Tube-based Robust MPC Processor-In-the-Loop Validation for Fixed-Wing UAVs

Martina Mammarella
Department of Mechanical
and Aerospace Engineering
Politecnico di Torino, Turin, Italy
Italian National Research Council - IEIIT
Email: martina.mammarella@ieiit.cnr.it

Elisa Capello
Department of Mechanical
and Aerospace Engineering
Politecnico di Torino, Turin, Italy
Italian National Research Council - IEIIT
Email: elisa.capello@polito.it

***Abstract*—Real systems, as Unmanned Aerial Vehicles (UAVs), are usually subject to environmental disturbances, which could compromise the mission accomplishment. For this reason, the main idea proposed in this research is the design of a robust controller, as autopilot control system candidate for a fixed-wing UAV. In detail, the inner loop of the autopilot system is designed with a tube-based robust model predictive control (TRMPC) scheme, able to handle additive noise. Moreover, the navigation outer loop is regulated by a proportional-integral-derivative controller. The proposed TRMPC is composed of two parts: (i) a linear nominal dynamics, evaluated online with an optimization problem, and (ii) a linear error dynamics, which includes a feedback gain matrix, evaluated offline. The key aspects of the proposed methodology are: (i) offline evaluation of the feedback gain matrix, and (ii) robustness to random, bounded disturbances. Moreover, a path-following algorithm is designated for the guidance task, which provides the reference heading angle as input to the control algorithm. Software-in-the-loop and processor-in-the-loop simulations have been performed to validate the proposed approach. The obtained performance have been evaluated in terms of tracking capabilities and computational load, assessing the real-time implementability compliance with the XMOS development board, selected as continuation of previous works.**

I. INTRODUCTION

In the last decades, autonomous Unmanned Aerial Vehicles (UAVs) have been able to execute inspection, surveillance and search-and-rescue missions, guaranteeing trajectory tracking and stability performance, thanks to tailored control schemes. Moreover, depending on the required task, these platforms, either fixed-wing UAVs (FW-UAVs) or multi-rotor systems, have proven themselves as suitable tools for testing advanced control techniques, even when real-time implementability is a strict requirement. Most of the off-the-shelf autopilots [1] incorporate control algorithms to meet the requirements of flight maneuvers and mission accomplishment. However, due to the complexity and the limited computational capability, most of them are based on Proportional Integral Derivative (PID) control algorithms. On the other hand, this category of controllers lacks of adaptivity with respect to unmodeled dynamics and external disturbance sources.

In real-case applications, mini-UAVs are usually subject to environmental disturbances, e.g. wind turbulence or wind gust. Hence, an increasing effort has been posed in designing robust controllers, which would allow to properly perform the desired mission even in the presence of disturbance sources. The proposed robust approaches range from PID [2], [3] and H_∞ [4], [5] approaches to adaptive control techniques as in [6], [7]. Another example is proposed in [8] where a $\mathcal{L}_1$ adaptive scheme has been designed for a mini-UAV autopilot, which has shown an inherent robustness to external and internal parameters variation. Even if all these strategies could guarantee robustness to bounded modeled disturbances if properly designed, they are not able to explicitly tackle with mission and system constraints unlike Model Predictive Control (MPC) schemes. For this reason, MPC has become widely used in many application domains, including path-following missions of FW-UAVs. For example, a two-loops algorithm was presented in [9], where an explicit MPC for pitch and roll control was combined with an $\mathcal{L}_1$ navigation loop for the altitude. In this work, the benefits of exploiting a more advanced control technique as MPC have been demonstrated comparing the proposed approach with a classical PID controller. In [10], a tracking nonlinear MPC has been designed for the lateral-directional dynamics control of a FW-UAV in the presence of wind disturbances. In this case, the internal loop control for the attitude is provided by a PD controller. Another example is presented in [11], in which different receding horizon methods for trajectory tracking, i.e. slow dynamics, are proposed. In a similar way, an MPC scheme based on the theory of minimum-peak performance is proposed in [12], ensuring a maximum deviation from the reference trajectory. In [13], two different Robust MPC (RMPC) schemes are proposed for the control of UAV translational dynamics and the main focus is on their robustness to additive disturbance without investigating the corresponding computational burden.

The focus of this work is the preliminary validation of a computationally-efficient robust MPC-based controller, the so-called Tube-based Robust MPC (TRMPC), to control a FW-UAV when atmospheric disturbances are acting on the aircraft. This predictive control technique has been proposed in [14] and [15] and it has been selected in this work due to its capability to guarantee robustness to persistence disturbance with a computational demand lower than other RMPC approaches. Indeed, while providing robustness to bounded disturbance, it maintains the computational efficiency of a classical MPC scheme thanks to its two-layer algorithm logic, as discussed in [15]. The main idea behind this control scheme consists of controlling the correspondent nominal, undisturbed system, subject to tightened constraints, to guarantee all possible real trajectories, each one characterized by a different realization of a disturbance sequence, to robustly in a time-varying sequences of sets, also known as *tube*. Hence, we first evaluate *offline* the tightened constraint sets and the feedback gain matrix that quadratically stabilizes the closed-loop system. In particular, a structured model uncertainties have been considered to address the robustness and stability of the system exploiting the well-known Edge Theorem to evaluate the stabilizing matrix. Then, we solve *online* a classical finite-horizon optimal receding problem considering the undisturbed nominal dynamics, obtaining the optimal nominal trajectory that will represent the center of the tube itself. Going into the details of the control strategy, the aforementioned TRMPC has been designed for the inner-loop dynamics, i.e. pitch, roll and airspeed, as well as for the altitude outer-loop, while a PID controller has been implemented for the control of the heading angle.

The work proposed in this paper represents the extension of [16], where the TRMPC has been preliminary tested considering both linear and nonlinear models and compared with a $\mathcal{L}_1$ strategy, already proposed in [17]. In particular, in [16], we focused on the control performance of the selected strategies and their capability to track simple step reference signals in the presence of additive noise. Moreover, to validate the robustness of TRMPC, the same control strategy has been applied to different aircraft configurations (different mass and inertia) and flight conditions (relative airspeed). Last, the simulation computational load has been assessed exploiting the *Real-Time Pacer* MATLAB/Simulink Toolbox (version 1.0.0.1). On the other hand, in this work, the TRMPC scheme has been exploited in a more wide and realistic scenario and it has been combined with the guidance algorithm proposed in [18] to control longitudinal and lateral-directional dynamics of a FW-UAVs while performing a classical patrolling mission over a selected area. Indeed, the FW-UAV is called to follow a butterfly pattern and the TRMPC shall guarantee the required trajectory and attitude tracking performance. Moreover, extensive and detailed analyses have been dedicated to study the effect of disturbance entity and prediction horizon on controller performance and related computational burden.

It is important to highlight that this work represent a continuation of previous works of the same research group, e.g. see [3], [17], [19], where the main goal is to assess the performance and computational load of different advanced control techniques at the same conditions. Hence, to be coherent with previous works and to obtain comparable data, the same mission, vehicle, and validation equipment, i.e. the microcontroller, has been considered also in this work. Further details can be found in Section 2 and Section 3. The effectiveness of the proposed robust approach has been already experimentally validated for a space rendezvous mission on an experimental testbed, as described in [20]. In that experimental campaign, the spacecraft dynamics was running at 0.01 s while the TRMPC controller was updated every 3 s to comply with the mission requirements and platform computational constraints[1]. The major differences of the application studied in this paper with respect to the space one can be summarized as follows: i) larger design space, i.e. 9 state variables and 3 control inputs; ii) faster dynamics characterizing UAVs with respect the orbital and attitude dynamics of spacecraft; iii) smaller control sample time, i.e. the controller is updated every 0.1 s; and iv) lower computational capabilities of the target development board, i.e. only 64 Kbytes of RAM and 8 Kbytes of OTP memory.

To properly evaluate the reliability of the proposed guidance and control scheme, the classical multi-step software verification $\&$ validation approach for model-based design has been followed. As described in [21], the first step consists in performing Software-In-the-Loop (SIL) testing, i.e. executing the controller algorithm for non-real-time execution on the same host platform that is used by the modeling environment. The next step involves the so-called Processor-In-the-Loop (PIL) testing, during which the model does a single calculation iteration. Then, inputs are calculated and passed to a PIL block, which passes the model inputs to the code running on the embedded microprocessor. In particular, the code executes on the actual embedded processor, using the embedded cross-compiler. Once the target processor computes the output data, they are passed back to model using the same PIL block. The two aforementioned testing phases are preliminary to the following phase, i.e. the Hardware-In-the-Loop (HIL) testing, which will serve as a final lab test phase before flight tests, in this particular case. Hence, in this work both SIL and PIL simulations have been performed, unlike what was done in [16]. In the latter case, PIL simulations have been executed exploiting a commercial development board, which features are similar to those of the autopilot equipped on-board the targeted FW-UAV, i.e. the mini-UAV

[1]The on-board computational capabilities were provided by a PC-104 form-factor on-board computer, based on an Intel Atom 1.6 GHz 32-bit processor, with 2 GB of RAM and an 8 GB solid-state drive. The real-time operating system was a Ubuntu 10.04, 32-bit server-edition with a Linux kernel 2.6.33.

MH850 developed at the Department of Mechanical and Aerospace Engineering of Politecnico di Torino ([8], [17]). The effectiveness of the TRMPC algorithm is validated in the presence of a fixed-direction wind turbulence and the implementability is tested evaluating the computational cost with respect to the limited capabilities of the development board, also considering the effect of varying disturbance and prediction horizon on the computational load.

The paper is organized as follows. The nonlinear and linearized aircraft models used for the UAV motion propagation and the controller design, respectively, are described in Section II, together with the SIL and PIL simulation environments, the guidance algorithm and the battery model. In Section III, the TRMPC algorithm is presented from a theoretical point-of-view. Then, SIL and PIL simulation results are described in details in Section IV considering a butterfly path, providing a thorough overview of the guidance and control scheme effectiveness and their real-time implementability and computational load. Finally, conclusions are drawn in Section V.

II. Aircraft Model and Simulation Environment Description

The aircraft considered in this work is the tailless fixed-wing mini-UAV MH850, developed at the Department of Mechanical and Aerospace Engineering of Politecnico di Torino ([8], [17]) (see Figure 1). With a mass of about 1 kg and a wingspan of about 85 cm, this UAV is able to fly for 45 minutes at a cruise speed of 13.5 m/s. The numerically-derived database comprehensive of all aerodynamic derivatives is thoroughly described in [22] and [23].

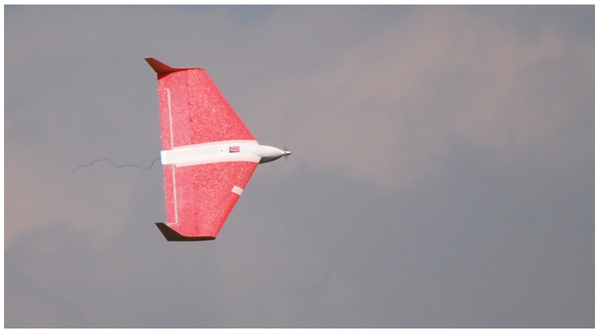


Fig. 1. The MH850 mini-UAV.

A. Nonlinear Aircraft Model

The nonlinear model considered for the aircraft dynamics is based on a set of nine equations, written in a body reference frame, as reported in [24]. In particular, the total airspeed $V = [u\ v\ w]$ can be decomposed in the longitudinal, lateral and vertical components along the three body axes, respectively, as

$$\dot{u} = \frac{F_X}{m_{UAV}} - qw + rv - g\sin\vartheta, \tag{1a}$$
$$\dot{v} = \frac{F_Y}{m_{UAV}} + pw - ru + g\cos\vartheta\sin\phi, \tag{1b}$$
$$\dot{w} = \frac{F_Z}{m_{UAV}} - pv + qu + g\cos\vartheta\cos\phi, \tag{1c}$$

with m_{UAV} the aircraft mass, $[F_X\ F_Y\ F_Z]^T$ the forces acting on the system, g the gravity acceleration, ϑ the pitch angle, and ϕ the roll angle. The angular speed components $[p\ q\ r]^T$ can be written as

$$\dot{p} = (c_1 r + c_2 p)q + c_3 L + c_4 N, \tag{2a}$$
$$\dot{q} = c_5 pr - c_6(p^2 - r^2) + c_7 M, \tag{2b}$$
$$\dot{r} = (c_8 p - c_2 r)q + c_4 L + c_9 N, \tag{2c}$$

where $[L\ M\ N]^T$ are the roll, pitch and yaw moments, respectively, J_i are the moments of inertia with $i = x, y, z, xz$, and the c_i coefficients formulation is reported in Table I. Furthermore, the aircraft attitude, expressed in terms of Euler angles $[\phi\ \ \vartheta\ \ \psi]^T$, is defined by the following kinematic equations

TABLE I
Formulation of Angular Rate Coefficients in (2)

Parameter	Value
Γ	$J_x J_z - J_{xz}^2$
c_1	$\frac{(J_y - J_z)J_z - J_{xz}^2}{\Gamma}$
c_2	$\frac{(J_x - J_y + J_z)J_{xz}}{\Gamma}$
c_3	$\frac{J_z}{\Gamma}$
c_4	$\frac{J_{xz}}{\Gamma}$
c_5	$\frac{J_z - J_x}{J_y}$
c_6	$\frac{J_{xz}}{J_y}$
c_7	$\frac{}{J_y}$
c_8	$\frac{(J_x - J_y)J_x + J_{xz}^2}{\Gamma}$
c_9	$\frac{J_x}{\Gamma}$

$$\dot{\phi} = p + q\sin\phi\tan\vartheta + r\cos\phi\tan\vartheta, \tag{3a}$$
$$\dot{\vartheta} = q\cos\phi - r\sin\phi, \tag{3b}$$
$$\dot{\psi} = \frac{q\sin\phi}{\cos\vartheta} + \frac{r\cos\phi}{\cos\vartheta}. \tag{3c}$$

The position vector $[x\ y\ h]^T$ is defined in the vehicle-carried vertical reference frame or North-East-Down (NED) frame, as in [24].

B. Linearized Aircraft Model

Starting from the nonlinear model previously introduced, a linearized system of equations in the body axes can be considered for the design of the robust controller, following the guidelines provided in [25], both for straight line flight and WPs transitions at non-zero turn rate. Moreover, the equations of motion linearization procedure results in the decoupling of the longitudinal and lateral-directional planes and each of them can be modeled with standard continuous time-invariant state space representation

$$\dot{x}(t) = Ax(t) + Bu(t), \tag{4}$$

where $x(t)$ is the state vector and $u(t)$ is the control signal. The state and input matrices, A and B respectively, are built according to [26] considering that the equilibrium state of (11) is zero, and the aerodynamic derivatives in the matrices are obtained by a validated software based on the extended lifting-line theory. Reference flight conditions for the model are speed $U_0 = 13.5$ m/s, altitude $h_0 = 100$ m, angle of attack $\alpha_0 \simeq \frac{w}{V} = 5.18$ deg and $\theta_0 = 5.18$ deg. The elements of the matrices are defined in the Appendix VI.

The state variables in the longitudinal plane are the longitudinal component of the total airspeed u, the angle of attack α, the pitch angle θ, the pitch rate q, and the altitude h; the controls are the throttle ΔT acting on u and the elevator deflection δ_e acting on θ. The resulting state space elements are

$$\begin{aligned} &x_{long}(t) = [u,\ \alpha,\ \theta,\ q,\ h]^T \in R^{n_{long}}, \\ &u_{long}(t) = [\Delta T, \delta_e]^T \in R^{m_{long}}, \\ &A_{long} \in R^{n_{long} \times n_{long}}, \\ &B_{long} \in R^{n_{long} \times m_{long}} \end{aligned} \tag{5}$$

with $n_{long} = 5$ and $m_{long} = 2$. The short period mode has natural frequency $\omega_{SP} = 16.6$ rad/s and damping $\zeta_{SP} = 0.49$, while phugoid mode has natural frequency $\omega_{PH} = 0.91$ rad/s and damping $\zeta_{PH} = 0.045$.

The lateral-directional states are the lateral component of the total airspeed v, the roll rate p, the yaw rate r and the roll angle ϕ. the only control is the aileron deflection δ_a acting on ϕ. The lateral-directional state space elements are

$$\begin{aligned} &x_{lat}(t) = [v,\ p,\ r,\ \phi]^T \in R^{n_{lat}}, \\ &u_{lat}(t) = \left[\ \delta_a\ \right] \in R^{m_{lat}}, \\ &A_{lat} \in R^{n_{lat} \times n_{lat}}, \\ &B_{lat} \in R^{n_{lat} \times m_{lat}} \end{aligned} \tag{6}$$

with $n_{lat} = 4$ and $m_{lat} = 1$. The state matrix A_{lat} has one real and negative eigenvalue corresponding to a stable roll mode, one real and positive eigenvalue showing a slightly unstable spiral mode, and a couple of complex conjugate eigenvalues for the Dutch Roll characterized natural frequency $\omega_{DR} = 5.9$ rad/s and damping $\zeta_{DR} = 0.12$. The decoupled linear system can be rewritten considering the two decoupled planes, as follow.

$$\dot{\mathrm{x}}_{long}(t) = \mathrm{A}_{long}\mathrm{x}_{long}(t) + \mathrm{B}_{long}\mathrm{u}_{long}(t), \tag{7a}$$
$$\dot{\mathrm{x}}_{lat}(t) = \mathrm{A}_{lat}\mathrm{x}_{lat}(t) + \mathrm{B}_{lat}\mathrm{u}_{lat}(t). \tag{7b}$$

It is important to highlight that the discretization of the previous continuous-time system dynamics has been obtained exploiting a zero-order hold methodology, where the equilibrium state of (11) is implicitly considered to be zero.

C. Simulation Environment and Processor-in-the-Loop

Autonomous flight of the MH850 aircraft is guaranteed by a custom-made autopilot [8] , with an open architecture re-programmable in flight. The vehicle is equipped with several sensors, including GPS, barometric sensor, differential pressure sensor and three-axis gyros and accelerometers.

The development of a system containing embedded software involves many test activities at different stages in the development process. First, a reliable simulation environment shall be developed, verified and validated via SIL simulations, in which the embedded software is tested within a simulated environment model but without any hardware. Hence, the SIL multi-rate simulator represented in Fig. 2 has been realized in a MATLAB/Simulink environment to perform preliminary validation of the flight software running over an Intel Core i7 $-$ 7500U with a CPU @2.70 GHz, a RAM of 16 GB and a 512 GB solid-state drive. According to the considered scenario, the WPs coordinates block provides the main features of the selected path in terms of WP identification number (ID_{WP}), North-East (NE) coordinates (N_{WP}, E_{WP}) and altitude (h_{WP}). These data, together with the UAV current NE position and heading angle ψ, represent the input for the guidance algorithm that supplies as main outputs the reference velocity V_{ref}, altitude h_{ref}, and heading angle ψ_{ref}. Further details about the guidance algorithm can be found in Section II-D. As previously anticipated, a PID controller is then exploited to control the heading angle, providing the reference roll angle ϕ_{ref} as output. Once defined all the reference signals, it is possible to observe from Fig. 2 that a so-called *rate transition* block, which allows to handle transfer of data between ports operating at different rates as in this specific case. Indeed, if the system dynamics and the guidance algorithm works at 100 Hz, the TRMPC algorithm is updated with a 10 Hz frequency. Thus, the system is fed with the same constant control output for ten consecutive steps, until the control algorithms is run again initialized with new initial conditions in terms of reference signals and current UAV state (x_{long_k}, x_{lat_k}). As highlighted in Fig. 2, the longitudinal TRMPC receives in input the reference velocity and altitude and provides as control output the throttle ΔT and the elevator deflection δ_e whereas the lateral-directional TRMPC receives the reference roll angle to supply the system with the optimal aileron deflection δ_a at each time step k. Further details about the selected control scheme can be found in Section III. Then, the three control outputs are reconditioned to 100 Hz to be fed to the continuous-time nonlinear aircraft model described in Section II-A. Once integrated the nine nonlinear equations, the continuous time longitudinal and lateral-directional states, (u, α, θ, q, h) and (v, p, r, ϕ) respectively, are converted into an output signal with a discrete sample time by a *zero-order hold* block before being fed again to the MPC controller blocks. Moreover, the total airspeed V and the resulting Direction Cosine Matrix (DCM) are provided at the flight data block, in order to obtain the updated UAV coordinates

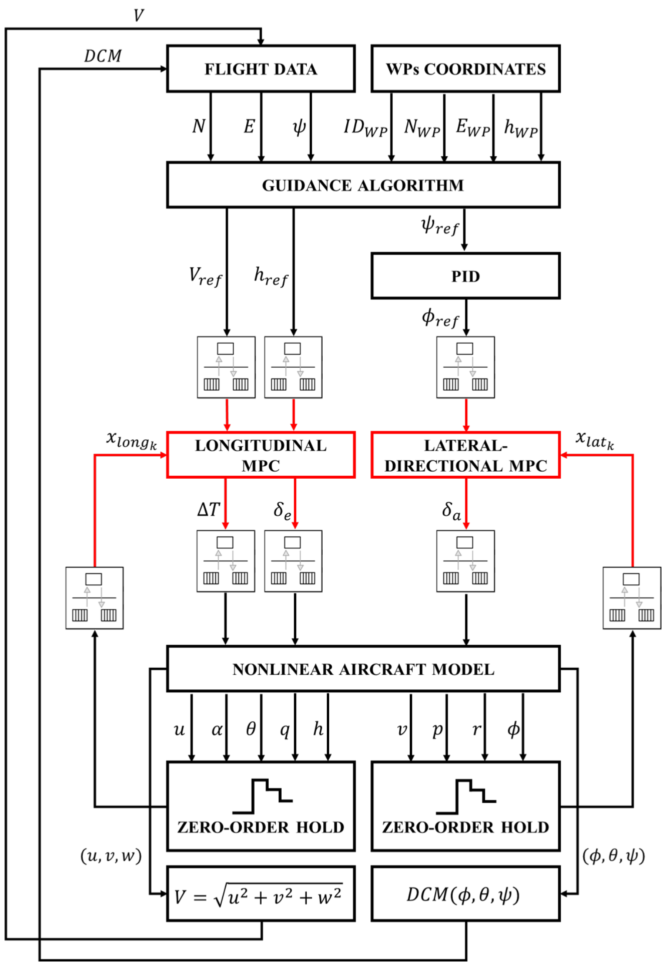


Fig. 2. SIL multi-rate software simulator architecture: (i) black lines @100 Hz; and (ii) red lines @10 Hz.

and attitude, first converting the body cruise speed V into the NED frame, i.e. $V_{NED} = DCM \cdot V$, and then integrating.

The next verification and validation phase consists in performing PIL simulations, cross-compiling and executing the code, which in this case is the control algorithm, on a target processor, following the classical step-by-step procedure described by the V-model for Software Development Life Cycle. Hence, to validate the real-time effectiveness of the proposed controller scheme, PIL simulations [2] have been performed.

In particular, the XMOS XK-1A low-cost commercial board, produced by XMOS Ltd (www.xmos.com), represented in Fig. 3, has been exploited. This board was selected because its features and capabilities, e.g. flash memory of 128 Kb and a CPU clock of 20 MHz, are similar to the MH850 customized autopilot [8], the desired flight mode can be easily implemented, and the board can be connected

[2]Processor-In-the-Loop (PIL) is a test technique that allows designers to evaluate a controller, running in a dedicated processor, of a plant which runs in an offline simulation platform, according to the definition provided in [27].

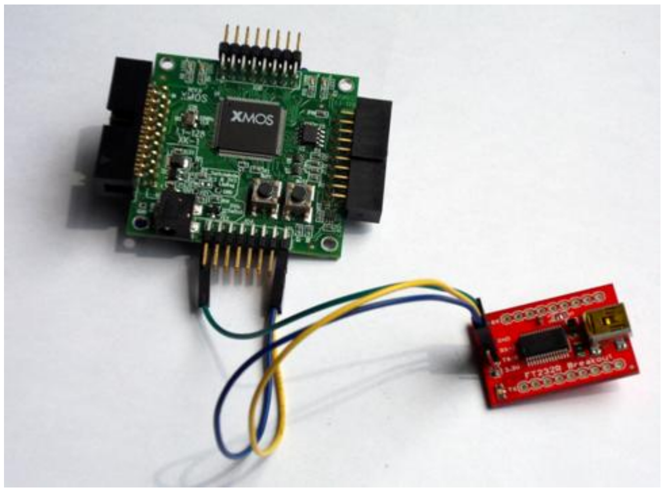


Fig. 3. XMOS development board.

to a laptop via USB. The XMOS board is characterized by the XS1-L1 multi-core multi-thread processor, able to perform several real-time tasks, is easy to program (XC language comparable to C language), and includes some additional commands for the management of ports and pins [28]. Moreover, if a new autopilot is selected, only variation on the communication protocols are necessary. As in [9], a PX4 could be used instead of XMOS board but it was not considered in this case since the performance of the MH850 autopilot are quite different with respect to the ones of a PX4 board. Moreover, the main idea is to compare the performance of the selected controller in the same hardware used for other control laws, see [3], [16]. In detail, different MPC controllers have been compared in [19], again exploiting the XMOS board for PIL validation. As previously stated, the main goal of this research is to demonstrate the effectiveness of the proposed approach in terms of on-board implementation and computational cost, using the same hardware, previously tested with different control laws.

To perform PIL simulations and to validate the computational compatibility among the proposed control scheme and the selected development board, the SIL simulator has been slightly modified, as represented in Fig. 4. First, it is important to observe that the software is running over two different hardware. In particular,the nonlinear dynamics, the guidance algorithm and the PID controller run over a Intel Core $i7-7500U$ PC whereas the two TRMPC schemes are compiled over the XMOS board, which is connected with the laptop via USB. A dedicated communication protocol allows to send the input signal, i.e. the reference velocity, altitude and roll angle and the current UAV states, from the PC to the board and, once solved the optimization problem, the control output are sent back to the laptop via USB cable. Of

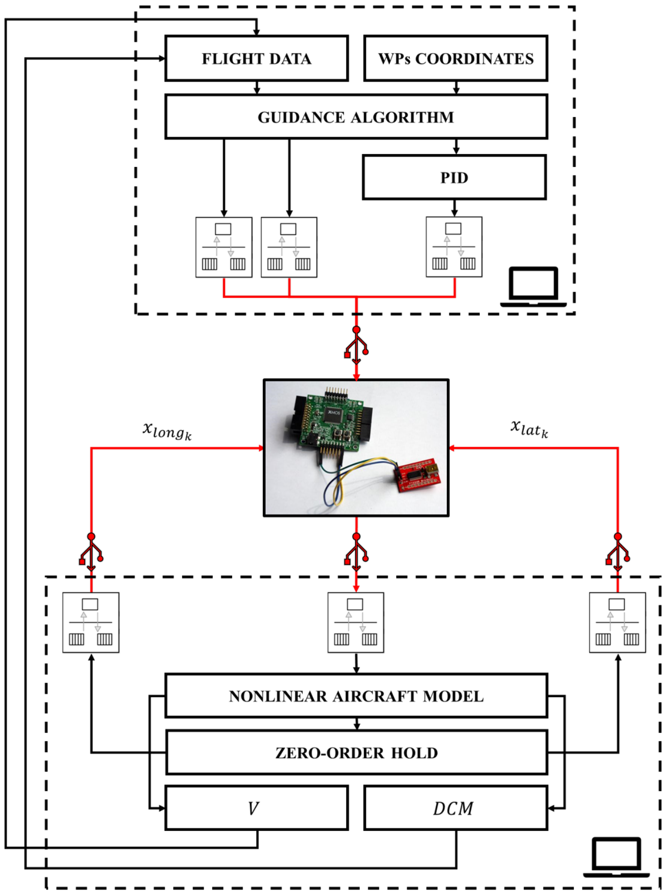


Fig. 4. PIL multi-rate hardware/software architecture: (i) black lines @100 Hz; and (ii) red lines @10 Hz.

course, this external communication introduces some delays[3] that could compromise the controller performance, as shown in Section IV.

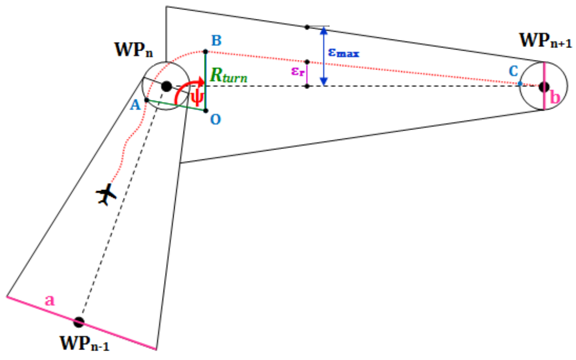
As previously anticipated, the main goal of this work was not only to analyze the effectiveness of the proposed control scheme but also to estimate the computational load required to online solve the optimization problem and the compatibility with the selected development board, and consequently with the MH850 autopilot for future Hardware-In-the-Loop (HIL) validation. Hence, a dedicated block for evaluating the computational load has been added in both SIL and PIL simulators to preliminary estimate the average and maximum execution time and to compare it with the time allocated to online solve the optimization problem, i.e. 0.1 s. Further details can be found in Section IV.

D. Guidance Algorithm

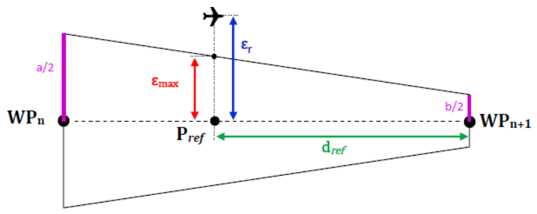
A simplified guidance algorithm, which is computationally efficient and waypoint-based, is proposed, starting from the work [18]. A given set of waypoints is considered, in terms of North (N) and East (E) coordinates. The altitude of each waypoint is the same and fixed during the flight, so a 2D path visualization is considered in the next sections. Some implementation aspects are included:

1) a trajectory smoother, to render the assigned trajectory feasible from a kinematic point-of-view in terms of both speed and turn rate constraint.
2) A cross-track error (CTE) ϵ_r is included as performance index for the aerial mapping capabilities of the guidance algorithm.
3) A look-ahead distance is included. This means that the UAV minimum distance from the following waypoint is defined including a proximity distance, a pre-defined value representing the condition discerning two consecutive waypoints.

The main features of the proposed algorithm can be evaluated from the following two figures.



(a) Guidance phases



(b) CTE and reference distances

Fig. 5. Guidance algorithm scheme.

According to these assumptions, the guidance profile can be divided into three phases, as presented in [18]. The first phase, identified in Fig. 5(a) by the red-dotted line before the point A, is the waypoint approach. The aircraft is flying with fixed velocity at a pre-defined altitude. The waypoint is assumed reached when the vehicle flies into the *imaginary* circle centered in the waypoint WP$_n$. The radius of that circle is set equal to to 20 m, i.e. the defined proximity distance according to the MH850 dynamic constraints.

The second phase is identified by the red dotted A-B curve in Fig. 5(a). In this phase, the FW-UAV turns around the waypoint with velocity profile compliant with the turn rate constraint, function of the speed of the UAV and of the bank angle. It starts when the distance between the UAV and

[3]These communication delays could be also considered as additional disturbance source to be considered during the controller design but in this paper they have not been taken into account since it was out-of-topic.

the waypoint is less or equal to the proximity distance. We assume that this phase ends when $|\psi_{UAV} - \psi_{WP}| < \Delta\psi$, with ψ_{UAV} current heading position of the UAV, ψ_{WP} is the heading angle of the segment that connects the UAV and the next waypoint WP$_{n+1}$. Finally, we assume $\Delta\psi = 5$ deg. Then, the segment B-C represents the last phase in Fig. 5(a). The straight flight begins at the end of the last turn and finishes once reached the next turn circle, following the CTE performance index requirement. The performance index requirement is evaluated as follows

$$\epsilon_r = \frac{|E_{UAV} - mN_{UAV} - (E_n - mN_n)}{\sqrt{m^2+1}}, \tag{8}$$

where the FW-UAV real position P_{UAV} is considered in terms of East and North coordinates, i.e. E_{UAV} and N_{UAV} respectively, while the segment connecting two waypoints in terms of previous waypoint WP$_n(E_n, N_n)$ and next waypoint WP$_{n+1}(E_{n+1}, N_{n+1})$. The coefficient m is equal to

$$m = \frac{E_{n+1} - E_n}{N_{n+1} - N_n}. \tag{9}$$

To avoid continuous corrections of the trajectory, a no correction zone is defined, in which the corrections on the heading angle are considered only when the UAV cross-track error is larger than an assigned value (i.e. maximum acceptable CTE). This region is defined including a major base (a in Figure 5(a)) and a minor base (b in Figure 5(a)). The maximum acceptable CTE ϵ_{max} is variable and decreases

$$\epsilon_{max} = \frac{d_{ref}(\frac{a}{2} - \frac{b}{2})}{d_s}, \tag{10}$$

where $d_{ref} = \sqrt{(N_{n+1} - N_{ref})^2 + (E_{n+1} - E_{ref})^2}$ is the reference distance, d_s is the length of the segment $\overline{WP_{n+1}WP_n}$. N_{ref} and E_{ref} are the coordinates of a reference point, that are evaluated from the reference heading angle ψ_{ref}. If $\epsilon_r > \epsilon_{max}$, a new heading angle is evaluated, as the heading angle of the segment between the UAV position and the next waypoint.

III. Tube-based Model Predictive Control

For the control algorithm design, let us consider the following discrete time-invariant state-space system in which persistent disturbances w$_k$ are included

$$\mathrm{x}_{k+1} = \mathrm{A}_d\mathrm{x}_k + \mathrm{B}_d\mathrm{u}_k + \mathrm{w}_k, \tag{11}$$

where x$_k$ and u$_k$ represent the discrete-time state vector and the control signal at time k, respectively. [4]

Let us assume that the system is required to satisfy hard constraints on both state and input

$$\mathrm{x}_k \in \mathbb{X}, \quad \mathrm{u}_k \in \mathbb{U}, \tag{12}$$

where $\mathbb{X} \subset \mathbb{R}^n$ and $\mathbb{U} \subset \mathbb{R}^m$ are compact and convex polytopes. For the definition of the disturbance, w$_k$ is considered as a realization of a stochastic process, an independent and identically distributed (i.i.d.) zero-mean random variable, with a convex and bounded support $\mathbb{W} \subset \mathbb{R}^n$ containing the origin.

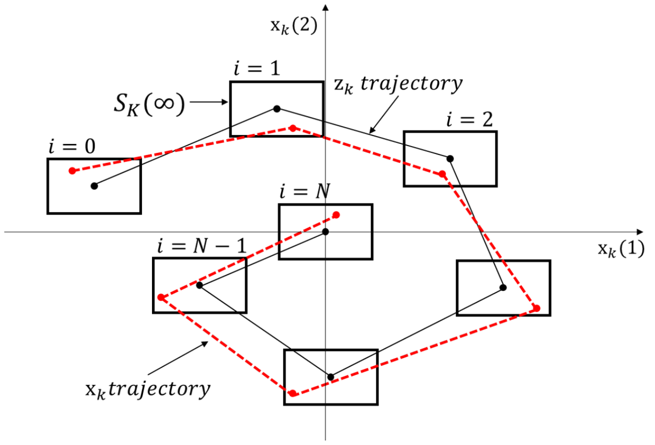


Fig. 6. Outer-bounding tube representation at the k-th time step over a prediction horizon of N.

The TRMPC approach is based on the concept of *tube* of state trajectories, each one representing an admissible disturbance sequence w over the observed time-window. The center of this tube corresponds to the nominal undisturbed trajectory, which dynamics is defined as

$$\mathrm{z}_{k+1} = \mathrm{A}_d\mathrm{z}_k + \mathrm{B}_d\mathrm{v}_k, \tag{13}$$

where z$_k$ and v$_k$ are the discrete-time nominal state and input, respectively. Fig. 6 provides a representation of the outer-bounding tube at the k-th time step centered on the nominal trajectory at each i-th step over a N prediction horizon.

The TRMPC allows to steer the uncertain trajectories to the nominal one, controlling the "center" of this tube via a classical MPC approach. In order to ensure the robustness of the algorithm, the constraint set imposed on the nominal system are tightened with respect to the initial ones in Eq. (12), following the guidelines provided in [15]. Then, exploiting the following time-varying feedback control law related to the i-th step ahead k

$$\mathrm{u}_{i|k} = \mathrm{v}_{i|k} + K(\mathrm{x}_{i|k} - \mathrm{z}_{i|k}), \tag{14}$$

where K is defined such that $\mathrm{A}_K = \mathrm{A}_d + B_dK$ is robustly stable, the closed-loop dynamics can be rewritten as follows

$$\mathrm{x}_{i+1|k} = (\mathrm{A}_d + \mathrm{B}_dK)\mathrm{x}_{i|k} + \mathrm{B}_d\mathrm{v}_{i|k} + \mathrm{w}_{i|k}. \tag{15}$$

Moreover, to stabilize the system with respect to parametric uncertainty q, ascribable for example to discrepancies between the mathematical model and the actual dynamics, neglected non-linearities and manufacturing process, a Linear Matrix Inequality (LMI) approach applied to the definition of

[4]With respect to the continuous time-invariant dynamics equations introduced in previous Section, $x_k \in \mathbb{R}^{11}$ and $u_k \in \mathbb{R}^3$. Splitting the aircraft dynamics into the longitudinal and lateral-directional planes, the discrete-time systems considered have the following dimensions: (i) longitudinal: $x_{k_{long}} \in \mathbb{R}^5$ and $u_{k_{long}} \in \mathbb{R}^2$; and (ii) lateral-directional: $x_{k_{long}} \in \mathbb{R}^4$ and $u_{k_{long}} \in \mathbb{R}$.

the closed-loop system Schur stability. Given any $\mathrm{x}_k \in \mathbb{X}_N$, where $\mathbb{X}_N$ defines the terminal state constraint set, it exists a $P \in \mathbb{R}^{n\times n}$, $P \succ 0$ such that

$$(\mathrm{A}_d + \mathrm{B}_d K)^T P (\mathrm{A}_d + \mathrm{B}_d K) + Q + K^T R K - P \preceq 0, \tag{16}$$

and the feedback gain matrix K quadratically stabilizes the system (15) with respect to the parametric uncertainty q. $Q \in \mathbb{R}^{n\times n}$, $Q \succ 0$, and $R \in \mathbb{R}^{m\times m}$, $R \succ 0$ are diagonal positive definite matrices. As proposed in [29], the LMI approach is applied to the well-known Edge Theorem, which guarantees the robust stability of a polytope of polynomials if and only if all exposed edge polynomial are stable. Let us assume that the uncertain vector $\mathrm{q} = [\mathrm{q}_1, ..., \mathrm{q}_l]$ is bounded in the hyper-rectangle B_q defined as

$$B_\mathrm{q} := \left\{ \mathrm{q} \in \mathbb{R}^l \mid \mathrm{q}_j \in [\mathrm{q}_j^-, \mathrm{q}_j^+], j = 1, ..., l \right\}. \tag{17}$$

and, consequently, let us define the following edge-uncertain system matrices $\mathrm{A}_d^- = \mathrm{A}_d(\mathrm{q}^-)$, $\mathrm{A}_d^+ = \mathrm{A}_d(\mathrm{q}^+)$, $\mathrm{B}_d^- = \mathrm{B}_d(\mathrm{q}^-)$, and $\mathrm{B}_d^+ = \mathrm{B}_d(\mathrm{q}^+)$. Then, solving the following LMIs system allows to obtain the feedback gain matrix K that satisfies

$$\begin{cases} Q + K^T R K + (\mathrm{A}_d^+ + \mathrm{B}_d^+ K)^T \tilde{P} (\mathrm{A}_d^+ + \mathrm{B}_d^+ K) - \tilde{P} \preceq 0, \\ Q + K^T R K + (\mathrm{A}_d^+ + \mathrm{B}_d^- K)^T \tilde{P} (\mathrm{A}_d^+ + \mathrm{B}_d^- K) - \tilde{P} \preceq 0, \\ Q + K^T R K + (\mathrm{A}_d^- + \mathrm{B}_d^+ K)^T \tilde{P} (\mathrm{A}_d^- + \mathrm{B}_d^+ K) - \tilde{P} \preceq 0, \\ Q + K^T R K + (\mathrm{A}_d^- + \mathrm{B}_d^- K)^T \tilde{P} (\mathrm{A}_d^- + \mathrm{B}_d^- K) - \tilde{P} \preceq 0, \end{cases} \tag{18}$$

and stabilizes the system (11) with respect to $\mathrm{q} \in B_\mathrm{q}$.

Since the system dynamics includes an unknown but bounded disturbance $\mathrm{w}_{i|k}$, it is possible to split the state $\mathrm{x}_{i|k}$ as

$$\mathrm{x}_{i|k} = \mathrm{z}_{i|k} + \mathrm{e}_{i|k}, \tag{19}$$

where $\mathrm{e}_{i|k}$ represents the deviation of the actual state $\mathrm{x}_{i|k}$ with respect to the nominal one $\mathrm{z}_{i|k}$ i step ahead time k. Thus, substituting (19) into (15), the error dynamics is described by

$$\mathrm{e}_{i+1|k} = (\mathrm{A}_d + \mathrm{B}_d K)\mathrm{e}_{i|k} + \mathrm{w}_{i|k}. \tag{20}$$

As anticipated before, tightened constraint sets shall considered for the nominal system, properly designed starting from an outer approximation of the minimal Robust Positive Invariant (mRPI) set for (20)

$$S_K(\infty) \doteq \sum_{j=0}^{\infty} \mathrm{A}_K^j \mathbb{W}, \tag{21}$$

in compliance with the guidelines provided in [15] and according to the following Definitions.

Definition 1 (Robust Positive Invariant set): Given the set $S \subseteq \mathbb{X}$ is said to be the Robust Positive Invariant (RPI) set if, for all $e_0 \in S$ and for any $w_k \in \mathbb{W}$, the condition $e_k \in S$ holds $\forall k \in \mathbb{N}_{\geq 0}$ [30].

Definition 2 (minimal Robust Positive Invariant set): The minimal Robust Positive Invariant (mRPI) set under (20) is the RPI set contained in every closed RPI set of (20) [14].

At this point, it is important to highlight that the set in (21) is the mRPI set for (20) because *only* additive disturbance has been considered affecting the system dynamics as in (11). Indeed, parametric uncertainty has been considered *only* for evaluating the feedback gain matrix K that quadratically stabilizes the closed-loop disturbed dynamics but no uncertainty has been included in the control design.

Hence, if the time-invariant control law (14) is employed and the nominal system (13) satisfies the tightened constraint sets

$$\begin{aligned} \mathrm{z}_{i|k} &\in \mathbb{Z} \subseteq \mathbb{X} \ominus S_K(\infty), \\ \mathrm{v}_{i|k} &\in \mathbb{V} \subseteq \mathbb{U} \ominus K S_K(\infty). \end{aligned} \tag{22}$$

the initial constraints $\mathrm{x}_{i|k} \in \mathbb{X}$ and $\mathrm{u}_{i|k} \in \mathbb{U}$ are robustly satisfied at each time step k. This assertion makes sens only if the disturbance set $\mathbb{W}$ is sufficiently small to ensure that the following Assumption holds, as supposed in the sequel.

Assumption 1 (Restricted Disturbances for Constraint Satisfaction): $S_K(\infty) \subseteq \mathbb{X}$ and $K S_K(\infty) \subseteq \mathbb{U}$ [15].

To easily compute the tightened constraint sets to be enforced in the optimization problem, the approach proposed in [15] has been followed and here briefly recalled.

Starting from the definition of the state constraint set and considering the state decomposition in (19) with where $e_{i|k} \in S_K(\infty)$, it follows that $H_x x_{i|k} \leq h_x$ if

$$H_x z_{i|k} \leq h_x - \Phi_\infty, \tag{23}$$

with $\Phi_\infty = \max\limits_{e_{i|k}} \left\{ H_x e_{i|k} \mid e_{i|k} \in S_K(\infty) \right\}$. Thus,

$$\hat{\mathbb{Z}} = \left\{ z_{i|k} \in \mathbb{R}^n \mid H_x z_{i|k} \leq h_x - \Phi_\infty \right\} \tag{24}$$

represents a suitable constraint set for the nominal state $z_{i|k}$ in order to obtain an inner approximation $\mathbb{Z}$ of $\hat{\mathbb{Z}}$, where $\hat{\mathbb{Z}} = \mathbb{X} \ominus S_K(\infty)$. To evaluate Φ_∞, it is possible to compute an upper bound of this set solving a simple linear programming. Indeed, if it holds that

$$A_K^T w_k \in \beta \mathbb{W}, \quad \forall w_k \in \mathbb{W}, \tag{25}$$

with $\beta \in (0, 1)$, then it follows that $\Phi_\infty \leq (1-\beta)^{-1} \Phi_T$ where

$$\Phi_T = \max_{w_{i|k}} \left\{ H_x \sum_{j=0}^{T-1} A_K^j \mathbb{W}, w_{i|k} \in \mathbb{W} \right\}, \tag{26}$$

is the solution of a linear programming problem. Hence, it is possible to obtain an upper bound of Φ_∞ properly selecting

β as close as desired to 1. Then, the constraint set $\mathbb{Z}$ can be defined by

$$\mathbb{Z} \doteq \left\{z_{\ell|k} \in \mathbb{R}^n \mid H_x z_{\ell|k} \leq h_x - (1-\beta)^{-1}\Phi_T\right\} \subseteq \hat{\mathbb{Z}}. \tag{27}$$

Analogously, the constraint set on the control input $\mathbb{V}$ can be approximated as

$$\mathbb{V} \doteq \left\{v_{i|k} \in \mathbb{R}^m \mid H_u v_{i|k} \leq h_u - (1-\beta)^{-1}K\Phi_T\right\} \subseteq \hat{\mathbb{V}}. \tag{28}$$

starting from the initial control input constraint set $\mathbb{U}$ and being $\hat{\mathbb{V}} = \mathbb{U} \ominus KS_K(\infty)$. Further details on the design process of the tightened constraint sets can be found in [15] and [29].

Then, the finite horizon optimal quadratic cost can be defined for the nominal dynamics in terms of nominal state $\mathrm{z}_{i|k}$ and nominal control input $\mathrm{v}_{i|k}$ as

$$J_N(\mathrm{z}_k, \mathbf{v}_k) = \sum_{i=0}^{N-1} (\mathrm{z}_{i|k}^T Q \mathrm{z}_{i|k} + \mathrm{v}_{i|k}^T R \mathrm{v}_{i|k}) + \mathrm{z}_{N|k}^T P \mathrm{z}_{N|k}, \tag{29}$$

where $\mathbf{v}_k$ represents the control sequence over a N-step prediction horizon. $P \in \mathbb{R}^{n\times n}$ is the solution of the discrete Algebraic Riccati equation [31]. Thus, the nominal finite horizon optimal control problem can be stated as follows

$$\min_{\mathbf{v}_k} \quad J_N(\mathrm{z}_k, \mathbf{v}_k) \tag{30a}$$

$$\begin{aligned} \text{s.t.} \quad & \mathrm{z}_{i+1|k} = \mathrm{A}_d \mathrm{z}_{i|k} + \mathrm{B}_d \mathrm{v}_{i|k}, \quad \mathrm{z}_{0|k} = \mathrm{x}_k, \\ & \mathrm{z}_{i|k} \in \mathbb{Z}, \quad i \in [1, N-1], \\ & \mathrm{v}_{i|k} \in \mathbb{V}, \quad i \in [0, N-1], \\ & \mathrm{z}_{N|k} \in \mathbb{Z}_N, \end{aligned} \tag{30b}$$

with $\mathbb{Z}_N \subseteq \mathbb{X}_N \ominus S_K(\infty)$. The first control action $\mathrm{v}_{0|k}^*$ of the optimal sequence $\mathbf{v}_k^*$, solution of (30), represents the optimal control applied to the nominal system while the correspondent control on the uncertain system is defined according to (14). The final TRMPC algorithm can be summarized as shown in Algorithm 1. It is important to clarify that, as already shown in Section II-C, the step 9 of Algorithm 1 is only valid in theory while in the case study presented in this paper, the control action is directly applied to the true model of the UAV instead of the simplified model used for control design.

IV. Simulation Results

The guidance and TRMPC control algorithms described in the previous Sections have been applied to control a FW-UAV, whose system nonlinear and linearized dynamics are reported in Section II and Section VI, respectively. As previously introduced, the TRMPC scheme has been exploited to control both longitudinal and lateral-directional dynamics, following the reference signals in terms of airspeed u_{ref}, altitude h_{ref} and roll angle ϕ_{ref} provided by the guidance algorithm. In particular, the TRMPC provides the control actions in terms of throttle , and aileron and elevator

Algorithm 1 TRMPC Algorithm

```
1:  procedure
2:      Offline: Evaluate the feedback gain matrix $K$ and the nominal constraint sets $\mathbb{Z}$ (27) and $\mathbb{V}$ (28).
3:      Online: Initialization: for $k = 0$, set $z_{0|k} = x_k = x_0$.
4:      At current time $k$, evaluate $x_k, z_k$.
5:      for $i = 0 : N-1$ do
6:          Solve (30)
7:      end for
8:      Get $\mathbf{v}_0^*$ and extract the first control action $v_0^*$.
9:      Evaluate $u_k$ according to (14).
10:     Evaluate $z_{k+1}$ applying $\mathbf{v}_0^*$ on (13) and $x_{k+1}$ applying $u_k$ on (11).
11: end procedure
```

deflections, δ_a and δ_e respectively, while a PID controller is used for the heading angle ψ to obtain ϕ_{ref} with respect to ψ_{ref}, function of the identified waypoints, as described in Section II-D.

In this Section, first SIL results are provided to validate the efficacy of the proposed guidance and control approach in a simulation environment, selecting a pre-defined butterfly path (see Fig. 7) as test case. On the other hand,

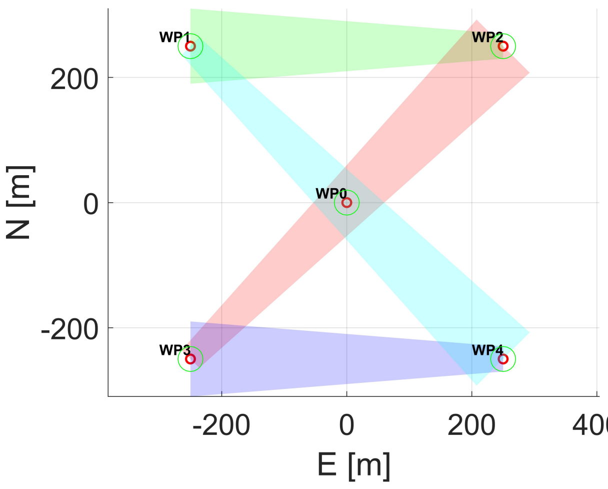


Fig. 7. Reference butterfly path.

the real-time implementability of the control algorithm has been demonstrated via PIL simulations, exploiting: (i) a MATLAB/Simulink simulator running over an Intel Core i7 − 7500U with a CPU @2.70 GHz, a RAM of 16 GB and a 512 GB solid-state drive, for the nonlinear system dynamics and the guidance algorithm; and (ii) the XMOS board, described in Section II-C and connected to the laptop via USB, for running the controller. Moreover, it is important to highlight that the quadratic programming solver *quadwright* proposed in [32], based on the optimization algorithm proposed in [33], has been selected to solve the online optimization problem since it was developed with a focus on efficient memory use, ease of implementation, and high speed convergence. Indeed, as shown in [32] and also proved in [34], the *quadwright* solver results the most performing when dealing with high-dimension problems, low-memory boards and fast dynamics. Thus, the high-efficiency solver allows to have computational loads similar among SIL and PIL simulations, as shown later (see Fig. 25 and Fig. 26), despite the significant difference among the computational capabilities among the SIL hardware and the XMOS board.

For the selected case study, the following initial flight conditions have been set: (i) $V_0 = 13.5$ m/s, (ii) $h_0 = 100$ m, (iii) $\alpha_0 = 5.18$ deg, (iv) $\gamma = 0$ deg, i.e. $\theta_0 = 5.18$ deg. For the definition of the guidance parameters, the turn radius is evaluated equal to $R = 22$ m and a maximum bank angle $\phi = 40$ deg is considered. The major base of the isosceles trapezoid of Figure 5(a) (segment a) is set equal to 70 m. Instead the minor base (segment b) corresponds to the diameter of the circle around the waypoint and it is set equal to 40 m. These parameters are used for the definition of the maximum cross-track error of Eq. (10).

The MPC parameters have been set uniformly within all the scenarios as well as the sample times: (i) system dynamics 0.01 s; (ii) TRMPC sample time = 0.1 s. The other MPC parameters are reported in Table II. Moreover, the robustly stabilizing matrices K_{long} and K_{lat} have been evaluated *offline*, exploiting typical robust tools, and their values are reported in Table II.

TABLE II
TRMPC PARAMETERS.

Parameter	Value
N_{long}	15
N_{lat}	30
$diag(Q_{long})$	$[10^6, 4\times10^1, 4\times10^1, 4\times10^1, 10^5]$
$diag(R_{long})$	$[4\times10^2, 3\times10^{-6}]$
K_{long}	$\begin{bmatrix} -9.6439 & 2.3128 & 0.9501 & 0.0752 & -2.8072 \\ 0.0115 & -1.5403 & 2.0980 & 0.0399 & 0.6683 \end{bmatrix}$
$diag(Q_{lat})$	$[10^1, 10^1, 10^1, 10^4]$
R_{lat}	10^4
K_{lat}	$[0.0462 \;\; -0.2879 \;\; -0.0321 \;\; -0.3576]$

A fixed-direction wind turbulence, modeled as random noise with uniform distribution and maximum intensity of ±1 m/s, represented a bounded persistent disturbance affecting the FW-UAV dynamics. Moreover, additional external noises have been included as affecting the other states, in analogy to those exploited in [19], and their values are reported in Table III.

On the other hand, the uncertainties considered exclusively for the offline evaluation of the feedback gain matrix K include a $\pm15\%$ variation of the following parameters, in addition to those parametric uncertainties ascribable to neglected nonlinearities: (i) cruise speed, envisioning flexibility to different flight conditions; (ii) vehicle mass, considering the possibility to exploit the same controller for slightly different vehicles belonging to the same fleet; (iii) FW-UAV inertia, due to manufacturing process. Hence, the following uncertainty sources have been included: (i) q_V for the cruise speed; (ii) q_m and q_I for the UAV mass and inertia, respectively; and (iii) q_{NL} due to neglected nonlinearities. Thus, the hyper-rectangle B_q is defined as

TABLE III
ADDITIVE DISTURBANCES CONSIDERED.

LONG. disturbance	Disturbance value	LAT.-DIR. disturbance	Disturbance value
d_u [m/s]	1	d_v [m/s]	1
d_α [rad]	10^{-2}	d_p [rad/s]	10^{-2}
d_θ [rad]	10^{-2}	d_r [rad/s]	10^{-2}
d_q [rad/s]	10^{-2}	d_ϕ [rad]	10^{-2}
d_h [m]	10^{-1}		

$$B_{\mathrm{q}} := \left\{ \mathrm{q} = \begin{bmatrix} \mathrm{q}_V \\ \mathrm{q}_m \\ \mathrm{q}_I \\ \mathrm{q}_{NL} \end{bmatrix} \in \mathbb{R}^4 \mid \mathrm{q} \in \begin{bmatrix} -2.025, 2.025 \\ -0.15, 0.15 \\ -0.0033, 0.0033 \\ -0.05, 0.05 \end{bmatrix} \right\}, \tag{31}$$

considering a $\pm5\%$ of uncertainty due to neglected nonlinearities. Thus, the state matrices A_{long} and A_{lat} have the following form

$$A_{long} = A_{long_0} + A^V_{long_q} + A^m_{long_q} + A^I_{long_q} + \mathrm{q}_{NL} A_{long_0}, \tag{32a}$$

$$A_{lat} = A_{lat_0} + A^V_{lat_q} + A^m_{lat_q} + A^I_{lat_q} + \mathrm{q}_{NL} A_{lat_0}, \tag{32b}$$

where the $A^\epsilon_{long_q}$ and $A^\epsilon_{lat_q}$ are the uncertain matrices related to cruise speed ($\epsilon = V$), mass ($\epsilon = m$), and inertia ($\epsilon = I$) while A_{long_0} and A_{lat_0} coincides with those *nominal* in (33) and (34), respectively (see also Appendix).

TABLE IV
INITIAL AND NOMINAL STATE AND INPUT CONSTRAINT BOUNDARIES FOR THE LONGITUDINAL DYNAMICS.

Variable	Initial MIN value	Initial MAX Value	Tightened MIN value	Tightened MAX value
u [m/s]	12	15	12.33	14.67
α [rad]	0.0698	0.1396	0.0865	0.1229
θ [rad]	0.0698	0.1396	0.0865	0.1229
q [rad/s]	-10	10	-9.9833	9.9833
h [m]	99	101	99.17	100.83
ΔT [-]	0	1	0.1	0.83
δ_e [rad]	-0.3491	0.3491	-0.3157	0.3157

In compliance with the assumption made in Section III about the definition of $\mathbb{X}$ and $\mathbb{U}$ as convex polytopes, the state and input constraints related to the specific case under analysis have been defined as linear inequalities in which each parameter is bounded among its minimum and maximum admissible values, e.g. $\eta_{min} \leq \eta \leq \eta_{max}$. Consequently, the tightened constraint sets $\mathbb{Z}$ and $\mathbb{V}$ result to be convex polytope as well. The aforementioned ranges are reported in Table IV for the longitudinal variables and in Table V for the lateral-directional variables. Both initial and tightened

TABLE V
INITIAL AND NOMINAL STATE AND INPUT CONSTRAINT BOUNDARIES FOR THE LATERAL-DIRECTIONAL DYNAMICS.

Variable	Initial MIN value	Initial MAX Value	Tightened MIN value	Tightened MAX value
v [m/s]	-1	1	-0.873	0.873
p [rad/s]	-10	10	-9.9833	9.9833
r [rad/s]	-10	10	-9.9833	9.9833
ϕ [rad]	-2π	2π	-6.27	6.27
δ_a [rad]	-0.3491	0.3491	-0.3324	0.3324

polytopes have been exemplifying represented in terms of airspeed components (u, v) and altitude h in Fig. 8 and in terms of control inputs $(\Delta T, \delta_e, \delta_a)$ in Fig. 9.

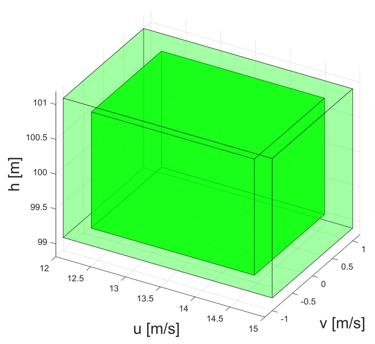


Fig. 8. Initial and tightened constraint set for airspeed components u and v and altitude h.

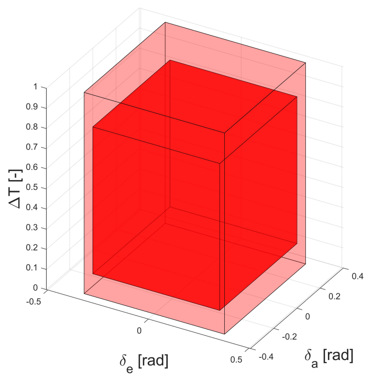


Fig. 9. Initial and tightened constraint set for control inputs, i.e. throttle ΔT, elevator deflection δ_e and aileron deflection δ_a.

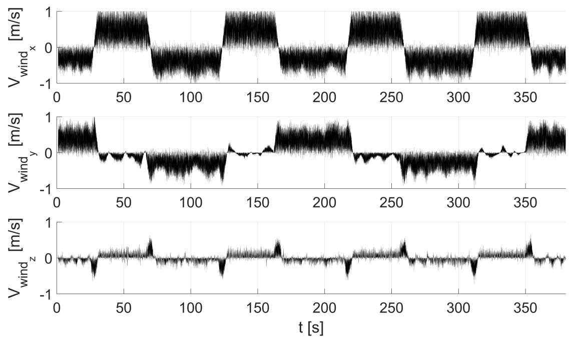


Fig. 10. Wind turbulence profile in the body reference frame.

A. Software- and Processor-In-the-Loop Results

As anticipated before, a butterfly path, defined by four waypoints plus a central one and envisioned for patrolling tasks over an area of interest, such as industrial warehouses and criminal neighborhood, has been analyzed to preliminary assess the effectiveness of the proposed guidance and control scheme here proposed and to preliminary validate their real-time implementability with SIL and PIL simulations.

The first step consisted in performing 5 different runs and the main results are shown in Fig. 11-Fig. 17. Starting from the velocity component u and v and the altitude represented in Fig. 11, it is possible to observe that the controller is able to track the reference signals $u_{ref} = 13.5$ m/s and $h_{ref} = 100$ m while guaranteeing constraint satisfaction also in the presence of a wind turbulence (see Fig. 10 for the wind turbulence profile).

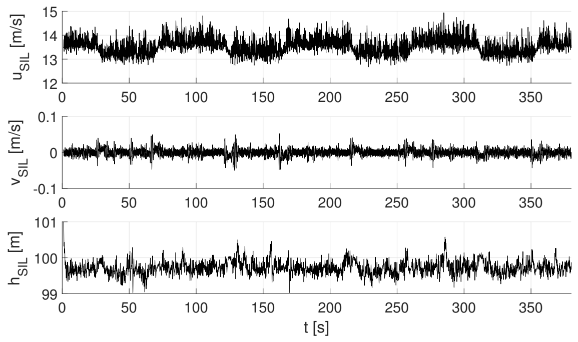


Fig. 11. SIL velocity components and altitude.

Indeed, the aircraft velocity is always within the given boundary (see Table IV) even if the the effect of disturbance can still be observed, mainly when the UAV is flying against wind. On the other hand, the lateral-directional component of the velocity and the altitude result less affected by the wind turbulence and they linger significantly close to their target values. In particular, the v component remain close to zero while the altitude averagely stays about 0.5 m far from the reference because of the persistent $-z$ wind turbulence component.

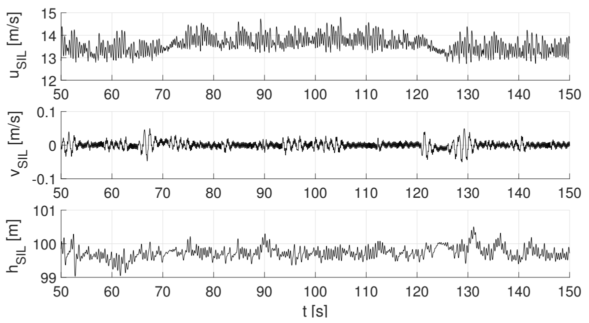


Fig. 12. Zoom-in on SIL velocity components and altitude.

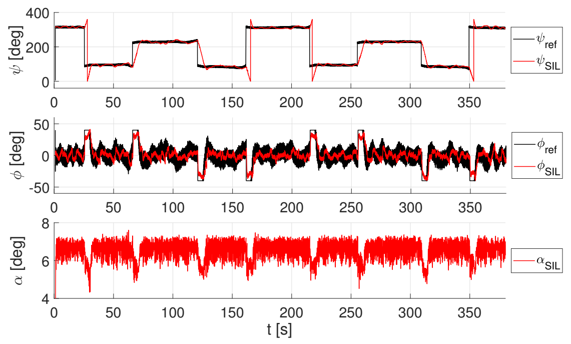


Fig. 13. SIL roll angle, heading angle and angle of attack.

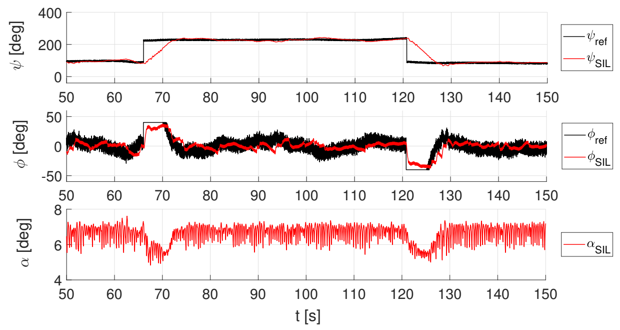


Fig. 14. Zoom-in on SIL roll angle, heading angle and angle of attack.

Fig. 13 provides an overview of the TRMPC capabilities for tracking the roll angle reference signal ϕ_{ref} provided by the PID controller, which in turn is in charge of following the heading angle reference signal ψ_{ref} fed by the guidance algorithm. In either cases, the tracking capabilities of both controllers are validated. In particular, even if the roll reference signal results quite noisy, the lateral-directional TRMPC is able to follow it mainly during turns (see ± 45 deg peaks in Fig. 13). Correspondingly, the angle of attack remains rather constant, with the exception of the turning phases, and in analogy to the altitude, it departs from the initial value of 5.18 deg because of the wind effect.

The TRMPC (and PID) tracking capabilities are also confirmed by Fig. 15 where the tracking errors are depicted with respect to longitudinal airspeed, altitude, and roll and

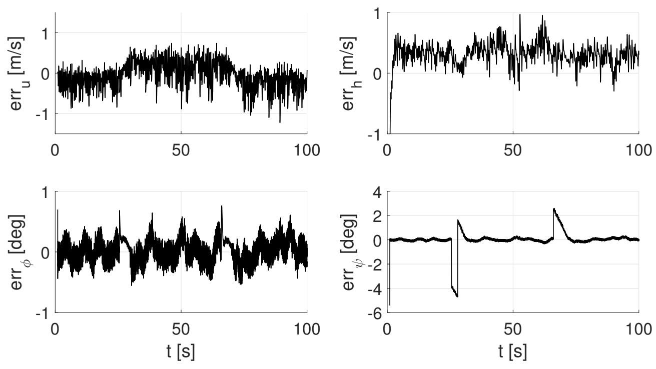


Fig. 15. SIL tracking errors for longitudinal velocity, altitude, and roll and heading angles.

heading angles[5]. In particular, the UAV airspeed and altitude deviations reflect the wind turbulence entity, i.e. it is in the order of about 1 m/s and 0.5 m, respectively. On the other

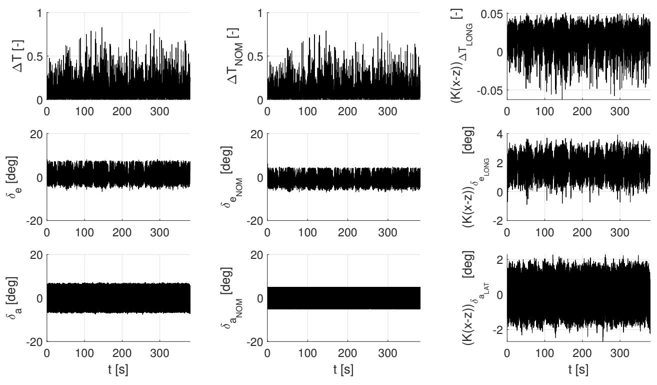


Fig. 16. SIL throttle and elevator and aileron deflections: (i) real (1st column); (ii) nominal (2nd column); and (iii) error component (3rd column).

hand, the effectiveness of the TRMPC and PID schemes for the attitude tracking is highlighted in the bottom subplots, where on one side the roll angle deviations are due to the noisy reference signal while the heading angle tracking is mainly affected during turn phases because of the PID response rate.

From the control input point-of-view, Fig. 16 provides an overview of the control demand during the entire maneuver, also highlighting the differences among the real applied control (first column), the optimal nominal control (second column), and the $K(x-z)$ control input component, which is representative of the discrepancies among the current state and the undisturbed one.

The last plot related to the SIL testing, i.e. Fig. 17, shows the UAV trajectories for all 5 runs, highlighting the capability of TRMPC to allow the aircraft to follow the given butterfly path while remaining in the coloured corridors defined by the guidance algorithm. It is possible to observe also that,

[5]The tracking errors have been reported only for the first 100 s of simulation to better highlight their profiles, which remains pretty constant for the entire mission

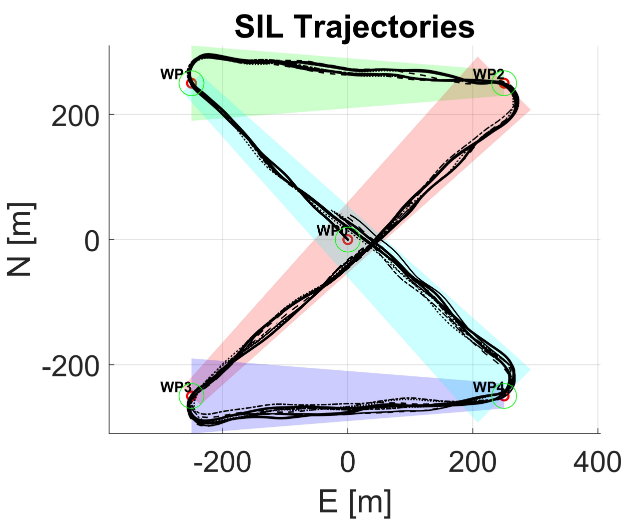


Fig. 17. SIL trajectories.

because of the presence of a wind turbulence, the trajectories do not result too smooth and they are all different due to the random nature of the disturbance itself.

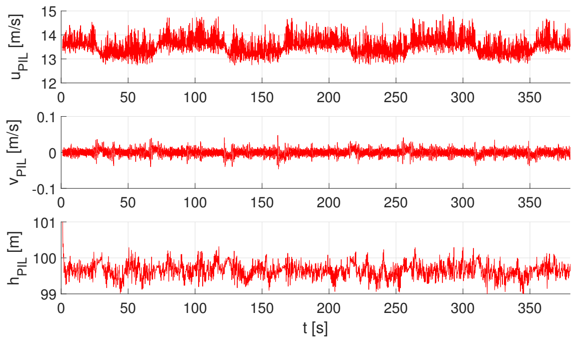


Fig. 18. PIL velocity components and altitude.

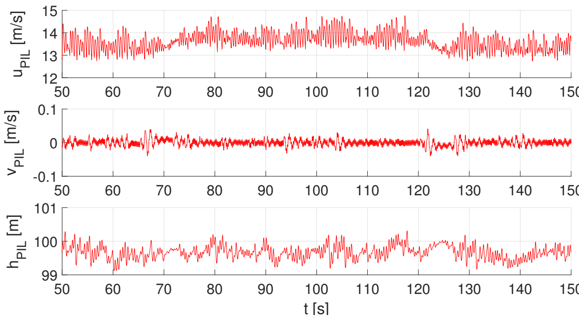


Fig. 19. Zoom-in on PIL velocity components and altitude.

On the other hand, Fig. 18-Fig. 24 represent the main outputs obtained during PIL testing. It is possible to observe the significant adherence among SIL and PIL results, thus highlighting the reliability of the simulation environment and the effectiveness of the control scheme also exploiting less performing hardware, i.e. XMOS board, than the SIL one for running the controllers.

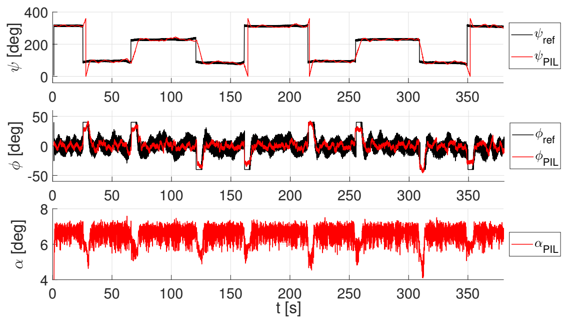


Fig. 20. PIL roll angle, heading angle and angle of attack.

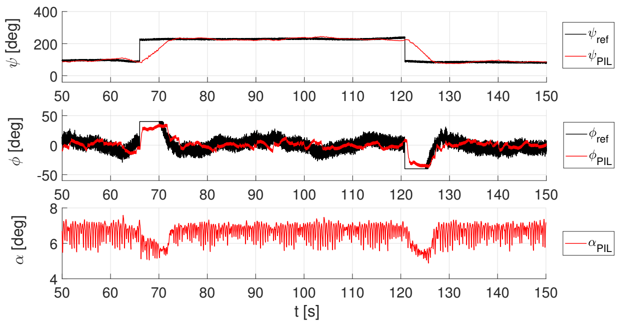


Fig. 21. Zoom-in on PIL roll angle, heading angle and angle of attack.

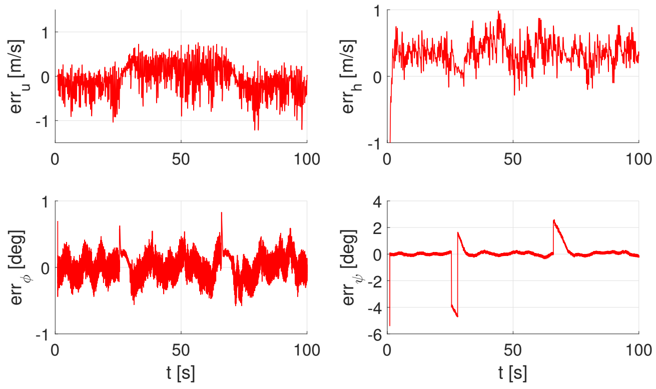


Fig. 22. PIL tracking errors for longitudinal velocity, altitude, and roll and heading angles.

Also for the 5 PIL trajectories, represented in Fig. 24, it is possible to observe the effect of random wind turbulence on the UAV profiles, which do not compromise the effectiveness of the TRMPC, confirmed by the fact that not only the aircraft always remains within the pre-defined boundaries but all mission and control constraints are satisfied, as shown in previous pictures.

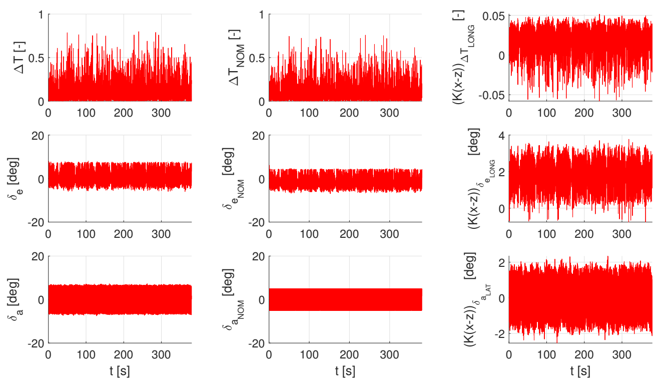


Fig. 23. PIL throttle and elevator and aileron deflections: (i) real (1st column); (ii) nominal (2nd column); and (iii) error component (3rd column).

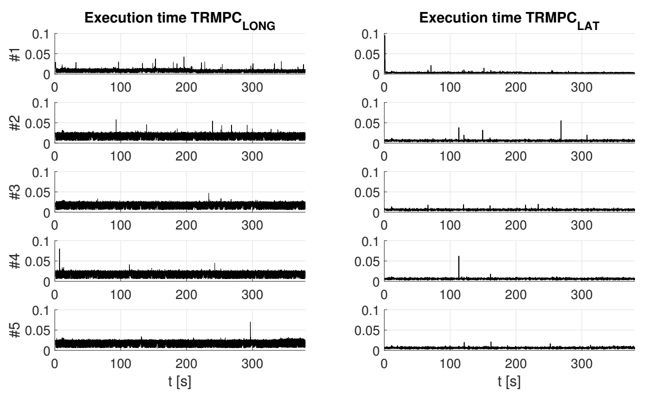


Fig. 25. SIL execution time [s].

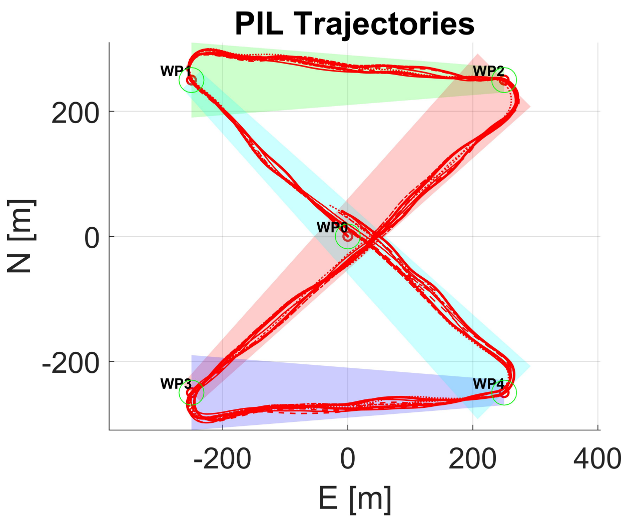


Fig. 24. PIL trajectories.

To preliminary assess the real-time compatibility of the proposed guidance and control approach with the capabilities of the target hardware, i.e. the XMOS development board and for analogy the MH850 autopilot, the execution time required by both longitudinal and lateral-directional TRMPC schemes have been evaluated during SIL and PIL simulations. The execution time has been evaluated within the control routine exploiting the *tic/toc* MATLAB function, which allows to estimate the elapsed time that occurs to execute a certain amount of *operations* [6].

Fig. 25 and Fig. 26 depict the execution time for all 5 SIL and PIL runs, respectively, over the entire simulation time. The results that the execution/wall time is much lower than the controller rate, i.e. 0.1 s. Table VI provides the average execution time values for both longitudinal (LONG) and lateral-directional (LAT-DIR) controllers.

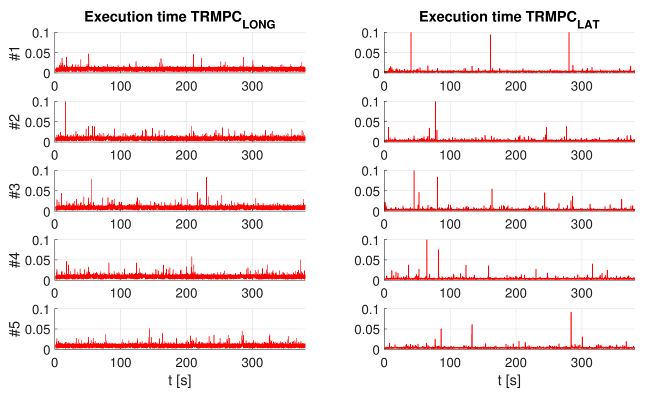


Fig. 26. PIL execution time [s].

Moreover, the random distribution of the over-rate peaks in both SIL and PIL simulations highlights that straight-line flight and turning phase have similar computational demand. However, it shall be highlighted that because the same linearized model has been exploited for both flight conditions, the performance of the controller could also be affected by this second issue during turns since the tracking task results more challenging than during straight-line flight and the impact of neglected nonlinearities is more relevant. On the other hand, it is possible to notice that a slightly higher computation load characterizes the PIL simulations and this behavior could be ascribed to the communication delay observed during PIL simulations. Indeed, with respect to SIL testing, the presence of an external microcontroller, over which the TRMPC is running and connected to the main simulator hardware by a USB cable, introduces some external delays that could affect the results and the computational compatibility.

Last, the effects of varying disturbance magnitude and prediction horizon over both computational load and communication delay have been investigated. In particular, the following test cases have been considered, with the other

[6]See also Mathworks *tic* and *toc* reference pages at mathworks.com/help/matlab/ref/tic.html and mathworks.com/help/matlab/ref/toc.html, respectively

TABLE VI
PIL AVERAGE/MAXIMUM EXECUTION TIME.

PIL ID	LONG Average Execution Time [s]	LAT-DIR Average Execution Time [s]
#1	0.0092	0.0036
#2	0.0099	0.0034
#3	0.0098	0.0035
#4	0.0098	0.0035
#5	0.0099	0.0034

parameters considered fixed: (i) half the wind turbulence intensity, i.e. $V_w = 0.5$ m/s; (ii) double the wind turbulence intensity, i.e. $V_w = 2$ m/s; (iii) a third the prediction horizons, i.e. $N_{long} = 5$ and $N_{lat} = 10$; and (iv) double the prediction horizons, i.e. $N_{long} = 30$ and $N_{lat} = 60$.

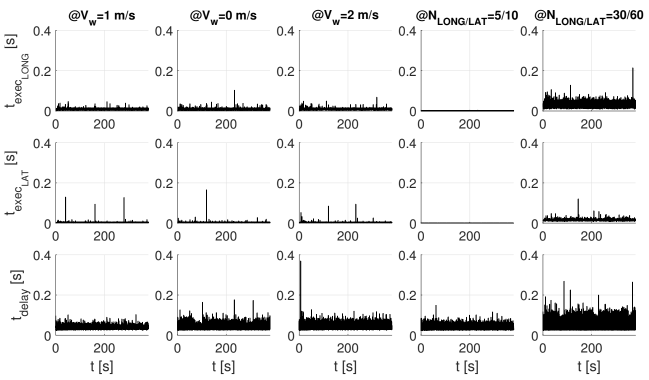


Fig. 27. Effects of varying disturbance and prediction horizon on PIL execution time and XMOS communication delay.

The results shown in Fig. 27 highlights that either increasing or decreasing the wind intensity does not affect the computational cost of the algorithm but only the tightened constraint sets definition. Instead, the variation of prediction horizon significantly impact on the computational load required to solve online the optimization problem, as shown in the last two columns of Fig. 27 related to the execution time t_{exec}. On the other hand, the results related to the communication delay t_{delay} confirm that this disturbance is independent from the controller design and the simulation setup but is simply inherited from the USB cable connection and it should disappear once HIL testing on the real MH850 autopilot will be performed, once the entire system will be simulated on the same hardware[7].

V. Conclusions

In this paper a Tube-based Robust Model Predictive Control (TRMPC) is proposed to control both the longitudinal and lateral-directional dynamics of a Fixed-Wing Unmanned Aerial Vehicle (FW-UAV) envisioning path-tracking mission. The proposed control strategy, combined with a waypoint-based guidance algorithm and a classical Proportional-Integral-Derivative (PID) control for the heading angle, has been validated via Software-In-the-Loop (SIL) and Processor-In-the-Loop (PIL) simulations, considering a butterfly-like path, which resembles a typical patrolling pattern. The results have shown that the TRMPC allows to robustly satisfy the mission tracking constraints during both SIL and PIL testing even in the presence of a wind turbulence disturbance. Moreover, the simulations have highlighted a significant adherence among SIL and PIL results, validating also the reliability of the simulation environment.

Real-time implementability has been preliminary validated analysing the execution time required by the controller tasks and comparing it with the time allocated for solving the optimal control problem. Furthermore, the effects of increasing/decreasing values of both disturbances and prediction horizon on the computational load have been investigated, and the results highlighted that enlarging the prediction horizon implies a higher computational demand, unlike increasing the disturbance magnitude. Last, communication delays, introduced by the USB cable that connects the simulator hardware (i.e. laptop) with the XMOS development board, were noted but without significantly compromising the controller performance. Furthermore, it was shown that this delay is intrinsic to the PIL architecture since it was not affected by changes on neither disturbance magnitude nor prediction horizon.

Once validated the efficacy and real-time implementability of the proposed control scheme via Hardware-In-the-Loop (HIL) up to system level and flight tests, envisioning simpler scenarios, future investigations will mainly focus on two directions. On one side, more challenging applications involving significant changes in the UAV dynamics will be considered, e.g. involving issues related to damaged airframe, icing or payload change that otherwise require gain scheduling. On the other hand, envisioning the emerging need to provide the same capabilities exploiting multiple simpler, lighter and cheaper vehicles instead of heavier, more expensive and complicated ones, we will focus on the control of a fleet of similar UAVs, devoted to autonomously and coordinately cooperate to fulfill a common task whether in the military or civilian fields, e.g. patrolling or harvesting scenarios. The main idea is to implement the same robust controller on each vehicle of the fleet, extending the approach proposed to explicitly include also parametric uncertainties that could rise from manufacturing processes and that make each vehicle unique. In this way, the fleet management will require the design of only two different controllers: (i) a low-level controller for the single-vehicle task, equally applied on each vehicle; and (ii) a high-level controller for the distributed trajectory generation strategy, i.e. a fleet formation controller. Eventually, a more performing microcontroller will be selected and equipped on the MH850, in line with the

[7]It is important to point out that different communication delays with respect to the one here considered could be observed exploiting different hardware or performing HIL simulations.

actual hardware equipped on board mini-UAV, e.g. with a i5 or i7 microprocessor, and a second verification and validation campaign will be performed.

VI. Appendix: State-Space Matrix Evaluation

If a complete system is considered, i.e. no decoupling between the longitudinal and lateral-directional planes is considered, the state vector for the system studied in Subsection II-D is defined as $x = [u\ v\ \alpha\ p\ q\ r\ \phi\ \theta\ \psi\ h]^T$. As detailed in Section II-A, the linearized system, here considered, decouple the longitudinal and lateral-directional plane. The state variables in the longitudinal plane are the longitudinal component of the total airspeed u, the angle of attack α, the pitch angle θ, the pitch rate q, and the altitude h. The controls are the throttle ΔT acting on u and the elevator deflection δ_e acting on θ. The resulting state space elements are

$$\begin{aligned} x_{lon}(t) &= [u,\ \alpha,\ \theta,\ q,\ h]^T \in R^{n_{lon}}, \\ u_{lon}(t) &= [\Delta T, \delta_e]^T \in R^{m_{lon}}, \\ A_{long} &\in R^{n_{long} \times n_{long}}, \\ B_{long} &\in R^{n_{long} \times m_{long}} \end{aligned} \tag{33}$$

with $n_{lon} = 5$ and $m_{lon} = 2$. The state matrix in the logitudinal plane can be defined as follows

$$A_{long} = \begin{bmatrix} X_u & X_\alpha & -g\cos\theta_0 & 0 & 0 \\ \frac{Z_u}{U_0 - Z_{\dot\alpha}} & \frac{Z_\alpha}{U_0 - Z_{\dot\alpha}} & \frac{-g\sin\theta_0}{U_0 - Z_{\dot\alpha}} & \frac{Z_q + U_0}{U_0 - Z_{\dot\alpha}} & 0 \\ 0 & 0 & 1 & 0 & 0 \\ M_u + \frac{M_{\dot\alpha} Z_u}{U_0 - Z_{\dot\alpha}} & M_\alpha + \frac{M_{\dot\alpha} Z_\alpha}{U_0 - Z_{\dot\alpha}} & \frac{-g\sin\theta_0 M_{\dot\alpha}}{U_0 - Z_{\dot\alpha}} & M_q + \frac{M_{\dot\alpha}(Z_q + U_0)}{U_0 - Z_{\dot\alpha}} & 0 \\ 0 & -U_0 & U_0 & 0 & 0 \end{bmatrix}.$$

The aerodynamic derivatives are defined as function of the flight conditions and of the aircraft mass, of the airfoil parameters, of the propeller.

In the analyzed case, dimensional aerodynamic derivatives are considered, so we have

$$\begin{aligned} X_u &= \tfrac{\rho S U_0}{2m}(-3C_{D_e} - C_{D_u}) \\ X_w &= \tfrac{\rho S U_0}{2m}(C_{L_e} - C_{D_\alpha}) \\ Z_u &= \tfrac{\rho S U_0}{2m}(-2C_{L_e} - C_{L_u}) \\ Z_w &= \tfrac{\rho S U_0}{2m}(-C_{L_\alpha} - C_{D_e}) \\ Z_q &= \tfrac{\rho S U_0 c}{4m}(-C_{L_q}) \\ M_u &= \tfrac{\rho S U_0 c}{2J_y}(C_{m_u}) \\ M_w &= \tfrac{\rho S U_0 c}{2J_y}(C_{m_\alpha}) \\ M_q &= \tfrac{\rho S U_0 c^2}{4J_y}(C_{m_q}) \end{aligned}$$

$C_{X_u} = C_{T_u} = -3C_{D_e} - C_{D_u}$, $C_{Leq} = \frac{2W/S}{\rho V^2} = 0.392$ rad^{-1} is the lift coefficient in equilibrium ($W = mg$ is the weight of the UAV, ρ is the air density), $C_{Deq} = C_{D0} + kC_{Leq}^2 = 0.029$ rad^{-1} and $C_{D\alpha} = 2kC_{L\alpha}C_{Leq} = 0.469$ rad^{-1} are function of C_{Leq}, $C_{Tu} = -3C_{Deq}$ Moreover, $X_\alpha = X_w U_0$ and $Z_\alpha = Z_w U_0$. Usually, $Z_{\dot\alpha} = 0$ and $M_{\dot\alpha} = 0$. Finally, C_{mu}, C_{Du} and C_{Lu} are zero for UAVs (i.e. subsonic aircraft) since no aeroelastic effects are considered. The other derivatives are defined in Table VII.

For the lateral-directional plane, the states are the lateral component of the total airspeed v, the roll rate p, the yaw rate r and the roll angle ϕ. the only control is the aileron deflection δ_a acting on ϕ. The lateral-directional state space elements are

$$\begin{aligned} x_{lat}(t) &= [v,\ p,\ r,\ \phi]^T \in R^{n_{lat}}, \\ u_{lat}(t) &= [\ \delta_a\ ] \in R^{m_{lat}}, \\ A_{lat} &\in R^{n_{lat} \times n_{lat}}, \\ B_{lat} &\in R^{n_{lat} \times m_{lat}} \end{aligned} \tag{34}$$

with $n_{lat} = 4$ and $m_{lat} = 1$.

$$A_{lat} = \begin{bmatrix} Y_v & Y_p & Y_r - U_0 & 0 \\ L_v & L_p & L_r & 0 \\ N_v & N_p & N_r & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

The dimensional derivatives are defined as

$$\begin{aligned} Y_v &= \tfrac{\rho S U_0}{2m}(C_{Y_\beta}) \\ Y_p &= \tfrac{\rho S U_0 b}{4m}(C_{Y_p}) \\ Y_r &= \tfrac{\rho S U_0 b}{4m}(C_{Y_r}) \\ L_v &= \tfrac{\rho S U_0 b}{2J_X}(C_{l_\beta}) \\ L_p &= \tfrac{\rho S U_0 b^2}{4J_X}(C_{l_p}) \\ L_r &= \tfrac{\rho S U_0 b^2}{4J_X}(C_{l_r}) \\ N_v &= \tfrac{\rho S U_0 b}{2J_z}(C_{n_\beta}) \\ N_p &= \tfrac{\rho S U_0 b^2}{4J_z}(C_{n_p}) \\ N_r &= \tfrac{\rho S U_0 b^2}{4J_z}(C_{n_r}) \end{aligned}$$

The control matrices can be written in the following way.

$$B_{long} = \begin{bmatrix} 1 & 0 \\ 0 & \frac{Z_{\delta_e}}{U_0 - Z_{\dot\alpha}} \\ 0 & 0 \\ 0 & M_{\delta_e} + \frac{M_{\dot\alpha} Z_{\delta_e}}{U_0 - Z_{\dot\alpha}} \end{bmatrix},$$

where $Z_{\delta e} = \frac{\rho S U_0^2}{2m}(-C_{L_{\delta_e}})$ and $M_{\delta e} = \frac{\rho S U_0^2 c}{2I_y}(C_{m_{\delta_e}})$. For the lateral-directional plane, the only control input is the aileron deflection δ_a

$$B_{lat} = \begin{bmatrix} Y_{\delta_a} \\ L_{\delta_a} \\ N_{\delta_a} \\ 0 \end{bmatrix},$$

where $Y_{\delta_a} = \frac{\rho S U_0^2}{2m}(-C_{Y_{\delta_a}})$, $L_{\delta_a} = \frac{\rho S U_0^2}{2J_x}(C_{l_{\delta_a}})$ and $N_{\delta_a} = \frac{\rho S U_0^2}{2J_z}(C_{n_{\delta_a}})$. The other derivatives are defined in Table VII.

For the uncertain matrices in (32), their formulation can be derived explicitly introducing the bounds of uncertain parameters $q_V^\pm$, $q_m^\pm$ and $q_I^\pm$ into the dimensional aerodynamic derivatives, thus obtaining the corresponding uncertain matrices to be exploited for LMI system.

TABLE VII
AERODYNAMIC DERIVATIVES OF THE FW-UAV.

Parameter [rad^{-1}]	Value
$C_{L\alpha}$	3.186
$C_{m\alpha}$	-0.524
$C_{L_{\dot{\alpha}}}$	0
$C_{m_{\dot{\alpha}}}$	0
C_{mq}	-1.375
C_{Y_β}	0
C_{Y_p}	-0.031
C_{Y_r}	0.069
C_{l_p}	-0.1192
C_{n_p}	0.0122
C_{l_β}	-0.201
C_{n_β}	0.031
C_{l_r}	0.024
C_{n_r}	-0.021
$C_{L_{\delta_e}}$	-0.895
$C_{m_{\delta_e}}$	-0.695
$C_{Y_{\delta_a}}$	0
$C_{l_{\delta_a}}$	0.153
$C_{n_{\delta_a}}$	0.012

REFERENCES

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

[2] R. GarcIa, F. Rubio, and M. Ortega, "Robust pid control of the quadrotor helicopter," *IFAC Proceedings Volumes*, vol. 45, no. 3, pp. 229 – 234, 2012.

[3] E. Capello, D. Sartori, G. Guglieri, and F. Quagliotti, "Robust assessment for the design of multi-loop proportional integrative derivative autopilot," *IET Control Theory Applications*, vol. 6, no. 11, pp. 1610–1619, 2012.

[4] J. López, R. Dormido, S. Dormido, and J. Gómez, "A robust h controller for an uav flight control system," *Scientific World Journal*, vol. 2015, no. 6, 2015.

[5] A. Jafar, S. Fasih-UR-Rehman, S. Fazal-UR-Rehman, and N. Ahmed, "H infinity controller for unmanned aerial vehicle against atmospheric turbulence," *Scientific World Journal*, vol. 11, no. 4, 2016.

[6] B. Bialy, J. Klotz, K. Brink, and W. Dixon, "Lyapunov-based robust adaptive control of a quadrotor uav in the presence of modeling uncertainties," in *2013 American Control Conference*, 2013, pp. 13–18.

[7] Z. Dydek, A. Annaswamy, and E. Lavretsky, "Adaptive control of quadrotor uavs: A design trade study with flight evaluations," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 4, pp. 1400–1406, 2013.

[8] E. Capello, G. Guglieri, P. Marguerettaz, and F. Quagliotti, "Preliminary assessment of flying and handling qualities for mini-uavs," *Journal of Intelligent & Robotic Systems*, vol. 65, no. 1, pp. 43–61, 2012.

[9] P. Oettershagen, A. Melzer, S. Leutenegger, K. Alexis, and R. Siegwart, "Explicit model predictive control and l 1-navigation strategies for fixed-wing uav path tracking," in *22nd Mediterranean Conference on Control and Automation*. IEEE, 2014, pp. 1159–1165.

[10] T. J. Stastny, A. Dash, and R. Siegwart, "Nonlinear mpc for fixed-wing uav trajectory tracking: Implementation and flight experiments," in *AIAA Guidance, Navigation, and Control Conference*, 2017, p. 1512.

[11] M. Kamel, T. Stastny, K. Alexis, and R. Siegwart, *Model Predictive Control for Trajectory Tracking of Unmanned Aerial Vehicles Using Robot Operating System*. Springer International Publishing, 2017, pp. 3–39.

[12] K. Alexis, C. Papachristos, R. Siegwart, and A. Tzes, "Robust model predictive flight control of unmanned rotorcrafts," *Journal of Intelligent & Robotic Systems*, vol. 81, no. 3, pp. 443–469, 2016.

[13] N. Michel, S. Bertrand, G. Valmorbida, S. Olaru, and D. Dumur, "Design and parameter tuning of a robust model predictive controller for uavs," in *20th IFAC World Congress*, 2017.

[14] B. Kouvaritakis and M. Cannon, *Model Predictive Control: Classical, Robust and Stochastic*. Advanced Textbooks in Control and Signal Processing, Springer, 2015.

[15] D. Mayne and J. Rawlings, *Model Predictive Control: Theory and Design*. Nob Hill Publishing, 2009.

[16] M. Mammarella and E. Capello, "A robust mpc-based autopilot for mini uavs," in *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2018, pp. 1227–1235.

[17] E. Capello, G. Guglieri, F. Quagliotti, and D. Sartori, "Design and validation of an l1 adaptive controller for mini-uav autopilot," *Journal of Intelligent & Robotic Systems*, vol. 69, no. 1, pp. 109–118, 2013.

[18] E. Capello, G. Guglieri, and G. Ristorto, "Guidance and control algorithms for mini-UAV autopilots," *Aircraft Engineering and Aerospace Technology*, vol. 89, no. 1, pp. 133–144, 2017.

[19] M. Mammarella, E. Capello, F. Dabbene, and G. Guglieri, "Sample-based smpc for tracking control of fixed-wing uav," *IEEE Control Systems Letters*, 2018.

[20] M. Mammarella, E. Capello, H. Park, and M. Guglieri, G. Romano, "Spacecraft proximity operations via tube-based robust model predictive control with additive disturbances," 2017.

[21] T. Erkkinen and M. Conrad, "Verification, validation, and test with model-based design," SAE Technical Paper, Tech. Rep., 2008.

[22] E. Capello, G. Guglieri, P. Marguerettaz, and F. Quagliotti, "Preliminary assessment of flying and handling qualities for mini-uavs," *Journal of Intelligent & Robotic Systems*, vol. 65, no. 1, pp. 43–61, 2012.

[23] E. Capello, G. Guglieri, and F. Quagliotti, "A software tool for mission design and autopilot integration: an application to micro aerial vehicles," in *Proceedings of the 2008 Summer Computer Simulation Conference*. Society for Modeling & Simulation International, 2008, p. 9.

[24] B. Etkin and L. Reid, *Dynamics of Flight: Stability and Control*. New York: John Wiley and Sons, 1996.

[25] C. Casarosa, *Meccanica del volo*. Plus-Pisa University Press, 2004, vol. 1.

[26] B. Stevens and F. Lewis, *Aircraft Control and Simulation*. New York: John Wiley and Sons, 2003.

[27] J. Mina, Z. Flores, E. López, A. Pérez, and J.-H. Calleja, "Processor-in-the-loop and hardware-in-the-loop simulation of electric systems based in fpga," in *2016 13th International Conference on Power Electronics (CIEP)*. IEEE, 2016, pp. 172–177.

[28] G. Martins, A. Moses, M. Rutherford, and K. Valavanis, "Enabling intelligent unmanned vehicles through xmos technology," *The Journal of Defense Modeling and Simulation*, vol. 9, no. 1, pp. 71–82, 2012.

[29] M. Mammarella, E. Capello, H. Park, G. Guglieri, and M. Romano, "Tube-based robust model predictive control for spacecraft proximity operations in the presence of persistent disturbance," *Aerospace Science and Technology*, 2018.

[30] F. Blanchini and S. Miani, *Set-theoretic methods in control*. Springer, 2008.

[31] A. A. Stoorvogel and A. Saberi, "The discrete algebraic riccati equation and linear matrix inequality," *Linear algebra and its applications*, vol. 274, no. 1-3, pp. 317–365, 1998.

[32] J. Currie, A. Prince-Pike, and D. Wilson, "Auto-code generation for fast embedded model predictive controllers," in *Proc. of International Conference Mechatronics and Machine Vision in Practice*, 2012.

[33] S. Wright, "Applying new optimization algorithms to model predictive control," Tech. Rep., 1996.

[34] M. Mammarella, M. Lorenzen, E. Capello, H. Park, F. Dabbene, G. Guglieri, M. Romano, and F. Allgöwer, "An offline-sampling smpc framework with application to autonomous space maneuvers," *IEEE Transactions on Control Systems Technology*, 2018.

[35] V. Dobrokhodov, I. Kaminer, I. Kitsios, E. Xargay, C. Cao, I. Gregory, N. Hovakimyan, and L. Valavani, "Experimental validation of l1 adaptive control: The rohrs counterexample in flight," *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 5, pp. 1311–1328, 2011.

[36] G. Vallabha. (2016) Real-time pacer for simulink. [Online]. Available: https://it.mathworks.com/matlabcentral/fileexchange/29107-real-time-pacer-for-simulink

[37] E. Capello, G. Guglieri, and G. Ristorto, "Guidance and control algorithms for mini uav autopilots," *Aircraft Engineering and Aerospace Technology*, vol. 89, no. 1, pp. 133–144, 2017.

[38] L. Novaro Mascarello, F. Quagliotti, and G. Ristorto, "A feasibility study of an harmless tiltrotor for smart farming applications," in *Unmanned Aircraft Systems (ICUAS), 2017 International Conference on*. IEEE, 2017, pp. 1631–1639.

[39] G. Padfield, *Helicopter Flight Dynamics: The Theory and Application of Flying Qualities and Simulation Modeling*. American Institute of Aeronautics and Astronautics, 1996.

[40] G. Vallabha, *Real-Time Pacer MATLAB/Simulink Toolbox (ver. 1.0.0.1)*, 2017.

[41] D. Hrovat, S. Di Cairano, H. E. Tseng, and I. V. Kolmanovsky, "The development of model predictive control in automotive industry: A survey," in *2012 IEEE International Conference on Control Applications*. IEEE, 2012, pp. 295–302.