

Intelligent Robust Control for Uncertain Nonlinear Multivariable Systems using Recurrent Cerebellar Model Neural Networks

Chiu-Hsiung Chen¹, Chang-Chih Chung², Fei Chao³,
Chih-Min Lin^{4*}, Imre J. Rudas⁵

¹ Electronic System Research Division, Chung-Shan Institute of Science and Technology, Tao-Yuan 320, Taiwan, E-mail: chchchen@cute.edu.tw

² Department of Electrical Engineering, Yuan Ze University, Chung-Li, Tao-Yuan 320, Taiwan, E-mail: s988505@mail.yzu.edu.tw

³ Department of Cognitive Science, Xiamen University, Xiamen, China

^{4*} Corresponding Author, School of Information Science and Engineering, Xiamen University, Xiamen, China; Department of Electrical Engineering and Innovation Center for Big Data and Digital Convergence, Yuan Ze University, Chung-Li, Tao-Yuan 320, Taiwan, E-mail: cml@saturn.yzu.edu.tw

⁵ Institute of Intelligent Engineering Systems, John von Neumann Faculty of Informatics, Óbuda University 1034, Budapest, Hungary, e-mail: rudas@uni-obuda.hu

Abstract: This paper develops an intelligent robust control algorithm for a class of uncertain nonlinear multivariable systems by using a recurrent-cerebellar-model-articulation-controller (RCMAC) and sliding mode technology. The proposed control algorithm consists of an adaptive RCMAC and a robust controller. The adaptive RCMAC is a main tracking controller utilized to mimic an ideal sliding mode controller, and the parameters of the adaptive RCMAC are on-line tuned by the derived adaptive laws from the Lyapunov function. Based on the H^∞ control approach, the robust controller is employed to efficiently suppress the influence of residual approximation error between the ideal sliding mode controller and the adaptive RCMAC, so that the robust tracking performance of the system can be guaranteed. Finally, computer simulation results on a Chua's chaotic circuit and a three-link robot manipulator are performed to verify the effectiveness and feasibility of the proposed control algorithm. The simulation results confirm that the developed control algorithm not only can guarantee the system stability but also achieve an excellent robust tracking performance.

Keywords: recurrent-cerebellar-model-articulation-controller (RCMAC); sliding mode control; H_∞ control; nonlinear multivariable systems

1 Introduction

In recent year, controls of uncertain nonlinear systems have been one of active research topics for many control engineering. Various control efforts have been utilized to design and analyze the uncertain nonlinear systems. Sliding mode control (SMC) has been confirmed as a powerful robust scheme for controlling the nonlinear systems with uncertainties [1], [2]. The most outstanding features of SMC are insensitive to system parameter variations, fast dynamic response and external disturbance rejection [1]. However, in practical applications, SMC suffers two main disadvantages. One is that it requires the system models that may be difficult to obtain in some cases. The other is that because the magnitude of uncertainty bound is unknown, the large uncertainty bound is often required to achieve robust characteristics; however, this will lead the control input chattering.

Neural networks (NNs) possess several advantages such as parallelism, fault tolerance, generalization and powerful approximation capabilities, so that NNs have been applied for system identifications and controls [3]-[6]. Some significant results indicate that the main property of NNs is adaptive learning so that it can uniformly approximate arbitrary input-output linear or nonlinear mappings on closed subsets. Based on this property, a number of researchers have proposed the NN-based adaptive sliding mode controllers which combine the advantages of the sliding mode control with robust characteristics and the NNs with on-line adaptive learning ability; so that the stability, convergence and robustness of the system can be improved [7]-[9]. For example, Lin and Hsu presented an NN-based hybrid adaptive sliding mode control system [7]; in this approach, NN is used as a compensation controller. In [8], Tsai etc. presented a neuro-sliding mode control that utilized two parallel neural networks to realize equivalent control and corrective control; thus the system performance can be improved and the chattering can be eliminated. In [9], Da introduced an identification-based sliding mode control and the bound of uncertainties is also not required. However, the above approaches suffer the computational complexity.

On the neural network structure aspect, NNs can be classified as feedforward neural network (FNN [3], [5], [8], [9]) and recurrent neural network (RNN [4], [6], [7]). As known, FNN is a static mapping. Moreover, the weight updates of FNNs do not utilize the internal network information so that the function approximation is sensitive to the training data. For RNNs, of particular interest is their ability to deal with time varying input or output through their own natural temporal operation [7]. Thus, RNN is a dynamic mapping and demonstrates good control performance in the presence of unmodelled dynamics. However, no matter for

FNNs or RNNs, the learning is slow since all the weights are updated during each learning cycle. Therefore, the effectiveness of NN is limited in problems requiring on-line learning.

Cerebellar-model-articulation-controller (CMAC) is classified as a non-fully connected perceptron-like associative memory network with overlapping receptive-fields [10]; and it intends to resolve the fast size-growing problem and the learning difficult in currently available types of neural networks (NNs). Comparing to neural networks, CMACs possess good generalization capability, fast learning ability and simple computation [10], [11]. This network has already been shown to be able to approximate a nonlinear function over a domain of interest to any desired accuracy [11]-[13]. For the reasons, CMACs have adopted widely for the closed-loop control of complex dynamical systems in recent literatures [14]-[17]. However, the major drawback of existing CMACs is that their application domain is limited to static problem due to their inherent network structure.

In order to resolve the static CMAC problem and preserve the main advantage of SMC with robust characteristics, this paper develops an intelligent robust control algorithm for a class of uncertain nonlinear multivariable systems via sliding mode technology. The proposed control system is comprised of an adaptive recurrent CMAC (RCMAC) and a robust controller. The adaptive RCMAC is a main tracking controller utilized to mimic an ideal sliding mode controller, and the parameters of the adaptive RCMAC are on-line tuned by the derived adaptive laws. Moreover, based on the H^∞ control approach, the robust controller is employed to efficiently suppress the influence of residual approximation error between the ideal sliding mode controller and the adaptive RCMAC, so that the robust tracking performance of the system can be guaranteed. Finally, two examples are presented to support the validity of the proposed control algorithm.

2 System Description

Consider the n th-order multivariable nonlinear systems expressed in the following form:

$$\begin{aligned} \mathbf{x}^{(n)}(t) &= \mathbf{f}(\underline{\mathbf{x}}(t)) + \mathbf{G}(\underline{\mathbf{x}}(t))\mathbf{u}(t) + \mathbf{d}(t), \\ \mathbf{y}(t) &= \mathbf{x}(t) \end{aligned} \quad (1)$$

where

- $\mathbf{u}(t) = [u_1(t), u_2(t), \dots, u_m(t)]^T \in \mathfrak{R}^m$ is the control input vector of the system,
- $\mathbf{y}(t) = \mathbf{x}(t) = [x_1(t), x_2(t), \dots, x_m(t)]^T \in \mathfrak{R}^m$ is the system output vector,
- $\underline{\mathbf{x}}(t) = [\mathbf{x}^T(t), \dot{\mathbf{x}}^T(t), \dots, \mathbf{x}^{(n-1)T}(t)]^T \in \mathfrak{R}^{mn}$ is the state vector of the system,

- $\mathbf{f}(\underline{\mathbf{x}}(t)) \in \mathfrak{R}^m$ is an unknown but bounded smooth nonlinear function,
- $\mathbf{G}(\underline{\mathbf{x}}(t)) \in \mathfrak{R}^{m \times m}$ is an unknown but bounded control input gain matrix,
- $\mathbf{d}(t) = [d_1(t), d_2(t), \dots, d_m(t)]^T \in \mathfrak{R}^m$ is an external bounded disturbance.

Assume that the nominal model of the multivariable nonlinear systems (1) can be represented as

$$\dot{\mathbf{x}}^{(n)}(t) = \mathbf{f}_n(\underline{\mathbf{x}}(t)) + \mathbf{G}_n \mathbf{u}(t), \quad (2)$$

where $\mathbf{f}_n(\underline{\mathbf{x}}(t))$ is the nominal function of $\mathbf{f}(\underline{\mathbf{x}}(t))$ and \mathbf{G}_n is the nominal constant gain of $\mathbf{G}(\underline{\mathbf{x}}(t))$. By appropriately choosing the control parameters and suitably arranging the control inputs and their directions, \mathbf{G}_n can be chosen to be positive definite and invertible. If the external disturbance and uncertainties are included, the multivariable nonlinear systems (1) can be described as

$$\begin{aligned} \dot{\mathbf{x}}^{(n)}(t) &= \mathbf{f}_n(\underline{\mathbf{x}}(t)) + \Delta \mathbf{f}(\underline{\mathbf{x}}(t)) + [\mathbf{G}_n + \Delta \mathbf{G}(\underline{\mathbf{x}}(t))] \mathbf{u}(t) + \mathbf{d}(t) \\ &= \mathbf{f}_n(\underline{\mathbf{x}}(t)) + \mathbf{G}_n \mathbf{u}(t) + \mathbf{l}(\underline{\mathbf{x}}(t), t), \end{aligned} \quad (3)$$

where $\Delta \mathbf{f}(\underline{\mathbf{x}}(t))$ and $\Delta \mathbf{G}(\underline{\mathbf{x}}(t))$ denote the system uncertainties, $\mathbf{l}(\underline{\mathbf{x}}(t), t)$ is referred to as the lumped uncertainty, defined as $\mathbf{l}(\underline{\mathbf{x}}(t), t) = \Delta \mathbf{f}(\underline{\mathbf{x}}(t)) + \Delta \mathbf{G}(\underline{\mathbf{x}}(t)) \mathbf{u}(t) + \mathbf{d}(t)$. Then (1) can be expressed as state and output equations as follows:

$$\begin{aligned} \dot{\underline{\mathbf{x}}}(t) &= \mathbf{A}_m \underline{\mathbf{x}}(t) + \mathbf{B}_m [\mathbf{f}_n(\underline{\mathbf{x}}(t)) + \mathbf{G}_n \mathbf{u}(t) + \mathbf{l}(\underline{\mathbf{x}}(t), t)], \\ \mathbf{y}(t) &= \mathbf{C}_m^T \underline{\mathbf{x}}(t), \end{aligned} \quad (4)$$

$$\text{where } \mathbf{A}_m = \begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \dots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \end{bmatrix}, \mathbf{B}_m = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{I} \end{bmatrix}, \mathbf{C}_m = \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}.$$

The objective of a control system is to design a suitable controller such that the system state vector $\underline{\mathbf{x}}(t)$ can track a desired trajectory $\underline{\mathbf{x}}_d(t) = [\mathbf{x}_d^T(t), \dot{\mathbf{x}}_d^T(t), \dots, \mathbf{x}_d^{(n-1)T}(t)]^T \in \mathfrak{R}^{mn}$. To begin with, define the tracking error $\mathbf{e}(t) = \mathbf{x}_d(t) - \mathbf{x}(t) \in \mathfrak{R}^m$, and the tracking error vector of the system is defined as $\underline{\mathbf{e}}(t) = [\mathbf{e}^T(t), \dot{\mathbf{e}}^T(t), \dots, \mathbf{e}^{(n-1)T}(t)]^T \in \mathfrak{R}^{mn}$. The reference trajectory dynamic equation can be expressed as

$$\dot{\underline{\mathbf{x}}}_d(t) = \mathbf{A}_m \underline{\mathbf{x}}_d(t) + \mathbf{B}_m \mathbf{x}_d^{(n)}(t). \quad (5)$$

Subtracting (4) from (5), gives

$$\dot{\underline{e}}(t) = \underline{A}_m \underline{e}(t) + \underline{B}_m [\underline{x}_d^{(n)}(t) - \underline{f}_n(\underline{x}(t)) - \underline{G}_n \underline{u}(t) - \underline{l}(\underline{x}(t), t)]. \quad (6)$$

3 Sliding Mode Control System

Sliding mode control (SMC) is one of the effective nonlinear robust control schemes since it provides system dynamics with an invariance property to uncertainties once the system dynamics are controlled in the sliding mode [1], [2]. In general, SMC design can be derived into two phases, that is the reaching phase and the sliding phase. The system state trajectory in the period of time before reaching the sliding surface is called the reaching phase. Once the system trajectory reaching the sliding surface, it stays on it and slides along the sliding surface to the origin is the sliding phase. When the states of the controlled system enter the sliding mode, the dynamics of the system are determined by the pre-specified sliding surface and are independent of uncertainties. In order to implement SMC, the first step is to select a sliding surface that models the desired closed-loop performance in state variable space. Then, design the control law such that the system state trajectories are forced toward the sliding surface and stay on it. Thus, the sliding hyperplane can be defined as:

$$s(\underline{e}(t)) = \left(\frac{d}{dt} + \lambda \right)^{n-1} \underline{e}(t) = \underline{K}^T \underline{e}(t), \quad (7)$$

where $\underline{K} = [\lambda^{n-1} \underline{I}, (n-1)\lambda^{n-2} \underline{I}, \dots, \underline{I}]^T \in \mathfrak{R}^{m \times m}$ satisfies that all roots of the equation:

$$q^{n-1} \underline{I} + (n-1)\lambda q^{n-2} \underline{I} + \dots + (n-1)\lambda^{n-2} q \underline{I} + \lambda^{n-1} \underline{I} = \mathbf{0} \quad (8)$$

are in the open left half-plane, in which q is the Laplace operator. The process of SMC can be divided into two phases, that is the reaching phase with $s(\underline{e}(t)) \neq \mathbf{0}$ and the sliding phase with $s(\underline{e}(t)) = 0$. If the sliding mode exists on the sliding surface, then the motion of the system is governed by the linear differential equation presented in (7) whose behavior is dictated by the sliding surface design [1], [2]. Thus, the tracking error vector decays exponentially to zero, so that perfect tracking can be asymptotically achieved. Thus the control objective becomes the design of a control law to force $s(\underline{e}(t)) = 0$. A sufficient condition for the existence and reachable of the sliding hyperplane in the system state space is to choose the control law such that the following reaching condition is satisfied:

$$\frac{1}{2} \frac{d}{dt} (s^T(\underline{e}(t)) s(\underline{e}(t))) = s^T(\underline{e}(t)) \dot{s}(\underline{e}(t)) = \sum_{i=1}^m s_i(t) \dot{s}_i(t) \leq - \sum_{i=1}^m \sigma_i |s_i(t)|, \quad (9)$$

where σ_i is a small positive constant. Taking the time derivative of both sides of (7) and using (6), yields

$$\dot{s}(\underline{e}(t)) = \mathbf{K}^T \dot{\underline{e}}(t) = \mathbf{K}^T \mathbf{A}_m \underline{e}(t) + \mathbf{x}_d^{(n)}(t) - \mathbf{f}_n(\underline{\mathbf{x}}(t)) - \mathbf{G}_n \mathbf{u}(t) - \mathbf{l}(\underline{\mathbf{x}}(t), t). \quad (10)$$

Therefore, an ideal sliding mode controller \mathbf{u}_{ISMC} which guarantees the reaching condition must satisfy the following condition:

$$\begin{aligned} \mathbf{s}^T(\underline{e}(t))\dot{s}(\underline{e}(t)) &= \mathbf{s}^T(\underline{e}(t))[\mathbf{K}^T \mathbf{A}_m \underline{e}(t) + \mathbf{x}_d^{(n)}(t) - \mathbf{f}_n(\underline{\mathbf{x}}(t)) - \mathbf{G}_n \mathbf{u}(t) - \mathbf{l}(\underline{\mathbf{x}}(t), t)] \\ &\leq -\sum_{i=1}^m \sigma_i |s_i(t)|. \end{aligned} \quad (11)$$

If the system dynamics and the lumped uncertainty are exactly known, an ideal sliding mode controller can be designed as follows to satisfy inequality (11)

$$\mathbf{u}_{ISMC} = \mathbf{G}_n^{-1} [-\mathbf{f}_n(\underline{\mathbf{x}}(t)) - \mathbf{l}(\underline{\mathbf{x}}(t), t) + \mathbf{x}_d^{(n)}(t) + \mathbf{K}^T \mathbf{A}_m \underline{e}(t) + \boldsymbol{\sigma} \operatorname{sgn}(\mathbf{s}(\underline{e}(t)))], \quad (12)$$

where $\operatorname{sgn}(\cdot)$ is a sign function and $\boldsymbol{\sigma} = \operatorname{diag}(\sigma_1, \dots, \sigma_i, \dots, \sigma_m)$. However, in practical applications, the dynamical functions are not precisely known, and the lumped uncertainty is always unknown. Therefore, the ideal sliding mode controller in (12) is unobtainable. Thus, an intelligent robust control algorithm based on RCMAC and sliding mode technology is proposed in the following section to achieve robust tracking performance.

4 Intelligent Robust Control Algorithm

The configuration of the intelligent robust control algorithm, which consists of an adaptive RCMAC and a robust controller, is depicted in Fig. 1.

The control system is assumed to take the following form:

$$\mathbf{u} = \mathbf{u}_{ARCMAC} + \mathbf{u}_{RC}, \quad (13)$$

where \mathbf{u}_{ARCMAC} is an adaptive RCMAC and \mathbf{u}_{RC} is a robust controller. The adaptive RCMAC \mathbf{u}_{ARCMAC} is a main tracking controller utilized to mimic the ideal sliding mode controller, and the parameters of the adaptive RCMAC are on-line tuned by the derived adaptive laws from the Lyapunov function. The robust controller \mathbf{u}_{RC} is employed to efficiently suppress the influence of residual approximation error between the ideal sliding mode controller and adaptive RCMAC, so that the robust tracking performance of the system can be guaranteed.

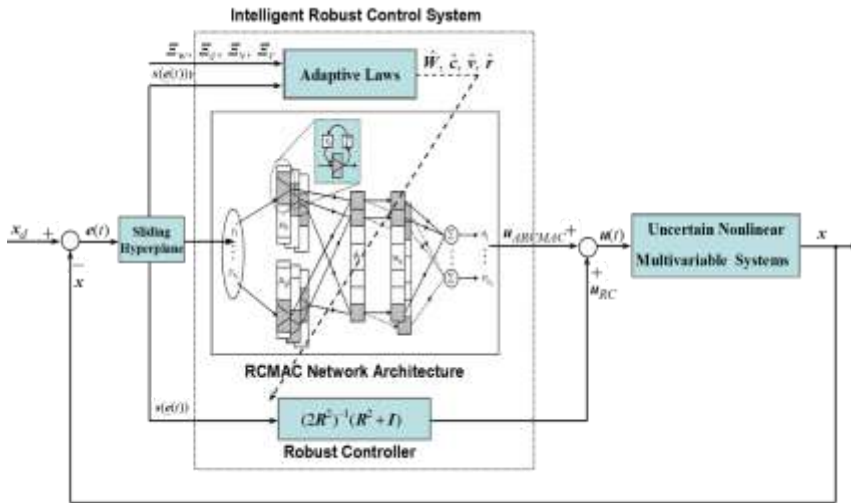


Figure 1
The configuration of the intelligent robust control system

4.1 Description of RCMAC

An RCMAC is proposed and shown in Fig. 2, in which T denotes a time delay. This RCMAC is composed of input space, association memory space with recurrent weights, receptive-field space, weight memory space and output space.

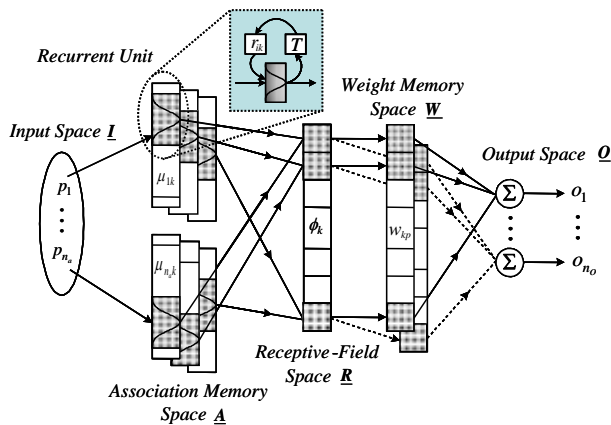


Figure 2
Architecture of an RCMAC

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

1) *Input space \underline{I}* : For a given $\mathbf{p} = [p_1, p_2, \dots, p_{n_a}]^T \in \mathfrak{R}^{n_a}$, where n_a is the number of input state variables, each input state variable p_i must be quantized into discrete regions (called *elements*) according to given control space. The number of elements, n_e , is termed as a resolution.

2) *Association memory space \underline{A}* : Several elements can be accumulated as a *block*, the number of blocks, n_b , is usually greater than or equal to two. \underline{A} denotes an association memory space with n_c ($n_c = n_a \times n_b$) components. In this space, each block performs a receptive-field basis function, the Gaussian function is adopted here as the receptive-field basis function, which can be represented as

$$\mu_{ik} = \exp\left[\frac{-(p_{rik} - c_{ik})^2}{v_{ik}^2}\right], \text{ for } k = 1, 2, \dots, n_b, \quad (14)$$

where μ_{ik} represents the output of the k -th receptive-field basis function for the i -th input with the mean c_{ik} and variance v_{ik} . In addition, the input of this block can be represented as

$$p_{rik}(t) = p_i(t) + r_{ik} \mu_{ik}(t-T), \quad (15)$$

where r_{ik} is the recurrent weight, and $\mu_{ik}(t-T)$ denotes the value of μ_{ik} through delay time T . It is clear that the input of this block contains the memory term $\mu_{ik}(t-T)$, which stores the past information of the network and presents a dynamic mapping. Figure 3 depicts the schematic diagram of a two-dimensional RCMAC with $n_e = 5$ and $n_f = 4$ (n_f is the number of elements in a complete block); in which p_1 is divided into blocks B_{a1} and B_{b1} , and p_2 is divided into blocks B_{a2} and B_{b2} . By shifting each variable an element, different blocks will be obtained. For instance, blocks B_{c1} and B_{d1} for p_1 , and blocks B_{c2} and B_{d2} for p_2 are possible shifted elements for the second layer; and B_{e1} and B_{f1} for p_1 , and B_{e2} and B_{f2} for p_2 for the third layer; and B_{g1} and B_{h1} for p_1 , and B_{g2} and B_{h2} for p_2 for the fourth layer. The receptive-field basis function μ_{ik} of each block in this space has three adjustable parameters c_{ik} , v_{ik} and r_{ik} .

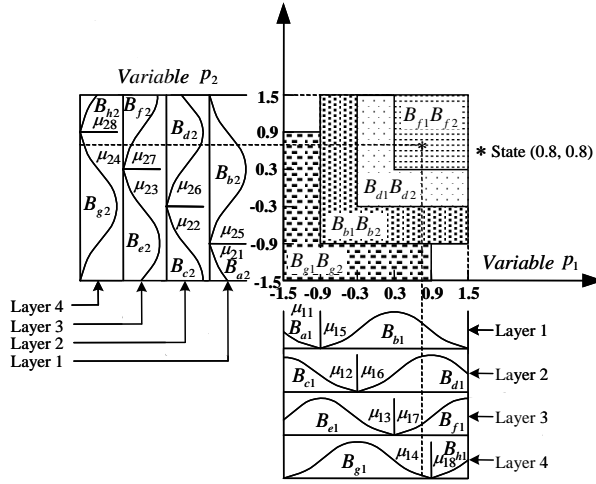


Figure 3

A two-dimensional RCMAC with $n_f = 4$ and $n_e = 5$

3) *Receptive-field space $\underline{\mathbf{R}}$* : Areas formed by blocks, referred to as $B_{d1}B_{a2}$ and $B_{b1}B_{b2}$ are called receptive-fields. The number of receptive-fields, n_d , is equal to n_b in this study. The k -th multi-dimensional receptive-field function is defined as

$$\phi_k(\mathbf{p}, \mathbf{c}_k, \mathbf{v}_k, \mathbf{r}_k) = \prod_{i=1}^{n_d} \mu_{ik} = \exp \left[\sum_{i=1}^{n_d} \frac{-(p_{rik} - c_{ik})^2}{v_{ik}^2} \right] \text{ for } k = 1, 2, \dots, n_d, \quad (16)$$

where $\mathbf{c}_k = [c_{1k}, c_{2k}, \dots, c_{n_d k}]^T \in \mathfrak{R}^{n_d}$, $\mathbf{v}_k = [v_{1k}, v_{2k}, \dots, v_{n_d k}]^T \in \mathfrak{R}^{n_d}$ and $\mathbf{r}_k = [r_{1k}, r_{2k}, \dots, r_{n_d k}]^T \in \mathfrak{R}^{n_d}$. The multi-dimensional receptive-field functions can be expressed in a vector form as

$$\Phi(\mathbf{p}, \mathbf{c}, \mathbf{v}, \mathbf{r}) = [\phi_1, \dots, \phi_k, \dots, \phi_{n_d}]^T, \quad (17)$$

where $\mathbf{c} = [\mathbf{c}_1^T, \dots, \mathbf{c}_k^T, \dots, \mathbf{c}_{n_d}^T]^T \in \mathfrak{R}^{n_d n_d}$, $\mathbf{v} = [\mathbf{v}_1^T, \dots, \mathbf{v}_k^T, \dots, \mathbf{v}_{n_d}^T]^T \in \mathfrak{R}^{n_d n_d}$ and $\mathbf{r} = [\mathbf{r}_1^T, \dots, \mathbf{r}_k^T, \dots, \mathbf{r}_{n_d}^T]^T \in \mathfrak{R}^{n_d n_d}$.

4) *Weight memory space $\underline{\mathbf{W}}$* : Each location of $\underline{\mathbf{R}}$ to a particular adjustable value in the weight memory space can be expressed as

$$\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_p, \dots, \mathbf{w}_{n_d}] = \begin{bmatrix} w_{11} & \cdots & w_{1p} & \cdots & w_{1n_d} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ w_{k1} & \cdots & w_{kp} & \cdots & w_{kn_d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_{n_d 1} & \cdots & w_{n_d p} & \cdots & w_{n_d n_d} \end{bmatrix} \in \mathfrak{R}^{n_d n_d}, \quad (18)$$

where $\mathbf{w}_p = [w_{1p}, \dots, w_{kp}, \dots, w_{n_p p}]^T \in \mathfrak{R}^{n_d}$, and w_{kp} denotes the connecting weight value of the p -th output associated with the k -th receptive-field.

5) *Output space* $\underline{\mathbf{O}}$: The output of RCMAC is the algebraic sum of the activated weights in the weight memory, and is expressed as

$$o_p = \mathbf{w}_p^T \Phi = \sum_{k=1}^{n_d} w_{kp} \phi_k, \text{ for } p=1, 2, \dots, n_o. \quad (19)$$

The outputs of RCMAC can be expressed in a vector notation as

$$\mathbf{o} = [o_1, \dots, o_p, \dots, o_{n_o}]^T = \mathbf{W}^T \Phi. \quad (20)$$

In the two-dimensional case shown in Fig. 3, the output of RCMAC is the sum of the value in receptive-fields $B_{b1}B_{b2}$, $B_{d1}B_{d2}$, $B_{f1}B_{f2}$ and $B_{g1}B_{g2}$, where the input state is (0.8,0.8). The architecture of RCMAC is designed to have the advantages of simple structure with dynamic characteristics. The role of the recurrent loops is to consider the past value of the receptive-field basis function in the association memory space. Thus, this RCMAC has dynamic characteristics.

4.2 Robust Controller Design

Subtracting (12) from (10), yields

$$\dot{s}(\underline{e}(t)) = \mathbf{G}_n [\mathbf{u}_{ISMC} - \mathbf{u}] - \sigma \operatorname{sgn}[s(\underline{e}(t))]. \quad (21)$$

Assume there exists an optimal RCMAC \mathbf{u}_{ARCMAC}^* to estimate the ideal sliding mode controller \mathbf{u}_{ISMC} such that

$$\mathbf{u}_{ISMC} = \mathbf{u}_{ARCMAC}^* (\mathbf{p}, \mathbf{W}^*, \mathbf{c}^*, \mathbf{v}^*, \mathbf{r}^*) + \boldsymbol{\varepsilon} \equiv \mathbf{W}^{*T} \Phi^* + \boldsymbol{\varepsilon}, \quad (22)$$

where $\boldsymbol{\varepsilon} = [\varepsilon_1, \dots, \varepsilon_i, \dots, \varepsilon_m]^T$ is a minimum reconstructed error vector; \mathbf{W}^* , Φ^* , \mathbf{c}^* , \mathbf{v}^* and \mathbf{r}^* are the optimal parameter matrix and vectors of \mathbf{W} , Φ , \mathbf{c} , \mathbf{v} and \mathbf{r} , respectively. However, the optimal RCMAC cannot be obtained; thus, an estimating RCMAC is used to estimate the optimal RCMAC. From (20), the control law (13) can be rewritten as follows:

$$\mathbf{u}(t) = \mathbf{u}_{ARCMAC} (\mathbf{p}, \hat{\mathbf{W}}, \hat{\mathbf{c}}, \hat{\mathbf{v}}, \hat{\mathbf{r}}) + \mathbf{u}_{RC} \equiv \hat{\mathbf{W}}^T \hat{\Phi} + \mathbf{u}_{RC}, \quad (23)$$

where $\hat{\mathbf{W}}$, $\hat{\Phi}$, $\hat{\mathbf{c}}$, $\hat{\mathbf{v}}$ and $\hat{\mathbf{r}}$ are the estimated matrix and vectors of \mathbf{W}^* , Φ^* , \mathbf{c}^* , \mathbf{v}^* and \mathbf{r}^* , respectively. Thus, the dynamic equation (21) can be expressed via (22) and (23) as

$$\dot{s}(\underline{e}(t)) = \mathbf{G}_n [\mathbf{u}_{ARCMAC}^* + \boldsymbol{\varepsilon} - \mathbf{u}_{ARCMAC} - \mathbf{u}_{RC}] - \sigma \operatorname{sgn}[s(\underline{e}(t))]$$

$$\begin{aligned}
&= \mathbf{G}_n [\mathbf{W}^{*T} \boldsymbol{\Phi}^* - \hat{\mathbf{W}}^T \hat{\boldsymbol{\Phi}} + \boldsymbol{\varepsilon} - \mathbf{u}_{RC}] - \sigma \operatorname{sgn}[s(\underline{\mathbf{e}}(t))] \\
&= \mathbf{G}_n [\tilde{\mathbf{W}}^T \boldsymbol{\Phi}^* + \hat{\mathbf{W}}^T \tilde{\boldsymbol{\Phi}} + \boldsymbol{\varepsilon} - \mathbf{u}_{RC}] - \sigma \operatorname{sgn}[s(\underline{\mathbf{e}}(t))], \tag{24}
\end{aligned}$$

where $\tilde{\mathbf{W}} = \mathbf{W}^* - \hat{\mathbf{W}}$ and $\tilde{\boldsymbol{\Phi}} = \boldsymbol{\Phi}^* - \hat{\boldsymbol{\Phi}}$. Moreover, the linearization technique is employed to transform the multi-dimensional receptive-field basis functions into a partially linear form. The expansion of $\tilde{\boldsymbol{\Phi}}$ in Taylor series can be obtained as

$$\begin{aligned}
\tilde{\boldsymbol{\Phi}} = \begin{bmatrix} \tilde{\phi}_1 \\ \vdots \\ \tilde{\phi}_k \\ \vdots \\ \tilde{\phi}_{n_d} \end{bmatrix} &= \begin{bmatrix} \left(\frac{\partial \phi_1}{\partial \mathbf{c}} \right)^T \\ \vdots \\ \left(\frac{\partial \phi_k}{\partial \mathbf{c}} \right)^T \\ \vdots \\ \left(\frac{\partial \phi_{n_d}}{\partial \mathbf{c}} \right)^T \end{bmatrix} \Big|_{\mathbf{c}=\hat{\mathbf{c}}} (\mathbf{c}^* - \hat{\mathbf{c}}) + \begin{bmatrix} \left(\frac{\partial \phi_1}{\partial \mathbf{v}} \right)^T \\ \vdots \\ \left(\frac{\partial \phi_k}{\partial \mathbf{v}} \right)^T \\ \vdots \\ \left(\frac{\partial \phi_{n_d}}{\partial \mathbf{v}} \right)^T \end{bmatrix} \Big|_{\mathbf{v}=\hat{\mathbf{v}}} (\mathbf{v}^* - \hat{\mathbf{v}}) + \begin{bmatrix} \left(\frac{\partial \phi_1}{\partial \mathbf{r}} \right)^T \\ \vdots \\ \left(\frac{\partial \phi_k}{\partial \mathbf{r}} \right)^T \\ \vdots \\ \left(\frac{\partial \phi_{n_d}}{\partial \mathbf{r}} \right)^T \end{bmatrix} \Big|_{\mathbf{r}=\hat{\mathbf{r}}} (\mathbf{r}^* - \hat{\mathbf{r}}) + \boldsymbol{\beta} \\
&\equiv \boldsymbol{\Phi}_c \tilde{\mathbf{c}} + \boldsymbol{\Phi}_v \tilde{\mathbf{v}} + \boldsymbol{\Phi}_r \tilde{\mathbf{r}} + \boldsymbol{\beta}, \tag{25}
\end{aligned}$$

where
$$\boldsymbol{\Phi}_c = \left[\frac{\partial \phi_1}{\partial \mathbf{c}}, \dots, \frac{\partial \phi_k}{\partial \mathbf{c}}, \dots, \frac{\partial \phi_{n_d}}{\partial \mathbf{c}} \right]^T \Big|_{\mathbf{c}=\hat{\mathbf{c}}} \in \mathfrak{R}^{n_d \times n_c n_d};$$

$$\boldsymbol{\Phi}_v = \left[\frac{\partial \phi_1}{\partial \mathbf{v}}, \dots, \frac{\partial \phi_k}{\partial \mathbf{v}}, \dots, \frac{\partial \phi_{n_d}}{\partial \mathbf{v}} \right]^T \Big|_{\mathbf{v}=\hat{\mathbf{v}}} \in \mathfrak{R}^{n_d \times n_v n_d};$$

$$\boldsymbol{\Phi}_r = \left[\frac{\partial \phi_1}{\partial \mathbf{r}}, \dots, \frac{\partial \phi_k}{\partial \mathbf{r}}, \dots, \frac{\partial \phi_{n_d}}{\partial \mathbf{r}} \right]^T \Big|_{\mathbf{r}=\hat{\mathbf{r}}} \in \mathfrak{R}^{n_d \times n_r n_d}, \quad \tilde{\mathbf{c}} = \mathbf{c}^* - \hat{\mathbf{c}}; \quad \tilde{\mathbf{v}} = \mathbf{v}^* - \hat{\mathbf{v}}; \quad \tilde{\mathbf{r}} = \mathbf{r}^* - \hat{\mathbf{r}} \text{ and}$$

$\boldsymbol{\beta} \in \mathfrak{R}^{n_d}$ is a vector of higher-order terms. Moreover, $\frac{\partial \phi_k}{\partial \mathbf{c}}$, $\frac{\partial \phi_k}{\partial \mathbf{v}}$ and $\frac{\partial \phi_k}{\partial \mathbf{r}}$ are defined as

$$\left[\frac{\partial \phi_k}{\partial \mathbf{c}} \right] = \left[\underbrace{0, \dots, 0}_{(k-1) \times n_c}, \frac{\partial \phi_k}{\partial c_{1k}}, \dots, \frac{\partial \phi_k}{\partial c_{n_c k}}, \underbrace{0, \dots, 0}_{(n_d-k) \times n_c} \right], \tag{26}$$

$$\left[\frac{\partial \phi_k}{\partial \mathbf{v}} \right] = \left[\underbrace{0, \dots, 0}_{(k-1) \times n_v}, \frac{\partial \phi_k}{\partial v_{1k}}, \dots, \frac{\partial \phi_k}{\partial v_{n_v k}}, \underbrace{0, \dots, 0}_{(n_d-k) \times n_v} \right], \tag{27}$$

$$\left[\frac{\partial \phi_k}{\partial \mathbf{r}} \right] = \left[\underbrace{0, \dots, 0}_{(k-1) \times n_r}, \frac{\partial \phi_k}{\partial r_{1k}}, \dots, \frac{\partial \phi_k}{\partial r_{n_r k}}, \underbrace{0, \dots, 0}_{(n_d-k) \times n_r} \right]. \tag{28}$$

Rewriting (25), it can be obtained that

$$\boldsymbol{\Phi}^* = \hat{\boldsymbol{\Phi}} + \boldsymbol{\Phi}_c \tilde{\mathbf{c}} + \boldsymbol{\Phi}_v \tilde{\mathbf{v}} + \boldsymbol{\Phi}_r \tilde{\mathbf{r}} + \boldsymbol{\beta}. \tag{29}$$

Substituting (25) and (29) into (24), yields

$$\begin{aligned} \dot{s}(\underline{e}(t)) &= \mathbf{G}_n [\tilde{\mathbf{W}}^T (\hat{\Phi} + \Phi_c \tilde{c} + \Phi_v \tilde{v} + \Phi_r \tilde{r} + \beta) + \hat{\mathbf{W}}^T (\Phi_c \tilde{c} + \Phi_v \tilde{v} + \Phi_r \tilde{r} + \beta) + \varepsilon - \mathbf{u}_{RC}] - \sigma \operatorname{sgn}[s(\underline{e}(t))] \\ &= \mathbf{G}_n [\tilde{\mathbf{W}}^T \hat{\Phi} + \hat{\mathbf{W}}^T (\Phi_c \tilde{c} + \Phi_v \tilde{v} + \Phi_r \tilde{r}) + \tilde{\mathbf{W}}^T (\Phi_c \tilde{c} + \Phi_v \tilde{v} + \Phi_r \tilde{r}) + \mathbf{W}^{*T} \beta + \varepsilon - \mathbf{u}_{RC}] - \sigma \operatorname{sgn}[s(\underline{e}(t))] \\ &= \mathbf{G}_n [\tilde{\mathbf{W}}^T \hat{\Phi} + \hat{\mathbf{W}}^T (\Phi_c \tilde{c} + \Phi_v \tilde{v} + \Phi_r \tilde{r}) + \omega - \mathbf{u}_{RC}] - \sigma \operatorname{sgn}[s(\underline{e}(t))], \end{aligned} \quad (30)$$

where the approximation error $\omega \equiv \mathbf{W}^{*T} \beta + \tilde{\mathbf{W}}^T (\Phi_c \tilde{c} + \Phi_v \tilde{v} + \Phi_r \tilde{r}) + \varepsilon$.

In case of the existence of ω , consider a specified H_∞ tracking performance [18]

$$\begin{aligned} \sum_{i=1}^m \int_0^T s_i^2(t) dt \leq s^T(0) \mathbf{G}_n^{-1} s(0) + \operatorname{tr}[\tilde{\mathbf{W}}^T(0) \Xi_w^{-1} \tilde{\mathbf{W}}(0)] + \tilde{c}^T(0) \Xi_c^{-1} \tilde{c}(0) \\ + \tilde{v}^T(0) \Xi_v^{-1} \tilde{v}(0) + \tilde{r}^T(0) \Xi_r^{-1} \tilde{r}(0) + \sum_{i=1}^m \lambda_i^2 \int_0^T \omega_i^2(t) dt, \end{aligned} \quad (31)$$

where Ξ_w , Ξ_c , Ξ_v and Ξ_r are diagonal positive constant learning-rate matrices, and λ_i is a prescribed attenuation constant. If the system starts with initial conditions $s(0) = \mathbf{0}$, $\tilde{\mathbf{W}}(0) = \mathbf{0}$, $\tilde{c}(0) = \mathbf{0}$, $\tilde{v}(0) = \mathbf{0}$, $\tilde{r}(0) = \mathbf{0}$, then the H_∞ tracking performance in (31) can be rewritten as

$$\sup_{\omega_i \in L_2[0, T]} \sum_{i=1}^m \left(\frac{\|s_i\|}{\|\omega_i\|} \leq \lambda_i \right), \quad (32)$$

where $\|s_i\|^2 = \int_0^T s_i^2(t) dt$ and $\|\omega_i\|^2 = \int_0^T \omega_i^2(t) dt$. This shows that λ_i is an attenuation level between the approximation error $\omega_i(t)$ and system output function $s_i(t)$. If $\lambda_i = \infty$, this is the case of minimum error tracking control without approximation attenuation [18]. Therefore, the following theorem can be stated and proved.

Theorem 1: Consider the n th-order multivariable nonlinear systems represented by (1). The intelligent robust control system is defined as in (13), in which the adaptive laws of RCMAC are designed as in (33)-(36) and the robust controller is designed as in (37). Then, the robust tracking performance in (31) can be achieved for the prescribed attenuation level λ_i , $i = 1, 2, \dots, m$, where $\mathbf{R} = \operatorname{diag}[\lambda_1, \lambda_2, \dots, \lambda_m] \in \Re^{m \times m}$ is a diagonal matrix.

$$\dot{\hat{\mathbf{W}}} = \Xi_w \hat{\Phi} s^T(\underline{e}(t)), \quad (33)$$

$$\dot{\hat{c}} = \Xi_c \Phi_c^T \hat{\mathbf{W}} s(\underline{e}(t)), \quad (34)$$

$$\dot{\hat{v}} = \Xi_v \Phi_v^T \hat{\mathbf{W}} s(\underline{e}(t)), \quad (35)$$

$$\dot{\hat{\mathbf{r}}} = \underline{\mathbf{E}}_r \underline{\Phi}_r^T \hat{\mathbf{W}} s(\underline{\mathbf{e}}(t)), \quad (36)$$

$$\mathbf{u}_{RC} = (2\mathbf{R}^2)^{-1}(\mathbf{R}^2 + \mathbf{I}) s(\underline{\mathbf{e}}(t)). \quad (37)$$

Proof: The Lyapunov function candidate is given by

$$V(s(\underline{\mathbf{e}}(t)), \tilde{\mathbf{W}}, \tilde{\mathbf{c}}, \tilde{\mathbf{v}}, \tilde{\mathbf{r}}) = \frac{1}{2} \left[s^T(\underline{\mathbf{e}}(t)) \mathbf{G}_n^{-1} s(\underline{\mathbf{e}}(t)) + tr(\tilde{\mathbf{W}}^T \underline{\mathbf{E}}_w^{-1} \tilde{\mathbf{W}}) + \tilde{\mathbf{c}}^T \underline{\mathbf{E}}_c^{-1} \tilde{\mathbf{c}} + \tilde{\mathbf{v}}^T \underline{\mathbf{E}}_v^{-1} \tilde{\mathbf{v}} + \tilde{\mathbf{r}}^T \underline{\mathbf{E}}_r^{-1} \tilde{\mathbf{r}} \right]. \quad (38)$$

Taking the derivative of the Lyapunov function and using (30), yields

$$\begin{aligned} \dot{V}(s(\underline{\mathbf{e}}(t)), \tilde{\mathbf{W}}, \tilde{\mathbf{c}}, \tilde{\mathbf{v}}, \tilde{\mathbf{r}}) &= s^T(\underline{\mathbf{e}}(t)) \mathbf{G}_n^{-1} \dot{s}(\underline{\mathbf{e}}(t)) + tr(\tilde{\mathbf{W}}^T \underline{\mathbf{E}}_w^{-1} \dot{\tilde{\mathbf{W}}}) + \tilde{\mathbf{c}}^T \underline{\mathbf{E}}_c^{-1} \dot{\tilde{\mathbf{c}}} + \tilde{\mathbf{v}}^T \underline{\mathbf{E}}_v^{-1} \dot{\tilde{\mathbf{v}}} + \tilde{\mathbf{r}}^T \underline{\mathbf{E}}_r^{-1} \dot{\tilde{\mathbf{r}}} \\ &= s^T(\underline{\mathbf{e}}(t)) [\tilde{\mathbf{W}}^T \hat{\Phi} + \hat{\mathbf{W}}^T (\Phi_c \tilde{\mathbf{c}} + \Phi_v \tilde{\mathbf{v}} + \Phi_r \tilde{\mathbf{r}}) + \omega - \mathbf{u}_{RC}] - s^T(\underline{\mathbf{e}}(t)) \mathbf{G}_n^{-1} \sigma sgn[s(\underline{\mathbf{e}}(t))] \\ &\quad - tr(\tilde{\mathbf{W}}^T \underline{\mathbf{E}}_w^{-1} \dot{\tilde{\mathbf{W}}}) - \tilde{\mathbf{c}}^T \underline{\mathbf{E}}_c^{-1} \dot{\tilde{\mathbf{c}}} - \tilde{\mathbf{v}}^T \underline{\mathbf{E}}_v^{-1} \dot{\tilde{\mathbf{v}}} - \tilde{\mathbf{r}}^T \underline{\mathbf{E}}_r^{-1} \dot{\tilde{\mathbf{r}}} \end{aligned} \quad (39)$$

It can be noted that $s^T(\underline{\mathbf{e}}(t)) \tilde{\mathbf{W}}^T \hat{\Phi} = tr(\tilde{\mathbf{W}}^T \hat{\Phi} s^T(\underline{\mathbf{e}}(t)))$ and $s^T(\underline{\mathbf{e}}(t)) \mathbf{G}_n^{-1} \sigma sgn[s(\underline{\mathbf{e}}(t))] \geq 0$, so (39) can be rewritten as

$$\begin{aligned} \dot{V}(s(\underline{\mathbf{e}}(t)), \tilde{\mathbf{W}}, \tilde{\mathbf{c}}, \tilde{\mathbf{v}}, \tilde{\mathbf{r}}) &\leq tr \left(\tilde{\mathbf{W}}^T [\hat{\Phi} s^T(\underline{\mathbf{e}}(t)) - \underline{\mathbf{E}}_w^{-1} \dot{\tilde{\mathbf{W}}}] \right) + \tilde{\mathbf{c}}^T (\Phi_c^T \hat{\mathbf{W}} s(\underline{\mathbf{e}}(t)) - \underline{\mathbf{E}}_c^{-1} \dot{\tilde{\mathbf{c}}}) \\ &+ \tilde{\mathbf{v}}^T (\Phi_v^T \hat{\mathbf{W}} s(\underline{\mathbf{e}}(t)) - \underline{\mathbf{E}}_v^{-1} \dot{\tilde{\mathbf{v}}}) + \tilde{\mathbf{r}}^T (\Phi_r^T \hat{\mathbf{W}} s(\underline{\mathbf{e}}(t)) - \underline{\mathbf{E}}_r^{-1} \dot{\tilde{\mathbf{r}}}) + [s^T(\underline{\mathbf{e}}(t)) \omega - s^T(\underline{\mathbf{e}}(t)) \mathbf{u}_{RC}]. \end{aligned} \quad (40)$$

From (33)-(36) and using (37), (40) can be rewritten as

$$\begin{aligned} \dot{V}(s(t), \tilde{\mathbf{W}}, \tilde{\mathbf{c}}, \tilde{\mathbf{v}}, \tilde{\mathbf{r}}) &\leq \sum_{i=1}^m [s_i(t) \omega_i(t) - s_i^2(t) \frac{\lambda_i^2 + 1}{2 \lambda_i^2}] \\ &= \sum_{i=1}^m [s_i(t) \omega_i(t) - \frac{s_i^2(t)}{2} - \frac{s_i^2(t)}{2 \lambda_i^2}] \\ &= \sum_{i=1}^m [-\frac{s_i^2(t)}{2} - \frac{1}{2} (\frac{s_i(t)}{\lambda_i} - \lambda_i \omega_i(t))^2 + \frac{\lambda_i^2 \omega_i^2(t)}{2}] \\ &\leq \sum_{i=1}^m [-\frac{s_i^2(t)}{2} + \frac{\lambda_i^2 \omega_i^2(t)}{2}]. \end{aligned} \quad (41)$$

Assuming $\omega_i \in L_2[0, T], \forall T \in [0, \infty)$, integrating the above equation from $t=0$ to $t=T$, yields

$$V(T) - V(0) \leq \sum_{i=1}^m [-\frac{1}{2} \int_0^T s_i^2(t) dt + \frac{\lambda_i^2}{2} \int_0^T \omega_i^2(t) dt]. \quad (42)$$

Since $V(T) \geq 0$, the above inequality implies the following inequality

$$\frac{1}{2} \sum_{i=1}^m \int_0^T s_i^2(t) dt \leq V(0) + \frac{1}{2} \sum_{i=1}^m \lambda_i^2 \int_0^T \omega_i^2(t) dt . \quad (43)$$

Using (38), the above inequality is equivalent to the following

$$\begin{aligned} \sum_{i=1}^m \int_0^T s_i^2(t) dt &\leq s^T(0) \mathbf{G}_n^{-1} s(0) + \text{tr}[\tilde{\mathbf{W}}^T(0) \mathbf{\Xi}_w^{-1} \tilde{\mathbf{W}}(0)] + \tilde{\mathbf{c}}^T(0) \mathbf{\Xi}_c^{-1} \tilde{\mathbf{c}}(0) \\ &\quad + \tilde{\mathbf{v}}^T(0) \mathbf{\Xi}_v^{-1} \tilde{\mathbf{v}}(0) + \tilde{\mathbf{r}}^T(0) \mathbf{\Xi}_r^{-1} \tilde{\mathbf{r}}(0) + \sum_{i=1}^m \lambda_i^2 \int_0^T \omega_i^2(t) dt . \end{aligned} \quad (44)$$

Thus the proof is completed.

5 Simulation Results

To illustrate the effectiveness of the proposed control system, it is applied to control a Chua's chaotic circuit and a three-links robot manipulator. Moreover, an adaptive fuzzy neural network controller (AFNNC) [19] and the proposed RCMAC are applied to these two systems for comparison.

Example 1: Chua's chaotic circuit

A typical Chua's chaotic circuit consists of one linear resistor (R), two capacitors (C_1, C_2), one inductor (L) and one nonlinear resistor ($g(v_{c_1})$) as shown in Fig. 4.

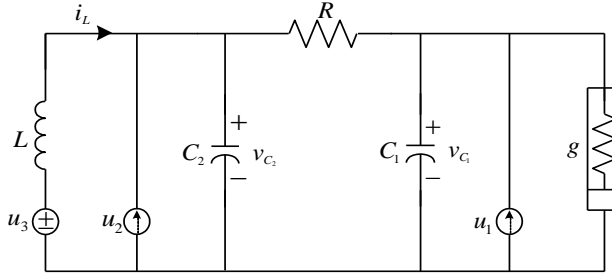


Figure 4
Chua's chaotic circuit

The dynamic equations of the Chua's circuit are written as [20]

$$\dot{v}_{c_1} = \frac{1}{C_1} \left(\frac{1}{R} (v_{c_2} - v_{c_1}) - g(v_{c_1}) + u_1(t) \right) + d_1(t), \quad (45)$$

$$\dot{v}_{c_2} = \frac{1}{C_2} \left(\frac{1}{R} (v_{c_1} - v_{c_2}) + i_L + u_2(t) \right) + d_2(t), \quad (46)$$

$$\dot{i}_L = \frac{1}{L}(-v_{C_2} + u_3(t)) + d_3(t), \quad (47)$$

where $\mathbf{u}(t) = [u_1(t), u_2(t), u_3(t)]^T$ denotes the control input and $\mathbf{d}(t) = [d_1(t), d_2(t), d_3(t)]^T$ denotes the external disturbance. The voltages $v_{C_1}(t)$, $v_{C_2}(t)$ and the current $i_L(t)$ are the state variables. Thus, the state vector of chaotic system is defined as $\mathbf{x}(t) = [v_{C_1}(t), v_{C_2}(t), i_L(t)]^T = [x_1(t), x_2(t), x_3(t)]^T$. The dynamic equation (45)-(47) can be rewritten as

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u}(t) + \mathbf{d}(t), \quad (48)$$

$$\text{where } \mathbf{f}(\mathbf{x}) = \begin{bmatrix} \frac{1}{C_1} \left(\frac{1}{R} (v_{C_2} - v_{C_1}) - g(v_{C_1}) \right) \\ \frac{1}{C_2} \left(\frac{1}{R} (v_{C_1} - v_{C_2}) + i_L \right) \\ \frac{1}{L} (-v_{C_2}) \end{bmatrix} \text{ and } \mathbf{G}(\mathbf{x}) = \text{diag} \begin{bmatrix} \frac{1}{C_1} & 0 & 0 \\ 0 & \frac{1}{C_2} & 0 \\ 0 & 0 & \frac{1}{L} \end{bmatrix}.$$

$$\text{The external disturbance is given as } \mathbf{d}(t) = \begin{bmatrix} d_1(t) \\ d_2(t) \\ d_3(t) \end{bmatrix} = \begin{bmatrix} \sin(2t) \exp(-0.2t) + 0.3 \\ \cos(2t) \exp(-0.2t) - 0.5 \\ \sin(3t) \exp(-0.2t) + 0.2 \end{bmatrix}.$$

The physical parameters of chaotic circuit are assumed as $R = R_0 + \Delta R$, $g(v_{C_1}) = g_0(v_{C_1}) + \Delta g(v_{C_1})$, $L = L_0 + \Delta L$, $C_1 = C_{10} + \Delta C_1$, $C_2 = C_{20} + \Delta C_2$, where R_0 , $g_0(v_{C_1})$, L_0 , C_{10} and C_{20} are the nominal values and ΔR , $\Delta g(v_{C_1})$, ΔL , ΔC_1 and ΔC_2 denote the unknown nonlinear time-varying perturbations [19]. The nominal values are given as $R_0 = 5$, $g_0(v_{C_1}) = -v_{C_1} + 0.02v_{C_1}^3$, $L_0 = 1$, $C_{10} = 1$, $C_{20} = 0.5$. The time-varying perturbations are $\Delta R = \sin(t/2)$, $\Delta g(v_{C_1}) = 0.2\sin(t)v_{C_1}$, $\Delta L = 0.15$, $\Delta C_1 = 0.1 + 0.1\cos(t/2)$, $\Delta C_2 = 0.1$. The desired trajectories come from the reference model outputs that are chosen as $\dot{x}_{di}(t) = -4x_{di}(t) + 4\gamma_i$, where γ_i is the input signal to the reference model. The initial conditions of the Chua's chaotic circuit and the reference models are given as $x_1(0) = -1$, $x_2(0) = 0$ and $x_3(0) = 0$, $x_{d1}(0) = 0$, $x_{d2}(0) = -1$, and $x_{d3}(0) = 1$. The reference inputs are unit periodic rectangular signals. For the proposed control scheme, the sliding hyperplane is design as $s(\mathbf{e}(t)) = \mathbf{e}(t)$. The proposed RCMAC is characterized as:

- number of input state variables: $n_a = 3$,
- number of elements for each state variable: $n_e = 5$ (elements),
- generalization: $n_f = 4$ (elements/ block),
- number of blocks for each state variable: $n_b = 2$ (blocks/layer) $\times 4$ (layer) = 8 (blocks),
- number of receptive-fields: $n_d = 2$ (receptive-fields/layer) $\times 4$ (layer) = 8 (receptive-fields),
- receptive-field basis functions: $\mu_{ik} = \exp[-(p_{rk} - c_{ik})^2 / v_{ik}^2]$ for $i = 1, 2, 3$ and $k = 1, 2, \dots, 8$.

The inputs of RCMAC are $s_1(t)$, $s_2(t)$ and $s_3(t)$; while the input spaces of input signals are normalized within $\{[-2, 2][-2, 2][-2, 2]\}$. The initial means of the Gaussian functions are divided equally and are set as $[c_{i1}, c_{i2}, c_{i3}, c_{i4}, c_{i5}, c_{i6}, c_{i7}, c_{i8}] = [-2.8, -2, -1.2, -0.4, 0.4, 1.2, 2, 2.8]$ and the initial variances are set as $v_{ik} = 1.6$ for $i = 1, 2, 3$ and $k = 1, 2, \dots, 8$. The learning-rate matrices of RCMAC are selected as $\mathbf{E}_w = 30\mathbf{I}_{8 \times 8}$, $\mathbf{E}_c = \mathbf{E}_v = \mathbf{E}_r = 0.5\mathbf{I}_{24 \times 24}$ and the specified attenuation constant diagonal matrix is $\mathbf{R} = 0.2\mathbf{I}_{3 \times 3}$.

The simulation results of AFNNC for the Chua's chaotic circuit are shown in Fig. 5. The trajectories of the system states are plotted in Figs. 5(a)-(c) for $v_{C_1}(t)$, $v_{C_2}(t)$ and $i_L(t)$, respectively. The associated control efforts $u_1(t)$, $u_2(t)$, $u_3(t)$ are depicted in Figs. 5(d)-(f). Moreover, the sliding hyperplanes $s_1(t)$, $s_2(t)$ and $s_3(t)$ are shown in Figs. 5(g)-(i). The simulation results of RCMAC for the Chua's chaotic circuit are shown in Fig. 6. The trajectories of the system states are plotted in Figs. 6(a)-(c) for $v_{C_1}(t)$, $v_{C_2}(t)$ and $i_L(t)$, respectively. The associated control efforts $u_1(t)$, $u_2(t)$, $u_3(t)$ are depicted in Figs. 6(d)-(f). Moreover, the sliding hyperplanes $s_1(t)$, $s_2(t)$ and $s_3(t)$ are shown in Figs. 6(g)-(i). From the simulation results, it can be seen that the proposed RCMAC can provide better control performance with smaller tracking error than the AFNNC.

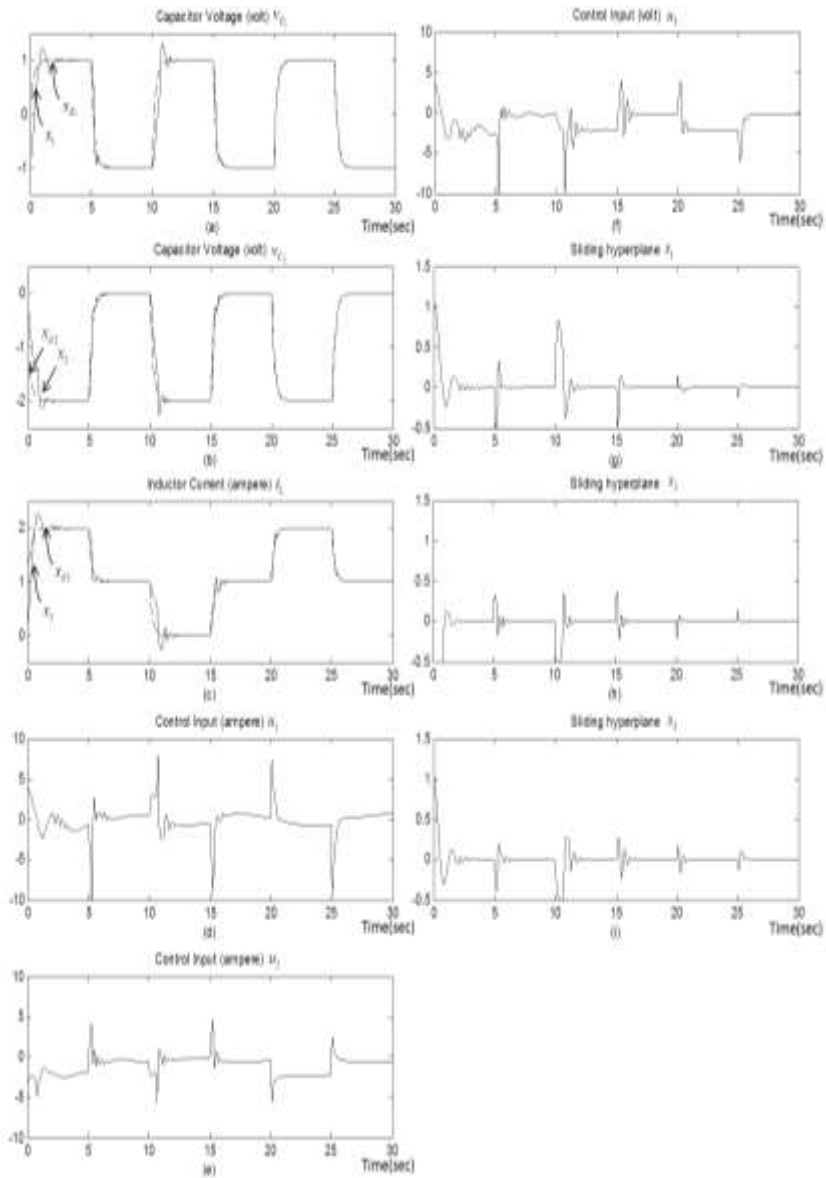


Figure 5

The numerical simulations of AFNNC for the Chua's chaotic circuit, (a)-(c) The trajectories of the system states, (d)-(f) The associated control efforts, (g)-(i) The sliding hyperplane

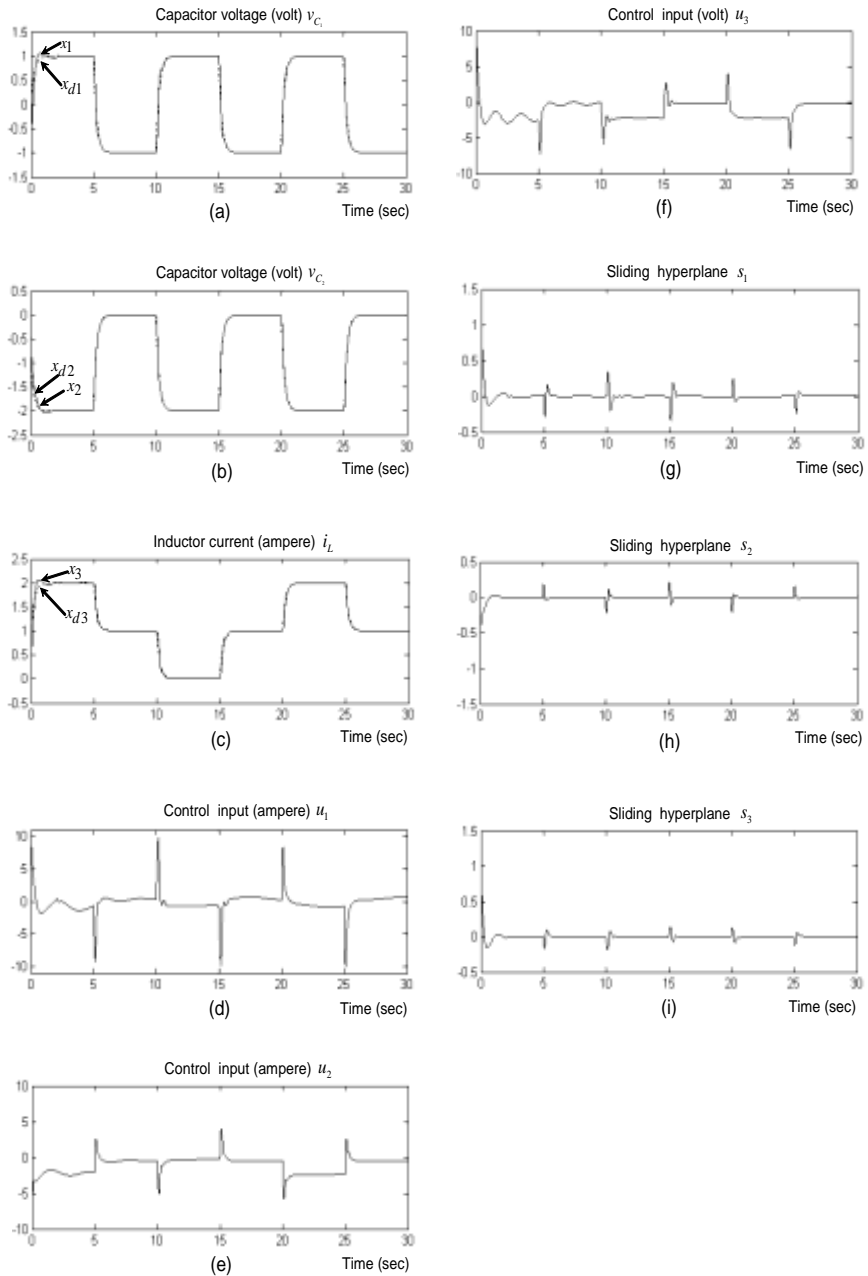


Figure 6

The numerical simulations of RCMAC for the Chua's chaotic circuit, (a)-(c) The trajectories of the system states, (d)-(f) The associated control efforts, (g)-(i) The sliding hyperplane

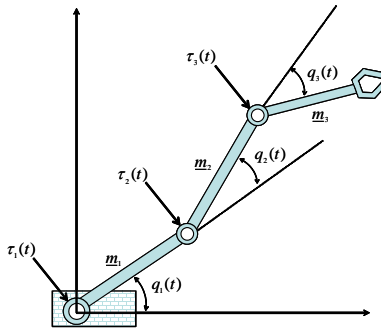
Example 2: A three-links robot manipulator

Figure 7

A three-links robot manipulator

A three-links robot manipulator is depicted as Fig. 7. The dynamic equation is given as follows [21]:

$$\underline{M}(q)\ddot{q} + \underline{C}(q, \dot{q})\dot{q} + \underline{g}(q) + \tau_d = \tau, \quad (49)$$

where

$$\underline{M}(q) = \begin{bmatrix} 2\underline{d}_1 + \underline{d}_4\underline{c}_2 + \underline{d}_5\underline{c}_{23} & \underline{d}_2 + \underline{d}_4\underline{c}_2 + \underline{d}_6\underline{c}_3 & \underline{d}_3 + \underline{d}_5\underline{c}_{23} + \underline{d}_6\underline{c}_3 \\ \underline{d}_4\underline{c}_2 + \underline{d}_5\underline{c}_{23} & \underline{d}_2 + \underline{d}_6\underline{c}_3 & \underline{d}_3 + \underline{d}_6\underline{c}_3 \\ \underline{d}_5\underline{c}_{23} & \underline{d}_6\underline{c}_3 & \underline{d}_3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix},$$

$$\underline{C}(q, \dot{q}) = \begin{bmatrix} \begin{Bmatrix} -\dot{q}_2\underline{d}_4\underline{s}_2 - \dot{q}_3\underline{d}_5\underline{s}_{23} \\ -\dot{q}_3\underline{d}_5\underline{s}_{23} - \dot{q}_3\underline{d}_6\underline{s}_3 \end{Bmatrix} & \begin{Bmatrix} -\dot{q}_2\underline{d}_4\underline{s}_2 - \dot{q}_3\underline{d}_5\underline{s}_{23} \\ -\dot{q}_3\underline{d}_6\underline{s}_3 - \dot{q}_3\underline{d}_5\underline{s}_{23} \\ -\dot{q}_1\underline{d}_4\underline{s}_2 - \dot{q}_1\underline{d}_5\underline{s}_{23} \end{Bmatrix} & \begin{Bmatrix} -\dot{q}_2\underline{d}_5\underline{s}_{23} - \dot{q}_3\underline{d}_5\underline{s}_{23} \\ -\dot{q}_3\underline{d}_6\underline{s}_3 - \dot{q}_1\underline{d}_5\underline{s}_{23} \\ -\dot{q}_1\underline{d}_6\underline{s}_3 - \dot{q}_2\underline{d}_6\underline{s}_3 \end{Bmatrix} \\ -\dot{q}_3\underline{d}_6\underline{s}_3 + \dot{q}_1\underline{d}_4\underline{s}_2 + \dot{q}_1\underline{d}_5\underline{s}_{23} & -\dot{q}_3\underline{d}_6\underline{s}_3 & -\underline{d}_6\underline{s}_3(\dot{q}_1 + \dot{q}_2 + \dot{q}_3) \\ \dot{q}_1\underline{d}_5\underline{s}_{23} + \dot{q}_1\underline{d}_6\underline{s}_3 + \dot{q}_2\underline{d}_6\underline{s}_3 & \underline{d}_6\underline{s}_3(\dot{q}_1 + \dot{q}_2) & 0 \end{bmatrix},$$

$$\underline{g}(q) = \begin{bmatrix} \frac{1}{2}\underline{a}_1\underline{c}_1 & \underline{a}_1\underline{c}_1 + \frac{1}{2}\underline{a}_2\underline{c}_{12} & \underline{a}_1\underline{c}_1 + \underline{a}_2\underline{c}_{12} + \frac{1}{2}\underline{a}_3\underline{c}_{123} \\ 0 & \frac{1}{2}\underline{a}_2\underline{c}_{12} & \underline{a}_2\underline{c}_{12} + \frac{1}{2}\underline{a}_3\underline{c}_{123} \\ 0 & 0 & \frac{1}{2}\underline{a}_3\underline{c}_{123} \end{bmatrix} \begin{bmatrix} \underline{m}_1\underline{g} \\ \underline{m}_2\underline{g} \\ \underline{m}_3\underline{g} \end{bmatrix} \quad \text{and}$$

$$\tau_d = \begin{bmatrix} 0.2 \sin(2t) \\ 0.1 \cos(2t) \\ 0.1 \sin(t) \end{bmatrix}.$$

In (39), $\mathbf{q} = [q_1(t), q_2(t), q_3(t)]^T \in \mathfrak{R}^3$ is the angular position vector, $\dot{\mathbf{q}}, \ddot{\mathbf{q}} \in \mathfrak{R}^3$ are the joint velocity and acceleration vector, respectively, $\underline{\mathbf{M}}(\mathbf{q}) \in \mathfrak{R}^{3 \times 3}$ is the inertia matrix, $\boldsymbol{\tau} \in \mathfrak{R}^3$ is the input torque vector, $\underline{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathfrak{R}^{3 \times 3}$ is the Coriolis/Centripetal matrix, $\underline{\mathbf{g}}(\mathbf{q}) \in \mathfrak{R}^3$ is the gravity vector, and $\boldsymbol{\tau}_d \in \mathfrak{R}^3$ is the external disturbance. The acceleration of gravity is $\underline{g} = 9.8 \text{ m/s}^2$. \underline{m}_i is the link mass; \underline{a}_i is the link length; the short hand notations are defined as $\underline{s}_{ij} = \sin(\underline{q}_i + \underline{q}_j)$, $\underline{c}_{ij} = \cos(\underline{q}_i + \underline{q}_j)$; and \underline{d}_i is defined as in Table 1. In Table 1, \underline{i}_i denotes the moment of inertia (kg m^2). The detail data of system parameters are given in Table 1.

Table 1
The system parameters of robot manipulator

\underline{d}_i	$\underline{d}_1 = 0.5[(0.25\underline{m}_1 + \underline{m}_2 + \underline{m}_3)\underline{a}_1^2 + \underline{i}_1]$ $\underline{d}_4 = [0.5\underline{m}_2 + \underline{m}_3]\underline{a}_1\underline{a}_2$	$\underline{d}_2 = 0.5[(0.25\underline{m}_2 + \underline{m}_3)\underline{a}_2^2 + \underline{i}_2]$ $\underline{d}_5 = 0.5\underline{m}_3\underline{a}_1\underline{a}_3$	$\underline{d}_3 = 0.5[(0.25\underline{m}_3)\underline{a}_3^2 + \underline{i}_3]$ $\underline{d}_6 = 0.5\underline{m}_3\underline{a}_2\underline{a}_3$
\underline{a}_i	$\underline{a}_1 = 0.5 \text{ m}$	$\underline{a}_2 = 0.4 \text{ m}$	$\underline{a}_3 = 0.3 \text{ m}$
\underline{m}_i	$\underline{m}_1 = 1.2 \text{ kg}$	$\underline{m}_2 = 1.5 \text{ kg}$	$\underline{m}_3 = 3.0 \text{ kg}$
\underline{i}_i	$\underline{i}_1 = 43.33 \times 10^{-3} \text{ kgm}^2$	$\underline{i}_2 = 25.08 \times 10^{-3} \text{ kgm}^2$	$\underline{i}_3 = 32.67 \times 10^{-3} \text{ kgm}^2$

The dynamic equation (52) can be expressed as

$$\ddot{\mathbf{x}}(t) = \mathbf{f}(\underline{\mathbf{x}}(t)) + \mathbf{G}(\underline{\mathbf{x}}(t))\mathbf{u}(t) + \mathbf{d}(t), \quad (50)$$

where

$$\mathbf{x}(t) \triangleq [q_1(t), q_2(t), q_3(t)]^T = [x_1(t), x_2(t), x_3(t)]^T,$$

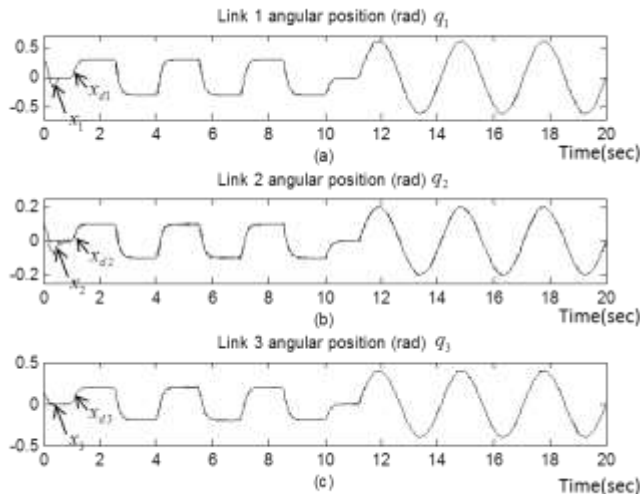
$$\mathbf{f}(\underline{\mathbf{x}}(t)) = -\underline{\mathbf{M}}^{-1}(\mathbf{q})[\underline{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \underline{\mathbf{g}}(\mathbf{q})], \quad \mathbf{G}(\underline{\mathbf{x}}(t)) = \underline{\mathbf{M}}^{-1}(\mathbf{q}), \quad \mathbf{d}(t) = -\underline{\mathbf{M}}^{-1}(\mathbf{q})\boldsymbol{\tau}_d$$

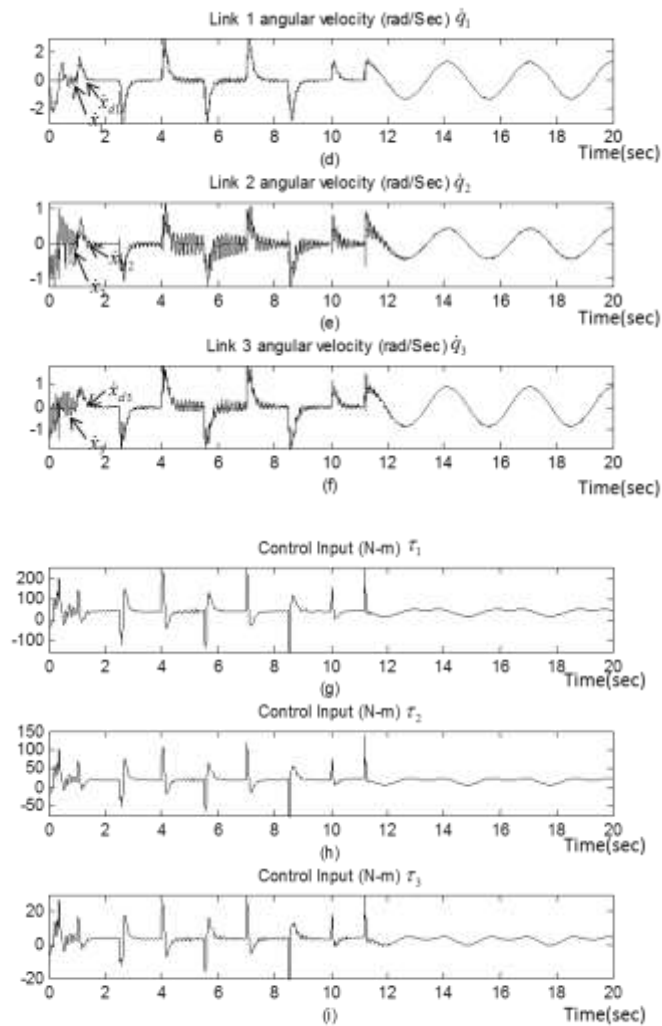
and $\mathbf{u}(t) = [\tau_1(t), \tau_2(t), \tau_3(t)]^T \in \mathfrak{R}^3$. The reference trajectories are described as a reference model output and a sinusoid function at different time. When $t \leq 11.2$ sec, the reference models are described as $\ddot{x}_{di}(t) = -21.13\dot{x}_{di}(t) - 111.63x_{di}(t) + 111.63\gamma_i$, for $i = 1, 2, 3$. The initial conditions of the robot manipulator are given as $x_1(0) = 0.3$, $x_2(0) = 0.1$, $x_3(0) = 0.2$, $\dot{x}_1(0) = 0$, $\dot{x}_2(0) = 0$ and $\dot{x}_3(0) = 0$. The initial conditions of the reference models are given as $x_{d1}(0) = 0$, $x_{d2}(0) = 0$, $x_{d3}(0) = 0$, $\dot{x}_{d1}(0) = 0$, $\dot{x}_{d2}(0) = 0$ and $\dot{x}_{d3}(0) = 0$. The reference inputs are unit periodic rectangular signals. When $t \geq 11.2$ sec, a sinusoid function command is used. For the proposed control scheme, the sliding hyperplane is designed as $s(\mathbf{e}(t)) = \dot{\mathbf{e}}(t) + 10\mathbf{e}(t)$. The proposed RCMAC is characterized as:

- number of input state variables: $n_a = 3$,
- number of elements for each state variable: $n_e = 5$ (elements),

- generalization: $n_f = 4$ (elements/ block),
- number of blocks for each state variable: $n_b = 2$ (blocks/layer) $\times 4$ (layer) = 8 (blocks),
- number of receptive-fields: $n_d = 2$ (receptive-fields/layer) $\times 4$ (layer) = 8 (receptive-fields),
- receptive-field basis functions: $\mu_k = \exp[-(p_{rk} - c_k)^2 / v_k^2]$ for $i = 1, 2, 3$ and $k = 1, 2, \dots, 8$.

The inputs of RCMAC are $s_1(t)$, $s_2(t)$ and $s_3(t)$; while the input spaces of input signals are normalized within $\{[-1.5, 1.5][-1.5, 1.5][-1.5, 1.5]\}$. The initial means of the Gaussian functions are divided equally and are set as $[c_{i1}, c_{i2}, c_{i3}, c_{i4}, c_{i5}, c_{i6}, c_{i7}, c_{i8}] = [-2.1, -1.5, -0.9, -0.3, 0.3, 0.9, 1.5, 2.1]$ and the initial variances are set as $v_k = 1.2$ for $i = 1, 2, 3$ and $k = 1, 2, \dots, 8$. The learning-rate matrices of RCMAC are chosen as $\bar{\mathbf{E}}_w = 50\mathbf{I}_{8 \times 8}$, $\bar{\mathbf{E}}_c = \bar{\mathbf{E}}_v = \bar{\mathbf{E}}_r = 0.05\mathbf{I}_{24 \times 24}$ and the specified attenuation constant diagonal matrix $\mathbf{R} = 0.35\mathbf{I}_{3 \times 3}$. The simulation results of AFNNC for the three-links robot manipulator are shown in Fig. 8. The trajectories of the system states are plotted in Figs. 8(a)-(f) for $q_1(t)$, $q_2(t)$, $q_3(t)$, $\dot{q}_1(t)$, $\dot{q}_2(t)$ and $\dot{q}_3(t)$, respectively. The associated control efforts $u_1(t)$, $u_2(t)$, $u_3(t)$ are depicted in Figs. 8(g)-(i). Moreover, the sliding hyperplanes $s_1(t)$, $s_2(t)$ and $s_3(t)$ are shown in Figs. 8(j)-(l).





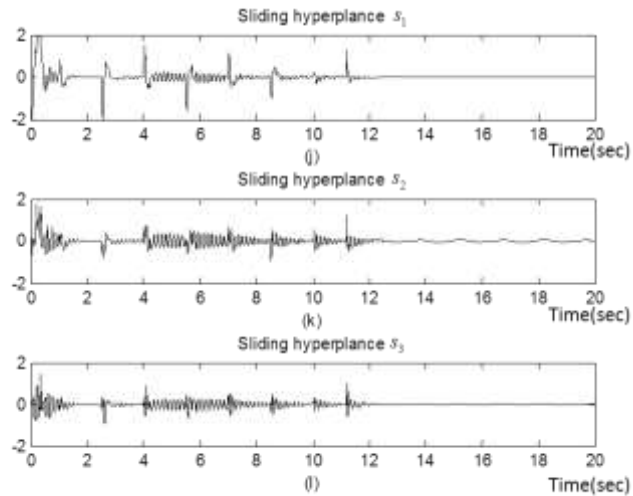
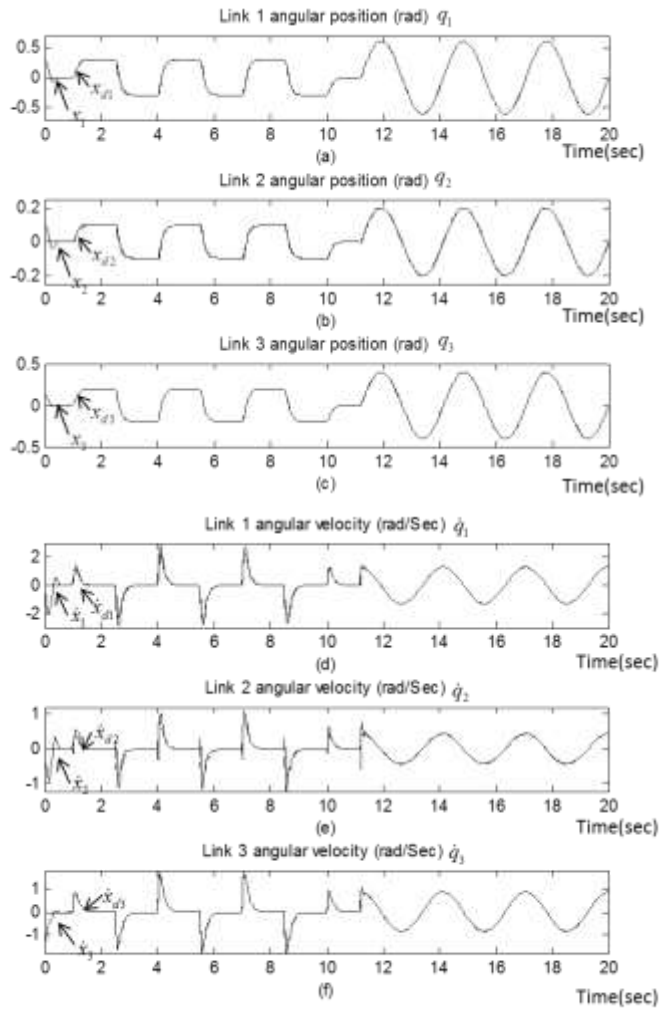


Figure 8

The numerical simulations of AFNNC for the three-links robot manipulator, (a)-(f) The trajectories of the system states, (g)-(i) The associated control efforts, (j)-(l) The sliding hyperplane

The simulation results of RCMAC for the three-links robot manipulator are shown in Fig. 9. The trajectories of the system states are plotted in Figs. 9(a)-(f) for $q_1(t)$, $q_2(t)$, $q_3(t)$, $\dot{q}_1(t)$, $\dot{q}_2(t)$ and $\dot{q}_3(t)$, respectively. The associated control efforts $u_1(t)$, $u_2(t)$, $u_3(t)$ are depicted in Figs. 9(g)-(i). The sliding hyperplanes $s_1(t)$, $s_2(t)$ and $s_3(t)$ are shown in Figs. 9(j)-(l). From the simulation results comparison, the proposed RCMAC can also achieve better control performance with smaller tracking error than the AFNNC. Moreover, the chattering phenomenon in AFNNC has been much reduced by applying RCMAC.



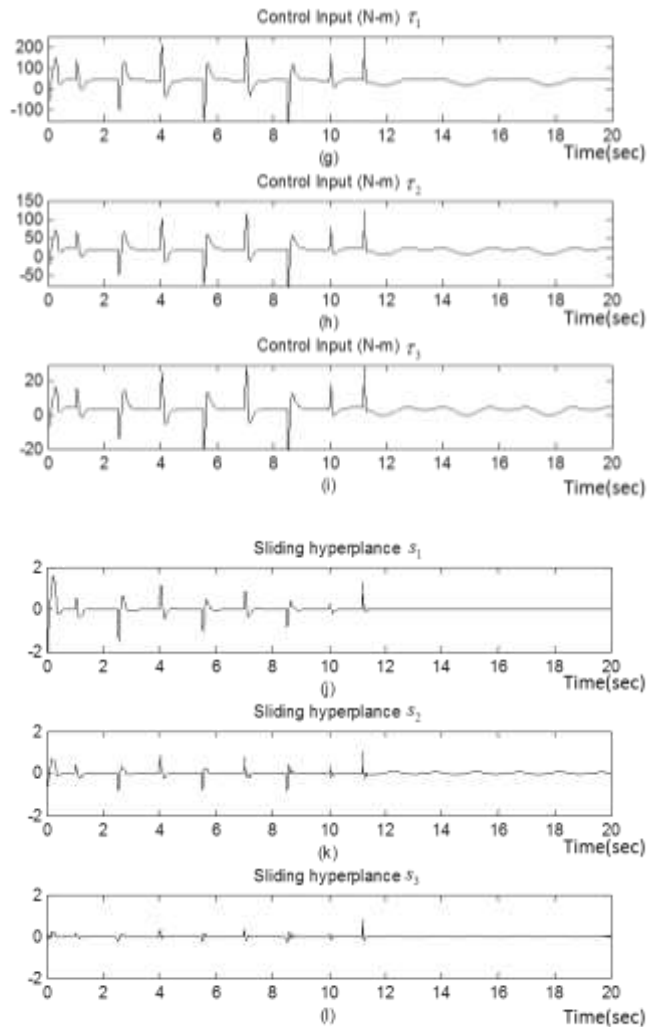


Figure 9

The numerical simulations of RCMAC for the three-links robot manipulator, (a)-(f) The trajectories of the system states, (g)-(i) The associated control efforts, (j)-(l) The sliding hyperplane

Conclusions

This paper proposes an intelligent robust control system for a class of uncertain nonlinear multivariable systems via sliding mode technology. The proposed control system consists of an adaptive RCMAC and a robust controller. The adaptive RCMAC is a main tracking controller utilized to mimic the ideal sliding

mode controller, and the parameters of the adaptive RCMAC are on-line tuned by the derived adaptive law from a Lyapunov function. Based on the H^∞ control approach, the robust controller is employed to efficiently suppress the influence of residual approximation error between the ideal sliding mode controller and adaptive RCMAC, so that the robust tracking performance of the system can be guaranteed. Finally, the simulation results of two multivariable nonlinear systems have demonstrated the effectiveness of the proposed control scheme.

References

- [1] J. J. E. Slotine and W. P. Li, *Applied Nonlinear Control*. Englewood Cliffs, NJ: Prentice-Hall, 1991
- [2] C. M. Lin and Y. J. Mon, "Decoupling Control by Hierarchical Fuzzy Sliding-Mode Controller," *IEEE Trans. Control Systems Technology*, Vol. 13, No. 4, pp. 593-598, 2005
- [3] C. M. Lin and C. F. Hsu, "Supervisory Recurrent Fuzzy Neural Network Control of Wing Rock for Slender Delta Wings," *IEEE Trans. Fuzzy Systems*, Vol. 12, No. 5, pp. 733-742, 2004
- [4] J. H. Park, S. H. Huh, S. H. Kim, S. J. Seo, and G. T. Park, "Direct Adaptive Controller for Nonaffine Nonlinear Systems using Self-Structuring Neural Networks," *IEEE Trans. Neural Networks*, Vol. 16, No. 2, pp. 414-422, 2005
- [5] C. F. Hsu, C. M. Lin and R. G. Yeh, "Supervisory Adaptive Dynamic RBF-based Neural-Fuzzy Control System Design for Unknown Nonlinear Systems," *Applied Soft Computing*, Vol. 13, No. 4, pp. 1620-1626, 2013
- [6] C. M. Lin, A. B. Ting, C. F. Hsu and C. M. Chung, "Adaptive Control for MIMO Uncertain Nonlinear Systems using Recurrent Wavelet Neural Network," *International Journal of Neural Systems*, Vol. 22, No. 1, pp.37-50, 2012
- [7] C. M. Lin and C. F. Hsu, "Neural-Network Hybrid Control for Antilock Braking Systems," *IEEE Trans. Neural Networks*, Vol. 14, No. 2, pp. 351-359, 2003
- [8] C. H. Tsai, H. Y. Chung, and F. M. Yu, "Neuro-Sliding Mode Control with its Applications to Seesaw Systems," *IEEE Trans. Neural Networks*, Vol. 15, No. 1, pp. 124-134, 2004
- [9] F. Da, "Decentralized Sliding Mode Adaptive Controller Design Based on Fuzzy Neural Networks for Interconnected Uncertain Nonlinear Systems," *IEEE Trans. Neural Networks*, Vol. 11, No. 6, pp. 1471-1480, 2000
- [10] J. S. Albus, "A New Approach to Manipulator Control: the Cerebellar Model Articulation Controller (CMAC)," *J. Dyn. Syst. Meas. Contr.*, Vol. 97, No. 3, pp. 220-227, 1975

-
- [11] F. J. Gonzalez-Serrano, A. R. Figueiras-Vidal and A. Artes-Rodriguez, "Generalizing CMAC Architecture and Training," *IEEE Trans. Neural Networks*, Vol. 9, No. 6, pp. 1509-1514, 1998
- [12] J. C. Jan and S. L. Hung, "High-Order MS_CMAC Neural Network," *IEEE Trans. Neural Networks*, Vol. 12, No. 3, pp. 598-603, 2001
- [13] C. M. Lin and Y. F. Peng, "Adaptive CMAC-based Supervisory Control for Uncertain Nonlinear Systems," *IEEE Trans. Systems, Man, and Cybernetics, Part B*, Vol. 34, No. 2, pp. 1248-1260, 2004
- [14] C. M. Lin and H. Y. Li, "A Novel Adaptive Wavelet Fuzzy Cerebellar Model Articulation Control System Design for Voice Coil Motors," *IEEE Trans. Industrial Electronics*, Vol. 59, No. 4, pp. 2024-2033, 2012
- [15] T. F. Wu, P. S. Tsai, F. R. Chang, and L. S. Wang, "Adaptive Fuzzy CMAC Control for a Class of Nonlinear Systems with Smooth Compensation," *IEE Proc.,-Control Theory and Applications*, Vol. 153, No. 6, pp. 647-657, 2006
- [16] C. H. Lee, F. Y. Chang and C. M. Lin, "An Efficient Interval Type-2 Fuzzy CMAC for Chaos Time-Series Prediction and Synchronization," *IEEE Trans. on Cybernetics*, Vol. 44, No. 3, pp. 329-341, 2014
- [17] C. M. Lin and H. Y. Li "Dynamic Petri Fuzzy Cerebellar Model Articulation Control System Design for Magnetic Levitation System," *IEEE Trans. Control Systems Tecchnology*, Vol. 23, No. 2, pp. 693-699, 2015
- [18] B. S. Chen, C. H. Lee, Y. C. Chang, " H_∞ Tracking Design of Uncertain Nonlinear SISO Systems: Adaptive Fuzzy Approach," *IEEE Trans. Fuzzy Systems*, Vol. 4, No. 1, pp. 32-43, 1996
- [19] F. J. Lin and P. H. Shen, "Adaptive Fuzzy-Neural-Network Control for a DSP-based Permanent Magnet Linear Synchronous Motor Servo Drive," *IEEE Transactions on Fuzzy Systems*, Vol. 14, No. 4, pp. 481-495, 2006
- [20] Y. C. Chang, "Robust H_∞ Control for a Class of Uncertain Nonlinear Time-Varying Systems and Its Application," *IEE Proc.,-Control Theory and Applications*, Vol. 151, No. 5, pp. 601-609, 2004
- [21] E. Kim, "Output Feedback Tracking Control of Robot Manipulators with Model Uncertainty via Adaptive Fuzzy Logic," *IEEE Trans. Fuzzy Systems*, Vol. 12, No. 3, pp. 368-378, 2004