FAULT-TOLERANT CONTROL OF A QUADROTOR DESPITE THE COMPLETE ROTOR FAILURE

Ali Jebelli,* Alireza Najafiyanfar,** Arezoo Mahahadi,*** and Mustapha C. E. Yagoub****

Abstract

In this paper, a control algorithm based on the Lyapunov stability theory and neural network adaptive scheme is proposed to efficiently regulate the position, attitude, and altitude of a quadrotor through a nonlinear dynamic model. Based on the Lyapunov stability theory, the controller allows the system to continue its task correctly even if one or two rotors of the quadrotor stop working and this is achieved without losing stability. Also, in the presence of parametric uncertainties, the coefficients of the controller are adaptively tuned by the neural network method. The obtained results demonstrated the proper performance of the control algorithm based on different operating conditions and scenarios. In fact, the obtained results demonstrated that the proposed controller exhibits desirable transient behaviour and performance stability. Therefore, for operational purposes where the stability and continuation of the quadrotor mission in case of rotor failure is very important, using the controller proposed in this research is very efficient. The proposed control algorithm is easy to implement, compatible with existing quadrotors, and does not significantly affect the overall energy consumption.

Key Words

UAV, fault tolerance, quadrotor, Lyapunov, flying robot

1. Introduction

Extensive research has been done about unmanned aerial vehicles (UAVs) such as quadrotors; many companies have invested in this field and a significant number of academic and industrial projects have been reported, from which we can state those focussing on the control of the stability and position of UAVs during their operation using different control approaches, such as sliding modeling technique [1], backstepping and/or adaptive methods [2], robust PID [3], [4], and linear quadratic Gaussian (LQG) control [5]. On the other hand, fault-tolerant control (FTC) issues have been arisen from defects in the UAV sensors, motors, or other segments [6]. FTC systems can be classified into passive and active [7]. Whenever there is a fault in a passive FTC system, the control structure does not change, and the control system is resistant to the faults [8], whereas, if there is a fault in an active FTC system, the control system will be reset [8]. In FTC systems, different methods have been explored to design the controllers in the case of rotor failure. The sliding method has been used in [9] and [10] to control the operating conditions in the case of rotor failure and disturbance, while authors in [11] and [12] used model predictive control to control the system. Robust adaptive control has been used in [13] to track the altitude and control the quadrotor attitude. In [14], a nonlinear discrete adaptive algorithm and a PID algorithm [15], have been used, respectively, within inner and outer loops to control the path tracking. Optimisation methods have been used in [16] to minimise the forces applied by the rotors in the case of their failure. Smart control methods, such as reinforcement learning [17], have been used for FTC. Fuzzy logic algorithm has been used in [18] to control a multirotor UAV; furthermore, it has been shown that the control algorithm has an adequate performance even in the case of two rotors failure. Usually, whenever the rotor fails, the controller is not capable of controlling one of the variables of roll, pitch, and yaw; needless to point that, yaw often ignored, *i.e.*, roll, pitch, and altitude are the only controlled variables. In fact, controlling the roll and pitch is of utmost importance since any small change in roll and pitch angles will cause the system to lose its stability; preventing the quadrotor from hitting the ground is the reason for altitude control [8]. However, lack of yaw control in the case of rotor failure will cause the quadrotor not to be capable of completing its task, that is to say, just to be able to have emergency landing. As an example, whenever the yaw value is not controlled in the case of rotor failure in quadrotors with cameras, the issue will lead to inability of imaging; on this basis, the yaw value should be controlled if the quadrotor aims to continue its task in spite of router failure.

In this research, a control algorithm based on the Lyapunov stability theory is designed to efficiently control

* RoboticC Inc., Ottawa, ON, Canada
** K.N. Toosi University of Technology, Tehran, Iran
*** Engineering Department, RoboticC Inc., Ottawa, ON, Canada
**** University of Ottawa, Ottawa, ON, Canada
Corresponding author: Ali Jebelli

Recommended by Jason Gu
(DOI: 10.2316/J.2024.206-0940)

the attitude, position, and altitude of the quadrotor. The designed controller is divided into four sub-controllers: attitude controller, altitude controller, position controller, and controller for calculating Euler angles. The proposed algorithm is a combination of the Lyapunov stability theory and the neural network adaptive scheme, in which the coefficients of each controller are adaptively tuned by the neural network method. The control algorithm is robust to the failure of the actuators, and if one or two rotors fail, the quadrotor can perform its tasks using only the remaining rotors. In fact, in case of failure of the rotors, the controller by adjusting the rotor angular velocity and the desired Euler angles causes the system to track the desired path without losing stability. This research includes mathematical modelling of the quadrotor and statement of the problem, design details of the controllers, including altitude, attitude, status, and the controller for calculating the desired Euler angles, and at the end, the results of the simulation, comparison, and conclusion are presented.

2. Mathematical Model

Quadrotor is an UAV with six degrees of freedom. It accounts for two pairs of rotors which rotate in opposite directions. The dynamical model of a given quadrotor UAV is presented in Fig. 1, where the state vector $[x, y, z]$ denotes the position of the center of the gravity of the quadrotor and the vector $[\dot{x}, \dot{y}, \dot{z}]$ denotes its linear velocity in the body frame; the three Euler angles $[\varphi, \theta, \psi]$ state for the roll, the pitch, and the yaw, respectively, while $[\dot{\varphi}, \dot{\theta}, \dot{\psi}]$ refers to its angle velocity in the body frame. The dynamic equations are set by defining the ground frame and the body frame [8]:

$$\begin{aligned}
\ddot{x} &= (\cos(\varnothing)\cos(\psi)\sin(\theta) + \sin(\varnothing)\sin(\psi)) \\
&\quad \frac{lk_f}{m}\left(\omega_1^2 + \omega_2^2 + \omega_1^2 + \omega_4^2\right) && (1)\\
\ddot{y} &= (\cos(\psi)\sin(\varnothing) - \cos(\varnothing)\sin(\psi)\sin(\theta)) \\
&\quad \frac{lk_f}{m}\left(\omega_1^2 + \omega_2^2 + \omega_1^2 + \omega_4^2\right) && (2)\\
\ddot{z} &= g - \cos(\varnothing)\sin(\theta)\frac{lk_f}{m}\left(\omega_1^2 + \omega_2^2 + \omega_1^2 + \omega_4^2\right) && (3)\\
\ddot{\varphi} &= \frac{(I_{B,xx} - I_{B,zz})}{I_{B,xx}}\dot{\theta}\dot{\psi} - \frac{K_{d,xx}\dot{\varphi}}{I_{B,xx}}\sqrt{\dot{\varphi}^2 + \dot{\theta}^2 + \dot{\psi}^2} \\
&\quad + \frac{lk_f}{I_{B,xx}}\left(\omega_2^2 - \omega_4^2\right) - \frac{\dot{\theta}I_{p,zz}}{I_{B,xx}}(\omega_1 + \omega_2 + \omega_3 + \omega_4) && (4)\\
\ddot{\theta} &= \frac{(I_{B,zz} - I_{B,xx})}{I_{B,xx}}\dot{\varphi}\dot{\psi} - \frac{K_{d,xx}\dot{\theta}}{I_{B,xx}}\sqrt{\dot{\varphi}^2 + \dot{\theta}^2 + \dot{\psi}^2} \\
&\quad + \frac{lk_f}{I_{B,xx}}\left(\omega_1^2 - \omega_3^2\right) + \frac{\dot{\varphi}I_{p,zz}}{I_{B,xx}}(\omega_1 + \omega_2 + \omega_3 + \omega_4) && (5)\\
\ddot{\psi} &= -\frac{K_{d,zz}\dot{\psi}}{I_{B,zz}}\sqrt{\dot{\varphi}^2 + \dot{\theta}^2 + \dot{\psi}^2} \\
&\quad + \frac{k_\tau}{I_{B,zz}}\left(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2\right) && (6)
\end{aligned}$$

Here, m denotes the total mass, g the acceleration of gravity, and l the distance from the center of each rotor to the center of gravity. $\omega_1, \omega_2, \omega_3$, and ω_4 stand for the angular speed of the propeller. From that, the control inputs u_1, u_2, u_3, and u_4 can be calculated by the following matrix system

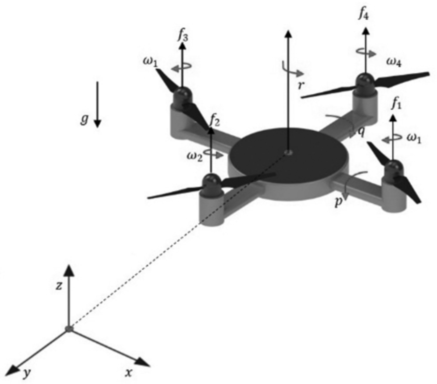


Figure 1. Definition of quadrotor body frame and rotor indices.

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -l & 0 & l & 0 \\ 0 & l & 0 & -l \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_1^2 \end{bmatrix} \tag{7}$$

For the quadrotor dynamic model given by (1)–(6), the following assumptions are made:

- The quadrotor structure is symmetric and rigid.
- The origin of the body frame and the center of gravity are the same.
- The axes of the body frame are coincident with the quadrotor inertia axes.

3. Statement of the Problem

Equations (1)–(6) have been reformatted as below for designing the controller.

$$\begin{aligned}
\ddot{x} &= (\cos(\varnothing)\cos(\psi)\sin(\theta) + \sin(\varnothing)\sin(\psi))\,u_1 && (8)\\
\ddot{y} &= (\cos(\psi)\sin(\varnothing) - \cos(\varnothing)\sin(\psi)\sin(\theta))\,u_1 && (9)\\
\ddot{z} &= g - \cos(\varnothing)\sin(\theta)\,u_1 && (10)\\
\ddot{\varphi} &= a_1\dot{\theta}\dot{\psi} - c_1\dot{\varphi} + u_2 - b_1\,\dot{\theta}\,\omega_r && (11)\\
\ddot{\theta} &= -a_1\dot{\varphi}\dot{\psi} - c_1\dot{\theta}\ + u_3 + b_1\,\dot{\varphi}\omega_r && (12)\\
\ddot{\psi} &= -c_2\dot{\psi} + u_4 && (13)
\end{aligned}$$

with

$$\begin{aligned}
u_1 &= \frac{lk_f}{m}\left(\omega_1^2 + \omega_2^2 + \omega_1^2 + \omega_4^2\right),\ u_2 = \frac{lk_f}{I_{B,xx}}\left(\omega_2^2 - \omega_4^2\right) \\
u_3 &= \frac{lk_f}{I_{B,xx}}\left(\omega_1^2 - \omega_3^2\right), \\
u_4 &= \frac{k_\tau}{I_{B,zz}}\left(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2\right) && (14)\\
a_1 &= \frac{(I_{B,xx} - I_{B,zz})}{I_{B,xx}},\quad \mathrm{b}_1 = \frac{I_{p,zz}}{I_{B,xx}} && (15)\\
c_1 &= \frac{K_{d,xx}}{I_{B,xx}}\sqrt{\dot{\varphi}^2 + \dot{\theta}^2 + \dot{\psi}^2},
\end{aligned}$$

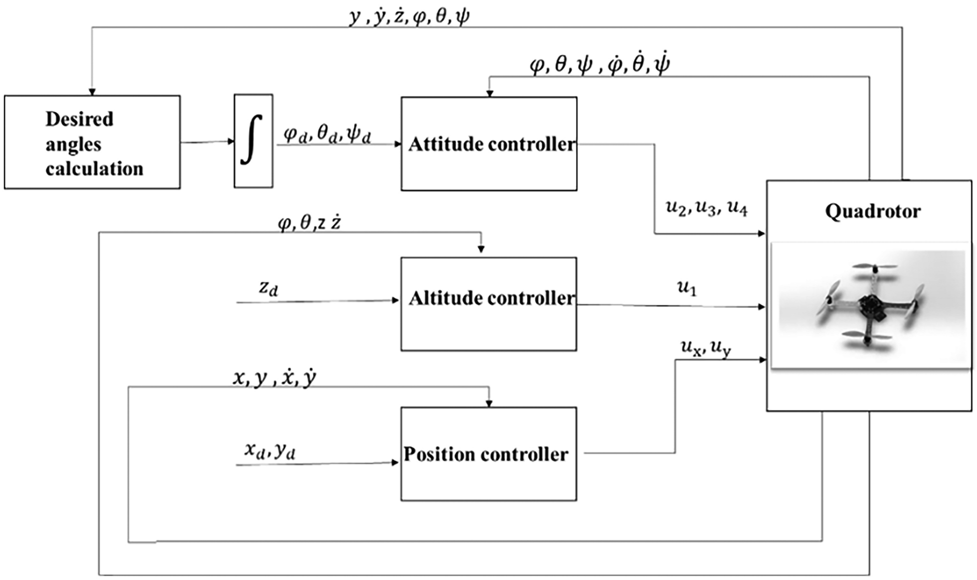

$y, \dot{y}, \dot{z}, \varphi, \theta, \psi$
$\varphi, \theta, \psi, \dot{\varphi}, \dot{\theta}, \dot{\psi}$
Desired angles calculation
$\int$
$\varphi_d, \theta_d, \psi_d$
Attitude controller
u_2, u_3, u_4
Quadrotor
$\varphi, \theta, z\, \dot{z}$
z_d
Altitude controller
u_1
u_x, u_y
$x, y, \dot{x}, \dot{y}$
x_d, y_d
Position controller

Figure 2. Control system design.

$$c_2 = \frac{K_{d,zz}}{I_{B,zz}} \sqrt{\dot{\varphi}^2 + \dot{\theta}^2 + \dot{\psi}^2} \tag{16}$$

$$\omega_r = (\omega_1 + \omega_2 + \omega_3 + \omega_4) \tag{17}$$

$\left[x, \dot{x}, y, \dot{y}, z, \dot{z}, \varphi, \dot{\varphi}, \theta, \dot{\theta}, \psi, \dot{\varphi}\right]$ is the system state vector and $[u_1, u_2, u_3, u_4]$ the vector of the system control inputs.

4. Controller Design

Figure 2 shows the structure of the quadrotor controller which includes attitude, altitude, and position controllers. The purpose of the controller design is to track the desired trajectories $[x_d, y_d, z_d, \varphi_d, \theta_d, \psi_d]$.

4.1 Altitude Control

Lyapunov function will be set as follows to control the quadrotor altitude and the convergence of the system to its desired value.

$$V_h = \frac{1}{2}\left[(z - z_d)^2 + \dot{z}^2\right] \tag{18}$$

Its first derivative can be expressed as

$$\dot{V}_h = \dot{z}\,(z - z_d) + \ddot{z}\dot{z} \tag{19}$$

Substituting (10) into (19) gives

$$\dot{V}_h = \dot{z}\,(z - z_d) + \dot{z}g - \cos(\varphi)\cos(\theta)\,\dot{z}u_1 \tag{20}$$

u_1 should be then selected as follows to keep the Lyapunov stability conditions

$$u_1 = \frac{(z - z_d) + g + k_z\dot{z}}{\cos(\varphi)\cos(\theta)} \tag{21}$$

By maintaining the condition $k_z > 0$, the stability condition of the Lyapunov function is satisfied:

$$\dot{V}_h = -k_z\dot{z}^2 < 0 \tag{22}$$

The parameter k_z can be determined based on the requirements for the steady-state tracking precision and the convergence speed of control. A neural network is deployed to adjust the coefficient k_z with the adaptation law derived from the conventional backpropagation algorithm. The neural network is trained by the specialised learning architecture [19] to minimise the performance error E:

$$E = \frac{1}{2}(z_d - z)^2 \tag{23}$$

Based on the gradient descent method [20], we have the following adaptation equation:

$$k_z = k_{z0} - \varepsilon_z \int_0^t \frac{\partial E}{\partial k_z} \mathrm{dt} \tag{24}$$

where ε_z is the learning rate which determines the convergence speed of neural network and k_{z0} the initial value of k_z. Using the chain rule:

$$\begin{aligned} \frac{\partial E}{\partial k_z} &= \frac{\partial E}{\partial z}\frac{\partial z}{\partial u_1}\frac{\partial u_1}{\partial k_z} \\ &= -(z_d - z)\,\frac{\partial z}{\partial u_1}\left(\frac{\dot{z}}{\cos(\varphi)\cos(\theta)}\right) \end{aligned} \tag{25}$$

Assuming $\frac{\partial z}{\partial u_1} = sign(\frac{\nabla z}{\nabla u_1})$ [19], we have:

$$\begin{aligned} k_z = k_{z0} + \varepsilon_z \int_0^t (z_d - z)\; sign\left(\frac{\nabla z}{\nabla u_1}\right) \\ \left(\frac{\dot{z}}{\cos(\varphi)\cos(\theta)}\right)\; \mathrm{dt} \end{aligned} \tag{26}$$

where ∇ is called the ascending or backward differences operator, such as $\nabla\, h_k = h_k - h_{k-1}$.

4.2 Attitude Control

Lyapunov function is based on the roll, pitch, and yaw variables to control the quadrotor attitude and the convergence of the roll, pitch, and yaw to their desired values.

$$\begin{aligned} V_A &= \frac{1}{2}\Big[(\varphi-\varphi_d)^2+\dot{\varphi}^2+(\theta-\theta_d)^2+\dot{\theta}^2 \\ &\quad +(\psi-\psi_d)^2+\dot{\psi}^2\Big] \end{aligned} \tag{27}$$

The derivative of the Lyapunov function is obtained as follows:

$$\begin{aligned} \dot{V}_A &= \dot{\varphi}(\varphi-\varphi_d)+\dot{\varphi}\ddot{\varphi}+\dot{\theta}(\theta-\theta_d)+\dot{\theta}\ddot{\theta} \\ &\quad +\dot{\psi}(\psi-\psi_d)+\dot{\psi}\ddot{\psi} \end{aligned} \tag{28}$$

Substituting (11)–(13) into (28) leads to

$$\begin{aligned} \dot{V}_A &= \dot{\varphi}\,(\varphi-\varphi_d)-c_1\dot{\varphi}^2+\dot{\varphi}\;u_2+\dot{\theta}(\theta-\theta_d)-c_1\dot{\theta}^2 \\ &\quad +\dot{\theta}\;u_3+\dot{\psi}(\psi-\psi_d)-c_2\dot{\psi}^2+\dot{\psi}\;u_3 \end{aligned} \tag{29}$$

u_2, u_3, and u_4 should be selected as follows for keeping the Lyapunov stability conditions.

$$\begin{aligned} u_2 &= -(\varphi-\varphi_d)-k_\varphi\dot{\varphi} \\ u_3 &= -(\theta-\theta_d)-k_\theta\dot{\theta} \\ u_4 &= -(\psi-\psi_d)-k_\psi\dot{\psi} \end{aligned} \tag{30}$$

By maintaining the condition $k_\varphi > 0, k_\theta > 0$ and $k_\psi > 0$, the stability condition of the Lyapunov function is satisfied:

$$\dot{V}_A = -c_1\dot{\varphi}^2-k_\varphi\dot{\varphi}^2-c_1\dot{\theta}^2-k_\theta\dot{\theta}^2-c_2\dot{\psi}^2-k_\psi\dot{\psi}^2<0 \tag{31}$$

Correspondingly, the neural network is used to adjust the positive coefficients k_φ, k_θ, and k_ψ, and the following results are obtained:

$$\begin{aligned} k_\varphi &= k_{\varphi 0}+\varepsilon_\varphi\int_0^t(\varphi_d-\varphi)\;\mathrm{sign}\left(\frac{\nabla\varphi}{\nabla u_2}\right)(-\dot{\varphi})\;\mathrm{dt} && (32) \\ k_\theta &= k_{\theta 0}+\varepsilon_\theta\int_0^t(\theta_d-\theta)\;\mathrm{sign}\left(\frac{\nabla\theta}{\nabla u_3}\right)\left(-\dot{\theta}\right)\;\mathrm{dt} && (33) \\ k_\psi &= k_{\psi 0}+\varepsilon_\psi\int_0^t(\psi_d-\psi)\,\mathrm{sign}\left(\frac{\nabla\psi}{\nabla u_4}\right)\left(-\dot{\psi}\right)\;\mathrm{dt} && (34) \end{aligned}$$

where $\varepsilon_\varphi, \varepsilon_\theta$, and ε_ψ are the respective learning rates that determine the convergence speed of neural network, and $k_{\varphi 0}$, $k_{\theta 0}$, and $k_{\psi 0}$ are the initial values of k_φ, k_θ, and k_ψ, respectively.

4.3 Position Control

Lyapunov function is defined as below to control the quadrotor position and the convergence of x and y to their desired values.

$$V_p = \frac{1}{2}\left[(x-x_d)^2+\dot{x}^2+(y-y_d)^2+\dot{y}^2\right] \tag{35}$$

with its derivative equals to

$$\dot{V}_p = \dot{x}(x-x_d)+\dot{x}\ddot{x}+\dot{y}(y-y_d)+\dot{y}\ddot{y} \tag{36}$$

Rewriting the acceleration components in the horizontal plane

$$\ddot{x} = u_x u_1,\ \ddot{y} = u_y u_1 \tag{37}$$

where

$$\begin{aligned} u_x &= \cos(\varphi)\cos(\psi)\sin(\theta)+\sin(\varphi)\sin(\psi) && (38) \\ u_y &= \cos(\psi)\sin(\varphi)-\cos(\varphi)\sin(\psi)\sin(\theta) && (39) \end{aligned}$$

Assuming small Euler angles leads to:

$$u_x = \theta_d\ ,\ u_y = \varphi_d \tag{40}$$

and substituting (37) into (36) gives

$$\dot{V}_p = \dot{x}\,(x-x_d)+\dot{x}u_1u_x+\dot{y}\,(y-y_d)+\dot{y}u_1u_y \tag{41}$$

u_x and u_y are selected as follows to keep the Lyapunov stability conditions.

$$\begin{aligned} u_x &= \frac{-(x-x_d)-k_x\dot{x}}{u_1} && (42) \\ u_y &= \frac{-(y-y_d)-k_y\dot{y}}{u_1} && (43) \end{aligned}$$

By maintaining the condition $k_x > 0$ and $k_y > 0$, the stability condition of the Lyapunov function is satisfied:

$$\dot{V}_p = -k_x\dot{x}^2-k_y\dot{y}^2<0 \tag{44}$$

Correspondingly, the neural network is used to adjust the positive coefficients k_x and k_y:

$$\begin{aligned} k_x &= k_{x0}+\varepsilon_x\int_0^t(x_d-x)\;\mathrm{sign}\left(\frac{\nabla x}{\nabla u_x}\right)\left(-\frac{\dot{x}}{u_1}\right)\;\mathrm{dt} && (45) \\ k_y &= k_{y0}+\varepsilon_y\int_0^t(y_d-y)\;\mathrm{sign}\left(\frac{\nabla y}{\nabla u_y}\right)\left(-\frac{\dot{y}}{u_1}\right)\;\mathrm{dt} && (46) \end{aligned}$$

where ε_x and ε_y are the respective learning rates that determine the convergence speed of the neural network, and k_{x0} and k_{y0} the initial values of k_x and k_y, respectively.

4.4 Desired Angle Calculation

Based on the Lyapunov method, the acceleration vector is calculated as [8]:

$$a^* = -(s-s_d)-g-k_1\dot{s} \tag{47}$$

a is the sum of the acceleration caused by the rotor's forces and a^* shows the desired value of this vector. Therefore, the coordinate system will be set such that z-axis corresponds to a^*; then the angular velocity of this system is calculated as:

$$\omega^{\mathrm{CE}} = \begin{bmatrix} \dot{\beta}_3\sin(\beta_2) \\ -\dot{\beta}_2 \\ \dot{\beta}_3\cos(\beta_2) \end{bmatrix} \tag{48}$$

Moreover:

$$\begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \omega^{\mathrm{BC}}+\omega^{\mathrm{CE}} \tag{49}$$

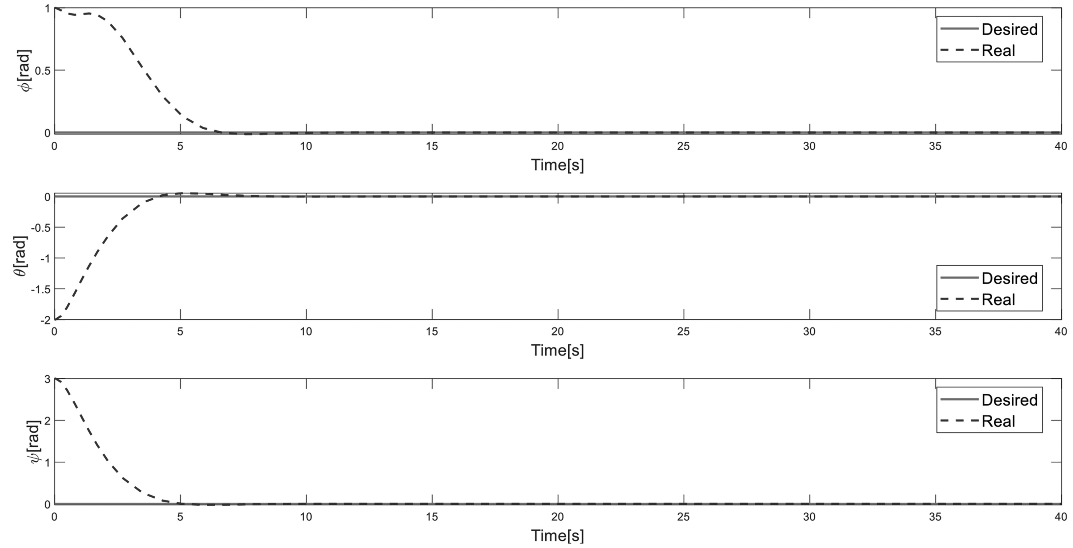


Figure 3. Euler angles.

$$\begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \dot{\beta}_3 \sin(\beta_2) \\ -\dot{\beta}_2 \\ \dot{\beta}_3 \cos(\beta_2) \end{bmatrix} + \begin{bmatrix} \cos(\alpha_3) & \cos(\alpha_1)\sin(\alpha_3) & 0 \\ -\sin(\alpha_3) & \cos(\alpha_1)\cos(\alpha_3) & 0 \\ 1 & -\sin(\alpha_1) & 1 \end{bmatrix} \begin{bmatrix} \dot{\alpha}_1 \\ \dot{\alpha}_2 \\ \dot{\alpha}_3 \end{bmatrix} \tag{50}$$

Selecting $\dot{\alpha}_1$, $\dot{\alpha}_2$, and $\dot{\alpha}_3$ as

$$\begin{cases} \dot{\alpha}_1 = -k_{\alpha_1}\alpha_1 \\ \dot{\alpha}_2 = -k_{\alpha_2}\alpha_2 \\ \dot{\alpha}_3 = 0 \end{cases} \tag{51}$$

leads to the desired value of the $\begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$ vector as:

$$\begin{bmatrix} \dot{\varphi}_d \\ \dot{\theta}_d \\ \dot{\psi}_d \end{bmatrix} = \begin{bmatrix} \dot{\beta}_3 \sin(\beta_2) \\ -\dot{\beta}_2 \\ \dot{\beta}_3 \cos(\beta_2) \end{bmatrix} + \begin{bmatrix} \cos(\alpha_3) & \cos(\alpha_1)\sin(\alpha_3) & 0 \\ -\sin(\alpha_3) & \cos(\alpha_1)\cos(\alpha_3) & 0 \\ 1 & -\sin(\alpha_1) & 1 \end{bmatrix} \begin{bmatrix} -k_{\alpha_1}\alpha_1 \\ -k_{\alpha_2}\alpha_2 \\ 0 \end{bmatrix} \tag{52}$$

5. Simulation and Tests Results for a Rotor Failure

5.1 System Parameters

In this section, the performance of the proposed controller is evaluated in the presence of parametric uncertainties and disturbances by simulation in the MATLAB software. The physical parameters of the quadrotor are set as follows: total mass $m = 0.5$ kg, gravitational acceleration $g = 9.81\ m/s^2$, distance from the center of each rotor to the center of the gravity of the quadrotor $l = 0.17\ m$, mass moments of inertia in the x, y, and z axes $I_B = \text{diag}([2.7, 2.7, 5.2])$, inertia of the propeller $I_{p,zz}$=1.5, drag coefficients $k_d = \text{diag}([0.7, 0.7, 1.4])$. In the numerical values of m, I_B, $I_{p,zz}$, and K_d parameters, 20% of uncertainty is considered. The purpose of the controller design is to track the following desired trajectory:

$$x_d = \begin{cases} 1 & \text{t} \ \leq \ 15 \text{ or } t \ > \ 25 \\ 0 & \text{otherwise} \end{cases},$$

$$y_d = \begin{cases} 1 & \text{t} \ \leq \ 10 \text{ or } t \ > \ 20 \\ 0 & \text{otherwise} \end{cases},$$

$$z_d = \begin{cases} 1 & \text{t} \ \leq \ 35 \\ 0 & \text{otherwise} \end{cases}$$

5.2 Simulation with All Rotors

5.2.1 Test 1 (deviation in Euler angles):

Euler angles and desired angular velocity were set to zero before starting this test. The results of the simulations are displayed for the initial deviation in Euler angles:

$$\varphi_0 = 1 \ , \theta_0 = -2, \psi_0 = 3, \ \dot{\varphi}_0 = 0 \ , \dot{\theta}_0 = 0, \dot{\psi}_0 = 0$$

As shown in Figs. 3 and 4, Euler angles and angular velocity have reached their desired value after an initial deviation of angles, *i.e.*, the system has been adequately stabilised.

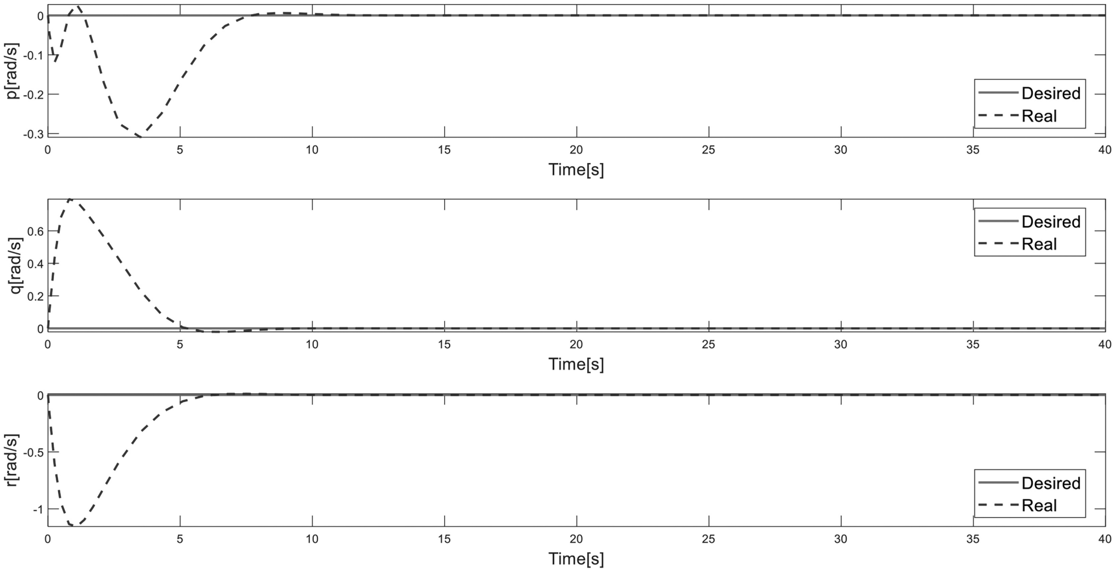


Figure 4. Angular velocity.

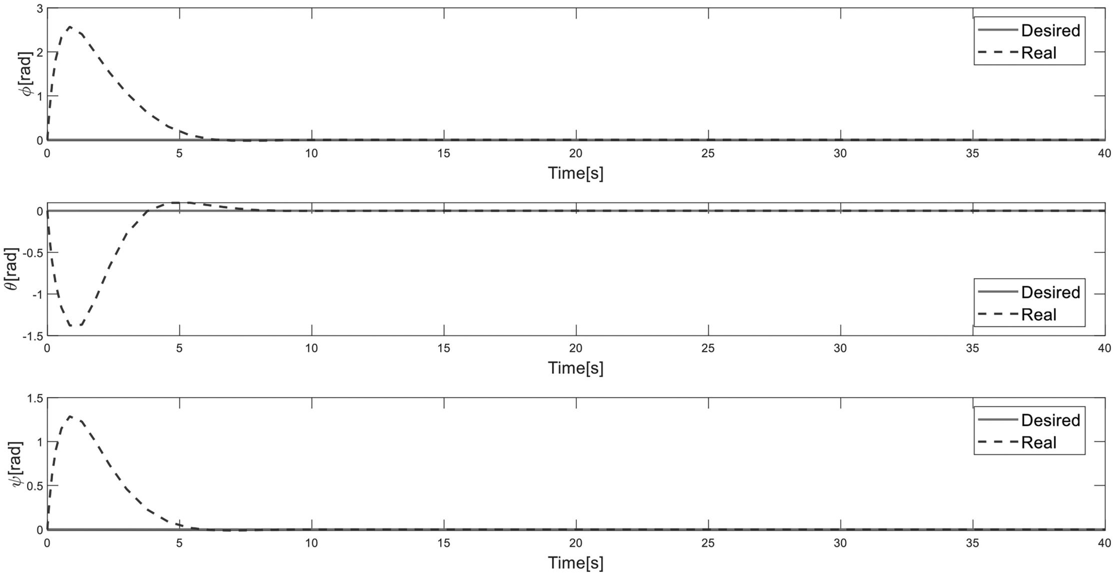


Figure 5. Euler angles.

5.2.2 Test 2 (Deviation in Angular Velocity)

Assuming that the vehicle body velocity is $\dot{\varphi}_0 = 10$, $\dot{\theta}_0 = -5, \dot{\psi}_0 = 5$. The simulation results shown in Figs. 5 and 6 indicate that the Euler angles and angular velocity have reached their desired value and the system has been stabilised.

5.2.3 Test 3 (Evaluating the Performance of the Altitude Controller Assuming a Rotor Failure)

In this test, the performance of the altitude controller was evaluated based on the following initial conditions $z_0 = 10$, $\dot{z}_0 = -5$. Also, the first rotor was turned off. The simulation results in Figs. 7 and 8 show that the system has reached its desired values and is stable despite the failure in one rotor.

5.2.4 Test 4 (Evaluating All the Integrated Controllers With a Rotor Failure)

In this test, we assumed a rotor failure and asked the quadrotor to still move along a rectangular route in the xy plane and return to its starting point. The desired route of this test has been defined as:

$$x_d = \begin{cases} 0 & \text{t} > 10 \\ 1 & \text{otherwise} \end{cases}, y_d = \begin{cases} 0 & \text{t} \leq 5 \text{ or } t > 15 \\ 1 & \text{otherwise} \end{cases}$$

As shown in Figs. 9 and 10, the quadrotor in fact moved along the predefined route even in the case of a rotor failure.

To further validate this test, we investigated the angular velocity of the four rotors (the first one being

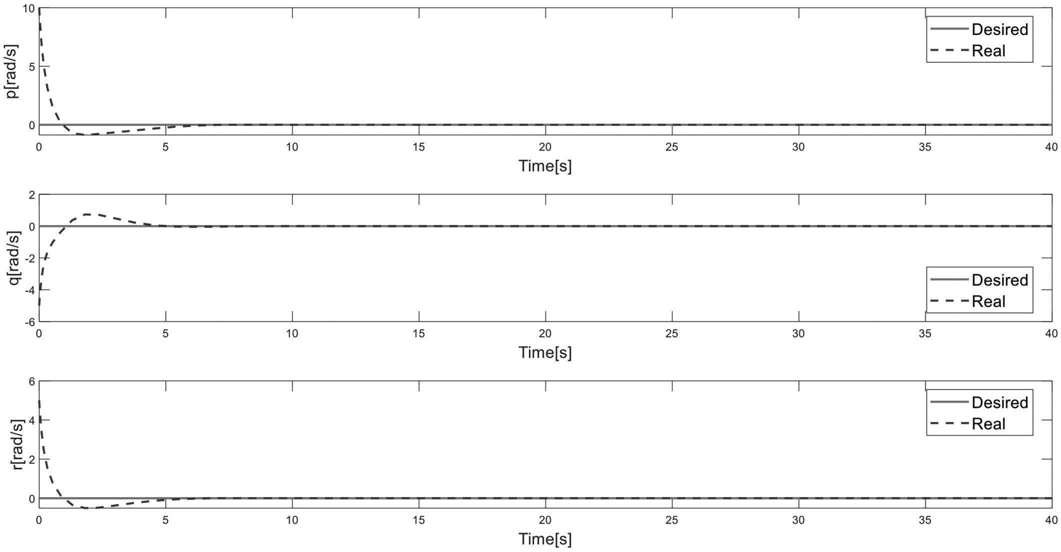


Figure 6. Angular velocity.

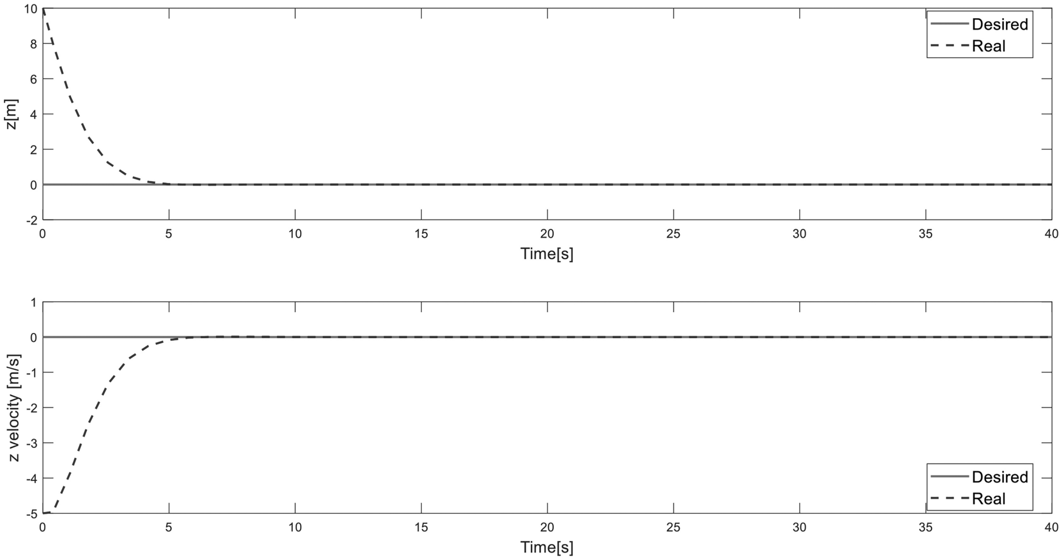


Figure 7. Altitude changes.

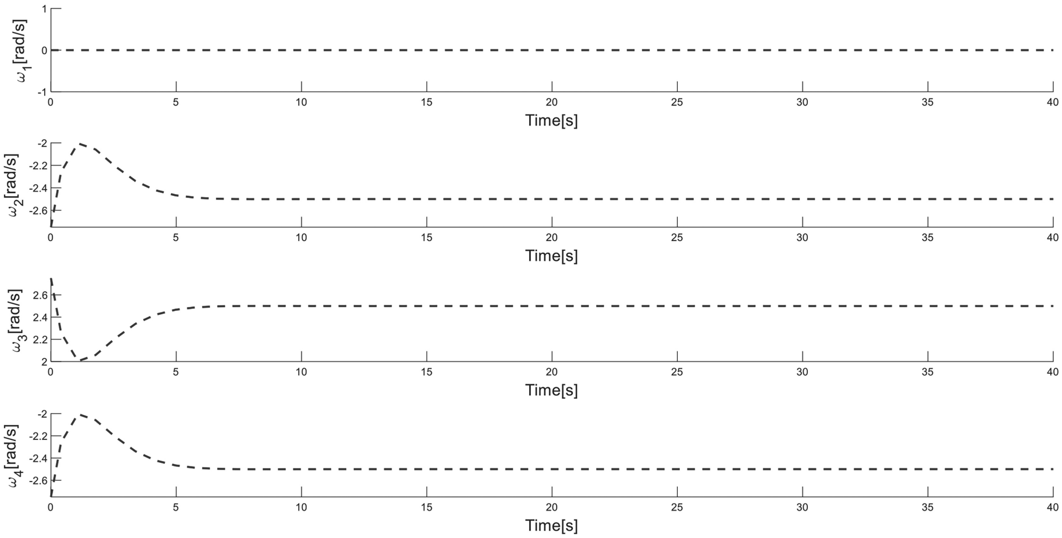


Figure 8. Rotor angular velocity.

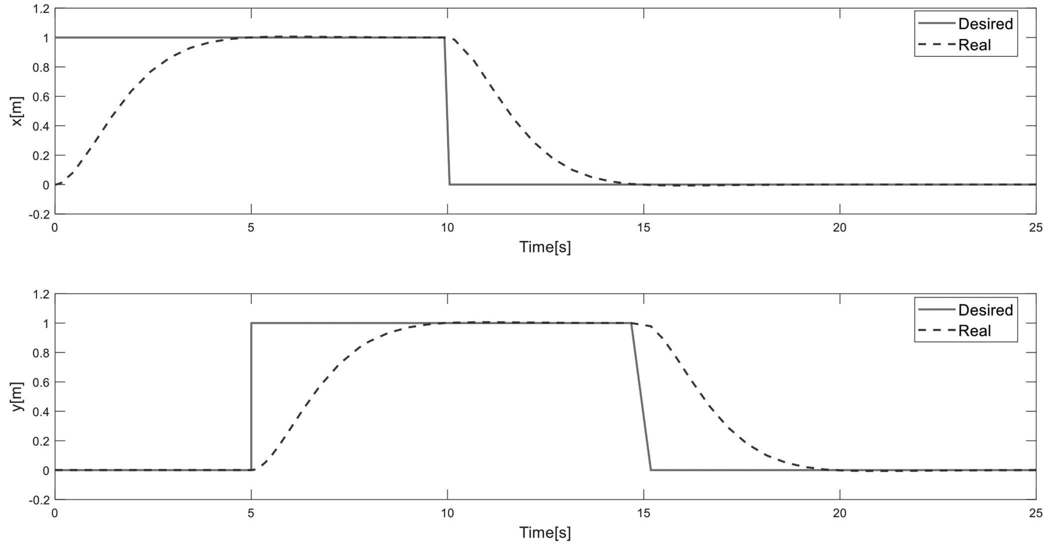


Figure 9. The system position.

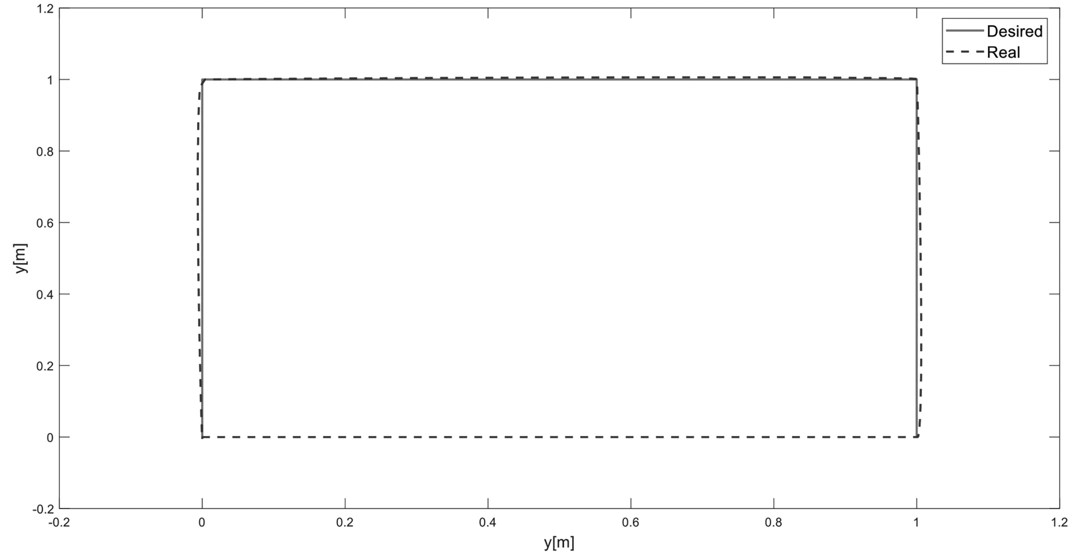


Figure 10. The system movement route in XY plane.

turned off). As shown in Fig. 11, the system velocity was stabilised.

5.3 Simulation with Two Rotor Failure

In this test, we assumed that two rotors failed. We then asked the quadrotor to move through the following defined route:

$$x_d = \begin{cases} 1 & \text{t} \leq 10 \text{ or } t > 20 \\ 0 & \text{otherwise} \end{cases},$$

$$y_d = \begin{cases} 1 & \text{t} \leq 15 \text{ or } t > 25 \\ 0 & \text{otherwise} \end{cases},$$

$$z_d = \begin{cases} 1 & \text{t} \leq 30 \\ 0 & \text{otherwise} \end{cases}$$

Two cases were considered to evaluate the controllers. First, the quadrotor does the task without the rotor failure. Second, two rotors of the quadrotor are turned off after 8 and 22 s, respectively. The simulation results in Fig. 12 show that the quadrotor does not lose its stability and completes its task with an acceptable performance in spite of two rotors failure.

Furthermore, Fig. 13 indicates that the quadrotor follows adequately the predetermined route. As shown in Fig. 12, the failure of the rotor does not change the position of the system and increase the error in the system, and the tracking of the desired value is done correctly. As for the angular velocity, Fig. 16, it demonstrated the good behaviour of the system even after the failure of two rotors (after, respectively, 8 and 22 s after the starting time).

The comparison of the simulation results is shown in Table 1 and as shown, the steady-state error in the state where the error occurs in the rotors is not much different from the state without error.

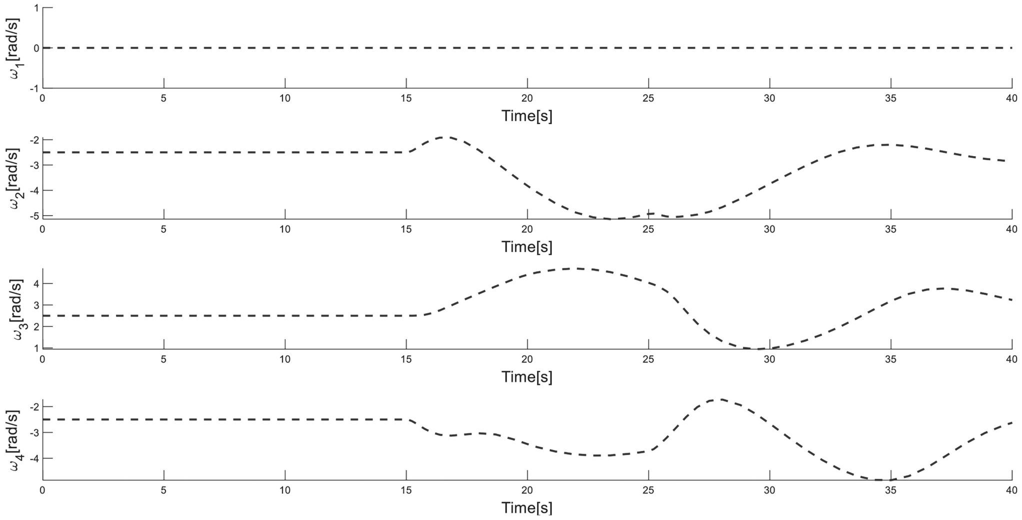


Figure 11. The rotor velocity.

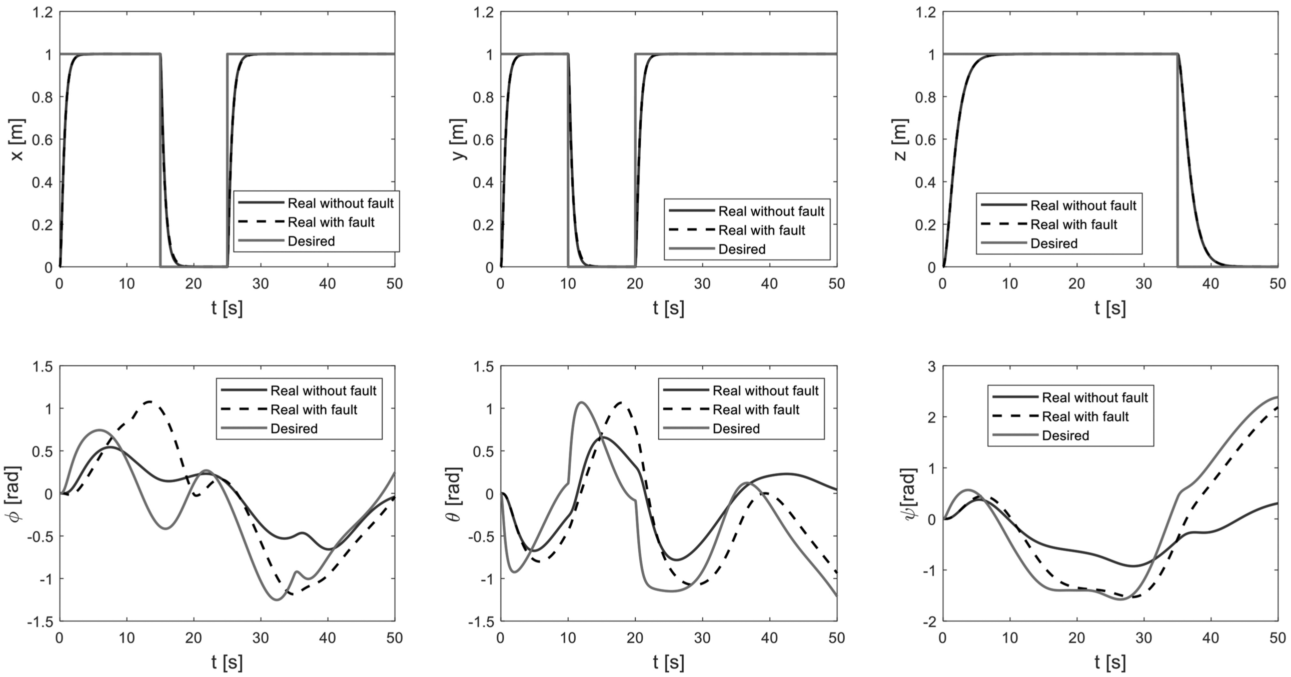


Figure 12. Position and Euler angles.

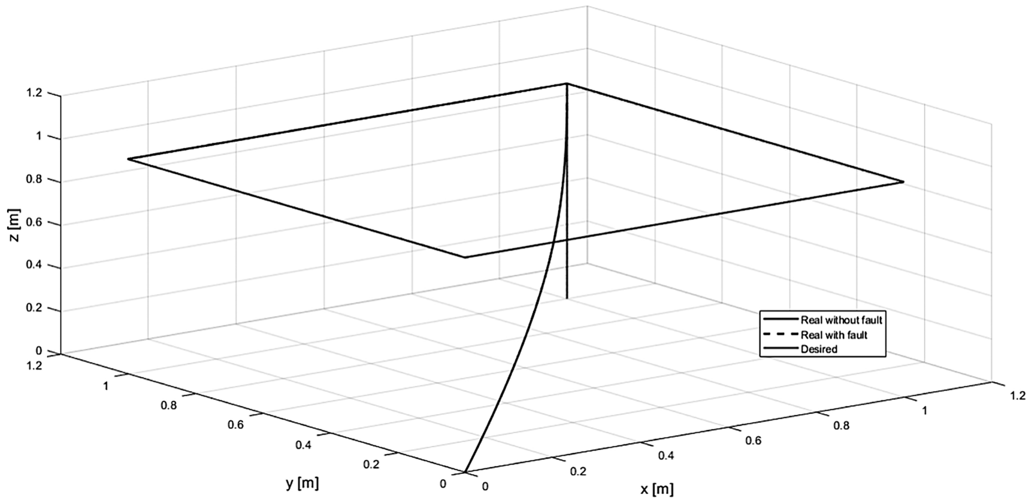


Figure 13. Quadrotor's path.

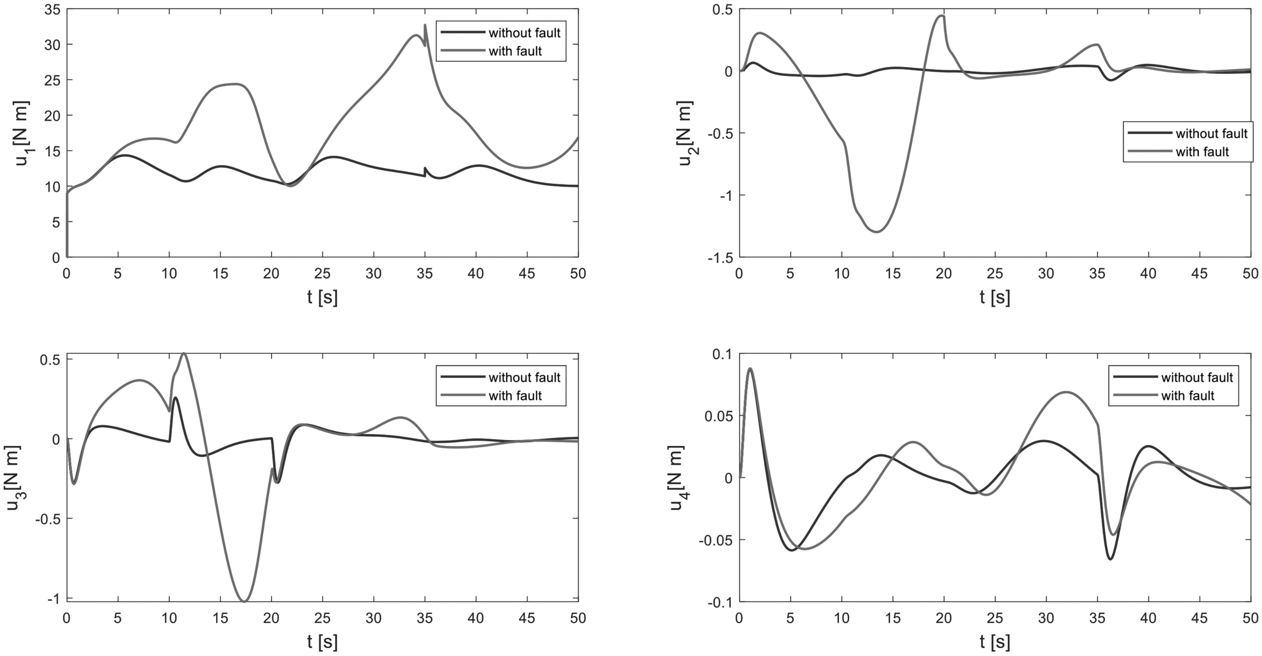


Figure 14. Control inputs.

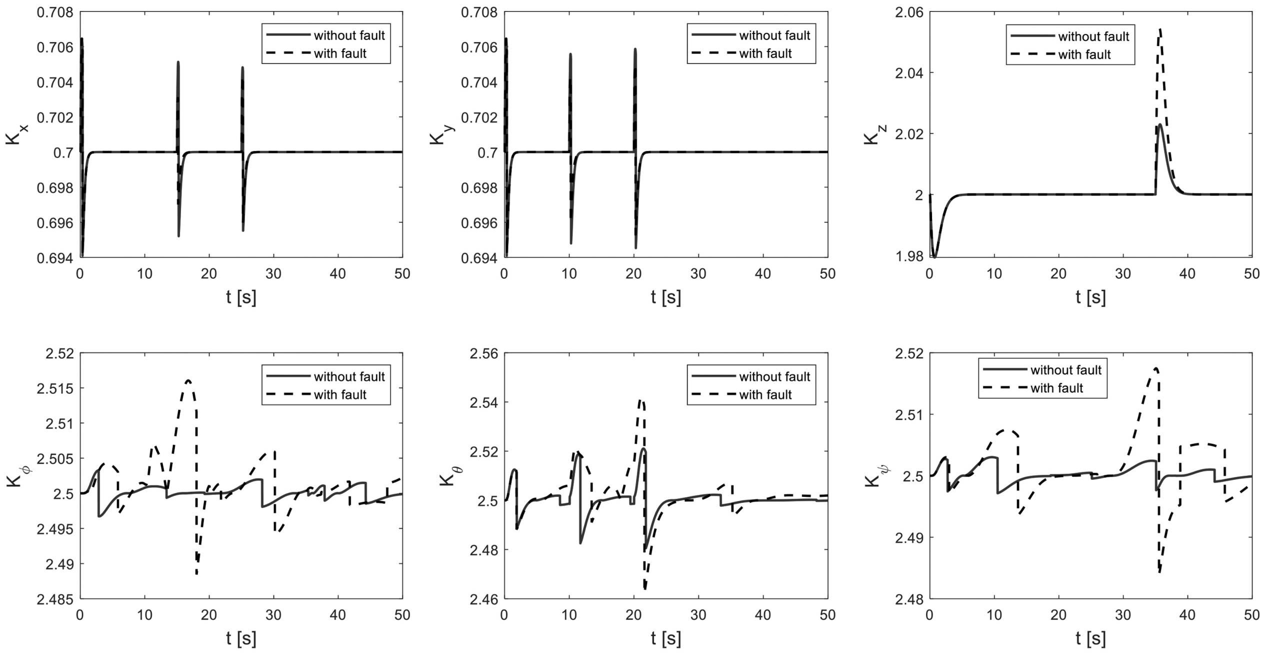


Figure 15. Adjustable parameters of controller.

Table 1
Quantitative Comparisons of Simulation Results

	Steady State Error	
Parameter	Without Fault	With Fault
x[m]	0.0011	0.0012
y[m]	3.16e-4	3.78e-4
z[m]	3.64e-10	3.64e-10

6. Conclusion

Controlling a UAV is of great importance to maintain its stability and proper manoeuvrability. Failure of rotors is one of the main issues that can severely affect the correct performance of the quadcopter, and even failure of rotors can cause the system to lose stability and crash. Therefore, one of the important challenges in designing the controller for these systems is maintaining stability in the event of such errors. In this paper, control algorithms based on Lyapunov's theory are proposed to control and track the

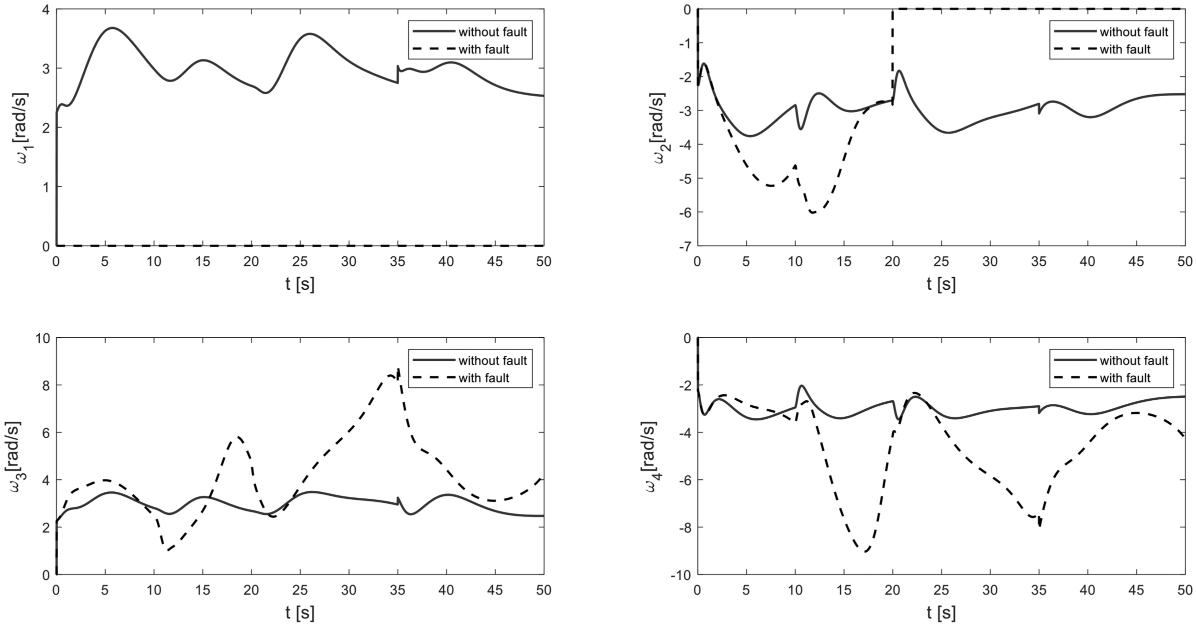


Figure 16. Rotors angular velocity (ω_1 at 8s, ω_2 at 22 s).

attitude, position, and altitude of a quadrotor with six degrees of freedom and nonlinear dynamic behaviour, it is proposed that even if the rotors fail, the quadrotor is able to continue its mission and land safely without losing the stability of the system.

In order to check the performance of the proposed designed controller, the dynamic model of the quadrotor along with the controller has been simulated in MATLAB/Simulink, and the main results are summarised as follows: (a) All state variables converge to their reference values in sequence, even if their reference values change suddenly at different instants, (b) Different paths of the quadrotor are obtained by changing the reference positions and different positions are also obtained by changing the reference angles, (c) Position and velocity tracking errors of all system state variables tend to zero, and according to the simulation results in this article, the controller has a very good performance.

References

[1] S. Islam, X.P. Liu, and A.E. Saddik, Adaptive sliding mode control of unmanned four rotor flying vehicle, *International Journal of Robotics and Automation, 30*, 2015, 140–148.

[2] X. Fu and H. Guo, Robust adaptive fault-tolerant control based on GBF-CMAC neural network for low-altitude UAV, *International Journal of Robotics and Automation, 38*, 2023, 267–276.

[3] R. Miranda-Colorado and L.T. Aguilar, Robust PID control of quadrotors with power reduction analysis, *ISA Transactions, 98*, 2020, 47–62.

[4] A. Jebelli and M.C.E. Yagoub, Fuzzy logic PID-based control design for a small underwater robot with minimum energy consumption, *International Journal of Mechanical Engineering and Robotics Research, 5*(3), 2016, 186–190.

[5] R. Fessi and S. Bouallègue, LQG controller design for a quadrotor UAV based on particle swarm optimisation, *International Journal of Automation and Control, 13*, 2019, 569–594.

[6] P. Lu and E.J. Van Kampen, Active fault-tolerant control for quadrotors subjected to a complete rotor failure, *Proc. 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Hamburg, 2015, 4698–4703.

[7] G. Arleo, F. Caccavale, G. Muscio, and F. Pierri, Control of quadrotor aerial vehicles equipped with a robotic arm, *Proc. 21St Mediterranean Conference On Control And Automation*, 2013, Platanias, 1174–1180.

[8] A. Jebelli, M.C.E. Yagoub, and B.S. Dhillon, Feedback linearization approach to fault tolerance for a micro quadrotor, *Proc. IEEE International Conference on Industrial Technology,* Lyon, 2018, 165–168.

[9] Z. Hou, P. Lu, and Z. Tu, Nonsingular terminal sliding mode control for a quadrotor UAV with a total rotor failure, *Aerospace Science and Technology, 98*, 2020, 105716.

[10] Y.A.A.Y. Aoki, Y. Asano, A. Honda, N. Motooka, and T. Ohtsuka, Nonlinear model predictive control of position and attitude in a hexacopter with three failed rotors, *IFAC-PapersOnLine, 51*, 2018, 228–233.

[11] F. Nan, S. Sun, P. Foehn, and D. Scaramuzza, Nonlinear MPC for quadrotor fault-tolerant control, *IEEE Robotics and Automation Letters, 7*, 2022, 5047–5054.

[12] R.C. Avram, X. Zhang, and J. Muse, Nonlinear adaptive fault-tolerant quadrotor altitude and attitude tracking with multiple actuator faults, *IEEE Transactions on Control Systems Technology, 26*, 2017, 701–707.

[13] D. Asadi, K. Ahmadi, and S.Y. Nabavi, Fault-tolerant trajectory tracking control of a quadcopter in presence of a motor fault, *International Journal of Aeronautical and Space Sciences, 23*, 2022, 129–142.

[14] C.D. Pose, J.I. Giribet, and A.S. Ghersin, Hexacopter fault tolerant actuator allocation analysis for optimal thrust, *Proc. International Conference on Unmanned Aircraft Systems*, Miami, FL, 2017, 663–671.

[15] A. Jebelli, H. Chaoui, A. Mahabadi, and B.S. Dhillon, Tracking and mapping system for an underwater vehicle in real position using sonar system, *International Journal of Robotics and Automation, 37*(1), 2022, 124–134.

[16] F. Fei, Z. Tu, D. Xu, and X. Deng, Learn-to-recover: Retrofitting uavs with reinforcement learning-assisted flight control under cyber-physical attacks, *Proc. IEEE International Conference on Robotics and Automation*, Paris, 2020, 7358–7364.

[17] A. Jebelli, A. Mahabadi, A. Nayak, and R. Ahmad, Increasing the operating depth of a Teflon underwater vehicle using a magnetic field *Ocean Engineering, 250*, 2022, 111078.

[18] A. Freddi, A. Lanzon, and S. Longhi, A feedback linearization approach to fault tolerance in quadrotor vehicles, *IFAC Proceedings Volumes, 44*, 2011, 5413–5418.

[19] J. Ye, Adaptive control of nonlinear PID-based analog neural networks for a nonholonomic mobile robot, *Neurocomputing, 71*, 2008, 1561–1565.

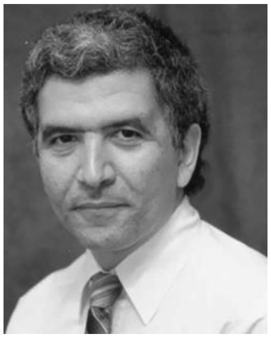
[20] H. Razmi and S. Afshinfar, Neural network-based adaptive sliding mode control design for position and attitude control of a quadrotor UAV, *Aerospace Science and Technology, 91*, 2019, 12–27.

Biographies

Alireza Najafiyanfar received the B.Sc. degree in control engineering from Tafresh University in 2018, and the M.S. degree in control engineering from K.N. Toosi University of Technology, Tehran, Iran, in 2021, where he is currently pursuing the Ph.D. degree. His research interests are hybrid systems, symbolic control, system identification, nonlinear control, intelligence control, industrial automation, autonomous robotics, and electric motor drives.

Ali Jebelli He received the master's degree (M.Eng.) in electrical–mechatronics and automatic control from the University Technology Malaysia in 2009, and the bachelor's degree in electrical power engineering in 2005. He received the master's degree and the Ph.D. degree in electrical and computer engineering from the University of Ottawa in 2014 and 2016, respectively. During his studies at the University of Ottawa, he worked as a Research Assistant and a Teacher Assistant with the Department of Mechanical Engineering and the School of Electrical Engineering and Computer Science, and during that time, he won several prestigious awards. His research interests include autonomous systems, intelligent control, robotics, mechatronics, electric motors drive, and solar and wind energy. He has authored or coauthored over 100 publications on these topics in international journals and referred conferences. Dr. Jebelli is currently leading the RoboticC Inc. group, a team interested in building agricultural robots to increase the quality and quantity of products and designing autonomous vehicles and drones. Prior to this position, he completed three post-doctoral positions: the first with the Department of Electrical Engineering and Computer Science and Mechanical Engineering, University of Ottawa and the second with the Department of Electronics, Carleton University, and the third with the Department of Mechanical Engineering, University of Alberta.

Arezoo Mahabadi received the B.Eng. degree (with Honors) in basic engineering science from the University of Tehran in 2019. Her research interests include intelligent systems, robotics, mechatronics, mathematical modelling of stationary field, computational electromagnetics solar energy systems, sonar and radar systems, electric motor drives, and energy storage and management. She is currently a supervisor of the Engineering Department with RoboticC Inc.



Mustapha C. E. Yagoub received the Dipl.-Ing. degree in electronics and the Magister degree in telecommunications from the École Nationale Polytechnique, Algiers, Algeria, in 1979 and 1987, respectively, and the Ph.D. degree from the Institut National Polytechnique, Toulouse, France, in 1994. After a few years working in industry as a Design Engineer, he joined the Institute of Electronics, Université des Sciences et de la Technologie Houari Boumédiene, Algiers, Algeria, first as a Lecturer from 1983 to 1991 and then as an Assistant Professor from 1994 to 1999. From 1996 to 1999, he has been the Head of the Communication Department. From 1999 to 2001, he was a Visiting Scholar with the Department of Electronics, Carleton University, Ottawa, ON, Canada, working on neural network applications in microwave areas. In 2001, he joined the School of Electrical Engineering and Computer Science, University of Ottawa, where he is currently a Professor. He has authored or coauthored over 300 publications on these topics in international journals and referred conferences. He authored *Conception de Circuits Linéaires Et Non Linéaires Micro-Ondes* (Cépadues, Toulouse, France, 2000), and coauthored *Computer Manipulation and Stock Price Trend Analysis* (Heilongjiang Education Press, Harbin, China, 2005). Dr. Yagoub is a Senior Member of the IEEE Microwave Theory and Techniques Society, a member of the Professional Engineers of Ontario, Canada.