



HAL
open science

Fault diagnosis and fault-tolerant control strategy for rotor failure in an octorotor

Majd Saied, Benjamin Lussier, Isabelle Fantoni, Clovis Francis, Hassan Shraim, Guillaume Sanahuja

► **To cite this version:**

Majd Saied, Benjamin Lussier, Isabelle Fantoni, Clovis Francis, Hassan Shraim, et al.. Fault diagnosis and fault-tolerant control strategy for rotor failure in an octorotor. IEEE International Conference on Robotics and Automation (ICRA 2015), May 2015, Seattle, Washington, United States. pp.5266 - 5271, 10.1109/ICRA.2015.7139933 . hal-01180492

HAL Id: hal-01180492

<https://hal.science/hal-01180492>

Submitted on 27 Jul 2015

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

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

Fault Diagnosis and Fault-Tolerant Control Strategy for Rotor Failure in an Octorotor

Majd Saied^{1,2}, Benjamin Lussier¹, Isabelle Fantoni¹, Clovis Francis², Hassan Shraim² and Guillaume Sanahuja¹

Abstract—This paper presents a fault tolerant approach for a coaxial octorotor regarding rotor failures. A complete architecture including error detection, fault isolation and system recovery is presented. The diagnosis system is designed with a nonlinear observer to generate residuals and an inference model to evaluate them and isolate the faulty motor. Once the motor failure is diagnosed, a recovery algorithm is applied. It uses the built-in hardware redundancy of the octorotor and compensates the loss of the failing motor by controlling its dual to keep a stable flight that allows the multicopter to continue its mission. This architecture is validated on real flights.

I. INTRODUCTION

Multicopters unmanned aerial vehicles (UAV) are expected to provide services in the near future in a wide range of industrial applications such as: railway monitoring, supervision of electrical power lines, development of the precision agriculture, package delivery, etc. They have several advantages over comparably-scaled helicopters and planes because of their take off, landing and hovering capabilities, their mechanical simplicity and their inherent robustness. However, many constraints slow down their integration into civil airspace, particularly a poor reliability record, and the absence of regulations and standards regarding their own usage.

In order to solve the first restriction, fault tolerant control (FTC) strategies applied to UAV multicopters have received considerable attention in the last years. For quadrotors, partial failure was investigated in [1] - [5], where various approaches for fault recovery have been proposed: model reference adaptive control, flatness-based adaptive control, etc. Robust controllers were also developed to stabilize the quadrotor in a degraded mode: sliding mode, backstepping approach, optimal LQR... A total failure was studied in [6] and [7] where the strategy is to sacrifice the yaw control so that the vehicle rotates freely around an axis, and then tilt this axis for translational control. For multicopters with more than four rotors (hexarotor, octorotor), the objective of the FTC strategy is to maintain normal and stable flight and full control of the system despite the complete failure of one or

more propellers [8] - [10]. The method used in these three papers is the control allocation.

Most of these papers consider only one aspect of the FTC, which is the recovery part. However, another critical part of fault tolerant mechanisms is error detection and diagnosis. Indeed, accurate information on the fault occurrence time and location, as well as its severity, are necessary for the recovery phase. Several algorithms were applied to quadrotors to estimate the faults using nonlinear observers, Kalman filter or polynomial observers in [11] - [14]. But to the best of our knowledge, no diagnosis strategy has been developed for a multicopter with more than four rotors.

In this paper, we propose a complete architecture for error detection and system recovery of an octorotor, using a coaxial arrangement of rotors. The detection algorithm is based on a nonlinear sliding mode observer using as entries the angular positions, and the recovery algorithm consists of synchronizing the uncontrollable failed rotor and its dual one, and reallocating the control effort on the six other motors.

This paper is organized as follows: Section II presents some definitions and concepts about Fault Tolerance (FT). Section III presents the dynamics of the octorotor and the controller used to stabilize the body angles and the altitude. Section IV is dedicated to the detection and recovery algorithms. The results are then validated in real experiments in section V, and the paper concludes with perspectives in section VI.

II. FAULT TOLERANT CONTROL PROBLEM : DEFINITIONS

This section presents some basic concepts and definitions regarding computing systems's dependability. More information can be found in [15]. Dependability is the ability to justifiably trust in the deliverance of a system's correct service. It has means (like fault tolerance), attributes (like safety and reliability) and threats (faults, errors and failures). A failure is an event that occurs when the behavior of the system deviates from the correct expected service. The previous incorrect state of the system is called an error, and the cause of this error is a fault. On a multicopter, faults affecting the actuators have been identified as the highest priority due to their impact on the system stability and their significant probability of occurrence [10].

FT is one of four means to attain dependability with fault elimination, fault prevention and fault prediction. Its purpose is to allow the system to operate properly despite the

¹Sorbonne Universités, Université de Technologie de Compiègne, CNRS, UMR 7253 Heudiasyc, 60200 Compiègne, France, Email: {majd.saied, benjamin.lussier, isabelle.fantoni, guillaume.sanahuja}@hds.utc.fr

²Université Libanaise, Faculté de Génie, Centre de Recherche Scientifique en Ingénierie (CRSI), Liban, Email: {cfrancis, hassan.shraim}@ul.edu.lb

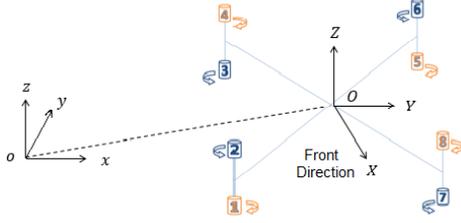


Fig. 1: The coaxial octotoror and the reference frames

presence of faults. The implementation of FT requires redundancy in the system: hardware and/or software redundancy. Hardware redundancy consists of using additional hardware to compensate failures. It is a fundamental technique to provide fault tolerance in safety critical systems like planes, trains or nuclear plants. The octotoror is an example of vehicles with built-in hardware redundancy and the proposed fault tolerant control strategy is based on this characteristic.

III. OCTOROTOR DYNAMICS AND CONTROL STRATEGY

In this section, we present the coaxial octotoror's dynamics and the control strategy that we use to stabilize the pose of the multirotor.

A. Dynamical Modeling of the Coaxial Octotoror

The configuration of the octotoror is represented in Fig. 1. It is similar to a quadrotor with two coaxial counter-rotating motors at the ends of each arm. This configuration has several advantages compared to the star configuration used in the literature [8], [9] in terms of stability and size. A classical star octotoror needs more arms, and each arm needs to be longer to guarantee adequate spacing among the rotors. We have adopted this configuration for its higher thrust to weight ratio.

Consider first a body-fixed reference frame R_B with the X, Y, Z axis originating at the center of mass of the vehicle. The X axis points to the front direction, the Y axis to the left direction and the Z axis upwards. Consider second an inertial frame R_I fixed to the earth $\{o, x, y, z\}$.

The equations governing the motion of the system are obtained using the Euler-Lagrange approach and give the commonly used model [16]:

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

We finally have the following expressions for the control inputs of the system and the disturbance Ω :

$$\begin{aligned} u_f &= F_{12} + F_{34} + F_{56} + F_{78} \\ \tau_\phi &= (F_{78} + F_{56} - F_{34} - F_{12}) * l * \frac{\sqrt{2}}{2} \\ \tau_\theta &= (F_{34} + F_{56} - F_{78} - F_{12}) * l * \frac{\sqrt{2}}{2} \\ \tau_\psi &= (\tau_2 + \tau_3 + \tau_6 + \tau_7) - (\tau_1 + \tau_4 + \tau_5 + \tau_8) \\ \Omega &= \omega_2 + \omega_3 + \omega_6 + \omega_7 - \omega_1 - \omega_4 - \omega_5 - \omega_8 \end{aligned} \quad (2)$$

In these equations, x , y and z denote the position of the vehicle and ϕ , θ , ψ the Euler angles with respect to the earth frame. p , q and r denote the angular velocities of the vehicle with respect to the body frame, I_{xx} , I_{yy} , I_{zz} and J_r are respectively the moments of inertia of the octotoror on the three axes with respect to R_B and the rotor inertia, l is the length of the arm, and ω_i is the speed of the rotor i . The thrust produced by each pair of coaxial rotors i and j is given by [17]:

$$\begin{aligned} F_{ij} &= \alpha_{ij} * (F_i + F_j) * (1 + \frac{S_s}{S_{prop}}) \\ F_i &= K_f * \omega_i^2 \end{aligned} \quad (3)$$

α_{ij} is the coefficient of loss of aerodynamic efficiency due to the aerodynamic interference between the upper and lower rotors of each pair of coaxial rotors. $S = (1 + \frac{S_s}{S_{prop}})$ represents the shape factor of the propellers, with S_s denoting the propeller's surface and S_{prop} the surface of the circle that the propeller would make when rotating. K_f is the thrust factor. The torque produced by each rotor is expressed by:

$$\tau_i = K_t * \omega_i^2 \quad (4)$$

with K_t is the reaction torque coefficient.

B. Control Strategy

A control strategy for stabilizing the octotoror while hovering is presented in this section. The altitude and the yaw positions are controlled by a PID controller that makes use of information obtained respectively from an ultrasonic sensor and an Inertial Measurement Unit (IMU):

$$u = K_p e + K_d \dot{e} + K_I \int_0^t e(\tau) d\tau \quad (5)$$

K_p , K_d and K_I are the controller's gains, and e is the state error compared to the desired position.

The roll and pitch angles are controlled using saturation functions, where each state is bounded separately. The stability of this control law is proved in [20].

$$\begin{aligned} \tau_\phi &= \frac{I_{xx}}{g} [\sigma_{py}(k_{py}(y - y_d)) + \sigma_{dy}(k_{dy}\dot{y}) - \\ &\quad \sigma_{p\phi}(k_{p\phi}\phi) - \sigma_{d\phi}(k_{d\phi}\dot{\phi})] \\ \tau_\theta &= -\frac{I_{yy}}{g} [\sigma_{px}(k_{px}(x - x_d)) + \sigma_{dx}(k_{dx}\dot{x}) - \\ &\quad \sigma_{p\theta}(k_{p\theta}\theta) - \sigma_{d\theta}(k_{d\theta}\dot{\theta})] \end{aligned} \quad (6)$$

where k_{py} , k_{dy} , $k_{p\phi}$, $k_{d\phi}$, k_{px} , k_{dx} , $k_{p\theta}$ and $k_{d\theta}$ are positive gains, and σ_{py} , σ_{dy} , $\sigma_{p\phi}$, $\sigma_{d\phi}$, σ_{px} , σ_{dx} , $\sigma_{p\theta}$, and $\sigma_{d\theta}$ are saturation functions defined as follows:

$$\begin{cases} \sigma_{b_i}(s) = b_i & \text{if } s > b_i \\ \sigma_{b_i}(s) = s & \text{if } -b_i < s < b_i \\ \sigma_{b_i}(s) = -b_i & \text{if } s < -b_i \end{cases} \quad (7)$$

IV. FAULT TOLERANT ARCHITECTURE

As mentioned before, fault tolerance is carried out via error detection and system recovery. These two components are detailed in this section.

A. Error Detection

The goal of the error detection component is to first detect the presence of an error in the system by calculating residuals that compare the predicted behavior of the system with the observed one. Second, it diagnoses the problem by applying classification or inference methods. We present in this section the residual generation method and then the residual evaluation that allows to detect and diagnose an error in the system.

1) *Residual generation*: To generate residuals, a nonlinear sliding mode observer is used. We define $x^T = [\phi \dot{\phi} \theta \dot{\theta} \psi \dot{\psi}]^T$ as the state vector, and $y = [\phi \theta \psi]^T$ as the output vector given by the IMU. Note that with these three outputs the system is observable. The angular velocities are also measured by the IMU but they are only used in the residuals. The observer is derived from (1) and has the form:

$$\begin{cases} \hat{x}_1(t) = \hat{x}_2 + \mu_1 \\ \hat{x}_2(t) = \frac{I_{yy} - I_{zz}}{I_{xx}} \hat{x}_4 \hat{x}_6 - \frac{J_r}{I_{xx}} \hat{x}_4 \Omega + \frac{1}{I_{xx}} \tau_\phi + \mu_2 \\ \hat{x}_3(t) = \hat{x}_4 + \mu_3 \\ \hat{x}_4(t) = \frac{I_{zz} - I_{xx}}{I_{yy}} \hat{x}_2 \hat{x}_6 + \frac{J_r}{I_{yy}} \hat{x}_2 \Omega + \frac{1}{I_{yy}} \tau_\theta + \mu_4 \\ \hat{x}_5(t) = \hat{x}_6 + \mu_5 \\ \hat{x}_6(t) = \frac{I_{xx} - I_{yy}}{I_{zz}} \hat{x}_2 \hat{x}_4 + \frac{1}{I_{zz}} \tau_\psi + \mu_6 \end{cases} \quad (8)$$

where:

$$\begin{cases} \mu_i = K_i \text{sign}(x_i - \hat{x}_i) \\ \mu_{i+1} = K_{i+1} \text{sign}(x_i - \hat{x}_i) \end{cases} \text{ for } i = 1, 3, 5 \quad (9)$$

K_i are positive gains.

The observer error dynamics are given by $\dot{\hat{x}}_i = x_i - \hat{x}_i$. According to [18], the convergence of this observer is conducted in two steps. The first step is the convergence of the measured variables, and the second step is the convergence of the errors $\hat{x}_{i=2,4,6}$. No injection is made for the channels ($i = 2, 4, 6$) before reaching the sliding manifolds $\text{sign}(\mu_i) = 0_{i=1,3,5}$.

The residuals are built as the differences between the system outputs and the observer outputs.

$$\begin{aligned} r_1(t) &= \phi(t) - \hat{\phi}(t) & r_2(t) &= \dot{\phi}(t) - \hat{\dot{\phi}}(t) \\ r_3(t) &= \theta(t) - \hat{\theta}(t) & r_4(t) &= \dot{\theta}(t) - \hat{\dot{\theta}}(t) \\ r_5(t) &= \psi(t) - \hat{\psi}(t) & r_6(t) &= \dot{\psi}(t) - \hat{\dot{\psi}}(t) \end{aligned} \quad (10)$$

They satisfy the following properties [19] :

- Invariance Relation: When no fault occurs, the mean of the residual is zero.
- Fault Detectability: When a fault occurs, the mean of the residual deviates from zero.

2) *Residual evaluation*: Once the residuals are generated, they are examined for the likelihood of faults by using fixed thresholds. A diagnostic inference model is illustrated in Table I. Given the previous residuals and the angular velocities from the IMU, it diagnoses the failing rotor based on a model analysis of the eight failure modes. The inferences are constructed using the residuals mentioned above to detect the occurrence of a fault in the system, and the information about the orientation of the octorotor are used in case of an error to

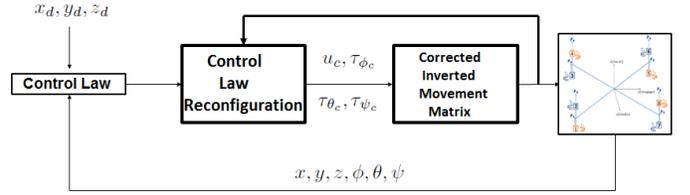


Fig. 2: Recovery Strategy for FT octorotor

isolate the failed actuator (see e.g. (11)). Note that a 1 in the table indicates that the residual exceeds the threshold. Since the body fixed frame is not within the arms of the octorotor, all residuals will be affected in case of the failure of any of the eight motors. This explains why all residuals exceed the thresholds when one actuator fails.

$$\begin{aligned} &\text{if } (r_i > \text{threshold}_i) \\ &\{\text{if } (\dot{\phi} < 0 \ \& \ \dot{\theta} < 0 \ \& \ \dot{\psi} < 0)\} \\ &\text{then motor 6 in failure} \end{aligned} \quad (11)$$

The algorithm in (11) is used to detect the failure of one motor. The fault diagnosis is more complicated for an octorotor than a quadrotor, because the number of faults to be estimated is higher. This implies that more outputs are needed to observe the system and to diagnose the faults.

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
r_1	1	1	1	1	1	1	1	1
r_2	1	1	1	1	1	1	1	1
r_3	1	1	1	1	1	1	1	1
r_4	1	1	1	1	1	1	1	1
r_5	1	1	1	1	1	1	1	1
r_6	1	1	1	1	1	1	1	1
$\dot{\phi}$	+	+	+	+	-	-	-	-
$\dot{\theta}$	+	+	-	-	-	-	+	+
$\dot{\psi}$	+	-	-	+	+	-	-	+

TABLE I: Diagnosis inference model for one rotor failure

B. System Recovery

The octorotor has a built-in hardware redundancy. Apart from its bigger payload, a motivation for using this type of multicopters instead of quadrotors is that the vehicle is able to maintain normal flight and full controllability in case of one motor failure (full stop or just loss of efficiency).

If a motor fails, a counteraction on its dual motor is sufficient to compensate this loss of thrust and to maintain a stable and safe flight. This can be shown in the expressions of τ_ϕ, τ_θ and τ_ψ in (2). The adopted recovery strategy is presented in Fig. 2. It is done in two steps. Once an error has been detected, the first step is to reduce the speed of the dual motor to compensate the missing amount of torque. The duality between the motors is based on the geometrical construction of the UAV and presented in Table II.

The second step is to increase the other motors speeds to generate sufficient thrust to maintain the UAV at the desired altitude. This is done in the inverted movement matrix where the motors thrust are computed from the four control efforts. In normal flight, the general form of the equation of the

speed of any motor i , $\omega_i = f(u_f, \tau_\phi, \tau_\theta, \tau_\psi)$, is given by:

$$\omega_i = \sqrt{\frac{1}{8} * \left(\frac{u_f}{\alpha_{ij}SK_f} \pm \frac{\tau_\phi}{\alpha_{ij}SK_{fd}} \pm \frac{\tau_\theta}{\alpha_{ij}SK_{fd}} \pm \frac{\tau_\psi}{K_t} \right)} \quad (12)$$

The positive sign in front of u_f indicates that all motors give positive thrust. The signs in front of the moments are selected depending on whether the force created by the motor generates a positive or negative moment. After an error detection, a correction factor is added to ensure that the same total thrust and moments as in normal flight are generated, resulting in equation (13):

$$\omega_i = p_i \sqrt{\left(\frac{1}{8} + c_i \right) \left(\frac{u_f}{\alpha_{ij}SK_f} \pm \frac{\tau_\phi}{\alpha_{ij}SK_{fd}} \pm \frac{\tau_\theta}{\alpha_{ij}SK_{fd}} \pm \frac{\tau_\psi}{K_t} \right)} \quad (13)$$

where p_i means the percentage of motor failure, and c_i is the correction factor depending on the number and percentage of motor failures. This method doesn't need the resolution of any optimization problem. This leads to a lower computational effort on the main controller unit and it is also much easier to validate (and ultimately to trust) than more complex control methods like control allocation. It is independent from the constraints on the inputs (motors' speeds) and it restores the system even when moving or following a given trajectory as will be demonstrated later in section V.

Motor	Dual Motor
1	6
2	5
3	8
4	7

TABLE II: Duality between motors

V. VALIDATION

An indoor test environment has been developed to validate the proposed fault tolerant architecture on a real octorotor. Fault injection is used to simulate motor failure.

A. Experimental Platform

The experimental UAV is shown in Fig. 3. It is a coaxial octorotor built at the Heudiasyc laboratory. It uses *BL2827* – 35 brushless motors driven with *BLCTRLV2* controllers (*Mikrokopter*) giving motors speeds measurements.

The multirotor is equipped with a *Microstrain 3DMGX3-25 IMU* composed of accelerometer, gyroscope, and magnetometer sensors giving Euler angles and rotation speed measurements at 100 Hz, and an ultrasonic sensor *SRF08* giving altitude measurements. The control law is executed in real time onboard the vehicle. The UAV's program is connected to a ground station where the parameters (control laws, filters...) are tuned during the system's development.

The octorotor's inertia was extracted from the software *Catia* and was found to be as follows: $I_{xx} = I_{yy} = 4.2 \times 10^{-2} \text{ Kg.m}^2$, $I_{zz} = 7.5 \times 10^{-2} \text{ Kg.m}^2$. The propeller's inertia was neglected. The vehicle's mass was measured to be 1.6 kg, and the distance from the center of mass to the center of the propellers is $l = 0.23 \text{ m}$.

The propellers were characterized using a force/torque sensor. The thrust and reaction torque coefficients were estimated as $K_f = 3 \times 10^{-5} \text{ N s}^2 / \text{rad}^2$ and $K_t = 7 \times 10^{-7} \text{ Nm/rad}^2$. They are able to produce thrust forces up to 6.5 N.



Fig. 3: Experimental Octorotor

B. Activity and Fault Injection

Two scenarios are considered. First, the octorotor hovers in the room. Second, the octorotor follows a circular trajectory. A motion capture system is used in this second scenario to obtain the position of the octorotor¹.

To illustrate a total failure in the propeller system, a motor is turned off by setting its power to zero from the ground station or by remote control.

C. Results on the Hovering Flight

The octorotor is first brought to a stable hovering position using the altitude and attitude controllers. The position is not controlled, the measurements available being only the Euler angles and the angular velocities. The motor 6 is turned off. The fault is detected online after 60 ms roughly, and the recovery algorithm is applied. Fig. 4 shows the motors speeds. It confirms that the fault is injected at time 20.27 s, and the detection is at time 20.86 s. The speeds of the motor 6 and its dual are shown to be null.

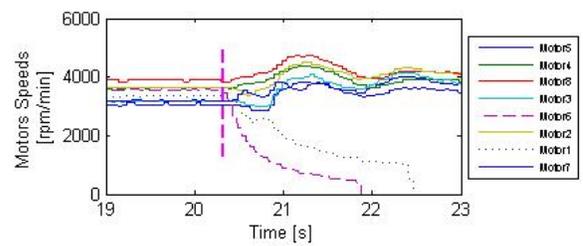


Fig. 4: Motors speeds during hovering flight when injecting total failure in rotor 6; The vertical dashed line indicates the injection fault time.

1) *Error detection*: When the actuator 6 stops, the dynamic of the real system behaves in a different way than the observer, and the residuals raise above the values they had in the faulty-free case. The parameters of the observers are given below : $K_1 = K_2 = 1$, $K_3 = K_4 = 0.7$, $K_5 = 1.7$, $K_6 = 1.4$.

The thresholds are determined from experiments and are selected in a way to avoid wrong decisions, such as false alarm or fault ignored.

¹The two experiments are shown in the video accompanying this paper.

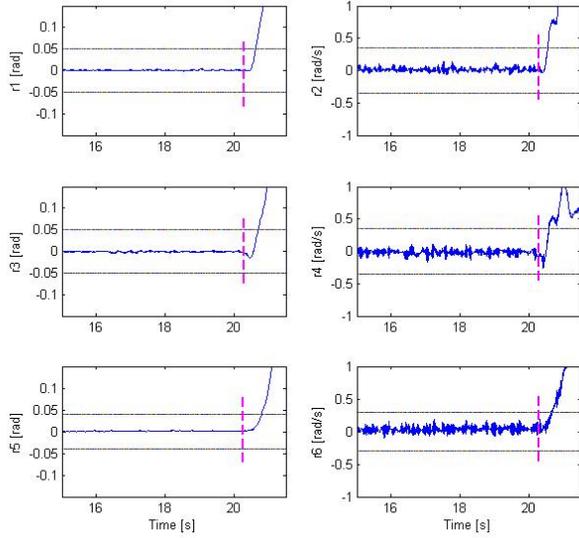


Fig. 5: Residuals on angles and angular velocities during hovering flight; The vertical dashed line indicates the injection fault time

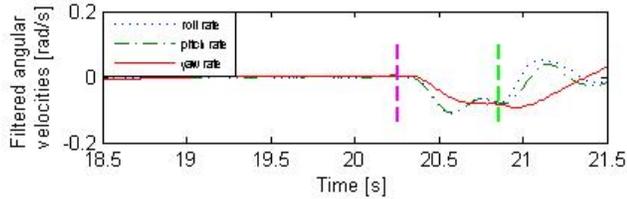


Fig. 6: Angular velocities during hovering flight [rad/s]; The first dashed line indicates the failure injection time and the second the detection and recovery time

Residuals are shown in Fig. 5, and the angular velocities in Fig. 6. A similar behavior can be seen if any of the actuators fails. As presented in section (IV-A.2), the diagnosis is based on the orientation of the octorotor after the failure occurs. A low pass filter is used to eliminate the noise on the angular velocities. As the filtered signals are used only to evaluate the signs of the speeds, the slight loss of information in filtering is inconsequential.

Fig. 6 shows that the three velocities are negative after the fault due to the displacement of the UAV in the direction of the faulty motor, which corresponds in this case to negative deviations around the three axis. According to Table I, the system thus correctly diagnoses a failure in motor 6.

2) *Recovery of the system:* As previously mentioned, the position is not controlled in this scenario. The recovery algorithm reacts 0.6 s after the fault injection. We present in Fig. 7 and Fig. 8 the Euler angles and the altitude respectively. The two vertical dashed lines mark the failure occurrence and its detection times respectively. It can be observed that the angles deviate quickly from their initial values after the failure has occurred, however the recovery algorithm is able to bring back the angles to stable values.

D. Results with a Circular Trajectory

In this scenario, the octorotor follows a circular trajectory. The UAV's position is measured using a motion capture system: An Optitrack system composed of 12 cameras. Due

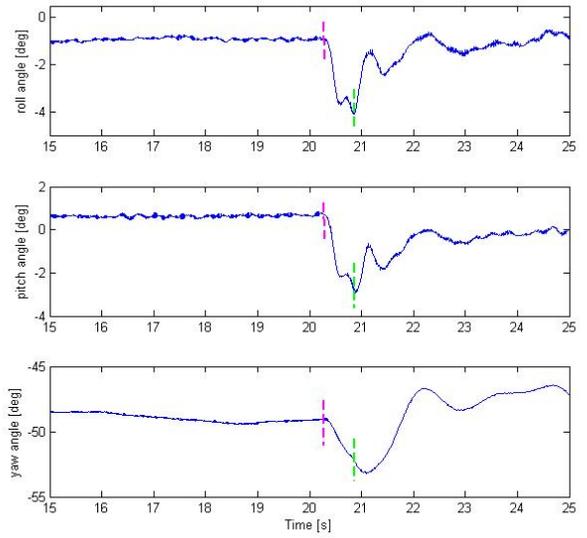


Fig. 7: Euler angles during hovering flight [deg]; The first dashed line indicates the failure injection time and the second the detection and recovery time.

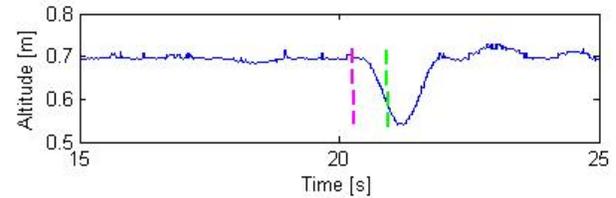


Fig. 8: Altitude during hovering flight [m]; The first dashed line indicates the failure injection time and the second the detection and recovery time

to the lack of time for experiments, only the recovery is tested in this case, assuming that the fault is detected after 1 s (a rather pessimistic value). Fig. 9 shows the UAV's trajectories with and without fault injection. In the latter case, the recovery algorithm was also able to quickly stabilize the pose of the system.

E. Discussion

The performed experiments show a good behavior of the detection and recovery algorithms after fault injection. The recovered behavior is very comparable to the nominal flight. However two points must be noted. First, in hovering mode, the octorotor will deviate slowly in position after the recovery since the position is not controlled in this case. Second, due to the reduction of the speed of the dual motor, a loss in the total thrust occurs. We solve this problem with a reconfiguration of the control law of the altitude by increasing the control gain. This is possible since the actuator limitations are not reached, but if the payload of the octorotor was greater, it may not have been able to stay exactly at the desired altitude.

The commonly used measurements for the diagnosis of a quadrotor are the positions and Euler angles [11] - [14]. In a closed space the positions can be given by a motion capture system, but a GPS is needed outdoor. The outputs of the GPS are biased in general and this may be a problem with future outdoors experiments. The proposed diagnosis strategy

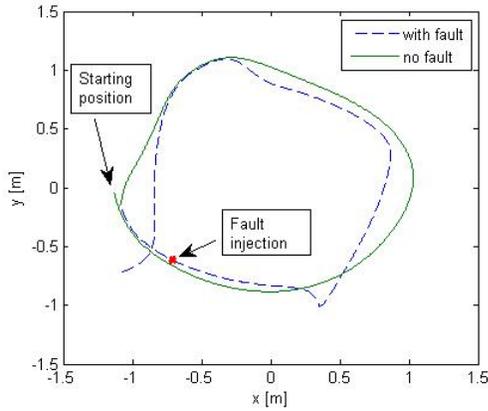


Fig. 9: Circle Trajectory of the octorotor with and without fault injection

based on the IMU measurements only, allows a detection in less than 1 s. The experiments show that the reconfiguration delay is a critical but somewhat lax parameter as recovery is still possible after a detection delay of 1.5 s. However, the lower the reconfiguration delay, the smaller the deviation from the stable position. This diagnosis algorithm is based on a sliding-mode observer and is thus robust to unknown parameter variations and noise. However significant disturbances and gusts of wind can still lead to false positives.

Compared to quadrotor case, the redundancy of the octorotor enables a full controllability of the UAV after the failure of one or more propellers. This fact may justify the additional costs and complexity in this architecture compared to the results of [7]. In this paper, we tolerated only one motor failure, but more failures can be detected and recovered from by applying new diagnoses and recovery parameters to the degraded mode.

Note that this architecture could be applied to any multirotor UAV with redundant actuators (hexarotor, star-shaped octorotor) by implementing minor modifications.

VI. CONCLUSION AND PERSPECTIVES

This paper presents an error detection and recovery architecture that allows an octorotor to maintain full controllability after losing one propeller. The proposed method takes full advantage of the octorotor configuration for FTC. It is based on analytical diagnosis and hardware redundancy. A sliding mode observer is used to estimate the angular speeds and to generate residuals that are used to detect a failing motor. The decision making algorithm is based on a set of if-else inference rules taking into account the signs of the angular velocities that are easy to trust and validate. To restore the system operation, the dual motor of the faulty one is invested.

In future works we intend to implement and compare a control allocation recovery method with the one presented in this paper. We also intend to tolerate successive motor failures in our UAV.

ACKNOWLEDGEMENTS

This work was carried out and funded in the framework of the Labex MS2T (Reference ANR-11-IDEX-0004-02) and the

ROBOTEX Equipment of Excellence (Reference ANR-10- EQPX-44). They were supported by the French Government, through the program Investments for the future managed by the National Agency for Research.

The authors would like to thank the Lebanese University research support program.

REFERENCES

- [1] A. Chamseddine et al., "Active fault-tolerant control system design with trajectory re-planning against actuator faults and saturation: Application to a quadrotor unmanned aerial vehicle," *Int. Journal of Adaptive Control and Signal Processing*, vol. 29, no. 1, pp. 1-23, 2015.
- [2] A. Chamseddine et al., "Flatness-based trajectory planning/replanning for a quadrotor unmanned aerial vehicle," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 4, pp. 2832-2848, 2012.
- [3] T. Li, Y.M. Zhang, and B.W. Gordon, "Nonlinear Fault-Tolerant Control of a Quadrotor UAV Based on Sliding Mode Control Technique," *IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes*, Mexico City, Mexico, August 29-31, 2012, pp. 1317-1322.
- [4] I. Sadeghzadeh et al., "Control Allocation and Re-allocation for a Modified Quadrotor Helicopter against Actuator Faults," *IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes*, Mexico City, Mexico, August 29-31, 2012, pp. 247-252.
- [5] M.H. Amoozgar, A. Chamseddine and Y. Zhang, "Fault-Tolerant Fuzzy Gain-Scheduled PID for a Quadrotor Helicopter Testbed in the Presence of Actuator Faults," *IFAC Conf. on Advances in PID Control*, Brescia, Italy, March 28-30, 2012, pp. 282-287.
- [6] A. Freddi, A. Lanzon, and S. Longhi, "A feedback linearization approach to fault tolerance in quadrotor vehicles," *IFAC World Congress*, Milano, Italy, August 28-September 2, 2011, pp. 5413-5418.
- [7] M.W. Mueller and R. D'Andrea, "Stability and control of a quadrotor despite the complete loss of one, two, or three propellers," *IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, China, May 31 - June 7, 2014, pp. 45-52.
- [8] H. Alwi and C. Edwards, "Fault Tolerant Control of an Octorotor Using LPV based Sliding Mode Control Allocation," *American Control Conference*, USA, Washington, DC, June 17-19, 2013, pp. 6505-6510.
- [9] A. Marks, J. F. Whidborne, and I. Yamamoto, "Control allocation for fault tolerant control of a VTOL octorotor," *UKACC International Conference on Control*, Cardiff, 3-5 September, 2012, pp. 357-362.
- [10] T. Schneider, "Fault-tolerant Multirotor Systems," *Master Thesis*, ETH Zurich, 2011.
- [11] M.H. Amoozgar, A. Chamseddine and Y.M. Zhang, "Experimental Test of a Two-Stage Kalman Filter for Actuator Fault Detection and Diagnosis of an Unmanned Quadrotor Helicopter," *J Intell Robot Systems*, vol. 70, issue 1-4, pp. 107-117, 2013.
- [12] C. Zhaohui et al., "Engineering Implementation on Fault Diagnosis for Quadrotors based on Nonlinear Observer," *Chinese Control and Decision Conference*, Guiyang, 25-27 May, 2013, pp. 2971-2975.
- [13] A.Freddi, S. Longhi and A. Monteri, "Actuator Fault Detection System for a Mini-Quadrotor," *International Symposium on Industrial Electronics (ISIE)*, Bari, 4-7 July, 2010, pp. 2055-2060.
- [14] H. Aguilar-Sierra et al., "Fault Estimation for a Quad-Rotor MAV Using a Polynomial Observer," *J Intell Robot Syst*, vol. 73, issue 1-4, pp. 455-468, 2014.
- [15] A. Avizienis et al., "Basic Concepts and Taxonomy of Dependable and Secure Computing," *IEEE Transactions on Dependable And Secure Computing*, vol. 1, no. 1, pp. 11-33, 2004.
- [16] S. Bouabdallah, "Design and Control of Quadrotors With Application to Autonomous Flying," *Ph.D thesis, Ecole Polytechnique Federale de Lausanne*, 2007.
- [17] A. Koehl, "Modélisation, Observation et Commande d'un drone miniature birotor coaxial," *Ph.D thesis, University of Nancy*, 2012.
- [18] T. Madani and A. Benallegue, "Sliding Mode Observer and Backstepping Control for a Quadrotor Unmanned Aerial Vehicle," *American Control Conference*, New York, 9-13 July, 2007, pp. 5887-5892.
- [19] I. Hwang et al., "A Survey of Fault Detection, Isolation, and Reconfiguration Methods," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 3, pp. 636-653, May 2010.
- [20] G. Sanahuja, P. Castillo and A. Sanchez, "Stabilization of n integrators in cascade with bounded input with experimental application to a VTOL laboratory system," *Int. Journal of Robust and Nonlinear Control*, vol. 20, no. 10, pp. 1129-1139, 2010.