Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 4343214, 11 pages
https://doi.org/10.1155/2020/4343214

Research Article

U-Model-Based Sliding Mode Controller Design for Quadrotor UAV Control Systems

Rui Wang,[1] **Lei Gao**,[1] **Chengrui Bai**,[1] **and Hui Sun**[1,2]

[1] *College of Information Engineering and Automation, Haihang Building, South Campus, Civil Aviation University of China, Tianjin 300300, China*
[2] *Department of Engineering Design and Mathematics, University of the West of England, Frenchy Campus, Coldharbour Lane, Bristol BS16 1QY, UK*

Correspondence should be addressed to Hui Sun; shhappy1@hotmail.com

Received 2 January 2020; Accepted 28 February 2020; Published 28 April 2020

Guest Editor: Weicun Zhang

This paper proposes a U-model-based fault-tolerant controller design method in order to ensure the unmanned aerial vehicle (UAV) flight performance when subject to the actuator failures. Depending on the decoupled quadrotor model, this paper presents a sliding mode control method based on U-model in detail and realizes fault-tolerant control for the quadrotor UAV with the stability theory and simulation experiment verifications. The results show that the new controller designed by using the U-model method can simplify the controller design process which has good fault-tolerant characteristics when actuator faults occur compared with the traditional method.

1. Introduction

Quadrotor vehicle is a typical UAV and has been widely used in reconnaissance, damage assessment, agricultural inspection, express delivery, formation performance, and other multiple different functions by carrying corresponding task equipment [1, 2]. To meet the vast missions quadrotor vehicles can perform, different types of vehicles spring up and the corresponding various controllers are required to meet performance requirements and demands. In addition, quadrotor UAV may suffer various problems such as gust disturbance, mechanical vibration, and actuator failures during flight. These problems may reduce the flight performance and bring greater difficulties for controller design. Therefore, the controller design method compensating among different models will improve the development process of quadrotor UAV while ensuring the control system performance.

The U-model was first proposed in 2002, and this method considers the controlled object as a unit when designing the controller and then designs a universal controller that meets the requirements of the controlled object [3]. The core process is to calculate the inverse of the object plant and integrate with the already designed universal controller to form a new controller. This universal controller is not necessary to be redesigned when fixed because of the performance requirements which are almost the same for different quadrotors. Under normal circumstances, aerial vehicle actuator failures include damage, stuck, floats, and saturation failures. These failures will affect system dynamics performances, even damage aerial vehicles [4]. In order to guarantee the safety for quadrotors, the fault-tolerant control technology has been widely employed in control system designs. Hao proposed an adaptive fault-tolerant control method for the quadrotor attitude system with portion of actuator failures [5–7]. Zhang developed the active fault-tolerant quadrotor UAV flight experiments based on sliding mode control and demonstrated that robustness of the active fault-tolerant is better than that of the passive one [8, 9]. Liu studied an improved fault-tolerant tracking controller for the quadrotor UAV which is able to compensate the efficiency loss due to actuator failures [10]. Meanwhile, multiple studies focus on generating observers to realize fault diagnosis and reconstruction. For example, Yang conducted system linear transformation and reconstructed faults by using sliding mode observers and the equivalent output injection method [11]. Gong proposed a fault diagnosis method based on the state observer technology and theoretically analyzed its robustness and fastness [12].

So far, a large amount of in-depth research on controller design and system construction methods for quadrotor UAVs are improving better and better. However, in specific design process, it is still necessary to spend a lot of time to design different controllers for different models. The main contribution of this paper is to develop a general controller design method for UAVs by combining U-model and the fault-tolerant control methods. The proposed method can simplify the design process, and the designed universal controller can be reused in the design of controllers for different UAV plants as well as guarantee the great control performances.

2. U-Model-Based Quadrotor UAV Model

2.1. Preliminary about Quadrotor UAV Dynamics. First, two basic coordinate systems are mentioned to obtain the mathematical model for the quadrotor, the inertial system E ($OXYZ$), and the airframe coordinate system B ($oxyz$), as shown in Figure 1.

The transformation matrix for converting coordinate systems is described as follows:

$$Q = Q_x Q_y Q_z = \begin{bmatrix} \cos\psi\cos\phi & \cos\psi\sin\theta\sin\psi & \cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi \\ \sin\psi\cos\theta & \sin\psi\sin\theta\sin\phi & \sin\psi\sin\theta\cos\phi - \sin\phi\cos\psi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix}, \tag{1}$$

where Φ, θ, and ψ are the roll angle, pitch angle, and yaw angle of the three axes of the inertial coordinate system, respectively. F_x, F_y, and F_z are defined as components of F in three coordinate axes of the airframe coordinate system. F is the external force applied on the quadrotor, m is the mass of the quadrotor, V is the speed of the aircraft, and M is the sum of the torques experienced by the quadrotor. According to the physical equation,

$$\begin{aligned} F &= m\frac{\mathrm{d}V}{\mathrm{d}t}, \\ M &= \frac{\mathrm{d}H}{\mathrm{d}t}. \end{aligned} \tag{2}$$

Gravity G, drag D_i, lift of a single rotor T_i are defined as follows

$$\begin{aligned} G &= mg, \\ D_i &= \frac{\rho C_d \omega_i^2}{2} = k_d \omega_i^2, \\ T_i &= \frac{\rho C_t \omega_i^2}{2} = k_t \omega_i^2. \end{aligned} \tag{3}$$

According to the force analysis, Newton's second law, and the dynamics and coordinate transformation (1), the linear motion equation can be obtained as follows:

$$\begin{cases} \ddot{x} = \dfrac{(F_x - K_1\dot{x})}{m} = \dfrac{\left(k_t \sum_{i=1}^{4} \omega_i^2 (\cos\psi\sin\theta\cos\phi + \sin\psi\sin\phi) - K_1\dot{x}\right)}{m}, \\ \ddot{y} = \dfrac{\left(F_y - K_2\dot{y}\right)}{m} = \dfrac{\left(k_t \sum_{i=1}^{4} \omega_i^2 (\sin s\psi\sin\phi\cos\phi - \cos\psi\sin\phi) - K_2\dot{y}) - K_2\dot{x}\right)}{m}, \\ \ddot{z} = \dfrac{(F_z - K_3\dot{z} - mg)}{m} = \dfrac{\left(k_t \sum_{i=1}^{4} \omega_i^2 (\cos\phi\cos\phi) - K_3\dot{z}\right)}{m} - g. \end{cases} \tag{4}$$

Furthermore,

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \dfrac{(p\cos\theta + q\sin\phi\sin\theta + r\cos\phi\sin\theta)}{\cos\theta} \\ q\cos\phi + r\sin\phi \\ \dfrac{(q\sin\phi + r\cos\phi)}{\cos\theta} \end{bmatrix}, \tag{5}$$

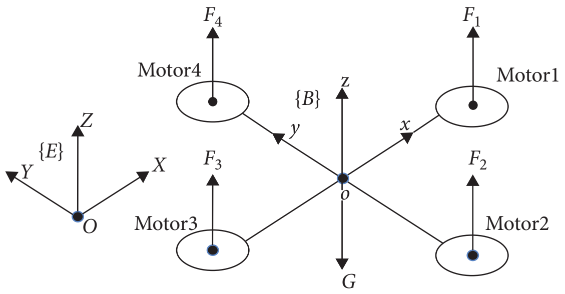


Figure 1: Rigid body force diagram of a quadrotor.

The angular motion equations can be derived by calculating angular momentum as follows:

$$\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \dfrac{\left[M_x + (I_x - I_z)\dot{\theta}\dot{\psi}\right]}{I_x} \\ \dfrac{\left[M_y + (I_z - I_x)\dot{\psi}\dot{\phi}\right]}{I_y} \\ \dfrac{\left[M_z + \left(I_x - I_y\right)\dot{\phi}\dot{\theta}\right]}{I_z} \end{bmatrix}, \qquad (6)$$

where M_x, M_y, and M_z are three-axial angular motion and I_x, I_y, and I_z are the moments of inertia.

Define u_1, u_2, u_3, and u_4 as control inputs for four independent channels of the quadrotor, respectively:

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} F_1 + F_2 + F_3 + F_4 \\ F_4 - F_2 \\ F_3 - F_1 \\ F_2 + F_4 - F_3 - F_1 \end{bmatrix} = \begin{bmatrix} k_t \sum_{i=1}^{4} \omega_i^2 \\ k_t \left(\omega_4^2 - \omega_2^2\right) \\ k_t \left(\omega_3^2 - \omega_1^2\right) \\ k \quad {}_d \left(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2\right) \end{bmatrix}, \qquad (7)$$

where u_1 is the vertical speed control amount, u_2 is the roll input control amount, u_3 is the pitch control amount, u_4 is the yaw control amount, and ω is the rotor speed.

The linear motion equation and the angular motion equation can be combined to obtain the nonlinear motion equation of the quadrotor. The mathematical model is described as follows:

$$\begin{aligned} \ddot{x} &= \frac{(\cos\psi \sin\theta \cos\phi + \sin\psi \sin\phi)u_1}{m}, \\ \ddot{y} &= \frac{(\sin\psi \sin\phi \cos\phi - \cos\psi \sin\phi)u_1}{m}, \\ \ddot{z} &= \frac{(\cos\phi \cos\theta)u_1}{m} - g, \\ \ddot{\phi} &= \frac{\left[lu_2 + \dot{\theta}\dot{\psi}\left(I_y - I_z\right)\right]}{I_x}, \\ \ddot{\theta} &= \frac{\left[lu_3 + \dot{\phi}\dot{\psi}\left(I_z - I_x\right)\right]}{I_y}, \\ \ddot{\psi} &= \frac{\left[u_4 + \dot{\phi}\dot{\theta}\left(I_x - I_y\right)\right]}{I_z}, \end{aligned} \qquad (8)$$

where l is the distance from the center of the rotor to the origin of the airframe coordinate system and k_i is the wind resistance coefficient. In the case of no wind and slow flight, the resistance coefficient is neglected [13, 14].

The system can be decoupled into independent subsystems including linear motion and angular motion. Furthermore, angular motion is not affected by linear motion. However, linear motion is affected by angular motion. When ignoring the additional small perturbations, the equation of motion of the quadrotor (8) can be obtained as follows [14]:

$$\begin{aligned} m\dot{\mathbf{x}} &= Ax + Bu, \\ x &= \left[\dot{x} \quad \dot{y} \quad \dot{z} \quad \dot{\theta} \quad \dot{\phi} \quad \dot{\psi} \quad \theta \quad \phi \quad \psi\right]^T, \\ u &= \left[u_1 \quad u_2 \quad u_3 \quad u_4\right]^T. \end{aligned} \qquad (9)$$

According to the literature [14], the parameters about the quadrotor is demonstrated in Table 1 and the corresponding continuous transfer functions for each control channel of the quadrotor system is listed in Table 2. The calculated discretized transfer functions for each control channel of the quadrotor system are listed in Table 3.

Table 1: Quadrotor parameters [14].

Parameters	Values
m (kg)	1.2
l (m)	0.2
$k_t \times 10^{-5}$ (N·s^2)	3.13
$k_d \times 10^{-7}$ (N·s^2)	7.5
$I_x \times 10^{-3}$ (kg·m^2)	2.353
$I_y \times 10^{-7}$ (kg·m^2)	2.353
$I_z \times 10^{-7}$ (kg·m^2)	5.262

Table 2: Continuous transfer function of each channel [14].

Channels	Transfer functions
Pitch channel	$G_1 = (\theta/u_1) = (56.95s + 4391/s^3 + 105s^2 + 870s + 4430)$
Roll channel	$G_2 = (\phi/u_2) = (65s + 4560/s^3 + 109s^2 + 1023s + 2935)$
Yaw channel	$G_3 = (\psi/u_3) = (105/s^2 + 413s)$
Z-axis motion	$G_4 = (z/u_4) = (1.63/s^2 + 5s)$

2.2. U-Model-Based Quadrotor UAV Model

2.2.1. Preliminary about U-Model. It is known that the controlled plant is considered as a unit when using the U-model to design a new controller. Then, according to the performance requirements for the plant, the new controller is formed by multiplication between the general controller and the inverse of the plant. Furthermore, the U-model is an expression for a class of smooth nonlinear objects. The nonlinear model for the system output can be expressed as a polynomial with $u(k-1)$[3]:

$$y(k) = \sum_{j=0}^{M} \alpha_j(k) u^j(k-1) + e(k), \tag{10}$$

where k is the time, $k \in N^+$, and M is the order of the model input $u(k-1)$. Parameter $\alpha_j(k)$ is a function of inputs $u(k-2), \ldots, u(k-n)$ and outputs. The errors, $e(k), \ldots, e(k-n)$, are unknown and unmeasured quantities. The control portion can be expressed as a power series of inputs $u(k-1)$ with time-varying parameter $\alpha_j(k)$. In order to apply the linear control design method to obtain the controller outputs, it can be further converted into the current form, that is,

$$U(k) = \sum_{j=0}^{M} \alpha_j(k) u^j(k-1) + e(k). \tag{11}$$

Because the U-model still keeps the characteristics of the controlled plant during inversion processes, it can improve the controller design efficiency.

Combining the U-model as mentioned in equation (10) with the pole placement described in the literature [15, 16] in detail, the method of designing the U-model-based pole placement controller is presented below according to each part of Figure 2.

Step 1. Regard the controlled plant as a unit. $U(k) = y(k) = y_d(k)$,

$$\begin{aligned} Ry_d(k) &= Ow(k) - Sy(k), \\ y_d(k) &= \frac{O}{R+S} w(k) = \frac{O}{A_c} w(k), \end{aligned} \tag{12}$$

$$R + S = A_c, \tag{13}$$

$$O = A_c(1), \tag{14}$$

$$\begin{aligned} R &= h^a + r_1 h^{a-1} + \cdots + r_a, \\ O &= o_0 h^b + o_1 h^{b-1} + \cdots + o_b, \\ S &= s_0 h^d + s_1 h^{d-1} + \cdots + s_d, \end{aligned} \tag{15}$$

where $w(k)$ is the reference input of the system. R, O, and T are polynomials about the forward operator h. a, b, and d are the highest powers in the forward operator in R, O, and S, respectively, and should satisfy $d < a$ and $b \leq a$. Parameters $r_{p1}, r_{p2}, r_{p3} \in R$, where $p1 = 1, \ldots, a$, $p2 = 1, \ldots, b$, and $p3 = 1, \ldots, c$. A_c is the characteristic equation polynomial of the closed-loop system, and y_d is the expected output of the system. A_c needs to be set during the design process according to the performance requirements of the actual controlled object.

Step 2. Calculate the general controller output, $U(k)$, and use the Newton–Raphson algorithm to solve the output $u(k-1)$ of the actual controller:

$$\begin{aligned} u_{\lambda+1}(k-1) &= u_\lambda(k-1) - \frac{\Theta[U_\lambda(k-1)] - U(k)}{\mathrm{d}\Theta[u(k-1)]/\mathrm{d}u(k-1)} \\ &= u_\lambda(k-1) - \left.\frac{\sum_{j=0}^{M} \alpha_j(k) u_\lambda^j(k-1) - U(k)}{\mathrm{d}\left[\sum_{j=0}^{M} \alpha_j(k) u_\lambda^j(k-1)\right]/\mathrm{d}u(k-1)}\right|_{u^j(k-1) = u_\lambda^j(k-1)}. \end{aligned} \tag{16}$$

Table 3: Discrete transfer function of each channel.

Channels	Transfer functions
Pitch channel	$G_1' = (0.1753\mathrm{z}^2 + 0.1199\mathrm{z} - 5.78 \times 10^{-4}/\mathrm{z}^3 - 1.128\mathrm{z}^2 + 0.4256z - 2.754 \times 10^{-5})$
Roll channel	$G_2' = (0.1763\mathrm{z}^2 + 0.1083\mathrm{z} - 7.907 \times 10^{-4}/\mathrm{z}^3 - 1.184\mathrm{z}^2 + 0.3666z - 1.846 \times 10^{-5})$
Yaw channel	$G_3' = (0.2481\mathrm{z} + 0.006/\mathrm{z}^2 + z - 1.158 \times 10^{-18})$
Z-axis motion	$G_4' = (0.006946\mathrm{z} - 5.881 \times 10^{-3}/\mathrm{z}^2 - 1.607z + 0.6065)$

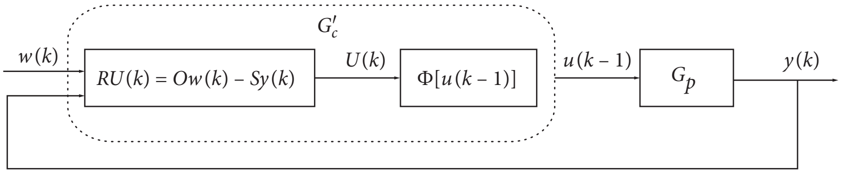


Figure 2: Design flow of the pole configuration controller [3].

Step 3. Regard $G_c G_p^{-1} = G_c'$ as the new controller of G_p. Now, $u(k-1)$ is the input of G_p which has been converted to U-model form.

2.2.2. Quadrotor UAV Controller Design. For the abovementioned quadrotor flight control system, a controller with a natural frequency of 1 rad/s and a damping ratio of 0.7 is designed to achieve zero steady-state error. The closed-loop characteristic equation is designed as

$$A_c = h^2 - 1.3205h + 0.4966. \tag{17}$$

Therefore, substitute $h = 1$ into (17), and then it is obtained from (14) that

$$O = A_c(1) = 1 - 1.3205 + 0.4966 = 0.1761. \tag{18}$$

So, the polynomials R and S can be expressed as

$$\begin{aligned} R &= h^2 + r_1 h + r_2, \\ S &= s_0 h + s_1. \end{aligned} \tag{19}$$

Substituting the characteristic equations (17) and (19) into (13), equation (12) can be derived:

$$\begin{aligned} r_2 + s_1 &= 0.4966, \\ r_1 + s_0 &= -1.3205. \end{aligned} \tag{20}$$

To ensure the convergence of $U(k)$, let $r_1 = -0.9$ and $r_2 = 0.009$, and then obtain $s_0 = -0.4205$ and $s_1 = 0.4876$. Therefore, the output of the universal controller is

$$\begin{aligned} U(k+1) = {} & 0.9U(k) - 0.009U(k-1) + 0.1761w(k-1) \\ & + 0.4205y(k) - 0.4876y(k-1). \end{aligned} \tag{21}$$

Design controllers for each of the four channels with the abovementioned universal controller, respectively.

(1) Convert the discretized transfer function of the pitch channel to a polynomial form as

$$\begin{aligned} y_1(k) = {} & 1.128y(k-1) - 0.4256y(k-2) + 2.754 \times 10^{-5} y(k-3) \\ & + 0.1753u(k-1) + 0.1199u(k-2) - 5.78 \times 10^{-4} u(k-3). \end{aligned} \tag{22}$$

The corresponding U-model form is

$$\begin{aligned} \alpha_{10} = {} & 1.128y(k-1) - 0.4256y(k-2) \\ & + 2.754 \times 10^{-5} y(k-3) + 0.1199u(k-2) - 5.78 \\ & \times 10^{-4} u(k-3), \\ \alpha_{11} = {} & 0.1753. \end{aligned} \tag{23}$$

The sample time is set 0.001s in this paper. The U-model-based pole placement method is used to design the output and the input responses of the pitch channel as shown in Figures 3(a) and 3(b). It can be seen from Figure 3(a) that the system has a small overshoot, almost zero steady-state error, and a fast response speed, which meet the performance requirements of a quadrotor during flight. It can be seen from Figure 3(b) that the controller designed according to the idea of U-model simplifies the design process of the controller while ensuring the uniqueness of the controller.

(2) Convert the discretized transfer function of the rollover channel to a polynomial form as

$$\begin{aligned} y_2(k) = {} & 1.184y(k-1) - 0.3666y(k-2) + 1.846 \\ & \times 10^{-5} y(k-3) + 0.1763u(k-1) + 0.1083u(k-2) \\ & - 7.907 \times 10^{-4} u(k-3). \end{aligned} \tag{24}$$

The corresponding U-model form is

$$\begin{aligned} \alpha_{20}(k) = {} & 1.184y(k-1) - 0.3666y(k-2) + 1.846 \\ & \times 10^{-5} y(k-3) + 0.1083u(k-2) - 7.907 \\ & \times 10^{-4} u(k-3), \\ \alpha_{21} = {} & 0.1763. \end{aligned} \tag{25}$$

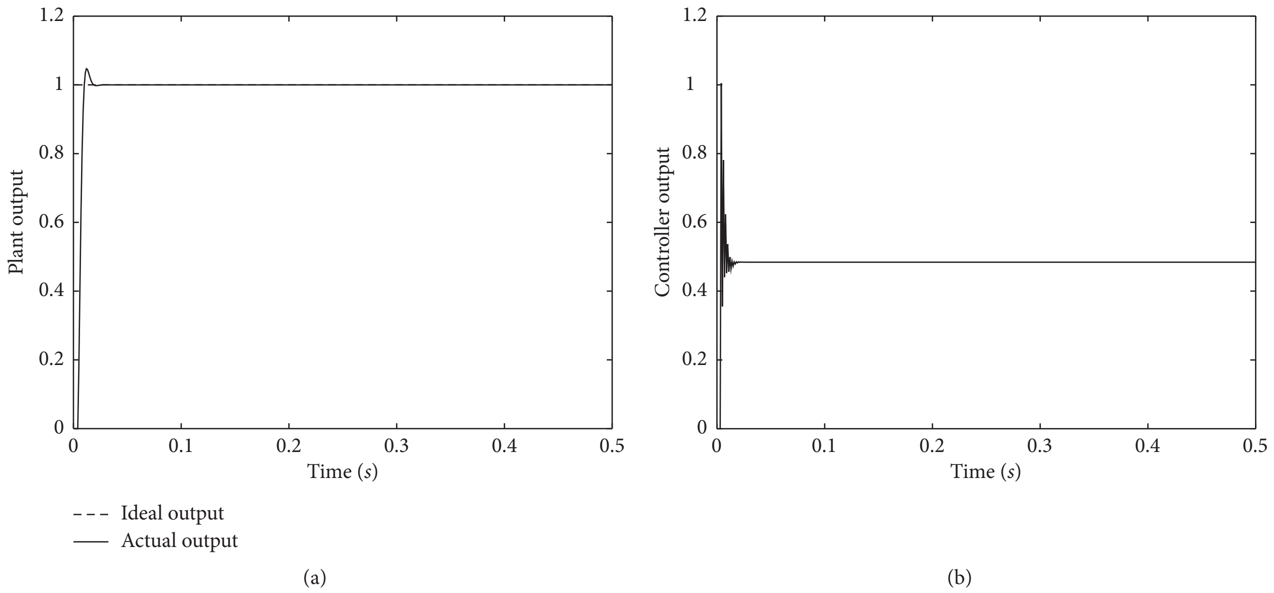


Figure 3: (a) Output of the pitch channel. (b) Input of the pitch channel.

The output and the input responses of the roll channel are designed using the U-model-based pole placement method as shown in Figures 4(a) and 4(b). The comparison between Figures 4(a) and 3(a) shows that the output performance of the controlled object is the same. At the same time, it is found that the input of the controlled object is different, which proves that the designed controller is unique when comparing Figure 4(b) to Figure 3(b).

(3) Convert the discretized transfer function of the yaw channel to a polynomial form as

$$y_3(k) = -y(k-1) + 1.158 \times 10^{-18} y(k-2) + 0.2481u(k-1) + 0.006u(k-2). \tag{26}$$

The corresponding U-model form is

$$\alpha_{30}(k) = -y(k-1) + 1.158 \times 10^{-18} y(k-2) + 0.006u(k-2),$$
$$\alpha_{31}(k) = 0.2481. \tag{27}$$

The U-model-based pole placement method is used to design the output and the input responses of the yaw channel as shown in Figures 5(a) and 5(b). The output of the controlled object is also the same as those in Figures 3(a) and 4(a); however, the output of the controller is different from those in Figures 3(b) and 4(b).

(4) Convert the discretized transfer function of the motion in the Z-axis direction into a polynomial form as

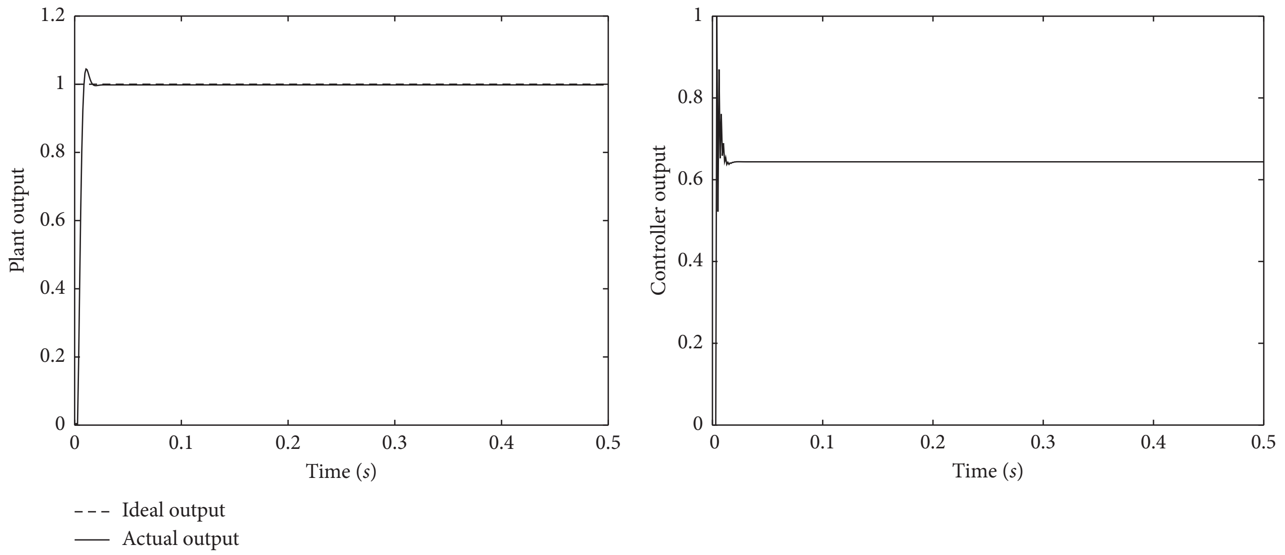
$$y_4(k) = 1.607y(k-1) - 0.6065y(k-2) + 6.946u(k-1) - 5.881 \times 10^{-3} u(k-2). \tag{28}$$

The corresponding U-model form is

$$\alpha_{40} = 1.607y(k-1) - 0.6065y(k-2) - 5.881 \times 10^{-3} u(k-2),$$
$$\alpha_{41} = 6.946. \tag{29}$$

The U-model-based pole placement method is used to design the output and the input responses of the Z-axis motion as shown in Figures 6(a) and 6(b). The output of the controlled object is also the same as those in Figures 3(a), 4(a), and 5(a), and the inputs of the controlled objects are different.

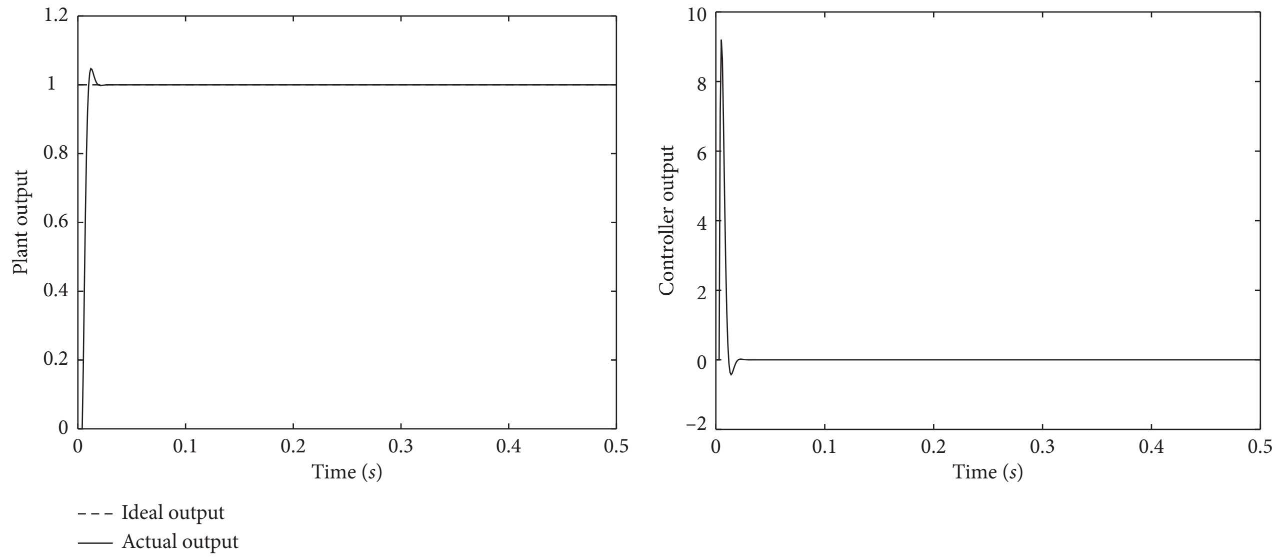
By comparing Figures 3(a)–6(a) for simulation results, it can be seen that the output responses of the controlled object are the same when the U-model controller is used to control the controlled object. Comparing Figures 3(b)–6(b), it can be seen that the designed controller outputs are different. The newly designed controller developed by using the U-model of the object is formed by $G_c G_p^{-1}$. Because G_p^{-1} varies depending upon the different plants, the newly designed controller issue is converted into calculating the inverse of the plant which simplifies the design process, meanwhile ensuring the performance requirements. Therefore, each newly designed controller works for the specific object.



(a)

(b)

Figure 4: (a) Output of the roll channel. (b) Input of the roll channel.



(a)

(b)

Figure 5: (a) Output of the yaw channel. (b) Input of yaw channel.

3. U-Block-Based Fault-Tolerant Control

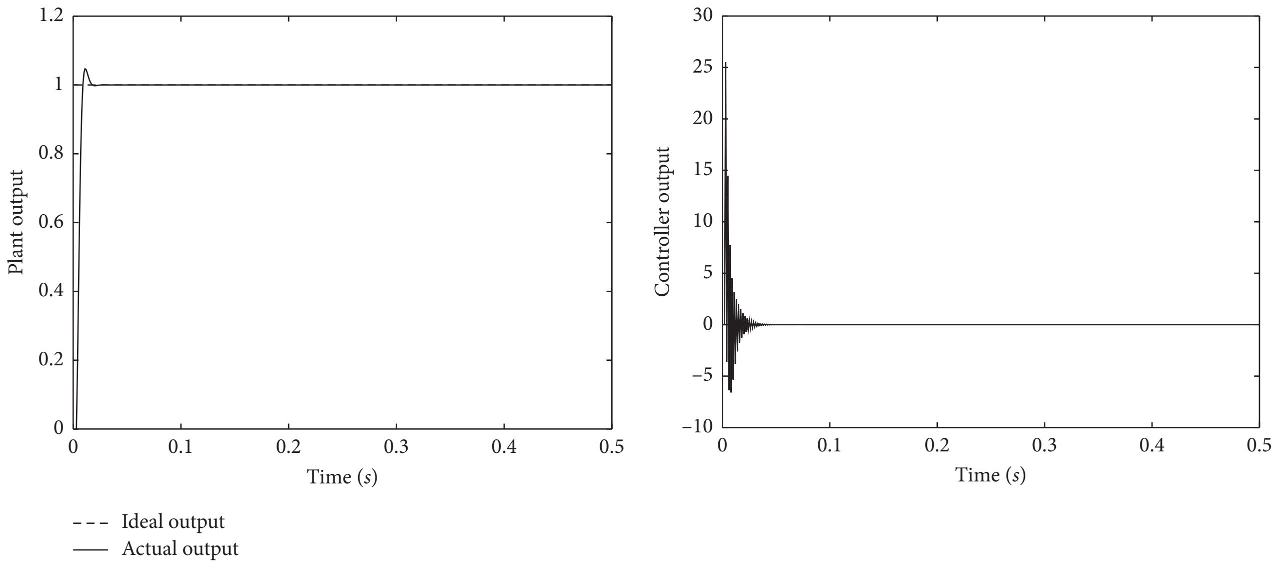
The abovementioned U-model method also works in state space form by using the method proposed in the literature [17]. This section discusses a U-model-based fault-tolerant control method employed in state space.

According to the specified characteristic equation A_c proposed in the U-model and the performance requirements of different controlled objects, A_c can be derived by placing reasonable poles:

$$A_c = (h - \vartheta_1) \cdots (h - \vartheta_t) = h^t + a_{c1}h^{t-1} + \cdots + a_{ct} = 0,$$
$$\frac{Y(h)}{w(h)} = \frac{A_c(1)}{h^t + a_{c1}h^{t-1} + \cdots + a_{ct}}. \quad (30)$$

Therefore, the corresponding state equation form in a controllable implementation is

$$\begin{aligned} x(k+1) &= Ax(k) + Bw(k), \\ y(k) &= Cx(k). \end{aligned} \quad (31)$$



(a)

(b)

FIGURE 6: (a) Z-axis motion output. (b) Z-axis motion input.

Furthermore,

$$\begin{bmatrix} x_1(k+1)\\ x_2(k+1)\\ \vdots \\ x_{t-1}(k+1)\\ x_t(k+1)\end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0\\ 0 & 0 & 1 & \cdots & 0\\ \vdots & & & & \\ 0 & 0 & 0 & \cdots & 1\\ -a_{ct} & -a_{c(t-1)} & -a_{c(t-2)} & \cdots & -a_{c1}\end{bmatrix} \begin{bmatrix} x_1(k)\\ x_2(k)\\ \vdots \\ x_{t-1}(k)\\ x_t(k)\end{bmatrix} + \begin{bmatrix} 0\\ 0\\ \vdots\\ 0\\ A_c(1)\end{bmatrix} w(k)$$

$$y(k) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0\end{bmatrix} \begin{bmatrix} x_1(k)\\ x_2(k)\\ \vdots \\ x_{t-1}(k)\\ x_t(k)\end{bmatrix}. \qquad (32)$$

When an unmanned aerial vehicle drone fails, the controlled object model can be expressed as

$$\begin{cases} \dot{x}(k) = Ax(k) + Bu(k) + f(k,x,u), \\ y(k) = Cx(k). \end{cases} \qquad (33)$$

The function $f(k,x,u)$ indicates that the actuator fails and is defined as

$$f(k,x,u) = B\xi(k,x,u), \qquad (34)$$

where $\xi(k,x,u)$ is an unknown and bounded function:

$$\|\xi(k,x,u)\| \le \gamma\|u\| + \varsigma(k,x), \qquad (35)$$

in which $\varsigma(k,x)$ indicates disturbances and $0 \le \gamma < 1$ indicates the damage degree of actuator failure [18]. So, system (33) can be further described as

$$\begin{cases} \dot{x}(k) = Ax(k) + B(I-\Gamma)u(k) + B\alpha(k,x), \\ y(k) = Cx(k), \end{cases} \qquad (36)$$

where $\Gamma = \operatorname{diag}(\gamma_1, \gamma_2, \ldots, \gamma_m)$ and γ_i is a scalar and satisfies

$$\begin{cases} \gamma_i = 0, & \text{no fault}, \\ 0 < \gamma_i < 1, & \text{fault on } i\text{th operation surface}, \end{cases} \qquad (37)$$
$$i = 1, 2, \ldots, m.$$

It is known that sliding mode control can realize the insensitivity and robustness of sliding mode for a kind of uncertainty and interference. Therefore, this paper employs the sliding mode control (SMC) method to design a U-model-based controller to improve the robustness of control systems subject to faults. For the second-order discrete system, let the position command be $w(k)$ and $dw(k)$ is the derivative of $w(k)$. Take $W = \begin{bmatrix} w(k) & dw(k) \end{bmatrix}^T$

and $W_1 = \left[w(k+1) \;\; dw(k+1) \right]^T$, and use linear extrapolation to predict $w(k+1)$ and $dw(k+1)$, *s.t.*

$$\begin{aligned} w(k+1) &= 2w(k) - w(k-1), \\ dw(k+1) &= 2\,dw(k) - dw(k-1). \end{aligned} \tag{38}$$

Design a sliding mode surface function as

$$s(k) = C_e(W(k) - x(k)), \tag{39}$$

where $C_e = \left[c \;\; 1 \right]$.

$$\begin{aligned} s(k+1) &= C_e(W(k+1) - x(k+1)) = C_e(W(k+1) \\ &\quad - Ax(k) - Bu(k)) \\ &= C_e\left(W(k+1) - C_eAx(k) - C_eBu(k)\right). \end{aligned} \tag{40}$$

Therefore, the designed control law is

$$u(k) = \left(C_eB\right)^{-1}\left(C_eW(k+1) - C_eAx(k) - s(k+1)\right). \tag{41}$$

For continuous sliding mode variable structure control, the commonly used approach is the exponential approach law:

$$\dot{s}(t) = -\varepsilon \operatorname{sgn}(s(t)) - \sigma s(t), \quad \varepsilon > 0, \sigma > 0. \tag{42}$$

Correspondingly, the discrete exponential approach law can be derived as

$$\begin{aligned} \frac{s(k+1) - s(k)}{T_s} &= -\varepsilon \operatorname{sgn}(s(k)) - \sigma s(k), \\ s(k+1) - s(k) &= -\sigma T_s s(k) - \varepsilon T_s \operatorname{sgn}(s(k)), \end{aligned} \tag{43}$$

where $\varepsilon > 0$, $\sigma > 0$, $1 - \sigma T_s > 0$, and T_s is the sampling period.

So, the discrete approach law is

$$s(k+1) = s(k) + T_s(-\varepsilon \operatorname{sgn}(s(k)) - \sigma s(k)). \tag{44}$$

Substituting (44) into equation (40), the discrete control law, $u(k)$, based on the exponential approach is obtained:

$$u(k) = \left(C_eB\right)^{-1}\left(C_eW(k+1) - C_eAx(k) - s(k) - ds(k)\right), \tag{45}$$

where $ds(k) = -\varepsilon T_s \operatorname{sgn}(s(k)) - \sigma T_s s(k)$, and $C_e = \left[c \;\; 1 \right]$.

Proof. To verify the stability of the selected sliding surface, the Lyapunov function is designed as

$$V(k) = \frac{1}{2}s(k)^2. \tag{46}$$

When the condition $\dot{V}(k) \leq 0$ is satisfied, the conclusion that the system is stable can be drawn:

$$\begin{aligned} \dot{V}(k) &= s(k)\dot{s}(k) \\ &= s(k)\left(\frac{s(k+1) - s(k)}{T_s}\right) \\ &= \frac{1}{T_s}s(k)\left[-\sigma T_s s(k) - \varepsilon T_s \operatorname{sgn}(s(k))\right] \\ &= -\sigma s^2(k) - \varepsilon s(k) \operatorname{sgn}(s(k)) \\ &\leq -\sigma s^2(k) - \varepsilon |s(k)|. \end{aligned} \tag{47}$$

Since $\varepsilon > 0$ and $\sigma > 0$, then $\dot{V}(k) \leq 0$ is obtained.

4. Analysis of Simulation Results

When partial actuator failures occur in the yaw channel, the discrete sliding mode control law is applied for designing a fault-tolerant controller for each channel, respectively. The discrete yaw channel model is transformed into a U-block form using the pole placement method. According to equations (26), (44), and (45), the corresponding state space equations are derived and the corresponding model matrices are

$$\begin{aligned} A &= \begin{bmatrix} 0 & 1 \\ -0.4966 & 1.3205 \end{bmatrix}, \\ B &= \begin{bmatrix} 0 \\ 0.1761 \end{bmatrix}, \\ C &= \begin{bmatrix} 1 & 0 \end{bmatrix}. \end{aligned} \tag{48}$$

The parameters of the controller according to (44) are $c = 50$, $\varepsilon = 5$, and $\sigma = 150$.

This paper conducts simulations when faults are inserted at time 2 s demonstrated in Figures 7(a)-7(b). The simulation selects the sampling period, 0.001 s. That is, the control system operates in normal conditions from 0 s to 2 s. However, at time 2 s, the actuator experienced a 50% performance failure. Meanwhile, the designed controller brings about a great tracking performance.

As illustrated in Figures 7(a) and 7(b), it can be seen that the design controller works properly to track the desired output of the system. Even though the actuator loses 50% operation efficiency at 2 s, it still maintains good tracking performance. Figure 7(b) illustrates the control error between the actual and expected outputs and obviously shows that the error falls within 5% and satisfies the performance requirements. The simulation results verify the feasibility of the designed controller by using the U-model and, furthermore, demonstrate that the U-model-based controller can simplify the controller design process. When combining the sliding mode control method, the proposed controller presents good fault-tolerance performance, good stability, and tracking performance.

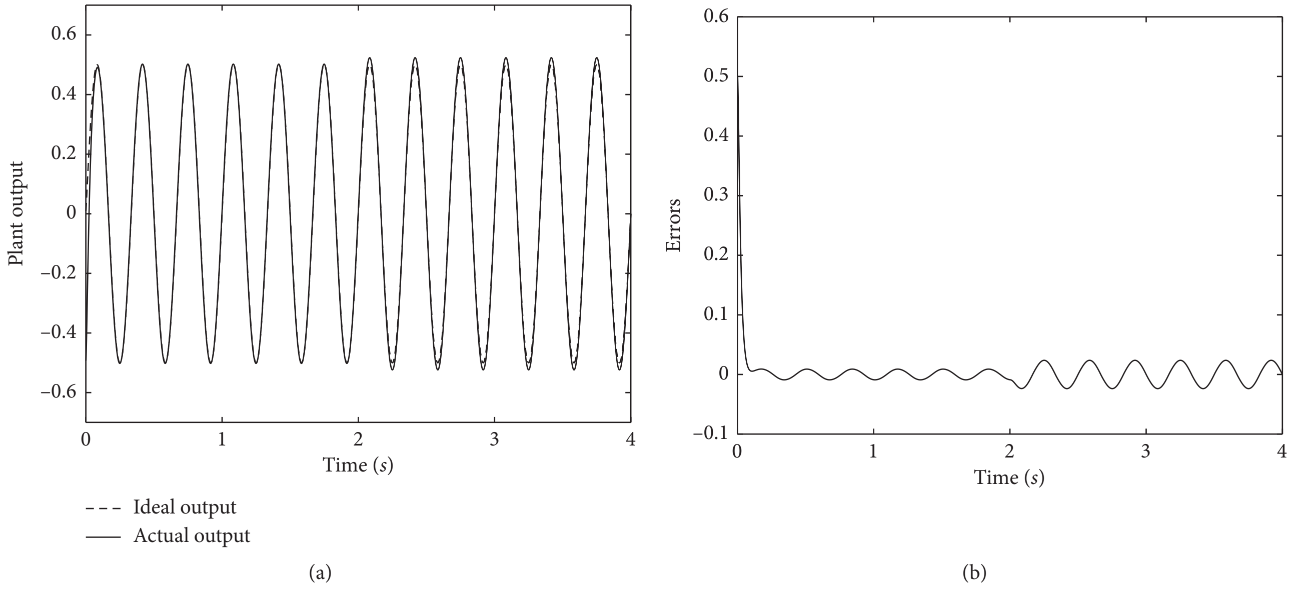


Figure 7: (a) Comparison between the actual output and the expected output when a fault occurs at 2 s. (b) Error between actual output and expected output.

5. Conclusions and Future Works

In this paper, the transfer function of each channel of the quadrotor control system is decoupled and converted into the U-block form by using the pole placement method. Furthermore, the actuator failure along with partial performance loss is considered and U-model-based sliding mode controller is designed and applied to a four-rotor flight control system in this paper. The simulations verify that the designed controller provides great fault-tolerant performance for quadrotor systems. The method proposed in this paper not only guarantees the performance requirements and simplifies the controller design process but also can be extended to other plants with the same performance requirements. This method can be mixed with other controller design methods to improve the stability of the system when suffering a fault as well in the future. The physical verification of this method will also be considered and conducted in the near future.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Fundamental Research Funds for the Central Universities-Civil Aviation University of China (3122019044).

References

[1] F. Kendoul, "Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems," *Journal of Field Robotics*, vol. 29, no. 2, pp. 315–378, 2012.

[2] L. Zhou, J. Zhang, H. She, and H. Jin, "Quadrotor UAV flight control via a novel saturation integral backstepping controller," *Automatika*, vol. 60, no. 2, pp. 193–206, 2019.

[3] Q. M. Zhu and L. Z. Guo, "A pole placement controller for non-linear dynamic plants," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 216, no. 6, pp. 467–476, 2002.

[4] F. Chen, D. Lu, and X. Li, "Robust observer based fault-tolerant control for one-sided lipschitz markovian jump systems with general uncertain transition rates," *International Journal of Control, Automation and Systems*, vol. 17, no. 7, pp. 1614–1625, 2019.

[5] H. Castañeda and J. Gordillo, "Spatial modeling and robust flight control based on adaptive sliding mode approach for a quadrotor MAV," *Journal of Intelligent & Robotic Systems: Theory & Applications*, vol. 93, no. 1-2, pp. 101–111, 2019.

[6] A.-R. Merheb, H. Noura, and F. Bateman, "Design of passive fault-tolerant controllers of a quadrotor based on sliding mode theory," *International Journal of Applied Mathematics and Computer Science*, vol. 25, no. 3, pp. 561–576, 2015.

[7] D.-T. Nguyen, D. Saussié, and L. Saydy, "Robust self-scheduled fault-tolerant control of a quadrotor UAV," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 5761–5767, 2017.

[8] T. Li, Y. Zhang, and B. W. Gordon, "Nonlinear fault-tolerant control of a quadrotor UAV based on sliding mode control technique," *IFAC Proceedings Volumes*, vol. 45, no. 20, pp. 1317–1322, 2012.

[9] Y. Zhang, X. Yu, B. Wang, and D. Liu, "Design and implementation of fault-tolerant control algorithms for an unmanned quadrotor system," *Control Engineering*, vol. 23, no. 12, pp. 1874–1882, 2016, in Chinese.

[10] Z. Liu, C. Yuan, X. Yu, and Y. Zhang, "Retrofit fault-tolerant tracking control design of an unmanned quadrotor helicopter considering actuator dynamics," *International Journal of*

Robust and Nonlinear Control, vol. 29, no. 16, pp. 5293–5313, 2017.

[11] Y. Dong, P. Yang, W. Ma, and B. Ma, “Sliding mode robust adaptive fault-tolerant control design for uncertain time-delay systems,” in *Proceedings of the 2016 IEEE Chinese Guidance, Navigation and Control Conference (CGNCC)*, pp. 2143–2147, IEEE, Nanjing, China, August 2016.

[12] X. Gong and L. Wang, “Fault-tolerant attitude stabilization control of four rotor aircraft,” *Electro-Optic and Control*, vol. 21, no. 7, pp. 14–18, 2014, in Chinese.

[13] C. Wu, H. Wang, Y. Zhang, J. Tan, and X. Ni, “Trajectory tracking of unmanned helicopter based on LADRC,” *Journal of Aeronautics*, vol. 36, no. 2, pp. 473–483, 2014, in Chinese.

[14] J. Li and Y. Li, “Dynamic modeling and PID control of quadrotor,” *Journal of Liaoning Technical University (Natural Science Edition)*, vol. 31, no. 1, pp. 116–119, 2012, in Chinese.

[15] Q. Zhu, S. Li, and D. Zhao, “A universal U-model based control system design,” in *Proceedings of the 33rd Chinese Control Conference*, pp. 1839–1844, IEEE, Nanjing, China, July 2014.

[16] M. Paluszek and S. Thomas, *Adaptive Control*, Apress, New York, NY, USA, 2017.

[17] Q. M. Zhu, D. Y. Zhao, and J. Zhang, “A general U-block model-based design procedure for nonlinear polynomial control systems,” *International Journal of Systems Science*, vol. 47, no. 14, pp. 3465–3475, 2016.

[18] H. Alwi and C. Edwards, “Fault detection and fault-tolerant control of a civil aircraft using a sliding-mode-based scheme,” *IEEE Transactions on Control Systems Technology*, vol. 16, no. 3, pp. 499–510, 2008.