Fault-tolerant control of a Quadrotor despite the complete rotor failure via adaptive Lyapunov-based control

Ali Jebelli1*, Alireza Najafiyanfar2, Mustapha C. E. Yagoub3

1Department of Mechanical Engineering, University of Alberta, Edmonton, Canada
2Faculty of Electrical Engineering, K.N. Toosi University of Technology, Tehran, Iran
3School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON Canada

*jebelli@ualberta.ca

Abstract—In this paper, an efficient method is proposed for the position and altitude tracking control of a quadrotor UAV through a nonlinear dynamic model in case of failure of one or two quadrotor rotors and in the presence of parametric uncertainties. In fact, the system continues its tasks correctly even if one or two rotors of the quadrotor stop working. The proposed method is a combination of the Lyapunov stability theory and the neural network adaptive scheme, in which a Lyapunov-based controller was designed for subsystems separately, and their coefficients adaptively tuned by the neural network method. Further, the performances of the proposed control method were evaluated. The simulated results demonstrated that the proposed controller exhibits desirable transient behavior and performance stability, is not sensitive to parameter variations, and has remarkable stability and performance robustness despite the complete rotor failure. Hence, for operational purposes where the stability and continuation of the mission in case of failure of the rotors are of vital importance, using an adaptive Lyapunov-based control approach is recommended.

Keywords—Unmanned aerial vehicle (UAV); Quadrotor; Lyapunov; Neural network; Fault Tolerance

1. INTRODUCTION

Extensive researches have 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 focusing 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], and Linear Quadratic Gaussian (LQG) control [4], to name a few. On the other hand, Fault Tolerant Control (FTC) issues have been arisen from defects in the AUV sensors, motors, or other segments [5]. Fault Tolerant Control systems can be classified into passive and active [6]. 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 [7]. Whereas, if there is a fault in an active FTC system, the control system will be reset [7].

In Fault Tolerant Control systems, different methods have been explored to design the controllers in the case of engine failure. Sliding method has been used in [8] to control the operating conditions in the case of engine failure and disturbance, while authors in [9] used Model Predictive Control to control the system. Robust adaptive control has been used in [10] to track the height and control the quadrotor attitude. In [11], a nonlinear discrete adaptive algorithm and a PID algorithm have been used respectively within the inner and outer loops to control the path tracking. Optimization methods have been used in [12] to minimize the forces applied by the rotors in the case of their failure. Smart control methods such as reinforcement learning [13] have been used for Fault Tolerant Control. Fuzzy logic algorithm has been used in [14] 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 engine fails, the controller is not capable of controlling one of the variables of roll, pitch and yaw; needless to point that, yaw is often ignored, i.e., roll, pitch and height 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 height control [15]. However, lack of yaw control in the case of engine 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 engine 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 rotor failure. In this paper with the motive to continue the mission of quadrotor in case of rotors failure, a control method based on the Lyapunov stability theory and adaptive control using neural network was designed to efficiently control the attitude, position and altitude of the quadrotor. The proposed control method is resistant to parametric uncertainties and failure of operators, and if one or two rotors fail, the quadrotor can perform its tasks using only the remaining rotors without losing stability and drastic changes in position.

This paper includes the following sections: In the second section, the mathematical modeling of the quadrotor is presented. In the third section, the design details of the altitude controller, position controller, and attitude controller are described. In the fourth section, the details of the controller of Euler angles are stated. The simulation and comparison results are presented in the fifth section and the conclusion is presented in the sixth section.

II. MATHEMATICAL MODEL

A quadrotor is an Unmanned Aerial Vehicle (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 Figure 1, where the state vector \([x, y, z, \varphi, \theta, \psi]\) 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 [16]:

\[
\begin{align*}
\ddot{x} &= (\cos(\varphi) \cos(\psi) \sin(\theta) + \\
&\sin(\varphi) \sin(\psi)) u_1 \\
\ddot{y} &= (\cos(\psi) \sin(\varphi) - \\
&\cos(\varphi) \sin(\psi) \sin(\theta)) u_1 \\
\ddot{z} &= g - \cos(\varphi) \sin(\theta) u_1 \\
\ddot{\varphi} &= \left(\frac{l_{B,xx}-l_{B,zz}}{l_{B,xx}}\right) \dot{\psi} \dot{\theta} - \\
&\frac{K_{d,yy}}{l_{B,xx}} \left(\dot{\varphi}^2 + \dot{\theta}^2 + \dot{\psi}^2 \dot{\varphi} + u_3 + \frac{l_{B,zz}}{l_{B,xx}} \dot{\theta} \omega_r \right)
\end{align*}
\]

Here, \([x, \dot{x}, y, \dot{y}, z, \dot{z}, \varphi, \dot{\varphi}, \theta, \dot{\theta}, \psi, \dot{\psi}]\) is the system state vector and \([u_1, u_2, u_3, u_4]\) the vector of the system control inputs. \(m\) 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 \(\omega_4\) stand for the angular speed of the propeller. \(\omega_r = (\omega_1 - \omega_2 + \omega_3 - \omega_4)\) denotes the overall residual rotor angular velocity. From that, the control inputs \(u_1, u_2, u_3\) and \(u_4\) can be calculated by the following matrix system

\[
\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -l & 0 & 0 & 0 \\ 0 & l & 0 & -l \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix}
\]

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 is one.
- The axes of the body-frame are coincident with the quadrotor inertia axes.

![Fig. 1 Definition of quadrotor body frame and rotor indices](image)

III. CONTROLLER DESIGN

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

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

\[ V_h = \frac{1}{2} [(z - z_d)^2 + \dot{z}^2] \]  

Its first derivative can be expressed as

\[ \dot{V}_h = \dot{z}(z - z_d) + \ddot{z} \]  

Substituting (3) into (9) gives

\[ \dot{V}_h = \dot{z}(z - z_d) + \dot{g} - \frac{\cos(\phi) \cos(\theta)}{\cos(\theta)} \dot{k}_x \]  

\( u_t \) should be then selected as following to keep the Lyapunov stability conditions

\[ u = \frac{(z - z_d) + g + k_x \dot{z}}{\cos(\phi) \cos(\theta)} \]  

in which the parameter \( k_x > 0 \) needs to be designed for control (11). It 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 \( k_x \) with the adaptation law derived from the conventional backpropagation algorithm. The neural network is trained by the specialized learning architecture [17]-[18] to minimize the performance error \( E \):

\[ E = \frac{1}{2} (z_d - z)^2 \]  

Based on the gradient descent method [18], we have the following adaptation equation:

\[ k_x = k_{x0} - \varepsilon_x \int_0^t \frac{\partial E}{\partial k_x} dt \]  

where \( \varepsilon_x \) is the learning rate which determines the convergence speed of neural network and \( k_{x0} \) the initial value of \( k_x \). Using the chain rule:

\[ \frac{\partial E}{\partial k_x} = \frac{\partial E}{\partial z} \frac{\partial z}{\partial k_x} = - (z_d - \dot{z}) \]  

Assuming \( \frac{\partial z}{\partial u_t} = \text{sign} \left( \frac{\nabla z}{\nabla u_t} \right) \) [21], substituting (14) into (13) gives:

\[ k_x = k_{x0} + \varepsilon_x \int_0^t (z_d - \dot{z}) \text{sign} \left( \frac{\nabla z}{\nabla u_t} \right) \frac{\dot{z}}{u_t} dt \]  

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

B. Position Control

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

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

with its derivative equals to

\[ \dot{V}_p = \dot{x}(x - x_d) + \ddot{x} + \dot{y}(y - y_d) + \ddot{y} \]  

Rewriting the acceleration components in the horizontal plane

\[ \ddot{x} = u_x u_1, \quad \ddot{y} = u_y u_1 \]  

Assuming small Euler angles leads to:

\[ u_x = \theta_d, \quad u_y = \phi_d \]  

and substituting (18) into (17) gives

\[ \dot{V}_p = \dot{x}(x - x_d) + \dot{u}_1 u_x + \dot{y}(y - y_d) + \dot{u}_1 u_y \]  

\( u_x \) and \( u_y \) are selected as following to keep the Lyapunov stability conditions.

\[ u_x = \frac{-(x - x_d) - k_x \dot{x}}{u_1} \]  

\[ u_y = \frac{-(y - y_d) - k_y \dot{y}}{u_1} \]  

in which \( k_x > 0 \) and \( k_y > 0 \). Correspondingly, the neural network is used to adjust the positive coefficients \( k_x \) and \( k_y \):

\[ k_x = k_{x0} + \varepsilon_x \int_0^t (x_d - \dot{x}) \text{sign} \left( \frac{\nabla x}{\nabla u_x} \right) \frac{\dot{x}}{u_1} dt \]  

\[ k_y = k_{y0} + \varepsilon_y \int_0^t (y_d - \dot{y}) \text{sign} \left( \frac{\nabla y}{\nabla u_y} \right) \frac{\dot{y}}{u_1} dt \]
\[ k_y = k_{y0} + \varepsilon_y \int_0^t (y_d - y) \text{sign} \left( \frac{y_d}{y} \right) \left( -\frac{y}{u_3} \right) \, dt \]  

(26)

where \( \varepsilon_x \) and \( \varepsilon_y \) 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.

### C. 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.

\[
V_A = \frac{1}{2} [(\varphi - \varphi_d)^2 + \phi^2 + (\theta - \theta_d)^2 + \theta^2 + (\psi - \psi_d)^2 + \psi^2] 
\]  

(27)

The derivative of the Lyapunov function is obtained as follows:

\[
\dot{V}_A = \dot{\varphi}(\varphi - \varphi_d) + \dot{\varphi}\dot{\varphi} + \dot{\theta}(\theta - \theta_d) + \dot{\theta}\dot{\theta} + \dot{\psi}(\psi - \psi_d) + \dot{\psi}\dot{\psi} 
\]  

(28)

Substituting (4)-(6) into (28) leads to

\[
\dot{V}_A = \dot{\varphi}(\varphi - \varphi_d) - c_1\dot{\varphi} + \phi u_2 + \theta(\theta - \theta_d) - c_2\dot{\theta} + \theta u_3 + \psi(\psi - \psi_d) - c_3\dot{\psi} + \psi u_3 
\]  

(29)

where \( u_2, u_3, \) and \( u_4 \) should be selected as following for keeping the Lyapunov stability conditions \( \dot{V}_A < 0 \).

\[
u_2 = -(\varphi - \varphi_d) - k_\varphi \dot{\varphi} 
\]  

(30)

\[
u_3 = - (\theta - \theta_d) - k_\theta \dot{\theta} 
\]  

\[
u_4 = - (\psi - \psi_d) - k_\psi \dot{\psi} 
\]

with \( k_\varphi > 0, k_\theta > 0 \) and \( k_\psi > 0 \).

Correspondingly, the neural network is used to adjust the positive coefficients \( k_\varphi, k_\theta \) and \( k_\psi \), and following results are obtained:

\[
k_\varphi = k_{\varphi0} + \varepsilon_\varphi \int_0^t (\varphi - \varphi_d) \text{sign} \left( \frac{\varphi}{\varphi_d} \right) (-\varphi) \, dt 
\]  

(31)

\[
k_\theta = k_{\theta0} + \varepsilon_\theta \int_0^t (\theta - \theta_d) \text{sign} \left( \frac{\theta}{\theta_d} \right) (-\theta) \, dt 
\]  

(32)

\[
k_\psi = k_{\psi0} + \varepsilon_\psi \int_0^t (\psi - \psi_d) \text{sign} \left( \frac{\psi}{\psi_d} \right) (-\psi) \, dt 
\]  

(33)

where \( \varepsilon_\varphi, \varepsilon_\theta, \) and \( \varepsilon_\psi \) are the respective learning rates that determine the convergence speed of neural network, and \( k_{\varphi0}, k_{\theta0}, \) and \( k_{\psi0} \) the initial values of \( k_\varphi, k_\theta \) and \( k_\psi \), respectively.

### IV. Desired Angle Calculation

Based on the Lyapunov method, the acceleration vector is calculated as [16]:

\[
a^* = -(s - s_d) - g - k_1 \dot{s} 
\]  

(34)

with \( a \) the sum of the acceleration caused by the engine’s forces and \( a^* \) the desired value of this vector. Therefore, the coordinate system will be set such that the z axis corresponds to \( \alpha^* \); then the angular velocity of this system is calculated as:

\[
\omega^E = \begin{bmatrix} \beta_3 \sin(\beta_2) \\ -\beta_2 \\ \beta_3 \cos(\beta_2) \end{bmatrix} 
\]  

(35)

Moreover:

\[
\begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \omega^E + \omega^E 
\]  

(36)

\[
\begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \beta_3 \sin(\beta_2) \\ -\beta_2 \\ \beta_3 \cos(\beta_2) \end{bmatrix} + \begin{bmatrix} \cos(\alpha_3) & \cos(\alpha_3) \sin(\alpha_3) & 0 \\ -\sin(\alpha_3) & \cos(\alpha_3) \cos(\alpha_3) & 0 \\ 1 & -\sin(\alpha_3) & 1 \end{bmatrix} \begin{bmatrix} \ddot{a}_1 \\ \ddot{a}_2 \\ \ddot{a}_3 \end{bmatrix} 
\]  

(37)

Selecting \( \ddot{a}_1, \ddot{a}_2, \) and \( \ddot{a}_3 \) as

\[
\begin{aligned}
\ddot{a}_1 &= -k_{a1} \alpha_1 \\
\ddot{a}_2 &= -k_{a2} \alpha_2 \\
\ddot{a}_3 &= 0
\end{aligned} 
\]  

(38)

leads to the desired as follows:

\[
\begin{bmatrix} \ddot{\varphi}_d \\ \ddot{\theta}_d \\ \ddot{\psi}_d \end{bmatrix} = \begin{bmatrix} \beta_3 \sin(\beta_2) \\ -\beta_2 \\ \beta_3 \cos(\beta_2) \end{bmatrix} + \begin{bmatrix} \cos(\alpha_3) & \cos(\alpha_3) \sin(\alpha_3) & 0 \\ -\sin(\alpha_3) & \cos(\alpha_3) \cos(\alpha_3) & 0 \\ 1 & -\sin(\alpha_3) & 1 \end{bmatrix} \begin{bmatrix} \ddot{a}_1 \\ \ddot{a}_2 \\ \ddot{a}_3 \end{bmatrix} 
\]  

(39)

### V. Simulations

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 \( I = diag([2.7,2.7,5,2]) \), inertia of the propeller \( I_{ppz} = 1.5 \), drag coefficients \( k_d = diag([0.7,0.7,1.4]) \). In the numerical values of
$m, I_B, I_{pzz},$ and $K_d$ parameters, 20% of uncertainty is considered. The purpose of the controller design is to track the following desired trajectory:

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

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

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

Two cases were considered to evaluate the controllers. In the first case, the quadrotor does the task without any rotor failure. In the second case, two rotors of the quadrotor are turned off, one rotor is turned off from the beginning and the second one is turned off after 20 seconds. The simulation results in figures 3-7 show that the quadrotor does not lose its stability and completes its task with acceptable performance in spite of two rotor failure. As it is clear from Figure 3, the failure of the rotor does not increase the error and changes in the position and the Euler angles of the system, and the tracking of the desired value is done correctly.

The Control inputs and adjustable parameters of the controllers are shown in figure 6 and figure 7, respectively. As for the engine angular velocity, figure 7, it demonstrated the good behavior of the system even after the failure of two rotors (one rotor from the beginning and one after 20 seconds).

Fig. 3 Position and Euler angles. Desired value, and real values without fault and with fault are shown by the red line, blue line, and black dashed line, respectively.

Fig. 4 Desired trajectory, and real trajectories without fault and with fault are shown by the red line, blue line, and black dashed line, respectively.

Fig. 5 Control inputs in cases without fault and with fault are shown by the blue line and red line, respectively.

Fig. 6 Adjustable parameters in cases without fault and with fault are shown by the blue line and black dashed line, respectively.

Quantitative comparisons of trajectory tracking results are given in TABLE I. As shown, the steady-state error does not change much if the rotors fail, which indicates the proper functioning of the control system. Therefore, the proposed method tracks the desired trajectory with acceptable transient and steady-state response characteristics and provides robustness against parametric uncertainties and failure rotor.
controlling different quadrotor systems. The state variables converge to their optimal allocation for fault resilience parametric uncertainty. Hexacopter fault tolerance is achieved by (i) designing a fault-tolerant controller, (ii) implementing the controller in simulation, and (iii) validating its performance in experiments. The controller performs very well as the state variables converge to their optimal allocation, even in cases with parametric uncertainty and fault occurrence.

Fig. 7 Rotor angular velocity ($\omega_1$ at 0s, $\omega_2$ at 20s) in cases without fault and with fault are shown by the blue line and black dashed line, respectively.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Steady State Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>x[m]</td>
<td>8.6597e-14</td>
</tr>
<tr>
<td></td>
<td>5.7954e-14</td>
</tr>
<tr>
<td>y[m]</td>
<td>7.2387e-14</td>
</tr>
<tr>
<td></td>
<td>6.5505e-14</td>
</tr>
<tr>
<td>z[m]</td>
<td>6.4717e-06</td>
</tr>
<tr>
<td></td>
<td>5.7172e-06</td>
</tr>
</tbody>
</table>

VI. CONCLUSION

Efficient controlling of UAV is of great importance to maintain its stability and proper maneuverability. Motor failure is one of the main issues that can severely affect the proper functioning of these systems, and even 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, a control method based on Lyapunov’s theory and an adaptive scheme based on neural network are proposed to control and track the state, position, and height of a quadrotor with six degrees of freedom and nonlinear dynamic behavior. Despite parametric uncertainties and in case of failure of the rotors, the quadrotor is able to continue its mission without losing the stability of the system. In order to check the performance of the designed controller, the dynamic model of the quadrotor along with the controller was simulated and the main results are summarized as follows: (i) the state variables converge to their reference values, even if their reference values change suddenly at different instants, (ii) different quadrotor trajectories are obtained by changing the reference positions and different conditions are also obtained by changing the reference angles, (iii) position and velocity tracking errors of all system state variables tend to zero, and (iv) the steady-state error does not change much if the rotors fail, so according to the simulation results, the controller performs very well.

REFERENCES