Mechatronics 73 (2021) 102455

Contents lists available at ScienceDirect

Mechatronics

journal homepage: www.elsevier.com/locate/mechatronics

Multiple-model control architecture for a quadrotor with constant unknown mass and inertia ☆

Pedro Outeiro *, Carlos Cardeira, Paulo Oliveira

IDMEC, Instituto Superior Técnico, Universidade de Lisboa, Avenida Rovisco Pais, 1049-001 Lisboa, Portugal

ARTICLE INFO

Keywords:
Flying robots
Guidance
Navigation and control
Control systems in vehicles
Flight dynamics identification

ABSTRACT

This paper presents a methodology for height and yaw angle control of a quadrotor that transports an unknown constant load added before the flight. Based on measurements from the onboard sensors, estimates of inertial parameters – mass and z-axis inertia – and state variables – vertical position, velocity, yaw angle and rate – are provided resorting to Multiple-Model Adaptive Estimators. The number and worst case performance of Kalman filters is selected based on the Baram Proximity Measure. The proposed control methods are a steady state Linear Quadratic Regulator (LQR) with integrative action for the height, and an LQR controller for the yaw angle. The overall system obtained is validated with load variations of up to 10% of the vehicle mass, both in simulation and experimentally, resorting to an off-the-shelf commercially available quadrotor.

1. Introduction

In recent years, Unmanned Aerial Vehicles (UAV) have become a popular topic, and have seen an increase in application of research projects or military applications. Consumer level uses include racing, photography, and filming, while commercial applications include transportation and delivery, manipulation [1], inspection [2], and surveillance/monitoring [3]. Some noteworthy transportation and delivery systems are the UPS medical samples transportation system [4] already in operation, and the Amazon Prime Air [5] delivery system that is currently under development.

For transport and delivery applications, from the control system design point of view, one major concern is the nature of the load. The variable nature of the transported goods leads to an infinite possible set of characteristics about the system. The mass, inertia, and position of the item affect the dynamics of the full system and, consequently, the performance of the control. Therefore, solutions for control and estimation that are robust to uncertain parameters are of great relevance.

Control methods for quadrotors can be found in [6–10], to mention a few. In [6], a method relying on saturated feedback and backstepping control is proposed, while [7] proposes an integral predictive/nonlinear robust control structure. In [8], a sliding mode control approach for robust landing and lift-off is proposed. These works propose robust solutions to persistent disturbances, but do not approach the problem of parametric uncertainty and state-estimation. Additionally, although robust methods can handle uncertainty, the resulting solutions tend to be over-conservative, as can be seen in [11], and have the potential to fail during longer tests. In [9], PID control is proposed for a Flymobile quadrotor. In [10], PID control is proposed for a biplane quadrotor.

The problem of load transportation has been addressed, for example, in [12–14], where suspended load cases are considered. In [12] a Model Predictive Control (MPC) approach is proposed and compared with Linear Quadratic Regulator (LQR) control. Better results were obtained using the MPC approach, but stability and convergence are only shown through testing, without any formal proof. In [13], a solution for trajectory generation and control is designed exploiting the fact that the system is differentially flat. Additionally, stability and convergence proofs are provided. In [14], an adaptive solution for an unknown mass of the load is considered, relying on classical PID control.

An alternative to Robust Control that is also designed for handling unknown parameters is Multiple-model methods. Multiple-model methods for piecewise constant unknown mass of quadrotors have been considered and tested in [15,16]. In [15], the method relied on a Multiple-Model Adaptive Estimator (MMAE) with a bank of Integrative Kalman Filters and an LQR controller with integrative action. Given the set of Kalman Filters with an integrative mechanism for gravitational force, it provided reduced state-estimation error. In [16], the method

☆ This paper was recommended for publication by Associate Editor YangQuan Chen.

* Corresponding author.
E-mail addresses: pedro.outeiro@tecnico.ulisboa.pt (P. Outeiro), carlos.cardeira@tecnico.ulisboa.pt (C. Cardeira), paulo.j.oliveira@tecnico.ulisboa.pt (P. Oliveira).

https://doi.org/10.1016/j.mechatronics.2020.102455
Received 20 October 2019; Received in revised form 10 September 2020; Accepted 31 October 2020

relied on a Multiple-Model Adaptive Controller (MMAC), extending the multiple-model framework from [15] to the control.

A common method for handling non-linear and parameter uncertainty is the Extended Kalman Filter (EKF) [17]. Other alternatives, such as the Unscented Kalman Filter (UKF), have been used in aggressive flight control for quadrotors in [18]. Although the EKF and UKF methods are widespread, there are no stability or robustness guarantees.

This work studies the control and estimation for the height and yaw angle of the quadrotor. The proposed solution is a Multiple-Model Adaptive Estimator (MMAE) based architecture. The height component relies on an LQR controller with integrative action and on Kalman filters with integrative component. The yaw component uses an LQR controller and Kalman filters. The onboard sensors used are the accelerometer, the ultrasound sonar, the gyroscope, and the magnetometer, usually available onboard these types of platforms. In addition to the on-board sensors, an indoor multiple-camera motion capture system is available.

The main contribution of this paper is an architecture for estimation and control of quadrotors in the presence of parametric uncertainties in mass and in z-axis inertia. A study on the impact of the variation in the unknown parameters is included. A stability verification is performed, and the state-error is computed to be zero (disregarding dissipative effects like drag).

This paper is organized as follows: the problem addressed is described in Section 2. The physical model considered is presented in Section 3. The control architecture proposed is detailed in Section 4. The solution for estimation is presented in Section 5, and in Section 6 the solution for the control problem is proposed. The analysis of stability and of null static-error for the full proposed solution are presented in Section 7. In Section 8 the quadrotor model and its sensors are presented. The implementation details are also discussed in this section. A verification on the stability is provided in Section 9. Simulation results are presented and discussed in Section 10. The experimental results are presented and analyzed in Section 11. Finally, some concluding remarks are drawn.

2. Problem statement

The simplified height and yaw dynamics of a quadrotor are

$$M\ddot{p}_z = h\left(T, g\right) \tag{1}$$

$$I_z\ddot{\psi} = f\left(\tau_z\right), \tag{2}$$

where h is the height acceleration function, f is the yaw acceleration function, p_z is the height, M is the mass of the drone, T is the thrust, g is the gravitational acceleration, ψ is the yaw angle, I_z is its z-axis inertia, and τ_z is the z-axis moment.

The height dynamics have a constant non-linear effect in the form of the gravitational force. However, control solutions with a constant compensation of the gravitational component can be used. These solutions assume that the mass is known *a priori*. The angular acceleration is related linearly to the provided moment, therefore this is a simple problem to solve. When a load with unknown mass (M_l) and inertia (I_l) is being transported the problem becomes more complex. Re-writing the equations, results in:

$$\left(M + M_l\right)\ddot{p}_z = h\left(T, g, M_l\right) \tag{3}$$

$$\left(I_z + I_l\right)\ddot{\psi} = f\left(\tau_z, I_l\right) \tag{4}$$

Central to this work, the solution for the control is not as immediate, the performance degrades, and the platform stability is compromised. The gravitational effect influenced by M_l is unknown. Only a lower bound for the gravity effect can be known *a priori*. Additionally, the $\left(M + M_l\right)\ddot{p}_z$ and $\left(I_z + I_l\right)\ddot{\psi}$ components present an added non-linearity to the problem. Since a single linearization of this equation, close to the hovering situation with nominal load, provides a solution that only works for small variations around a specific mass and inertia, the use of standard linear solutions is not an interesting approach and alternative solutions should be exploited.

Given the non-linearity of the dynamics, the estimation problem is also harder. Additionally, the available sensors of the quadrotor do not provide a measurement of the vertical velocity, and the use of optical flow techniques relying on the ground pointing camera provides poor estimates. To tackle the optimal control problem and filter the sensory data, estimates of the state variables are required. Due to the non-linearity and the unreliability of a linearization, Kalman filters are not an option.

The roll and pitch components were not considered, as they are an equivalent problem to the yaw, having a model of similar structure.

3. Physical model

In this section, the dynamics of quadrotor are studied for the height and yaw components.

The full dynamics of a quadrotor are presented in [19], providing the equations

$$M\ddot{\boldsymbol{p}} = \boldsymbol{R}\begin{pmatrix}0\\0\\T\end{pmatrix} - Mg\begin{pmatrix}0\\0\\1\end{pmatrix}, \quad \boldsymbol{p} = \begin{pmatrix}p_x\\p_y\\p_z\end{pmatrix} \tag{5}$$

$$\boldsymbol{I}\dot{\boldsymbol{\Omega}} = -\boldsymbol{\Omega}\times\boldsymbol{I}\boldsymbol{\Omega} + \boldsymbol{\tau}, \quad \boldsymbol{\eta} = \begin{pmatrix}\phi\\\theta\\\psi\end{pmatrix}, \quad \boldsymbol{\tau} = \begin{pmatrix}\tau_x\\\tau_y\\\tau_z\end{pmatrix}, \quad \boldsymbol{I} = \begin{bmatrix}I_x & 0 & 0\\0 & I_y & 0\\0 & 0 & I_z\end{bmatrix} \tag{6}$$

$$\boldsymbol{R} = \begin{bmatrix}c\psi c\theta - s\phi s\psi s\theta & -c\phi s\psi & c\psi s\theta + c\theta s\phi s\psi\\ c\theta s\psi + c\psi s\phi s\theta & c\phi c\psi & s\psi s\theta - c\psi c\theta s\phi\\ -c\phi s\theta & s\phi & c\phi c\theta\end{bmatrix}, \quad \dot{\boldsymbol{R}} = \boldsymbol{R}\boldsymbol{\Omega}_{\times}. \tag{7}$$

In the presented model, $\boldsymbol{p}$ stands for inertial position, $\boldsymbol{\Omega}$ is the body angle rates, $\boldsymbol{\eta}$ stands for the Euler angles, M is the mass, $\boldsymbol{I}$ is the inertia matrix, $\boldsymbol{R}$ is the rotation matrix from the body fixed coordinate frame to the inertial frame, g is the gravitational acceleration, T is the thrust provided by the rotors, and $\boldsymbol{\tau}$ is the angular moments provided by the rotors. The notation $\boldsymbol{\Omega}_{\times}$ denotes the skew-symmetric matrix, such that $\boldsymbol{\Omega}_{\times}v = \boldsymbol{\Omega}\times v$ for the vector cross product $\times$ and any vector $v \in \mathbb{R}^3$. Additionally, ϕ is the roll angle, θ is the pitch angle, and ψ is the yaw angle. The abbreviations c and s stand for the cosinus and sinus trigonometric functions, respectively.

For the purpose of designing the controllers and estimators, it is necessary to isolate the relevant dynamics for the components being studied. These models are provided in the following two subsections.

3.1. Height dynamics

Taking the dynamics of a quadrotor as shown in (5)–(7) and assuming zero roll and pitch angles, the dynamics of the height can be described by

$$M\ddot{p}_z = a_z T - Mg,$$

where a_z is the thrust gain from the command input, which is assumed to be constant. Given the available measurements, these dynamics can be written in state-space form as

$$\begin{aligned}\underbrace{\begin{bmatrix}\ddot{p}_z\\\dot{p}_z\end{bmatrix}}_{\dot{\boldsymbol{x}}_z} &= \underbrace{\begin{bmatrix}0 & 0\\1 & 0\end{bmatrix}}_{\boldsymbol{A}_z}\underbrace{\begin{bmatrix}\dot{p}_z\\p_z\end{bmatrix}}_{\boldsymbol{x}_z} + \underbrace{\begin{bmatrix}\frac{a_z}{M}\\0\end{bmatrix}}_{\boldsymbol{B}_z}T - \begin{bmatrix}g\\0\end{bmatrix}\\ \underbrace{\begin{bmatrix}\ddot{p}_z\\p_z\end{bmatrix}}_{\boldsymbol{y}_z} &= \underbrace{\begin{bmatrix}0 & 0\\0 & 1\end{bmatrix}}_{\boldsymbol{C}_z}\begin{bmatrix}\dot{p}_z\\p_z\end{bmatrix} + \underbrace{\begin{bmatrix}\frac{a_z}{M}\\0\end{bmatrix}}_{\boldsymbol{D}_z}T - \begin{bmatrix}g\\0\end{bmatrix}.\end{aligned} \tag{8}$$

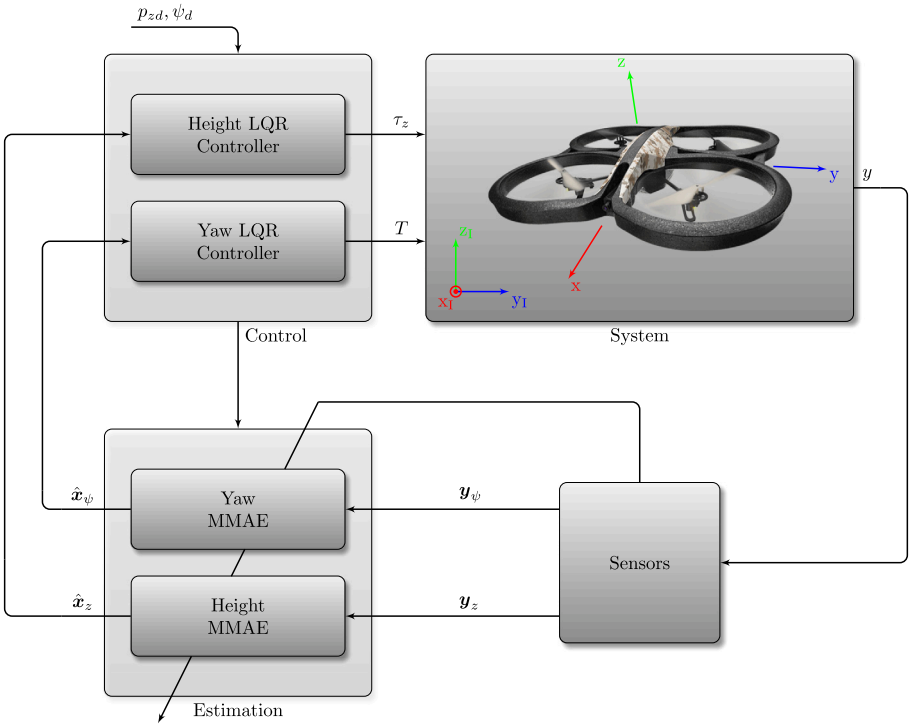


Fig. 1. Architecture with multiple-model estimation.

3.2. Yaw dynamics

Also from the dynamics of a quadrotor in (5)–(7), isolating the yaw dynamics, the resulting behavior can be described by

$$I_z\ddot{\psi} = a_\psi \tau_z + \left(I_x - I_y\right)\dot{\theta}\dot{\phi}, \tag{9}$$

where a_ψ is the z-axis moment gain from the command input. Given the available sensors and discarding the Coriolis effect, these dynamics can be written in state-space form as

$$\begin{aligned} \underbrace{\begin{bmatrix}\ddot{\psi}\\ \dot{\psi}\end{bmatrix}}_{\dot{x}_\psi} &= \underbrace{\begin{bmatrix}0 & 0\\ 1 & 0\end{bmatrix}}_{A_\psi}\underbrace{\begin{bmatrix}\dot{\psi}\\ \psi\end{bmatrix}}_{x_\psi} + \underbrace{\begin{bmatrix}\frac{a_\psi}{I_z}\\ 0\end{bmatrix}}_{B_\psi}\tau_z \\ \underbrace{\begin{bmatrix}\dot{\psi}\\ \psi\end{bmatrix}}_{y_\psi} &= \underbrace{\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}}_{C_\psi} x_\psi. \end{aligned} \tag{10}$$

3.3. Inertia behavior analysis

A particular issue for the yaw rotation is that the inertia of a system depends on the relative position of its components. To perform system identification for the inertia, no high frequency variations are expected during the test. The rotation of the rotors provides high speed variations. However, its relative center of mass remains constant. Assuming the rotor blades to be approximately flat in the z-plane, by the Parallel Axis Theorem the rotation does not affect the z-axis inertia.

The position of the load can also be an issue. To minimize effects on the x and y rotational behavior, the load is assumed to be positioned along the body-frame z-axis, making the position of the load constant and known *a priori*.

4. Control architecture

To solve the unknown load transportation problem, with stable solutions and with zero steady state error on the height and yaw angle based on accurate state estimates, the control system architecture is presented in Fig. 1. At its core, LQR control and linear Kalman filtering are the solutions foreseen. For the height components, the control uses integrative action to compensate the gravitational force, and the estimation uses an integrative component to correct the gravitational force used by the filters.

Estimators will provide the state data, while filtering the sensor noise. The control is responsible for providing the desired thrust and z-axis moment to reach the reference height and yaw angle, based on the state estimates of the filters.

The proposed architecture (Fig. 1) uses a Multiple-Model Adaptive Estimator (MMAE) for joint parameter and state estimation. The reference values for the control of the height and yaw are, respectively, p_{zd} and ψ_d. The state estimates for the height and yaw components are, respectively, $\hat{\boldsymbol{x}}_z$ and $\hat{\boldsymbol{x}}_\psi$. $\boldsymbol{y}$ is the drone data.

5. Estimation

In this Section the estimation solution is discussed. First the Kalman filters are presented. Afterwards, the multiple-model estimation framework is presented, followed by the Baram Proximity Measure (BPM), which can be used to help on the selection of the number of models. Finally, the Kalman filters with integrative component are described.

5.1. Kalman filter

The Kalman filter is an optimal filter for linear systems disturbed by zero mean white Gaussian noise. The model for these systems is

$$\begin{cases} \dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{F}\boldsymbol{v} \\ \boldsymbol{y} = \boldsymbol{C}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{u} + \boldsymbol{w} \end{cases}, \tag{11}$$

where $\boldsymbol{v}$ is the process noise and $\boldsymbol{w}$ is the sensor noise. These zero mean noises are characterized by a covariance matrix for the process noise $\boldsymbol{Q}$ and a covariance matrix for the sensor noise $\boldsymbol{R}$, respectively. The Kalman filter minimizes the covariance matrix of the error $\boldsymbol{e}$:

$$\begin{cases} \boldsymbol{e} = \boldsymbol{x} - \hat{\boldsymbol{x}} \\ \dot{\boldsymbol{e}} = (\boldsymbol{A} - \boldsymbol{L}\boldsymbol{C})\,\boldsymbol{e} + \boldsymbol{F}\boldsymbol{v} - \boldsymbol{L}\boldsymbol{w}, \end{cases} \tag{12}$$

via the selection of the optimal gain for the filter $\boldsymbol{L}$, denominated as Kalman gain. The error covariance dynamics follows the Riccati equation

$$\dot{\boldsymbol{P}} = \boldsymbol{A}\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}' - \boldsymbol{P}\boldsymbol{C}'\boldsymbol{R}^{-1}\boldsymbol{C}\boldsymbol{P} + \boldsymbol{F}'\boldsymbol{Q}\boldsymbol{F}, \tag{13}$$

where $\boldsymbol{P}$ is the covariance matrix of the error. The resulting Kalman gain calculation is given by

$$\boldsymbol{L} = \boldsymbol{P}\boldsymbol{C}'\boldsymbol{R}^{-1}. \tag{14}$$

In the steady state case, $\dot{\boldsymbol{P}}$ is zero and the popular algebraic Riccati equation results, see [20] for details.

5.2. Multiple-model adaptive estimation

The MMAE algorithm (see [21] and the references therein) is a combined state-estimation and parameter identification method. It provides a solution for parametric uncertainties and for non-linear state-estimation (using different linearizations). As the name implies, it relies on multiple models for the same system, but assuming different values of the unknown parameter (or linearization points). For each of the underlying models, a Kalman filter is designed, providing accurate estimates for its assumed model. The MMAE uses the data provided from each filter and merges it to provide better state-estimates than those obtained using a single model. Additionally, by assessing which of the models provides the best estimates, the value of the unknown parameter is estimated.

The Bayesian Posterior Probability Estimator (PPE) assesses the accuracy through the residues of the known sensory data, by assigning a probability to each filter. These probabilities are called posterior probabilities

$$p_i = \text{Prob}(H = H_i \mid Y),\ i = 1, \dots, N, \tag{15}$$

and are the probabilities of the unknown parameter H matching one of the models, given the set of past data Y. The number of models is N and the value of the parameter for each model is H_i. The continuous approximation of this discrete problem, yields the posterior probability derivatives $\dot{p}_i$. This derivative [22,23] can be computed using the probabilities p_i and the residues of the filters $\boldsymbol{r}_i$ according to

$$\dot{p}_i = -\left(1 - \frac{\boldsymbol{\beta}_i e^{-\frac{1}{2}\boldsymbol{w}_i}}{\sum_{j=1}^{n} p_j \boldsymbol{\beta}_j e^{-\frac{1}{2}\boldsymbol{w}_j}}\right) p_i \tag{16}$$

$$\boldsymbol{\beta}_i = \frac{1}{(2\pi)^{\frac{h}{2}}\sqrt{|\boldsymbol{S}_i|}} \tag{17}$$

$$\boldsymbol{w}_i = \boldsymbol{r}_i'\boldsymbol{S}_i^{-1}\boldsymbol{r}_i, \tag{18}$$

where h represents the number of sensors used. The residual covariance matrix of each filter ($\boldsymbol{S}_i$) is used as a weighting parameter in the calculations, $\boldsymbol{\beta}_i$ is a weighting parameter based on the residual covariance and number of sensors, $|\cdot|$ stands for the determinant of a square matrix, and $\boldsymbol{w}_i$ is a quadratic weighting parameter for the residue which also uses the residual covariance.

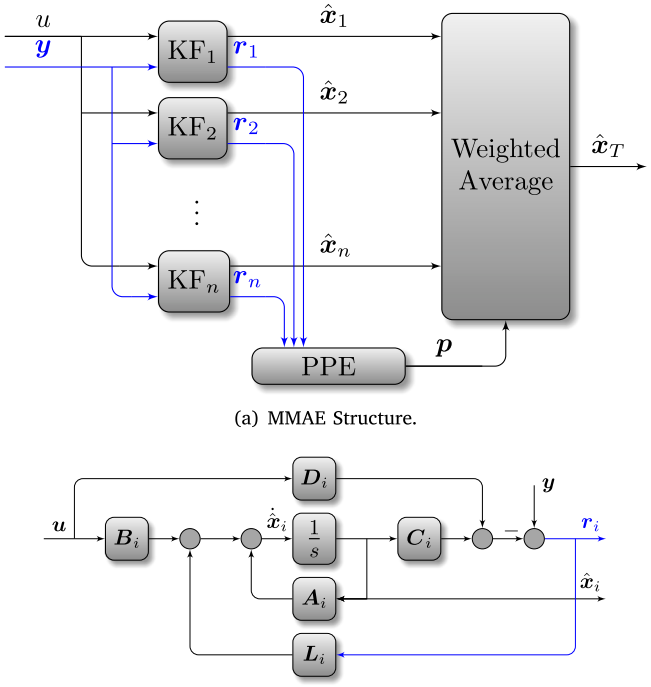
This method is required to select the value of the unknown parameter (or the closest). For that purpose, a convergence condition must be met for the probability of the closest model, as detailed in the following theorem from [23]:

Theorem 5.1. *Let $i^* \in \{1, 2, \dots, N\}$ be an index of a parameter vector in κ and $I := \{1, 2, \dots, N\} \setminus i^*$ an index set. Suppose that there exists a positive constant T such that for all $t \geq 0$ and all $j \in I$ the following condition holds:*

$$\frac{1}{T}\int_t^{t+T} (\omega_{i^*}(\tau) - ln\beta_j(\tau))d\tau < \frac{1}{T}\int_t^{t+T} (\omega_j(\tau) - ln\beta_{i^*}(\tau))d\tau. \tag{19}$$

Then, $p_{i^*}(t)$ satisfies

$$\lim_{t\to\infty} p_{i^*}(t) = 1. \tag{20}$$



(a) MMAE Structure.

(b) i[th] Kalman Filter.

Fig. 2. MMAE structure and individual Kalman filter structure.

Conversely, if (20) is observed, then there exists a positive constant T such that for all $t \geq 0$ and all $j \in I$[1]

$$\frac{1}{T}\int_t^{t+T} (\omega_{i^*}(\tau) - ln\beta_j(\tau))d\tau \leq \frac{1}{T}\int_t^{t+T} (\omega_j(\tau) - ln\beta_{i^*}(\tau))d\tau. \quad \blacksquare \tag{21}$$

Although in this work the study was made for the CT-MMAE (continuous time), the findings still hold true for the discrete equivalent.

Given the method for updating the posterior probabilities, an error in the estimation will always be present if the real value of the parameter does not match the value assumed by any of the models. The number of filters has to be selected taking into account the acceptable estimation error threshold when the system does not match any of the models. Moreover, the added computational weight of using more filters has to be considered. This will be discussed in Section 5.3.

The state estimation of the MMAE can be obtained with a switching or weighted average, see [21] for details. In switching the state estimation matches that of the filter with the highest posterior probability. In the weighted average the state-estimation of all filters is averaged using the posterior probabilities as a weighting factor:

$$\hat{\boldsymbol{x}}_T = \sum_{j=1}^{n} p_k(t)\hat{\boldsymbol{x}}_j. \tag{22}$$

In this work, the weighted average method is used, as it provides low pass filtered state estimates. The resulting structure is depicted in Fig. 2a.

The structure of the Kalman filter is presented in Fig. 2b, where s is the Laplace indeterminate.

5.3. Baram proximity measure

The Baram Proximity Measure (BPM) is a measure proposed in [24] for deciding which models to use in multiple-model, see for example [21] and [25]. The BPM provides an adequate distance metric, for

[1] Note that in (21) the inequality is not strict.

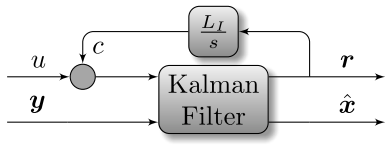


Fig. 3. Kalman filter with integrative component.

stochastic systems, between the real model and the Kalman filter with its underlying model. The BPM can be computed with:

$$\begin{aligned} BPM &= log(|\boldsymbol{S}|) + tr(\boldsymbol{S}^{-1}\boldsymbol{\gamma}) \\ \boldsymbol{\gamma} &= \frac{\boldsymbol{r}'\boldsymbol{r}}{length(\boldsymbol{r})^2}, \end{aligned} \tag{23}$$

where $\boldsymbol{S}$ is the residual covariance matrix of the filter and $\boldsymbol{r}$ is the residue obtained from using the filter.

To perform the model selection using this measure, one possible approach requires the selection of the range for the unknown parameter, and of the maximum allowed value of BPM. The first model can be selected by increasing the parameter until the BPM of the minimum value of the range does not exceed the maximum BPM. Afterwards, the following models will be required to ensure that the minimum BPM curve of the current and previous models does not exceed the maximum. Finally, models will be added until the minimum BPM curve of all models does not exceed the maximum allowed BPM.

5.4. Kalman filter with integrative component

The use of integral action in control is common for ensuring robustness to perturbations and to ensure null steady state error. Since the gravitational force is unknown and the use of multiple models does not compensate for errors in the assumed gravitational force in non-matching cases, the height filters require a mechanism that allows for the correction of the assumed force. For this purpose, the residue of the filter can be used for the adjustment of the gravitational force (c). By creating a feedback loop to the actuation input (u) with an integrator, it allows for the height estimate to follow the height measurement closely, and provides a more accurate estimate of the velocity. For tuning purposes a gain (L_I) can be given to the integration, allowing to adjust the overshoot and speed of the estimate, with a block diagram as depicted in Fig. 3.

Since the proposed method provides a solution for the gravitational force issue, it could be considered sufficient to use a single model approach. However, even disregarding the gravitational force, the mass still impacts on the dynamics of the quadrotor. Larger differences between the assumed and real values of the mass lead to higher error in the velocity estimate. Therefore, the use of multiple-model methods is still beneficial.

To prove that the Integrative Kalman filter provides zero residue for the integrated variable, it is necessary to analyze the transfer function of the sensor measurement to the estimate. Equation

$$\frac{\hat{p}_z}{p_z} = \frac{L_{22}Ms^2 + (L_{12}M - L_{21}L_I a_z)s + L_I(1 - L_{11})a_z}{Ms^3 + L_{22}Ms^2 + (L_{12}M - L_{21}L_I a_z)s + L_I(1 - L_{11})a_z} \tag{24}$$

is obtained after tedious but straightforward computations. In this function, it can be observed that the steady state gain is one, indicating that the filter follows the measurement of the height.

Additionally, the steady state gain for the actuation input has to be zero. Analyzing the transfer function

$$\frac{\hat{p}_z}{u} = \frac{a_z s}{Ms^3 + L_{22}Ms^2 + (L_{12}M - L_{21}L_I a_z)s + L_I(1 - L_{11})a_z}, \tag{25}$$

it can be seen that the steady state gain is zero.

6. Control

In this section, the control methods are described. The two proposed controllers are presented: the Linear Quadratic Regulator (LQR) and the LQR with integrative action. Additionally, the requirement of zero steady state error is studied.

6.1. LQR

The LQR is an optimal controller for linear systems. It results from solving an optimal control problem for minimizing the cost function

$$\boldsymbol{J} = \int_t^T \left[\boldsymbol{x}'\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{u}'\boldsymbol{R}\boldsymbol{u}\right] d\tau, \tag{26}$$

where the matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ are, respectively, the relative weight of the error and the relative weight of the energy in the optimization. The solution of this problem follows the Riccati equation

$$-\dot{\boldsymbol{\mathcal{M}}} = \boldsymbol{\mathcal{M}}\boldsymbol{A} + \boldsymbol{A}'\boldsymbol{\mathcal{M}} - \boldsymbol{\mathcal{M}}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}'\boldsymbol{\mathcal{M}} + \boldsymbol{Q}, \tag{27}$$

where $\boldsymbol{\mathcal{M}}$ is the minimizing matrix. The resulting LQR gain calculation is given by

$$\boldsymbol{K} = \boldsymbol{R}^{-1}\boldsymbol{B}'\boldsymbol{\mathcal{M}}. \tag{28}$$

In the steady state case, $\dot{\boldsymbol{\mathcal{M}}}$ is zero and the popular algebraic Riccati equation results, see [26] for details.

An LQR controller is proposed for the control of the yaw angle. A major requirement for the control is to ensure zero steady state error. In this case, an LQR controller achieves this condition (assuming the absence of drag effects on the dynamics). The resulting z-axis moment calculation is

$$\tau_z = -K_{\dot{\psi}}\dot{\psi} + K_{\psi}(\psi_d - \psi), \tag{29}$$

where K_{ψ} and $K_{\dot{\psi}}$ are the yaw rate and yaw gains, respectively. Given the closed loop transfer function for the system with an LQR controller

$$\frac{\psi}{\psi_d} = \frac{a_{\psi}K_{\psi}}{I_z s^2 + a_{\psi}K_{\dot{\psi}}s + aK_{\psi}}, \tag{30}$$

the steady state gain is one, and the controller provides zero steady state error. The resulting gain matrix is represented by $\boldsymbol{K} = [K_{\dot{\psi}}\ K_{\psi}]$.

The LQR design possesses phase margin of at least 60°, infinite gain margin, and gain reduction tolerance of −6 dB, which grants it some robustness.

6.2. LQR with integrative action

As mentioned in Section 2, unknown gravitational force precludes the use of standard approaches. If this component of the dynamics is not correctly compensated, there will always be a steady state error. There is, however, a version of the LQR controller that is capable of controlling a system in the presence of perturbations, like unmodeled dynamics (a relevant example in quadrotors is wind). The LQR controller with integrative action is a slight variation, consisting of a cascading controller with an inner feedback of all the state variables and an outer layer that integrates the difference between reference and current value of the control variable, thus being proposed for the height control.

To obtain a controller with these characteristics using LQR control, it is only necessary to modify the model of the dynamics when calculating the LQR gains. By using the modified version of the model

$$\boldsymbol{A}_I = \begin{bmatrix} \boldsymbol{A} & 0 \\ -\boldsymbol{C} & 0 \end{bmatrix} \quad \boldsymbol{B}_I = \begin{bmatrix} \boldsymbol{B} \\ 0 \end{bmatrix}, \tag{31}$$

there is a state variable associated with the integration that is used for defining the integrative control gain.

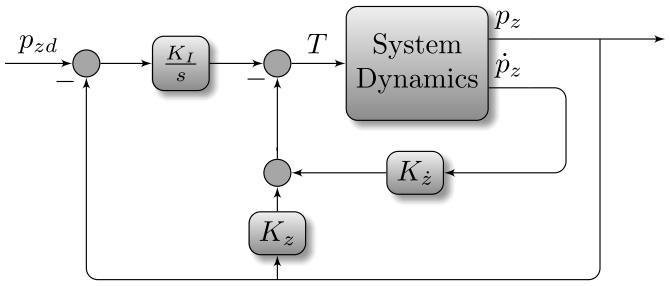


Fig. 4. LQR with integrative component.

Having a modified model, the next step is very straightforward, just calculate the LQR gains for the new model. Finally, from the resulting gains, two different sets of gains are obtained. $\boldsymbol{K}$ is the vector of gains for the state variables, $\boldsymbol{K}_{IT}$ is the gain matrix for the expanded state vector, K_z is the position gain, $K_{\dot{z}}$ is the velocity gain, and K_I is the gain for the integrative component, selected according to $\boldsymbol{K}_{calc}$:

$$\boldsymbol{K}_{calc} = [\boldsymbol{K}| - K_I] \quad \boldsymbol{K}_{IT} = [\boldsymbol{K} \quad K_I] \quad \boldsymbol{K} = [K_{\dot{z}} \quad K_z]$$

This provides the following thrust calculation:

$$T = -K_{\dot{z}}\dot{p}_z - K_z p_z + \frac{K_I}{s}\left(p_{zd} - p_z\right),$$

which leads to the structure illustrated in Fig. 4.

One of the requirements for the control is to ensure zero steady state error. To analyze the error, the system is separated into two components, one that has no gravity and receives a reference height and one that receives a zero reference height and has gravity, considered as an input.

The transfer function for the reference height and for the gravity are, respectively:

$$\frac{p_z}{p_{zd}} = \frac{K_I a_z}{Ms^3 + K_{\dot{z}}a_z s^2 + K_z a_z s + K_I a_z} \tag{32}$$

$$\frac{p_z}{g} = \frac{Ms}{Ms^3 + K_{\dot{z}}a_z s^2 + K_z a_z s + K_I a_z}. \tag{33}$$

In steady state, the gain associated to the reference height case is unitary, which means that it goes to the desired height. In the gravity case, the gain is zero, which implies that the gravity causes no deviation from the desired height.

From this, it is concluded that the addition of the integrative action provides a zero steady state error solution.

7. Linear stability analysis and steady state error analysis

The stability and assurance of zero static error of the proposed control system architecture are central to the operation of autonomous aerial vehicles in the presence of unknown constant parameters. To analyze this issue, the following lemma from [21] is instrumental:

Lemma 7.1. *In the case where the real system has an unknown constant parameter that matches the underlying model of one of the filters in the filter bank, the corresponding posterior probability will tend to one and all the other probabilities will go to zero.* ■

Thus the following lemma can be stated:

Lemma 7.2. *For a system in the conditions of the previous lemma and based on the Separation Theorem, there is a finite time instant T_{fin} such that, for $t > T_{fin}$, all variables of the closed loop control system are bounded and the yaw rate error converges to zero.* ■

The proof of the lemma is based on the assumptions previously outlined, namely assuming that one Kalman filter is based on an underlying model with the correct parameter. Thus, the eigenvalues of the controller and the estimator are recovered. By the separation principle, if the controller is stable and the estimator is stable, then the overall system is stable.

7.1. Height control

The resulting $\boldsymbol{A}_z$ matrix of the system is

$$\boldsymbol{A}_z = \begin{bmatrix} 0 & 0 & -\frac{K_{\dot{z}}a_z}{M_r} & -\frac{K_z a_z}{M_r} & \frac{K_I a_z}{M_r} & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & L_{12} & A_{33} & A_{34} & A_{35} & A_{36} \\ 0 & L_{22} & \frac{K_{\dot{z}}L_{21}a_z}{M_m} + 1 & \frac{K_z L_{21}a_z}{M_m} - L_{22} & -\frac{K_I L_{21}a_z}{M_m} & -\frac{L_I L_{21}a_z}{M_m} \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 \end{bmatrix}$$

$$A_{33} = -\frac{K_{\dot{z}}a_z}{M_m}\left(1 - L_{11}\right) \quad A_{34} = -L_{12} - \frac{K_z a_z}{M_m}\left(1 - L_{11}\right)$$

$$A_{35} = \frac{K_I a_z}{M_m}\left(1 - L_{11}\right) \quad A_{36} = \frac{L_I a_z}{M_m}\left(1 - L_{11}\right), \tag{34}$$

where M_m is the model mass, M_r is the real mass, a_z is an input factor, L are the Kalman gains and K are the controller gains. Additionally, the $\boldsymbol{B}_z$ matrix for the reference value is

$$\boldsymbol{B}_z = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}^T, \tag{35}$$

and the $\boldsymbol{C}_z$ matrix for the system height is

$$\boldsymbol{C}_z = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{36}$$

The steady state is analyzed using

$$\boldsymbol{G}_z = -\boldsymbol{C}_z \boldsymbol{A}_z^{-1} \boldsymbol{B}_z, \tag{37}$$

which yields a gain of one, as intended.

Additionally, to analyze the steady state error induced by the gravitational force, the matrix

$$\boldsymbol{B}_z = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{38}$$

is used instead of the previous one. Re-using (37), the steady state gain obtained is zero, proving that the unknown gravitational force does not cause steady state error.

7.2. Yaw control

The resulting $\boldsymbol{A}_\psi$ matrix of the system is

$$\boldsymbol{A}_\psi = \begin{bmatrix} 0 & 0 & -\frac{K_\psi a_\psi}{I_r} & -\frac{K_\psi a_\psi}{I_r} \\ 1 & 0 & 0 & 0 \\ L_{11} & L_{12} & -L_{11} - \frac{K_\psi a_\psi}{I_m} & -L_{12} - \frac{K_\psi a_\psi}{I_m} \\ L_{21} & L_{22} & 1 - L_{21} & -L_{22} \end{bmatrix}, \tag{39}$$

where I_m is the model inertia, I_r is the real inertia, a_ψ is an input factor, L are the Kalman gains and K are the controller gains. Additionally, the $\boldsymbol{B}_\psi$ matrix for the reference value is

$$\boldsymbol{B}_\psi = \begin{bmatrix} \frac{K_\psi a_\psi}{I_r} & 0 & \frac{K_\psi a_\psi}{I_m} & 0 \end{bmatrix}^T, \tag{40}$$

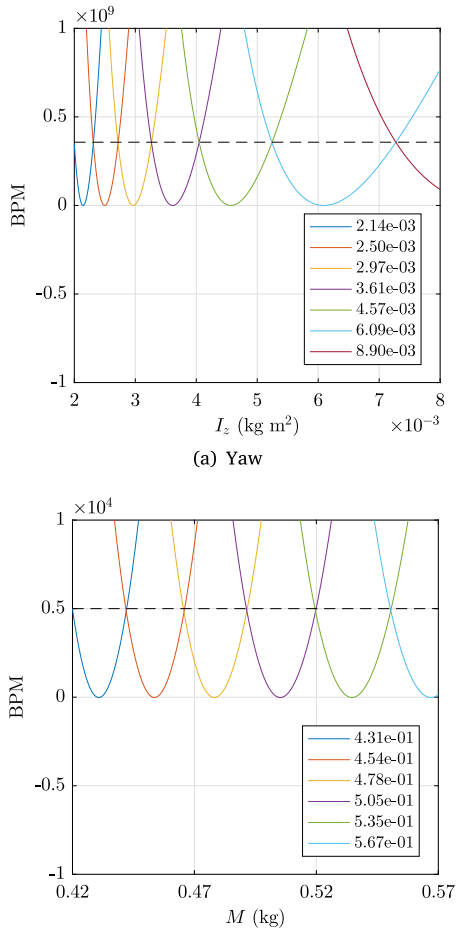
and the $\boldsymbol{C}_\psi$ matrix for the system yaw angle is

$$\boldsymbol{C}_\psi = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix}. \tag{41}$$

The steady state is analyzed using

$$\boldsymbol{G}_\psi = -\boldsymbol{C}_\psi \boldsymbol{A}_\psi^{-1} \boldsymbol{B}_\psi, \tag{42}$$

which yields a gain of one, as intended.



(a) Yaw

(b) Height

Fig. 5. BPM curves.

Table 1
Nominal parameters.

M (kg)	I_x (kg m^2)	I_y (kg m^2)	I_z (kg m^2)
0.475	2.2383×10^{-3}	2.9858×10^{-3}	4.8334×10^{-3}

8. Implementation

The model used for assessing the performance of the proposed solution is the Parrot Ar.Drone 2.0 by Parrot SA. It is an off-the-shelf commercially available, general purpose quadrotor designed for users without any drone piloting skills. The nominal parameters of the drone are presented in Table 1.

This model is equipped with an Inertial Measurement Unit (IMU) composed of a triad of accelerometers, a three-axis gyroscope and a three-axis magnetometer. Other sensors are available onboard such as an ultrasound sensor, a barometer and two cameras. One camera on the front and an optical-flow camera on the bottom. The rotors of the quadrotor receive PWM commands. Therefore, before the actuation inputs are sent to the quadrotor, these are converted to the equivalent PWM commands for each rotor.

The initial value of the posterior probabilities p_i in the MMAE, known as the *a priori* probabilities, are initialized equally for the N filters $(1/N)$. This initialization is commonly used when there is no *a priori* knowledge to support higher or lower probability at start.

The model selection was performed using the BPM analysis presented in Section 5.3, with the objective of assessing the number of required models and the value of the unknown parameter for each model. The BPM analysis for the yaw component of the problem addressed in this paper yields the results in Fig. 5a. Note that, for a maximum BPM of 3.6×10^8 and for the range of values presented, the obtained number of models is seven and the resulting inertia values assumed for each model are as presented in the legend of the figure.

A similar analysis for the problem addressed in this paper, with only a height and z-axis acceleration, yields the results in Fig. 5b. Here, it can be observed that, for a maximum BPM of 5000 and for the range of values presented, the obtained number of models is six and the mass values assumed for each model are as presented in the legend of Fig. 5b.

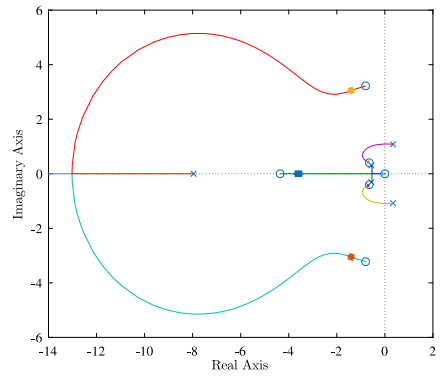
For the purpose of implementing the control system in the experimental test, the AR.Drone 2.0 Quadcopter Embedded Coder [27] was used. This package provides a target, deploying the code generated from the developed Simulink to the quadrotor, and using the Simulink during runtime for data logging and sending commands. Wi-Fi communication is used for deployment, data-logging, and sending commands. This package was chosen because it provides a Simulink based environment for development, and allows direct access to the sensors and actuators.

To ensure the stabilization of the quadrotor, the X and Y position and angles were regulated using a cascading PID control system for defining the necessary torques. The sensors used were the accelerometer that provides the accelerations, the ultrasound sonar that provides the height, the gyroscope that provides the angular rates, and the magnetometer that provides the yaw angle. An indoor multiple-camera motion capture system (Qualisys) was used for obtaining the X and Y position and angles. The setup of the experiment is presented in Fig. 6. The load with constant unknown mass and inertia is added before the flight. However, if mass or inertia change along the flights the models will estimate the new mass and inertia and adjust the control for the new values.

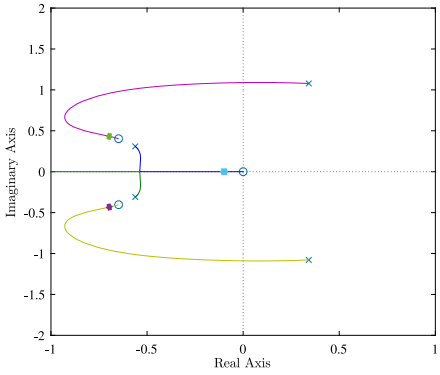
Fig. 6. Experimental setup.

9. Stability verification

In this section, a simple verification of the stability of the system is provided. For the height, the design mass is 0.445 kg, and the real mass is 0.47 kg. The obtained set of eigenvalues is $eig = \{-1.4648 \pm 3.0317i, -0.7023 \pm 0.4351i, -3.5159, -0.1077\}$. As expected, all eigenvalues have a negative real component and thus the overall closed loop system is stable. For the yaw, the design inertia is 5.1×10^{-3} kg m^2, and the

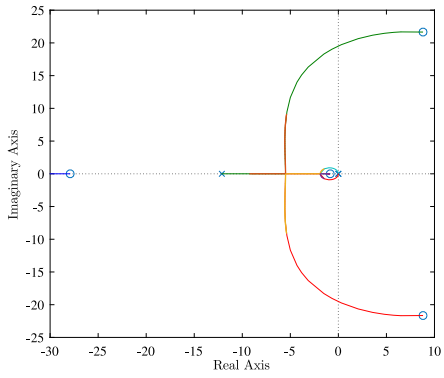


(a) Full

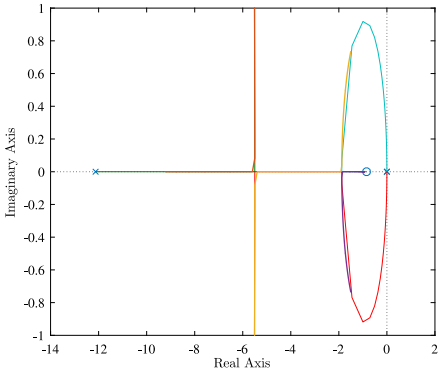


(b) Zoom

Fig. 7. Root locus mass.



(a) Full



(b) Zoom

Fig. 8. Root locus inertia.

real inertia is 4.8×10^{-3} kg m^2. The obtained set of eigenvalues is $eig = \{-0.9873,\ -5.563 \pm 5.981i,\ -196.1\}$. As expected, all eigenvalues have a negative real component and thus the overall closed loop system is stable.

Additionally, to analyze the effect of the value of the model mass on the stability, a root locus like analysis is performed. Using

$$f_z = \boldsymbol{C}_z(s\boldsymbol{I} - \boldsymbol{A}_z)^{-1}\boldsymbol{B}_z \tag{43}$$

with (34), (35), and (36), this provides the transfer function of the system. Taking the resulting denominator $den(f_z)$, it can be re-written in the form of

$$den(f_z) = f_1 \boldsymbol{M}_m + f_2. \tag{44}$$

Plotting the root locus for f_1/f_2, allows for the desired analysis and produces the graph in Fig. 7. There is a pole in the origin for infinite mass, and there are two eigenvalues with positive real part for masses below 0.035 kg. All of these mass values are outside the range specified in this paper, ensuring the stability of the system. The eigenvalues for the minimum and maximum values of the mass considered in this article are highlighted with thicker lines.

Finally, to analyze the effect of the value of the model inertia on the stability, a root locus like analysis is performed. Using

$$f_\psi = \boldsymbol{C}_\psi(s\boldsymbol{I} - \boldsymbol{A}_\psi)^{-1}\boldsymbol{B}_\psi \tag{45}$$

with (39), (40), and (41), this provides the transfer function of the system. Taking the resulting denominator $den(f_\psi)$, it can be re-written in the form of

$$den(f_\psi) = f_1 \boldsymbol{I}_m + f_2. \tag{46}$$

Plotting the root locus for f_1/f_2, allows for the desired analysis and produces the graph in Fig. 8. There are two eigenvalues in the origin for zero inertia, and there are two eigenvalues with positive real part for inertia above 0.035 kg m^2. All of these inertia values are outside the range specified in this paper, ensuring the stability of the system. The eigenvalues for the minimum and maximum values of the inertia considered in this article are highlighted with thicker lines.

10. Simulation results

To validate if the proposed solution works, a simulation was prepared. The parameters of the drone used for its setup are a mass M of 0.454 kg without a load and 0.473 kg with a load, an inertia I_z of $2.966{\times}10^{-3}$ kg m^2 without a load and $4.833{\times}10^{-3}$ with a load. The input factors a_z and a_ψ are set to one. The model values for the multiple-model architecture are the same as the ones presented in Section 8. For the purpose of the Kalman gain computations, the covariance of the sensor noise are defined as $R_{\ddot{z}} = 0.03$ for the z-axis acceleration, $R_z = 2.4 \times 10^{-7}$ for the height, $R_{\dot{\psi}} = 0.03$ for the yaw rate, and as $R_\psi = 5{\times}10^{-7}$ for the yaw, while the process noise of the height is given a covariance of $Q_z = 0.005$, and the process noise of the yaw is given a covariance of $Q_\psi = 10^{-6}$. The height LQR gains are $\boldsymbol{K} = [0.959\ \ 1.114]$ and $K_I = 0.477$. The yaw LQR gains are $\boldsymbol{K} = [9.436\ \ 8.968] \times 10^{-2}$. The reference value for the height is a unitary step beginning at the start of the simulation, and for the yaw is a square wave of amplitude $\pi/2$. The height control results are presented first, followed by the yaw results. First, the results for the height and yaw control are analyzed with the load being added mid-flight, at the start of the third rotation. Afterwards, a comparison results with changes of −20% to 20% from the drone real values is performed.

10.1. Height control

The results for the height are presented in Fig. 9. The settling time (5%) is at 4 s. There is no overshoot, but there was an undershoot. The

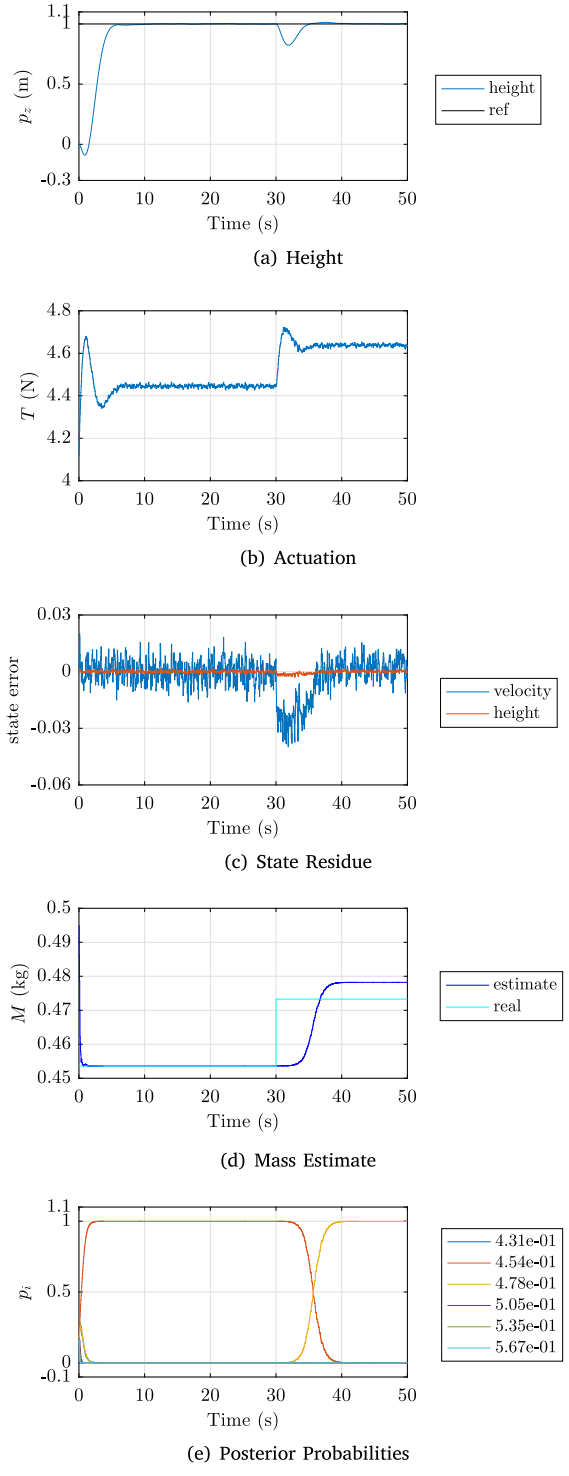


Fig. 9. Height simulation results.

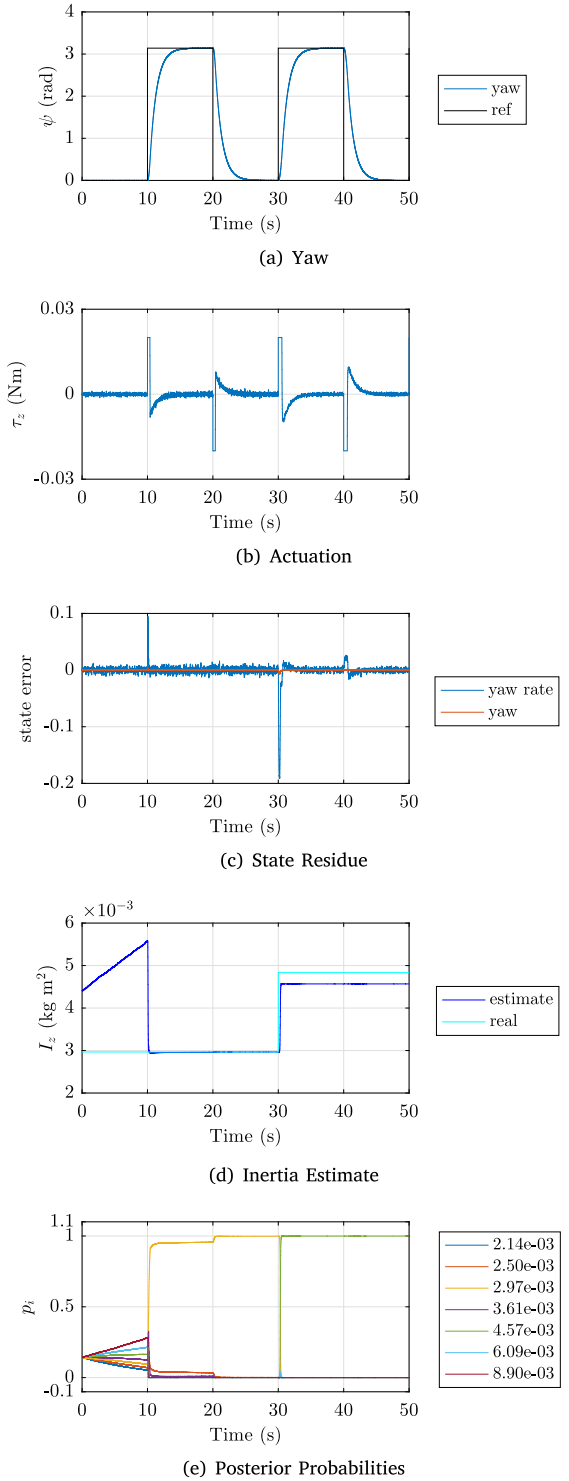
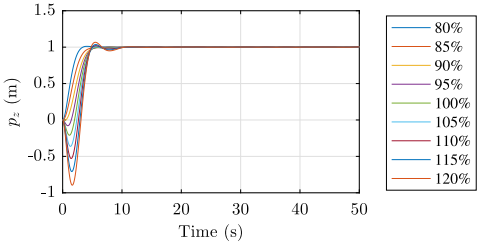


Fig. 10. Yaw simulation results.

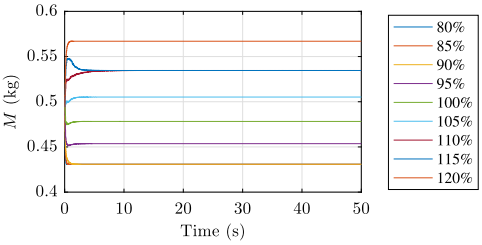
actuation does not saturate. The estimation error for the height is at most 0.002 m, while for the velocity it peaks at about 0.04 m/s, after adding the load. The model selection ended after 2 s at the start of the test and after 10 s after adding the load. The selection of the closest model is observed in both phases. The mass estimate for the no load phase converged to the correct value, because it is a matching case. In the load phase, the inertia has an error, as it is a non-matching case. It can be observed that after the load is added, the steady-state error continues to be zero.

10.2. Yaw control

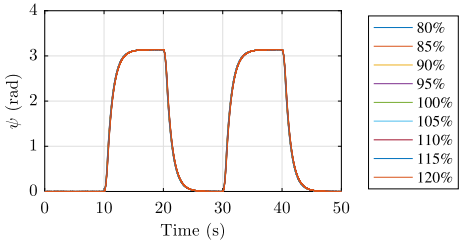
The results for the height are presented in Fig. 10. The settling times (5%) are at 4 s, and there is no overshoot. The actuation only saturated at the start of each rotation. The estimation error zeroes after 1 s, and peaks at the start of the rotations. The peaks in the load phase (approx. 0.2 max for the yaw rate) are larger due to it being a non-matching case, causing higher error. There is a large peak during the first rotation of each phase that is caused by the model selection and is not repeated in the following rotations. The maximum probability is attributed to



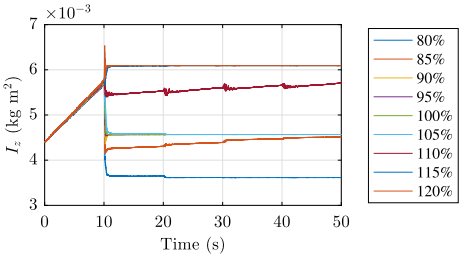
(a) Height



(b) Mass Estimate



(c) Yaw



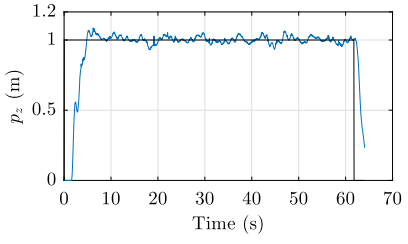
(d) Inertia Estimate

Fig. 11. Tests with different masses/inertias (percentage value of quadrotor nominal values indicated in the legends).

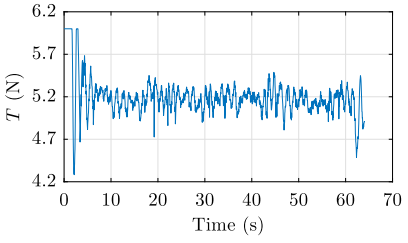
the closest model, but the highest inertia model is selected initially. There is an error in the inertia estimate of the load phase, as it is a non-matching case. The posterior probability of the closest model for the no load phase reaches 1 during the second rotation, while for the load phase it achieves maximum probability during the first rotation.

10.3. System parameters change comparison

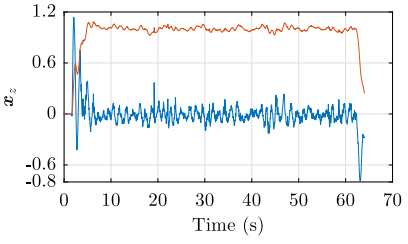
The results in Fig. 11 show changes from −20% to 20% from the drone nominal values (use of a lighter battery, different load, different drone guard, etc.). The settling time of the height increases with the increasing mass, but achieves low overshoot. The settling time of the yaw is identical in all tests. The mass estimates converge in less than 5 s. Most estimates of the inertia converge in the two first rotations. However, the 85% case almost converged only by the end of the simulation, and the 110% case is seen converging, but would require further excitation to converge.



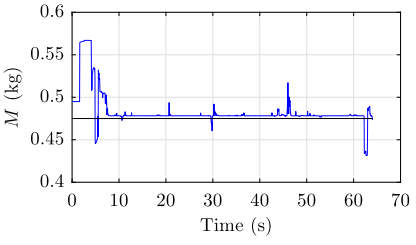
(a) Height



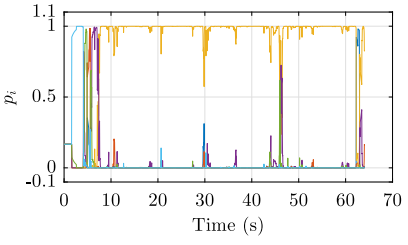
(b) Actuation



(c) State Estimate



(d) Mass Estimate

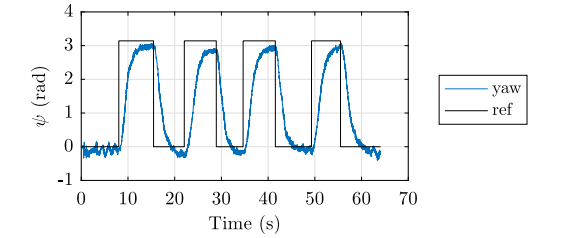


(e) Posterior Probabilities

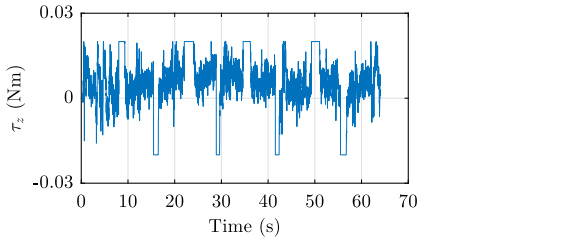
Fig. 12. Height experimental results.

11. Experimental results

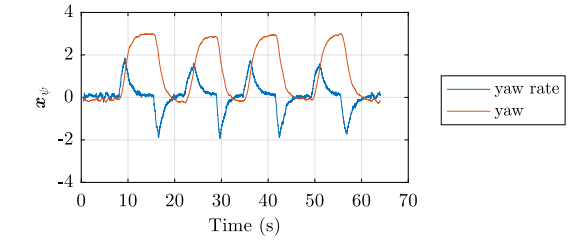
To test if the proposed solutions works, an experimental test was prepared. The real parameters of the drone are a mass M of 0.475 kg and an inertia I_z of 4.8334×10^{-3} kg m^2. The input factor a_z is set to 0.9 and a_ψ is set to 0.52. No test was performed with a load, as it was difficult to increase significantly the inertia without affecting the remainder of the dynamics. The model values for the multiple-model architectures are the same as the ones presented in Section 8. For the purpose of the Kalman gain calculations, the covariance of the sensor noise is defined as $R_{\ddot{z}} = 0.03$ for the z-axis acceleration, $R_z = 4.6 \times 10^{-7}$



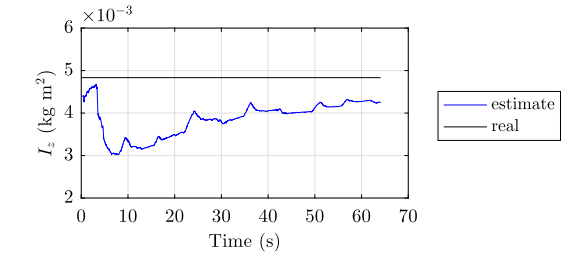
(a) Yaw



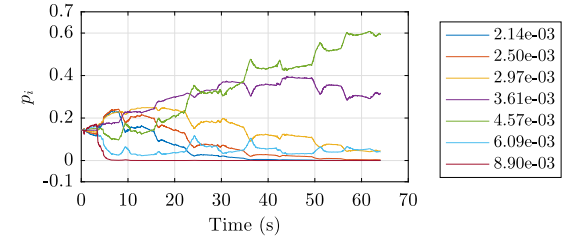
(b) Actuation



(c) State Estimate



(d) Inertia Estimate



(e) Posterior Probabilities

Fig. 13. Yaw experimental results.

for the height, $R_{\dot{\psi}} = 0.03$ for the yaw rate, and as $R_{\psi} = 4.26 \times 10^{-4}$ for the yaw, while the process noise of the height is given a covariance of $Q_z = 0.00$, and the process noise of the yaw is given a covariance of $Q_{\psi} = 10^{-4}$. The height LQR gains are $\boldsymbol{K} = [1.223\ 1.879]$ and $K_I = 1.330$. The yaw LQR gains are $\boldsymbol{K} = [9.446\ \ 8.975] \times 10^{-2}$. The tests were performed by lifting-off to a height of 1 m, followed by eight 180° rotations, and ending with landing.

11.1. Height control

The results of the test are presented in Fig. 12. The settling time (5%) with a desired height of 1 m is at 7 s with a 9% overshoot. The actuation only saturated during the lift-off. The highest probability is attributed to the closest model. There are peaks in the probabilities of the other models at the during lift-off and landing, but this does not seem to affect the selection of the closest model. The model selection in this experiment is more sensitive and slower than in the simulation, taking 8 s to settle. The mass estimate has a low error, as it is a matching case.

11.2. Yaw control

The results of the test are presented in Fig. 13. The settling time (5%) with 180° rotations is at 4 s with no overshoot. There is a static error in all rotations, where the maximum observed value is 0.25 rad. The actuation only saturates during the start of each rotation. The probabilities evolve in a similar fashion to what was observed in simulation, changing mostly during the rotations, but in a much slower manner. The highest probability is attributed to the closest model after 35 s. The inertia estimate has a low error, as it is a matching case.

12. Conclusion

This paper proposed and studied the application of Multiple-Model Kalman filtering and LQR control for transportation of unknown loads with quadrotors. The unknown parameters tested were the mass and z-axis inertia of the load. The sensors used for the proposed solution were an accelerometer, an ultrasound sonar, a gyroscope, and a magnetometer. The solution was studied in simulation and experimentally using the Ar.Drone 2.0. The control and estimation systems provided low settling times and no overshoot in simulation. Additional simulations were performed with variations of up to 20% from the drone nominal values. For the purpose of limiting the overshoot, variations of up to 10% are recommended. In the experiment, the settling times increased slightly and a small overshoot was observed in the height. Zero steady state error was achieved in simulation for the height and yaw angle and experimentally only for the height. The parameter estimation from the MMAE was capable of selecting the closest model in the filter bank for both mass and inertia. Parameter errors were only detected for non-matching cases.

CRediT authorship contribution statement

Pedro Outeiro: Methodology, Software, Validation, Writing - original draft. **Carlos Cardeira:** Supervision, Writing - review & editing, Project administration, Funding acquisition. **Paulo Oliveira:** Supervision, Writing - review & editing, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is financed by national funds through FCT, Portugal – Foundation for Science and Technology, I.P., through IDMEC, under LAETA, projects UIDB/50022/2020, and REPLACE (LISBOA-01-0145-FEDER-032107). This work was also supported by FCT through the scholarship SFRH/BD/147035/2019.

References

[1] Jimenez-Cano A, Martin J, Heredia G, Ollero A, Cano R. Control of an aerial robot with multi-link arm for assembly tasks. In: 2013 IEEE international conference on robotics and automation (ICRA). 2013, p. 4916–21. http://dx.doi.org/10.1109/ICRA.2013.6631279.

[2] Sanchez-Cuevas PJ, Heredia G, Ollero A. Multirotor UAS for bridge inspection by contact using the ceiling effect. In: 2017 international conference on unmanned aircraft systems (ICUAS). 2017, p. 767–74. http://dx.doi.org/10.1109/ICUAS.2017.7991412.

[3] Merino L, Caballero F, Martínez-de-Dios JR, Maza I, Ollero A. An unmanned aircraft system for automatic forest fire monitoring and measurement. J Intell Robot Syst 2012;65(1):533–48. http://dx.doi.org/10.1007/s10846-011-9560-x.

[4] UPS partners with matternet to transport medical samples via drone across hospital system in Raleigh, N.C.. 2019, https://pressroom.ups.com/pressroom/ContentDetailsViewer.page?ConceptType=PressReleases&id=1553546776652-986, [Online; accessed 16-April-2019].

[5] Amazon prime air website. 2019, https://www.amazon.com/Amazon-Prime-Air/b?node=8037720011, [Online; accessed 16-April-2019].

[6] Casau P, Sanfelice RG, Cunha R, Cabecinhas D, Silvestre C. Robust global trajectory tracking for a class of underactuated vehicles. Automatica 2015;58:90–8. http://dx.doi.org/10.1016/j.automatica.2015.05.011, URL http://www.sciencedirect.com/science/article/pii/S0005109815002101.

[7] Raffo GV, Ortega MG, Rubio FR. An integral predictive/nonlinear H∞ control structure for a quadrotor helicopter. Automatica 2010;46(1):29–39. http://dx.doi.org/10.1016/j.automatica.2009.10.018, URL http://www.sciencedirect.com/science/article/pii/S0005109809004798.

[8] Cabecinhas D, Naldi R, Silvestre C, Cunha R, Marconi L. Robust landing and sliding maneuver hybrid controller for a quadrotor vehicle. IEEE Trans Control Syst Technol 2016;24(2):400–12. http://dx.doi.org/10.1109/TCST.2015.2454445.

[9] Jeong S, Jung S. A quad-rotor system for driving and flying missions by tilting mechanism of rotors: From design to control. Mechatronics 2014;24(8):1178–88. http://dx.doi.org/10.1016/j.mechatronics.2014.09.006, URL http://www.sciencedirect.com/science/article/pii/S0957415814001330.

[10] Chipade VS, Abhishek, Kothari M, Chaudhari RR. Systematic design methodology for development and flight testing of a variable pitch quadrotor biplane VTOL UAV for payload delivery. Mechatronics 2018;55:94–114. http://dx.doi.org/10.1016/j.mechatronics.2018.08.008, URL http://www.sciencedirect.com/science/article/pii/S0957415818301351.

[11] Lehtomaki N, Sandell N, Athans M. Robustness results in linear-quadratic Gaussian based multivariable control designs. IEEE Trans Automat Control 1981;26(1):75–93. http://dx.doi.org/10.1109/TAC.1981.1102565.

[12] Notter S, Heckmann A, Mcfadyen A, Gonzalez LF. Modelling, simulation and flight test of a model predictive controlled multirotor with heavy slung load. 49, 2016, http://dx.doi.org/10.1016/j.ifacol.2016.09.032,

[13] Sreenath K, Michael N, Kumar V. Trajectory generation and control of a quadrotor with a cable-suspended load - a differentially-flat hybrid system. In: 2013 IEEE international conference on robotics and automation. 2013, p. 4888–95. http://dx.doi.org/10.1109/ICRA.2013.6631275.

[14] Dai S, Lee T, Bernstein DS. Adaptive control of a quadrotor UAV transporting a cable-suspended load with unknown mass. In: 53rd IEEE conference on decision and control. 2014, p. 6149–54. http://dx.doi.org/10.1109/CDC.2014.7040352.

[15] Outeiro P, Cardeira C, Oliveira P. LQR/MMAE height control system of a quadrotor for constant unknown load transportation. In: 2018 13th APCA international conference on automatic control and soft computing (CONTROLO). 2018, p. 389–94. http://dx.doi.org/10.1109/CONTROLO.2018.8514545.

[16] Outeiro P, Cardeira C, Oliveira P. MMAC height control system of a quadrotor for constant unknown load transportation. In: 2018 IEEE/RSJ international conference on intelligent robots and systems (IROS). 2018, p. 4192–7. http://dx.doi.org/10.1109/IROS.2018.8594215.

[17] Leishman RC, Macdonald JC, Beard RW, McLain TW. Quadrotors and accelerometers: State estimation with an improved dynamic model. IEEE Control Syst Mag 2014;34(1):28–41. http://dx.doi.org/10.1109/MCS.2013.2287362.

[18] Loianno G, Brunner C, McGrath G, Kumar V. Estimation, control, and planning for aggressive flight with a small quadrotor with a single camera and IMU. IEEE Robot Autom Lett 2017;2(2):404–11. http://dx.doi.org/10.1109/LRA.2016.2633290.

[19] Mahony R, Kumar V, Corke P. Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. IEEE Robot Autom Mag 2012;19(3):20–32. http://dx.doi.org/10.1109/MRA.2012.2206474.

[20] Gelb A. Applied optimal estimation. MIT press; 1974.

[21] Fekri S. Robust adaptive MIMO control using multiple-model hypothesis testing and mixed-μ synthesis (Ph.D. thesis), Instituto Superior Técnico; 2002.

[22] Chang CB, Athans M. State estimation for discrete systems with switching parameters. IEEE Trans Aerosp Electron Syst 1978;AES-14(3):418–25. http://dx.doi.org/10.1109/TAES.1978.308603.

[23] Hassani V, Pedro Aguiar A, Pascoal AM, Athans M. Further results on plant parameter identification using continuous-time multiple-model adaptive estimators. In: Proceedings of the 48h IEEE conference on decision and control (CDC) held jointly with 2009 28th Chinese control conference. 2009, p. 7261–6. http://dx.doi.org/10.1109/CDC.2009.5400434.

[24] Baram Y. Information, consistent information and dynamic system identification (Ph.D. thesis), Massachusetts Institute of Technology; 1976.

[25] Gaspar T, Oliveira P, Silvestre C. Model-based filters for 3-D positioning of marine mammals using AHRS- and GPS-equipped UAVs. IEEE Trans Aerosp Electron Syst 2015;51(4):3307–20. http://dx.doi.org/10.1109/TAES.2015.140748.

[26] Friedland B. Control system design: An introduction to state-space methods. Dover Publications; 2005.

[27] Lee D. AR.Drone 2.0 support from embedded coder. 2017, https://www.mathworks.com/hardware-support/ar-drone.html.

Pedro Outeiro is a Ph.D. student from Instituto Superior Técnico (IST), Lisbon, Portugal. He completed his master's degree in mechanical engineering with a thesis on Control and Estimation Methods for Unknown Load Transportation with Quadrotors. He received three diplomas for academic excellence and one for academic merit during his studies.

Carlos B. Cardeira was born in Quinjenje, Angola, and received the engineering and master of science degrees in 1986 and 1991, in electrical engineering from Instituto Superior Técnico in Lisbon — Portugal. In 1994 he received the Ph.D. in electrical engineering and computer science from the Institut National Polytechnique de Lorraine in Nancy — France. He is a member of the Center of Intelligent Systems of the IDMEC research laboratory and teaches at Instituto Superior Técnico in Lisbon courses in Mechatronics Systems, Industrial Automation and Informatics areas. He made several post-docs and sabbatical leaves, namely in IRIT and LAAS in Toulouse — France, CERN in Geneva — Switzerland and Schneider-Electric in Seligenstadt — Germany.

Paulo Oliveira (SM IEEE) received the Ph.D. degree in Electrical and Computer Engineering in 2002, and the Habilitation in Mechanical Engineering in 2016, all from Instituto Superior Técnico (IST), Lisbon, Portugal. He holds a joint position as Full Professor in the Mechanical Engineering and Electrotechnical and Computer Engineering Departments of IST, since 2020, and Senior Researcher in the Associated Laboratory for Energy, Transports, and Aeronautics. His research interests are in the area of Autonomous Robotic Vehicles with a focus on the fields of Mechatronic Systems Integration, Sensor Fusion, GPS and Positioning Systems, and Guidance, Navigation and Control Systems (GNC). He is author or coauthor of more than 85 journal papers (90% in first quartile), and 180 conference communications and participated in more than 40 European and Portuguese research projects, over the last 30 years.