INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
Online ISSN 1841-9844, ISSN-L 1841-9836, Volume: 17, Issue: 4, Month: August, Year: 2022
Article Number: 4606, https://doi.org/10.15837/ijccc.2022.4.4606

CCC Publications
AGORA UNIVERSITY PRESS

A Novel Self-organizing Fuzzy Cerebellar Model Articulation Controller Based Overlapping Gaussian Membership Function for Controlling Robotic System

Thanh-Quyen Ngo, Dinh-Khoi Hoang, Trong-Toan Tran, Thanh-Thuan Nguyen
Van-Tho Nguyen, Long-Ho Le

Thanh-Quyen Ngo*
Faculty of Electrical Engineering Technology
Industrial of University Ho Chi Minh City, Vietnam
*Corresponding author: ngothanhquyen@iuh.edu.vn

Dinh-Khoi Hoang
Faculty of Electrical Engineering Technology
Industrial of University Ho Chi Minh City, Vietnam
hoangdinhkhoi@iuh.edu.vn

Trong-Toan Tran
Faculty of Electrical Engineering Technology
Industrial of University Ho Chi Minh City, Vietnam
trantrongtoan@iuh.edu.vn

Thanh-Thuan Nguyen
Faculty of Electrical Engineering Technology
Industrial of University Ho Chi Minh City, Vietnam
nguyenthanhthuan@iuh.edu.vn

Van-Tho Nguyen
Faculty of Electrical Engineering Technology
Industrial of University Ho Chi Minh City, Vietnam
nguyenvantho@iuh.edu.vn

Long-Ho Le
Faculty of Electrical Engineering Technology
Industrial of University Ho Chi Minh City, Vietnam
lelongho@iuh.edu.vn

Abstract

This paper introduces an effective intelligent controller for robotic systems with uncertainties. The proposed method is a novel self-organizing fuzzy cerebellar model articulation controller (NSOFC) which is a combination of a cerebellar model articulation controller (CMAC) and sliding mode control (SMC). We also present a new Gaussian membership function (GMF) that is designed by the combination of the prior and current GMF for each layer of CMAC. In addition, the relevant data of the prior GMF is used to check tracking errors more accurately. The inputs of the proposed controller can be mixed simultaneously between the prior and current states according to the corresponding errors. Moreover, the controller uses a self-organizing approach which can increase or decrease the number of layers, therefore the structures of NSOFC can be adjusted automatically. The proposed method consists of a NSOFC controller and a compensation controller. The NSOFC controller is used to estimate the ideal controller, and the compensation controller is used to eliminate the approximated error. The online parameters tuning law of NSOFC is designed based on Lyapunov's theory to ensure stability of the system. Finally, the experimental results of a 2 DOF robot arm are used to demonstrate the efficiency of the proposed controller.

Keywords: Self-organizing technique, cerebellar model articulation controller, manipulator system, cerebellar model articulation controller, Gaussian membership function.

1 Introduction

Robots has been used widely in many different applications such as automated manufacture, transportation or in daily services. Technology in controlling robotic systems has been heavily investigated in recent decades [1][2][3][4][5][6]. One of the challenging problems when designing controller for robotic systems is uncertainties which come from several aspects such as unknown parameters, unknown frictions, and external disturbances, etc. Although the researches mentioned above also consider uncertainties and noises in their works, the results presented are still mainly relied on the nominal parameters of the systems. However, in many practical applications, the parameters of the robots usually change from time to time due to the change of the momentum of the systems or different loads [7]. For example, a robot continuously moves an object between two stations will make the centrifugal of it changes between two different values [8]. There are also cases where the robot has to carry different types of objects due to customer's demands [9][10]. In general, we may know the nominal parameters of the system and the load themselves but we do not know how they will change when the system is operated. Not to mention, there are also different noises that affects the system when it is running, which then add more uncertainties into the dynamic model. In general, it is difficult to calculate accurately the mathematical model of the robotic system due to the reasons mentioned above and therefore designing the controller based on dynamic system model is not a trivial task. One of the main challenges here is how to learn the unknown parameters of the system and how to adapt when these parameters change. There are several methods have been researched to tackle this problem such as evolutionary algorithms, sliding mode control, robust control, neuron network, etc.

Evolutionary algorithms [28] (EAs) are one of the learning methods that mimicking biological evolution in nature. It is a well-known class of optimization algorithms to deal with black-box optimization problems. One of the main advantages is that EAs do not require any prior knowledge about the problem, but rather start with a randomly distributed population and update the population by reproducing offspring with unique operators. There are several methods have been studied such as genetic algorithm (GA), particle swam optimization (PSO), differential evolution (DE), opposition-based learning, etc. Many studies in this area focus on different searching methods to increase the convergence speed of the learning procedure. In [29], the author presents a modified opposition-based learning method that uses a beta distribution with a selection switching scheme and a partial dimension changing scheme to accelerate the convergence speed. In [30], M. Elsisi developed a learning method called future search algorithm (FSA), which is a heuristic algorithm that mimic the person's life to search for unknown parameters. The idea of FSA is based on the behavior of a person trying to search for his best life by imitating the successful persons. One of the advantages of FSA is that it can update the random initial and utilizes the local search between people and the global search between the histories optimal persons to achieve the best solutions. It is shown in the paper that FSA has low computational effort and high convergence speed compared to other heuristic algorithms such as GA, PSO. However, these methods are currently applied successfully only in theory and mainly focus on the learning procedure.

Taking a step further, M. Elsisi utilized lightning search algorithm (LSA) [31] to find the optimal parameters for a variable structure controller (VSC) that is used to control a nuclear reactor power system [32]. In general, the main problem when designing a VSC controller is the choice of the switching vector and the feedback gains. This can usually be done by using a searching algorithm which is LSA in this work. The author also incorporated Lyapunov function when designing the controller, such that ensures the stability of the system. The controller is then verified in simulation of controlling a nuclear reactor. The author showed that VSC based on LSA has better performance compared to VSC based on other GA which is the main purpose of this work. However, the convergence speed of LSA is still approximately 1.5 minutes on a rather high-end computer, which make it incapable of controlling fast systems such as motors or robotic systems.

With the same approach as VSC, the author in [33] takes advantage of a searching algorithm named multi-tracker optimization algorithm (MTOA) to properly select the parameters of a model predictive controller (MPC). MPC is a powerful control method that utilizes the past and current values of the output to predict the behavior of the system in the future. However, MPC needs to be properly configured by selecting the weight matrices of the objective function, the prediction horizon, control horizon and the sample time. An inappropriate selection may lead to less effective in controlling or less efficient in computation time. The proposed modified MTOA in this work is carried out based on opposition-based learning (OBL) and quasi OBL approaches. The proposed approach improves the exploration behavior of MTOA to prevent it from becoming trapped in a local minimum. The author then applied the MPC controller with the parameters are tuned by the proposed searching algorithm on a robotic manipulator to track different linear and nonlinear trajectories. The results show the improvement in term of tracking error compared to other approaches such as GA-based PID, original MTOA-based MPC, etc. However, the searching part in this work is still performed offline and the obtained parameters are then used to control the system after that.

Also using MPC-based controller to control autonomous vehicle under vision dynamic, the author in [34] comes up with a different searching approach based on AI-techniques to tune the parameters of the controller. The AI-based searching method is called social ski driver algorithm (SSDA) which imitates the behavior of the sky drivers' tracking during the downhill. SSDA combines the advantages of different AI techniques in literatures to overcome the trapping in local optima. Like previous approach, SSDA is used in this work to tune the parameters of a hybrid discrete-time Laguerre function MPC (DTLF-MPC) to improve the performance of the vehicle system and reduce the computational effort. The main results in the paper show that the convergence speed of the algorithm is fast enough to be calculated online, which is the main advantage of this work. However, as the searching method is based on AI-technique, there is no stability proof of the controller is provided.

Recently, many researches focus on intelligent control technologies such as Fuzzy Neural Network (FNN), Neuron Network (NN), Cerebellar Model Articulation Controller (CMAC), etc. The general requirement of these intelligent controllers is to reduce the impacts of unknown parameters and unstructured disturbances by utilizing the learning abilities of controlled networks without the need of knowing details about the system during the design phase. This is usually done by approximating a part of the nonlinear model of the system or the whole of it using the learning ability of the controllers [11]. Intelligent controllers has been used and applied successfully in many applications, especially in adaptive control [12][13][14][15]. In general, there are two commonly approaches that have been used the most in this area: first, the cerebellar model articulation controller is designed by combining the advantages of model-based controllers such as sliding mode, linear feedback controllers, etc and the advantages of FNN, NN and CMAC to approximate the unknown nonlinear equation in the dynamic model of the system [11], second, an intelligent-based approach is constructed to identify the complete mathematical model of the system. The advantage of the second method is that the dynamic model can be completely unknown but it requires a large amount of time for the algorithm to converge. Nevertheless, both approaches have been researched and applied successfully in many real-life application.

Generally, FNN, NN are able to approximate any nonlinear function with their learning ability. However, the convergence speed is rather slow. To overcome this problem, CMAC is proposed by Albus in [16] for identifying and controlling complex dynamic systems. CMAC has several advantages compare to the normal NN such as its fast learning speed, more general and can be implemented easily in different hardwares [17][18][19]. Chiang and Lin [20] and Chih-Min Lin; Hsin-Yi Li [21] continued to develop CMAC network with Gaussian and Wavelet basic functions. In their work, they analyze the convergence capability of the network and present different results to prove the efficiency of the method [17][18][19][20][21]. However, in most of the aforementioned researches about CMAC, the structure of it is established to be unchanged, hence it is difficult to calculate the optimal memory space for the learning speed of the network. If the memory space is too large, the learning speed will be decreased as it requires more computational effort from the hardware. Therefore, it is difficult to implement this method in online applications. Therefore, several researches [22][23][24] are proposed and successfully implemented CMAC network with self-organizing structure to reduce the size of the memory using different techniques such as group data, multilayer hierarchical CMAC model, Shannon's entropy measure and golden-section search method to increase or decrease number of layers of the network.

To increase the adaptive capability of CMAC network for online learning and reduce the computational effort when the system converges, this study proposes a novel self-organizing FCMAC called NSOFC for robotic systems with uncertainties. This proposed control scheme allows RCMAC in [22][25] have better local respond to determine the input values for RFBFs unit and the weight of the hypercubes. The approach uses the integrated sliding surface and the overlapping of prior and current GMFs at each layer to create a synthesis of two states for predicting the estimated errors. Then the self-organizing method is used to determine whether it is necessary to increase or decrease the number of current layers. The proposed control scheme is more general than the local feedback control scheme. Utilizing the recursive unit, it is possible to determine a new membership function, the input values for RFBFs and the weight values of hypercubes for CMAC. The adaptive PI is also employed for adjusting the adaptive parameters. Moreover, the Lyapunov function is used to ensure the stability of the system. Finally, the experimental results of the 2 DOF robot arm are presented to prove the efficiency of the proposed control scheme.

2 System description

In general, a dynamic equation of an n joint robotic system can be written as follow:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau \tag{1}$$

where q, $\dot{q}$, $\ddot{q} \in R^n$ are the joint position, velocity, and acceleration vectors respectively, $M(q) \in R^{n\times n}$ is the inertia matrix, $C(q,\dot{q})$ represents the centrifugal and Coriolis forces, $G(q) \in R^{n\times 1}$ represents the gravity vector, $\tau \in R^{n\times 1}$ is the applied control torque.

Equation (1) can be reformulated as:

$$\begin{aligned}\ddot{q} &= -M^{-1}(q)\left(C(q,\dot{q})\dot{q} + G(q)\right) + M^{-1}(q)\tau \\ &= f(x) + g(x)\tau\end{aligned} \tag{2}$$

$$f(x) = \begin{bmatrix} f_1(x) \\ f_2(2) \\ \vdots \\ f_n(x) \end{bmatrix} = -M^{-1}(q)\left(C(q,\dot{q})\dot{q} + G(q)\right) \in R^{n\times 1},$$

$$g(x) = \begin{bmatrix} g_{11}(x) & \cdots & g_{1n}(x) \\ \vdots & \ddots & \vdots \\ g_{n1}(x) & \cdots & g_{nn}(x) \end{bmatrix} = M^{-1}(q) \in R^{n\times n}.$$

where $f(x)$, $g(x)$ are nonlinear dynamic functions. These functions are usually difficult to obtain due to measurement errors, different loads, frictions, and external disturbances. To solve this problem, we assume that $f_0(x) \in R^{n\times 1}$, $g_0(x) \in R^{n\times n}$ are the nominal component of the real value of $f(x)$, $g(x)$ respectively. We also assume that $l(x)$ is the sum of the bounded and uncertain components.

Suppose that the system (1) is controllable and is inversible for all q.

Then equation (2) can be rewritten as:

$$\ddot{q}(t) = f_0(x) + g_0(x)\tau + l(x) \tag{3}$$

where $x = \begin{bmatrix} q^T & \dot{q}^T \end{bmatrix}^T$ is the joint position and velocity vectors respectively. In addition, tracking error can be defined as:

$$e = q_d - q \tag{4}$$

where $q_d = [q_{d1} \quad q_{d2} \cdots q_{dn}]^T \in R^n$ is the desired joint position. Then, the tracking error vector can be written as:

$$\bar{e} = \begin{bmatrix} e^T & \dot{e}^T \cdots e^{(n-1)T} \end{bmatrix}^T \in R^{m \times n} \tag{5}$$

Next, we define the sliding surface:

$$s = e^{n-1} + k_1 e^{n-2} + \quad \cdots \quad + k_n \int_0^t e(\tau) d\tau \tag{6}$$

where $s(t) = [s_1(t) \quad s_2(t) \cdots s_i(t) \cdots s_{n_i}(t)] \in R^{n_i}$, and $n_i = m$, $k_i = diag(k_{i1}, k_{i2}, \ldots, k_{im}) \in R^{m \times m}$ is a positive constant matrix.

If $f_0(x)$, $g_0(x)$ and $l(x)$ are known accurately, then the ideal controller can be obtained as below:

$$u_{id} = g_0^{-1}(x) \left[\ddot{q}_d - f_0(x) - l(x) + k^T e^T \right] \tag{7}$$

Substituting (7) into (2), we get the dynamic error equation:

$$\dot{s} = e^{(n)} + k^T \bar{e} = 0 \tag{8}$$

In (8), tracking error $\bar{e} \to 0$ when $t \to \infty$ if k is chosen appropriately according to Hurwitz stability criterion. However, $l(x)$ usually cannot be verified in practical applications. Thus, u_{id} in (7) cannot be identified.

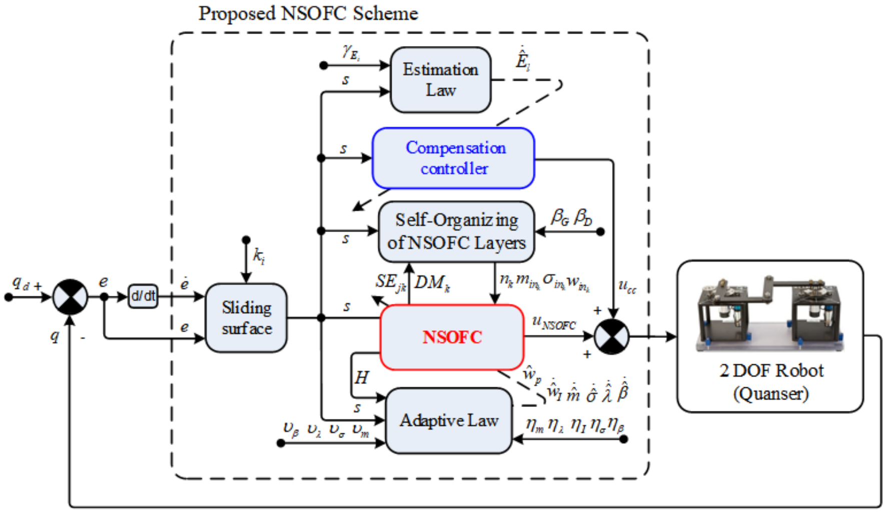
Proposed NSOFC Scheme
γ_{E_i}
s
Estimation Law
$\hat{\dot{E}}_i$
Compensation controller
Self-Organizing of NSOFC Layers
$\beta_G\ \beta_D$
k_i
q_d +
e
d/dt
$\dot{e}$
e
q
-
Sliding surface
SE_{jk} DM_k
$n_k\ m_{in_k}\ \sigma_{in_k}\ w_{in_k}$
u_{cc}
u_{NSOFC}
+
+
NSOFC
H
$\hat{\dot{w}}_p$
$\hat{\dot{w}}_l\ \hat{\dot{m}}\ \hat{\dot{\sigma}}\ \hat{\dot{\lambda}}\ \hat{\dot{\beta}}$
Adaptive Law
$\upsilon_\beta\ \upsilon_\lambda\ \upsilon_\sigma\ \upsilon_m$
$\eta_m\ \eta_\lambda\ \eta_l\ \eta_\sigma\ \eta_\beta$
2 DOF Robot
(Quanser)

Figure 1: Block diagram of NSOFC control system

It can be seen from Eq. (8) that the error will reach zero if the matrix K is selected such that all roots of the characteristic equation stay in the left half of the “s” plane. However, u_{id} in Eq. (7) cannot be determined in the real scenario since $l(x)$ is unknown. Therefore, we propose a new controller which consists of a NSOFC controller, and an adaptive compensation described as follow:

$$u = u_{NSOFC} + u_{cc} \tag{9}$$

where u_{NSOFC} is the main controller which approximates the ideal controller in (7) and the compensation controller is used to estimate the estimated error.

3 Design the NSOFC Controller

3.1 Fuzzy CMAC controller with overlapping Gaussian membership function

Both aforementioned CMAC [27] and RCMAC [26] controllers are only designed with one GMF activated at each layer. Our proposed controller uses two GMF for each layer instead. Table 1 shows the comparison between the local feedback RCMAC and the proposed controller. The proposed control scheme has two adaptive coefficients Λ_k and β_k, therefore it is more flexible for the control system to identify the input values of RFBFs and hypercubes. When a trajectory shifts from the current activating state to the next state, the prior GMF still being activated as well as the current GMF. The self-organizing technique is used to adjust the number of layers in FCMAC. If the error exceeds the threshold, a new layer will be created. The NSOFC structure uses overlapping GMF, presented in Fig. 2, which includes three continuous mappings and one output, can be written as follow:

$$\begin{cases} I \to A \quad (Input \to Associated\ memory) \\ A \to R \quad (Associated\ memory \to Receptive\ field) \\ R \to W \quad (Receptive\ field \to Network\ weight\ memory) \end{cases} \tag{10}$$

$$OutputO : O(I) = WH^{synthesis}, \tag{11}$$

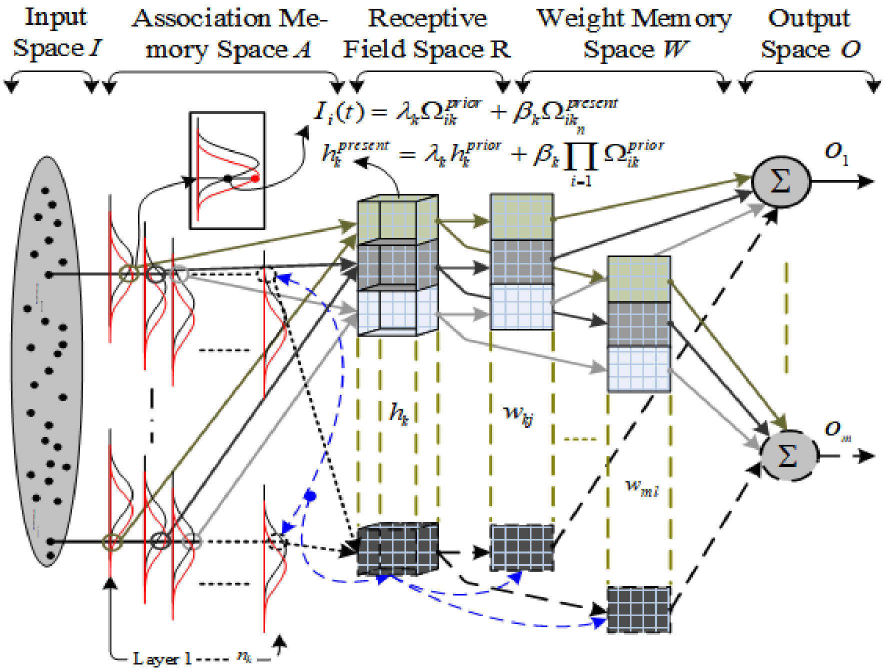

Figure 2: Structure of the proposed NSOFC with overlapping GMF

The following fuzzy control laws are used for the proposed controller:

$$\begin{gathered} R^{jk} : If\ I_1\ is\ \Omega_{1k}, I_2\ is\ \Omega_{2k}, \ldots,\ and\ I_{n_i}\ is\ \Omega_{n_ik}\ then \\ \hat{\text{w}}_{jk} = \upsilon_P\left(\hat{\text{w}}_{jkP}\right) + \upsilon_I\left(\hat{\text{w}}_{jkI}\right) \\ \text{for } j = 1, 2, \ldots, m \text{ and } k = 1, 2, \ldots, n_k. \end{gathered} \tag{12}$$

Input state variables $I = [I_1 \quad I_2 \quad \cdots \quad I_{ni}]^T$ are connected to association memory A only contains one block for each layer. Each variable is arranged in one block of each layer. At each block, there are two GMFs define one synthesis RFBF. These two GMFs, $\Omega_{ik}^{present}$ và Ω_{ik}^{prior} are presented in Fig. 3. The value of Ω_{ik}^{prior} is stored in memory, and the values of $\Omega_{ik}^{present}$, Ω_{ik}^{prior} are recursively calculated

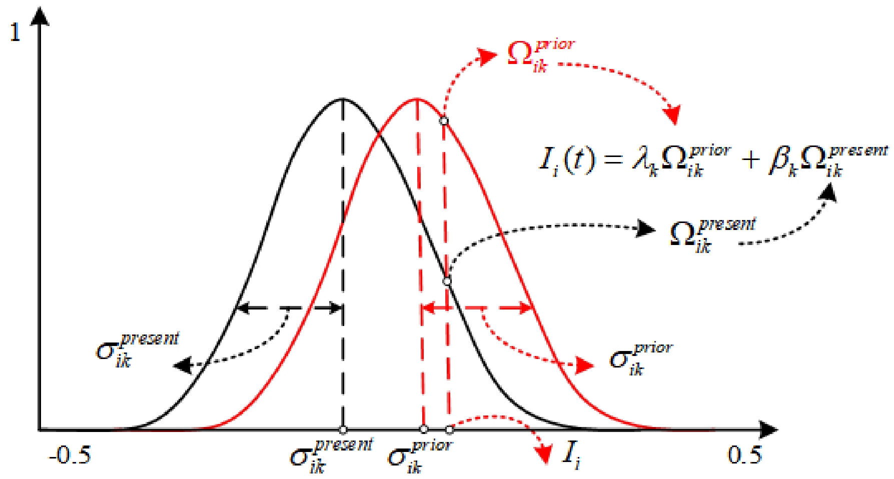

Figure 3: Two overlapping GMFs for calculation of the synthesis RFBF

Table 1: Comparison between the proposed controller and the local feedback RCMAC

	Input value of RFBF	λ_k	β_k
Local feedback RCMAC [27]	$I_i(t)=\lambda_k\Omega_{ik}^{prior}+\beta_k I_i=\lambda_k\Omega_{ik}^{prior}+I_i$	Online tuning	1
Proposed controller	$I_i(t)=\lambda_k\Omega_{ik}^{prior}+\beta_k\Omega_{ik}^{present}$	Online tuning	Online tuning

(a)

Value of hypercube block	
Local feedback RCMAC [27]	$H^{synthesis}=\prod_{i=1}^{n}\Omega_{ik}^{present}=\prod_{i=1}^{n}exp\left[-\frac{\left(I_i-m_{ik}^{present}\right)^2}{\left(\sigma_{ik}^{present}\right)^2}\right]$
Proposed controller	$H^{synthesis}=\lambda_k h_k^{prior}+\beta_k\prod_{i=1}^{n}exp\left[-\frac{\left(I_i-m_{ik}^{present}\right)^2}{\left(\sigma_{ik}^{present}\right)^2}\right]$

(b)

as follow:

$$\Omega_{ik}^{prior}(I_i)=exp\left[-\frac{(I_i-m_{ik}^{prior})^2}{(\sigma_{ik}^{prior})^2}\right],\quad i=1,2,\ldots,n_i,\quad k=1,2,\ldots,n_k \tag{13}$$

$$\Omega_{ik}^{present}(I_i)=exp\left[-\frac{(I_i-m_{ik}^{present})^2}{(\sigma_{ik}^{present})^2}\right],\quad i=1,2,\ldots,n_i,\quad k=1,2,\ldots,n_k \tag{14}$$

The input values of RFBF are defined as:

$$I_i(t)=\lambda_k\Omega_{ik}^{prior}+\beta_k\Omega_{ik}^{present} \tag{15}$$

where $\Omega_{ik}^{present}$ is the current RFBF, $m_{ik}^{present}$ is the mean value and $\sigma_{ik}^{present}$ is the current state variance of the k^{th} layer corresponding to the i^{th} input variable. Similarly, Ω_{ik}^{prior} is the prior RFBF, m_{ik}^{prior} is the prior mean value and σ_{ik}^{prior} is the prior state variance. λ_k and β_k are the adaptive estimation gains.

The prior state is stored in the memory as an estimation for the next state. Therefore, when the current state is activated, it will combine with the prior state to estimate the next state.

The example in Fig. 4 illustrates the 2D space, the output from the NSOFC is the sum values of the present (6, 5) and the prior (5,4) hypercubes. One synthesis hypercube is defined as below:

$$H^{synthesis}\equiv H=\lambda H^{prior}+\beta H^{present}=\lambda_k H_k^{prior}+\beta_k\prod_{i=1}^{n_i}\Omega_{ik}^{present} \tag{16}$$

Where

$\lambda=[\lambda_1,\ \lambda_2,\ \cdots\ ,\ \lambda_k,\ \cdots,\ \lambda_{n_k}]$ and $\beta=[\beta_1,\ \beta_2,\ \cdots\ ,\ \beta_k,\ \cdots,\ \beta_{n_k}]$ are the adaptive estimated weights of the prior and current hypercubes.

$H^{\text{prior}} = \prod_{i=1}^{n_i} \Omega_{\text{ik}}^{\text{prior}} = [h_1^{\text{prior}}\ \ h_2^{\text{prior}}\ \ \cdots\ \ h_k^{\text{prior}}\ \ \cdots\ \ h_{n_k}^{\text{prior}}]^T$ are the values of the prior hypercubes.

$H^{\text{present}} = \prod_{i=1}^{n_i} \Omega_{\text{ik}}^{\text{present}} = [h_1^{\text{present}}\ \ h_2^{\text{present}}\ \ \cdots\ \ h_k^{\text{present}}\ \ \cdots\ \ h_{n_k}^{\text{present}}]^T$ are the values of the current hypercubes.

$H = \lambda H^{\text{prior}} + \beta H^{\text{present}} = [\ h_1,\ h_1,\ h_2,\ ...,\ h_k,\ \ ...,\ h_{n_k}]\ ^T \in \mathcal{R}^{n^k}$ are the synthesis values of the prior and current hypercubes.

For NSOFC, the weight gain w and hypercube h are defined as $v_{\text{jk}} = \text{w}_{\text{jk}} h_k$, where w_{jk} is the weight of the k^{th} hypercube respectively with the j^{th} output.

The j^{th} output is defined as:

$$u = O = W^T H = [\text{w}_1\ \ \text{w}_2\ \ \cdots\ \ \text{w}_j\ \ \cdots\ \ \text{w}_m]^T \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_{n_k} \end{bmatrix} \tag{17}$$

$$u = O_j = W^T \left[\lambda \text{H}^{\text{prior}} + \beta H^{\text{present}}\right],\ j = 1,\ 2,\ \cdots,\ m \tag{18}$$

Where $W = [\text{w}_1\ \ \text{w}_2\ \ \cdots\ \ \text{w}_j\ \ \cdots\ \ \text{w}_m]$, w_j= $[\text{w}_{1j}\ \ \text{w}_{2j}\ \ \cdots\ \ \text{w}_{n_k j}]\ ^T$.

Design an optimal controller u^*:

$$u_{\text{id}} = u^* + \Delta\ \ =\ \ W^{*T} H^* + \Delta \tag{19}$$

Where Δ is the approximated error and W^*, H^*, m^*, σ^* represent the optimal parameter matrices and vectors of W, H, m, σ respectively. In practice, these optimal matrices and vectors are difficult to verify the values which approximate to ideal controller. Therefore,$\widehat{u}$ is estimated online to define u^*.

The estimated function $\widehat{u}$ is defined as:

$$\widehat{u} = \widehat{W}^T \widehat{H} \tag{20}$$

Where $\widehat{W}$and $\widehat{H}$ are the estimated parameter matrix and vector of W^*, H^* respectively, and $\widehat{m}$, $\widehat{\sigma}$, $\widehat{\lambda}$, $\widehat{\beta}$ are the estimated parameter vectors of m^*, σ^*, λ^* và β^* respectively.

The estimation error is defined as:

$$\widetilde{u} = u_{\text{id}} - \widehat{u}_{\text{NSOFC}} = W^{*T} H^* - \widehat{W}^T \widehat{H} + \Delta = \widetilde{W}^T \widetilde{H} + \widehat{W}^T \widetilde{H} + \widetilde{W}^T \widehat{H} + \Delta \tag{21}$$

Where $\widetilde{W} = W^* - \widehat{W}$ and $\widetilde{H} = H^* - \widehat{H}$. Here Taylor expansion is used to convert nonlinear functions into linear functions [12]:

$$\widetilde{H} = H_m \widetilde{m} + H_\sigma \widetilde{\sigma} + H_\lambda \widetilde{\lambda} + H_\beta \widetilde{\beta} + \Theta \tag{22}$$

Where $\widetilde{m} = m^* - \widehat{m},\ \widetilde{\sigma} = \sigma^* - \widehat{\sigma},\ \widetilde{\lambda} = \lambda^* - \widehat{\lambda},\ \widetilde{\beta} = \beta^* - \widehat{\beta},\ \Theta$ is a high-order vector.

$$H_m = \left[\ \frac{\partial h_1}{\partial m}\ \frac{\partial h_2}{\partial m} \cdots\ \frac{\partial h_{n_k}}{\partial m}\right]^T \Big|_{\cdot m = \widehat{m}}\ ,\ H_\sigma = \left[\ \frac{\partial h_1}{\partial \sigma}\ \frac{\partial h_2}{\partial \sigma}\ ...\ \frac{\partial h_{n_k}}{\partial \sigma}\right]^T \Big|_{\cdot \sigma = \widehat{\sigma}}\ ,$$

$$H_\lambda = \left[\ \frac{\partial h_1}{\partial \lambda}\ \frac{\partial h_2}{\partial \lambda} \cdots\ \frac{\partial h_{n_k}}{\partial \lambda}\right]^T \Big|_{\cdot \lambda = \widehat{\lambda}}\ ,\ H_\beta = \left[\ \frac{\partial h_1}{\partial \beta}\ \frac{\partial h_2}{\partial \beta}\ ...\ \frac{\partial h_{n_k}}{\partial \beta}\right]^T \Big|_{\cdot \beta = \widehat{\beta}}\ ,$$

Substitute (22) into (21), we get:

$$\begin{aligned} \widetilde{u} &= \widetilde{W}^T \widetilde{H} + \widehat{W}^T \left(H_m \widetilde{m} + H_\sigma \widetilde{\sigma} + H_\lambda \widetilde{\lambda} + H_\beta \widetilde{\beta} + \Theta\right) + \widetilde{W}^T \widehat{H} + \Delta \\ &= \widetilde{W}^T \widehat{H} + \widetilde{m}^T H_m^T \widehat{W} + \widetilde{\sigma}^T H_\sigma^T \widehat{W} + \widetilde{\lambda}^T H_\lambda^T \widehat{W} + \widetilde{\beta}^T H_\beta^T \widehat{W} + \widehat{W}^T \Phi + \widetilde{W}^T \widetilde{H} + \Delta \end{aligned} \tag{23}$$

To increase the convergence of the output weight, W^* is divided into two parts [12]:

$$W^* = v_p \text{w}_p^* + v_i \text{w}_i^* \tag{24}$$

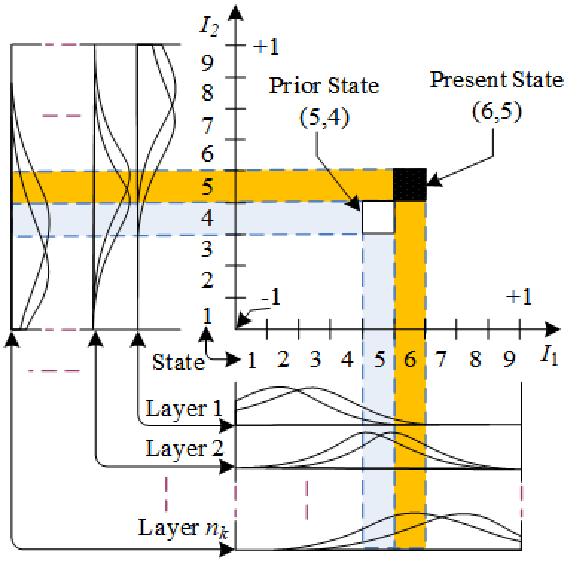

Figure 4: 2D structure of NSOFC with the overlapping GMFs

Where w_p^* and w_i^* represents the proportional element and integral element of W^* and $\mathrm{w}_i^* = \int_0^t \mathrm{w}_p^* d\tau$ respectively, v_p and v_i are positive constants. Similarly, $\widehat{W}$ is also divided into two parts

$$\widehat{W} = v_P \hat{\mathrm{w}}_P + v_I \hat{\mathrm{w}}_I \tag{25}$$

Where $\hat{\mathrm{w}}_P$ and $\hat{\mathrm{w}}_I$ are the proportional and integral elements of $\widehat{W}$, $\hat{\mathrm{w}}_I = \int_0^t \hat{\mathrm{w}}_P d\tau$. Therefore, $\widetilde{W}$ is defined as follow:

$$\widetilde{W} = v_I \tilde{\mathrm{w}}_I - v_P \hat{\mathrm{w}}_P + v_P \mathrm{w}_P^* \tag{26}$$

Where $\tilde{\mathrm{w}}_I$ = w_i*- $\hat{\mathrm{w}}_I$. Substitute (26) into (23), we get:

$$\begin{aligned}\tilde{u} &= (v_I \tilde{\mathrm{w}}_I - v_P \hat{\mathrm{w}}_P + v_P \mathrm{w}_P^*)^T \hat{H} + \tilde{m}^T H_m^T \widehat{W} + \tilde{\sigma}^T H_\sigma^T \widehat{W} + \tilde{\lambda}^T H_\lambda^T \widehat{W} + \tilde{\beta}^T H_\beta^T \widehat{W} + \widehat{W}^T \Theta + \widetilde{W}^T \tilde{H} + \Delta \\ &= v_I \tilde{\mathrm{w}}_I^T \hat{H} - v_P \hat{\mathrm{w}}_P^T \hat{H} + \tilde{m}^T H_m^T \widehat{W} + \tilde{\sigma}^T H_\sigma^T \widehat{W} + \tilde{\lambda}^T H_\lambda^T \widehat{W} + \tilde{\beta}^T H_\beta^T \widehat{W} + \varepsilon\end{aligned} \tag{27}$$

Where the uncertain coefficient $\varepsilon = v_P v_p^{*T} \hat{H} + \widehat{W}^T \Theta + \widetilde{W}^T \tilde{H} + \Delta$ is the approximation of total errors, bounded by $0 \leq \|\varepsilon\| \leq E_l$, where E_l is a positive constant.

3.2 Self-organizing structure

The online self-learning for structure and parameters of NSOFC is proposed as in Fig. 1. The first task of structured learning is to verify whether it is necessary to add a new layer into the associated memory A and create hypercubes with its weight memory. If the current inputs are in the range of the defined data, then NSOFC does not create a new layer but updates the parameters for the current laws.

Mean distance, DM_k, is defined as:

$$\mathrm{DM}_k = \|I - m_k\|_2 \tag{28}$$

Where $m_k = [\ m_{1k},\ ...,\ m_{ik},\ \ ...,\ m_{n_\mathrm{i}\ \mathrm{k}}\ ]^T$

To check whether a new layer need to be added [20]:

$$\widehat{k} = arg \min_{1 \leq k \leq n_k} \mathrm{DM}_k \tag{29}$$

If $\max_i\ \mathrm{DM}_{\widehat{k}} > \beta_G$ then a new layer is created, where β_G is the pre-defined threshold.

If the distance between input data and the mean values are bigger than the threshold, it means the current values of the current GMF is too small, then a new layer is added. The layer number is increased as follow:

$$n_k(t+1) = n_k(t) + 1 \tag{30}$$

Where $n_k(t)$ is the current layer number at t.

The output weight values are randomly generated for a new layer, the mean value, and the variance of GMFs are defined as:

$$m_{\text{in}_k} = I_i \tag{31}$$

$$\sigma_{\text{in}_k} = \sigma_{\text{int}} \tag{32}$$

$$\mathrm{w}_{\text{in}_k} = \delta_r \tag{33}$$

Where I_i is the new input, σ_{int} is the pre-defined constant, and δ_r is a random constant.

To learn the structure, any unnecessary layer must be deleted. To decrease the layer number [20], the ratio of the k^{th} element of j^{th} output is defined as:

$$\text{SE}_{\text{jk}} = v_{\text{jk}}/O_j \tag{34}$$

Then the maximum ratio of the j^{th} output respective to the minimum element is computed as:

$$\widetilde{k} = arg \min_{1 \le k \le n_k} \max_{1 \le j \le m} \text{SE}_{\text{jk}} \tag{35}$$

$$\begin{aligned} &\text{If } \text{SE}_{\text{jk}} \le \beta_D \\ &\text{Then the } k^{\text{th}} \text{ layer must be deleted, where } \beta_D \text{ is the pre-defined threshold.} \end{aligned} \tag{36}$$

The function (36) shows that if the element of the current layer is smaller than the threshold then it will be deleted.

3.3 Adaptive NSOFC control system

The NSOFC system uses adaptive PI algorithm which is showed in Fig. 1. The control law (9) is substituted into (3). Using (6) and (7), then the sum errors function become:

$$\dot{s} = e^{(n)} + k^T \overline{e} = g_0(u_{\text{id}} - \widehat{u}_{\text{NSOFC}} - u_{\text{cc}}) \tag{37}$$

Using (27), then (37) is rewritten as:

$$\dot{s} = g_0(v_I \widetilde{\mathrm{w}}_I^T \widehat{H} - v_P \widehat{\mathrm{w}}_P^T \widehat{H} + \widetilde{m}^T H_m^T \widehat{W} + \widetilde{\sigma}^T H_\sigma^T \widehat{W} + \widetilde{\lambda}^T H_\lambda^T \widehat{W} + \widetilde{\beta}^T H_\beta^T \widehat{W} + \varepsilon - u_{\text{cc}}) \tag{38}$$

To ensure the stability of the control system, a Lyapunov function is chosen as:

$$V = \frac{1}{2} s^T g_0^{-1} s + \frac{v_I}{2} \text{tr}\left(\widetilde{\mathrm{w}}_I^T \widetilde{\mathrm{w}}_I\right) + \frac{1}{2v_m} \widetilde{m}^T \widetilde{m} + \frac{1}{2v_\sigma} \widetilde{\sigma}^T \widetilde{\sigma} + \frac{1}{2v_\lambda} \widetilde{\lambda}^T \widetilde{\lambda} + \frac{1}{2v_\beta} \widetilde{\beta}^T \widetilde{\beta} + \frac{1}{2v_{D_e}} \widetilde{E}_l^2 \tag{39}$$

Where v_I, v_p, v_m, v_σ, v_λ, v_β and E_l are the positive constants with the role of learning rate. The uncertain bound of E_l cannot be measured in practice, therefore the estimated bound $\widehat{E}_l$ is used. The error bound is computed as: $\widetilde{E}_l = E_l - \widehat{E}_l$.

Differentiate equation (39), and using (38), we get:

$$\begin{aligned} \dot{V} &= s^T g_0^{-1} \dot{s} - v_I \text{tr}\left(\widetilde{\mathrm{w}}_I^T \dot{\widehat{\mathrm{w}}}_I\right) - \frac{1}{v_m} \widetilde{m}^T \dot{\widehat{m}} - \frac{1}{v_\sigma} \widetilde{\sigma}^T \dot{\widehat{\sigma}} - \frac{1}{v_\lambda} \widetilde{\lambda}^T \dot{\widehat{\lambda}} - \frac{1}{v_\beta} \widetilde{\beta}^T \dot{\widehat{\beta}} - \frac{1}{v_{D_e}} \widetilde{E}_l \dot{\widehat{E}}_l \\ &= v_I \sum_{j=1}^{m} \widetilde{\mathrm{w}}_{\text{jI}}^T (s_j \widehat{H} - \dot{\widehat{\mathrm{w}}}_{\text{jI}}) + \widetilde{m}^T \left(s^T H_m^T \widehat{W} - \frac{\dot{\widehat{m}}}{v_m}\right) + \widetilde{\sigma}^T \left(s^T H_\sigma^T \widehat{W} - \frac{\dot{\widehat{\sigma}}}{v_\sigma}\right) \\ &+ \widetilde{\lambda}^T \left(s^T H_\lambda^T \widehat{W} - \frac{\dot{\widehat{\lambda}}}{v_\lambda}\right) + \widetilde{\beta}^T \left(s^T H_\beta^T \widehat{W} - \frac{\dot{\widehat{\beta}}}{v_\beta}\right) - v_P \sum_{j=1}^{m} \widehat{\mathrm{w}}_{\text{jP}}^T s_j \widehat{H} + s^T (\varepsilon - u_{\text{cc}}) - \frac{1}{v_{E_l}} \widetilde{E}_l \dot{\widehat{E}}_l \end{aligned} \tag{40}$$

The functions $v_I \text{tr}\left(\widetilde{\mathrm{w}}_I^T \dot{\widehat{\mathrm{w}}}_I\right) = v_I \sum_{j=1}^{m} \widetilde{\mathrm{w}}_{\text{jI}}^T \widetilde{W}_{\text{jI}}$ and $s^T \widetilde{\mathrm{w}}_I^T \widehat{H} = \sum_{j=1}^{m} s_j \widetilde{\mathrm{w}}_{\text{jI}}^T \widehat{H}$ are used.

Then the adaptive laws are written as follow:

$$\widehat{\mathrm{w}}_{\mathrm{jP}} = s_j\widehat{H} - \eta_I(\widehat{\mathrm{w}}_{\mathrm{jI}} - \mathrm{w}_0) \tag{41}$$

$$\dot{\widehat{\mathrm{w}}}_{\mathrm{jI}} = s_j\widehat{H} - \eta_I(\widehat{\mathrm{w}}_{\mathrm{jI}} - \mathrm{w}_0) \tag{42}$$

$$\dot{\widehat{m}} = \upsilon_m[s^T H_m^T \widehat{W} - \eta_m(\widehat{m} - m_0)] \tag{43}$$

$$\dot{\widehat{\sigma}} = \upsilon_\sigma[s^T H_\sigma^T \widehat{W} - \eta_\sigma(\widehat{\sigma} - \sigma_0)] \tag{44}$$

$$\dot{\widehat{\lambda}} = \upsilon_\lambda[s^T H_\lambda^T \widehat{W} - \eta_\lambda(\widehat{\lambda} - \lambda_0)] \tag{45}$$

$$\dot{\widehat{\beta}} = \upsilon_\beta[s^T H_\beta^T \widehat{W} - \eta_\beta(\widehat{\beta} - \beta_0)] \tag{46}$$

Where η_I, η_m, η_σ, η_λ and η_β are small positive constants; w_0, m_0, σ_0, λ_0, and β_0 are the initial estimation vectors of W^*, m^*, σ^*, λ^*, and β^* respectively, then the compensation controller is written as:

$$u_{\mathrm{cc}} = \widehat{E}_l tanh\left(\frac{s}{\xi}\right) \tag{47}$$

The estimation law for the bound:

$$\dot{\widehat{E}}_l = \upsilon_{E_l}\left[s\tanh\left(\frac{s}{\xi}\right) - \eta_{E_l}(\widehat{E}_l - \widehat{E}_{l0})\right] \tag{48}$$

Where tanh(.) is a hyperpol tengent function, E_{l0} is the initial value of E_l , ξ and η_{E_l} are the small positive constants. Using (41) – (48), (40) becomes:

$$\begin{aligned}
\dot{V} &= \sum_{j=1}^{m} \widetilde{\mathrm{w}}_{\mathrm{jI}}^T \upsilon_I \eta_I\left(\widehat{\mathrm{w}}_{\mathrm{jI}} - \mathrm{w}_0\right) + \widetilde{m}^T \eta_m\left(\widehat{m} - m_0\right) + \widetilde{\sigma}^T \eta_\sigma\left(\widehat{\sigma} - \sigma_0\right) + \widetilde{\lambda}^T \eta_\lambda\left(\widehat{\lambda} - \lambda_0\right) + \widetilde{\beta}^T \eta_\beta\left(\widehat{\beta} - \beta_0\right) \\
&- \upsilon_P \sum_{j=1}^{m} \widehat{\mathrm{w}}_{\mathrm{jP}}^T\left[\widehat{\mathrm{w}}_{\mathrm{jP}} + \eta_I\left(\widehat{\mathrm{w}}_{\mathrm{jI}} - \mathrm{w}_0\right)\right] + s\varepsilon - s\widehat{E}_l \tanh\left(\frac{s}{\xi}\right) - \frac{1}{\upsilon_{E_l}}\widetilde{E}_l \upsilon_{E_l}\left[s\tanh\left(\frac{s}{\xi}\right) - \eta_{E_l}\left(\widehat{E}_l - \widehat{E}_{l0}\right)\right] \\
&\le \sum_{j=1}^{m} \widetilde{\mathrm{w}}_{\mathrm{jI}}^T \upsilon_I \eta_I\left(\widehat{\mathrm{w}}_{\mathrm{jI}} - \mathrm{w}_0\right) + \widetilde{m}^T \eta_m\left(\widehat{m} - m_0\right) + \widetilde{\sigma}^T \eta_\sigma\left(\widehat{\sigma} - \sigma_0\right) + \widetilde{\lambda}^T \eta_\lambda\left(\widehat{\lambda} - \lambda_0\right) + \widetilde{\beta}^T \eta_\beta\left(\widehat{\beta} - \beta_0\right) \\
&- \sum_{j=1}^{m} \widehat{\mathrm{w}}_{\mathrm{jP}}^T \upsilon_P \eta_I\left(\widehat{\mathrm{w}}_{\mathrm{jI}} - \mathrm{w}_0\right) + \left|s^T\right| E_l - s^T\widehat{E}_l \tanh\left(\frac{s}{\xi}\right) - \widetilde{E}_l\left[s\tanh\left(\frac{s}{\xi}\right) - \eta_{E_l}\left(\widehat{E}_l - \widehat{E}_{l0}\right)\right]
\end{aligned} \tag{49}$$

With any $\xi > 0$:

$$0 \le \left|s^T\right| - s^T \tanh\left(\frac{s}{\xi}\right) \le \kappa\xi \tag{50}$$

Where K is a positive constant which satisfies K = exp(-(K + 1)) [13].
Using (50), (49) becomes:

$$\begin{aligned}
\dot{V} &= \sum_{j=1}^{m} \widetilde{\mathrm{w}}_{\mathrm{jI}}^T \upsilon_I \eta_I\left(\widehat{\mathrm{w}}_{\mathrm{jI}} - \mathrm{w}_0\right) + \widetilde{m}^T \eta_m\left(\widehat{m} - m_0\right) + \widetilde{\sigma}^T \eta_\sigma\left(\widehat{\sigma} - \sigma_0\right) + \widetilde{\lambda}^T \eta_\lambda\left(\widehat{\lambda} - \lambda_0\right) + \widetilde{\beta}^T \eta_\beta\left(\widehat{\beta} - \beta_0\right) \\
&- \sum_{j=1}^{m} \widehat{\mathrm{w}}_{\mathrm{jP}}^T \upsilon_P \eta_I\left(\widehat{w}_{\mathrm{jI}} - \mathrm{w}_0\right) + E_l\kappa\xi + \widetilde{E}_l\left[\eta_{E_l}\left(\widehat{E}_l - \widehat{E}_{l0}\right)\right] \\
&\le -\frac{1}{2}\upsilon_I\eta_I\Xi_{\mathrm{w}_I} - \frac{1}{2}\eta_m\Xi_m - \frac{1}{2}\eta_\sigma\Xi_\sigma - \frac{1}{2}\eta_\lambda\Xi_\lambda - \frac{1}{2}\eta_\beta\Xi_\beta - \frac{1}{2}\upsilon_P\eta_I\Xi_{\mathrm{w}_P} - \frac{1}{2}\eta_{E_l}\Xi_{E_l} + E_l\kappa\xi
\end{aligned} \tag{51}$$

Where

$$\Xi_{\mathrm{w}_I} = \sum_{j=1}^{m} \left[\left\| \mathrm{w}_{\mathrm{jI}}^{*} - \widehat{\mathrm{w}}_{\mathrm{jI}} \right\|^2 + \left\| \mathrm{w}_{\mathrm{jI}}^{*} - \mathrm{w}_0 \right\|^2 - \left\| \widehat{\mathrm{w}}_{\mathrm{jI}} - \mathrm{w}_0 \right\|^2 \right] \tag{52}$$

$$\Xi_m = \left[\left\| m^* - \widehat{m} \right\|^2 + \left\| m^* - m_0 \right\|^2 - \left\| \widehat{m} - m_0 \right\|^2 \right] \tag{53}$$

$$\Xi_\sigma = \left[\left\| \sigma^* - \widehat{\sigma} \right\|^2 + \left\| \sigma^* - \sigma_0 \right\|^2 - \left\| \widehat{\sigma} - \sigma_0 \right\|^2 \right] \tag{54}$$

$$\Xi_\lambda = \left[\left\| \lambda^* - \widehat{\lambda} \right\|^2 + \left\| \lambda^* - \lambda_0 \right\|^2 - \left\| \widehat{\lambda} - \lambda_0 \right\|^2 \right] \tag{55}$$

$$\Xi_\beta = \left[\left\| \beta^* - \widehat{\beta} \right\|^2 + \left\| \beta^* - \beta_0 \right\|^2 - \left\| \widehat{\beta} - \beta_0 \right\|^2 \right] \tag{56}$$

$$\Xi_{\mathrm{w}_P} = \sum_{j=1}^{m} \left[\left\| \mathrm{w}_{\mathrm{jP}}^{*} - \widehat{\mathrm{w}}_{\mathrm{jI}} \right\|^2 + \left\| \mathrm{w}_{\mathrm{jP}}^{*} - \mathrm{w}_0 \right\|^2 - \left\| \widehat{\mathrm{w}}_{\mathrm{jP}} - \widehat{\mathrm{w}}_{\mathrm{jI}} \right\|^2 - \left\| \widehat{\mathrm{w}}_{\mathrm{jP}} - \mathrm{w}_0 \right\|^2 \right] \tag{57}$$

$$\Xi_{E_l} = \left[\left| E_l - \widehat{E}_l \right|^2 + \left| E_l - E_{l0} \right|^2 - \left| \widehat{E}_l - E_{l0} \right|^2 \right] \tag{58}$$

Then (51) is rewritten:

$$\begin{aligned} \dot{V} \le & -\frac{1}{2} \left[\upsilon_I \eta_I \sum_{j=1}^{m} \left\| \widetilde{\mathrm{w}}_{\mathrm{jI}} \right\|^2 + \eta_m \left\| \widetilde{m} \right\|^2 + \eta_\sigma \left\| \widetilde{\sigma} \right\|^2 + \eta_\lambda \left\| \widetilde{\lambda} \right\|^2 + \eta_\beta \left\| \widetilde{\beta} \right\|^2 + \eta_{E_l} \left| \widetilde{E}_l \right|^2 \right] \\ & + \frac{1}{2} \left[\upsilon_I \eta_I \sum_{j=1}^{m} \left\| \widehat{\mathrm{w}}_{\mathrm{jI}} - \mathrm{w}_0 \right\|^2 + \eta_m \left\| \widehat{m} - m_0 \right\|^2 + \eta_\sigma \left\| \widehat{\sigma} - \sigma_0 \right\|^2 + \eta_\lambda \left\| \widehat{\lambda} - \lambda_0 \right\|^2 + \eta_\beta \left\| \widehat{\beta} - \beta_0 \right\|^2 \right] \\ & + \upsilon_P \eta_I \sum_{j=1}^{m} \left(\left\| \widetilde{\mathrm{w}}_{\mathrm{jP}} - \widehat{\mathrm{w}}_{\mathrm{jI}} \right\|^2 + \left\| \widetilde{\mathrm{w}}_{\mathrm{jP}} - \mathrm{w}_0 \right\|^2 \right) + \eta_{E_l} \left| E_l - E_{l0} \right|^2 + E_l \kappa \xi \end{aligned} \tag{59}$$

Where $\|\ .\|$ is Eulicdean norm. Using (39), (59) becomes:

$$\dot{V} \le -\rho_1 V + \rho_2 \tag{60}$$

Where ρ_1 and ρ_2 are the constants defined as:

$$\rho_1 = min(\eta_I,\ \eta_m \upsilon_m, \eta_\sigma \upsilon_\sigma, \eta_\lambda \upsilon_\lambda, \eta_\beta \upsilon_\beta, \eta_{E_l} \upsilon_{E_l}) \tag{61}$$

$$\rho_2 = \frac{1}{2} \left[\begin{array}{c} \upsilon_I \eta_I \sum_{j=1}^{m} \left\| \widehat{\mathrm{w}}_{\mathrm{jI}} - \mathrm{w}_0 \right\|^2 + \eta_m \left\| \widehat{m} - m_0 \right\|^2 + \eta_\sigma \left\| \widehat{\sigma} - \sigma_0 \right\|^2 + \eta_\lambda \left\| \widehat{\lambda} - \lambda_0 \right\|^2 + \eta_\beta \left\| \widehat{\beta} - \beta_0 \right\|^2 \\ + \upsilon_P \eta_I \sum_{j=1}^{m} \left(\left\| \widetilde{\mathrm{w}}_{\mathrm{jP}} - \widehat{\mathrm{w}}_{\mathrm{jI}} \right\|^2 + \left\| \widetilde{\mathrm{w}}_{\mathrm{jP}} - \mathrm{w}_0 \right\|^2 \right) + \eta_{E_l} \left| E_l - E_{l0} \right|^2 + \rho_1 s^2 \end{array} \right] + E_l \kappa \xi \tag{62}$$

If (62) satisfies:

$$0 \le V(t) \le \Psi + (V(0) - \Psi)\, exp(-\rho_1 t) \tag{63}$$

Where $\Psi = \frac{\rho_2}{\rho_1} > 0$, then e, w_I, σ, λ, $\beta$and E_l steadily bounded. Using (39) and (60) and given any $\xi > 2\Psi$, if T exists with any $t \ge T$, then error satisfied:

$$|e(t)| \le \xi \tag{64}$$

Thus, the proposed NSOFC is stable.

4 Experimental results

The real model of the robot is presented in Fig. 5 while it is simplified as a 2D model in Fig. 6 to obtain the dynamic equation of the system. The two servos are located at joint A and B with the respective control angles as θ_A and θ_B. The four arms of the robot have the same length L_b and are linked to each other. The end-effector is located at joint E. The positions of joint A and B in the

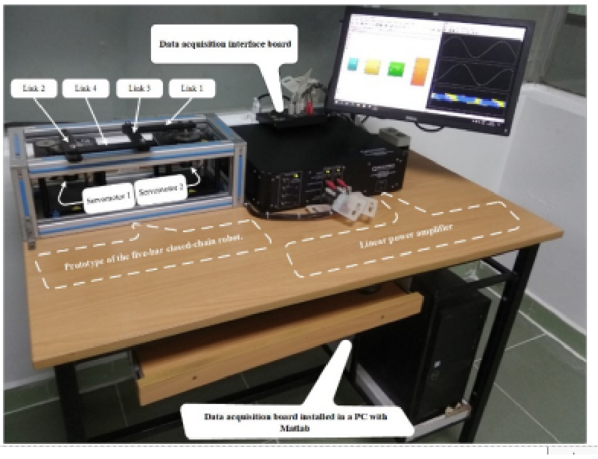
Data acquisition interface board
Link 2
Link 4
Link 3
Link 1
Servomotor 1
Servomotor 2
Prototype of the five-bar closed-chain robot.
Linear power amplifier
Data acquisition board installed in a PC with Matlab

Figure 5: Quanser 2-DOF robot model

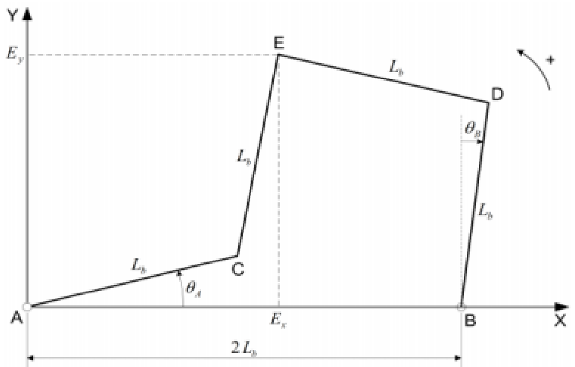

Figure 6: Quanser robot model in XY plane

current X-Y coordinate system are (0,0) and ($2L_b$,0) respectively. In this experiment, the end-effector of the robot is controlled to follow a circular trajectory defined as

$$\begin{cases} E_{\text{xd}} = Rcos(\omega) \\ E_{\text{yd}} = Rsin(\omega) \end{cases}$$

with R $= 0.75$, $\omega = t$ (rad/s). The starting position of the robot is $E_0 = \begin{bmatrix} L_b & L_b \end{bmatrix}^T$ with $\theta_0 = \begin{bmatrix} 0 & 0 \end{bmatrix}^T$. The WFCMAC controller has the form: $\tau = \tau_{\text{WFCMAC}} + \tau_{\text{ASIFC}}$ and the parameters of the two controllers are chosen as in Table 2. The system uses an NI PCIe-6351 card to collect sensor data and control the robot.

The experimental results of the two controllers are presented in Fig. 7, 8, 9. The results of WFCMAC (a) are on the left side and the results of NSOSC (b) are on the right side. Position and tracking errors of the two joints are illustrated in Fig. 7, the end-effector's position, tracking errors are illustrated in Fig. 8. The control efforts are illustrated in Fig. 9.

Moreover, the tracking errors of the end-effector of the two controllers are defined base on the error standards of IAE, ISE, ITAE, ITSE and ISTE, are also presented in Table 3.

The proposed controller presents the high accurate tracking ability, tracking speed, and quickly eliminate error which approaching to 0 even the system is an uncertain one. Here we use the IAE, ISE, ITAE, ITSE and ISTE error standards to compare the two controllers. The result is showed in Table 3. It is showed that the proposed NSOFC has the data much smaller than the data of the WFCMAC controller. It proves that the proposed controller has higher efficiency than the conventional CMAC controller. It also proves that the modification of the number of layer help flexibly change for well adaptive ability of the control system even the external disturbances existing in the control plant.

5 Conclusion and future work

This paper proposes a novel NSOFC controller for controlling uncertain robot arm systems. In the proposed NSOFC, the dynamic of the control system is not needed. The online adjusting law of adaptive parameters of the NSOFC and the compensation controller is defined based on Lyapunov stability theorem to ensure the stability of the system. This paper not only develops successfully the NSOFC control system for an uncertain non-linear system, but also derives the online self-organization algorithm for control the layer number of the NSOFC controller to reduce the memory space as much as possible. Finally, the experimental results prove that the proposed controller is possible to track the given trajectories. This controller can also be applied for the other uncertain n DOF robots, even uncertain non-linear systems. For future work, the control scheme can be extended to more complex robotic systems such as articulated robots with higher degree of freedom. In addition, the learning rules of the parameters of NSOFC still need to be further improved to reduce the "chattering" effect

Table 2: The parameters comparison between two controllers: WFCMAC and NSOFC

	WFCMAC controller	**NSOFC controller**
Main controller	*- Layer number:* $n_k = 5$ *(fixed)* *- Block number:* $n_j = 11$ *- Sliding surface:* $s(t) = \dot{e}(t) + ke(t)$, $k = 100I_2$ *- Initial mean and variance of GMF:* $m_i = [-1 - 0.8 - 0.6\ \cdots 0.6\ 0.8\ 1]$ $\sigma_i = 0.15$ *- Initial output weight:* $w_0 = 0.05$ *- Learning rate:* $\eta = 0.001, v = 0.01$	*- Layer number:* $n_k = 5$ *(can be changed)* *- Sliding surface:* $s(t) = \dot{e}(t) + ke(t)$, $k = 100I_2$ *- Initial mean and variance of GMF:* $m_i^{\text{prior}} = [-2 - 1\ 0\ 1\ 2]$ $m_i^{\text{present}} = [-2 - 1\ 0\ 1\ 2]$ $\sigma_i^{\text{prior}} = 0.15$ $\sigma_i^{\text{present}} = 0.15$ *- Initial output weight:* $w_0 = 10$ *- Learning rate:* $\eta = 0.001, v = 0.01$ *- Estimated coefficients:* $\lambda_0 = 2$ *and* $\beta_0 = 10$ *- Thresholds:* $\beta_G = 0.35$ *and* $\beta_D = 0.01$
Compensation controller	*- Fuzzy set's parameters:* $a_i = 0.2$ $b_{\text{io}} = [-1 - 0.75 - 0.5\ 0\ 0.5\ 0.75\ 1]$ *- Bound's estimation function:* $c_{\text{io}} = [-3.5 - 2.5 - 1.5\ 0\ 1.5\ 2.5\ 3.5]$ *- Learning rate:* $\beta = 0.01$	*- Bound's estimation function:* $E_l = 0.1$ *- Learning rate:* $\eta_E = 0.001, v_E = 0.01$

Table 3: The comparison between two controllers:WFCMAC and NSOFC

Comparison Efficiency	**Controller WFCMAC**	**Controller NSOFC**
IAE	0.3062	0.2062
ISE	0.0202	0.0220
ITAE	1.4115	0.7175
ITSE	0.0233	0.0078
ISTE	0.1527	0.0412

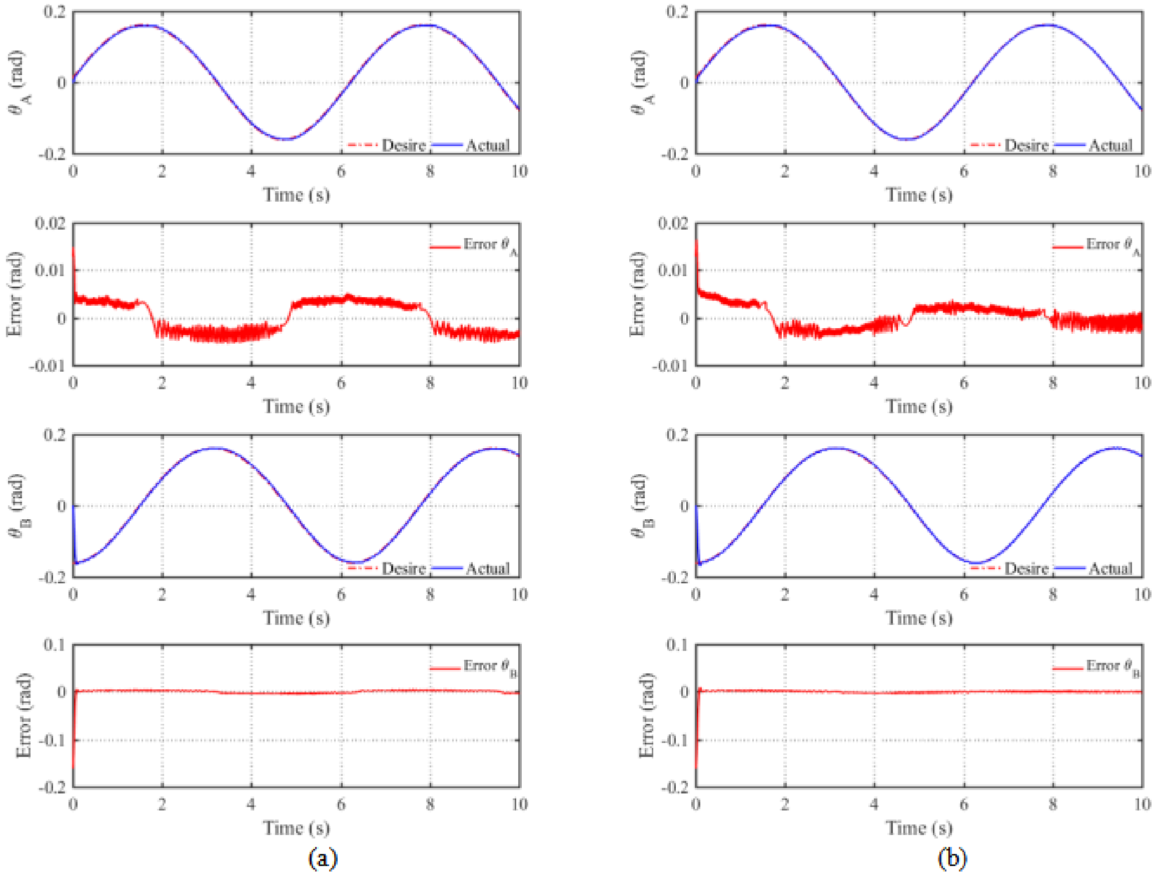

Figure 7: Position responses, tracking errors of arm's angles of two controllers (a) WFCMAC and (b) NSOFC.

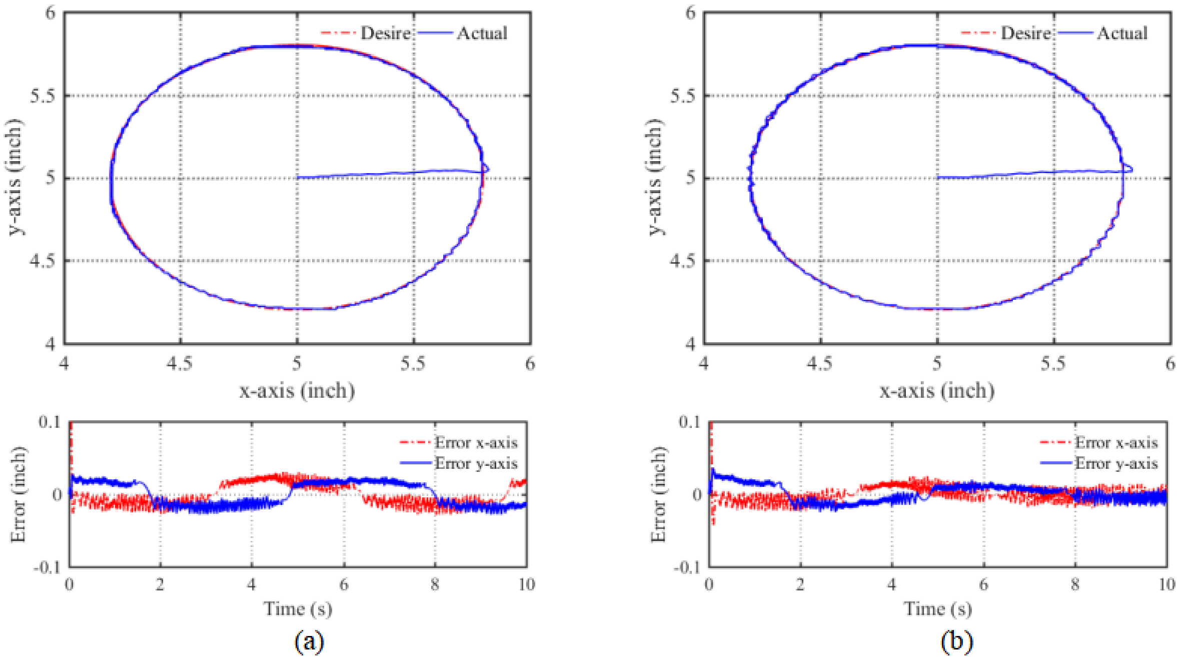

Figure 8: End-effector's position response, tracking errors of the two controllers (a) WFCMAC (b) NSOFC.

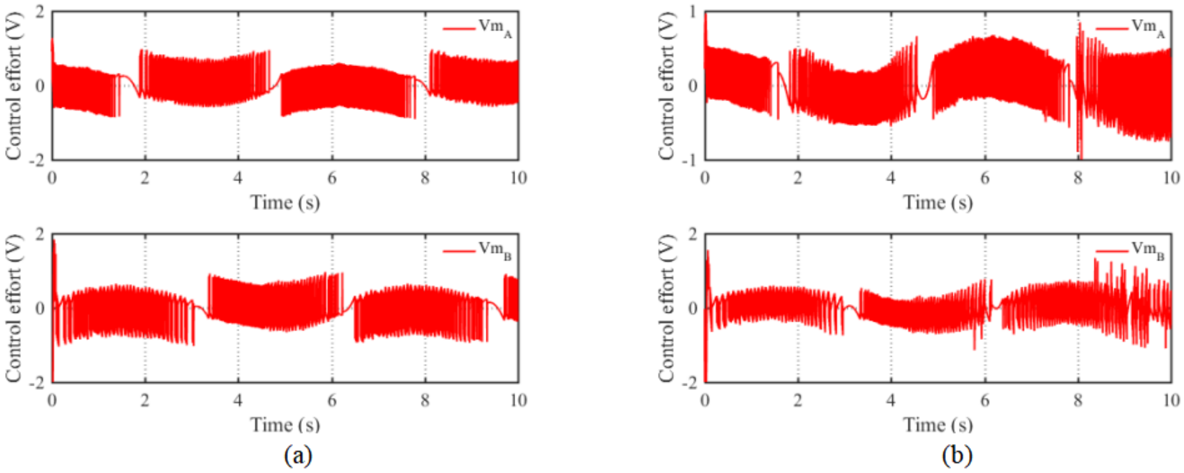

Figure 9: The joint control efforts of the two controllers (a) WFCMAC (b) NSOFC.

in the control signal. The proposed controller can serve as the base for more complicated tasks such as trajectory planning problem for the end-effector of robotic arm using image processing.

References

[1] A. Y. R. Ruiz and B. Chandrasekaran (2020). A Robotic Control System Using Robot Operating System and MATLAB for Sensor Fusion and Human-Robot Interaction, *10th Annual Computing and Communication Workshop and Conference (CCWC)*, 0550–0555, 2020.

[2] M. Stănese, M. Şuşcă, V. Mihaly and I. Naşcu (2020). Design and Control of a Self-Balancing Robot, *IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*, 1–6, 2020.

[3] J. Jiang, A. McCoy, E. Lee and L. Tan (2017). Development of a motion controlled robotic arm, *IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*, 101–105, 2017.

[4] N. Kandalaft, P. S. Kalidindi, S. Narra and H. N. Saha (2018). Robotic arm using voice and Gesture recognition, *IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, 1060–1064, 2018.

[5] J.-H. Park, P. Stegall, and S. K. Agrawal (2015). Dynamic brace for correction of abnormal postures of the human spine, *in Proceedings 2015 IEEE International Conference on Robotics and Automation (ICRA)*, 5922–5927, 2015.

[6] G. Peng, C. Yang, W. He et al (2019). Force sensorless admittance control with neural learning for robots with actuator saturation, *IEEE Transactions on Industrial Electronics*, vol. 67, no. 4, 3138–3148, 2019.

[7] L. Hao; R. Pagani; M. Beschi; G. Legnani (2021). Dynamic and Friction Parameters of an Industrial Robot: Identification, Comparison and Repetitiveness Analysis, *Robotics*, 10–49, 2021.

[8] K. Harada et al. (2005). A Humanoid Robot Carrying a Heavy Object, *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 1712–1717, 2005.

[9] R. Kumar, S. Lal, S. Kumar and P. Chand (2014). Object detection and recognition for a pick and place Robot, *Asia-Pacific World Congress on Computer Science and Engineering*, 1–7, 2014.

[10] N. Neumann, S. Meyer zu Borgsen, P. Lücking and S. Wachsmuth (2018). An automated pick-and-place benchmarking system in robotics, *IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, 243–249, 2018.

[11] M. N. Duc and T. N. Trong (2014). Neural network structures for identification of nonlinear dynamic robotic manipulator, *IEEE International Conference on Mechatronics and Automation*, 1575–1580, 2014.

[12] N. Golea, A. Golea, and K. Benmahammed (2002). Fuzzy model reference adaptive control, *IEEE Transactions on Fuzzy Systems*, vol. 10, 436–444, 2002.

[13] J. H. Park, S. J. Seo, and G. T. Park (2003). Robust adaptive fuzzy controller for nonlinear system using estimation of bounds for approximation errors, *Fuzzy Sets and Systems*, vol. 133, 19–36, 2003.

[14] C.J.B.Macnab (2016). Using RBFs in a CMAC to prevent parameter drift in adaptive control, *Neurocomputing*, vol. 205, 45–52, 2016.

[15] NGO, Thanh Quyen et al. (2014). Robust Adaptive Neural-Fuzzy Network Tracking Control for Robot Manipulator, *International Journal of Computers Communications & Control*, vol. 7, no. 2, 341–352, 2014.

[16] J. S. Albus (1975). A new approach to manipulator control: The cerebellar model articulation controller (CMAC), *Journal of Dynamic Systems, Measurement, and Control*, vol. 97, no. 3, 220–227, 1975.

[17] H. Shiraishi, S. L. Ipri, and D. D. Cho (1995). CMAC neural network controller for fuel-injection systems, *IEEE Trans. Control Syst. Technol.*, vol. 3, no. 1, 32–38, 1995.

[18] S. Jagannathan, S. Commuri, and F. L. Lewis (1998). Feedback linearization using CMAC neural networks, *Automatica*, vol. 34, no. 3, 547–557, 1998.

[19] Y. H. Kim and F. L. Lewis (2000). Optimal design of CMAC neural-network controller for robot manipulators, *IEEE Trans. Syst. Man Cybern. C, Appl. Rev.*, vol. 30, no. 1, 22–31, 2000.

[20] C. Lin and T. Chen (2009). Self-Organizing CMAC Control for a Class of MIMO Uncertain Nonlinear Systems, *IEEE Transactions on Neural Networks*, vol. 20, no. 9, 1377–1384, 2009.

[21] Chih-Min Lin, and Hsin-Yi Li (2012). A Novel Adaptive Wavelet Fuzzy Cerebellar Model Articulation Control System Design for Voice Coil Motors, *IEEE Trans. Ind. Electron.*, vol. 59, no. 4, 2024–2033, 2012.

[22] H. C. Lu, C. Y. Chuang (2011). Robust parametric CMAC with self-generating design for uncertain nonlinear systems, *Neurocomputing*, vol. 74, no. 4, 549–562, 2011.

[23] J. Zhang et al. (2020). A Novel Self-Organizing Emotional CMAC Network for Robotic Control, *International Joint Conference on Neural Networks (IJCNN)*, 1–6, 2020.

[24] Lee HM, Chen CM, Lu YF (2003). A self-organizing HCMAC neural-network classifier, *IEEE Trans Neural Netw.*, 14(1):15-27. PMID: 18237986, 2003.

[25] C. Min, L. Y. Chen, C. H. Chen (2007). RCMAC Hybrid Control for MIMO Uncertain Nonlinear Systems Using Sliding-Mode Technology, *IEEE Trans. Neural Netw.*, vol. 18, no. 3, 708–720, 2007.

[26] T. T. Huynh, T. L. Le, and C. M. Lin (2018). A TOPSIS multi-criteria decision method-based intelligent recurrent wavelet CMAC control system design for MIMO uncertain nonlinear systems, *Neural Computing and Applications*, 2018.

[27] F. O. Rodriguez, W. Yu, and M. A. Moreno-Armendariz (2008). Nonlinear systems identification via two types of recurrent fuzzy CMAC, *Neural Processing Letters*, vol. 28, 49–62, 2008.

[28] T. Bäck and H. Schwefel (1993). An Overview of Evolutionary Algorithms for Parameter Optimization, *Evolutionary Computation*, vol. 1, no. 1, 1–23, 1993, doi: 10.1162/evco.1993.1.1.1.

[29] S. Park and J. Lee (2016). Stochastic Opposition-Based Learning Using a Beta Distribution in Differential Evolution, *IEEE Transactions on Cybernetics*, vol. 46, no. 10, 2184–2194, 2016, doi: 10.1109/TCYB.2015.2469722.

[30] Elsisi, Mahmoud. (2019). Future search algorithm for optimization, *Evolutionary Intelligence*, 12. 10.1007/s12065-018-0172-2.

[31] Hussain Shareef, Ahmad Asrul Ibrahim, Ammar Hussein Mutlag (2015). Lightning search algorithm, *Applied Soft Computing*, vol. 36, 315–333, 2015, ISSN 1568-4946, https://doi.org/10.1016/j.asoc.2015.07.028.

[32] Elsisi, Mahmoud & Abdelfattah, Hany. (2019). New design of variable structure control based on lightning search algorithm for nuclear reactor power system considering load-following operation, *Nuclear Engineering and Technology*, vol. 52, 2019.

[33] Elsisi, Mahmoud (2020). Optimal design of nonlinear model predictive controller based on new modified multitracker optimization algorithm, *International Journal of Intelligent Systems*, 2020, 35. 10.1002/int.22275.

[34] Elsisi, M., & Ebrahim, M. A. (2021). Optimal design of low computational burden model predictive control based on SSDA towards autonomous vehicle under vision dynamics, *International Journal of Intelligent Systems*, 2021, 36(11), 6968–6987. Portico. https://doi.org/10.1002/int.22576.

Cite this paper as:

Ngo T-Q.; Hoang D.-K.; Tran T.-T.; Nguyen T.-T.; Nguyen V.-T.; Le L.-H.(2022). A Novel Self-organizing Fuzzy Cerebellar Model Articulation Controller Based Overlapping Gaussian Membership Function for Controlling Robotic System, *International Journal of Computers Communications & Control*, 17(4), 4606, 2022.

https://doi.org/10.15837/ijccc.2022.4.4606