

Geometric 3D Path-Following Control for a Fixed-Wing UAV on $SO(3)^*$

Venanzio Cichella[†]

Naval Postgraduate School, Monterey, CA 93943

Enric Xargay[‡]

University of Illinois at Urbana-Champaign, Urbana, IL 61801

Vladimir Dobrokhodov[§], Isaac Kaminer[¶]

Naval Postgraduate School, Monterey, CA 93943

António M. Pascoal^{||}

Instituto Superior Técnico, Lisbon, 1049 Portugal

Naira Hovakimyan^{**}

University of Illinois at Urbana-Champaign, Urbana, IL 61801

This paper addresses the problem of steering an Unmanned Aerial Vehicle along a given path. In the setup adopted, the vehicle is assigned a nominal path and a speed profile along it. The vehicle is then tasked to follow this nominal path independently of the temporal assignments of the mission, which is in contrast to “open-loop” trajectory tracking maneuvers. The paper builds on previous work by the authors on path-following control and derives a new control algorithm that uses the Special Orthogonal group $SO(3)$ in the formulation of the attitude control problem. This formulation avoids the geometric singularities and complexities that appear when dealing with local parameterizations of the vehicle’s attitude, and leads thus to a *singularity-free* path-following control law. Flight test results performed in Camp Roberts, CA, demonstrate the efficacy of the path-following control algorithm developed in this paper.

I. Introduction

Numerous problems related to motion control of autonomous vehicles (including air, land, and marine robots) have been studied in recent years [1]. Comprehensive overviews on motion control can be found in [2,3]. The problems addressed in the literature can be classified in three groups: *point stabilization* –the goal is to stabilize the vehicle at a given target point with a desired orientation; *trajectory tracking* –the vehicle is required to track a time parameterized reference; and *path following* –the vehicle is required to converge to and follow a path, without any temporal specifications. Path-Following (PF) control has received relatively less attention than point stabilization and trajectory tracking. Pioneering work can be found in [4] and [5] where a solution for wheeled robots is presented. A solution to the PF problem for marine vehicles has been reported in [6] and [7]. For a PF solution for Unmanned Aerial Vehicles (UAVs) the reader is referred to [8]. The underlying assumption in PF control is that the PF algorithm acts only on the attitude control effectors of the vehicle so as to steer it along the path, independently of the temporal assignments

*Research supported in part by projects USSOCOM, ONR under Contract N00014-11-WX20047, ONR under Contract N00014-05-1-0828, AFOSR under Contract No. FA9550-05-1-0157, ARO under Contract No. W911NF-06-1-0330, and CO3AUVs of the EU (Grant agreement n. 231378)

[†]Research Associate, Dept. of Mechanical & Astronautical Engineering; venanzio.cichella@gmail.com.

[‡]Doctoral Student, Dept. of Aerospace Engineering; xargay@illinois.edu. Student Member AIAA.

[§]Research Associate Professor, Dept. of Mechanical & Astronautical Engineering; vldobr@nps.edu. Senior Member AIAA.

[¶]Professor, Dept. of Mechanical & Astronautical Engineering; kaminer@nps.edu. Member AIAA.

^{||}Associate Professor, Laboratory of Robotics and Systems in Engineering and Science (LARSyS), IST, Technical University of Lisbon, Portugal; antonio@isr.ist.utl.pt. Member AIAA.

^{**}Professor, Dept. of Mechanical Science and Engineering; nhovakim@illinois.edu. Associate Fellow AIAA.

of the mission. The approach is thus in contrast to the trajectory tracking, for which it is proven in [9] that there exist fundamental performance limitations that cannot be overcome by any controller structure. The limitations of trajectory tracking are also discussed in detail in [10], where it is shown that, perfect tracking of any reference signal is possible in the presence of non-minimum phase zeros, whereas this is not the case with unstable zero dynamics. Moreover, smoother convergence to the path is typically achieved with PF control laws when compared to the behavior obtained with trajectory tracking algorithms, and the control signals are less likely to be saturated.

This paper presents a new solution to the problem of 3D path following. The main novelty of this work consists in the use of the Special Orthogonal group $SO(3)$ in the formulation of the attitude control problem. By doing so, the class of vehicles for which the design procedure is applicable is quite general and includes any vehicle that can be modeled as a rigid-body subject to controlled linear and angular velocities. Furthermore, contrary to most of the approaches described in the literature [1–3], the controller proposed does not suffer from geometric singularities that appear due to the local parametrization of the vehicle's rotation matrix.

The developed methodology for PF control unfolds in two basic steps. First, for a given mission involving a typical fixed-wing UAV, a feasible spatial path $p_d(\ell)$ and a feasible speed profile $v_d(\ell)$, both conveniently parameterized by a virtual arc length $\ell \in [0, \ell_f]$, are generated satisfying the mission requirements. This step relies on optimization methods that take explicitly into account initial and final boundary conditions, a performance criterion to be optimized, and simplified vehicle dynamics. The second step consists of steering the vehicle along its assigned path $p_d(\ell)$, which is done independently of the temporal assignments of the mission. This step relies on a nonlinear PF control algorithm yielding robust performance of a UAV executing various aggressive missions. The solution to the PF problem leaves the speed profile of the vehicle as an extra degree of freedom that can be exploited in future developments.

Another key feature of the framework presented in this paper is that it leads to a *multiloop control structure* in which an inner-loop controller stabilizes the vehicle dynamics and provides reference tracking capabilities, while a guidance outer-loop controller is designed to control the vehicle kinematic, providing PF capabilities. In particular, this approach allows to take explicit advantage of the fact that normally UAVs are equipped with commercial autopilots (AP). For the purpose of this paper, we assume that the UAV is equipped with an AP that stabilizes the aircraft dynamics and provides angular-rate as well as speed tracking capabilities. Therefore, we address only the design of a new outer-loop algorithm for aggressive PF control for a single UAV, which replaces the standard waypoint-based guidance systems of traditional AP. It is thus assumed that the AP inner-loop is given and it guarantees reference following capability.

In particular, in this paper, Lyapunov direct method is used to proof the stability and convergence properties of the developed PF algorithm. For the problem of path generation, we avail ourselves of previous work on the development of algorithms that are suitable for real-time computation of feasible paths and speed profiles for autonomous vehicles. The reader is referred to [11] for a detailed formulation of the path-generation problem.

This paper is organized as follows. In Section II a formulation of the PF problem is derived at a kinematic level. The kinematic equations of the path-following error dynamics are defined and the system is characterized by defining appropriate PF error variables and input signals. Section III presents a solution to the PF control problem in a 3D space. A formal statement of the stability and convergence properties of the PF control algorithm is provided in this Section. Section IV provides an overview of simulation and flight test results demonstrating the benefits and efficacy of the PF algorithm developed. We first provide a brief description of the Hardware-in-the-loop (HIL) simulator architecture and simulation results. Next, we present flight test results conducted in Camp Roberts, CA. Section V contains the main conclusions and provides directions for future work.

II. Path-Following for a Fixed-Wing UAV: Problem Formulation

Pioneering work in the area of path following can be found in [4], where an elegant solution to the PF problem of a wheeled robot was presented at the kinematic level. In the setup adopted, the kinematic model of the vehicle was derived with respect to a Frenet-Serret frame moving along the path, playing the role of a *virtual target vehicle* to be tracked by the real vehicle. The origin of the Frenet-Serret frame was placed at the point on the path *closest* to the real vehicle. This initial work spurred a great deal of activity in the literature addressing the PF problem. Of particular interest is the work reported in [12], in which the authors reformulated the setup used in [4] and derived a feedback control law that steers a wheeled robot

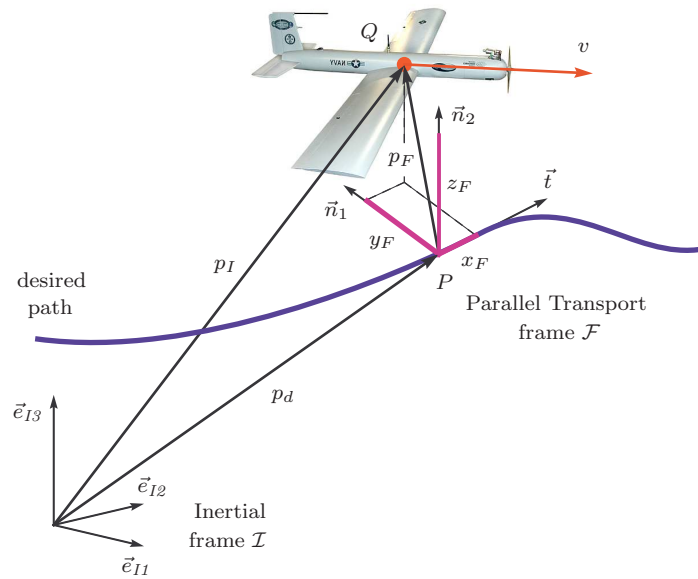


Figure 1: Following a virtual target vehicle. Problem geometry.

along a desired path and overcomes stringent initial condition constraints adopted in [4]. The key to the algorithm in [12] was the addition of another degree of freedom to the rate of progression of the virtual target, that is in contrast with the strategy for placement of the origin of the Frenet-Serret frame adopted in [4]. The algorithm presented in [12] was extended to the 3D case in [8]. The algorithm developed in [8] relies on the insight that a UAV can follow a given path using only its attitude, thus leaving its speed as an extra degree of freedom to be used in future developments. The key idea of the algorithm is to use the vehicle's attitude control effectors to follow a virtual target running along the path.

The solution to the path-following problem described in this paper uses the same approach presented in [8], but uses the Special Orthogonal group $SO(3)$ to describe the path-following attitude error dynamics, rather than a local parametrization in terms of Euler angles. This new formulation leads to a *singularity-free* controller, a fundamental property that will be discussed and demonstrated later in this paper.

Similar to [8,12], we introduce a reference frame attached to this virtual target and define a generalized error vector between this virtual vehicle and a velocity frame attached to the actual vehicle. With this setup, the PF control problem is reduced to driving this generalized error vector to zero by using only UAV's angular rates, while following any feasible speed profile. Next, we characterize the dynamics of the kinematic errors between the vehicle and its virtual target.

Figure 1 captures the geometry of the problem at hand. Let \mathcal{I} denote an inertial reference frame $\{\vec{e}_{I1}, \vec{e}_{I2}, \vec{e}_{I3}\}$, and let $p_I(t)$ be the position of the center of mass Q of the UAV in this inertial frame. Further, let P be a point on the desired path that plays the role of the center of mass of the virtual target, and let $p_d(\ell)$ denote its position in the inertial frame. Here ℓ is a parameterizing variable that defines the position of the virtual target on the path, and its rate of progression along the path may be conveniently selected. Endowing the point P with an extra degree of freedom is the key to the PF algorithm in [12] and its extension to the 3D case described in [8].

Define also a *Parallel Transport Frame* \mathcal{F} attached to the point P on the path and characterized by the orthonormal vectors $\{\vec{t}(\ell), \vec{n}_1(\ell), \vec{n}_2(\ell)\}$, which satisfy the following frame equations [13,14]:

$$\begin{bmatrix} \frac{d\vec{t}}{d\ell}(\ell) \\ \frac{d\vec{n}_1}{d\ell}(\ell) \\ \frac{d\vec{n}_2}{d\ell}(\ell) \end{bmatrix} = \begin{bmatrix} 0 & k_1(\ell) & k_2(\ell) \\ -k_1(\ell) & 0 & 0 \\ -k_2(\ell) & 0 & 0 \end{bmatrix} \begin{bmatrix} \vec{t}(\ell) \\ \vec{n}_1(\ell) \\ \vec{n}_2(\ell) \end{bmatrix},$$

where the parameters $k_1(\ell)$ and $k_2(\ell)$ are related to the polar coordinates of curvature $\kappa(\ell)$ and torsion $\tau(\ell)$

as

$$\kappa(\ell) = (k_1^2(\ell) + k_2^2(\ell))^{\frac{1}{2}},$$

$$\tau(\ell) = -\frac{d}{d\ell} \left(\tan^{-1} \left(\frac{k_2(\ell)}{k_1(\ell)} \right) \right).$$

The dynamics of \mathcal{F} can be characterized as follows:

$$\begin{aligned} \frac{d\vec{t}}{dt} &= (k_1(\ell) \vec{n}_1 + k_2(\ell) \vec{n}_2) \dot{\ell}, \\ \frac{d\vec{n}_1}{dt} &= -k_1(\ell) \vec{t} \dot{\ell}, \\ \frac{d\vec{n}_2}{dt} &= -k_2(\ell) \vec{t} \dot{\ell}. \end{aligned} \quad (1)$$

The choice of a parallel transport frame, unlike a *Frenet-Serret frame*, ensures that this moving frame is well defined when the path has a vanishing second derivative; this issue is discussed in detail in Appendix A. The vectors $\{\vec{t}, \vec{n}_1, \vec{n}_2\}$ define an orthonormal basis of \mathcal{F} , in which the unit vector $\vec{t}(\ell)$ defines the tangent direction to the path at the point determined by ℓ , while $\vec{n}_1(\ell)$ and $\vec{n}_2(\ell)$ define the normal plane perpendicular to $\vec{t}(\ell)$. This orthonormal basis can be used to construct the rotation matrix $R_F^I(\ell) = [\{\vec{t}\}_I; \{\vec{n}_1\}_I; \{\vec{n}_2\}_I]$ from \mathcal{F} to \mathcal{I} . Furthermore, the angular velocity of \mathcal{F} with respect to \mathcal{I} , resolved in \mathcal{F} , can be easily expressed in terms of the parameters $k_1(\ell)$ and $k_2(\ell)$ as

$$\{\omega_{F/I}\}_F = \begin{bmatrix} 0, & -k_2(\ell) \dot{\ell}, & k_1(\ell) \dot{\ell} \end{bmatrix}^\top. \quad (2)$$

The angular velocity expressed in (2) can be derived from (1). Also, let $p_F(t)$ be the position of the vehicle center of mass Q in \mathcal{F} , and let $x_F(t)$, $y_F(t)$, and $z_F(t)$ be the components of the error vector $p_F(t)$ resolved in \mathcal{F} , that is

$$\{p_F\}_F = \begin{bmatrix} x_F, & y_F, & z_F \end{bmatrix}^\top.$$

Finally, let \mathcal{W} denote a vehicle-carried *velocity frame* $\{\vec{w}_1, \vec{w}_2, \vec{w}_3\}$ with its origin at the UAV center of mass and its x -axis aligned with the velocity vector of the UAV. In this paper, $q(t)$ and $r(t)$ are the y -axis and z -axis components, respectively, of the vehicle's rotational velocity resolved in the \mathcal{W} frame. With a slight abuse of notation, $q(t)$ and $r(t)$ will be referred to as *pitch rate* and *yaw rate*, respectively, in the \mathcal{W} frame.

With the above notations, the kinematic error dynamics of the UAV with respect to the virtual target is characterized and the *position-error* dynamics is derived. Since

$$p_I = p_d(\ell) + p_F,$$

then

$$\dot{p}_I]_I = \dot{\ell} \vec{t} + \omega_{F/I} \times p_F + \dot{p}_F]_F$$

where $\cdot]_I$ and $\cdot]_F$ are used to indicate that the derivatives are taken in \mathcal{I} and \mathcal{F} , respectively. Since we also have that

$$\dot{p}_I]_I = v \vec{w}_1,$$

where $v(t)$ denotes the magnitude of the UAV's velocity vector, the PF kinematic *position-error* dynamics of the UAV with respect to the virtual target can be written as

$$\dot{p}_F]_F = -\dot{\ell} \vec{t} - \omega_{F/I} \times p_F + v \vec{w}_1. \quad (3)$$

Resolved in \mathcal{F} , the above equation takes the following form:

$$\begin{bmatrix} \dot{x}_F \\ \dot{y}_F \\ \dot{z}_F \end{bmatrix} = -\begin{bmatrix} \dot{\ell} \\ 0 \\ 0 \end{bmatrix} - \left(\begin{bmatrix} 0 \\ -k_2(\ell) \dot{\ell} \\ k_1(\ell) \dot{\ell} \end{bmatrix} \times \begin{bmatrix} x_F \\ y_F \\ z_F \end{bmatrix} \right) + R_W^F \begin{bmatrix} v \\ 0 \\ 0 \end{bmatrix}.$$

To derive the *attitude*-error dynamics of the UAV with respect to its virtual target, we first introduce the auxiliary frame $\mathcal{D} \{\vec{b}_{1D}, \vec{b}_{2D}, \vec{b}_{3D}\}$, which will define the desired direction of the UAV velocity vector and will be used to shape the approach attitude to the path as a function of the “cross-track” error. The frame \mathcal{D} has its origin at the UAV center of mass and the vectors $\vec{b}_{1D}(t)$, $\vec{b}_{2D}(t)$, and $\vec{b}_{3D}(t)$ are defined as

$$\begin{aligned}\vec{b}_{1D} &\triangleq \frac{d\vec{t} - y_F \vec{n}_1 - z_F \vec{n}_2}{(d^2 + y_F^2 + z_F^2)^{\frac{1}{2}}}, \\ \vec{b}_{2D} &\triangleq \frac{y_F \vec{t} + d \vec{n}_1}{(d^2 + y_F^2)^{\frac{1}{2}}}, \\ \vec{b}_{3D} &\triangleq \vec{b}_{1D} \times \vec{b}_{2D},\end{aligned}\tag{4}$$

with d being a (positive) constant characteristic distance. Clearly, as shown in Figure 2^a, when the vehicle is far from the desired path, the vector $\vec{b}_{1D}(t)$ becomes quasi-perpendicular to $\vec{t}(\ell)$ (Step A). As the vehicle comes closer to the path and the cross-track error becomes smaller, the orientation of $\vec{b}_{1D}(t)$ tends to $\vec{t}(\ell)$ (Step B). Finally, when the position error becomes zero, \vec{b}_{1D} coincides with $\vec{t}(\ell)$ (Step C). The unit vector $\vec{b}_{1D}(t)$ defines thus the desired direction of the UAV’s velocity vector, therefore shaping the approach attitude to the path. We also note that the orthonormal basis $\{\vec{b}_{1D}, \vec{b}_{2D}, \vec{b}_{3D}\}$ can be used to construct the rotation matrix $R_D^I = [\{\vec{b}_{1D}\}_I; \{\vec{b}_{2D}\}_I; \{\vec{b}_{3D}\}_I]$ from \mathcal{D} to \mathcal{I} . Therefore, the rotation matrix $R_D^F(t) \in \text{SO}(3)$ is given by

$$\begin{aligned}R_D^F &= (R_F^I)^\top R_D^I = [\{\vec{t}\}_I; \{\vec{n}_1\}_I; \{\vec{n}_2\}_I]^\top [\{\vec{b}_{1D}\}_I; \{\vec{b}_{2D}\}_I; \{\vec{b}_{3D}\}_I] \\ &= \begin{bmatrix} \frac{d}{(d^2 + y_F^2 + z_F^2)^{\frac{1}{2}}} & \frac{y_F}{(d^2 + y_F^2)^{\frac{1}{2}}} & \frac{z_F d}{(d^2 + y_F^2 + z_F^2)^{\frac{1}{2}} (d^2 + y_F^2)^{\frac{1}{2}}} \\ \frac{-y_F}{(d^2 + y_F^2 + z_F^2)^{\frac{1}{2}}} & \frac{d}{(d^2 + y_F^2)^{\frac{1}{2}}} & \frac{-y_F z_F}{(d^2 + y_F^2 + z_F^2)^{\frac{1}{2}} (d^2 + y_F^2)^{\frac{1}{2}}} \\ \frac{-z_F}{(d^2 + y_F^2 + z_F^2)^{\frac{1}{2}}} & 0 & \frac{(d^2 + y_F^2)^{\frac{1}{2}}}{(d^2 + y_F^2 + z_F^2)^{\frac{1}{2}}} \end{bmatrix}.\end{aligned}$$

Next, let $\tilde{R}(t) \in \text{SO}(3)$ be the rotation matrix from \mathcal{W} to \mathcal{D} , that is

$$\tilde{R} \triangleq R_W^D = R_F^D R_W^F = (R_D^F)^\top R_W^F.$$

and define the real-valued error function on $\text{SO}(3)$:

$$\Psi(\tilde{R}) = \frac{1}{2} \text{tr} \left[(\mathbb{I}_3 - \Pi_R^\top \Pi_R) (\mathbb{I}_3 - \tilde{R}) \right],\tag{5}$$

where Π_R is defined as:

$$\Pi_R \triangleq \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Note that the term $(\mathbb{I}_3 - \Pi_R^\top \Pi_R)$ is the following “selector” term:

$$(\mathbb{I}_3 - \Pi_R^\top \Pi_R) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},\tag{6}$$

which is used here to extract the first column from the matrix $(\mathbb{I}_3 - \tilde{R})$. Selecting the first column (only x -axis direction) is all that is necessary to make the vehicle converge and follow the desired path^b. Then, the function $\Psi(\tilde{R})$ in (5) can be expressed in terms of the entries of $\tilde{R}(t)$ as:

$$\Psi(\tilde{R}) \triangleq \frac{1}{2} (1 - \tilde{R}_{11}),$$

^aWe notice that, for the sake of clarity, Figure 2 illustrates a 2D case in which the path-following position error z_F is assumed to be always equal to zero.

^bNote that the x -axis direction of the UAV is strictly dependent from the angular velocities $q(t)$ and $r(t)$ of \mathcal{W} , which are the angular velocities, respectively, about the y - and z -axis.

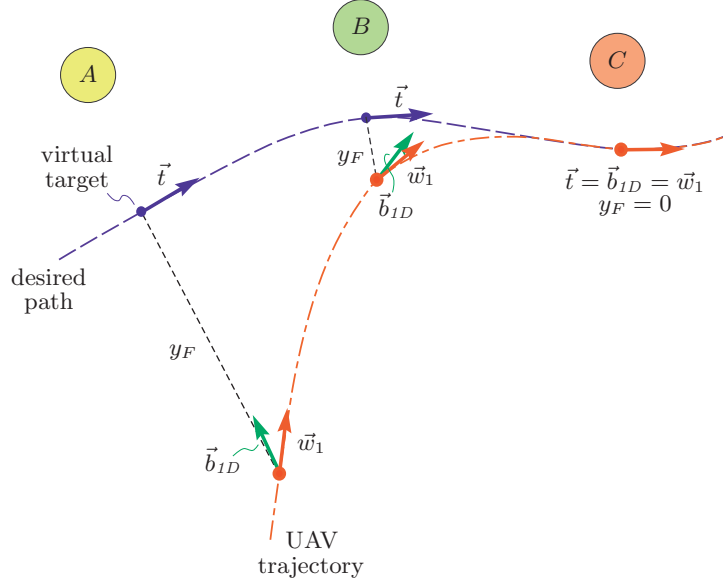


Figure 2: The auxiliary frame \mathcal{D} is used to shape the approach attitude to the path as a function of the “cross-track” error.

where $\tilde{R}_{11}(t)$ denotes the $(1, 1)$ entry of $\tilde{R}(t)$. Therefore, $\Psi(\tilde{R})$ is positive-definite about $\tilde{R}_{11} = 1$. Note that $\tilde{R}_{11} = 1$ corresponds to the situation when the velocity vector of the UAV is aligned with the unit vector $\tilde{b}_{1D}(t)$, which defines the desired attitude.

The attitude kinematics equation

$$\dot{\tilde{R}} = \dot{R}_W^D = R_W^D (\{\omega_{W/D}\}_W)^\wedge = \tilde{R} (\{\omega_{W/D}\}_W)^\wedge,$$

where $(\cdot)^\wedge : \mathbb{R}^3 \rightarrow so(3)$ denotes the *hat map* (see Appendix B), can be used to derive the time derivative of $\Psi(\tilde{R})$, which is given by:

$$\begin{aligned} \dot{\Psi}(\tilde{R}) &= -\frac{1}{2} \text{tr} \left[(\mathbb{I}_3 - \Pi_R^\top \Pi_R) \dot{\tilde{R}} \right] \\ &= -\frac{1}{2} \text{tr} \left[(\mathbb{I}_3 - \Pi_R^\top \Pi_R) \tilde{R} (\{\omega_{W/D}\}_W)^\wedge \right]. \end{aligned}$$

Property (17) of the hat map (see Appendix B) leads to

$$\dot{\Psi}(\tilde{R}) = \frac{1}{2} \left(\left((\mathbb{I}_3 - \Pi_R^\top \Pi_R) \tilde{R} - \tilde{R}^\top (\mathbb{I}_3 - \Pi_R^\top \Pi_R) \right)^\vee \right)^\top \{\omega_{W/D}\}_W,$$

where $(\cdot)^\vee : so(3) \rightarrow \mathbb{R}^3$ denotes the *vee map*, which is defined as the inverse of the hat map. It's easy to show that the first component of $\left((\mathbb{I}_3 - \Pi_R^\top \Pi_R) \tilde{R} - \tilde{R}^\top (\mathbb{I}_3 - \Pi_R^\top \Pi_R) \right)^\vee$ is equal to zero. Therefore, we can also write

$$\dot{\Psi}(\tilde{R}) = \frac{1}{2} \left(\left((\mathbb{I}_3 - \Pi_R^\top \Pi_R) \tilde{R} - \tilde{R}^\top (\mathbb{I}_3 - \Pi_R^\top \Pi_R) \right)^\vee \right)^\top \Pi_R^\top \Pi_R \{\omega_{W/D}\}_W,$$

or equivalently

$$\dot{\Psi}(\tilde{R}) = \left(\frac{1}{2} \Pi_R \left((\mathbb{I}_3 - \Pi_R^\top \Pi_R) \tilde{R} - \tilde{R}^\top (\mathbb{I}_3 - \Pi_R^\top \Pi_R) \right)^\vee \right)^\top \Pi_R \{\omega_{W/D}\}_W. \quad (7)$$

Next, we define the attitude error $e_{\tilde{R}}(t)$ as:

$$e_{\tilde{R}} \triangleq \frac{1}{2} \Pi_R \left((\mathbb{I}_3 - \Pi_R^\top \Pi_R) \tilde{R} - \tilde{R}^\top (\mathbb{I}_3 - \Pi_R^\top \Pi_R) \right)^\vee,$$

that is equal to

$$e_{\tilde{R}} = \frac{1}{2} \begin{bmatrix} \tilde{R}_{13}, & -\tilde{R}_{12} \end{bmatrix}^\top,$$

which allows to rewrite (7), using *dot product* notation, in a more compact form:

$$\dot{\Psi}(\tilde{R}) = e_{\tilde{R}} \cdot (\Pi_R \{\omega_{W/D}\}_W).$$

It is worth noting that, if $\|e_{\tilde{R}}\| = 0$, then $\tilde{R}_{11} = 1$. Finally, noting that $\{\omega_{W/F}\}_W$ can be expressed as

$$\begin{aligned} \{\omega_{W/D}\}_W &= \{\omega_{W/I}\}_W + \{\omega_{I/F}\}_W + \{\omega_{F/D}\}_W \\ &= \{\omega_{W/I}\}_W - \{\omega_{F/I}\}_W - \{\omega_{D/F}\}_W \\ &= \{\omega_{W/I}\}_W - R_F^W \{\omega_{F/I}\}_F - R_D^W \{\omega_{D/F}\}_D \\ &= \{\omega_{W/I}\}_W - R_D^W (R_F^D \{\omega_{F/I}\}_F + \{\omega_{D/F}\}_D) \\ &= \{\omega_{W/I}\}_W - \tilde{R}^\top (R_F^D \{\omega_{F/I}\}_F + \{\omega_{D/F}\}_D), \end{aligned}$$

one can write

$$\dot{\Psi}(\tilde{R}) = e_{\tilde{R}} \cdot \left(\Pi_R \left(\{\omega_{W/I}\}_W - \tilde{R}^\top (R_F^D \{\omega_{F/I}\}_F + \{\omega_{D/F}\}_D) \right) \right),$$

or equivalently

$$\dot{\Psi}(\tilde{R}) = e_{\tilde{R}} \cdot \left(\begin{bmatrix} q \\ r \end{bmatrix} - \Pi_R \tilde{R}^\top (R_F^D \{\omega_{F/I}\}_F + \{\omega_{D/F}\}_D) \right). \quad (8)$$

This equation describes the PF kinematic *attitude-error* dynamics of the frame \mathcal{W} with respect to the frame \mathcal{D} . The PF kinematic-error dynamics \mathcal{G}_e are thus obtained by combining Eqs. (3) and (8):

$$\mathcal{G}_e : \begin{cases} \dot{p}_F]_F = -\dot{\ell} \vec{t} - \omega_{F/I} \times p_F + v \vec{w}_1, \\ \dot{\Psi}(\tilde{R}) = e_{\tilde{R}} \cdot \left(\begin{bmatrix} q \\ r \end{bmatrix} - \Pi_R \tilde{R}^\top (R_F^D \{\omega_{F/I}\}_F + \{\omega_{D/F}\}_D) \right). \end{cases} \quad (9)$$

In the kinematic-error model in (9), $q(t)$ and $r(t)$ play the role of control inputs, while the rate of progression $\dot{\ell}(t)$ of the point P along the path becomes an extra variable that can be manipulated at will. At this point, it is convenient to formally define the PF generalized error vector $x_{PF}(t)$ as

$$x_{PF} \triangleq \begin{bmatrix} p_F^\top, & e_{\tilde{R}}^\top \end{bmatrix}^\top. \quad (10)$$

Using the formulation above and given a feasible spatially defined path $p_d(\ell)$, we next define the problem of path following for a single vehicle.

Definition 1 (Path-Following Problem (PFP)) *For a given UAV, design feedback control laws for pitch rate $q(t)$, yaw rate $r(t)$, and rate of progression of the virtual target along the path $\dot{\ell}(t)$ such that all closed-loop signals are bounded and the kinematic PF generalized error vector $x_{PF}(t)$ converges to a neighborhood of the origin, independently of the temporal assignments of the mission.*

Stated in simple terms, the problem above amounts to designing feedback laws so that a UAV converges to and remains inside a tube centered on the desired path curve assigned to this UAV, for an arbitrary speed profile.

III. 3D Path Following

This section describes an outer-loop 3D path-following nonlinear control algorithm that uses vehicle angular rates to steer the vehicle along the spatial path $p_d(\ell)$ for any feasible speed profile. The PF controller design builds on the previous work by the authors on PF control of small UAVs, reported in [8], and derives new PF control laws on $SO(3)$. In this paper, we address only the kinematic equations of the UAV by taking $q(t)$ and $r(t)$ as virtual outer-loop control inputs. In particular, similarly to the approach used in [11], we demonstrate that there exist stabilizing functions for $q(t)$ and $r(t)$ leading to local exponential stability of the origin of \mathcal{G}_e (Eq. (9)) with a prescribed domain of attraction.

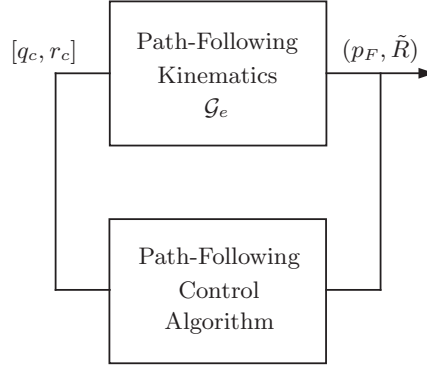


Figure 3: Path-following closed-loop system for a single UAV solved at a kinematic level.

III.A. Nonlinear Control Design using UAV Kinematics

Recall from Section II that the main objective of the PF control algorithm is to drive the position error $p_F(t)$ and the attitude error $e_{\tilde{R}}(t)$ to zero. At the kinematic level, these objectives can be achieved by determining feedback control laws for $q(t)$, $r(t)$, and $\dot{\ell}(t)$ that ensure that the origin of the kinematic-error equations in (9) is exponentially stable with a given domain of attraction. Figure 3 presents the kinematic closed-loop system considered in this section.

To solve the PF problem, we first let the rate of progression of the point P be governed by

$$\dot{\ell} = (v \vec{w}_1 + K_\ell p_F) \cdot \vec{t}, \quad (11)$$

where K_ℓ is a positive constant gain and p_F is the path-following position error vector defined earlier in the paper. Then, the control inputs $q_c(t)$ and $r_c(t)$ chosen as

$$\begin{bmatrix} q_c \\ r_c \end{bmatrix} \triangleq \Pi_{\tilde{R}} \tilde{R}^\top (R_F^D \{\omega_{F/I}\}_F + \{\omega_{D/F}\}_D) - 2K_{\tilde{R}} e_{\tilde{R}}, \quad (12)$$

where $K_{\tilde{R}}$ is also a positive constant gain, stabilize the subsystem \mathcal{G}_e (Eq. (9)). A formal statement of this result is given in the lemma below.

Lemma 1 Assume that the UAV speed $v(t)$ verifies the following bounds:

$$0 < v_{\min} \leq v(t) \leq v_{\max}, \quad \forall t \geq 0, \quad (13)$$

where v_{\min} and v_{\max} denote respectively the minimum and maximum operating speeds of the UAV. If, for given positive constants $c < \frac{1}{\sqrt{2}}$ and c_1 , one chooses the PF control parameters K_ℓ , $K_{\tilde{R}}$, and d such that

$$K_{\tilde{R}} K_p > \frac{v_{\max}^2}{c_1^2 (1 - 2c^2)^2}, \quad (14)$$

where K_p is defined as

$$K_p \triangleq \min \left\{ K_\ell, \frac{v_{\min}}{(d^2 + c^2 c_1^2)^{\frac{1}{2}}} \right\}.$$

then the control inputs in (12), together with the rate of progression of the virtual target in (11), ensure that the origin of the kinematic-error equations in (9) is exponentially stable with guaranteed rate of convergence

$$\lambda_{\text{PF}}^* \triangleq \frac{K_p + K_{\tilde{R}}(1 - c^2)}{2} - \frac{1}{2} \left((K_p - K_{\tilde{R}}(1 - c^2))^2 + \frac{4(1 - c^2)}{c_1^2 (1 - 2c^2)^2} v_{\max}^2 \right)^{\frac{1}{2}}, \quad (15)$$

and corresponding domain of attraction

$$\Omega_c \triangleq \left\{ (p_F, \tilde{R}) \in \mathbb{R}^3 \times \text{SO}(3) \mid \Psi(\tilde{R}) + \frac{1}{c_1^2} \|p_F\|^2 \leq c^2 < \frac{1}{2} \right\}. \quad (16)$$

Proof: The proof of this result, which uses some insights from [15], can be found in [16]. \square

Remark 1 The choice of the characteristic distance d in the definition of the auxiliary frame \mathcal{D} (see Eqs. (4)) can be used to adjust the rate of convergence for the PF closed-loop system. This is consistent with the fact that a large parameter d reduces the penalty for cross-track position errors, which results in a slow rate of convergence of the UAV to the path. Figure 4 clearly illustrates this point. When $d \sim \infty$, the UAV never converges to the path (Figure 4a). For large values of d , the rate of convergence of the UAV to the desired path is slow (Figure 4b), which implies that the UAV takes a large amount of time to reach the desired path. On the other hand, small values of d allow for a high rate of convergence to increase (subject to the design of the gains K_ℓ and $K_{\tilde{R}}$), which however might result in oscillatory path-following behavior (Figure 4c).

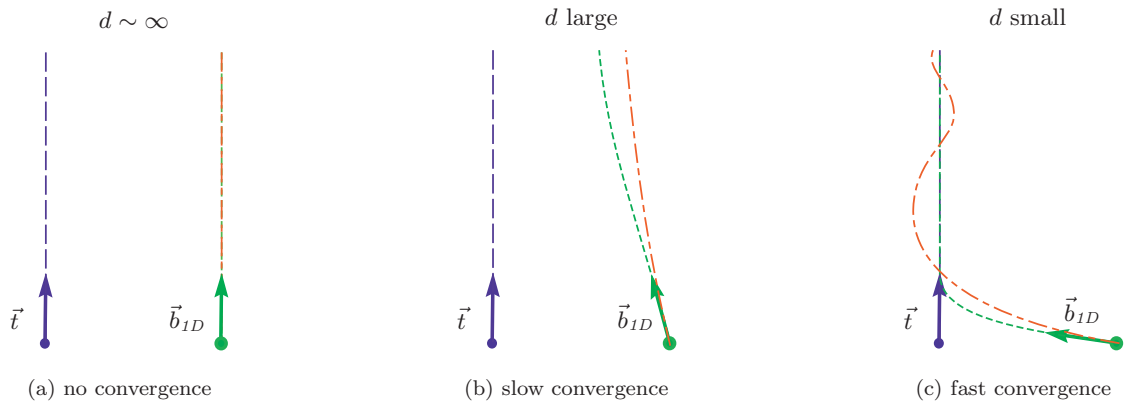


Figure 4: Effect of the characteristic distance d on the convergence of the UAV to the path. (blue: desired path; green: desired approach curve; red: resulting UAV trajectory)

IV. Experimental Results

This section presents a set of key experimental results that illustrates the performance of the developed PF algorithm on a small tactical fixed-wing UAV, demonstrating the benefits of the proposed framework. The PF control law was first implemented in a HIL simulation environment, and then tested in flight at Camp Roberts, CA. For the sake of completeness, this section includes a description of the Rapid Flight Control Prototyping System (RFCPS) [17] used for these tests.

IV.A. Airborne System Architecture

The PF control algorithm was implemented on an experimental UAV Rascal operated by NPS, and thoroughly tested in HIL simulations and in numerous flights at Camp Roberts, CA. The payload bay of the aircraft was used to house the Piccolo Plus AP [17] and a PC104 embedded computer running the algorithms in real-time at 100 Hz, while communicating with the AP over a full duplex serial link at 50 Hz. The main command and control link of the AP is not used in the experiment but preserved for safety reasons. Instead, the onboard avionics were augmented with a wireless mesh communication link to allow for real-time control, tuning, and performance monitoring of the developed software. In particular, this link is used to (bidirectionally) exchange telemetry data in real-time between the AP and the ground control computer. This telemetry includes positional, velocity, acceleration, and rates data, as well as control messages of the Piccolo communication protocol [17]. The experimental setup is shown in Figure 5. The main benefit of this configuration relies on two primary facts. First, the control code resides onboard and directly communicates

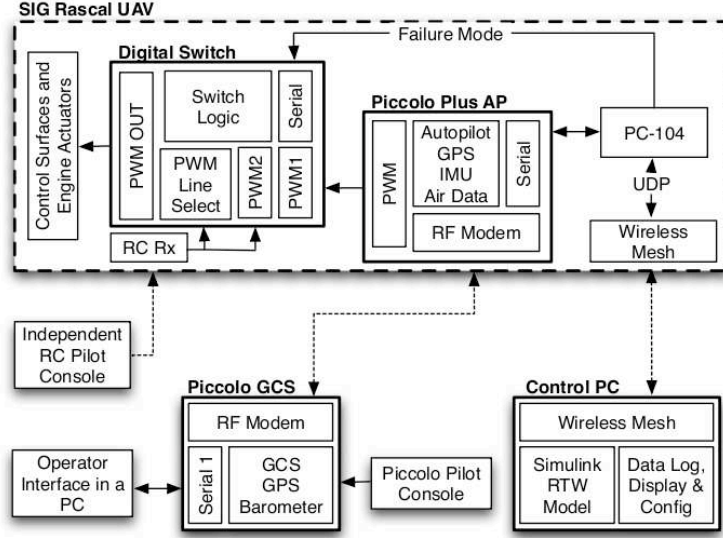


Figure 5: Avionics architecture.

with the inner-loop controller, therefore eliminating any communications delays and dropouts. Second, both the HIL architecture and the actual flight setup—including any possible online modification of the control system parameters—are identical. This allows for a seamless transition of the developed algorithm from the simulation environment to flight testing. More details on the architecture of the developed flight-test system and its current applications can be found in [18].

The concept of operations for both HIL simulation experiments and in-flight testing distinguishes several sequential phases that facilitate the tuning of the PF algorithm and ensure a safe operation of the UAV. Initially, while the AP is in a conventional waypoint navigation mode, a request is sent from the ground control computer to the onboard PC104 over a wireless link. This request sends the desired initial (I.C.) and final condition (F.C.) for the path generation, and the control parameters for the outer-loop PF controller. The I.C. along with the F.C. provide boundary conditions for the path generation algorithm. As soon as a feasible path is generated onboard, the entire onboard segment transitions to the *path-following mode*, and from that moment on, the onboard controller tracks the desired path until the UAV arrives at the F.C., upon which the system can be either automatically stopped, transferring the simulated UAV to the standard waypoint mode, or new I.C. and F.C. can be automatically specified allowing for the experiment to be continued.

IV.B. Hardware in the Loop Simulation Results

HIL simulation results demonstrating the efficiency of the PF control law are shown in Figures 6 and 7. These figures present the results obtained for two different scenarios; the first one considers a UAV that is tasked to follow a straight line, while the second one illustrates the performance of the path-following control law for a path with aggressive turns. The data presented next include the 2D projection of the desired path and the actual UAV track, the commanded $r_{\text{cmd}}(t)$ and measured $r(t)$ turn-rate responses, and the path-following errors x_F , y_F , and z_F resolved in the parallel transport frame. The set of the PF control parameters used during these HIL simulation experiments is given by:

$$d = 75 \text{ m}, \quad K_{\bar{R}} = 1.25, \quad K_{\ell} = 2.5.$$

The speed command is fixed at 22 m/s, while the commanded turn rate is limited to 0.3 rad/s. For safety purposes, the bank angle is limited to 25 deg, which reduces the turn-rate capability to about 0.2 rad/s.

Figure 6 shows the HIL simulation results for the first scenario, where the UAV is tasked to follow a straight path starting with an initial error of 200 m in the y -axis. The characteristic distance d is significantly smaller than the initial cross-track error, which results in an aggressive approach to the path. In fact, the turn-rate command saturates at the beginning of the path-following maneuver. The UAV converges to the

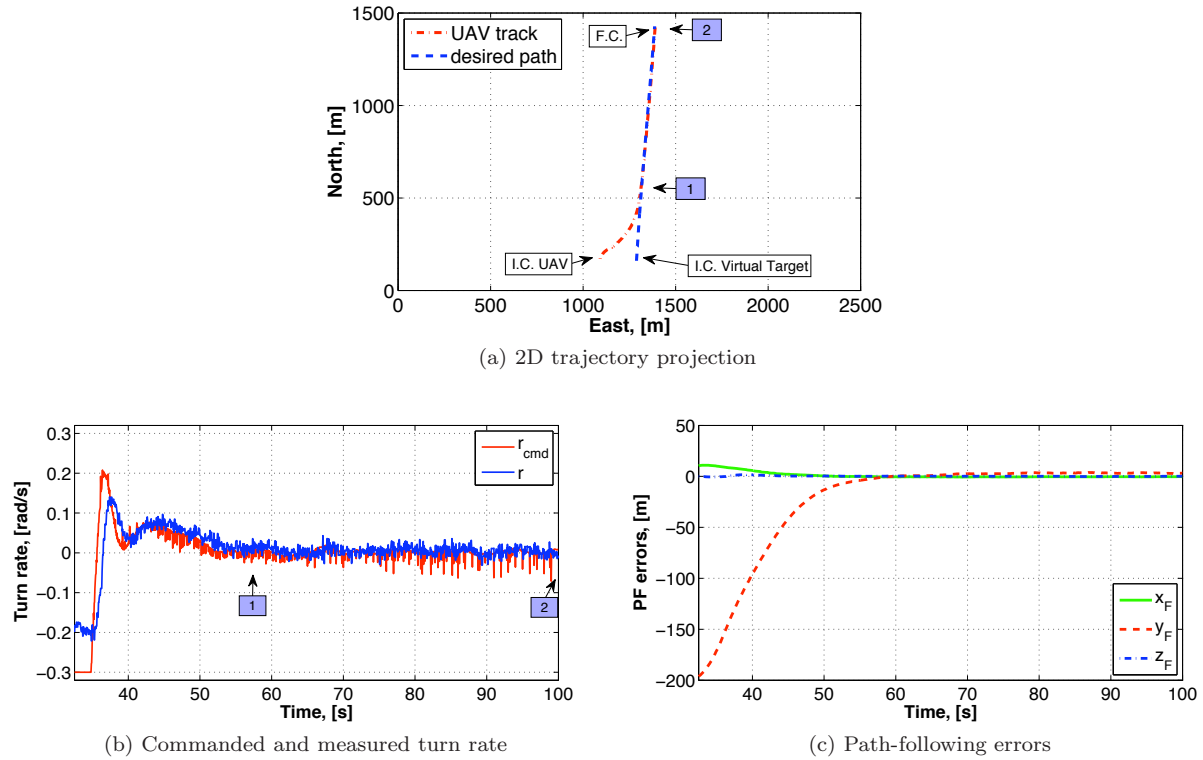


Figure 6: HIL Simulation 1: Straight line with initial offset PF error on y -axis.

path in about 60 s and, from point 1 to point 2, the UAV perfectly follows the path, with the turn rate close to zero (Figure 6b) and the PF errors below 5 m (Figure 6c). The use of a parallel transport frame (rather than a Frenet-Serret frame) in the problem formulation is of particular importance in this simulation scenario, where the desired path has zero curvature and the Frenet-Serret frame is thus not well defined. In fact, the use of Frenet-Serret frames in such scenarios might cause a “destabilizing effect” that leads to an oscillatory path-following behavior; this issue is illustrated in detail in Appendix A.

In the second simulation scenario (Figure 7), the UAV is tasked to follow an S-shaped path with two aggressive turns, starting again with an initial error of 200 m in the y -axis. Figure 7a presents the 2D horizontal projections of the desired path and the actual UAV track, while Figures 7b and 7c show respectively the turn-rate command with the turn rate of the UAV and the path-following errors. Initially, similar to the previous scenario, the characteristic distance d is significantly smaller than the initial cross-track error, which results in an initial aggressive right turn of the UAV towards the desired path. The turn-rate command saturates again for a few seconds at the beginning of the path-following maneuver. After this initial aggressive turn, the UAV smoothly converges to the path with turn-rate commands within the achievable ± 0.2 rad/s range. During this approach phase, the UAV is also able to negotiate a sharp right turn maintaining a small cross-track position error. Next, the UAV follows a straight leg, keeping the turn rate close to zero and the PF errors below 5 m. The UAV performs then a sharp left turn, which leads to turn-rate command saturation for about 6 s, and results in the UAV accumulating a cross-track error of approximately 20 m. Finally, the UAV converges back to the desired path in about 10 s, maintaining PF position errors below 5 m.

IV.C. Flight Test Results

This section presents flight-test results of the real-time implementation of the PF control system developed in this paper. In particular, we consider again two scenarios; the first one considers a UAV that is tasked to follow “mild” path with small initial cross-track position errors, while the second one considers the case of a

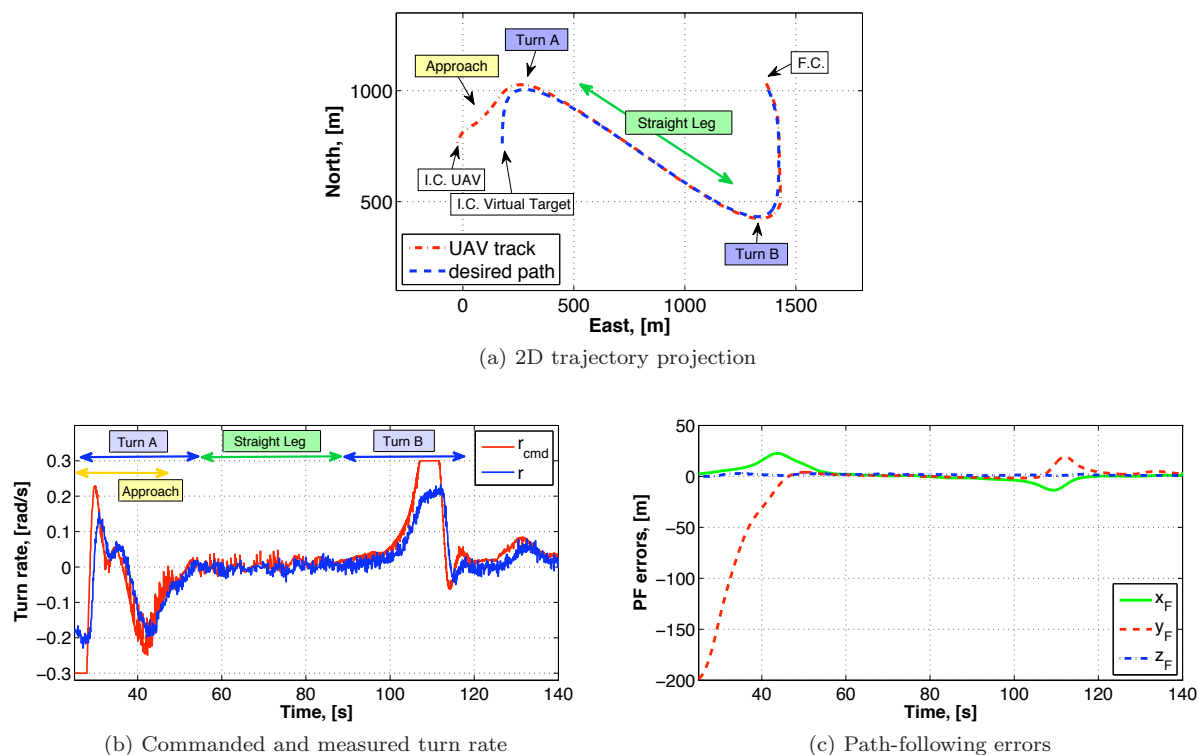


Figure 7: HIL Simulation 2: S-shaped path with initial offset PF error on y -axis.

UAV following a quasi-straight path, but starting with a large initial error in both the x - and y -axes. The purpose of this second scenario illustrates the convergence properties of the closed-loop system in the presence of both cross-track and along-track positional errors. The data presented next include the 2D projection of the desired path and the actual UAV track, and the path-following errors x_F , y_F , and z_F resolved in the parallel transport frame. The same set of the PF control parameters is used during the flight testing:

$$d = 75 \text{ m}, \quad K_{\bar{R}} = 1.25, \quad K_{\ell} = 2.5.$$

During these flight tests, the speed command was fixed at 22 m/s, while the commanded turn rate was limited to 0.12 rad/s for safety reasons.

The results for the first flight-test scenario are shown in Figure 8, which include the 2D horizontal projections of the desired path and the actual UAV track (Figure 8a), and the PF position errors (Figure 8b). Results show that the UAV is able to follow the path, keeping the cross-track PF position errors within ± 7 m during the whole experiment, and down to ± 3 m after the initial convergence phase. This scenario is a clear example of the fact that the developed PF control architecture outperforms the functionality of the conventional waypoint navigation method, enabling a UAV with an off-the-shelf autopilot to follow with high accuracy a predetermined path that it was not otherwise designed to follow.

Finally, in the second flight-test scenario (Figure 9), the UAV is tasked to follow a quasi-straight path starting with large initial cross-track and along-track position errors. During the approach phase, the large initial cross-track error causes the turn-rate command to saturate at 0.12 rad/s, which results in a smooth convergence to the desired path. The UAV converges to the path in about 35 s, and its trajectory during this initial convergence phase can be seen in Figure 9a. The convergence of the PF position errors to a neighborhood of the origin is illustrated in Figure 9b. In particular, this figure shows that the feedback law derived for the rate of progression of the virtual target along the path (see Eq. (11)) results in a robust convergence of the along-track position error along to a neighborhood of the origin. Moreover, similar to the first HIL simulation scenario, the use of a parallel transport frame in the problem formulation is of particular importance in this scenario, where the desired path has zero curvature.

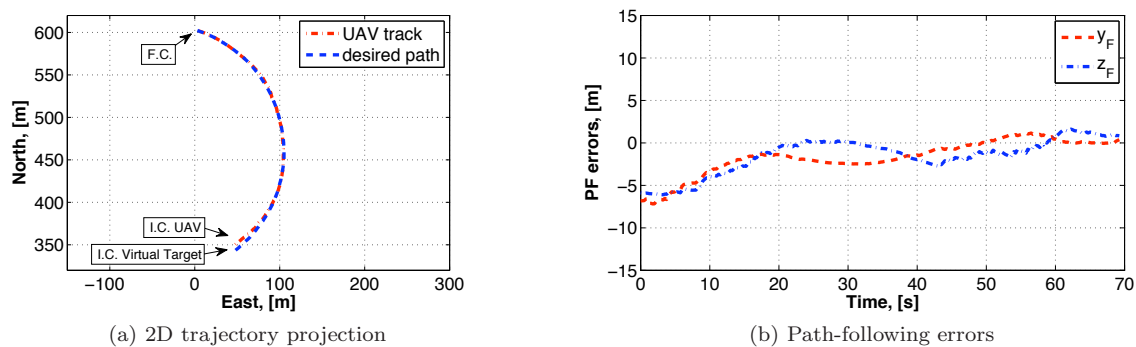


Figure 8: Flight Test 1: Path following of a “mild” path with small initial cross-track position errors.

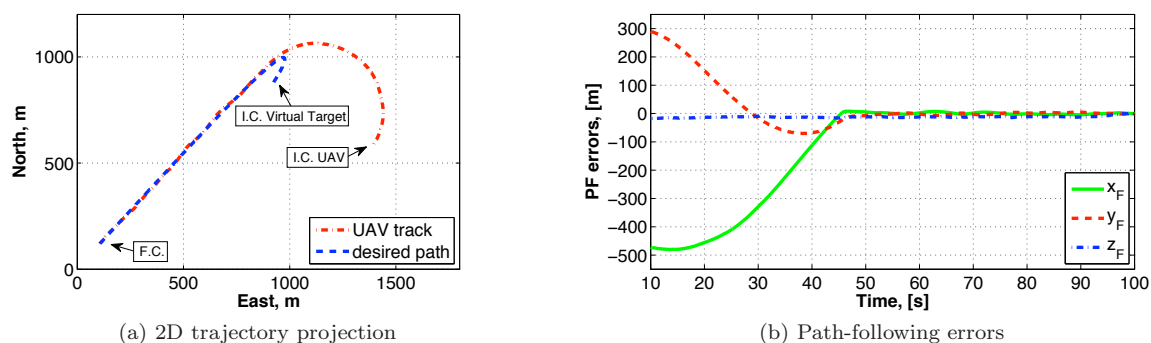


Figure 9: Flight Test 2: Path following of a quasi-straight path starting with large initial cross-track and along-track position errors.

V. Conclusions

This paper presented a new solution to the problem of 3D path-following control for a single UAV. The novelty of the proposed solution relies on the use of the Special Orthogonal group $SO(3)$ in the formulation of the path-following attitude control problem. This formulation avoids the geometric singularities and complexities that appear when dealing with local parameterizations of the vehicle’s attitude, and also the ambiguities when using quaternions for attitude representation. The approach also yields an inner-outer control structure, and it thus allows to take explicit advantage of the fact that UAVs are normally equipped with commercial autopilots providing angular-rate and speed tracking capabilities.

The developed architecture outperforms the functionality of the conventional waypoint navigation method, enabling a UAV with an off-the-shelf autopilot to follow a predetermined aggressive path that it was not otherwise designed to follow. Both theoretical and practical aspects of the problem are addressed. Hardware-in-the-loop simulations and flight test results illustrate the efficacy of the framework developed for path-following control. Future work will extend the developed path-following control architecture to multiple UAV scenarios, including realistic ISR missions, multiple UAV collision avoidance, and integration of UAVs with manned aircraft.

References

- ¹A. Pedro Aguiar and António M. Pascoal. Dynamic positioning and way-point tracking of underactuated AUVs in the presence of ocean currents. *International Journal of Control*, 80(7):1092–1108, July 2007.
- ²Antonios Tsourdos, Brian A. White, and Madhavan Shanmugavel. *Cooperative Path Planning of Unmanned Aerial Vehicles*. John Wiley & Sons, Chichester, UK, 2011.

³Researchers and Collaborators, Control Science Center of Excellence, U.S. Air Force Research Laboratories. *UAV Cooperative Decision and Control: Challenges and Practical Approaches*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2009.

⁴Alain Micaelli and Claude Samson. Trajectory tracking for unicycle-type and two-steering-wheels mobile robot. Technical Report 2097, INRIA, Sophia-Antipolis, France, November 1993.

⁵Claude Samson and Karim Ait-Abderrahim. Mobile robot control part 1: Feedback control of a non-holonomic mobile robots. Technical Report 1288, INRIA, Sophia-Antipolis, France, June 1990.

⁶Pedro Encarnação, António M. Pascoal, and Murat Arcak. Path following for marine vehicles in the presence of unknown currents. In *6th IFAC Symposium on Robot Control*, Vienna, Austria, 2000.

⁷Pedro Encarnação and António M. Pascoal. 3D path following for autonomous underwater vehicles. In *39th IEEE Conference on Decision and Control*, Sydney, Australia, 2000.

⁸Isaac Kaminer, António Pascoal, Enric Xargay, Naira Hovakimyan, Chengyu Cao, and Vladimir Dobrokhodov. Path following for unmanned aerial vehicles using \mathcal{L}_1 adaptive augmentation of commercial autopilots. *Journal of Guidance, Control and Dynamics*, 33(2):550–564, March–April 2010.

⁹A. Pedro Aguiar, João P. Hespanha, and Petar V. Kokotović. Performance limitations in reference tracking and path following for nonlinear systems. *Automatica*, 44(3):598–610, March 2008.

¹⁰María M. Seron, Julio H. Braslavsky, Petar V. Kokotović, and David Q. Mayne. Feedback limitations in non-linear systems: From Bode integrals to cheap control. *IEEE Transactions on Automatic Control*, 44(4):829–833, 1999.

¹¹Enric Xargay, Vladimir Dobrokhodov, Isaac Kaminer, António Pascoal, Naira Hovakimyan, and Chengyu Cao. Cooperative control of autonomous systems. Submitted to *IEEE Control Systems Magazine*, 2010.

¹²Didik Soetanto, Lionel Lapierre, and António M. Pascoal. Adaptive, non-singular path-following control of dynamic wheeled robots. In *International Conference on Advanced Robotics*, pages 1387–1392, Coimbra, Portugal, June–July 2003.

¹³Richard L. Bishop. There is more than one way to frame a curve. *The American Mathematical Monthly*, 82(3):246–251, 1975.

¹⁴Andrew J. Hanson and Hui Ma. Parallel transport approach to curve framing. Technical report, Indiana University Compute Science Department, 1995.

¹⁵Taeyoung Lee, Melvin Leok, and N. Harris McClamroch. Control of complex maneuvers for a quadrotor UAV using geometric methods on $SE(3)$. *IEEE Transactions on Automatic Control*, 2010. Submitted. Available online: [arXiv:1003.2005v3](https://arxiv.org/abs/1003.2005v3).

¹⁶Enric Xargay, Isaac Kaminer, António M. Pascoal, Naira Hovakimyan, Vladimir Dobrokhodov, Venanzio Cichella, A. Pedro Aguiar, and Reza Ghabcheloo. Time-critical cooperative path following of multiple UAVs over time-varying networks. To be submitted to *IEEE Transactions on Control System Technology*, 2011.

¹⁷J. Burl. Piccolo/Piccolo Plus Autopilots - Highly Integrated Autopilots for Small UAVs. <http://cloudcaptech.com>.

¹⁸Vladimir Dobrokhodov, Oleg Yakimenko, Kevin D. Jones, Isaac Kaminer, Eugene Bourakov, Ioannis Kitsios, and Mariano Lizarraga. New generation of rapid flight test prototyping system for small unmanned air vehicles. In *Proc. of AIAA Modelling and Simulation Technologies Conference*, Hilton Head Island, SC, August 2007. AIAA-2007-6567.

Appendix

A. The Frenet Frame and the Parallel Transport Frame

In the past, the Frenet-Serret Frame has been used in numerous works as a frame formulation for the generated reference path. The Frenet-Serret Frame is defined as follow: if $\vec{p}(t)$ is a thrice-differentiable space curve, its tangent, binormal, and normal vectors at a point on the curve are given by

$$\begin{aligned}\vec{T}(t) &= \frac{\vec{p}'(t)}{\|\vec{p}'(t)\|} \\ \vec{B}(t) &= \frac{\vec{p}'(t) \times \vec{p}''(t)}{\|\vec{p}'(t) \times \vec{p}''(t)\|} \\ \vec{N}(t) &= \vec{B}(t) \times \vec{T}(t).\end{aligned}$$

Intuitively, the Frenet frame's normal vector $\vec{N}(t)$ always points toward the center of the osculating circle. Thus, when the orientation of the osculating circle changes drastically, or the second derivative of the curve becomes very small (i.e. straight line), the Frenet frame behaves erratically or may even become undefined.

On the other hand, the mathematical properties of the Parallel Transport Frame follow from the observation that, while $\vec{T}(t)$ for a given model is unique, we may choose any convenient arbitrary basis $(\vec{N}_1(t), \vec{N}_2(t))$ for the remainder of the frame, so long as it is in the plane perpendicular to $\vec{T}(t)$ at each point. If derivatives of $(\vec{N}_1(t), \vec{N}_2(t))$ depends only on $\vec{T}(t)$ and not each other, we can make $\vec{N}_1(t)$ and $\vec{N}_2(t)$ vary smoothly throughout the path regardless of the curvature. Section IV.B shows that, while the UAV follows a straight line, no singularities are presented, therefore the path following algorithm is not affected.

B. The *hat* and *vee* maps¹⁵

The *hat map* $(\cdot)^\wedge : \mathbb{R}^3 \rightarrow so(3)$ is defined as

$$(x)^\wedge = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}$$

for $x = [x_1, x_2, x_3]^\top \in \mathbb{R}^3$. The inverse of the hat map is referred to as the *vee map* $(\cdot)^\vee : so(3) \rightarrow \mathbb{R}^3$. A property of the hat and vee maps used in this paper is given below:

$$\text{tr}[(x)^\wedge M] = \text{tr}[M(x)^\wedge] = \frac{1}{2} \text{tr}[(x)^\wedge (M - M^\top)] = -x \cdot (M - M^\top)^\vee, \quad (17)$$

for any $x \in \mathbb{R}^3$, and $M \in \mathbb{R}^{3 \times 3}$. We refer to [15] for further details on the hat and vee maps.