*Research Article*

# Adaptive Neural Network Dynamic Inversion with Prescribed Performance for Aircraft Flight Control

**Wendong Gai,[1,2] Honglun Wang,[2] Jing Zhang,[1] and Yuxia Li[1]**

[1] *College of Information and Electrical Engineering, Shandong University of Science and Technology, Qingdao 266590, China*
[2] *Science and Technology on Aircraft Control Laboratory, Beihang University, Beijing 100191, China*

Correspondence should be addressed to Honglun Wang; hl_wang_2002@126.com

An adaptive neural network dynamic inversion with prescribed performance method is proposed for aircraft flight control. The aircraft nonlinear attitude angle model is analyzed. And we propose a new attitude angle controller design method based on prescribed performance which describes the convergence rate and overshoot of the tracking error. Then the model error is compensated by the adaptive neural network. Subsequently, the system stability is analyzed in detail. Finally, the proposed method is applied to the aircraft attitude tracking control system. The nonlinear simulation demonstrates that this method can guarantee the stability and tracking performance in the transient and steady behavior.

## 1. Introduction

Flight control design for aircraft continues to be one of the most important problems in the world of automatic control. The problem is driven by the nonlinear and uncertain nature of aircraft dynamics. Traditionally, the solution to this problem is to design the linear controller using linearized aircraft models at multiple trimmed conditions. And this procedure is time consuming and expensive.

Control of aircraft by dynamic model inversion is well known and has been applied to the control of high angle of attack fighter aircraft [1, 2]. The primary drawback of dynamic inversion for aircraft flight control is the need for high-fidelity nonlinear model which must be inverted in real time. However, it is difficult to obtain the exact aircraft dynamic model in practice. The neural network augmented model inversion in the attitude angular loop is implemented to compensate the model inversion error, and it uses proportional-derivative desired dynamics to design the attitude control system for the helicopter [3] and tilt-rotor aircraft [4].

The asymptotic tracking can be achieved using this method. However, the transient behavior of the output signals could be oscillatory when the tracking error magnitude is decreased by increasing the adaption rate. Several solutions [5–8] have been proposed to overcome this problem. These methods guarantee the convergence of tracking error, but the required tracking error upper bounds can't be accurately computed. A new adaptive control method with prescribed performance is presented in [9], and this method guarantees the transient state tracking error in the prespecified performance bound. And this method is used to improve the performance of the planar two-link articulated manipulator [10, 11] and the 6-DOF PUMA 560 arm [12].

It is very important for aircraft to track the attitude command with a desired transient and steady performance, when the aircraft finishes the special flight tasks, such as automated aerial refueling [13, 14] and transition flight control [15, 16].

In this paper, we will investigate the aircraft attitude control problem of guaranteeing transient and steady performance in the adaptive compensation control system. By employing the prescribed performance bounds proposed in [9], we propose a new adaptive neural network dynamic inversion method. With certain transformation method, a new transformed error system is obtained through considering the prescribed performance bound into the original attitude control system. An adaptive dynamic inversion controller is designed for the transformed system. It is ensured

that the tracking error is guaranteed inside the prescribed error bound as long as the transformed error system is stable.

The paper is organized as follows: the problem and the control configuration are introduced in Section 2. Section 3 presents the adaptive neural network dynamic inversion with prescribed performance design, stability analysis, model error analysis, and neural network structure. And the simulations are described in Section 4. Finally, this paper concludes in Section 5.

## 2. Aircraft Nonlinear Attitude Angle Model

The aircraft nonlinear attitude dynamic model can be presented as

$$
\begin{aligned}
\dot{\phi} &= p + (r\cos\phi + q\sin\phi)\tan\theta, \\
\dot{\theta} &= q\cos\phi - r\sin\phi, \\
\dot{\psi} &= \frac{(r\cos\phi + q\sin\phi)}{\cos\theta}, \\
\dot{p} &= (c_1 r + c_2 p)q + c_3\bar{L} + c_4 N, \\
\dot{q} &= c_5 pr - c_6\left(p^2 - r^2\right) + c_7 M, \\
\dot{r} &= (c_8 p - c_2 r)q + c_4\bar{L} + c_9 N,
\end{aligned} \tag{1}
$$

where $\phi$, $\theta$, and $\psi$ are the roll, pitch, and yaw attitude angles. $p$, $q$, and $r$ are the roll, pitch, and yaw angular rates. $c_1,\ldots,c_9$ can be found in [17]. $\bar{L}$, $M$, and $N$ are the roll, pitch, and yaw moments, which can be described as

$$
\begin{aligned}
\bar{L} &= \frac{\rho_a V^2 SbC_l}{2}, \\
M &= \frac{\rho_a V^2 S\bar{c}C_m}{2}, \\
N &= \frac{\rho_a V^2 SbC_n}{2},
\end{aligned} \tag{3}
$$

where $\rho_a$ is the air density, $S$ is the wing reference area, $b$ is the wing span, $V$ is the flight velocity, and $\bar{c}$ is the wing mean geometric chord. $C_l$, $C_m$, and $C_n$ are the rolling, pitching, and yawing moment coefficients described as

$$
\begin{aligned}
C_l &= C_{l\beta}\beta + C_{l\bar{p}}\bar{p} + C_{l\bar{r}}\bar{r} + C_{l\delta_a}\delta_a + C_{l\delta_r}\delta_r, \\
C_m &= C_{m,\alpha=0} + C_{m\alpha}\alpha + C_{m\bar{q}}\bar{q} + C_{m\bar{\alpha}}\bar{\alpha} + C_{m\delta_e}\delta_e, \\
C_n &= C_{n\beta}\beta + C_{n\bar{p}}\bar{p} + C_{n\bar{r}}\bar{r} + C_{n\delta_a}\delta_a + C_{n\delta_r}\delta_r,
\end{aligned} \tag{4}
$$

where $C_{(*)}$ is the aerodynamic derivatives. $\alpha$ and $\beta$ are the angles of attack and sideslip. $\delta_a$, $\delta_e$, and $\delta_r$ are the aileron, elevator, and rudder deflections, which are the control actuators of the aircraft. $\bar{p}$, $\bar{q}$, $\bar{r}$, and $\bar{\alpha}$ are defined by

$$
\begin{aligned}
\bar{p} &= pb/(2V), & \bar{r} &= rb/(2V) \\
\bar{q} &= q\bar{c}/(2V), & \bar{\alpha} &= \dot{\alpha}\bar{c}/(2V)
\end{aligned} \tag{5}
$$

and $\dot{\alpha}$ is the derivative of the angle of attack.

Substituting (3)-(4) into (2), and (2) can be rewritten in the affine nonlinear form as

$$
\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} f_p \\ f_q \\ f_r \end{bmatrix} + G_u \begin{bmatrix} \delta_a \\ \delta_e \\ \delta_r \end{bmatrix}, \tag{6}
$$

where $f_q$, $f_q$, $f_r$, and $G_u$ are

$$
\begin{aligned}
f_p &= (c_1 r + c_2 p)q + c_3 M_x^0 + c_4 M_z^0, \\
f_q &= c_5 pr - c_6\left(p^2 - r^2\right) + c_7 M_y^0, \\
f_r &= (c_8 p - c_2 r)q + c_4 M_x^0 + c_9 M_z^0,
\end{aligned} \tag{7}
$$

$$
G_u = \frac{\rho_a V^2 S}{2}\begin{bmatrix} c_3 & 0 & c_4 \\ 0 & c_7 & 0 \\ c_4 & 0 & c_9 \end{bmatrix}\begin{bmatrix} bC_{l\delta_a} & 0 & bC_{l\delta_r} \\ 0 & \bar{c}C_{m\delta_e} & 0 \\ bC_{n\delta_a} & 0 & bC_{n\delta_r} \end{bmatrix}, \tag{8}
$$

and

$$
\begin{aligned}
M_x^0 &= \frac{\rho_a V^2 Sb\left(C_{l\beta}\beta + C_{l\bar{p}}\bar{p} + C_{l\bar{r}}\bar{r}\right)}{2}, \\
M_y^0 &= \frac{\rho_a V^2 S\bar{c}\left(C_{m,\alpha=0} + C_{m\alpha}\alpha + C_{m\bar{q}}\bar{q} + C_{m\bar{\alpha}}\bar{\alpha}\right)}{2}, \\
M_z^0 &= \frac{\rho_a V^2 Sb\left(C_{n\beta}\beta + C_{n\bar{p}}\bar{p} + C_{n\bar{r}}\bar{r}\right)}{2}.
\end{aligned} \tag{9}
$$

According to (1), we can derive the second derivatives of attitude angles as follows:

$$
\left[\ddot{\phi}, \ddot{\theta}, \ddot{\psi}\right]^T = L(\phi, \theta)\left[\dot{p}, \dot{q}, \dot{r}\right]^T + g\left(\phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi}\right), \tag{10}
$$

where

$$
L(\phi, \theta) = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix}, \tag{11}
$$

$$
g\left(\phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi}\right) = \begin{bmatrix} \dot{\theta}\dot{\psi}\sec\theta + \dot{\phi}\dot{\theta}\tan\theta \\ -\dot{\phi}\dot{\psi}\cos\theta \\ \dot{\phi}\dot{\theta}\sec\theta + \dot{\theta}\dot{\psi}\tan\theta \end{bmatrix}. \tag{12}
$$

Substituting (6) into (10), we obtain

$$
\left[\ddot{\phi}, \ddot{\theta}, \ddot{\psi}\right]^T = L(\phi, \theta)\left[f_p, f_q, f_r\right]^T + g\left(\phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi}\right) \\ + L(\phi, \theta)G_u[\delta_a, \delta_e, \delta_r]^T. \tag{13}
$$

## 3. Prescribed Performance-Based Adaptive Neural Network Dynamic Inversion Design

The aircraft attitude model shown in (13) can be represented in the following shorthand notation:

$$
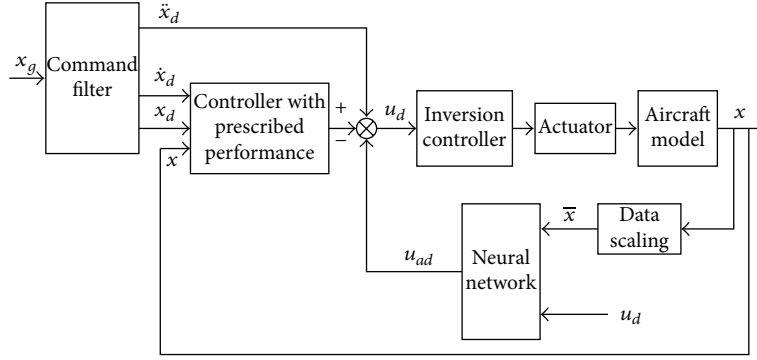\ddot{x} = f(x, \dot{x}) + g(x)u, \tag{14}
$$

FIGURE 1: Adaptive neural network dynamic inversion with prescribed performance architecture.

where the controlled state $x = [\phi, \theta, \psi]^T$ and the control vector $u = [\delta_a, \delta_e, \delta_r]^T$. $f(x, \dot{x})$ and $g(x)$ are nonlinear functions.

The state tracking error is defined as

$$e(t) = x(t) - x_d(t), \tag{15}$$

where $x_d(t)$ is the desired state vector.

The proposed control architecture of the aircraft attitude control system is shown in Figure 1.

### 3.1. Dynamic Inversion.

This section will show a brief introduction of dynamic inversion. And the readers could derive much more details from the reference [2].

We seek to linearize a nonlinear system through computing dynamic inversion to cancel the nonlinearities in the system. The aircraft dynamics are shown in (14). The number of control inputs and controlled states must be the same; that is to say, the nonlinear function $g(x)$ is invertible. Then, the control input can be calculated by

$$u_c = g^{-1}(x)(u_m - f(x, \dot{x})), \tag{16}$$

where $u_m$ is the desired response of $\ddot{x}$. Replacing the $u$ in the right of (14) by the $u_c$ from (16), we derive

$$\ddot{x} = u_m \tag{17}$$

and any nonlinearities in $f(x, \dot{x})$ and $g(x)$ are cancelled.

The achieved system dynamics will match the chosen desired dynamics when there are no errors between the design model and real object. However, the model error is inevitable. So a new method is proposed to compensate the model error and guarantee the system performances in the transient and steady behavior.

### 3.2. Performance Function and Error Transformation

*Definition 1* (see [9]). A smooth function $\rho : \mathfrak{R}_+ \rightarrow \mathfrak{R}_+$ can be called a performance function if the following conditions are satisfied:

$$\rho(t) > 0, \qquad \dot{\rho}(t) < 0,$$
$$\lim_{t \to \infty} \rho(t) = \rho_\infty > 0. \tag{18}$$

For example, a performance function is

$$\rho(t) = (\rho_0 - \rho_\infty) e^{-lt} + \rho_\infty, \tag{19}$$

where $\rho_0$, $\rho_\infty$ and $l$ are positive constants, $\rho_0$ is the initial tracking error $e(t)$, and $\rho_\infty$ is the maximum allowable tracking error $e(t)$ at the steady state. The decrement of tracking error $e(t)$ will decrease when the parameter $l$ decreases. And we can derive the first and second derivatives of $\rho(t)$ as follows:

$$\dot{\rho}(t) = -l(\rho_0 - \rho_\infty) e^{-lt},$$
$$\ddot{\rho}(t) = l^2(\rho_0 - \rho_\infty) e^{-lt}. \tag{20}$$

Then by satisfying the following condition:

$$-\underline{\delta}\rho(t) < e(t) < \overline{\delta}\rho(t), \quad \forall t \ge 0, \tag{21}$$

where $0 \le \underline{\delta}$, and $\overline{\delta} \le 1$ are prescribed scalars; the objective of guaranteeing transient and steady performance can be derived.

*Remark 2.* According to (21), $-\underline{\delta}\rho(0)$ and $\overline{\delta}\rho(0)$ are the lower bound of the negative overshoot and upper bound of the positive overshoot of $e(t)$, respectively. And a lower bound of the convergence speed of $e(t)$ is introduced by the decreasing rate of $\rho(t)$.

*Remark 3.* By changing the parameters of performance function $\rho(t)$ and the positive prescribed scalars $\underline{\delta}$, and $\overline{\delta}$, the maximum overshoot and convergence rate of $e(t)$ can be modified.

To transform the original system with the constrained tracking error performance (in (21)) into an equivalent constrained one, an error transformation is introduced. And the error transformation is defined as

$$e(t) = \rho(t) S(\varepsilon), \tag{22}$$

where $\varepsilon$ is the transformed error and a smooth and strictly increasing function $S$ has the following properties:

$$-\underline{\delta} < S(\varepsilon) < \overline{\delta}, \tag{23}$$

$$\lim_{\varepsilon \to -\infty} S(\varepsilon) = -\underline{\delta}, \qquad \lim_{\varepsilon \to +\infty} S(\varepsilon) = \overline{\delta}, \tag{24}$$

$$S(0) = 0. \tag{25}$$

According to the first property in (23) and $\rho(t) > 0$, we have

$$-\underline{\delta}\rho(t) < \rho(t)S(\varepsilon) < \overline{\delta}\rho(t). \tag{26}$$

According to (19), we obtain

$$-\underline{\delta}\rho(t) < e(t) < \overline{\delta}\rho(t). \tag{27}$$

In addition, from the third property in (25), $\lim_{t \to \infty} e(t) = 0$ can be achieved if $\lim_{t \to \infty} \varepsilon(t) = 0$ is satisfied.

Then (22) can be described as

$$\varepsilon(t) = S^{-1}\left(\frac{e(t)}{\rho(t)}\right). \tag{28}$$

**Lemma 4** (see [9]). *Consider system in (14), the transient and steady state tracking error behavior bounds described by the performance function $\rho(t)$ and the error transformation equation (22). The following results hold.*

(a) *The system in (14) is invariant under the error transformation equation (22).*

(b) *Stabilization of the transformed system using (28) is sufficient to guarantee the prescribed performance.*

In what follows, an adaptive neural network dynamic inversion method is proposed to stabilize the transformed system using (28).

### 3.3. Controller Design and Stability Analysis

*Assumption 5.* The desired states $x_d(t)$ are known bounded time functions, with known bounded derivatives.

*Assumption 6.* The states $x(t)$ of the nonlinear system in (14) are available for measurement.

We define the following error function $E_i(t)$, which describes the dynamics of the new error system using the error transformation equation (28).

$$E_i(t) = \dot{\varepsilon}_i(t) + \eta_i \varepsilon_i(t), \quad i = p, q, r, \tag{29}$$

where $\eta_i, i = p, q$, and $r$ are positive constants to be chosen.

We define

$$\lambda(t) = \frac{e(t)}{\rho(t)}. \tag{30}$$

The derivative of (28) is

$$\dot{\varepsilon} = \frac{\partial S^{-1}}{\partial \lambda}\dot{\lambda}, \tag{31}$$

where

$$\dot{\lambda} = \frac{\dot{e}(t)}{\rho(t)} - \frac{e(t)\dot{\rho}(t)}{\rho^2(t)}. \tag{32}$$

And the second derivative of (28) is

$$\ddot{\varepsilon} = \frac{\partial\left(\partial S^{-1}/\partial\lambda\right)}{\partial\lambda}\left(\frac{\dot{e}(t)}{\rho(t)} - \frac{e(t)\dot{\rho}(t)}{\rho^2(t)}\right)^2$$
$$+ \frac{\partial S^{-1}}{\partial\lambda}\left[\frac{\ddot{e}(t)}{\rho(t)} - \frac{2\dot{e}(t)\dot{\rho}(t)}{\rho^2(t)}\right. \tag{33}$$
$$\left. - \frac{e(t)\ddot{\rho}(t)}{\rho^2(t)} + \frac{2\dot{e}(t)\dot{\rho}^2(t)}{\rho^3(t)}\right].$$

Then we compute the time derivative of $E_p(t)$ for the roll error as

$$\dot{E}_p(t) = \ddot{\varepsilon}_p(t) + \eta_p \dot{\varepsilon}_p(t). \tag{34}$$

And the pitch and yaw errors are derived by the similar method.

Substituting (30)–(33) into (34), we obtain

$$\dot{E}_p = \frac{\partial S_p^{-1}}{\partial\lambda_p}\frac{1}{\rho_p(t)}\ddot{e}_p(t) + E_p^M, \tag{35}$$

where

$$E_p^M = \frac{\partial\left(\partial S_p^{-1}/\partial\lambda_p\right)}{\partial\lambda_p}\left(\frac{\dot{e}_p(t)}{\rho_p(t)} - \frac{e_p(t)\dot{\rho}_p(t)}{\rho_p^2(t)}\right)^2$$
$$+ \eta_p\frac{\partial S_p^{-1}}{\partial\lambda_p}\dot{\lambda}_p - \frac{\partial S_p^{-1}}{\partial\lambda_p}$$
$$\times \left[\frac{2\dot{e}_p(t)\dot{\rho}_p(t)}{\rho_p^2(t)} + \frac{e_p(t)\ddot{\rho}_p(t)}{\rho_p^2(t)} - \frac{2\dot{e}_p(t)\dot{\rho}_p^2(t)}{\rho_p^3(t)}\right]. \tag{36}$$

Then we can derive

$$\dot{E} = E_R\ddot{e}(t) + E_M, \tag{37}$$

where $\dot{E} = [\dot{E}_p, \dot{E}_q, \dot{E}_r]^T$, $\ddot{e}(t) = [\ddot{e}_p(t), \ddot{e}_q(t), \ddot{e}_r(t)]^T$, $E_M = [E_p^M, E_q^M, E_r^M]^T$, and $E_R$ is

$$E_R = \begin{bmatrix} E_p^R & & \\ & E_q^R & \\ & & E_r^R \end{bmatrix}, \tag{38}$$

where $E_p^R = (\partial S_p^{-1}/\partial \lambda_p)/\rho_p(t)$, $E_q^R = (\partial S_q^{-1}/\partial \lambda_q)/\rho_q(t)$, $E_r^R = (\partial S_r^{-1}/\partial \lambda_r)/\rho_r(t)$, and $\ddot{e}_p(t)$, $\ddot{e}_q(t)$, $\ddot{e}_r(t)$, are

$$\ddot{e}_p(t) = \ddot{\phi} - \ddot{\phi}_d,$$
$$\ddot{e}_q(t) = \ddot{\theta} - \ddot{\theta}_d, \tag{39}$$
$$\ddot{e}_r(t) = \ddot{\psi} - \ddot{\psi}_d.$$

To simplify the controller design progress, we linearize (2) in an equilibrium point which is the steady wings-level flight state.

$$[\dot{p}, \dot{q}, \dot{r}]^T = A_\omega [V_0 + \Delta V, \alpha_0 + \Delta \alpha, \beta_0 + \Delta \beta, p_0$$
$$+ \Delta p, q_0 + \Delta q, r_0 + \Delta r]^T \tag{40}$$
$$+ B_\omega [\delta_{a0} + \Delta \delta_a, \delta_{e0} + \Delta \delta_e, \delta_{r0} + \Delta \delta_r]^T,$$

where the $A_\omega$ and $B_\omega$ are the appropriate dimension constant matrixes, $\beta_0 = p_0 = q_0 = r_0 = \delta_{a0} = \delta_{r0} = 0$. And $V_0$, $\alpha_0$, and $\delta_{e0}$ are the flight velocity, angle of attack and elevator deflection in some equilibrium point, respectively. The symbol $\Delta$ represents the small perturbation from the equilibrium value.

According to (2), (13), (14), and (40), we can obtain

$$\ddot{x} = F(x) + G(x)\Delta u + \chi, \tag{41}$$

where $\Delta u = [\Delta \delta_a, \delta_{e0} + \Delta \delta_e, \Delta \delta_r]^T$, and

$$F(x) = [F_p, F_q, F_r]^T = g(\phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi})$$
$$+ L(\phi, \theta) A_\omega [V_0 + \Delta V, \alpha_0 + \Delta \alpha, \Delta \beta, \Delta p, \Delta q, \Delta r]^T,$$
$$G(x) = [G_p, G_q, G_r]^T = L(\phi, \theta) B_\omega$$
$$\chi = [\chi_p, \chi_q, \chi_r]^T, \tag{42}$$

where $\chi$ is the model error which will be analyzed in Section 3.4.

The formula $\ddot{x} = F(x) + G(x)\Delta u$ in (41) is named as the design model in some equilibrium point, which is different from the real nonlinear model in (14). And the difference between the design model and the nonlinear model is represented by the symbol $\chi$, which will be compensated by the adaptive neural network.

Because there are three channels in the attitude control and the form of each channel is the same, consider the following Theorem 7 for the roll channel. And the pitch and yaw channels are similar.

**Theorem 7.** *Considering Assumption 5, Assumption 6, and the nonlinear system in* (14), *all the signals are bounded, and the tracking error* $e(t)$ *satisfies the performance described by the performance function* $\rho(t)$, *if the control input of system satisfies the following formula.*

*The control input of roll channel is*

$$u_p = \delta_a = G_p^{-1}\left[-F_p - \left(E_p^R\right)^{-1}\left(E_p^M + k_p E_p\right) + \ddot{\phi}_d - u_p^{ad}\right]. \tag{43}$$

*The adaptive signal of roll channel is*

$$u_p^{ad} = w_p^T g_p. \tag{44}$$

*The neural network weight update law of roll channel is*

$$\dot{\widetilde{w}}_p = \begin{cases} \gamma_p\left(g_p(E_p)^T E_p^R + \sigma_p w_p\right), & \|E_p\| > \zeta_p, \\ 0 & \|E_p\| \leq \zeta_p, \end{cases}$$
$$\zeta_p = \frac{\|E_p^R\| h_p + \sqrt{\left(\|E_p^R\| h_p\right)^2 + k_p \sigma_p \left(w_p^{\max}\right)^2}}{2k_p}, \tag{45}$$

*where the vector* $g_p$ *is a set of basis functions to approximate the uncertainty and the neural network weight vector* $w_p$ *is the set of coefficients of each basis function in the roll channel. The adaptation gain* $\gamma_p$ *determines the learning rate of neural network. The* $\sigma_p$ *is a modification term to limit the growth of the neural network weights. The constant* $k_p$ *is positive. And the positive constant* $h_p$ *is the neural network approximate error which is bounded. The neural network weight error is*

$$\widetilde{w}_p = w_p - w_p^*, \tag{46}$$

*where the* $w_p^*$ *is the true value of the neural network weight in the roll channel.*

*Proof.* A suitable Lyapunov function of roll channel will be

$$V_p = \begin{cases} \dfrac{1}{2}\left(E_p\right)^T E_p + \dfrac{1}{2\gamma_p}\left(\widetilde{w}_p\right)^T \widetilde{w}_p, & \|E_p\| > \zeta_p, \\ \dfrac{1}{2}\left(E_{p0}\right)^T E_{p0} + \dfrac{1}{2\gamma_p}\left(\widetilde{w}_p\right)^T \widetilde{w}_p, & \|E_p\| \leq \zeta_p, \end{cases} \tag{47}$$

where $\|E_{p0}\| = \zeta_p$ and $\zeta_p$ is to be defined later.

Firstly, if $\|E_p\| > \zeta_p$ is satisfied, then the time derivative of (47) is given by

$$\dot{V}_p = \left(E_p\right)^T \dot{E}_p + \frac{1}{\gamma_p}\left(\widetilde{w}_p\right)^T \dot{\widetilde{w}}_p. \tag{48}$$

Substituting (37) into (48), we derive

$$\dot{V}_p = \left(E_p\right)^T\left[E_p^M + E_p^R\left(\ddot{\phi} - \ddot{\phi}_d\right)\right] + \frac{1}{\gamma_p}\left(\widetilde{w}_p\right)^T \dot{\widetilde{w}}_p. \tag{49}$$

Considering (41)-(42) and (49), we have

$$\dot{V}_p = \left(E_p\right)^T\left[E_p^M + E_p^R\left(F_p + G_p u_p + \chi_p - \ddot{\phi}_d\right)\right]$$
$$+ \frac{1}{\gamma_p}\left(\widetilde{w}_p\right)^T \dot{\widetilde{w}}_p. \tag{50}$$

Let the control input $u_p$ satisfy (43), then (50) can be described as

$$\dot{V}_p = \left(E_p\right)^T \left[-k_p E_p + E_p^R \left(\chi_p - u_p^{ad}\right)\right] + \frac{1}{\gamma_p} \left(\widetilde{w}_p\right)^T \dot{\widetilde{w}}_p. \quad (51)$$

Substituting (44)–(46) into (51), we obtain

$$\begin{aligned}
\dot{V}_p = &-k_p \left(E_p\right)^T E_p + \left(E_p\right)^T E_p^R \\
&\times \left(\chi_p - \left(w_p^*\right)^T g_p\right) + \sigma_p \left(\widetilde{w}_p\right)^T w_p.
\end{aligned} \quad (52)$$

By using the norms of the terms on the right side of (52), we obtain the following inequality:

$$\begin{aligned}
\dot{V}_p \leq &-k_p \left\|E_p\right\|^2 + \left\|E_p\right\| \left\|E_p^R\right\| \\
&\times \left\|\chi_p - \left(w_p^*\right)^T g_p\right\| + \sigma_p \left\|\widetilde{w}_p\right\| \left\|w_p\right\|.
\end{aligned} \quad (53)$$

In addition, the approximate error of neural network is bounded, so the following equation is satisfied:

$$\left\|\chi_p - \left(w_p^*\right)^T g_p\right\| \leq h_p. \quad (54)$$

The maximum weight of ideal neural network in the roll channel is $w_p^{\max}$, so we have

$$\left\|w_p^*\right\| \leq w_p^{\max}. \quad (55)$$

Substituting (46) and (54)-(55) into (53), we get

$$\dot{V}_p \leq -k_p \left\|E_p\right\|^2 + \left\|E_p\right\| \left\|E_p^R\right\| h_p + \sigma_p \left(w_p^{\max} \left\|w_p\right\| - \left\|w_p\right\|^2\right). \quad (56)$$

Considering (56), we obtain

$$\begin{aligned}
\dot{V}_p \leq &-k_p \left\|E_p\right\|^2 + \left\|E_p\right\| \left\|E_p^R\right\| h_p \\
&- \sigma_p \left(\left\|w_p\right\| - \frac{w_p^{\max}}{2}\right)^2 + \sigma_p \left(\frac{w_p^{\max}}{2}\right)^2.
\end{aligned} \quad (57)$$

If the system is stable, then $\dot{V}_p < 0$. And (57) can be transformed to the following formula:

$$k_p \left\|E_p\right\|^2 - \left\|E_p\right\| \left\|E_p^R\right\| h_p - \sigma_p \left(\frac{w_p^{\max}}{2}\right)^2 > 0. \quad (58)$$

Then we can derive

$$\left\|E_p\right\| > \frac{\left\|E_p^R\right\| h_p + \sqrt{\left(\left\|E_p^R\right\| h_p\right)^2 + k_p \sigma_p \left(w_p^{\max}\right)^2}}{2 k_p} = \zeta_p. \quad (59)$$

Next, if $\left\|E_p\right\| \leq \zeta_p$ is satisfied, then the time derivative of (47) is derived by

$$\dot{V}_p = \frac{1}{\gamma_p} \left(\widetilde{w}_p\right)^T \dot{\widetilde{w}}_p. \quad (60)$$

Here the weight update law is $\dot{w}_p = \dot{\widetilde{w}}_p = 0$, and $\dot{V}_p = 0$. Therefore, the system is stable, and all the signals are bounded. Considering Lemma 4, the tracking error $e(t)$ satisfied the performance described by the performance function $\rho(t)$.

This completes the proof. □

### 3.4. Analysis of the Model Error.

According to (2)–(5), the moment model is nonlinear, complicated, and must be continuously varying with the flight condition. For simplicity, the linear model of (40) in an equilibrium point is used to replace the nonlinear model of (2).

We define the model error $\Lambda = \left[\Lambda_p, \Lambda_q, \Lambda_r\right]^T$, which is the error between the linear model Equation (40) and the nonlinear model equation (6). And the $\Lambda$ is

$$\begin{aligned}
\Lambda = &\left[f_p, f_q, f_r\right]^T - A_\omega \left[V_0 + \Delta V, \alpha_0 + \Delta\alpha, \Delta\beta, \Delta p, \Delta q, \Delta r\right]^T \\
&+ G_u \left[\delta_a, \delta_e, \delta_r\right]^T - B_\omega \left[\Delta\delta_a, \delta_{e0} + \Delta\delta_e, \Delta\delta_r\right]^T.
\end{aligned} \quad (61)$$

Then (6) can be rewritten as

$$\begin{aligned}
\left[\dot{p}, \dot{q}, \dot{r}\right]^T = &A_\omega \left[V_0 + \Delta V, \alpha_0 + \Delta\alpha, \Delta\beta, \Delta p, \Delta q, \Delta r\right]^T \\
&+ B_\omega \left[\Delta\delta_a, \delta_{e0} + \Delta\delta_e, \Delta\delta_r\right]^T - \Lambda.
\end{aligned} \quad (62)$$

Substituting (62) into (10), we have

$$\ddot{x} = F(x) + G(x) \Delta u - L(\phi, \theta) \Lambda. \quad (63)$$

Comparing (63) to (41), we obtain

$$\chi = -L(\phi, \theta) \Lambda. \quad (64)$$

Therefore, the model error mainly depends on the different equilibrium points, attitude angles, actuator deflections, and so on.

### 3.5. Neural Network Structure.

The first step in determining the appropriate network structure is identifying the network inputs. Based on the analysis of model error sources described in Section 3.3, there are three main categories of inputs: the attitude angles, attitude angle rates, and actuator deflections.

A Sigma-Pi neural network [18] is used to compensate the model error $\chi$, and the basis function of the pitch channel $g_q$ is

$$g_q = \text{kron}\left(\text{kron}\left(C_{1q}, C_{2q}\right), C_{3q}\right), \quad (65)$$

where $\text{kron}(\cdot, \cdot)$ represents the Kronecker products and is defined as follows:

$$C_{1q} = \left[1, \bar{\theta}, \bar{\theta}^2\right]^T, \quad C_{2q} = \left[1, \bar{q}\right]^T, \quad C_{3q} = \left[1, \bar{\delta}_e, \bar{\delta}_e^2\right]^T, \quad (66)$$

Figure 2: Neural network structure.



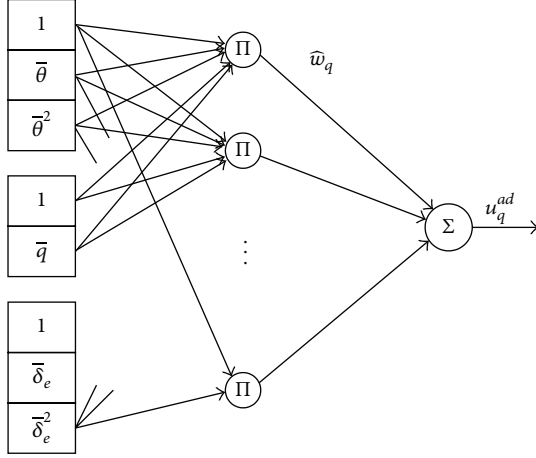Figure 3: Command filter.

where $\bar{\theta}, \bar{q}, \bar{\delta}_c$ and $\bar{\delta}_e$ are normalized variables between $-1$ and 1. The normalization function is

$$y = f(x) = \frac{2}{1 + e^{-0.1x}} - 1, \tag{67}$$

where $x$ is the input parameter and $y$ is the output parameter.

And a general description of the neural network is shown in Figure 2.

And the basis function of roll channel $g_p$ and the basis function of yaw channel $g_r$ can be derived similarly as follows:

$$g_k = \text{kron}\left(\text{kron}\left(\text{kron}\left(\text{kron}\left(C_1, C_2\right), C_3\right), C_4\right), C_k\right), \tag{68}$$

where $k = p, r$. Then $C_i, i = 1, 2, 3, 4, k$ is

$$C_1 = \left[1, \bar{\phi}, \bar{\phi}^2\right]^T, \quad C_2 = \left[1, \bar{p}\right]^T, \quad C_3 = \left[1, \bar{r}\right]^T,$$

$$C_4 = \left[1, \bar{\psi}, \bar{\psi}^2\right]^T, \quad C_p = \left[1, \bar{\delta}_a, \bar{\delta}_a^2\right]^T, \quad C_r = \left[1, \bar{\delta}_r, \bar{\delta}_r^2\right]^T. \tag{69}$$

## 4. Simulation Results

In this section, we consider the attitude angles control problem for a fixed-wing aircraft, and the initial flight state is the wings-level flight. Then the attitude angles commands in three channels will be tracked, respectively.

In the following simulation, the initial flight height and velocity are 6000 m and 190 m/s, and the initial attitude angles and angular rates including $\phi$, $\theta$, $\psi$, $p$, $q$, and $r$ are zeros. In addition, all the initial actuator deflections are zeros.

The error transformation function [19] in the simulation is described as

$$S(\varepsilon) = \frac{\bar{\delta}e^{(\varepsilon+y)} - \underline{\delta}e^{-(\varepsilon+y)}}{e^{(\varepsilon+y)} + e^{-(\varepsilon+y)}}, \tag{70}$$

where $y = \ln(\bar{\delta}/\underline{\delta})/2$. It can be shown that $S(\varepsilon)$ satisfies the properties in (23)–(25).

The attitude angles commands of three channels are transformed into the desired attitude angles commands
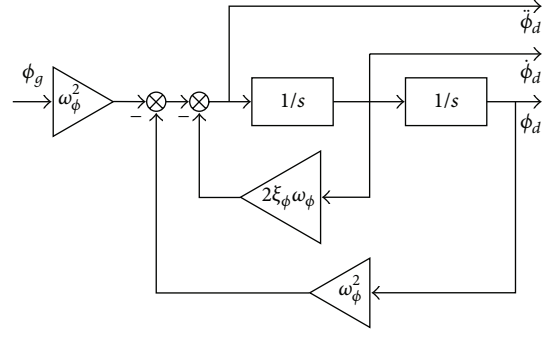
Table 1: Performance parameters.

| $\rho_0^\phi$ | $-12°$ | $\rho_0^\psi$ | $-8°$ | $\rho_0^\theta$ | $-10°$ |
|---|---|---|---|---|---|
| $\rho_\infty^\phi$ | $-0.3°$ | $\rho_\infty^\psi$ | $-0.2°$ | $\rho_\infty^\theta$ | $-0.2°$ |
| $l_\phi$ | 0.7 | $l_\psi$ | 0.7 | $l_\theta$ | 0.7 |
| $\underline{\delta}_\phi$ | 0.6 | $\underline{\delta}_\psi$ | 0.5 | $\underline{\delta}_\theta$ | 0.6 |
| $\bar{\delta}_\phi$ | 1 | $\bar{\delta}_\psi$ | 1 | $\bar{\delta}_\theta$ | 1 |

Table 2: Controller parameters.

| $k_p$ | 10 | $k_r$ | 10 | $k_q$ | 10 |
|---|---|---|---|---|---|
| $\gamma_p$ | 200 | $\gamma_r$ | 200 | $\gamma_q$ | 50 |
| $\sigma_p$ | 0.1 | $\sigma_r$ | 0.1 | $\sigma_q$ | 0.3 |

through the command filters. And the structure of command filter for the roll channel is shown in Figure 3. In addition, the desired attitude angles commands for yaw and pitch channels can be obtained by the similar command filters.

The command filter parameters are set as $\xi_i = 1$, $\omega_i = 2.5$, and $i = \phi, \psi, \theta$.

Design the control inputs with prescribed performance for three channels through the procedures in Section 3.2. The performance and controller parameters are shown in Tables 1 and 2.

*Remark 8.* For the performance function $\rho(t)$, $\rho_0$ is derived by subtracting the attitude command from the initial attitude angle. $\rho_\infty$ is the allowable attitude tracking error at the steady state. And the decrement of tracking error $e(t)$ will decrease when the parameter $l$ decreases.

*Remark 9.* For the controller parameters, the adaptation gain $\gamma$ will improve the attitude tracking performance, especially, when there are much larger model errors. The $\sigma_p$ is a modification term to limit the growth of the neural network weights; therefore, it is small. The transient performance of attitude tracking error can be improved by increasing the parameter $k$; however, the increase will increase the magnitude of the control input. Then a compromise must be reached.
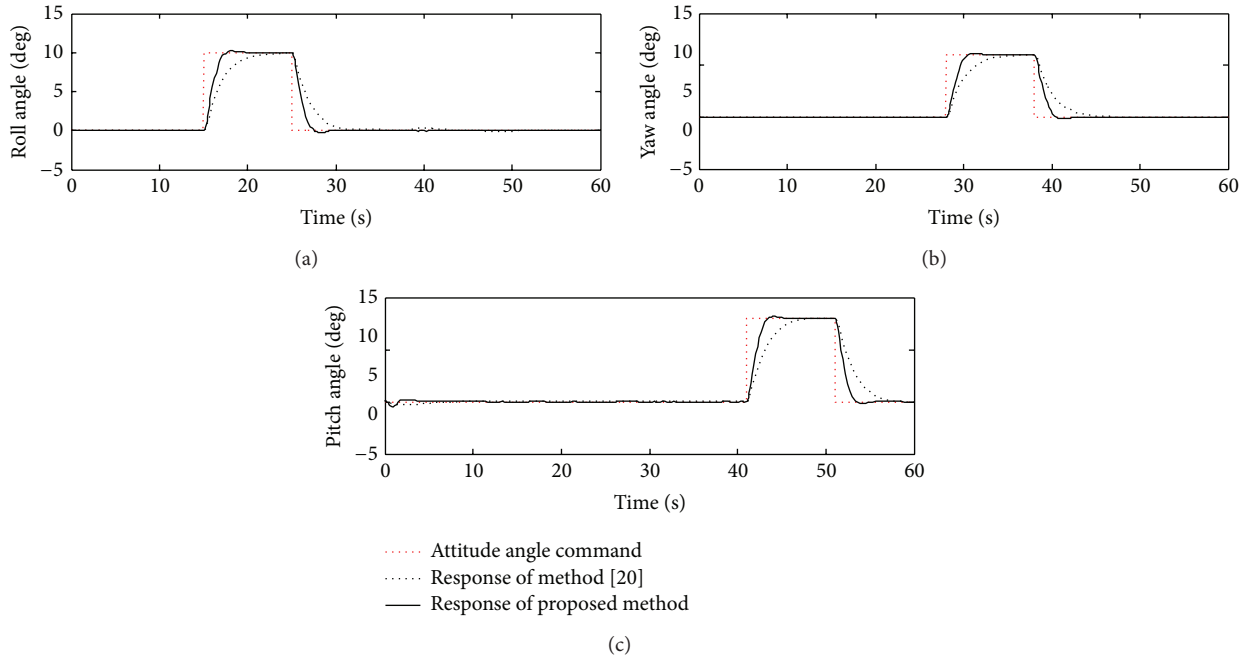
(a)



(b)



(c)

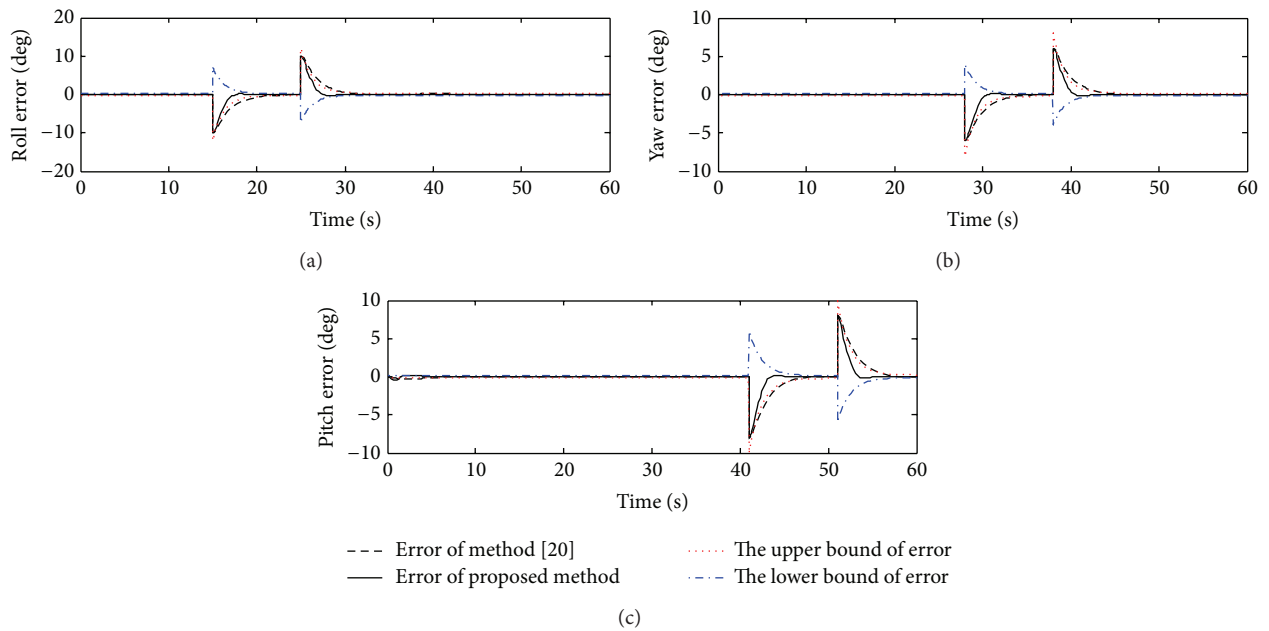FIGURE 4: Responses of the attitude angles.



(a)



(b)



(c)

FIGURE 5: Tracking errors of the attitude angles.

The design model I is derived at the trimmed flight condition of 6000 m and 190 m/s, and the model error is small.

The aircraft tracks the attitude angles commands from the initial flight state. And the attitude angles tracking responses and tracking errors are shown in Figures 4 and 5.

The two methods have achieved the attitude angles command tracking. Figure 4 shows the better attitude responses are achieved by the proposed method compared to the

method in [20]. And the coupling among different channels is smaller when the proposed method is used. For example, the roll angle response has a less change when the aircraft tracks the yaw command. In Figure 5, the attitude angles tracking errors satisfy the prescribed performance bound with the proposed method in the dynamic and steady state. The main reason is that the method in [20] does not consider the performance bound defined by the performance function $\rho(t)$ in the design process.
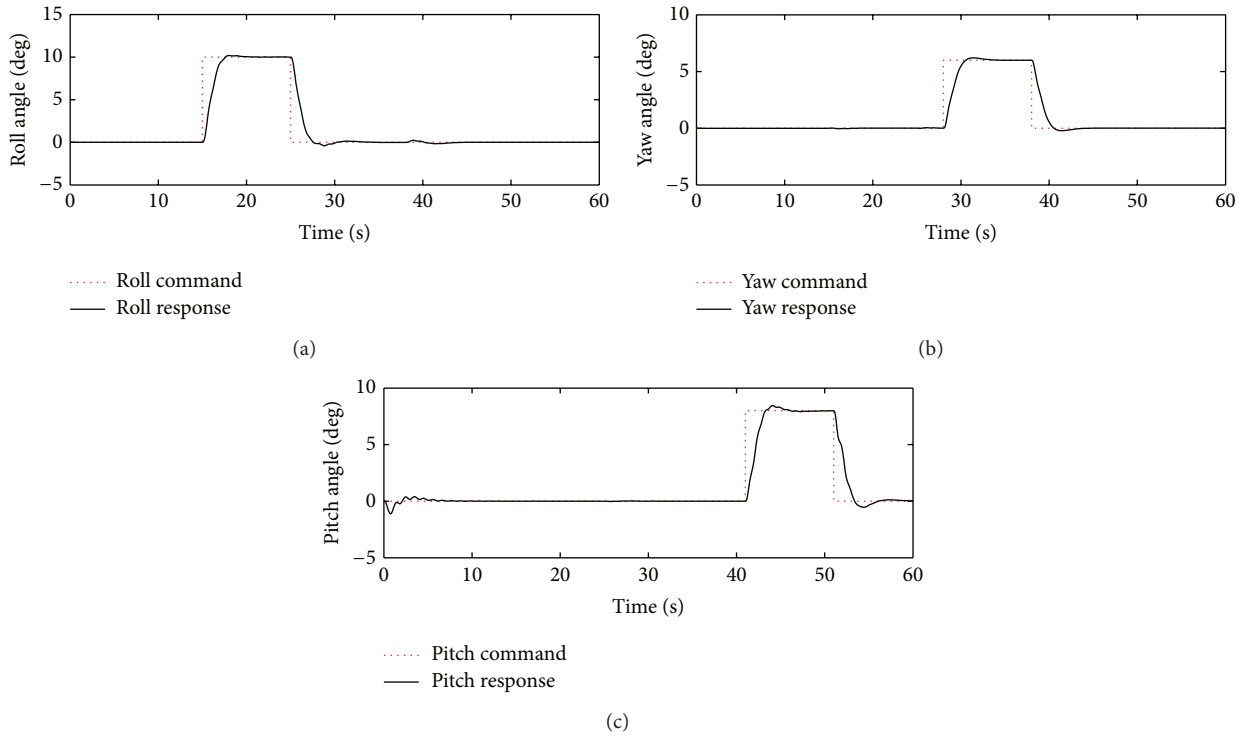
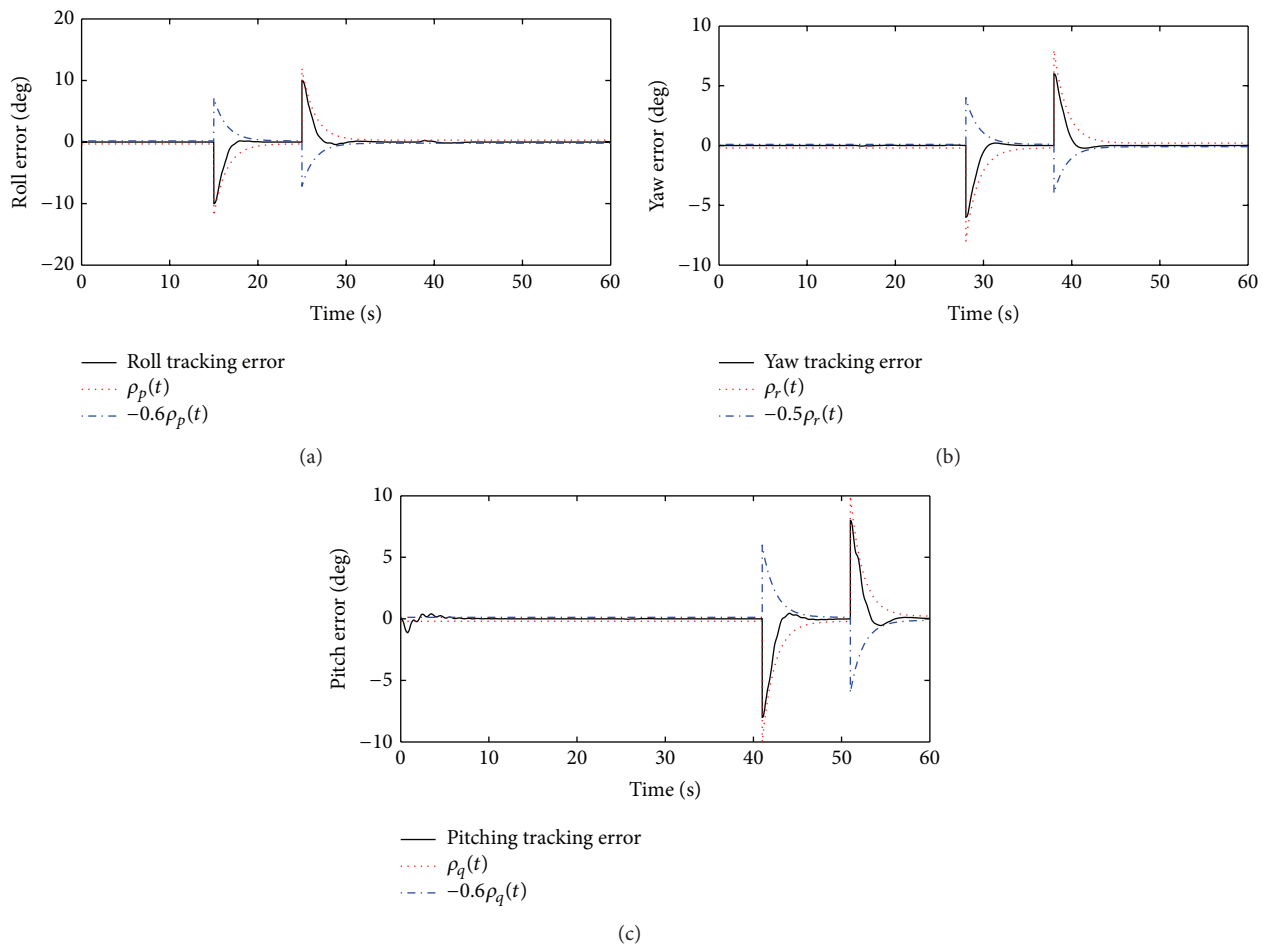FIGURE 6: Responses of the Attitude angles with model error.



FIGURE 7: Tracking errors of the attitude angles with model error.
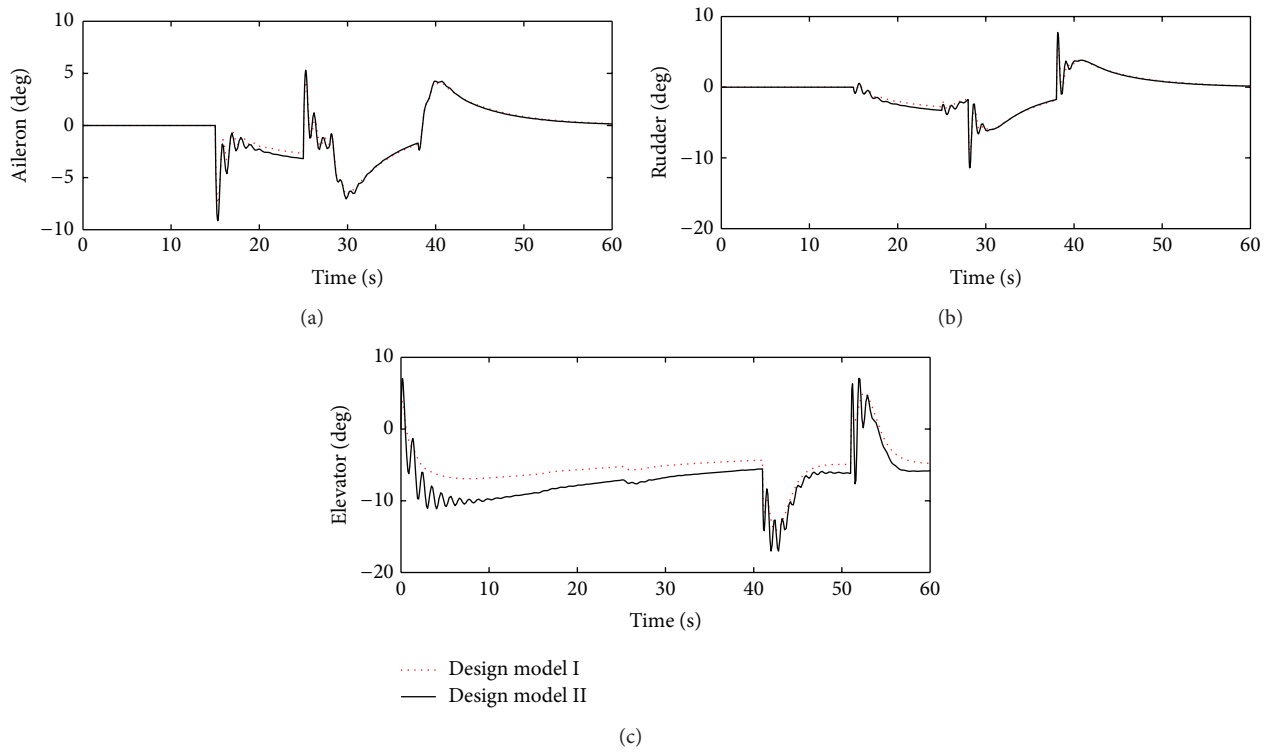
(a)



(b)



Design model I
Design model II

(c)

FIGURE 8: Deflections of the control actuators in two design models.



(a)


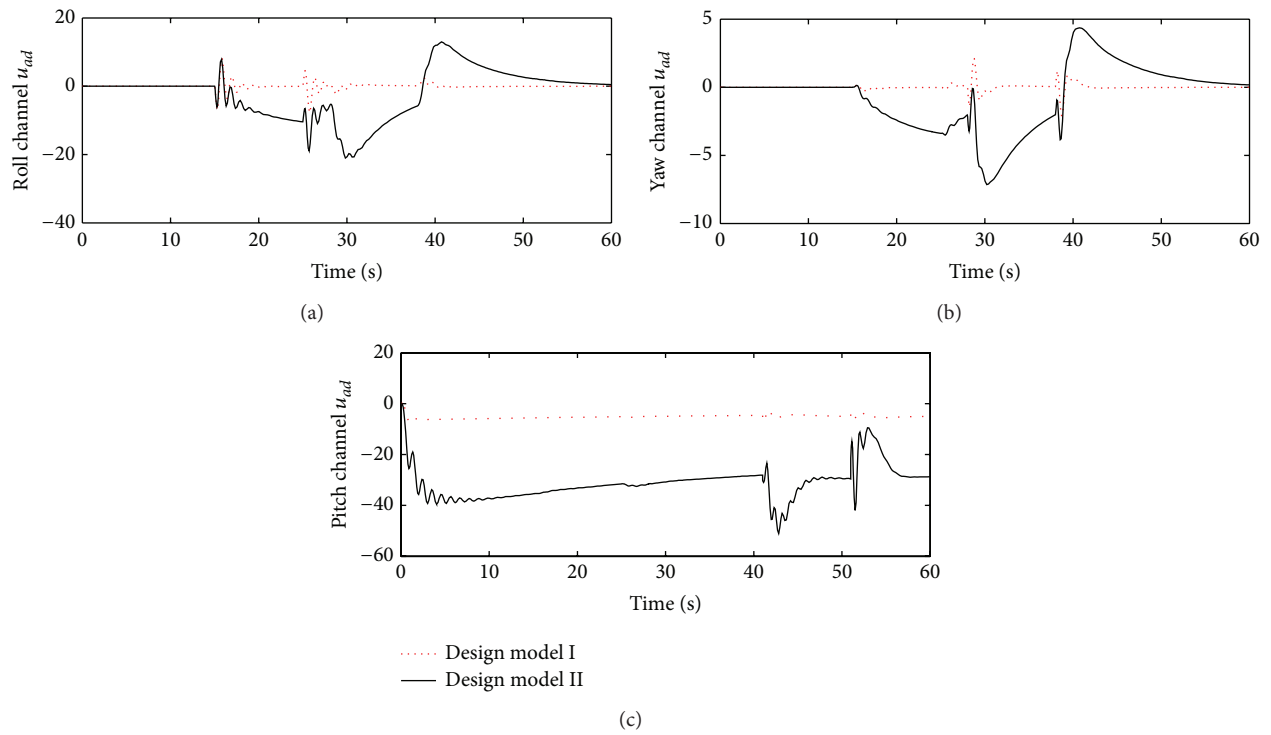
(b)



Design model I
Design model II

(c)

FIGURE 9: The outputs of neural network in three channels.

In the real flight control system, there must be the model error. In order to verify that the similar tracking performance is also achieved when there is the large model error, we have conducted the following simulation study.

The flight condition is the same, and the initial flight height and velocity are 6000 m and 190 m/s. However, the design model II used to design the attitude angles controllers is derived at the trimmed flight condition of 4000 m and 150 m/s. Apparently, the model error is large.

And the attitude angles tracking responses and tracking errors are shown in Figures 6 and 7.

Figures 6 and 7 show the attitude angles tracking errors still satisfy the prescribed performance bound, although the model error is large in this situation. In addition, Figures 6 and 7 show the track performance is similar when the different design models are used.

The control actuators deflections for three channels are compared in Figure 8 when the two design models are used.

Figure 8 shows the deflections of the control actuators using the design model II increase to derive the desired attitude angles tracking performance. In addition, the outputs of neural network in three channels are shown in Figure 9.

Figure 9 shows the outputs of neural network using the design model II are larger than the one using the design model I. The main reason is that the model error is larger when the design model II is used, and the larger outputs of neural network are used to compensate the large model error.

## 5. Conclusion

In this paper, an adaptive neural network dynamic inversion with prescribed performance method is proposed for aircraft attitude control. By incorporating the adaptive neural network dynamic inversion with the prescribed performance concept, the proposed method guarantees the system tracking error satisfies the prescribed performance bound in the transient and steady behavior. The nonlinear simulation of the aircraft also verifies the effectiveness of the proposed approach.

Further investigation is needed for the situations in the presence of the external wind disturbance and unmodeled dynamics. And, these design parameters in this method should be decreased and optimized to achieve a real application.

## Acknowledgments

## References

[1] S. A. Snell, D. F. Enns, and W. L. Garrard Jr., "Nonlinear inversion flight control for a supermaneuverable aircraft," *Journal of Guidance, Control, and Dynamics*, vol. 15, no. 4, pp. 976–984, 1992.

[2] D. Enns, D. Bugajski, R. Hendrick, and G. Stein, "Dynamic inversion: an evolving methodology for flight control design," *International Journal of Control*, vol. 59, no. 1, pp. 71–91, 1994.

[3] J. Leitner, A. Calise, and J. V. R. Prasad, "Analysis of adaptive neural networks for helicopter flight control," *Journal of Guidance, Control, and Dynamics*, vol. 20, no. 5, pp. 972–979, 1997.

[4] R. Rysdyk and A. J. Calise, "Robust nonlinear adaptive flight control for consistent handling qualities," *IEEE Transactions on Control Systems Technology*, vol. 13, no. 6, pp. 896–910, 2005.

[5] C. Cao and N. Hovakimyan, "Design and analysis of a novel $L_1$ adaptive control architecture with guaranteed transient performance," *IEEE Transactions on Automatic Control*, vol. 53, no. 2, pp. 586–591, 2008.

[6] W. Gai, H. Wang, and D. Li, "Trajectory tracking for automated aerial refueling based on adaptive dynamic inversion," *Journal of Beijing University of Aeronautics and Astronautics*, vol. 38, no. 5, pp. 585–590, 2012 (Chinese).

[7] H. Xu and P. A. Ioannou, "Robust adaptive control for a class of MIMO nonlinear systems with guaranteed error bounds," *IEEE Transactions on Automatic Control*, vol. 48, no. 5, pp. 728–742, 2003.

[8] V. Stepanyan and K. Krishnakumar, "Adaptive control with reference model modification," *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 4, pp. 1370–1374, 2012.

[9] C. P. Bechlioulis and G. A. Rovithakis, "Adaptive control with guaranteed transient and steady state tracking error bounds for strict feedback systems," *Automatica*, vol. 45, no. 2, pp. 532–538, 2009.

[10] C. P. Bechlioulis and G. A. Rovithakis, "Prescribed performance adaptive control for multi-input multi-output affine in the control nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 55, no. 5, pp. 1220–1226, 2010.

[11] A. Kostarigka and G. Rovithakis, "Adaptive dynamic output feedback neural network control of uncertain MIMO nonlinear systems with prescribed performance," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 1, pp. 138–149, 2012.

[12] C. P. Bechlioulis, Z. Doulgeri, and G. A. Rovithakis, "Guaranteeing prescribed performance and contact maintenance via an approximation free robot force/position controller," *Automatica*, vol. 48, no. 2, pp. 360–365, 2012.

[13] M. L. Fravolini, A. Ficola, G. Campa, M. R. Napolitano, and B. Seanor, "Modeling and control issues for autonomous aerial refueling for UAVs using a probe-drogue refueling system," *Aerospace Science and Technology*, vol. 8, no. 7, pp. 611–618, 2004.

[14] J. P. Nalepka and J. L. Hinchman, "Automated aerial refueling: extending the effectiveness of unmanned air vehicles," in *Proceedings of the AIAA Modeling and Simulation Technologies Conference and Exhibit*, pp. 240–247, August 2005, AIAA Paper 2005-6005.

[15] Y. Xili, F. Yong, and Z. Jihong, "Transition flight control of two vertical/short takeoff and landing aircraft," *Journal of Guidance, Control, and Dynamics*, vol. 31, no. 2, pp. 371–385, 2008.

[16] W. Gai and H. Wang, "Closed-loop dynamic control allocation for aircraft with multiple actuators," *Chinese Journal of Aeronautics*, vol. 26, no. 3, pp. 676–686, 2013.

[17] B. Stevens and L. Lewis, *Aircraft Control and Simulation*, Wiley, New York, NY, USA, 2nd edition, 2003.

[18] B. S. Kim and A. J. Calise, "Nonlinear flight control using neural networks," *Journal of Guidance, Control, and Dynamics*, vol. 20, no. 1, pp. 26–33, 1997.

[19] W. Wang and C. Wen, "Adaptive actuator failure compensation control of uncertain nonlinear systems with guaranteed transient performance," *Automatica*, vol. 46, no. 12, pp. 2082–2091, 2010.

[20] R. T. Rysdyk and A. J. Calise, "Adaptive model inversion flight control for tilt-rotor aircraft," *Journal of Guidance, Control, and Dynamics*, vol. 22, no. 3, pp. 402–407, 1999.