Journal of Intelligent & Fuzzy Systems 36 (2019) 2465–2480
DOI:10.3233/JIFS-171523
IOS Press

Dynamic TOPSIS fuzzy cerebellar model articulation controller for magnetic levitation system

Chih-Min Lin[a,*] and Tuan-Tu Huynh[a,b]
[a]*Department of Electrical Engineering, Yuan Ze University, Chung-Li, Taoyuan, Taiwan, R.O.C.*
[b]*Department of Electrical Electronic and Mechanical Engineering, Lac Hong University, Bien Hoa, Dong Nai, Vietnam*

Abstract. This study proposes a fuzzy Cerebellar Model Articulation Controller (CMAC) using a dynamic Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) technique for dealing with the metallic sphere position control of a magnetic levitation system (MLS). The proposed Dynamic TOPSIS Fuzzy CMAC (DTFCMAC) incorporates a multi-criteria decision analysis with a fuzzy structure to decrease the computational load for parameter learning and to enhance the fuzzy reasoning inference for a CMAC. The Shannon entropy index is used to derive the objective weights for the evaluation criterion. By combining entropy weight and TOPSIS, the optimal threshold value for suitable firing nodes is determined automatically and easily. In the proposed method, the dynamic back-propagation algorithm is applied to train the proposed DTFCMAC online. Moreover, to guarantee the convergence of output tracking error for periodic command tracking, analytical methods developed from a discrete-type Lyapunov function are used to determine the optimal learning-rate parameters for the proposed DTFCMAC. The proposed DTFCMAC is applied to the MLS, and its performance is verified through simulations and experiments. Our findings indicate that the proposed DTFCMAC control system achieves stability and desired control performance for the MLS.

Keywords: Dynamic, TOPSIS, entropy, fuzzy inference system, cerebellar model articulation controller, and magnetic levitation system

1. Introduction

The magnetic levitation system (MLS) produces magnetic force using the electric current flowing through a coil. This technique is used to suspend an object in the air without incurring mechanical contact, friction or noise and can be used for precise positioning. Because of these advantages, it has many applications, such as for maglev trains, magnetic bearings, wind tunnels, and conveyor systems. With the rapid development of industrial technology, MLSs have become increasingly widespread. How to design a simple and effective control algorithm is an important issue of MLSs, which has attracted much research interest.

Over a decade, many new control methods have been proposed for MLSs. In 2006, Chiang et al. proposed the concept of integral variable structure grey control [1]. Yang et al. [2] published the method of adaptive robust output feedback control [2] and a robust nonlinear output feedback control in 2009 [3]. In addition, a robust dynamic sliding mode controller using adaptive recurrent neural network was recommended by Lin et al. [4]. Then in 2011,

*Corresponding author. Chih-Min Lin, Yuan Ze University, Chung-Li, Taoyuan, 320 Taiwan, R.O.C. Tel.: +886 3 4638800 2209; Fax: +886 3 4536022; E-mail: cml@saturn.yzu.edu.tw.

Lin et al. offered an adaptive PID controller [5]. Three years later, Lin et al. introduced a function-link cerebellar model articulation controller (CMAC) [6]. Complementing these control methods, Lin et al. [7] proposed a Dynamic Petri Fuzzy CMAC (DPFCMAC) [7]. However, these controllers have some drawbacks. They are difficult to explain and sensitive to the training data, and require complex computation for better control performance. In DPFCMAC introduced by Lin et al. [7], the threshold value of receptive-field basis function in association memory space is between 0 and 0.2 so the values of receptive-field basis function in association memory space must have a value that is larger than the threshold value. Lin et al. [7] demonstrated that if this threshold value is small, the tracking error is large. A more detail of this threshold value is presented in the following section.

Various Multi-Criteria-Decision-Making (MCDM) approaches have been developed and applied to solve many problems [8–10]. As one of the well-known classical MCDM approaches, TOPSIS (technique for order preference by similarity to ideal solution) method was firstly developed by Hwang and Yoon [11]. For TOPSIS, the most preferred alternative must have the shortest distance from the positive ideal solution and the farthest distance from the negative ideal solution. In this work, the TOPSIS method involves the following seven steps: (1) create an evaluation matrix, (2) normalize the decision matrix, (3) compute the weighted normalized decision matrix, (4) determine the worst alternative ($\boldsymbol{A}'$) and the best alternative ($\boldsymbol{A}^*$), (5) calculate the separations of an alternative from $\boldsymbol{A}^*$ and $\boldsymbol{A}'$, (6) calculate the threshold index (i.e., the relative proximity of an alternative to $\boldsymbol{A}^*$), and (7) select the optimal threshold index to obtain the appropriate values of criteria. Generally, the advantages of TOPSIS include: (a) simple and logically understandable model, (b) respectable computational efficiency and (c) the ability to compute the relative performance for each alternative in a simple mathematical form.

A cerebellar model articulation controller (CMAC) is classified as a non-fully connected perceptron-like associative memory network with overlapping receptive fields [12]. It serves to resolve the fast size-growing problem and the learning difficulty in currently available types of neural networks (NNs). In recent decades, CMAC has many advantages more than NNs [13]. CMACs have been widely applied for the control of complex dynamic systems because they allow simple computation and have good generalization capability and fast learning properties [14, 15]. Because of these advantages, this study incorporates a TOPSIS with a fuzzy CMAC into a novel Dynamic TOPSIS Fuzzy CMAC (DTFCMAC) to determine suitable firing nodes, based on threshold values. In recent years, to achieve better learning performance for CMACs, Petri net has been used to construct a DPFCMAC [7]. However, the threshold value is chosen between 0 and 0.2 through trial-and-error. In this study, a TOPSIS is combined with entropy method to determine the threshold value automatically. Because the TOPSIS transition space filters unimportant input variables, it alleviates the computational burden that accompanies the learning of parameters and increases the number of fuzzy reasoning inference of a multi-input CMAC. The proposed DTFCMAC combines the advantages of CMAC, fuzzy system and TOPSIS. In this study, the proposed DTFCMAC is applied to control an MLS. Simulation and experimental results of the MLS are presented to confirm the validity of the proposed control strategy.

Notations: $\Re^{m_i \times n_j}$ defines all real matrices with dimension $m_i \times n_j$. "*" and "′" represent the optimal term and worst term, respectively. m_{ijk}, v_{ijk}, and f_{ijk} are a mean parameter, a variance parameter and a value of receptive-field basis function with the i-th input, the j-th layer and the k-th block, respectively. In addition, $\boldsymbol{F}$ is an evaluation matrix, e_{n_y} is an entropy index of criterion n_y, dd_{n_y} is the degree of diversification of the measurement quality, w_{n_y} is the objective weight for each attribute, $\mathbf{CC}^*_{m_x}$ is the optimal value of the relative closeness to the ideal solution, b_{th} is the dynamic threshold for selecting appropriate firing rules, and μ_{ijk} is the suitable firing value in the TOPSIS transition space. γ_w, γ_m, and γ_v are the learning-rate parameters of the weight, the mean, and the variance of the Gaussian function, respectively. λ_0 is the error term.

2. Description of magnetic levitation system

2.1. Magnetic levitation system for experiment

The magnetic levitation system (MLS) is nonlinear and unstable, and it is sensitive to initial conditions and noises [5]. The system is divided into two parts: the peripheral circuit and the MLS, as shown in Fig. 1. The proposed DTFCMAC is implemented with the PCI-1716 multi-function card, which has 250 kS/s, 16-bit, 16-ch, an onboard 1K sample FIFO buffer for

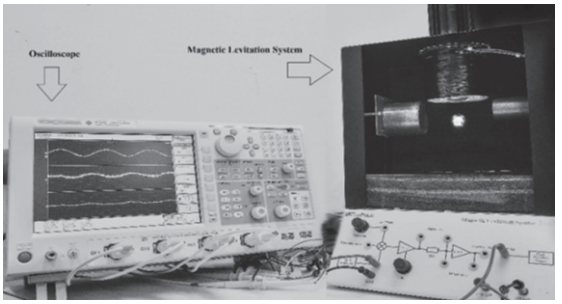

Fig. 1. Hardware experimental environment.

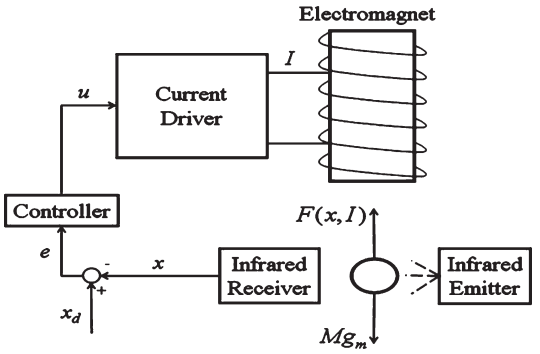

Fig. 2. The construction of magnetic levitation system [5].

A/D, 6 single-ended or 8 differential or a combination of analog inputs, 16-bit A/D converter, with up to 250 kHz sampling rate, 2 analog output channels, 16-ch digital input, and 16-ch digital output.

2.2. Modeling of magnetic levitation system for simulation [4–6]

Figure 2 shows a single-axis MLS. The control input is a voltage, which is converted into a current through the current driver. When current flows, the electromagnet creates a corresponding magnetic field in its surroundings. The metallic sphere moves along the vertical axis of the electromagnet. The position of the metallic sphere is measured by an infrared sensor.

According to Newton's second law of motion, the behavior of the metallic sphere is specified by the following equation:

$$Ma = F(x, I) - Mg_m, \tag{1}$$

where $M(kg)$ is the mass of the metallic sphere, $g_m(m/s^2)$ denotes the acceleration due to gravity, $a(m/s^2)$ is the acceleration, $x(m)$ is the distance of the sphere from the electromagnet, $I(A)$ is the current, and $F(x, I)(N)$ is the magnetic control force.

The magnetic control force of MLS in an $(x - I)$ equation is expressed as:

$$F(x, I) = \frac{\mu_0 N^4 I^2 S}{8} \left[(x+l) \ln \left| \frac{R_2 + \sqrt{R_2^2 + (x+h)^2}}{R_1 + \sqrt{R_1^2 + (x+h)^2}} \right| + x \ln \left| \frac{R_1 + \sqrt{R_1^2 + x^2}}{R_2 + \sqrt{R_2^2 + x^2}} \right| \right]^2, \tag{2}$$

where $\mu_0(Wb/Am)$ is the permeability of free space, $h(m)$ is the length of coil, N is the number of turns per meter, $S(m^2)$ is the material surface crossed by the magnetic flux, $R_1(m)$ is the minimum coil radius, and $R_2(m)$ is the maximum coil radius.

Equations (1 and 2) are used for simulation only in this work. During the design and practical implementation of the proposed controller, these equations are not required.

3. TOPSIS method and structure of the proposed DTFCMAC

The system dynamics of an MLS is intricate and highly nonlinear; hence, it is difficult to design a suitable control scheme without mathematical dynamic modeling for achieving highly precise position control. Thus, the proposed DTFCMAC is used, which incorporates the TOPSIS method and the fuzzy inference system with a CMAC. The structure of the proposed DTFCMAC is presented as follows:

3.1. Dynamic TOPSIS fuzzy cerebellar model articulation controller

This work proposes a novel DTFCMAC as shown in Fig. 3. This proposed DTFCMAC uses the following fuzzy inference rules:

$$R^l: \text{If } I_1 \text{ is } f_{1jk}, I_2 \text{ is } f_{2jk}, \ldots, \text{ and } I_{n_i} \text{ is } f_{n_i jk}, \text{ then } o_{jk} = w_{jk} \text{ for } j = 1, 2, \ldots, n_j, \ k = 1, 2, \ldots, n_k, \text{ and } l = 1, 2, \ldots, n_l, \tag{3}$$

where n_i is the input dimension, n_j is the number of layers for each input dimension, n_k is the number of blocks

for each layer, n_l is the number of fuzzy rules, f_{ijk} is the fuzzy set for the i-th input, j-th layer and k-th block, and w_{jk} is a singleton output weight in the consequent part. A schematic diagram for a two-dimensional ($n_i = 2$) DTFCMAC with four layers ($n_j = 4$), with three blocks ($n_k = 3$) in each layer, is described in Fig. 4.

A DTFCMAC comprises an input space, an association memory space, a TOPSIS transition space, a receptive field space, a weight memory space and an output space. The signal propagation in each space is described as follows:

1) Input: Set a vector $I = [I_1, \ldots, I_i, \ldots, I_{n_i}]^T \in \Re^{n_i}$, each input state variable I_i is quantized into discrete regions (called elements or neurons) according to the specific control space. The number of elements n_e is termed as a resolution; for example, $n_e = 9$ in Fig. 4.

2) Association Memory Space (Membership Function): Several elements are accumulated as a block. In this space, each block performs a receptive-field basis function. The Gaussian function is used as a receptive-field basic function, which is represented as:

$$f_{ijk}(F_{ijk}) = \exp(-F_{ijk}^2), \quad \text{for } i = 1, 2, \ldots, n_i,\ j = 1, 2, \ldots, n_j, \quad \text{and } k = 1, 2, \ldots,\ n_k, \tag{4}$$

$$\text{where } F_{ijk} = \frac{I_i - m_{ijk}}{v_{ijk}},$$

$m_i jk$ is a mean parameter, and $v_i jk$ is a variance parameter.

3) *TOPSIS Transition Space:* TOPSIS is a multi-criteria decision analysis method, that was firstly developed by Hwang and Yoon in 1981 [11], this method was improved by Hwang et al. in 1993 [16]. The main technique of TOPSIS method has been presented in [11, 17]. In this space, TOPSIS method is used to determine the optimal threshold value for suitable firing nodes. In addition, the entropy method is used to derive the weights for the evaluation criteria. All the steps of TOPSIS are described in detail as following:

Table 1 includes all of the values for the receptive-field basis function in association memory space. Let $A_1, A_2, \ldots, A_m$ be the alternatives and $C_1, C_2, \ldots, C_n$ be the criteria. Alternatives are the layers and criterion is the receptive-field basis function f_{ijk} or a block in the CMAC of the proposed control system. Each alternative has n_k blocks. Each block has a specific value f_{ijk} and each of these values is calculated in Equation (4).

i. Create an evaluation matrix:
Set the matrix $\boldsymbol{F} = (f_{ijk})_{(m\times n)} = (q_{m_x n_y})_{(m\times n)}$,

$$F = \begin{bmatrix} f_{111} & f_{112} & f_{113} & \cdots & f_{11n_k} \\ f_{121} & f_{122} & f_{123} & \cdots & f_{12n_k} \\ f_{131} & f_{132} & f_{133} & \cdots & f_{13n_k} \\ f_{141} & f_{142} & f_{143} & \cdots & f_{14n_k} \\ f_{211} & f_{212} & f_{213} & \cdots & f_{21n_k} \\ f_{221} & f_{222} & f_{223} & \cdots & f_{22n_k} \\ f_{231} & f_{232} & f_{233} & \cdots & f_{23n_k} \\ f_{241} & f_{242} & f_{243} & \cdots & f_{24n_k} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ f_{ij1} & f_{ij2} & f_{ij3} & \cdots & f_{ijn_k} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ f_{n_i n_j n_1} & f_{n_i n_j n_2} & f_{n_i n_j n_k} & \cdots & f_{n_i n_j n_k} \end{bmatrix}$$

$$= \begin{bmatrix} q_{11} & q_{12} & q_{13} & \cdots & q_{1n_y} & \cdots & q_{1n} \\ q_{21} & q_{22} & q_{23} & \cdots & q_{2n_y} & \cdots & q_{2n} \\ q_{31} & q_{32} & q_{33} & \cdots & q_{3n_y} & \cdots & q_{3n} \\ q_{41} & q_{42} & q_{43} & \cdots & q_{4n_y} & \cdots & q_{4n} \\ q_{51} & q_{52} & q_{53} & \cdots & q_{5n_y} & \cdots & q_{5n} \\ q_{61} & q_{62} & q_{63} & \cdots & q_{6n_y} & \cdots & q_{6n} \\ q_{71} & q_{72} & q_{73} & \cdots & q_{7n_y} & \cdots & q_{7n} \\ q_{81} & q_{82} & q_{83} & \cdots & q_{8n_y} & \cdots & q_{8n} \\ \vdots & \vdots & \vdots & & \vdots & \cdots & \vdots \\ q_{m_x 1} & q_{m_x 2} & q_{m_x 3} & \cdots & q_{m_x n_y} & \cdots & q_{m_x n} \\ \vdots & \vdots & \vdots & & \vdots & \cdots & \vdots \\ q_{m1} & q_{m2} & q_{m3} & \cdots & q_{mn_y} & \cdots & q_{mn} \end{bmatrix} \tag{5}$$

where m is the number of alternatives and n is the number of criteria, $i = 1, 2, \ldots, n_i,\ j = 1, 2, \ldots, n_j$, and $k = 1, 2, \ldots, n_k$.

In this study, the number of inputs, the number of layers and the number of blocks are $n_i = 2,\ n_j = 4$ and $n_k = 3$, respectively. Therefore, the proposed DTFCMAC has eight layers in total and each layer has three blocks; thus in this case, the matrix F has eight alternatives ($m = 8$) and each alternative has three criteria ($n = 3$).

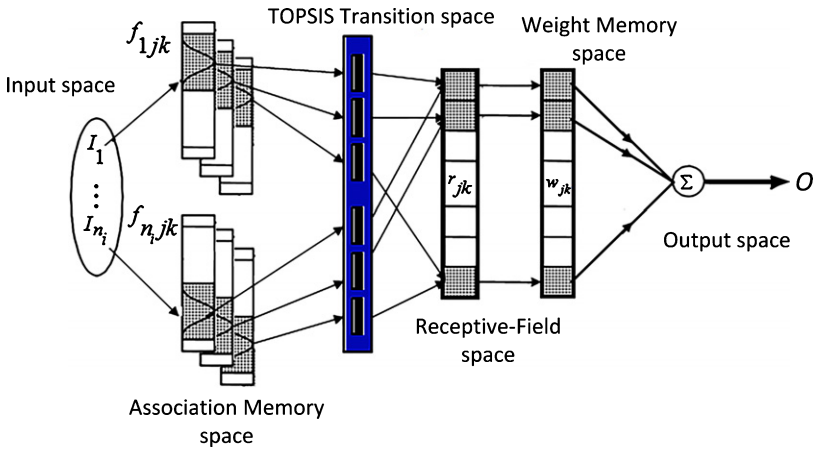

Fig. 3. Architecture of the proposed DTFCMAC.

The alternatives and criteria in matrix F are expressed as:

$$A_1 = [\,q_{11}\ q_{12}\ q_{13}\,];$$
$$A_2 = [\,q_{21}\ q_{22}\ q_{23}\,]$$
$$A_3 = [\,q_{31}\ q_{32}\ q_{33}\,]$$
$$A_4 = [\,q_{41}\ q_{42}\ q_{43}\,]$$
$$A_5 = [\,q_{51}\ q_{52}\ q_{53}\,] \quad C_1 = \begin{bmatrix} q_{11} \\ q_{21} \\ q_{31} \\ q_{41} \\ q_{51} \\ q_{61} \\ q_{71} \\ q_{81} \end{bmatrix};$$
$$A_6 = [\,q_{61}\ q_{62}\ q_{63}\,]$$
$$A_7 = [\,q_{71}\ q_{72}\ q_{73}\,]$$
$$A_8 = [\,q_{81}\ q_{82}\ q_{83}\,] \tag{6}$$

$$C_2 = \begin{bmatrix} q_{12} \\ q_{22} \\ q_{32} \\ q_{42} \\ q_{52} \\ q_{62} \\ q_{72} \\ q_{82} \end{bmatrix} \qquad C_3 = \begin{bmatrix} q_{13} \\ q_{23} \\ q_{33} \\ q_{43} \\ q_{53} \\ q_{63} \\ q_{73} \\ q_{83} \end{bmatrix}$$

ii. The matrix $\boldsymbol{F} = Q_{m_x n_y}$ is then normalized

If the value of matrix F has different dimensions, it must be normalized. This implies that all the values of $Q_{m_x n_y}$ are between 0 and 1.

$$Q_{m_x n_y} = \frac{q_{m_x n_y}}{\sqrt{\sum_{m_x=1}^{m_x=m} q_{m_x n_y}^2}}, \text{ for } m_x = 1, \ldots, m, \text{ and } n_y = 1, \ldots, n \tag{7}$$

- $q_{m_x n_y}$ is a quality value at criterion n_y for alternative m_x of input I_i.
- $\sqrt{\sum_{m_x=1}^{m_x=m} q_{m_x n_y}^2}$ is the summation of all quality values $q_{m_x n_y}$ at criterion n_y for all alternatives m_x of input I_i.
- $Q_{m_x n_y}$ is the measurement quality value. It is the ratio of quality value $q_{m_x n_y}$ to the sum of all quality values.

iii. Calculate the weighted normalized decision matrix:

$$v_{m_x n_y} = \left(w_{n_y} Q_{m_x n_y}\right)_{m \times n} \quad \text{for } m_x = 1, \ldots, m, \text{ and } n_y = 1, \ldots, n, \tag{8}$$

$$\sum_{n_y=1}^{n} w_{n_y} = 1 \text{ and } 0 < w_{n_y} < 1. \tag{9}$$

Determination of the Evaluation Criteria Weights

Shannon has proposed the entropy concept, which has been highlighted by Zeleny, for deciding the objective weights of criteria [18, 19]. In this study, the Shannon entropy method is used to derive the weights for the evaluation criteria. An entropy can be defined as [18, 19]

$$\mathbf{H}(x_1, x_2, \ldots, x_{n_k}) = -\sum_{i=1}^{n_x} x_i \log(x_i). \tag{10}$$

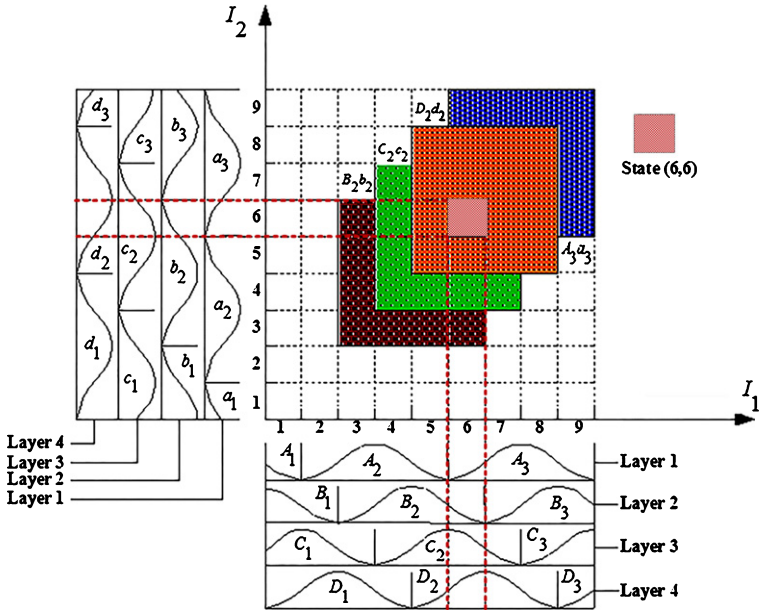

Fig. 4. The organization of a 2-D fuzzy CMAC.

Table 1
Decision table in the TOPSIS method

Alternatives A_m	Layers	Criteria C_n	Values of criteria
		n_k blocks	
Alternative 1	Layer 1	$A_1, A_2, A_3, \ldots, A_{n_k}$	$f_{111} f_{112} f_{113} \ldots f_{11n_k}$
Alternative 2	Layer 2	$B_1, B_2, B_3, \ldots, B_{n_k}$	$f_{121} f_{122} f_{123} \ldots f_{12n_k}$
Alternative 3	Layer 3	$C_1, C_2, C_3, \ldots, C_{n_k}$	$f_{131} f_{132} f_{133} \ldots f_{13n_k}$
Alternative 4	Layer 4	$D_1, D_2, D_3, \ldots, D_{n_k}$	$f_{141} f_{142} f_{143} \ldots f_{14n_k}$
Alternative 5	Layer 5	$a_1, a_2, a_3, \ldots, a_{n_k}$	$f_{211} f_{212} f_{213} \ldots f_{21n_k}$
Alternative 6	Layer 6	$b_1, b_2, b_3, \ldots, b_{n_k}$	$f_{221} f_{222} f_{223} \ldots f_{22n_k}$
Alternative 7	Layer 7	$c_1, c_2, c_3, \ldots, c_{n_k}$	$f_{231} f_{232} f_{233} \ldots f_{23n_k}$
Alternative 8	Layer 8	$d_1, d_2, d_3, \ldots, d_{n_k}$	$f_{241} f_{242} f_{243} \ldots f_{24n_k}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
Alternative m_x	Layer m_x	$x_1, x_2, x_3, \ldots, x_{n_k}$	$f_{ij1} f_{ij2} f_{ij3} \ldots f_{ijn_k}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
Alternative m	Layer m	$z_1 z_2 z_3 \ldots z_{n_k}$	$f_{n_i n_j 1} f_{n_i n_j 2} f_{n_i n_j 3} \ldots f_{n_i n_j n_k}$

where x_i $(i = 1, 2, \ldots, n_x)$ are the probabilities of random variable being computed from a probability mass function X. Entropy is a measurement of the degree of disorder in a system. The entropy value e_{n_y} of criterion n_y is defined as

$$e_{n_y} = -\frac{1}{\ln(m)} \sum_{m_x=1}^{m} Q_{m_x n_y} \ln\left(Q_{m_x n_y}\right), \quad \text{for } m_x = 1, \ldots, m, \text{ and } n_y = 1, \ldots, n, \tag{11}$$

where $e_{n_y} \in [0, 1]$, e_{n_y} is an entropy index of criterion n_y, which is a sum of all measurement quality values $Q_{m_x n_y}$ at criterion n_y for all alternatives, and the constant $k = \frac{1}{\ln(m)}$ is used to guarantee $0 \le e_{n_y} \le 1$.

For a given criterion n_y, the larger the diversification of $Q_{m_x n_y}$ becomes, the smaller e_{n_y} is; this means the more important is the role that is played by criterion n_y in the comparison. Therefore, set

$$dd_{n_y} = 1 - e_{n_y}, n_y \in [1, n], \tag{12}$$

where dd_{n_y} is the degree of diversification of the measurement quality, and the larger dd_{n_y} is, the more important the criterion is.

The objective weight for each criterion is calculated as:

$$w_{n_y} = \frac{dd_{n_y}}{\sum_{n_y=1}^{n} dd_{n_y}}, \tag{13}$$

and it is viewed as the degree of importance for criterion n_y.

Finally, by following these steps, the entropy weight vector $\mathbf{W} = [w_1, w_2, \ldots, w_{n_y}, \ldots, w_n]$ is obtained.

iv. Determine the worst alternative (A') and the best alternative (A^*):

This step determines $\boldsymbol{A}^*$ and $\boldsymbol{A}'$

$$\boldsymbol{A}^* = \left\{ v_1^*, \ldots, v_{n_y}^*, \ldots, v_n^* \right\}, \tag{14}$$

where

$$v_{n_y^*} = \left\{ \max(\boldsymbol{v}_{m_x n_y}) \text{ if } n_y \in \bar{\boldsymbol{J}}^*; \min(\boldsymbol{v}_{m_x n_y}) \text{ if } n_y \in \underline{\boldsymbol{J}}' \right\}.$$

$$\boldsymbol{A}' = \{v_{1'}, \ldots, v'_{n_y}, \ldots, v_{n'}\}, \tag{15}$$

where

$$v_{n_y'} = \{\min(\boldsymbol{v}_{m_x n_y}) \text{ if } n_y \in \bar{\boldsymbol{J}}^*; \max(\boldsymbol{v}_{m_x n_y}) \text{ if } n_y \in \underline{\boldsymbol{J}}'\}.$$

$\bar{\boldsymbol{J}}^*$ is a set of positive criteria or positive impacts.
$\underline{\boldsymbol{J}}'$ is a set of negative criteria or negative impacts.

v. Calculate the separation distance for each alternative:

The separation from the best alternative is:

$$\mathbf{S}_{\mathbf{m}_x^*} = \sqrt{\sum_{n_y=1}^{n} \left(v_{n_y^*} - \mathbf{v}_{m_x n_y} \right)^2}$$

$$\text{for } m_x = 1, \ldots, m, \text{ and } n_y = 1, \ldots, n, \tag{16}$$

Similarly, the separation from the worst alternative is:

$$\mathbf{S}'_{\mathbf{m}_x} = \sqrt{\sum_{n_y=1}^{n} \left(v'_{n_y} - \mathbf{v}_{m_x n_y} \right)^2} \tag{17}$$

$$\text{for } m_x = 1, \ldots, m, \text{ and } n_y = 1, \ldots, n,$$

vi. Calculate the relative closeness to the ideal solution $\mathbf{CC}_{m_x}^*$:

$$\mathrm{CC}_{m_x}^* = \frac{\mathbf{S}'_{m_x}}{\left(\mathbf{S}_{m_x}^* + \mathbf{S}'_{m_x} \right)}, \ 0 < \mathbf{CC}_{m_x}^* < 1, \tag{18}$$

where $\mathbf{CC}_{m_k}^* = \left[CC_1^*, CC_2^*, \ldots, CC_{m_k}^*, \ldots, CC_m^* \right]$.

$\mathbf{CC}_{m_x}^* = 1$ if and only if the alternative solution has the best condition.

$\mathbf{CC}_{m_x}^* = 0$ if and only if the alternative solution has the worst condition.

This step is to find the optimal value of $\mathbf{CC}_{m_x}^*$, which is also the threshold value of f_{ijk} for selecting the suitable firing rule.

In this study, the threshold values are calculated as:

$$b_{th} = \max(\mathbf{CC}_{m_k}^*) \tag{19}$$

Particularly, if the threshold value is set manually to obtain the fuzzy rules, the proposed DTFCMAC becomes:

1. If the threshold value $b_{th} = 0$ is set, the proposed DTFCMAC is the same as the FCMAC [20] because all the rules are fired at this case.
2. If the threshold value $b_{th} = \beta_c$ is set, the proposed DTFCMAC is the same as the DPFCMAC [7].

vii. Select the best value of μ_{ijk}:

The optimal threshold value for selecting a suitable firing node is expressed as:

$$\mu_{ijk} = \begin{cases} f_{ijk}, & f_{ijk} \geqslant b_{th} \\ 0, & f_{ijk} < b_{th}, \end{cases} \tag{20}$$

where μ_{ijk} is the suitable firing value and b_{th} is a dynamic threshold value.

If the values of the receptive-field basis function in association memory space are larger than the threshold value, the fuzzy rules are fired and their values are unaltered. Otherwise, the fuzzy rules are not activated and their values are set to zero. Equation (20) is applied to calculate the multi-dimensional receptive-field function presented in Equation (21) and the error term in Section 4.2.

4) Receptive-field space (hypercube space): The multi-dimensional receptive-field function is determined as:

$$r_{jk} = \prod_{i=1}^{n_i} \mu_{ijk} = \prod_{i=1}^{n_i} f_{ijk}(F_{ijk}) = \prod_{i=1}^{n_i} \exp\left[- \left(\frac{I_i - m_{ijk}}{v_{ijk}} \right)^2 \right] \tag{21}$$

$$\text{for } i = 1, 2, \ldots, n_i, \ j = 1, 2, \ldots, n_j \text{ and } k = 1, 2, \ldots, n_k,$$

where r_{jk} is associated with the j th layer and the kth block.

The multi-dimensional receptive-field functions are expressed in vector form as:

$$r = [r_{11} \dots r_{1n_k},\ r_{21} \dots r_{2n_k}, r_{n_j1} \dots r_{n_jn_k}]^T \in \Re^{n_jn_k} = [r_1, \dots, r_l, \dots, r_{n_l}]^T \in \Re^{n_l}. \tag{22}$$

5) Weight memory: Each location for the receptive-field to a particular adjustable value in the weight memory space is expressed as:

$$w = [w_{11} \dots w_{1n_k}, w_{21} \dots w_{2n_k}, w_{n_j1} \dots w_{n_jn_k}]^T \in \Re^{n_jn_k} = [w_1, \dots, w_l, \dots w_{n_l}]^T \in \Re^{n_l}, \tag{23}$$

where w_{jk} denotes the connecting weight value for the output that is associated with the jth layer and kth block.

6) Output: The output for the proposed DTFCMAC is the algebraic sum of the activated weighted receptive-field, and is expressed as:

$$u_{DTFCMAC} = o = w^T r = \sum_{j=1}^{n_j} \sum_{k=1}^{n_k} w_{jk} r_{jk} = \sum_{l=1}^{n_l} w_l r_l. \tag{24}$$

4. The DTFCMAC control system for MLS

4.1. The DTFCMAC-based feedback control system

This section introduces the proposed DTFCMAC control system for effective control of the sphere position of an MLS. The block diagram of the proposed DTFCMAC control system for the MLS is depicted in Fig. 5, where x_d is the reference signal and x is the tracking output signal of the MLS. The inputs of the proposed DTFCMAC control system are the tracking error $e(K) = x_d(K) - x(K)$ and the derivative of tracking error $\Delta e(K) = e(K+1) - e(K)$, and the output of the proposed DTFCMAC control system is $u_{DTFCMAC}$. The newly designed DTFCMAC structure is described in Fig. 3, with the TOPSIS transition layer being the most important difference between the FCMAC and the proposed DTFCMAC.

4.2. Online learning algorithm

In this study, to point out the online learning algorithm of the proposed DTFCMAC, we use the supervised gradient descent method. The energy function E is defined as:

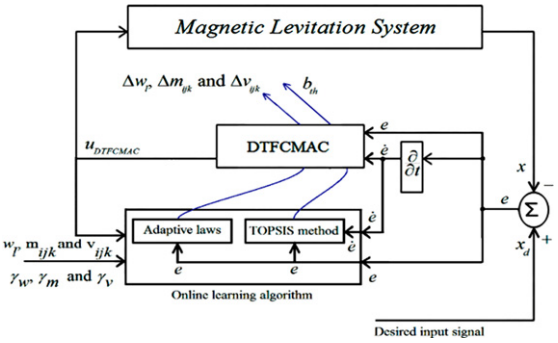

Fig. 5. Block diagram of DTFCMAC control system.

$$E = \frac{1}{2}(x_d - x)^2 = \frac{1}{2}e^2. \tag{25}$$

With the energy function E, the error term to be propagated is specified by:

$$\lambda_0 = -\frac{\partial E}{\partial u_{DTFCMAC}} = \frac{\partial E}{\partial e}\frac{\partial e}{\partial x}\frac{\partial x}{\partial u_{DTFCMAC}} = e\frac{\partial x}{\partial u_{DTFCMAC}}, \tag{26}$$

and the updated weight is derived using the following equation:

$$\Delta w_l = -\gamma_w \frac{\partial E}{\partial w_l} = -\gamma_w \frac{\partial E}{\partial u_{DTFCMAC}}\frac{\partial u_{DTFCMAC}}{\partial w_l} = \gamma_w \lambda_0 r_l, \tag{27}$$

where γ_w is a learning-rate parameter for all w_l. The connective weights are updated according to the following equation:

$$w_l(K+1) = w_l(K) + \Delta w_l(K), \tag{28}$$

where K denotes the number of iterations. When the weights in the rule layer are unified, the error term is calculated and propagated by the following equation:

$$\xi_l = \frac{\partial E}{\partial r_l} = \left[-\frac{\partial E}{\partial u_{DTFCMAC}}\right]\left(\frac{\partial u_{DTFCMAC}}{\partial r_l}\right) = \begin{cases} \lambda_0 w_l, & r_l \neq 0 \\ 0, & r_l = 0. \end{cases} \tag{29}$$

In TOPSIS layer, the error term is calculated as the following equation, and based on the Equation (20), it is obtained as:

$$\rho_{ijk} = -\frac{\partial E}{\partial f_{ijk}} = \left[-\frac{\partial E}{\partial u_{DTFCMAC}}\frac{\partial u_{DTFCMAC}}{\partial r_l}\right]\left(\frac{\partial r_l}{\partial f_{ijk}}\right) = \begin{cases} \xi_l r_l = \lambda_0 w_l r_l, & \mu_{ijk} = f_{ijk},\ f_{ijk} \geqslant b_{th} \\ 0, & \mu_{ijk} = 0,\ f_{ijk} < b_{th}. \end{cases} \tag{30}$$

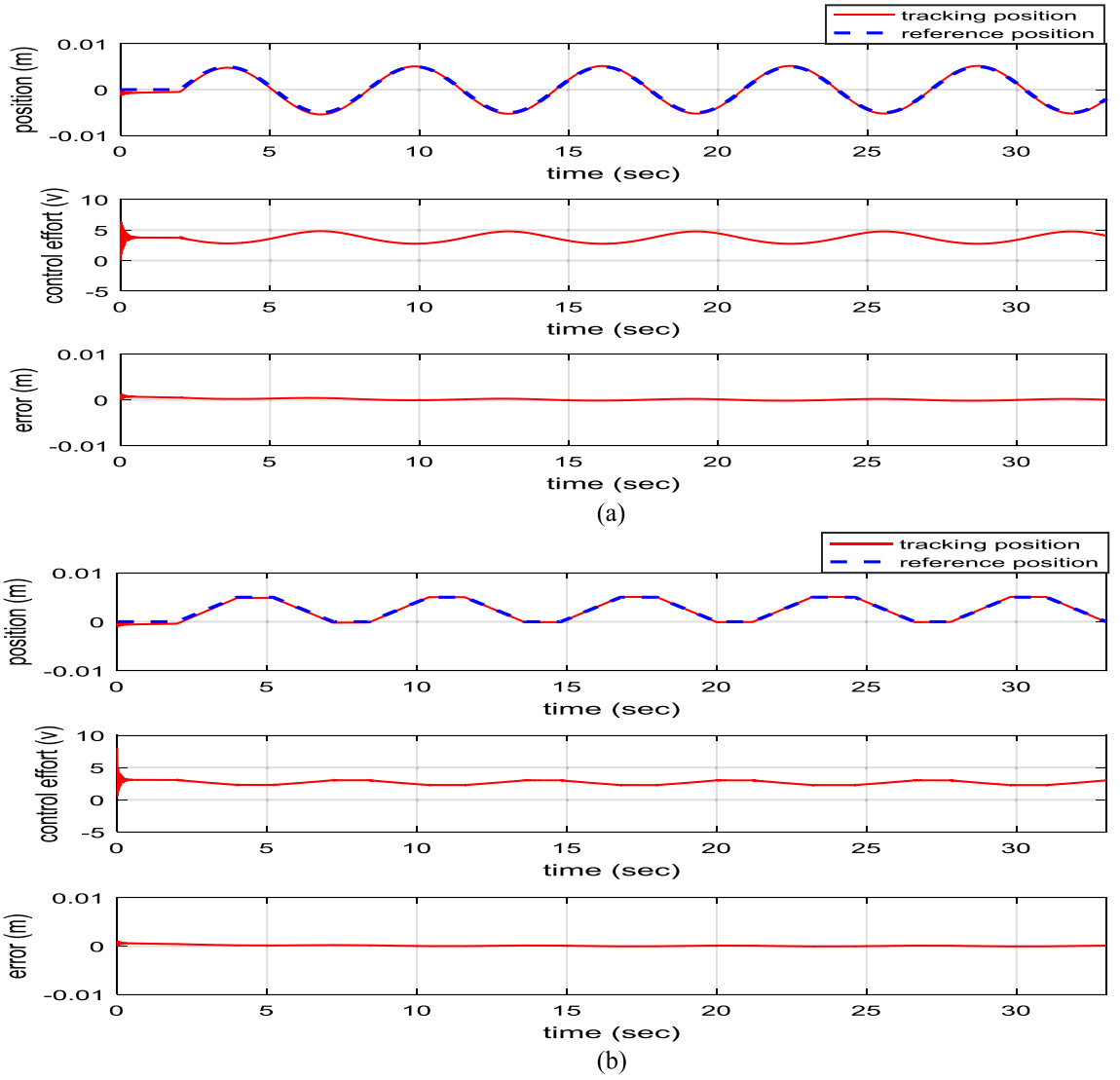

Fig. 6. Simulation results of the PID controller (a) Sinusoidal command and (b) Trapezoid command.

The updation laws of m_{ijk} and v_{ijk} are denoted as:

$$\Delta m_{ijk} = -\gamma_m \frac{\partial E}{\partial m_{ijk}} = \gamma_m \left[-\frac{\partial E}{\partial u_{DTFCMAC}} \frac{\partial u_{DTFCMAC}}{\partial f_{ijk}} \right] \left(\frac{\partial f_{ijk}}{\partial m_{ijk}} \right), = \gamma_m \rho_{ijk} \frac{2(I_i - m_{ijk})}{v_{ijk}^2} \tag{31}$$

$$\Delta v_{ijk} = -\gamma_v \frac{\partial E}{\partial v_{ijk}} = \gamma_v \left[-\frac{\partial E}{\partial u_{DTFCMAC}} \frac{\partial u_{DTFCMAC}}{\partial f_{ijk}} \right] \left(\frac{\partial f_{ijk}}{\partial v_{ijk}} \right), = \gamma_v \rho_{ijk} \frac{2(I_i - m_{ijk})^2}{v_{ijk}^3} \tag{32}$$

where γ_m and γ_v are the learning-rate parameters of the mean and the variance of the Gaussian function, respectively. The mean and variance are updated as follows:

$$m_{ijk}(K+1) = m_{ijk}(K) + \Delta m_{ijk}(K), \tag{33}$$

$$v_{ijk}(K+1) = v_{ijk}(K) + \Delta v_{ijk}(K). \tag{34}$$

If the plant model is obtainable, then the Jacobian of the system $\partial x/\partial u_{DTFCMAC}$ can be calculated. If the plant model is unknown, then $\partial x/\partial u_{DTFCMAC}$ cannot be obtained. Though an intelligent identifier can be applied to identify the system model [21], but heavy computation exertion certainly occurs. In this work, we used a simple approximation of propagation error term as [22]:

$$\lambda_0 \cong \Delta e(K) + e(K). \tag{35}$$

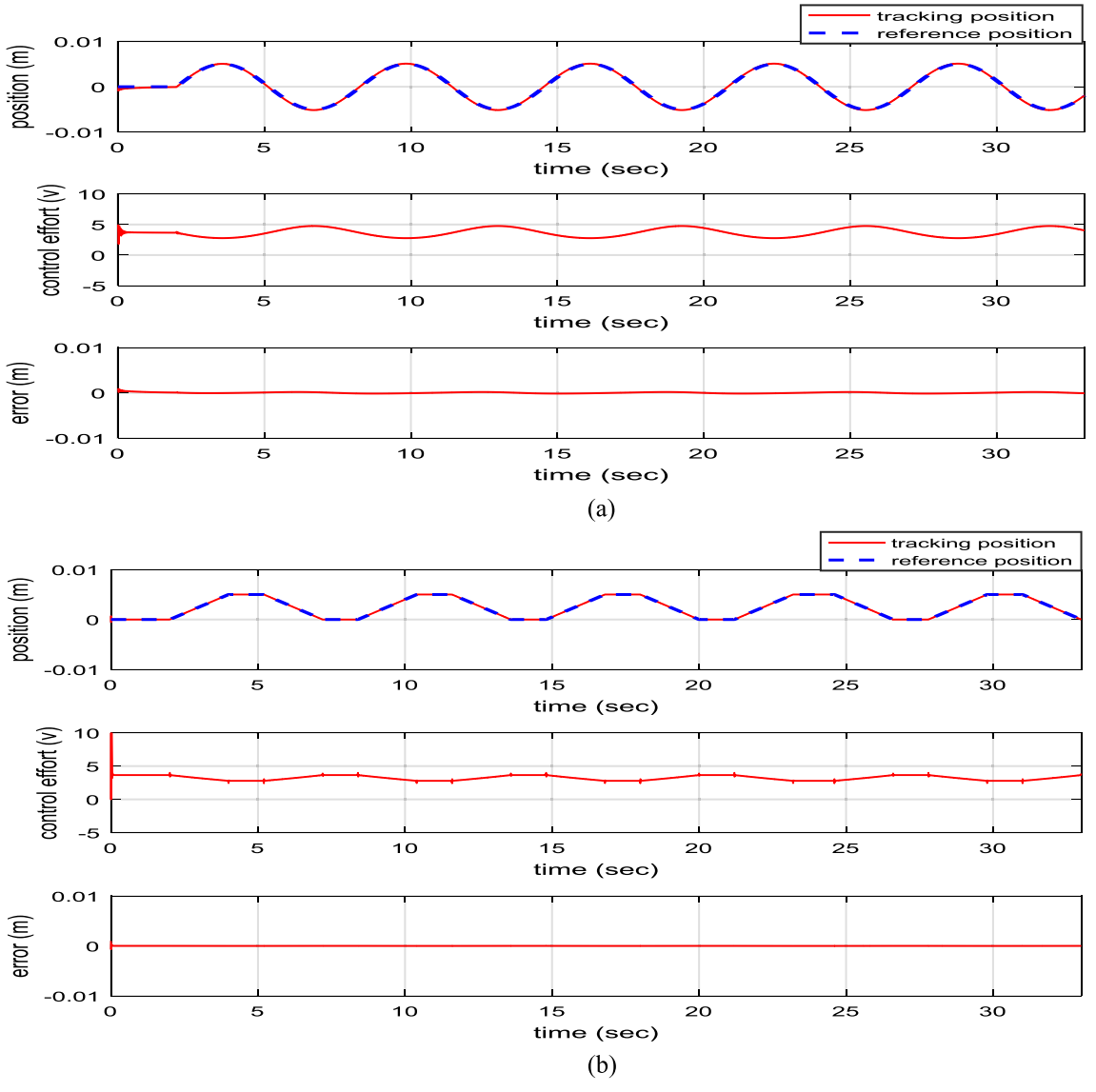

Fig. 7. Simulation results of the DPFCMAC (a) Sinusoidal command and (b) Trapezoid command.

4.3. Stable convergence analyses

The learning laws in (27), (31) and (32) require the appropriate selection of the learning rates γ_w, γ_m, and γ_v, respectively. If the learning rates are set too small, the parameters for the proposed DTFCMAC control system can converge easily. However, it decreases the learning efficiency. On the contrary, if large values are given for the learning rates, the learning speed is fast. Nevertheless, the proposed control system becomes more unstable when the parameters cannot converge quickly. In this study, the convergence analysis is derived to specify the learning rates to assure convergence of the tracking error.

Theorem 1. *Set γ_z is the learning rate for the proposed DTFCMAC control system and define $T_z(K) = \frac{\partial u_{DTFCMAC}}{\partial z}$, for $z = w, m,$ and v. Then the convergence of tracking error is guaranteed if γ_z is selected as*:

$$0 < \gamma_z < \frac{2}{||T_z(K)||^2 \left[\frac{\lambda_0}{e(K)}\right]^2} \tag{36}$$

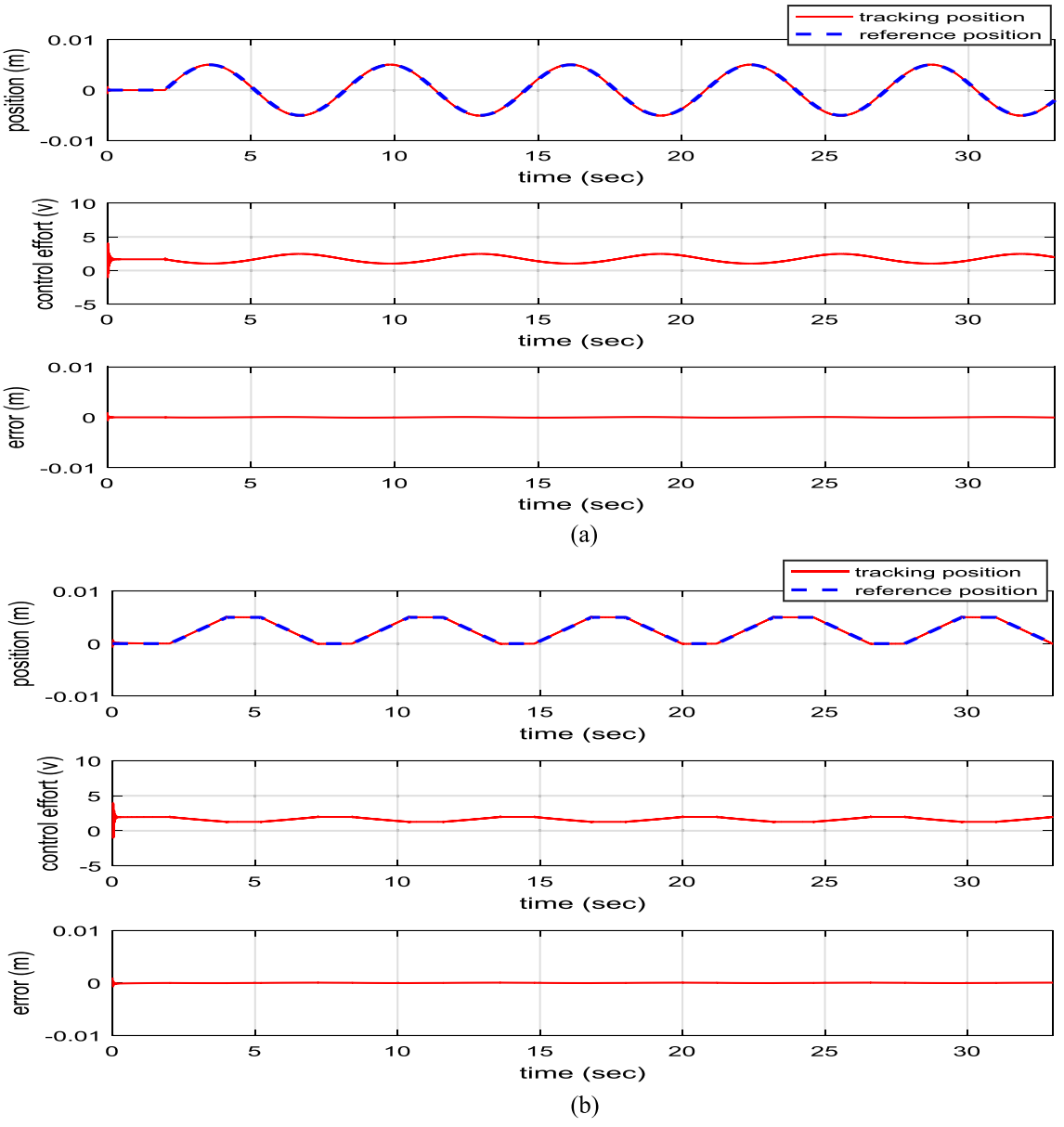

Fig. 8. Simulation results of the proposed DTFCMAC (a) Sinusoidal command and (b) Trapezoid command.

where $\|\cdot\|$ is the Euclidean norm. Moreover, the variable optimal learning rates which achieve the fastest convergence are obtained as:

$$r_z^* = \frac{1}{\|T_z(K)\|^2 \left[\frac{\lambda_0}{e(K)}\right]^2} \tag{37}$$

Proof. Since $T_z(K) = \frac{\partial u_{DTFCMAC}}{\partial z}$ for $z = w,\ m$ and v. $T_w(K)$, $T_m(K)$ and $T_v(K)$ are calculated as follows:

$$T_w(K) = \frac{\partial u_{DTFCMAC}}{\partial w} = \left[\frac{\partial u_{DTFCMAC}}{\partial w_1}, \ldots, \frac{\partial u_{DTFCMAC}}{\partial w_l}, \ldots, \frac{\partial u_{DTFCMAC}}{\partial w_{n_i}}\right]^T, \tag{38}$$

$$T_m(K) = \frac{\partial u_{DTFCMAC}}{\partial m} = \left[\left(\frac{\partial u_{DTFCMAC}}{\partial m_{1jk}}\right)^T, \ldots, \left(\frac{\partial u_{DTFCMAC}}{\partial m_{ijk}}\right)^T, \ldots, \left(\frac{\partial u_{DTFCMAC}}{\partial m_{n_i jk}}\right)^T\right]^T, \tag{39}$$

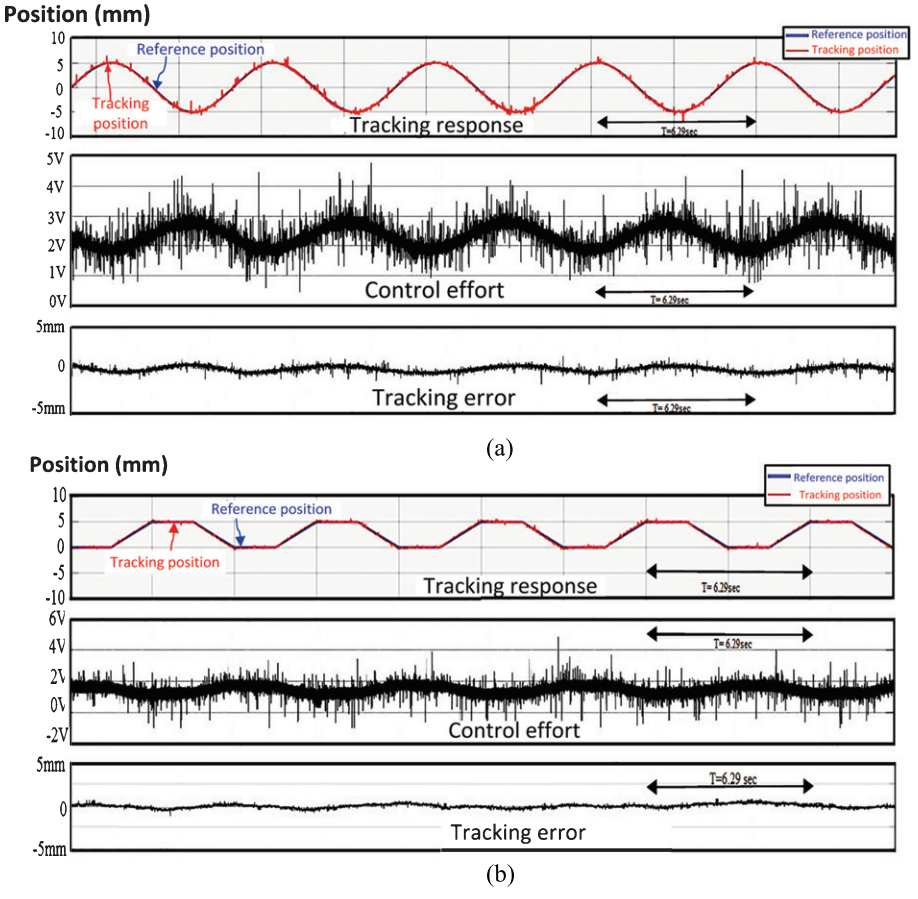

Fig. 9. Experiment results of PID controller (a) Sinusoidal command and (b) Trapezoid command.

where

$$\frac{\partial u_{DTFCMAC}}{\partial m_{n_i jk}} = \left[\frac{\partial u_{DTFCMAC}}{\partial m_{n_i 11}}, \ldots, \frac{\partial u_{DTFCMAC}}{\partial m_{n_i 1 n_k}}, \frac{\partial u_{DTFCMAC}}{\partial m_{n_i 21}}, \ldots \right.$$
$$\left. \ldots, \frac{\partial u_{DTFCMAC}}{\partial m_{n_i 2 n_k}}, \ldots, \frac{\partial u_{DTFCMAC}}{\partial m_{n_i n_j 1}}, \ldots, \frac{\partial u_{DTFCMAC}}{\partial m_{n_i n_j n_k}}\right]^T, \tag{40}$$

$$T_v(K) = \frac{\partial u_{DTFCMAC}}{\partial v} =$$
$$\left[\left(\frac{\partial u_{DTFCMAC}}{\partial v_{1jk}}\right)^T, \ldots, \left(\frac{\partial u_{DTFCMAC}}{\partial v_{ijk}}\right)^T, \ldots, \left(\frac{\partial u_{DTFCMAC}}{\partial v_{n_i jk}}\right)^T\right]^T, \tag{41}$$

where

$$\frac{\partial u_{DTFCMAC}}{\partial v_{n_i jk}} = \left[\frac{\partial u_{DTFMAC}}{\partial v_{n_i 11}}, \ldots, \frac{\partial u_{DTFMAC}}{\partial v_{n_i 1 n_k}}, \frac{\partial u_{DTFMAC}}{\partial v_{n_i 21}}, \ldots \right.$$
$$\left. \ldots, \frac{\partial u_{DTFCMAC}}{\partial v_{n_i 2 n_k}}, \ldots, \frac{\partial u_{DTFCMAC}}{\partial v_{n_i n_j 1}}, \ldots, \frac{\partial u_{DTFCMAC}}{\partial v_{n_i n_j n_k}}\right]^T, \tag{42}$$

and $\frac{\partial u_{DTFCMAC}}{\partial w_1} = r_1$, $\frac{\partial u_{DTFCMAC}}{\partial m_{ijk}} = w_l r_l \frac{2(I_i - m_{ijk})}{v_{ijk}^2}$,

and $\frac{\partial u_{DTFCMAC}}{\partial v} = w_l r_l \frac{2(I_i - m_{ijk})^2}{v_{ijk}^2}$

Define a Lyapunov function as:

$$V(K) = \frac{1}{2} e^2(K). \tag{43}$$

Then the change of the Lyapunov function is obtained as:

$$\Delta V(K) = V(K+1) - V(K) = \frac{1}{2}\left[e^2(K+1) - e^2(K)\right]. \tag{44}$$

The error difference is signified by:

$$e(K+1) = e(K) + \Delta e(K) = e(K) + \left[\frac{\partial e(K)}{\partial z}\right]^T \Delta z. \tag{45}$$

Using (26), yields

$$\frac{\partial e}{\partial z} = \frac{\partial e}{\partial x} \frac{\partial x}{\partial u_{DTFCMAC}} \frac{\partial u_{DTFCMAC}}{\partial z} = -\frac{\lambda_0}{e(K)} T_z(K). \tag{46}$$

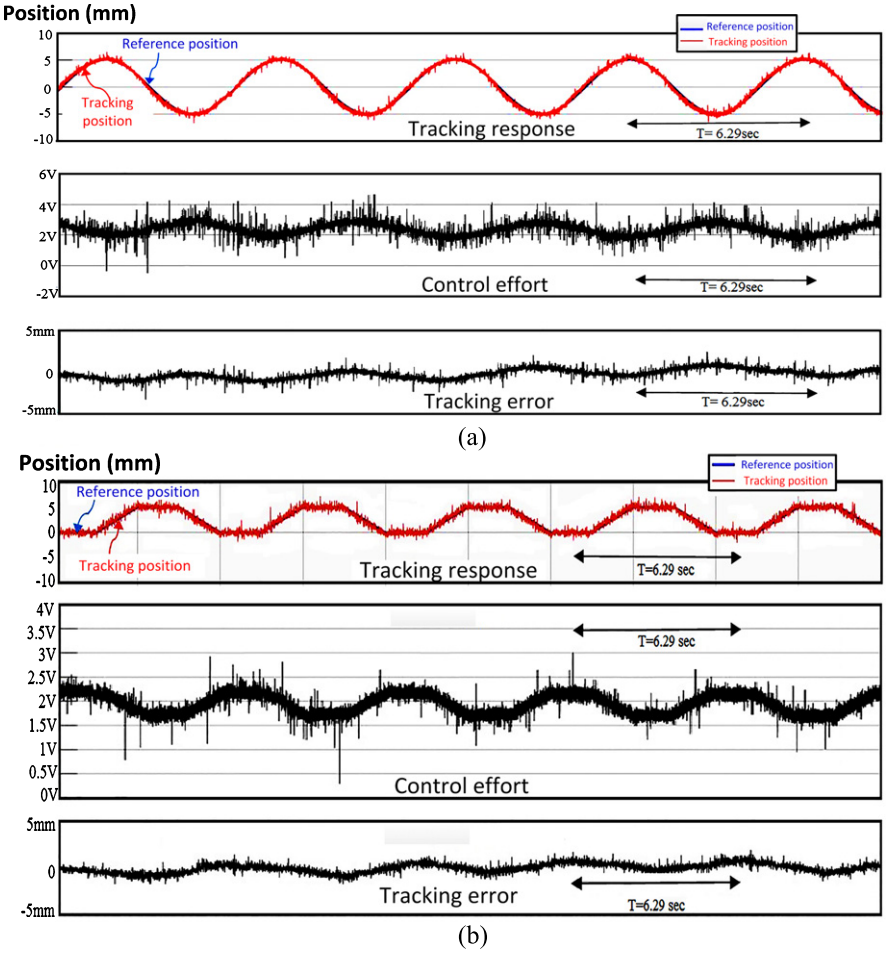

Fig. 10. Experiment results of DPFCMAC (a) Sinusoidal command and (b) Trapezoid command.

Therefore,

$$\begin{aligned} e(K+1) &= e(K) - \left[\frac{\lambda_0}{e(N)} T_z(K)\right]^T \gamma_z \lambda_0 T_z(K). \\ &= e(K)\left[1 - \gamma_z \left(\frac{\lambda_0}{e(N)}\right) T_z^T(K) T_z(K)\right] \end{aligned} \tag{47}$$

From (44) and (47), $\Delta V(K)$ is represented as:

$$\Delta V(K) = \frac{1}{2}\gamma_z \lambda_0^2 \|T_z(K)\|^2 \left[\gamma_z \left(\frac{\lambda_0}{e(K)}\right)^2 \|T_z(K)\|^2 - 2\right]. \tag{48}$$

If γ_z is chosen as in (36), $\Delta V(K)$ in (48) is less than zero, the Lyapunov stability of $V(K) > 0$ and $\Delta V(K) < 0$ is guaranteed. Therefore, the convergence of tracking error $e(K)$ is guaranteed. Moreover, the optimal learning rates which achieve the fastest convergence correspond to:

$$\begin{aligned} 2\gamma_z^* \left[\frac{\lambda_0}{e(K)}\right]^2 \|T_z(K)\|^2 - 2 = 0 \\ \gamma_z^* = \frac{1}{\|T_z(K)\|^2 \left[\frac{\lambda_0}{e(K)}\right]^2}, \end{aligned} \tag{49}$$

which come from the derivative of (48) with respect to γ_z and equals zero. By using this learning-rate value (49), the stable convergence of tracking error $e(K)$ is guaranteed as fast as possible.

5. Simulation and experimental results

5.1. Simulation results

The specifications of the MLS are as follows:

$N = 2850$, $r_1 = 0.012(m)$, $r_2 = 0.038(m)$, $A = 0.005515(m^2)$, $\mu_0 = 4\pi.10^{-7}(Wb/Am)$, $M = 0.0216(kg)$, and $g_m = 9.8(m/s^2)$.

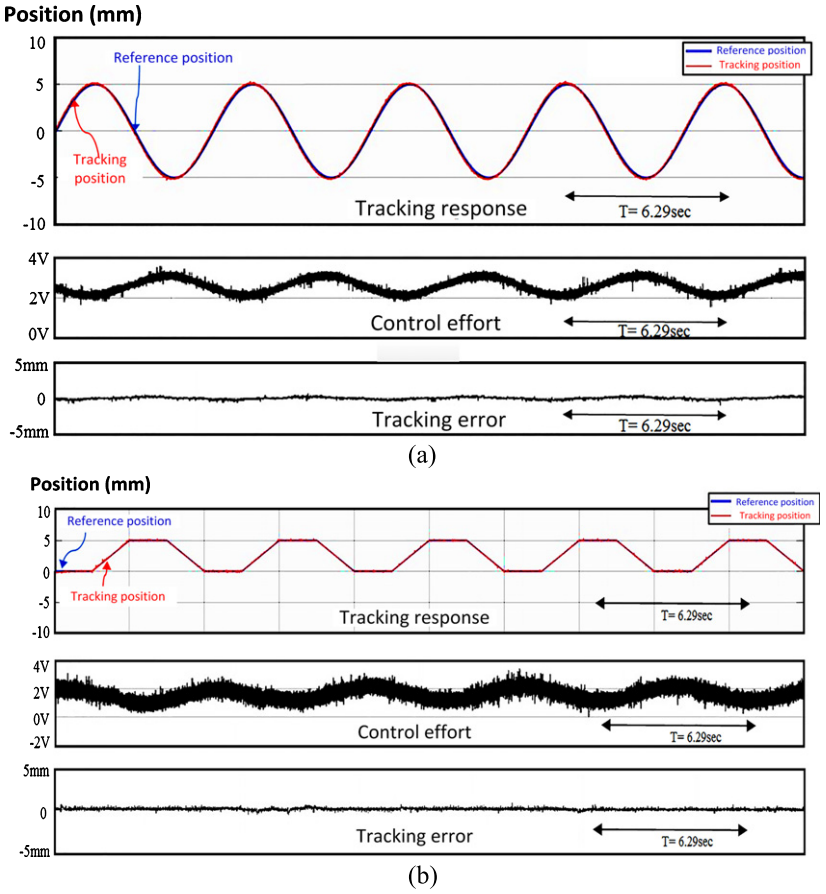

Fig. 11. Experiment results of the proposed DTFCMAC (a) Sinusoidal command and (b) Trapezoid command.

To compare the control efficiency, a conventional PID controller, a DPFCMAC introduced by Lin et al. [7] and the proposed DTFCMAC are used to control the MLS. For the conventional PID controller, the PID gains are $K_P = 5000$, $K_I = 1000$, and $K_D = 3$. For the DPFCMAC, the parameters for calculating the threshold value are chosen as $a = 0.5$ and $\beta = 50$. For both the proposed DTFCMAC and DPFCMAC, the initial learning rates are chosen as $\gamma_w = 10$, $\gamma_m = 1$ and $\gamma_v = 0.1$, and then they are adjusted through (49). The initial position of the metallic sphere is set at –1 mm. The MLS is controlled to follow a sinusoidal command signal of sphere displacement of 1 mm, as shown in Figs. 6(a)–8(a), and a trapezoid command signal of sphere displacement of 1 mm, as shown in Figs. 6(b)–8(b).

The respective threshold values of the DPFCMAC are 0.35 with sinusoidal command and 0.0545 with trapezoid command. In the proposed DTFCMAC, the threshold values are between 0.4864 and 0.5187 with sinusoidal command, and between 0.4962 and 1 with trapezoid command, respectively. The comparisons of these simulation results are listed in Table 2. The RMSE value for the proposed DTFCMAC is clearly smaller than those for the PID controller and the DPFCMAC. In addition, the threshold values for the proposed DTFCMAC control system are also obviously larger than that for the DPFCMAC. The simulation results show that the proposed DTFCMAC achieves the best control performance with the smallest RMSE. In our study, the computation time for each epoch of the PID and DPFCMAC algorithms are 0.000075 second and 0.00014 second, respectively; and the computation time for the proposed DTFCMAC algorithm is 0.00023 second. It shows that the proposed method takes a little longer computation time than the other methods, but it is acceptable. Therefore, the proposed method can achieve better control performance by sacrificing a little longer computation time. The comparison

Table 2
Comparison of simulation results in RMSE

Controller	Computation time (s)	Sinusoidal command (mm)	Threshold values for sinusoidal command	Trapezoid command (mm)	Threshold values for trapezoid command	Degree of algorithm complexity
PID controller	0.000075	0.1179		0.1052		Simple
DPFCMAC	0.00014	0.0431	$d_{th} = 0.35$	0.0321	$d_{th} = 0.0545$	More complex than the PID controller
DTFCMAC	0.00023	0.0238	$d_{th} \in [0.4864, 0.5187]$	0.0187	$d_{th} \in [0.4962, 1]$	A little bit more complex than the DPFCMAC

Table 3
Comparison of experimental results in RMSE

Controller	Sinusoidal command (mm)	Threshold values for sinusoidal command	Trapezoid command (mm)	Threshold values for trapezoid command
PID controller	0.1176		0.0831	
DPFCMAC	0.2835	$d_{th} \in [0, 0.2]$	0.2805	$d_{th} \in [0, 0.2]$
DTFCMAC	0.01287	$d_{th} \in [0.4216, 0.5214]$	0.01174	$d_{th} \in [0.4237, 1]$

Table 4
The number of firing rules and the total number of CMAC parameters

Structure	Number of firing rules for each state variable	The total number of CMAC parameters
FCMAC [26]	36 (fixed)	84 (fixed)
DPFCMAC [7]	Less than 36	Less than 84
DTFCMAC	Less than DPFCMAC's	Less than DPFCMAC's

of degree of algorithm complexity is also shown in Table 2.

5.2. Experimental results

In this study, we use a practical experimental MLS, which is shown in Fig. 1. To compare the control efficiency, a conventional PID controller, a DPFCMAC introduced by Lin et al. [7], and the proposed DTFCMAC are used to control the MLS. For the conventional PID controller, the PID gains are $K_P = 4.4$, $K_I = 2.46$ and $K_D = 0.17$. For the DPFCMAC, the parameters for calculating the threshold value are chosen as $a = 0.2$ and $\beta = 20$. For the proposed DTFCMAC control system, the initial learning rates are chosen as $\gamma_w = 0.01$, $\gamma_m = 0.001$ and $\gamma_v = 0.001$, and then they are adjusted through (49). The experimental results for a sinusoidal command signal with initial control parameters are shown in Figs. 9(a)–11(a), and those for a trapezoid command signal are shown in Figs. 9(b)–11(b). From these experimental results, our findings indicate that the proposed DTFCMAC control system controls the MLS to follow the command signal well.

Moreover, the threshold values for the DPFCMAC are between 0 and 0.2 with both sinusoidal command and trapezoid command. In the proposed DTFCMAC, the threshold values are between 0.4216 and 0.5214 with sinusoidal command, and between 0.4237 and 1 with trapezoid command, respectively. The comparisons of these experimental results with both sinusoidal command signal and trapezoid command signal are summarized in Table 3.

As seen, the RMSE values for the proposed DTFCMAC are obviously smaller than those for the PID controller and the DPFCMAC. In addition, the threshold values for the proposed controller are obviously larger than that for the DPFCMAC, and the number of firing rules and the total number of CMAC parameters are also less than that for the DPFCMAC. In addition, the number of firing rules and the total number of CMAC parameters are shown in Table 4. According to the simulation and the experimental results, highly accurate position tracking responses are achieved by using the proposed DTFCMAC. Compared with the other control systems, chatter and transient tracking error are greatly reduced by the proposed DTFCMAC control system.

6. Conclusion

In this paper, the proposed DTFCMAC control system is successfully applied to control the position of the metallic sphere in a magnetic levitation system. This controller incorporates a TOPSIS with a fuzzy CMAC,

to select suitable firing nodes. The TOPSIS is combined with entropy method to determine the threshold values automatically. This control system is designed without hard computation, and possesses the advantages of the CMAC, such as, reducing the computation burden of parameter learning. According to the adaptive tuning laws, good control performance is achieved by modifying the control parameters to reduce the approximation error. Furthermore, to guarantee the system stability, a Lyapunov function is used to determine the optimal learning-rate parameter. Finally, the proposed intelligent DTFCMAC is implemented with the PCI-1716 multifunction card, to achieve the design goals of small size, low cost, fast execution speed and high flexibility. Both simulation and experimental results of the MLS show that using the proposed controller can achieve highly accurate position tracking responses. The proposed control method is a viable alternative for the control of non-linear systems to achieve better performance. Therefore, the proposed control method is suitable for this problem and is also applicable to other unknown non-linear systems.

Acknowledgments

This paper was supported in part by the National Science Council of the Republic of China under Grand NSC 101-2221-E-155-026-MY3.

References

[1] H.K. Chiang, C.A. Chen and M.Y. Li, Integral variable-structure grey control for magnetic levitation system, *IEE Proceedings – Electric Power Applications* **153** (2006), 809–814.

[2] Z.J. Yang, K. Kunitoshi, S. Kanae and K. Wada, Adaptive robust output-feedback control of a magnetic levitation system by K-filter approach, *IEEE Transactions on Industrial Electronics* **55** (2008), 390–399.

[3] Z.J. Yang, Y. Fukushima, S. Kanae and K. Wada, Robust non-linear output-feedback control of a magnetic levitation system by k-filter approach, *IET Control Theory & Applications* **3** (2009), 852–864.

[4] F.J. Lin, S.Y. Chen and K.K. Shyu, Robust dynamic sliding-mode control using adaptive RENN for magnetic levitation system, *IEEE Transactions on Neural Networks* **20** (2009), 938–951.

[5] C.M. Lin, M.H. Lin and C.W. Chen, SoPC-based adaptive PID control system design for magnetic levitation system, *IEEE Systems Journal* **5** (2011), 278–287.

[6] C.M. Lin, Y.L. Liu and H.Y. Li, SoPC-based function-link cerebellar model articulation control system design for magnetic ball levitation systems, *IEEE Transactions on Industrial Electronics* **61** (2014), 4265–4273.

[7] C.M. Lin and H.Y. Li, Dynamic petri fuzzy cerebellar model articulation controller design for a magnetic levitation system and a two-axis linear piezoelectric ceramic motor drive system, *IEEE Transactions on Control Systems Technology* **23** (2015), 693–699.

[8] H.M. Adel and K. Fatemeh, An extension of fuzzy TOPSIS for a group decision making with an application to tehran stock exchange, *Applied Soft Computing* **52** (2017), 1084–1097.

[9] E. Wang, N. Alp, J. Shi, C. Wang, X. Zhang and H. Chen, Multi-criteria building energy performance benchmarking through variable clustering based compromise TOPSIS with objective entropy weighting, *Energy* **125** (2017), 197–210.

[10] S.H. Mousavi-Nasab and A. Sotoudeh-Anvari, A comprehensive MCDM-based approach using TOPSIS, COPRAS and DEA as an auxiliary tool for material selection problems, *Materials & Design* **121** (2017), 237–253.

[11] C.L. Hwang and K. Yoon, *Multiple Attribute Decision Making: Methods and Applications*, Springer-Verlag, Berlin, 1981.

[12] J.S. Albus, A new approach to manipulator control: The cerebellar model articulation controller (CMAC), *Journal of Dynamic Systems, Measurement, and Control* **97** (1975), 220–227.

[13] C.M. Lin, M.H. Lin and R.G. Yeh, Synchronization of unified chaotic system via adaptive wavelet cerebellar model articulation controller, *Neural Computing and Applications* **23** (2013), 965–973.

[14] C.M. Lin, Y.F. Peng and C.F. Hsu, Robust cerebellar model articulation controller design for unknown nonlinear systems, *IEEE Transactions on Circuits and Systems II: Express Briefs* **51** (2004), 354–358.

[15] J. Zhao and C.M. Lin, Wavelet-TSK-type fuzzy cerebellar model neural network for uncertain nonlinear systems, *IEEE Transactions on Fuzzy Systems* (2018). DOI: 10.1109/TFUZZ.2018.2863650

[16] C.L. Hwang, Y.J. Lai and T.Y. Liu, A new approach for multiple objective decision making, *Computers & Operations Research* **20** (1993), 889–899.

[17] R.V. Rao, *Decision Making in the Manufacturing Environment*, Springer, London, 2007.

[18] C.E. Shannon, A mathematical theory of communication, *Mobile Computing and Communications Review* **5** (2001), 3–55.

[19] M. Zeleny, *Multiple Criteria Decision Making*, McGraw-Hill, New York, 1982.

[20] C.M. Lin, T.T. Huynh and T.L. Le, Adaptive TOPSIS fuzzy CMAC back-stepping control system design for nonlinear systems, *Soft Computing* (2018). DOI: 10.1007/s00500-018-3333-4

[21] T.W. Chow and Y. Fang, A recurrent neural-network-based real-time learning control strategy applying to nonlinear systems with unknown dynamics, *IEEE Transactions on Industrial Electronics* **45** (1998), 151–161.

[22] F.J. Lin, C.H. Lin and C.M. Hong, Robust control of linear synchronous motor servo drive using disturbance observer and recurrent neural network compensator, *IEE Proceedings – Electric Power Applications* **147** (2000), 263–272.