A Mixed Gaussian Membership Function Fuzzy CMAC for a Three-Link Robot

Tuan-Tu Huynh *, *Member*, *IEEE*, Chih-Min Lin *, *IEEE Fellow*, Tien-Loc Le, *Member*, *IEEE*,

and Zhixiong Zhong, *Member*, *IEEE*

***Abstract*— This research produces a mixed Gaussian membership function (GMF) fuzzy cerebellar model articulation controller (CMAC) for a three-link robot. A mixed GMF is created using the current and the previous GMFs on each layer of CMAC to detect errors efficiently, so a mixed GMF fuzzy CMAC (MGMFFC) is able to train parameters efficiently and constructs the MGMFFC structure automatically. A Lyapunov cost function and the gradient descent techniques are utilized to get the adaptive with guaranteed system's stable. Simulation studies for a three-link robot show that the MGMFFC attains favorable tracking performance.**

***Index Terms*— Gaussian membership function, fuzzy inference system, CMAC, three-link robot.**

I. Introduction

Nowadays, intelligent controllers have been studied for many fields, such as medical diagnosis system, machine learning system, noise filtering system and control system, etc. [1-4]. The intelligent control structures using fuzzy inference systems, various neural networks, traditional or modified CMACs have been used for uncertain nonlinear systems for improving the learning ability [5-8].

The control for a practical system is usually affected by uncertainty, nonlinearity and computational load problems. Therefore, some new techniques have been designed to detect error efficiently over the years. Recently, a parameterized type-2 fuzzy membership function was designed for practical applications [9]. Memari et al. proposed an intuitionistic fuzzy technique for choosing the correct sustainable supplier that involves many sub-criteria for a manufacturer [10]. A type-2 wavelet membership function was embedded in a CMAC for uncertain nonlinear systems [11]. Mohammadzadeh et al. introduced a type-3 fuzzy inference set for modeling the uncertain systems [12].

This research presents a mixed Gaussian membership function fuzzy CMAC (MGMFFC) for a three-link robot. The proposed controller uses a mixed GMF integrated into a fuzzy CMAC to detect the error values efficiently. A Lyapunov function and the gradient descent algorithm are employed to provide the online learning laws and to prove the system's stability. Simulation results using the proposed MGMFFC for a three-link robot will be given to illustrate the effectiveness of the proposed control algorithm.

II. Problem Formulation

An n-th-order MIMO nonlinear system is given as:

$$\begin{cases} \boldsymbol{q}^{(n)}(t) = \boldsymbol{G}(\boldsymbol{\psi}(t)) + \boldsymbol{H}(\boldsymbol{\psi}(t))\boldsymbol{u}_C(t) + \boldsymbol{l}(\boldsymbol{\psi}(t)) \\ \boldsymbol{y}_{out}(t) = \boldsymbol{q}(t) \end{cases} \tag{1}$$

where $\boldsymbol{q}(t) \in \Re^m$ is a state vector, $\boldsymbol{y}_{out}(t) \triangleq [y_{out1}(t)\ y_{out2}(t) \cdots y_{outm}(t)]^T \in \Re^m$ is a vector for the system outputs, and $\boldsymbol{u}_C(t) = [u_{C1}(t),\ u_{C2}(t),\ \cdots,\ u_{Cm}(t)]^T \in \Re^m$ is the vector for the control inputs. Define $\boldsymbol{\psi}(t) = \left[\boldsymbol{q}^T(t) \quad \dot{\boldsymbol{q}}^T(t) \quad \ldots \quad \boldsymbol{q}^{(n\text{-}1)^T}(t)\right]$ $\in \Re^{n\times m}$ as the vector for system states that is measurable. $\boldsymbol{G}(\boldsymbol{\psi}(t)) \in \Re^m$ and $\boldsymbol{H}(\boldsymbol{\psi}(t)) = diag(h_1, h_2, \cdots, h_m) \in \Re^{m\times m}$ are bounded nonlinear functions, $\boldsymbol{H}^{-1}(\boldsymbol{\psi}(t))$ is invertible and $\boldsymbol{l}(\boldsymbol{\psi}(t)) \in \Re^m$ is the bounded unknown disturbance.

The desired trajectory signal $\boldsymbol{y}_{REF}(t) \triangleq [y_{REF1}(t)\ y_{REF2}(t) \cdots y_{REFm}(t)]^T \in \Re^m$. The tracking error is then defined as:

$$\boldsymbol{e}(t) \triangleq \boldsymbol{y}_{REF}(t) - \boldsymbol{y}_{out}(t) \tag{2}$$

Define the vector of tracking error as:

$$\boldsymbol{e}_S(t) \triangleq \left[\boldsymbol{e}^T(t)\ \dot{\boldsymbol{e}}^T(t) \cdots \boldsymbol{e}^{(n-1)T}(t)\right]^T \in \Re^{n\times m} \tag{3}$$

An sliding function is defined as:

$$\boldsymbol{s}(\boldsymbol{e}_S(t),\ t) \triangleq \boldsymbol{e}^{n-1}(t) + \boldsymbol{\Omega}_1\boldsymbol{e}^{n-2}(t) + \cdots + \boldsymbol{\Omega}_L\int_0^t \boldsymbol{e}(\tau)d\tau \tag{4}$$

This work was supported by the Government Guiding Regional Science and Technology Development under Grant (2019L3009) and the Ministry of Science and Technology of Republic of China under Grant MOST 106-2221-E-155-002-MY3.

Chih-Min Lin is now with the Department of Electrical Engineering, Yuan Ze University, Taoyuan, Taiwan, R.O.C. (e-mail: cml@saturn.yzu.edu.tw).

Tuan-Tu Huynh is now with the Department of Electrical Engineering, Yuan Ze University, Taoyuan, Taiwan, R.O.C.; and with Department of Electrical Electronic and Mechanical Engineering, Lac Hong University, Bien Hoa, Vietnam (e-mail: huynhtuantu@saturn.yzu.edu.tw).

Tien-Loc Le is now with the Faculty of Mechanical and Aerospace, Sejong University, Seoul 143-747(05006), Korea; and with Department of Electrical Electronic and Mechanical Engineering, Lac Hong University, Bien Hoa, Vietnam. (e-mail: tienloc@lhu.edu.vn).

Zhixiong Zhong is now with Key Laboratory of Information Processing and Intelligent Control, Minjiang University, Fuzhou 350121, China. (email: zhixiongzhong2012@126.com)

* Corresponding author: Tuan-Tu Huynh and Chih-Min Lin

978-1-7281-6932-3/20/$31.00 ©2020 IEEE

where $\boldsymbol{\Omega}_j \in \Re^{m\times m}$, for $j=1,2,\ldots,L$ are the matrices with positive constant and $\boldsymbol{\Omega} \triangleq [\boldsymbol{\Omega}_1^T, \boldsymbol{\Omega}_2^T \cdots, \boldsymbol{\Omega}_L^T]^T \in \Re^{Lm\times m}$. If $\boldsymbol{G}(\boldsymbol{\psi}(t))$, $\boldsymbol{H}(\boldsymbol{\psi}(t))$ and $\boldsymbol{l}(\boldsymbol{\psi}(t))$ are known exactly, then an ideal controller can be given as:

$$\boldsymbol{u}_{\mathrm{ID}}^*(t) = \boldsymbol{H}^{-1}(\boldsymbol{\psi}(t))\,[\boldsymbol{y}_{ref}^{(n)}(t) - \boldsymbol{G}(\boldsymbol{\psi}(t)) - \boldsymbol{l}(\boldsymbol{\psi}(t)) + \boldsymbol{\Omega}^T \boldsymbol{e}_S(t)] \cdot \tag{5}$$

Substituting (5) into (1), gives:

$$\dot{s}(\boldsymbol{e}_S(t),\ t) = \boldsymbol{e}^{(n)}(t) + \boldsymbol{\Omega}^T \boldsymbol{e}_S(t) = 0 \cdot \tag{6}$$

However, $\boldsymbol{u}_{\mathrm{ID}}^*(t)$ in (5) is unattainable due to $\boldsymbol{l}(\boldsymbol{\psi}(t))$ is always unknown in practical applications. Therefore, a MGMFFC is proposed to imitate the ideal controller. The proposed MGMFFC is shown in Fig. 1, which consists of a MGMFFC, $\boldsymbol{u}_{MGMFFC}$, and a robust controller, $\boldsymbol{u}_{RB}$, as follows:

$$\boldsymbol{u}_C(t) = \boldsymbol{u}_{MGMFFC}(t) + \boldsymbol{u}_{RB}(t) \tag{7}$$

where $\boldsymbol{u}_{MGMFFC}(t)$ is the main controller and the robust controller, $\boldsymbol{u}_{RB}(t)$, compensates the approximation error for $\boldsymbol{u}_{\mathrm{ID}}^*(t)$ and $\boldsymbol{u}_{MGMFFC}(t)$ to give robust control performance.

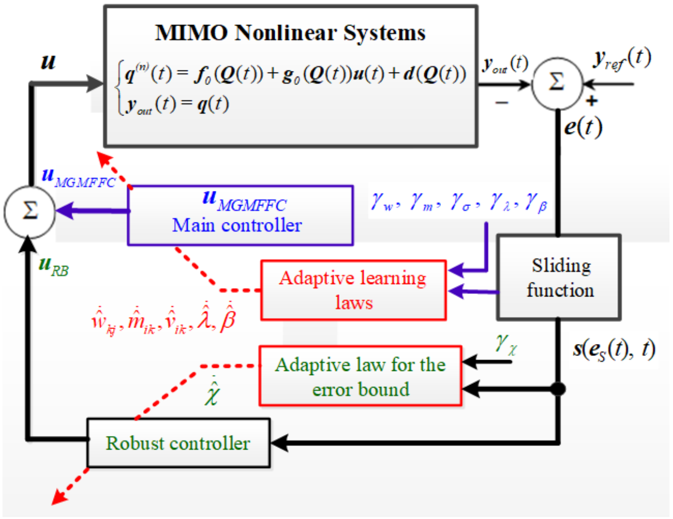
MIMO Nonlinear Systems
$\begin{cases} \boldsymbol{q}^{(n)}(t) = \boldsymbol{f}_0(\boldsymbol{Q}(t)) + \boldsymbol{g}_0(\boldsymbol{Q}(t))\boldsymbol{u}(t) + \boldsymbol{d}(\boldsymbol{Q}(t)) \\ \boldsymbol{y}_{out}(t) = \boldsymbol{q}(t) \end{cases}$
$\boldsymbol{u}$
$\boldsymbol{y}_{out}(t)$
$\boldsymbol{y}_{ref}(t)$
Σ
−
+
$\boldsymbol{e}(t)$
$\boldsymbol{u}_{MGMFFC}$
Σ
$\boldsymbol{u}_{MGMFFC}$
Main controller
$\gamma_w, \gamma_m, \gamma_\sigma, \gamma_\lambda, \gamma_\beta$
Sliding function
$\boldsymbol{u}_{RB}$
Adaptive learning laws
$\dot{\hat{w}}_{kj}, \dot{\hat{m}}_{ik}, \dot{\hat{v}}_{ik}, \dot{\hat{\lambda}}, \dot{\hat{\beta}}$
γ_χ
$s(\boldsymbol{e}_S(t),\ t)$
Adaptive law for the error bound
$\dot{\hat{\chi}}$
Robust controller

Fig. 1. Block diagram for the MGMFFC

III. Design of MGMFFC

A. Mixed Gaussian Membership Function Fuzzy CMAC

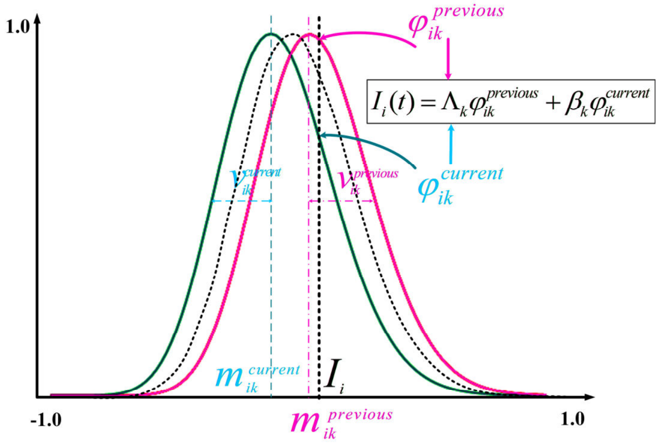

Fig. 2. A mixed Gaussian membership function

This research uses a mixed GMF as shown in Fig. 3 [13]. There are two GMFs on a layer (i.e. previous and current GMFs), and they are activated simultaneously. It implies that the GMF of the previous state will remain active while the next state is activated.

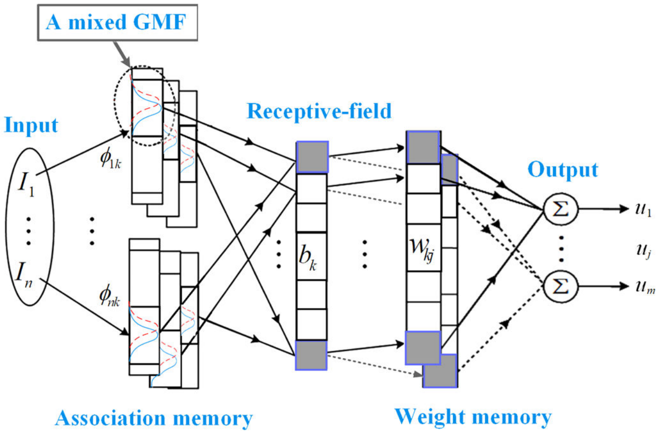

Fig. 3. Architecture of the proposed MGMFFC

This MGMFFC contains five structures:

1) **Input:** $\boldsymbol{I} = [I_1 \quad I_2 \quad \cdots \quad I_n]^T$.

2) **Association memory:** Two GMFs, $\varphi_{ik}^{current}$ and $\varphi_{ik}^{previous}$, are handled as a mixed GMF. The value of $\varphi_{ik}^{previous}$ and $\varphi_{ik}^{current}$ are defined as:

$$\varphi_{ik}^{current} = \exp\left[-\frac{\left(I_i - m_{ik}^{current}\right)^2}{\left(v_{ik}^{current}\right)^2}\right], \tag{8}$$

$$\varphi_{ik}^{previous} = \exp\left[-\frac{\left(I_i - m_{ik}^{previous}\right)^2}{(v_{ik}^{previous})^2}\right], \tag{9}$$

where $m_{ik}^{previous}$ and $m_{ik}^{current}$ are the previous and current means and $v_{ik}^{previous}$ and $v_{ik}^{current}$ are the previous and current variances for k th layer and i th input. $i=1,2,\ldots,n$ and $k=1,2,\ldots,n_k$.

The previous GMF, $\varphi_{ik}^{previous}$, is saved in memory to forecast the subsequent state.

3) **Receptive-field:** A 2D example for calculating the synthesis hypercubes is presented in Fig. 4. The synthesis hypercube is the sum of two values of the current state (6, 5) and the previous state (4, 7). The synthesis hypercube is $b_k^{synthesis}$, which is defined as:

$$b_k^{synthesis} \triangleq \Lambda_k b_k^{previous} + \beta_k \prod_{i=1}^{n} \varphi_{ik}^{current} \tag{10}$$

where β_k and Λ_k are respectively adaptive gains and $b_k^{previous}$ is the previous hypercube value.

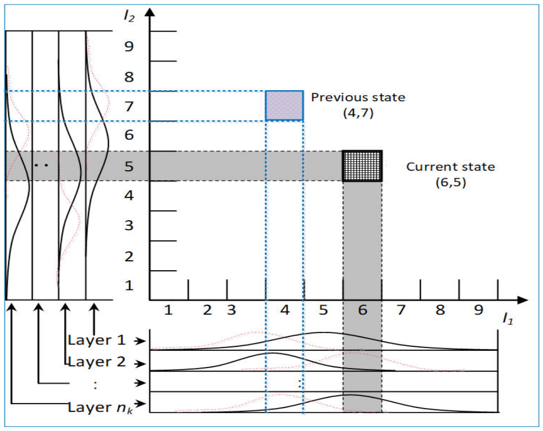

Fig. 4. A 2D example for calculating the synthesis hypercubes

4) **Weight memory:** The weight is define as:

$$\boldsymbol{w}_j = \begin{bmatrix} w_{1j} & \cdots & w_{kj} & \cdots & w_{n_k j} \end{bmatrix}^T \tag{11}$$

where w_{kj} is the output weight.

$$\boldsymbol{W} = \begin{bmatrix} w_1 & \cdots & w_j & \cdots & w_m \end{bmatrix} \tag{12}$$

5) **Output:** The *j*-th output of MGMFFC is calculated as

$$u_j = \sum_{k=1}^{n_k} w_{kj} b_k^{synthesis} = \sum_{k=1}^{n_k} w_{kj} \left[\Lambda_k b_k^{previous} + \beta_k \prod_{i=1}^{n} \varphi_{ik}^{current} \right], \tag{13}$$

The MGMFFC output is rewritten via a vector form:

$$\boldsymbol{u} = \boldsymbol{W}\boldsymbol{b} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n_k} \\ w_{21} & w_{22} & \cdots & w_{2n_k} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m1} & w_{m2} & \cdots & w_{mn_k} \end{bmatrix} \begin{bmatrix} b_1^{synthesis} \\ b_2^{synthesis} \\ \vdots \\ b_{n_k}^{synthesis} \end{bmatrix} \tag{14}$$

B. Adaptive Learning Laws for MGMFFC

Taking the derivative of $\boldsymbol{s}(\boldsymbol{e}_S(t),\ t)$ and using (1), yields

$$\dot{\boldsymbol{s}}(\boldsymbol{e}_S(t),\ t) = \boldsymbol{e}^{(n)}(t) + \boldsymbol{\Omega}^T \boldsymbol{e}_S(t)$$
$$= -\boldsymbol{G}(\boldsymbol{\psi}(t)) - \boldsymbol{H}(\boldsymbol{\psi}(t))\boldsymbol{u}(t) + \boldsymbol{y}_{ref}^{(n)}(t) - \boldsymbol{l}(\boldsymbol{\psi}(t)) + \boldsymbol{\Omega}^T \boldsymbol{e}_S(t) \tag{15}$$

Insert (7) into (15) and multiply both sides by $\boldsymbol{s}^T(\boldsymbol{e}_S(t),\ t)$, gives:

$$\boldsymbol{s}^T(\boldsymbol{e}_S(t),t)\ \dot{\boldsymbol{s}}(\boldsymbol{e}_S(t),t) = -\boldsymbol{s}^T(\boldsymbol{e}_S(t),t)\boldsymbol{G}(\boldsymbol{\psi}(t)) -$$
$$\boldsymbol{s}^T(\boldsymbol{e}_S(t),t)\ \boldsymbol{H}(\boldsymbol{\psi}(t))\left[\boldsymbol{u}_{MGMFFC}(t) + \boldsymbol{u}_{RB}(t)\right] +$$
$$\boldsymbol{s}^T(\boldsymbol{e}_S(t),t)\left[\boldsymbol{y}_{ref}^{(n)}(t) - \boldsymbol{l}(\boldsymbol{\psi}(t)) + \boldsymbol{\Omega}^T \boldsymbol{e}_S(t)\right] \tag{16}$$

where

$$\boldsymbol{u}_{MGMFFC}(t) = \boldsymbol{H}^{-1}(\boldsymbol{\psi}(t))\ [\boldsymbol{y}_{ref}^{(n)}(t) - \boldsymbol{G}(\boldsymbol{\psi}(t)) - \boldsymbol{l}(\boldsymbol{\psi}(t)) + \boldsymbol{\Omega}^T \boldsymbol{e}_S(t)] \tag{17}$$

Define a Lyapunov cost function as:

$$V\left(\boldsymbol{s}(\boldsymbol{e}_S(t),t)\right) = \frac{1}{2}\boldsymbol{s}^T(\boldsymbol{e}_S(t),t)\,\boldsymbol{s}(\boldsymbol{e}_S(t),t) \tag{18}$$

$\Rightarrow \dot{V}\left(\boldsymbol{s}(\boldsymbol{e}_S(t),t)\right) = \boldsymbol{s}^T(\boldsymbol{e}_S(t),t)\,\dot{\boldsymbol{s}}(\boldsymbol{e}_S(t),t)$. The aim is to minimize $\boldsymbol{s}^T(\boldsymbol{e}_S(t),\ t)\dot{\boldsymbol{s}}(\boldsymbol{e}_S(t),\ t)$ for reaching fast convergence of $\boldsymbol{s}$. The gradient descent technique is then used so the parameters are updated through the online adaptive laws as follows:

$$\dot{\hat{w}}_{kj} = -\gamma_w \frac{\partial \boldsymbol{s}^T(\boldsymbol{e}_S(t),\ t)\dot{\boldsymbol{s}}(\boldsymbol{e}_S(t),\ t)}{\partial w_{kj}} = -\gamma_w \frac{\partial \boldsymbol{s}^T(\boldsymbol{e}_S(t),\ t)\dot{\boldsymbol{s}}(\boldsymbol{e}_S(t),\ t)}{\partial u_j}\frac{\partial u_j}{\partial w_{kj}}$$
$$= \gamma_w s_j(t) g_{0j} b_k^{synthesis} \tag{19}$$

$$\dot{\hat{m}}_{ik} = -\gamma_m \frac{\partial \boldsymbol{s}^T(\boldsymbol{e}_S(t),\ t)\dot{\boldsymbol{s}}(\boldsymbol{e}_S(t),\ t)}{\partial m_{ik}} = -\gamma_m \frac{\partial \boldsymbol{s}^T(\boldsymbol{e}_S(t),\ t)\dot{\boldsymbol{s}}(\boldsymbol{e}_S(t),\ t)}{\partial u_j}\frac{\partial u_j}{\partial \varphi_{ik}^{current}}\frac{\partial \varphi_{ik}^{current}}{\partial m_{ik}}$$
$$= \gamma_m s_j(t) g_{0j} w_{kj} \beta_k \prod_{i=1}^{n} \varphi_{ik}^{current} \frac{2(I_i - m_{ik})}{v_{ik}^2} \tag{20}$$

$$\dot{\hat{v}}_{ik} = -\gamma_v \frac{\partial \boldsymbol{s}^T(\boldsymbol{e}_S(t),\ t)\dot{\boldsymbol{s}}(\boldsymbol{e}_S(t),\ t)}{\partial v_{ik}} = -\gamma_v \frac{\partial \boldsymbol{s}^T(\boldsymbol{e}_S(t),\ t)\dot{\boldsymbol{s}}(\boldsymbol{e}_S(t),\ t)}{\partial u_j}\frac{\partial u_j}{\partial \varphi_{ik}^{current}}\frac{\partial \varphi_{ik}^{current}}{\partial v_{ik}}$$
$$= \gamma_v s_j(t) g_{0j} w_{kj} \beta_k \prod_{i=1}^{n} \varphi_{ik}^{current} \left[\frac{2(I_i - m_{ik})^2}{v_{ik}^3}\right] \tag{21}$$

$$\dot{\hat{\Lambda}}_k = -\gamma_\Lambda \frac{\partial \boldsymbol{s}^T(\boldsymbol{e}_S(t),\ t)\dot{\boldsymbol{s}}(\boldsymbol{e}_S(t),\ t)}{\partial \hat{\Lambda}_k} = -\gamma_\Lambda \frac{\partial \boldsymbol{s}^T(\boldsymbol{e}_S(t),\ t)\dot{\boldsymbol{s}}(\boldsymbol{e}_S(t),\ t)}{\partial u_j}\frac{\partial u_j}{\partial b_k^{previous}}\frac{\partial b_k^{previous}}{\partial \Lambda_k}$$
$$= \gamma_\Lambda s_j(t) g_{0j} w_{kj}^2 \Lambda_k b_k^{previous} \tag{22}$$

$$\dot{\hat{\beta}}_k = -\gamma_\beta \frac{\partial \boldsymbol{s}^T(\boldsymbol{e}_S(t),\ t)\dot{\boldsymbol{s}}(\boldsymbol{e}_S(t),\ t)}{\partial \hat{\beta}_k} = -\gamma_\beta \frac{\partial \boldsymbol{s}^T(\boldsymbol{e}_S(t),\ t)\dot{\boldsymbol{s}}(\boldsymbol{e}_S(t),\ t)}{\partial u_j}\frac{\partial u_j}{\partial \varphi_{ik}^{current}}\frac{\partial \varphi_{ik}^{current}}{\partial \beta_k}$$
$$= \gamma_\beta s_j(t) g_{0j} w_{kj}^2 \beta_k \prod_{i=1}^{n} \varphi_{ik}^{current} \tag{23}$$

where u_j is the *j*th element of u, and γ_m, γ_σ, γ_w ,γ_β and γ_Λ are respectively learning rates for means, variances, weights, and adaptive gains for the current and previous states.

C. Robust controller design

This study uses a $\mathrm{sgn}(.)$ function robust compensation controller to cover the approximation error:

$$\boldsymbol{u}_{RB} = -\boldsymbol{H}^{-1}(\boldsymbol{\psi}(t))\hat{\chi}\,\mathrm{sgn}(\boldsymbol{s}(\boldsymbol{e}_S(t),\ t)) \tag{24}$$

The error bound is updated online by:

$$\dot{\hat{\chi}} = \gamma_\chi |\boldsymbol{s}| \tag{25}$$

where γ_χ are positive learning rates.

Proof:

Define a Lyapunov function as

$$V(\boldsymbol{s}(\boldsymbol{e}_S(t),\ t), \tilde{\chi}) = \frac{\boldsymbol{s}^T\boldsymbol{s}}{2} + \frac{\tilde{\chi}^2}{2\gamma_\chi}, \tag{26}$$

Differentiating (26) with respect to time then using (15) and (24), gives:

$$\dot{V}(\boldsymbol{s}(\boldsymbol{e}_S(t),\ t), \tilde{\chi}) = \boldsymbol{s}^T\dot{\boldsymbol{s}} + \frac{\tilde{\chi}\dot{\tilde{\chi}}}{\gamma_\chi}$$
$$= \boldsymbol{s}^T(\boldsymbol{\varepsilon} - \hat{\chi}\,\mathrm{sgn}(\boldsymbol{s})) + \frac{\tilde{\chi}\dot{\tilde{\chi}}}{\gamma_\chi}$$
$$= \left(\boldsymbol{s}^T\boldsymbol{\varepsilon} - \hat{\chi}|\boldsymbol{s}|\right) + \frac{\tilde{\chi}\dot{\tilde{\chi}}}{\gamma_\chi}. \tag{27}$$

The approximation error $\boldsymbol{\varepsilon}$ is the minimum reconstructed error between $\boldsymbol{u}_{\mathrm{ID}}^{*}(t)$ and $\boldsymbol{u}_{MGMFFC}(t)$, and it is assumed to be bounded by $0 \le |\boldsymbol{\varepsilon}| \le \chi$.

If the adaptive law of the error bound is selected as

$$\dot{\tilde{\chi}} = -\dot{\hat{\chi}} = -\gamma_{\chi}|\boldsymbol{s}|, \tag{28}$$

then (27) becomes:

$$\dot{V}(\boldsymbol{s}(\boldsymbol{e}_S(t),\ t), \tilde{\chi}) = \boldsymbol{s}^T\boldsymbol{\varepsilon} - \hat{\chi}|\boldsymbol{s}| - (\chi - \hat{\chi})|\boldsymbol{s}|$$

$$= (\boldsymbol{s}^T\boldsymbol{\varepsilon} - \chi|\boldsymbol{s}|) \le (|\boldsymbol{\varepsilon}||\boldsymbol{s}| - \chi|\boldsymbol{s}|)$$

$$= -(\chi - |\boldsymbol{\varepsilon}|)|\boldsymbol{s}| \le 0. \tag{29}$$

Since $\dot{V}(\boldsymbol{s}(\boldsymbol{e}_S(t),\ t), \tilde{\chi})$ is negative semi-definite that is $\dot{V}(\boldsymbol{s}(\boldsymbol{e}_S(t),\ t), \tilde{\chi}) \le \dot{V}(\boldsymbol{s}(0), \tilde{\chi}(0))$, it implies that $\boldsymbol{s}$ and $\tilde{\chi}$ are bounded. Let function $\xi \equiv (\chi - |\boldsymbol{\varepsilon}|)\boldsymbol{s} \le (\chi - |\boldsymbol{\varepsilon}|)|\boldsymbol{s}| \le -\dot{V}(\boldsymbol{s}(\boldsymbol{e}_S(t),\ t), \tilde{\chi})$, and integrate χ with respect to time, yields

$$\int_0^t \xi(\tau)d\tau \le \dot{V}(\mathbf{s}(0), \tilde{\chi}(0)) - \dot{V}(\mathbf{s}, \tilde{\chi}). \tag{30}$$

Because $\dot{V}(\mathbf{s}(0), \tilde{\chi}(0))$ is bounded, and $\dot{V}(\mathbf{s}, \tilde{\chi})$ is non-increasing and bounded, the following result is obtained:

$$\lim_{t\to\infty}\int_0^t \xi(\tau)d\tau < \infty. \tag{31}$$

$\dot{\xi}$ is bounded, so $\lim_{t\to\infty}\xi(t) = 0$. Then, $\boldsymbol{s} \to 0$ when $t \to \infty$. Finally, the MGMFFC is asymptotically stable. Therefore, the proof is complete.

IV. Simulation Results

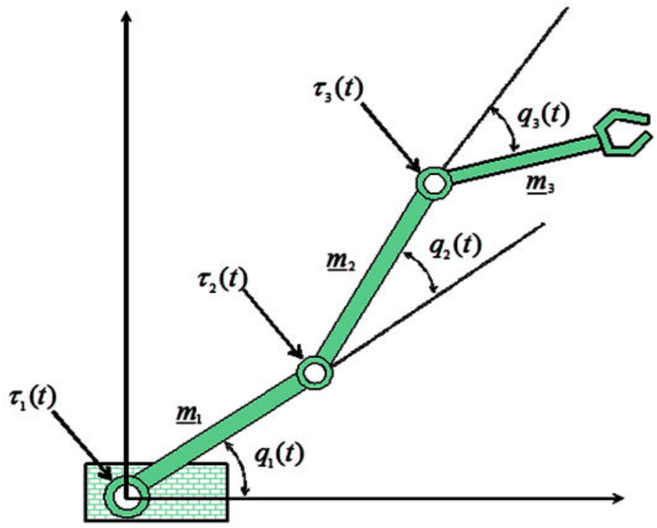

Fig. 5. A three-link robot

The dynamic equation is taken in [14] as:

$$\boldsymbol{\Psi}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{\Gamma}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{\tau}_d = \boldsymbol{\tau}, \tag{32}$$

where

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

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

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

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

Table 1
The Parameters of Three-Link Robot

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

All system parameters for the three-link robot are listed in Table 1.

Definitions of variables in (32) are described in Table 2 as follows:

TABLE 2
DEFINITIONS OF VARIABLES FOR THREE-LINK ROBOT

$\boldsymbol{q}=[q_1(t),q_2(t),\, q_3(t)]^T\in\Re^3$	The vector of angular positions
$\dot{\boldsymbol{q}},\ \ddot{\boldsymbol{q}}\in\Re^3$	The vector of joint velocities and accelerations
$\boldsymbol{\Psi}(\boldsymbol{q})\in\Re^{3\times3}$	The inertia matrix
$\boldsymbol{\tau}\in\Re^3$	The input torques
$\boldsymbol{\tau}_d\in\Re^3$	The external disturbances
$\boldsymbol{\Gamma}(\boldsymbol{q},\dot{\boldsymbol{q}})\in\Re^{3\times3}$	The Coriolis/Centripetal matrix
$\boldsymbol{g}(\boldsymbol{q})\in\Re^3$	The vector of gravity, in which $\underline{g}=9.8\ m/s^2$
$\underline{m}_i$	The *i*th link mass
$\underline{a}_i$	The *i*th link length
$\underline{c}_{ij}=cos(\underline{q}_i+\underline{q}_j)$ and $\underline{s}_{ij}=sin(\underline{q}_i+\underline{q}_j)$	The short hand notations, in which $\underline{i}_i$ is the moment of inertia ($kg\ m^2$) and $\underline{d}_i$ is defined in the first row of Table 1

Equation (32) is rewritten as:

$$\ddot{\boldsymbol{\psi}}(t)=\boldsymbol{G}(\boldsymbol{\psi}(t))+\boldsymbol{H}(\boldsymbol{\psi}(t))\,\underline{\boldsymbol{u}}(t)+\boldsymbol{D}(\boldsymbol{\psi}(t)), \tag{33}$$

where $\boldsymbol{\psi}(t)\ \underline{\underline{\Delta}}\ [q_1(t),\ q_2(t),\ q_3(t)]^T\equiv[x_1(t),\ \ x_2(t),\ \ x_3(t)]^T$, $\boldsymbol{G}(\boldsymbol{\psi}(t))=-\boldsymbol{\Psi}^{-1}(\boldsymbol{q})\left[\boldsymbol{\Gamma}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}+\boldsymbol{g}(\boldsymbol{q})\right]$, $\boldsymbol{H}(\boldsymbol{\psi}(t))=\boldsymbol{\Psi}^{-1}(\boldsymbol{q})$, $\boldsymbol{D}(\boldsymbol{\psi}(t))=-\boldsymbol{\Psi}^{-1}(\boldsymbol{q})\ \boldsymbol{\tau}_d$ and $\boldsymbol{u}(t)\underline{\underline{\Delta}}[\tau_1(t),\tau_2(t),\ \tau_3(t)]^T\in\Re^3$. When $t\le 11.2$ seconds, the reference commands are given as: $\ddot{x}_{di}(t)=\zeta_1\times\dot{x}_{di}(t)+\zeta_2\times x_{di}(t)+\zeta_3\ \times\gamma_i$, in which $i=1,\ 2$ and 3, $\zeta_1=-21.13,\ \zeta_2=-111.63,\ \zeta_3=111.63$. The initial conditions are $x_3(0)=0.2, x_2(0)=0.1, x_1(0)=0.3,\ \dot{x}_3(0)=0,\ \dot{x}_2(0)=0, \dot{x}_1(0)=0,$ $x_{d3}(0)=0, x_{d2}(0)=0, x_{d1}(0)=0,\ \ \dot{x}_{d3}(0)=0, \dot{x}_{d2}(0)=0,\ \dot{x}_{d1}(0)=0.$ When $t\ge 11.2$ seconds, the sinusoid function commands are activated as: $x_{d1}(t)=0.6\sin(2.2t)$, $x_{d2}(t)=0.2\sin(2.2\mathrm{t})$, $x_{d3}(t)=0.4\sin(2.2\mathrm{t})$. The sliding function is $\boldsymbol{s}(t)=\ \dot{\boldsymbol{e}}(t)+5.1\boldsymbol{e}(t)$. The input ranges are normalized within $\{[-1.8,1.8][-1.8,1.8][-1.8,1.8]\}$. The initial means of GMFs for the previous state and the current state set as $[-2.1,-1.6,-0.9,-0.5,-0.3,0.3,0.5,0.9,1.6,2.1]$, the initial variances are $v_{ik}=1.1$, for $k=1,2,\cdots,10$ and $i=1,2$ and 3. The learning rates for MGMFFC are $\gamma_I=1,\ \gamma_P=1,$ $\gamma_w=\gamma_m=\gamma_\sigma=0.05,\ \eta_I=\eta_m=\eta_\sigma=\eta_\Lambda=\eta_\beta=\eta_{D_e}=0.01,\ \delta=0.1,$ $\gamma_{D_e}=0.1,\ \gamma_\beta=1,$ and $\gamma_\Lambda=0.1$. The initial value of $\boldsymbol{\Lambda}_0$ and $\boldsymbol{\beta}_0$ are set randomly between -1 and 1. In order to show the effectiveness of the MGMFFC, the TFLFCM (TOPSIS Function-link CMAC) [15] is also used for the three-link robot to compare their performance. The angular trajectories and the joint velocities are respectively plotted in Figs. 6 and 8 (a)-(c). Enlarge of angular trajectories and joint velocities are shown in Figs. 7 (a)-(c) and Figs. 9 (a)-(c), respectively. The control efforts and their enlargements are displayed in Figs. 10 and 11 (a)-(c). The tracking errors and the enlargement are plotted in Figs. 12 and 13 (a)-(c). Finally, two adaptive gains $\boldsymbol{\Lambda}$ and $\boldsymbol{\beta}$ of the MGMFFC are displayed in Fig. 14. The simulation results indicate that the proposed MGMFFC achieves excellent control performance under external disturbance. The tracking errors for three links converge quickly to zero (see Figs. 12 and 13 (a)-(c)). The tracking angular positions of three links follow the reference angular positions well (see Figs. 6 and 7 (a)-(c)). The response of the proposed MGMFFC is fast (see Fig. 10 and 11 (a)-(c)). The proposed MGMFFC achieves a favorable tracking response when two adaptive prediction gains $\boldsymbol{\Lambda}$ and $\boldsymbol{\beta}$ are adjusted online as shown in Fig. 14. The total RMSE (root mean square error) for the TFLFCM and the proposed MGMFFC are measured in Table 3. In summary, Table 3 and the simulation results confirm that the proposed MGMFFC achieves better tracking performance with quicker convergence and smaller tracking error than for TFLFCM.

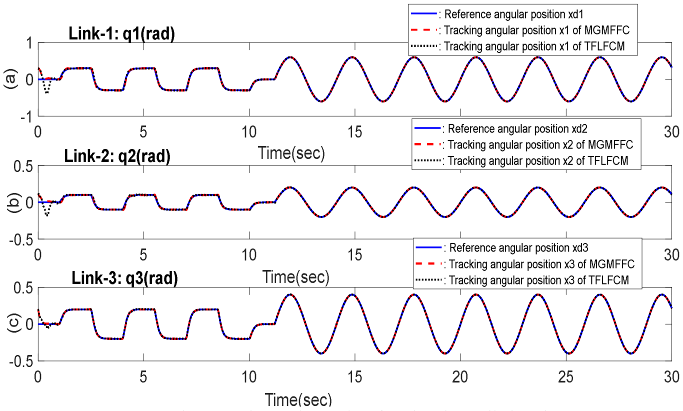

Fig. 6. The angular trajectories for the three-link robot

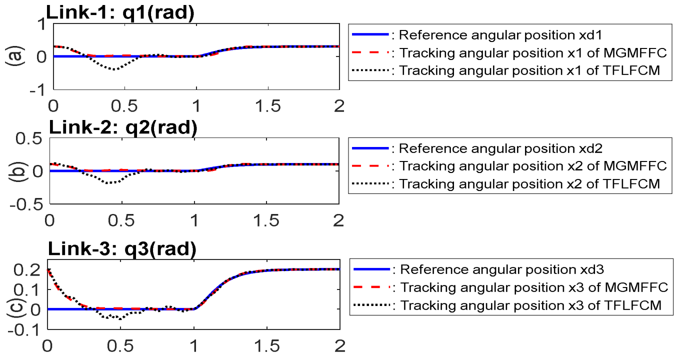

Fig. 7. Enlarge of angular trajectories for the three-link robot

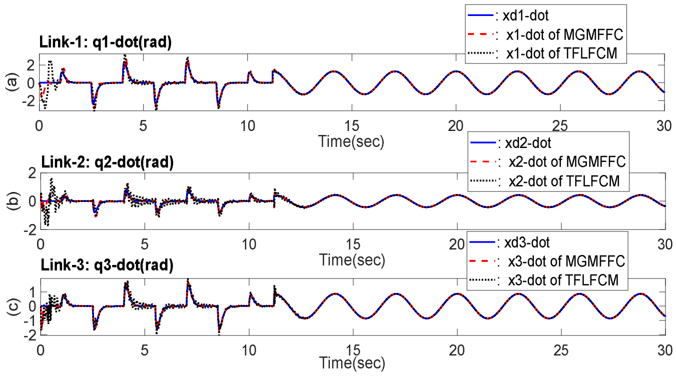

Fig. 8. The joint velocities for the three-link robot

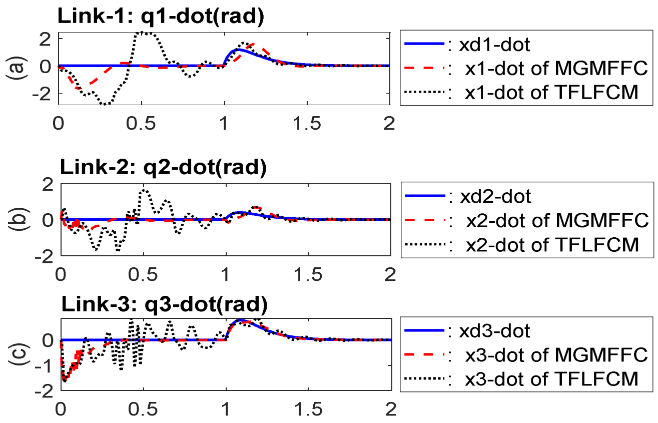

Fig. 9. Enlarge of joint velocities for the three-link robot

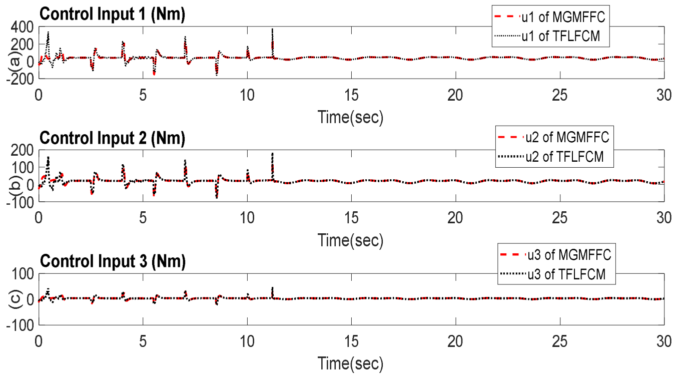

Fig. 10. The control efforts for the three-link robot

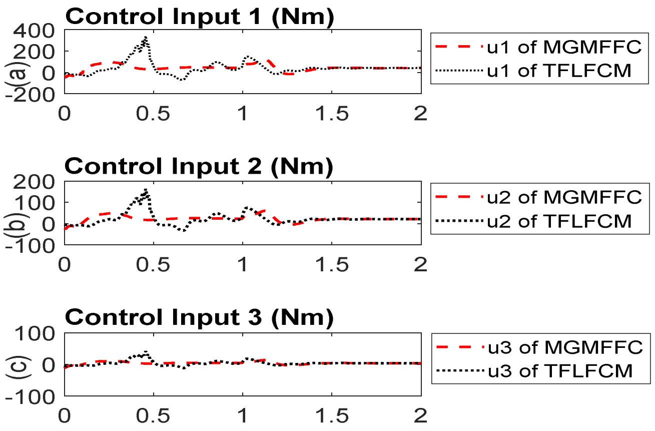

Fig. 11. Enlarge of control efforts for the three-link robot

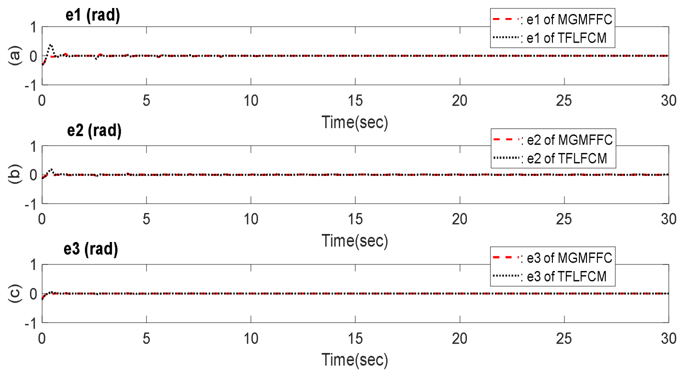

Fig. 12. The tracking errors for the three-link robot

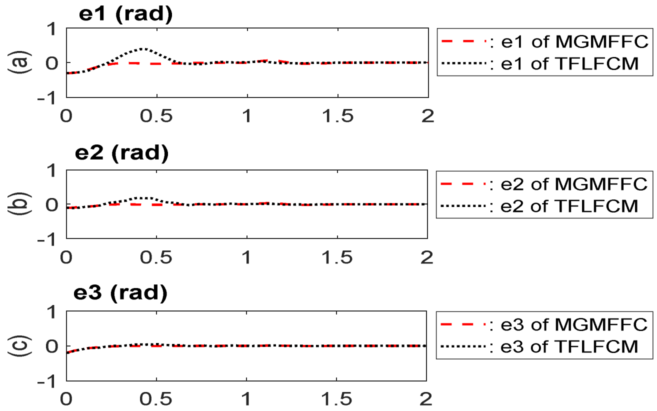

Fig. 13. Enlarge of tracking errors for the three-link robot

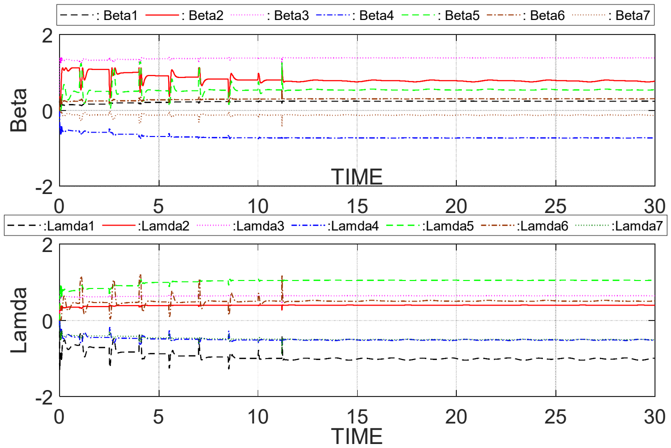

Fig. 14. Two adaptive gains Λ and β of the MGMFFC

TABLE 3
TOTAL RMSE FOR THREE-LINK ROBOT

	RMSE1	RMSE2	RMSE3
TFLFCM	0.11	0.06	0.04
MGMFFC	0.09	0.04	0.01

V. CONCLUSION

This research shows a mixed Gaussian membership function-based fuzzy CMAC for a three-link robot. The proposed MGMFFC is the main controller and its robust ability for handling system uncertainty has been demonstrated. A sgn(.) robust compensation controller is also applied to cover approximation errors. The online learning laws for all system parameters are derived from the gradient descent algorithm. Future studied can apply the proposed method for a practical model.

REFERENCES

[1] T.-L. Le, T.-T. Huynh, L.-Y. Lin, C.-M. Lin, and F. Chao, "A K-means interval type-2 fuzzy neural network for medical diagnosis," *International Journal of Fuzzy Systems*, pp. 1-12. DOI: 10.1007/s40815-019-00730-x

[2] J. Liu, C. M. Lin, and F. Chao, "Gradient boost with convolution neural network for stock forecast," in *Advances in Computational Intelligence Systems, Springer International Publishing*, 2020, pp. 155-165.

[3] T. L. Le, T. T. Huynh, and C. M. Lin, "Adaptive filter design for active noise cancellation using recurrent type-2 fuzzy brain emotional learning neural network," *Neural Computing and Applications*, 2019. DOI: 10.1007/s00521-019-04366-8

[4] C. M. Lin and T. T. Huynh, "Dynamic TOPSIS fuzzy cerebellar model articulation controller for magnetic levitation system," *Journal of Intelligent & Fuzzy Systems*, vol. 36, no. 3, pp. 2465-2480, 2019.

[5] J. M. Mendel, R. Chimatapu, and H. Hagras, "Comparing the performance potentials of singleton and non-singleton type-1 and interval type-2 fuzzy systems in terms of sculpting the state space," *IEEE Transactions on Fuzzy Systems*, 2019. DOI: 10.1109/TFUZZ.2019.2916103

[6] T. L. Le, T. T. Huynh, C. M. Lin, and F. Chao, "Breast cancer diagnosis using k-means type-2 fuzzy neural network," in *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2018, pp. 4150-4154.

[7] T. Tao and S. F. Su, "CMAC-based previous step supervisory control schemes for relaxing bound in adaptive fuzzy control," *Applied Soft Computing*, vol. 11, no. 8, pp. 5715-5723, 2011.

[8] T. T. Huynh, T. L. Le, and C. M. Lin, "A TOPSIS multi-criteria decision method-based intelligent recurrent wavelet CMAC control system design for MIMO uncertain nonlinear systems," Neural Computing and Applications, pp. 1-19, 2018. DOI: 10.1007/s00521-018-3795-4

[9] P. Melin, E. Ontiveros-Robles, C. I. Gonzalez, J. R. Castro, and O. Castillo, "An approach for parameterized shadowed type-2 fuzzy membership functions applied in control applications," *Soft Computing*, vol. 23, no. 11, pp. 3887-3901, 2019.

[10] A. Memari, A. Dargi, M. R. Akbari Jokar, R. Ahmad, and A. R. Abdul Rahim, "Sustainable supplier selection: A multi-criteria intuitionistic fuzzy TOPSIS method," *Journal of Manufacturing Systems*, vol. 50, pp. 9-24, 2019.

[11] T. L. Le, T. T. Huynh, and C. M. Lin, "Self-evolving interval type-2 wavelet cerebellar model articulation control design for uncertain nonlinear systems using PSO," *International Journal of Fuzzy Systems*, pp. 1-18. DOI: 10.1007/s40815-019-00735-6

[12] A. Mohammadzadeh, M. H. Sabzalian, and W. Zhang, "An interval type-3 fuzzy system and a new online fractional-order learning algorithm: theory and practice," *IEEE Transactions on Fuzzy Systems,* 2019. DOI: 10.1109/TFUZZ.2019.2928509

[13] Tuan-Tu Huynh et al., ‘A new self-organizing fuzzy cerebellar model articulation controller for uncertain nonlinear systems using overlapped gaussian membership functions’, *IEEE Transactions on Industrial Electronics*, 2019. DOI: 10.1109/TIE.2019.2952790

[14] E. Kim, “Output feedback tracking control of robot manipulators with model uncertainty via adaptive fuzzy logic,” *IEEE Transactions on Fuzzy Systems*, vol. 12, pp. 368-378, 2004.

[15] C. M. Lin and T. T. Huynh, "Function-link fuzzy cerebellar model articulation controller design for nonlinear chaotic systems using topsis multiple attribute decision-making method," *International Journal of Fuzzy Systems*, vol. 20, no. 6, pp. 1839-1856, 2018.