Addressing Unmodeled Path-Following Dynamics via Adaptive Vector Field
A UAV Test Case

Fari, Stefano; Wang, Ximan; Roy, Spandan; Baldi, Simone

DOI
10.1109/TAES.2019.2925487

Publication date
2020

Document Version
Accepted author manuscript

Published in
IEEE Transactions on Aerospace and Electronic Systems

Citation (APA)
Fari, S., Wang, X., Roy, S., & Baldi, S. (2020). Addressing Unmodeled Path-Following Dynamics via Adaptive Vector Field: A UAV Test Case. *IEEE Transactions on Aerospace and Electronic Systems*, *56*(2), 1613-1622. https://doi.org/10.1109/TAES.2019.2925487

Addressing Unmodelled Path-Following Dynamics via Adaptive Vector Field: a UAV Test Case

Stefano Farì, Ximan Wang, Spandan Roy, and Simone Baldi

***Abstract*—The actual performance of model-based path-following methods for Unmanned Aerial Vehicles (UAVs) show considerable dependence on the wind knowledge and on the fidelity of the dynamic model used for design. This work analyzes and demonstrates the performance of an adaptive Vector Field (VF) control law which can compensate for the lack of knowledge of the wind vector and for the presence of unmodelled course angle dynamics. Extensive simulation experiments, calibrated on a commercial fixed-wing UAV and proven to be realistic, show that the new VF method can better cope with uncertainties than its standard version. In fact, while the standard VF approach works perfectly for ideal first-order course angle dynamics (and perfect knowledge of the wind vector), its performance degrades in the presence of unknown wind or unmodelled course angle dynamics. On the other hand, the estimation mechanism of the proposed adaptive VF effectively compensates for wind uncertainty and unmodelled dynamics, sensibly reducing the path-following error as compared to the standard VF.**

***Index Terms*—Adaptive Vector Field, fixed-wing UAV, path-following, unmodelled course angle dynamics.**

I. INTRODUCTION

Born initially for military applications, Unmanned Aerial Vehicles (UAVs) have nowadays also civil applications, such as monitoring, aerial mapping, small cargo deliveries, search and rescue operations [1]. UAVs must rely on accurate path-following algorithms: wind disturbances, unmodelled dynamics, and the quality of sensing and control, are all critical limits to the achievable accuracy [2]–[5]. Taking into account that UAVs must operate in windy environments where wind speeds are 20-50% of the UAV airspeed, the design of high-performance path-following strategies is compelling.

Path-following techniques can be developed using geometric or control-theoretic approaches [6]. The first class include the pure pursuit and line-of-sight guidance laws [7]–[11], which make use of a virtual target point where the UAV is directed to. Control-theoretic techniques include PIDs, linear quadratic control, sliding-mode control, model predictive control, adaptive control and their variants [12]–[16]. The Vector Field (VF) approach [17]–[22] sits in between these two classes, by combining geometric reasoning with a sliding-mode technique. The goal of the VF approach is to drive the cross-track error to zero by providing a field of desired (inertial-referenced) course angles for each point around the desired path [17]. An extension of the VF to curves/lines in n-dimensional spaces was studied in [18]. In [19] the VF idea is extended to decentralized navigation with collision avoidance. In [20] a hybrid VF is proposed as a combination of Lyapunov analysis and geometry. In [21] the vector fields are made time-varying so that a team of UAVs can reach a rendezvous position. The VF-based strategy of [22] can consider input and state constraints. The VF idea has also been considered for simultaneous localization and mapping [23], [24]. In the survey [6], many path-following methods are applied to point-mass UAV kinematics. It is shown that VF achieves the lowest steady-state error at the price of tuning more parameters.

The actual performance of path-following methods show considerable dependence on the wind knowledge and on the fidelity of the dynamic model used for design. In particular, the standard VF method crucially relies on two assumptions: known constant wind and first-order course dynamics [17]. Both assumptions can be found in all aforementioned VF works [17]–[22], but unfortunately are seldom met in practice. The first assumption has been relaxed in [25] by augmenting the VF path-following control law with adaptation. However, relaxing the second assumption is a relevant but, to the best of the authors' knowledge, open problem. This work aims at answering the following question: can VF adaptation compensate not only for unknown winds, but also for unmodelled course angle dynamics?

To answer this question, we show in Section III how unmodelled course angle dynamics arise from the low-level autopilot layer: such dynamics are derived after reverse-engineering ArduPilot [26], a popular open-source autopilot. A complete fixed-wing UAV simulator is designed, whose system parameters are derived for a commercial UAV, the HobbyKing Bixler. Physical and low-level control parameters are derived for the Bixler and proven to be realistic. In Section IV the standard and adaptive VF strategies are discussed, while Section V provides extensive simulation comparisons. As compared to literature, the crucial contribution of this work is showing that the estimation mechanism of the adaptive VF effectively compensates for the unmodelled UAV course angle dynamics. The proposed method outperforms even 'ideal' VF strategies, relying on perfect knowledge of the wind vector, but based on simplified first-order course angle dynamics.

This research was partially supported by the "Fundamental Research Funds for The Central Universities"' under the project RECON-STRUCT, Grant No. 103009004 and by the joint TU Delft - CSSC Project MULTI-COORD *(corr. author: S. Baldi)*.

S. Farì is with German Aerospace Center (DLR), Bremen, Germany This research was performed while he was with Polytechnic of Milan and with Delft Center for Systems and Control, TU Delft stefano.fari@dlr.de

S. Roy is with Delft Center for Systems and Control, TU Delft, The Netherlands s.roy-2@tudelft.nl

X. Wang is with System Engineering Research Institute, China State Shipbuilding Corporation (CSSC), Beijing, China, and with Delft Center for Systems and Control, TU Delft, The Netherlands wangxm614@163.com

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

II. Preliminaries

State	Description
p_n	Inertial north position along $\mathbf{i}^i$ in $\mathcal{F}^i$
p_e	Inertial east position along $\mathbf{j}^i$ in $\mathcal{F}^i$
p_d	Inertial down position along $\mathbf{k}^i$ in $\mathcal{F}^i$
u	Body frame velocity along $\mathbf{i}^b$ in $\mathcal{F}^b$
v	Body frame velocity along $\mathbf{j}^b$ in $\mathcal{F}^b$
w	Body frame velocity along $\mathbf{k}^b$ in $\mathcal{F}^b$
ϕ	Roll angle defined with respect to $\mathcal{F}^{v2}$
θ	Pitch angle defined with respect to $\mathcal{F}^{v1}$
ψ	Yaw angle defined with respect to $\mathcal{F}^{v}$
p	Roll rate measured along $\mathbf{i}^b$ in $\mathcal{F}^b$
q	Pitch rate measured along $\mathbf{j}^b$ in $\mathcal{F}^b$
r	Yaw rate measured along $\mathbf{k}^b$ in $\mathcal{F}^b$

TABLE I: UAV states ($\mathcal{F}^i$ is the inertial frame according to the North-East-Down (NED) convention, $\mathcal{F}^b$ is the body frame, $\mathcal{F}^v$, $\mathcal{F}^{v1}$, $\mathcal{F}^{v2}$ are the vehicle frame and the intermediate vehicle frames arising from the ($\mathbf{k}^i$-$\mathbf{j}^{v1}$-$\mathbf{i}^{v2}$) Euler rotations).

A. Equations of motion

Any UAV can be modelled using a 6-DOF rigid body equations of motions [27], briefly recalled hereafter:

$$\begin{bmatrix}\dot{p}_n\\ \dot{p}_e\\ \dot{p}_d\end{bmatrix} = \mathcal{R}_{vb}^{-1}(\phi,\theta,\psi)\begin{bmatrix}u\\ v\\ w\end{bmatrix}, \quad \begin{bmatrix}\dot{u}\\ \dot{v}\\ \dot{w}\end{bmatrix} = \begin{bmatrix}rv-qw\\ pw-ru\\ qu-pv\end{bmatrix} + \frac{1}{m}\begin{bmatrix}f_x\\ f_y\\ f_z\end{bmatrix}$$

$$\begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix} = \begin{bmatrix}1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta)\\ 0 & \cos(\phi) & \sin(\phi)\\ 0 & \sin(\phi)/\cos(\theta) & \cos(\psi)/\cos(\theta)\end{bmatrix}\begin{bmatrix}p\\ q\\ r\end{bmatrix}$$

$$\begin{bmatrix}\dot{p}\\ \dot{q}\\ \dot{r}\end{bmatrix} = \begin{bmatrix}\Gamma_1 pq - \Gamma_2 qr\\ \Gamma_5 pr - \Gamma_6(p^2-r^2)\\ \Gamma_7 pq - \Gamma_1 qr\end{bmatrix} + \begin{bmatrix}\Gamma_3\mathcal{L} + \Gamma_4\mathcal{N}\\ \frac{1}{J_y}\mathcal{M}\\ \Gamma_4\mathcal{L} + \Gamma_8\mathcal{N}\end{bmatrix}$$

where the twelve state variables used to derive the equations of motion are in Table I. With $\mathcal{R}_{vb}$ we denote the rotation matrix from the vehicle to the body frame (located at the aircraft center of mass). The terms f_x, f_y, and f_z are the forces acting on $\mathbf{i}^b$, $\mathbf{j}^b$ and $\mathbf{k}^b$, respectively (comprising propulsion, aerodynamic, and gravity forces). Finally, $\mathcal{L}$, $\mathcal{M}$, $\mathcal{N}$ are the rolling, pitching and yawing moments about the same axes. The mass of the UAV is m and

$$\Gamma_1 = \frac{J_{xz}(J_x - J_y + J_z)}{J_x J_z - J_{xz}^2}, \quad \Gamma_2 = \frac{J_z(J_z - J_y) + J_{xz}^2}{J_x J_z - J_{xz}^2}$$

$$\Gamma_3 = \frac{J_z}{J_x J_z - J_{xz}^2}, \quad \Gamma_4 = \frac{J_{xz}}{J_x J_z - J_{xz}^2}, \quad \Gamma_5 = \frac{J_z - J_x}{J_y}$$

$$\Gamma_7 = \frac{(J_x - J_y)J_x + J_{xy}^2}{J_x J_z - J_{xz}^2}, \quad \Gamma_8 = \frac{J_x}{J_x J_z - J_{xz}^2}, \quad \Gamma_6 = \frac{J_{xy}}{J_y}$$

where the J-terms are components of the inertia tensor $\mathbf{J}$. The complete 6-DOF equations of motion, as well as more details on the equations, can be found in [27], [28].

B. The wind triangle

Let $\mathbf{V}_g$ be the UAV ground speed relative to the inertial frame. Airspeed $\mathbf{V}_a$, ground speed $\mathbf{V}_g$, and wind speed $\mathbf{V}_w$ are related via the so-called wind triangle (cf. Fig. 1)

$$\mathbf{V}_a = \mathbf{V}_g - \mathbf{V}_w. \tag{1}$$

An inertial-referenced angle, called course angle χ, is introduced, which represents the angle between the true north and the projection of $\mathbf{V}_g$ on the horizontal plane ($\mathbf{i}^b$,$\mathbf{j}^b$); χ constitutes the control variable for the guidance logic. In this work we model the wind as the composition of a steady-state, a dynamic, and a slowly time-varying wind

$$\mathbf{V}_w = \mathbf{V}_{w,s} + \mathbf{V}_{w,d} + \mathbf{V}_{w,v} \tag{2}$$

where the dynamic part $\mathbf{V}_{w,d}$ represents wind turbulence, obtained by passing white noise through appropriate forming filters [29], and the time-varying part $\mathbf{V}_{w,v}$ is taken as a sinusoidal or multi-sinusoidal term perturbing the constant wind amplitude and direction.

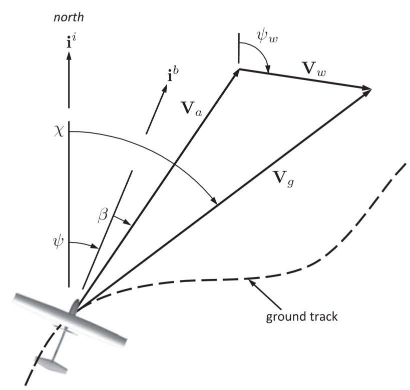


Fig. 1: The wind triangle (figure adapted from [28]). Here, β represents the side-slip angle of the UAV, χ the course angle, ψ the yaw angle, and ψ_w the wind angle.

III. UAV course angle dynamics

The primary goal of the autopilot layer is to provide low-level controllers to govern the UAV inertial position (p_n, p_e, p_d) and attitude (ϕ, θ, ψ). This implies computing the trim (equilibrium) states and inputs: the book [28] shows that, with the standard assumption of decoupled longitudinal/lateral dynamics, the linearized lateral dynamics around the trim equilibrium can be described by the following transfer functions:

$$\text{Roll angle} \quad \phi(s) = \frac{a_{\phi_2}}{s(s+a_{\phi_1})}\left(\delta_a(s) + \tilde{d}_\phi(s)\right) \tag{3}$$

$$\text{Course angle} \quad \chi(s) = \frac{g}{V_g s}\left(\phi(s) + \tilde{d}_\chi(s)\right) \tag{4}$$

for appropriate constants a_{ϕ_1}, a_{ϕ_2}, where δ_a is the aileron command and $\tilde{d}_\phi$ and $\tilde{d}_\chi$ are disturbances coming from cross-effects of neglected dynamics. Let us focus only on the lateral dynamics, most relevant to path following; details on the longitudinal UAV control loop can be found in [30].

A. Arising of unmodelled course angle dynamics

In practice, the cross-effects $\tilde{d}_\phi$ and $\tilde{d}_\chi$ are neglected to obtain first- or second-order linearized dynamics, and thus simplify the low-level control design via PID controllers [28]. ArduPilot employs a special nested structure for lateral control, depicted in Fig. 2. It consists of three nested loops: the inner

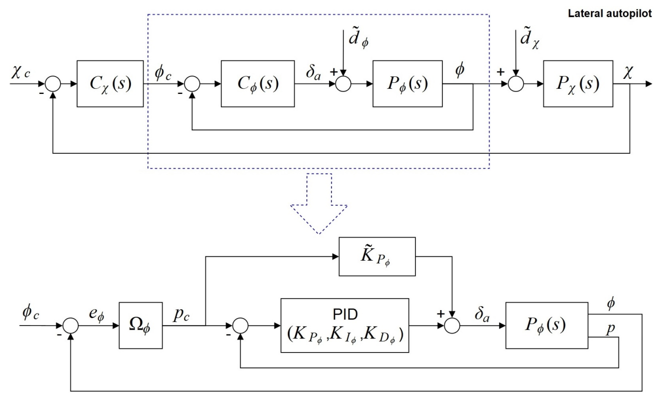


Fig. 2: UAV course and roll control (the variables with subscript c indicate controlled variables of the nested loops).

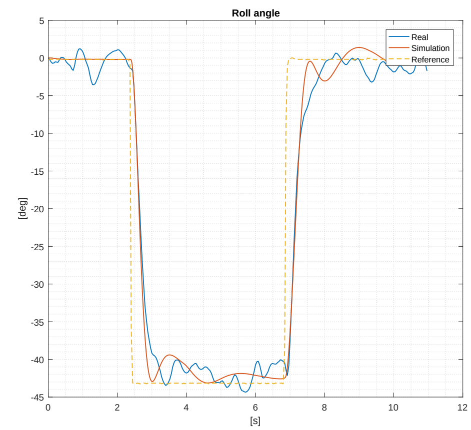


Fig. 3: Comparison of actual and simulated roll dynamics.

one controls the roll rate p; the second one controls the roll angle ϕ; the outer one controls the course angle χ. In the inner loop, a PID controller (whose gains K_{P_ϕ}, K_{I_ϕ}, K_{D_ϕ} are arranged in a special ad-hoc structure, see the Ardupilot documentation [31]) is combined with a feed-forward gain $\tilde{K}_{P_\phi}$; at the second loop there is a proportional gain Ω_ϕ. From the structure and parameters of all blocks in Fig. 2, the nested loop gives rise to a third-order transfer function

$$\frac{\phi(s)}{\phi_c(s)} = \frac{2017.8}{(s^2 + 8.467s + 44.88)(s + 45)} \tag{5}$$

where ϕ_c is the commanded roll angle in the inner loop. The numerical values for the parameters in (5) arise from $a_{\phi_1} = 9.344$, $a_{\phi_2} = 68.46$ (identified specifically for the Bixler), and the gains $K_{P_\phi} = 0.7$, $K_{I_\phi} = 0.1$, $K_{D_\phi} = 0.01$, $\tilde{K}_{P_\phi} = 0$, $\Omega_\phi = 2.22$ (obtained from the ArduPilot Autotuning function). The last high-frequency pole in (5) represent actuator dynamics identified specifically for the Bixler.

Remark 1: The advantage of using ArduPilot is the possibility to reverse-engineering the ArduPilot open-source software, so as to simulate the ArduPilot-controlled behavior with good accuracy. To check the validity of (5), the actual UAV closed-loop roll dynamics have been compared to the simulated roll dynamics: Fig. 3 shows that the responses from simulations and from the actual UAV Bixler are comparable.

From (4), neglecting $\tilde{d}_\chi$, we see that the transfer function from roll angle ϕ to course angle χ is just $g/(V_g s)$. For $V_g =$ 15 m/s and $g = 9.81$ m/s^2, this results in

$$\frac{\chi(s)}{\phi_c(s)} = \frac{\phi(s)}{\phi_c(s)} \frac{g}{V_g s} = \frac{0.654\phi(s)}{s\phi_c(s)}. \tag{6}$$

The loop is finally closed by a proportional controller $C_\chi(s) =$ 0.7 (whose value is chosen so as to guarantee a sufficient frequency separation between the inner and outer lateral loops), resulting in the final fourth-order course angle dynamics

$$\frac{\chi(s)}{\chi_c(s)} = \frac{923.72}{(s + 0.51)(s^2 + 7.97s + 40.38)(s + 44.99)} \tag{7}$$

where χ_c is the commanded course angle. Approximated first-order course dynamics can be obtained from neglecting high-order dynamics in (6), closing the loop with the same C_χ

$$\left(\frac{\chi(s)}{\chi_c(s)}\right)_{1st} = \frac{\frac{C_\chi g}{sV_g}}{1 + \frac{C_\chi g}{sV_g}} = \frac{\alpha_\chi}{s + \alpha_\chi} \Rightarrow \dot{\chi} = \alpha_\chi(\chi_c - \chi) \tag{8}$$

where $\alpha_\chi = 0.4578$ is the first-order time constant. Fig. 4 shows that the first-order approximation necessarily creates unmodelled dynamics: in particular, the -3dB bandwidth of (7) is 11% larger than the bandwidth of (8), clearly affecting the operating range of a UAV.

Remark 2: ArduPilot is not only a platform for low-level UAV control, but it also assists in capturing the motivation of this work. In fact, through the nested architecture of ArduPilot, we can quantify the distance between the first-order dynamics (8) *and the fourth-order dynamics* (7)*. The next step, answered in Sects. IV and V, is to discover how such a distance affects VF path-following performance, as the reported conventional VF methods rely on* (8)*.*

B. Ardupilot-controlled simulation platform

In order to design realistic path-following tests, a Matlab-based UAV simulation platform has been developed, which includes all UAV and environmental dynamics. All drag and lift coefficients of the UAV have been derived as look-up tables by means of USAF Digital DATCOM [32], after inputting the geometric characteristics of the Bixler. Additionally, by reverse-engineering the ArduPilot autopilot layer, the UAV simulator includes all low-level controllers (roll, pitch, altitude, airspeed, side-slip and course). A screenshot of the UAV simulator is in Fig. 5, together with the actual Bixler during some field tests performed to identify pitch and roll dynamics.

IV. Vector-Field Path Following

As standard in literature, straight-line and orbit path are considered for VF path following [28]. In the following we describe the standard VF method along with its issues (section IV.A) and the proposed adaptive VF (section IV.B).

A. Standard Vector Field

The control laws for standard Vector Field [17] are recalled.

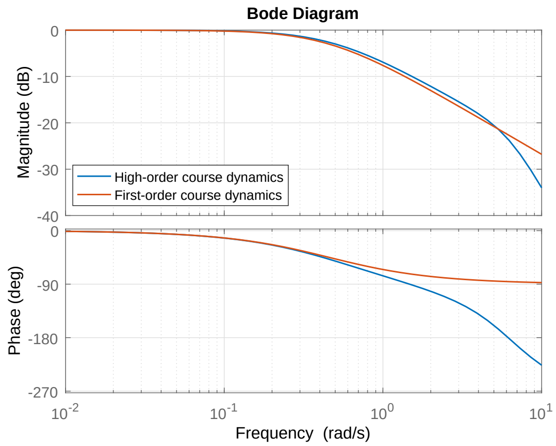


Fig. 4: Comparison of the Bode diagrams of $\chi(s)/\chi_c(s)$.

1) Straight Line Following: The reference course vector field meant to drive the UAV on the straight line path is

$$\chi_d(e_{py}) = \chi_q - \chi_\infty \frac{2}{\pi} \tan^{-1}(k_{sl} e_{py}) \tag{9}$$

where e_{py} is the cross-track error, χ_q is the angle between the reference line and the north, χ_∞ is a parameter in $(0, \frac{\pi}{2}]$ which is the course reference when the error is large, and k_{sl} is a tuning parameter governing the vector field smoothness. The cross-track error e_{py} is calculated with respect to the straight-line path frame. In [17] it is shown that, for first-order course dynamics as in (8), the control law which is able to let $\chi \to \chi_d$ and $e_{py} \to 0$ as $t \to \infty$ is

$$\begin{aligned}\chi_c = &-\zeta_{sl}\tilde{\chi} + \chi - \chi_\infty \frac{2}{\pi}\frac{\beta_s V_g}{\alpha_\chi} \sin(\chi - \chi_q) \\ &- \frac{\kappa_{sl}}{\alpha_\chi} \operatorname{sat}\left(\frac{\tilde{\chi}}{\varepsilon_{sl}}\right)\end{aligned} \tag{10}$$

where $\tilde{\chi} = \chi - \chi_d$, $\beta_s = k_{sl}/(1 + (k_{sl} e_{py})^2)$, $V_g = \|\mathbf{V}_g\|$, ζ_{sl} provides a stability margin, and κ_{sl}, ε_{sl} govern the aggressiveness and chattering of the control action.

2) Orbit Path Following: The desired course vector field which drives the aircraft to loiter on an orbit path is

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

where is $\tilde{d} = d - R$, d is the distance of the UAV from the orbit center, R the orbit radius and γ is the angle between the north and the UAV position with respect to the orbit center. The parameter λ is 1 for clockwise orbit path and -1 for counter-clockwise orbit path. In [17] it is shown that, for first-order course dynamics as in (8), the control law which is able to let $\chi \to \chi_d$ and $\tilde{d} \to 0$ as $t \to \infty$ is

$$\begin{aligned}\chi_c = &-\zeta_o\tilde{\chi} + \chi + \frac{V_g}{\alpha_\chi d}\sin(\chi - \gamma) \\ &+ \beta_o \frac{\lambda V_g}{\alpha_\chi}\cos(\chi - \gamma) - \frac{\kappa_o}{\alpha_\chi}\operatorname{sat}\left(\frac{\tilde{\chi}}{\varepsilon_o}\right)\end{aligned} \tag{12}$$

where $\beta_o = k_o/(1 + (k_o\tilde{d})^2)$, and the parameters k_o, κ_o, ε_o, ζ_o are defined similarly to the straight-line case.

Remark 3: Not only (10) *and* (12) *are based on the assumption that V_g is known, but the stability analysis in [17] is based on first-order course dynamics as in* (8)*. Therefore, even full wind knowledge, i.e. $V_g = ||\boldsymbol{V}_a + \boldsymbol{V}_{w,s} + \boldsymbol{V}_{w,d} + \boldsymbol{V}_{w,v}||$,* (10) *and* (12) *may perform poorly in the presence of high-order course angle dynamics as in* (7)*. Section V will verify this to be true, especially for orbit paths.*

B. Adaptive Vector Field

In view of the last remark, a relevant question is whether there exists an mechanism to automatically 'adaptat' the VF in such a way to compensate for unmodelled dynamics: in the following we will illustrate how to adapt V_g depending on the cross-track error. We will consider $\hat{V}_g$ in place of V_g, where $\hat{V}_g$ is adapted by an auxiliary differential equation.

1) Straight Line Following: The estimation dynamics for a straight-line path is

$$\dot{\hat{V}}_g = \Gamma_{sl}\mu_{sl}\tilde{\chi}\chi_\infty\beta_s\frac{2}{\pi}\sin(\chi - \chi_q) + F_{sl} - \sigma_{sl}\Gamma_{sl}\hat{V}_g \tag{13}$$

where Γ_{sl} is the estimator gain, μ_{sl} is a weighting term and σ_{sl} adds damping action. The feed-forward term F_{sl} accounts for the variation of V_g with respect to the course angle χ

$$F_{sl} = \frac{\partial \hat{V}_g}{\partial \chi}\left[-\chi_\infty\frac{2}{\pi}\beta_s\hat{V}_g\sin(\chi - \chi_q) - \kappa_{sl}\operatorname{sat}\left(\frac{\tilde{\chi}}{\varepsilon_{sl}}\right)\right] \tag{14}$$

where the partial derivative is approximated by

$$\begin{aligned}\frac{\partial \hat{V}_g}{\partial \chi} \simeq W_s \Big[&\sin(\psi_{w,s} - \chi) + \left(V_a^2 - W_s^2\sin^2(\psi_{w,s} - \chi)\right)^{-\frac{1}{2}} \\ &\cdot \sin(\psi_{w,s} - \chi)\cos(\psi_{w,s} - \chi)\Big]\end{aligned} \tag{15}$$

with $W_s = \|\mathbf{V}_{w,s}\|$ and $\psi_{w,s}$ having similar meaning as in Fig. 1, but for the steady-state wind component $\mathbf{V}_{w,s}$.

2) Orbit Following: The estimation for an orbit path is

$$\begin{aligned}\dot{\hat{V}}_g = &-\Gamma_o\mu_o\tilde{\chi}\left(\frac{1}{d}\sin(\chi - \gamma) + \lambda\beta_o\cos(\chi - \gamma)\right) \\ &+ F_o - \sigma_o\Gamma_o\hat{V}_g\end{aligned} \tag{16}$$

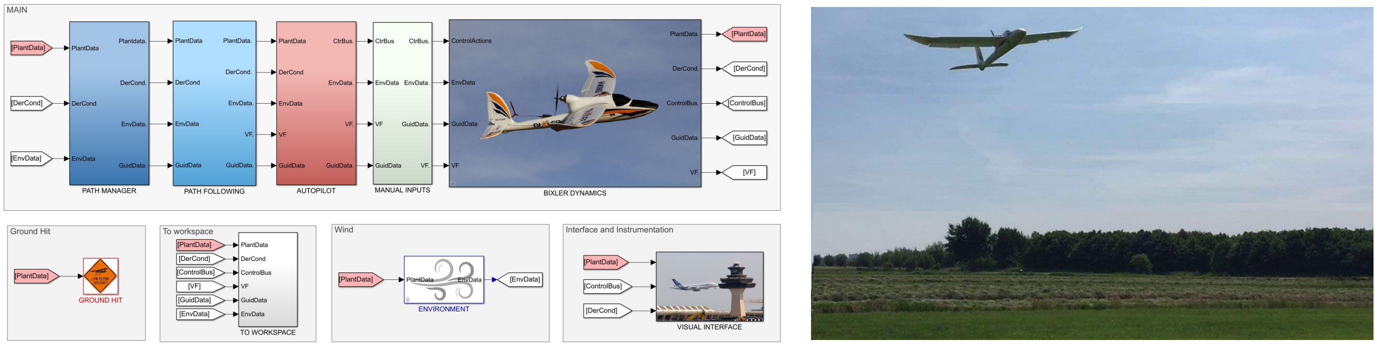
$$\begin{aligned}F_o = &\frac{\partial \hat{V}_g}{\partial \chi}\left[\frac{\hat{V}_g}{d}\sin(\chi - \gamma) + \lambda\beta_o\hat{V}_g\cos(\chi - \gamma)\right. \\ &\left. - \kappa_o\operatorname{sat}\left(\frac{\tilde{\chi}'}{\varepsilon_o}\right)\right].\end{aligned} \tag{17}$$

Stability/robustness of the proposed adaptive VF are given in the following theorem in terms of uniform ultimately boundedness (UUB) (it is well known that in the presence of disturbances and unmodelled dynamics asymptotic Lyapunov stability cannot be guaranteed, cf. [33, Sects. 8.4 and 8.5]).

Theorem 1: Consider the course angle dynamics

$$\dot{\chi} = \alpha_\chi(\chi_c - \chi) + \Delta, \tag{18}$$

where $|\Delta| \le \bar{\Delta}$ is a disturbance term with given upper bound $\bar{\Delta}$ which arises from the unmodelled dynamics of (8). Furthermore, assume that the unknown V_g is slowly time-varying with $|V_g| \le \bar{V}_g$ and $|\dot{V}_g| \le \bar{\bar{V}}_g$, for some unknown



(a) UAV simulator (in Matlab-Simulink environment)

(b) Actual Bixler during tests

Fig. 5: UAV simulator and actual Bixler during field experiments.

$\bar{V}_g$ and $\bar{\bar{V}}_g$. Then, the control laws (10), (12) (with $\hat{V}_g$ in place of V_g) and the adaptive laws (13), (16) guarantee uniform ultimate boundedness of $\left\| \left[e_{py},\ \rho^{1/2}\tilde{\chi},\ \Gamma_{sl}^{-1/2}\Theta \right] \right\|^2$, with bound B as in (20).

Proof. See Appendix.

Remark 4: The importance of Theorem 1 is to recast the path-following problem as an adaptive control with robustness against unmodelled course angle dynamics and slowly-time varying wind. Note that the assumption of bounded $|V_g|$ and $|\dot{V}_g|$ requires $\mathbf{V}_w$ and $\mathbf{V}_a$ to be bounded and with bounded derivative. As this situation is often met in practice, such assumption is commonly adopted in VF approaches [17].

Remark 5: The proposed adaptive VF requires to tune two extra pairs of parameters: the adaptive gains Γ_s and Γ_o should be high enough to speed convergence, while being aware that too high adaptive gains might lead to oscillations and eventually instability [33, Sect. 8.3]). The gains σ_s and σ_o are leakage gains [33, Sect. 8.4] that should be larger if the bounds of the unmodelled dynamics are large.

V. Algorithm Evaluation

In this section, the performance of the adaptive VF is assessed, as compared to the standard VF method of [17] and to an ideal VF method, with the following wind knowledge:

- Standard VF: (10), (12) with $V_g(t) = ||\mathbf{V}_a(t) + \mathbf{V}_{w,s}||$;
- Ideal VF: (10), (12) with $V_g(t) = ||\mathbf{V}_a(t) + \mathbf{V}_{w,s} + \mathbf{V}_{w,d}(t) + \mathbf{V}_{w,v}(t)||$;
- Adaptive VF: (10), (12) with $V_g(t) = \hat{V}_g(t)$ and (13), (16) with $\hat{V}_g(0) = ||\mathbf{V}_a(0) + \mathbf{V}_{w,s}||$.

The standard VF knows only the constant wind component, the ideal VF knows constant, dynamic and time-varying wind components, while the adaptive VF estimates all components starting from the initial knowledge of constant wind.

Scenario	Constant wind	Turbulence	S.T.V. wind
#1	$\|\|\mathbf{V}_{w,s}\|\| = 0$ m/s	No $\mathbf{V}_{w,d}$	No $\mathbf{V}_{w,v}$
#2	$\|\|\mathbf{V}_{w,s}\|\| = 4$ m/s	No $\mathbf{V}_{w,d}$	No $\mathbf{V}_{w,v}$
#3	$\|\|\mathbf{V}_{w,s}\|\| = 4$ m/s	Yes	No $\mathbf{V}_{w,v}$
#4	$\|\|\mathbf{V}_{w,s}\|\| = 4$ m/s	Yes	Yes

TABLE II: Flight environmental conditions (S.T.V. = Slowly time-varying).

Four different wind scenarios have been defined, summarized in Table II, so as to draw conclusions on the effectiveness of adaptation in different conditions. For each scenario, we simulate a straight-line and an orbit path. The constant wind direction is 240 deg. A Dryden dynamic wind model is configured following the Military specifications MIL-F-8785C (altitude of 50 m, turbulence intensity of 2.15 m/s on the $\mathbf{i}^b$ and $\mathbf{j}^b$ axes, and 1.4 m/s on the $\mathbf{k}^b$ axis, with wavelengths of 200 m). The slowly time-varying wind is taken as a sinusoid of 0.01 rad/s, perturbing both wind magnitude and direction with amplitudes 3 m/s and π rad, respectively.

Two experimental sets are performed:

- Using ideal first-order course angle dynamics;
- Using the more realistic UAV model (with high-order course angle dynamics).

The first set has the purpose of testing the algorithms in the ideal scenario. In such a way we can see how the performance degrades in non-ideal scenarios.

The performance of the standard, adaptive and ideal VF are evaluated using the RMS steady-state cross-track error, calculated in the last portion of the path when e_{py} or $\tilde{d}$ have converged. The parameters k_{sl}, κ_{sl}, ε_{sl}, k_o, κ_o, ε_o, Γ_{sl}, σ_{sl}, Γ_o, and σ_o, summarized in Table III, have been tuned so as to find a good compromise between convergence speed and no oscillations. The scaling parameters μ_{sl} and μ_o are chosen as the ratio between the initial cross-track error and the maximum course error, i.e. $\mu_{sl} = (e_{py}(0)/\pi)^2$, $\mu_o = (\tilde{d}(0)/\pi)^2$.

χ_∞	k_{sl}, k_o	κ_{sl}, κ_o	ε_{sl}, ε_o
$\pi/2$	0.1 m^{-1}	$\pi/2$ rad^2/s	1 rad

Γ_{sl}	Γ_o	σ_{sl}, σ_o	ζ_{sl}, ζ_o
0.5	0.1	0.001	0.001

TABLE III: Standard and Adaptive Vector Field parameters.

A. Simplified first-order model

For the first-order course dynamics (8), Table IV reports the RMS steady-state errors in all environmental conditions.

The following observations can be drawn from Table IV:

- With perfect knowledge of the wind and assuming simplified first-order dynamics, the ideal VF achieves zero steady-state error in all wind conditions;

Straight-line path following			
Scenario	Standard VF	Adaptive VF	Ideal VF
#1	0.00	0.00	0.00
#2	0.00	0.00	0.00
#3	0.16	0.12	0.00
#4	0.17	0.12	0.00
Orbit path following			
Scenario	Standard VF	Adaptive VF	Ideal VF
#1	0.00	0.00	0.00
#2	0.00	0.00	0.00
#3	0.29	0.14	0.00
#4	0.31	0.14	0.00

TABLE IV: Vector field RMS steady-state errors in meters (first-order course dynamics).

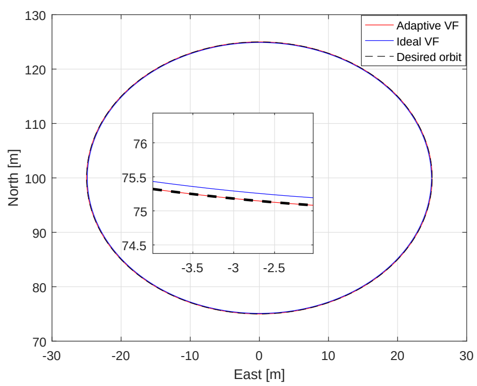


Fig. 6: Ideal vs Adaptive VF during orbit following with high-order dynamics (Scenario #1). To highlight the path following error of the ideal VF, a zoom is shown in the small picture.

- In the absence of wind, or with only constant wind (Scenarios #1 and #2) also the standard and the adaptive VF can achieve zero steady-state-error;
- The adaptive VF outperforms the standard VF in Scenarios #3 and #4 (error reduction $>20\%$ for the straight line and $>50\%$ for the orbit), when unmodelled wind components cannot be accounted by the standard VF.

B. Realistic UAV model

Using the more realistic UAV model, Table V reports the RMS steady-state errors in all environmental conditions.

Straight-line path following			
Scenario	Standard VF	Adaptive VF	Ideal VF
#1	0.00	0.00	0.00
#2	0.00	0.00	0.00
#3	0.26	0.25	0.26
#4	0.24	0.24	0.20
Orbit path following			
Scenario	Standard VF	Adaptive VF	Ideal VF
#1	0.10	0.00	0.10
#2	0.10	0.00	0.10
#3	0.39	0.21	0.31
#4	1.29	0.80	1.09

TABLE V: Vector field RMS steady-state errors in meters (high-order course dynamics).

Table V demonstrates that the ideal VF has lost its perfect performance shown in Table IV. That is, even with full knowledge of the wind, the ideal VF cannot cope with unmodelled dynamics. Other observations drawn from Table V:

- For straight line, the improvement of the adaptive VF is often small. This can be explained by the fact that the unmodelled UAV dynamics are not 'excited' by the straight-line path. The term 'excited' is used in the sense of *persistency of excitation*, a concept well known in adaptive control [33, Sect. 5.2] and referring to the number of sinusoids contained in a signal. The higher the frequency content of a signal flowing across the closed-loop system, the more the unmodelled dynamics of the system will 'manifest' and make the tracking error different than zero, which in turn will activate the adaptive law. Persistency of excitation is reflected in the path-following problem by the fact that the periodic motion induced by the orbit path contains sinusoidal components which are absent in the straight line. In fact, the poor excitation of the straight-line makes all algorithms achieve zero errors in Scenarios #1 and #2 despite the unmodelled dynamics;
- In the orbit scenario, apparently the excitation induced by periodic motion activates the adaptive law and makes the adaptive VF attain drastic improvements, outperforming not only the standard VF, but also the ideal VF. Remarkably, in Scenario #1 the adaptive VF achieves zero tracking error by completely compensating unmodelled dynamics, as shown in Fig. 6: on the other hand, the unmodelled dynamics prevent the ideal VF from perfectly following the orbit (the steady-state error is around 0.11m). Also in Scenario #2 the adaptive VF drives the error to zero by counteracting the constant wind disturbances, while in Scenarios #3 and #4 the error is reduced by 46% and 61% respectively.

C. On transient performance and a priori wind knowledge

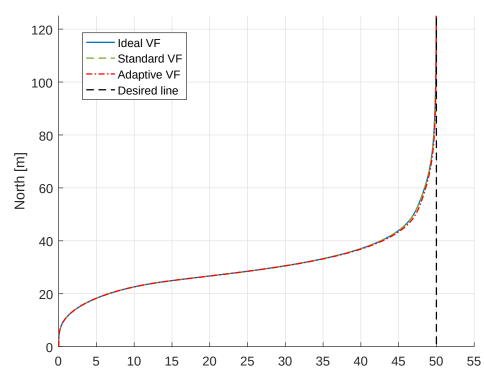


Fig. 7: Transient of Ideal, Standard and Adaptive VF during line following with high-order dynamics (Scenario #2).

To comment on transient performance of the VF algorithms, Table VI collects the errors calculated from the beginning of the trajectory till when the error is less than 1 meter (the values are higher than the previous tables, because the UAV

Straight-line path following

Scenario	Standard VF	Adaptive VF	Ideal VF
#1	23.53	23.52	23.53
#2	26.24	26.20	26.24
#3	26.63	26.58	26.62
#4	27.32	27.08	27.13

Orbit path following

Scenario	Standard VF	Adaptive VF	Ideal VF
#1	31.02	31.01	31.02
#2	32.54	32.53	32.54
#3	33.02	33.00	32.99
#4	33.86	33.85	33.73

TABLE VI: Vector field RMS transient errors in meters (high-order course dynamics).

Straight-line path following

Scenario	Standard VF	Adaptive VF
#5	0.35	0.32
#6	0.40	0.36

Orbit path following

Scenario	Standard VF	Adaptive VF
#5	0.69	0.33
#6	0.81	0.45

TABLE VII: Vector Field RMS steady-state errors in meters (UAV flight simulator experiments).

starts 50 meters away from the desired trajectory). Table VI reveals that the transient performance of the adaptive VF is close and in most cases slightly better than the non-adaptive versions (cf. Fig. 7 for straight-line following in Scenario #2). The explanation is twofold: (a) as indicated previously, the initial estimated ground speed in the adaptive VF results from the vector sum of airspeed and constant wind, which is good starting point feasible for implementation (the same a priori knowledge as the standard VF); (b) the estimator (13) and (16), being Lyapunov-based, contributes to stability by driving the error to zero. While the transient performance of any adaptive algorithm inevitably benefits from good initial knowledge of the uncertain parameters [33, Sect. 4.3.7], it is worth remarking that, thanks to the estimator (13) and (16), the a priori knowledge of standard VF is *not* requested in the adaptive VF, giving benefits in unknown wind environments.

D. UAV flight simulator experiments

Finally, some extra experiments using a UAV flight simulator developed by the System Engineering Research Institute, China State Shipbuilding Corporation, have been performed in order to further validate the methodology. Due to non-disclosure agreements, the details of the simulator cannot be reported. The experiments have been performed on a simulated Bixler UAV whose first-order time constant was identified to be $\alpha_\chi = 0.4213$. The environmental conditions for the UAV flight simulator experiments were:

- *Scenario #5*: Wind direction 220 deg, wind speed 4.0 m/s (including turbulence around such nominal values);
- *Scenario #6*: Wind direction 310 deg, wind speed 5.0 m/s (including turbulence around such nominal values).

Table VII reports the RMS steady-state errors. The experiments overall confirm the simulation results.

VI. Conclusions

The goal of this paper was to analyze and demonstrate the performance of a new robust adaptive Vector Field control law which exploits an estimator to compensate for the lack of knowledge of the wind vector and for unmodelled course dynamics (as arising from the low-level control structure of the autopilot). Extensive experiments have shown that the new Vector Field method operates better than its standard version in windy environments, and that the estimator can effectively compensate unmodelled course angle dynamics.

Further developments of the adaptive Vector Field could be extension of the approach in three dimensional paths, and addressing time-delayed measurements.

References

[1] L. Marconi, F. Basile, G. Caprari, R. Carloni, P. Chiacchio, C. Hurzeler, V. Lippiello, R. Naldi, J. Nikolic, B. Siciliano, S. Stramigioli, and E. Zwicker, "Aerial service robotics: The AIRobots perspective," in *2012 2nd International Conference on Applied Robotics for the Power Industry (CARPI)*, 2012, pp. 64–69.

[2] A. P. Aguiar, J. P. Hespanha, and P. V. Kokotović, "Performance limitations in reference tracking and path following for nonlinear systems," *Automatica*, vol. 44, no. 3, pp. 598–610, 2008.

[3] L. Furieri, T. Stastny, L. Marconi, R. Siegwart, and I. Gilitschenski, "Gone with the wind: Nonlinear guidance for small fixed-wing aircraft in arbitrarily strong windfields," in *2017 American Control Conference (ACC'17)*, 2017, pp. 4254–4261.

[4] J. Chang, J. Cieslak, J. Davila, 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.

[5] D. Invernizzi and M. Lovera, "Trajectory tracking control of thrust-vectoring UAVs," *Automatica*, vol. 95, pp. 180 – 186, 2018.

[6] P. B. 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*, vol. 34, no. 1, pp. 42–59, 2014.

[7] T. Yamasaki, H. Takano, and Y. Baba, "Robust path-following for UAV using pure pursuit guidance," in *Aerial Vehicles*. InTech, 2009.

[8] G. Ambrosino, M. Ariola, U. Ciniglio, F. Corraro, E. De Lellis, and A. Pironti, "Path generation and tracking in 3-d for uavs," *IEEE Transactions on Control Systems Technology*, vol. 17, no. 4, pp. 980–988, 2009.

[9] 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.

[10] M. Kothari, I. Postlethwaite, and D.-W. Gu, "UAV path following in windy urban environments," *Journal of Intelligent & Robotic Systems*, vol. 74, no. 3-4, pp. 1013–1028, 2014.

[11] N. Cho and Y. Kim, "Three-dimensional nonlinear differential geometric path-following guidance law," *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 12, pp. 948–954, 2015.

[12] H. Chao, Y. Cao, and Y. Chen, "Autopilots for small unmanned aerial vehicles: a survey," *International Journal of Control, Automation and Systems*, vol. 8, no. 1, pp. 36–44, 2010.

[13] F. Gavilan, R. Vazquez, and S. Esteban, "Trajectory tracking for fixed-wing UAV using model predictive control and adaptive backstepping," *1st IFAC Workshop on Advanced Control and Navigation for Autonomous Aerospace Vehicles (ACNAAV15)*, pp. 132–137, 2015.

[14] J. Chang, J. Cieslak, J. Davila, A. Zolghadri, and J. Zhou, "Analysis and design of second-order sliding-mode algorithms for quadrotor roll and pitch estimation," *ISA Transactions*, vol. 71, pp. 495–512, 2017.

[15] G. Casadei, L. Furieri, N. Mimmo, R. Naldi, and L. Marconi, "Internal model-based control for loitering maneuvers of UAVs," in *2016 European Control Conference (ECC)*, 2016, pp. 672–677.

[16] D. Invernizzi, P. Panizza, F. Riccardi, S. Formentin, and M. Lovera, "Data-driven attitude control law of a variable-pitch quadrotor: a comparison study," *20th IFAC Symposium on Automatic Control in Aerospace (ACA 2016)*, vol. 49, no. 17, pp. 236 – 241, 2016.

[17] 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.

[18] 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.
[19] D. V. Dimarogonas, "Sufficient conditions for decentralized potential functions based controllers using canonical vector fields," *IEEE Transactions on Automatic Control*, vol. 57, no. 10, pp. 2621–2626, 2012.
[20] H. Chen, K. Chang, and C. S. Agate, "UAV path planning with tangent-plus-Lyapunov vector field guidance and obstacle avoidance," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 2, pp. 840–856, 2013.
[21] W. Jiang, D. Wang, Y. Wang, and Z. A. Ali, "UAV rendezvous based on time-varying vector fields," *Electronics Letters*, vol. 53, no. 10, pp. 653–655, 2017.
[22] J. L. G. Olavo, G. D. Thums, T. A. Jesus, L. C. de Arajo Pimenta, L. A. B. Torres, and R. M. Palhares, "Robust guidance strategy for target circulation by controlled UAV," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 3, pp. 1415–1431, 2018.
[23] 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.
[24] S. Lee, J. Jung, S. Kim, I. Kim, and H. Myung, "DV-SLAM (dual-sensor-based vector-field SLAM) and observability analysis," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 2, pp. 1101–1112, 2015.
[25] 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'17)*, 2017, pp. 1127–1132.
[26] "Roll, pitch and yaw controller tuning," 2016. [Online]. Available: http://ardupilot.org/plane/docs/roll-pitch-controller-tuning.html
[27] R. F. Stengel, *Flight dynamics*. Princeton University Press, 2015.
[28] R. W. Beard and T. W. McLain, *Small unmanned aircraft: Theory and practice*. Princeton University Press, 2012.
[29] J. W. Langelaan, N. Alley, and J. Neidhoefer, "Wind field estimation for small unmanned aerial vehicles," *Journal of Guidance Control and Dynamics*, vol. 34, no. 4, p. 1016, 2011.
[30] S. Farì, "Guidance and control for a fixed-wing UAV," Master's thesis, Politecnico di Milano, 2017.
[31] ArduPilot. Roll, Pitch and Yaw Controller Tuning. [Online]. Available: http://ardupilot.org/plane/docs/roll-pitch-controller-tuning.html#roll-controller
[32] A. Ul Haque, W. Asrar, A. A. Omar, E. Sulaeman, and J. Ali, "Comparison of digital DATCOM and wind tunnel data of a winged hybrid airships generic model," in *Applied Mechanics and Materials*, vol. 629, 2014, pp. 36–41.
[33] P. A. Ioannou and J. Sun, *Robust adaptive control*. PTR Prentice-Hall Upper Saddle River, NJ, 1996.
[34] S. Roy, S. B. Roy, and I. N. Kar, "Adaptive-robust control of Euler-Lagrange systems with linearly parametrizable uncertainty bound," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 5, pp. 1842–1850, 2018.

APPENDIX

Stability analysis for adaptive vector field

For lack of space, the stability analysis is provided for the straight line case only. The stability analysis for the orbit path follows similar steps and is left to the reader.

The adaptive law of $\hat{V}_g$ is derived based on the Lyapunov argument below. Let $\Theta = \hat{V}_g - V_g$ be the estimation error. Consider the Lyapunov function $\mathcal{V} = \mathcal{V}_1 + \rho\mathcal{V}_2 + \frac{1}{2}\Gamma_{sl}^{-1}\Theta^2$, with $\mathcal{V}_1 = \frac{1}{2}e_{py}^2$, $\mathcal{V}_2 = \frac{1}{2}\tilde{\chi}^2$ whose derivative is

$$\dot{\mathcal{V}} = \dot{\mathcal{V}}_1 + \rho\dot{\mathcal{V}}_2 + \Gamma_{sl}^{-1}\Theta\dot{\Theta}$$

where ρ is the positive user-designed weight. Substitute (10) (with $\hat{V}_g$ in place of V_g) into the derivative of $\mathcal{V}$

$$\begin{aligned}\dot{\mathcal{V}} = \dot{\mathcal{V}}_1 + \rho\tilde{\chi}[\chi_\infty \frac{2\beta_s}{\pi}(\hat{V}_g - V_g)(\sin(\chi - \chi_q) \\ - \kappa_{sl} sat(\frac{\tilde{\chi}}{\epsilon_{sl}})] + \Gamma^{-1}(\hat{V}_g - V_g)(\dot{\hat{V}}_g - \dot{V}_g).\end{aligned}$$

Now substitute (13) (omitting F_{sl} for simplicity)

$$\begin{aligned}\dot{\mathcal{V}} &= \dot{\mathcal{V}}_1 - \rho\zeta_{sl}\tilde{\chi}^2 - \rho\kappa_{sl}\tilde{\chi} sat(\frac{\tilde{\chi}}{\epsilon_{sl}}) + \{(\dot{\hat{V}}_g - \dot{V}_g)\Gamma_{sl}^{-1} \\ &\quad + \rho\tilde{\chi}\chi_\infty \frac{2\beta_s}{\pi}\sin(\chi - \chi_q)\}(\hat{V}_g - V_g) + \rho\tilde{\chi}\Delta \\ &\le \dot{\mathcal{V}}_1 - \rho\zeta_{sl}\tilde{\chi}^2 - \rho\kappa_{sl}\tilde{\chi} sat(\frac{\tilde{\chi}}{\epsilon_{sl}}) - \sigma_{sl}\Theta^2 \\ &\quad + \rho\,|\tilde{\chi}|\,\bar{\Delta} - \sigma_{sl}\Theta(-\Gamma_{sl}^{-1}\dot{V}_g\sigma_{sl}^{-1} - V_g)\end{aligned}$$

Using the design condition $\kappa_{sl} \ge \bar{\Delta}$ we can remove the third and fifth terms of the last inequality when $\frac{\tilde{\chi}}{\epsilon_{sl}} \ge 1$ (the analysis for $\frac{\tilde{\chi}}{\epsilon_{sl}} < 1$ straightforwardly leads to boundedness, cf. [34]). Furthermore, after applying the inequality $-a^2 + ab \le -\frac{a^2}{2} + \frac{b^2}{2}$ (valid for any a and b) to the fourth and last term above, we write

$$\dot{\mathcal{V}} \le \dot{\mathcal{V}}_1 - \rho\zeta_{sl}\tilde{\chi}^2 - \frac{\sigma_{sl}}{2}\Theta^2 + \frac{\sigma_{sl}(V_g + \dot{V}_g\Gamma_{sl}^{-1}\sigma_{sl}^{-1})^2}{2}. \quad (19)$$

Since we assume that the wind changes in a slowly time-varying fashion, the magnitude of $\dot{V}_g$ will be bounded. Also, the ground velocity V_g is bounded. Therefore we can bound $\left|V_g + \dot{V}_g\Gamma_{sl}^{-1}\sigma_{sl}^{-1}\right| \le \Delta_V$, for some unknown Δ_V. Then, using the definition of the Lyapunov function $\mathcal{V}$ and the analysis of $\dot{\mathcal{V}}_1$ carried out in [17], we have that (19) implies

$$\dot{\mathcal{V}} \le -\kappa\mathcal{V} - (\bar{\kappa} - \kappa)\mathcal{V} + C$$

where $\zeta_{sl} \ge \frac{1}{2}\sigma_{sl}\Gamma_{sl}$, $0 < \kappa < \bar{\kappa}$, $\bar{\kappa} = \sigma_{sl}\Gamma_{sl}$, and $C = \frac{\sigma_{sl}\bar{\Delta}_V^2}{2}$, i.e. C is proportional to $\bar{\Delta}_V^2$ (upper bound on the combined variation of the wind). After defining the scalar $B = C/(\bar{\kappa} - \kappa)$ we straightforwardly obtain the bound

$$\left\|\left[e_{py},\ \rho^{1/2}\tilde{\chi},\ \Gamma_{sl}^{-1/2}\Theta\right]\right\|^2 \le \max\{\mathcal{V}(0), B\} \quad (20)$$

i.e. e_{py}, $\tilde{\chi}$, Θ converge inside a compact set and stay bounded.