Hexacopter fault tolerant actuator allocation analysis for optimal thrust

Claudio D. Pose*, Juan I. Giribet*† and Alejandro S. Ghersin‡
* Grupo de Procesamiento de Señales, Identificación y Control (GPSIC)
Departamento de Ingeniería Electrónica, Universidad de Buenos Aires, Argentina
† Instituto Argentino de Matemática - Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Argentina.
‡ Departamento de Ingeniería Electrónica, Instituto Tecnológico de Buenos Aires (ITBA) and CONICET, Argentina
Email: {cldpose, jgiribet}@fi.uba.ar; aghersin@itba.edu.ar

***Abstract*—Recently, it was shown that an hexagon shaped hexarotor vehicle with tilted rotors, is capable of fault tolerant attitude and altitude control. In this work, we propose a strategy to select the signals commanded to each rotor in order to achieve a desired torque and vertical force. The proposed strategy is optimal in the sense that minimizes the maximum force exerted by the rotors.**

A comparison with the commonly used strategy based upon the Moore-Penrose pseudoinverse is carried out. It is shown that, with the optimal strategy proposed here, maneuverability is improved, because the new method takes into account the actuators constraints. Although the optimal strategy is computationally more demanding than the classical method, the additional computational burden is not significant when both strategies are compared in a real application. To show this, both algorithms were programmed in an autopilot based on an ARM Cortex M3 microcontroller, and the experimental results are presented.

***Index Terms*—Fault tolerant control, Unmanned aerial vehicle, Actuator Allocation.**

I. Introduction

Multi-rotor micro aerial vehicles (MAVs) have become very popular in recent years, due to the fact that the electronic systems needed to fly them have increased their availability, through an impressive reduction of cost, size and weight. Several applications have emerged where MAVs have proved extremelly useful. As this technology has become more popular, the number of accidents reported has been increasing. In this context, fault tolerance becomes a critical issue.

A recent survey on fault tolerant control for multi-rotor MAVs can be found in [1]. In [2], the study of fault tolerant controls has been carried out for multi-rotor vehicles with different number of rotors.

The minimum number of rotors needed to achieve a fault tolerant control for multi-rotor MAVs, is an issue that has been discussed in [3], [4], [5], [6], among others. In [5], a study on the trade off between number of rotors, maneuverability, efficiency and redundancy is carried out. As it is shown in these works, in the case of failure on a rotor, an hexagon shaped hexarotor without its motors tilted (see Fig. 1), will see its performance degraded, due to the fact that the attitude controller will be unable to reject disturbance torques in certain directions. This means that this kind of vehicle is not fault tolerant.

There are some known solutions to the full controllability problem for fault tolerant multirotors. The octocopter solution [7] requires more actuators, increasing the mechanical redundancy; other mechanical designs make use of servomotors in order to change the position and orientation of the motors [8]; bidirectional rotating motors are proposed as well [5], with the disadvantage of generating thrust in the opposite direction.

Most frequently, multirotor helicopters such as the classical quadrotor, the hexagon shaped hexarotor, or the octagon shaped octocopter, have the spinning direction of their motors set in an alternated fashion. Namely the adjacent motors have reverse spinning direction with respect to one another. A "*de facto*" notation is usually employed to describe this, where "**P**" denotes clockwise spinning direction of a motor and "**N**" denotes a counterclockwise one. With this notation in hand, the hexagon shaped hexacopter studied within this article, will have a NPNPNP setup for the spinning direction of its rotors, from number 1 to number 6 (see Fig.1). Another spinning direction setup for hexacopters proposed in the literature, is to reconfigure the spinning direction of the motors, for instance a PPNNPN configuration. This allows to maintain full controllability for total rotor failures, but only in the first four actuators of the PPNNPN setup ([3], [9], [10], [11]). In this work we assume that failures are identically likely to appear in any motor, then this configuration can not be considered fault tolerant.

To overcome the lack of fault tolerance of the standard hexagon shaped hexarotor, in [6], instead of the standard design with rotors pointing in the vertical direction, an alternative is proposed, which turns out to be completely controllable even in case of failure in one rotor. It is shown that by tilting the rotors with respect to the horizontal plane that contains the motors, towards the vehicle's vertical axis (see Fig. 2), fault tolerant control can be achieved without losing control neither in attitude nor in altitude, even with a faulty motor. An aspect that must be pointed out about this design, is that tilting the motors (the multi-rotor's arms) is an already established practice, as it allows for a more stable vehicle given it lowers the center of mass. With the addition of a retractable landing skid, this also provides a clearer line of sight for camera-type payloads placed under the vehicle.

There is plenty of research on actuators and sensors failure detection. A number of references can be cited, where both performance degradation and total failure are considered ([11],

[12], [13], [14], [15], [16]); in particular, for vehicles with eight rotors, an error detection and fault isolation technique based on non-linear observers and LPV control have been presented in [17] and [18], respectively. However, this research only focuses on the control strategy after a failure appears in one of the actuators. The failure is assumed to be a total shut-down of one of the motors, thus not generating any thrust or torque.

Here, the design approach proposed in [6] is adopted, but employing a different rule for actuator allocation. To reach a desired torque and vertical force, the method proposed in [6] relays on the calculation of the Moore-Penrose pseudoinverse of a given matrix, which is the most common method to compute the rotors' commands. However, the solution rendered by this method may not be feasible, because it does not take into account the constraints of the actuators, more specifically, the maximum and minimum forces that may be exerted by each rotor. This is the reason why a new strategy is proposed. Instead of minimizing the energy of the signals commanded to each rotor, something that is achieved employing the Moore-Penrose pseudoinverse, the control action exerted by the most stressed motor is taken into account. It is shown that with the method proposed here, it is possible to achieve torques and vertical forces in cases where the method based on the Moore-Penrose pseudoinverse gives unfeasible solutions. Similar methods have been proposed in the literature (see for instance [19] and references therein). But, for the design proposed in [6] it is possible to find the optimal solution with a simple algorithm that can be run on a microcontroller of moderate performance without floating point hardware capabilities.

In this work we present this algorithm and a comparison between it and the Moore-Penrose pseudoinverse performance through numerical flight simulation. We also show the comparison of the computational load when we use both strategies within a control algorithm running on a computer based on an ARM-Cortex M3 microcontroller.

II. Notation

Given a matrix $X \in \mathbb{R}^{n\times m}$, $N(X)$ denotes its kernel and $X^\dagger \in \mathbb{R}^{m\times n}$ its Moore-Penrose pseudoinverse. Given a matrix $X \in \mathbb{R}^{n\times m}$, we denote $\tilde{X}_i \in \mathbb{R}^{n\times m}$ as the matrix X, with the *i*-th column replaced by zeros. This matrix will be used when referring to a motor failure in the *i*-th motor. For a vector $x \in \mathbb{R}^n$, its *i*-th component is denoted by x_i. Its Euclidean norm is given by $\|x\|_2 = \sqrt{x^T x}$, and its infinity norm by

$$\|x\|_\infty = \max_{i=1,\dots,n} |x_i|.$$

A vector $x \in \mathbb{R}^n$ will be called non–negative (positive), and denoted $x \geq 0$ ($x > 0$) , if each of its components are non–negative (positive) numbers, i.e., $x_1 \geq 0, \dots, x_n \geq 0$ ($x_1 > 0, \dots, x_n > 0$).

The following short notation: $c\gamma = \cos\gamma$ and $s\gamma = \sin\gamma$, will be used in the sequel.

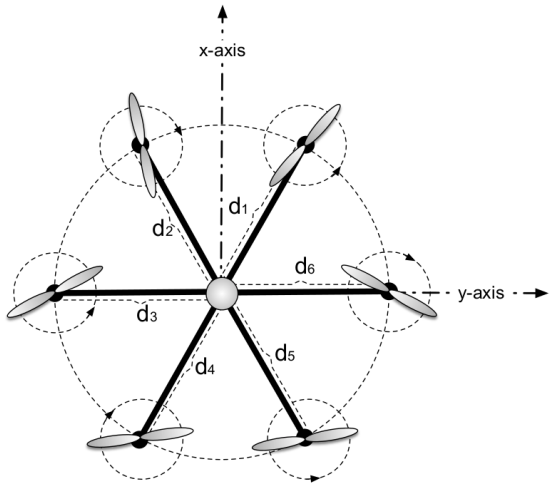

Fig. 1. Top view for a standard hexacopter motor distribution.

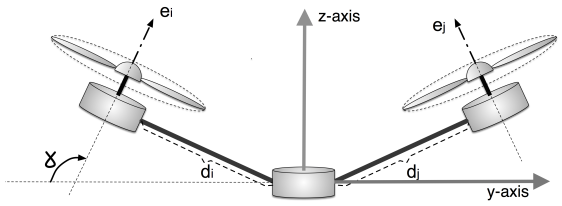

Fig. 2. Side view for a standard hexacopter motor distribution, with tilted motors

III. Vehicle model

In this section a model for the vehicle's actuator allocation problem, is presented.

Considering unidirectional spinning motors, it is assumed that each motor exerts a force $f_i \in [0, F_M]$. In practice, each motor is commanded through a Pulse Width Modulated (PWM) signal u_i, which goes from 0 to 100%. Near the nominal operating point, a linear relation between the PWM percentage and the exerted force is assumed, with $f_i = k_f u_i$. It is also considered that each motor exerts a torque on its spinning axis, $m_i = (-1)^i k_t u_i$. The k_f and k_t, constants are usually established experimentally. The constant $\tilde{k}_t := \frac{k_t}{k_f}$ is defined as it will used in what follows, for the sake of clarity.

As it was mentioned above, by tilting the rotors an angle γ (as seen in Fig. 2), fault tolerant control can be achieved without losing control neither in attitude nor in the vertical direction (thrust), even with a faulty motor. Here we assume that the hexarotor has this fault tolerant configuration (see Fig. 2).

We will define l as the length of the arm measured from the center of the vehicle to the motor axis along the arm, i.e. $l = \|d_i\|_2$, with $i = 1, ..., 6$. The angle γ will be the tilting angle of the motor, measured as depicted in Fig. 2. To simplify the notation, a parameter α is defined in the following way:

$$\alpha = \alpha(\gamma) = \frac{\tilde{k}_t}{\sqrt{3}\ l \tan(\gamma)} \tag{1}$$

In [6, Theorem 3], it was shown that this parameter should be chosen such that $0 < |\alpha| < 1$ so that fault tolerance can be achieved. In what follows, it is assumed that α satisfies this condition.

Let M_x, M_y and M_z, be the control torques exerted by the motors on the vehicle. Also let F_z be the resultant force exerted by the motors along the vehicle's z axis. When all motors are working properly, the relation between the (M_x, M_y, M_z, F_z) 4-tuple, and the f forces vector is given by the following equation:

$$\begin{bmatrix} M_x \\ M_y \\ M_z \\ F_z \end{bmatrix} = A(\gamma, \alpha) \cdot f, \qquad \text{with} \qquad f = \begin{bmatrix} f_1 \\ \vdots \\ f_6 \end{bmatrix}. \tag{2}$$

The force-torque matrix $A = A(\gamma, \alpha) \in \mathbb{R}^{4\times 6}$ is given by,

$$A = \tag{3}$$
$$\begin{bmatrix} \tilde{k}_t c\gamma[& -1 & \frac{\alpha+1}{2\alpha} & \frac{\alpha+1}{2\alpha} & -1 & \frac{\alpha-1}{2\alpha} & \frac{\alpha-1}{2\alpha}] \\ \tilde{k}_t\sqrt{3}c\gamma[& -\frac{1}{3\alpha} & \frac{3\alpha-1}{6\alpha} & -\frac{3\alpha-1}{6\alpha} & \frac{1}{3\alpha} & \frac{3\alpha+1}{6\alpha} & -\frac{3\alpha+1}{6\alpha}] \\ \tilde{k}_t s\gamma[& 1 & -1 & 1 & -1 & 1 & -1] \\ -s\gamma[& 1 & 1 & 1 & 1 & 1 & 1] \end{bmatrix}.$$

In order to mathematically represent the case of a failure in the *i*-th rotor, the A force-torque matrix, should be replaced by $\tilde{A}_i$ matrix, hence:

$$\begin{bmatrix} M_x \\ M_y \\ M_z \\ F_z \end{bmatrix} = \tilde{A}_i(\gamma, \alpha) \cdot f \tag{4}$$

The problem to study consists in solving the following inverse problem: given a desired torque-force 4-tuple (M_x, M_y, M_z, F_z), we want to find an $f \in \mathbb{R}^6$ that solves equation (4). As stated before, to be valid, a solution must be positive, since the force that the motors can exert is only in one direction. Moreover, the forces' modulus, must be lower than the maximum thrust that motors can generate, i.e., $f_i \in [0, F_M]$, for $i = 1, ..., 6$.

The most common and frequently used solution for this problem relies on the Moore-Penrose pseudoinverse of A, as it gives the minimum $\|\cdot\|_2$ (*euclidean norm*) solution and, consequently, the lowest power configuration for the motors to achieve the desired torques and thrust:

$$f_0 = {\tilde{A}_i}^{\dagger}(\gamma, \alpha) \cdot \begin{bmatrix} M_x \\ M_y \\ M_z \\ F_z \end{bmatrix}. \tag{5}$$

However, this may not always yield a valid solution, as stated before.

IV. Computing the optimal $\|\cdot\|_\infty$ solution

Without loss of generality due to vehicle symmetry, only the case of a total failure in motor number 2 will be considered.

Given a desired torque–force $(M_x, M_y, M_z, F_z) \in \mathbb{R}^4$ with $F_z > 0$, the set of solutions of equation (4) can be written as:

$$f = \underbrace{\tilde{A}_2^{\dagger}(\gamma, \alpha) \cdot \begin{bmatrix} M_x \\ M_y \\ M_z \\ F_z \end{bmatrix}}_{f_0} + \beta w \tag{6}$$

with $w \in N(\tilde{A}_2)$, $\beta \in \mathbb{R}$ and f_0 being the minimal euclidean norm given by the Moore-Penrose pseudinverse. The kernel of $\tilde{A}_2 = \tilde{A}_2(\gamma, \alpha)$ is given by,

$$N(\tilde{A}_2) = span\left\{ \overbrace{\begin{pmatrix} \frac{3}{4}\frac{(\alpha-1)(\alpha+\frac{1}{3})}{\alpha} \\ 0 \\ -\frac{3}{4}\frac{(\alpha+1)(\alpha-\frac{1}{3})}{\alpha} \\ -\frac{3}{4}\frac{\alpha^{\alpha}_{2+\frac{1}{3}}}{\alpha} \\ 1 \\ \frac{3}{4}\frac{\alpha^2+\frac{1}{3}}{\alpha} \end{pmatrix}}^{w} \right\}. \tag{7}$$

Remark 1. *Notice that, since $0 < |\alpha| < 1$ ([6, Theorem 3]), certain components of w could be zero, namely w_1 or w_3 for $\alpha = -\frac{1}{3}$ or $\alpha = \frac{1}{3}$, respectively. It can be shown, that this observation carried out when analyzing the case of a failure in rotor number 2, is valid similarly in the case of a failure in any other motor. Apart to the component of w corresponding to the faulty motor (w_2 in this case), only one more component of w could be zero.*

Taking into account the constrains ($f_i \in [0, F_M]$, for every $i = 1, ..., 6$), the possible values of β must satisfy:

$$0 \leq f_0 + \beta w \leq F_M \tag{8}$$

which renders the following set of constraints on β:

$$f_{0_i} + \beta w_i \geq 0 \Rightarrow \begin{cases} \beta \geq \frac{-f_{0_i}}{w_i} & \text{if} \quad w_i > 0 \\ \beta \leq \frac{-f_{0_i}}{w_i} & \text{if} \quad w_i < 0 \end{cases} \tag{9}$$

$$f_{0_i} + \beta w_i \leq F_M \Rightarrow \begin{cases} \beta \leq \frac{F_M - f_{0_i}}{w_i} & \text{if} \quad w_i > 0 \\ \beta \geq \frac{F_M - f_{0_i}}{w_i} & \text{if} \quad w_i < 0 \end{cases} \tag{10}$$

$$i = 1, 2, ..., 6$$

If $w_i = 0$ for any $i = 1, ..., 6$, one of two possibilities exist: 1) if the corresponding f_{0_i} is equal or greater than zero, no additional restrictions appear. But if $f_{0_i} < 0$, there is not any value of β that satisfies Eq. (8), and a range for β does not exist.

Suppose that there exists a solution $f \in \mathbb{R}^6$ to Eq.(6) which satisfies the constraints $f_i \in [0, F_M]$ for every $i = 1, ..., 6$. This means that there exists a real nonempty interval $\mathcal{I}$ such that, if

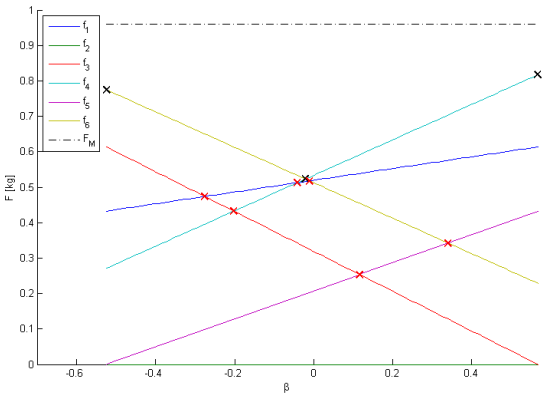

Fig. 3. Linear equations for β

$\beta \in \mathcal{I}$, the set of feasible solutions are given by equation (6). In this case, there is a value that will be denoted β_{opt} so that the force exerted by the most stressed motors will be minimal. More specifically, we are interested in finding $\beta_{opt} \in \mathcal{I}$ such that,

$$\|f_0 + \beta_{opt} w\|_\infty \leq \|f_0 + \beta w\|_\infty, \tag{11}$$

for every $\beta \in \mathcal{I}$.

In equation (6), it can be seen that the set of feasible forces is described by six linear equations where β is the optimization variable. The next lemma characterizes the optimal point β_{opt}.

Lemma 1. *Let $f_0, w \in \mathbb{R}^n$, and $a < b$ such that $f_0 + \beta w \geq 0$ for every $\beta \in [a, b]$. Suppose that $w_i \neq 0$ for every $i = 1, .., n$ and the pairs $(w_i, f_{0_i}) \neq (w_j, f_{0_j})$ for every $i \neq j$. If $\beta_0 \in (a, b)$ satisfies $\|f_0 + \beta_0 w\|_\infty \leq \|f_0 + \beta w\|_\infty$ for every $\beta \in (a, b)$, then there exists $j \neq i$ such that $f_{0i} + \beta_0 w_i = f_{0j} + \beta_0 w_j$.*

In Fig. 3 an example is shown for a given torque and force. The valid range of β can be seen, with the potentially possible optimal points (black crosses), and the intersection points that are invalid (red crosses) due to the fact that in those cases, some motors would be exerting a greater force than the ones corresponding to the intersection points.

V. Algorithm

In Algorithm 1, the proposed method for finding the optimal β is presented. It requires that a range of possible values β exists, and also some *a priori* data consisting of the kernel of $\tilde{A}_i$ (vector w).

The procedure goes as follows. First of all f_0 is computed (Eq. 6). It must be verified from an *a priori* analysis, if any w_i is equal to zero (see Remark 1), and in that case, if the corresponding f_{0_i} is positive or zero in order to ensure whether a solution exists. If that is not the case, no value of β that yields a positive force set solution exists.

For each pair of motors i and j, not considering neither the faulty motor nor the $w_i = 0$ motor, compute the β_{int} value, which renders equal forces for motors i and j. Also, if both motors share the same w_i and f_{0_i}, the intersection computation must be skipped as it is the same line. Compute the corresponding force value as well, using the computed f_0 and w. The intersection may or may not occur in the valid range of β, with only the former case being relevant for the solution. The forces exerted by the intersected motors at this β_{int} value must be equal or greater than the the forces calculated for the rest of the motors. If these conditions are both satisfied, this may be a possible optimal point. In Algorithm 1, Ω denotes the set of pairs (β, f_{max}) that contain the information of the value of β and the maximum force exerted at the valid intersections.

Two other pairs are considered as possible optimal solutions, being the boundaries of β, and the force exerted in those points by the most stressed motor(s). Finally, the β_{opt} is the one associated with the minimum force of all pairs in Ω.

Algorithm 1 Minimum search algorithm

Require: $\beta_{min} \leq \beta_{max}$
Require: m_F = # of the motor that is failing
Require: $\#M$ = # of motors
Require: $w = null(\tilde{A}_i)$
Require: $q = [M_x \ M_y \ M_z \ F_z]'$
procedure GET_β_{opt}
  $f_0 = \tilde{A}^{\dagger}_{m_i} \cdot q$
  if m_F == even **then**
    x=m_F+1
  else
    x=m_F-1
  end if
  if check_zero(f_{0_x},w_x) == ERROR **then**
    break
  end if
  for i=1 ...#M-1 **do**
    for j=2 ...#M **do**
      if $i, j \neq m_F \cap$!(w_i=$w_j \cap f_{0_i}$=f_{0_j}) **then**
        β_{int} = get_intersection(mot_i,mot_j, f_0, w)
        check1: $\beta_{max} \geq \beta_{int} \geq \beta_{min}$
        check2: $f_k \leq f_i$, $f_k \leq f_j$, $k = 1 \ldots \#M$
        if check1 and check2 **then**
          $\Omega \leftarrow (\beta_{int}, f_{int})$
        end if
      end if
    end for
  end for
  $\Omega \quad (\beta_{min}, f_{max}^{\beta_{min}})$
  $\Omega \quad (\beta_{max}, f_{max}^{\beta_{max}})$
  $(\beta_{opt}, f_{\beta_{opt}}) \leftarrow \min_f \Omega$
end procedure

VI. Results

In this section simulated results of the actuator allocation method will be shown. For simplicity, all of them are based upon commanding a non-zero torque on one axis at a time (for

TABLE I
VEHICLE TECHNICAL SPECS

Variable	Value	Units
l (arm length)	0.375	m
P (weight)	2	kg
F_M (max. thrust)	0.950	kg
γ	107	deg
k_f	0.0667	-
k_t	0.0334	-
$\tilde{k}_t$	0.5	-

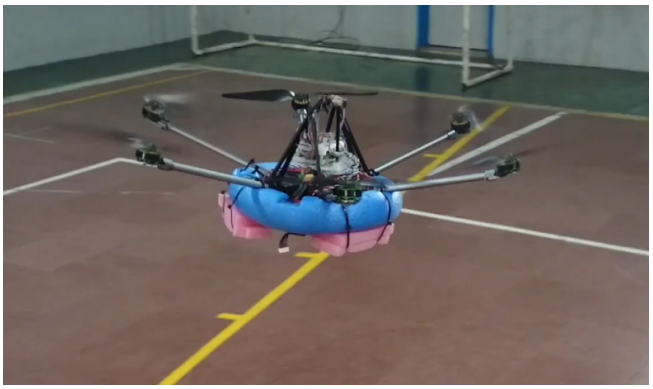

Fig. 4. Hexacopter used for testing purposes.

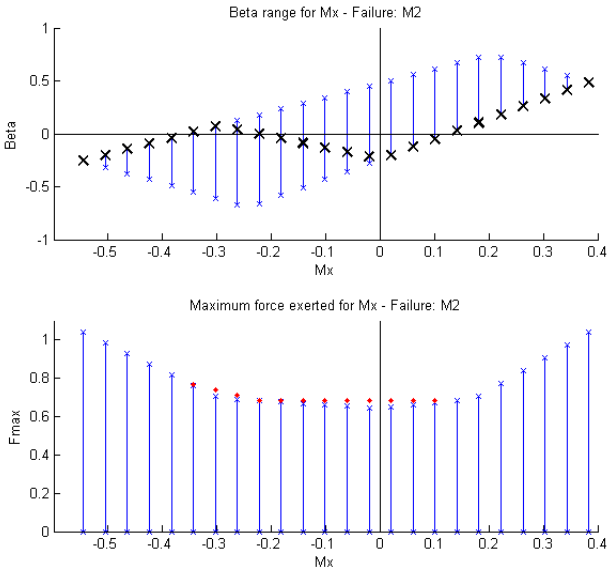

Fig. 5. Optimal β for motor 2 failure and varying M_x.

instance $M_x \neq 0$), while the torques commanded on the other to axes, are set to zero (for instance $M_y = 0$ and $M_z = 0$). The vertical thrust is always set to be equal to the weight of the vehicle, so as to try to maintain the hovering state.

In order to carry out all simulations, the model of a custom built hexacopter was used. The parameters of the vehicle's model, are listed in Table I. A photograph of the vehicle in flight can be seen in Fig. 4, where it can be noticed that only five motors are spinning, illustrating fault tolerance. A short video clip of this flight is available as well ([20]).

Observe that, there is an output of Algorithm 1 if and only if there exists solution of Eq. (4). Furthermore, the output of this algorithm will give the solution that minimizes the force of the most stressed motor. In general $\beta \neq 0$, with $\beta = 0$ corresponding to the Moore-Penrose pseudoinverse solution.

In Figures 5, 6 and 7, three cases are presented. Case 1 (Fig.5): $M_x \neq 0$, $M_y = 0$, $M_z = 0$. Case 2 (Fig.6): $M_x = 0$, $M_y \neq 0$, $M_z = 0$. Case 3 (Fig.7): $M_x = 0$, $M_y = 0$, $M_z \neq 0$. For each figure, the upper curve depicts the variation of the valid β range. For a discrete number of torque values on the abscissae axes, the vertical blue lines, represent the interval of valid β values such that a solution of Eq. (4) exists, with M_x, M_y, M_z in $\mathbf{kg} \cdot \mathbf{m}$ units. The computed optimal β_{opt} for each commanded torque is marked with cross.

The lower graphic shows the force exerted by the most stressed motor (in **kg**) when using the solution associated with β_{opt} (solid blue line) in comparison with the same value corresponding to the pseudoinverse solution (red dots, only if the pseudoinverse yields a valid force set, i.e., if $\beta = 0$ belongs to the β range).

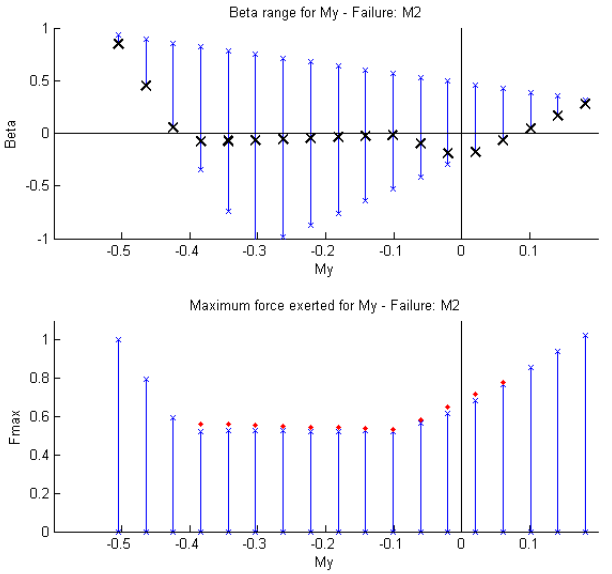

Fig. 6. Optimal β for motor 2 failure and varying M_y

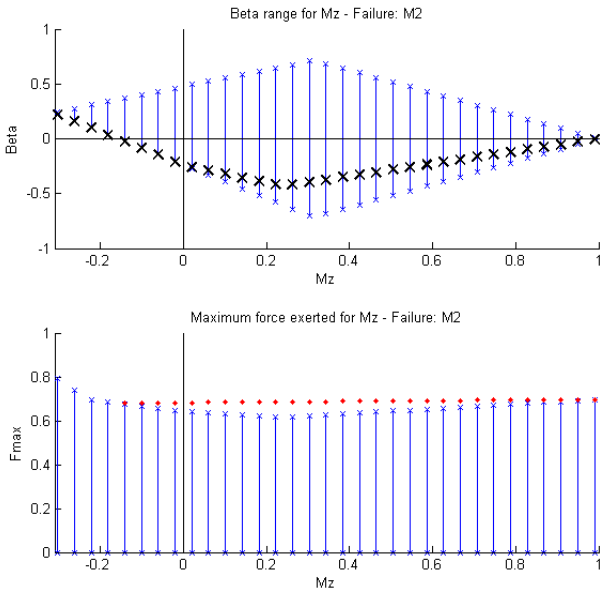

Fig. 7. Optimal β for motor 2 failure and varying M_z

It can be seen that the proposed solution presents two advantages with respect to the pseudoinverse. First, the most stressed motor reduces its exerted force, working further away from saturation. This is more evident when torque along the z axis is exerted, where it is shown that the force is reduced as much as 70 grams, and 40 grams in the hovering case. This amount represents approximately an 8% lower overall force for the motor in the best case. While in normal flying condition the motors are working at around 60% of their maximum capacity, when a motor fails there appears one motor that is always much more stressed than the others, resulting in a near-saturation working condition. Thus the importance of relieving the working condition of this particular motor, even a little.

The second improvement, that can be seen in all the figures, is that the proposed solution yields an optimal β even in cases where the pseudoinverse cannot render a valid force set, thus allowing for a larger range of achievable moments in each axis. For the results shown, the achievable range of torques goes from -0.34 $\sim$ 0.14 to -0.54 $\sim$ 0.38 for M_x, giving an improved range of +92%, from -0.42 $\sim$ 0.08 to -0.5 $\sim$ 0.18 for M_y (+36%) and from -0.14 $\sim$ 0.99 to -0.3 $\sim$ 0.99 for M_z (+14%), in the particular case for a failure in motor 2.

Also, in Fig. 8, an extreme case is presented. This is exactly the same plot as before, but considering that the maximum force that can be exerted by any motor is much lower (for example, when using smaller motors). In this case, the pseudoinverse will never give a valid force set solution, whereas the proposed solution does, and for a large range of desired moments. While not depicted here, for the same situation along the x axis the graphic is similar. In the case of the y axis there exists a range of valid pseudoinverse solutions

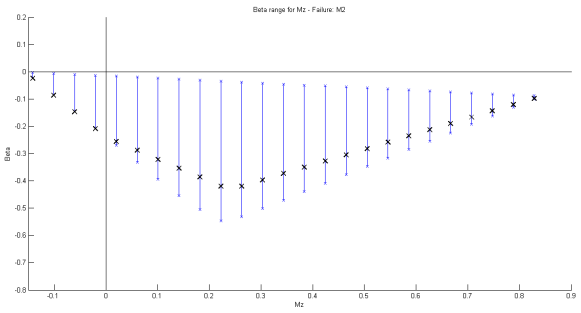

Fig. 8. Optimal β for motor 2 failure and varying M_z - Worst case scenario

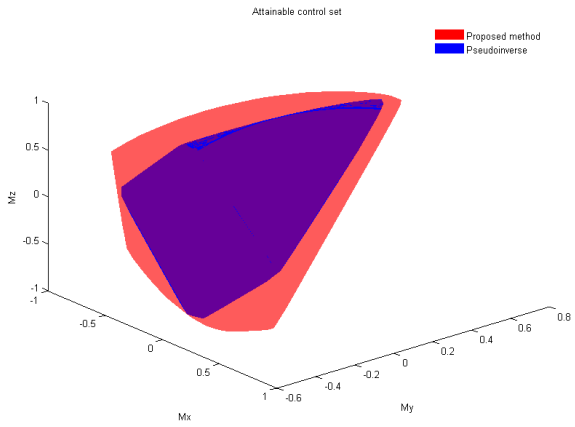

Fig. 9. Achievable set of moments for pseudoinverse and proposed algorithm solutions

which does not include $M_y = 0$, while for the proposed solution it does, so the latter offers the possibility of achieving hovering.

Finally, in Fig. 9, the set of moments (for vertical thrust matching the vehicle's weight) that can be reached using the pseudoinverse solution versus the proposed algorithm solution are illustrated. It can be seen that the former is strictly contained in the latter, so the proposed solution will always allow for a larger set of achievable control action.

However, there are some points to consider before assuming the proposed solution may be always better than the pseudoinverse one.

Firstly, one may consider that, as the pseudoinverse yields the minimum norm solution, the power consumed by the vehicle would be the lowest possible. This is true, but, when comparing the total power consumed by the motors in each case, using experimental data on the force-current relation, it is noted that the additional power needed is as little as 0.1% more.

Secondly, all the calculations presented here need to be carried out on microcontroller based flight computer, where the proposed algorithm must be implemented. Frequently, the flight computer's microcontroller is a processor with limited

resources. A question that might be raised concerns the computation capacity required to run the proposed algorithm, and if the overall performance of the vehicle may be affected.

Thus, in the section VIII, a comparison will be presented showing the computational burden undertaken by a given flight computer, when solving the torque-force problem through the pseudoinverse, and when running the proposed algorithm.

VII. Simulations

For comparison between the proposed solution and the pseudoinverse solution during a flight, an hexacopter MATLAB/Simulink model was used. The simulator is identical for both solutions except for the control allocator, which obtains the motor force set from the desired torques and vertical force. The control system will try to maintain a fixed position in the air.

In the following simulation, a lateral wind is emulated by rising a perturbation torque on the **y** axis linearly in 2 seconds to $0.68Nm$, holding it for a second, and going back to zero in two seconds. As the control system will try to compensate this by generating an opposed torque of equal magnitude, and considering that the pseudoinverse does not give a solution tat satisfies the force constraints ($f_i \in [0, F_M]$ for every $i = 1, ..., 6$) for the desired torque (while the proposed solution does), performance will be degraded, leading to instability during a brief time because the solution corresponding to the pseudoinverse forces the motors to operate in saturation. When the perturbation torque magnitude decreases enough to get a valid motor force with the pseudoinverse solution, the system regains control and goes back to the reference position.

In Fig. 10, the lower graphic shows the simulated disturbance torques exerted on the three axes, where only the one exerted on the **y** axis rises. In the upper graphic, the vehicle's transient behavior can be seen, showing the responses in the roll, pitch and yaw angles, for the two simulated actuator allocation strategies. While the perturbations are zero or low enough, an almost identical behavior is achieved with the pseudo-inverse and with the new method. But, when the magnitude of the perturbation is too high for the pseudoinverse solution to get a valid set of motor commands, performance begins to degrade, as the vehicle cannot hold the position and begins to drift, due to the fact that the disturbance torque cannot be compensated. On the other hand, the new proposed solution achieves the desired torques and vertical force and as a consequence it is able to hold the vehicle's desired orientation and position. Note how the proposed solution changes mainly the pitch angle to compensate for the perturbation along the **y** axis, while the pseudoinverse solution makes different maneuvers due to not considering the saturation of the motors.

The transient behavior of the vehicles position, as achieved by both solutions, is shown in Fig. 11. A remarkable difference can be appreciated, especially in the upper curve. As it can be seen, using the proposed actuator allocation technique, the displacement mainly takes place along the direction expected (latitude), considering the simulated applied

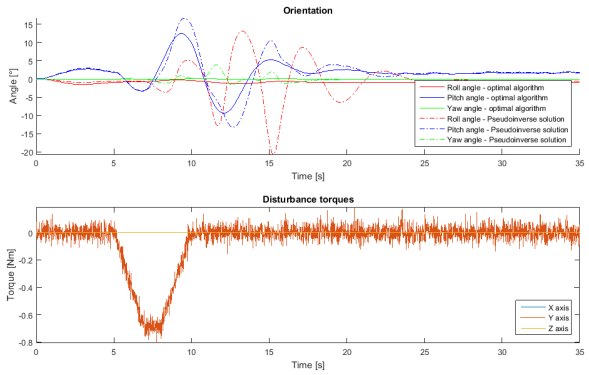

Fig. 10. Orientation of the vehicle and perturbations for optimal and pseudoinverse solutions

disturbance torque. The pseudo-inverse method on the other hand, exhibits a poorer response.

Also in Fig. 12, the signal commanded to each motor is shown. Once the diturbance torque is too high and the pseudoinverse solution finds holding position hard, the PWM values lead one of the motors to a full stop (6sec), and another one to saturation (11sec - 15sec). On the other hand, throughout this simulation, the proposed solution leads to a valid motor force set, thus leading the PWM signals to remain within valid operating limits.

Finally, in Fig. 13, a similar simulation is carried out, but with the maximum torque exerted in the **y** axis being $0.86Nm$. In this case, the magnitude of the disturbance is too high and the pseudoinverse cannot obtain a valid force solution during a long period of time, so the vehicle becomes unstable, resulting in a half barrel roll and finally a crash landing. But again in this case, the proposed solution always gives a valid motor force set solution, using mainly the pitch angle to compensate the disturbance.

VIII. Experimental results

The flight computer used in this experiment is a custom design developed within the GPSIC Lab ([21], [22]) for research purposes, with the sensors and all the electronics needed to operate a multirotor-type flying vehicle. The board's sensors are an MPU6000 Inertial Measurments Unit (IMU), a BMP180 barometer, and an HMC5883 digital compass (see Fig. 14). The board's microcontroller is an LPC1769 from NXP, an ARM Cortex-M3 class microcontroller running at 100MHz which does not have a Floating Point Unit (FPU).

The on-board computer's tasks include sensors data acquisition, sensors data fusion, and PID control of the altitude, roll, pitch and yaw axes. Also, the actuator allocation signals for the motors must be computed, with telemetry and radio control reception as well.

We will refer to the control loop as the piece of code that makes an estimation of the orientation of the vehicle, then through a PID controller obtains the desired torques and vertical force, and finally uses the chosen control allocation

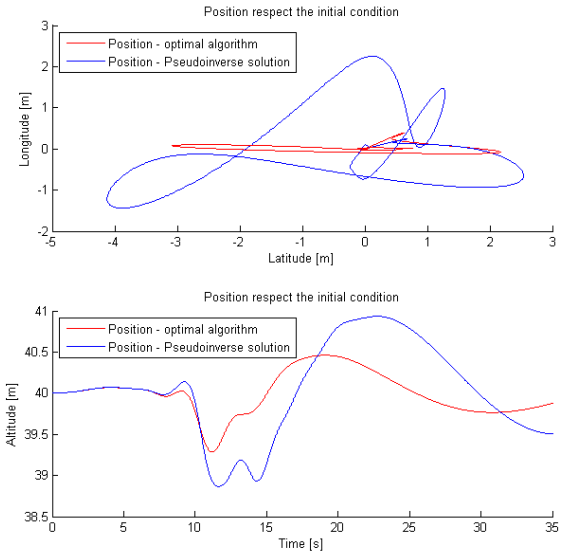

Fig. 11. Position of the vehicle for optimal and pseudoinverse solutions

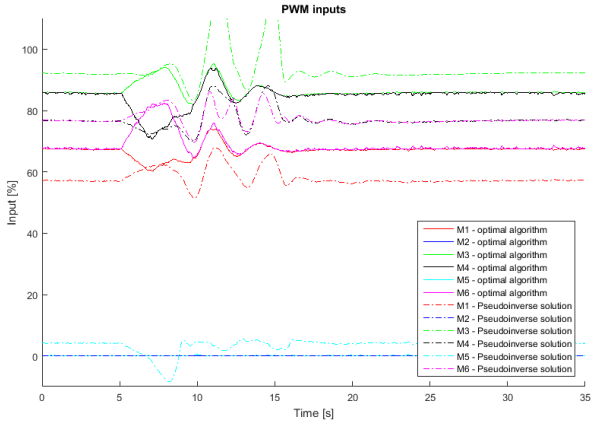

Fig. 12. PWM values for optimal and pseudoinverse solutions

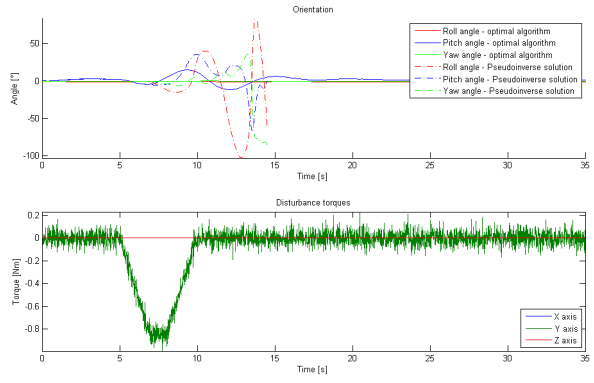

Fig. 13. Orientation of the vehicle and perturbations for optimal and pseudoinverse solutions, for a torque that canot be compensated by the latter

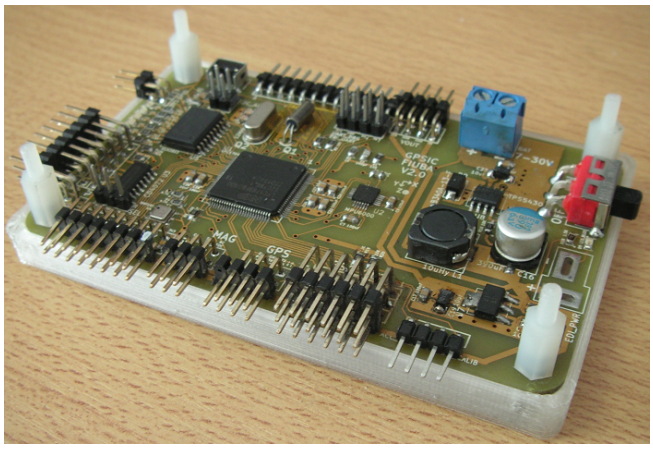

Fig. 14. Flight computer used in experimental tests

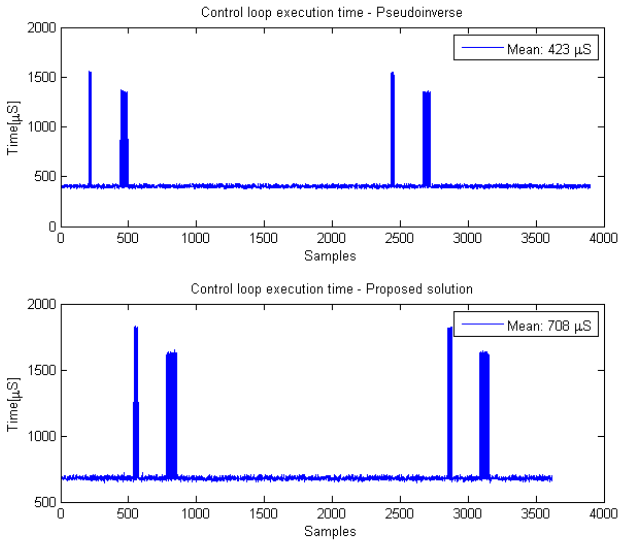

Fig. 15. Execution time for the control loop

solution to obtain the force set to command the motors. The execution time of the control loop, will be of interest for the pseudoinverse and the proposed solution, to get and idea of the extra computing time that the latter requires. It is worth mentioning that the control loop task may be interrupted by other tasks with higher priorities (that may or may not have constant timing), so its execution time will not be constant.

The results are shown in Fig. 15. The control loop in this example runs at $200Hz$, every $5mS$, and the pseudoinverse solution is executed, on average, in $423\mu S$, while the proposed solution increases the average (and also peak) execution time in $\sim 300\mu S$. While the time increase seems large at first sight, the control loop execution time roughly takes on average less than 15% of the time between loops, with peaks that do not reach 40% of that time.

With the current state of the art, more advanced micro-

controllers with higher capacity and FPU are used in flight controllers, where the processing time would be much lower and not present any limitation with timing restrictions.

IX. Conclusions

An immediate conclusion of this work, is that the actuator allocation problem in multirotors turns out to be the key to solving important practical issues, such as fault tolerance.

More specifically, it can be said that on one hand, the research presented in [6] probed for the hexagon shaped hexarotor, that only through working with tilted rotors is fault tolerance achievable. On the other hand, the investigation presented here probes the usefulness of minimizing the $\|\cdot\|_\infty$ norm leaving the $\|\cdot\|_2$ norm behind, as it renders various benefits.

Not only does the approach proposed here allow for a larger control allocation space than the one obtainable through the Moore-Penrose pseudoinverse, but this is also carried out at a reasonable computational cost, with barely any penalty on battery charge duration.

In the simulations, it has been shown that greater torques in all directions may me reached. It is also shown that, the proposed algorithm effectively reduces the stress on the fastest spinning motor, which is critical as its saturation usually renders a highly degraded performance. Moreover, this allows for maneuvers that are impossible to achieve using the pseudoinverse solution.

In the computational burden experiment, it has been proved that the additional computing effort needed to implement the proposed method is not significantly higher than the pseudoinverse one, even running on a modest microcontroller.

The algorithm still remains to be tested in real flight, using the shown vehicle which is currently using the pseudoinverse solution (see [20]).

Acknowledgment

This work has been partially supported by Universidad Nacional de la Patagonia Austral (PI29/C066), Agencia Nacional de Promoción Científica y Tecnológica, FONCYT PICT 2014-2055 (Argentina), and ITBA Grant ITBACyT-28.

Claudio Pose would like to thank Peruilh Foundation for their support.

References

[1] K. P. Valavanis and G. J. Vachtsevanos, Eds., *Handbook of Unmanned Aerial Vehicles*. Springer Netherlands, 2015.

[2] G. X. Du, Q. Quan, B. Yang, and K. Y. Cai, "Controllability analysis for multirotor helicopter rotor degradation and failure," *Journal of Guidance, Control and Dynamics*, vol. 38, no. 5, pp. 978–985, 2015.

[3] T. Schneider, "Fault-tolerant multirotor systems," Master's thesis, Swiss Federal Institute of Technology (ETH), 2011.

[4] M. W. Mueller and R. D'Andrea, "Stability and control of a quadrocopter despite the complete loss of one, two, or three propellers," in *IEEE International Conference on Robotics and Automation*, May 2014, pp. 45–52.

[5] M. Achtelik, K. M. Doth, D. Gurdan, and J. Stumpf, "Design of a multi rotor mav with regard to efficiency, dynamics and redundancy," in *AIAA Guidance, Navigation, and Control Conference*, 2012.

[6] J. I. Giribet, R. Sánchez-Peña, and A. Ghersin, "Analysis and design of a tilted rotor hexacopter for fault tolerance," *IEEE - Trans. on Aerospace and Electronics Systems*, vol. 52, no. 4, 2016.

[7] A. Marks, J. F. Whidborne, and I. Yamamoto, "Control allocation for fault tolerant control of a VTOL octorotor," in *Proceedings of 2012 UKACC International Conference on Control*, Sept 2012, pp. 357–362.

[8] P. Segui-Gasco, Y. Al-Rihani, H. S. Shin, and A. Savvaris, "A novel actuation concept for a multi rotor UAV," in *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, May 2013, pp. 373–382.

[9] G. Du, Q. Quan, B. Yang, and K. Cai, "Controllability analysis for a class of multirotors subject to rotor failure/wear," *CoRR*, vol. abs/1403.5986, 2014. [Online]. Available: http://arxiv.org/abs/1403.5986

[10] T. Schneider, G. Ducard, R. Konrad, and S. Pascal, "Fault-tolerant Control Allocation for Multirotor Helicopters Using Parametric Programming," in *International Micro Air Vehicle Conference and Flight Competition (IMAV)*, Braunschweig, Germany, Jul. 2012. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01302202

[11] D. Vey and J. Lunze, "Experimental evaluation of an active fault-tolerant control scheme for multirotor UAVs," in *3rd International Conference on Control and Fault-Tolerant Systems*, September 2016, pp. 119–126.

[12] X. Qi, D. Theilliol, J. Qi, Y. Zhang, and J. Han, "A literature review on fault diagnosis methods for manned and unmanned helicopters," in *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, May 2013, pp. 1114–1118.

[13] D. Vey and J. Lunze, "Structural Reconfigurability Analysis of Multirotor UAVs after Actuator Failures," in *54th Conference on Decision and Control*, December 2015, pp. 5097–5104.

[14] A. Freddi, S. Longhi, A. Monteriù, and M. Prist, "Actuator fault detection and isolation system for an hexacopter," in *10th International Conference on Mechatronic and Embedded Systems and Applications*, September 2014.

[15] Y. A. Younes, A. Rabhi, H. Noura, and A. E. Hajjaji, "Sensor fault diagnosis and fault tolerant control using intelligent-output-estimator applied on quadrotor UAV," in *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, June 2016, pp. 1117–1123.

[16] L. Qin, X. He, Y. Zhou, and D. Zhou, "Fault-tolerant control for a quadrotor unmanned helicopter subject to sensor faults," in *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, June 2016, pp. 1280–1286.

[17] M. Saied, B. Lussier, I. Fantoni, C. Francis, H. Shraim, and G. Sanahuja, "Fault diagnosis and fault-tolerant control strategy for rotor failure in an octorotor," in *IEEE International Conference on Robotics and Automation*, May 2015, pp. 5266–5271.

[18] H. Alwi and C. Edwards, "Fault tolerant control of an octorotor using lpv based sliding mode control allocation," in *American Control Conference*, June 2013, pp. 6505–6510.

[19] G. Ducard and M. D. Hua, "Discussion and practical aspects on control allocation for a multi-rotor helicopter," in *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 38, Sep. 2011, pp. 95–100.

[20] "Fault tolerant control for a tilted rotor hexacopter," https://www.youtube.com/watch?v=a3-R2M5xWGA, accessed: 2017-02-06.

[21] "Grupo de Procesamiento de Señales, Identificación y Control - Home page," http://psic.fi.uba.ar.

[22] P. Moreno, C. Pose, and J. I. Giribet, "Ins/ultrasound navigation system," in *Congreso Argentino de Sistemas Embebidos*, U. de Buenos Aires, Ed., vol. 1, no. 1. SASE, 2015, pp. 10–18.