Fault Tolerant Control for Multiple Successive Failures in an Octorotor: Architecture and Experiments
Majd Saied, Benjamin Lussier, Isabelle Fantoni, Clovis Francis, Hassan Shraim

To cite this version:

HAL Id: hal-01215427
https://hal.archives-ouvertes.fr/hal-01215427
Submitted on 14 Oct 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 Tolerant Control for Multiple Successive Failures in an Octorotor: Architecture and Experiments

Majd Saied1,2, Benjamin Lussier1, Isabelle Fantoni1, Clovis Francis2 and Hassan Shraim2

Abstract—This paper presents a fault tolerant control strategy based on an offline control mixing for an octorotor unmanned aerial vehicle (UAV) regarding several rotor failures. This strategy consists of a set of explicit laws, computed offline, each one dedicated to a fault situation. The corresponding law is selected according to the output of a fault detection and isolation (FDI) module. This module is designed with a non-linear sliding mode observer. The main advantage of this architecture is the deterministic character of the solution, its fast operation and the low computational load. The effectiveness of this approach is illustrated through real experimental application to a coaxial octorotor, where up to four motor failures are considered.

I. INTRODUCTION

With growing demands for safety and reliability, and an increasing awareness about the risks associated with system malfunction, dependability has become an essential concern in modern technological systems, particularly safety-critical systems such as aircraft. This has led to the design and development of fault-tolerant control systems (FTC).

The last decade witnessed many developments in the area of fault detection and diagnosis and fault tolerant control for Unmanned Aerial Rotary Wing Vehicles [1]. In particular, there has been extensive work on stability improvements for quadrotors in case of partial failures [2]-[4]. Recently, a few work addressed the problem of quadrotor’s complete propeller failure. In [5], the strategy proposed is to give up controlling the vehicle’s yaw angle, and use the remaining propellers to achieve horizontal spin. A backstepping approach was presented in [6] to control the quadrotor to follow a planned emergency landing. In [7], periodic solutions were exploited and experimentally validated to control the quadrotor in case of single, two opposing, or three propellers failures.

It is fairly straightforward to demonstrate, according to these works, that complete loss of a rotor for a quadrotor results in a vehicle that is not fully controllable. An obvious alternative is then to consider multirotors with redundant actuators (octoctors [8]-[10], hexarotors [11]-[13]). Inherent redundancy available in these vehicles is exploited, in the event of an actuator failure, to redistribute the control effort among the remaining working actuators such that stability and complete controllability are retained. This is known as the Control Allocation problem.

In the literature, several methods have been proposed [16]: direct control allocation, linear and quadratic programming, etc. The main technique inherent to these methods is to solve online a constrained optimization problem. The pseudo Inverse and Cascade Inverse methods are non-optimal techniques based on the pseudo inverse calculation of the control input matrix [8], [9]. Recently, it has been proposed to solve optimization problems offline by multi-parametric programming. The explicit solution is presented as piecewise affine functions, which can be evaluated online through a binary search tree. This approach was used in [11] and tested in real flights on an hexarotor. However, applying this method for an octorotor requires a lot of memory for storage, leads to a huge search tree and therefore increases the runtime of the control allocation.

In this paper, a complete error detection, fault diagnosis and system recovery architecture for a coaxial octorotor is presented. An offline rule-based control mixing and allocation is proposed for fault recovery when one or more rotors failure occur. This type of offline approach was first used by [14] and [15] to tolerate failures on the subscale suborbital demonstrator X-33 and a conventional aircraft respectively. Our approach is based on a set of rules computed offline and stored in a look-up table. A model-based error detection and fault diagnosis module is proposed to activate the appropriate recovery mixer when necessary. This module uses a non-linear sliding mode observer for residual generation and a diagnosis table for failure identification. Only few papers about fault tolerant control of overactuated multirotors consider the fault diagnosis in their studies, others assuming that the faults are perfectly identified. However, this is not a realistic hypothesis since fault diagnosis is a very challenging topic, especially for systems with redundant actuators. Moreover, to our knowledge, this paper describes the first real experiments of multiple failures in coaxial multirotors.

This paper is organized as follows: Section II presents the dynamic model of the coaxial octorotor. Section III is dedicated to the detection and recovery algorithms. The results are then validated with real experiments in section IV, and the paper concludes with perspectives in section V.

II. MODELING

The coaxial octorotor configuration (Fig. 1) developed in [17] is used as a test platform for the proposed fault tolerant control architecture. Its dynamic equations can be retrieved by means of the Euler-Lagrange formalism [18], and are given by:

\[ \dot{X}(t) = f(X(t)) + g(X(t))\tau(t) \] (1)
The thrust force and the torque produced by each propeller is proportional to the square of the angular velocity:

\[ F_i = K_f \omega_i^2 \]
\[ \tau_i = K_t \omega_i^2 \]

(7)

\( l \) is the arm length, \( K_f \) and \( K_t \) are the thrust and drag coefficients.

The thrust produced by each pair of coaxial rotors i and j is given by [20]:

\[ F_{ij} = \alpha_{ij} * (F_i + F_j) * (1 + \frac{S_s}{S_{prop}}) \]
\[ F_i = K_f \omega_i^2 \]

(8)

\( \alpha_{ij} \) 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.

III. FAULT TOLERANT CONTROL STRATEGY

The fault tolerant control architecture developed in this section is designed to accommodate multiple successive failures. It consists of a fault detection, fault identification and a fault recovery modules.

A. Fault Detection and Identification

The proposed diagnosis includes two separate fault detection and identification cascaded components. A second order sliding mode observer yielding insensitivity to unknown parameter variations and noise is used for residual generation. The residual evaluation is based on a mathematical analysis of the octorotor’s dynamics in the presence of faults.

1) Error Detection: A sliding mode observer based on the so-called Super-Twisting Second Order Sliding Mode Approach is used [19]. We define \( x_1 = [\phi \theta \psi]^T \) as the measured angular position vector and \( x_2 = [\dot{\phi} \dot{\theta} \dot{\psi}]^T \) as the angular velocity vector. The observer has the form:

\[ \dot{x}_1 = x_2 + z_1 \]
\[ \dot{x}_2 = f(t, x_1, \dot{x}_2, \tau) + z_2 \]

(9)

where \( \dot{x}_1 \) and \( \dot{x}_2 \) are the state estimations, \( z_1 \) and \( z_2 \) are the output injections

\[ z_1 = \lambda |x_1 - \hat{x}_1| \frac{1}{2} \text{sign}(x_1 - \hat{x}_1) \]
\[ z_2 = \alpha \text{sign}(x_1 - \hat{x}_1) \]

(10)

\( \lambda \) and \( \alpha \) are positive gains that satisfy the inequalities:

\[ \alpha > f^+ \]
\[ \lambda > \frac{f^+ + (\alpha + f^+)(1 + p)}{(1 - p)} \]

(11)

Assuming that the system is bounded, this ensures the existence of a positive matrix \( f^+ \) such as:

\[ |f(t, x_1, x_2, \tau) - f(t, x_1, \dot{x}_2, \tau) + \zeta(t, x_1, x_2, \tau)| < f^+ \]

(12)

and \( p \) is some chosen constant, such as \( 0 < p < 1 \). The finite time convergence analysis proof is given in [19]. The...
residuals are generated using the difference between the real and the estimated output of the system:

\[
\begin{align*}
 r_1 &= \dot{\phi} - \dot{\phi}; & r_4 &= \dot{\phi} - \dot{\phi} \\
 r_2 &= \theta - \dot{\theta}; & r_5 &= \dot{\theta} - \dot{\theta} \\
 r_3 &= \psi - \dot{\psi}; & r_6 &= \dot{\psi} - \dot{\psi}
\end{align*}
\]  
(13)

When the absolute value of any residual exceeds a specific detection threshold, an error is considered present in the system.

2) Fault Diagnosis: Our fault diagnosis procedure consists of a decision-making process based on threshold logic and decision functions. Theoretical fault signatures are deduced from faulty system behavior, and the fault isolation consists of matching the real fault signature with some of those theoretical ones.

The diagnostic factors in our study are the residuals derived from the angles and angular velocities. Table I is obtained from a mathematical study of the faulty system’s dynamics [17]. \(r_i\) indicates the residuals obtained from (13) and \(f_i\) indicates a failure on motor \(i\) using the signs of the predefined residuals.

<table>
<thead>
<tr>
<th>(f_1)</th>
<th>(f_2)</th>
<th>(f_3)</th>
<th>(f_4)</th>
<th>(f_5)</th>
<th>(f_6)</th>
<th>(f_7)</th>
<th>(f_8)</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

TABLE I: Diagnosis inference model based on the residuals signs

B. System Recovery

For the system recovery, an offline nonlinear constrained optimization approach is used, where the solutions to a predefined set of failures are calculated offline. During the UAV’s operation, the appropriate control mixing is selected from the output of the FDI module according to the fault diagnosis.

The octorotor is equipped with eight actuators (motors/propellers): four rotating clockwise, four rotating counter-clockwise. The controller generates a virtual control command \(v = [u_f \quad \tau_\phi \quad \tau_\theta \quad \tau_\psi]^T\). This virtual command is distributed among the set of \(n\) healthy actuators where \(n \leq 8\).

The problem is formulated as:

\[
Bu(t) = v(t)
\]

\[
u_{\text{min}} \leq u \leq u_{\text{max}}
\]

\[
u = [\omega_1^2 \omega_2^2 ... \omega_8^2]^T
\]

and \(B\) is the control effectiveness matrix deduced from (6) and defined as:

\[B = \begin{bmatrix} t_1 & ... & t_8 \\
 r_1 & ... & r_8 \\
 p_1 & ... & p_8 \\
 y_1 & ... & y_8 \end{bmatrix}
\]

with:

\[
t_i = \sigma_{ij}K_f.S
\]

\[
r_i = d_{ci}r_iK_f.SL\sqrt{2}/2
\]

\[
p_i = d_{pi}r_iK_f.SL\sqrt{2}/2
\]

\[
y_i = d_{yi}y_iK_f
\]

\(t_i, r_i, p_i\) and \(y_i\) are respectively the thrust and the torques around the three axes produced by the actuator \(i\), \(d_{ci}/y_i\) is either 1 or -1 depending on whether the force created by the motor generates a positive or negative moment.

For a stable hovering flight, it is necessary to ensure that the moments around the three axes are null:

\[
u_f = mg \quad \tau_\phi = 0
\]

\[
\tau_\theta = 0 \quad \tau_\psi = 0
\]

The problem is thus formulated as: Given \(v = [mg \quad 0 \quad 0 \quad 0]^T\), the objective of the control allocation is to find the vector \(u\) corresponding to the motors inputs. The system has 4 equations and \(K\) unknown variables: \(K = 8\) in fault-free nominal case and \(4 \leq K < 8\) in faulty mode. Multiple solutions can be found for each fault situation, however we choose the explicit and deterministic solution which minimizes the energy consumption during flight by solving the optimization problem below:

\[
\min_{u} \frac{1}{2}u^Tu
\]

such that:

\[
\begin{cases}
u_{\text{min}} \leq u \leq u_{\text{max}} \\
Bu = v
\end{cases}
\]

where \(u_{\text{max}}\) is determined according to each fault. This solution will not be the best one in every situation, but guarantees the best control allocation in hovering flight and displacements with small angles.

Different fault configurations will be detailed in the next subsections. Up to four motor failures will be considered. For an octorotor, there exist \(N_f = \sum_{i=1}^{4}C_i = 162\) fault cases if we consider that at least four motors are healthy. However, because of the symmetry in the octorotor’s configuration, only some cases are presented. The others are symmetrically equivalent to one of these.

1) Case 1: Fault-Free Mode: In normal flight, the general form of the equation of the motor’s \(i\) speed, \(\omega_i = f(u_f, \tau_\phi, \tau_\theta, \tau_\psi)\), is given by (19). The positive sign in front of \(u_f\) indicates that all motors give positive thrust. The signs in front of the moments are selected from (6): depending on the motor’s position in the UAV (i.e. its number), the force created by the motor generates a positive or negative moment.

\[
\omega_i = \left[\frac{1}{8}\left(\frac{u_f}{\sigma_{ij}K_f.S} \pm \frac{\tau_\phi}{\sigma_{ij}K_f.SL} \pm \frac{\tau_\theta}{\sigma_{ij}K_f.SL} \pm \frac{\tau_\psi}{K_fL}\right)\right]
\]

with \(L = l \ast \sqrt{2}/2\)

2) Case 2: One Motor Failure: In case of a motor’s failure, a possible solution of (17) is to stop its dual motor to keep the three moments around the axes null (a control strategy that we used in our first experiments [17]). However, the optimization program (18) gives us another solution that can be applied without the loss of the dual motor.
Considering for example that motor 1 failed, the proposed allocation is to reduce the power of its dual to the half, and to increase the powers of the three upper remaining motors by a factor of 1.5.

\[
\begin{align*}
F'_0 &= 0.5F_0 \\
F'_i &= 1.5F_i & i &= 2, 4, 8 \\
F'_i &= F_i & i &= 3, 5, 7
\end{align*}
\] (20)

Table: Failed Motors and Reallocated inputs

<table>
<thead>
<tr>
<th>Failed Motors</th>
<th>Reallocated inputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 &amp; 2</td>
<td>( F'_i = 0 ) ( i = 5, 6 )</td>
</tr>
<tr>
<td></td>
<td>( F'_i = 2F_i ) ( i = 3, 4, 7, 8 )</td>
</tr>
<tr>
<td>1 &amp; 3</td>
<td>( F'_i = 2F_i ) ( i = 2, 4 )</td>
</tr>
<tr>
<td></td>
<td>( F'_i = F_i ) ( i = 5, 6, 7, 8 )</td>
</tr>
<tr>
<td>1 &amp; 4</td>
<td>( F'_i = 2F_i ) ( i = 2, 3, 5, 8 )</td>
</tr>
<tr>
<td></td>
<td>( F'_i = F_i ) ( i = 2, 3, 5, 7 )</td>
</tr>
<tr>
<td>1 &amp; 5</td>
<td>( F'_i = 2F_i ) ( i = 3, 4, 6, 7 )</td>
</tr>
<tr>
<td></td>
<td>( F'_i = F_i ) ( i = 3, 4, 5, 6 )</td>
</tr>
<tr>
<td>1 &amp; 6</td>
<td>( F'_i = 2F_i ) ( i = 2, 3, 4, 5, 7 )</td>
</tr>
<tr>
<td>1 &amp; 7</td>
<td>( F'_i = 2F_i ) ( i = 2, 3, 5, 8 )</td>
</tr>
<tr>
<td>1 &amp; 8</td>
<td>( F'_i = 2F_i ) ( i = 2, 4, 5, 7 )</td>
</tr>
<tr>
<td></td>
<td>( F'_i = F_i ) ( i = 3, 6 )</td>
</tr>
</tbody>
</table>

Table: Failed Motors and Reallocated inputs

<table>
<thead>
<tr>
<th>Failed Motors</th>
<th>Reallocated inputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 &amp; 2 &amp; 3 (or 4 or 7 or 8)</td>
<td>( F'_i = 0 ) ( i = 5, 6 )</td>
</tr>
<tr>
<td>1 &amp; 2 &amp; 3 (or 5 or 6)</td>
<td>( F'_i = 2F_i ) ( i = 3, 4, 7, 8 )</td>
</tr>
<tr>
<td>1 &amp; 3 &amp; 4</td>
<td>( F'_i = 0 ) ( i = 6 )</td>
</tr>
<tr>
<td>1 &amp; 3 &amp; 5</td>
<td>( F'_i = 1.5F_i ) ( i = 2, 6, 8 )</td>
</tr>
<tr>
<td>1 &amp; 3 &amp; 6 (or 7)</td>
<td>( F'_i = F_i ) ( i = 2, 4, 5, 7 )</td>
</tr>
<tr>
<td>1 &amp; 3 &amp; 7</td>
<td>( F'_i = 2F_i ) ( i = 2, 4, 5, 7 )</td>
</tr>
<tr>
<td>1 &amp; 4 &amp; 5</td>
<td>( F'_i = 0 ) ( i = 8 )</td>
</tr>
<tr>
<td>1 &amp; 4 &amp; 6 (or 7)</td>
<td>( F'_i = 2F_i ) ( i = 2, 3, 5, 8 )</td>
</tr>
<tr>
<td>1 &amp; 5 &amp; 6</td>
<td>( F'_i = 0 ) ( i = 7 )</td>
</tr>
<tr>
<td>1 &amp; 5 &amp; 7</td>
<td>( F'_i = 2F_i ) ( i = 2, 4, 8 )</td>
</tr>
<tr>
<td>1 &amp; 6 &amp; 7</td>
<td>( F'_i = 0 ) ( i = 4 )</td>
</tr>
<tr>
<td>1 &amp; 6 &amp; 8</td>
<td>( F'_i = F_i ) ( i = 2, 4, 5, 7 )</td>
</tr>
<tr>
<td>1 &amp; 7 &amp; 8</td>
<td>( F'_i = F_i ) ( i = 3 )</td>
</tr>
</tbody>
</table>

Table: Failed Motors and Reallocated inputs

<table>
<thead>
<tr>
<th>Failed Motors</th>
<th>Reallocated inputs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 &amp; 2 &amp; 3 &amp; 5 &amp; 6</td>
<td>( F'_i = 2F_i ) ( i = 3, 4, 7, 8 )</td>
</tr>
<tr>
<td>1 &amp; 3 &amp; 5 &amp; 7</td>
<td>( F'_i = 2F_i ) ( i = 2, 4, 6, 8 )</td>
</tr>
<tr>
<td>1 &amp; 4 &amp; 6 &amp; 7</td>
<td>( F'_i = 2F_i ) ( i = 2, 3, 5, 8 )</td>
</tr>
<tr>
<td>2 &amp; 3 &amp; 5 &amp; 8</td>
<td>( F'_i = 2F_i ) ( i = 1, 4, 6, 7 )</td>
</tr>
<tr>
<td>2 &amp; 4 &amp; 5 &amp; 7</td>
<td>( F'_i = 2F_i ) ( i = 1, 3, 6, 8 )</td>
</tr>
<tr>
<td>2 &amp; 4 &amp; 6 &amp; 8</td>
<td>( F'_i = 2F_i ) ( i = 1, 3, 5, 7 )</td>
</tr>
<tr>
<td>3 &amp; 4 &amp; 7 &amp; 8</td>
<td>( F'_i = 2F_i ) ( i = 1, 2, 3, 5, 6 )</td>
</tr>
</tbody>
</table>

Table: Failed Motors and Reallocated inputs

3) Case 3: Two Motors Failures: Due to the problem’s symmetry, we will consider only the cases where motors 1 and 8 fail, with \( i = 2, 3, ..., 8 \) respectively. The new inputs will be directly presented in Table II as in (20). The most critical cases are when two coaxial motors fail, or two motors rotating in the same direction fail (see cases 1&2, 1&4 and 1&6 in Table II). In these cases, based on this method, a synchronization between the failed motors and their duals is necessary to reach the equilibrium. Although a theoretical solution for (17) is found for these failures configurations, no controllability study was done before for this type of multirotor, and this solution has not been experimentally validated.

4) Case 4: Three Motors Failures: For the same reason as in case 3, only combinations of the failures of motor 1 with two other motors are considered in Table III. Note that the dashed line indicates that no solution was found with this combination, and another FT mechanism may be needed to avoid system’s failure (such as [7] for example).

5) Case 5: Four motors failures: In case of four motors failures, solutions for (17) were only found for the combinations presented in Table IV.

IV. VALIDATION

In this section, the complete fault tolerance strategy is tested in real experiments on an octorotor UAV. Fault injection is used to simulate motor failure, by sending commands to stop the failed motors at desired times.

A. Experimental platform

The experimental UAV is shown in Fig. 2. It is a coaxial octorotor built at the Heudiasyc laboratory. Its parameters are given in Table V:

| \( K_f \) | Thrust factor | \( 3 \times 10^{-5} \) Ns²/rad² |
| \( K_t \) | drag factor | \( 7 \times 10^{-2} \) Nm/rad² |
| \( m \) | mass of the vehicle | 1.6 kg |
| \( I_{xx}, I_{yy} \) | Inertia | \( 4.2 \times 10^{-2} \) Kg.m² |
| \( I_{zz} \) | Inertia | \( 7.5 \times 10^{-2} \) Kg.m² |

Table V: The model’s parameters

B. Results

The octorotor is brought to a hovering stable flight. The control loop for longitudinal and lateral positions of the UAV uses input from an Optitrack motion capture system to offer a
Fig. 3: Motors speeds after four motors failures [rpm]; Faults are injected respectively on motors 6, 2, 4 and 8 at times $t_6 = 33.73$ s, $t_2 = 41.05$ s, $t_4 = 49.2$ s and $t_8 = 57.6$ s. The dashed lines indicate the fault injections times.

Fig. 4: Residuals on Euler angles after motors failures. Faults are injected respectively on motors 6 and 2 at times $t_6 = 33.73$ s and $t_2 = 41.05$ s. The dashed lines indicate the fault injections times.

Fig. 5: Residuals on angular velocities after motors failures. Faults are injected respectively on motors 6 and 2 at times $t_6 = 33.73$ s and $t_2 = 41.05$ s. The dashed lines indicate the fault injections times.

Fig. 6: Altitude after four motors failure [m]; The octorotor takes off at time $t = 15$ s, then faults are injected respectively on motors 6, 2, 4 and 8 at times $t_6 = 33.73$ s, $t_2 = 41.05$ s, $t_4 = 49.2$ s and $t_8 = 57.6$ s. The octorotor lands at time $t = 68$ s. The dashed lines indicate the fault injections times.
A fault tolerant control strategy for a coaxial octocopter regarding one motor failure was presented in our previous work [17]. The recovery strategy consists of synchronizing the uncontrollable failed rotor and its dual one, and realocating the control effort on the six other motors. Since the dual motor is stopped, the system could not be recovered in all two motors failures configurations. With the strategy proposed in this paper, the solution is optimized such that there will be a problem only in the case of two coaxial motors failures. Moreover, more successive failures can be tolerated, and the total thrust in case of faults is optimized.

Compared to other works studying the fault tolerant control of quadrotors under several rotors failures [7], this strategy has a main advantage: the possibility of tolerating multiple failures without losing complete controllability. It thus allows the octocpter to keep a stable flight and to continue its mission. Moreover, we find this solution much easier to implement on large UAVs than that of [7], at the cost of having more actuators in the system.

V. CONCLUSIONS

This paper presents a complete fault tolerant control strategy that allows a coaxial octocopter to maintain full controllability after losing up to four motors. The FDI is based on a sliding mode observer associated with an inference model for fault isolation. The recovery is based on an offline computation of a set of control mixing laws matched to a number of predefined failure combinations. Experimental validations are presented with good results with two successive failures for diagnosis, and four successive failures for recovery.

In future works, we intend to validate the diagnosis for more than two motors failures and to consider the two coaxial motors failure case, by studying the controllability of the system in this situation and then applying an emergency solution. We also intend to consider transient faults and external perturbations (such as the wind).

ACKNOWLEDGEMENTS

This work was carried out and funded in the framework of the Labex MS2T (Reference ANR-11-IDEX-0004-02) and the ROBTEX 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 and to express their gratitude to Guillaume Sanahuja, engineer in Heudiasyc Laboratory, for his support in performing our real-time experiments.

REFERENCES