

Vector Field Based Sliding Mode Control of Curved Path Following for Miniature Unmanned Aerial Vehicles in Winds*

WANG Yajing · WANG Xiangke · ZHAO Shulong · SHEN Lincheng

DOI: 10.1007/s11424-018-8006-y

Received: 4 January 2018

©The Editorial Office of JSSC & Springer-Verlag GmbH Germany 2018

Abstract In this paper, a curved path following control algorithm for miniature unmanned aerial vehicles (UAVs) in winds with constant speed and altitude is developed. Different to the widely considered line or orbit following, the curved path to be followed is defined in terms of the arc-length parameter, which can be straight lines, orbits, B-splines or any other curves provided that they are smooth. The proposed path following control algorithm, named by VF-SMC, is combining the vector field (VF) strategy with the sliding mode control (SMC) method. It is proven that the designed algorithm guarantees the tracking errors to be a bounded ball in the presence of winds, with the aid of the Lyapunov method and the BIBO stability. The algorithm is validated both in Matlab-based simulations and high-fidelity semi-physical simulations. In Matlab-based simulations, the proposed algorithm is verified for straight lines, orbits and B-splines to show its wide usage in different curves. The high-fidelity semi-physical simulation system is composed of actual autopilot controller, ground station and X-Plane flight simulator in-loop. In semi-physical simulations, the proposed algorithm is verified for B-spline path following under various gain parameters and wind conditions thoroughly. All experiments show the accuracy in curved path following and the excellent robustness to wind disturbances of the proposed algorithm.

Keywords Curved path following, miniature UAVs, sliding mode control, vector field, wind disturbance.

1 Introduction

There has been considerable interest over the last few years in the development of miniature unmanned aerial vehicles (UAVs), because of its low-cost in sensors, electronics, and airframes, and wide applications in military, entertainment, academics among aircraft hobbyists, academic

WANG Yajing · WANG Xiangke · ZHAO Shulong · SHEN Lincheng

College of Mechatronic Engineering and Automation, National University of Defense Technology, Changsha 410073, China. Email: wangyajing12@nudt.edu.cn; xkwang@nudt.edu.cn.

*This research was supported by the National Natural Science Foundation of China under Grant No. 61403406.

◇ This paper was recommended for publication by Guest Editor LIU Tengfei.

researchers, and industries^[1–3]. In different applications, following autonomously a predefined path is a basic requirement for the UAVs. Therefore, path following control, which aims at forcing a UAV to converge to and follow a desired path without any temporal specifications, is necessary, and in fact has been taken as a basic function for a miniature UAV with autopilot. A basic requirement for these path following algorithms is that they must be accurate and robust to wind disturbances^[3].

Several approaches have been proposed for path following problems for different unmanned vehicles. In general, as reported in [3], the approaches can be divided into two categories, namely, the geometric methods and the control techniques. The geometric techniques, including pure pursuit, line-of-sight (LOS), pure pursuit with line-of-sight (PLOS), nonlinear guidance law (NLGL), L1 method, and so on, are usually adopted a geometric method to obtain a virtual target point (VTP), and then force the UAV to track the VTP in realtime^[4–6]. In commercial autopilot systems, the VTP algorithms are the dominated methods for the path following problem, because of its easy application. For example, the Pixhawk autopilot uses the L1 algorithm^[7], Paparazzi autopilot uses the carrot-chasing algorithm^[8], and NLGL algorithm has been tested on a Piccolo autopilot^[9]. In these methods, how to update the VTP on a curve is a crucial issue.

The control techniques usually design control commands based on different (nonlinear) control theoretics to make cross-track errors of the UAV converge to zero^[10–17]. Conversely to the VTP methods, the objective is not to track a moving point, but to get onto the path while flying at a prescribed airspeed. For example, through the construction of vector fields surrounding the path to be followed, the vector-field (VF) method is developed for the UAV path following control, in order to guarantee the tracking errors asymptotically approach zero even in the presence of constant wind disturbances^[11]. In [14], a hybrid controller is designed by combing adaptive switching supervisory control with a nonlinear Lyapunov-based tracking control law to solve the path following problem of global boundedness and convergence of the position tracking error to a neighborhood of the origin. The path following strategies in wind with roll and flight path angle constraints are derived for fixed-wing UAVs following straight lines and orbits by using the theory of nested saturations^[13]. It is known that it is hard to obtain the accurate dynamical models of miniature UAVs, because of its low cost. Therefore, most of the existing control techniques have not been employed in the miniature UAV. Whereas, the VF path following algorithm has been tested on a Kestrel autopilot^[18], because of its easy implementation.

With the repaid developments of the usefulness of UAVs, the capability to plan curved paths and to follow them accurately is of great importance. Many of the algorithms designed to solve the guidance and control problem rely on a decomposition approach, first solving a path planning problem, applying smoothing constraints, forming a trajectory that conforms to the path, and then using a control loop to follow this trajectory^[19]. Whiles, in most of the motion plan missions, the planned paths are curves, even spline curves in most cases, rather than straight-lines and orbits (see the surveys [19–22]). For example, for missions to avoid the emergent targets, the UAV is required to follow a spline curve generated by a Voronoi graph.

In addition, in the problem of finding a trajectory that minimizes some cost functional is of interest, the path is often a curve^[20]. For missions involving coverage path planning among a team of UAVs, precise path tracking is often crucial to achieving the cooperation objective^[22].

It is worth pointing out that the existing usable path following methods are hard to follow the curved path. For example, the work done in [4, 5, 10–14], is only for fixed-wing UAV following straight lines and orbits. For example, for the VF method and its extension, path following control laws are developed only for straight-line paths and circular arcs and orbits. To the best of our knowledge, all autopilot systems, including the Pixhawk, Paparazzi and so on, are not provided the curved path following function till now. Instead, only straight-line and orbit paths (or combinations of lines and orbits) are considered for the guidance control in autopilots, which are both with constant curvatures. However, arbitrary curves with varying curvatures will greatly improve the complexity of the systems, and thus, bring challenges to the controllers design and the theoretical analysis. Though L1 approach presented in [6] provided a tight tracking for curved trajectories, the trajectories are normally perturbed curved lines from a nominal straight lines, and its implementation in the Pixhawk autopilot cannot follow a curved path.

In the real world, wind disturbances are inevitable for flights of UAVs. Effective path tracking algorithms must overcome the effect of the present disturbances both in theory and in applications. Given that wind disturbances vary constantly and are often not easily predicted, path tracking can be very challenging in anything other than calm conditions. Though part of the aforementioned methods considered the wind disturbances, it is hard to apply to the curved path following directly.

The study presented in this paper focuses on the curved path following problem for miniature UAVs flying in winds with constant speed. Building on the concept of vector fields, a sliding mode control type path following control law, named VF-SMC (vector field based sliding mode control), is proposed. Unlike other path following methods, the path to be followed is defined in terms of the arc-length parameter, which can be straight lines, orbits, B-splines or any other curves provided it is smooth and differentiable. A unique contribution of the paper is to deal with curved path following problem, by combination the vector field strategy with the sliding mode control method, that guarantees tracking errors asymptotically approach zero, even in the presence of constant wind disturbances. Further, the only used measurement in the path following control is the course angle of the UAV, therefore it is easy to be implemented in the autopilot. The wind perturbation is treated as an unmatched uncertainty in systems, and the BIBO (bounded-input, bounded-output) stability is used to analyse the stability of the systems. It is proven that the designed algorithm guarantees the tracking errors to be a bounded ball in the presence of winds, with the aid of the Lyapunov method and the BIBO stability. Following, the proposed algorithm was validated in closed-loop system in Matlab-based simulations, and then employed in high-fidelity semi-physical simulations with autopilot hardware and X-Plane flight simulator in-loop under various gain parameters and wind conditions. In Matlab-based simulations, the proposed algorithm is verified for straight lines, orbits and B-splines path following to show its wide usage in different curves. In high-fidelity semi-physical simulations,

the proposed algorithm is verified for B-splines path following thoroughly. To verify the effectiveness of the proposed path following algorithms, different gain parameters are considered in experiments, and the ones with best performance are selected. To verify the robustness to wind disturbances, winds with different speeds and directions are further considered in experiments to test the proposed algorithm under different low-wind and high-wind conditions. All experiments show the accuracy in curved path following and excellent robustness to wind disturbances.

The rest of the paper is organized as follows. Section 2 provides the statement of path following problem with respect to the Serret-Ferret frame. Then the sliding mode control law design based on the vector field is developed in Section 3, with the vector field design and stability analysis. The proposed control strategy was validated in Matlab-based simulations, and in high-fidelity semi-physical simulations with autopilot hardware and X-Plane flight simulator in-loop under various gain parameters and wind conditions in Section 4. Finally, the last section draws the conclusions and proposes some future works.

2 Problem Statements

In many typical missions of UAV, the air speed and the altitude are kept constant^[1, 3, 20]. That is to say, it is assumed that the UAV autopilot has an air-speed controller that holds the air speed constant or approximate constant, and has an altitude hold controller for constant UAV altitude. Therefore, this paper considers the curved path following problem of a fixed-wing UAV with the assumption that the altitude and airspeed are held constant (or nearly so).

Suppose that the air speed V and the course angle (ψ ; inertial speed heading) of the UAV are directly measured by using an on-board sensor. A simplified kinematic model in the two dimensional plane in winds is given as follows

$$\begin{cases} \dot{x} = V \cos \psi + W_x, \\ \dot{y} = V \sin \psi + W_y, \\ \dot{\psi} = \omega, \end{cases} \quad (1)$$

where (x, y) denotes the inertial position, ω is the course rate of the UAV, and (W_x, W_y) represent the components of the wind velocity $V_w = \sqrt{W_x^2 + W_y^2}$ bounded by $V_{w_{\max}}$.

Denote the inertial position of the UAV by $\mathbf{p} = [x, y]^T \in \mathbb{R}^2$. According to (1), it is obtained that

$$\dot{\mathbf{p}} = [\dot{x}, \dot{y}]^T = R(\psi) \begin{bmatrix} V \\ 0 \end{bmatrix} + \begin{bmatrix} W_x \\ W_y \end{bmatrix}, \quad (2)$$

where

$$R(\psi) = \begin{bmatrix} \cos \psi & -\sin \psi \\ \sin \psi & \cos \psi \end{bmatrix}. \quad (3)$$

Formally, the path following problem can be defined by using Serret-Ferret frame^[23]. The geometric path is defined in terms of the arc-length parameters, provided that it is smooth. For any given arc-length parameter l , the inertial position of the point on the path associated with l is denoted by $\mathbf{q}(l) = [x_q(l), y_q(l)] \in \mathcal{R}^2$, at which the Serret-Frenet frame is attached, and moves along the path with speed \dot{l} . The x -axis of the Serret-Frenet frame is aligned with the tangent vector to the path at $\mathbf{q}(l)$ and has an angle $\psi_f(l)$ with respect to the inertial frame. It is clear that

$$\dot{\mathbf{q}}(l) = [\dot{x}_q(l), \dot{y}_q(l)]^T = R(\psi_f) \begin{bmatrix} \dot{l} \\ 0 \end{bmatrix}, \quad (4)$$

where

$$R(\psi_f) = \begin{bmatrix} \cos \psi_f & -\sin \psi_f \\ \sin \psi_f & \cos \psi_f \end{bmatrix} \quad (5)$$

is the rotation matrix from the Serret-Frenet frame to the inertial frame.

Let the error vector \mathbf{e} in the Serret-Frenet frame be decomposed in the along-track error e_s and the cross-track error e_d . Then the inertial error vector is obtained by

$$\mathbf{e} = [e_s, e_d]^T = R^T(\psi_f)(\mathbf{p} - \mathbf{q}(l)). \quad (6)$$

By differentiating equation (6) with respect to time, it follows that

$$\dot{\mathbf{e}} = \dot{R}^T(\psi_f)(\mathbf{p} - \mathbf{q}(l)) + R^T(\psi_f)(\dot{\mathbf{p}} - \dot{\mathbf{q}}(l)). \quad (7)$$

Note that for the rotational matrix $R \in SO(2)$, it is obtained that^[24]

$$\dot{R}(\psi_f) = R(\psi_f)S(\dot{\psi}_f), \quad (8)$$

with

$$S(\dot{\psi}_f) = \begin{bmatrix} 0 & -\dot{\psi}_f \\ \dot{\psi}_f & 0 \end{bmatrix}. \quad (9)$$

Note that $S(\dot{\psi}_f) = -S^T(\dot{\psi}_f)$ is a skew-symmetric matrix.

Considering (2), (4) and (8), Formula (7) is rewritten by

$$\dot{\mathbf{e}} = S^T(\dot{\psi}_f)R^T(\psi_f)(\mathbf{p} - \mathbf{q}(l)) + R^T(\psi_f) \left(R(\psi) \begin{bmatrix} V \\ 0 \end{bmatrix} + \begin{bmatrix} W_x \\ W_y \end{bmatrix} - R(\psi_f) \begin{bmatrix} \dot{l} \\ 0 \end{bmatrix} \right).$$

Together with $R^T(\psi_f)R(\psi) = R(\psi - \psi_f)$, it follows

$$\dot{\mathbf{e}} = -S(\dot{\psi}_f)\mathbf{e} + R(\psi - \psi_f) \begin{bmatrix} V \\ 0 \end{bmatrix} + R^T(\psi_f) \begin{bmatrix} W_x \\ W_y \end{bmatrix} - \begin{bmatrix} \dot{l} \\ 0 \end{bmatrix}. \quad (10)$$

Further, denote the error course angle $\tilde{\psi}$ by $\tilde{\psi} \triangleq \psi - \psi_f$. Hence, we obtain the time derivative of $\tilde{\psi}$

$$\dot{\tilde{\psi}} = \dot{\psi} - \dot{\psi}_f = \omega - \kappa(l)\dot{l},$$

because of

$$\dot{\psi}_f = \frac{d\psi_f}{dt} = \frac{d\psi_f}{ds} \frac{ds}{dt} = \kappa(l)\dot{l},$$

where $\kappa(l)$ is the curvature of the path at $\mathbf{q}(l)$ with respect to l .

Therefore, the error kinematic model of a fixed-wind UAV for the path following problem with respect to the Serret-Frame is

$$\dot{e}_s = V \cos \tilde{\psi} - (1 - \kappa(l)e_d)\dot{l} + W_{ex}, \quad (11)$$

$$\dot{e}_d = V \sin \tilde{\psi} - \kappa(l)e_s\dot{l} + W_{ey}, \quad (12)$$

$$\dot{\tilde{\psi}} = \omega - \kappa(l)\dot{l}, \quad (13)$$

where

$$W_{ex} = W_x \cos \psi_f + W_y \sin \psi_f, \quad (14)$$

$$W_{ey} = -W_x \sin \psi_f + W_y \cos \psi_f, \quad (15)$$

and $V_w = \sqrt{W_x^2 + W_y^2} = \sqrt{W_{ex}^2 + W_{ey}^2}$ is bounded.

3 Main Results

3.1 Vector Field Design

According to the kinematics (11)–(13), a differentiable vector field $\psi_d(e_d) \in (-\psi^\infty, \psi^\infty)$ is constructed, where $\psi^\infty \in (0, \pi/2]$, and $-\psi_d(e_d)$ is a class \mathcal{K} function and globally Lipschitz, with respect to the cross-track error e_d , satisfying

$$\underline{k}_{\psi_d}|e_d| \leq |-\psi_d(e_d)| \leq \bar{k}_{\psi_d}|e_d|, \quad \forall e_d, \quad (16)$$

where $\bar{k}_{\psi_d} > \underline{k}_{\psi_d} > 0$ are Lipschitz constants. It is noted that

$$e_d \sin(\psi_d(e_d)) \leq 0, \quad \forall e_d, \quad (17)$$

as e_d and $\psi_d(e_d)$ are with different signs exactly, and the $\sin(\cdot)$ function is strictly increasing in $(-\pi/2, \pi/2)$.

Remark 3.1 A possible vector field, for example, can be depicted by

$$\psi_d(e_d) = -\psi^\infty \frac{e^{2ke_d} - 1}{e^{2ke_d} + 1}, \quad (18)$$

where $k > 0$ and $\psi^\infty \in (0, \pi/2]$. It is clear that

$$e_d \sin(\psi_d(e_d)) \leq 0, \quad \forall e_d.$$

And further, when $|e_d| \leq e_d^m$, we have

$$4k\psi^\infty \frac{e^{2ke_d^m}}{(e^{2ke_d^m} + 1)^2} |e_d| \leq |-\psi_d(e_d)| \leq k\psi^\infty |e_d|.$$

The evolutions of $\psi_d(e_d)$ in (18) with different parameters k are illustrated in Figure 1.

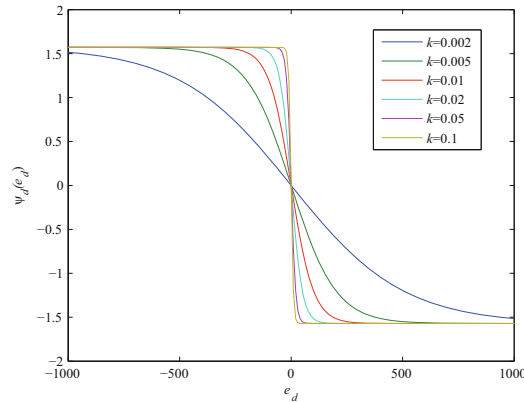


Figure 1 Illustration of $\psi_d(e_d)$ depicted by (18) with respect to e_d

The desired course of the UAV is specified the by direction of the vector field, namely, the error course angle $\tilde{\psi}$ is required to approach the vector field. Therefore, when the cross-track error e_d is large, the UAV is directed to approach the path with error course ψ^∞ and that, as e_d approaches zero, the course angle tends to the direction of the path. Figure 2 provides an illustration of the vector field from (18) with $k = 0.005$ for curved path following.

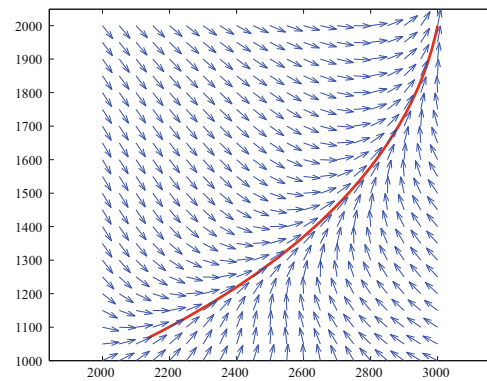


Figure 2 Vector field for curved path following by using (18) with $k = 0.005$

3.2 Control Law Design

It is clear that the wind perturbation is non-vanishing, therefore, it is treated as an unmatched uncertainty in systems. In the following, symbol ψ_d is used to represent $\psi_d(e_d)$ for

short. The variable structure control method is rendered to conclude the controller. In kinematics (11)–(13), the states are $\mathbf{x} = [e_s, e_d, \tilde{\psi}]$.

The slide mode is defined with aid of the vector field, that is,

$$s(\mathbf{x}) = \psi_d - \tilde{\psi}. \quad (19)$$

Considering the equivalent control $s(\mathbf{x}) = 0$, namely, $\psi_d(e_d) = \tilde{\psi}$, we have

Lemma 3.2 When $\psi_d(e_d) = \tilde{\psi}$, taking

$$\dot{l} = k_s e_s + V_c \cos \tilde{\psi}, \quad (20)$$

where $k_s > 0$ is a positive, the error vector $\mathbf{e} = [e_s, e_d]^T$ are globally uniformly ultimately bounded, namely, there exist positive constant b and T independent of t_0 , such that

$$\|e(t_0)\| \leq a \Rightarrow \|e(t)\| \leq b, \quad \forall t \geq t_0 + T$$

for arbitrary large a .

Proof For the equivalent control, a Lyapunov function candidate

$$W_1(e) = \frac{1}{2} \|\mathbf{e}\|_2^2 = \frac{1}{2} (e_s^2 + e_d^2) \quad (21)$$

is employed. Clearly, the function $W_1(e)$ is positive definite and radially unbounded. By Lemma 4.3 in [25], there exist class \mathcal{K} functions α_1 and α_2 , satisfied globally

$$\alpha_1(\|\mathbf{e}\|) \leq W_1(\mathbf{e}) \leq \alpha_2(\|\mathbf{e}\|). \quad (22)$$

The derivative of W_1 along the trajectory of the system is given by

$$\dot{W}_1(e) = e_d \dot{e}_d + e_s \dot{e}_s.$$

By using of (11)–(12), yields

$$\begin{aligned} \dot{W}_1 &= e_d(V \sin \tilde{\psi} - \kappa(s) e_s \dot{s} + W_{ex}) + e_s(V \cos \tilde{\psi} - (1 - \kappa(s) e_d) \dot{s} + W_{ey}) \\ &= e_d V \sin \tilde{\psi} + e_s V \cos \tilde{\psi} - e_s \dot{s} + e_d W_{ex} + e_s W_{ey}. \end{aligned}$$

Considering (20), we have

$$\begin{aligned} \dot{W}_1 &= e_d V \sin \tilde{\psi} - k_s e_s^2 + e_d W_{ex} + e_s W_{ey} \\ &= V e_d \sin \psi_d - k_s e_s^2 + e_d W_{ex} + e_s W_{ey}, \end{aligned}$$

when $\psi_d(e_d) = \tilde{\psi}$.

Further, as $\psi_d \in (-\pi/2, \pi/2)$, it is known that $\sin(\psi_d)$ is monotone increasing with respect to ψ_d , therefore, we have $\frac{2}{\pi} |\psi_d| \leq |\sin \psi_d| \leq |\psi_d|$. Moreover, if and only if $\psi_d = 0$, the equivalence is achieved. Therefore,

$$\frac{2}{\pi} |e_d| |\psi_d| \leq |e_d \sin \psi_d| \leq |e_d| |\psi_d|.$$

As $e_d \sin \psi_d$ and $e_d \psi_d$ are both always less than zero, we have

$$e_d \psi_d \leq e_d \sin \psi_d \leq \frac{2}{\pi} \psi_d e_d \leq 0.$$

Considering ψ_d is globally Lipschitz, (16), yields

$$-\bar{k}_{\psi_d} e_d^2 \leq e_d \sin \psi_d \leq -\frac{2k_{\psi_d}}{\pi} e_d^2 \leq 0. \quad (23)$$

Therefore, we have

$$\dot{W}_1 \leq -V \frac{2k_{\psi_d}}{\pi} e_d^2 - k_s e_s^2 + e_d W_{ex} + e_s W_{ey}. \quad (24)$$

Denote $\mathbf{W}_e = [W_{ex}, W_{ey}]^T$. We rewrote $e_d W_{ex} + e_s W_{ey}$ as $\mathbf{W}_e^T \mathbf{e}$ and used $\mathbf{y}^T \mathbf{x} \leq \|\mathbf{x}\|_2 \|\mathbf{y}\|_2$, then

$$e_d W_{ex} + e_s W_{ey} \leq \|\mathbf{e}\|_2 \|\mathbf{W}_e\|_2. \quad (25)$$

By using (14) and (15) yields:

$$e_d W_{ex} + e_s W_{ey} \leq \|\mathbf{e}\|_2 \sqrt{W_x^2 + W_y^2} = V_w \|\mathbf{e}\|_2. \quad (26)$$

Considering the wind velocity V_w is bounded by $V_{w_{\max}}$, yields:

$$e_d W_{ex} + e_s W_{ey} \leq V_{w_{\max}} \|\mathbf{e}\|_2. \quad (27)$$

Substituting (27) into (24), we obtain

$$\dot{W}_1 \leq -k_{\min} \|\mathbf{e}\|_2^2 + V_{w_{\max}} \|\mathbf{e}\|_2,$$

where $k_{\min} = \min(V \frac{2k_{\psi_d}}{\pi}, k_s)$. Denote $0 < \gamma < k_{\min}$. The foregoing inequality can be rewritten as

$$\dot{W}_1 \leq -(k_{\min} - \gamma) \|\mathbf{e}\|_2^2 - \gamma \|\mathbf{e}\|_2^2 + V_{w_{\max}} \|\mathbf{e}\|_2.$$

Then

$$\dot{W}_1 \leq -(k_{\min} - \gamma) \|\mathbf{e}\|_2^2, \quad \forall \|\mathbf{e}\|_2 \geq \frac{V_{w_{\max}}}{\gamma}.$$

By Theorem 4.18 in [25], we conclude that the solutions are globally ultimately bounded, namely, there exist a positive constant b and T independent of t_0 , such that

$$\|e(t_0)\| \leq a \Rightarrow \|e(t)\| \leq b, \quad \forall t \geq t_0 + T$$

for arbitrary large a .

Further, the ultimate bound is given by

$$b = \alpha_1^{-1} \left(\alpha_2 \left(\frac{V_{w_{\max}}}{\gamma} \right) \right). \quad (28)$$

The proof is finished. ■

Lemma 3.3 For the sliding surface s defined by (19), the reaching law, with constant plus proportional rate reaching form, defined by

$$\dot{s} = -k_{\text{sgn}}\text{sgn}(s) - k_{\omega}s, \quad (29)$$

where k_{sgn} and k_{ω} are positive, will result $\tilde{\psi}$ converging to the sliding surface S in a finite time. Namely, the control input is

$$\omega = -k_{\omega}(\tilde{\psi} - \psi_d) - k_{\text{sgn}}\text{sgn}(\tilde{\psi} - \psi_d) + \kappa(s)\dot{l} + \dot{\psi}_d. \quad (30)$$

Proof Considering a Lypunov function candidate

$$W_2(s) = \frac{1}{2}s^2.$$

If and only if $s = 0$, i.e., $\tilde{\psi} = \psi_d$, the equivalence is achieved.

By differentiating both sides of W_2 with respect to time, and using (29), it is obtained clearly that

$$\dot{W}_2(s) = -k_{\text{sgn}}\text{sgn}(s)s - k_{\omega}s^2 \leq 0.$$

Considering $\text{sgn}(s)s \geq 0$ iff $s = 0$, the equal is achieved, and $k_{\text{sgn}} > 0$, yields:

$$\dot{W}_2(s) \leq -k_{\text{sgn}}\sqrt{2W_2(s)} - 2k_{\omega}W_2(s).$$

Therefore, the trajectory reaches the manifold $s = 0$ in finite time and, once on the manifold, it cannot leave.

In addition, differentiating both sides of Equation (19), we have $\dot{s} = \dot{\psi}_d - \dot{\tilde{\psi}}$. With aid of (13) and (19), it is obtained that

$$\dot{\psi}_d - (\omega - \kappa(s)\dot{l}) = -k_{\omega}(\psi_d - \tilde{\psi}) - k_{\text{sgn}}\text{sgn}(\psi_d - \tilde{\psi}),$$

which results in (30). ■

Remark 3.4 The reaching law can also be designed by different forms, as analysing in [26], such as the constant rate reaching law $\dot{s} = -k_{\text{sgn}}\text{sgn}(s)$, and the power rate reaching law $\dot{s} = -k_{\text{sgn}}|s|^{\alpha}\text{sgn}(s)$, $0 < \alpha < 1$.

Composing of the two above lemmas, we can conduct our first main result.

Theorem 3.5 The system of kinematics (11)–(13), with control laws (20) and (30), is globally uniformly ultimately bounded.

Proof From Lemmas 3.2 and 3.3, we know that when the states are on the switching surface, the sliding surface takes place, making the UAV follow the curved path with the desired approach angle in finite time; otherwise, the reaching law runs, to transfer the states into the sliding mode, which will result in the error vector \mathbf{e} uniformly ultimately bounded in winds. Thus, in this way, the overall system is globally uniformly ultimately bounded intuitionistically^[26]. The theorem can be concluded directly.

Remark 3.6 Recall in Equation (22). Define $\alpha_1(x) = \lambda_1 x^2$ and $\alpha_2(x) = \lambda_2 x^2$, where $\lambda_1 < 1/2$ and $\lambda_2 > 1/2$. Therefore, the infimum of the ultimate bound given in (28) is

$$\inf_{\text{sub}} b = \liminf_{\lambda_2 \rightarrow 1/2^+ \lambda_1 \rightarrow 1/2^-} \sqrt{\frac{\lambda_2}{\lambda_1}} \frac{V_{w_{\max}}}{\gamma} = \inf \frac{V_{w_{\max}}}{\gamma} = \frac{V_{w_{\max}}}{k_{\min}}. \quad (31)$$

Clearly, the bigger k_{\min} , the smaller the infimum of the ultimate bound. Note that $k_{\min} = \min(V \frac{2k_{\psi_d}}{\pi}, k_s)$. Notice that k_s can be selected based on a given radius of the errors and k_{ψ_d} is related to the construction of the function ψ_d (thus, k_{ψ_d} can also be selected for a given radius of the errors). Therefore, to improve the accuracy of the tracking performance, i.e., to reduce the infimum of the ultimate bound, k_s should be bigger enough such as $k_s \geq V \frac{2k_{\psi_d}}{\pi}$, for a constant V and the given vector field function $\psi_d(e_d)$.

Remark 3.7 To avoid the chattering, replace the signum function by a high-slope saturation function, that is the

$$u = -\beta(x) \text{sat}\left(\frac{s}{\mu}\right),$$

where saturation function defined by

$$\text{sat}(y) = \begin{cases} y, & \text{if } |y| \leq 1, \\ \text{sgn}(y), & \text{if } |y| > 1, \end{cases}$$

and μ is a positive constant. And further, many methods are developed to overcome chattering in practice, for example, continuous function $\frac{z_i}{|z_i| + \delta}$, where δ is a positive number which is selected to reduce the chattering problem, can be employed instead of the sign function $\frac{z_i}{|z_i|}$.

4 Simulations and Experiments

The proposed control strategy is validated in closed-loop system in Matlab-based simulations, and then employed in high-fidelity semi-physical simulations with autopilot hardware and X-Plane flight simulator in-loop under various conditions.

4.1 Simulations in Matlab

With the previous description as theoretical foundation, the curved path following algorithm using VF-SMC method with the vector field in (18) is embodied as Algorithm 1.

In Algorithm 1, function $\text{curvefun}(s)$ is monotonic to the arc-length s . It is used to describe the characters of the reference path, and to return the position $P_f(s)$, the angle $\psi_f(s)$ and the curvature $\kappa_f(s)$, with respect to the parameter s . In details, items $\chi_f(s)$ and $\kappa_f(s)$ are obtained as follows,

$$\chi_f(s) = \tan^{-1} \frac{y'}{x'}, \quad \kappa_f(s) = \frac{x'y'' - x''y'}{(x'^2 + y'^2)^{3/2}},$$

where $(\cdot)'$ and $(\cdot)''$ are the derivatives with respect to s . It is worthy pointing out that function $\text{curvefun}(s)$ can be any type of curves, provided it is smooth and differentiable. Therefore, in the sequel, we will show the effectiveness of the VF-SMC algorithm employed in the line

Algorithm 1 Curved path following algorithm using VF-SMC

- 1: Initialize: k_s , k_ω , and k ;
- 2: Obtain the state of UAV: $[P, \psi]$;
- 3: $[P_f, \psi_f, \kappa_f] = \text{curvefun}(s)$;
- 4: $e_\psi = \psi - \psi_f$;
- 5: Compute along-track and cross-track errors by using

$$\begin{bmatrix} e_s \\ e_d \end{bmatrix} = \begin{bmatrix} \cos \psi_f & \sin \psi_f \\ -\sin \psi_f & \cos \psi_f \end{bmatrix} (P - P_f);$$

- 6: $\dot{s} = k_s e_s + V \cos(e_\psi)$;
- 7: Set $\chi^\infty = \pi/2$;
- 8: $\delta(e_d) = -\chi^\infty \frac{e^{2ke_d} - 1}{e^{2ke_d} + 1}$;
- 9: $\delta'(e_d) = -4k\chi^\infty \frac{e^{2ke_d}}{(e^{2ke_d} + 1)^2}$;
- 10: $\Delta_\psi = e_\psi - \delta(e_d)$, and then limit Δ_ψ into $(-\pi, \pi]$;
- 11: $\omega = -k_\omega \Delta_\psi + \kappa_f \dot{s} + \delta'(e_d) (V \sin \Delta_\psi - \kappa_f e_s \dot{s}) + \kappa_f \dot{s}$.

path following, orbit path following and B-spline path following, by giving different $\text{curvefun}(s)$ function, respectively.

All simulations are performed by an m -file with sampled time 0.02s. The inertial speed of the UAV is set to be $V = 20\text{m/s}$, and the used parameters are all set to be $k_s = 1$, $k_\omega = 1$ and $k = 0.01$ in Matlab-based simulations.

4.1.1 Line Path Following

The reference path is given by a line, described by $P(s) = p_0 + s[\cos \theta; \sin \theta]$, where $p_0 = [120; -50]$ and $\theta = \pi/4$. The result of the simulation is shown in Figure 3. It is clear that the UAV can track the reference line well.

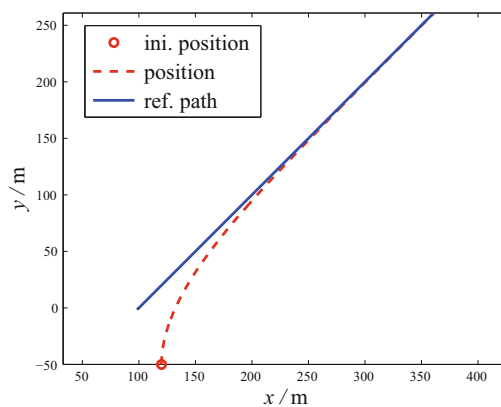


Figure 3 Reference line path and actual trajectory of UAV with VF-SMC in Matlab simulations

4.1.2 Orbit Path Following

The reference path is given by a orbit, described by $P(s) = p_0 + R[\cos s/R; \sin s/R]$, where $p_0 = [120; -50]$ and $R = 200$. The result of the simulation is shown in Figure 4. It is clear that the UAV can track the reference orbit well.

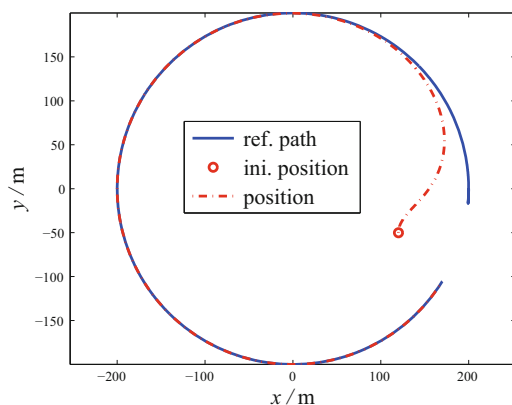


Figure 4 Reference orbit path and actual trajectory of UAV with VF-SMC in Matlab simulations

4.1.3 B-Spline Path Following

The reference path is given by a cubic B-spline, which is generated by 5 positions that the UAV must be passed over. To accord with the case of the X-Plane based simulations, the positions that the UAV must be passed over are represented by the longitude and latitude in same height. Then, taking the first position as the origin, and setting the NED (north-east-down) coordination, all the positions are transformed into a plane coordinate, in which the B-spline path following experiments are done. The 5 positions to be passed over by the UAV are given in Table 1, with their longitudes, latitudes, and (x, y) -coordinates in the plane coordinate. The results of the simulations are shown in Figure 5. It is clear that the UAV can track the reference B-spline well.

Table 1 the points that must be passed over

Num	1	2	3	4	5
Lon	-124.3241	-124.3178	-124.3150	-124.3171	-124.3244
Lat	48.2669	48.2681	48.2646	48.2610	48.2622
x	0	133.4326	-255.7535	-656.0650	-522.6165
y	0	466.3098	673.5792	518.1391	-22.2055

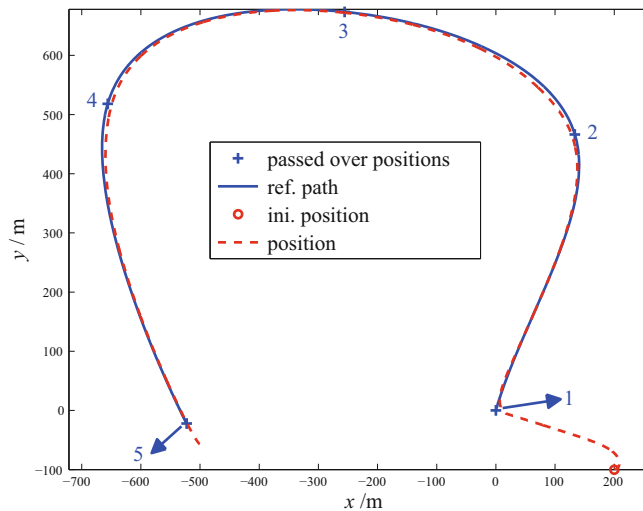


Figure 5 Reference B-spline path and actual trajectory of the UAV with VF-SMC in Matlab simulations

4.2 Experiments in High-Fidelity Semi-Physical Environment

To further verify the proposed tracking law, a high-fidelity semi-physical simulation environment is constructed, which is composed of an actual auto-pilot, a ground station and a X-Plane flight simulator, as shown in Figure 6. The bidirectional communications between the X-Plane and the auto-pilot, the auto-pilot and the ground state are both done by ethernet networks.

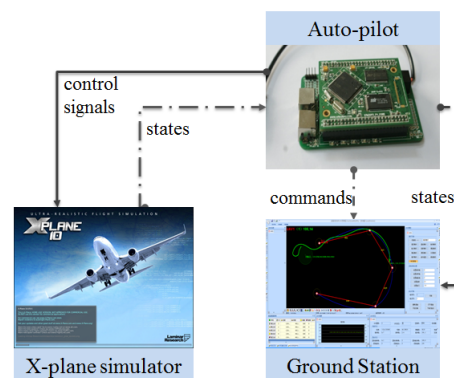


Figure 6 The high-fidelity semi-physical simulation environment, constructed by an actual autopilot, a ground station and an X-Plane flight simulator

The *X*-Plane flight simulator is the world's most comprehensive and powerful flight simulator, and it offers the most realistic flight model available. It is in fact an engineering tool that can be used to predict the flying qualities of fixed- and rotary-wing aircraft with incredible accuracy. Because *X*-Plane predicts the performance and handling of almost any aircraft, it is a great tool for pilots to keep up their currency in a simulator that flies like the real plane, for engineers to predict how a new airplane will fly, and for aviation enthusiasts to explore the world of aircraft flight dynamics^[27]. In the experiments, a widely used airplane, the Great Planes PT-60 RC plane is used. The frame of the used plane with its main parameters are shown in Figure 7.



Figure 7 Great Planes PT-60 RC plane, and its main parameters in *X*-Plane flight simulator

The auto-pilot, with its hardware and embedded software, and the ground station, are both developed by our group, which have been tested in flights^[28]. The main characters of the auto-pilot are with

- 1) Two Cortex-M4 ARM: 168MHz;
- 2) Double 100M network interfaces;
- 3) Supporting record by SD card;
- 4) 8M SRAM, 6 UART, 3 SPI, 2 CAN, and 16 AD conversion with 12 bits.

We employ the proposed VF-SMC tracking law (20) and (30) to track the reference cubic B-spline path. Algorithm 1, together with the cubic B-spline generation algorithm, is realized in the embedded software in the ARM of the auto-pilot by C code. The positions[†] that the UAV must be passed over, with height 100m, are the same as those in the simulations in Matlab. The states of the UAV, such as the longitude, latitude, height, and Euler angles, are transmitted by UDP protocols to the ground station from auto-pilot. The desired inertial speed in speed hold mode is set to be 15m/s.

The related videos about the construction of the environment, and the flying and tracking process in experiments, are attached in the supplementary video materials. The experimental results with parameters $k_s = 1$, $k_\omega = 1$ and $k = 0.01$, the same as that in the proceed simulations, are shown in Figure 8, and the evolutions of the errors between the positions of UAV and the curve path are shown in Figure 9. Note that in Figure 9, the error $d(t)$ is the nearest distance between the positions $P(t)$ of UAV and the curve path $q(s)$.

[†]The Huanghua International Airport at Changsha, the city where the authors' institute located, is chosen in the experiments, and these points (longitudes and latitudes) are selected randomly near the airport.

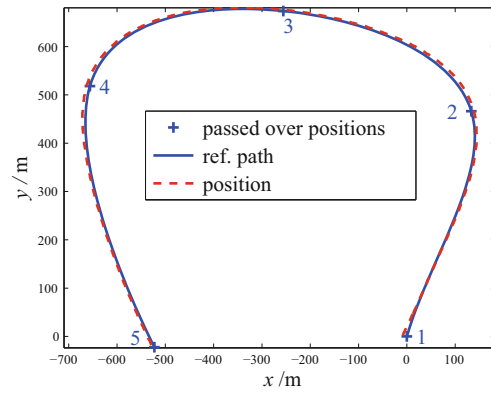


Figure 8 Reference path and actual trajectory of the UAV with VF-SMC in experiments of high-fidelity semi-physical environment

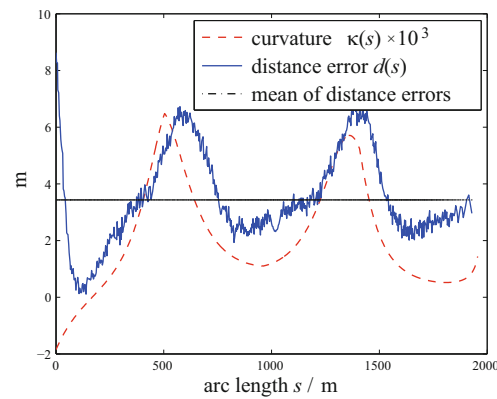


Figure 9 Evolutions of errors between UAV's positions and the curve path, mean of the errors and the curvatures of the curved path, with respect to s , in experiments of high-fidelity semi-physical environment

Although the parameters k_s , k_ω and k in Algorithm 1 are not tuned, it is clear that, from the videos and the figures, the proposed VF-SMC following law can follow the curved path successfully in experiments. The mean path-following error and standard deviation were 3.4347m and 1.6107m, respectively. The arc length of the cubic B-spline is 1986.5m, and the actual distance of the UAV flown is 2013.1m. The actual distance flown was 26.6m more than the desired distance, which is an error of only 1.34%. Investigating Figure 8 and the errors in Figure 9, it is intuitively found that the two local maximums of the errors are near positions 2 and 4 in the path, where are the local maximums of the curvatures in the path. For comparisons, the curvatures of the curve are also shown in Figure 9. It is clearly shown that the errors are heavily related to the curvatures, with a time delay. The bigger the curvature is, the larger the error is. It may be partly because the performances of the algorithm depend on the curvature

of the path, partly because the maneuverability of the used airplane is limited, which confined the UAV to follow the sharp curves.

The performance of the algorithm depends on the selected gain parameters. To further test the effectiveness of the proposed path-following algorithms, different parameters k_s , k_ω and k are considered in experiments. As the work done in [3], two metrics are defined, i.e., total control effort (U) and total error (D). For the proposed VF-SMC algorithm, the control input $u(t) = \omega(t)$, and correspondingly $U = \sum_{t=0}^{t=T} u(t)^2$; while the total error is $D = \sum_{s=0}^{s=ST} d(s)^2$, where ST is the total length of the curve, and $d(t)$ is the nearest distance between positions $P(t)$ of the UAV and the curved path $q(s)$. The total control effort quantifies the control demands of the algorithm; the total error is the offset of the vehicle from the desired path. 48 experiments with different gain parameters are performed, in which the gain parameters k_s , k_ω and k are selected from the sets $k_s \in (0.1, 0.5, 1, 1.5)$, $k_\omega \in (0.1, 0.5, 1, 1.5)$ and $k \in (0.005, 0.01, 0.05)$. The comprehensive comparison made of performance metrics for different parameters are shown in Figure 10.

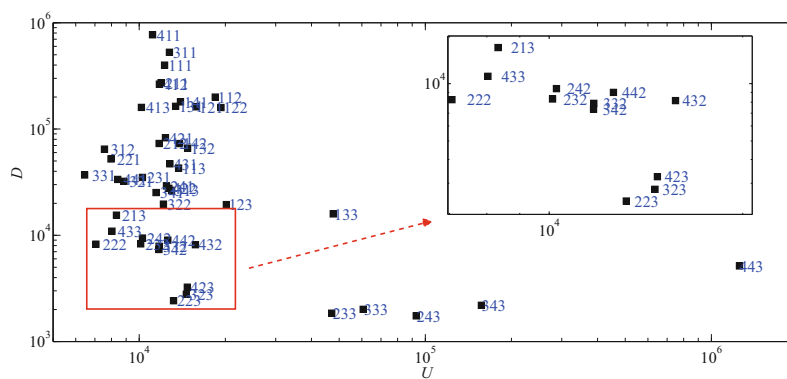
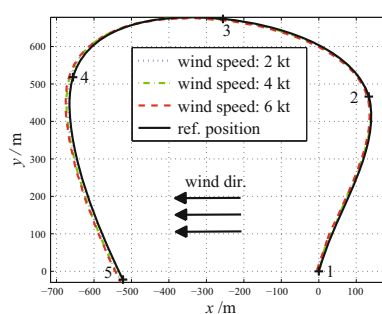


Figure 10 The Pareto front of the VF-SMC path-following algorithm for different gain parameters. The gain parameters k_s , k_ω and k are selected from the sets $k_s \in (0.1, 0.5, 1, 1.5)$, $k_\omega \in (0.1, 0.5, 1, 1.5)$ and $k \in (0.005, 0.01, 0.05)$. The text label for each point refers to the index of the gain parameters $k_s k_\omega k$. For example, 221 refers to $k_s(2) = 0.2$, $k_\omega(2) = 0.2$, $k(1) = 0.005$

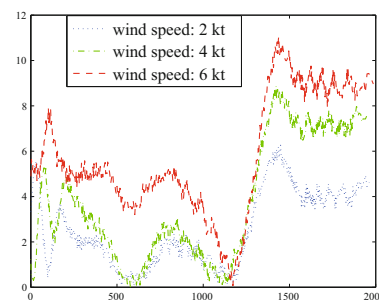
In general, investigating and analysing Figure 10 carefully, for parameter $k_s \in [0.1, 1.5]$, when k_s is located in $[0.5, 1]$, it always achieves better performances; for parameter $k \in [0.005, 0.05]$, the bigger k is, the better the performances are; whereas for parameter $k_\omega \in [0.1, 1.5]$, the total error D is not so sensitive to k_ω when $k \in [0.5, 1.5]$, however the total control effort U will become bigger when k_ω is increased. Considering D and U together, the gain parameters that have the minimum total error and lowest control effort are selected. Using Figure 10 as reference, it shows that the 223 case, i.e., $k_s(2) = 0.2$, $k_\omega(2) = 0.2$ and $k(1) = 0.05$, is to be the best of several potential gain parameters.

In the real world, wind disturbances are inevitable for the flights of UAVs. Effective path tracking algorithms must overcome the effect of the present disturbances. Therefore, in the sequel, winds are further considered in experiments to verify the robustness to wind disturbances of the proposed algorithm. To analyze the performance of the algorithms in the presence of varying wind conditions, winds with different constant speeds and directions are added in experiments regularly. It is known that for Miniature UAVs, wind speeds are commonly 20%–60% of the desired airspeed^[11]. Therefore, the experiments are divided into two groups, viz. the low-wind and the high-wind conditions. In low-wind conditions, the wind speeds are no more than 25% of the commanded airspeed; whileas in high-wind conditions, the wind speeds are around 50% of the command airspeed. All experiments are executed with same initial conditions and using the previously selected best gain parameters $k_s(2) = 0.2$, $k_\omega(2) = 0.2$ and $k(1) = 0.05$, and the commanded airspeed is set to be 15m/s.

In low-wind conditions, the speeds of winds are set to be 2kt[‡], 4kt and 6kt, and the coming directions of winds are set to be 0°, 120° and 240° in the NED coordination, respectively. Thus for these tests the wind speeds are 6.86%, 13.72% and 20.57% of the commanded airspeed, respectively. The flight paths, and the evolutions of errors between positions of UAV and the curved path in experiments are shown in Figures 11–13, and the mean errors, together with standard deviations and max errors under different conditions are listed in Table 2.



(a) Reference path and actual trajectory of UAV



(b) Evolutions of errors between UAV's positions and the curved path

Figure 11 Reference path and actual trajectory of UAV, and evolutions of errors between UAV's positions and the curved path, acted by VF-SMC algorithm under low-wind conditions, whose coming direction is 0° in the NED coordination in experiments of high-fidelity semi-physical environment

For 2kt winds, the mean path errors are about 2m–3m, while the standard deviations of the errors are about 2m. For the 4kt winds, the mean path errors are about 3m–4m, and the standard deviations of the errors are 2m–3m. For the 6kt winds, the means and standard deviations of the error are no more than 6.1m and 3.8m, respectively. In all cases, as expected,

[‡]kt means knots (nautical miles) per hour, which is widely used in air and marine travel. 1 kt \approx 1.852km/h \approx 0.5144m/s.

the largest errors occur when the winds are perpendicular to the course of the UAV; and in general, the larger the angle between the wind's direction and the course of UAV is, the larger the error is, as the sideslip of the UAV, caused by the winds and roll-angle limits, becomes larger.

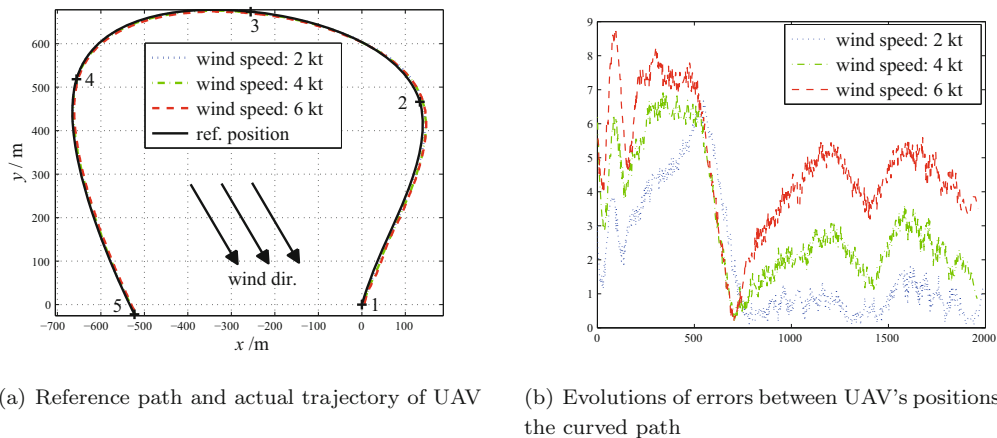


Figure 12 Reference path and actual trajectory of UAV, and evolutions of errors between UAV's positions and the curved path, acted by VF-SMC algorithm under low-wind conditions, whose coming direction is 120° in the NED coordination in experiments of high-fidelity semi-physical environment

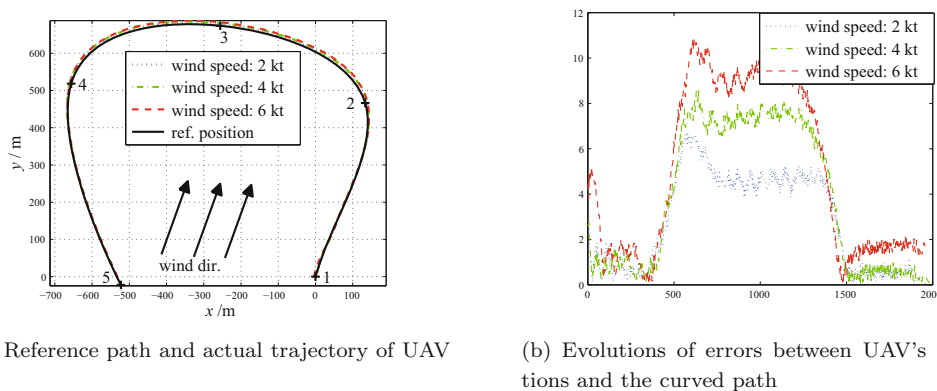


Figure 13 Reference path and actual trajectory of UAV, and evolutions of errors between UAV's positions and the curved path, acted by VF-SMC algorithm under low-wind conditions, whose coming direction is 240° in the NED coordination in experiments of high-fidelity semi-physical environment

Table 2 Mean errors, standard deviations and max errors under low-wind conditions

Speed	Dir.	Mean Err.	Std. Dev.	Max Err.	Ref. to
2 kt	0°	2.6320m	1.6326m	6.2597m	Figure 11
2 kt	120°	1.8358m	1.7096m	6.6648m	Figure 12
2 kt	240°	2.7987m	2.0693m	6.7257m	Figure 13
4 kt	0°	4.0623m	2.7318m	8.8753m	Figure 11
4 kt	120°	3.1406m	1.7936m	6.9075m	Figure 12
4 kt	240°	3.8704m	3.1628m	8.6438m	Figure 13
6 kt	0°	6.0059m	2.6765m	11.0620m	Figure 11
6 kt	120°	4.7554m	1.8211m	8.7542m	Figure 12
6 kt	240°	5.0449m	3.7198m	10.8908m	Figure 13

In high-wind conditions, the coming directions of winds are also set to be 0°, 120° and 240° in the NED coordination as the same as the low-wind conditions; while, the speeds of winds are enlarged to 10kt, 15kt, 20kt and 25kt, respectively, which are up to 34.29%, 51.44%, 68.59% and 85.73% of the commanded airspeed. The flight paths, and the evolutions of errors between positions of UAV and the curved path in experiments are shown in Figures 14–16, and the mean errors, standard deviations and max errors under different conditions are listed in Table 3. For 10kt, 15kt, 20kt and 25kt winds, the mean path errors are no more than 9.3m, 12.8m, 15.9m, and 18.6m, with the related deviations no more than 4.6m, 5.3m, 6.9m and 9.9m, respectively. At large, the UAV can follow the curved path in these high-wind conditions. The max errors between positions of UAV and the curved path, for 10kt, 15kt, 20kt winds, are about 15m, 22m and 30m, respectively; however, when the wind speed is up to 25kt, the max errors are increased to 40m. And in the 25kt wind conditions, the slideslips of the UAV are quite serious, which bring the big overshoot of the UAV's trajectory.

Table 3 Mean errors, standard deviations and max errors under high-wind conditions

Speed	Dir.	Mean Err.	Std. Dev.	Max Err.	Ref. to
10 kt	0°	9.1766m	3.2107m	15.3740m	Figure 14
10 kt	120°	6.7781m	2.6942m	12.0960m	Figure 15
10 kt	240°	7.1788m	4.5415m	15.3133m	Figure 16
15 kt	0°	12.7750m	4.0803m	23.7683m	Figure 14
15 kt	120°	10.1986m	3.6158m	19.1103m	Figure 15
15 kt	240°	9.7120m	5.2303m	21.2844m	Figure 16
20 kt	0°	15.8385m	5.8405m	31.9346m	Figure 14
20 kt	120°	11.5145m	4.5424m	18.1890m	Figure 15
20 kt	240°	11.7902m	6.9160m	32.1015m	Figure 16
25 kt	0°	18.5661m	9.9454m	46.0430m	Figure 14
25 kt	120°	12.8684m	6.6055m	33.1715m	Figure 15
25 kt	240°	13.5450m	8.0865m	38.9701m	Figure 16

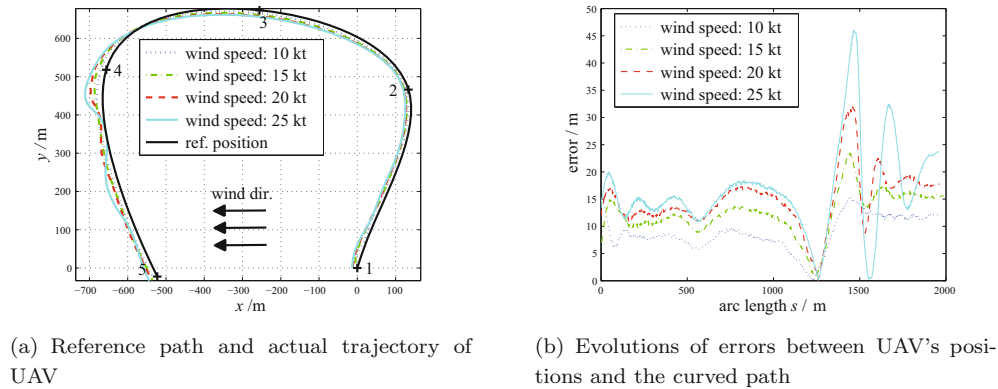


Figure 14 Reference path and actual trajectory of UAV, and evolutions of errors between UAV's positions and the curved path, acted by VF-SMC algorithm under high-wind conditions, whose coming direction is 0° in the NED co-ordination in experiments of high-fidelity semi-physical environment

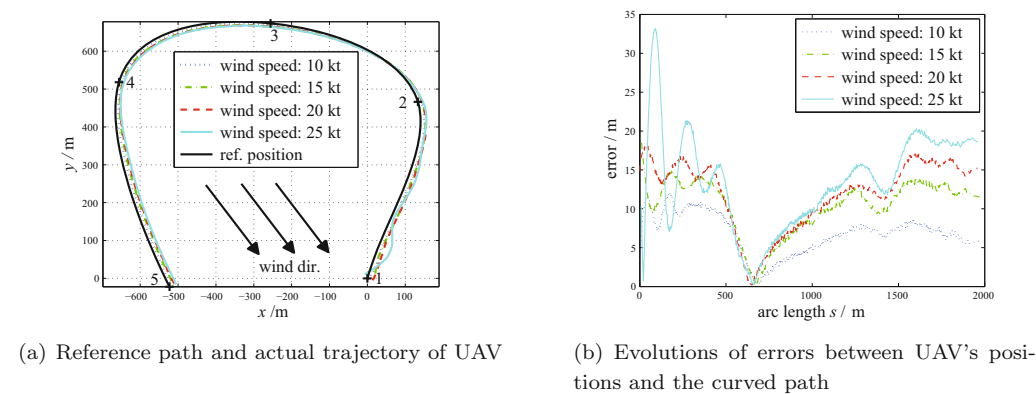
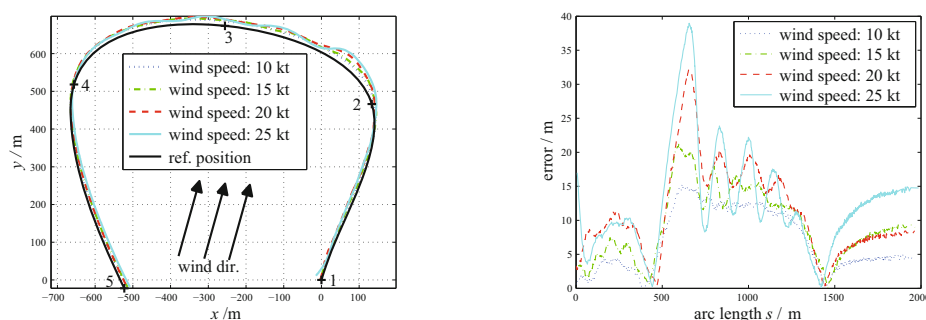


Figure 15 Reference path and actual trajectory of UAV, and evolutions of errors between UAV's positions and the curved path, acted by VF-SMC algorithm under high-wind conditions, whose coming direction is 120° in the NED co-ordination in experiments of high-fidelity semi-physical environment

5 Conclusions and Further Work

In this paper, a new curved path following method, VF-SMC algorithm, for the miniature fixed-wing UAV path following in winds, based on the concept of vector fields and sliding mode control, has been proposed. Using Lyapunov stability criteria, the stabilities of the reaching law and the slidemode are conducted, resulting in the globally uniformly bounded stability of the path following system for arbitrary curves. The effectiveness of the proposed algorithm has been demonstrated both in a Matlab-based closed-loop simulation system, and in a high-fidelity semi-physical closed-loop simulation system with auto-pilot and X-Plane flight simulator in-loop. Unlike the existing method confined to follow straight lines and orbit paths, the proposed algorithm can follow the curved path in winds. In future, input constraints will be considered

for the control law design, together with the case with varying speeds and altitudes of UAVs. And the flight experiments in a fixed-wing MAV are employed on the way. More comparisons with different path following algorithms are also be considered.



(a) Reference path and actual trajectory of UAV

(b) Evolutions of errors between UAV's positions and the curved path

Figure 16 Reference path and actual trajectory of UAV, and evolutions of errors between UAV's positions and the curved path, acted by VF-SMC algorithm under high-wind conditions, whose coming direction is 240° in the NED coordination in experiments of high-fidelity semi-physical environment

References

- [1] Beard R W and McLain T W, *Small Unmanned Aircraft: Theory and Practice*, Princeton University Press, New Jersey, 2012.
- [2] Kumar V and Michael N, Opportunities and challenges with autonomous micro aerial vehicles, *The International Journal of Robotics Research*, 2012, **31**(11): 1279–1291.
- [3] Sujit P B, Saripalli S, and Sousa J B, Unmanned aerial vehicle path following: A survey and analysis of algorithms for fixed-wing unmanned aerial vehicles, *IEEE Control Systems*, 2014, **34**(1): 42–59.
- [4] Conte G, Dynamic 3D path following for an autonomous helicopter, *Proc of IFAC Symposium on Intelligent Autonomous Vehicles*, 2004, 5–7.
- [5] Ambrosino G, Ariola M, Ciniglio U, et al., Path generation and tracking in 3-D for UAVs, *IEEE Transactions on Control Systems Technology*, 2009, **17**(4): 980–988.
- [6] Park S, Deyst J, and How J P, Performance and Lyapunov stability of a nonlinear path-following guidance method, *Journal of Guidance Control and Dynamics*, 2007, **30**(6): 1718–1728.
- [7] Computer Vision and Geometry Lab of ETH Zurich, and Autonomous Systems Lab, <https://pixhawk.org/modules/pixhawk>, 2017.
- [8] ENAC University of Toulouse, and MAVlab of TU-Delft, <http://paparazzi.enac.fr>, 2017.
- [9] Cloud Cap Technology, <http://www.cloudcaptech.com>, 2017.
- [10] Coelho P and Nunes U, Path-following control of mobile robots in presence of uncertainties, *IEEE Transactions on Robotics*, 2005, **21**(2): 252–261.

- [11] Nelson D R, Barber D B, McLain T W, et al., Vector field path following for miniature air vehicles, *IEEE Transactions on Robotics*, 2007, **23**(3): 519–529.
- [12] Ratnoo A, Sujit P B, and Kothari M, Adaptive optimal path following for high wind flights, *IFAC Proceedings Volumes*, 2011, **44**(1): 12985–12990.
- [13] Beard R W, Ferrin J, and Humpherys J, Fixed wing UAV path following in wind with input constraints, *IEEE Transactions on Control Systems Technology*, 2014, **22**(6): 2103–2117.
- [14] Aguiar A P and Hespanha J P, Trajectory-tracking and path-following of underactuated autonomous vehicles with parametric modeling uncertainty, *IEEE Transactions on Automatic Control*, 2007, **52**(8): 1362–1379.
- [15] Dacic D B, Nesic D, and Kokotovic P V, Path-following for nonlinear systems with unstable zero dynamics, *IEEE Transactions on Automatic Control*, 2007, **52**(3): 481–487.
- [16] Ostertag E, An improved path-following method for mixed H_2/H_∞ controller design, *IEEE Transactions on Automatic Control*, 2008, **53**(8): 1967–1971.
- [17] Morro A, Sgorbissa A, and Zaccaria R, Path following for unicycle robots with an arbitrary path curvature, *IEEE Transactions on Robotics*, 2011, **27**(5): 1016–1023.
- [18] Lockheed Martin, <http://www.lockheedmartin.com/us/products/procerus/kestrel.html>, 2017.
- [19] LaValle S M, *Planning Algorithms*, Cambridge University Press, Cambridge, 2006.
- [20] Goerzen C, Kong Z, and Mettler B, A survey of motion planning algorithms from the perspective of autonomous UAV guidance, *Journal of Intelligent and Robotic Systems*, 2010, **57**(1–4): 65.
- [21] Dadkhah N and Mettler B, Survey of motion planning literature in the presence of uncertainty: Considerations for UAV guidance, *Journal of Intelligent & Robotic Systems*, 2012, **65**(1–4): 233–246.
- [22] Galceran E and Carreras M, A survey on coverage path planning for robotics, *Robotics and Autonomous Systems*, 2013, **61**(12): 1258–1276.
- [23] Aguiar A P, Hespanha J P, and Kokotovi P V, Performance limitations in reference tracking and path following for nonlinear systems, *Automatica*, 2008, **44**(3): 598–610.
- [24] Murray R M, Li Z, Sastry S S, et al., *Mathematical Introduction to Robotic Manipulation*, CRC Press, Florida, 1994.
- [25] Khalil H K, *Nonlinear Control*, Prentice Hall, Englewood, 2014.
- [26] Gao W and Hung J C, Variable structure control of nonlinear systems: A new approach, *IEEE Transactions on Industrial Electronics*, 1993, **40**(1): 45–55.
- [27] Laminar Research, <http://www.x-plane.com/manuals/desktop/>, 2017.
- [28] Wang X, Zhu H, Zhang D, et al., Vision-based detection and tracking of a mobile ground target using a fixed-wing UAV, *International Journal of Advanced Robotic Systems*, 2014, **11**(156): 1–11.