Journal of Robotics and Control (JRC)
Volume 4, Issue 4, 2023
ISSN: 2715-5072, DOI: 10.18196/jrc.v4i4.18943

An Application of Modified T2FHC Algorithm in Two-Link Robot Controller

Thanh Quyen Ngo [1*], Thanh Hai Tran [2], Tong Tan Hoa Le [3], Binh Minh Lam [4]
[1, 3, 4] Faculty of Electrical Engineering Technology, Industrial University of Ho Chi Minh, Vietnam
[2] Office of Planning and Investment, Industrial University of Ho Chi Minh, Vietnam
Email: [1] ngothanhquyen@iuh.edu.vn, [2] tranthanhhai@iuh.edu.vn, [3] letongtanhoa@iuh.edu.vn, [4] lambinhminh@iuh.edu.vn
*Corresponding Author

***Abstract*—Parallel robotic systems have shown their advantages over the traditional serial robots such as high payload capacity, high speed, and high precision. Their applications are widespread from transportation to manufacturing fields. Therefore, most of the recent studies in parallel robots focus on finding the best method to improve the system accuracy. Enhancing this metric, however, is still the biggest challenge in controlling a parallel robot owing to the complex mathematical model of the system. In this paper, we present a novel solution to this problem with a Type 2 Fuzzy Coherent Controller Network (T2FHC), which is composed of a Type 2 Cerebellar Model Coupling Controller (CMAC) with its fast convergence ability and a Brain Emotional Learning Controller (BELC) using the Lyaponov-based weight updating rule. In addition, the T2FHC is combined with a surface generator to increase the system flexibility. To evaluate its applicability in real life, the proposed controller was tested on a Quanser 2-DOF robot system in three case studies: no load, 180 g load and 360 g load, respectively. The results showed that the proposed structure achieved superior performance compared to those of available algorithms such as CMAC and Novel Self-Organizing Fuzzy CMAC (NSOF CMAC). The Root Mean Square Error (RMSE) index of the system that was 2.20E-06 for angle A and 2.26E-06 for angle B and the tracking error that was -6.42E-04 for angle A and 2.27E-04 for angle B demonstrate the good stability and high accuracy of the proposed T2FHC. With this outstanding achievement, the proposed method is promising to be applied to many applications using nonlinear systems.**

Keywords*—Self-Organizing Technique; Cerebellar Model Articulation Controller; Adaptive Control; Brain Emotional Learning Network.

I. INTRODUCTION

The parallel robot systems are widely used in daily life due to the brevity and simplicity of each kinematic chain. This gives the systems the ability to eliminate the influences from the external or internal environment. The ruggedness, durability, and small size of parallel robots have led to many applications in medical fields, manufacturing, transportation... [1–3]. However, understanding the mathematical model of a parallel robot is a complex challenge because the issues related to the weight and the moment of inertia at each joint are difficult to determine [4–7].

Parallel robotic systems often experience non-linearity and require complex control algorithms to ensure accuracy and quality. Nonlinear factors include friction, limited working space and the effects of external forces. In the control of nonlinear systems, many methods have been developed and applied. These methods can be divided into two main groups as model-based and non-model-based controls. In the non-model control group, there are methods such as PID [8] and fuzzy learning [9-11] are typical methods. The common feature of these methods is that no detailed mathematical model of the system is required. Instead, they focus on adjusting control parameters based on feedback from the system to achieve the best performance. However, the limitation of this method is that it is difficult to optimize the control performance.

On the other hand, in the model-based control group, it is necessary to ensure exactly the mathematical equations describing the system to design the control algorithm. Methods such as PD [12], Computational Torque Control (CTC) [13], nonlinear CTC [14], and adaptive control [15–17]. Although these algorithms show high accuracy in theory, they often suffer from problems of instability and low performance when implemented in practice.

An improved solution, the approximation method, combines the advantages of both previous methods. It uses an approximation model to generate an approximation function to control the system. Some traditional approximation control models are adaptive neural network control [18], adaptive control with mobile robot [19–20]. This approach provides a flexible solution for complex systems that does not require fully precise modeling as the model-driven control or relies on complex optimization processes. However, this method may not achieve as high accurate as the model-based control due to the use of the approximate model. At the same time, the approximation method can be sensitive to errors and fluctuations of the system. In addition, the method requires specialized knowledge and much experience to maintain the efficiency and stability of the system.

Researchers [21-22] proposed a controller using a Radial Baseline Functional Neural Network (RBFNN). The main purpose of RBFNN is to approximate a non-linear function from the input to the corresponding output. RBFNN was chosen for its robust and flexible approximation for non-linear systems. With the RBFNN structure, it is possible to use a non-linear basis function such as a Gaussian function to represent the non-linear features of the system. By adjusting the corresponding weights of the Gaussian function, RBFNN can approximate non-linear functions accurately and efficiently [23-24]. However, RBFNN also has limitations regarding its responsiveness to orbitals. Due to the non-linear

nature of RBFNN, approximating trajectories with high complexity or high accuracy requirements can be difficult. In addition, adjusting the weights of the RBFNN to meet the specific requirements of the system can be a big challenge, especially for a very complex model.

The use of neural networks in the dynamic control of robots poses two major challenges. First, the neural network in the robot controller must ensure sufficient nonlinear learning capability to effectively approximate the ideal controllers using online learning. The Cerebellar Model Coupling Controller (CMAC) is one effective model using neural networks. This method has been applied in many applications thanks to its fast-learning convergence and simple structure [25–27]. However, the model is limited in data processing and requires human intervention in determining the structure and parameters of the system.

Second, neural network controllers must contain enough tunable parameters to eliminate the uncertainty components of nonlinear systems. To deal with the uncertain components, recent studies on intelligent control have proposed directly incorporating human expertise into neural networks [28–31]. Fuzzy inference systems have been used as adaptive controllers for robots [32–36], showing one of the most successful applications of fuzzy logic systems [37–40]. Naturally, neural networks have been developed in various ways to deal with the uncertainty components [41–46]. However, the limitation of the tuning parameters leads to a degradation of the controller performance [46]. Hence, it is necessary for a better solution to handle complex control tasks.

In literature [47–50], the authors proposed a new type of neural network inheriting from CMAC and Brain Emotional Learning Controller (BELC) to solve the first problem. A typical BELC network consists of a sensory subsystem and a neural network judgment subsystem [47-48]. The network judgment subsystem indirectly affects the sensory subsystem's output based on input values [49-50]. The input of the BELC is the output of the CMAC network. The weights in the two subsystems are adjusted based on the errors of the input and output respectively. After calculating the system, BELC generates the final output. Based on the interaction of the two subsystems, the overall quality of the system is improved.

For the second problem, a Type 2 Fuzzy Inference System (T2FIS) was introduced into the neural network. The Type 2 fuzzy structure provides greater flexibility because it contains more tunable parameters to handle uncertain components [51–52]. Notably, the Type 2 fuzzy sets can reduce the number of samples to be computed in T2FIS [53–55]. Therefore, T2FIS has been adopted in many applications in robot control [56–59].

A new neural network that integrates T2FIS and components from CMAC and BELC was presented in [60] to solve both the two above problems. This network was named Type 2 Fuzzy Hybrid Control (T2FHC). However, the given trajectory of the system was quite simple. The parameters of the Gauss function were fixed and the uncertainty components were given. Meanwhile, the theory of the T2FHC was tested using simulation only, which limited the verification of the theory.

In this paper, a modified T2FHC neural network is proposed for solving nonlinear control systems. Different from the original T2FHC, the CMAC employs a self-Organizing structure to automatically change the parameters without human intervention, which solves the limitations of CMAC. In addition, the system integrates the BELC network into the network topology of the CMAC to improve the system predictability. To verify its feasibility, the modified T2FHC algorithm was applied to control a Quanser 2-DOF robot with a complex trajectory. Three case studies of the experiments were conducted, including no load, 180 g load, and 360 g load.

The research contributions of this study include:

- An improved T2FHC model by adding a self-organizing structure so that the Gaussian function parameters can update themselves.
- Practical experiments to confirm the theory of T2FHC with three case studies.
- The comparison results of the proposed model, the CMAC, and the NSOF CMAC to show the superior performance of the modified T2FHC neural network.

The paper is organized as follows. Section 2 describes the kinematic equation of a two-order parallel robot. Section 3 introduces the theory of the T2FHC model and the improvements in this study. Section 4 presents the experimental set up. Section 5 shows the experimental results of the improved T2FHC algorithm, the CMAC, and the NSOF CMAC. Finally, conclusions and discussions are given in Section 6.

II. DYNAMIC EQUATION DESCRIPTION

In this paper, a robotic Quanser 2-DOF controller would be used to test the proposed algorithm. Fig. 1 shows the diagram of a 2-DOF robot. The equation of motion of the robot is expressed in Lagrange in the equation (1).

$$M(q')\ddot{q}' + C(q', \dot{q}')\dot{q}' + g(q') = K_\tau I_m \quad (1)$$

Where $q', \dot{q}', \ddot{q}' \in R^n$ are the vectors of position, speed, and acceleration of the joint, respectively; $M(q') \in R^{nxn}$ is matrix of inertia moment; $C(q', \dot{q}') \in R^{nxn}$ expresses the matrix of centripetal and Coriolis forces; $g(q') \in R^{nx1}$ is the gravitation vector; $I_{mi}(i = 1, 2)$ is the ith amarture current of the two servos; $K_\tau = diag(K_{\tau 1}, K_{\tau 2})$ is the constant representing the mechanical and electrical transient between the current and the torque.

In this paper, a robotic Quanser 2 DOF controller as shown in Fig. 1 is used to verify the dynamic properties given in section IV. Rewrite (1) the kinematics equation of the robot manipulator obtained as equation (2).

$$\ddot{q}' = -M^{-1}(q')[C(q', \dot{q}')\dot{q}' + g(q')] + M^{-1}(q')K_\tau I_m$$
$$= F\left(\underline{x}(t)\right) + G\left(\underline{x}(t)\right)K_\tau I_m$$

$$F\left(\underline{x}(t)\right) = \begin{bmatrix} F_1\left(\underline{x}(t)\right) \\ F_2\left(\underline{x}(t)\right) \\ \vdots \\ F_N\left(\underline{x}(t)\right) \end{bmatrix} = -M^{-1}(q')[C(q',\dot{q}')\dot{q}' + g(q')]$$

$$G\left(\underline{x}(t)\right) = \begin{bmatrix} G_{11}\left(\underline{x}(t)\right) & \cdots & G_{1n}\left(\underline{x}(t)\right) \\ \vdots & \ddots & \vdots \\ G_{n1}\left(\underline{x}(t)\right) & \cdots & G_{nn}\left(\underline{x}(t)\right) \end{bmatrix} = M^{-1}(q') \quad (2)$$

Where, $F\left(\underline{x}(t)\right)$, $G\left(\underline{x}(t)\right)$ are nonlinear dynamic functions which are difficult to determine exactly or can not even obtain. So, we can not establish a model-based control system. In order to with this problem, here we assume that actual value $f(x)$ and $g(x)$ can be separated as nominal part denoted by $F\left(\underline{x}(t)\right)$, $G\left(\underline{x}(t)\right)$. $l(\underline{x}(t))$is represented as the unknown lumped uncertainly and $\underline{x}(t) = \left[x^T(t)\ \dot{x}^T(t)\ \ldots\ x^{(n-1)T}(t)\right]$ is a vector that represents the joint position and velocity. Finally, the system (2) can be rewritten as equation (3).

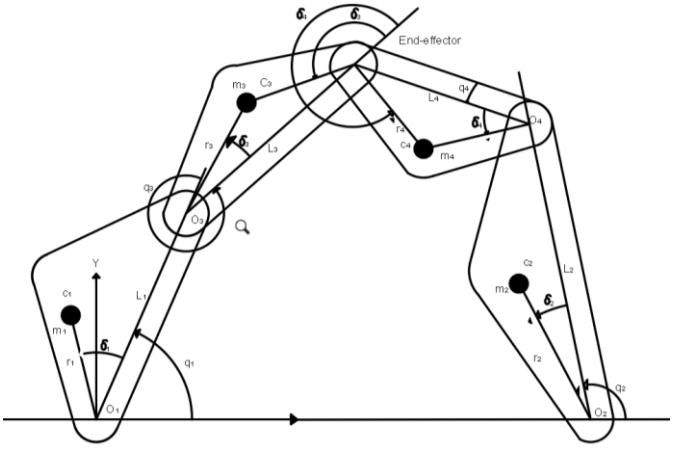

Fig. 1. A 2-DOF robot diagram

$$\ddot{q}'(t) = f\left(\underline{x}(t)\right) + G\left(\underline{x}(t)\right)K_\tau I_m + l(\underline{x}(t)) \quad (3)$$

Where $\ddot{q}'(t) \in R^k$ is the output of the system.

The control problem is to force $q'(t)$ of the system to track the derised reference trajectory $q_d'(t) \in R^k$. The tracking error is identified as $e(t) = q_d'(t) - q'(t)$, $e(t) \in R^k$. The system error is defined as equation (4).

$$\underline{e}(t) = \left[e^T\ \dot{e}^T, \ldots, e^{(n-1)T}\right]^T \quad (4)$$

The sliding surface is defined as equation (5).

$$s\left(\underline{e}(t)\right) = e^{(n-1)}(t) + \zeta_1 e^{(n-2)}(t) + \cdots + \zeta_{n-1}e(t) + \zeta_n \int_0^t e(t)dt \quad (5)$$

Where $s = [s_1\ s_2\ \ldots\ s_k]^T$, $\zeta_i = diag\left(\zeta_{i_1}, \zeta_{i_2}, \ldots, \zeta_{i_k}\right)$ with $i = 1, 2, 3, \ldots, n$. ζ_i is found to satistfy the Hurwitz polynomial. Differentiating $s\left(\underline{e}(t)\right)$ with respect to time leads to equation (6).

$$\dot{s}\left(\underline{e}(t)\right) = e^{(n)}(t) + \zeta_1 e^{(n-1)}(t) + \cdots + \zeta_n e(t) = C^T\dot{\underline{e}}(t) + K^T\underline{e}(t) \quad (6)$$

Where $C = [0\ 0\ \ldots\ I]^T$ and $K = [\zeta_n\ \zeta_{n-1}\ \ldots\ \zeta_1]^T$ denotes the feedback gain matrix.

If the nominal parts $F\left(\underline{x}(t)\right)$, $G\left(\underline{x}(t)\right)$and the uncertainly $l(\underline{x}(t))$ are exactly known, then an ideal controller can be designed as equation (7).

$$u_{ISM} = G_n^{-1}\left[x_d^{(n)} - f_n(\underline{x}) - l(\underline{x}, t) + K^T\underline{e} + \varrho sgn\left[s\left(\underline{e}(t)\right)\right]\right] \quad (7)$$

where $\varrho sgn\left[s\left(\underline{e}(t)\right)\right]$ is the learning law of the sliding surface generator and $\varrho > 0$. The error e is handled by the sliding surface s as equation (8).

$$\dot{s}\left(\underline{e}(t)\right) = G_n[u_{ISM} - u] - \varrho sgn\left[s\left(\underline{e}(t)\right)\right] \quad (8)$$

III. MODIFIED T2FHC CONTROLLERS

Fig. 2 illustrates the structure of the proposed controller. The controller consists of three components: 1) a sliding surface generator, 2) a modified Type-2 Fuzzy Hybrid Controller (m-T2FHC), and 3) a robust controller (RC). The input is pre-processed by the sliding surface generator before being fed to both the m-T2FHC and the RC. The outputs of these controllers are then combined to create the final actuating signal to control the system.

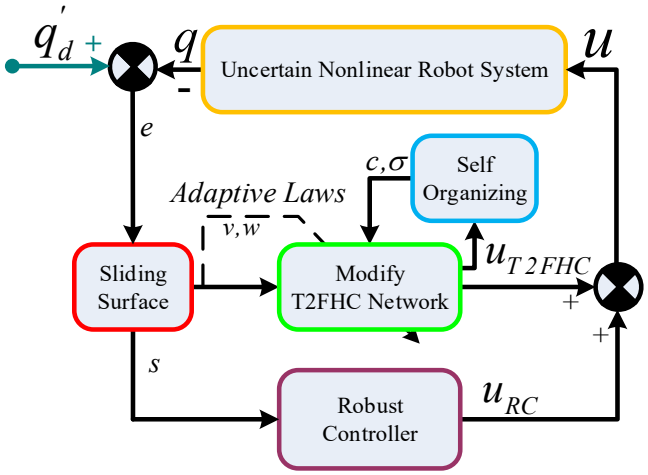

Fig. 2. Modified T2FHC control for uncertain nonlinear robotic systems

A. The T2FHC Structure:

Fig. 3 presents the structure of the T2FHC. It is composed of the main components of the Type-2 CMAC network and the BELC network. In particular, the input layer X, the fuzzification layer F, and the receptive layer T are those from the Type-2 CMAC neural network [46], while the remaining layers follows the BELC. The two weight vectors in the weight memory T are summed in the summarization layer S before being fed into the output layer Y for the final outputs.

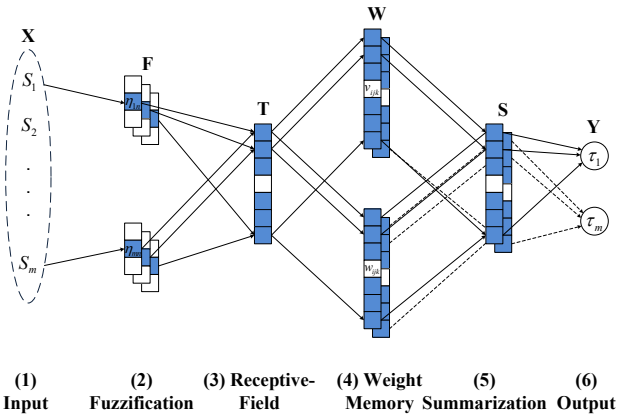

Fig. 3. The structure of the T2FHC

1) Input layer X:

The input of the network is the continuous signals from the sliding surface generator, that is $X = [s_1, s_2]^T$.

2) Fuzzification Layer F:

The underlying Type-1 Gaussian membership function within each block can be represented as equation (9).

$$\mu_{\tilde{F}_{ijk}}(x_i) = T(x_i, c_{ijk}, \sigma_{ijk}) = \exp\left(-\frac{(x_i - c_{ijk})^2}{2 \cdot \sigma_{ijk}^2}\right) \tag{9}$$

Where x_i represents the ith input of the network; c_{ijk} is the mean and σ_{ijk} is the variance for the kth layer corressonding to the ith input at the current state; $\overline{c}_{ijk}$ is the upper bound and $\underline{c}_{ijk}$ is the lower bound of $\mu_{\tilde{F}_{ijk}}(x_i)$ ($c_{ijk} \in [\underline{c}_{ijk}, \overline{c}_{ijk}]$).

The upper and lower bounds $\overline{\mu}_{\tilde{F}_{ijk}}$ and $\underline{\mu}_{\tilde{F}_{ijk}}$ of $\mu_{\tilde{F}_{ijk}}$ are defined as in equation (10).

$$\underline{\mu}_{\tilde{F}_{ijk}}(x_i) = \begin{cases} T\left(x_i, \overline{c}_{ijk}, \sigma_{ijk}\right), & x_i < \dfrac{\underline{c}_{ijk} + \overline{c}_{ijk}}{2} \\ T\left(x_i, \underline{c}_{ijk}, \sigma_{ijk}\right), & x_i > \dfrac{\underline{c}_{ijk} + \overline{c}_{ijk}}{2} \end{cases} \tag{10}$$

$$\overline{\mu}_{\tilde{F}_{ijk}}(x_i) = \begin{cases} T\left(x_i, \overline{c}_{ijk}, \sigma_{ijk}\right), & x_i < \underline{c}_{ijk} \\ 1, & \underline{c}_{ijk} < x_i < \overline{c}_{ijk} \\ T\left(x_i, \underline{c}_{ijk}, \sigma_{ijk}\right), & x_i > \overline{c}_{ijk} \end{cases} \tag{11}$$

Each block has three adjustable parameters: 1) the upper bound of the mean value; 2) the lower bound of the mean; and 3) the variance σ of the Type-1 Gaussion function.

3) Receptive layer T:

The receptive layer is defined as equation (12).

$$\mathbb{F}_\lambda = \left[\underline{\mathbb{F}}_\lambda \ \ \overline{\mathbb{F}}_\lambda\right] = \left[\prod_{i=1}^{n_i} \underline{\mu}_{\tilde{F}_{ijk}} \ \ \prod_{i=1}^{n_i} \overline{\mu}_{\tilde{F}_{ijk}}\right]^T \tag{12}$$

Where F_λ representing the λ th receptive field, $\lambda \in \{1, 2, \ldots, n\_T\}$. This layer means that the output space is limited by the upper bound $(\underline{F}_\lambda)$ and the lower bound $(\overline{F}_\lambda)$.

4) Weight layer W:

In this layer, each receptive field of the previous layer is calculated by the corresponding weights v_{λ_q} and w_{λ_q}. The weights all have a left bound and a right bound, which are correlated with v_{λ_q} and w_{λ_q} as equation (13) to (16).

$$v_{\lambda_q} = \left[v_{\lambda_q}^l \ \ v_{\lambda_q}^r\right] \tag{13}$$

$$w_{\lambda_q} = \left[w_{\lambda_q}^l \ \ w_{\lambda_q}^r\right] \tag{14}$$

Where l and r are the left and right bounds of the fields relative to v_{λ_q} or w_{λ_q}.

$$\dot{v}_{\lambda_q} = \alpha\left[\mathbb{F}_\lambda \cdot \left(max\left[0, d_q - a_q\right]\right)\right] \tag{15}$$

$$\dot{w}_{\lambda_q} = \beta\left[\mathbb{F}_\lambda \cdot \left(u_{T2FHC_q} - d_q\right)\right] \tag{16}$$

Where α and β is the learning rate of the update rules, a_q and u_{T2FHC_q} is the output of v_{λ_q} and w_{λ_q}; d_p is a parameter defined by equation (17).

$$d_p = b_i \cdot x_i + c_q \cdot u_{T2FHC_q} \tag{17}$$

Where b_i and c_q are the gain parameters. When v_{λ_q} and w_{λ_q} share the same implementation structure that can be expressed as equation (18).

$$W = \begin{bmatrix} w_{11} & \cdots & w_{1o} & \cdots & w_{1p} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ w_{k1} & \cdots & w_{ko} & \cdots & w_{kp} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ w_{n_T 1} & \cdots & w_{n_T o} & \cdots & w_{n_T p} \end{bmatrix} \tag{18}$$

Where p is the dimensionality of the network's output.

5) Summarization Layer S:

$a_q = \sum_{\lambda=1}^{n_\lambda} f_{\lambda_q} v_{\lambda_q}$ and $o_q = \sum_{\lambda=1}^{n_\lambda} f_{\lambda_q} w_{\lambda_q}$ [61].

The output of the summarization layer therefore can be seen in equation (19).

$$S_{net} = a_q - o_q = \sum_{\lambda=1}^{n_\lambda} \mathbb{F}_{\lambda_q}\left(v_{\lambda_q} - w_{\lambda_q}\right) \tag{19}$$

Where Z_{λ_q} is the summarized weight, which is defined by equation (20).

$$Z_{\lambda_q} = \left[Z^l{}_{\lambda_q} \ Z^r{}_{\lambda_q}\right]^T = \left[\left(v^l{}_{\lambda_q} - w^l{}_{\lambda_q}\right) \ \ \left(v^r{}_{\lambda_q} - w^r{}_{\lambda_q}\right)\right]^T \tag{20}$$

The type-reduction method as reported in and is applied as in equation (21) and (22).

$$y_q^l = \frac{\sum_{\lambda=1}^{\mathcal{L}} \overline{\mathbb{F}}_\lambda Z^l{}_{\lambda_q} + \sum_{\lambda=\mathcal{L}+1}^{n_\lambda} \underline{\mathbb{F}}_\lambda Z^l{}_{\lambda_q}}{\sum_{\lambda=1}^{\mathcal{L}} \overline{\mathbb{F}}_\lambda + \sum_{\lambda=\mathcal{L}+1}^{n_\lambda} \underline{\mathbb{F}}_\lambda} \tag{21}$$

$$y_q^r = \frac{\sum_{\lambda=1}^{\mathcal{R}} \overline{\mathbb{F}}_\lambda Z^l{}_{\lambda_q} + \sum_{\lambda=\mathcal{R}+1}^{n_\lambda} \underline{\mathbb{F}}_\lambda Z^l{}_{\lambda_q}}{\sum_{\lambda=1}^{\mathcal{R}} \overline{\mathbb{F}}_\lambda + \sum_{\lambda=\mathcal{R}+1}^{n_\lambda} \underline{\mathbb{F}}_\lambda} \tag{22}$$

Where,

$$Z_{\lambda_q} = \left[Z^l{}_{\lambda_q}\ Z^r{}_{\lambda_q}\right]^T = \left[Z^l{}_{1_q}, Z^l{}_{2_q}, \dots, Z^l{}_{n_{\lambda_q}}\ \ Z^r{}_{1_q}, Z^r{}_{2_q}\right]^T$$

Output Layer Y: The qth output is simply computed by equation (23).

$$Y_q = \frac{y_q^l + y_q^r}{2} \tag{23}$$

B. The Online Learning Rules

Fig. 4 shows the schematic of a two-dimensional T2FHC. Applying the gradient descent method, the $v_{\lambda_q}^l, v_{\lambda_q}^r, w_{\lambda_q}^l, w_{\lambda_q}^r$ parameter of the updating rules are defined as equations (24) to (27).

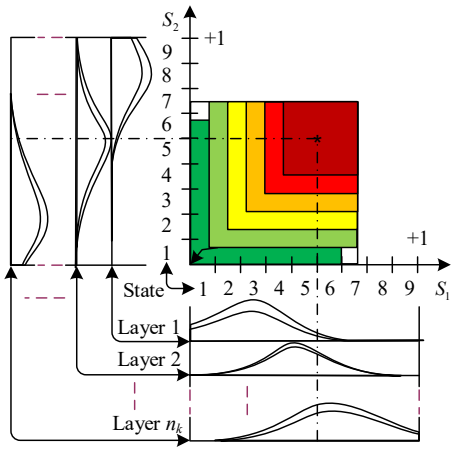

Fig. 4. Schematic of 2-D T2FHC network

$$v_{\lambda_q}^l(k+1) = v_{\lambda_q}^l(k) + \dot{v}_{\lambda_q}^l \tag{24}$$

$$v_{\lambda_q}^r(k+1) = v_{\lambda_q}^r(k) + \dot{v}_{\lambda_q}^r \tag{25}$$

$$w_{\lambda_q}^l(k+1) = w_{\lambda_q}^l(k) + \dot{w}_{\lambda_q}^l \tag{26}$$

$$w_{\lambda_q}^r(k+1) = w_{\lambda_q}^r(k) + \dot{w}_{\lambda_q}^r \tag{27}$$

Where, $\hat{\sigma}_{i_\lambda}^l$ and $\hat{\sigma}_{i_\lambda}^r$ denote the adjustments of σ_{i_λ}; $(\dot{v}_{\lambda_q}^l, \dot{v}_{\lambda_q}^r)$ and $(\dot{w}_{\lambda_q}^l, \dot{w}_{\lambda_q}^r)$ indicate the left and right bound weight of v_{λ_q} and w_{λ_q}.

C. Algorithm of the m-T2FHC Network

a) Normalize each dimension x_i of X form 0 to n_R;
b) Compute F_λ using equation (10) to (12).
c) Calculate Z_{λ_q} in equation (20), and then y_q^l and y_q^r in equation (21) and (22).
d) Derive the output Y_q of the network by equation (23).
e) Expand the Taylor series to calculate $\underline{c}_{i_\lambda}$, $\overline{c}_{i\lambda}$, σ_{i_λ}.
f) Update $v_{\lambda_q}^l, v_{\lambda_q}^r, w_{\lambda_q}^l, w_{\lambda_q}^r$.

The parallel robot is a complex nonlinear system, so multiple layers must be selected. This helps to increase the responsiveness of the network structure for the system. In this study, we chose n_λ = 11 that is the last case in [60]. The remaining parameters are given as in equations (28) to (33).

$$\mathbb{F}_\lambda^l = \frac{\overline{\mathbb{F}}_\lambda}{\sum_{\lambda=1}^{\mathcal{L}} \overline{\mathbb{F}}_\lambda + \sum_{\lambda=\mathcal{L}+1}^{n_\mathcal{L}} \underline{\mathbb{F}}_\lambda} \tag{28}$$

$$\mathbb{F}_\lambda^r = \frac{\underline{\mathbb{F}}_\lambda}{\sum_{\lambda=1}^{\mathcal{R}} \overline{\mathbb{F}}_\lambda + \sum_{\lambda=\mathcal{R}+1}^{n_\mathcal{R}} \underline{\mathbb{F}}_\lambda} \tag{29}$$

$$\dot{v}^l{}_{\lambda_q} = \alpha\left[\mathbb{F}_\lambda^l \cdot \left(max\left[0, d_q - a_q\right]\right)\right] \tag{30}$$

$$\dot{w}^l{}_{\lambda_q} = \beta\left[\mathbb{F}_\lambda^l \cdot \left(u_{T2FHC_q} - d_q\right)\right] \tag{31}$$

$$\dot{v}^r{}_{\lambda_q} = \alpha\left[\mathbb{F}_\lambda^r \cdot \left(max\left[0, d_q - a_q\right]\right)\right] \tag{32}$$

$$\dot{w}^r{}_{\lambda_q} = \beta\left[\mathbb{F}_\lambda^r \cdot \left(u_{T2FHC_q} - d_q\right)\right] \tag{33}$$

The above algorithm has described in detail the working process of the proposed T2FHC network. The complexity of the network depends on the number of inputs (n_i), number of block types (n_T) and the number of outputs (n_o). The values of n_i and n_o are determined once the controlled system is specified.

Denoting u_{T2FHC}^* as the ideal output of the T2FHC network, the output u_{ISM} of the sliding control can be found in equation (34).

$$u_{ISM} = Z^{*T}\mathbb{F}^* + \varepsilon = u_{T2FHC}^*(X, Z^*, c^*, \sigma^*) + \varepsilon \tag{34}$$

with Z^*, c^*, σ^*, $\mathbb{F}^*$the ideal parameters and ε a vector of minimum errors.

These ideal values are impractical, though. Thus, the output of the T2FHC network is approximated with the equation (35).

$$u = \hat{u}_{T2FHC}\left(X, \hat{Z}, \hat{c}, \hat{\sigma}\right) + u_{RC} = \hat{Z}^T\widehat{\mathbb{F}} + u_{RC} \tag{35}$$

Replacing the equations (34), (35) into (8), the output of T2FHC is expressed as equation (36).

$$\dot{s}\left(\underline{e}(t)\right) = G_n\left[\tilde{Z}^T\mathbb{F}^* + \hat{Z}^T\tilde{F} + \varepsilon - u_{RC}\right] - \varrho sgn\left[s\left(\underline{e}(t)\right)\right] \tag{36}$$

Setting $\tilde{Z} = Z^* - \hat{Z}$, the practical output of the network $\tilde{u}_{T2FHC}$ is correlated with $\tilde{Z}$ in the equation (37).

$$\tilde{u}_{T2FHC} = u_{T2FHC}^* - \hat{u}_{T2FHC} = Z^{*T}\mathbb{F}^* - \hat{Z}^T\widehat{\mathbb{F}} \tag{37}$$

Where, $Z^*$and $\hat{Z}$ are the corresponding outputs of the nwtworks. They are calculated to minimize the system error $s\left(\underline{e}(t)\right)$ and then we get the equation (38).

$$lim\left|\tilde{u}_{T2FHC_q}\right| = lim\left|\left(Z_q^{*T} - \hat{Z}_q^T\right)\mathbb{F}^*\right| = \left|\gamma_q\right| \tag{38}$$

The Taylor series is extended to convert a nonlinear function to a linear function in equation (39).

$$\widetilde{\mathbb{F}} = \begin{bmatrix} \widetilde{\mathbb{F}}_1 \\ \vdots \\ \widetilde{\mathbb{F}}_\lambda \\ \vdots \\ \widetilde{\mathbb{F}}_{n_\lambda} \end{bmatrix} = \begin{bmatrix} \left(\frac{\partial \mathbb{F}_1}{\partial c}\right)^T \\ \vdots \\ \left(\frac{\partial \mathbb{F}_\lambda}{\partial c}\right)^T \\ \vdots \\ \left(\frac{\partial \mathbb{F}_{n_\lambda}}{\partial c}\right)^T \end{bmatrix}\Bigg\|_{m=\hat{m}} (c^* - \hat{c}) + \begin{bmatrix} \left(\frac{\partial \mathbb{F}_1}{\partial \sigma}\right) \\ \vdots \\ \left(\frac{\partial \mathbb{F}_\lambda}{\partial \sigma}\right) \\ \vdots \\ \left(\frac{\partial \mathbb{F}_{n_\lambda}}{\partial \sigma}\right) \end{bmatrix}\Bigg\|_{\sigma=\hat{\sigma}}$$

$$(\sigma^* - \hat{\sigma}) + \beta \equiv f_m\tilde{m} + f_\sigma\tilde{\sigma} + \beta \quad (39)$$

With β is a vector of higher-order terms, $\left[\left(\frac{\partial \mathbb{F}_\lambda}{\partial c}\right)\right]$ and $\left[\left(\frac{\partial \mathbb{F}_\lambda}{\partial \sigma}\right)\right]$ are defined as equation (42) and (43).

$$\left[\frac{\partial \mathbb{F}_\lambda}{\partial c}\right] = \left[0, \dots, 0, \frac{\partial \mathbb{F}_\lambda}{\partial c_{1\lambda}}, \dots, \frac{\partial \mathbb{F}_\lambda}{\partial c_{n_i\lambda}}, 0, \dots, 0\right] \quad (42)$$

$$\left[\frac{\partial \mathbb{F}_\lambda}{\partial \sigma}\right] = \left[0, \dots, 0, \frac{\partial \mathbb{F}_\lambda}{\partial \sigma_{1\lambda}}, \dots, \frac{\partial \mathbb{F}_\lambda}{\partial \sigma_{n_i\lambda}}, 0, \dots, 0\right] \quad (43)$$

Putting (38) and (39) into (36), the T2FHC output becomes obtained in equation (44).

$$\dot{s}\left(\underline{e}(t)\right) = G_n\left[\hat{Z}^T(f_m\tilde{m} + f_\sigma\tilde{\sigma}) + \omega - u_{RC}\right] - \varrho sgn\left[s\left(\underline{e}(t)\right)\right] \quad (44)$$

Where, ω is the approximated error that is formatted by $\omega = \hat{Z}^T\beta + \varepsilon + \gamma$. The robust controller is then designed as in equation (45).

$$u_{RC} = (2R^2)^{-1}(R^2 + I)s\left(\underline{e}(t)\right) \quad (45)$$

Where, R is a positive diagonal matrix, $R = 0.75 * diag(\varphi_1, \varphi_2, \dots, \varphi_i)$. Thus, the output of the whole system is the summary of u_{T2FHC} and u_{RC}, becomes obtained in equation (46).

$$u = u_{T2FHC} + u_{RC} \quad (46)$$

D. Self-Organizing for Gauss Functions

Different from [60], this article uses the Self-Organizing structure of [29]. Accordingly, expand the Taylor series to get the convergence error and the best approximation for the value m.

$$DM_k = \left\| \left[\frac{\partial \mathbb{F}_\lambda}{\partial \boldsymbol{c}}\right] - c_k \right\|_2$$

Where $c_k = [\, c_{1k}, \dots, c_{ik,} \dots, c_{n_i k}\,]^T$. Defined method to add a new layer can be calculated by a formula (47).

$$\hat{k} = arg \min_{1 \le k \le n_k} DM_k \quad (47)$$

If $\max_i DM_{\hat{k}} > K_g$ then a new class is created with K_g is a given limit value. If the new input value is greater than the current value, a new layer will be created with the equation (48).

$$n_k(t+1) = n_k(t) + 1 \quad (48)$$

Where $n_k(t)$ is the current number of classes at time t. Random values will be generated in the new weight class including the original mean and variance as shown in the equation (49) and (50).

$$c_{in_k} = I_i \quad (49)$$

$$\sigma_{in_k} = \sigma_{i\hat{k}} \quad (50)$$

Where, I_i is the new input, $z_{i\hat{k}}$ is a predefined constant.

In order to maintain the structure of the network, the self-Organizing structure will have to consider reducing the number of layers. To reduce the number of kth component layers of the jth output is defined as in equation (51).

$$MM_{jk} = \frac{v_{jk}}{o_j} \quad (51)$$

Find the maximum value of the nth output corresponding to the smallest component using the equation (52).

$$\tilde{k} = arg \min_{1 \le k \le n_k} \max_{1 \le j \le m} MM_{jk} \quad (52)$$

If $MM_{jk} \le K_c$, then the *kth* layer must be deleted with K_c being the given limit value.

IV. APPLICATION IN QUANSER 2 DOF ROBOT

The T2FHC network parameters shown in Fig. 5 are listed in Table I. Where n_I, n_R, n_T, n_λ are the inputs of the network, the regions in the input layer, the block number, and the number of reception fields. $m_{i\lambda}$ and $o_{i\lambda}$ denote the Gaussian function parameters, n_m, n_o are the brain emotional learning rates, b and c are the gain parameters of the brain emotional learning. After many tests, the following parameters were selected to optimize the system and improve control quality.

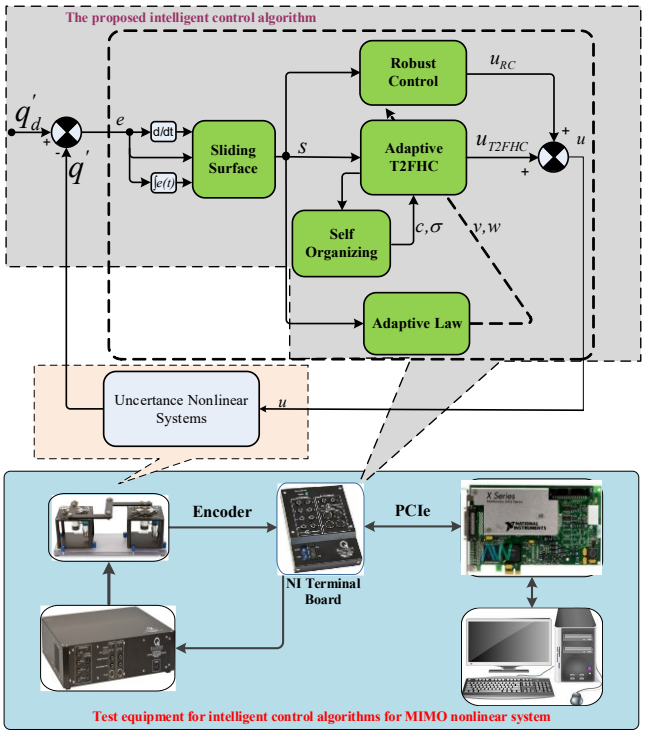

Fig. 5. Overall structure diagram of the system

TABLE I. INITIALIZED PARAMETER VALUES OF THE PROPOSED NEURAL NETWORK

n_I	2
n_R	5
n_T	4
n_B	2
n_λ	11
$m_{i\lambda}$	[-1 -0.8 -0.6 -0.4 -0.2 0 0.2 0.4 0.6 0.8 1]
$o_{i\lambda}$	1.2
n_m, n_o	0.001
α, β	0.5
b, c	8,2

To validate the effectiveness of the proposed method, we apply the T2FHC model on 1 Quanser 2 DOF Robot. Three case studies were experimented including with load, 180g load, 360g load.

Quanser's 2-step parallel robot experimental model is depicted in Fig. 6. The model consists of 4 link rods, two motors with a maximum speed of 6000v/min, two encoders with a resolution of 4096 pulses/revolutions, a power amplifier and a ni_pcie_6351-0 card that collects data connecting the robot to a computer.

Fig. 6. Realistic model of experimental system

V. EXPERIMENTAL RESULTS

To see the superiority of T2FHC, during the system operation, we assume that there is noise due to loading into the system. We set the weight at 15cm×7cm×1.2cm and the weight at 60g.

During the time t=0 to t=3 the robot is idling, from t=3 to t=7 we put a 180g weight into the system. To increase the testability of T2FHC, from t=7 onwards we add, and total weight is 360g.

Looking at Fig. 7 it can be seen that when adding each mass to the system in turn for the first time, the robot works normally. This shows that T2FHC has the ability to handle and adjust automatically to maintain stable operation of the robot. At the second time, when the second mass is added (t=7), a deviation occurs in the trajectory of the robot shown in Fig. 7(a). However, the T2FHC system immediately learns and returns the robot to the given orbit. This demonstrates the T2FHC's ability to learn and respond in real time, helping to maintain the robot's stability and accuracy in noisy environments.

Looking at Fig. 8, we can see that, from the time before t = 6, the systematic error does not change much. At time t=7 onwards, the system starts to oscillate, and the error generated is large (above 0.05) in Fig. 8(a). However, the improved T2FHC system responded and returned the system to its given trajectory.

Fig. 9 is the actual trajectory corresponding to the entire operation of the system. Although during operation, errors (greater than 0.05) are generated as shown in Fig. 8(a). But the system works fine with some trajectory difference when the load is applied.

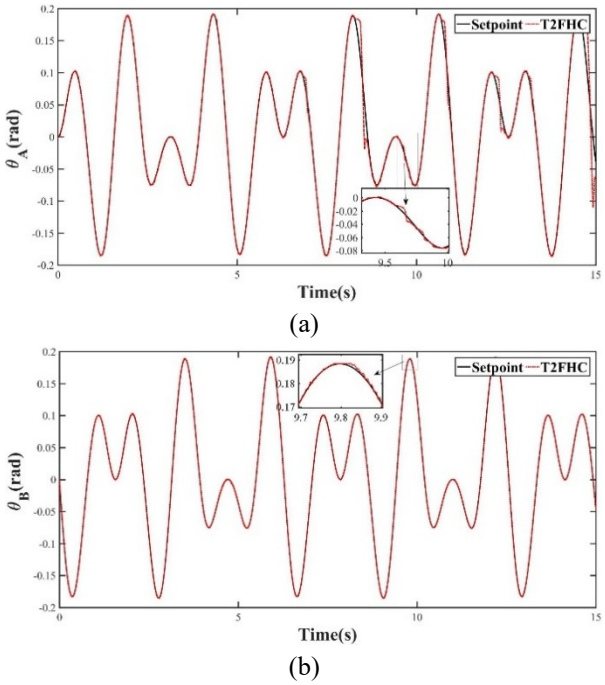

Fig. 7. Positions of: (a) motor A; (b) motor B

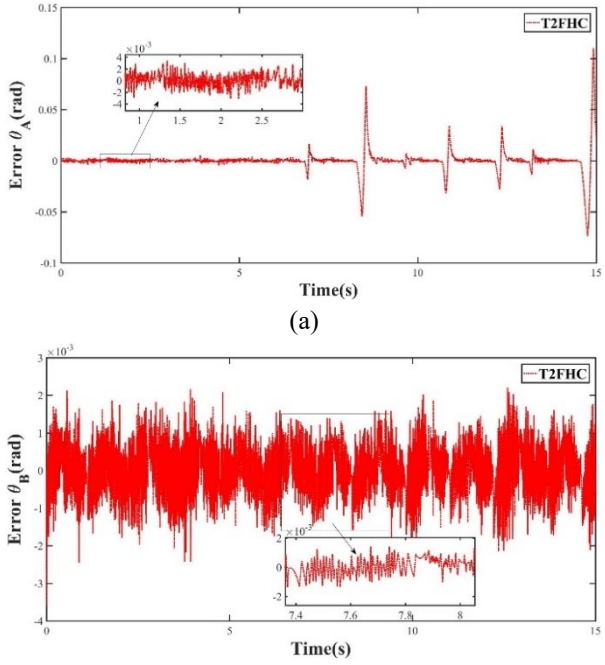

Fig. 8. Tracking errors of the proposed modified T2FHC control system at joints (a) motor A; (b) motor B

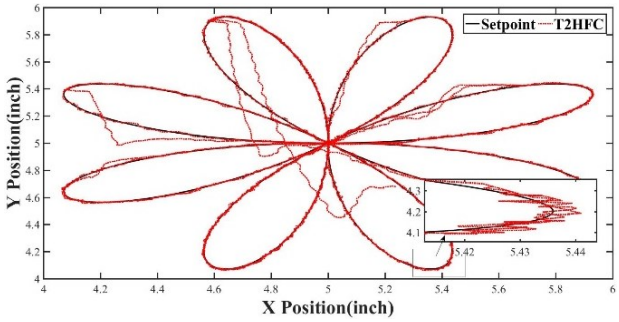

Fig. 9. The actual trajectory of the controller

A. Compare with Other Controllers

The CMAC method [25] and the NSOFC method [29] are said to have the ability and speed to obtain high accuracy for nonlinear systems. We will compare based on two real cases, one is the system idling and the other is the system running at different loads. Then compare the results to see if T2FHC is better than the current 2 methods, CMAC and NSOFC

1. **Case 1**: No load.

As can be seen from Fig. 10, the improved T2FHC shows less oscillation error (-6.42e-04 in Fig. 10(a) motor A and 2.27e-04 in Fig. 10(b) motor B) compared with other controllers. This shows that the proposed method has better stability and ability to pursue generated funds than NSOFC and CMAC. However, it is still not possible to conclude that the current method is more optimal than the remaining methods. Therefore, several more tests should be performed for a more detailed evaluation of T2FHC.

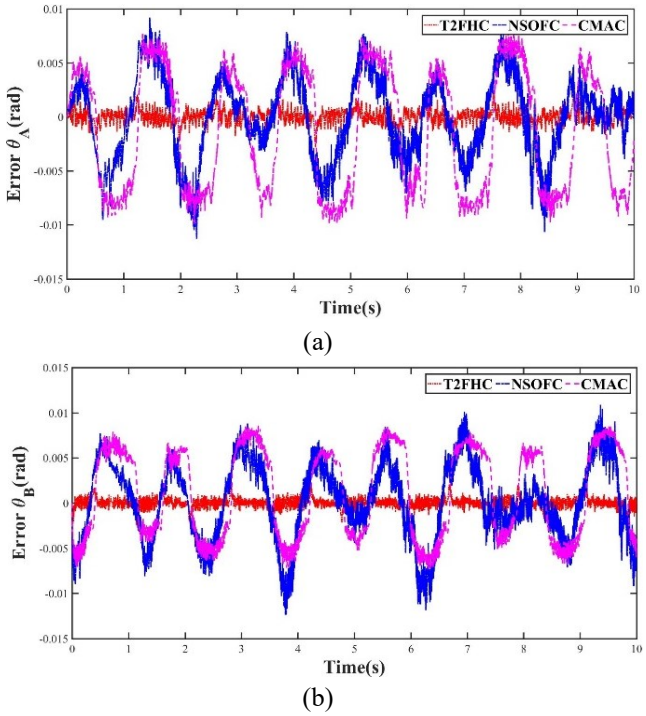

Fig. 10. Tracking errors of the proposed modified T2FHC control system and CMAC, NSOFC at joints (a) motor A; (b) motor B

Fig. 11 shows the actual trajectories of the controllers. The modify T2FHC is the most prominent controller as it adheres to the given trajectory through the magnified image in the lower right corner. In general, the CMAC or NSOFC controllers are still capable of responding to the system but are still not effective at orbital angles.

In conclusion, though modify T2FHC has shown outstanding performance and good stability under the present test conditions. But more tests need to be continued to evaluate in more detail the control quality and its advantages over other methods.

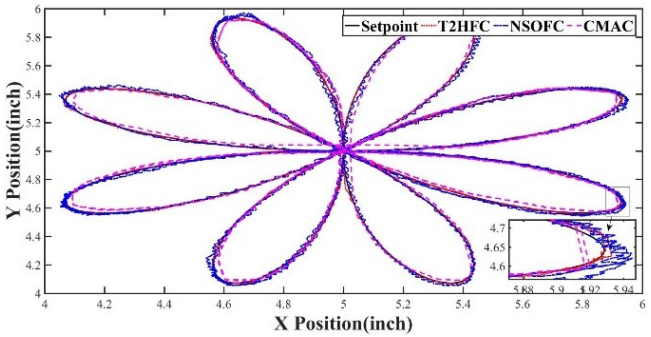

Fig. 11. Realistic trajectories of the controllers

2. **Case 2**: Put a total of 180g upload to the robot.

As can be seen from Fig. 12, the improved T2FHC shows less oscillation error (-1.07e-03 in Fig. 12(a) motor A and 4.24e-04 in Fig. 12(b) motor B) compared with other controllers. However, at time t=2s and t=8s in Fig. 12(a), CMAC and NSOFC have a mutation in error. In Fig. 12(b), the CMAC has a very large error of 5.67e-04 compared to the other two controllers.

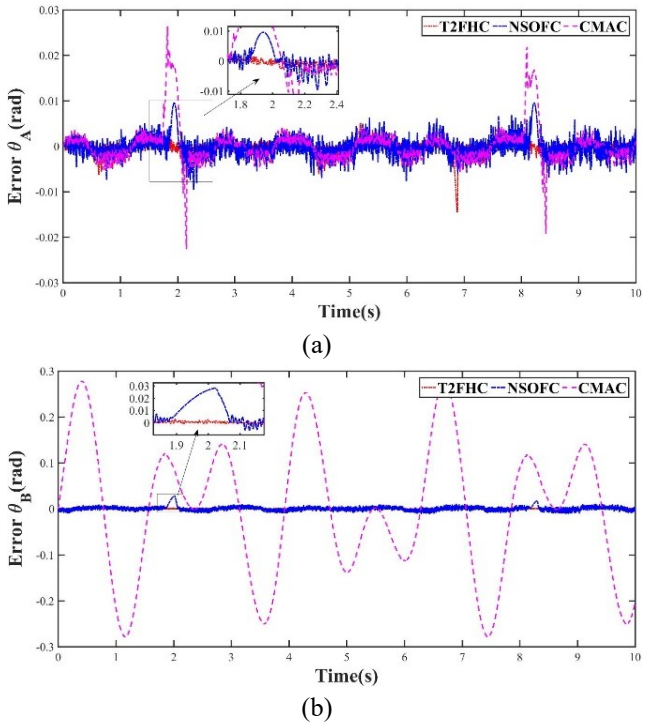

Fig. 12. Tracking errors of the proposed modified T2FHC control system and CMAC, NSOFC at joints (a) motor A; (b) motor B

Fig. 13 shows the actual trajectories of the controllers. The modify T2FHC is the most prominent controller as it adheres to the given trajectory through the magnified image in the lower right corner. CMAC has better response at orbital angles compared to Fig. 11. In contrast, NSOFC has chattering phenomenon, which reduces controller quality.

However, the controllers still adapt to situations where the actual trajectory is not the same as given. By updating the new parameters, the controller will try to bring the system to a stable state and follow the trajectory more accurately. This process may take a short time to achieve and depends on the complexity of the system and controller.

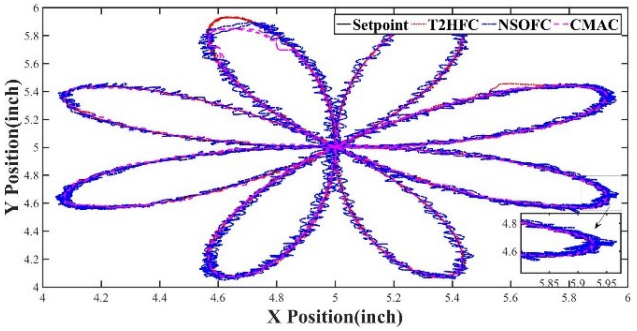

Fig. 13. Realistic trajectories of the controllers

3. **Case 3**: Put a total of 360g upload to the robot.

As can be seen from Fig. 14, the improved T2FHC shows less oscillation error (-1.69e-04 in Fig. 14(a) motor A and 1.22e-04 in Fig. 14(b) motor B) compared with other controllers. Like case 2, at time t=2s and t=8s in Fig. 14(a), CMAC and NSOFC have a mutation in error. In Fig. 14(b), the CMAC has a very large error of 5.67e-04 compared to the other two controllers. However, this time the improved T2FHC had a large error difference at t=4.5s in Fig. 14(a).

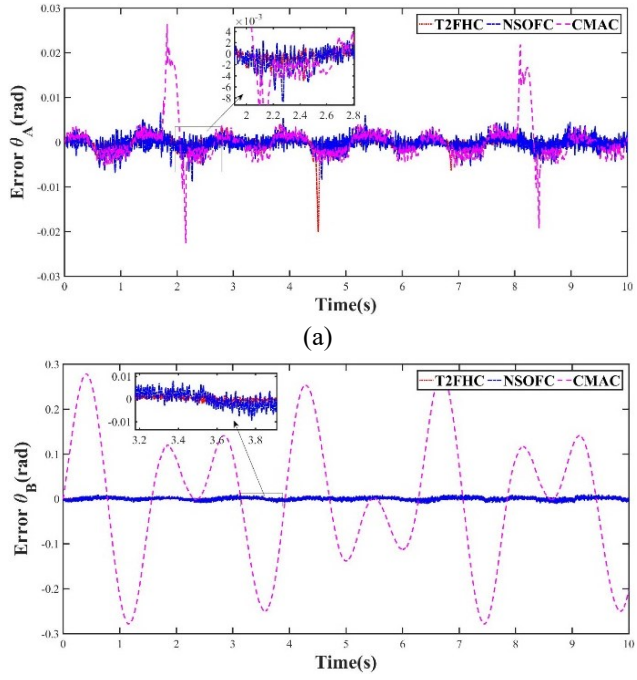

Fig. 14. Tracking errors of the proposed modified T2FHC control system and CMAC, NSOFC at joints (a) motor A; (b) motor B

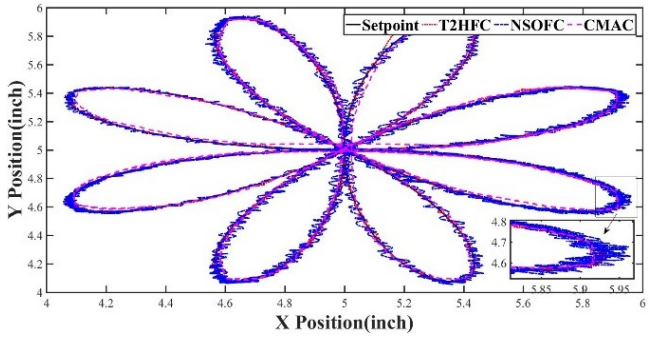

Fig. 15. Realistic trajectories of the controllers

Based on the experiment, it can be seen how the controllers work at different noise levels. All controllers respond to orbital tracking, but both NSOFC and CMAC appear to have performed poorly. As for the T2FHC, it once again confirms its confidence in keeping the error as low as possible in Fig. 14 and maintaining stability in any situation. At this point, it can be tentatively concluded that the improved T2FHC is a remarkable controller as its responsiveness has been demonstrated through the above experiments. The detailed system parameters are shown in Table II.

In summary, T2FHC is a remarkable and responsive controller in controlling non-linear systems. It has shown its reliability in the above experiments by its ability to maintain stability and keep errors as low as possible. This is a remarkable result and shows the potential of modify T2FHC in driving non-linear systems.

TABLE II. PARAMETERS OF CMAC, NSOFC AND MODIFY T2FHC CONTROLLERS IN CASES

m(g)	Joint		NSOFC	CMAC	T2FHC
0	A	RMSE	1.35e-05	1.50e-04	2.20e-06
		Error	-6.45e-04	-2.58e-03	-6.42e-04
	B	RMSE	3.34e-05	1.35e-04	2.26e-06
		Error	4.47e-04	1.56e-03	2.27e-04
180	A	RMSE	1.41e-05	1.57e-04	4.82e-06
		Error	2.27e-04	2.72e-04	-1.07e-03
	B	RMSE	6.55e-05	3.24e-05	5.96e-06
		Error	1.90e-04	5.67e-04	4.24e-04
360	A	RMSE	1.06e-05	4.08e-04	4.66e-06
		Error	-1.04e-04	-1.78e-04	-1.69e-04
	B	RMSE	8.07e-05	1.94e-04	7.38e-06
		Error	9.45e-05	8.07e-04	1.22e-04

Table II shows the results of three different sets (NSOFC, CMAC and T2FHC) when applied to the Quanser 2 DOF robot. This is a regression problem; the goal is to find a function that can represent the relationship between input variables (position of robot arm) and output variable (joint angle of robot arm. Algorithms can learn this function from training data and then use it to predict match angle for new positions.

One way to evaluate the performance of a controller is to use mean squared error (RMSE) and error. Both metrics can be negative or positive, depending on whether the model predicts higher or lower than the actual value. The lower the RMSE and the error, the more accurate the model.

According to the results in the table, it can be concluded:

- The modify T2FHC algorithm is the most efficient in the controllers because of the RMSE and the lowest error for all cases. This shows that modify T2FHC can approximate the function representing the joint angle of the robot arm most accurately. At the same time, it is possible to handle various uncertain components in the system.
- The NSOFC algorithm is the least efficient of the three models. Although it has a lower RMSE and error than

CMAC, the functional approximation of robot arm joints is better than CMAC. But it has chattering phenomenon that causes the quality of the system to go down.

- The CMAC algorithm is the model with the average performance of the three controllers because the RMSE and error are average for the joints and orbital positions. This shows that CMAC can approximate the function representing the joint angle of the robot arm quite well, but not the best. In addition, CMAC can also encounter some problems when predicting the match angle for locations far from the training data. For example, joint A is at 360 degrees, where it has the largest error of the three controllers.

VI. CONCLUSION AND DISCUSSION

This paper proposes an integrated learning network from two main components, Type-2 fuzzy CMAC and BELC. Along with that, the sliding surface generation method is used to control the nonlinear system. One of the remarkable points of the proposed method is the robustness of the control system to uncertain components. This is especially important in practical applications where uncertainty and volatility are frequent. Fuzzy Type 2 CMAC and BELC integrated control systems can respond to variations. Thus, T2FHC achieves stable and accurate performance in monitoring and control.

Comparisons with existing control methods are an important part of the evaluation of the proposed method. Comparative studies focus on CMAC and NSOFC algorithms. By comparing with other methods, the strengths and weaknesses of the proposed method can be identified. This helps to improve and optimize more in the future. Indeed, tests were conducted to check the efficiency of the system. The results show that using T2FHC (Type-2 fuzzy CMAC and BELC) provides more accurate and stable position tracking than using CMAC and NSOFC. This again confirms the reliability of the theory when applied in practice, especially in nonlinear systems. However, this paper could be improved in several points.

One point that needs improvement is the flexibility of the parameters in the network. In this paper, the parameters are selected based on many experiments and practical experience. Research on improving the method of choosing parameters will have important implications in applying this method in practice.

In addition, continuing to research and test the system on many different models and conditions will also be a potential development direction. This will help to evaluate and determine the applicability of the method in real-life situations. In the process of implementing control systems in real applications, there will be limitations and challenges. One of the challenges is hardware constraints, including computational resource requirements and the ability to integrate with existing systems. Considering these limitations will help to come up with the right solution to deploy the control system effectively.

In addition, the operating environment is also an important factor to consider. The control system is subject to noise, variability, and heterogeneous environmental conditions. This poses a challenge in ensuring the stability and performance of the control system under extreme conditions.

Further experiments or simulations are required to confirm and evaluate the performance of the control system in more complex situations. This includes testing the system on more complex models, as well as evaluating the system's ability to manipulate different reference trajectories. Conducting these experiments and simulations will give a clearer view of the capabilities and limitations of the control system.

Finally, extend the actual research using the proposed method. Considering the applicability for robotic systems with a higher number of linked commands compared to a completely different system is a potential development direction. This will contribute to expanding the applicability of the method in areas such as robotics, automation systems and industrial applications.

In summary, the paper introduces the integrated learning network from Type-2 fuzzy CMAC and BELC and uses the strong control method sliding surface generator to control the uncertain nonlinear system. The experimental results show the efficiency of the proposed network and confirm the reliability of the theory. For improvement, flexibility of parameters and further study of different conditions are potential directions for this paper.

REFERENCES

[1] Y. Sun and T. C. Lueth, "Design of Bionic Prosthetic Fingers Using 3D Topology Optimization," *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pp. 4505-4508, 2021, doi: 10.1109/EMBC46164.2021.9630390.

[2] Y. Sun and T. C. Lueth, "SGCL: A B-Rep-Based Geometry Modeling Language in MATLAB for Designing 3D-Printable Medical Robots," *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, pp. 1388-1393, 2021, doi: 10.1109/CASE49439.2021.9551400.

[3] Y. Sun, Y. Liu, N. Zhou, and T. C. Lueth, "A MATLAB-Based Framework for Designing 3D Topology Optimized Soft Robotic Grippers," *2021 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pp. 1283-1289, 2021, doi: 10.1109/AIM46487.2021.9517350.

[4] N. Edoimioya and C. E. Okwudire, "A Generalized and Efficient Control-Oriented Modeling Approach for Vibration-Prone Delta 3D Printers Using Receptance Coupling," in *IEEE Transactions on Automation Science and Engineering*, vol. 20, no. 3, pp. 1539-1550, July 2023, doi: 10.1109/TASE.2022.3197057.

[5] A. Fathy, M. Ashraf, and A. El-Badawy, "Computed torque control of a prismatic-input delta parallel robot," *2022 4th Novel Intelligent and Leading Emerging Sciences Conference (NILES)*, pp. 137-141, 2022, doi: 10.1109/NILES56402.2022.9942389.

[6] O. Ersoy, M. C. Yildirim, A. Ahmad, O. D. Yirmibesoglu, N. Koroglu, and O. Bebek, "Design and Kinematics of a 5-DOF Parallel Robot for Beating Heart Surgery," *2019 IEEE 4th International Conference on Advanced Robotics and Mechatronics (ICARM)*, pp. 274-279, 2019, doi: 10.1109/ICARM.2019.8833632.

[7] M. Wang, J. -A. Leal-Naranjo, M. Ceccarelli, and S. Blackmore, "A Novel Two-Degree-of-Freedom Gimbal for Dynamic Laser Weeding: Design, Analysis, and Experimentation," in *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 6, pp. 5016-5026, Dec. 2022, doi: 10.1109/TMECH.2022.3169593.

[8] Z. Li and S. Li, "Saturated PI Control for Nonlinear System With Provable Convergence: An Optimization Perspective," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 2, pp. 742-746, Feb. 2021, doi: 10.1109/TCSII.2020.3007879.

[9] T. Q. Ngo, T. N. A. Nguyen, N. T. P. Le, D. C. Pham, and N. D. Ngo, "Adaptive Tracking Control Based on Recurrent Wavelet Fuzzy CMAC for Uncertain Nonlinear Systems," *International Journal of Control and Automation*, vol. 11, no. 1, pp. 75-90, 2018.

[10] A. -V. Nguyen, V. -T. Ngo, W. -J. Wang, V. -P. Vu, T. -Q. Ngo, and A. –. Nguyen, "Fuzzy Logic Based LQG Controller Design for Inverted Pendulum On Cart," *2021 International Conference on System Science and Engineering (ICSSE)*, pp. 387-392, 2021, doi: 10.1109/ICSSE52999.2021.9538411.

[11] M. A. Llama, R. Kelly, and V. Santibanez, "Stable computed-torque control of robot manipulators via fuzzy self-tuning," in *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 30, no. 1, pp. 143-150, Feb. 2000, doi: 10.1109/3477.826954.

[12] G. Guoyou, J. Chunsheng, C. Tao, H. Chun, W. Lina, and L. Zifan, "Research on Robot Position Control Based on Improved PD algorithm," *2019 34rd Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, pp. 437-439, 2019, doi: 10.1109/YAC.2019.8787668.

[13] Q. Li, J. Gao, Q. Wang, and R. Kennel, "Model Predictive Torque Control of Induction Motor Drives with Computed Torque for Servo Press," *2020 IEEE 9th International Power Electronics and Motion Control Conference (IPEMC2020-ECCE Asia)*, pp. 3063-3067, 2020, doi: 10.1109/IPEMC-ECCEAsia48364.2020.9368211.

[14] N. T. Minh Nguyet and D. X. Ba, "A Computed Torque Controller for Robotic Manipulators Using Nonlinear Neural Network," *2022 6th International Conference on Green Technology and Sustainable Development (GTSD)*, pp. 487-492, 2022, doi: 10.1109/GTSD54989.2022.9989043.

[15] Z. Chu, D. Zhu and S. X. Yang, "Observer-Based Adaptive Neural Network Trajectory Tracking Control for Remotely Operated Vehicle," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 7, pp. 1633-1645, July 2017, doi: 10.1109/TNNLS.2016.2544786.

[16] P. Agand and M. A. Shoorehdeli, "Adaptive Model Learning of Neural Networks with UUB Stability for Robot Dynamic Estimation," *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1-6, 2019, doi: 10.1109/IJCNN.2019.8851793.

[17] R. Bouzaiene, S. Hafsi, and F. Bouani, "Adaptive neural network PID controller for nonlinear systems," *2021 IEEE 2nd International Conference on Signal, Control and Communication (SCC)*, pp. 264-269, 2021, doi: 10.1109/SCC53769.2021.9768352.

[18] W. He, Y. Sun, Z. Yan, C. Yang, Z. Li, and O. Kaynak, "Disturbance Observer-Based Neural Network Control of Cooperative Multiple Manipulators With Input Saturation," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 5, pp. 1735-1746, May 2020, doi: 10.1109/TNNLS.2019.2923241.

[19] J. Boo and D. Chwa, "Fuzzy Integral Sliding Mode Observer-Based Formation Control of Mobile Robots With Kinematic Disturbance and Unknown Leader and Follower Velocities," in *IEEE Access*, vol. 10, pp. 76926-76938, 2022, doi: 10.1109/ACCESS.2022.3192839.

[20] W. Wang, Z. Han, K. Liu, and J. Lü, "Distributed Adaptive Resilient Formation Control of Uncertain Nonholonomic Mobile Robots Under Deception Attacks," in *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 9, pp. 3822-3835, Sept. 2021, doi: 10.1109/TCSI.2021.3096937.

[21] Z. Sun *et al*., "RBF Neural Network-Sliding Model Control Approach for Lower Limb Rehabilitation Robot Based on Gait Trajectories of SEMG Estimation," *2019 Tenth International Conference on Intelligent Control and Information Processing (ICICIP)*, pp. 14-19, 2019, doi: 10.1109/ICICIP47338.2019.9012202.

[22] X. Guo, Z. Li, and G. Sun, "The Robot Arm Control Based on RBF with Incremental PID and Sliding Mode Robustness," *2019 WRC Symposium on Advanced Robotics and Automation (WRC SARA)*, pp. 97-102, 2019, doi: 10.1109/WRC-SARA.2019.8931971.

[23] J. Shi, L. Xu, G. Cheng, J. Xu, S. Chen, and X. Liang, "Trajectory Tracking Control Based on RBF Neural Network of The Lower Limb Rehabilitation Robot," *2020 IEEE International Conference on Mechatronics and Automation (ICMA)*, pp. 117-123, 2020, doi: 10.1109/ICMA49215.2020.9233545.

[24] R. Xi, Z. Yang, and X. Xiao, "Adaptive Neural Network Observer Based PID-Backstepping Terminal Sliding Mode Control for Robot Manipulators," *2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pp. 209-214, 2020, doi: 10.1109/AIM43001.2020.9158859.

[25] V. -P. Ta, X. -K. Dang, and T. -Q. Ngo, "Adaptive tracking control based on CMAC for nonlinear systems," *2017 International Conference on System Science and Engineering (ICSSE)*, pp. 494-498, 2017, doi: 10.1109/ICSSE.2017.8030923.

[26] 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.

[27] T. -T. Huynh, C. -M. Lin, T. -L. Le, and Z. Zhong, "A Mixed Gaussian Membership Function Fuzzy CMAC for a Three-Link Robot," *2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp. 1-7, 2020, doi: 10.1109/FUZZ48607.2020.9177761.

[28] F. -J. Lin, I. -F. Sun, K. -J. Yang, and J. -K. Chang, "Recurrent Fuzzy Neural Cerebellar Model Articulation Network Fault-Tolerant Control of Six-Phase Permanent Magnet Synchronous Motor Position Servo Drive," in *IEEE Transactions on Fuzzy Systems*, vol. 24, no. 1, pp. 153-167, Feb. 2016, doi: 10.1109/TFUZZ.2015.2446535.

[29] T. -Q. Ngo, D. -K. Hoang, T. T. Tran, T. -T. Nguyen, V. -T Nguyen, and L. -H. Le, "A Novel Self-organizing Fuzzy Cerebellar Model Articulation Controller Based Overlapping Gaussian Membership Function for Controlling Robotic System," *International Journal Of Computers Communications & Control*, vol. 17, no. 4, 4606, 2022.

[30] T. -Q. Ngo, M. -K. Duong, D. C. Pham, and D. -N. Nguyen, "Adaptive Wavelet CMAC Tracking Control for Induction Servomotor Drive System," *Journal of Electrical Engineering & Technology*, vol. 14, no. 1, pp. 209-218, 2019.

[31] T. Q. Ngo, T. T. H. Le, B. M. Lam, and T. K. Pham, "Adaptive Single-Input Recurrent WCMAC-Based Supervisory Control for De-icing Robot Manipulator," *Journal of Robotics and Control (JRC)*, vol. 4, no. 4, pp. 438-451, 2023.

[32] C. -L. Hwang and W. -L. Fang, "Global Fuzzy Adaptive Hierarchical Path Tracking Control of a Mobile Robot With Experimental Validation," in *IEEE Transactions on Fuzzy Systems*, vol. 24, no. 3, pp. 724-740, June 2016, doi: 10.1109/TFUZZ.2015.2476519.

[33] C. -F. Wu, B. -S. Chen, and W. Zhang, "Multiobjective Investment Policy for a Nonlinear Stochastic Financial System: A Fuzzy Approach," in *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 2, pp. 460-474, April 2017, doi: 10.1109/TFUZZ.2016.2574926.

[34] H. S. Selaka, K. A. T. S. Perera, M. A. W. T. Deepal, P. D. R. Sanjeewa, H. P. C. Sirithunge, and A. G. B. P. Jayasekara, "Fuzzy-Bot: A Food Serving Robot as a Teaching and Learning Platform for Fuzzy Logic," *2018 Moratuwa Engineering Research Conference (MERCon)*, pp. 565-570, 2018, doi: 10.1109/MERCon.2018.8421898.

[35] N. Li, Y. Li, and W. Xiang, "The adaptive fuzzy tracking control for double inverted pendulums in the presence of unknown control directions," *2020 2nd International Conference on Industrial Artificial Intelligence (IAI)*, pp. 1-6, 2020, doi: 10.1109/IAI50351.2020.9262238.

[36] H. Su and W. Zhang, "Adaptive fuzzy control of MIMO nonlinear systems with fuzzy dead zones," *Applied Soft Computing*, vol. 80, pp. 700-711, 2019.

[37] K. Cahyono, I. K. Wibowo, and M. M. Bachtiar, "A New Kicker System of Wheeled Soccer Robot ERSOW Using Fuzzy Logic Method," *2020 International Electronics Symposium (IES)*, pp. 219-225, 2020, doi: 10.1109/IES50839.2020.9231882.

[38] F. Ardilla, M. I. Mas'udi, and I. K. Wibowo, "Fusion of Feedforward and Feedback Control Using Fuzzy for Active Handling and Dribbling System in MSL Robot Soccer," *International Journal on Advance Science, Engineering and Information Technology*, vol. 9, no. 6, pp. 1950-1958, 2019.

[39] S. Wang, "Mobile Robot Path Planning based on Fuzzy Logic Algorithm in Dynamic Environment," *2022 International Conference on Artificial Intelligence in Everything (AIE)*, pp. 106-110, 2022, doi: 10.1109/AIE57029.2022.00027.

[40] A. Alotaibi and S. Anwar, "A Fuzzy Logic Based Piezoresistive/Piezoelectric Fusion Algorithm for Carbon Nanocomposite Wide Band Strain Sensor," in *IEEE Access*, vol. 9, pp. 14752-14764, 2021, doi: 10.1109/ACCESS.2020.3049081.

[41] I. Reguii, I. Hassani, and C. Rekik, "Neuro-fuzzy Control of a Mobile Robot," *2022 IEEE 21st international Ccnference on Sciences and*

Techniques of Automatic Control and Computer Engineering (STA), pp. 45-50, 2022, doi: 10.1109/STA56120.2022.10018999.

[42] S. Hanana, N. Krichen, and M. S. Masmoudi, "Dual Fuzzy Logic Controllers for Omnidirectional Robot Navigation and Obstacle Avoidance," *2022 IEEE 21st international Ccnference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*, pp. 100-105, 2022, doi: 10.1109/STA56120.2022.10019080.

[43] M. N. Hidayati, D. Adzkiya, and H. Nurhadi, "Motion Control Design and Analysis of UR5 Collaborative Robots Using Fuzzy Logic Control (FLC) Method," *2021 International Conference on Advanced Mechatronics, Intelligent Manufacture and Industrial Automation (ICAMIMIA)*, pp. 162-167, 2021, doi: 10.1109/ICAMIMIA54022.2021.9807732.

[44] R. K. Ikbar, E. Mulyana, R. Mardiati, and R. R. Nurmalasari, "Fire Fighting Robot Using Flame Detector and Ultrasonic Based on Fuzzy Logic Control," *2022 16th International Conference on Telecommunication Systems, Services, and Applications (TSSA)*, pp. 1-6, 2022, doi: 10.1109/TSSA56819.2022.10063922.

[45] B. Shi, S. Xu, and G. Zhao, "Variable Universe Type-II Fuzzy Logic Control Design for the GOOGOL's Two-Wheeled Self-Balancing Robot," *2020 Chinese Automation Congress (CAC)*, pp. 1500-1505, 2020, doi: 10.1109/CAC51589.2020.9326544.

[46] C. -M. Lin, M. -S. Yang, F. Chao, X. -M. Hu, and J. Zhang, "Adaptive Filter Design Using Type-2 Fuzzy Cerebellar Model Articulation Controller," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 10, pp. 2084-2094, Oct. 2016, doi: 10.1109/TNNLS.2015.2491305.

[47] P. K. Muthusamy, M. Garratt, H. Pota, J. Wang, and J. M. Kok, "Bidirectional Fuzzy Brain Emotional Learning Control for Aerial Robots," *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 146-153, 2018, doi: 10.1109/SSCI.2018.8628809.

[48] C. -M. Lin and P. K. Muthusamy, "Intelligent brain emotional learning control system design for nonlinear systems," *2017 11th Asian Control Conference (ASCC)*, pp. 958-963, 2017, doi: 10.1109/ASCC.2017.8287300.

[49] Q. Wu *et al*., "Self-Organizing Brain Emotional Learning Controller Network for Intelligent Control System of Mobile Robots," in *IEEE Access*, vol. 6, pp. 59096-59108, 2018, doi: 10.1109/ACCESS.2018.2874426.

[50] M. Bajelani, S. A. Khalilpour, M. I. Hosseini, H. D. Taghirad, and P. Cardou, "Brain Emotional Learning based Intelligent Controller for a Cable-Driven Parallel Robot," *2021 9th RSI International Conference on Robotics and Mechatronics (ICRoM)*, pp. 37-42, 2021, doi: 10.1109/ICRoM54204.2021.9663454.

[51] G. Ran, C. Li, H. -K. Lam, D. Li, and C. Han, "Event-Based Dissipative Control of Interval Type-2 Fuzzy Markov Jump Systems Under Sensor Saturation and Actuator Nonlinearity," in *IEEE Transactions on Fuzzy Systems*, vol. 30, no. 3, pp. 714-727, March 2022, doi: 10.1109/TFUZZ.2020.3046335.

[52] M. A. Sanchez, J. R. Castro, V. O. Miramontes, and L. Cervantes, "Hybrid learning for general type-2 TSK fuzzy logic systems," *Algorithms*, vol. 10, no. 3, p. 99, 2017.

[53] K. Shiev, S. Ahmed, N. Shakev, and A. V. Topalov, "Trajectory control of manipulators using an adaptive parametric type-2 fuzzy cmac friction and disturbance compensator," *Novel Applications of Intelligent Systems*, pp. 63-82, 2016.

[54] P. D. Pour, K. M. J. Alsayegh, and M. A. Jaradat, "Type-2 Fuzzy Adaptive PID Controller for Differential Drive Mobile Robot: A Mechatronics Approach," *2022 Advances in Science and Engineering Technology International Conferences (ASET)*, pp. 1-6, 2022, doi: 10.1109/ASET53988.2022.9734882.

[55] Y. Saidi, M. Tadjine, and A. Nemra, "Robust Waypoints navigation using Fuzzy Type 2 Controller," 2019 International Conference on Advanced Electrical Engineering (ICAEE), pp. 1-6, 2019, doi: 10.1109/ICAEE47123.2019.9015101.

[56] O. Castillo and P. Melin, "An Approach for Optimization of Intuitionistic and Type-2 Fuzzy Systems in Pattern Recognition Applications," *2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp. 1-5, 2019, doi: 10.1109/FUZZ-IEEE.2019.8858951.

[57] A. Al-Mahturi, F. Santoso, M. A. Garratt, and S. G. Anavatti, "An Intelligent Control of an Inverted Pendulum Based on an Adaptive Interval Type-2 Fuzzy Inference System," *2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp. 1-6, 2019, doi: 10.1109/FUZZ-IEEE.2019.8858948.

[58] F. Cuevas, O. Castillo, and P. C. Antonio, "Towards an Adaptive Control Strategy Based on Type-2 Fuzzy Logic for Autonomous Mobile Robots," *2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp. 1-6, 2019, doi: 10.1109/FUZZ-IEEE.2019.8858801.

[59] W. -C. Su, C. -F. Juang, and C. -M. Hsu, "Multiobjective Evolutionary Interpretable Type-2 Fuzzy Systems With Structure and Parameter Learning for Hexapod Robot Control," in *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 5, pp. 3066-3078, May 2022, doi: 10.1109/TSMC.2021.3063778.

[60] F. Chao, D. Zhou, C. -M. Lin, L. Yang, C. Zhou, and C. Shang, "Type-2 Fuzzy Hybrid Controller Network for Robotic Systems," in *IEEE Transactions on Cybernetics*, vol. 50, no. 8, pp. 3778-3792, Aug. 2020, doi: 10.1109/TCYB.2019.2919128.

[61] C.-M. Lin and C.-C. Chung, "Fuzzy brain emotional learning control system design for nonlinear systems," *Int. J. Fuzzy Syst*., vol. 17, no. 2, pp. 117–128, 2015.