FPGA-based real-time implementation of an adaptive RCMAC control system

Chih-Min Lin[1] Chun-Fei Hsu[2] Chao-Ming Chung[1]
[1] Department of Electrical Engineering
Yuan Ze University,
Chung-Li, Tao-Yuan, 320,
Taiwan, Republic of China
cml@saturn.yzu.edu.tw; s948507@mail.yzu.edu.tw
[2] Department of Electrical Engineering
Chung Hua University,
Hsinchu, 300, Taiwan, Republic of China
fei@chu.edu.tw

Abstract: - The main advantage of the recurrent cerebellar model articulation controller (RCMAC) is its rapid learning rate compared to other neural networks. This paper proposes an adaptive RCMAC control system for a brushless DC (BLDC) motor. The proposed control scheme is composed of an RCMAC controller and a robust compensator. The RCMAC controller is used to mimic an ideal controller, and the robust controller is designed to compensate for the approximation error between the ideal controller and the RCMAC controller. The Lyapunov stability theory is utilized to derive the parameter tuning algorithm, so that the stability of the closed-loop system can be achieved. As compared with standard adaptive controller, the proposed control scheme does not require persistent excitation condition. Then, the developed adaptive RCMAC control system is implemented on a field programmable gate array (FPGA) chip for controlling a brushless DC motor. Experimental results reveal that the proposed adaptive RCMAC control system can achieve favorable tracking performance. Since the developed adaptive RCMAC control system uses a robust compensator to compensate for the approximation error, there is no chattering phenomenon in the control effort. Thus, the proposed control system is more suitable for real-time practical control applications.

Key-Words: - BLDC; FPGA implementation; RCMAC; adaptive control; Lyapunov function; neural control.

1 Introduction

With the learning ability of the neural network (NN), NNs have widely been recognized as powerful tool in industrial control, commercial prediction, image processing applications and etc [1]. Recently, the adaptive NN control technique has represented an alternative design method for various unknown nonlinear control systems [2-6]. Based on the approximation ability property of NN, the adaptive NN controllers have been developed to compensate for the effects of nonlinearities and system uncertainties [2-4]. To obtain the fast learning property and good generalization capability, the cerebellar model articulation controller (CMAC) has been proposed [7]. CMAC is classified as a non-fully connected perceptron-like associative memory network with overlapping receptive-fields. CMAC has been already validated that it can approximate a nonlinear function over a domain of interest to any desired accuracy. There has been considerable interest in exploring the applications of CMAC to deal with the nonlinearity and uncertainty of control systems [8-11]. It has been shown that the adaptive CMAC control systems can achieve better control performance than adaptive NN control systems by appropriately designing the CMAC controller [8].

According to the structure, the NNs can be mainly classified as feedforward neural networks (FNNs) and recurrent neural networks (RNNs) [1]. As known, FNN is a static mapping. Without the aid of tapped delays, FNNs are unable to represent a dynamic mapping. As far as RNNs are concerned, their ability to deal with time varying input or output through their own natural temporal operation is of particular interest. Thus, RNN is a dynamic mapping and demonstrates good control performance in the presence of unmodelled dynamics [1, 12-14]. In this study, a recurrent CMAC (RCMAC), which involves dynamic elements in the form of feedback connections that are used as internal memories, is developed to design the controller. RCMAC has advantages over CMAC in its dynamic response and its information storing ability.

Because of the brushless direct current (BLDC) motor has the advantages of simple structure, high torque, reliability, small inertia, low noise and longer lifetime (no brush erosion), it has gradually been used in the motion control applications such as electric vehicles, aeronautics, robotics, dynamic actuation and micro electric motor cars [15]. However, BLDC motor is a nonlinear system whose internal parameter values will change slightly with different input commands and environments. Using the BLDC motor in high-performance drivers require advance and robust control methods. Recently, several investigations have been carried out by applying various control algorithms to control the BLDC motors [16-18]. Liu et al. proposed a PI controller based on the completely understanding of the model and through some time-consuming design procedures to achieve instantaneous torque control with reduced torque ripple [16]. However, their performances generally depend on the working point, thus the control parameters which want to ensure proper behavior in all operating conditions are difficult to design. Rubaai et al. proposed an adaptive fuzzy-neural-network controller to achieve a satisfactory tracking performance [17]. In order to ensure the system stability, a compensation controller will be designed to dispel the approximation error. However, the most frequently used of compensation controller is like a sliding-mode control, which requires the bound of the approximation error. To solve above problems, a robust adaptive fuzzy-neural-network controller had been developed [18]. Though the robust tracking performance can be achieved, the used neural network is a feedforward neural network. It may be selected with a sufficiently large number of hidden neurons, in which the computation loading is heavy. Moreover, the control effort may lead to a large control signal as the specified robustness is increased.

To tackle the nonlinear problem of BLDC motor, this paper proposes an adaptive RCMAC control with a new training algorithm. The proposed adaptive RCMAC control is composed of an RCMAC controller and a robust compensator. The RCMAC controller is used to mimic an ideal controller and it presents the main controller. The robust compensator is designed to achieve L_2 tracking performance with desired attenuation level. Since RCMAC has an internal feedback loop, it captures the dynamic response of system with external feedback through delays. Moreover, all the parameters of the adaptive RCMAC control are tuned in the Lyapunov sense, thus the stability can be guaranteed. Finally, the proposed adaptive RCMAC control is implemented based on a field programmable gate array (FPGA) chip for possible low-cost and high-performance industrial applications. The experimental results demonstrate that the proposed adaptive RCMAC control scheme can achieve favorable control performance.

2 RCMAC approximator

The network structure of RCMAC is shown in Fig. 1. The architecture of RCMAC includes input space, association memory space, receptive-field space, weight memory space, output space and recurrent weight. The output of RCMAC can be expressed as

$$y = \sum_{q=1}^{n_R} w_q \, b_q \tag{1}$$

where b_q is the receptive-field basis function of the qth receptive-field, w_q denotes the connecting weight value of the qth receptive-field, n_R is the number of receptive-field, and the receptive-field basis function is defined as

$$b_q(\mathbf{z}) = \prod_{j=1}^{n} \varphi_{jk}(z_{rj}) \text{ , for } q = 1, 2, \ldots, n_R \tag{2}$$

where $\mathbf{z} = [z_1, z_2, \cdots, z_n]^T \in R^n$ is the input vector and the Gaussian function is adopted as the receptive-field basis function which can be represented as

$$\varphi_{jk}(z_j) = \exp\left[\frac{-(z_{rj} - m_{jk})^2}{\sigma_{jk}^2}\right] \text{, for } k = 1, 2, \cdots n_B \tag{3}$$

where $\varphi_{jk}(z_j)$ presents the kth block of the jth input z_j with the mean m_{jk} and variance σ_{jk} and n_B is the number of blocks. In addition, the input of this block can be represented as

$$z_{rj} = z_j(N) + r_{jk}\varphi_{jk}(N-1) \tag{4}$$

where r_{jk} is the recurrent weight of the recurrent unit. It is clear that the input of this block contains a memory term $\varphi_{jk}(N-1)$, which stores the past information of the network. This is the apparent difference between the proposed RCMAC and the conventional CMAC. For ease of notation, the output of RCMAC can be expressed in a vector notation as

$$y = \mathbf{w}^T \boldsymbol{\Theta}(\mathbf{z}, \mathbf{m}, \boldsymbol{\sigma}, \mathbf{r}) \tag{5}$$

where

$$\mathbf{w} = [w_1, ..., w_q, ..., w_{n_R}]^T \tag{6}$$

$$\boldsymbol{\Theta} = [b_1, ..., b_q, ..., b_{n_R}]^T \tag{7}$$

$$\mathbf{m} = \left[m_{11}, ..., m_{n1}, m_{12}, ..., m_{n2}, ..., m_{1n_B}, ..., m_{nn_B}\right]^T \tag{8}$$

$$\boldsymbol{\sigma} = \left[\sigma_{11}, ..., \sigma_{n1}, \sigma_{12}, ..., \sigma_{n2}, ..., \sigma_{1n_B}, ..., \sigma_{nn_B}\right]^T \tag{9}$$

$$\mathbf{r} = \left[r_{11}, ..., r_{n1}, r_{12}, ..., r_{n2}, ..., r_{1n_B}, ..., r_{nn_B}\right]^T . \tag{10}$$

This implies that there exists an RCMAC of (5) such that it can uniformly approximate a nonlinear even time-varying function Ω. By the universal approximation theorem, there exist an ideal weight vectors such that [19, 20]

$$\Omega = y^* + \Delta = \mathbf{w}^{*T}\mathbf{\Theta}(\mathbf{z},\mathbf{m}^*,\boldsymbol{\sigma}^*,\mathbf{r}^*) + \Delta$$
$$= \mathbf{w}^{*T}\mathbf{\Theta}^* + \Delta \quad (11)$$

where Δ denotes the approximation error, $\mathbf{w}^*$ and $\mathbf{\Theta}^*$ are the optimal parameter vectors of $\mathbf{w}$ and $\mathbf{\Theta}$, respectively, and $\mathbf{m}^*$, $\boldsymbol{\sigma}^*$ and $\mathbf{r}^*$ are the optimal parameter vectors of $\mathbf{m}$, $\boldsymbol{\sigma}$ and $\mathbf{r}$, respectively. In fact, the optimal parameter vectors that are needed to best approximate a given nonlinear function Ω cannot be determined. Thus, an estimation function is defined as

$$\hat{y} = \hat{\mathbf{w}}^T\mathbf{\Theta}(\mathbf{z},\hat{\mathbf{m}},\hat{\boldsymbol{\sigma}},\hat{\mathbf{r}}) = \hat{\mathbf{w}}^T\hat{\mathbf{\Theta}} \quad (12)$$

where $\hat{\mathbf{w}}$ and $\hat{\mathbf{\Theta}}$ are the estimated parameter vectors of $\mathbf{w}$ and $\mathbf{\Theta}$, respectively, and $\hat{\mathbf{m}}$, $\hat{\boldsymbol{\sigma}}$ and $\hat{\mathbf{r}}$ are the estimated parameter vectors of $\mathbf{m}$, $\boldsymbol{\sigma}$ and $\mathbf{r}$, respectively. Define the estimation error as

$$\tilde{y} = \Omega - \hat{y} = \mathbf{w}^{*T}\mathbf{\Theta}^* - \hat{\mathbf{w}}^T\hat{\mathbf{\Theta}} + \Delta$$
$$= \tilde{\mathbf{w}}^T\tilde{\mathbf{\Theta}} + \hat{\mathbf{w}}^T\tilde{\mathbf{\Theta}} + \tilde{\mathbf{w}}^T\hat{\mathbf{\Theta}} + \Delta \quad (13)$$

where $\tilde{\mathbf{w}} = \mathbf{w}^* - \hat{\mathbf{w}}$ and $\tilde{\mathbf{\Theta}} = \mathbf{\Theta}^* - \hat{\mathbf{\Theta}}$. In the following, some tuning laws will be derived to on-line tune the parameters of RCMAC to achieve favorable estimation of a nonlinear function. To achieve this goal, the Taylor expansion linearization technique is employed to transform the nonlinear function into a partially linear form, i.e. [19]

$$\tilde{\mathbf{\Theta}} = \mathbf{A}^T\tilde{\mathbf{m}} + \mathbf{B}^T\tilde{\boldsymbol{\sigma}} + \mathbf{C}^T\tilde{\mathbf{r}} + \mathbf{H} \quad (14)$$

where $\tilde{\mathbf{m}} = \mathbf{m}^* - \hat{\mathbf{m}}$; $\tilde{\boldsymbol{\sigma}} = \boldsymbol{\sigma}^* - \hat{\boldsymbol{\sigma}}$; $\tilde{\mathbf{r}} = \mathbf{r}^* - \hat{\mathbf{r}}$; $\mathbf{H}$ is a vector of higher-order terms; $\mathbf{A} = \left[\frac{\partial\Theta_1}{\partial\mathbf{m}}\frac{\partial\Theta_2}{\partial\mathbf{m}}\cdots\frac{\partial\Theta_{n_R}}{\partial\mathbf{m}}\right]\Big|_{\mathbf{m}=\hat{\mathbf{m}}}$; $\mathbf{B} = \left[\frac{\partial\Theta_1}{\partial\boldsymbol{\sigma}}\frac{\partial\Theta_2}{\partial\boldsymbol{\sigma}}\cdots\frac{\partial\Theta_{n_R}}{\partial\boldsymbol{\sigma}}\right]\Big|_{\boldsymbol{\sigma}=\hat{\boldsymbol{\sigma}}}$; $\mathbf{C} = \left[\frac{\partial\Theta_1}{\partial\mathbf{r}}\frac{\partial\Theta_2}{\partial\mathbf{r}}\cdots\frac{\partial\Theta_{n_R}}{\partial\mathbf{r}}\right]\Big|_{\mathbf{r}=\hat{\mathbf{r}}}$. Substituting of (14) into (13) yields

$$\tilde{y} = \tilde{\mathbf{w}}^T\tilde{\mathbf{\Theta}} + \hat{\mathbf{w}}^T(\mathbf{A}^T\tilde{\mathbf{m}} + \mathbf{B}^T\tilde{\boldsymbol{\sigma}} + \mathbf{C}^T\tilde{\mathbf{r}} + \mathbf{h}) + \tilde{\mathbf{w}}^T\hat{\mathbf{\Theta}} + \Delta$$
$$= \tilde{\mathbf{w}}^T\hat{\mathbf{\Theta}} + \tilde{\mathbf{m}}^T\mathbf{A}\hat{\mathbf{w}} + \tilde{\boldsymbol{\sigma}}^T\mathbf{B}\hat{\mathbf{w}} + \tilde{\mathbf{r}}^T\mathbf{C}\hat{\mathbf{w}} + \varepsilon \quad (15)$$

where $\hat{\mathbf{w}}^T\mathbf{A}^T\tilde{\mathbf{m}} = \tilde{\mathbf{m}}^T\mathbf{A}\hat{\mathbf{w}}$, $\hat{\mathbf{w}}^T\mathbf{B}^T\tilde{\boldsymbol{\sigma}} = \tilde{\boldsymbol{\sigma}}^T\mathbf{B}\hat{\mathbf{w}}$ and $\hat{\mathbf{w}}^T\mathbf{C}^T\tilde{\mathbf{r}} = \tilde{\mathbf{r}}^T\mathbf{C}\hat{\mathbf{w}}$ are used since they are scales, and the uncertain term $\varepsilon = \hat{\mathbf{w}}^T\mathbf{h} + \tilde{\mathbf{w}}^T\tilde{\mathbf{\Theta}} + \Delta$ denotes the approximation error.

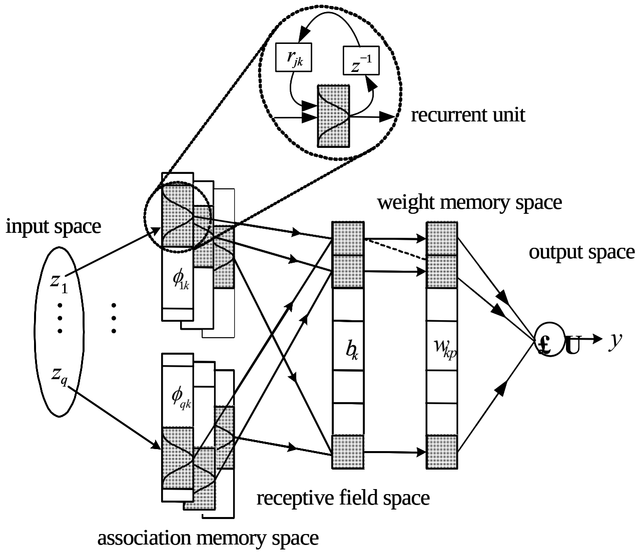

Fig. 1 The architecture of RCMAC.

3 Adaptive RCMAC control system design

3.1 Modelling of a brushless DC motor

The system equations of BLDC motor driver in a *d-q* model can be expressed as [15]

$$i_{qs} = -\frac{R_s}{L_q}i_{qs} - \frac{L_d}{L_q}\omega_r i_{ds} + \frac{1}{L_q}v_{qs} - \frac{\lambda_m}{L_q}\omega_r \quad (16)$$

$$i_{ds} = -\frac{R_s}{L_d}i_{ds} + \frac{L_q}{L_d}\omega_r i_{qs} + \frac{1}{L_d}v_{ds} \quad (17)$$

$$L_q = L_{is} + L_{mq} \quad (18)$$

$$L_d = L_{is} + L_{md} \quad (19)$$

$$T_e = \frac{3}{2}\frac{N}{2}[\lambda_m i_{qs} + (L_d - L_q)i_{qs}i_{ds}] \quad (20)$$

where i_{ds} and i_{qs} represent the d and q axes stator currents, respectively, R_s is the stator resistance, L_d and L_q are the d and q axes stator inductances, respectively, V_{ds} and V_{qs} are the d and q axes stator voltage, respectively, L_{is} is the stator leakage inductance, L_{md} and L_{mq} are the d and q axes magnetizing inductances, respectively, ω_r is the electrical rotor angular velocity, λ_m is the flux linkage of the permanent magnet and N is the number of poles. Considering the mechanical load, the dynamic equation of BLDC motor driver can be written as

$$J\frac{2}{N}\dot{\omega}_r + B\frac{2}{N}\omega_r = T_e - T_L \quad (21)$$

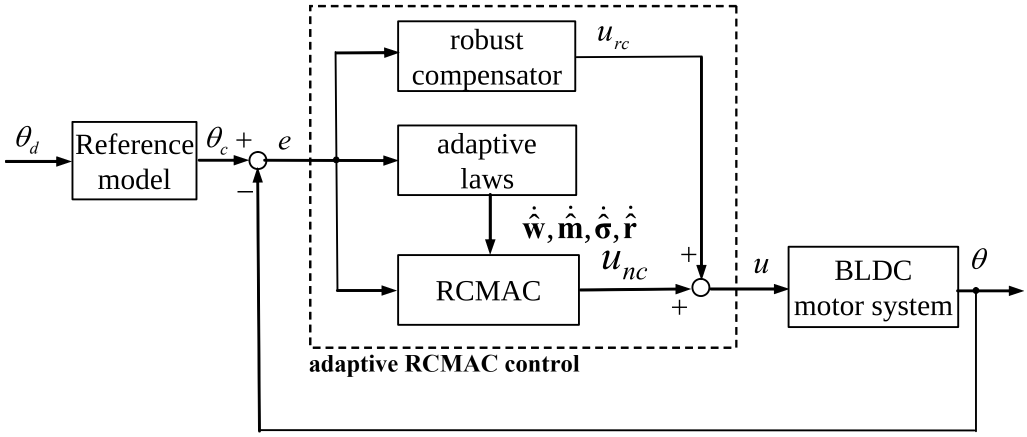

Fig. 2 Block diagram of adaptive RCMAC control system.

where J is the inertia of the system, B is the damping coefficient, and T_L is the load disturbance. By using the field-oriented control, it can make i_{ds} become zero. Therefore, the equation of BLDC motor driver can be rewritten as [15]

$$i_{qs} = -\frac{R_s}{L_q} i_{qs} + \frac{1}{L_q} V_{qs} - \frac{\lambda_m}{L_q} \omega_r \tag{22}$$

$$\dot{\omega}_r = \frac{3}{2}\frac{1}{J}(\frac{N}{2})^2 \lambda_m i_{qs} - \frac{B}{J}\omega_r - \frac{N}{2J} T_L \tag{23}$$

and the torque equation is expressed as

$$T_e = \frac{3}{2}\frac{N}{2}\lambda_m i_{qs} = k_t i_{qs} \tag{24}$$

where $k_t = \frac{3}{2}\frac{N}{2}\lambda_m$ is the constant gain. From (23) and (24), it can obtain

$$\ddot{\theta} = f\dot{\theta} + gu + h \tag{25}$$

where $\theta = \int \omega_r dt$ is the position of the rotor, $f = -\frac{B}{J}$, $g = \frac{N}{2}\frac{k_t}{J}$, $h = -\frac{N}{2J} T_L$, and $u = i_{qs}$ is the control effort.

3.2 Control system design

The control objective of the BLDC motor driver is to find a control law so that the rotor position θ can track the position command θ_c closely. Thus, define the tracking error as

$$e = \theta_c - \theta . \tag{26}$$

Assume that the parameters of the controlled system in (24) are well known, there exits an ideal controller [21]

$$u^* = g^{-1}(-f\dot{\theta} - h + \ddot{\theta}_c + k_1\dot{e} + k_2 e) \tag{27}$$

where k_1 and k_2 are positive constants. Applying the ideal controller (27) into (25) results in the following error dynamics

$$\ddot{e} + k_1\dot{e} + k_2 e = 0 . \tag{28}$$

If k_1 and k_2 are chosen such that all roots of the polynomial $h(s) \triangleq s^2 + k_1 s + k_2$ lie strictly in the open left half of the complex plane, then it implies that $\lim_{t\to\infty} e = 0$ for any starting initial conditions. However, since the system dynamics f and g, and the disturbance h may be unknown or perturbed in practical applications, the ideal controller u^* in (27) cannot be precisely obtained.

To tackle this problem, the adaptive RCMAC control system for BLDC motor driver is proposed and shown in Fig. 2, where the control law is designed as

$$u = u_{nc} + u_{rc} . \tag{29}$$

The RCMAC controller u_{nc} is used to mimic the ideal controller u^* and the robust compensator u_{rc} is used to compensate for the difference between the RCMAC controller and the ideal controller. Substituting (29) into (25) and using (27), the error dynamic equation can be obtained as

$$\dot{\mathbf{e}} = \mathbf{A_m e} + \mathbf{b}(u^* - u_{nc} - u_{rc}) \tag{30}$$

where $\mathbf{A_m} = \begin{bmatrix} 0 & 1 \\ -k_2 & -k_1 \end{bmatrix}$ and $\mathbf{b} = [0,1]^T$. Using the approximation ability of RCMAC in (15), (30) can be rewritten as

$$\dot{\mathbf{e}} = \mathbf{A_m e} + \mathbf{b}(\tilde{\mathbf{w}}^T \hat{\boldsymbol{\Theta}} + \tilde{\mathbf{m}}^T \mathbf{A}\hat{\mathbf{w}} + \tilde{\boldsymbol{\sigma}}^T \mathbf{B}\hat{\mathbf{w}} + \tilde{\mathbf{r}}^T \mathbf{C}\hat{\mathbf{w}} + \varepsilon - u_{rc}) \tag{31}$$

In case of the existence of ε, consider a specified L_2 tracking performance [22], [23]

$$\int_0^T \mathbf{e}^T \mathbf{Q e}\, dt \le \mathbf{e}^T(0)\mathbf{P e}(0) + \frac{1}{\eta_w}\tilde{\mathbf{w}}^T(0)\tilde{\mathbf{w}}(0) + \frac{1}{\eta_m}\tilde{\mathbf{m}}^T(0)\tilde{\mathbf{m}}(0) + \frac{1}{\eta_\sigma}\tilde{\boldsymbol{\sigma}}^T(0)\tilde{\boldsymbol{\sigma}}(0) + \frac{1}{\eta_r}\tilde{\mathbf{r}}^T(0)\tilde{\mathbf{r}}(0) + \rho^2\int_0^T \varepsilon^2 dt \tag{32}$$

where $\mathbf{Q}$ and $\mathbf{P}$ are symmetric positive definite matrices, η_w, η_m, η_σ and η_r are positive constants, and ρ is a prescribed attenuation level. If the system starts with initial conditions $\mathbf{e}(0)=0$, $\tilde{\mathbf{w}}(0)=0$, $\tilde{\mathbf{m}}(0)=0$, $\tilde{\boldsymbol{\sigma}}(0)=0$ and $\tilde{\mathbf{r}}(0)=0$, the L_2 tracking performance in (32) can be rewritten as

$$\sup_{\varepsilon\in L_2[0,T]}\frac{\|\mathbf{e}\|}{\|\varepsilon\|}\le\rho \tag{33}$$

where $\|\mathbf{e}\|^2=\int_0^T\mathbf{e}^T\mathbf{Q}\mathbf{e}dt$ and $\|\varepsilon\|^2=\int_0^T\varepsilon^2dt$. The attenuation constant ρ can be specified by the designer to achieve the desired attenuation ratio between $\|\mathbf{e}\|$ and $\|\varepsilon\|$. If $\rho=\infty$, this is the case of minimum error tracking control without disturbance attenuation [22].To guarantee the stability of the adaptive RCMAC control system, the Lyapunov function is defined as

$$V=\frac{1}{2}\mathbf{e}^T\mathbf{P}\mathbf{e}+\frac{1}{2\eta_w}\tilde{\mathbf{w}}^T\tilde{\mathbf{w}}+\frac{1}{2\eta_m}\tilde{\mathbf{m}}^T\tilde{\mathbf{m}}+\frac{1}{2\eta_\sigma}\tilde{\boldsymbol{\sigma}}^T\tilde{\boldsymbol{\sigma}}+\frac{1}{2\eta_r}\tilde{\mathbf{r}}^T\tilde{\mathbf{r}} \tag{34}$$

and define the Riccati-like equation as

$$\boldsymbol{\Lambda}^T\mathbf{P}+\mathbf{P}^T\boldsymbol{\Lambda}+\mathbf{Q}-\frac{2}{\delta}\mathbf{P}\mathbf{b}\mathbf{b}^T\mathbf{P}+\frac{1}{\rho^2}\mathbf{P}\mathbf{b}\mathbf{b}^T\mathbf{P}=0 \tag{35}$$

where $2\rho^2\ge\delta$ and δ is a positive gain which can be specified to proportional to ρ^2. Taking the derivative of Lyapunov function in (34) and using (31) and (35), yields

$$\begin{aligned}\dot V&=\frac{1}{2}\dot{\mathbf{e}}^T\mathbf{P}\mathbf{e}+\frac{1}{2}\mathbf{e}^T\mathbf{P}\dot{\mathbf{e}}+\frac{1}{\eta_w}\tilde{\mathbf{w}}^T\dot{\tilde{\mathbf{w}}}+\frac{1}{\eta_m}\tilde{\mathbf{m}}^T\dot{\tilde{\mathbf{m}}}\\&\quad+\frac{1}{\eta_\sigma}\tilde{\boldsymbol{\sigma}}^T\dot{\tilde{\boldsymbol{\sigma}}}+\frac{1}{\eta_r}\tilde{\mathbf{r}}^T\dot{\tilde{\mathbf{r}}}\\&=-\frac{1}{2}\mathbf{e}^T(\boldsymbol{\Lambda}^T\mathbf{P}+\mathbf{P}^T\boldsymbol{\Lambda})\mathbf{e}+\tilde{\mathbf{w}}^T(\mathbf{e}^T\mathbf{P}\mathbf{b}\hat{\boldsymbol{\Theta}}-\frac{\dot{\hat{\mathbf{w}}}}{\eta_w})\\&\quad+\tilde{\mathbf{m}}^T(\mathbf{e}^T\mathbf{P}\mathbf{b}\mathbf{A}\hat{\mathbf{w}}-\frac{\dot{\hat{\mathbf{m}}}}{\eta_m})+\tilde{\boldsymbol{\sigma}}^T(\mathbf{e}^T\mathbf{P}\mathbf{b}\mathbf{B}\hat{\mathbf{w}}-\frac{\dot{\hat{\boldsymbol{\sigma}}}}{\eta_\sigma})\\&\quad+\tilde{\mathbf{r}}^T(\mathbf{e}^T\mathbf{P}\mathbf{b}\mathbf{C}\hat{\mathbf{w}}-\frac{\dot{\hat{\mathbf{r}}}}{\eta_r})+\mathbf{e}^T\mathbf{P}\mathbf{b}(\varepsilon-u_{rc})\,.\end{aligned} \tag{36}$$

If the adaptation laws of RCMAC controller are chosen as

$$\dot{\hat{\mathbf{w}}}=\eta_w\mathbf{e}^T\mathbf{P}\mathbf{b}\hat{\boldsymbol{\Theta}} \tag{37}$$

$$\dot{\hat{\mathbf{m}}}=\eta_m\mathbf{e}^T\mathbf{P}\mathbf{b}\mathbf{A}\hat{\mathbf{w}} \tag{38}$$

$$\dot{\hat{\boldsymbol{\sigma}}}=\eta_\sigma\mathbf{e}^T\mathbf{P}\mathbf{b}\mathbf{B}\hat{\mathbf{w}} \tag{39}$$

$$\dot{\hat{\mathbf{r}}}=\eta_r\mathbf{e}^T\mathbf{P}\mathbf{b}\mathbf{C}\hat{\mathbf{w}} \tag{40}$$

then (36) can be rewritten as

$$\dot V=-\frac{1}{2}\mathbf{e}^T(\boldsymbol{\Lambda}^T\mathbf{P}+\mathbf{P}^T\boldsymbol{\Lambda})\mathbf{e}+\mathbf{e}^T\mathbf{P}\mathbf{b}(\varepsilon-u_{rc})\,. \tag{41}$$

If the robust compensator is chosen as

$$u_{rb}=\frac{1}{\delta}\mathbf{b}^T\mathbf{P}\mathbf{e} \tag{42}$$

and use Riccati-like equation (35), then (41) can be rewritten as

$$\begin{aligned}\dot V&=\frac{1}{2}\mathbf{e}^T(-\mathbf{Q}+\frac{2}{\delta}\mathbf{P}\mathbf{b}\mathbf{b}^T\mathbf{P}-\frac{1}{\rho^2}\mathbf{P}\mathbf{b}\mathbf{b}^T\mathbf{P})\mathbf{e}\\&\quad+\mathbf{e}^T\mathbf{P}\mathbf{b}\varepsilon-\frac{1}{\delta}\mathbf{e}^T\mathbf{P}\mathbf{b}\mathbf{b}^T\mathbf{P}\mathbf{e}\\&=-\frac{1}{2}\mathbf{e}^T\mathbf{Q}\mathbf{e}-\frac{1}{2\rho^2}\mathbf{e}^T\mathbf{P}\mathbf{b}\mathbf{b}^T\mathbf{P}\mathbf{e}+\mathbf{e}^T\mathbf{P}\mathbf{b}\varepsilon\\&=-\frac{1}{2}\mathbf{e}^T\mathbf{Q}\mathbf{e}-\frac{1}{2}(\frac{\mathbf{e}^T\mathbf{P}\mathbf{b}}{\rho}-\rho\varepsilon)^2+\frac{1}{2}\rho^2\varepsilon^2\\&\le-\frac{1}{2}\mathbf{e}^T\mathbf{Q}\mathbf{e}+\frac{1}{2}\rho^2\varepsilon^2\,.\end{aligned} \tag{43}$$

Assume $\varepsilon\in L_2[0,T]$, $\forall T\in[0,\infty]$. Integrating (43) yields

$$V(T)-V(0)\le-\frac{1}{2}\int_0^T\mathbf{e}^T\mathbf{Q}\mathbf{e}dt+\frac{1}{2}\rho^2\int_0^T\varepsilon^2dt\,. \tag{44}$$

Since $V(T)\ge 0$, (48) implies the following inequality

$$\frac{1}{2}\int_0^T\mathbf{e}^T\mathbf{Q}\mathbf{e}dt\le V(0)+\frac{1}{2}\rho^2\int_0^T\varepsilon^2dt\,. \tag{45}$$

Using (34), (45) is equivalent to (32). Since $V(0)$ is finite, if the approximation error $\varepsilon\in L_2$, that is $\int_0^T\varepsilon^2dt<\infty$, the adaptive RCMAC control system is asymptotically stable with L_2 tracking performance in the Lyapunov sense.

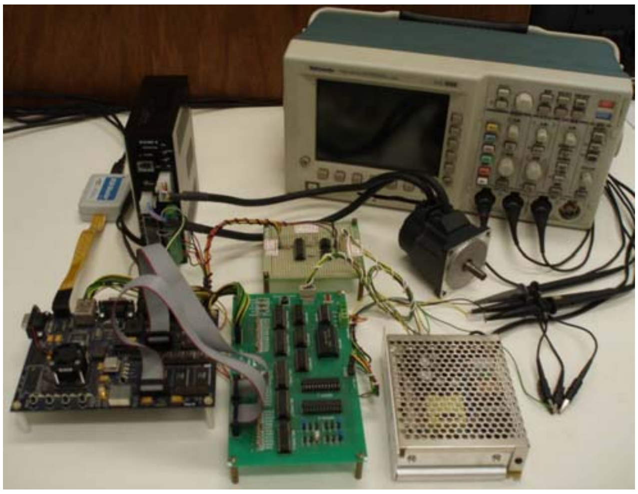

Fig. 3 Hardware experimental environment.

4 Experimental results

Field programmable gate array (FPGA) is a fast prototyping IC component. This kind of IC incorporates the architecture of a gate array and programmability of a programmable logic device. Instead of being restricted to any predetermined hardware function, an FPGA allows designers to program product features and functions, adapt to new standards, and reconfigure hardware for specific applications even after the product has been installed in the field. The designers can use an FPGA to implement any logical function that an application-specific integrated circuit (ASIC) could perform, but the ability to update the functionality after shipping offers advantages for many applications. An FPGA implementation is capable of providing real-time performance with designer controlled power consumption and computationally intensive by digital signal processing tasks.

In this paper, an FPGA with the Nios II processor and BLDC motor are used to construct the hardware experimental environment as shown in Fig. 3. In hardware selection, the Altera Stratix II series FPGA chip is used to construct the BLDC motor control system for hardware implementation. Comparing the FPGA-based control experimental setup with PC-based control setup and DSP-based control setup, the advantages for implementation with an FPGA (instead of an ASIC) not only includes rapid prototyping, shorter time to market, the ability to re-program in the field for debugging, lower NRE costs, long product life cycle to mitigate obsolescence risk, but also consumes less power, in terms of core IC power consumption and especially in terms of the board-level power consumption, than the PC and DSP implementation. FPGA with the system on programming chip (SOPC) structure contains floating calculations. SOPC Builder is an exclusive Quartus® II software tool enabling users to rapidly and easily build systems and evaluate embedded systems [24].

The block diagram of BLDC motor control system is combined by the hardware program modules and software Nios II programming interface, as depicted in Fig. 4. The hardware program modules include the frequency divider module, the position command generator module, the encoder counter module, the function generator module and the D/A signal controller module. The frequency divider module is used to supply the system clock (clk) for all module circuits, the position command generator generates the position command correspond with reference model, the encoder counter module uses the accumulator to increase the resolution of the rotor position for 12 bits up to 16 bits, the function generator module uses the look-up table skill to produce the receptive-field basis function in equation (3) and the D/A signal controller module is used to control the external D/A converter circuit. The BLDC motor system offers high performance and simple operation from a compact driver and motor. The specifications of the adopted BLDC motor system manufactured by the Orientalmotor Company are outlined in Table 1 [25].

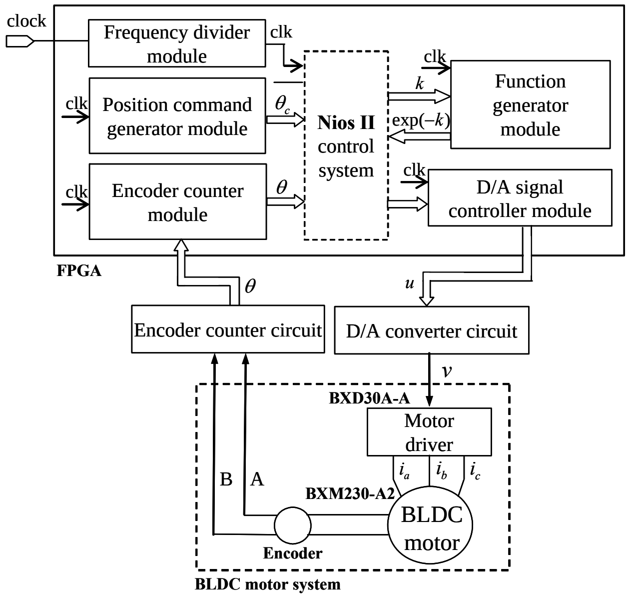

Fig. 4 Block diagram of BLDC motor control system.

Table 1 The specifications of BLDC motor system

Output power HP (W)	1/25HP (30W)
Power supply	Single-phase 100~115VAC
Rated current	1.4 A
Gear/ shaft type	Round Shaft
Variable speed range	30 ~ 3000 r/min
Rated torque	$0.1\ \mathrm{N \cdot m}$
Moment of inertia	$1.5 \cdot 10^{-4}\ \mathrm{kg \cdot m^2}$
Load of inertia	$0.088 \cdot 10^{-4}\ \mathrm{kg \cdot m^2}$
Components	BXD30A-A (Driver) BXM230-A2 (Motor)
Control detection system	Optimal encoder (500P/R)

The external peripheral interfaces are used to transmit and receive the motor driver signals through 12-bits optical encoder counter circuit, two 12-bits D/A converter circuit and a motor driver signal circuit. Additionally, every IC that connects with the FPGA board uses asynchronous bus transceiver IC to protect the current reflows to FPGA board. The 12-bits optical encoder counter circuit is comprised of an encoder signal delay IC, motor rotational direction gauge IC, and three 4-bits up-down counter ICs. The purpose of the 12-bits optical encoder counter circuit is designed to receive and calculate the rotator angle of the BLDC motor from the optical encoder. The encoder signal delay IC is used to delay the A and B phase signals of the optical encoder for multiply the resolution. The motor rotational direction gauge IC uses those signals to gauge the rotator direction of the BLDC motor. Finally, the actually rotator angle of the BLDC motor can be obtained by three 4-bis synchronous up/down counter ICs. For the choice for D/A converter circuit, two D/A ICs are adopted to transfer the digital signals from the FPGA to the analog signals for controlling the BLDC motor and observing the control performance. This kind of IC has two channels, which have three output voltage ranges of 0 to +5V, 0 to +10V and -5V to +5V. The switching approach is used for selecting different output voltage range. The motor driver signal circuit includes the optical encoder signal circuit and the motor rotational direction control signal circuit. The optical encoder signal circuit is designed to raise the encoder signal voltage up from the motor driver. The motor rotational direction control signal circuit is used to raise the motor rotational direction control voltage up from driver.

The proposed control algorithm is realized in the Nios II programming interface. The software flowchart of the control algorithm is shown in Fig. 5. In the main program, the controller parameters are initialized. Next, an interrupt interval for the interrupt service routine (ISR) with a 1msec sampling rate is set. Then, the controller sample times can be governed by the built-in timer, which generates periodic interruptions.

Table 2 Fuzzy control rules base

$\dot{e}$ \ e	NB	NS	ZO	PS	PB
NB	-4.0	-3.0	-2.0	-1.0	0.0
NS	-3.0	-2.0	-1.0	0.0	1.0
ZO	-2.0	-1.0	0.0	1.0	2.0
PS	-1.0	0.0	1.0	2.0	3.0
PB	0.0	1.0	2.0	3.0	4.0

In order to illustrate the effectiveness of the proposed design method, a fuzzy controller [26] and the proposed adaptive RCMAC control are compared. In the experiments, a second-order transfer function with 0.3 sec rise time, of the following form is chosen as the reference model for the periodic step command

$$\frac{\omega_n^2}{s^2+2\xi\omega_n s+\omega_n^2}=\frac{400}{s^2+40s+400} \tag{46}$$

where s is the Laplace operator, ξ is damping ratio (set as one for critical damping) and ω_n is un-damped natural frequency. For demonstrating the tracking ability, the frequencies of reference trajectories are changed after the 5.5th sec.

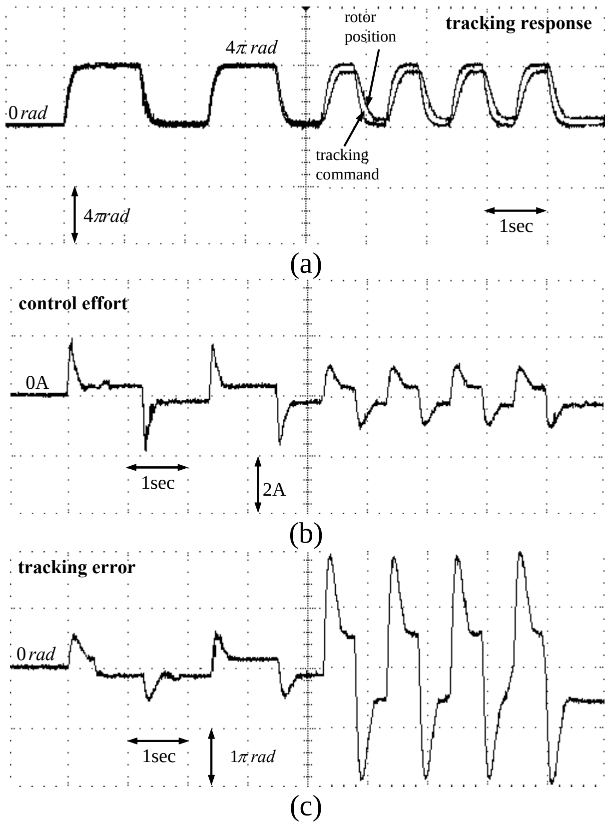

Fig. 6 Experimentation results of fuzzy control.

First a fuzzy controller is applied to the BLDC motor deriver. The fuzzy control rules are given in the following form: [26]

Rule i:

IF e is F_e^i and $\dot{e}$ is $F_{\dot{e}}^i$, THEN u is ρ_i (47)

where ρ_i , $i=1,2,...,n$ are the singleton control actions, and F_e^i and $F_{\dot{e}}^i$ are the labels of the fuzzy sets with Gaussian membership function. The fuzzy rules are summarized in Table 2, where the fuzzy labels used in this paper are negative big (NB), negative medium (NM), negative small (NS), zero (ZO), positive small (PS), positive medium (PM) and positive big (PB). The fuzzy rules in Table 2 are constructed in such a way that e and $\dot{e}$ will approach to zero with fast rise time and without large overshoot. The defuzzification of the controller output is accomplished by the method of sum of weightings. The experimental results of the fuzzy controller are shown in Fig. 6. The tracking response is depicted in Fig. 6(a), the associated control effort is depicted in Fig. 6(b), and the tracking error is depicted in Fig. 6(c), respectively. From the experimental results, the fuzzy controller can achieve tracking performance for a start; however, the degenerate tracking response occurs as the frequency of the input command is increased at the 5.5th second.

Then, the proposed adaptive RCMAC control is applied to control the BLDC motor deriver again. It should be emphasized that the development of adaptive RCMAC control system does not need to know the system dynamics of the controlled system. For practical implementation, the parameters of the adaptive RCMAC control system can be on-line tuned by the proposed adaptive laws without the need of system parameters. For the control system, choose $\mathbf{Q} = \mathbf{I}$, $k_1 = 2$, $k_2 = 1$. By solving $\mathbf{A}_\mathbf{m}^T \mathbf{P} + \mathbf{P}\mathbf{A}_\mathbf{m} = -\mathbf{Q}$, obtain

$$\mathbf{P} = \begin{bmatrix} 1.7625 & 0.7812 \\ 0.7812 & 0.8088 \end{bmatrix}. \tag{48}$$

The learning-rates for adaptive laws are chosen as $\eta_w = 0.02$, $\eta_m = \eta_\sigma = \eta_r = 0.0002$ and $\delta = 0.5$. All the gains are chosen to achieve better transient control performance in considering the requirement of stability and possible operating conditions. The experimental results of the adaptive RCMAC control system are depicted in Fig. 7. The tracking response is depicted in Fig. 7(a), the associated control effort is depicted in Fig. 7(b), and tracking error is depicted in Fig. 7(c), respectively. It can be seen that there is no chattering phenomena in the control effort and perfect tracking response can be obtained after initial transient response. From the comparison of experimental results between Figs. 6 and 7, it is shown that the adaptive RCMAC control system can achieve better tracking performance than fuzzy control method.

5 Conclusions

In this paper, an adaptive RCMAC control system with a training algorithm for a brushless DC motor has been successfully developed and implemented based on the field programmable gate array (FPGA) approach. The proposed control scheme is composed of an RCMAC controller and a robust compensator. The RCMAC controller is used to mimic an ideal controller, and the robust compensator is designed to compensate for the approximation error between the ideal controller and the RCMAC controller. The Lyapunov stability theory is utilized to derive the parameter tuning algorithm, so the proof of stability analytic shows that the output of the system can asymptotically stable with L_2 tracking performance in the Lyapunov sense. The implementation of the control system using the FPGA can achieve the characteristics of small size, fast execution speed and less memory. Finally, the effectiveness of the proposed adaptive RCMAC control has been verified by experimental results. The experimental results demonstrate that the proposed adaptive RCMAC control scheme can achieve favorable control performance.

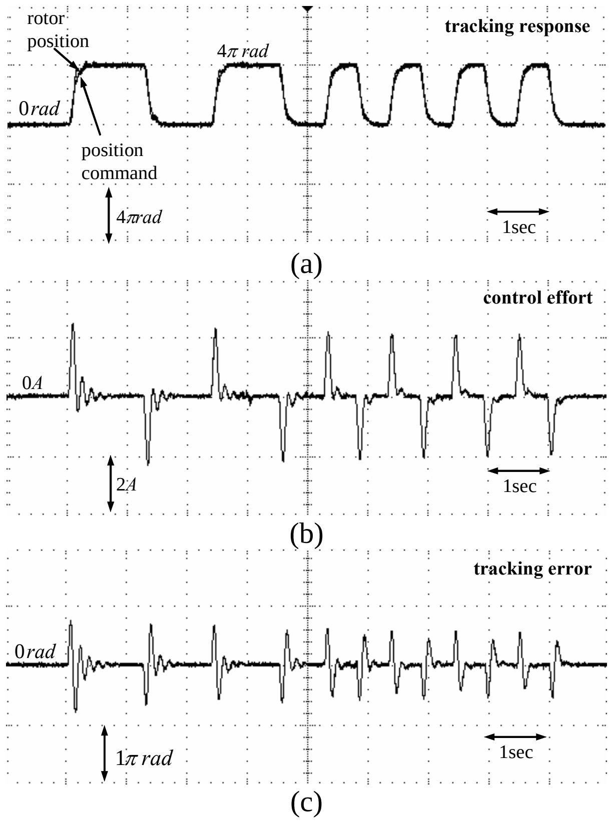

Fig. 7 Experimentation results of adaptive RCMAC control.

References:

[1] C.T. Lin, C.S.G. Lee, *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems*, Prentice -Hall, Englewood Cliffs, NJ, 1996.

[2] C.F. Hsu, C.M. Lin, T.Y. Chen, Neural-network-identification-based adaptive control of wing rock motion, *IEE Proc Control Theory Appl.*, Vol.152, No.1, 2005, pp. 65-71.

[3] M.A. Duarte-Mermoud, A.M. Suarez, D.F. Bassi, Multivariable predictive control of a pressurized tank using neural networks, *Neural Comput Appl.*, Vol.15, No.1, 2005, pp. 18-25.

[4] Y.G. Leu, W.Y. Wang, T.T. Lee, Observer-based direct adaptive fuzzy-neural control for

nonaffine nonlinear systems, *IEEE Trans Neural Netw.*, Vol.16, No.4 , 2005, pp. 853-861.

[5] M. Zheng, M. Tarbouchi, D. Bouchard, J. Dunfield, FPGA implementation of a neural network control system for a helicopter, *WSEAS Trans Systems*, Vol.5, 2006, pp. 1748-1751.

[6] C. Lee, C. Lin, FPGA implementation of a fuzzy cellular neural network for image processing, *WSEAS Trans Circuits and Systems*, Vol.6, 2007, pp. 414-419.

[7] S.F. Su, T. Tao, T.H. Hung, Credit assigned CMAC and its application to online learning robust controllers, *IEEE Trans Syst Man Cybern B*, Vol.33, No.2, 2003, pp. 202-213.

[8] Y.F. Peng, R.J. Wai, C.M. Lin, Implementation of LLCC-resonant driving circuit and adaptive CMAC neural network control for linear piezoelectric ceramic motor, *IEEE Trans Ind Electron.*, Vol.51, No.1, 2004, pp. 35-48.

[9] C.M. Lin, Y.F. Peng, Adaptive CMAC-based supervisory control for uncertain nonlinear systems, *IEEE Trans Syst Man Cybern B*, Vol.34, No.2, 2004, pp. 1248-1260.

[10] C.M. Lin, Y.F. Peng, Missile guidance law design using adaptive cerebellar model articulation controller, *IEEE Trans Neural Netw.*, Vol.16, No.3, 2005, pp. 636-644.

[11] S.F. Su, Z.J. Lee, Y.P. Wang, Robust and fast learning for fuzzy cerebellar model articulation controllers, *IEEE Trans Syst Man Cybern B*, Vol.36, No.1, 2006, pp. 203-208.

[12] C.M. Lin, C.F. Hsu, Supervisory recurrent fuzzy neural network control of wing rock for slender delta wings, *IEEE Trans Fuzzy Syst.*, Vol.12, No.5, 2004, pp. 733-742.

[13] C.S. Leung, A.C. Tsoi, Combined learning and pruning for recurrent radial basis function networks based on recursive least square algorithms, *Neural Comput Appl.*, Vol.15, No.1, 2005, pp. 62-78.

[14] Y. Maeda, Y. Fukuda, T. Matsuoka, Pulse density recurrent neural network systems with learning capability using FPGA, *WSEAS Trans Circuits and Systems*, Vol.7, 2008, pp. 321-330.

[15] Y. Dote, S. Kinoshita, *Brushless Servomotors: Fundamentals and Applications*, Clarendon Press Oxford, 1990.

[16] Y. Liu, Z.Q. Zhu, D, Howe Direct torque control of brushless DC drives with reduced torque ripple, *IEEE Trans. Ind. Appl.*, Vol.41, No.2, 2005, pp. 599-608.

[17] A. Rubaai, D. Ricketts, M.D. Kankam, Development and implementation of an adaptive fuzzy-neural-network controller for brushless drives, *IEEE Trans. Ind. App.l*, Vol.38, No.2, 2002, pp. 441-447.

[18] A. Rubaai, A.R. Ofoli, D. Cobbinah, DSP-based real-time implementation of a hybrid H^{∞} adaptive fuzzy tracking controller for servo-motor drives, *IEEE Trans Ind Appl.*, Vol.43, No.2, 2007, pp. 476-484.

[19] C.M. Lin, C.H. Chen, Robust fault-tolerant control for biped robot using recurrent cerebellar model articulation controller, *IEEE Trans Syst Man Cybern B*, Vol.37, No.1, 2007, pp. 110-123.

[20] L.X. Wang, *Adaptive Fuzzy Systems and Control: Design and Stability Analysis*. Prentice-Hall, Englewood Cliffs, NJ, 1994.

[21] J.J.E. Slotine, W.P. Li, *Applied nonlinear control*, Prentice-Hall, Englewood Cliffs, NJ, 1991.

[22] W.Y. Wang, M.L. Chan, C.C.J. Hsu, T.T. Lee, H^{∞} tracking-based sliding mode control for uncertain nonlinear systems via an adaptive fuzzy-neural approach, *IEEE Trans. Syst., Man, Cybern. B: Cybern.*, Vol. 32, No. 4, 2002, pp. 483-492.

[23] Y.F. Peng, C.M. Lin, Intelligent motion control of linear ultrasonic motor with H^{∞} tracking performance, *IET Contr. Theory Appl.*, Vol. 1, No. 1, 2007, pp. 9-17.

[24] [Online] http://www.altera.com/

[25] [Online] http://www.orientalmotor.com/

[26] L.X. Wang, *A Course in Fuzzy Systems and Control*, Prentice-Hall, Englewood Cliffs, NJ, 1997.

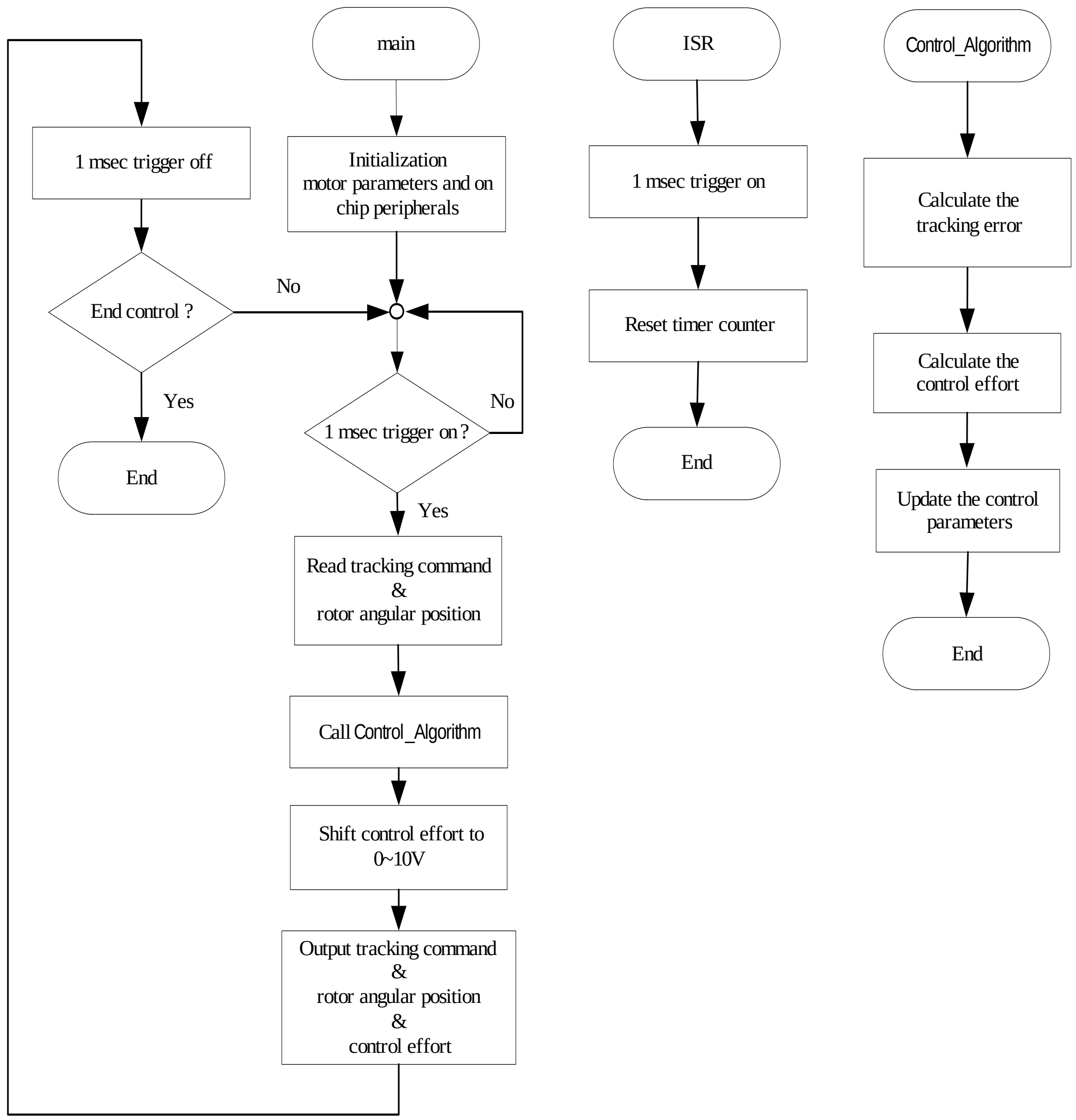

Fig. 5 Flowchart of the software implemented.