Model Reference Adaptive and Gain Scheduling Control for variable payload UAV Quadcopters

Jorge Diaz Luengo,* Joel Bordeneuve-Guibé, and Francois Defay
ISAE-SUPAERO, Université de Toulouse

ABSTRACT

This work addresses the flight control of a UAV subject to an important and sudden modification of its dynamics during the flight. Namely a UAV Quadcopter with variable payload during a mass drop mission is considered; at least 30% of the nominal weight of the drone is dropped. The control objective is to guarantee the same level of performance whatever the configuration of the UAV: loaded or unloaded. Two adaptive strategies are considered: Direct Model Reference Adaptive Control and Gain Scheduling Control. A feasibility study between both strategies is carried out, alongside a detailed comparison on the relative efficacy and ease of implementation of each. Finally, both controllers are integrated on ISAE-SUPAERO simulation facilities and tested in real flight. The superiority of Model Reference Adaptive Control is proven, not only in terms of dynamic behaviour, but also for its simplicity and robustness. The properties of autotuning and adaptation towards an unknown and disturbed flight make it a valuable solution for the flight control of UAVs.

1 INTRODUCTION

Quadcopters applications are growing since many years and the variety of application cases is increasing. In particular load transportation and mass dropping represent new challenging applications and can be used in many fields as rescue [1], delivery, medical assistance [2], agriculture, army... Thanks to the high payload and easily balanced center of mass, quadcopters are often used for transport missions [3]. Many others applications are focused on the use of multiple UAVs [4], a way of carrying heavy payloads with several small UAVs, mainly because of the actual regulation about maximum weight of UAVs.
Variable payload dropping has been addressed in [5] and remains a field with various interesting control problems.
Even if control and guidance of UAVs is an active and open domain of research, non standard configurations generate new needs in terms of performance keeping, recovery and safe flight guaranty. Although classical control methods are generally favoured, the need of more specific schemes able to "adapt themselves" becomes real.
This project is born from the interest of the ISAE Micro Drones Team to implement Adaptive Control algorithms on its competing UAVs, especially in those tasks involving a significant change in mass and/or payload of the UAV. The aim is to design an adaptive controller which is strong enough to properly cope with a drastic change of mass and inertia, while at the same time being flexible enough to be implemented in multiple UAVs.
Over the years, various controllers have been implemented for quadcopters as backstepping [6], Model predictive control [7], adaptive control [8] for different applications. In this paper it has been decided to compare Model Reference Adaptive and Gain Scheduling Control which are two possible candidates to solve the problem.
MRAC is a well known direct adaptive scheme, where the controller parameters are updated without any estimation of an open loop system model, like in indirect adaptive control. MRAC has proven to be very efficient and robust against model mismatches and uncertainties. Morevover it turns out that its implementation is very easy and adapted to demanding real time applications. Besides its adaptibility, the MRA controller can also be designed with a focus on autotuning[9].
On the other hand, gain scheduling control is very popular because basically the overall structure of an existing control scheme is preserved; only the controller parameters are scheduled as a function of an external variable. Many flight control systems rely on this solution. Obviously gain scheduling control belongs to the family of adaptive control.
This paper is organized as follows: in the first section two techniques of adaptive control are briefly presented: direct model reference adaptive control and gain scheduling control. Among their differences, the common factor between both techniques is the willingness of keeping the baseline controller structure already usually used for UAVs flight control. Then a detailed modelization of UAV dynamics is presented together with the global simulation and real-time control environment. The adopted methodology is also detailed and the flight results are shown and analyzed.

2 ADAPTIVE CONTROL

2.1 Direct Model Reference Adaptive Control

An adaptive system can be thought as having two loops (Figure 1): The first one being a classical feedback loop that includes the process and the control law, and the second one

*Email address(es): jorge.diaz-luengo@student.isae-supaero.com, joel.bordeneuve@isae-supaero.fr, francois.defay@isae-supaero.fr

being a parameter adjustment loop which makes it possible to compute the right parameters for the control law. Model Reference Adaptive Control is based on the parametrization of the controller as to obtain a plant whose behaviour mimics that of a reference system [10]. Therefore, the mechanism for adjusting the controller parameters is based on the comparison between the behaviour of the controlled system and of an explicit model reference, as seen in Figure 1 below.

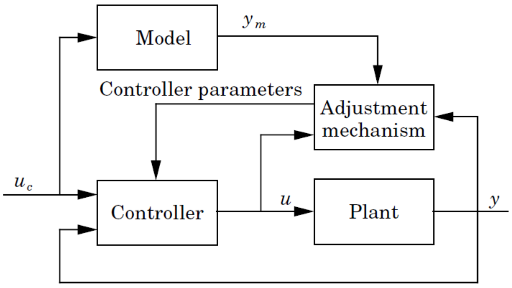

Figure 1: Block diagram of a MRA Controller

Considering the system output y and the reference model output y_m, the adaptive control law will explicitly be based on the output error $e = y - y_m$. For instance, the basic control law for a first order system can be written:

$$u(t) = \theta_1 u_c(t) + \theta_2 y(t) \tag{1}$$

where the controller gains θ_1 and θ_2 are being updated as:

$$\frac{d\theta_i}{dt} = -e\gamma\frac{\partial\theta_i}{\partial e} \qquad i = 1, 2 \tag{2}$$

where γ is an adaptation gain for the so called MIT Rule [11]. For a detailed proof of the convergence, especially Lyapunov Stability, please refer to [11]

2.2 Gain Scheduling

Gain Scheduling consists on designing a controller whose parameters change according to the operating conditions, but in a **pre-programmed manner**. This way, GS controllers provide a quick system reaction, whilst being reasonably straightforward to be implemented [11].

The general block diagram of a Gain Scheduling controlled process is seen in Figure 2 below:

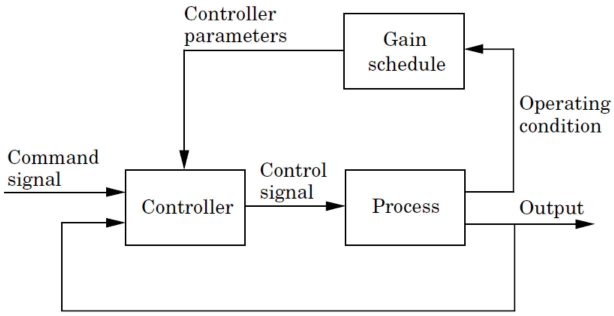

Figure 2: Block Diagram of a Gain Scheduling Controller

Basically, the design of such a controller is performed through two main steps [11]:

1. **Identification of the auxiliary variables**. These are the variables whose dynamics change, affecting the behaviour of the system and demanding flexibility from the controller. In our case, appropriate variables could be the **mass** and **inertia** of the drone, since they quantify the difference between the 2 possible states of the plant: loaded and unloaded.
2. **Gain Scheduling tables:** These tables gather the different controller tunings, each controller being "optimized" for one state of the system.

3 UAV MODELIZATION AND SIMULATION

The following modelization was based on [7], by using the general equations proposed by [12] as the general equations of Flight Dynamics. The UAV is modelled as an unalterable (rigid) body. Solving its dynamics means computing, for each time instant t the attitude and position of its body frame with respect to an inertial frame (Figure 3).

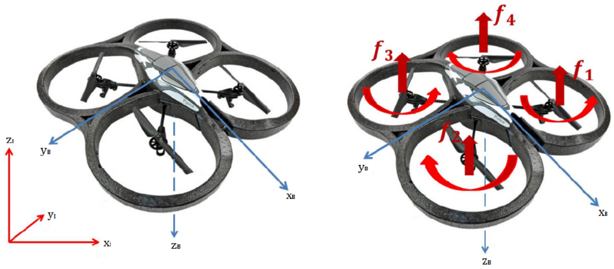

Figure 3: Inertial and Body frames

The aim is hence to compute the **attitude vector** $\vec{\eta}^B$ **and angular velocities vector** $\dot{\vec{\eta}}^B$ **expressed in the Body Frame** (^{B}), and the **position vector** $\vec{X}^I$ expressed in the Inertial Frame (^{I}).

$$\vec{X}^I = (x, y, z) \qquad \vec{\eta}^B = (\phi, \theta, \psi) \qquad \vec{\omega}^B = (p, q, r) \tag{3}$$

Now, applying the Rigid Body equations proposed by [12] and [7] and taking $\overline{\overline{I}}$ symmetric in all 3 axis, it comes:

$$\dot{p} = I_x^{-1}(M_\phi + qr(I_y - I_z)) \tag{4}$$

$$\dot{q} = I_y^{-1}(M_\theta + pr(I_z - I_x)) \tag{5}$$

$$\dot{r} = I_z^{-1}(M_\psi + pq(I_x - I_y)) \tag{6}$$

The corresponding equations for the position vector in the inertial frame are then:

$$\ddot{x} = Fm(sin(\phi)sin(\psi) + cos(\phi)sin(\theta)cos(\psi)) \tag{7}$$

$$\ddot{y} = (-sin(\phi)cos(\psi) + cos(\phi)sin(\theta)sin(\psi))Fm \tag{8}$$

$$\ddot{z} = Fm(-g + cos(\phi)cos(\theta) \tag{9}$$

The core structure of the Matlab/Simulink environment is depicted in Figure 4. This 8-block structure, suitable for integration with ROS/OROCOS, enables the environment to feature:

1. **Guidance and navigation indoors** (data given by an Optitrack system), and **outdoors** (GPS, Wifi or similar).
2. **Real time** simulations.
3. **Identical** structure for all drone models.
4. **Same environment** for simulations and flight tests (easy and quick switch between both).

Concerning the controller core design, the adopted strategy is driven by the following principles:

- Decentralized controllers for each axis (no cross coupling).
- Two cascade loops for each axis: **PI controller** for the position error, **P controller** for the velocity error. Derivative terms will not be considered due to Optitrack Noise effects. Hence 3 gains have to be tuned (K_P, K_I, K_{Pv}) per axis.
- **Attitude compensator:** the yaw angle will be kept null at all times.

4 Control Design Methodology

The mission defined to test and compare each controller covers a 120s flight composed of a first phase with a 400g extra mass (take-off, stabilization, vertical step changes), a second phase (horizontal circular motion) during which the mass is dropped, and a last phase with various vertical step changes and a final landing.

4.1 Model Reference Adaptive Controller

The 3 gains per axis are then adapted using the classical MRAC procedure: we first consider an initial rough tuning of the gains K_P, K_I, and K_{Pv}. Then the applied gains are computed as:

$$K_{Pa} = \theta_P K_P \qquad K_{Ia} = \theta_I K_I \qquad K_{Pva} = \theta_{Pv} K_{Pv} \tag{10}$$

where $\theta_P, \theta_I, \theta_{Pv}$ iare updated using (2).

As for the reference model, the dynamics on each axis have been modelled as a 2^{nd} order system, with the following requirements: **Natural frequency:** $4.5rad/s$, to enable a fast response and **overdamping** ($\zeta = 0.95$) to remove oscillations.

4.2 Gain Scheduling Controller

We have considered two models for the UAV: one with the extra load (before dropping it) and one without it. Thus two controllers have been designed independently in order to reach the aforementioned closed loop 2^{nd} order behaviour. Consequently the total mass of the UAV is the "switching" parameter between both controllers. Then it is critical to estimate in real time the mass of the drone, in order to identify the time at which the mass drop takes place, hence the appropriate switching time between both controllers.

The mass change is estimated from the following:

1. **Vertical velocity Vz**: this will be differentiated to compute the vertical acceleration.
2. **Throttle coefficient**: by using the motor model, the throttle enables calculation of the propeller speeds $w(rad/s)$, and hence, propulsion force $F(Newtons) = P_w * w^2$. The values for the thrust coefficient P_w are known.

The mass estimation is based on the following equation:

$$m(\ddot{z} + g) = F = 4P_w w^2 \tag{11}$$

where there are 2 approaches:

1. **Naive:** divide the force estimation by the $(\ddot{z}+g)$ signal. We obtain a **very reactive estimation**, but prone to error (e.g. the steps in Z are interpreted as mass changes).
2. **Recursive Least Squares:** with forgetting factor $\lambda = 0.98$. We obtain a **slower but more stable estimation**.

The actual estimation combines both ideas, in order to ensure both fast identification of the mass drop and error detection.

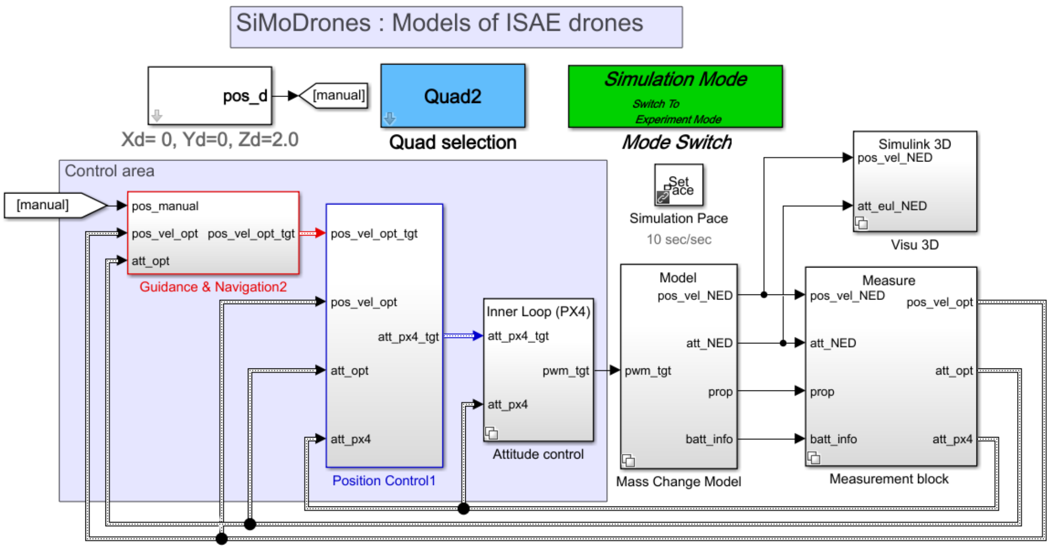

Figure 4: Full Control and Guidance drone model (MRAC)

5 FLIGHT TESTS RESULTS WITH MRAC AND GAIN SCHEDULING

The results are exposed respectively in Figures 5 and 6 for both MRAC and Gain Scheduling controllers. Apart from a good tracking in the trajectory and each individual axis, it is extremely desirable to have a damped throttle behaviour, especially to save battery life. Notice that mass drop takes place after about 60s. Also included are the evolution of the adaptation gains and the mass estimation during the trajectory.

- **Loaded performance:** It is observed that the MRAC suffers from a slower take-off. This is due to the fact that around 5s are needed for the parameters to converge, because the initial gains K_P, K_I, and K_{Pv} are clearly badly tuned. Once in the air, the performance is very satisfactory, with a maximum 6 cm overshoot during the step sequence, and almost negligible static error during the circle/ellipse trajectory.

  The GS controller however, offers a better performance in terms of reactivity (faster take-off), as it is not subjected to the convergence of the MRAC parameters. Nevertheless, this greater reactivity comes at the price of a 30 cm oscillation in XY, corrected only 5s after take-off.

- **Unloaded performance:** As expected, once the parameters have converged again after the mass drop, the MRAC performance is almost identical to the loaded state, and follows that of the reference system.

  On the other hand, the GS controller offers a less impressive performance, characterized by 6 cm overshoots. This is due to the high integral gain present in the unloaded state. Unfortunately, this high gain is essential to ensure a fast convergence after the mass drop. An optimum had to be found between a fast convergence after the drop and a proper unloaded performance, resulting in the displayed trajectory.

- **Drop performance:** Both controllers prevent the drone from gaining more than 30 cm of altitude after the drop. However, while the GS takes **9s** to bring the drone back to its original altitude, the MRAC only takes **6s**. This 3s advantage is a great feature of the MRAC.

- **MRAC parameters convergence:** As a rule of thumb, it takes **10s** for the parameters to converge, either at the beginning of the mission and after the drop. The use of more noisy trajectories or higher adaptation gain would decrease the convergence time, but it would also result in a greater stress on the engines.

- **Mass estimation:** This is the main drawback of the GS. The "naive" estimation proved to be extremely volatile, usually inducing false mass changes during the step trajectories. On the other hand, the RLS estimation, whilst better, resulted too slow.

As of today, the GS controller requires a precise knowledge of the drone mass and payload, only to deliver a performance that is only superior to that of the MRAC during the loaded segment of the mission. On the other hand, the MRAC offers a good performance throughout the entire mission, including better convergence after the drop, without any initial information about the drone mass or payload. Moreover, tests at very

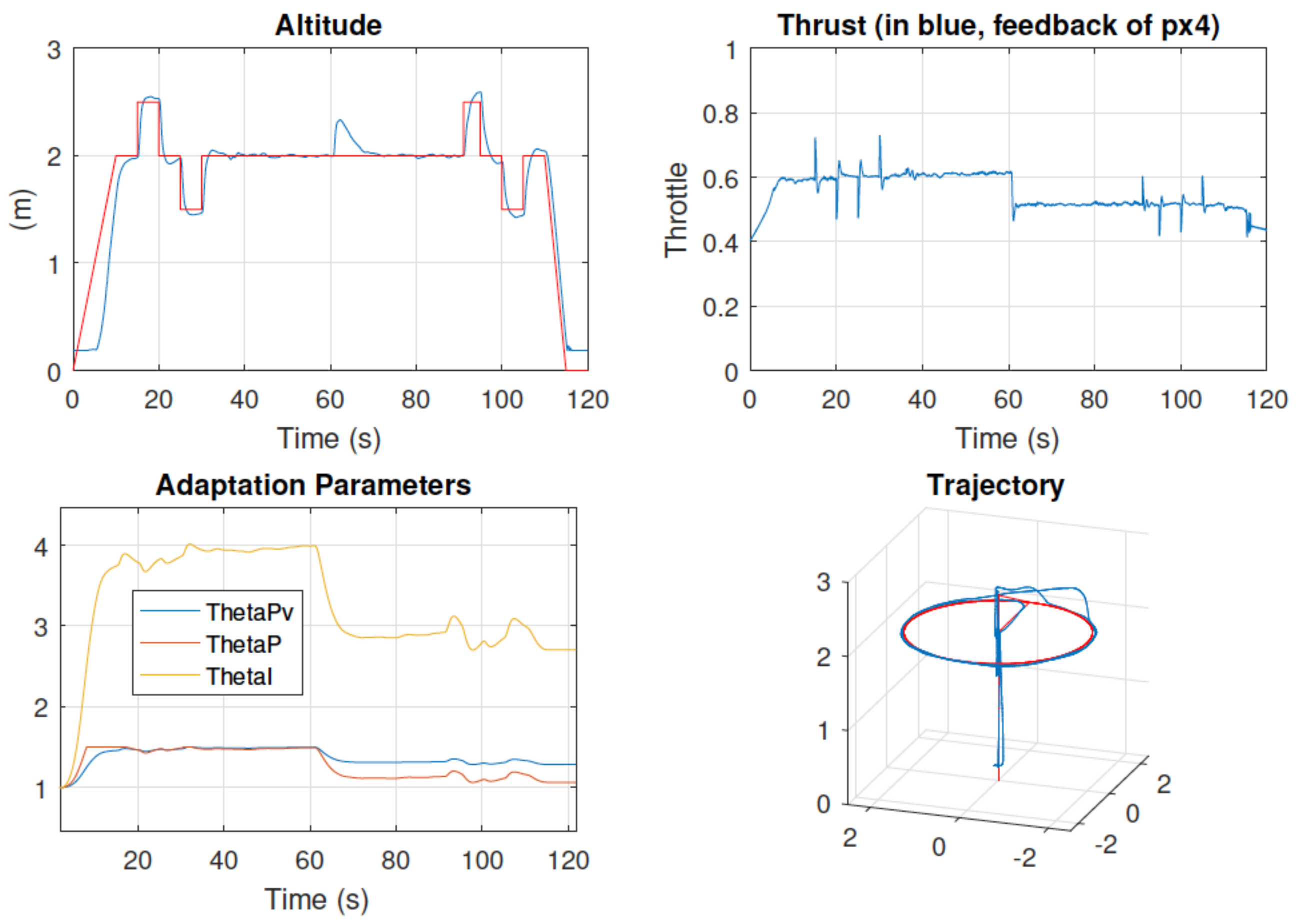

Figure 5: Circle mission results with MRAC (400g payload)

low altitude (subjecting the drone to ground effect) were also achieved with success, highlighting the flexibility and adaptability of the MRAC.

6 Analysis of the adaptation process

In order to highlight the potentialities of adapting the controller parameters, two experiments have been conducted. During the first one the adaptation has been frozen after some time to evaluate the expected loss of performances. For the second one, the initial UAV has been replaced by a completely different one but keeping the same controller.

6.1 *Freezing the adaptation*

The performance with full adaptation has already been presented in Figure 5. In the new test recorded in Figure 7, the convergence of the adaptation parameters is voluntarily **frozen** after 40s. Thus, convergence to the unloaded state will never be achieved.

By comparing Figures 5 and 7:

1. **Mass drop performance:** As expected, with the frozen adaptation the recovery is much slower, taking **18s** in comparison with the **6s** achieved with the full adaptation. Nevertheless, in both cases the drone only climbs **33 cm** before starting the recovery.
2. **Unloaded performance:** Due to the slow convergence from the mass drop, the drone does not have enough time to perform the full step sequence (since the mission is fixed at 120 s for battery reasons). However, on the final steps it can be seen that the behaviour of the drone is slightly better than the one recorded in the fully adapted case (Figure 5).

It can be concluded that the dynamic adaptation allows the drone to better **deal with the extreme disturbance** caused by the mass drop. The initial adaptation (after the take-off) provides however a **better tracking**.

6.2 *Replacing the UAV*

The formerly UAV (Mikrokopter Mk.6) has been replaced by a Parrot AR drone. The previous structure of the

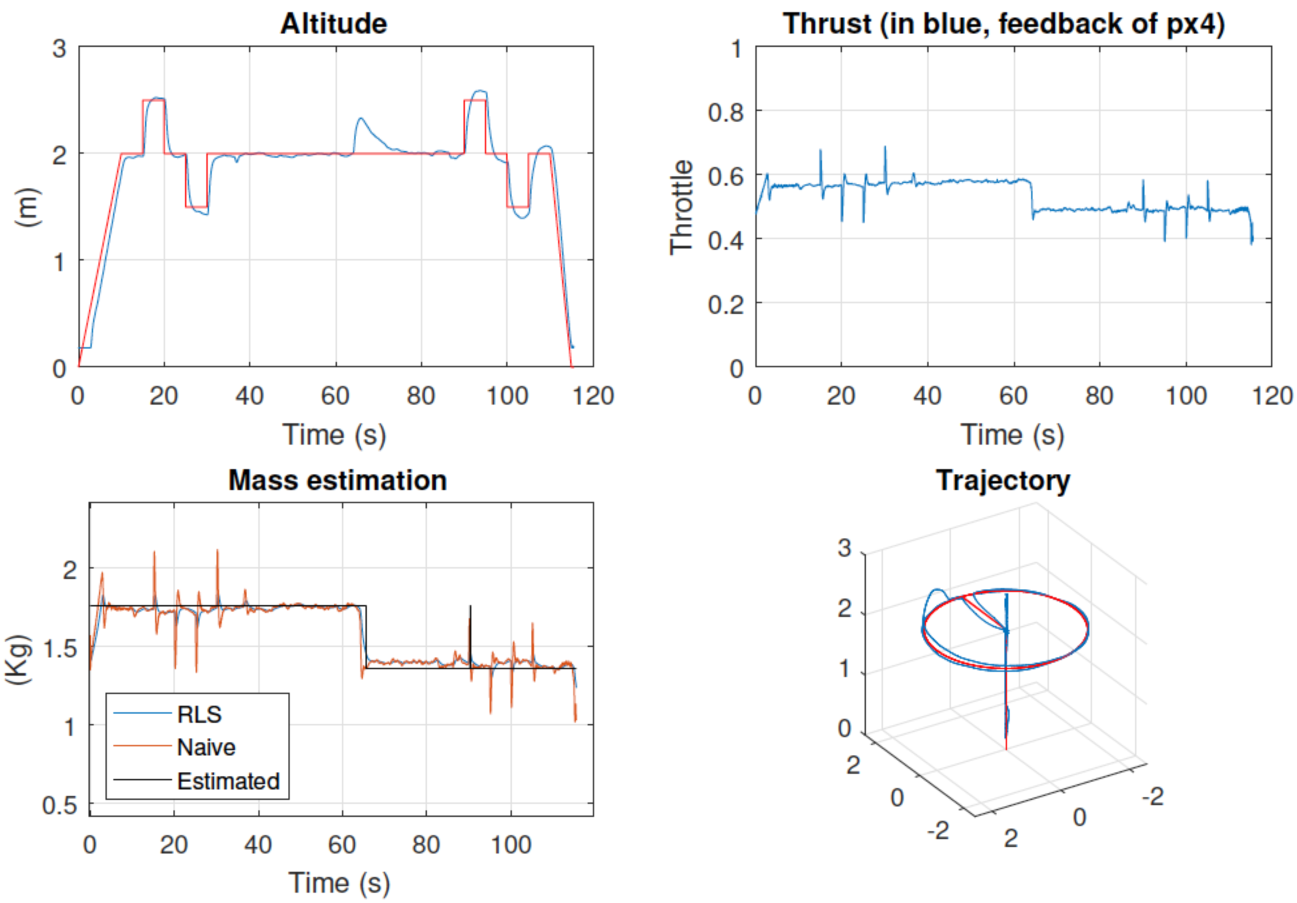

Figure 6: Circle mission results with GS (400g payload)

Specifications	Parrot AR	Mikrokopter Mk.6
Weight (kg)	0.713	1.360
Arm length (cm)	18	22.5
Propeller ⌀(inch)	8	10

Table 1: Parrot AR vs Mikrokopter Mk.6

controller is not modified, the initial tuning of the controller gains either. As seen in Figure 8, even when using a completely different drone, the MRAC ensures convergence of the controller gains, and a safe flight is possible.

A summary of the differences between the Parrot AR used in this flight and the Mikrokopter Mk.6 used elsewhere is given in Table 1 to illustrate the huge differences between the 2 drones; highlighting the MRAC convergence.

7 Conclusion

The objective of this work was to demonstrate that an adaptive controller represents a real alternative for the control of UAVs subject to severe changes of their dynamic behaviour. Model Reference Adaptive Control has proven to be a convincing and reliable solution. Indeed it allows to enrich a baseline controller with two important properties:

- autotuning of the controller gains from poor initial conditions.
- adaptation of the controller gains when needed.

Besides, the MRAC also proved itself very flexible, and may be used to control different drones which are forced into missions characterized by extreme perturbations. Future work may include more intensive testing of the behaviour of the current MRAC Controller for coping with specific conditions related to UAVs flight, like ground effect disturbances or engine failure.
Another interesting axis concerns the adaptation gains: Even if they represent an extra degree of freedom to improve the overall closed loop behaviour, their thin tuning may result time consuming. This is why we are actually working on the design of L1 type adaptive controllers, which belong to the

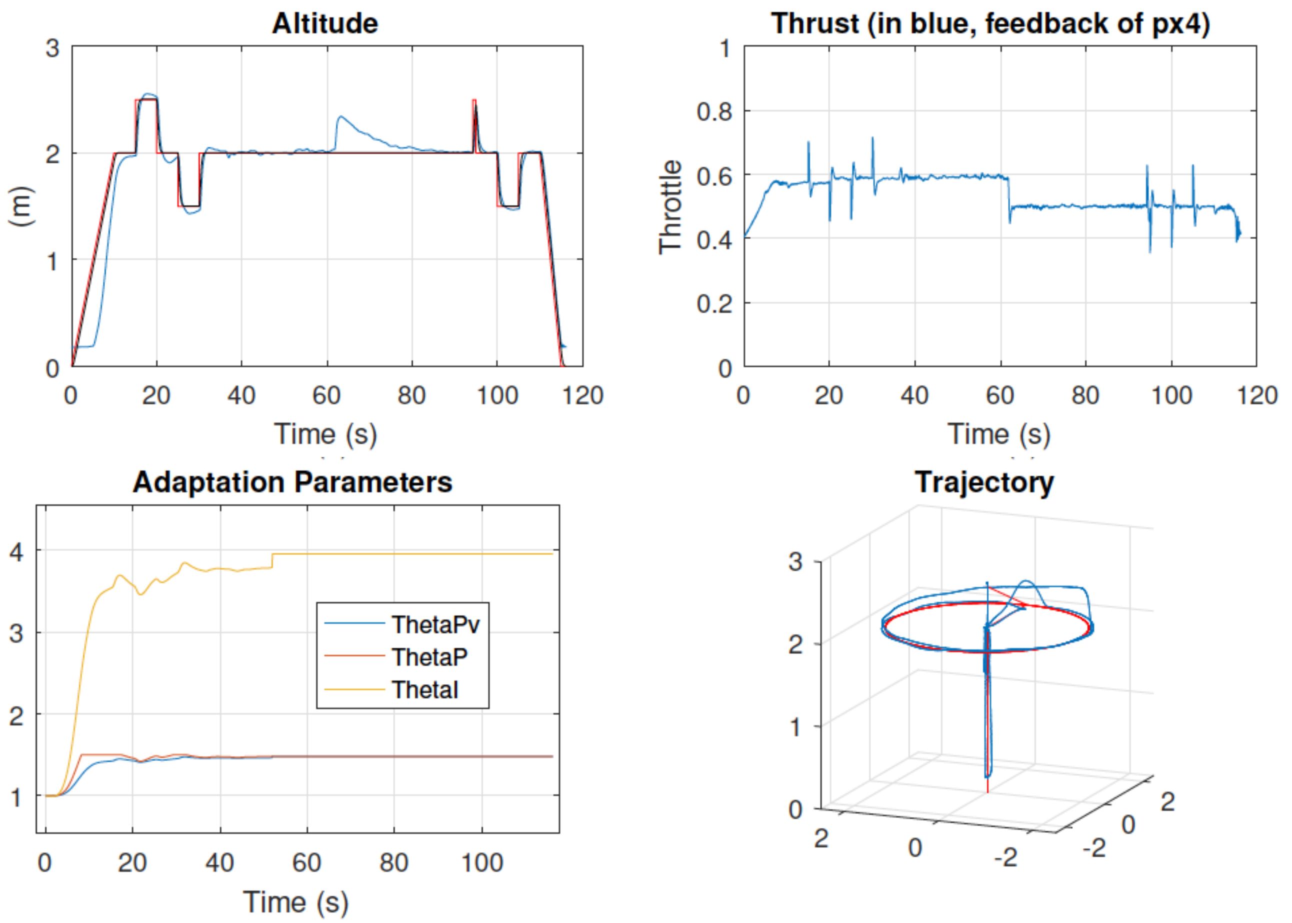

Figure 7: Fixed Adaptation Parameters with MRAC

same family while exhibiting interesting convergence properties when very high tuning gains are applied.

Finally, the Gain Scheduling controller also offers new windows of research. Since its performance is very sensible to the mass estimation, more elaborate methods of systems identification, like Kalman filtering, would be a great addition to the existing controller.

REFERENCES

[1] J. Roberts, D. Frousheger, B. Williams, D. Campbell, and R. Walker. How the outback challenge was won: The motivation for the uav challenge outback rescue, the competition mission, and a summary of the six events. *IEEE Robotics Automation Magazine*, 23(4):54–62, Dec 2016.

[2] Saleen Bhattarai, Kushal Poudel, Nishant Bhatta, Sanjay Mahat, Sudip Bhattarai, and Kaman S Thapa Magar. Modeling and development of baseline guidance navigation and control system for medical delivery uav. In *2018 AIAA Information Systems-AIAA Infotech@ Aerospace*, page 0508. 2018.

[3] Daniel Mellinger, Michael Shomin, Nathan Michael, and Vijay Kumar. *Cooperative Grasping and Transport Using Multiple Quadrotors*, pages 545–558. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[4] Kimon P. Valavanis and George J. Vachtsevanos. *Future of Unmanned Aviation*, pages 2993–3009. Springer Netherlands, Dordrecht, 2015.

[5] B. Min, J. Hong, and E. T. Matson. Adaptive robust control (arc) for an altitude control of a quadrotor type uav carrying an unknown payloads. pages 1147–1151, Oct 2011.

[6] Maxime Lecointe, Caroline Ponzoni Carvalho Chanel, and Franois Defay. Backstepping control law application to path tracking with an indoor quadrotor. In *Proceedings of European Aerospace Guidance Navigation and Control Conference (EuroGNC)*, pages pp.1–19, Toulouse, FR, April 2015.

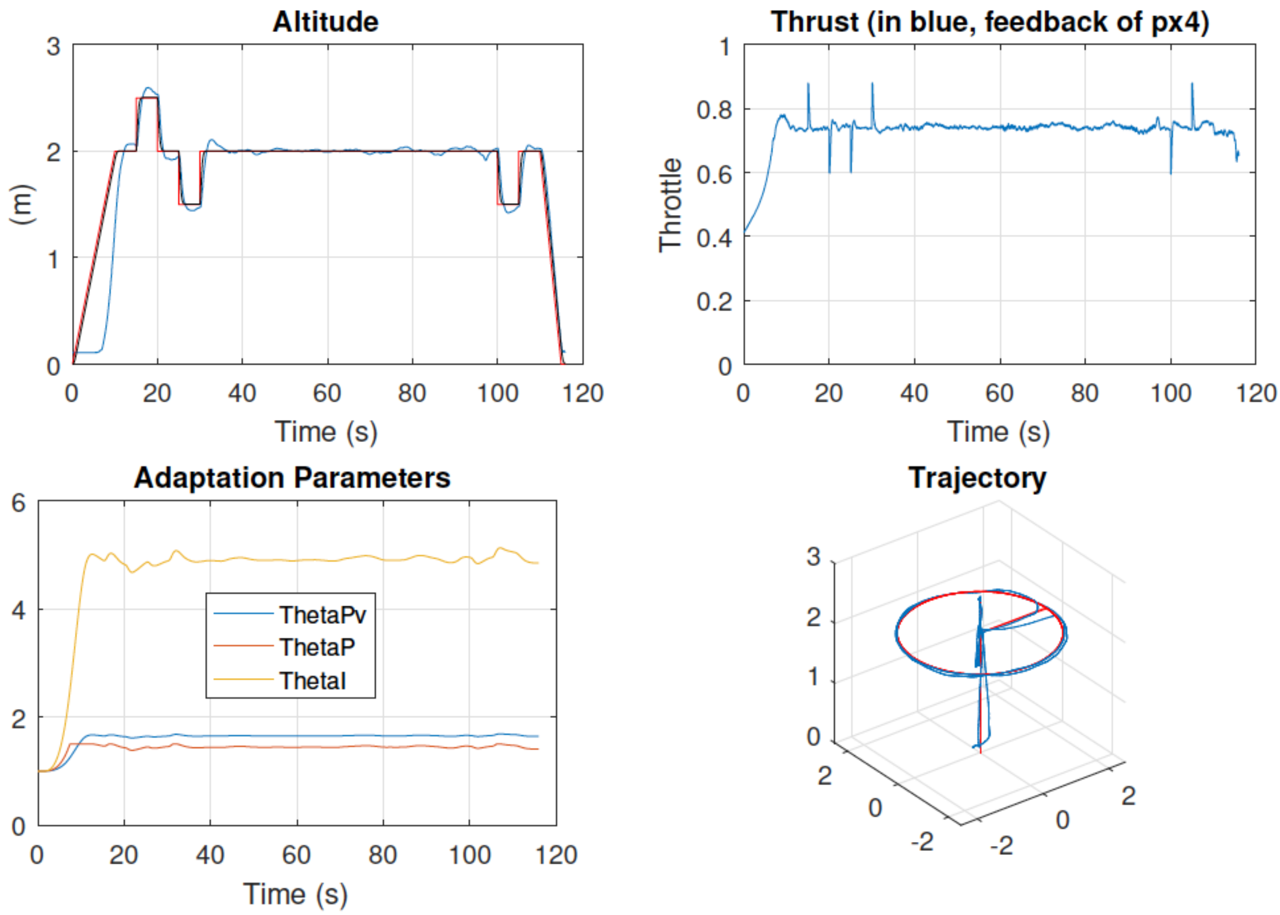

Figure 8: MRAC test with Parrot AR

[7] J. A. Maces Hernandez. *Landing an UAV in a moving platform, Master Research Project S2*. ISAE-SUPAERO, Toulouse, 2016.

[8] Yann Ameho, Fabien Niel, François Defaÿ, Jean-Marc Biannic, and Caroline Bérard. Adaptive control for quadrotors. In *ICRA*, pages 5396–5401, 2013.

[9] Teppo Luukkonen. Modelling and control of quadcopter. *Aalto University, Independent research project in applied mathematics*, pages 1400–1406, 2011.

[10] E Lavretsky and Kevin A. Wise. *Robust and Adaptive Control*. Springer London, London, 2013.

[11] Karl J. Astrm and Bjrn Wittenmark. *Adaptive Control*. Addison-Wesley, Reading, Massachusetts, 1995.

[12] M. A. Gmez Tierno, M. Prez Corts, and C. Puentes Mrquez. *Mecanica del Vuelo [Spanish]*. Garceta, Madrid, 2012.