2018 13th APCA International Conference on Automatic Control and Soft Computing (CONTROLO)
June 4-6, 2018, Ponta Delgada, Azores, Portugal

LQR/MMAE Height Control System of a Quadrotor for Constant Unknown Load Transportation

Pedro Outeiro[2], Carlos Cardeira[1,2], Paulo Oliveira[1,2]
[1] IDMEC, [2] 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 controller is the sub-optimal steady state solution of an LQR problem, where an integrative effect is added to guarantee null steady state position error on average. The in flight load weight estimation problem is tackled resorting to a multiple-model adaptive estimator, providing also vertical velocity estimates. The control system obtained is validated both in simulation and 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

In recent years, there has been an increasing interest in the use of UAVs (Unmanned Aerial Vehicles) for transportation purposes (classified as delivery drones), with companies like Amazon testing product delivery systems using quadrotors. The high manoeuvrability of these vehicles and their hovering capabilities make them ideal for operation in environments with multiple obstacles and, therefore, for use in urban environments. For these same reasons, they prove to be of great interest for the transportation of cargo. Although applications using these vehicles for transportation are recent or still under development, the interest at a research level is still intense. In [1], [2] control methods were proposed for the transportation of tethered known loads, where in [1] a Mixed Integer Quadratic Program for aggressive manoeuvring is used and in [2] cooperative methods are proposed. However, applications where the load is unknown are far more uncommon. In this paper, methods to tackle the uncertainty are proposed both for the estimation and the control of the height of the quadrotor.

A Multi-Model Adaptive Estimator (MMAE) [3] using integrative Kalman filters is proposed and for control a Linear-Quadratic Regulator (LQR) with integrative action is considered. 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 physycal model considered is presented in Section III. The solution for estimation is presented in Section IV, and in Section V the solution for the control is proposed. The verification of stability for the full proposed solution is presented in Section VI. Simulation results are presented and discussed in Section VII. In Section VIII the quadrotor model and its sensors are presented, and the implementation details are discussed. The experimental results are presented ans analysed in Section IX. Finally, some concluding remarks are given in Section X.

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:

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

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.

III. Physical Model

Consider the dynamics of a quadrotor as shown in [4]. Assuming zero roll and pitch angles and adding a linear drag

978-1-5386-5346-6/18/$31.00 ©2018 IEEE

of coefficient γ, the dynamics in the vertical direction can be described by:

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

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

$$\begin{bmatrix}\ddot{z}\\ \dot{z}\end{bmatrix} = \begin{bmatrix}-\frac{\gamma}{M+m} & 0\\ 1 & 0\end{bmatrix}\begin{bmatrix}\dot{z}\\ z\end{bmatrix} + \begin{bmatrix}\frac{a}{M+m}\\ 0\end{bmatrix}u - \begin{bmatrix}g\\ 0\end{bmatrix}$$
$$z = \begin{bmatrix}0 & 1\end{bmatrix}\begin{bmatrix}\dot{z}\\ z\end{bmatrix} \quad (3)$$

That will be used in the remaining of the paper.

IV. Estimation

In this Section the estimation solution is discussed. First the multiple-model estimation framework is presented followed by the integrative Kalman filter, which is used as a replacement for the classical filters in the multi-model method. Finally, the details of the filters used are presented.

A. Multiple-Model Adaptive Estimation

The Multiple-Model Adaptive Estimation (MMAE) algorithm [3] is a combined state-estimation and system identification method. Its uses include providing a solution for parametric uncertainties and for non-linear state-estimation (using different linearisations). As the name implies, it relies on multiple models for the same system, but assuming different values of the parameter (or linearisation points). For each of these underlying models a Kalman filter is designed, providing accurate optimal estimates for its assumed model. The merging and processing of the information provided by the bank of filters is computed resorting to the computation of the Bayesian Posterior Probability Estimator (PPE) that selects the most accurate filter.

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 giving a higher or lower probability at start. Following values are called the posterior probabilities $\mathbf{P}_{prob_k}(t+1)$, calculated iteratively using the past probabilities ($\mathbf{P}_{prob_k}(t)$) and residues of the filters ($\mathbf{e}_i$) according to (4-6) [5] (h represents the number of sensors used). The residual covariance matrix of each filter ($\mathbf{S}_i$) is also 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 $\mathbf{w}_i(t)$ is a quadratic weighting parameter for the residue which also uses the residual covariance.

$$\mathbf{P}_{prob_k}(t+1) = \frac{\boldsymbol{\beta}_k(t+1)e^{-\frac{1}{2}\mathbf{w}_k(t+1)}}{\sum_{j=1}^{n}\boldsymbol{\beta}_j(t+1)e^{-\frac{1}{2}\mathbf{w}_j(t+1)}}\mathbf{P}_{prob_k}(t) \quad (4)$$

$$\boldsymbol{\beta}_i(t+1) = \frac{1}{(2\pi)^{\frac{h}{2}}\sqrt{det\mathbf{S}_i(t+1)}} \quad (5)$$

$$\mathbf{w}_i(t+1) = \mathbf{e}'(t+1)\mathbf{S}_i^{-1}(t+1)\mathbf{e}(t+1) \quad (6)$$

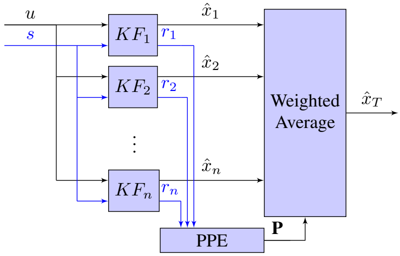


Fig. 1. MMAE Structure.

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 filters, see [3] for details on a method that could be used to tackle this problem.

The state estimation of the MMAE algorithm can be obtained in two fashions: switching or weighted average (see [3]). 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 (7). It was opted to use the average weight, as it provides low pass filtered state estimates.

$$\hat{\mathbf{x}}_T = \sum_{j=1}^{n}\mathbf{P}_k(t)\hat{\mathbf{x}}_j \quad (7)$$

The resulting structure is shown in Fig. 1.

To prevent computational errors in the real time implementation, a lower bound (LB) should be used, in which case (4) is replaced with (8).

$$\mathbf{P}_{prob_k}(t+1) = \frac{max(\boldsymbol{\beta}_k(t+1)e^{-\frac{1}{2}\mathbf{w}_k(t+1)}, LB)}{\sum_{j=1}^{n}max(\boldsymbol{\beta}_j(t+1)e^{-\frac{1}{2}\mathbf{w}_j(t+1)}, LB)}\mathbf{P}_{prob_k}(t) \quad (8)$$

A final computational concern is ensuring that, given a change in the parameter, the MMAE is always capable of changing the posterior probabilities. As the probabilities of the models that were not selected tend to zero, they will reach a value below the floating-point accuracy and will eventually round to zero and remain zero thereafter. To prevent this, a second lower bound is required for the probabilities. This results in a residual effect for all filters in the weighted average

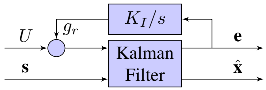


Fig. 2. Kalman Filter with Integrative Component.

used for the state-estimation. meaning that this lower bound would be preferably a low number. Tuning the probability lower bound requires taking into account two factors: higher values increase the residual effect of the filters and lower values increase the time it would take for the probability of a model to increase from this value.

B. Kalman Filter with Integrative Component

As discussed in Section IV-A, the MMAE algorithm can still result in an error in the state-estimation when the models assumed do not match the reality. In the case of the unknown load this becomes even more present, as the gravitational force acts as a constant unknown influence on the system. The design of the Kalman filters by treating the gravitational forces as a bias of the actuation, as shown in (9), allows for the use of a linear filter. However, the selective nature of the MMAE proved that this feedback would only be accurate if one of the models matched reality. An additional mechanism was necessary to account for the error of the assumed gravitational force in the filters.

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

In control techniques the use of integrative components is common, like the LQR with integrative action and the PID. However, in estimation this is not a common approach. Since the residue (e) was already a must for the MMAE, it was possible to use it to adjust the gravitic 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 this method solves 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 MMAE algorithm is still beneficial.

C. 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. Therefore, the model for the purpose of designing the filters is represented by the restructuring proposed in (9) for the state-variables and the state-space outputs are presented in (10).

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

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).

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

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

V. Control

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

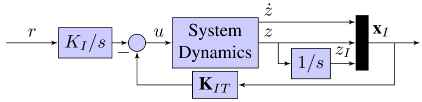


Fig. 3. LQR with Integrative Component.

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

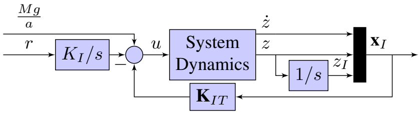


Fig. 4. LQR with Integrative Action and Quadrotor Mass compensation.

associated with the integration that would be used for defining the integrative control gain.

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

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. $\mathbf{K}$ is the vector of gains for the state variables and K_I is the gain for the integrative component, selected according to $\mathbf{K}_{calc}$:

$$\mathbf{K}_{calc} = [\mathbf{K}| - K_I] \quad \mathbf{K}_{IT} = [\mathbf{K} \quad K_I] \quad \mathbf{K} = [K_v \quad K_h]$$

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 treated as an input.

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 one, 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.

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

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

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 IV-B will also be used next 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.

VI. 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 [3] is instrumental:

Lemma 6.1: 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, without proof, due to the lack of space:

Lemma 6.2: 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.

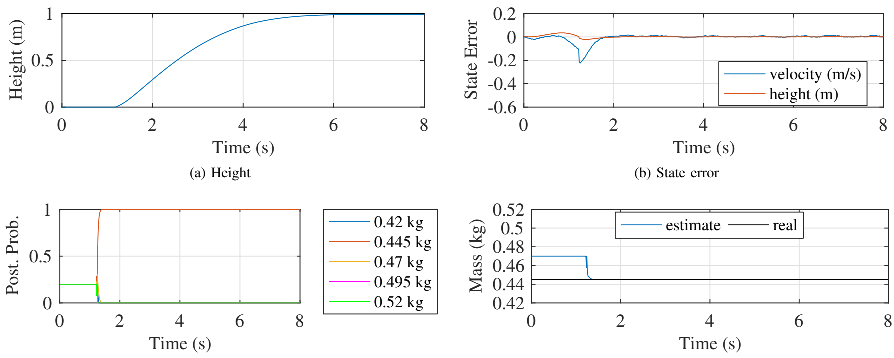
The sketch of the lemma is based on the assumptions previously outlined, namely assuming that one Kalman filter is based on an underlying model with the correct parameter and the eigenvalues of the controller and the estimator are recovered. In the present case, using a Kalman filter designed for a mass of 0.47 kg and with $K_I = 4$, an LQR controller assuming 0.42 kg and a real mass of 0.445 kg, we obtain the eigenvalues $eig = \{-0.747, -0.083, -3.554 \pm 3.565i, -0.703 \pm 0.989i\}$. As expected, all have negative real part and thus the overall closed loop system is stable.

VII. 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. Limitations related to the available thrust of the motor limited the range of load masses to a maximum of 0.1 kg and five masses 0.025 kg apart were used for the MMAE. 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 calculated LQR integrative gain was $K_I = 0.5$ and the state gains were $K = [0.8877 \quad 1.1494]$ for velocity and height respectively. 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, where it is observed that there is a 1 second lift-off period, followed immediately by the filter reset. The settling time (5%) is at 4.8 seconds and the height stabilises around the 6 second mark. There is no overshoot and the filters settle 0.5 seconds after resetting. At the same time that the state error settles, the 0.445 kg filter is selected, providing accurate mass estimation. However, the mass estimate in cases where the real mass



(a) Height (b) State error (c) Posterior probabilities (d) Mass estimate

Fig. 5. Simulation results.

does not match any of the filter masses can have an error. These results were promising and this approach passed onto experimental testing afterwards.

VIII. 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.

For the purpose of implementing the control system, the AR.Drone 2.0 Quadcopter Embedded Coder [6] 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 VII 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. 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 [7] 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.

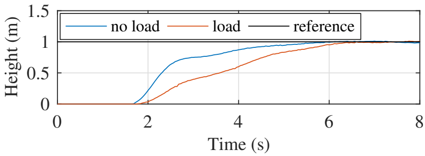
Additionally, the mass of the quadrotor was higher than expected having an added 0.05 kg without using a load.

IX. Experimental Results

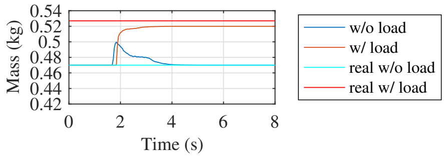
In this section, the analysis of the results of the experiments is presented.

The results obtained with the set-up in Section VIII are presented in Fig. 6 and 7. The total mass of the load used was 0.057 kg. It is observed in Fig. 6a that the settling times (5%) are 5 and 6 seconds for the no load and with load cases respectively. The one meter height is reached at 5.5 and 6.5 seconds respectively. The height estimate is smoother and follows the measurement very closely for both cases, as seen in Fig. 6c and 6d. The estimatd velocity is smooth despite the sensitivity of the accelerometer and seems coherent with the height data. In the initial stage of flight in Fig. 6e, the filter that was given more probability was the one with 0.495 kg, but the filter with the correct mass was selected in the end. In Fig. 6f the selection of the filter matching the closest mass was also observed and always had the highest probability. Additionally, the selection of the mass with no load settled in about 2.5 seconds, while for the load case it settled in approximately 1 seconds. The settling time of the probabilities is further corroborated in Fig. 6b, 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 and being outside of the range considered for the MMAE.

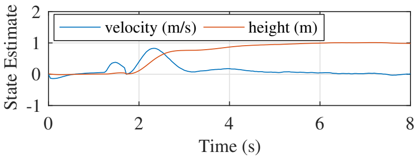
From the observations made regarding the data, it is inferred that the control and estimation provide good results. The settling time disregarding the lift-off time, which is associated with the compensation of the mass added to the drone and the unmodelled dynamics of the rotors (the response time to



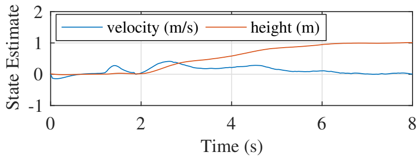
(a) Height measurement from both tests



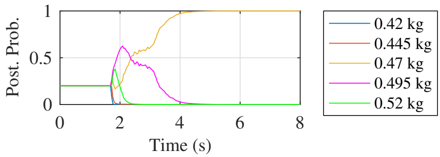
(b) Mass estimate from both tests



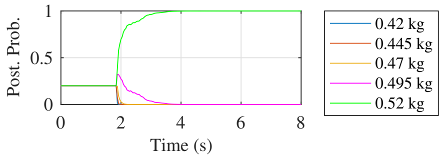
(c) State estimation for the no load test



(d) State estimation for the load test



(e) Posterior probabilities for the no load (0.47 kg) test



(f) Posterior probabilities for the load (0.527 kg) test

Fig. 6. Experimental results.

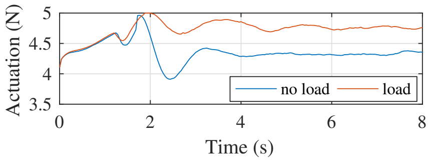


Fig. 7. Experimental values for the actuation of both tests

commands), is under 4 seconds. The estimation is smooth and provides a good estimate of the velocity.

X. Conclusion

This paper proposed and studied the application of LQR control and an extension to Multiple-Model Kalman filtering for transportation of unknown loads with quadrotors using standard on-board sensors. The proposed controller is an LQR with integrative action and the proposed estimator is an MMAE using 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 provided good results both is simulation and in testing. The response is fast in both components and the mass estimation works for matching cases and is otherwise capable of selecting the closest mass in the filter bank.

Additional work can be pursued by considering the use of a Multiple-Model Adaptive Controller (MMAC) [3] algorithm to integrate both estimation and control into one module, which would extend the multiple-model benefits to the control, which have been shown to integrate well with integrative methods.

Acknowledgments

This work was supported by FCT, through IDMEC, under LAETA, project UID/EMS/50022/2013.

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] S. Fekri, "Robust adaptive mimo control using multiple-model hypothesis testing and mixed-μ synthesis," Ph.D. dissertation, Instituto Superior Tcnico, 2002.
[4] 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.
[5] M. Athans, "Multiple-model adaptive estimation (mmae)," Slides, 2001.
[6] Darenlee, "Ar.drone 2.0 support from embedded coder," June 2017, https://www.mathworks.com/hardware-support/ar-drone.html.
[7] P. Oliveira, "Periodic and non-linear estimators with applications to the navigation of ocean vehicles," Ph.D. dissertation, Instituto Superior Técnico, 2002.