Journal of Uncertain Systems
Vol.5, No.3, pp.163-174, 2011
Online at: www.jus.org.uk

B-spline Functional Link Network Adaptive Control for Hypersonic Vehicles Using Swarm Intelligent Optimizer

Yanli Du*, Qingxian Wu, Changsheng Jiang, Lu Cheng
College of Automation Engineering, Nanjing University of Aeronautics & Astronautics
Nanjing 210016, Jiangsu, China

Received 17 July 2009; Revised 25 January 2010

Abstract

This paper proposes a new recurrent neural network (NN) adaptive control method for air-breathing hypersonic vehicles (AHVs) in the presence of dynamical parameter uncertainties and disturbances. The control law mainly consists of the optimal generalized predictive control (OGPC) algorithm, and an adaptive adjustment of B-spline recurrent functional link network (BRFLN). The BRFLN, a new NN presented here, is applied to approximating dynamical uncertainties and disturbances. Moreover, a guaranteed global convergence algorithm, the stochastic particle swarm optimizer (SPSO) is employed to learn the self-connection weights of the BRFLN, which can improve the adaptability of the BRFLN. It is proved that the closed-loop system error is uniformly ultimately bounded. Finally, simulation results show that the controller designed attains a satisfactory performance for the AHV attitude tracking.

Keywords: predictive control, functional link network, B-spline, stochastic PSO, hypersonic vehicle

1 Introduction

Air-breathing hypersonic vehicles (AHVs) offer a promising and cost-effective technology for access to space and prompt global strike capabilities. Their guidance and control law design has become one of the greatest aeronautical research challenges. Generally, AHVs fly in the airspace from 20km to 100km above the earth. The resistance and energy consumption upon them are relatively low among this airspace, whereas there exist rarefied air, complicated temperature variation and disturbances such as gust, wind shear and possible turbulence. Furthermore, some key factors add plant parameter dynamical uncertainties to the AHVs' control problems, e.g. the rapid change of mass distribution, the motor thrust misalignment and the lack of a broad flight dynamic database.

To improve the control performance of AHVs under strong uncertainties and disturbances, some researchers proposed different robust or adaptive control methods. In [20], an adaptive sliding controller was designed for the longitudinal dynamics of an AHV and it possessed robustness with respect to parameter uncertainties. Fiorentini [4] adopted a robust nonlinear sequential loop-closure approach to design dynamic state-feedback controller under uncertainties. Kuipers [11] utilized the robust adaptive multiple model controller for the AHV trajectory control. However, uncertain parameter variations in these literatures were restricted within a given range, and hence these methods are inadequate to deal with the wide range of flight conditions due to the large dynamical parameter uncertainties. Moreover, these literatures do not consider the AHV attitude control system which is practically more sensitive to uncertainties and disturbances than the trajectory control system. Therefore, this paper will discuss the design of the AHV attitude control system in presence of dynamical disturbances and parameter uncertainties.

In recent years, intelligent techniques using fuzzy logic (FL) or neural networks (NNs) have been combined with nonlinear control methods and have revealed a nice effect on flight control problems [16, 18]. Especially, NN adaptive control is used because of their adaptability and robustness to unmodeled system and their hardware implementation capability. Nevertheless, FL or feed-forward NN belongs to the static mapping method and may be a good analytic tool to the relation existing in long-term. To the approximation of temporal or short-term dynamic variation, recurrent/dynamic neural network (RNN) possesses a superior capability as compared to feed-forward NN [6, 12]. If designed properly, RNN will attain a better learning result in smaller dimensions and less computational

* Corresponding author. Email: duylnuaa@yahoo.cn (Y. Du).

cost. In fact, the uncertainties/ disturbances during hypersonic flight are dynamically changeable functions and their variation may be random fluctuation, but few literatures in flight control field have considered large dynamical uncertainties. Hence a new simple RNN, B-spline recurrent functional-link network (BRFLN), is proposed and combined with optimal generalized predictive control (OGPC) method [1] to improve the attitude tracking performance of AHVs. The feedforward and feedback weights of BRFLN are online tuned by the derived adaptive law based on Lyapunov stability theorem. The self-connection weights of BRFLN are online learned by the stochastic PSO (SPSO) for simulating high-order nonlinear functions and strengthening the adaptability of the network. Moreover, we also design a robust controller to achieve desired control precision.

The paper is organized as follows: in Section 2, the OGPC algorithm with uncertainty is derived. Section 3 discusses the design procedure of BRFLN-based OGPC and the learning of BRFLN self-connection weights with SPSO. In Section 4, we analyse the stability of the close-loop control system. Finally, the presented method is applied to the attitude tracking control of the AHV and simulation results are given followed by the conclusion.

2 The OGPC with Uncertainty

The OGPC is a closed-form nonlinear predictive control law proposed by Chen et al. This method overcomes the main shortcoming of traditional predictive control algorithms. That is, the OGPC avoids the online computation, and hence it is an effective algorithm for fast time-varying systems [1, 8]. However, uncertainties/disturbances have not been considered in above literatures. So the OGPC with uncertainty will be derived in the following study. Let us consider a nonlinear system under uncertainties/disturbances described by

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{f}(\boldsymbol{x}(t)) + \boldsymbol{g}_1(\boldsymbol{x}(t))\boldsymbol{u}(t) + \boldsymbol{g}_2(\boldsymbol{x}(t))\boldsymbol{D} \\ \boldsymbol{y}(t) &= \boldsymbol{h}(\boldsymbol{x}(t)) \end{aligned} \tag{1}$$

where $\boldsymbol{x} \in \boldsymbol{R}^n, \boldsymbol{u} \in \boldsymbol{R}^m$ and $\boldsymbol{y} \in \boldsymbol{R}^m$ are the state vector, input and output, respectively. $\boldsymbol{g_2 D} \in \boldsymbol{R}^n$ denotes the lumped disturbances which are dynamically changeable and they include unmodeled dynamics, parameter variation and external disturbances etc. It is assumed that $\boldsymbol{f}(\boldsymbol{x}) \in \boldsymbol{R}^n$, $\boldsymbol{g}_1(\boldsymbol{x}) \in \boldsymbol{R}^{n\times m}$ and $\boldsymbol{g}_2(\boldsymbol{x}) \in \boldsymbol{R}^{n\times m}$ are smooth functions in terms of x and the following assumptions [1] are imposed on the system (1):
1) The output and reference signal are many times continuously differentiable with respect to *t*; 2) All states are available; 3) The system input and lumped disturbances have the relative degree ρ, and the zero dynamics are stable.

An optimal tracking problem is specified as: a controller is designed such that the close-loop system is asymptotically stable and the output of system (1) optimally tracks a prescribed reference output, in terms of a given performance index as

$$J = \frac{1}{2}\int_0^T \boldsymbol{e}^{\mathrm{T}}(t+\tau)\boldsymbol{e}(t+\tau)d\tau \tag{2}$$

where *T* is the predictive period, *e* is the output tracking error, defined by $\boldsymbol{e}(t+\tau) = \boldsymbol{y}(t+\tau) - \boldsymbol{y}_r(t+\tau)$ with $\boldsymbol{y}(t+\tau)$ the predictive output during *T* and $\boldsymbol{y}_r(t+\tau)$ the prescribed reference output. The control weighting term is not included in (2) following [1, 5].

Repeated differentiating up to ρ+r times of the output y with respect to *t*, we obtain

$$\begin{aligned} &\dot{\boldsymbol{y}}(t) = L_f \boldsymbol{h}(\boldsymbol{x}(t)) \\ &\quad\vdots \\ &\boldsymbol{y}^{[\rho-1]}(t) = L_f^{\rho-1}\boldsymbol{h}(\boldsymbol{x}(t)) \\ &\boldsymbol{y}^{[\rho]}(t) = L_f^{\rho}\boldsymbol{h}(\boldsymbol{x}(t)) + L_{g_1}L_f^{\rho-1}\boldsymbol{h}(\boldsymbol{x}(t))\boldsymbol{u}(t) + L_{g_2}L_f^{\rho-1}\boldsymbol{h}(\boldsymbol{x}(t))\boldsymbol{D} \\ &\quad\vdots \\ &\boldsymbol{y}^{[\rho+r]}(t) = L_f^{\rho+r}\boldsymbol{h}(\boldsymbol{x}(t)) + \boldsymbol{p}_{r1}(\boldsymbol{u}(t),\boldsymbol{x}(t),\boldsymbol{D}) + \boldsymbol{p}_{r2}(\dot{\boldsymbol{u}}(t),\boldsymbol{u}(t),\boldsymbol{x}(t),\boldsymbol{D}) + \cdots + \\ &\qquad \boldsymbol{p}_{rr}(\boldsymbol{u}^{[r-1]}(t),\cdots,\boldsymbol{u}(t),\boldsymbol{x}(t),\boldsymbol{D}) + L_{g_1}L_f^{\rho-1}\boldsymbol{h}(\boldsymbol{x})\boldsymbol{u}^{[r]}(t) \end{aligned} \tag{3}$$

where L_f denotes the Lie derivative, ρ is the system relative degree, *r* is the control order of $\boldsymbol{u}$, and $\boldsymbol{p}_{11}, \boldsymbol{p}_{21}, \boldsymbol{p}_{22}, \cdots, \boldsymbol{p}_{r1}, \cdots, \boldsymbol{p}_{rr}$ are complex functions in terms of $\boldsymbol{u}$, $\dot{\boldsymbol{u}}$, $\cdots$, $\boldsymbol{u}^{[r-1]}$ and $\boldsymbol{x}$. $\boldsymbol{y}(t+\tau)$ and $\boldsymbol{y}_r(t+\tau)$ at the time τ are approximately predicted by their Taylor-series expansions up to (ρ+*r*)th order. That is

$$\boldsymbol{y}(t+\tau) \cong \boldsymbol{\Gamma}(\tau)\bar{\boldsymbol{Y}}(t), \qquad \boldsymbol{y}_r(t+\tau) \cong \boldsymbol{\Gamma}(\tau)\bar{\boldsymbol{Y}}_r(t) \tag{4}$$

where $\bar{\boldsymbol{Y}}(t)=[\boldsymbol{y}(t)^{\mathrm{T}},\dot{\boldsymbol{y}}(t)^{\mathrm{T}},\cdots,\boldsymbol{y}^{[\rho+r]}(t)^{\mathrm{T}}]^{\mathrm{T}}$, $\bar{\boldsymbol{Y}}_r(t)=[\boldsymbol{y}_r(t)^{\mathrm{T}},\dot{\boldsymbol{y}}_r(t)^{\mathrm{T}},\cdots,\boldsymbol{y}_r^{[\rho+r]}(t)^{\mathrm{T}}]^{\mathrm{T}}$ and $\boldsymbol{\Gamma}(\tau)=[1 \quad \bar{\tau} \quad \cdots \quad \bar{\tau}^{\rho+r}/(\rho+r)!]$ with $\bar{\tau}=diag(\tau,\cdots,\tau)\in\boldsymbol{R}^{m\times m}$. Thus, the predictive error in (2) can be written as

$$\boldsymbol{e}(t+\tau)=\boldsymbol{\Gamma}(\tau)(\bar{\boldsymbol{Y}}(t)-\bar{\boldsymbol{Y}}_r(t)). \tag{5}$$

Letting $\left.\dfrac{\partial J}{\partial \boldsymbol{u}}\right|_{u=u^*}=0$ and consulting the proof of Theorem 1 in [1], we can prove that the optimal nonlinear control law which minimizes the receding horizon performance index (2) is given by

$$\boldsymbol{u}(t)=-(\boldsymbol{G}_1(\boldsymbol{x}))^{-1}(\boldsymbol{F}(\boldsymbol{x})+\boldsymbol{K}\boldsymbol{M}_\rho-\boldsymbol{y}_r^{[\rho]}(t)+\boldsymbol{H}(\boldsymbol{x})\boldsymbol{D}) \;=\bar{\boldsymbol{u}}-(\boldsymbol{G}_1(\boldsymbol{x}))^{-1}\boldsymbol{H}(\boldsymbol{x})\boldsymbol{D} \tag{6}$$

where $\boldsymbol{F}(\boldsymbol{x})=L_f^\rho\boldsymbol{h}(\boldsymbol{x})$, $\boldsymbol{G}_1(\boldsymbol{x})=L_{g_1}L_f^{\rho-1}\boldsymbol{h}(\boldsymbol{x})$, $\boldsymbol{H}(\boldsymbol{x})=L_{g_2}L_f^{\rho-1}\boldsymbol{h}(\boldsymbol{x})$ (The definitions of $\boldsymbol{K}$ and $\boldsymbol{M}_\rho$ come from [1]) and $\bar{\boldsymbol{u}}$ represents the control law of a nominal system. Accordingly, the robust OGPC law may be designed as

$$\boldsymbol{u}=\bar{\boldsymbol{u}}-\boldsymbol{u}_{ad}-\boldsymbol{u}_r,\quad \boldsymbol{u}_{ad}=\boldsymbol{G}_0(\boldsymbol{x})\boldsymbol{v}_{ad},\quad \boldsymbol{u}_r=\boldsymbol{G}_0(\boldsymbol{x})\boldsymbol{v}_{ad},\quad \boldsymbol{G}_0(\boldsymbol{x})=\boldsymbol{G}_1^{-1}(\boldsymbol{x})\boldsymbol{H}(\boldsymbol{x}) \tag{7}$$

where $\boldsymbol{u}_{ad}$ is the control adjustment in the presence of uncertainties/disturbances and $\boldsymbol{v}_{ad}$ is the estimation of $\boldsymbol{D}$. $\boldsymbol{u}_r$ denotes a robust control (RC) item which is used for offsetting the error between $\boldsymbol{D}$ and $\boldsymbol{v}_{ad}$, i.e., $\boldsymbol{v}_r=\boldsymbol{D}-\boldsymbol{v}_{ad}$. Obviously, the control law $\boldsymbol{u}$ can be employed to counteract negative effects that lumped disturbances induce. Because $\boldsymbol{D}$ is a dynamically changeable function, dynamic mapping methods can be fit for the approximation of $\boldsymbol{D}$ better than static ones. Thus, a new RNN, the BRFLN is presented to approach $\boldsymbol{D}$ for reducing the computation load.

3 The Design of BRFLN-based OGPC

3.1 The Design of B-spline RFLN

Functional link network (FLN) proposed by Pao [13], is a single-layer NN with functionally mapped inputs. It is capable of forming arbitrarily complex decision regions. The prime advantage of the FLN is its reduced computational complexity without any sacrifice on its performance [7, 17, 19, 22]. For example, the FLN-based nonlinear equalizers offer better performance than the MLP equalizer, the RBF equalizer and the polynomial perceptron network (PPN) equalizer [19, 22] in terms of the mean square error level, convergence rate, and computational complexity. In order to approximate dynamical uncertainties during hypersonic flight, we propose a new RNN, the BRFLN. This network not only possesses characteristics as those of the FLN, but also has a lower computational complexity than RNNs with hidden layers. Hence it is quite suitable for the control realization of AHVs.

Since a feedforward NN combined with recurrent feedback and input time delays can effectively capture the optimal temporal profiles of a dynamical system, the FLN outputs are directly fed into the input layer to build the structure of BRFLN. In addition, we add the self-feedback links on the context nodes so as to simulate high-order systems [21]. The BRFLN does not have hidden layers, which greatly reduces the computation loads of the network and enables the BRFLN to be employed in the online training. The architecture can be shown as Fig. 1.

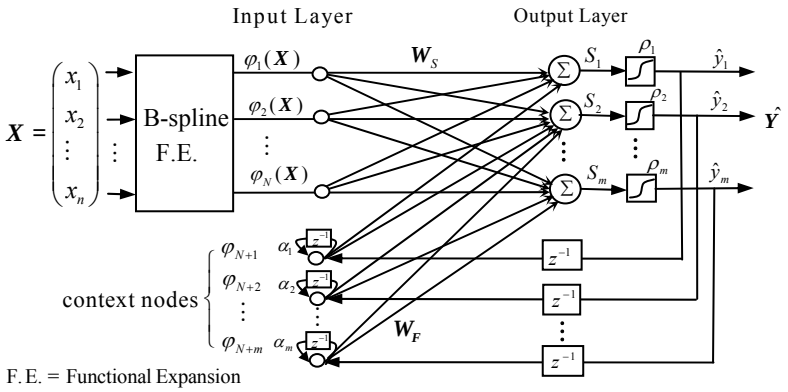


Figure 1: Structure of the BRFLN

The BRFLN output is described by

$$\boldsymbol{v}_{ad} = \hat{\boldsymbol{Y}}(\boldsymbol{k}+1) = \rho\left(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{X}(k+1),\hat{\boldsymbol{Y}}(k),\boldsymbol{\alpha}(k))\right) \tag{8}$$

where the weight matrix $\boldsymbol{W} = \left[\boldsymbol{W}_S^{\mathrm{T}}, \boldsymbol{W}_F^{\mathrm{T}}\right]^{\mathrm{T}} \in \boldsymbol{R}^{(N+m)\times m}$, $\boldsymbol{W}_F = \begin{bmatrix} w_{1,\mathrm{N+1}} & w_{2,\mathrm{N+1}} & \cdots & w_{\mathrm{m,N+1}} \\ w_{1,\mathrm{N+2}} & w_{2,\mathrm{N+2}} & \cdots & w_{\mathrm{m,N+2}} \\ \vdots & \vdots & & \vdots \\ w_{1,\mathrm{N+m}} & w_{2,\mathrm{N+m}} & \cdots & w_{\mathrm{m,N+m}} \end{bmatrix}$, and $\boldsymbol{W}_S = \begin{bmatrix} w_{11} & w_{21} & \cdots & w_{\mathrm{m1}} \\ w_{12} & w_{22} & \cdots & w_{\mathrm{m2}} \\ \vdots & \vdots & & \vdots \\ w_{1\mathrm{N}} & w_{2\mathrm{N}} & \cdots & w_{\mathrm{mN}} \end{bmatrix}$. $\boldsymbol{X} \in \boldsymbol{A} \subset \boldsymbol{R}^n$ is the input pattern vector and $\boldsymbol{\alpha} = (\alpha_1 \cdots \alpha_m)^{\mathrm{T}}$ is the self-connection weight matrix. $\boldsymbol{\Phi} = [\boldsymbol{\Phi}_1^{\mathrm{T}}(\boldsymbol{X}), \boldsymbol{\Phi}_2^{\mathrm{T}}(\hat{\boldsymbol{Y}}, \boldsymbol{\alpha})]^{\mathrm{T}} \in \boldsymbol{R}^{N+m}$ and $\boldsymbol{\Phi}_1(\boldsymbol{X}) = (\varphi_1(\boldsymbol{X})\ \varphi_2(\boldsymbol{X}) \cdots \varphi_{\mathrm{N}}(\boldsymbol{X}))^{\mathrm{T}} \in \boldsymbol{R}^N$ denotes the basis function matrix satisfying: 1) $\varphi_1 = 1$, 2) the subset $\mathcal{B}_j = \{\varphi_i \in \mathcal{L}(\boldsymbol{A})\}_{i=1}^{j}$ $(j = 1, \cdots, N)$ is a linearly independent set, and 3) $\sup_j [\sum_{i=1}^{j} \|\varphi_i\|_A^2]^{1/2} < \infty$. Such orthogonal functions include Chebyshev, Legendre, Hermite polynomials etc. Different from traditional FLN, the proposed network employs B-spline basis functions as mapped inputs. In many application fields, the spline approximation is attractive in terms of some properties such as continuity, local controllability, low approximation error and simple implementation. It can reach a higher accuracy than polynomials approximation. For instance, the central B-spline function is denoted by [15]

$$N_1(x) = \begin{cases} -|x|+1, & |x| < 1 \\ 0, & |x| \geq 1 \end{cases}, \qquad N_2(x) = \begin{cases} -x^2 + 3/4, & |x| \leq 1/2 \\ 1/2 \cdot (x^2 - 3|x| + 9/4), & 1/2 \leq |x| \leq 3/2, \end{cases}$$

$$N_3(x) = \begin{cases} 1/2 \cdot |x|^3 - x^2 + 2/3, & |x| \leq 1 \\ -1/6 \cdot |x|^3 + x^2 - 2|x| + 4/3, & 1 < |x| < 2 \end{cases}, \quad \cdots, \quad N_n(x) = \sum_{j=0}^{n+1} (-1)^j C_{n+1}^j \cdot (x + \frac{n+1}{2} - j)_+^n \Big/ n!, \quad -\infty < x < \infty \tag{9}$$

where $(x+a)_+^n = \begin{cases} (x+a)^n & x+a \geq 0 \\ 0 & x+a < 0. \end{cases}$ The support set of B-spline functions is described by $SuppN_n \subseteq \left(-(n+1)/2, (n+1)/2\right)$ and $N_i = 0$ when x is outside of $SuppN_i$. Generally, we are concerned mainly with splines of order $n \leq 3$ as these are simple for implementation and most frequently used in practice, i.e.

$$\boldsymbol{\Phi}_1(\boldsymbol{X}) = (1 \quad N_1(x_1) \quad \cdots \quad N_3(x_1)\, N_1(x_2) \quad \cdots N_3(x_2) \cdots \quad \cdots N_3(x_n))^{\mathrm{T}} \in \boldsymbol{R}^N$$

and $SuppN \subseteq (-2, 2)$.

Additionally, let $\boldsymbol{S}$ defined by $\boldsymbol{S} = \boldsymbol{W}^{\mathrm{T}}\boldsymbol{\Phi} = (S_1 \quad S_2 \quad \cdots \quad S_m)^{\mathrm{T}}$, then $\rho(\boldsymbol{S}) = (\rho_1 \quad \rho_2 \quad \cdots \quad \rho_m)^{\mathrm{T}}$ will be expressed as

$$\rho_j = \tanh(S_j) = (1 - e^{-2S_j}) / (1 + e^{-2S_j}), \quad j = 1,2,\cdots,m, \qquad \rho_j' = 1 - \rho_j^2. \tag{10}$$

3.2 Self-connection Weights Learned by SPSO

There are two parts of parameters to be determined in the BRFLN. Firstly, the feedforward and feedback weights $\boldsymbol{W}$ need the online learning. The adaptive law of $\boldsymbol{W}$ is derived based on Lyapunov stability theorem so as to guarantee the stability of the close-loop system. This content will be discussed in Section 4. Secondly, it is important to notice that the input layer consists of the functional expansion outputs and context nodes outputs. Each context node gives a summation at the instant k between its input $\hat{y}_p$ and its previous output weighted by a self-connection weight α_p. The value range of α_p is $[0,1)$. Then, the output of context nodes is denoted by

$$\boldsymbol{\Phi}_2(\hat{\boldsymbol{Y}}, \boldsymbol{\alpha}) = (\varphi_{N+1} \quad \varphi_{N+2} \ldots \varphi_{N+m})^{\mathrm{T}} \in \boldsymbol{R}^m,$$

and

$$\varphi_i(k+1) = \alpha_p \varphi_i(k) + \hat{y}_p(k), \quad i = N+1, \cdots, N+m; \quad p = 1, \cdots, m \tag{11}$$

where $\varphi_i(k+1)$ represents the output at the instant *k*+1 and we define that $|\varphi_i(k+1)| \le 1.0$.

The self-connection gives the nodes the capacity to memorize a certain past of the input data and simulate high-order systems. The easier way to obtain α_p is to fix it a priori, but how to get this proper value is actually a problem. Moreover, fixed α_p will influence the adaptability and approximation precision of the BRFLN. In this research, we consider employing a swarm intelligent optimizer, Stochastic Particle Swarm Optimizer (SPSO) to learn α_p.

Particle Swarm Optimizer (PSO) [9] is newly developed evolutionary technique and it is an algorithm to simulate the movement of flocks of birds. Like other population-based search algorithms, PSO is initialized with a swarm of random solutions (particles). Each particle flies in D-dimensional problem space with a velocity which is adjusted at each time step. The particle flies towards a position which depends on its own past best position and the best position of its neighbors. The quality of a particle position is evaluated by a problem-specific objective function (fitness). Due to the simple concept and quick convergence, PSO has gained much attention and wide applications in different fields nowadays. However, the performance of PSO greatly depends on its parameters and it suffers from the "curse of dimensionality". Moreover, PSO may prematurely converge on suboptimal solutions which are not even local extrema.

Stochastic PSO [2] is an improvement upon conventional PSO. It has strong exploration ability, fast convergence speed and is guaranteed to convergence at the global optimization solution with probability one. For the SPSO, the inertia weight *w* in the conventional PSO updating equations [9] is equal to zero. This operation will reduce its global search capability, but increase the local search power. In order to improve global capability, SPSO retains the best history position and generates a new position randomly in the searching space. The algorithm is described as

$$\begin{aligned} &V_i(t+1) = c_1 r_1 (P_i - x_i(t)) + c_2 r_2 (P_g - x_i(t)) \\ &x_i(t+1) = x_i(t) + V_i(t+1) \\ &P_j = x_j(t+1) \qquad \text{generate randomly} \\ &P_i = \begin{cases} P_i, & f(P_i) < f(x_i(t+1)) \\ x_i(t+1), & f(P_i) \ge f(x_i(t+1)) \end{cases} \\ &P_g' = \arg\min\{f(P_i) \mid i = 1, \cdots, s\}, \quad P_g = \arg\min\{f(P_g'), f(P_g)\} \end{aligned} \tag{12}$$

where $V_i(t)$ is the current flight velocity of the *i*th particle, $x_i(t)$ represents the current position of the *i*th particle, c_1 and c_2 are two positive acceleration constants, r_1 and r_2 are two random constants satisfying the uniform distribution U(0,1), P_j is the best history position of the swarm, P_i is the personal best position of the *i*th particle and P_g is the best position of the entire swarm. If $P_j = P_g$, then particle *j*'s position $x_j(t + 1)$ needs initializing randomly and other particles are manipulated according to the particle updating equation in (12). Thus, the global search capability can be enhanced.

Due to the limited space of the paper, we do not expatiate on the theory of SPSO which is discussed in [2]. We emphasize how to realize the online learning of self-connection weight α_p with the SPSO. In this study, $(\alpha_1, \cdots, \alpha_p, \cdots, \alpha_m)$ is just a m-dimension particle of the swarm. The selection of the particle evaluation function borrows the thought of the OGPC, that is, the prediction output defined on finite horizon is carried out via Taylor series expansion. Consequently, we use the performance index (2) as the evaluation function and $\boldsymbol{e}(t+\tau)$ is described by

$$\boldsymbol{e}(t+\tau) = \boldsymbol{\Gamma}(\tau) \cdot (\bar{\boldsymbol{Y}}(t) - \bar{\boldsymbol{Y}}_r(t)). \tag{13}$$

As (3) and (13) show, we know that the predictive error $\boldsymbol{e}(t+\tau)$ is the function of $\boldsymbol{x}$, $\boldsymbol{u}$ and $\boldsymbol{D}$. The uncertainties $\boldsymbol{D}$ will be replaced by $\boldsymbol{v}_{ad} + \boldsymbol{v}_r$ during the controller design. Since $\boldsymbol{v}_{ad}$ is related to $\alpha_p (p = 1, \cdots, m)$ from (8), different particle $(\alpha_1, \cdots, \alpha_p, \cdots, \alpha_m)$ will own different value of $\boldsymbol{e}(t+\tau)$ and different value of the evaluation function accordingly. The particle which possesses the minimum evaluation value is the best individual. Thus, there are following assumptions on the optimal weight matrix and output of the BRFLN.

Assumption 1: On a compact region $\Theta_{\boldsymbol{e}} \subset \boldsymbol{R}^m$, define the optimal weight matrix as

$$\boldsymbol{W}^* = \arg\min_{\boldsymbol{W}} \left\{ \sup_{\boldsymbol{e} \in \Theta_{\boldsymbol{e}}} \left\| \boldsymbol{D} - \boldsymbol{v}_{ad} \right\| \right\} \tag{14}$$

where $\boldsymbol{e}$ is the BRFLN input vector, $\boldsymbol{W}^*$ satisfies $\|\boldsymbol{W}^*\| \le \bar{W}$ and $\|\cdot\|$ denotes the Frobenius norm.

Assumption 2: On a compact region $\Theta_e \subset \boldsymbol{R}^m$, define the optimal output of the BRFLN as

$$\boldsymbol{v}_{ad}^* = \rho(\boldsymbol{W}^{*\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{X},\hat{\boldsymbol{Y}},\boldsymbol{\alpha}^*)) = \boldsymbol{D} - \boldsymbol{\varepsilon}, \quad \|\boldsymbol{\varepsilon}\| \le \psi_\varepsilon^* \tag{15}$$

where $\boldsymbol{\alpha}^*$ is the optimal self-connection weight matrix, $\boldsymbol{\varepsilon}$ is the approximation error of the BRFLN, and $\psi_\varepsilon^* \ge 0$ is an unknown bound.

3.3 The Design of BRFLN-based OGPC

Now we can give the structure of the proposed BRFLN-based OGPC method.

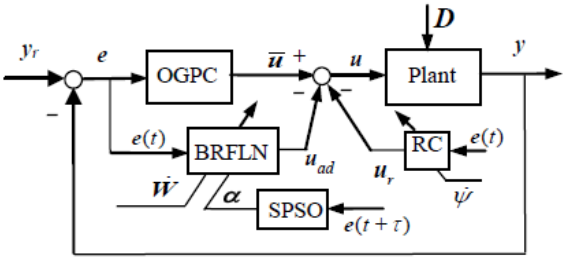


Figure 2: Structure of BRFLN-based robust OGPC

In this figure, $\boldsymbol{u}_{ad}$ and $\boldsymbol{u}_r$ constitute the adjustment of the control law. Using (7) and (8), we have $\boldsymbol{u}_{ad} = \boldsymbol{G}_0(\boldsymbol{x})\rho(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha}))$ where W and $\boldsymbol{\alpha}$ are in need of online learning. Additionally, the RC gain, i.e. ψ, has its adaptive law which may lead to a less conservative controller design. The derivation process of the adaptive laws will be discussed as Section 4.

4 The Stability Analysis of Close-loop System

It is necessary to obtain the error state equation of the close-loop system before the stability analysis. Let $\boldsymbol{K}$ in (6) be partitioned as $(\boldsymbol{K}_1^{\mathrm{T}} \cdots \boldsymbol{K}_i^{\mathrm{T}} \cdots \boldsymbol{K}_m^{\mathrm{T}})^{\mathrm{T}}$ where $\boldsymbol{K}_i = (k_{i,0}, 0, \cdots, k_{i,1}, 0, \cdots, \cdots, k_{i,\rho-1}, 0, \cdots) \in \boldsymbol{R}^{1\times m\rho}$. Then, substituting (7) into (3) and rearranging the expression $\boldsymbol{y}^{[\rho]}(t)$ yield

$$\dot{\boldsymbol{E}} = \boldsymbol{A}\boldsymbol{E} + \boldsymbol{B}\boldsymbol{H}\left(\boldsymbol{D} - \boldsymbol{v}_{ad} - \boldsymbol{v}_r\right) \tag{16}$$

where $\boldsymbol{E} = (\hat{\boldsymbol{e}}_1, \hat{\boldsymbol{e}}_2, \cdots, \hat{\boldsymbol{e}}_m)^{\mathrm{T}} \in \boldsymbol{R}^{m\rho}$, $\hat{\boldsymbol{e}}_i = (e_i^{[0]}, e_i^{[1]}, \cdots, e_i^{[\rho-1]})^{\mathrm{T}} \in \boldsymbol{R}^{\rho}$, $\boldsymbol{A}=diag(\boldsymbol{A}_1, \cdots, \boldsymbol{A}_m) \in \boldsymbol{R}^{m\rho\times m\rho}$, $\boldsymbol{B}=diag(\boldsymbol{B}_1, \cdots, \boldsymbol{B}_m)$ $\in \boldsymbol{R}^{m\rho\times m}$, $\boldsymbol{A}_i = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -k_{i,0} & -k_{i,1} & -k_{i,2} & \cdots & -k_{i,\rho-1} \end{bmatrix} \in \boldsymbol{R}^{\rho\times\rho}$ and $\boldsymbol{B}_i = (0 \quad 0 \quad \cdots \quad 1)^{\mathrm{T}} \in \boldsymbol{R}^{\rho}$. From (8) and (15), we get

$$\boldsymbol{D} - \boldsymbol{v}_{ad} = \boldsymbol{v}_{ad}^* + \boldsymbol{\varepsilon} - \boldsymbol{v}_{ad} = \rho\left(\boldsymbol{W}^{*\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha})\right) + \boldsymbol{\varepsilon} - \rho\left(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha})\right). \tag{17}$$

Then, $\rho\left(\boldsymbol{W}^{*\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha})\right)$ can be expressed as

$$\rho\left(\boldsymbol{W}^{*\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha})\right) = \rho\left[\boldsymbol{W}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha}) + (\boldsymbol{W}^{*\mathrm{T}} - \boldsymbol{W}^{\mathrm{T}})\boldsymbol{\Phi}(\boldsymbol{\alpha})\right] = \rho\left(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha}) + \tilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha})\right) \tag{18}$$

where $\tilde{\boldsymbol{W}} = \boldsymbol{W}^* - \boldsymbol{W}$. Whence the Taylor series expansion for (18) can be written as

$$\rho\left(\boldsymbol{W}^{*\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha})\right) = \rho(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha})) + \rho'(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha}))\tilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha}) + \boldsymbol{o}(\tilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha}))^2 \tag{19}$$

where $\rho'(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha})) = d\rho/dz|_{z=\boldsymbol{W}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha})}$ and $\boldsymbol{o}(\tilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha}))^2$ represents higher-order terms. Denoting $\boldsymbol{\rho}' = \rho'(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha}))$ and substituting (19) into (17), we have

$$\boldsymbol{D} - \boldsymbol{v}_{ad} = \boldsymbol{\rho}'\tilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha}) + \boldsymbol{o}(\tilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha}))^2 + \boldsymbol{\varepsilon}. \tag{20}$$

Hence, the system error state equation can be described by

$$\dot{\boldsymbol{E}} = \boldsymbol{AE} + \boldsymbol{BH}\left(\boldsymbol{\rho}'\tilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha}) + \boldsymbol{o}(\tilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha}))^2 + \boldsymbol{\varepsilon} - \boldsymbol{v}_r\right). \tag{21}$$

Now we consider the expression $\boldsymbol{r}^{\mathrm{T}}\boldsymbol{o}(\tilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha}))^2$. The use of (19) yields

$$\begin{aligned}\boldsymbol{r}^{\mathrm{T}}\boldsymbol{o}(\tilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha}))^2 &= \boldsymbol{r}^{\mathrm{T}}[\rho(\boldsymbol{W}^{*\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha})) - \rho(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha})) - \rho'(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha}))\tilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha})] \\ &= \boldsymbol{r}^{\mathrm{T}}\rho(\boldsymbol{W}^{*\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha})) - \boldsymbol{r}^{\mathrm{T}}\rho(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha})) + \boldsymbol{r}^{\mathrm{T}}\boldsymbol{\rho}'\boldsymbol{W}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha}) - \boldsymbol{r}^{\mathrm{T}}\boldsymbol{\rho}'\boldsymbol{W}^{*\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha}),\end{aligned}$$

with $\boldsymbol{r}$ a vector to be mentioned in Theorem 1. Therefore,

$$\begin{aligned}\left|\boldsymbol{r}^{\mathrm{T}}\boldsymbol{o}(\tilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha}))^2\right| &\le \left|\rho(\boldsymbol{W}^{*\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha})) - \rho(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha}))\right|\|\boldsymbol{r}\|_1 + tr(\boldsymbol{\rho}'\boldsymbol{W}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha})\boldsymbol{r}^{\mathrm{T}}) - tr(\boldsymbol{W}^{*\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha})\boldsymbol{r}^{\mathrm{T}}\boldsymbol{\rho}') \\ &\le 2\|\boldsymbol{r}\|_1 + \left\|\boldsymbol{\rho}'\boldsymbol{W}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha})\boldsymbol{r}^{\mathrm{T}}\right\| + \left\|\boldsymbol{W}^{*}\right\|\cdot\left\|\boldsymbol{\Phi}(\boldsymbol{\alpha})\boldsymbol{r}^{\mathrm{T}}\boldsymbol{\rho}'\right\|\end{aligned} \tag{22}$$

where $\|\cdot\|_1$ represents the absolute value sum of matrix elements. Let $s_o = (\|\boldsymbol{r}\|_1 + \left\|\boldsymbol{\rho}'\boldsymbol{W}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha})\boldsymbol{r}^{\mathrm{T}}\right\| + \left\|\boldsymbol{\Phi}(\boldsymbol{\alpha})\boldsymbol{r}^{\mathrm{T}}\boldsymbol{\rho}'\right\|)/\|\boldsymbol{r}\|$ and $\psi_o^* = \max\left\{2, \left\|\boldsymbol{W}^{*}\right\|\right\}$. So the inequality (22) satisfies

$$\left|\boldsymbol{r}^{\mathrm{T}}\boldsymbol{o}(\tilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha}))^2\right| \le \psi_o^* s_o \|\boldsymbol{r}\|. \tag{23}$$

Then, define

$$\sigma^* = s_o + 1,\ \psi_m^* = \max\left\{\psi_o^*, \psi_\varepsilon^*\right\},\ \tilde{\psi} = \psi_m^* - \psi \tag{24}$$

where $\tilde{\psi}$ is the estimation error of the robust gain. The stability of the close-loop system with BRFLN-based OGPC method is discussed in Theorem 1.

Theorem 1. Assume that the system (1) is controlled by (7) and (25). If the online adjustment parameters are tuned by (26), then the close-loop system error $\boldsymbol{E}$ and estimation errors ($\tilde{\boldsymbol{W}}$, $\tilde{\psi}$) are uniformly ultimately bounded.

$$\boldsymbol{v}_{ad} = \rho(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha})),\ \ \boldsymbol{v}_r = \psi\sigma^*\tanh(\sigma^*\boldsymbol{r}/\delta),\ \ \boldsymbol{r} = (\boldsymbol{BH})^{\mathrm{T}}\boldsymbol{PE}(t) \tag{25}$$

$$\dot{\boldsymbol{W}} = \boldsymbol{\Gamma}_W(\boldsymbol{\Phi}(\boldsymbol{\alpha})\boldsymbol{r}^{\mathrm{T}}\boldsymbol{\rho}' - K_W\boldsymbol{W}),\ \ \dot{\psi} = \lambda_\psi\left(\sigma^*\boldsymbol{r}^{\mathrm{T}}\tanh(\sigma^*\boldsymbol{r}/\delta) - K_\psi\psi\right) \tag{26}$$

where δ, K_W, λ_ψ and K_ψ are positive design constants, $\boldsymbol{\Gamma}_W$ is a positive definite matrix and $\tanh(x)$ denotes the hyperbolic tangent function.

Proof. Let the Lyapunov function candidate be given by

$$V = \frac{1}{2}\boldsymbol{E}^{\mathrm{T}}\boldsymbol{PE} + \frac{1}{2}tr(\tilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{\Gamma}_W^{-1}\tilde{\boldsymbol{W}}) + \frac{1}{2\lambda_\psi}\tilde{\psi}^2 \tag{27}$$

where $\boldsymbol{P} = diag(\boldsymbol{P}_1, \cdots, \boldsymbol{P}_m) \in \boldsymbol{R}^{m\rho\times m\rho}$ and $\boldsymbol{P}_i$ is a positive definite matrix which satisfies

$$\boldsymbol{P}_i\boldsymbol{A}_i + \boldsymbol{A}_i^{\mathrm{T}}\boldsymbol{P}_i = -\boldsymbol{Q}_i,\ \boldsymbol{Q}_i = \boldsymbol{Q}_i^{\mathrm{T}} > 0 \tag{28}$$

with $\boldsymbol{A}_i$ the system state matrix and $\boldsymbol{Q}_i$ an arbitrary definite matrix. Moreover, $\boldsymbol{P}$ satisfies

$$0 < \beta_1\boldsymbol{I} \le \boldsymbol{P}(t) \le \beta_2\boldsymbol{I} \qquad \forall t \ge t_0 \tag{29}$$

where β_1 and β_2 are positive constants.

Differentiating (27) and substituting (21), (25) and (28) into $\dot{V}$ yield

$$\begin{aligned}\dot{V} &= \frac{1}{2}\boldsymbol{E}^{\mathrm{T}}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{P} + \boldsymbol{PA})\boldsymbol{E} + \boldsymbol{E}^{\mathrm{T}}\boldsymbol{PBH}[\boldsymbol{\rho}'\tilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha}) + \boldsymbol{o}(\tilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha}))^2 + \boldsymbol{\varepsilon} - \boldsymbol{v}_r] + tr(\tilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{\Gamma}_W^{-1}\dot{\tilde{\boldsymbol{W}}}) + \frac{1}{\lambda_\psi}\tilde{\psi}\dot{\tilde{\psi}} \\ &= -\frac{1}{2}\boldsymbol{E}^{\mathrm{T}}\boldsymbol{QE} + \boldsymbol{r}^{\mathrm{T}}\left[\boldsymbol{\rho}'\tilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha}) + \boldsymbol{o}(\tilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha}))^2 + \boldsymbol{\varepsilon} - \boldsymbol{v}_r\right] + tr[\tilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{\Gamma}_W^{-1}(\dot{\boldsymbol{W}}^{*} - \dot{\boldsymbol{W}})] \quad + \frac{1}{\lambda_\psi}\tilde{\psi}(\dot{\psi}^{*} - \dot{\psi}) \\ &= -\frac{1}{2}\boldsymbol{E}^{\mathrm{T}}\boldsymbol{QE} + \boldsymbol{r}^{\mathrm{T}}\left[\boldsymbol{o}(\tilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha}))^2 + \boldsymbol{\varepsilon} - \boldsymbol{v}_r\right] + \boldsymbol{r}^{\mathrm{T}}\boldsymbol{\rho}'\tilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha}) + tr(-\tilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{\Gamma}_W^{-1}\dot{\boldsymbol{W}} + \tilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha})\boldsymbol{r}^{\mathrm{T}}\boldsymbol{\rho}' - \tilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha})\boldsymbol{r}^{\mathrm{T}}\boldsymbol{\rho}') - \frac{1}{\lambda_\psi}\tilde{\psi}\dot{\psi}.\end{aligned}$$

Choosing a BRFLN online tuning law and the adaptive law for $\psi(t)$ as (26), we get

$$\dot{V} = -\frac{1}{2}\boldsymbol{E}^{\mathrm{T}}\boldsymbol{QE} + \boldsymbol{r}^{\mathrm{T}}\boldsymbol{o}(\tilde{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{\alpha}))^2 + K_W tr[\tilde{\boldsymbol{W}}^{\mathrm{T}}(\boldsymbol{W}^{*} - \tilde{\boldsymbol{W}})] + \boldsymbol{r}^{\mathrm{T}}\boldsymbol{\varepsilon} - \psi\sigma^*\boldsymbol{r}^{\mathrm{T}}\tanh(\sigma^*\boldsymbol{r}/\delta) - \tilde{\psi}\sigma^*\boldsymbol{r}^{\mathrm{T}}\tanh(\sigma^*\boldsymbol{r}/\delta) + K_\psi\tilde{\psi}\psi\ .$$

From (15), (23) and (24), we obtain

$$\dot{V} \le -\frac{1}{2}\underline{\lambda}(\boldsymbol{Q})\|\boldsymbol{E}\|^2 + \psi_o^* s_o\|\boldsymbol{r}\| + \psi_\varepsilon^*\|\boldsymbol{r}\| + K_W tr(\|\tilde{\boldsymbol{W}}\|\overline{W} - \|\tilde{\boldsymbol{W}}\|^2) - \psi_m^*\sigma^*\boldsymbol{r}^{\mathrm{T}}\tanh(\sigma^*\boldsymbol{r}/\delta) + K_\psi\tilde{\psi}(\psi_m^* - \tilde{\psi})$$

$$\le -\frac{1}{2}\underline{\lambda}(\boldsymbol{Q})\|\boldsymbol{E}\|^2 + \psi_m^*\sigma^*\|\boldsymbol{r}\| - \psi_m^*\sigma^*\boldsymbol{r}^{\mathrm{T}}\tanh(\sigma^*\boldsymbol{r}/\delta) + K_W(\|\tilde{\boldsymbol{W}}\|\bar{W} - \|\tilde{\boldsymbol{W}}\|^2) + K_\psi(\frac{1}{2}\psi_m^{*2} - \frac{1}{2}\tilde{\psi}^2) \tag{30}$$

where $\underline{\lambda}(\cdot)$ denotes the matrix minimum eigenvalue. Next applying Lemma 1 in [14] yields

$$\psi_m^*\sigma^*\|\boldsymbol{r}\| - \psi_m^*\sigma^*\boldsymbol{r}^{\mathrm{T}}\tanh(\sigma^*\boldsymbol{r}/\delta) \le \psi_m^*\kappa\delta$$

where $\kappa = m\zeta$, m is the dimension of r and $\varsigma = 0.2785$. Therefore, (30) becomes

$$\dot{V} \le -\frac{1}{2}\underline{\lambda}(\boldsymbol{Q})\|\boldsymbol{E}\|^2 + \psi_m^*\kappa\delta + K_W(\frac{1}{2}\bar{W}^2 - \frac{1}{2}\|\tilde{\boldsymbol{W}}\|^2) + \frac{1}{2}K_\psi\psi_m^{*2} - \frac{1}{2}K_\psi\tilde{\psi}^2 .$$

If we define $\bar{\alpha} = 2\psi_m^*\kappa\delta + K_W\bar{W}^2 + K_\psi\psi_m^{*2} > 0$, then (30) can be written as

$$\dot{V} \le -\frac{1}{2}\underline{\lambda}(\boldsymbol{Q})\|\boldsymbol{E}\|^2 - \frac{1}{2}K_W\|\tilde{\boldsymbol{W}}\|^2 - \frac{1}{2}K_\psi\tilde{\psi}^2 + \frac{1}{2}\bar{\alpha} . \tag{31}$$

Thus, $\dot{V}$ is negative when one of the following inequalities is satisfied.

$$\|\boldsymbol{E}\| > \sqrt{\bar{\alpha}/\underline{\lambda}(\boldsymbol{Q})}, \quad \|\tilde{\boldsymbol{W}}\| > \sqrt{\bar{\alpha}/K_W}, \quad |\tilde{\psi}| > \sqrt{\bar{\alpha}/K_\psi} . \tag{32}$$

Set $C_1 = \min(\underline{\lambda}(\boldsymbol{Q})/\bar{\lambda}(\boldsymbol{P}), K_W\underline{\lambda}(\boldsymbol{\Gamma}_W), K_\psi\lambda_\psi)$ and $C_2 = \bar{\alpha}/2$. Then, (31) may be rewritten as

$$\dot{V} \le -\underline{\lambda}(\boldsymbol{Q})/\bar{\lambda}(\boldsymbol{P})\cdot\frac{1}{2}\boldsymbol{E}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{E} - K_W\underline{\lambda}(\boldsymbol{\Gamma}_W)\cdot\frac{1}{2}tr(\tilde{\boldsymbol{W}}^T\boldsymbol{\Gamma}_W^{-1}\tilde{\boldsymbol{W}}) - K_\psi\lambda_\psi\cdot\frac{1}{2\lambda_\psi}\tilde{\psi}^2 + C_2 \le -C_1V + C_2 . \tag{33}$$

Integrating (33) over [t_0, t], we have

$$V(t) \le \frac{C_2}{C_1} + (V(t_0) - \frac{C_2}{C_1})e^{-C_1(t-t_0)} . \tag{34}$$

Hence, the errors including $\boldsymbol{E}$, $\tilde{\boldsymbol{W}}$ and $\tilde{\psi}$ turn out to be uniformly ultimately bounded. The control law that minimizes the performance index (2) is given by (7) and (25). □

Remark 1: The main design work lies in the online learning of the BRFLN and adaptive adjustment of the robust gain. In fact, the control algorithm is not limited to the OGPC. In this paper, the OGPC is utilized mainly because it is a high efficient algorithm with good performance, which is crucial for the realization of attitudes control system. If some proper nonlinear control methods are combined with the BRFLN, we may apply it to the dynamic systems which don't satisfy the assumptions of OGPC. Anyway, a nonlinear control algorithm should be selected according to the object characteristics.

5 Controller Design for AHVS and Simulation Results

The AHV simulation model comes from the hypersonic vehicle of winged-cone configuration [10, 23]. Six degree-of-freedom and twelve states equations can be simplified to the affine nonlinear functions as

$$\begin{cases} \dot{\boldsymbol{\Omega}} = \boldsymbol{f}_s + \boldsymbol{g}_{s1}\boldsymbol{\omega} + \boldsymbol{g}_{s2}\boldsymbol{D}_s \\ \boldsymbol{y}_s = \boldsymbol{\Omega} \end{cases} \qquad \begin{cases} \dot{\boldsymbol{\omega}} = \boldsymbol{f}_f + \boldsymbol{g}_{f1}\boldsymbol{M}_C + \boldsymbol{g}_{f2}\boldsymbol{D}_f \\ \boldsymbol{y}_f = \boldsymbol{\omega} \end{cases} \tag{35}$$

where $\boldsymbol{\Omega} = [\alpha, \beta, \mu]^{\mathrm{T}}$ is the slow-loop state vector, the AHV attitude angle, and $\boldsymbol{\omega} = [p, q, r]^{\mathrm{T}}$ is the fast-loop state vector which includes three angular velocities. $\boldsymbol{M}_C = \boldsymbol{g}_{f\delta}\boldsymbol{\delta}_C \in \boldsymbol{R}^3$ is the AHV control moment. $\boldsymbol{\delta}_C$ denotes the control surface deflection. $\boldsymbol{g}_{s1}, \boldsymbol{g}_{f1} \in \boldsymbol{R}^{3\times3}$ are invertible matrixes and $\boldsymbol{f}_s, \boldsymbol{f}_f \in \boldsymbol{R}^3$, $\boldsymbol{g}_{s2} = \boldsymbol{g}_{f2} = \boldsymbol{I}_{3\times3}$. The concrete expressions of above matrixes are specified in [23]. In general, they are functions of attitudes, angular velocities, aerodynamic coefficients, aerodynamic moment coefficients etc. $\boldsymbol{D}_s = [D_\alpha, D_\beta, D_\mu]^{\mathrm{T}}$ and $\boldsymbol{D}_f = [D_p, D_q, D_r]^{\mathrm{T}}$ are the dynamical uncertainties and disturbances upon the slow and fast loop respectively.

The application of (35) yields that the relative degree ρ is equal to 1, and the control order *r* is set zero. Then from (6) and (7), the slow-loop and fast-loop control laws of the AHV are given by

$$\boldsymbol{\omega} = \boldsymbol{u}_s = -(\boldsymbol{G}_1(\boldsymbol{x}))^{-1}(\boldsymbol{F}(\boldsymbol{x}) + \boldsymbol{K}\boldsymbol{M}_\rho - \boldsymbol{y}_r^{[\rho]}(t)) - \boldsymbol{G}_0(\boldsymbol{x})\boldsymbol{v}_{ad} - \boldsymbol{G}_0(\boldsymbol{x})\boldsymbol{v}_r = -\boldsymbol{g}_{s1}^{-1}(\boldsymbol{f}_s + \boldsymbol{K}_s\boldsymbol{e}_s - \dot{\boldsymbol{y}}_{sr}) - \boldsymbol{g}_{s1}^{-1}\boldsymbol{\omega}_{ad} - \boldsymbol{g}_{s1}^{-1}\boldsymbol{\omega}_r , \tag{36}$$

$$\boldsymbol{M}_C = \boldsymbol{u}_f = -\boldsymbol{g}_{f1}^{-1}(\boldsymbol{f}_f + \boldsymbol{K}_f\boldsymbol{e}_f - \dot{\boldsymbol{y}}_{fr}) - \boldsymbol{g}_{f1}^{-1}\boldsymbol{M}_{ad} - \boldsymbol{g}_{f1}^{-1}\boldsymbol{M}_r \tag{37}$$

where $\boldsymbol{K}_s = \bar{\boldsymbol{\Gamma}}_{rr}^{-1}\bar{\boldsymbol{\Gamma}}_{\rho r}^{\mathrm{T}} = diag\{3/(2T_s), 3/(2T_s), 3/(2T_s)\}$ and $\boldsymbol{K}_f = diag\{3/(2T_f), 3/(2T_f), 3/(2T_f)\}$. The adaptive OGPC structure for attitude and angular velocity tracking is shown in Fig. 3.

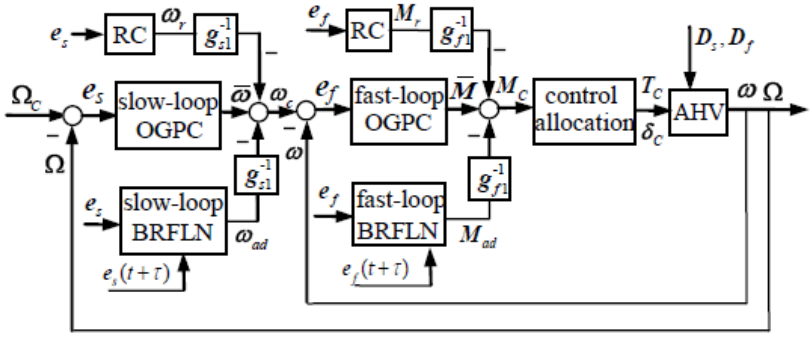


Figure 3: The BRFLN-based OGPC control structure of the AHV

Assume that the AHV is flying with the velocity of 2500m/s and flight height of 35km. $\alpha_0 = 0.5^\circ$, $\beta_0 = 0.2^\circ$, $\mu_0 = 0.0^\circ$ and $p_0 = q_0 = r_0 = 0$rad/s are the initial attitudes and angular velocities. The control surface limiting is $\pm 30^\circ$ and pseudo-inverse control allocation is employed in the system. The slow-loop predictive time Ts is set 0.45s, and that of the fast-loop is 0.4s. Suppose that there are -25%～+30% and -30%～+35% triangle-function-waving uncertainty of the aerodynamic coefficients and aerodynamic moment coefficients respectively. Besides, the disturbance moment upon the AHV fast-loop system is defined by

$$d_f = 6.0\times 10^5 \left(r_1\cdot(\cos(r_1\cdot 8t)+0.1)\cdot\cos(7t),\quad r_2\cdot(\sin(r_2\cdot 5t)+0.2),\quad r_3\cdot\cos(9t)\cdot\sin(r_3\cdot 6t)\right)^{\mathrm{T}}$$

where r_i is a random value within [0.5,1.5]. Then there exist dynamically changeable $\boldsymbol{D}_s$ and $\boldsymbol{D}_f$ in (35). Fig. 4 displays the control effect of the OGPC in the presence of uncertainties and disturbances. The results show that the closed-loop system is not stable and aero control surfaces vibrate during most of the simulation time. Hence, the AHV in hypersonic flight is sensitive to the parameter uncertainty and external disturbance. The single OGPC law can not satisfy the precision and robustness requirements.

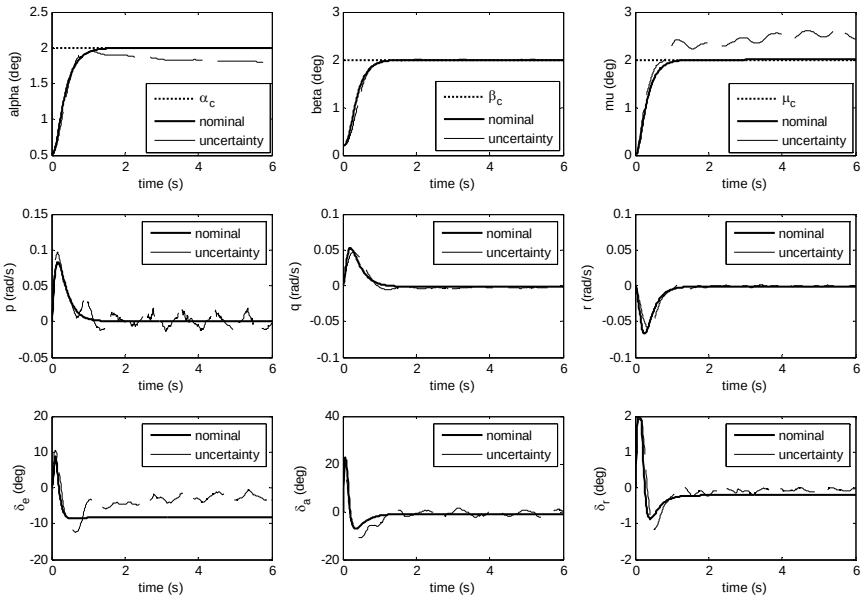
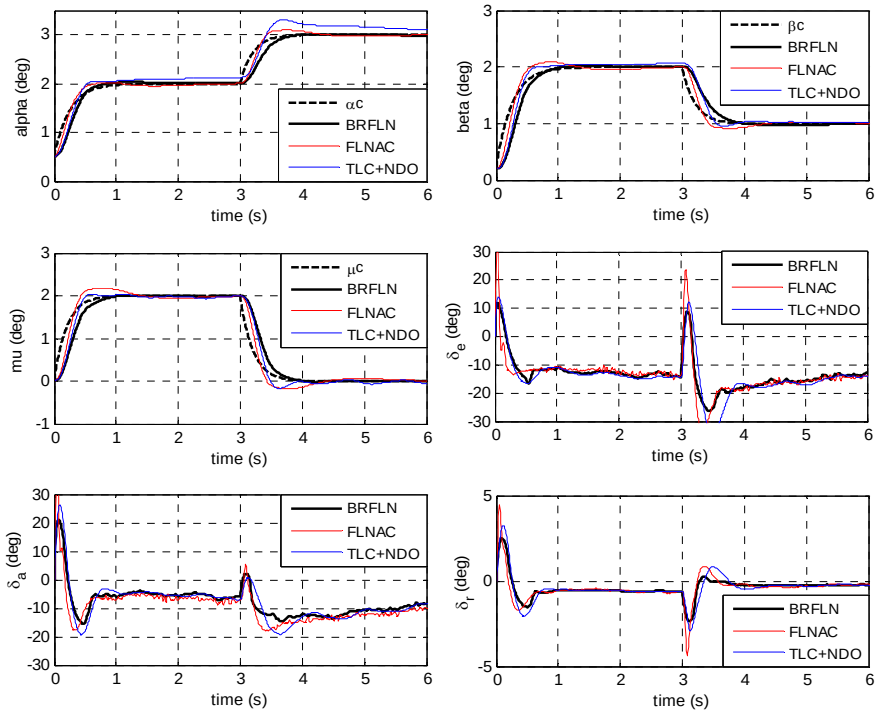


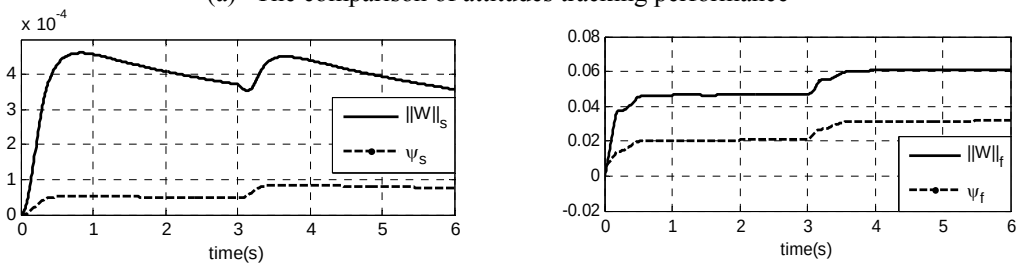
Figure 4: The attitudes tracking performance of the OGPC

Next, the slow and fast loop BRFLN adaptive laws are applied to the AHV attitudes control. The original inputs of slow-loop BRFLN, $\boldsymbol{e}_s = [\alpha-\alpha_c, \beta-\beta_c, \mu-\mu_c]^{\mathrm{T}}$, are expanded into 7-dimensional patterns by the two-order B-spline transformations of each of the three input signals. Fast-loop BRFLN has the same structure as that of the slow-loop. Other parameters are designed as: $\delta_s = 0.5$, $\boldsymbol{\Gamma}_{Ws} = 0.1\cdot\boldsymbol{I}_{10\times 10}$, $K_{Ws} = 0.5$, $\lambda_{\psi s} = 0.5$, $K_{\psi s} = 0.01$; $\delta_f = 0.1$, $\boldsymbol{\Gamma}_{Wf} = 0.1\cdot\boldsymbol{I}_{10\times 10}$, $K_{Wf} = 0.05$, $\lambda_{\psi f} = 0.01$, $K_{\psi f} = 0.1$. The initial values of slow and fast loop BRFLN weights are

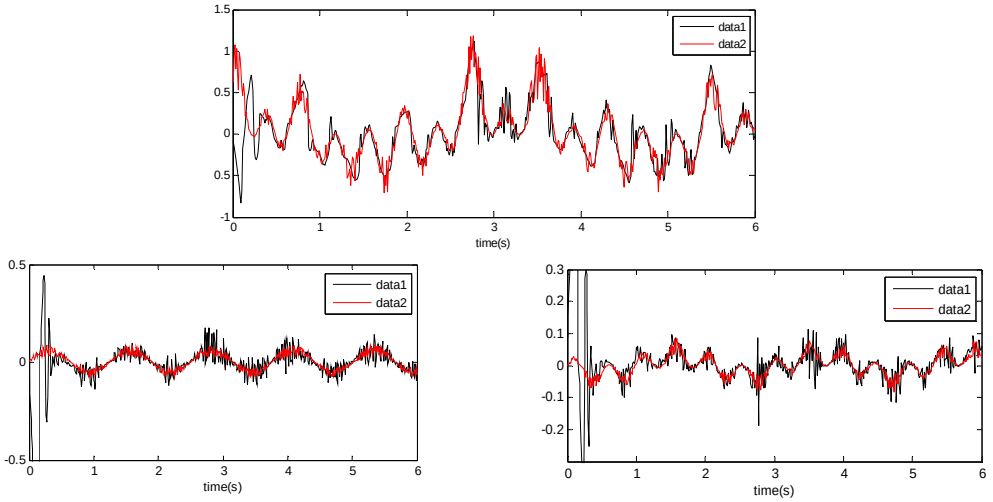
set zero, moreover $\psi_{s0}=0$ and $\psi_{f0}=0.005$. α_{ps} and α_{pf} are learned online once 0.5s by SPSO whose parameters are: Population size =10, c_1 =c_2=1.8, $V_{\max}$=2.0, Iteration times=30.



(a) The comparison of attitudes tracking performance



(b) The adaptive parameters of BRFLN-based OGPC



(c) The approximation effect of the fast-loop uncertainties

Figure 5: The attitudes tracking performance of BRFLN-based OGPC

The reference attitudes and the output tracking performance by the BRFLN-based OGPC law are shown as Fig.5. Additionally, the control results of OGPC with FLN adaptive control (FLNAC) method [3] and TLC with nonlinear disturbance observer (NDO) are also provided in the figure. The FLN is expanded into 10-dimensional patterns by the three-order Legendre transformations.

As Fig. 5(a) shows, the BRFLN-based OGPC possesses stronger adaptability over TLC+NDO whose parameters designed are fixed. It reaches higher accuracy and better dynamic performance than the FLNAC with OGPC. Besides, the control input of the BRFLN+OGPC holds smaller deflection and milder vibration than those of the FLNAC and TLC+NDO. In Fig. 5(c), we also provide the approximation performance of the BRFLN-based OGPC. Data 1 denotes the outputs of the BRFLN and robust item, and data 2 represents the fast-loop uncertainties in the system. The nice approximation effect is attained by the method designed.

6 Conclusion

The BRFLN-based OGPC method applied to the AHV control is presented in this paper. There exist dynamically changeable uncertainties/disturbances during hypersonic flight. Hence, system robustness is critical for the stable and precise control of the AHV. RNN possesses superior capability as compared to static mapping methods for the approximation of dynamical variation functions, so a new BRFLN adaptive learning method is combined with the controller design. Furthermore, the self-connection weights of the BRFLN are learned online by the SPSO in order to simulate high-order uncertainty functions and strengthen the adaptability of the network. Simulation results for the attitudes control of the AHV show that the proposed method possesses higher precision and stronger adaptability over the methods compared.

References

[1] Chen, W.H., D.J. Balance, and P.J. Gawthrop, Optimal control of nonlinear systems: a predictive control approach, *Automatica*, vol.39, no.6, pp.633–641, 2003.

[2] Cui, Z.H., and J.C. Zeng, A guaranteed global convergence particle swarm optimizer, *Lecture Notes in Computer Science*, vol.3066, pp.762–767, 2004.

[3] Du, Y.L., Q.X. Wu, C.S. Jiang, and L. Zhou, Robust optimal predictive control for a near-space vehicle based on functional link network disturbance observer, *Journal of Astronautics*, vol.30, no.4, pp.174–182, 2009.

[4] Fiorentini, L., M.A. Bolender, A. Serrani, and D.B. Doman, Robust nonlinear sequential loop closure control design for an air-breathing hypersonic vehicle model, *Proceeding of American Control Conference*, pp.3458–3463, 2008.

[5] Gawithrop, P.J., H. Demircioglu, and I.I. Siller-Alcala, Multivariable continuous-time generalized predictive control: a state-space approach to linear and nonlinear systems, *IEE Proc.- Control Theory Appl.*, vol.145, no.3, pp.241–250, 1998.

[6] Hsu, C.F., Adaptive recurrent neural network control using a structure adaptation algorithm, *Neural Comput & Applic*, vol.18, no.2, pp.115–125, 2009.

[7] Hu, Y.C., and F.M. Tseng, Functional-link net with fuzzy integral for bankruptcy prediction, *Neurocomputing*, vol.70, pp.2959–2968, 2007.

[8] Jiang, T.Z., C. Chen, and Q. Ai, Study of decentralized nonlinear predictive control for steam valve of turbine-generator, *Proceedings of the CSEE*, vol.26, no.20, pp.22–26, 2006.

[9] Kennedy, J., and R.C. Eberhart, *Swarm Intelligence*, Morgan Kaufmann, San Mateo, CA, 2001.

[10] Keshmiri, S., and M.D. Mirmirani, Six-DOF modeling and simulation of a generic hypersonic vehicle for conceptual design studies, *Modeling and Simulation Technologies Conference and Exhibit*, pp.1–12, 2004.

[11] Kuipers, M., P. Ioannou, B. Fidan, and M. Mirmirani, Robust adaptive multiple model controller design for an airbreathing hypersonic vehicle model, *AIAA Guidance, Navigation and Control Conference and Exhibit*, pp.1–20, 2008.

[12] Lin, F.J., H.J. Shieh, P.K. Huang, and P.H. Shieh, An adaptive recurrent radial basis function network tracking controller for a two-dimensional piezo-positioning stage, *IEEE Trans. on Ultrasonics, Ferroelectrics, and Frequency Control*, vol.55, no.1, pp.183–198, 2008.

[13] Pao, Y.H., Neural-net computing and intelligent control systems, *International Journal of Control*, vol.56, no.2, pp.263–289, 1992.

[14] Polycarpou, M.M., Stable adaptive neural control scheme for nonlinear systems, *IEEE Trans. on Automat. Contr.*, vol.41, no.3, pp.447–451, 1996.

[15] Samadi, S., M.O. Ahmad, and M.N.S. Swamy, Characterization of B-spline digital filters, *IEEE Trans on Circuits and Systems—I: Regular Papers*, vol.51, no.4, pp.808–816, 2004.

[16] Tee, K.P., S.S. Ge, and F.E.H. Tay, Adaptive neural network control for helicopters in vertical flight, *IEEE Trans on Control Syst Technol,* vol.16, no.4, pp.753–762, 2008.

[17] Toh, K.A., W.Y. Yau, B.N. Chatterji, and et al., Fingerprint and speaker verification decisions fusion using functional link artificial neural networks, *IEEE Trans. on Systems, Man, and Cybernetics-Part C*, vol.35, no.3, pp.357–370, 2005.

[18] Wang, Y.H., Q.X. Wu, C.S. Jiang, and G.Y. Huang, Guaranteed cost fuzzy output feedback control via LMI method for re-entry attitude dynamics, *Journal of Uncertain Systems*, vol.1, no.4, pp.291–302, 2007.

[19] Weng, W.D., C.S. Yang, and R.C. Lin, A channel equalizer using reduced decision feedback Chebyshev functional link artificial neural networks, *Information Sciences*, vol.177, no.13, pp.2642–2654, 2007.

[20] Xu, H., M. Mirmirani, and P.A. Ioannou, Adaptive sliding mode control design for a hypersonic flight vehicle, *Journal of Guidance, Control, and Dynamics*, vol.27, no.5, pp.829–838, 2004.

[21] Yoo, S.J., J.B. Park, and Y.H. Choi, Indirect adaptive control of nonlinear dynamic systems using self recurrent wavelet neural networks via adaptive learning rates, *Information Sciences*, vol.177, no.15, pp.3074–3098, 2007.

[22] Zhao, H.Q., and J.S. Zhang, Adaptively combined FIR and functional link artificial neural network equalizer for nonlinear communication channel, *IEEE Trans. on Neural Networks*, vol.20, no.4, pp.665–674, 2009.

[23] Zhu, L., *Robust Adaptive Control for Uncertain Nonlinear Systems and Its Applications to Aerospace Vehicles*, Ph. D. Thesis, Nanjing University of Aeronautics & Astronautics, China, 2007.