A Novel Self-Organizing Emotional CMAC Network for Robotic Control*

Juncheng Zhang[1], Quanfeng Li[1], Xiang Chang[2], Fei Chao[1,2], Chih-Min Lin[3], Longzhi Yang[4], Tuân Tú Huỳnh[3], Ling Zheng[1], Changle Zhou[1], and Changjing Shang[2]

***Abstract*—This paper proposes a self-organizing control system for uncertain nonlinear systems. The proposed neural network is composed of a conventional brain emotional learning network (BEL) and a cerebellar model articulation controller network (CMAC). The input value of the network is feed to a BEL channel and a CMAC channel. The output of the network is generated by the comprehensive action of the two channels. The structure of the network is dynamic, using a self-organizing algorithm allows increasing or decreasing weight layers. The parameters of the proposed network are on-line tuned by the brain emotional learning rules; the updating rules of CMAC and the robust controller are derived from the Lyapunov function; in addition, stability analysis theory is used to guaranty the proposed controller's convergence. A simulated mobile robot is applied to prove the effectiveness of the proposed control system. By comparing with the performance of other neural-network-based control systems, the proposed network produces better performance.**

I. Introduction

The control of nonlinear systems is regarded as an important research topic in control engineering. In particular, controllers must well handle both nonlinearly and uncertainty features existing in controlled systems [1], [2]. However, difficulties of mathematical modeling for the nonlinear and uncertain features impede the development of high-performance controllers [3], [4]. Artificial neural networks possess several intelligent features such as self-learning and self-adaptation. Therefore, several artificial neural network-based controllers have been rapidly developed, so as to obtain fast convergence and dynamic response [5].

In particular, dynamic systems must require controllers to have fast dynamic response and rapidly arrange efficient computational resources, so as to reduce the system response latency. Several forward neural network-based controllers used additional resource allocation mechanisms to improve the dynamic response performance [6]. In addition, our previous work [7] designed a neural network containing the key mechanisms of a typical brain emotional learning controller network and a self-organizing radial basis function network. However, such additional mechanisms can only increase or delete one neuron for each time; therefore, a more efficient way of resource allocation is required. In addition, Huynh et al. established a self-organizing network for controlling nonlinear systems [8], [9]. In this work, more weights were added or removed for each turning epoch.

We noticed that the brain emotional learning controller network (BEL) can have better performances on dynamic response due to its dual transmission channels. In Huynh et al.'s work [9], the two channels in BEL share several weights; besides, the network structures of the two channels are identical. As a single channel neural network, CMAC is widely used in the robotic control system [10]. Therefore, it is a promising way to use CMAC's network structure to improve BEL, so as to possess both computational efficiency and nonlinear approximation ability into network-based controllers.

Based on the above considerations, this work develops a new type of self-organizing neural network, called self-organizing emotional cerebellar model articulation controller (SOECMAC), which integrates a CMAC network and a brain emotional learning controller network (BELC). In particular, a self-organizing structure is embedded into the proposed network to determine when to create a new layer or to delete a layer of weights. The rule of increasing and deleting for CMAC layers is introduced. To ensure robust tracking performance, the adaptive laws of CMAC and the robust controller are derived from the Lyapunov function. Based on the new network, a robotic control system comprising a SOECMAC network controller and a robust controller is established.

The remainder of this paper is organized as follows. Section II introduces the architecture and implementation details of the proposed SOECMAC neural network. Section III presents the proposed neural network-based control system and update laws. Section IV provides simulation results and analyzes the effectiveness of the proposed control system. Section V concludes the work.

II. Proposed Network

A. Network Structure of SOECMAC

The proposed network is shown in Fig. 1. The main structure is based on the conventional BEL; thus, the proposed network contains two channels: brain emotional learning network (BEL) channel and cerebellar model articulation controller network (CMAC) channel. The output of the proposed network

This work was supported by the National Natural Science Foundation of China (No. 61673322, 61673326, and 91746103), the Fundamental Research Funds for the Central Universities (No. 20720190142), and the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No. 663830.

[1]J. Zhang, Q. Li, F. Chao, L. Zheng, and C. Zhou are with the School of Informatics, Xiamen University, China fchao@xmu.edu.cn

[2]F. Chao, X. Chang, and C. Shang are with the Institute of Mathematics, Physics and Computer Science, Aberystwyth University, UK

[3]C.-M. Lin and T. T. Huỳnh are with the Department of Electrical Engineering, Yuan Ze University, Taiwan

[4]L. Yang is with the Department of Computer and Information Sciences, Northumbria University, UK

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

is $u_i = b_i - h_i$, where b_i are outputs of the BEL and h_i are outputs of the CMAC. SOECMAC is composed of the input space (I), association memory space for BEL channel (M_1) and CMAC channel (M_2), receptive-field space (J), weight memory space (V, W), and sub-output space (A, O). The signal propagation and basic function of each space are specified as follows.

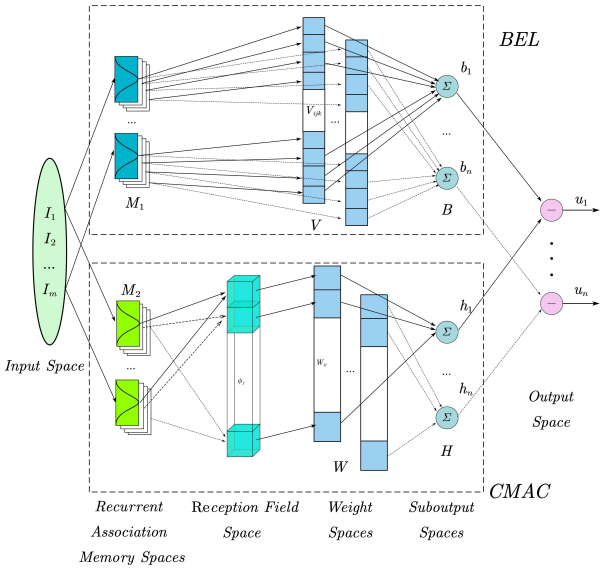

Fig. 1. The architecture of the proposed SOECMAC neural network.

1) ***Input Space****:* $I = [I_1, I_2, ..., I_l]^T \in R^l$ is an input vector that is fed to both BEL and CMAC channels simultaneously; l is the dimension of the input.

2) ***BEL Channel****:* ***Association memory space*** *(M_1)****:*** M_1 consists of a group of blocks, where M_1 consists of n_b blocks in the BEL, every block is represented as a gaussian basis function. The output of the first layer is defined as:

$$\xi_{ij} = exp\left(-\frac{(I_i - c_{ij})^2}{v_{ij}^2}\right) \tag{1}$$

where I_i denotes the i-th input; c_{ij} and v_{ij} are the means and variances of BEL, respectively; $i = 1, 2, ..., l, j = 1, 2, ..., n_b$. The block matrix of BEL Ξ is defined as:

$$\Xi = [\xi_{11} \dots \xi_{1n_b} \dots \xi_{l1} \dots \xi_{ln_b}]^T \in R^{ln_b} \tag{2}$$

Weight Memory Space *(V)****:*** v_{ijk} is the weight of i-th output, j-th input, k-th block of BEL.

Sub-output Space *(A)****:*** a_i is i-th output of BEL,

$$a_i = \sum_{j=1}^{l}\sum_{k=1}^{n_b} v_{ijk}\xi_{ij} \tag{3}$$

3) ***CMAC Channel****:* ***Association memory space*** *(M_2)****:*** M_2 consists of a group of blocks, where M_2 consists of n_k blocks in the CMAC, each block is represented as a gaussian basis function; the output of the first layer is defined as:

$$f_{ij} = exp\left(-\frac{(I_i - m_{ij})^2}{z_{ij}^2}\right) \tag{4}$$

where m_{ij} and z_{ij} are the means and variances of CMAC, respectively; $i = 1, 2, ..., l, j = 1, 2, ..., n_k$.

Reception-field Space *(J)****:*** Each block in the Reception-field Space is the product of the corresponding block of the association memory space M_2, defined as:

$$\phi_j = \prod_{i=1}^{l} f_{ij} \tag{5}$$

The block matrix of CMAC Φ is defined as:

$$\Phi = [\phi_1\ \phi_2 \dots \phi_{n_k}]^T \in R^{n_f} \tag{6}$$

Weight Memory Space *(W)****:*** w_{ij} is the weight of i-th output, j-th block of the Reception-field Space.

Sub-output Space *(O)****:*** o_i is i-th output of CMAC,

$$o_i = \sum_{j=1}^{n_k} w_{ij}\phi_j \tag{7}$$

4) Output Space: a and o are two output vector, which are represented as:

$$a = [a_1, a_2, ..., a_n]^T = V^T\Xi \tag{8}$$

$$o = [o_1, o_2, ..., o_n]^T = W^T\Phi \tag{9}$$

where n denotes the dimension of the output. The total output of the SOECMAC is the result of the interaction of BEL and CMAC, defined as follows:

$$u = a - o = V^T\Xi - W^T\Phi \tag{10}$$

B. Self-Organizing of SOECMAC

The proposed method uses both structural and parametric self-learning. For the SOECMAC, the first task is to determine whether to add a new layer in the association memory of CMAC channel, and form its concurrent hypercube and weight memory. If a current input is inside the boundary of the data sets, the SOECMAC does not create a new layer, but it updates the parameters for the current rules. Note that only CMAC channel is adjustable and BEL channel is fixed.

The distance of mean, $\boldsymbol{DM}_k$, is calculated as:

$$\boldsymbol{DM}_k = \|\boldsymbol{I} - \boldsymbol{m}_k\|_2 \tag{11}$$

where $\boldsymbol{m}_k = [m_{1k}...m_{ik}...m_{lk}]^T$. Define a MAX-MIN method to determine whether to add a new layer:

$$\hat{k} = \arg\min_{1 \leqslant k \leqslant n_k} \boldsymbol{DM}_k \tag{12}$$

If $\max_i \boldsymbol{DM}_{\hat{k}} > K_g$, then a new layer should be generated, where K_g is a predefined generated threshold. This means that for a new input data, if the distance between input data and the mean is too large for the existing clusters, which means that the current value of the existing basis function is too small, then a new cluster should be generated. The number of layers is increased as:

$$n_k\,(t+1) = n_k\,(t) + 1 \tag{13}$$

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

where $n_k(t)$ is the number of existing layers at time t. For the new layer, the weight memory space will be randomly generated, and the initial mean and variance of Gaussian basis function in association memory space will be defined as:

$$m_{in_k} = I_i \tag{14}$$

$$z_{in_k} = z_{i\hat{k}} \tag{15}$$

Another self-organizing structure is consider whether to delete the existing layer. The scale of kth element of the jth output is defined as:

$$\boldsymbol{MM}_{jk} = \frac{v_{jk}}{o_j} \tag{16}$$

Find the maximum scale of n output and the corresponding minimum component:

$$\tilde{k} = \arg \min_{1 \leqslant k \leqslant n_k} \max_{1 \leqslant j \leqslant n} \boldsymbol{MM}_{jk} \tag{17}$$

If $\boldsymbol{MM}_{j\tilde{k}} \leqslant K_c$, then the kth layer will be deleted. where K_c is a predefined deleting threshold. This shows that the smallest part of the current layer is less than the deletion threshold, so it must be deleted.

C. Parameter Update

The weights in the BEL channel are updated based on the brain emotional learning rule, Δv, are defined by:

$$\Delta v = \alpha \left[\Xi \times \max\left(0, d - b\right)\right] \tag{18}$$

$$d = \gamma \times I + \mu \times u_{SOCB} \tag{19}$$

where α is a learning rate, d is composed of the input I and the output u_{SOCB}, γ and μ is the learning rate. The update law of BEL channel's weights is defined as:

$$v\left(t+1\right) = v\left(t\right) + \Delta v \tag{20}$$

The tunable parameters of the CMAC channel are w, m and z. Therefore, to obtain more robust performance, the parameters are update by Lyapunov stability analysis theory, rather than the brain emotional method. The detailed update laws of w, m and z are described in Section III.

$$w\left(t+1\right) = w\left(t\right) + \Delta w \tag{21}$$

$$m\left(t+1\right) = m\left(t\right) + \Delta m \tag{22}$$

$$z\left(t+1\right) = z\left(t\right) + \Delta z \tag{23}$$

III. Neural Network Control System

The proposed SOECMAC control system is shown in Fig. 2. The controller consists of a sliding surface, an SOECMAC network, and a robust controller. The input of the proposed controller is errors of uncertain nonlinear systems. The error values are processed by the sliding surface and delivered to the SOBECMAC network and the robust controller. A nonlinear system with uncertainties can be formulated as:

$$\boldsymbol{x}^{(n)}(t) = f(\boldsymbol{x}(t)) + g(\boldsymbol{x}(t))u(t) + d(t) \tag{24}$$

where $\boldsymbol{x}(t) = \left[x^{(n-1)}(t) \dots \dot{x}(t)\ x(t)\right] \in R^{n \times l}$ denotes the system's state vector, $u(t) = \left[u_1(t), u_2(t) \dots u_n(t)\right]^T \in R^n$ is controller's output vector, $f(\boldsymbol{x}(t))$ and $g(\boldsymbol{x}(t))$ are unknown but bounded system model and $d(t)$ is the unknown external disturbance.

The nominal model of this kind of a nonlinear system is defined as

$$\boldsymbol{x}^{(n)}(t) = f_0(\boldsymbol{x}(t)) + g_0 u(t) \tag{25}$$

where $f_0(\boldsymbol{x}(t))$ and g_0 denotes the nominal part of $f(\boldsymbol{x}(t))$ and $g(\boldsymbol{x}(t))$, respectively. Eq. 25 can be represent as

$$\begin{aligned} x^{(n)}(t) &= f_0(x(t)) + \Delta f(x) + g_0 u(t) + \Delta g(x(t))u(t) + d(t) \\ &= f_0(x(t)) + g_0 u(t) + l(x(t), t) \end{aligned} \tag{26}$$

where $l(x(t), t) = \Delta f(x) + \Delta g(x(t))u(t) + d(t)$ is referred to as the lumped uncertainty.

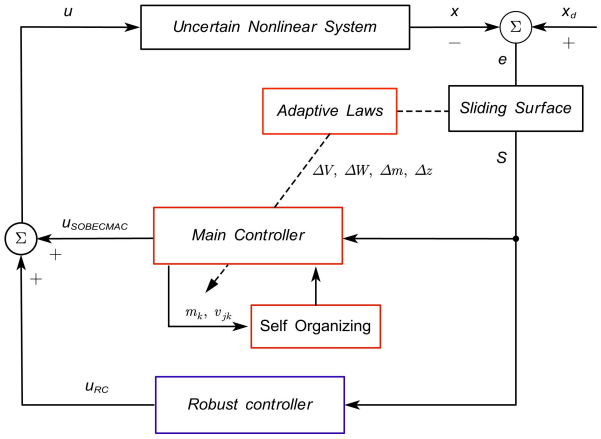

Fig. 2. Design of control system.

Assume the reference tracking state vector, $\boldsymbol{x}_d(t)$ is defined as $\boldsymbol{x}_d(t) = \left[x_d^{(n-1)}(t) \dots \dot{x}_d(t)\ x_d(t)\right] \in R^{n \times l}$, the tracking error of the system is defined as:

$$\boldsymbol{e}(t) = \boldsymbol{x}_d(t) - \boldsymbol{x}(t) = \left[e^{(n-1)}(t) \dots \dot{e}(t)\ e(t)\right] \tag{27}$$

An ideal controller u^* can be designed if nominal functions $f_0(\boldsymbol{x}(t))$, g_0 and lumped uncertainty $l(x(t), t)$ are exactly known.

$$u^*(t) = g_0^{-1}\left[x_d(n)(t) - f_0(x(t)) - l(x(t), t) + \boldsymbol{K}^T e(t)\right] \tag{28}$$

where $\boldsymbol{K} = \left[K_n \dots K_2, K_1\right]^T$ is the feedback gain vector. Substituting the ideal controller Eq.28 into Eq.26 gives the error dynamic equation as follows:

$$e^{(n)}(t) + \boldsymbol{K}^T e(t) = 0 \tag{29}$$

If $\boldsymbol{K}$ is chosen to correspond to the coefficients of a Hurwitz polynomial, it implies that $\lim_{t \to \infty} e(t) = 0$. However, since $l(x(t), t)$ is not known accurately, the ideal controller u^* is unobtainable. Thus, an SOBECMAC controller is developed to approximate the ideal controller.

The sliding surface is defined as:

$$s(t) = e^{(n-1)}(t) + K_1 e^{(n-2)}(t) + \dots + K_n \int e(t)dt \tag{30}$$

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

The derivative of Eq.30 can be represent as

$$\begin{aligned}\dot{s}(t) &= e^{(n)}(t) + \boldsymbol{K}^T e \\ &= x_d\left({}^{(n)}(t) - f_0(x(t)) - g_0 u(t) - l(x(t),t) + \boldsymbol{K}^T e(t)\right.\end{aligned} \tag{31}$$

There is an approximate error $\varepsilon(t) = u^*(t) - u_{SOBECMAC}(t)$ between the proposed controller and the ideal controller. Then, we have:

$$\dot{s}(t) = g_0\left[u^*(t) - u(t)\right] \tag{32}$$

where $u(t) = u_{SOBECMAC}(t) + u_{RC}(t)$, $u_SOBECMAC$ is the controller of SOECMAC, u_{RC} is a robust controller, employed to suppress the influence of residual approximation error between the SOECMAC controller and the ideal controller. Assume that an optimal SOECMAC controller exists in the ideal sliding model controller u^*.

$$u^* = u^*_{REWNN}(t) + \varepsilon = V^{*T}\hat{\Xi} - W^{*T}\Phi^* + \varepsilon \tag{33}$$

where ε is a minimum error vector, V^*, W^*, Ξ^*, Φ^* are optimal parameter matrixes and vectors of V, W, Ξ, Φ. However, the optimal controller is unable to achieve, hence an online estimated controller is used to approach the optimal controller.

$$u = u_{SOBECMAC} + u_{RC} = \hat{V}^T\hat{\Xi} - \hat{W}^T\hat{\Phi} + u_{RC} \tag{34}$$

where $\hat{V}, \hat{W}, \hat{\Xi}, \hat{\Phi}$ are the estimate values of the optimal parameter matrix and vectors of V^*, W^*, Ξ^*, Φ^*, respectively. Subtracting Eq. 33 from 34 gives the estimation error

$$\begin{aligned}\tilde{u} &= u^* - u = V^{*T}\hat{\Xi} - W^{*T}\Phi^* + \varepsilon - \hat{V}^T\hat{\Xi} + \hat{W}^T\hat{\Phi} - u_{RC} \\ &= \tilde{V}^T\hat{\Xi} - \tilde{W}^T\Phi^* - \hat{W}^T\tilde{\Phi} + \varepsilon - u_{RC}\end{aligned} \tag{35}$$

where $\tilde{V} = V^* - \hat{V}, \tilde{\Xi} = \Xi^* - \hat{\Xi}, \tilde{W} = W^* - \tilde{W}, \tilde{\Phi} = \Phi^* - \hat{\Phi}$. The expansion of $\tilde{\Phi}$ in the Taylor series can be obtained as

$$\begin{aligned}\tilde{\Phi} &= \begin{pmatrix}\widetilde{\phi}_1 \\ \vdots \\ \tilde{\phi}_{n_d}\end{pmatrix} = \left.\begin{pmatrix}\left(\frac{\partial\phi_1}{\partial m}\right)^T \\ \vdots \\ \left(\frac{\partial\phi_{n_d}}{\partial m}\right)^T\end{pmatrix}\right|_{m=\hat{m}} (m^* - \hat{m}) \\ &+ \left.\begin{pmatrix}\left(\frac{\partial\phi_1}{\partial z}\right)^T \\ \vdots \\ \left(\frac{\partial\phi_{n_d}}{\partial z}\right)^T\end{pmatrix}\right|_{z=\hat{z}} (z^* - \hat{z}) + O_{t1} \\ &= \Phi_m\tilde{m} + \Phi_z\tilde{z} + O_t\end{aligned} \tag{36}$$

where Φ_m, Φ_z are defined by:

$$\begin{aligned}\Phi_m &= \left.\left[\frac{\partial\phi_1}{\partial m}, \ldots, \frac{\partial\phi_{n_d}}{\partial m}\right]^T\right|_{m=\tilde{m}} \\ \Phi_z &= \left.\left[\frac{\partial\phi_1}{\partial z}, \ldots, \frac{\partial\phi_{n_d}}{\partial z}\right]^T\right|_{z=\tilde{z}}\end{aligned} \tag{37}$$

where $\tilde{m} = m^* - \hat{m}, \tilde{z} = z^* - \hat{z}$, O_t is a vector with higher order terms. Thus we have:

$$\Phi^* = \hat{\Phi} + \tilde{\Phi} = \hat{\Phi} + \Phi_m\tilde{m} + \Phi_z\tilde{z} + O_t \tag{38}$$

Substituting Eq.38 to Eq.35 and Eq.32, yields

$$\begin{aligned}\tilde{u} =& \tilde{V}^T\hat{\Xi} - \tilde{W}^T\left(\hat{\Phi} + \Phi_m\tilde{m} + \Phi_z\tilde{z} + O_t\right) \\ &- \hat{W}^T\left(\Phi_m\tilde{m} + \Phi_z\tilde{z} + O_t\right) + \varepsilon - u_{RC} \\ =& \tilde{V}^T\hat{\Xi} - \tilde{W}^T\hat{\Phi} - \hat{W}^T\left(\Phi_m\tilde{m} + \Phi_z\tilde{z}\right) - u_{RC} + \theta\end{aligned} \tag{39}$$

where $\theta = \tilde{W}^T\Phi_m^T\tilde{m} + \hat{W}^T\Phi_z^T\tilde{z} + W^{*T}O_t + \varepsilon$ is a combined error of CMAC. Where $\tilde{V} = V^* - \hat{V} = [\tilde{v}_1, \tilde{v}_2, \ldots, \tilde{v}_l]$ is an approximation error weight matrix of BEL, A kind of H_∞ tracking performance is designed as:

$$\begin{aligned}\sum_{i=1}^{l}\int_0^T s_i^2(t)dt =& s^T(0)g_o^{-1}s(0) + tr\left[\tilde{W}^T(0)\eta_W^{-1}\tilde{W}(0)\right] \\ &+ \tilde{m}^T(0)\eta_m^{-1}\tilde{m}(0) + \tilde{z}^T(0)\eta_z^{-1}\tilde{z}(0) \\ &+ \sum_{i=1}^{l}\lambda_i^2\int_0^T\theta_i^2(t)dt + \sum_{i=1}^{l}\int_0^T\tilde{v}_i^2(t)dt\end{aligned} \tag{40}$$

where η_m, η_z, η_W are diagonal positive constant learning rate, and λ_i is an attenuation constant. The initial conditions of system are set as $s(0) = 0, \tilde{m} = 0, \tilde{z} = 0, \tilde{W} = 0$; then Eq. 40 can be represent as:

$$\sum_{i=1}^{l}\int_0^T s_i^2(t)dt \le \sum_{i=1}^{l}\lambda_i^2\int_0^T\theta_i^2(t)dt + \sum_{i=1}^{l}\int_0^T\tilde{v}_i^2(t)dt \tag{41}$$

Assume that the approximation error between the proposed SOBECMAC and an ideal controller are bounded, which means $\theta \in L_2[0, M_1], \tilde{v} = L_2[0, M_2], \forall M_1, M_2 \in [0, \infty)$. Therefore $\sum_{i=1}^m\int_0^T\theta_i^2(t)dt \le N_1, \sum_{i=1}^m\int_0^T\tilde{v}_i^2(t)dt \le N_2$, where N_1 and N_2 are positive constants. If $\sum_{i=1}^m\int_0^T s_i^2(t)dt = \infty$. The system will be unstable. There, the following must hold in order to make sure the controlled system is stable:

$$\sum_{i=1}^{m}\int_0^T s_i^2(t)dt \le \|\lambda\|^2 N_1 + N_2 < \infty \tag{42}$$

In order to ensure the stability of the system, the update rules of SOBECMAC and the design of robust controllers must follow the Lyapunov stability theory. The update rules of BEL channel have been given in the previous article. The whole update rules of CMAC channel are designed as:

$$\Delta W = -\eta_W hat\Phi s^T(t) \tag{43}$$

$$\Delta m = -\eta_m\Phi_m^T\hat{W}s^T(t) \tag{44}$$

$$\Delta z = -\eta_z\Phi_z^T\hat{W}s^T(t) \tag{45}$$

The robust controller is designed as:

$$u_{RC} = \left(2R^2\right)^{-1}\left[\left(I + \Xi^2\right)R^2 + I\right]s^T(t) \tag{46}$$

where $R = \mathrm{diag}\left[\lambda_1\lambda_2\ldots\lambda_l\right]$ is a diagonal matrix of robust controller.

Proof. The Lyapunov function is given by:

$$\begin{aligned}L(s(t), \tilde{V}, \tilde{W}, \tilde{m}, \tilde{z}) = \frac{1}{2}[\, & s^T(t)g_0^{-1}s(t) + tr\left[\tilde{W}^T\eta_W^{-1}\tilde{W}\right] + \\ & tr\left[\tilde{V}^T\alpha^{-1}\tilde{V}\right] + \tilde{m}^T\eta_m^{-1}\tilde{m} + \tilde{z}^T\eta_z^{-1}\tilde{z}\,]\end{aligned} \tag{47}$$

Taking the derivative of the Lyapunov function, yields:

$$\begin{aligned}
&\dot{L}(s(t),\tilde{V},\tilde{W},\tilde{m},\tilde{z}) = s^T(t)g_0^{-1}\dot{s}(t) + tr\left[\tilde{V}^T\alpha^{-1}\dot{\tilde{V}}\right] \\
&+ tr\left[\tilde{W}^T\eta_W^{-1}\dot{\tilde{W}}\right] + \tilde{m}^T\eta_m^{-1}\dot{\tilde{m}} + \tilde{z}^T\eta_z^{-1}\dot{\tilde{z}} \\
&= s^T(t)g_0^{-1}\dot{s}(t) - tr\left[\tilde{V}^T\alpha^{-1}\dot{\hat{V}}\right] - tr\left[\tilde{W}^T\eta_W^{-1}\dot{\hat{W}}\right] \\
&- \tilde{m}^T\eta_m^{-1}\dot{\hat{m}} - \tilde{z}^T\eta_z^{-1}\dot{\hat{z}} \\
&= s^T(t)(\tilde{V}^T\hat{\Xi} - \tilde{W}^T\hat{\Phi} - \hat{W}^T(\Phi_m\tilde{m} + \Phi_z\tilde{z}) - u_{RC} + \theta) \\
&- tr\left[\tilde{V}^T\alpha^{-1}\dot{\hat{V}}\right] - tr\left[\tilde{W}^T\eta_W^{-1}\dot{\hat{W}}\right] - \tilde{m}^T\eta_m^{-1}\dot{\hat{m}} - \tilde{z}^T\eta_z^{-1}\dot{\hat{z}} \\
&\leq -tr\left[\tilde{W}\left(s(t)\hat{\Phi} + \eta_W^{-1}\hat{W}\right)\right] - \tilde{m}\left[s^T(t)\hat{W}\Phi_m + \eta_m^{-1}\dot{\hat{m}}\right] \\
&- \tilde{z}\left[s^T(t)\hat{W}\Phi_z + \eta_z^{-1}\dot{\hat{z}}\right] + s^T(t)\tilde{V}\hat{\Xi} + s^T(t)(\theta - u_{RC})
\end{aligned} \tag{48}$$

From Eq. 18 we can see, if $d - b \leq 0$, then $\Delta v = 0$, and if $d - b \geq 0$, $\Delta v = \alpha\left[Xi \times (d-b)\right] > 0$. Given that $\tilde{V} \in L_2[0, M_2]$, it can be derived that $-tr\left[\tilde{V}^T\alpha^{-1}\dot{\hat{V}}\right] \leq 0$. Substitute Eq. 43-46 into 48, yields:

$$\dot{L}(s(t),\tilde{V},\tilde{W},\tilde{m},\tilde{z}) \leq -\frac{1}{2}s^T(t)s(t) + \frac{1}{2}\lambda^2\theta^2 + \frac{1}{2}\widetilde{V}^T\widetilde{V} \tag{49}$$

Intergrating Eq.49 from $t = o$ to $t = T$, yeilds

$$\begin{aligned}
L(T) - L(0) = &-\frac{1}{2}\sum_{i=1}^{l}\int_0^T s_i^2(t)dt + \frac{1}{2}\sum_{i=1}^{l}\lambda_i^2\int_0^T\theta_i^2(t)dt \\
&+ \frac{1}{2}\sum_{i=1}^{l}\int_0^T\tilde{v}_i^2(t)dt
\end{aligned} \tag{50}$$

Since $L(T) > 0, L(0) > 0$, from Eq.41 and 42, it is shown that $\sum_{i=1}^{m}\int_0^T s_i^2(t)dt < \infty$; i.e., the cumulative error is not divergent, it is bounded, thus the stability of the proposed system is proved.

IV. EXPERIMENTATION

In order to illustrate the effectiveness of the proposed control system, a simulated mobile robot system was used in the experimentation. Also, systematically compare the proposed control system with two other comparison controllers, PID controller and SMC with improved fuzzy brain emotional learning model-based controller (iFBEL) [7].

Fig. 3 shows a mobile robot with two coaxially mounted driven wheels and a passive wheel. In the figure, r is the radius of the wheel, $2R$ is the distance between the two wheels, P is the midpoint of the two wheels, C is the center of gravity of the robot. Then the robot position information can be expressed by $q = [x_c \quad y_c \quad \theta]^T$.where x_c and y_c the coordinates of C; θ is the angle between the robot and the reference coordinate system. As for $v\,(t) = [v \quad w]^T$, v and w are the translation and angular velocity of the robot.

The dynamic equation of mobile robot presented as follow:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) + F(\dot{q}) + \tau_d = B(q)\tau - A(q)\psi \tag{51}$$

where $\dot{q}$ is the velocity vector of position and orientation; $\ddot{q}$ is the acceleration vector of position and direction; $M(q)$ is inertia matrix; $C(q,\dot{q})$ is the Coriolis/Centripetal matrix; $g(q)$ is the gravity vector, in this experiment, $g(q) = 0$; $F(\dot{q})$ denotes a friction vector; τ_d denotes external disturbance; $B(q)$ is the input transformation matrix; τ denotes the control input vector; $A(q)$ is a constraint matrix; and ψ is a Lagrange multiplier vector.

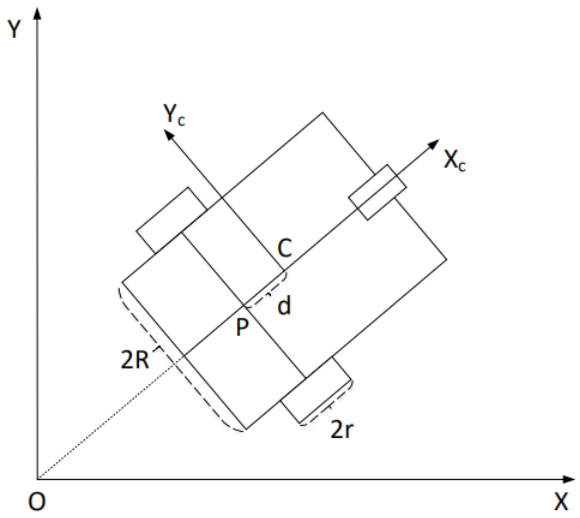

Fig. 3. Wheeled moblie robot model

TABLE I
INITIAL PARAMETERS OF SOBECMAC

	BEL	CMAC
n_b and n_k	9	9
$\alpha\ \gamma\ \mu$ and η_w	0.1 0.1 0.1	0.1
η_m and η_z	0.1	0.1
c and m	(-1.5 1.5)	(-1 1)
v and z	1	1

The reference trajectory of a mobile robot can be described as $q_r = [x_r \quad y_r \quad \theta_r]^T$, the expected velocity can be obtained by using the velocity reference model $v_r(t) = [v_r \quad w_r]^T$. Thus, the velocity error is defined by:

$$e_d = v_r - v = [e_v \quad e_w]^T \tag{52}$$

In the simulation, the parameters of mobile robot are set as: $m = 10kg, I = 5kg \cdot m^2, R = 0.2m, r = 0.05m, d = 0.0m, F(\dot{q}) = 0$. The external disturbance τ_d is defined as:

$$\tau_d = 20\begin{bmatrix}\sin(t)\\ \cos(t)\end{bmatrix} \tag{53}$$

The reference trajectory is designed as:

$$\begin{cases} x_r = v_r \cdot \cos(w) \\ y_r = v_r \cdot \sin(w) \end{cases} \tag{54}$$

The initial state: $q_r = \begin{bmatrix}2 & 0 & \frac{\pi}{2}\end{bmatrix}^T$, $q = \begin{bmatrix}1 & 0 & \frac{\pi}{2}\end{bmatrix}^T$, $v_r = 0.2m/s$, $w_r = 0.1rad/s$. Table I shows the initial parameters of the SOBECMAC network.

Fig. 4 illustrates the simulated position response of the mobile robot. The entire tracking time is 65s. The left figure

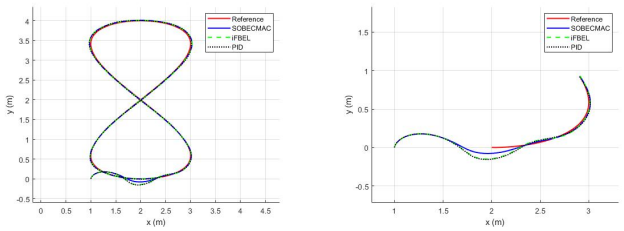

Fig. 4. Simulation results of the PID, iFBEL and SOECMAC controllers

shows the whole tracking process; and the right figure shows the tracking process of the first 10s, in order to observe the differences of each controller in detail. The red solid line denotes the reference trajectory; the green dash line, the black dotted line, and blue solid line present the tracking trajectories of the iFBEL, PID, and SOBECMAC controllers.

In this experiment, the SOBECMAC achieved a favorable tracking performance: After a few adjustments, the SOBECMAC can quickly catch the reference trajectory, and closely follow the reference trajectory. In contrast, the trajectories of PID and iFBEL required a longer time to approximate the target trajectory.

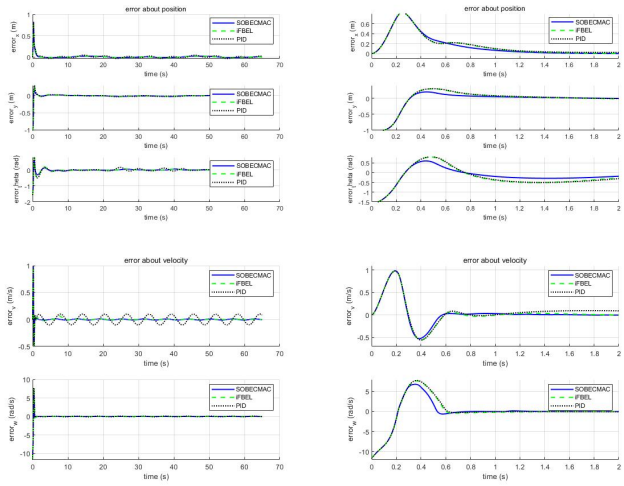

Fig. 5. Simulation results of the proposed control system. (a) Position tracking error; (b) Position tracking error in the first 2 seconds; (c) Velocity tracking error; (d) Velocity tracking error in the first 2 seconds.

Fig. 5 illustrates the error of position and velocity, the right column shows the error of the first 2 seconds. In this figure, all the controllers can have stable performance. However, the SOBECMAC controller had a faster and smoother speed to drive the system to converge.

In order to compare performance differences, Table II shows the quantitative comparisons of the PID, iFBEL, and the proposed SOBECMAC. According to this table, SOBECMAC has the best tracking performance in position, angle, velocity, and angular velocity. Therefore, the proposed SOBECMAC control system has better tracking performance than those of other control systems in this simulation.

TABLE II
QUANTITATIVE COMPARISONS OF THE PID, IFBEL, AND PROPOSED SOECMAC

	PID	iFBEL	SOECMAC
P_e	6.4923e-03	6.1187e-03	**5.8230e-03**
θ_e	1.3981e-02	1.2281e-02	**8.4403e-03**
v_e	7.5887e-03	2.6597e-03	**2.5114e-03**
w_e	4.0888e-01	3.9870e-01	**3.1704e-01**

V. CONCLUSION

This paper proposed a novel self-organizing emotional CMAC neural network-based robotic controller. The newly developed network integrated the components of CMAC and BEL; in particular, a type of organization mechanism was adopted to dynamically adjust the structure of association memory space. The parameters were adjusted by using Lyapunov theorem. The simulation of the uncertain nonlinear system had proved the effectiveness of the proposed control system. This work can be further improved. It is worthwhile to focus on the interpretability of brain emotional learning network; also, the proposed controller will be used in physical robotic hardware experiments.

REFERENCES

[1] Z. Chen, F. Huang, W. Chen, J. Zhang, W. Sun, J. Chen, J. Gu, and S. Zhu, "RBFNN-based adaptive sliding mode control design for delayed nonlinear multilateral tele-robotic system with cooperative manipulation," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2019.

[2] D. Zhou, M. Shi, F. Chao, C.-M. Lin, L. Yang, C. Shang, and C. Zhou, "Use of human gestures for controlling a mobile robot via adaptive cmac network and fuzzy logic controller," *Neurocomputing*, vol. 282, pp. 218 – 231, 2018.

[3] F. Chao, D. Zhou, C. Lin, L. Yang, C. Zhou, and C. Shang, "Type-2 fuzzy hybrid controller network for robotic systems," *IEEE Transactions on Cybernetics*, pp. 1–15, 2019.

[4] W. Fang, F. Chao, C.-M. Lin, L. Yang, C. Shang, and C. Zhou, "An improved fuzzy brain emotional learning model network controller for humanoid robots," *Frontiers in Neurorobotics*, vol. 13, p. 2, 2019. [Online]. Available: https://www.frontiersin.org/article/10.3389/fnbot.2019.00002

[5] T. Huynh, C. Lin, T. Le, H. Cho, T. T. Pham, N. Le, and F. Chao, "A new self-organizing fuzzy cerebellar model articulation controller for uncertain nonlinear systems using overlapped gaussian membership functions," *IEEE Transactions on Industrial Electronics*, pp. 1–1, 2019.

[6] F. Chao, Z. Zhu, C. Lin, H. Hu, L. Yang, C. Shang, and C. Zhou, "Enhanced robotic hand-eye coordination inspired from human-like behavioral patterns," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 10, no. 2, pp. 384–396, June 2018.

[7] Q. Wu, C. Lin, W. Fang, F. Chao, L. Yang, C. Shang, and C. Zhou, "Self-organizing brain emotional learning controller network for intelligent control system of mobile robots," *IEEE Access*, vol. 6, pp. 59 096–59 108, 2018.

[8] T. Huynh, T. Le, and C. Lin, "Self-organizing recurrent wavelet fuzzy neural network-based control system design for MIMO uncertain nonlinear systems using TOPSIS method," *Int. J. Fuzzy Syst.*, vol. 21, no. 2, pp. 468–487, 2019.

[9] C. Lin, R. Ramarao, and S. H. Gopalai, "Self-organizing adaptive fuzzy brain emotional learning control for nonlinear systems," *Int. J. Fuzzy Syst.*, vol. 21, no. 7, pp. 1989–2007, 2019. [Online]. Available: https://doi.org/10.1007/s40815-019-00698-8

[10] W. Fang, F. Chao, L. Yang, C.-M. Lin, C. Shang, C. Zhou, and Q. Shen, "A recurrent emotional cmac neural network controller for vision-based mobile robots," *Neurocomputing*, vol. 334, pp. 227 – 238, 2019.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE