Submitted to the AIAA Science and Technology Forum and Exposition 2016
San Diego, CA, 4-8 January, 2016

Identification and reconfigurable Control of Impaired Multi-Rotor Drones

Vahram Stepanyan [1]
University of California Santa Cruz, Santa Cruz, CA 95064

Kalmanje Krishnakumar [2]
NASA Ames Research Center, Moffett Field, CA 94035

Alfredo Bencomo [3]
Stinger Ghaffarian Technologies Inc., Moffett Field, CA 94035

The paper presents an algorithm for control and safe landing of impaired multi-rotor drones when one or more motors fail simultaneously or in any sequence. It includes three main components: an identification block, a reconfigurable control block, and a decisions making block. The identification block monitors each motor load characteristics and the current drawn, based on which the failures are detected. The control block generates the required total thrust and three axis torques for the altitude, horizontal position and/or orientation control of the drone based on the time scale separation and nonlinear dynamic inversion. The horizontal displacement is controlled by modulating the roll and pitch angles. The decision making algorithm maps the total thrust and three torques into the individual motor thrusts based on the information provided by the identification block. The drone continues the mission execution as long as the number of functioning motors provide controllability of it. Otherwise, the controller is switched to the safe mode, which gives up the yaw control, commands a safe landing

[1] Senior Research Scientist, University Affiliated Research Center, NASA Ames Research Center/Mail Stop 269/1, AIAA Senior Member, email: vahram.stepanyan@nasa.gov

[2] Autonomous Systems and Robotics Branch Chief, Intelligent Systems Division, NASA Ames Research Center/Mail Stop 269/1, AIAA Associate Fellow, email: kalmanje.krishnakumar@nasa.gov

[3] Autonomous Systems and Robotics Technical Area Liaison, NASA Ames Research Center/Mail Stop 269/3, email: alfredo.bencomo@nasa.gov

spot and descent rate while maintaining the horizontal attitude.

I. Introduction

Drones are becoming increasingly popular for research, commercial and military applications due to their affordability resulting from their small size, low cost and simple hardware structure. One of the critical aspects of these uses is the reliability of the drones while maintaining their affordability. In particular, the civilian applications of drones are subject to safety requirements for the drones themselves and for the environment they are operating in. Due to weight and cost constraints, the hardware redundancy is not an option in improving the reliability and safety of the drones, which make them vulnerable to motor failures leading to potentially unsafe operations or collisions.

Majority of the existing approaches is related to the fault estimation and control problems of impaired drones with partial loss of actuator effectiveness and employ robust, adaptive and gain scheduling control strategies to follow desired commands. A review of some early results on the control problem of multi-rotor drones with actuator faults can be found in [19], and experimental results on some of the actuator fault-tolerant control techniques for a quadrotor can be found in [4].

Recent reports in the field of the fault tolerant control design use both direct and indirect approaches. In the first case, the controller is designed without explicitly identifying the faults. In [15], a proportional-integral-derivative (PID) controller is designed off-line for each fault of the quadrotor's actuators and a gain scheduling is implemented on-line assuming that the fault is known. In [17], an adaptive feedback linearization technique is presented for fault recovery of a quadrotor that is subject to a partial loss of effectiveness in one or more actuators. A dynamic inversion controller augmented with an off-line trained single network adaptive critic is applied to control an uncertain quadrotor in [18], where the uncertainties are estimated on-line using another neural network. In [1], a fault tolerant control scheme for multi-rotor drones with high actuator redundancy is presented, which is based on the integral sliding mode and fixed control allocation. A quaternion-based adaptive attitude control for a quadrotor in the presence of external disturbances and partial loss of rotor effectiveness is proposed in [20]. A cascaded controller for a hexacopter is presented in [8] using an extended state observer, which estimates modeling errors and propulsion efficiency degradation.

In the indirect approach, first the faults are estimated, then proper controllers are designed. In many case the main tool is Thau observer (see for example [10], [11], [3]). In [10], Thau observer is used to design a fault diagnostic system while stabilizing the quadrotor at low speed with a controller based on the nested saturation. Thau observer based actuator faults detection and isolation scheme for a hexacopter is presented in [11]. In [3], an adaptive Thau observer is used to estimate the quadrotor actuator faults, rate them based on the predefined fault-tolerant boundaries and compensate for depending on the severity levels. Other approaches use model-based observations ([12]), Kalman filter ([21]), interacting multiple model filter and switching multi-model predictive control ([2]) and polynomial observer ([5]).

On the other hand, the identification and control of multi-rotor drones get more complex when one or more motors completely fail leading to controllability loss of one or more degrees of freedom. Few approaches have been reported in this case.

When the drone has enough actuator redundancy, for example as in an octocopter, and the failures are known, control allocation schemes can be used to handle rotor failure [14]. Otherwise, not all degrees of freedom can be controlled properly. In [13], a controller is design in case of a single rotor failure in quadrotor vehicles using robust feedback linearization sacrificing the yaw directional controllability and assuming that the failure is known. Periodic solutions for a quadrotor with a known single, two opposing, or three propellers lost are presented in [16]. In each case, the drone spins about an axis found from some equilibrium conditions and fixed in the body frame. Only in two motor failure case this axis is vertical permitting a safe landing, which essentially resembles the solution in [13]. In [6], an iterative on-line optimization method is applied to a quadrotor way point tracking with single and double rotor failure. However, the real-time convergence may be an issue for small drones with restricted computational power. In [9], an algorithm for the on-line detection of a single motor failure and a control allocation technique is proposed, assuming that inertial forces and torques acting on the multi-rotor vehicle and motor thrusts can be measured, which may not be the case for some drones.

This paper presents an algorithm for control and safe landing of impaired multi-rotor drones when one or more motors failing simultaneously or in any sequence. It includes three main components: an identification block, a reconfigurable control block, and a decisions making block. The identification block monitors each motor load characteristics and the current drawn, based on which the failures are detected. The control block generates the required total thrust and three axis torques for the altitude, horizontal position and/or orientation control of the drone based on the time scale separation and nonlinear dynamic inversion. The altitude is directly controlled by the total thrust generated by the motors. The horizontal displacement as well as the orientation of the drone are controlled using time scale separation and nonlinear dynamic inversion, where the torques are used to control the fastest variables, that is angular rates, which are used to control the corresponding orientation angles. The last step is to use roll and pitch angles to control the horizontal displacement of the drone. The decision making algorithm maps the total thrust and three torques into the individual motor thrusts based on the information provided by the identification block. The drone continues the mission execution as long as the number of functioning motors are at least four, which is sufficient for the control of its altitude and orientation. Otherwise, the controller is switched to the safe mode, which gives up the yaw control, commands a safe landing spot and descent rate while maintaining the horizontal attitude.

Our approach extends the result of [16], [13] and [9] by allowing more than one failure at a time and introducing more reliable and computationally inexpensive identification method. In addition, if a failed motor starts producing a thrust, our algorithm detects the change and appropriately reconfigures the controller.

II. Dynamic Model

The mathematical model of the drone is obtained using Newton-Euler formalism considering only rigid body motions. Let the position of the center of mass of the drone in the inertial frame F_I with vertical z-axis be

$$\boldsymbol{r} = x\boldsymbol{i} + y\boldsymbol{j} + z\boldsymbol{k}\,, \tag{1}$$

where $\boldsymbol{i},\ \boldsymbol{j},\ \boldsymbol{k}$ are the corresponding unit vectors. The translational dynamics of the drone satisfy the equation

$$M\ddot{\boldsymbol{r}}(t) = L_{B/I}(t)\boldsymbol{T}(t) + \boldsymbol{D}(t) + M\boldsymbol{g}\,, \tag{2}$$

where M is the mass,

$$L_{B/I} = \begin{bmatrix} \cos\theta\cos\psi & \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi \\ \cos\theta\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix},$$

is the rotation matrix from the body frame F_B to F_I with $\phi,\ \eta,\ \psi$ being the associated Euler angles (see for example [7], p. 313), $\boldsymbol{T}(t)$ is the total thrust vector generated by the motors, $\boldsymbol{D}(t)$ is the atmospheric drag force, and $\boldsymbol{g} = [0\ 0\ \ -g]^\top$ is the gravity acceleration.

It is assumed that F_B frame is aligned with the drone's principle axis of inertia, all motors generate thrust in the z-direction in F_B frame, that is $\boldsymbol{T} = [0\ 0\ T]^\top$ in F_B, where $T = \sum_{i=1}^n f_i$, f_i is the thrust generated by the i-th motor, and the atmospheric drag force is proportional to inertial velocity, that is $\boldsymbol{D}(t) = -k_t\dot{\boldsymbol{r}}(t)$. With this assumptions the translational dynamics can be written coordinate-wise

$$\begin{aligned} M\ddot{x}(t) &= T(t)\left[\cos\phi(t)\sin\theta(t)\cos\psi(t) + \sin\phi(t)\sin\psi(t)\right] - k_t\dot{x}(t) \\ M\ddot{y}(t) &= T(t)\left[\cos\phi(t)\sin\theta(t)\sin\psi(t) - \sin\phi(t)\cos\psi(t)\right] - k_t\dot{y}(t) \\ M\ddot{z}(t) &= T(t)\cos\phi(t)\cos\theta(t) - k_t\dot{z}(t) - Mg\,, \end{aligned} \tag{3}$$

The rotational dynamics of the drone are given in the frame F_B as follows [16]

$$J\dot{\boldsymbol{\omega}}(t) + J_r\sum_{i=1}^n \dot{\boldsymbol{\Omega}}_i(t) = -\boldsymbol{\omega}^\times(t)\left[J\boldsymbol{\omega}(t) + J_r\sum_{i=1}^n(\boldsymbol{\omega}(t) + \boldsymbol{\Omega}_i(t))\right] + \boldsymbol{\tau}(t) + \boldsymbol{\tau}_D(t)\,, \tag{4}$$

where $\boldsymbol{\omega} = [p\ q\ r]^\top$ is the angular rate of F_B with respect to the inertial frame F_I expressed in F_B, $J = diag(J_1,\ J_2,\ J_3)$ is the inertia matrix of the drone, $J_r = diag(0,\ 0,\ J_{r3})$ is the inertia matrix of the rotors (assuming identical for all of them), $\boldsymbol{\Omega}_i = [0\ 0\ \Omega_i]^\top$ is the i-th rotor angular rate in the frame F_B, $\boldsymbol{\tau}$ is the torque generated by the motors, $\boldsymbol{\tau}_D$ is the aerodynamic drag torque. The notation $\boldsymbol{a}^\times$ is introduced for the cross product operator, which for a vector $\boldsymbol{a} = [a_1\ a_2\ a_3]^\top$ is defined as

$$\boldsymbol{a}^{\times} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} .$$

Assuming that the aerodynamic drag torque is linear in angular rate $\boldsymbol{\tau}_D = -k_r \boldsymbol{\omega}$, and neglecting the contribution of the rotors on the left hand side of (4), we write the rotational dynamics componentwise

$$\begin{aligned} J_1 \dot{p}(t) &= -(J_3 - J_2) q(t) r(t) - J_{r3} q(t) \Omega(t) + \tau_1(t) - k_r p(t) \\ J_2 \dot{p}(t) &= -(J_1 - J_3) p(t) r(t) + J_{r3} p(t) \Omega(t) + \tau_2(t) - k_r q(t) \\ J_3 \dot{p}(t) &= -(J_2 - J_1) p(t) q(t) + \tau_3(t) - k_r r(t) \, , \end{aligned} \tag{5}$$

where we denote $\Omega = \sum_{i=1}^{n} (-1)^i \Omega_i$. The angular rate $\boldsymbol{\omega}$ is related to the Euler angles by means of the kinematic equations

$$\begin{bmatrix} \dot{\phi}(t) \\ \dot{\theta}(t) \\ \dot{\psi}(t) \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi(t)\tan\theta(t) & \cos\phi(t)\tan\theta(t) \\ 0 & \cos\phi(t) & -\sin\phi(t) \\ 0 & \sin\phi(t)\sec\theta(t) & \cos\phi(t)\sec\theta(t) \end{bmatrix} \begin{bmatrix} p(t) \\ q(t) \\ r(t) \end{bmatrix} . \tag{6}$$

III. Identification

The mathematical model of the rotors, which are driven by the identical DC motors, is given by the Kirchoff's current law

$$L \dot{i}_j(t) + R i_j(t) = v_j(t) - k_b \Omega_j(t) \, , \tag{7}$$

where L is the motor inductance, R is the motor resistance, $i_j(t)$ is the current flowing through j-th motor, $v_j(t)$ is the voltage input to the j-th motor, k_b is the back EMF constant, and $\Omega_j(t)$ is the j-th motor angular rate, which satisfies the differential equation

$$J_{r3} \dot{\Omega}_j(t) + k_d \Omega_j(t) = k_m i_j(t) - \tau_j^l(t) \, , \tag{8}$$

where k_d is th damping (friction) coefficient, and $\tau_j^l(t)$ is the load torque experienced by the j-th motor. Combining the two equations (7) and (8) in Laplace domain, we can write

$$(Ls + R) i_j(s) = v_j(s) - \frac{k_b}{J_{r3} s + k_d} \left[k_m i_j(s) - \tau_j^l(s) \right] \, , \tag{9}$$

solving which for $i_j(s)$ we obtain

$$i_j(s) = \frac{J_{r3}s + k_d}{(J_{r3}s + k_d)(Ls + R) + k_b k_m} v_j(s) + \frac{k_b k_m}{(J_{r3}s + k_d)(Ls + R) + k_b k_m} \tau_j^l(s)\,. \tag{10}$$

Assuming the motors are small with low inductances, the equation (10) can be simplified as

$$i_j(s) = \frac{J_{r3}s + k_d}{RJ_{r3}s + Rk_d + k_b k_m} v_j(s) + \frac{k_b k_m}{RJ_{r3}s + Rk_d + k_b k_m} \tau_j^l(s)\,. \tag{11}$$

It can be noticed that the transfer function

$$\frac{k_b k_m}{RJ_m s + Rk_d + k_b k_m}$$

is strictly positive real and the corresponding impulse response function

$$h(t) = \frac{k_b k_m}{RJ_{r3}} e^{-\frac{Rk_d + k_b k_m}{RJ_{r3}} t}$$

is always positive. Therefore, the load torque contribution in the motor current is always positive. This implies that the motor current sharply drops when the load torque vanishes, that is when the propeller has separated from the drone's body. Hence, this type of failure can be identified by comparing the motor current with a threshold obtained a priori for each type of motor. When the motor stops rotating, that is $\Omega_j(t) = 0$, it follows from (7) that $i_j = v_j/R$.

Now, we present two different identification schemes. The first scheme is a simple one and mainly pertains to the drones with limited computational capabilities. It includes a preflight motor testing to determine the minimum throttle setting necessary for the take off, which corresponds to the each motor thrust equal to Mg/n (wight/number of motors), then to determine the motor current with the propeller off for that throttle setting. The resulting current measurement sets the threshold. In some cases, this threshold value can be even provided by the manufacturer with the drone specifications. In the flight regime, the measured current is compared with the threshold for each motor. Dropping the current below the threshold indicates the failure of the corresponding motor.

The second scheme, which can be used by the drones with more computational power, is based on the generation of the reference current values according to the equations

$$\begin{aligned} i_j^{sr}(s) &= \frac{1}{Ls + R} v_j(t) \qquad (12)\\ i_j^{po}(s) &= \frac{J_m s + k_d}{(J_m s + k_d)(Ls + R) + k_b k_m} v_j(s)\,. \end{aligned}$$

which are driven by the same voltage input as the corresponding motors. The reference current $i_j^{nr}(t)$ corresponds to the non-rotating motor, and $i_j^{po}(t)$ corresponds to the motor current with the propeller off. Comparing the measurements of the actual motor current with $i_j^{nr}(t)$ and $i_j^{po}(t)$ we can determine if the corresponding motor is healthy, stopped rotating or the propeller has separated from the drone's body.

IV. Control Design

In this section we design a controller for the multi-rotor drone independent of the system's health. Since there are only four independent control inputs, we are able to track four independent commands. In this paper we choose altitude ($z_{com}(t)$), horizontal position ($x_{com}(t)$ and $y_{com}(t)$) and the yow angle or camera direction ($\psi_{com}(t)$) commands to track. The drones altitude is directly controlled by the total thrust generated by the rotors. For the horizontal position a cascaded control architecture is adopted, which is justified by the time scale separation between slow horizontal position, fast attitude and faster angular rate variables. Accordingly, there are three control loops. The outer loop generates roll and pitch commands ($\phi_{com}(t)$ and ($\theta_{com}(t)$) such that $x(t)$ and $y(t)$ respectively track the desired commands $x_{com(t)}$ and $y_{com}(t)$, the middle loop generates the angular rate commands ($p_{com}(t),\ q_{com}(t),\ r_{com}(t)$) such that the attitude variables $\phi(t)$, $\theta(t)$ and $\psi(t)$ respectively track the attitude commands $\phi_{com}(t)$, $\theta_{com}(t)$ and $\psi_{com}(t)$, and the inner loop generates the require torques $\tau_1(t)$, $\tau_2(t)$ and $\tau_3(t)$ such that the angular rates track the rate commands generated by the middle loop.

A. Altitude Control

We design the altitude control using the last equation in (3). The objective of this controller is to track the reference signal $z_{ref}(t)$, which is generated through a reference dynamics

$$\ddot{z}_{ref}(t) = -c_{z1}\dot{z}_{ref}(t) - c_{z2}\left[z_{ref}(t) - z_{com}(t)\right] \,, \tag{13}$$

where $z_{com}(t)$ represents the altitude command, $c_{z1} > 0$ and $c_{z2} > 0$ are the design parameters representing the performance characteristics, such as desired altitude change rate, overshoot, etc.

For this design we use dynamic inversion method and set

$$T(t) = \frac{1}{\cos\phi(t)\cos\theta(t)}\left[Mv_z(t) + k_t\dot{z}(t) + Mg\right] , \tag{14}$$

where $v_z(t)$ is the pseudo control, which is defined from the perspective of exponential stability and transient performance of the error signal $e_z(t) = z(t) - z_{ref}(t)$. To this end we set

$$v_z(t) = -c_{z1}\dot{z}(t) - c_{z2}\left[z(t) - z_{com}(t)\right] , \tag{15}$$

which results in the exponentially stable error dynamics

$$\ddot{e}_z(t) = -c_{z1}\dot{e}_z(t) - c_{z2}e_z(t)\,. \tag{16}$$

B. Horizontal Position Control

We assume that $-\pi/2 < \phi, \theta < /pi/2$, that is there are no flip over maneuvers. This assumption ensures that the functions $\sin\phi$ and $\sin\theta$ are one-to-one invertible. We apply the dynamic inversion technique to the firs two equations in (3) and set

$$\begin{aligned} \ddot{x}(t) &= v_x(t) \\ \ddot{y}(t) &= v_y(t)\,, \end{aligned} \tag{17}$$

where the pseudo controls $v_x(t)$ and $v_y(t)$ are defined from the perspective of tracking the reference models

$$\begin{aligned} \ddot{x}_{ref}(t) &= -c_{x1}\dot{x}_{ref}(t) - c_{x2}\left[x_{ref}(t) - x_{com}(t)\right] \\ \ddot{y}_{ref}(t) &= -c_{y1}\dot{y}_{ref}(t) - c_{y2}\left[y_{ref}(t) - y_{com}(t)\right] , \end{aligned} \tag{18}$$

where $x_{com}(t)$ and $y_{com}(t)$ are the horizontal position commands, $c_{x1} > 0,\ c_{x2} > 0,\ c_{y1} > 0,\ c_{y2} > 0$ are the design parameters representing the performance characteristics. Similar to the previous case, these pseudo controls are set to

$$\begin{aligned} v_x(t) &= -c_{x1}\dot{x}(t) - c_{x2}\left[x(t) - x_{com}(t)\right] \\ v_y(t) &= -c_{y1}\dot{y}(t) - c_{y2}\left[y(t) - y_{com}(t)\right] . \end{aligned} \tag{19}$$

The roll and pitch commands are obtained by solving the nonlinear system of equations

$$T(t)\left[\cos\phi(t)\sin\theta(t)\cos\psi(t)+\sin\phi(t)\sin\psi(t)\right]-k_t\dot{x}(t)=Mv_x(t) \tag{20}$$

$$T(t)\left[\cos\phi(t)\sin\theta(t)\sin\psi(t)-\sin\phi(t)\cos\psi(t)\right]-k_t\dot{y}(t)=Mv_y(t)\,. \tag{21}$$

Multiplying the first equation by $\cos\psi(t)$ and the second equation by $\sin\psi(t)$, and adding them results in

$$T(t)\cos\phi(t)\sin\theta(t)=\left[Mv_x(t)+k_t\dot{x}(t)\right]\cos\psi(t)+\left[Mv_y(t)+k_t\dot{y}(t)\right]\sin\psi(t)\,.$$

Similarly, multiplying the first equation by $\sin\psi(t)$ and the second equation by $\cos\psi(t)$, and subtracting them results in

$$T(t)\sin\phi(t)=\left[Mv_x(t)+k_t\dot{x}(t)\right]\sin\psi(t)-\left[Mv_y(t)+k_t\dot{y}(t)\right]\cos\psi(t)\,.$$

The attitude angle commands are easily obtained by inverting the *sin* function

$$\begin{aligned}
\phi_{com}(t) &= \sin^{-1}\left(\frac{1}{T(t)}\left\{\left[Mv_x(t)+k_t\dot{x}(t)\right]\sin\psi(t)-\left[Mv_y(t)+k_t\dot{y}(t)\right]\cos\psi(t)\right\}\right) \qquad (22)\\
\theta_{com}(t) &= \sin^{-1}\left(\frac{1}{T(t)\cos\phi(t)}\left\{\left[Mv_x(t)+k_t\dot{x}(t)\right]\cos\psi(t)+\left[Mv_y(t)+k_t\dot{y}(t)\right]\sin\psi(t)\right\}\right)\,.
\end{aligned}$$

which result in the following exponentially stable dynamics for error signals $e_x(t)=x(t)-x_{ref}(t)$ and $e_y(t)=y(t)-y_{ref}(t)$

$$\begin{aligned}
\ddot{e}_x(t) &= -c_{x1}\dot{e}_x(t)-c_{x2}e_x(t) \qquad (23)\\
\ddot{e}_y(t) &= -c_{y1}\dot{e}_y(t)-c_{y2}e_y(t)\,.
\end{aligned}$$

C. Roll Control

Here, the objective is to design a control torque $\tau_1(t)$ such that the roll angle $\phi(t)$ tracks the reference signal $\phi_{ref}(t)$ generated through the dynamics

$$\dot{\phi}_{ref}(t)=-c_\phi\left[\phi_{ref}(t)-\phi_{com}(t)\right]\,, \tag{24}$$

where $c_\phi>0$ is the time constant. For this purpose, we use time scale separation and dynamic inversion. First we derive an expression for the desired roll rate using the first equation in (6)

$$p_{com}(t)=c_\phi^p e_\phi(t)+c_\phi^i\int_0^t e_\phi(t)dt+\dot{\phi}_{ref}(t)-q(t)\sin\phi(t)\tan\theta(t)-r(t)\cos\phi(t)\tan\theta(t)\,, \tag{25}$$

where $e_\phi(t) = \phi(t) - \phi_{ref}(t)$ is the roll angle tracking error, $c_\phi^p > 0$ and $c_\phi^i > 0$ are the proportional and integral gains, then we derive the required control torque using the first equation in (5)

$$\tau_1(t) = -c_p e_p(t) + k_r p(t) + \frac{J_3 - J_2}{J_1} q(t) r(t) + J_{r3} q(t) \Omega(t) \,, \tag{26}$$

where $c_p > 0$ is the control gain, and $e_p(t) = p(t) - p_{com}(t)$ is the roll rate tracking error.

D. Pitch Control

Now, we design a control torque $\tau_2(t)$ such that the pitch angle $\theta(t)$ tracks the reference signal $\theta_{ref}(t)$, which satisfies the dynamics

$$\dot{\theta}_{ref}(t) = -c_\theta \left[\theta_{ref}(t) - \theta_{com}(t)\right] \,, \tag{27}$$

where $c_\theta > 0$ is a design parameter. Using time scale separation and dynamic inversion, we first derive the desired pitch rate command from the second equation in (6)

$$q_{com}(t) = c_\theta^p e_\theta(t) + c_\theta^i \int_0^t e_\theta(t) dt + \dot{\theta}_{ref}(t) - q(t) \sin\phi(t) \tan\theta(t) - r(t) \cos\phi(t) \tan\theta(t) \,, \tag{28}$$

where $e_\theta(t) = \theta(t) - \theta_{ref}(t)$ is the pitch angle tracking error, $c_\theta^p > 0$ and $c_\theta^i > 0$ are the proportional and integral gains, then we obtain the required control torque from the second equation in (5)

$$\tau_2(t) = -c_q e_q(t) + k_r q(t) + \frac{J_1 - J_3}{J_2} p(t) r(t) - J_{r3} p(t) \Omega(t) \,, \tag{29}$$

where $c_q > 0$ is a design parameter, and $e_q(t) = q(t) - q_{com}(t)$ is the pitch rate tracking error.

E. Yaw Control

Finally, we design a control torque $\tau_3(t)$ such that the yaw angle $\psi(t)$ tracks the reference signal $\psi_{ref}(t)$ generated through the dynamics

$$\dot{\psi}_{ref}(t) = -c_\psi \left[\psi_{ref}(t) - \psi_{com}(t)\right] \,, \tag{30}$$

where $c_\psi > 0$ is a design parameter. Similar to the previous case, we first derive an expression for the desired yaw rate using the third equation in (6)

$$r_{com}(t) = c_\psi^p e_\psi(t) + c_\psi^i \int_0^t e_\psi(t) dt + \dot{\psi}_{ref}(t) - q(t) \sin\phi(t) \sec\theta(t) - r(t) \cos\phi(t) \sec\theta(t) \,, \tag{31}$$

where $e_\psi(t) = \psi(t) - \psi_{ref}(t)$ is the yaw angle tracking error, $c^p_\psi > 0$ and $c^i_\psi > 0$ are the proportional and integral gains. The required control torque is designed using the third equation in (5)

$$\tau_3(t) = -(c_r - k_r)e_r(t) + \frac{J_1 - J_3}{J_3} p(t)q(t)\,, \tag{32}$$

where $c_r > 0$ is the control gain, and $e_r(t) = r(t) - r_{com}(t)$ is the yaw rate tracking error.

V. Decision Making

For a nominal multi-rotor drone, the total thrust and three torques are related to the individual motor thrusts trough a control allocation matrix $B \in \mathbb{R}^{4\times n}$, which is defined as

$$\underbrace{\begin{bmatrix} T \\ \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix}}_{\boldsymbol{u}} = \underbrace{\begin{bmatrix} 1 & 1 & \dots & 1 \\ b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ b_{31} & b_{32} & \dots & b_{3n} \end{bmatrix}}_{B} \underbrace{\begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix}}_{\boldsymbol{f}}\,, \tag{33}$$

where the coefficients $b_{ij},\ i = 1, 2,\ j = 1, \dots, n$ are easily derived from the geometry of the drone, and $b_{3j} = (-1)^j d,\ j = 1, \dots, n$, where d is the ratio between the drag and the thrust coefficients of the propeller blade. Therefore, the individual thrust settings can be found by solving the equation (33) for $\boldsymbol{f}$.

When the j-th motor failure is identified, f_j is set to zero and removed from the right hand side of (33), and the j-th column of matrix B is deleted. Let the number of failed motor be m $(0 \le m \le n - 5)$ and $\rho = n - m$ $(\rho \ge 5)$. Then matrix B is $(4 \times \rho)$ - dimensional. That is, the equation (33) has more than one solution, and some optimization routine can be applied. Here, we use simple Moore-Penrose inverse to obtain the solution

$$\boldsymbol{f} = B^\top (BB^\top)^{-1} \boldsymbol{u}\,. \tag{34}$$

When $\rho = 4$, then B is an invertible (4×4) matrix, hence a unique solution exists

$$\boldsymbol{f} = B^{-1} \boldsymbol{u}\,. \tag{35}$$

Therefore, in case of $\rho \geq 4$ the decision making logic does not alter the external position and/or attitude commands.

When $\rho < 4$, controlling for independent states is not achievable anymore, and the safe landing mode is activated. In this mode, our algorithm gives up the yaw control, and commands a safe landing of the drone. It is worth to notice that the horizontal position control is still achievable, but it requires roll and pitch angle modulation, which is not a safe maneuver when the drone is close to the ground. For this reason, the safe landing algorithm is executed in two steps. First step is to maintain a safe altitude while moving the drone in the hovering position over the safe landing spot. This altitude depends on the environment such as buildings, people, natural obstacles etc., or on the mission such as carrying a tethered load etc., and is set by the pilot or autopilot. The second step is to set $z_{com}(t) = 0$ (landing), $\phi_{com}(t) = 0$ and $\theta_{com}(t) = 0$ (horizontal attitude), and compute $T(t)$, $\tau_1(t)$ and $\tau_2(t)$ according to equation (14), (26) and (29). The individual motor thrusts are computed according to (35) by first deleting the last row of the matrix B and removing τ_3 from $\boldsymbol{u}$.

We summarize the decision making algorithm as follows

- If $m = 0$, use (34) with nominal B to compute individual motor thrusts $f_j,\ j = 1, \ldots, n$.
- If $m > 0$, set $f_{j_1} = 0, \ldots, f_{j_m} = 0$ and delete the $j_1, \ldots, j_m$ columns of matrix B, where the indexes $j_1, \ldots, j_m$ of failed rotors are provided by the identification block.
- If $\rho > 4$, use (34) to compute individual motor thrusts $f_j,\ j \in \{1, \ldots, n\} - \{j_1, \ldots, j_m\}$
- If $\rho = 4$, use (35) to compute individual motor thrusts $f_j,\ j \in \{1, \ldots, n\} - \{j_1, \ldots, j_m\}$
- If $\rho < 4$, activate the safe landing mode.

VI. Simulation Results

For the demonstration of the benefits of the proposed algorithm we use the quadrotor Armattan CF-226 presented in Figure 1 in our numeriacal simulation study. The quadrotor technical specifications are: Flight Controller - 3DR Pixhawk (Firmware 3.2.1), 32-bit STM32F427 Cortex M4 core with FPU, 168 MHz/256 KB RAM/2 MB Flash, 32 bit STM32F103 failsafe co-processor; Sensors - ST Micro L3GD20 3-axis 16-bit gyroscope, ST Micro LSM303D 3-axis 14-bit accelerometer /

magnetometer, Invensense MPU 6000 3-axis accelerometer/gyroscope, MEAS MS5611 barometer; Motors - T-Motor MN1806; ESCs - F-12A Fire Red series SimonK; Battery - Turnigy 1800mAh 3S 30C Lipo; Propellers - APC 5.5x4.5; Total mass: 0.516 kg (with battery). The frame is in "x" - configuration with arm lengths of 0.125 m.

Figure 2 displays motor current readings for the quadrotor with two propellers off. It can be observed that the corresponding readings are much less than the healthy motors current reading as predicted by the identification algorithm. From the readings it can be concluded that the identification threshold can be set 1.4 amps.

In the simulation experiment, the drone is commanded to climb to an altitude of 6 meters from the zero initial position. At $t = 20sec$ one of the motors fails. The drone is commanded to stabilize at the altitude of 6 meters for 5 sec, then to land. At time $t = 20$ sec the control is switched to the safe mode. Figure 3 displays the algorithm performance for the position control. Obviously the control objective has bee met.

Figure 4 displays the orientation angles of the drone. It can be seen that the yaw angle is growing starting at $t = 20sec$, when the safe mode controller has taken over giving up the yaw control. This indicates that the drone goes into a spin, the rate of which is associated with the imbalance between the right and left rotating motors.

VII. Conclusion

We have presented an algorithm for detecting/identifying the failures of the multi-rotor drones, the reconfigurable controller capable of continuing or aborting the mission based on the switching logic of the decision making algorithm, assuming that the parameters and the dynamics of the drone are known. The performance of the presented algorithm is guaranteed as long as the drone retains controllability in vertical direction and the stabilizability in the horizontal plane. The benefits of the proposed architecture have been demonstrated in simulations. Future research will include extension of the proposed algorithms to the case of uncertain drones as well as in flight experiments.

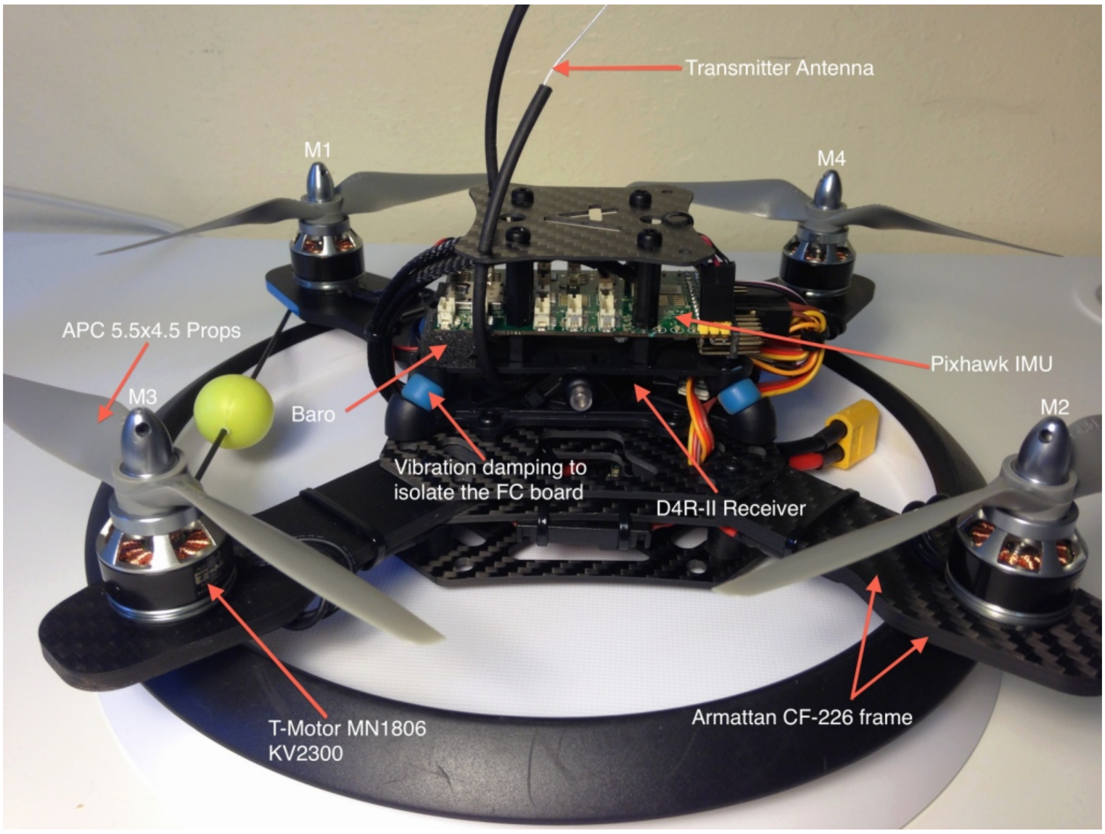

Fig. 1 Quadrotor Armattan CF-226.

References

[1] H. Alwi, M. T. Hamayun, and C. Edwards. An Integral Sliding Mode Fault Tolerant Control Scheme for an Octorotor Using Fixed Control Allocation. *In Proceedings of the 13th International Workshop*

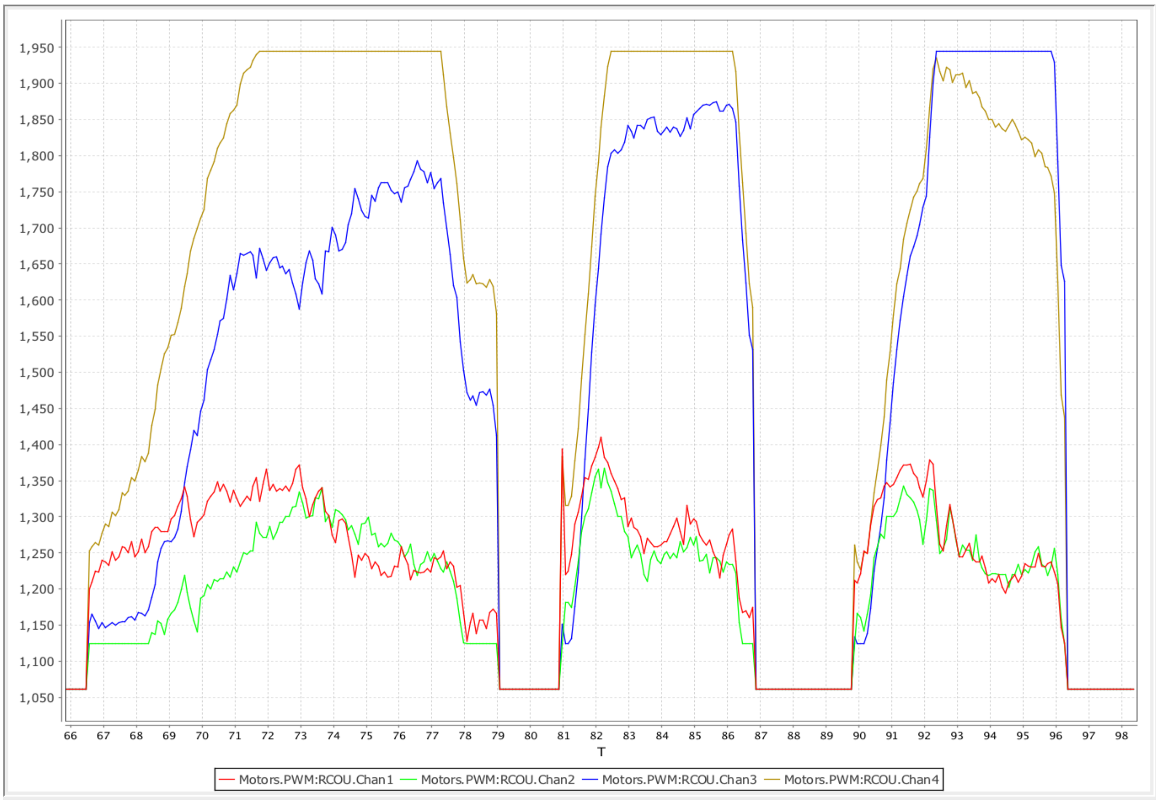

Fig. 2 Motor current readings.

on Variable Structure Systems, DOI: 10.1109/VSS.2014.6881156, pages 1 – 6, 2014.

[2] A. S. Candido, R. K. H. Galvao, and T.Yoneyama. Actuator Fault Diagnosis and Control of a Quadrotor. *In Proceedings of the 12th IEEE International Conference on Industrial Informatics, DOI: 10.1109/INDIN.2014.6945530*, pages 310 – 315, 2014.

[3] Z. Cen, H. Noura, and Y. Al Younes. Systematic Fault Tolerant Control Based on Adaptive Thau Observer Estimation for Quadrotor UAVs. *International Journal of Applied Mathematics and Computer Science, DOI: 10.1515/amcs-2015-0012*, 25(1):159?174, 2015.

[4] A. Chamseddine, Y. Zhang, C.-A. Rabbath, J. Apkarian, and C. Fulford. Model Reference Adaptive Fault Tolerant Control of a Quadrotor UAV. *In Proceedings of the AIAA Infotech@Aerospace Conference*, 2011.

[5] G. R. F. Colunga, H. Aguilar-Sierra, R. Lozano, and S. Salazar. Fault Estimation and Control for a Quad-Rotor MAV Using a Polynomial Observer. Part I: Fault Detection. *In the book: M. A. Armada and A. Sanfeliu and M. Ferre, Advances in Robotics, Springer International Publishing*, 1:151–170, 2014.

[6] C. de Crousaz, F. Farshidian, M. Neunert, and J. Buchli. Unified Motion Control for Dynamic Quadro-

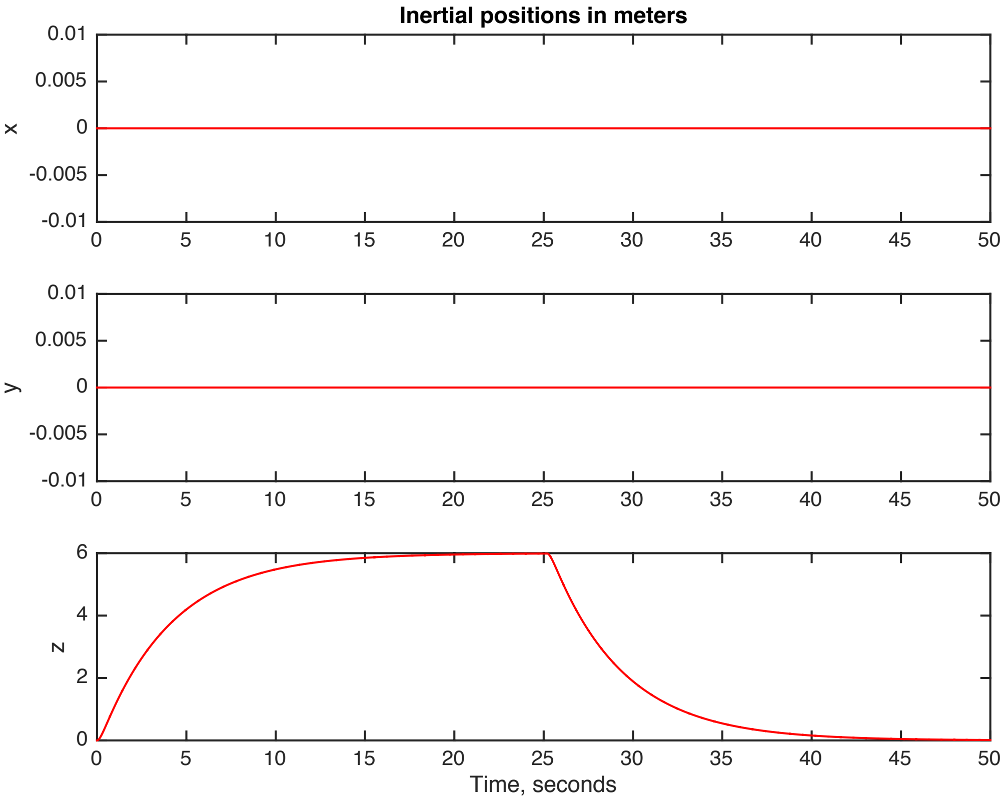

Fig. 3 The drone inertial position in meters.

tor Maneuvers Demonstrated on Slung Load and Rotor Failure Tasks. *In Proceedings of the IEEE International Conference on Robotics and Automation, Hong Kong, China*, 2015.

[7] B. Etkin and L. D. Reid. *Dynamics of Flight: Stability and Control.* New Yourk: Wiley, 1996.

[8] G. P. Falconi, C. D. Heise, and F. Holzapfel. Fault-Tolerant Position Tracking of a Hexacopter Using an Extended State Observer. *In Proceedings of the 6th International Conference on Automation, Robotics and Applications (ICARA), DOI: 10.1109/ICARA.2015.7081207*, pages 550 – 556, 2015.

[9] M. Frangenberg, J. Stephan, and W. Fichter. Fast Actuator Fault Detection and Reconfiguration for Multicopters (AIAA 2015-1766). *In Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2015.

[10] A. Freddi, S. Longhi, and A. Monteriu. Actuator Fault Detection System for a Mini-Quadrotor. *In Proceedings of the IEEE International Symposium on Industrial Electronics, DOI: 10.1109/ISIE.2010.5637750*, pages 151–170, 2010.

[11] A. Freddi, S. Longhi, A. Monteriu, and M. Prist. Actuator Fault Detection and Isolation System for a Hexacopter. *In Proceedings of the 10th IEEE/ASME International Conference on Mechatronic and*

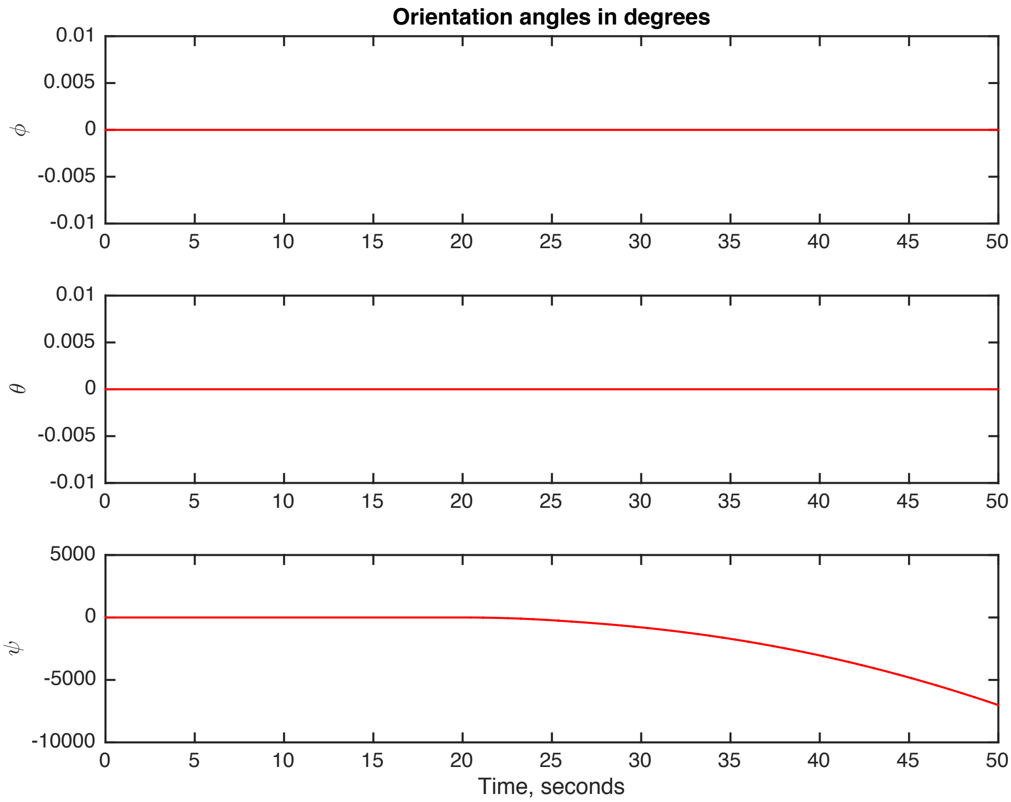

Fig. 4 The drone orientation angles in degrees.

Embedded Systems and Applications, DOI: 10.1109/MESA.2014.6935563, pages 1–6, 2014.

[12] X. Gong, C. Peng, Y. Tian, Y. Bai, C. Zhao, Q. Gao, and Z. Xu. Reliable Attitude Stability Control of Quadrotor Based on Fault Tolerant Approach. *In Proceedings of the International Conference on Mechatronics and Automation, DOI: 10.1109/ICMA.2012.6283389*, pages 1015 – 1020, 2012.

[13] A. Lanzon, A. Freddi, and S. Longhi. Flight Control of a Quadrotor Vehicle Subsequent to a Rotor Failure. *Journal of Guidance, Control, and Dynamics*, 37:580–591, 2014.

[14] A. Marks, J. F. Whidborne, and I. Yamamoto. Control Allocation for Fault Tolerant Control of a VTOL Octorotor. *In Proceedings of the UKACC International Conference on Control, Cardiff, UK*, 2012.

[15] A. Milhim, Y. Zhang, and C.-A. Rabbath. Gain Scheduling Based PID Controller for Fault Tolerant Control of Quad-Rotor UAV. *In Proceedings of the AIAA Infotech@Aerospace Conference*, 2010.

[16] M. W. Mueller and R. D'Andrea. Stability and control of a quadrocopter despite the complete loss of one, two, or three propellers. *In Proceedings of the IEEE International Conference on Robotics and Automation, Hong Kong, China*, 2014.

[17] M. Ranjbaran and K. Khorasani. Generalized Fault Recovery of an Under-Actuated Quadrotor Aerial Vehicle. *In Proceedings of the American Control Conference, Montre?al, CA*, 2012.

[18] R. Padhi S. N. Tiwari. DR-SNAC Aided Dynamic Inversion Controller for Robust Trajectory Tracking of a Micro-Quadrotor. *In Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2013.

[19] I. Sadeghzadeh and Y. Zhang. A Review on Fault-Tolerant Control for Unmanned Aerial Vehicles (UAVs). *In Proceedings of the AIAA Infotech@Aerospace Conference*, 2011.

[20] Q. Shen, D. Wang, S. Zhu, and E. K. Poh. Fault-Tolerant Attitude Tracking Control for a Quadrotor Aircraft. *In Proceedings of the 53rd IEEE Conference on Decision and Control, DOI: 10.1109/CDC.2014.7040349*, pages 6129 – 6134, 2014.

[21] B. Yu, Y. Zhang, Y. Yit, Y. Qu, and P. Lu. Fault Detection for Partial Loss of Effectiveness Faults of Actuators in a Quadrotor Unmanned Helicopter. *In Proceedings of the 11th World Congress on Intelligent Control and Automation Shenyang, China*, 2014.