International Journal of Control and Automation
Vol. 9, No. 8 (2016), pp.69-78
http//dx.doi.org/10.14257/ijca.2016.9.8.08

Model-free Self-adaptive Control of The Nonlinear System Based on Feedback Linearization

Hong cheng Zhou * and Juan Yang**
** Institute of Information, JinLing Institute of Technology, Nanjing 211169, China*
*** Institute of Information, JinLing Institute of Technology, Nanjing 211169, China*
**zhouhc8@163.com **yangjuan@jit.edu.cn*

Abstract

In fact, dynamic models of course channels of small unmanned helicopters with different types and different parameters are different, and the establishment of course channel model is complex. It is necessary to research free model control methods without relying on the mathematical models. As one of the most important methods in model free control fields, intelligent control method has drawn extensive attention of scholars. But it is in the stage of theory research, and the actual application is not much. This paper tries a design of course channel controller of helicopters with the intelligent control method. Depending on the observer performance, the paper puts forward the model free self-adaptive control-filtration backstepping control method for the marine power system. The two developing model free self-adaptive backstepping control command filters mainly solve three problems.The simulation results show that this method not only ensures the stability of the closed-loop control system of the control marine electric power system, but also can determine the speed state and the unknown dynamic model.

***Keywords**: state observer; free model; intelligent control; backstepping; control filter*

1 Problem Posing

1.1. Input-Output Feedback Linearization

Consider the single-input and single-output affine system with the unknown model and the known order

$$\begin{aligned} \dot{\bar{x}} &= f_1(\bar{x}) + g_1(\bar{x})u \\ y &= h_1(\bar{x}) \end{aligned} \tag{1}$$

where f_1, g_1 and h_1 are sufficiently smooth in the domain $D \subset R^n$. Mapping $f_1 : D \to R^n$ and mapping $g_1 : D \to R^n$ are called as the vector fields on D. Derivative $\dot{y}$ is

$$\dot{y} = \frac{\partial h_1}{\partial \bar{x}}\left[f_1(\bar{x}) + g_1(\bar{x})u \right] \overset{def}{=} L_{f_1} h_1(\bar{x}) + L_{g_1} h_1(\bar{x})u \tag{2}$$

where $L_f h_1(\bar{x}) = \frac{\partial h_1}{\partial x} f_1(\bar{x})$ is referred to as the Lie derivative of h_1 on or along f_1. This representation is similar to the derivative of h_1 along the trajectory of system $\dot{\bar{x}} = f_1(\bar{x})$. In [1] I. Raptis *et al*. have investigated the linear and nonlinear control, when repeatedly calculate the derivative on the same vector field or a new vector field, the new representation is more convenient. For example, it is necessary to use the following expression:

ISSN: 2005-4297 IJCA

$$\begin{aligned}
&L_{g_1} L_{f_1} h_1(\bar{x}) = \frac{\partial (L_{f_1} h_1)}{\partial \bar{x}} g_1(\bar{x}) \\
&L_{f_1}^2 h_1(\bar{x}) = L_{f_1} L_{f_1} h_1(\bar{x}) = \frac{\partial (L_{f_1} h_1)}{\partial \bar{x}} f_1(\bar{x}) \\
&L_{f_1}^k h_1(\bar{x}) = L_{f_1} L_{f_1}^{k-1} h_1(\bar{x}) = \frac{\partial (L_{f_1}^{k-1} h_1)}{\partial \bar{x}} f_1(\bar{x}) \\
&L_{f_1}^0 h_1(\bar{x}) = h_1(\bar{x})
\end{aligned} \tag{3}$$

If $L_{g_1} h_1(\bar{x}) = 0$,then $\dot{y} = L_{f_1} h_1(\bar{x})$, which has nothing to do with u . If continue to calculate the second derivative of y , denoted as $y^{(2)}$, we can get

$$y^{(2)} = \frac{\partial (L_{f_1} h_1)}{\partial \bar{x}} \left[f_1(\bar{x}) + g_1(\bar{x}) u \right] = L_{f_1}^2 h_1(\bar{x}) + L_{g_1} L_{f_1} h_1(\bar{x}) u \tag{4}$$

By the same token, if $L_{f_1} L_{g_1} h_1(\bar{x}) = 0$, then $y^{(2)} = L_{f_1}^2 h_1(\bar{x})$, which has nothing to do with u . Through repeating this process, it can be seen that if $h_1(\bar{x})$ satisfies $L_{g_1} L_{f_1}^{i-1} h_1(\bar{x}) = 0$, $i = 1, 2, \cdots, \rho - 1$, $L_{g_1} L_{f_1}^{\rho} h_1(\bar{x}) \neq 0$, then u will not appear in the equation of $y, \dot{y}, \cdots, y^{(\rho-1)}$, but appear in the equation of $y^{(\rho)}$ with a nonzero coefficient, *i.e.*,

$$y^{(\rho)} = L_{f_1}^{\rho} h_1(\bar{x}) + L_{g_1} L_{f_1}^{\rho-1} h_1(\bar{x}) u \tag{5}$$

Define $x = [x_1, x_2, \cdots, x_\rho] = [y, y^2, \cdots, y^{\rho-1}]$, $f(\bar{x}) = L_{f_1}^{\rho} h_1(\bar{x})$, and $g(\bar{x}) = L_{g_1} L_{f_1}^{\rho-1} h_1(\bar{x})$ .Then the equation (5) can be denoted as the following state equations:

$$\begin{aligned}
&\dot{x}_1 = x_2 \\
&\dot{x}_i = x_{i+1} \\
&\cdots \\
&\dot{x}_\rho = f(\bar{x}) + g(\bar{x}) \cdot u(t) \\
&y = x_1
\end{aligned} \tag{6}$$

1.2. Higher-Order Neural Network Model

It has been proved in theory that even if there is only one hidden layer neural network, as long as the number of neurons in the layer is large enough, in the compact set it can uniformly asymptotically approximate any continuous nonlinear function. Therefore, in [2] X. Su *et al*. have investigated the novel control design on discrete-time, using the neural network for the identification and the modeling of the dynamic system has become an effective method and means.

Recurrent neural network is a dynamic network with feedback. Its remarkable characteristic is the neuron connection owning the feedback system, *i.e.*, the output of a layer is sent back to the same layer or the input of the previous layer through the connection weight. This is different from the feedforward neural network whose structure is hierarchical; its information is passed up in sequence; the cell of the first layer is connected with all cells of the second layer, and the cell of the second layer is also connected with all cells of its upper layer, according to this rule, until the output layer. While in the regression network, it always takes the previous output back to the input circularly, so its output depends not only on the current input, but also on the previous output. The network has the ability to reflect the dynamic characteristics through storing the internal state, which can reflect the dynamic characteristics of the system more directly and vividly. Thus, the system has the ability to adapt to the time-varying characteristics, representing the direction of the development of the neural network.

Below take a simple network structure as an example to introduce the higher-order regression neural network model.

Assume that each neuron state is described by the following differential equation:

$$\dot{\lambda}_i = a_i \lambda_i + \sum_j w_{ij} \eta_j \tag{7}$$

where λ_i is the *ith* neuron state; a_i is a constant; w_{ij} expresses the connection weight between the *jth* input and the *ith* neuron; η_j is the *jth* input of the above neuron; it can be the external input, and also can be through the S function, such as the neuron state under the function of $\eta_j = S(\lambda_j)$, where $S(\cdot)$ denotes the $S-type$ nonlinear function.

For the dynamic behaviour and the stability characteristic of the neural network model represented by equation (7), Hopfield and many scholars have conducted in-depth and meticulous researches. In [3] Z. Che *et al*. have investigated the feed-forward neural networks training, the research results show that the model has achieved good results on some application aspects, such as associative memory, *etc*. But at the same time, because of the simple structure it has exposed some limitations.

In the second-order regression neural network model, the total input of the neurons is not only the linear combination of η_j, but also the combination of pairwise products, such as $\eta_j \eta_k$. And extending in this way, the input may include the higher-order connections of input multiplication, such as three inputs ($\eta_j \eta_k \eta_i$), or four inputs, even more inputs. Thus form the recurrent high-order neural networks(RHONN).

Now take the recurrent high-order neural network formed by n neurons and m inputs as an example to illustrate. The state of the neuron is determined by the following differential equation:

$$\dot{\lambda}_i = -a_i \lambda_i + \sum_{k=1}^{L} w_{ik} \prod_{j \in I_k} \eta_j^{d_j(k)} \tag{8}$$

where λ_i is the *ith* neuron state; $\{I_1, I_2, \cdots, I_L\}$ is the disorderliness L subset in the set $\{1, 2, \cdots, m+n\}$; λ_i is a real coefficient; w_{ik} is the adjustable weight of neural network; $d_j(k)$ is a nonnegative integer; η is an input vector of the neuron, which is defined as follows:

$$\eta = [\eta_1, \cdots, \eta_n, \eta_{n+1}, \cdots, \eta_{n+m}]^T = [S(\lambda_1), \cdots, S(\lambda_n), S(u_1), \cdots, S(u_m)]^T \tag{9}$$

where $\upsilon = [u_1, u_2, \cdots, u_m]^T$ is the external input vector of neural network. $S(\cdot)$ is a monotone increasing differentiable S-type function, defined as:

$$S(\lambda) = \frac{\alpha}{1 + e^{-\beta\lambda}} + \varepsilon \tag{10}$$

where α, β are positive reals, and ε is a small real. For example, when $\alpha = \beta = 1, \varepsilon = 0$, formula (10) denotes a logistic function, and when $\alpha = \beta = 2, \varepsilon = -1$, it denotes a hyperbolic tangent function. The $S-type$ activation functions are very commonly used functions in neural network applications.

Here introduce L-dimensional vector z, which is defined as

$$z = [z_1, z_2, \cdots, z_L]^T = \left[\prod_{j \in I_1} y_j^{d_j(1)}, \cdots, \prod_{j \in I_L} y_j^{d_j(L)} \right]^T \tag{11}$$

So formula (8), the high-order regression neural network model, is transformed into:

$$\dot{\lambda}_i = -a_i \lambda_i + \sum_{k=1}^{L} w_{ik} z_k, i = 1, \cdots, n \quad (12)$$

Further, define the adjustable parametric vector $W_i = \left[w_{i,1}, \cdots, w_{i,L} \right]$, then formula (12) is transformed into

$$\dot{\lambda}_i = -a_i \lambda_i + W_i^T z, i = 1, \cdots, n \quad (13)$$

Here $\{W_i : i = 1, 2, \cdots, n\}$ is the adjustable weight of neural network, and the coefficients $\{a_i : i = 1, 2, \cdots, n\}$ denote the basic structure parameters of network, which are fixed during the network training. In order to guarantee that the input and the output of each neuron are bounded and stable, set a_i to a positive.

2. Dynamic Model Identification

For the convenience of the model identification, the formula (6) can be written as follows:

$$\begin{aligned} \dot{x} &= Ax + B\left(f(\bar{x}) + g(\bar{x}) \cdot u(t) \right) \\ y &= C^T x \end{aligned} \quad (14)$$

where

$$A_{\rho \times \rho} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ & & & \cdots & \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad B_{\rho \times 1} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}, \quad C_{\rho \times 1} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

For (14), based on RHONN mentioned in the above section, the observer is designed as follows

$$\begin{aligned} \dot{\hat{x}} &= A\hat{x} + B\left(W^{1T} z(\bar{x}) + W^{2T} z(\bar{x}) \cdot u(t) \right) + K(y - C^T \hat{x}) \\ \hat{y} &= C^T \hat{x} \end{aligned} \quad (15)$$

where $\hat{x} = \left[\hat{x}_1, \hat{x}_2, \cdots, \hat{x}_\rho \right]$ is the observed value of (14); $K = \left[k_1, k_2, \cdots, k_\rho \right]^T$ is the observer gain, and $W^{1,2T} = \left[W_1^{1,2}, \cdots, W_n^{1,2} \right]$. Define observation error $\tilde{x} = x - \hat{x}$ and output error $\tilde{y} = y - \hat{y}$. In [4] M. Zhang *et al*. have investigated the adaptive fault tolerant attitude control. From (15) and (14), the dynamic equation of the observation error can be obtained as follows

$$\dot{\tilde{x}} = \bar{A}\tilde{x} + B\left[\tilde{W}^{1T} z(\bar{x}) + \varepsilon_1 + \tilde{W}^{2T} z(\bar{x}) u + \varepsilon_2 u \right] \quad (16)$$

where $\bar{A} = A - KC^T$, $\tilde{W}^{1,2} = W_*^{1,2} - W^{1,2}$, and $W_*^{1,2}$ is the optimal weight matrix. $\varepsilon_{1,2}$ is the function estimation error of RHONN, and satisfies the bounded condition $\left| \varepsilon_{1,2} \right| \leq \bar{\varepsilon}_{1,2}$.

Theorem 1: for the RHONN observer designed by formula (14), in the case of the weight satisfying the self-adaptive adjustment rule as denoted in the following formula (17), it can guarantee that the observation error is uniformly ultimately bounded(UUB).

$$\begin{aligned} \dot{W}^1 &= \tilde{y}\Gamma_1 z(\bar{x}) - \kappa_1 \Gamma_1 W^1 \\ \dot{W}^2 &= \tilde{y}\Gamma_2 z(\bar{x}) u - \kappa_2 \Gamma_2 W^2 \end{aligned} \quad (17)$$

Prove: we have considered Lyapunov function.

$$V = \underbrace{\tilde{x}^T P \tilde{x}}_{V_1} + \underbrace{\tilde{W}^{1T} \Gamma^{-1} \tilde{W}^1 + \tilde{W}^{2T} \Gamma^{-1} \tilde{W}^2}_{V_2} \quad (18)$$

By taking the derivative of V_1, we can obtain

$$\dot{V}_1 = \tilde{x}^T(\bar{A}^T P + P\bar{A})\tilde{x} + 2\tilde{x}^T PB(\varepsilon_1 + \varepsilon_2 u) + 2\tilde{y}\left[\tilde{W}^{1T} z(\bar{x}) + \tilde{W}^{2T} z(\bar{x})u\right] \quad (19)$$

Because $\left|\varepsilon_{1,2}\right| \le \epsilon_{1,2}$, $|u| \le \max\left\{\left|u^{\min}\right|, \left|u^{\max}\right|\right\}$, we can get $B(\varepsilon_1 + \varepsilon_2 u) \le \Upsilon$, where

$$\Upsilon = \epsilon_1 + \epsilon_2 \cdot \max\left\{\left|u^{\min}\right|, \left|u^{\max}\right|\right\} \quad (20)$$

Taking advantage of Young inequation, we can obtain

$$2\tilde{x}^T PB(\varepsilon_1 + \varepsilon_2 u) \le \tilde{x}^T P^2 \tilde{x} + \Upsilon^2 \quad (21)$$

Consider the following generic algebraic Riccati inequation

$$\bar{A}^T P + P\bar{A} + P^2 \le -Q \quad (22)$$

where Q is a positive definite matrix. Substituting (21) into (19), we can get

$$\dot{V}_1 \le -\tilde{x}^T Q\tilde{x} + 2\tilde{y}\left[\tilde{W}^{1T} z(\bar{x}) + \tilde{W}^{2T} z(\bar{x})u\right] + \Upsilon^2 \quad (23)$$

Taking the weight adjustment rule into $\dot{V}_2$, it is obtained that $\dot{V}$ satisfies the relation below.

$$\begin{aligned}
\dot{V} &\le -\tilde{x}^T Q\tilde{x} + 2\kappa_1 \tilde{W}^{1T} W^1 + 2\kappa_2 \tilde{W}^{2T} W^2 + \Upsilon^2 \\
&\le -\lambda_{\min}(Q)\left\{\|\tilde{x}\|^2 + \frac{2\kappa_1}{\lambda_{\min}(Q)}\left(\left\|\tilde{W}^{1T}\right\| + \frac{1}{2}\left\|W_*^1\right\|\right)^2\right. \\
&\quad + \frac{2\kappa_2}{\lambda_{\min}(Q)}\left(\left\|\tilde{W}^{2T}\right\| + \frac{1}{2}\left\|W_*^2\right\|\right)^2 \\
&\quad \left. - \left[\frac{\kappa_1}{2\lambda_{\min}(Q)}\left\|W_*^1\right\|^2 + \frac{\kappa_2}{2\lambda_{\min}(Q)}\left\|W_*^2\right\|^2 + \frac{1}{\lambda_{\min}(Q)}\Upsilon^2\right]\right\}
\end{aligned} \quad (24)$$

So when the state estimation error

$$\|\tilde{x}\| > \sqrt{\frac{\kappa_1\|W_*^1\|^2}{2\lambda_{\min}(Q)} + \frac{\kappa_2\|W_*^2\|^2}{2\lambda_{\min}(Q)} + \frac{\Upsilon^2}{\lambda_{\min}(Q)}} = B_1$$

or the weight estimation error

$$\|\tilde{W}^{1T}\| > \sqrt{\frac{\|W_*^1\|^2}{4} + \frac{\kappa_2\|W_*^2\|^2}{4\kappa_1} + \frac{\Upsilon^2}{2\kappa_1}} - \frac{\|W_*^1\|}{2} = B_2$$

$$\|\tilde{W}^{2T}\| > \sqrt{\frac{\kappa_1\|W_*^1\|^2}{4\kappa_2} + \frac{\|W_*^2\|^2}{4} + \frac{\Upsilon^2}{2\kappa_2}} - \frac{\|W_*^1\|}{2} = B_3$$

In [5] B. Song *et al*. have investigated the nonlinear dynamic modeling and control, it is ensured that $\dot{V} < 0$. Through the above analysis, we can get uniform ultimate boundedness (UUB).

3. Controller Design

3.1. Output-Feedback Control

Define the reference trajectory $x^d = [x_1^d, x_2^d, \ldots, x_\rho^d]^T = [y^d, \dot{y}^d, \ldots, y^{(\rho)d}]^T$. Here the controller is designed as follows:

$$u = \frac{1}{W^{2T} z(\bar{x})}\left(-W^{1T} z(\bar{x}) + y^{(\rho)d} + \boldsymbol{K}^T \hat{e}\right) \quad (25)$$

where $\hat{e} = x^d - \hat{x}$. $\boldsymbol{K}$ is the feedback gain of the controller, which satisfies Hurwitz condition. In [6] D. Xu *et al*. have investigated the global robust tracking control of non-affine nonlinear systems, by substituting the controller formula (25) into formula (15), a closed-loop dynamic equation is obtained below.

$$\dot{\hat{e}} = A_c \hat{e} - KC^T \tilde{x} \tag{26}$$

where $A_c = A - B\boldsymbol{K}^T$. Solve the equation (26), and we can get

$$\hat{e}(t) = \mathrm{e}^{A_c t}\hat{e}(0) + \int_0^t \mathrm{e}^{A_c(t-\tau)} KC^T \tilde{x}(\tau)\mathrm{d}\tau \tag{27}$$

From the Theorem 1, we know that $\bar{\tilde{x}} = \sup_t |\tilde{x}| = B_1$. Calculate the absolute values of formulas on both ends of the equation (27), and we can get

$$\begin{aligned} |\hat{e}(t)| &\le \left|\mathrm{e}^{A_c t}\right| |e(0)| + \int_0^t \left|\mathrm{e}^{A_c(t-\tau)}\right| \mathrm{d}\tau \left|KC^T\right| \bar{\tilde{x}} \\ &\le m\mathrm{e}^{-\alpha t}|\hat{e}(0)| + \frac{m\left|KC^T\right|}{\alpha} B_1 \end{aligned} \tag{28}$$

where m and α are positive definite constants for satisfying inequation $\left|\mathrm{e}^{A_c(t-\tau)}\right| \le m\mathrm{e}^{-\alpha(t-\tau)}$.

From the above analysis, we can see that the closed-loop control system can not guarantee that the output approximates the tracking reference trajectory. So it is necessary to add a controller to guarantee the asymptotic stability of the closed-loop system. The design of the supervision controller provides a necessary way for it. Therefore add the supervision controller to the above-mentioned controller to meet the need of asymptotic stability.

3.2. The Design of Supervision Controller

In the system (6), assume that the upper bound of $f(\bar{\boldsymbol{x}})$ and the lower bound of $g(\bar{\boldsymbol{x}})$ are known, *i.e.*, there are confirmable functions $f^U(\bar{\boldsymbol{x}})$ and $g_L(\bar{\boldsymbol{x}})$,so that $|f(\bar{\boldsymbol{x}})| < f^U(\bar{\boldsymbol{x}})$, $0 < g_L(\bar{\boldsymbol{x}}) \le g(\bar{\boldsymbol{x}})$.

The main controller design given by the former is

$$u = u_{nn}(\bar{x}) = \frac{1}{W^{2T} z(\bar{x})}\left(-W^{1T} z(\bar{x}) + y^{(\rho)d} + \boldsymbol{K}^T \hat{e}\right) \tag{29}$$

In [7] H. Kim *et al*. have investigated the parameter identification and design of a robust attitude controller, in order to ensure the stability of the closed-loop system, it is necessary to design a controller., and the closed-loop system with the controller is required to be global stable. Add the supervision controller $u_s(\bar{x})$ on the controller $u_{nn}(\bar{x})$. $u_s(\bar{x})$ is not zero only if the output tracking error $e = x^d - x$ reaches the boundary of the constraint set $\{e : |e| \le M_e\}$, where M_e is a real bigger than zero given by the designer.

Hence, the global controller with the supervision control can be redesigned as

$$u = u_{nn}(\bar{x}) + I^* u_s(\bar{x}) \tag{30}$$

where I^* is an indicator function. The main task of the control is still undertaken by $u_{nn}(\bar{x})$. In [8] X. Zhao *et al*. have investigated the adaptive guaranteed cost control approach, through designing supervision controller u_s , for all $t > 0$, there is $\|e\| \le M_e$. The control strategy of the supervision controller in formula (30) is that when $\|e\| \ge M_e$, $I^* = 1$ and when $\|e\| \le M_e$, $I^* = 0$.

Put formula (30) into formula (6), and it is obtained that

$$x_1^{(\rho)} = f(\bar{x}) + g(\bar{x})(u_{nn}(\bar{x}) + I^* u_s(\bar{x})) \tag{31}$$

In order to prove the stability, the equation of the closed-loop system needs to be written as a vector form.

$$u^* = \frac{1}{g(\bar{x})}\left(-f(\bar{x}) + y^{(\rho)d} + \boldsymbol{K}^T e\right)$$

Substitute the ideal controller into the formula (31), and we can get

$$e^{(n)} = -K^T e + g(\bar{x})\left[u_{nn}(\bar{x}) - u^* + I^* u_s\right] \tag{32}$$

Define $\boldsymbol{b}_g = \begin{bmatrix} 0 & 0 & \cdots & 0 & g(\bar{x}) \end{bmatrix}^T$, and formula (3.43) can be written as a vector form.

$$\dot{e} = \Lambda e + \boldsymbol{b}_g\left[u_{nn}(\bar{x}) - u^* + I^* u_s\right] \tag{33}$$

Stability Analysis:

Lyapunov function is defined as

$$V_3 = \frac{1}{2}e^T P_1 e \tag{34}$$

where P_1 is the positive definite symmetric matrix which satisfies Lyapunov function, and it also satisfies

$$\Lambda^T P_1 + P_1 \Lambda = -Q_1 \tag{35}$$

where $Q_1 > 0$ is given by the designer. Because Λ is stable, such P_1 always exists. Considering that when $\|e\| \ge M_e$, $I^* = 1$, it is obtained that

$$\begin{aligned}
\dot{V}_3 &= \frac{1}{2}\dot{e}^T P_1 e + \frac{1}{2}e^T P_1 \dot{e} \\
&= \frac{1}{2}\left(\Lambda e + \boldsymbol{b}_g\left[u_{nn}(\bar{x}) - u^* + I^* u_s\right]\right)^T P_1 e + \frac{1}{2}e^T P\left(\Lambda e + \boldsymbol{b}_g\left[u_{nn}(\bar{x}) - u^* + I^* u_s\right]\right) \\
&= \frac{1}{2}e^T(\Lambda^T P_1 + P_1\Lambda)e + \frac{1}{2}\boldsymbol{b}_g{}^T P_1 e\left[u_{nn}(\bar{x}) - u^* + u_s\right] + \frac{1}{2}e^T P_1 \boldsymbol{b}_g\left[u_{nn}(\bar{x}) - u^* + u_s\right] \\
&= -\frac{1}{2}e^T Q_1 e + e^T P_1 \boldsymbol{b}_g\left[u_{nn}(\bar{x}) - u^* + u_s\right] \\
&\le \left|e^T P_1 \boldsymbol{b}_g\right|(\left|u_{nn}(\bar{x})\right| + \left|u^*\right|) + e^T P_1 \boldsymbol{b}_g u_s
\end{aligned} \tag{36}$$

In order to ensure $\dot{V}_3 \le 0$, the supervision control item u_s can be designed as

$$u_s = -\operatorname{sgn}(e^T P_1 \boldsymbol{b}_g)\left[\frac{1}{g_L}\left(f^U + \left|\boldsymbol{K}^T e\right|\right) + \left|u_{nn}\right|\right] \tag{37}$$

Substitute formula (37) into formula (36), and we can get

$$\dot{V}_3 \le 0$$

In formula (30), I^* is a step function. When e touches the boundary $\|e\| = M_e$, the supervision controller begins to work. When e comes back to the inside of constraint condition $\|e\| = M_e$, the supervision controller stops working. Therefore, in [9] H.Wang *et al*. have investigated the robust multi-mode flight control design, the system comes under attack when across the border. One way to overcome this "oscillation" is making I^* continuously change between 0 and 1. We can choose as follows:

$$I^* = \begin{cases} 0, & \|e\| < a \\ \dfrac{\|e\| - a}{M_e - a}, & a \le \|e\| < M_e \\ 1, & \|e\| \ge M_e \end{cases} \tag{38}$$

where $a \in (0, M_e)$ is a constant given by the designer. Then in formula (37), when e changes to M_x from a, supervision controller u_s will continuously vary from halted state to “maximum” 1. It can be seen from the above analysis that through designing the supervision controller u_s, we can try to make $\|e\| \le M_e$.

4. Simulation and Verification

Considering the single mechanical arm, its dynamic equations are as follows

$$\dot{x} = f(x) + g(x)u$$
$$y = h(x)$$

where $f(x) = \begin{bmatrix} x_2 \\ -\dfrac{mgl\sin x_1}{2M} \end{bmatrix}$, $g(x) = \begin{bmatrix} 0 \\ \dfrac{1}{M} \end{bmatrix}$, and $h(x) = x_1$. The physical parameters of the model are taken as follows: $m = 1$, $l = 1$, $g = 9.8$ and $M = 0.5$ .In the simulation, $K = [300, 600]^T$, $\Gamma_1 = diag[10^4]$, $\Gamma_2 = diag[10^3]$, $\kappa_1 = \kappa_2 = 0.01$ .In the supervision controller, $f^U = 10$, $g_L = 2$. The output setting trajectory is $y^d = \sin t$.

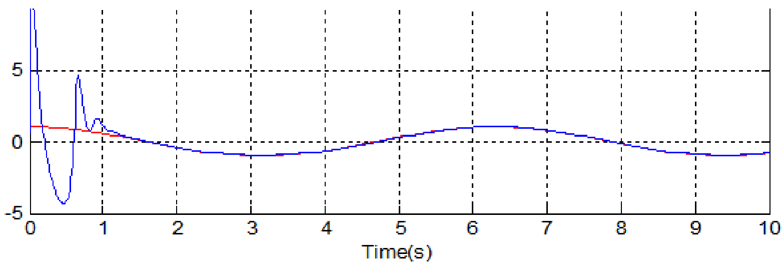
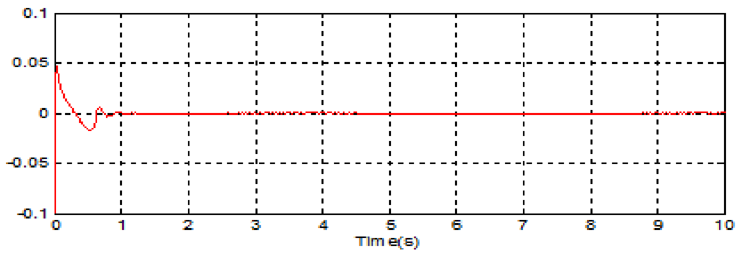
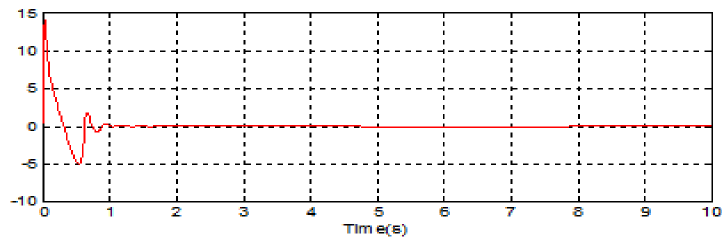


Figure 1. The Rate Tracking



(a) the observation error of x_1



(b) the observation error of x_2

Figure 2. Error Results of State Observation

The simulation is divided into two parts. The first part is the response curve of the closed-loop system without applying the supervision controller. The second part is the effect after applying the supervision controller.

(Simulation 1): Not apply the supervision controller

When there is no external disturbance, the simulation results are shown in Figure 1 and Figure 2. Through the simulations, it can be seen that choosing the appropriate parameters and neural network, not applying the supervision controller can also ensure that the system has good tracking performance.

When there is the external disturbance $d(t) = 4\sin(10t)$, the simulation results are shown in Figure 3 and Figure 4. It can be seen from the simulation results that when there is a larger interference, the tracking performance of the system declines.

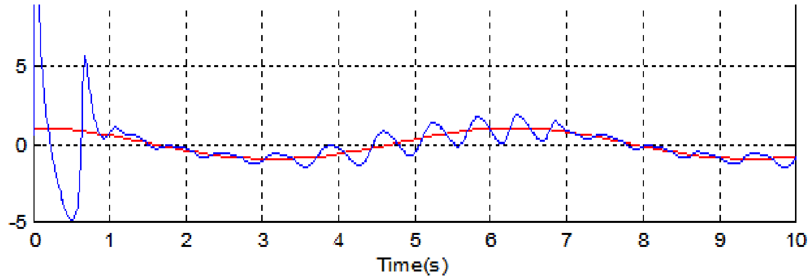
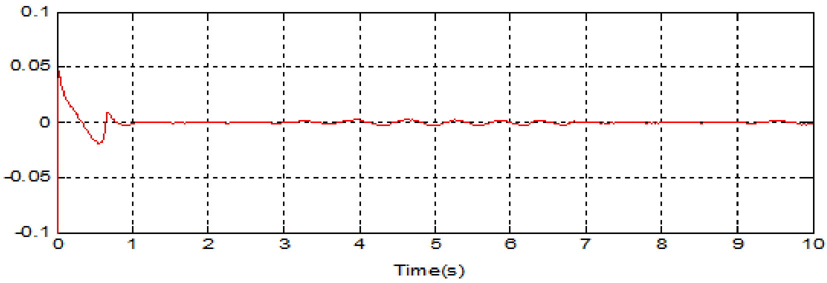
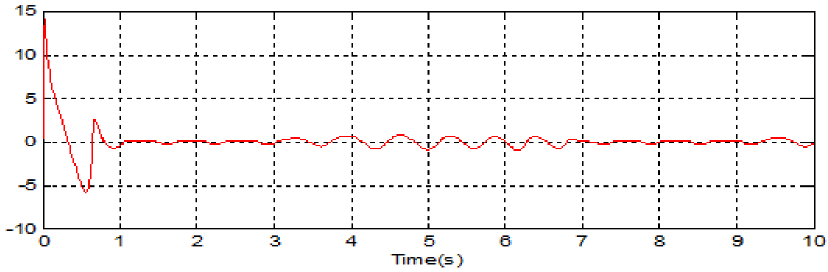


Figure 3. The Rate Tracking



(a) the observation error of x_1



(b) the observation error of x_2

Figure 4. Error Results of State Observation

(Simulation 2): apply the supervision controller

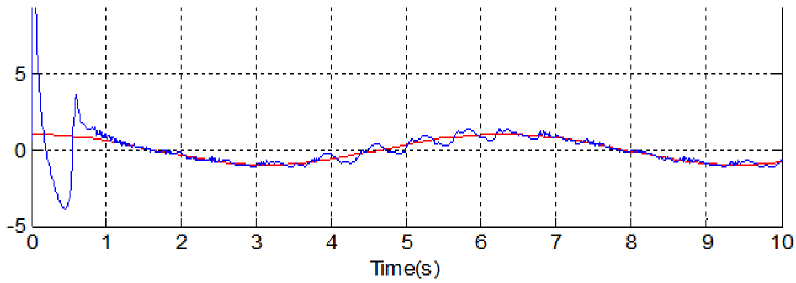


Figure 5. The Rate Tracking

Still consider that there is the external disturbance $d(t) = 4\sin(10t)$, the simulation result of applying the supervision controller is shown in Figure 5. It can be seen from the simulation that after applying the supervision controller, the output tracking performance of the system does not decline obviously. Thus it can be seen that, the introduction of the supervision controller will not only greatly increase the robustness of the system, but also increase the anti-jamming.

5. Conclusion

For the small UAV course system existing disturbances, the paper has considered the flight control problem, and designed the indirect fuzzy self-adaptive supervised course controller. Firstly, for the helicopter course system, take the model transformation, *i.e.*, transform the system into a second-order nonlinear time-varying system. Secondly, for the system design an indirect fuzzy self-adaptive course controller. Since the designed fuzzy self-adaptive course controller depends on the selection of the fuzzy rule base, in order to reduce the dependence, continue to design a supervision controller on the second layer to make the stability of the closed-loop system guaranteed. The combination of the two methods can make the controller design have a great degree of freedom and flexibility, so that it can greatly reduce the dependency of control on whether the selection of the rule base is correct or not. Finally, the designed method is applied to the control of the small-scale helicopter course system, that realize the tracking control of helicopter course channels, and achieve the good control performance and effectiveness.

Acknowledgment

This work was supported by Natural Science Research Project (No.61302167), Natural Science Foundation of Jiangsu Province No.15KJB510012.

References

[1] I. Raptis and K. Valavanis, “Linear and nonlinear control of small-scale unmanned helicopters”, Springer Verlag, Berlin, **(2011)**.
[2] X. Su, P. Shi, L. Wu and Y. Song, “A novel control design on discrete-time Takagi-Sugeno fuzzy systems with time-varying delays”, IEEE Trans on Fuzzy Systems, vol. 21(, no. 4, **(2013)**, pp. 655-671.
[3] Z. Che, T. Chiang and Z. Che, “Feed-forward neural networks training: a comparision between genetic algorithm and back-forward learning algorithm”, International Journal of Innovative Computing, Information and Control, vol. 7, no. 10, **(2011)**, pp. 5839-5850.
[4] M. Zhang, L. Yin and L. Qiao, “Adaptive fault tolerant attitude control for cube satellite in low earth orbit based on dynamic neural network”, International Journal of Innovative Computing, Information and Control, vol. 10, no. 5, **(2014)**, pp. 1843-1852.
[5] B. Song, J. Mills, Y. Liu and C. Fan, “Nonlinear dynamic modeling and control of a small-scale helicopter”, International Journal of Control, Automation, and Systems, vol. 8, no. 3, **(2010)**, pp. 534-543.
[6] D. Xu, B. Jiang and P. Shi, “Global robust tracking control of non-affine nonlinear systems with application to yaw control of UAV helicopter”, International Journal of Control, Automation and Systems, vol. 11, no. 5, **(2013)**, pp. 957-965.
[7] H. Kim, H. Dharmayanda, T. Kang, A. Budiyono, G. Lee and W. Adiprawita, “Parameter identification and design of a robust attitude controller using H1 methodology for the raptor E620 small-scale helicopter”, International Journal of Control, Automation, and Systems, vol. 10, no. 1, **(2012)**, pp. 88-101.
[8] X. Zhao and J. Han, “Yaw control of helicopter: an adaptive guaranteed cost control approach”, International Journal of Innovative Computing, Information and Control, vol. 5, no. 8, **(2009)**, pp. 2267-2276.
[9] H. Wang, A. Mian, D.Wang and H. Duan, “Robust multi-mode flight control design for an unmanned helicopter based on multi-loop structure”, International Journal of Control, Automation, and Systems, vol. 7, no. 5, **(2009)**, pp. 723-730.