Int. J. Fuzzy Syst. (2017) 19(5):1362–1374
DOI 10.1007/s40815-017-0301-6

PSO-Self-Organizing Interval Type-2 Fuzzy Neural Network for Antilock Braking Systems

Chih-Min Lin[1] · Tien-Loc Le[1,2]

Received: 25 July 2016 / Revised: 8 December 2016 / Accepted: 3 February 2017 / Published online: 1 March 2017

Abstract Nowadays, the antilock braking system (ABS) is the standard in all modern cars. The function of ABS is to optimize the maximize wheel traction by preventing wheel lockup during braking, so it can help the drivers to maintain steering maneuverability. In this study, a self-organizing interval type-2 fuzzy neural network (SOT2FNN) control system is designed for antilock braking systems. This control system comprises a main controller and a robust compensation controller; the SOT2FNN as the main controller is used to mimic an ideal controller, and the robust compensation controller is developed to eliminate the approximation error between the main controller and the ideal controller. To guarantee system stability, adaptive laws for adjusting the parameters of SOT2FNN based on the gradient descent method are proposed. However, in control design, the learning rates of adaptive law are very important and they significantly affect control performance. The particle swarm optimization method is therefore applied to find the optimal learning rates for the weights in reduction layer and also for the means, the variances of the Gaussian functions in the input membership functions. Finally, the numerical simulations of ABS response in different road conditions are provided to illustrate the effectiveness of the proposed approach.

Keywords Type-2 fuzzy logic system · Antilock braking system · Particle swarm optimization · Self-organizing learning algorithm

1 Introduction

The antilock braking system is a safety–critical system in modern cars, which was first invented for the aviation industry in 1920 [1]. During the braking action, ABS helps maintain the steerability of the car by limiting the longitudinal slip. The challenging problem in designing an ABS is that the vehicle braking dynamics are highly nonlinear and the environmental parameters are unknown. Since most commercial ABS control methods do not have an adaptive function to deal with this problem, their performance is often degraded under harsh road conditions [2, 3]. Recently, many intelligent control methods such as fuzzy control, neural network control, sliding mode control, hybrid control and cerebellar model articulation controller (CMAC) have been applied in ABS to improve the performance [3–11]. In 2010, Sharkawy presented a fuzzy self-tuning PID controller for ABS [5]. In 2012, Corno et al. [7] provided hybrid ABS control using force measurement. Following that, in 2013, Lin and Li introduced a self-organizing function-link fuzzy CMAC for ABS [8], where CMAC is a type of neural network based on a model of the mammalian cerebellum [12]. In 2016, Peric et al. [9] presented an ABS control using a quasi-sliding mode control with neural network estimator. However, some of the above methods are complex, and their parameters adjusting laws are not effective enough; furthermore, most control performance can be further improved. In this paper, a self-organizing interval type-2 fuzzy neural network control system will be proposed to control ABS. The

✉ Chih-Min Lin
cml@saturn.yzu.edu.tw

Tien-Loc Le
tienloclhu@gmail.com

[1] Yuan Ze University, Taoyuan, Taiwan

[2] Department of Electrical Electronic and Mechanical Engineering, Lac Hong University, BienHoa, Vietnam

proposed controller incorporates the advantages of sliding mode control to enhance stability and performance of the control system [13].

In recent years, the type-2 fuzzy logic system (T2FLS) has attracted many researchers in many application fields such as control problem, system identification, prediction and classification [14–22]. The fuzzy set was first introduced by Zadeh in 1965 [23], and shortly thereafter, it was applied in many fields due to its advantageous properties such as flexibility, intuitive knowledge and easy computation. However, in many cases, this type-1 fuzzy logic system has difficultly achieving optimal performance because it cannot effectively deal with the uncertainty coming from the internal and external disturbances of the plant [14]. To solve this problem, in 1975 Zadeh developed the concept of type-2 fuzzy sets [24], which are potentially more suitable for the plant with uncertainty. Moreover, a T2FLS has the potential to outperform a T1FLS, because it has more design degrees of freedom. The T1FLS can be viewed as a special case of T2FLS. In T2FLS, if the means and variances of upper and lower membership functions are the same, then it can be reduced to a T1FLS. So T2FLS is more general than T1FLS. The T2FLS with more complex computation led to the appearance of interval type-2 fuzzy logic systems (IT2FLS), which is a special case of T2FLS simplifying the computation operations in the fuzzification and defuzzification of fuzzy logic systems [25]. By combining the fuzzy logic system with a neural network, several fuzzy neural systems have been presented [26–29]. However, in the design of fuzzy neural systems, it is often difficult to determine a suitable network size, so some papers have dealt with the self-organization of fuzzy neural systems [8, 30].

In many adaptive controllers, the learning rates are highly influential to the system performance. Most papers in the literature often need to use a trial-and-error method to determine the learning rates of parameter adaptive laws, but it is difficult to obtain the most suitable value and it always takes a long time. The PSO algorithm is an optimization technique first introduced in 1995 by Kennedy and Eberhart [31]. It optimizes a problem by modifying each candidate solution based on the best performance of the swarm. In the beginning, the particles (set as learning rates) will be initialized with a group of random solutions. At each iteration, the best solution can be shared among other particles and every particle will move to follow the best solution. The PSO algorithm is especially effective in solving nonlinear problems in control systems; so in the last decade, it has been widely used to optimize the parameters for many controllers such as PID, LQR, fuzzy and neural network [32–36]. In 2011, Bingül and Karahan [35] proposed PSO fuzzy logic controller for a 2 DOF robot. In their study, the PSO algorithm was used to tune the antecedent and consequent parameter of fuzzy law, which used fixed structure of type-1 fuzzy system. The advance of this method is its simplicity in calculation, so it is easier to be implemented than other traditional methods. However, because of with fixed structure, it led to difficulty to determine the suitable size of fuzzy membership functions and fuzzy rules. In [37], a self-evolving algorithm for an interval type-2 fuzzy neural network (IT2FNN) has been established, which can auto-generate the network structure. The advance of this approach is the initial rule-based and membership function can be empty initially and then can be constructed by itself, but one cannot use the knowledge about the system to design the initial membership function and the initial rule to obtain better result; furthermore, one also cannot apply off-line PSO algorithm to train the network. In [38], a self-organizing IT2FNN was introduced, which can self-construct the network size. Nevertheless, in this paper, the learning rate in adaptive law is fixed and chosen by the trial-and-error method.

The motivation of this study is to construct a self-organizing type-2 fuzzy neural network by applying a self-organizing algorithm for the type-2 fuzzy neural systems, and the PSO algorithm will be applied to find the optimal learning rates of SOT2FNN; and this network is referred to as a PSO-SOT2FNN. Then, it is applied to control the antilock braking systems. The main contributions of this paper are (1) successful apply off-line and online PSO algorithm to find the optimal learning rates of the adaptive laws of SOT2FNN, (2) the self-organizing algorithm is used to achieve the suitable structure of IT2FNN, (3) all the adaptive laws are designed based on the gradient descent method, and the stability of the control system is proved by the Lyapunov function and (4) the simulation results illustrate that the proposed control system can effectively achieve braking responses for the ABS under different road conditions.

The remainder of this paper is as follows. The problem formulation of ABS is formulated in Sect. 2. The design of the PSO-SOT2FNN control system is presented in Sect. 3. The simulation results are provided in Sect. 4 to demonstrate the effectiveness of the proposed approach, and a conclusion is given in Sect. 5.

2 Formulation of ABS

In antilock braking systems, the control objective is to regulate the wheel slip to maximize the coefficient of friction between the wheel and the road for any given road surface. In general, the coefficient of friction μ during a braking operation can be described as a function of the slip λ, which is defined as

$$\lambda(t) = \frac{\omega_{\mathrm{v}}(t) - \omega_{\omega}(t)}{\omega_{\mathrm{v}}(t)} \tag{1}$$

where $\omega_{\mathrm{v}}(t)$ and $\omega_{\omega}(t)$ are the angular velocity of vehicle and the wheel, respectively.

The relationship between the braking coefficient of friction μ and the wheel slip λ depends on the tire–road conditions and has been measured in [39], as shown in Fig. 1 [3, 6]. From (1), it is obvious when $\omega_{\mathrm{v}}(t) = \omega_{\omega}(t)$, the slip λ is 0% representing the free rolling wheel condition, while when $\omega_{\omega}(t) = 0$, the slip λ is 100% corresponding to the wheel being completely locked. The angular velocity is defined as

$$\omega_{\mathrm{v}}(t) = \frac{V_{\mathrm{v}}(t)}{R_{\omega}} \tag{2}$$

where R_{ω} is the radius of the wheel, $V_{\mathrm{v}}(t)$ is the velocity of the vehicle. Applying Newton's law, the dynamic equations for the wheel and the vehicle during a nominal braking are derived [3, 6]. The acceleration of the vehicle is determined by summing the total forces applied.

$$\dot{V}_{\mathrm{v}}(t) = \frac{-1}{M_{\mathrm{v}}}[4F_{\mathrm{t}}(t) + B_{\mathrm{v}}V_{\mathrm{v}}(t) + F_{\theta}(\theta)] \tag{3}$$

where M_{v} is the mass of the vehicle, B_{v} is the vehicular viscous friction, $F_{\mathrm{t}}(t)$ is the tractive force and $F_{\theta}(\theta)$ is the force applied to the car which results from a vertical gradient in the road so

$$F_{\theta}(\theta) = M_{\mathrm{v}}g\sin(\theta) \tag{4}$$

where θ is the angle of inclination of the road, and g is the gravitational acceleration constant. The tractive force $F_{\mathrm{t}}(t)$ is given by

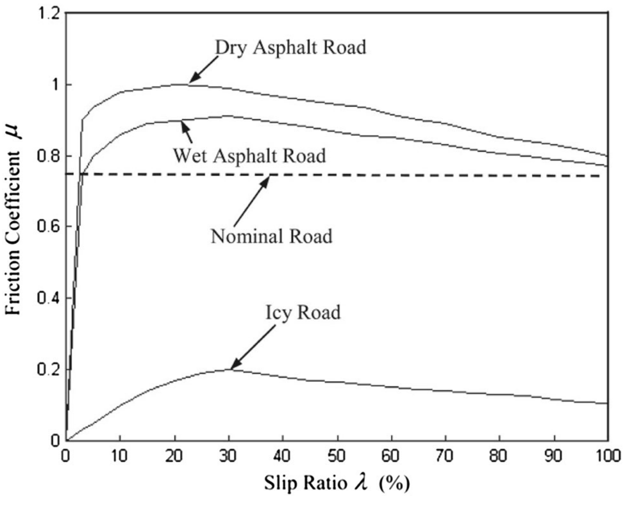

Fig. 1 Relation between road–wheel friction coefficients (μ) and slip (λ) for various road surfaces [3, 39]

$$F_{\mathrm{t}}(t) = \mu(\lambda)N_{\mathrm{v}}(\theta) \tag{5}$$

where $N_{\mathrm{v}}(\theta)$ is the nominal reaction force applied to the wheel. For this model, assume the vehicle has four wheels and the weight of the vehicle is evenly distributed between these wheels. Then, the nominal reaction force on each wheel $N_{\mathrm{v}}(\theta)$ can be expressed by

$$N_{\mathrm{v}}(\theta) = \frac{M_{\mathrm{v}}g}{4}\cos(\theta) \tag{6}$$

The wheel dynamic is determined by summing the rotational torque to yield

$$\dot{\omega}_{\omega}(t) = \frac{1}{J_{\omega}}[-T_{\mathrm{b}}(t) - B_{\omega}\omega_{\omega}(t) + T_{\mathrm{t}}(t)] \tag{7}$$

where J_{ω} is the rotational inertia of the wheel, B_{ω} is the viscous friction of the wheel, $T_{\mathrm{b}}(t)$ is the braking torque and $T_{\mathrm{t}}(t)$ is the torque generated due to the slip between the wheel and the road surface. In general, $T_{\mathrm{t}}(t)$ is the torque generated by the tractive force $F_{\mathrm{t}}(t)$ as

$$T_{\mathrm{t}}(t) = R_{\omega}F_{\mathrm{t}}(t) \tag{8}$$

Taking the derivative from (1)

$$\dot{\lambda}(t) = \frac{(1-\lambda(t))\dot{\omega}_{\mathrm{v}} - \dot{\omega}_{\omega}}{\omega_{\mathrm{v}}} = \frac{-\dot{\omega}_{\mathrm{v}}}{\omega_{\mathrm{v}}}\lambda(t) + \frac{\dot{\omega}_{\mathrm{v}} - \dot{\omega}_{\omega}}{\omega_{\mathrm{v}}} \tag{9}$$

Substituting (2), (3) and (7) into (9) gives

$$\dot{\lambda}(t) = f(\lambda;t) + g(t)u(t) \tag{10}$$

where $f = \frac{4F_{\mathrm{t}}+B_{\mathrm{v}}R_{\omega}\omega_{\mathrm{v}}+F_{\theta}}{M_{\mathrm{v}}R_{\omega}\omega_{\mathrm{v}}}\lambda - \frac{(4F_{\mathrm{t}}+B_{\mathrm{v}}R_{\omega}\omega_{\mathrm{v}}+F_{\theta})J_{\omega}-(B_{w}\omega_{w}-T_{\mathrm{t}})M_{\mathrm{v}}R_{\omega}}{M_{\mathrm{v}}R_{\omega}\omega_{\mathrm{v}}J_{\omega}}$ and $g = \frac{1}{J_{\omega}\omega_{\mathrm{v}}}$ is the control gain, $u = T_{\mathrm{b}}$ is the control effort. This equation is nonlinear and involves uncertainties in the parameters. When uncertainties and measurement noise are under consideration, (10) can be reformulated as

$$\begin{aligned}\dot{\lambda}(t) &= [f_0(\lambda;t) + \Delta f(\lambda;t)] + [g_0(t) + \Delta g(t)]u(t) + n(t) \\ &= f_0(\lambda;t) + g_0(t)u(t) + \beta(\lambda;t)\end{aligned} \tag{11}$$

where $f_0(\lambda;t)$ and $g_0(t)$ are the nominal parts of $f(\lambda;t)$ and $g(t)$, respectively, $\Delta f(\lambda;t)$ and $\Delta g(t)$ are the uncertainties of system, $n(t)$ denotes the measurement noise, $\beta(\lambda;t)$ is referred to as the lumped uncertainty and is defined as

$$\beta(\lambda;t) = \Delta f(\lambda;t) + \Delta g(t)u(t) + n(t) \tag{12}$$

In the case that the lumped uncertainty is known exactly, then an ideal controller can be designed as follows:

$$u^{*}(t) = g_0^{-1}(t)\left[\dot{\lambda}_{\mathrm{d}}(t) - f_0(\lambda;t) - \beta(\lambda;t) + k\lambda_{\mathrm{e}}(t)\right] \tag{13}$$

where k is a gain, and $\lambda_{\mathrm{e}}(t)$ is defined as

$$\lambda_{\mathrm{e}}(t) = \lambda_{\mathrm{d}}(t) - \lambda(t) \tag{14}$$

where $\lambda_{\mathrm{d}}(t)$ is the desired slip trajectory.

The goal of the controller produces a control signal $u(t)$, which can force the system output $\lambda(t)$ to track the reference trajectory $\lambda_d(t)$ to maximize the road/wheel friction, regardless of variations in vehicular speed. The desired slip is usually set at 20% based on Fig. 1 to meet the maximum friction coefficient. Substituting (13) into (11), the error dynamics can be obtained as

$$\dot{\lambda}_e(t) + k\lambda_e(t) = 0 \tag{15}$$

From (15), it is obvious if k is selected to correspond to the coefficients of the Hurwitz polynomial, then the tracking error can approach zero $\lim_{t\to\infty} \lambda_e(t) = 0$, and then, the system can follow the desired slip trajectory $\lambda_d(t)$. However, the lumped uncertainty $(\boldsymbol{x}(t))$ cannot be precisely known in practical applications. Therefore, the ideal controller in (13) is unobtainable. Thus, a PSO-SOT2FNN control system is proposed in the following section to achieve the desired control performance.

3 PSO-SOT2FNN Control System

The structure of the PSO-SOT2FNN control system is shown in Fig. 2, consisting of an SOT2FNN controller and a robust compensation controller. The SOT2FNN is the main controller, and its parameters can be adjusted based on the derived adaptive law; and the learning rates in adaptive laws can be updated using the PSO algorithm.

3.1 Interval Type-2 Fuzzy Neural Network

This section introduces the structure of an interval type-2 fuzzy neural network (IT2FNN), consisting of an input layer, a membership layer, a rule layer, a type reduction layer and an output layer, as shown in Fig. 3. The IF–THEN rule for IT2FNN has the following form:

$$\begin{array}{ll} \text{Rule i:} \quad \text{IF} \quad x_1\,\text{is}\,\tilde{X}_1^i & \text{and}\ldots\text{and}x_n\;is\,\tilde{X}_n^i \\ \text{THEN} \quad w\,\text{is}\,\tilde{W}^i \quad i = 1,\ldots,M & \end{array} \tag{16}$$

where $\tilde{X}_j^i$ and $\tilde{W}^i$ are the interval type-2 fuzzy membership function input and output, respectively. M is the total number of rulers and $j = 1,\ldots,n$ where n is the number of the inputs in the first layer.

The signal propagation and basic function in each layer are described as follows.

1. Layer 1 (Input Layer): there are no weights in this layer, and each node is an input variable which will be transmitted directly to layer 2. In this study, the inputs are the error and the derivative of the error.
2. Layer 2 (membership function layer): *each node* in this layer defines an interval type-2 membership function (IT2MF) to perform the fuzzification operation. The IT2MF $\tilde{X}_j^i$ is defined by an interval type-2 Gaussian membership function, which has a standard deviation σ and an uncertain mean $m \in [m_1, m_2]$ (Fig. 4); and it can be described as:

$$\mu_{\tilde{X}_j^i} = \exp\left\{-\frac{1}{2}\left(\frac{x_j - m_j^i}{\sigma_j^i}\right)^2\right\} \equiv O\left(m_j^i, \sigma_j^i, x_j\right) \tag{17}$$

The footprint of uncertainty (FOU) of IT2MF can be represented as a bounded interval denoted by $\bar{\mu}_j^i, \underline{\mu}_j^i$ the upper and lower membership function (UMF and LMF), respectively. The output of each node in this

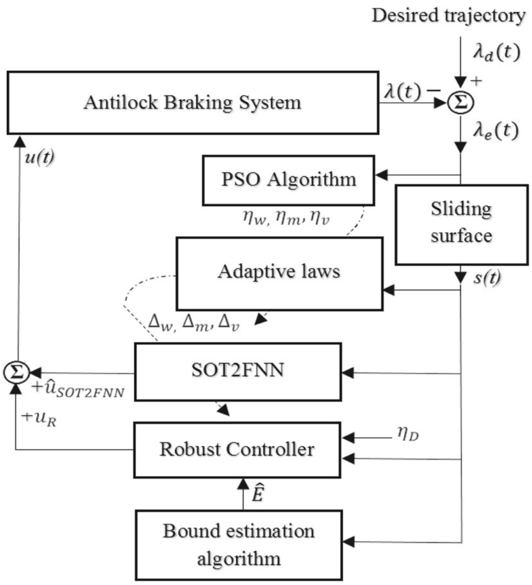

Fig. 2 Block diagram of PSO-SOT2FNN control system

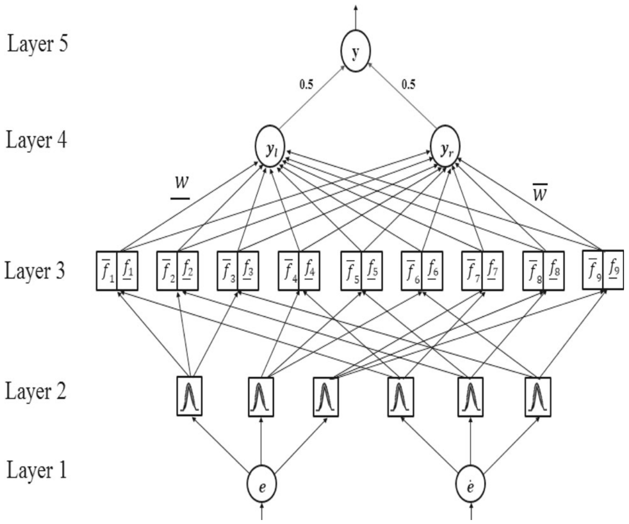
Layer 5
y
0.5
0.5
Layer 4
y_l
y_r
$\underline{w}$
$\overline{w}$
Layer 3
$\bar{f}_1$ $\underline{f}_1$
$\bar{f}_2$ $\underline{f}_2$
$\bar{f}_3$ $\underline{f}_3$
$\bar{f}_4$ $\underline{f}_4$
$\bar{f}_5$ $\underline{f}_5$
$\bar{f}_6$ $\underline{f}_6$
$\bar{f}_7$ $\underline{f}_7$
$\bar{f}_8$ $\underline{f}_8$
$\bar{f}_9$ $\underline{f}_9$
Layer 2
Layer 1
e
ė

Fig. 3 Structure of an IT2FNN

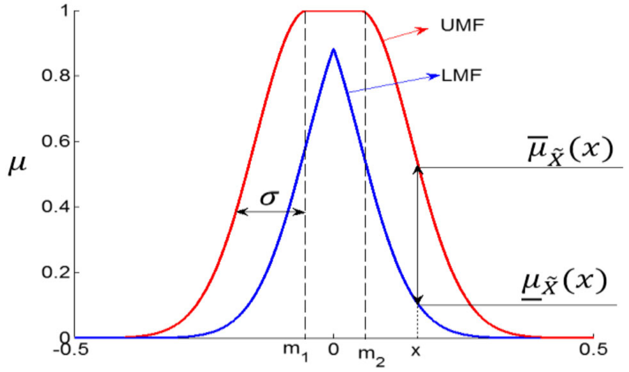

Fig. 4 Interval type-2 fuzzy Gaussian membership function

layer can be represented as an interval $\left[\underline{\mu}_j^i, \bar{\mu}_j^i\right]$ and is given by

$$\bar{\mu}_j^i(x_j) = \begin{cases} O\left(m_{j1}^i, \sigma_j^i, x_j\right), & x_j < m_{j1}^i \\ 1, & m_{j1}^i \le x_j \le m_{j2}^i \\ O\left(m_{j2}^i, \sigma_j^i, x_j\right), & x_j > m_{j2}^i \end{cases} \tag{18}$$

$$\underline{\mu}_j^i(x_j) = \begin{cases} O\left(m_{j2}^i, \sigma_j^i, x_j\right), & x_j \le \dfrac{m_{j1}^i + m_{j2}^i}{2} \\ O\left(m_{j1}^i, \sigma_j^i, x_j\right), & x_j > \dfrac{m_{j1}^i + m_{j2}^i}{2} \end{cases} \tag{19}$$

3. Layer 3 (rule layer): each node in this layer is a rule node and performs the fuzzy firing operation using an algebraic product *t*-norm operation. The output of a rule node is a firing strength F^i, which is an interval type-1 fuzzy set. The firing strength is computed as follows [25]:

$$F^i = \left[\underline{f}^i, \bar{f}^i\right] \tag{20}$$

where

$$\bar{f}^i = \prod_{j=1}^{n} \bar{\mu}_j^i, \underline{f}^i = \prod_{j=1}^{n} \underline{\mu}_j^i \tag{21}$$

4. Layer 4 (type reduction layer): In this layer, the center of set type reduction and the Karnik–Mendel (KM) algorithms are used to compute the interval output $[y_l, y_r]$ [40], which can be derived from the consequent centroid set $[\underline{w}^i, \bar{w}^i]$ and firing strengths $\left[\underline{f}^i, \bar{f}^i\right]$. The left and right outputs y_l and y_r can be computed by

$$y_l = \frac{\sum_{i=1}^{M} f_l^i \underline{w}^i}{\sum_{i=1}^{M} f_l^i} \tag{22}$$

$$y_r = \frac{\sum_{i=1}^{M} f_r^i \bar{w}^i}{\sum_{i=1}^{M} f_r^i} \tag{23}$$

where f_l^i and f_r^i are chosen as

$$f_l^i = \begin{cases} \bar{f}^i, & i \le L \\ \underline{f}^i, & i > L \end{cases} \tag{24}$$

$$f_r^i = \begin{cases} \underline{f}^i, & i \le R \\ \bar{f}^i, & i > R \end{cases} \tag{25}$$

where L and R are the left and right switch points, respectively.

5. Layer 5 (output layer): this layer performs the defuzzification operation from the interval set in the output of layer 4, and the output can be computed by the average of y_l and y_r. Finally, the output of IT2FNN can be expressed as

$$y = \frac{y_l + y_r}{2} \tag{26}$$

3.2 Self-Organizing for Type-2 Fuzzy Neural Network

In the design structure of IT2FNN, it is difficult to determine the number of rules for the system to obtain favorable performance, and many previous published papers used a trial-and-error method to determine the rule's number. However, using this method takes a long time and the performance cannot be guaranteed. The idea of a self-organizing IT2FNN is to use the firing strength of each rule to determine whether a new rule needs to be generated or an inappropriate existing rule needs to be deleted.

Consider the process of generating the new rules. The mathematical description of a rule can be expressed as a cluster, and the degree of input data that belongs to a cluster can be considered according to its firing strength. The center of interval firing strength can be computed by

$$f_c^i = \frac{1}{2}\left(\underline{f}^i + \bar{f}^i\right) \tag{27}$$

If $\left(f_c^I \le f_{\mathrm{th}}\right)$ means a new input data fall outside the boundary of all existing clusters, then the SOT2FNN will generate a new rule to cover it, where f_{th} is the prespecified threshold and f_c^I is the maximum value of the center interval firing strength and I is the index of the maximum value given by

$$I = \arg\max_i \left[f_c^1, f_c^2, \ldots f_c^M\right] \tag{28}$$

In the new rule, the initial values for the uncertain mean and the variance of the type-2 fuzzy system are defined as

$$\left[m_{j1}^{M(t)+1}, m_{j2}^{M(t)+1}\right] = \left[x_j(t) - \Delta x, x_j(t) + \Delta x\right] \quad j = 1, \ldots, n \tag{29}$$

$$\sigma_j^{M(t)+1} = \beta . \left| x_j(t) - \left(\frac{m_{j1}^I + m_{j2}^I}{2}\right) \right| \tag{30}$$

$$\left[\underline{w}^i, \bar{w}^i\right] = [0, 0] \tag{31}$$

where $M(t)$ is the total number of rules at time t, Δx is a half of the uncertainty mean value. If the chosen Δx is extremely small, it will be similar to a type-1 fuzzy set. For an extremely large value of Δx, the uncertainty will cover most of the input domain and cover other existing IT2MFs. The positive parameter β decides the initial cluster width. A large β value leads to a large overlapping degree between these fuzzy sets and generates a smaller number of rules [15]. Conversely, if β is an extremely small value, it will lead to a small overlapping degree between these fuzzy sets and will generate a huge number of rules.

Choosing the values of Δx and β is the design of the initial values for the uncertainty mean and variance. This paper sets $\Delta x = 0.1$ and $\beta = 0.5$ based on the adaptive law designed in the following, and the uncertainty mean and the variance can be updated to a suitable value.

Another process of self-organizing is to delete the existing rules. The rules will be deleted if they contribute less than the predefined deleting threshold value K_{del}. The ratio used to evaluate the contribution of the ith rule in IT2FNN is defined as

$$K_l^i = \frac{f_l^i \underline{w}^i}{y_l} \quad \text{and} \quad K_r^i = \frac{f_r^i \bar{w}^i}{y_r} \tag{32}$$

If $K_l^i < K_{\text{del}}$ and $K_r^i < K_{\text{del}}$, then the ith rule in the left or in the right is deleted. With this automatic way of generating and pruning, the proposed self-organizing IT2FNN can obtain the most suitable number of rules.

3.3 Adaptive Type-2 Fuzzy Neural Network Controller Design

Since the lumped uncertainty in (11) is unknown, assume there is an optimal SOT2FNN controller u^*_{SOT2FNN} to approximate the ideal controller $u^*(t)$ in (13)

$$u^*(t) = u^*_{SOFT2NN}\left(\underline{w}^*, \bar{w}^*, m^*_{j1}, m^*_{j2}, \sigma^*_j, t\right) + \varepsilon(t) \tag{33}$$

where $\varepsilon(t)$ denotes the approximation error, and $\underline{w}^*, \bar{w}^*, m^*_{j1}, m^*_{j2}\text{and}\sigma^*_j$ are the optimal parameters of $\underline{w}, \bar{w}, m_{j1}\text{and}\sigma_j$, respectively. However, the optimal controller u^*_{SOT2FNN} cannot be obtained. Thus, an estimation controller $\hat{u}_{\text{SOT2FNN}}$ can be designed to online estimate u^*_{SOT2FNN}, and the overall controller can be designed as

$$\hat{u}(t) = \hat{u}_{\text{SOT2FNN}}\left(\underline{\hat{w}}, \hat{\bar{w}}, \hat{m}_{J1}, \hat{m}_{J2}, \hat{\sigma}_j, t\right) + \hat{u}_R(t) \tag{34}$$

where $\hat{u}_R$ is a robust compensation controller used to cope with the approximation error between the estimation controller $\hat{u}_{\text{SOT2FNN}}$ and the optimal controller u^*_{SOT2FNN}; and $\underline{\hat{w}}, \hat{\bar{w}}, \hat{m}_{J1}, \hat{m}_{J2}, \hat{\sigma}_j, t$ are the estimation of $\underline{w}^*, \bar{\text{w}}^*, m^*_{j1}$, m^*_{j2}, σ^*_j, respectively.

A sliding surface $s(t)$ is defined as

$$s(t) \triangleq \lambda_{\text{e}}(t) + k\int_0^t \lambda_{\text{e}}(\tau)\text{d}\tau \tag{35}$$

Taking the derivative of (35) and using (11), (14) and (34), yields

$$\begin{aligned} \dot{s}(t) &= \dot{\lambda}_{\text{e}}(t) + k\lambda_{\text{e}}(t) \\ &= -f_0(\lambda; t) - g_0(x)(\hat{u}_{\text{SOT2FNN}}(t) + \hat{u}_R(t)) + \dot{\lambda}_{\text{d}}(t) \\ &\quad - (\lambda; t) + k\lambda_{\text{e}}(t) \end{aligned} \tag{36}$$

Choosing a Lyapunov cost function as $V_1(s(t)) =\frac{1}{2}s^2(t)$, take the derivative of the cost function $\dot{V}_1(s(t)) = s(t)\dot{s}(t)$. The aim is to tune the parameter values $\underline{\hat{w}}, \hat{\bar{w}}, \hat{m}_{J1}, \hat{m}_{J2}, \hat{\sigma}_j$ so $\dot{V}_1(s(t))$ is minimized to achieve rapid convergence of $s(t)$. Using the gradient descent method, the parameters of type-2 fuzzy system can be updated by the following equations:

$$\underline{\hat{w}}^i(t+1) = \underline{\hat{w}}^i(t) - \hat{\eta}_w \frac{\partial s(t)\dot{s}(t)}{\partial \underline{\hat{w}}^i} \tag{37}$$

$$\hat{\bar{w}}^i(t+1) = \hat{\bar{w}}^i(t) - \hat{\eta}_w \frac{\partial s(t)\dot{s}(t)}{\partial \hat{\bar{w}}^i} \tag{38}$$

$$\hat{m}^i_{j1}(t+1) = \hat{m}^i_{j1}(t) - \hat{\eta}_m \frac{\partial s(t)\dot{s}(t)}{\partial \hat{m}^i_{j1}} \tag{39}$$

$$\hat{m}^i_{j2}(t+1) = \hat{m}^i_{j2}(t) - \hat{\eta}_m \frac{\partial s(t)\dot{s}(t)}{\partial \hat{m}^i_{j2}} \tag{40}$$

$$\hat{\sigma}^i_j(t+1) = \hat{\sigma}^i_j(t) - \hat{\eta}_\sigma \frac{\partial s(t)\dot{s}(t)}{\partial \hat{\sigma}^i_j} \tag{41}$$

where $\hat{\eta}_w, \hat{\eta}_m, \hat{\eta}_\sigma$ are the learning rates with positive number and they can be obtained by PSO algorithm, and the detail of PSO will be presented in the following Sect. 3.4.

Using the chain rule, the derivation in (37)–(41) can be expressed as follows:

$$\begin{aligned} \frac{\partial s(t)\dot{s}(t)}{\partial \underline{\hat{w}}^i} &= \frac{\partial s(t)\dot{s}(t)}{\partial \hat{u}_{\text{SOT2FNN}}} \frac{\partial \hat{u}_{\text{SOT2FNN}}}{\partial y_l} \frac{\partial y_l}{\partial \underline{\hat{w}}^i} \\ &= -\frac{1}{2} g_0(x) s(t) \frac{f_l^i}{\sum_{i=1}^M f_l^i} \end{aligned} \tag{42}$$

$$\begin{aligned} \frac{\partial s(t)\dot{s}(t)}{\partial \underline{\hat{w}}^i} &= \frac{\partial s(t)\dot{s}(t)}{\partial \hat{u}_{\text{SOT2FNN}}} \frac{\partial \hat{u}_{\text{SOT2FNN}}}{\partial y_r} \frac{\partial y_r}{\partial \hat{\bar{w}}^i} \\ &= -\frac{1}{2} g_0(x) s(t) \frac{f_r^i}{\sum_{i=1}^M f_r^i} \end{aligned} \tag{43}$$

$$\begin{aligned} \frac{\partial s(t)\dot{s}(t)}{\partial \hat{m}^i_{j1}} &= \frac{1}{2} \frac{\partial s(t)\dot{s}(t)}{\partial \hat{u}_{\text{SOT2FNN}}} \left(\frac{\partial y_l}{\partial f_l^i} \frac{\partial f_l^i}{\partial \hat{m}^i_{j1}} + \frac{\partial y_r}{\partial f_r^i} \frac{\partial f_r^i}{\partial \hat{m}^i_{j1}} \right) \\ &= -\frac{1}{2} g_0(x) s(t) \left(\frac{(\underline{w}^i - y_l)}{\sum_{i=1}^M f_l^i} \frac{\partial f_l^i}{\partial \hat{m}^i_{j1}} + \frac{(\bar{w}^i - y_r)}{\sum_{i=1}^M f_r^i} \frac{\partial f_r^i}{\partial \hat{m}^i_{j1}} \right) \end{aligned} \tag{44}$$

$$\frac{\partial s(t)\dot{s}(t)}{\partial \hat{m}_{j2}^{i}} = \frac{1}{2}\frac{\partial s(t)\dot{s}(t)}{\partial \bar{u}_{\mathrm{SOT2FNN}}}\left(\frac{\partial y_l}{\partial f_l^i}\frac{\partial f_l^i}{\partial \hat{m}_{j2}^{i}} + \frac{\partial y_r}{\partial f_r^i}\frac{\partial f_r^i}{\partial \hat{m}_{j2}^{i}}\right)$$
$$= -\frac{1}{2}g_0(x)s(t)\left(\frac{(\underline{w}^i - y_l)}{\sum_{i=1}^{M} f_l^i}\frac{\partial f_l^i}{\partial \hat{m}_{j2}^{i}} + \frac{(\bar{w}^i - y_r)}{\sum_{i=1}^{M} f_r^i}\frac{\partial f_r^i}{\partial \hat{m}_{j2}^{i}}\right) \tag{45}$$

$$\frac{\partial s(t)\dot{s}(t)}{\partial \hat{\sigma}_{j}^{i}} = \frac{1}{2}\frac{\partial s(t)\dot{s}(t)}{\partial \hat{u}_{\mathrm{SOT2FNN}}}\left(\frac{\partial y_l}{\partial f_l^i}\frac{\partial f_l^i}{\partial \hat{\sigma}_{j}^{i}} + \frac{\partial y_r}{\partial f_r^i}\frac{\partial f_r^i}{\partial \hat{\sigma}_{j}^{i}}\right)$$
$$= -\frac{1}{2}g_0(x)s(t)\left(\frac{(\underline{w}^i - y_l)}{\sum_{i=1}^{M} f_l^i}\frac{\partial f_l^i}{\partial \hat{\sigma}_{j}^{i}} + \frac{(\bar{w}^i - y_r)}{\sum_{i=1}^{M} f_r^i}\frac{\partial f_r^i}{\partial \hat{\sigma}_{j}^{i}}\right) \tag{46}$$

The elements f_l^i and f_r^i in (44, 45, 46) are $\underline{f}^i$ or $\bar{f}^i$ depending on (24) and (25). Considering both cases, obtain

$$\frac{\partial \underline{f}^i}{\partial \hat{m}_{j1}^{i}} = \frac{\partial \underline{f}^i}{\partial \underline{\mu}_j^i}\frac{\partial \underline{\mu}_j^i}{\partial \hat{m}_{j1}^{i}} = \begin{cases} \underline{f}^i \times \dfrac{x_j - m_{j1}^i}{\left(\sigma_j^i\right)^2}, & x_j > \dfrac{m_{j1}^i + m_{j2}^i}{2} \\ 0, & \text{otherwise} \end{cases} \tag{47}$$

$$\frac{\partial \bar{f}^i}{\partial \hat{m}_{j1}^{i}} = \frac{\partial \bar{f}^i}{\partial \bar{\mu}_j^i}\frac{\partial \bar{\mu}_j^i}{\partial \hat{m}_{j1}^{i}} = \begin{cases} \bar{f}^i \times \dfrac{x_j - m_{j1}^i}{\left(\sigma_j^i\right)^2}, & x_j \le m_{j1}^i \\ 0, & \text{otherwise} \end{cases} \tag{48}$$

$$\frac{\partial \bar{f}^i}{\partial \hat{m}_{j2}^{i}} = \frac{\partial \bar{f}^i}{\partial \hat{\mu}_j^i}\frac{\partial \hat{\mu}_j^i}{\partial \hat{m}_{j2}^{i}} = \begin{cases} \bar{f}^i \times \dfrac{x_j - m_{j2}^i}{\left(\sigma_j^i\right)^2}, & x_j \le \dfrac{m_{j1}^i + m_{j2}^i}{2} \\ 0, & \text{otherwise} \end{cases} \tag{49}$$

$$\frac{\partial \bar{f}^i}{\partial \hat{m}_{j2}^{i}} = \frac{\partial \bar{f}^i}{\partial \bar{\mu}_j^i}\frac{\partial \bar{\mu}_j^i}{\partial \hat{m}_{j2}^{i}} = \begin{cases} \bar{f}^i \times \dfrac{x_j - m_{j2}^i}{\left(\sigma_j^i\right)^2}, & x_j > m_{j2}^i \\ 0, & \text{otherwise} \end{cases} \tag{50}$$

$$\frac{\partial \underline{f}^i}{\partial \hat{\sigma}_{j}^{i}} = \frac{\partial \underline{f}^i}{\partial \hat{\mu}_j^i}\frac{\partial \hat{\mu}_j^i}{\partial \hat{\sigma}_{j}^{i}} = \begin{cases} \underline{f}^i \times \dfrac{\left(x_j - m_{j2}^i\right)^2}{\left(\sigma_j^i\right)^3}, & x_j \le \dfrac{m_{j1}^i + m_{j2}^i}{2} \\ \underline{f}^i \times \dfrac{\left(x_j - m_{j1}^i\right)^2}{\left(\sigma_j^i\right)^3}, & x_j > \dfrac{m_{j1}^i + m_{j2}^i}{2} \end{cases} \tag{51}$$

$$\frac{\partial \bar{f}^i}{\partial \hat{\sigma}_{j}^{i}} = \frac{\partial \bar{f}^i}{\partial \bar{\mu}_j^i}\frac{\partial \bar{\mu}_j^i}{\partial \hat{\sigma}_{j}^{i}} = \begin{cases} \bar{f}^i \times \dfrac{\left(x_j - m_{j1}^i\right)^2}{\left(\sigma_j^i\right)^3}, & x_j < m_{j1}^i \\ \bar{f}^i \times \dfrac{\left(x_j - m_{j2}^i\right)^2}{\left(\sigma_j^i\right)^3}, & x_j < m_{j2}^i \\ 0, & \text{otherwise} \end{cases} \tag{52}$$

3.4 The Particle Swarm Optimization (PSO) Algorithm

The PSO algorithm is shown in Fig. 5. At the beginning of the off-line PSO algorithm, the swarm (n_{p} set value of learning rates $\hat{\eta}_w, \hat{\eta}_m, \hat{\eta}_\sigma$) are randomly initialized. Then, apply each particle of swarm to SOT2FNN based on the fitness function to obtain the best set values of the particle ($p_{\mathrm{Best}\,q}^{l}$). After trying all set values, calculate the best position of the swarm ($g_{\mathrm{Best}\,q}^{l}$) and update the new value for each particle. Finally, the main loop will run again with the new swarm until it reaches the fixed number of iterations. The formula for the updating law is

$$p_q^l(n+1) = p_q^l(n) + v_q^l(n+1) \tag{53}$$

and

$$v_q^l(n+1) = v_q^l(n) + c_1 * \mathrm{rand}(\cdot) * \left[p_{\mathrm{Best}\,q}^{l} - p_q^l(n)\right] + c_2 * \mathrm{rand}(\cdot) * \left[g_{\mathrm{Best}\,q}^{l} - p_q^l(n)\right] \tag{54}$$

where $v_q^l(n)$ and $p_q^l(n)$ are, respectively, the current velocity and current position of the particle, rand($\cdot$) is a random number in [0, 1]; c_1 and c_2 are the positive acceleration factors related to the local and global information, respectively, $q = 1, 2, \ldots, n_p$ (n_{p} is the population size) and $l = 1, 2, \ldots, n_d$ (n_{d} is the dimension of each particle). In this study, the fitness function is chosen based on the root mean square error (RMSE) of the slip tracking error.

The online PSO operation is the same as the off-line PSO, with the only two small differences being the initial value of the swarm and the running time in each epoch. In the online PSO, the initial values of the swarm are setup based on the best optimal value gotten from off-line PSO, and the epoch time is short to response to the online real-time application.

3.5 Robust Compensation Control

The robust compensation controller is designed to deal with the effect of the approximation error in (33). Assume this approximation error can be bounded by an uncertainty bound E and $0 \le \varepsilon(t) \le E$, where E is assumed to be a constant during the observation. However, it is difficult to know E precisely. Thus, an estimated value $\hat{E}(t)$ is used. Define the estimation error of the uncertainty bound

$$\hat{E}(t) = E - \hat{E}(t) \tag{55}$$

The robust compensation controller $u_R(t)$ is chosen as

$$u_R(t) = \hat{E}(t)\mathrm{sgn}(s(t)) \tag{56}$$

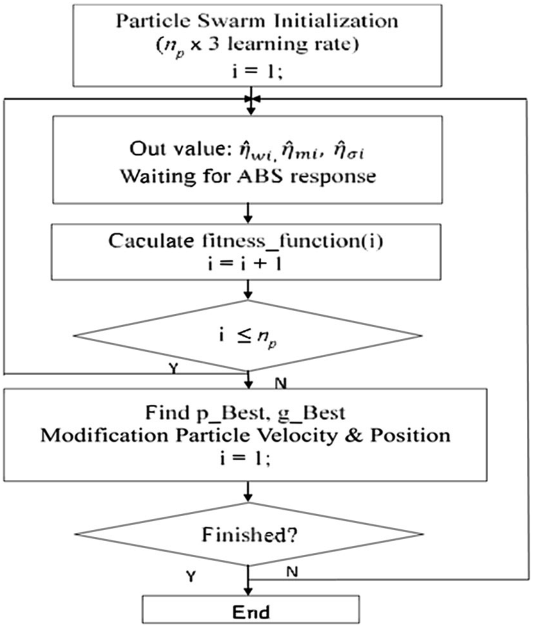
Particle Swarm Initialization
(n_p x 3 learning rate)
i = 1;
Out value: $\hat{\eta}_{wi}, \hat{\eta}_{mi}, \hat{\eta}_{\sigma i}$
Waiting for ABS response
Caculate fitness_function(i)
i = i + 1
i $\le n_p$
Y
N
Find p_Best, g_Best
Modification Particle Velocity & Position
i = 1;
Finished?
Y
N
End

Fig. 5 Flowchart of PSO

Using some straightforward manipulation, the error equation can be obtained

$$\dot{s} = \dot{\lambda}_{\mathrm{e}}(t) + k\lambda_{\mathrm{e}}(t) = \varepsilon(t) - u_R(t) \tag{57}$$

Define a Lyapunov function as

$$V_2\big(s(t),\tilde{E}(t)\big) = \frac{1}{2}s^2(t) + \frac{\tilde{E}^2(t)}{2\eta_D} \tag{58}$$

where η_D is the learning rate of $\hat{E}$.

Take the derivative of (58) and use (56), (57), then

$$\begin{aligned} \dot{V}_2\big(s(t),\tilde{E}(t)\big) &= s(t)\dot{s}(t) + \frac{\tilde{E}(t)\dot{\tilde{E}}(t)}{\eta_D} \\ &= s(t)(\varepsilon(t) - u_R(t)) + \frac{\tilde{E}(t)\dot{\tilde{E}}(t)}{\eta_D} \\ &= s(t)\varepsilon(t) - \hat{E}(t)|s(t)| + \frac{\tilde{E}(t)\dot{\tilde{E}}(t)}{\eta_D} \end{aligned} \tag{59}$$

If the adaptive law for E is chosen as (56) and since E is a constant, so $\dot{\tilde{E}}(t) = -\dot{\hat{E}}(t)$. Then (59) can be rewritten as

$$\begin{aligned} \dot{V}_2\big(s(t),\tilde{E}(t)\big) &= s(t)\varepsilon(t) - \hat{E}(t)|s(t)| - \big(E - \hat{E}(t)\big)|s(t)| \\ &= s(t)\varepsilon(t) - E|s(t)| \le |s(t)||\varepsilon(t)| - E|s(t)| \\ &= -|s(t)|(E - |\varepsilon(t)|) \le 0 \end{aligned} \tag{60}$$

Since $\dot{V}_2\big(s(t),\tilde{E}(t)\big)$ is negative semidefinite, that is $V_2\big(s(t),\tilde{E}(t)\big) \le V\big(s(0),\tilde{E}(0)\big)$, it implies $s(t)$ and $\tilde{E}(t)$ are bounded. Let the function $\Omega \equiv (E - \varepsilon(t))s \le (E - |\varepsilon(t)|)|s| \le -\dot{V}_2\big(s(t),\tilde{E}(t)\big)$ and integrate $\Omega(t)$ with respect to time, then it is easy to obtain

$$\int_0^t \Omega(\tau)\mathrm{d}\tau \le V_2\big(s(0),\tilde{E}(0)\big) - V_2\big(s(t),\tilde{E}(t)\big) \tag{61}$$

Since $V_2\big(s(0),\tilde{E}(0)\big)$ is bounded, and $V_2\big(s(t),\tilde{E}(t)\big)$ is non-increasing and bounded, so from (61) it is obtained

$$\lim_{t\to\infty}\int_0^t \Omega(\tau)d\tau < \infty. \tag{62}$$

Also, $\dot{\Omega}(\tau)$ is bounded, so by Barbalat's lemma [41], $\lim_{t\to\infty}\Omega = 0$. That is, $s(t) \to 0$ as $t \to \infty$. As a result, the PSO-SOT2FNN control system can achieve robust tracking performance.

4 Simulation Results

This section describes the simulation results of ABS control under different road conditions. The parameters of ABS, the initial conditions and the slip command are chosen to be the same as in [3, 6], which is shown in Table 1. Figure 1 clearly shows for different road conditions that the maximal value of the tractive forces is near 20%, so the value $\lambda_c(t)$ is chosen as 0.2. After the transition

Table 1 Model parameters of ABS

Parameter	Value
M_v (kg)	4 × 342
B_v (Ns)	6
J_w (N m s^2)	1.13
R_w (m)	0.33
B_w (Ns)	4
g (m/s^2)	9.8

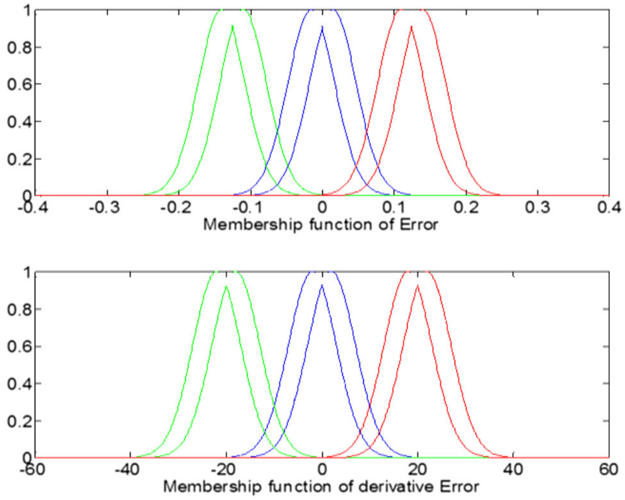

Fig. 6 Membership function for SOT2FNN

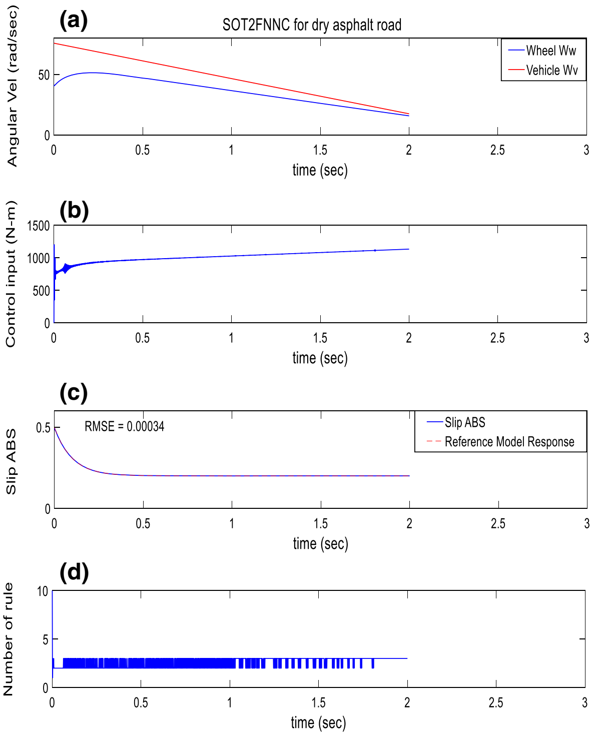

Fig. 7 Simulation result of SOT2FNN control ABS for a dry asphalt road: (**a**) The angular velocity of the wheel and the vehicle, (**b**) Control force, (**c**) Reference model response and ABS response, (**d**) Number of the rule

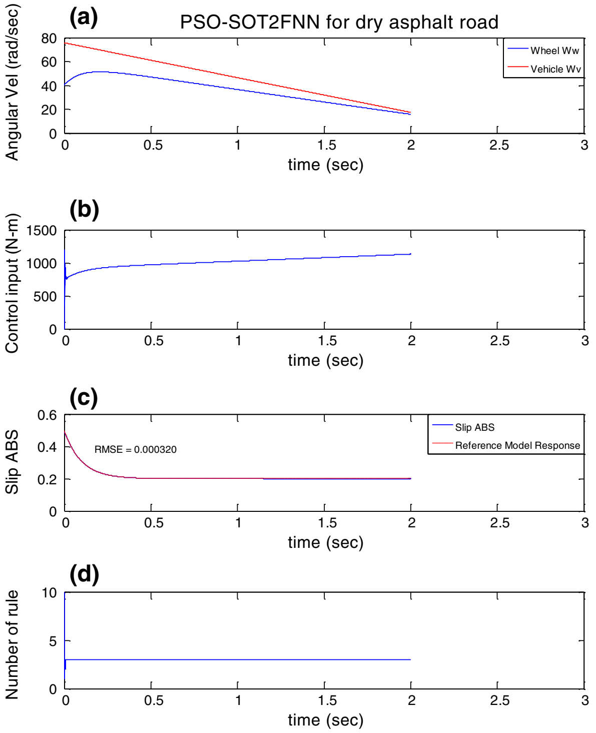

Fig. 8 Simulation result of PSO-SOT2FNN control ABS for a dry asphalt road: (**a**) The angular velocity of the wheel and the vehicle, (**b**) Control force, (**c**) Reference model response and ABS response, (**d**) Number of the rule

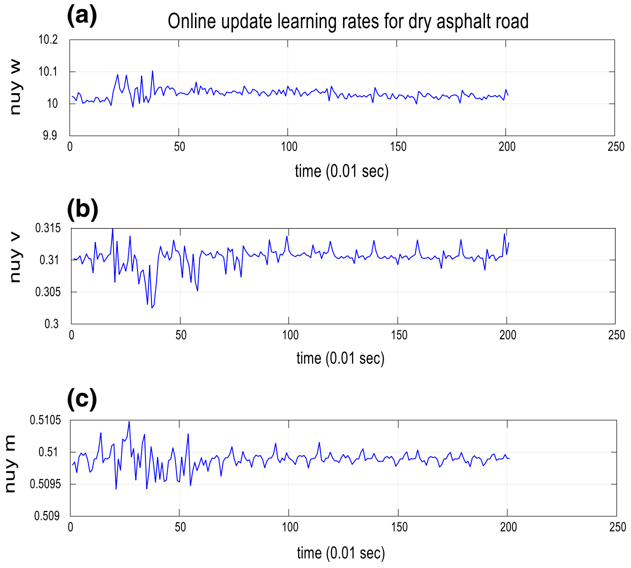

Fig. 9 Using online PSO update learning rates for a dry asphalt road: (**a**) Update learning rates for the weights, (**b**) Update learning rates for the variances, (**c**) Update learning rates for the means

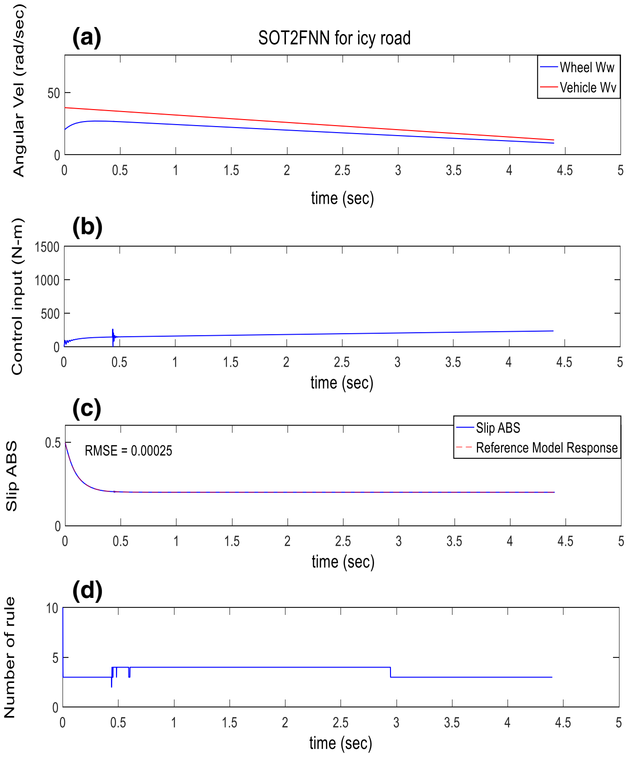

Fig. 10 Simulation result of SOT2FNN control ABS for an icy road: (**a**) The angular velocity of the wheel and the vehicle, (**b**) Control force, (**c**) Reference model response and ABS response, (**d**) Number of the rule

equation $\dot{\lambda}_{\mathrm{d}}(t) = -10\lambda_{\mathrm{d}}(t) + 10\lambda_{\mathrm{c}}(t)$, the desired slip trajectory $\lambda_{\mathrm{d}}(t)$ can be obtained. The maximum braking torque is limited to 1200 N-m. The velocity of a vehicle and

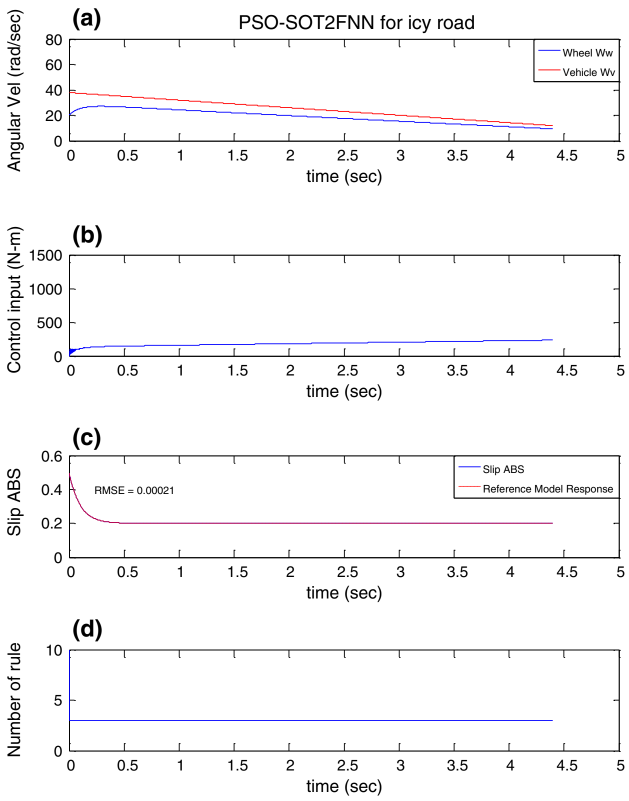

Fig. 11 Simulation result of PSO-SOT2FNN control ABS for an icy road: (**a**) The angular velocity of the wheel and the vehicle, (**b**) Control force, (**c**) Reference model response and ABS response, (**d**) Number of the rule

Table 2 Comparison of the RMSE of the slip tracking error

	Case 1	Case 2	Case 3
NN hybrid control [6]	0.0049	0.0173	0.0168
Intelligent hybrid control [8]	0.0005	0.0023	0.0010
PID-fuzzy [5]	0.0006	0.0028	0.0017
SOT2FNN	0.00034	0.00025	0.00096
PSO-SOT2FNN	0.00032	0.00021	0.00094

Case 1: dry asphalt road, case 2: icy road, case 3: wet asphalt road to icy road

its wheel reaches almost zero at low speed, which means the magnitude of the slip tends to infinity as vehicle speed approaches zero. Therefore, effective control is applied until the vehicle slows to 5 m/s. To conduct the simulations, consider the braking action occurring when the vehicle is moving at a velocity of 25 and 12.5 m/s on an asphalt road and icy road, respectively.

To demonstrate the effectiveness of the PSO algorithm, the simulation results of SOT2FNN and PSO-SOT2FNN are conducted for three different road conditions. The parameters for PSO are $n_p = 20$, $n_d = 3$, $c1 = c2 = 0.07$.

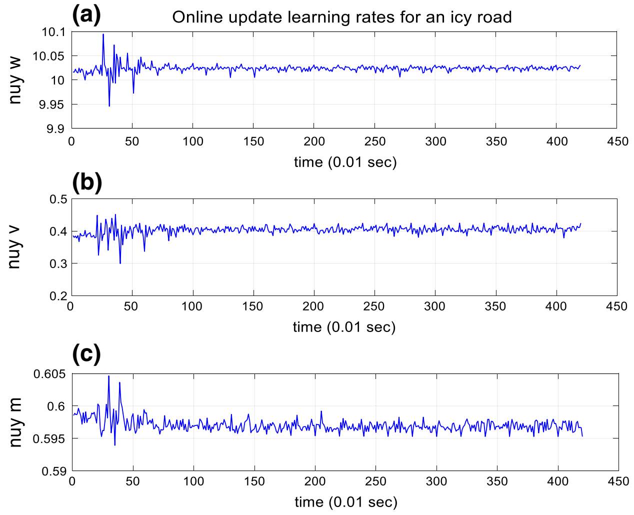

Fig. 12 Using online PSO update learning rates for an icy road: (**a**) Update learning rates for the weights, (**b**) Update learning rates for the variances, (**c**) Update learning rates for the means

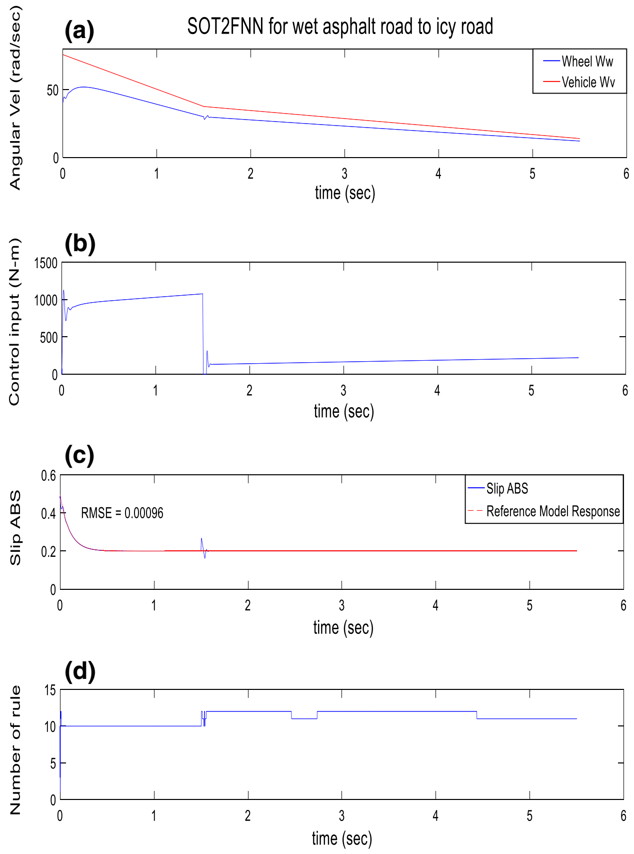

Fig. 13 Simulation result of SOT2FNN control ABS for wet asphalt road to icy road: (**a**) The angular velocity of the wheel and the vehicle, (**b**) Control force, (**c**) Reference model response and ABS response, (**d**) Number of the rule

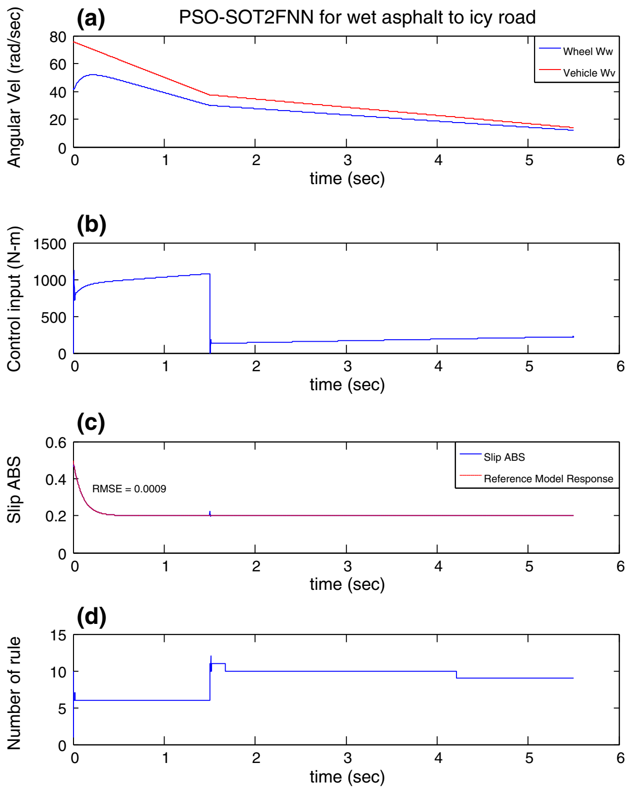

Fig. 14 Simulation result of PSO-SOT2FNN control ABS for wet asphalt road to icy road: (**a**) The angular velocity of the wheel and the vehicle, (**b**) Control force, (**c**) Reference model response and ABS response, (**d**) Number of the rule

The initial Gaussian membership functions are shown in Fig. 6. The ABS simulations for the dry asphalt road and icy road are shown in Figs. 7, 8, 10 and 11, respectively. The simulation results for the transition from a wet asphalt road to an icy road are shown in Figs. 13 and 14. In all figures, part (a) shows the angular velocity of the wheel and the vehicle, part (b) is the control force, part (c) shows the reference model response and the ABS response and part (d) is the change of the rule number. Figures 9, 12 and 15 show the change of the learning rates. From these figures, it is obvious for all road conditions the PSO-SOT2FNN control system can rapidly achieve satisfactory control performance with small RMSE. Figures 7, 10 and 13 show without suitable learning rates that it will lead to many chattering in the control effort and the number of the rule used will be frequently changed. With the online PSO algorithm for updating learning rates, the simulation results in Figs. 8, 11 and 14 show the ability of PSO-SOT2FNN for achieving favorable control performance. Finally, in the same conditions, Table 2 shows the comparison results of RMSE of slip tracking error between the proposed control method and the methods in [5, 6, 8]. This also demonstrates the superiority of the proposed PSO-SOT2FNN over the other control methods (Figs. 8, 11 and 14).

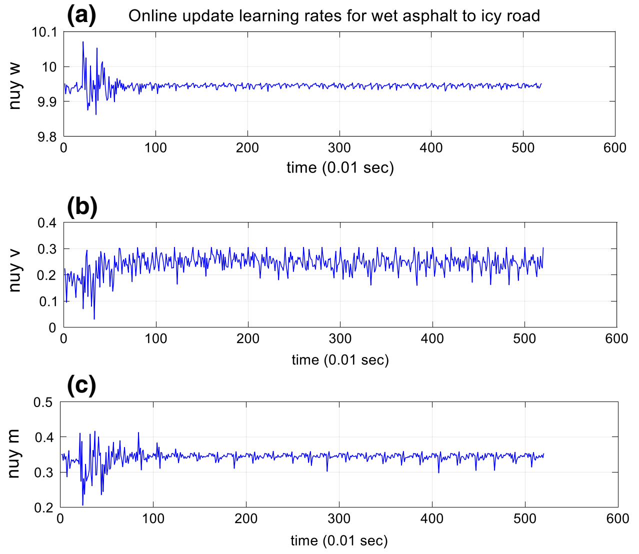

Fig. 15 Using online PSO update learning rates for a wet asphalt road to icy road: (**a**) Update learning rates for the weights, (**b**) Update learning rates for the variances, (**c**) Update learning rates for the means

In our research, the off-line PSO is trained as 500 epochs with 20 sets of learning rates. For case 1, the total elapsed time is 0.395 s and the total time for training is 1.097 h. For case 2, the total elapsed time is 0.682 s and the total time for training is 1.896 h. For case 3, the total elapsed time is 0.948 s and the total time for training is 2.635 h. After off-line training, for the real-time online control, the computing time is 0.000906 and 0.001236 s, respectively, for SOT2FNN and PSO-SOT2FNN at each control iteration. The simulations were done on Windows 7 64-bit and the processor is Core i5-4460 3.2 GHz, RAM 8 GB.

5 Conclusion

In this study, a PSO-SOT2FNN controller combined with a robust compensation controller is proposed to control an ABS, with the performance of the control system being illustrated under various road conditions. The major features of this study include the development of a PSO-SOT2FNN controller with parameter adaptive laws, the PSO algorithm for the relevant learning rates, the effective self-organizing algorithm for suitable construction of IT2FNN and the stability analysis of the control system. The simulation results of ABS control have shown the superiority of the proposed control method than the other control methods. The proposed control method can also be suitable for a large class of unknown nonlinear systems, since this design algorithm does not need to know the exact model of controlled systems. Applying the developed control algorithm to real systems will be our future work.

Acknowledgements The authors appreciate the financial support in part from the Nation Science Council of Republic of China under Grant NSC 101-2221-E-155-026-MY3.

References

1. Dadashnialehi, A., Hadiashar, A., Cao, Z., Kapoor, A.: Intelligent sensorless ABS for in-wheel electric vehicles. IEEE Trans. Ind. Electron. **61**(8), 1957–1969 (2014)
2. Wei, Z., Xuexun, G.: An ABS control strategy for commercial vehicle. IEEE Trans. Mechatron. **20**(1), 384–392 (2015)
3. Lin, C.M., Hsu, C.F.: Self-learning fuzzy sliding-mode control for antilock braking systems. IEEE Trans. Control Syst. Technol. **11**(2), 273–278 (2003)
4. Poursamad, A.: Adaptive feedback linearization control of antilock braking systems using neural networks. Mechatronics **19**, 767–773 (2009)
5. Sharkawy, A.B.: Genetic fuzzy self-tuning PID controllers for antilock braking systems. Eng. Appl. Artif. Intell. **23**, 1041–1052 (2010)
6. Lin, C.M., Hsu, C.F.: Neural-network hybrid control for antilock braking systems. IEEE Trans. Neural Networks **14**(2), 351–359 (2003)
7. Corno, M., Gerard, M., Verhaegen, M., Holweg, E.: Hybrid ABS control using force measurement. IEEE Trans. Control Syst. Technol. **20**(5), 1223–1235 (2012)
8. Lin, C.M., Li, H.Y.: Intelligent hybrid control system design for antilock braking systems using self-organizing function-link fuzzy cerebellar model articulation controller. IEEE Trans. Fuzzy Syst. **21**(6), 1044–1055 (2013)
9. Peric, S.L., Antic, D.S., Milovanovic, M.B., Mitic, D.B., Milojkovic, M.T., Nikolic, S.S.: Quasi-sliding mode control with orthogonal endocrine neural network-based estimator applied in anti-lock braking system. IEEE Trans. Mechatron. **21**(2), 754–764 (2016)
10. Topalov, A.V., Oniz, Y., Kayacan, E., Kaynak, O.: Neuro-fuzzy control of antilock braking system using sliding mode incremental learning algorithm. Neurocomputing **74**, 1883–1893 (2011)
11. Tang, Y., Zhang, X., Zhang, D., Zhao, G., Guan, X.: Fractional order sliding mode controller design for antilock braking systems. Neurocomputing **111**, 122–130 (2013)
12. Albus, J.S.: A new approach to manipulator control: the cerebellar model articulation controller (CMAC). J. Dyn. Syst. Meas. Control **97**(3), 220–227 (1975)
13. Hung, J.Y., Gao, W., Hsu, J.C.: Variable structure control: a survey. IEEE Trans. Ind. Electron. **40**(1), 2–22 (1993)
14. Lin, Y.Y., Liao, S.H., Chang, J.Y., Lin, C.T.: Simplified interval type-2 fuzzy neural networks. IEEE Trans. Neural Netw. Learn. Syst. **25**(5), 959–969 (2014)
15. Juang, C.F., Tsao, Y.W.: A type-2 self-organizing neural fuzzy system and its FPGA implementation. IEEE Trans. Syst. Man Cybern. B Cybern. **38**(6), 1537–1548 (2008)
16. Tao, C.W., Chang, C.W., Taur, J.S.: A simplify type reduction for interval type-2 fuzzy sliding controllers. Int. J. Fuzzy Syst. **15**(4), 460–470 (2013)
17. Chang, Y.H., Chen, C.L., Chan, W.S., Lin, H.W.: Type-2 fuzzy formation control for collision-free multi-robot systems. Int. J. Fuzzy Syst. **15**(4), 435–451 (2013)
18. Ma, X., Wu, P., Zhou, L., Chen, H., Zheng, T., Ge, J.: Approaches based on interval type-2 fuzzy aggregation operators for multiple attribute group decision making. Int. J. Fuzzy Syst. **18**(4), 697–715 (2016)
19. Castillo, O., Castro, J.R., Melin, P., Rodriguez-Diaz, A.: Application of interval type-2 fuzzy neural networks in non-linear identification and time series prediction. Soft Comput. **18**(6), 1213–1224 (2014)
20. Pratama, M., Lu, J., Zhang, G.: Evolving type-2 fuzzy classifier. IEEE Trans. Fuzzy Syst. **24**(3), 574–589 (2016)
21. Pratama, M., Zhang, G., Er, M.J., Anavatti, S.: An incremental type-2 meta-cognitive extreme learning machine. IEEE Trans. Cybern. **47**(2), 339–353 (2016)
22. Sabahi, K., Ghaemi, S., Pezeshki, S.: Gain scheduling technique using MIMO type-2 fuzzy logic system for LFC in restructure power system. Int. J. Fuzzy Syst. (2016). doi:10.1007/s40815-016-0240-7
23. Zadeh, L.A.: Fuzzy sets. Inf. Control **8**, 338–353 (1965)
24. Zadeh, L.A.: The concept of a linguistic variable and its application to approximate reasoning. Inf. Sci. **8**(3), 199–249 (1975)
25. Liang, Q., Mendel, M.: Interval type-2 fuzzy logic systems: theory and design. IEEE Trans. Fuzzy Syst. **8**(5), 535–550 (2000)
26. Farahani, M., Ganjefar, S.: An online trained fuzzy neural network controller to improve stability of power systems. Neurocomputing **162**, 245–255 (2015)
27. Liu, Y.C., Liu, S.Y., Wang, N.: Fully-tuned fuzzy neural network based robust adaptive tracking control of unmanned underwater vehicle with thruster dynamics. Neurocomputing **196**, 1–13 (2016)
28. Castro, J.R., Castillo, O., Melin, P., Rodriguez-Diaz, A.: A hybrid learning algorithm for a class of interval type-2 fuzzy neural networks. Inf. Sci. **179**(13), 2175–2193 (2009)
29. Pan, Y., Er, M.J., Li, X., Yu, H., Gouriveau, R.: Machine health condition prediction via online dynamic fuzzy neural networks. Eng. Appl. Artif. Intell. **35**, 105–113 (2014)
30. Wang, N., Er, M.J., Meng, X.: A fast and accurate online self-organizing scheme for parsimonious fuzzy neural networks. Neurocomputing **72**, 3818–3829 (2009)
31. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. Proc. IEEE Conf. Neural Netw. **4**, 1942–1948 (1995)
32. Lin, C.M., Lin, M.H., Chen, C.W.: SoPC based adaptive PID control system design for magnetic levitation system. IEEE Syst. J. **5**(2), 278–287 (2011)
33. Lin, C.J., Hong, S.J.: The design of neuro-fuzzy networks using particle swarm optimization and recursive singular value decomposition. Neurocomputing **71**, 297–310 (2007)
34. Ranjani, M., Murugesan, P.: Optimal fuzzy controller parameters using PSO for speed control of quasi-Z source DC/DC converter fed drive. Appl. Soft Comput. **27**, 332–356 (2015)
35. Bingül, Z., Karahan, O.: A fuzzy logic controller tuned with PSO for 2 DOF robot trajectory control. Exp. Syst. Appl. **38**(1), 1017–1031 (2011)
36. Kumar, E.V., Raaja, R.S., Jerome, J.: Adaptive PSO for optimal LQR tracking control of 2 DoF laboratory helicopter. Appl. Soft Comput. **41**, 77–90 (2016)
37. Juang, C.F., Tsao, Y.W.: A self-evolving interval type-2 fuzzy neural network with online structure and parameter learning. IEEE Trans. Fuzzy Syst. **16**(6), 1411–1424 (2008)
38. Lin, C.M., Chen, Y.M., Hsueh, C.S.: A self-organizing interval type-2 fuzzy neural network for radar emitter identification. Int. J. Fuzzy Syst. **16**(1), 20–30 (2014)
39. Harned, J., Johnston, L., Scharpf, G.: Measurement of tire brake force characteristics as related to wheel slip (antilock) control system design. SAE Pap. **78**, 909–925 (1969)
40. Mendel, J.M.: Uncertain Rule-Based Fuzzy Logic System: Introduction and New Directions. IEEE Computational Intelligence Magazine. Prentice-Hall, Upper Saddle River (2001)
41. Slotine, J.J.E., Li, W.P.: Applied Nonlinear Control, Englewood Cliffs. Prentice-Hall, Englewood Cliffs (1991)

Chih-Min Lin was born in Taiwan, in 1959. He received the B.S. and M.S. degrees from Department of Control Engineering and the Ph.D. degree from Institute of Electronics Engineering, National Chiao Tung University, Hsinchu, Taiwan, in 1981, 1983 and 1986, respectively. He is currently the Vice President of Yuan Ze University, Taoyuan, Taiwan. His current research interests include fuzzy neural network, cerebellar model articulation controller, intelligent control systems and signal processing. He has published 171 journal papers and 168 conference papers. Dr. Lin was the recipient of an Honor Research Fellow at the University of Auckland, Auckland, New Zealand, from 1997 to 1998. He also serves as an Associate Editor of *IEEE Transactions on Cybernetics* and *IEEE Transactions on Fuzzy Systems*. He is an IEEE Fellow.

Tien-Loc Le was born in Vietnam, in 1985. He received the B.E. degree in electronics and telecommunication engineering from the LacHong University, Vietnam in 2009, where he is currently a lecturer, and M.E. degree in electrical engineering from Hochiminh University of Technical Education, Vietnam, in 2012. He is currently working toward the Ph.D. degree with Department of Electrical Engineering, Yuan Ze University, Taoyuan, Taiwan. His research interest includes intelligent control systems, fuzzy neural network and cerebellar model articulation controller.