Expert Systems with Applications 38 (2011) 11866–11872

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

FPGA-based adaptive PID control of a DC motor driver via sliding-mode approach

Chun-Fei Hsu *, Bore-Kuen Lee

Department of Electrical Engineering, Chung Hua University, Hsinchu, 300 Taiwan, Republic of China

ARTICLE INFO

Keywords:
PID control
Adaptive control
DC motor driver
FPGA chip

ABSTRACT

The proportional-integral-derivative (PID) controller has been extensively applied in practical industry due to its appealing characteristics such as simple architecture, easy design and parameter tuning without complicated computation. However, the PID controller usually needs some a priori manual retuning to make a successful industrial application. To attack this problem, this paper proposes an adaptive PID (APID) controller which is composed of a PID controller and a fuzzy compensator. Without requiring preliminary offline learning, the PID controller can automatically online tune the control gains based on the gradient descent method and the fuzzy compensator is designed to eliminate the effect of the approximation error introduced by the PID controller upon the system stability in the Lyapunov sense. Finally, the proposed APID control system is applied to a DC motor driver and implemented on a field-programmable gate array (FPGA) chip for possible low-cost and high-performance industrial applications. It is shown by the experimental results that the favorable position tracking performance for the DC motor driver can be achieved by the proposed APID control scheme after learning of the controller parameters.

1. Introduction

An ideal controller required reasonably precise knowledge of the dynamic model of a process to achieve a favorable control performance (Slotine & Li, 1991). Since the system parameters and external load disturbance may be unknown or perturbed, the ideal controller cannot be implemented in real-time practical applications. If all uncertainties existed in the controlled system are bounded, a sliding-mode control can provide system dynamics with an invariance property to uncertainties once the system dynamics is controlled in the sliding mode (Slotine & Li, 1991). Once the states of the controlled system enter the sliding mode, the system dynamics will be determined by the choice of a sliding hyperplane and will be independent of uncertainties and external disturbances. However, the sliding-mode control strategy usually suffers from large control chattering caused by a switching function in the control law. It may wear mechanism coupling and excite unmodeled system dynamics.

Due to its appealing advantages such as simple architecture and easy design, the proportional-integral-derivative (PID) control has been extensively applied in practical industry up to date, even though lots of new control techniques have been proposed recently. The key for designing the PID controller is the determination of the proportional gain, integral gain and derivative gain. However, these control gains of the PID controller need some manual retuning to make a successful industrial application. Recently, the PID controller has been combined with different methods such as fuzzy control (Santos & Dexter, 2002), neural network (Ren, Chen, & Chen, 2008), evolutionary algorithm (Coelho & Bernert, 2009; Juang, Huang, & Liu, 2008) and adaptive control (Chang, Hwang, & Hsieh, 2002; Chang & Yan, 2005; Yu, Chang, & Yu, 2007). However, some of these methods require heavy computation loading and some cannot achieve a satisfactory performance.

A field-programmable gate array (FPGA) chip incorporates the architecture of gate arrays and the programmability of a programmable logic device. An FPGA chip usually consists of more than thousands of logic gates. Some of these logic gates are combined together to form a configurable logic block thereby simplifying high-level circuit design. All the internal logic elements and all the control procedures of the FPGA are executed continuously and simultaneously (http://www.altera.com/). The circuits and algorithms can be developed by the very high speed integrated circuit hardware description language. Therefore, the execution time of FPGA is faster than either the digital signal processor (DSP) or the personal computer (PC). Recently, some researchers adopted the FPGA chip to implement control schemes to allow possible low-cost and high-performance industrial applications (Kung & Tsai, 2007; Lin, Teng, Chen, & Hung, 2008; Shao & Sun, 2007).

In this paper, an adaptive PID (APID) control system is proposed via the sliding-mode approach. The proposed APID control system is composed of a PID controller and a fuzzy compensator. The PID controller can automatically online tune the PID controller gains based on the gradient descent method and the fuzzy compensator is designed to guarantee system stability in the Lyapunov stability sense. The proposed APID controller is then applied to a DC motor

* Corresponding author.
E-mail addresses: fei@chu.edu.tw (C.-F. Hsu), bklee@chu.edu.tw (B.-K. Lee).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.02.185

driver in which the hardware is implemented on an FPGA chip for real-time practical applications. Finally, to investigate the effectiveness of the proposed APID control system, a comparison among the fuzzy sliding-mode control (Choi, Kwak, & Kim, 1999), the adaptive robust PID control (Chang & Yan, 2005), the supervisory recurrent fuzzy network control (Lin & Hsu, 2004) and the proposed APID control is made. Compared with the other control schemes, experimental results show that the proposed APID scheme can achieve a more favorable tracking performance without the chattering phenomena in the control effort after learning of the controller parameters.

2. Problem statement

The motion equation of a DC motor driver can be simplified as (Elmas, Ustun, & Sayan, 2008; Pisano, Davila, Fridman, & Usai, 2008)

$$J\ddot{\theta}(t) + B\dot{\theta}(t) = T_e(t) \tag{1}$$

where J is the moment of inertia, B is the damping coefficient, $\theta(t)$ is the rotor position and $T_e(t)$ is the electric torque. The electric torque $T_e(t)$ is defined as

$$T_e(t) = K_t i_a(t) \tag{2}$$

where K_t is the torque constant and $i_a(t)$ is the torque current. The electric equation of a DC motor driver can be simplified as

$$v_a(t) = R_a i_a(t) + K_b\dot{\theta}(t) + L_a\frac{di_a(t)}{dt} \tag{3}$$

where R_a is the armature resistance, K_b is the back electromotive force coefficient, L_a is the armature inductance and $v_a(t)$ is the applied voltage. Since the armature inductance can be negligible, the dynamics of the DC motor driver can be represented in the following form

$$\ddot{\theta}(t) = f(\theta,\dot{\theta}) + gu(t) \tag{4}$$

where $f(\theta,\dot{\theta}) = -\left(\frac{B}{J} + \frac{K_tK_b}{JR_a}\right)\dot{\theta}(t)$, $g = \frac{K_t}{JR_a} > 0$ is a constant and $u(t) = v_a(t)$ is the control input. The control objective of the DC motor driver is to find a control law so that the rotor position $\theta(t)$ can track the rotor command $\theta_c(t)$ closely. Define a tracking error as

$$e(t) = \theta_c(t) - \theta(t) \tag{5}$$

Assuming that all the parameters in (4) are known, there exits an ideal controller (Slotine & Li, 1991)

$$u^*(t) = \frac{1}{g}[-f(\theta,\dot{\theta}) + \ddot{\theta}_c(t) + k_1\dot{e}(t) + k_2e(t)] \tag{6}$$

where k_1 and k_2 are positive constants. Applying the ideal controller (6) to (4), we can obtain that

$$\ddot{e}(t) + k_1\dot{e}(t) + k_2e(t) = 0 \tag{7}$$

If k_1 and k_2 are chosen to correspond to the coefficients of a Hurwitz polynomial, then $\lim_{t\to\infty}e(t) = 0$ for any starting initial condition. Since the terms $f(\theta,\dot{\theta})$ and g in the system dynamics may be unknown or perturbed in practical applications, the ideal controller cannot be precisely obtained.

To attack this problem, we can take the well known advantage of the sliding-mode control scheme by invoking its insensitivity to parameter variations and external disturbance once the system trajectory reaches and stays on the sliding surface. Before designing the sliding-mode control, rewriting (4), the nominal model of the DC motor driver can be represented as follows

$$\ddot{\theta}(t) = f_n(\theta,\dot{\theta}) + g_nu(t) \tag{8}$$

where $f_n(\theta,\dot{\theta})$ and g_n are the terms that represent the nominal behavior of $f(\theta,\dot{\theta})$ and g, respectively. If the system uncertainties occur, i.e., the system parameters deviate from their nominal values and/or an external disturbance is added into the system, the dynamics of the DC motor driver can be modified as

$$\begin{aligned}\ddot{\theta}(t) &= [f_n(\theta,\dot{\theta}) + \Delta f(\theta,\dot{\theta})] + (g_n + \Delta g)u(t)\\ &= f_n(\theta,\dot{\theta}) + g_nu(t) + w(t)\end{aligned} \tag{9}$$

where $\Delta f(\theta,\dot{\theta})$ and Δg denote the system uncertainties and $w(t)$ is called the lumped uncertainty which is defined as $w(t) = \Delta f(\theta,\dot{\theta}) + \Delta gu(t)$. The lumped uncertainty is assumed to be bounded, i.e., $|w(t)| \leqslant W$ where W is a given positive constant. It is well known that the insensitivity of the controlled system to uncertainties exists in the sliding mode but not during the reaching phase. The system dynamic in the reaching phase is still influenced by uncertainties. In this paper, we consider a total sliding surface as $s(t) = 0$ with

$$s(t) = C(\mathbf{e}) - C(\mathbf{e}_0) - \int_0^t \frac{\partial C(\mathbf{e})}{\partial \mathbf{e}}\mathbf{\Lambda}\mathbf{e}d\tau \tag{10}$$

where $\mathbf{e} = [e(t), \dot{e}(t)]^T$, $\mathbf{\Lambda}$ is a matrix, $C(\mathbf{e})$ is the vector to be designed and $\mathbf{e}_0$ is the initial state of $\mathbf{e}(t)$. Since we have $s(t) = 0$ at $t = 0$, there is no reaching phase as in the sliding-mode control. An associate sliding-mode control law is given by (Slotine & Li, 1991)

$$u_{sc} = u_{eq} + u_{ht} \tag{11}$$

where the equivalent controller u_{eq} is represented as

$$u_{eq} = \frac{1}{g_n}[-f_n(\theta,\dot{\theta}) + \ddot{\theta}_c(t) + k_1\dot{e}(t) + k_2e(t)] \tag{12}$$

and the hitting controller u_{ht} for guaranteeing the system stability is designed as

$$u_{ht} = \frac{W}{g_n}\mathrm{sgn}[s(t)] \tag{13}$$

where $\mathrm{sgn}(\cdot)$ is the sign function. Substituting (11) into (9) yields

$$\dot{\mathbf{e}} = \mathbf{A}\mathbf{e} + \mathbf{b}\{-w(t) - W\mathrm{sgn}[s(t)]\} \tag{14}$$

where $\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -k_2 & -k_1 \end{bmatrix}$ and $\mathbf{b} = [0, 1]^T$. Differentiating (10) and using (14), one can obtain

$$\dot{s}(t) = -w(t) - W\mathrm{sgn}[s(t)] \tag{15}$$

Consider a Lyapunov function candidate in the following form as

$$V_1(s,t) = \frac{1}{2}s^2(t) \tag{16}$$

Differentiating (16) with respect to time and using (15), we have

$$\begin{aligned}\dot{V}_1(s,t) &= s(t)\dot{s}(t) = -w(t)s(t) - W|s(t)|\\ &\leqslant |w(t)||s(t)| - W|s(t)| = -(W - |w(t)|)|s(t)| \leqslant 0\end{aligned} \tag{17}$$

In summary, the sliding-mode control law can guarantee the system stable in the sense of the Lyapunov theorem (Slotine & Li, 1991). However, the sliding-mode control strategy usually suffers from large control chattering caused by the sign function in the control law. It may wear mechanism coupling and excite unmodeled system dynamics. Moreover, switching of high frequency due to the sign function is not easy to implement due to physical limitations on the actuators used in the plants.

3. Design of the APID control system

The block diagram of the adaptive PID (APID) control system for a DC motor driver is proposed as shown in Fig. 1, where the controller output is defined as

$$u_{ac} = u_{pid} + u_{fc} \tag{18}$$

The PID controller u_{pid} is utilized to approximate the ideal controller u^* and the fuzzy compensator u_{fc} is designed to suppress the influence of the residual approximation error between the ideal controller and the PID controller. In this paper, the PID controller is designed as

$$u_{pid} = \hat{k}_P e(t) + \hat{k}_I \int_0^t e(\tau)d\tau + \hat{k}_D \dot{e}(t) \tag{19}$$

where $\hat{k}_P$, $\hat{k}_I$ and $\hat{k}_D$ are the estimation values of the proportional gain, integral gain and derivative gain, respectively. The key for designing a PID controller is the determination of these controller gains.

3.1. Online learning algorithm

This paper proposes a self-tuning algorithm to online determine the controller gains $\hat{k}_P$, $\hat{k}_I$ and $\hat{k}_D$. Substituting (18) into (4) and using (6) yield

$$\dot{\mathbf{e}} = \mathbf{Ae} + \mathbf{b}g(u^* - u_{pid} - u_{fc}) \tag{20}$$

Differentiating (10) and using (20) lead to

$$\dot{s}(t) = g(u^* - u_{pid} - u_{fc}) \tag{21}$$

Multiplying both sides of (21) by $s(t)$ gives

$$s(t)\dot{s}(t) = gs(t)(u^* - u_{pid} - u_{fc}) \tag{22}$$

An important concept of the sliding-mode control is to make the system satisfy the reaching condition and guarantee the sliding condition. According to the gradient descent method and the chain rule, $\hat{k}_P$, $\hat{k}_I$ and $\hat{k}_D$ gains are updated by following tuning laws (Chang & Yan, 2005; Lin & Hsu, 2004)

$$\dot{\hat{k}}_P = -\eta_p \frac{\partial s(t)\dot{s}(t)}{\partial u_{pid}} \frac{\partial u_{pid}}{\partial \hat{k}_P} = \eta_P g s(t) e(t) \tag{23}$$

$$\dot{\hat{k}}_I = -\eta_I \frac{\partial s(t)\dot{s}(t)}{\partial u_{pid}} \frac{\partial u_{pid}}{\partial \hat{k}_I} = \eta_I g s(t) \int_0^t e_i(\tau)d\tau \tag{24}$$

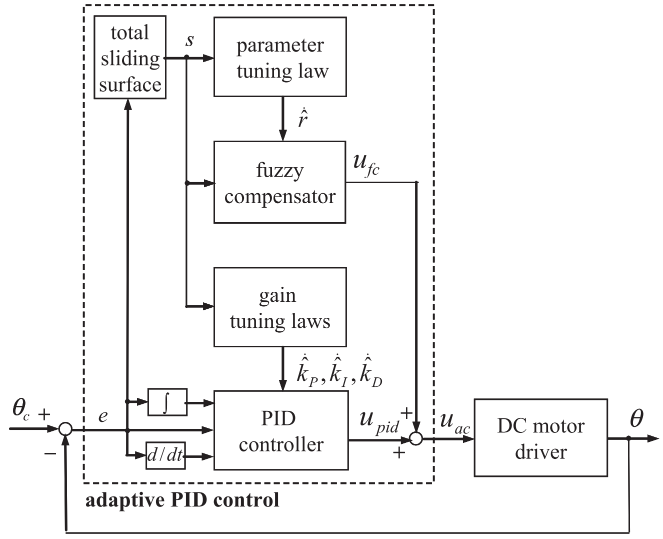

Fig. 1. The block diagram of the APID control system for a DC motor driver.

$$\dot{\hat{k}}_D = -\eta_D \frac{\partial s(t)\dot{s}(t)}{\partial u_{pid}} \frac{\partial u_{pid}}{\partial \hat{k}_D} = \eta_D g s(t) \dot{e}(t) \tag{25}$$

where η_P, η_I and η_D are the learning rates of $\hat{k}_P$, $\hat{k}_I$ and $\hat{k}_D$, respectively. To deal with the unavailable system dynamics, g is rewritten as $|g|\text{sgn}(g)$. Therefore, the update laws of the adaptive PID controller shown in (23)–(25) can be rewritten as follows

$$\dot{\hat{k}}_P = \beta_P \text{sgn}(g) s(t) e(t) = \beta_P s(t) e(t) \tag{26}$$

$$\dot{\hat{k}}_I = \beta_I \text{sgn}(g) s(t) \int_0^t e(\tau)d\tau = \beta_I s(t) \int_0^t e(\tau)d\tau \tag{27}$$

$$\dot{\hat{k}}_D = \beta_D \text{sgn}(g) s(t) \dot{e}(t) = \beta_D s(t) \dot{e}(t) \tag{28}$$

where the terms $\eta_P|g|$, $\eta_I|g|$ and $\eta_D|g|$ are absorbed into the new positive learning rates β_P, β_I and β_D, respectively. These new learning rates are specified by the user. Consequently, only the sign of g is required in the design procedure and it can be easily obtained from the physical characteristic of the controlled system. Here, for the DC motor driver, we have $\text{sgn}(g) = 1$.

3.2. Fuzzy compensator design

Assume the fuzzy compensator has three fuzzy rules in the rule base as given in the following form (Hsu, Cheng, & Lee, 2009; Wai, 2007)

$$\text{Rule 1}: \text{If } s(t) \text{ is PE, then } u_{fc} \text{ is P} \tag{29}$$

$$\text{Rule 2}: \text{If } s(t) \text{ is ZO, then } u_{fc} \text{ is } Z \tag{30}$$

$$\text{Rule 3}: \text{If } s(t) \text{ is NE, then } u_{fc} \text{ is N} \tag{31}$$

where the triangular-typed function and the fuzzy singleton are used to define the membership functions of the fuzzy sets in the IF part and the THEN part in each fuzzy rule as depicted in Fig. 2(a) and (b), respectively. The defuzzification of the output is accomplished by the method of center average

$$u_{fc} = \frac{\sum_{i=1}^{3} r_i w_i}{\sum_{i=1}^{3} w_i} = r_1 w_1 + r_2 w_2 + r_3 w_3 \tag{32}$$

where $0 \leqslant w_1 \leqslant 1$, $0 \leqslant w_2 \leqslant 1$ and $0 \leqslant w_3 \leqslant 1$ are the firing strength of rules 1, 2, and 3, respectively and the relation $w_1 + w_2 + w_3 = 1$ is valid according to the special triangular membership function-based fuzzy system. To reduce computation loading, let $r_1 = \hat{r}$, $r_2 = 0$ and $r_3 = -\hat{r}$. Hence, for any input value s, only one of four conditions will occur according to Fig. 2(a) (Wai, 2007).

Condition 1: Only rule 1 is triggered ($s > s_a$, $w_1 = 1$, $w_2 = w_3 = 0$)

$$u_{fc} = r_1 = \hat{r} \tag{33}$$

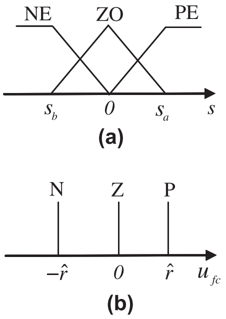

Fig. 2. Membership functions. (a) The input fuzzy sets. (b) The output fuzzy sets.

Condition 2: Rules 1 and 2 are triggered simultaneously $(0 < s \leqslant s_a, 0 < w_1, w_2 \leqslant 1, w_3 = 0)$

$$u_{fc} = r_1 w_1 = \hat{r} w_1 \tag{34}$$

Condition 3: Rules 2 and 3 are triggered simultaneously $(s_b < s \leqslant 0, w_1 = 0, 0 < w_2, w_3 \leqslant 1)$

$$u_{fc} = r_3 w_3 = -\hat{r} w_3 \tag{35}$$

Condition 4: Only rule 3 is triggered $(s \leqslant s_b, w_1 = w_2 = 0, w_3 = 1)$

$$u_{fc} = r_3 = -\hat{r} \tag{36}$$

Then, then Eqs. (33)–(36) can be rewritten as

$$u_{fc} = \hat{r}(w_1 - w_3) \tag{37}$$

Moreover, it can be shown (Wai, 2007) that

$$s(t)(w_1 - w_3) = |s(t)||(w_1 - w_3)| \geqslant 0 \tag{38}$$

3.3. Stability analysis

Assume that the idea controller can be reformulated as

$$u^* = u_{sc}(\hat{k}_P, \hat{k}_I, \hat{k}_D) + \varepsilon(t) \tag{39}$$

where $\varepsilon(t)$ denotes the approximation error and is assumed to be bounded by $0 \leqslant |\varepsilon(t)| \leqslant E$ where E is a positive constant. Then, (21) can be rewritten as

$$\dot{s}(t) = g[\varepsilon(t) - u_{fc}] \tag{40}$$

To guarantee the stability of the APID control system, a Lyapunov function candidate is defined as

$$V_2(s, t) = \frac{1}{2} s^2(t) \tag{41}$$

Differentiating (41) with respect to time and using (40), we can obtain

$$\begin{aligned} \dot{V}_2(s, t) &= s(t)\dot{s}(t) = g s(t)[\varepsilon(t) - u_{fc}] \\ &\leqslant g[|s(t)||\varepsilon(t)| - \hat{r} s(t)(w_1 - w_3)] \\ &= g(|s|(t)|\varepsilon(t)| - \hat{r}|s(t)||w_1 - w_3|) \\ &= -g|s(t)||w_1 - w_3|\left(\hat{r} - \frac{|\varepsilon(t)|}{|w_1 - w_3|}\right) \end{aligned} \tag{42}$$

If the following inequality holds

$$\hat{r} > \frac{|\varepsilon(t)|}{|w_1 - w_3|} \tag{43}$$

then the sliding condition $\dot{V}_2(s, t) \leqslant 0$ can be satisfied. Due to the unknown bound of the approximation error, the value $\hat{r}$ cannot be exactly obtained in advance in practical applications. According to (43), there exists an ideal value r^* defined as

$$r^* = \sup_t \frac{|\varepsilon(t)|}{|w_1 - w_3|} + \kappa \tag{44}$$

where κ is a positive constant so that if $\hat{r} = r^*$, then the sliding condition is satisfied. For further analysis, a simple adaptive algorithm is utilized in this study to estimate the ideal value r^*, and its estimated error is defined as

$$\tilde{r} = r^* - \hat{r} \tag{45}$$

where $\hat{r}$ is the estimated value of the ideal value of r^*. Then, define a new Lyapunov function candidate in the following form

$$V_3(s, \tilde{r}, t) = \frac{1}{2} s^2(t) + \frac{g}{2\eta_r} \tilde{r}^2 \tag{46}$$

where the positive constant η_r is the learning rate for estimating $\hat{r}$. Differentiating (46) with respect to time and using (40) and (44), it can be obtained that

$$\begin{aligned} \dot{V}_3(s, \tilde{r}, t) &= s(t)\dot{s}(t) + \frac{g}{\eta_r}\tilde{r}\dot{\tilde{r}} \\ &= g s(t)\varepsilon(t) - g\hat{r} s(t)(w_1 - w_3) + \frac{g}{\eta_r}\tilde{r}\dot{\tilde{r}} \\ &= g s(t)\varepsilon(t) - g\hat{r} s(t)(w_1 - w_3) \\ &\quad + \frac{g}{\eta_r}\tilde{r}\dot{\tilde{r}} + g r^* s(t)(w_1 - w_3) - g r^* s(t)(w_1 - w_3) \\ &= g s(t)\varepsilon(t) + g\tilde{r} s(t)(w_1 - w_3) + \frac{g}{\eta_r}\tilde{r}\dot{\tilde{r}} - g r^* s(t)(w_1 - w_3) \\ &\leqslant g|s(t)||\varepsilon(t)| + \tilde{r} g\left[s(t)(w_1 - w_3) + \frac{\dot{\tilde{r}}}{\eta_r}\right] - g r^* s(t)(w_1 - w_3) \end{aligned} \tag{47}$$

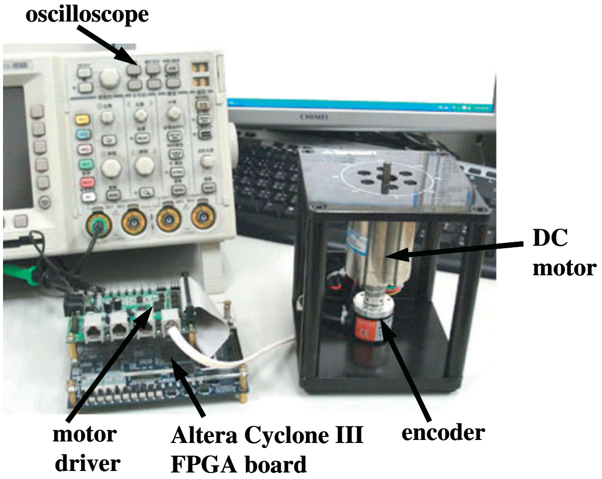

Fig. 3. Experimental setup.

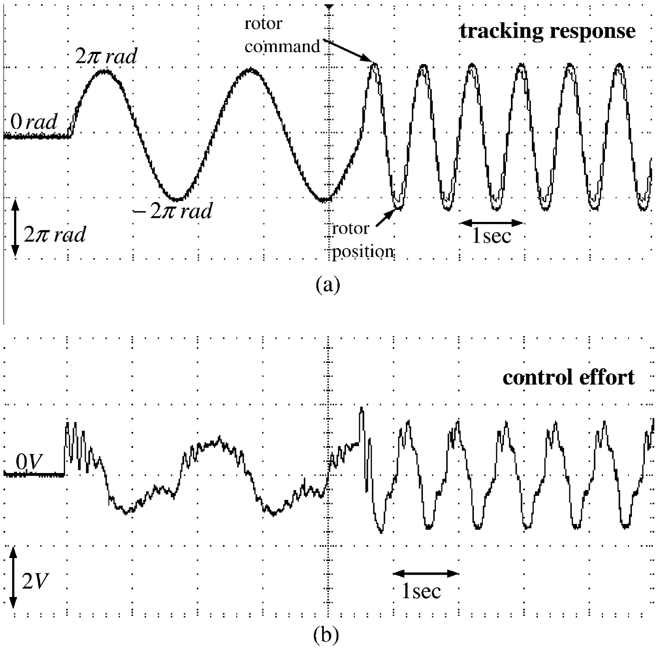

Fig. 4. Experimental results of the fuzzy sliding-mode control.

Choose the parameter tuning law as

$$\dot{\hat{r}} = -\dot{\tilde{r}} = -\eta_r s(t)(w_1 - w_3) \tag{48}$$

and using (44), (47) becomes

$$\begin{aligned}\dot{V}_3(s,\tilde{r},t) &\leqslant g|s(t)||\varepsilon(t)| - gr^* s(t)(w_1 - w_3)\\ &= g|s(t)||\varepsilon(t)| - gr^*|s(t)||w_1 - w_3|\\ &= -g\kappa|s(t)||w_1 - w_3| \leqslant 0\end{aligned} \tag{49}$$

Since $\dot{V}_3(s,\tilde{r},t)$ is negative semidefinite, that is $V_3(s,\tilde{r},t) \leqslant V_3(s,\tilde{r},0)$, this implies that $s(t)$ and $\tilde{r}$ are bounded. Define a function $\Omega(t)$ as $\Omega(t) \equiv g\kappa|s(t)||w_1 - w_3|$. Integrating $\Omega(t)$ with respect to time, we can then obtain

$$\int_0^t \Omega(\tau)d\tau \leqslant V_3(s,\tilde{r},0) - V_3(s,\tilde{r},t) \tag{50}$$

Because $V_3(s,\tilde{r},0)$ is bounded and $V_3(s,\tilde{r},t)$ is nonincreasing as well as bounded, the following result can be obtained

$$\lim_{t\to\infty} \int_0^t \Omega(\tau)d\tau < \infty \tag{51}$$

Moreover, since $\dot{\Omega}(t)$ is bounded, by Barbalat's Lemma (Slotine & Li, 1991), we have $\lim_{t\to\infty}\Omega(t) = 0$, which implies that $s \to 0$ as $t \to \infty$. As a result, the stability of the proposed APID control system can be guaranteed.

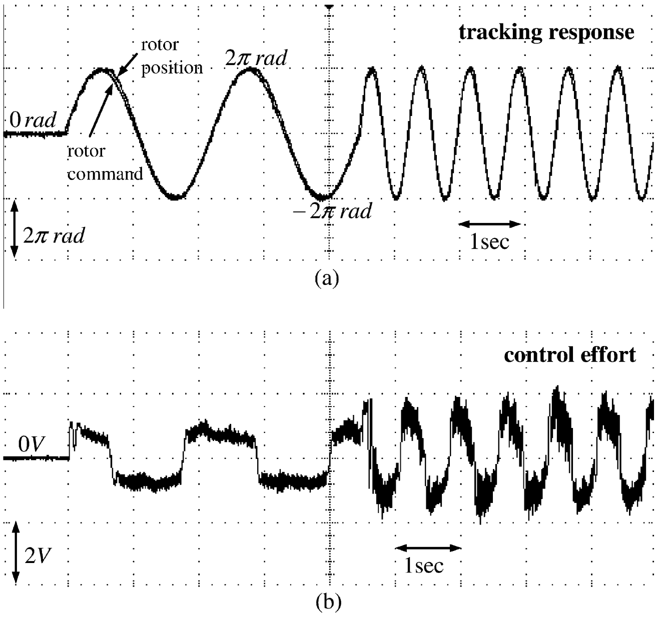

Fig. 5. Experimental results of the adaptive robust PID control.

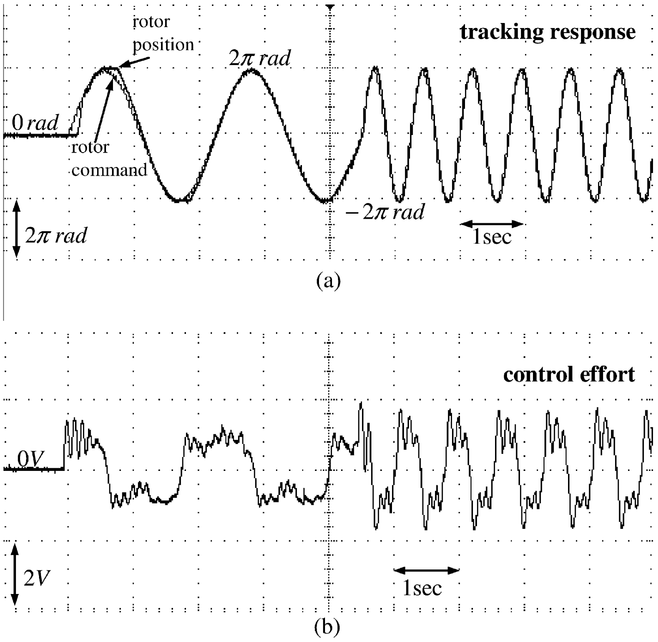

Fig. 7. Experimental results of the APID control.

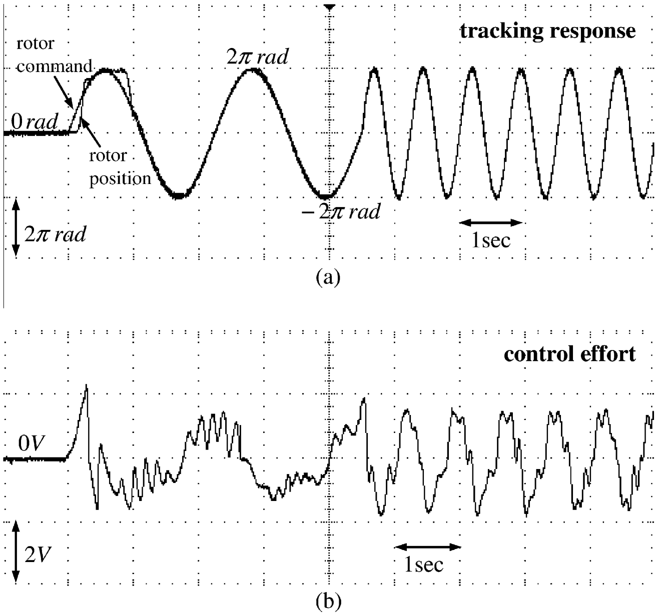

Fig. 6. Experimental results of the supervisory recurrent fuzzy network control.

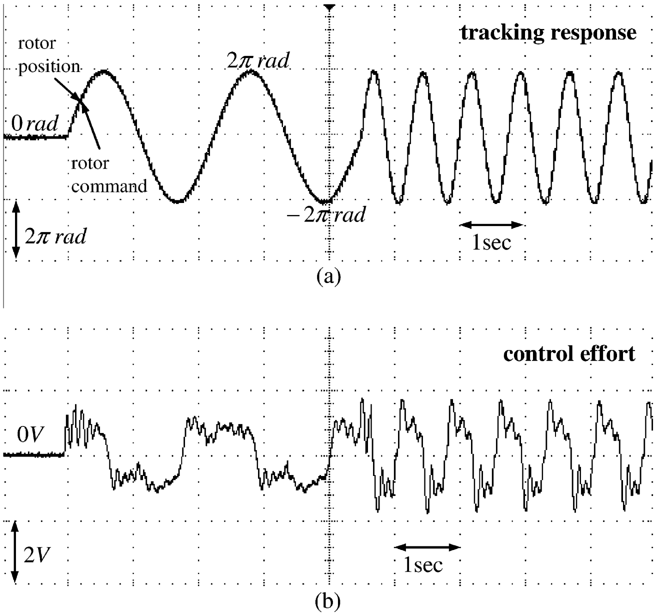

Fig. 8. Experimental results of the trained APID control.

Table 1
Comparison of controller characteristics.

Controller	Controller parameters	Stability proof	Robustness	Chattering phenomena	Computation loading
Fuzzy sliding-mode control	Trial and error	No	Middle	No	Light
Adaptive robust PID control	On-line learning	Yes	Excellent	Yes	Light
Supervisory recurrent fuzzy network control	On-line learning	Yes	Excellent	No	Heavy
APID control	On-line learning	Yes	Excellent	No	Light

4. Experimental setup and 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. The advantages of implementing a controller based on FPGA include shorter development cycles, lower cost, small size, fast system execute speed and high flexibility (http://www.altera.com/). Altera offers a variety of development tools to accelerate the embedded design process. The SOPC Builder is a system development tool for creating systems including processors, peripherals and memories. It eliminates manual system integration tasks and allows users to focus on custom logic design to develop differential system features. The Quartus® II software is the development tool for programmable logic devices and the Nios® II Integrated Development Environment including a complete set of C/C++ software development tools is a collection of software generation, management, and deployment tools for the Nios II processor (http://www.altera.com/). The proposed APID control system is applied to a DC motor driver to show its effectiveness by using the Altera Cyclone III 3C16 FPGA device, the Altera Quartus II software, the Nios II Integrated Development Environment and the VHDL hardware description language to implement the hardware of the control system. The experimental setup is shown in Fig. 3. To investigate the effectiveness of the proposed APID control system, a comparison among the fuzzy sliding-mode control (Choi et al., 1999), the adaptive robust PID control (Chang & Yan, 2005), the supervisory recurrent fuzzy network control (Lin & Hsu, 2004) and the proposed APID control is made.

4.1. Comparison methods

First, the fuzzy sliding-mode controller (Choi et al., 1999) is applied to the DC motor driver. The fuzzy control rules are given in the following form

$$\text{Rule } i : \text{IF } s(t) \text{ is } F_s^i, \text{THEN } u \text{ is } \rho_i \tag{52}$$

where, in the ith rule, ρ_i is the singleton control action and F_s^i is the fuzzy set of $s(t)$. The fuzzy rules are constructed so as to make $s(t)$ approach to zero with fast rise time and without large overshoot. The fuzzy rules should be pre-constructed by a time-consuming trial-and-error tuning procedure to achieve the desired performance. The defuzzification of the controller output is accomplished by the method of sum of weightings. The experimental results of the fuzzy sliding-mode control system are shown in Fig. 4 where the tracking response and the associated control effort are shown in Fig. 4(a) and (b), respectively. The experimental results show that a favorable tracking performance can be achieved for a low-frequency sinusoidal position command. However, the tracking performance will be gradually deteriorated if the frequency of the position command is increased.

Then, the adaptive robust PID controller (Chang & Yan, 2005) is applied to the DC motor driver again. To ensure system stability, a switching compensator which requires a bound of the approximation error is used. If a not sufficiently large bound is chosen, then system stability cannot be guaranteed. On the contrary, if an overly large bound is applied, the control effort will cause chattering phenomena to wear the bearing mechanism. The experimental results of the adaptive robust PID controller with an upper bound of the approximation error being set as $E = 0.25$ are shown in Fig. 5 where the tracking response is illustrated in Fig. 5(a) and the associated control effort is in Fig. 5(b). The robust tracking performance of the adaptive robust PID controller can be achieved even under frequency change of the position command. Though a favorable tracking response can be obtained, the chattering phenomenon of the associated control effort as shown in Fig. 5(b) is undesirable. A trade-off problem between chattering and control accuracy arises in the adaptive robust PID control scheme.

To relax the requirement of a bound of the approximation error, the supervisory recurrent fuzzy network control (Lin & Hsu, 2004) is applied to the DC motor driver again. A bound estimation mechanism is incorporated to estimate the bound of the approximation error, and thus the chattering phenomenon of the control effort can be abated. The experimental results of the supervisory recurrent fuzzy network control are shown in Fig. 6. The tracking response is shown in Fig. 6(a) and the associated control effort is in Fig. 6(b). Though a favorable tracking response can be achieved after learning of the controller parameters even under frequency change of the position command, it suffers from heavy computation loading and the hardware is difficult to implement.

4.2. APID control method

The proposed APID control is now applied to the DC motor driver. It should be emphasized that the development of the APID control system does not need to know the system dynamics of the DC motor driver. For practical implementation, the parameters of the APID control system can be online tuned by the proposed adaptive laws. The control parameters of the proposed APID control are selected as $k_1 = 10$, $k_2 = 25$, $\beta_P = 10$, $\beta_I = \beta_D = 0.1$ and $\eta_r = 1$. All the gains are chosen to satisfy the required conditions on stability. The experimental results of the proposed APID control system are shown in Fig. 7. The tracking response is shown in Fig. 7(a) and the associated control effort is shown in Fig. 7(b). The experimental results show the proposed APID control with fuzzy compensator not only can achieve a favorable tracking performance but also does not cause the chattering phenomena in the associated control efforts. Further, the trained APID control system is applied to the DC motor driver again. The experimental results of the trained APID control are shown in Fig. 8. The tracking response is shown in Fig. 8(a) and the associated control effort is shown in Fig. 8(b). The experimental results show the tracking performance of the APID control system is further improved when the initial values are trained and it can achieve a more favorable robust characteristic for the command frequency variation.

5. Conclusions

This paper has successfully implemented a real-time adaptive PID (APID) control system via the sliding-mode control approach, where a PID controller used to mimic an ideal controller is the main controller and a fuzzy compensator is designed to eliminate the effect of the approximation error introduced by the PID controller upon the system stability in the Lyapunov sense. An online parameter training algorithm based on the gradient descent method is proposed to increase the learning capability and to attain the Lyapunov stability. The proposed APID control system is applied to a DC motor driver. Additionally, the comparison of controller characteristics among the fuzzy sliding-mode control (Choi et al., 1999), the adaptive robust PID control (Chang & Yan, 2005), the supervisory recurrent fuzzy network control (Lin & Hsu, 2004) and the proposed APID control is made in Table 1. It shows that the proposed APIDC system is more suitable for the rotor position control of the DC motor drive system.

Acknowledgments

The authors appreciate the partial financial support from the National Science Council of Republic of China under the grant NSC 98-2221-E-216-040.

References

Chang, W. D., Hwang, R. C., & Hsieh, J. G. (2002). A self-tuning PID control for a class of nonlinear systems based on the Lyapunov approach. *Journal of Process Control, 12*(2), 233–242.

Chang, W. D., & Yan, J. J. (2005). Adaptive robust PID controller design based on a sliding mode for uncertain chaotic systems. *Chaos, Solitons and Fractals, 26*(1), 167–175.

Choi, B. J., Kwak, S. W., & Kim, B. K. (1999). Design of a single-input fuzzy logic controller and its properties. *Fuzzy Sets and Systems, 106*(3), 299–308.

Coelho, L. S., & Bernert, D. L. A. (2009). PID control design for chaotic synchronization using a tribes optimization approach. *Chaos, Solitons and Fractals, 42*(1), 634–640.

Elmas, C., Ustun, O., & Sayan, H. H. (2008). A neuro-fuzzy controller for speed control of a permanent magnet synchronous motor drive. *Expert Systems with Applications, 34*(1), 657–664.

Hsu, C. F., Cheng, K. H., & Lee, T. T. (2009). Robust wavelet-based adaptive neural controller design with a fuzzy compensator. *Neurocomputing, 73*(1), 423–431.

Juang, J. G., Huang, M. T., & Liu, W. K. (2008). PID control using presearched genetic algorithms for a MIMO system. *IEEE Transactions on Systems, Man, Cybernetics, Part C, 38*(5), 716–727.

Kung, Y. S., & Tsai, M. H. (2007). FPGA-based speed control IC for PMSM drive with adaptive fuzzy control. *IEEE Transactions on Power Electronics, 22*(6), 2476–2486.

Lin, C. M., & Hsu, C. F. (2004). Supervisory recurrent fuzzy neural network control of wing rock for slender delta wings. *IEEE Transactions on Fuzzy Systems, 12*(5), 733–742.

Lin, F. J., Teng, L. T., Chen, C. Y., & Hung, Y. C. (2008). FPGA-based adaptive backstepping control system using RBFN for linear induction motor drive. *IET Electric Power Applications, 2*(6), 325–340.

Pisano, A., Davila, A., Fridman, L., & Usai, E. (2008). Cascade control of PM DC drives via second-order sliding-mode technique. *IEEE Transactions on Industrial Electronics, 55*(11), 3846–3854.

Ren, T. J., Chen, T. C., & Chen, C. J. (2008). Motion control for a two-wheeled vehicle using a self-tuning PID controller. *Control Engineering Practice, 16*(3), 365–375.

Santos, M., & Dexter, A. L. (2002). Control of a cryogenic process using a fuzzy PID scheduler. *Control Engineering Practice, 10*(10), 1147–1152.

Shao, X., & Sun, D. (2007). Development of a new robot controller architecture with FPGA-based IC design for improved high-speed performance. *IEEE Transactions on Industrial Informatics, 3*(4), 312–321.

Slotine, J. E., & Li, W. (1991). *Applied nonlinear control*. Englewood Cliffs, New Jersey: Prentice-Hall.

Wai, R. J. (2007). Fuzzy sliding-mode control using adaptive tuning technique. *IEEE Transactions on Industrial Electronics, 54*(1), 586–594.

Yu, D. L., Chang, T. K., & Yu, D. W. (2007). A stable self-learning PID control for multivariable time varying systems. *Control Engineering Practice, 15*(12), 1577–1587.