



Comparative study of self tuning, adaptive and multiplexing FTC strategies for successive failures in an Octorotor UAV

Hussein Hamadi, Benjamin Lussier, Isabelle Fantoni, Clovis Francis, Hassan Shraim

► To cite this version:

Hussein Hamadi, Benjamin Lussier, Isabelle Fantoni, Clovis Francis, Hassan Shraim. Comparative study of self tuning, adaptive and multiplexing FTC strategies for successive failures in an Octorotor UAV. *Robotics and Autonomous Systems*, 2020, 133, pp.103602. 10.1016/j.robot.2020.103602 . hal-02939565

HAL Id: hal-02939565

<https://hal.science/hal-02939565>

Submitted on 15 Sep 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Comparative study of self tuning, adaptive and multiplexing FTC strategies for successive failures in an Octorotor UAV

Hussein Hamadi^{a,b,*}, Benjamin Lussier^a, Isabelle Fantoni^{a,c}, Clovis Francis^b, Hassan Shraim^b

^a*Sorbonne Universités, Université de Technologie de Compiègne, CNRS, UMR 7253 Heudiasyc, 60200 Compiègne, France*

^b*Université Libanaise, Faculté de Génie, Centre de Recherche Scientifique en Ingénierie (CRSI), Liban*

^c*Laboratoire des Sciences du Numérique de Nantes (LS2N), UMR CNRS 6004, 1 rue de la Noë, 44321 Nantes, France*

Abstract

This paper presents three fault-tolerant control (FTC) strategies for a coaxial octorotor unmanned aerial vehicle (UAV) regarding motor failures. The first FTC is based on a control mixing strategy which consists of a set of control laws designed offline, each one dedicated to a specific fault situation. The second FTC, a robust adaptive sliding mode control allocation is presented, where the control gains of the controller are adjusted online in order to redistribute the control signals among the healthy motors in order to stabilize the overall system. The third FTC strategy is a new strategy proposed in this article, which is based on a self-tuning sliding mode control (STSMC) where the control gains are readjusted based on the detected error to maintain the stability of the system. Multiple indoor experiments on an octorotor UAV are conducted to show and compare the effectiveness and the behavior of each FTC scheme after successive faults are injected. More specifically, we inject complete actuator's failures into the top four motors of our octorotor. Every strategies show good fault tolerance results, although the control mixing method performs slightly better overall while the adaptive method performs slightly worse. However, the control mixing method requires a huge design effort to take into account as much situations as possible, while the adaptive method and the STSMC only require to determine a few gains. The adaptive method do not need fault detection to operate, but it thus does not provide information on the system's health without an additional fault identification and diagnosis mechanism, while both the control mixing method and the STSMC provide such information.

Keywords: fault-tolerant control, sliding mode control, robust control, multiplexing, control allocation, actuator redundancy, adaptive control, UAV

1. Introduction

Nowadays, several applications fields rely on the use of drones or UAVs. As they are small, light, and maneuverable, drones have many uses in both military and civilian domains, such as fire detection, inspection of power lines, flying surveillance, aerial photography, roof and solar panels inspection, 3d mapping, etc. In order to accomplish these missions safely, UAVs must be equipped with an onboard computer or autopilot, different sensors (GPS, IMU, magnetometer, barometer...) and monitoring systems to ensure the full functionality of the drone.

From a dependability point of view, UAVs are primarily concerned with the safety and reliability attributes, which are respectively the absence of catastrophic consequences on the users and the environment, and the deliverance of correct service [1]. However, these concerns can be contradictory, since in some critical situations safety concerns would require to safely stop the UAV while reliability concerns would require to continue the mission. In this paper, we focus on the safety attribute, which needs fault tolerance mechanisms to cope with possible faults and failures of actuators during flights, and to preserve the performance of the system without imperiling itself and its surroundings.

A motor failure can be described as a partial or total loss of actuator effectiveness. A motor failure can be first considered as an error in the system until it has undesired consequences on the system's behavior, which will usually happen quickly. Fault-tolerant control (FTC) strategies for UAVs have recently received significant attention in research fields due to the increasing awareness about the risks resulting from components failures, and the need for reliable and safe systems in critical applications [2].

Among different types of multirotor UAVs, the quadrotor has been widely used by the academic communities due to its simple design and dynamics. This allows to easily develop and test several control algorithms [3]. In the literature, different strategies were developed considering the FTC problem against partial actuator failures of

*Corresponding author

Email address: hussein.hamadi@hds.utc.fr (Hussein Hamadi)

a quadrotor. In [4], a reconfigurable FTC is proposed, based on a trajectory re-planning scheme and an online decision making strategy using differential flatness. This strategy consists in synthesizing a reconfigurable feedback control with a modified reference trajectory once an actuator fault has been diagnosed by a fault detection and diagnosis scheme, which uses a parameter estimation based unscented Kalman filter. In [5], the authors proposed a nonlinear adaptive feedback linearization strategy where the solution takes into account the management of the control authority by incorporating the post-fault model of the actuator. This guarantees an acceptable performance in presence of certain types of faults in the actuators. Other proposed strategies addressing partial actuator failures can be found in [6], [7] and [8], where the authors proposed robust and adaptive control laws based on sliding mode and backstepping theories to compensate the effectiveness loss in actuators.

A complete loss of a motor/propeller of the quadrotor leads to the inability to fully control the system's attitude [9]. Due to this difficulty, there are few works considering the case of complete rotor loss: [10], [11], [12]. The most common solution is to sacrifice the controllability of the yaw angle to still be able to command the UAV's position and altitude by making it continuously rotate around its Z axis.

In order to maintain complete control of the UAV in the case of one or more motor failures, one solution is to consider multirotors with redundant actuators, i.e., hexarotors [13] and octorotors [14]. The main advantage of such configurations is the possibility of tolerating multiple failures without losing complete controllability. It thus allows the UAV to keep a stable flight and possibly continue its mission. In the present paper, we will consider a coaxial octorotor configuration to compare three FTC recovery algorithms in real experiments.

The first FTC is based on a control mixing strategy which consists in a set of control laws designed offline, each one is dedicated to a specific fault situation. The second FTC consists in a robust adaptive sliding mode control allocation (ASMCA), where the control gains of the controller are adjusted online in order to redistribute the control signals among the healthy motors in order to stabilize the overall system. The third FTC strategy is a new strategy proposed in this article, which is based on a self-tuning sliding mode control (STSMC) where the control gains are readjusted depending on the detected error to maintain the stability of the system. The fault tolerance characteristics and aspects of the three FTC schemes are analyzed and compared after conducting a set of 60 experiments both in hovering and trajectory tracking modes. During these experiments, we inject complete actuator's failures into the top four motors of our octorotor. Based on our experimental results, the three FTC strategies show good fault tolerance results. The control mixing method outperforms slightly the two other strategies while the adaptive method performs slightly worse. However, the control mixing method requires a huge design effort to take into account as much different faults as possible, while the adaptive method and the STSMC only require to determine few gains. In the adaptive method, no fault detection module is required and thus it does not provide information on the system's health, while both the control mixing and the STSMC methods provide such information.

This paper is organized as follows: Section 2 presents an overview on fault tolerant control schemes. Related works are given in Section 3. In Section 4, we present the nonlinear equations of motion of the coaxial octorotor, and the design of three different FTC schemes are developed in Section 5 and 6. The experimental results are followed in Section 7, to test the effectiveness of each FTC recovery strategy with an analysis of the corresponding behavior. The obtained results are discussed in Section 8. Finally, some conclusions and perspectives are summarized in Section 9.

2. Fault-Tolerant control classification

In the control system's literature, fault-tolerance control systems (FTC) are divided into two categories, namely active and passive FTCs.

In active fault-tolerant control (AFTC) strategies, the fault tolerance process is carried out via three successive steps: (1) Error detection, using a fault detection FD unit, (2) fault diagnosis, which identifies the occurrence and the type of a fault after the detection, (3) system recovery. In the AFTC approach, it is required to have a fault detection and identification (FDI) process. This FDI detects and localizes the faults occurring in the system. Then, the system recovery is done through redundancies in the system, using projection-based methods or online redesign/adaptation methods. In projection based methods, a set of new pre-computed control laws are predefined offline, and associated with a fault pattern. When the FDI diagnoses and isolates a fault, a corresponding control law is selected from the set to deal with this particular situation. The drawback of this approach is that it is restricted to the finite number of faults that have been considered offline. In online based redesign/adaptation methods, control parameters or even the whole structure of the controller are reconfigured online using computation algorithms to guarantee the system stability in faulty situations. In both cases, since we are monitoring the system health, we can know if the system is getting close to its limits (particularly in term of redundancy), and putting the system in a safe state can be realized if the system is deemed not safe enough (such as an emergency landing or an order to return to base). The main drawback of AFTCs is that the fault is not tolerated during the time needed to detect and isolate it, possibly leading to a system's failure if the detection takes too much time.

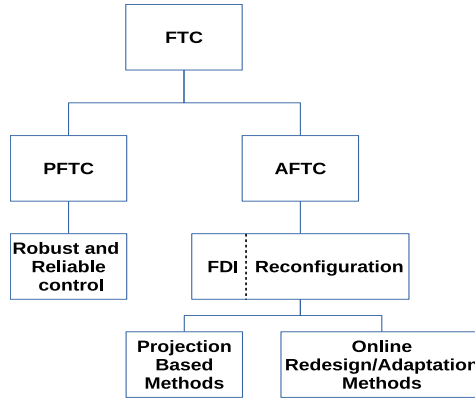


Figure 1: Classification of FTCs

In contrast to AFTCs, passive fault-tolerant control (PFTC) strategies consists in masking the faults for example by using a robust and reliable controller able to deal with all expected fault without the need to detect and identify them. As a good example, an adaptive sliding mode control allocation (ASMCA) is proposed in [15] to automatically redistribute the control signals to the fault-free motors. The main advantage of using PFTCs is that it can avoid the time delay needed to detect the error since it is not required. However, in this case, we don't monitor the system health, and this could lead eventually to a failure when the system reaches its fault-tolerance limits. Moreover, adaptation to failing situations from robustness is generally not as efficient to recover from a particular failure as a specifically designed recovery mechanism. A classification of FTCs is presented in Fig. 1, and a more detailed description is given in [16].

3. State of the art on fault-tolerant control for UAVs through actuators redundancy

We present in this section previous work relevant to our topic: that is FTC strategies that allow the system to remain fully controllable. The added hardware redundancy in coaxial octorotors, though allowing more dependability in the system, complexifies the definition of the control law as it makes the system over-actuated. A full analysis considering different actuators configurations, and fault tolerance capabilities regarding failures is given in [17]. The actuators redundancy is considered as an important factor in any FTC strategy implementation, since it allows to maintain complete controllability of the system not only with partial faults, but also with up to four total motors failures. However, an effective motor control allocation of the remaining healthy motors is required to achieve acceptable performance [18].

In [19], a time delay fault tolerant control (TDC) is proposed to maintain attitude stabilization after a fault on one or more rotors by treating the fault as a disturbance. Another work [20] suggests a fault-tolerant control strategy using cooperation between a radial based function neural network, fuzzy logic control and sliding mode control (SMC) technique in presence of actuator faults, to alleviate the chattering and to maintain good tracking of the system. In [21], two FTC schemes using linear parameter varying system representation, with a combination of SMC theory and control allocation, are developed and tested in the presence of uncertainty, as well as faults and failures. In the first scheme, the knowledge of the rotor effectiveness is required in order to apply an online control allocation methodology, and to redistribute the control signal to working motors, but in the second scheme this knowledge is not necessary. In [22], a neural network, an interval type-2 fuzzy logic control approach and a sliding mode control technique are used to design a controller, named fault tolerant neural network interval type-2 fuzzy sliding mode controller. This control scheme has many advantages, since it allows avoiding difficult modeling, attenuating the chattering effect of the SMC, reducing the number of rules for the fuzzy controller, and guaranteeing the stability and the robustness of the system.

Another fault tolerant control scheme based on nonlinear model predictive control for a Y6 coaxial tricopter is proposed in [23]. In this study, a cascaded closed-loop control methodology is proposed which incorporates a reconfigurable low-level controller. The effectiveness of the presented fault-tolerant scheme is validated by following an 8-shaped trajectory with a complete loss of one rotor. In [24], the authors proved that a passive FTC controller based on a second order sliding mode control outperforms an active FTC controller designed using a pseudo-inverse dynamic control re-allocation and a first order sliding mode control.

A global active FTC of underactuated UAV with redundant actuators (hexarotors and octorotors) is proposed in [25] where the entire architecture contains a baseline controller (adopted from [26]), and a fault detection and

isolation scheme based on a robust parameter identification approach. The FDI identifies the generated thrust and torques by the actuators in normal and faulty situations and a controller reconstruction module then calculates a feasible solution to the control allocation problem in faulty cases. In [27], a fault-tolerant control for an octorotor UAV is proposed based on a combination of Backstepping and Dynamic Surface Control. In this method, the uncertainties and parameters variations caused by faults and failures are estimated and then taken into account in the control allocation algorithm, based on a Moore-Penrose pseudoinverse [28], which distributes the control efforts among the actuators and minimizes the energy consumption. Another FTC strategy based on the reconfiguration of the remaining rotors was proposed in [29], to handle the disturbance torque and the large yaw rate which occurs after the failure of a one rotor in a hexarotor UAV. The main issue in all these different studies is that only simulations are presented to validate the effectiveness of the proposed methods.

In contrast, a few studies have addressed the problem of fault-tolerant control for actuators faults in redundant UAVs providing experimental validations for the proposed strategies, such as in [30], where the authors proposed a fault detection and a fault-tolerant control FTC scheme which can handle up to two actuators failure for an hexarotor. This FTC scheme uses a nonlinear Thau observer to estimate the states of the UAV and to detect the actuators failures. It also uses a sliding mode and disturbance observer to stabilize the UAV despite the existence of disturbances. However, the authors cited that for the frames that have more than six motors (like octorotor), the fault detection and fault-tolerant control schemes are more complex, which is out of scope of their research and will be investigated in further studies.

In our opinion, the most important studies related to our work are given in [31] and [15]. In [31], an offline control mixing fault-tolerant control strategy is detailed, which consists in computing a set of explicit laws, where each one is dedicated to a fault situation and obtained by solving an optimization problem considering this particular each fault situation. Fault detection and isolation must be associated with the FTC scheme to match a corresponding control mixing law. From this detection, a lookup table containing all possible faults and failures combinations is used to select the control law to apply for the system recovery. In [15], an online ASMCA scheme is proposed as a passive robustness approach to adapt to motors failures. The stability of the overall system is verified using the Lyapunov theory. However, the convergence time may vary depending on the gain tuning process and the initial condition of the system.

In this paper, as a new FTC algorithm, a self-tuning sliding mode control (STSMC) FTC method is proposed to deal with motors failures. An online reconfigurable sliding mode control allocation is developed, where the control gain's reconfiguration process is automatically done depending on a FDI unit. This FDI unit is also based on a sliding mode observer linked with an updated model for the fault detection and identification process. We will detail and compare these last three different techniques in the rest of this paper.

4. Octorotor Dynamics

In this section, the dynamic model of the coaxial octorotor (Fig. 2) is presented. This configuration was developed in [32] and will be used to test the FTC schemes, since it has many advantages over the star-shaped configuration [21] regarding stability and size factors. First, for design purposes, some assumptions are made:

- The UAV frame is considered rigid and symmetric, thus the inertia matrix is diagonal.
- The reference trajectory controllers are continuous, differentiable and their derivatives are bounded.
- The vehicle attitudes (roll, pitch and yaw angles) are constrained to $(-\pi/2 < \phi, \theta < \pi/2)$ and $(-\pi \leq \psi \leq \pi)$. This means that the UAV can not follow acrobatic trajectories like looping or flying upside-down.
- All the control inputs are bounded, i.e., $u_f \in]0, 1]$ and $\tau_\phi, \tau_\theta, \tau_\psi \in [-1; 1]$
- The system dynamics are limited to small angles and small variations of linear and angular velocities. This behavior is common in practice with UAV

In order to obtain the nonlinear equations governing the motion of the coaxial octorotor, we need first to consider two frames: R_e and R_b , where R_e is the fixed earth frame $\{X_e, Y_e, Z_e\}$ and R_b is the body frame $\{X_b, Y_b, Z_b\}$ with its origin at the center of mass of the octorotor.

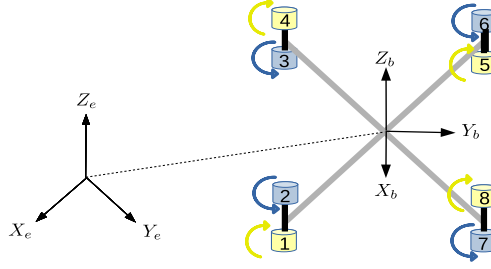


Figure 2: Coaxial octorotor configuration

Following the Newton-Euler approach, the full mathematical model of the vehicle dynamics can be obtained as follows [15]:

$$\begin{cases} m\ddot{x} &= (\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi)u_f \\ m\ddot{y} &= (\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi)u_f \\ m\ddot{z} &= (\cos \phi \cos \theta)u_f - mG \\ I_{xx}\dot{p} &= (I_{yy} - I_{zz})qr + \tau_\phi \\ I_{yy}\dot{q} &= (I_{zz} - I_{xx})pr + \tau_\theta \\ I_{zz}\dot{r} &= (I_{xx} - I_{yy})pq + \tau_\psi \end{cases} \quad (1)$$

where x , y and z denote the position in the fixed earth frame, and ϕ , θ and ψ the Euler angles of the vehicle. p , q and r are the angular rates expressed in the body frame. I_{xx} , I_{yy} and I_{zz} represent the inertia values of the vehicle, G represents the gravitational constant and finally the virtual input vector \mathbf{u} is defined as:

$$\mathbf{u} = [u_f \ \tau_\phi \ \tau_\theta \ \tau_\psi]^T \quad (2)$$

with u_f , τ_ϕ , τ_θ and τ_ψ representing respectively the total thrust and the three torques of the system (roll, pitch and yaw).

The virtual inputs of the system are related to the force and torque of each motor by the following expressions:

$$\begin{aligned} u_f &= F_{12} + F_{34} + F_{56} + F_{78} \\ \tau_\phi &= \frac{\sqrt{2}}{2}d(F_{78} + F_{56} - F_{34} - F_{12}) \\ \tau_\theta &= \frac{\sqrt{2}}{2}d(F_{34} + F_{56} - F_{78} - F_{12}) \\ \tau_\psi &= (\tau_2 + \tau_3 + \tau_6 + \tau_7) - (\tau_1 + \tau_4 + \tau_5 + \tau_8) \end{aligned} \quad (3)$$

where d is the length of the arm, and the force F_i produced by each motor is proportional to the square of the angular speed :

$$\begin{aligned} F_i &= K_f \omega_i^2 \\ \tau_i &= K_t \omega_i^2 \\ i &= 1, \dots, 8 \end{aligned} \quad (4)$$

where K_f and K_t represent respectively the thrust and drag coefficients of the actuators. The force produced by each pair of coaxial motors i and j is given by [32]:

$$F_{ij} = \alpha_{ij}(F_i + F_j)(1 + \frac{S_s}{S_{prop}}) \quad (5)$$

¹⁶⁵ S_s and S_{prop} denote respectively the propeller's surface and the area of the circle covered by the motion of the propeller. Thus we can define $S = (1 + \frac{S_s}{S_{prop}})$ as the shape factor of the propellers. Due to the aerodynamic interference between each coaxial pair, the total aerodynamic efficiency is decreased. This loss of aerodynamic efficiency is represented by the constant α_{ij} .

\mathbf{T} is the transformation matrix from the body angular rates and the rates of the Euler angles:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \mathbf{T} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (6)$$

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \cos \theta \\ 0 & -\sin \phi & \cos \phi \cos \theta \end{bmatrix} \quad (7)$$

For controller design purposes, it is worth to note that for small changes of roll and pitch angles, \mathbf{T} is very close to the identity matrix, thus the rates of the Euler angles can replace the body angular rates in equation (1), and the system dynamics become:

$$\begin{cases} m\ddot{x} &= (\cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi)u_f \\ m\ddot{y} &= (\cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi)u_f \\ m\ddot{z} &= (\cos\phi \cos\theta)u_f - mG \\ I_{xx}\ddot{\phi} &= (I_{yy} - I_{zz})\dot{\theta}\dot{\psi} + \tau_\phi \\ I_{yy}\ddot{\theta} &= (I_{zz} - I_{xx})\dot{\phi}\dot{\psi} + \tau_\theta \\ I_{zz}\ddot{\psi} &= (I_{xx} - I_{yy})\dot{\theta}\dot{\phi} + \tau_\psi \end{cases} \quad (8)$$

5. Problem formulation

In this section, we present four aspects of the control law common to the three FTC schemes presented in section 6. The first point is the modelization of the control inputs and the motors failures. The second point is the definition of the control allocation problem. The third point is the determination of subsystems for the control laws. Finally, the fourth point is a common sliding manifold for the SMC of the three FTC.

5.1. Modelization of the control inputs and the motors health

First, let us consider the general nonlinear system model:

$$\begin{aligned} \dot{\mathbf{X}} &= \mathbf{f}(\mathbf{X}) + \mathbf{g}(\mathbf{X})\mathbf{u} \\ \mathbf{u} &= \mathbf{B}\mathbf{L}\mathbf{u}^* \end{aligned} \quad (9)$$

where \mathbf{u} is the virtual control input, that is the desired thrust and the moments, \mathbf{u}^* represents the real control input, that is the each motor's desired power, and $\mathbf{X} \in \mathbb{R}^n$ is the state vector of the system, defined in the next section. The nonlinear functions $\mathbf{f}(\mathbf{X}) \in \mathbb{R}^n$ and $\mathbf{g}(\mathbf{X}) \in \mathbb{R}^n$ are respectively the drift vector field and the control vector field. $\mathbf{L} = \text{diag}(l_1, \dots, l_m)$ with m as the number of actuators and $0 \leq l_i \leq 1$ representing the control effectiveness of the actuators, so if $l_i = 1$, the i -th actuators is working perfectly, whereas $l_i = 0$ means the complete failure of the i -th actuators, and $0 < l_i < 1$ denotes a partial failure on the i -th actuator. $\mathbf{B} \in \mathbb{R}^{p \times m}$ with $p < m$ is the control effectiveness matrix, representing the relation between the virtual and real control inputs. Moreover, as \mathbf{u} is bounded from our fourth assumption (Section 4), we can say from (9) that \mathbf{u}^* belongs to a compact set Σ defined as:

$$\Sigma = \{\mathbf{u}^* \in \mathbb{R}^m \mid u_{imin}^* \leq u_i^* \leq u_{imax}^* \mid i = 1, \dots, m\} \quad (10)$$

5.2. Control allocation problem

In the coaxial octorotor configuration, the UAV is equipped with eight actuators where each pair is on the same arm and has the same axle, but rotates in opposite directions (four motors rotating clockwise, four motors rotating counter-clockwise). Usually, FTC schemes take advantage of the added hardware redundancy in order to achieve fast responses when a motor failure occurs. During fault-free operation, the high level controller generates virtual control inputs $\mathbf{u} = [u_f \quad \tau_\phi \quad \tau_\theta \quad \tau_\psi]^T$. This virtual input vector \mathbf{u} is redistributed among the set of m healthy motors where $m \leq 8$. This redistribution is known as the control allocation. The control allocation problem is formulated as:

$$\begin{aligned} \mathbf{B}\mathbf{L}\mathbf{u}^*(t) &= \mathbf{u}(t) \\ u_{imin}^* &\leq u_i^* \leq u_{imax}^* \end{aligned} \quad (11)$$

where $\mathbf{u}^* = [\omega_1^2 \quad \omega_2^2 \quad \dots \quad \omega_8^2]^T$, and the control effectiveness matrix \mathbf{B} is defined as:

$$\mathbf{B} = \begin{bmatrix} t_1 & \dots & t_8 \\ r_1 & \dots & r_8 \\ p_1 & \dots & p_8 \\ y_1 & \dots & y_8 \end{bmatrix} \quad (12)$$

with:

$$\begin{aligned}
t_i &= \alpha_{ij} K_f S \\
r_i &= \pm \alpha_{ij} K_f S d \frac{\sqrt{2}}{2} \\
p_i &= \pm \alpha_{ij} K_f S d \frac{\sqrt{2}}{2} \\
y_i &= \pm K_t
\end{aligned} \tag{13}$$

where r_i , p_i and y_i are of the same sign than the moment generated by the i -th motor.

Since the coaxial octorotor is an over-actuated system, the control allocation problem has a finite number of solutions. In this regard, by taking into account some optimization criterion and actuators constraints, it is possible to find the optimal solution by considering the quadratic programming approach based on minimizing the control input as follows:

$$\begin{aligned}
J &= \argmin \mathbf{u}^{*T} W_i \mathbf{u}^* \\
\text{such that: } &\mathbf{B} \mathbf{u}^* = \mathbf{u}
\end{aligned} \tag{14}$$

where J is the value of the function to be minimized, which has the explicit solution:

$$\mathbf{u}^* = W_i \mathbf{B}^T (\mathbf{B} W_i \mathbf{B}^T)^{-1} \mathbf{u} \tag{15}$$

where $W_i = W_i^T$ is a symmetric positive definite weighting matrix. This method is called the weighted pseudo-inverse method which is often used to solve unconstrained linear control allocation problems. However, in order to counteract the saturation of the healthy actuators in case of failures, another scheme needs to be proposed to update the weighting matrix W_i , which is given at the end of Section 6.1.

5.3. Common subsystems formulation for all FTC schemes

Since the controller stabilizes both the altitude z and the attitude ϕ, θ, ψ of the octorotor, we will consider the state vector as

$$\begin{aligned}
\mathbf{X} &= [x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7 \ x_8]^T \\
&= [z \ \dot{z} \ \phi \ \dot{\phi} \ \theta \ \dot{\theta} \ \psi \ \dot{\psi}]^T
\end{aligned} \tag{16}$$

Given the nonlinear equations of motion in (8), the octorotor dynamic model can be divided into four subsystems as follows:

$$\text{Altitude subsystem} \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f_1 + g_1 u_1 \end{cases} \tag{17}$$

$$\text{Roll subsystem} \begin{cases} \dot{x}_3 = x_4 \\ \dot{x}_4 = f_2 + g_2 u_2 \end{cases} \tag{18}$$

$$\text{Pitch subsystem} \begin{cases} \dot{x}_5 = x_6 \\ \dot{x}_6 = f_3 + g_3 u_3 \end{cases} \tag{19}$$

$$\text{Yaw subsystem} \begin{cases} \dot{x}_7 = x_8 \\ \dot{x}_8 = f_4 + g_4 u_4 \end{cases} \tag{20}$$

where f_i and g_i are defined as:

$$\begin{aligned}
f_1 &= -G \\
f_2 &= x_6 x_8 (I_{yy} - I_{zz}) / I_{xx} \\
f_3 &= x_4 x_8 (I_{zz} - I_{xx}) / I_{yy} \\
f_4 &= x_4 x_6 (I_{xx} - I_{yy}) / I_{zz} \\
g_1 &= \cos(x_3) \cos(x_5) / m \\
g_2 &= 1 / I_{xx} \\
g_3 &= 1 / I_{yy} \\
g_4 &= 1 / I_{zz}
\end{aligned} \tag{21}$$

With the formulation above, and for control design purposes, it is useful to represent each subsystem as a single input nonlinear system given by:

$$\begin{cases} \dot{x}_{2i-1} = x_{2i} \\ \dot{x}_{2i} = f_i + g_i u_i \end{cases} \quad \text{with } i = 1, 2, 3, 4 \tag{22}$$

5.4. Shared sliding manifold function for all FTC schemes

We present here a sliding manifold for an UAV's sliding mode controller. This sliding manifold will be used on the three presented FTCs, in order to compare only the impact of the fault tolerant recovery mechanisms as much as possible.

The integral sliding manifold S_i is proposed as follows:

$$S_i = \{\mathbf{X} \in \mathbb{R}^n \mid s_i(\mathbf{X}) = 0\} \tag{23}$$

The integral switching function $s(\mathbf{X})$ is defined as:

$$s_i(\mathbf{X}) = \dot{e}_i + \lambda_i e_i + k_i \int e_i dt - \lambda_i e_{i0} - \dot{e}_{i0} \tag{24}$$

where λ_i and k_i are the gains for the sliding variables which must be selected as positive values but not too high to provide a smooth response of the system, and (25) gives e_i for (17), (18), (19) and (20) with $i = 1, \dots, 4$, such that e_i and \dot{e}_i represent the position and velocity errors between the real states x_i measured by the vehicle sensors, and the desired values x_i^d calculated by the trajectory generator. e_{i0} and \dot{e}_{i0} are the initial errors.

$$\begin{aligned}
\dot{e}_i &= x_{2i} - x_{2i}^d \\
e_i &= x_{2i-1} - x_{2i-1}^d
\end{aligned} \tag{25}$$

6. FTCs recovery applied to coaxial octorotor

When a motor failure occurs, the control allocation should redistribute in a specific way some control signals to the healthy actuators in order to compensate for the loss of thrust. This could be done by changing the weighting matrix W_i in the STSMC and control mixing techniques, or by modifying gains in the higher level controller u_i to attain the new stability conditions as it is done in the ASMCA and STSMC techniques. In this section, we will first present our contribution for FTC methods: a self-tuning sliding mode control (STSMC) that uses error detection to modify multiplexing and the control law's gains and to adapt to the erroneous state. Then, we will introduce two other FTC methods: a FTC based on off-line control mixing [31] and an adaptive sliding mode control allocation (ASMCA) for future comparison and validation of our proposed recovery method.

6.1. Self-Tuning sliding mode control (STSMC)

We propose in this section an AFTC scheme which automatically tunes itself to an error once this error has been detected and identified. The detection and isolation can be done for example by using additional current sensors equipped to each actuator (see chapter 5 in [33]). This FDI module must identify the faulty actuator and provide the effectiveness loss of each actuator for our method to be applied.

Let us recall the subsystems formulation presented in (17) to (20), the definition of the integral sliding variable in (24) and the definition of the system errors given in (25). The goal of the controller is to drive the sliding variable

s_i and its first derivative to zero, which implies that the difference between the system's position and the desired one will also be driven to zero and the system will follow the desired trajectory. To do this, we design a control law u that ensure the following desired dynamics of the sliding variables s_i :

$$\dot{s}_i = -K_i \text{sat}(s_i/\epsilon_i) \quad (26)$$

where K_i are positive gains of the discontinuous part of the control law and ϵ_i are small positive values defining the boundaries of the linear part of the $\text{sat}(\cdot)$ function, defined as:

$$\text{sat}(s_i/\epsilon) = \begin{cases} \text{sign}(s_i) & \text{if } |s_i| > \epsilon_i \\ s_i/\epsilon_i & \text{if } |s_i| \leq \epsilon_i \end{cases} \quad (27)$$

From Eq. (24) and (25), the first derivative of s_i is calculated as:

$$\begin{aligned} \dot{s}_i &= \ddot{e}_i + \lambda_i \dot{e}_i + k_i e_i \\ &= (\dot{x}_{2i} - \dot{x}_{2i}^d) + \lambda_i \dot{e}_i + k_i e_i \end{aligned} \quad (28)$$

By substituting the expressions of \ddot{x}_{2i} from the subsystems (17), (18), (19) and (20), we get:

$$\dot{s}_i = (f_i + g_i u_i - \ddot{x}_{2i}^d) + \lambda_i \dot{e}_i + k_i e_i \quad (29)$$

In order to obtain the desired dynamics of the sliding variable, we choose the control law as follows:

$$u_i = \frac{1}{g_i} (\dot{x}_{2i}^d - \lambda_i \dot{e}_i - k_i e_i - f_i) - \frac{1}{g_i} K_i \text{sat}(s_i/\epsilon_i) \quad (30)$$

for $i = 1, \dots, 4$

By substituting this control law in Eq. (29), the desired dynamics of the sliding variable (26) are established.

Proposition 1. *Given the nonlinear subsystems (17), (18), (19) and (20), by applying the control law (30) and by choosing $K_i \geq \eta_i > 0$, where η_i is any fixed positive value, the condition $s_i(\mathbf{X}) = 0$ is satisfied and thus the system errors e_i and \dot{e}_i converge to zeros, meaning that the system's state will eventually be the desired one.*

Proof. Consider the following Lyapunov function V_i :

$$V_i = \frac{1}{2} s_i^2 \quad (31)$$

Then the derivative of this function would be by using (24), (25) and (17) to (20):

$$\begin{aligned} \dot{V}_i &= s_i \dot{s}_i \\ &= s_i (\ddot{e}_i + \lambda_i \dot{e}_i + k_i e_i) \\ &= s_i (\dot{x}_{2i} - \dot{x}_{2i}^d + \lambda_i \dot{e}_i + k_i e_i) \\ &= s_i (f_i + g_i u_i - \dot{x}_{2i}^d + \lambda_i \dot{e}_i + k_i e_i) \end{aligned} \quad (32)$$

and by substituting (30) into (32), we get:

$$\begin{aligned} \dot{V}_i &= -K_i s_i \text{sat}(s_i/\epsilon) \\ &\leq -\eta_i s_i \text{sat}(s_i/\epsilon) \\ &\leq -\eta_i |s_i| \end{aligned} \quad (33)$$

Thus the system satisfies the η -reachability condition, and the system will asymptotically reach the desired trajectory. The proof is satisfied for $i = 1, \dots, 4$. \square

In faulty scenarios, actuators failures are modeled as losses in the control effectiveness. We represent these losses by the $\mathbf{L} = \text{diag}(l_1, \dots, l_8)$ matrix, where $l_i = 1$ and $l_i = 0$ indicate respectively that the i -th motor is fully healthy or completely failing. Then the real force \overline{F}_i produced by each actuator is actually:

$$\overline{F}_i = l_i F_i \quad (34)$$

where F_i is the force produced in fault-free situation. In this situation, the real virtual input vector $\bar{\mathbf{u}} = [\bar{u}_f \quad \bar{\tau}_\phi \quad \bar{\tau}_\theta \quad \bar{\tau}_\psi]^T$ produced by the system becomes:

$$\begin{bmatrix} \bar{u}_f \\ \bar{\tau}_\phi \\ \bar{\tau}_\theta \\ \bar{\tau}_\psi \end{bmatrix} = \begin{bmatrix} u_f \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} + \begin{bmatrix} \Delta u_f \\ \Delta \tau_\phi \\ \Delta \tau_\theta \\ \Delta \tau_\psi \end{bmatrix} \quad (35)$$

where $\mathbf{u} = [u_f \quad \tau_\phi \quad \tau_\theta \quad \tau_\psi]^T$ and $\Delta \mathbf{u} = [\Delta u_f \quad \Delta \tau_\phi \quad \Delta \tau_\theta \quad \Delta \tau_\psi]^T$ are respectively the nominal virtual control input calculated by the controller and the input caused by the erroneous actuators, or input fault vector. In practice, an error could be caused either by a damage affecting the motor or the propeller. In this work, we do not distinguish these faults.

Depending on the faulty actuator, additive faults in torques ($\Delta \tau_\phi$, $\Delta \tau_\theta$, $\Delta \tau_\psi$) can be either positive or negative depending on the direction of the actuator rotation and its position with respect to the center of gravity of the vehicle. In contrast to faults altering torques, the additive faults affecting the thrust (Δu_f) can only be negative since the generated thrust doesn't depend on the rotation direction nor on the actuator's position.

First, we consider that a single fault is present in the system. In this case, the additive faults input can be defined as:

$$\begin{aligned} \Delta u_f &= -(1 - l_j)F_j\alpha_{jk}S \\ \Delta \tau_\phi &= \pm(1 - l_j)F_j\alpha_{jk}Sd\frac{\sqrt{2}}{2} \\ \Delta \tau_\theta &= \pm(1 - l_j)F_j\alpha_{jk}Sd\frac{\sqrt{2}}{2} \\ \Delta \tau_\psi &= \pm(1 - l_j)\tau_j \end{aligned} \quad (36)$$

where the j -th and k -th terms represent respectively the faulty actuators and its redundant actuator. Note that this notation will be adopted in the rest of this chapter. The signs of the additive faults in torques are given in Table 1.

Actuator _{i}	$\Delta \tau_\phi$	$\Delta \tau_\theta$	$\Delta \tau_\psi$
1	+	+	+
2	+	+	-
3	+	-	-
4	+	-	+
5	-	-	+
6	-	-	-
7	-	+	-
8	-	+	+

Table 1: Additive faults in torques according to the failing actuator

A new control law $\mathbf{u}^f = [u_f^f \quad \tau_\phi^f \quad \tau_\theta^f \quad \tau_\psi^f]^T$ is designed to recover the system from this faulty situation, such that:

$$\mathbf{u} = \mathbf{u}^f + \Delta \mathbf{u} \quad (37)$$

Thereby, the subsystems (17), (18), (19) and (20), become:

$$\text{Altitude subsystem} \begin{cases} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= f_1 + g_1 u_f^f + g_1 \Delta u_f \end{cases} \quad (38)$$

$$\text{Roll subsystem} \begin{cases} \dot{x}_3 &= x_4 \\ \dot{x}_4 &= f_2 + g_2 \tau_\phi^f + g_2 \Delta \tau_\phi \end{cases} \quad (39)$$

$$\text{Pitch subsystem} \begin{cases} \dot{x}_5 &= x_6 \\ \dot{x}_6 &= f_3 + g_3 \tau_\theta^f + g_3 \Delta \tau_\theta \end{cases} \quad (40)$$

$$\text{Yaw subsystem} \begin{cases} \dot{x}_7 &= x_8 \\ \dot{x}_8 &= f_4 + g_4 \tau_\psi^f + g_4 \Delta \tau_\psi \end{cases} \quad (41)$$

Let us have:

$$\mathbf{u}^f = \begin{bmatrix} u_1^f & u_2^f & u_3^f & u_4^f \end{bmatrix}^T = \begin{bmatrix} u_1^f & u_2^f & u_3^f & u_4^f \end{bmatrix}^T \quad (42)$$

We propose the following control law to recover the system from erroneous situations:

$$\begin{aligned} u_i^f &= \frac{1}{g_i}(\dot{x}_{2i}^d - \lambda_i \dot{e}_i - k_i e_i - f_i) - \frac{1}{g_i} K_i^f \text{sat}(s_i/\epsilon_i) \\ K_i^f &= K_i + \Delta u_{ij}^{max}, \quad i = 1, \dots, 4 \end{aligned} \quad (43)$$

where Δu_{ij}^{max} is the additive fault term of the input fault vector Δu_j^{max} due to the fault injected to the j -th actuator given in Eq. (44). In fact, Δu_{ij}^{max} is positive and bounds the maximal value of Δu_i . Thus, by adding this positive value to the control gain K_i^f of the proposed control law (43), we can prove (using the Proposition 2 and its proof) that the system will be able to maintain its overall stability in case of actuator failures.

It can be seen that the control parameter K_i^f updates itself over time based on the fault detection information carried by the additive bounded input fault vector Δu_j^{max} . Thus, the proposed controller adapts over time depending on the actuators faults in order to maintain the stability of the system.

The expression of the input fault vector Δu_j^{max} is given by:

$$\Delta u_j^{max} = \begin{bmatrix} \Delta u_{fj}^{max} \\ \Delta \tau_{\phi j}^{max} \\ \Delta \tau_{\theta j}^{max} \\ \Delta \tau_{\psi j}^{max} \end{bmatrix} = \begin{bmatrix} \Delta u_{1j}^{max} \\ \Delta u_{2j}^{max} \\ \Delta u_{3j}^{max} \\ \Delta u_{4j}^{max} \end{bmatrix} = \begin{bmatrix} l_j F_j^{max} \alpha_{jk} S \\ l_j F_j^{max} \alpha_{jk} S d \frac{\sqrt{2}}{2} \\ l_j F_j^{max} \alpha_{jk} S d \frac{\sqrt{2}}{2} \\ l_j \tau_j^{max} \end{bmatrix} \quad (44)$$

where F_j^{max} and τ_j^{max} for our experimental drone's actuators have been identified in [33]. The l_j are the effectiveness losses detected by the FDI mechanism, as previously stated in this section, and the other terms are dependent of the system and are identified for our experimental UAV in section 7.

Proposition 2. *Given the nonlinear subsystems (17), (18), (19) and (20), by applying the new control law (43) and by choosing $K_i \geq \eta_i > 0$ in a similar way than for the fault-free control law, the condition $s_i(\mathbf{X}) = 0$ is satisfied and thus the system errors e_i and \dot{e}_i converge to zeros, meaning that the system's state will eventually be the desired one.*

Proof. Consider the following Lyapunov function V_i :

$$V_i = \frac{1}{2} s_i^2 \quad (45)$$

Then the derivative of this function would be

$$\begin{aligned} \dot{V}_i &= s_i \dot{s}_i \\ &= s_i(\ddot{e}_i + \lambda_i \dot{e}_i + k_i e_i) \\ &= s_i(\ddot{x}_{2i} - \dot{x}_{2i}^d + \lambda_i \dot{e}_i + k_i e_i) \\ &= s_i(f_i + g_i u_i^f + g_i \Delta u_i - \dot{x}_{2i}^d + \lambda_i \dot{e}_i + k_i e_i) \end{aligned} \quad (46)$$

and by substituting (43) into (46), we get:

$$\begin{aligned} \dot{V}_i &= -K_i^f s_i \text{sat}(s_i/\epsilon) + g_i \Delta u_i s_i \\ &= -K_i s_i \text{sat}(s_i/\epsilon) - g_i \Delta u_{ij}^{max} s_i \text{sat}(s_i/\epsilon) + g_i \Delta u_i s_i \\ &= -K_i |s_i| - g_i (\Delta u_{ij}^{max} |s_i| - \Delta u_i s_i) \end{aligned} \quad (47)$$

Let us recall the expressions of g_i (from Eq. (21):

$$\begin{aligned} g_1 &= \cos(\phi) \cos(\theta)/m \\ g_2 &= 1/I_{xx} \\ g_3 &= 1/I_{yy} \\ g_4 &= 1/I_{zz} \end{aligned} \quad (48)$$

Since the parameters I_{xx}, I_{yy} and I_{zz} are all positive, thus the terms g_2, g_3, g_4 are positives. Moreover, since the roll and pitch angles are constrained to $(-\pi/2 < \phi, \theta < \pi/2)$, and using the fact that m is the mass of the UAV which is a positive value, we can conclude that g_1 is also positive. Then, the following inequality holds true:

$$g_i > 0 \quad (49)$$

And, as Δu_{ij}^{max} is positive and bounds the maximal value of Δu_i , we can write:

$$\Delta u_{ij}^{max} > |\Delta u_i| \quad (50)$$

Thus, using the Eq. (49) and (50), we have:

$$\begin{aligned} \dot{V}_i &= -K_i |s_i| - g_i(\Delta u_{ij}^{max} |s_i| + \Delta u_i s_i) \\ &\leq -K_i |s_i| \\ &\leq -\eta_i |s_i| \end{aligned} \quad (51)$$

Then the system satisfies the η -reachability condition, and the system will asymptotically track the desired trajectory. \square

In case of multiple faults, the term Δu_{ij}^{max} is replaced by the sum of the input fault vectors corresponding to all the injected faults. The additive faults input for multiple faults thus becomes:

$$\begin{aligned} \Delta u_f &= -\sum_j (1 - l_j) F_j \alpha_{jk} S \\ \Delta \tau_\phi &= \sum_j \pm (1 - l_j) F_i \alpha_{jk} S d \frac{\sqrt{2}}{2} \\ \Delta \tau_\theta &= \sum_j \pm (1 - l_j) F_i \alpha_{jk} S d \frac{\sqrt{2}}{2} \\ \Delta \tau_\psi &= \sum_j \pm (1 - l_j) \tau_i \end{aligned} \quad (52)$$

where the j -th terms are the indices of all the failing actuators and, as we mentioned previously, the k -th terms represent the redundant actuators of the j -th failing actuators. The same proof of stability used in the case of one fault holds true in the case of multiple faults by adding the sum of the upper bounds $\sum_j \Delta u_{ij}^{max}$ of all the injected faults in equation (43).

Note that when an error occurs, the weighting matrix W_i , in Eq (15), is updated according to the output of the FDI module, i.e., the detected fault information, namely $w_i = 1/l_i$. When the i -th actuator becomes faulty, its weight value increases and its control signal also increases. If the control signal exceeds the power limit of the actuator, it will cause saturation. The following weighting algorithm \hat{W}_i is always used in (15) instead of W_i to avoid this actuators saturation::

$$\begin{aligned} \hat{W}_i &= \Xi W_i \Xi \quad \text{with} \quad \Xi = \text{diag}(\xi_1, \xi_2, \dots, \xi_8) \\ \xi_i &= \begin{cases} (1 + \eta_1) \frac{u_i^*}{\eta_2 u_{imax}^*}, & u_i^* > \eta_2 u_{imax}^* \\ 1, & \eta_2 u_{imin}^* \leq u_i^* \leq \eta_2 u_{imax}^* \end{cases} \end{aligned} \quad (53)$$

where η_1 and η_2 are small positive values with $0 < \eta_2 \leq 1$.

6.2. AFTC based on an offline control mixing

Since the coaxial octorotor configuration is an over-actuated system, there exists a finite number of solutions to the control allocation problem (11). In [31], a static reallocation for each motors failures case is obtained by resolving the optimization problem (14) for each fault situation.

By considering that at least four motors are healthy, the number of complete failures combinations is $N_f = \sum_{i=1}^4 C_4^i = 162$.

However, due to the symmetry in the coaxial octorotor's configuration, the number of dissimilar combinations is highly reduced to 27.

In this FTC scheme, the same baseline control law (30) is used in all cases, i.e., in nominal and faulty situations. However, as mentioned in [31], after each error detection, a new set of control gains depending of the new system's configuration must be associated to the control law. These gains must be determined empirically for every one of the erroneous configuration that we want to consider. For complete motors failures, the number of dissimilar configurations can be reduced to 27 as previously stated. However, if we want to take into account partial motors failures, this number can grow to hundreds.

6.2.1. Fault-Free Mode

In nominal behavior, the expression of the i th motor speed ω_i is given by (54):

$$\omega_i = \sqrt{\frac{1}{8} \cdot \left(\frac{u_f}{\alpha_{ij} \cdot K_f \cdot S} \pm \frac{\tau_\phi}{\alpha_{ij} \cdot K_f \cdot S \cdot d \cdot \frac{\sqrt{2}}{2}} \pm \frac{\tau_\theta}{\alpha_{ij} \cdot K_f \cdot S \cdot d \cdot \frac{\sqrt{2}}{2}} \pm \frac{\tau_\psi}{K_t} \right)} \quad (54)$$

In fact, the above expression of ω_i represents the solution of the equation (11) by taking $L = I_{8 \times 8}$.

6.2.2. One complete failure

To deal with the case of one motor's complete failure, the solution proposed in [31] from the resolution of 14 (with one $l_i = 0$) is to reduce the power of its dual to the half, and to increase the powers of the three upper remaining actuators by a factor of 1.5. For example considering a complete failure of motor 1 ($l_1 = 0$), we would have:

$$\begin{aligned} F'_6 &= 0.5F_6 \\ F'_i &= 1.5F_i \quad i = 2, 4, 8 \\ F'_i &= F_i \quad i = 3, 5, 7 \end{aligned} \quad (55)$$

Failed Motors	Reallocated inputs
1 & 2	$F'_i = 0 \quad i = 5, 6$ $F'_i = 2F_i \quad i = 3, 4, 7, 8$
1 & 3	$F'_i = 2F_i \quad i = 2, 4$ $F'_i = F_i \quad i = 5, 6, 7, 8$
1 & 4	$F'_i = 2F_i \quad i = 2, 3, 5, 8$ $F'_i = 0 \quad i = 6, 7$
1 & 5	$F'_i = F_i \quad i = 2, 3, 6, 7$ $F'_i = 2F_i \quad i = 4, 8$
1 & 6	$F'_i = \frac{4}{3}F_i \quad i = 2, 3, 4, 5, 7, 8$
1 & 7	$F'_i = 2F_i \quad i = 2, 8$ $F'_i = F_i \quad i = 3, 4, 5, 6$
1 & 8	$F'_i = 2F_i \quad i = 2, 4, 5, 7$ $F'_i = 0 \quad i = 3, 6$

Table 2: Static reallocation in case of two motors failures, adopted from [31]

Failed Motors	Reallocated inputs
1 & 2 & (3 or 4 or 7 or 8)	—
1 & 2 & (5 or 6)	$F'_i = 2F_i \ i = 3, 4, 7, 8$ $F'_i = 0 \ i = 6 \text{ or } 5$
1 & 3 & 4	—
1 & 3 & 5	$F'_i = 1.5F_i \ i = 2, 6, 8$ $F'_i = 2.5F_i \ i = 4$ $F'_i = F_i \ i = 7$
1 & 3 & (6 or 8)	$F'_i = 2F_i \ i = 2, 4, 5, 7$ $F'_i = 0 \ i = 8 \text{ or } 6$
1 & 3 & 7	$F'_i = 2.5F_i \ i = 2$ $F'_i = 1.5F_i \ i = 4, 6, 8$ $F'_i = F_i \ i = 5$
1 & 4 & 5	—
1 & 4 & (6 or 7)	$F'_i = 2F_i \ i = 2, 3, 5, 8$ $F'_i = 0 \ i = 7 \text{ or } 6$
1 & 5 & 6	$F'_i = 2F_i \ i = 3, 4, 7, 8$ $F'_i = 0 \ i = 2$
1 & 5 & 7	$F'_i = 2.5F_i \ i = 8$ $F'_i = 1.5F_i \ i = 2, 4, 6$ $F'_i = 1F_i \ i = 3$
1 & 6 & 7	$F'_i = 2F_i \ i = 2, 3, 5, 8$ $F'_i = 0 \ i = 4$
1 & 6 & 8	$F'_i = 2F_i \ i = 2, 4, 5, 7$ $F'_i = 0 \ i = 3$
1 & 7 & 8	—

Table 3: Static reallocation in case of three motors failures, adopted from [31]

Failed Motors	Reallocated inputs
1 & 2 & 5 & 6	$F'_i = 2F_i \ i = 3, 4, 7, 8$
1 & 3 & 5 & 7	$F'_i = 2F_i \ i = 2, 4, 6, 8$
1 & 4 & 6 & 7	$F'_i = 2F_i \ i = 2, 3, 5, 8$
2 & 3 & 5 & 8	$F'_i = 2F_i \ i = 1, 4, 6, 7$
2 & 4 & 5 & 7	$F'_i = 2F_i \ i = 1, 3, 6, 8$
2 & 4 & 6 & 8	$F'_i = 2F_i \ i = 1, 3, 5, 7$
3 & 4 & 7 & 8	$F'_i = 2F_i \ i = 1, 2, 5, 6$

Table 4: Static reallocation in case of four motors failures, adopted from [31]

6.2.3. Two, three and four complete motors failures

As mentioned before, in case of multiple complete failures, many solutions of the optimization problem (14) are symmetrically equivalent since the frame configuration of the coaxial octorotor is symmetric. So, without loss of generality, only the cases where motors 1 and $i = 2, 3, \dots, 8$ fail are presented in Tables 2 and 3. Also, the Table 4 presents all the possible cases of four motors failures. Note that the dashed line indicates that no solution was found with this combination [31].

6.3. Adaptive sliding mode control allocation (ASMCA)

In this robust approach, an adaptive control allocation and re-allocation strategy [15] is proposed to redistribute control signals among healthy motors. This ASMCA is employed to maintain the overall system performance, providing tolerance to motors faults but also robustness to external perturbations.

The proposed control law is as follows:

$$u_i = \hat{\Upsilon}_i(\dot{x}_{2i}^d - \lambda_i \dot{e}_i - k_i e_i - f_i) - \hat{\Upsilon}_i K_i * \text{sat}(s_i/\epsilon_i) \quad (56)$$

The parameter $\hat{\Upsilon}_i$ is adaptively adjusted online when there is an error between the baseline controller u_i and the desired one u_i^d in the case of actuator faults or external perturbations. The online update scheme of the parameter $\hat{\Upsilon}_i$ is given by:

$$\dot{\hat{\Upsilon}}_i = (-\dot{x}_{2i}^d + \lambda_i \dot{e}_i + k_i e_i + f_i) + K_i * \text{sat}(s_i/\epsilon_i) s_{\Delta_i} \quad (57)$$

where $s_{\Delta_i} = s_i - \epsilon \text{sat}(s_i/\epsilon)$. Further information about this algorithm are given in [15].

Symbole	Parameter	Value	Unit
K_f	Thrust factor	$3 * 10^{-5}$	$N.s^2 / rad^2$
K_t	Drag factor	$7 * 10^{-7}$	$N.m.s^2 / rad^2$
I_{xx}, I_{yy}	Inertia	$4.2 * 10^{-2}$	$Kg.m^2$
I_{zz}	Inertia	$7.5 * 10^{-2}$	$Kg.m^2$
d	length of the arm	0.23	m
m	mass	1.6	Kg
F^{max}	Max forces	5	N
$\tau_\phi^{max}, \tau_\theta^{max}$	Max torque	2	Nm
τ_ψ^{max}	Max torque	0.08	Nm

Table 5: Coaxial octorotor parameters

The main steps of the adaptive allocation are conducted as follows:

1. When there exists an error between the actual baseline controller u_i and the desired controller u_{id} , which means that the position and the velocity of the system are different from the desired values, thus the parameter $\hat{\Upsilon}_i$ is updated following the equation (57) to maintain the system performance.
2. The control effectiveness matrix B_i is updated accordingly through a defined relationship $\hat{B}_i = \hat{\Upsilon}_i * T_i$, where the relationship between g_i from equations (17) to (20) and B_i is bounded by $T_i = g_i B_i$.
3. The control signals are redistributed to the healthy motors through the relationship $u^* = W_i \hat{B}^T (\hat{B} W_i \hat{B}^T)^{-1} u$

It is proved in [15] that for a nonlinear system with bounded disturbance, the sliding motion will be achieved and maintained by applying the feedback control law (56) and the online adaptation scheme (57) and by an appropriate choice of the discontinuous gains K_i .

7. Experimental Validation

In this section, we demonstrate the performance of our proposed STSMC. Also, we test and compare the behavior of the three FTC schemes in real indoor experiments. In these experiments, four successive fault injections are used to simulate the total failures of top actuators of the coaxial octorotor (actuators 6, 2, 4 and 8) by sending stop commands at desired times.

7.1. Experimental platform

The experimental coaxial octorotor is shown in Fig. 5. It was designed and built at the Heudiasyc Laboratory. Its parameters are given in Table 5.

The localization is done using an Optitrack motion capture system which provides the position measurements (position x, y, z and heading ψ in Earth frame) to the position control loop of the UAV. In addition, the octorotor is equipped with a 3DM-GX3-25 Microstain IMU composed of accelerometers, gyroscopes, and a magnetometer to provide inertial measurements and static/dynamic orientation. Also, an ultrasonic telemeter SRF08 is added to enhance the altitude measurement.

An embedded computer module is mounted on the drone, namely the IGEPv2. It is a fanless computer platform with a compact size and low power consumption which is programmed using a C++ compiler. The states of the vehicle are measured by the sensors and transmitted to the IGEPv2 through different communication interfaces (serial communication, wifi, etc.). The software used to write applications for the octorotor is called FL-AIR, which has been developed at the Heudiasyc laboratory to ease development and integration of research algorithms. FL-AIR is based on Linux and is compatible with real time features (Xenomai). Also, a ground station is built to monitor all the octorotor states in real time.

Finally, in order to compare the FTC methods, we use the same initial control parameters for all FTC control schemes, i.e. in (43) and (56):

$$\begin{aligned}
\lambda_1 &= 3, & \lambda_2 &= 5, & \lambda_3 &= 5, & \lambda_4 &= 4 \\
k_1 &= 3, & k_2 &= 4, & k_3 &= 4, & k_4 &= 5 \\
K_1 &= 6, & K_2 &= 5, & K_3 &= 5, & K_4 &= 6 \\
\epsilon &= 0.15
\end{aligned} \tag{58}$$

Note that for the reconfigurable control mixing technique, we assign new control parameters after each failure injection to meet the new condition of the system. The gains for each recovery configuration are determined experimentally and are not given here for lack of space. On the other hand, for the STSMC and ASMCA, the initial control parameters (58) are adjusted online with respect to equations (43) and (56) respectively.

7.2. Fault detection and isolation (FDI) using current sensors

As mentioned before, the system recovery in AFTCs is depending upon the occurring errors in the system being detected and identified. Therefore, a fault detection and isolation (FDI) unit is required to detect the failed component and to estimate the fault parameters. Fault detection and isolation techniques can be classified into two categories, namely Model-based FDI and Model-free FDI.

In Model-based FDI, the detection, isolation and identification of faults are done by comparing the output of the system available measurements with *a priori* information represented by the dynamics equations of the system. In Model-free FDI, the detection relies only on real-time or historical data collected from the sensors and measurements in order to detect and isolate failed components without the need of any information about the system dynamics.

In our experiments, the FDI is supposed to be carried out by comparing each actuator's current to the current that it should theoretically have given the desired command (a Model Based FDI). This technique was successfully tested in [33], but we only simulated it in our experiments in order to focus on the reconfiguration problem of the FTC.

Fig. 3 and 4 illustrate the motor currents during a real hovering flight. Note that after the injection of the first failure, the current measurements take about one second before converging to zero. We chose in our experiments to fix the simulated error detection time at 0.5s as we consider from this data that an 80% decrease of motor current could be correlated to the complete failure of the actuator.

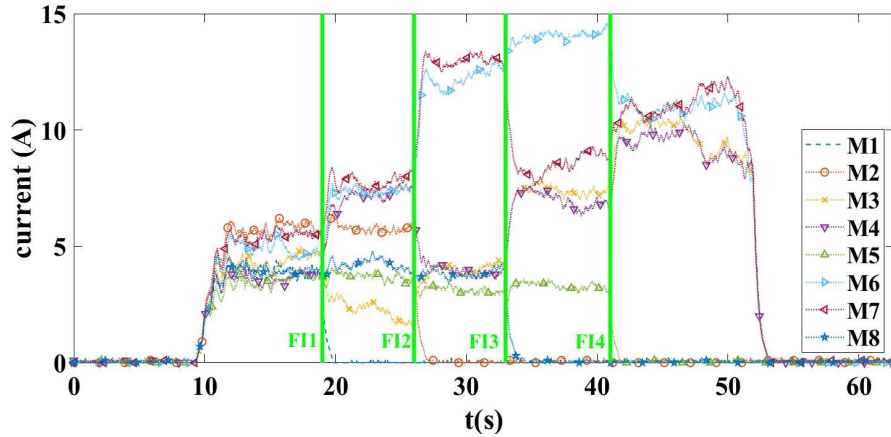


Figure 3: Motors currents during real hover flight and successive failures injections (FI)

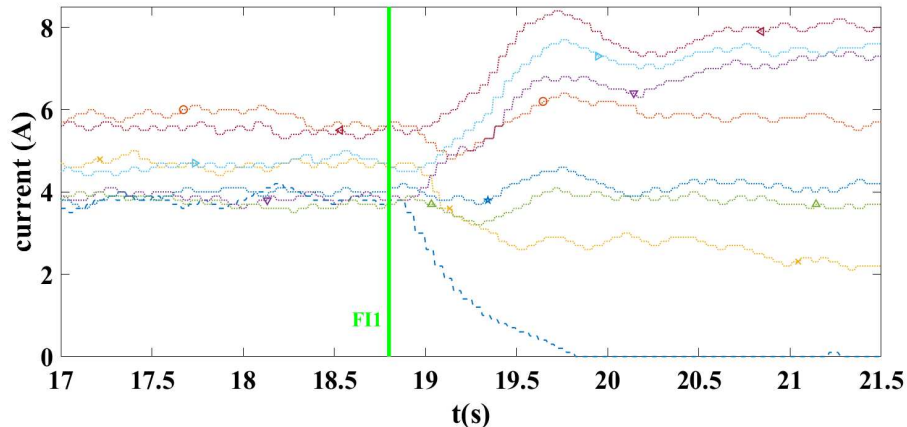


Figure 4: Zoomed view of motors currents during real hover flight and successive failures injections (FI)



Figure 5: Experimental Coaxial Octorotor

7.3. Experimental Results

To compare the three FTC schemes, two indoor scenarios are considered: (1) The coaxial octocopter is required to do a stable flight at 1 meter altitude. (2) The octocopter is required to follow a squared trajectory of 2 meters sides at the same altitude of 1 meter. For each FTC scheme, we realized 10 experimental tests of each flight, thus a total of 60 flight tests were carried out, as each test is not completely reproducible but can be influenced by external perturbations such as sensors noises or communications delays.

To study this number of experimentation, we plot the results using the box and whisker technique, since it is able to summarize a set of data and is ideal for comparing distributions. In this representation the box ends represent the two middle quartiles of data measurements (between 25% and 75%), while the whiskers refer to the higher and lower quartile (respectively, between 0% and 25%, and between 75% and 100%). Finally, the mean and the median of the observations are represented respectively by a star and an horizontal line inside the box.

7.3.1. Hovering flight

To characterize the performance of the three FTC schemes in handling motor failures, first we performed 10 trials of the hovering scenario using each method (STSMC, Multiplexing, Adaptive). In each experiment, the top motors (successively 6, 2, 4 and 8) are turned off from the ground station at the same times $t_1 = 18.28s$, $t_2 = 25.8s$, $t_3 = 32.8s$ and $t_4 = 40.8s$. Position data of the different trials are collected between $t = 17s$ and $t = 45s$ right before the first injection, and after the last one. The three controllers were equally stable at the beginning of the data collection. All the data are illustrated in Fig. 6, Fig. 7 and Fig. 8. To simplify the analysis of the performance, we calculated the average positions of the trials for each FTC scheme, and we illustrate them in Fig. 9, tables 7 and 8. By comparing these results, it can be seen that in all cases the altitude errors are less important than the (x, y) position errors. This is because all the eight actuators contribute to the height control in the same direction and the loss of one motor is a loss of 1/8 of the total number of actuators in this direction. However, for the roll and pitch control we have 4 actuators generating positive orientations and the 4 remaining actuators generating negative orientations, and the loss of one motor is then a loss of 1/4 of the total number of actuators contributing to the same torque. The loss of one motor has thus more influence on the (x, y) position than on the altitude.

The altitude tracking performance of the trials averages is shown in Fig. 10.

7.3.2. Trajectory Tracking flight

To characterize the performance of the three FTC schemes in handling motor failures when following trajectories, we realized another flight scenario, where the coaxial octocopter has to follow a squared path starting from the initial position (corner A ; $x = 0$, $y = 0$) as shown in Fig. 15. This is the case of a trajectory tracking problem where the desired positions and velocities are time-dependent and the system is required to follow the squared reference and reach successively the ABCDA corners of the rectangle as shown in Fig. 15. Again, we performed 10 trials using each FTC scheme, where the failures to the top motors (successively 6, 2, 4 and 8) are injected in all experiments at the same times $t_1 = 18.28s$, $t_2 = 25.8s$, $t_3 = 32.8s$ and $t_4 = 40.8s$. Data on all experiments are illustrated in Fig. 11, Fig. 12 and Fig. 13, and the average positions and errors information are presented respectively in Fig. 14, Tables 9 and 10.

In order to generate the position and velocity of the reference trajectory, the algorithm 1 is implemented in the system. The desired velocity is calculated based on the predefined maximal acceleration $A_{max} = 0.2m/s^2$ in our case. Furthermore, a saturation value of $V_{max} = 0.3m/s$ is chosen in order to limit the maximum velocity of the UAV. Also note that the desired position x_d is calculated by considering timesteps of $\Delta T = 0.01s$.

Fig. 16 and 17 show the evolution of the positions and translational velocities in the x and y directions during the trajectory following the square path ABCDA. We can see that all experiments were conducted using the same

Algorithm 1: Trajectory generator for position displacement in segment AB

Data: e_x, V_x : position error in the x direction between the desired position and the actual current position and UAV translational speed in the x direction

x_d, V_d : desired x position and velocity

ΔT : time difference between two trajectory generation cycles.

Constant: V_{max}, A_{max} : maximum allowable velocity and acceleration

```
1 begin
2   if  $|e_x| > V_{max}^2/A_{max}$  then
3     if  $V_x < V_{max}$  then
4        $V_d = V_x + A_{max} * \Delta T$ 
5     else
6        $V_d = V_{max}$ 
7   else
8      $V_d = V_x - A_{max} * \Delta T$ 
9    $x_d = x_d + V_d * \Delta T$ 
```

reference values for the positions and velocities with respect to time. Although these references are calculated online, the UAV's behaviors in the different methods are still similar enough to generate the same references. Fig. 18 shows the attitude of the UAV during the experiments. We can see that the angles stayed small even after the fault injections, respecting this particular assumption in our model. Finally, the motor speeds of the different actuators using respectively the Adaptive, the STSMC and the Multiplexing schemes are depicted in Fig. 19, 20 and 21. These results show that no saturation were encountered during the flights, which means that the saturation algorithm for the weighting matrix 53 was never activated.

The video of the experiments can be found at <https://youtu.be/Y73o4oESbIs>.

8. Discussion

In this section, we will discuss and compare the different behaviors and aspects of the three FTC schemes considering five aspects: fault tolerance efficiency, trajectory tracking performance, development costs, computation time and health monitoring. Those different aspects are summarized in Table 6.

8.1. Fault Tolerance Efficiency

All three FTCs give good results and successfully tolerate the four successive faults. However, it can be seen on Fig. 10 and 15 that compared to the Adaptive method, the Multiplexing and the STSMC have a smaller response time in compensating for the effects of the injected faults. This is probably due to the fact that the adaptation process of the adaptive method needs some time to converge to the required control parameters, even though the active fault tolerance methods are applied after a detection time delay of 0.5s. Also, Fig. 19, 20 and 21 show that the healthy actuators have similar behaviors in all methods and are able to tolerate the faults without saturation.

8.2. Trajectory Tracking Performance

From Table 7, 8, 9 and 10, we can see that the lowest errors are recorded when using the Multiplexing method. This is probably due to its parameters being obtained by resolving an optimization problem, while the STSMC gains bounds the error (see equation (43)) and the adaptive gains are based on a comparison with noisy sensors outputs. Also, the adaptive method shows slightly bigger errors than the STSMC. Fig. 16 shows that the three methods follows a similar path with respect to time. Fig. 17 and 18 show similar behavior of the angles and translational velocities in the different methods.

8.3. Development cost

The Multiplexing FTC method is very costly to develop since it requires to determine the control law's gains for every possible faults situations. This cost becomes even greater if we take into account partial failures in addition to complete ones. On the other hand, the STSMC and the Adaptive method can both be used to deal with all possible faulty scenarios without more development effort. Nevertheless, the STSMC still requires more knowledge about the model of the system, since a thruster model identification must be done to determine the thrust and drag coefficients of the actuator/propeller system in order to obtain the tuning law.

	STSMC	Multiplexing	Adaptive
Fault Tolerance efficiency	+	++	+
Trajectory Tracking performance	+	++	+
Ease of Development	+	-	+
Low computation time	+	+	-
Health monitoring	+	+	-

Table 6: Comparison of aspects and features of the different FTC schemes

8.4. Computation time

The averages of the computation time of each FTC during an experiment (from $t = 15s$ to $t = 45s$) of the baseline controller for each FTC scheme are given in Table 11. It shows that the STSMC and the Adaptive methods are more expensive than the Multiplexing method in terms of computational complexity by a ratio of 1.5. This is certainly because they involve online computation of the controller parameters while the Multiplexing method just switch from one control law to another.

8.5. Health monitoring

An advantage of using active fault tolerance schemes (Multiplexing and STSMC) is that we can know if the system is getting close to its fault tolerance limit since we are monitoring the system health. This is not possible in the case of the Adaptive method, and therefore it may not be as reliable in critical cases. Nevertheless, the Adaptive method also adds more robustness to the system against external disturbances and does not need a FDI mechanism. Also note that an added FDI mechanism to the Adaptive method, even if not required, would allow to monitor the system's health in the same way than the two other methods.

FTC	RMS e_x (m)	RMS e_y (m)	RMS e_z (m)
STSMC	0.0835	0.0777	0.0453
Multiplexing	0.0621	0.0707	0.0448
Adaptive	0.0975	0.0882	0.0614

Table 7: hover flight RMS errors

FTC	e_{xmax} (m)	e_{ymax} (m)	e_{zmax} (m)
STSMC	0.222	0.151	0.143
Multiplexing	0.204	0.139	0.135
Adaptive	0.213	0.161	0.175

Table 8: hover flight Max errors

FTC	RMS e_x (m)	RMS e_y (m)	RMS e_z (m)
STSMC	0.0882	0.0909	0.0553
Multiplexing	0.0723	0.0798	0.049
Adaptive	0.0950	0.1009	0.0616

Table 9: Square Trajectory flight RMS errors

FTC	e_{xmax} (m)	e_{ymax} (m)	e_{zmax} (m)
STSMC	0.224	0.249	0.155
Multiplexing	0.218	0.239	0.141
Adaptive	0.281	0.238	0.180

Table 10: Square Trajectory flight Max errors

FTC	CPU clock (s)
STSMC	0.110196
Multiplexing	0.075216
Adaptive	0.115344

Table 11: Time complexity of each FTC method between $t_4 = 17s$ and $t_4 = 45s$ during an experiment.

9. Conclusion and perspectives

Based on the experimental results we obtained, it can be concluded that the proposed STSMC, the Multiplexing, and the Adaptive fault tolerant control strategies all allow a coaxial octorotor to maintain stability after losing up to four motors.

Although the Multiplexing FTC scored the lowest position errors and time complexity, it is not efficient in development cost as every possible fault scenarios must be studied both theoretically and experimentally to determine respectively the multiplexing parameters of the control law and its gains. Meanwhile, the proposed STSMC can be considered as a good practical alternative to the Multiplexing strategy, since it shows better performance and faster response time compared to the Adaptive method and allows to monitor the system's health with almost no added development cost.

In future works, we intend to consider partial faults and simultaneous faults/failures to conduct better analyses and to study more extensively the advantages of each FTC strategy. We also intend to conduct the same experiments outdoor by adding the wind compensation strategy proposed in [34]. Also, finite-time sliding mode controller [35] will be considered as the baseline controller for the FTCs strategies in order to study how it improves the system's stability during recovery. Furthermore, we intend to investigate the saturation problems which may occur after failures in heavy maneuvers such as high speeds or sharp turns, and to extend our method with a global solution to prevent this problem.

ACKNOWLEDGMENTS

This work was carried out in the framework of the Labex MS2T (Reference ANR-11-IDEX-0004-02) and the ROBOTEX Equipment of Excellence (Reference ANR-10- EQPX-44). It was supported by the French Government, through the program *Investments for the future* managed by the National Agency for Research and by the National Council for Scientific Research in Lebanon and the Lebanese University research support program.

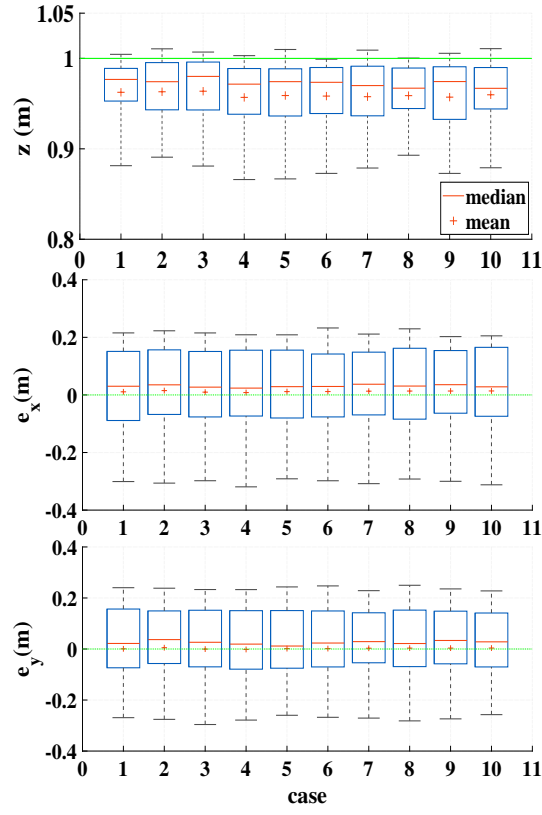


Figure 6: Box and whisker plot for 10 trials of the Adaptive method after four successive failures in hover flight, showing the altitude $z(m)$, and position errors $e_x(m)$ and $e_y(m)$ between $t=17s$ and $t=45s$.

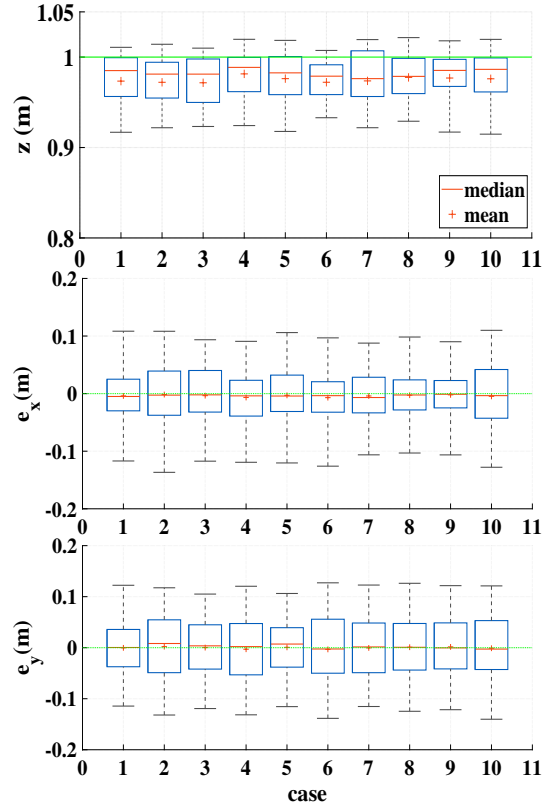


Figure 7: Box and whisker plot for 10 trials of the Multiplexing method after four successive failures in hover flight, showing the altitude $z(m)$, and position errors $e_x(m)$ and $e_y(m)$ between $t=17s$ and $t=45s$.

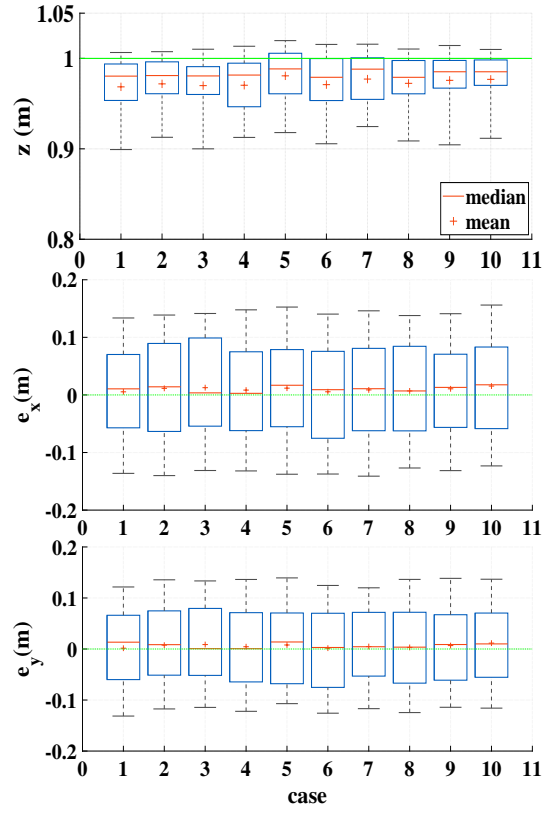


Figure 8: Box and whisker plot for 10 trials of the STSMC method after four successive failures in hover flight, showing the altitude z (m), and positions x (m) and y (m) between $t=17$ s and $t=45$ s.

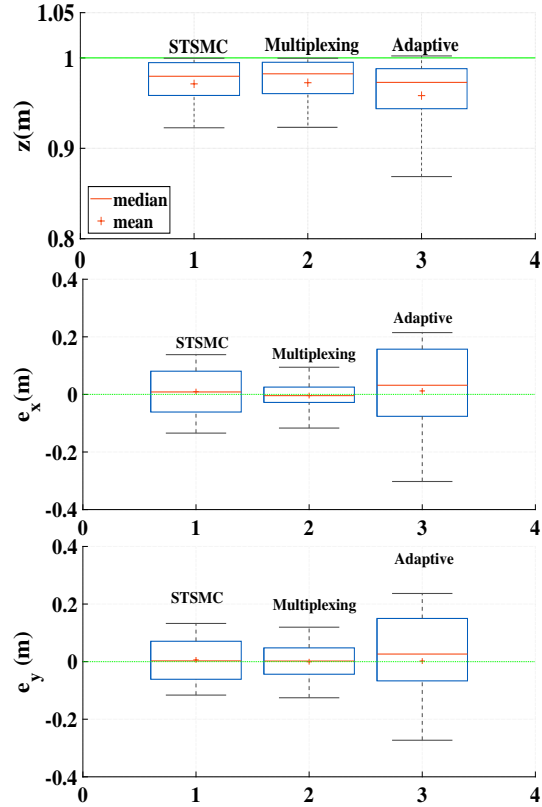


Figure 9: Box and whisker plot for means of each FTC scheme trials after four successive failures in hover flight, showing the altitude z (m), and positions x (m) and y (m) between $t=17$ s and $t=45$ s.

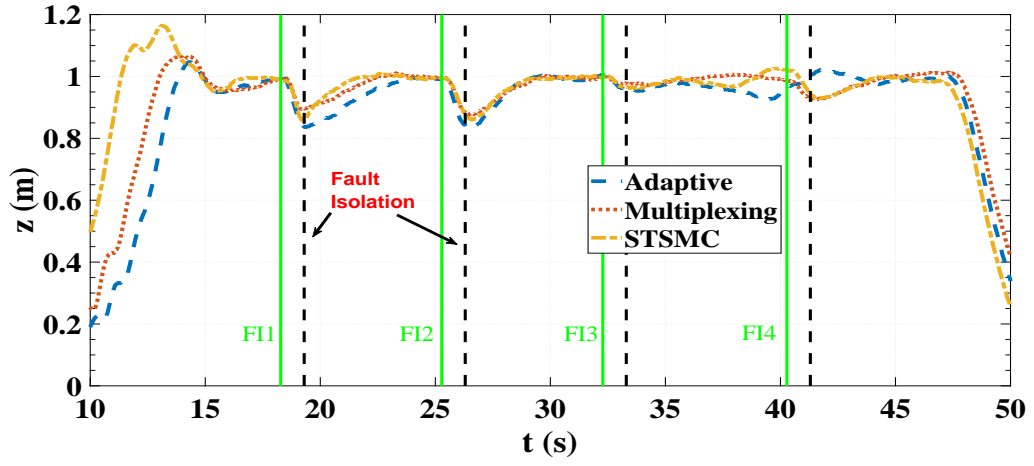


Figure 10: Behavior of 10 trials means of Adaptive, STSMC, and Multiplexing methods after four successive failures injection to the top motors in hover flight

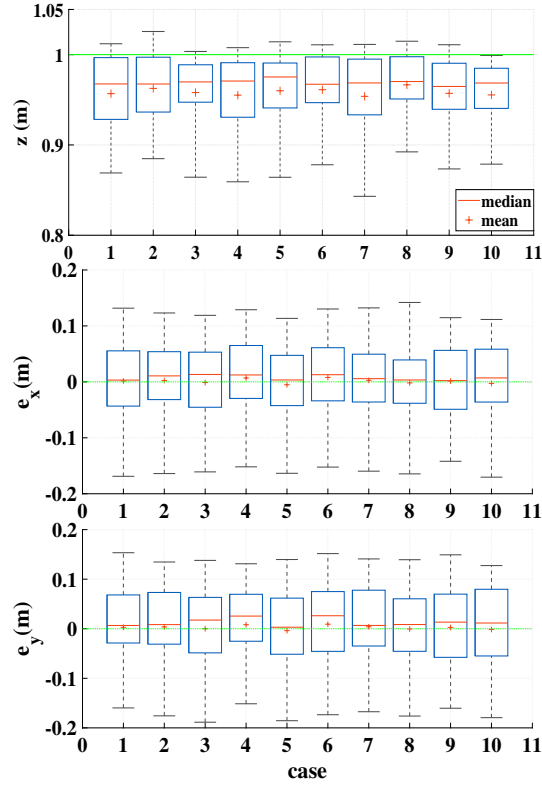


Figure 11: Box and whisker plot for 10 trials of the Adaptive method after four successive failures in square trajectory flight, showing the altitude $z(m)$, and position errors $e_x(m)$ and $e_y(m)$ between $t=17s$ and $t=45s$.

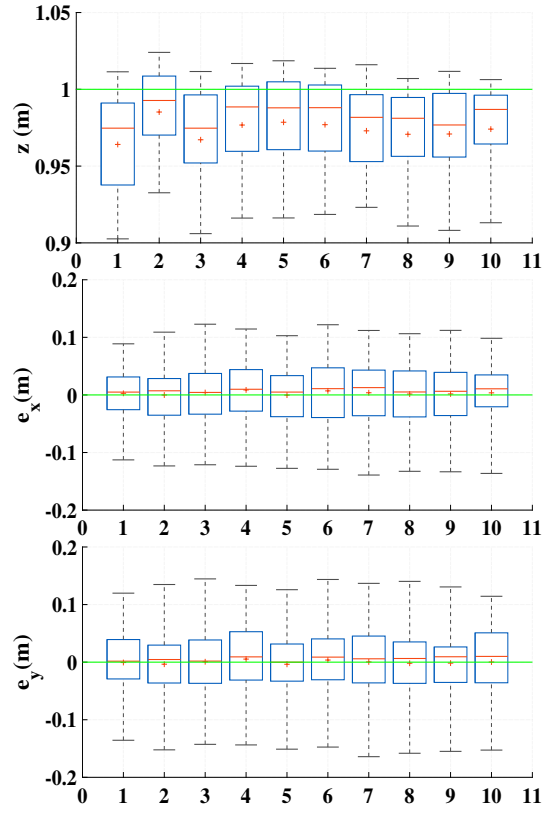


Figure 12: Box and whisker plot for 10 trials of the STSMC method after four successive failures in square trajectory flight, showing the altitude $z(m)$, and position errors $e_x(m)$ and $e_y(m)$ between $t=17s$ and $t=45s$.

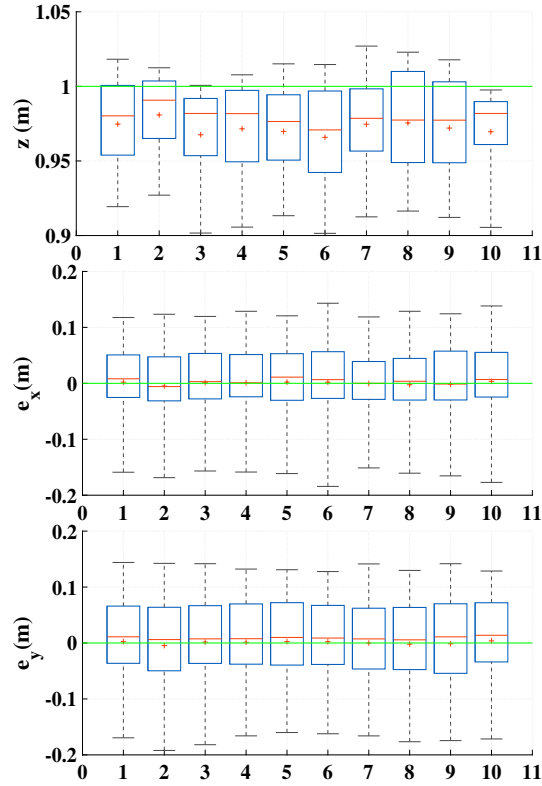


Figure 13: Box and whisker plot for 10 trials of the Multiplexing method after four successive failures in square trajectory flight, showing the altitude $z(m)$, and position errors $e_x(m)$ and $e_y(m)$ between $t=17s$ and $t=45s$.

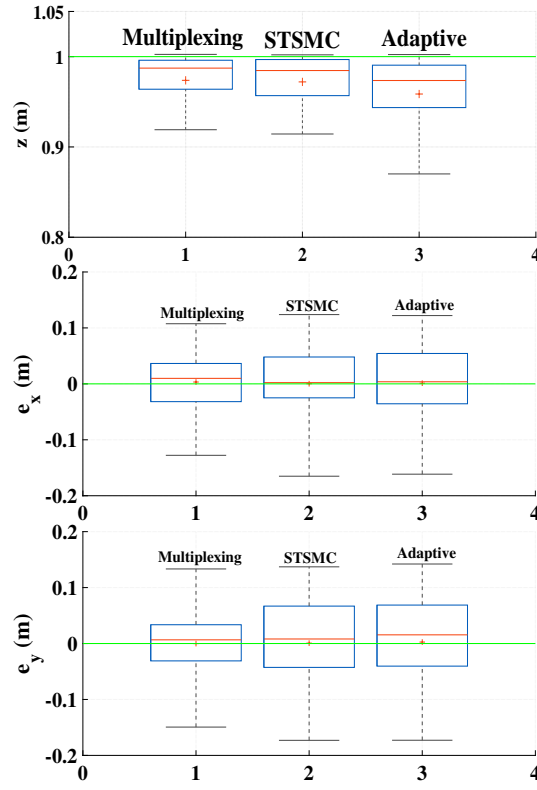


Figure 14: Box and whisker plot for means of each FTC scheme trials after four successive failures in square trajectory flight, showing the altitude z (m), and position errors e_x (m) and e_y (m) between $t=17$ s and $t=45$ s.

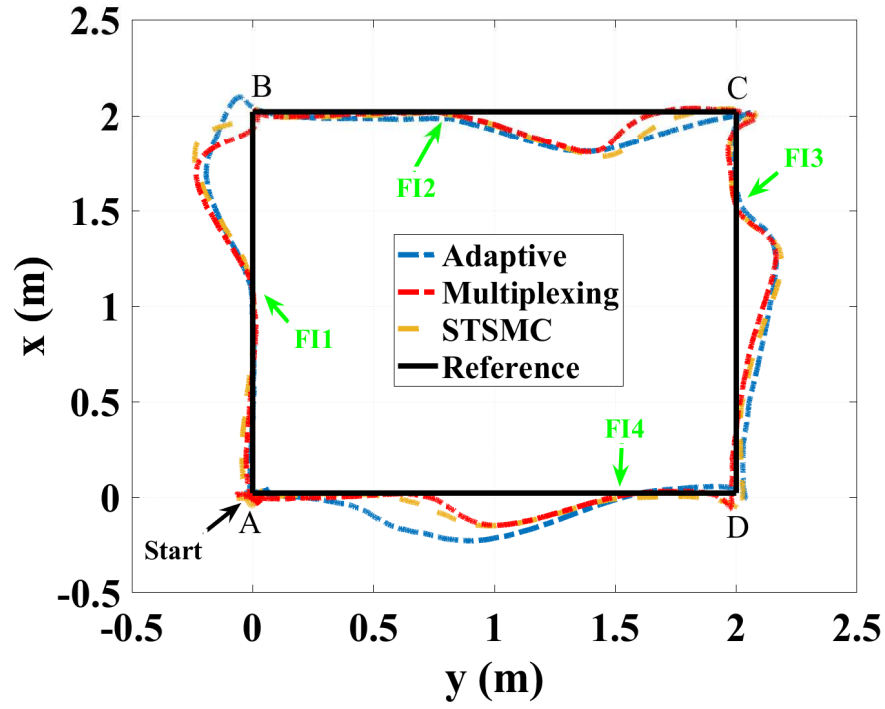


Figure 15: Behavior of 10 trials means of Adaptive, STSMC, and Multiplexing methods after four successive failures injection to the top motors in square trajectory flight in ABCDA direction

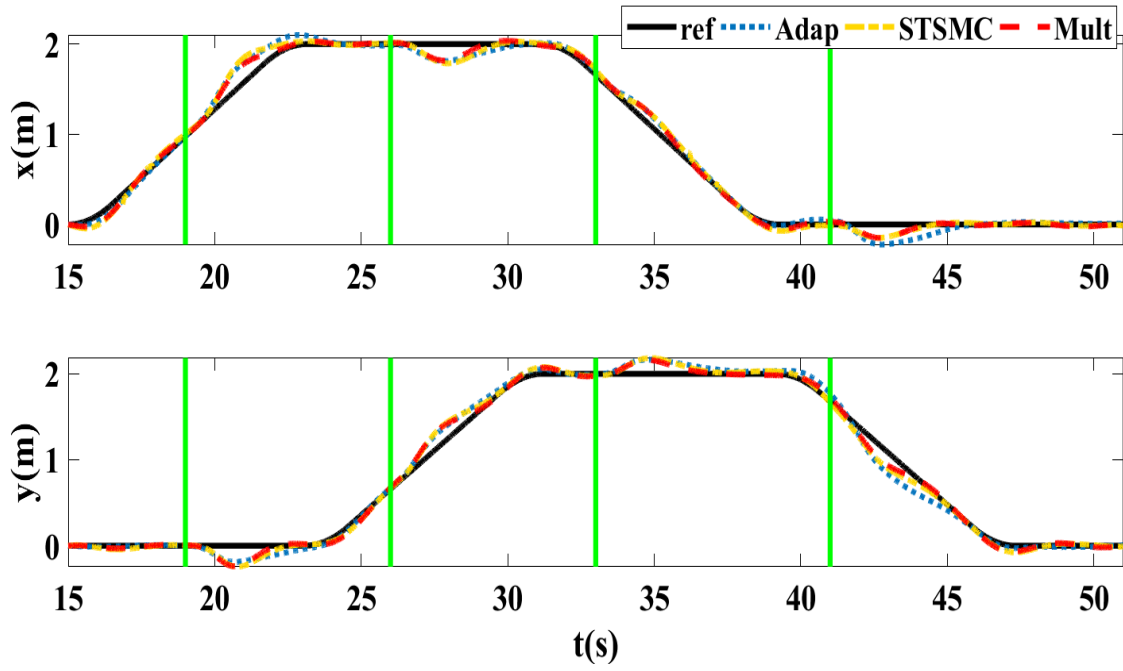


Figure 16: Evolution of x and y positions with respect to time of 10 trials means of Adaptive, STSMC, and Multiplexing methods after four successive failures injection to the top motors in square trajectory flight in ABCDA direction

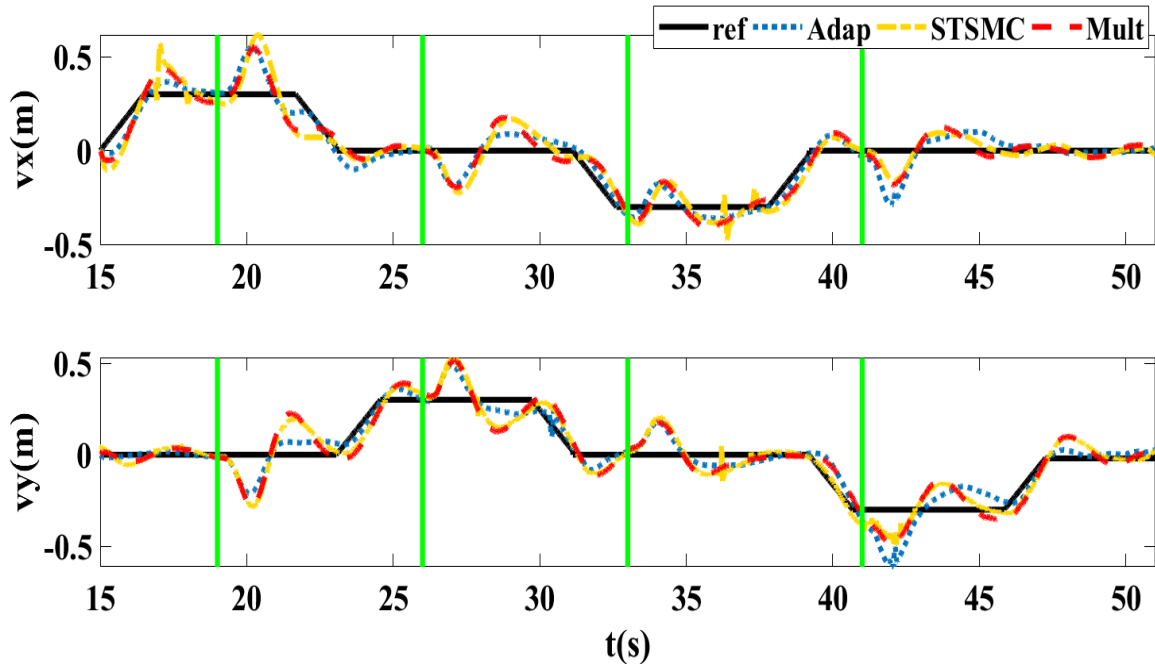


Figure 17: Evolution of x and y translational velocities with respect to time of 10 trials means of Adaptive, STSMC, and Multiplexing methods after four successive failures injection to the top motors in square trajectory flight in ABCDA direction

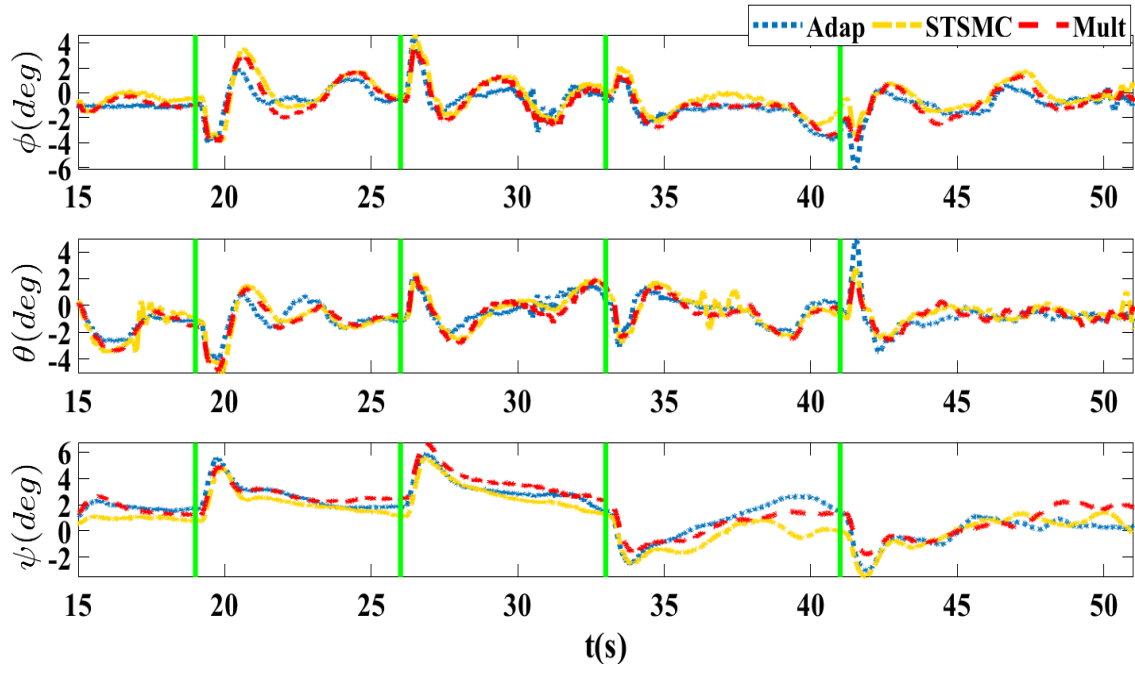


Figure 18: Evolution of ϕ , θ and ψ with respect to time of 10 trials means of Adaptive, STSMC, and Multiplexing methods after four successive failures injection to the top motors in square trajectory flight in ABCDA direction

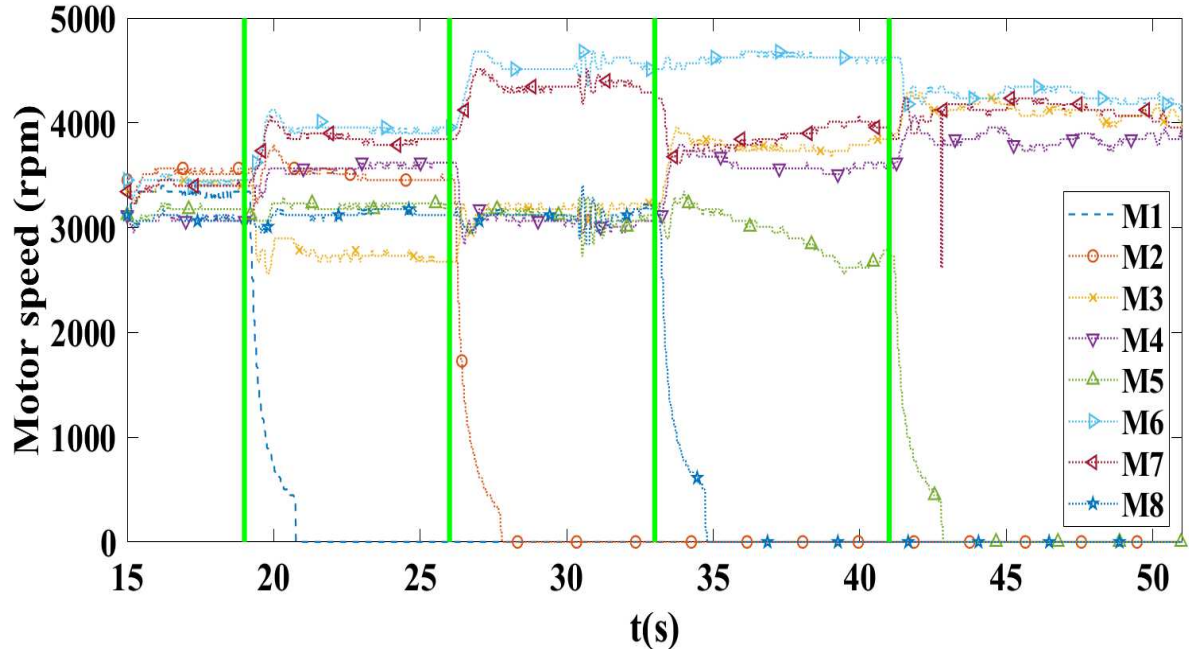


Figure 19: Evolution of motor speeds with respect to time of 10 trials means of Adaptive method after four successive failures injection to the top motors in square trajectory flight in ABCDA direction

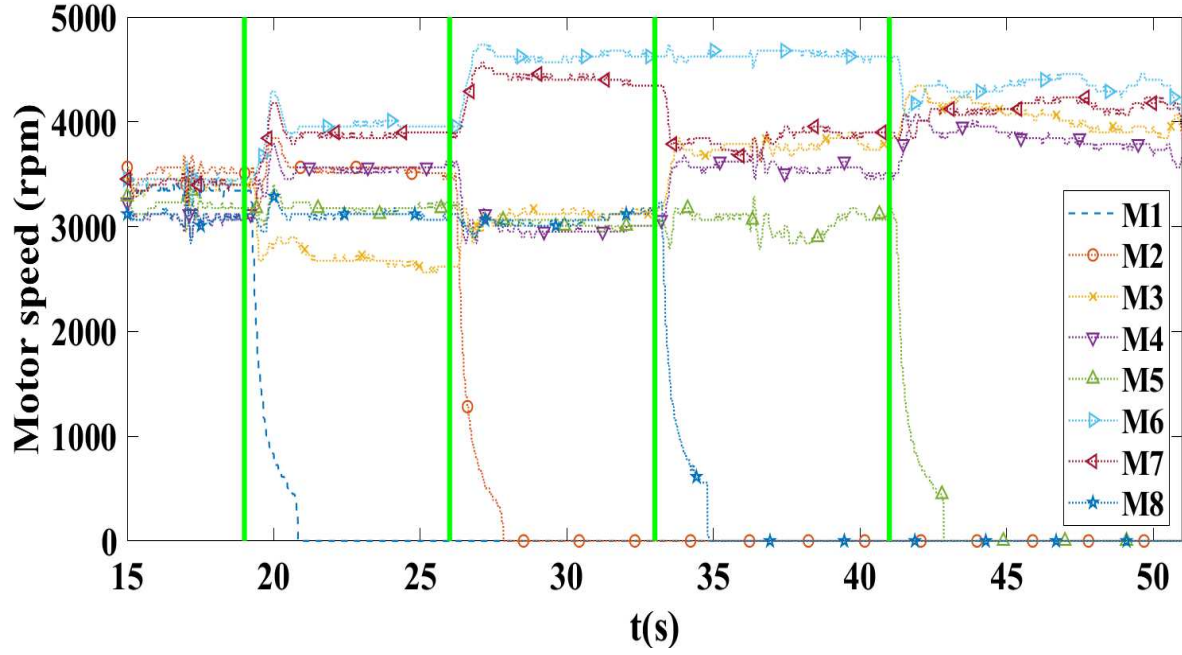


Figure 20: Evolution of motor speeds with respect to time of 10 trials means of STSMC method after four successive failures injection to the top motors in square trajectory flight in ABCDA direction

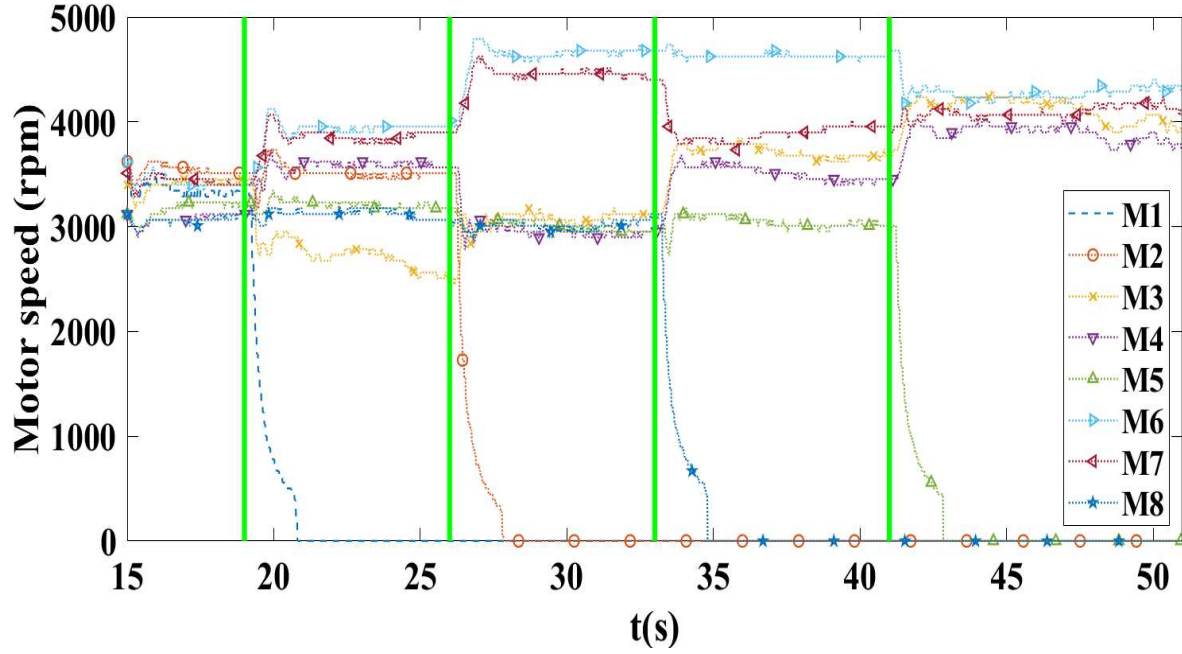


Figure 21: Evolution of motor speeds with respect to time of 10 trials means of Multiplexing method after four successive failures injection to the top motors in square trajectory flight in ABCDA direction

References

- [1] A. Avizienis, J.-C. Laprie, B. Randell, C. Landwehr, Basic concepts and taxonomy of dependable and secure computing, IEEE Transactions on dependable and secure computing 1 (2004) 11–33.
- [2] Y. Zhang, J. Jiang, Bibliographical review on reconfigurable fault-tolerant control systems, Annual Reviews in Control 32 (2) (2008) 229 – 252.

- [3] R. Amin, L. Aijun, S. Shamshirband, A review of quadrotor uav: control methodologies and performance evaluation, *International Journal of Automation and Control* 10 (2) (2016) 87–103.
- [4] A. Chamseddine, D. Theilliol, Y. Zhang, C. Join, C.-A. Rabbath, Active fault-tolerant control system design with trajectory re-planning against actuator faults and saturation: Application to a quadrotor unmanned aerial vehicle, *International Journal of Adaptive Control and Signal Processing* 29 (1) (2015) 1–23.
- [5] M. Ranjbaran, K. Khorasani, Fault recovery of an under-actuated quadrotor aerial vehicle, in: 49th IEEE Conference on Decision and Control (CDC), IEEE, 2010, pp. 4385–4392.
- [6] R. C. Avram, X. Zhang, J. Muse, Nonlinear adaptive fault-tolerant quadrotor altitude and attitude tracking with multiple actuator faults, *IEEE Transactions on Control Systems Technology* 26 (2018) 701–707.
- [7] H. Wang, Y. Zhang, Y. Yi, J. Xin, D. Liu, Nonlinear tracking control methods applied to qball-x4 quadrotor uav against actuator faults, in: IEEE Chinese Control and Decision Conference (CCDC), 2016, pp. 3478–3483.
- [8] Z. Song, K. Sun, Attitude tracking control of a quad-rotor with partial loss of rotation effectiveness, *Asian Journal of Control* 19 (2017) 1812–1821.
- [9] A. Lanzon, A. Freddi, S. Longhi, Flight control of a quadrotor vehicle subsequent to a rotor failure, *Journal of Guidance, Control, and Dynamics* 37 (2014) 580–591.
- [10] M. W. Mueller, R. D’Andrea, Stability and control of a quadrocopter despite the complete loss of one, two, or three propellers, in: IEEE International Conference on Robotics and Automation (ICRA), 2014, pp. 45–52.
- [11] P. Lu, E.-J. van Kampen, Active fault-tolerant control for quadrotors subjected to a complete rotor failure, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2015, pp. 4698–4703.
- [12] H. Okazaki, S. Yin, K. Isogai, H. Nakano, Motor speed control signals for multirotor flights in the presence of complete propeller motor failures, in: IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS), 2018, pp. 384–387.
- [13] T. Schneider, G. Ducard, K. Rudin, P. Strupler, Fault-tolerant control allocation for multirotor helicopters using parametric programming, *International Micro Air Vehicle Conference* (2012).
- [14] M. Saied, H. Shraim, C. Francis, I. Fantoni, B. Lussier, Controllability analysis and motors failures symmetry in a coaxial octorotor, in: IEEE Third International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAECE), 2015, pp. 245–250.
- [15] B. Wang, Y. Zhang, J.-C. Ponsart, D. Theilliol, Fault-tolerant adaptive control allocation for unmanned multi-rotor helicopter, *Elsevier, IFAC-PapersOnLine* 50 (2017) 5269–5274.
- [16] R. J. Patton, *Fault-Tolerant Control*, Springer London, London, 2015, pp. 422–428.
- [17] M. Saied, H. Shraim, B. Lussier, I. Fantoni, C. Francis, Local controllability and attitude stabilization of multirotor uavs: Validation on a coaxial octorotor, *Elsevier, Robotics and Autonomous Systems* 91 (2017) 128–138.
- [18] Y. Zhang, V. S. Suresh, B. Jiang, D. Theilliol, Reconfigurable control allocation against aircraft control effector failures, in: IEEE International Conference on Control Applications (CCA), 2007, pp. 1197–1202.
- [19] J. Lee, D. Shin, H. Ryu, D. Lee, D. H. Shim, Fault tolerant adaptive control using time delay control scheme under motor faults of octocopter, in: IEEE 7th International Conference on Systems and Control (ICSC), 2018, pp. 123–128.
- [20] S. Zeghlache, H. Mekki, A. Bouguerra, A. Djerioui, Actuator fault tolerant control using adaptive rbfn fuzzy sliding mode controller for coaxial octorotor uav, *Elsevier ISA Transactions* 80 (2018) 267–278.
- [21] H. Alwi, C. Edwards, Sliding mode fault-tolerant control of an octorotor using linear parameter varying-based schemes, *IET Control Theory & Applications* 9 (2015) 618–636.
- [22] S. Zeghlache, D. Saigaa, K. Kara, Fault tolerant control based on neural network interval type-2 fuzzy sliding mode controller for octorotor uav, *Springer, Frontiers of Computer Science* 10 (2016) 657–672.

- [23] M. Mehndiratta, E. Kayacan, Reconfigurable fault-tolerant nmpc for y6 coaxial tricopter with complete loss of one rotor, in: IEEE Conference on Control Technology and Applications (CCTA), 2018, pp. 774–780.
- 565 [24] A.-R. Merheb, H. Noura, F. Bateman, Passive and active fault tolerant control of octorotor uav using second order sliding mode control, in: IEEE Conference on Control Applications (CCA), 2015, pp. 1907–1912.
- [25] Y. Yu, Y. Dong, Global fault-tolerant control of underactuated aerial vehicles with redundant actuators, International Journal of Aerospace Engineering 2019 (2019).
- 570 [26] Y. Yu, X. Ding, A global tracking controller for underactuated aerial vehicles: design, analysis, and experimental tests on quadrotor, IEEE/ASME Transactions on Mechatronics 21 (5) (2016) 2499–2511.
- [27] A. Baldini, R. Felicetti, A. Freddi, S. Longhi, A. Monteriù, Octarotor fault tolerant control via dynamic surface control, in: IEEE 18th European Control Conference (ECC), 2019, pp. 3892–3897.
- [28] T. A. Johansen, T. I. Fossen, Control allocation—a survey, Automatica 49 (5) (2013) 1087–1103.
- 575 [29] R. Chen, Z. Zhang, P. Zhang, M. Mangold, Fault tolerant control for hexacopter with reducing yaw rate, in: IEEE 4th Conference on Control and Fault Tolerant Systems (SysTol), 2019, pp. 171–176.
- [30] N. P. Nguyen, N. X. Mung, S. K. Hong, Actuator fault detection and fault-tolerant control for hexacopter, Sensors 19 (21) (2019) 4721.
- 580 [31] M. Saied, B. Lussier, I. Fantoni, C. Francis, H. Shraim, Fault tolerant control for multiple successive failures in an octorotor: Architecture and experiments, in: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015, pp. 40–45.
- [32] M. Saied, B. Lussier, I. Fantoni, C. Francis, H. Shraim, G. Sanahuja, Fault diagnosis and fault-tolerant control strategy for rotor failure in an octorotor, in: IEEE International Conference on Robotics and Automation (ICRA), 2015, pp. 5266–5271.
- 585 [33] M. Saied, Fault-tolerant control of an octorotor unmanned aerial vehicle under actuators failures, Ph.D. thesis, Université de Technologie de Compiègne (2016).
- [34] H. Hamadi, B. Lussier, I. Fantoni, C. Francis, H. Shraim, Observer-based super twisting controller robust to wind perturbation for multirotor uav, in: International Conference on Unmanned Aircraft Systems (ICUAS), 2019.
- 590 [35] J. Geng, Y. Wang, A finite-time sliding-mode controller design for convergence at a desired time and global robust control, in: IEEE 35th Chinese Control Conference (CCC), 2016, pp. 550–555.