An Iterative Model Predictive Control Algorithm for UAV Guidance

FRANCISCO GAVILAN, Senior Member
RAFAEL VAZQUEZ, Member, IEEE
EDUARDO F. CAMACHO, Fellow, IEEE
Universidad de Sevilla
Sevilla, Spain

A novel aircraft path–following guidance algorithm based on model predictive control is proposed in this paper. The algorithm tracks a precomputed trajectory and produces reference commands for a low-level attitude controller. To solve the associated nonlinear optimization problem, an iterative scheme is proposed, using as a feasible hotstart the guidance solution provided by a well-behaved L1 navigation law. Simulations show the effectiveness of the algorithm, even in the presence of disturbances such as wind.

Manuscript received February 26, 2014; revised October 19, 2014; released for publication March 20, 2015.

DOI. No. 10.1109/TAES.2015.140153.

Refereeing of this contribution was handled by M. Idan.

Authors' addresses: Universidad de Sevilla, Dpto. Ingenieria Aeroespacial, Camino de los Descubrimientos s.n. Sevilla, 41092 Spain. Corresponding author is R. Vazquez, E-mail: (rvazquez1@us.es).

I. INTRODUCTION

Recent years have seen a considerable growth in the development of flight control systems for unmanned air vehicles (UAVs). One of the key components of a flight control system (often grossly simplified) is the guidance system, which should guarantee that the airplane is able to safely and efficiently accomplish its mission; a good design should also be able to improve the autonomy and the performance of the aircraft.

The object of this paper is the design of a path-following guidance law that generates commands to the attitude control system to follow a given reference trajectory. Thus, a path-following guidance algorithm requires a compatible flight planner and attitude control system; the outputs of the flight planner and the inputs of the attitude controller become, respectively, the inputs and outputs of the guidance law. This three-layer aircraft control framework, with guidance in the middle layer, is common in UAVs (see, e.g., [1, 2]).

Traditionally, airplane guidance systems have been based on classical missile guidance laws (see [3–5]). For instance, in [6, 7], proportional navigation and pure pursuit are applied to UAVs. Similarly, in [8], a "good helmsman behavior" strategy is employed for UAV path following. The basis of these algorithms is to introduce a *virtual* target that follows the reference and then apply missile guidance to track the virtual target. While these laws are simple (being easily implemented onboard) and robust, they are reactive (not using future information about the reference) and tend to be aggressive, often demanding high control power (leading to efficiency loss).

These limitations have motivated the development of guidance laws based on modern control techniques, such as gain scheduling [9], optimal control [10], sliding mode control [11], control Lyapunov functions [12], vector fields [13], or even extremum seeking [14]. An excellent survey of different guidance techniques for fixed wing UAV can be found in [15].

A control method with potential applicability to guidance is model predictive control (MPC) [16]. This technique is based on the idea of computing an optimal sequence of control signals that optimizes a certain cost function, possibly with state and control constraints. Once the first inputs are applied and the aircraft moves, the prediction and optimization steps are again repeated, and a new control sequence is obtained, thus closing the loop. Unfortunately, this procedure is difficult to implement because it requires solving a nonlinear optimization problem online. In addition, there is no guarantee of finding a feasible control solution (which could compromise the mission safety).

Examples of MPC guidance already exist. For instance, in [17], the authors introduce a decentralized MPC controller for coordination of two UAV for seeking a ground reference (in which the nonlinear optimization problem is solved by the gradient method). Another cooperative MPC strategy is shown in [18], which

addresses the coordination of multiple UAV solving a distributed mixed-integer linear programming (MILP) problem. Similarly, [19] presents a MPC guidance scheme based on a robust MILP optimization problem with applications to UAV collision avoidance. On the other hand, [20] formulates a predictive guidance law on the basis of a linearized prediction model, leading to a simpler and easy-to-implement optimization process. In [21], MPC is applied for cooperative control of multiple vehicles, including communication failures. Another example of MPC applied to cooperative search by multiple UAVs is given in [22]. In [23], the authors develop MPC guidance laws with controlled arrival time. Similarly, [24] shows both experiments and simulations of an MPC guidance algorithm based on offline linearizations computed along a reference trajectory. Finally, in [25], genetic algorithms are used to solve the associated optimization problem.

In this work, we propose a novel path-following guidance strategy based on MPC that computes optimized values of airspeed, flight path angle, and bank angle. These are commanded to a low-level attitude control system in order to make the UAV follow a precomputed reference trajectory (given by straight flight segments). Inspired by [26], the nonlinear optimization problem is addressed by iteratively solving a set of linearized problems around a nominal controls sequence. Also, the algorithm is initialized by a complementary low-level guidance law based on L_1 navigation law (L1), a robust law that guarantees having an initial feasible solution and that provides stability properties [27].

The contribution of this paper is the design of a guidance law that enjoys the benefits of MPC techniques (in the sense of optimality, constraints handling, and use of precomputed reference knowledge) and solves its instantaneity and feasibility problems by the use of L1 to generate a feasible solution and linearization to quickly improve the solution. To our knowledge, our approach is novel compared with previous works. Additional features include time synchronization and disturbance rejection, properties that are not usually provided by other algorithms in the literature.

The paper is structured as follows. First, Section II presents the model used in this work. Section III formulates the MPC guidance law. Section IV deals with obtaining a feasible initial solution by using L1 guidance, while Section V gives a practical algorithm to find the MPC solution. Simulation results are shown in Section VI, and, finally, Section VII closes the paper with some remarks.

II. AIRPLANE MODEL

The low-level attitude controller used in this work (see [28]) is able to follow airspeed (V), flight path angle (γ), and bank angle (ϕ) references. Thus, the guidance law has to generate these inputs to follow a reference trajectory. To compute the airplane position from the inputs, a 3-DoF

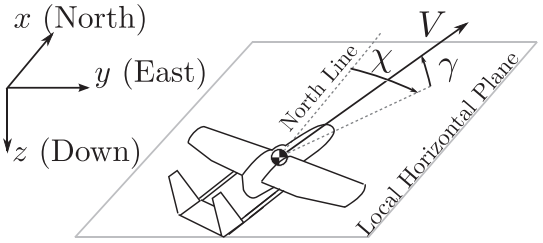

Fig. 1. Airplane model.

kinematic aircraft model is used [29]:

$$\frac{\mathrm{d}x}{\mathrm{d}t} = V\cos\gamma\cos\chi + w_x, \tag{1}$$

$$\frac{\mathrm{d}y}{\mathrm{d}t} = V\cos\gamma\sin\chi + w_y, \tag{2}$$

$$\frac{\mathrm{d}z}{\mathrm{d}t} = -V\sin\gamma + w_z, \tag{3}$$

where, as can be seen in Fig. 1, x, y, and z are the coordinates of the center of gravity (formulated in a NED frame); w_x, w_y, and w_z denote, respectively, the north, east, and down components of the wind speed; and χ is the heading angle. For the sake of clarity, the wind speed will be omitted momentarily (its effect will be included afterward as an additive disturbance).

Most previous works use a discretization scheme based on the heading angle, velocity, and flight path angle. In particular, if the heading angle is assumed constant at each discretization step, this is equivalent to assuming that the airplane trajectory is split in straight segments with constant velocity, flight path angle, and heading angle (see, e.g., [30]). In addition, it is common to uncouple the horizontal and vertical kinematics by assuming $\cos\gamma \approx 1$ and $\sin\gamma \approx \gamma$. While these assumptions provide a simple prediction law (simplifying the optimization problem), they unrealistically imply that the airplane can perform instantaneous changes of the heading angle. We instead use the bank angle (a much faster state variable) and split the trajectory in segments with constant velocity V, constant path angle γ, and constant bank angle ϕ. These hypotheses imply that the horizontal movement is composed by constant curvature segments (with continuously changing heading angles), which is a more realistic approximation. In addition, this strategy implies that there is no need of a low-level heading controller since this angle is now a state variable, directly managed by the guidance system. Also, it allows explicitly including limitations in turns as constraints.

Thus, we consider that the aircraft performs steady horizontal turns with zero sideslip and small values of γ. Then it follows that

$$\frac{\mathrm{d}\chi}{\mathrm{d}t} = \frac{g\tan\phi}{V}. \tag{4}$$

Next, a discrete-time evolution law is obtained from the continuous model (1)–(4). Assume a sampling interval of

duration T_s and denote by V_k, γ_k, ϕ_k, x_k, y_k, z_k, and χ_k the values of the states and inputs at time t_k.

Under the assumption of constant V_k, γ_k, and ϕ_k, the time evolution of the heading angle for $t \in [t_k, t_{k+1}]$ is

$$\chi(t) = \frac{g \tan \phi_k}{V_k} (t - t_k) + \chi_k. \tag{5}$$

Substituting (5) in (1)–(3) and integrating for $t \in [t_k, t_{k+1}]$, one obtains

$$x_{k+1} = \frac{V_k \cos \gamma_k T_s}{\kappa_k} (\sin(\kappa_k + \chi_k) - \sin \chi_k) + x_k, \tag{6}$$

$$y_{k+1} = \frac{V_k \cos \gamma_k T_s}{\kappa_k} (\cos \chi_k - \cos(\kappa_k + \chi_k)) + y_k, \tag{7}$$

$$z_{k+1} = -V_k \sin \gamma_k T_s + z_k, \tag{8}$$

$$\chi_{k+1} = \kappa_k + \chi_k, \tag{9}$$

where for compactness we denote

$$\kappa_k = \frac{g \tan \phi_k T_s}{V_k}. \tag{10}$$

This discrete (nonlinear) model provides the future state $\mathbf{x}_{k+1} = [x_{k+1} \quad y_{k+1} \quad z_{k+1} \quad \chi_{k+1}]^T$ as a function of their previous value $\mathbf{x}_k$ and the present inputs $\mathbf{u}_k = [V_k \quad \gamma_k \quad \kappa_k]^T$.

III. PREDICTIVE GUIDANCE FORMULATION

MPC requires a prediction model to compute (at each sampling time) an optimized control sequence. In practice, given that the model is not exact and there are disturbances, the prediction is not perfect, and one closes the loop by applying only the first control signals and next recomputing the optimal input sequence. Thus, MPC requires solving an optimization problem at each time step.

Since the prediction model (6)–(9) is nonlinear in the control sequence, one would need to solve a challenging nonlinear optimization program at every time step. This is unfeasible given the onboard computational limitations.

To overcome this problem, first we compute a *feasible* control sequence from a complementary and robust guidance algorithm that does not require optimization. Next, we obtain a *linear* (and explicit) prediction model by linearizing (6)–(9) around this precomputed control sequence. Then a standard linear optimization problem is solved, and its solution is used to improve the linearization and optimize again. This iterative process is followed until a local minimum is found or the computational time is up. Furthermore, to cope with disturbances (e.g., due to wind), an online disturbance estimator is included in the control strategy.

Next, we separately describe the state prediction model, the disturbance estimator, the constraints formulation, and the definition of the objective function.

A. Linearization of the State Prediction Model

Denote the prediction horizon by N_p. Using (9) to propagate the heading angle,

$$\chi_k = \sum_{i=0}^{i=k-1} \kappa_i + \chi_0. \tag{11}$$

Substituting this equation into (6) and (7), one has

$$\mathbf{x}_{k+1} = \mathbf{f}_k(V_k, \gamma_k, \kappa_k, \kappa_{k-1}, \ldots, \kappa_0, \chi_0) + \boldsymbol{\delta}_k + \mathbf{x}_k, \tag{12}$$

where we have introduced a vector $\boldsymbol{\delta}$ to model additive disturbances (such as wind) and where $\mathbf{f}_k$ is defined as

$$\mathbf{f}_k = \begin{bmatrix} 2\frac{V_k \cos \gamma_k T_s}{\kappa_k} \cos\left(\frac{\kappa_k}{2} + \sum_{i=0}^{i=k-1} \kappa_i + \chi_0\right) \sin \frac{\kappa_k}{2} \\ 2\frac{V_k \cos \gamma_k T_s}{\kappa_k} \sin\left(\frac{\kappa_k}{2} + \sum_{i=0}^{i=k-1} \kappa_i + \chi_0\right) \sin \frac{\kappa_k}{2} \\ -V_k \sin \gamma_k T_s \\ \kappa_k \end{bmatrix}. \tag{13}$$

Notice that future states are a nonlinear function of the control signals but depend linearly on the previous states. Since disturbances are unknown, the prediction of future states has to be statistical. Taking the mathematical expectation in (12) and iterating, one can establish the following recursive law:

$$\mathbf{x}_{k+1|0} = \sum_{i=0}^{i=k} \left(\mathbf{f}_i(V_i, \gamma_i, \kappa_i, \kappa_{i-1}, \ldots, \kappa_0, \chi_0) + \bar{\boldsymbol{\delta}}_i\right) + \mathbf{x}_0, \tag{14}$$

where $\mathbf{x}_{i|j}$ denotes the prediction of the state in the instant j given the measurement in the instant j and $\bar{\boldsymbol{\delta}}_i = \mathrm{E}[\boldsymbol{\delta}_i]$.

Assume small increments in the control signals:

$$V_i = \bar{V}_i + \Delta V_i, \quad \gamma_i = \bar{\gamma}_i + \Delta \gamma_i, \quad \kappa_i = \bar{\kappa}_i + \Delta \kappa_i, \tag{15}$$

where $\overline{(\cdot)}_i$ denotes the nominal value of a control signal at the time i, while $\Delta(\cdot)_i$ denotes the increment respect to its nominal value. The complete sequence of control signals during the prediction horizon N_p can be written in a compact fashion by introducing the following "stack" vector:

$$\mathbf{u_S} = \begin{bmatrix} \mathbf{u}_0^T & \mathbf{u}_1^T & \cdots & \mathbf{u}_{N_p-1}^T \end{bmatrix}^T \tag{16}$$

and similarly for the nominal control signals ($\bar{\mathbf{u}}_\mathbf{S}$) and increments ($\Delta \mathbf{u_S}$). Linearizing $\mathbf{f}_i$ from (14) around $\bar{\mathbf{u}}_\mathbf{S}$, one has

$$\mathbf{x}_{k+1|0} = \sum_{i=0}^{i=k} \left(\mathbf{f}_i(\bar{\mathbf{u}}_\mathbf{S}, \chi_0) + \frac{\partial \mathbf{f}_i}{\partial \mathbf{u_S}}(\bar{\mathbf{u}}_\mathbf{S}, \chi_0) \Delta \mathbf{u_S} + \bar{\boldsymbol{\delta}}_i\right) + \mathbf{x}_0, \tag{17}$$

where the matrix $\frac{\partial \mathbf{f}_i}{\partial \mathbf{u_S}}(\bar{\mathbf{u}}_\mathbf{S}, \chi_0)$ (given in the appendix) can be obtained explicitly; thus, the linearization is instantly computed. Introducing now

$$\mathbf{x_S} = \begin{bmatrix} \mathbf{x}_{1|0}^T & \mathbf{x}_{2|0}^T & \cdots & \mathbf{x}_{N_p|0}^T \end{bmatrix}^T \tag{18}$$

and defining $\boldsymbol{\delta_S}$ as the stack vector containing the sequence of mean additive disturbances during the prediction

horizon, the prediction model (17) can be written as

$$\mathbf{x_S} = \mathbf{F} + \mathbf{G_u}\Delta\mathbf{u_S} + \mathbf{G}_\delta\boldsymbol{\delta}_\mathbf{S}, \tag{19}$$

where $\mathbf{F}$ and $\mathbf{G_u}$ are defined as

$$\mathbf{F} = \begin{bmatrix} \mathbf{f}_0(\bar{\mathbf{u}}_\mathbf{S}, \chi_0) + \mathbf{x}_0 \\ \mathbf{f}_1(\bar{\mathbf{u}}_\mathbf{S}, \chi_0) + \mathbf{f}_0(\bar{\mathbf{u}}_\mathbf{S}, \chi_0) + \mathbf{x}_0 \\ \vdots \\ \mathbf{f}_{N_p-1}(\bar{\mathbf{u}}_\mathbf{S}, \chi_0) + \cdots + \mathbf{f}_0(\bar{\mathbf{u}}_\mathbf{S}, \chi_0) + \mathbf{x}_0 \end{bmatrix}, \tag{20}$$

$$\mathbf{G_u} = \begin{bmatrix} \frac{\partial \mathbf{f}_0}{\partial \mathbf{u_S}}(\bar{\mathbf{u}}_\mathbf{S}, \chi_0) \\ \frac{\partial \mathbf{f}_1}{\partial \mathbf{u_S}}(\bar{\mathbf{u}}_\mathbf{S}, \chi_0) + \frac{\partial \mathbf{f}_0}{\partial \mathbf{u_S}}(\bar{\mathbf{u}}_\mathbf{S}, \chi_0) \\ \vdots \\ \frac{\partial \mathbf{f}_{N_p-1}}{\partial \mathbf{u_S}}(\bar{\mathbf{u}}_\mathbf{S}, \chi_0) + \cdots + \frac{\partial \mathbf{f}_0}{\partial \mathbf{u_S}}(\bar{\mathbf{u}}_\mathbf{S}, \chi_0) \end{bmatrix}, \tag{21}$$

and $\mathbf{G}_\delta$ is a block-diagonal matrix, each block being $\mathbf{I}_{4\times 4}$.

B. Estimator of Disturbances

Equation (19) requires values for $\bar{\boldsymbol{\delta}}_i$. If one can measure past disturbances, the following simple estimator[1] can be used:

$$\hat{\boldsymbol{\delta}}_k = \frac{\sum_{i=0}^{k-1} \mathrm{e}^{-\lambda(k-i)}\boldsymbol{\delta}_i}{\sum_{i=0}^{k-1} \mathrm{e}^{-\lambda(k-i)}}, \tag{22}$$

which is the expression of a weighted average with exponential weights that emphasize recent measurements. In (22), $\hat{\boldsymbol{\delta}}_k$ is the estimate of $\bar{\boldsymbol{\delta}}_k$ based on past disturbances, and $\lambda > 0$ is a *forgetting factor* that controls the decay of past measurements.

Following [31], if we define

$$\rho_k = \sum_{i=0}^{k-1} \mathrm{e}^{-\lambda(k-i)} = \frac{\mathrm{e}^{-\lambda}\left(1 - \mathrm{e}^{-\lambda k}\right)}{1 - \mathrm{e}^{-\lambda}}, \tag{23}$$

estimator (22) can be recursively computed as $\hat{\boldsymbol{\delta}}_0 = 0$,

$$\hat{\boldsymbol{\delta}}_k = \frac{\mathrm{e}^{-\lambda}}{\rho_k}\left(\rho_{k-1}\hat{\boldsymbol{\delta}}_{k-1} + \boldsymbol{\delta}_{k-1}\right). \tag{24}$$

Since this approach requires measuring *past disturbances*, a straightforward approach [31] is to use the difference between the real airplane state at each sampling time and the expected state from the prediction in the previous sampling time:

$$\boldsymbol{\delta}_{k-1} = \mathbf{x}_k - \mathbf{f}_k(V_{k-1}, \gamma_{k-1}, \kappa_{k-1}, \chi_{k-1}) - \mathbf{x}_{k-1}. \tag{25}$$

Note that measurement noises in (25) would falsify the estimation of disturbances, in particular those related to estimation of position and velocity. If the aircraft has a well-designed navigation system (typically some type of Kalman filter using IMU, GPS, magnetometer, and barometric measurements), these errors should remain bounded.

[1] More advanced estimators, such as the extended Kalman filter or the unscented Kalman filter, could be used. However, to reduce the computational burden and to simplify the discussion and since we require only the mean of the disturbances, we have chosen this rather simple approach.

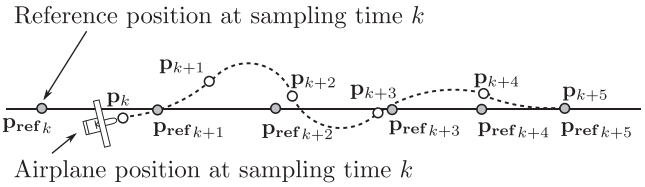

Fig. 2. Oscillations due to choice of cost function.

Disturbances are sampled at a higher frequency than the main guidance law, applying (26) each T_δ seconds ($T_\delta < T_s$). That is,

$$\boldsymbol{\delta}'_{j-1} = \mathbf{x}_j - \mathbf{f}_j\left(\hat{V}_{j-1}, \hat{\gamma}_{j-1}, \hat{\kappa}_{j-1}, \chi_{j-1}\right)\frac{T_\delta}{T_s} - \mathbf{x}_{j-1}, \tag{26}$$

where $\hat{V}$, $\hat{\gamma}$, and $\hat{\kappa}$ denote the real measured controls (not the commanded ones) at each sampling time and $\boldsymbol{\delta}'$ is the vector of mean disturbances along each oversampled interval.

C. Formulation of Constraints

First, due to operational limitations of the airplane, there are maximum and minimum values for the control signals:

$$\mathbf{u}_{min} \le \mathbf{u}_k \le \mathbf{u}_{max}. \tag{27}$$

Writing these limitations as functions of the increments $\Delta\mathbf{u}$,

$$\mathbf{u}_{min} - \bar{\mathbf{u}}_k \le \Delta\mathbf{u}_k \le \mathbf{u}_{max} - \bar{\mathbf{u}}_k. \tag{28}$$

Also, the absolute values of $\Delta\mathbf{u}_k$ must be bounded to ensure that the linearization holds:

$$-\boldsymbol{\delta}\mathbf{u} \le \Delta\mathbf{u}_k \le \boldsymbol{\delta}\mathbf{u}, \tag{29}$$

See Section VI for the specific values of these constraints.

D. Objective Function

The goals of the path-following guidance law, in order of relevancy, are the following:

1) The airplane must follow a precomputed reference path (given as a concatenation of straight flight segments).
2) The guidance law must avoid excessive oscillations.
3) The airplane and reference trajectories should be synchronized in time, if possible.

If a standard quadratic cost penalizing the position error at each sampling time is used, we would minimize the difference between the trajectory and *virtual waypoints*. As seen in Fig. 2 (where $\mathbf{p}_k$ and $\mathbf{p}_{\mathrm{ref}\,k}$ denote, respectively, the airplane and the reference positions at the instant k), this approach might lead to an oscillatory trajectory.

To avoid this situation, an alternative formulation is proposed. First, following Fig. 3, the distance between the aircraft and the tracked reference *segment* is penalized.

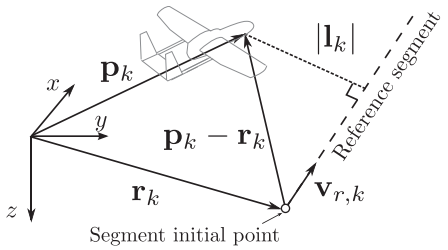

Fig. 3. Distance between the airplane's position and the reference flight segment.

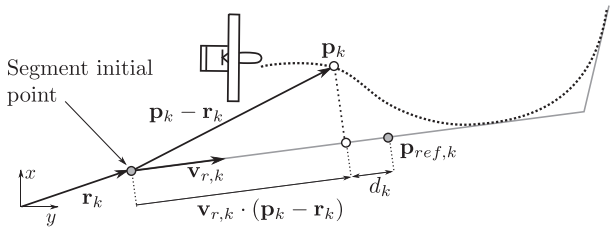

Fig. 4. Projection of the position on the reference segment.

Denoting by $\mathbf{r}_k = [r_{x,k}, r_{y,k}, r_{z,k}]^T$ the position of the starting point of the tracked flight segment at the time t_k and by $\mathbf{v}_{r,k} = [v_{x,k}, v_{y,k}, v_{z,k}]^T$ its unitary direction vector, the distance between the aircraft and the segment is

$$\begin{aligned}\mathbf{l}_k &= \mathbf{v}_{r,k} \wedge (\mathbf{p}_k - \mathbf{r}_k) \\ &= \begin{bmatrix} 0 & -v_{z,k} & v_{y,k} \\ v_{z,k} & 0 & -v_{x,k} \\ -v_{y,k} & v_{x,k} & 0 \end{bmatrix} (\mathbf{p}_k - \mathbf{r}_k). \end{aligned} \tag{30}$$

Using (30), we penalize the path-tracking error defining[2]

$$\begin{aligned} J_{1,k} = \sum_{i=1}^{N_p} \Big[&\left(\mathbf{V}_{k+i} \left(\mathbf{x}_{k+i|k} - \hat{\mathbf{r}}_{k+i}\right)\right)^T \mathbf{R}_{1,k+i} \\ &\times \left(\mathbf{V}_{k+i} \left(\mathbf{x}_{k+i|k} - \hat{\mathbf{r}}_{k+i}\right)\right)\Big], \end{aligned} \tag{31}$$

where

$$\mathbf{V}_i = \begin{bmatrix} 0 & -v_{z,i} & v_{y,i} & 0 \\ v_{z,i} & 0 & -v_{x,i} & 0 \\ -v_{y,i} & v_{x,i} & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \ \hat{\mathbf{r}}_k = \begin{bmatrix} r_{x,k} \\ r_{y,k} \\ r_{z,k} \\ 0 \end{bmatrix} \tag{32}$$

and $\mathbf{R}_{1,k+i}$ is a diagonal weight matrix.

However, during turns, minimizing (31) does not guarantee a turn in the correct direction (a segment flown in both directions results in the same cost). This is solved by including time synchronization. Consider the airplane's position error (respect to the reference points at each sampling time) projected on the reference segment. From Fig. 4, this error is

[2] One can find in the literature some related works, such as [30], which uses a cost function similar to (31), with reference heading angles to avoid ambiguity in turn direction. However, in that reference, the time synchronization problem is not addressed.

$$\begin{aligned} d_k &= \mathbf{v}_{r,k} \cdot (\mathbf{p}_k - \mathbf{r}_k) - \mathbf{v}_{r,k} \cdot \left(\mathbf{p}_{\mathbf{ref}k} - \mathbf{r}_k\right) \\ &= \mathbf{v}_{r,k} \cdot \left(\mathbf{p}_k - \mathbf{p}_{\mathbf{ref}k}\right). \end{aligned} \tag{33}$$

Then, introducing the square of this error, we define

$$\begin{aligned} J_{2,k} = \sum_{i=1}^{N_p} \Big[&\left(\mathbf{v}_{k+i} \left(\mathbf{x}_{k+i|k} - \mathbf{x}_{\mathbf{ref}k+i}\right)\right)^T R_{2,k+i} \\ &\times \left(\mathbf{v}_{k+i} \left(\mathbf{x}_{k+i|k} - \mathbf{x}_{\mathbf{ref}k+i}\right)\right)\Big], \end{aligned} \tag{34}$$

where $R_{2,k+i}$ is a scalar weight and $\mathbf{v}_k$ and $\mathbf{x}_{\mathbf{ref}\ k}$ are

$$\mathbf{v}_k = \begin{bmatrix} \mathrm{c}\left(\chi_{r,k}\right) \mathrm{c}\left(\gamma_{r,k}\right) & \mathrm{s}\left(\chi_{r,k}\right) \mathrm{c}\left(\gamma_{r,k}\right) & -\mathrm{s}\left(\gamma_{r,k}\right) & 0 \end{bmatrix}, \tag{35}$$

$$\mathbf{x}_{\mathbf{ref}k} = \begin{bmatrix} \mathbf{p}_{\mathbf{ref}k}^T & 0 \end{bmatrix}^T, \tag{36}$$

where $\chi_{r,k}$ and $\gamma_{r,k}$ are, respectively, the heading and the trajectory angle of the reference segment r at the instant k and c(·) and s(·) denote, respectively, the cosine and sine.

Now, to avoid excessive control usage and oscillations, define

$$\begin{aligned} J_{3,k} = &\sum_{i=1}^{N_p-1} (\mathbf{u}_{k+i} - \mathbf{u}_{k+i-1})^T \mathbf{Q}_{k+i} (\mathbf{u}_{k+i} - \mathbf{u}_{k+i-1}) \\ &+ (\mathbf{u}_k - \hat{\mathbf{u}}_k)^T \mathbf{Q}_k (\mathbf{u}_k - \hat{\mathbf{u}}_k), \end{aligned} \tag{37}$$

where $\hat{\mathbf{u}}_k$ represents the actual, real values of V_k, γ_k, and κ_k (regulated by a continuous-time low-level controller whose reference is $\mathbf{u}_k$). Then the first term weights the control signals increments along the prediction horizon, whereas the last term weights the increment of the reference control signal above its real value.

The cost function is set as the sum of the previous three, added from the present time interval from k up to the prediction horizon $k + N_p$:

$$J\left(\mathbf{x}_k, \mathbf{\Delta u_S}\right) = J_{1,k} + J_{2,k} + J_{3,k}. \tag{38}$$

The weight matrices in (38) are

$$\mathbf{Q}_i = k_Q \,\mathrm{diag}\left(\frac{1}{\delta V^2}, \frac{1}{\delta \gamma^2}, \frac{1}{\delta \kappa^2}\right), \tag{39}$$

$$\mathbf{R}_{1,i} = k_{R_1} \zeta_i \,\mathrm{diag}\,(1, 1, 1, 0), \tag{40}$$

$$R_{2,i} = k_{R_2} \zeta_i, \tag{41}$$

where δV, $\delta \gamma$, and $\delta \kappa$ are the ones used in (29) and are introduced to be able to compare the control variables. The scalar weights k_Q, k_{R_1}, and k_{R_2} are given in Section VI-C. Additionally, ζ_i is a function introduced to avoid penalizing errors during the first sampling times (allowing some transients), defined as

$$\zeta_i = \begin{cases} 0, & \text{If } i \leq 3, \\ 1, & \text{If } i \in \left[4, N_p\right]. \end{cases} \tag{42}$$

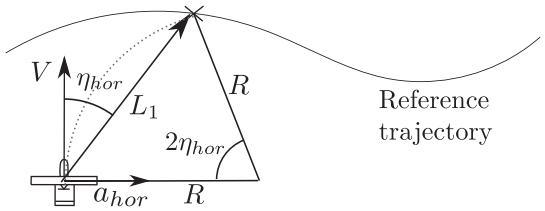

Fig. 5. Diagram for the L1 guidance law.

IV. INITIALIZATION PROBLEM: L_1 NAVIGATION LAW

The procedure outlined in Section III-A requires an initial control sequence for linearization. We obtain it from a nonlinear guidance law that we refer to as L_1 navigation law (L1). Introduced in [27], this guidance law has been widely used; see, for instance, [32, 33]. The law ensures the availability of feasible solutions with negligible computational burden.

Following [27], one introduces a virtual target (at certain distance from the aircraft, L_1) that follows the reference trajectory at the reference speed. L1 then computes accelerations, which are commanded to a low-level control system. Since the flight path angles are assumed small, this algorithm can be split in horizontal and vertical guidance laws. In this paper, we introduce a slight modification of L1 that provides commands in flight path angle (vertical channel) and bank angle (horizontal channel) and takes into account airplane physical limitations. Velocity is set at the same value as the virtual target.

A. Horizontal Guidance Law

Consider horizontal steady turns as depicted in Fig. 5. From [27], the horizontal acceleration computed by the guidance algorithm is

$$a_{hor} = \frac{2V^2 \sin \eta_{hor}}{L_1}.$$

For a steady turn, the airplane load factor is given by

$$n = \sqrt{\left(\frac{a_{hor}}{g}\right)^2 + 1} = \sqrt{\left(\frac{2V^2 \sin \eta_{hor}}{gL_1}\right)^2 + 1}, \quad (43)$$

where the relationship between turn radius R, the virtual target distance L_1, and the target angle η_{hor} can be obtained from Fig. 5. To get the bank angle, we use the following equation from flight mechanics:

$$\phi = \operatorname{sgn}(\sin \eta_{hor}) \arccos \frac{1}{n}, \quad (44)$$

where the sign operator is introduced for right- and left-turn disambiguation.

This guidance scheme lets us introduce bounds in the load factor ($n_{\max}$), avoiding adverse situations, such as stalls or structural problems; thus, if in (44) $n > n_{max}$, we just set $n = n_{max}$. Moreover, turnover maneuvers are also allowed since if $|\eta_{hor}| \geq \frac{\pi}{2}$ is demanded, the guidance law commands a maximum load factor turn. These maneuvers would not be easy to capture with linearization given their extreme nonlinear nature.

B. Vertical Guidance Law

Following a similar approach to L1, the vertical acceleration that tracks the virtual target is given by

$$a_{ver} = N_{ver} \frac{V^2}{L_1} \sin \eta_{ver}, \quad (45)$$

where N_{ver} is a gain and η_{ver} is the vertical target angle, defined as η_{hor}. Thus,

$$\Delta\gamma \approx \frac{a_{ver}}{V} T_s = N_{ver} \frac{V}{L_1} \sin \eta_{ver} T_s. \quad (46)$$

V. GUIDANCE ALGORITHM

In this section, we present an algorithm to solve the nonlinear optimization problem for the MPC stated in Section V. An iterative optimization process is used so that in each step, a linearized MPC problem around a feasible control sequence is solved; this solution is subsequently refined in the next iteration step. This iterative process is "hotstarted" by the robust L1 law of Section IV. Next, the algorithm is described in detail.

1) *Initial control sequence:*
   a) *L_1 navigation law ("hotstart"):* If there is no feasible solution available from the previous time step, the L1 method described in Section IV is applied sequentially along the prediction horizon N_p to find a control sequence along the prediction horizon:
      i) Using the state measurement at the current sampling time (denoted by $\mathbf{x}_k$), V_k, γ_k and ϕ_k are computed as in Section IV.
      ii) The state estimation at the next step ($\mathbf{x}_{k+1}$) is computed using (6)–(9).
      iii) Repeat until the controls $\mathbf{u}_{\mathrm{k}}, \ldots, \mathbf{u}_{\mathrm{k+N_p-1}}$ are computed.
   b) *Predictive guidance ("refinement"):* If there exists an optimized guidance solution from the previous time step, that control sequence (removing the first component) is used as a starting point. The last component is computed using the L1 method.
2) *Linearization:* Linearize around the initial control sequence using the prediction model (19).
3) *Optimization:* Based on Section V, we define the following optimization problem:

$$\begin{aligned} \min_{\mathbf{\Delta u_S}} \quad & J(\mathbf{x}_k, \mathbf{\Delta u_S}) \\ \text{s.t.} \quad & \mathbf{\Delta u}_k \geq \max(\mathbf{u}_{min} - \bar{\mathbf{u}}_k, \; -\boldsymbol{\delta}\mathbf{u}), \\ & \mathbf{\Delta u}_k \leq \min(\mathbf{u}_{max} - \bar{\mathbf{u}}_k, \, \boldsymbol{\delta}\mathbf{u}). \end{aligned} \quad (47)$$

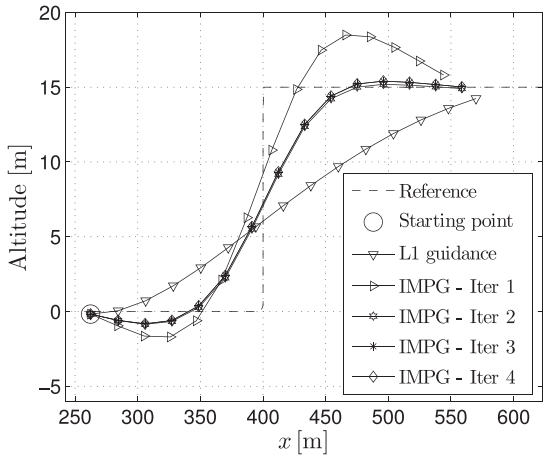

Fig. 6. Vertical profile of successive trajectories computed along the iterations.

TABLE I
Evolution of the Cost Function Along the Iterations

	L1	Iteration 1	Iteration 2	Iteration 3	Iteration 4
Cost ($\times$ 10^2)	14.1059	8.9940	3.2152	3.1888	3.1888
Time (s)	0.04	0.12	0.16	0.19	0.21

If (47) is solved in the allowable computation time, a new optimized control sequence is obtained: $\mathbf{u_S} = \bar{\mathbf{u}}_\mathbf{S} + \boldsymbol{\Delta}\mathbf{u_S}$. If (47) fails or time reaches its limit, the last nominal control sequence is used, that is, $\mathbf{u_S} = \bar{\mathbf{u}}_\mathbf{S}$.

4) *Nominal sequence updating and iteration:* If there is time, the optimized control sequence is considered as the new nominal control sequence ($\bar{\mathbf{u}}_\mathbf{S} = \mathbf{u_S}$), and steps 2–3 are repeated; this procedure is further iterated until the control solution converges (the cost function does not improve significantly) or time is up.

To illustrate convergence of the iterative algorithm, Fig. 6 shows several vertical profiles computed along successive iterations (with parameters given in Section VI). Each iteration clearly refines the guidance solution, improving tracking. Table I shows the cost function during the iterations and the cumulated computation time at the end of each iteration. A local minimum was reached, halting the iterative algorithm.

VI. SIMULATION RESULTS

We next present a simulation study showing the performance of iterative model predictive guidance (referred to as IMPG in this section). First, we test the algorithm for an example plane mission to compare it with other guidance algorithms (Section VI-A). We follow with a more in-depth study for a 3D reconnaissance mission (Section VI-B). We finish with a parametric study in Section VI-C.

Some basic parameters of the simulation were $T_s = 1$ s, $T_\delta = T_s/10 = 0.1$ s, $V_{min} = 15$ m/s, $V_{max} = 30$ m/s, $\gamma_{min} = -15^\circ$, $\gamma_{max} = 15^\circ$, $\kappa_{min} = -0.6540 T_s$ rad, $\kappa_{max} = 0.6540 T_s$ rad, $N_{ver} = 1.5$, $L_1 = 150$ m, $\lambda = 0.23$. Other parameters are discussed in Section VI-C. The simulation test bed is a nonlinear model of a light aircraft (Cefiro UAV) designed and built at the University of Seville (Spain). Cefiro is a 23-kg airplane designed with a cruise speed of 20 m/s. The model uses the general 6-DoF equations of motion (see, e.g., [34]), together with a nonlinear aerodynamic and propulsive model. Additionally, there is a low-level attitude controller that makes the airplane follow references in airspeed, flight path angle, and bank angle, maintaining zero sideslip. Further details regarding the attitude controller, the geometry of Cefiro, and its aerodynamic and propulsive models are given in [28, 35, 37]. The reference trajectories used in the simulations have been obtained using an optimal trajectory generator [37].

In what follows, to compare results, we define several metrics. First, the mean mission control effort is quantified as

$$CE = \frac{1}{N_{CE}} \sum_k \left[(\mathbf{u}_k - \hat{\mathbf{u}}_k)^T \bar{\mathbf{Q}}_k (\mathbf{u}_k - \hat{\mathbf{u}}_k) \right]^{\frac{1}{2}}, \tag{48}$$

using a normalized weight matrix $\bar{\mathbf{Q}}_k$, defined as in (39), with $k_Q = 1$ and a number of samples N_{CE} (defined as the mission time span divided by the IMPG sampling time). This nondimensional definition takes into account the heterogeneous nature of the control signals.

To measure mission tracking performance, we might use the L_2 norm of the oversampled tracking error at all times:

$$TE = \frac{1}{N_{TE}} \sum_{t \in S} \|\mathbf{p}(t) - \mathbf{p_{ref}}(t)\|_2, \tag{49}$$

where S is a set containing the oversampled times (we have chosen an oversampling frequency 10 times faster than the one used in the IMPG algorithm), and N_{TE} is again the number of samples (in this case, the mission time span divided by the oversampling time). This metric, however, is not fair to guidance laws without time synchronization capabilities. Thus, for comparison with such laws, we define a metric based on (30) considering only deviations with respect to the reference flight segments:

$$PE = \frac{1}{N_{TE}} \sum_{t \in S} \|\mathbf{l}(t)\|_2, \tag{50}$$

It is important to remark that these metrics are useful when comparing different guidance algorithms performing the same mission, but they should not be understood as a general performance index of a guidance method in any scenario since the achieved values will depend noticeably on the number of turns in the mission (which are the maneuvers in which more error and control usage is accumulated).

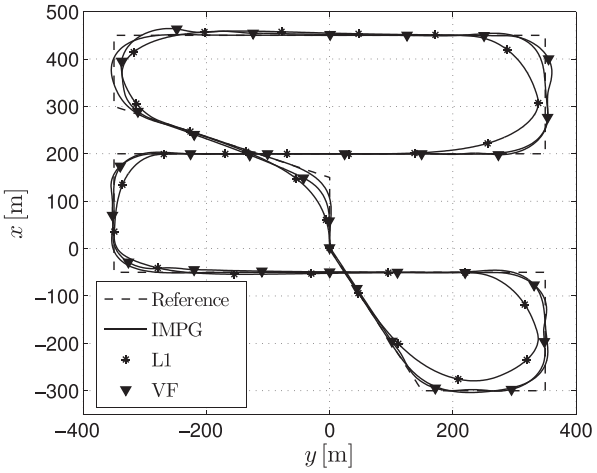

Fig. 7. Comparison between different guidance strategies.

TABLE II
Comparison Between IMPG, VF, and L1 for a Plane Mission

Algorithm	CE	PE (m)
L1	0.2951	9.9512
VF	3.4802	4.4065
IMPG	0.1479	2.8874

A. Example Plane Mission

We compare, in the absence of wind and for an example 2D mission, several guidance laws. The objective of this comparative analysis is to understand the path-following capabilities of three rather different algorithms: IMPG, which is a predictive law; L1 (as described in Section IV), which is LOS based; and vector field (VF; see [38]), which modifies the system dynamics to makes the desired path attractive).

The implemented VF guidance law is based on [38] and (12), with the tuning parameters $k = 0.04$, $\xi^{\infty} = 1.3$ rad, $\varepsilon = 1$ rad, and $\kappa = 0.1680$. Additionally, a low-level PID heading controller for VF has been implemented, with a typical turning speed $\alpha = 0.28$ rad/s. To make VF less reactive and able to anticipate corners, the next reference flight segment is used when the cross-track error with respect to it is less than 120 m.

Fig. 7 shows the resulting trajectories from using the three different guidance algorithms. Compared with VF and L1, IMPG shows excellent tracking capabilities. L1 is perhaps too anticipative because it reacts as soon as the point being tracked takes a turn. On the other hand, VF is more precise in the path following but less efficient in the control usage. The results of Table II support these intuitive conclusions. For fairness, it must be mentioned that both L1 and VF have negligible computational costs and have proven mathematical properties, such as stability.

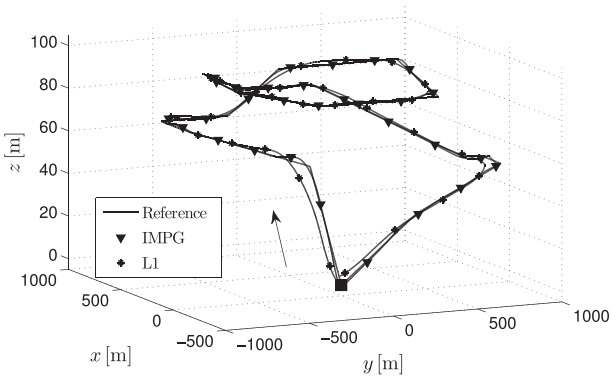

Fig. 8. Example exploration mission (3D view).

B. Example 3D Exploration Mission with Moderate Crosswind

Next, we perform a more in-depth study in which the guidance system is required to follow a 3D reference trajectory (a reconnaissance mission around a square area), including time synchronization and wind disturbances. Only IMPG and L1 are considered. Computational times are given at the end.

The random wind in the simulation was modeled as

$$w_x = \bar{V}_w \cos\bar{\xi}_w + \delta w_x, \tag{51}$$

$$w_y = \bar{V}_w \sin\bar{\xi}_w + \delta w_y, \tag{52}$$

$$w_z = \delta w_z, \tag{53}$$

where $\bar{V}_w$ and $\bar{\xi}_w$ are the mean wind speed and direction and δw_i are stationary Gaussian random processes satisfying

$$\delta\dot{w}_i = \begin{cases} 0, & \text{if } |\delta w_i| \geq \delta w_{max} \\ \nu(\sigma), & \text{else.} \end{cases} \tag{54}$$

In (54), $\nu(\sigma)$ is Gaussian white noise with standard deviation σ, and a maximum deviation from the mean has been imposed to avoid excessive (and unrealistic) wind excursions from its nominal value. Chosen values were $V_w = 4$ m/s, $\xi_w = 210^{\circ}$, $\sigma = 10^{-2}$m/s, and $\delta w_{max} = 3$ m/s.

Fig. 8 depicts the 3D view of the reference and the path followed by the airplane. L1 keeps the airplane close to the reference path even in presence of wind but with increasing errors at turns and climbs/descents. On the other hand, IMPG performs better than L1, accomplishing tight turns at corners and without appreciable error in the steady flight segments despite the wind, thanks to the online disturbance estimator.

Fig. 9 depicts the wind components during the mission as well as its estimation by the IMPG law, which closely follow the real wind. Notice that this ability to estimate wind could be a useful feature to obtain approximate wind maps with small UAVs.

Figs. 10 and 11 show the evolution of the airplane position and the control signals for both guidance methods

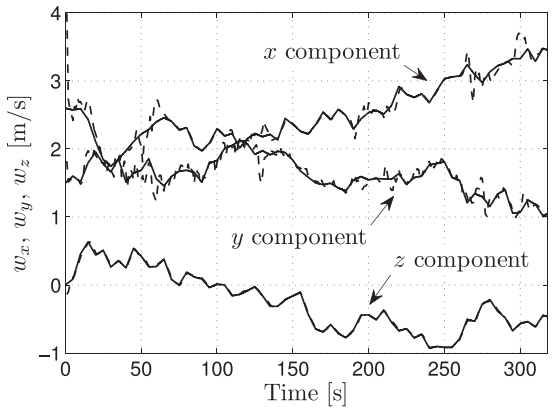

Fig. 9. Time evolution of the *x*, *y*, and *z* components of the wind speed (solid lines) and the estimated disturbances (dashed lines).

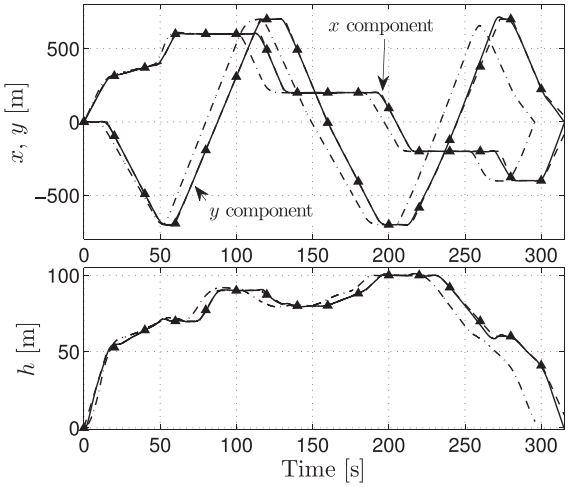

Fig. 10. Time evolution of the airplane position. Solid line: IMPG; dashed-dotted: L1. Dashed line with bullets denotes the reference.

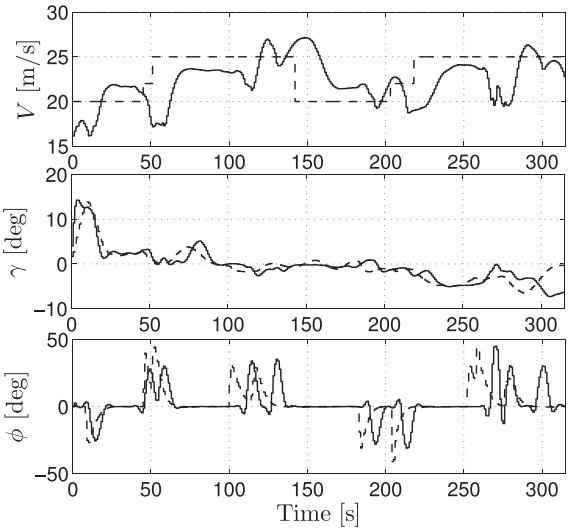

Fig. 11. Time evolution of the control signals computed by the IMPG (solid line) and the L1 (dashed line) algorithms.

TABLE III
Comparison Between IMPG and L1 for a 3D Mission

Algorithm	CE	PE (m)	Algorithm	CE	PE (m)
L1	0.1579	6.0949	IMPG	0.1567	1.9466

with respect to the reference trajectory. While L1 accumulates appreciable delays during the mission (as time does not enter explicitly in its formulation), IMPG is able to synchronize with the reference, compensating time gaps at turns, and reaching the end point with less than 1 s of delay.

The simulation was run on an Intel i7 2.2 GHz CPU. The algorithms were implemented in Matlab (without code compilation). The iterative algorithm was stopped when the variation of the cost function was less than one (typical values of the cost function are about 10^3); with this bound, convergence was quick, with a mean number of iterations of 2.9373 and a maximum number of six. Moreover, the mean time to run one iteration was 0.0579 s (using Matlab's quadprog, with 42 states), while the computation time of L1 was found negligible. During the overall mission, the maximum amount of time to complete the iterative IMPG algorithm was 0.3750 s (with five iterations). Compared with the sampling time, the computational burden is acceptable (37.5%) and could be further reduced by using a compiled programming language and/or a faster optimizer. Thus, real-time execution of the algorithm is doable, and therefore its onboard implementation is feasible.

To further compare IMPG with L1, we show in Table III the control usage and trajectory tracking performance of both methods using metrics (48) and (50). IMPG has a 1% of control effort saving with respect to the classical L1 law and a much better path-following performance and, in addition, time synchronization capabilities.

C. Parametric Study

There are several design parameters in the IMPG formulation (see Section III). To get more insight into the impact of these parameters in the IMPG performance, a parametric study is carried out. Thus, a batch of simulations are performed for a short mission similar to the one presented in Section VI-B.

Fig. 12 shows the effect of N_p (the prediction horizon) on the final total mission cost (computed from (38) but using the real state and the real applied controls), the mean computation time for a single iteration of IMPG, the mean total time to solve the optimization at each sampling time, and the maximum time for an iteration during the whole mission. It can be seen that mission cost reaches a minimum at $N_p = 14$ (selected value). This behavior is explained since the greater N_p is, the more future information about the reference trajectory can be used. However, large N_p values degrade the guidance effectiveness since the algorithm is forced to optimize a

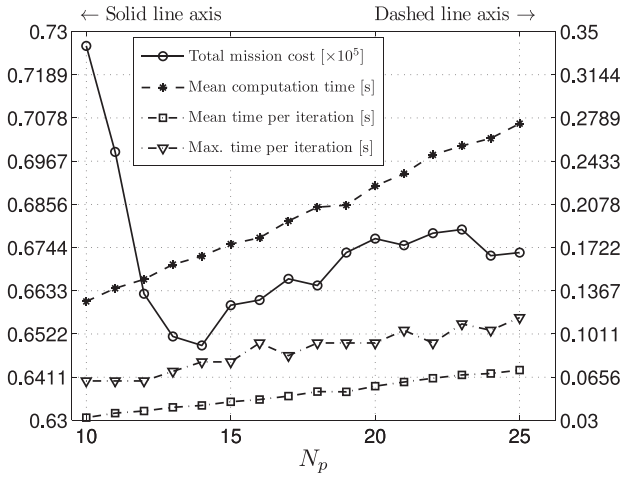

Fig. 12. Influence of N_p in the mission cost and computational times.

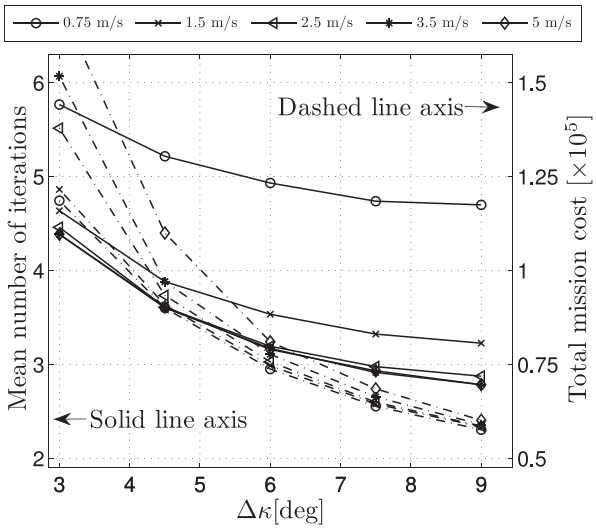

Fig. 13. Mean number of iterations and total mission cost as function of $\Delta\kappa$ with ΔV as a parameter (see the legend).

rather long future trajectory, diluting the effect of the first control signal (the one effectively applied). On the other hand, the computational burden increases monotonically with N_p as each increment implies three new state variables in the optimization. At the selected horizon ($N_p = 14$), the computation times are affordable, with a mean value of 0.1733 s, far below the employed sampling time ($T_s = 1$ s). Even in the worst-case scenario, the maximum time per iteration is still small (83.3 ms) so that at least one iteration can be computed before stopping the iterative process to avoid overruns. Finally, the mean number of iterations (not shown in the figure) has been found to be quite insensible to N_p, having values near to 3.2.

Next, the maximum increment of the control signals in (29), ΔV, $\Delta\gamma$, and $\Delta\kappa$ (which ensure the validity of the linearized model), are studied. Figs. 13 and 14 show the mean number of iterations and the total mission cost as functions of these parameters. Fig. 13 shows that both $\Delta\kappa$

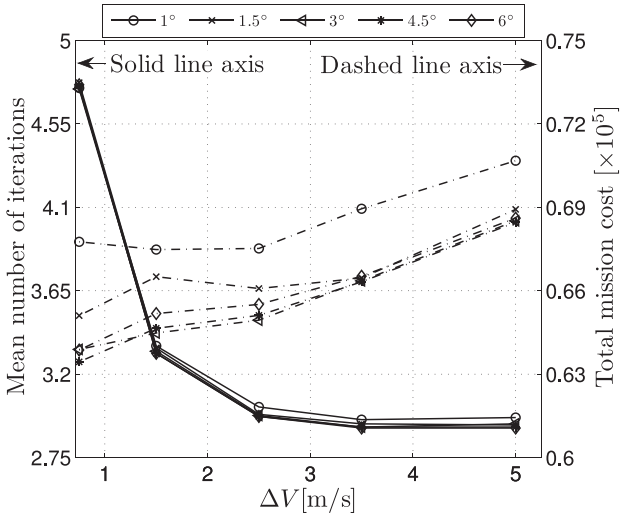

Fig. 14. Mean number of iterations and total mission cost as function of the parameters and ΔV and $\Delta\gamma$.

and ΔV have a strong influence in the algorithm performance. Higher values of $\Delta\kappa$ lead to less cost and iterations, but the curves flatten for $\Delta\kappa > 7°$. Large values of $\Delta\kappa$ might lead to less accurate predictions, which could subsequently have a negative impact in integrity. Thus, a value of $\Delta\kappa = 7.5°$ was selected.

With respect to ΔV, both figures show that values below 2 m/s raise considerably the mean number of iterations and reduce the mission cost (and vice versa). Thus, a trade-off value $\Delta V = 2.5$ m/s was chosen.

Fig. 13 shows that $\Delta\gamma$ has a weak impact in both performance indicators, with curves for values greater than 1.5° almost overlapping. This is due to changes in the reference flight path angle being relatively small compared to changes in heading angles or velocity. Thus, $\Delta\gamma = 3°$ was selected.

Finally, the weights in the cost function $\mathbf{R_{1}}_i$, $\mathbf{R_{2}}_i$, and $\mathbf{Q}_i$, used, respectively, in (31), (34), and (37), are studied. Following [15], a Pareto analysis is performed to have a comprehensive view of the influence of those weights in the achievement of the problem's objectives (path following, time synchronization, and control saving), quantified using the metrics (48) and (49) for the control effort and the tracking error, respectively. A batch of mission simulations has been performed varying the weight matrices. They have been kept diagonal (following the structure of (39–41)). Only the scalar weights k_Q and k_{R2} are varied since only the ratio between these two parameters and k_{R1} (set to 10) matters. Thus, the following sets have been considered:

$$k_Q \in [0.05, 0.1, 0.5, 1, 5, 10, 30, 60, 100], \tag{55}$$

$$k_{R2} \in [0.01, 0.05, 0.1, 0.5, 1, 5, 10, 50, 100]. \tag{56}$$

Fig. 15 (left) shows the metrics for each combination of values of k_Q and k_{R2}. Fig. 15 (right) shows a zoom close to the origin (low tracking error and control effort). A

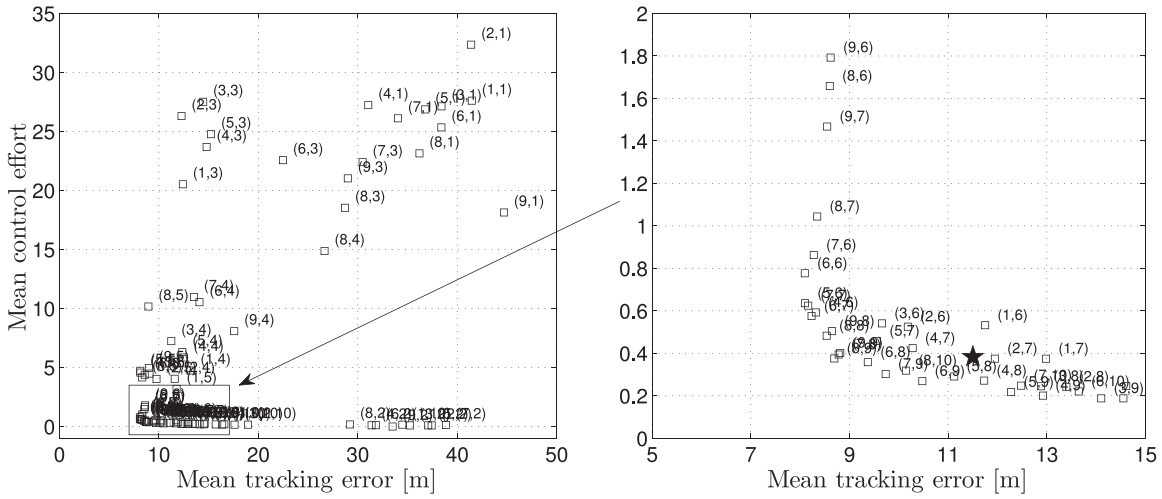

Fig. 15. Pareto analysis of the influence of the weights in the IMPG performance. Control effort and tracking error are defined in (48) and (49). Text annotations define, for each point, which component of (55) and (56) is selected for k_Q and k_{R2} (in that order). The star is the selected point.

Pareto front emerges, representing the best trade-off between the objectives. The weight combinations $k_Q = 30$ and $k_{R_2} = 0.1$ (depicted by a star) have been selected.

VII. CONCLUSIONS

In this paper, a new MPC path-following guidance algorithm has been introduced to follow a precomputed reference trajectory. The main features of the algorithm are the use of the bank angle instead of the heading angle, an explicit linearization scheme to solve nonlinear optimization, disturbance estimation, and the use of a modified robust L_1 navigation law as hotstart and backup, thus guaranteeing feasibility. Simulations show good path-seeking performance and time synchronization capabilities, even in the presence of moderate wind. The iterative optimization process converges quickly, making it suitable for onboard implementation in UAVs. Finally, in contrast to the most common guidance algorithms (based on waypoints), this guidance law uses a continuous reference trajectory with passage times, which can be preoptimized for an efficient mission accomplishment.

Undergoing research is focused on the implementation of the guidance law in the real airplane for flight experiments and the extension of this algorithm to formation flying. In addition, the use of robust model predictive schemes [31] to (mathematically) guarantee good performance and constraint satisfaction even in the presence of wind disturbances will be analyzed. These schemes would require employing a more advanced disturbance estimator (e.g., an extended or unscented Kalman filter).

APPENDIX. LINEAR APPROXIMATION FOR STATE PREDICTION

This appendix shows the explicit linearization of $\mathbf{f}$ used in Section III-A. Starting from the definition of $\mathbf{f}$ in (13) and the linearized prediction model (17), one can write each column of the Jacobian (57) as

$$\frac{\partial \mathbf{f}_i}{\partial \mathbf{u_S}}(\bar{\mathbf{u}}_\mathbf{S}, \chi_0) = \left[\mathbf{\Theta}_{4\times 2}\frac{\partial \mathbf{f}_i}{\partial \kappa_0}|\mathbf{\Theta}_{4\times 2}\frac{\partial \mathbf{f}_i}{\partial \kappa_1}|\cdots|\frac{\partial \mathbf{f}_i}{\partial V_i}\frac{\partial \mathbf{f}_i}{\partial \gamma_i}\frac{\partial \mathbf{f}_i}{\partial \kappa_i}|\mathbf{\Theta}_{4\times(N_p-1-i)}\right]. \tag{57}$$

$$\frac{\partial \mathbf{f}_k}{\partial V_k} = \begin{bmatrix} 2\frac{\mathrm{c}_{\bar{\gamma}_k}T_s}{\bar{\kappa}_k}\cos\left(\frac{\bar{\kappa}_k}{2}+\sum_{i=0}^{k-1}\bar{\kappa}_i+\chi_0\right)\mathrm{s}_{\frac{\bar{\kappa}_k}{2}} \\ 2\frac{\mathrm{c}_{\bar{\gamma}_k}T_s}{\bar{\kappa}_k}\sin\left(\frac{\bar{\kappa}_k}{2}+\sum_{i=0}^{k-1}\bar{\kappa}_i+\chi_0\right)\mathrm{s}_{\frac{\bar{\kappa}_k}{2}} \\ -\mathrm{s}_{\gamma_k}T_s \\ 0 \end{bmatrix}, \tag{58}$$

$$\frac{\partial \mathbf{f}_k}{\partial \gamma_k} = \begin{bmatrix} -2\frac{\bar{V}_k\mathrm{s}_{\bar{\gamma}_k}T_s}{\bar{\kappa}_k}\cos\left(\frac{\bar{\kappa}_k}{2}+\sum_{i=0}^{k-1}\bar{\kappa}_i+\chi_0\right)\mathrm{s}_{\frac{\bar{\kappa}_k}{2}} \\ -2\frac{\bar{V}_k\mathrm{s}_{\bar{\gamma}_k}T_s}{\bar{\kappa}_k}\sin\left(\frac{\bar{\kappa}_k}{2}+\sum_{i=0}^{k-1}\bar{\kappa}_i+\chi_0\right)\mathrm{s}_{\frac{\bar{\kappa}_k}{2}} \\ -V_k\mathrm{c}_{\bar{\gamma}_k}T_s \\ 0 \end{bmatrix}, \tag{59}$$

$$\frac{\partial \mathbf{f}_k}{\partial \kappa_k} = \begin{bmatrix} \bar{V}_k\mathrm{c}_{\bar{\gamma}_k}T_s\left(\frac{\bar{\kappa}_k\mathrm{c}_{(\bar{\kappa}_k+\bar{\chi}_k)}-\mathrm{s}_{(\bar{\kappa}_k+\bar{\chi}_k)}+\mathrm{s}_{\bar{\chi}_k}}{\bar{\kappa}_k^2}\right) \\ \bar{V}_k\mathrm{c}_{\bar{\gamma}_k}T_s\left(\frac{\bar{\kappa}_k\mathrm{s}_{(\bar{\kappa}_k+\bar{\chi}_k)}+\mathrm{c}_{(\bar{\kappa}_k+\bar{\chi}_k)}-\mathrm{c}_{\bar{\chi}_k}}{\bar{\kappa}_k^2}\right) \\ 0 \\ 1 \end{bmatrix}, \tag{60}$$

$$\frac{\partial \mathbf{f}_k}{\partial \kappa_i} = \begin{bmatrix} -2\frac{\bar{V}_k\mathrm{c}_{\bar{\gamma}_k}T_s}{\bar{\kappa}_k}\sin\left(\frac{\bar{\kappa}_k}{2}+\bar{\chi}_k\right)\mathrm{s}_{\frac{\bar{\kappa}_k}{2}} \\ 2\frac{\bar{V}_k\mathrm{c}_{\bar{\gamma}_k}T_s}{\kappa_k}\cos\left(\frac{\bar{\kappa}_k}{2}+\bar{\chi}_k\right)\mathrm{s}_{\frac{\bar{\kappa}_k}{2}} \\ 0 \\ 0 \end{bmatrix}, \quad i<k, \tag{61}$$

where $\bar{\chi}_k = \sum_{i=0}^{i=k-1} \bar{\kappa}_i + \chi_0$ and $s_{(\cdot)}$ and $c_{(\cdot)}$, respectively, denote sine and cosine.

Note that (58–61) may present numerical problems at straight flight segment since then $\bar{\kappa}_k$ (which appears in some denominators) is zero. This is, however, a solvable singularity; approximating f_k and (58–61) when $\bar{\kappa}_k \to 0$, one gets

$$\mathbf{f}_k \approx \left(\lim_{\bar{\kappa}_k \to 0} \mathbf{f}_k\right) + \left(\lim_{\bar{\kappa} \to 0} \frac{\partial \mathbf{f}_k}{\partial \bar{\kappa}_k}\right) \bar{\kappa}_k, \tag{62}$$

$$\frac{\partial \mathbf{f}_k}{\partial (\cdot)} \approx \left(\lim_{\bar{\kappa}_k \to 0} \frac{\partial \mathbf{f}_k}{\partial (\cdot)}\right) + \left(\lim_{\bar{\kappa}_k \to 0} \frac{\partial^2 \mathbf{f}_k}{\partial \bar{\kappa}_k \partial (\cdot)}\right) \bar{\kappa}_k. \tag{63}$$

Computing the limits appearing in (62) and (63), one has, when $\kappa_k \to 0$, the following expressions (the subindex k is omitted for clarity):

$$\mathbf{f} \to \begin{bmatrix} \bar{V} c_{\bar{\gamma}} T_s c_{\bar{\chi}} \\ \bar{V} c_{\bar{\gamma}} T_s s_{\bar{\chi}} \\ -\bar{V} s_{\bar{\gamma}} T_s \\ 0 \end{bmatrix}, \frac{\partial \mathbf{f}}{\partial \kappa} \to \begin{bmatrix} -\bar{V} c_{\bar{\gamma}} T_s \frac{s_{\bar{\chi}}}{2} \\ \bar{V} c_{\bar{\gamma}} T_s \frac{c_{\bar{\chi}}}{2} \\ 0 \\ 1 \end{bmatrix},$$

$$\frac{\partial \mathbf{f}}{\partial V} \to \begin{bmatrix} c_{\gamma} T_s c_{\bar{\chi}} \\ c_{\gamma} T_s s_{\bar{\chi}} \\ -s_{\bar{\gamma}} T_s \\ 0 \end{bmatrix}, \frac{\partial \mathbf{f}}{\partial \gamma} \to \begin{bmatrix} -\bar{V} s_{\bar{\gamma}} T_s c_{\bar{\chi}} \\ -\bar{V} s_{\bar{\gamma}} T_s s_{\bar{\chi}} \\ -\bar{V} c_{\bar{\gamma}} T_s \\ 0 \end{bmatrix},$$

$$\frac{\partial \mathbf{f}}{\partial \kappa_i} \to \begin{bmatrix} -\bar{V} c_{\gamma} T_s s_{\bar{\chi}} \\ \bar{V} c_{\gamma} T_s c_{\bar{\chi}} \\ 0 \\ 0 \end{bmatrix}, \frac{\partial^2 \mathbf{f}}{\partial \kappa \partial V} \to \begin{bmatrix} -c_{\bar{\gamma}} T_s \frac{s_{\bar{\chi}}}{2} \\ c_{\bar{\gamma}} T_s \frac{c_{\bar{\chi}}}{2} \\ 0 \\ 0 \end{bmatrix},$$

$$\frac{\partial^2 \mathbf{f}}{\partial \kappa \partial \gamma} \to \begin{bmatrix} \bar{V} s_{\gamma} T_s \frac{s_{\chi}}{2} \\ -\bar{V} s_{\gamma} T_s \frac{c_{\chi}}{2} \\ 0 \\ 0 \end{bmatrix}, \frac{\partial^2 \mathbf{f}}{\partial \kappa \partial \kappa_i} \to \begin{bmatrix} \bar{V} c_{\gamma} T_s \frac{c_{\chi}}{2} \\ -\bar{V} c_{\gamma} T_s \frac{s_{\chi}}{2} \\ 0 \\ 0 \end{bmatrix},$$

$$\frac{\partial^2 \mathbf{f}}{\partial \kappa^2} \to \begin{bmatrix} -\bar{V} c_{\gamma} T_s \frac{c_{\chi}}{3} \\ -\bar{V} c_{\gamma} T_s \frac{s_{\chi}}{3} \\ 0 \\ 0 \end{bmatrix}.$$

REFERENCES

[1] Peng, K., Cai, G., Chen, B., Dong, M., Lum, K., and Lee, T.
Design and implementation of an autonomous flight control law for a UAV helicopter.
Automatica, **45** (2009), 2333–2338.

[2] Voelker, A., Dördelmann, M., Kouramas, K., Pistikopoulos, E. N., Shanmugavel, M., Tsourdos, A., White, B., and Zbikowski, R.
Three-layer aircraft control: Path-planning, guidance, model predictive controller as autopilot.
In *18th IFAC Symposium on Automatic Control in Aerospace*, Nara, Japan, 2010.

[3] Adler, F.
Missile guidance by three-dimensional proportional navigation.
Journal of Applied Physics, **27**, 5 (1956), 500–507.

[4] Blakelock, J. H.
Automatic Control of Aircraft and Missiles (2nd ed.). New York, NY: Wiley, 1991.

[5] Mahapatra, P., and Shukla, U.
Accurate solution of proportional navigation for maneuvering targets.
IEEE Transactions on Aerospace and Electronic Systems, **25**, 1 (1989), 81–89.

[6] Yamasaki, T., Sakaida, H., Enomoto, K., Takano, H., and Baba, Y.
Robust trajectory-tracking method for UAV guidance using proportional navigation.
In *Proceedings of the International Conference on Control, Automation and Systems 2007*, Seoul, Korea, 2007.

[7] Fradkov, A., and Andrievsky, B.
Combined adaptive controller for UAV guidance.
European Journal of Control, **11**, 1 (2005), 71–81.

[8] Rysdyk, R.
Unmanned aerial vehicle path following for target observation in wind.
Journal of Guidance, Control, and Dynamics, **29**, 5 (2006), 1092–1100.

[9] Kaminer, I., Pascoal, A., Hallberg, E., and Silvestre, C.
Trajectory tracking for autonomous vehicles: An integrated approach to guidance and control.
Journal of Guidance, Control, and Dynamics, **21**, 1 (1998), 29–38.

[10] Shneydor, N.
Missile Guidance and Pursuit. Cambridge, UK: Horwood Publishing, 1998.

[11] McGee, T., and Hedrick, J.
Path planning and control for multiple point surveillance by an unmanned aircraft in wind.
In *Proceedings of the American Control Conference*, Minneapolis, MN, 2006, 4261–4266.

[12] Ren, W., and Beard, R.
Trajectory tracking for unmanned air vehicles with velocity and heading rate constraints.
IEEE Transactions on Control Systems Technology, **12**, 5 (2004), 706–716.

[13] Chen, H., Chang, K., and Agate, C.
UAV path planning with tangent-plus-Lyapunov vector field guidance and obstacle avoidance.
IEEE Transactions on Aerospace and Electronic Systems, **49**, 2 (2013), 840–856.

[14] Binetti, P., Ariyur, K., Krstić, M., and Bernelli, F.
Formation flight optimization using extremum seeking feedback.
Journal of Guidance, Control, and Dynamics, **26**, 1 (2003), 132–142.

[15] Sujit, P., Saripalli, S., and Sousa, J.
Unmanned aerial vehicle path following: A survey and analysis of algorithms for fixed-wing unmanned aerial vehicles.
IEEE Control Systems Magazine, **34**, 1 (2014), 42–59.

[16] Camacho, E. F., and Bordons, C.
Model Predictive Control (2nd ed.). New York, NY: Springer-Verlag, 2004, pp. 131–205.

[17] Kim, S., Oh, H., and Tsourdos, A.
Nonlinear model predictive coordinated standoff tracking of a moving ground vehicle.
Journal of Guidance, Control, and Dynamics, **36**, 2 (2013), 557–566.

[18] Kuwata, Y., and How, J., Sr.

Cooperative distributed robust trajectory optimization using receding horizon MILP.
IEEE Transactions on Control Systems Technology, **19**, 2 (2011), 423–431.

[19] Richards, A., and How, J.
"Model predictive control of vehicle maneuvers with guaranteed completion time and robust feasibility."
Proceedings of the American Control Conference, **5** (2003), 4034–4040.

[20] Keviczky, T., and Balas, G.
Software-enabled receding horizon control for autonomous unmanned aerial vehicle guidance.
Journal of Guidance, Control, and Dynamics, **29**, 3 (2006), 680–694.

[21] Izadi, H., Gordon, B., and Zhang, Y.
Hierarchical decentralized receding horizon control of multiple vehicles with communication failures.
IEEE Transactions on Aerospace and Electronic Systems, **49**, 2 (2013), 744–759.

[22] Riehl, J., Collins, G., and Hespanha, J.
Cooperative search by UAV teams: A model predictive approach using dynamic graphs.
IEEE Transactions on Aerospace and Electronic Systems, **47**, 4 (2011), 2637–2656.

[23] Belanger, J., Desbiens, A., and Gagnon, E.
UAV guidance with control of arrival time.
In *Proceedings of the 2007 American Control Conference*, New York, NY, 2007, 4488–4493.

[24] Prodan, I., Olaru, S., Bencatel, R., Borges de Sousa, J., Stoica, C., and Niculescu, S.-I.
Receding horizon flight control for trajectory tracking of autonomous aerial vehicles.
Control Engineering Practice, **21**, 10 (2013), 1334–1349.

[25] Guglieri, G., Quagliotti, F., and Speciale, G.
Optimal trajectory tracking for an autonomous UAV.
Automatic Control in Aerospace Journal, **1** (2011), 1–9.

[26] Vazquez, R., Gavilan, F., and Camacho, E. F.
Trajectory planning for spacecraft rendezvous with on/off thrusters.
In *IFAC World Congress*, Milan, Italy, 2011, 8473–8478.

[27] Park, S., Deyst, J., and How, J.
Performance and Lyapunov stability of a nonlinear path-following guidance method.
Journal of Guidance, Control, and Dynamics, **30**, 6 (2007), 1718–1728.

[28] Gavilan, F.
Sistemas de control y guiado para vehículos aéreos no tripulados: Diseño de algoritmos y sistemas embarcados.
Ph.D. dissertation, University of Seville, 2012.
http://aero.us.es/rvazquez/FGJ_thesis.pdf.

[29] Hull, D. G.
Fundamentals of Airplane Flight Mechanics. New York, NY: Springer-Verlag, 2007.

[30] Kang, Y., and Hedrick, J.
Linear tracking for a fixed-wing UAV using nonlinear model predictive control.
IEEE Transactions on Control Systems Technology, **17**, 5 (2009), 1202–1210.

[31] Gavilan, F., Vazquez, R., and Camacho, E.
Chance-constrained model predictive control for spacecraft rendezvous with disturbance estimation.
Control Engineering Practice, **20**, 2 (2012), 111–122.

[32] Ducard, G., and Geering, H. P.
"A computationally efficient guidance system for a small UAV."
In *Proceedings of the 4th International Conference on Informatics in Control, Automation, and Robotics*, J. Zaytoon, J.-L. Ferrier, J. Andrade-Cetto, and J. Filipe, Eds. Angers, France, 2007, INSTICC Press, 2007, pp. 124–130.

[33] Ducard, G. J. J.
Fault-Tolerant Flight Control and Guidance Systems: Practical Methods for Small Unmanned Aerial Vehicles.
London, United Kingdom: Springer, 2009.

[34] Stevens, B., and Lewis, F.
Aircraft Control and Simulation (2nd ed.). New York, NY: Wiley, 2003.

[35] Gavilan, F., Vazquez, R., and Acosta, J.-A.
Adaptive control for aircraft longitudinal dynamics with thrust saturation.
Journal of Guidance, Control, and Dynamics, **38**, 4 (2015), 651–661.

[36] Gavilan, F., Vazquez, R., and Esteban, S.
Trajectory tracking for fixed-wing UAV using model predictive control and adaptive backstepping.
In *Proceedings of the Workshop on Advanced Control and Navigation for Autonomous Aerospace Vehicles*, Seville, Spain, 2015, 132–137.

[37] Valenzuela, A., Rivas, D., Gavilán, F., and Vázquez, R.
Optimal trajectory generation for UAVs using dynamic trajectory modeling.
In *Proceedings of the Research, Development and Education on Unmanned Aerial Systems Workshop (RED-UAS)*, Seville, Spain, 2011, 29–40.

[38] Nelson, D., Barber, D., McLain, T., and Beard, R.
Vector field path following for miniature air vehicles.
IEEE Transactions on Robotics, **23**, 3 (2007), 519–529.

Francisco Gavilan received the M.Eng. and Ph.D. degrees in aerospace engineering and the M.S. degree in control theory from the University of Seville, Seville, Spain, in 2007, 2012, and 2009, respectively. Since 2007, he has been assistant professor with the Aerospace Engineering Department, University of Seville. His research interests include control theory, applied control to aerospace vehicles, UAV design, and orbital mechanics.

Rafael Vazquez (S'05—M'08—SM'15) received the M.S. and Ph.D. degrees in aerospace engineering from the University of California, San Diego (USA), and degrees in electrical engineering and mathematics from the University of Seville, Seville, Spain. He is currently an associate professor in the Aerospace Engineering Department, University of Seville. His research interests include control theory, distributed parameter systems, and optimization, with applications to flow control, ATM, UAVs, and orbital mechanics. He is coauthor of the book *Control of Turbulent and Magnetohydrodynamic Channel Flows* (2007).

Eduardo F. Camacho (F'11) received the Ph.D. degree in electrical engineering from the University of Seville, Seville, Spain. He is now a full professor with the Department of System Engineering and Automatic Control, University of Seville. He is author of *Model Predictive Control in the Process Industry* (1995), *Advanced Control of Solar Plants* (1997), and *Model Predictive Control* (1999; 2nd ed., 2004), *Control e Instrumentacin de Procesos Quimicos* and *Control of Dead-Time Processes* (2007), and *Control of Solar Systems* (2011). He has served on various IFAC technical committees and chaired the IFAC Publication Committee from 2002 to 2005. He was the president of the European Control Association (2005–2007) and chaired the IEEE/CSS International Affairs Committee (2003–2006), chair of the IFAC Policy Committee, and a member of the IEEE/CSS Board of Governors. He has acted as evaluator of projects at the national and European levels and was appointed manager of the Advanced Production Technology Program of the Spanish National R&D Program (1996–2000). He was one of the Spanish representatives on the Program Committee of the Growth Research Program and expert for the Program Committee of the NMP research priority of the European Union. He has carried out reviews and editorial work for various technical journals and many conferences. At present, he is one of the editors of the IFAC journal *Control Engineering Practice*, editor at large of the *European Journal of Control*, and subject editor of *Optimal Control: Methods and Applications*. Dr. Camacho is an IFAC fellow. He was publication chair for the IFAC World Congress 2002 and general chair of the joint IEEE CDC and ECC 2005 and co–general chair of the joint 50th IEEE CDC-ECC 2011.