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

Chih-Min Lin¹ Chun-Fei Hsu² Chao-Ming Chung¹
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 [2-4]. 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. 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.

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
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

$$y = \sum_{q=1}^{n_q} w_q b_q$$

where $b_q$ is the receptive-field basis function of the $q$th receptive-field, $w_q$ denotes the connecting weight value of the $q$th receptive-field, and the receptive-field basis function is defined as

$$b_q(z) = \prod_{j=1}^{n} \phi_{q_k}(z_{j_k})$$

where $z = [z_j, z_{j+1}, \cdots, z_n]^T \in \mathbb{R}^n$ is the input vector and the Gaussian function is adopted as the receptive-field basis function which can be represented as

$$\phi_{q_k}(z_{j_k}) = \exp \left( \frac{-(z_{j_k} - m_{j_k})^2}{\sigma_{j_k}^2} \right), \text{ for } k = 1, 2, \cdots, n_g$$

where $\phi_{q_k}(z_{j_k})$ presents the $k$th block of the $j$th input $z_j$ with the mean $m_{j_k}$ and variance $\sigma_{j_k}$ and $n_g$ is the number of blocks. In addition, the input of this block can be represented as

$$z_{j_k} = z_j(N) + r_{j_k} \phi_{q_k}(N-1)$$

where $r_{j_k}$ is the recurrent weight of the recurrent unit. It is clear that the input of this block contains a memory term $\phi_{q_k}(N-1)$, 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

$$y = w^T \Theta(z, m, \sigma, r)$$

where

$$w = [w_1, w_2, \cdots, w_n]^T$$

$$\Theta = [b_1, b_2, \cdots, b_n]^T$$

$$m = [m_1, m_2, \cdots, m_{n_q}]^T$$

$$\sigma = [\sigma_1, \sigma_2, \cdots, \sigma_{n_q}]^T$$

$$r = [r_1, r_2, \cdots, r_{n_q}]^T$$
This implies that there exists an RCMAC of (5) such that it can uniformly approximate a nonlinear even time-varying function $\Omega$. By the universal approximation theorem, there exist an ideal weight vectors such that [19, 20]

$$\Omega = y' + \Delta = w'^T\Theta(z, m', \sigma', r') + \Delta$$

where $\Delta$ denotes the approximation error, $w'$ and $\Theta'$ are the optimal parameter vectors of $w$ and $\Theta$, respectively, and $m'$, $\sigma'$ and $r'$ are the optimal parameter vectors of $m$, $\sigma$ and $r$, respectively. In fact, the optimal parameter vectors that are needed to best approximate a given nonlinear function $\Omega$ cannot be determined. Thus, an estimation function is defined as

$$\hat{y} = \hat{w}'\Theta(z, \hat{m}, \hat{\sigma}, \hat{r}) = \hat{w}'\hat{\Theta}$$

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

$$\tilde{y} = \hat{y} - \hat{w}'\hat{\Theta} = \hat{w}'\hat{\Theta} + \Delta$$

where $\tilde{y} = \hat{y} - \hat{w}'\hat{\Theta}$ and $\hat{\Theta} = \Theta' - \hat{\Theta}$. In the following, some tuning laws will be derived to online 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]

$$\tilde{\Theta} = A'T\hat{m} + B'T\hat{\sigma} + C'T\hat{r} + H$$

where $\hat{m} = m' - \hat{m}$; $\hat{\sigma} = \sigma' - \hat{\sigma}$; $\hat{r} = r' - \hat{r}$; $H$ is a vector of higher-order terms;

$$A = \begin{bmatrix} \frac{\partial \Theta_1}{\partial m_1} & \cdots & \frac{\partial \Theta_1}{\partial m_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial \Theta_n}{\partial m_1} & \cdots & \frac{\partial \Theta_n}{\partial m_n} \end{bmatrix}_{m \rightarrow \hat{m}}; \quad B = \begin{bmatrix} \frac{\partial \Theta_1}{\partial \sigma_1} & \cdots & \frac{\partial \Theta_1}{\partial \sigma_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial \Theta_n}{\partial \sigma_1} & \cdots & \frac{\partial \Theta_n}{\partial \sigma_n} \end{bmatrix}_{\sigma \rightarrow \hat{\sigma}};$$

$$C = \begin{bmatrix} \frac{\partial \Theta_1}{\partial r_1} & \cdots & \frac{\partial \Theta_1}{\partial r_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial \Theta_n}{\partial r_1} & \cdots & \frac{\partial \Theta_n}{\partial r_n} \end{bmatrix}_{r \rightarrow \hat{r}}.$$ Substituting of (14) into (13) yields

$$\tilde{y} = \hat{w}'\hat{\Theta} + \tilde{w}'(A'T\hat{m} + B'T\hat{\sigma} + C'T\hat{r} + h) + \tilde{w}'\tilde{\Theta} + \Delta$$

where $\tilde{w}'A'T\hat{m} = \tilde{m}'A\hat{w}$, $\tilde{w}'B'T\hat{\sigma} = \tilde{\sigma}'B\hat{w}$ and $\tilde{w}'C'T\hat{r} = \tilde{r}'C\hat{w}$ are used since they are scales, and the uncertain term $\varepsilon = \tilde{w}'h + \tilde{w}'\tilde{\Theta} + \Delta$ 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]

$$i_q = -\frac{R_s}{L_q}i_q - \frac{L_m}{L_q}\omega_l i_d + \frac{1}{L_d}v_q - \frac{L_m}{L_q}\omega_r$$

$$i_d = -\frac{R_s}{L_d}i_d + \frac{L_m}{L_d}\omega_l i_q + \frac{1}{L_d}v_d$$

$$L_q = L_{iq} + L_{mr}$$

$$L_d = L_{id} + L_{md}$$

$$T_r = \frac{3N}{2} [\lambda_m i_q + (L_d - L_q)i_di_d]$$

where $i_d$ and $i_q$ 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, $V_d$ and $V_q$ are the $d$ and $q$ axes stator voltage, respectively, $L_{iq}$ is the stator leakage inductance, $L_{md}$ and $L_{mq}$ are the $d$ and $q$ axes magnetizing inductances, respectively, $\omega_r$ is the electrical rotor angular velocity, $\lambda_m$ 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

$$J \frac{2}{N} \dot{\omega}_r + B \frac{2}{N} \omega_r = T_r - T_L$$
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 \( i_d \) become zero. Therefore, the equation of BLDC motor driver can be rewritten as [15]

\[
i_{qv} = -\frac{R_q}{L_q} i_{qv} + \frac{1}{L_q} V_{qv} - \frac{\lambda_m}{L_q} \omega_r \tag{22}
\]

\[
\dot{\omega}_r = \frac{3}{2} \left( \frac{N}{2} \right) \lambda_m i_{qv} - \frac{B}{J} \omega_r - \frac{N}{2} T_L \tag{23}
\]

and the torque equation is expressed as

\[
T_e = \frac{3}{2} \frac{N}{2} \lambda_m i_{qv} = k_i i_{qv} \tag{24}
\]

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

\[
\dot{\theta} = f \dot{\theta} + g u + h \tag{25}
\]

where \( \theta = \int \omega_r dt \) is the position of the rotor,

\[
f = -\frac{B}{J}, \quad g = \frac{N}{2} \frac{k_i}{J}, \quad h = -\frac{N}{2} T_L, \quad \text{and} \quad u = i_{qv} \quad \text{is the control effort.}
\]

### 3.2 Control system design

The control objective of the BLDC motor driver is to find a control law so that the rotor position \( \theta \) can track the position command \( \theta_c \) 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]

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

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

\[
\dot{e} + k_1 \dot{e} + k_2 e = 0 \tag{28}
\]

If \( k_i \) and \( k_s \) are chosen such that all roots of the polynomial \( h(s) \Delta s^2 + k_i s + k_2 \) 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

\[
u = u_{nc} + u_r \quad \text{where} \quad u_r = A_{nc} e + b(u^* - u_{nc} - u_{nc}) \tag{30}
\]

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

\[
\dot{e} = A_m e + b(u^* - u_{nc} - u_{nc}) \tag{31}
\]

In case of the existence of \( \varepsilon \), consider a specified \( L_2 \) tracking performance [22], [23]

\[
\int_0^T e^T Q e \, dt \leq e^T(0) P e(0) + \frac{1}{\eta_w} \tilde{w}^T(0) \tilde{w}(0)
\]

\[
+ \frac{1}{\eta_m} \tilde{m}^T(0) \tilde{m}(0) + \frac{1}{\eta_o} \tilde{\sigma}^T(0) \tilde{\sigma}(0)
\]

\[
+ \frac{1}{\eta_r} \tilde{r}^T(0) \tilde{r}(0) + \rho^2 \int_0^T e^2 \, dt \tag{32}
\]
where \( Q \) and \( P \) are symmetric positive definite matrices, \( \eta_v, \eta_w, \eta_m \) and \( \eta_e \) are positive constants, and \( \rho \) is a prescribed attenuation level. If the system starts with initial conditions \( e(0) = 0 \), \( \dot{w}(0) = 0 \), \( \dot{m}(0) = 0 \), \( \dot{\sigma}(0) = 0 \) and \( \dot{r}(0) = 0 \), the \( L_2 \) tracking performance in (32) can be rewritten as

\[
\sup_{t \in [0, T]} \left\| e(t) \right\|^2 \leq \rho \tag{33}
\]

where \( \left\| e \right\|^2 = \int_0^T e^T Q e dt \) and \( \left\| \dot{e} \right\|^2 = \int_0^T \dot{e}^T \dot{e} dt \). The attenuation constant \( \rho \) can be specified by the designer to achieve the desired attenuation ratio between \( \| e \| \) and \( \| \dot{e} \| \). 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

\[
V = \frac{1}{2} e^T P e + \frac{1}{2 \eta_v} \dot{w}^T \dot{w} + \frac{1}{2 \eta_m} \dot{m}^T \dot{m}
+ \frac{1}{2 \eta_e} \dot{\sigma}^T \dot{\sigma} + \frac{1}{2 \eta_r} \dot{r}^T \dot{r} \tag{34}
\]

and the derivative of Lyapunov function in (34) and using (31) and (35), yields

\[
\dot{V} = \frac{1}{2} \dot{e}^T P e + \frac{1}{2} e^T \dot{P} e + \frac{1}{2 \eta_v} \ddot{w}^T \dot{w} + \frac{1}{2 \eta_m} \ddot{m}^T \dot{m} 
+ \frac{1}{\eta_v} \dddot{\sigma}^T \dot{\sigma} + \frac{1}{\eta_r} \dddot{r}^T \dot{r}
= -\frac{1}{2} e^T (\Lambda^T P + P^T \Lambda) e + \dot{w}^T (\dot{e}^T P \dot{\Theta} - \dot{\dot{w}})
+ \dot{m}^T (\dot{e}^T P \dot{A} \dot{w} - \dot{\dot{m}}) + \dot{\sigma}^T (\dot{e}^T P \dot{B} \dot{w} - \dot{\dot{\sigma}})
+ \dot{r}^T (\dot{e}^T P \dot{C} \dot{w} - \dot{\dot{r}}) + e^T P (\dot{e} - u_e). \tag{36}
\]

If the adaptation laws of RCMAC controller are chosen as

\[
\dot{w} = \eta_v e^T P \dot{\Theta} \tag{37}
\]
\[
\dot{m} = \eta_m e^T P \dot{A} \dot{w} \tag{38}
\]
\[
\dot{\sigma} = \eta_e e^T P \dot{B} \dot{w} \tag{39}
\]
\[
\dot{r} = \eta_r e^T P \dot{C} \dot{w} \tag{40}
\]
then (36) can be rewritten as

\[
\dot{V} = -\frac{1}{2} e^T (\Lambda^T P + P^T \Lambda) e + e^T P (\dot{e} - u_e) \tag{41}
\]

If the robust compensator is chosen as

\[
u_e = \frac{1}{\delta} b^T \dot{P} e \tag{42}
\]
and use Riccati-like equation (35), then (41) can be rewritten as

\[
\dot{V} = -\frac{1}{2} e^T (\Lambda^T P + P^T \Lambda) e + e^T P (\dot{e} - u_e)
+ e^T P b e - \frac{1}{\delta} e^T P b \dot{P} e
= -\frac{1}{2} e^T Q e - \frac{1}{2 \rho^2} e^T P b \dot{P} e + e^T P \dot{e}
= -\frac{1}{2} e^T Q e - \frac{1}{2} (e^T P b - \rho \delta) e^T e + \frac{1}{2 \rho^2} e^T e
\leq -\frac{1}{2} e^T Q e + \frac{1}{2} \rho^2 \dot{e}^T e. \tag{43}
\]

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

\[
V(T) - V(0) \leq -\frac{1}{2} \int_0^T e^T Q e dt + \frac{1}{2 \rho^2} \int_0^T \dot{e}^T e dt. \tag{44}
\]

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

\[
\frac{1}{2} \int_0^T e^T Q e dt \leq V(0) + \frac{1}{2 \rho^2} \int_0^T \dot{e}^T e dt. \tag{45}
\]

Using (34), (45) is equivalent to (32). Since \( V(0) \) is finite, if the approximation error \( e \in L_2 \), that is \( \int_0^T \dot{e}^T e dt < \infty \), 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.](image)

Table 1 The specifications of BLDC motor system

<table>
<thead>
<tr>
<th>Output power HP (W)</th>
<th>1/25HP (30W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power supply</td>
<td>Single-phase</td>
</tr>
<tr>
<td>Gear/ shaft type</td>
<td>Round Shaft</td>
</tr>
<tr>
<td>Variable speed range</td>
<td>30 ~ 3000 r/min</td>
</tr>
<tr>
<td>Rated torque</td>
<td>0.1 N·m</td>
</tr>
<tr>
<td>Moment of inertia</td>
<td>1.5·10⁻⁴ kg·m²</td>
</tr>
<tr>
<td>Load of inertia</td>
<td>0.088·10⁻⁴ kg·m²</td>
</tr>
<tr>
<td>Components</td>
<td>BXD30A-A (Driver)</td>
</tr>
<tr>
<td>Control detection system</td>
<td>Optimal encoder (500P/R)</td>
</tr>
</tbody>
</table>

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 refloows 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-bits
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
to raise 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

<table>
<thead>
<tr>
<th></th>
<th>e</th>
<th>dot e</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB</td>
<td>-4.0</td>
<td>-3.0</td>
</tr>
<tr>
<td>NS</td>
<td>-3.0</td>
<td>-2.0</td>
</tr>
<tr>
<td>ZO</td>
<td>-2.0</td>
<td>-1.0</td>
</tr>
<tr>
<td>PS</td>
<td>-1.0</td>
<td>0.0</td>
</tr>
<tr>
<td>PB</td>
<td>0.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>

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

\[
\omega_n^2 = \frac{400}{s^2 + 2\zeta\omega_n s + \omega_n^2} = \frac{s^2 + 40s + 400}{s^2 + 2\zeta\omega_n s + \omega_n^2}
\]

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

![Fuzzy control rules base](image)

![Experimentation results of fuzzy control](image)

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 \( F_e^i \), THEN \( u \) is \( \rho_i \)  

(47)

where \( \rho_i \), \( i = 1,2,\ldots,n \) are the singleton control
actions, and \( F_e^i \) and \( F_e^i \) 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.
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 \( Q = I \), \( k_1 = 2 \), \( k_2 = 1 \). By solving \( A_m^T P + PA_m = -Q \), obtain

\[
P = \begin{bmatrix} 1.7625 & 0.7812 \\ 0.7812 & 0.8088 \end{bmatrix}.
\] (48)

The learning-rates for adaptive laws are chosen as \( \eta_{w} = 0.02 \), \( \eta_w = \eta_r = \eta_i = 0.0002 \) 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:


Calculate the control effort

Control Algorithm

Initialization motor parameters and on chip peripherals

main

ISR

1 msec trigger on

Calculate the tracking error

Reset timer counter

Control Algorithm

1 msec trigger on

Calculate the control effort

Update the control parameters

End

Update the control parameters

End

Fig. 5 Flowchart of the software implemented.