Aerospace Science and Technology 117 (2021) 106899

Contents lists available at ScienceDirect

Aerospace Science and Technology

www.elsevier.com/locate/aescte

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

Pedro Outeiro *, Carlos Cardeira, Paulo Oliveira

IDMEC, Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal

ARTICLE INFO

Article history:
Received 1 November 2020
Received in revised form 12 May 2021
Accepted 11 June 2021
Available online 19 June 2021
Communicated by Christian Circi

Keywords:
Mobile robots
Estimation
Parameter estimation
Optimal control

ABSTRACT

This paper proposes an architecture based on a multiple-model solution for height and yaw control of a quadrotor transporting an unknown constant load, added before the flight. Estimates of the inertial parameters (mass and z-axis inertia) and state variables (vertical position, velocity, yaw angle and rate) are obtained using data from the onboard sensors. A Multiple-Model Adaptive Controller (MMAC) architecture is proposed. The control of each partial model is based on a steady state Linear Quadratic Regulator (LQR), using integrative action for the height control. The resulting system 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

Nowadays, the popularity of Unmanned Aerial Vehicles (UAV) has been rising. There is a growing number of research projects, and military/commercial applications. There is also a rising usage of drones at the consumer level for transportation and delivery [1], inspection [2], surveillance/monitoring [3], racing, photography, and filming. Examples of companies that are considering using drones for deliveries include UPS [4] and Amazon [5].

The main concern in transport and delivery applications, when designing the control system, is the nature of the load. Having to handle multiple different cargo, implies that there is high variation in the nature of the load. This presents a concern as it affects the performance of the control. Therefore, control and estimation solutions that are robust to uncertain parameters are highly desirable.

Control solutions for quadrotors can be found in [6–14], to mention a few. [6] presents a method based on saturated feedback and backstepping control. [7] presents an integral predictive/nonlinear robust control structure. In [8], the control in robust landing and lift-off is studied, proposing a sliding-mode approach. The solutions proposed in these works are robust 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 overconservative. In [9] an integral backstepping sliding mode control solution is proposed for handling external uncertain disturbances. In [10], a robust adaptive backstepping fast terminal sliding mode controller is presented for quadrotors with parametric uncertainties and external disturbances. In [11], a fixed-time integral-type sliding mode controller is presented for quadrotor attitude stabilization under actuator failures. Sliding mode control methods are appealing because of theirs robustness, but tend to produce much higher control activity than other controllers. A robust visual servoing switching-topology formation control solution for a team of quadrotors is presented in [12]. Using an external target provides important position information, but the resulting control becomes dependent on finding these references. Adaptive control is proposed in [13] for autonomous ship landing of multirotor UAVs. In [13], it is assumed that the quadrotor could communicate with the ship to obtain data, while this paper aims to use the onboard sensors. A flatness-based finite-time leader–follower formation controller is proposed in [14] for multiple quadrotors with external disturbances. In [14], the simplified model is obtained by assuming that part of its higher order dynamics are external disturbances with unknown upper bounds.

Load transportation with quadrotors has been considered in, for example, in [15–18], where the case of suspended loads was studied. A Model Predictive Control (MPC) approach is proposed and compared with Linear Quadratic Regulator (LQR) control in [15]. This paper presented results showing better performance using the MPC approach, but stability and convergence are only shown through testing, without any formal proof. A solution for trajectory generation and control based on the differentially flatness property of the system is presented in [16] with stability and con-

* Corresponding author.
E-mail address: pedro.outeiro@tecnico.ulisboa.pt (P. Outeiro).

https://doi.org/10.1016/j.ast.2021.106899
1270-9638/

vergence proofs. In [17], an adaptive solution for an unknown mass of the load is considered, relying on classical PID control. Adaptive fractional order sliding mode control is proposed in [18] for a quadrotor with a varying load.

A different approach for handling unknown parameters is Multiple-model methods. Multiple-model methods for piecewise constant unknown mass of quadrotors have been considered and tested in [19,20]. In [19], a preliminary study on a Multiple-Model Adaptive Estimator (MMAE) based solution using a bank of Integrative Kalman Filters and LQR control with integrative action is presented. By using Kalman Filters with an integrative mechanism for gravitational force, it reduced the state-estimation error. In [20], an alternative Multiple-Model Adaptive Controller (MMAC) was proposed, providing multiple-model framework that included the control component. Exploratory MMAE estimation has also been considered for the z-axis inertia in [21].

The Extended Kalman Filter (EKF) [22] is a widely used method for handling non-linear and parameter uncertainty. Other variations of the EKF, such as the Unscented Kalman Filter (UKF), have seen use in aggressive flight control for quadrotors in [23]. Although the EKF and UKF methods are widespread, there are no stability or robustness guarantees.

This work addresses the control and estimation problem for the height and yaw angle of the quadrotor. The presented solution is a Multiple-Model Adaptive Controller (MMAC) based architecture. The control is based on steady state Linear Quadratic Regulators (LQR), using integrative action for the height control. The estimation relies on Kalman filters, using an integrative component for the height. The onboard sensors used are the accelerometer, the ultrasound sonar, the gyroscope, and the magnetometer, commonly available onboard these types of vehicles. In addition to the onboard sensors, an indoor multiple-camera motion capture system is available.

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

This paper is organized as follows: the problem is presented in Section 2. The physical model used is shown in Section 3. The control architecture proposed is explained in Section 4. The solution for estimation is detailed in Section 5, and in Section 6 the solution for the control problem is presented. The analysis of stability and of null static-error for the full solution are shown in Section 7. In Section 8 the quadrotor model and its sensors are detailed. The implementation is 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

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

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

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 include the constant gravitational force, which imposes a non-linearity to the model. However, if there is *a priori* knowledge of the mass of the quadrotor, a compensation to the required thrust can be performed directly. The angular acceleration does not suffer from the same constant influence, providing a simpler problem. When a transported load has unknown mass (M_l) and inertia (I_l), a more complicated problem arises. Adjusting the equations to take this into account, provides

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

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

The driving interest of this work is that the solution for the control is not direct, worse performance is reached, and the platform stability is compromised. The physical problems include an unknown gravitational effect caused by M_l. Only a lower bound for this effect can be known *a priori*. Furthermore, the $(M + M_l)\ddot{p}_z$ and $(I_z + I_l)\ddot{\psi}$ components also add non-linearity to the problem. Using a single linearization of these equations, close to the hovering situation with nominal load, results in a solution that is best suited for small variations around a nominal mass and inertia. Therefore, standard linear solutions pose performance and stability concerns and different solutions should be explored.

An additional complication presented by the non-linearity of the dynamics is that there is a need for capable estimators. Furthermore, there is no vertical velocity measurement provided by the board sensor suite, and exploiting a ground camera with optical flow techniques provides poor estimates. To handle the optimal control problem and filter the sensory data, estimates of the state variables are necessary. Due to the non-linearity and the under performance achieved through linearization, a single Kalman filter is not an option.

3. Physical model

In this section, the dynamics of quadrotor for the height and yaw components are provided. This paper uses bold lowercase and bold uppercase symbols to represent vector and matrices, respectively.

The dynamical model of a quadrotor follows the equations

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

$$\boldsymbol{I}\dot{\boldsymbol{\Omega}} = -\boldsymbol{\Omega}\times\boldsymbol{I}\boldsymbol{\Omega} + \boldsymbol{\tau} \qquad \boldsymbol{\eta} = \begin{pmatrix}\phi\\\theta\\\psi\end{pmatrix} \qquad \boldsymbol{\tau} = \begin{pmatrix}\tau_x\\\tau_y\\\tau_z\end{pmatrix} \tag{6}$$

$$\boldsymbol{I} = \begin{bmatrix}I_x & 0 & 0\\0 & I_y & 0\\0 & 0 & I_z\end{bmatrix} \qquad \dot{\boldsymbol{R}} = \boldsymbol{R}\boldsymbol{\Omega}_\times \tag{7}$$

$$\boldsymbol{R} = \begin{bmatrix}c\psi c\theta - s\phi s\psi s\theta & -c\phi s\psi & c\psi s\theta + c\theta s\phi s\psi\\ c\theta s\psi + c\psi s\phi s\theta & c\phi c\psi & s\psi s\theta - c\psi c\theta s\phi\\ -c\phi s\theta & s\phi & c\phi c\theta\end{bmatrix} \tag{8}$$

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

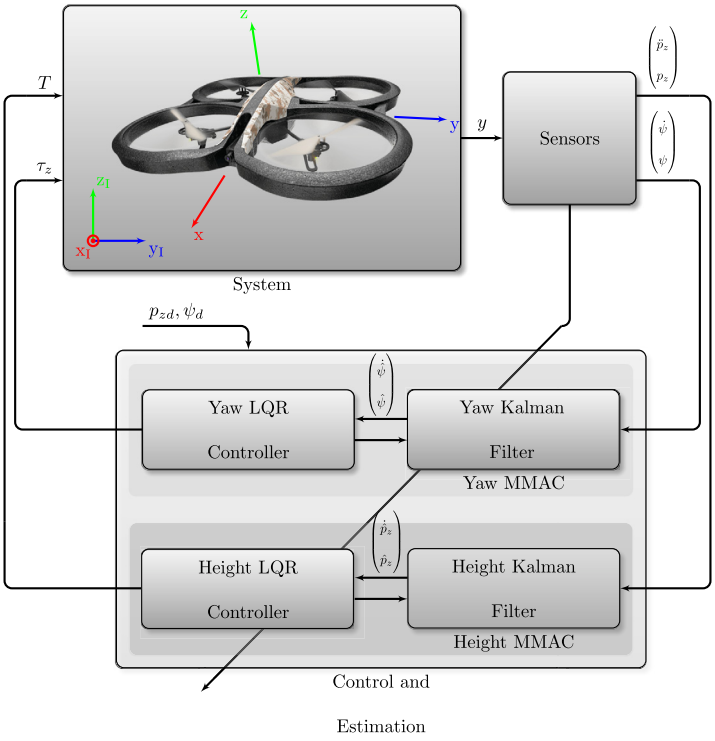


Fig. 1. Architecture with Multiple-Model Control.

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 from (5)-(7) and assuming zero roll and pitch angles, the dynamics of the height can be modeled as

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

where a_z stands for the thrust gain from the command input, which is assumed to be constant. Given the available measurements, the state-space form of the dynamics is

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

$$\underbrace{\begin{bmatrix}\ddot{p}_z\\ p_z\end{bmatrix}}_{\boldsymbol{y}_z} = \underbrace{\begin{bmatrix}0 & 0\\ 0 & 1\end{bmatrix}}_{\boldsymbol{C}_z}\begin{bmatrix}\dot{p}_z\\ p_z\end{bmatrix} + \underbrace{\begin{bmatrix}\frac{a_z}{M}\\ 0\end{bmatrix}}_{\boldsymbol{D}_z} T - \begin{bmatrix}g\\ 0\end{bmatrix} \tag{9}$$

3.2. Yaw dynamics

For the yaw dynamics the resulting behavior is described by

$$I_z\ddot{\psi} = a_\psi \tau_z + \left(I_x - I_y\right)\dot{\theta}\dot{\phi} \tag{10}$$

where a_ψ stands for the z-axis moment gain from the command input. Based on the available sensors and discarding the Coriolis effect, the state-space representation of the dynamics can be written as

$$\underbrace{\begin{bmatrix}\ddot{\psi}\\ \dot{\psi}\end{bmatrix}}_{\dot{\boldsymbol{x}}_\psi} = \underbrace{\begin{bmatrix}0 & 0\\ 1 & 0\end{bmatrix}}_{\boldsymbol{A}_\psi}\underbrace{\begin{bmatrix}\dot{\psi}\\ \psi\end{bmatrix}}_{\boldsymbol{x}_\psi} + \underbrace{\begin{bmatrix}\frac{a_\psi}{I_z}\\ 0\end{bmatrix}}_{\boldsymbol{B}_\psi}\tau_z$$

$$\underbrace{\begin{bmatrix}\dot{\psi}\\ \psi\end{bmatrix}}_{\boldsymbol{y}_\psi} = \underbrace{\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}}_{\boldsymbol{C}_\psi}\boldsymbol{x}_\psi \tag{11}$$

3.3. Inertia behavior analysis

A particular concern pertaining to the yaw rotation is that the overall inertia of an object depends on the relative position of its parts. Using system identification to assess the inertia would require the absence of high frequency variations during the test. The rotation of the rotors causes high speed variations. However, the relative center of mass of the rotors remains constant. Assuming the rotor blades to be approximately flat in the z-plane, by the Parallel Axis Theorem the z-axis inertia is not influenced by the rotation.

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

4. Control architecture

To achieve a stable solution with zero steady state error for the unknown load transportation problem, the control system architecture in Fig. 1 is proposed. As its foundation, LQR control and linear Kalman filtering are used. For the height loop, integrative components are added to the control and estimation to compensate the unknown gravitational force.

The state variables are obtained through the use of estimators, which has the added benefit of filtering the sensor noise. The control adjusts the thrust and z-axis moment in order for the system

to achieve the reference height and yaw angle, based on the state estimates of the filters.

The proposed architecture (Fig. 1) uses a Multiple-Model Adaptive Control (MMAC) for simultaneous control and parameter estimation. The reference values are p_{zd} and ψ_d, representing the desired height and yaw, respectively. The state estimates are $\hat{\boldsymbol{x}}_z$ and $\hat{\boldsymbol{x}}_\psi$, representing the state estimate vectors for the height and yaw components, respectively. The full drone data is represented as $\boldsymbol{y}$.

5. Estimation

In this section the estimation components of the solution are discussed, these being the classical Kalman filters and the added integrative component.

5.1. Kalman filter

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

$$\begin{cases} \dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{F}\boldsymbol{v} \\ \boldsymbol{y} = \boldsymbol{C}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{u} + \boldsymbol{w} \end{cases} \tag{12}$$

where $\boldsymbol{v}$ represents the process noise and $\boldsymbol{w}$ stands for the sensor noise. These random noises are assumed to have zero mean and are represented by the covariance matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ for the process and sensor noise, respectively. The Kalman filter minimizes the covariance matrix of the error $\boldsymbol{e}$:

$$\begin{cases} \boldsymbol{e} = \boldsymbol{x} - \hat{\boldsymbol{x}} \\ \dot{\boldsymbol{e}} = (\boldsymbol{A} - \boldsymbol{L}\boldsymbol{C})\,\boldsymbol{e} + \boldsymbol{F}\boldsymbol{v} - \boldsymbol{L}\boldsymbol{w} \end{cases} \tag{13}$$

via the optimization of the filter gain $\boldsymbol{L}$, denominated as Kalman gain. The error covariance dynamics follows the Ricatti equation

$$\dot{\boldsymbol{P}} = \boldsymbol{A}\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A}' - \boldsymbol{P}\boldsymbol{C}'\boldsymbol{R}^{-1}\boldsymbol{C}\boldsymbol{P} + \boldsymbol{F}'\boldsymbol{Q}\,\boldsymbol{F} \tag{14}$$

where $\boldsymbol{P}$ stands for the covariance matrix of the error. The resulting Kalman gain calculation follows

$$\boldsymbol{L} = \boldsymbol{P}\boldsymbol{C}'\boldsymbol{R}^{-1} \tag{15}$$

In a steady state case, $\dot{\boldsymbol{P}}$ is zero and yields the well known algebraic Ricatti equation, see [24] for details.

5.2. Kalman filter with integrative component

Integrators play a fundamental role in control systems. The addition of an integrative component to a control loop is usually used when there are concerns regarding the robustness of the overall system to perturbations and the steady state error. Since the overall mass of the system is unknown, it is not possible to directly factor the correct gravitational force into the estimation. Even when using multiple models, if there is no model matching the real mass, there will always be an incorrect force assumed in the filters. To reduce the error in the assumed force of each filter, the height filters require a component capable of adjusting this force. The proposed mechanism is a feedback loop of the integrated residue of the height to actuation input (u). This loop is responsible for adjusting an added force (c) to compensate the error in the assumed force. This forces the filter to thoroughly follow the measured height, and increases the accuracy of the velocity estimate. With tuning in mind, a gain (L_I) can be added to the feedback loop. The resulting structure is shown in Fig. 2.

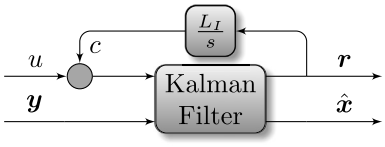


Fig. 2. Kalman Filter with Integrative Component.

With the addition of the integrative component the use of a single model approach could be considered, but the mass continues to affect the dynamics of the quadrotor even when not considering the gravitational force. This would impact the accuracy of the velocity estimate, where a multiple-model method can provide improvement for the performance.

Proving that the Integrative Kalman filter provides zero residue, for the integrated variable, requires the analysis of the transfer function of the sensor measurement to the estimate. Equation

$$\frac{\hat{p}_z}{p_z} = \frac{L_{22}Ms^2 + (L_{12}M - L_{21}L_I a_z)s + L_I(1 - L_{11})a_z}{Ms^3 + L_{22}Ms^2 + (L_{12}M - L_{21}L_I a_z)s + L_I(1 - L_{11})a_z} \tag{16}$$

is reached after tedious but straightforward computations. In this equation, the steady state gain has a value of one, indicating that the filter closely tracks the measurement of the height.

Furthermore, the steady state gain for the actuation input is required to be zero. Analyzing the transfer function

$$\frac{\hat{p}_z}{u} = \frac{a_z s}{Ms^3 + L_{22}Ms^2 + (L_{12}M - L_{21}L_I a_z)s + L_I(1 - L_{11})a_z} \tag{17}$$

it is observed that the steady state gain is zero.

6. Control

In this section, the control components are described. The two proposed controllers and the multiple-model architecture are presented: the Linear Quadratic Regulator (LQR), Multiple-Model Adaptive Control (MMAC), and the LQR with integrative action. Furthermore, the requirement of zero steady state error is studied.

6.1. LQR

The LQR is an optimal controller for linear systems. Its formulation is based on solving an optimal control problem for minimizing the cost function

$$J = \int_t^T \left[\boldsymbol{x}'\boldsymbol{Q}\,\boldsymbol{x} + \boldsymbol{u}'\boldsymbol{R}\boldsymbol{u}\right] d\tau \tag{18}$$

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

$$-\dot{\boldsymbol{M}} = \boldsymbol{M}\boldsymbol{A} + \boldsymbol{A}'\boldsymbol{M} - \boldsymbol{M}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}'\boldsymbol{M} + \boldsymbol{Q} \tag{19}$$

where the minimizing matrix is represented by $\boldsymbol{M}$. The control gain calculation follows

$$\boldsymbol{K} = \boldsymbol{R}^{-1}\boldsymbol{B}'\boldsymbol{M} \tag{20}$$

In the steady state case, $\dot{\boldsymbol{M}}$ is zero and the popular algebraic Ricatti equation results, see [25] for details. This method provides static gains.

An LQR controller is proposed for the control of the yaw angle. A necessary condition for the control is to ensure zero steady state

error, which an LQR controller achieves in this case (assuming the absence of drag effects on the dynamics). The z-axis moment is therefore calculated using

$$\tau_z = -k_{\dot{\psi}}\dot{\psi} + k_{\psi}(\psi_d - \psi) \tag{21}$$

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

$$\frac{\psi}{\psi_d} = \frac{a_{\psi}k_{\psi}}{I_z s^2 + a_{\psi}k_{\dot{\psi}}s + a_{\psi}k_{\psi}} \tag{22}$$

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

6.2. Multiple-model adaptive control

The MMAC algorithm (see [26] and the references therein) is a combined state-estimation, control, and parameter identification method. It is a method designed for handling parametric uncertainty and non-linear control (using different linearizations). As suggested by its name, it makes use of multiple models of the same system, based on different assumptions of the unknown parameter (or linearization points). For each of these models, a Kalman filter and LQR controller pair is designed, providing actuation better suited to its assumed model. The MMAC uses the data provided from each controller and merges it to provide better control than when using a single model. Furthermore, by assessing which of the models provides the best estimates, the value of the unknown parameter is estimated.

The Bayesian Posterior Probability Estimator (PPE) uses the residues of the known sensory data to determine the accuracy of each filter by assigning it a probability (posterior probabilities). These probabilities are calculated using

$$p_i = \text{Prob}(H = H_i \mid Y) \qquad i = 1, \ldots, n \tag{23}$$

which represents the probabilities of the unknown parameter H matching one of the models, given the set of past data Y. n represents the number of models and H_i stands for the value of the parameter for each model. A continuous approximation for this discrete problem, provides a calculation of the posterior probability derivatives $\dot{p}_i$. This derivative [27,28] can be computed using the probabilities p_i and the residues of the filters $\boldsymbol{r}_i$ using

$$\dot{p}_i = -\left(1 - \frac{\boldsymbol{\beta}_i e^{-\frac{1}{2}\boldsymbol{w}_i}}{\sum_{j=1}^{n} p_j \boldsymbol{\beta}_j e^{-\frac{1}{2}\boldsymbol{w}_j}}\right) p_i \tag{24}$$

$$\boldsymbol{\beta}_i = \frac{1}{(2\pi)^{\frac{h}{2}}\sqrt{det\,\boldsymbol{S}_i}} \tag{25}$$

$$\boldsymbol{w}_i = \boldsymbol{r}_i' \boldsymbol{S}_i^{-1} \boldsymbol{r}_i \tag{26}$$

where the number of sensors used is represented by h. The residual covariance matrix of each filter ($\boldsymbol{S}_i$) acts as a weighting parameter for the calculations, $\boldsymbol{\beta}_i$ represents the weighting parameter obtained from residual covariance and number of sensors, and $\boldsymbol{w}_i$ stands for a quadratic weighting parameter for the residue which also relies on the residual covariance.

The actuation provided by the MMAC algorithm can be obtained with a switching or weighted average based on the actuations of each controller, see [26] for details. In switching the actuation matches that of the controller with the highest posterior probability. In the weighted average the actuation of all controllers is averaged using the posterior probabilities as a weighting factor,

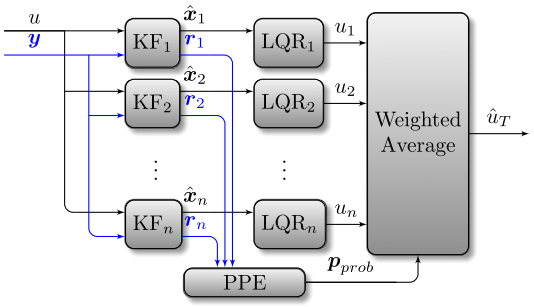


Fig. 3. MMAC Structure.

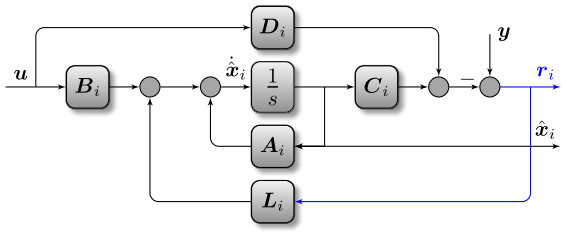


Fig. 4. MMAC Sub-components.

as shown in (27). In this work, it was adopted the use of the average weight, as it provides low pass filtered actuation, i.e.

$$\hat{u}_T = \sum_{i=1}^{n} p_i(t) u_i \tag{27}$$

The estimate of the parameter is also obtained using the weighted average of the model nominal parameters. The resulting structure is depicted in Figs. 3 and 4.

6.3. LQR with integrative action

In cases where the estimated mass does not match the real value, the error in assumed gravitational force generates steady state error. However, using integrative action for the LQR controller makes it capable of controlling a system in the presence of perturbations, like unmodelled dynamics (a suitable example for quadrotors is wind). The LQR controller with integrative action consists of a cascading controller with an outer layer that integrates the difference between reference and current value of the control variable and inner feedback of all the state variables, and is therefore proposed for the height control.

To obtain this controller using the LQR calculations already discussed, it is only necessary to modify the model of the dynamics when calculating the LQR gains. By using the modified version of the model

$$\boldsymbol{A}_I = \begin{bmatrix} \boldsymbol{A} & 0 \\ -\boldsymbol{C} & 0 \end{bmatrix} \quad \boldsymbol{B}_I = \begin{bmatrix} \boldsymbol{B} \\ 0 \end{bmatrix} \tag{28}$$

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

Using the modified model in the LQR gains computation yields $\boldsymbol{k}_{calc}$. The resulting gains include two different sets: $\boldsymbol{k}$ and k_I. $\boldsymbol{k}$ stands of the vector of gains for the state variables, and k_I represents the gain for the integrative component. $\boldsymbol{k}$ is composed of k_z and $k_{\dot{z}}$, which stand for the position gain and velocity gain, respectively.

$$\boldsymbol{k}_{calc} = [\boldsymbol{k} | -k_I] \quad \boldsymbol{k} = [k_{\dot{z}} \quad k_z]$$

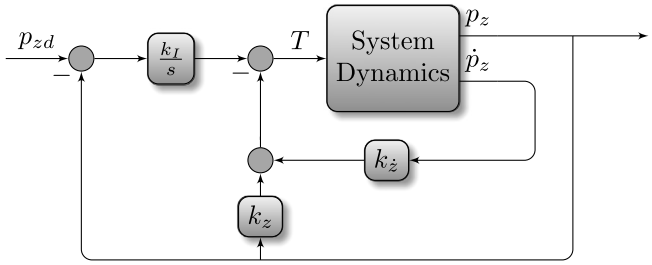


Fig. 5. LQR with Integrative Component.

This provides the following thrust calculation:

$$T = -k_{\dot{z}}\dot{p}_z - k_z p_z + \frac{k_I}{s}(p_{zd} - p_z)$$

which leads to the structure illustrated in Fig. 5.

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:

$$\frac{p_z}{p_{zd}} = \frac{k_I a_z}{Ms^3 + k_{\dot{z}} a_z s^2 + k_z a_z s + k_I a_z} \tag{29}$$

$$\frac{p_z}{g} = \frac{Ms}{Ms^3 + k_{\dot{z}} a_z s^2 + k_z a_z s + k_I a_z} \tag{30}$$

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

Stability and null static error are central to the desired control system architecture and are essential for the operation of autonomous aerial vehicles in the presence of parametric uncertainty. To analyze this issue, the following lemma from [26] 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 velocity and yaw rate errors converge to zero.* ■

The proof of the lemma is based on the assumptions previously outlined, namely assuming that one underlying model has the correct parameter. Thus, the eigenvalues of the controller and the estimator are recovered. By the separation principle, given a stable controller and a stable estimator, the resulting system is stable.

7.1. Height control

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

$$\boldsymbol{A}_z = \begin{bmatrix} 0 & 0 & -\frac{k_{\dot{z}} a_z}{M_r} & -\frac{k_z a_z}{M_r} & \frac{K_I a_z}{M_r} & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & L_{12} & A_{33} & A_{34} & A_{35} & A_{36} \\ 0 & L_{22} & A_{43} & A_{44} & A_{45} & A_{46} \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 \end{bmatrix} \tag{31}$$

$$A_{33} = -\frac{k_{\dot{z}} a_z}{M_m}(1 - L_{11}) \quad A_{34} = -L_{12} - \frac{k_z a_z}{M_m}(1 - L_{11})$$

$$A_{35} = \frac{k_I a_z}{M_m}(1 - L_{11}) \quad A_{36} = \frac{L_I a_z}{M_m}(1 - L_{11})$$

$$A_{43} = \frac{k_{\dot{z}} L_{21} a_z}{M_m} + 1 \quad A_{44} = \frac{k_z L_{21} a_z}{M_m} - L_{22}$$

$$A_{45} = -\frac{k_I L_{21} a_z}{M_m} \quad A_{46} = -\frac{L_I L_{21} a_z}{M_m}$$

where M_m represents the model mass, and M_r stands for the real mass. The input factor is represented by a_z, the L matrix and K vector represent the Kalman and controller gains, respectively. Additionally, the $\boldsymbol{B}_z$ matrix for the reference value is

$$\boldsymbol{B}_z = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}^T \tag{32}$$

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

$$\boldsymbol{C}_z = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{33}$$

The steady state is analyzed using

$$\boldsymbol{G}_z = -\boldsymbol{C}_z \boldsymbol{A}_z^{-1} \boldsymbol{B}_z \tag{34}$$

which yields a gain of one, as intended.

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

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

is used instead of the previous one. Re-using (34), 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

$$\boldsymbol{A}_\psi = \begin{bmatrix} 0 & 0 & -\frac{k_{\dot{\psi}} a_\psi}{I_r} & -\frac{k_\psi a_\psi}{I_r} \\ 1 & 0 & 0 & 0 \\ L_{11} & L_{12} & -L_{11} - \frac{k_{\dot{\psi}} a_\psi}{I_m} & -L_{12} - \frac{k_\psi a_\psi}{I_m} \\ L_{21} & L_{22} & 1 - L_{21} & -L_{22} \end{bmatrix} \tag{36}$$

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

$$\boldsymbol{B}_\psi = \begin{bmatrix} \frac{k_\psi a_\psi}{I_r} & 0 & \frac{k_\psi a_\psi}{I_m} & 0 \end{bmatrix}^T \tag{37}$$

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

$$\boldsymbol{C}_\psi = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix} \tag{38}$$

The steady state is analyzed using

$$\boldsymbol{G}_\psi = -\boldsymbol{C}_\psi \boldsymbol{A}_\psi^{-1} \boldsymbol{B}_\psi \tag{39}$$

which yields a gain of one, as intended.

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}

Table 2
Mass Model Parameters.

Model	1	2	3	4	5	6
M (kg)	0.431	0.454	0.478	0.505	0.535	0.567

Table 3
Inertia Model Parameters.

Model	1	2	3	4	5	6	7
$I_z \times 10^{-3}$ (kg m^2)	2.14	2.50	2.97	3.61	4.57	6.09	8.90

8. Implementation

Testing of the proposed solution was performed using the Parrot Ar.Drone 2.0 by Parrot SA. This off-the-shelf commercially available quadrotor is designed for users without any drone piloting skills. It comes equipped with an Inertial Measurement Unit (IMU) composed of a triad of accelerometers, a three-axis gyroscope and a three-axis magnetometer. Other sensors present are an ultrasound sensor, a barometer and two cameras. One front facing camera and a ground facing optical-flow camera. The rotors of the quadrotor are controlled through PWM commands. Therefore, the actuation inputs from the control are converted to the equivalent PWM commands for each rotor. The nominal parameters for this quadrotor are shown in Table 1.

Posterior probabilities p_i in the MMAC require an initialization value, the *a priori* probabilities. In the proposed solution, these probabilities are set equally for the n filters $(1/n)$. This is a common approach for cases where there is no *a priori* knowledge that points to a specific model or subset of models being more likely at start. The model selection was performed using the Baram Proximity Measure analysis, as it had been used in previous work with multiple-model methods for the adjustment of the model parameter and number of models. The obtained parameters for each of the models are detailed in Tables 2 and 3.

For the experimental test, the control system was implemented using the AR.Drone 2.0 Quadcopter Embedded Coder [29], as it provides a Simulink based environment for developing the software that will be deployed to the quadrotor and includes methods for accessing the sensors and actuators.

As the stabilization of the quadrotor is a necessary condition to perform the experimental testing, the X and Y position and angles control uses cascading PID control to adjust the torques. For these components, the sensory information was provided by the accelerometer, the ultrasound sonar, the gyroscope, and the magnetometer. Since the tests were performed indoors, the X and Y position measurements are provided by an indoor multiple-camera motion capture system. The experimental setup is presented in Fig. 6. The load is added before the flight.

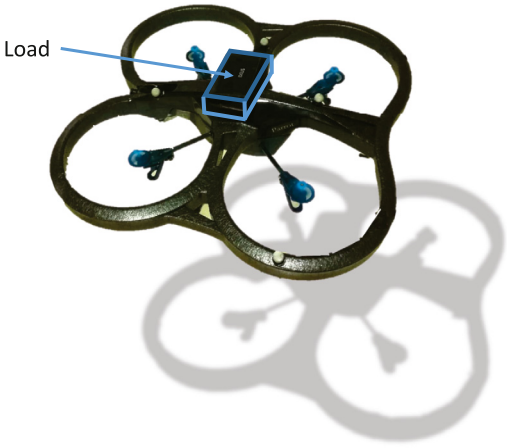


Fig. 6. Experimental Setup.

9. Stability verification

In this section, a simple verification of the system stability is provided. The design mass is 0.445 kg, and the real mass is 0.47 kg. The resulting eigenvalues for the height control are $\{-1.4648 \pm 3.0317i,\ -0.7023 \pm 0.4351i,\ -3.5159,\ -0.1077\}$. As intended, the real component of the eigenvalues are negative, providing an overall closed loop system that is stable. The design inertia is 5.1×10^{-3} kg m^2, and the real inertia is 4.8×10^{-3} kg m^2. The resulting eigenvalues for the yaw control are $\{-0.9873,\ -5.563 \pm 5.981i,\ -196.1\}$. As intended, the real component of the eigenvalues are negative, providing an overall closed loop system that is stable.

Furthermore, to assess the influence of the value of the model mass on the stability, a root locus like analysis is performed. Using

$$f_z = \boldsymbol{C}_z(s\boldsymbol{I} - \boldsymbol{A}_z)^{-1}\boldsymbol{B}_z \tag{40}$$

with (31), (32), and (33), this provides the transfer function of the system. Taking the resulting denominator $den(f_z)$, it can be re-written in the form of

$$den(f_z) = f_1 M_m + f_2 \tag{41}$$

Plotting the root locus for f_1/f_2 in Fig. 7, it is possible to perform the intended analysis. 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.

Performing the same analysis to assesses the influence of the value of the model inertia on the stability, results in

$$f_\psi = \boldsymbol{C}_\psi(s\boldsymbol{I} - \boldsymbol{A}_\psi)^{-1}\boldsymbol{B}_\psi \tag{42}$$

Using (36), (37), and (38), this provides the transfer function of the system. Taking the resulting denominator $den(f_\psi)$, it can be re-written in the form of

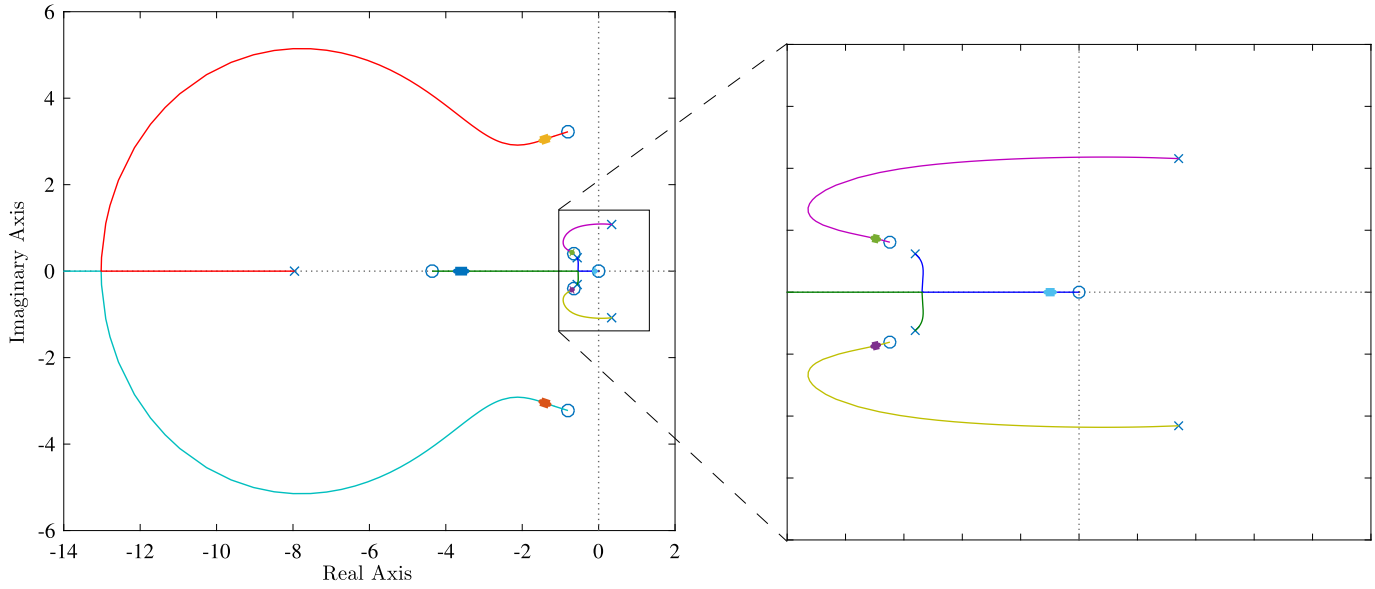


Fig. 7. Root Locus Mass.

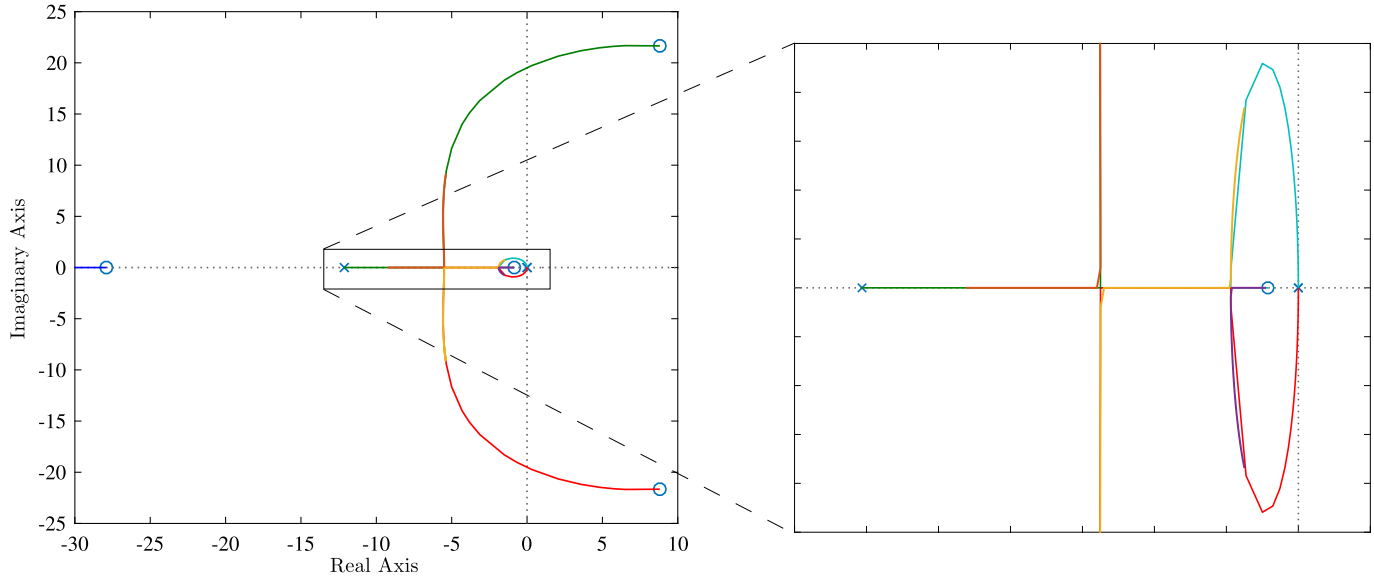


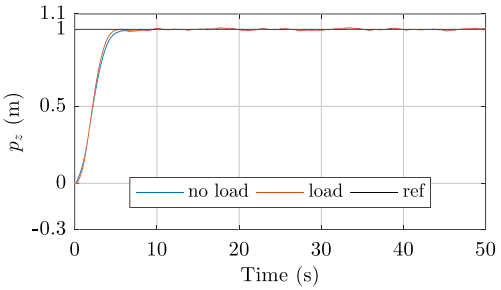
Fig. 8. Root Locus Inertia.

$$den(f_\psi) = f_1 I_m + f_2 \tag{43}$$

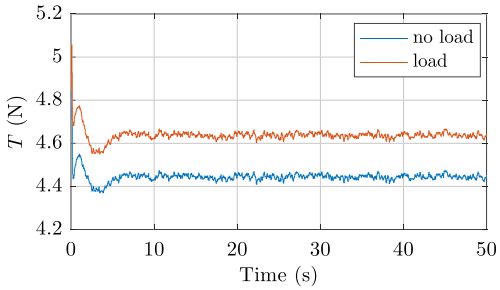
Plotting the root locus for f_1/f_2 in Fig. 8, it is possible to perform the intended analysis. There are two eigenvalues in the origin for zero inertia, and there are two eigenvalues with positive real part for inertia above 0.035 kg m^2. All of these inertia values are outside the range specified in 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

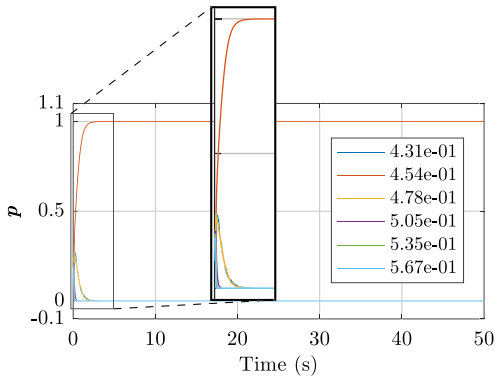
The first stage of testing was performed using simulation. The quadrotor parameters used for its setup are a mass M of 0.454 kg with just the drone and 0.473 kg carrying a load, an inertia I_z of 2.966×10^{-3} kg m^2 with just the drone and 4.833×10^{-3} carrying a load. The input factors a_z and a_ψ are kept at one. For the calculation of the Kalman gains, the covariance of the sensor noise is set as $R_{\ddot{z}} = 0.03$ for the z-axis acceleration, $R_z = 2.4 \times 10^{-7}$ for the height, $R_{\ddot{\psi}} = 0.03$ for the yaw rate, and as $R_\psi = 5 \times 10^{-7}$ for the yaw. Furthermore, the process noise covariances are inserted into the calculation as $Q_z = 0.005$ and $Q_\psi = 10^{-6}$, for the height and yaw respectively. The height LQR gains calculation uses $Q_{z/s} = 3.75$, $Q_z = 5$, $Q_{\dot{z}} = 0.001$, and $R_z = 15$. The yaw LQR gains calculation uses $Q_\psi = 1$, $Q_{\dot{\psi}} = 1$, and $R_\psi = 1$. The reference value for the height is one, while the yaw has a square wave of amplitude π as a reference. The simulation uses the full dynamics and not the simplified dynamics used for the proposed control and estimation design. The results are separated into two sections pertaining to the relevant control loop. A comparison with single-model approaches is also provided at the end of this section. The tests follow a weaker assumption that the model is within the considered range, instead of matching one of the considered models.



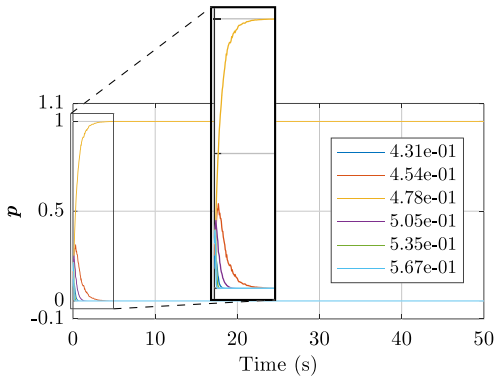
(a) Height for Both Tests



(b) Actuation for Both Tests



(c) Posterior Probabilities for the No Load Test



(d) Posterior Probabilities for the Load Test

Fig. 9. Height Simulation Results. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

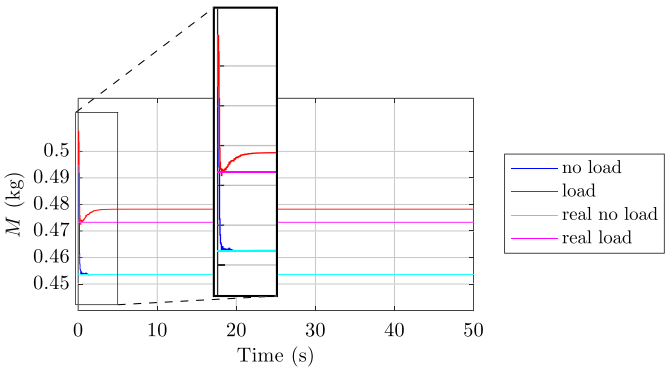


Fig. 10. Height Simulation Results: Mass Estimate for Both Tests.

10.1. Height control

The simulation results for the height are presented in Figs. 9 and 10. The settling times (5%) are at 4 seconds for both tests. There is no overshoot in both tests. The actuation does not saturate. The model selection ended after 2 seconds in the no load test and after 3 seconds in the load test. The selection of the closest model is observed in both cases. The mass estimate for the no load case converged to the correct value, because it is a matching case. In the load case, the inertia has an error, as it is a non-matching case.

10.2. Yaw control

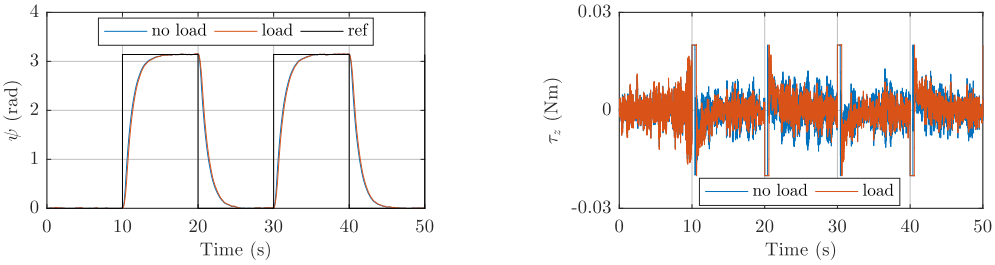
The simulation results for the yaw are presented in Figs. 11 and 12. The settling times (5%) are at 4 seconds in both tests. There is no overshoot in both tests. The actuation only saturated at the start of each rotation. The maximum probability is attributed to the closest model, but the two lowest inertia models are selected initially in both cases before the system had rotated. There is an error in the inertia estimate of the load test, as it is a non-matching case. The posterior probability of the closest model for the no load test gets close to 1 during the second rotation, while for the load test it achieves maximum probability during the first rotation.

10.3. Comparison between multiple-model and single-model approaches

A comparison between the multiple-model approach and single-model approaches is presented in Fig. 13. In this analysis, the overshoot/undershoot is analyzed at different real values for the mass and inertia. Since the height can suffer from overshoot and undershoot problems simultaneously, the metric used for the performance of the height controller is the overshoot minus the undershoot. Since the yaw is tested alternating between 0 and π radians, the analysis of the overshoot is enough. In the case of the height, the MMAC provides consistently low results, while the single-model tests show a quadratic behavior centered around the nominal model parameter. In the case of the yaw, the MMAC provides a more consistent result than the single-model tests. However, it is observed that in the Model 7 (highest model inertia) test there is an abrupt increase in overshoot at low inertias (below 2.5×10^{-3} kg m^2). Despite not obeying to the assumption of one of the models matching, the performance degrades gracefully. Increasing the number of models, or individually tuning the filters/controllers could be used to improve the MMAC performance.

11. Experimental results

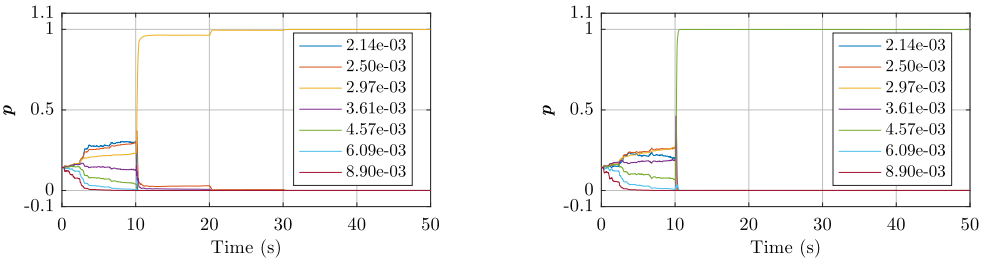
To truly test the performance of the proposed architecture, an experiment was prepared. The real mass M is 0.475 kg and the



(a) Yaw for Both Tests

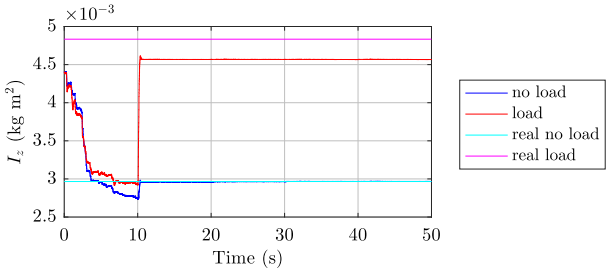
(b) Actuation for Both Tests

Fig. 11. Yaw Simulation Results.



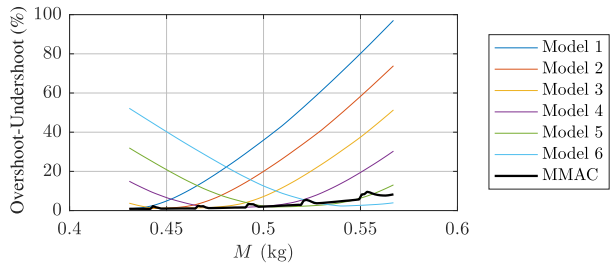
(a) Posterior Probabilities for the No Load Test

(b) Posterior Probabilities for the Load Test

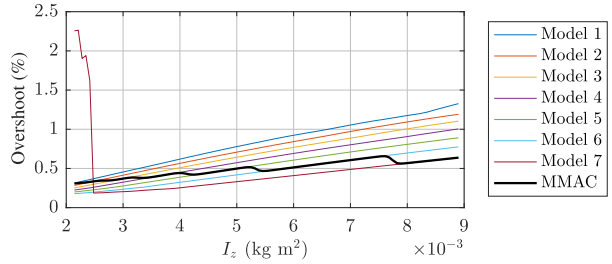


(c) Inertia Estimate for Both Tests

Fig. 12. Yaw Simulation Results (Cont.)

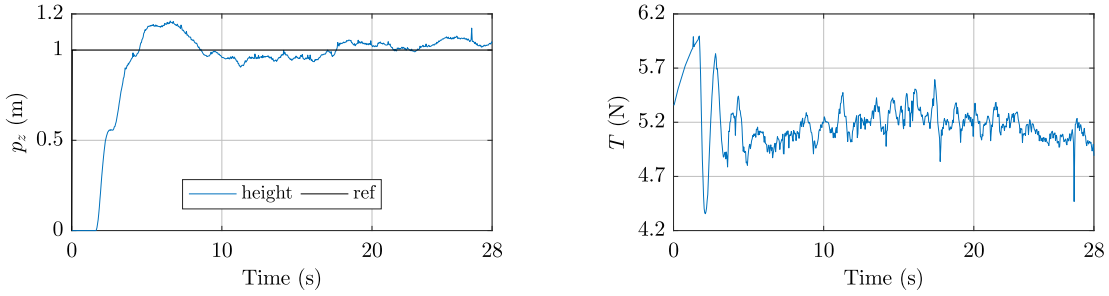


(a) Height



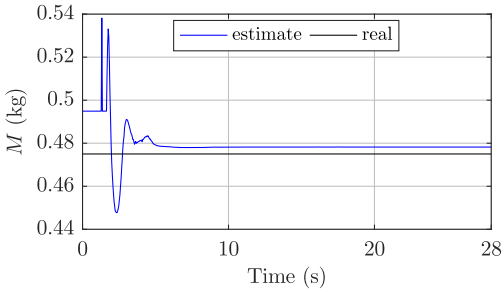
(b) Yaw

Fig. 13. Overshoot/Undershoot Comparison Between Multiple-Model and Single-Model Approaches.

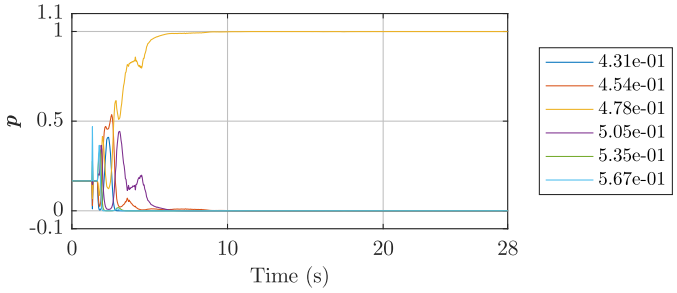


(a) Height (b) Actuation

Fig. 14. Height Experimental Results.



(a) Mass Estimate



(b) Posterior Probabilities

Fig. 15. Height Experimental Results (Cont.)

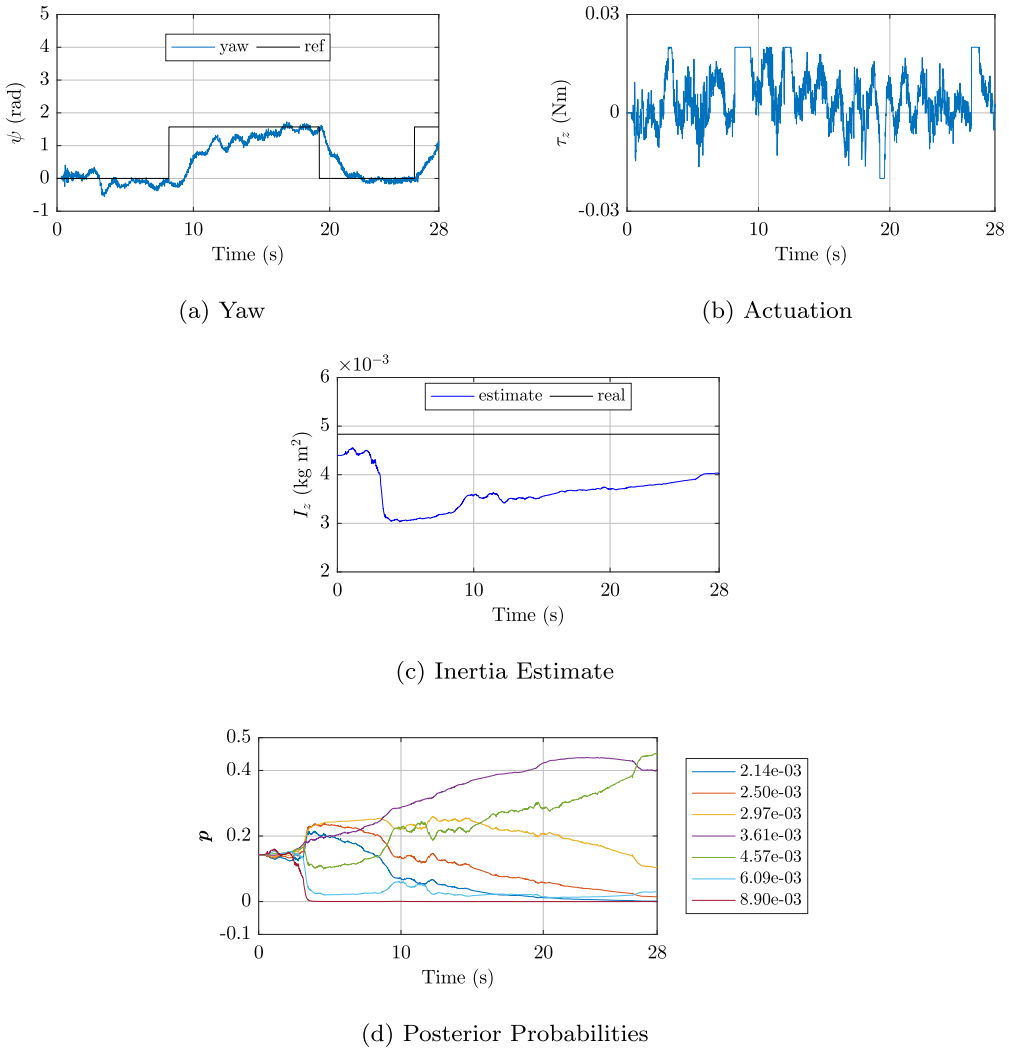
inertia I_z is 4.8334×10^{-3} kg m^2. The input factors a_z and a_ψ are set to 0.9 and 0.7, respectively. No test was performed with a load, as it was difficult to increase significantly the inertia without affecting the remainder of the dynamics. For the purpose of the Kalman gain calculations, the covariance of the sensor noise is defined as $R_{\ddot{z}} = 0.03$ for the z-axis acceleration, $R_z = 4.6 \times 10^{-7}$ 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 $Q_z = 0.05$, and the process noise of the yaw is given a covariance of $Q_\psi = 10^{-4}$. The height LQR gains calculations were performed using $Q_{z/s} = 10$, $Q_z = 1$, $Q_{\dot{z}} = 0.001$, and $R_z = 20$. The yaw LQR gains calculations were done using $Q_\psi = 1$, $Q_{\dot{\psi}} = 1$, and $R_\psi = 300$. The test consisted of lifting-off to a height of 1 meter, followed by a sequence of three 90° rotations.

11.1. Height control

The results of the test are presented in Figs. 14 and 15. The settling time (5%) achieved is 8 seconds with a 17% overshoot. The highest probability is given to the closest model. The probabilities vary during the start of the lift-off, but the selection of the closest model appears to not degrade from this. The model selection in this test takes longer than in the simulation, settling in 6 seconds. Higher overshoot and slower convergence occur due to several external disturbances. 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. 16. The settling times (5%) observed are 8 seconds in the first rotation and 4 seconds in the following rotations. There is no static error in the first two rotations. The actuation only saturates at the beginning of each rotation. The probabilities display a similar behavior to what was seen in simulation, evolving more visibly during the rotations, but comparatively slower. Slower convergence occurs due to several external disturbances and noise. The highest probability is given to the closest model after 26 seconds. The inertia estimate



(a) Yaw

(b) Actuation

(c) Inertia Estimate

(d) Posterior Probabilities

Fig. 16. Yaw Experimental Results.

has an error of 8×10^{-4} by the end of the test, but is seen converging to the real value.

12. Conclusion

This paper analyses the use of Multiple-Model Kalman filtering and LQR control, in an integrated architecture, for transportation of an unknown load with quadrotors. The unknown parameters that were studied were the mass and z-axis inertia. The sensors utilized were an accelerometer, an ultrasound sonar, a gyroscope, and a magnetometer. Simulation and experimental tests were performed using the Ar.Drone 2.0. The control and estimation systems yielded low settling times and no overshoot in simulation. In the experiment, the settling times saw a slight increase and the height had some overshoot. Zero steady state error was observed in simulation and experimentally for both components. The parameter estimation from the MMAC selected the closest model in the filter bank for both components. Parametric estimation errors were only obtained in non-matching cases, and in the experimental results for the yaw. Despite using a control method designed for linear systems, the simulation using the complete system and the experimental test show that this did not invalidate the proposed solution. To further explore parametric uncertainty in quadrotors, the authors are considering the use of nonlinear adaptive control for future work.

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.

Acknowledgements

This work is financed by national funds through FCT – 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] A. Jimenez-Cano, J. Martin, G. Heredia, A. Ollero, R. Cano, Dynamics, control and planning for cooperative manipulation of payloads suspended by cables from multiple quadrotor robots, in: 2013 IEEE International Conference on Robotics and Automation (ICRA), 2013, pp. 4916–4921.

[2] P.J. Sanchez-Cuevas, G. Heredia, A. Ollero, Multirotor uas for bridge inspection by contact using the ceiling effect, in: 2017 International Conference on Unmanned Aircraft Systems (ICUAS), 2017, pp. 767–774.

[3] L. Merino, F. Caballero, J.R. Martínez-de Dios, I. Maza, A. Ollero, An unmanned aircraft system for automatic forest fire monitoring and measurement, J. Intell. Robot. Syst. 65 (1) (2012) 533–548, https://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., https://pressroom.ups.com/pressroom/ContentDetailsViewer.page?ConceptType=PressReleases%26id=1553546776652-986, 2019. (Accessed 16 April 2019) [Online].
[5] Amazon Prime Air Website, https://www.amazon.com/Amazon-Prime-Air/b?node=8037720011, 2019. (Accessed 16 April 2019) [Online].
[6] P. Casau, R.G. Sanfelice, R. Cunha, D. Cabecinhas, C. Silvestre, Robust global trajectory tracking for a class of underactuated vehicles, Automatica 58 (2015) 90–98, https://doi.org/10.1016/j.automatica.2015.05.011.
[7] G.V. Raffo, M.G. Ortega, F.R. Rubio, An integral predictive/nonlinear H∞ control structure for a quadrotor helicopter, Automatica 46 (1) (2010) 29–39, https://doi.org/10.1016/j.automatica.2009.10.018.
[8] D. Cabecinhas, R. Naldi, C. Silvestre, R. Cunha, L. Marconi, Robust landing and sliding maneuver hybrid controller for a quadrotor vehicle, IEEE Trans. Control Syst. Technol. 24 (2) (2016) 400–412, https://doi.org/10.1109/TCST.2015.2454445.
[9] Z. Jia, J. Yu, Y. Mei, Y. Chen, Y. Shen, X. Ai, Integral backstepping sliding mode control for quadrotor helicopter under external uncertain disturbances, Aerosp. Sci. Technol. 68 (2017) 299–307, https://doi.org/10.1016/j.ast.2017.05.022.
[10] M. Labbadi, M. Cherkaoui, Robust adaptive backstepping fast terminal sliding mode controller for uncertain quadrotor uav, Aerosp. Sci. Technol. 93 (2019) 105306, https://doi.org/10.1016/j.ast.2019.105306.
[11] W. Gong, B. Li, Y. Yang, H. Ban, B. Xiao, Fixed-time integral-type sliding mode control for the quadrotor uav attitude stabilization under actuator failures, Aerosp. Sci. Technol. 95 (2019) 105444, https://doi.org/10.1016/j.ast.2019.105444.
[12] H. Liu, Y. Lyu, W. Zhao, Robust visual servoing formation tracking control for quadrotor uav team, Aerosp. Sci. Technol. 106 (2020) 106061, https://doi.org/10.1016/j.ast.2020.106061.
[13] K. Xia, S. Lee, H. Son, Adaptive control for multi-rotor uavs autonomous ship landing with mission planning, Aerosp. Sci. Technol. 96 (2020) 105549, https://doi.org/10.1016/j.ast.2019.105549.
[14] X. Ai, J. Yu, Flatness-based finite-time leader–follower formation control of multiple quadrotors with external disturbances, Aerosp. Sci. Technol. 92 (2019) 20–33, https://doi.org/10.1016/j.ast.2019.05.060.
[15] S. Notter, A. Heckmann, A. Mcfadyen, L. Gonzalez, Modelling, simulation and flight test of a model predictive controlled multirotor with heavy slung load, IFAC-PapersOnLine 49 (17) (2016) 182–187, https://doi.org/10.1016/j.ifacol.2016.09.032.
[16] K. Sreenath, N. Michael, V. Kumar, 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, pp. 4888–4895.
[17] S. Dai, T. Lee, D.S. Bernstein, Adaptive control of a quadrotor uav transporting a cable-suspended load with unknown mass, in: 53rd IEEE Conference on Decision and Control, 2014, pp. 6149–6154.
[18] M. Vahdanipour, M. Khodabandeh, Adaptive fractional order sliding mode control for a quadrotor with a varying load, Aerosp. Sci. Technol. 86 (2019) 737–747, https://doi.org/10.1016/j.ast.2019.01.053.
[19] Pedro Outeiro, Carlos Cardeira, Paulo Oliveira, 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, pp. 389–394.
[20] Pedro Outeiro, Carlos Cardeira, Paulo Oliveira, 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, pp. 4192–4197.
[21] Pedro Outeiro, Carlos Cardeira, Paulo Oliveira, MMAE/LQR yaw control system of a quadrotor for constant unknown inertia, IFAC-PapersOnLine 52 (12) (2019) 170–175, https://doi.org/10.1016/j.ifacol.2019.11.193, 21st IFAC Symposium on Automatic Control in Aerospace ACA 2019.
[22] R.C. Leishman, J.C. Macdonald, R.W. Beard, T.W. McLain, Quadrotors and accelerometers: state estimation with an improved dynamic model, IEEE Control Syst. Mag. 34 (1) (2014) 28–41, https://doi.org/10.1109/MCS.2013.2287362.
[23] G. Loianno, C. Brunner, G. McGrath, V. Kumar, Estimation, control, and planning for aggressive flight with a small quadrotor with a single camera and imu, IEEE Robot. Autom. Lett. 2 (2) (2017) 404–411.
[24] A. Gelb, Applied Optimal Estimation, MIT Press, 1974.
[25] B. Friedland, Control System Design: An Introduction to State-Space Methods, Dover Publications, 2005.
[26] S. Fekri, Robust adaptive mimo control using multiple-model hypothesis testing and mixed-μ synthesis, Ph.D. thesis, Instituto Superior Técnico, 2002.
[27] C.B. Chang, M. Athans, State estimation for discrete systems with switching parameters, IEEE Trans. Aerosp. Electron. Syst. AES-14 (3) (1978) 418–425, https://doi.org/10.1109/TAES.1978.308603.
[28] V. Hassani, A. Pedro Aguiar, A.M. Pascoal, M. Athans, 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, pp. 7261–7266.
[29] D. Lee, Ar.drone 2.0 support from embedded coder, https://www.mathworks.com/hardware-support/ar-drone.html, June 2017.