Risk-sensitive motion planning for MAVs based on mission-related fault-tolerant analysis

Nedim Osmic, Adnan Tahirovic & Ivan Petrovic

Published online: 13 Mar 2020.

Submit your article to this journal

Article views: 332

View related articles

View Crossmark data

Taylor & Francis
Taylor & Francis Group

Risk-sensitive motion planning for MAVs based on mission-related fault-tolerant analysis

Nedim Osmic [a], Adnan Tahirovic [a] and Ivan Petrovic[b]

[a]Department of Automatic Control and Electronics, Faculty of Electrical Engineering, University of Sarajevo, Sarajevo, Bosnia and Herzegovina; [b]Department of Control and Computer Engineering, Faculty of Electrical Engineering and Computing, University of Zagreb, Zagreb, Croatia

ABSTRACT

Multirotor Aerial Vehicles may be fault-tolerant by design when rotor-failure is possible to measure or identify, especially when a large number of rotors are used. For instance, an octocopter can be capable to complete some missions even when a double-rotor fault occurs during the execution. In this paper, we study how a rotor-failure reduces the vehicle control admissible set and its importance with respect to the selected mission, i.e. we perform mission-related fault-tolerant analysis. Furthermore, we propose a risk-sensitive motion-planning algorithm capable to take into account the risks during the planning stage by means of mission-related fault-tolerant analysis. We show that the proposed approach is much less conservative in terms of selected performance measures than a conservative risk planner that assumes that the considered fault will certainly occur during the mission execution. As expected, the proposed risk-sensitive motion planner is also readier for accepting failures during the mission execution than the risk-insensitive approach that assumes no failure will occur.

1. Introduction

Nowadays, there exist different design solutions for Multirotor Aerial Vehicle (MAV) from micro and mini MAVs to heavy MAVs with large endurance [1]. Due to characteristics such as small geometries, vertical takeoff and landing, low cost, simple construction, degrees of freedom, their inherent manoeuvrability, ability to perform tasks that are difficult for humans where the risks of injury are high, MAVs have become the most popular type of unmanned aerial vehicles.

MAVs perform various types of tasks including search and rescue missions in indoor and outdoor environments [2,3], aerial construction [4], precision agriculture [5], disaster management [6], power line and structural inspection [7–9], exploration and mapping of unknown environments [10–13], remote sensing [14], aerial transportation [15], monitoring and analysis of traffic [16], surveillance [17,18], swarming [19] and use as educational platforms [20]. The requirement of each of these applications is that the MAV correctly and reliably performs the task for which it is designed.

Regardless of the structural design type, different faults on a MAV may occur. The fault can affect actuators, sensors, controller or can be of a structural nature. If a failure occurs, the mission execution may be stopped. To increase the likelihood of mission accomplishment, different redundancy can be implemented including redundancy in propulsion system as well. The

paper [21] shows that it is possible to control all degrees of freedom of the octocopter except the yaw angle for any potential double-rotor-fault scenario (the yaw controllability is preserved even in 89% of those scenarios). In [22–26], the authors addressed possibility of preserving the controllability of a system for different rotor faults by increasing the number of rotors, or using a rotor with a possibility to tilt motors [27,28]. In [29,30], the authors have investigated a control strategy for a quadcopter in the case of losing a single, two opposing, or three propellers.

On the other hand, regardless of whether the configuration of a MAV is redundant, the control algorithm has a significant role in improving fault-tolerance of the MAV system. If a control algorithm is fault-ignorant, having redundant components does not necessary increase reliability of the MAV system or probability of completing the mission. The control algorithms that inherently posses a certain level of robustness with respect to possible failures increase reliability of the system. There is a large number of methods developed within the framework of fault-tolerant control including sliding mode control [31], adaptive fault-tolerant control [32,33], control allocation method for MAVs [34,35], reconfigurable control [36], backstepping method [37], model predictive control [38], control based on linear quadratic regulator [39], fuzzy predictive control [40] and many others.

In this paper, we propose a novel motion planning algorithm which takes into account potential rotor-failures of the MAV during the planning stage, named here as risk-sensitive planner (RSP). The RSP planner is much more prepared for rotor-faults during the mission execution than the planner ignorant to those potential faults, named here as risk-insensitive planner (RIP). Additionally, the proposed planner is much less conservative comparing to the approach which plans the mission assuming the faults will certainly occur during the execution, named as risk-conservative planner (RCP). To do so, we propose a procedure for (i) finding a reduced fault-dependent control admissible region, (ii) replacing that region with a set of inequality constraints, (iii) carefully selecting some of the inequality constraints based on fault-tolerant analysis of the given mission, and (iv) forming the final optimization framework which includes the selected constraints.

The paper is organized as follows. The octocopter model is presented in Section 2 as an illustrative example of MAV systems. Section 3 includes a controllability test for extracting the fault-dependent control admissible set for any fault considered. An RLS-based fault-tolerant tracking control is included in Section 4 to use a unique control framework for different planners and faults during the validation stage. Section 5 introduces the proposed risk-sensitive motion planner and provides a validation of the approach for a given mission. Section 6 concludes the paper.

2. MAV model

In order to achieve satisfactory control of a MAV and to eventually improve reliability of the mission execution during the planning stage as well, it is necessary to understand all stability-influential elements of its design. To that purpose, the octocopter model derived and analysed in [41] and [42] will be presented as a representative MAV example in order to easily deduce a generalized UAV model in the planar plane with a symmetrical configuration design consisting of $2n$ DC motors, where for $n = \{2, 3, 4\}$ we have well-known MAV designs including quadcopter, hexacopter and octocopter, respectively.

2.1. Frames of reference and control inputs

The octocopter system with a PNPNPNPN configuration design, where P and N indicate clockwise (CW) and counter-clockwise (CCW) directions of rotation of a related DC motor, and its body and ground fixed frames are shown in Figure 1. Octocopter is constructed with eight DC motors accompanied by the propellers. Each motor with propeller is mounted on arm with length l. The adjacent arms are equally distant from each other by $45°$ angles ($360°/n$).

Two reference frames are used to derive the model, one for a local coordinate system $\{o\}$ attached to the UAV and one representing a global coordinate system $\{g\}$ fixed to the ground. For $\{g\}$, the ENU convention is used to represent the axes, meaning that the axes X_B, Y_B and Z_B are pointing to the north, east and up, where $\boldsymbol{x} = [x\ y\ z]^T$ and $\boldsymbol{\Psi} = [\phi\ \theta\ \psi]^T$ indicate the position and the Euler-based orientation. The linear velocity $\boldsymbol{v} = [u\ v\ w]^T$ and the angular velocity $\boldsymbol{P} = [P\ Q\ R]^T$ are represented in $\{o\}$. The positive directions of ϕ, θ and ψ are chosen to coincide with the positive directions of P, Q and R, respectively.

The MAV rotors together generate the total thrust force T and torques $\boldsymbol{\tau} = [\tau_x\ \tau_y\ \tau_z]$ acting around the axes of the local coordinate system. One can approximately calculate generated forces F_i by using the expression $F_i = b\Omega_i^2\ (i = \overline{1, \ldots, 8})$ [43], where $b\ [\text{Ns}^2/\text{rad}^2]$ is the rotor thrust constant and $\Omega_i\ [\text{rad/s}]$ the angular velocity of the ith rotor. From the Newton's third law one can obtain all generated counter torques, $M_i\ (i = \overline{1, \ldots, 8})$, in the form $M_i = d\Omega_i^2$, where $d\ [\text{Nms}^2/\text{rad}^2]$ represents the rotor drag coefficient.

The system control input can be expressed as:

$$\boldsymbol{u} = \boldsymbol{A}\boldsymbol{\Omega}_s = \begin{bmatrix} T & \boldsymbol{\tau} \end{bmatrix}^T = \begin{bmatrix} T & \tau_x & \tau_y & \tau_z \end{bmatrix}^T, \quad (1)$$

where $\boldsymbol{\Omega}_s$ represents the squared rotor velocity vector given as $\boldsymbol{\Omega}_s = [\ \Omega_1^2\ \Omega_2^2\ \Omega_3^2\ \Omega_4^2\ \Omega_5^2\ \Omega_6^2\ \Omega_7^2\ \Omega_8^2\ ]^T$ and $\boldsymbol{A}$ being the system actuation matrix defined as:

$$\boldsymbol{A} = \begin{bmatrix} b & b & b & b \\ bl & \frac{\sqrt{2}}{2}bl & 0 & -\frac{\sqrt{2}}{2}bl \\ 0 & -\frac{\sqrt{2}}{2}bl & -bl & -\frac{\sqrt{2}}{2}bl \\ -d & d & -d & d \\ b & b & b & b \\ -bl & -\frac{\sqrt{2}}{2}bl & 0 & \frac{\sqrt{2}}{2}bl \\ 0 & \frac{\sqrt{2}}{2}bl & bl & \frac{\sqrt{2}}{2}bl \\ -d & d & -d & d \end{bmatrix}. \quad (2)$$

Changing the rotor velocity of the motors in range $0 \leq \Omega_i \leq \Omega_{\max}$, $i = \overline{1, \ldots, 8}$, the different thrust force (T) and the torque (τ_x, τ_y and τ_z) about the x, y and z axes can be achieved.

2.2. Octopcopter kinematics and dynamics

The octocopter kinematic model of the linear motion along the X_B, Y_B and Z_B axes of the global coordinate system as $\dot{x}$, $\dot{y}$ and $\dot{z}$, respectively, can be represented as

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^T = \boldsymbol{R}(\phi, \theta, \psi)\,\boldsymbol{v}, \quad (3)$$
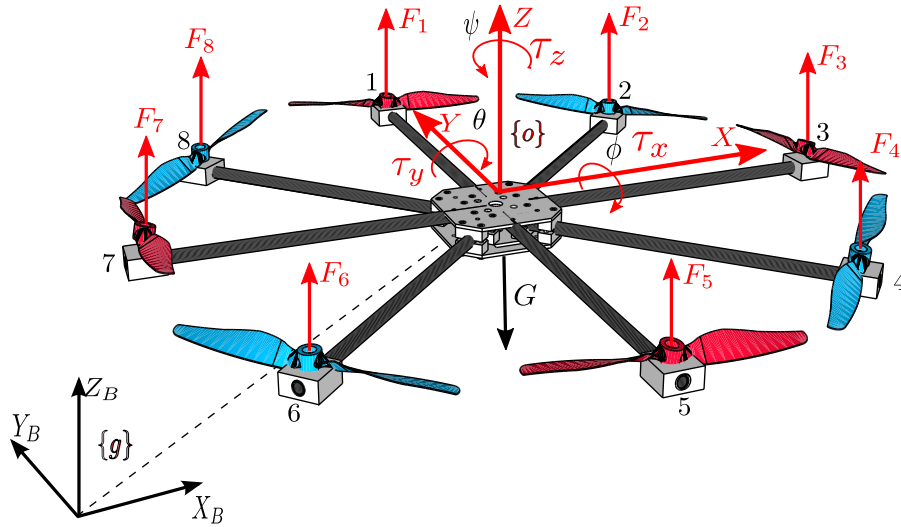
Figure 1. Body and ground fixed frames for a PNPNPNPN octocopter system, where P and N indicate counter (motors M1, M3, M5 and M7) and clockwise (motors M2, M4, M6 and M8) directions [41].

where for the ZYX convention (which is used in this paper), the rotation matrix $\boldsymbol{R}(\phi, \theta, \psi)$ is defined as

$$\boldsymbol{R}(\phi, \theta, \psi)$$
$$= \begin{bmatrix} c_\psi\, c_\theta & c_\psi\, s_\theta\, s_\phi - s_\psi\, c_\phi & c_\psi\, s_\theta\, c_\phi + s_\psi\, s_\phi \\ s_\psi\, c_\theta & s_\psi\, s_\theta\, s_\phi + c_\psi\, c_\phi & s_\psi\, s_\theta\, c_\phi - c_\psi\, s_\phi \\ -s_\theta & c_\theta\, s_\phi & c_\theta\, c_\phi \end{bmatrix}, \tag{4}$$

where c and s indicate *cosine* and *sine* functions.

The octocopter kinematic model of the angular motion can be represented as

$$\dot{\boldsymbol{\Psi}} = \boldsymbol{R}_A^{-1}(\phi, \theta, \psi)\, \boldsymbol{P}, \tag{5}$$

with the angular rotation matrix $\boldsymbol{R}_A^{-1}$ being

$$\boldsymbol{R}_A^{-1}(\phi, \theta, \psi) = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & \dfrac{s_\phi}{c_\theta} & \dfrac{c_\phi}{c_\theta} \end{bmatrix}, \tag{6}$$

where t indicates *tangens* function.

The dynamics of the octocopter linear motion can be described by

$$\dot{\boldsymbol{v}} = \begin{bmatrix} 0 \\ 0 \\ \dfrac{T}{m_o} \end{bmatrix} + g \begin{bmatrix} s_\theta \\ -s_\phi\, c_\theta \\ -c_\phi\, c_\theta \end{bmatrix} - \boldsymbol{S}\boldsymbol{v}, \tag{7}$$

where m_o represents the mass of the system, g is the gravitational acceleration and $\boldsymbol{S}$ is the skew-symmetric matrix

$$\boldsymbol{S} = \begin{bmatrix} 0 & -R & Q \\ R & 0 & -P \\ -Q & P & 0 \end{bmatrix}. \tag{8}$$

The dynamics of the angular motion of the octocopter can be described by

$$\dot{\boldsymbol{P}} = \boldsymbol{J}^{-1}\left(\boldsymbol{\tau} - \boldsymbol{SJP}\right), \tag{9}$$

where $\boldsymbol{J} = \mathrm{diag}\left(\begin{bmatrix} I_{xx} & I_{yy} & I_{zz} \end{bmatrix}\right)$ represents the inertia tensor with components I_{xx}, I_{yy} and I_{zz} determined by Huygens-Steiner theorem [44].

Equations (3)–(9) represent the kinematic and dynamic motion of any MAV with $2n$ DC motors mounted in a planar plane providing a parametrized model dependent on m_o and $\boldsymbol{J}$.

2.3. Model generalization

Equations (3)–(9) that describe the kinematic and dynamic motion of the octocopter can be used for any MAV with $2n$ ($n \geq 2$) rotors in the planar plane with a symmetrical configuration, where the dimension of the activation matrix $\boldsymbol{A}$ is $4 \times 2n$ and b, l and d depend on the selected MAV design parameters. For instance, in case of a quadcopter with PNPN configuration the actuation matrix is

$$\boldsymbol{A} = \begin{bmatrix} b & b & b & b \\ bl & 0 & -bl & 0 \\ 0 & -bl & 0 & bl \\ -d & d & -d & d \end{bmatrix}, \tag{10}$$

while for a hexacopter with PNPNPN and PPNNPN configurations, the related actuation matrices become

$$\boldsymbol{A} = \begin{bmatrix} b & b & b & b & b & b \\ bl & \dfrac{bl}{2} & -\dfrac{bl}{2} & -bl & -\dfrac{bl}{2} & \dfrac{bl}{2} \\ 0 & -\dfrac{\sqrt{3}}{2}bl & -\dfrac{\sqrt{3}}{2}bl & 0 & \dfrac{\sqrt{3}}{2}bl & \dfrac{\sqrt{3}}{2}bl \\ -d & d & -d & d & -d & d \end{bmatrix} \tag{11}$$

$$A = \begin{bmatrix} b & b & b & b & b & b \\ bl & \dfrac{bl}{2} & -\dfrac{bl}{2} & -bl & -\dfrac{bl}{2} & \dfrac{bl}{2} \\ 0 & -\dfrac{\sqrt{3}}{2}bl & -\dfrac{\sqrt{3}}{2}bl & 0 & \dfrac{\sqrt{3}}{2}bl & \dfrac{\sqrt{3}}{2}bl \\ -d & -d & d & d & -d & d \end{bmatrix}. \tag{12}$$

By observing these three examples, one can determine the actuation matrix of any MAV in the planar plane with a symmetrical configuration consisting of $2n$ DC motors, where the first motor coincides with the positive direction of y axis and all subsequent DC motors are equally distant by $\frac{360^\circ}{2n}$. Namely, the first row consists the parameter b only. The elements of the second and the third row affect the rotation about the x and y axes, respectively. They are formed as *cosine* and *sine* values of the angles by which the related DC motor is displaced with respect to the origin of the x and y axes, respectively. The elements of the fourth row depend on the drag coefficient d and the direction of DC motor rotation, where the positive sign reflects clockwise direction. For instance, if the successive DC motors have different directions the related actuation matrix can be represented in a generalized form by

$$A = \begin{pmatrix} b & \cdots & b \\ bl \cdot \cos\left(\dfrac{360 \cdot k_1}{2n}\right) & \cdots & bl \cdot \cos\left(\dfrac{360 \cdot k_i}{2n}\right) \\ -bl \cdot \sin\left(\dfrac{360 \cdot k_1}{2n}\right) & \cdots & \sin\left(\dfrac{360 \cdot k_i}{2n}\right) \\ -d & \cdots & d \end{pmatrix}$$

$$\begin{pmatrix} \cdots & b \\ \cdots & bl \cdot \cos\left(\dfrac{360 \cdot k_{2n}}{2n}\right) \\ \cdots & -bl \cdot \sin\left(\dfrac{360 \cdot k_{2n}}{2n}\right) \\ \cdots & d \end{pmatrix}, \tag{13}$$

where $k_i = i - 1$, $i = 1, 2, \ldots, 2n$. This type of model is used in paper [41,42,45,46].

2.4. Simple PD tracking controller with control allocation

The task of the tracking controller is to ensure that a MAV follows the mission-related trajectory. In this section we briefly present the proportional-differential (PD) control scheme depicted in Figure 2, which is a classical control architecture [43,47] presented in more details in [45].

The controlled variables are the vehicle position x, y, z and its attitude $\mathbf{\Psi}$, while the overall architecture includes xy, altitude, attitude and motor controllers as well as the control allocation and system dynamics consisted of the motor, actuation and MAV (e.g. octocopter) dynamics (see Figure 2).

The desired linear motion reference values along mission trajectory are applied to the xy controller, while the corresponding desired attitude and altitude reference values are handled by the related controllers. The outputs of these controllers form the desired values of the total force and torques to the control allocation algorithm to deal with the over-actuated system. This algorithm then distributes these desired values onto the desired velocity vector $\mathbf{\Omega}_{ref}$ to provide speeds for each motor. For the purpose of this work, we use a pseudo-inverse control allocation [21,48]. The motor controller is used as a low-level controller to force the motor velocity vector $\mathbf{\Omega}$ to follow the reference values from $\mathbf{\Omega}_{ref}$.

To design a PD tracking controller, it is common practice to linearize the octocopter dynamics around the hover configuration

$$\begin{aligned} (x_e, y_e, z_e) &= (x, y, z), \\ (u_e, v_e, w_e) &= (0, 0, 0), \\ (P_e, Q_e, R_e) &= (0, 0, 0), \\ (\phi_e, \theta_e, \psi_e) &= (0, 0, \psi). \end{aligned} \tag{14}$$

By the linearization of the nonlinear system model presented in Section 2, one can obtain

$$\dot{\mathbf{x}} = \mathbf{R}(Z, \psi_e)\,\mathbf{v} \tag{15}$$

$$\dot{\mathbf{\Psi}} = \mathbf{P} \tag{16}$$

$$\dot{\mathbf{v}} = \begin{bmatrix} 0 \\ 0 \\ \dfrac{T}{m_o} \end{bmatrix} + g \begin{bmatrix} \theta \\ -\phi \\ -1 \end{bmatrix} \tag{17}$$

$$\dot{\mathbf{P}} = \mathbf{J}^{-1}\boldsymbol{\tau}, \tag{18}$$

where $\mathbf{R}(Z, \psi_e)$ represents the rotation matrix around the z-axis. Substitution of (17) in (15) yields

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = g \begin{bmatrix} c_{\psi_e} & -s_{\psi_e} \\ s_{\psi_e} & c_{\psi_e} \end{bmatrix} \begin{bmatrix} \theta \\ -\phi \end{bmatrix} \tag{19}$$

and

$$\ddot{z} = \frac{T}{m} - g. \tag{20}$$

As it can be noticed from (20), altitude can be directly controlled with thrust T. If $e_z = z_{ref} - z$ is now the altitude tracking error, the control law can be chosen as

$$T^{des} = m_o\left(g + \ddot{z}_{ref} + K_{dz}\dot{e}_z + K_{pz}e_z\right), \tag{21}$$

where z_{ref} is the reference altitude, K_{pz} is the proportional gain, K_{dz} the derivative gain. In a similar manner, substitution of (18) in (16) yields

$$\ddot{\mathbf{\Psi}} = \mathbf{J}^{-1}\boldsymbol{\tau}. \tag{22}$$

It can be observed from (22) that the attitude can be directly controlled with torque vector $\boldsymbol{\tau}$. If now $e_\Psi =$
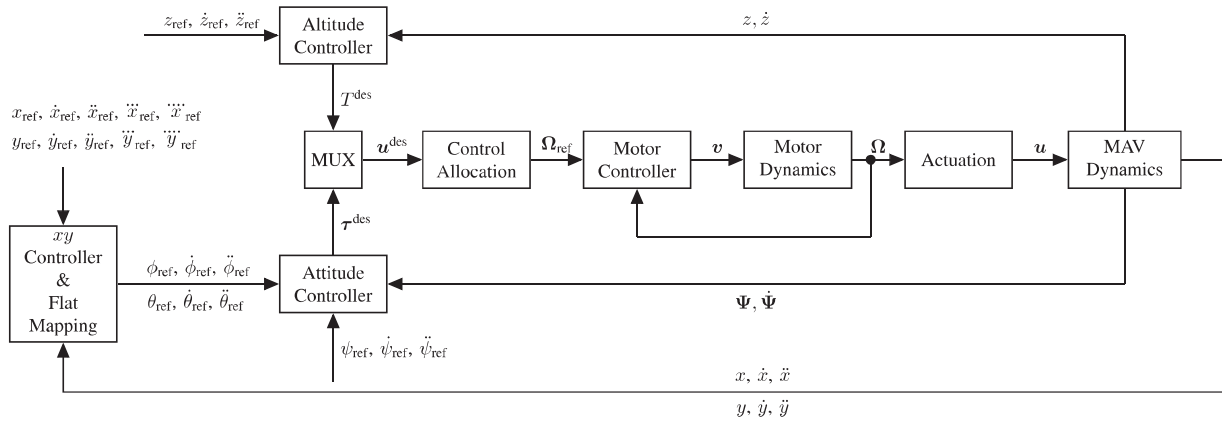
Figure 2. Control architecture.

$\Psi_{\text{ref}} - \Psi$ is the attitude tracking error, the control law can be chosen as

$$\boldsymbol{\tau}^{\text{des}} = \boldsymbol{J}\left(\ddot{\Psi}_{\text{ref}} + K_d \dot{\boldsymbol{e}}_\Psi + K_p \boldsymbol{e}_\Psi\right), \qquad (23)$$

where Ψ_{ref} is the reference attitude, K_p is the proportional gain, K_d the derivative gain.

Clearly, all generalized forces are stand for attitude and altitude control. Therefore, in order to achieve tracking of the x and y position coordinates, we need a flat mapping between those positions and the altitude and attitude coordinates. For that reason, let the tracking errors $e_x = x_{\text{ref}} - x$ and $e_x = y_{\text{ref}} - y$ be introduced, where x_{ref} and y_{ref} are the reference values. If one wants the tracking errors to decay exponentially, it is sufficient that the following holds

$$\begin{bmatrix} \ddot{e}_x \\ \ddot{e}_y \end{bmatrix} + K_d \begin{bmatrix} \dot{e}_x \\ \dot{e}_y \end{bmatrix} + K_p \begin{bmatrix} e_x \\ e_y \end{bmatrix} = \mathbf{0}_{2\times 1}, \qquad (24)$$

which can be rewritten as (25)

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} \ddot{x}_{\text{ref}} \\ \ddot{y}_{\text{ref}} \end{bmatrix} + K_d \begin{bmatrix} \dot{e}_x \\ \dot{e}_y \end{bmatrix} + K_p \begin{bmatrix} e_x \\ e_y \end{bmatrix}. \qquad (25)$$

From (19) and assuming that $\Psi_e = \Psi_{\text{ref}} = const$, the reference values of the roll ϕ_{ref} and pitch θ_{ref} become

$$\begin{bmatrix} \phi_{\text{ref}} \\ \theta_{\text{ref}} \end{bmatrix} = \frac{1}{g} \begin{bmatrix} s\psi_{\text{ref}} & -c\psi_{\text{ref}} \\ c\psi_{\text{ref}} & s\psi_{\text{ref}} \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix}. \qquad (26)$$

The attitude controller (23) needs the first and the second derivation of the roll and pitch reference values. By differentiating (26), we get

$$\begin{bmatrix} \dot{\phi}_{\text{ref}} \\ \dot{\theta}_{\text{ref}} \end{bmatrix} = \frac{1}{g} \begin{bmatrix} s\psi_{\text{ref}} & -c\psi_{\text{ref}} \\ c\psi_{\text{ref}} & s\psi_{\text{ref}} \end{bmatrix} \begin{bmatrix} \dddot{x} \\ \dddot{y} \end{bmatrix},$$

$$\begin{bmatrix} \ddot{\phi}_{\text{ref}} \\ \ddot{\theta}_{\text{ref}} \end{bmatrix} = \frac{1}{g} \begin{bmatrix} s\psi_{\text{ref}} & -c\psi_{\text{ref}} \\ c\psi_{\text{ref}} & s\psi_{\text{ref}} \end{bmatrix} \begin{bmatrix} \ddddot{x} \\ \ddddot{y} \end{bmatrix}. \qquad (27)$$

If $\dddot{x}, \dddot{y}, \ddddot{x}, \ddddot{y}$ are known, then one can obtain $\dot{\phi}_{\text{ref}}, \dot{\theta}_{\text{ref}}, \ddot{\phi}_{\text{ref}}$ and $\ddot{\theta}_{\text{ref}}$ from (27). By additional differentiation

of (25), we get $(\dddot{x}, \dddot{y})$ and $(\ddddot{x}, \ddddot{y})$ as follows

$$\begin{bmatrix} \dddot{x} \\ \dddot{y} \end{bmatrix} = \begin{bmatrix} \dddot{x}_{\text{ref}} \\ \dddot{y}_{\text{ref}} \end{bmatrix} + K_d \begin{bmatrix} \ddot{e}_x \\ \ddot{e}_y \end{bmatrix} + K_p \begin{bmatrix} \dot{e}_x \\ \dot{e}_y \end{bmatrix},$$

$$\begin{bmatrix} \ddddot{x} \\ \ddddot{y} \end{bmatrix} = \begin{bmatrix} \ddddot{x}_{\text{ref}} \\ \ddddot{y}_{\text{ref}} \end{bmatrix} + K_d \begin{bmatrix} \dddot{e}_x \\ \dddot{e}_y \end{bmatrix} + K_p \begin{bmatrix} \ddot{e}_x \\ \ddot{e}_y \end{bmatrix}. \qquad (28)$$

Equations (25)–(28) represent the xy position controller and the flat mapping between (x, y) and (ϕ, θ). The proposed controller is able to track the reference x_{ref}, y_{ref}, and z_{ref} as well as ψ_{ref}. The proposed architecture has been exploited in [45] to control the position and orientation of the octocopter.

It is necessary to emphasize that the reference trajectories of the x and y position coordinates must be at least four times differentiable, while trajectories for the altitude z and orientation ψ must have the first and the second derivation. These trajectories are provided by an adequate motion planning algorithm, while references for ϕ and θ orientation coordinates are provided as the output of the xy controller.

To illustrate the proposed control architecture for the position and orientation tracking, we can consider tracking the Vivian curve (octocopter with PPN-NPPNN configuration) in the three-dimensional space represented in Figure 3. One can observe a small-error tracking of the referent trajectory. It is important to note that the simulation was conducted with a healthy system, that is without any fault. For the case when the 3rd DC motor is in a fault state, it can be seen from Figures 4 and 5 that the presented control architecture fails to track the reference trajectory.

3. Analysis of fault-dependent MAV manoeuvrability

For the purpose of developing a motion planner and estimating the possibility of completing a pre-planned mission, it is necessary to determine whether the system is capable of generating necessary thrust and torques, to be able to reach the points generated by the
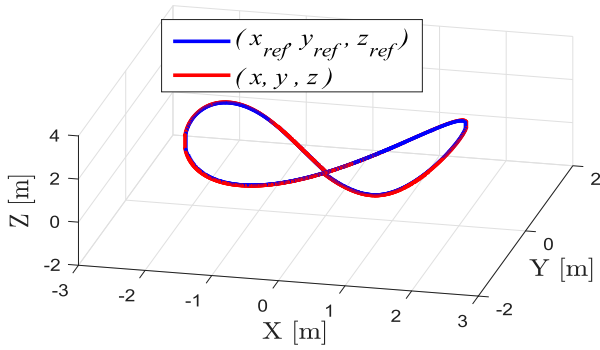
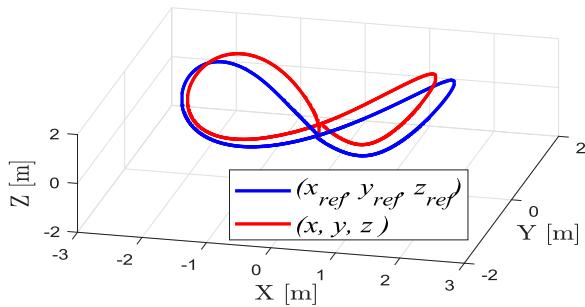**Figure 3.** Trajectory tracking with a healthy system (an octocopter with the PPNNPPNN configuration).

Figure 4. Trajectory tracking with a faulty system (an octocopter with the PPNNPPNN configuration and the 3rd motor in a faulty state).
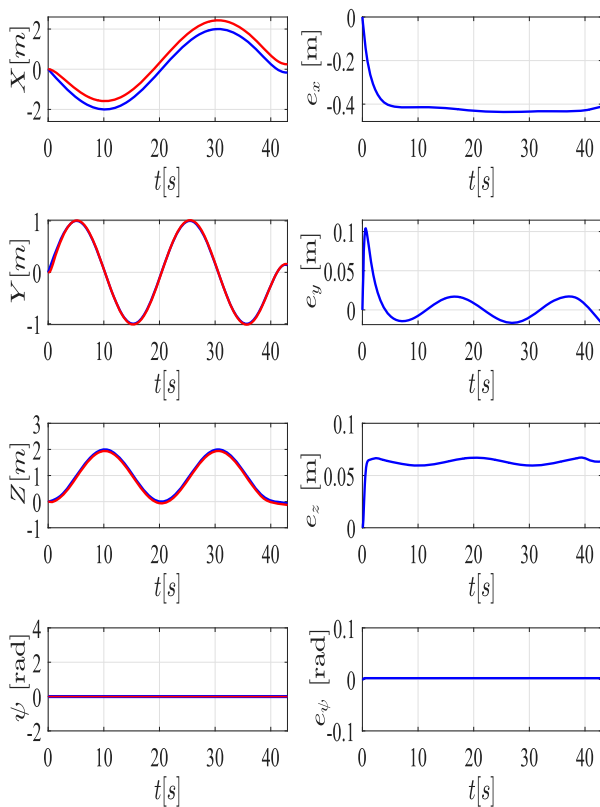

Figure 5. Trajectory tracking with a faulty system presented for each position and orientation (an octocopter with the PPNNPPNN configuration and the 3rd motor in a faulty state). Left: reference and achieved values, Right: tracking errors.

motion planner with available DC motors. As shown in Section 2.4, regardless whether the system has a redundant actuator or not, it is possible to have a case when the control algorithm is not able to track the referent trajectory (fault state caused by a failure occurred on one of the motors). This problem has been addressed in [21–26] for a MAV with a fixed rotor and classical configurations (quadcopter, hexacopter and octocopter). In addition, in [27] and [28], the controllability analysis has been considered for a MAV designed with a tilted rotor, while for non-classical (coaxial) octocopters the analysis has been given in [49]. In this paper, an empirical method is developed which can be used for any MAV configuration designed with different number of rotors and their directions of rotation.

3.1. Fault-dependent admissible set of thrust force and torques

To find the admissible set of thrust force and torques in control space, it is necessary to check whether the system can reach and stay in hovering point without any rotation. For illustration purposes, consider again an octocopter with the PNPNPNPN configuration. The relation between the control inputs u (the reference thrust force T and torques τ) and the rotation velocity $\mathbf{\Omega}_s$ of DC motors is given with $u = A\mathbf{\Omega}_s$ (see Section 2), where the control vector u is represented by

$$u = \begin{bmatrix} T & \boldsymbol{\tau} \end{bmatrix}^T = \begin{bmatrix} T & \tau_{x_o} & \tau_{y_o} & \tau_{z_o} \end{bmatrix}^T, \quad (29)$$

where $\mathbf{\Omega}_s \in D_{\Omega_s} \subset \mathbb{R}^8$ and $u \in D_u \subset \mathbb{R}^4$. The set D_{Ω_s} is defined based on the velocity constraints of DC motors

$$0 \leq \Omega_i^2 \leq \Omega_{\max}^2, \quad i = \overline{1, \ldots, 8}. \quad (30)$$

Knowing that the DC motor velocity is limited between 0 and $\omega_{\max}$ (30) and the mapping is defined by the linear relation $u = A\mathbf{\Omega}_s$, it means that the set D_u represents a polytope in space $\mathbb{R}^4$. If velocities of all DC motors are equal to zero when all components of the control input are zero-valued, we get the first point in control space. If we now set the rotation velocity of the first DC motor to its maximum value, we get the second point in control space. The number of these combinations is 2^{2n}, where n is the number of DC motors. For the octocopter example, it is possible to construct one hyper-plane for each tuple (4 control components) of the total of 256 points. However, only those hyper-planes that form an outer region are of interest. In this way, one can construct a convex polytope-like admissible region in four-dimensional space. Since the obtained region is constructed in four-dimensional control space, we consider the case for $\tau_z = 0$ for illustration purposes. In this case, an orthogonal projection of the polytope of the set D_u (with coordinates T, τ_x and τ_y) is shown in Figure 6.
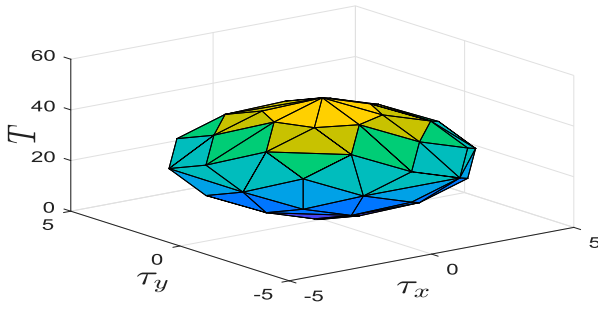
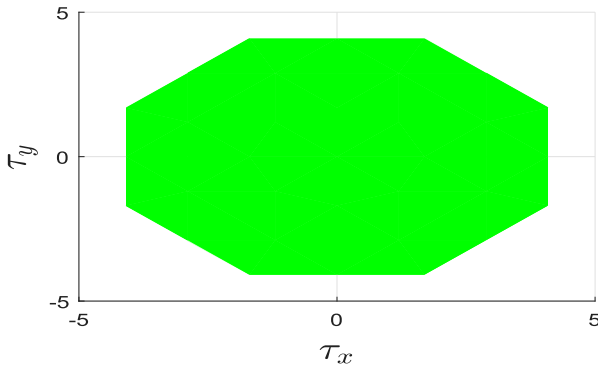**Figure 6.** Representation of the four-dimensional admissible region in three-dimensional space when $\tau_z = 0$.

Figure 7. The projection of torques τ_x, τ_y onto the plane, $T = mg$, along its orthogonal direction for a healthy octocopter with the PNPNPNPN configuration.
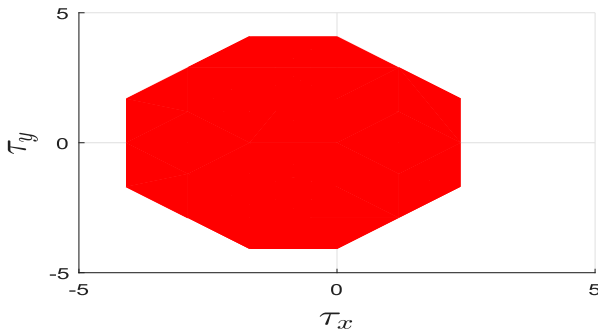

Figure 8. The projection of torques τ_x and τ_y onto the plane, $T = mg$, along its orthogonal direction, when a failure occurred on 1st DC motor for octocopter with the PNPNPNPN configuration.

In order for the octocopter system to be stable at the hovering point, it is obvious that the thrust force should be $T = mg$. If the represented three-dimensional set from Figure 6 is projected onto the plane $T = mg$, then the projection is shown in Figure 7. It can be observed that the torques τ_x and τ_y have symmetric values and that they are linearly dependent, meaning that is not possible to simultaneously reach maximal values of the torques τ_x and τ_y. Consider now that the DC motor 1 is in a fault state. The projection of the torques τ_x and τ_y onto the plane $T = mg$ is shown in Figure 8. In case of a double fault (DC motors 1 and 2), the projection of the torques onto the plane $T = mg$ is shown in Figure 9.

It can be seen from Figure 8 and 9 that the admissible set for τ_x and τ_y is reduced with respect to a healthy system shown in Figure 7. Depending on the type and
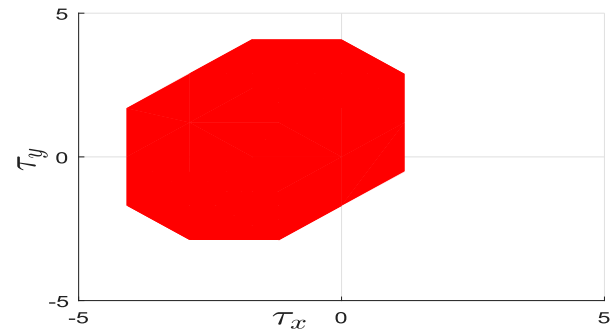
Figure 9. The projection of torques τ_x and τ_y onto the plane, $T = mg$, along its orthogonal direction, when a double fault occurs on DC motors 1 and 2 for octocopter with the PNPNPNPN configuration.

combination of faults occurred, some of the planned manoeuvers for stabilizing a hovering state will not be possible. The obtained admissible sets have been illustrated only to understand that each DC motor has a different effect on the generation of thrust T and the torques τ_x, τ_y and τ_z.

3.2. Fault-dependent controllability test procedure

To understand whether an MAV is controllable (or at least stabilizable) in case of a single fault (or any multiple faults combination), we propose a testing procedure to check whether a hovering state is reachable or not. We say that a MAV is controllable with respect to a certain state in case there is a control input that moves the MAV to that state. In case the controller is not capable to influence yaw-torque, τ_z we say it is stabilizable in that state. The latter means that the vehicle is capable to remain at the given position only by rotating around z-axes.

The task of the control allocation algorithm is to distribute DC motor velocities $\mathbf{\Omega}_s$ to each motor in order to achieve the referent thrust force and torques for reaching a waypoint generated by a motion planner. For all MAVs for which $n > 2$, there is an infinite number of realization to achieve the same result in case a feasible solution exists. To check whether a feasible solution exists, we define the optimization problem to generate the optimal solution $\mathbf{\Omega}_s^*$ that minimizes the square-error between the reference $\mathbf{u}_{ref}$ and achieved control over the feasible control region ($0 \le \mathbf{\Omega}_s \le \omega_{max}^2$)

$$\mathbf{\Omega}_s^* = \underset{0 \le \mathbf{\Omega}_s \le \omega_{max}^2}{\text{argmin}} (\|\boldsymbol{e_p}\|^2)$$
$$= \underset{0 \le \mathbf{\Omega}_s \le \omega_{max}^2}{\text{argmin}} (\|\boldsymbol{u}_{ref} - A\mathbf{\Omega}_s\|^2). \quad (31)$$

In case there is no solution to the optimization problem (31), the MAV is unstable. In case (31), with a feasible solution yielding zero-valued $\boldsymbol{e_p}$, the MAV is controllable. In case there is a feasible solution without zero-valued $\boldsymbol{e_p}$, we have two additional cases. If we allow

Table 1. Quadcopter: Analysis of single-fault cases.

Fault	$e_p = u - A\Omega_s$
Motor 1	$[0.0 \quad -0.03 \quad 0.0 \quad 0.21]^T$
Motor 2	$[0.0 \quad 0.0 \quad 0.03 \quad -0.21]^T$
Motor 3	$[0.0 \quad 0.03 \quad 0.0 \quad 0.21]^T$
Motor 4	$[0.0 \quad 0.0 \quad -0.03 \quad -0.21]^T$

Table 2. Hexacopter: Analysis of single-fault cases for the PNPNPN and PPNNPN configurations.

Fault	e_p for PNPNPN configuration	e_p for PPNNPN configuration
M1	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$
M2	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$
M3	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$
M4	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$
M5	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$
M6	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$

Table 3. Hexacopter: Analysis of double-fault cases for the PNPNPN configuration.

Fault	$e_p = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T$	$e_p = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$
M12	$[0.0 \quad -0.44 \quad 0.25 \quad 0.0]^T$	$[0.0 \quad -0.44 \quad 0.25]^T$
M13	$[0.0 \quad 0.0 \quad 0.03 \quad 0.13]^T$	$[0.0 \quad 0.0 \quad 0.0]^T$
M14	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$	$[0.0 \quad 0.0 \quad 0.0]^T$
M15	$[0.0 \quad -0.2 \quad 0.0 \quad 0.12]^T$	$[0.0 \quad 0.0 \quad 0.0]^T$
M16	$[0.0 \quad -0.6 \quad -0.3 \quad 0.0]^T$	$[0.0 \quad -0.44 \quad -0.25]^T$
M23	$[0.0 \quad 0.0 \quad 0.51 \quad 0.0]^T$	$[0.0 \quad 0.5 \quad 0.0]^T$
M24	$[0.0 \quad 0.24 \quad 0.42 \quad -0.13]^T$	$[0.0 \quad 0.0 \quad 0.0]^T$
M25	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$	$[0.0 \quad 0.0 \quad 0.0]^T$
M26	$[0.0 \quad -0.04 \quad 0.0 \quad -0.12]^T$	$[0.0 \quad 0.0 \quad 0.0]^T$
M34	$[0.0 \quad 0.64 \quad 0.36 \quad 0.0]^T$	$[0.0 \quad 0.44 \quad 0.25]^T$
M35	$[0.0 \quad 0.04 \quad 0.0 \quad 0.12]^T$	$[0.0 \quad 0.0 \quad 0.0]^T$
M36	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$	$[0.0 \quad 0.0 \quad 0.0]^T$
M45	$[0.0 \quad 0.04 \quad -0.25 \quad 0.12]^T$	$[0.0 \quad 0.61 \quad -0.35]^T$
M46	$[0.0 \quad 0.02 \quad -0.03 \quad -0.12]^T$	$[0.0 \quad 0.0 \quad 0.0]^T$
M56	$[0.0 \quad 0.0 \quad -0.5 \quad 0.0]^T$	$[0.0 \quad 0.0 \quad -0.51]^T$

free movements around z-axis, we exclude the torque τ_z from u_{ref} and repeat the optimization. In case we get a zero-valued error, the MAV is stabilizable, although it will be rotating around z-axis. Otherwise, the system will be unstable.

By checking whether the hovering point is reachable or not, we can understand if the MAV is capable for given mission regardless the faults occurred. In the following subsection, we thoroughly analyse controllability of different types of MAVs by examining different single and multiple faults.

3.3. Fault-dependent controllability analysis for quadcopter, hexacopter and octocopter

3.3.1. Quadcopter

First, we consider quadcopter with DC motors without fault states. The quadcopter has the following parameters [42]: $m_o = 1.32$ [kg], $l = 0.211$ [m], $I_{xx} = I_{yy} = 0.0128$ [kg m^2], $I_{zz} = 0.0239$ [kg m^2], $I_{zzm} = 4.3 \cdot 10^{-5}$ [kg m^2], $b = 9.9865 \cdot 10^{-6}$ [N s^2/rad^2], $d = 1.5978 \cdot 10^{-7}$ [N m s^2/rad^2], $\omega_{max} = 840$ [rad/s].

Solving the optimization problem (31) where the referent thrust force and the torques are given with $u = [T \quad \tau]^T = [mg \, 0 \, 0 \, 0]^T$, we get $\omega_1 = \omega_2 = \omega_3 = \omega_4 = 569.35$ [rad/s] and $e_p = [0 \, 0 \, 0 \, 0]^T$. This means that the hovering point is reached and it is possible to stabilize the MAV at this point. Let now the same optimization problem be considered for the quadcopter case for each possible single fault with the same reference $u = [T \, \tau]^T = [mg \, 0 \, 0 \, 0]^T$. The results of these optimizations are shown in Table 1. As it can be seen, the quadcopter cannot be fully controlled at the hovering point for any single failure. Such cases are indicated by red colour in Table 1. For example, in case of a failure occurred on the DC motor 1, quadcopter is stabilizable, but there is a constant rotation in the negative direction about the x-axis as well as intense rotation about the z-axis. The results obtained were expected and are in line with the state-of-the-art work. In [25], the authors have shown that the quadcopter does not have a redundant configuration and its controllability will be lost in case any of DC motors fails.

3.3.2. Hexacopter

In this subsection, we analyse two types of hexacopter with PNPNPN and PPNNPN rotation configuration. The remaining parameters of the hexacopter are: $m_o = 1.54$ [kg], $l = 0.211$ [m], $I_{xx} = I_{yy} =$

0.0168 [kg m^2], $I_{zz} = 0.0308$ [kg m^2], $I_{zzm} = 2 \cdot 10^{-5}$ [kg m^2], $b = 8.5485 \cdot 10^{-6}$ [N s^2/rad^2], $d = 1.3678 \cdot 10^{-7}$ [N ms^2/rad^2], $\omega_{max} = 874$ [rad/s].

First, we consider the controllability of both configurations without fault states by using the same control reference $u = \begin{bmatrix} T & \tau \end{bmatrix}^T = \begin{bmatrix} mg & 0 & 0 & 0 \end{bmatrix}^T$. As it can be seen from Table 2, both versions of the hexacopter are inherently fault-tolerant with respect to a single DC motor failure. For this reason, we will consider different double-fault cases (left columns in Tables 3 and 4) and the cases where the hovering point is stabilizable but not controllable when the reference is used in the form $u = \begin{bmatrix} T & \tau_x & \tau_y \end{bmatrix}^T = \begin{bmatrix} mg & 0 & 0 \end{bmatrix}^T$ (right columns in Tables 3 and 4). Although the MAV may loose controlability, the latter case is important to be discovered since a safety landing can be performed which can protect the vehicle and its equipment from potential damage.

From the results presented in Tables 3 and 4 related to the two different hexacopter orientation configurations, 80% of total double-fault cases lead to the loss of controllability (cases with at least one non-zero values in left columns), while 40% cases are unstable without possibility for a safe landing (cases with at least one non-zero values in right columns). This further means that the hexacopter will have a potential to continue the mission only in three cases (cases with all zero values in left columns) and to be safe in 60% (cases with all zero values in right columns). One can also conclude that the hexacopter is single-fault-tolerant, while it is quite

Table 4. Hexacopter: Analysis of double-fault cases for the PPNNPN configuration.

Fault	$e_p = [0 \quad 0 \quad 0 \quad 0]^T$	$e_p = [0 \quad 0 \quad 0]^T$
M12	$[0.0 \quad -0.43 \quad 0.27 \quad 0.18]^T$	$[0.0 \quad -0.44 \quad 0.25]^T$
M13	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$	$[0.0 \quad 0.0 \quad 0.0]^T$
M14	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$	$[0.0 \quad 0.0 \quad 0.0]^T$
M15	$[0.0 \quad 0.0 \quad -0.01 \quad 0.07]^T$	$[0.0 \quad 0.0 \quad 0.0]^T$
M16	$[0.0 \quad -0.44 \quad -0.25 \quad -0.15]^T$	$[0.0 \quad -0.44 \quad -0.25]^T$
M23	$[0.0 \quad 0.0 \quad 0.51 \quad 0.0]^T$	$[0.0 \quad 0.0 \quad 0.5]^T$
M24	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$	$[0.0 \quad 0.0 \quad 0.0]^T$
M25	$[0.0 \quad 0.0 \quad 0.0 \quad 0.04]^T$	$[0.0 \quad 0.0 \quad 0.0]^T$
M26	$[0.0 \quad 0.0 \quad 0.0 \quad -0.05]^T$	$[0.0 \quad 0.0 \quad 0.0]^T$
M34	$[0.0 \quad 0.43 \quad 0.27 \quad -0.02]^T$	$[0.0 \quad 0.44 \quad 0.25]^T$
M35	$[0.0 \quad 0.0 \quad 0.0 \quad 0.05]^T$	$[0.0 \quad 0.0 \quad 0.0]^T$
M36	$[0.0 \quad 0.0 \quad 0.0 \quad -0.05]^T$	$[0.0 \quad 0.0 \quad 0.0]^T$
M45	$[0.0 \quad 0.44 \quad -0.25 \quad 0.14]^T$	$[0.0 \quad 0.44 \quad -0.25]^T$
M46	$[0.0 \quad 0.0 \quad 0.0 \quad -0.08]^T$	$[0.0 \quad 0.0 \quad 0.0]^T$
M56	$[0.0 \quad -0.51 \quad 0.0 \quad 0.0]^T$	$[0.0 \quad 0.0 \quad -0.51]^T$

Table 5. Octocopter: Analysis of single-fault cases for the PNPNPNPN and PPNNPPNN configurations.

Fault	e_p for PNPNPNPN config.	e_p for PPNNPPNN config.
M1	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$
M2	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$
M3	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$
M4	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$
M5	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$
M6	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$
M7	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$
M8	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$

Table 6. Octocopter: Analysis of double-fault cases for the PNPNPNPN configuration.

Fault	$e_p = [0 \quad 0 \quad 0 \quad 0]^T$	$e_p = [0 \quad 0 \quad 0]^T$
M12	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$	$[0.0 \quad 0.0 \quad 0.0]^T$
M13	$[0.0 \quad 0.0 \quad 0.0 \quad 0.01]^T$	$[0.0 \quad 0.0 \quad 0.0]^T$
M14	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$	$[0.0 \quad 0.0 \quad 0.0]^T$
M15	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$	$[0.0 \quad 0.0 \quad 0.0]^T$
M16	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$	$[0.0 \quad 0.0 \quad 0.0]^T$
M17	$[0.0 \quad 0.0 \quad 0.0 \quad 0.01]^T$	$[0.0 \quad 0.0 \quad 0.0]^T$
M18	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$	$[0.0 \quad 0.0 \quad 0.0]^T$
M23	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$	$[0.0 \quad 0.0 \quad 0.0]^T$
M24	$[0.0 \quad 0.0 \quad 0.0 \quad -0.01]^T$	$[0.0 \quad 0.0 \quad 0.0]^T$
M25	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$	$[0.0 \quad 0.0 \quad 0.0]^T$
M26	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$	$[0.0 \quad 0.0 \quad 0.0]^T$
M27	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$	$[0.0 \quad 0.0 \quad 0.0]^T$
M28	$[0.0 \quad 0.0 \quad 0.0 \quad 0.01]^T$	$[0.0 \quad 0.0 \quad 0.0]^T$
M34	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$	$[0.0 \quad 0.0 \quad 0.0]^T$
M35	$[0.0 \quad 0.0 \quad 0.0 \quad 0.01]^T$	$[0.0 \quad 0.0 \quad 0.0]^T$
M36	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$	$[0.0 \quad 0.0 \quad 0.0]^T$
M37	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$	$[0.0 \quad 0.0 \quad 0.0]^T$
M37	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$	$[0.0 \quad 0.0 \quad 0.0]^T$
M45	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$	$[0.0 \quad 0.0 \quad 0.0]^T$
M46	$[0.0 \quad 0.0 \quad 0.0 \quad -0.01]^T$	$[0.0 \quad 0.0 \quad 0.0]^T$
M47	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$	$[0.0 \quad 0.0 \quad 0.0]^T$
M48	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$	$[0.0 \quad 0.0 \quad 0.0]^T$
M56	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$	$[0.0 \quad 0.0 \quad 0.0]^T$
M57	$[0.0 \quad 0.0 \quad 0.0 \quad 0.01]^T$	$[0.0 \quad 0.0 \quad 0.0]^T$
M58	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$	$[0.0 \quad 0.0 \quad 0.0]^T$
M67	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$	$[0.0 \quad 0.0 \quad 0.0]^T$
M68	$[0.0 \quad 0.0 \quad 0.0 \quad -0.01]^T$	$[0.0 \quad 0.0 \quad 0.0]^T$
M78	$[0.0 \quad 0.0 \quad 0.0 \quad 0.0]^T$	$[0.0 \quad 0.0 \quad 0.0]^T$

sensitive to double faults in terms of mission execution. However, it possesses some safety robustness. Since the results are equal, there is no advantage for using any particular hexacopter configuration.

3.3.3. Octocopter

In this subsection we address two different orientation configurations, PNPNPNPN and PPNNPPNN, for different single and double faults. The octocopter has the following parameters [41] $m_o = 1.8$ [kg], $l = 0.211$ [m], $I_{xx} = I_{yy} = 0.0429$ [kg m^2], $I_{zz} = 0.0748$ [kg m^2], $I_{zzm} = 2 \cdot 10^{-5}$ [kg m^2], $b = 8.5485 \cdot 10^{-6}$ [N s^2/rad^2], $d = 1.3678 \cdot 10^{-7}$ [N ms^2/rad^2], $\omega_{\max} = 874$ [rad/s].

The analysis of single-fault cases for both configurations is presented in Table 5, while for double-fault cases in Tables 6 and 7. It is evident that both octocopter configurations are fully insensitive with respect to single failures in terms of their potential to continue the mission execution. From the results presented in Table 6 related to PNPNPNPN configurations, 28% of total double-fault cases lead to the loss of controllability (cases with at least one non-zero value in left column), while there are no unstable cases without possibility for a safe landing. This means that this octocopter configuration will have a potential to continue the mission in 72% of cases (cases with all zero values in left columns) and to be safe in 100% (cases with all zero values in right columns). For the hexacopter PPNNPPNN configuration we have 14% of controllability loss, no unstable cases, 86% potential to continue mission and 100% safety (Table 7). This analysis shows that a careful selection of the octocopter configuration may additionally influence the MAV overall manoeuvrability and keep the MAV ready to execute the mission under variety of faulty states. For instance, when the probability of double fault is high, we can increase mission reliability by choosing PPNNPPNN configuration. However, the performed analysis can be generalized for any MAV with $2n$ pair of DC motors constructed within a planar plane. Triple or quadruple faults can be also analysed in the same way. However, the probability of such occurrence is much lower than for single or double faults, so they are not considered in this paper.

4. RLS-based fault-tolerant tracking control

In order to exploit the results of the fault-dependent MAV manoeuvrability analysis, presented in Section 3 and to provide a unique testbed for performance analysis of considered motion planners, a mechanism for failure detection and fault-tolerant tracking control is needed. In this section, we briefly describe the approach we previously presented in [45].

Table 7. Octocopter: Analysis of double-fault cases for the PPN-NPPNN configuration.

Fault	$e_p = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T$	$e_p = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$
M12	$\begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.13 \end{bmatrix}^T$	$\begin{bmatrix} 0.0 & 0.0 & 0.0 \end{bmatrix}^T$
M13	$\begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}^T$	$\begin{bmatrix} 0.0 & 0.0 & 0.0 \end{bmatrix}^T$
M14	$\begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}^T$	$\begin{bmatrix} 0.0 & 0.0 & 0.0 \end{bmatrix}^T$
M15	$\begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}^T$	$\begin{bmatrix} 0.0 & 0.0 & 0.0 \end{bmatrix}^T$
M16	$\begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}^T$	$\begin{bmatrix} 0.0 & 0.0 & 0.0 \end{bmatrix}^T$
M17	$\begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}^T$	$\begin{bmatrix} 0.0 & 0.0 & 0.0 \end{bmatrix}^T$
M18	$\begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}^T$	$\begin{bmatrix} 0.0 & 0.0 & 0.0 \end{bmatrix}^T$
M23	$\begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}^T$	$\begin{bmatrix} 0.0 & 0.0 & 0.0 \end{bmatrix}^T$
M24	$\begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}^T$	$\begin{bmatrix} 0.0 & 0.0 & 0.0 \end{bmatrix}^T$
M25	$\begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}^T$	$\begin{bmatrix} 0.0 & 0.0 & 0.0 \end{bmatrix}^T$
M26	$\begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}^T$	$\begin{bmatrix} 0.0 & 0.0 & 0.0 \end{bmatrix}^T$
M27	$\begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}^T$	$\begin{bmatrix} 0.0 & 0.0 & 0.0 \end{bmatrix}^T$
M28	$\begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}^T$	$\begin{bmatrix} 0.0 & 0.0 & 0.0 \end{bmatrix}^T$
M34	$\begin{bmatrix} 0.0 & 0.01 & 0.02 & -0.13 \end{bmatrix}^T$	$\begin{bmatrix} 0.0 & 0.0 & 0.0 \end{bmatrix}^T$
M35	$\begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}^T$	$\begin{bmatrix} 0.0 & 0.0 & 0.0 \end{bmatrix}^T$
M36	$\begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}^T$	$\begin{bmatrix} 0.0 & 0.0 & 0.0 \end{bmatrix}^T$
M37	$\begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}^T$	$\begin{bmatrix} 0.0 & 0.0 & 0.0 \end{bmatrix}^T$
M37	$\begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}^T$	$\begin{bmatrix} 0.0 & 0.0 & 0.0 \end{bmatrix}^T$
M45	$\begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}^T$	$\begin{bmatrix} 0.0 & 0.0 & 0.0 \end{bmatrix}^T$
M46	$\begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}^T$	$\begin{bmatrix} 0.0 & 0.0 & 0.0 \end{bmatrix}^T$
M47	$\begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}^T$	$\begin{bmatrix} 0.0 & 0.0 & 0.0 \end{bmatrix}^T$
M48	$\begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}^T$	$\begin{bmatrix} 0.0 & 0.0 & 0.0 \end{bmatrix}^T$
M56	$\begin{bmatrix} 0.0 & 0.02 & -0.01 & 0.13 \end{bmatrix}^T$	$\begin{bmatrix} 0.0 & 0.0 & 0.0 \end{bmatrix}^T$
M57	$\begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}^T$	$\begin{bmatrix} 0.0 & 0.0 & 0.0 \end{bmatrix}^T$
M58	$\begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}^T$	$\begin{bmatrix} 0.0 & 0.0 & 0.0 \end{bmatrix}^T$
M67	$\begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}^T$	$\begin{bmatrix} 0.0 & 0.0 & 0.0 \end{bmatrix}^T$
M68	$\begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}^T$	$\begin{bmatrix} 0.0 & 0.0 & 0.0 \end{bmatrix}^T$
M78	$\begin{bmatrix} 0.0 & -0.01 & -0.03 & -0.13 \end{bmatrix}^T$	$\begin{bmatrix} 0.0 & 0.0 & 0.0 \end{bmatrix}^T$

4.1. Fault-tolerant PD tracking control

To design fault-tolerant control, it is necessary to include information about faulty states of DC motors into the actuation matrix. Now we can rewrite the actuation matrix as:

$$u = A \text{diag}(\mathbf{\Omega}_s)\, \theta, \tag{32}$$

where $\theta = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 & \theta_4 & \theta_5 & \theta_6 & \theta_7 & \theta_8 \end{bmatrix}^T$ represents the faulty states of DC motors. The coefficients $0 \leq \theta_i \leq 1$ ($i = \overline{1, \ldots, 8}$) represent the failure level of related DC motor, where $\theta = 1$ represents fully available motor, $\theta = 0$ failed motor while all values in between represent a partial loss of DC motor functionality. If we rewrite the actuation matrix as $A = \begin{bmatrix} A_T & A_1 & A_2 & A_3 \end{bmatrix}^T$, the components of the τ_x, τ_y and τ_z vectors of the controllable control signals u can be represented as weighted scalar products in the form

$$T = A_T \text{diag}(\mathbf{\Omega}_s)\, \theta$$
$$\tau_x = A_1 \text{diag}(\mathbf{\Omega}_s)\, \theta$$
$$\tau_y = A_2 \text{diag}(\mathbf{\Omega}_s)\, \theta$$
$$\tau_z = A_3 \text{diag}(\mathbf{\Omega}_s)\, \theta. \tag{33}$$

As we can see from (33), the control output can be represented by its four components that have a linear

dependence. Based on these values, it is possible to estimate the parameter θ in a least-squares manner. For the estimation, it is necessary to know all parameters in (33). The basic requirement is that the values of the actuation matrix A are a priori known. For the detection and isolation of failures, we only use gyroscopic data for their inherent accuracy of measurement so that the first equality from (33) is not necessary, so the final prediction is

$$\tau_x = A_1 \text{diag}(\mathbf{\Omega}_s)\, \theta$$
$$\tau_y = A_2 \text{diag}(\mathbf{\Omega}_s)\, \theta \tag{34}$$
$$\tau_z = A_3 \text{diag}(\mathbf{\Omega}_s)\, \theta.$$

If we have N measurements at time instances 1 to N, then for each sample $i = \overline{1, \ldots, N}$, based on (34), we have the following model to predict the output

$$\hat{\tau}_x(i) = \mathbf{\Psi}_1^r(i)\theta,$$
$$\hat{\tau}_y(i) = \mathbf{\Psi}_2^r(i)\theta, \tag{35}$$
$$\hat{\tau}_z(i) = \mathbf{\Psi}_3^r(i)\theta,$$

where each regressor of output models is defined as

$$\mathbf{\Psi}_1^r(i) = A_1 \text{diag}(\mathbf{\Omega}_s(i)),$$
$$\mathbf{\Psi}_2^r(i) = A_2 \text{diag}(\mathbf{\Omega}_s(i)), \tag{36}$$
$$\mathbf{\Psi}_3^r(i) = A_3 \text{diag}(\mathbf{\Omega}_s(i)).$$

Using the property that each tensor can be represented as a matrix using the skew-symmetric matrix S defined with (8), we can represent the inertia tensor $J = \text{diag}\left(\begin{bmatrix} I_{xx} & I_{yy} & I_{zz} \end{bmatrix}\right)$ as a symmetric matrix. Now, we can rewrite (9) as

$$\tau_x(i) = I_{xx}\dot{P}(i) - (I_{yy} - I_{zz})Q(i)R(i),$$
$$\tau_y(i) = I_{yy}\dot{Q}(i) - (I_{zz} - I_{xx})P(i)R(i), \tag{37}$$
$$\tau_z(i) = I_{zz}\dot{R}(i) - (I_{xx} - I_{yy})P(i)Q(i).$$

We can now formulate the FDI technique for the propulsion system as an recursive least square (RLS) estimation problem of the rotor capacity vector θ in the following way:

$$\tau = \begin{bmatrix} \tau_x(1) & \tau_y(1) & \tau_z(1) & \cdots & \tau_x(N) & \tau_y(N) & \tau_z(N) \end{bmatrix}^T, \tag{38}$$

where the data matrix $\mathbf{\Psi}^r$ has the form

$$\mathbf{\Psi}^r = [\mathbf{\Psi}_1^{rT}(1) \quad \mathbf{\Psi}_2^{rT}(1) \quad \mathbf{\Psi}_3^{rT}(1) \quad \cdots$$
$$\mathbf{\Psi}_1^{rT}(N) \quad \mathbf{\Psi}_2^{rT}(N) \quad \mathbf{\Psi}_3^{rT}(N)]^T. \tag{39}$$

Now, by using a classical non-recursive least-square method [50], we can express the coefficient $\hat{\theta}$ as

$$\hat{\theta} = \left(\mathbf{\Psi}^{rT}\mathbf{\Psi}^r\right)^{-1}\mathbf{\Psi}^{rT}\tau \tag{40}$$

In this paper, we use the RLS method based on (40). It is necessary to emphasize that the proposed technique for the RLS problem has a linear configuration,

while obtaining values for the vector rotor capacity comes from a MAV model, which is nonlinear (9). The proposed method applies to any type of MAV with $2n$ rotors mounted in a planar plane. The values for θ_i ($i = \overline{1, \dots, 8}$) are not constant all the time. To apply the proposed technique to an MAV, the RLS with a forgetting factor is used. The values for θ_i obtained by the proposed RLS algorithm are used as a feedback to update the actuation matrix. Note that θ and $\boldsymbol{\Omega}_s$ represent vectors of the same size, so (32) can be rewrite as

$$\boldsymbol{u} = \boldsymbol{A}\mathrm{diag}\left(\boldsymbol{\theta}\right)\boldsymbol{\Omega}_s. \qquad (41)$$

Furthermore, by introducing a new matrix $\boldsymbol{B}$ as $\boldsymbol{B} = \boldsymbol{A}\mathrm{diag}(\boldsymbol{\theta})$, we can calculate the velocity for all DC motors that can achieve the reference thrust and the torques as

$$\boldsymbol{\Omega}_s = \boldsymbol{B}^+\boldsymbol{u}, \qquad (42)$$

where $\boldsymbol{B}^+ = \boldsymbol{B}^T(\boldsymbol{B}\boldsymbol{B}^T)^{-1}$.

4.2. Simulation results for fault-tolerant PD tracking control

In this subsection, we present the simulation results for the RLS-based technique for detection and isolation of DC motor failures on MAVs. We used a fault-tolerant PD tracking control system around the hovering configuration. To illustrate that the designed controller can handle a faulty state on MAV, we consider the case described in Section 2.4 (octocopter with the PPNNPPNN configuration). In this case, we have a failure occurred on the 3rd motor at time $t = 5$ [s]. This case can be expressed by the rotor capacity vector $\boldsymbol{\theta} = [\,1\ 1\ 0\ 1\ 1\ 1\ 1\ 1\,]^T$.

It should be emphasized that the presented approach was tested for different classes of possible faults (different numbers of faulty rotors and different values of rotor capacity). For the purpose of the RLS algorithm (38), we need to determine 8 unknowns parameters θ_i, so we need at least 8 equations. It follows, that the number of samples N must be at least $N \geq 8$. However, to eliminate the impact of noise, it is advisable to use a larger number of equations, that is $N \gg 8$. So, we set the forgetting factor to 0.8 in order to take into account the measurements from the previous 0.8 s to provide a sufficient number of samples for the RLS algorithm.

Figures 10 and 11 depict the performance of the proposed RLS-based PD tracking controller with control allocation for a failure occurred on the 3rd motor at $t = 5$ [s]. Figure 12 shows that the relative capacity of each rotor is properly diagnosed.

From Figure 11, we can conclude that the presented RLS controller has an acceptable tracking performance (it is similar to the tracking performance for a healthy MAV). This can be attributed to the inherent fault-tolerability of the octocopter platform,
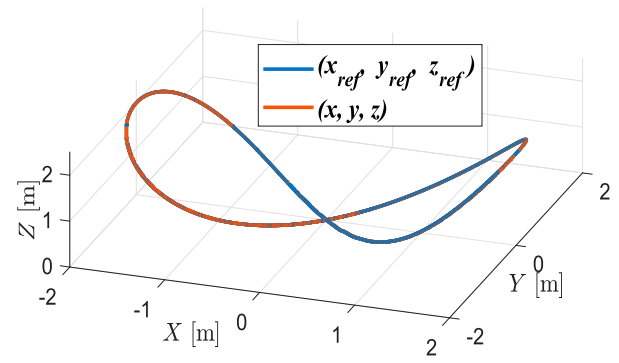

Figure 10. 3D visualization of tracking performance (an octocopter with the PPNNPPNN configuration) for a failure occurred on the 3rd motor at $t = 5$ [s].
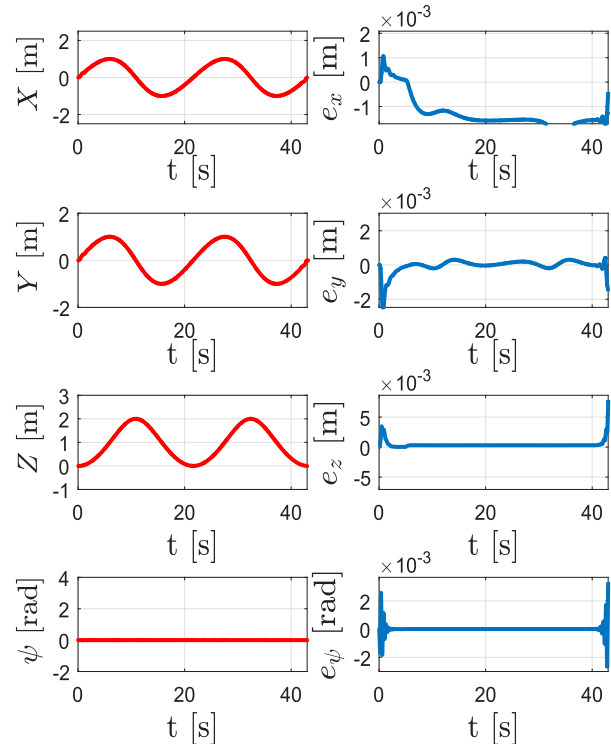

Figure 11. Left: reference values, Right: tracking error for a failure occurred on the 3rd motor at $t = 5$ [s].

as can be expected based on the analysis of fault-dependent manoeuvrability for octocopter provided in Section 3.3.3).

5. Motion planner algorithm based on mission-related fault-tolerant analysis

This section presents a novel motion planner for MAVs based on admissible set of thrust force and torque obtained through fault-dependent manoeuvrability analysis presented in Section 3.

5.1. Presentation of the admissible set of thrust force and torques with a set of inequality constraints

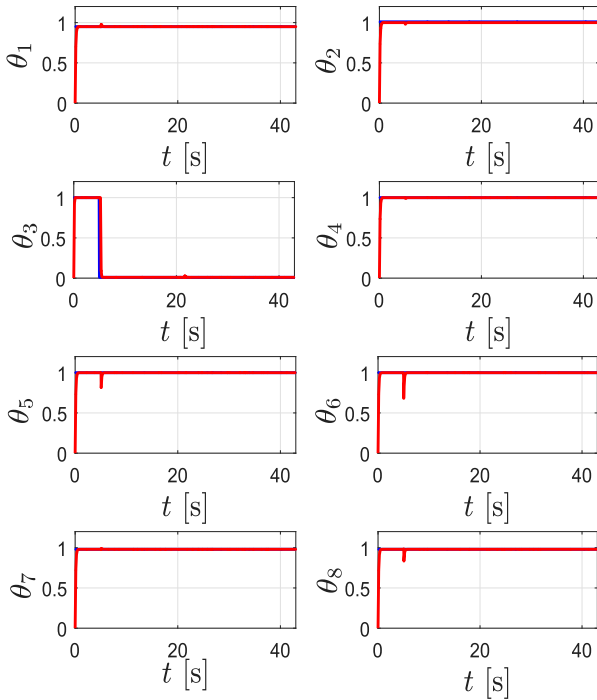As shown in Section 3.1, the admissible set for thrust force and torques can be determined depending on the

Figure 12. Estimation of relative capacity of each rotor during the tracking task.

number of DC motors used, the orientation configuration and the states of DC motors (with or without faults). This admissible set has a convex polytope-like form in four-dimensional space.

Each of the outer sides of the polytope can be represented by its related hyper-plane based on the four points that form that side, that is

$$aT + b\tau_x + c\tau_y + d\tau_z \leqslant e, \qquad (43)$$

where a, b, c, d and e are the slope coefficients of the individual axes. Since the polytope-like admissible set is composed of a large number of such hyper-planes, it can be represented as a set of inequalities that fully describes the admissible set of thrust force and torques for each specific MAV design. For instance, for an octocopter without failure, it turned out, the related polytope can be described by 617 inequalities. In case any of DC motor is in a fault state, the number of inequalities decreases, while in an extreme case when all DC motors are in failure modes, the admissible set is reduced to a single point at origin, hence the system is fully uncontrollable.

The obtained inequalities can be further used in motion planning to generate a feasible trajectory that depends on the initial admissible set of thrust force and torques (or the resulting polytope). In the next subsection, we describe a novel motion planner named as risk-sensitive planner (RSP) based on a careful selection of some of the inequalities that describe the admissible set (only a few of them), where the selection process depends on the required mission.

5.2. Selected optimization framework for motion planning

In Section 2.4, it is shown that for the tracking reference trajectory the position coordinates x and y must be at least four times differentiable, and the heights z and orientations ψ are at least twice differentiable. References to ϕ and θ orientation coordinates are obtained as a consequence of the control of the x and y position coordinates. Accordingly, the height coordinates z and the orientation coordinates ψ behave as a double integrator, that is

$$\ddot{q} = u, \qquad (44)$$

and the x and y position coordinates can be approximated by a quadruple integrator

$$\ddddot{q} = u, \qquad (45)$$

where q, $\dot{q}$ and $\ddot{q}$ are the generalized coordinate, velocity and acceleration respectively, and $q = [x\, y\, z\, \psi]$.

The minimization of acceleration (44) and snap (45) directly yields the minimization of the generalized forces which act on the system. This further results in the minimization of energy consumption while taking into account the constraints imposed on the trajectory. This consequently means that the battery consumption during the mission will be minimal. Detailed description of motion planning based on minimal acceleration and snap can be found in [51].

For this reason, motion planning problem can be described as a fixed finite-time optimization problem given as

$$\underset{0 \leqslant t \leqslant T}{\text{minimize}} \, (\|\ddot{q}\|^2)$$

subject to

$$
\begin{aligned}
q_{min} &\leqslant q \leqslant q_{max} \\
\dot{q}_{min} &\leqslant \dot{q} \leqslant \dot{q}_{max} \\
\ddot{q}_{min} &\leqslant \ddot{q} \leqslant \ddot{q}_{max},
\end{aligned}
\qquad (46)
$$

where the fixed finite-time represents the mission execution time T. The waypoints, as part of the given mission, through which the MAV is supposed to pass should also be included in the optimization framework as desired constraints. One way to include these constraints is to impose hard constraints into the optimization. To ensure that the planner can be risk-averse, if necessary, we need to allow the motion planner to be capable of generating trajectories that may deviate from the waypoints. In addition, the deviation from the waypoints can be also used as a performance measure for the given mission. To do so, we include these constraints into the objective function by penalizing large deviations from the given waypoints as

$$\underset{0 \leqslant t \leqslant T}{\text{minimize}} \, (\|\ddddot{q}\|^2 + \sum_i \alpha_i (\|q - q_i\|^2))$$

subject to

$$q_{\min} \leqslant q \leqslant q_{\max}$$

$$\dot{q}_{\min} \leqslant \dot{q} \leqslant \dot{q}_{\max}$$

$$\ddot{q}_{\min} \leqslant \ddot{q} \leqslant \ddot{q}_{\max} \tag{47}$$

where weights $0 \leq \alpha_i \leq 1$ are used for describing how important it is to pass through some of the waypoints q_i during the mission execution. For the purpose of this work, we set $\alpha_i = 1$ for all i.

Since (19), (20) and (22)) give the relation between T, τ_x, τ_y and τ_z and q, it is now possible to include the inequalities (or some of them) that describe the fault-dependent admissible set into the optimization framework. This gives the final form of the optimization framework used for the proposed RSP motion planner:

$$\underset{0 \leqslant t \leqslant T}{\text{minimize}} \, (\|\ddot{\boldsymbol{q}}\|^2 + \sum_i \alpha_i(\|q - q_i\|^2))$$

subject to

$$q_{\min} \leqslant q \leqslant q_{\max} \tag{48}$$

$$\dot{q}_{\min} \leqslant \dot{q} \leqslant \dot{q}_{\max}$$

$$aT + b\boldsymbol{\tau_x} + c\boldsymbol{\tau_y} + d\boldsymbol{\tau_z} \leqslant e_i$$

In order to take into account any possible failure during the motion planning stage, one can include all constraints related to the admissible set of that failure. In this paper, we call such a planner a risk-conservative planner (RCP). However, the RCP planner would be quite conservative, so in the following subsection we describe how to select some of the inequalities by carefully examination of the given mission to form the RSP planner. It should be noted that the decision on which failures and their related admissible sets to include should follow from the failure mode and effects analysis (FMEA) [21,52].

5.3. Risk-sensitive motion planner based on mission-related fault-tolerant analysis

In this subsection we describe how to take possible failures into account during the motion planning stage by means of their related admissible sets. By doing so, we aim to include the associated risk into the planner in order to increase reliability of the mission execution in terms of satisfactory performance. This will be done at the cost of a much smaller performance deterioration than in case when a RCP planner is used for which all constraints related to risk-dependent admissible sets are included. As expected, the planner will require a bit more time to complete the mission than the RIP planner.

When a fault occurs the MAV may be in a position and orientation such that the control allocation

is capable to produce desired thrust force and torques for the remaining DC motors without any effect on the mission performance. Contrary, the MAV may be in such a position and orientation to significantly deteriorate the performance. The idea behind the proposed RSP motion planner is to carefully select a certain number of fault-dependent inequalities to retain the overall performance of a healthy system as much as possible. By doing so, the planner aims to minimally reduce the vehicle manoeuvrability (much less than in case of the RCP motion planner) in order to decrease those states in which the vehicle might significantly deteriorate the performance when any of selected fault occurs.

The overall steps for mission-related fault-tolerant analysis and for designing the proposed RSP motion planner can be summarized as follows:

(a) Select the failure modes of interest based on the FMEA analysis (e.g. single motor failures).
(b) Determine the minimum mission execution time for the RIP motion planner to achieve a feasible solution (that is, passing through waypoints).
(c) Determine the minimum mission execution times of the RCP motion planner for all selected failure modes to achieve feasible solutions (that is, passing through the waypoints).
(d) Set the maximum time of all minimum times obtained in step to be the mission time in order to ensure that the RCP planner provides feasible solutions for each failure modes.
(e) Find all inequalities associated to the fault dependent admissible sets for each selected failure mode from step (a), which are not satisfied during the mission execution obtained with the RIP motion planer for the given mission time from step (d).
(f) Form the final optimization framework for the RSP motion planner by including all constraints found in step (e). Determine the minimum mission execution time for the RSP motion planner to achieve a feasible solution. This optimization framework represents the proposed RSP motion planner.

The presented design steps can be explained as follows. First, we perform the FMEA analysis in order to find the most critical failure mode that will be taken into account during the planning stage (step (a)). Second, we determined the minimum execution time for the RIP planner (step (b)). The minimum mission execution time represents a time for which the optimization framework still gives a feasible solution, that is, the solution which ensures passing through the waypoints. Since the proposed design steps can be conducted offline, that is before the mission execution, this minimum value can be easily found by incrementally decreasing the time and checking whether the related solution is feasible or not. Then, for all selected failure modes from

Table 8. Performance comparison between the RIP and RCP approaches (Case 1, steps (a), (b) and (c)).

MAV	T [s]	e_R [m]	e_R/N [m]	e_Ψ [rad]	e_Ψ/N [rad]
RIP	16	0.0412	0.0019	4e−12	4e−13
RCP	20	0.0566	0.0027	1e−18	5e−21

Table 9. Performance comparison between the RIP, RCP and RSP approaches for $T = 20$ s (Case 1, steps (d) and (e)).

MAV	T	e_R [m]	e_Ψ [rad]	Violetad constraints
RIP	20	0.038	5e−19	51
RCP	20	0.057	1e−18	0
RSP	20	0.038	1.25e−18	1

Table 10. Performance comparison between the RIP, RCP and RSP approaches for their minimum execution times (Case 2, steps (a), (b) and (c)).

MAV	T [s]	e_R [m]	e_R/N [m]	e_Ψ [rad]	e_Ψ/N [rad]
RIP	16	0.0412	0.0019	4e−12	2e−13
RCP M16	26	0.286	0.014	2e−17	1e−19
RSP	18	0.234	0.011	1.2e−9	6e−11

Table 11. The number of violated inequalities related to double-fault admissible sets (M16) by different planners (Case 2).

MAV	T	Violetad constraints
RIP	26	80
RCP M16	26	0
RSP	26	21

step (a), we determine their minimum execution times (step (c)). In step (d), we select the worst-case (maximum time) from step (c) to be the mission time in order to ensure that all planners provide feasible solutions. This is also important for a fair comparison of all planners by means of the performance measured by deviation from the waypoints. Otherwise, some of the planners would be infeasible. In step (e), we first find the admissible sets for each failure mode and determine their related inequality sets. Then, we test the RIP motion planner, given the mission time from step (d), in order to find only those inequalities which are not possible to satisfy for the considered mission. To do so, we check the thrust force and the torques obtained by the RIP motion planner against the related admissible sets for each failure mode. In step (f), we form the final optimization framework by including a constraint set consisted of the inequalities extracted from step (e) and the admissible set for the healthy system.

5.4. Results for a given mission

The mission is defined in the form of Vivian curve as in proceeding sections. 21 points have been generated uniformly along the curve to define the waypoints. To test the quality of generated trajectories, we use two errors, for position e_R and orientation e_Ψ, as

$$e_R = \sum_i \sqrt{(x_i - x_{ref_i})^2 + (y_i - y_{ref_i})^2 + (z_i - z_{ref_i})^2} \tag{49}$$

$$e_\Psi = \sum_i \sqrt{(\psi_i - \psi_{ref_i})^2}. \tag{50}$$

Case 1: For illustration purposes, we first consider only one single failure on DC motor 1 in order to take it into account in the motion planning stage (step (a)). In step (b) and (c) we determined the minimum times to get a feasible solution for the RIP motion planner, which is 16 [s], and for the RCP motion planner, which is 20 [s] (Table 8).

In accordance to step (d), we then choose the mission time to be 20 [s] for the next step. In accordance with step (e), for the thrust forces and torques obtained by the RIP planner, we find all inequalities from the fault-dependent admissible set which are not satisfied during the mission execution. For the considered example, in the case of the RIP planner, the obtained thrust force and torques violate only two inequalities (out of 440 that describe the admissible set of thrust force

and torques) in 111 cases related to 99 discrete positions along the mission curve (out of possible 210). By including only these two additional constraints into the final optimization (step (f)), we get the RSP planner. We have observed that the RSP planner has violated only one inequality constraint at only one position (Table 9). The comparison of all three planners (RIP, RCP and RSP) for the maximum time obtained (step (d)), is given in Table 9. It can be seen that the RIP and RSP planners obtained similar performances, better than in case of the RCP approach. On the other side, the RIP planner violated the constraints in 51 positions (which might lead to a high deterioration of performance if the fault occurs), while the RSP violated only one inequality constraint at only one position.

Case 2: Consider now a double-fault case (Motor 1 and 6). As expected, we can see from Table 10, that the RIP planner obtained the best performance. The RSP and RCP planner have similar performance, except that the RCP planner needs a bit more time ($T = 26$ s vs. $T = 18$ s) to complete the mission. This is due to a more restrictive set of inequality constraints included in the optimization for the RCP planner. However, in the case when the execution time is set to the maximum time (see Table 11) obtained from these planners ($T = 26$ s), the RIP planner has violated fault-dependent inequality constraints 80 times at 73 positions, while the RSP motion planner only 21 times at 21 positions. This indicates that the RSP motion planner is readier than the RIP planner in case this double fault occurs. Unlike the RIP motion planner, we note here that any higher multiple-faults will substantially decrease the manoeuvrability space for the RCP planner.

Table 12. Performance comparison between the RIP, RCP and RSP planners (Case 3).

MAV	T	e_R [m]	e_R/N [m]	e_Ψ [rad]	e_Ψ/N [rad]
RIP	16	0.0412	0.0019	4e−12	4e−13
RCP M1	20	0.057	0.0027	1e−18	5e−21
RCP M2	20	0.038	0.018	4e−18	2e−19
RCP M3	20	0.038	0.018	7e−12	3.4e−13
RCP M4	28	0.5	0.0238	1.4e−8	6.9e−10
RCP M5	24	0.26	0.012	0.0012	5.9e−4
RCP M6	24	0.43	0.021	9e−12	4.4e−13
RCP M7	20	0.073	0.0035	1.5e−10	7.4e−12
RCP M8	20	0.038	0.018	7e−12	3.4e−13
RSP	28	0.54	0.025	7e−6	3.5e−7

Table 13. The number of violated inequalities related to all single-fault admissible sets (M_i) by the RIP, RCP and RSP planners (Case 3).

MAV	$M1$	$M2$	$M3$	$M4$	$M5$	$M6$	$M7$	$M8$
RIP	52	0	108	206	0	108	24	136
RCP M1	0	0	107	179	0	89	20	116
RCP M2	52	0	110	205	0	108	24	136
RCP M3	53	0	0	185	0	87	23	137
RCP M4	52	0	99	0	0	99	24	136
RCP M5	52	0	101	197	0	99	24	136
RCP M6	52	0	101	197	0	0	24	136
RCP M7	52	0	101	191	0	95	0	134
RCP M8	52	0	108	197	0	108	24	0
RSP	0	0	118	15	0	5	0	0

Case 3: In this case we address all possible single faults (8 for an octocopter). Table 12 shows the results obtained by the RIP, RCP and RSP planners. The RIP and RSP planners as well as different variants of the RCP planner related to different single-faults, that is RCP M_i, $i = \overline{1, \ldots, 8}$. The RCP M_i planner takes into account only the admissible set related to the fault on the ith motor, meaning that the planner is conservatively prepared only for that fault. It should be noted that the RCP planner is not able to take all 8 single faults simultaneously into account during the planning stage since the final admissible set would be an empty set. However, the RSP planner is capable to address all M_i single faults simultaneously. As previously explained, this is possible since the RSP planner takes only a few inequality constraints for the admissible sets related to all M_i faults, in order to form the optimization framework. For this reason, the RSP is capable to provide a feasible solution, unlike the RCP planner. One can observe from Table 12 that the RSP planner needs more time ($T = 28$ s) to complete the mission in a satisfactory manner. This is due to the fact that it is the only planner that takes all 8 single-faults into account. However, for the mission execution time set to the maximum $T = 28$ s, one can see from Table 13 that the number of violated inequalities related to all single-fault admissible sets (M_i) was significantly smaller for the RSP motion planner with respect to other planners. As expected, the RIP planner violated the largest number of those constraints which makes it unprepared for any single-fault occurrence during the mission executions. It is worth mentioning that the constraints of the admissible set

M3 are violated in a huge number except by the RCP planner (RCP M3) that takes into account those constraints in the planning stage. This is probably due to the selected mission which requires such manoeuvers sensitive to those constraints. An additional interesting observation regarding the admissible sets M2 and M5 is that all planners have managed to satisfy all related constraints during the whole mission. Finally, as expected, all RCP planners, RCP M_i, have satisfied all constraints related to their own admissible sets M_i.

6. Conclusions

In order to understand how an unexpected rotor-fault may influence the considered MAV mission, we have provided a procedure for obtaining the reduced control admissible set and illustrated it with three most frequently used types of vehicles, quad-, hexa-, and octocopter for different combinations of rotor-faults. We have also briefly presented an RLS tracking controller used to handle every fault which might potentially occur during the mission execution in order to provide a unique testbed for performance analysis of considered motion planners. We have also shown how the reduced control admissible set can be described by a set of inequality constraints and which optimization framework can be appropriate to design the RSP planner. However, we have illustrated that only a subset from those constraints, which is directly related to the given mission, can be used within the optimization to get a less-restricted form of risk-averse motion planner than in case of a highly conservative approach (RCP). We have also devised steps to design such a risk-motion planner for any MAV.

Disclosure statement

No potential conflict of interest was reported by the author(s).

ORCID

Nedim Osmic http://orcid.org/0000-0001-8132-4418
Adnan Tahirovic http://orcid.org/0000-0003-2049-8523

References

[1] Hassanalian M, Abdelkefi A. Classifications, applications, and design challenges of drones: a review. Progr Aerosp Sci. 2017;91:99–131.
[2] Tomic T, Schmid K, Lutz P, et al. Toward a fully autonomous UAV: research platform for indoor and outdoor urban search and rescue. IEEE Rob Autom Mag. 2012;19(3):46–56.
[3] Blom JD. Unmanned aerial systems: A historical perspective. Vol. 45. Kansas: Combat Studies Institute Press; 2010.
[4] Lindsey Q, Mellinger D, Kumar V. Construction of cubic structures with quadrotor teams. Proceedings of Robotics: Science & Systems VII; London, England. 2011; Cambridge, MA: The Mit Press.

[5] Zhang C, Kovacs JM. The application of small unmanned aerial systems for precision agriculture: a review. Precis Agric. 2012;13(6):693–712.

[6] Maza I, Caballero F, Capitán J, et al. Experimental results in multi- UAV coordination for disaster management and civil security applications. J Intell Rob Syst. 2011;61(1-4):563–585.

[7] Golightly I, Jones D. Visual control of an unmanned aerial vehicle for power line inspection. 12th International Conference on Advanced Robotics, 2005. ICAR'05. Proceedings; Seattle, WA, USA. IEEE; 2005. p. 288–295.

[8] Dios M-D, Ollero A. Automatic detection of windows thermal heat losses in buildings using UAVs. Automation Congress, 2006. WAC'06. World; Budapest, Hungary. IEEE; 2006. p. 1–6.

[9] Metni N, Hamel T. A UAV for bridge inspection: visual servoing control law with orientation limits. Autom Constr. 2007;17(1):3–10.

[10] Fraundorfer F, Heng L, Honegger D. Vision-based autonomous mapping and exploration using a quadrotor MAV. 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); Vilamoura, Portugal. IEEE; 2012. p. 4557–4564.

[11] Tahirovic A. Discussion on: control and navigation in manoeuvres of formations of unmanned mobile vehicles. Eur J Control. 2013;19(2):172.

[12] Tahirovic A, Astolfi A. A convergent solution to the multi-vehicle coverage problem. 2013 American Control Conference; Washington, DC, USA. IEEE; 2013. p. 4635–4641.

[13] Tahirovic A, Brkic M, Bostan A. A receding horizon scheme for constrained multi-vehicle coverage problems. 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC); Budapest, Hungary. IEEE; 2016. p. 004652.

[14] Flener C, Vaaja M, Jaakkola A, et al. Seamless mapping of river channels at high resolution using mobile lidar and UAV-photography. Remote Sens. 2013;5(12):6382–6407.

[15] Michael N, Fink J, Kumar V. Cooperative manipulation and transportation with aerial robots. Auton Robots. 2011;30(1):73–86.

[16] Puri A. A survey of unmanned aerial vehicles (UAV) for traffic surveillance. Tampa (FL): Department of Computer Science and Engineering, University of South Florida; 2005.

[17] Freed M, Harris R, Shafto M. Human-interaction challenges in UAV-based autonomous surveillance. Proceedings of the 2004 Spring Symposium on Interactions Between Humans and Autonomous Systems Over Extended Operations. 2004, Moffett Field, CA, United States.

[18] Girard AR, Howell AS, Hedrick JK. Border patrol and surveillance missions using multiple unmanned air vehicles. 43rd IEEE Conference on Decision and Control, 2004. CDC. Vol. 1. Nassau, Bahamas. IEEE; 2004. p. 620–625.

[19] Kushleyev A, Mellinger D, Powers C, et al. Towards a swarm of agile micro quadrotors. Auton Robots. 2013;35(4):287–300.

[20] Krajník T, Vonásek V, Fišer D. Ar-drone as a platform for robotic research and education. Research and education in robotics-EUROBOT 2011; Berlin, Heidelberg. Springer; 2011. p. 172–186.

[21] Schneider T. Fault-tolerant multirotor systems [MSC thesis]. ETH Zurich: Swiss Federal Institute of Technology; 2011.

[22] Du G, Quan Q, Yang B. Controllability analysis for a class of multirotors subject to rotor failure/wear. Journal of Intelligent & Robotic Systems. 2015;78(Springer): 143–157.

[23] Du G-X, Quan Q, Cai K-Y. Controllability analysis and degraded control for a class of hexacopters subject to rotor failures. J Intell Rob Syst. 2015;78(1):143–157.

[24] Shi D, Yang B, Quan Q. Reliability analysis of multicopter configurations based on controllability theory. 2016 35th Chinese Control Conference (CCC); Chengdu, China. IEEE; 2016. p. 6740–6745 .

[25] Vey D, Lunze J. Structural reconfigurability analysis of multirotor UAVs after actuator failures. 2015 54th IEEE Conference on Decision and Control (CDC); Osaka, Japan. IEEE; 2015. p. 5097–5104.

[26] Lunze J. From fault diagnosis to reconfigurable control: a unified concept. 2016 3rd Conference on Control and Fault-Tolerant Systems (SysTol); Barcelona, Spain. IEEE; 2016. p. 413–421.

[27] Michieletto G, Ryll M, Franchi A. Control of statically hoverable multi-rotor aerial vehicles and application to rotor-failure robustness for hexarotors. 2017 IEEE International Conference on Robotics and Automation (ICRA); Singapore, Singapore. IEEE; 2017. p. 2747–2752.

[28] Mehmood H, Nakamura T, Johnson EN. A maneuverability analysis of a novel hexarotor UAV concept. 2016 International Conference on Unmanned Aircraft Systems (ICUAS); Arlington, VA, USA. IEEE; 2016. p. 437–446.

[29] Mueller MW, D'Andrea R. Stability and control of a quadrocopter despite the complete loss of one, two, or three propellers. 2014 IEEE International Conference on Robotics and Automation (ICRA); Hong Kong, China. IEEE; 2014. p. 45–52.

[30] Mueller MW, D'Andrea R. Relaxed hover solutions for multicopters: application to algorithmic redundancy and novel vehicles. Int J Rob Res. 2016;35(8):873–889.

[31] Wang T, Xie W, Zhang Y. Sliding mode fault tolerant control dealing with modeling uncertainties and actuator faults. Transactions¡/DIFdel¿ISA Trans. 2012;51(3):386–392.

[32] Xu Q, Yang H, Jiang B. Adaptive fault-tolerant control design for UAVs formation flight under actuator faults. 2013 International Conference on Unmanned Aircraft Systems (ICUAS); Atlanta, GA, USA. IEEE; 2013. p. 1097–1105.

[33] Jin X, Yang G, Peng L. Robust adaptive tracking control of distributed delay systems with actuator and communication failures. Asian J Control. 2012;14(5):1282–1298.

[34] Casavola A, Garone E. Fault-tolerant adaptive control allocation schemes for overactuated systems. Int J Robust Nonlinear Control. 2010;20(17):1958–1980.

[35] Oppenheimer MW, Doman DB, Bolender MA. Control allocation for over-actuated systems. 2006 14th Mediterranean Conference on Control and Automation; Ancona, Italy. IEEE; 2006. p. 1–6.

[36] Drozeski GR, Saha B, Vachtsevanos GJ. A fault detection and reconfigurable control architecture for unmanned aerial vehicles. 2005 IEEE Aerospace Conference; Big Sky, MT, USA. IEEE; 2005. p. 1–9.

[37] Harkegard O, Glad ST. A backstepping design for flight path angle control. Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No. 00CH37187), Vol. 4. Sydney, NSW, Australia. IEEE; 2000. p. 3570–3575.

[38] Khan R, Williams P, Hill R. Fault tolerant flight control system design for UAV's using nonlinear model predictive control. 2011 Australian Control Conference; Melbourne, VIC, Australia. IEEE; 2011. p. 297–302.

[39] Lemos JM, Sampaio I, Rijo M. Actuator fault tolerant lqg control of a water delivery canal. 2013 Conference on Control and Fault-Tolerant Systems (SysTol); Nice, France. IEEE; 2013. p. 432–437.

[40] Mendonça LF, Vieira SM, Sousa JM. Fault accommodation using fuzzy predictive control. 2006 IEEE International Conference on Fuzzy Systems; Vancouver, BC, Canada. IEEE; 2006. p. 1535–1542.

[41] Osmić N, Kurić M, Petrović I. Detailed octorotor modeling and PD control. 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC); Budapest, Hungary. IEEE; 2016. p. 002182.

[42] Kuric M, Osmic N, Tahirovic A. Multirotor aerial vehicle modeling in modelica. Proceedings of the 12th International Modelica Conference, Prague, Czech Republic, May 15–17, 2017, no. 132. Linköping University Electronic Press; 2017. p. 373–380.

[43] Mahony R, Kumar V, Corke P. Multirotor aerial vehicles: modeling, estimation, and control of quadrotor. IEEE Rob Autom Mag. 2012;19(3):20–32.

[44] Morin D. Introduction to classical mechanics: with problems and solutions. New York: Cambridge University Press; 2008.

[45] Kuric M, Lacevic B, Osmic N. RLS-based fault-tolerant tracking control of multirotor aerial vehicles. 2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM); Munich, Germany. IEEE; 2017. p. 1148–1153.

[46] Velagić J, Osmić N, Puščul B. Identification, model validation and control of an octorotor unmanned aerial vehicle. 2018 IEEE 16th International Conference on Industrial Informatics (INDIN); Porto, Portugal. IEEE; 2018. p. 381–387 .

[47] Bangura M, Mahony R. Real-time model predictive control for quadrotors. IFAC Proc Volumes. 2014;47(3):11773–11780.

[48] Basson L. Control allocation as part of a fault-tolerant control architecture for UAVs [Ph.D. dissertation] Stellenbosch: University of Stellenbosch, 2011.

[49] Saied M, Shraim H, Francis C. Controllability analysis and motors failures symmetry in a coaxial octorotor. 2015 Third International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE); Beirut, Lebanon. IEEE; 2015. p. 245–250.

[50] Isermann R, Münchhof M. Identification of dynamic systems: an introduction with applications. Berlin: Springer Science & Business Media; 2010.

[51] Mellinger D, Kumar V. Minimum snap trajectory generation and control for quadrotors. 2011 IEEE International Conference on Robotics and Automation; Shanghai, China. IEEE; 2011. p. 2520–2525.

[52] Osmic N, Tahirbegovic A, Tahirovic A. Failure mode and effects analysis for large scale multirotor unmanned aerial vehicle controlled by moving mass system. 2018 IEEE International Systems Engineering Symposium (ISSE); Rome, Italy. IEEE; 2018. p. 1–8.