Adaptive vector field guidance without a priori knowledge of course dynamics and wind

Ximan Wang, Spandan Roy, Stefano Farì and Simone Baldi *Senior member, IEEE*

***Abstract*—The high maneuverability of fixed wing Unmanned Aerial Vehicles (UAVs) exposes these systems to several dynamical and parametric uncertainties, severely affecting the fidelity of modelling and causing limited guidance autonomy. This work shows enhanced autonomy via adaptation mechanisms embedded in the guidance law: a vector field method is proposed not requiring a priori knowledge of the UAV course time constant, of coupling effects, and of wind amplitude/direction. Stability and performance are assessed using Lyapunov theory. The method is tested on software-in-the loop and hardware-in-the-loop UAV platforms, showing that the proposed guidance law outperforms state-of-the-art guidance controllers and standard vector-field approaches in the presence of significant uncertainty.**

***Index Terms*—Vector field, fixed-wing UAV, adaptive guidance, unknown dynamics, adaptive sliding mode control.**

I. Introduction

Fixed-wing Unmanned Aerial Vehicles (UAVs) are emerging in several fields, due to their aerodynamic efficiency as compared to standard aircrafts built under the constraints imposed by the presence of a human pilot [1]–[3]. To replace the human pilot, autopilot software suites for fixed-wing UAVs (ArduPilot, PX4, DJI, NAVIO2, AscTec Trinity, just to name a few) use measured/reconstructed states [4]–[8] to control aileron, rudder, elevator and thrust, so as to reach the set points provided by the *guidance law* [9]. Proposed guidance laws include geometric approaches [10], [11], acceleration-based control [12], model predictive control [13], and many more [14], [15]. This work focuses on *vector field guidance*, a method originally proposed in [16] and further improved [17]–[19], based on the generation of a field of desired courses as set points to the autopilot. Accordingly, we follow the modelling and control architecture from the book of the proposers of vector field guidance [9], noting that a similar architecture is adopted by most of the aforementioned autopilot software suites. The vector field method has become standard even beyond UAV applications: extensions of the method have appeared for n-dimensional navigation [20], localization [21], obstacle avoidance and formation control [22].

Typical guidance laws are designed under the assumptions that UAV parameters (e.g. roll/pitch/course time constants) are known, course dynamics are linear, longitudinal and lateral motions are not coupled [9]. However, high maneuverability exposes fixed-wing UAVs to unmodelled dynamics and parametric uncertainties, which affect the fidelity of the UAV model and degrade the ideal performance [23], [24]. Studies on wind compensation [25]–[27] and guidance [28], [29] have shown that guidance performance is severely compromised by uncertain dynamics. *Adaptive* guidance ideas have been shown to compensate different levels of uncertainty: a not-exhaustive list comprises estimation methods [30]–[32], model reference adaptive control [33]–[35], switching control [36], $\mathcal{L}_1$ adaptive control [37], deep learning [38], [39], among others. Despite the progress in the field, no vector-field approach has been proposed for the relevant problem of guidance with *no a priori knowledge of UAV course dynamics and wind environment*. Previous studies by the same authors [33], still required knowledge of course time constant, nominal knowledge of wind and a priori bounded unmodelled dynamics. The main contributions of this work are:

- Achieving vector-field path following without structural knowledge of the unmodelled coupling effects and without a priori knowledge of the course time constant and of wind amplitude/direction;
- Connect the adaptive vector field method to the uncertainty framework of adaptive sliding mode cf. [40]–[43] and references therein, while extending it to consider unmodelled dynamics without a priori constant bound.

The first contribution is made possible by including estimation in the guidance laws, to compensate the uncertain terms. The second contribution is possible by considering a state-dependent uncertainty bound. Stability and performance (in the sense of uniformly ultimately boundedness, i.e. convergence of the tracking error to a tunable bound) is proven using Lyapunov theory, and effectiveness is tested on software-in-the-loop and hardware-in-the-loop platforms comprising full UAV dynamics, wind effects and Ardupilot/PX4 autopilots. The proposed method outperforms the state-of-the-art vector field approaches in dealing with significant uncertainty.

The rest of the paper is organized as follows: Sect. II describes the uncertainty setting for UAV dynamics, Sect. III recalls the standard vector field guidance, and Sect. IV presents the proposed adaptive vector field guidance. Simulations are in Sect. V, with concluding remarks in Sect. VI.

This research was partially supported by Natural Science Foundation of China grant 62073074, Double Innovation Plan grant 4207012004, Special Funding for Overseas grant 6207011901, Research Fund for International Scientists grant 62150610499, and IHFC India grant 'Aerial Manipulation' GP/2021/DA/033. *(corresponding author: S. Baldi).*

X. Wang is with Delft Center for Systems and Control, TU Delft, The Netherlands X.Wang-15@tudelft.nl

S. Farì is with German Aerospace Center (DLR), Bremen, Germany, and performed this research at Delft Center for Systems and Control, TU Delft, The Netherlands stefano.fari@dlr.de

S. Roy is with Robotics Research Centre, International Institute of Information Technology Hyderabad, India spandan.roy@iiit.ac.in

S. Baldi is with School of Mathematics, Southeast University, Nanjing, China, and with Delft Center for Systems and Control, TU Delft, The Netherlands s.baldi@tudelft.nl

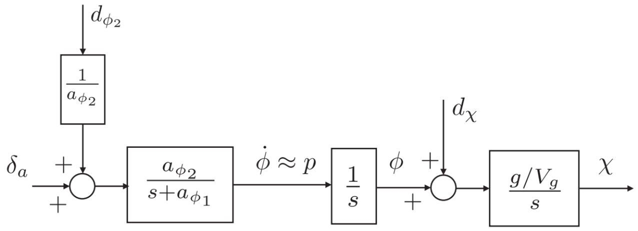

Figure 1. Block diagram for simplified course dynamics (adapted from Figure 5.3 in [9])

II. UAV uncertainty setting

Fixed-wing UAVs can be modelled using 6-DOF Euler-Lagrange equations of motion [9, Chap. 3]. However, to the purpose of guidance, the overall dynamics are usually simplified: after ignoring coupling effects, the dynamics of the roll angle ϕ can be described by [9, Chap. 5]

$$\dot{\phi} = p + d_{\phi_1} \tag{1}$$

where p is the roll rate, and d_{ϕ_1} is an aggregate disturbance

$$d_{\phi_1} = q \sin\phi \tan\theta + r \cos\phi \tan\theta \tag{2}$$

where θ is the pitch angle and r the yaw rate.

After differentiating (1), the block diagram in Fig. 1 can be obtained, showing how the aileron input δ_a and the disturbance d_{ϕ_2} affect the dynamics of the course angle χ

$$\begin{aligned} d_{\phi_2} \triangleq & \dot{d}_{\phi_1} + \Gamma_1 pq - \Gamma_2 qr + \frac{1}{2}\rho {V_a}^2 Sb \times \\ & \left[C_{p_0} + C_{p_\beta}\beta - C_{P_p}\frac{b}{2V_a}(d_{\phi_1}) + C_{_r}\frac{br}{2V_a} + C_{p_{\delta_r}}\delta_r \right] \end{aligned} \tag{3}$$

where β is the side slip angle, q is the pitch rate, V_a the airspeed, ρ the air density, S and b are geometric parameters of the aileron, $\Gamma_{(\cdot)}$ are coefficients related to the inertia matrix of the UAV, and $C_{(\cdot)}$ are coefficients related to the aerodynamics of the UAV. Fig. 1 and (3) clearly show that unmodelled state-dependent terms are aggregated in d_{ϕ_2}, and similar holds for the disturbance d_χ shown in Fig. 1 (the interested reader can refer to the details in [9, Chap. 6]). These disturbances take a very complex form and depends on many parameters. However, despite the presence of state-dependent terms, it is common in the literature (refer to the same book [9, Chaps. 9 & 10] or to [16], [17], [21], [22], [44]) to assume the disturbance to be *bounded a priori* and the course dynamics to be the following *ideal dynamics* for guidance purposes

$$\dot{\chi} = \alpha(\chi^c - \chi). \tag{4}$$

Here, χ is the course of the UAV, representing the angle between the north and the ground velocity V_g; χ^c is the command course from the controller, and α is a positive constant that defines the response speed of the course-hold loop (a cascaded PID not shown in Fig. 1 but present in most autopilot software suites for fixed-wing UAVs). Two comments with respect to (4) are in order:

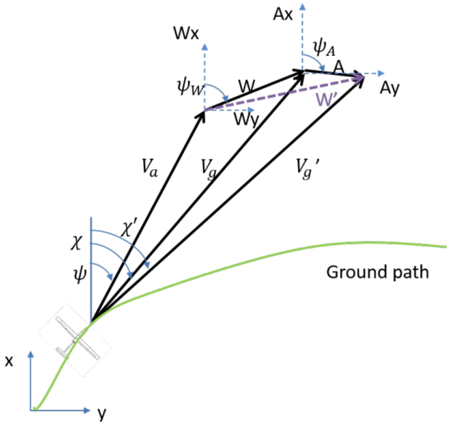

Figure 2. The wind triangle for a fixed-wing UAV. Note that calculating the groundspeed V_g or V_g' requires a priori knowledge of the wind.

1) The dynamics (4) *rely on the assumption that longitudinal and lateral dynamics are decoupled*: in this work, we consider more realistic course dynamics

$$\dot{\chi} = \alpha(\chi^c - \chi) + \Delta(\chi) \tag{5}$$

where $\Delta(\chi)$ is an uncertainty term. The disturbances (2)-(3) reveal that finding a closed-form structure for the term $\Delta(\chi)$ is difficult. We follow an approach motivated by the control-theoretic framework of sliding mode [45, Assumpt. A2, eq. (8)], showing that for a first order system $\dot{x} = f(x) + u + \Delta(x)$, nonlinear unmodelled dynamics $\Delta(x)$ can be represented as

$$||\Delta(x)|| \leq c_0 + c_1 ||x||, \tag{6}$$

where c_0, c_1 are some constants. We will consider unmodelled course dynamics as in (13) in Sect. IV.

2) The steps in [9], [44] show how α in (4) is *affected in a complex way by aerodynamic coefficients which cannot be perfectly known*, and can even change depending on the altitude and velocity. Therefore, the parameter α in (5) should be considered as uncertain or even unknown.

Fig. 2 shows that the wind affecting the airspeed V_a comprises a constant component (with magnitude W and angle ψ_W, giving the nominal groundspeed V_g) and time-varying perturbation (with amplitude $A(t)$ and angle $\psi_A(t)$, giving the actual groundspeed V_g'). Time-varying wind perturbations are typically neglected, resulting in the guidance dynamics

$$\begin{aligned} \dot{x} &= V_a \cos\psi + W \cos\psi_W = V_g \cos\chi \\ \dot{y} &= V_a \sin\psi + W \sin\psi_W = V_g \sin\chi \end{aligned} \tag{7}$$

where ψ is the heading angle between the north and the airspeed velocity V_a, x and y are the coordinate of the earth frame. A third comment follows:

3) The wind introduces another source of uncertainty. The uncertainty in (7) is reflected in the fact that *the ground speed V_g is not known* since a possibly unknown wind component influences it, as shown in Fig. 2.

It is worth mentioning that aspects 2) and 3) are overlooked both in the standard guidance literature, and also in the aforementioned framework of sliding mode, thus requiring a different design departing from existing frameworks.

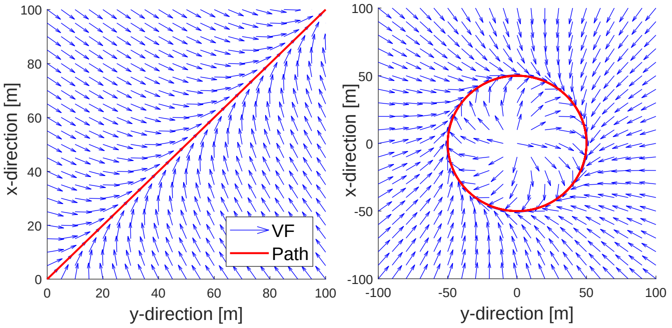

Figure 3. Vector fields for straight line and orbit paths.

III. Standard Vector-Field Guidance

The vector field (VF) is based on specifying a desired course at a certain coordinate, to guide the UAV towards some paths. Two primitive paths are considered: the straight line and the orbit path, with fields of desired courses shown in Fig. 3.

A. Straight-Line Guidance

As in [16], let us consider without loss of generality a straight line parallel to the x-axis. The VF which describes the reference course to drive the UAV on the line is

$$\chi^d(e_y) = -\chi_\infty \frac{2}{\pi} \tan^{-1}(k e_y) \tag{8}$$

where e_y is the tracking error (distance in y-direction), $\chi_\infty \in (0, \frac{\pi}{2}]$ is the course reference when the error is large, and k a governs the VF smoothness. If the straight line is not parallel to the x-axis as in Fig. 3, it suffices to use the rotation matrix from inertial to path frame. In [16] it is shown that the control law which is able to let $\chi \to \chi^d$ and $e_y \to 0$ as $t \to \infty$ is

$$\chi^c = \chi - \chi_\infty \frac{2}{\pi} \frac{\beta_s V_g}{\alpha} \sin(\chi) - \frac{\kappa}{\alpha} \operatorname{sat}\left(\frac{\tilde{\chi}}{\varepsilon}\right) \tag{9}$$

where $\tilde{\chi} = \chi - \chi^d$, $\beta_s = k/(1 + (k e_y)^2)$, κ and ε are parameters governing control aggressiveness and counteracting a possible chattering in the control action, and

$$\operatorname{sat}(x) = \begin{cases} x & \text{if } |x| < 1, \\ \operatorname{sgn}(x) & \text{otherwise.} \end{cases} \tag{10}$$

It is worth mentioning that the continuous sat function is used in [16] to approximate the behavior of a sgn function and avoid discontinuity in the closed-loop solutions.

B. Orbit Guidance

The strategy for orbit guidance builds course VF around the desired orbit (cf. Fig. 3):

$$\chi^d(\tilde{d}) = \gamma + \lambda\left(\frac{\pi}{2} + \tan^{-1}(k\tilde{d})\right) \tag{11}$$

where is $\tilde{d} = d - R$, d is the UAV distance from the orbit center, R is the orbit radius and γ is the angle between the north and the UAV position with respect to the orbit center. For easiness of analysis, the UAV position is expressed in polar coordinates: λ is 1 for clockwise path and -1 for counter-clockwise path. In [16] it is shown that the control law which is able to let $\chi \to \chi^d$ and $\tilde{d} \to 0$ as $t \to \infty$ is

$$\chi^c = \chi + \frac{V_g}{\alpha d} \sin(\chi - \gamma) + \beta_o \frac{\lambda V_g}{\alpha} \cos(\chi - \gamma) - \frac{\kappa}{\alpha} \operatorname{sat}\left(\frac{\tilde{\chi}}{\varepsilon}\right) \tag{12}$$

where $\beta_o = k/(1 + (k\tilde{d})^2)$, and the parameters k, κ, ε are similar to the straight-line case. The proof of the Lyapunov stability for (9) and (12) is given in [16] and will not be further discussed. One crucial observation on (9) and (12) follows.

Remark 1. *The guidance laws* (9) *and* (12) *require knowledge of the course time constant α, and of the groundspeed V_g. Fig. 2 and* (7) *show that the groundspeed requires knowledge of the wind. No guidance law has been proposed in VF literature [20], [21], [46] in the absence of such prior knowledge.*

IV. Adaptive Vector-Field Guidance

To depart from the ideal assumptions in the literature, the following state dependency of uncertainty Δ is considered:

$$|\Delta(\chi)| \leq c_0 + c_1 \left|\tilde{\chi} + \chi^d\right| \leq \kappa_0 + \kappa_1 |\tilde{\chi}| \tag{13}$$

for some scalars κ_0, $\kappa_1 \in \mathbb{R}^+$. We have used (6) and the fact that χ^d is bounded by definition. Under the assumption that κ_0, κ_1 are known, the modelling approach (13) was proposed in sliding mode literature (cf. [45, eq. (8)] and related works) as a way to model complex (state-dependent) disturbances. Notice that (13) includes the fact that $\Delta(\chi)$ *may not be bounded a priori by a constant*. However, we want to deal with κ_0, κ_1 being unknown, which is not considered in standard sliding mode literature. It is worth mentioning that even *adaptive* sliding mode literature (cf. [40]–[43] and related works) models uncertainty as $|\Delta(\chi)| \leq \kappa_0$, with possibly unknown κ_0: state-dependencies entering through the course cannot be fully captured by this approach.

For state-dependent uncertainties, the following is a standard notion of stability [47, Def. 4.6]:

Definition 1. *The solutions of a nonlinear system $\dot{x} = f(x)$ are Uniformly Ultimately Bounded (UUB) with ultimate bound b if there exist positive constants b and c and for every $a \in (0, c)$, there is a time $T(a, b)$ such that*

$$\|x(0)\| \leq a \Rightarrow \|x(t)\| \leq b, \quad \forall t \geq T(a, b). \tag{14}$$

We now propose an adaptive VF with the distinguishing feature of compensating for lack of knowledge of α, of V_g and of state-dependent $\Delta(\chi)$. We introduce appropriate estimators and refer to the approach as *adaptive VF guidance*. It will be proven that the proposed approach achieves UUB solutions, with the ultimate bound being a performance indicator.

A. Straight-Line Adaptive Guidance

Since $\alpha > 0$, (5) can be written as

$$\overline{\alpha}\dot{\chi} = -\chi + \chi^c + \overline{\Delta}, \tag{15}$$

where $\overline{\alpha} \triangleq 1/\alpha, \overline{\Delta} \triangleq \Delta/\alpha$. For control design purposes, the derivative of (8) is calculated in [16] as

$$\dot{\chi}^d = -\chi_\infty \frac{2}{\pi} \beta_s V_g \sin(\chi), \tag{16}$$

Being V_g unknown, $\dot{\chi}^d$ is not available for control deign. Then observing (13), (15) we have

$$\left|\overline{\Delta}\right| \leq \kappa_0^* + \kappa_1^* \left|\tilde{\chi}\right|, \tag{17}$$

where $\kappa_0^* \triangleq \kappa_0/\alpha, \kappa_1^* \triangleq \kappa_1/\alpha$ are unknown positive constants. For ease of controller design, let us also define $\kappa_2^* \triangleq \overline{\alpha} V_g$, which is also an unknown positive constant.

Based on the uncertainty structure (17), a guidance law is proposed as

$$\chi^c = -\Lambda\tilde{\chi} + \chi - \hat{\kappa}_2 \chi_\infty \frac{2}{\pi}\beta_s \sin(\chi) - \rho\,\mathrm{sat}\left(\frac{\tilde{\chi}}{\varepsilon}\right), \tag{18a}$$

$$\rho = \hat{\kappa}_0 + \hat{\kappa}_1|\tilde{\chi}|, \tag{18b}$$

where $\Lambda \in \mathbb{R}^+$ is a user-defined scalar, and $\hat{\kappa}_i$ are the estimates of κ_i^* $i = 0, 1, 2$, evaluated via the following adaptive laws:

$$\dot{\hat{\kappa}}_0 = |\tilde{\chi}| - \zeta_0\hat{\kappa}_0, \tag{19a}$$

$$\dot{\hat{\kappa}}_1 = |\tilde{\chi}|^2 - \zeta_1\hat{\kappa}_1, \tag{19b}$$

$$\dot{\hat{\kappa}}_2 = \chi_\infty \frac{2}{\pi}\beta_s \sin(\chi)\tilde{\chi} - \zeta_2\hat{\kappa}_2, \tag{19c}$$

$$\text{with } \hat{\kappa}_i(0) > 0, \ i = 0, 1, 2, \tag{19d}$$

where $\zeta_i \in \mathbb{R}^+$ are user-defined scalars.

The following result can be derived:

Theorem 1. *By employing the guidance law (18), the resulting trajectories of the UAV (15) and the parameters in the adaptive law (19) are Uniformly Ultimately Bounded (UUB).*

Proof. See the Appendix. Tunability of the ultimate bound is elaborated at the end of the proof according to (39)-(40) and standard Lyapunov arguments [47, Sect. 4.8]. □

B. Orbit Adaptive Guidance

For control design purposes, the derivative of (11) is calculated in [16] as

$$\dot{\chi}^d = V_g\left(\frac{\sin(\chi-\gamma)}{d} + \lambda\beta_o\cos(\chi-\gamma)\right). \tag{20}$$

The corresponding guidance law is defined as

$$\chi^c = -\Lambda\tilde{\chi} + \chi + \hat{\kappa}_2\left(\frac{\sin(\chi-\gamma)}{d} + \lambda\beta_o\cos(\chi-\gamma)\right) - \rho\,\mathrm{sat}\left(\frac{\tilde{\chi}}{\varepsilon}\right), \tag{21a}$$

$$\rho = \hat{\kappa}_0 + \hat{\kappa}_1|\tilde{\chi}|, \tag{21b}$$

with the following adaptive laws:

$$\dot{\hat{\kappa}}_0 = |\tilde{\chi}| - \zeta_0\hat{\kappa}_0, \tag{22a}$$

$$\dot{\hat{\kappa}}_1 = |\tilde{\chi}|^2 - \zeta_1\hat{\kappa}_1, \tag{22b}$$

$$\dot{\hat{\kappa}}_2 = -\left(\frac{\sin(\chi-\gamma)}{d} - \lambda\beta_o\cos(\chi-\gamma)\right)\tilde{\chi} - \zeta_2\hat{\kappa}_2, \tag{22c}$$

$$\text{with } \hat{\kappa}_i(0) > 0, \ i = 0, 1, 2. \tag{22d}$$

with similar design parameters as before.

The following result can be derived:

Theorem 2. *By employing the guidance law (21), the resulting trajectories of the UAV (15) and the parameters in the adaptive law (22) are Uniformly Ultimately Bounded (UUB).*

Proof. See the Appendix. Tunability of the ultimate bound is elaborated at the end of the proof according to (47) and standard Lyapunov arguments [47, Sect. 4.8]. □

Remark 2. *Differently from* (9) *and* (12)*, no a priori knowledge of course time constant, wind and unmodelled dynamics is required; the gains $\hat{\kappa}_0$ and $\hat{\kappa}_1$ compensate online the uncertainty term* (17)*, stemming from the unmodelled term* (13)*; the gain $\hat{\kappa}_2$ plays the role of an estimator for the ground velocity. The course time constant α is estimated jointly via $\hat{\kappa}_0$, $\hat{\kappa}_1$, $\hat{\kappa}_2$ (as κ_0^*, κ_1^*, κ_2^* all contain $1/\alpha$). These estimation actions mark a difference with the standard adaptive-free VF and with other adaptive-free robust methods.*

Remark 3. *The adaptive laws in* (19) *and* (22) *reveal that the control gains adjust automatically according to the tracking error, thanks to the effect of the stabilizing leakage terms $-\zeta_i\hat{\kappa}_i$, $i = 0, 1, 2$. In other words, the adaptive laws keep a balance between increasing the estimates when the error is large, and keep the estimates bounded. As $\zeta_0, \zeta_1, \zeta_2$ become smaller, adaptation is faster. However, this might lead to larger gains $\hat{\kappa}_0, \hat{\kappa}_1\hat{\kappa}_2$ (i.e. the uncertainty can be overestimated) and high control input. This indicates a trade-off between small control inputs and robustness to unmodeled dynamics.*

Remark 4. *The proposed guidance laws* (18) *and (21) share a structure similar to (adaptive) sliding mode*

$$\chi^c = \chi - \Lambda\tilde{\chi} + \dot{\chi}^d - \rho\,\mathrm{sat}\left(\frac{\tilde{\chi}}{\varepsilon}\right). \tag{23}$$

The main differences are that $\dot{\chi}^d$ is given a priori in adaptive sliding mode control (whereas we include adaptation due to the uncertainty in V_g), and that ρ estimates a constant bound for the uncertainty (whereas we estimate a state-dependent bound). First-order dynamics (5) *usually assume the rudder loop to be well tuned and damped. If this is not the case and sideslip dynamics generate moderately damped second-order Dutch roll dynamics (e.g. triggered by cross-wind) [48], one can in principle consider such effects as additional unmodelled dynamics. Accordingly, one could consider a more complex description of the uncertainty by adding extra terms, e.g. a quadratic term*

$$|\Delta(\chi)| \leq \kappa_0 + \kappa_1\left|\tilde{\chi}\right| + \kappa_q\left|\tilde{\chi}\right|^2 \tag{24}$$

with unknown scalars κ_0, κ_1, κ_q. Besides the previously introduced adaptive laws for $\hat{\kappa}_0$, $\hat{\kappa}_1$, this eventually leads to an additional adaptation term

$$\rho = \hat{\kappa}_0 + \hat{\kappa}_1|\tilde{\chi}| + \hat{\kappa}_q|\tilde{\chi}|^2 \tag{25a}$$

$$\dot{\hat{\kappa}}_q = |\tilde{\chi}|^3 - \zeta_q\hat{\kappa}_q. \tag{25b}$$

Structural knowledge can be embedded in the upper bound (24)*, provided that the upper bound is linear in the uncertain parameters. Alternatively, the structure* (23) *can be modified*

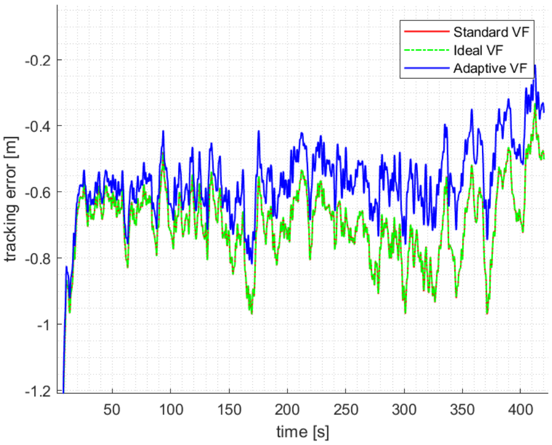

Figure 4. Straight line, Scenario 3: tracking error (the standard and ideal VF have similar performance and their lines overlap)

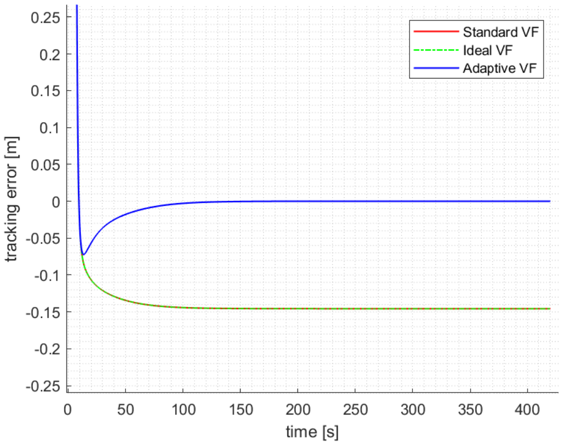

Figure 5. Orbit, Scenario 1: tracking error (the standard and ideal VF have the same performance and their lines overlap)

in the sense of sliding mode for second-order dynamics:

$$\chi^c = \chi - \Lambda\tilde{\xi} + \ddot{\chi}^d - \rho\,\mathrm{sat}\left(\frac{\tilde{\xi}}{\varepsilon}\right), \tag{26}$$

with $\tilde{\xi} = \dot{\tilde{\chi}} + \Lambda_2\tilde{\chi}$, *for* $\Lambda, \Lambda_2 > 0$. *This road is not explored here to avoid departing too much from the original vector field idea, and it can be a relevant future work.*

V. Experimental results

The performance of the proposed adaptive VF is assessed, as compared to the standard VF and to an ideal VF method, which differ for the following a priori knowledge (cf. Fig. 2):

- Standard VF: knowledge of the time constant α is needed, and only the constant wind component is known, i.e. $V_g(t) = ||V_a(t) + W(t)||$;
- 'Ideal' VF: knowledge of the time constant α is needed, and both constant and time-varying wind components are known, i.e $V_g(t) = ||V_a(t) + W'(t)||$ (we put 'ideal' in quotes because this approach still relies on simplified course dynamics, leading to degraded performance);
- Adaptive VF: the time constant α and all wind components are estimated.

The standard and ideal VF are inspired by the recent works [19], [33], where it is further illustrated that the VF in general does not give optimality guarantees in the sense of 'optimal control'. However, as the final goal of any guidance law is the minimization of a tracking error, such a tracking error can be considered a measure of optimality and evaluated experimentally. Experiments are carried out on a software-in-the-loop UAV platform where the functionalities of the ArduPilot autopilot are replicated in Matlab, and on a hardware-in-the-loop UAV platform where a PX4 autopilot hardware is connected to a Gazebo/ROS environment. Therefore, the experiments include the autopilot inner-loop dynamics (cascaded loops) embedded in ArduPilot/PX4, and allow to capture realistic effects of the inner loop on the guidance layer. Note that ArduPilot and PX4 are open-source suites constantly updated by a large UAV community, i.e. they represent the newest state of the art in the field.

The experiments offer a way to compare different sliding mode techniques in view of the following facts:

- The standard VF is essentially a sliding mode control method which assumes parametric knowledge of the course time constant and the nominal wind;
- The 'ideal' VF is also a sliding mode control method, but with more knowledge of the wind disturbance;
- Our adaptive VF is an advanced adaptive sliding mode control without parametric knowledge. Yet, it is different from standard adaptive sliding mode control since the latter still requires nominal parametric knowledge and assumes the uncertainty to be bounded a priori.

A. Software-in-the-loop experiments

The 6 degrees-of-freedom fixed-wing UAV and wind dynamics have been implemented in a Matlab software-in-the-loop UAV platform developed at TU Delft, which replicates the open-source ArduPilot autopilot code (cf. [49] for implementation details and for all the details about the UAV model, which is a based on a Hobby-King Bixler UAV). We take the following environmental conditions: constant wind amplitude is $W = 4$ m/s with wind angle $\psi_W = 230°$; and a Dryden turbulence [9, Sect. 4.4]. To draw conclusions on the effectiveness of adaptation in different conditions, all environmental conditions have been combined to obtain three wind scenarios, summarized in Table I.

The reader is referred to previous work by the same authors [33] to see how high-order state-dependent unmodelled dy-

Table I
Flight environmental conditions

Scenario	Constant wind	Turbulence
#1	No	No
#2	Yes	No
#3	Yes	Yes

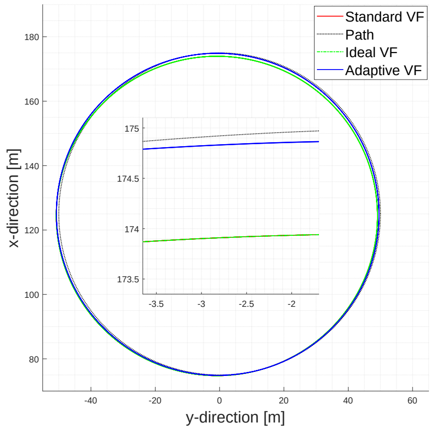

Figure 6. Orbit, Scenario 2: path in x-y plane (the small box is a zoom to highlight the improved tracking of the adaptive VF)

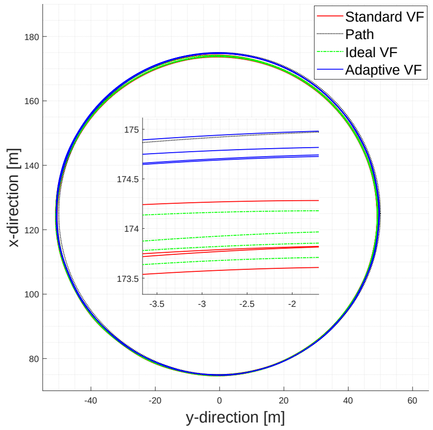

Figure 7. Orbit, Scenario 3: path in x-y plane (the small box is a zoom to highlight the improved tracking of the adaptive VF)

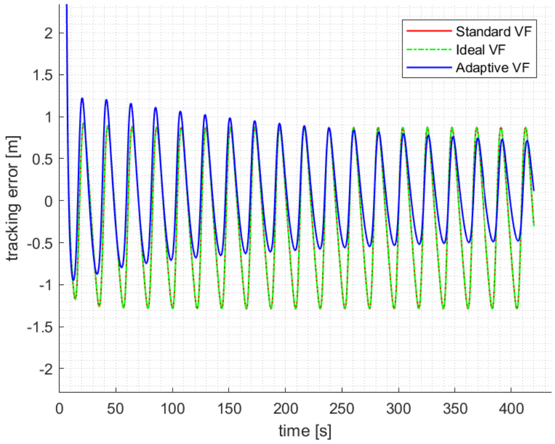

Figure 8. Orbit, Scenario 2: tracking error (the standard and ideal VF have the same performance and their lines overlap)

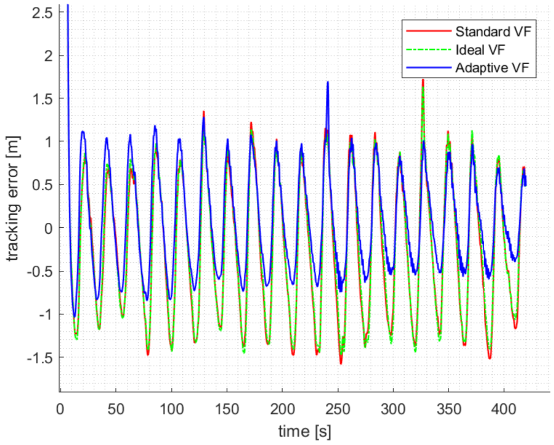

Figure 9. Orbit, Scenario 3: tracking error (the standard and ideal VF have similar performance and their lines almost overlap)

namics arise from approximating the Bixler course dynamics as first-order dynamics (5). The first-order time constant of the course dynamics can be estimated as $\alpha = 0.4578$. Both the standard and the ideal VF use this time constant.

Table II
PARAMETERS OF THE GUIDANCE LAWS

χ^{∞}	k	ε	κ	ζ_0, ζ_1	ζ_2
$\pi/2$	0.1 m^{-1}	1 rad	$\pi/2$ rad^2/s	0.01	0.001

The performance of the standard, adaptive and ideal VF are first evaluated on primitive paths (straight line and orbit), using the root mean square (RMS) steady-state tracking error calculated in the last portion of the path when e_{py} or $\tilde{d}$ have converged. The parameters χ^{∞}, k, ε, κ, ζ_0, ζ_1 and ζ_2 in Table II have been tuned so as to find a good compromise between convergence speed and smooth response.

Tables III and IV (straight line and orbit, respectively) highlight how the proposed adaptive VF outperforms, in all scenarios, the standard and the ideal VF. Note that in Scenarios 1 and 2 the standard and the ideal VF have exactly the same performance since there is no wind perturbation. For the straight line case, Fig. 4 clearly shows that the adaptive VF better counteracts with time the effect of the wind in Scenario 3 (38% improvement). Even with exact knowledge of the wind, the ideal VF performs quite poorly, due to the inaccurate knowledge of α (the adaptive VF again gives 38% improvement). Something similar also occurs in Scenario 2 (constant wind) and will not be shown due to space limitations.

For the orbit case, Fig. 5 clearly shows that the standard

Table III
STRAIGHT-LINE RMS TRACKING ERRORS (IN PARENTHESES IS THE LOSS OF PERFORMANCE AGAINST THE ADAPTIVE VF)

Scenario	Standard VF RMS error	'Ideal' VF RMS error	Adaptive VF RMS error
#1	0 **(+0%)**	0 **(+0%)**	0
#2	0.654 **(+38%)**	0.653 **(+38%)**	0.472
#3	0.673 **(+38%)**	0.673 **(+38%)**	0.488

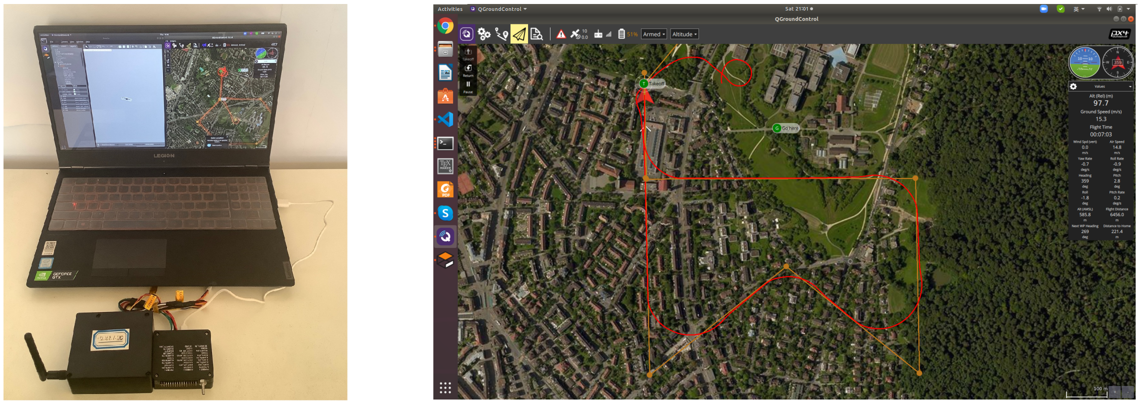

Figure 10. Setup for hardware-in-the-loop experiments: a PX4/Raspberry Pi 3B+ controller uses MAVROS node to share data with a Gazebo 3D simulator

and the ideal VF have a steady-state tracking error: such error is completely removed by the adaptive VF. A significant reduction of the tracking error (76-89%) by the adaptive VF also occurs in Scenarios 2 and 3, which are depicted in Figs. 6-9 in terms of tracking error and path in the x-y plane. In all cases it can be seen that the adaptation mechanism reduces the oscillations of the error: oscillations are present due to the fact that the wind effect changes when the UAV is travelling along the orbit. Due space limitations, the adaptive gains $\kappa_0, \kappa_1 \kappa_2$ are not shown, but one can verify the boundedness of the gains directly from the adaptive laws (19) and (22): if the tracking error $\tilde{\chi}$ is bounded, then $\hat{\kappa}_i$s are bounded using bounded-input-bounded-output notions, as ζ_is are positive constants.

Table IV
ORBIT RMS TRACKING ERRORS (IN PARENTHESES IS THE LOSS OF PERFORMANCE AGAINST THE ADAPTIVE VF)

Scenario	Standard VF RMS error	'Ideal' VF RMS error	Adaptive VF RMS error
#1	0.146 $(+\infty)$	0.146 $(+\infty)$	0
#2	0.776 **(+76%)**	0.776 **(+76%)**	0.441
#3	0.821 **(+89%)**	0.798 **(+84%)**	0.434

B. *Hardware-in-the-loop experiments*

A hardware-in-the-loop UAV platform is set up using the PX4 open-source flight controller with Raspberry Pi 3B+, ROS with MAVROS communication node (to communicate with PX4), and Gazebo as a 3D UAV simulator (cf. Fig. 10, left). PX4 is another popular autopilot suite: its inner-loop dynamics implement an TECS-L1 guidance law[1]: also, it allows to program in C++ other control laws: in this study we programmed both the standard VF and the adaptive VF in the PX4/Raspberry Pi 3B+ hardware (the ideal VF could not be programmed because the Gazebo wind environment can provide the wind time-varying component as a measurement).

Gazebo is used not only as a 3D simulator for rendering of environments, but also as a physical simulator of the UAV dynamics in 6 degrees of freedom. The UAV model is generated in Gazebo following the tutorial[2]: it is a 1.5kg standard structure fixed-wing UAV including aileron, rudder and elevator. The rotor is one puller at the head of the UAV and the airspeed is in the range [10 - 25]m/s (refer to the template[3]). The subsystems are connected as follows: Gazebo simulates and visualizes the world environment and the UAV, and it provides the sensor data to PX4; PX4 calculates the guidance commands depending on the embedded algorithm and send them back to Gazebo; finally, Gazebo delivers the commands to the UAV after simulating the actuator dynamics. As compared to the software-in-the-loop experiments, the hardware-in-the-loop UAV platform is also able to simulate state estimation errors (GPS and IMU measurement errors and the sensor fusion layer) which therefore add more realism to the experiments[4].

Similarly to the software-in-the-loop experiments, we define several wind scenarios and paths to test the performance in different environments. We have a scenario with average wind 2 m/s with variance of 0.5 m/s and direction $\psi_W = 45°$ (Scenario #4), and a scenario with average wind of 5 m/s, direction $\psi_W = 45°$, variance of 0.5 m/s and gusts up to 7 m/s (Scenario #5). We define three paths: a straight line path, an orbit path, and a combined path with lines and orbits (cf. Fig. 10, right). The results of the guidance laws are shown in Table V in terms of RMS error. Notice that the standard VF is implemented in two conditions: one with $\alpha = 0.4578$, and one where α has been carefully tuned so as to improve performance. Because the adaptive VF is able to improve even over the optimized standard VF, this further validates the effectiveness of the proposed strategy: even if the optimized α makes the standard VF at least four times better, still 3-19% improvements are observed thanks to adaptation. As compared to the previous tables, it can be seen that Scenario #5 is quite extreme for the UAV, but still the proposed adaptive VF outperforms all strategies. The TECS-L1 guidance works good for orbit following under low wind (Scenario #4, only 1% degradation) but is less effective for straight line and high wind, 102-462% degradation).

[1] Total energy control system with position control based on L1 norm https://docs.px4.io/master/en/config_fw/advanced_tuning_guide_fixedwing.html

[2] http://gazebosim.org/tutorials

[3] https://github.com/PX4/PX4-SITL_gazebo/blob/e580bbcd1eb6902c658ed3ece3b3b28dfd57eb17/models/plane/plane.sdf.jinja

[4] https://docs.px4.io/master/en/simulation/gazebo.html

Table V
3D SIMULATION RMS TRACKING ERRORS (IN PARENTHESES IS THE LOSS OF PERFORMANCE AGAINST THE ADAPTIVE VF)

	TECS-L1 RMS error	Standard VF RMS error ($\alpha = 0.4578$)	Standard VF RMS error (optimized α)	Adaptive VF RMS error
Line				
#4	5.161 **(+462%)**	1.419 **(+54%)**	1.036 **(+13%)**	0.919
#5	11.62 **(+102%)**	12.20 **(+112%)**	5.932 **(+3%)**	5.762
Orbit				
#4	2.319 **(+1%)**	2.613 **(+14%)**	2.370 **(+3%)**	2.300
#5	5.810 **(+102%)**	6.105 **(+112%)**	2.966 **(+3%)**	2.881
Combi.				
#4	6.705 **(+333%)**	4.416 **(+185%)**	1.839 **(+19%)**	1.548
#5	18.95 **(+233%)**	16.68 **(+194%)**	5.804 **(+2%)**	5.683

Overall the simulations show that the proposed adaptive VF, by compensating for the lack of knowledge in course dynamics and wind environment, can bring to improved guidance performance in several wind and path scenarios.

VI. CONCLUSIONS

As compared to state-of-the-art guidance for fixed-wing UAVs, this work has proposed a novel guidance law that does not require precise knowledge of the course time constant, while the course dynamics can be affected by state-dependent uncertainty representing couplings. A dedicated control design and stability analysis was given to address these challenges. The effectiveness of the proposed method in handling such uncertainty was tested on software-in-the-loop and hardware-in-the-loop UAV platforms, showing that the proposed method outperforms several guidance approaches relying on precise UAV dynamics. An interesting question is the level of uncertainty that makes the system fail: to the best of our experience, failure will be largely dependent on how the underlying autopilot layer (low-level control) is tuned. For our autopilot (tuned via the AutoTune procedure[5] of Ardupilot), we never experienced system failure. It is intuitive to expect that when the autopilot layer is poorly tuned, any guidance algorithm can do little to cope with this situation: investigating this point in an analytic or numerical way could be an interesting future work. Further connecting the vector field idea to higher order (adaptive) sliding mode [42], [50] is another interesting topic for further study.

APPENDIX

Before starting the analysis, let us notice that the combination of the adaptive laws (19a)-(19b), (22a)-(22b) and the initial conditions (19d),(22d) imply that

$$\hat{\kappa}_0(t), \hat{\kappa}_1(t) \geq 0, \forall t \geq 0. \tag{27}$$

for both straight line and orbit path.

PROOF OF THEOREM 1 (STRAIGHT-LINE CASE)

The closed-loop stability in the straight-line case is analysed using the following Lyapunov function

$$W = \frac{1}{2}\overline{\alpha}\tilde{\chi}^2 + \frac{1}{2}\sum_{i=0}^{2}(\hat{\kappa}_i - \kappa^*)^2. \tag{28}$$

Define an overall uncertainty term

$$\Delta^c \triangleq \overline{\Delta} + \kappa_2^*\chi_\infty \frac{2}{\pi}\frac{k}{1+(ke_y)^2}\sin(\chi) \tag{29}$$

Observing the structure of $\mathrm{sat}(\cdot)$ as in (18a), the overall stability analysis is carried out for the following two cases, using the common Lyapunov function (28).

Case (i): $|\tilde{\chi}| \geq \epsilon$

Using (15) and (18), the time-derivative of (28) yields

$$\begin{aligned}\dot{W} &= \tilde{\chi}(-\chi + \chi^c + \Delta^c) + \sum_{i=0}^{2}(\hat{\kappa}_i - \kappa_i^*)\dot{\hat{\kappa}}_i \\ &\leq -\Lambda\tilde{\chi}^2 - (\hat{\kappa}_0 - \kappa_0^*)|\tilde{\chi}| - (\hat{\kappa}_1 - \kappa_1^*)|\tilde{\chi}|^2 \\ &\quad - (\hat{\kappa}_2 - \kappa_2^*)\chi_\infty\frac{2}{\pi}\beta_s\tilde{\chi}\sin(\chi) + \sum_{i=0}^{2}(\hat{\kappa}_i - \kappa_i^*)\dot{\hat{\kappa}}_i. \end{aligned} \tag{30}$$

From (19a)-(19c) we have

$$\sum_{j=0}^{1}(\hat{\kappa}_j - \kappa_j^*)\dot{\hat{\kappa}}_j = (\hat{\kappa}_j - \kappa_j^*)|\tilde{\chi}|^{j+1} - \zeta_j\hat{\kappa}_j^2 + \zeta_j\hat{\kappa}_j\kappa_j^*, \tag{31}$$

$$(\hat{\kappa}_2 - \kappa_2^*)\dot{\hat{\kappa}}_2 = (\hat{\kappa}_2 - \kappa_2^*)\chi_\infty\frac{2}{\pi}\beta_s\tilde{\chi}\sin(\chi) - \zeta_2\hat{\kappa}_2^2 + \zeta_2\hat{\kappa}_2\kappa_2^*. \tag{32}$$

The following simplifications can be made for $i = 0, 1, 2$

$$\begin{aligned}\hat{\kappa}_i\kappa_i^* - \hat{\kappa}_i^2 &= -\left(\frac{\hat{\kappa}_i}{\sqrt{2}} - \frac{\kappa_i^*}{\sqrt{2}}\right)^2 - \frac{\hat{\kappa}_i^2}{2} + \frac{{\kappa_i^*}^2}{2} \\ &\leq -\left(\frac{\hat{\kappa}_i}{\sqrt{2}} - \frac{\kappa_i^*}{\sqrt{2}}\right)^2 + \frac{{\kappa_i^*}^2}{2}. \end{aligned} \tag{33}$$

Substituting (31)-(33) into (30) yields

$$\dot{W} \leq -\Lambda\tilde{\chi}^2 - \sum_{i=0}^{2}\left(\frac{\zeta_i(\hat{\kappa}_i - \kappa_i^*)^2}{2} - \frac{\zeta_i{\kappa_i^*}^2}{2}\right). \tag{34}$$

Using the definition of W in (28) yields

$$\dot{W} \leq -\overline{\varrho}W + \frac{1}{2}\sum_{i=0}^{2}\zeta_i{\kappa_i^*}^2, \tag{35}$$

where $\overline{\varrho} \triangleq \frac{\min_i\{\Lambda,\ \zeta_i/2\}}{\max\{\overline{\alpha}/2,\ 1/2\}} > 0$ by design.

Define a scalar $0 < \delta < \overline{\varrho}$. Then, $\dot{W}$ in (35) simplifies to

$$\dot{W} \leq -\delta W - (\overline{\varrho} - \delta)W + \frac{1}{2}\sum_{i=0}^{2}\zeta_i{\kappa_i^*}^2. \tag{36}$$

Defining a scalar $\mathcal{B}_1 \triangleq \frac{\sum_{i=0}^{2}\zeta_i{\kappa_i^*}^2}{2(\overline{\varrho}-\delta)}$, it can be noticed that $\dot{W} \leq -\delta W$ when $W \geq \mathcal{B}_1$.

Case (ii): $|\tilde{\chi}| < \epsilon$.

Using (15) and (18), for this case we have

$$\dot{W} \leq -\Lambda\tilde{\chi}^2 - \rho\frac{|\tilde{\chi}|^2}{\epsilon} + |\bar{\Delta}||\tilde{\chi}| + (\hat{\kappa}_2 - \kappa_2^*)\beta_s\tilde{\chi} + \sum_{i=0}^{2}(\hat{\kappa}_i - \kappa_i^*)\dot{\hat{\kappa}}_i$$

[5] https://ardupilot.org/plane/docs/automatic-tuning-with-autotune.html

$$\leq -\Lambda\tilde{\chi}^2 + \kappa_0^*|\tilde{\chi}| + \kappa_1^*|\tilde{\chi}|^2 + (\hat{\kappa}_2 - \kappa_2^*)\chi_\infty \frac{2}{\pi}\beta_s \sin(\chi)\tilde{\chi} + \sum_{i=0}^{2}(\hat{\kappa}_i - \kappa_i^*)\dot{\hat{\kappa}}_i \tag{37}$$

Then, following the same lines of proof as in Case (i) we have

$$\dot{W} \leq -\delta W - (\overline{\varrho} - \delta)W + \frac{1}{2}\sum_{i=0}^{2}\zeta_i {\kappa_i^*}^2 + \hat{\kappa}_0|\tilde{\chi}| + \hat{\kappa}_1|\tilde{\chi}|^2. \tag{38}$$

In Case (ii) we have $|\tilde{\chi}| < \epsilon$. From (19a)-(19b) it can be noted that $|\tilde{\chi}| \in \mathcal{L}_\infty \Rightarrow \hat{\kappa}_0, \hat{\kappa}_1 \in \mathcal{L}_\infty$. Therefore, $\exists \varsigma \in \mathbb{R}^+$ such that $(\hat{\kappa}_0|\tilde{\chi}| + \hat{\kappa}_1|\tilde{\chi}|^2) \leq \varsigma$, yielding

$$\dot{W} \leq -\delta W - (\overline{\varrho} - \delta)W + \frac{1}{2}\sum_{i=0}^{2}\zeta_i {\kappa_i^*}^2 + \varsigma \tag{39}$$

and $\dot{W} \leq -\delta W$ holds when $W \geq \mathcal{B}_2 \triangleq \frac{\frac{1}{2}\sum_{i=0}^{2}\zeta_i {\kappa_i^*}^2 + \varsigma}{\overline{\varrho} - \delta}$.

Observing the results of Cases (i) and (ii) ((36) and (39)), we get $\dot{W} \leq -\delta W$ when $W \geq \max\{\mathcal{B}_1,\ \mathcal{B}_2\}$ and the closed-loop system is UUB, implying $\tilde{\chi}, \hat{\kappa}_i \in \mathcal{L}_\infty$ for $i = 0, 1, 2$. Further, the Lyapunov function as in (28) yields $W \geq (1/2)\overline{\alpha}\tilde{\chi}^2$. Therefore, following the definition of ultimate bound as in [47, Sect. 4.8], the ultimate bound $\mathcal{B}_s$ on straight line path tracking error $\tilde{\chi}$ is found to be

$$\mathcal{B}_s = \sqrt{\frac{2\max\{\mathcal{B}_1,\ \mathcal{B}_2\}}{\overline{\alpha}}}. \tag{40}$$

Tunability: the ultimate bound on the path tracking error can be considered as a performance indicator. From the structures of the error bounds $\mathcal{B}_1$ and $\mathcal{B}_2$ (as below (36) and (39)), one can derive that a high value of Λ and low values of ζ_i improve tracking accuracy. However, it should be noticed that increasing Λ or decreasing ζ_i result in higher control input (due to the larger values of ρ): the trade-off between tracking error and control effort is standard in control, and requires to tune these parameters according to the application requirements.

Proof of Theorem 2 (Orbit case)

The stability analysis for the orbit path follows similar steps of straight path case, with Lyapunov function (28) and *overall uncertainty term* for the orbital path as

$$\Delta^c \triangleq \overline{\Delta} + \kappa_2^*\left(\frac{\sin(\chi - \gamma)}{d} + \lambda\beta_o\cos(\chi - \gamma)\right) \tag{41}$$

Observing the structure of $\mathrm{sat}(\cdot)$ as in (21a), the overall stability analysis is carried out for the following two cases, using the common Lyapunov function (28).

Case (i): $|\tilde{\chi}| \geq \epsilon$. Using (15) and (20), we get

$$\begin{aligned}\dot{W} &= \tilde{\chi}(-\chi + \chi^c + \Delta^c) + \sum_{i=0}^{2}(\hat{\kappa}_i - \kappa_i^*)\dot{\hat{\kappa}}_i \\ &\leq -\Lambda\tilde{\chi}^2 - (\hat{\kappa}_0 - \kappa_0^*)|\tilde{\chi}| - (\hat{\kappa}_1 - \kappa_1^*)|\tilde{\chi}|^2 + \sum_{i=0}^{2}(\hat{\kappa}_i - \kappa_i^*)\dot{\hat{\kappa}}_i \\ &\quad + (\hat{\kappa}_2 - \kappa_2^*)\left(\frac{\sin(\chi - \gamma)}{d} + \lambda\beta_o\cos(\chi - \gamma)\right)\tilde{\chi}.\end{aligned} \tag{42}$$

From (22a)-(22c) we have

$$\sum_{j=0}^{1}(\hat{\kappa}_j - \kappa_j^*)\dot{\hat{\kappa}}_j = (\hat{\kappa}_j - \kappa_j^*)|\tilde{\chi}|^{j+1} - \zeta_j\hat{\kappa}_j^2 + \zeta_j\hat{\kappa}_j\kappa_j^*, \tag{43}$$

$$\begin{aligned}(\hat{\kappa}_2 - \kappa_2^*)\dot{\hat{\kappa}}_2 &= (\hat{\kappa}_2 - \kappa_2^*)\tilde{\chi}\left(\frac{\sin(\chi - \gamma)}{d} + \lambda\beta_o\cos(\chi - \gamma)\right) \\ &\quad - \zeta_2\hat{\kappa}_2^2 + \zeta_2\hat{\kappa}_2\kappa_2^*.\end{aligned} \tag{44}$$

The same simplifications (33) apply to the orbit case, leading along similar steps to

$$\dot{W} \leq -\delta W - (\overline{\varrho} - \delta)W + \frac{1}{2}\sum_{i=0}^{2}\zeta_i {\kappa_i^*}^2. \tag{45}$$

Defining the scalar $\mathcal{B}_1$ as before, we have $\dot{W} \leq -\delta W$ when $W \geq \mathcal{B}_1$.

Case (ii): $|\tilde{\chi}| < \epsilon$. Using (15) and (20), for this case we get

$$\begin{aligned}\dot{W} &\leq -\Lambda\tilde{\chi}^2 - \rho(|\tilde{\chi}|^2/\epsilon) + |\bar{\Delta}||\tilde{\chi}| + \sum_{i=0}^{2}(\hat{\kappa}_i - \kappa_i^*)\dot{\hat{\kappa}}_i \\ &\quad + (\hat{\kappa}_2 - \kappa_2^*)\left(\frac{\sin(\chi - \gamma)}{d} + \lambda\beta_o\cos(\chi - \gamma)\right)\tilde{\chi} \\ &\leq -\Lambda\tilde{\chi}^2 + \kappa_0^*|\tilde{\chi}| + \kappa_1^*|\tilde{\chi}|^2 + \sum_{i=0}^{2}(\hat{\kappa}_i - \kappa_i^*)\dot{\hat{\kappa}}_i \\ &\quad + (\hat{\kappa}_2 - \kappa_2^*)\left(\frac{\sin(\chi - \gamma)}{d} + \lambda\beta_o\cos(\chi - \gamma)\right)\tilde{\chi}.\end{aligned} \tag{46}$$

We obtain that $\exists \varsigma \in \mathbb{R}^+$ such that $(\hat{\kappa}_0|\tilde{\chi}| + \hat{\kappa}_1|\tilde{\chi}|^2) \leq \varsigma$, giving

$$\dot{W} \leq -\delta W - (\overline{\varrho} - \delta)W + \frac{1}{2}\sum_{i=0}^{2}\zeta_i {\kappa_i^*}^2 + \varsigma \tag{47}$$

and $\dot{W} \leq -\delta W$ when $W \geq \mathcal{B}_2$. The results (45) and (47) reveal that $\dot{W} \leq -\delta W$ when $W \geq \max\{\mathcal{B}_1,\ \mathcal{B}_2\}$ and the closed loop is UUB, implying $\tilde{\chi}, \hat{\kappa}_i \in \mathcal{L}_\infty$ for $i = 0, 1, 2$. Tunability: following similar lines of Theorem 1, the ultimate bound $\mathcal{B}_o$ on path tracking error is analogous to (40), i.e.

$$\mathcal{B}_o = \sqrt{\frac{2\max\{\mathcal{B}_1,\ \mathcal{B}_2\}}{\overline{\alpha}}}. \tag{48}$$

Therefore, similar trade-offs arise: increasing Λ or decreasing ζ_i result in smaller ultimate bound but may result in higher control.

References

[1] J. M. Levin, A. A. Paranjape, and M. Nahon, "Agile maneuvering with a small fixed-wing unmanned aerial vehicle," *Robotics and Autonomous Systems*, vol. 116, pp. 148 – 161, 2019.

[2] D. Mehanovic, D. Rancourt, and A. L. Desbiens, "Fast and efficient aerial climbing of vertical surfaces using fixed-wing UAVs," *IEEE Robotics and Automation Letters*, vol. 4, no. 1, pp. 97–104, 2019.

[3] P. Sun, B. Zhu, Z. Zuo, and M. V. Basin, "Vision-based finite-time uncooperative target tracking for uav subject to actuator saturation," *Automatica*, vol. 130, p. 109708, 2021.

[4] H. J. Kim, M. Kim, H. Lim, C. Park, S. Yoon, D. Lee, H. Choi, G. Oh, J. Park, and Y. Kim, "Fully autonomous vision-based net-recovery landing system for a fixed-wing UAV," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 4, pp. 1320–1333, 2013.

[5] L. Fusini, T. I. Fossen, and T. A. Johansen, "Nonlinear observers for GNSS- and camera-aided inertial navigation of a fixed-wing UAV," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 5, pp. 1884–1891, 2018.

[6] J. Chang, J. Cieslak, J. Dávila, J. Zhou, A. Zolghadri, and Z. Guo, "A two-step approach for an enhanced quadrotor attitude estimation via IMU data," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 3, pp. 1140–1148, 2018.

[7] S. Baldi, S. Roy, K. Yang, and D. Liu, "An underactuated control system design for adaptive autopilot of fixed-wing drones," *IEEE/ASME Transactions on Mechatronics*, pp. 1–12, 2022.

[8] A. Bosso, C. Conficoni, D. Raggini, and A. Tilli, "A computational-effective field-oriented control strategy for accurate and efficient electric propulsion of unmanned aerial vehicles," *IEEE/ASME Transactions on Mechatronics*, pp. 1–1, 2020.

[9] R. W. Beard and T. W. McLain, *Small Unmanned Aircraft: Theory and Practice*. Princeton University Press, 2012.

[10] N. Cho and Y. Kim, "Optimality of augmented ideal proportional navigation for maneuvering target interception," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 2, pp. 948–954, 2016.

[11] D. Invernizzi, M. Lovera, and L. Zaccarian, "Dynamic attitude planning for trajectory tracking in thrust-vectoring UAVs," *IEEE Transactions on Automatic Control*, 2019.

[12] A. Galffy, M. Böck, and A. Kugi, "Nonlinear 3D path following control of a fixed-wing aircraft based on acceleration control," *Control Engineering Practice*, vol. 86, pp. 56 – 69, 2019.

[13] M. Mammarella, E. Capello, F. Dabbene, and G. Guglieri, "Sample-based SMPC for tracking control of fixed-wing UAV," *IEEE Control Systems Letters*, vol. 2, no. 4, pp. 611–616, 2018.

[14] J.-M. Kai, T. Hamel, and C. Samson, "A unified approach to fixed-wing aircraft path following guidance and control," *Automatica*, vol. 108, p. 108491, 2019.

[15] B. Li, J. Sun, W. Zhou, C. Y. Wen, K. H. Low, and C. K. Chen, "Transition optimization for a VTOL tail-sitter UAV," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 5, pp. 2534–2545, 2020.

[16] D. R. Nelson, D. B. Barber, T. W. McLain, and R. W. Beard, "Vector field path following for miniature air vehicles," *IEEE Transactions on Robotics*, vol. 23, no. 3, pp. 519–529, 2007.

[17] 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 Systems Magazine*, vol. 34, no. 1, pp. 42–59, 2014.

[18] K. Tanaka, M. Tanaka, A. Iwase, and H. O. Wang, "A rational polynomial tracking control approach to a common system representation for unmanned aerial vehicles," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 2, pp. 919–930, 2020.

[19] W. Yao, H. G. de Marina, B. Lin, and M. Cao, "Singularity-free guiding vector field for robot navigation," *IEEE Transactions on Robotics*, vol. 37, no. 4, pp. 1206–1221, 2021.

[20] V. M. Goncalves, L. C. A. Pimenta, C. A. Maia, B. C. O. Dutra, and G. A. S. Pereira, "Vector fields for robot navigation along time-varying curves in n -dimensions," *IEEE Transactions on Robotics*, vol. 26, no. 4, pp. 647–659, 2010.

[21] J. Gutmann, E. Eade, P. Fong, and M. E. Munich, "Vector field SLAM—localization by learning the spatial variation of continuous signals," *IEEE Transactions on Robotics*, vol. 28, no. 3, pp. 650–667, 2012.

[22] J. Kwon and D. Chwa, "Hierarchical formation control based on a vector field method for wheeled mobile robots," *IEEE Transactions on Robotics*, vol. 28, no. 6, pp. 1335–1345, 2012.

[23] D. Tang, Q. Fang, L. Shen, and T. Hu, "Onboard detection-tracking-localization," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 3, pp. 1555–1565, 2020.

[24] V. P. Tran, F. Santoso, M. Garratt, and I. R. Petersen, "Distributed formation control using fuzzy self-tuning of strictly negative imaginary consensus controllers in aerial robotics," *IEEE/ASME Transactions on Mechatronics*, pp. 1–1, 2020.

[25] S. S. Mulgund and R. F. Stengel, "Optimal nonlinear estimation for aircraft flight control in wind shear," *Automatica*, vol. 32, no. 1, pp. 3 – 13, 1996.

[26] R. W. Beard, J. Ferrin, and J. Humpherys, "Fixed wing UAV path following in wind with input constraints," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 6, pp. 2103–2117, 2014.

[27] S. R. Arya, S. Rao, and B. Dattaguru, "Effect of asymmetric control constraints on fixed-wing UAV trajectories," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 3, pp. 1407–1419, 2019.

[28] R. Rysdyk, "Unmanned aerial vehicle path following for target observation in wind," *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 5, pp. 1092–1100, 2006.

[29] S. Baldi, D. Sun, G. Zhou, and D. Liu, "Adaptation to unknown leader velocity in vector-field uav formation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 58, no. 1, pp. 473–484, 2022.

[30] N. Cho, S. Lee, J. Kim, Y. Kim, S. Park, and C. Song, "Wind compensation framework for unpowered-aircraft using online waypoint correction," *IEEE Transactions on Aerospace and Electronic Systems*, 2019.

[31] T. I. Fossen, K. Y. Pettersen, and R. Galeazzi, "Line-of-sight path following for dubins paths with adaptive sideslip compensation of drift forces," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 820–827, 2015.

[32] Y. Wang, Y. Yue, M. Shan, L. He, and D. Wang, "Formation reconstruction and trajectory replanning for Multi-UAV patrol," *IEEE/ASME Transactions on Mechatronics*, pp. 1–1, 2021.

[33] B. Zhou, H. Satyavada, and S. Baldi, "Adaptive path following for unmanned aerial vehicles in time-varying unknown wind environments," in *2017 American Control Conference (ACC)*, 2017, pp. 1127–1132.

[34] S. Fari, X. Wang, S. Roy, and S. Baldi, "Addressing unmodelled path-following dynamics via adaptive vector field: a UAV test case," *IEEE Transactions on Aerospace and Electronic Systems*, 2019.

[35] Z. Zhen, G. Tao, C. Yu, and Y. Xue, "A multivariable adaptive control scheme for automatic carrier landing of UAV," *Aerospace Science and Technology*, vol. 92, pp. 714 – 721, 2019.

[36] A. P. Aguiar and J. P. Hespanha, "Trajectory-tracking and path-following of underactuated autonomous vehicles with parametric modeling uncertainty," *IEEE Transactions on Automatic Control*, vol. 52, no. 8, pp. 1362–1379, 2007.

[37] N. Hovakimyan and C. Cao, *$\mathcal{L}_1$ adaptive control theory: guaranteed robustness with fast adaptation*. Society for Industrial and Applied Mathematics, 2010.

[38] R. Chai, A. Tsourdos, A. Savvaris, S. Chai, Y. Xia, and C. L. P. Chen, "Design and implementation of deep neural network-based control for automatic parking maneuver process," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2020.

[39] ——, "Six-DOF spacecraft optimal trajectory planning and real-time attitude control: A deep neural network-based approach," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 11, pp. 5005–5013, 2020.

[40] Y. Shtessel, M. Taleb, and F. Plestan, "A novel adaptive-gain supertwisting sliding mode controller: Methodology and application," *Automatica*, vol. 48, no. 5, pp. 759–769, 2012.

[41] K. Lu and Y. Xia, "Finite-time attitude control for rigid spacecraft-based on adaptive super-twisting algorithm," *IET Control Theory & Applications*, vol. 8, no. 15, pp. 1465–1477, 2014.

[42] C. Edwards and Y. B. Shtessel, "Adaptive continuous higher order sliding mode control," *Automatica*, vol. 65, pp. 183–190, 2016.

[43] H. Obeid, L. M. Fridman, S. Laghrouche, and M. Harmouche, "Barrier function-based adaptive sliding mode control," *Automatica*, vol. 93, pp. 540–544, 2018.

[44] B. L. Stevens, F. L. Lewis, and E. N. Johnson, *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems, 3rd edition*. John Wiley & Sons, Inc., 2015.

[45] V. I. Utkin and A. S. Poznyak, "Adaptive sliding mode control with application to super-twist algorithm: Equivalent control method," *Automatica*, vol. 49, no. 1, pp. 39–47, 2013.

[46] H. Oh, S. Kim, H. Shin, and A. Tsourdos, "Coordinated standoff tracking of moving target groups using multiple UAVs," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 2, pp. 1501–1514, 2015.

[47] H. K. Khalil, "Nonlinear systems," *Prentice Hall*, 2002.

[48] A. A. Paranjape, S. Chung, and J. Kim, "Novel dihedral-based control of flapping-wing aircraft with application to perching," *IEEE Transactions on Robotics*, vol. 29, no. 5, pp. 1071–1084, 2013.

[49] S. Farì, "Guidance and control for a fixed-wing UAV," Master's thesis, Politecnico di Milano, 2017.

[50] Y. B. Shtessel, J. A. Moreno, and L. M. Fridman, "Twisting sliding mode control with adaptation: Lyapunov design, methodology and application," *Automatica*, vol. 75, pp. 229–235, 2017.

Ximan Wang received the B.Sc. degree in control system engineering from Taiyuan University, China, in 2014, and the M.Sc. degree in automotive control system engineering from University of Sheffield, in 2016. He was a Senior Engineer at Systems Engineering Research Institute, Beijing, China, and he is now pursuing the PhD at the Delft Center for Systems and Control, Delft University of Technology with research interests in adaptive optimization for control and UAV adaptive control.

Spandan Roy received the B.Tech. in electronics and communication engineering from Techno India (Salt Lake), West Bengal University of Technology, Kolkata, India, in 2011, the M.Tech. in mechatronics from Academy of Scientific and Innovative Research, New Delhi, India, in 2013, and the Ph.D. in control and automation from Indian Institute of Technology Delhi, India, in 2018. He is currently Assistant Professor with Robotics Research Center, International Institute of Information Technology Hyderabad, India. Previously, he was Postdoc Researcher with Delft Center for System and Control, TU Delft, The Netherlands. His research interests include artificial delay based control, adaptive-robust control, switched systems and its applications in Euler-Lagrange systems.

Stefano Farì received the B.Sc. and M.Sc. in automation engineering from Politecnico di Milano, Italy, performing his master thesis as guest researcher at the Delft Center for Systems and Control, Delft University of Technology, The Netherlands. He has worked at Piaggio Aerospace, Savona, Italy, as flight control system engineer and he is now working as GNC engineer at German Aerospace Center (DLR), Bremen, Germany.

Simone Baldi (M'14, SM'19) received the B.Sc. in electrical engineering, and the M.Sc. and Ph.D. in automatic control engineering from University of Florence, Italy, in 2005, 2007, and 2011. He is professor at School of Mathematics and School of Cyber Science and Engineering, Southeast University, with guest position at Delft Center for Systems and Control, TU Delft, where he was assistant professor. He was awarded outstanding reviewer for *Applied Energy* (2016) and *Automatica* (2017). He is a subject editor of *Int. Journal of Adaptive Control and Signal Processing* and associate editor of *IEEE Control Systems Letters*. His research interests are adaptive and learning systems with applications in unmanned vehicles and smart energy systems.