IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 9, 2018, accepted October 2, 2018, date of publication October 8, 2018, date of current version October 31, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2874426

Self-Organizing Brain Emotional Learning Controller Network for Intelligent Control System of Mobile Robots

QIUXIA WU[1], CHIH-MIN LIN[1,2], (Fellow, IEEE), WUBING FANG[1], FEI CHAO[1,3], (Member, IEEE), LONGZHI YANG[4], (Senior Member, IEEE), CHANGJING SHANG[3], AND CHANGLE ZHOU[1]

[1]Fujian Province Key Laboratory of Brain-Inspired Computing, Cognitive Science Department, School of Informatics, Xiamen University, Xiamen 361005, China
[2]Department of Electrical Engineering, Yuan Ze University, Taoyuan City 32003, Taiwan
[3]Department of Computer Science, Institute of Mathematics, Physics and Computer Science, Aberystwyth University, Aberystwyth SY23 3DB, U.K.
[4]Department of Computer and Information Sciences, Northumbria University, Newcastle upon Tyne NE1 8ST, U.K.

Corresponding author: Fei Chao (fchao@xmu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grants 61673322, 61673326, and 91746103, in part by the Fundamental Research Funds for the Central Universities under Grant 20720160126, in part by the Natural Science Foundation of Fujian Province of China under Grants 2017J01128 and 2017J01129, and in part by the European Union's Horizon 2020 Research and Innovation Programme through the Marie Sklodowska-Curie under Grant 663830.

ABSTRACT The trajectory tracking ability of mobile robots suffers from uncertain disturbances. This paper proposes an adaptive control system consisting of a new type of self-organizing neural network controller for mobile robot control. The newly designed neural network contains the key mechanisms of a typical brain emotional learning controller network and a self-organizing radial basis function network. In this system, the input values are delivered to a sensory channel and an emotional channel, and the two channels interact with each other to generate the final outputs of the proposed network. The proposed network possesses the ability of online generation and elimination of fuzzy rules to achieve an optimal neural structure. The parameters of the proposed network are online tunable by the brain emotional learning rules and gradient descent method; in addition, the stability analysis theory is used to guarantee the convergence of the proposed controller. In the experimentation, a simulated mobile robot was applied to verify the feasibility and effectiveness of the proposed control system. The comparative study using the cutting-edge neural network-based control systems confirms that the proposed network is capable of producing better control performances with high computational efficiency.

INDEX TERMS Mobile robot, neural network control, self-organizing neural network, brain emotional learning controller network.

I. INTRODUCTION

Autonomous mobile robots or vehicles are very useful in many application fields. Recently, the requirement growth of mobile robots in industrial and cargo applications becomes more and more rapid [1]–[5]. Current intelligent mobile robots broadly cover cutting-edge sciences and technologies of sensors, computer vision, artificial intelligence and other disciplines [6]–[8]. In particular, as a nonholonomic control system, the trajectory tracking problem is a typical and complicated research topic of mobile robots, since solving such problem must deal with a large number of uncertain and nonlinear disturbances [9]–[13]. Several current work suggested to use the robust optimal control for mobile robots [14], [15]. In order to better solve the problem, it is necessary to create a self-adaptive intelligent controller to handle the disturbances. Many modern control methods, such as fuzzy logic based and artificial neural network-based methods, have been utilized to control uncertain nonlinear systems.

The dynamic control of mobile robots faces two major challenges. First, an artificial neural network-based controller in mobile robots must contain enough self-adaptation and non-linear learning abilities. Many research applied adaptive neural network controllers to solve the tracking control problem of mobile robots and other dynamic systems [15]–[20]. Many studies used artificial neural networks as inverse modeling controllers to control mobile robots [5], [21]. However, these studies merely took neural network-based controller's output errors as learning assessments to update network weights. To achieve better performance for controlling mobile robot systems, neural networks also require to consider robot's overall performance to adjust their control parameters. Recently, a number of emotional learning methods [22]–[24], inspired by the learning architectures of human brain [25]–[27], are developed to build robot controllers [28], [29]; e.g., Jafari *et al.*'s [30] work developed and implemented a novel biologically inspired intelligent tracking controller for unmanned aircraft systems in presence of uncertain system dynamics and disturbance. Especially, a brain emotional learning controller network (BELC) not only uses network output errors to adjust its network weights, but also benefits from using the network's emotional output as an overall performance to tune its parameters [31], [32]. However, there is still room to improve the BELC's the non-linear approximation ability, which is limited by its static network structure.

Second, to deal with unexpected and time-varying disturbances in mobile robot systems, neural network-based controllers must be able to fast arrange efficient computational resources. Several existing studies focused on building a incremental structure in their robot neural network control systems; e.g., Chao *et al.*'s [33]–[35] applied a resource allocation network to build robotic hand-eye coordination systems. There studies merely considered the incremental computing neurons, but ignored to remove less important neurons from their neural network based controllers; also, the stabilities of these control systems cannot be guaranteed. Other studies [36]–[39] suggested to use a ''pruning'' mechanism with adaptive learning methods to improve their network's computational efficiency. Indeed, such approaches can control their computational resources. Nevertheless, for controlling mobile robots, it is still a challenging task to quickly react to external disturbances that appears suddenly; therefore, both the computational efficiency and non-linear approximation ability must be satisfied in the network-based controllers of mobile robots.

This paper aims to address both challenges. A new type of self-organizing neural network, called self-organizing brain emotional learning controller network (SOBELC), is developed. The SOBELC network combines the key mechanisms of a fuzzy brain emotional learning controller network (FBELC) [31] and a self-organizing radial basis function network (RBF) [40]. An FBELC network contains a sensory system and a neural network judgement system, which are inspired by human brain's amygdala and orbitofrontal cortices, respectively [31], [41]. In this work, the network judgement system is established by introducing a self-organizing mechanism, which is developed from the RBF network [40]. The weights of the two systems are adjusted based on an emotional cue function [42], [43], which is calculated from the both system's inputs and outputs. After the emotional learning process, the proposed network produces the final outputs by integrating the two systems' outputs. Thus, the proposed network not only retains its non-linear approximation ability, but also possesses the allocation capability of computational resources. With the support of the proposed SOBELC network, this work further develops an intelligent control system for dynamic non-linear control of mobile robots. The experimental results demonstrate the feasibility of the proposed learning method; in addition, the tracking ability of the mobile robot, the stabilization ability, and the robustness are improved to a certain extent. This work is evaluated by a simulated environment; however, various levels of unexpected disturbances are added into the experimental system, so as to simulate real-world applications. If the robot can well handle these simulated disturbances, the SOBELC network-based controller can also have good performances in the real world.

The remainder of this paper is organized as follows. Section II describes the dynamic model of a mobile robot. Section III introduces the detailed implementations of the proposed neural network. Section IV describes the intelligent control system using the proposed neural network for mobile robot control, including a discussion of the controller's update laws. Section V presents the experimental results and comparisons. Finally, Section VI concludes the paper and points out potential future work.

II. DYNAMIC MODEL OF MOBILE ROBOT

A wheeled mobile robot shown in Fig. 1-a, is a typical non-holonomic mechanical system. The robot has two coaxially mounted driven wheels and a passive wheel. The driven wheels are responsible for movements and steering of the mobile robot. Fig. 1-b illustrates a tracking procedure diagram of mobile robots. The tracking procedure is assumed that a task mobile robot can trace a virtual mobile robot (reference robot) that can perfectly move along predefined trajectories. Therefore, the optimal control scenario of the task robot is to enable its position and orientation to be consistent with those of the reference robot. The position and orientation errors between the task robot and the reference robots are used as the robot controller's inputs.

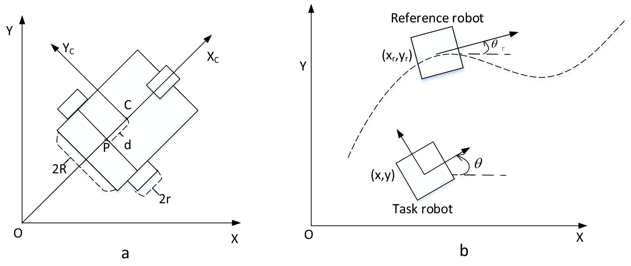


FIGURE 1. Wheeled mobile robot model and tracking procedure. a) A wheeled mobile robot .

In Fig. 1-a, the radius of the driven wheel is r, the distance between two driven wheels is $2R$, P is the center position of the two driven wheel's axis, C is the mobile robot's center of gravity, and d is the distance between P and C. In the Descartes coordinate system, the position of the mobile robot is represented by the vector $q = [x_c \quad y_c \quad \theta]^T$, where x_c and y_c the coordinates of C; and θ is the angle between the coordinate system x_c, y_c and Descartes coordinate system.

In general, the dynamic equation of a nonholonomic mobile robot system with n-dimensional state and m dimension constraints can be expressed as:

$$M(q)\ddot{q} + V_m(q,\dot{q})\dot{q} + G(q) + F(\dot{q}) + \tau_d = B(q)\tau - A(q)\lambda \tag{1}$$

where $\dot{q}$ is a velocity vector of the position and orientation; $\ddot{q}$ is an acceleration vector of the position and orientation; $M(q) \in R^{n\times n}$ is the positive definite symmetric inertia matrix; $V_m(q,\dot{q}) \in R^{n\times n}$ is matrix of the radial force and Costa force; $G(q,\dot{q}) \in R^n$ is gravity matrix; and $F(\dot{q}) \in R^n$ is friction. Here, assume that the mobile moves on the horizontal ground, $G(q) = 0$ and $F(\dot{q}) = 0$; $\tau_d \in R^n$ denotes the bounded unknown disturbances; $B(q) \in R^{n\times n}$ is the input transformation matrix; $\tau \in R^n$ is the control input vector; $A(q)$ is a constraint matrix; and $\lambda \in R^m$ is the restrain force.

Thus, the kinetic parameters of the mobile robot model in Fig. 1-A are obtained by Euler-Lagrange equation; the parameters are defined as follows:

$$M(q) = \begin{bmatrix} m & 0 & md\sin\theta \\ 0 & m & -md\cos\theta \\ md\sin\theta & -md\cos\theta & I \end{bmatrix} \tag{2}$$

$$C(q,\dot{q}) = \begin{bmatrix} 0 & 0 & md\dot{\theta}\sin\theta \\ 0 & 0 & md\dot{\theta}\cos\theta \\ 0 & 0 & 0 \end{bmatrix} \tag{3}$$

$$B(q) = \frac{1}{r}\begin{bmatrix} \cos\theta & \cos\theta \\ \sin\theta & \sin\theta \\ R & -R \end{bmatrix} \tag{4}$$

$$\tau = [\tau_r \quad \tau_l] \tag{5}$$

where m is the weight of the mobile robot; I represents the moment of inertia; and τ_r and τ_l are the torques of the right and left wheels, respectively.

Usually, a general constraint for mobile robots is: The mobile robot's movements only contain pure rolling without slipping [32]. Thus, we have:

$$\dot{x_c}\sin\theta - \dot{y_c}\cos\theta = \dot{\theta}d \tag{6}$$

where d is defined in Fig. 1-A; then, (6) can be rewritten as:

$$A(q)\dot{q} = 0 \tag{7}$$

In this paper, $S(q)$ and $\mathrm{v}(t)$ are defined by:

$$S(q) = \begin{bmatrix} \cos\theta & d\sin\theta \\ \sin\theta & -d\cos\theta \\ 0 & 1 \end{bmatrix} \tag{8}$$

$$\mathrm{v} = \begin{bmatrix} \upsilon \\ \omega \end{bmatrix} \tag{9}$$

where υ is the linear velocities, and ω is the angular velocities. In terms of the nonholonomic constraint mentioned above, the kinematics model is then obtained as:

$$\dot{q} = \begin{bmatrix} \dot{x}_1 \\ \dot{y}_2 \\ \dot{\theta}_3 \end{bmatrix} = \begin{bmatrix} \cos\theta & d\sin\theta \\ -\sin\theta & d\cos\theta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \upsilon \\ \omega \end{bmatrix} = S(q)\mathrm{v}(t) \tag{10}$$

Then, the tracking comparatively error against the position e_p is defined by:

$$e_p = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix}, \tag{11}$$

and tracking comparatively error against the velocity e_v is defined as:

$$e_{\mathrm{v}} = \mathrm{v}_r - \mathrm{v} \begin{bmatrix} \upsilon_r - \upsilon \\ \omega_r - \omega \end{bmatrix} \tag{12}$$

where the reference velocities v_r can be respectively defined as:

$$\mathrm{v}_r = \begin{bmatrix} \upsilon_r\cos e_3 + k_1\, e_1 \\ \omega_r + k_2\upsilon_r e_2 + k_3\upsilon_r\sin e_3 \end{bmatrix} \tag{13}$$

where k_1, k_2, and k_3 are pre-defined parameters; and the definition of v_r is various; in this work, the reference velocity model defined in (13) is selected from Blažič's work [10]. Thus, the velocity error, e_c, can be obtained by $e_c = v_c - v$.

Then, left-multiply $S^t(q)$ to (1), the dynamic equation of the mobile robot is obtained by:

$$\overline{M}\dot{\mathrm{v}}(t) + \overline{V}_m\mathrm{v}(t) + \overline{F} + \overline{\tau}_d = \overline{B}\tau \tag{14}$$

where $\overline{M} = S^TMS$, $\overline{V}_m = S^T(M\dot{S} + V_mS)$, $\overline{\tau}_d = S^T\tau_d$, $\overline{\tau} = \overline{B}\tau$, $\overline{B} = S^TB$; and $\overline{M}$, $\overline{C}$, and $\overline{B}$ are defined by:

$$\overline{M} = \begin{bmatrix} m & 0 \\ 0 & I - md^2 \end{bmatrix}, \quad \overline{C} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad \overline{B} = \begin{bmatrix} \frac{1}{r} & \frac{1}{r} \\ \frac{R}{r} & -\frac{R}{r} \end{bmatrix}. \tag{15}$$

Based on [44], the parameter matrix of (14) has the following properties:

Property 1: $(M - 2V_m)$ is skew-symmetric, i.e.,

$$x^T(M - 2V_m)x = 0, \quad \forall x \neq 0. \tag{16}$$

Then, derivative e_c and substitute it into (14), the dynamic equation against the velocity error is defined by:

$$\overline{M}\dot{e}_c = -\overline{V}_m(q,\dot{q})e_c + \overline{\tau} \tag{17}$$

III. SELF-ORGANIZING BRAIN EMOTIONAL LEARNING CONTROLLER NETWORK

In order to improve the non-linear approximation ability and computational efficiency of the BELC network, a new neural network is established by introducing a self-organizing mechanism to BELC. The resulted network not only takes the advantages of BELC and RBF neural networks in handling uncertain situations, but also enjoys the benefit of self-organizing mechanism for computational resource allocation. The architecture and self-organizing method of the new proposed neural network are described as following subsections.

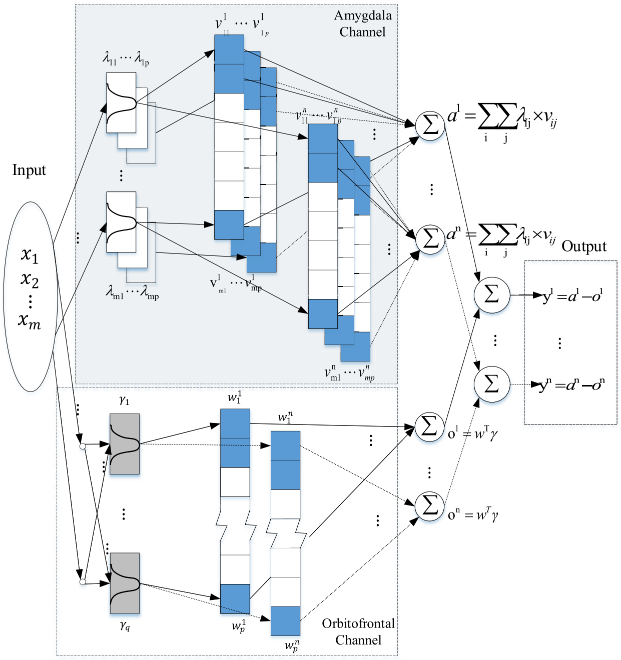

Amygdala Channel
$v^1_{11} \cdots v^1_{1p}$
$\lambda_{11} \cdots \lambda_{1p}$
$v^n_{11} \cdots v^n_{1p}$
Input
x_1
x_2
x_m
$a^1 = \sum_i \sum_j \lambda_{ij} \times v_{ij}$
$a^n = \sum_i \sum_j \lambda_{ij} \times v_{ij}$
Output
$\lambda_{m1} \cdots \lambda_{mp}$
$v^1_{m1} \cdots v^1_{mp}$
$y^1 = a^1 - o^1$
$y^n = a^n - o^n$
$v^n_{m1} \cdots v^n_{mp}$
γ_1
w^1_1
w^n_1
$o^1 = w^T \gamma$
$o^n = w^T \gamma$
γ_q
w^1_p
w^n_p
Orbitofrontal Channel

FIGURE 2. The architecture of the SOBELC network.

A. NEURAL NETWORK IMPLEMENTATION

The network architecture of the proposed SOBELC network shown in Fig. 2. The main structure of SOBELC is inspired by the specification of a BELC network. Inputs of SOBELC are mapped into two channels: a orbitofrontal channel (shown in the upper shade box in Fig. 2) and a amygdala channel. Such two-channel structure mimics that of a human brain, which has an orbitofrontal and an amygdala cortices [31], [45], [46]. The orbitofrontal cortex channel represents an emotional process in the network and the amygdala cortex channel does a sensory-motor process. Each channel also contains a receptive-field space and a weight memory. The receptive-field space calculates the activate level for the weight memory; then, the two weight vectors are aggregated and delivered to the output layer for the generation of the network's final output. The implementation of the proposed neural network is specified as follows.

1) **Input:** The input of a SOBELC network is a continuous multi-dimensional signal. A given input signal is presented as $X = [x_1, \ldots, x_i, \ldots, x_m]^T \in R^m$, where m is the input dimension. Then, X is sent to the orbitofrontal and amygdala channels synchronously.

2) **Orbitofrontal Channel:** This channel is partially implemented by a fuzzy cerebellar mode articulation controller neural network, which is established in [32]. Inputs of the channel are distributed to a number of corresponding receptive fields, within each of which the Gaussian membership function calculates the total firing strength from the network's inputs. Thus, the Gaussian function in this channel is defined as:

$$\lambda_{ij} = exp(\frac{-(x_i - \zeta_{ij})^2}{\sigma_{ij}^2}) \tag{18}$$

where $i \in R^m$, x_i denotes the ith input, $j \in R^p$, j is the jth receptive field and p denotes the number of receptive field's layers; in addition, λ_{ij}, ζ_{ij}, and σ_{ij} denote the membership function, uncertain mean value, and variance value for the jth receptive field of the ith input, respectively.

Each receptive-field is mapped to a corresponding weight. The entire weight space, V_{OC}, in this channel is defined as:

$$V_{OC} = V_{i \times j \times k} = \begin{bmatrix} \nu_1 \\ \vdots \\ \nu_n \end{bmatrix} \tag{19}$$

where k denotes the kth output; thus ν_k is defined as:

$$\nu_k = \begin{bmatrix} \nu_{11} & \ldots & \nu_{1p} \\ \vdots & \ddots & \vdots \\ \nu_{n_i 1} & \ldots & \nu_{n_i p} \end{bmatrix}. \tag{20}$$

3) **Amygdala Channel:** The amygdala channel is established by using a self-organizing radius basis function neural network [36]. The receptive-field layer is composed of a set of RBF neurons; thus, the jth neuron's output is defined as:

$$\Theta_j(\left\| x - \psi_j \right\|, \gamma_j) = exp(\frac{-(x_i - \psi_j^i)^2}{\gamma_j^{i2}}) \tag{21}$$

where $i \in R^m$; in addition, ψ_j^i and γ_j^i denote the center and width of the jth neucon of the ith input. Note that, in the Orbitofrontal channel, the number of receptive fields is static; however, the number of neurons in the amygdala channel is dynamic and is adjusted during the adaptive control process. The self-organizing rule is defined in Section III-B.

Each neuron links to a weight value in the weight space, W_{AC}, which is defined as:

$$W_{AC} = W_{q \times n_k} = \begin{bmatrix} w_{11} & \ldots & w_{1n_k} \\ \vdots & \ddots & \vdots \\ w_{q1} & \ldots & w_{qn_k} \end{bmatrix}. \tag{22}$$

4) **Output:** Outputs of the two channels, u^k_{OC} and u^k_{AC}, are presented as:

$$u^k_{OC} = V^T_{OC} \Lambda(\mathrm{x}, \zeta, \sigma) \tag{23}$$
$$u^k_{AC} = W^T_{AC} \Theta(\mathrm{x}, \psi, \gamma) \tag{24}$$

where $k \in R^n$. Based on the BELC structure, the output of SOBELC is $y^k = u^k_{OC} - u^k_{AC}$; thus, the output can be further presented as:

$$y^k = V^T_{OC} \Lambda(\mathrm{x}, \zeta, \sigma) - W^T_{AC} \Theta(\mathrm{x}, \psi, \gamma) \tag{25}$$

B. SELF-ORGANIZATION MECHANISM

In the on-line learning process, the number of neurons in the amygdala channel can dynamic change. If the number of neurons in the hidden layer, κ, is too small, the network's process ability may not be efficient for control tasks. Conversely, if the number is too large, the computational cost is too heavy, so that it is might be unsuitable for online real-time control. In order to find a balance between the network performance and the computational cost, an online learning algorithm is established to increase or decrease the number of neurons in SOBELC. Based on this consideration, the on-line learning algorithm consists of two steps of process: increasing process and decreasing process, which are shown in Fig. 3 and specified as follows:

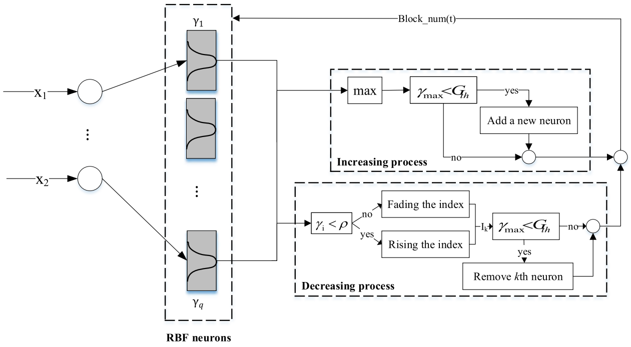


FIGURE 3. Self-organizing mechanism for SOBELC network. The algorithm consists of two steps of process: increasing process and decreasing process.

1) NEURON INCREASING

First of all, the leaning algorithm is to decide whether or not to add a new neuron into the amygdala channel. For each input data, x_i, the activation values of the amygdala channel's existing hidden neurons is used to represent x_i's membership degree that the input data belongs to the existing neurons. If the current activation value is low, this situation indicates that the existing neural neurons are not sensitive to the input data, more processing neurons are required in the amygdala channel; otherwise, if the value is high, it is not necessary to add a new neuron. To simplify the calculation, the maximum activation value of the existing neurons is used to represent the activation grade of the all neurons. Thus, the maximum value, Θ_{max}, is defined as:

$$\Theta_{max} = \max_{1 \le j \le \kappa(t)} \Theta_j \tag{26}$$

where $q(t)$ is the number of the existing neurons at the t-th time.

A pre-defined threshold, $G_{th} \in (0, 1)$, determines whether a new neuron can be added. Thus, if $\gamma_{max} \le G_{th}$, then a new hidden neuron is created and added in the hidden layer in the amygdala channel. The new neuron's parameters, (ψ_q^{new}, γ_q^{new}, and w^{new}), are initialized as:

$$\begin{cases} \psi_\kappa^{new} = x_t \\ \gamma_\kappa^{new} = \overline{\gamma} \\ w^{new} = 0 \\ \kappa_{t+1}^{new} = \kappa_t + 1 \end{cases} \tag{27}$$

where x_t is the new incoming data at the t-th time, $\overline{\gamma}$ is the mean width of radial basis function of the existing neurons, and w^{new} is easily set at 0.

2) NEURON DECREASING

To save the computational cost, the on-line learning algorithm can remove existing neural neuron, whose processing results cannot impact the overall results of the entire network. Thus, a neuron's significance, I, is defined to determine whether a neuron must be removed. Therefore, if the j-th neuron's significance, I_j, is larger than a threshold, P_{th}, then the j-th neuron is retained in the hidden layer. Otherwise, if $I_j \le P_{th}$, the j-th neuron must be removed from its network. I_j can be updated by using the status of Θ_j: If Θ_j is smaller than elimination threshold value, ρ, then I_j will have a slight decrease; otherwise, I_j will have a small increase. The updating rule of I_j is summarized as follows:

$$I_j(t+1) = \begin{cases} I_j(t)\exp(-\tau_1) & \text{if } \Theta_j < \rho \\ I_j(t)[2 - \exp(-\tau_2(1 - I_j(t)))] & \text{if } \Theta_j \ge \rho \end{cases} \tag{28}$$

where the initial value of each I is set at 1; $j \in [1, q]$ is the number of current neurons; and τ_1 and τ_2 are two pre-defined constant values. The complete algorithm is summarized as a pseudo-code shown in Algorithm 1.

Algorithm 1 Self-Organization Mechanism

```
Initialize $G_{th} = 0.1$, $\rho = 0.1$, $\tau_1 = 0.01$, and $\tau_2 = 0.05$;
Calculate $\Theta_{max}$;
if $\Theta_{max} \le G_{th}$ then
    Set $\gamma_{\kappa+1}^{new}$, $q(t+1)$ by (27);
end if
for $j = 0$ to $\kappa(t)$ do
    Update $I_j$ by Eqn. 28;
    if $I_j < P_{th}$ then
        Remove $\psi_j$, $\gamma_j$, and $w_j$;
        Update $\kappa$;
    end if
end for
```

C. PARAMETER UPDATE

The weights in the orbitofrontal channel are updated based on the brain emotional learning rule, the weight update values, $\dot{v}_{iq}$, are defined by:

$$\dot{v}_{iq} = \eta_z(h_i \times \max[0, d_q - u_{OC}^q]) \tag{29}$$

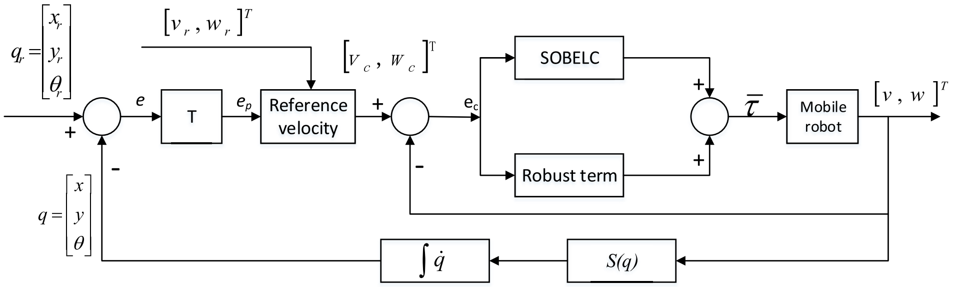


FIGURE 4. Wheeled mobile robot control and tracking system.

where η_z denotes a learning rate; and d_q denote an emotional cue parameter, defined by:

$$d_q = \sum_{i=1}^{m} \beta_{iq} \times SI_i + c_q \times y_k \tag{30}$$

where β_{iq} and c_q are gain parameters, which are determined in practical control problems. Thus, the updating law for the orbitofrontal channel's weights is defined as:

$$v_{iq}(t+1) = v_{iq}(t) + \dot{v_{iq}}. \tag{31}$$

The tunable parameters of the amygdala channel are w, ψ, and γ; therefore, to obtain more robust performance, the parameters are updated by Lyapunov stability analysis theory, rather than the brain emotional method. The updating values, $\dot{w}$, $\dot{\psi}$ and $\dot{\gamma}$, are described in Section IV and the updating laws of the amygdala channel are defined as:

$$w_{ijk}(t+1) = w_{ijk}(t) + \dot{w}_{ijk} \tag{32}$$

$$\hat{\psi}_{ijk}(t+1) = \hat{\psi}_{ijk}(t) + \dot{\hat{\psi}}_{ijk} \tag{33}$$

$$\hat{\gamma}_{ijk}(t+1) = \hat{\gamma}_{ijk}(t) + \dot{\hat{\gamma}}_{ijk} \tag{34}$$

IV. NEURAL NETWORK CONTROL SYSTEM

The proposed SOBELC network is used to form a new network controller for mobile robot control problems. The structure of the proposed controller is illustrated in Fig. 4. The control system is designed based on Jin and Wang's work [47]. The errors, e_p, are the position differences between the reference robot and the task robot. Combing with the reference velocity model listed in Section II, and the velocity reference error, e_c, is used as the controller's input. The controller minimizes e_c and produce control values as system outputs. The controller is comprised of two sub-systems, including a SOBELC network, and a baseline robust controller. The input error values are fed into the two sub-systems for control signal generation. The control signals generated from both controllers are then aggregated to produce the final output of the overall control system. Thus, the output of the entire controller is computed by:

$$u = u_{SOBELC} + u_r \tag{35}$$

where u_{SOBELC} and u_r denote the outputs of the SOBELC network and robust controllers, respectively.

Assume that there exits an ideal SOBELC controller, u^*_{SOBELC}, to approximate the target u^*, which is presented as follows:

$$\begin{aligned} u^* &= u^*_{SOBELC}(w^*, \psi^*, \gamma^*) + \varepsilon \\ &= V^{*T}\hat{\Lambda} - W^{*T}\Theta^* + \varepsilon \end{aligned} \tag{36}$$

where ε is an approximation error, and w^*, ψ^*, and γ^* are the optimal parameters of u^*_{SOBELC}. However, since the optimal nonlinear approximation u^*_{SOBELC} cannot be obtained, the online estimating $\hat{u}_{SOBELC}$ must estimate the optimal u^*_{SOBELC}. Therefore, the approximate optimal value of the network-based controller can be defined as:

$$\begin{aligned} \hat{u} &= \hat{u}_{SOBELC}(\hat{w}, \hat{\psi}, \hat{\gamma}) + u_r \\ &= \hat{V}^T\hat{\Lambda} - \hat{W}^T\hat{\Theta} + u_r \end{aligned} \tag{37}$$

where $\hat{w}$, $\hat{\psi}$, and $\hat{\gamma}$ are the estimated value of w^*, ψ^*, and γ^*, respectively; and u_r denotes output of the robust controller used to eliminate the error between the ideal controller u^* and the actual controller $\hat{u}_{SOBELC}$. Consider both (36) and (37), the approximation difference, $\tilde{u}$, between u^* and $\hat{u}$ can be defined as:

$$\begin{aligned} \tilde{u} &\equiv u^* - \hat{u} \\ &= V^{*T}\hat{\Lambda} - W^{*T}\Theta^* + \varepsilon - \hat{V}^T\hat{\Lambda} + \hat{W}^T\hat{\Theta} - u_r \\ &= \tilde{V}^T\hat{\Lambda} - \tilde{W}^T\Theta^* - \hat{W}^T\tilde{\Theta} + \varepsilon - u_r \end{aligned} \tag{38}$$

where $\tilde{W} = W^* - \hat{W}$, $\tilde{V} = V^* - \hat{V}$ and $\tilde{\Theta} = \Theta^* - \hat{\Theta}$. The Taylor linearization method is applied to expand the Gaussian functions into partially linear form; thus, the expansion of $\hat{\Theta}$ in a Taylor series can be obtained by:

$$\tilde{\Theta} = \begin{bmatrix} \tilde{\Theta}_1 \\ \vdots \\ \tilde{\Theta}_l \\ \vdots \\ \tilde{\Theta}_{n_l} \end{bmatrix} = \begin{bmatrix} \left(\dfrac{\partial\Theta_1}{\partial\psi}\right)^T \\ \vdots \\ \left(\dfrac{\partial\Theta_l}{\partial\psi}\right)^T \\ \vdots \\ \left(\dfrac{\partial\Theta_{n_l}}{\partial\psi}\right)^T \end{bmatrix} \Bigg|_{\psi=\hat{\psi}} (\psi^* - \hat{\psi})$$

$$+\left.\begin{bmatrix}\left(\frac{\partial\Theta_1}{\partial\gamma}\right)^T\\ \vdots\\ \left(\frac{\partial\Theta_l}{\partial\gamma}\right)^T\\ \vdots\\ \left(\frac{\partial\Theta_{n_l}}{\partial\gamma}\right)^T\end{bmatrix}\right|_{\gamma=\hat{\gamma}}(\gamma^*-\hat{\gamma})+\mathrm{O}_t$$
$$=\Theta_\psi^T\tilde{\psi}+\Theta_\gamma^T\tilde{\gamma}+\mathrm{O}_t \tag{39}$$

where $\tilde{\psi}=\psi^*-\hat{\psi}$, $\tilde{\gamma}=\gamma^*-\hat{\gamma}$ and $O_t\in R^{n_l}$ is a vector of higher-order terms; $\frac{\partial\Theta_l}{\partial\psi}$ and $\frac{\partial\Theta_l}{\partial\gamma}$ are defined as:

$$\left[\frac{\partial\Theta_l}{\partial\psi}\right]=\Bigg[\underbrace{0,\ldots,0}_{(l-1)\times n_i},\frac{\partial r_l}{\partial\psi_{1l}},\ldots,\frac{\partial\Theta_l}{\partial\psi_{n_il}},\underbrace{0,\ldots,0}_{(n_l-l)\times n_i}\Bigg]^T \tag{40}$$

$$\left[\frac{\partial\Theta_l}{\partial\gamma}\right]=\Bigg[\underbrace{0,\ldots,0}_{(l-1)\times n_i},\frac{\partial\Theta_l}{\partial\gamma_{1l}},\ldots,\frac{\partial\Theta_l}{\partial\gamma_{n_il}},\underbrace{0,\ldots,0}_{(n_l-l)\times n_i}\Bigg]^T. \tag{41}$$

Rewrite (39), we have:

$$\Theta^*=\hat{\Theta}+\Theta_\psi^T\tilde{\psi}+\Theta_\gamma^T\tilde{\gamma}+O_t \tag{42}$$

Substituting (42) and (39) into (38), yield

$$\begin{aligned}\tilde{u}&=\tilde{V}^T\hat{\Lambda}-\tilde{W}^T(\hat{\Theta}+\Theta_\psi\tilde{\psi}+\Theta_\gamma\tilde{\gamma}+O_t)\\&\quad-\hat{W}^T(\Theta_\psi\tilde{\psi}+\Theta_\gamma\tilde{\gamma}+O_t)+\varepsilon-u_r\\&=\tilde{V}^T\hat{\Lambda}-\tilde{W}^T\hat{\Theta}-\hat{W}^T(\Theta_\psi\tilde{\psi}+\Theta_\gamma\tilde{\gamma})-u_r+\xi\end{aligned} \tag{43}$$

where ξ denotes the approximation error term, and $\xi=\tilde{W}^T\Theta_\psi^T\tilde{\psi}+\hat{W}^T\Theta_\gamma^T\tilde{\gamma}+W^{*T}O_t+\varepsilon$. ξ is supposed to be bounded by $0\le|\xi|_\infty\le\xi_p$, in which $\xi_{P_{2\times1}}$ is a positive constant matrix.

The robust control is designed as follows:

$$u_r=\frac{(I+\Lambda^2)R^2+I}{2R^2}e_c^T \tag{44}$$

where R is a positive diagonal matrix, $R=diag(\phi_1,\phi_2,\cdots,\phi_i)$, ϕ_i is a robust attenuation coefficient that is specified by designers.

In order to guarantee the entire control system can retain convergence, the parameters updating values must be determined by using the Lyapunov stability theory. The proposed control system shown in Fig. 4 convert the error of the mobile robot's position and orientation to the reference velocity error. Therefore, a Lyapunov function is defined as:

$$L=L_1(e_p,t)+L_2(e_c,t) \tag{45}$$

where L_1 and L_2 are presented as follow:

$$L_1(e_p,t)=k_1(e_x^2+e_y^2)+\frac{2k_1}{k_2}(1-\cos e_\theta) \tag{46}$$

$$\begin{aligned}L_2(e_c,t)=\frac{1}{2}[&e_c^T\bar{M}e_c+tr[\tilde{W}^T\eta_W^{-1}\tilde{W}]+\tilde{\psi}^T\eta_\psi^{-1}\tilde{\psi}\\&+\tilde{\gamma}^T\eta_\gamma^{-1}\tilde{\gamma}+tr[\tilde{V}^T\alpha^{-1}\tilde{V}]]\end{aligned} \tag{47}$$

where $e_p=[e_x,e_y,e_\theta]^T$, and $e_c=[e_v,e_w]^T$. The proof of asymptotic stability of $L_1(e_p,t)$ is given in Blažič's work [10]. Therefore, if $L_2(e_c,t)$ achieves stable, the entire control system can be guaranteed. Derivative (47) and substitute (17) and (43) to the derivation (Note that: the estimation error of the derived torque $\bar{\tilde{\tau}}$ is the estimation error $\tilde{u}$), then we have:

$$\begin{aligned}\dot{L_2}&=e_c^T\bar{M}\dot{e}_c+\frac{1}{2}[e_c^T\dot{\bar{M}}e_c]+tr[\tilde{W}^T\eta_W^{-1}\dot{\tilde{W}}]\\&\quad+\tilde{\psi}^T\eta_\psi^{-1}\dot{\tilde{\psi}}+\tilde{\gamma}^T\eta_\gamma^{-1}\dot{\tilde{\gamma}}+tr[\tilde{V}^T\alpha^{-1}\dot{\tilde{V}}]\\&=e_c^T(-\bar{V}_me_c+\bar{\tilde{\tau}})+\frac{1}{2}[e_c^T\dot{\bar{M}}e_c]+tr[\tilde{W}^T\eta_W^{-1}\dot{\tilde{W}}]\\&\quad+\tilde{\psi}^T\eta_\psi^{-1}\dot{\tilde{\psi}}+\tilde{\gamma}^T\eta_\gamma^{-1}\dot{\tilde{\gamma}}+tr[\tilde{V}^T\alpha^{-1}\dot{\tilde{V}}]\\&=\frac{1}{2}e_c^T(\dot{\bar{M}}-2\bar{V}_m)e_c+e_c^T\bar{\tilde{\tau}}+tr[\tilde{W}^T\eta_W^{-1}\dot{\tilde{W}}]\\&\quad+\tilde{\psi}^T\eta_\psi^{-1}\dot{\tilde{\psi}}+\tilde{\gamma}^T\eta_\gamma^{-1}\dot{\tilde{\gamma}}+tr[\tilde{V}^T\alpha^{-1}\dot{\tilde{V}}]\\&=\frac{1}{2}e_c^T(\dot{\bar{M}}-2\bar{V}_m)e_c+e_c^T\tilde{u}-tr[\tilde{W}^T\eta_W^{-1}\dot{\hat{W}}]\\&\quad-\tilde{\psi}^T\eta_\psi^{-1}\dot{\hat{\psi}}-\tilde{\gamma}^T\eta_\gamma^{-1}\dot{\hat{\gamma}}-tr[\tilde{V}^T\alpha^{-1}\dot{\hat{V}}]\\&=e_c^T\tilde{V}\hat{\Lambda}-e_c^T\tilde{W}\hat{\Theta}-e_c^T\hat{W}(\Theta_\psi\tilde{\psi}+\Theta_\gamma\tilde{\gamma})+e_c^T(\xi-u_r)\\&\quad-tr[\tilde{W}^T\eta_W^{-1}\dot{\hat{W}}]-\tilde{\psi}^T\eta_\psi^{-1}\dot{\hat{\psi}}-\tilde{\gamma}^T\eta_\gamma^{-1}\dot{\hat{\gamma}}-tr[\tilde{V}^T\alpha^{-1}\dot{\hat{V}}]\\&\le-tr[\tilde{W}(e_c^T\hat{\Theta}+\eta_W^{-1}\dot{\hat{W}})]-\tilde{\psi}[e_c^T\hat{W}\Theta_\psi+\eta_\psi^{-1}\dot{\hat{\psi}}]\\&\quad-\tilde{\gamma}[e_c^T\hat{W}\Theta_\gamma+\eta_\gamma^{-1}\dot{\hat{\gamma}}]+e_c^T\tilde{V}\hat{\Lambda}+e_c^T(\xi-u_r)\end{aligned} \tag{48}$$

Since $\dot{\hat{V}}=0$ when $d_q-u_{OC}^q\le0$ and $\dot{\hat{V}}=\eta_z\cdot\Lambda\cdot[d_q-u_{OC}^q]>0$ if $d_q-u_{OC}^q>0$, we have $-tr[\tilde{V}^T\alpha^{-1}\dot{\hat{V}}]\le0$.

Based on (48), the updating values of $\dot{w}$, $\dot{\psi}$ and $\dot{\gamma}$ are defined as:

$$\dot{\hat{W}}=-\eta_W\mathrm{e}_c^T\hat{\Theta} \tag{49}$$
$$\dot{\hat{\psi}}=-\eta_\psi\mathrm{e}_c^T\Theta_\psi\hat{W} \tag{50}$$
$$\dot{\hat{\gamma}}=-\eta_\gamma\mathrm{e}_c^T\Theta_\gamma\hat{W} \tag{51}$$

By using the updating laws in (49), (50), and (51), and the robust controller's definition (44), (48) can be rewritten as:

$$\begin{aligned}\dot{L_2}&\le e_c^T\tilde{V}\hat{\Lambda}+e_c^T(\xi-u_r)\\&=e_c^T\tilde{V}\hat{\Lambda}+e_c^T\xi-\frac{1}{2}e_c^Te_c-\frac{1}{2}\frac{e_c^Te_c}{r^2}-\frac{1}{2}e_c^Te_c\hat{\Lambda}\hat{\Lambda}^T\\&=-\frac{1}{2}e_c^Te_c-\frac{1}{2}[\frac{e_c}{r}-r\xi]^2-\frac{1}{2}[e_c^T\hat{\Lambda}-\tilde{V}]^2\\&\quad+\frac{1}{2}r^2\xi^2+\frac{1}{2}\tilde{V}^T\tilde{V}\\&\le-\frac{1}{2}e_c^Te_c+\frac{1}{2}r^2\xi^2+\frac{1}{2}\tilde{V}^T\tilde{V}\end{aligned} \tag{52}$$

Integrate (52) from $t=0$ to $t=T$, then:

$$\int_0^T L_2dt\le\int_0^T\sum_{i=1}^{n}\left[-\frac{e_{ci}^2(t)}{2}+\frac{r_i^2e_{ci}^2(t)}{2}+\frac{\tilde{v}_i^2}{2}\right]dt. \tag{53}$$

Then, (53) is rewritten as:

$$L_2(T)-L_2(0) \le \sum_{i=0}^{n}\left[-\frac{1}{2}\int_0^T e_{ci}(t)dt+\frac{r_i^2}{2}\int_0^T \xi_i^2(t)dt+\frac{1}{2}\int_0^T \tilde{\nu}_i(t)dt\right] \tag{54}$$

Since $L_2(T) \ge 0$, (54) is simplified as:

$$\frac{1}{2}\sum_{i=0}^{n}\int_0^T e_{ci}^2(t)dt \le L_2(0)+\frac{1}{2}\sum_{i=0}^{n} r_i^2\int_0^T \xi_i^2(t)dt + \frac{1}{2}\sum_{i=0}^{m}\int_0^T \tilde{\nu}_i(t)dt. \tag{55}$$

Then, we have:

$$\sum_{i=0}^{n}\int_0^T e_{ci}^2(t)dt \le \sum_{i=0}^{n} e_{ci}^2(0)+\sum_{i=0}^{n} r_i^2\int_0^T \xi_i^2(t)dt + \frac{1}{2}\sum_{i=0}^{m}\int_0^T \tilde{\nu}_i(t)dt. \tag{56}$$

If $\int_0^T \xi_i^2(t)dt < \infty$, $\int_0^T \nu_i^2(t)dt < \infty$, then for all T, there exists $\int_0^T e_{ci}^2(t)dt < \infty$; therefore, the asymptotic stability of L_2 has been proved to be completed.

In summary, as $t \rightarrow \infty$, $\{e_p, e_c\} \rightarrow 0$, and $\{\dot{e}_p, \dot{e}_c\} \rightarrow 0$; the tracking system is asymptotically stable proved by the Lyapunov stability theory.

V. EXPERIMENTATIONS

A. EXPERIMENTAL SETUP

The proposed controller with the new SOBELC is applied to a mobile robot system, so as to verify the controller's effectiveness and efficacy. The experiments are based on a simulation mobile robot, which tracks a predefined reference trajectory. The reference trajectory has two patterns; the first pattern with the time $0 \le t < 65$ is designed as:

$$\begin{cases} x_r = \upsilon_r \cdot \cos(\omega) \\ y_r = \upsilon_r \cdot \sin(\omega) \end{cases} \tag{57}$$

and the second pattern with the time $65 < t \le 130$ is defined as:

$$\begin{cases} x_r = \upsilon_r \cdot \cos(2\omega) \\ y_r = \upsilon_r \cdot \sin(\omega) \end{cases} \tag{58}$$

where the initial velocity of the reference robot is set as: $\upsilon_r = 0.2m/s$ and $\omega_r = 0.1rad/s$ and that of the task robot is set as the identical value. In addition, the initial position and orientation of the reference mobile robot are set as $qr = [2 \quad 0 \quad \pi/2]^T$ and those of the tracking one are set as $q = [1 \quad 0 \quad \pi/2]^T$.

In the experiments, the task mobile robot's parameters are set as: $m = 10\ kg$, $I = 5kg \cdot m^2$, $R = 0.2m$, $r = 0.05m$, $d = 0.0m$, $F(\dot{q}) = 0$. In addition, the disturbance, $\overline{\tau}_d$, is defined as:

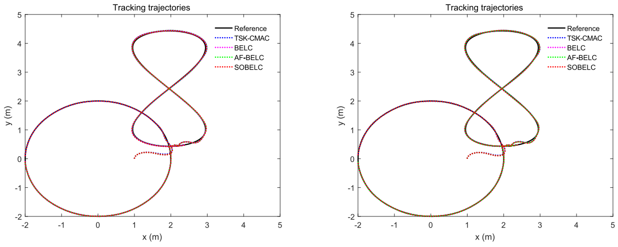
$$\overline{\tau}_d = \begin{bmatrix} \delta \sin(4t) \\ \delta \cos(4t) \end{bmatrix}. \tag{59}$$

where δ denotes the level of disturbance. In this experiment, two levels of disturbances are used to evaluate the proposed network controller; thus, δ is set at 10 and 20, respectively.

In order to compare the effectiveness of the proposed method, a TSK CMAC" network based controller [48] (labeled as "TSK-CMAC"), a fuzzy BEL network based controller [31] (labeled as "BELC"), and an adaptive BEL network based controller [43] (labeled as "AF-BELC") are included in the same experiments. The parameters of the SOBELC neural network model are selected as: $G_{th} = 0.5$, $P_{th} = 0.1$, $\rho = 0.1$, $\kappa = 9$, $\tau_1 = 0.01$, and $\tau_2 = 0.05$. The following paragraphs describe three experiments, each of which applies one disturbance level.

B. RESULTS

Fig. 5 demonstrates the simulated position response of the mobile robot under the three levels of disturbances. Each of the sub-figures contains the reference trajectory (red solid line), the TSK-CMAC output trajectory (blue dotted line), the BELC output trajectory (pink dotted line), the adaptive BELC output trajectory (green dotted line), and the proposed SOBELC output trajectory (red dotted line). In the experiment, the tracking time-length is $130s$; however, from $0s$ to $65s$, the mobile robot is required to track the circular trajectory in the down-left of the sub-figure; then, from $65s$ to $130s$, the robot must track the trajectory similar to digit number "8".

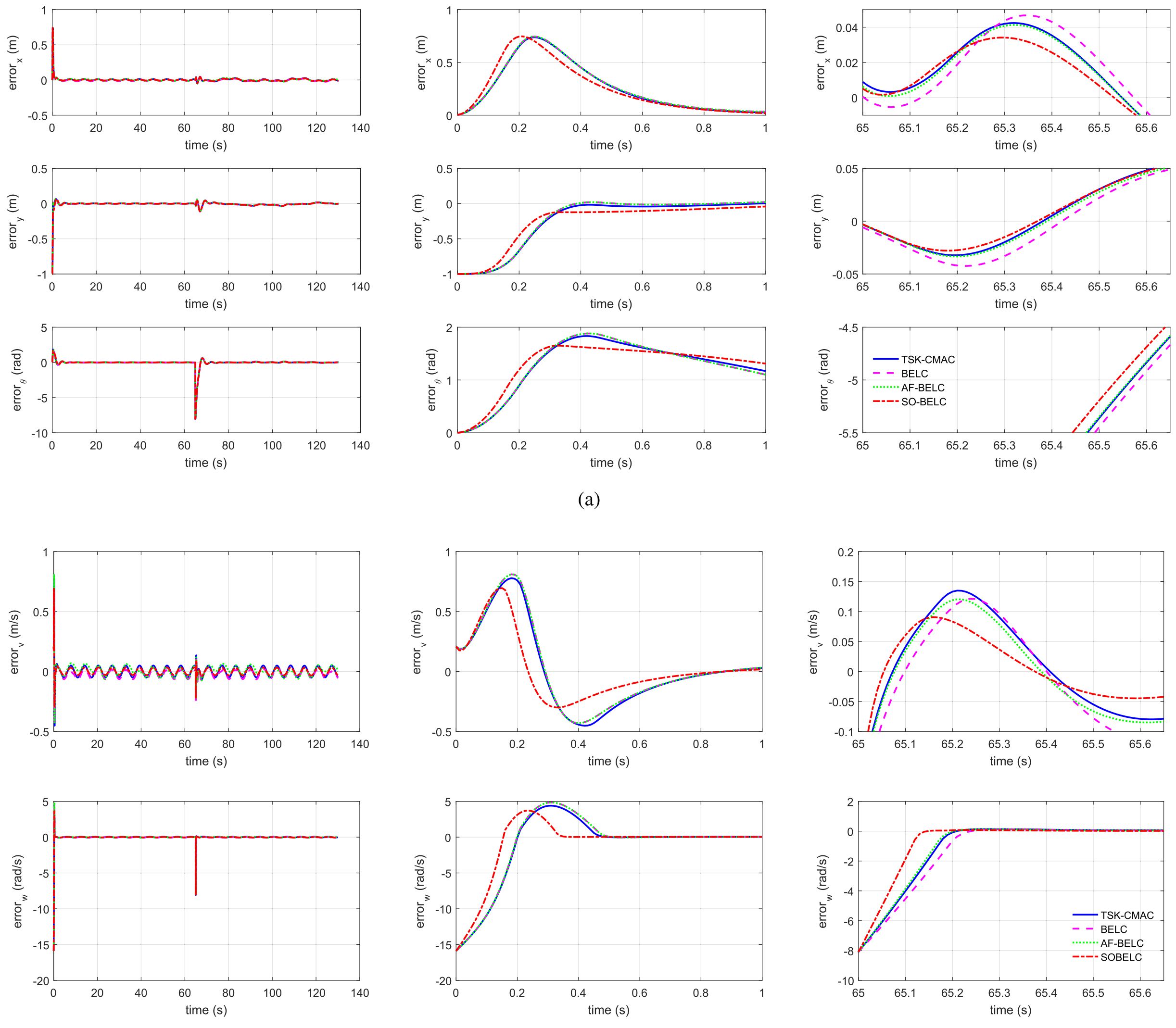


(a) (b)

FIGURE 5. Trajectories of the mobile robot for the four controllers with τ_d : δ = 10 and 20. (a) τ_d : δ = 10. (b) τ_d : δ = 20.

In Fig. 5, the performances of the four controllers are very close to each other; their output trajectories are almost coincided and closely reach the reference trajectory. This situation proves that the entire neural network-based controller possesses strong applicability and generality. However, in order to clearly identify the tracking performance of the four controllers, the position and velocity errors of the controllers with various disturbances at around $0s$ and $65s$ are magnified and shown in Figs. 6 and 7.

Fig. 6a shows position and orientation errors of the four network controllers with $\delta = 10$. As defined in (11),



(a)

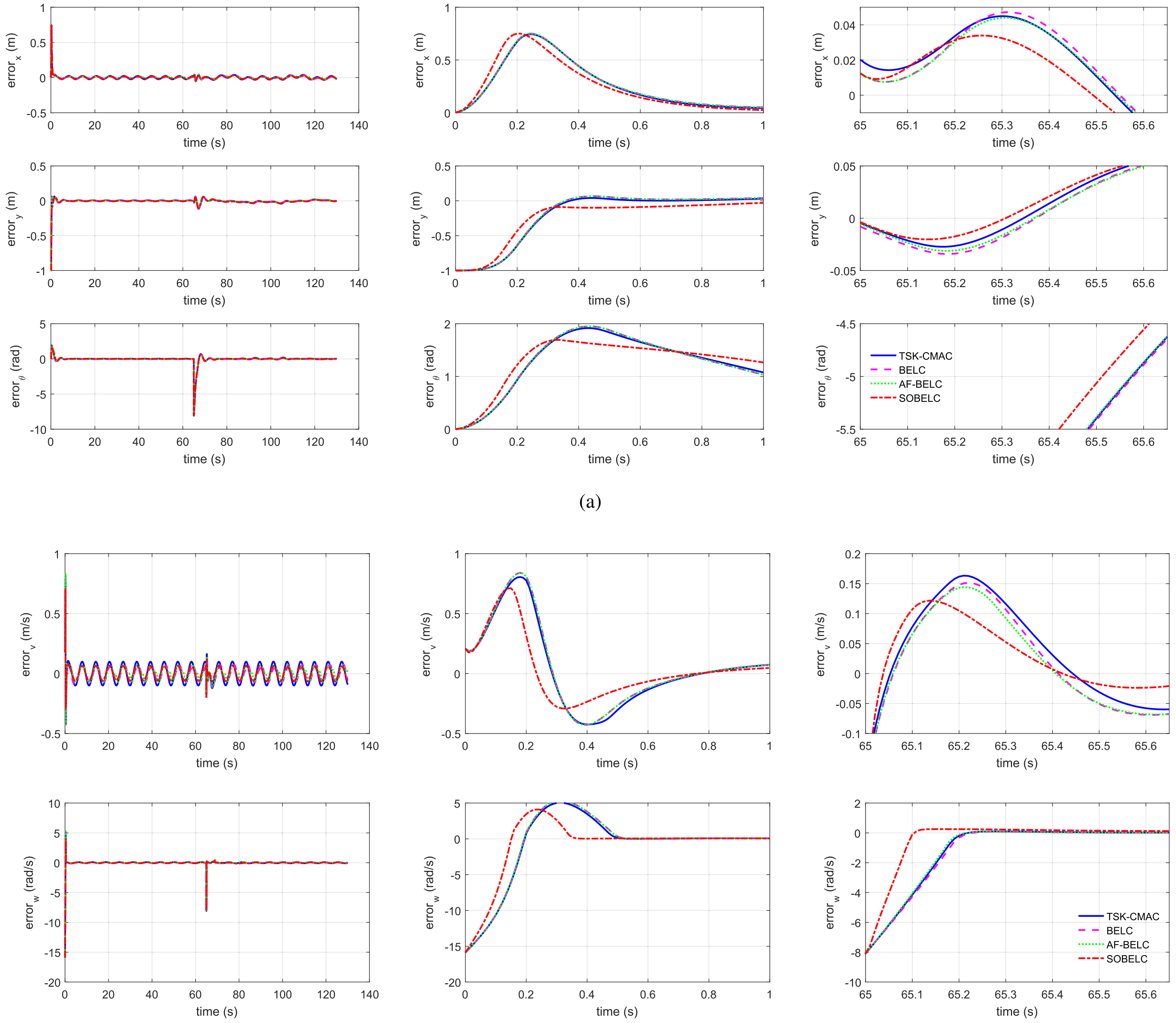
(b)

FIGURE 6. Tracking performance of the four network controllers with τ_d : $\delta = 10$. (a) Position and orientation errors of the four controllers. (b) Velocity errors of the four controllers .

the errors consist of x, y, and θ components. The left column shows the entire tracking procedure and the right column shows the magnification of the errors at around 65s. Three plots in the left column indicate the three errors in x, y, and θ; the other three in the middle column indicate the three magnified versions of those left ones after a few seconds of tracking; and the rest three in the right column show the magnified plots after the 65s.

In this figure, the left three plots reveal the similar situation with that of Fig. 5: the four network-based controllers can rapidly react to tracking errors. However, in the magnified plots at 0s, The TSK-CMAC, BELC, and AF-BELC controllers generate very similar results regarding the position and orientation tracking errors, since the three curves are almost coincided. However, the tracking performance of the SOBELC controller is much better than those of the TSK-CMAC, BELC, and AF-BELC controllers at 0s. The SOBELC exhibits fast error convergence speed in x and y; in addition, the SOBELC can achieve less overshoot in θ. At 65s, the SOBELC controller also exhibits the faster error convergence speed than those of the rest three controllers. The performances of the TSK-CMAC and AF-BELC controllers are very similar; however, the BELC controller has the worst performance.

Fig. 6b shows velocity errors of the four network controllers with $\delta = 10$, and the line types of this figure are identical with those of Fig. 6a. As defined in (12), the velocity error consists of υ and ω components. At 0s, the SOBELC controller exhibits clearly advantages in velocity tracking in both υ and ω, since the SOBELC controller has less overshoot and smoother convergence curve. Especially, the performance of the SOBELC in ω has a significant advantage.



(a)

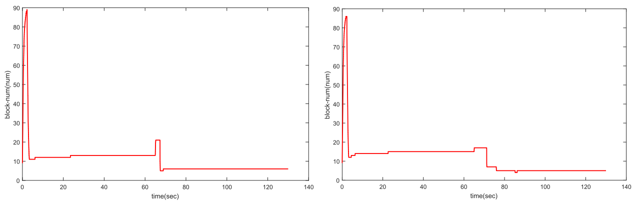
(b)

FIGURE 7. **Tracking performance of the four network controllers with τ_d : $\delta = 20$. (a) Position and orientation errors of the four controllers . (b) Velocity errors of the four controllers.**

At 65*s*, the performances of the four controllers are about the same; the SOBELC controller has a slim leading over the rest three controllers. However, the BELC controller's performance is the worst in the four controllers. As considered the entire tracking performance of the four network controllers with $\delta = 10$, the proposed SOBELC network has a better ability to dynamically control mobile robots.

Fig. 7 illustrates the position, orientation, and velocity tracking performances the four network controllers with $\delta = 20$; the legend of this figure is identical to that of Fig. 6. In Fig. 7a, the entire performances of the four controllers remain consistent with those under $\delta = 10$. The tracking performance of the SOBELC is much better than those of the other three controllers, since the controller demonstrated faster error convergence speed across all these experiments. It is interesting to note that the entire performance of the SOBELC under $\delta = 20$ is better than those under $\delta = 10$. This phenomenon is related to the combination of the outputs of the self-organizing mechanism and the robust term in the SOBELC controller of the proposed system. If a larger disturbance presents, both components must adjust their weights or parameters simultaneously, resulting in different tracking performances from those under small disturbances. Also, the output value range of the robust controller is larger than that of the network; therefore, the robust controller can act faster for larger disturbances. A formal analysis of such situation remains as an important piece of active future work.

Based on the above analysis, Fig. 8 shows the dynamic self-organizing process during the experiments with $\delta = 10$ and 20. Note that, the SOBELC contains two processing channels, Fig. 8 merely shows the hidden neurons of the amygdala channel. The initial size of neurons in the SOBEL



(a)

(b)

FIGURE 8. Self-organizing performance of SOBELC with $\tau_d = 10$ and 20. (a) $\tau_d : \delta = 10$. (b) $\tau_d : \delta = 20$.

is set at 9 in the both levels of disturbance. Under $\delta = 10$, the size increases rapidly just after the mobile robot starts to move; however, when the size reaches 95, it rapidly reduces and remain stable at 11 neurons. At the 65s, the trajectory pattern is changed so that the tracking error have a large shock. Therefore, the size increases to 20, then immediately returns and remains at 8. Under $\delta = 20$, the increasing situation at the beginning is very similar to that under $\delta = 10$: the size has rapid increasing and decreasing. However, after the 65s, the increasing of the size is not remarkable; in particular, the size remains at 11. The more neurons reveal that the SOBELC controller consumes more processing resources to handle the larger disturbances. Also, the dynamic size proves the SOBELC is able to improve the efficiency of resource utilizing for mobile robots.

C. DISCUSSIONS

The quantitative performance comparisons of using the TSK-CMAC, BELC, AF-BELC, and SOBEL for mobile robot control are summarized in Table 1. The accumulated RMSE values of the mobile robot's position, orientation, and velocity against time are used to measure the performance. The time-length of the accumulated RMSE is through the entire tracking process. This table shows the proposed SOBEL controller has the tracking performances under both levels of disturbances. Under $\delta = 20$, the performances of the TSK-CMAC, BELC, and AF-BELC controllers are lower than those under $\delta = 10$; such lower performances indicate that larger disturbances become a challenging task for these structure-fixed network-based controllers. In contrast, the self-organizing mechanism assigns computational resources based on the overall performances of the controllers. The dynamic neuron revising mechanism of the proposed SOBELC automatically adjusts the number of neurons in the hidden layer, in addition to the coupled robust controller, which leads to better tracking performances than the TSK-CMAC, BELC, and AF-BELC controllers, especially when a larger disturbance presents. Note that the SOBELC controller is not allowed to produce too many new neurons to sacrifice the online in time performance, and the size of the hidden layer eventually converges at a reasonably low value. Therefore, the experimental investigations confirm that the proposed SOBELC is more capable in dealing with external disturbances and allocating computational resources.

TABLE 1. Comparison of TSK-CMAC, BELC, AF-BELC, and SOBELC controllers for mobile robot.

τ_d : δ	TSK-CMAC	BELC	AF-BELC	SOBELC
10	0.5646	0.5823	0.5673	**0.4805**
20	0.5812	0.5834	0.5780	**0.4681**

VI. CONCLUSION

This paper focused on the trajectory tracking problem of a nonholonomic mobile robot. A self-organizing neural network was established by integrating the key components of self-organizing RBF and BELC networks. The proposed network was updated by following the brain emotional learning rules and Lyapunov stability theory; and a self-organizing mechanism can automatically add new hidden neurons and prune insignificant neurons. Moreover, the proposed new neural network used the Lyapunov stability theory to guarantee that the updating laws of the network's parameters ensure the convergence of the control system. Experimental results demonstrated the proposed neural network controller can better resist the influences of outside disturbance to improve the trajectory tracking performance and the efficiency of nonlinear function approximation.

There is still room to improve this research. Cutting-edge Type-2 fuzzy inference techniques may be employed in the proposed neural network controller, so as to obtain a better processing ability for uncertainties. More research effort is required to investigate why better control performances are achievable in an environment with larger disturbance. In addition, this initial report of the proposed approach focuses on the simulated mobile robot control only, but practical mobile robots are more appealing to fully discover the potential of the proposed network.

REFERENCES

[1] F. Dayoub, T. Morris, and P. Corke, "Rubbing shoulders with mobile service robots," *IEEE Access*, vol. 3, pp. 333–342, 2015.

[2] Y. Zhang, G. Liu, and B. Luo, "Finite-time cascaded tracking control approach for mobile robots," *Inf. Sci.*, vol. 284, pp. 31–43, Nov. 2014.

[3] C. Son, "Intelligent rule-based sequence planning algorithm with fuzzy optimization for robot manipulation tasks in partially dynamic environments," *Inf. Sci.*, vol. 342, pp. 209–221, May 2016.

[4] G. Tagne, R. Talj, and A. Charara, "Design and comparison of robust nonlinear controllers for the lateral dynamics of intelligent vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 3, pp. 796–809, Mar. 2016.

[5] R. Wu, C. Zhou, F. Chao, Z. Zhu, C.-M. Lin, and L. Yang, "A developmental learning approach of mobile manipulator via playing," *Frontiers Neurorobotics*, vol. 11, p. 53, Oct. 2017. [Online]. Available: https://www.frontiersin.org/article/10.3389/fnbot.2017.00053

[6] Y. Song, Y. Li, and X. Xiao, "Control system design and study for an automatic mobile robot," in *Proc. IEEE Int. Conf. Inf. Automat.*, Aug. 2015, pp. 3118–3123.

[7] J. Liao, Z. Chen, and B. Yao, "Performance-oriented coordinated adaptive robust control for four-wheel independently driven skid steer mobile robot," *IEEE Access*, vol. 5, pp. 19048–19057, 2017.

[8] D. Zhou *et al.*, "Use of human gestures for controlling a mobile robot via adaptive CMAC network and fuzzy logic controller," *Neurocomputing*, vol. 282, pp. 218–231, Mar. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231217318490

[9] L. N. Tan, "Omnidirectional-vision-based distributed optimal tracking control for mobile multirobot systems with kinematic and dynamic disturbance rejection," *IEEE Trans. Ind. Electron.*, vol. 65, no. 7, pp. 5693–5703, Jul. 2017.

[10] S. Blazic, "On periodic control laws for mobile robots," *IEEE Trans. Ind. Electron.*, vol. 61, no. 7, pp. 3660–3670, Jul. 2014.

[11] R. Wang, C. Hu, F. Yan, and M. Chadli, "Composite nonlinear feedback control for path following of four-wheel independently actuated autonomous ground vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 7, pp. 2063–2074, Jul. 2016.

[12] Y. Yan and Y. Li, "Mobile robot autonomous path planning based on fuzzy logic and filter smoothing in dynamic environment," in *Proc. Intell. Control Automat.*, Jun. 2016, pp. 1479–1484.

[13] B. Shen, H. Tan, Z. Wang, and T. Huang, "Quantized/saturated control for sampled-data systems under noisy sampling intervals: A confluent vandermonde matrix approach," *IEEE Trans. Autom. Control*, vol. 62, no. 9, pp. 4753–4759, Sep. 2017.

[14] N. T. Luy, "Robust adaptive dynamic programming based online tracking control algorithm for real wheeled mobile robot with omni-directional vision system," *Trans. Inst. Meas. Control*, vol. 39, no. 6, pp. 832–847, 2016.

[15] N. T. Luy, N. T. Thanh, and H. M. Tri, "Reinforcement learning-based intelligent tracking control for wheeled mobile robot," *Trans. Inst. Meas. Control*, vol. 36, no. 7, pp. 868–877, 2014.

[16] R.-J. Wai and Y.-W. Lin, "Adaptive moving-target tracking control of a vision-based mobile robot via a dynamic petri recurrent fuzzy neural network," *IEEE Trans. Fuzzy Syst.*, vol. 21, no. 4, pp. 688–701, Aug. 2013.

[17] Y. J. Mon and C. M. Lin, "Image processing based obstacle avoidance control for mobile robot by recurrent fuzzy neural network," *J. Intell. Fuzzy Syst.*, vol. 26, no. 6, pp. 2747–2754, 2014.

[18] D. Zhou, F. Chao, Z. Zhu, C.-M. Lin, and C. Zhou, "A novel approach to a mobile robot via multiple human body postures," in *Proc. 12th World Congr. Intell. Control Automat. (WCICA)*, Jun. 2016, pp. 1463–1468.

[19] B. Shen, Z. Wang, and H. Qiao, "Event-triggered state estimation for discrete-time multidelayed neural networks with stochastic parameters and incomplete measurements," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 5, pp. 1152–1163, May 2017.

[20] H. Liu, Z. Wang, B. Shen, and F. E. Alsaadi, "H_∞ state estimation for discrete-time memristive recurrent neural networks with stochastic time-delays," *Int. J. Gen. Syst.*, vol. 45, no. 5, pp. 633–647, 2016, doi: 10.1080/03081079.2015.1106731.

[21] Z. Zhu, F. Chao, X. Zhang, M. Jiang, and C. Zhou, "A developmental approach to mobile robotic reaching," in *Proc. 8th Int. Conf. Intell. Robot. Appl. (ICIRA)*, H. Liu, N. Kubota, X. Zhu, R. Dillmann, and D. Zhou, Eds. Portsmouth, U.K.: Springer, Aug. 2015, pp. 284–294.

[22] E. Lotfi, A. Khosravi, and S. Nahavandi, "Facial emotion recognition using emotional neural network and hybrid of fuzzy c-means and genetic algorithm," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, Jul. 2017, pp. 1–6.

[23] M. Jafari, H. Xu, and L. R. G. Carrillo, "Brain emotional learning-based intelligent controller for flocking of multi-agent systems," in *Proc. Amer. Control Conf.*, May 2017, pp. 1996–2001.

[24] E. Lotfi, O. Khazaei, and F. Khazaei, "Competitive brain emotional learning," *Neural Process. Lett.*, vol. 47, no. 2, pp. 745–764, 2017.

[25] K. Lundagard and C. Balkenius, "A computational model of emotional learning in the amygdala," in *From Animals to Animats 6: Proceedings of the Sixth International Conference on Simulation of Adaptive Behavior*, J.-A. Meyer, A. Berthoz, D. Floreano, H. L. Roitblat, S. W. Wilson, Eds. Cambridge, MA, USA: MIT Press, 2000.

[26] C. Balkenius and J. Morén, "Emotional learning: A computational model of the amygdala," *Cybern. Syst.*, vol. 32, no. 6, pp. 611–636, 2001.

[27] H. Mirhajianmoghadam, M.-R. Akbarzadeh-T, and E. Lotfi, "A harmonic emotional neural network for non-linear system identification," in *Proc. Iranian Conf. Electr. Eng.*, May 2016, pp. 1260–1265.

[28] C.-F. Hsu and T.-T. Lee, "Emotional fuzzy sliding-mode control for unknown nonlinear systems," *Int. J. Fuzzy Syst.*, vol. 19, no. 3, pp. 942–953, Jun. 2017, doi: 10.1007/s40815-016-0216-7.

[29] Q. Zhou, F. Chao, and C.-M. Lin, "A functional-link-based fuzzy brain emotional learning network for breast tumor classification and chaotic system synchronization," *Int. J. Fuzzy Syst.*, May 2017, doi: 10.1007/s40815-017-0326-x.

[30] M. Jafari, R. Fehr, L. R. G. Carrillo, and H. Xu, "Brain emotional learning-based intelligent tracking control for unmanned aircraft systems with uncertain system dynamics and disturbance," in *Proc. Int. Conf. Unmanned Aircraft Syst.*, Jun. 2017, pp. 1470–1475.

[31] C.-M. Lin and C.-C. Chung, "Fuzzy brain emotional learning control system design for nonlinear systems," *Int. J. Fuzzy Syst.*, vol. 17, no. 2, pp. 117–128, 2015.

[32] D. Zhou, F. Chao, C.-M. Lin, L. Yang, M. Shi, and C. Zhou, "Integration of fuzzy CMAC and BELC networks for uncertain nonlinear system control," in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, Jul. 2017, pp. 1–6.

[33] F. Chao, X. Zhang, H.-X. Lin, C.-L. Zhou, and M. Jiang, "Learning robotic hand-eye coordination through a developmental constraint driven approach," *Int. J. Autom. Comput.*, vol. 10, no. 5, pp. 414–424, Oct. 2013, doi: 10.1007/s11633-013-0738-5.

[34] F. Chao *et al.*, "A developmental approach to robotic pointing via human–robot interaction," *Inf. Sci.*, vol. 283, pp. 288–303, Nov. 2014.

[35] F. Chao *et al.*, "Enhanced robotic hand–eye coordination inspired from human-like behavioral patterns," *IEEE Trans. Cogn. Devel. Syst.*, vol. 10, no. 2, pp. 384–396, Jun. 2018.

[36] C.-F. Hsu, C.-M. Lin, and C.-M. Chung, "Design of a growing-and-pruning adaptive RBF neural control system," in *Proc. Int. Conf. Mach. Learn. Cybern.*, vol. 6, Jul. 2009, pp. 3252–3257.

[37] C.-H. Kao, C.-F. Hsu, and H.-S. Don, "Design of an adaptive self-organizing fuzzy neural network controller for uncertain nonlinear chaotic systems," *Neural Comput. Appl.*, vol. 21, no. 6, pp. 1243–1253, Sep. 2012, doi: 10.1007/s00521-011-0537-2.

[38] C. M. Lin and H. Y. Li, "Intelligent hybrid control system design for antilock braking systems using self-organizing function-link fuzzy cerebellar model articulation controller," *IEEE Trans. Fuzzy Syst.*, vol. 21, no. 6, pp. 1044–1055, Dec. 2013.

[39] H.-G. Han, S. Zhang, and J.-F. Qiao, "An adaptive growing and pruning algorithm for designing recurrent neural network," *Neurocomputing*, vol. 242, pp. 51–62, Jun. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231217303296

[40] C.-F. Hsu, C.-J. Chiu, and J.-Z. Tsai, "Indirect adaptive self-organizing RBF neural controller design with a dynamical training approach," *Expert Syst. Appl.*, vol. 39, no. 1, pp. 564–573, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417411010098

[41] C. Lucas, D. Shahmirzadi, and N. Sheikholeslami, "Introducing belbic: Brain emotional learning based intelligent controller," *Intell. Automat. Soft Comput.*, vol. 10, no. 1, pp. 11–21, 2004.

[42] E. Lotfi and M. R. Akbarzadeh-T, *A Winner-Take-all Approach to Emotional Neural Networks With Universal Approximation Property*. Amsterdam, The Netherlands: Elsevier, 2016.

[43] C.-C. Chung and C.-M. Lin, "Fuzzy brain emotional cerebellar model articulation control system design for multi-input multi-output nonlinear," *Acta Polytechnica Hungarica*, vol. 12, no. 4, pp. 39–58, 2015.

[44] R. Fierro and F. L. Lewis, "Control of a nonholonomic mobile robot using neural networks," *IEEE Trans. Neural Netw.*, vol. 9, no. 4, pp. 589–600, Jul. 1998.

[45] M. Jafari, A. M. Shahri, and S. B. Shuraki, "Speed control of a digital servo system using brain emotional learning based intelligent controller," in *Proc. Power Electron., Drive Syst. Technol. Conf.*, Feb. 2013, pp. 311–314.

[46] N. Lotfi, E. Lotfi, R. Mirzaei, A. Khosravi, and S. Nahavandi, "Nonlinear programming problem solving based on winner take all emotional neural network for tensegrity structure design," in *Proc. Int. Joint Conf. Neural Netw.*, Jul. 2016, pp. 826–831.

[47] J. Jin and Y. Wang, "Fuzzy CMAC-based trajectory tracking control for nonholonomic mobile robot," *Comput. Eng. Appl.*, 2015.

[48] C. M. Lin and H. Y. Li, "Adaptive dynamic sliding-mode fuzzy CMAC for voice coil motor using asymmetric Gaussian membership function," *IEEE Trans. Ind. Electron.*, vol. 61, no. 10, pp. 5662–5671, Oct. 2014.

QIUXIA WU received the M.Eng. degree in cognitive science and technology from Xiamen University, China, in 2017. She continues her research in the Cognitive Science Department, Xiamen University. Her research interests include Chinese calligraphic robots and developmental learning algorithms.

CHIH-MIN LIN (M'87–SM'99–F'10) was born in Changhua, Taiwan, in 1959. He received the B.S. and M.S. degrees from the Department of Control Engineering, National Chiao Tung University, Hsinchu, Taiwan, in 1981 and 1983, respectively, and the Ph.D. degree from the Institute of Electronics Engineering, National Chiao Tung University, in 1986. He is a Visiting Professor with the Cognitive Science Department, Xiamen University. He is currently a Chair Professor and a Vice President of Yuan Ze University, Taiwan. He has authored over 170 journal papers. His current research interests include fuzzy neural network, cerebellar model articulation controller, intelligent control systems, and signal processing. He received the Honor Research Fellowship from The University of Auckland, Auckland, New Zealand, from 1997 to 1998. He serves as an Associate Editor for the IEEE Transactions on Cybernetics and the IEEE Transactions on Fuzzy Systems.

WUBING FANG received the B.Eng. degree from the Yinchuan College, China University of Mining and Technology, in 2013. He is currently pursuing the M.Sc. degree with the Cognitive Science Department, Xiamen University. From 2013 to 2016, he was an Engineer with Shanxi Xintianyuan Pharmaceutical Chemical Co., Ltd. His research interests include neural network control, robotics, and deep neural networks.

FEI CHAO (M'11) received the B.Sc. degree in mechanical engineering from Fuzhou University, China, in 2004, the M.Sc. degree (Hons.) in computer science from the University of Wales, Aberystwyth, U.K., in 2005, and the Ph.D. degree in robotics from Aberystwyty University, Aberystwyth, U.K., in 2009. From 2009 to 2010, he was a Research Associate with Aberystwyth University, under the supervision of his Ph.D. supervisor and a Prof. M. H. Lee. He is currently an Associate Professor with the Cognitive Science Department, Xiamen University, China. He is an Adjunct Research Fellow with the Department of Computer Science, Aberystwyth University.

Dr. Chao has authored over 30 peer-reviewed journal and conference papers. His research interests include developmental robotics, machine learning, and optimization algorithms. He is a member of CCF and CAAI. He is a Vice Chair of the IEEE Computer Intelligence Society Xiamen Chapter.

LONGZHI YANG (M'12–SM'18) is currently a Programme Leader and a Senior Lecturer with Northumbria University, Newcastle upon Tyne, U.K. His research interests include computational intelligence, machine learning, big data, computer vision, intelligent control systems, and the application of such techniques in real-world uncertain environments. He received the Best Student Paper Award from the 2010 IEEE International Conference on Fuzzy Systems. He is the Founding Chair of the IEEE Special Interest Group on Big Data for Cyber Security and Privacy.

CHANGJING SHANG received the Ph.D. degree in computing and electrical engineering from Heriot-Watt University, U.K. She was with Heriot-Watt University, Loughborough University, and Glasgow University. She is currently a University Research Fellow with the Department of Computer Science, Institute of Mathematics, Physics and Computer Science, Aberystwyth University, U.K. Her research interests include pattern recognition, data mining and analysis, space robotics, and image modeling and classification.

CHANGLE ZHOU received the Ph.D. degree from Peking University in 1990. He is currently a Professor with the Cognitive Science Department, Xiamen University, where he is also the Director of the Fujian Provincial Key Laboratory of Brain-like Intelligent Systems and the Laboratory of Art, Mind and Computation. He is also an Affiliated Professor of linguistics and applied linguistics with the Humanity College, Zhejiang University, and an Affiliated Professor with the Philosophy Department, Xiamen University. His research interests lie in the areas of artificial intelligence. His scientific contribution to the AI has more to do with machine consciousness and the logic of mental self-reflection, and beyond AI project, he also carries out research on a host of other topics including computational brain modeling, computational modeling of analogy and metaphor and creativity, computational musicology, and information processing of data regarding traditional Chinese medicine. His philosophical works lie in ancient oriental thoughts of Chinese, such as ZEN, TAO, and YI viewed from science.