Journal of Computers Vol. 28, No. 1, 2017, pp. 247-260
doi:10.3966/199115592017022801019

The Tracking Control Design of Adaptive Fuzzy CMAC for an Omni-Directional Mobile Robot

Ter-Feng Wu[1*], Hsu-Chih Huang[1], Pu-Sheng Tsai[2],
Nien-Tsu Hu[3], and Zhi-Qiang Yang[1]

[1] Department of Electrical Engineering, National Ilan University
Yilan 260, Taiwan, ROC
{tfwu, hchuang}@niu.edu.tw; mack14520@hotmail.com

[2] Department of Electronic Engineering, China University of Science and Technology
Taipei 115, Taiwan, ROC
tps@ee2.cust.edu.tw

[3] Chemical Systems Research Division, National Chung-Shan Institute of Science and Technology
Taoyuan 325, Taiwan, ROC
nientsu.hu@gmail.com

Received 18 June 2015; Revised 19 February 2016; Accepted 27 December 2016

Abstract. This work presents an adaptive fuzzy cerebellar model articulation controller (AFCMAC) for solving the tracking control problem for an omni-directional mobile robot. First, fuzzy logic and CMAC are combined, coupled with a triangular basis function that is embedded in the hypercube receptive-field space to yield non-constant differentiable basis functions, which simplify the complex structure and reduce the number of input space dimensions of CMAC. Both adaptive and control laws are developed to tune all of the control parameters online, accommodating uncertainty in tracking control for an omni-directional mobile robot. Hardware agencies and drive circuits are developed. This proposed AFCMAC is implemented in a high-performance field-programmable gate array (FPGA) chip using a hardware/software co-design method and the Qsys design concept with a reusable user IP (Intellectual Property) core library. An omni-directional mobile robot is thus produced. Finally, the trajectory tracking controllability of the robot along both straight and elliptical paths is simulated to evaluate the effectiveness of the proposed AFCMAC and establish the feasibility and performance of uncertain tracking control.

Keywords: adaptive control, cerebellar model articulation controller (CMAC), fuzzy set, omni-directional mobile robot

1 Introduction

In recent years, much research has been performed into control problems associated with nonlinear control systems [1]. Systems theory and the traditional feedback control theory have led to various control schemes that depend on exact mathematical system dynamical models. However, most of these approaches involve many uncertainties when implemented in practice, which may be associated with incompleteness of the model or several unresolvable disturbances. A well-developed control system must exhibit high dynamic performance and robustness even when implemented under complexity and uncertainty [2-4].

Intelligent control theory is utilized in dynamic models to eliminate uncertainties that are caused by disturbances. Systems that are based on the theory include neural networks (NNs) [5-6] fuzzy control of

* Corresponding Author

complex processes with reference to human experience [7-8] and robust control, which is resistant to disturbances [9]. Drawing on Marr's model, Albus established a cerebellar model known as the "cerebella model articulation controller" (CMAC) [10]. A neural network that models the structure and function of the human cerebellum, the CMAC exhibits motor behaviors without using any complex algorithm; it is a table look-up mechanism that comprises a series of function mapping.

The cerebellar model has advantages over conventional NNs; it is composed of simple theoretical structures, requires easy computation, exhibits fast learning and favorable local generalization, approximates multiple non-linear systems, and can be easily implemented in both hardware and software. If the CMAC is merged into a fuzzy set, its complex structure can be simplified and its input space dimensions can be reduced. Adaptive and control laws can be developed to tune all of the control gains online, accommodating uncertainty in an omni-directional robot without any learning phase [11].

Over the last few years, the trajectory tracking control of four-wheeled omni-directional mobile robots [12] has become the focus of much academic interest. However, the control structure of the CMAC cannot be utilized to solve the trajectory tracking control problem of such a robot. To enable highly precise trajectory tracking control of the four-wheeled omni-directional mobile robot, this work demonstrates a modified CMAC system as a response to improve upon the limited mathematical descriptions in the conventional one, and proposes an AFCMAC that is equipped with an embedded switch robust controller to accommodate uncertainties and unpredictable disturbances of the multi-dimensional nonlinear system. Based on the previous work of [13], a Lyapunov stability analysis is carried out so that all of the closed-loop signals are bounded and the tracking errors can converge asymptotically to zero. This proposed methodology is instrumental in the design of an intelligent motion controller of the omni-directional mobile robot.

2 Structure of Controller

2.1 Omni-Directional Mobile Robot

Fig. 1 shows the geometric structure of the omni-directional mobile robot. Owing to the structural symmetry of the vehicle, its center of geometry is also the center of mass. The kinematic model of this robot follows, where θ denotes the orientation of the vehicle, which is positive in the counterclockwise direction [12-13].

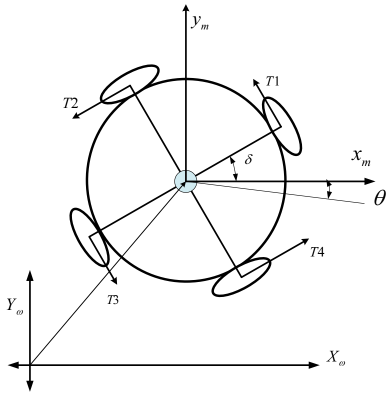

Fig. 1. Structure of omni-directional mobile robot

According to the previous work [13], the kinematic model of an omni-directional mobile robot can be described as follows:

$$\upsilon(t) = \begin{bmatrix} \upsilon_1(t) \\ \upsilon_2(t) \\ \upsilon_3(t) \\ \upsilon_4(t) \end{bmatrix} = \begin{bmatrix} r\omega_1(t) \\ r\omega_2(t) \\ r\omega_3(t) \\ r\omega_4(t) \end{bmatrix} = P(\theta(t)) \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \end{bmatrix} \tag{1}$$

$$P(\theta(t))=\begin{bmatrix}-\sin(\delta+\theta) & \cos(\delta+\theta) & L\\ -\cos(\delta+\theta) & -\sin(\delta+\theta) & L\\ \sin(\delta+\theta) & -\cos(\delta+\theta) & L\\ \cos(\delta+\theta) & \sin(\delta+\theta) & L\end{bmatrix} \quad \textbf{(2)}$$

where $\omega_i(t), i=1,2,3,4$ is the angular velocity of each wheel; r denotes the radius of each wheel, and L represents the distance from the center of the platform to the center of each wheel.. Notably, although the matrix $P(\theta(t))$ is nonsingular for all θ, its inverse matrix can be found using $P^{\#}(\theta(t))P(\theta(t))=I$, and expressed as

$$P^{\#}(\theta(t))=\begin{bmatrix}\frac{-\sin(\delta+\theta)}{2} & \frac{-\cos(\delta+\theta)}{2} & \frac{\sin(\delta+\theta)}{2} & \frac{\cos(\delta+\theta)}{2}\\ \frac{\cos(\delta+\theta)}{2} & \frac{-\sin(\delta+\theta)}{2} & \frac{-\cos(\delta+\theta)}{2} & \frac{\sin(\delta+\theta)}{2}\\ \frac{1}{4L} & \frac{1}{4L} & \frac{1}{4L} & \frac{1}{4L}\end{bmatrix} \quad \textbf{(3)}$$

2.2 CMAC Design

The physical system to be controlled is assumed to have only one control input, and all of the state variables are assumed to be available. Accordingly, a single-output CMAC is designed, with the structure that is shown in Fig. 2, and an output that is given by

$$z_{CMAC}=F(\mathbf{s}) \quad \textbf{(4)}$$

where $F:\mathbf{R}^L\rightarrow\mathbf{R}$ is a nonlinear function of the CMAC input variable $\mathbf{s}=[s_1,\cdots,s_L]^T\in S\subset\mathbf{R}^L$. To mimic the operation of the human cerebellum, the inputs (sensors) are related to the output (response) by an association mechanism with association memory space A. Any element in A comprises 0s and 1s, as determined by the pattern of the inputs. Vector $\mathbf{w}$ is the CMAC weight vector. Mathematically, the relation (1) can be represented by a pair of mappings,

$$G:S\rightarrow A;\ \mathbf{s}\mapsto G(\mathbf{s})=\mathbf{a}(\mathbf{s})\in A \quad \textbf{(5)}$$

$$P:A\rightarrow\mathbf{R};\ \mathbf{a}\mapsto P(\mathbf{a}). \quad \textbf{(6)}$$

In particular, the function P can be chosen to generate output z_{CMAC}, as follows.

$$z_{CMAC}=P(\mathbf{a})=\mathbf{a}^T\mathbf{w} \quad \textbf{(7)}$$

where $\mathbf{a}\in\mathbf{R}^M$ is the relevance vector; $\mathbf{W}\equiv\left[\mathbf{w}_1\ \cdots\ \mathbf{w}_b\right]\in\mathbf{R}^{M\times b}$ denotes the CMAC weight matrix, and $\mathbf{w}_i\in\mathbf{R}^M, i=1,\cdots,b$

Fig. 3 shows a possible partition of the input variables of the CMAC, in which both s_1 and s_2 are divided into seven quantified elements, so that 49 inputs states, with $m,n=-3,-2,-1,0,1,2,3$, are formed yielding 49 inputs states, with $m,n=-3,-2,-1,0,1,2,3$. These elements are further grouped into three blocks, (A, B, C) and (a, b, c), respectively for s_1 and s_2 in the first layer. Their combinations ***Aa***, ***Ab***, ***Ac***, ***Ba***, ***Bb***, ***Bc***, ***Ca***, ***Cb***, ***Cc*** are represent the hypercubes. Shifting the first layer step-by-step on the blocks yields the second and third layers, so that an association vector $\mathbf{a}$ is formed as $\mathbf{a}^T\equiv$[***Aa Ab Ac Ba Bb Bc Ca Cb Cc Dd De Df Ed Ee Ef Fd Fe Ff Gg Gh Gi Hg Hh Hi Ig Ih Ii***].

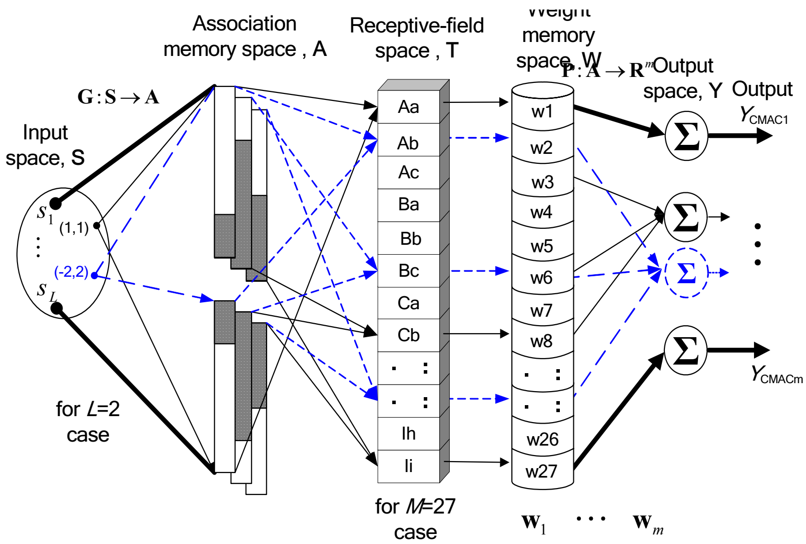

Fig. 2. Architecture of CMAC

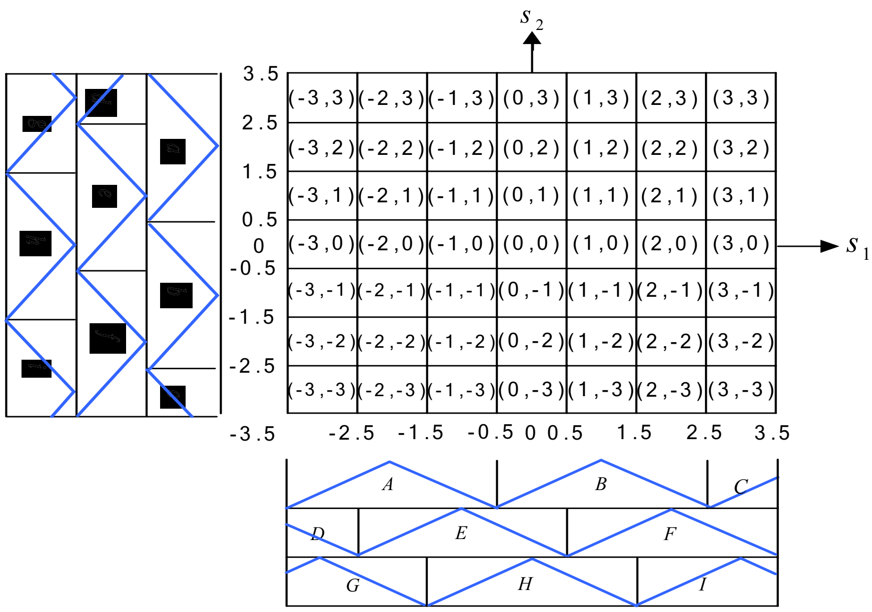

Fig. 3. Partition of two-dimensional CMAC memory unit

For example, if $[s_1, s_2]^T = [0.6, 0.7]^T \in S$, then the corresponding input state is (1, 1) and the associated hypercubes are *Ba*, *Fe*, and *Hh*, yielding $\mathbf{a} = [0001000000000000010000010000]^T$, which in turn yields the following indexing matrix.

$$\mathrm{a} = \begin{bmatrix} \mathrm{a}_1^T \\ \mathrm{a}_2^T \\ \vdots \\ \mathrm{a}_{16}^T \end{bmatrix} = \begin{bmatrix} 0001000000000000010000010000 \\ 0000000010000001000000001000 \\ \vdots \\ 0001000100010000000000000000 \end{bmatrix} \tag{8}$$

Sequential triangular states are embedded into each block variable, as presented in Fig. 4. The relevant equation is expressed as follows.

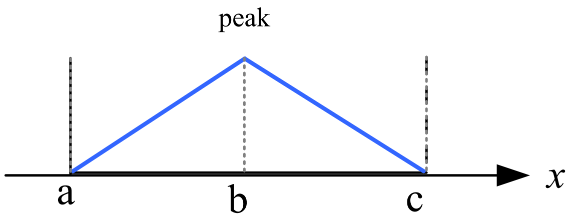

Fig. 4. Triangular states

$$tri(x)=\begin{cases}\dfrac{x-a}{b-a}, & a<x<b\\ \dfrac{c-x}{c-b}, & b<x<c\\ 1, & x=b\end{cases} \tag{9}$$

where a is the lowest valueof any of the block variables; b is the central point of the embedded triangular function blocks; c is the largest value among the block variables, and x represents input variables whose values influence the output of the regions that are covered by functions.

2.3 FCMAC Design

Fig. 5 presents a concept map of fuzzy sets that are integrated with the CMAC that presents a possible partition of the input variables of the CMAC. For a two-input problem, a fuzzy system with N fuzzy rules may be designed, each of which has the form,

$$\mathbf{R}^{(i)}: \textbf{IF } s_1 \text{ is } F_1^i \text{ and } s_2 \text{ is } F_2^i, \textbf{ THEN } z_f^{(i)} \text{ is } \mathbf{a}_i^T\mathbf{w} \tag{10}$$

where $i=1,2,\cdots,N$, and the THEN-part is extracted from the CMAC. Given the membership function of fuzzy set $F_k^i, k=1,2,$ denoted as $\mu_{F_k^i}$, the following defuzzification process is used to compute the output z_{FCMAC}.

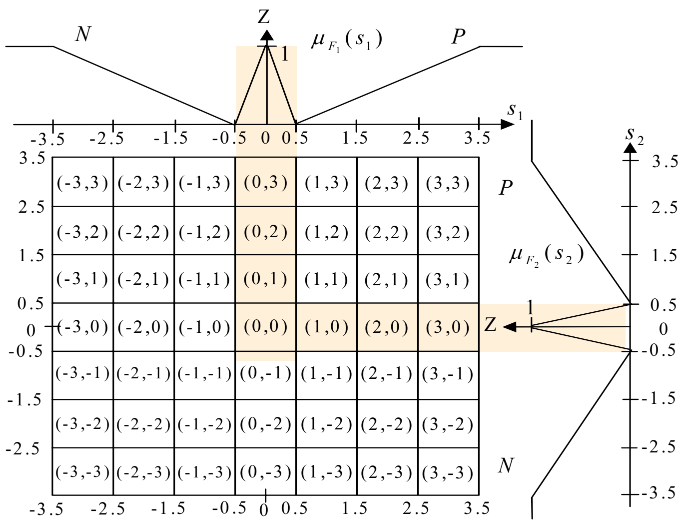

Fig. 5. Conceptual map of (fuzzy sets integrated integration of fuzzy sets with CMAC

$$z_{FCMAC} = \frac{v_1 \mathbf{a}_1^T \mathbf{w} + v_2 \mathbf{a}_2^T \mathbf{w} + \cdots + v_N \mathbf{a}_N^T \mathbf{w}}{v_1 + v_2 + \cdots + v_N} = \frac{\sum_{i=1}^{N} v_i \mathbf{a}_i^T \mathbf{w}}{\sum_{i=1}^{N} v_i} \quad \textbf{(11)}$$

where $v_i = \prod_{k=1}^{2} \mu_{F_k^i}(s_k)$. The preceding equation may be re-written compactly as

$$z_{FCMAC} = \mathbf{h}^T \mathbf{A} \mathbf{w} \quad \textbf{(12)}$$

where

$$\mathbf{h} = \begin{bmatrix} h_1 & h_2 & \cdots & h_N \end{bmatrix}^T,\ h_i = \frac{v_i}{\sum_{i=1}^{N} v_i}\ \ \mathbf{A} = \left[\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_N\right]^T \quad \textbf{(13)}$$

$$\text{and } \mathbf{A} = \begin{bmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \\ \vdots \\ \mathbf{a}_N^T \end{bmatrix} \quad \textbf{(14)}$$

In *Equation* (12), matrix ***A*** (determined by the CMAC) and vector ***h*** (determined by fuzzy rules) are typically fixed, but the weight vector ***w*** is adjustable herein.

In solving the two-input problem, a set of membership functions can be selected, as displayed in Fig. 5, in which *P* (positive), *Z* (zero), and ***N*** (negative) fuzzy sets are imposed on each variable. Therefore, nine fuzzy sets with nine association vectors, $\mathbf{a}_1$, $\mathbf{a}_2$, $\mathbf{a}_3$, $\mathbf{a}_4$, $\mathbf{a}_5$, $\mathbf{a}_6$, $\mathbf{a}_7$, $\mathbf{a}_8$, and $\mathbf{a}_9$, are attached to (*P*, *P*), (*P*, *Z*), (*P*, *N*), (*Z*, *P*), (*Z*, *Z*), (*Z*, *N*), (*N*, *P*), (*N*, *Z*), and (*N*, *N*), respectively. The CMAC includes 49 association vectors in the CMAC, but the FCMAC has only nine. To determine $\mathbf{a}_i$ in the FCMAC, the logical operation 'OR' is performed on all possible association vectors in the same group in the CMAC, yielding association vector, $\mathbf{a}$:

$$\mathbf{a} = \begin{bmatrix} \mathbf{a}_1^T \\ \mathbf{a}_2^T \\ \mathbf{a}_3^T \\ \mathbf{a}_4^T \\ \mathbf{a}_5^T \\ \mathbf{a}_6^T \\ \mathbf{a}_7^T \\ \mathbf{a}_8^T \\ \mathbf{a}_9^T \end{bmatrix} = \begin{bmatrix} 0001001000000001100000110110 \\ 0000100100000000010000010010 \\ 0000110110000000001000011011 \\ 0001000000001100000000110000 \\ 0000100000000010000000010000 \\ 0000110000000001000000011000 \\ 100000000110110000110110000 \\ 0100000000100100000010010000 \\ 0110000000001001000011011000 \end{bmatrix} \quad \textbf{(15)}$$

2.4 Design of AFCMAC System

Based on the work of [14] and the kinematic model of the omni-direction mobile robot (1), the following mathematical model of the multi-dimensional nonlinear system of the robot is developed.

$$\mathbf{y}^{(\mathbf{n})} = \mathbf{f}(\mathbf{x}) + \mathbf{G}(\mathbf{x})\mathbf{u} \quad (16)$$

To merge an output feedback sliding-mode control algorithm into the multi-dimensional nonlinear system, an ideal adaptive law,

$$\dot{\hat{\mathbf{W}}}_i = -\gamma_1 \tilde{y}_i \mathbf{A}^T \mathbf{h}\ ,\ i = 1, \cdots, m \quad \textbf{(17)}$$

$$\dot{\hat{D}} = \gamma_2 \|\tilde{\mathbf{y}}\|_1, \hat{D}(0) > 0 \tag{18}$$

and a control law,

$$\mathbf{u} = \mathbf{G}^{\#}(\mathbf{x})\left[\mathbf{y}_d^{(\mathbf{n})} - \mathbf{C}\tilde{\mathbf{x}}\right] + (\mathbf{h}^T\mathbf{A}\hat{\mathbf{W}})^T - \hat{\mathbf{D}} \cdot \operatorname{sgn}(\boldsymbol{\phi}), \tag{19}$$

are designed.

These two laws are used to construct an AFCMAC system (see Fig. 6) to accommodate the uncertainties and the unpredictable disturbances of a multi-dimensional nonlinear system. Next, a Lyapunov stability analysis is performed to verify the proposition that all signals in a closed-loop system are bounded. The Barbălat lemma is then invoked to determine that the tracking error $\tilde{\mathbf{x}}$ converges asymptotically to zero [1], [13].

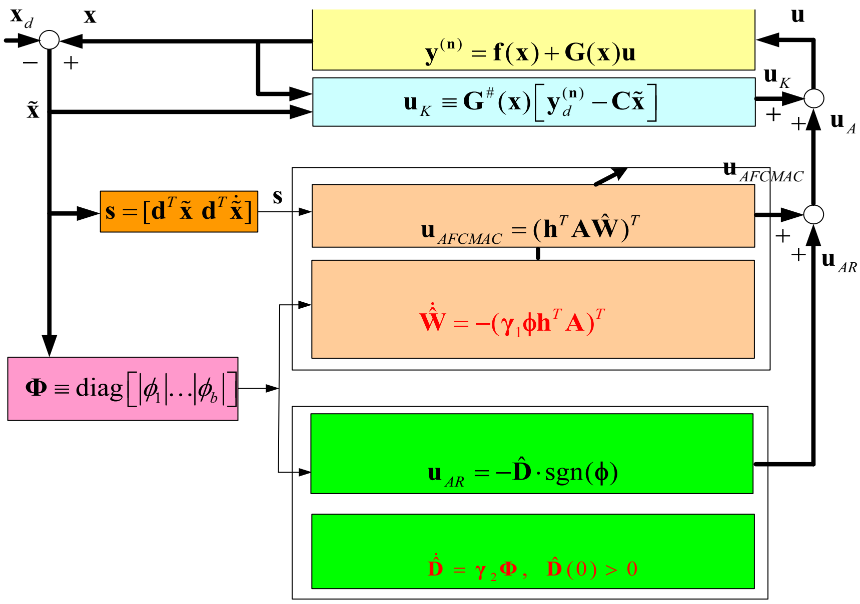

Fig. 6. Architecture of AFCMAC system

3. Systems Simulation

3.1 Simulation Parameters

First, a geometric analysis of a four-wheeled mobile robot along an especially designed straight path (without any angular deflection) is analyzed. (The initial point is $(x_d(0), y_d(0)) = (0,0)$ and the final point is $(x_d(6), y_d(6)) = (6,6)$. An initial point $(x(0), y(0)) = (1,0)$ on the straight path, along with the omni-directional mobile robot travels, is used to (realize OR achieve) trajectory tracking control.

Next, an elliptical path $x^2 + y^2 = 1$, is designed; it is expressed by the parametric equation:

$$\begin{cases} x_d = \cos(t) \\ y_d = \sin(t) \quad , \quad 0 \le t \le 6 \\ \theta_d = 0 \end{cases} \tag{20}$$

The initial point is $(x_d(0), y_d(0)) = (1,0)$ and the final point is $(x_d(6), y_d(6)) = (0.9602,\text{-}0.2794)$. The initial point $(x(0), y(0)) = (1,0)$ on the elliptical path, along which the robot travels, is used to realize trajectory tracking control.

3.2 Simulation Results

Fig. 7 displays the AFCMAC at the initial point $(x_d(0), y_d(0)) = (1,0)$ and the final point $(x_d(6), y_d(6)) = (6,6)$ of the desired input. Fig. 7 reveals that the commands from the starting position $(x(0), y(0)) = (1,0)$ of the omni-directional mobile robot can be captured in an input-output diagram. As evidenced by tracking errors between the initial point and the starting position of the straight-line trajectory, the AFCMAC performs smooth compensation in tracking the target along the line.

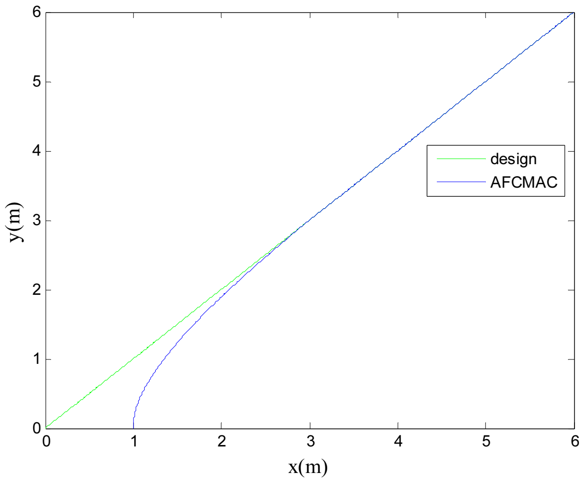

Fig. 7. Tracking response of AFCMAC for straight trajectory

Fig. 8 plots the distribution of tracking errors by the AFCMAC along the straight-line trajectory, and reveals that position errors converge exponentially.

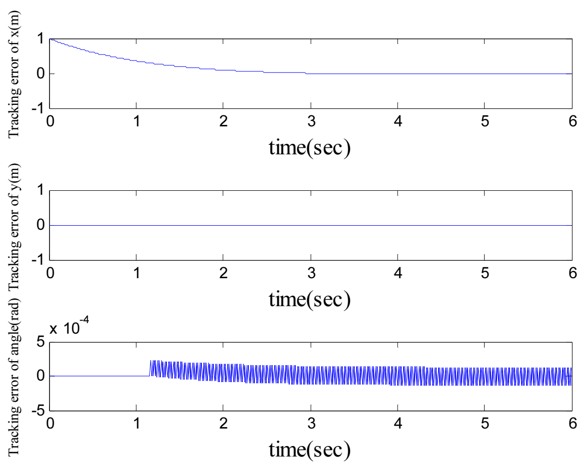

Fig. 8. Tracking error distribution of AFCMAC for straight-line trajectory

Fig. 9 plots the speed response of the straight-line trajectory tracking of the AFCMAC, and reveals smooth, stable speed responses.

Fig. 10 shows an input-output diagram that is derived from the commands associated with the elliptical trajectory tracking of the desired input $x_d = \cos(t), y_d = \sin(t), \theta_d = 0$. The results of a system simulation suggest that the AFCMAC exhibits favorable tracking performance when used along an elliptical path.

Fig. 11 plots the tracking error distribution of the AFCMAC for the elliptical trajectory, and the tracking errors in elliptical trajectory tracking exhibit favorable convergence.

Fig. 12 plots the speed response of the elliptical trajectory tracking of the AFCMAC, verifying the smooth, stable speed of the device.

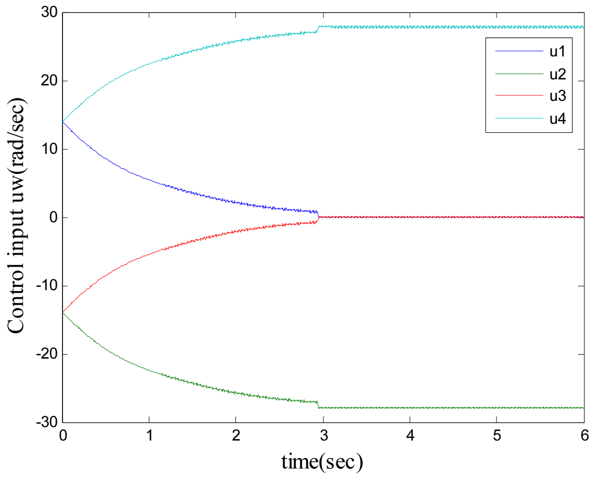

Fig. 9. Speed response of straight-line trajectory tracking by the AFCMAC

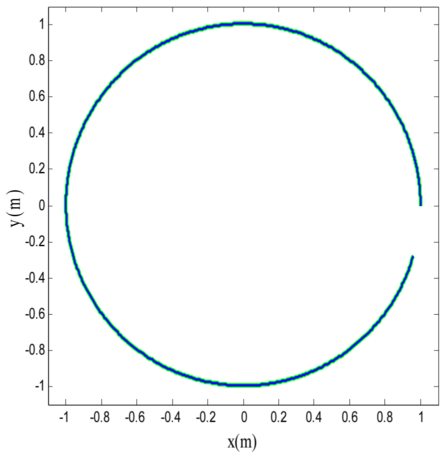

Fig. 10 Tracking response of AFCMAC with elliptical trajectory

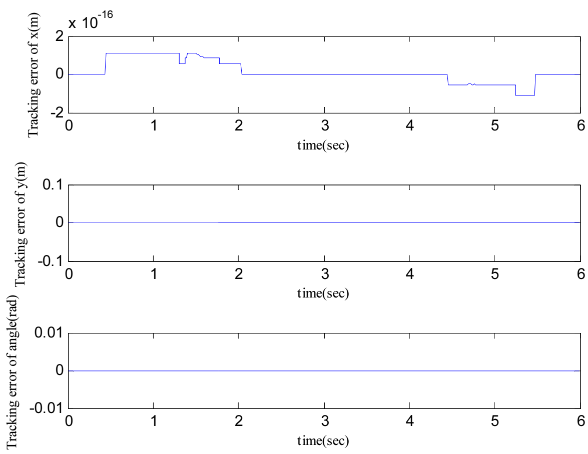

Fig. 11. Tracking error distribution of AFCMAC with elliptical trajectory

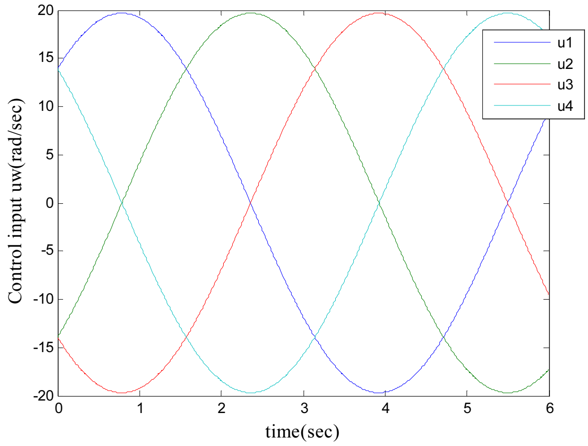

Fig. 12 Speed response of elliptical trajectory tracking by AFCMAC

4 Experiment Results

4.1 Design of Hardware of Omni-Directional Mobile Robot

As presented in Fig. 13, the experimental Qsys-based omnidirectional mobile service robot has the following components; (i) one compact personal computer (PC); (ii) four encoders that are mounted on the driving motors; (iii) one 12V battery; (iv) four DC12V brushless servomotors and their drivers; (v) one four-wheel omnidirectional platform and its controller on an Altera DE2-115 development board, and (vi) eight laser scanners. Four driving omnidirectional wheels are driven by three DC12V brushless servomotors with four mounted encoders. The proposed controller was implemented using C/C++ code and standard programming techniques in the Altera Nios II embedded processor. The FPGA chip integrated the embedded processor, RTOS, and VHDL-based IP circuits to perform the adaptive control of the mobile platform. All of the experiments were performed using system parameters L=20 cm and r=5.08 cm.

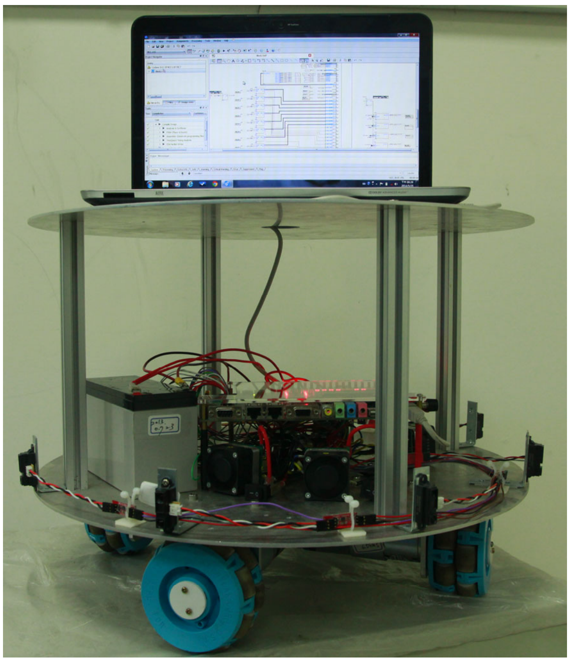

Fig. 13. Comprehensive physical map of experimental omni-directional mobile robot

4.2 Use of AFCMAC in Trajectory Tracking

Tracking of Point-to-Point Positions. First, the initial parameters of both feedback timing and target trajectory are specified, as follows.

1. Initial parameters for feedback timing

$$t_f = 0.1 \text{ (s)}$$

where t_f is the time that is taken by the embedded system to receive feedback from the motor of the omni-directional mobile robot.

2. Initial parameters of target trajectory

Initial coordinate: $\begin{bmatrix} x_0 & y_0 & \theta_0 \end{bmatrix} = \begin{bmatrix} 0 \text{ (cm)} & 0 \text{ (cm)} & 0 \text{ (rad)} \end{bmatrix}$;

objective function: $\begin{bmatrix} x_r & y_r & \theta_r \end{bmatrix} = \left[100 \cdot \cos\left(\frac{n\pi}{4}\right) \text{ (cm)} \quad 100 \cdot \sin\left(\frac{n\pi}{4}\right) \text{ (cm)} \quad 0 \text{ (rad)} \right]$

where $n = 1, \cdots, 8$.

Fig. 14 presents a diagram of the experiment in which an AFCMAC performs point-to-point tracking of a smooth trajectory. Clearly, when the proposed AFCMAC is utilized in an omni-directional mobile robot in point-to-point tracking control, the robot can move from its starting point to the precise location of the target.

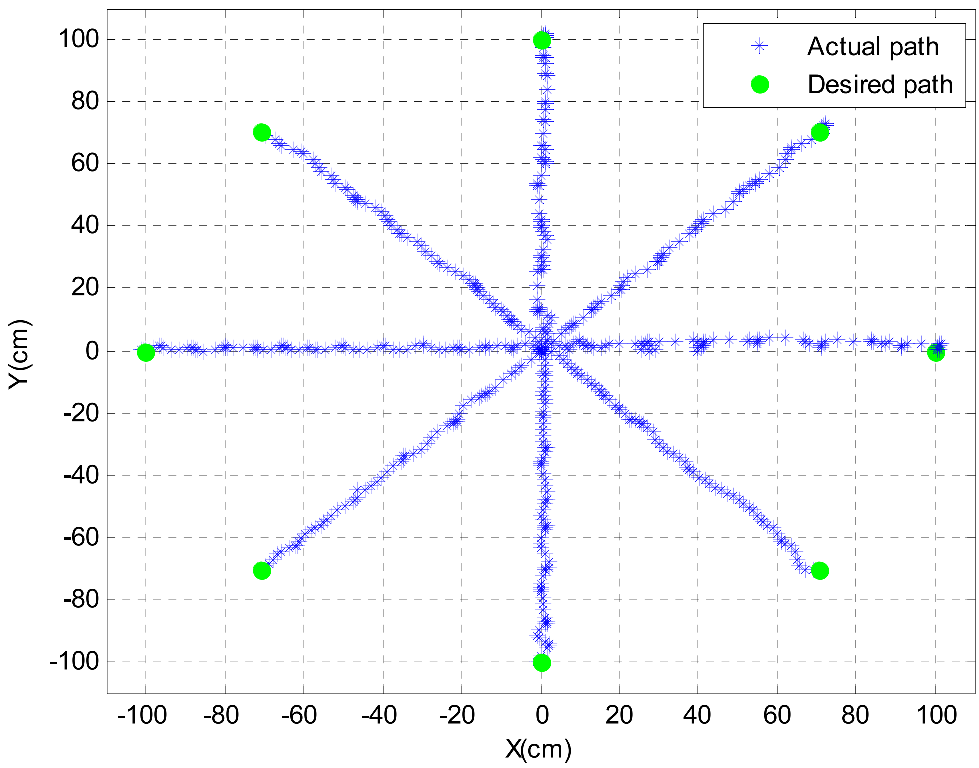

Fig. 14. Experiment on AFCMAC in point-to-point tracking response

Straight-Line Tracking Trajectory. A straight path is designed with the initial point at $(x_d(0), y_d(0)) = (0,0)$ and the final point at $(x_d, y_d) = (100,100)$. (The initial point $(x(0), y(0)) = (50,0)$ on the straight path, along which the omni-directional mobile robot moves, is utilized to achieve trajectory tracking control.

Fig. 15 shows the tracking response of the AFCMAC with a straight-line trajectory. The figure reveals that the proposed AFCMAC controls an omni-directional mobile robot with any initial point and performs accurate straight-line trajectory tracking.

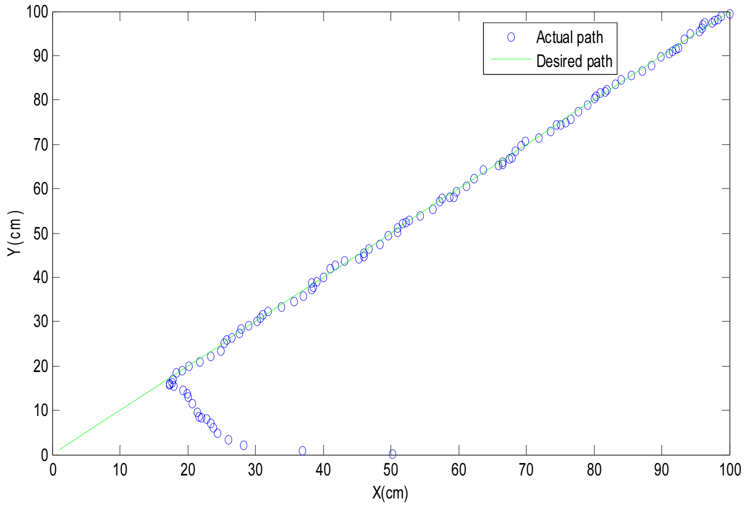

Fig. 15. Experiment on tracking response of AFCMAC for straight-line trajectory

Fig. 16 displays the experiment on the tracking error distribution of the AFCMAC with a straight-line trajectory. The figure reveals that as time passes, tracking errors along the straight-line trajectory converge consistently.

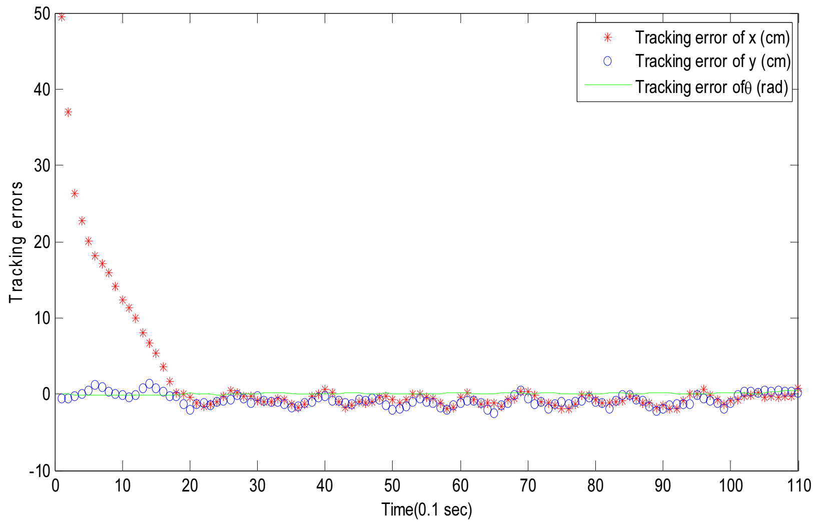

Fig. 16. Experiment on tracking error distribution for straight-line trajectory

Elliptical Trajectory Tracking. Parameters for an elliptical path are as follows.

$$\begin{cases} x_d = 100\cos(\theta) \\ y_d = 150\sin(\theta) \quad ,0 \le \theta \le 2\pi \text{ and } 0 \le \theta \le 2\pi \\ \theta_d = 0 \end{cases} \tag{21}$$

The initial point is $(x_d(0), y_d(0)) = (100,0)$ and the final point is $(x_d(2\pi), y_d(2\pi)) = (100,0)$. (The initial point $(x(0), y(0)) = (0,0)$ on the elliptical path, along which the omni-directional mobile robot moves, is used to achieve trajectory tracking control.

Fig. 17 displays the experiment on the tracking response of the AFCMAC along an elliptical trajectory. The figure verifies that the proposed AFCMAC controls an omni-directional mobile robot with any initial point and can perform accurate elliptical trajectory tracking.

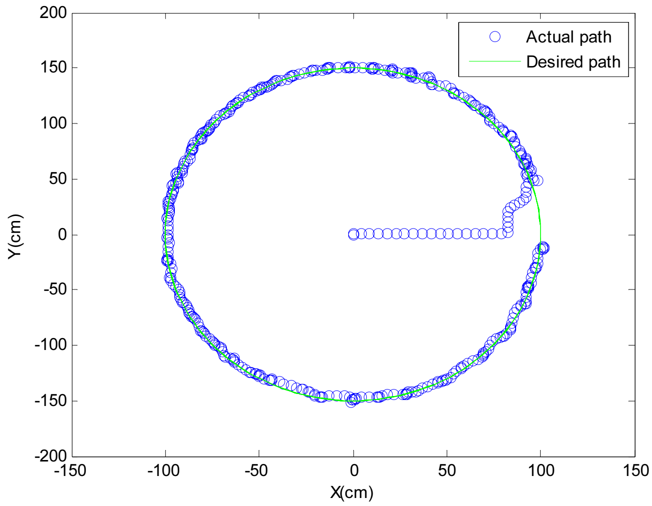

Fig. 17. Experiment on tracking response of AFCMAC with elliptical trajectory

Fig. 18 presents an experiment on the tracking error distribution of AFCMAC with a straight-line trajectory. The figure reveals that as time passes, tracking errors with the elliptical trajectory converge consistently.

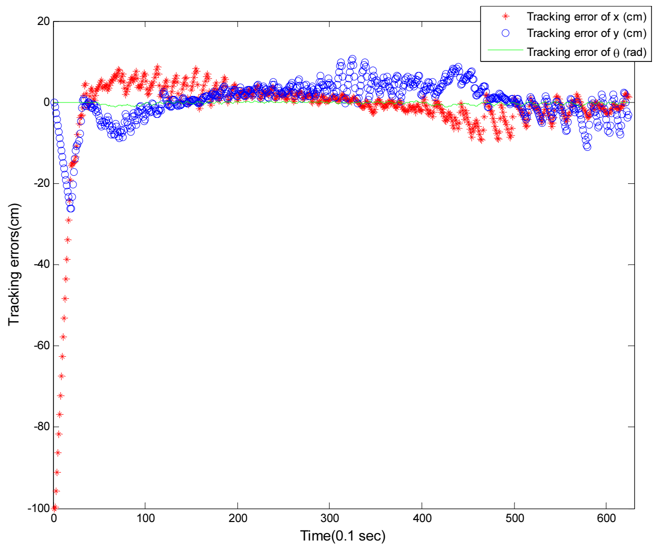

Fig. 18. Experiment on tracking error distribution of AFCMAC with elliptical trajectory

5 Conclusions

This work proposed a modified AFCMAC scheme that is used in the trajectory tracking control of a four-wheeled omni-directional mobile robot. The proposed scheme can be used to solve the tracking control problem of a class of nonlinear uncertain systems. The fuzzy cerebellar model is combined with both adaptive and control laws to enable the controller gains or weights to be adjusted online. The FCMAC is trained to approximate the optimal weights, with the CMAC trained in 49 input states to have a high resolution, so that the error between the input and the desired output is small. The AFCMAC *in all instances* an output feedback sliding-mode control algorithm and ideal adaptive and control laws to reduce its tracking error to zero and to accommodate its approximation error.

Finally, the proposed AFCMAC scheme is implemented on a field-programmable gate array chip, completing the co-design of hardware and software. The Qsys system is used to present results of the simulation of the real-time trajectory tracking control of the omni-directional mobile robot to demonstrate the effectiveness and the high performance of the proposed AFCMAC.

References

[1] J.J.E. Slotine, W. Li, Applied Nonlinear Control, Prentice-Hall, 1991.

[2] G. Maione, High-speed digital realizations of fractional operators in the delta domain, IEEE Transactions on Automatic Control 56(3)(2011) 697-702.

[3] Z.Y. Shen, J. Zhao, J. Xu, X.S. Gu, Nonlinear unknown input observer design by LMI for Lipschitz nonlinear systems, in: Proc. of The 8th World Congress on Intelligent Control and Automation (WCICA), 2010.

[4] F.H. Hsiao, Delay-dependent exponential optimal H-infinty synchronization for Nonidentical chaotic systems via neural-network-based approach, Abstract and Applied Analysis (Article ID 294892), 2013.

[5] K.S. Narendra, K. Parthasarathy, Identification and control of dynamical systems using neural networks, IEEE Transactions *on* Neural Networks 1(1)(1990) 4-27.

[6] J.R. Noriega, H. Wang, A direct sdaptive neural-network control for unknown nonlinear systems and its application, IEEE Transactions *on* Neural Networks 9(1)(1998) 27-34.

[7] T. Takagi, M. Sugeno, Fuzzy identification of systems and its application to modeling and control, IEEE Transactions on Systems, Man, and Cybernetics 15(1)(1985) 116-132.

[8] C.W. Tao, Adaptive fuzzy PIMD controller for systems with uncertain Deadzones, IEEE Transactions *on* Systems, Man, and Cybernetics, Part A 32(5)(2002) 614-620.

[9] W.S. Lin, C.S. Chen, Robust adaptive sliding mode control using fuzzy modelling for a class of uncertain MIMO nonlinear systems, IEE Proceedings - Control Theory and Applications 149(3)(2002) 193-201.

[10] J. S. Albus, A new approach to manipulator control: The cerebellar model articulation controller (CMAC), Journal of Dynamic Systems, Measurement, and Control 97(3)(1975) 220-227.

[11] T.F. Wu, P.S. Tsai, F.R. Chang, L.S. Wang, Adaptive fuzzy CMAC control for a class of nonlinear systems with smooth compensation, IEE Proceeding- Control Theory and Applications 153(6)(2006) 647-657.

[12] H.C. Huang, Intelligent motion control for four-wheeled omnidirectional mobile robots using ant colony optimization, Next Wave in Robotics: Communications in Computer and Information Science 212(2011) 94-106.

[13] T.F. Wu, H.C. Huang, S.S.-D. Xu, N.T. Hu, Adaptive fuzzy CMAC controller design for tracking of four-wheeled omnidirectional mobile robot, Innovation, in: T.H. Meen, S.D. Prior, & A.D.K.T. Lam (Eds.), Communication and Engineering-Meen, Taylor & Francis Group, London, 2013, pp. 341-346.