2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Madrid, Spain, October 1-5, 2018

MMAC Height Control System of a Quadrotor for Constant Unknown Load Transportation

Pedro Outeiro, Carlos Cardeira, Paulo Oliveira
IDMEC, Instituto Superior Técnico, Universidade de Lisboa, Portugal
{pedro.outeiro, carlos.cardeira, paulo.j.oliveira}@tecnico.ulisboa.pt

Abstract— **This paper presents a methodology for height control of a quadrotor that transports a constant unknown load, given the estimates on both weight and state variables, based on measurements from motion sensors installed on-board. The proposed control and estimation framework is a Multi-Model Adaptive Controller using LQR with integrative action and Kalman filter with integrative component. The control system obtained is validated both in simulation and experimentally, resorting to an off-the-shelf commercially available quadrotor equipped with an IMU, an ultrasound height sensor, and a barometer, among other sensors.**

I. Introduction

Delivery drones have recently been identified as a new class of unmanned aerial vehicles (UAV) utilized for transportation of packages, food or other goods. An example of these aplications is the design of a multi-tiered warehouse for a drone delivery system, that has been patented by Amazon. Although not yet implemented, Amazon has shown great interest in pursuing this project. The versatility of these vehicles makes them ideal for urban environments and, therefore, for transportation of cargo. There is extensive work on control and estimation for quadrotors and applications for known load transportation can be found in both [1] and [2]. However, little work has been produced for cases where the load is unknown. The multi-model approach has been considered in [3], but its application was limited to the estimation. In this paper, the extension of the multi-model functionality to the control is considered.

A Multi-Model Adaptive Controller (MMAC) [4] is proposed, resorting to a bank of integrative Kalman filters and LQR with integrative action. The motion sensors used are an accelerometer and an ultrasound sensor, which could be replaced by the barometer for higher altitude flight, all usually available on-board these types of platforms.

This paper is organized as follows: the problem addressed in this work is described in Section II. The physical model considered is presented in Section III. The multi-model algorithm is presented in Section IV. The solution for estimation is presented in Section V, and in Section VI the solution for the control problem is proposed. The verification of stability for the full proposed solution is presented in Section VII. Simulation results are presented and discussed in Section VIII. In Section IX the quadrotor model and its sensors are presented, and the implementation details are discussed. The experimental results are presented and analysed in Section X. Finally, some concluding remarks are drawn.

II. Problem Statement

$$M\ddot{z} = f\left(u, \dot{z}, g\right) \tag{1}$$

The height dynamics of a quadrotor is shown in (1), where z is the height of the quadrotor, M is its mass, u is the thrust and g is the acceleration of gravity, assumed constant at the mission environment. Although it is a non-linear equation, control solutions with an LQR controller and a constant compensation of the gravitational component can be used. However, this is not as simple for the problem of unknown load mass (m) transportation. Re-writing the equation, results in

$$(M+m)\ddot{z} = f\left(u, \dot{z}, g, m\right) \tag{2}$$

In this case, the solution for the control is not as immediate, the performance degrades, and the platform stability is compromised. The gravitational effect influenced by m is, therefore, unknown. Only a lower bound for the gravity effect can be known *a priori*. Additionally, the $(M+m)\ddot{z}$ component presents an added non-linearity to the problem. Since linearisation of this equation would limit a solution to only work for a specific mass and possibly a small range of masses, the use of standard linear solutions is out of question and alternative solutions should be considered.

Given the non-linearity of the dynamics, the estimation problem is harder. Additionally, the available sensors of the quadrotor do not provide a measurement of the z velocity or has poor quality, if based on optical flow techniques. To tackle the optimal control problem, the velocity is needed and, in the absence of sensory data, an estimate is required. Due to the non-linearity and the unreliability of a linearisation, linear Kalman filters are not an option and the non-linear version, the Extended Kalman Filter (EKF), poses the possibility of divergence, which is undesirable. Given these limitations, an alternative solution using the acceleration and position is required. Additionally, while robust control is a possible control possibility, it generates over conservative solutions. Because of this limitation, alternative methods are considered for this problem.

III. Physical Model

Consider the dynamics of a quadrotor as shown in [5]. Assuming zero roll and pitch angles and adding a linear drag of coefficient γ, the dynamics in the vertical direction can be described by:

$$(M+m)\ddot{z} = au - \gamma\dot{z} - (M+m)g \tag{3}$$

Where a is the thrust gain from the command input. These dynamics can be written in state-space format as:

$$\begin{aligned}\begin{bmatrix}\ddot{z}\\ \dot{z}\end{bmatrix} &= \begin{bmatrix}-\frac{\gamma}{M+m} & 0\\ 1 & 0\end{bmatrix}\begin{bmatrix}\dot{z}\\ z\end{bmatrix} + \begin{bmatrix}\frac{a}{M+m}\\ 0\end{bmatrix}u - \begin{bmatrix}g\\ 0\end{bmatrix}\\ z &= \begin{bmatrix}0 & 1\end{bmatrix}\begin{bmatrix}\dot{z}\\ z\end{bmatrix}\end{aligned} \tag{4}$$

That will be used in the remaining of the paper.

IV. Multi-Model Algorithm

The Multiple-Model Adaptive Control (MMAC) algorithm [4] is a combined state-estimation, control and system identification method. Its uses include providing a solution for parametric uncertainties and for non-linear control (using different linearisations). It performs the necessary estimation and control for different models of the system in study. The models are defined based on different parameter values (or linearisation points). For each model, a combined estimation and control system is designed, providing accurate actuation for the assumed model. The merging and processing of the information provided by the bank of estimators and controllers is achieved resorting to the computation of the Bayesian Posterior Probability Estimator (PPE) that selects the most accurate model.

The PPE assesses the accuracy through the residues of the known sensory data, by assigning a probability to each filter. The initial value of these probabilities are known as the *a priori* probabilities and are commonly initialized equal for the n filters ($1/n$), unless there is *a priori* knowledge to support higher or lower probability at start. The posterior probabilities $\boldsymbol{P}_{prob_k}(t+1)$, can be computed iteratively using the past probabilities $\boldsymbol{P}_{prob_k}(t)$ and residues of the filters $\boldsymbol{e}_i$ according to (5-7) [6], 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. $\boldsymbol{\beta}_i(t)$ is a weighting parameter based on the residual covariance and number of sensors, and $\boldsymbol{w}_i(t)$ is a quadratic weighting parameter for the residue which also uses the residual covariance.

$$\boldsymbol{P}_{prob_k}(t+1) = \frac{\boldsymbol{\beta}_k(t+1)e^{-\frac{1}{2}\boldsymbol{w}_k(t+1)}}{\sum_{j=1}^{n}\boldsymbol{\beta}_j(t+1)e^{-\frac{1}{2}\boldsymbol{w}_j(t+1)}}\boldsymbol{P}_{prob_k}(t) \tag{5}$$

$$\boldsymbol{\beta}_i(t+1) = \frac{1}{(2\pi)^{\frac{h}{2}}\sqrt{det\boldsymbol{S}_i(t+1)}} \tag{6}$$

$$\boldsymbol{w}_i(t+1) = \boldsymbol{e}'(t+1)\boldsymbol{S}_i^{-1}(t+1)\boldsymbol{e}(t+1) \tag{7}$$

Given the method for updating the probabilities, the PPE will cause the probability of the filter model with the least residue to tend to one. This means that it selects the most accurate model and uses exclusively its data for estimation. Additionally, an error in the estimation will always be incurred 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 estimation error when the system does not match any of the models, while also considering the added computational weight of using more models, see [4] for details on a method that could be used to tackle formally this problem.

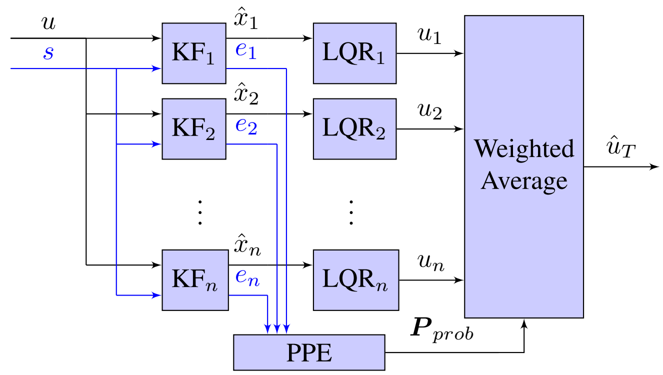

Fig. 1. MMAC Structure.

The total actuation of the MMAC algorithm can be obtained with a switching or weighted average, see [4] 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, as shown in (8). In this work, it was adopted the use of the average weight, as it provides low pass filtered actuation, i.e.

$$\hat{u}_T = \sum_{j=1}^{n}\boldsymbol{P}_{prob_k}(t)u_j \tag{8}$$

The resulting structure is depicted in Fig. 1.

V. Estimation

In this Section the estimation solution is discussed. First the integrative Kalman filter is presented as an approach to handle the unknown gravitational force, followed by the detailing of the filter design.

A. Kalman Filter with Integrative Component

$$\begin{bmatrix}\ddot{z}\\ \dot{z}\end{bmatrix} = \underbrace{\begin{bmatrix}-\frac{\gamma}{M+m} & 0\\ 1 & 0\end{bmatrix}}_{\boldsymbol{A}}\underbrace{\begin{bmatrix}\dot{z}\\ z\end{bmatrix}}_{\boldsymbol{x}} + \underbrace{\begin{bmatrix}\frac{a}{M+m}\\ 0\end{bmatrix}}_{\boldsymbol{B}}\underbrace{\left(u - \frac{(M+m)g}{a}\right)}_{U} \tag{9}$$

As discussed in Section IV, the MMAC algorithm does not ensure accurate state-estimation for cases where the estimated parameter does not match the real value. For the case of the transportation of an unknown load, the parametric error poses the added problem of the mass defining the gravitational force acting on the quadrotor. By using the gravitational forces as a bias of the actuation, as shown

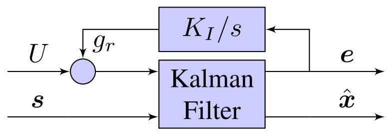

Fig. 2. Kalman Filter with Integrative Component.

in (9), it is possible to design linear filters that account for the gravitational force. However, the selective nature of the MMAC revealed that a more accurate estimation of the gravitational force was needed to minimize the effect of the uncertainty on the state-estimation. Therefore, an added mechanism was required to adjust the estimated force of the filters.

In control techniques, there is the option of using an integrative component. Although this is common in control, it is not usual for its use to be seen in estimation techniques. Since it was necessary for the Kalman filters to provide the residue ($\boldsymbol{e}$) to the MMAC algorithm, it could be used for the adjustment of the gravitational force (g_r). By creating a feedback loop to the actuation input (u) with an integrator, it would allow for the height estimate to follow the height measurement closely and provide a more accurate estimate of the velocity. For tuning purposes a gain can be given to the integration, allowing to adjust the overshoot and speed of the estimate. The resulting structure would resemble Fig. 2.

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 if disregarding the gravitational force, the mass still has weight on the dynamics of the quadrotor and the larger the difference in the assumed mass of the filter the higher the error of the velocity estimate. Therefore, the use of the MMAC algorithm is still beneficial.

B. Filter Design

The available sensors for height are the accelerometer and the ultrasound, providing the acceleration and height respectively, and were therefore used for the design of the filters. The model for the purpose of designing the filters is represented by the restructured state dynamics proposed in (9) and the state-space outputs shown in (10).

$$\begin{bmatrix}\ddot{z}\\ z\end{bmatrix} = \underbrace{\begin{bmatrix}-\frac{\gamma}{M+m} & 0\\ 0 & 1\end{bmatrix}}_{\boldsymbol{C}}\begin{bmatrix}\dot{z}\\ z\end{bmatrix} + \underbrace{\begin{bmatrix}\frac{a}{M+m}\\ 0\end{bmatrix}}_{\boldsymbol{D}} U \tag{10}$$

Using this model, the Kalman gains L are obtained and are combined with the integrative gain for the residue of the height to provide the filter presented in (11) and (12).

$$\begin{bmatrix}\ddot{\hat{z}}\\ \dot{\hat{z}}\\ \dot{g}_r\end{bmatrix} = \begin{bmatrix}\boldsymbol{A}-\boldsymbol{LC} & \boldsymbol{B}-\boldsymbol{LD}\\ \begin{bmatrix}0 & -\boldsymbol{K}_I\end{bmatrix} & 0\end{bmatrix}\begin{bmatrix}\dot{\hat{z}}\\ \hat{z}\\ g_r\end{bmatrix} + \begin{bmatrix}\boldsymbol{B}-\boldsymbol{LD} & \boldsymbol{L}\\ 0 & \begin{bmatrix}0 & \boldsymbol{K}_I\end{bmatrix}\end{bmatrix}\begin{bmatrix}U\\ \boldsymbol{s}\end{bmatrix} \tag{11}$$

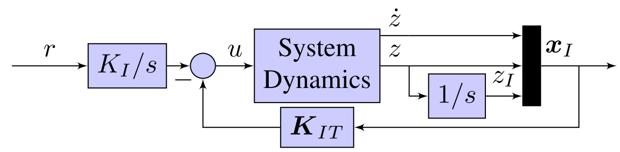

Fig. 3. LQR with Integrative Component.

$$\begin{bmatrix}e_{\ddot{z}}\\ e_z\end{bmatrix} = \begin{bmatrix}\begin{bmatrix}1 & 0\end{bmatrix}(-\boldsymbol{A}+\boldsymbol{LC}) & \begin{bmatrix}1 & 0\end{bmatrix}(-\boldsymbol{B}+\boldsymbol{LD})\\ \begin{bmatrix}0 & -1\end{bmatrix} & 0\end{bmatrix}\begin{bmatrix}\dot{\hat{z}}\\ \hat{z}\\ g_r\end{bmatrix} + \begin{bmatrix}\begin{bmatrix}1 & 0\end{bmatrix}(-\boldsymbol{B}+\boldsymbol{LD}) & \begin{bmatrix}1 & 0\end{bmatrix}(-\boldsymbol{L}+\boldsymbol{I})\\ 0 & \begin{bmatrix}0 & 1\end{bmatrix}\end{bmatrix}\begin{bmatrix}U\\ \boldsymbol{s}\end{bmatrix} \tag{12}$$

VI. CONTROL

In this section the LQR with integrative action is presented followed by the detailing of the controller design applied in the solution.

A. LQR with Integrative Action

As mentioned in Section II, the major impediment to using classical approaches to control is the unknown gravitational force. If this component of the dynamics is not correctly compensated, there will always be a static error. There is, however, a version of the LQR controller that is capable of controlling a system in the presence of perturbations, like unmodelled dynamics (a relevant example in quadrotors would be 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, which is equivalent to the structure shown in Fig. 3.

To obtain a controller like this 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 presented in (13), there would be a state-variable associated with the integration that would be used for defining the integrative control gain.

$$\boldsymbol{A}_I = \begin{bmatrix}\boldsymbol{A} & 0\\ -\boldsymbol{C} & 0\end{bmatrix} \quad \boldsymbol{B}_I = \begin{bmatrix}\boldsymbol{B}\\ 0\end{bmatrix} \tag{13}$$

Having a modified model, the next step is very straightforward, just calculate the LQR gains for the new model. Finally, from the resulting gains we obtain two different sets. $\boldsymbol{K}$ is the vector of gains for the state variables and K_I is the gain for the integrative component, selected according to $\boldsymbol{K}_{calc}$:

$$\boldsymbol{K}_{calc} = [\boldsymbol{K} | -K_I] \quad \boldsymbol{K}_{IT} = [\boldsymbol{K} \quad K_I] \quad \boldsymbol{K} = [K_1 \quad K_2]$$

One of the requirements for the control is ensuring zero static error. Thus, the analysis if the LQR controller with integrative action meets this requirement is performed here. For this, the system is separated into two components, one who has no gravity and receives a reference height and one that receives a zero reference height and has gravity. Additionally, the gravity will be considered as an input.

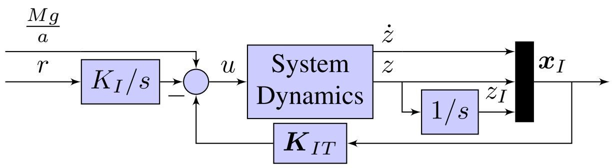

Fig. 4. LQR with Integrative Action and Quadrotor Mass compensation.

The transfer function for the reference height is as displayed in (14) and the transfer function for the gravity is presented in (15). 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.

$$\frac{K_I a}{s\left(s\left((M+m)\,s+K_1 a+\gamma\right)+K_2 a\right)+K_I a} \tag{14}$$

$$\frac{\gamma s}{s\left(s\left((M+m)\,s+K_1 a+\gamma\right)+K_2 a\right)+K_I a} \tag{15}$$

From this, it is concluded that the addition of the integrative action provided a zero static error solution.

B. Controller Design

Since this is a methodology for linear systems, a restructuring of the dynamics is required. Therefore, the restructuring used in Section V-A will also be used on the design of the controller. As mentioned in Section II, feedback linearisation was a possible solution for the known load case, but it can still be used with an unknown load. Using a feedback linearisation of the mass of the quadrotor would lessen the work required by the LQR controller with integrative action and would result in a faster control. The controller structure used is, therefore, presented in Fig. 4.

VII. Stability Verification

The stability of the proposed control system architecture is central to the operation of autonomous aerial vehicles in the presence of unknown constant parameters. To analyse this issue, the following lemma from [4] 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 such that, for $t > T$, all variables of the closed loop control system are bounded and the height error converges to zero.

The sketch 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 and the eigenvalues of the controller and the estimator are recovered. In the present case, the design mass was 0.47 kg, the Kalman filter used $K_I = 4$, and the real mass was 0.445 kg. The obtained eigenvalues were $eig = \{-0.7923, -0.0928, -3.5154 \pm 3.5320i, -0.7595 \pm 0.9019i\}$. As expected, all have negative real part and thus the overall closed loop system is stable.

VIII. Simulation Results

In this section, the preparation and results of the simulation are presented.

To validate if the proposed solution could work a simulation was prepared. The parameters of the drone used for its setup are a mass M of 0.42 kg, a drag coefficient γ of 0.1 and an input factor a of 1. The load used had a mass of 0.05 kg. Limitations related to the available thrust of the motor limited the range of load masses to a maximum of 0.15 kg and five masses 0.025 kg apart were used for the MMAC. The simulated load was given a mass of 0.025 kg. For the purpose of the Kalman gain calculations, the covariance of the sensor noise was defined as 1 and 0.001 for the acceleration and height respectively, while the process noise was given a covariance of 0.5. The integrative gain of the filters was set as 4. The PPE was set to hold until a height of 0.01 m had been reached, which also triggered the filters' reset. The LQR gains were calculated using $Q = diag\left([0.001 \quad 5 \quad 3.75]\right)$ and $R = 15$. A stop condition for the integrative component of the control was set for the thrusts outside a range of 3 to 5 N.

The results of the simulation are presented in Fig. 5 and 6. The settling time (5%) is at 4.5 seconds and the height stabilises around the 5 second mark for both tests without overshoot. The filter selection was made in under 0.5 seconds in both tests, providing accurate mass estimation in the no load case and having a slight error in the load case, as the mass of the no load matched one of the filters and the mass of the load case did not. The MMAC did not attribute full probability to the selected filter in the load case, having a small residue for the 0.495 kg filter and matching the larger mass of the test in relation to the nearest filter. The actuation did not saturate and was quick to adjust to fit the case in testing, as evidenced by the quick change in actuation for the no load case. These results were promising and provided similar results independently of whether the mass matched the assumed masses of the filters or not. Given these results, the approach passed onto experimental testing afterwards.

IX. Implementation

In this section the experimental set-up is presented.

The model that was used for testing is the Parrot Ar.Drone 2.0 by Parrot SA. It is an off-the-shelf commercially available, general purpose drone designed for users without any drone piloting skills. It is capable of recording video and is a good starter drone, due to its stability.

This model is equipped with an Inertial Measurement Unit (IMU) composed of a three-axis accelerometer, a three-axis gyroscope and a three-axis magnetometer. Other sensors are available on board such as an ultrasound sensor, a barometer and two cameras. One normal camera on the front and an optical-flow camera on the bottom.

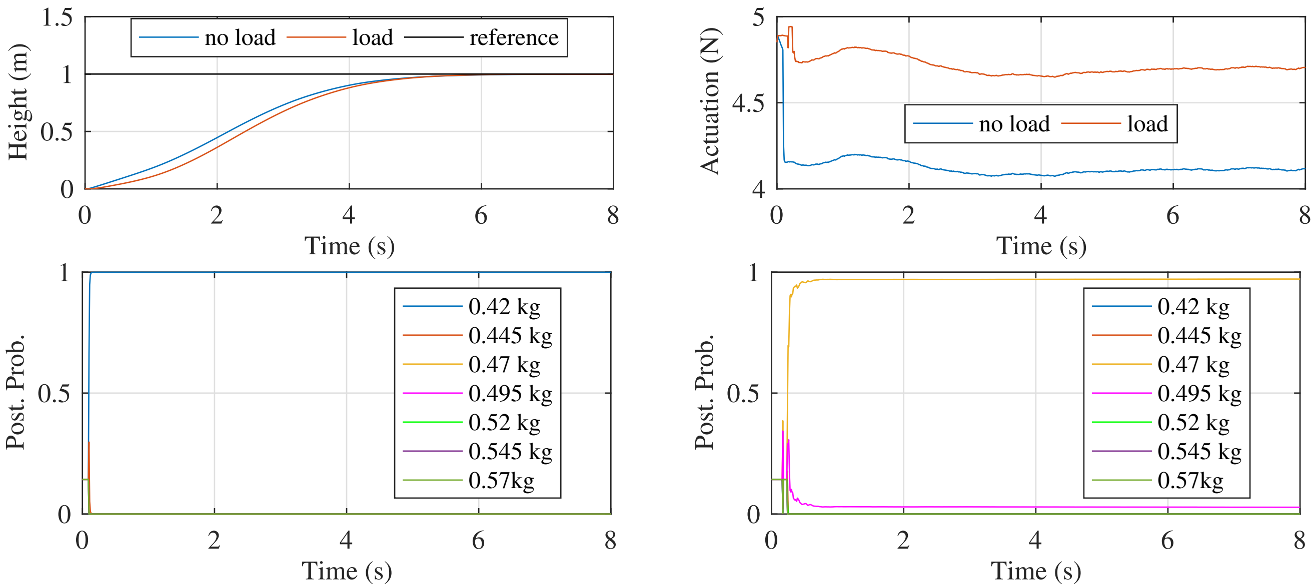

Fig. 5. Simulation results: upper left - (a) height for both tests, upper right - (b) actuation for both tests, bottom left - (c) posterior probability no load test, bottom right - (d) posterior probability load test.

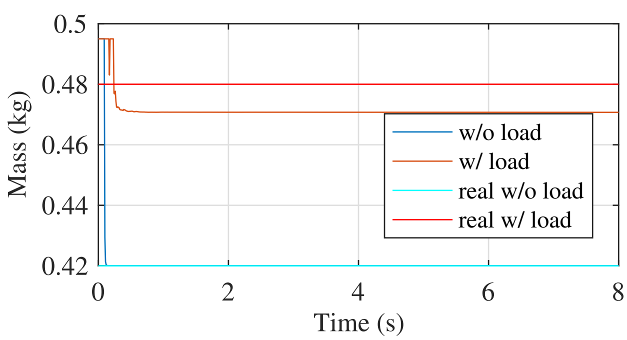

Fig. 6. Simulation results: mass estimate for both tests.

For the purpose of implementing the control system, the AR.Drone 2.0 Quadcopter Embedded Coder [7] was used, as it provided a Simulink based environment for development of software to run in the quadrotor and allows direct access to its sensors and actuators.

The controller and estimator from Section VIII were transferred onto the Embedded Coder Simulink environment. For the estimator, the thrust gain a had to be changed to 1.05 to ensure accurate mass estimation, given experimental calibration results. To assure the stabilisation of the quadrotor, the roll angle, pitch angle, and yaw velocity were regulated using an LQR control system for defining the necessary torques. The sensors used for angular control were the accelerometer and the gyroscope, for the angle and angular velocity respectively. Originally, the angle was filtered with a complementary filter [8] using angular data for low frequency and angle rate data for high frequency, but the unreliability of the angle measurement provided by the accelerometer led to switching the frequency bands for the sensors.

The commands for the rotors were defined by calculating the necessary thrust from each rotor from the thrust and torques and converting them into their equivalent PWM commands.

X. Experimental Results

In this section, the analysis of the results of the experiments is presented.

The experimental results are presented in Fig. 7 and 8. It is observed in Fig. 7a that the settling times (5%) are 4 and 6.5 seconds for the no load and with load cases respectively. The one meter height is reached at 4.5 and 6.5 seconds respectively. There was a 4% overshoot in the no load test. The actuation saturates only momentarily and stabilises with limited variation, as seen in Fig. 7b. In the initial stage of flight in Fig. 7c, the filter that was given more probability was the one with a mass of 0.495 kg, but the filter with the correct mass was selected in the end. In Fig. 7d the selection of the filter matching the closest mass was also observed and always had the highest probability. Additionally, the selection of the mass for with no load settled in about 2.5 seconds, while for the load case it settled in approximately 2 seconds. The settling time of the probabilities is further corroborated in Fig. 8, where it can also be observed the accurate estimation of the mass in the no load test and an error of 0.007 kg in the load test, due to it not matching the filters.

From the observations made regarding the data, it is inferred that the control and estimation provide good results. The settling time disregarding the lift-off time, which is associated with the compensation of the mass added to the drone and the unmodelled dynamics of the rotors (the response time to commands), is under 4 seconds. The estimation is smooth and provides a good estimate of the velocity.

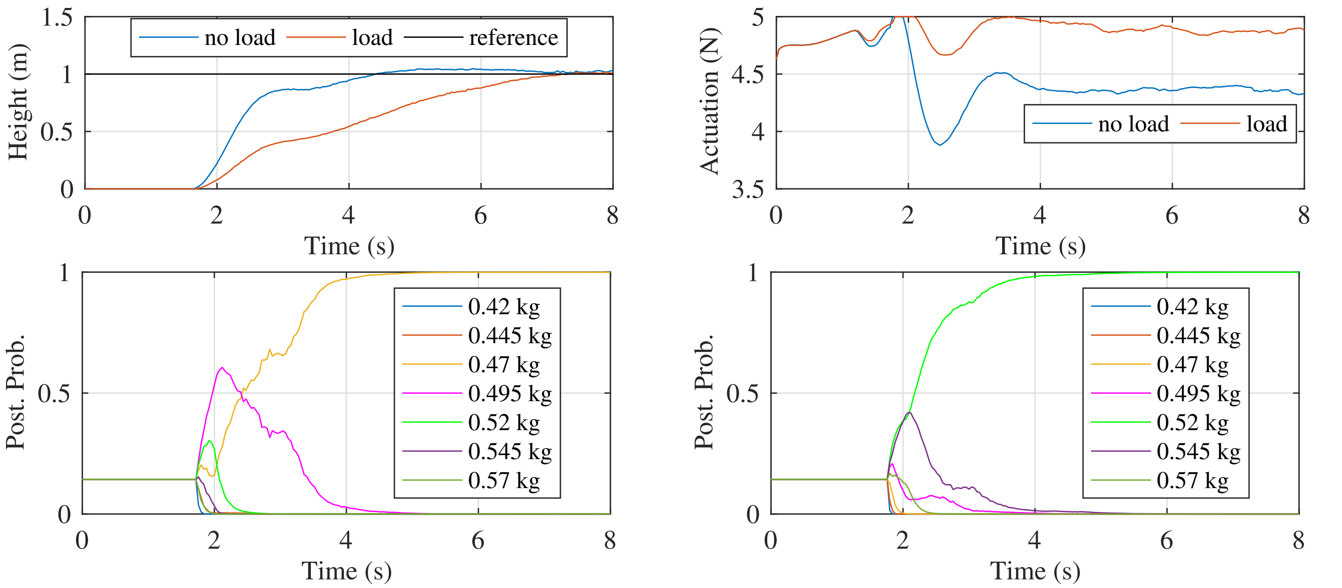

Fig. 7. Experimental results: upper left - (a) height for both tests, upper right - (b) actuation for both tests, bottom left - (c) posterior probability no load test, bottom right - (d) posterior probability load test.

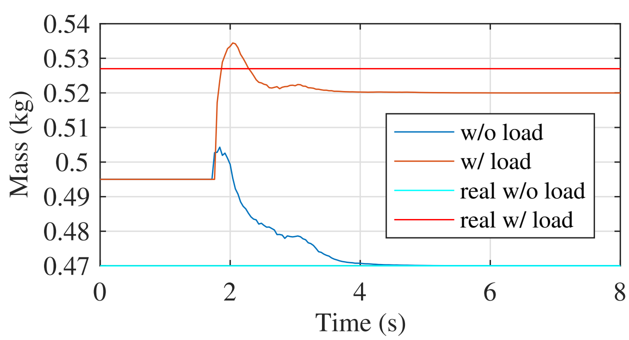

Fig. 8. Experimental results: mass estimates for both tests.

Conclusion

This paper proposed and studied the application of Multiple-Model Kalman filtering and LQR control for transportation of unknown loads with quadrotors using standard on-board sensors. The proposed controllers are LQR with integrative action combined with gravitational force compensation and the proposed filters are integrative Kalman filters. The sensors used for the proposed solution were an accelerometer and a ultrasound sensor. The solution was studied in a simulation and experimentally using the Ar.Drone 2.0. The control and estimation systems provided good results both in simulation and during real time in-flight testing. The mass estimation from the MMAC works for matching cases and is otherwise capable of selecting the closest mass in the filter bank. There was no major difference in the results of the control in simulation between testing with and without a load, while experimentally it was observed that the control was faster in the no load case.

Acknowledgments

This work was supported by FCT, through IDMEC, under LAETA, project UID/EMS/50022/2013 and project LOTUS of FCT - PTDC/EEI-AUT/5048/2014.

References

[1] S. Tang and V. Kumar, "Mixed integer quadratic program trajectory generation for a quadrotor with a cable-suspended payload," in *IEEE International Conference on Robotics and Automation*, 2015.

[2] K. Sreenath and V. Kumar, "Dynamics, control and planning for cooperative manipulation of payloads suspended by cables from multiple quadrotor robots," in *Robotics: Science and Systems (RSS)*, 2013.

[3] P. Outeiro, P. Oliveira, and C. Cardeira, "Lqr/mmae height control system of a quadrotor for constant unknown load transportation," accepted to Controlo 2018, 2018.

[4] S. Fekri, "Robust adaptive mimo control using multiple-model hypothesis testing and mixed-μ synthesis," Ph.D. dissertation, Instituto Superior Técnico, 2002.

[5] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: Modeling, estimation and control of quadrotor," *IEEE Robotics and Automation Magazine*, vol. 19, no. 3, pp. 20–32, September 2012.

[6] M. Athans, "Multiple-model adaptive estimation (mmae)," Slides, 2001.

[7] Darenlee, "Ar.drone 2.0 support from embedded coder," June 2017, https://www.mathworks.com/hardware-support/ar-drone.html.

[8] P. Oliveira, "Periodic and non-linear estimators with applications to the navigation of ocean vehicles," Ph.D. dissertation, Instituto Superior Técnico, 2002.