Two-wheeled Robot Control Based on Self-tuning Output Recurrent CMAC

Chih-Hui Chiu, Wen-Ru Tsai, Ming-Hung Chou, and Ya-Fu Peng

Abstract—In this study, a self-tuning output recurrent cerebellar model articulation controller (SORCMAC) is investigated to control a two-wheeled robot. Since it captures the system dynamics, the proposed SORCMAC has superior capability to the conventional cerebellar model articulation controller in efficient learning mechanism and dynamic response. The dynamic gradient descent method is also adopted to online adjust the SORCMAC parameters. Finally, the effectiveness of the proposed control system is verified by the simulations of two-wheeled robot control. Simulation results show that the two-wheeled robot can be controlled stably with uncertainty disturbance by using the proposed SORCMAC.

Index Terms— SORCMAC, two-wheeled robot, gradient descent method.

I. INTRODUCTION

In the past several years, there are several literatures to study the topic "wheeled robot". The main topic of this research is how to keep the robot's balance. Such a system would become a primary tool for studying balance in active balancing system. It is also to be an important precursor to the field of legged machines and locomotion studies for robotics. There are many researches in this filed. In [1], a trajectory tracking algorithm based on a linear state space model was proposed to control mobile robot. Graser et al. [2] used Newtonian approach and linearization method to get a dynamic model to design a controller for a mobile. In [3], a dynamic model was derived with respect to the wheel motor torques as input while taking the nonholonomic no-slip constrains into considerations. Furthermore, two controllers were proposed to stabilize the vehicle's pitch and position.

Until now, at least one commercial product is available such as "Segway".

Modern control system often requires high-speed high-accuracy linear motions. These linear motions are usually realized using the rotary motors with a mechanical transmission, such as reduction gears and lead screw. These

Manuscript received *December 30, 2008.* This work was supported by the National Science Council of Taiwan, R.O.C. under the Grant NSC96-2221-E-155-072

C.H. Chiu is with Department of Electrical Engineering, Yuan-Ze University, Chung-li, Taoyuan 320, Taiwan. (e-mail: chchiu@saturn.yzu.edu.tw).

W.R. Tsai is with Department of Electrical Engineering, Chung Yuan Christian University, Chung-li, Taoyuan 320, Taiwan.

M.H. Chou is with Department of Electrical Engineering, Yuan-Ze University, Chung-li, Taoyuan 320, Taiwan.

Y. F. Peng is with the Electrical Engineering Department, University of Ching-Yun, Chung-Li, Tao-Yuan, 320, Taiwan, R.O.C. (e-mail: yfpeng@cyu.edu.tw).

transmission mechanisms not only significantly reduce the linear motion speed and dynamic response, but also introduce the backlash and large friction. A two-wheeled robot can besimply considered as an inverted pendulum on a two coaxial wheels which have independent drives. This system is composed of a chassis carrying a dc motor coupled to a gearbox for each wheel. The two dc motors are all the rotary motors. It is obviously the transmission loss exists. Therefore, its mathematical model will be complex and the motor parameters are time-varying due to increasing in temperature and changing in motor drive operating condition [4,5]. For control viewpoint, the conventional control technologies always need a good understanding of the plant models. Since a two-wheeled robot dynamic model is difficult to obtain, it is very difficult to control a two-wheeled robot using the conventional control technologies. In recent years, many advanced control techniques have been adopted for a two-wheeled robot control [1-3,6]; however, most of these methods need the plant model and some of these design procedures are overly complex.

Recently, many researches have been done on the applications of neural networks (NNs) for identification and control of dynamic systems [7-12]. Many authors have suggested NNs as powerful building blocks for a wide class of complex nonlinear system control strategies when there exists no complete model information or, even, a controlled plant is considered as a "black box" [7]. According to the structure, the NNs can be mainly classified as feedforward neural networks (FNNs) [8], [9] and recurrent neural networks (RNNs) [10,11]. The most useful property of NNs is their ability to uniformly approximate arbitrary input-output linear or nonlinear mappings on closed subsets. Based on this property, the NN-based controllers have been developed to compensate the effects of nonlinearities and system uncertainties in control system, so that the stability, convergence and robustness of the system can be improved. Moreover, RNN has capabilities superior to FNN, such as the dynamic response and information storing ability [10,11]. Since an RNN has an internal feedback loop, it captures the dynamic response of system with external feedback through delays. Thus, the RNN is a dynamic mapping and demonstrates good control performance in presence of unmodelled dynamics. However, no matter FNNs or RNNs, the learning is slow since all the weights are updated during each learning cycle. Therefore, the effectiveness of NN is limited in problems requiring on-line learning.

The cerebellar model articulation controller (CMAC) has been adopted widely for the closed-loop control of complex dynamical systems owing to its fast learning property, good

generalization capability, and simple computation [13,14]. The CMAC is a non-fully connected perceptron-like associative memory network with overlapping receptive-fields. The application of CMAC is not only limited to control problem but also to model-free function approximation. This network has been already validated that it can approximate a nonlinear function over a domain of interest to any desired accuracy. The advantages of using CMAC over conventional NN in many practical applications have been presented in recent literatures [15-16]. The conventional CMAC uses constant binary or triangular receptive-field basis functions. The disadvantage is that their derivative information is not preserved. For acquiring the derivative information of input and output variables, Chiang and Lin developed a CMAC with differentiable Gaussian receptive-field basis function, and provided the convergence analyses of this network [17]. In [18], an optimal CMAC has been proposed for the robot manipulator control. However, the major drawback of existing CMAC is that they belong to static networks. In other words, the application domain of CMAC will be limited to static mapping due to its feedforward network structure [19,20].

In this study, an self-tuning output recurrent cerebellar model articulation controller for a two-wheeled robot is investigated. Its architecture is a modified model of the conventional CMAC network to attain a small number of receptive-fields for capturing the system dynamics and converting the static CMAC into a dynamic one. Since it captures the dynamic response, the SORCMAC will demonstrate good control performance in presence of unmodelled system. The parameters of SORCMAC are on-line tuned by the derived adaptive laws automatically. Finally, simulation results show the proposed SORCMAC can move the robot forward and back ward stably with the uncertainty disturbance.

II. ARCHITECTURE OF TWO-WHEELED ROBOT

Figure 1 shows the robot with its three degree of freedom. The robot can be able to rotate around the z axis and its vertical axis. The whole system needs six state space variables to fully describe the dynamics of three degree of freedom [2]. The movement of the rotation around the z axis (pitch) can be described by the angle θ_P and the corresponding angular velocity ω_P . The linear movement of the chassis is the position x_{RM} and the speed v_{RM} . Moreover, the robot can rotate around its vertical axis (yaw) with the associated angle δ and angular velocity $\dot{\delta}$. f_{dP} , f_{dRL} and f_{dRR} are the forces applied to the center of gravity of the robot, to the center of the left wheel, to the center of the left wheel, respectively [2]. The dynamic mathematical model of the left-hand wheel can be described as follow: [2]

$$\ddot{x}_{RL}M_{RL} = f_{dRL} - H_L + H_{TL} \tag{1}$$

$$\ddot{y}_{RL}M_{RL} = V_{TL} - M_{RLg} - V_L \tag{2}$$

$$\ddot{\theta}_{RL}J_{RL} = C_L - H_{TL}R \tag{3}$$

where J_{RL} is the moment of inertia of the rotating masses with respect to the z axis; M_{RL} is the mass of the rotating masses connected to the left and right wheels; J_{P0} is moment of inertia of the chassis with respect to the z axis; $J_{p\delta}$ is moment of inertia of the chassis with respect to the z axis; M_p is mass of the chassis;

R: Radius of the wheels; D is lateral distance between the contact patches of the wheels; L is distance between the z axis and CG of the chassis. For the chassis,

$$\ddot{x}_p M_p = f_{dP} - H_R + H_L \tag{4}$$

$$\ddot{y}_p M_p = V_R + V_L - M_{Pg} - F_{C\theta}$$
⁽⁵⁾

$$\ddot{\Theta}_{P}J_{P\theta} = (V_{R} + V_{L})L\sin\theta_{P} - (H_{L} + H_{R})L\cos\theta_{P}$$
(6)
$$-(C_{L} + C_{R})$$
$$\ddot{\delta J}_{P\delta} = (H_{L} - H_{R})\frac{D}{2}$$
(7)

where $H_L, H_R, H_{TL}, H_{TR}, V_L, V_R, V_{TL}$, and V_{TR} represent reaction forces between the different free bodies.



Fig. 1 Two-wheeled robot with its three degree of freedom

Obviously, the linear motion is realized using the rotary motors with reduction gears. These reduction gears not only significantly reduce the linear motion speed and dynamic response, but also introduce the backlash and large friction. It is obviously the transmission loss exists. Therefore, the mathematical model of such a system is complex and the motor parameters are time-varying due to increasing in temperature and changing in motor drive operating condition. For control viewpoint, the conventional control technologies always need a good understanding of the plant models. Since the accurate dynamic model is difficult to obtain, it is very difficult to control a two-wheeled robot using the conventional control technologies.

In this study, a model free self-tuning output recurrent cerebellar model articulation controller for a two-wheeled robot is investigated.

III. SORCMAC CONTROL SYSTEM

An SORCMAC control system is proposed in this section. The configuration of the proposed control system is shown in Fig. 2, where the reference angle signal θ is specified by a reference model following a command input θ^* and θ_n is the angle output of the two-wheeled robot system. Moreover, the reference signal $\Delta \theta$ is specified by a reference model following a position command input x_m and the position output of the robot x. Clearly, $\Delta \theta$ is obtained from the robot's position error. It is a virtual angle which is used to move the robot forward or backward to the target position. In this paper, a virtual angle can be gotten as $\Delta \theta(N) = k_x (x_m(N) - x(N))$ where k_x is a small positive constant. Obviously, $\Delta \theta(N)$ will be zero when x(N) equals to $x_m(N)$. It means that the robot moves to the target position already. The inputs of SORCMAC are the mix angle tracking error of the robot's body $e_m(N) = \theta(N) - \theta_p(N) - \Delta \theta(N)$ and change-of-error $\Delta e_m(N) = e_m(N) - e_m(N-1)$; the output of SORCMAC is the control signal u. Here, the model of the robot system is unknown.



Fig. 2 Block diagram of two-wheeled robot control system

3.1 Description of ORCMAC

An output recurrent cerebellar model articulation controller (ORCMAC) is proposed and shown in Fig. 3*a*. The signal propagation and the basic function in each space are introduced as follows.

1) Input space X: For a given $\mathbf{x} = [x_1, x_2, \dots, x_n]^T \in \Re^n$, each input state variable x_i must be quantized into discrete

regions (called *elements*) according to given control space. The number of elements, n_E , is termed as a resolution.

2) Association memory space A: Several elements can be accumulated as a *block*, the number of blocks, n_B , in ORCMAC is usually greater than two. The A denotes an association memory space with n_A ($n_A = n \times n_B$) components. In this space, each block performs a *receptive-field basis function*, which can be defined as rectangular or triangular or any continuously bounded function (e.g., Gaussian [18], [19] or B-spline [13], [21]). The Gaussian function is adopted here as the receptive-field basis function, which can be represented as

$$\phi_{ik} = \exp\left[\frac{-(x_{rik} - m_{ik})^2}{v_{ik}^2}\right], \text{ for } k = 1, 2, \dots n_B \quad (8)$$

where ϕ_{ik} represents the *k*th block of the *i*th input x_i with the mean m_{ik} and variance v_{ik} . In addition, the input of this block for discrete time *N* can be represented as

$$x_{ri}(N) = x_i(N) + r_i y(N-1)$$
 (9)

where r_i is the recurrent weight of the recurrent unit. It is clear that the input of this block contains the memory terms y(N-1), which store the past information of the network. This is the apparent difference between the proposed ORCMAC and the conventional CMAC. Figure 3*b* depicts the schematic diagram of two-dimensional ORCMAC operations with $n_E = 9$ and $\rho = 4$ (ρ is the number of elements in a complete block).



Fig. 3 *a* Architecture of SORCMAC, *b* A two-dimensional SORCMAC with $\rho = 4$ and $n_E = 9$

3) Receptive-field space T: The number of receptive-field, n_R , is equal to n_B in this study. Each location of A

corresponds to a receptive-field. The *multidimensional receptive-field function* is defined as

$$b_{k}(\mathbf{x},\mathbf{m}_{k},\mathbf{v}_{k},\mathbf{r}_{k}) = \prod_{i=1}^{n} \phi_{ik} = \exp\left[-\left(\sum_{i=1}^{n} \frac{(x_{rik} - m_{ik})^{2}}{v_{ik}^{2}}\right)\right] \quad (10)$$

where b_k is associated with the *k*th receptive-field, $\mathbf{m}_k = [m_{1k}, m_{2k}, \cdots, m_{nk}]^T \in \mathfrak{R}^n$, $\mathbf{v}_k = [v_{1k}, v_{2k}, \cdots, v_{nk}]^T$, $\in \mathfrak{R}^n$ and $\mathbf{r}_k = [r_1, r_2, \cdots, r_n]^T \in \mathfrak{R}^n$. The multidimensional receptive-field function can be expressed in a vector form as

$$\Gamma(\mathbf{x}, \mathbf{m}, \mathbf{v}, \mathbf{r}) = [b_1, \cdots, b_k, \cdots, b_{n_R}]^T$$
(11)
where $\mathbf{m} = [\mathbf{m}_1^T, \cdots, \mathbf{m}_k^T, \cdots, \mathbf{m}_{n_R}^T]^T \in \mathfrak{R}^{nn_R}$, $\mathbf{v} = [\mathbf{v}_1^T, \cdots, \mathbf{v}_{n_R}^T]^T$

$$\mathbf{v}_k^T, \cdots, \mathbf{v}_{n_R}^T$$
]^T $\in \mathfrak{R}^{nn_R}$ and $\mathbf{r} = [\mathbf{r}_1^T, \cdots, \mathbf{r}_k^T, \cdots, \mathbf{r}_{n_R}^T]^T \in \mathfrak{R}^{nn_R}$.

4) Weight memory space W: Each location of T to a particular adjustable value in the weight memory space can be expressed as $\mathbf{w} = [w_1, \dots, w_k, \dots, w_{n_R}]^T \in \Re^{n_R}$ where w_k denotes the connecting weight value of the output associated with the *k*th receptive-field.

5) Output space Y: The output of ORCMAC is the algebraic sum of the activated weights in the weight memory space, and is expressed as

$$y = \mathbf{w}^T \Gamma(\mathbf{x}, \mathbf{m}, \mathbf{v}, \mathbf{r}) = \sum_{k=1}^{n_R} w_k b_k(\mathbf{x}, \mathbf{m}_k, \mathbf{v}_k, \mathbf{r}_k) \quad (12)$$

A two-dimensional ORCMAC is shown in Fig. 8b.

3.2 On-line learning algorithm

Selections of the connective weight **w**, recurrent weight **r**, mean **m** and variance **v** of the receptive-field basis functions will significantly affect the performance of ORCMAC. Inappropriate recurrent weights and receptive-field basis functions will degrade the ORCMAC learning performance. For achieving effective learning, an on-line learning algorithm, which is derived using the supervised gradient descent method, is introduced so that it can real-time adjust the recurrent weights and means and variances of the receptive-field basis functions. Define the cost function E as

$$E = \frac{1}{2} (\theta - \theta_p - \Delta \theta)^2 = \frac{1}{2} e_m^2$$
(13)

The error term to be propagated is given by

$$\rho_p = -\frac{\partial E}{\partial u} = -\frac{\partial E}{\partial e_m} \frac{\partial e_m}{\partial \theta_p} \frac{\partial \theta_p}{\partial u} = e_m \frac{\partial \theta_p}{\partial u}$$
(14)

and the learning algorithm, based on gradient descent method for w_k , can be derived as

$$\Delta w_k = -\beta_w \frac{\partial E}{\partial w_k} = -\beta_w \frac{\partial E}{\partial u} \frac{\partial u}{\partial w_k} = \beta_w \rho_p b_k \qquad (15)$$

where the positive factor β_w is the learning-rate for the output weight w_k . The connective weight can be updated according to the following equation:

$$w_k(N+1) = w_k(N) + \Delta w_k(N)$$
 (16)

Moreover, the mean, variance and recurrent weight of the Gaussian receptive-field basis functions can be also adjusted in the following equation:

$$\Delta m_{ik} = -\beta_m \frac{\partial E}{\partial m_{ik}} = \beta_m \rho_p w_k b_k \frac{2(x_{ri} - m_{ik})}{v_{ik}^2}$$
(17)

$$\Delta v_{ik} = -\beta_v \frac{\partial E}{\partial v_{ik}} = \beta_v \rho_p w_k b_k \frac{2(x_{ri} - m_{ik})^2}{v_{ik}^3}$$
(18)

$$\Delta r_i = -\beta_r \frac{\partial E}{\partial r_i} = \beta_r \rho_p \frac{1}{r_i} y(N-1)$$
(19)

where the positive factors β_m , β_v and β_r are the learningrates for the mean, variance and recurrent weight, respectively. Then the updated laws of mean, variance and recurrent weight are given as follows:

$$m_{ik}(N+1) = m_{ik}(N) + \Delta m_{ik}(N)$$
(20)

$$v_{ik}(N+1) = v_{ik}(N) + \Delta v_{ik}(N)$$
(21)

$$r_i(N+1) = r_i(N) + \Delta r_i(N) \tag{22}$$

If the plant model is obtainable, then the Jacobian of the system $\partial \theta_p / \partial u$ can be calculated. If the plant model is unknown, then $\partial \theta_p / \partial u$ cannot be obtained. Although intelligent identifier [10] can be implemented to identify the system model, heavy computation effort is required. A simple approximation of the propagation error term of the system can be used as follows [22]:

$$\rho_p \cong \Delta e_m + e_m \tag{23}$$

According to Theorem 1, the optimal learning-rate parameters can be gotten automatically and the convergence of tracking error is guaranteed.

IV. SIMULATIONS

In this section, the simulations of the two-wheeled robot using the effective SORCMAC based on the proposed learning laws will be demonstrated. The description of the two-wheeled robot and its model are shown in Section 2. For all the simulations shown in this section, the parameter values of the robot are listed in Table 1 [3].

Table 1. Simulation parameters

M_{b}	35 Kg	I_{xx}	2.1073 kg m^2
M_w	5 Kg	I yy	1.8299 kg m^2
R	0.25 m	I _{zz}	0.6490 kg m^2
C _z	<i>R</i> , 3 <i>R</i> m	I_{wa}	$0.1563 \mathrm{kg} \mathrm{m}^2$
b	0.2 m	I _{wd}	0.0781 kg m^2

The SORCMAC used in this study can be characterized as $\rho = 4$, $n_E = 5$, $n_B = n_R = 2 \times 4$. The inputs of the SORCMAC are the angle error of the robot's body, e_m , and the change rate of the angle error, Δe_m . The receptive-field basis functions are chosen as $\phi_{ik} (x_{ri}) = \exp[-(x_{rik} - m_{ik})^2 / v_{ik}^2]$ for i = 1, 2 and $k = 1, 2, \dots, 8$. Additionally, The initial values of the parameters are chosen as $r_1 = 0.1$, $r_2 = 0.1$, $m_{i1} = -4$, $m_{i2} = -3$, $m_{i3} = -1$, $m_{i4} = -0.5$, $m_{i5} = 0.5$, $m_{i6} = 1, m_{i7} = 3, m_{i8} = 4$ and $v_{ik} = 2$ for all i and k.

In order to exhibit the goodness control performance and the robustness of the two-wheeled robot by the proposed

SORCMAC, several simulation conditions are adopted. Moreover, the external disturbance d is added and is assumed to be 0.0873 during the all simulations. Fig. 4 is the performance of the robot in the normal condition. The target position of the robot is 1.5m. Also, Conditions S1 denotes the extra mass 5Kg is added on the robot. Conditions S2 denotes the existence of the extra mass as S1 with the difference center of mass of the robot body about 3L (half length of the robot). The simulation results are given in Fig. 5 to Fig. 6, respectively. These simulation results show that favorable control performance can be achieved by using the optimal learning-rates SORCMAC. Thus, this SORCMAC control system can satisfy the requirement of the two-wheeled robot control.







Fig. 4 Tracking response, self-tuning cerebellar model articulation controller in normal condition for two-wheeled robot



Fig. 5 Condition S1, self-tuning output recurrent cerebellar model articulation controller with an extra mass (5Kg) for two-wheeled robot





Fig. 6 Condition S2, self-tuning output recurrent cerebellar model articulation controller with an extra mass (5Kg) and the difference center of mass of the robot body (3L) for two-wheeled robot

V.CONCLUSIONS

In this paper, the controller design of the two-wheeled robot is studied. An self-tuning output recurrent cerebellar model articulation controller (SORCMAC) has been proposed for the two-wheeled robot control, in which the dynamic model of the robot is unknown. In the SORCMAC, the parameters of ORCMAC are on-line adjusted. Finally, the simulation results show that the robot can stand upright and move forward and backward stably with uncertainty disturbance.

REFERENCES

- Y. –S. Ha and S. Yuta, "Trajectory tracking control for navigation of the inverse pendulum type self-contained mobile robot," *Robot. Autonom. Syst.*, vol. 17, pp. 65-80, 1996
- [2] Grasser, F., D'Arrigo, A., Colombi, S., and Rufer, A.C., 'JOE: a mobile, inverted pendulum', *IEEE Trans. Industrial Electronics*, 2002, 39, (1), pp. 107-114
- [3] Pathak, K, Franch, J., and Agrawal, S.K.: 'Velocity and position control of a wheeled inverted pendulum by partial feedback linearization', *IEEE Trans. Roboits*, 2005, 21, (3), pp. 505-513
- [4] Sashida, T., and Kenjo, T.: 'An introduction to ultrasonic motors' (Clarendon Press, Oxford, 1993)
- [5] He S., Chen W., Tao X., and Chen Z.: 'Standing wave bi-directional linearly moving ultrasonic motor', *IEEE Trans. Ultrason., Ferroelect.,* and Freq. Contr., 1998, 45, (5), pp. 1133-1139
- [6] Er, M.J., Kee, B.H., and Tan, C.C., "Design and development of an intelligent control for a pole-balancing robot," *Elsevier Science*. *Microprocessors and Microsystems*, 2002, 26, pp. 433-448
- [7] Agarwal, A.: 'A systematic classification of neural-network-based control', *IEEE Contr. Syst. Mag.*, 1997, **17**, pp. 75-93
- [8] Kuschewski, J.G., Hui, S., and Zak, S.H.: 'Application of feed-forward neural networks to dynamical system identification and control', *IEEE*

Trans. Contr. Syst. Technol., 1993, 1, (1), pp. 37-49

- [9] Lin, C.M., and Hsu, C.F.: 'Neural-network-based adaptive control for induction servomotor drive system', *IEEE Trans. Ind. Electron.*, 2002, 49, (1), pp. 115-123
- [10] Chow, T.W.S., and Fang, Y.: 'A recurrent neural-network-based real-time learning control strategy applying to nonlinear systems with unknown dynamics', *IEEE Trans. Ind. Electron.*, 1998, 45, (1), pp. 151-161
- [11] Lin, C.M., and Hsu, C.F.: 'Recurrent neural network adaptive control of wing-rock motion', J. Guid. Control Dyn., 2002, 25, (6), pp. 1163-1165
- [12] Hwang, C.L., Jan, C., and Chen, Y.H.: 'Piezomechanics using intelligent variable-structure control', *IEEE Trans. Ind. Electron.*, 2001, 48, (1), pp. 47-59
- [13] Lane, S.H., Handelman, D.A., and Gelfand, J.J.: 'Theory and development of higher-order CMAC neural networks', *IEEE Contr. Syst. Mag.*, 1992, **12**, (2), pp. 23-30
- [14] Hwang, K.S., and Lin, C.S.: 'Smooth trajectory tracking of three-link robot: a self-organizing CMAC approach', *IEEE Trans. Syst., Man, Cybern. B*, 1998, **28**, (5), pp. 680-692
- [15] Jan, J.C., and Hung, S.L.: 'High-order MS_CMAC neural network', *IEEE Trans. Neural Networks*, 2001, **12**, (3), pp. 598-603
- [16] Chen, J.Y., Tsai, P.S., and Wong, C.C.: 'Adaptive design of a fuzzy cerebellar model arithmetic controller neural network', *IEE Proc.*, *Control Theory Appl.*, 2005, **152**, (2), pp. 133-137
- [17] Chiang, C.T., and Lin, C.S.: 'CMAC with general basis functions', *Neural Networks*, 1996, 9, (7), pp. 1199-1211.
- [18] Kim, Y.H., and Lewis, F.L.: 'Optimal design of CMAC neural-network controller for robot manipulators', *IEEE Trans. Syst.*, *Man, Cybern. C*, 2000, **30**, (1), pp. 22-31
- [19] Lee, C.H. and Teng, C.C.: 'Identification and control of dynamic systems using recurrent fuzzy neural networks', *IEEE Trans. Fuzzy Systems*, 2000, 8, (4), pp. 349-366
- [20] Lin, F.J., Shyu, K.K., and Wai, R.J.: 'Recurrent-fuzzy-neural-network sliding-mode controlled motor-toggle servomechanism', *IEEE/ASME Trans. Mechatron.*, 2001, 6, (4), pp. 453-466
- [21] Jagannathan, S.: 'Discrete-time CMAC NN control of feedback linearizable nonlinear systems under a persistence of excitation', *IEEE Trans. Neural Networks*, 1999, **10**, (1), pp. 128-137
- [22] Lin, F.J., Lin, C.M., and Hong, C.M.: 'Robust control of linear synchronous motor servodrive using disturbance observer and recurrent neural network compensator', *IEE Proc. Electr. Power Appl.*, 2000, 147, (4), pp. 263-272