A Recurrent Emotional CMAC Neural Network Controller for Vision-based Mobile Robots

Wubing Fang[1], Fei Chao[1,d,*], Longzhi Yang[b], Chih-Min Lin[c], Changjing Shang[d], Changle Zhou[1], Qiang Shen[d]

[a] *Cognitive Science Department, School of Information Science and Engineering, Xiamen University, China*
[b] *Department of Computer and Information Sciences, Northumbria University, UK*
[c] *Department of Electrical Engineering, Yuan Ze University, Taiwan*
[d] *Institute of Mathematics, Physics and Computer Science, Aberystwyth University, UK*

Abstract

Vision-based mobile robots often suffer from the difficulties of high nonlinear dynamics and precise positioning requirements, which leads to the development demand of more powerful nonlinear approximation in controlling and monitoring of mobile robots. This paper proposes a recurrent emotional cerebellar model articulation controller (RECMAC) neural network in meeting such demand. In particular, the proposed network integrates a recurrent loop and an emotional learning mechanism into a cerebellar model articulation controller (CMAC), which is implemented as the main component of the controller module of a vision-based mobile robot. Briefly, the controller module consists of a sliding surface, the RECMAC, and a compensator controller. The incorporation of the recurrent structure in a slide model neural network controller ensures the retaining of the previous states of the robot to improve its dynamic mapping ability. The convergence of the proposed system is guaranteed by applying the Lyapunov stability analysis theory. The proposed system was validated and evaluated by both simulation and a practical moving-target tracking task. The experimentation demonstrated that the proposed system outperforms other popular neural

*Corresponding author

Email addresses: fangwubing@stu.xmu.edu.cn (Wubing Fang), fchao@xmu.edu.cn (Fei Chao), longzhi.yang@northumbria.ac.uk (Longzhi Yang), cml@saturn.yzu.edu.tw (Chih-Min Lin), cns@aber.ac.uk (Changjing Shang), dozero@xmu.edu.cn (Changle Zhou), qqs@aber.ac.uk (Qiang Shen)

network-based control systems, and thus it is superior in approximating highly nonlinear dynamics in controlling vision-based mobile robots.

Keywords: Mobile robot, recurrent neural network, network based controller

1. Introduction

Along with the rapid development of computer vision technologies, various vision-based mobile robots have been proposed and widely used in many real-world service applications [1, 2, 3, 4, 5, 6]. Note that the design and implementation of mobile robots are challenging due to its non-linearity and non-holonomicity, which has led to a large number of research projects in this area [7, 8, 9, 10, 11]. The traditional control methods work only when the detailed system parameters and accurate position information of the tracking objects are available [12]. This has to be achieved in an environment with highly nonlinear dynamics and uncertain disturbances, where the input chattering of the control systems caused by the disturbances seriously affects the performance and even stability of the control systems [13, 14, 15]. Therefore, it is important to develop a system with high tracking performance to support the vision-based mobile robots, which are currently facing two main challenges as discussed below.

The control systems of mobile robots must be equipped with sufficient nonlinear learning abilities, as the first main challenge, to deal with highly nonlinear dynamics. Feedforward artificial neural networks have been broadly employed for identification and control of mobile robot systems, as neural networks are able to approximate arbitrary nonlinear functions, and thus to reduce the chattering phenomenon of mobile robots [16, 17, 18, 19]. One type of neural networks, Cerebellar Model Articulation Controller (CMAC) network has been widely used in the field of robot motion control, due to its simple structure and rapid learning convergence [20, 21]. For instance, adaptive CMAC networks have been applied to control nonlinear dynamic robot systems, which demonstrated fast response in experiments [21, 22]. Also, Brain Emotional Learning network (BEL) is recognized for its powerful nonlinear approximation characteristic [23, 24, 25, 26]. Such BEL neural network is composed of a sensory neural network representing the orbitofrontal cortex in a human brain, and an emotional neural network referring to the amygdala cortex. Many BEL-based network controllers produce good performances in controlling dynamic systems [27, 28, 29, 30]. The control performance is expected to be greatly improved if the fast responsive ability can be integrated with the excellent nonlinear approximation ability.

The control system is also required to have the ability to handle unexpected uncertainties, which forms the second challenge. If a feedforward neural network is applied, it must include sufficient hidden neurons to represent dynamic responses, which typically leads to bigger computational costs and more serious feedback delay. It has been reported in multiple pieces of work to integrate the recurrent loop to the feedforward neural networks to form a new type of neural network, recurrent neural network (RNN) in addressing this challenge [31, 32, 33]. Since the dynamic response of a system is captured without the use of external feedback through delays, the integrated recurrent loop allows networks to remember the past states of the system and to learn knowledge of the system dynamics implicitly [34, 35]. Based on this, a neural network with a recurrent loop often demonstrates good control performance in the presence of system uncertainties, though there is still room for improvement regarding the nonlinear approximation ability of current RNN models.

This paper proposes a new recurrent neural network which is embedded in a network controller to improve the visual tracking performance of vision-based mobile robots, and thus to address the above challenges. In particular, the proposed recurrent emotional Cerebellar Model Articulation Controller (RECMAC) integrates a CMAC network, an additional emotional network, and a recurrent loop structure, inspired by the FBEL network; and a typical sliding model control structure is adopted to build the network controller. The RECMAC network and the robust controller jointly form the robot control system for moving-target tracking tasks. The RECMAC network, acting as a primary controller, is designed for imitating an ideal controller, while the robust controller, performing as am indirect controller, is served for reducing the approximation errors between the ideal controller and the RECMAC. The Lyapunov stability theory is used in this work to guarantee the stability of the global control system and derive the update laws of the RECMAC. Experiments based on a numerical simulation and a real mobile robot were used for system validation and evaluation. The experimental results demonstrate the feasibility of the proposed recurrent network, which shows the effectiveness of control using multi-neural networks.

The reminder of this paper is organised as follows: Section 2 introduces a basic sliding mode control system for mobile robots. Section 3 describes the proposed RECMAC network in detail. Section 4 presents the RECMAC based network control system, proves the stability of global control system using the Lyapunov stability theory, and derives the update laws of the RECMAC. Section 5 reports the experimental results and discusses how emotional network improves the nonlinear ability of CMAC. Finally, Section 6 concludes the paper and points out future work.

2. Background

A mobile robot is a highly nonlinear system. For a given vision-based target tracking task, any small external disturbances and/or visual input instabilities can seriously affect the tracking performances of mobile robots. The sliding mode control (SMC) has been considered as an effective mobile robot control method once the state of a robot system reaches a sliding surface, that is the SMC can well handle external interference and system uncertainties caused by input instability [36, 37]. Without lose generality, a nth-order mobile robot control system with mth-order input and output states can be expressed as follow:

$$x^{(n)}(t) = f(\underline{x}(t)) + G(\underline{x}(t))u(t) + d(t), \tag{1}$$

where $\underline{x}(t) = \begin{bmatrix} x^{(n-1)}(t) & \dots & \dot{x}(t) & x(t) \end{bmatrix} \in \Re^{m\times n}$ is the system state vector, $u(t) = [u_1(t), u_2(t), \dots, u_m(t)]^T \in \Re^m$ is the control input vector, $f(\underline{x}(t)) \in \Re^m$ is an unknown but bounded nonlinear function, $G(\underline{x}(t)) \in \Re^{m\times m}$ is an unknown but bounded gain matrix, $d(t) = [d_1(t), d_2(t), \dots, d_m(t)]^T \in \Re^m$ is an external disturbance.

The nominal model of (1) is defined as:

$$x^{(n)}(t) = f_0(\underline{x}(t)) + G_0 u(t), \tag{2}$$

where $f_0(\underline{x}(t))$ is nominal function of $f(\underline{x}(t))$, $G_0 = diag[g_1 g_2 \dots g_m] \in \Re^{m\times m}$ is nominal function of $G(\underline{x}(t))$, for $i = 1, 2, \dots, m$, g_i are nominal gain constants, by suitably arranging the control inputs and appropriately choosing the control parameters, G_0 can be positive definite and invertible. Eq. (2) that can then be represented as:

$$\begin{aligned} x^{(n)}(t) =& f_0(\underline{x}(t)) + \triangle f(\underline{x}(t)) + G_0 u(t) + \triangle G(\underline{x}(t))u(t) + d(t) \\ =& f_0(\underline{x}(t)) + G_0 u(t) + \varepsilon(\underline{x}(t), t), \end{aligned} \tag{3}$$

where $\varepsilon(\underline{x}(t), t) = \triangle f(\underline{x}(t)) + \triangle G(\underline{x}(t))u(t) + d(t)$ denotes the external disturbances and lumped uncertainties. $\underline{x}_d(t) = [x_d^{(n-1)T}(t), \dots, \dot{x}_d^T(t), x_d^T(t)]^T \in \Re^{m\times n}$ denotes the trajectory of target which the robot will be tracked. The tracking error vector is thus defined as:

$$\underline{e}(t) = \begin{bmatrix} e^{(n-1)}(t) & e^{(n-2)}(t) & \dots & \dot{e}(t) & e(t) \end{bmatrix}^T \in \Re^{m\cdot n},$$

where $e(t) = x_d(t) - x(t)$ denotes the tracking error.

An ideal sliding surface is defined as:

$$
\begin{aligned}
s(\underline{e}(t)) &= \begin{pmatrix} s_1 \\ s_2 \\ \vdots \\ s_m \end{pmatrix} \\
&= \begin{bmatrix} e_1^{(n-1)}(t)+ & \lambda_{11} e_1^{(n-2)}(t) + \cdots + & \lambda_{n1} \int_0^T e_1(t)dt \\ e_2^{(n-1)}(t)+ & \lambda_{12} e_2^{(n-2)}(t) + \cdots + & \lambda_{n2} \int_0^T e_2(t)dt \\ \vdots & \vdots & \vdots \\ e_m^{(n-1)}(t)+ & \lambda_{1m} e_m^{(n-2)}(t) + \cdots + & \lambda_{nm} \int_0^T e_m(t)dt \end{bmatrix} \\
&= \begin{bmatrix} 1 & & & \lambda_{11} & & & \lambda_{n1} & & \\ & \ddots & & & \ddots & & & \ddots & \\ & & 1 & & & \lambda_{1m} & & & \lambda_{nm} \end{bmatrix} \begin{bmatrix} \underline{e}(t) \\ \int_0^T e(t)dt \end{bmatrix} \\
&= \bar{J} \begin{bmatrix} \underline{e}(t) \\ \int_0^T e(t)dt \end{bmatrix},
\end{aligned} \tag{4}
$$

where $\bar{J} = [I, J] = \begin{bmatrix} I & \lambda_1 I & \dots & \lambda_n I \end{bmatrix} \in \Re^{m \times (m+1)n}$; $\lambda_j = [\lambda_{1j} \dots \lambda_{nj}]^T \in \Re^n$ ($\lambda \in \{1, 2, \cdots, m\}$) are the roots of the equation: $q^n + \lambda_1 q^{n-1} + \cdots + \lambda_{n-1} q + \lambda_n = 0$; and q is a Laplace operator that is in the open left half-plane. Taking the time derivative of (4), the following yields:

$$
\begin{aligned}
\dot{s}(\underline{e}(t)) &= \bar{J} \begin{bmatrix} \dot{\underline{e}}(t) \\ e(t) \end{bmatrix} = \bar{J} \begin{bmatrix} e^{(n)}(t) \\ \underline{e}(t) \end{bmatrix} \\
&= e^{(n)}(t) + J\underline{e}(t) = x_d^{(n)}(t) - x^{(n)}(t) + J\underline{e}(t) \\
&= x_d^{(n)}(t) - f_0(\underline{x}(t)) - G_0 u(t) - \varepsilon(\underline{x}(t), t) + J\underline{e}(t)
\end{aligned} \tag{5}
$$

where $\dot{\underline{e}}(t) = \begin{bmatrix} e^{(n)}(t) & e^{(n-1)}(t) & \dots & \dot{e}(t) \end{bmatrix}^T \in \Re^{m \cdot n}$.

For the existence and reachability of sliding surface, the control law of a robot system should satisfy the following inequation:

$$
\frac{1}{2} \frac{d}{dt}(s_i^2) \leq -\sum_{i=1}^{m} \sigma_i |s_i| \tag{6}
$$

for $\sigma_i > 0$, $i = 1, 2, \dots, m$.

Applying (5) into (6), the following is derived:

$$s^T(\underline{e}(t))\dot{s}(\underline{e}(t)) = s^T(\underline{e}(t))[x_d^{(n)}(t) - f_0(\underline{x}(t)) - G_0 u(t) - \varepsilon(\underline{x}(t), t) + J\underline{e}(t)] \leq -\sum_{i=1}^{m} \sigma_i |s_i|. \tag{7}$$

If the lumped uncertainty $\varepsilon(\underline{x}(t), t)$ and the system dynamic are known exactly, the ideal sliding mode controller (ISMC) is designed as:

$$u_{ISMC} = G_0^{-1}[x_d^{(n)}(t) - f_0(\underline{x}(t)) - \varepsilon(\underline{x}(t), t) + J\underline{e}(t) + \sigma sgn(s(\underline{e}(t)))], \tag{8}$$

where $sgn(\cdot)$ is a sign function.

Unfortunately, it is extremely difficult to practically define the dynamical function and to measure the lumped uncertainty of system. Therefore, the ideal sliding mode controller defined in (8) is generally unobtainable. However, if the ideal sliding mode controller can be represented by a neural network, the dynamical function of the system can be explicitly represented, and the robustness of SMC can be exploited [29]. This in the same time requires higher nonlinear approximation ability of the system dynamics for the highly nonlinear characteristics of vision-based mobile robot.

3. The Proposed RECMAC Network

This paper combines the efforts of multiple neural networks to collectively mimic the ideal sliding surface. In order to accurately simulate the nonlinear mobile robot, an emotional networks is integrated into a CMAC network as an additional component, with the support of a recurrent loop structure, and the combined network is named as recurrent emotional cerebellar model articulation controller (RECMCA). The configuration of proposed RECMCA network is illustrated in Fig. 3. The outputs of the system are $u_i = b_i - h_i, i = 1, 2, \ldots, m$, where b_i are the outputs of the Recurrent Emotional Network (REN) and h_i are the outputs of the Recurrent CMAC (RCMAC). REN includes the input space (I), recurrent association memory space (M_1), weight memory space (K), and sub-output space (B). While RCMAC shares the input space with REN and contains the recurrent association memory space (M_2), receptive-field space

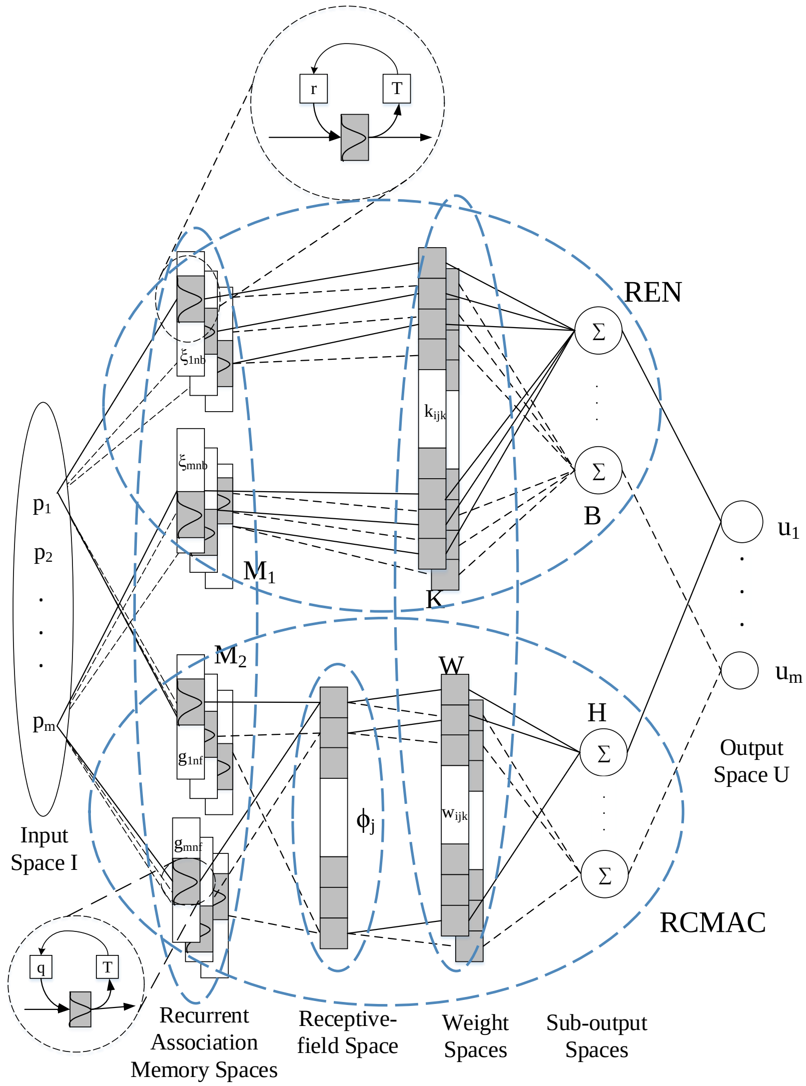

Figure 1: The configuration of the proposed RECMAC network.

(R), weight memory space (W), and sub-output space (H). These spaces are specified as below.

1. *Input Space* I : $p = [p_1, p_2, \ldots, p_m]^T \in \Re^m$ is an input vector that are fed to both REN and RCMAC, simultaneously.

2. *Recurrent Association Memory Spaces* M_1 and M_2 : M_1 and M_2 consist of a group of blocks, the number of blocks, n_b and n_f for REN and RCMAC, respectively. n_b and n_f is larger than or equal to two. Every block is represented as a Gaussian basis function, i.e. ξ is for REN and g is for RCMAC. This ξ is defined as:

$$\xi_{ij} = exp[-\frac{(p_{b_{ij}} - c_{ij})^2}{v_{ij}^2}], \tag{9}$$

   where c_{ij} and v_{ij} are the means and variances of REN, respectively; $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n_b$. $p_{b_{ij}}$ denotes the input of the recurrent structure of REN. The definition of $p_{b_{ij}}$ is given in the Recurrent Structure subsection.

   The block matrix of REN Ξ is defined as:

$$\Xi = \begin{bmatrix} \xi_{11} & \ldots & \xi_{1n_b} & \ldots & \xi_{m1} & \ldots & \xi_{mn_b} \end{bmatrix}^T \in \Re^{mn_b}. \tag{10}$$

   For RCMAC, g_{ij} is defined by:

$$g_{ij} = exp[-\frac{(p_{g_{ij}} - y_{ij})^2}{z_{ij}^2}], \tag{11}$$

   where y_{ij}, and z_{ij} are the means and variances of RCMAC, respectively; $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n_f$. In addition, the definition of $p_{g_{ij}}$ is also given in the Recurrent Structure subsection.

3. *The Recurrent Structure*: The recurrent structure is added to each unit of the recurrent association memory space as illustrated in Fig. 3. Therefore, the input of each unit consists of two parts, one is the current input $p(t)$ at time t; and the other one is the output of the recurrent structure at time $t - \Gamma$ (Γ denotes a time unit). The output of REN is $\xi(t)$, and the output

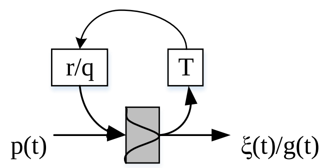

Figure 2: The recurrent structure in RECMAC network.

of RCMAC is $g(t)$. Therefore, the overall input of REN and RCMAC can be expressed as:

$$p_{b_{ij}}(t) = p_i(t) + r_{ij}\xi_{ij}(t - \Gamma), \tag{12}$$

$$p_{g_{ij}}(t) = p_i(t) + q_{ij}g_{ij}(t - \Gamma), \tag{13}$$

where r and q are the recurrent coefficients of the REN and RCMAC, respectively.

The recurrent structure makes the network working in a dynamic way by remembering the past states of the network, which are especially helpful in the tasks of moving target tracking for mobile robots.

4. *Receptive-field Space* R : Each component in the receptive-field space is the product of corresponding components of recurrent association memory space M_2, which is defined as:

$$\phi_j = \prod_{i=1}^{m} g_{ij} = \prod_{i=1}^{m} exp[-\frac{(p_{g_{ij}} - y_{ij})^2}{z_{ij}^2}] = exp[-\sum_{i=1}^{m} \frac{(p_{g_{ij}} - y_{ij})^2}{z_{ij}^2}], \tag{14}$$

where $j = 1, 2, \ldots, n_f$. The block matrix of RCMAC Φ is defined as:

$$\Phi = \begin{bmatrix} \phi_{11} & \ldots & \phi_{1n_f} & \ldots & \phi_{m1} & \ldots & \phi_{mn_f} \end{bmatrix}^T \in \Re^{mn_f}. \tag{15}$$

5. *Weight Memory Spaces* K and W : κ_{ijk} is the weight of the ith output, the jth input, and the kth block of REN; and ω_{ijk} is the weight of ith

output, jth layer, and the kth block of RCMAC. Thus, K is defined by:

$$K = \begin{bmatrix} \kappa_{1jk} & \kappa_{2jk} & \dots & \kappa_{mjk} \end{bmatrix}$$
$$= \begin{bmatrix} \kappa_{111} & \kappa_{211} & \cdots & \kappa_{m11} \\ \vdots & \vdots & & \vdots \\ \kappa_{11n_b} & \kappa_{21n_b} & \cdots & \kappa_{m1n_b} \\ \kappa_{121} & \kappa_{221} & \cdots & \kappa_{m21} \\ \vdots & \vdots & & \vdots \\ \kappa_{12n_b} & \kappa_{22n_b} & \cdots & \kappa_{m2n_b} \\ \vdots & \vdots & & \vdots \\ \kappa_{1m1} & \kappa_{2m1} & \cdots & \kappa_{mm1} \\ \vdots & \vdots & & \vdots \\ \kappa_{1mn_b} & \kappa_{2mn_b} & \cdots & \kappa_{mmn_b} \end{bmatrix} \in \Re^{mn_b \times m} \tag{16}$$

W is defined by:

$$W = \begin{bmatrix} \omega_{1jk} & \omega_{2jk} & \dots & \omega_{mjk} \end{bmatrix}$$
$$= \begin{bmatrix} \omega_{111} & \omega_{211} & \cdots & \omega_{m11} \\ \vdots & \vdots & & \vdots \\ \omega_{11n_f} & \omega_{21n_f} & \cdots & \omega_{m1n_f} \\ \omega_{121} & \omega_{221} & \cdots & \omega_{m21} \\ \vdots & \vdots & & \vdots \\ \omega_{12n_f} & \omega_{22n_f} & \cdots & \omega_{m2n_f} \\ \vdots & \vdots & & \vdots \\ \omega_{1m1} & \omega_{2m1} & \cdots & \omega_{mm1} \\ \vdots & \vdots & & \vdots \\ \omega_{1mn_f} & \omega_{2mn_f} & \cdots & \omega_{mmn_f} \end{bmatrix} \in \Re^{mn_f \times m}. \tag{17}$$

6. *Suboutput Space* B and H : b_i and h_i are the ith outputs of both REN

and RCMAC, which are represented as:

$$b_i = \sum_{j=1}^{m} \sum_{k=1}^{n_b} \kappa_{ijk} \xi_{jk}, \tag{18}$$

$$h_i = \sum_{j=1}^{m} \sum_{k=1}^{n_f} \omega_{ijk} \phi_{jk}. \tag{19}$$

In the above equations, b and h denote the output vectors, which are represented as:

$$b = \begin{bmatrix} b_1 & b_2 & \dots & b_m \end{bmatrix}^T = K^T \cdot \Xi, \tag{20}$$

$$h = \begin{bmatrix} h_1 & h_2 & \dots & h_m \end{bmatrix}^T = W^T \cdot \Phi. \tag{21}$$

7. *Output Space* U : The output of RECMAC, u_i, is a equation of the outputs of both REN and RCMAC, which is defined as:

$$u_i = b_i - h_i = \sum_{j=1}^{m} \sum_{k=1}^{n_b} \kappa_{ijk} \xi_{jk} - \sum_{j=1}^{m} \sum_{k=1}^{n_f} \omega_{ijk} \phi_{jk}. \tag{22}$$

Let u denote the final output of the entire network, which is expressed as:

$$u = b - h = K^T \cdot \Xi - W^T \cdot \Phi. \tag{23}$$

The overall computing procedure of the proposed RECMAC network is summarized in pseudo-code, as shown in Algorithm 1.

4. The Control System of Vision-based Robots

The RECMAC proposed in the last section mimics the sliding surface, which is used as a primary controller in the overall control system; this works with a robust controller, as a supplementary indirect controller, jointly performing control tasks for vision-based robots. The framework of the proposed vision-based mobile robot control system is illustrated in Fig. 3.

The stability of the global control system can be proven using the Lyapunov stability theory; from this, a set of update laws for the RECMAC network are

Algorithm 1 The pseudocode of RECMAC network

1: Normalize each dimension x_i of X;

2: Compute ξ_{ij} and g_{ij} by using (11) and (9);

3: Update ξ_{ij} and g_{ij} by computing $p_{b_{ij}}$ and $p_{g_{ij}}$;

4: Compute ϕ_j by using (14);

5: Compute Ξ and Φ by using (10) and (15);

6: Compute b_i and h_i by using (18), then compute suboutputs b and h by using (20);

7: Compute the output u of network by using (23);

8: update $\dot{\hat{K}}$, $\dot{\hat{W}}$, $\dot{\hat{y}}$, $\dot{\hat{z}}$, $\dot{\hat{q}}$, $\dot{\hat{c}}$, $\dot{\hat{v}}$, and $\dot{\hat{r}}$ by using updating rules (38) and (40).

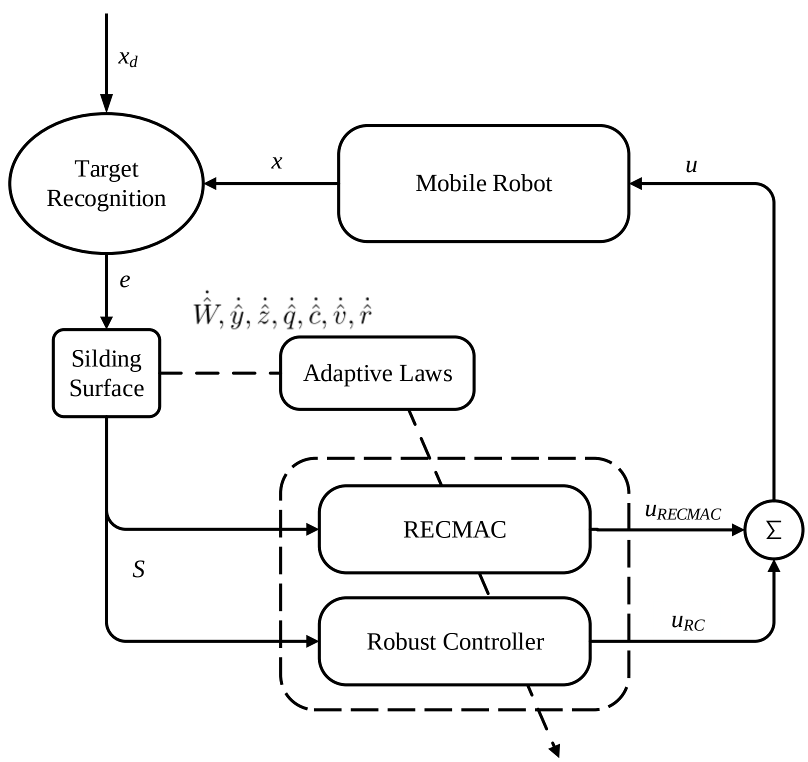

Figure 3: The proposed RECMAC-based Control System.

derived to support the proposed control system. Following the discussions in Section 3. The following yields by subtracting (8) from (5):

$$\dot{s}(\underline{e}(t)) = G_0[u_{ISMC} - u] - \sigma sgn[s(\underline{e}(t))]. \tag{24}$$

Assume that there exists an optimal RECMAC, u^*_{RECMAC}, to imitate an ideal sliding mode controller u_{ISMC}, that ϵ is a minimum error vector, that K^* and W^* are optimal weight matrixes, and that Ξ^* and Φ^* are optimal weight matrixes of the optimal RECMAC, respectively. Then the output of the optimal ISMC is:

$$\begin{aligned} u_{ISMC} &= u^*_{RECMAC} + \epsilon = (u_{REN} - u_{RCMAC})^* + \epsilon \\ &= (K^T\Xi - W^T\Phi)^* + \epsilon = K^{*T}\Xi^* - W^{*T}\Phi^* + \epsilon. \end{aligned} \tag{25}$$

The final output of RECMAC is u and actual outputs of REN and RCMAC are u_{REN} and u_{RCMAC}, respectively. u_{RC} is the output of the robust controller. $\hat{K}, \hat{W}, \hat{\Phi}$ and $\hat{\Xi}$ are estimated matrixes of K^*, W^*, Φ^*, Ξ^*, respectively. The actual output of the entire controller is then defined by:

$$u = u_{RECMAC} + u_{RC} = \hat{K}^T\hat{\Xi} - \hat{W}^T\hat{\Phi} + u_{RC}. \tag{26}$$

Taking (25) and (26) into (24), the following can be derived:

$$\begin{aligned} \dot{s}(\underline{e}(t)) =& G_0[K^{*T}\Xi^* - W^{*T}\Phi^* + \epsilon - \hat{K}^T\hat{\Xi} + \hat{W}^T\hat{\Phi} - u_{RC}] - \sigma sgn[s(\underline{e}(t))] \\ =& G_0[\tilde{K}^T\Xi^* + \hat{K}^T\tilde{\Xi} - \tilde{W}^T\Phi^* - \hat{W}^T\tilde{\Phi} + \epsilon - u_{RC}] - \sigma sgn[s(\underline{e}(t))], \end{aligned} \tag{27}$$

where $\tilde{\Phi} = \Phi^* - \hat{\Phi}, \tilde{K} = K^* - \hat{K}, \tilde{\Xi} = \Xi^* - \hat{\Xi}$, and $\tilde{W} = W^* - \hat{W}$. A partially linear form of the receptive-field basis function vector $\tilde{\Xi}$ in Taylor series can be described as:

$$\tilde{\Xi} = \begin{pmatrix} \tilde{\xi_1} \\ \vdots \\ \tilde{\xi_{n_d}} \end{pmatrix} = \begin{pmatrix} (\frac{\partial \xi_1}{\partial c})^T \\ \vdots \\ (\frac{\partial \xi_{n_d}}{\partial c})^T \end{pmatrix} |_{c=\hat{c}}(c^* - \hat{c}) + \beta_1$$
$$+ \begin{pmatrix} (\frac{\partial \xi_1}{\partial v})^T \\ \vdots \\ (\frac{\partial \xi_{n_d}}{\partial v})^T \end{pmatrix} |_{v=\hat{v}}(v^* - \hat{v}) + \begin{pmatrix} (\frac{\partial \xi_1}{\partial r})^T \\ \vdots \\ (\frac{\partial \xi_{n_d}}{\partial r})^T \end{pmatrix} |_{r=\hat{r}}(r^* - \hat{r}) \tag{28}$$
$$= \Xi_c \tilde{c} + \Xi_v \tilde{v} + \Xi_r \tilde{r} + \beta_1,$$

where Ξ_c, Ξ_v and Ξ_r are defined by:

$$\begin{cases} \Xi_c = [\dfrac{\partial \xi_1}{\partial c}, \ldots, \dfrac{\partial \xi_{n_d}}{\partial c}]^T|_{c=\hat{c}} \in \Re^{n_d \times n_b n_d} \\ \Xi_v = [\dfrac{\partial \xi_1}{\partial v}, \ldots, \dfrac{\partial \xi_{n_d}}{\partial v}]^T|_{v=\hat{v}} \in \Re^{n_d \times n_b n_d} \\ \Xi_r = [\dfrac{\partial \xi_1}{\partial r}, \ldots, \dfrac{\partial \xi_{n_d}}{\partial r}]^T|_{r=\hat{r}} \in \Re^{n_d \times n_b n_d}, \end{cases} \tag{29}$$

where $\tilde{c} = c^* - \hat{c}, \tilde{v} = v^* - \hat{v}, \tilde{r} = r^* - \hat{r}$, and β_1 is a higher-order vector. Rewriting (28) with $\tilde{\Xi} = \Xi^* - \hat{\Xi}$ leads to:

$$\Xi^* = \hat{\Xi} + \tilde{\Xi} = \hat{\Xi} + \Xi_c \tilde{c} + \Xi_v \tilde{v} + \Xi_r \tilde{r} + \beta_1. \tag{30}$$

Also, a partially linear form of the receptive-field basis function vector $\tilde{\Phi}$ in Taylor series is described as:

$$\tilde{\Phi} = \begin{pmatrix} \tilde{\phi_1} \\ \vdots \\ \tilde{\phi_{n_d}} \end{pmatrix} = \begin{pmatrix} (\frac{\partial \phi_1}{\partial y})^T \\ \vdots \\ (\frac{\partial \phi_{n_d}}{\partial y})^T \end{pmatrix} |_{y=\hat{y}}(y^* - \hat{y}) + \beta_2$$
$$+ \begin{pmatrix} (\frac{\partial \phi_1}{\partial z})^T \\ \vdots \\ (\frac{\partial \phi_{n_d}}{\partial z})^T \end{pmatrix} |_{z=\hat{z}}(z^* - \hat{z}) + \begin{pmatrix} (\frac{\partial \phi_1}{\partial q})^T \\ \vdots \\ (\frac{\partial \phi_{n_d}}{\partial q})^T \end{pmatrix} |_{q=\hat{q}}(q^* - \hat{q}) \tag{31}$$
$$= \Phi_y \tilde{y} + \Phi_z \tilde{z} + \Phi_q \tilde{q} + \beta_2,$$

where Φ_y, Φ_z and Φ_q are defined by:

$$\begin{cases} \Phi_y = [\frac{\partial \phi_1}{\partial y}, \dots, \frac{\partial \phi_{n_d}}{\partial y}]^T|_{y=\hat{y}} \in \Re^{n_d \times n_f n_d} \\ \Phi_z = [\frac{\partial \phi_1}{\partial z}, \dots, \frac{\partial \phi_{n_d}}{\partial z}]^T|_{z=\hat{z}} \in \Re^{n_d \times n_f n_d} \\ \Phi_q = [\frac{\partial \phi_1}{\partial q}, \dots, \frac{\partial \phi_{n_d}}{\partial q}]^T|_{q=\hat{q}} \in \Re^{n_d \times n_f n_d}, \end{cases} \tag{32}$$

where $\tilde{y} = y^* - \hat{y}, \tilde{z} = z^* - \hat{z}, \tilde{q} = q^* - \hat{q}$, β_2 are higher-order vectors. Rewriting (31) with $\tilde{\Phi} = \Phi^* - \hat{\Phi}$, yields

$$\Phi^* = \hat{\Phi} + \tilde{\Phi} = \hat{\Phi} + \Phi_y \tilde{y} + \Phi_z \tilde{z} + \Phi_q \tilde{q} + \beta_2 \tag{33}$$

Substituting (30) and (33) to (27), Eq. 27 can be re-expressed as:

$$\begin{aligned} \dot{s}(\underline{e}(t)) =& G_0[\tilde{K}^T(\hat{\Xi} + \Xi_c \tilde{c} + \Xi_v \tilde{v} + \Xi_r \tilde{r} + \beta_1) + \hat{K}^T(\Xi_c \tilde{c} + \Xi_v \tilde{v} + \Xi_r \tilde{r} + \beta_1) \\ & - \tilde{W}^T(\hat{\Phi} + \Phi_y \tilde{y} + \Phi_z \tilde{z} + \Phi_q \tilde{q} + \beta_2) - \hat{W}^T(\Phi_y \tilde{y} + \Phi_z \tilde{z} + \Phi_q \tilde{q} + \beta_2) \\ & + \epsilon - u_{RC}] - \sigma sgn[s(\underline{e}(t))] \\ =& G_0[\hat{K}^T(\Xi_c \tilde{c} + \Xi_v \tilde{v} + \Xi_r \tilde{r}) - \hat{W}^T(\Phi_y \tilde{y} + \Phi_z \tilde{z} + \Phi_q \tilde{q}) \\ & + \tilde{K}^T \hat{\Xi} - \tilde{W}^T \hat{\Phi} + \tau - u_{RC}] - \sigma sgn[s(\underline{e}(t))], \end{aligned} \tag{34}$$

where $\tau = K^{*T}\beta_1 + W^{*T}\beta_2 + \tilde{K}_T(\Xi_c \tilde{c} + \Xi_v \tilde{v} + \Xi_r \tilde{r}) + \tilde{W}_T(\Phi_y \tilde{y} + \Phi_z \tilde{z} + \Phi_q \tilde{q}) + \epsilon$ is a combined error of RCMAC while $\tilde{K} = K^* - \hat{K} = [\tilde{k_1}, \tilde{k_2}, \dots, \tilde{k_m}]^T \in \Re^{m \times m n_b}$ is an approximation error weight matrix of REN. A kind of H_∞ tracking performance [15] is considered for the existence of τ and $\tilde{K}$ as:

$$\begin{aligned} & \sum_{i=1}^{m} \int_0^T s_i^2(t)dt \le s^T(0)G_0^{-1}s(0) + tr[\tilde{W}^T(0)\eta_W^{-1}\tilde{W}(0)] + \tilde{c}^T(0)\eta_c^{-1}\tilde{c}(0) \\ & + \tilde{v}^T(0)\eta_v^{-1}\tilde{v}(0) + \tilde{r}^T(0)\eta_r^{-1}\tilde{r}(0) + \tilde{y}^T(0)\eta_y^{-1}\tilde{y}(0) + \tilde{z}^T(0)\eta_z^{-1}\tilde{z}(0) \\ & + \tilde{q}^T(0)\eta_q^{-1}\tilde{q}(0) + \sum_{i=1}^{m} \lambda_i^2 \int_0^T \tau_i^2(t)dt + \sum_{i=1}^{m} \int_0^T \tilde{k}_i^2(t)dt, \end{aligned} \tag{35}$$

where $\eta_W, \eta_c, \eta_v, \eta_r, \eta_y, \eta_z$ and η_q are diagonal positive constant learning-rate matrices, and λ_i is an attenuation constant. The initial conditions of the system are set as $s(0) = 0, \tilde{W}(0) = 0, \tilde{c}(0) = 0, \tilde{v}(0) = 0, \tilde{r}(0) = 0, \tilde{y}(0) = 0, \tilde{z}(0) =$

$0, \tilde{q}(0) = 0$, then Eq. (35) can be rewritten as:

$$\sum_{i=1}^{m} \int_0^T s_i^2(t)dt \leq \sum_{i=1}^{m} \lambda_i^2 \int_0^T \tau_i^2(t)dt + \sum_{i=1}^{m} \int_0^T \tilde{k}_i^2(t)dt. \tag{36}$$

Assume that the approximation error between the proposed RECMAC and an ideal controller are bounded, which means $\tau \in L_2[0, T_1]$ and $\tilde{k} \in L_2[0, T_2]$ with $\forall T_1, T_2 \in [0, \infty)$. Therefore $\int_0^T \tau_i^2(t)dt \leq N_1$ and $\int_0^T \tilde{k}_i^2(t)dt \leq N_2$, where N_1 and N_2 are big positive constants. If $\sum_{i=1}^{m} \int_0^T s_i^2(t)dt = \infty$, the approximation error is diverging and the controlled system will be unstable. Therefore, the following must hold in order to make sure the controlled system is stable:

$$\sum_{i=1}^{m} \int_0^T s_i^2(t)dt \leq ||\lambda_i||^2 N_1 + N_2 < \infty. \tag{37}$$

Then, in order to guarantee the system's stability, the update laws of both RECMAC and the robust controller must be designed by following the Lyapunov stability theory.

Theorem: For the nonlinear vision-based mobile robot as represented by (1), the update laws of the parameters of proposed RECMAC are described from (40) to (46), in which the update rules of REN is designed as in (38) and (39) [24]. Note that as an external network added to CMAC, the emotional network has its own update rules of weights, the update rules of emotional network is analyzed in Section 5.3 in details. The adaptive laws of robust controller are derived as (47):

$$\dot{\hat{K}} = \alpha[\Xi \times max(0, d - b)], \tag{38}$$

$$d = \gamma \times p + \mu \times u_{RECMAC}, \tag{39}$$

where α is a learning-rate constant, d is composed of the input vector p and the output vector u_{RECMAC} with the learning constants γ and μ. The update laws of the parameters of the proposed RECMAC are described as:

$$\dot{\hat{W}} = -\eta_W \hat{\Phi} s^T(\underline{e}(t)) \tag{40}$$

$$\dot{\hat{y}} = -\eta_y \Phi_y^T \hat{W} s^T(\underline{e}(t)) \tag{41}$$

$$\dot{\hat{z}} = -\eta_z \Phi_z^T \hat{W} s^T(\underline{e}(t)) \tag{42}$$

$$\dot{\hat{q}} = -\eta_q \Phi_q^T \hat{W} s^T(\underline{e}(t)) \tag{43}$$

$$\dot{\hat{c}} = \eta_c \Xi_c^T \hat{K} s^T(\underline{e}(t)) \tag{44}$$

$$\dot{\hat{v}} = \eta_v \Xi_v^T \hat{K} s^T(\underline{e}(t)) \tag{45}$$

$$\dot{\hat{r}} = \eta_r \Xi_r^T \hat{K} s^T(\underline{e}(t)) \tag{46}$$

$$u_{RC} = (2R^2)^{-1}[(I + \Xi^2)R^2 + I]s^T(\underline{e}(t)), \tag{47}$$

where $R = diag\begin{bmatrix}\lambda_1 & \lambda_2 & \dots & \lambda_m\end{bmatrix} \in \Re^{m\times m}$ is a diagonal matrix of robust controller.

Proof: The Lyapunov function is given by:

$$L(s(\underline{e}(t)), \tilde{K}, \tilde{W}, \tilde{c}, \tilde{v}, \tilde{r}, \tilde{y}, \tilde{z}, \tilde{q}) = \frac{1}{2}[s^T(\underline{e}(t))G_0^{-1}s(\underline{e}(t)) + tr[\tilde{K}^T\alpha^{-1}\tilde{K}]]$$
$$+ \tilde{c}^T\eta_c^{-1}\tilde{c} + \tilde{v}^T\eta_v^{-1}\tilde{v} + \tilde{r}^T\eta_r^{-1}\tilde{r} + \tilde{y}^T\eta_y^{-1}\tilde{y} + \tilde{z}^T\eta_z^{-1}\tilde{z} + \tilde{q}^T\eta_q^{-1}\tilde{q} + tr[\tilde{W}^T\eta_W^{-1}\tilde{W}]. \tag{48}$$

Taking the derivative of the Lyapunov function and using (27), the following yields:

$$
\begin{aligned}
&\dot{L}(s(\underline{e}(t)),\tilde{K},\tilde{W},\tilde{c},\tilde{v},\tilde{r},\tilde{y},\tilde{z},\tilde{q})\\
=&s^T(\underline{e}(t))G_0^{-1}\dot{s}(\underline{e}(t))+tr[\tilde{K}^T\alpha^{-1}\dot{\tilde{K}}]+\tilde{c}^T\eta_c^{-1}\dot{\tilde{c}}+\tilde{v}^T\eta_v^{-1}\dot{\tilde{v}}\\
&+\tilde{r}^T\eta_r^{-1}\dot{\tilde{r}}+\tilde{y}^T\eta_y^{-1}\dot{\tilde{y}}+\tilde{z}^T\eta_z^{-1}\dot{\tilde{z}}+\tilde{q}^T\eta_q^{-1}\dot{\tilde{q}}+tr[\tilde{W}^T\eta_W^{-1}\dot{\tilde{W}}]\\
=&s^T(\underline{e}(t))G_0^{-1}\dot{s}(\underline{e}(t))-tr[\tilde{K}^T\alpha^{-1}\dot{\hat{K}}]-\tilde{c}^T\eta_c^{-1}\dot{\hat{c}}-\tilde{v}^T\eta_v^{-1}\dot{\hat{v}}\\
&-\tilde{r}^T\eta_r^{-1}\dot{\hat{r}}-\tilde{y}^T\eta_y^{-1}\dot{\hat{y}}-\tilde{z}^T\eta_z^{-1}\dot{\hat{z}}-\tilde{q}^T\eta_q^{-1}\dot{\hat{q}}-tr[\tilde{W}^T\eta_W^{-1}\dot{\hat{W}}]\\
=&s^T(\underline{e}(t))\tilde{K}\hat{\Xi}-s^T(\underline{e}(t))\tilde{W}\hat{\Phi}+s^T(\underline{e}(t))\hat{K}(\Xi_c\tilde{c}+\Xi_v\tilde{v}+\Xi_r\tilde{r})\\
&-s^T(\underline{e}(t))\hat{W}(\Phi_y\tilde{y}+\Phi_z\tilde{z}+\Phi_q\tilde{q})-tr[\tilde{K}^T\alpha^{-1}\dot{\hat{K}}]\\
&-\tilde{c}^T\eta_c^{-1}\dot{\hat{c}}-\tilde{v}^T\eta_v^{-1}\dot{\hat{v}}-\tilde{r}^T\eta_r^{-1}\dot{\hat{r}}-\tilde{y}^T\eta_y^{-1}\dot{\hat{y}}-\tilde{z}^T\eta_z^{-1}\dot{\hat{z}}-\tilde{q}^T\eta_q^{-1}\dot{\hat{q}}\\
&-tr[\tilde{W}^T\eta_W^{-1}\dot{\hat{W}}]+s^T(\underline{e}(t))(\tau-u_{RC})-s^T(\underline{e}(t))G_0^{-1}\sigma sgn[s(\underline{e}(t))]\\
\leq&-tr[\tilde{W}^T(s(\underline{e}(t))\hat{\Phi}+\eta_W^{-1}\dot{\hat{W}})]+\tilde{c}[s^T(\underline{e}(t))\hat{K}\Xi_c-\eta_c^{-1}\dot{\hat{c}}]\\
&+\tilde{v}[s^T(\underline{e}(t))\hat{K}\Xi_v-\eta_v^{-1}\dot{\hat{v}}]+\tilde{r}[s^T(\underline{e}(t))\hat{K}\Xi_r-\eta_r^{-1}\dot{\hat{r}}]\\
&-\tilde{y}[s^T(\underline{e}(t))\hat{W}\Phi_y+\eta_y^{-1}\dot{\hat{y}}]-\tilde{z}[s^T(\underline{e}(t))\hat{W}\Phi_z+\eta_z^{-1}\dot{\hat{z}}]\\
&-\tilde{q}[s^T(\underline{e}(t))\hat{W}\Phi_q+\eta_q^{-1}\dot{\hat{q}}]+s^T(\underline{e}(t))\tilde{K}\hat{\Xi}+s^T(\underline{e}(t))(\tau-u_{RC}).
\end{aligned}
\tag{49}
$$

If $d_i-b\leq 0$, then $\dot{\hat{K}}=0$; and if $d-b>0$, then $\dot{\hat{K}}=\alpha\cdot\Xi\cdot[d-b]>0$. Given that $\tilde{K}\in L_2[0,T_2]$, it can be derived that $-tr[\tilde{K}^T\alpha^{-1}\dot{\hat{K}}]\leq 0$. Substitute (40)-(47) into (49), the following yields:

$$
\begin{aligned}
&\dot{L}(s(\underline{e}(t)),\tilde{W},\tilde{K},\tilde{c},\tilde{v})\leq s^T(\underline{e}(t))\tilde{K}\hat{\Xi}+s^T(\underline{e}(t))(\tau-u_{RC})\\
=&s^T(\underline{e}(t))\tilde{K}\hat{\Xi}+s^T(\underline{e}(t))\tau-\frac{1}{2}s^T(\underline{e}(t))s(\underline{e}(t))-\frac{1}{2}\frac{s^T(\underline{e}(t))s(\underline{e}(t))}{\lambda^2}-\frac{1}{2}s^T(\underline{e}(t))s(\underline{e}(t))\hat{\Xi}\hat{\Xi}^T\\
=&-\frac{1}{2}s^T(\underline{e}(t))s(\underline{e}(t))-\frac{1}{2}[\frac{s(\underline{e}(t))}{\lambda}-\lambda\tau]^2-\frac{1}{2}[s(\underline{e}(t))^T\hat{\Xi}-\tilde{K}]^2+\frac{1}{2}\lambda^2\tau^2+\frac{1}{2}\tilde{K}^T\tilde{K}\\
\leq&-\frac{1}{2}s^T(\underline{e}(t))s(\underline{e}(t))+\frac{1}{2}\lambda^2\tau^2+\frac{1}{2}\tilde{K}^T\tilde{K}.
\end{aligned}
\tag{50}
$$

Integrating (50) from $t=0$ to $t=T$, the following can be derived:

$$
L(T)-L(0)\leq -\frac{1}{2}\sum_{i=1}^{m}\int_0^T s_i^2(t)dt+\frac{1}{2}\sum_{i=1}^{m}\lambda_i^2\int_0^T\tau_i^2(t)dt+\frac{1}{2}\sum_{i=1}^{m}\int_0^T\tilde{k}_i(t)dt.
\tag{51}
$$

Since $L(T) > 0$ and $L(0) > 0$, from (36) and (37), it can be derived that $\sum_{i=1}^{m} \int_0^T s_i^2(t)dt < \infty$; thus the stability of the proposed system is proved.

5. Experimentation

The proposed RECMAC-based controller was applied to a mobile robot for the task of moving target tracking for system validation and evaluation. This experiment was firstly simulated, which systematically compares the performance of the ECMAC controller without the use of the recurrent loop structure, the CMAC controller without the presence of the emotional network, and the proposed RECMAC controller. Then, the experiment was practically carried out using a vision-based mobile robot in a real-world environment. These two experiments are detailed in the following two subsections.

5.1. Numerical Simulation

The process of moving-target tracking in this experiment was simulated in Matlab. The simulated mobile robot was required to track a virtual mobile object, which moves along a predicted reference trajectory; thus the object detection function was omitted in this simulation. A fixed distance, $d = 0.1m$, must be maintained between the mobile robot and the virtual object. Note that the reference trajectory point had a fixed velocity. In order to capture the reference trajectory point, an ideal velocity state of the mobile robot was obtained by applying the Blazic's work [38]. Therefore, in this experiment, the tracking problem was converted to a control problem of the mobile robot in achieving the ideal velocity states.

The reference trajectory included two paths. In the first stage from $t = 0$ to $t = 65$, the trajectory was a circle, and this moved to the second path at the time point $t = 65s$. The changing of the path was designed to evaluate the robustness and response speed of the simulated robot controller. The reference

	REN	RCMAC
Number of blocks n_b and n_f	8	8
Initialization range of mean c_{ij} and y_{ij}	$[-2.0, 2.0]$	$[-1.8, 1.8]$
Initial variances v_{ij} and z_{ij}	0.01	0.5
Initial weights K_{ijk} and W_{ijk}	$[-0.5, 0.5]$	$[-0.5, 0.5]$
Learning rates of weights α, γ, μ and η_W	$0.01, 0.05, 0.01$	0.001
Learning rates of mean η_c and η_y	0.01	0.01
Learning rates of variances η_v and η_z	0.001	0.001
Learning rates of recurrent η_r and η_q	0.001	0.001

Table 1: Parameters of REN and RCMAC.

trajectory used in this experiment are:

$$\begin{cases} x_r = \upsilon_r \cdot \cos(\omega) \\ y_r = \upsilon_r \cdot \sin(\omega), \qquad\qquad t = 0 - 65s, \end{cases} \tag{52}$$

$$\begin{cases} x_r = \upsilon_r \cdot \cos(2\omega) \\ y_r = \upsilon_r \cdot \sin(\omega), \qquad\qquad t = 65 - 150s. \end{cases} \tag{53}$$

The initial velocities of the reference trajectory point and the mobile robot were $\upsilon_r = 0.2m/s$ and $\omega_r = 0.1rad/s$, and their initial positions were $q_r = [2 \quad 0]^T$ and $q = [1 \quad 0]^T$. The initial orientation of the mobile robot was $\omega = \pi/2$. The errors between the ideal velocity and the actual velocity of the mobile robot were fed into the RECMAC network, and the outputs of the network were the velocity of the left and right wheels of mobile robot. The parameters of the applied REN and RCMAC are summarized in Table 1.

The tracking performances of the mobile robot are shown in Fig. 4. The black solid line represents the reference trajectory; and the red dotted line, the blue dotted line, and the green dotted line indicate the tracking trajectories of the mobile robot controlled by the RECMAC, the ECMAC and the CMAC controllers, respectively. The tracking trajectory of the RECMAC controller was

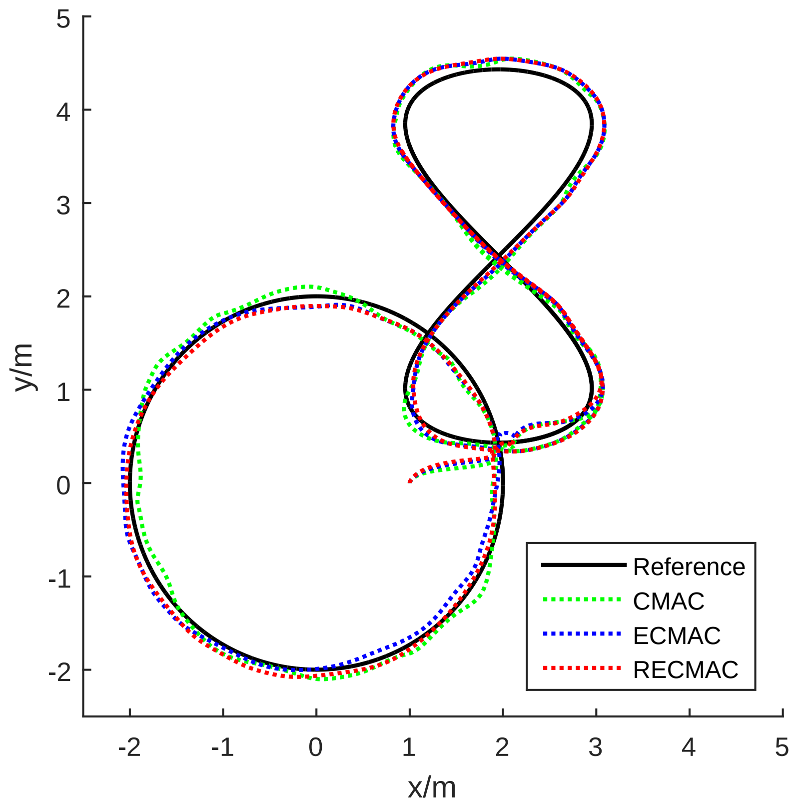

Figure 4: The tracking trajectory of the mobile robot. The black solid line represents the trajectory of the target, while the red dotted line, the blue dotted line and the green dotted line indicate the tracking trajectories of the mobile robot controlled by the RECMAC, ECMAC and CMAC controller, respectively.

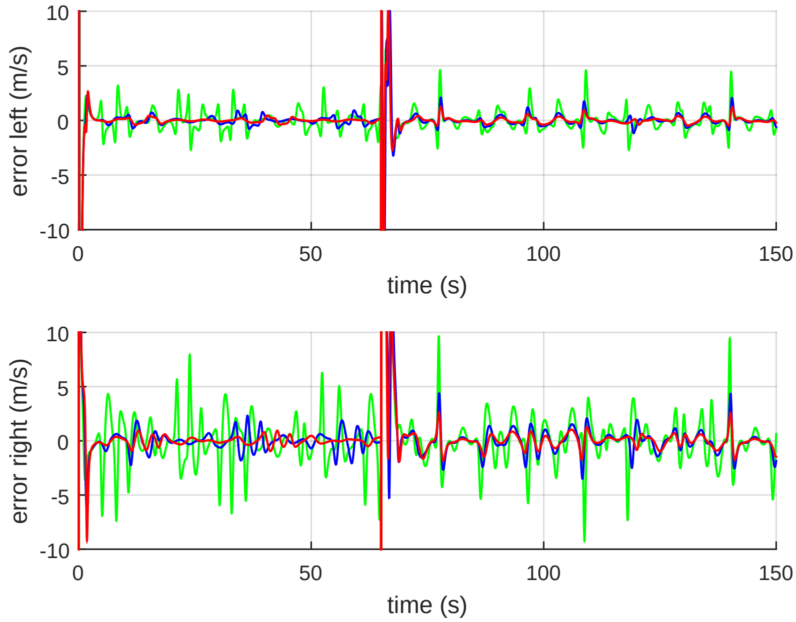

Figure 5: The velocity errors of left and right wheels of mobile robot. The red line, the blue line and the green line are the velocity errors of RECMAC, ECMAC and CMAC controller, respectively.

smoother than that of the EMAC controller, which did not have a recurrent loop structure. The better performance was led by the inclusion of the recurrent loop units, which retained previous states of the system; the previous states can assist the network to handle dynamic situations. The tracking performance of the CMAC controller was the worst within the three. Interestingly, in the second path, the tracking trajectories of the RECMAC and the ECMAC controllers coincide exactly. This expected results was led by the learning ability of the networks in moving target tracking.

The velocity errors of left and right wheels of the mobile robot were shown in Fig. 5. The performance of the proposed RECMAC controller was superior to that of ECMAC controller in velocity control, since RECMAC has a smoother error curve. The error curve of CMAC controller was extremely steep. This simulation indicates that the CMAC is not able to handle the uncertainty as

	CMAC	ECMAC	RECMAC
RMSE of velocity(left)	0.0288	0.0129	**0.0051**
RMSE of velocity(right)	0.0484	0.0321	**0.0237**

Table 2: The accumulated RMSE values of each controller.

efficient as the RECMAC is. Also, there existed a time period of adjustment when trajectory changes after $t = 65s$, as indicated in the figure. The RMSE velocity values of the left and right wheels are shown in Table 2. It is clear from the table, that the control performance and response speed of RECMAC were better than those of the ECMAC and the CMAC.

5.2. Experiment in Real-world Environment

Experiments on a practical mobile robot were provided to evaluate the applicability of the proposed RECMAC controller in a real-world environment. The experimental set up of the robot tracking task is shown in Fig. 6. The task involved two robots: one as the target which moved along a reference trajectory, and aother as the tracer which tracks the target robot. The tracking trajectory was a circle ($radius = 2m$) following a straight line ($s = 10m$); A default distance constraint, $d = 2m$, between the two robots was applied to the task robot.

The task mobile robot was eqipped with a RGB camera, which has two free wheels and two differential driving wheels under a STM32 microprocessor equipped with 265k FlashROM and 48k RAM. The target robot was the Pioneer mobile robot with a blue block on it as the tracking target. The task robot used a simple but effective color-based detection approach to determine and locate the target. The raw RGB images were captured by the camera, which were fed to the color-based detection programme. The color-based programme firstly mapped the images in the HSV color space; then, the coordinates of the target were obtained using the histogram equalization and binarization. From this, the errors were forwarded into the RECMAC network which were the difference

Figure 6: Experimental environment of the practical mobile robot, where a task mobile robot was tracing a target robot moving along a reference trajectory with a random changing velocity.

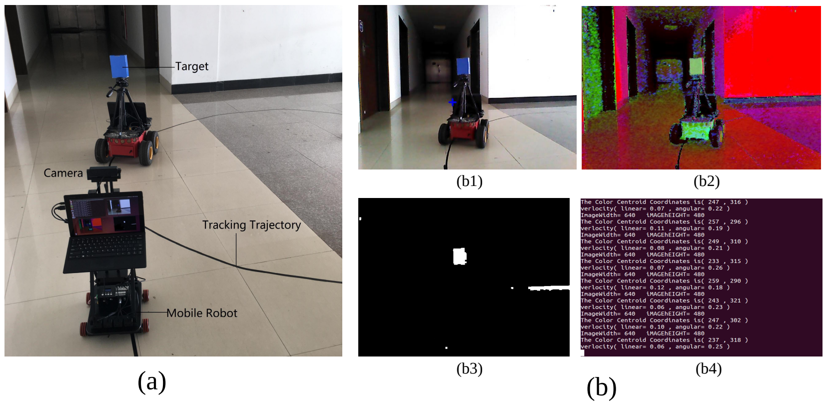

Figure 7: Image processing procedure. (a) The bird-view of the experiment scene. (b) The color based detection programme included in the mobile robot and shown on the control PC. (b1) An example image captured by the camera mounted on the task camera. (b2) The example image in the HSV color space. (b3) The converted binary image with white area presenting the target coordinate. (b4) The color centroid coordinates of the target in the camera frame.

between the coordinates of the target and the center of the camera frame. The outputs of the RECMAC were the velocity values of left and right wheels of task robot.

Fig. 7 illustrates the image processing procedure. The bird view of the experiment scene is shown in Fig. 7-(a), whilst a screenshot of the image processing program running on the control PC of the task robot is illustrated in Fig. 7-(b). The image processing programme detected the target and calculated tracking errors. Figs. 7-(b1) and 7-(b2) show the raw image amputated by the camera and the converted image in the HSV color space, respectively. Fig. 7-(b3) is the binary image, where the white area present the target coordinate. Fig. 7-(b4) shows the coordinates of the target within the camera frame.

The ECMAC controller and the proposed RECMAC controller were applied in controlling the vision-based mobile robot in this experiment to demonstrate the role of the recurrent loop structure in moving-target tracking. Figs. 8 and 9 show the tracking errors of the target robot with a static or a random-changing velocity, respectively. The tracking errors were represented as two

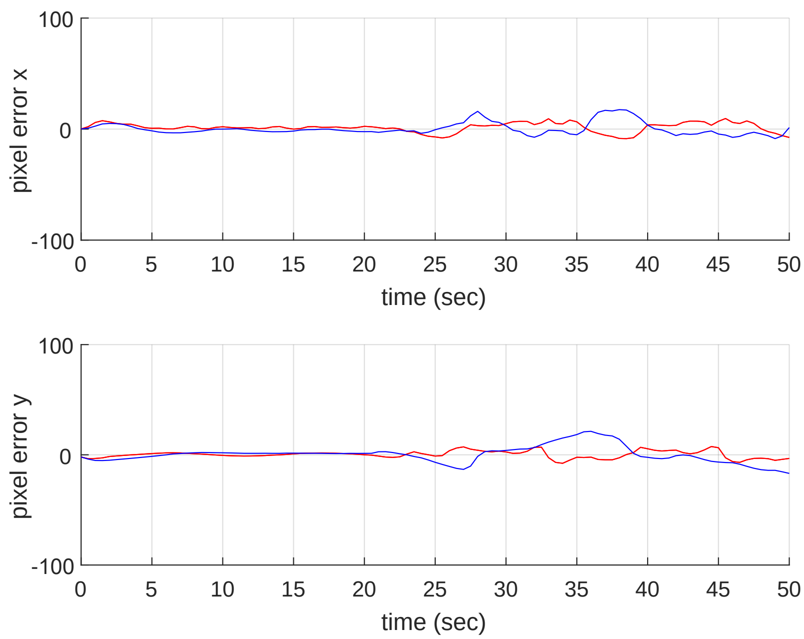

Figure 8: The position errors of X-axis and Y-axis of mobile robot in the case of target robot with a fixed velocity. The red line and the blue line are the errors of RECMAC and ECMAC controller, respectively.

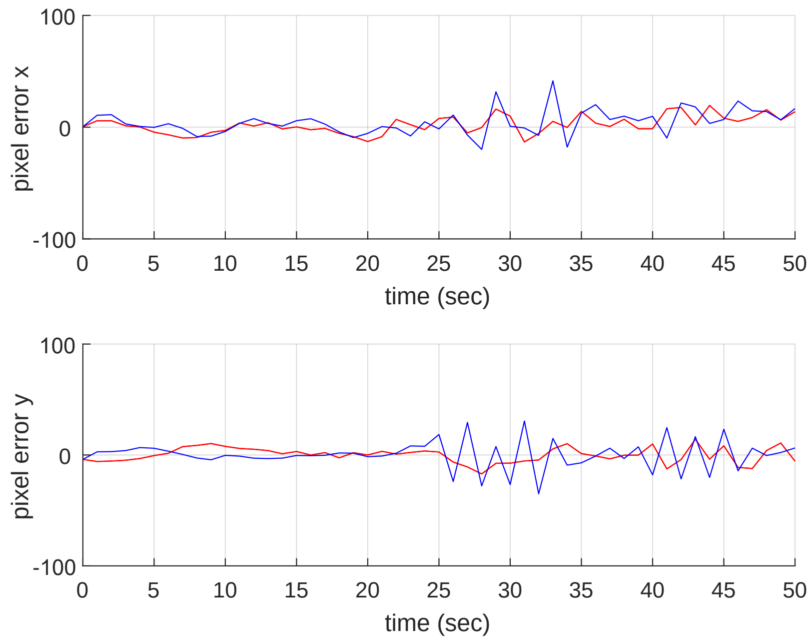

Figure 9: The position errors of X-axis and Y-axis of mobile robot in the case of target robot with a random changing velocity. The red line and the blue line are the errors of RECMAC and ECMAC controller, respectively.

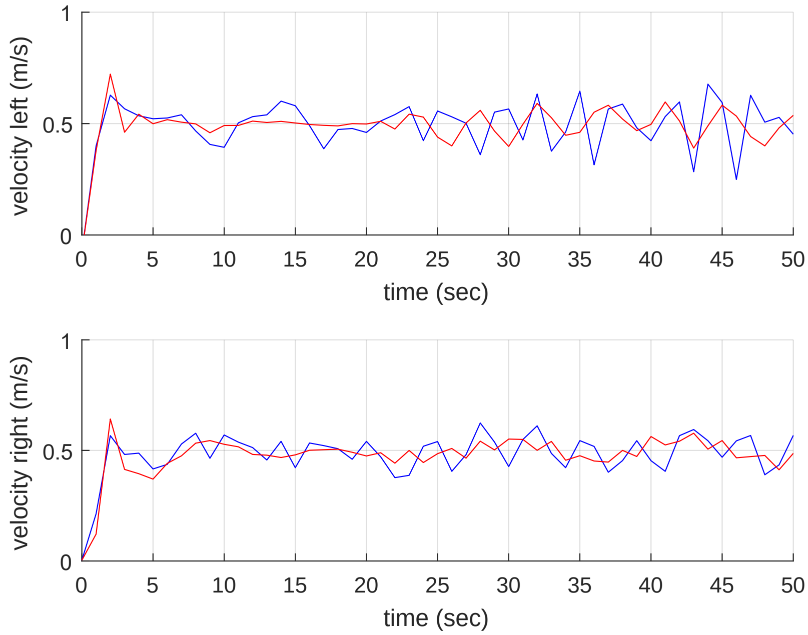

Figure 10: The velocity of left and right wheels of mobile robot. The red line and the blue line are the velocity of RECMAC and ECMAC controller, respectively.

values indicating the coordinate errors of the mobile robot. The velocity values of the left and right wheels of the mobile robot in chasing the random-changing velocity target is shown in Fig. 10. In these three figures, the red and the blue lines denote the RECMAC controller and the ECMAC controller, respectively.

In both Figs. 8 and 9, from $t = 0s$ to about $25s$, the performances in X-axis and Y-axis of the two controllers are close; this is because the target robot moves along a straight line, the tracking difficulty was low. In contrast, from about $t = 25s$ to $t = 50s$, the robot moved along a circular trajectory; due to the recurrent loop structure of RECMAC, the error change range of RECMAC is much smaller than that of ECMAC. Therefore, the RECMAC network exhibited advantages in controlling a dynamic mobile robot. In particular, Fig. 9 demonstrates that advantages became more significant when the target robot had the random-changing velocity.

The velocity changing curves of the right and left wheels of the robot is shown in Fig. 10. The implication of Fig. 10 was similar to that in Figs. 8 and 9: the velocity curves of the RECMAC controller is much smoother than those of the RCMAC. Note that, from time $t = 25s$ to $t = 50s$, the RECMAC network rarely led to any drastic velocity changes, but this is not the case for the compared counterparts. This proves that the presence of the recurrent loop in the network generally improves the performance of the controller in dynamic environments.

5.3. Discussions

Based the experimental results, the proposed RECMAC controller shown better nonlinear approximation ability and faster response speed than those of the ECMAC and CMAC, whilst the ECMAC controller generally outperformed the CMAC controller. This is consistent with the biological model which includes a biological-plausible mechanism. As the motor control center, the cerebellum in human brain controls all of the low level movements of a human body, whilst the human emotions usually play an important role in retaining or enhancing human motions. The CMAC component of the RECMAC network performs similar function which simulates the function of human cerebellum. The amygdala component works as an emotional controller to adjust emotion in executing motion control.

The proposed RECMAC neural network in this paper is composed of two self-complete sub-networks, compared with the work reported in [29]. The input, in this work, will be delivered to the two networks, and the outputs from the two are merged together in the output layer. The two sub-networks produce similar functions with those of the amygdala and the cerebellum, respectively. For instance, when a human is making decision, emotional stimulus usually affects the decision results. Correspondingly, in the RECMAC network, the output of the RCMAC network can be affected by the output of the emotional network output as expressed in Eq. (23).

The relationship between the two sub networks are defined by the emotional updating rules, i.e., Eqs. (38) and (39), which are different with the two sets of rules owned by the two sub networks. The updated values of the emotional network takes the output of the RCMAC network into account. Dynamic changes in a given target tracking talk usually bring larger tracking errors, as to increase the outputs of the RECMAC network. Such changes can be well handled by the proposed RMCAC as expressed in (39) which increases the weight adjusting values in response to the dynamic changes. All these mechanism ensures the faster response speed of the RECMAC network.

6. Conclusion

This paper proposes a new recurrent neural network, RECMAC, which is used to build the network controller for vision-based robots. By integrating the emotional network and recurrent loop into CMAC, the nonlinear approximation ability and dynamic characteristics of the system were improved. The proposed network was validated by a simulation and applied to the controller of a practical vision-based mobile robot. The controller performed satisfactorily in the mobile object tracking task, which demonstrates the power of the proposed neural network.

Despite of the good performance, there is still room for improvement. It is expected that the application of a self-organization mechanism in RECMAC would make the network more flexible in a dynamic environment, which will be investigated in the future. In addition, the proposed network is currently applied to the task of target tracking only; it will be worthwhile to further explore the application domain such that the proposed system can contribute to the filed more broadly.

Acknowledgement

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. 20720160126), Natural Science Foundation of

Fujian Province of China (No. 2017J01128 and 2017J01129), and the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No. 663830.

[1] M. Shiomi, F. Zanlungo, K. Hayashi, T. Kanda, Towards a socially acceptable collision avoidance for a mobile robot navigating among pedestrians using a pedestrian model, International Journal of Social Robotics 6 (3) (2014) 443–455.

[2] Y. Zhu, T. Zhang, J. Song, X. Li, A new hybrid navigation algorithm for mobile robots in environments with incomplete knowledge, Knowledge-Based Systems 27 (3) (2012) 302–313.

[3] C. J. Ostafew, A. P. Schoellig, T. D. Barfoot, Learning-based nonlinear model predictive control to improve vision-based mobile robot path-tracking in challenging outdoor environments, in: IEEE International Conference on Robotics and Automation, 2016, pp. 4029–4036.

[4] A. Pandey, S. Kumar, K. K. Pandey, D. R. Parhi, Mobile robot navigation in unknown static environments using anfis controller , Perspectives in Science 8 (2016) 421–423.

[5] F. Ciccozzi, D. D. Ruscio, I. Malavolta, P. Pelliccione, Adopting mde for specifying and executing civilian missions of mobile multi-robot systems, IEEE Access 4 (2) (2016) 6451–6466.

[6] D. Zhou, F. Chao, Z. Zhu, C. M. Lin, C. Zhou, A novel approach to a mobile robot via multiple human body postures, in: World Congress on Intelligent Control and Automation, 2016, pp. 1463–1468.

[7] A. K. Khalaji, S. A. A. Moosavian, Robust adaptive controller for a tractorctrailer mobile robot, IEEE/ASME Transactions on Mechatronics 19 (3) (2014) 943–953.

[8] F. Dayoub, T. Morris, P. Corke, Rubbing shoulders with mobile service robots, IEEE Access 3 (2015) 333–342.

[9] J. R. Ruiz-Sarmiento, C. Galindo, J. Gonzalez-Jimenez, Building multiversal semantic maps for mobile robot operation, Knowledge-Based Systems 119 (1) (2017) 257–272.

[10] L. N. Tan, Omnidirectional vision-based distributed optimal tracking control for mobile multi-robot systems with kinematic and dynamic disturbance rejection, IEEE Transactions on Industrial Electronics PP (99) (2017) 1–1.

[11] J. Liao, Z. Chen, B. Yao, Performance-oriented coordinated adaptive robust control for four-wheel independently driven skid steer mobile robot, IEEE Access PP (1) (2017) 1–1.

[12] Z. Li, C. Yang, C. Y. Su, J. Deng, W. Zhang, Vision-based model predictive control for steering of a nonholonomic mobile robot, IEEE Transactions on Control Systems Technology 24 (2) (2016) 553–564.

[13] E. A. Boldbaatar, C. M. Lin, Self-learning fuzzy sliding-mode control for a water bath temperature control system, International Journal of Fuzzy Systems 17 (1) (2015) 31–38.

[14] C. H. Lee, F. Y. Chang, C. M. Lin, An efficient interval Type-2 fuzzy CMAC for chaos time-series prediction and synchronization, IEEE Transactions on Cybernetics 44 (3) (2014) 329–341.

[15] C. H. Chen, C. C. Chung, F. Chao, C. M. Lin, I. J. Rudas, Intelligent robust control for uncertain nonlinear multivariable systems using recurrent cerebellar model neural networks, Acta Polytechnica Hungarica 12 (5) (2015) 7–33.

[16] C. M. Lin, T. Y. Chen, Self-organizing CMAC control for a class of mimo uncertain nonlinear systems, IEEE Transactions on Neural Networks 20 (9) (2009) 1377–1384.

[17] Y. J. Mon, C. M. Lin, Anfis-based integral terminal sliding mode control for disturbed chaotic system, Journal of Intelligent and Fuzzy Systems 27 (1) (2013) 443–450.

[18] N. T. Luy, Robust adaptive dynamic programming based online tracking control algorithm for real wheeled mobile robot with omni-directional vision system, Transactions of the Institute of Measurement and Control 39 (6).

[19] Q. Zhou, F. Chao, C. M. Lin, A functional-link-based fuzzy brain emotional learning network for breast tumor classification and chaotic system synchronization, International Journal of Fuzzy Systems (7) (2017) 1–17.

[20] C. M. Lin, Y. L. Liu, H. Y. Li, Sopc-based function-link cerebellar model articulation control system design for magnetic ball levitation systems, IEEE Transactions on Industrial Electronics 61 (8) (2014) 4265–4273.

[21] D. Zhou, M. Shi, F. Chao, C. M. Lin, L. Yang, C. Shang, C. Zhou, Use of human gestures for controlling a mobile robot via adaptive CMAC network and fuzzy logic controller, Neurocomputing 282 (5) (2018) 218–231.

[22] C. M. Lin, C. S. Hsueh, Adaptive EKF-CMAC-Based multisensor data fusion for maneuvering target, IEEE Transactions on Instrumentation and Measurement 62 (7) (2013) 2058–2066.

[23] C. Balkenius, J. Morn, Emotional learning: A computational model of the amygdala, Cybernetics and Systems 32 (6) (2001) 611–636.

[24] K. Lundagard, C. Balkenius, A computational model of emotional learning in the amygdala, in: The Amygdala, in Jean-Arcady Meyer, Alain Berthoz, Dario Floreano, Herbert L, 2000.

[25] E. Lotfi, M. R. Akbarzadeht, Emotional brain-inspired adaptive fuzzy decayed learning for online prediction problems, in: IEEE International Conference on Fuzzy Systems, 2013, pp. 1–7.

[26] E. Lotfi, O. Khazaei, F. Khazaei, Competitive brain emotional learning, Neural Processing Letters (4) (2017) 1–20.

[27] E. Lotfi, S. Setayeshi, S. Taimory, A Neural Basis Computational Model of Emotional Brain for Online Visual Object Recognition, Taylor and Francis, Inc., 2014.

[28] C. C. Chung, C. M. Lin, Fuzzy brain emotional cerebellar model articulation control system design for multi-input multi-output nonlinear, Acta Polytechnica Hungarica 12 (4) (2015) 39–58.

[29] D. Zhou, F. Chao, C. M. Lin, L. Yang, M. Shi, C. Zhou, Integration of fuzzy CMAC and BELC networks for uncertain nonlinear system control, in: IEEE International Conference on Fuzzy Systems, 2017, pp. 1–6.

[30] M. Jafari, H. Xu, L. R. G. Carrillo, Brain emotional learning-based intelligent controller for flocking of multi-agent systems, in: American Control Conference, 2017, pp. 1996–2001.

[31] R. J. Wai, Y. W. Lin, Adaptive moving-target tracking control of a vision-based mobile robot via a dynamic petri recurrent fuzzy neural network, IEEE Transactions on Fuzzy Systems 21 (4) (2013) 688–701.

[32] C. M. Lin, E. A. Boldbaatar, Autolanding control using recurrent wavelet elman neural network, IEEE Transactions on Systems Man and Cybernetics Systems 45 (9) (2015) 1281–1291.

[33] M. Faisal, M. Algabri, M. A. Bencherif, H. Dhahri, M. A. Rahhal, Human expertise in mobile robot navigation, IEEE Access 6 (9) (2017) 1694–1705.

[34] C. M. Lin, C. S. Hsueh, C. H. Chen, Robust adaptive backstepping control for a class of nonlinear systems using recurrent wavelet neural network, Neurocomputing 142 (1) (2014) 372–382.

[35] F. J. Lin, I. F. Sun, K. J. Yang, J. K. Chang, Recurrent fuzzy neural cerebellar model articulation network fault-tolerant control of six-phase perma-

nent magnet synchronous motor position servo drive, IEEE Transactions on Fuzzy Systems 24 (1) (2016) 153–167.

[36] C. M. Lin, T. L. Le, T. T. Huynh, Self-evolving function-link interval type-2 fuzzy neural network for nonlinear system identification and control, Neurocomputing.

[37] C. M. Lin, E. A. Boldbaatar, Fault accommodation control for a biped robot using a recurrent wavelet elman neural network, IEEE Systems Journal 11 (4) (2017) 2882–2893.

[38] S. Blazic, On periodic control laws for mobile robots, IEEE Transactions on Industrial Electronics 61 (7) (2014) 3660–3670.