



Partial engine fault detection and control of a Quadrotor considering model uncertainty

Davood Asadi*¹ 

¹Adana Alparslan Türkeş Science and Technology University, Aerospace Engineering Faculty, Adana, Turkey

Keywords

Fault tolerant Control
Fault detection
Engine Fault
Autonomous Emergency Landing

ABSTRACT

This paper presents a trajectory tracking fault-tolerant control strategy inside an autonomous emergency landing architecture to control a quadrotor in case of partial rotor fault. The proposed architecture, which is composed of required hardware and subsystems, aims to ensure a fully autonomous safe landing of the impaired quadrotor to a suitable landing site. The controller strategy, which is tried to be coincident with the proposed emergency landing architecture and the Pixhawk autopilot contains a cascade three-loop structure of adaptive sliding mode and a modified PID algorithm along with a fault detection algorithm. The adaptive sliding mode and the PID algorithms are applied to the fast dynamics of angular velocity rates and the position control of the quadrotor, respectively. A lightweight fault detection algorithm is developed to detect and identify the partial faults of engine using the controller outputs and the filtered angular rates. The simulation results demonstrate that the proposed fault-tolerant controller can control the multi-rotor in partial engine faults with satisfactory tracking performance. The results also demonstrate the effect of fault detection time delay on the overall control performance.

1. INTRODUCTION

The development of multi-rotor Unmanned Aerial Vehicles (UAVs) is being increased in both military and civil applications due to significant benefits such as compactness, maneuverability, mechanical simplicity, quick deployment time, low cost, and ability to operate in outdoor and indoor environments. This development grows the demand for the vital problem of safety for the UAVs (Chen et al., 2015; Saied et al., 2015; Mazeh et al. 2018). One common problem regarding the safety of multi-rotor UAVs is the occurrence of actuator fault (engine fault or propeller damage). The fault of actuator might lead to a crash and loss of system or might endanger human beings to injury risks in the environment of operation (Giribet et al., 2018, Nguyen et al., 2019, Lopez-Franco et al., 2017). Redundancy is not a logical solution for multi-rotors due to added weight and complexity to the system. Therefore, automation enhancement has attracted the researchers as a viable solution to the challenging problem of flight safety of multi-rotor UAVs in emergency flight conditions [Asadi et al., 2017a, b; Asadi et al. 2013; Asadi et al. 2014]. Fault-Tolerant Control (FTC) and Fault Detection and

Identification (FDI) modules are key elements in the Autonomous Emergency Landing Architecture (AELA) for safe landing of UAVs in emergency flight conditions. In addition to the FTC and FDI modules, trajectory planning and landing site detection are other required subsystems in the AELA structure, as proposed in this paper.

Different control algorithms ranging from linear to nonlinear, sliding mode, robust, and adaptive control algorithms have been proposed in the literature to control the faulty aerial vehicles. In (Lanzon et al., 2014), a controller is presented for a quadrotor with single rotor failure applying robust feedback linearization and assuming that the failure is known. In Ref (Milton et al., 2019), a trajectory tracking adaptive dynamic controller was proposed to guide an unmanned aerial vehicle (UAV). Stability and control of quadcopters experiencing one, two, or three rotor failures are presented in (Mueller and Andrea 2014). Several control methods based on L_1 adaptive control (Ahmadi et al. 2017), Robust adaptive Control (Asadi and Ahmadi, 2020), sliding mode control (SMC) (Sharifi et al., 2010), and adaptive SMC techniques (Asadi et al., 2017, Barghandan et al., 2017) are proposed to handle model uncertainties and faults. Considering

* Corresponding Author

^{*}(dasadihendoustani@atu.edu.tr) ORCID ID 0000 – 0002 – 2066 – 6016

Cite this article

Asadi D (2022). Partial engine fault detection and control of a Quadrotor considering model Uncertainty, Turkish Journal of Engineering – 2022; 6(2); 106-117

model-based control, the authors of (Mohammadi and Shahri 2013) proposed a model reference adaptive control (MRAC) technique to control a quadrotor with parametric and non-parametric uncertainties in the model. Supposing loss in the efficiency of a quadrotor’s propeller, control algorithms are proposed in (Ranjbaran and Khorasani 2010, Zhang and Chamseddine, 2012) to estimate the aerial vehicle model after the failure, guaranteeing the stability of the platform. A back stepping approach is proposed in (Khebbache, et al., 2012) for the performance loss of the motors. In fault-tolerant control of multi-rotors, the existence of a fast converging algorithm is necessary for fault detection. Fault detection approaches can be categorized into model-based, signal-based, knowledge-based, and active diagnosis techniques (Zhiwei et al. 2015). (Amoozgar, et al. 2013) proposed a two-stage Kalman filter to detect, isolate, and estimate possible faults in each motor whereas the method was evaluated on a UAV testbed. (Avram, et al., 2017) presented a fault identification scheme for actuator faults using nonlinear adaptive estimation technique. In Ref. (Cen, et al., 2014), an adaptive Thau observer was developed to estimate and detect the actuator faults. An on-line detection algorithm was proposed to detect a single motor failure and a control allocation technique is proposed to control the drone in (Frangenberg, et al., 2015). A parity space method with recursive least squares algorithm was introduced in (Han et al., 2018), for actuator faults detection and identification of a drone.

Landing site detection and trajectory planning modules are other critical issues besides the fault detection and control algorithm for safe landing of an impaired drone. Therefore, an Autonomous Emergency Landing Architecture (AELA) based on the Pixhawk autopilot is proposed for emergency landing of an impaired multi-rotor drone. Moreover, this research presents a fault-tolerant trajectory tracking control strategy, which can be implemented as a subsystem in the AELA. Therefore, this research is innovative in several ways including; 1) suggestion of AELA strategy to enable the faulty drone for a safe landing, 2) development of a complete mathematical nonlinear model considering dynamic uncertainties due to rotors’ gyroscopic effects and airframe drag along with partial actuator fault effect, 3) application of a cascade fault-tolerant adaptive sliding model controller integrated with a modified PID algorithm with anti-windup filter, 4) application of a fast converging fault detection algorithm for partial engine fault detection based on the extended Kalman filter estimate of states. In order to be coincident with the Pixhawk software structure and having minimum required changes, a three-loop control algorithm similar to the Pixhawk autopilot is proposed in the controller design section.

In section II the AELA structure and in section III the mathematical modeling is discussed. Section IV introduces the fault tolerant control and fault detection algorithm. Sections V presents the simulation results that validate the proposed approach. Finally, section VI points out the key conclusions.

2. Autonomous Emergency Landing Architecture

In fault scenarios where continuation of flight is not possible or endangers the flight safety, the emergency flight system is triggered to recover the drone’s stability and safely guide it toward a suitable landing site (Nguyen et al., 2019). There are several challenges respecting the safe landing of an impaired drone such as obstacle detection, suitable landing site detection/selection, fault detection and identification, characterizing the aircraft’s kinematic constraints, trajectory planning to the desired landing area, and control of the faulty drone toward the landing site. To cover these challenges, an Autonomous Emergency Landing Architecture (AELA) is proposed according to Figure 1. The AELA is translated to an architecture consisting of various subsystems that are capable of landing a faulty drone to the desired landing site over a designed trajectory without colliding with human or animal. In the proposed AELA, the fault detection algorithm identifies the fault based on the filtered IMU (Inertial Measurement Unit) and GPS data applying the Kalman filter. In the meanwhile, on-line images in the drone’s Field of View (FOV), taken by the onboard camera along with image processing algorithms are implemented for suitable landing site detection as well as obstacles/human detection in the vicinity of the desired landing site. The area that is free from the obstacle and closest to the center of the image is defined as the most immediate and suitable landing zone. After characterizing a suitable landing site, the trajectory generation unit generates a controllable trajectory toward the selected landing site. The controller generates the required total thrust and three-axis torques to track the desired trajectory toward the landing site. A Multi-Criteria Decision-Making (MCDM) algorithm manages the strategic decision and controls the whole architecture. The emergency landing mission continues as long as the number of functioning motors provides controllability of the drone otherwise MCDM switches to a safe crash.

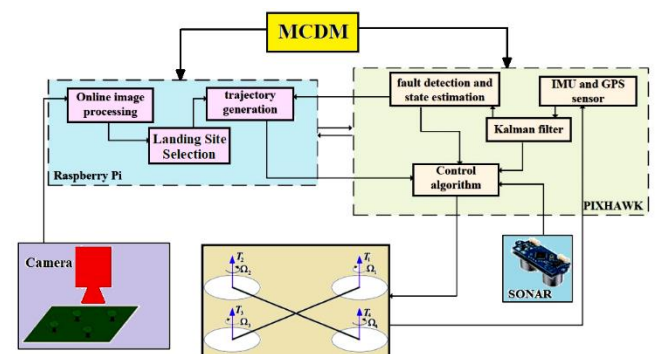


Figure 1. Autonomous Emergency Landing Architecture with Subsystems

The Pixhawk flight controller which is equipped with GPS, IMU, and SONAR is proposed as the autopilot board, which enables the engine control, navigation, and communication with the ground station (Desaraju et al., 2015). The Pixhawk applies an open-source flight control software (Github PX4 Documentation), which can be used as the baseline for the developed algorithms required for emergency landing. The controller

algorithms, the guidance and navigation algorithms, estimators, and filters, can be modified according to the required proposed algorithms for AELA. Image processing and trajectory planning task is executed on the “Raspberry Pie 3” board. A connection between the Raspberry Pie and Pixhawk is required to transmit the position of the landing area and obstacles for guidance and control of the drone.

3. Mathematical Model

In this section, the quadrotor model, the engine model, as well as the motor mixer equations are presented.

3.1. Quadrotor Model

The S500 frame (Fig. 3) with the EMAX MT3506 650KV motors with 11-inch diameter propellers have been selected in this research. The quadrotor parameters which is used in the simulations are given in Table 1.

Table 1. Quadrotor frame Specifications

Quadrotor Parameters	
Mass, m	1.6 kg
Thrust Coefficient, b	4.068×10^{-7} N/rpm ²
Drag Coefficient, d	8.43×10^{-9} Nm/rpm ²
Moment arm, l	0.243 m
Moment of Inertia about x-axis, I_{xx}	0.0213 kg.m ²
Moment of Inertia about y-axis, I_{yy}	0.0221 kg.m ²
Moment of Inertia about z-axis, I_{zz}	0.028 kg.m ²
Total rotational moment of inertia, J_T	6.8×10^{-5} kg.m ²

3.2. Dynamic Equations

The translational and rotational equations of the quadrotor in the body frame are presented in Eq.s (1) and (2), respectively. As depicted in Figure 2, the quadrotor consists of four motors, which motors number one and three rotate clock-wise with velocities Ω_1, Ω_3 , respectively, whereas the other two other motors (2, 4) rotate in the opposite (counter-clockwise) direction with velocities Ω_2, Ω_4 .

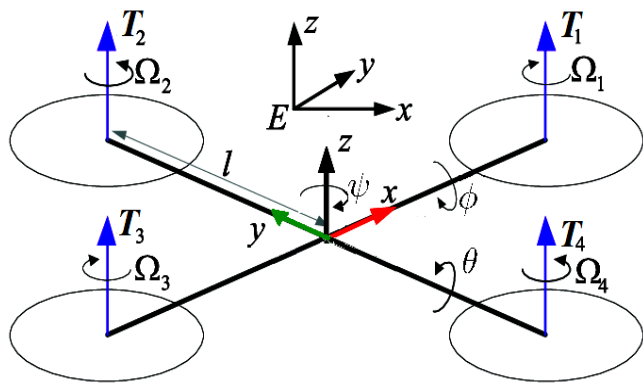


Figure 2. Schematic representation of quadrotor

Applying the rigid-body equations of motion and Euler angle transformation (Eq. (4)), the complete dynamical model of the quadrotor is presented as below:

3.2.1. Translational dynamics

$$\begin{aligned} \ddot{x} &= (\sin \psi \sin \phi + \cos \psi \sin \theta \cos \phi) \frac{u_z}{m} - \frac{k_x}{m} \dot{x} \\ \ddot{y} &= (-\cos \psi \sin \phi + \sin \psi \sin \theta \cos \phi) \frac{u_z}{m} - \frac{k_y}{m} \dot{y} \end{aligned} \quad (1)$$

$$\ddot{z} = -g + (\cos \theta \cos \phi) \frac{u_z}{m} - \frac{k_z}{m} \dot{z}$$

where x, y , and z are the position of quadrotor center of mass in inertial frame and ψ, θ, ϕ are the Euler angles, which represent the body-frame rotation with respect to the Inertial frame.

The input u_z is the total lift force (u_z) generated by propellers in z -direction. $k_x = k_y = 5.5e-4$ N/m/s, $k_z = 6.3e-4$ N/m/s are translational drag coefficients. The last terms in the above equation regarding the airframe drag are applied in the equations of motion but is considered as model uncertainty, when designing the controller algorithm.

3.2.2. Rotational dynamics

$$\begin{aligned} \dot{p} &= \frac{I_{yy} - I_{zz}}{I_{xx}} q r + \frac{u_\phi}{I_{xx}} + \frac{J_T}{I_{xx}} q \Omega - \frac{k_\phi}{I_{xx}} p^2 \\ \dot{q} &= \frac{I_{zz} - I_{xx}}{I_{yy}} p r + \frac{u_\theta}{I_{yy}} - \frac{J_T}{I_{yy}} p \Omega - \frac{k_\theta}{I_{yy}} q^2 \\ \dot{r} &= \frac{I_{xx} - I_{yy}}{I_{zz}} p q + \frac{u_\psi}{I_{zz}} - \frac{k_\psi}{I_{zz}} r^2 \end{aligned} \quad (2)$$

where I_{xx}, I_{yy} and I_{zz} are the moment of inertia in x, y and z direction, respectively. The quadrotor inputs are represented by u_ϕ, u_θ, u_ψ , which are moments about x, y, z axes, respectively. $k_\phi = k_\theta = 5.5e-4$ N/rad/s, $k_\psi = 6.35e-4$ N/rad/s are rotational drag coefficients. J_T is the moment of inertia of each motor and Ω represents the propellers total speed as below:

$$\Omega = -\Omega_1 - \Omega_2 + \Omega_3 + \Omega_4 \quad (3)$$

The drag and gyroscopic terms in the rotational dynamics equations are the unknown terms and are considered as the model uncertainty.

3.2.3. Euler Equations

$$\begin{aligned} \dot{\phi} &= p + q \sin \phi \tan \theta + r \cos \phi \tan \theta \\ \dot{\theta} &= q \cos \phi - r \sin \phi \\ \dot{\psi} &= \frac{1}{\cos \theta} [q \sin \phi + r \cos \phi] \end{aligned} \quad (4)$$

Where ψ, θ, ϕ are the Euler angles, which represent the body frame rotation with respect to the inertial frame.

3.3. Rotor Dynamics

The thrust generated by the engines are modeled as a first-order system to account for the engines dynamic for variation of rotational speed:

$$u_i = K \frac{\omega_0}{S + \omega_0} u_i \quad (5)$$

where u_i is the i -th engine input which is the PWM reference signal to the engines, K is the motor gain, and ω_0 is the bandwidth of the engines. The engines' thrust force and torque depend on the rotational velocity, propeller diameter, as well as the aerodynamics characteristics of blades as below:

$$\begin{aligned} T_i &= C_t \rho \Omega_i^2 D^4 = b \Omega_i^2 \\ Q_i &= C_d \rho \Omega_i^2 D^5 = k T_i = d \Omega_i^2, \quad k = 2.07e-2 \quad m \end{aligned} \quad (6)$$

where C_t , C_d are thrust and drag coefficients, ρ is air density, Ω_i is the rotational speed of each engine in rpm, and D is the propeller diameter. The numerical values of b and d are introduced in Table 1. Accordingly, the actuation inputs around body-frame are expressed as follows:

$$\begin{aligned} u_z &= T_1 + T_2 + T_3 + T_4 = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ u_\phi &= l(T_2 - T_4) = bl(\Omega_2^2 - \Omega_4^2) \\ u_\theta &= l(T_3 - T_1) = bl(\Omega_3^2 - \Omega_1^2) \\ u_\psi &= k(T_1 + T_3 - T_2 - T_4) = d(\Omega_1^2 + \Omega_3^2 - \Omega_2^2 - \Omega_4^2) \end{aligned} \quad (7)$$

The autopilot outputs ($u_z, u_\phi, u_\theta, u_\psi$) must be translated into each motor inputs to send the signal to the quadrotor speed controls then apply the related PWM signal to each quadrotor's engine.

3.4. Motor Mixer

Given that the controller computes the intermediate control inputs ($u_z, u_\phi, u_\theta, u_\psi$), these are transformed into the engines desired rotational velocities. Typically, this transformation is called motor mixing. By inverting the relationship of Eq. (7), motor mixer expression is derived as below:

$$\begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} = \begin{bmatrix} \frac{1}{4b} & 0 & \frac{-1}{2lb} & \frac{1}{4d} \\ \frac{1}{4b} & \frac{1}{2lb} & 0 & \frac{-1}{4d} \\ \frac{1}{4b} & 0 & \frac{1}{2lb} & \frac{1}{4d} \\ \frac{1}{4b} & \frac{-1}{2lb} & 0 & \frac{-1}{4d} \end{bmatrix} \begin{bmatrix} u_z \\ u_\phi \\ u_\theta \\ u_\psi \end{bmatrix} \quad (8)$$

3.5. Engine fault modeling

The fault considered in this paper is a sudden partial loss of engine propeller. If damage occurs on the propeller, the thrust and torque of the faulty engine can be expressed as a parametric uncertainty as follows:

$$\begin{aligned} T_{if} &= T_i + \Delta T_i = b \Omega_i^2 + \Delta b \Omega_i^2, \quad \Delta b = -f_i b \\ Q_{if} &= Q_i + \Delta Q_i = d \Omega_i^2 + \Delta d \Omega_i^2, \quad \Delta d = -f_i d \end{aligned} \quad (9)$$

Where Δb , Δd are bounded variation of propeller effectiveness respecting its nominal values and can be represented as $-b \leq \Delta b \leq 0$, $-d \leq \Delta d \leq 0$ and f_i is the i -

th engine fault. Therefore, the actual signal (U) generated by the faulty actuator (U^f) is as follows:

$$U^f(t) = (1 - \Gamma)U(t) \quad (10)$$

$$\Gamma = \begin{cases} 0 & t < t_f \\ \text{diag}(f_1, f_2, f_3, f_4) & t > t_f \end{cases}$$

Where $0 \leq f_i < 1$ and $f_i = 0, f_i = 1$ represent the healthy and the fully damaged rotor, respectively and t_f is the time that fault occurs.

4. Fault Tolerant Control Strategy

In this section, by applying the multiple-timescales approach, the rotational and translational dynamics are separated by assuming that the rotational dynamics is much faster than the translational dynamics. In the PX4 software of the Pixhawk autopilot, a three-loop cascade architecture composed of a rate controller, an attitude controller, and a position controller exists to control the aerial vehicles.

Thus, in order to be coincident and consistent with the Pixhawk software structure and having minimum required changes on the software, a three-loop control algorithm is proposed in this paper. According to Fig. 3, for the fast inner-loop, a nonlinear adaptive sliding mode controller is applied and for the two slow outer-loops of attitude and position, the desired dynamics and PID algorithm similar to the PX4 algorithm with some modifications is proposed.

Based on the rotational dynamics according to Eq. (2), the uncertainty due to engine fault, and the fault model based on Eq. (11), the nonlinear model of quadrotor can be transformed into an affine control model as below:

$$\dot{x} = f(x) + \Delta f(x) + [g(x) + \Delta g]U \quad (11)$$

where $x \in \mathfrak{R}^3$ is the vector of rotational velocities ($x = [p, q, r]$); $U \in \mathfrak{R}^3$ is the plant control input vector ($U = [u_\phi, u_\theta, u_\psi]$); $f(x) \in \mathfrak{R}^3$ and $g(x) \in \mathfrak{R}^{3 \times 3}$ are known differentiable vector functions and input matrices, respectively, $\Delta g(x) \in \mathfrak{R}^{3 \times 3}$ is the unknown bounded perturbations due to the propeller damage, and $\Delta f(x) \in \mathfrak{R}^3$ is an unknown bounded uncertainty caused by propellers gyroscopic effects and airframe drag as defined below:

$$f(x) = \begin{bmatrix} \frac{I_{yy} - I_{zz}}{I_{xx}} qr \\ \frac{I_{zz} - I_{xx}}{I_{yy}} pr \\ \frac{I_{xx} - I_{yy}}{I_{zz}} pq \end{bmatrix}, \quad \Delta f(x) = \begin{bmatrix} \frac{J_T}{I_{xx}} q \Omega - \frac{k_\phi}{I_{xx}} p^2 \\ -\frac{J_T}{I_{yy}} p \Omega - \frac{k_\theta}{I_{yy}} q^2 \\ -\frac{k_\psi}{I_{zz}} r^2 \end{bmatrix} \quad (12)$$

$$g(x) = \text{diag}\left(\frac{l}{I_{xx}}, \frac{l}{I_{yy}}, \frac{l}{I_{zz}}\right), \quad (13)$$

$$\Delta g(x) = \text{diag}\left(\frac{-l f_1}{I_{xx}}, \frac{-l f_2}{I_{yy}}, \frac{-l f_3}{I_{zz}}\right)$$

4.1. Inner-Loop Controller Design

For the inner-loop rotational dynamics an adaptive sliding mode controller is proposed. Accordingly, the sliding function is defined based on the error dynamics as below:

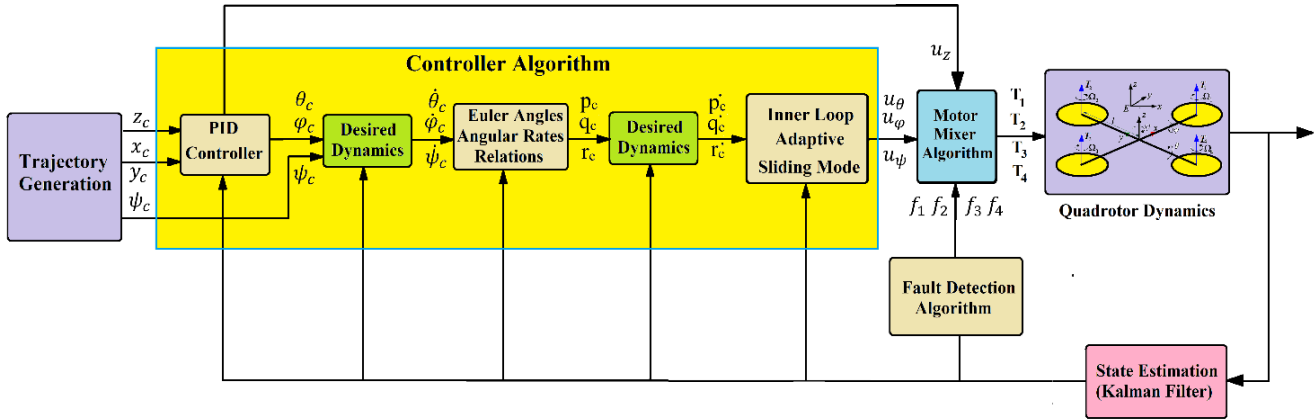


Figure 3. Three-Loop Fault-Tolerant Control Block Diagram

$$S = e + k \int_0^t e d\tau, S, e \in \mathbb{R}^3 \quad (14)$$

$$S = [s_1, s_2, s_3], e = [e_1, e_2, e_3]$$

where e is the error defined as $e_i = x_{c_i} - x_i$, and $k = \text{diag}\{k_i\}$, $i = 1, 2, 3$ is the gain defined to achieve the desired dynamics. Based on the introduced sliding function, a positive definite Lyapunov function is defined and the sliding mode algorithm is derived based on the time derivation of the Lyapunov function:

$$\dot{V} = \frac{1}{2} S^T S \quad (15)$$

$$\dot{V} = S^T (\dot{x}_c - f(x) - \Delta f(x) - (g(x) - \Delta g(x))\delta + k e)$$

$$\dot{V} = S^T (\psi_0 - (g(x) - \Delta g(x))\delta), \psi_0 = \dot{x}_c - f - \Delta f + k e$$

By forcing the time derivative of the Lyapunov function to be negative $\dot{V} < 0$, the error dynamics can be reduced. Proper selection of input commands can guarantee the stability of the rotation dynamics as below:

$$\delta_c = \hat{\delta}_{c_{eq}} + \text{diag}(\rho_i) g^{-1}(x) \text{SIGN}(S)$$

$$\hat{\delta}_{c_{eq}} = g^{-1}(x) \hat{\psi}_0, \hat{\psi}_0 = \dot{x}_c - f(x) + k e \quad (16)$$

$$\rho_i = \rho_{b_i} + k_{\rho_i} \int_0^t |s_i| d\tau, \rho_i > 0$$

where SIGN (S) is the sign function operating on the sliding function and other terms are defined as below:

$$\varphi_0 = \psi_0 - (I + E_0) \hat{\psi}_0, \varphi_0 \in \mathbb{R}^3, I \in \mathbb{R}^{3 \times 3}$$

$$E_0 = \Delta g_1 g_1^{-1}, E_0 \in \mathbb{R}^{3 \times 3}, \quad (17)$$

$$\tilde{\rho}_i = \rho_{b_i} - \rho_i, \rho_{b_i} = \frac{\gamma_i}{1 - L_i}$$

Where $\dot{\gamma}$ and L_i are the logical bounds on the functions φ_0 and E_0 , respectively and the ρ_i values are also defined based on the aforementioned bounds as below:

$$|\varphi_{0i}| < \gamma_i < \infty, |E_{0ij}| < \sigma_{ij}, \sum_{j=1}^3 \sigma_{ij} < L_i < 1, \rho_i > \frac{\gamma_i}{1 - L_i} \quad (18)$$

$$\forall i = 1, 2, 3, \forall j = 1, 2, 3$$

The output commands of the sliding mode controller in Eq. (16) has undesirable chattering problem during the system operation on the sliding function and therefore must be removed. Thus, a continuous standard saturation function, $\text{sat}()$, is defined and replaced instead of SIGN(S) function to avoid this problem as below:

$$\text{sat}(s_i/\varepsilon_i) = \begin{cases} 1 & \text{if } s_i > \varepsilon_i \\ s_i/\varepsilon_i & \text{if } |s_i| \leq \varepsilon_i \\ -1 & \text{if } s_i < -\varepsilon_i \end{cases} \quad (19)$$

Where ε_i is a constant value defined by the designer to avoid the chattering problem. Applying the adaptive sliding controller in Eq. (18), the asymptotically tracking of the inner-loop desired inputs $x = \{p_c, q_c, r_c\}$ can be ensured by demonstrating that the Lyapunov function is negative. The details of the stability proof and adaptive sliding mode control algorithm has been presented in our previous research (Asadi et al., 2017)

A proportional-integral (PI) feedback controller is applied to track the reference model and generating the pseudo input \dot{x}_c (Asadi et al., 2017) according to Fig. 4.

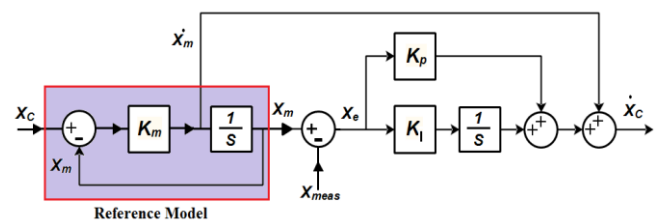


Figure 4. Desired Dynamics Block Diagram

4.2. Outer Loop Controller Design

For the outer-loop position control, a modified PID control algorithm is applied. Based on the desired trajectory and their first and second derivatives, the dynamics of the position error can be derived as:

$$\begin{aligned} \ddot{P}_e + K_d \dot{P}_e + K_p P_e + K_i \int P_e dt = 0 \\ P_d = [x_d, y_d, z_d], P_e = P_d - P \end{aligned} \quad (20)$$

Where P_d is the desired position with bounded first and second derivatives, P_e is the positions error, and the PID gains (K_p, K_d, K_i) are derived based on the conditions of Routh-Hurwitz to exponentially converge the error to zero. Based on the error dynamics, the following equation can be derived:

$$\ddot{P} = \ddot{P}_d + K_d \dot{P}_e + K_p P_e + K_i \int P_e dt = 0 \quad (21)$$

Based on the desired positions and translational dynamics of Eq. (1), the desired Euler angles can be derived as below:

$$\begin{cases} \theta_c = \arcsin\left(\frac{m\ddot{x}}{\cos\psi_d u_z}\right) \\ \varphi_c = -\arcsin\left(\frac{m\ddot{y}}{\cos\psi_d u_z}\right) \end{cases} \quad (22)$$

where in the above equation $u_z = m\sqrt{\ddot{x} + \ddot{y} + (\ddot{z} + g)}$ and the desire heading angle (ψ_d) is imposed by the trajectory generation module.

4.3. State Estimation

The extended Kalman filter is applied for the quadrotor state estimation using the GPS, IMU, magnetometer, SONAR, and accelerometers data in the presence of sensor noise and system disturbance. IMU measures the angular rates, magnetometer, and accelerometer data are applied to estimate the Euler angles (based on Eq. (22)), and SONAR and GPS are used to measure the altitude and position. Extended Kalman filters apply a two-step recursive process, which predicts the estimates, by applying prior knowledge of the states and then correct the states by using the current measurements. The nonlinear dynamics of quadrotor can be shown as below:

$$\begin{aligned} x_k &= f(x_{k-1}, u_k) + w_k \\ z_k &= h(x_k) + v_k \end{aligned} \quad (23)$$

where u_k is the control input and w_k and v_k are the process and output noises with assumed zero mean and with covariances Q_k and R_k , respectively. By linearizing the nonlinear system about each trim condition, the linear state equation and the error covariance matrix is derived as:

$$\begin{aligned} \hat{x}_{k|k-1} &= F_k \hat{x}_{k-1|k-1} + B_{k-1} u_{k-1} \\ P_{k|k-1} &= F_k P_{k-1|k-1} F_k^T + Q_{k-1} \end{aligned} \quad (24)$$

Where $\hat{x}_{k|k-1}$ is the state estimation, F_k is the transition matrix, u_{k-1} is the control input, B_{k-1} is the control input matrix, and $P_{k|k-1}$ is the error covariance matrix. The readers are referred to Ref (Tarhan and Altug, 2011, Merheb 2016) for more details about the Extended Kalman filter and the linearization of state.

4.4. Fault detection and Identification

The relationship between the controller output and each engine thrust considering the effect of fault is as follows:

$$\begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} = \begin{bmatrix} \frac{1}{4(1-f_1)} & 0 & \frac{-1}{2(1-f_1)} & \frac{1}{4(1-f_1)} \\ \frac{1}{4(1-f_2)} & \frac{1}{2(1-f_2)} & 0 & \frac{-1}{4(1-f_2)} \\ \frac{1}{4(1-f_3)} & 0 & \frac{1}{2(1-f_3)} & \frac{1}{4(1-f_3)} \\ \frac{1}{4(1-f_4)} & \frac{-1}{2(1-f_4)} & 0 & \frac{-1}{4(1-f_4)} \end{bmatrix} \begin{bmatrix} u_z \\ u_\theta \\ u_\varphi \\ u_\psi \end{bmatrix} \quad (25)$$

Since there is no sensor to measure the engine thrust, estimated states by Kalman filter and the controller outputs are used to detect and identify the partial propeller fault. In the case of propeller damage, the objective is to detect the faulty engine and identify the magnitude of fault f_i .

The differences between the controller outputs and the angular rates after one sample time (T_s) are used for fault detection. The criteria for the occurrence of partial damage to each engine is as follows:

- If $\Delta u_\theta < u_{\theta_{threshold}}$ and $\Delta q > q_{threshold}$, number **one** engine is faulty.
- If $\Delta u_\theta < u_{\theta_{threshold}}$ and $\Delta q > q_{threshold}$, number **two** engine is faulty.
- If $\Delta u_\varphi < u_{\varphi_{threshold}}$ and $\Delta p > p_{threshold}$, number **three** engine is faulty.
- If $\Delta u_\varphi < u_{\varphi_{threshold}}$ and $\Delta p > p_{threshold}$, number **four** engine is faulty.

where $\Delta u_\varphi, \Delta u_\theta$ are the input change one step time before the fault time as $\Delta u_\varphi = u_\varphi(t_f - T_s) - u_\varphi(t_f - 2T_s)$ and Δp and Δq are the change of angular velocity at the time of fault as $\Delta p = p(t_f) - p(t_f - T_s)$. The angular rates are measured using the extended Kalman filter. The presented control architecture is able to effectively track the desired trajectory considering uncertainty and limited amounts of propeller damage but is not able to control the system in the presence of severe faults. Therefore, the threshold values are determined based on the performance of the controller without fault detection algorithm.

Table 2 presents a lookup table which is prepared by injecting different magnitudes of the fault and measuring the filtered variation of angular rotations ($\Delta p, \Delta q$) in the simulation. A fixed time-step of $T_s = 0.01$ has been applied in the simulation. Based on the quadrotor rotational equations the rotational rates can be estimated as below:

$$\frac{(p_k - p_{k+1})}{T_s} \cong \frac{l}{I_x} (T_4 k_4 - T_2 k_2) \quad (26)$$

Where $k_i = 1 - f_i$ in the above equation. Considering each engine thrust force equal to $T_4 = T_2 = W/4$ in hover

trim condition and considering that damage occurs on one of the engines, $\Delta k_p = k_4 - k_2$ can be estimated as below:

$$\Delta k_p \cong \frac{4I_x}{lW} \left(\frac{\Delta p}{T_s} \right) \quad (27)$$

Therefore, if $\Delta p < 0$, then $\Delta k_p < 0$ meaning that engine number four is faulty otherwise if $\Delta p > 0$, then $\Delta k_p > 0$ and therefore number two engine is faulty. Since it is assumed that only one engine is faulty, the percentage of fault magnitude is equal to $f = 1 - \Delta k_p$.

Table 2. Look up table for engine fault estimation, values in deg/s

Engine	10%	20%	30%	40%	50%
No.1	$\Delta q = -2.55$	$\Delta q = -5.1$	$\Delta q = -7.66$	$\Delta q = -9.87$	$\Delta q = -12.62$
	$ \Delta p < p_{threshold}$	$ \Delta p < p_{threshold}$	$ \Delta p < p_{threshold}$	$ \Delta p < p_{threshold}$	$ \Delta p < p_{threshold}$
No.2	$\Delta p = 2.55$	$\Delta p = 5.1$	$\Delta p = 7.66$	$\Delta p = 9.87$	$\Delta p = 12.62$
	$ \Delta q < q_{threshold}$	$ \Delta q < q_{threshold}$	$ \Delta q < q_{threshold}$	$ \Delta q < q_{threshold}$	$ \Delta q < q_{threshold}$
No.3	$\Delta q = 2.55$	$\Delta q = 5.1$	$\Delta q = 7.66$	$\Delta q = 9.87$	$\Delta q = 12.62$
	$ \Delta p < p_{threshold}$	$ \Delta p < p_{threshold}$	$ \Delta p < p_{threshold}$	$ \Delta p < p_{threshold}$	$ \Delta p < p_{threshold}$
No.4	$\Delta p = -2.55$	$\Delta p = -5.1$	$\Delta p = -7.66$	$\Delta p = -9.87$	$\Delta p = -12.62$
	$ \Delta q < q_{threshold}$	$ \Delta q < q_{threshold}$	$ \Delta q < q_{threshold}$	$ \Delta q < q_{threshold}$	$ \Delta q < q_{threshold}$

The same formulation and results can be derived for the pitch dynamics respecting the percentage of fault regarding number one and three engines. Table 2 shows the signs and magnitudes of rotational angular velocity variation in different percentages of propeller fault where the maximum fault is considered to be 50% of partial damage on one of the quadrotor engine's propeller.

The magnitude of thresholds in **Table 2** is determined based on the magnitudes of Δp and Δq for 5% of propeller damage. It will be shown that the proposed adaptive sliding mode controller is robust and has desirable performance for 5% of damage to any one of the engine's propeller.

5. Simulation results

Several numerical simulations are considered in the presence of partial loss of rotor effectiveness to verify the performance of the proposed three-loop fault-tolerant controller. First, the robustness of the controller is investigated without applying the fault detection

algorithm. Next, the performance of the fault-tolerant control is examined for different percentages of fault on different engines.

5.1. Controller without fault detection algorithm

The robustness of the proposed adaptive sliding mode controller is examined in the presence of 5% and 10% of the propeller fault of number one engine. Fig. 5 illustrates the controller performance for the above cases. The fault is introduced to the quadrotor model at $t=10s$ while the control algorithm tries to recover the quadrotor stability and tracking the desired inputs of position (x, y, h) and a constant heading rate (ψ) . Accordingly, the proposed controller performance is satisfactory for 5% of the fault but is not desirable in 10% of fault. Based on the simulations it can be inferred that the proposed controller is robust for 10% of the engine fault but does not have a good tracking performance. Therefore, the fault detection algorithm will be applied for the faults bigger than 5%.

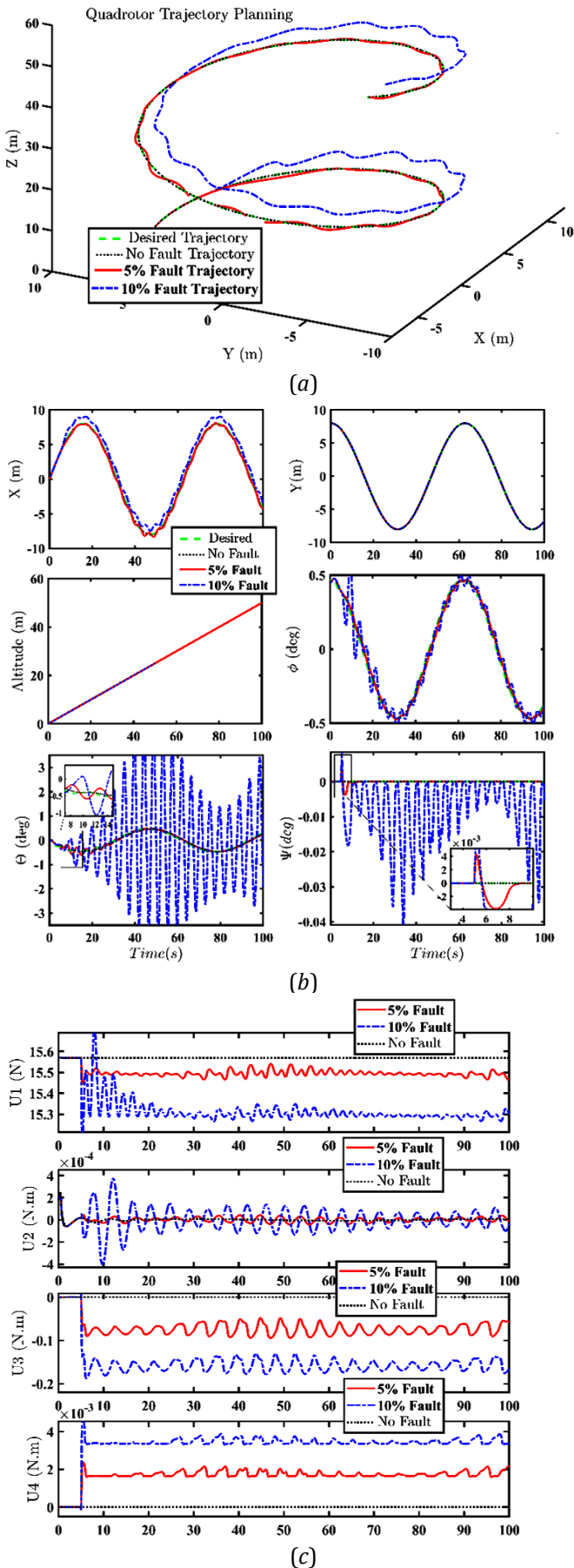


Fig. 5 Controller performance without fault detection algorithm for: no fault, 5% propeller fault, and 10% of fault, (a) 3-D trajectory, (b) Position and Euler angles, (c) Controller outputs

5.2. Fault-tolerant Controller performance

The proposed controller must recover the stability of quadrotor and track the desired trajectory in the presence of abrupt propeller effectiveness up to 50% of damage. For the propeller faults more than 50% the faulty engine will be saturated and cannot generate enough thrust to control all three rotations of quadrotor around x, y, and z axes. For more severe faults or complete loss of one engine, which is not the case of analysis in this paper, the control of one axis (around z) is neglected and the other two axes (roll and pitch) are controlled. According to the lookup table of Table 2, the variation of pitch and roll rates is used to detect and identify the partial rotor damage. For this purpose, the Extended Kalman Filter (EKF) is applied to minimize the effect of noise.

The performance of the proposed fault-tolerant controller is examined in two separate fault scenarios. 40% of fault is introduced to the engine number 2, in the first scenario. Fig. 6 illustrates the tracking performance of the controller in the first scenario. Fig.10a represents the position and Euler angles tracking. A sudden increase in roll angle and yaw angle is experienced due to the propeller damage at $t=10s$ but the controller satisfactorily recovers the roll and yaw angles and tracks the desired input values. Fig.6 (b, c) illustrates the controller outputs and each engine rotational velocity that affect the thrust force, respectively. Accordingly, the rotational speed of the faulty engine suddenly increases to compensate for the thrust reduction of the due to 40% of propeller damage.

Fig.6d illustrate the variation of estimated pitch rate, which is used for fault estimation. At time $t_f = 10s$, the pitch rate suddenly changes from around zero to 12 deg/s which indicates 50% of fault in the second engine as described in Table 2. In fact, the fault of number 2 engine induces an abrupt positive roll rate and consequently positive roll angle. Fig.6e depicts the estimated percentage of fault which occurs one time-step ahead at $t = 10.01s$. According to the described fault detection logic, the value of fault will be determined based on the initial jump of the angular rates bigger than a threshold value, which is considered as 2 deg/s . In fact, this value is less than the effect of 5% damage, which can be tackled by the controller without fault detection and is bigger than the effect of state and output noises.

Scenario 2, investigates the control algorithm performance for the partial damage to the propeller of engine number one. The fault to engine one induces a sudden negative pitch rate according to Eq. (1). Simulation results of different fault cases of 30%, 40%, and 50% demonstrate the acceptable performance of the proposed strategy in controlling the drone in the presence of noise and engine fault. Fig. 7a and Fig 7b illustrates the Euler angles and the rotational speeds of each engine, respectively. Accordingly, a sudden increase in the rotational speed of engine number one is obvious. Fig. 7(c, d) illustrate the fault detection procedure based on the variations of angular rates and the values presented in Table 2 for the 40% of the fault to engine number 1. At $t_f = 10 s$, the pitch rate suddenly changes from zero to -10 deg/s .

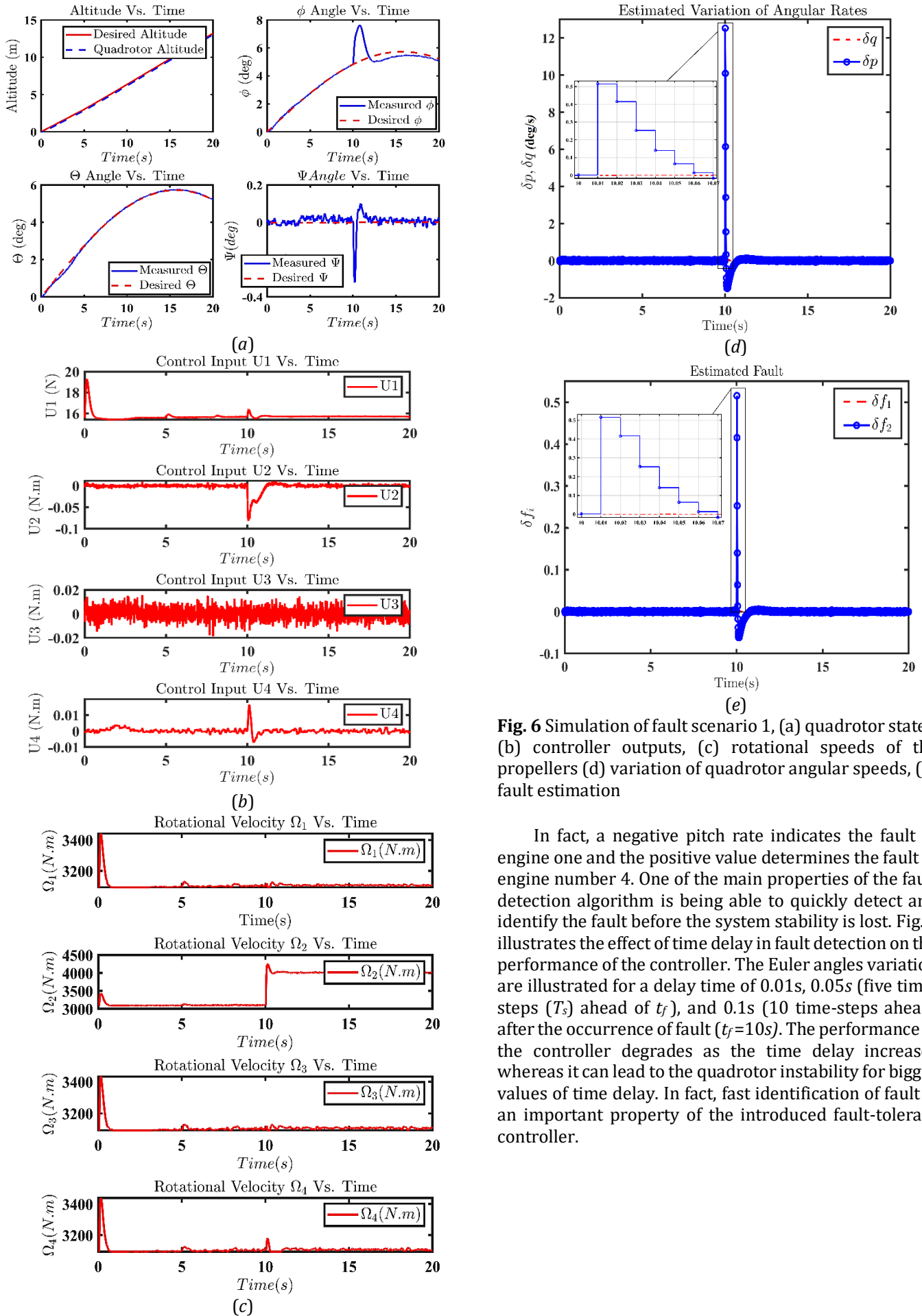


Fig. 6 Simulation of fault scenario 1, (a) quadrotor states, (b) controller outputs, (c) rotational speeds of the propellers (d) variation of quadrotor angular speeds, (e) fault estimation

In fact, a negative pitch rate indicates the fault of engine one and the positive value determines the fault in engine number 4. One of the main properties of the fault detection algorithm is being able to quickly detect and identify the fault before the system stability is lost. Fig. 8 illustrates the effect of time delay in fault detection on the performance of the controller. The Euler angles variation are illustrated for a delay time of 0.01s, 0.05s (five time-steps (T_s) ahead of t_f), and 0.1s (10 time-steps ahead) after the occurrence of fault ($t_f=10s$). The performance of the controller degrades as the time delay increases whereas it can lead to the quadrotor instability for bigger values of time delay. In fact, fast identification of fault is an important property of the introduced fault-tolerant controller.

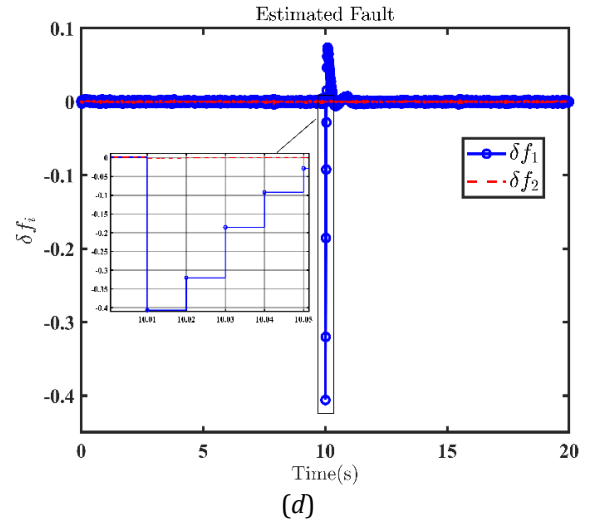
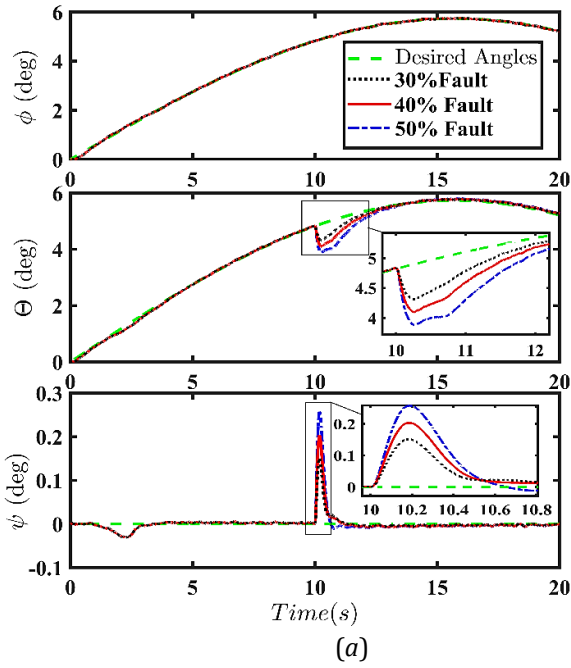


Fig. 7 Simulation of fault scenario 2, (a) quadrotor Euler angles, (b) rotational speeds of the propellers (c) variation of quadrotor angular speeds, (d) fault estimation

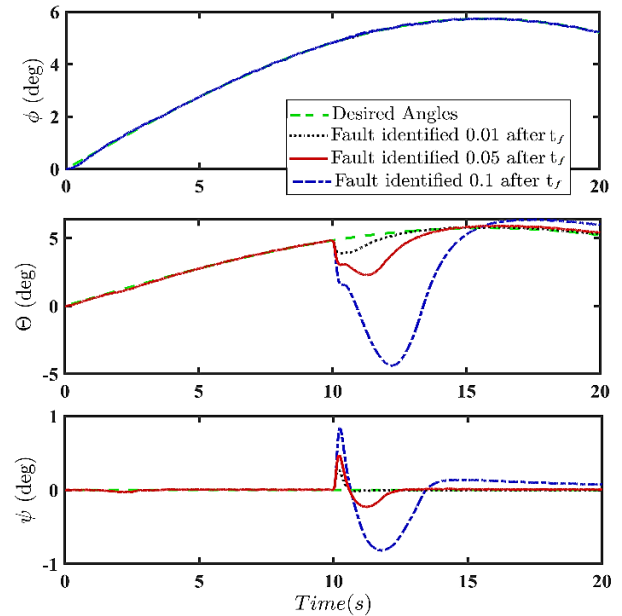
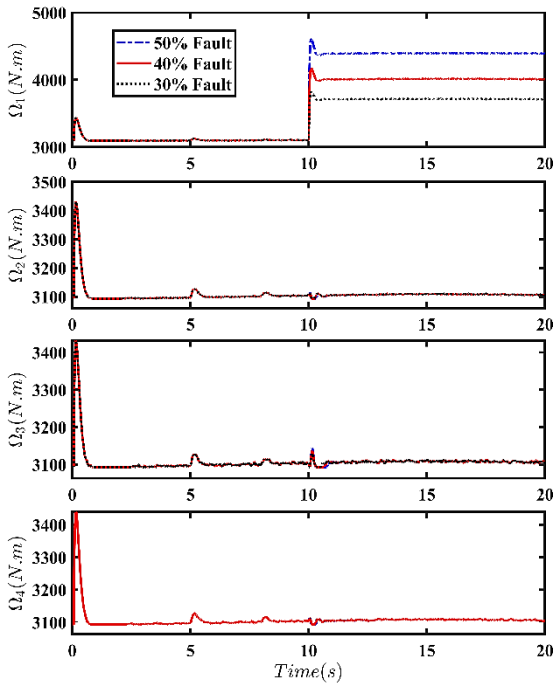
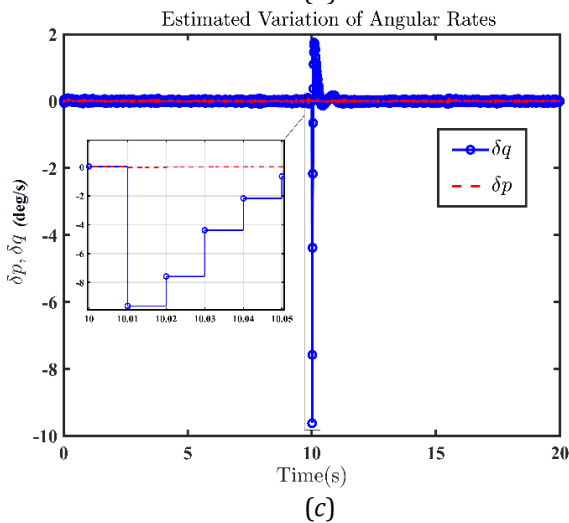


Fig. 8 Effect of fault detection time delay on the performance of the controller

6. CONCLUSION

This paper presents an autonomous emergency landing architecture (AELA) for the recovery and safe landing of a drone in presence of the engine fault. Inside the AELA, a fault-tolerant trajectory tracking controller including a fault detection algorithm and a three-loop adaptive sliding mode controller is proposed. The fault detection algorithm uses the filtered angular rates based on the extended Kalman filter and the autopilot commands to detect and identify the magnitude of partial loss of propeller effectiveness. The simulation results demonstrate that the adaptive controller without fault detection algorithm is robust to 10% of propeller damage but with considerable degradation of performance. Therefore, the threshold values which determine the occurrence of a fault is defined based on 5% of damage in which the controller performance without fault detection



algorithm is satisfactory. The simulation results demonstrate that the proposed controller has satisfactory performance considering rotor damages up to 50%. For more severe damages, the engine thrust saturates and trajectory tracking control cannot be achieved. In addition, the simulation results demonstrate the effect of time-delay in the fault detection process on the overall controller performance and therefore the effectiveness of the proposed fast converging fault detection algorithm. In future research, the proposed effective algorithm and AELA structure will be implemented on the drone in flight tests to verify the performance of the algorithms in different fault scenarios.

ACKNOWLEDGEMENT

This research was supported by Scientific Research Project Unit (BAP) of Adana Alparslan Türkeş Science and Technology University with the project number of 19119002.

Conflicts of interest

The authors declare no conflicts of interest.

REFERENCES

- Ahmadi K, Asadi D, Pazooki F (2017). Nonlinear L1 adaptive control of an airplane with structural damage, Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering, 233(1), 341-353; 0954410017730088.
- Amoozgar M H, Chamseddine A & Zhang Y (2013). 'Experimental test of a two-stage Kalman filter for actuator fault detection and diagnosis of an unmanned quadrotor helicopter. J. Intell. Robot. Syst., vol. 70, pp. 107–117, Apr. 2013.
- Asadi D & Ahmadi K (2020). Nonlinear Robust adaptive control of an airplane with structural damage, Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering.
- Asadi D & Bagherzadeh S (2017a). Nonlinear adaptive sliding mode tracking control of an airplane with wing damage, Proceedings of the Institution of Mechanical Eng., Part G: Journal of Aerospace Engineering, 232 (8), 1405-1420.
- Asadi D, Atkins E M (2017b). Multi-Objective Weight Optimization for Trajectory Planning of an Airplane with Structural Damage. Journal of Intelligent & Robotic Systems.
- Asadi D, Sabzehparvar M & Talebi H A (2013). Damaged airplane flight envelope and stability evaluation. Aircraft Engineering and Aerospace Technology, Vol. 85, No. 3, pp. 186-198. Doi: 10.1108/00022661311313623.
- Asadi D, Sabzehparvar M, Atkins E M & Talebi H A (2014). Damaged Airplane Trajectory Planning based on Flight Envelope and Stability of Motion Primitives. Journal of Aircraft, Vol. 51, No. 6, pp. 1740-1757. doi: 10.2514/1.C032422.
- Avram R C, Zhang X & Muse J (2017). 'Quadrotor actuator fault diagnosis and accommodation using nonlinear adaptive estimators. IEEE Trans. Control Syst. Technol., 25(6), 2219–2226.
- Barghandan S, Badamchizadeh M A & Jahed-Motlagh, M R (2017). Improve Adaptive Fuzzy Sliding Mode Controller for Robust Fault-tolerant of a Quadrotor. Int. J. Control Autom. Syst., 15, 427–441.
- Cen Z, Noura H, Susilo B T & Younes Y A (2014). Robust fault diagnosis for quadrotor UAVs using adaptive Thau observer," J. Intell. Robot. Syst., 73(1), 573–588.
- Chen F, Wu Q, Jiang B & Tao G (2015). A reconfiguration scheme for quadrotor helicopter via simple adaptive control and quantum logic. IEEE Trans. on Industrial Electronics, 62 (7), 4328–4335.
- Desaraju V R, Michael N, Humenberger M, Brockers R, Weiss S, Nash J Y (2015). Vision-based landing site evaluation and informed optimal trajectory generation toward autonomous rooftop landing, Auton Robot 39:445–463.
- Frangenberg M, Stephan J & Fichter W (2015). Fast Actuator Fault Detection and Reconfiguration for Multicopters (AIAA 2015-1766). In Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, 2015.
- Gao Z, Carlo Cecati & Steven X D (2015). A Survey of Fault Diagnosis and Fault-Tolerant Techniques Part I: IEEE Transactions on Industrial Electronics, 62, 6.
- Giribet J I, Pose C D, Ghersin A S & Mas I (2018) Experimental Validation of a Fault-tolerant Hexacopter with Tilted Rotors International Journal of Electrical and Electronic Engineering & Telecommunications, 7(2), 58-65.
- Github PX4 Documentation, <https://github.com/kylemanna/px4firmware/tree/master/Documentation>.
- Han W, Wang Z & Yi S (2018). Fault estimation for a quadrotor unmanned aerial vehicle by integrating the parity space approach with recursive least squares. Proc. Inst. Mech. Eng. G, J. Aerosp. Eng., 232(4) 783–796.
- Khebbache H, Sait B, Yacef F & Soukkou Y (2012). Robust stabilization of a quadrotor aerial vehicle in presence of actuator faults. International Journal of Information Technology, Control and Automation, 2(2), 1–13.
- Lanzon A, Freddi A & Longhi S (2014). Flight Control of a Quadrotor Vehicle Subsequent to a Rotor Failure. Journal of Guidance, Control, and Dynamics, 37:580–591.
- Lopez-Franco C, Gomez-Avila J, Alanis A Y, Arana-Daniel, N & Villasenor C (2017). Visual Servoing for an autonomous Hexarotor Using a Neural Network Based PID Controller, Sensors, 17, 1865; doi:10.3390/s17081865.
- Mazeh H, Saied M, Shraim H & Francis C (2018). Fault-tolerant control of an hexarotor unmanned aerial vehicle applying outdoor tests and experiments. Int. Fed. Autom. Control, 51, 312–317.
- Merheb A R (2016). Diagnostic and Fault-Tolerant Control Applied to an Unmanned Aerial Vehicle, Aix Marseille University, Ph.D. Thesis.
- Milton C P S, Claudio R, Jorge S, Mario S-F & Ricardo C, (2019) An Adaptive Dynamic Controller for

- Quadrotor to Perform Trajectory Tracking Tasks, *J. of Intel. Robotic Systems*, 93, 5–16.
- Mohammadi M & Shahri A M (2013). Adaptive nonlinear stabilization control for a quadrotor UAV: theory, simulation and experimentation. *J. Intell. Robot. Syst.* 72(1), 105–122.
- Mueller, M W & Andrea R D (2014). Stability and control of a quadcopter despite the complete loss of one, two, or three propellers. *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 45–52
- Nguyen N P, Xuan Mung N & Hong S K (2019). Actuator Fault Detection and Fault-Tolerant Control for Hexacopter. *Sensors*, 19, 4721; doi: 10.3390/s19214721.
- Ranjbaran M, Khorasani K (2010). Fault recovery of an under-actuated quadrotor aerial vehicle. *49th IEEE Conference on Decision and Control, Atlanta, GA*, 4385–4392.
- Saied M, Lussier B, Fantoni I, Francis C, Shraim H & Sanahuja G (2015) "Fault diagnosis and fault-tolerant control strategy for rotor failure in an octarotor," in *Proc. IEEE ICRA*.
- Sharifi F, Mirzaei M, Gordon, B W & Zhang Y M (2010). Fault-tolerant control of a quadrotor UAV using sliding mode control. *Proceedings of the Conference on Control and Fault-tolerant Systems, Nice, France*, 6–7 October 2010; pp. 239–244.
- Tarhan M & Altug E (2011). EKF Based Attitude Estimation and Stabilization of a Quadrotor UAV Using Vanishing Points in Catadioptric images. *J. Intell Syst*, Vol. 62, 2011, pp. 587-607 DOI 10.1007/s10846-010-9459-y.
- Zhang Y & Chamseddine A (2012). Fault-tolerant flight control techniques with application to a quadrotor UAV testbed. *Automatic Flight Control Systems - Latest Developments*, T. Lombaerts, Ed. *InTech*, 2012, pp. 119–150.



© Author(s) 2021.

This work is distributed under <https://creativecommons.org/licenses/by-sa/4.0/>