Fault Tolerant Control for Quadrotor via Backstepping Approach

Xiaobing Zhang¹, Youmin Zhang², Chun-Yi Su³, and Ying Feng⁴

Department of Mechanical and Industrial Engineering
Concordia University, Montreal, Canada, H3G 1M8

Different from the previous work for quadrotor which mostly discusses application of different controllers to quadrotor control under normal flight conditions, this paper introduces a new design strategy which considers fault tolerant capability of the quadrotor control problems. By utilizing Passive Fault Tolerant Control Systems (PFTCS) concept with backstepping control approach, the trajectory tracking control of the quadrotor UAV has been achieved and simulation results have shown the effectiveness of the proposed backstepping control with higher controller gains when different levels of actuator faults occur in the quadrotor UAV.

Nomenclature

x, y, z = position information
u₁ = height control input
u₂, u₃, u₄ = roll, pitch, yaw control input respectively
m = mass of quad-rotor
\( \ddot{\phi}, \ddot{\theta}, \ddot{\psi} \) = roll angular acceleration, pitch angular acceleration, yaw angular acceleration
\( \phi, \theta, \psi \) = roll angle, pitch angle and yaw angle
k_d = drag coefficients
l = distance of motor from pivot centre
J_x = Initial moment of x
J_y = Initial moment of y
J_z = Initial moment of z
g = gravity acceleration

I. Introduction

Once a system has been completely constructed, it is impossible to guarantee that the system can operate properly and predict when the fault will happen. Fault Tolerant Control Systems (FTCS) design can ensure the system to be in an acceptable performance level when the components of the system, such as sensors, actuators, system/plant components and controllers, have any malfunctions occurred.

FTCS is a control system with the ability to tolerate faults automatically and to achieve the desired performance under the fault conditions.¹ FTCS can be designed in two different types which are known as Passive Fault Control Systems (PFTCS) and Active Fault Control Systems (AFTCS). PFTCS are designed with fixed controller such that the controller is robust enough to withstand the possible malfunctions during system operation life. AFTCS are designed based on reconfigurable controller with a supporting scheme called Fault Detection and Diagnosis (FDD).

¹ Graduate Student, Department of Mechanical and Industrial Engineering, Concordia University, Montreal, Canada, H3G 1M8.
² Associate Professor, Department of Mechanical and Industrial Engineering, Concordia University, Montreal, Canada, H3G 1M8. Senior member, AIAA. Email: ymzhang@encs.concordia.ca. Corresponding author.
³ Professor, Department of Mechanical and Industrial Engineering, Concordia University, Montreal, Canada, H3G 1M8.
⁴ Post Doctoral Researcher, Department of Mechanical and Industrial Engineering, Concordia University, Montreal, Canada, H3G 1M8.

American Institute of Aeronautics and Astronautics
scheme to provide on-line system operating conditions for controller reconfiguration once a fault occurs during system operation. Comparing with AFTCS, PFTCS does not need reconfigure control law and a FDD scheme. Hence, PFTCS are generally simple but with probably limited ability to deal with more fault types or severe faults. Meanwhile, AFTCS can be considered as real time systems because desired performance demands the system reacts properly in the limited time after fault occurrence. AFTCS are also known as self-repairing or self-designing control systems.

Nowadays, control methods mostly used for fault-free operation of quadrotor UAV are sliding mode control, sliding mode control mixed with backstepping, adaptive backstepping control, feedback linearization, backstepping, PID and LQ control techniques, LQR control design. Every method has its own advantages and drawbacks. Backstepping controller highly guarantee the convergence to the desired values while it needs to design complex Lypunov functions; Sliding mode control is insensitive to the model errors and parametric uncertainties and other disturbances, and it can be response quickly. However, sliding mode control can bring the chattering phenomena.

There are some controllers utilized in FTCS field, such as the observer-based controller for the unified continuous/discrete-time formulation, LQG controller for linear systems with sensor partial failures, sliding mode controller for an uncertain MIMO aircraft model F-18, and feedback law. Some control laws employed in AFTCS are the same as those for PFTCS (apply sliding mode controller and concerns feedback control).

Comparing with a lot of work done in FTCS area and control design for quadrotor, there is not much research connecting both together, especially applying the FTCS on quadrotor based on backstepping controller. Therefore, the main objective and contribution of this paper is to analyze the system responses based on PFTCS approach via utilizing backstepping theory for handling different levels of actuator faults in quadrotor UAV. Section IV will clarify the details and simulate various situations including single failure and simultaneous failure conditions.

II. Model

The quadrotor is an under-actuated system because it has six-degree of freedom while it has only four inputs. The collective input (or throttle input) is the sum of the thrusts of each motor. The four rotors have been divided into front and back rotors (1 & 3) and left and right ones (2 & 4). The front and back rotors rotate in counter-clockwise direction while the other two in clockwise direction. All of the movements can be controlled by the changes of each rotor speed. Vertical flight is achieved by increasing all of the rotors’ speed to move up or decreasing the speed to go down. Roll motion can be controlled by decreasing (increasing) the left rotor speed while increasing (decreasing) the right rotor speed to make the quadrotor roll left (right). Pitch motion can be controlled by decreasing (increasing) the front rotor speed while increasing (decreasing) the rear rotor speed to make the quadrotor up (down). Yaw moment is a little different, which depends on all of the rotors’ speed. When front and rear pair spins slower (faster) than left and right pair, the quadrotor will move in positive (negative) direction (counter-clockwise/clockwise direction).
Fig. 1 shows the structure of the quadrotor. The inertial frame $E = \{x_E, y_E, z_E\}$ is fixed with the earth, $B = \{x_B, y_B, z_B\}$ represents the body frame fixed with quadrotor body, $P = \{x, y, z\}$ is the position of the quadrotor mass centre expressed in the inertial frame. $F_1, F_2, F_3, F_4$ are each Propeller’s thrust respectively, and $m$ is the mass of the quadrotor. Meanwhile, the Euler angles are roll angle $\phi \in (-\frac{\pi}{2}, \frac{\pi}{2})$, pitch angle $\theta \in (-\frac{\pi}{2}, \frac{\pi}{2})$ and yaw angle $\psi \in (-\frac{\pi}{2}, \frac{\pi}{2})$ respectively. The rotation matrices from body frame to earth frame can be obtained as

$$L = \begin{pmatrix}
cos\theta\cos\psi & \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi \\
cos\theta\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi \\
-sin\theta & sin\phi\cos\theta & cos\phi\cos\theta
\end{pmatrix}$$

According to Newton-Euler equation:

$$m\ddot{v}_b + \omega_b \times m\dot{v}_b = F_b$$
$$J\dot{\omega}_b + \omega_b \times J\omega_b = T_B$$

where $\omega_b = \begin{pmatrix} P \\ Q \\ R \end{pmatrix}$ is quadrotor’s angular velocity, $v_b$ is body velocity, $T_b$ denotes the quadrotor’s body moment, $J = \begin{pmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{pmatrix}$ is inertial matrix of the rigid body.

**Dynamic Equations:**

$$\begin{align*}
\dot{u} &= \frac{1}{m} (mgsin\theta - D_x) + \psi v - \theta w \\
\dot{v} &= \frac{1}{m} (-mgsin\phi cos\theta - D_y) + w\phi - \psi u \\
\dot{w} &= \frac{1}{m} (u_1 - D_z - mgcos\phi cos\theta) + \theta u - \phi v
\end{align*}$$

**Navigation Equations:**

$$\begin{align*}
\dot{x} &= \frac{u_1}{m} (cos\psi sin\theta cos\phi + sin\phi sin\psi) - \frac{k_{d1}\dot{x}}{m} \\
\dot{y} &= \frac{u_1}{m} (sin\psi sin\theta cos\phi - cos\phi sin\alpha) - \frac{k_{d2}\dot{y}}{m} \\
\dot{z} &= \frac{u_1}{m} cos\theta cos\phi - \frac{k_{d3}\dot{z}}{m} - g
\end{align*}$$

**Moment Equations:**

$$\begin{align*}
\dot{\phi} &= \frac{1}{J_x} [u_2l - k_{d4}\phi - \theta\psi(J_x - J_y)] \\
\dot{\theta} &= \frac{1}{J_y} [u_3l + k_{d5}\theta - \phi\psi(J_x - J_y)] \\
\dot{\psi} &= \frac{1}{J_z} [u_4l + k_{d6}\psi - \phi\theta(J_y - J_x)]
\end{align*}$$

**Control Input:**
where \(x, y, z\) are position information; \(u_1\) is height control input; \(u_2, u_3, u_4\) denotes roll, pitch, yaw control input respectively; \(\phi, \dot{\phi}, \ddot{\phi}\) respectively denotes roll angular acceleration, pitch angular acceleration, yaw angular acceleration; \(\dot{\theta}, \ddot{\theta}, \psi, \dot{\psi}\) represents roll angular velocity, pitch angular velocity, yaw angular velocity; \(k_{di}\) denotes drag coefficients; \(l\) is distance of motor from pivot centre.

### III. Backstepping Controller Design

Backstepping design refers to “step back” to the control input, and a major advantage of backstepping design is its flexibility to avoid cancellation of useful nonlinearities and pursue the objectives of stabilization and tracking, rather than those of linearization. Recursively constructed backstepping controller employs the control Lyapunov function (CLF) to guarantee the global stability.\(^2\)

Considering the following nonlinear system

\[
\dot{x} = f(x) + g(x) \xi \\
\dot{\xi} = u
\]

where \(x\) is the state vector and \(\xi\) as its control.

Firstly the error \(e\) between the actual value \(x\) and the desired value \(x_r\) is defined as:

\[
e = x_r - x
\]

Secondly the system states are defined as:

\[
\begin{align*}
x_1 &= x \\
x_2 &= \dot{x}_1 = \dot{x} \\
z_1 &= e = x_r - x_1
\end{align*}
\]

The key of backstepping is to choose certain variables as virtual controls. Assuming \(x_2\) is the virtual control variable, and \(a(z_1)\) is the function which makes the \(z_1\) approach to the zero

\[
z_2 = a(z_1) - x_2
\]

The derivative of \(z_1\) equals to

\[
\dot{z}_1 = \dot{x}_r - \dot{x}_1 = \dot{x}_r + z_2 - a(z_1)
\]

Choose \(a(z_1) = \dot{x}_r + k_1 z_1 (k_1 > 0)\), and build the Lyapunov function as \(V_1 = \frac{1}{2} z_1^2\), then it can be obtained that

\[
\dot{V}_1 = z_1 \dot{z}_1 = z_1 (z_2 - k_1 z_1) = -k_1 z_1^2 + z_1 z_2
\]

where \(z_1\) subsystem will be stable only if \(\dot{V}_1 < 0\).

The next step is to build the Lyapunov function to make \(z_2 \to 0\)

\[
\begin{align*}
V_2 &= V_1 + \frac{1}{2} z_2^2 \\
\dot{V}_2 &= \dot{V}_1 + z_2 \dot{z}_2 = -k_1 z_1^2 + z_2 \left[ z_1 + \dot{x}_r + k_1 \dot{z}_1 - \dot{x}_2 \right]
\end{align*}
\]

To obtain \(\dot{V}_2 < 0 (k_2 > 0)\),
\[
\begin{align*}
\dot{x}_2 &= \dot{x}_r + (1 - k_1^2)z_1 + (k_1 + k_2)z_2 \\
\dot{z}_1 &= x_r - x_t \\
\dot{z}_2 &= \dot{x}_r + k_1z_1 - x_2 \quad (k_1 > 0, k_2 > 0)
\end{align*}
\]

The control law of quadrotor 6DOF system can be obtained as follows:

\[
\mathbf{u} = \begin{pmatrix}
\frac{m}{u_1} \left[ \dot{\mathbf{r}}_x + \frac{k_u}{m} \mathbf{x} + z_1 (k_1^2 - 1) - z_2 (k_1 + k_2) \right] \\
\frac{m}{u_2} \left[ \dot{\mathbf{r}}_y + \frac{k_u}{m} \mathbf{y} + z_1 (k_1^2 - 1) - z_2 (k_1 + k_2) \right] \\
\frac{m}{\cos \theta \cos \psi} \left[ \dot{z}_r + \frac{k_u}{m} + g + z_1 (k_1^2 - 1) - z_2 (k_1 + k_2) \right] \\
\frac{J_x}{I} [\dot{\psi} (J_y - J_z) + k_{d_1} \dot{\psi}] + \frac{J_y}{I} [\dot{\theta} (J_z - J_x) + k_{d_2} \dot{\theta}] + \frac{J_z}{I} [\dot{\phi} (J_x - J_y) + k_{d_3} \dot{\phi}] + \frac{J_x}{I} [\dot{\psi} (J_y - J_z) + k_{d_4} \dot{\psi}] + \frac{J_y}{I} [\dot{\phi} (J_z - J_x) + k_{d_5} \dot{\phi}] + \frac{J_z}{I} [\dot{\theta} (J_x - J_y) + k_{d_6} \dot{\theta}]
\end{pmatrix}
\]

\[
\mathbf{u} = \begin{pmatrix}
u_1 \\
u_2 \\
u_3 \\
u_4
\end{pmatrix} = \begin{pmatrix}
b_1 + b_3 + b_2 + b_4 \\
b_2 - b_4 \\
b_1 - b_3 \\
b_2 + b_3 - b_2 - b_4
\end{pmatrix} \omega^2
\]

IV. Fault Tolerant Control Design and Simulation

This section will focus on PFTCS design based on the backstepping control approach.

From Eq. (5) and \( F_l = D \omega_l^2 \) where \( D \) is the coefficient and the \( \omega_l \) is the angular velocity of each propeller, the control inputs can be rewritten as follows in order to design the controller simply:

Note that the drag coefficients \( k_{d_1} \) are assumed too small to be concerned.

By presuming possible faults according to the real quadrotor structure which has four actuators, several actuator fault scenarios (single, double, triple and quadruple partial actuator faults) have been simulated and evaluated. The simulation results will be discussed under different flight situations in this section.

The parameters of the quadrotor used in dynamic modeling are given in Table 1 below.

<table>
<thead>
<tr>
<th>symbol</th>
<th>Description</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m )</td>
<td>Quadrotor mass</td>
<td>0.6150</td>
<td>kg</td>
</tr>
<tr>
<td>( l )</td>
<td>Distance from CG</td>
<td>0.3050</td>
<td>m</td>
</tr>
<tr>
<td>( J_x )</td>
<td>Initial moment</td>
<td>0.0154</td>
<td>kg( \cdot )m(^2)</td>
</tr>
<tr>
<td>( J_y )</td>
<td>Initial moment</td>
<td>0.0154</td>
<td>kg( \cdot )m(^2)</td>
</tr>
<tr>
<td>( J_z )</td>
<td>Initial moment</td>
<td>0.0309</td>
<td>kg( \cdot )m(^2)</td>
</tr>
</tbody>
</table>

The initial position is \((x, y, z) = (0, 0.3, 0)\), and initial Euler angles are \((30, 30, 30)\). The final Euler angles are \((10, 10, 10)\). Desired trajectory can be expressed as:

\[
\begin{align*}
X_r &= 0.5 \cos 0.5t \\
Y_r &= 0.5 \sin 0.5t \\
Z_r &= 3
\end{align*}
\]

A. Fault Scenarios Analysis

The fault scenarios occur under these situations: 1) \( b_1 \) decreases 50% of its normal value, 2) \( b_1 \) and \( b_2 \) decrease 50% of its normal value, 3) \( b_1, b_2 \) and \( b_3 \) decrease 50% of its normal value, 4) \( b_1, b_2, b_3 \) and \( b_4 \) decrease 50% of its normal value.
normal value respectively (red dot line shows the fault scenarios). $b_1$, $b_2$, $b_3$, $b_4$ are coefficient of actuator 1, actuator 2, actuator 3, actuator 4 respectively. From Eq. 14, how each fault scenario affects the system can be shown in the Table 2.

<table>
<thead>
<tr>
<th>Fault Types</th>
<th>$u_1$</th>
<th>$u_2$</th>
<th>$u_3$</th>
<th>$u_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$b_1$ 50% loss</td>
<td>decrease</td>
<td>no effect</td>
<td>decrease</td>
<td>decrease</td>
</tr>
<tr>
<td>$b_1$, $b_2$ 50% loss</td>
<td>decrease</td>
<td>decrease</td>
<td>decrease</td>
<td>no effect</td>
</tr>
<tr>
<td>$b_1$, $b_2$, $b_3$ 50% loss</td>
<td>decrease</td>
<td>decrease</td>
<td>no effect</td>
<td>decrease</td>
</tr>
<tr>
<td>$b_1$, $b_2$, $b_3$, $b_4$ 50% loss</td>
<td>decrease</td>
<td>no effect</td>
<td>no effect</td>
<td>no effect</td>
</tr>
</tbody>
</table>

These analysis results will be proved in the following section.

B. Fault Simulation

The faults happen when fault time $t_f$ equals to 5 second. Simulation will been implemented under each fault scenario discussed before and grouped into two types with different controller gains, $k_1=1$ & $k_2=3$ and $k_1=5$ & $k_2=30$. The case of controller gain $k_1=1$ & $k_2=3$ will be discussed firstly.

![3D trajectory comparison with $k_1=1$ and $k_2=3$ under different fault conditions](image)

**Figure 2.** 3D trajectory comparison with $k_1=1$ and $k_2=3$ under different fault conditions

![Position comparison between fault-free and single fault in $b_1$, with controller gains $k_1=1$ and $k_2=3$](image)

**Figure 3 a)** Position comparison between fault-free and single fault in $b_1$, with controller gains $k_1=1$ and $k_2=3
Table 3. Position errors with \( k_1 = 1 \) & \( k_2 = 3 \) when \( b_1 \) decreases 50% (unit: m)

<table>
<thead>
<tr>
<th>Fault Types</th>
<th>x steady-state error range</th>
<th>y steady-state error range</th>
<th>z steady-state error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fault free</td>
<td>([-0.0175, 0.0175])</td>
<td>([-0.0175, 0.0175])</td>
<td>(-0.0312)</td>
</tr>
<tr>
<td>( b_1 )</td>
<td>([-0.0104, 0.0104])</td>
<td>([-0.0104, 0.0104])</td>
<td>(0.5735)</td>
</tr>
</tbody>
</table>

Figure 3 b) Position error comparison between fault-free and single fault in \( b_1 \), with controller gains \( k_1 = 1 \) and \( k_2 = 3 \)

Figure 4 a) Angle comparison between fault-free and single fault in \( b_1 \), with controller gains \( k_1 = 1 \) and \( k_2 = 3 \)

Figure 4 b) Angle error comparison between fault-free and single fault in \( b_1 \), with controller gains \( k_1 = 1 \) and \( k_2 = 3 \)
### Table 4. Angle errors with $k_1=1$ & $k_2=3$ when $b_1$ decreases 50% (unit: deg)

<table>
<thead>
<tr>
<th>Fault Types</th>
<th>Roll angle error</th>
<th>Pitch angle error</th>
<th>Yaw angle error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fault free</td>
<td>-3.4461e-13</td>
<td>-3.4461e-13</td>
<td>1.7177e-12</td>
</tr>
<tr>
<td>$b_1$</td>
<td>-3.4461e-13</td>
<td>27.6024</td>
<td>27.6032</td>
</tr>
</tbody>
</table>

The position values of $x$ and $y$ change lightly since sine of the Euler angles is small (see Eq. (3)). Meanwhile, when faults occur at $t_f$ which equals to 5 second, $z$ value alters so greatly that the stable error has exceeded 5% when one actuator decreases 50%. Actuator $b_1$ decreasing 50% efficiency affects altitude, pitch angle and yaw angle which can be seen from Table 3 and Table 4. This control scheme with controller gains $k_1=1$ & $k_2=3$ can’t accommodate the double, triple, and quadruple faults. To improve the performance, controller gains are increased from $k_1=1$ & $k_2=3$ to $k_1=5$ & $k_2=30$. Simulation results for single, double, triple and quadruple partial actuator faults are shown and analyzed as follows.

![Figure 5. 3D trajectory comparison with $k_1=5$ and $k_2=30$ under different fault situations](image)

![Figure 6 a) Position error comparison between fault-free and single fault in $b_1$, with controller gains $k_1=5$ and $k_2=30](image)
With higher gains $k_1=5$ & $k_2=30$, more actuator faults can be accommodated comparing with lower gains $k_1=1$ & $k_2=3$. As can be seen in Table 5, tracking errors increase while more faults happen, and steady-state errors in $z$ can meet the 5% requirement even when all of the actuators fail. Fig. 5 shows that the tracking trajectory can converge at the desired trajectory faster by increasing the controller gains comparing with Fig.3. However, this PFTCS based on backstepping scheme still has no capability to handle the faults that actuator 1 and actuator 2 lose their efficiency simultaneously since the controller gains are fixed.

**Table 5** Position errors with $k_1=5$ & $k_2=30$ when $b_1$, $b_1$&$b_2$, $b_1$&$b_2$&$b_3$&$b_4$ decrease 50% separately (unit: m)

<table>
<thead>
<tr>
<th>Fault Types</th>
<th>$x$ steady-state error range</th>
<th>$y$ steady-state error range</th>
<th>$z$ steady-state error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fault free</td>
<td>${\text{negative} \ 4.8408, 4.8408} \times 10^{-4}$</td>
<td>${\text{negative} \ 4.8408, 4.8408} \times 10^{-4}$</td>
<td>$\text{negative} \ 8.2687 \times 10^{-4}$</td>
</tr>
<tr>
<td>$b_1$</td>
<td>${\text{negative} \ 2.8091, 2.8091} \times 10^{-4}$</td>
<td>${\text{negative} \ 2.8091, 2.8091} \times 10^{-4}$</td>
<td>$0.0152$</td>
</tr>
<tr>
<td>$b_1$&amp;$b_2$</td>
<td>${\text{negative} \ 1.2592, 1.2592} \times 10^{-4}$</td>
<td>${\text{negative} \ 1.2592, 1.2592} \times 10^{-4}$</td>
<td>$0.0472$</td>
</tr>
<tr>
<td>$b_1$&amp;$b_2$&amp;$b_3$&amp;$b_4$</td>
<td>${\text{negative} \ 3.2959, 3.2959} \times 10^{-4}$</td>
<td>${\text{negative} \ 3.2959, 3.2959} \times 10^{-4}$</td>
<td>$0.0632$</td>
</tr>
</tbody>
</table>
Figure 7 a) Angle error comparison between fault-free and single fault in b1, with controller gains $k_1=5$ and $k_2=30$

Figure 7 b) Angle error comparison between fault-free and triple faults in b1, b2 and b3, with controller gains $k_1=5$ and $k_2=30$

Figure 7 c) Angle error comparison between fault-free and quadruple faults in all quadrotors, with controller gains $k_1=5$ and $k_2=30$
Table 6 Angle errors with k₁=5 & k₂=30 when b₁ & b₂&b₃&b₄ decrease 50% separately (unit: deg)

<table>
<thead>
<tr>
<th>Fault Types</th>
<th>Roll steady-state error</th>
<th>Pitch steady-state error</th>
<th>Yaw steady-state error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fault free</td>
<td>-1.5454e-13</td>
<td>-1.5454e-13</td>
<td>8.5265e-14</td>
</tr>
<tr>
<td>b₁</td>
<td>-1.5454e-13</td>
<td>-0.4430</td>
<td>-0.4430</td>
</tr>
<tr>
<td>b₁,b₂,b₃</td>
<td>-0.4430</td>
<td>-2.0428e-13</td>
<td>-0.4430</td>
</tr>
<tr>
<td>b₁,b₂,b₃,b₄</td>
<td>8.8818e-14</td>
<td>8.8818e-14</td>
<td>-8.52665e-14</td>
</tr>
</tbody>
</table>

Fig. 7 and Table 6 present how the different faults affect the Euler angles. The change tendency is corresponding with Table 2 which discussed in the former section. Moreover, the steady-state errors of Euler angles are satisfactory for the 5% requirement.

V. Conclusion and Future Work

By adopting the backstepping control approach and utilizing the quadrotor mathematical model, a Passive Fault Tolerant Control Systems (PFTCS) control law based on backstepping control principle was designed and evaluated by single, triple and quadruple partial actuator faults in this paper. Based on the simulation results, the system maintains acceptable performance with partial actuator faults with higher controller gains. However, as PFTC employs the fixed controller gains, it has the limited ability to accommodate more severe fault situations even though the control gains are increased. As one of the future works, investigation of FTCS design based on adaptive or reconfigurable backstepping control approach is under development.

References


