

## HEALTH-AWARE AND FAULT-TOLERANT CONTROL OF AN OCTOROTOR UAV SYSTEM BASED ON ACTUATOR RELIABILITY

JEAN CARLO SALAZAR <sup>a,\*</sup>, ADRIÁN SANJUAN <sup>a</sup>, FATIHA NEJJARI <sup>a</sup>, RAMON SARRATE <sup>a</sup>

<sup>a</sup>Polytechnic University of Catalonia (UPC)  
Research Center for Supervision, Safety and Automatic Control (CS2AC)  
10, Rambla Sant Nebridi, Terrassa, Spain  
e-mail: {jean.salazar, adrian.sanjuan}@upc.edu,  
{fatiha.nejjari, ramon.sarrate}@upc.edu

A major goal in modern flight control systems is the need for improving reliability. This work presents a health-aware and fault-tolerant control approach for an octorotor UAV that allows distributing the control effort among the available actuators based on their health information. However, it is worth mentioning that, in the case of actuator fault occurrence, a reliability improvement can come into conflict with UAV controllability. Therefore, system reliability sensitivity is redefined and modified to prevent uncontrollable situations during the UAV's mission. The priority given to each actuator is related to its importance in system reliability. Moreover, the proposed approach can reconfigure the controller to compensate actuator faults and improve the overall system reliability or delay maintenance tasks.

**Keywords:** prognostics and health management, health-aware control, fault-tolerant control, reliability analysis, octorotor, UAV.

### 1. Introduction

Unmanned aerial vehicles (UAVs) are well-suited to a wide range of mission scenarios, such as search, rescue, supervision and inspection, among others. Redundancy in sensors, actuators and all other essential components of UAVs plays a vital role in increasing flight safety and mission accomplishment in the case of degradation or fault/failure occurrence in those components.

In particular, the use of a multicopter with eight actuators (octocopter), instead of using a quadcopter, makes the UAV capable of maintaining normal flight and accomplishing the mission despite the occurrence of failures in one or more rotors/propellers.

On the one hand, several reconfiguration control techniques, such as gain-scheduled PID (Milhim *et al.*, 2010) or sliding mode control (SMC) (Alwi and Edwards, 2006; Merheb *et al.*, 2015) have been applied to compensate the fault effect. On the other hand, techniques which hide the fault from the controller point of view have also been used in fault-tolerant control (FTC) when the system has actuator redundancy. In the work of Rotondo

*et al.* (2015), an FTC into a robust linear parameter varying (LPV) polytopic framework was proposed. A review of FTC techniques for quadrotors is presented by Zhang *et al.* (2013). In the work of Cen *et al.* (2015), an active and passive FTC scheme based on fault estimation for a quadrotor actuator is proposed. Another popular technique is control allocation (Johansen and Fossen, 2013).

However, to improve system performance during UAVs mission operations, it could be more appropriate to avoid fault occurrence than tolerate them. In this sense, a new paradigm in which the use of both control and reliability theories has emerged in terms of health-aware control (HAC). The reliability of UAVs and their components against faults and failures is one of the most important objectives for the safety of critical systems.

The aim of an HAC system is to modify the control actions based on system reliability information provided by a proper online prognostics tool even in the presence of actuator/sensor faults. This leads to an increase in the operation time of the system (Salazar *et al.*, 2015; 2016; Khelassi *et al.*, 2011).

---

\*Corresponding author

The control of multirotor systems and particularly the octorotor is a topic of interest and a challenging problem due to their under-actuated nature and nonlinear dynamics. Several control techniques have been used, such as model predictive control (MPC) (Raffo *et al.*, 2010; Liu *et al.*, 2012; Abdolhosseini *et al.*, 2013), PID (Rinaldi *et al.*, 2014), or LQR (Marks *et al.*, 2012; Adîr and Stoica, 2012).

Thanks to its actuator redundancy, the octorotor has the potential to improve safety and reliability. Several approaches have been proposed and used to manage the redundancy of this kind of systems by distributing the control effort among a set of rotors. One approach to achieve this goal consists in using an optimal control design to shape in one step the closed-loop dynamics and actuator control distribution. Another technique is separating the control task from the effort distribution one. In this case, a separate control allocation module is introduced in the closed loop to distribute the control effort among the actuators (Khelassi *et al.*, 2011; Durham, 1993; Bodson, 2002; Salazar *et al.*, 2015; 2017). The control allocation approach allows the faulty system to be accommodated without the need to modify the controller.

This work presents the advantages of component and system reliability integration into the control by a control allocation scheme. The proposed approach is applied to a multirotor UAV system. Such kind of systems has actuators redundancy which allows the design of controllers that can optimize the distribution of the control effort in such a way that the reliability of the system is preserved or even extended. The objective is to combine a deterministic part related to system dynamics and a stochastic part related to system reliability. The resulting scheme provides control performance and preserves system reliability in faulty and non-faulty scenarios.

The proposed approach can also be used to schedule maintenance tasks based on the reliability of the actuators. This ability is very useful in the case of systems whose maintenance actions represents a costly process either by their downtime cost or their economic cost in components, or labor cost, among others.

The case study is an octorotor UAV system that has eight propellers in the configuration I (Fig. 1). Four propellers can rotate in a clockwise direction, while the remaining ones can rotate anti-clockwise. The octorotor is moved by changing the rotor speeds. For example, by increasing or decreasing together the speeds of the eight propellers, a vertical motion is achieved. Changing only the speeds of the propellers situated at the opposite locations produces either roll or pitch rotation, coupled with the corresponding lateral motion. Finally, yaw rotation results from the difference in the counter-torque between each pair of propellers.

Additionally, the octorotor has actuator redundancy that can offer a degree of over-actuation as it can work

with at least four propellers forming a quadrotor structure.

The paper is organized as follows. The dynamic modeling of the octorotor UAV system is detailed in Section 2. In Section 3, the controller design of the UAV system is presented. Section 4 addresses the problem of control effort redistribution. In Section 5, reliability modeling is presented. The HAC scheme is presented in Section 6. The simulation results are presented in Section 7 and, finally, some conclusions are given.

## 2. Octorotor dynamics

To describe the dynamics of a multirotor, it is necessary to define two frames in which it will operate: the inertial frame and the body frame. The former  $\{\mathbf{I}\}$  is static and represents the reference of the multirotor while the latter  $\{\mathbf{B}\}$  is defined by the orientation of the multirotor and it is situated in its center of mass. The two frames are related by the rotation matrix (1).  $\mathbf{R}_B^I$  transforms a vector in body reference to a vector in inertial reference. In this case, the Euler angles, namely, the roll angle ( $\phi$ ), the pitch angle ( $\theta$ ) and the yaw angle ( $\psi$ ), are used to model this rotation following the sequence  $z - y - x$ :

$$\mathbf{R}_B^I = \begin{bmatrix} c(\psi)c(\theta) & c(\psi)s(\theta)s(\phi) - s(\psi)c(\phi) \\ s(\psi)c(\theta) & s(\psi)s(\theta)s(\phi) + c(\psi)c(\phi) \\ -s(\theta) & c(\theta)s(\phi) \\ & c(\psi)s(\theta)c(\phi) + s(\psi)s(\phi) \\ & s(\psi)s(\theta)c(\phi) - c(\psi)s(\phi) \\ & c(\theta)c(\phi) \end{bmatrix}, \quad (1)$$

where  $s(\cdot)$  and  $c(\cdot)$  denote  $\sin(\cdot)$  and  $\cos(\cdot)$ , respectively. A similar notation will be used for  $\tan(\cdot)$  which will be denoted by  $t(\cdot)$ .

The dynamics of a multirotor can be defined using the Newton and Euler equations (Mahony *et al.*, 2012) describing the translation and rotation of the rigid body:

$$\dot{\boldsymbol{\xi}}_I = \mathbf{v}_I, \quad (2)$$

$$\dot{\mathbf{v}}_I = \frac{1}{m}(\mathbf{f}_I), \quad (3)$$

$$\dot{\boldsymbol{\eta}}_I = \mathbf{W}_\eta \boldsymbol{\omega}_B, \quad (4)$$

$$\dot{\boldsymbol{\omega}}_B = \frac{1}{\mathbf{J}}(\boldsymbol{\tau}_B - \boldsymbol{\omega}_B \times \mathbf{J}\boldsymbol{\omega}_B), \quad (5)$$

where  $\mathbf{v}_I = [v_x \ v_y \ v_z]^T$  is the linear speed vector in the inertial frame,  $\boldsymbol{\xi}_I = [x \ y \ z]^T$  is the position vector in the inertial frame,  $\boldsymbol{\eta} = [\phi \ \theta \ \psi]^T$  is the orientation vector,  $\boldsymbol{\omega}_B = [p \ q \ r]^T$  is the body angular velocity vector,  $m$  is the mass of the vehicle,  $\mathbf{J}$  is the inertia tensor,  $\mathbf{f}_I$  and  $\boldsymbol{\tau}_B$  represent the external forces and torques applied to the UAV and  $\mathbf{W}_\eta$ , which represents the transformation matrix of angular velocities from the body frame to the

inertial frame, given by (Blakelock, 1991)

$$\mathbf{W}_\eta = \begin{bmatrix} 1 & s(\phi)t(\theta) & c(\phi)t(\theta) \\ 0 & c(\phi) & -s(\phi) \\ 0 & \frac{s(\phi)}{c(\theta)} & \frac{c(\phi)}{c(\theta)} \end{bmatrix}. \quad (6)$$

The external forces interacting with the vehicle are the lift of the rotors ( $T$ ), the translational drag and the gravity ( $g$ ). The external torques are the torques generated by the motors ( $\tau_x$ ,  $\tau_y$  and  $\tau_z$ ) and the rotational drag.

The model has been developed under the following assumptions (Freddi *et al.*, 2011):

- The structure of the UAV is symmetrical.
- The body and propellers are rigid.
- The free stream air velocity is zero.
- The motor dynamics can be neglected since they are relatively fast.
- The blade flexibility can be neglected considering that it is relatively small.
- The inertia tensor of the octorotor body is diagonal  $\mathbf{J} = \text{diag}(J_{xx}, J_{yy}, J_{zz})$ .
- The inertia of the octorotor body is much larger than that of the propeller (it includes the rotating parts of the rotor), so  $\mathbf{J} \gg \mathbf{J}_{\text{prop}}$ .
- Translational and rotational drags are negligible.

The multirotor UAV model is obtained by expanding Eqns. (2)–(5), doing the transformations from the body to the inertial frame (1), and applying the previous assumptions:

$$\dot{x}_I = v_x, \quad (7)$$

$$\dot{y}_I = v_y, \quad (8)$$

$$\dot{z}_I = v_z, \quad (9)$$

$$\dot{v}_x = \frac{1}{m} [c(\phi)s(\theta)c(\psi) + s(\phi)s(\psi)]T, \quad (10)$$

$$\dot{v}_y = \frac{1}{m} [c(\phi)s(\theta)s(\psi) - s(\phi)c(\psi)]T, \quad (11)$$

$$\dot{v}_z = \frac{1}{m} [c(\phi)c(\theta)]T - g, \quad (12)$$

$$\dot{\phi} = p + s(\phi)t(\theta)q + c(\phi)t(\theta)r, \quad (13)$$

$$\dot{\theta} = c(\phi)q - s(\phi)r, \quad (14)$$

$$\dot{\psi} = \frac{s(\phi)}{c(\theta)}q + \frac{c(\phi)}{c(\theta)}r, \quad (15)$$

$$\dot{p} = \frac{1}{J_{xx}} [-(J_{zz} - J_{yy})qr - J_p q \Omega_p + \tau_x], \quad (16)$$

$$\dot{q} = \frac{1}{J_{yy}} [(J_{zz} - J_{xx})pr + J_p p \Omega_p + \tau_y], \quad (17)$$

$$\dot{r} = \frac{1}{J_{zz}} [-(J_{yy} - J_{xx})pq + \tau_z], \quad (18)$$

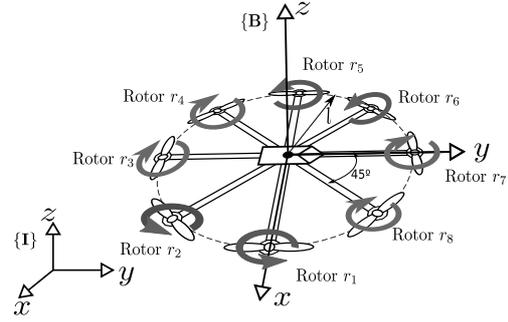


Fig. 1. Octorotor PPNNPPNN structure.

Table 1. Parameter values.

Parameter	Symbol	Value
Body inertia	$J_{xx} = J_{yy}$	$150 \cdot 10^{-3} \text{ [kg}\cdot\text{m}^2]$
Body inertia	$J_{zz}$	$400 \cdot 10^{-3} \text{ [kg}\cdot\text{m}^2]$
Propeller inertia	$J_p$	$104 \cdot 10^{-6} \text{ [kg}\cdot\text{m}^2]$
Mass	$m$	8 [kg]
Arm length	$l$	0.4 [m]
Thrust factor	$k_b$	$54.2 \cdot 10^{-6} \text{ [N}\cdot\text{s}^2]$
Drag factor	$k_d$	$813 \cdot 10^{-9} \text{ [m]}$

where  $J_p$  is the inertia moment of the motor (rotating parts) and the propeller around the  $z$ -axis.

Then, for the PPNNPPNN octorotor structure, as the one presented in Fig. 1, where  $P$  and  $N$  define positive and negative reactive motor torques, respectively, (represented as arrows in Fig. 1),

$$\Omega_p = -|\Omega_1| - |\Omega_2| + |\Omega_3| + |\Omega_4| - |\Omega_5| - |\Omega_6| + |\Omega_7| + |\Omega_8|, \quad (19)$$

where  $\Omega_i$  is the angular velocity of the  $i$ -th motor. It is assumed that the thrust force of a propeller is proportional to the angular velocity squared, such that  $u_{T_i} = k_b \Omega_i^2$ , where  $u_{T_i}$  is the thrust generated by motor  $i$  and  $k_b$  is the trust factor. The parameter values which define the octorotor model are presented in Table 1.

Equations (7)–(18) describe the model for a generic multirotor structure where the control action is given by  $\mathbf{u}_v = [T \ \tau_x \ \tau_y \ \tau_z]^T$ .

Generally,  $\mathbf{u}_v$  is referred to as the virtual control action due to the fact that no physical actuators correspond to the forces and momentums which it describes. In this work, the control allocation block transforms and redistributes the control effort ( $\mathbf{u}_v$ ) generated by the inner control loop into the thrust that each motor should deliver ( $\mathbf{u}_T$ ) as presented in Section 4.

### 3. Controller structure

In this work, the control of the UAV is a cascade structure (Fig. 2), where two LQR controllers, inner and outer, define the dynamics of the octorotor, and the

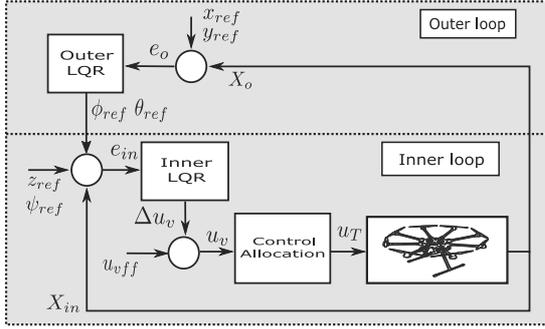


Fig. 2. Control scheme.

control allocation distributes the control effort among the actuators.

The linearized model needed for the design of the LQR controller is obtained by applying the Taylor series approximation at the hover position, which corresponds to the situation where the  $x$ - $y$  planes of both frames ( $\{I\}$  and  $\{B\}$ ) (Fig. 1) are parallel and the motors generate a lifting force equal to the weight of the octorotor. The state and input are considered to be  $\mathbf{x} = [x \ y \ z \ \phi \ \psi \ v_x \ v_y \ v_z \ p \ q \ r]^T$  and  $\mathbf{u}_v = [T \ \tau_x \ \tau_y \ \tau_z]^T$ , respectively. The LQR controller is designed following the methodology proposed by Ogata (1995).

**3.1. Cascade structure controller.** The inner and outer LQR linearized models are detailed below.

The inner loop focuses on the orientation and the altitude of the UAV and is defined as

$$\begin{aligned} \dot{\mathbf{e}}_{in}(t) &= \mathbf{A}_{in}\mathbf{e}_{in}(t) + \mathbf{B}_{in}\Delta\mathbf{u}_v(t), \\ \dot{\mathbf{e}}_{in}(t) &= \begin{bmatrix} \mathbf{0}_{4 \times 4} & \mathbf{I}_{4 \times 4} \\ \mathbf{0}_{4 \times 4} & \mathbf{0}_{4 \times 4} \end{bmatrix} \mathbf{e}_{in}(t) + \begin{bmatrix} \mathbf{0}_{4 \times 4} \\ \boldsymbol{\beta}_{in} \end{bmatrix} \Delta\mathbf{u}_v(t), \end{aligned} \quad (20)$$

where  $\boldsymbol{\beta}_{in} = \text{diag}(1/m, 1/J_{xx}, 1/J_{yy}, 1/J_{zz})$ ,  $\mathbf{I}_{4 \times 4}$  is the identity matrix, and  $\mathbf{e}_{in}$  is the inner state vector:

$$\mathbf{e}_{in} = \mathbf{X}_{ref,in} - \mathbf{X}_{in} = [e_z \ e_\phi \ e_\theta \ e_\psi \ e_{v_z} \ e_p \ e_q \ e_r]^T, \quad (21)$$

where  $\mathbf{X}_{ref,in}$  denotes the altitude, the orientation, the linear velocity ( $z$ -axis) and the angular velocity set-points, and  $\Delta\mathbf{u}_v$  is the inner input vector denoted as

$$\Delta\mathbf{u}_v = [\Delta T \ \Delta\tau_x \ \Delta\tau_y \ \Delta\tau_z]^T, \quad (22)$$

with  $\Delta\mathbf{u}_v = \mathbf{u}_v - \mathbf{u}_{vff} = \mathbf{u}_v - [mg \ 0 \ 0 \ 0]^T$ , and  $\mathbf{u}_{vff}$  is the feed-forward virtual input that keeps the system at the equilibrium point (hover position) in open loop.

The outer loop focuses on the position of the UAV

( $x$ - $y$  plane). Its model is as follows:

$$\begin{aligned} \dot{\mathbf{e}}_o(t) &= \mathbf{A}_o\mathbf{e}_o(t) + \mathbf{B}_o\Delta\mathbf{u}_o(t), \\ \dot{\mathbf{e}}_o(t) &= \begin{bmatrix} \mathbf{0}_{2 \times 2} & \mathbf{I}_{2 \times 2} \\ \mathbf{0}_{2 \times 2} & \mathbf{0}_{2 \times 2} \end{bmatrix} \mathbf{e}_o(t) + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & g \\ -g & 0 \end{bmatrix} \Delta\mathbf{u}_o(t), \end{aligned} \quad (23)$$

where  $\mathbf{e}_o$  is the outer state vector denoted as

$$\mathbf{e}_o = \mathbf{X}_{ref,o} - \mathbf{X}_o = [e_x \ e_y \ e_{v_x} \ e_{v_y}]^T, \quad (24)$$

where  $\mathbf{X}_{ref,o}$  is the position and linear velocity set-point ( $x$ - $y$  plane), and  $\Delta\mathbf{u}_o$  is the outer input vector denoted as

$$\Delta\mathbf{u}_o = [\Delta\phi \ \Delta\theta]^T = [\phi_{ref} \ \theta_{ref}]^T. \quad (25)$$

#### 4. Control allocation loop

The control distribution task will be provided by the control allocation block. It consists in distributing the control effort generated in the inner loop ( $\mathbf{u}_v$ ) among the actuators ( $\mathbf{u}_T$ ) available in the system (Johansen and Fossen, 2013).

Taking into account that the virtual control actions are always in the attainable control set (ACS) (Schneider et al., 2012) of the actuators, the most extended solution is to use the generalized inverse, which is described as follows:

$$\begin{aligned} \min_{\mathbf{u}_T \in \mathbb{R}} \quad & \frac{1}{2}(\mathbf{u}_T)^T \mathbf{W}(\mathbf{u}_T), \\ \text{subject to} \quad & \mathbf{u}_v = \mathbf{B}_{CA}\mathbf{u}_T. \end{aligned} \quad (26)$$

If  $\mathbf{B}_{CA}$  has full rank, then this weighted least squares problem has the following explicit solution (Johansen and Fossen, 2013):

$$\mathbf{u}_T = \mathbf{W}^{inv}\mathbf{B}_{CA}^T(\mathbf{B}_{CA}\mathbf{W}^{inv}\mathbf{B}_{CA}^T)^{-1}\mathbf{u}_v, \quad (27)$$

where  $\mathbf{W}^{inv} = \mathbf{W}^{-1}$ , and  $\mathbf{B}_{CA}$  is a function of the structural matrix of the PPNNPPNN octorotor ( $\mathbf{B}_{str}$ ):

$$\mathbf{u}_v = \mathbf{B}_{CA}\mathbf{u}_T = \mathbf{B}_{str}\boldsymbol{\Xi}\mathbf{u}_T, \quad (28)$$

where  $\boldsymbol{\Xi} = \text{diag}(\xi_1, \xi_2, \dots, \xi_8)$  denotes the control effectiveness matrix ( $\xi_i = 1$  defines the nominal behavior and a complete failure corresponds to  $\xi_i = 0$ ) and  $\mathbf{B}_{str}$  is the matrix that contains the relation between the thrust generated by the actuators and the virtual actions,

$$\begin{aligned} \mathbf{B}_{str} &= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -ls(45^\circ) & -l & -ls(45^\circ) \\ -l & -lc(45^\circ) & 0 & +lc(45^\circ) \\ +k_d/k_b & +k_d/k_b & -k_d/k_b & -k_d/k_b \end{bmatrix} \end{aligned}$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & ls(45^\circ) & l & ls(45^\circ) \\ l & lc(45^\circ) & 0 & -lc(45^\circ) \\ +k_d/k_b & +k_d/k_b & -k_d/k_b & -k_d/k_b \end{bmatrix}, \quad (29)$$

where  $k_d$  is the motor drag factor and  $l$  is the distance between the center of mass to the rotor center.

Note that, the higher the diagonal component of the matrix  $\mathbf{W}^{inv}$ , the greater the corresponding actuator usage, and vice-versa. In this work, the control inputs are computed based on the redistributed pseudo-inverse (RPI) (Johansen and Fossen, 2013) and on the cascaded generalized inverse (CGI) (Bordignon and Durham, 1995) through Algorithm 1.

It involves the partition of the virtual control vector ( $\mathbf{u}_v$ ) into  $n$  subvectors (Steps 1–2):

$$\mathbf{u}_v = \sum_{i=1}^n \mathbf{u}_{v_i}, \quad (30)$$

where  $\mathbf{u}_{v_i}$  is a subvector that contains a group of virtual actions with comparable characteristics or relevance. In this work, these variables will be clustered depending on how they affect to the actuator degradation.

Next, a matrix  $\mathbf{W}_i^{inv}$  is defined for each  $\mathbf{u}_{v_i}$  applying given criteria (Step 4).

Then, the generalized inverse (31) is applied to compute  $\mathbf{u}_{T_i}$  (Steps 5–7):

$$\mathbf{u}_{T_i} = \mathbf{W}_i^{inv} \mathbf{B}_{CA}^T (\mathbf{B}_{CA} \mathbf{W}_i^{inv} \mathbf{B}_{CA}^T)^{-1} \mathbf{u}_{v_i}, \quad (31)$$

and  $\mathbf{u}_T$  (Step 8):

$$\mathbf{u}_T = \sum_{i=1}^n \mathbf{u}_{T_i}. \quad (32)$$

Algorithm 1 stops once  $\mathbf{u}_T$  is between the limits of the actuator (Step 9):

$$\underline{\mathbf{u}}_T < \mathbf{u}_T < \bar{\mathbf{u}}_T. \quad (33)$$

If this is not satisfied, return to Step 4 and redefine the weights.

Note that, contrary to what is done in the RPI methodology where saturated actuators are neglected, in the proposed approach, if the condition (33) is not satisfied, the weights must be redefined following alternative criteria. Also, as long as a solution exists for (26), the following choice will provide a feasible solution:  $\mathbf{W}_i^{inv} = \mathbf{I}, \forall i \in [1, n]$ , with  $\mathbf{I} \in \mathbb{R}^{8 \times 8}$  being the identity matrix.

Therefore, the proposed control allocation scheme tries to distribute the control actions depending on the system reliability ( $\mathbf{W}_i^{inv}$ ) and the actuator faults ( $\Xi$ ).

---

**Algorithm 1.** Distribution of the control effort.

---

- 1: Given  $\mathbf{u}_v$
  - 2: Cluster  $\mathbf{u}_v$  into  $n$  subvectors ( $\mathbf{u}_{v_i}$ ) (30)
  - 3: **repeat**
  - 4:   Define  $\mathbf{W}_i^{inv}$  following given criteria  $i$
  - 5:   **for**  $i = 1$  **to**  $n$  **do**
  - 6:     Compute  $\mathbf{u}_{T_i}$  as (31)
  - 7:   **end for**
  - 8:   Compute  $\mathbf{u}_T$  as (32)
  - 9: **until**  $\{\mathbf{u}_T > \underline{\mathbf{u}}_T \text{ and } \mathbf{u}_T < \bar{\mathbf{u}}_T\}$
- 

## 5. Reliability modeling

Reliability is defined as the probability that components, units, equipment, and systems will perform satisfactorily for a specified period of time under specified operating conditions and environments (Gertsbakh, 2001).

In particular, the reliability of the  $i$ -th component of a system can be expressed as

$$R_i(t) = R_i^0 e^{-\int_0^t \lambda_i(\tau) d\tau}, \quad \forall i = 1, \dots, m, \quad (34)$$

where  $R_i^0$  is the initial reliability at the mission initial time and  $\lambda_i(t)$  is the failure rate of the  $i$ -th actuator, respectively.

Several definitions of the failure rate can be found in the literature. In this work, a definition based on the proportional hazard proposed by Cox (1972) is used,

$$\lambda_i(t) = \lambda_i^0 \cdot g_i(\ell, \vartheta), \quad \forall i = 1, \dots, m, \quad (35)$$

where  $\lambda_i^0$  represents the nominal failure rate of the  $i$ -th component and  $g_i(\ell, \vartheta)$  is a load function also known as a covariate, which represents the effect of stress on the component failure rate as a function of the employed load ( $\ell$ ) and a component parameter ( $\vartheta$ ).

Different definitions of  $g_i(\ell, \vartheta)$  exist in the literature. For example, in the work of Khelassi *et al.* (2011), a load function based on the root-mean-square of the employed control input until the end of the mission ( $t_f$ ) was proposed. This load function is used to distribute the control efforts between the redundant actuators, and the control action is calculated using a reliable state feedback controller. Salazar *et al.* (2017) defined the covariate function as the cumulative control effort during the usage life of the  $i$ -th actuator.

In this work, the covariate of the  $i$ -th actuator is expressed through function  $g_i(u_i^{\text{norm}}(t), \dot{u}_i^{\text{norm}}(t))$  as

$$g_i(\cdot) = 1 + \beta_i \int_0^t u_i^{\text{norm}} dv + \gamma_i \int_0^t \dot{u}_i^{\text{norm}}(v) dv, \quad (36)$$

with  $g_i(\cdot)$  is defined as a cumulative function of the normalized control effort and its normalized derivative for the  $i$ -th actuator from the beginning of the mission up to

the current time  $t$ , and  $\beta_i$  and  $\gamma_i$  are constant parameters that describe the contribution of each term. Normalized values belong to the interval  $[0,1]$ , with 0 representing no contribution and 1 the maximum contribution to actuator degradation. Then, replacing (36) in (35) yields

$$\lambda_i(t) = \lambda_i^0 \left( 1 + \beta_i \int_0^t u_i^{\text{norm}} dv + \gamma_i \int_0^t \dot{u}_i^{\text{norm}}(v) dv \right). \quad (37)$$

In contrast to our previous work (Salazar et al., 2017), the failure rate (37) considered here includes a term related to control variations which represents the actuator degradation due to fatigue.

This definition implies that actuators are under a reliability decay due to the baseline failure rate which is increased when the actuators are used.

**5.1. System reliability.** The overall system reliability can be computed by means of its structure function. It expresses the state of the system in terms of the state of its components, allowing the computing of the system reliability. This system structure could be serial, parallel, or a mix of both. In complex structures (i.e., a bridge structure), the structure function can be computed following the pivotal decomposition method (Gertsbakh, 2001) or based on the set of minimal path sets or minimal cut sets. Alternatively, system reliability can be modeled using a dynamic Bayesian network (DBN) (Salazar et al., 2015).

In this work, octorotor motors are the only components that are considered in the system reliability computation. This computation is based on determining the minimal path sets.

Although the octorotor system has eight actuators ( $r_i \forall i \in [1, 8]$ ), control performance is guaranteed as long as, at least, certain quadrotor configurations are operational (Sanjuan et al., 2019) (see Fig. 4). Based on them, the following minimal path sets are defined:

$$\begin{aligned} \zeta_1 &: \{r_1, r_3, r_5, r_7\}, & \zeta_2 &: \{r_2, r_4, r_6, r_8\}, \\ \zeta_3 &: \{r_2, r_3, r_6, r_7\}, & \zeta_4 &: \{r_1, r_4, r_5, r_8\}. \end{aligned} \quad (38)$$

Given  $s$  minimal path sets, it is possible to compute the structure function as

$$\Phi = 1 - \prod_{j=1}^s \left( 1 - \prod_{i \in \zeta_j} X_i \right), \quad (39)$$

where  $X_i$  is a random binary variable representing the state of the  $i$ -th rotor (up or down). To compute the reliability of the system, the structure function expression must be previously expanded. Then, the expanded expression should be simplified taking into account that, due to its binary nature, any power of  $X_i$  is equal to  $X_i$ . The system reliability expression ( $R_S$ ) is obtained from

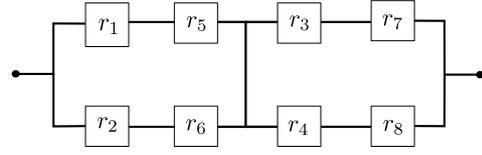


Fig. 3. System reliability block diagram.

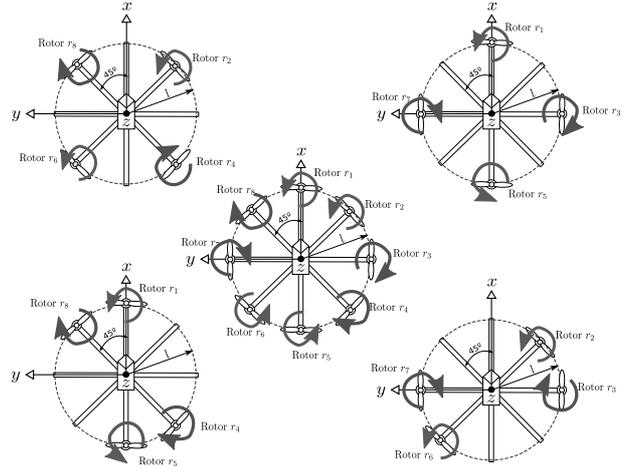


Fig. 4. Octorotor controllable configurations.

the resulting structure function expression by replacing  $X_i$  with the component reliability  $R_i$ .

Figure 3 represents the reliability block diagram of the octorotor system based on its minimal path sets ( $\zeta_i$ ). The resulting expression for the computation of its system reliability is

$$\begin{aligned} R_S &= R_1 R_3 R_5 R_7 + R_1 R_4 R_5 R_8 + R_2 R_3 R_6 R_7 \\ &+ R_2 R_4 R_6 R_8 - R_1 R_2 R_3 R_5 R_6 R_7 \\ &- R_1 R_2 R_4 R_5 R_6 R_8 - R_1 R_3 R_4 R_5 R_7 R_8 \\ &- R_2 R_3 R_4 R_6 R_7 R_8 + R_1 R_2 R_3 R_4 R_5 R_6 R_7 R_8. \end{aligned} \quad (40)$$

**5.2. Reliability importance measure.** The sensitivity of system reliability to the  $i$ -th actuator reliability (Birnbau’s measure) (Birnbau, 1969) is defined as

$$I_{B_i}(t) = \frac{\partial R_S(t)}{\partial R_i(t)} = R_S(1_i, t) - R_S(0_i, t), \quad (41)$$

where  $R_S(1_i, t)$  denotes the system reliability when the  $i$ -th actuator is perfectly reliable, and  $R_S(0_i, t)$  denotes the system reliability when the  $i$ -th actuator is faulty. This index indicates how sensitive system reliability is against changes in particular actuator reliability.

This measure will be later used as a criterion to define the  $\mathbf{W}^{\text{inv}}$  (27) of the control allocation procedure.

## 6. Health-aware control methodology

The health-aware control methodology uses the weight matrix  $\mathbf{W}^{\text{inv}}$  (27) to incorporate the health state of the actuators, represented by their reliability level and their importance for system reliability, as a criterion to redistribute the control effort in the control allocation module.

In fact, this methodology can be applied to satisfy two different objectives during the mission. On the one hand, we have an objective aimed at maximizing the system lifetime. In effect, enhancing system reliability at mission time  $t_f$  will lead to an increase in system lifetime. On the other hand, there is an objective aimed at maximizing individual rotor lifetime. In fact, incrementing rotor reliability will enhance rotor lifetime. In this case, it is assumed that predictive maintenance is performed on the system when any rotor reliability decreases below a predefined reliability threshold  $R_{\text{th}}$ .

Usually, actuators are brushless rotors, therefore maintenance could be just a visual inspection and a check of the fastening of the propellers. Therefore, different cases have been generated to maximize each of these policies as discussed below.

**Nominal case.** This case represents the behavior of the system when reliability is not taken into account and, consequently, all rotors have the same importance. Therefore, the weights (31) of the control allocation block are  $\mathbf{W}^{\text{inv}} = \mathbf{I}$ .

**Case A.** The objective is to maximize the overall system reliability at mission time ( $R_s(t_f)$ ). In this case, the weights (31) of the control allocation block are

$$\mathbf{W}^{\text{inv}} = \text{diag}(1 - \mathbf{I}_{\mathbf{B}}), \quad (42)$$

where  $\mathbf{I}_{\mathbf{B}}$  is a measure of the importance of each actuator for the system reliability (41). Therefore, it is expected that applying a smaller load on the rotors with higher  $\mathbf{I}_{\mathbf{B}}$  will improve system reliability at mission time.

**Case B.** The objective is to maximize the time at which the individual rotor reliability decreases below  $R_{\text{th}}$ .

The policy consists in reducing the load applied to the actuators that are more degraded. Thus, the weights will be given by

$$\mathbf{W}^{\text{inv}} = \text{diag}(w_i^{\text{inv}}), \quad \forall i \in [1, 8], \quad (43)$$

where  $w_i^{\text{inv}} = 60\sigma(25(R_i - 0.95)) + 5R_i$ , with  $\sigma$  being the sigmoid function. Note that the objective of (43) is to increase the difference between the weights when their values are near to 1.

As the degradation of the actuators (37) is a function of the magnitude and variations in the control input,  $u_i$  and  $\dot{u}_i$ , weighting them equally with a unique  $\mathbf{W}^{\text{inv}}$  matrix might lead to a suboptimal behavior of each criterion.

Moreover, the UAV actuators have two tasks. The first one is to generate a thrust that compensates the weight of the vehicle such that the hover position can be achieved and guaranteed. The second task is to follow the reference trajectory by generating an imbalance between the forces produced by the rotors. Taking this fact into account, the following partition of the virtual control action is proposed according to (30):

$$\mathbf{u}_v = \mathbf{u}_{v_1} + \mathbf{u}_{v_2}, \quad (44)$$

where  $\mathbf{u}_{v_1} = [T \ 0 \ 0 \ 0]^T$  is the constant load that the actuators should generate and  $\mathbf{u}_{v_2} = [0 \ \tau_x \ \tau_y \ \tau_z]^T$  is the variable load which allows modifying its orientation, and position. Taking into account the different criteria proposed, the weights of  $\mathbf{u}_{v_1}$  are defined as (42) for Case A and (43) for Case B. Stronger restrictions are imposed on  $\mathbf{u}_{v_2}$  because it is assumed that this virtual action causes a larger degradation than  $\mathbf{u}_{v_1}$ . Algorithm 1 has been applied to generate the thrust of the rotors when different weights for the virtual control inputs are involved. Then, the selected criteria for  $\mathbf{W}_i^{\text{inv}}$  are as follows.

**Case A\*.** As in Case A, the objective is to maximize system reliability. Therefore, the assignment done in (42) is used for the first part of the virtual control input ( $\mathbf{u}_{v_1}$ ), and for the second part ( $\mathbf{u}_{v_2}$ ) the assignment is

$$\mathbf{W}_2^{\text{inv}} = \text{diag}((1 - \mathbf{I}_{\mathbf{B}})^5). \quad (45)$$

**Case B\*.** As in Case B, the objective is to maximize the maintenance time  $t_m$  of the actuators. For the first part of the virtual control action ( $\mathbf{u}_{v_1}$ ), the assignment in (43) is used, and for the second part ( $\mathbf{u}_{v_2}$ ) the assignment is

$$\mathbf{W}_2^{\text{inv}} = \text{diag}(w_{2,i}^{\text{inv}}) \quad \forall i \in [1, 8], \quad (46)$$

where  $w_{2,i}^{\text{inv}} = 300\sigma(30(R_i - 0.8)) + 5R_i$ .

## 7. Simulation results

**7.1. Simulation setup.** The reliability behavior of the PPNPPNN octorotor is studied during a cornfield aerial

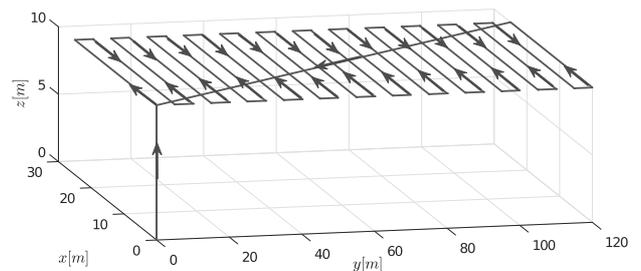


Fig. 5. UAV reference trajectory. The arrows represent the orientation of the octorotor.

supervision involving different criteria. The cornfield has an area of 3000 m<sup>2</sup> which is overflown by the aircraft at an altitude of 10 m following the grid path and returning to the starting point as presented in Fig. 5.

The octorotor is regulated by a cascade LQR control structure (detailed in Section 3) and the simulation, whose parameters are presented in Table 3, is done using the non-linear model of the PPNPPNN octorotor. To illustrate the different policies proposed, higher rotor failure rate values have been considered, so that they are scaled to the mission time. It must be highlighted that a rotor reliability threshold ( $R_{th}$ ) of 0.6 has been considered. It is assumed that the degradation of the actuator increases when its reliability ( $R_i$ ) is lower than that reliability threshold (i.e., parameter  $\beta_i$  in (37) is 10 times higher).

In initial conditions of the system, Actuator 8 starts with a lower reliability ( $R_8^0$ ). Thus, this component will define the system behavior. Note that the transitions of the control effort are generated by the trajectory reference that the UAV must follow and the performance of the controller. The chosen dynamics of the controller are very fast in order to increase the degradation produced by the fatigue of the rotors.

The objectives presented in Section 6 have been studied in both the nominal conditions of the octorotor and when a rotor fails at some point during the mission.

**7.2. Fault-free scenario.** Cases A and B are analyzed when all actuators are working properly.

**Scenario 1 (Cases A and B).** The simulation results are summarized in Table 2, where  $R_s(t_f)$  is system reliability at the end of the mission and  $t_m$  is the time when an actuator requires a maintenance (when  $R_i = R_{th}$ ). The dash symbol means that the maintenance time is higher than the end of mission time ( $t_m > t_f$ ). The reliability of each component ( $R_i$ ) is presented in Fig. 8, whereas the system reliability ( $R_s$ ) is shown in Fig. 6. The thrust generated by the actuators under the different criteria is shown in Fig. 12.

Applying the generalized inverse (27), it can be seen that system reliability is slightly better in Case A, where the sensitivity is used as weights (42), compared to the nominal case. Also, in comparison with the nominal case, in Case B the reliability of the most critical actuator (Rotor 8) is maintained 100 seconds longer; before that, a maintenance task should be performed.

**Scenario 2 (Cases A\* and B\*).** The obtained results applying the proposed approach with Algorithm 1 are presented in Table 2. With the proposed methodology (Case A\*) the reliability of the system is improved from 0.271 to 0.302 compared with that of Case A, (Fig. 6). On the other hand, the maintenance time is increased around 100 seconds (Case B\*) with respect to Case B and

Case study	$R_s(t_f)$	Maintenance time ( $t_m$ [s])								Case study	$R_s(t_f)$	Maintenance time ( $t_m$ [s])							
		$r_1$	$r_2$	$r_3$	$r_4$	$r_5$	$r_6$	$r_7$	$r_8$			$r_1$	$r_2$	$r_3$	$r_4$	$r_5$	$r_6$	$r_7$	$r_8$
Nominal	0.269	448	-	-	-	660	-	-	279	Nominal	0.069	Faulty	443	-	-	626	-	238	
A ( $W_1 = W_2$ )	0.271	424	-	-	-	599	-	268	A2 ( $W_1 = W_2$ )	0.068	Faulty	437	-	610	588	-	230		
A* ( $W_1 \neq W_2$ )	0.302	367	-	-	651	493	-	260	A*2	0.075	Faulty	434	-	550	502	-	216		
B ( $W_1 = W_2$ )	0.264	507	641	-	-	-	-	384	B ( $W_1 = W_2$ )	0.061	Faulty	418	-	-	-	-	337		
B* ( $W_1 \neq W_2$ )	0.255	539	586	-	-	-	-	475	B*	0.062	Faulty	436	668	589	661	-	676	417	

Table 2. Policy results in fault-free and faulty scenarios (rotor 1 blocked).

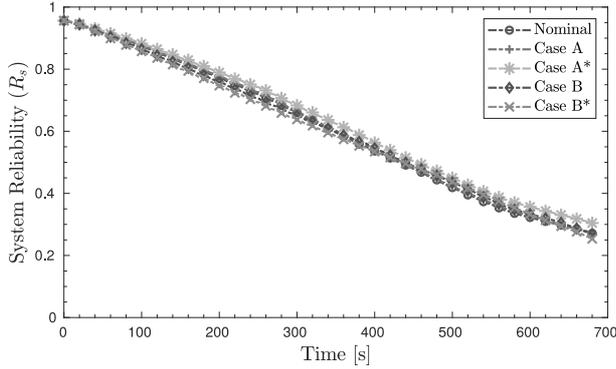


Fig. 6. System reliability (fault-free scenario).

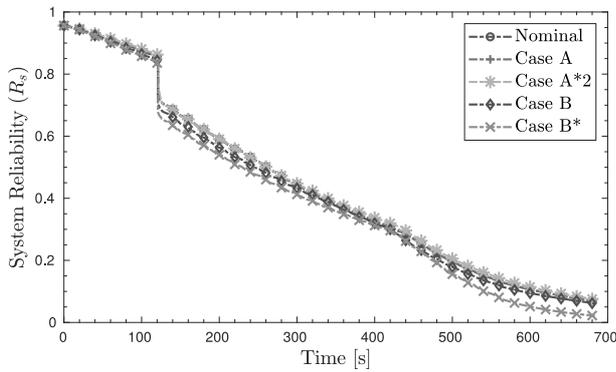


Fig. 7. System reliability (scenario for a fault in Rotor 1).

approximately 200 seconds with respect to the nominal case (Fig. 8).

**7.3. Faulty scenario.** For the faulty scenario, it has been considered that the rotor 1 becomes faulty at the time of 120 seconds. It is also assumed that a fault detection and diagnosis (FDD) module detects and estimates the fault in a deterministic time of 1 second after its occurrence. The FDD information is then introduced in the effectiveness matrix  $\Xi$  (28) in the control allocation block.

It has been observed that in Case A the octorotor becomes uncontrollable after the occurrence of a fault. This is due to the fact that reliability and controllability objectives are in conflict in the faulty scenario. Analyzing the minimal path sets of the system (38), it is possible to determine that when the rotor 1 is faulty the paths  $\zeta_1$  and  $\zeta_2$  become unavailable. Hence, the most important rotors are 2 and 6 because they belong to the active paths. This behavior poses a problem for system controllability because rotors 1 and 2 of the octorotor PPNNPPNN cannot be faulty or have restricted use ( $W_{i,i}^{inv} \rightarrow 0$ ) at the same time.

Algorithm 2 can be represented as in Fig. 10, where two switches ( $Sw_1$  and  $Sw_2$ ) are added to the diagram of Fig. 3. The switches are closed in the presence of

Table 3. Simulation parameters.

Parameter	Symbol	Value
Inner sampling time	$t_{s_i}$	0.05 [s]
Outer sampling time	$t_{s_o}$	0.25 [s]
Mission time	$t_f$	684 [s]
LQR controller		
$Q$ inner loop	$\mathbf{Q}_{in}$	diag([100.0 820.7 820.7 131.3 25.0 8.2 8.2 3.6])
$R$ inner loop	$\mathbf{R}_{in}$	$\mathbf{I}_{[4 \times 4]}$
$Q$ outer loop	$\mathbf{Q}_o$	$\mathbf{I}_{[4 \times 4]}$
$R$ outer loop	$\mathbf{R}_o$	$\mathbf{I}_{[2 \times 2]}$
Rotor parameters		
$\beta; \gamma; R_{th}$		3; 323; 0.6
Initial reliability	$R_i^0$	diag([0.83 0.95 0.89 0.97 0.94 1 1 0.72])
Baseline failure rate	$\lambda_i^0$	$1.86 \cdot 10^{-4} [s^{-1}]$
Initial conditions		
controlled outputs	$\mathbf{y}(0)$	$0_x, 0_y, 0_z, 0_\psi$
states	$\mathbf{x}(0)$	$\mathbf{0}_{[12 \times 1]}$
Feed-forward input	$\mathbf{u}_{off}$	$[mg, 0, 0, 0]^T$

a failure of those actuators belonging to their respective mesh. Note that Algorithm 2 is used for the purpose of computing the  $I_{B_i}$  measure when an actuator state DOWN. This measure allows building  $\mathbf{W}^{inv}$  in Cases A and A\*, and they have been denoted as Cases A2 and A\*2, respectively. Their component reliability will still be computed following (34). Note that the PPNNPPNN octorotor UAV becomes uncontrollable if contiguous rotors rotating in the same direction are faulty (or not used). Thus, the minimal cut sets of the structure are

$$\begin{aligned} C_1 &: \{r_1, r_2\}, & C_2 &: \{r_3, r_4\}, \\ C_3 &: \{r_5, r_6\}, & C_4 &: \{r_7, r_8\}. \end{aligned} \quad (47)$$

The minimal cut sets are composed of the rotors whose faults or limitation use cause the loss of the total controllability of the system.

Intuitively, observing Fig.1, if Rotors 1 and 2 are faulty, then there will be more rotors rotating clockwise than counterclockwise. Thus, it is impossible to find a feasible solution for the control allocation problem without losing controllability. In the presence of actuator faults, a relaxation of weight assignments should be taken. Hence, an approach has been developed to compute  $I_{B_i}$ , taking into account the critical component sets or cut sets (47), in order to achieve the objectives of Section 6 without losing controllability. This problem is handled as proposed in Algorithm 2.

Algorithm 2 aims to assign the same importance to all actuators located in the mesh which contain a faulty

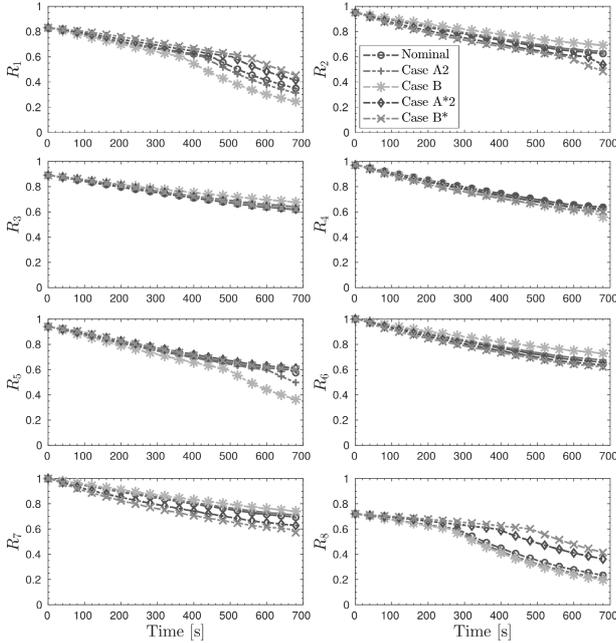


Fig. 8. Component reliability (fault-free scenario).

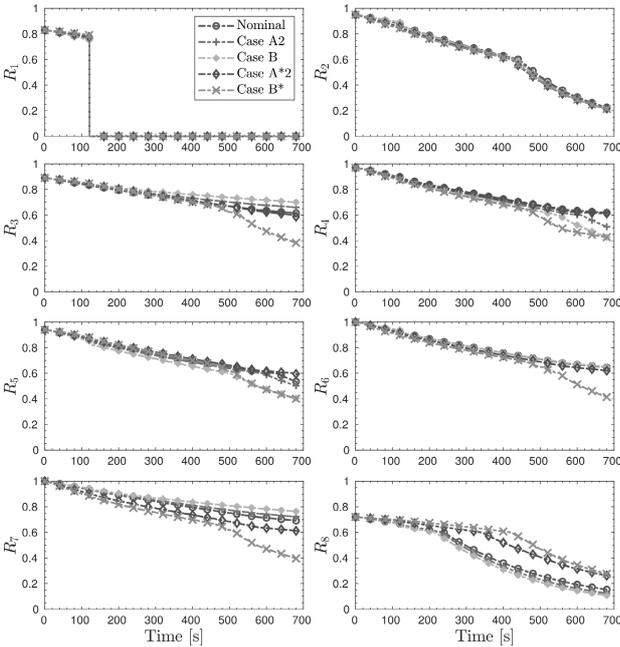


Fig. 9. Component reliability (scenario for a fault in Rotor 1).

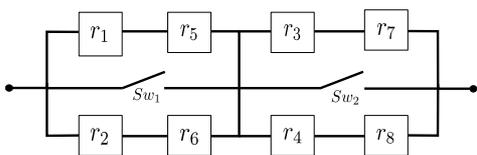


Fig. 10. Relaxed system reliability block diagram.

**Algorithm 2.**  $I_B$  soft computing.

- 1: Given Ms
- 2: **if** ( $M_{s1} = \text{DOWN}$ ) **OR** ( $M_{s2} = \text{DOWN}$ ) **OR** ( $M_{s5} = \text{DOWN}$ ) **OR** ( $M_{s6} = \text{DOWN}$ ) **then**
- 3:  $R_1 = R_2 = R_5 = R_6 = 1$
- 4: **end if**
- 5: **if** ( $M_{s3} = \text{DOWN}$ ) **OR** ( $M_{s4} = \text{DOWN}$ ) **OR** ( $M_{s7} = \text{DOWN}$ ) **OR** ( $M_{s8} = \text{DOWN}$ ) **then**
- 6:  $R_3 = R_4 = R_7 = R_8 = 1$
- 7: **end if**
- 8: Compute  $I_B$  (41)

rotor or when the reliability of the rotor is lower than a specific value ( $R_{th}$ ). The octorotor system has two meshes; one with the rotors {1, 2, 5, 6} and the other with the rotors {3, 4, 7, 8}; see Fig. 3.

$M_{si}$ , in Algorithm 2, is the state that actuators can take, i.e., DOWN when the actuator is faulty or its reliability is lower than a threshold ( $R_{th}$ ), and UP otherwise.

$$M_{s_i} = \begin{cases} \text{UP} & \text{if } R_i > R_{th} \text{ and } r_i \text{ is not faulty,} \\ \text{DOWN} & \text{otherwise.} \end{cases} \quad (48)$$

For instance, if the rotor 1 is faulty, Actuators 2, 5 and 6 have the same importance, and then switch  $Sw_1$  (Fig. 10) is closed following Algorithm 2 (Cases A2 and A\*2).

The trade-off between reliability and controllability has been studied by applying the three cases defined before. The obtained results are presented in Table 2 and Figs. 7–13. Note that, if the diagnosis block detects that an actuator is faulty, its probability of being functional is zero, and hence its reliability becomes zero.

In Case A2 the system reliability at the end of the mission is worse than in the nominal case. Nevertheless, applying the proposed approach of Case A\*2, a slight improvement is achieved (see Fig. 7). Moreover, in Case B\*, the system will need a maintenance of all rotors at around 500 seconds. The maintenance time of the most degraded rotor is improved by 180 seconds over the nominal case (Fig. 9).

Figure 11 shows the control performance when a fault in Rotor 1 occurs while tracking a section of the reference trajectory given in Fig. 5. Note that the controller is able to stabilize the system as soon as the fault is detected.

The reliability improvement is achieved by setting the control action according to the chosen criteria for both scenarios: the fault-free case (Fig. 12) and the one where Rotor 1 is faulty (Fig. 13).

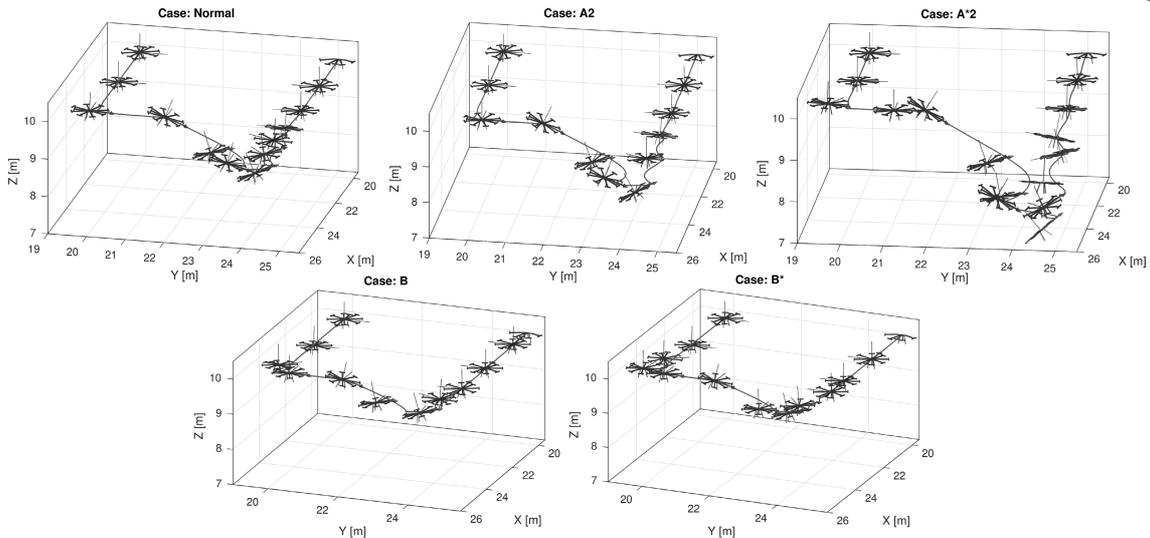


Fig. 11. Octorotor recovery behavior when Actuator 1 fails.

## 8. Conclusions

In this work, the behavior of the PPNNPPNN octorotor when its reliability is taken into account in the control loop of the system has been studied and presented. Several policies have been proposed to distribute the control effort under a control allocation scheme.

As a key contribution, in this work, the system reliability block diagram has been obtained from the system functioning analysis showing that at least certain 4-rotor configurations are required.

On the one hand, it has been considered that the component degradation depends on the load applied and its derivative. As a consequence, virtual control actions do not have the same importance. Therefore, an algorithm has been developed to solve the control allocation problem by taking into account the effect of each virtual control action on the degradation of the component. Results demonstrate that the algorithm maximizes the objective of the studied scenarios.

On the other hand, it has been observed that the controllability and reliability objectives are in conflict when a total fault is introduced in the system. Therefore, it is necessary to redefine system reliability sensitivity using Algorithm 2 to prevent an uncontrollable situation during the mission of the UAV.

It must be highlighted that the proposed methodology to solve this conflict between controllability and reliability is applicable in the case of multirotors. Simulation results demonstrate the validity of this approach in terms of reliability and controllability of the octorotor through an effective health management of the UAV actuator effort.

Further theoretical developments should be made in order to determine the efficiency of this method in other application domains. Moreover, research could address

the integration of system reliability information on other FTC strategies such as active or passive approaches.

Concerning system reliability computation, a more detailed study on the link between system reliability and controllability should be pursued.

## Acknowledgment

This work has been partially supported by the Spanish State Research Agency (AEI) and the European Regional Development Fund (ERFD) through the project SCAV (ref. MINECO DPI2017-88403-R), by AGAUR through the contract FI-AGAUR (ref. 2017FI-B00285) and by the DGR of Generalitat de Catalunya (SAC group ref. 2017/SGR/482).

## References

- Abdolkhosseini, M., Zhang, Y. and Rabbath, C. (2013). An efficient model predictive control scheme for an unmanned quadrotor helicopter, *Journal of Intelligent & Robotic Systems* **70**(1–4): 27–38.
- Adir, V. and Stoica, A. (2012). Integral LQR control of a star-shaped octorotor, *INCAS BULLETIN* **4**(2): 3–18.
- Alwi, H. and Edwards, C. (2006). Sliding mode FTC with on-line control allocation, *Proceedings of the 45th IEEE Conference on Decision and Control, San Diego, CA, USA*, pp. 5579–5584.
- Birbaum, Z. (1969). On the importance of different components in a multicomponent system, in P. Krishnaiah (Ed.), *Multivariate Analysis*, Vol. II, Academic Press, New York, NY, pp. 581–592.
- Blakelock, J. (1991). *Automatic Control of Aircraft and Missiles*, John Wiley & Sons, New York, NY.

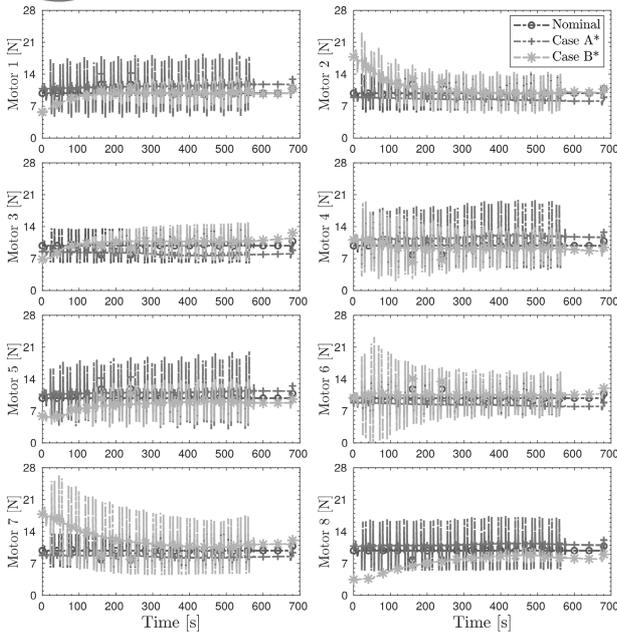


Fig. 12. Thrust generated by the actuators under different criteria (fault-free scenario).

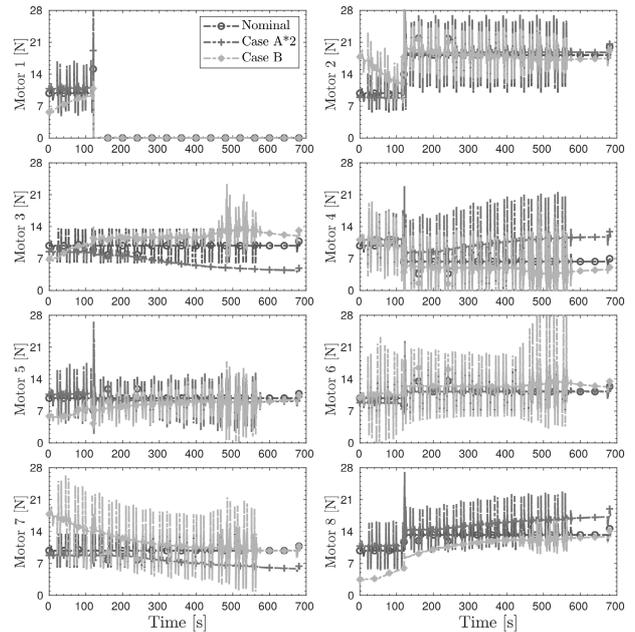


Fig. 13. Thrust generated by the actuators under different policies (scenario for a fault in Rotor 1).

- Bodson, M. (2002). Evaluation of optimization methods for control allocation, *Journal of Guidance, Control, and Dynamics* **25**(4): 703–711.
- Bordington, K. and Durham, W. (1995). Closed-form solutions to constrained control allocation problem, *Journal of Guidance, Control, and Dynamics* **18**(5): 1000–1007.
- Cen, Z., Noura, H. and Younes, Y.A. (2015). Systematic fault tolerant control based on adaptive Thau observer estimation for quadrotor UAVs, *International Journal of Applied Mathematics and Computer Science* **25**(1): 159–174, DOI: 10.1515/amcs-2015-0012.
- Cox, D.R. (1972). Regression models and life-tables, *Journal of the Royal Statistical Society B (Methodological)* **34**(2): 187–220.
- Durham, W.C. (1993). Constrained control allocation, *Journal of Guidance, Control, and Dynamics* **16**(4): 717–725.
- Freddi, A., Lanzon, A. and Longhi, S. (2011). A feedback linearization approach to fault tolerance in quadrotor vehicles, *IFAC Proceedings Volumes* **44**(1): 5413–5418.
- Gertsbakh, I.B. (2001). *Reliability Theory: With Applications to Preventive Maintenance*, 2nd Edn, Springer, New York, NY.
- Johansen, T. and Fossen, T. (2013). Control allocation—a survey, *Automatica* **49**(5): 1087–1103.
- Khelassi, A., Theilliol, D., Weber, P. and Ponsart, J. (2011). Fault-tolerant control design with respect to actuator health degradation: An LMI approach, *Proceedings of the IEEE International Conference on Control Applications (CCA), Denver, CO, USA*, pp. 983–988.
- Liu, C., Chen, W.-H. and Andrews, J. (2012). Tracking control of small-scale helicopters using explicit nonlinear MPC

- augmented with disturbance observers, *Control Engineering Practice* **20**(3): 258–268.
- Mahony, R., Kumar, V. and Corke, P. (2012). Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor, *IEEE Robotics Automation Magazine* **19**(3): 20–32.
- Marks, A., Whidborne, J. and Yamamoto, I. (2012). Control allocation for fault tolerant control of a VTOL octorotor, *2012 UKACC International Conference on Control (CONTROL), Cardiff, UK*, pp. 357–362.
- Merheb, A.-R., Noura, H. and Bateman, F. (2015). Design of passive fault-tolerant controllers of a quadrotor based on sliding mode theory, *International Journal of Applied Mathematics and Computer Science* **25**(3): 561–576, DOI: 10.1515/amcs-2015-0042.
- Milhim, A., Zhang, Y. and Rabbath, C.-A. (2010). Gain scheduling based PID controller for fault tolerant control of quad-rotor UAV, *Proceedings of AIAA Infotech@Aerospace 2010, Atlanta, GA, USA*, pp. 1–13.
- Ogata, K. (1995). *Discrete-time Control Systems*, 2nd Edn, Prentice-Hall, Upper Saddle River, NJ.
- Raffo, G., Ortega, M. and Rubio, F. (2010). An integral predictive/nonlinear control structure for a quadrotor helicopter, *Automatica* **46**(1): 29–39.
- Rinaldi, F., Gargioli, A. and Quagliotti, F. (2014). PID and LQ regulation of a multirotor attitude: Mathematical modelling, simulations and experimental results, *Journal of Intelligent & Robotic Systems* **73**(1–4): 33–50.
- Rotondo, D., Nejjari, F. and Puig, V. (2015). Robust quasi-LPV model reference FTC of a quadrotor UAV subject to actuator faults, *International Journal of Applied Mathematics and Computer Science* **25**(1): 7–22, DOI: 10.1515/amcs-2015-0001.

- Salazar, J., Nejjari, F., Sarrate, R., Weber, P. and Theilliol, D. (2016). Reliability importance measures for a health-aware control of drinking water networks, *Proceedings of the 3rd Conference on Control and Fault-Tolerant Systems (SysTol)*, Barcelona, Spain, pp. 572–578.
- Salazar, J., Sanjuan, A., Nejjari, F. and Sarrate, R. (2017). Health-Aware control of an octorotor UAV system based on actuator reliability, *Proceedings of the 4th International Conference on Control, Decision and Information Technologies (CoDIT)*, Barcelona, Spain, pp. 815–820.
- Salazar, J., Weber, P., Nejjari, F., Theilliol, D. and Sarrate, R. (2015). MPC framework for system reliability optimization, in Z. Kowalczyk (Ed.), *Advanced and Intelligent Computations in Diagnosis and Control*, Springer International Publishing, Cham, pp. 161–177.
- Sanjuan, A., Nejjari, F. and Sarrate, R. (2019). Reconfigurability analysis of multirotor UAVs under actuator faults, *Proceedings of the 4th Conference on Control and Fault-Tolerant Systems (SysTol)*, Casablanca, Morocco, pp. 26–31.
- Schneider, T., Ducard, G., Konrad, R. and Pascal, S. (2012). Fault-tolerant control allocation for multirotor helicopters using parametric programming, *International Micro Air Vehicle Conference and Flight Competition (IMAV)*, Braunschweig, Germany, pp. 1–8.
- Zhang, Y., Chamseddine, A., Rabbath, C., Gordon, B., 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.
- Jean C. Salazar** received his PhD degree in automation, robotics, and vision from Universitat Politècnica de Catalunya (UPC), Terrassa, Spain, in 2018. He has visited the Research Center for Automatic Control (CRAN) at the University of Lorraine in Nancy, France. His research fields are model-based diagnosis, prognosis, and the integration of control with prognosis and health management strategies.
- Adrián Sanjuan** received his BS and MS degrees from the Universitat Politècnica de Catalunya (UPC), Spain. Since 2017, he has been a PhD student at the Research Center for Supervision, Safety and Automatic Control (CS2AC) at UPC. His main research interests include artificial intelligence, identification, fault detection and isolation (FDI) and fault tolerant control (FTC) of non-linear dynamic systems.
- Fatiha Nejjari** is an associate professor with the Department of Automatic Control, Universitat Politècnica de Catalunya (UPC). She is also a member of the Advanced Control Systems (SAC) research group of the Research Center for Supervision, Safety and Automatic Control (CS2AC) at UPC. Her main research areas include LPV control, fault detection and isolation, and fault tolerant control of dynamic systems. She has published various papers in journals and international conferences, and has participated in several European projects and networks related to these topics.
- Ramon Sarrate** received his MS and PhD degrees in industrial engineering from Universitat Politècnica de Catalunya (UPC), Terrassa, Spain, in 1994 and 2003, respectively. He is currently an assistant professor with the Department of Automatic Control at UPC. His present research interests include model-based fault diagnosis and hybrid systems. He has been involved in several national and European research projects, and has published several papers in scientific journals and international conference proceedings.

Received: 30 December 2018

Revised: 21 October 2019

Accepted: 17 November 2019