Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 53-2 (2020) 9340–9347

Universal Adaptive Fault-Tolerant Control of a Multicopter UAV ⋆

Duc-Tien Nguyen * **David Saussié** * **Lahcen Saydy** *

* *Mobile Robotics and Autonomous Systems Laboratory, Polytechnique Montreal, Canada (e-mail: {duc-tien.nguyen, d.saussie, lahcen.saydy}@polymtl.ca).*

Abstract: This paper presents a universal adaptive fault-tolerant control (FTC) design for multicopter unmanned aerial vehicles (UAVs). The proposed architecture consists of a two-loop control structure: a fault-tolerant controller generates normalized virtual control inputs to track the desired trajectory subject to actuator faults, and an adaptive augmentation controller deals with system uncertainties and also balances the design requirements for specific platform. The FTC approach is based on gain-scheduling control in the framework of structured $\mathcal{H}_\infty$ synthesis. In order to implement the overall control system on most types of multicopter UAVs, an adaptive mapping algorithm is proposed. High fidelity simulations and experimental results, performed on various multicopters with different payload and configuration, show the effectiveness and robustness of the proposed approach in accommodating different levels of actuator degradation including total failures of the rotors as well as unknown mass and inertia, all using a single controller with fixed coefficients.

Copyright © 2020 The Authors. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0)

Keywords: Multicopter, Fault-Tolerant Control, Robust Adaptive Control, Gain-Scheduling, Structured H-Infinity Control.

1. INTRODUCTION

The ability of multicopter unmanned aerial vehicles (UAVs) to perform vertical take-off and landing as well as stationary hover flight makes them intensively useful for various commercial and military applications (e.g., reconnaissance and exploration, parcel delivery, or search-and-rescue operations). In order to achieve high autonomy and safety flight, UAVs should possess a fault-tolerant ability to accommodate malfunctions in actuators, sensors or other system components. Several fault tolerant control (FTC) approaches have been developed in the literature but typically limited to a specific platform. Designing a universal fault-tolerant controller which is capable of maneuvering for a variety of multicopter UAVs seems to be an open research problem.

An adequate survey on two main categories of FTC (active and passive) can be found in (Zhang and Jiang, 2008). In this paper, we consider an active FTC system for a multicopter that uses an online automatic redesign mechanism. Various methods have been developed in the literature such as gain-scheduling (GS) (Zhang et al., 2013; Nguyen et al., 2017), control allocation (CA) (Yoon et al., 2016; Falconi and Holzapfel, 2016; Nguyen et al., 2018), or robust adaptive control (Mallavalli and Fekih, 2018). As reported in (Zhang et al., 2013), a gain-scheduled proportional-integral-derivative (PID) controller was implemented on a quadcopter and experimentally tested with a 18% loss of actuator effectiveness (LAE) in all motors. However, the fault detection and diagnosis (FDD) module was not integrated into the complete FTC system. Furthermore, the use of look-up tables to store pre-computed controller gains and switching between them during the flight can produce undesirable transients (Zhang and Jiang, 2008). In order to alleviate these problems, an integrated system composed of a two-stage Kalman filter (TSKF) and an $\mathcal{H}_\infty$ gain-scheduled controller was proposed in our previous works (Nguyen et al., 2017). In this case, the scheduled gains are automatically adjusted as functions of the scheduling variables, i.e., the LAE factors given by the TSKF. A good behavior in fault recovery after multiple successive losses of control effectiveness in a quadcopter was shown. In addition, a combination of this control design technique and structured $\mathcal{H}_\infty$ synthesis can offer a more robust performance (Gahinet and Apkarian, 2011; Lhachemi et al., 2014).

By dealing directly with the loss of effectiveness of each actuator, this control design exhibits a good behavior in fault recovery. However, the more actuators the vehicle has, the more complex the control design is. In fact, the computational cost becomes more expensive due to the larger number of faulty models considered as well as controller coefficients. Moreover, the control law depends on the multicopter configuration, meaning that it is different for each kind of UAV. On the other hand, the computational burden of updating the controller gains is not suitable for low-cost implementations. To tackle the problem, we propose a FTC scheme relying on the loss of virtual control effectiveness (LVE), given by an appropriate FDD system. The parametric gains are chosen as smooth functions of LVE factors, rather than in terms of LAE. The goal is to reduce the complexity of the tuning process and to find a general control structure applicable

⋆ This work was supported by NSERC under grant numbers RGPIN-2014-03942 and RGPIN-2012-122106.

2405-8963 Copyright © 2020 The Authors. This is an open access article under the CC BY-NC-ND license.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2020.12.2390

to any UAVs including fixed-wing vehicles. Besides, using slower-varying LVE is more suitable with gain scheduling control (Rugh and Shamma, 2000).

We are also interested in the design of a generalized control law which is applicable to most traditional multicopters. Indeed, the effect of the motors is the same: they create a thrust force and a reaction torque along a body fixed axis. Then, using the normalized virtual inputs, the control strategy is not dependent on the multicopter configuration. In order to adapt to the design requirements as well as the system uncertainties, a model reference adaptive control (MRAC) augmentation is proposed. The algorithm computes propeller's thrust force to track the reference position and orientation with respect to the dynamic feasibility of the specific platform.

In summary, the proposed controller has many advantages such as simplification of the control structure with explicit formulas to redesign automatically in real-time, smooth transitions between operating points in the presence of actuator faults, low computational load during the offline optimization process, as well as applicability for different platforms with the same controller parameters. The remainder of the paper is organized as follows. Section 2 describes the mathematical model of a multicopter UAV. The detailed design procedure of the proposed FTC system is presented in Section 3. Section 4 introduces the adaptive augmentation strategy. Simulation and experimental results are given in Section 5. Finally, the conclusion and future works are conducted in Section 6.

2. PLANT DESCRIPTION

This section first introduces the modeling of a multicopter UAV, followed by the simplified model and the plant linearization that will be used for the controller design.

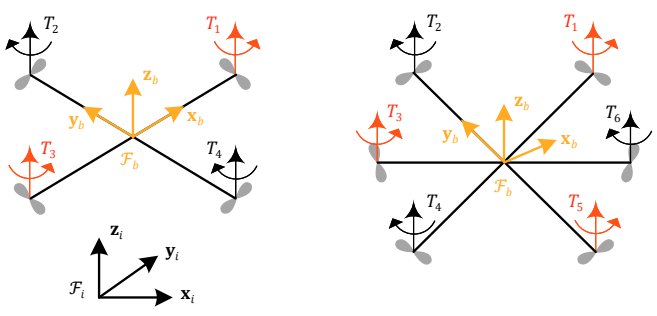


Fig. 1. Frames of reference with illustrations of a quadcopter (left) and an hexacopter (right).

2.1 Multicopter Dynamics

Let $\mathcal{F}_i$ denote the inertial frame centered at a fixed local reference point and respecting the East-North-Up (ENU) orientation convention, and let $\mathcal{F}_b$ denote the body frame with origin at the UAV's center of mass (CM) with the Forward-Left-Up (FLU) convention as in Fig. 1. In the following, the superscript of a vector indicates the frame in which it is expressed, e.g., $\mathbf{p}^i$ or $\mathbf{p}^b$.

The position vector of the CM in $\mathcal{F}_i$ is denoted by $\mathbf{p}^i_{\mathrm{CM}/i} = [x\ y\ z]^\top$, and the orientation of $\mathcal{F}_b$ relative to $\mathcal{F}_i$ is described by three Euler angles $\mathbf{\Phi} = [\phi\ \theta\ \psi]^\top$, representing respectively roll, pitch and yaw. The rotation matrix $\mathbf{R}_{b/i}$ resulting from a yaw-pitch-roll sequence reads:

$$\mathbf{R}_{b/i} = \begin{bmatrix} \mathrm{c}_\theta \mathrm{c}_\psi & \mathrm{c}_\theta \mathrm{s}_\psi & -\mathrm{s}_\theta \\ \mathrm{s}_\phi \mathrm{s}_\theta \mathrm{c}_\psi - \mathrm{c}_\phi \mathrm{s}_\psi & \mathrm{s}_\phi \mathrm{s}_\theta \mathrm{s}_\psi + \mathrm{c}_\phi \mathrm{c}_\psi & \mathrm{s}_\phi \mathrm{c}_\theta \\ \mathrm{c}_\phi \mathrm{s}_\theta \mathrm{c}_\psi + \mathrm{s}_\phi \mathrm{s}_\psi & \mathrm{c}_\phi \mathrm{s}_\theta \mathrm{s}_\psi - \mathrm{s}_\phi \mathrm{c}_\psi & \mathrm{c}_\phi \mathrm{c}_\theta \end{bmatrix}$$
$$\text{with } \ c_x := \cos x, \ s_x := \sin x.$$

The rotation matrix $\mathbf{R}_{b/i}$ belongs to the special orthogonal group SO(3), and thus one has $\mathbf{R}_{i/b} = \mathbf{R}^{-1}_{b/i} = \mathbf{R}^\top_{b/i}$. The angular kinematics of the UAV is given by Euler's kinematic equation:

$$\dot{\mathbf{\Phi}} = \mathbf{H}(\mathbf{\Phi})\boldsymbol{\omega}^b_{b/i} \tag{1}$$

where $\boldsymbol{\omega}^b_{b/i} = [p\ q\ r]^\top$ is the angular velocity of $\mathcal{F}_b$ with respect to $\mathcal{F}_i$ expressed in $\mathcal{F}_b$, and $\mathbf{H}(\mathbf{\Phi})$ is the transformation matrix for angular velocities given by:

$$\mathbf{H}(\mathbf{\Phi}) = \begin{bmatrix} 1 & \mathrm{s}_\phi \mathrm{t}_\theta & \mathrm{c}_\phi \mathrm{t}_\theta \\ 0 & \mathrm{c}_\phi & -\mathrm{s}_\phi \\ 0 & \mathrm{s}_\phi / \mathrm{c}_\theta & \mathrm{c}_\phi / \mathrm{c}_\theta \end{bmatrix} \quad \text{with } \mathrm{t}_x := \tan x$$

The application of Newton's second law yields the dynamic equations of motion:

$$m\ddot{\mathbf{p}}^i_{\mathrm{CM}/i} = m\mathbf{g}^i + \mathbf{R}_{i/b}\mathbf{F}^b \tag{2}$$
$$\mathbf{I}_b \dot{\boldsymbol{\omega}}^b_{b/i} = -\boldsymbol{\omega}^b_{b/i} \times \mathbf{I}_b \boldsymbol{\omega}^b_{b/i} + \mathbf{M}^b \tag{3}$$

where $\mathbf{F}^b$ and $\mathbf{M}^b$ are the forces and the moments created by the rotors, $\mathbf{g}^i = [0\ 0\ -g]^\top$ the gravity vector, m the mass of the multicopter, $\mathbf{I}_b$ the inertia matrix about the center of mass and $\times$ the cross product. In frame $\mathcal{F}_b$, the forces and the moments are denoted by:

$$\mathbf{F}^b = [0\ \ 0\ \ T]^\top, \ \ \mathbf{M}^b = [L\ \ M\ \ N]^\top$$

with T represents the total thrust, L, M and N the roll, pitch and yaw moments respectively.

Considering a multicopter system with n actuators (rotors and propellers) attached to a rigid body frame. Each actuator $i \in \{1, \ldots, n\}$ produces a thrust T_i and a reaction torque M_i along the $\mathbf{z}_b$-axis given by:

$$T_i = k_T \omega_i^2, \ \ M_i = -\operatorname{sign}(\omega_i) k_D \omega_i^2$$

where k_T the thrust coefficient, k_D the drag factor, and ω_i the rotor rotational speed. According to the $\mathbf{z}_b$-axis pointing upwards (Fig. 1), ω_i is positive if the i-th rotor rotates anticlockwise and vice versa. The relation between the virtual control input vector $\mathbf{v} = [T\ L\ M\ N]^\top$ and the thrust vector generated by the actuators $\mathbf{u} = [T_1\ \ldots\ T_n]^\top$ is obtained from:

$$\mathbf{v} = \mathbf{B}_{\mathrm{CA}}\mathbf{u} \tag{4}$$

where $\mathbf{B}_{\mathrm{CA}}$ is the control allocation (CA) matrix depending on the multicopter configuration. For example, an hexacopter UAV model with standard NPNPNP [1] rotor arrangement (Fig. 1) has the following mapping matrix:

$$\mathbf{B}_{\mathrm{CA}} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ \frac{d}{2} & d & \frac{d}{2} & -\frac{d}{2} & -d & -\frac{d}{2} \\ -\frac{d\sqrt{3}}{2} & 0 & \frac{d\sqrt{3}}{2} & \frac{d\sqrt{3}}{2} & 0 & -\frac{d\sqrt{3}}{2} \\ -c & c & -c & c & -c & c \end{bmatrix}$$

with d the arm length and $c = k_D/k_T$ the force-to-moment scaling factor. In the case of a quadcopter according to the "+" configuration (Fig. 1), the CA matrix is given by:

[1] The letters P and N denote a positive and a negative reaction torque about the $\mathbf{z}_b$-axis, respectively.

$$\mathbf{B}_{\mathrm{CA}} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & d & 0 & -d \\ -d & 0 & d & 0 \\ -c & c & -c & c \end{bmatrix}$$

Instead of controlling individually motor's thrust force, it can be more convenient to use the virtual input $\mathbf{v}$ to command particular movements of the drone. Thus, the desired thrust force for each motor is allocated by:

$$\mathbf{u} = \mathbf{B}_{\mathrm{CA}}^{\dagger}\mathbf{v} \tag{5}$$

where † is the Moore-Penrose pseudo-inverse.

2.2 Simplification and Linearised model

Since all UAVs share the same dynamic equations (1-3) with different parameters (mass and inertia), we first define the normalized forces and moments as follows:

$$\bar{\mathbf{v}} = \left[\bar{T}\ \bar{L}\ \bar{M}\ \bar{N}\right]^{\top} = \left[T/m\ \ \left(\mathbf{I}_b^{-1}\mathbf{M}^b\right)^{\top}\right]^{\top} \tag{6}$$

Thus, we are interested in the design of a *single* controller with *fixed* parameters that generate the normalized vector $\bar{\mathbf{v}}$. The change of system input requires an additional mapping equation to get the virtual control signal:

$$\mathbf{v} = \Upsilon\bar{\mathbf{v}} \ \text{ with } \Upsilon = \mathrm{diag}\left(m, \mathbf{I}_b\right) \tag{7}$$

Such a universal controller can be used on most types of multicopter UAVs. Besides, due to the fact that the proper parameters of each multicopter are not considered in the controller formulation, an adaptive mechanism is proposed to address this issue.

Moreover, the following assumptions about the simplified model are required during the design of the controller:

Assumption 1. The UAV structure is symmetric with diagonal inertia matrix, i.e., $\mathbf{I}_b = \mathrm{diag}\left(I_{xx}, I_{yy}, I_{zz}\right)$.

Assumption 2. The actuator dynamics, including rotor speed controllers, are considered sufficiently fast to neglect them; thus the control inputs are directly the thrust forces created by the propellers.

Assumption 3. The position $\mathbf{p}^i_{\mathrm{CM}/i}$, attitude $\mathbf{\Phi}$, linear and angular velocities $\dot{\mathbf{p}}^i_{\mathrm{CM}/i}$, $\boldsymbol{\omega}^b_{b/i}$ of the UAV are known.

These assumptions are justified for the majority of traditional multicopter UAVs.

In order to design the controller, the nonlinear model (1-3) is trimmed and linearized by assuming hovering flight ($T = mg$, $L = M = N = \phi = \theta = 0$) with CM position at the origin of $\mathcal{F}_i$ ($x = y = z = 0$) and null yaw ($\psi = 0$) for simplicity's sake. To this end, each rotor i delivers $T_i = mg/n$. This yields the classic linearized equations:

$$\begin{aligned} \Delta\ddot{x} &= g\Delta\theta & \Delta\dot{p} &= \Delta\bar{L} & \Delta\dot{\phi} &= \Delta p \\ \Delta\ddot{y} &= -g\Delta\phi & \Delta\dot{q} &= \Delta\bar{M} & \Delta\dot{\theta} &= \Delta q \\ \Delta\ddot{z} &= \Delta\bar{T} & \Delta\dot{r} &= \Delta\bar{N} & \Delta\dot{\psi} &= \Delta r \end{aligned} \tag{8}$$

where Δ denotes the deviation of a variable from its equilibrium value[2].

3. UNIVERSAL FTC DESIGN

As reported in our previous works, a state feedback controller of a quadcopter (Nguyen et al., 2017) and a control allocation algorithm of a hexacopter (Nguyen et al., 2018) relying on gain-scheduling control in the framework of structured $\mathcal{H}_\infty$ synthesis showed a good fault recovery capability in both simulation and experimental results. Taking advantage of this design approach, we propose a generalized fault-tolerant controller that produces the normalized virtual input $\bar{\mathbf{v}}$, which is capable of performing on most multicopter UAVs. Note that the system uncertainties such as unknown mass are not taken into account during the design of the controller. In order to compensate the effect of unknown parameters, an adaptive augmentation will be introduced in the next section.

In order to facilitate the FTC design, the control architecture is separated into four state feedback controllers with integral actions (Fig. 2). Due to the similarity in the design process, only the x-pitch controller is further explained.

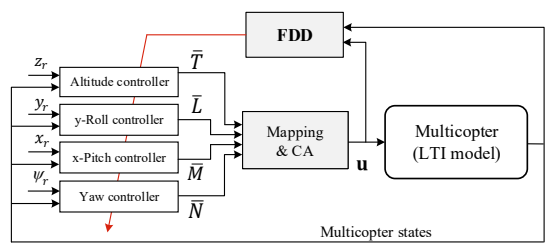


Fig. 2. Universal FTC architecture for design purposes.

3.1 Multi-models and gain-scheduling synthesis

Assuming no uncertainties in the plant dynamic, the reduced LTI state-space model used for x-pitch FTC design including the LVE factor can be represented as:

$$\boldsymbol{\Delta}\dot{\mathbf{x}}_x = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & g & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{A}_x} \boldsymbol{\Delta}\mathbf{x}_x + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1-\gamma_M \end{bmatrix}}_{\mathbf{B}_x} \Delta\bar{M} \tag{9}$$

with the sub-model state vector $\boldsymbol{\Delta}\mathbf{x}_x = [\Delta x\ \Delta\dot{x}\ \Delta\theta\ \Delta q]^{\top}$ and $\gamma_M \in [0, 1]$ represents the loss of pitch moment effectiveness, as will be discussed later in this section. $\gamma_M = 1$ means that the vehicle can not generate any pitch moment and $\gamma_M = 0$ means that it is completely healthy. In the framework of x-pitch FTC design, a set of 17 plant models for different values of γ_M in the interval $[0, 0.8]$ is considered. It is worth noting that a larger range of γ_M can cause a violation of design requirements due to high gains or, worse, a divergence of the tuning process.

The control formulation is achieved by augmenting the linear system matrices (9) with the new state Δe_{xI} that represent the integral of the error signal:

$$\Delta\dot{e}_{xI} = \Delta x_r - \Delta x$$

where Δx_r is the reference input of the regulated output $\Delta x = [1\ 0\ 0\ 0]\,\boldsymbol{\Delta}\mathbf{x}_x = \mathbf{C}_x\boldsymbol{\Delta}\mathbf{x}_x$. The system representation is then augmented with the new state as follows:

$$\begin{bmatrix} \boldsymbol{\Delta}\dot{\mathbf{x}}_x \\ \Delta\dot{e}_{xI} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{A}_x & 0 \\ -\mathbf{C}_x & 0 \end{bmatrix}}_{\mathbf{A}_a} \begin{bmatrix} \boldsymbol{\Delta}\mathbf{x}_x \\ \Delta x_i \end{bmatrix} + \underbrace{\begin{bmatrix} \mathbf{B}_x \\ 0 \end{bmatrix}}_{\mathbf{B}_a} \boldsymbol{\Delta}\bar{M} + \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} \Delta x_r$$

where $\mathbf{0}$ denotes the null vector of appropriate dimensions. The augmented pair $(\mathbf{A}_a, \mathbf{B}_a)$ is controllable if and only if

[2] Actually 0 for most of them.

$\gamma_M \neq 1$. In this case, the state feedback control law with integral action that will bring the output tracking error $\Delta\dot{e}_{xI}$ to zero is then:

$$\Delta\bar{M} = -\mathbf{K}_x \mathbf{\Delta x}_x + K_{i,x}\Delta e_{xI}$$

Nominal controller In the nominal case ($\gamma_M = 0$), the controller gains $\mathbf{K}_x = [K_x \ K_{\dot{x}} \ K_\theta \ K_q]$ and $K_{i,x}$ can be designed using LQR synthesis or classical linear control techniques. Thanks to the knowledge gathered on our system through flight experiments, the closed-loop poles are placed at $[-2.5, \ -2.25 \pm 3.85j, \ -1.45 \pm 1.65j]$.

Fault-tolerant controller The goal of the FTC is to maintain the system stability and to meet acceptable performance in both nominal and degraded modes of operation. To this end, the state feedback gain $\mathbf{K}_x$ and the integrator gain $K_{i,x}$ are modeled as quadratic functions of the scheduling variable γ_M.

$$\mathbf{K}_x = \mathbf{K}_{x_0} + \mathbf{K}_{x_1}\gamma_M + \mathbf{K}_{x_2}\gamma_M^2$$
$$K_{i,x} = K_{i,x_0} + K_{i,x_1}\gamma_M + K_{i,x_2}\gamma_M^2$$

The entries of the matrices $\mathbf{K}_{x_j} \in \mathbb{R}^{1\times 4}$ and the scalars K_{i,x_j}, for $j \in [0, 2]$, are the tunable parameters of the corresponding gains. Thus, there are 15 coefficients that the tuning software adjusts to meet the tuning goals on the entire set of models. In order to precisely compare the performance of the nominal and the proposed controllers in the same context, the coefficients $\mathbf{K}_{x_0}$ and K_{i,x_0} can be fixed by the nominal gain values, i.e., are not free to be tuned. Indeed, the same control inputs are generated by the nominal and proposed controllers in the fault-free situation ($\gamma_M = 0$). Moreover, only 10 coefficients have to be tuned in this design. Note that the normalized value of γ_M in range $[-1, 1]$ can be used to improve the performance of the optimization solver. However, the desired behavior in the nominal operation will be changed.

In this paper, we use the MATLAB Robust Control Toolbox function `systune` to tune these coefficients[3] subject to specified design requirements. The tuning is performed by leveraging on the multi-model capabilities of `systune`. Indeed, several models with different LVE values can be considered at the same time during the synthesis. The tuning procedure is detailed in the next subsection.

3.2 Robust $\mathcal{H}_\infty$ synthesis

In the framework of structured $\mathcal{H}_\infty$ control, the design requirements can be defined as frequency weights with MATLAB function `frd` to create the frequency-response data (FRD) model. For example, in order to reject disturbances, the FRD model created by `frd`([0.01 0.01 1], [0 0.1 1]) ensures that a disturbance injected at actuator inputs is suppressed at the output by a gain less than 0.01 (-40 dB), in the low frequency range $[0, 0.1]$ rad/s. This tuning goal is converted into the regularized formulation of $\mathcal{H}_\infty$ constraint by `systune`:

$$\|WT\|_\infty < 1 \tag{10}$$

where T denotes the closed-loop transfer function from specified inputs to outputs and W is a frequency weighting function specifying the requirement. In this case,

$$W = \frac{1}{(s+0.1)^2}$$

is a low-pass filter obtained using the MATLAB function `getWeight`. Similarly, the noise introduced by the sensors is attenuated at the output using the FRD model specified by `frd`([1 1 0.001 0.0001], [0 10 100 1000]). The equivalent regularized weighting function W_2 of this requirement is a high-pass filter:

$$W_2 = \frac{1250000(s+10)^3}{(s+125000)(s+100)^2}$$

More control requirements for a medium size multicopter can be found in Tab. 1 such as response time, overshoot, closed-loop pole confinement constraints, etc. Due to the fact that the control design is not for aggressive tracking purpose, the tracking requirement is only restricted in the low frequency range[4] $[0, 0.1]$ rad/s. This relaxation makes it easier to achieve the desired performance.

Table 1. Design requirements.

Requirement		Objective
1	tracking (x, y, z)	response time 3 s
		steady-state error $< 0.01\%$
	tracking (ψ)	response time 4 s
		steady-state error $< 0.01\%$
2	overshoot	1%
3	close-loop poles	damping constant > 0.6
		decay rate > 1.5
4	open-loop stability	gain margin > 12 dB
		phase margin $> 45°$
5	disturbance rejection	`frd`([0.01 0.01 1], [0 0.1 1])
6	noise rejection	`frd`([1 1 10^{-3} 10^{-4}], [0 10 100 1000])
7	actuator saturation	`frd`([0.05 0.5 5], [0 0.1 1])

The robust tuning algorithm `systune` is performed on a quad-core PC using MATLAB 2019b. After 40 iterations, the highest $\mathcal{H}_\infty$ norm of this synthesis is 1.19 as summarized in Tab. 2. Therefore the requirements are not entirely, but almost satisfied.

Table 2. $\mathcal{H}_\infty$ norm of tuning goals.

Requirement	1	2	3	4	5	6	7
$\mathcal{H}_\infty$ norm	1.18	0.93	1.19	0.6	0.25	1	0.44

The synthesis yields the resulting gain curve given in Fig. 3. By estimating online the LVE factor γ_M, the controller gain can be updated in real time to maintain an adequate level of performance. In fact, using the proposed controller, the closed-loop poles still remain in the required damping region (Fig. 4a). Hence the system's stability and performance are always guaranteed (Fig. 4b). Conversely, the feedback system using the nomimal controller (not scheduled with γ_M) begins to be unstable when $\gamma_M \geq 0.6$. Considering the full system, the instability can come earlier due to the losses of other virtual control inputs as it will be seen in Sec. 5.

To complete the design, the robustness of both nominal and proposed controllers with respect to actuator faults are compared using the $\mathcal{H}_\infty$ norm of the combined objective for different faulty models (γ_M from 0 to 1). As shown in Tab. 3, the nominal controller is rather sensitive to

[3] Defined with the MATLAB function `tunableGain`.

[4] Specified by the `Focus` property of the `TuningGoal` object.

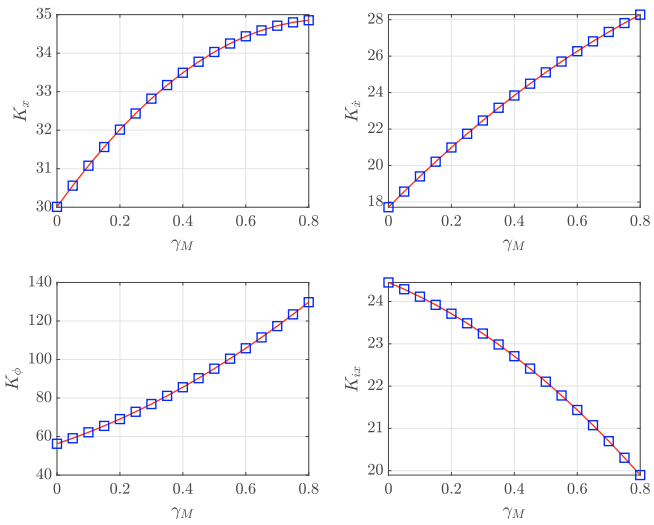
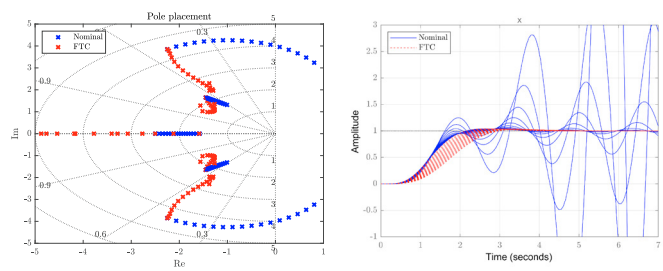


Fig. 3. Evolution of controller gains versus the LVE.



(a) Closed-loop poles (b) Step response

Fig. 4. Closed-loop system response.

change in the losses of control effectiveness and unstable if $\gamma_M \geq 0.6$, while the scheduled $\mathcal{H}_\infty$ controller is very robust to these faults as expected. This significant improvement is the main advantage of the proposed synthesis.

Table 3. Final $\mathcal{H}_\infty$ norm for different faulty models.

γ_M	0	0.1	0.2	0.4	0.6	0.8	0.9	1
Nominal	1.19	1.43	1.80	3.71	∞	∞	∞	∞
FTC	1.19	1.16	1.14	1.18	1.11	1.19	3.69	∞

3.3 Fault detection and diagnosis

Since the proposed fault-tolerant control system requires the magnitude of LVE to recalculate its gains, the fault estimation is also an important task. However it is worthy to note that the FDD design is not the main focus of the paper. First, we define the virtual control effectiveness as the possibility to generate the total thrust, roll, pitch and yaw moments of the vehicle. For example, the LVE of an hexacopter is related to its LAE by the following equation:

$$\begin{bmatrix}\gamma_T\\ \gamma_L\\ \gamma_M\\ \gamma_N\end{bmatrix} = \begin{bmatrix}1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6\\ 1/8 & 1/4 & 1/8 & 1/8 & 1/4 & 1/8\\ 1/4 & 0 & 1/4 & 1/4 & 0 & 1/4\\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6\end{bmatrix}\begin{bmatrix}\gamma_1\\ \vdots\\ \gamma_6\end{bmatrix} \tag{11}$$

where $\gamma_i \in [0,1]$ represents the LAE of the ith actuator: $\gamma_i = 0$ means that the ith actuator is completely healthy and $\gamma_i = 1$ means that it is completely faulty. The parameters $\gamma_{T/L/M/N}$ that are also in the interval $[0,1]$ denote respectively the losses of total thrust, roll, pitch and yaw moments. For a quadcopter in the plus "+" configuration, it would be:

$$\begin{bmatrix}\gamma_T\\ \gamma_L\\ \gamma_M\\ \gamma_N\end{bmatrix} = \begin{bmatrix}1/4 & 1/4 & 1/4 & 1/4\\ 0 & 1/2 & 0 & 1/2\\ 1/2 & 0 & 1/2 & 0\\ 1/4 & 1/4 & 1/4 & 1/4\end{bmatrix}\begin{bmatrix}\gamma_1\\ \vdots\\ \gamma_4\end{bmatrix} \tag{12}$$

Due to the symmetric configuration, the transformation matrix from LAE to LVE of an "×" quadcopter is $\frac{1}{4}\mathbf{1}_4$, with $\mathbf{1}_4$ denoting the 4×4 matrix of ones.

To achieve a successful control system reconfiguration, we use a simple FDD module to estimate the LAE levels as quickly as possible. The fault information can be easily calculated by comparing the individual motor command and the actual rotational speed provided by the encoders that are available on our experimental platform. In order to verify the robustness of the proposed control under a more realistic LAE and LVE identification from FDD, the two-stage Kalman filter is performed on a quadcopter as in (Zhang et al., 2013; Nguyen et al., 2017).

In addition, the LVE estimation could be performed by a separated FDD scheme instead of using (11) with the LAE levels. However it is out of the scope of the present paper.

3.4 Implantation on the nonlinear system

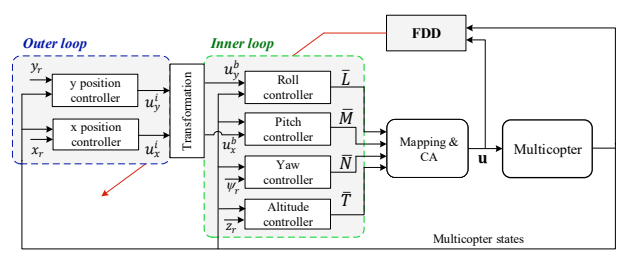


Fig. 5. Controller implementation.

After the synthesis, the universal controller is implemented on the nonlinear system where the current heading angle ψ has to be taken into account. Regarding the natural decoupling of the linearized system (8) and the subsequent feedback matrices, the inner and outer loops can be derived as depicted in Fig. 5. The inner loop focuses on the altitude and the attitude of the UAV while the outer loop focuses on its position in the $\mathbf{x}_i$-$\mathbf{y}_i$ plane. First, the altitude and yaw controller loops are directly deduced as:

$$\bar{T} = g + K_{i,z}\int(z_r - z)\,\mathrm{d}t - K_z z - K_{\dot{z}}\dot{z} \tag{13}$$

$$\bar{N} = K_{i,\psi}\int(\psi_r - \psi)\,\mathrm{d}t - K_\psi \psi - K_r r \tag{14}$$

Since multicopters are underactuated systems, x and y positions are controlled through combined pitch and roll angle demands. By design, the proposed controller only works for $\psi = 0$, and a change of coordinates must be operated to take into account the current value of ψ. To control UAV's position, the outer loop computes command signals in inertial frame $\mathcal{F}_i$:

$$\begin{aligned} u_x^i &= K_{i,x}\int(x_r - x)\,\mathrm{d}t - K_x x - K_{\dot{x}}\dot{x}\\ u_y^i &= K_{i,y}\int(y_r - y)\,\mathrm{d}t - K_y y - K_{\dot{y}}\dot{y}\end{aligned} \tag{15}$$

These signals are converted to body frame $\mathcal{F}_b$ by performing a rotation of angle ψ around $\mathbf{z}_i$-axis:

$$\begin{bmatrix}u_x^b\\ u_y^b\end{bmatrix} = \begin{bmatrix}\cos\psi & \sin\psi\\ -\sin\psi & \cos\psi\end{bmatrix}\begin{bmatrix}u_x^i\\ u_y^i\end{bmatrix} \tag{16}$$

The inner loop control laws which directly produce the normalized virtual control inputs are then :

$$\begin{aligned} \bar{L} &= u_y^b - K_\phi \phi - K_p p \\ \bar{M} &= u_x^b - K_\theta \theta - K_q q \end{aligned} \tag{17}$$

Finally, the mapping equation (7) and the classical control allocation law (5) are used to generate the thrust force to each motor. Moreover, the control signals are filtered by a first order low-pass filter with a time constant of 0.01 s to consider the unmodeled actuator dynamics.

4. MRAC AUGMENTATION DESIGN

This section introduces an adaptive strategy to address the system uncertainties. The design is based on a MRAC augmentation of the universal fault-tolerant controller developed in the previous section. The general architecture is depicted in Fig. 6.

Regarding the controller implementation (Fig. 5), only the inner loop is influenced by the system uncertainties, i.e, mass and inertia. In order to simplify the design without missing the current heading angle ψ in the implementation, the following plant is considered:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\Lambda\mathbf{v} + \mathbf{B}_r\mathbf{r} \tag{18}$$

where $\mathbf{x} = [z\ \dot{z}\ e_{zI}\ \phi\ p\ \theta\ q\ \psi\ r\ e_{\psi I}]^\top$ is the extended state vector with the integral of the tracking errors e_{zI} and $e_{\psi I}$, $\mathbf{v}$ is the same input vector as in (4), $\mathbf{\Lambda} \in \mathbb{R}^{4\times 4}$ is an unknown positive definite diagonal matrix, and $\mathbf{r} = \left[z_r\ u_y^b\ u_x^b\ \psi_r\right]^\top$ is considered as an external bounded command vector. The block matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{B}_r$ are known and can be obtained from Eqs. (6-8).[5] Regarding the UAV dynamic, the pair $(\mathbf{A}, \mathbf{B\Lambda})$ is controllable if and only if $\mathbf{\Lambda}$ has no zero diagonal element.

To restore the expected FTC tracking performance in the presence of the uncertain matrix Λ, the controller is augmented with an adaptive element $\mathbf{v}_{ad}$:

$$\mathbf{v} = \mathbf{v}_0 + \mathbf{v}_{ad} \tag{19}$$

where $\mathbf{v}_0 = \Upsilon_0\bar{\mathbf{v}} = -\Upsilon_0\mathbf{K}\mathbf{x}$ is the output of the state feedback fault-tolerant controller and Υ_0 represents the nominal mapping. Substituting (19) into (18) yields:

$$\dot{\mathbf{x}} = (\mathbf{A} - \mathbf{B}\mathbf{\Upsilon_0}\mathbf{K})\mathbf{x} + \mathbf{B}\mathbf{\Lambda}\left[\mathbf{v}_{ad} + \mathbf{\Theta}\mathbf{v}_0\right] + \mathbf{B}_r\mathbf{r} \tag{20}$$

where $\mathbf{\Theta} := \mathbf{I}_4 - \mathbf{\Lambda}^{-1}$, with $\mathbf{I}_4$ the 4×4 identity matrix.

Let consider the following reference model:

$$\dot{\mathbf{x}}_{ref} = \mathbf{A}_{ref}\mathbf{x}_{ref} + \mathbf{B}_r\mathbf{r} \tag{21}$$

where $\mathbf{A}_{ref} = (\mathbf{A} - \mathbf{B}\mathbf{\Upsilon_0}\mathbf{K})$ is Hurwitz by FTC design and $\mathbf{x}_{ref}(t)$ represents the system behavior in the nominal case $\mathbf{\Lambda} = \mathbf{I}_4$. The control goal is to force the system state $\mathbf{x}(t)$ to globally asymptotically track the state $\mathbf{x}_{ref}(t)$ of the reference model. In other words, the state tracking error $\mathbf{e}(t) = \mathbf{x}_{ref}(t) - \mathbf{x}(t)$ globally asymptotically tends to zero, as $t \to \infty$. Comparing (20) with the desired dynamic (21), the adaptive element $\mathbf{v}_{ad}$ is chosen as:

$$\mathbf{v}_{ad} = -\hat{\mathbf{\Theta}}\mathbf{v}_0 \tag{22}$$

where $\hat{\mathbf{\Theta}}$ is the estimation of the unknown matrix $\mathbf{\Theta}$. It can be seen that if the ideal adaptive gain $\hat{\mathbf{\Theta}} = \mathbf{\Theta}$ exists, the perfect model matching is hold, but it is not the case in practice due to the unknown parameter $\mathbf{\Lambda}$. The dynamics of the adaptive gain is then defined using the Lyapunov-based approach with the classical law:

$$\dot{\hat{\mathbf{\Theta}}} = \mathbf{\Gamma}\mathbf{v}_0\mathbf{e}^\top\mathbf{P}\mathbf{B} \tag{23}$$

where $\mathbf{\Gamma}$, $\mathbf{P}$ and $\mathbf{Q}$ are symmetric positive definite matrices of appropriate dimension. Furthermore, $\mathbf{\Gamma}$ is diagonal and represents the rate of adaptation. $\mathbf{P}$ is the unique solution of the algebraic Lyapunov equation $\mathbf{P}\mathbf{A}_{ref} + \mathbf{A}_{ref}^\top\mathbf{P} = -\mathbf{Q}$ for an adequately chosen $\mathbf{Q} = \mathbf{Q}^\top > 0$.

In order to prove the stability of the overall system, consider the tracking error dynamic obtained by subtracting (20) from (21):

$$\dot{\mathbf{e}} = \mathbf{A}_{ref}\mathbf{e} - \mathbf{B}\mathbf{\Lambda}\mathbf{\Delta\Theta}\mathbf{v}_0 \tag{24}$$

where $\mathbf{\Delta\Theta} := \mathbf{\Theta} - \hat{\mathbf{\Theta}}$ is the parameter estimation error. Consider the following radially unbounded Lyapunov candidate function:

$$V = \mathbf{e}^\top\mathbf{P}\mathbf{e} + \text{trace}\left(\mathbf{\Delta\Theta}\mathbf{\Gamma}^{-1}\mathbf{\Delta\Theta}^\top\mathbf{\Lambda}\right)$$

The time derivative of V, evaluated along the trajectory of the error model (24), is given by:

$$\dot{V} = -\mathbf{e}^\top\mathbf{Q}\mathbf{e} + 2\,\text{trace}\left[\mathbf{\Delta\Theta}\left(\mathbf{\Gamma}^{-1}\dot{\hat{\mathbf{\Theta}}} - \mathbf{v}_0\mathbf{e}^\top\mathbf{P}\mathbf{B}\right)\mathbf{\Lambda}\right]$$

It can be proved that the adaptive law (23) leads to $\dot{V} \leq 0$. By using Barbalat's lemma, the closed-loop error dynamic is globally asymptotically stable (Lavretsky and Wise, 2012). As a results, the tracking error asymptotically tends to zero, i.e., $\lim_{t\to\infty}\mathbf{e}(t) = 0$.

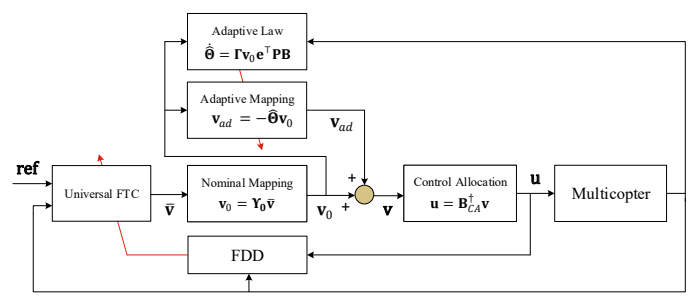


Fig. 6. Overview of the controller architecture.

Remark 1. The reference model can be simply chosen as $\mathbf{A}_{ref} = (\mathbf{A} - \mathbf{B}\mathbf{\Upsilon_0}\mathbf{K_0})$ where $\mathbf{K}_0$ is the scheduled gain value in the fault-free situation. This choice makes sure that the reference model is not changed during the flight.

Remark 2. In order to adapt the design requirements for specific multicopter UAVs, an appropriate reference model $\mathbf{A}_{ref}$ can be chosen by designers. In this case, its diagonal block structure has to be respected.

Remark 3. The unknown matrix $\mathbf{\Lambda}$ can represent the losses of control effectiveness. It means that the proposed adaptive algorithm also has the ability of a passive fault-tolerant controller. However, this aspect is not the main focus of the present paper.

5. RESULTS

Simulation and flight experiment results are conducted to verify the effectiveness of the proposed scheme. First, to demonstrate the portability of the proposed FTC system, the controller is implemented in two different platforms: a small quadcopter and a medium hexacopter, whose

[5] In this section, the deviation notation Δ is omitted for brevity.

physical parameters including actuator constraints can be found in (Nguyen et al., 2017, 2018). Table 4 summarizes the test cases considered in this work. Results of both hovering and trajectory tracking performances subject to actuator faults including total failures of the rotors (case 2) and system uncertainties (case 3) are presented.

Table 4. Test scenarios

Case	Multicopter	Type	Faults/Uncertainties
1	Hexacopter NPNPNP	Experiment Hovering	$\gamma_1 = 0.2, \gamma_2 = 0.9$
2	Hexacopter NNPPNP	Simulation Hovering	$\gamma_1 = \gamma_3 = 1$
3	Quadcopter	Simulation Tracking	$\gamma_1 = 0.35$ $m = 1.5m_0, \mathbf{I}_b = 1.5\mathbf{I}_{b,0}$

Furthermore, the same adaptive parameters and scheduled gains are utilized for both multicopters. Using the simplified model and design requirements gathered in Tab. 1, the synthesis yields the following scheduled feedback gains:

$$\mathbf{K}_x = [30.0\ 17.7\ 56.3\ 9.9] + [11.4\ 17.4\ 54.8\ 12.2]\,\gamma_M + [-6.6\ -5.3\ 46.3\ 22.9]\,\gamma_M^2$$
$$\mathbf{K}_y = [-27.6\ -15.4\ 51\ 9.3] + [-80.8\ -50.4\ 156\ 29]\,\gamma_L$$
$$\mathbf{K}_z = [9.6\ 4.4] + [7.8\ 8.2]\,\gamma_T$$
$$\mathbf{K}_\psi = [5.1\ 3.2] + [7.1\ 5.7]\,\gamma_N$$

and scheduled integral gains:

$$K_{i,x} = 24.5 - 3\gamma_M - 3.3\gamma_M^2 \qquad K_{i,y} = -21.4 - 61.6\gamma_L$$
$$K_{i,z} = 8 + 2.8\gamma_T \qquad K_{i,\psi} = 3.2 + 4.5\gamma_N$$

The adaptive augmentation algorithm is designed using:

$$\mathbf{\Gamma} = \mathrm{diag}\,(10, 1, 1, 10)$$
$$\mathbf{Q} = \mathrm{diag}\,(\mathbf{I}_3, 0.5\mathbf{I}_2, 0.5\mathbf{I}_2, 5\mathbf{I}_3)$$

Note that the evolution of x-pitch controller gains versus the LVE (Fig. 3) is almost linear. Therefore, we chose affine functions instead of quadratic functions in the design of y-roll, z and ψ controllers, for simplicity.

5.1 Case 1

The proposed FTC system is implemented in the *Asctec Firefly* hexacopter from Ascending Technologies. The overall system setup can be found in (Nguyen et al., 2018). Here, the loss of actuator effectiveness is artificially injected by introducing a weight on the individual motor speed commands. A total rotor failure is not appropriate to operate on our hexacopter due to its traditional NPNPNP rotor arrangement. Indeed, the hexacopter is immediately uncontrollable if one motor fails. On the other hand, the test case with two rotor degradation levels of 90% and 20% can be considered as an approximate test scenario. A limit has been imposed on the rotor rotational speeds, i.e., $\omega_i \leq 10000\,\mathrm{rpm}$ leading to the thrust limit, $T_i \leq 9.37\,\mathrm{N}$. The moment inputs are limited as follows: $-2\,\mathrm{N{\cdot}m} \leq L, M \leq 2\,\mathrm{N{\cdot}m}$ and $-0.2\,\mathrm{N{\cdot}m} \leq N \leq 0.2\,\mathrm{N{\cdot}m}$.

As shown in Fig. 7, the LAE/LVE factors are precisely calculated using measured motor speeds. Due to sensor noises, some small data jumps are also observed.

Figure 8 provides the evolution of the proportional gains versus the LVE value during flight test.

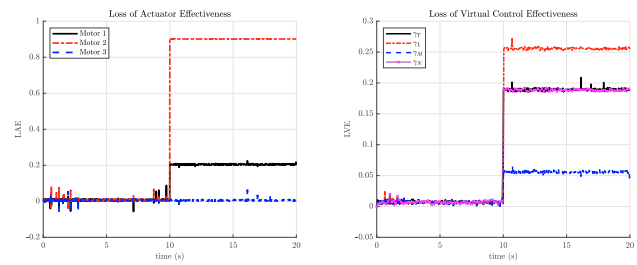


Fig. 7. Case 1: Calculated LAE and LVE.

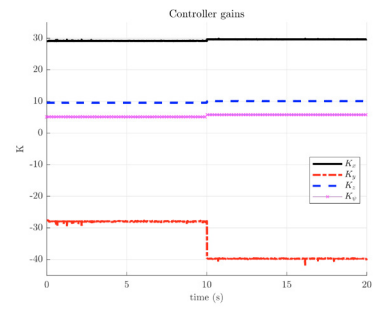


Fig. 8. Case 1: Evolution of controller gains.

Using the proposed reconfiguration, the controller can rapidly regulate the thrust of each motor as illustrated in Fig. 9. We can also observe that steady-state motor's speeds before the fault are not similar due to the fact that the hexacopter is not perfectly symmetric.

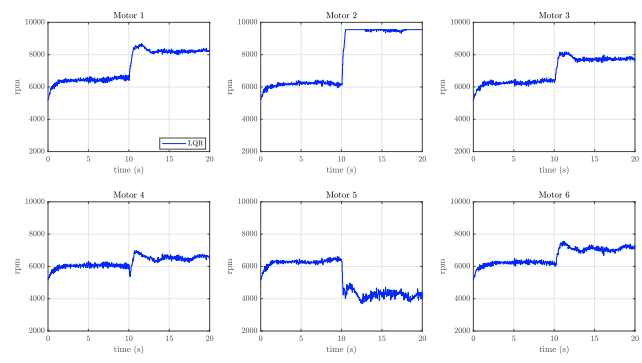


Fig. 9. Case 1: Motor commands.

As a result, the UAV can return to its hover position (at an altitude of 1m) after the fault occurs as shown in Fig. 10. The result is better than expected, since the nominal controller (without reconfiguration) failed in maintaining the system stability. The nearest faulty situation that it can handle is about 20% and 65% faults in the two first motors. It can be noticed that the instability comes from the biggest loss of roll moment (25%). In this case, a significant change of the proposed $y-$roll controller gains (Fig. 8) would be able to generate an appropriate torque acting on the UAV to maintain its stability.

5.2 Case 2

In the second case, we use RotorS [6] to simulate an AscTec Firefly hexacopter testbed on Gazebo using Robot Operat-

[6] https://github.com/ethz-asl/rotors_simulator

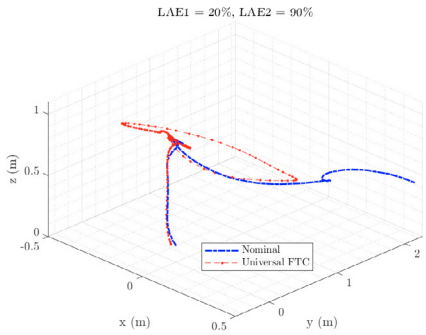
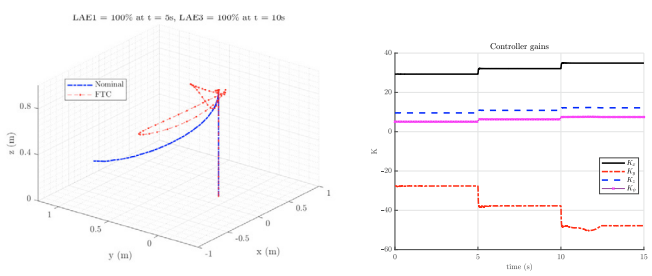


Fig. 10. Case 1: NPNPNP Hexacopter stability improvement with LAE1 20% and LAE2 90%.

ing System (ROS). The goal is to evaluate the performance of the controller in a more challenging situation where two complete losses of the first and the third rotor are introduced at $t = 5s$ and $t = 10s$ respectively. The gain profile and system response are given in Fig. 11.



(a) Fault recovery (b) Controller gains

Fig. 11. Case 2: NNPPNP Hexacopter stability improvement with two successive rotor failures.

It can be seen that the nominal controller failed in maintaining the system stability while the proposed controller was still working well with an appropriate reconfiguration.

5.3 Case 3

Finally, the control algorithm is required to track the following trajectory:

$$[x_r \;\; y_r \;\; z_r \;\; \psi_r]^\top = [\cos(t) \;\; \sin(t) \;\; 1.5 \;\; 0]^\top$$

To evaluate as completely as possible the performance of the proposed FTC system, a loss of 35% control effectiveness was injected to the first motor of a small size quadcopter with unknown mass and inertia (Tab. 4).

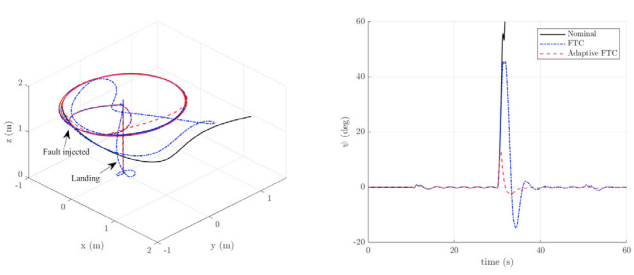


Fig. 12. Case 3: Quadcopter tracking performance with unknown mass, inertia and actuator fault.

As displayed in Fig. 12, the nominal controller with simple integral actions can not handle the fault while the universal FTC keeps the drone tracking the desired trajectory. In addition, the system performance is considerably improved by the adaptive augmentation. Its efficiency is shown with smaller heading angle, faster fault recovery as well as better landing curve.

6. CONCLUSION

This paper introduced a universal adaptive fault tolerance strategy subject to actuator faults and system uncertainties, which is applicable to most traditional multicopters. Several simulation and experimental tests were successfully performed on different UAVs using the same controller with constant parameters. High performance of the proposed scheme was shown in both fault recovery capability and adaptation. Future work will focus on the LVE estimation as mentioned in 3.3 and experiment tests on other multicopters.

REFERENCES

Falconi, G.P. and Holzapfel, F. (2016). Adaptive fault tolerant control allocation for a hexacopter system. In *American Control Conference (ACC)*.

Gahinet, P. and Apkarian, P. (2011). Structured $\mathcal{H}_\infty$ synthesis in MATLAB. *IFAC Proceedings Volumes*, 44(1), 1435 – 1440.

Lavretsky, E. and Wise, K. (2012). *Robust and Adaptive Control: With Aerospace Applications*. Springer London.

Lhachemi, H., Saussie, D., and Zhu, G. (2014). A robust and self-scheduled longitudinal flight control system: a multi-model and structured $\mathcal{H}_\infty$ approach. In *AIAA Guidance, Navigation, and Control Conference*.

Mallavalli, S. and Fekih, A. (2018). A fault tolerant tracking control for a quadrotor UAV subject to simultaneous actuator faults and exogenous disturbances. *International Journal of Control*, 0(0), 1–14.

Nguyen, D.T., Saussie, D., and Saydy, L. (2018). Fault-tolerant control of a hexacopter UAV based on self-scheduled control allocation. In *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, 385–393.

Nguyen, D.T., Saussié, D., and Saydy, L. (2017). Robust self-scheduled fault-tolerant control of a quadrotor UAV. In *20th World Congress of the International Federation of Automatic Control (IFAC)*.

Rugh, W.J. and Shamma, J.S. (2000). Research on gain scheduling. *Automatica*, 36(10), 1401 – 1425.

Yoon, H.J., Cichella, V., and Hovakimyanx, N. (2016). Robust adaptive control allocation for an octocopter under actuator faults. In *AIAA Guidance, Navigation, and Control Conference*.

Zhang, Y.M., Chamseddine, A., Rabbath, C.A., Gordon, B.W., Su, C.Y., Rakheja, S., Fulford, C., Apkarian, J., and Gosselin, P. (2013). Development of advanced FDD and FTC techniques with application to an unmanned quadrotor helicopter testbed. *Journal of the Franklin Institute*, 350(9), 2396–2422.

Zhang, Y.M. and Jiang, J. (2008). Bibliographical review on reconfigurable fault-tolerant control systems. *Annual Reviews in Control*, 32(2), 229–252.