



**HAL**  
open science

## Predictive Control for Autonomous Aerial Vehicles Trajectory Tracking

Ionela Prodan, Ricardo Bencatel, Sorin Olaru, João Borges de Sousa, Cristina Stoica, Silviu-Iulian Niculescu

► **To cite this version:**

Ionela Prodan, Ricardo Bencatel, Sorin Olaru, João Borges de Sousa, Cristina Stoica, et al.. Predictive Control for Autonomous Aerial Vehicles Trajectory Tracking. NMPC 2012 - 4th IFAC Conference on Nonlinear Model Predictive Control, Aug 2012, Noordwijkerhout, Netherlands. pp.508-513, 10.3182/20120823-5-NL-3013.00078 . hal-00708556

**HAL Id: hal-00708556**

**<https://centralesupelec.hal.science/hal-00708556>**

Submitted on 16 Mar 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Predictive Control for Autonomous Aerial Vehicles Trajectory Tracking<sup>\*</sup>

Ionela Prodan<sup>\*,\*\*</sup> Ricardo Bencatel<sup>\*\*\*</sup> Sorin Oлару<sup>\*</sup>  
João Borges de Sousa<sup>\*\*\*</sup> Cristina Stoica<sup>\*</sup>  
Silviu-Iulian Niculescu<sup>\*\*</sup>

<sup>\*</sup> *SUPELEC Systems Sciences (E3S) - Automatic Control Department,  
3 rue Joliot Curie, 91192, Gif sur Yvette, France,  
{ionela.prodan,sorin.olaru,cristina.stoica}@supelec.fr*

<sup>\*\*</sup> *Laboratory of Signals and Systems, SUPELEC - CNRS, 3 rue Joliot  
Curie, 91192, Gif sur Yvette, France  
{ionela.prodan,silviu.niculescu}@lss.supelec.fr*

<sup>\*\*\*</sup> *Department of Electrical and Computer Engineering, School of  
Engineering, University of Porto, 4200, Porto, Portugal  
{ricardo.bencatel,joao.sousa}@fe.up.pt*

---

**Abstract:** This paper addresses a predictive control strategy for Unmanned Air Vehicles in the presence of bounded disturbances. The goal is to guarantee tracking capabilities with respect to a reference trajectory which is pre-specified using the differential flatness formalism. Furthermore, an off-line linearization strategy of the nonlinear model of the vehicle along the flat trajectory is proposed. Since the reference trajectory is available beforehand, an optimization problem which minimizes the tracking error for the vehicle is formulated in a predictive control framework. The proposed method exhibits effective performance validated through software-in-the-loop simulations for the control of Unmanned Aerial Vehicles (UAVs).

Keywords: Unmanned Aerial Vehicle (UAV), differential flatness, constrained Model Predictive Control (MPC)

---

## 1. INTRODUCTION

The trajectory tracking problem represents a fundamental problem in control and coordination of dynamical systems, such as wheeled robots, spacecrafts, aircrafts, missiles, surface vessels, underwater vehicles or unmanned aerial vehicles (UAVs) (see, for instance, [Valavanis, 2007], [Aguiar and Hespanha, 2007] and the references therein). This problem is even more challenging because most of these dynamical systems are nonlinear, underactuated and exhibit nonholonomic constraints [Li and Canny, 1993], [Reyhanoglu et al., 1999].

A widely used technique in control community is *Model Predictive Control* (MPC) (see, for instance, [Mayne et al., 2000], [Maciejowski, 2002] for basic notions in MPC) due to its ability to handle control and state constraints, while offering good performance specifications. There are various applications in the literature where real-time MPC is applied to autonomous vehicle maneuvering problems. For example, [Keviczky and Balas, 2006] uses a predictive guidance controller for an autonomous UAV and a fault detection filter for taking into account the disturbances. Mixed-Integer Programming (MIP) techniques combined with receding horizon strategy was useful to coordinating the efficient interaction of multiple UAVs in scenarios with many sequential tasks and tight timing constraints (see,

[How et al., 2004], [Schouwenaars et al., 2005]). Furthermore, some works investigate the capability of Nonlinear MPC for tracking control. Among these contributions, [Kim et al., 2002] formulates a nonlinear MPC algorithm combined with the gradient-descent method for trajectory tracking, and [Fontes et al., 2009] proposes a two-layer control scheme composed by a nonlinear and a linear predictive controller for a group of nonholonomic vehicles moving in formation. However, it is important to point out that nonlinear MPC is computationally involved and thus, not necessarily suitable for a wide class of applications. This highlights the importance of developing simpler real-time optimization problems embedded within predictive control formulations for plants described by nonlinear models. In this sense, the authors of [Falcone et al., 2007] consider a MPC tracking controller based on successive “*on-line*” linearizations of the nonlinear model of the corresponding plant. Yet, the computational complexity of the proposed MPC scheme remains significant.

The main goal of the present paper is to develop a trajectory tracking predictive controller for a particular class of vehicles in the presence of bounded disturbances.

In a first stage, a specified trajectory is generated for a vehicle using the *differential flatness formalism* (see the work of [Fliess et al., 1995] for detailed notions of flatness). The proposed trajectory generation mechanism takes into account way-point conditions and furthermore, allows us to obtain “*off-line*” linearizations of the nonlinear vehicle

---

<sup>\*</sup> The research of Ionela Prodan is financially supported by the EADS Corporate Foundation (091-AO09-1006).

model along the flat trajectory. For reducing the computation effort we use the “nominal” behavior of the vehicle and consider safety region (the reader can refer to our previous work [Prodan et al., 2011]) around it to compensate for the effects of the disturbances affecting the “real” system. Furthermore, these regions can be defined within the theory of invariant sets [Rakovic et al., 2011] in order to avoid recomputations during the real-time functioning [Mayne et al., 2006].

In a second stage, since the reference trajectory is available beforehand, a *real-time optimization problem* which *minimizes the tracking error* for the vehicle is solved based on a prediction of the future evolution of the system [Bemporad and Morari, 1999], [Goodwin et al., 2005].

The following notations will be used throughout the paper. A Voronoi region,  $\mathcal{V}_i$  associated to a collection of points  $p_i$  is defined as  $\mathcal{V}_i = \{x \in \mathbb{R}^n : d(x, p_i) \leq d(x, p_r), \forall i \neq r\}$ , where  $d(x, y)$  denotes the distance between the points  $x$  and  $y$ . Minkowski’s addition of two sets  $\mathcal{X}$  and  $\mathcal{Y}$  is defined as  $\mathcal{X} \oplus \mathcal{Y} = \{x + y : x \in \mathcal{X}, y \in \mathcal{Y}\}$ . Let  $x_{k+1|k}$  denote the value of  $x$  at time instant  $k + 1$ , predicted upon the information available at time  $k \in \mathbb{N}$ . We write  $R \succ (\succeq)0$  to denote that  $R$  is a positive (semi)definite matrix.

## 2. PREREQUISITES

This section describes the vehicle’s model and the trajectory generation mechanism. Furthermore, an off-line linearization strategy of the nonlinear model is developed by using available operating points along the reference trajectory.

### 2.1 Vehicle Model

The framework presented here can be adapted to vehicles moving in 2D or 3D. Nevertheless, in the present work we explore the case of an airplane in which the autopilot forces coordinated turns (zero side-slip) at a fixed altitude. The airplane is represented by the following kinematic model [Bencatel et al., 2011]:

$$\begin{aligned} \dot{x}(t) &= V_a(t) \cos \Psi(t) + W_x, \\ \dot{y}(t) &= V_a(t) \sin \Psi(t) + W_y, \\ \dot{\Psi}(t) &= \frac{g \tan \Phi(t)}{V_a(t)}, \end{aligned} \quad (1)$$

where the state variables are represented by the position  $(x(t), y(t))$  and the heading angle  $\Psi(t) \in [0, 2\pi]$ . The input signals are the air-relative velocity  $V_a(t)$  and the bank angle  $\Phi(t) \in [-0.43, 0.43]$  radians, respectively<sup>1</sup>. Furthermore, we assume a nearly null angle of attack and that the autopilot provides a higher bandwidth regulator for the bank angle, making its dynamics negligible when compared to the heading dynamics.  $W_x$  and  $W_y$  are the wind velocity components on the  $x$  and  $y$  axis. The vehicle model is a nonholonomic system. It is completely controllable, but it cannot make instantaneous turns in certain directions.

*Remark 1.* Note that, the nonlinear systems used in a wide class of practical applications are usually differentially flat.

<sup>1</sup> The airspeed and the bank angle are regarded as the autopilot pseudo-controls.

In this sense, the dynamics we study in the present paper are the norm rather than the exception. For the cases where flat outputs are not possible, other strategies for trajectory generation can be used [Aguilar and Hespanha, 2007], [Castillo et al., 2007].

### 2.2 Flat trajectory generation

Consider notations:

$$\xi(t) = [x(t) \ y(t) \ \Psi(t)]^T, \quad (2)$$

$$u(t) = [V_a(t) \ \Phi(t)]^T, \quad (3)$$

denoting the state vector and the input vector, respectively. Then, the general system (1) can be described as:

$$\dot{\xi}(t) = f(\xi(t), u(t)), \quad (4)$$

where  $f(\cdot, \cdot) : \mathbb{R}^3 \times \mathbb{R}^2 \rightarrow \mathbb{R}^3$  is the state derivation function.

In the following, we require the determination of a reference trajectory  $(\xi^{ref}(t), u^{ref}(t))$  that steers the model (1) from an initial state  $\xi^{ref}(t_0)$  to a final state  $\xi^{ref}(t_f)$ , over a fixed time interval  $[t_0, t_f]$ . Since the aforementioned system is controllable and allows the existence of flat outputs (see, for instance [Van Nieuwstadt and Murray, 1998], [De Doná et al., 2009]), we consider flatness theory notions in order to provide the required reference trajectory.

The systems’ state and input will be represented as functions of a finite dimensional mapping  $z(t)$  and a finite number of its derivatives (in this particular case it will be shown that the second order derivative suffices):

$$\begin{aligned} \xi^{ref}(t) &= \eta_0(z(t), \dot{z}(t)), \\ u^{ref}(t) &= \eta_1(z(t), \dot{z}(t), \ddot{z}(t)). \end{aligned} \quad (5)$$

In the case of the dynamics (1), the vector  $z(t) = [z_1(t) \ z_2(t)]^T \in \mathbb{R}^2$ , called the flat output is defined as:

$$\begin{aligned} z_1(t) &= x(t), \\ z_2(t) &= y(t). \end{aligned} \quad (6)$$

It can be shown that, the corresponding reference state and input for the system (4) are obtained by replacing the reference flat output (6) into equations (5):

$$\begin{aligned} \xi^{ref}(t) &= \left[ z_1(t) \ z_2(t) \ \arctan \left( \frac{\dot{z}_2(t)}{\dot{z}_1(t)} \right) \right]^T, \\ u^{ref}(t) &= \left[ \sqrt{\dot{z}_1^2(t) + \dot{z}_2^2(t)} \ \arctan \left( \frac{1}{g} \frac{\ddot{z}_2(t)\dot{z}_1(t) - \dot{z}_2(t)\ddot{z}_1(t)}{\sqrt{\dot{z}_1^2(t) + \dot{z}_2^2(t)}} \right) \right]^T, \end{aligned} \quad (7)$$

where  $t \in [t_0, t_f]$ .

For a practical implementation, the flat output  $z(t)$  is seen as a weighed sum of basis mappings. Imposing boundary constraints for the evolution of the differentially flat systems (see, for instance [De Doná et al., 2009]) a flat output  $z(t)$  can be generate by the resolution of a *linear* system of equalities. The idea used in our approach is to further introduce a set of way-points through which the vehicle must pass<sup>2</sup>:

$$\mathbb{P} \triangleq \{p^i = (\xi^i, u^i), \ i = 0, \dots, N_p\}, \quad (9)$$

where  $N_p$  is the number of chosen way-points.

<sup>2</sup> Hereafter whenever we use the subscript we refer to time and when we use the superscript we index a point from a set of points.

The list of way-points is assumed as being provided by an operator (which can oversee the operation) and incorporates control requirements: obstacle avoidance, check points, etc. Note that polynomial basis functions are a poor choice because their dimension depends on the number of constraints. This means that they are sensitive to the number of way-points. More precisely, in this case the trajectory needs to be computed on segments (i.e., each segment taken between two consecutive way-points). Therefore, additionally constraints are imposed leading to an increased number of polynomial basis functions beyond reasonable computation limits. To overcome this issues, we used B-splines functions which are the best choice in the sense that their degree does not depend on the number of way-points. Actually, the degree depends only up to which derivative we want to assure continuity. In our particular case, the third degree order suffices to assure smooth bank and velocity commands. As an illustrative example Figure 1 shows a flat trajectory which passes through 4 way-points as in (9).

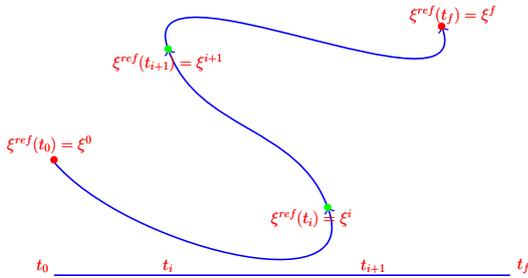


Fig. 1. Flat trajectory which passes through 4 way-points.

### 2.3 Linearization of the vehicle model

For computation purposes, it is convenient to use the discretized model of the nonlinear system<sup>3</sup> (4):

$$\xi_{k+1} = f^d(\xi_k, u_k). \quad (10)$$

In the sequel, we consider the linearization problem of the nonlinear discretized system (10). We take here a piece-wise affine (PWA) approach, that is, we consider a collection of points in which we pre-compute linear approximations of (10):

$$\mathbb{L} \triangleq \{l^j = (\xi^j, u^j), \quad j = 0, \dots, N_l\}, \quad (11)$$

with  $N_l$  the number of chosen linearization points.

For a given point  $l^j \in \mathbb{L}$  we consider the following Taylor decomposition:

$$f^d(\xi_k, u_k) = f^d(\xi^j, u^j) + A_j(\xi_k - \xi^j) + B_j(u_k - u^j) + \underbrace{\dots}_{\beta_j(\xi_k, u_k)}, \quad (12)$$

where the matrices  $A_j \in \mathbb{R}^{3 \times 3}$  and  $B_j \in \mathbb{R}^{3 \times 2}$  are defined as

$$A_j = \frac{\partial f^d}{\partial \xi} \Big|_{\xi^j, u^j}, \quad B_j = \frac{\partial f^d}{\partial u} \Big|_{\xi^j, u^j} \quad (13)$$

<sup>3</sup> In general, time discretization of the system dynamics (4) is performed using the classical Euler method or Adams-Bashforth multi-step methods.

and  $\beta_j(\xi_k, u_k) \in \mathbb{R}^3$  represents the terms of the Taylor decomposition of rank greater than 1 (i.e., the nonlinear residue of the linearization):

$$\beta_j(\xi_k, u_k) = f^d(\xi_k, u_k) - f^d(\xi^j, u^j) - A_j(\xi_k - \xi^j) - B_j(u_k - u^j), \quad (14)$$

for all  $j = 0, \dots, N_l$ . Therefore, the system (10) can be linearized in  $l^j \in \mathbb{L}$  by the following dynamics:

$$\xi_{k+1} = f_j^d(\xi_k, u_k) \triangleq A_j \xi_k + B_j u_k + r_j, \quad (15)$$

with the affine constant terms  $r_j \in \mathbb{R}^3$  defined as:

$$r_j = f^d(\xi^j, u^j) - A_j \xi^j - B_j u^j, \quad (16)$$

for all  $j = 0, \dots, N_l$ .

In the following we consider a procedure of selecting between the predefined linearization points (11) for the current input/state values. To this end, we partition the state-space into a collection of Voronoi cells:

$$\mathcal{V}_j = \{\xi : \|\xi - \xi^j\| \leq \|\xi - \xi^r\|, \quad \forall r \neq j\}, \quad (17)$$

where each cell consists of all points whose linearization error is less for linearization around point  $\xi^j$  than for any other point  $\xi^r$  from  $\mathbb{L}$ , with  $r, j = 0, \dots, N_l$ . This allows to select the linearization point during runtime by a simple membership testing:

$$\xi_{k+1} = f_j^d(\xi_k, u_k), \quad \forall (\xi_k, u_k) \in \mathcal{V}_j. \quad (18)$$

Since  $\beta_j(\xi_k, u_k) = f^d(\xi, u) - f_j^d(\xi, u)$  it is clear that the linearization error is related to its corresponding cell,  $\mathcal{V}_j$ :

$$\|\beta_j(\xi_k, u_k)\| \leq \max_{(\xi, u) \in \mathcal{V}_j} \|f^d(\xi, u) - f_j^d(\xi, u)\|. \quad (19)$$

Consequently, a Voronoi decomposition with bounded cells of small diameter is desirable in order to have a “good” PWA approximation of the function (10).

*Remark 2.* An a priori computation of (13), (14) and (16) in all feasible combinations of inputs and states is difficult to handle. As such, we prefer to select the linearization points (11) along (and around) the flat trajectory under the assumption (to be tested later) that the real trajectory will stay “close enough” and thus, the chosen linearization points will remain relevant to the problem at hand [Fagiano et al., 2009].

*Proof of concept:* In order to better explain the linearization strategy, we illustrated in Figure 2 the continuous trajectory of the nonlinear system (4) (in blue) and the piecewise linearized trajectory (in red). Therefore, the linear system (15) describes the small deviations of the nonlinear system (10) trajectory from the reference state trajectory  $\xi^{ref}(t)$  described by (7) when an input reference  $u^{ref}(t)$  described by (8) is applied. We considered several linearization points (denoted as black dots)<sup>4</sup> and constructed the Voronoi cells as in (17).

By linearizing the reference trajectory we have now all the tools necessary for the control part of the trajectory tracking problem, which will be detailed in the forthcoming section.

## 3. TRAJECTORY TRACKING PROBLEM

Since the reference trajectory is available beforehand (through the use of flatness procedures), an optimization

<sup>4</sup> Note that the way-points (9) can also be considered between the linearization points (11).

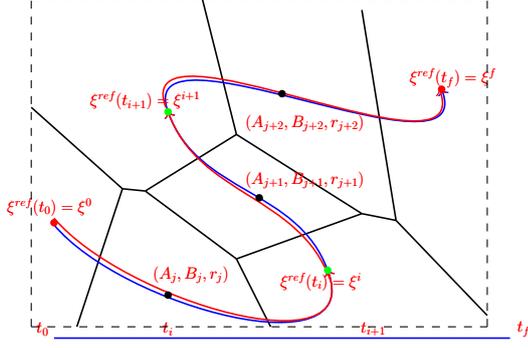


Fig. 2. Real and linearized trajectories and the bounded Voronoi cells.

problem which minimizes<sup>5</sup> the nominal tracking error for the vehicle is formulated in a predictive control framework. Roughly, the vehicle must follow the reference trajectory using the available information over a finite time horizon in the presence of constraints.

For a practical implementation we consider the recursive construction of an optimal control sequence  $\mathbf{u} = \{u_{k|k}, u_{k+1|k}, \dots, u_{k+N-1|k}\}$  over a *finite* constrained receding horizon, which leads to a predictive control policy:

$$u^* = \arg \min_{\mathbf{u}} \sum_{s=0}^{N-1} (\| \xi_{k+s|k} - \xi_{k+s|k}^{ref} \|_Q + \| u_{k+s|k} - u_{k+s|k}^{ref} \|_R), \quad (20)$$

subject to the set of constraints:

$$\begin{cases} \xi_{k+s+1|k} = A_j \xi_{k+s|k} + B_j u_{k+s|k} + r_j, \\ \xi_{k+s|k} \in \mathcal{X}, \quad s = 1, \dots, N-1, \\ u_{k+s|k} \in \mathcal{U}, \quad s = 1, \dots, N-1. \end{cases} \quad (21)$$

with  $(\xi_{k+s|k}, u_{k+s|k}) \in \mathcal{V}_j$  as defined in (17),  $j = 1, \dots, N_l$ . Here  $Q = Q^T \succeq 0$ ,  $R \succ 0$  are weighting matrices and  $N$  denotes the length of the prediction horizon.

The optimization problem (20) has to be solved subject to the dynamical constraints (21). In the same time, other security or performance specifications can be added to the system trajectory. These physical limitations (velocity, bank command) are stated in terms of hard constraints on the internal state variables and input control action as in (21). Note that the sets  $\mathcal{X}$ ,  $\mathcal{U}$  have to take into account the reference tracking type of problem delineated in (20). Thus, the absolute limitations have to be adjusted according to the reference signals. In the original state space coordinates, these constraints will describe a tube around the reference trajectory.

The trajectory obtained by applying the optimal control  $u^*$  computed in (20)–(21) is “nominal”, in the sense that it does not consider either exogenous noises (i.e., the wind) or the state-dependent linearization error (i.e., the term  $\beta_j(\xi_k, u_k)$  from (12)). There are different approaches in the literature which deal with the reference trajectory tracking problem for dynamical systems affected by disturbances. A classical method is based on the tube MPC approach (for details, the reader is referred to [Mayne et al., 2006],

<sup>5</sup> Since the nominal trajectory has to respect state and input constraints, the vehicle may not follow exactly the reference trajectory.

[Rakovic et al., 2011]), where a nominal trajectory is controlled and the real trajectory is kept into a tube around the nominal one through a suitable control action.

The “real” trajectory takes into account all the perturbations:

$$\xi_{k+1}^o = A_j \xi_k^o + B_j u_k^o + r_j + \beta_j(\xi_k^o, u_k^o) + w_k, \quad (22)$$

where the bounded perturbation  $w_k$  denotes the wind.

Subsequently, subtracting (15) from (22) we obtain the tracking error  $z_k = \xi_k^o - \xi_k$  measuring the difference between real ( $\xi_k^o$ ) and nominal ( $\xi_k$ ) trajectories:

$$z_{k+1} = A_j z_k + B_j u_k^\delta + \beta_j(\xi_k^o, u_k^o) + w_k, \quad (\xi_k^o, u_k^o) \in \mathcal{V}_j, \quad (23)$$

where  $u_k^\delta = u_k^o - u_k$  denotes the difference between “real” and “nominal” control actions.

Considering that the perturbations are bounded,<sup>6</sup> as long as a stabilizable control action:

$$u_k^\delta = \mathcal{K}(\xi_k, \xi_k^o) \quad (24)$$

exists, we can guarantee that the real trajectory (22) remains “close” enough to the nominal one (15). Actually, there exists a sequence of bounded sets which describe the tube:

$$z_k \in S_k \leftrightarrow \xi_k^o \in \{\xi_k\} \oplus S_k, \quad \forall k \geq 0. \quad (25)$$

For LTI dynamics, the choice of (24) is simply a gain matrix which makes the closed-loop dynamics stable. Here, due to the switched nature of (15) we need also to switch between the gain matrices:

$$\mathcal{K}(\xi_k, \xi_k^o) = K_j(\xi_k^o - \xi_k), \quad j = 0, \dots, N_l, \quad (26)$$

with each gain  $K_j$  stabilizing the pair  $(A_j, B_j)$  in (15). Then, we obtain the switched system:

$$z_{k+1} = (A_j + B_j K_j) z_k + \beta_j(\xi_k^o, u_k^o) + w_k, \quad (\xi_k^o, u_k^o) \in \mathcal{V}_j \quad (27)$$

which, under mild assumptions (that the switches between modes of functioning are not happening often) is stable.

Of practical interest is the computation of the “tube” defined by sets  $S_k$ . For LTI dynamics, the sets  $S_k$  are constructed robust positively invariant (see previous work [Prodan et al., 2011]) in order to minimize the on-line computations. Here the dynamics of the vehicle change whenever the linearization point changes and it may not be possible to describe the set linking the “nominal” and the “real” dynamics in the same manner. In this case, a hybrid structure can be proposed. That is, we compute robust invariant sets for each of the linearized dynamics and change between them (or scaled versions of them) when necessary.

#### 4. SIMULATION RESULTS

The current research is to develop software-in-the-loop simulations and subsequent flight tests for the control of small Unmanned Aerial Vehicles (UAVs). The vehicle dynamics is simulated by a Piccolo software ([http://www.cloudcaptive.com/piccolo\\_system.shtm](http://www.cloudcaptive.com/piccolo_system.shtm)) and then, the control algorithm is transmitted in real flight simulations through a communication routine running on a PC on the ground. Furthermore, the autopilot sends the telemetry data to the ground station through a 2.4GHz Piccolo protocol link.

<sup>6</sup> If the cell  $\mathcal{V}_j$  is bounded, then the nonlinear residue  $\beta_j(\xi_k, u_k)$  is also bounded.

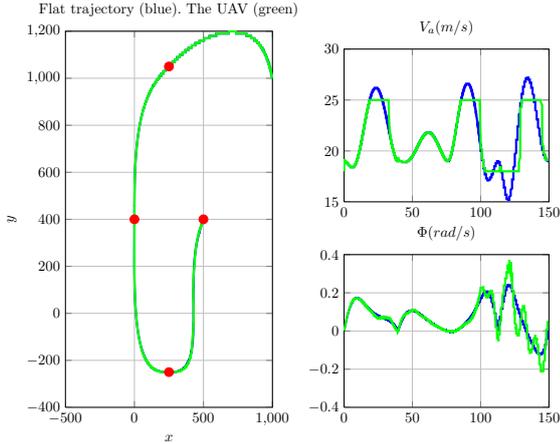


Fig. 3. Reference trajectory and actual UAV motion.

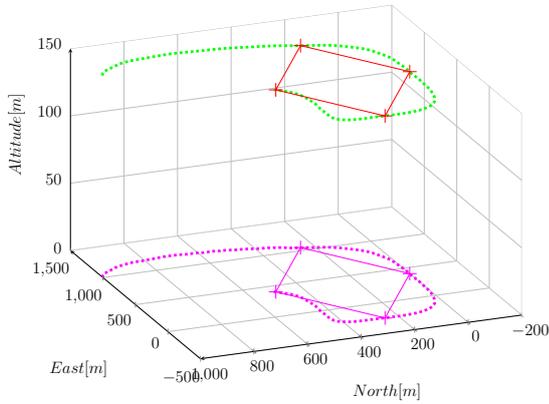


Fig. 4. Actual UAV motion and its projection on the x-y space.

In the present paper we provide software-in-the-loop simulations and we are currently working on validating the approach on the associated hardware. Therefore, instead of the usage of electrical interfaces, software interface provided by Piccolo is used here which allows a direct information-technical communication with the simulation. The control algorithm is running in MATLAB which is interfaced with Piccolo software using DUNE and IMC messaging system (see, for details [Pinto et al., 2012]).

Furthermore, to test the proposed trajectory tracking method we use an extended unicycle aircraft model (for low-level control, [Bencatel et al., 2011]) with 12 states, in a 3 DOF simulation. The simulated UAV is restricted to a range of speed between 18 and 25 meters per second and to a maximum bank angle of 25 degrees. The simulations incorporates perturbations, e.g., the wind. We assume that the intensity of the wind is bounded for some reasonable values (e.g., a maximum speed of 20-knots).

As an exemplification, we now describe simulation results that illustrate the performance of the proposed predictive tracking controller. The objective is to force the UAV to track 4 way-points given as in (9) (denoted as red dots in Figure 3). In a first stage, using the results in Section 2 we generate a flat trajectory starting from the current position of the UAV and passing through the given way-points. In a second stage, we use the linearized model (see the linearization procedure in Subsection 2.3) for the control

part of the trajectory tracking problem with the following tuning parameters:

- the sample time is  $T = 0.05$  s;
- constraints on the input: the velocity  $V_a \in [18, 25]$  m/s, the bank angle  $\Phi \in [-0.43, 0.43]$  rad, the  $W_x$  and  $W_y$  components of the wind are limited to less than 20 knots;
- small variations on the velocity and bank command are admitted : the rate of change of  $V_a$  is limited to the maximum acceleration the aircraft can produce, i.e.,  $0.1 \sim 0.2$  m/s<sup>2</sup>; the variation of  $\Phi$  is limited to  $30 \sim 60$  degrees/s;
- the weights matrices in (20) are  $Q = 5 \cdot 10^3 \cdot I_3$ ,  $R = I_2$ ;
- the prediction horizon is  $N = 5$ .

Good tracking performances for the given reference trajectory (depicted in blue in Figure 3) are obtained for the UAV, whose evolution is depicted in green in Figure 3. Moreover, the constraints on the velocity and bank commands are satisfied as illustrated in green in the same figure. It is worth mentioning that an increase in the prediction horizon leads to better tracking performances thus, imposing a trade-off between complexity and precision.

Figure 4 depicts a 3D simulation of the UAV evolution which tracks very well the given way-points. In the same figure we represented the projection of the trajectory on the ground.

The predictive tracking controller was also tested in simulations with different wind conditions with a maximum speed of 20-knots and showed good robustness. Figure 5 shows the reference trajectory in solid blue and the tracking performance of the UAV with and without wind conditions. The main limitations against improved performance are the wind magnitude and the numerical issues (density of linearization points, prediction horizon length, etc.).

## 5. CONCLUSIONS

This paper addresses a predictive control strategy for Unmanned Aerial Vehicles (UAVs) in the presence of bounded disturbances. A reference trajectory is pre-specified using differential flatness formalism. Furthermore, an “off-line” linearization strategy of the nonlinear model of the vehicle along the flat trajectory is proposed. For reducing the computation effort we use the “nominal” behavior of the vehicle and consider a safety region around it to compensate for the effects of the disturbances affecting the “real” system. Next, an optimization problem which minimizes the tracking error for the vehicle is formulated in a predictive control framework. Software-in-the-loop simulations for the control of (UAVs) are presented. We are currently working on hardware-in-the-loop simulations and on the extension of the proposed trajectory tracking strategy for UAVs formations with non-convex collision avoidance constraints.

## REFERENCES

- Aguiar A.P. and Hespanha J.P. (2007): Trajectory-tracking and path-following of underactuated autonomous vehicles with parametric modeling uncertainty. *IEEE TAC*, 52(8):1362–1379.

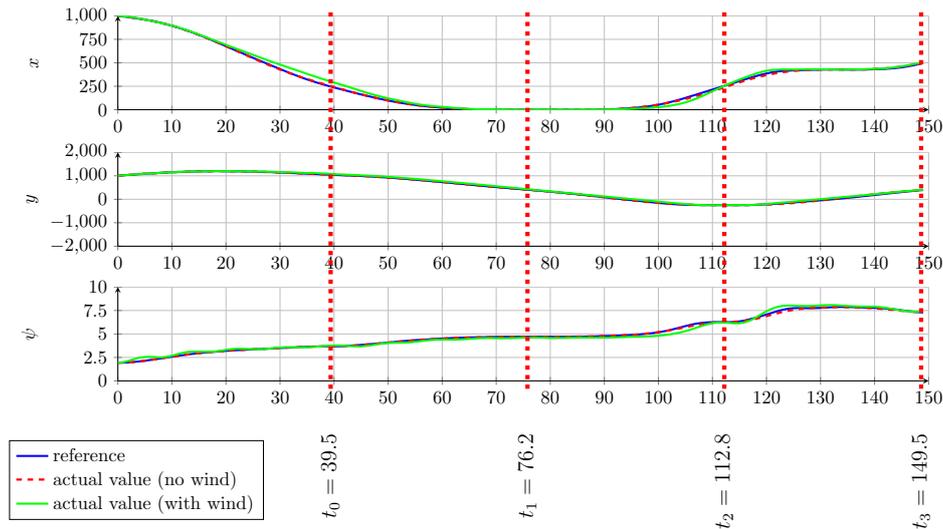


Fig. 5. Comparison between actual and reference UAV motion.

- Bemporad A. and Morari. M. (1999): Robust model predictive control: A survey. *Robustness in identification and control*, pp. 207–226.
- Bencatel R., Faied M., Sousa J.B., and Girard A.R. (2011): Formation control with collision avoidance. In *Proceedings of the 50th IEEE CDC-ECC*, pp. 591–596, Orlando, USA.
- Castillo C.L., Moreno W., and Valavanis K.P. (2007): Unmanned helicopter waypoint trajectory tracking using model predictive control. In *Proceedings of the IEEE MCC*, pp. 1–8.
- De Doná J., Suryawan F., Seron M., and Lévine J. (2009): A flatness-based iterative method for reference trajectory generation in constrained NMPC. *Nonlinear Model Predictive Control*, pp. 325–333.
- Fagiano, L. and Canale, M. and Milanese, M. (2009): Set Membership approximation of discontinuous NMPC laws. In *Proceedings of the 48th IEEE CDC and the 28th CCC*, pp. 8636–8641.
- Falcone P., Tufo M., Borrelli F., Asgari J., and Tseng H.E. (2007): A linear time varying model predictive control approach to the integrated vehicle dynamics control problem in autonomous systems. In *Proceedings of the 46th IEEE CDC*, pp. 2980–2985.
- Fliess M., Lévine J., Martin P., and Rouchon P. (1995): Flatness and defect of non-linear systems: introductory theory and examples. *International Journal of Control*, 61(6):1327–1361.
- Fontes F., Fontes D., and Caldeira A. (2009): Model predictive control of vehicle formations. *Optimization and Cooperative Control Strategies*, pp. 371–384.
- Goodwin G.C., Seron M., and De Dona J. (2005): *Constrained control and estimation: an optimization approach*. Springer Verlag.
- How J., King E., and Kuwata Y. (2004): Flight demonstrations of cooperative control for UAV teams. In *AIAA 3rd Workshop and Exhibit “Unmanned Unlimited” Technical Conference*, pp. 20–23.
- Keviczky T. and Balas G.J. (2006): Software-enabled receding horizon control for autonomous unmanned aerial vehicle guidance. *Journal of Guidance Control and Dynamics*, 29(3):680–694.
- Kim H.J., Shim D.H., and Sastry S. (2002): Nonlinear model predictive tracking control for rotorcraft-based unmanned aerial vehicles. In *Proceedings of the 21st American Control Conference*, volume 5, pp. 3576–3581.
- Li Z. and Canny J. (1993): *Nonholonomic motion planning*, volume 192. Kluwer Academic Pub.
- Maciejowski J.M. (2002): *Predictive control: with constraints*. Pearson education.
- Mayne D.Q., Rawlings J.B., Rao C.V., and Scokaert P.O. (2000): Constrained model predictive control: Stability and optimality. *Automatica*, 36:789–814
- Mayne D.Q., Raković S.V., Findeisen R., and Allgöwer F. (2006): Robust output feedback model predictive control of constrained linear systems. *Automatica*, 42(7):1217–1222.
- Pinto J., Calado P., Braga J., Dias P., Martins R., and Marques E. (2012): Implementation of a control architecture for networked vehicle systems. In *Proceedings of the IFAC Workshop on NGCUV*.
- Prodan I., Olaru S., Stoica C., and Niculescu S.I. (2011): Predictive control for tight group formation of multi-agent systems. In *Proceedings of the 18th IFAC World Congress*, pp. 138–143
- Rakovic S.V., Kouvaritakis B., Cannon M., Panos C., and Findeisen R. (2011): Fully parameterized tube MPC. In *Proceedings of the 18th IFAC World Congress*, volume 18, pp. 197–202.
- Reyhanoglu M., Van der Schaft A., McClamroch N.H., and Kolmanovsky I. (1999): Dynamics and control of a class of underactuated mechanical systems. *IEEE Transactions on Automatic Control*, 44(9):1663–1671.
- Schouwenaars T., Valenti M., Feron E., and How J. (2005): Implementation and flight test results of MILP-based uav guidance. In *Proceedings of the IEEE Aerospace Conference*, pp. 1–13.
- K. Valavanis. (2007): *Advances in unmanned aerial vehicles: state of the art and the road to autonomy*, volume 33. Springer Verlag
- Van Nieuwstadt M.J. and Murray R.M. (1998): Real-time trajectory generation for differentially flat systems. *International Journal of Robust and Nonlinear Control*, 8(11):995–1020.