 2021; accepted August 26, 2021. Recommended by Technical Editor Y. Shtessel and Senior Editor N. G. Tsagarakis. This work was supported in part by the General Research Fund of Research grants Council Hong Kong under Grant 11210017 and in part by the Early Career Scheme Project of Research Grants Council Hong Kong under Grant 21202816. *(Corresponding author: Zheng Gong.)*

Zian Wang, Zheng Gong, and Jinfa Xu are with the Department of Aerospace Engineering, College of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China (e-mail: wangzian@nuaa.edu.cn; matthewzhenggong@nuaa.edu.cn; xjfae@nuaa.edu.cn).

Jin Wu and Ming Liu are with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong, China (e-mail: jin_wu_uestc@hotmail.com; eelium@ust.hk).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TMECH.2021.3110262.

Digital Object Identifier 10.1109/TMECH.2021.3110262

I. Introduction

MOST autopilot systems are limited to simple straight-line and circular-arc tracking functions, which restrict their freedom during flight missions. With the rapid development of intelligent planning technology for unmanned combat aircraft vehicles (UCAVs), emerging autopilot technologies must be suitable for complex missions. A key component of autopilot technology is the guidance law ensuring that the vehicle can precisely track complex maneuvering trajectories and perform dynamic target-tracking tasks. This article describes the design of a simple, real time, and precise 3-D path-following guidance law for UCAVs. To facilitate a flight test, the flight control system (FCS) considered in this article is separated into guidance and control loops. The inner loop is organized so as to follow the acceleration commands generated by the guidance loop. Primitive paths can be organized as straight-line, circular-arc, and parameterized curves generated by a path planner according to the specific mission, physical environment, and constraints on the vehicle's dynamics. Moreover, in scenarios involving combat, the guidance law can also be combined with an online, real-time trajectory generator under stringent constraints.

Various studies have investigated 2-D path-following strategies [1]. The literature on 3-D path-following is quite limited. The conventional proportional-integral-derivative controllers [2], [3] derived from small perturbation theory perform well in the case of small heading/cross-track errors in the linear domain, but are poor at tracking complex maneuvering paths. Other, nonlinear, approaches to this problem can be summarized as follows:

1) Optimal offset-free design [4]–[6].
2) Vector field-based design [7].
3) Error-elimination-based design [8]–[10].
4) Virtual target-based design [11]–[14].

The weaknesses of the optimal offset-free and the error regulation approaches are that the form of guidance commands is model-dependent and comparatively complicated, and is hard to implement into the autopilot for 3-D path-following. Most vector-field-based methods proposed for 3-D path solutions are only for certain types of curves such as a straight-line or a circular-arc, and are not applicable to general 3-D curves. Virtual target algorithms, e.g., pure pursuit guidance [11] and line-of-sight guidance [13], originate from the missile guidance applications, and have the advantage of simplicity, low computational cost, and ease of deployment to autopilots. In these methods, a guidance command that follows an imaginary target point moving along the reference path can be easily acquired, with little limits on the path forms. So, it is possible for these algorithms to provide a general 3-D path following solution.

The most popular virtual target-based algorithm for solving fixed-wing lateral navigation problems is the $\mathcal{L}_1$ guidance law [15], which is a benchmark algorithm for open-source autopilots (e.g., the Pixhawk Autopilot [16]). The $\mathcal{L}_1$ guidance law takes a look-ahead point as the reference for predictive behavior and defines a specified look-ahead length between the vehicle and the look-ahead point to provide smooth incidence commands to the reference path. This guidance law has been extended to maintain a consistent time response in the error vector by calculating the adaptive look-ahead length with a constant period and damping, regardless of ground velocity [17]. Thomas [18] improved its scope of application under the conditions of a strong wind field and small loiter radius. Furthermore, for 3-D path-following extension, the work in [19] and [20] took advantage of the $\mathcal{L}_1$ guidance law for 3-D Dubins paths during an emergency landing. Cho [14] developed a "look-ahead effect" guidance law for parametrical 3-D paths, in which a virtual target direction using the "look-ahead angle" and "radially shifted distance" is applied to generate guidance commands instead of the line-of-sight vector. However, Cho's geometric method involves complicated coordinate transformations, which imposes restrictions on the parameterized 3-D path forms. There is still room for improvement in the practicality of the "look-ahead effect" algorithms. Moreover, from a practical perspective, due to the inherent longitudinal-lateral decoupling dynamics of fixed-wing vehicle, the deviation in navigation should be projected into a dual-plane, which in turn prompts the guidance law to supply longitudinal-lateral commands. In this way, the design of a guidance algorithm decouples the longitudinal plane from the lateral plane so as to be compatible with commercial autopilots, and these features have been readily accepted by practitioners [21], [22]. This prompted us to extend the use of planar guidance methods to solve the 3-D path-following problem.

The main contribution of this article is the development of a simple but effective 3-D nonlinear path-following guidance algorithm that extends the law of planar $\mathcal{L}_1$ guidance to a 3-D curved path by searching for the line of sight (look-ahead vector) from the position of a vehicle relative to a virtual target-point. The look-ahead vector is projected to the longitudinal (vertical) and lateral (horizontal) planes to create the longitudinal-lateral acceleration commands for the inner control loop. For straight-line and circular-arc paths, the look-ahead vectors can be calculated analytically [19], [20], but for 3-D parameterized paths, the problem of determining the virtual target-point often needs to be transformed into a nonlinear optimization problem. However, the numerical solution then tends to converge to a local extremum and the global optimum is typically not unique. To solve this problem, a finite, iterative numerical search method is designed to acquire the closest virtual target-point by introducing auxiliary dynamics to the path-related parameter. The idea of search-based navigation does not strictly depend on the type of curve, which renders it suitable for utilizing the time variable as an auxiliary dynamic to tightly track the discrete optimal trajectory. To avoid the initial position of the vehicle extending beyond the specified look-ahead length from the desired path, an adaptive $\mathcal{L}_1$ length is correspondingly constructed. The advantages of being able to avoid complicated coordinate transformations (e.g., the Serret–Frenet frame) and easily adapt to different types of paths (e.g., polynomial paths or the discrete optimal trajectories) make the guidance strategy sufficiently flexible for deployment in a variety of flight missions. The efficacy of the proposed 3-D guidance algorithm is also tested in real flight.

The rest of this article is organized as follows. Section II describes the design of the guidance law, before Section III presents an experimental validation of the assumptions. Finally, Section IV concludes this article. Throughout in this article, bold notation denotes vectors in $\mathbb{R}^3$ and $\|\bullet\|$ denotes the Euclidean norm.

II. Design of Nonlinear Guidance Law

For the purpose of this article, only the acceleration inputs are required to drive a vehicle to follow a desired path, while maintaining a given velocity. Ignoring the inner loop dynamics and wind speed, the kinematic equations of motion from the guidance inputs can be organized as

$$\begin{cases} \dot{x}_r = \|\boldsymbol{V}_g\| \cos\chi \cos\gamma \\ \dot{y}_r = \|\boldsymbol{V}_g\| \sin\chi \cos\gamma \\ \dot{z}_r = \|\boldsymbol{V}_g\| \sin\gamma \\ \dot{\chi} = \dfrac{a_{yc}}{\|\boldsymbol{V}_g\| \cos\gamma} \\ \dot{\gamma} = \dfrac{a_{zc} - g\cos\gamma}{\|\boldsymbol{V}_g\|} \end{cases} \tag{1}$$

where x_r, y_r, z_r represent the position of the vehicle in the local tangent frame, $\boldsymbol{V}_g$ is the inertial velocity vector of the vehicle, g denotes gravitational acceleration, χ and γ denote the track angle and path angle, and a_{yc} and a_{zc} are acceleration inputs to the kinematic motion generated by the guidance law.

The path-following problem relies on the insight that the guidance generates appropriate control actions, a_{yc} and a_{zc}, to merge the track of vehicle into the desired path and align the velocity of the vehicle with the desired path. In this article, the key idea of the path-following algorithm based on the $\mathcal{L}_1$ planar guidance law is to use the acceleration commands to drive the vehicle to follow the virtual target along the desired path. A detailed discussion is given in the following sections.

A. Description of Planar Nonlinear Guidance Law

The approach to guidance described here originates from a pure pursuit-based 2-D path following method [18]. An imaginary target point moves along the reference path, and the line-of-sight vector from the vehicle to this point defines the look-ahead vector $\boldsymbol{L}$, as indicated in Fig. 1. By choosing an appropriate and

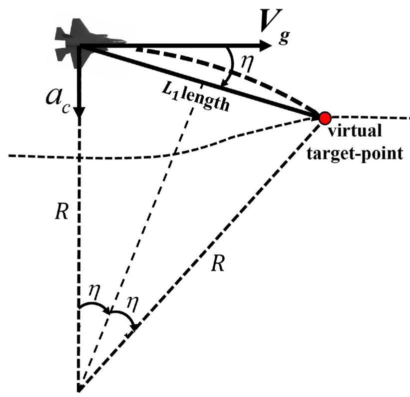


Fig. 1. Geometry of the $\mathcal{L}_1$ look-ahead point-based guidance law.

certain $\mathcal{L}_1$ look-ahead length $\|\boldsymbol{L}\|$, the guidance logic generates acceleration commands to steer the vehicle toward the virtual target.

As shown in Fig. 1, the guidance logic derives from the geometric relationship. $\|\boldsymbol{L}\| = 2R\sin\eta$ is readily available and then the centripetal acceleration required to track the dashed circle (radius R) is equal to the lateral acceleration command [15], $a_c = \frac{\|\boldsymbol{V}_g\|^2}{R} = 2\frac{\|\boldsymbol{V}_g\|^2}{\|\boldsymbol{L}\|}\sin\eta$. In pursuit of a consistent dynamics independent of inertial velocity, the planar $\mathcal{L}_1$ guidance logic is restructured here as [17], [18]

$$a_c = k_L \frac{\|\boldsymbol{V}_g\|}{q_L}\sin\eta \tag{2}$$

where η is the lateral look-ahead angle, and q_L and k_L denote the $\mathcal{L}_1$ ratio and $\mathcal{L}_1$ gain, respectively.

By designing the desired period P_L and damping ξ_L, q_L, and k_L can be constructed as

$$\begin{cases} q_L = \frac{P_L \xi_L}{\pi} \\ k_L = 4\xi_L^2 \end{cases}. \tag{3}$$

Correspondingly, the $\mathcal{L}_1$ look-ahead length is adaptively obtained from the vehicle's ground velocity according to

$$\|\boldsymbol{L}\| = q_L \|\boldsymbol{V}_g\|. \tag{4}$$

Equations (3) and (4) transform the important design of the $\mathcal{L}_1$ length into the choice of period P_L and damping ratio ξ_L. Moreover, P_L and ξ_L can be easy chosen according to the separation criterion between the timescales of guidance and control loops. Some attributes can be inferred from (2)–(4).

1) The direction of acceleration is determined by the look-ahead angle. The aim of the guidance law is to align the vector of the vehicle's inertial velocity with the $\mathcal{L}_1$ look-ahead vector. When the vehicle is far away (close to) the desired path, the guidance logic tends to rotate the vehicle's inertial velocity vector to approach the desired path at a large (small) incident angle.
2) Limited by the nominal $\mathcal{L}_1$ length, the situation in which the vehicle is far from the reference path can increase the cross-track deviation to greater than the $\mathcal{L}_1$ length, causing the virtual target-point to be undefined. Even if the vehicle is less than the specific $\mathcal{L}_1$ length to the path, a large overshoot response occurs owing to the large look-ahead angle.
3) As discussed in [17], the undamped natural frequency of the linearized response of the cross-track deviation becomes $2\pi/P_L$, which is independent of the vehicle's ground velocity. In other words, the time response of the error vector maintains a dynamic consistency while following an arbitrary and feasible ground velocity profile.

The major benefits of the $\mathcal{L}_1$ guidance law include its intuitiveness, convenient parameter tuning, easy-to-implement guidance logic, and flexible adaptation to various 2-D parameterized paths. To extend the $\mathcal{L}_1$ guidance law to the precise tracking of a 3-D parameterized path or a discretized optimal trajectory, the look-ahead vector is projected into two perpendicular planes to provide longitudinal and lateral acceleration commands a_{zc} and a_{yc}, respectively. The parameterized path and issues relating to optimal trajectory tracking are discussed in the following section.

B. Virtual Target Point of Parameterized Paths

In general, it is assumed that the set of parameterized path segments can be expressed as

$$\boldsymbol{p}(t) = \{x_p(t), y_p(t), z_p(t) \,|t \in [0, t^*]\} \tag{5}$$

where the desired path $\boldsymbol{p}(t)$ is parameterized by a single variable and t represents the generalized path-related parameter, i.e., a designed mission time, with t^* as the designed mission duration, or the length of a path segment with t^* denoting the length of the path.

As described by (2), the key procedure of the $\mathcal{L}_1$ guidance law is to obtain the look-ahead angle, which is equivalent to determining the virtual target point $\boldsymbol{P}(t)$ on $\boldsymbol{p}(t)$. Two significant rules of the $\mathcal{L}_1$ guidance law are as follows: 1) the magnitude of the look-ahead vector has an $\mathcal{L}_1$ length specified by (4), and 2) the look-ahead point is ahead of the aircraft. This geometric relationship can be expressed as

$$\begin{cases} \|\boldsymbol{P}(t) - \boldsymbol{r}\| = \|\boldsymbol{L}\| \\ [\boldsymbol{P}(t) - \boldsymbol{r}]^\top \hat{\boldsymbol{T}}_D(t) \geq 0 \end{cases} \tag{6}$$

where $\boldsymbol{r} = (x_r, y_r, z_r)^\top$ denotes the position of the vehicle, $\boldsymbol{D}(t)$ denotes the closest projection point of $\boldsymbol{r}$ on $\boldsymbol{p}(t)$, and $\hat{\boldsymbol{T}}_D(t)$ is the unit tangent vector of $\boldsymbol{D}(t)$.

For straight-line and circular paths, the first term of (6) can be analytically solved. For a curved path, however, it is impossible to obtain a closed-form solution. An effective approach in this case is to transform the first term of (6) into a numerical optimization problem. We call the problem $\boldsymbol{P}0$, and it is given by

$$\boldsymbol{P}(t) := \arg\min_{\boldsymbol{P}(t)} \left(\|\boldsymbol{L}\| - \|\boldsymbol{P}(t) - \boldsymbol{r}\|\right)^2, \quad \text{s.t. } 0 \leq t \leq t^*. \tag{7}$$

An efficient, embedded, real-time nonlinear optimization solver, e.g., the ACADO Toolkit [23] or FORCES, is used to solve this problem. However, the numerical solution tends to converge to the local extremum and the global optimum cannot be uniquely determined. As illustrated in Fig. 2, there exist multiple virtual

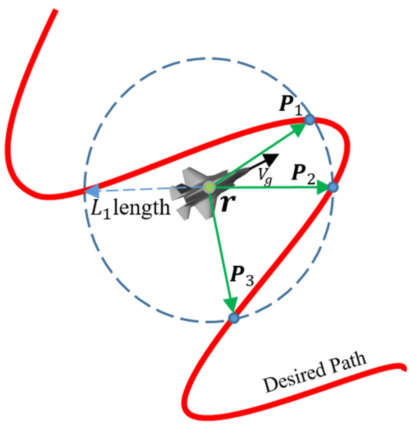


Fig. 2. Multiple virtual target points at $\mathcal{L}_1$ length from the vehicle.

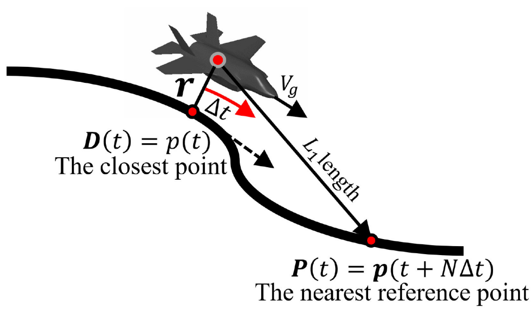


Fig. 3. Proposed numerical search approach to acquire the virtual target point.

target points, $\boldsymbol{P}_1$, $\boldsymbol{P}_2$, and $\boldsymbol{P}_3$, at the same $\mathcal{L}_1$ length from the vehicle, but the correct virtual target point is $\boldsymbol{P}_1$. Thus, it appears that forcibly solving for $\boldsymbol{P}0$ is inadvisable.

To overcome this difficulty, a finite and iterative numerical search is designed to acquire the closest virtual target point by introducing auxiliary dynamics for the path-related parameter. Because the virtual point moves along the desired path from $t = 0$ to $t = t^*$, and its positional behavior is monotonic, as well as $\dot{t} \geq 0$, the direction of the search is always forward along the desired path. Thus, after a finite number of iterations, a feasible virtual target point is obtained in front of the vehicle. This search approach is summarized in the two-step procedure shown in Fig. 3.

We first determine the closest projection point $\boldsymbol{D}(t)$ to $\boldsymbol{r}$ on $\boldsymbol{p}(t)$, defined as

$$\boldsymbol{D}(t) := \arg\min_{t} \|\boldsymbol{r} - \boldsymbol{p}(t)\|, \quad \text{s.t. } t_0 < t < t^* \tag{8}$$

where t represents the desired path-related parameter of $\boldsymbol{D}(t)$ that belongs to $t \in [t_0, t^*]$, and t_0 denotes the initial path-related parameter. In the initial condition t_0 is given as $t_0 = 0$, and in the next sampling step, t_0 is replaced with the last value of t. If the parameterized path is in polynomial form, the MPSolve solver [24], [25], which provides a precise approximation of the roots of polynomials, is used to solve for the extremum of $h(t) = \|\boldsymbol{r} - \boldsymbol{P}(t)\|$. However, this method requires the extrema to be determined for multiple roots by the restriction of the second term of (6), which is impractical. If $h(t)$ includes trigonometric terms or other nonconvex properties, employing an embedded nonlinear optimizer to solve (8) is effective. If we choose auxilary variables to represent trigonometric terms, the optimization turns into a constrained polynomial optimization problem of the unity trigonometric constraint, such as $\sin^2 x + \cos^2 x = 1$, where x is an arbitrary real number. In this case, the new polynomial optimization problem can be cast into finding solutions by zeroing the Lagrangian gradient with respect to optimization variables and an additional Lagrange multiplier. This will become a polynomial-system solving problem. However, since the system is quadratic, all solutions can be found very flexibly via global method like the Groebner-basis method [26]. Once all local minima have been found out, inserting them back to the original problem gives the globally optimal solution by sorting the values of objective function. Such an operation normally takes within a second on most modern computers. Thus, the computational burden for calculating a globally optimal initial condition is guaranteed. Note that the $\mathcal{L}_1$ look-ahead vector does not strictly depend on the $\boldsymbol{D}(t)$, and that $\boldsymbol{D}(t)$ is only used as providing an initial position for searching $\boldsymbol{P}(t)$. This flash feature makes it more reasonable to optimize the solution for $\boldsymbol{D}(t)$ instead of $\boldsymbol{P}(t)$. In general, $\boldsymbol{D}(t)$ can be uniquely determined by online optimizer with an uniqueness of closest point assumption. Sequential quadratic programming is a very effective algorithm for solving nonlinear constrained optimization problems, which divides the original problem into a series of quadratic programming subproblems. Once the Hessian matrix of the objective function can be guaranteed to be positive semidefinite, there must be a global minimum solution. The cost of embedded optimizer is the additional demand for high-performance hardware resources.

By introducing an auxiliary factor for the path-related parameter, Δt, the approximate virtual target point $\boldsymbol{P}(t)$ can be expressed as

$$\|\boldsymbol{L}\| = \|\boldsymbol{r} - \boldsymbol{P}(t)\| \approx \|\boldsymbol{r} - \boldsymbol{p}(t + N\Delta t)\| \tag{9}$$

where N is the finite number of iterations, $N \in [0, N_{\max}]$, and $N_{\max}$ is the maximum allowable number of iterations, also called search horizon. Increasing $N_{\max}$ enlarges the search space but also increases the problem size. The choice of $N_{\max}$ represents a balance between the frequency of guidance and the calculation capability of the on-board processor. Δt is related to the total path length, guidance period, and the vehicle's ground velocity, which can be fixed or adaptive. $N_{\max}\Delta t$ determines the search length, which should satisfy $\|\boldsymbol{p}(t + N\Delta t)\| - \|\boldsymbol{p}(t)\| > \|\boldsymbol{L}\|$.

To guarantee the rationality of the accuracy of the $\mathcal{L}_1$ length, a region of trust is imposed by applying the constraint

$$\big|\|\boldsymbol{r} - \boldsymbol{p}(t + N\Delta t)\| - \|\boldsymbol{L}\|\big| \leq \delta \tag{10}$$

where δ is the region of trust of the $\mathcal{L}_1$ length. Because the $\mathcal{L}_1$ length is proportional to the guidance period, a smaller value will leads to higher guidance bandwidth. To preserve the timescale separation criterion, the value of δ should be proportional to the $\mathcal{L}_1$ length.

In the method used to search for the virtual target point, the position of $\boldsymbol{D}(t)$ in the first step provides the initial position

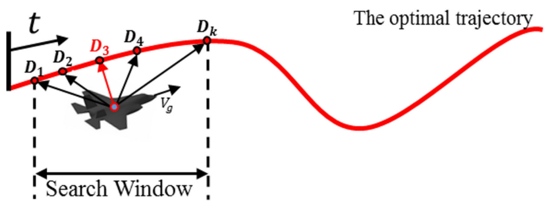


Fig. 4. Geometry of numerical search-based nearest projection point, $\boldsymbol{D}$ in a search window for a discrete optimal trajectory, where $\boldsymbol{D}_k$ (for $k = 1, 2, \ldots$) is the kth discrete point.

of $\boldsymbol{P}(t)$. This means that the precise position of $\boldsymbol{D}(t)$ is not required.

C. Virtual Target Point of Discrete Optimal Trajectory

To achieve the rapidness of optimization algorithms, one methodology is to simplify the problem by using specific form of suboptimal trajectory, e.g., polynomial, Bezier curves, to reduce the dimension of optimal variables. And our proposed method is well applicable to such analytical trajectory. Another mainstream methodology is to customize and improve the state-of-the-art optimization, e.g., convex optimization and pseudospectral transcription, by using direct methods to product discretization points that can preserve physical properties. For instance, in the aerospace applications with regard to rockets, e.g., SpaceX's Falcon-9 and Blue Origin's New Shepard, landing guidance issue is successfully addressed by using online convex trajectory optimization in a discretization points manner. For UCAV applications, limited by the hardware of autopilot, discretization points are usually used to represent the 3-D trajectory. With the rapid development of numerical optimization technology, online trajectory optimization has emerged as an effective way to solve real-time motion planning problems [27]. The form of discretization points also provides a simple and convenient navigation management manner for tracking complicated maneuver trajectories, which is beneficial to practical operation. Thus, the closed-loop tracking of the discrete optimal trajectory has attracted research interest [4]–[6], [28]. As with parameterized paths, e.g., polynomial and Bezier curves, the numerical search-based approach attempts to obtain the virtual target point $\boldsymbol{P}(t)$ for the discrete optimal trajectory. The first step involves determining the closest projection point, $\boldsymbol{D}(t)$. As the optimal trajectory is not analytical, it is inappropriate to utilize an optimizer to solve (8). The following two points are fully considered: 1) For a trajectory generated online, the initial position of the vehicle is close to the beginning of the trajectory, and the initially estimated $\boldsymbol{D}(t)$ is located near the beginning of the trajectory. 2) The $\mathcal{L}_1$ guidance law has a slackness feature related to the position of $\boldsymbol{D}(t)$, and need to only obtain its approximate position. According to these characteristics, a search horizon window is designed to obtain the approximate nearest projection point $\boldsymbol{D}(t)$ by taking the shortest distance between discrete points in the search window and the vehicle as the imaginary approximate point, $\boldsymbol{D}(t)$. As shown in Fig. 4, $\boldsymbol{D}_3$ is the nearest projection point to the discrete point set $\{\boldsymbol{D}_1, \boldsymbol{D}_2, \ldots, \boldsymbol{D}_k\}$, where k is the number of discrete points, expressed as

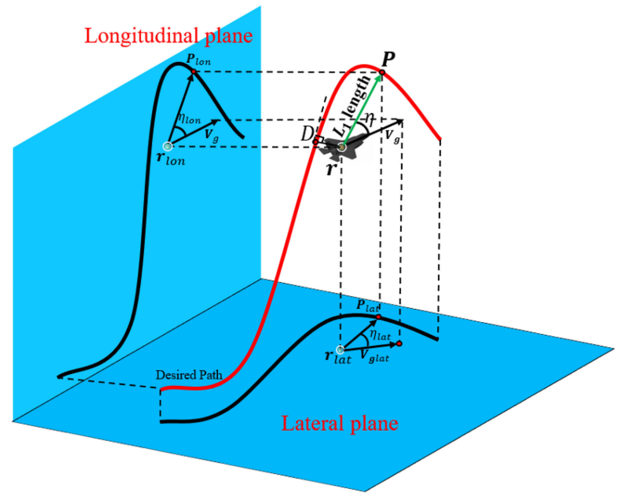


Fig. 5. Dual-plane decoupling $\mathcal{L}_1$ guidance law for a curved path. Note that $\boldsymbol{r}_*$, $\boldsymbol{V}_g$, and η_* (for $* \in \{\mathrm{lat},\ \mathrm{lon}\}$) describe the projections of $\boldsymbol{r}$ and $\boldsymbol{V}_g$ on the longitudinal and lateral planes, respectively.

$$\min_k \{\boldsymbol{D}_1, \boldsymbol{D}_2, \ldots, \boldsymbol{D}_k\}. \tag{11}$$

Once the approximate nearest projection point has been acquired, the next step takes the prespecified time as an auxiliary factor to obtain $\boldsymbol{P}(t)$ using (9) and (10).

D. Dual-Plane Decoupling Nonlinear Guidance Law

Unlike the 3-D, nonlinear, differential geometric path-following guidance law proposed in [14], which generates a normal acceleration command vector perpendicular to the inertial velocity, the method proposed in this article involves splitting the $\mathcal{L}_1$ guidance law into longitudinal and lateral planes. Dual-plane decoupling acceleration commands are generated for an inner-loop controller. Such a decoupling guidance framework preserves the simplicity of the $\mathcal{L}_1$ guidance law and extends the look-ahead point-based method to 3-D applications. This is broadly accepted among aviation engineers, and makes it easy to deploy in a variety of flight missions. As depicted in Fig. 5, the geometric feature is considered to design the guidance logic.

The lateral look-ahead angle η_{lat} and its acceleration command a_{yc} can be organized as

$$\begin{cases} a_{yc} = k_L^{\mathrm{lat}} \frac{\|\boldsymbol{V}_g\|}{q_L^{\mathrm{lat}}} \sin \eta_{\mathrm{lat}} \\ \eta_{\mathrm{lat}} = \tan^{-1} \frac{y_p - y_r}{x_p - x_r} - \tan^{-1} \frac{\dot{y}_r}{\dot{x}_r} \end{cases} \tag{12}$$

where k_L^{lat} is the lateral $\mathcal{L}_1$ gain, with $k_L^{\mathrm{lat}} = 4(\eta_L^{\mathrm{lat}})^2$; q_L^{lat} is the lateral $\mathcal{L}_1$ ratio, with $q_L^{\mathrm{lat}} = \frac{P_L^{\mathrm{lat}} \xi_L^{\mathrm{lat}}}{\pi}$; P_L^{lat} and ξ_L^{lat} are the desired lateral period and damping, respectively, and η_{lat} is limited to $\eta_{\mathrm{lat}} \in [-\frac{\pi}{2}, \frac{\pi}{2}]$.

Similar to the lateral channel, the longitudinal look-ahead angle η_{lon} and its acceleration command a_{zc} can be rewritten as

$$\begin{cases} a_{zc} = k_L^{\mathrm{lon}} \frac{\|\boldsymbol{V}_g\|}{q_L^{\mathrm{lon}}} \sin \eta_{\mathrm{lon}} + g \cos \gamma \\ \eta_{\mathrm{lon}} = \tan^{-1} \frac{z_p - z_r}{\sqrt{(x_p - x_r)^2 + (y_p - y_r)^2}} - \tan^{-1} \frac{\dot{z}_r}{\sqrt{\dot{x}_r^2 + \dot{y}_r^2}} \end{cases} \tag{13}$$

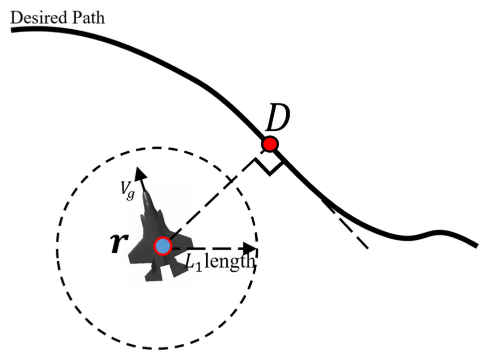


Fig. 6. Geometry of $\mathcal{L}_1$ length becomes undefined for a vehicle far from the desired path.

where k_L^{lon} is the longitudinal $\mathcal{L}_1$ gain, with $k_L^{\text{lon}} = 4(\eta_L^{\text{lon}})^2$; q_L^{lon} is the longitudinal $\mathcal{L}_1$ ratio, with $q_L^{\text{lon}} = \frac{P_L^{\text{lon}} \xi_L^{\text{lon}}}{\pi}$; P_L^{lon} and ξ_L^{lon} are the desired longitudinal period and damping, respectively, and η_{lon} is limited to $\eta_{\text{lon}} \in [-\frac{\pi}{2}, \frac{\pi}{2}]$. The second term added to the first expression in (13) is used to balance the effect of gravity.

Because the curved path can be regarded as a combination of two planar curves projected to the longitudinal and lateral planes, the proof of the asymptotic stability property of decoupled implementation for 3-D curves is equivalent to combination of stability analysis done in each plane. The stability of the look-ahead point-based guidance law for a planar path with constant curvature has been discussed in [14] and [15], and the stability of a planar curved path can be considered in terms of the case when the curvature of the path varies.

E. Extending Nonlinear Guidance Logic to Practice

One inherent weakness of the $\mathcal{L}_1$ guidance law is that the vehicle should be within the $\mathcal{L}_1$ length of the reference path, satisfying $\|\boldsymbol{r}(t) - \boldsymbol{D}(t)\| \leq \|\boldsymbol{L}\|$. This situation is easily violated when the cross-track error is greater than the $\mathcal{L}_1$ length, or when the initial position of the vehicle is far from the reference path, as depicted in Fig. 6. This leads to the case where there is no intersection between the $\mathcal{L}_1$ look-ahead vector and the desired path.

To solve this problem, an additional degree of freedom is introduced to the adaptive $\mathcal{L}_1$ length to satisfy the condition $\|\boldsymbol{r} - \boldsymbol{D}(t)\| \leq \|\boldsymbol{L}\|$. To preserve scalability, the adaptive $\mathcal{L}_1$ length is M times $\|\boldsymbol{r} - \boldsymbol{D}(t)\|$, where $M \geq 1$. Note that increasing M makes the guidance bandwidth weaker while decreasing M makes the overshoot response more aggressive. The choice of M is important in balancing the tradeoff between the guidance bandwidth and the system overshoot.

The corresponding adaptive $\mathcal{L}_1$ ratio and $\mathcal{L}_1$ length may be rewritten as

$$\begin{cases} \|\boldsymbol{L}\| = M \, \|\boldsymbol{r} - \boldsymbol{D}(t)\| \\ q_L^* = \frac{M\|\boldsymbol{r} - \boldsymbol{D}(t)\|}{\boldsymbol{V}_g} \end{cases} \tag{14}$$

where $* \in \{\text{lat}, \ \text{lon}\}$.

The corresponding protection guidance law involves the following.

1) Obtaining the closest projection point $\boldsymbol{D}(t)$ of $\boldsymbol{r}$ on $\boldsymbol{p}(t)$.

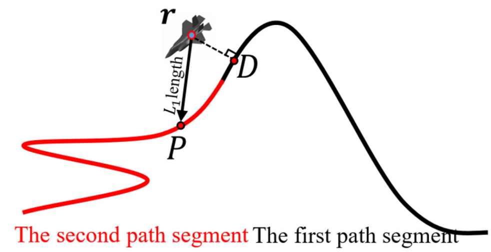


Fig. 7. Geometry of the switching logics of two curved path segments.

2) Calculating the cross-track error $\|\boldsymbol{r} - \boldsymbol{D}(t)\|$, and the $\mathcal{L}_1$ length and $\mathcal{L}_1$ ratio according to (3) to (4).
3) Making a judgment regarding the violation criteria according to the code
If $\|\boldsymbol{L}\| \leq \|\boldsymbol{r} - \boldsymbol{D}(t)\|$ **then**
Update $\mathcal{L}_1$ ratio and $\mathcal{L}_1$ length according to (14).
end.

To avoid division by zero, the guidance law requires the ground velocity to be limited to $\|\boldsymbol{V}_g\| \geq V_{g\,\min}$, where $V_{g\,\min}$ is the minimum ground velocity. Constrained by the vehicle dynamics, the available acceleration, and the corresponding η_{lon} and η_{lat} are also limited. In particular, the lateral acceleration is limited according to $|a_{yc}| \leq a_{yc}^{\max}$; the corresponding constraints $\eta_{\text{lat}}^{\max}$ and $\eta_{\text{lat}}^{\min}$ can then be constructed as

$$\eta_{\text{lat}}^* = \text{sgn}(\eta_{\text{lat}}^*) \sin^{-1} \frac{a_{yc}^{\max} q_L^{\text{lat}}}{k_L^{\text{lat}} \, \|\boldsymbol{V}_g\|} \tag{15}$$

where $* \in \{\min, \ \max\}$; sgn denotes the sign function, $a_{yc}^{\max}$ is the maximum value of the lateral acceleration command, and $\eta_{\text{lat}}^{\max}$ is the maximum value of the lateral look-ahead angle.

The longitudinal acceleration is limited according to $a_{zc}^{\min} \leq a_{zc} \leq a_{zc}^{\max}$; the corresponding constraints $\eta_{\text{lon}}^{\max}$ and $\eta_{\text{lon}}^{\min}$ can then be written as

$$\eta_{\text{lon}}^* = \sin^{-1} \frac{(a_{zc}^* - g\cos\gamma) \, q_L^{\text{lon}}}{k_L^{\text{lon}} \, \|\boldsymbol{V}_g\|} \tag{16}$$

where $* \in \{\min, \ \max\}$; $a_{zc}^{\min}$ and $a_{zc}^{\max}$ are the minimum and the maximum values of the longitudinal acceleration commands, respectively; $\eta_{\text{lon}}^{\max}$ and $\eta_{\text{lon}}^{\min}$ are the maximum and minimum values of the longitudinal look-ahead angles, respectively.

Another extension involves the switching logic of multisegment paths. Because the $\mathcal{L}_1$ virtual target point $\boldsymbol{P}(t)$ is ahead of the closest projection point $\boldsymbol{D}(t)$, it reaches the terminal point of the given path before $\boldsymbol{D}(t)$, as depicted in Fig. 7. The effective switching logic states that once $\boldsymbol{P}(t)$ reaches the terminal point of the given path, the next segment of the path is immediately activated. In the next guidance period, the UCAV tracks the latest path.

III. Experimental Results

In this section, numerical simulations and flight tests are presented to verify the usefulness of the guidance law developed in Section II, and examine its efficacy. A multisegment bow-tie-shaped path and a discrete optimal trajectory are first used to validate the path-following performance and protection

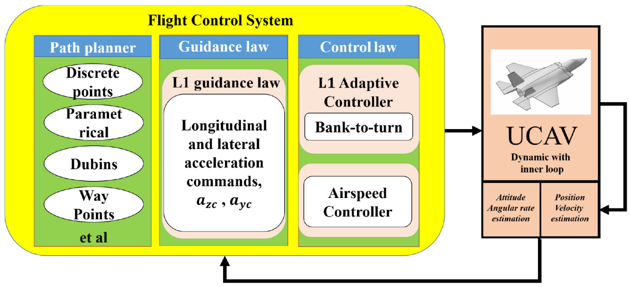


Fig. 8. Complete control architecture.

logic of the guidance strategy. A Monte Carlo simulation under parameter perturbation is then analyzed. Finally, the results of flight tests are used to determine the tracking performance of the proposed method.

The practical guidance and control two-loop structure design is used to facilitate a flight test. The guidance law uses projection to the dual-plane to supply the longitudinal and lateral acceleration commands for the pitch and roll channels. For inner-loop design, the bank-to-turn (BTT) technology is adopted to enhance maneuverability. The BTT scheme enables the UCAV to execute quick maneuvers in two procedures. First, the UCAV quickly rolls the lift plane to the desired direction of maneuvering. Second, it pulls-up the acceleration to allow the flight velocity to point in the desired direction. The inertial guidance commands, a_{yc} and a_{zc} are transformed into the body-axis frame, and the stability controllers are correspondingly constructed to track the normal acceleration a_{bzc} and bank roll angle ϕ_c commands given by (17), maintaining a sideslip angle of zero

$$\begin{cases} \phi_c = \tan^{-1} \frac{a_{zc}}{a_{yc}} \\ a_{bzc} = \sqrt{a_{zc}^2 + a_{yc}^2} \end{cases}. \tag{17}$$

The $\mathcal{L}_1$ adaptive control enhanced technology [29]–[31] is adopted to facilitate the design of the BTT autopilot for the acceleration and roll channels. With a well-designed inner-loop controller, the normal acceleration a_{bz} and the roll angle ϕ can be used to track the guidance commands tightly. The complete control architecture for the autopiloted flight maneuvers is depicted in Fig. 8.

A. Performance of Guidance Algorithm in Normal Simulink

To demonstrate the performance of the guidance algorithm, the multisegment bow-tie-shaped path parameterized in [6] and [32] is followed. Some configurations of the drone in the simulation were also described as in [6]. Related guidance parameters were set to $P_L^{\text{lon}} = P_L^{\text{lat}} = 10$ and $\xi_L^{\text{lon}} = \xi_L^{\text{lat}} = 0.707$, with an associated crossover frequency of 0.628rad/s. For the inner loop, the bandwidth of the acceleration and roll channels was set to 2–3rad/s. In addition, the additional auxiliary factor Δt was set to 0.01. To verify the protection strategy, the drone's initial position was set to be far from the desired path, and M was set to 1.2 to avoid a large overshoot. Other conditions of the drone's initial state were $x_r = 130\,\text{m}$, $y_r = 150\,\text{m}$, $z_r = 200\,\text{m}$, $\chi = -\pi\,\text{rad}$, $\gamma = 0\,\text{rad}$, $\phi = 0\,\text{rad}$, and $\|\boldsymbol{V}_g\| = 20\,\text{m/s}$. And

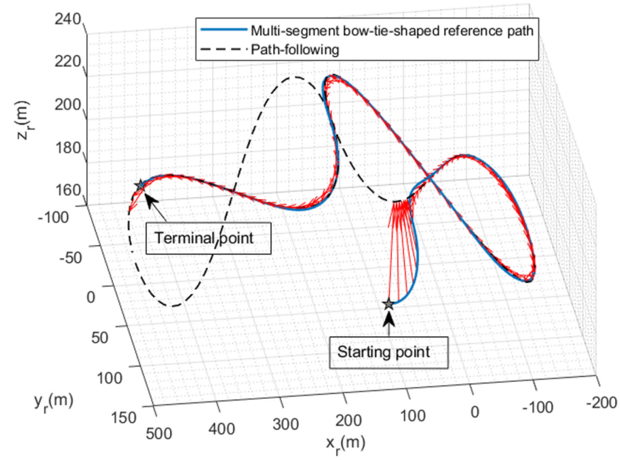


Fig. 9. Multisegment bow-tie-shaped path-following in a simulation when the drone was far from the desired path.

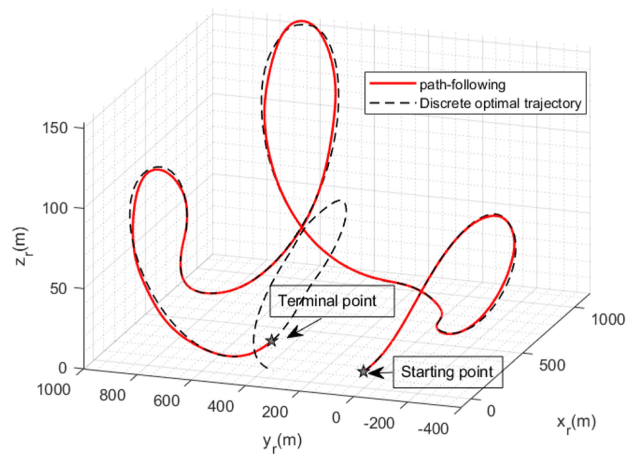


Fig. 10. Discrete optimal trajectory-tracking simulation for the UCAV.

the acceleration and bank angle commands related to vehicle's physical restrictions were limited to $a_{bzc} \in [6, 25]\text{m/s}^2$ and $\phi_c \in [-0.6, 0.6]\text{rad}$. Detailed simulation with complete verification protection strategy was illustrated in Fig. 9. The drone first tracked the right bow-tie-shaped segment; once the virtual target point was near the terminus of this segment, the next segment was activated, which allowed the drone to follow the correct path. The adaptive $\mathcal{L}_1$ vector, shown as the red arrow, was always ahead of the drone while acting as a predictive observer. The path-following profile in Fig. 9, showed that the algorithm achieved satisfactory tracking performance. Furthermore, the implementation of a discrete optimal trajectory tracking simulation was illustrated in Fig. 10. This indicated that the decoupling path-following algorithm exhibited satisfactory tracking accuracy for parameterized or discrete trajectories.

B. Monte Carlo Simulation Under Parameter Perturbation

The robustness of the $\mathcal{L}_1$ guidance law under parametric perturbations is considered in this section. The application scenario involved classic ultralow-altitude ground attacks. The UCAV

TABLE I
RANGES OF UNCERTAIN PARAMETERS IN THE MONTE CARLO SIMULATIONS

T	C_L	C_D	ρ
$\pm 10\%$	$\pm 10\%$	$\pm 10\%$	$\pm 5\%$

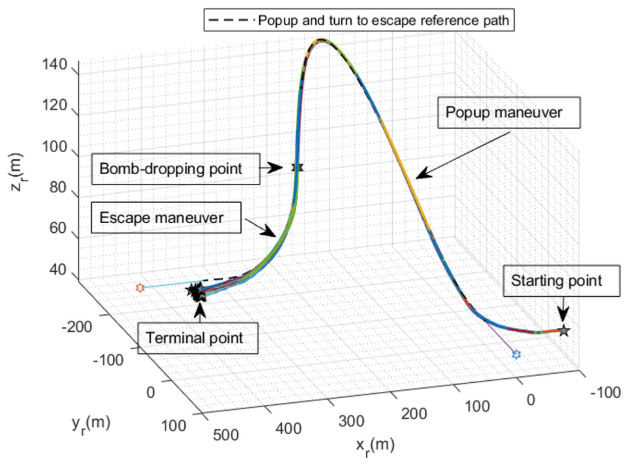


Fig. 11. Results of the Monte Carlo simulations of a ground attack task.

executed a jump-dive bomb-dropping (pop-up) attack maneuver aimed at a ground target ending with a low-altitude escape. Yakimenko's method [33] for determining the near-optimal trajectory was used to design two short-term maneuvers within a ground attack task. For parametric uncertainties, perturbations in the lift factor C_L, drag factor C_D, trust T, and atmospheric density ρ were considered as crucial. The extreme uncertainties in the parametric ranges were illustrated in Table I. For wind disturbance, the Dryden wind turbulence model and horizontal wind model were constructed using a six degree-of-freedom simulation. The guidance-related parameters and auxiliary factor were identical to those detailed in Section III-A, and the corresponding Monte Carlo simulations with randomly perturbed coefficients were executed 500 times; the results were shown in Fig. 11. The results displayed satisfactory path-following performance in the presence of uncertainties.

C. Results of Real Flight Test

This section presents the results of a flight test involving path-following. All experiments were executed using an on-board autopilot with an STM32F765VGT6 216 MHz CPU. The autopilot possessed a sensor redundancy configuration of three-degree for angular motion measurement, two-degree for position measurement, and three-degree for altitude and velocity measurements; these settings were considered secure and reliable. The stability controllers ran in hard real time at 400 Hz and the guidance law ran at 10 Hz. The FCS was built into MATLAB/Simulink and the embedded coder was used to generate the C code for direct deployment. The path-following guidance algorithm with the $\mathcal{L}_1$ adaptive augmentation was thoroughly tested in hardware in-the-loop simulations to guarantee satisfactory tracking performance before the real flight tests. A full duplex serial link at

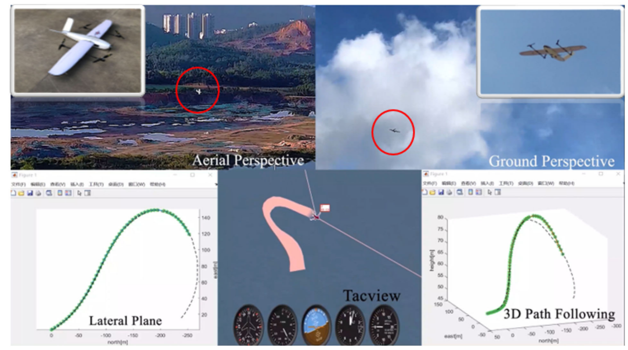
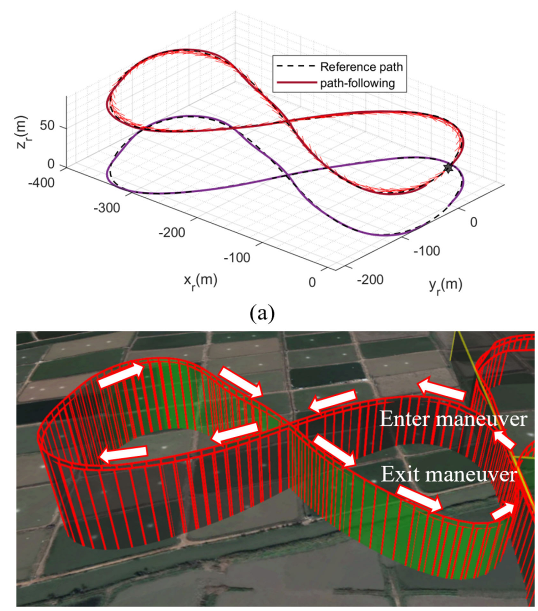


Fig. 12. Spatial maneuver flight test demo for a UCAV.



(a)

(b)

Fig. 13. Results of test flight of bow-tie-shaped path following. (a) The flight experiment of a bow-tie-shaped path. (b) The effect drawing of the flight experiment.

5 Hz was used to exchange telemetry data between the ground station and autopilot. The flight test procedure was as follows. First, the UCAV flew using conventional waypoint navigation mode. Second, the maneuver mode was activated allowing the UCAV to follow a predefined maneuver path. Finally, when the UCAV arrived at the end of the path, it automatically exited maneuver mode and returned to normal flight mode. During the flight, the autopilot continuously recorded logs and transmitted vehicle telemetry information to the ground, and the ground station monitored the flight status in real time. A maneuvering mission scenario is presented in Fig. 12, and a video demonstrating the path-following performance is available at (https://youtu.be/I5Hmm68aLUc).

The prototype UCAV shown in the video had a wingspan of 2.1 m, a body length of 1.2 m, and a weight of 6 kg. Due to vehicle's performance-related restrictions, the acceleration command was limited to $a_{bzc} \in [6, 40]\mathrm{m/s^2}$ and the bank angle was limited to $\phi_c \in [-1.4, 1.4]\mathrm{rad}$. The related control parameters

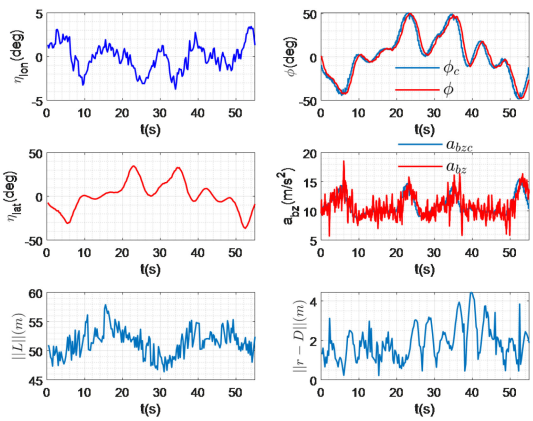
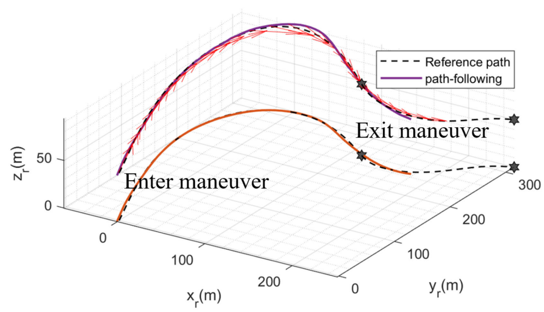
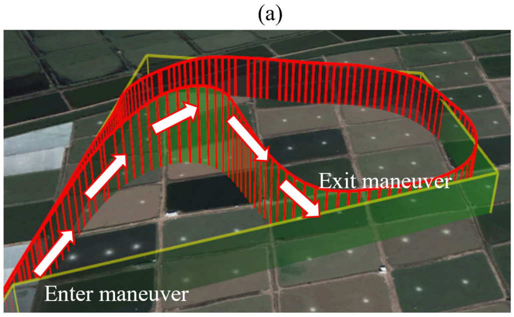


Fig. 14. Time histories for the flight parameters of the bow-tie-shaped maneuver shown in Fig. 13.



(a)



(b)

Fig. 15. Results of test flights of path following, showing segments of the ground attack maneuver. (a) The flight experiment of a spatial maneuver path. (b) The effect drawing of the flight experiment.

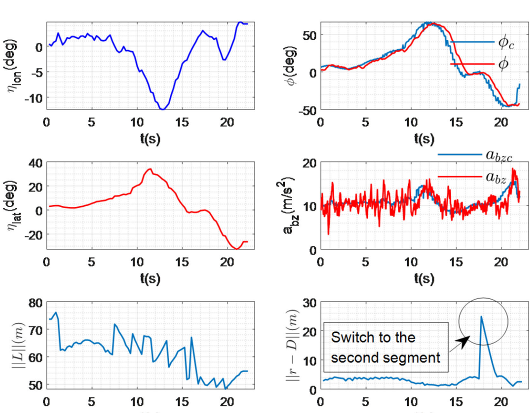
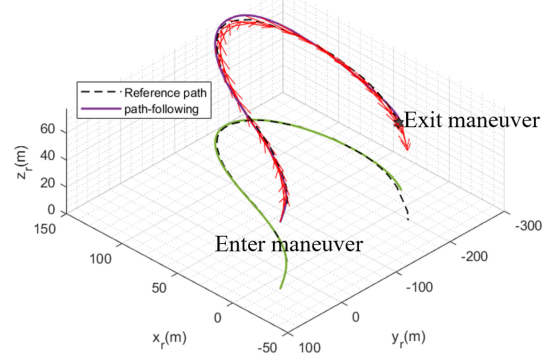
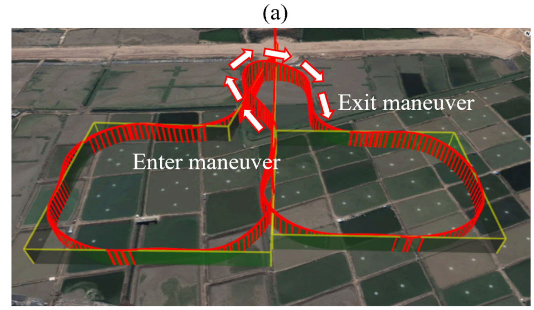


Fig. 16. Time histories for the flight parameters of the ground attack maneuver shown in Fig. 15.



(a)



(b)

Fig. 17. Results of test flight of the discrete optimal trajectory. (a) The flight experiment of a discrete optimal trajectory. (b) The effect drawing of the flight experiment.

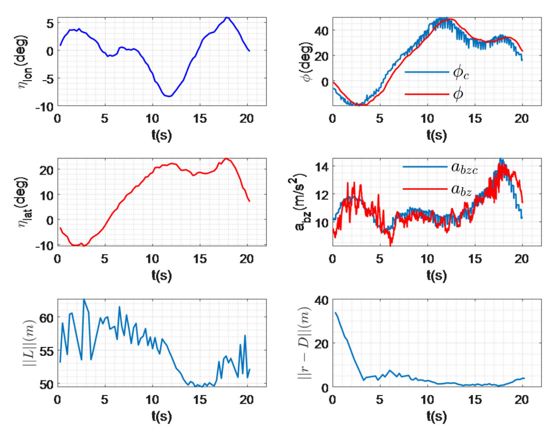


Fig. 18. Time histories for the flight parameters corresponding to the maneuver shown in Fig. 17.

and auxiliary factor were adjusted to those detailed in Section III-A. The desired airspeed of $20\,\mathrm{m/s}$ was constantly maintained. Three cases of flight tests were implemented to tack preplanned maneuver trajectories. As the path planner incorporated the vehicle performance, the UCAV could tightly track the planned climbing, descending, and roll maneuvers. The planned trajectories included parameterized and discrete forms. Figs. 14 and 13 showed flight data for path-following in case of a bow-tie-shaped path, showing the look-ahead angles, η_{lon} and η_{lat}, the corresponding adaptive $\mathcal{L}_1$ length, $\|\boldsymbol{L}\|$, the absolute value of the cross track error $\|\boldsymbol{r}(t) - \boldsymbol{D}(t)\|$, control inputs of acceleration and roll angle a_{bzc} and ϕ_c, and the states of acceleration and the roll angle a_{bz} and ϕ. Affected by atmospheric disturbances and the poor signal-to-noise ratio of the accelerometer, oscillations were observed in the acceleration channel. In addition, oscillations in the $\mathcal{L}_1$ length were caused by the variable ground speed and the numerical searching method. Nevertheless, the decoupling guidance strategy ensured reasonably small tracking errors in the lateral plane, and the cross-track error remained within $\pm 4\ \mathrm{m}$ during the whole experiment. This verified the adaptability of the decoupling guidance approach in dealing with differences in control accuracy and response between channels. As also seen in Fig. 14, benefitting from the BTT design, the maximum bank roll angle reached 60° and the maximum acceleration was greater than 1.5 times the acceleration of gravity during the maneuvering. Figs. 16 and 15 showed the free switching of the guidance strategy for multisegment paths. In the last subgraph of Fig. 16, a sharp increase in the cross-track error occurred as a result of the switching strategy when the next segment path was activated. It was emphasized that the switching logic could be adapted to different types of combined curved paths. A combination of the discrete optimal trajectory and actual track was shown in Figs. 18 and 17. The three real flight cases indicate that, despite the nonanalytical character of the $\mathcal{L}_1$ vector, the guidance strategy enables satisfactory tracking performance. The numerical search method ensured that a credible virtual target point can be found at anytime. The guidance algorithm was constructed using a path-independent assumption, and it was clear that the proposed guidance law eliminates complicated coordinate transformations to make it easy to adjust between various paths types.

IV. Conclusion

This article has described a simple but effective 3-D path-following guidance law. The proposed algorithm extends the planar $\mathcal{L}_1$ guidance law to follow a general 3-D path by projecting the look-ahead vector onto longitudinal and lateral planes, thus preserving the predictable behavior of the pursuit-based method. The look-ahead vector is constructed using a finite and iterative numerical search method by introducing an auxiliary path-related parameter for the searching dynamics. The proposed guidance eliminates complicated coordinate transformations and is built on the path-independent assumption, thus making it easy to adjust between various types of paths, including discrete optimal trajectories. To apply the guidance law, an adaptive $\mathcal{L}_1$ length and corresponding protection strategies were formulated and tested. Numerical simulations and flight tests were also performed to verify the precise tracking performance of the proposed guidance algorithm. Despite the nonanalytical character of the $\mathcal{L}_1$ look-ahead vector, the guidance strategy enables satisfactory tracking performance.

References

[1] P. Sujit, S. Saripalli, and J. B. Sousa, "Unmanned aerial vehicle path following: A survey and analysis of algorithms for fixed-wing unmanned aerial vehicless," *IEEE Control Syst. Mag.*, vol. 34, no. 1, pp. 42–59, Feb. 2014.

[2] N. Yoshitani, "Flight trajectory control based on required acceleration for fixed-wing aircraft," in *Proc. 27th Int. Congr. Aeronautical Sci.*, 2010, vol. 10, pp. 1–10.

[3] L. Qian, S. Graham, and H. H.-T. Liu, "Guidance and control law design for a slung payload in autonomous landing a drone delivery case study," *IEEE/ASME Trans. Mechatronics*, vol. 25, no. 4, pp. 1773–1782, Aug. 2020.

[4] F. Gavilan, R. Vazquez, and E. F. Camacho, "An iterative model predictive control algorithm for UAV guidance," *IEEE Trans. Aerosp. Elect. Syst.*, vol. 51, no. 3, pp. 2406–2419, Jul. 2015.

[5] S. Kim, H. Oh, and A. Tsourdos, "Nonlinear model predictive coordinated standoff tracking of a moving ground vehicle," *AIAA J. Guid. Control Dyn.*, vol. 36, no. 2, pp. 557–566, 2013.

[6] J. Yang, C. Liu, M. Coombes, Y. Yan, and W.-H. Chen, "Optimal path following for small fixed-wing UAVs under wind disturbances," *IEEE Trans. Control Syst. Tech.*, vol. 29, no. 3, pp. 996–1008, May 2021.

[7] D. R. Nelson, D. B. Barber, T. W. McLain, and R. W. Beard, "Vector field path following for small unmanned air vehicles," in *Proc. IEEE Amer. Control Conf.*, 2006, pp. 5788–5794,.

[8] D. Cabecinhas, C. Silvestre, P. Rosa, and R. Cunha, "Path-following control for coordinated turn aircraft maneuvers," in *Proc. AIAA Guid., Navigat. Control Conf. Exhibit.*, 2007, pp. 1–19.

[9] T. Yamasaki, S. Balakrishnan, and H. Takano, "Separate-channel integrated guidance and autopilot for automatic path-following," *AIAA J. Guid. Control Dyn.*, vol. 36, no. 1, pp. 25–34, 2013.

[10] Y. Wang, W. Zhou, J. Luo, H. Yan, H. Pu, and Y. Peng, "Reliable intelligent path following control for a robotic airship against sensor faults," *IEEE/ASME Trans. Mechatronics*, vol. 24, no. 6, pp. 2572–2581, Dec. 2019.

[11] S. Park, "Design of three-dimensional path following guidance logic," *Int. J. Aerosp. Eng.*, vol. 2018, 2018, Art. no. 9235124.

[12] T. Yamasaki, H. Takano, and Y. Baba, "Robust path-following for UAV using pure pursuit guidance," in *Aerial Vehicles*. London, U.K.: IntechOpen, 2009.

[13] R. Rysdyk, "Unmanned aerial vehicle path following for target observation in wind," *AIAA J. Guid. Control Dyn.*, vol. 29, no. 5, pp. 1092–1100, 2006.

[14] N. Cho, Y. Kim, and S. Park, "Three-dimensional nonlinear differential geometric path-following guidance law," *AIAA J. Guid. Control Dyn.*, vol. 38, no. 12, pp. 2366–2385, 2015.

[15] S. Park, J. Deyst, and J. P. How, "Performance and Lyapunov stability of a nonlinear path following guidance method," *AIAA J. Guid. Control Dyn.*, vol. 30, no. 6, pp. 1718–1728, 2007.

[16] L. Meier, P. Tanskanen, L. Heng, G. H. Lee, F. Fraundorfer, and M. Pollefeys, "PIXHAWK: A micro aerial vehicle design for autonomous flight using onboard computer vision," *Auton. Robot.*, vol. 33, no. 1/2, pp. 21–39, 2012.

[17] R. Curry, M. Lizarraga, B. Mairs, and G. H. Elkaim, "L2, an improved line of sight guidance law for UAVs," in *Proc. IEEE Amer. Control Conf.*, 2013, pp. 1–6.

[18] T. Stastny, "L1 guidance logic extension for small UAVs: Handling high winds and small loiter radii," *CoRR*, vol. abs/1804.0, 2018.

[19] P. Eng, L. Mejias, X. Liu, and R. Walker, "Automating human thought processes for a UAV forced landing," *J. Intell. Robot. Syst.*, vol. 57, no. 1–4, pp. 329–349, 2010.

[20] P. Eng, "Path planning, guidance and control for a UAV forced landing," Ph.D. dissertation, School of Engineering Systems, Queensland Univ. Technol., Brisbane, QLD, Australia, 2011.

[21] J. Patrikar, V. R. Makkapati, A. Pattanaik, H. Parwana, and M. Kothari, "Nested saturation based guidance law for unmanned aerial vehicles," *J. Dyn. Syst., Meas., Control*, vol. 141, no. 7, 2019, Art. no. 071008.

[22] W. B. Randal, F. Jeff, and H. Jeffrey, "Fixed wing UAV path following in wind with input constraints," *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 6, pp. 2103–2117, Nov. 2014.

[23] B. Houska, H. J. Ferreau, and M. Diehl, "Acado toolkit-an open-source framework for automatic control and dynamic optimization," *Opt. Control Appl. Method.*, vol. 32, no. 3, pp. 298–312, 2011.

[24] D. A. Bini and G. Fiorentino, "Design, analysis, and implementation of a multiprecision polynomial rootfinder," *Numer. Algorithms*, vol. 23, no. 2/3, pp. 127–173, 2000.

[25] D. A. Bini and L. Robol, "Solving secular and polynomial equations: A multiprecision algorithm," *J. Comput. Appl. Math.*, vol. 272, pp. 276–292, 2014.

[26] J. Wu, M. Liu, Y. Zhu, Z. Zou, and M. Dai, "Globally optimal symbolic hand-eye calibration," *IEEE/ASME Trans. Mechatronics*, vol. 26, no. 3, pp. 1369–1379, Jun. 2021.

[27] D. Morgan, S.-J. Chung, and F. Y. Hadaegh, "Model predictive control of swarms of spacecraft using sequential convex programming," *AIAA J. Guid. Control Dyn.*, vol. 37, no. 6, pp. 1725–1740, 2014.

[28] G. Garcia and S. Keshmiri, "Nonlinear model predictive controller for navigation, guidance and control of a fixed-wing UAV," in *Proc. AIAA Guid., Navigat., Control Conf.*, 2011, pp. 1–14.

[29] C. Cao and N. Hovakimyan, "Design and analysis of a novel L1 adaptive controller, part I: Control signal and asymptotic stability," in in *Proc. IEEE Amer. Control Conf.*, 2006, pp. 3397–3402.

[30] C. Cao and N. Hovakimyan, "Design and analysis of a novel L1 adaptive controller, part II: Guaranteed transient performance," in *Proc. IEEE Amer. Control Conf.*, 2006, pp. 3403–3408.

[31] I. Kaminer, E. Xargay, N. Hovakimyan, and C. Cao, "Path following for small unmanned aerial vehicles using L1 adaptive augmentation of commercial autopilots," *AIAA J. Guid. Control Dyn.*, vol. 33, no. 2, pp. 550–564, 2010.

[32] E. Kayacan and R. Maslim, "Type-2 fuzzy logic trajectory tracking control of quadrotor VTOL aircraft with elliptic membership functions," *IEEE/ASME Trans. Mechatronics*, vol. 22, no. 1, pp. 339–348, Feb. 2017.

[33] Y. A. Oleg, "Direct method for rapid prototyping of near-optimal aircraft trajectories," *AIAA J. Guid. Control Dyn.*, vol. 23, no. 5, pp. 865–875, 2000.

Zian Wang was born in March, 1994, in Taizhou, Zhejiang, China. He is currently working toward the Ph.D. degree in aeronautical engineering from the Nanjing University of Aeronautics and Astronautics, Nanjing, China.

He held three search positions with Shenyang Aircraft Design and Research Institute, in 2016, China Academy of Space Technology, in 2018, and Commercial Aircraft Corporation of China Ltd., in 2020. His research interests include advanced STOVL and eVTOL aircrafts, flight dynamics modeling and analysis, motion planning, guidance law, adaptive control, active disturbance rejection control, model predictive control, cooperative and formation control, and flight-testing.

Zheng Gong received the Ph.D. degree in aeronautical engineering from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2011.

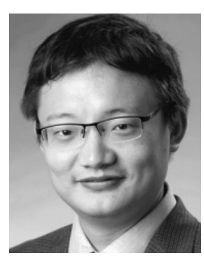
He is currently a Lecturer with the College of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics. He was a Visiting Scholar with the University of Bristol, Bristol, U.K., in 2014. He leads the Flight Dynamics Group, Nanjing University of Aeronautics and Astronautics. His research interests include advanced aircraft systems, advanced missile systems, flight dynamics modeling and analysis, flying qualities evaluation, flight parameter identification, forces combat efficiency, trajectory optimization, flight control system design, simulations, and flight-testing.

Jinfa Xu received the Ph.D. degree in aeronautical engineering from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 1996.

He is currently a Professor with the College of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics. He was a Visiting Scholar with Genex Techonogies Inc., USA, in 2000–2002. He leads the Helicopter Flight Control and Simulation Group, Nanjing University of Aeronautics and Astronautics. His research interests include aircraft navigation guidance and control, advanced flight control systems design, helicopter flight dynamics modeling and control, embedded flight control system implementation, system parameter identification, flight simulations and 3-D visual reality, and flight experimentation and testing.

Jin Wu was born in May, 1994, in Zhenjiang, Jiangsu, China. He received the B.S. degree in automation from the University of Electronic Science and Technology of China, Chengdu, China in 2016. He is currently a PhD student in automation in the RAM-LAB of HKUST, under the supervision of Prof. Ming Liu. His research interests include robot navigation, multi-sensor fusion, automatic control, and mechatronics.

His research interests include robot navigation, multisensor fusion, automatic control, and mechatronics.



Ming Liu (Senior Member, IEEE) received the B.A. degree in automation from Tongji University, Shanghai, China, in 2005, and the Ph.D. degree from the Department of Mechanical and Process Engineering, ETH Zurich, Zurich, Switzerland, in 2013, supervised by Prof. Roland Siegwart.

During his master's study with Tongji University, he stayed one year with the Erlangen-Nunberg University and Fraunhofer Institute IISB, Erlangen, Germany, as a Master Visiting Scholar. He is currently with the Electronic and Computer Engineering, Computer Science and Engineering Department, Robotics Institute, The Hong Kong University of Science and Technology, Hong Kong, as an Associate Professor. His research interests include dynamic environment modeling, deep-learning for robotics, 3-D mapping, machine learning, and visual control.

Dr. Liu is currently an Associate Editor for the IEEE ROBOTICS AND AUTOMATION LETTERS, *International Journal of Robotics and Automation*, *IET Cyber-Systems and Robotics*, IEEE IROS Conference 2018–2021. He was a Guest Editor of special issues in IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING. He is a Program Committee member of Robotics: Science and Systems (RSS) 2021.