



Design and real-time implementation of a wireless autopilot using multivariable predictive generalized minimum variance control in the state-space

Antonio Silveira^{a,*}, Anderson Silva^b, Antonio Coelho^c, José Real^a, Orlando Silva^a

^a Laboratory of Control and Systems, Federal University of Pará, Belém, 66075-110, Brazil

^b Federal Institute of Education, Science and Technology of Pará, Parauapebas, 68515-000, Brazil

^c Department of Automation and Systems, Federal University of Santa Catarina, Florianópolis, 88040-900, Brazil

ARTICLE INFO

Article history:

Received 6 October 2019
Received in revised form 10 June 2020
Accepted 25 June 2020
Available online 3 July 2020
Communicated by Hever Moncayo

Keywords:

Unmanned aerial vehicle
Autopilot
Minimum variance control
Kalman filter
Predictive control

ABSTRACT

The contribution of this work is the numerical simulation and the experimental assessment of a network distributed control system using an unmanned aerial vehicle and a remote master controller connected by a wireless network. A novel multi-input multi-output long-range predictive horizon minimum variance control approach is developed and implemented in altitude, heading, lateral and longitudinal velocities control problems under the influence of process noise, measurement noise and time-delay. The proposed autopilots exhibited damped and fast response speeds with optimal control and output variance minimization, outperforming a classic model predictive control approach in convincing experimental field tests in real-world environments and scenarios.

© 2020 Elsevier Masson SAS. All rights reserved.

1. Introduction

In this work it is uniquely investigated the application of the generalized minimum variance in the state-space method (GMVSS) for multi-input multi-output system (MIMO) to design altitude-hold, heading-hold, lateral and longitudinal speed-hold autopilots for a quadrotor, where the autopilot and the quadrotor are connected over a wireless network and are assessed in experimental field tests in real-world environments.

This kind of distributed control system counts on embedded stability and control augmentation systems aboard the quadrotors, but the required maneuvers come from a remote guidance and navigation system [1–3]. In some of these applications the framework is based on user multicast datagram protocol over wireless networks [4], configuring a stochastic scenario by means of possible network packet dropouts [5] together with control and sensor transmission latency and noise [6].

To overcome these problems, model predictive control (MPC) algorithms are gaining more space in the aerospace industry and academia [7]. Quadrotor autopilot design and real-time implemen-

tation of such algorithms are coping with increased computational power of microprocessors and efficient programming of embedded control systems using certain design softwares, such as MATLAB[®] and Simulink[®] with automatic code generation [8,9]. The benefits of MPCs are being extended for compensators commonly designed for aerospace systems, such as lead, lag, proportional-integral-derivative controllers using hybrid topologies to increase the accuracy of satellite-carrier boosters landing point, for turbofan engine control, for satellite control, trajectory optimization and obstacle avoidance, and so on [10–14].

Despite this increase in MPC applied to aerospace systems, in most cases, the drawback is the computational burden and the algorithms are still being investigated in theory and/or simulation. Even though, MPCs capable of dealing with stochastic Gaussian uncertainties, noise and time delays, explicitly, will play a pivotal role in a number of aspects across aerospace engineering [7], and in this state of flux, quadrotors are becoming interesting test beds for MPC and other advanced control algorithms, since they allow the assessment of aerospace systems concepts in a simple and safe manner, such as in [15–20].

To the best of authors' knowledge, the GMVSS is being proposed and assessed for the first time as a MIMO control system design technique and in quadrotor flight control systems. It derives from the minimum variance regulator proposed by Karl Johan Åström in 1970 [21] that culminated in two well-known industry-

* Corresponding author.

E-mail addresses: asilveira@ufpa.br (A. Silveira), anderson.silva@ifpa.edu.br (A. Silva), antonio.arc@ufsc.br (A. Coelho), real@ufpa.br (J. Real), orfosi@ufpa.br (O. Silva).

standard MPCs: generalized minimum variance control [22] and generalized predictive control (GPC) [23]. These are stochastic predictive control techniques since they explicitly account for the stochasticity of a controlled auto-regressive (integrated) moving average model (i.e., with colored noise). As predictive minimum variance control methods, they estimate future stochastic disturbances to counteract before they occur.

Despite the benefits regarding stochastic MPC, practitioners were lured by the strong potential of long-range prediction to increase robustness and stability margins of control-loops [24], such that the stochastic approach in MPC design is not yet a common practice in industrial control systems and linear stochastic predictive control methods are still open for investigation [25]. In the aerospace field, such control systems are in transit from simulations [26] to implementations in experimental laboratory essays [9].

The GMVSS has been assessed previously in industrial process control, outperforming traditional predictive control methods in the linear [27] and non-linear [28] cases. Its major differences to classic MPC are: it is a non-receding-horizon control technique, uses only two tuning parameters and does not require the solution of Diophantine equations as in generalized predictive control. The negative aspect, however, is the dependence of stochastic model estimation techniques such as state-space extended (recursive) least-squares and Kalman filter or observer/Kalman filter identification methods in order to fully exploit optimal results in the minimum variance sense [21]. Such increased model complexity, unfortunately, also increases the computational burden beyond classical stochastic MPCs, thus not providing any improvements to this problem.

Several advanced control methods, gaining more attention in quadrotor autopilot design, also have a strong dependence on estimation techniques, such as artificial neural networks to aid sliding mode control [29,19,16]. However, they lack real flight experiments and a reshape to discrete-time systems, whilst in some cases, the control signals are not even shown or discussed [30], even though the problem with control signal chattering is well-known to be seen in such control approaches in the regulatory case.

Minimum variance control gained popularity after the seminal work of Åström and Wittenmark in 1973 on self-tuning regulators [31]. The minimum variance was achieved in the regulatory control case, despite an unfeasible control signal in practice, due to excessive chattering and unrealizable amplitudes. Such problems were solved after detuned cases appeared and in [27] it has been shown that GMVSS could recover the ideal theoretical result, by means of long-range prediction, with a physically realizable control signal.

All previous works regarding GMVSS control have reported applications to single-input, single-output (SISO) systems. Thus, it still lacks the mathematical development and the analysis in MIMO control system design and application for decentralized and centralized MIMO control syntheses.

The non-adaptive MIMO GMVSS control is proposed and assessed in this present work for quadrotor altitude-hold, compass heading-hold and lateral/longitudinal speed-hold autopilots, in order to optimally counteract to stochastic disturbances and achieve fast transitory response speeds with minimum variance control effort. Numerical simulations and experimental results are compared to an industrial-standard MPC technique.

Beyond this introductory part, this work is organized as follows: the GMVSS method for a class of MIMO systems is presented; the multi-input multi-output linear discrete-time model of the quadrotor in the state-space is derived; simulations and experimental results are presented and followed by the conclusions.

2. GMVSS design for MIMO systems

Consider the non-singular, controllable and observable MIMO linear discrete-time stochastic system model

$$x(k) = Ax(k-1) + Bu(k-d) + \Gamma w(k-1), \quad (1)$$

$$y(k) = Cx(k) + v(k), \quad (2)$$

where $x(k) \in \mathbb{R}^n$ is the vector of n state variables, $u(k) \in \mathbb{R}^{n_u}$ is the vector of n_u inputs, $y(k) \in \mathbb{R}^{n_y}$ of $n_y = n_u$ outputs, $w(k) \in \mathbb{R}^n$ and $v(k) \in \mathbb{R}^{n_y}$ are Gaussian disturbance vectors with respectively $\sigma_{w_1}^2, \dots, \sigma_{w_n}^2$ and $\sigma_{v_1}^2, \dots, \sigma_{v_{n_y}}^2$ variances. The $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times n_u}$, $C \in \mathbb{R}^{n_y \times n}$, $\Gamma \in \mathbb{R}^{n \times n}$ are the matrices of the system, $d \geq 1$ is the discrete time-delay (including the zero-order-hold delay).

The GMVSS method consists in finding the control input $u(k)$ for the system in (1) to minimize the performance index

$$J = \phi^2(k + N_y) \quad (3)$$

based on the generalized output

$$\phi(k + N_y) = y(k + N_y) - y_r(k + N_y) + \Lambda u(k), \quad (4)$$

in which N_y is the prediction horizon, $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_{n_u})$ is a diagonal matrix of energy weighting factors λ_i , $i = 1, \dots, n_u$, for the $u_1(k), \dots, u_{n_u}(k)$ inputs and $y_r(k)$ is a known vector of references. Since $x(k)$ and $y(k)$ are vectors of random variables due to the random process noise $w(k)$ and measurements noise $v(k)$, respectively, so is the cost in (3).

The formulation considered in this present work differs from [27] in the following aspects: it covers the MIMO case; the process noise is not the same as the measurement noise; the stochasticity of the minimum variance control is reviewed as it follows.

To define a control problem that does not depend on a particular initial condition of $x(k)$ and $y(k)$ or sequences $w(k)$ and $v(k)$, the expected performance index is used instead:

$$J_{\text{exp}} = \mathbf{E} \left[\phi^2(k + N_y) \right], \quad (5)$$

where $\mathbf{E}[\cdot]$ denotes the mathematical expectation of the generalized output. To find $u(k)$ that minimizes J_{exp} , $\phi(k + N_y)$ must be exactly measurable at time k , which is not true since (1) is a stochastic system as shown as follows:

$$\begin{aligned} x(k) &= Ax(k-1) + Bu(k-d) + \Gamma w(k-1) \\ x(k+1) &= Ax(k) + Bu(k-d+1) + \Gamma w(k) \\ x(k+2) &= A^2x(k) + ABu(k-d+1) + A\Gamma w(k) \\ &\quad + Bu(k-d+2) + \Gamma w(k+1) \\ &\quad \vdots \\ x(k+N_y) &= A^{N_y}x(k) + \sum_{i=1}^{N_y} A^{(N_y-i)}Bu(k-d+i) \end{aligned} \quad (6)$$

$$+ \sum_{i=1}^{N_y} A^{(N_y-i)}\Gamma w(k-1+i)$$

and the predicted output is $y(k + N_y) = Cx(k + N_y) + v(k + N_y)$, or in its expanded form

$$\begin{aligned} y(k + N_y) &= CA^{N_y}x(k) + \sum_{i=1}^{N_y} CA^{(N_y-i)}Bu(k-d+i) \\ &\quad + \sum_{i=1}^{N_y} CA^{(N_y-i)}\Gamma w(k-1+i) + v(k + N_y). \end{aligned} \quad (7)$$

Since $y(k + N_y)$ cannot be determined exactly due to the unknown future influence of both $w(k)$ and $v(k)$, their future effects are estimated by means of the following separation of present and future terms:

$$\sum_{i=1}^{N_y} CA^{(N_y-i)} \Gamma w(k-1+i) + v(k+N_y) = \underbrace{CA^{(N_y-i)} \Gamma w(k)}_{\text{Present}} + \underbrace{\sum_{i=2}^{N_y} CA^{(N_y-i)} \Gamma w(k-1+i) + v(k+N_y)}_{\text{Future}}. \quad (8)$$

By neglecting future disturbance terms, but applying the correction term $w(k) = y(k) - C\bar{x}(k)$, with $\bar{x}(k)$ the Kalman filter estimated state, to predict the future state based on data available up to instant k , i.e. $\hat{x}(k + N_y|k)$, the predicted state of the system is given by

$$\hat{x}(k + N_y|k) = (A^{N_y} - FC)\bar{x}(k) + \sum_{i=1}^{N_y} A^{(N_y-i)} Bu(k - N_y + i) + Fy(k). \quad (9)$$

This system is known as the minimum variance predictor of GMVSS, where

$$F = A^{(N_y-1)}L, \quad (10)$$

in which L is the gain of the steady-state optimal Kalman filter estimator for the system (1), given by

$$\bar{x}(k+1) = (A - LC)\bar{x}(k) + Bu(k-d+1) + Ly(k). \quad (11)$$

L can be solved offline, for the non-adaptive case, by iterating the estimator algebraic Riccati difference equation [32],

$$S(i+1) = AS(i)A^T - AS(i)C^T (CS(i)C^T + R_{KF})^{-1} CS(i)A^T + Q_{KF}. \quad (12)$$

When $i \rightarrow \infty$, starting with a high magnitude $S(0)$, the steady-state error covariance matrix S_∞ is used to solve L , such that

$$L = AS_\infty C^T (CS_\infty C^T + R_{KF})^{-1}. \quad (13)$$

For the minimum variance case, the Kalman filter weighting matrices Q_{KF} and R_{KF} must cope with the system (1), which are based on the covariance matrices of $w(k)$ and $v(k)$, respectively $Q = \text{diag}(\sigma_{w_1}^2, \dots, \sigma_{w_n}^2)$ and $R = \text{diag}(\sigma_{v_1}^2, \dots, \sigma_{v_{n_y}}^2)$. The required weighting matrices are defined as in [32]:

$$Q_{KF} = \Gamma Q \Gamma^T, \quad R_{KF} = R. \quad (14)$$

Using (9), the predicted output based on information known up to time k , i.e. $\hat{y}(k + N_y|k) = C\hat{x}(k + N_y|k)$, is substituted into J_{exp} so to obtain $u(k)$ that minimizes the expected cost

$$\hat{J} = \hat{\phi}^2(k + N_y|k), \quad (15)$$

subjected to the stochastic system in (1).

The GMVSS control law for the MIMO case proposed in this paper is derived by solving $\partial \hat{J} / \partial u(k) = 0$, which leads to the MIMO GMVSS control law:

$$\begin{bmatrix} \sum_{i=1}^{N_y} CA^{(N_y-i)} Bq^{-(N_y-i)} + \Lambda \\ \end{bmatrix} u(k) = y_r(k + N_y) - (CA^{N_y} - CFC)\bar{x}(k) - CFy(k), \quad (16)$$

where q^{-1} is the backward shift operator.

For the MIMO design approach, the l.h.s. of (16) can be rewritten in a vector-matrix form as shown as it follows:

$$\begin{aligned} (CB + \Lambda)u(k) + \sum_{i=1}^{N_y-1} CA^{(N_y-i)} Bq^{-(N_y-i)}u(k) \\ = (CB + \Lambda)u(k) + \left[CA^{(N_y-1)}Bu(k - N_y + 1) \right. \\ \left. + CA^{(N_y-2)}Bu(k - N_y + 2) + \dots \right. \\ \left. + CA^2Bu(k - 2) + CABu(k - 1) \right] \\ = (CB + \Lambda)u(k) + \underbrace{\begin{bmatrix} CAB \\ CA^2B \\ \vdots \\ CA^{(N_y-1)}B \end{bmatrix}^T}_{\tilde{U}} \begin{bmatrix} u(k-1) \\ u(k-2) \\ \vdots \\ u(k - N_y + 1) \end{bmatrix} \end{aligned} \quad (17)$$

Since $n_u = n_y$, the GMVSS control design in the MIMO case will be feasible if and only if $(CB + \Lambda)$ is invertible, such that

$$u(k) = (CB + \Lambda)^{-1} \left[y_r(k + N_y) - \tilde{U} \right. \\ \left. - (CA^{N_y} - CFC)\bar{x}(k) - CFy(k) \right]. \quad (18)$$

Analyzing the control law in (16), it is possible to observe that the computational burden will increase as N_y is increased, since it will require $N_y + 1$ control input regressors (i.e., $u(k-1), \dots, u(k - N_y + 1)$).

2.1. System augmentation by integrator addition

In order to add integrators to the inputs of the system to try and track step-like reference signals for servo control, the system in (1) can be augmented in the following manner:

$$\begin{bmatrix} y(k+1) \\ \Delta x(k+1) \end{bmatrix} = \begin{bmatrix} I & CA \\ 0 & A \end{bmatrix} \begin{bmatrix} y(k) \\ \Delta x(k) \end{bmatrix} + \begin{bmatrix} CB \\ B \end{bmatrix} \Delta u(k-d+1) \\ + \begin{bmatrix} I & C\Gamma \\ 0 & \Gamma \end{bmatrix} \begin{bmatrix} z(k) \\ w(k) \end{bmatrix}, \quad (19)$$

$$y_a(k) = y(k) = \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} y(k) \\ \Delta x(k) \end{bmatrix}, \quad (20)$$

where the augmented covariance matrices

$$Q_a = \text{diag}(\text{diag}(R) \text{diag}(Q)), \quad (21)$$

$$R_a = R, \quad (22)$$

are considered, since $z(k) = v(k+1)$. $x_a(k) = [y(k) \quad \Delta x(k)]^T$ is the augmented state vector, $y_a(k)$ is the output of the augmented system and $\Delta u(k)$ is the control increment, such that $\Delta = 1 - q^{-1}$. Then,

$$u(k) = u(k-1) + \Delta u(k), \quad (23)$$

is the control signal to be applied to the system input, which can easily be implemented on a digital computer to achieve incremental control.

3. Estimation of the quadrotor MIMO stochastic model

In this work the MIMO process encompasses a distributed system comprised of a ground control station, a wireless network and the quadrotor (model AR.Drone 2.0 manufactured by Parrot

Drones SAS). This system was modeled using experimental system identification by means of non-recursive least-squares estimation based on input/output registered flight data. To link the quadrotor and the ground computer it was used the software development kit developed by [33] for MATLAB® using the sampling-time of $T_s = 0.065$ second, which is the manufacturer standard [34].

3.1. Multivariable stochastic state-space model identification

The non-recursive least-squares estimation algorithm is used in a combined system identification form [35], since the deterministic state-space model identification (i.e. the estimation of A, B) is followed by the stochastic identification (i.e., estimation of Γ, Q, R).

Assuming that all the state-variables are being measured or estimated, the non-recursive least-squares parametric estimator can be used to identify the following system parameters:

$$\begin{bmatrix} x_1(k) \\ \vdots \\ x_n(k) \end{bmatrix} = \begin{bmatrix} \hat{a}_{11} & \cdots & \hat{a}_{1n} \\ \vdots & \ddots & \vdots \\ \hat{a}_{n1} & \cdots & \hat{a}_{nn} \end{bmatrix} \begin{bmatrix} x_1(k-1) \\ \vdots \\ x_n(k-1) \end{bmatrix} + \begin{bmatrix} \hat{b}_{11} & \cdots & \hat{b}_{1n_u} \\ \vdots & \ddots & \vdots \\ \hat{b}_{n1} & \cdots & \hat{b}_{nn_u} \end{bmatrix} \begin{bmatrix} u_1(k-d) \\ \vdots \\ u_{n_u}(k-d) \end{bmatrix} + \begin{bmatrix} \hat{\gamma}_{11} & \cdots & \hat{\gamma}_{1n} \\ \vdots & \ddots & \vdots \\ \hat{\gamma}_{n1} & \cdots & \hat{\gamma}_{nn} \end{bmatrix} \begin{bmatrix} w_1(k-1) \\ \vdots \\ w_n(k-1) \end{bmatrix}, \quad (24)$$

$$\begin{bmatrix} y_1(k) \\ \vdots \\ y_{n_y}(k) \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{n_y \times n_y} & \mathbf{0}_{n_y \times (n-n_y)} \end{bmatrix} \begin{bmatrix} x_1(k) \\ \vdots \\ x_n(k) \end{bmatrix} + \begin{bmatrix} v_1(k) \\ \vdots \\ v_{n_y}(k) \end{bmatrix}. \quad (25)$$

The estimated parameters vectors are defined as

$$\begin{aligned} \hat{\theta}_1^T &= [\hat{a}_{11} \cdots \hat{a}_{1n} \hat{b}_{11} \cdots \hat{b}_{1n_u} \hat{\gamma}_{11} \cdots \hat{\gamma}_{1n}] \\ &\quad \vdots \\ \hat{\theta}_n^T &= [\hat{a}_{n1} \cdots \hat{a}_{nn} \hat{b}_{n1} \cdots \hat{b}_{nn_u} \hat{\gamma}_{n1} \cdots \hat{\gamma}_{nn}] \end{aligned} \quad (26)$$

and future observations are based on the following vector of regressors:

$$\phi_{ls}^T(k) = [x(k-1)]_{1 \times n}^T [u(k-d)]_{1 \times n_u}^T [w(k-1)]_{1 \times n}^T. \quad (27)$$

The vectors of regressors, of a total of N samples, are organized to form the matrix of regressors,

$$\Phi = \begin{bmatrix} \phi_{ls}^T(k) \\ \vdots \\ \phi_{ls}^T(N) \end{bmatrix}_{N \times (2n+n_u)}, \quad (28)$$

leading to the least-squares estimation error covariance matrix,

$$\mathbf{P} = \left(\Phi^T \Phi \right)_{(2n+n_u) \times (2n+n_u)}^{-1}. \quad (29)$$

Considering X_1, \dots, X_n the measurements vectors, the matrix formed by the vectors of parameters can be calculated in the following manner:

$$\begin{bmatrix} \hat{\theta}_1 & \cdots & \hat{\theta}_n \end{bmatrix}_{(2n+n_u) \times n} = \mathbf{P} \Phi^T [X_1 \cdots X_n]_{N \times n}. \quad (30)$$

To identify the state Gaussian noise vector and its input matrix, $\Gamma w(k-1)$, as well as the output process noise vector, $v(k)$, the identification procedure is made in a combined form [35] in order to estimate $\Gamma w(k-1)$ and $v(k)$, as the estimated state error and estimated output error, respectively. In the first run of the algorithm the deterministic part is estimated, such that

$$x(k) = \bar{A}x(k-1) + \bar{B}u(k-d) + \bar{w}(k-1), \quad (31)$$

where \bar{A}, \bar{B} are provisory estimated matrices and the estimation residuals, $\bar{w}(0), \dots, \bar{w}(N-1)$, are then used in the complete vector of regressors shown in (27), to estimate the complete combined deterministic-stochastic model. Alternatively, observer/Kalman filter identification or extended recursive least-squares estimator can be applied for the adaptive control synthesis or when ill-conditioned (non-invertible) $\Phi^T \Phi$ occurs.

3.2. System identification applied to the quadrotor

Consider the following state vector,

$$x^T = [\phi \ \theta \ u_{spd} \ v_{spd} \ \psi \ r \ h \ w_{spd}], \quad (32)$$

where $\phi(k)$ is the roll angle [rad], $\theta(k)$ is the pitch angle [rad], $u_{spd}(k)$ is the longitudinal speed [m/s], $v_{spd}(k)$ is the lateral speed [m/s], $\psi(k)$ and $r(k)$ are respectively the yaw/heading angle [rad] and the yaw angular speed [rad/s], $h(k)$ and $w_{spd}(k)$ are respectively the altitude [m] and the vertical speed [m/s]. The quadrotor orientation is given in the x, y, z axes defined in the North-East-down system [36], such that right-handed rotation about the x -axis gives positive roll; about the y -axis gives positive pitch; about the z -axis gives positive yaw.

The considered input vector of the system is given by

$$u^T(k) = [u_v(k) \ u_u(k) \ u_\psi(k) \ u_h(k)] \quad (33)$$

where all inputs were defined in the range of $[-1, 1]$ and $u_v(k)$ is the lateral thrust, $u_u(k)$ is the longitudinal thrust, $u_\psi(k)$ is the yaw thrust, $u_h(k)$ is the vertical thrust.

In order to simplify the system identification, the procedure was subdivided into three subsystems: the horizontal speed system (influence of $u_v(k)$ and $u_u(k)$ to lateral/longitudinal dynamics), the yaw/heading system (influence of $u_\psi(k)$ to yaw dynamics and the altitude system ($u_h(k)$ to vertical dynamics).

The complete system model was identified in three batch procedures for the three systems considered, such that in Fig. 1 it is shown the model validation results for the Lateral/Longitudinal dynamics and in Fig. 2, for the yaw and altitude dynamics, where it is possible to observe how close the simulated outputs fit to the real registered outputs. To validate the model, a new data set of inputs and outputs were used, where this new set of inputs was used to excite the identified model to produce a data set of simulated outputs to be compared to the real measured outputs. The system identification and system validation datasets were obtained during real flight experiments with the quadrotor, while registering state and inputs variables shown in (32) and (33), respectively.

The system time delay of $d = 3$ was observed experimentally and estimated system matrices and covariance matrices are given by:

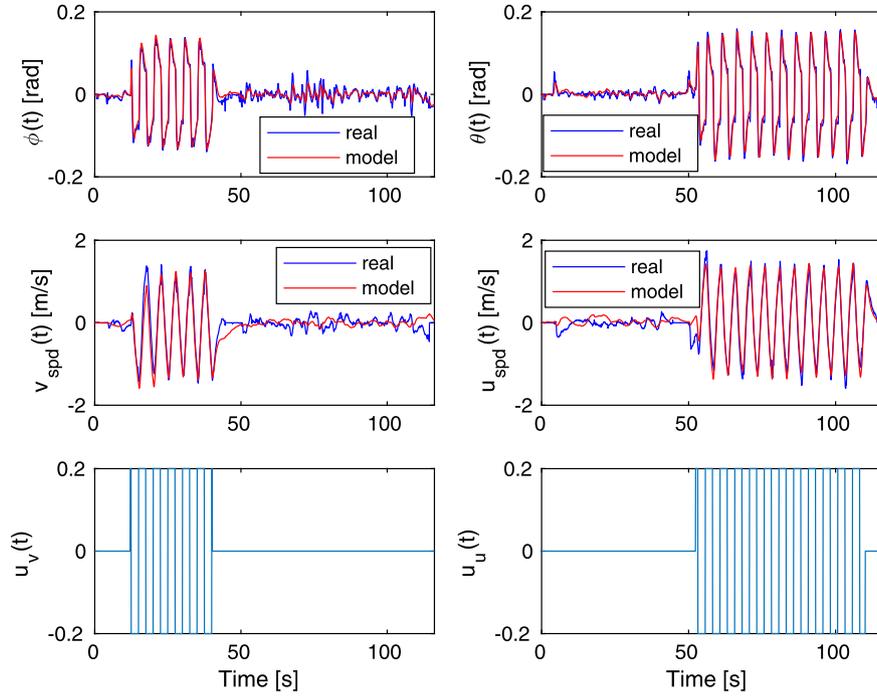


Fig. 1. Lateral/Longitudinal system model validation.

$$A = \begin{pmatrix} 0.8177 & -0.00617 & -0.0006665 & -0.007602 & 0 & 0 & 0 & 0 \\ 0.005481 & 0.8208 & 0.009405 & 0.0007079 & 0 & 0 & 0 & 0 \\ 0.02713 & -0.7337 & 0.9996 & -0.0005473 & 0 & 0 & 0 & 0 \\ 0.7289 & -0.01162 & 0.001246 & 0.9989 & 0.00126 & 0.008644 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.01938 & 0.133 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.002158 & 0.00117 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.0332 & 0.018 \end{pmatrix} \quad (36)$$

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0.9983 & 0.04624 & 0 & 0 \\ -0.02621 & 0.7114 & 0 & 0 \\ 0 & 0 & 0.9992 & 0.01185 \\ 0 & 0 & -0.01249 & 0.1823 \end{pmatrix} \quad (34)$$

$$B = \begin{pmatrix} 0.09348 & 0.002153 & 0 & 0 \\ -0.001933 & 0.09677 & 0 & 0 \\ -0.007098 & 0.003633 & 0 & 0 \\ -0.007148 & 0.002068 & 0 & 0 \\ 0 & 0 & 0.02743 & 0 \\ 0 & 0 & 0.4219 & 0 \\ 0 & 0 & 0 & 0.03733 \\ 0 & 0 & 0 & 0.5742 \end{pmatrix} \quad (35)$$

$$\Gamma = \begin{pmatrix} 0.1174 & 0.006617 & 0.001413 & -0.001458 \\ -0.0005712 & 0.1364 & -0.001605 & -0.001304 \\ 0.04352 & 0.2938 & -0.03775 & 0.01228 \\ -0.3667 & -0.05233 & 0.00218 & -0.0197 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$Q = \text{diag} (0.0002268 \ 0.0002791 \ 0.01412 \ 0.01568 \ 0.1367 \ 0.006427 \ 0.003046 \ 0.01989) \quad (37)$$

$$R = \text{diag} (0.03363 \ 0.02296 \ 0.01367 \ 0.003046) \quad (38)$$

For the required autopilots in this project, the output vector is $y(k) = [v_{\text{spd}}(k) \ u_{\text{spd}}(k) \ \psi(k) \ h(k)]^T$, such that the output matrix is given by

$$C = \begin{pmatrix} 0 & 0 & 0 & 1.0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.0 & 0 \end{pmatrix} \quad (39)$$

4. Simulation and experimental results

In this section, the MIMO GMVSS controller will be investigated in several different simulation conditions: i) in centralized MIMO control of lateral/longitudinal speeds, heading angle and altitude, where the influence of the prediction horizon and control effort weighting are assessed in terms of control performance and computational load of the algorithm; ii) in decentralized MIMO control of the heading angle and altitude to assess different prediction horizons between the three designed autopilots; iii) in comparison to an industry-standard MPC; iv) in the assessment of the sensitivity to stochastic disturbances to assess the minimum variance performance. Finally, the MIMO GMVSS is applied to the real quadrotor in an outdoor flight under light and moderate wind gust

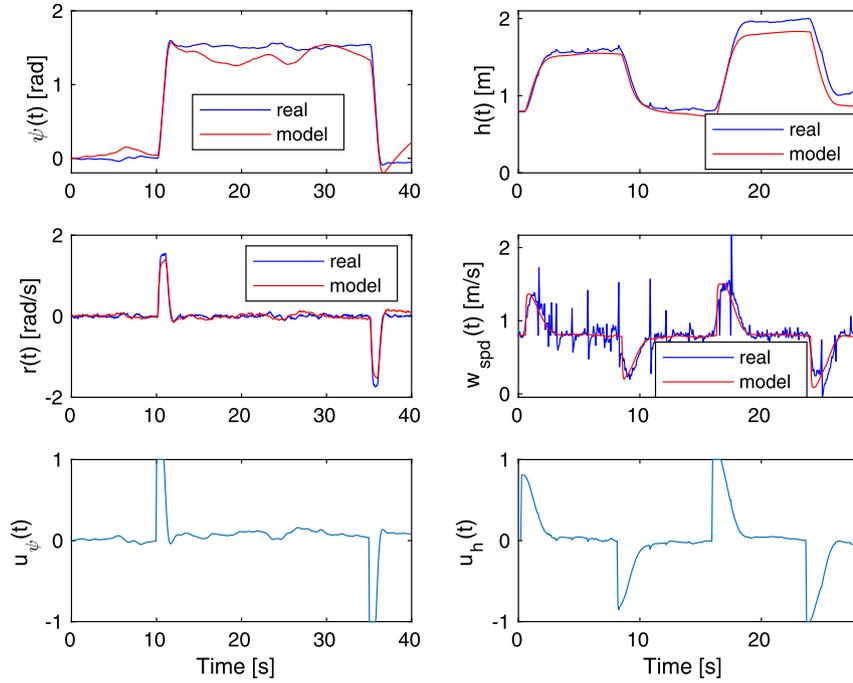


Fig. 2. Yaw and altitude models validation.

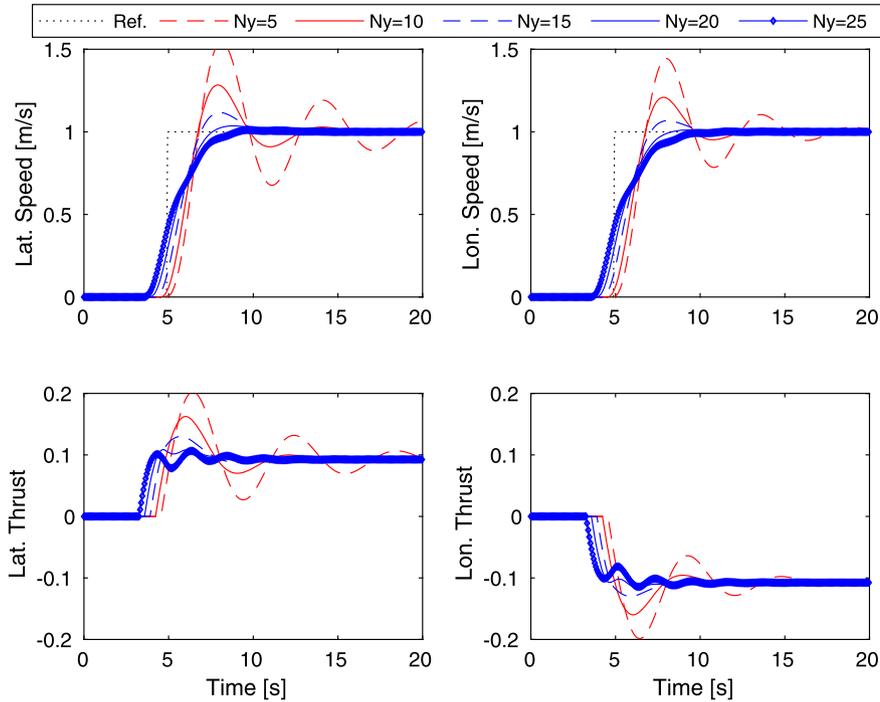


Fig. 3. Lateral/Longitudinal speed hold autopilots in the MIMO case for different N_y values.

conditions. Experimental results are also compared to industry-standard MPC.

4.1. Quadrotor autopilot requirements

To the best of authors' knowledge, there is no standard specifications for quadrotors autopilots to date. So far, the major concern regards integrating unmanned aerial systems (UAS) into air-traffic management (ATM), establishing rules at which drone manufacturers must be aware of, such as the European Organisation for the Safety of Air Navigation "UAS ATM Flight Rules", for example.

Thus, in this work we propose quadrotor autopilot specifications based on ATM spatial separation rules.

According to [36], the minimum vertical separation of a modern highly automated passenger aircrafts is 300 m (1000 ft) and its altitude hold autopilot typically holds the aircraft well within ± 60 m (200 ft). The proportional ratio to set the hold autopilot specification is 1/5. Based on this and on the fact that quadrotors can fly indoors and outdoors, a 1 m for vertical and horizontal separation rule is being considered and by the 1/5 ratio, the altitude autopilot must hold within ± 0.2 m, ± 0.2 m/s for velocities autopilots and ± 0.2 rad for the heading autopilot.

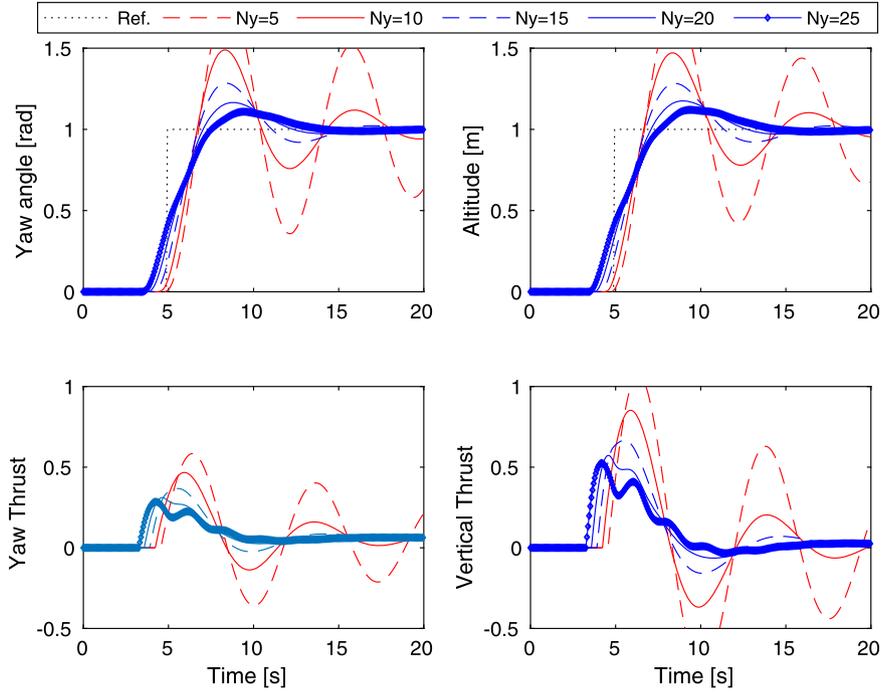


Fig. 4. Heading and altitude hold autopilots in the MIMO case for different N_y values.

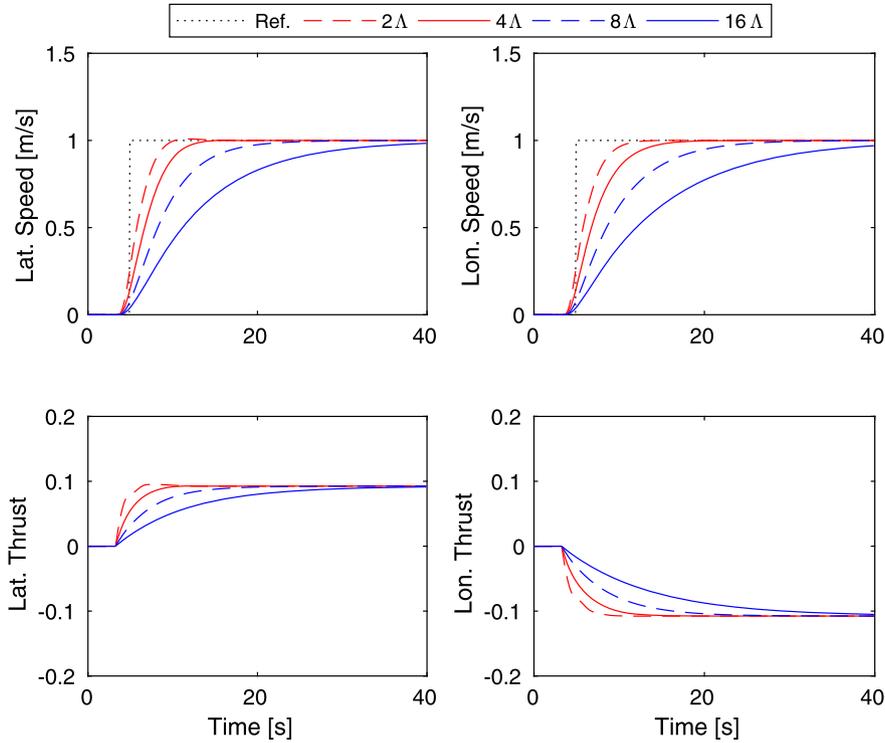


Fig. 5. Lateral/Longitudinal speed hold autopilots in the MIMO case for different Λ values.

The transitory and the steady-state behavior will be evaluated using step-like reference sequences such that the vector of references is defined as follows:

$$y_r(k) = [v_{Ref}(k) \ u_{Ref}(k) \ \psi_{Ref}(k) \ h_{Ref}(k)]^T,$$

where the subscript *Ref* regards reference values for the controlled outputs.

4.2. Centralized MIMO control autopilot

In centralized MIMO control design, a single prediction horizon is considered for the three autopilots under investigation. The design model was augmented to include integrators to its inputs and the following described procedures were executed to tune and select N_y and Λ : $N_y = d + 1$ was set in order to find a stabilizing control weighting matrix $\Lambda = \text{diag}(\lambda_v, \lambda_u, \lambda_\psi, \lambda_h)$. $\Lambda = \text{diag}(90, -90, 30, 15)$ was selected (where the negative value for

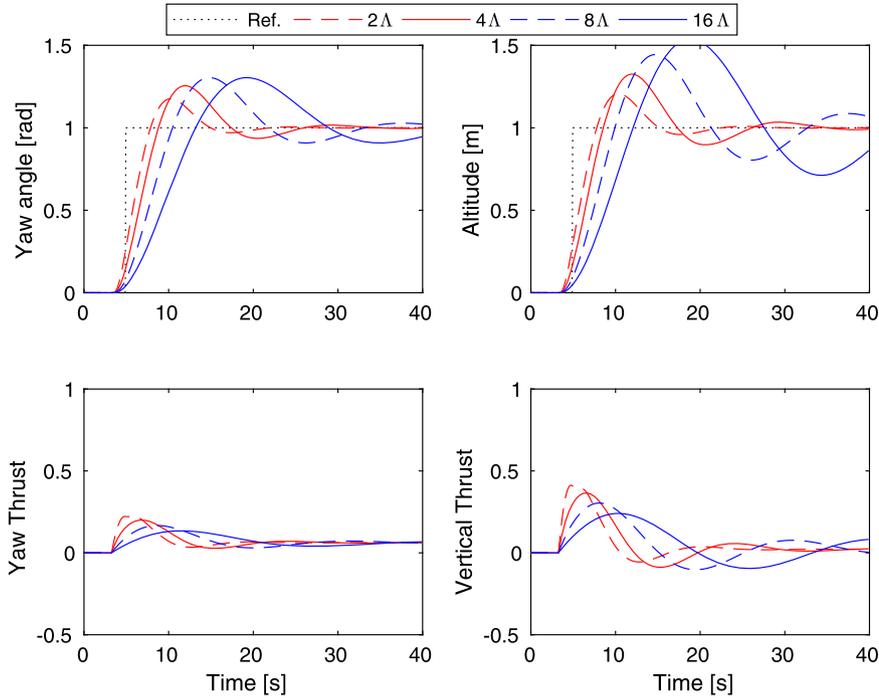


Fig. 6. Heading and altitude hold autopilots in the MIMO case for different Λ values.

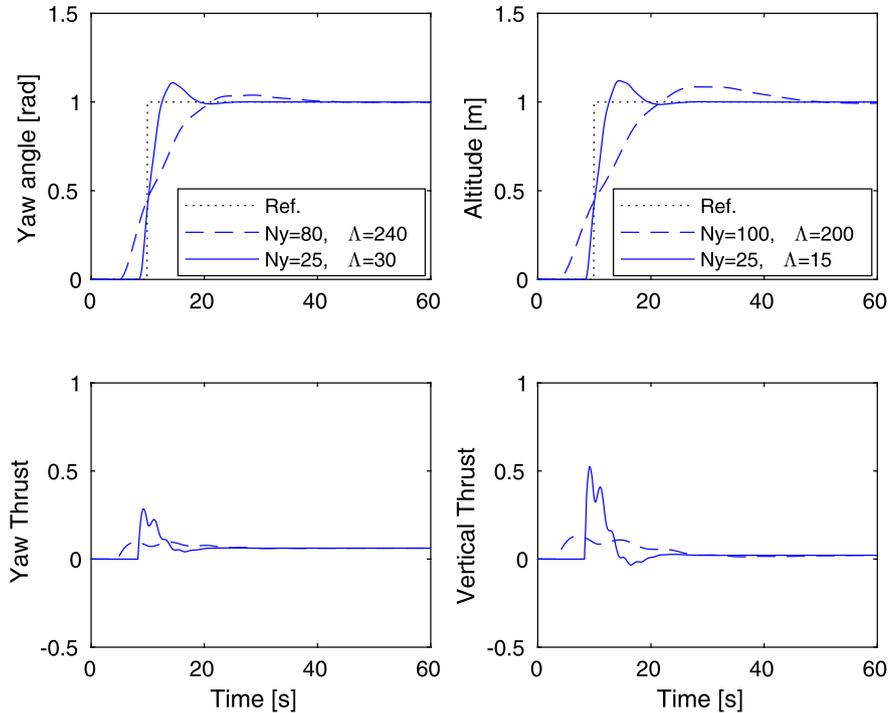


Fig. 7. Heading hold and altitude hold autopilots results with decentralized MIMO control.

λ_u was required due to the negative DC gain of the longitudinal system) and the set of $N_y = \{5, 10, 15, 20, 25\}$ was assessed to evaluate the influence of prediction on the transitory behavior and on the computational load by means of the average time evaluation of the algorithm iterations.

In Fig. 3 it is shown how the increase in the prediction horizon affects the damping of the lateral/longitudinal speeds, reducing oscillations and the settling time, progressively. In Fig. 4, the same effect on the damping was observed, however it can be seen that

$N_y = 25$ was not sufficient to avoid increasing the yaw angle and the altitude overshoots.

In terms of the computational load, the average time of the algorithm iterations for the prediction set $N_y = \{5, 10, 15, 20, 25\}$ was, respectively: 100, 115, 133, 147, 153 microseconds. A 4th Generation Intel Core i5-4200U CPU at 1.6 GHz, 4 GB of RAM, with Ubuntu 18.04 and MATLAB® R2018a was used to run the simulations.

In Figs. 5 and 6 it is shown the influence of the Λ matrix, while maintaining $N_y = 25$. The following tuning set, based on

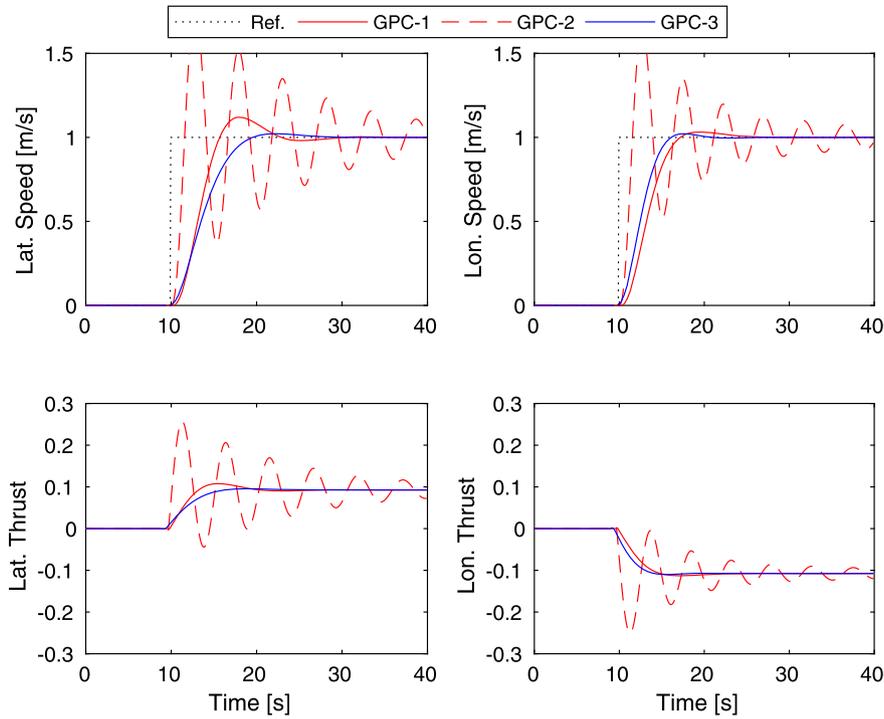


Fig. 8. Lateral/Longitudinal speed hold autopilot results with decentralized GPC control.

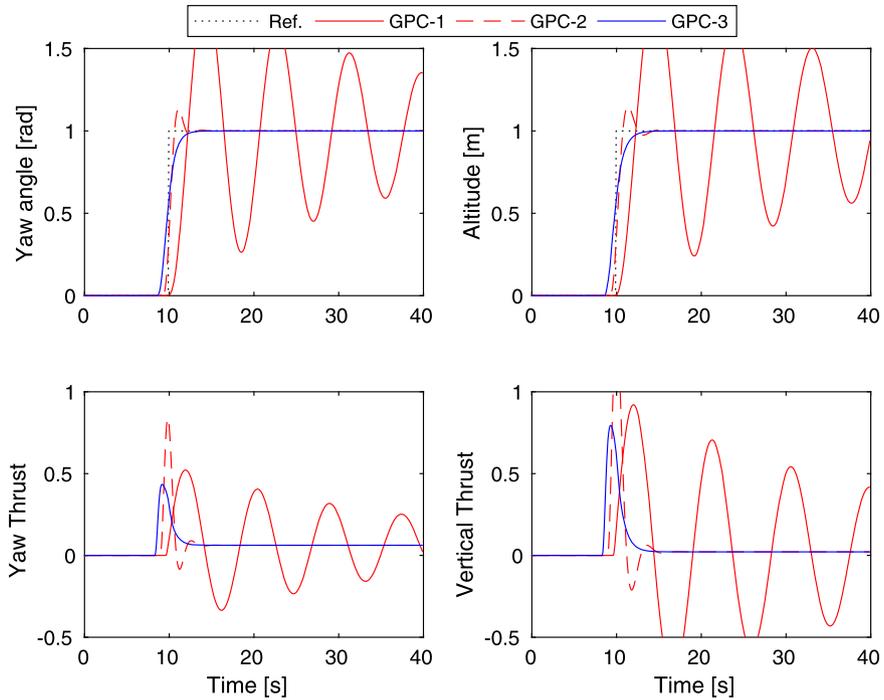


Fig. 9. Heading hold and altitude hold autopilots results with decentralized GPC control.

$\Lambda = \text{diag}(90, -90, 30, 15)$, was selected: $\{2\Lambda, 4\Lambda, 8\Lambda, 16\Lambda\}$. It can be seen that by increasing the magnitude of the Λ matrix, leads to a more conservative control action, but it does not mitigate oscillations of the yaw and altitude control problems. Instead, the overshoot and the settling time has increased. To try to overcome these problems with these autopilots, the heading-hold and altitude-hold systems will be assessed in a decentralized form in the next subsection.

4.3. Decentralized GMVSS control

In this section the yaw and altitude control systems are designed independently of the lateral/longitudinal speed controller, allowing independent prediction horizons, such that three control systems are designed: centralized MIMO lateral/longitudinal speed hold; SISO heading hold and SISO altitude hold autopilots.

In Fig. 7 it is shown the results of the heading hold using $N_y = 80$, $\Lambda = 240$ and the altitude hold using $N_y = 100$, $\Lambda = 200$,

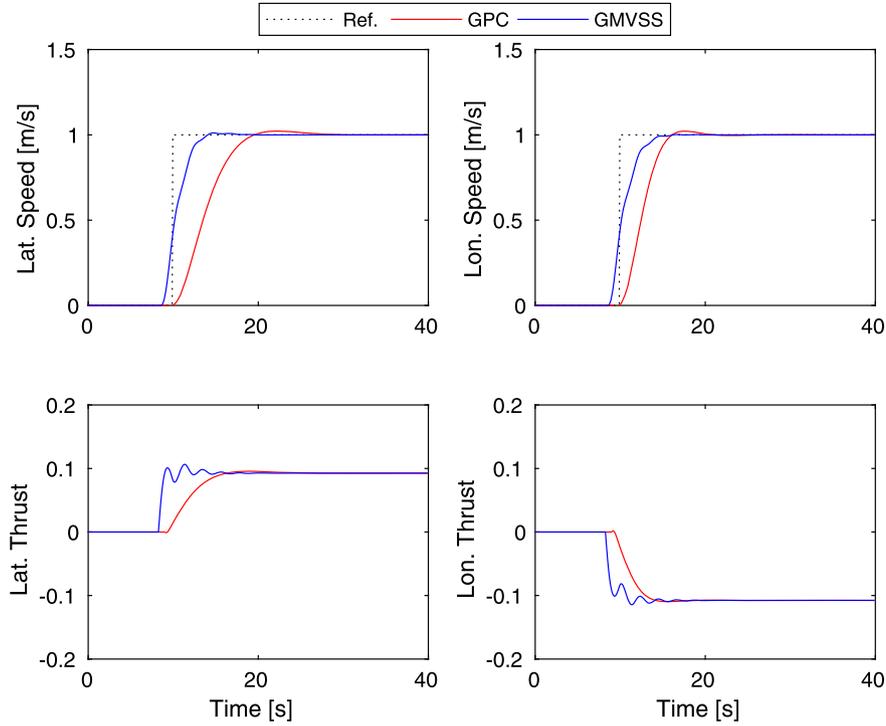


Fig. 10. Lateral/Longitudinal speed hold autopilot: GPC versus GMVSS control comparison.

selected by trial and error while trying to reduce the overshoot. Unfortunately, the settling time has increased significantly in face of previously assessed tuning parameters for the same autopilots, degrading the response speed that might be required to execute high speed maneuvers. It is also important to remark that the overshoots are not violating the *quadrotor autopilot requirements* proposed in Section 4.1.

4.4. GMVSS autopilots versus GPC autopilots

The GPC is assessed in this subsection in order to compare its results to the proposed GMVSS MIMO controller. The GPC autopilot was implemented based on the system polynomial model [37], obtained after transforming the state-space realization into its transfer function description.

The GMVSS autopilots are simulated using $\Lambda = \text{diag}(90, -90, 30, 15)$ and $N_y = 25$, while the GPC tuning is modified to establish the comparison. The GPC, however, uses four tuning parameters: the $N_1 = d$ to compensate the process time-delay, the control weighting factor λ_i , $i = 1, \dots, n_u$, the output prediction horizon N_y , and the control prediction horizon N_u , which refers to future control actions in a receding-horizon control scheme [24].

From a deterministic viewpoint, N_u in the GPC can be set as small as possible, since it is considered that the control increment $\Delta u(k)$ will tend to zero in the near future, but it will never happen from the stochastic viewpoint and this parameter will be further assessed in the subsection of stochastic disturbance sensitivity analysis and minimum variance performance evaluation.

In Figs. 8 and 9, the GPC simulation results are shown. Three tuning sets were considered: GPC-1 $N_y^{\text{spd}} = 5$, $N_y^{\psi} = 5$, $N_y^h = 5$ and $\lambda_{\text{gpc}} = \{90, 90, 30, 15\}$; GPC-2 $N_y^{\text{spd}} = 7$, $N_y^{\psi} = 15$, $N_y^h = 15$ and $\lambda_{\text{gpc}} = \{90, 90, 30, 15\}$; GPC-3 $N_y^{\text{spd}} = 15$, $N_y^{\psi} = 25$, $N_y^h = 25$ and $\lambda_{\text{gpc}} = \{4 \times 10^4, 2 \times 10^4, 30, 15\}$. $N_u = d$ was used with these three cases.

From the results shown in Figs. 8 and 9 it was selected the GPC-3 tuning case as the best one in terms of response speed and

damping. This result is plotted against the GMVSS best result and shown in Figs. 10 and 11 in order to assess the performance of the autopilots without the presence of process and measurement noises. It is possible to observe that in this deterministic analysis the GMVSS presented fast and damped output responses in the lateral/longitudinal speed control, whereas the GPC has outperformed the GMVSS in the yaw and altitude control.

4.5. Autopilots minimum variance performance

In order to hold speed, altitude and compass heading, the designed autopilots must be robust to wind, to ground effect and wall effect of turbulent air coming back and surrounding the quadrotor, along with other parasitic dynamics in general. These effects are being represented by the stochastic disturbances of the modeled system in the form of process noises and measurement noises. In this subsection, designed autopilots are assessed in terms of their sensitivity to stochastic disturbances and their minimum variance performance. The selected tuning for the GMVSS in this essay is $\Lambda = \text{diag}(90, -90, 30, 15)$ and $N_y = 25$. For the GPC, the analysis starts with GPC-3 tuning, using $N_u = d$.

It is possible to observe in Figs. 12 and 13 that the GMVSS method outperformed the traditional MPC in the minimum variance performance, showing a much lower sensitivity to stochastic disturbances with different variance values, concerning different system dynamics, such as horizontal speeds, yaw angle and altitude.

The performance of the controlled outputs appears to be similar, but the stochastic performance of the control signal of the GPC method, in this particular case study, is physically unrealizable, since it violates the input saturation constraints of $[-1, 1]$. The same experiment was then repeated using GPC's $N_u = N_y$ and the results are shown in Figs. 14 and 15. It can be observed the reduction in the control signal chattering, in the GPC autopilots when $N_u = N_y$. However, this result is still unrealizable in practice due to excessive chattering and high amplitude values in the con-

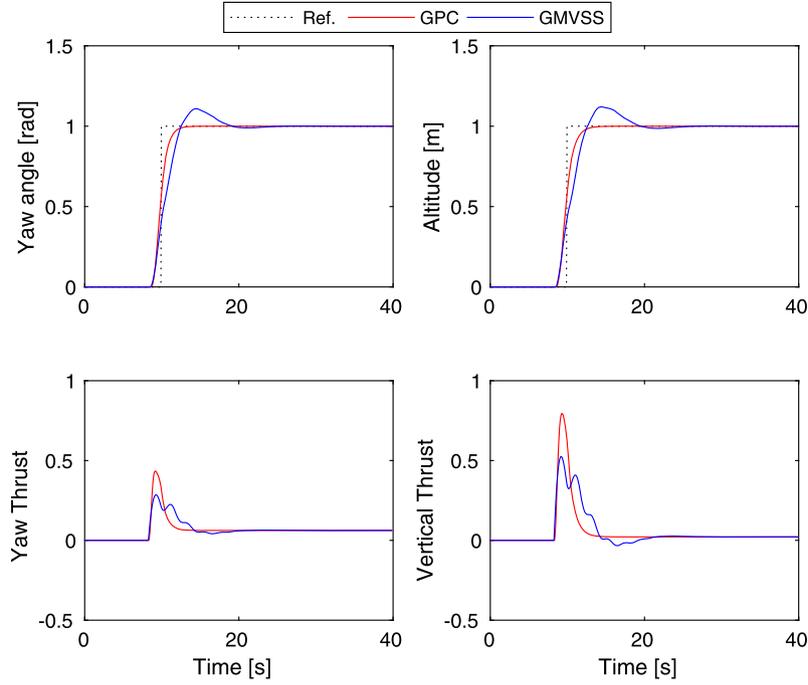


Fig. 11. Heading hold and altitude hold autopilots: GPC versus GMVSS control comparison.

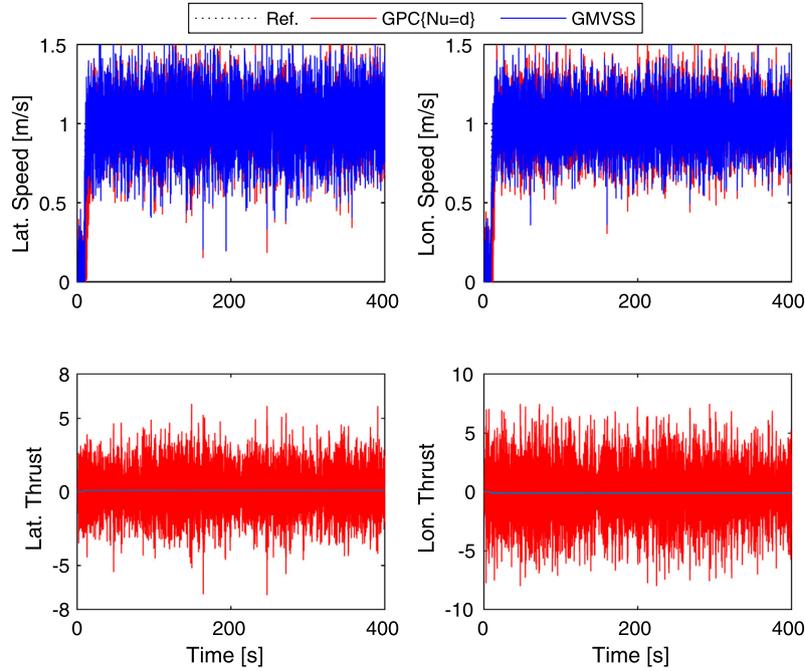


Fig. 12. Lateral/Longitudinal speed hold autopilot: GPC with $N_u = d$ versus GMVSS minimum variance performance.

trol signal required to produce similar GMVSS controlled output performance.

Minimum variance performance results can be summarized in terms of controlled output signals and control signals variances, as presented in Tables 1 and 2, respectively. The lowest the variance, the lowest is the deviation from the mean value of the considered variable. From the controlled output perspective, it means better reference tracking and regulation, and from the control signal perspective, it means less power consumption and less wearing of actuator devices.

The enhanced stochastic performance of GMVSS in this case study came with the increased computational load of the algo-

Table 1
Outputs minimum variance performance indices.

Method	$\sigma_{v_{sdp}}^2$	$\sigma_{u_{sdp}}^2$	σ_{ψ}^2	σ_h^2
GMVSS	0.0579	0.0477	0.0378	0.0269
GPC $\{N_u = N_y\}$	0.0670	0.0561	0.0403	0.0272
GPC $\{N_u = d\}$	0.0660	0.0566	0.0431	0.0273

rithm when compared to the GPC. The average iteration time of the GMVSS was 153 μ s, while the GPC was 65 μ s.

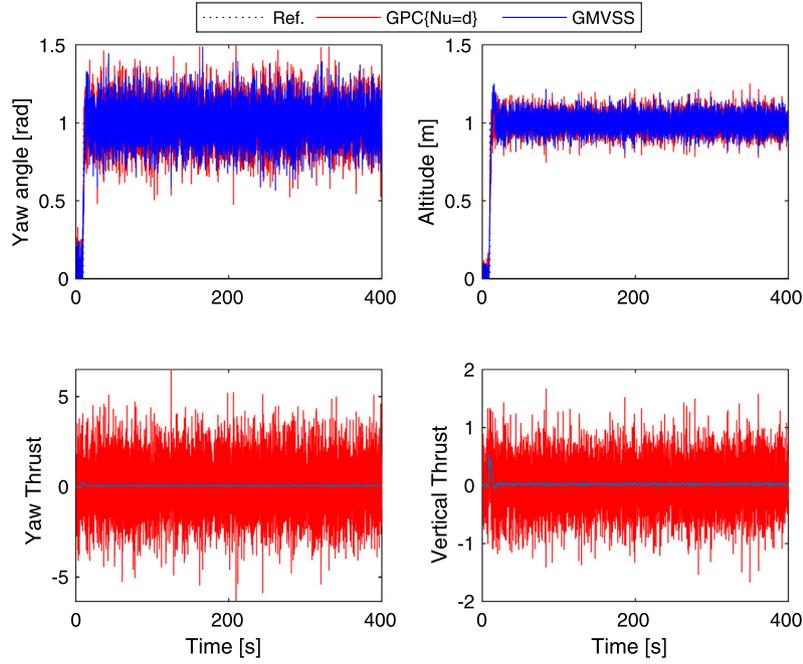


Fig. 13. Heading hold and altitude hold autopilots: GPC with $N_u = d$ versus GMVSS minimum variance performance.

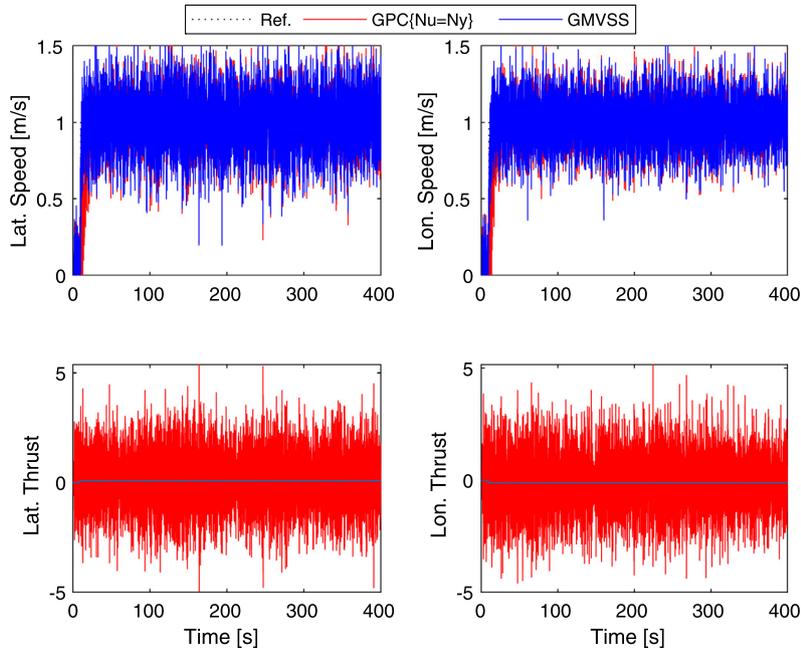


Fig. 14. Lateral/Longitudinal speed hold autopilot: GPC with $N_u = N_y$ versus GMVSS minimum variance performance.

Table 2
Inputs minimum variance performance indices.

Method	$\sigma_{u_v}^2$	$\sigma_{u_u}^2$	$\sigma_{u_\psi}^2$	$\sigma_{u_h}^2$
GMVSS	0.0002	0.0002	0.0005	0.0015
GPC($N_u = N_y$)	1.8924	1.8476	0.7565	0.0550
GPC($N_u = d$)	2.5561	6.2198	2.4754	0.1872

4.6. Experimental flight results

To give a proof of concept, the designed GMVSS autopilots were tested with the real quadrotor using MATLAB® and Simulink® with automatic code generation and the software development kit of [33]. The tests occurred in outdoor flights at a parking lot of the

Federal University of Pará near the Guamá River (cf. Fig. 16) in two different weather conditions: under light wind gusts and moderate wind gusts. The GPC autopilots were tested only under light wind conditions due to safety reasons.

The weather forecast, for the time of the flight under light winds, was: wind direction 59° , wind speed 2 m/s (4 kts), wind gusts 3.6 m/s (7 kts), air temperature $27 - 32^\circ\text{C}$, air pressure 1010 hPa. For the flight under moderate winds, was: wind direction 88° , wind speed 3.6 m/s (7 kts), wind gusts 6.2 m/s (12 kts), air temperature $28 - 32^\circ\text{C}$, air pressure 1009 hPa. The data was obtained from the closest weather station located at the International Airport of Belém, located in the northern Brazil.

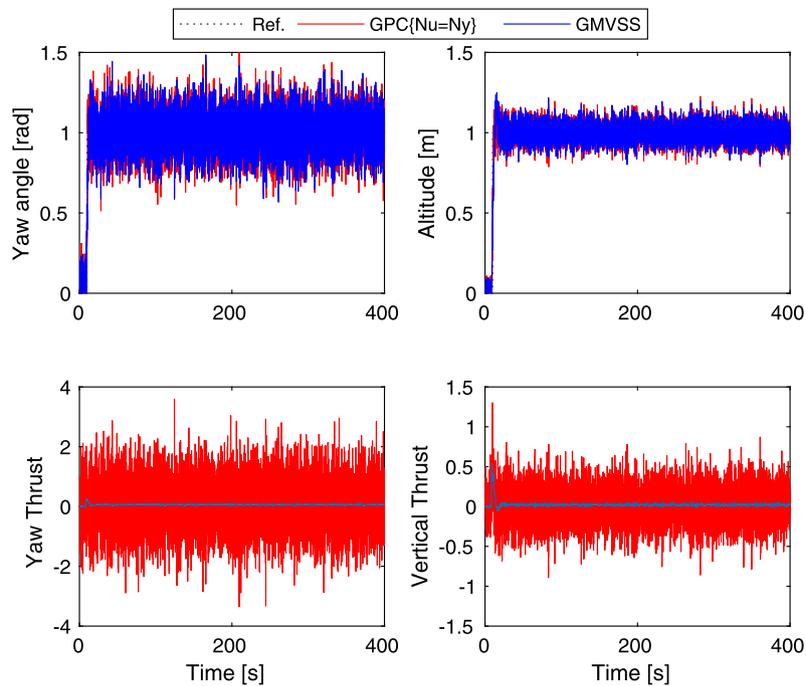


Fig. 15. Heading hold and altitude hold autopilots: GPC with $N_u = N_y$ versus GMVSS minimum variance performance.



Fig. 16. Quadrotor flying at a parking lot of the Federal University of Pará near the Guamá River located in northern Brazil.

4.6.1. Flight under light wind with GMVSS

In Figs. 17 and 18, the GMVSS autopilots results under light wind gusts are shown. All remote wireless autopilots were already engaged from takeoff to landing, however, during these terminal stages the quadrotor cannot accept any remote control commands and integrator windup effect occurs. Even though, GMVSS was capable of handling windup and the arbitrary initial condition, guaranteeing speed hold, heading hold and altitude hold functions, with similar results as observed in the simulations. For example, comparing the speed hold results to the ones seen in Fig. 14, it is possible to observe that the noises seen in practice and its effects in closed-loop control are similar to those observed in the simulations.

For reference signal tracking, speed tracking tests were avoided due to safety reasons and only the GMVSS heading and altitude autopilots were assessed in flight. In Figs. 19 and 20 these results are shown, where it is possible to observe that despite the noise

and disturbances in the real experiment, the yaw and the altitude references were tracked with small overshoots, similar to those observed in the simulation results (cf. Fig. 11).

4.6.2. Flight under light wind with GPC

The simulation results of the GPC autopilots anticipated an unfeasible implementation of such controllers, since all observed control signals violated the input saturation constraint of $[-1, 1]$ (cf. Figs. 14 and 15). Despite anticipating the problem seen in the simulations, the GPC was assessed during a short time flight in three unsuccessful attempts to takeoff and control the quadrotor. These results were omitted since the flight vehicle became unstable and required emergency landings.

In order to cope with inputs saturation limits, the GPC autopilot code was modified to saturate the control vector in case of any violation and the quadrotor was able to fly for a longer period in order to provide data for comparisons. The results are

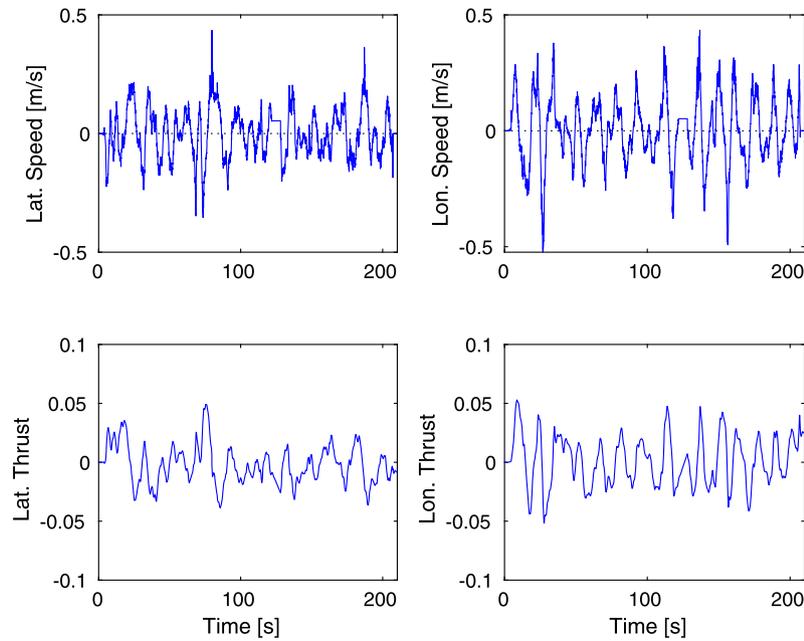


Fig. 17. GMVSS lateral/longitudinal speed-hold autopilots under light wind gusts.

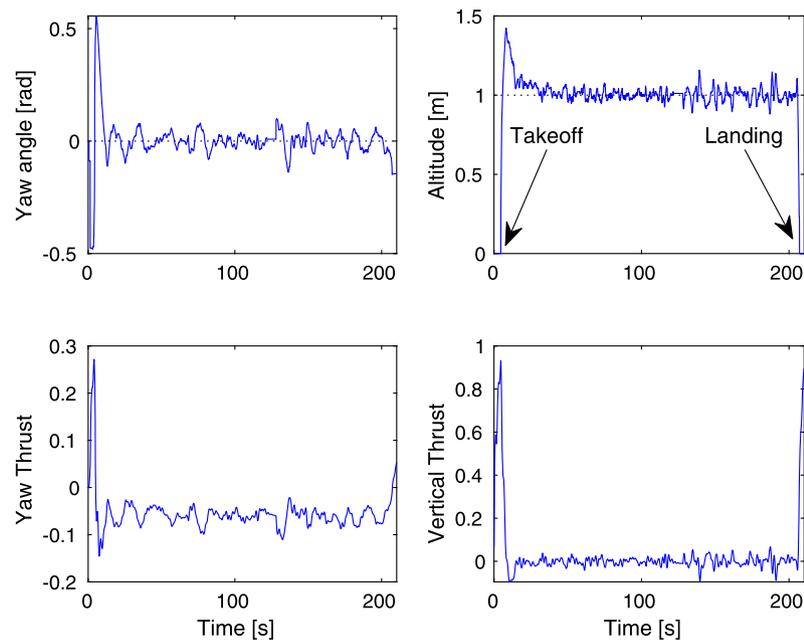


Fig. 18. GMVSS yaw-hold and altitude-hold autopilots under light wind gusts.

shown in Figs. 21 and 22. Differently from the GMVSS, the integrator windup effect, along with arbitrary initial conditions and noise, made the GPC speed hold and altitude hold autopilots oscillate, causing lateral and longitudinal dispersions beyond the available physical space of the flight location, forcing a premature landing. This modified GPC, however, has performed surprisingly well in the heading hold function, as can be observed in Fig. 22.

4.6.3. Flight under moderate wind with GMVSS

To assess the wireless GMVSS autopilots under stronger winds, more flight tests were conducted in a day with wind gusts of over 6 m/s. In Figs. 23 and 24, it is possible to observe that moderate wind gusts affected more the lateral and longitudinal speeds, but could not compromise the vehicle stability, confirming the robustness of the autopilots under stronger winds.

Since the data is noisy and it might be difficult to differentiate these results from previous ones using the graphics alone, it is shown in Tables 3, 4 and 5, the variances of the outputs, of the inputs and the mean values of the outputs, respectively. In these tables, the subscripts L and M are respective to light and moderate wind conditions. These performance indices were calculated based on the regulatory behavior, i.e., excluding terminal stages of takeoff and landing.

The outputs and inputs variances, respectively shown in Tables 3 and 4, allow the evaluation of the control effort or the power that was required to keep the controlled variables as close as possible to the desired values. The presented data, reassures, with a real experiment, the superiority of GMVSS when compared to an industry-standard MPC, at least in this particular case study with remote automatic control of a quadrotor.

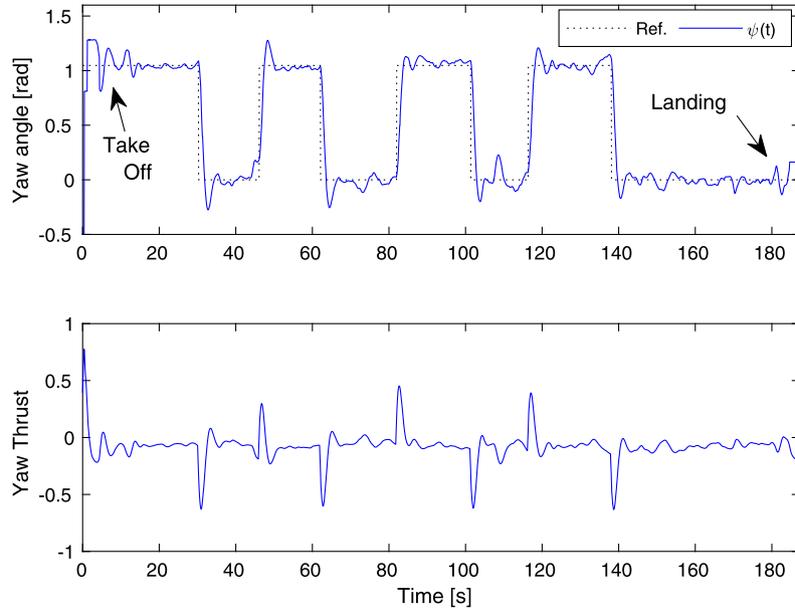


Fig. 19. GMVSS yaw-hold reference signal tracking under light wind gusts.

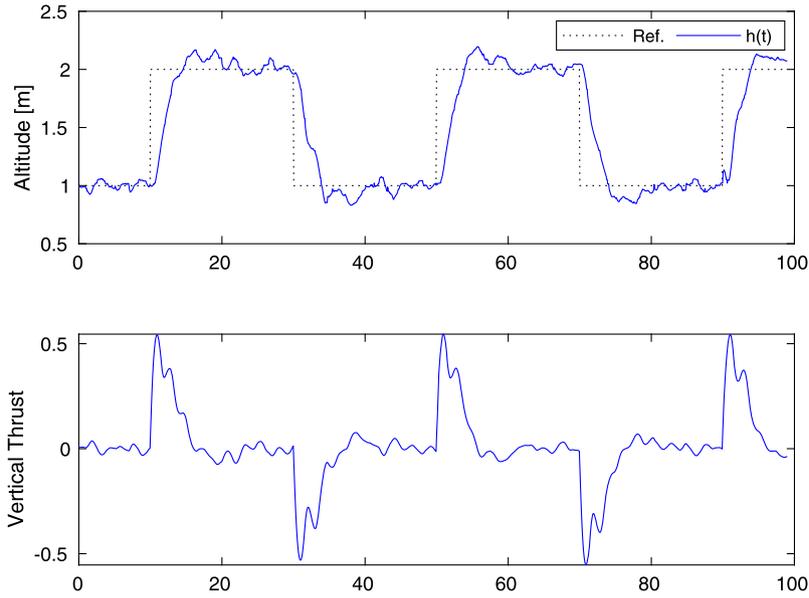


Fig. 20. GMVSS altitude-hold reference signal tracking under light wind gusts.

Table 3
Outputs variances from real flight data.

Case	$\sigma_{v_{spd}}^2$	$\sigma_{u_{spd}}^2$	σ_{ψ}^2	$\sigma_{\dot{h}}^2$
GMVSS _L	0.0100	0.0174	0.0013	0.0014
GMVSS _M	0.0651	0.0768	0.0064	0.0035
GPC _L	0.2789	0.1816	0.0001	0.0604

Table 4
Inputs variances from real flight data.

Case	$\sigma_{\dot{v}}^2$	$\sigma_{\dot{u}}^2$	$\sigma_{\dot{\psi}}^2$	$\sigma_{\dot{h}}^2$
GMVSS _L	0.0002	0.0003	0.0002	0.0003
GMVSS _M	0.0012	0.0019	0.0009	0.0008
GPC _L	0.2147	0.1766	0.0013	0.6175

Table 5
Outputs mean values from real flight data.

Case	\bar{v}_{spd}	\bar{u}_{spd}	$\bar{\psi}$	$\bar{\dot{h}}$
GMVSS _L	0.0013	0.0037	0.0008	1.0024
GMVSS _M	-0.0034	-0.0023	0.0001	1.0074
GPC _L	-0.1318	-0.1697	0.0000	0.99943

The mean values of the outputs, calculated from registered flight data and which are shown in Table 5, allow the assess-

ment of designed autopilots in terms of speed, heading and altitude average dispersion. Based on the data shown in Table 5, all designed autopilots presented mean values close to the desired values, whereas the GPC, in the heading hold function, has shown a surprisingly superior performance.

The overall result, based on the presented experimental data and focusing on the trade-off between control effort and tracking error variance minimization, is that the GMVSS could achieve superior performance for this case study on remote wireless autopilots for a quadrotor.

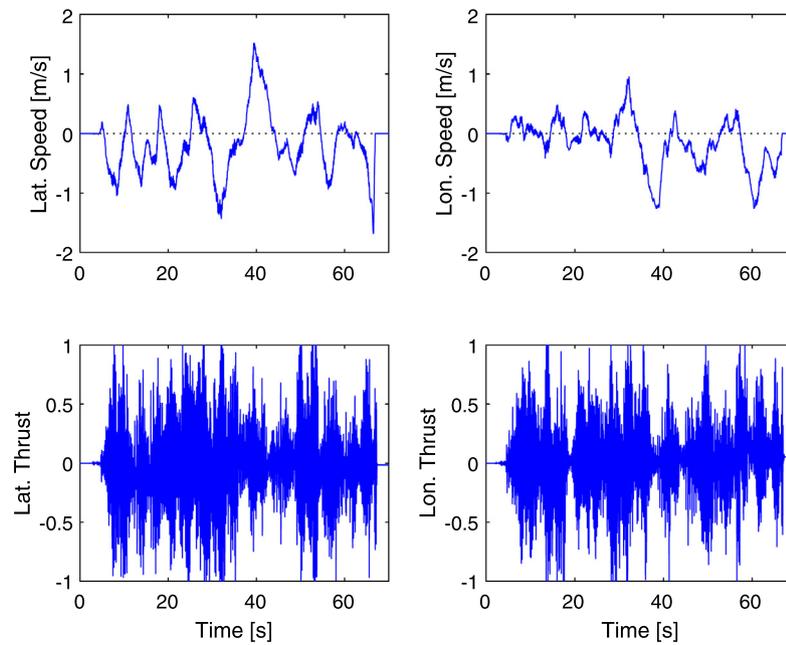


Fig. 21. GPC lateral/longitudinal speed-hold autopilots with control saturation. Real flight under light wind gusts.

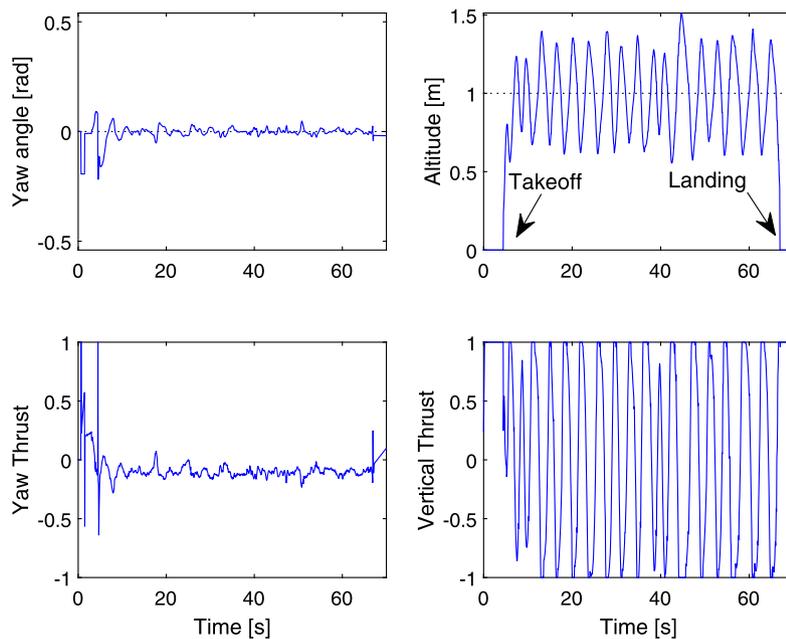


Fig. 22. GPC yaw-hold and altitude-hold autopilots with control saturation. Real flight under light wind gusts.

5. Conclusions

In this work, a non-adaptive multivariable control design technique, based on predictive generalized minimum variance control in the state-space, was proposed and applied in simulation and in real-time implementation to control a quadrotor. Designed autopilots and the flight vehicle were connected by a wireless network, where sensor noise, stochastic disturbances and time-delay were handled altogether by the proposed MIMO GMVSS method.

Assessed results, simulated and experimental, were compared to an industry-standard MPC, confirming the superiority of the MIMO GMVSS in this particular case study, handling sensor noise and stochastic disturbances, such as light and moderate wind gusts, in the following autopilot functions: lateral/longitudinal speed hold, heading hold and altitude hold.

Despite showing a superior minimum variance performance, the GMVSS presented a more demanding computational load when compared to the GPC. The average iteration time of the MIMO GMVSS algorithm was more than twice the average time of the GPC. However, stochastic MPC techniques are still in a state of flux [25] along with advances in microprocessors technology to support new algorithms and further developments of GMVSS in the adaptive, constrained and non-linear cases.

Declaration of competing interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

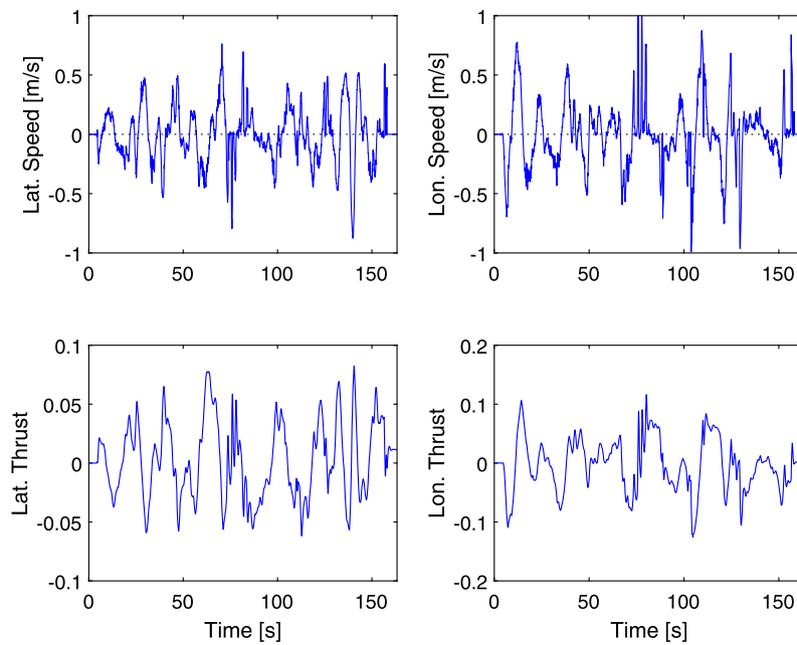


Fig. 23. GMVSS lateral/longitudinal speed-hold autopilots. Real flight under moderate wind gusts.

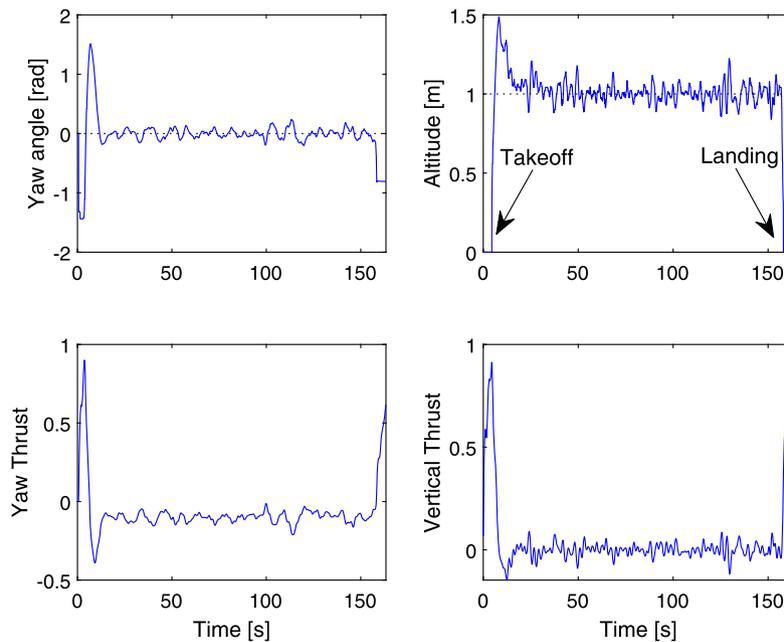


Fig. 24. GMVSS yaw-hold and altitude-hold autopilots. Real flight under moderate wind gusts.

Acknowledgements

The authors thankfully acknowledge the financial support of the Brazilian National Council for Scientific and Technological Development (CNPq) under the project number 408559/2016-0.

References

- [1] D. Brescianini, M. Hehn, R. D'Andrea, Quadcopter pole acrobatics, in: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2013, pp. 3472–3479.
- [2] F. Augugliaro, S. Lupashin, M. Hamer, C. Male, M. Hehn, M.W. Mueller, J.S. Willmann, F. Gramazio, M. Kohler, R. D'Andrea, The flight assembled architecture installation: cooperative construction with flying machines, *IEEE Control Syst. Mag.* 34 (4) (2014) 46–64, <https://doi.org/10.1109/MCS.2014.2320359>.
- [3] A.P. Schoellig, H. Siegel, F. Augugliaro, R. D'Andrea, So you think you can dance? Rhythmic flight performances with quadcopters, in: *controls and art*, in: *Controls and Art*, Springer International Publishing, 2014, pp. 73–105.
- [4] S. Lupashin, M. Hehn, M.W. Mueller, A.P. Schoellig, M. Sherback, R. D'Andrea, A platform for aerial robotics research and demonstration: the flying machine arena, *Mechatronics* 24 (1) (2014) 41–54.
- [5] Z. Wang, S. Fukushima, Control strategy for networked control systems with time delay and packet dropout using linear matrix inequalities, *EURASIP J. Wirel. Commun. Netw.* 42 (2020) 1–11, <https://doi.org/10.1186/s13638-019-1556-4>.
- [6] M. Hofer, M. Muehlebach, R. D'Andrea, Application of an approximate model predictive control scheme on an unmanned aerial vehicle, in: 2016 IEEE International Conference on Robotics and Automation (ICRA), 2016, pp. 2952–2957.
- [7] U. Eren, A. Prach, B.B. Koçer, S.V. Raković, E. Kayacan, B. Açikmese, Model predictive control in aerospace systems: current state and opportunities, *J. Guid. Control Dyn.* 40 (7) (2017) 1541–1566, <https://doi.org/10.2514/1.6002507>.

- [8] C. Greatwood, A.G. Richards, Reinforcement learning and model predictive control for robust embedded quadrotor guidance and control, *Auton. Robots* 43 (7) (2019) 1681–1693, <https://doi.org/10.1007/s10514-019-09829-4>.
- [9] M. Mammarella, M. Lorenzen, E. Capello, H. Park, F. Dabbene, G. Guglieri, M. Romano, F. Allgöwer, An offline-sampling SMPC framework with application to autonomous space maneuvers, *IEEE Trans. Control Syst. Technol.* 28 (2) (2020) 388–402.
- [10] F.A. de Almeida, Constrained dynamic compensation with model predictive control for tracking, *Aerosp. Sci. Technol.* 93 (2019), <https://doi.org/10.1016/j.ast.2019.105340>.
- [11] F. Tavakoli, A.B. Novinzadeh, Designing a closed-loop guidance system to increase the accuracy of satellite-carrier boosters' landing point, *Aerosp. Sci. Technol.* 76 (2018) 242–249, <https://doi.org/10.1016/j.ast.2018.01.031>.
- [12] M. Montazeri-Gh, A. Rasti, A. Jafari, M. Ehteshami, Design and implementation of MPC for turbofan engine control system, *Aerosp. Sci. Technol.* 92 (2019) 99–113, <https://doi.org/10.1016/j.ast.2019.05.061>.
- [13] A. Weiss, U.V. Kalabic, S.D. Cairano, Station keeping and momentum management of low-thrust satellites using MPC, *Aerosp. Sci. Technol.* 76 (2018) 229–241, <https://doi.org/10.1016/j.ast.2018.02.014>.
- [14] Q. Hu, J. Xie, X. Liu, Trajectory optimization for accompanying satellite obstacle avoidance, *Aerosp. Sci. Technol.* 82–83 (2018) 220–233, <https://doi.org/10.1016/j.ast.2018.08.033>.
- [15] K. Siddhardha, Autonomous reduced-gravity enabling quadrotor test-bed: design, modelling and flight test analysis, *Aerosp. Sci. Technol.* 86 (2019) 64–77, <https://doi.org/10.1016/j.ast.2019.01.014>.
- [16] H. Razmi, S. Afshinfar, Neural network-based adaptive sliding mode control design for position and attitude control of a quadrotor UAV, *Aerosp. Sci. Technol.* 91 (2019) 12–27, <https://doi.org/10.1016/j.ast.2019.04.055>.
- [17] H. Keshavarzian, K. Daneshjou, Modified under-actuated quadrotor model for forwarding flight in the presence of ground effect, *Aerosp. Sci. Technol.* 89 (2019) 242–252, <https://doi.org/10.1016/j.ast.2019.04.001>.
- [18] X. Wang, S. Sun, E.-J. van Kampen, Q. Chu, Quadrotor fault tolerant incremental sliding mode control driven by sliding mode disturbance observers, *Aerosp. Sci. Technol.* 87 (2019) 417–430, <https://doi.org/10.1016/j.ast.2019.03.001>.
- [19] M. Labbadi, M. Cherkaoui, Robust adaptive backstepping fast terminal sliding mode controller for uncertain quadrotor UAV, *Aerosp. Sci. Technol.* 93 (2019), <https://doi.org/10.1016/j.ast.2019.105306>.
- [20] J. Zhang, D. Gu, Z. Ren, B. Wen, Robust trajectory tracking controller for quadrotor helicopter based on a novel composite control scheme, *Aerosp. Sci. Technol.* 85 (2019) 199–215, <https://doi.org/10.1016/j.ast.2018.12.013>.
- [21] K.J. Åström, *Introduction to Stochastic Control Theory*, Dover Books on Electrical Engineering Series, Dover Publications, 1970.
- [22] D.W. Clarke, P.J. Gawthrop, Self-tuning controller, *Proc. Inst. Electr. Eng.* 122 (9) (1975) 929–934, <https://doi.org/10.1049/piee.1975.0252>.
- [23] D.W. Clarke, C. Mohtadi, P.S. Tuffs, Generalized predictive control – part I: the basic algorithm, *Automatica* 23 (2) (1987) 137–148, [https://doi.org/10.1016/0005-1098\(87\)90087-2](https://doi.org/10.1016/0005-1098(87)90087-2).
- [24] R.R. Bitmead, M. Gevers, V. Wertz, *Adaptive Optimal Control: The Thinking Man's GPC*, Prentice Hall, 1990.
- [25] B. Kouvaritakis, M. Cannon, *Model Predictive Control*, Springer, 2016.
- [26] M. Mammarella, E. Capello, F. Dabbene, G. Guglieri, Sample-based SMPC for tracking control of fixed-wing UAV, *IEEE Control Syst. Lett.* 2 (4) (2018) 611–616.
- [27] A. Silveira, R. Trentini, A. Coelho, R. Kutzner, L. Hofmann, Generalized minimum variance control under long-range prediction horizon setups, *ISA Trans.* 62 (1) (2016) 325–332.
- [28] M. Maboodi, E.F. Camacho, A. Khaki-Sedigh, Non-linear generalised minimum variance control state space design for a second-order Volterra series model, *Int. J. Syst. Sci.* 46 (14) (2015) 2607–2616, <https://doi.org/10.1080/00207721.2013.874509>.
- [29] B. Tian, L. Liu, H. Lu, Z. Zuo, Q. Zong, Y. Zhang, Multivariable finite time attitude control for quadrotor UAV: theory and experimentation, *IEEE Trans. Ind. Electron.* 65 (3) (2018) 2567–2577, <https://doi.org/10.1109/tie.2017.2739700>.
- [30] Y. Choi, H. Ahn, Nonlinear control of quadrotor for point tracking: actual implementation and experimental tests, *IEEE/ASME Trans. Mechatron.* 20 (3) (2015) 1179–1192, <https://doi.org/10.1109/TMECH.2014.2329945>.
- [31] K.J. Åström, B. Wittenmark, On self-tuning regulators, *Automatica* 9 (1973) 195–199.
- [32] F.L. Lewis, L. Xie, D. Popa, *Optimal and Robust Estimation*, 2nd edition, CRC Press. Taylor & Francis Group, 2008.
- [33] D.E. Sanabria, P.J. Mosterman, ARDrone simulink development kit v1.13, mathworks.com/matlabcentral/fileexchange, 2014 (Accessed 31 March 2018).
- [34] T.T. Mac, C. Copot, R.D. Keyser, C.M. Ionescu, The development of an autonomous navigation system with optimal control of an UAV in partly unknown indoor environment, *Mechatronics* 49 (2018) 187–196.
- [35] F. Vicario, OKID as a general approach to linear and bilinear system identification, Ph.D. thesis, Graduate School of Arts and Sciences, Columbia University, 2014.
- [36] B.L. Stevens, F.L. Lewis, E.N. Johnson, *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, 3rd edition, John Wiley & Sons, Inc., 2016.
- [37] E.F. Camacho, C.A. Bordons, *Model Predictive Control*, 2nd edition, Springer-Verlag, London, 2007.