Available online at www.sciencedirect.com
ScienceDirect
Fuzzy Sets and Systems 159 (2008) 2627–2649
FUZZY sets and systems
www.elsevier.com/locate/fss

Adaptive asymmetric fuzzy neural network controller design via network structuring adaptation

Chun-Fei Hsu[a,*], Ping-Zong Lin[b], Tsu-Tian Lee[c], Chi-Hsu Wang[b]

[a] *Department of Electrical Engineering, Chung Hua University, Hsinchu 300, Taiwan, Republic of China*
[b] *Department of Electrical and Control Engineering, National Chiao-Tung University, Hsinchu 300, Taiwan, Republic of China*
[c] *Department of Electrical Engineering, National Taipei University of Technology, Taipei 106, Taiwan, Republic of China*

Received 26 March 2007; received in revised form 31 January 2008; accepted 31 January 2008
Available online 21 February 2008

Abstract

This paper proposes a self-structuring fuzzy neural network (SFNN) using asymmetric Gaussian membership functions in the structure and parameter learning phases. An adaptive self-structuring asymmetric fuzzy neural-network control (ASAFNC) system which consists of an SFNN controller and a robust controller is proposed. The SFNN controller uses an SFNN with structure and parameter learning phases to online mimic an ideal controller, simultaneously. The structure learning phase consists of the growing-and-pruning algorithms of fuzzy rules to achieve an optimal network structure, and the parameter learning phase adjusts the interconnection weights of neural network to achieve favorable approximation performance. The robust controller is designed to compensate for the modeling error between the SFNN controller and the ideal controller. An online training methodology is developed in the Lyapunov sense, and thus the stability of the closed-loop control system can be guaranteed. Finally, the proposed ASAFNC system is applied to a second-order chaotic dynamics system. The simulation results show that the proposed ASAFNC can achieve favorable tracking performance.

Keywords: Fuzzy neural network; Asymmetric Gaussian membership function; Structure adaptation algorithm; Adaptive control

Contents

* Corresponding author. Tel.: +886 35186399; fax: +886 35186436.
E-mail address: fei@chu.edu.tw (C.-H. Hsu).

0165-0114/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.fss.2008.01.034

1. Introduction

Recently, the fuzzy neural network (FNN), which incorporates the advantages of fuzzy inference and neuro-learning, has been an interesting topic. The FNN possesses the merits of the low-level learning and computational power of neural network, and the high-level human knowledge representation and thinking of fuzzy theory [14,17]. Due to their learning ability, FNNs are increasingly receiving attention in solving the control problems [1,8,12,22]. Although the neuro-learning structure can tune membership functions and fuzzy rules automatically, the structure of the FNN should be determined in advance by trial-and-error. It is difficult to consider the balance between the rule number and the desired performance. As a result, if the number of fuzzy rules is chosen too large, the computation loading is heavy so that it is not suitable for practical applications. If the number of fuzzy rules is chosen too small, the control performance may be not good enough to achieve the desired performance.

To solve the problem of determining the structure in FNN approaches, much interest has been focused on the self-structuring fuzzy neural network (SFNN) approach [6,7,9,10,13]. The self-structuring approach demonstrates the properties of automatic generating rules for FNN without needing preliminary knowledge. In general, the mathematical description of the existing rules can be expressed as a set of clusters. As usually seen in other self-structuring approaches, the new membership function is generated when a new input signal is too far from the current clusters, and an existing rule is deleted when the fuzzy rule is insignificant. SFNNs also have been adopted widely for the control of complex dynamic systems due to their good generalization capability, structural adaptation, and simple computation [3,4,16,15,25,26]. Some of them use the gradient descent method to derive the parameter learning algorithms; however, they cannot guarantee the system stability [16,15]. Some of them derive the parameter learning algorithms based on the Lyapunov function to guarantee system stability; however, the structure learning algorithm is too complex [3,25,26]. Some of them proposed a simple growing-and-pruning algorithm to self-structure the FNN online with symmetric membership functions; however, this will result in slow learning speed [4].

This paper proposes an SFNN in which the learning phase considers both the structure and parameter learning phases. The structure adaptation is described as follows. A new rule is generated when a new input signal is too far from the current clusters. To avoid the unrestricted growth of membership functions and fuzzy rules, this paper uses an exponential function to calculate the significant indexes of each existing fuzzy rule. The exponential function can gradually rise or decrease the significant index values for each rule. If the fuzzy rule of SFNN is insignificant, it will be removed to reduce the computation load; and if the fuzzy rule of SFNN is significant, it will be retained. Thus, the SFNN can self-structure the fuzzy rules online to achieve an optimal network structure. Moreover, by accommodating the left-sided and right-sided spreads into a standard Gaussian membership functions, the asymmetric Gaussian membership functions can upgrade the learning capability and flexibility of a neural network [20].

Therefore, an adaptive self-structuring asymmetric fuzzy neural-network control (ASAFNC) system, which consists of an SFNN controller and a robust controller, is proposed. The SFNN controller utilizes an SFNN to mimic an ideal controller, and the robust controller is designed to compensate for the modeling error between the SFNN controller and the ideal controller. The learning phase of SFNN includes the structure learning phase and the parameter learning phase. The structure learning phase consists of the growing-and-pruning algorithms of fuzzy rules to achieve an optimal network structure, and the parameter learning phase adjusts the interconnection weights of neural network to achieve favorable approximation performance. All the parameters of ASAFNC are tuned online based on the Lyapunov stability to achieve favorable performance. Finally, the effectiveness of the proposed ASAFNC scheme is demonstrated by simulations. The simulation results show that not only favorable tracking performance can be achieved but also a concise network structure can be obtained by the proposed structure learning method.

2. Problem statement

Consider the nth-order nonlinear dynamic system of the form

$$x^{(n)} = f(\mathbf{x}) + u, \tag{1}$$

where $\mathbf{x} = [x \ \dot{x} \ \cdots \ x^{(n-1)}]^{\mathrm{T}}$, which is assumed to be available for measurement, is the state vector of the system, $f(\mathbf{x})$ is the system dynamics equation, and u is the control effort. The control objective is to find a control law so that the state trajectory x can track a command trajectory x_{c}, and thus a tracking error is defined as

$$e = x_{\mathrm{c}} - x. \tag{2}$$

If the system dynamics $f(\mathbf{x})$ in (1) is well known, there exists an ideal controller as [19]

$$u^* = -f(\mathbf{x}) + x_c^{(n)} + k_n e^{(n-1)} + \cdots + k_2 \dot{e} + k_1 e, \tag{3}$$

where $k_i, i = 1, 2, \ldots, n$ are non-zero positive constants. Substituting (3) into (1) yields

$$e^{(n)} + k_n e^{(n-1)} + \cdots + k_2 \dot{e} + k_1 e = 0. \tag{4}$$

If k_i are chosen to correspond to the coefficients of a Hurwitz polynomial whose roots lie strictly in the open left half of the complex plane, then $\lim_{t\to\infty} e = 0$ can be implied for any starting initial conditions. However, because the system dynamics $f(\mathbf{x})$ may be unknown or perturbed in practice, the ideal control law u^* in (3) cannot be implemented easily. To solve the problem of the model-based control approach for real-time implementation, adaptive fuzzy neural-network control (AFNC) techniques have been developed to control these kinds of unknown nonlinear dynamic systems [1, 8, 12, 22]. These techniques use an FNN to estimate the plant or controller parameters online. If the FNN is used to estimate the model of the plant, it is called an indirect AFNC, and if the FNN is used to estimate the controller of the plant, it is called a direct AFNC [23].

3. ASAFNC design

Based on the direct AFNC concept, this paper proposes an ASAFNC system as shown in Fig. 1.The ASAFNC system is composed of an SFNN controller and a robust controller as

$$u_{\mathrm{ac}} = u_{\mathrm{sfnn}} + u_{\mathrm{rb}}, \tag{5}$$

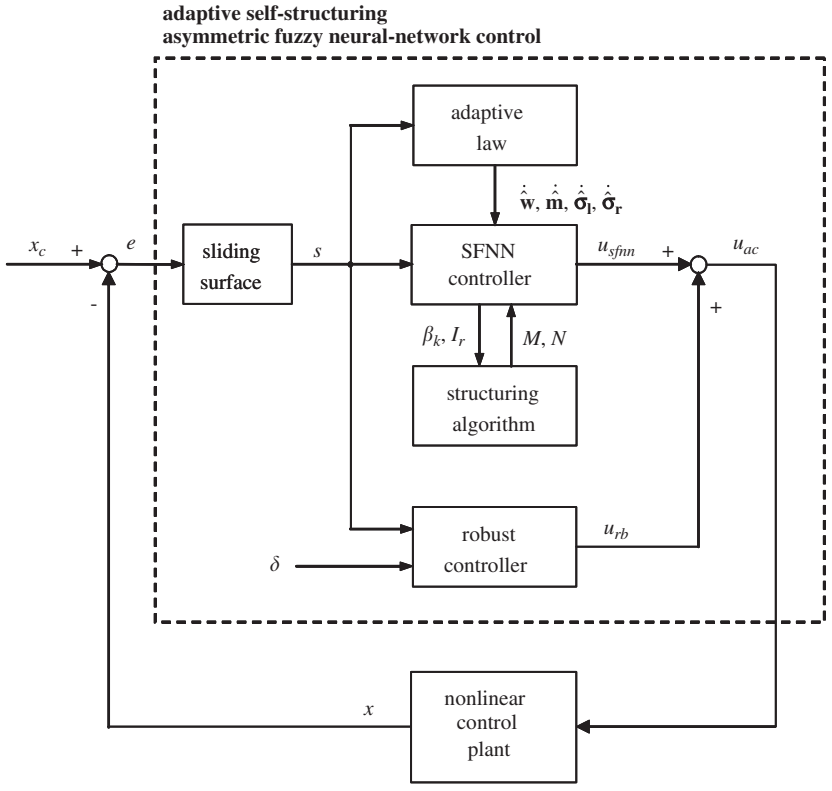

Fig. 1. The block diagram of ASAFNC system.

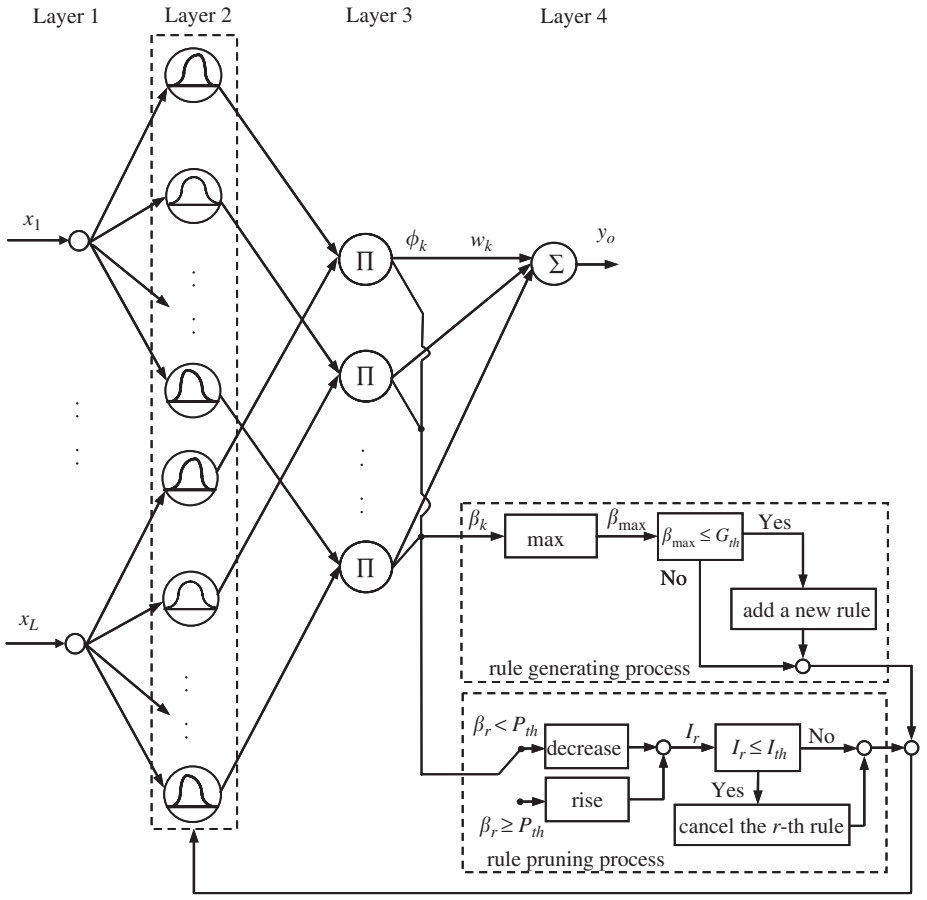

Fig. 2. The structure of SFNN.

where a sliding surface is defined as

$$s = e^{(n-1)} + k_n e^{(n-2)} + \cdots + k_2 e + k_1 \int_0^t e \, \mathrm{d}\tau. \tag{6}$$

The SFNN controller u_{sfnn} utilizes the SFNN with asymmetric Gaussian membership functions to mimic the ideal controller in (3), and the robust controller u_{rb} is designed to compensate for the modeling error between the SFNN controller u_{sfnn} and the ideal controller u^*.

3.1. Description of SFNN

Fig. 2 shows the configuration of the proposed SFNN which is composed of the input, the membership, the rule, and the output layers. Layer 1 accepts the input variables. Nodes at layer 2 are term nodes which act as membership functions to represent the terms of the respective linguistic variables. The asymmetric Gaussian membership function constituted by a center, a left-side variance, and a right-side variance is considered. Nodes of layer 3 are regarded as fuzzy rules. The links before layer 3 represent the preconditions of rules and the links after layer 3 represent the consequences. Layer 4 is the output layer, where the node in this layer is the output of the neural network. The output of the SFNN with N existing fuzzy rules is given as

$$y_{\mathrm{o}} = \sum_{k=1}^{N} w_k \phi_k(\mathbf{x}) \tag{7}$$

in which w_k is the output action strength associated with the kth rule and ϕ_k is the response of the firing weight for an input vector $\mathbf{x} = [x_1 \ x_2 \ \cdots \ x_L]^{\mathrm{T}}$ and composed of membership function defined as [20]

$$\zeta_{ij} = \begin{cases} \exp\left(-\dfrac{(x_i - m_{ij})^2}{(\sigma_{ij}^{\mathrm{l}})^2}\right) & \text{if } -\infty < x_i \leqslant m_{ij}, \\ \exp\left(-\dfrac{(x_i - m_{ij})^2}{(\sigma_{ij}^{\mathrm{r}})^2}\right) & \text{if } m_{ij} \leqslant x_i < \infty, \end{cases} \qquad j = 1, 2, \ldots, M, \tag{8}$$

where M is the total number of membership functions with respect to the respective input node; m_{ij} , σ_{ij}^{l}, and σ_{ij}^{r} are the mean, left-side variance, and right-side variance of the asymmetric Gaussian function in the jth term of the ith input linguistic variable x_i, respectively. However, σ_{ij}^{l} and σ_{ij}^{r} may become zero in the training procedure, the membership function ζ_{ij} will not be defined. To avoid this problem, this paper considers a membership function form as [23]

$$\zeta_{ij} = \begin{cases} \exp\left(-\dfrac{(x_i - m_{ij})^2}{(\sigma_{ij}^{\mathrm{l}})^2 + \varpi}\right) & \text{if } -\infty < x_i \leqslant m_{ij}, \\ \exp\left(-\dfrac{(x_i - m_{ij})^2}{(\sigma_{ij}^{\mathrm{r}})^2 + \varpi}\right) & \text{if } m_{ij} \leqslant x_i < \infty, \end{cases} \qquad j = 1, 2, \ldots, M, \tag{9}$$

where ϖ is a small positive constant**.** Then, the associated firing strength can be defined as

$$\phi_k = \prod_{j=1}^{M} \zeta_{jk}. \tag{10}$$

To note easily, define vectors $\mathbf{m}$, $\boldsymbol{\sigma}_{\mathbf{l}}$, and $\boldsymbol{\sigma}_{\mathbf{r}}$ collecting all parameters of SFNN as

$$\mathbf{m} = [m_{11} \ \cdots \ m_{L1} \ m_{12} \ \cdots \ m_{L2} \ \cdots\cdots \ m_{1M} \ \cdots \ m_{LM}]^{\mathrm{T}}, \tag{11}$$

$$\boldsymbol{\sigma}_{\mathbf{l}} = [\sigma_{11}^{\mathrm{l}} \ \cdots \ \sigma_{L1}^{\mathrm{l}} \ \sigma_{12}^{\mathrm{l}} \ \cdots \ \sigma_{L2}^{\mathrm{l}} \ \cdots\cdots \ \sigma_{1M}^{\mathrm{l}} \ \cdots \ \sigma_{LM}^{\mathrm{l}}]^{\mathrm{T}}, \tag{12}$$

$$\boldsymbol{\sigma}_{\mathbf{r}} = [\sigma_{11}^{\mathrm{r}} \ \cdots \ \sigma_{L1}^{\mathrm{r}} \ \sigma_{12}^{\mathrm{r}} \ \cdots \ \sigma_{L2}^{\mathrm{r}} \ \cdots\cdots \ \sigma_{1M}^{\mathrm{r}} \ \cdots \ \sigma_{LM}^{\mathrm{r}}]^{\mathrm{T}}. \tag{13}$$

Thus, the output of the SFNN can be represented in a vector form as

$$y_{\mathrm{o}} = \mathbf{w}^{\mathrm{T}} \boldsymbol{\varphi}(\mathbf{x}, \mathbf{m}, \boldsymbol{\sigma}_{\mathbf{l}}, \boldsymbol{\sigma}_{\mathbf{r}}), \tag{14}$$

where $\mathbf{w} = [w_1 \ w_2 \ \cdots \ w_N]^{\mathrm{T}}$ and $\boldsymbol{\varphi} = [\phi_1 \ \phi_2 \ \cdots \ \phi_N]^{\mathrm{T}}$. For the FNN approaches, the structure of the FNN should be determined in advance by trial-and-error. However, it is difficult to consider the balance between the rule number and the desired performance. Therefore, the structure adaptation algorithm which contains the growing and pruning of membership functions and fuzzy rules is proposed in this paper. The descriptions are given as follows.

In the structure growing process, the mathematical description of the existing rules can be expressed as a set of clusters. For constructing the initial fuzzy rules of the SFNN, the fuzzy clustering method is used to partition a set of data into a number of overlapping clusters based on the distance in a metric space between the data points and the cluster prototypes. Each cluster in the product space of the input–output data represents a rule. The firing strength of a rule for each incoming data x_i can be represented as the degree that the incoming data belong to the cluster [13]. If the value of firing strength is too small, it indicates that the input value is on the edge of range of the existing membership functions. Under this situation, the output will cause unsatisfactory performance. Therefore, a new membership function and a new fuzzy rule should be generated to improve the performance.

The firing strength from (10) is used as the degree measure

$$\beta_k = \phi_k, \quad k = 1, 2, \ldots, N(t), \tag{15}$$

where $N(t)$ is the number of the existing fuzzy rules at the time t. Define the maximum degree $\beta_{\max}$ as

$$\beta_{\max} = \max_{1 \leqslant k \leqslant N(t)} \beta_k. \tag{16}$$

If $\beta_{\max} \leqslant G_{\text{th}}$ is satisfied, where $G_{\text{th}} \in (0, 1)$ is a pre-given threshold, the incoming data are far from the edge of range of the existing membership functions. Hence, a new membership function is generated. The mean and the standard deviation of the new membership function and the weight are selected as follows:

$$m_i^{\text{new}} = x_i, \tag{17}$$

$$\sigma_i^{\text{l,new}} = \sigma_i, \tag{18}$$

$$\sigma_i^{\text{r,new}} = \sigma_i, \tag{19}$$

$$w_{\text{new}} = 0, \tag{20}$$

where x_i is the new incoming data and σ_i is a pre-specified constant.

To avoid the unrestricted growth of network structure and an overload computation, the pruning algorithm is developed to eliminate irrelevant fuzzy rules. In Ref. [4], a significance index is determined for the importance of the fuzzy rules. The elimination algorithm is derived from the observation that if the significance index gets fading when the rule firing weight is smaller than threshold value and if the significance index gets fixation when the rule firing weight is larger than threshold value [4]. In this paper, when the rth firing strength β_r is smaller than the threshold value P_{th}, it indicates that the relationship becomes weak between the input and the rth rule. Then, the significant index of rth fuzzy rules will be decreased. When the rth firing strength β_r is larger than the threshold value P_{th}, it indicates that the incoming inputs fall into the range of the rth fuzzy rule. Thus, the significant index of rth fuzzy rules should be raised. The significance index is determined for the rth rules can be given as

$$I_r(t+1) = \begin{cases} I_r(t) \cdot \exp(-\tau_1) & \text{if } \beta_r < P_{\text{th}}, \\ I_r(t) \cdot [2 - \exp(-\tau_2(1 - I_r(t)))] & \text{if } \beta_r \geqslant P_{\text{th}}, \end{cases} \quad r = 1, 2, \ldots, N(t), \tag{21}$$

where I_r is the significant index of the rth rule and its initial value is 1, P_{th} is the pruning threshold value, and τ_1 and τ_2 are the designed constant. Exponential functions in (21) are used to rise or decrease the values of significant index in [0, 1]. If $I_r \leqslant I_{\text{th}}$ is satisfied, where I_{th} is another pre-given threshold, the$r$th fuzzy rule will be deleted. For real-time implementation, if the computation load is the important issue, a large P_{th} should be chosen, so that more fuzzy rules can be pruned. This operation will prevent the fuzzy rule, which may be less used but still significant, from being deleted in the training process. Hence, the computation load would be reduced.

In summary, the flow chart of the structure learning algorithm is shown in Fig. 3. The major contributions of the SFNN are: (1) SFNN can be operated directly without spending much time pre-determining membership functions and fuzzy rules and (2) the computation load can be reduced simultaneously.

3.2. Approximation of SFNN

An optimal SFNN controller can be designed to approximate the ideal controller (3) even under the structural change of neural network, such that [18,20]

$$u^* = u^*_{\text{sfnn}} + \Delta = \mathbf{w}^{*\text{T}}\boldsymbol{\varphi}(\mathbf{x}, \mathbf{m}^*, \boldsymbol{\sigma}_\mathbf{l}^*, \boldsymbol{\sigma}_\mathbf{r}^*) + \Delta = \mathbf{w}^{*\text{T}}\boldsymbol{\varphi}^* + \Delta, \tag{22}$$

where $\boldsymbol{\varphi}^* = \boldsymbol{\varphi}(\mathbf{x}, \mathbf{m}^*, \boldsymbol{\sigma}_\mathbf{l}^*, \boldsymbol{\sigma}_\mathbf{r}^*)$, Δ denotes the approximation error, and $\mathbf{w}^*$, $\mathbf{m}^*$, $\boldsymbol{\sigma}_\mathbf{l}^*$, and $\boldsymbol{\sigma}_\mathbf{r}^*$ are the optimal vectors. In fact, the optimal vectors that best approximate a given nonlinear function are difficult to determine. Thus, an estimated SFNN controller will be introduced to mimic the ideal controller as

$$u_{\text{sfnn}} = \hat{\mathbf{w}}^{\text{T}}\boldsymbol{\varphi}(\mathbf{x}, \hat{\mathbf{m}}, \hat{\boldsymbol{\sigma}}_\mathbf{l}, \hat{\boldsymbol{\sigma}}_\mathbf{r}) = \hat{\mathbf{w}}^{\text{T}}\hat{\boldsymbol{\varphi}}, \tag{23}$$

where $\hat{\boldsymbol{\varphi}} = \boldsymbol{\varphi}(\mathbf{x}, \hat{\mathbf{m}}, \hat{\boldsymbol{\sigma}}_\mathbf{l}, \hat{\boldsymbol{\sigma}}_\mathbf{r})$ and $\hat{\mathbf{w}}$, $\hat{\mathbf{m}}$, $\hat{\boldsymbol{\sigma}}_\mathbf{l}$, and $\hat{\boldsymbol{\sigma}}_\mathbf{r}$ are the estimated vectors of $\mathbf{w}$, $\mathbf{m}$, $\boldsymbol{\sigma}_\mathbf{l}$, and $\boldsymbol{\sigma}_\mathbf{r}$, respectively. Moreover, the optimal vectors can be further defined as [20]

$$(\mathbf{w}^*, \mathbf{m}^*, \boldsymbol{\sigma}_\mathbf{l}^*, \boldsymbol{\sigma}_\mathbf{r}^*) = \underset{\hat{\mathbf{w}} \in \boldsymbol{\Omega}_\mathbf{w}, \hat{\mathbf{m}} \in \boldsymbol{\Omega}_\mathbf{m}, \hat{\boldsymbol{\sigma}}_\mathbf{l} \in \boldsymbol{\Omega}_{\boldsymbol{\sigma}_\mathbf{l}}, \hat{\boldsymbol{\sigma}}_\mathbf{r} \in \boldsymbol{\Omega}_{\boldsymbol{\sigma}_\mathbf{r}}}{\arg\min} \left[\sup_{\mathbf{x} \in \boldsymbol{\Omega}_\mathbf{x} \times R} \left| u^*_{\text{sfnn}}(\mathbf{x}) - u_{\text{sfnn}}(\mathbf{x}, \hat{\mathbf{m}}, \hat{\boldsymbol{\sigma}}_\mathbf{l}, \hat{\boldsymbol{\sigma}}_\mathbf{r}) \right| \right], \tag{24}$$

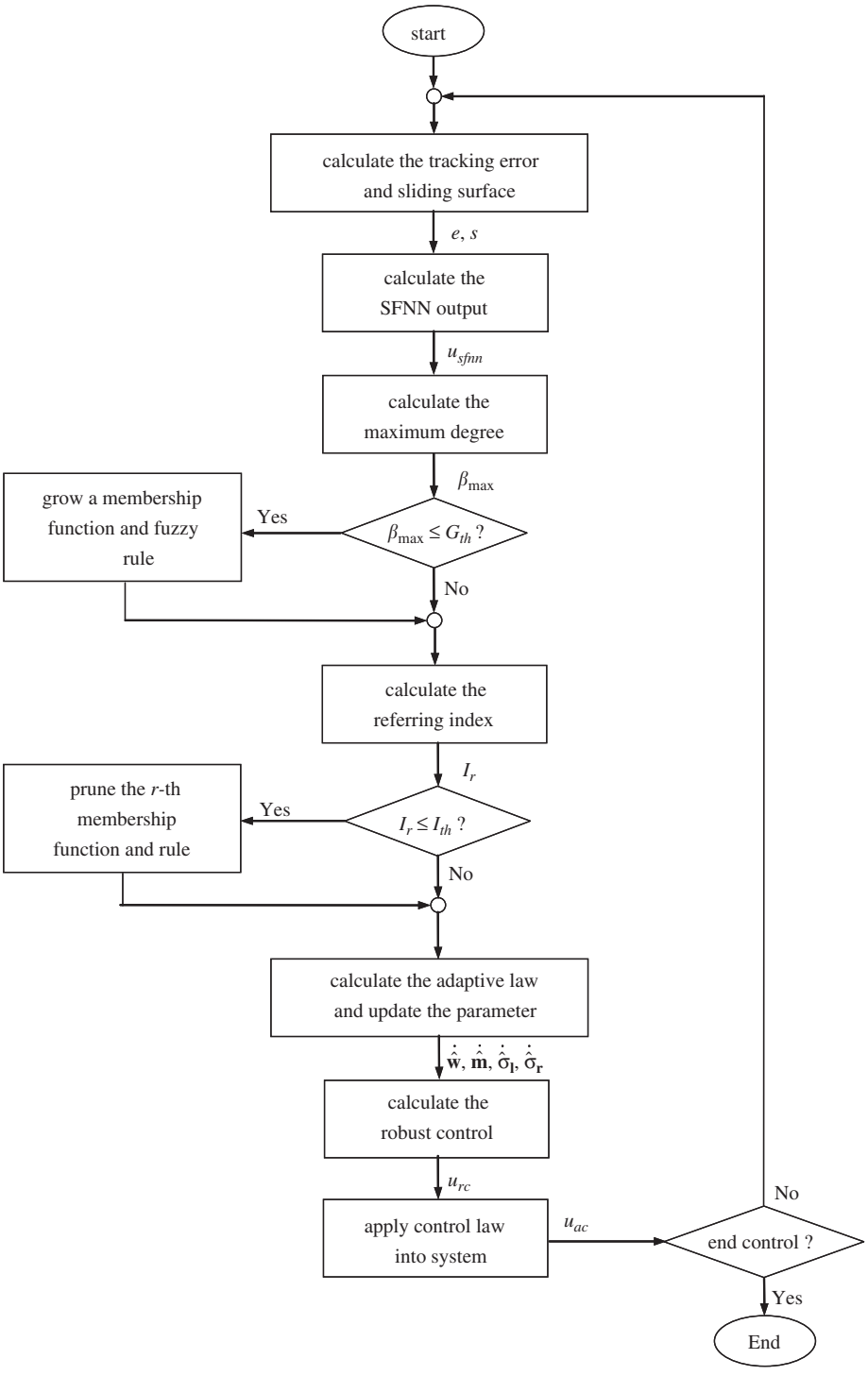

Fig. 3. The flow chart of the ASAFNC system.

where

$$\mathbf{\Omega_w} = \{\hat{\mathbf{w}} : \|\hat{\mathbf{w}}\| \leqslant D_{\mathbf{w}}\}, \tag{25}$$

$$\mathbf{\Omega_m} = \{\hat{\mathbf{m}} : \|\hat{\mathbf{m}}\| \leqslant D_{\mathbf{m}}\}, \tag{26}$$

$$\mathbf{\Omega}_{\boldsymbol{\sigma}_\mathbf{l}} = \{\hat{\boldsymbol{\sigma}}_\mathbf{l} : \|\hat{\boldsymbol{\sigma}}_\mathbf{l}\| \leqslant D_{\boldsymbol{\sigma}_\mathbf{l}}\}, \tag{27}$$

$$\mathbf{\Omega}_{\boldsymbol{\sigma}_\mathbf{r}} = \{\hat{\boldsymbol{\sigma}}_\mathbf{r} : \|\hat{\boldsymbol{\sigma}}_\mathbf{r}\| \leqslant D_{\boldsymbol{\sigma}_\mathbf{r}}\}, \tag{28}$$

where $D_\mathbf{w}$, $D_\mathbf{m}$, $D_{\boldsymbol{\sigma}_\mathbf{l}}$, and $D_{\boldsymbol{\sigma}_\mathbf{r}}$ are positive constants specified by designers. There exists Δ^* which is a finite positive constant such that the inequality $\|\Delta\| \leqslant \Delta^*$ can be held. Define a modeling error, $\tilde{u}$, as

$$\tilde{u} = u^* - u_{\text{sfnn}} = \tilde{\mathbf{w}}^\mathrm{T}\tilde{\boldsymbol{\varphi}} + \hat{\mathbf{w}}^\mathrm{T}\tilde{\boldsymbol{\varphi}} + \tilde{\mathbf{w}}^\mathrm{T}\hat{\boldsymbol{\varphi}} + \Delta, \tag{29}$$

where $\tilde{\mathbf{w}} = \mathbf{w}^* - \hat{\mathbf{w}}$ and $\tilde{\boldsymbol{\varphi}} = \boldsymbol{\varphi}^* - \hat{\boldsymbol{\varphi}}$. In the following, the linearization technique is employed to transform the nonlinear fuzzy function into a partially linear form so that the expansion $\tilde{\boldsymbol{\varphi}}$ can be expressed as [5]

$$\begin{aligned}\tilde{\boldsymbol{\varphi}} &= \begin{bmatrix} \tilde{\phi}_1 \\ \tilde{\phi}_2 \\ \vdots \\ \tilde{\phi}_N \end{bmatrix} = \left.\begin{bmatrix} \dfrac{\partial\phi_1}{\partial\mathbf{m}} \\ \dfrac{\partial\phi_2}{\partial\mathbf{m}} \\ \vdots \\ \dfrac{\partial\phi_N}{\partial\mathbf{m}} \end{bmatrix}\right|_{\mathbf{m}=\hat{\mathbf{m}}} (\mathbf{m}^* - \hat{\mathbf{m}}) + \left.\begin{bmatrix} \dfrac{\partial\phi_1}{\partial\boldsymbol{\sigma}_\mathbf{l}} \\ \dfrac{\partial\phi_2}{\partial\boldsymbol{\sigma}_\mathbf{l}} \\ \vdots \\ \dfrac{\partial\phi_N}{\partial\boldsymbol{\sigma}_\mathbf{l}} \end{bmatrix}\right|_{\boldsymbol{\sigma}_\mathbf{l}=\hat{\boldsymbol{\sigma}}_\mathbf{l}} (\boldsymbol{\sigma}_\mathbf{l}^* - \hat{\boldsymbol{\sigma}}_\mathbf{l}) + \left.\begin{bmatrix} \dfrac{\partial\phi_1}{\partial\boldsymbol{\sigma}_\mathbf{r}} \\ \dfrac{\partial\phi_2}{\partial\boldsymbol{\sigma}_\mathbf{r}} \\ \vdots \\ \dfrac{\partial\phi_N}{\partial\boldsymbol{\sigma}_\mathbf{r}} \end{bmatrix}\right|_{\boldsymbol{\sigma}_\mathbf{r}=\hat{\boldsymbol{\sigma}}_\mathbf{r}} (\boldsymbol{\sigma}_\mathbf{r}^* - \hat{\boldsymbol{\sigma}}_\mathbf{r}) + \mathbf{h} \\ &= \boldsymbol{\varphi}_\mathbf{m}^\mathrm{T}\tilde{\mathbf{m}} + \boldsymbol{\varphi}_{\boldsymbol{\sigma}_\mathbf{l}}^\mathrm{T}\tilde{\boldsymbol{\sigma}}_\mathbf{l} + \boldsymbol{\varphi}_{\boldsymbol{\sigma}_\mathbf{r}}^\mathrm{T}\tilde{\boldsymbol{\sigma}}_\mathbf{r} + \mathbf{h},\end{aligned} \tag{30}$$

where $\mathbf{h}$ is a vector of higher-order terms, $\tilde{\mathbf{m}} = \mathbf{m}^* - \hat{\mathbf{m}}$, $\tilde{\boldsymbol{\sigma}}_\mathbf{l} = \boldsymbol{\sigma}_\mathbf{l}^* - \hat{\boldsymbol{\sigma}}_\mathbf{l}$, and $\tilde{\boldsymbol{\sigma}}_\mathbf{r} = \boldsymbol{\sigma}_\mathbf{r}^* - \hat{\boldsymbol{\sigma}}_\mathbf{r}$. Substituting (30) into (29), (29) can be rewritten as

$$\begin{aligned}\tilde{u} &= \tilde{\mathbf{w}}^\mathrm{T}\tilde{\boldsymbol{\varphi}} + \hat{\mathbf{w}}^\mathrm{T}(\boldsymbol{\varphi}_\mathbf{m}^\mathrm{T}\tilde{\mathbf{m}} + \boldsymbol{\varphi}_{\boldsymbol{\sigma}_\mathbf{l}}^\mathrm{T}\tilde{\boldsymbol{\sigma}}_\mathbf{l} + \boldsymbol{\varphi}_{\boldsymbol{\sigma}_\mathbf{r}}^\mathrm{T}\tilde{\boldsymbol{\sigma}}_\mathbf{r} + \mathbf{h}) + \tilde{\mathbf{w}}^\mathrm{T}\hat{\boldsymbol{\varphi}} + \Delta \\ &= \tilde{\mathbf{w}}^\mathrm{T}\hat{\boldsymbol{\varphi}} + \tilde{\mathbf{m}}^\mathrm{T}\boldsymbol{\varphi}_\mathbf{m}\hat{\mathbf{w}} + \tilde{\boldsymbol{\sigma}}_\mathbf{l}^\mathrm{T}\boldsymbol{\varphi}_{\boldsymbol{\sigma}_\mathbf{l}}\hat{\mathbf{w}} + \tilde{\boldsymbol{\sigma}}_\mathbf{r}^\mathrm{T}\boldsymbol{\varphi}_{\boldsymbol{\sigma}_\mathbf{r}}\hat{\mathbf{w}} + \varepsilon,\end{aligned} \tag{31}$$

where $\tilde{\mathbf{m}}^\mathrm{T}\boldsymbol{\varphi}_\mathbf{m}\hat{\mathbf{w}} = \hat{\mathbf{w}}^\mathrm{T}\boldsymbol{\varphi}_\mathbf{m}^\mathrm{T}\tilde{\mathbf{m}}$, $\tilde{\boldsymbol{\sigma}}_\mathbf{l}^\mathrm{T}\boldsymbol{\varphi}_{\boldsymbol{\sigma}_\mathbf{l}}\hat{\mathbf{w}} = \hat{\mathbf{w}}^\mathrm{T}\boldsymbol{\varphi}_{\boldsymbol{\sigma}_\mathbf{l}}^\mathrm{T}\tilde{\boldsymbol{\sigma}}_\mathbf{l}$, $\tilde{\boldsymbol{\sigma}}_\mathbf{r}^\mathrm{T}\boldsymbol{\varphi}_{\boldsymbol{\sigma}_\mathbf{r}}\hat{\mathbf{w}} = \hat{\mathbf{w}}^\mathrm{T}\boldsymbol{\varphi}_{\boldsymbol{\sigma}_\mathbf{r}}^\mathrm{T}\tilde{\boldsymbol{\sigma}}_\mathbf{r}$, and the uncertain term $\varepsilon = \hat{\mathbf{w}}^\mathrm{T}\mathbf{h} + \tilde{\mathbf{w}}^\mathrm{T}\tilde{\boldsymbol{\varphi}} + \Delta$. The higher-order term $\mathbf{h}$ satisfies

$$\begin{aligned}\|\mathbf{h}\| &= \|\tilde{\boldsymbol{\varphi}} - \boldsymbol{\varphi}_\mathbf{m}^\mathrm{T}\tilde{\mathbf{m}} - \boldsymbol{\varphi}_{\boldsymbol{\sigma}_\mathbf{l}}^\mathrm{T}\tilde{\boldsymbol{\sigma}}_\mathbf{l} - \boldsymbol{\varphi}_{\boldsymbol{\sigma}_\mathbf{r}}^\mathrm{T}\tilde{\boldsymbol{\sigma}}_\mathbf{r}\| \\ &\leqslant \|\tilde{\boldsymbol{\varphi}}\| + \|\boldsymbol{\varphi}_\mathbf{m}^\mathrm{T}\|\|\tilde{\mathbf{m}}\| + \|\boldsymbol{\varphi}_{\boldsymbol{\sigma}_\mathbf{l}}^\mathrm{T}\|\|\tilde{\boldsymbol{\sigma}}_\mathbf{l}\| + \|\boldsymbol{\varphi}_{\boldsymbol{\sigma}_\mathbf{r}}^\mathrm{T}\|\|\tilde{\boldsymbol{\sigma}}_\mathbf{r}\| \\ &\leqslant \mathbf{c}_0 + \mathbf{c}_1\|\tilde{\mathbf{m}}\| + \mathbf{c}_2\|\tilde{\boldsymbol{\sigma}}_\mathbf{l}\| + \mathbf{c}_3\|\tilde{\boldsymbol{\sigma}}_\mathbf{r}\|,\end{aligned} \tag{32}$$

where $\mathbf{c}_0$, $\mathbf{c}_1$, $\mathbf{c}_2$, and $\mathbf{c}_3$ are bounded positive constants satisfying $\|\tilde{\boldsymbol{\varphi}}\| \leqslant c_0$, $\|\boldsymbol{\varphi}_\mathbf{m}^\mathrm{T}\| \leqslant c_1$, $\|\boldsymbol{\varphi}_{\boldsymbol{\sigma}_\mathbf{l}}^\mathrm{T}\| \leqslant c_2$, $\|\boldsymbol{\varphi}_{\boldsymbol{\sigma}_\mathbf{r}}^\mathrm{T}\| \leqslant c_3$. The existence of $\mathbf{c}_0$, $\mathbf{c}_1$, $\mathbf{c}_2$, and $\mathbf{c}_3$ is assured due to the fact that Gaussian function and its derivative are always bounded by constants. Moreover, $\tilde{\mathbf{w}}$, $\tilde{\mathbf{m}}$, $\tilde{\boldsymbol{\sigma}}_\mathbf{l}$, and $\tilde{\boldsymbol{\sigma}}_\mathbf{r}$ satisfy

$$\|\tilde{\mathbf{w}}\| = \|\mathbf{w}^* - \hat{\mathbf{w}}\| \leqslant \|\mathbf{w}^*\| + \|\hat{\mathbf{w}}\| \leqslant D_\mathbf{w} + \|\hat{\mathbf{w}}\|, \tag{33}$$

$$\|\tilde{\mathbf{m}}\| = \|\mathbf{m}^* - \hat{\mathbf{m}}\| \leqslant \|\mathbf{m}^*\| + \|\hat{\mathbf{m}}\| \leqslant D_\mathbf{m} + \|\hat{\mathbf{m}}\|, \tag{34}$$

$$\|\tilde{\boldsymbol{\sigma}}_\mathbf{l}\| = \|\boldsymbol{\sigma}_\mathbf{l}^* - \hat{\boldsymbol{\sigma}}_\mathbf{l}\| \leqslant \|\boldsymbol{\sigma}_\mathbf{l}^*\| + \|\hat{\boldsymbol{\sigma}}_\mathbf{l}\| \leqslant D_{\boldsymbol{\sigma}_\mathbf{l}} + \|\hat{\boldsymbol{\sigma}}_\mathbf{l}\|, \tag{35}$$

$$\|\tilde{\boldsymbol{\sigma}}_\mathbf{r}\| = \|\boldsymbol{\sigma}_\mathbf{r}^* - \hat{\boldsymbol{\sigma}}_\mathbf{r}\| \leqslant \|\boldsymbol{\sigma}_\mathbf{r}^*\| + \|\hat{\boldsymbol{\sigma}}_\mathbf{r}\| \leqslant D_{\boldsymbol{\sigma}_\mathbf{r}} + \|\hat{\boldsymbol{\sigma}}_\mathbf{r}\|. \tag{36}$$

Next, the uncertain term ε is satisfied

$$\begin{aligned}|\varepsilon| &= \|\tilde{\mathbf{w}}^\mathrm{T}(\boldsymbol{\varphi}_\mathbf{m}^\mathrm{T}\tilde{\mathbf{m}} + \boldsymbol{\varphi}_{\boldsymbol{\sigma}_\mathbf{l}}^\mathrm{T}\tilde{\boldsymbol{\sigma}}_\mathbf{l} + \boldsymbol{\varphi}_{\boldsymbol{\sigma}_\mathbf{r}}^\mathrm{T}\tilde{\boldsymbol{\sigma}}_\mathbf{r} + \mathbf{h}) + \hat{\mathbf{w}}^\mathrm{T}\mathbf{h} + \Delta\| \\ &= \|\tilde{\mathbf{w}}^\mathrm{T}\boldsymbol{\varphi}_\mathbf{m}^\mathrm{T}\tilde{\mathbf{m}} + \tilde{\mathbf{w}}^\mathrm{T}\boldsymbol{\varphi}_{\boldsymbol{\sigma}_\mathbf{l}}^\mathrm{T}\tilde{\boldsymbol{\sigma}}_\mathbf{l} + \tilde{\mathbf{w}}^\mathrm{T}\boldsymbol{\varphi}_{\boldsymbol{\sigma}_\mathbf{r}}^\mathrm{T}\tilde{\boldsymbol{\sigma}}_\mathbf{r} + \mathbf{w}^{*T}\mathbf{h} + \Delta\| \\ &\leqslant \mathbf{c}_1(D_\mathbf{w} + \|\hat{\mathbf{w}}\|)(D_\mathbf{m} + \|\hat{\mathbf{m}}\|) + \mathbf{c}_2(D_\mathbf{w} + \|\hat{\mathbf{w}}\|)(D_{\boldsymbol{\sigma}_\mathbf{l}} + \|\hat{\boldsymbol{\sigma}}_\mathbf{l}\|) + \mathbf{c}_3(D_\mathbf{w} + \|\hat{\mathbf{w}}\|)(D_{\boldsymbol{\sigma}_\mathbf{r}} + \|\hat{\boldsymbol{\sigma}}_\mathbf{r}\|) \\ &\quad + D_\mathbf{w}[\mathbf{c}_0 + \mathbf{c}_1(D_\mathbf{m} + \|\hat{\mathbf{m}}\|) + \mathbf{c}_2(D_{\boldsymbol{\sigma}_\mathbf{l}} + \|\hat{\boldsymbol{\sigma}}_\mathbf{l}\|) + \mathbf{c}_3(D_{\boldsymbol{\sigma}_\mathbf{r}} + \|\hat{\boldsymbol{\sigma}}_\mathbf{r}\|)] + \Delta^* \\ &= [\Theta_1, \Theta_2, \Theta_3, \Theta_4, \Theta_5, \Theta_6, \Theta_7, \Theta_8][1, \|\hat{\mathbf{w}}\|, \|\hat{\mathbf{m}}\|, \|\hat{\boldsymbol{\sigma}}_\mathbf{l}\|, \|\hat{\boldsymbol{\sigma}}_\mathbf{r}\|, \|\hat{\mathbf{m}}\|\|\hat{\mathbf{w}}\|, \|\hat{\boldsymbol{\sigma}}_\mathbf{l}\|\|\hat{\mathbf{w}}\|, \|\hat{\boldsymbol{\sigma}}_\mathbf{r}\|\|\hat{\mathbf{w}}\|]^\mathrm{T}, \\ &= \mathbf{\Theta}^\mathrm{T}\mathbf{\Gamma},\end{aligned} \tag{37}$$

where $\mathbf{\Theta} = [\Theta_1, \Theta_2, \Theta_3, \Theta_4, \Theta_5, \Theta_6, \Theta_7, \Theta_8]^\mathrm{T}$, $\Theta_1 = (\mathbf{c}_0 + 2\mathbf{c}_1 D_\mathbf{m} + 2\mathbf{c}_2 D_{\boldsymbol{\sigma}_\mathbf{l}} + 2\mathbf{c}_3 D_{\boldsymbol{\sigma}_\mathbf{r}})D_\mathbf{w} + \Delta^*$, $\Theta_2 = \mathbf{c}_1 D_\mathbf{m} + \mathbf{c}_2 D_{\boldsymbol{\sigma}_\mathbf{l}} + \mathbf{c}_3 D_{\boldsymbol{\sigma}_\mathbf{r}}$, $\Theta_3 = 2\mathbf{c}_1 D_\mathbf{w}$, $\Theta_4 = 2\mathbf{c}_2 D_\mathbf{w}$, $\Theta_5 = 2\mathbf{c}_3 D_\mathbf{w}$, $\Theta_6 = \mathbf{c}_1$, $\Theta_7 = \mathbf{c}_2$, $\Theta_8 = \mathbf{c}_3$ and

$\mathbf{\Gamma} = [1, \|\hat{\mathbf{w}}\|, \|\hat{\mathbf{m}}\|, \|\hat{\boldsymbol{\sigma}}_\mathbf{l}\|, \|\hat{\boldsymbol{\sigma}}_\mathbf{r}\|, \|\hat{\mathbf{m}}\|\|\hat{\mathbf{w}}\|, \|\hat{\boldsymbol{\sigma}}_\mathbf{l}\|\|\hat{\mathbf{w}}\|, \|\hat{\boldsymbol{\sigma}}_\mathbf{r}\|\|\hat{\mathbf{w}}\|]^\mathrm{T}$. Since $\mathbf{\Theta}$ is a bounded vector, if $\mathbf{\Gamma}$ can be guaranteed to be bounded, the uncertain term ε is thus bounded. The analysis of boundness of $\mathbf{\Gamma}$ will be given in the next subsection.

3.3. ASAFNC design

Substituting (5) into (1) and using (3) and (6), yields

$$\dot{s} = u^* - u_{\mathrm{sfnn}} - u_{\mathrm{rb}}. \tag{38}$$

By using (31), (38) can be rewritten as

$$\dot{s} = \tilde{\mathbf{w}}^\mathrm{T}\hat{\boldsymbol{\varphi}} + \tilde{\mathbf{m}}^\mathrm{T}\boldsymbol{\varphi}_\mathbf{m}\hat{\mathbf{w}} + \tilde{\boldsymbol{\sigma}}_\mathbf{l}{}^\mathrm{T}\boldsymbol{\varphi}_{\boldsymbol{\sigma}_\mathbf{l}}\hat{\mathbf{w}} + \tilde{\boldsymbol{\sigma}}_\mathbf{r}^\mathrm{T}\boldsymbol{\varphi}_{\boldsymbol{\sigma}_\mathbf{r}}\hat{\mathbf{w}} + \varepsilon - u_{\mathrm{rb}}. \tag{39}$$

If ε exists, consider a specified L_2 tracking performance [5,24]

$$\int_0^T s^2(t)\,\mathrm{d}t \leqslant s^2(0) + \delta^2 \int_0^T \varepsilon^2(t)\,\mathrm{d}t + \frac{1}{\eta_w}\tilde{\mathbf{w}}^\mathrm{T}(0)\tilde{\mathbf{w}}(0) + \frac{1}{\eta_m}\tilde{\mathbf{m}}^\mathrm{T}(0)\tilde{\mathbf{m}}(0) + \frac{1}{\eta_{\sigma_\mathrm{l}}}\tilde{\boldsymbol{\sigma}}_\mathbf{l}^\mathrm{T}(0)\tilde{\boldsymbol{\sigma}}_\mathbf{l}(0) + \frac{1}{\eta_{\sigma_\mathrm{r}}}\tilde{\boldsymbol{\sigma}}_\mathbf{r}^\mathrm{T}(0)\tilde{\boldsymbol{\sigma}}_\mathbf{r}(0), \tag{40}$$

where η_w, η_m, η_{σ_l}, and η_{σ_r} are the positive-constant learning rates, and δ is a prescribed attenuation constant. If the system starts with initial conditions $s(0) = 0$, $\tilde{\mathbf{w}}(0) = 0$, $\tilde{\mathbf{m}}(0) = 0$, $\tilde{\boldsymbol{\sigma}}_\mathbf{l}(0) = 0$ and $\tilde{\boldsymbol{\sigma}}_\mathbf{r}(0) = 0$, the L_2 tracking performance in (40) can be rewritten as

$$\sup_{\varepsilon \in L_2[0,T]} \frac{\|s\|}{\|\varepsilon\|} \leqslant \delta, \tag{41}$$

where $\|s\|^2 = \int_0^T s^2(t)\,\mathrm{d}t$ and $\|\varepsilon\|^2 = \int_0^T \varepsilon^2(t)\,\mathrm{d}t$. If $\delta = \infty$, this is the case of minimum error tracking control without disturbance attenuation. To determine the adaptive laws of the parameters of ASAFNC appropriately and guarantee the closed-loop system stability, the Lyapunov function candidate is defined as

$$V = \frac{1}{2}s^2 + \frac{\tilde{\mathbf{w}}^\mathrm{T}\tilde{\mathbf{w}}}{2\eta_w} + \frac{\tilde{\mathbf{m}}^\mathrm{T}\tilde{\mathbf{m}}}{2\eta_m} + \frac{\tilde{\boldsymbol{\sigma}}_\mathbf{l}^\mathrm{T}\tilde{\boldsymbol{\sigma}}_\mathbf{l}}{2\eta_{\sigma_\mathrm{l}}} + \frac{\tilde{\boldsymbol{\sigma}}_\mathbf{r}^\mathrm{T}\tilde{\boldsymbol{\sigma}}_\mathbf{r}}{2\eta_{\sigma_\mathrm{r}}}. \tag{42}$$

Differentiating (42) with respect to time and using (39) yield

$$\begin{aligned}
\dot{V} &= s\dot{s} + \frac{\tilde{\mathbf{w}}^\mathrm{T}\dot{\tilde{\mathbf{w}}}}{\eta_w} + \frac{\tilde{\mathbf{m}}^\mathrm{T}\dot{\tilde{\mathbf{m}}}}{\eta_m} + \frac{\tilde{\boldsymbol{\sigma}}_\mathbf{l}^\mathrm{T}\dot{\tilde{\boldsymbol{\sigma}}}_\mathbf{l}}{\eta_{\sigma_\mathrm{l}}} + \frac{\tilde{\boldsymbol{\sigma}}_\mathbf{r}^\mathrm{T}\dot{\tilde{\boldsymbol{\sigma}}}_\mathbf{r}}{\eta_{\sigma_\mathrm{r}}} \\
&= s(\tilde{\mathbf{w}}^\mathrm{T}\hat{\boldsymbol{\varphi}} + \tilde{\mathbf{m}}^\mathrm{T}\boldsymbol{\varphi}_\mathbf{m}\hat{\mathbf{w}} + \tilde{\boldsymbol{\sigma}}_\mathbf{l}^\mathrm{T}\boldsymbol{\varphi}_{\boldsymbol{\sigma}_\mathbf{l}}\hat{\mathbf{w}} + \tilde{\boldsymbol{\sigma}}_\mathbf{r}^\mathrm{T}\boldsymbol{\varphi}_{\boldsymbol{\sigma}_\mathbf{r}}\hat{w} + \varepsilon - u_{\mathrm{rb}}) + \frac{\tilde{\mathbf{w}}^\mathrm{T}\dot{\tilde{\mathbf{w}}}}{\eta_w} + \frac{\tilde{\mathbf{m}}^\mathrm{T}\dot{\tilde{\mathbf{m}}}}{\eta_m} + \frac{\tilde{\boldsymbol{\sigma}}_\mathbf{l}^\mathrm{T}\dot{\tilde{\boldsymbol{\sigma}}}_\mathbf{l}}{\eta_{\sigma_\mathrm{l}}} + \frac{\tilde{\boldsymbol{\sigma}}_\mathbf{r}^\mathrm{T}\dot{\tilde{\boldsymbol{\sigma}}}_\mathbf{r}}{\eta_{\sigma_\mathrm{r}}} \\
&= \tilde{\mathbf{w}}^\mathrm{T}\left(s\hat{\boldsymbol{\varphi}} + \frac{1}{\eta_w}\dot{\tilde{\mathbf{w}}}\right) + \tilde{\mathbf{m}}^\mathrm{T}\left(s\boldsymbol{\varphi}_\mathbf{m}\hat{\mathbf{w}} + \frac{1}{\eta_m}\dot{\tilde{\mathbf{m}}}\right) + \tilde{\boldsymbol{\sigma}}_\mathbf{l}^\mathrm{T}\left(s\boldsymbol{\varphi}_{\boldsymbol{\sigma}_\mathbf{l}}\hat{\mathbf{w}} + \frac{1}{\eta_{\sigma_\mathrm{l}}}\dot{\tilde{\boldsymbol{\sigma}}}_\mathbf{l}\right) \\
&\quad + \tilde{\boldsymbol{\sigma}}_\mathbf{r}^\mathrm{T}\left(s\boldsymbol{\varphi}_{\boldsymbol{\sigma}_\mathbf{r}}\hat{\mathbf{w}} + \frac{1}{\eta_{\sigma_\mathrm{r}}}\dot{\tilde{\boldsymbol{\sigma}}}_\mathbf{r}\right) + s(\varepsilon - u_{\mathrm{rb}}).
\end{aligned} \tag{43}$$

Choose the adaptive laws as

$$\dot{\tilde{\mathbf{w}}} = -\dot{\hat{\mathbf{w}}} = -\eta_w s\hat{\boldsymbol{\varphi}}, \tag{44}$$

$$\dot{\tilde{\mathbf{m}}} = -\dot{\hat{\mathbf{m}}} = -\eta_m s\boldsymbol{\varphi}_\mathbf{m}\hat{\mathbf{w}}, \tag{45}$$

$$\dot{\tilde{\boldsymbol{\sigma}}}_\mathbf{l} = -\dot{\hat{\boldsymbol{\sigma}}}_\mathbf{l} = -\eta_{\sigma_\mathrm{l}} s\boldsymbol{\varphi}_{\boldsymbol{\sigma}_\mathbf{l}}\hat{\mathbf{w}}, \tag{46}$$

$$\dot{\tilde{\boldsymbol{\sigma}}}_\mathbf{r} = -\dot{\hat{\boldsymbol{\sigma}}}_\mathbf{r} = -\eta_{\sigma_\mathrm{r}} s\boldsymbol{\varphi}_{\boldsymbol{\sigma}_\mathbf{r}}\hat{\mathbf{w}} \tag{47}$$

and the robust controller is designed as

$$u_{\mathrm{rb}} = \frac{\delta^2 + 1}{2\delta^2} s. \tag{48}$$

Thus, (43) can be rewritten as

$$\begin{aligned}\dot{V} &= s\left(\varepsilon - \frac{\delta^2+1}{2\delta^2}s\right)\\ &= s\varepsilon - \frac{s^2}{2} - \frac{s^2}{2\delta^2}\\ &= -\frac{s^2}{2} - \frac{1}{2}\left(\frac{s}{\delta} - \varepsilon\delta\right)^2 + \frac{1}{2}\varepsilon^2\delta^2\\ &\leqslant -\frac{1}{2}s^2 + \frac{1}{2}\varepsilon^2\delta^2.\end{aligned} \tag{49}$$

Assume $\varepsilon \in L_2[0, T]$, $\forall T \in [0, \infty)$. Integrating the above equation from $t = 0$ to T yields

$$V(T) - V(0) \leqslant -\frac{1}{2}\int_0^T s^2\,\mathrm{d}t + \frac{1}{2}\delta^2\int_0^T \varepsilon^2\,\mathrm{d}t. \tag{50}$$

Since $V(t) \geqslant 0$, we can arrange (50) as follows:

$$\frac{1}{2}\int_0^T s^2\,\mathrm{d}t \leqslant V(0) + \frac{1}{2}\delta^2\int_0^T \varepsilon^2\,\mathrm{d}t \tag{51}$$

which is equivalent to inequality (40) , i.e., L_2 tracking performance. Assume $\varepsilon \in L_2$, then the sliding surface s will converge to a certain small boundary. It is implied that the tracking error e will also converge to a certain small boundary [24].

3.4. Boundary analysis using projection algorithm

Although the stability of ASAFNC can be guaranteed, the parameters $\hat{\mathbf{w}}$, $\hat{\mathbf{m}}$, $\hat{\boldsymbol{\sigma}}_{\mathbf{l}}$, and $\hat{\boldsymbol{\sigma}}_{\mathbf{r}}$ cannot be guaranteed within a desired bound value by using the adaptive laws (44)–(47). According to the projection algorithm [11], the adaptive laws can be modified as follows. The adaptive law of weight is

$$\dot{\hat{\mathbf{w}}} = \begin{cases} \eta_w s\hat{\boldsymbol{\varphi}} & \text{if} \|\hat{\mathbf{w}}\| < D_{\mathbf{w}} \text{ or } (\|\hat{\mathbf{w}}\| = D_{\mathbf{w}} \text{ and } s\hat{\mathbf{w}}^{\mathrm{T}}\hat{\boldsymbol{\varphi}} \leqslant 0),\\ \mathbf{Pr}(\eta_w s\hat{\boldsymbol{\varphi}}) & \text{if } (\|\hat{\mathbf{w}}\| = D_{\mathbf{w}} \text{ and } s\hat{\mathbf{w}}^{\mathrm{T}}\hat{\boldsymbol{\varphi}} > 0),\end{cases} \tag{52}$$

where the projection operator is given as

$$\mathbf{Pr}(\eta_w s\hat{\boldsymbol{\varphi}}) = \eta_w s\hat{\boldsymbol{\varphi}} - \eta_w s\frac{\hat{\mathbf{w}}^{\mathrm{T}}\hat{\boldsymbol{\varphi}}}{\|\hat{\mathbf{w}}\|^2}\hat{\mathbf{w}}. \tag{53}$$

The adaptive law of mean of asymmetric membership function is

$$\dot{\hat{\mathbf{m}}} = \begin{cases} \eta_m s\boldsymbol{\varphi}_{\mathbf{m}}\hat{\mathbf{w}} & \text{if } \|\hat{\mathbf{m}}\| < D_{\mathbf{m}} \text{ or } (\|\hat{\mathbf{m}}\| = D_{\mathbf{m}} \text{ and } s\hat{\mathbf{m}}^{\mathrm{T}}\boldsymbol{\varphi}_{\mathbf{m}}\hat{\mathbf{w}} \leqslant 0),\\ \mathbf{Pr}(\eta_m s\boldsymbol{\varphi}_{\mathbf{m}}\hat{\mathbf{w}}) & \text{if } (\|\hat{\mathbf{m}}\| = D_{\mathbf{m}} \text{ and } s\hat{\mathbf{m}}^{\mathrm{T}}\boldsymbol{\varphi}_{\mathbf{m}}\hat{\mathbf{w}} > 0),\end{cases} \tag{54}$$

where the projection operator is given as

$$\mathbf{Pr}(\eta_m s\boldsymbol{\varphi}_{\mathbf{m}}\hat{\mathbf{w}}) = \eta_m s\boldsymbol{\varphi}_{\mathbf{m}}\hat{\mathbf{w}} - \eta_m s\frac{\hat{\mathbf{m}}^{\mathrm{T}}\boldsymbol{\varphi}_{\mathbf{m}}\hat{\mathbf{w}}}{\|\hat{\mathbf{m}}\|^2}\hat{\mathbf{m}}. \tag{55}$$

The adaptive law of left-side variance of asymmetric membership function is

$$\dot{\hat{\boldsymbol{\sigma}}}_{\mathbf{l}} = \begin{cases} \eta_{\sigma_{\mathbf{l}}} s\boldsymbol{\varphi}_{\boldsymbol{\sigma}_{\mathbf{l}}}\hat{\mathbf{w}} & \text{if } \|\hat{\boldsymbol{\sigma}}_{\mathbf{l}}\| < D_{\boldsymbol{\sigma}_{\mathbf{l}}} \text{ or } (\|\hat{\boldsymbol{\sigma}}_{\mathbf{l}}\| = D_{\boldsymbol{\sigma}_{\mathbf{l}}} \text{ and } s\hat{\boldsymbol{\sigma}}_{\mathbf{l}}^{\mathrm{T}}\boldsymbol{\varphi}_{\boldsymbol{\sigma}_{\mathbf{l}}}\hat{\mathbf{w}} \leqslant 0),\\ \mathbf{Pr}(\eta_{\sigma_{\mathbf{l}}} s\boldsymbol{\varphi}_{\boldsymbol{\sigma}_{\mathbf{l}}}\hat{\mathbf{w}}) & \text{if } (\|\hat{\boldsymbol{\sigma}}_{\mathbf{l}}\| = D_{\boldsymbol{\sigma}_{\mathbf{l}}} \text{ and } s\hat{\boldsymbol{\sigma}}_{\mathbf{l}}^{\mathrm{T}}\boldsymbol{\varphi}_{\boldsymbol{\sigma}_{\mathbf{l}}}\hat{\mathbf{w}} > 0),\end{cases} \tag{56}$$

where the projection operator is given as

$$\mathbf{Pr}(\eta_{\sigma_l} s\boldsymbol{\varphi}_{\boldsymbol{\sigma}_l}\hat{\mathbf{w}}) = \eta_{\sigma_l} s\boldsymbol{\varphi}_{\boldsymbol{\sigma}_l}\hat{\mathbf{w}} - \eta_{\sigma_l} s\frac{\hat{\boldsymbol{\sigma}}_{\mathbf{l}}^{\mathrm{T}}\boldsymbol{\varphi}_{\boldsymbol{\sigma}_l}\hat{\mathbf{w}}}{\|\hat{\boldsymbol{\sigma}}_{\mathbf{l}}\|^2}\hat{\boldsymbol{\sigma}}_{\mathbf{l}}. \tag{57}$$

The adaptive law of right-side variance of asymmetric membership function is

$$\dot{\hat{\boldsymbol{\sigma}}}_{\mathbf{r}} = \begin{cases} \eta_{\sigma_r} s\boldsymbol{\varphi}_{\boldsymbol{\sigma}_r}\hat{\mathbf{w}} & \text{if } \|\hat{\boldsymbol{\sigma}}_{\mathbf{r}}\| < D_{\boldsymbol{\sigma}_r} \text{ or } (\|\hat{\boldsymbol{\sigma}}_{\mathbf{r}}\| = D_{\boldsymbol{\sigma}_r} \text{ and } s\hat{\boldsymbol{\sigma}}_{\mathbf{r}}^{\mathrm{T}}\boldsymbol{\varphi}_{\boldsymbol{\sigma}_r}\hat{\mathbf{w}} \leqslant 0), \\ \mathbf{Pr}(\eta_{\sigma_r} s\boldsymbol{\varphi}_{\boldsymbol{\sigma}_r}\hat{\mathbf{w}}) & \text{if } (\|\hat{\boldsymbol{\sigma}}_{\mathbf{r}}\| = D_{\boldsymbol{\sigma}_r} \text{ and } s\hat{\boldsymbol{\sigma}}_{\mathbf{r}}^{\mathrm{T}}\boldsymbol{\varphi}_{\boldsymbol{\sigma}_r}\hat{\mathbf{w}} > 0), \end{cases} \tag{58}$$

where the projection operator is given as

$$\mathbf{Pr}(\eta_{\sigma_r} s\boldsymbol{\varphi}_{\boldsymbol{\sigma}_r}\hat{\mathbf{w}}) = \eta_{\sigma_r} s\boldsymbol{\varphi}_{\boldsymbol{\sigma}_r}\hat{\mathbf{w}} - \eta_{\sigma_r} s\frac{\hat{\boldsymbol{\sigma}}_{\mathbf{r}}^{\mathrm{T}}\boldsymbol{\varphi}_{\boldsymbol{\sigma}_r}\hat{\mathbf{w}}}{\|\hat{\boldsymbol{\sigma}}_{\mathbf{r}}\|^2}\hat{\boldsymbol{\sigma}}_{\mathbf{r}}. \tag{59}$$

If the initial value of $\hat{\mathbf{w}}$ is bounded (i.e., $\hat{\mathbf{w}}(0) \in \boldsymbol{\Omega}_{\mathbf{w}}$), $\|\hat{\mathbf{w}}\|$ is bounded by the constraint set $\boldsymbol{\Omega}_{\mathbf{w}}$ for all $t \geqslant 0$. Similarly, the results can also be derived that $\|\hat{\mathbf{m}}\|$ is bounded by the constraint set $\boldsymbol{\Omega}_{\mathbf{m}}$ if $\hat{\mathbf{m}}(0) \in \boldsymbol{\Omega}_{\mathbf{m}}$; $\|\hat{\boldsymbol{\sigma}}_{\mathbf{l}}\|$ is bounded by the constraint set $\boldsymbol{\Omega}_{\boldsymbol{\sigma}_l}$ if $\hat{\boldsymbol{\sigma}}_{\mathbf{l}}(0) \in \boldsymbol{\Omega}_{\boldsymbol{\sigma}_l}$; and $\|\hat{\boldsymbol{\sigma}}_{\mathbf{r}}\|$ is bounded by the constraint set $\boldsymbol{\Omega}_{\boldsymbol{\sigma}_r}$ if $\hat{\boldsymbol{\sigma}}_{\mathbf{r}}(0) \in \boldsymbol{\Omega}_{\boldsymbol{\sigma}_r}$, for all $t \geqslant 0$. Thus, the fact that the uncertain term ε is bounded can be guaranteed by the modified adaptive laws (52), (54), (56) and (58). Next, define some variables as

$$J_{\mathbf{w}} = \tilde{\mathbf{w}}^{\mathrm{T}}\left(s\hat{\boldsymbol{\varphi}} + \frac{1}{\eta_w}\dot{\tilde{\mathbf{w}}}\right), \tag{60}$$

$$J_{\mathbf{m}} = \tilde{\mathbf{m}}^{\mathrm{T}}\left(s\boldsymbol{\varphi}_{\mathbf{m}}\hat{\mathbf{w}} + \frac{1}{\eta_m}\dot{\tilde{\mathbf{m}}}\right), \tag{61}$$

$$J_{\boldsymbol{\sigma}_l} = \tilde{\boldsymbol{\sigma}}_{\mathbf{l}}^{\mathrm{T}}\left(s\boldsymbol{\varphi}_{\boldsymbol{\sigma}_l}\hat{\mathbf{w}} + \frac{1}{\eta_{\sigma_l}}\dot{\tilde{\boldsymbol{\sigma}}}_{\mathbf{l}}\right) \tag{62}$$

and

$$J_{\boldsymbol{\sigma}_r} = \tilde{\boldsymbol{\sigma}}_{\mathbf{r}}^{\mathrm{T}}\left(s\boldsymbol{\varphi}_{\boldsymbol{\sigma}_r}\hat{\mathbf{w}} + \frac{1}{\eta_{\sigma_r}}\dot{\tilde{\boldsymbol{\sigma}}}_{\mathbf{r}}\right). \tag{63}$$

If the projection algorithm is taken place, the property $\tilde{\mathbf{z}}^{\mathrm{T}}\hat{\mathbf{z}} = 0.5(\|\mathbf{z}^*\|^2 - \|\hat{\mathbf{z}}\|^2 - \|\tilde{\mathbf{z}}\|^2) < 0$ is applied according to $\|\hat{\mathbf{z}}\| = \boldsymbol{\Omega}_{\mathbf{z}} > \|\mathbf{z}^*\|$, where $\mathbf{z} = \hat{\mathbf{w}}$, $\hat{\mathbf{m}}$, $\hat{\boldsymbol{\sigma}}_{\mathbf{l}}$, and $\hat{\boldsymbol{\sigma}}_{\mathbf{r}}$. Thus, the following equations can be obtained:

$$J_{\mathbf{w}} = \frac{s}{2}\frac{(\|\mathbf{w}^*\|^2 - \|\hat{\mathbf{w}}\|^2 - \|\tilde{\mathbf{w}}\|^2)}{\|\hat{\mathbf{w}}\|^2}\hat{\mathbf{w}}^{\mathrm{T}}\hat{\boldsymbol{\varphi}} \leqslant 0 \quad \text{for } (\|\hat{\mathbf{w}}\| = D_{\mathbf{w}} \text{ and } s\hat{\mathbf{w}}^{\mathrm{T}}\hat{\boldsymbol{\varphi}} > 0), \tag{64}$$

$$J_{\mathbf{m}} = \frac{s}{2}\frac{(\|\mathbf{m}^*\|^2 - \|\hat{\mathbf{m}}\|^2 - \|\tilde{\mathbf{m}}\|^2)}{\|\hat{\mathbf{m}}\|^2}\hat{\mathbf{m}}^{\mathrm{T}}\boldsymbol{\varphi}_{\mathbf{m}}\hat{\mathbf{w}} \leqslant 0 \quad \text{for } (\|\hat{\mathbf{m}}\| = D_{\mathbf{m}} \text{ and } s\hat{\mathbf{m}}^{\mathrm{T}}\boldsymbol{\varphi}_{\mathbf{m}}\hat{\mathbf{w}} > 0), \tag{65}$$

$$J_{\boldsymbol{\sigma}_l} = \frac{s}{2}\frac{(\|\boldsymbol{\sigma}_{\mathbf{l}}^*\|^2 - \|\hat{\boldsymbol{\sigma}}_{\mathbf{l}}\|^2 - \|\tilde{\boldsymbol{\sigma}}_{\mathbf{l}}\|^2)}{\|\hat{\boldsymbol{\sigma}}_{\mathbf{l}}\|^2}\hat{\boldsymbol{\sigma}}_{\mathbf{l}}^{\mathrm{T}}\boldsymbol{\varphi}_{\boldsymbol{\sigma}_l}\hat{\mathbf{w}} \leqslant 0 \quad \text{for } (\|\hat{\boldsymbol{\sigma}}_{\mathbf{l}}\| = D_{\boldsymbol{\sigma}_l} \text{ and } s\hat{\boldsymbol{\sigma}}_{\mathbf{l}}^{\mathrm{T}}\boldsymbol{\varphi}_{\boldsymbol{\sigma}_l}\hat{\mathbf{w}} > 0) \tag{66}$$

and

$$J_{\boldsymbol{\sigma}_r} = \frac{s}{2}\frac{(\|\boldsymbol{\sigma}_{\mathbf{r}}^*\|^2 - \|\hat{\boldsymbol{\sigma}}_{\mathbf{r}}\|^2 - \|\tilde{\boldsymbol{\sigma}}_{\mathbf{r}}\|^2)}{\|\hat{\boldsymbol{\sigma}}_{\mathbf{r}}\|^2}\hat{\boldsymbol{\sigma}}_{\mathbf{r}}^{\mathrm{T}}\boldsymbol{\varphi}_{\boldsymbol{\sigma}_r}\hat{\mathbf{w}} \leqslant 0 \quad \text{for } (\|\hat{\boldsymbol{\sigma}}_{\mathbf{r}}\| = D_{\boldsymbol{\sigma}_r} \text{ and } s\hat{\boldsymbol{\sigma}}_{\mathbf{r}}^{\mathrm{T}}\boldsymbol{\varphi}_{\boldsymbol{\sigma}_r}\hat{\mathbf{w}} > 0). \tag{67}$$

Then, the derivative of Lyapunov function (43) can be rewritten as

$$\begin{aligned} \dot{V} &= J_{\mathbf{w}} + J_{\mathbf{m}} + J_{\boldsymbol{\sigma}_l} + J_{\boldsymbol{\sigma}_r} + s(\varepsilon - u_{\mathrm{rb}}) \\ &\leqslant s(\varepsilon - u_{\mathrm{rb}}). \end{aligned} \tag{68}$$

By substituting the robust controller (48), (68) can be rewritten as

$$\begin{aligned}\dot{V} &\leqslant s\left(\varepsilon - \frac{\delta^2+1}{2\delta^2}s\right)\\ &= -\frac{s^2}{2} - \frac{1}{2}\left(\frac{s}{\delta} - \varepsilon\delta\right)^2 + \frac{1}{2}\varepsilon^2\delta^2\\ &\leqslant -\frac{1}{2}s^2 + \frac{1}{2}\varepsilon^2\delta^2. \end{aligned} \tag{69}$$

Using the same discussion in Section 3.3, the stability of the system with the projection algorithm can also be guaranteed.

4. Simulation results

In this section, the proposed ASAFNC is applied to a second-order chaotic dynamics system to verify its effectiveness. This scheme emphasizes that the parameter and network structure of the SFNN can be tuned online by the proposed algorithm. Consider a second-order chaotic dynamics system such as Duffing's equation describing a special nonlinear circuit or a pendulum moving in a viscous medium as follows [5]:

$$\ddot{x} = f(\mathbf{x}) + u, \tag{70}$$

where $f(\mathbf{x}) = -p\dot{x} - p_1 x - p_2 x^3 + q\cos(\omega t)$ is the system dynamics, t is the time variable, ω is the frequency, u is the control force, and p, p_1, p_2, and q are real constants. The solutions of (70) may exhibit periodic depending on the choice of these constants, i.e., it is almost periodic and chaotic behavior. The open-loop system behavior, i.e., $u = 0$, is simulated with $p = 0.4$, $p_1 = -1.1$, $p_2 = 1.0$, and $\omega = 1.8$ for observing the chaotic unpredictable behavior. The phase plane plots with an initial condition point (0, 0) are shown in Figs. 4(a) and (b) for $q = 1.95$ and 7.00, respectively. The uncontrolled chaotic system has different trajectories for different values of q. To illustrate the effectiveness of the proposed design method, a comparison among a fix-structure AFNC using symmetric Gaussian membership functions [21], a fix-structure AFNC using asymmetric Gaussian membership functions [2], and the proposed ASAFNC is made.

The simulation results of fix-structure AFNC using three symmetric membership functions are shown in Fig. 5. The tracking responses of state x are shown in Figs. 5(a) and (d); the tracking responses of state $\dot{x}$ are shown in Figs. 5(b) and (e); and the associated control efforts are shown in Figs. 5(c) and (f) for $q = 1.95$ and 7.00, respectively. The simulation results show that the tracking responses decline when membership functions are selected insufficiently. Next, the simulation results of fix-structure AFNC using 20 symmetric membership functions are shown in Fig. 6. The tracking responses of state x are shown in Figs. 6(a) and (d); the tracking responses of state $\dot{x}$ are shown in Figs. 6(b) and (e); and the associated control efforts are shown in Figs. 6(c) and (f) for $q = 1.95$ and 7.00, respectively. The simulation results show that the favorable tracking performance can achieve; however, the computation load is heavy. These results demonstrate the fact that it is difficult to consider the balance between the rule number and the desired performance.

To show that the learning capability of neural network can be upgraded as using the asymmetric Gaussian membership functions, the fix-structure AFNC using asymmetric Gaussian membership functions is applied to chaotic dynamics system again. The simulation results of fix-structure AFNC using three asymmetric membership functions are shown in Fig. 7. The tracking responses of state x are shown in Figs. 7(a) and (d); the tracking responses of state $\dot{x}$ are shown in Figs. 7(b) and (e); and the associated control efforts are shown in Figs. 7(c) and (f) for $q = 1.95$ and 7.00, respectively. The simulation results show that the favorable tracking performance can be achieved. Next, the simulation results of fix-structure AFNC using 20 asymmetric membership functions are shown in Fig. 8. The tracking responses of state x are shown in Figs. 8(a) and (d); the tracking responses of state $\dot{x}$ are shown in Figs. 8(b) and (e); and the associated control efforts are shown in Figs. 8(c) and (f) for $q = 1.95$ and 7.00, respectively. The simulation results show that the favorable tracking performance can achieve; however, the computation load is heavy. Comparing with Figs. 5 and 7, and Figs. 6 and 8 shows that the adaptive fuzzy neural network with asymmetric membership functions performs better than the adaptive fuzzy neural network with symmetric membership functions. However, the structure of the FNN should still be determined by trial-and-error.

To solve this problem, the proposed ASAFNC is applied to the chaotic dynamics system. The parameters of ASAFNC system are selected as $k_1 = 2$, $k_2 = 1$, $\eta_w = 50$, $\eta_m = \eta_{\sigma_1} = \eta_{\sigma_r} = 0.5$, $G_{\text{th}} = 0.4$, $I_{\text{th}} = 0.1$, $P_{\text{th}} = 0.1$, $\tau_1 = 0.01$,

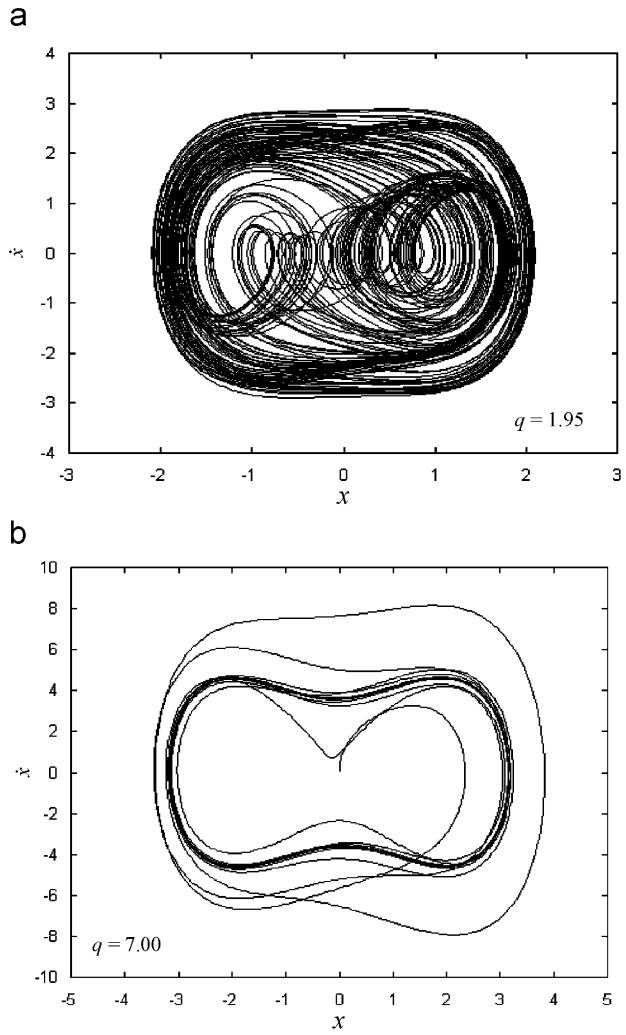

Fig. 4. Phase plane of uncontrolled chaotic dynamics system.

$\tau_2 = 0.05$, $\sigma_i = 0.6$, $\varpi = 0.1$, and $\delta = 0.6$. All the gains in the proposed control system are chosen to achieve the best transient control performance considering the stability and possible operating conditions. The parameters η_w, η_m, η_{σ_1}, and η_{σ_r} are the leaning rates of SFNN. If the leaning rates are chosen too small, the parameter convergence of SFNN will be easily achieved; however, this will result in slow learning speed. On the other hand, if the leaning rates are chosen too large, the learning speed will be fast; however, the SFNN system may become more unstable. The simulation results of ASAFNC for $q = 1.95$ and 7.00 are shown in Figs. 9 and 10, respectively. The tracking responses of state x are shown in Figs. 9(a) and 10(a); the tracking responses of state $\dot{x}$ are shown in Figs. 9(b) and 10(b); the associated control efforts are shown in Figs. 9(c) and 10(c); the number of fuzzy rules is shown in Figs. 9(d) and 10(d); and the final shapes of membership functions are shown in Figs. 9(e) and 10(e), respectively. These results state that the rule number and good tracking performance can be considered simultaneously in the simulation procedure. To demonstrate the control performance of the proposed ASAFNC system with different reference trajectories, the command $x_c(t) = \sin(1.5t) + 0.5\cos(3.5t)$ is examined here. The simulation results for $q = 1.95$ and 7.00 are shown in Figs. 11 and 12, respectively. The tracking responses of state x are shown in Figs. 11(a) and 12(a); the tracking responses of state $\dot{x}$ are shown in Figs. 11(b) and 12(b); the associated control efforts are shown in Figs. 11(c) and 12(c); the number of fuzzy rules is shown in Figs. 11(d) and 12(d); and the final shapes of membership functions are shown in Figs. 11(e) and 12(e), respectively. The simulation results show that the proposed ASAFNC system, which

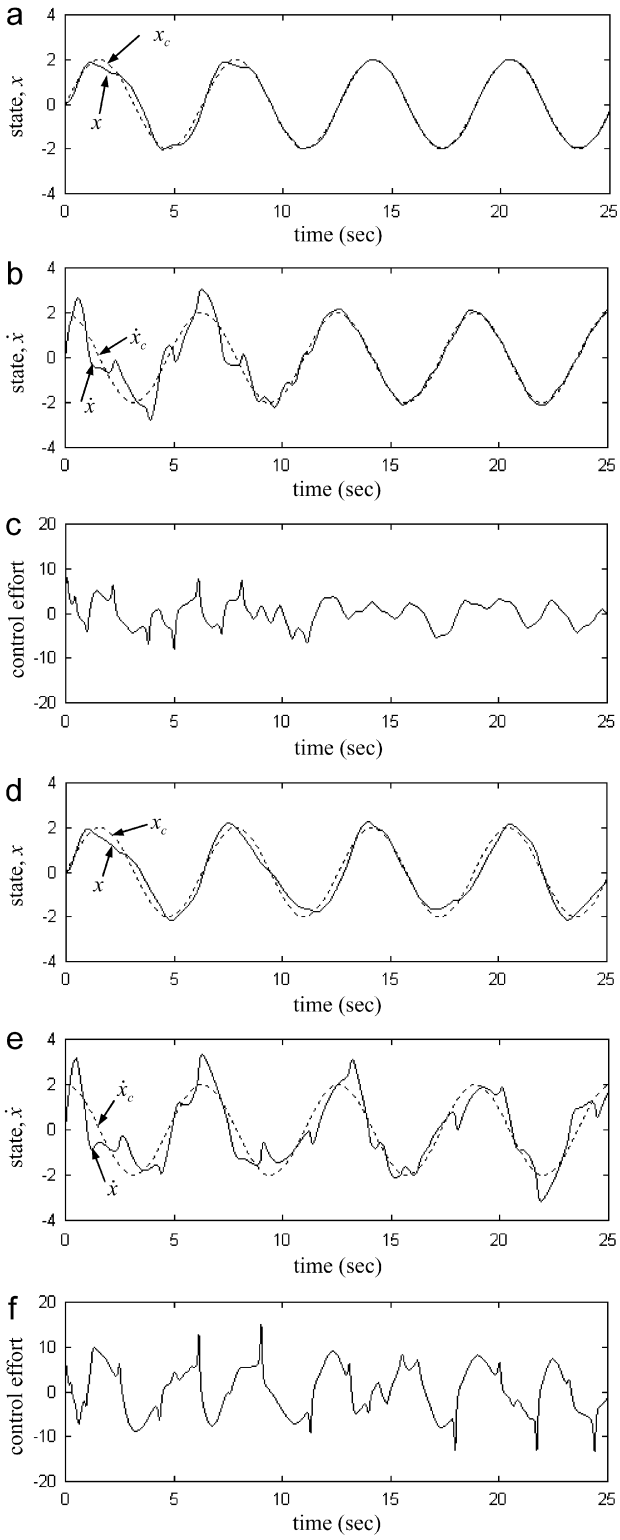

Fig. 5. Simulation results of AFNC using three symmetric membership functions.

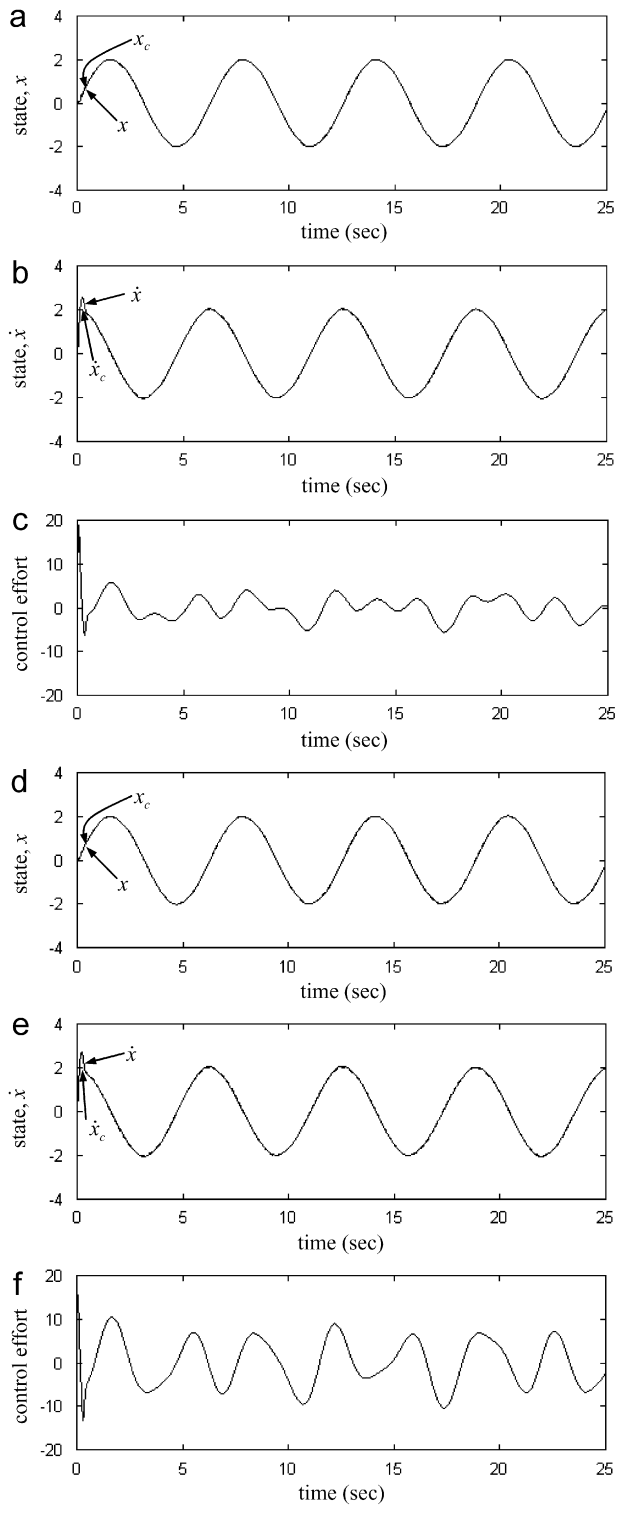

Fig. 6. Simulation results of AFNC using 20 symmetric membership functions.

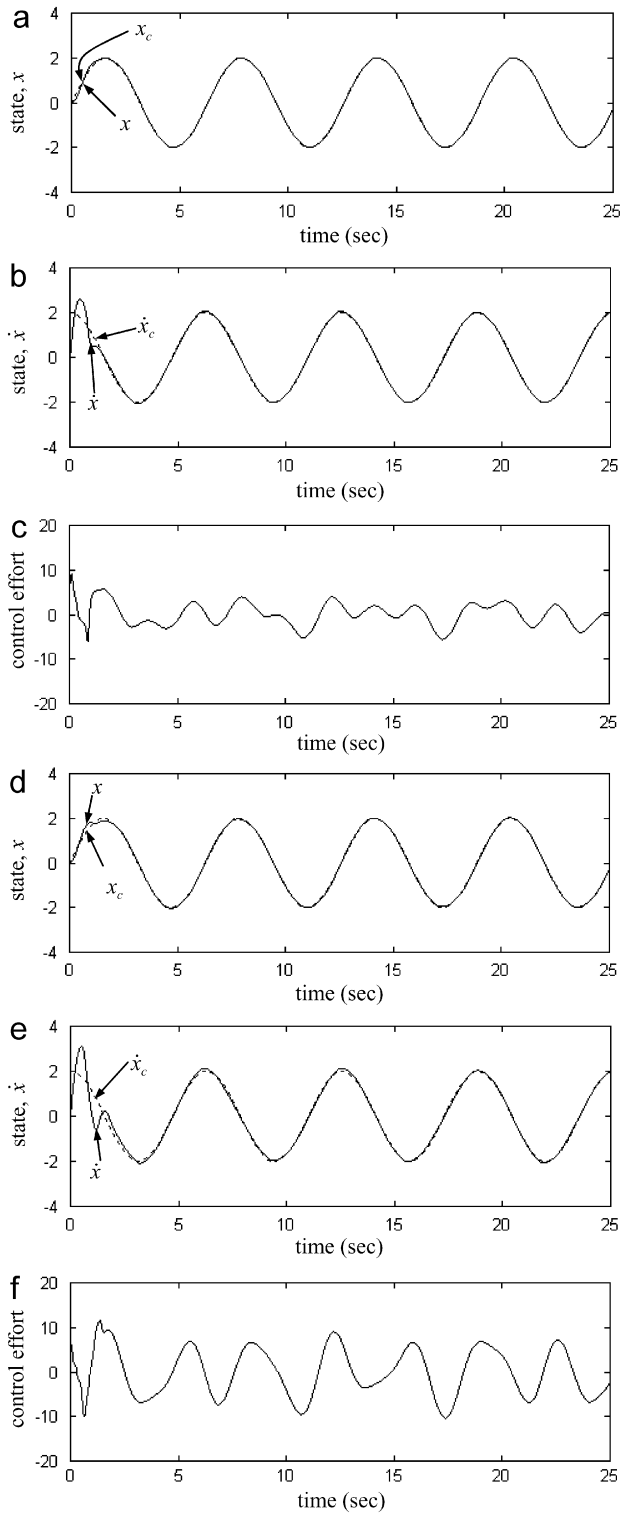

Fig. 7. Simulation results of AFNC using three asymmetric membership functions.

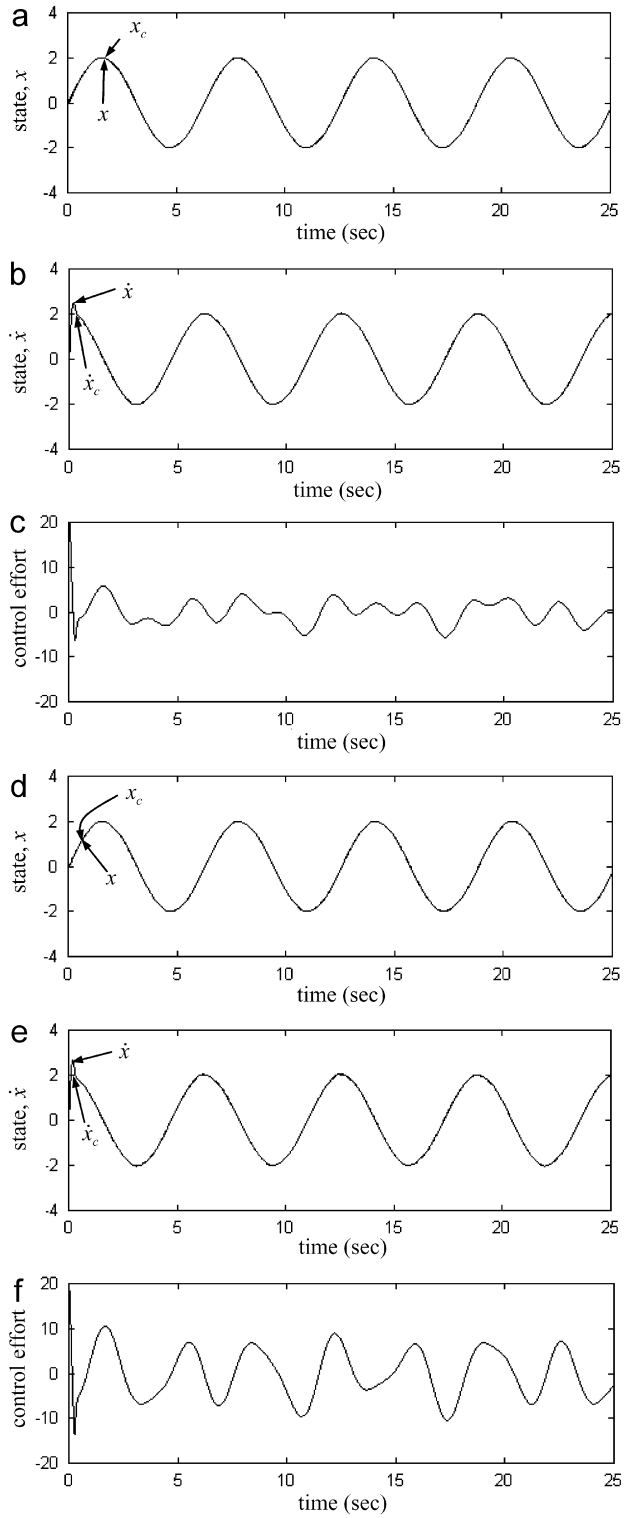

Fig. 8. Simulation results of AFNC using 20 asymmetric membership functions.

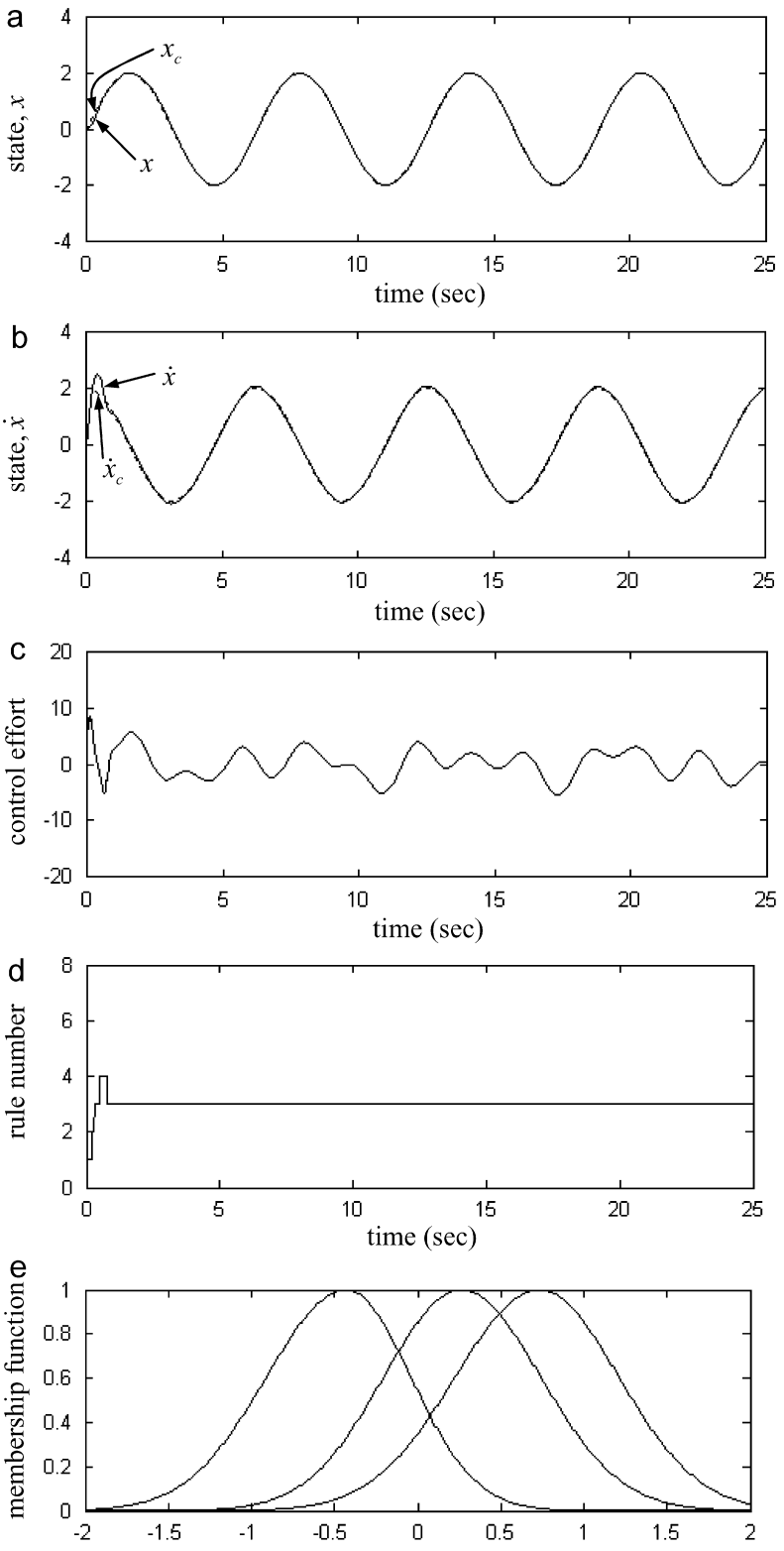

Fig. 9. Simulation results of ASAFNC for $q = 1.95$.

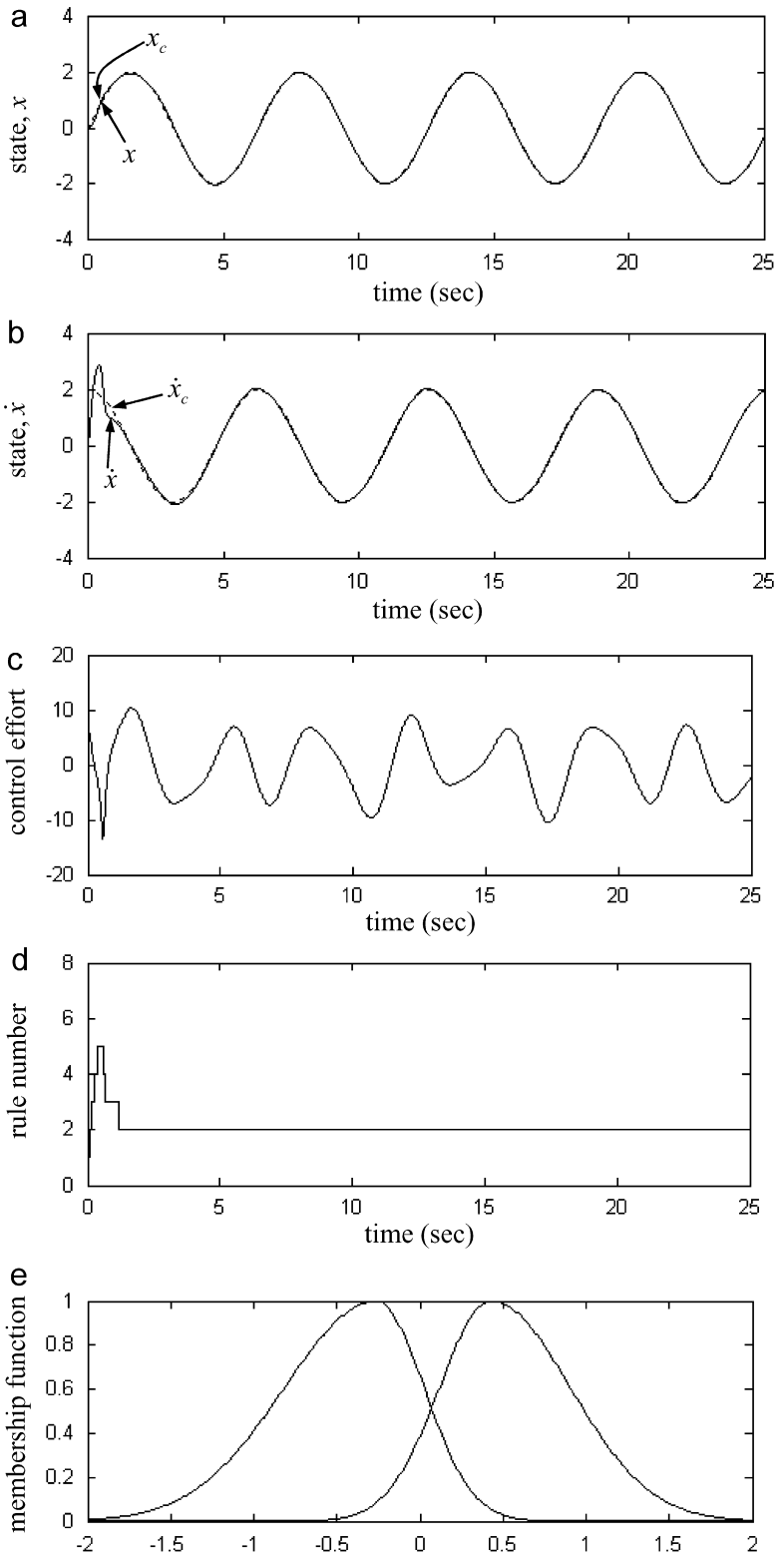

Fig. 10. Simulation results of ASAFNC for $q = 7.00$.

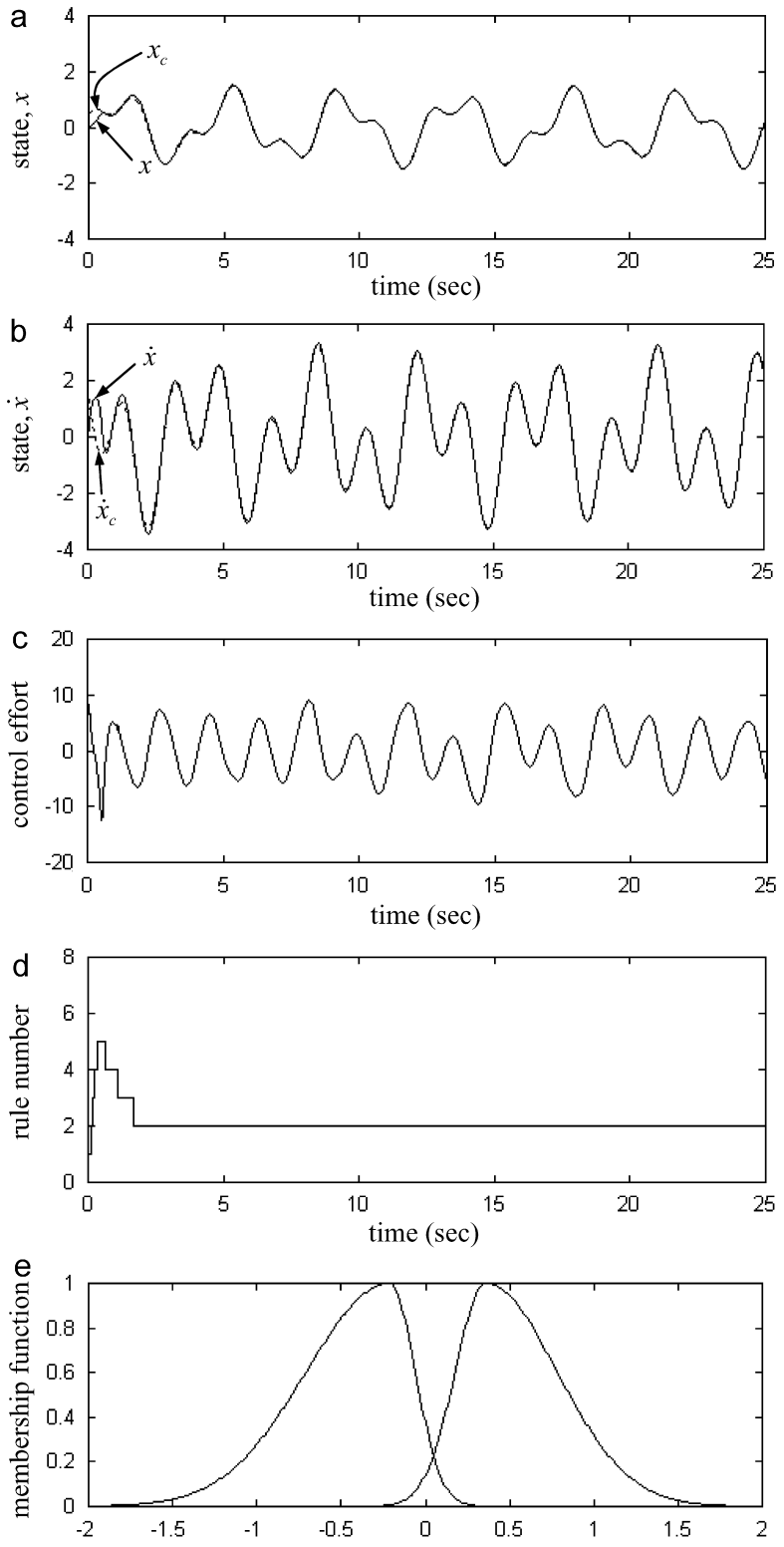

Fig. 11. Simulation results of ASAFNC for $q = 1.95$ with different trajectory.

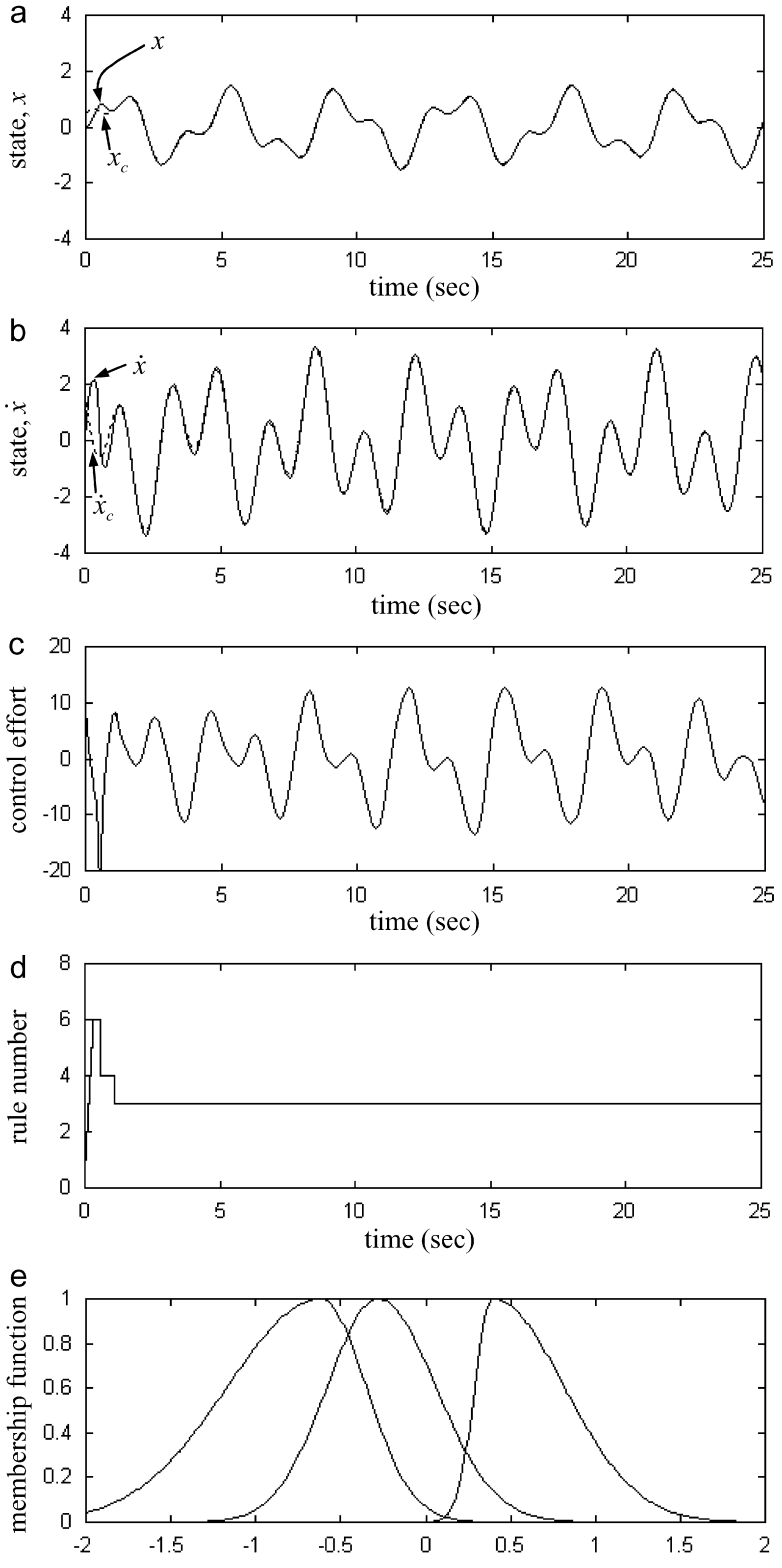

Fig. 12. Simulation results of ASAFNC for $q = 7.00$ with different trajectory.

includes SFNN with the asymmetric Gaussian membership function, can achieve satisfactory tracking responses in the presence of different reference trajectories. Moreover, a concise SFNN structure can be obtained by the proposed self-structuring mechanism and the online learning algorithms.

5. Conclusions

This paper develops an ASAFNC system, which consists of an SFNN controller and a robust controller. In the SFNN controller, SFNN using asymmetric Gaussian membership functions is utilized to mimic an ideal controller in the structure and parameter learning phases. The structure learning phase of SFNN is used to find how many rules and membership functions are necessary, and the parameter learning phase of SFNN is concerned with the parameter values of membership functions in the premise part and the crisp value in the consequence part. The robust controller is designed to compensate for the modeling error between the SFNN controller and the ideal controller. An online training methodology is developed in the Lyapunov sense, and thus the stability of the closed-loop control system can be guaranteed. Finally, the simulation results of a chaotic dynamics system show that the ASAFNC can achieve favorable tracking performance without control system dynamics.

Acknowledgments

The authors appreciate the partial financial support from the National Science Council of Republic of China under Grant NSC 96-2218-E-216-001. The authors would like to express their gratitude to the Area Editor and Reviewers for their valuable comments and suggestions.

References

[1] A. Chatterjee, K. Pulasinghe, K. Watanabe, K. Izumi, A particle-swarm-optimized fuzzy-neural network for voice-controlled robot systems, IEEE Trans. Industrial Electron. 52 (2005) 1478–1489.
[2] K.H. Cheng, C.F. Hsu, C.M. Lin, T.T. Lee, C. Li, Fuzzy-neural sliding-mode control for DC–DC converters using asymmetric Gaussian membership functions, IEEE Trans. Industrial Electron. 54 (2007) 1528–1536.
[3] Y. Gao, M.J. Er, Online adaptive fuzzy neural identification and control of a class of MIMO nonlinear systems, IEEE Trans. Fuzzy Systems 11 (2003) 462–477.
[4] C.F. Hsu, Self-organizing adaptive fuzzy neural control for a class of nonlinear systems, IEEE Trans. Neural Networks 18 (2007) 1232–1241.
[5] C.F. Hsu, C.M. Lin, T.T. Lee, Wavelet adaptive backstepping control for a class of nonlinear systems, IEEE Trans. Neural Networks 17 (2006) 1175–1183.
[6] G.B. Huang, P. Saratchandran, N. Sundararajan, An efficient sequential learning algorithm for growing and pruning RBF (GAP-RBF) networks, IEEE Trans. Systems Man Cybernet. 34 (2004) 2284–2292.
[7] G. Leng, T.M. McGinnity, G. Prasad, An approach for on-line extraction of fuzzy rules using a self-organising fuzzy neural network, Fuzzy Sets and Systems 150 (2005) 211–243.
[8] Y.G. Leu, W.Y. Wang, T.T. Lee, Observer-based direct adaptive fuzzy-neural control for nonaffine nonlinear systems, IEEE Trans. Neural Networks 16 (2005) 853–861.
[9] C. Li, C.Y. Lee, K.H. Cheng, Pseudo-error-based self-organizing neuro-fuzzy system, IEEE Trans. Fuzzy Systems 12 (2004) 812–819.
[10] C.J. Lin, Y.J. Xu, A self-adaptive neural fuzzy network with group-based symbiotic evolution and its prediction applications, Fuzzy Sets and Systems 157 (2006) 1036–1056.
[11] C.M. Lin, C.F. Hsu, Supervisory recurrent fuzzy neural network control of wing rock for slender delta wings, IEEE Trans. Fuzzy Systems 12 (2004) 733–742.
[12] C.M. Lin, Y.F. Peng, Adaptive CMAC-based supervisory control for uncertain nonlinear systems, IEEE Trans. Systems Man Cybernet. 34 (2004) 1248–1260.
[13] C.T. Lin, W.C. Cheng, S.F. Liang, An on-line ICA-mixture-model-based self-constructing fuzzy neural network, IEEE Trans. Circuits Systems I 52 (2005) 207–221.
[14] C.T. Lin, C.S.G. Lee, Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems, Prentice-Hall, Englewood Cliffs, NJ, 1996.
[15] F.J. Lin, C.H. Lin, A permanent-magnet synchronous motor servo drive using self-constructing fuzzy neural network controller, IEEE Trans. Energy Conversion 19 (2004) 66–72.
[16] F.J. Lin, C.H. Lin, P.H. Shen, Self-constructing fuzzy neural network speed controller for permanent-magnet synchronous motor drive, IEEE Trans. Fuzzy Systems 9 (2001) 751–759.
[17] D. Nauck, F. Klawonn, R. Kruse, Foundations of Neuro-Fuzzy Systems, Wiley, New York, 1997.
[18] J.H. Park, G.T. Park, S.H. Kim, C.J. Moon, Direct adaptive self-structuring fuzzy controller for nonaffine nonlinear system, Fuzzy Sets and Systems 153 (2005) 429–445.
[19] J.J.E. Slotine, W.P. Li, Applied Nonlinear Control, Prentice-Hall, Englewood Cliffs, NJ, 1991.

[20] C.S. Velayutham, S. Kumar, Asymmetric subsethood-product fuzzy neural inference system (ASuPFuNIS), IEEE Trans. Neural Networks 16 (2005) 160–174.
[21] R.J. Wai, C.M. Lin, C.F. Hsu, Adaptive fuzzy sliding-mode control for electrical servo drive, Fuzzy Sets and Systems 143 (2004) 295–310.
[22] C.H. Wang, H.L. Liu, T.C. Lin, Direct adaptive fuzzy-neural control with state observer and supervisory controller for unknown nonlinear dynamical systems, IEEE Trans. Fuzzy Systems 10 (2002) 39–49.
[23] L.X. Wang, Adaptive Fuzzy Systems and Control: Design and Stability Analysis, Prentice-Hall, Englewood Cliffs, NJ, 1994.
[24] W.Y. Wang, M.L. Chan, C.C.J. Hsu, T.T. Lee, H^{∞} tracking-based sliding mode controller for uncertain nonlinear systems via an adaptive fuzzy-neural approach, IEEE Trans. Systems Man Cybernet. 32 (2002) 483–492.
[25] S. Wu, M.J. Er, Dynamic fuzzy neural networks—a novel approach to function approximation, IEEE Trans. Systems Man Cybernet. 30 (2000) 358–364.
[26] S. Wu, M.J. Er, Y. Gao, A fast approach for automatic generation of fuzzy rules by generalized dynamic fuzzy neural networks, IEEE Trans. Fuzzy Systems 9 (2001) 578–594.