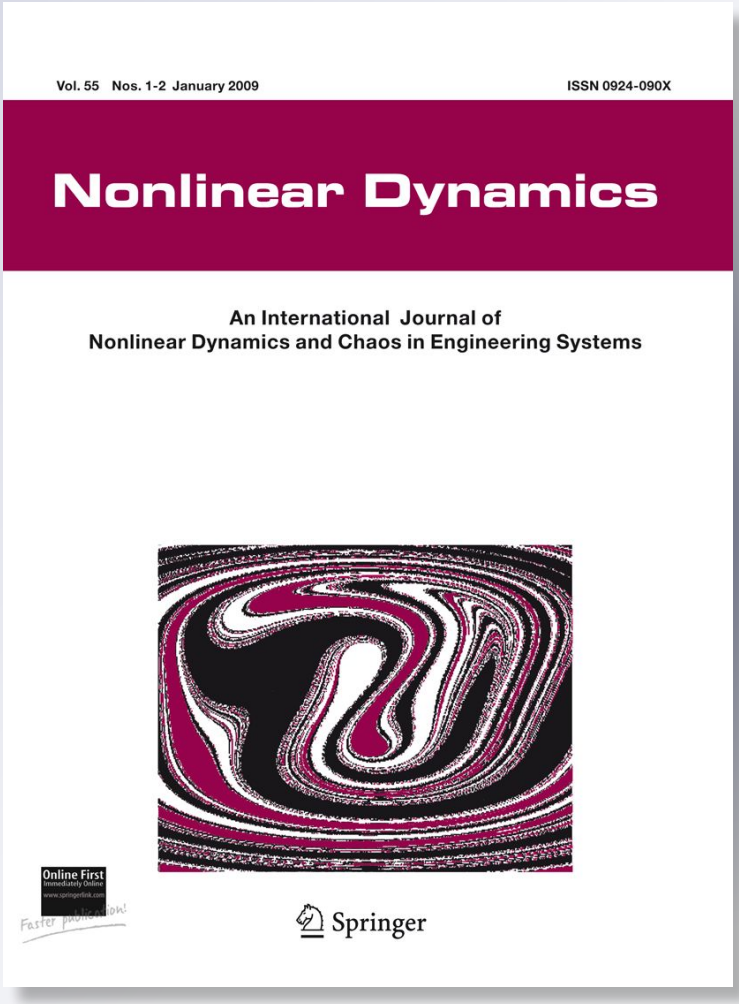
Intelligent tracking control of a DC motor driver using self-organizing TSK-type fuzzy neural networks

Chun-Fei Hsu

Nonlinear Dynamics
An International Journal of Nonlinear Dynamics and Chaos in Engineering Systems

ISSN 0924-090X
Volume 67
Number 1

Nonlinear Dyn (2011) 67:587-600
DOI 10.1007/s11071-011-0012-8



Springer

Nonlinear Dyn (2012) 67:587–600
DOI 10.1007/s11071-011-0012-8

ORIGINAL PAPER

Intelligent tracking control of a DC motor driver using self-organizing TSK-type fuzzy neural networks

Chun-Fei Hsu

Received: 13 December 2010 / Accepted: 3 March 2011 / Published online: 6 April 2011

Abstract In this paper, a self-organizing Takagi–Sugeno–Kang (TSK) type fuzzy neural network (STFNN) is proposed. The self-organizing approach demonstrates the property of automatically generating and pruning the fuzzy rules of STFNN without the preliminary knowledge. The learning algorithms not only extract the fuzzy rule of STFNN but also adjust the parameters of STFNN. Then, an adaptive self-organizing TSK-type fuzzy network controller (ASTFNC) system which is composed of a neural controller and a robust compensator is proposed. The neural controller uses an STFNN to approximate an ideal controller, and the robust compensator is designed to eliminate the approximation error in the Lyapunov stability sense without occurring chattering phenomena. Moreover, a proportional-integral (PI) type parameter tuning mechanism is derived to speed up the convergence rates of the tracking error. Finally, the proposed ASTFNC system is applied to a DC motor driver on a field-programmable gate array chip for low-cost and high-performance industrial applications. The experimental results verify the system stabilization and favorable tracking performance, and no chattering phenomena can be achieved by the proposed ASTFNC scheme.

C.-F. Hsu (✉)
Department of Electrical Engineering, Chung Hua University, Hsinchu, 300, Taiwan, ROC
e-mail: fei@chu.edu.tw

Keywords Adaptive control · Neural control · Fuzzy neural network · Self-organizing · DC motor driver

1 Introduction

Fuzzy system consists of a group of fuzzy IF-THEN rules. To design a fuzzy system automatically, several approaches have been proposed to generate the fuzzy IF-THEN rules from numerical data. This is an active research topic in the area of fuzzy neural networks (FNNs) [1, 2]. The FNN possesses advantages of both fuzzy systems and neural networks. It combines the capability of fuzzy reasoning in handling uncertain information and the capability of artificial neural networks in learning from process. Since the FNNs are universal approximators, the adaptive FNN control schemes have grown rapidly in many previous published papers [3–5]. The most important feature of these adaptive FNN control schemes is the self-learning ability that FNNs are used to approximate arbitrary linear or nonlinear mappings through online learning algorithms without requiring preliminary offline tuning. Convergence analysis of most of the learning algorithms were derived based on the Lyapunov stability theorem or the gradient decent method. So the stability, convergence, and robustness of the closed-loop control system can be improved.

Though a favorable control performance can be achieved by these adaptive FNN controllers in [3–5], the learning process is only parameter learning in

which the parameters of the membership functions and fuzzy rules are adjusted but the structure of FNN is fixed in advance. It is difficult to consider the balance between the number of fuzzy rules and the desired performance. If the number of fuzzy rules is chosen too large, the computation loading is heavy, so they are unsuitable for practical applications. If the number of fuzzy rules is chosen too small, the learning performance may be not good enough to achieve a desired control performance due to the inevitable approximation error. To attack the problem of structure determination in the FNN approaches, many published papers focused on the self-organizing fuzzy neural network (SFNN) approach [6–8]. The self-organizing approach demonstrates the property of automatic generating and pruning fuzzy rules of FNN without the preliminary knowledge. The learning algorithms not only can extract the fuzzy rule but also can adjust the parameters of FNN. Recently, there has been considerable interest in exploring the applications of SFNN to deal with the unknown nonlinear control systems [9–13]. However, some are too complex, some cannot avoid the structure growing unboundedly, and some lack online adaptation ability.

Moreover, since the number of the fuzzy rules in FNN and SFNN is finite for the real-time practical applications, the approximation errors cannot be evitable. To ensure system's stability, a switching compensator was used to dispel the approximation error; however, the switching compensator required the bound of the approximation error, and it will cause chattering phenomena [14]. To reduce the chattering phenomenon, a sign function in the switching compensator can be replaced by a saturation function [14]. However, a trade-off problem between chattering and control accuracy arises. To attack this problem reducing the chattering phenomenon, several compensators were studied [15–18]. Wu et al. [15] presented a smooth compensator to guarantee system stable; however, the tracking error can exponentially converge to a small neighborhood of the trajectory command. Hsu et al. [16] proposed a fuzzy compensator to completely remove the chattering phenomena; however, the adaptive law will make it go to infinity. Some researchers combined the robust control approach to remove the influence of the external disturbance and approximation error in [17, 18]. However, the control effort may lead to a large control signal as a specified attenuation level is chosen small. In this paper, a robust compensator, which is a combination of the proportional controller and the sliding-mode controller using a continuous modulation function, is designed to eliminate the approximation error. This approach integrates their merits of proportional control and sliding-mode control so that the chattering phenomenon can be removed.

According to the network output form, FNNs can be divided into two types, are Mamdani-type FNN and Takagi–Sugeno–Kang (TSK) type FNN [1, 19, 20]. The output weights are equipped with singleton-type form in the Mamdani-type FNN and with functional-type form in the TSK-type FNN. The TSK-type FNN provides a more powerful representation than the Mamdani-type FNN. Recently, the adaptive TSK-type FNN control schemes have grown rapidly in many previous published papers [21–23]. These adaptive TSK-type FNN controllers proposed in [21–23] can achieve a favorable control performance; however, the learning process is only parameter learning, but the structure of FNN is fixed. Based on this observation, design of a self-organizing TSK-type fuzzy neural network (STFNN) should be a better choice. This paper proposes an STFNN which not only has a functional-type form output to provide high learning performance and good generalization capability but also has the property of automatic generating and pruning fuzzy rules without a preliminary knowledge.

An adaptive self-organizing TSK-type fuzzy network control (ASTFNC) system is developed in this paper. The proposed ASTFNC system is composed of a neural controller and a robust compensator. The neural controller uses an STFNN to approximate an ideal controller, and the robust compensator is utilized to eliminate the approximation error between the neural controller and ideal controller without occurring chattering phenomena. Further, to speed up the convergence of the tracking errors, this paper derives the adaptation tuning algorithms in a proportional-integral (PI) type form in the sense of Lyapunov stability. Finally, the proposed ASTFNC system is applied to a DC motor driver, and it is implemented on a field-programmable gate array (FPGA) chip for low-cost and high-performance industrial applications. In the experimental study, it is shown that a high tracking performance and no chattering phenomena can be achieved by the proposed ASTFNC system. Moreover, the proposed self-organizing method demonstrates the properties of generating and pruning the fuzzy rules automatically with a simple computation.

2 DC motor driver and ideal controller

2.1 DC motor driver

The motion equation of a DC motor driver can be simplified as [24, 25]

$$J\ddot{\theta}(t) + B\dot{\theta}(t) = T_e(t) \tag{1}$$

where J is the moment of inertia, B is the damping coefficient, $\theta(t)$ is the rotor position and $T_e(t)$ denotes the electric torque. The electric torque is defined as

$$T_e(t) = K_t i_a(t) \tag{2}$$

where K_t is the torque constant, and $i_a(t)$ is the torque current. The electric equation of a DC motor driver can be simplified as [24, 25]

$$v_a(t) = R_a i_a(t) + K_b\dot{\theta}(t) + L_a\frac{di_a(t)}{dt} \tag{3}$$

where R_a is the motor resistance, K_b is the back electromotive force coefficient, L_a is the motor inductance, and $v_a(t)$ is the DC motor voltage. By considering the motor inductance approximated to zero, the DC motor driver system can be represented in the following form:

$$\ddot{\theta}(t) = f(t) + gu(t) \tag{4}$$

where $f(t) = -(\frac{B}{J} + \frac{K_t K_b}{JR_a})\dot{\theta}(t)$, $g = \frac{K_t}{JR_a}$ is a constant, and $u(t) = v_a(t)$ is the control input.

2.2 Ideal controller design

Rewriting (4), the nominal model of the DC motor driver can be represented as

$$\ddot{\theta}(t) = f_n(t) + g_n u(t) \tag{5}$$

where $f_n(t)$ and g_n are the mappings representing the nominal behavior of $f(t)$ and g, respectively. If uncertainties occur, i.e., the system parameters deviate from their nominal values or an external disturbance is added into the system, the system dynamic can be modified as

$$\begin{aligned}\ddot{\theta}(t) &= \big[f_n(t) + \Delta f(t)\big] + (g_n + \Delta g)u(t) + d(t)\\ &= f_n(t) + g_n u(t) + w(t)\end{aligned} \tag{6}$$

where $\Delta f(t)$ and Δg denote the system uncertainties, $d(t)$ is the external disturbance, and $w(t)$ is called the lumped uncertainty defined as $w(t) = \Delta f(t) + \Delta g u(t) + d(t)$. The control objective is to find a control law so that the rotor position $\theta(t)$ can track a rotor command $\theta_c(t)$. To achieve the control objective, the tracking error is defined as

$$e(t) = \theta_c(t) - \theta(t) \tag{7}$$

and the sliding surface is defined as

$$s(t) = \dot{e}(t) + k_1 e(t) + k_2\int_0^t e(\tau)\,d\tau. \tag{8}$$

If the system uncertainties in (6) are well known, there exists an ideal controller as [26]

$$\begin{aligned}u^*(t) = g_n^{-1}\big[&-f_n(t) - w(t) + \ddot{\theta}_c(t) + k_1\dot{e}(t)\\ &+ k_2 e(t) + Ks(t)\big]\end{aligned} \tag{9}$$

where k_1, k_2, and K are nonzero positive constants. Substituting (9) into (6) yields

$$\ddot{e}(t) + k_1\dot{e}(t) + k_2 e(t) = -Ks(t) = \dot{s}(t). \tag{10}$$

Consider the Lyapunov function candidate in the following form:

$$V_1(t) = \frac{1}{2}s^2(t). \tag{11}$$

Differentiating (11) with respect to time and using (10), we obtain

$$\dot{V}_1(t) = s(t)\dot{s}(t) = -Ks^2(t) \le 0. \tag{12}$$

In summary, the ideal controller can guarantee the stability in the Lyapunov sense [26, 27]. Though a favorable control performance can be achieved by the ideal control system, it needs the dynamic characteristic of the control plant to design the control law. Since the dynamics is usually nonlinear and a precise model is difficult to be obtained, the ideal control system is difficult to be implemented.

3 Design of the ASTFNC system

To efficiently and precisely control the DC motor driver, an adaptive self-organizing TSK-type fuzzy network control (ASTFNC) system is studied. The

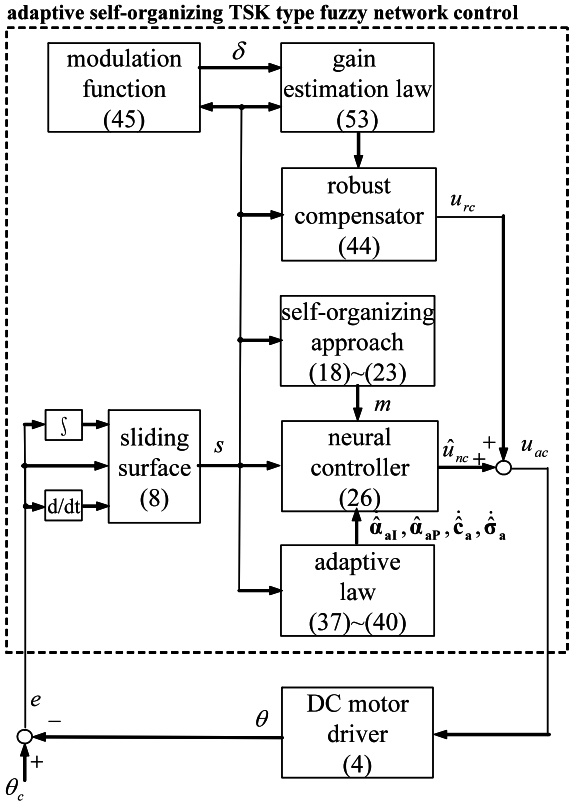


Fig. 1 The block diagram of the ASTFNC system for a DC motor driver

proposed ASTFNC system is composed of a neural controller and a robust compensator as shown in Fig. 1, where the controller output is defined as

$$u_{\mathrm{ac}}(t) = \hat{u}_{\mathrm{nc}}(t) + u_{\mathrm{rc}}(t). \tag{13}$$

The neural controller $\hat{u}_{\mathrm{nc}}(t)$ using an STFNN is designed to approximate the ideal controller, and the robust compensator $u_{\mathrm{rc}}(t)$ is designed to compensate for the difference introduced by the neural controller without occurring chattering phenomena.

3.1 Description of STFNN

The developed STFNN is shown in Fig. 2, and each rule in a TSK-type form is given as [1]

$$\text{Rule } i\text{:} \quad \text{IF } s \text{ is } A_i, \text{ THEN } u_{\mathrm{nc}} = \alpha_{i0} + \alpha_{i1} s \tag{14}$$

where s and u_{nc} are the input and output variables of STFNN, respectively; in the ith fuzzy rule, A_i is the

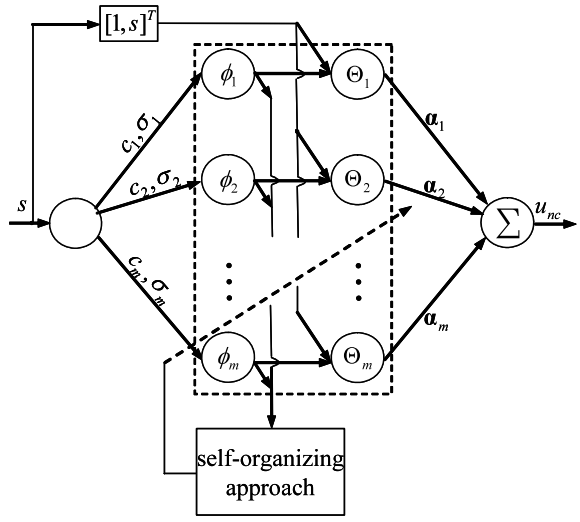


Fig. 2 Self-organizing TSK-type fuzzy neural network

fuzzy set, and α_{i0} and α_{i1} are the adjustable parameters. For the ith fuzzy set A_i, the Gaussian fuzzy set with membership function is used as

$$\phi_i(s) = \exp\left[\frac{-(s - c_i)^2}{\sigma_i^2}\right] \tag{15}$$

where c_i and σ_i denote the center and width of the fuzzy set A_i, respectively. Assuming that there are m rules in STFNN, the output according to the simple weighted sum method would be obtained as

$$u_{\mathrm{nc}} = \sum_{i=1}^{m} \boldsymbol{\alpha}_i^T \mathbf{s} \phi_i(s) \tag{16}$$

where $\boldsymbol{\alpha}_i = [\alpha_{i0}, \alpha_{i1}]^T$ is the parameter vector, and $\mathbf{s} = [1, s]^T$ is the input vector. Then, the output of STFNN can represented in a vector form as

$$u_{\mathrm{nc}} = \boldsymbol{\alpha}^T \boldsymbol{\Theta}(s, \mathbf{c}, \boldsymbol{\sigma}) \tag{17}$$

where $\mathbf{c} = [c_1, c_2, \ldots, c_m]^T$, $\boldsymbol{\sigma} = [\sigma_1, \sigma_2, \ldots, \sigma_m]^T$, $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1^T, \boldsymbol{\alpha}_2^T, \ldots, \boldsymbol{\alpha}_m^T]^T$, and $\boldsymbol{\Theta} = [\phi_1 \mathbf{s}^T, \phi_2 \mathbf{s}^T, \ldots, \phi_m \mathbf{s}^T]^T$.

It is well known that the amount of the fuzzy rules is difficult to select. A trade-off problem between the computation loading and the learning performance arises. A structure learning algorithm including how to generate and prune the fuzzy rules of STFNN is introduced in this paper. The first process of the structure learning is to determine whether to generate a new

fuzzy rule. If a new input data falls within the boundary of clusters, STFNN will not generate a new fuzzy rule but update the parameters of existing fuzzy rules. Consider a distance of mean as [28]

$$d_i = |s - c_i| \quad \text{for } k = 1, 2, \ldots, m. \tag{18}$$

Find the minimum distance as

$$d_{\min} = \min_{1 \le i \le m} d_i. \tag{19}$$

If the minimum distance is large, this means that the input data falls outside of the existing fuzzy rules. It implies that if $d_{\min} \ge d_{\text{th}}$ is satisfied, where d_{th} a pregiven threshold, then a new fuzzy rule should be generated. For the new fuzzy rule, the parameters will be defined as

$$\boldsymbol{\alpha}^{\text{new}} = \mathbf{0} \tag{20}$$

$$c^{\text{new}} = s \tag{21}$$

$$\sigma^{\text{new}} = \bar{\sigma} \tag{22}$$

where $\bar{\sigma}$ is a prespecified value. If the desired performance is the important issue, a small d_{th} should be chosen, so that more fuzzy rules can be generated.

To avoid overload computation loading, consider whether to delete the existing fuzzy rule but is inappropriate. When the kth firing strength Θ_k is smaller than an elimination threshold Θ_{th}, this means that the relationship becomes weak between the input data and the kth firing strength. This fuzzy rule may be less or never used. Then, it will reduce the value of the kth significance index. Based on this observation, a significance index of the kth fuzzy rule is determined as follows [11]:

$$I_k(t+1) = \begin{cases} I_k(t)\exp(-\tau) & \text{if } \Theta_k < \Theta_{\text{th}} \\ I_k(t) & \text{if } \Theta_k \ge \Theta_{\text{th}} \end{cases} \tag{23}$$

where I_k is the significance index of the kth fuzzy rule whose initial value is 1, and τ is the elimination speed constant. If $I_k \le I_{\text{th}}$ is satisfied, where I_{th} a pregiven threshold, then the kth layer will be deleted. For real-time implementation, if the computation load is the important issue, a large I_{th} should be chosen, so that more fuzzy rules can be pruned.

The main property of STFNN regarding feedback control purpose is the universal function approximation property. It implies that there exists an expansion of (17) such it can uniformly approximate an ideal controller $u^*(t)$ as [11]

$$u^*(t) = \boldsymbol{\alpha}^{*T}\boldsymbol{\Theta}(s, \mathbf{c}^*, \boldsymbol{\sigma}^*) + \Delta = \boldsymbol{\alpha}^{*T}\boldsymbol{\Theta}^* + \Delta \tag{24}$$

where Δ is the approximation error, $\boldsymbol{\alpha}^*$ and $\boldsymbol{\Theta}^*$ are the optimal parameter vectors of $\boldsymbol{\alpha}$ and $\boldsymbol{\Theta}$, respectively, and $\mathbf{c}^*$ and $\boldsymbol{\sigma}^*$ are the optimal parameter vectors of $\mathbf{c}$ and $\boldsymbol{\sigma}$, respectively. Let the number of optimal fuzzy rules be m^* and divide it into two parts. The first part contains m neurons that are the parameter vector of the activated part, and the second part contains $m^* - m$ neurons that are the parameter vector of the inactivated parts. Hence, the optimal weights $\boldsymbol{\alpha}^*$, $\boldsymbol{\Theta}^*$, $\mathbf{c}^*$, and $\boldsymbol{\sigma}^*$ are classified in two parts such as [11]

$$\begin{aligned} \boldsymbol{\alpha}^* &= \begin{bmatrix} \boldsymbol{\alpha}_a^* \\ \boldsymbol{\alpha}_i^* \end{bmatrix}, \quad \boldsymbol{\Theta}^* = \begin{bmatrix} \boldsymbol{\Theta}_a^* \\ \boldsymbol{\Theta}_i^* \end{bmatrix}, \\ \mathbf{c}^* &= \begin{bmatrix} \mathbf{c}_a^* \\ \mathbf{c}_i^* \end{bmatrix}, \quad \text{and} \quad \boldsymbol{\sigma}^* = \begin{bmatrix} \boldsymbol{\sigma}_a^* \\ \boldsymbol{\sigma}_i^* \end{bmatrix} \end{aligned} \tag{25}$$

where $\boldsymbol{\alpha}_a^*, \boldsymbol{\Theta}_a^*, \mathbf{c}_a^*$, and $\boldsymbol{\sigma}_a^*$ are activated parts, and $\boldsymbol{\alpha}_i^*, \boldsymbol{\Theta}_i^*, \mathbf{c}_i^*$, and $\boldsymbol{\sigma}_i^*$ are inactivated parts. Since these optimal parameter vectors of the activated parts are unobtainable to best approximation, an estimated STFNN is defined as

$$\hat{u}_{\text{nc}}(t) = \hat{\boldsymbol{\alpha}}_a^T\boldsymbol{\Theta}(s, \hat{\mathbf{c}}_a, \hat{\boldsymbol{\sigma}}_a) = \hat{\boldsymbol{\alpha}}_a^T\hat{\boldsymbol{\Theta}}_a \tag{26}$$

where $\hat{\boldsymbol{\alpha}}_a$, $\hat{\boldsymbol{\Theta}}_a$, $\hat{\mathbf{c}}_a$, and $\hat{\boldsymbol{\sigma}}_a$ are the estimated parameter vectors of $\boldsymbol{\alpha}_a^*$, $\boldsymbol{\Theta}_a^*$, $\mathbf{c}_a^*$, and $\boldsymbol{\sigma}_a^*$, respectively. To speed up the convergence, the optimal parameter vector of the activated parts $\boldsymbol{\alpha}_a^*$ is decomposed into two parts as [29]

$$\boldsymbol{\alpha}_a^* = \eta_P\boldsymbol{\alpha}_{aP}^* + \eta_I\boldsymbol{\alpha}_{aI}^* \tag{27}$$

where $\boldsymbol{\alpha}_{aP}^*$ and $\boldsymbol{\alpha}_{aI}^*$ are the proportional and integral terms of $\boldsymbol{\alpha}_a^*$, respectively, η_P and η_I are positive coefficients, and $\boldsymbol{\alpha}_{aI}^* = \int_0^t \boldsymbol{\alpha}_{aP}^*\, d\tau$. Similarly, the estimation parameter vector $\hat{\boldsymbol{\alpha}}_a$ of the activated parts is decomposed into two parts as

$$\hat{\boldsymbol{\alpha}}_a = \eta_P\hat{\boldsymbol{\alpha}}_{aP} + \eta_I\hat{\boldsymbol{\alpha}}_{aI} \tag{28}$$

where $\hat{\boldsymbol{\alpha}}_{aP}$ and $\hat{\boldsymbol{\alpha}}_{aI}$ are the proportional and integral terms of $\hat{\boldsymbol{\alpha}}_a$, respectively, and $\hat{\boldsymbol{\alpha}}_{aI} = \int_0^t \hat{\boldsymbol{\alpha}}_{aP}\, d\tau$. Thus, $\tilde{\boldsymbol{\alpha}}_a = \boldsymbol{\alpha}_a^* - \hat{\boldsymbol{\alpha}}_a$ can be expressed as

$$\tilde{\boldsymbol{\alpha}}_a = \eta_I\tilde{\boldsymbol{\alpha}}_{aI} - \eta_P\hat{\boldsymbol{\alpha}}_{aP} + \eta_P\boldsymbol{\alpha}_{aP}^* \tag{29}$$

where $\tilde{\boldsymbol{\alpha}}_{aI} = \boldsymbol{\alpha}_{aI}^* - \hat{\boldsymbol{\alpha}}_{aI}$. Define the estimated error $\tilde{u}(t)$ as

$$\begin{aligned}\tilde{u}(t) &= u^*(t) - \hat{u}_{\mathrm{nc}}(t)\\ &= \boldsymbol{\alpha}_a^{*T}\boldsymbol{\Theta}_a^* + \boldsymbol{\alpha}_i^{*T}\boldsymbol{\Theta}_i^* - \hat{\boldsymbol{\alpha}}_a^T\hat{\boldsymbol{\Theta}}_a + \Delta\\ &= \tilde{\boldsymbol{\alpha}}_a^T\hat{\boldsymbol{\Theta}}_a + \hat{\boldsymbol{\alpha}}_a^T\tilde{\boldsymbol{\Theta}}_a + \tilde{\boldsymbol{\alpha}}_a^T\tilde{\boldsymbol{\Theta}}_a + \boldsymbol{\alpha}_i^{*T}\boldsymbol{\Theta}_i^* + \Delta\\ &= (\eta_I\tilde{\boldsymbol{\alpha}}_{aI} - \eta_P\hat{\boldsymbol{\alpha}}_{aP} + \eta_P\boldsymbol{\alpha}_{aP}^*)^T\hat{\boldsymbol{\Theta}}_a\\ &\quad + \hat{\boldsymbol{\alpha}}_a^T\tilde{\boldsymbol{\Theta}}_a + \tilde{\boldsymbol{\alpha}}_a^T\tilde{\boldsymbol{\Theta}}_a + \boldsymbol{\alpha}_i^{*T}\boldsymbol{\Theta}_i^* + \Delta\\ &= \eta_I\tilde{\boldsymbol{\alpha}}_{aI}^T\hat{\boldsymbol{\Theta}}_a - \eta_P\hat{\boldsymbol{\alpha}}_{aP}^T\hat{\boldsymbol{\Theta}}_a + \eta_P\boldsymbol{\alpha}_{aP}^{*T}\hat{\boldsymbol{\Theta}}_a\\ &\quad + \hat{\boldsymbol{\alpha}}_a^T\tilde{\boldsymbol{\Theta}}_a + \tilde{\boldsymbol{\alpha}}_a^T\tilde{\boldsymbol{\Theta}}_a + \boldsymbol{\alpha}_i^{*T}\boldsymbol{\Theta}_i^* + \Delta \qquad (30)\end{aligned}$$

where $\tilde{\boldsymbol{\alpha}}_a = \boldsymbol{\alpha}_a^* - \hat{\boldsymbol{\alpha}}_a$ and $\tilde{\boldsymbol{\Theta}}_a = \boldsymbol{\Theta}_a^* - \hat{\boldsymbol{\Theta}}_a$. The Taylor expansion linearization technique is employed to transform the nonlinear function into a partially linear form [14], i.e.,

$$\tilde{\boldsymbol{\Theta}}_a = \mathbf{A}^T\tilde{\mathbf{c}}_a + \mathbf{B}^T\tilde{\boldsymbol{\sigma}}_a + \mathbf{h} \qquad (31)$$

where $\tilde{\mathbf{c}}_a = \mathbf{c}_a^* - \hat{\mathbf{c}}_a, \tilde{\boldsymbol{\sigma}}_a = \boldsymbol{\sigma}_a^* - \hat{\boldsymbol{\sigma}}_a, \mathbf{h}$ is a vector of high-order terms,

$$\mathbf{A} = \left[\frac{\partial\Theta_1}{\partial\mathbf{c}_a}\ \frac{\partial\Theta_2}{\partial\mathbf{c}_a}\cdots\frac{\partial\Theta_m}{\partial\mathbf{c}_a}\right]\Bigg|_{\mathbf{c}_a=\hat{\mathbf{c}}_a},$$

and

$$\mathbf{B} = \left[\frac{\partial\Theta_1}{\partial\boldsymbol{\sigma}_a}\ \frac{\partial\Theta_2}{\partial\boldsymbol{\sigma}_a}\cdots\frac{\partial\Theta_m}{\partial\boldsymbol{\sigma}_a}\right]\Bigg|_{\boldsymbol{\sigma}_a=\hat{\boldsymbol{\sigma}}_a}.$$

Substitution of (31) into (30) yields

$$\begin{aligned}\tilde{u}(t) &= \eta_I\tilde{\boldsymbol{\alpha}}_{aI}^T\hat{\boldsymbol{\Theta}}_a - \eta_P\hat{\boldsymbol{\alpha}}_{aP}^T\hat{\boldsymbol{\Theta}}_a + \eta_P\boldsymbol{\alpha}_{aP}^{*T}\hat{\boldsymbol{\Theta}}_a\\ &\quad + \hat{\boldsymbol{\alpha}}_a^T\left(\mathbf{A}^T\tilde{\mathbf{c}}_a + \mathbf{B}^T\tilde{\boldsymbol{\sigma}}_a + \mathbf{h}\right) + \tilde{\boldsymbol{\alpha}}_a^T\tilde{\boldsymbol{\Theta}}_a\\ &\quad + \boldsymbol{\alpha}_i^{*T}\boldsymbol{\Theta}_i^* + \Delta\\ &= \eta_I\tilde{\boldsymbol{\alpha}}_{aI}^T\hat{\boldsymbol{\Theta}}_a - \eta_P\hat{\boldsymbol{\alpha}}_{aP}^T\hat{\boldsymbol{\Theta}}_a + \tilde{\mathbf{c}}_a^T\mathbf{A}\hat{\boldsymbol{\alpha}}_a\\ &\quad + \tilde{\boldsymbol{\sigma}}_a^T\mathbf{B}\hat{\boldsymbol{\alpha}}_a + \varepsilon \qquad (32)\end{aligned}$$

where $\hat{\boldsymbol{\alpha}}_a^T\mathbf{A}^T\tilde{\mathbf{c}}_a = \tilde{\mathbf{c}}_a^T\mathbf{A}\hat{\boldsymbol{\alpha}}_a$ and $\hat{\boldsymbol{\alpha}}_a^T\mathbf{B}^T\tilde{\boldsymbol{\sigma}}_a = \tilde{\boldsymbol{\sigma}}_a^T\mathbf{B}\hat{\boldsymbol{\alpha}}_a$ are used since they are scalars, and $\varepsilon = \hat{\boldsymbol{\alpha}}_a^T\mathbf{h} + \tilde{\boldsymbol{\alpha}}_a^T\tilde{\boldsymbol{\Theta}}_a + \boldsymbol{\alpha}_i^{*T}\boldsymbol{\Theta}_i^* + \eta_P\boldsymbol{\alpha}_{aP}^{*T}\hat{\boldsymbol{\Theta}}_a + \Delta$ denotes the lump of approximation error which is assumed to be bounded by $0 \le |\varepsilon| \le E$ with E is a positive constant.

3.2 ASTFNC system design without a robust compensator

Substituting (13) into (6) and using (9) yields

$$\begin{aligned}&\ddot{e}(t) + k_1\dot{e}(t) + k_2e(t)\\ &= g_n\left[u^*(t) - \hat{u}_{\mathrm{nc}}(t) - u_{\mathrm{rc}}(t)\right] - Ks(t)\\ &= \dot{s}(t). \qquad (33)\end{aligned}$$

Using the approximation property (32), (33) can be rewritten as

$$\begin{aligned}\dot{s}(t) = g_n\big[&\eta_I\tilde{\boldsymbol{\alpha}}_a^T\hat{\boldsymbol{\Theta}}_a - \eta_P\hat{\boldsymbol{\alpha}}_{aP}^T\hat{\boldsymbol{\Theta}}_a + \tilde{\mathbf{c}}_a^T\mathbf{A}\hat{\boldsymbol{\alpha}}_a + \tilde{\boldsymbol{\sigma}}_a^T\mathbf{B}\hat{\boldsymbol{\alpha}}_a\\ &+ \varepsilon - u_{\mathrm{rc}}(t)\big] - Ks(t). \qquad (34)\end{aligned}$$

To prove the stability of the ASTFNC system without a robust compensator, define a Lyapunov function candidate in the following form:

$$\begin{aligned}V_2(t) = \frac{1}{2}s^2(t) + g_n\Bigg(&\frac{\eta_I}{2}\tilde{\boldsymbol{\alpha}}_{aI}^T\tilde{\boldsymbol{\alpha}}_{aI} + \frac{1}{2\eta_c}\tilde{\mathbf{c}}_a^T\tilde{\mathbf{c}}_a\\ &+ \frac{1}{2\eta_\sigma}\tilde{\boldsymbol{\sigma}}_a^T\tilde{\boldsymbol{\sigma}}_a\Bigg) \qquad (35)\end{aligned}$$

where η_c and η_σ are the positive learning rates. Differentiating (35) with respect to time and using (34), we obtain

$$\begin{aligned}\dot{V}_2(t) &= s(t)\dot{s}(t) + \eta_I g_n\tilde{\boldsymbol{\alpha}}_{aI}^T\dot{\tilde{\boldsymbol{\alpha}}}_{aI} + \frac{g_n}{\eta_c}\tilde{\mathbf{c}}_a^T\dot{\tilde{\mathbf{c}}}_a + \frac{g_n}{\eta_\sigma}\tilde{\boldsymbol{\sigma}}_a^T\dot{\tilde{\boldsymbol{\sigma}}}_a\\ &= g_ns(t)\big[\eta_I\tilde{\boldsymbol{\alpha}}_{aI}^T\hat{\boldsymbol{\Theta}}_a - \eta_P\hat{\boldsymbol{\alpha}}_{aP}^T\hat{\boldsymbol{\Theta}}_a + \tilde{\mathbf{c}}_a^T\mathbf{A}\hat{\boldsymbol{\alpha}}_a\\ &\quad + \tilde{\boldsymbol{\sigma}}_a^T\mathbf{B}\hat{\boldsymbol{\alpha}}_a + \varepsilon\big] - Ks^2(t)\\ &\quad + \eta_I g_n\tilde{\boldsymbol{\alpha}}_{aI}^T\dot{\tilde{\boldsymbol{\alpha}}}_{aI} + \frac{g_n}{\eta_c}\tilde{\mathbf{c}}_a^T\dot{\tilde{\mathbf{c}}}_a + \frac{g_n}{\eta_\sigma}\tilde{\boldsymbol{\sigma}}_a^T\dot{\tilde{\boldsymbol{\sigma}}}_a\\ &= \eta_I g_n\tilde{\boldsymbol{\alpha}}_{aI}^T\left[s(t)\hat{\boldsymbol{\Theta}}_a + \dot{\tilde{\boldsymbol{\alpha}}}_{aI}\right]\\ &\quad + g_n\tilde{\mathbf{c}}_a^T\left[s(t)\mathbf{A}\hat{\boldsymbol{\alpha}}_a + \frac{\dot{\tilde{\mathbf{c}}}_a}{\eta_c}\right]\\ &\quad + g_n\tilde{\boldsymbol{\sigma}}_a^T\left[s(t)\mathbf{B}\hat{\boldsymbol{\alpha}}_a + \frac{\dot{\tilde{\boldsymbol{\sigma}}}_a}{\eta_\sigma}\right]\\ &\quad + g_ns(t)\big[-\eta_P\hat{\boldsymbol{\alpha}}_{aP}^T\hat{\boldsymbol{\Theta}}_a + \varepsilon\big] - Ks^2(t). \qquad (36)\end{aligned}$$

Choose the adaptive laws as

$$\hat{\boldsymbol{\alpha}}_{aP} = s(t)\hat{\boldsymbol{\Theta}}_a \qquad (37)$$

$$\dot{\hat{\boldsymbol{\alpha}}}_{aI} = -\dot{\tilde{\boldsymbol{\alpha}}}_{aI} = s(t)\hat{\boldsymbol{\Theta}}_a \qquad (38)$$

$$\dot{\hat{\mathbf{c}}}_a = -\dot{\tilde{\mathbf{c}}}_a = \eta_c s(t)\mathbf{A}\hat{\boldsymbol{\alpha}}_a \tag{39}$$

$$\dot{\hat{\boldsymbol{\sigma}}}_a = -\dot{\tilde{\boldsymbol{\sigma}}}_a = \eta_\sigma s(t)\mathbf{B}\hat{\boldsymbol{\alpha}}_a \tag{40}$$

so that (36) can be rewritten as

$$\begin{aligned}\dot{V}_2(t) &= g_n\varepsilon s(t) - Ks^2(t) - \eta_P g_n \hat{\boldsymbol{\alpha}}_{aP}^T\hat{\boldsymbol{\alpha}}_{aP} \\ &\le g_n|\varepsilon|\big|s(t)\big| - Ks^2(t) \\ &\le g_n E\big|s(t)\big| - Ks^2(t) \\ &= -K\left(\big|s(t)\big| - \frac{g_n E}{2K}\right)^2 + \frac{g_n^2E^2}{4K}.\end{aligned} \tag{41}$$

Define the set

$$\Omega_s(E) = \left\{s(t) : K\left(\big|s(t)\big| - \frac{g_n E}{2K}\right)^2 \le \frac{g_n^2E^2}{4K}\right\}. \tag{42}$$

If $s(t) \notin \Omega_s(E)$, $\dot{V}_2(t)$ is still negative, and the tracking error converges to the set $\Omega_s(E)$. But, if $s(t) \in \Omega_s(E)$, it is possible $\dot{V}_2(t) > 0$, which implies that $\hat{\boldsymbol{\alpha}}_{aP}, \hat{\boldsymbol{\alpha}}_{aI}, \hat{\mathbf{c}}_a$, and $\hat{\boldsymbol{\sigma}}_a$ may drift to infinity over time.

3.3 ASTFNC system design with a robust compensator

There are three different regions of $s(t)$ divided by $\Omega_s(E)$ and $\Omega_s(E,\Phi)$ as shown in Fig. 3. The set $\Omega_s(E,\Phi)$ is defined as

$$\begin{aligned}&\Omega_s(E,\Phi) \\ &= \left\{s(t) : K\left(\big|s(t)\big| - \frac{g_n E}{2K}\right)^2 \le \frac{g_n^2E^2}{4K} + \Phi\right\}\end{aligned} \tag{43}$$

where Φ is a small positive constant representing the width of the transition region $\Omega_s(E,\Phi)$. The proposed robust compensator is designed as

$$u_{\mathrm{rc}} = \delta\hat{K}_p s(t) + (1-\delta)E\,\mathrm{sgn}\big[s(t)\big] \tag{44}$$

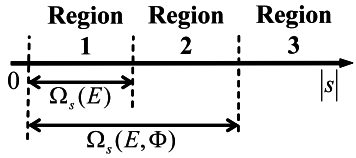


Fig. 3 The three different regions divided by $\Omega_s(E)$ and $\Omega_s(E,\Phi)$

where $\hat{K}_p$ is a positive coefficient designed by the designer, and the modulation function $\delta \in [0,1]$ is chosen as

$$\delta = \begin{cases} 1 & \text{if } s(t) \in \Omega_s(E) \\ 1 - \dfrac{K(|s(t)| - \frac{g_n E}{2K})^2 - \frac{g_n^2E^2}{4K}}{\Phi} & \text{if } s(t) \notin \Omega_s(E) \text{ and } s(t) \in \Omega_s(E,\Phi) \\ 0 & \text{if } s(t) \notin \Omega_s(E,\Phi). \end{cases} \tag{45}$$

To prove the stability of the ASTFNC system with a robust compensator, consider the Lyapunov function candidate in the following form:

$$\begin{aligned}V_3(t) = \frac{1}{2}s^2(t) + g_n\Bigg(&\frac{\eta_I}{2}\tilde{\boldsymbol{\alpha}}_{aI}^T\tilde{\boldsymbol{\alpha}}_{aI} + \frac{1}{2\eta_c}\tilde{\mathbf{c}}_a^T\tilde{\mathbf{c}}_a \\ &+ \frac{1}{2\eta_\sigma}\tilde{\boldsymbol{\sigma}}_a^T\tilde{\boldsymbol{\sigma}}_a\Bigg).\end{aligned} \tag{46}$$

Differentiating (46) with respect to time and using (34) and (44), we obtain

$$\begin{aligned}\dot{V}_3(t) &= s(t)\dot{s}(t) + \eta_I g_n\tilde{\boldsymbol{\alpha}}_{aI}^T\dot{\tilde{\boldsymbol{\alpha}}}_{aI} + \frac{g_n}{\eta_c}\tilde{\mathbf{c}}_a^T\dot{\tilde{\mathbf{c}}}_a + \frac{g_n}{\eta_\sigma}\tilde{\boldsymbol{\sigma}}_a^T\dot{\tilde{\boldsymbol{\sigma}}}_a \\ &= g_n s(t)\big(\eta_I\tilde{\boldsymbol{\alpha}}_{aI}^T\hat{\boldsymbol{\Theta}}_a - \eta_P\hat{\boldsymbol{\alpha}}_{aP}^T\hat{\boldsymbol{\Theta}}_a + \tilde{\mathbf{c}}_a^T\mathbf{A}\hat{\boldsymbol{\alpha}}_a \\ &\quad + \tilde{\boldsymbol{\sigma}}_a^T\mathbf{B}\hat{\boldsymbol{\alpha}}_a + \varepsilon - u_{\mathrm{rc}}\big) - Ks^2(t) \\ &\quad + \eta_I g_n\tilde{\boldsymbol{\alpha}}_{aI}^T\dot{\tilde{\boldsymbol{\alpha}}}_{aI} + \frac{g_n}{\eta_c}\tilde{\mathbf{c}}_a^T\dot{\tilde{\mathbf{c}}}_a + \frac{g_n}{\eta_\sigma}\tilde{\boldsymbol{\sigma}}_a^T\dot{\tilde{\boldsymbol{\sigma}}}_a \\ &= \eta_I g_n\tilde{\boldsymbol{\alpha}}_{aI}^T\big[s(t)\hat{\boldsymbol{\Theta}}_a + \dot{\tilde{\boldsymbol{\alpha}}}_{aI}\big] \\ &\quad + g_n\tilde{\mathbf{c}}_a^T\left[s(t)\mathbf{A}\hat{\boldsymbol{\alpha}}_a + \frac{\dot{\tilde{\mathbf{c}}}_a}{\eta_c}\right] \\ &\quad + g_n\tilde{\boldsymbol{\sigma}}_a^T\left[s(t)\mathbf{B}\hat{\boldsymbol{\alpha}}_a + \frac{\dot{\tilde{\boldsymbol{\sigma}}}_a}{\eta_\sigma}\right] \\ &\quad + g_n s(t)\big(-\eta_P\hat{\boldsymbol{\alpha}}_{aP}^T\hat{\boldsymbol{\Theta}}_a + \varepsilon - u_{\mathrm{rc}}\big) - Ks^2(t) \\ &\le g_n s(t)\big[\varepsilon - \delta\hat{K}_p s(t) - (1-\delta)E\,\mathrm{sgn}\big(s(t)\big)\big] \\ &\quad - Ks^2(t) \\ &\le g_n E\big|s(t)\big| - \delta g_n\hat{K}_p s^2(t) - (1-\delta)g_n E\big|s(t)\big| \\ &\quad - Ks^2(t) \\ &= \delta g_n E\big|s(t)\big| - \delta g_n\hat{K}_p s^2(t) - Ks^2(t) \\ &= -\delta g_n\big(\hat{K}_p\big|s(t)\big| - E\big)\big|s(t)\big| - Ks^2(t).\end{aligned} \tag{47}$$

If the inequality

$$\hat{K}_p > \frac{E}{|s(t)|} \tag{48}$$

holds, then $\dot{V}_3(t) \leq 0$ can be satisfied. Owing to the unknown approximation error bound E, the value $\hat{K}_p$ cannot be exactly obtained in advance for practical applications. According to (48), there exists the following ideal value K_p^*:

$$K_p^* = \frac{E + r}{|s(t)|} \tag{49}$$

where r is a positive constant. Thus, a simple adaptive algorithm is utilized to estimate the ideal value of K_p^*, and its estimated error is defined as

$$\tilde{K}_p = K_p^* - \hat{K}_p. \tag{50}$$

Then, consider the Lyapunov function candidate in the following form:

$$V_4(t) = \frac{1}{2}s^2(t) + g_n\left(\frac{\eta_I}{2}\tilde{\boldsymbol{\alpha}}_{aI}^T\tilde{\boldsymbol{\alpha}}_{aI} + \frac{1}{2\eta_c}\tilde{\mathbf{c}}_a^T\tilde{\mathbf{c}}_a + \frac{1}{2\eta_\sigma}\tilde{\boldsymbol{\sigma}}_a^T\tilde{\boldsymbol{\sigma}}_a + \frac{1}{2\eta_k}\tilde{K}_p^2\right) \tag{51}$$

where η_k is the positive learning rate. Differentiating (51) with respect to time and using (34), (37)–(40), (44), and (49), we obtain

$$\begin{aligned}
\dot{V}_4(t) &= s(t)\dot{s}(t) + \eta_I g_n\tilde{\boldsymbol{\alpha}}_{aI}^T\dot{\tilde{\boldsymbol{\alpha}}}_{aI} + \frac{g_n}{\eta_c}\tilde{\mathbf{c}}_a^T\dot{\tilde{\mathbf{c}}}_a + \frac{g_n}{\eta_\sigma}\tilde{\boldsymbol{\sigma}}_a^T\dot{\tilde{\boldsymbol{\sigma}}}_a + \frac{g_n}{\eta_k}\tilde{K}_p\dot{\tilde{K}}_p \\
&= g_n s(t)\big(\eta_I\tilde{\boldsymbol{\alpha}}_{aI}^T\hat{\boldsymbol{\Theta}}_a - \eta_P\hat{\boldsymbol{\alpha}}_{aP}^T\hat{\boldsymbol{\Theta}}_a + \tilde{\mathbf{c}}_a^T\mathbf{A}\hat{\boldsymbol{\alpha}}_a + \tilde{\boldsymbol{\sigma}}_a^T\mathbf{B}\hat{\boldsymbol{\alpha}}_a + \varepsilon - u_{\mathrm{rc}}\big) - Ks^2(t) + \eta_I\tilde{\boldsymbol{\alpha}}_{aI}^T\dot{\tilde{\boldsymbol{\alpha}}}_{aI} \\
&\quad + \frac{g_n}{\eta_c}\tilde{\mathbf{c}}_a^T\dot{\tilde{\mathbf{c}}}_a + \frac{g_n}{\eta_\sigma}\tilde{\boldsymbol{\sigma}}_a^T\dot{\tilde{\boldsymbol{\sigma}}}_a + \frac{g_n}{\eta_k}\tilde{K}_p\dot{\tilde{K}}_p \\
&= \eta_I g_n\tilde{\boldsymbol{\alpha}}_{aI}^T\big[s(t)\hat{\boldsymbol{\Theta}}_a + \dot{\tilde{\boldsymbol{\alpha}}}_{aI}\big] + g_n\tilde{\mathbf{c}}_a^T\left[s(t)\mathbf{A}\hat{\boldsymbol{\alpha}}_a + \frac{\dot{\tilde{\mathbf{c}}}_a}{\eta_c}\right] + g_n\tilde{\boldsymbol{\sigma}}_a^T\left[s(t)\mathbf{B}\hat{\boldsymbol{\alpha}}_a + \frac{\dot{\tilde{\boldsymbol{\sigma}}}_a}{\eta_\sigma}\right] \\
&\quad + g_n s(t)\big(-\eta_P\hat{\boldsymbol{\alpha}}_{aP}^T\hat{\boldsymbol{\Theta}}_a + \varepsilon - u_{\mathrm{rc}}\big) - Ks^2(t) + \frac{g_n}{\eta_k}\tilde{K}_p\dot{\tilde{K}}_p \\
&\leq g_n s(t)\big[\varepsilon - \delta\hat{K}_p s(t) - (1-\delta)E\,\mathrm{sgn}\big(s(t)\big)\big] - Ks^2(t) + \frac{g_n}{\eta_k}\tilde{K}_p\dot{\tilde{K}}_p
\end{aligned}$$

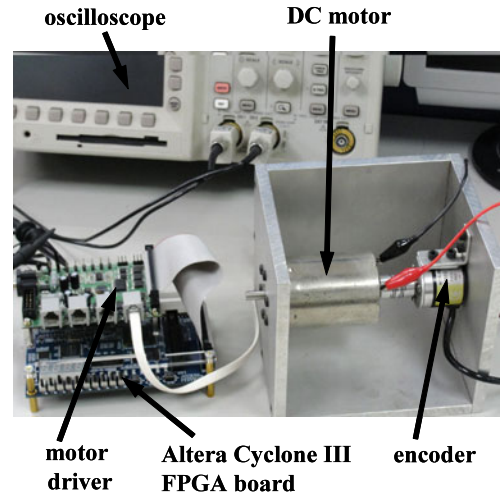


Fig. 4 The FPGA-based experimental setup

$$\begin{aligned}
&\leq \delta g_n E\big|s(t)\big| - \delta g_n\hat{K}_p s^2(t) - Ks^2(t) + \frac{g_n}{\eta_k}\tilde{K}_p\dot{\tilde{K}}_p.
\end{aligned} \tag{52}$$

Choosing the gain estimation law as

$$\dot{\hat{K}}_p = -\dot{\tilde{K}}_p = \eta_k\delta\big|s(t)\big|^2 \tag{53}$$

(52) becomes

$$\begin{aligned}
\dot{V}_4(t) &\leq \delta g_n E\big|s(t)\big| - \delta g_n K_p^* s^2(t) - Ks^2(t) \\
&= -\delta g_n r|s| - Ks^2(t) \\
&\leq -Ks^2(t) \leq 0.
\end{aligned} \tag{54}$$

Since $\dot{V}_4(t)$ is negative semidefinite, that is, $V_4(t) \leq V_4(0)$, it follows that $s(t), \tilde{\boldsymbol{\alpha}}_{aI}, \tilde{\mathbf{c}}_a, \tilde{\boldsymbol{\sigma}}_a$, and $\tilde{K}_p$ are bounded. Let the function $\psi(t) \equiv Ks^2(t) \leq -\dot{V}_4(t)$, and integrate $\psi(t)$ with respect to time. Then we obtain

$$\int_0^t \psi(\tau)\,d\tau \leq V_4(0) - V_4(t). \tag{55}$$

Because $V_4(0)$ is bounded and $V_4(t)$ is nonincreasing and bounded, we obtains that

$$\lim_{t\to\infty}\int_0^t \psi(\tau)\,d\tau < \infty. \tag{56}$$

By Barbalat's lemma [26], $\lim_{t\to\infty}\psi(t) = 0$. That is, $s(t) \to 0$ as $t \to \infty$. As a result, the stability of the ASTFNC system with a robust compensator can be guaranteed.

4 Experimental results

FPGA is a fast prototyping integrated circuit (IC) component. This kind of IC incorporates the design of a gate array and the programmability of a programmable logic device. It consists of thousands of logic gates, some of which are combined together to form a configurable logic block, thereby simplifying high-level circuit design [30, 31]. The advantage of a controller implemented on an FPGA chip includes shorter development cycles, lower cost, small size, fast system execute speed, and high flexibility. The FPGA-based experimental setup is shown in Fig. 4. This study uses the Altera Cyclone III series FPGA chip and the circuits, and algorithms are developed in the VHSIC hardware description language (VHDL). To investigate the effectiveness of the proposed ASTFNC system, the adaptive FNN control with a switching compensator [3]

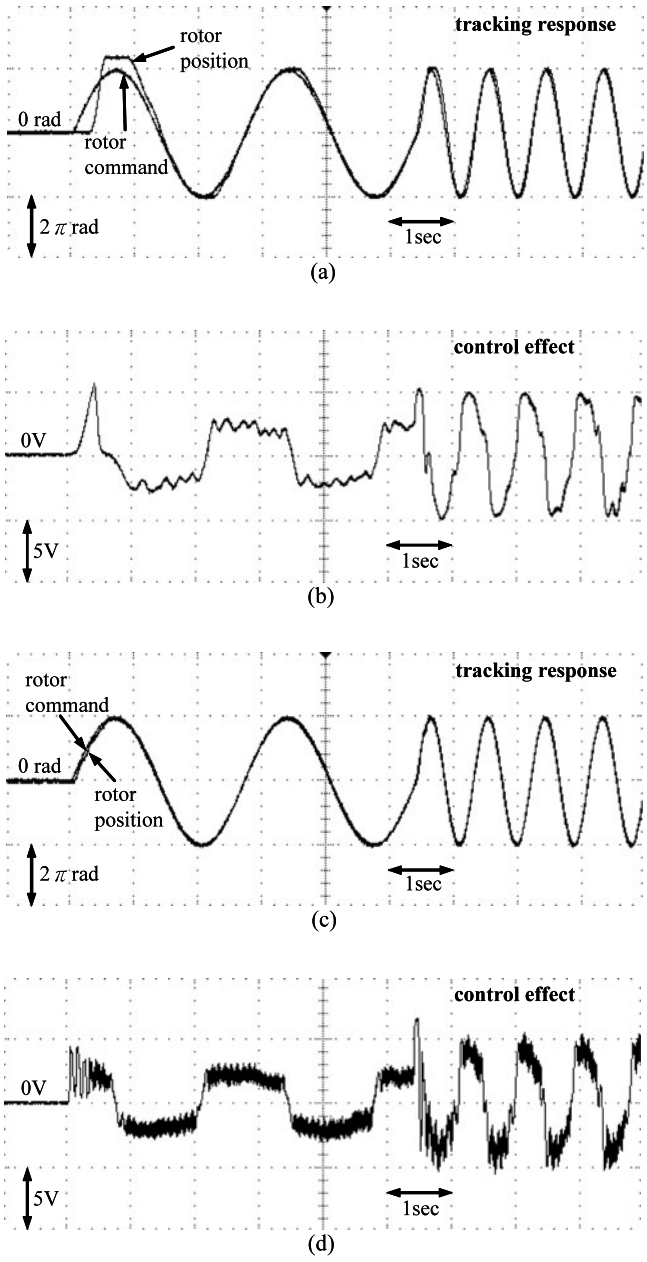


Fig. 5 Experimental results of the adaptive FNN control [3]

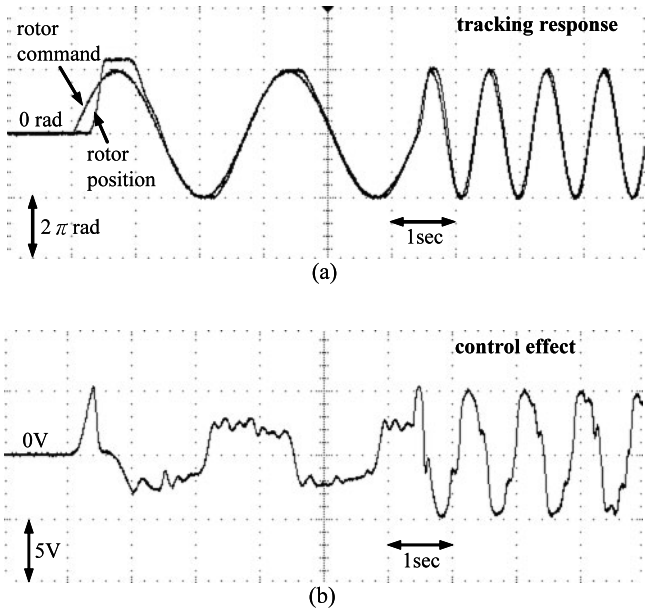


Fig. 6 Experimental results of the robust adaptive FNN control [30]

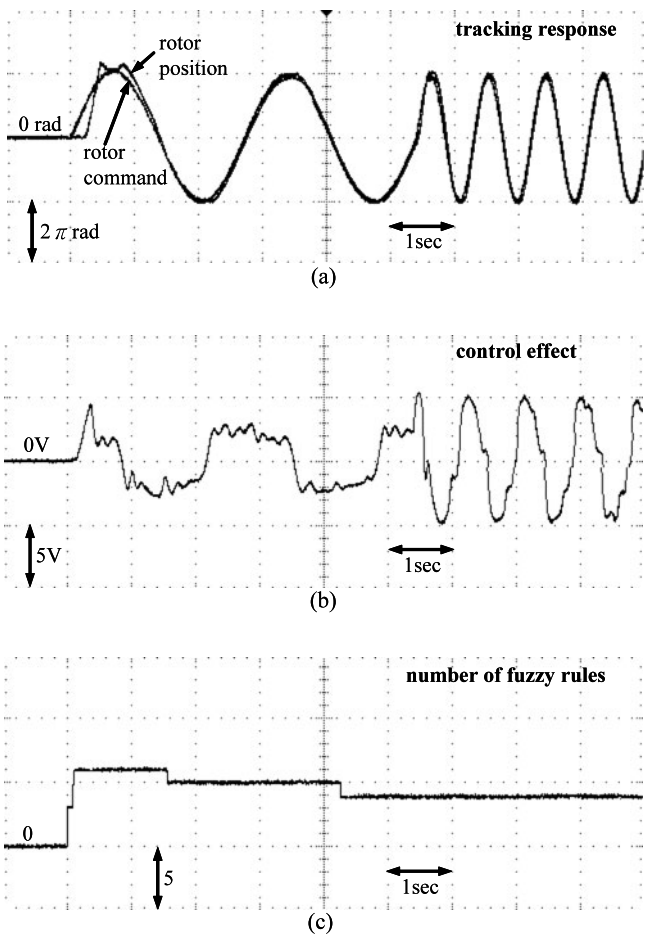


Fig. 7 Experimental results of the ASTFNC system using integral-type learning algorithm

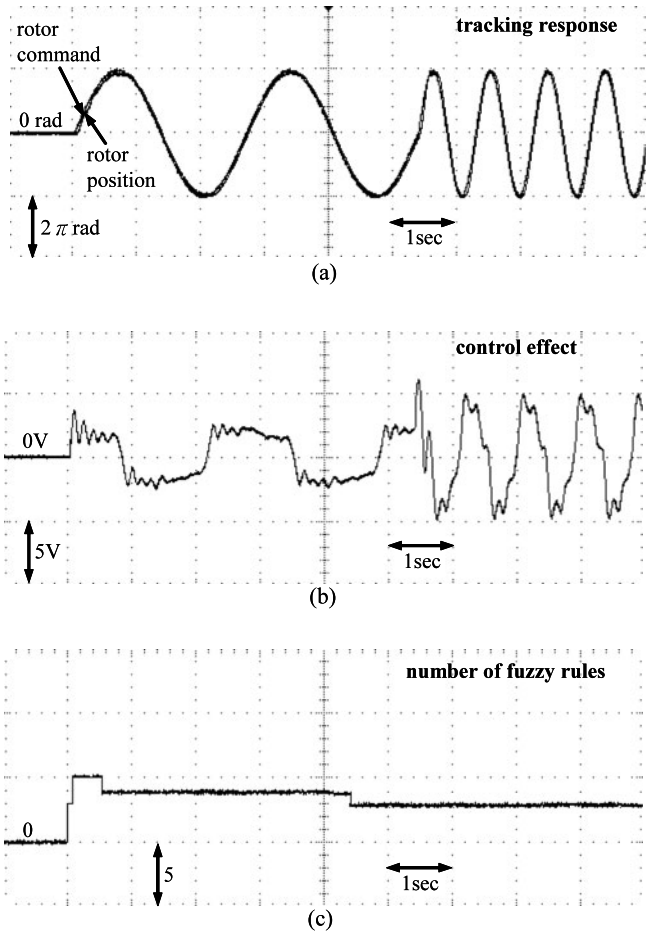


Fig. 8 Experimental results of the ASTFNC system using PI-type learning algorithm

and the robust adaptive FNN control with a supervisor compensator [32] are considered for comparison.

First, the adaptive FNN controller with a switching compensator [3] is applied to the DC motor driver. The switching compensator required the bound of the approximation error. The experimental results of the adaptive FNN control with a switching compensator are shown in Fig. 5. When the bound of approximation error is chosen small, the tracking response is shown in Fig. 5(a), and the associated control effort is shown in Fig. 5(b). When the bound of approximation error is chosen large, the tracking response is shown in Fig. 5(c), and the associated control effort is shown in Fig. 5(d). The experimental results show that a favorable control performance can be achieved after the controller parameters become well learned. Unfortunately, to guarantee the system stability, a switching compensator should be used, but the undesirable chattering phenomenon occurs as shown in Fig. 5(d). A trade-off problem between chattering phenomenon and control accuracy arises.

Then, the robust adaptive FNN controller with a supervisor compensator [32] is applied to the DC motor driver again. The supervisory compensator combined the sliding-mode control and the adaptive control by using a modulation function such that it could take the advantage of the robust and adaptive properties to deal with the approximation error. The experimental results of the robust adaptive FNN controller with a supervisor compensator are shown in Fig. 6. The tracking response is shown in Fig. 6(a), and the associated control effort is shown in Fig. 6(b). The experimental results show that the favorable control performance can be achieved and the chattering phenomena of control efforts can be removed. However, the convergence of the tracking error is slow since the parameter adaptation law was designed in an integral type form.

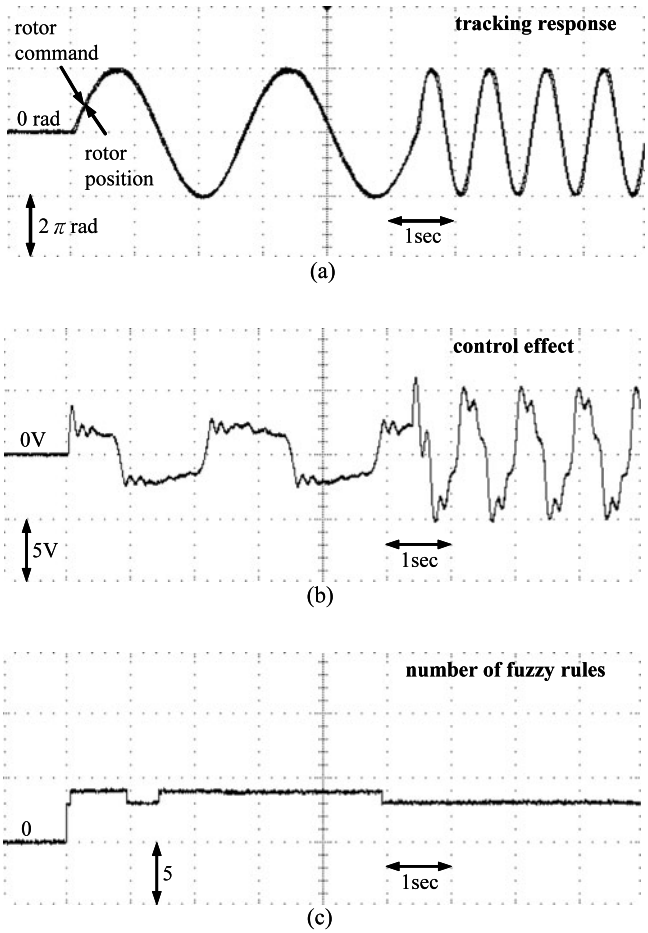


Fig. 9 Experimental results of the trained ASTFNC system using PI-type learning algorithm

Finally, the proposed ASTFNC system is applied to the DC motor driver again. The control parameters are selected as $k_1 = 2$, $k_2 = 1$, $\eta_I = 1$, $\eta_P = \eta_c = \eta_\sigma = \eta_k = 0.1$, $K = 1$, $E = 1$, $\Phi = 1$, $d_{\text{th}} = 0.3$, $I_{\text{th}} = 0.1$, $\Theta_{\text{th}} = 0.2$, $\tau = 0.01$, and $\bar{\sigma} = 1.0$. These parameters are selected through some trails. By choosing the values of k_1 and k_2 properly, the desired system dynamics such as settling time can be easily designed by the second-order system. The parameters η_P, η_I, η_c, and η_σ are the leaning rates of the neural control, and the parameter η_k is the leaning rate of the robust compensator. If the leaning rates are chosen small, the parameter convergence will be easily achieved; however, this will result in slow learning speed. On the other hand, if the leaning rates are chosen large, the learning speed will be fast; however, the system may become unstable.

To compare the tracking efficiency, the ASTFNC system with using integral-type learning algorithm is applied first. This is a special case of the developed ASTFNC scheme for $\eta_P = 0$. The experimental results of the ASTFNC system with using integral-type learning algorithm are shown in Fig. 7. The tracking response is shown in Fig. 7(a), the associated control effort is shown in Fig. 7(b), and the number of fuzzy rule is shown in Fig. 7(c). From the experimental results, not only the perfect tracking responses can be achieved without any chattering phenomena occurring, but also a concise network structure can be obtained by the proposed self-organizing method. However, the convergence of the tracking error is slow.

To speed up the convergence, the experimental results of the ASTFNC system with using PI-type learning algorithm are shown in Fig. 8. The tracking response is shown in Fig. 8(a), the associated control effort is shown in Fig. 8(b), and the number of fuzzy rule is shown in Fig. 8(c). It is shown that not only the perfect tracking responses can be achieved with-

out any chattering phenomena occurring, but also the convergent speed of the tracking error can speed up because the proposed PI-type adaptation learning algorithm is applied. Further, the trained ASTFNC system with using PI-type learning algorithm is applied to the DC motor driver again. The experimental results of the trained ASTFNC system using PI-type learning algorithm are shown in Fig. 9. The tracking response is shown in Fig. 9(a), the associated control effort is shown in Fig. 9(b), and the number of fuzzy rule is shown in Fig. 9(c). The experimental results show that the system stabilization, favorable tracking performance, and no chattering phenomena can be achieved using the proposed ASTFNC system with using PI-type learning algorithm.

5 Conclusions

This paper develops a novel self-organizing TSK-type fuzzy neural network (STFNN). The STFNN demonstrates the properties of generating and pruning the fuzzy rules automatically. Then, an adaptive self-organizing Takagi–Sugeno–Kang (TSK) type fuzzy network control (ASTFNC) system is proposed for a DC motor driver. To speed up the convergence of the tracking error, a proportional-integral (PI) type adaptation tuning mechanism is derived in the Lyapunov stability sense. Finally, this paper has successfully implemented the proposed ASTFNC system in the VHSIC hardware description language on a field programmable gate array chip. To investigate the effectiveness of the proposed ASTFNC system, a comparison among the adaptive FNN control with a switching compensator, the robust adaptive FNN control with a supervisor compensator, and the proposed ASTFNC is made. It is verified by the experimental study that the developed model-free ASTFNC system with using PI-type learning algorithm is more suitable for a DC motor driver.

Acknowledgements The authors are grateful to the reviewers for their valuable comments. The authors appreciate the partial financial support from the National Science Council of Republic of China under grant NSC 99-2628-E-216-002.

References

1. Lin, C.T., Lee, C.S.G.: Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems. Prentice-Hall, Englewood Cliffs (1996)
2. Jang, R., Sun, C.T., Mizutani, E.: Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence. Prentice-Hall, Englewood Cliffs (1997)
3. Lin, C.M., Hsu, C.F.: Supervisory recurrent fuzzy neural network control of wing rock for slender delta wings. IEEE Trans. Fuzzy Syst. **12**(5), 733–742 (2004)
4. Leu, Y.G., Wang, W.Y., Lee, T.T.: Observer-based direct adaptive fuzzy-neural control for nonaffine nonlinear systems. IEEE Trans. Neural Netw. **16**(4), 853–861 (2005)
5. Cheng, K.H., Hsu, C.F., Lin, C.M., Lee, T.T., Li, C.: Fuzzy-neural sliding-mode control for DC–DC converters using asymmetric Gaussian membership functions. IEEE Trans. Ind. Electron. **54**(3), 1528–1536 (2007)
6. Li, C., Lee, C.Y., Cheng, K.H.: Pseudo-error-based self-organizing neuro-fuzzy system. IEEE Trans. Fuzzy Syst. **12**(6), 812–819 (2004)
7. Lin, C.T., Cheng, W.C., Liang, S.F.: An on-line ICA-mixture-model-based self-constructing fuzzy neural network. IEEE Trans. Circuits Syst. I **52**(1), 207–221 (2005)
8. Juang, C.F., Wang, C.Y.: A self-generating fuzzy system with ant and particle swarm cooperative optimization. Expert Syst. Appl. **36**(3), 5362–5370 (2009)
9. Gao, Y., Er, M.J.: Online adaptive fuzzy neural identification and control of a class of MIMO nonlinear systems. IEEE Trans. Fuzzy Syst. **11**(4), 462–477 (2003)
10. Lin, F.J., Lin, C.H.: A permanent-magnet synchronous motor servo drive using self-constructing fuzzy neural network controller. IEEE Trans. Energy Convers. **19**(1), 66–72 (2004)
11. Hsu, C.F.: Self-organizing adaptive fuzzy neural control for a class of nonlinear systems. IEEE Trans. Neural Netw. **18**(4), 1232–1241 (2007)
12. Cheng, K.H.: Auto-structuring fuzzy neural system for intelligent control. J. Franklin Inst. **346**(3), 267–288 (2009)
13. Chen, C.S.: Dynamic structure adaptive neural fuzzy control for MIMO uncertain nonlinear systems. Inf. Sci. **179**(15), 2676–2688 (2009)
14. Lin, C.M., Hsu, C.F.: Neural network hybrid control for antilock braking systems. IEEE Trans. Neural Netw. **14**(2), 351–359 (2003)
15. Wu, T.F., Tsai, P.S., Chang, F.R., Wang, L.S.: Adaptive fuzzy CMAC control for a class of nonlinear system with smooth compensation. IEE Proc., Control Theory Appl. **153**(6), 647–657 (2006)
16. Hsu, C.F., Cheng, K.H., Lee, T.T.: Robust wavelet-based adaptive neural controller design with a fuzzy compensator. Neurocomputing **73**(1), 423–431 (2009)
17. Peng, Y.F., Lin, C.M.: Intelligent motion control of linear ultrasonic motor with H^∞ tracking performance. IEE Proc., Control Theory Appl. **1**(1), 9–17 (2007)
18. Kang, Q., Wang, W.: Adaptive fuzzy controller design for a class of uncertain nonlinear MIMO systems. Nonlinear Dyn. **59**(4), 579–591 (2010)
19. Lin, C.J.: An efficient immune-based symbiotic particle swarm optimization learning algorithm for TSK-type neuro-fuzzy networks design. Fuzzy Sets Syst. **159**(21), 2890–2909 (2008)
20. Juang, C.F., Lo, C.: Zero-order TSK-type fuzzy system learning using a two-phase swarm intelligence. Fuzzy Sets Syst. **159**(21), 2910–2926 (2008)

21. Wai, R.J., Chen, P.C.: Intelligent tracking control for robot manipulator including actuator dynamics via TSK-type fuzzy neural network. IEEE Trans. Fuzzy Syst. **12**(4), 552–560 (2004)
22. Lin, F.J., Shen, P.H., Chou, P.H., Yang, S.L.: TSK-type recurrent fuzzy network for dsp-based permanent-magnet linear synchronous motor servo drive. IEE Proc., Electr. Power Appl. **153**(6), 921–931 (2006)
23. Chen, C.H., Lin, C.J., Lin, C.T.: A functional-link-based neuro-fuzzy network for nonlinear system control. IEEE Trans. Fuzzy Syst. **16**(5), 1362–1378 (2008)
24. Jang, J.O.: Neural network saturation compensation for DC motor systems. IEEE Trans. Ind. Electron. **54**(3), 1763–1767 (2007)
25. Nouri, K., Dhaouadi, R., Benhadj Braiek, N.: Adaptive control of a nonlinear DC motor drive using recurrent neural networks. Appl. Soft Comput. **8**(1), 371–382 (2008)
26. Slotine, J.J.E., Li, W.P.: Applied Nonlinear Control. Prentice-Hall, Englewood Cliffs (1991)
27. Almutairi, N.B., Zribi, M.: On the sliding mode control of a ball on a beam system. Nonlinear Dyn. **59**(1), 221–238 (2010)
28. Lin, C.M., Chen, T.Y.: Self-organizing CMAC control for a class of MIMO uncertain nonlinear systems. IEEE Trans. Neural Netw. **20**(9), 1377–1384 (2009)
29. Hsu, C.F., Chung, C.M., Lin, C.M., Hsu, C.Y.: Adaptive CMAC neural control of chaotic systems with a PI-type learning algorithm. Expert Syst. Appl. **36**(9), 11836–11843 (2009)
30. See http://www.altera.com/
31. See http://www.terasic.com.tw/
32. Chen, C.S., Chen, H.H.: Robust adaptive neural-fuzzy-network control for the synchronization of uncertain chaotic systems. Nonlinear Anal., Real World Appl. **10**(3), 1466–1479 (2009)