DOI:
10.1109/TSMC.2023.3311540

Document Version
Peer reviewed version

Citation for published version (APA):
Zhong, Z., Lam, H-K., Ying, H., & Xu, G. (2023). CMAC-based SMC for uncertain descriptor systems using reachable set learning. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 1-11. https://doi.org/10.1109/TSMC.2023.3311540

CMAC-Based SMC for Uncertain Descriptor Systems Using Reachable Set Learning

Zhixiong Zhong, Hak-Keung Lam, *Fellow, IEEE*, Hao Ying, *Fellow, IEEE*, and Ge Xu

***Abstract*—This paper introduces a novel sliding mode control (SMC) law to achieve trajectory tracking for a class of descriptor systems with unknown uncertainties. It approximates the uncertainties by a cerebellar model articulation control (CMAC) neural network. We formulate the problem of training the CMAC as a scheme of estimating a reachable set for a discrete-time nonlinear system. A new online learning algorithm based on output feedback control of reachable set estimation is developed and the approximation error is bounded in an ellipsoidal reachable set. In order to dispel the effect of the approximation error of the CMAC, we develop a compensation controller by using the reachable set bounds. Controller gains and parameters of the learning algorithm are obtained via linear matrix inequalities (LMIs). Our computer simulation results show that the proposed CMAC-based SMC technique can achieve convergent tracking errors. The technique is applied to a salient permanent magnet synchronous motor (PMSM) in our lab and demonstrates excellent performance.**

Keywords: Uncertain descriptor systems, sliding mode control (SMC), cerebellar model articulation control (CMAC), reachable set estimation.

I. Introduction

Over the past few decades, sliding mode control (SMC) has been regarded as an efficient robust control strategy for various classes of uncertain systems, such as uncertain stochastic systems [1], [2], uncertain linear systems [3], uncertain nonlinear systems [4], [5]. The essence of SMC is to always drive control system states toward a given sliding mode surface [6], [7]. However, when uncertain systems exist unknown dynamics, the sliding mode motion can work well only if the information of the unknown dynamic is available. Thus the robust use of SMC to stability analysis of unknown dynamic systems is difficult to be implemented [8]–[10]. Neural networks (NNs) are constructed from a plentiful parallel structure, which makes them capable of approximating a nonlinear function for an arbitrary precision. The utilization of NNs for stability analysis of unknown dynamic systems has become widespread and effective [11]–[13]. However, in each learning cycle all the weights of NNs need to keep updating, and the essence of such learning is global and time-consuming. Thus, the efficacy of the multilayer NNs is generally limited when considering real-time control issues [14]–[17].

Thanks to the introduction of CMAC model, the fast convergence speed and good generalization capability in the identification and control of complex dynamical systems can be obtained [18]–[20]. The essence of CMAC is constructed from an association-memory network with nonfully connected forms and lapped receptive fields. It has been demonstrated that CMAC can be used for approximating a nonlinear function to any given precision [21]. In traditional CMAC, its receptive field space applies the constant binary function or the triangular basis one in the sense that their derivative characteristics can not be collected. The work of [22] has introduced a new CMAC neural network using a Gaussian basis function with the differentiable characteristic in its receptive field space, where the derivative information is acquired from the input-output relations, and the convergence analysis for the proposed new CMAC network model is performed. Another important topic in training of a CMAC neural network has been presented in the open literature. The gradient descent algorithms such as back propagation (BP), search the parameter weights of the network model toward the steepest descent direction for minimizing the approximation error [23]–[25]. It has been regarded as a basic method for training CMAC model in control system applications. However, the main drawbacks are its slow convergence speed and incapable of obtaining the global minimum [26].

Descriptor systems, are also named as singular systems. They can be utilized to represent these systems, which are difficult to be represented by normal models [27]. Descriptor systems have attracted considerable interests in the literature for a variety of practical applications such as economics, robotics, electrical and chemical systems [28], [29]. Meanwhile, in practical control systems always exist unknown uncertainties, which include unmodeled characteristics, modeling errors and unknown disturbances. These conditions might deteriorate the performance of control systems and even lead to instability.

To the best of our knowledge, there is little literature on the SMC design of trajectory tracking for descriptor systems with unknown uncertainties. This motivates our present research. This paper introduces a novel SMC strategy of trajectory tracking for a class of descriptor systems with unknown uncertainties. First, a CMAC neural network is employed to achieve the approximation of the unknown uncertainties in the considered system, and is embedded into the SMC controller. Then, the training problem of the CMAC neural network

The work described in this paper was supported by the Central Government Drects Special Funds for Scientific and Technological Development , China under Grant 2019L3009.

Zhixiong Zhong and Ge Xu are with the Key Laboratory of Information Processing and Intelligent Control, Minjiang University, Fuzhou 350121, China. Email:zhixiongzhong2012@126.com; xuge@pku.edu.cn

Hak-Keung Lam is with the Department of Engineering, King's College London, Strand, London, WC2R 2LS, United Kingdom. Email: hak-keung.lam@kcl.ac.uk

Hao Ying is with the Department of Electrical and Computer Engineering, Wayne State University, Detroit, MI 48202, USA. Email: hao.ying@wayne.edu

model is cast into the estimation scheme of reachable set for a discrete-time nonlinear system. A new online learning algorithm based on the output-feedback control method is developed and the approximation error is bounded in an ellipsoidal reachable set. Moreover, based on the obtained boundary of reachable set, a compensation controller is utilized to remove the negative impact induced by the approximated error. After the help of the Lyapunov theory, the convergence of both the closed-loop SMC system and the training CMAC neural networks can be guaranteed. It will be shown that the controller gains and the online learning parameters are solved by using the LMI optimization techniques. Finally, a simulation application of salient PMSM demonstrates the remarkable effectiveness and superiority.

The main contributions to the CMAC-based SMC scheme proposed in this paper are summarized as below:

i) This paper examines the issue of trajectory tracking for a class of descriptor systems with unknown uncertainties. We propose a CMAC-based SMC scheme that combines the merits of CMAC and SMC. In this scheme, the CMAC model is utilized to achieve the approximation of the unknown uncertainties and the SMC strategy guarantees the reachability of the trajectory error subject to the given sliding mode surface. It achieves not only the accurate approximation of uncertain dynamics by using CMAC model but also preserves the advantages of rapid response and robustness characteristic of the SMC technique.

ii) A new online learning law that is based on the reachable set estimation of output-feedback control method is proposed for training CMAC model. Our approach differs from previous approaches, we do not focus on the widely-used gradient descent algorithm learning framework with slow convergence speed or non-global minimum. In the online learning settings, the training problem of CMAC model is cast into the estimation framework of the reachable set for discrete-time nonlinear system, and the approximation error is bounded in an ellipsoidal reachable set.

iii) The proposed CMAC-based SMC scheme reformulates the SMC of trajectory tracking for uncertain descriptor system and online learning of the CMAC as a convex optimization problem readily solved by the standard LMI toolbox. A high tracking accuracy and convergence speed can be guaranteed, and the minimum approximation error of the unknown dynamics can be obtained. Compared with the work of [30]–[32], the gradient-descent based learning is used for the online learning of CMAC model that yields several learning rates are chosen a priori. However, there are no simple methods to choose the online learning rates, such that the minimum approximation error can be achieved.

The outline of this paper is arranged as below. Section II presents the problem formulation. The CMAC-based SMC is considered in Section III. The control of a salient PMSM system is used in Section IV to demonstrate the effectiveness and superiority of the proposed methods, which is summarized by some conclusions in Section V.

Notations. $\Re^{n\times m}$ denotes the real matrix with the $n\times m$ dimension and $\Re^{n}$ is the Euclidean space with the n-dimensional characteristics. A^{-1} is the inverse of the matrix A, and A^T is its transpose. $P>0$ represents that the matrix P is positive-definite, and $P\geq 0$ means that the matrix P is positive semi-definite. Sym$\{A\}$ means $A+A^T$, where $A\in\Re^{n\times n}$. The term $\star$ denotes symmetry, for example $\begin{bmatrix} A & \star \\ B & C \end{bmatrix}$, where $\star = B^T$. $\mathbf{I}_n$ is n-dimensional identity matrix and $0_{m\times n}$ is the zero matrix with $m\times n$ dimension. $\|x(t)\|_2$ denotes the Euclidean norm of the vector $x(t)$, and $\|x(t)\|_\infty$=$\sup_{t\geq 0}\{|x(t)|\}$.

II. Problem Formulation

This paper considers a class of descriptor systems with unknown uncertainties as below:

$$E\dot{x}(t) = Ax(t) + Bu(t) + N(t), \tag{1}$$

where $x(t)\in\Re^{n_x}$ and $u(t)\in\Re^{n_u}$ denote the system state and the control input, respectively. E, A and B are the known system parameters, where E may be a singular matrix with the rank $(E) = n_r \leq n_x$. $N(t)\in\Re^{n_x}$ is the total unknown uncertainty, which includes unmodeled characteristics, modeled errors and unknown disturbances.

This paper aims at designing a tracking control strategy such that the system state $x(t)$ tracks a given trajectory signal $x_d(t)$. To do so, we first denote the tracking error as the following form:

$$e(t) = x_d(t) - x(t). \tag{2}$$

Based on the representation of the tracking error (2), we introduce an integral-type sliding surface function as below [33]:

$$s(t) = GEe(t) - \int_0^t K_e e(s)ds, \tag{3}$$

where $G\in\Re^{n_u\times n_x}$ is a given matrix, which can guarantee that the matrix GB is nonsingular. $K_e\in\Re^{n_u\times n_x}$ is a parameterized matrix, which will be designed later.

If the unknown uncertainty $N(t)$ can be accurately known, then a perfect controller is expressed as

$$u^*(t) = (GB)^{-1}\left[GE\dot{x}_d(t) - GAx(t) - GN(t) - K_e e(t)\right], \tag{4}$$

where the ideal controller $u^*(t)$ ensures $\dot{s}(t)=0$.

Note that in practical applications the uncertainty $N(t)$ is generally unknown, thus the ideal controller in (4) is unavailable. In this paper the unknown uncertainty $N(t)$ will be approximated by employing a CMAC neural network model, and a CMAC-based SMC law will be introduced in the following sections, which raise the following two key questions:

Q-1. How to design a CMAC approximator with online learning algorithm such that the convergence of approximation error is guaranteed and the approximation error is bounded in an ellipsoidal reachable set?

Q-2. How to give a design result of the CMAC-based SMC strategy such that the tracking trajectories are driven onto the predefined sliding mode surface $s(t)=0$?

Remark 2.1. It is noted that, in each learning cycle of most NNs all the weights should be updated, thus the essence of such kind of learning is both global and slow. CMAC is constructed from an association-memory network with the nonfully connected form and the lapped receptive space. It has been shown that CMAC is able to achieve fast learning, such that it is suitable for implementing the real-time control [12], [18].

Remark 2.2. Note that, it is difficult to evaluate the training performance of the CMAC neural network model by using the gradient-descent learning. This paper casts the CMAC's training into the estimation framework of the reachable set for discrete-time nonlinear system. In this case, an online learning algorithm based on output-feedback control method is used instead of the gradient-descent method for the CMAC's training. Furthermore, the convergence of CMAC neural network model is established and its approximation error can be bounded in an ellipsoidal reachable set.

III. CMAC-Based Sliding-Mode Control

This section first introduces a main controller in which a CMAC model is used to achieve the approximation of the unknown lumped uncertainty. Then the training problem of the CMAC model is formulated into the framework of the reachable set estimation for discrete-time nonlinear system, and a new online learning algorithm based on output-feedback control method is developed. Finally, a compensation controller is introduced to offset the gap between the unknown lumped uncertainty and its approximation.

A. Framework of CMAC-based SMC

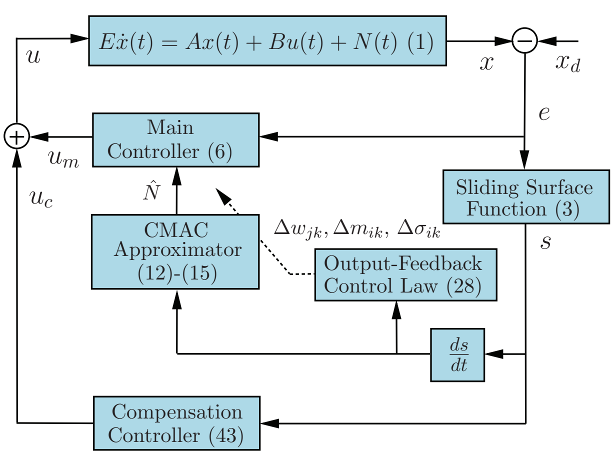

Fig. 1. Framework of the CMAC-based SMC strategy.

Fig.1 depicts the framework of the CMAC-based SMC strategy, which consists of the main controller and the compensation one as follows:

$$u(t) = u_m(t) + u_c(t), \tag{5}$$

where $u_m(t)$ is the main controller, which is used instead of the perfect controller (4). In the main controller, a CMAC model integrated into SMC strategy is introduced to achieve the approximation of the unknown lumped uncertainty. The online learning of the CMAC model is facilitated by using an output-feedback-based control law, which ensures that the convergence of CMAC model is established and its approximation error can be bounded in an ellipsoidal reachable set; The compensation controller $u_c(t)$ is utilized to offset the gap between the unknown lumped uncertainty and its approximation, which helps the main controller drive the tracking error trajectories of the uncertain descriptor system onto the given sliding mode surface.

B. Main controller

By introducing the CMAC-based function approximator, we give the following main controller instead of the perfect controller (4):

$$u_m(t) = (GB)^{-1}\left[GE\dot{x}_d(t) - GAx(t) - G\hat{N}(t) - K_e e(t)\right], \tag{6}$$

where $\hat{N}(t)$ is the approximation of $N(t)$.

Now, by submitting the main controller (6) into the system (1), and taking the derivative of the tracking error, the closed-loop tracking error dynamics is described as

$$E\dot{e}(t) = \left(\bar{A} + \bar{B}K_e\right)e(t) + \bar{W}(t), \tag{7}$$

where

$$\begin{aligned} &\bar{A} = A - \bar{B}GA, \bar{B} = B\,(GB)^{-1}, \\ &\bar{W}(t) = \bar{B}G\hat{N}(t) - N(t) + \left(E - \bar{B}GE\right)\dot{x}_d(t) - \bar{A}x_d(t). \end{aligned} \tag{8}$$

Based on the resulting closed-loop tracking control system (7), the following result is devoted to solve the design of the matrix parameter K_e.

Lemma 1: The tracking error dynamics (7) with the parameter matrices $\{E, \bar{A}, \bar{B}\}$ is robust stable with the $\mathcal{H}_\infty$ performance index $\|e(t)\|_2 \leq \gamma \left\|\bar{W}(t)\right\|_2$ if there exist the matrix $0 < X = X^T \in \Re^{n_x \times n_x}$, and the matrix $\bar{K}_e \in \Re^{n_u \times n_x}$, such that is minimized γ subject to the following matrix inequalities:

$$X^T E^T = EX \geq 0, \tag{9}$$

$$\begin{bmatrix} \mathrm{Sym}\left(\bar{A}X + \bar{B}\bar{K}_e\right) & X^T & \mathbf{I} \\ \star & -\mathbf{I} & 0 \\ \star & 0 & -\gamma^2\mathbf{I} \end{bmatrix} < 0. \tag{10}$$

Moreover, if the above matrix inequalities have a feasible solution then the gain matrix K_e in (6) can be given by

$$K_e = \bar{K}_e X^{-1}. \tag{11}$$

Proof: The result can be obtained straightly from Lemma 2 of [34], whose proof is thus deleted. ∎

Remark 3.1. Note that, the matrix inequality in (9) is a positive semi-definite form and the solution used standard LMI technique becomes a difficult task. For a singular matrix E_0, where rank $(E_0) = n_r \leq n_x$, we can specify the matrices M and N with the nonsingular characteristic to achieve $E_0 = M \begin{bmatrix} \mathbf{I} & 0 \\ 0 & 0 \end{bmatrix} N$. Without loss of generality, we consider a special case with the matrix $E = \begin{bmatrix} \mathbf{I} & 0 \\ 0 & 0 \end{bmatrix}$, and then the matrix X is denoted as $X = \begin{bmatrix} X_1 & 0 \\ X_2 & X_3 \end{bmatrix}$, where $0 < X_1 = X_1^T \in \Re^{n_r \times n_r}, X_2 \in \Re^{(n_x - n_r) \times n_r}, X_3 \in \Re^{(n_x - n_r) \times (n_x - n_r)}$. In this case, the semi-definite matrix inequality in (9) holds, and the result in Lemma 1 is formulated into LMIs [35].

C. CMAC model

The proposed CMAC model includes three mappings and one output. All functional mappings of the CMAC model are shown as follows [12]:

$$\text{Mapping: } I \to A, \tag{12}$$
$$\text{Mapping: } A \to R, \tag{13}$$
$$\text{Mapping: } R \to W, \tag{14}$$
$$\text{Output computation: } O(I) = W^T \Gamma(I), \tag{15}$$

where I, A, R and W denote the input, the associative memory, the receptive field, and the weighted memory spaces, respectively; O is the output. In every mapping space the corresponding basic function is introduced as follows:

1) Mapping $I \to A$: In the input space, $I = [I_1, I_2, \cdots, I_{n_i}]$ denotes a n_i-dimensional input variable with $i \in \mathcal{I} := \{1, 2, \ldots, n_i\}$. Each input datum is partitioned to n_k neurons by virtue of Gaussian basis function, and these neurons are placed in the associative memory space A with the set $k \in \mathcal{K} := \{1, 2, \ldots, n_k\}$. Considering the descriptions above, each neuron can be given by

$$\alpha_{ik}(I_i) = \exp\left[-\frac{(I_i - m_{ik})^2}{\sigma_{ik}^2}\right], i \in \mathcal{I}, k \in \mathcal{K} \tag{16}$$

where m_{ik} and σ_{ik} denote the mean and the variance relative to the input variable, respectively.

2) Mapping $A \to R$: The receptive field space R has a k-layer region, which is called hypercube represented by

$$\beta_k(I) = \prod_{i=1}^{n_i} \alpha_{ik}(I_i), k \in \mathcal{K} \tag{17}$$

where $\beta_k(I)$ denotes the general basis function for representing the hypercube.

3) Mapping $R \to W$: Each region in the receptive field space R is connected to the weighted memory W with the corresponding specific value, that is

$$\theta_{jk}(I) = w_{jk} \beta_k(I), j \in \mathcal{M}, k \in \mathcal{K} \tag{18}$$

where w_{jk} denotes the weight relative to the k-th hypercube in the j-th output, $j \in \mathcal{M} := \{1, 2, \ldots, n_m\}$.

4) Output: The output of CMAC includes n_m subspaces, and each subspace is the sum of n_k particular adjustable parameters, which is given by

$$O_j(I) = \sum_{k=1}^{n_k} \theta_{jk}(I), j \in \mathcal{M}. \tag{19}$$

Thus, the output of CMAC is calculated by

$$O(I) = \sum_{j=1}^{n_m} O_j(I). \tag{20}$$

Remark 3.2. Note that, the receptive field space of the traditional CMAC applies the constant binary function or the triangular basis one in the sense that their derivative characteristics can not be collected. Therefore, this paper proposes the CMAC used the Gaussian basis function as shown in (12)-(20), which overcomes the drawback of using the constant binary function or the triangular basis one [36].

D. Formulation of CMAC training

In the previous subsection, we have introduced CMAC model. The following subsections will focus on training C-MAC model. First, it follows the descriptions from (12)-(20) showing that the input-output relationship of the CMAC's dynamics can be represented as

$$\begin{aligned} O(t) &= W^T \Gamma(I(t)) \\ &= f(w_{jk}(t), m_{ik}(t), \sigma_{ik}(t), I(t)). \end{aligned} \tag{21}$$

Now, we regard $\dot{s}(t)$ and $\hat{N}(t)$ as the input and output of CMAC's dynamics (21), respectively. Thus, we have

$$\begin{aligned} \hat{N}(t) &= W^T \Gamma(\dot{s}(t)) \\ &= f(w_{jk}(t), m_{ik}(t), \sigma_{ik}(t), \dot{s}(t)). \end{aligned} \tag{22}$$

Using the first-order Taylor approximation [37], it can be shown that

$$\begin{aligned} \hat{N}(t+1) = \hat{N}(t) &+ \frac{d\hat{N}(t)}{dw_{jk}} \Delta w_{jk}(t) \\ &+ \frac{d\hat{N}(t)}{dm_{ik}} \Delta m_{ik}(t) + \frac{d\hat{N}(t)}{d\sigma_{ik}} \Delta \sigma_{ik}(t) \\ &+ \frac{d\hat{N}(t)}{d\dot{s}(t)} \Delta \dot{s}(t) + \phi(t), \end{aligned} \tag{23}$$

where $\frac{d\hat{N}(t)}{dw_{jk}}, \frac{d\hat{N}(t)}{dm_{ik}}, \frac{d\hat{N}(t)}{d\sigma_{ik}}, \frac{d\hat{N}(t)}{d\dot{s}(t)}$ are the partial derivatives with respect to $w_{jk}(t)$, $m_{ik}(t)$, $\sigma_{ik}(t)$, $\dot{s}(t)$, respectively; $\Delta w_{jk}(t), \Delta m_{ik}(t), \Delta \sigma_{ik}(t), \Delta \dot{s}(t)$ are the difference terms, and $\phi(t)$ is the residual signal.

Then, by recalling the sliding surface function $s(t)$ in (3), and taking its derivative, and using the relations of (1), (2) and (6), we have

$$\dot{s}(t) = G\varepsilon(t), \tag{24}$$

where $\dot{s}(t)$ is the derivative information of the sliding surface function, and $\varepsilon(t)$ is the approximation error subject to $\varepsilon(t) = \hat{N}(t) - N(t)$.

We further subtract $N(t+1)$ to the both sides of (23), then the following output-feedback control system can be obtained as:

$$\varepsilon(t+1) = \varepsilon(t) + \mathcal{B}(t)\mathcal{U}(t) + \mathcal{W}(t), \tag{25}$$
$$\mathcal{Y}(t) = G\varepsilon(t), \tag{26}$$

where

$$\begin{aligned} &\mathcal{B}(t) = \left[\begin{array}{ccc} \frac{d\hat{N}(t)}{dw_{jk}} & \frac{d\hat{N}(t)}{dm_{ik}} & \frac{d\hat{N}(t)}{d\sigma_{ik}} \end{array}\right], \\ &\mathcal{U}(t) = \left[\begin{array}{ccc} \Delta w_{jk}^T(t) & \Delta m_{ik}^T(t) & \Delta \sigma_{ik}^T(t) \end{array}\right]^T, \\ &\mathcal{W}(t) = \frac{d\hat{N}(t)}{d[G\varepsilon(t)]} G\Delta\varepsilon(t) + \phi(t) - \Delta N(t), \\ &\Delta\varepsilon(t) = \varepsilon(t+1) - \varepsilon(t), \Delta N(t) = N(t+1) - N(t), \end{aligned} \tag{27}$$

with the following partial derivatives:

$$\begin{aligned} \frac{d\hat{N}(t)}{dw_{jk}} &= \prod_{i=1}^{n_i} \beta_{ik}(\dot{s}_i), \beta_{ik}(\dot{s}_i) = \exp\left[-\frac{(\dot{s}_i - m_{ik})^2}{\sigma_{ik}^2}\right], \\ \frac{d\hat{N}(t)}{dm_{ik}} &= \sum_{j=1}^{n_m} \sum_{k=1}^{n_k} w_{jk} \prod_{i=1}^{n_i} \left[\beta_{ik}(\dot{s}_i) \frac{2(\dot{s}_i - m_{ik})}{\sigma_{ik}^2}\right], \\ \frac{d\hat{N}(t)}{d\sigma_{ik}} &= \sum_{j=1}^{n_m} \sum_{k=1}^{n_k} w_{jk} \prod_{i=1}^{n_i} \left[\beta_{ik}(\dot{s}_i) \frac{2(\dot{s}_i - m_{ik})^2}{\sigma_{ik}^3}\right]. \end{aligned} \tag{28}$$

Remark 3.3. It is noted that when the approximation error $\varepsilon(t)$ tends to zero then the weights $\{w_{jk}(t), m_{ik}(t), \sigma_{ik}(t)\}$ are around the real values. Therefore, in this case, the difference terms $\{\Delta w_{jk}(t), \Delta m_{ik}(t), \Delta\sigma_{ik}(t), \Delta\dot{s}(t)\}$ are around zero, which means the residual signal $\phi(t)$ will be around zero.

Remark 3.4. Note that when using the CMAC model, a general question is raised: How to train the CMAC model? In the other words, the task is to explore a weight update law for refreshing the hypercube weight w_{jk}, and the mean m_{ik}, and the variance σ_{ik} in the CMAC model.

Remark 3.5. Note that the learning problem of CMAC's weights has been formulated into a robust output-feedback control framework of the discrete-time nonlinear system as shown in (25) and (26). More specifically, the approximation error of CMAC model is denoted as the system state $\varepsilon(t)$, and the learning law of CMAC's weights is considered as the control input $\mathcal{U}(t)$, which decides the refresh rates of weight values, and the term $\mathcal{W}(t)$ is regarded as the disturbance.

Remark 3.6. For the above-mentioned robust control problem, we aim at designing an output-feedback controller with reachable set estimation, which ensures the CMAC model is robust convergence and its approximation error is bounded in an ellipsoidal reachable set.

E. Online learning law based on output-feedback control

The previous subsection has formulated the training problem of CMAC model into a robust control framework of discrete-time nonlinear system as shown in (25) and (26). It is natural to specify the online learning law of CMAC model as the following output-feedback controller:

$$\mathcal{U}(t) = \mathcal{K}_y \mathcal{Y}(t), \tag{29}$$

where $\mathcal{K}_y$ is the controller gain, which will be designed later.

Now, by submitting the controller (29) into the system (25) and (26), and taking the relation of (24), the closed-loop output-feedback control system is given by

$$\varepsilon(t+1) = \left(\mathbf{I} + \bar{\mathcal{K}}_y\right)\varepsilon(t) + \mathcal{W}(t), \tag{30}$$

where $\bar{\mathcal{K}}_y = \mathcal{B}(t)\mathcal{K}_y G$ and $\mathcal{W}(t)$ is further assumed to be satisfied with the following condition:

$$\mathcal{W}^T(t)\mathcal{W}(t) \le \bar{\mathcal{W}}^2, \tag{31}$$

where $\bar{\mathcal{W}}$ is a constant.

Here, our aim at designing the online learning law based on the output-feedback controller as shown in (29) such that the approximated error dynamics of CMAC in (30) is with the following reachable set bounding:

$$\begin{aligned} \mathbb{S} \triangleq \{&\varepsilon(t) \in \Re^{n_x} \,|\varepsilon(t) \text{ and } \mathcal{W}(t) \text{ are subject to} \\ &(30) \text{ and } (31), \text{respectively}, t \ge 0\}. \end{aligned} \tag{32}$$

The reachable set of the closed-loop output-feedback control system in (30) subject to an ellipsoid is given by [38]

$$\mathbb{E} \triangleq \left\{\varepsilon(t) \mid \varepsilon^T(t) P\varepsilon(t) < 1,\ \varepsilon(t) \in \Re^{n_x}\right\}, \tag{33}$$

where $P = P^T > 0$.

Note that, the learning problem of CMAC's weights has been formulated into a reachable set estimation framework of the discrete-time nonlinear output-feedback control system as shown in (32). In the following, we first present a formulated analysis result for the estimation problem of the reachable set. This is derived to answer the first question (**Q-1**).

Theorem 1: An online learning law (29) can guarantee the convergence of CMAC's approximation error with the reachable set in (32), if there exist the matrices $\{0 < P = P^T \in \Re^{n_x \times n_x}, G \in \Re^{n_u \times n_x}, \mathcal{K}_y \in \Re^{n_u \times n_u}\}$ and the positive scalars $\{0 < a < 1, \bar{\mathcal{W}}\}$, such that the following matrix inequality holds,

$$\begin{bmatrix} -aP & 0 & P + G^T\mathcal{K}_y^T\mathcal{B}^T(t)P \\ 0 & -\frac{1-a}{\bar{\mathcal{W}}^2}\mathbf{I} & P \\ \star & \star & -P \end{bmatrix} < 0. \tag{34}$$

Proof: We firstly define $V(t) = \varepsilon^T(t)P\varepsilon(t)$ and $\Delta V(t) = V(t+1) - V(t)$, where $0 < P = P^T \in \Re^{n_x \times n_x}$. Further, the performance index is introduced as below:

$$\begin{aligned} J(t) &= \Delta V(t) + (1-a)V(t) - \frac{1-a}{\bar{\mathcal{W}}^2}\mathcal{W}^T(t)\mathcal{W}(t) \\ &= \varepsilon^T(t+1)P\varepsilon(t+1) - \varepsilon^T(t)P\varepsilon(t) \\ &\quad + (1-a)\varepsilon^T(t)P\varepsilon(t) - \frac{1-a}{\bar{\mathcal{W}}^2}\mathcal{W}^T(t)\mathcal{W}(t) \\ &= \varepsilon^T(t+1)P\varepsilon(t+1) - a\varepsilon^T(t)P\varepsilon(t) \\ &\quad - \frac{1-a}{\bar{\mathcal{W}}^2}\mathcal{W}^T(t)\mathcal{W}(t) \\ &= \left[\left(\mathbf{I} + \bar{\mathcal{K}}_y\right)\varepsilon(t) + \mathcal{W}(t)\right]^T P[\star] \\ &\quad - a\varepsilon^T(t)P\varepsilon(t) - \frac{1-a}{\bar{\mathcal{W}}^2}\mathcal{W}^T(t)\mathcal{W}(t) \\ &= \begin{bmatrix} \varepsilon(t) \\ \mathcal{W}(t) \end{bmatrix}^T \begin{bmatrix} \left(\mathbf{I} + \bar{\mathcal{K}}_y\right)^T \\ \mathbf{I} \end{bmatrix} P[\star] \begin{bmatrix} \varepsilon(t) \\ \mathcal{W}(t) \end{bmatrix} \\ &\quad + \begin{bmatrix} \varepsilon(t) \\ \mathcal{W}(t) \end{bmatrix}^T \begin{bmatrix} -aP & 0 \\ 0 & -\frac{1-a}{\bar{\mathcal{W}}^2}\mathbf{I} \end{bmatrix} \begin{bmatrix} \varepsilon(t) \\ \mathcal{W}(t) \end{bmatrix}, \end{aligned} \tag{35}$$

where $\bar{\mathcal{K}}_y = \mathcal{B}(t)\mathcal{K}_y G$, and $0 < a < 1$ is a constant scalar.

It can be seen from (35) that the following matrix inequality holds

$$\begin{bmatrix} \left(\mathbf{I} + \bar{\mathcal{K}}_y\right)^T \\ \mathbf{I} \end{bmatrix} P[\star] + \begin{bmatrix} -aP & 0 \\ 0 & -\frac{1-a}{\bar{\mathcal{W}}^2}\mathbf{I} \end{bmatrix} < 0, \tag{36}$$

which guarantees $J(t) < 0$. By applying Schur complement lemma [39], the result on (34) can be obtained directly from (36). Thus we complete the verification of the robust stability for the closed-loop control system model in (30).

Since $J(t) < 0$, we have

$$V(t) < 1 + (V(0) - 1)a^k. \tag{37}$$

Note that when $\varepsilon(0)$ satisfies $V(0) \le 1$, the inequality in (37) implies $V(t) < 1$. Thus we complete this proof to the reachable set in (32). ■

It is also noted that the matrix inequality in (34) is nonlinear because of the coupling term $\mathcal{B}(t)\mathcal{K}_y GP$. For simplicity in the controller design procedure, we can specify the controller gain $\mathcal{K}_y$ as

$$\mathcal{K}_y = \mathcal{B}^T(t)\left(\mathcal{B}(t)\mathcal{B}^T(t)\right)^{-1}K_y, \tag{38}$$

where K_y is a constant controller gain. In this case, the output-feedback control system in (30) is rewritten as

$$\varepsilon(t+1) = (\mathbf{I} + K_y G)\varepsilon(t) + \mathcal{W}(t). \tag{39}$$

Based on the new closed-loop error dynamics in (39), an online learning law for updating the CMAC's weights is given by a parameterized representation of the output-feedback controller gains in terms of the feasible solutions to the following LMI.

Lemma 2: An online learning law (29) can guarantee the convergence of CMAC's approximation error with the reachable set in (32), if there exist the matrices $\{0 < P = P^T \in \Re^{n_x \times n_x},\ G \in \Re^{n_u \times n_x},\ \bar{K}_y \in \Re^{n_x \times n_u}\}$, and the positive scalars $\{0 < a < 1,\ \bar{\mathcal{W}}\}$, such that the following LMI holds,

$$\begin{bmatrix} -aP & 0 & P + G^T \bar{K}_y^T \\ 0 & -\frac{1-a}{\bar{\mathcal{W}}^2}\mathbf{I} & P \\ \star & \star & -P \end{bmatrix} < 0. \tag{40}$$

Moreover, if the above LMI has a feasible solution then the matrix parameter K_y in (38) can be given by

$$K_y = P^{-1}\bar{K}_y. \tag{41}$$

Proof: The result can be obtained straightly from Theorems 1, whose proof is thus deleted. ■

Remark 3.7. Here, our aim at designing the output-feedback controller (29) such that the minimum bounding for the reachable set of CMAC's approximation error can be obtained. To do so, we borrow the work of [40] to maximize δ with the constrain $\delta\mathbf{I} < P$. By using Schur complement [39], the optimization problem can be easily solved as below:

$$\text{Minimize } \bar{\delta} \text{ subject to the constrains: } \begin{bmatrix} \bar{\delta}\mathbf{I} & \mathbf{I} \\ \star & P \end{bmatrix} \geq 0 \text{ and } (40),$$

where $\bar{\delta} = \delta^{-1}$.

Remark 3.8. It is noted that when the matrix term $\mathcal{B}(t)\mathcal{B}^T(t)$ in (38) is singular, it should be replaced by $\mathcal{B}^T(t)\left[\mathcal{B}(t)\mathcal{B}^T(t) + \rho\mathbf{I}\right]^{-1}$ with a small positive scalar ρ.

Remark 3.9. Thanks to the online learning law based on the reachable set estimation of output-feedback control as shown in (29), the learning law for updating CMAC's weights is given by

$$\Delta w_{jk}(t) = \left[\frac{d\hat{N}(t)}{dw_{jk}}\right]^T \Phi^{-1}(t) K_y \dot{s}(t),$$

$$\Delta m_{ik}(t) = \left[\frac{d\hat{N}(t)}{dm_{ik}}\right]^T \Phi^{-1}(t) K_y \dot{s}(t),$$

$$\Delta \sigma_{ik}(t) = \left[\frac{d\hat{N}(t)}{d\sigma_{ik}}\right]^T \Phi^{-1}(t) K_y \dot{s}(t),$$

where $\Phi(t) = \left[\frac{d\hat{N}(t)}{dw_{jk}}\right]\left[\frac{d\hat{N}(t)}{dw_{jk}}\right]^T + \left[\frac{d\hat{N}(t)}{dm_{ik}}\right]\left[\frac{d\hat{N}(t)}{dm_{ik}}\right]^T + \left[\frac{d\hat{N}(t)}{d\sigma_{ik}}\right]\left[\frac{d\hat{N}(t)}{d\sigma_{ik}}\right]^T$.

F. Design of compensation controller

In the previous subsections, we propose the CMAC's approximator and its learning law. We now focus on designing a compensation controller, which is used to dispel the effect of the approximation error. before moving on we recall the approximation error of the unknown lumped uncertainty in (24) as below:

$$\varepsilon(t) = \hat{N}(t) - N(t), \tag{42}$$

where $N(t)$ denotes the unknown uncertainty, and $\hat{N}(t)$ denotes the approximation of $N(t)$ by using the CMAC neural network model.

Here, our aim is to design a compensation controller, which will dispel the impact induced by the approximated error. It follows from Remark 3.6 that the approximation error can be bounded by

$$0 \leq \|G\varepsilon(t)\|_\infty \leq C, \tag{43}$$

where C is a positive constant.

Now, we introduce a compensation controller as below:

$$u_c(t) = (GB)^{-1} C \text{sgn}(s(t)), \tag{44}$$

where sgn($\star$) denotes a switching sign function.

Then, by taking the derivative of the integral-type sliding surface function (3) and submitting the main controller in (6), we have

$$\dot{s}(t) = G\varepsilon(t) - GB(GB)^{-1} C \text{sgn}(s(t)). \tag{45}$$

Based on the sliding mode dynamics (45), we derive a sufficient criteria for designing a CMAC-based SMC law as shown in (5), which drives the tracking error trajectories onto the given sliding mode surface $s(t) = 0$. This is derived to answer the second question (**Q-2**).

Theorem 2: The main controller (6) and the compensation controller (44) can ensure that the tracking error trajectories of the uncertain descriptor system (1) are driven onto the given sliding mode surface $s(t) = 0$.

Proof: We firstly consider the following Lyapunov function:

$$V(t) = \frac{1}{2}s^T(t)s(t). \tag{46}$$

By calculating the derivative of $V(t)$, and using the relation of (45), we have

$$\begin{aligned} \dot{V}(t) &= s^T(t)\dot{s}(t) \\ &= s^T(t)G\varepsilon(t) - s^T(t)C\text{sgn}(s(t)) \\ &\leq \|s(t)\|_1 \|G\varepsilon(t)\|_1 - \|s(t)\|_1 C \\ &\leq 0. \end{aligned} \tag{47}$$

It can be seen from (46) that the $\dot{V}(t) \leq 0$, which means the main controller (6) and the compensation controller (43) can drive the tracking error trajectories onto the specified sliding mode surface $s(t) = 0$. Therefore, the proof is completed. ■

G. Design procedure for CMAC-based SMC strategy

The detailed calculating steps to solve the tracking problem of the considered uncertain descriptor system can be summarized as below:

1) Use Lemma 1 to obtain the matrix gain K_e and choose the suitable matrix G, and then construct an integral-type sliding surface function in (3) for the system (1);

2) Introduce CMAC neural network model with the form of (12)-(15);

3) Construct the main controller $u_m(t)$ as shown in (6);

4) Use Lemma 2 to obtain the controller gain K_y, and construct the output-feedback control law in (29) to update the weights $\{\Delta w_{jk}(t), \Delta m_{ik}(t), \Delta\sigma_{ik}(t)\}$ of the CMAC model;

5) Use Remark 3.6 to obtain the bounding of the reachable set $\bar{\delta}$ and calculate the constant C in (43);

6) Construct the compensation controller $u_c(t)$ as shown in (44);

7) Apply the main controller and the compensation controller as shown in (6) and (44) to the system (1).

IV. Demonstrative Examples

This section considers a salient permanent magnet synchronous motor (PMSM), where its stator inductance is not equal around the airgap. In the PMSM system, its control is generally implemented by using the vector modulation. For the control of speed, torque, and position, the big trouble that arises, particularly in the regulation of torque relative to stator current. Therefore, a new transformation called the $d-q$ coordinates is used to depict the relation between the stator current and the voltage vector [41].

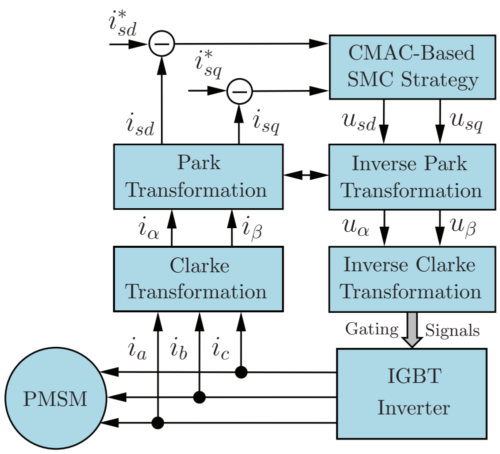

Fig. 2. Control framework of PMSM using the CMAC-based SMC method.

Fig. 2 shows the control framework of PMSM using the CMAC-based SMC method proposed in this paper. In the $d-q$ coordinates the stator inductance is equal to L_d on the d axis and is also equal to L_q on the q axis. The evolution of these $d-q$ currents can be described as follows [6], [41]:

$$L_d \frac{di_{sd}}{dt} = -R_s i_{sd} - \omega_r L_q i_{sq} + u_{sd},$$

$$L_q \frac{di_{sq}}{dt} = -R_s i_{sq} - \omega_r L_q i_{sd} + u_{sq} - \omega_r \psi_m,$$

where L_d and L_q are the stator inductance; R_s denotes the stator resistance; i_{sd} and i_{sq} are regarded as the stator current; u_{sd} and u_{sq} are the stator voltage; ω_r is the electrical rotor frequency; ψ_m is the rotor flux linkage from the permanent magnets.

Further, we consider the unknown uncertainties ϕ_1 and ϕ_2 are involved in the $d-q$ currents, respectively. The aim at controlling the stator currents i_{sd} and i_{sq} to follow the references i_{sd}^* and i_{sq}^* with the accuracy and robustness, respectively. Here, we first define

$$E = \begin{bmatrix} L_d & 0 \\ 0 & L_q \end{bmatrix}, x(t) = \begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix}, x_d(t) = \begin{bmatrix} i_{sd}^* \\ i_{sq}^* \end{bmatrix},$$

$$A = \begin{bmatrix} -R_s & -\omega_r L_q \\ -\omega_r L_q & -R_s \end{bmatrix}, B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

$$u(t) = \begin{bmatrix} u_{sd} \\ u_{sq} \end{bmatrix}, N(t) = \begin{bmatrix} \phi_1 \\ \phi_2 - \omega_r \psi_m \end{bmatrix}.$$

Then, the salient PMSM system is depicted by the state-space formulas as shown in (1).

In this simulation, the system parameters are $L_d = 0.3562$H, $L_q = 0.5298$H, $R_s = 0.2762$, $\Omega, \omega_r = 50$Hz. The detailed calculating steps to solve the current tracking problem of the considered salient PMSM system are summarized as below:

1) Use Theorem 1 to obtain the controller gain $K_e = \begin{bmatrix} -86.7994 & 16.3123 \\ -24.8099 & -131.2388 \end{bmatrix}$ and choose the matrix $G = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, and then construct an integral-type sliding surface function as the form of (3);

2) Introduce CMAC network model with the form of (12)-(15);

3) Construct the main controller $u_m(t)$ as the form of (6);

4) Give $a = 0.9$ and $\bar{\mathcal{W}} = 0.0476$ and use Lemma 1 to obtain the controller gain $K_y = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$, and construct an online learning law based on the output-feedback controller with ρ =0.00001 as follows:

$$\mathcal{U}(t) = \mathcal{B}^T(t)\left[\mathcal{B}(t)\mathcal{B}^T(t) + \rho\mathbf{I}\right]^{-1} K_y G \varepsilon(t),$$

where $\mathcal{B}(t)$ is defined in (27);

5) Use Remark 3.6 to obtain the reachable set bounding $\delta = 0.0153$ and calculate the bounding $C = 0.2010$ in (43);

6) Construct the compensation controller $u_c(t)$ as the form of (44);

7) Give the initial conditions of the control system as $x(0) = [0.5, -0.5]^T$, $x_d(t) = [0, 0.8 * \sin(5t)]^T$. Note that the choice of x_d can be different according to the torque (refer to [42] for details). We assume that the unknown uncertainties $\phi_1 = 0.2 \sin 10t$ and $\phi_2 = 0.2 \cos 10t$. Now, we apply the integral-type sliding surface function, the CMAC neural network model, the main controller, and the compensation controller. When using the proposed CMAC's learning with the output-feedback control method, the currents of the $d-q$ coordinates and their references are shown in Figs. 3 and 4, respectively. In each cycle of learning, the updated weights of CMAC neural network model can be shown in Fig. 5. Fig. 6 shows the response of the CMAC-based sliding mode control input. However, when considering the CMAC model with the gradient-descent based learning [30]–[32], Figs. 7 and 8 show the responses of the $d-$axis and $q-$axis currents and their references, respectively. Define the convergence performance as $\sum_{i=1}^{n} \|e(i)\|_2 / n$, where n is the number of iterations. Fig. 9 plots the convergence performances of CMAC for the reachable set learning law proposed in this paper and the gradient-descent based learning law proposed in [30]–[32]. It is easy to see that the high accuracy and convergence speed to the tracking responses of the $d-q$ currents can be realized by using the CMAC-based SMC method proposed in this paper.

Remark 4.1. Note that the work of [30]–[32] proposes the gradient-descent based learning for the training of C-MAC neural network without convergence analysis. The main drawbacks of the method are its slow convergence speed and its inability to ensure global minimum. Moreover, this method introduces the learning rates subject to some positive scalars to be searched or manually prescribed. Taking different learning-rate values may lead to instability of CMAC model. To effectively train the CMAC model, this paper formulates the training problem of CMAC model into the estimation framework of the reachable set for discrete time nonlinear system. An online learning law based on the output-feedback control method is developed and the approximation error of CMAC model is bounded in an ellipsoidal reachable set.

Remark 4.2. It is also noted that the proposed CMAC-

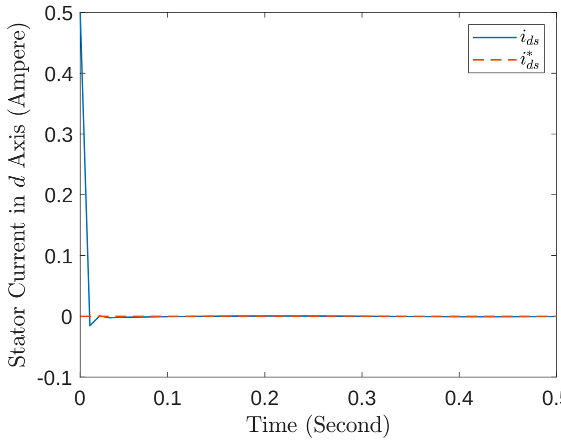

Fig. 3. Responses of the $d-$axis current using CMAC' learning based on the output-feedback control method.

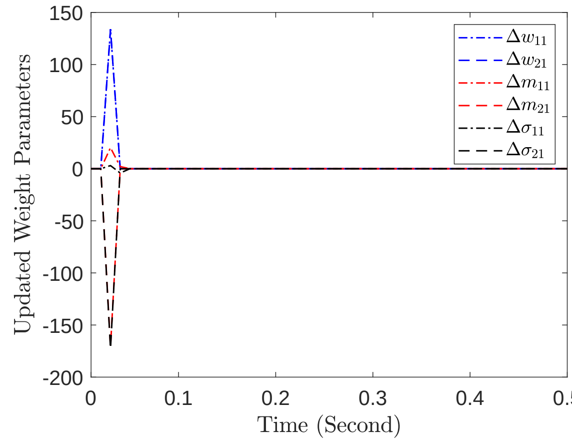

Fig. 5. The updates of CMAC weights using the online learning algorithm based on the output-feedback control method.

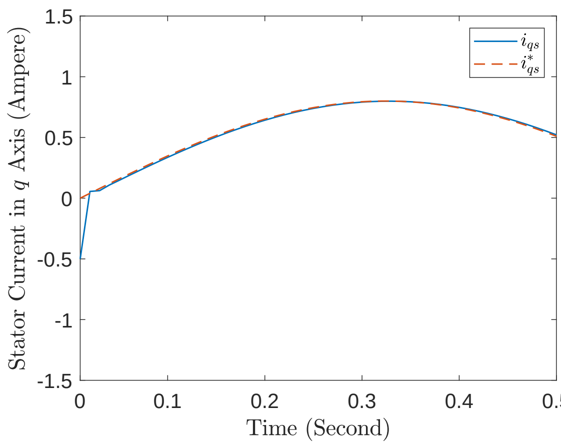

Fig. 4. Responses of the $q-$axis current using CMAC' learning based on the output-feedback control method.

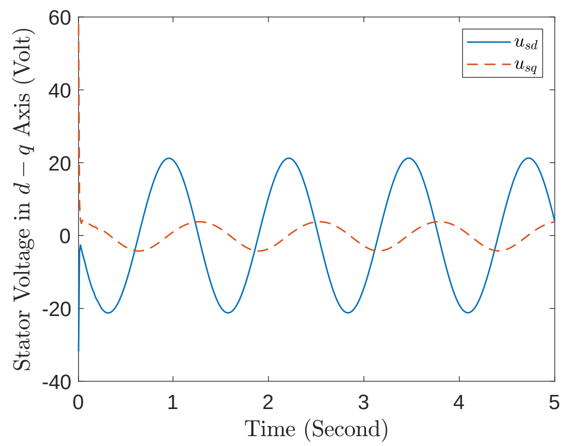

Fig. 6. Responses of the CMAC-based sliding mode control input.

based SMC carries the advantages of both the reachable set method and the SMC technique at the same time. In this sense, the training of CMAC model is robust convergence with the minimum approximation error by using the online learning law (29), and the SMC strategy ensures the reachability of the system trajectories subject to the given sliding mode surface. Figs. 3-4 have shown that the methods proposed in this paper achieve the fast response and high-accuracy tracking performance against unknown dynamics in comparison with the gradient-descent based learning proposed in [30]–[32].

V. Conclusions

The issue of SMC trajectory tracking for uncertain descriptor systems with unknown dynamics has been examined. The CMAC-based SMC scheme was developed and all the control gains and the online learning parameters are obtained by calculating a set of LMIs. The effectiveness of the proposed CMAC-based SMC scheme is illustrated by controlling a salient PMSM system. The simulation result shows that the methods proposed in this paper achieve the fast response and high-accuracy tracking performance against unknown dynamics in comparison with the gradient-descent based learning proposed in [30]–[32]. Further studies would focus on more effective neural-learning algorithms using the robust control method for identification and control of nonlinear systems.

References

[1] H. Chen, C. Lim, and P. Shi, "Robust $\mathscr{H}_\infty$-based control for uncertain stochastic fuzzy switched time-delay systems via integral sliding mode strategy," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 30, no. 2, pp. 382-396, 2022.

[2] R. Nie, S. He, F. Liu, and X. Luan, "Sliding mode controller design for conic-type nonlinear semi-Markovian jumping systems of time-delayed Chua's circuit," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 4, pp. 2467-2475, 2019.

[3] Michael V. Basin, P. Yu, and Y. Shtessel, "Hypersonic missile adaptive sliding mode control using finite- and fixed-time observers," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 1, pp. 930-941, 2017.

[4] X. Zhao, H. Yang, and G. Zong, "Adaptive neural hierarchical sliding mode control of nonstrict-feedback nonlinear systems and an application to electronic circuits," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 7, pp. 1394-1404, 2017.

[5] S. He, W. Lyu, and F. Liu, "Robust $\mathcal{H}_\infty$ sliding mode controller design of a class of time-delayed discrete conic-type nonlinear systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 2, pp. 885-892, 2018.

[6] J. Song, Y. Wang, Y. Niu, H.K. Lam, S. He, and H. Liu, "Periodic event-triggered terminal sliding mode speed control for networked PMSM system: A GA-optimized extended state observer approach," *IEEE/ASME Transactions on Mechatronics*, doi: 10.1109/TMECH.2022.3148541.

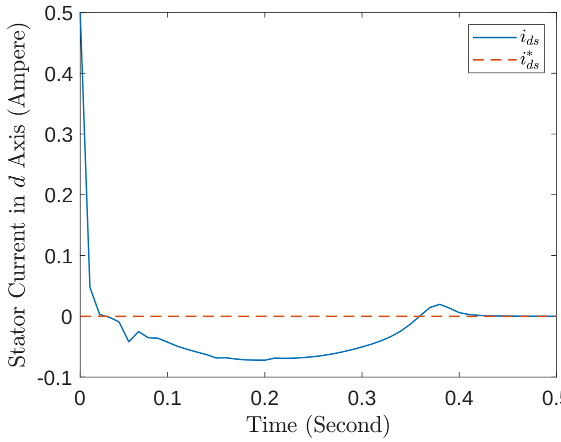

Fig. 7. Responses of the $d-$axis current using CMAC model with the gradient-descent based learning [30]–[32].

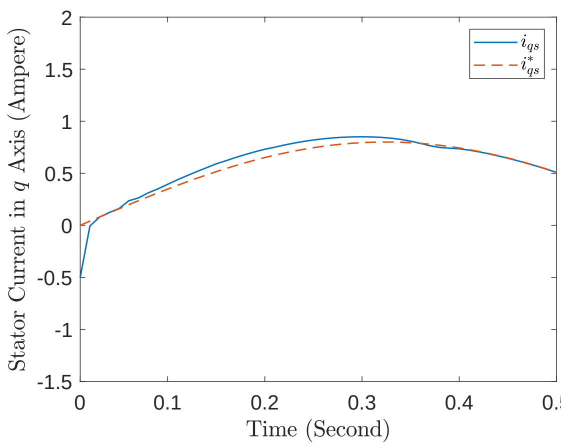

Fig. 8. Responses of the $q-$axis current using CMAC model with the gradient-descent based learning [30]–[32].

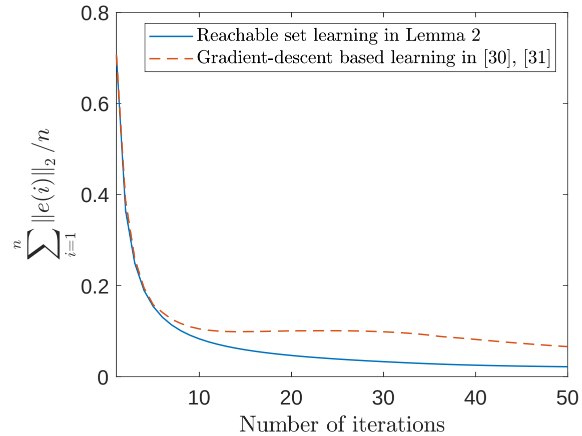

Fig. 9. Convergence performances of CMAC for the reachable set learning and gradient-descent based learning.

[7] Y. Kao, J. Xie, C. Wang, and H.R. Karimi, "A sliding mode approach to $\mathscr{H}_\infty$ non-fragile observer-based control design for uncertain Markovian neutral-type stochastic systems," *Automatica*, vol. 52, pp. 218-226, 2015.

[8] X. Yu, and O. Kaynak, "Sliding-mode control with soft computing: A survey," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 9, pp. 3275-3285, 2009.

[9] H. Zhang, Q. Qu, G. Xiao, and Y. Cui, "Optimal guaranteed cost sliding mode control for constrained-input nonlinear systems with matched and unmatched disturbances," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 6, pp. 2112-2126, 2018.

[10] Y. Wang, X. Xie, M. Chadli, S. Xie, and Y. Peng, "Sliding-mode control of fuzzy singularly perturbed descriptor systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 29, no. 8, pp. 2349-2360, 2021.

[11] C. Chen, Y. Liu, and G. Wen, "Fuzzy neural network-based adaptive control for a class of uncertain nonlinear stochastic systems," *IEEE Transactions on Cybernetics*, vol. 44, no. 5, pp. 583-593, 2014.

[12] C. Lin, M. Prasad, and A. Saxena, "An improved polynomial neural network classifier using real-coded genetic algorithm," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 11, pp. 1389-1401, 2015.

[13] D. Li, C. Philip Chen, Y. Liu, and S. Tong, "Neural network controller design for a class of nonlinear delayed systems with time-varying full-state constraints," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 9, pp. 2625-2636, 2019.

[14] C. Lin, and C.F. Hsu, "Neural-network hybrid control for antilock braking systems," *IEEE Transactions on Neural Networks*, vol. 14, no. 2, pp. 351-359, 2003.

[15] Y. Liu, S. Liu, Y. Wang, F. Lombardi, and J. Han, "A survey of stochastic computing neural networks for machine learning applications," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 7, pp. 2809-2824, 2020.

[16] Y. Liu, L. Tang, S. Tong, and C. Philip Chen, "Adaptive NN controller design for a class of nonlinear MIMO discrete-time systems," *IEEE transactions on neural networks and learning systems*, vol. 26, no. 5, pp. 1007-1018, 2015.

[17] R. Rahimilarki, Z. Gao, A. Zhang, and R. Binns, "Robust neural network fault estimation approach for nonlinear dynamic systems with applications to wind turbine systems," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 12, pp. 6302-6312, 2019.

[18] J. Albus, "A new approach to manipulator control: The cerebellar model articulation controller," *Transaction of the ASME Journal of Dynamic Systems*, vol. 97, no. 3, pp. 270-277, 1975.

[19] S. Jagannathan, S. Commuri, and F. L. Lewis, "Feedback linearization using CMAC neural networks," *Automatica*, vol. 34, no. 3, pp. 547-557, 1998.

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

[21] C. Lin, Y. Liu, and H. Li, "SoPC-based function-link cerebellar model articulation control system design for magnetic ball levitation systems," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 8, pp. 4265-4273, 2013.

[22] C. T. Chiang, and C. S. Lin, "CMAC with general basis functions," *ISA Transactions*, vol. 9, no. 7, pp. 1199-1211, 1996.

[23] H. Ying, F. Lin, and R. Sherwin, "Fuzzy discrete event systems with gradient-based online learning," *in Proc. FUZZ-IEEE Conf.*, New Orleans, LA, USA, pp. 1-6, 2019.

[24] Y. Lin, J. Chang, and C. Lin, "A TSK-type-based self-evolving compensatory interval type-2 fuzzy neural network (TSCIT2FNN) and its applications," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 1, pp. 447-459, 2013.

[25] C. Lin, R. Ramarao, and S. Gopalai, "Self-organizing adaptive fuzzy brain emotional learning control for nonlinear systems," *International Journal of Fuzzy Systems*, vol. 21, no. 7, pp. 1989-2007, 2019.

[26] X. Jing, and L. Cheng, "An optimal PID control algorithm for training feedforward neural networks," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 6, pp. 2273-2283, 2013.

[27] G. Liu, S. Xu, Y. Wei, Z. Qi, and Z. Zhang, "New insight into reachable set estimation for uncertain singular time-delay systems," *Applied Mathematics and Computation*, vol. 320, pp. 769-780, 2018.

[28] S. Yin, H. Gao, J. Qiu, and O. Kaynak, "Descriptor reduced-order sliding mode observers design for switched systems with sensor and actuator faults," *Automatica*, vol. 76, pp. 282-292, 2017.

[29] M. Chadli, and M. Darouach, "Novel bounded real lemma for discrete-time descriptor systems: Application to $\mathcal{H}_\infty$ control design," *Automatica*, vol. 48, no. 2, pp. 449-453, 2012.

[30] C. Lin, and C.F. Hsu, "Self-organizing CMAC control for a class of MIMO uncertain nonlinear systems," *IEEE Transactions on Neural Networks*, vol. 20, no. 9, pp. 1377-1384, 2009.

[31] T. Huynh, C. Lin, T. Le, and Z. Zhong, "A mixed gaussian membership

function fuzzy CMAC for a three-link robot," *in Proc. FUZZ-IEEE Conf.*, Glasgow, UK, pp. 1-6, 2020.

[32] C. Lin, and C.F. Hsu, "Adaptive CMAC-based supervisory control for uncertain nonlinear systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 2, pp. 1248-1260, 2004.

[33] H. Li, J. Wang, H.K. Lam, and Q. Zhou, "Adaptive sliding mode control for interval type-2 fuzzy systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 12, pp. 1654-1663, 2016.

[34] S. Xu, P. Dooren, R. tefan, and J. Lam, "Robust stability and stabilization for singular systems with state delay and parameter uncertainty," *IEEE Transactions on Automatic Control*, vol. 47, no. 7, pp. 1122-1128, 2002.

[35] M. Chadli, and T.M. Guerra, "LMI solution for robust static output feedback control of discrete Takagi-Sugeno fuzzy models," *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 6, pp. 1160-1165, 2012.

[36] T. Huynh , C. Lin , T. Le , H. Cho, T.T. Pham, N. Le, and F. Chao, "A new self-organizing fuzzy cerebellar model articulation controller for uncertain nonlinear systems using overlapped gaussian membership functions," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 11, pp. 9671-9682, 2020.

[37] R.S. Murray, *Schaum's Mathematical Handbook of Formulas and Tables*, Mcgraw-Hill, Inc., 1968.

[38] B. Zhang, J. Lam, and S. Xu, "Reachable set estimation and controller design for distributed delay systems with bounded disturbances," *Journal of the Franklin Institute*, vol. 351, no. 6, pp. 3068-3088, 2014.

[39] Y. Wei, J. H. Park, J. Qiu and H. Jung, "Reliable output feedback control for piecewise affine systems with Markov-type sensor failure," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 65, no. 7, pp. 913-917, 2018.

[40] E. Fridman and U. Shaked, "On reachable sets for linear systems with delay and bounded peak inputs," *Automatica*, vol. 39, no. 11, pp. 2005-2010, 2012.

[41] Neugebauer, Henrik. Parameter identification of a permanent magnet synchronous motor. MS thesis. 2012.

[42] T. Jahns, G. Kliman, and T. Neumann, "Interior permanent-magnet synchronous motors for adjustable-speed drives," *IEEE Transactions on Industry Applications*, vol. 4, pp. 738-747, 1986.