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 landing mode, which gives up the yaw control, commands a safe

[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

landing spot and lands the drone 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 [22]. 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, to rate them based on the predefined fault-tolerant boundaries and to 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 presented for the 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 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 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 remaining healthy motors deliver sufficient thrust 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 ϕ, η, ψ 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 $\boldsymbol{D}(t) = -k_t\boldsymbol{v}(t)$. With this assumptions the translational dynamics can be written as

$$\begin{aligned} \dot{\boldsymbol{r}}(t) &= \boldsymbol{v}(t) \\ M\dot{\boldsymbol{v}}(t) &= T(t)L^{(3)}_{B/I}(t) - k_t\boldsymbol{v}(t) + M\boldsymbol{g}\,, \end{aligned} \tag{3}$$

where $L^{(3)}_{B/I}$ denotes the third column of matrix $L_{B/I}$.

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

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

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. 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 as

$$J\dot{\boldsymbol{\omega}}(t) = -\boldsymbol{\omega}(t)\times J\boldsymbol{\omega}(t) + J_{r3}\Omega(t)\bar{\boldsymbol{\omega}}(t) + \boldsymbol{\tau}(t) - k_r\boldsymbol{\omega}(t)\,, \tag{5}$$

where we denote $\Omega = \sum_{i=1}^{n} \beta_i \Omega_i$ ($\beta_i = 1$ for motors rotating counterclockwise and $\beta_i = -1$ otherwise) and $\bar{\boldsymbol{\omega}} = [-q \quad p \quad 0]^\top$. The angular rate $\boldsymbol{\omega}$ is related to the Euler angles $\boldsymbol{E} = [\phi \quad \theta \quad \psi]^\top$ by means of the kinematic equations

$$\dot{\boldsymbol{E}}(t) = H(t)\boldsymbol{\omega}(t)\,, \tag{6}$$

where we denote

$$H = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix}.$$

III. Identification

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

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

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 the 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.

Alternatively, when the motor stops rotating, that is $\Omega_j(t) = 0$, it follows from (7) that $i_j(t) = v_j(t)/R$, which is the maximum current flowing through the motor for a given input voltage $v_J(t)$.

The above considerations make the bases for a simple but conservative failure identification algorithm, which mainly pertains to the drones with limited computational capabilities. It requires the measurements of the motor input voltage and output current for each motor. The ratio $\rho_j(t) = \frac{i_j(t)}{v_j(t)}$ is always less than $1/R$. Therefore, the inequality $\rho_j(t) > 1/R - \delta_1$, where the threshold δ_1 can be experimentally determined based on the sensor characteristics, implies that the j-th motor stopped operating. On the other hand, when the external load τ_j^l is zero, it follows from the equation

$$i_j(s) = \frac{J_{r3}s + k_d}{(J_{r3}s + k_d)(Ls + R) + k_b k_m} v_j(s) \tag{12}$$

that

$$|i_j(t)| \leq \frac{k_d}{Rk_d + k_b k_m} \max_{t \in [t_1, t_2]} [v_j(t)] \tag{13}$$

for any time interval $t_1 \leq t \leq t_2$, where $\frac{k_d}{Rk_d + k_b k_m}$ is the H_∞ norm of the transfer function $\frac{J_{r3}s + k_d}{(J_{r3}s + k_d)(Ls + R) + k_b k_m}$. Therefore, the inequality

$$i_j(s) \leq \frac{k_d}{Rk_d + k_b k_m} v_j(t) + \delta_2,$$

where the threshold δ_2 can be experimentally determined for each type of motor, implies that the propeller of the j-th motor failed.

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

$$
\begin{aligned}
i_j^{nr}(t) &= \frac{1}{Ls+R} v_j(t) \\
i_j^{po}(t) &= \frac{J_m s + k_d}{(J_m s + k_d)(Ls+R) + k_b k_m} v_j(t)\,.
\end{aligned}
\tag{14}
$$

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 controllers for the nominal multi-rotor drone using the total thrust $T(t)$ and three components of the torque $\boldsymbol{\tau}(t)$ generated by the motors as control input. Since there are only four independent inputs, we are able to track four independent external commands for the nominal drone. These can be the drone's 1) inertial position and yaw angle or camera direction commands (Position control mode), 2) inertial velocity and yaw angle commands (Velocity control mode), or 3) altitude and orientation angles commands (Attitude control mode). Here, we adopt a cascaded control architecture, which is justified by the time scale separation between slow position, fast attitude and faster angular rate variables.

A. Position Control

The objective of this controller is to track the reference signal $\boldsymbol{r}_{ref}(t)$, which is generated through a reference dynamics

$$
\dot{\boldsymbol{r}}_{ref}(t) = -c_r \left[\boldsymbol{r}_{ref}(t) - \boldsymbol{r}_{com}(t)\right]\,, \tag{15}
$$

driven by the external position command $\boldsymbol{r}_{com}(t)$, where $c_r > 0$ is a design parameter. The control law is defined according to equation

$$
\boldsymbol{v}_{com}(t) = -c_r \boldsymbol{e}_r(t) + \dot{\boldsymbol{r}}_{ref}(t)\,, \tag{16}
$$

where $\boldsymbol{e}_r(t) = \boldsymbol{r}(t) - \boldsymbol{r}_{ref}(t)$ is the position tracking error. Substituting the control law in the translational dynamics results in the exponentially stable error dynamics

$$\dot{\boldsymbol{e}}_r(t) = -c_r \boldsymbol{e}_r(t)\,. \tag{17}$$

The obtained control signal $\boldsymbol{v}_{com}(t)$ is used in the velocity control scheme to obtain the required total thrust and the roll and pitch attitude commands.

B. Velocity Control

The objective of this controller is to track the reference signal $\boldsymbol{v}_{ref}(t)$, which is generated through a reference dynamics

$$\dot{\boldsymbol{v}}_{ref}(t) = -c_v \left[\boldsymbol{v}_{ref}(t) - \boldsymbol{v}_{com}(t)\right]\,, \tag{18}$$

where $c_v > 0$ is a design parameter, $\boldsymbol{v}_{com}(t)$ is the velocity command obtained for the position control or is an independent external command depending on the mission. The control law is defined according to equation

$$T(t)L^{(3)}_{B/I}(t) = k_t \boldsymbol{v}(t) - M\boldsymbol{g} - Mc_v \boldsymbol{e}_v(t) + M\dot{\boldsymbol{v}}_{ref}(t)\,, \tag{19}$$

where $\boldsymbol{e}_v(t) = \boldsymbol{v}(t) - \boldsymbol{v}_{ref}(t)$ is the velocity tracking error. Substituting the control law in the translational dynamics results in the exponentially stable error dynamics

$$\dot{\boldsymbol{e}}_v(t) = -c_v \boldsymbol{e}_v(t)\,. \tag{20}$$

The required total thrust and orientation angle commands are obtained from (19) assuming that $-\pi/2 < \phi, \theta < \pi/2$, that is there are no flip-over maneuvers. This assumption ensures that the functions $\cos\phi$ and $\cos\theta$ are nonzero, and the $\sin\phi$ and $\sin\theta$ are one-to-one invertible. It follows from the equation (19) written component-wise

$$\begin{aligned}
T(t)\left[\cos\phi(t)\sin\theta(t)\cos\psi(t) + \sin\phi(t)\sin\psi(t)\right] &= k_t v_x(t) - Mc_v e_{v_x}(t) + Mv_{x_{ref}}(t) \triangleq s_x(t)\\
T(t)\left[\cos\phi(t)\sin\theta(t)\sin\psi(t) - \sin\phi(t)\cos\psi(t)\right] &= k_t v_y(t) - Mc_v e_{v_y}(t) + Mv_{y_{ref}}(t) \triangleq s_y(t)\\
T(t)\cos\phi(t)\cos\theta(t) &= k_t v_z(t) + Mg - Mce_{v_z}(t) + Mv_{z_{ref}}(t) \triangleq s_z(t)\,.
\end{aligned} \tag{21}$$

that the total thrust is readily obtained from the third equation as

$$T(t) = \frac{s_z(t)}{\cos\phi(t)\cos\theta(t)} \,. \tag{22}$$

which basically controls the drone's altitude or vertical speed. Next, we multiply the first equation by $\cos\psi(t)$, the second equation by $\sin\psi(t)$, and adding them obtain

$$T(t)\cos\phi(t)\sin\theta(t) = s_x(t)\cos\psi(t) + s_y(t)\sin\psi(t)\,, \tag{23}$$

Similarly, we multiply the first equation by $\sin\psi(t)$, the second equation by $\cos\psi(t)$, and subtracting them obtain

$$T(t)\sin\phi(t) = s_x(t)\sin\psi(t) - s_y(t)\cos\psi(t)\,. \tag{24}$$

The attitude angle commands are easily obtained from (24) and (23) by inverting the *sin* function

$$\begin{aligned} \phi_{com}(t) &= \sin^{-1}\left(\frac{s_x(t)\sin\psi(t) - s_y(t)\cos\psi(t)}{T(t)}\right) \\ \theta_{com}(t) &= \sin^{-1}\left(\frac{s_x(t)\cos\psi(t) + s_y(t)\sin\psi(t)}{T(t)\cos\phi(t)}\right)\,. \end{aligned} \tag{25}$$

C. Attitude Control

Now, we derive the control torque for the rotational dynamics (4) and (6) such that the Euler angle $\boldsymbol{E}(t)$ tracks the reference signal $\boldsymbol{E}_{ref}(t)$ generated through the dynamics

$$\dot{\boldsymbol{E}}_{ref}(t) = -c_E\left[\boldsymbol{E}_{ref}(t) - \boldsymbol{E}_{com}(t)\right]\,, \tag{26}$$

where $c_\phi > 0$ is a design constant and $\boldsymbol{E}_{com}(t) = [\phi_{com}(t) \;\; \theta_{com}(t) \;\; \psi_{com}(t)]^\top$ is the Euler angles command, which can be as the command obtained from the perspective of the position or velocity control as well as an independent command depending on the mission. Using time scale separation and dynamic inversion techniques, we first derive an expression for the desired angular rate from the equation (6)

$$\boldsymbol{\omega}_{com}(t) = H^{-1}(t)[-c_\omega \boldsymbol{e}_E(t) + \dot{\boldsymbol{E}}_{ref}(t)]\,, \tag{27}$$

where $\boldsymbol{e}_E(t) = \boldsymbol{E}(t) - \boldsymbol{E}_{ref}(t)$ is the attitude angles tracking error, $c_E > 0$ is the control gain, and $H^{-1}(t)$ is the inverse of the matrix $H(t)$ given by

$$H^{-1}(t) = \begin{bmatrix} 1 & 0 & -\sin\theta(t) \\ 0 & \cos\phi(t) & \sin\phi(t)\cos\theta(t) \\ 0 & -\sin\phi(t) & \cos\phi(t)\cos\theta(t) \end{bmatrix} .$$

then we derive the required control torque using equation (5)

$$\boldsymbol{\tau}(t) = \boldsymbol{\omega}(t) \times J\boldsymbol{\omega}(t) - J_{r3}\Omega(t)\bar{\boldsymbol{\omega}}(t) + k_r\boldsymbol{\omega}(t) + J[-c_\omega \boldsymbol{e}_\omega(t) + \dot{\boldsymbol{\omega}}_{ref}(t)] , \tag{28}$$

where $c_\omega > 0$ is the control gain, $\boldsymbol{e}_\omega(t) = \boldsymbol{\omega}(t) - \boldsymbol{\omega}_{ref}(t)$ is the angular rate tracking error, which satisfies the exponentially stable dynamics

$$\dot{\boldsymbol{e}}_\omega(t) = -c_\omega \boldsymbol{e}_\omega(t) , \tag{29}$$

and the signal $\boldsymbol{\omega}_{ref}(t)$ is generated through the reference dynamics

$$\dot{\boldsymbol{\omega}}_{ref}(t) = -c_\omega \left[\boldsymbol{\omega}_{ref}(t) - \boldsymbol{\omega}_{com}(t)\right] . \tag{30}$$

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

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

where the coefficients $b_{ij},\ i = 1, 2,\ j = 1, \dots, n$ are easily derived from the geometry of the drone, and $b_{3j} = \pm d,\ j = 1, \dots, n$, where d is the ratio between the drag and the thrust coefficients of the propeller blade, and the sign depends on the direction of rotation. Therefore, the individual thrust settings can be found by solving the equation (31) for $\boldsymbol{f}$ subject to motor physical constraints

$$0 \le f_{\min} \le f_j \le f_{\max}, \quad j = 1, \dots, n , \tag{32}$$

where $f_{\min}$ and $f_{\max}$ are minimum and maximum allowable motor thrusts. In general, B is invertible for the quadrotor ($n = 4$), resulting in the unique solution of the equation (31)

$$\boldsymbol{f} = B^{-1}\boldsymbol{u}\,. \tag{33}$$

In this case, violation of the motor constraints implies that the external command cannot be implemented. For $n > 4$ the equation (31) is under determined, implying that there can be more than one solution. Therefore a control allocation method can be used to determine the individual motor thrust.

Here, we use Redistributed Pseudo Inverse (RPI) method, the details of which can be found in [14] and in references therein. The method iteratively calculates the Moore-Penrose pseudo inverse solution of (31) by setting the saturated motor thrust to its limit and removing it from subsequent pseudo inverse solutions. The algorithm initializes a vector $\boldsymbol{a} \in \mathbb{R}^n$ to zero, stores the original matrix B in B_0, and computes the solution of (31) according to

$$\boldsymbol{f} = -\boldsymbol{a} + B^\top (BB^\top)^{-1}[\boldsymbol{u} + B_0\boldsymbol{a}]\,. \tag{34}$$

If for any $j = 1, \ldots, n$ $f_{\min} \geq f_j$ or $f_{\max} \leq f_j$, then the algorithm sets $a_j = -f_{\min}$ or $a_j = -f_{\max}$ respectively, zeros out the j-th column of B and computes $\boldsymbol{f}$ using the modified $\boldsymbol{a}$ and B in (34).

We modify the RPI algorithm to incorporate our motor failure identification scheme. When the j-th motor failure is identified, f_j is set to zero and removed from the right hand side of (31), and the j-th column of matrix B is removed before RPI is applied, thus decreasing the size of the control allocation problem.

The resulting iterative process may output no feasible solution in the presence of the motor failures. An obvious case is the number of healthy motors reaching three, resulting in the overdetermined system (31). This is the case for quadrotors with a single motor failure. The loss of controllability of the multi-rotor drones may occur even with more than three healthy motors, when the motor failures result in the force imbalance in one or more directions, which is expressed in RPI finding no feasible solution for the control allocation problem (31) with constraints (32).

When RPI algorithm gives no solution in the presence of the motor failures, 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 may still be 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 constant 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 $z_{com}(t) = 0$ (if the reliable altitude measurement is available) or $v_{z_{com}}(t) = -v_l$, where v_l denotes a safe landing speed, $\phi_{com}(t) = 0$ and $\theta_{com}(t) = 0$ (horizontal attitude), then to compute the total thrust $T(t)$ according to equation (22), and first two components of $\boldsymbol{\tau}(t)$ according to equations (27) and (28). Before computing the individual motor thrusts using RPI, the last row of the matrix B and τ_3 from $\boldsymbol{u}$ are removed. In the case of quadrotors, the resulting equation (31) is square with an invertible B matrix, and the solution is given by (33).

We summarize the decision making algorithm as follows. Let the number of failed motor be m $(0 \leq m \leq n-3)$.

- $m = 0$: if $n = 4$ apply (33), if $n > 4$ apply RTI for (31) with constraints (32).
- $m > 0$: 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 motors are provided by the identification block.
- If $n = 4$, activate the safe landing mode.
- If $n > 4$, apply RTI for (31) with modified B matrix and constraints (32) to compute individual motor thrusts $f_j,\ j \in \{1, \ldots, n\} - \{j_1, \ldots, j_m\}$. If no solution exist, 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 total mass is 0.516 kg (with battery), the frame is in "x" - configuration with arm lengths of 0.125 m.

Figure 2 displays motor current readings in the test with the propeller separating from the

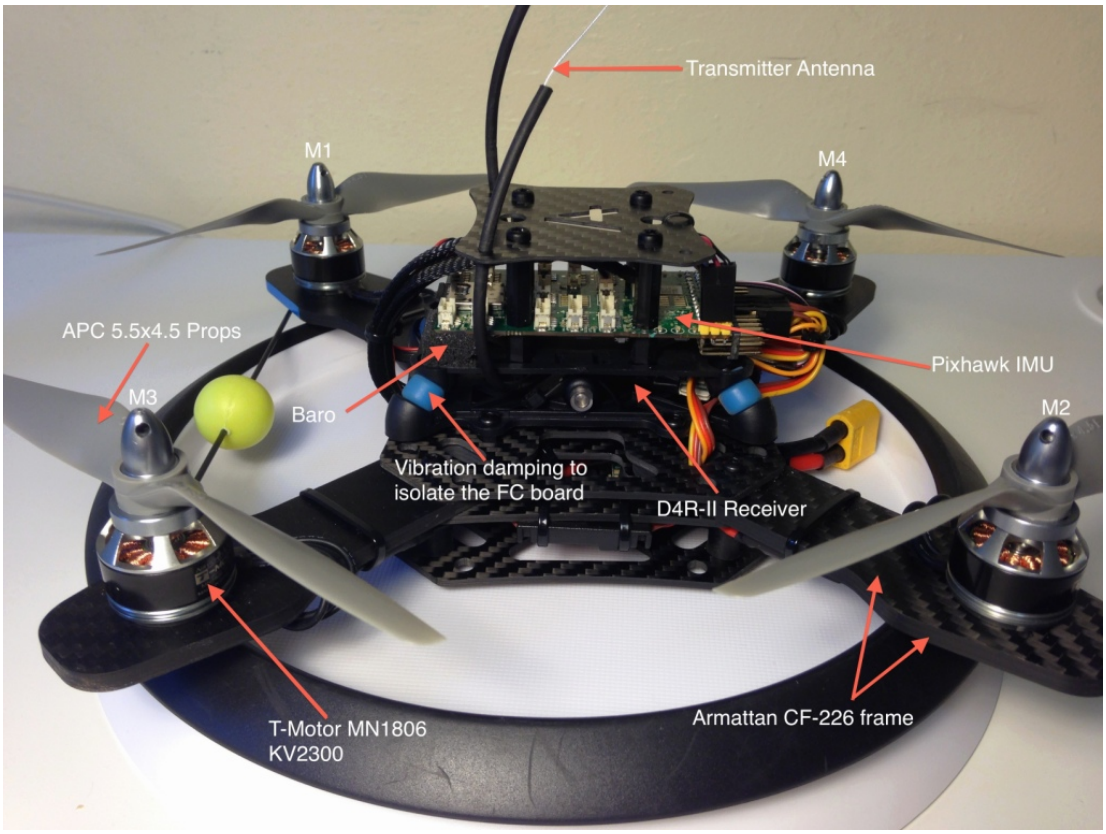


Fig. 1 Quadrotor Armattan CF-226.

motor. It can be observed that the current sharply drops at 44.01 sec when the propeller comes off as predicted by the identification algorithm. From the readings it can be concluded that the

identification threshold can be set at 0.75 amps.

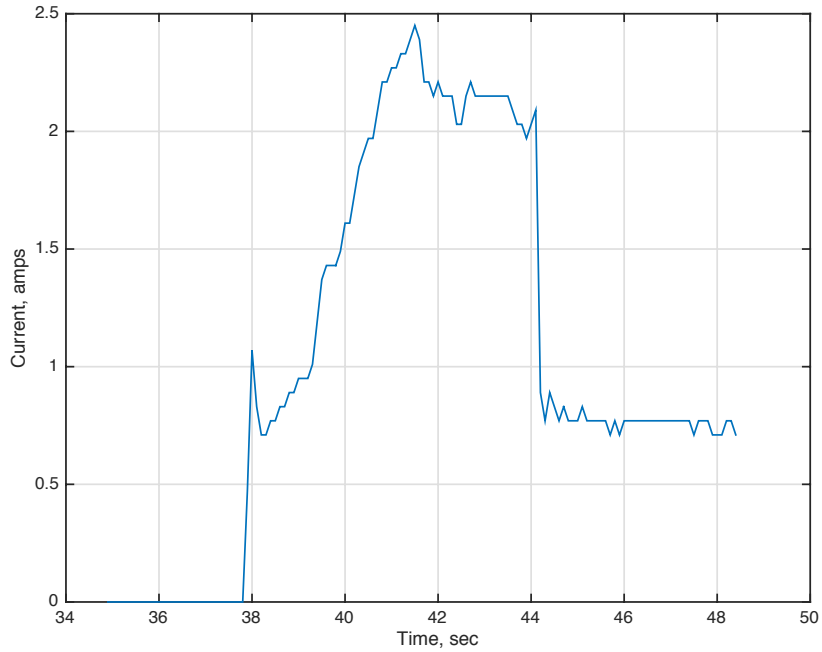
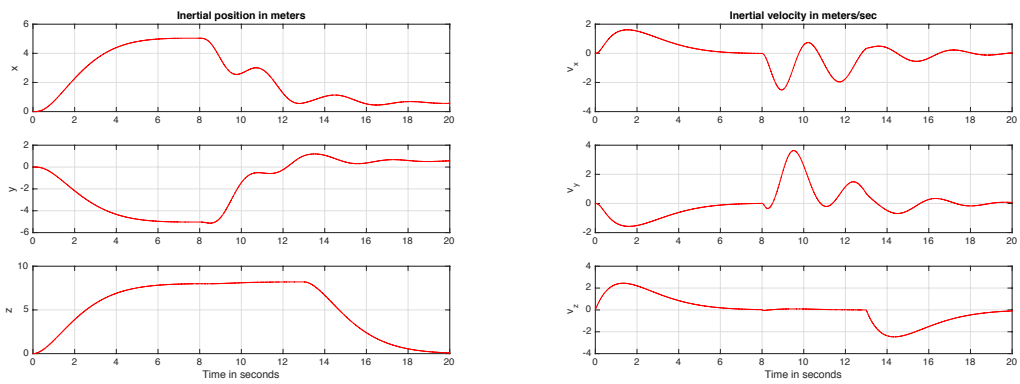


Fig. 2 Motor current readings.

In the simulation experiment, the drone is commanded to move to the position $x = 5$, $y = -5$, $z = 8$ meters from the zero initial position. At $t = 8sec$ the motor number 2 fails. 40 milliseconds time delay is introduced to simulate the failure detection and identification time, after which the safe landing mode is activated. The drone is commanded to maintain the altitude and



(a) The drone inertial position time history

(b) The drone inertial velocity time history

Fig. 3 The controller performance in translational dynamics with motor failure at 8 sec

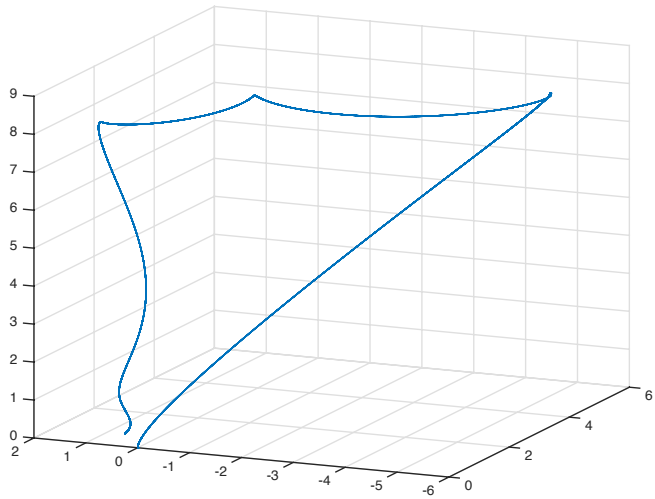
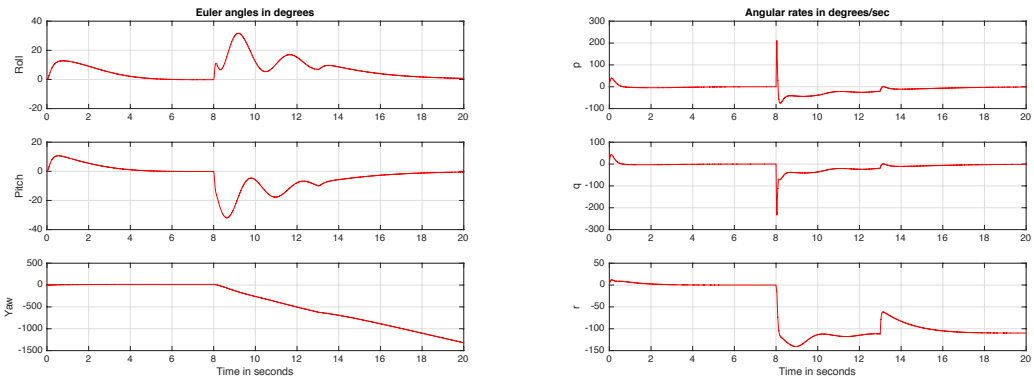


Fig. 4 The drone's inertial trajectory.



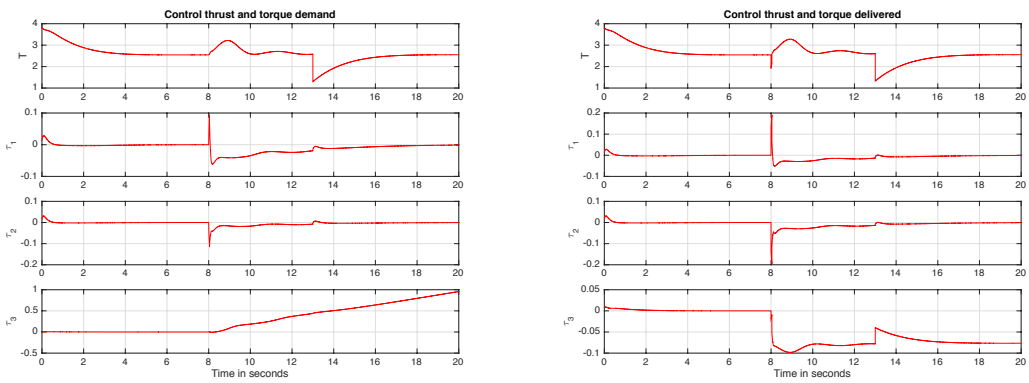
(a) The drone Euler angle time history

(b) The drone abgular rate time history

Fig. 5 The controller performance in rotational dynamics with motor failure at 8 sec

move to the $x = 0$, $y = 0$ position in 5 sec, then to land. Figure 3(a) displays the inertial position time histories. It can be observed that the presented safe landing algorithm performance is satisfactory with a little offset in horizontal position when the drone is maintaining the altitude while looking for a safe spot to land. After the drone lands (the altitude reaches to zero), its inertial velocities converge to zero as it can be seen from Figure 3(b). The drone's trajectory is presented in Figure 4.

Figure 5(a) displays the attitude angles time histories of the drone. It can be seen that the yaw angle is growing starting at $t = 8sec$, when the safe mode controller has taken over giving up



(a) The control signal time history

(b) The actual thrust and torque time history

Fig. 6 The required total thrust and torque vs the actually delivered total thrust and torque with motor failure at 8 **sec**

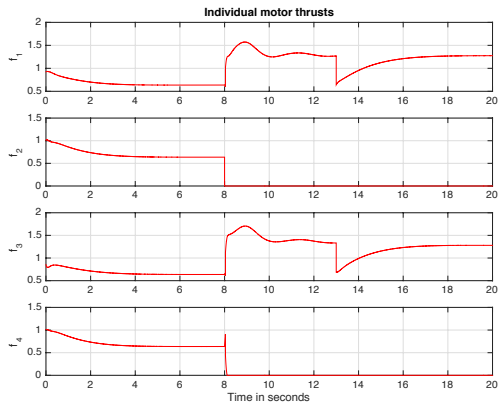


Fig. 7 The individual motor thrusts.

the yaw control. This indicates that the drone goes into a spin, the rate of which converges to a constant value as it can be observed from Figure 5(b). The roll and pitch angles and angular rates converge to zero, thus providing a safe landing configuration.

Figure 6(a) displays the total thrust and torque demanded by the controller. It can be observed that the controller demands a continuously growing positive yaw torque after the failure occurrence, however the decision making algorithm activates the safe landing mode which gives up the yaw control. This creates a torque imbalance in yaw direction, and a negative yaw torque is delivered to the drone forcing it to spin in the negative direction as it can be observed from Figure 6(b). Figure 7 displays the individual motor thrust curves. It can be observed that after the failure of motor

number 2, the controller commands zero thrust for the motor number 4, thus preventing flip-over of the drone.

VII. Conclusion

We have presented an algorithm for detecting/identifying the failures of the multi-rotor drone's motors. A 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, is also presented. The performance of the algorithms 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 parameters as well as test the algorithm in realistic flight experiments.

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 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 Quadrotor 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 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, Montreal, 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.

[22] NASA is actively conducting research in the domain of Unmanned Aerial System (UAS) Traffic Management (UTM) with the near-term goal to develop and demonstrate UTM to safely enable low-altitude airspace and UAS operations within the next five years (see http://utm.arc.nasa.gov/index.shtml)