

유도전동기 속도제어를 위한 개선된 신경회로망 기반 자기동조 퍼지 PID 제어기 설계

論 文

51B-12-6

Improved Neural Network-Based Self-Tuning Fuzzy PID Controller for Induction Motor Speed Control

金相敏* · 韓宇勇** · 李昌求***

(Sang-min Kim · Woo-yong Han · Chang-goo Lee)

Abstract - This paper presents a neural network based self-tuning fuzzy PID control scheme with variable learning rate for induction motor speed control. When induction motor is continuously used long time, its electrical and mechanical parameters will change, which degrade the performance of PID controller considerably. This paper re-analyzes the fuzzy controller as conventional PID controller structure, introduces a single neuron with a back-propagation learning algorithm to tune the control parameters, and proposes a variable learning rate to improve the control performance. Proposed scheme is simple in structure and computational burden is small. The simulation using Matlab/Simulink and the experiment using dSPACE(DS1102) board are performed to verify the effectiveness of the proposed scheme.

Key Words : Fuzzy PID, Self-Tuning, Matlab/Simulink, dSPACE(DS1102), Induction motor

1. 서 론

AC 전동기 벡터 제어를 위해 사용하는 PID 속도제어기는 구조가 단순하고 견실하며 현장 운전자에게 익숙하다는 장점을 가지고 있어서 산업현장에 널리 사용되고 있다[1-2]. 그러나 AC 전동기 제어시스템은 비선형성, 전동기 파라미터들의 시변 특성에 의하여 제어성능이 저하되기 때문에 이를 보상하기 위하여 제어계수들의 재동조가 필요하다. 이러한 문제를 해결하기 위하여 제어계수들을 온라인 동조하는 파라미터 동조기법이나 적응제어 알고리즘들이 많이 제안되어 왔다. 최근 일부 논문에서는 퍼지 PID 제어나 뉴로-퍼지 제어 등 신경회로망과 퍼지를 이용한 제어기법들이 연구되고 있다[3-4]. 이중에서도 퍼지를 이용한 계수 동조는 비선형 제어와 온라인 동조에 효과적이지만 퍼지 제어규칙의 생성이 매우 시간 소모적이고 실제 적용이 어렵다. look-up table 방식으로 양자화하여 설계할 경우 메모리를 많이 차지하고, 적응 퍼지 시스템을 이용하여 퍼지논리를 수식적으로 구현하게 되면 적응성은 뛰어나지만 연산량이 너무 많다는 단점이 있다[5]. 한편 신경회로망을 이용한 계수 동조에서는 역전파 알고리즘이 가장 널리 사용되고 있는데, 역전파 알고리즘의 특성상 학습계수가 증가하면 수렴속도가 향상되는 반면에 외란에 대한 강인성이 저하되어 시스템이 쉽게 불안정해질 수 있다는 단점이 있다[6-8].

본 논문에서는 퍼지의 선형성을 이용하여 PI-PD형 퍼지 PID 제어기의 퍼지 연산부를 간략화 시키고 일반 PID 제어

기와 유사한 제어기 출력에 가장 큰 영향을 미치는 출력측 스캘링 계수를 단일 신경 회로망 구조로 변형한 신경 회로망 기반 자기동조 퍼지 PID 제어기(SF-PID)를 제시하였다. SF PID 제어기는 구조가 단순하고 계산량이 적어 실시간 제어가 용이하고, PID 제어기 구조를 유지함으로써 model-free 동조기법 등에 의해 결정된 PID 계수를 기반으로 초기값을 설정할 수 있으며 초기 제어구간에서의 안정성을 보장할 수 있다. 또한 단일 신경 회로망 구조를 이용함으로써 신경회로망의 초기 연결강도와 계산량에 대한 문제점을 해결하고 오차의 부호 정보에 따라 학습계수를 변화시키는 가변 학습계수 역전파 알고리즘을 사용하여 외란에 강인하면서도 빠른 수렴 속도를 갖도록 하였다. 유도전동기 동특성 변화에 대한 제안한 제어기의 성능을 확인하기 위하여, 먼저 PC 상에서 Matlab/Simulink를 이용한 시뮬레이션을 수행하고 dSPACE(이하 DS1102) 보드를 이용한 실험시스템을 구성하여 유도전동기 정역 속도제어 실험을 수행하였다.

2. 유도전동기 동적 상태 방정식

3상 유도전동기의 동적 상태방정식을 고정자 기준 dq 고정 좌표계에서 표현하면 식 (1)과 같다[9].

$$\begin{aligned} p i_s &= -a_1 i_s + a_2 \phi_r - a_3 n_p K \omega_r \phi_r + a_0 V_s \\ p \phi_r &= -a_4 \phi_r + n_p K \omega_r \phi_r + a_5 i_s \\ p \omega_r &= -a_6 \omega_r + a_7 (T_e - T_L) \end{aligned} \quad (1)$$

여기서

$$\begin{aligned} i_s &= [i_{ds}, i_{qs}]^T: \text{고정자 전류}, \phi_r = [\phi_{dr}, \phi_{qr}]^T: \text{회전자 자속} \\ V_s &= [V_{ds}, V_{qs}]^T: \text{고정자 전압}, \omega_r: \text{회전자 각속도}, \\ p &: \text{미분 연산자}, n_p: \text{폴수}, a_0 = L_r / (L_s L_r - L_m^2), \\ a_1 &= a_0 (R_s + R_r L_m^2 / L_r^2) \end{aligned}$$

* 正 會 員 : 全 北 大 工 大 電 子 情 報 工 學 部 博 士 課 程

** 正 會 員 : 全 州 工 業 大 電 氣 科 教 授 · 工 博

*** 正 會 員 : 全 北 大 工 大 電 子 情 報 工 學 部 教 授

接 受 日 字 : 2002年 4月 21日

最 終 完 了 : 2002年 11月 4日

$$a_2 = a_0 L_m / (L_r T_r), \quad a_3 = a_0 L_m / L_r, \quad a_4 = 1 / T_r,$$

$$a_5 = L_m / T_r, \quad a_6 = B / J, \quad a_7 = 1 / J$$

$$K = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \quad T_r = L_r / R_r; \text{ 회전자 시정수}$$

R_s (R_r): 고정자(회전자) 저항

L_s (L_r): 고정자(회전자) 인덕턴스

L_m : 상호인덕턴스, B : 마찰계수, J : 관성모멘트

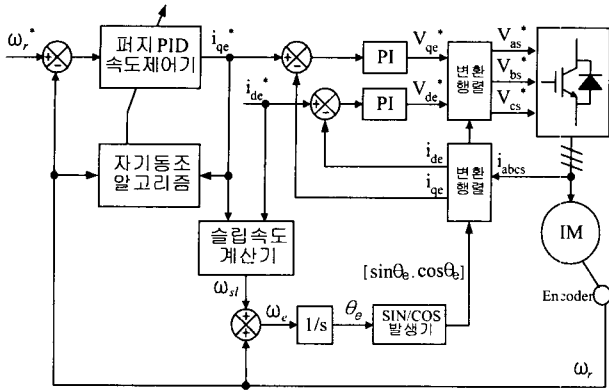


그림 1 간접벡터 유도전동기 구동 시스템

Fig. 1 Indirect Field Oriented IM drives

토크 방정식은 식 (2)와 같이 표현할 수 있다.

$$T_e = \frac{3}{2} n_p \frac{L_m}{L_r} (\phi_{dr}^s i_{qs}^s - \phi_{qr}^s i_{ds}^s) \quad (2)$$

그림 1은 전압원 인버터로 구동되는 유도전동기의 간접벡터 제어 시스템을 나타낸다. 회전자 각속도 ω_r 와 dq축 전류 명령 i_{qs}^* , i_{ds}^* 은 변환행렬(고정좌표계 \leftrightarrow 동기좌표계)을 만드는데 사용된다. 본 논문에서는 속도제어를 위하여 자기동조 퍼지 PID 제어기를 이용한다.

3. 신경회로망 기반 자기동조 퍼지 PID 제어기

3.1 선형 퍼지 제어기

퍼지 제어기는 일반적으로 비선형 특성을 지닌다. 그러나 특별한 형태의 소속 함수와 추론방법 그리고 제어규칙 등을 이용하여 선형적인 특성을 갖게 할 수 있다[10-11]. 임의의 플랜트에 적용한 두 개의 입력과 한 개의 출력을 갖는 일반적인 퍼지 제어기의 구조는 그림 2와 같다.

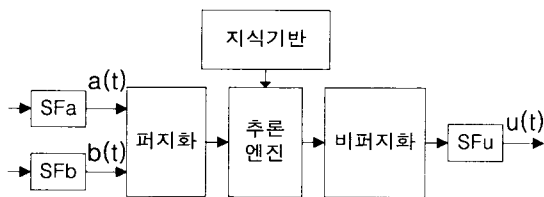


그림 2 일반적인 퍼지 제어기 구조

Fig. 2 The structure of the general fuzzy controller

제어 대상이 주어지면 퍼지 제어기의 출력 변수를 결정하고, 제어방법에 따라 필요한 입력변수의 종류를 결정한다. 입력신호의 대집합을 정규화하고 정규화된 두 입력신호를 각각 $a(t)$, $b(t)$ 라고 할 때 J. H. Im가 제안한 방법을 이용하여 선형 퍼지 제어기를 설계하면 제어기의 출력은 식 (3)과 같이 표현되고 제어기 입출력 변수 사이에는 식 (4)와 같은 관계가 성립하므로 퍼지 제어기의 출력은 언제나 두 입력의 단순합이 된다[12].

$$u = \left(\frac{a_{i+1}-a}{a_{i+1}-a_i} \frac{b_{j+1}-b}{b_{j+1}-b_j} \right) u_{i,j} + \left(\frac{a-a_i}{a_{i+1}-a_i} \frac{b_{j+1}-b}{b_{j+1}-b_j} \right) u_{(i+1),j} + \left(\frac{a_{i+1}-a}{a_{i+1}-a_i} \frac{b-b_j}{b_{j+1}-b_j} \right) u_{i,(j+1)} + \left(\frac{a-a_i}{a_{i+1}-a_i} \frac{b-b_j}{b_{j+1}-b_j} \right) u_{(i+1),(j+1)} \quad (3)$$

여 기 서

$$a_{i+1} = a_i + \frac{1}{N}, \quad b_{j+1} = b_j + \frac{1}{N}, \quad u_{i+1,j} = u_{i,j+1} = u_{i,j} + \frac{1}{N}, \quad u_{i+1,j+1} = u_{i,j} + \frac{2}{N} \quad (4)$$

3.2 퍼지 PID 제어기 설계와 초기 동조

앞 절에서 기술한 방식에 따라 퍼지 제어기를 설계하면 그림 3과 같은 형태의 퍼지 PID 제어기를 얻을 수 있다. Sat(Saturation)은 각각의 입력을 최대값 1, 최소값 -1로 제한하기 위하여 사용되었다.

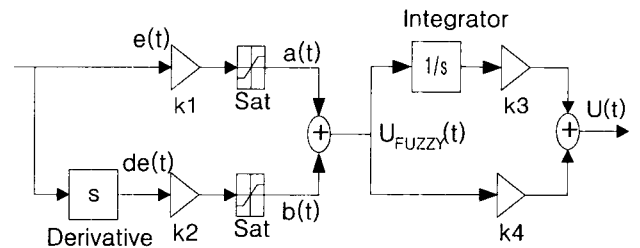


그림 3 간략화된 퍼지 PID 제어기

Fig. 3 Simplified fuzzy PID controller

이산시간 제어에서 퍼지 제어기의 출력 $U_{FUZZY}(z)$ 와 플랜트에 인가되는 입력 $U(z)$ 는 식 (5)와 같다.

$$U_{FUZZY}(z) = \left(k_1 + \frac{k_2}{T} \frac{z-1}{z} \right) e(z) \quad (5)$$

$$U(z) = \left[k_3 \frac{Tz}{z-1} + k_4 \right] U_{FUZZY}(z)$$

여기서 T 는 샘플링 주기, $e(z)$ 는 기준입력과 플랜트 출력사이의 오차, $k_1 \sim k_4$ 는 비례이득이다.

이산시간 제어에서 일반 PID 제어기의 출력은 식 (6)과 같이 표현된다.

$$u_{PID}(z) = \left[K_p + K_d \frac{z-1}{Tz} + K_i \frac{Tz}{z-1} \right] e(z) \quad (6)$$

여기서 K_p, K_i, K_d 는 각각 비례, 적분, 미분이득이다.

일반 PID 제어기 계수와 퍼지 PID 제어기 계수사이에 식 (7)의 관계가 성립할 때 일반 PID 제어기와 퍼지 PID 제어기는 동일한 성능을 갖는다.

$$K_p = k_1 k_4 + k_2 k_3, K_d = k_2 k_4, K_i = k_1 k_3 \quad (7)$$

일반 PID 제어기와 퍼지 PID 제어기의 부분적인 동일성으로부터 일반 PID 제어기 계수 초기 동조법을 기반으로 하여 퍼지 PID 제어기 계수값을 설정할 수 있다[13].

3.3 퍼지 PID 제어기의 온라인 동조

그림 3과 같은 구조의 퍼지 PID 제어기는 퍼지 추론이나 퍼지 연산을 하지는 않지만 특정한 형태의 소속 함수와 추론방법, 그리고 제어규칙을 사용한 퍼지 제어기와 동일한 특성을 가짐을 보였다. 그러므로 기존의 퍼지 제어기법을 동일하게 적용할 수 있다. Ishida는 일반 PID 제어기와 비슷한 구조를 갖는 신경회로망 기반 제어기를 구성하고 비선형 시스템에 적용하여 좋은 결과를 얻었다[6]. 이를 바탕으로 퍼지형 PID 제어기의 출력계수를 그림 4와 같이 단일 신경회로망 구조로 변형하였다. 그림 4에 대한 이산화된 수식은 식 (8)과 같다.

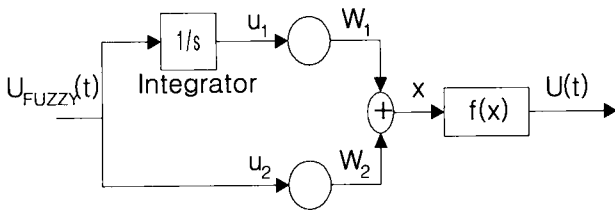


그림 4 신경회로망 구조의 퍼지 PID 제어기 출력부
Fig. 4 Output part of fuzzy PID controller

$$\begin{aligned} u_1(kT) &= u_1((k-1)T) + U_{FUZZY}(kT) T \\ u_2(kT) &= U_{FUZZY}(kT) \\ x(kT) &= u_1(kT) W_1(kT) + u_2(kT) W_2(kT) \\ U(kT) &= f(u_1(kT), u_2(kT), W_1(kT), W_2(kT)) \end{aligned} \quad (8)$$

본 논문에서 $U(kT)$ 는 q축 전류 명령 i_{qs}^* 이다. W_1, W_2 는 신경회로망의 연결 강도이며 그림 4의 퍼지 PID 제어기 출력 계수 k_3, k_4 와 같은 의미를 갖는다. $f(\cdot)$ 는 활성화 함수로서 본 논문에서는 $f(x)=x$ 의 선형 함수를 사용하였으며 성능 지수 E 를 식 (9)와 같이 정의하고 성능 지수 E 가 최소화되도록 역전파 알고리즘을 적용하였다.

$$E = \frac{1}{2} \{s_1 e(kT)^2 + s_2 u(kT)^2\} \quad (9)$$

여기서 $e(kT)=r(kT)-y(kT)$, s_1 과 s_2 는 scaling factor이다.

$$\begin{aligned} W_1(kT) &= W_1((k-1)T) - \eta \frac{\partial E}{\partial W_1} \Big|_{kT} \\ W_2(kT) &= W_2((k-1)T) - \eta \frac{\partial E}{\partial W_2} \Big|_{kT} \end{aligned} \quad (10)$$

본 논문에서 $r(kT)$ 와 $y(kT)$ 는 각각 기준 속도 ω_r^* 와 전동기의 회전자 속도 ω_r 이며 η 는 학습계수이다. 연쇄 법칙을 이용하여 식을 전개하고 다시 정리하면 식 (11)을 얻을 수 있다.

$$\begin{aligned} W_1(kT) &= W_1((k-1)T) - \eta_2 u_1(kT) + \eta_1 e(kT) u_1 \frac{\partial y}{\partial u} \Big|_{kT} \\ W_2(kT) &= W_2((k-1)T) - \eta_2 u_2(kT) + \eta_1 e(kT) u_2 \frac{\partial y}{\partial u} \Big|_{kT} \end{aligned} \quad (11)$$

여기서 $\eta_1 = \eta \cdot s_1, \eta_2 = \eta \cdot s_2$ 이고 η 는 학습계수이다.

학습계수 η 는 일반적으로 상수를 사용하며 수렴속도를 결정하는 중요한 요소가 된다. 학습계수를 크게 하면 수렴속도는 향상되지만 외란 등에 민감하게 되고 쉽게 불안정해지기 쉽다. 반면에 학습계수가 작으면 수렴속도가 현저히 저하된다. 이러한 단점을 보완하기 위하여 학습계수를 시스템의 오차 정보에 따라 식 (12)와 같이 가변하도록 하였다.

$$\begin{aligned} \eta &= \frac{\eta_{\max} - \eta_{\min}}{2} \left[\frac{1 - e^{-(\gamma - \epsilon/2)\delta}}{1 + e^{-(\gamma - \epsilon/2)\delta}} + 1 \right] + \eta_{\min} \\ \delta &= \frac{2R}{\epsilon} \end{aligned} \quad (12)$$

γ 는 시스템 오차와 관련된 정보로써 초기값은 0이며 현재의 오차부호와 한 스텝 이전 오차부호를 비교하여 부호가 같으면 1씩 증가하고 부호가 다를 경우 초기값으로 초기화된다. ϵ 은 사용자가 설정하는 값이며 $\gamma = \epsilon/2$ 일 때 $\eta = (\eta_{\max} - \eta_{\min})/2$ 가 된다. δ 는 $\gamma = \epsilon/2$ 에서의 기울기를 결정한다.

4. 시뮬레이션 결과

앞 절에서 제시한 SF-PID 제어기의 성능을 평가하기 위하여 Matlab/Simulink 상에서 벡터제어 유도전동기의 시뮬레이션과 실험을 수행하였다. 실험에 사용한 3상 유도전동기 모델은 표 1에 제시한 사양을 갖는다.

표 1 유도전동기의 파라미터

Table 1 Induction motor parameters

파라미터	공칭 값
정격출력, P_{rated}	400 [W]
정격전류, I_{rated}	3.5 [A]
정격전압, V_{rated}	120 [V]
폴수, n_p	1
고정자저항, R_s	3.68 [Ω]
회전자저항, R_r	2.4 [Ω]
고정자인덕턴스, L_s	470.6 [mH]
회전자인덕턴스, L_r	470.6 [mH]
자화인덕턴스, L_m	441.8 [mH]
관성모멘트, J	0.007257 [kg·m ²]

그림 5는 운전자가 알고 있는 관성모멘트 값이 실제값의 0.25배이고, 100[rad/s]→-100[rad/s]로 정역운전할 때의 회전자 속도 및 dq축 전류 파형을 나타낸다. 그림 5(a)는 일반적인 PID 제어를 사용한 경우로써 약 30%의 오버슈트가 발생하는 것을 알 수 있다. 그림 5(b)는 SF-PID 제어를 사용한 경우로써 초기에 오버슈트가 발생하지만 시간에 지남에 따라 학습효과에 의하여 오버슈트가 없어지고 정상상태오차도 0으로 됨을 알 수 있다. dq축 전류 파형을 살펴보면 그림 5(a)의 경우에 토크 성분 전류 i_{qe} 가 느리게 변하고 그에 따른 자속성분 전류 i_{de} 값에 대한 영향이 지속됨으로써 d축과 q축 전류의 커플링(coupling) 현상이 사라지지 않아 정상상태에서도 벡터제어가 적절히 실현되지 않고 있다. 그림 5(b)의 경우는 학습효과에 의하여 i_{qe} 가 빠르게 수렴하게 되어 i_{de} 가 대략 0.3[s]의 과도상태를 지나면 커플링 현상이 사라지는 것을 알 수 있다.

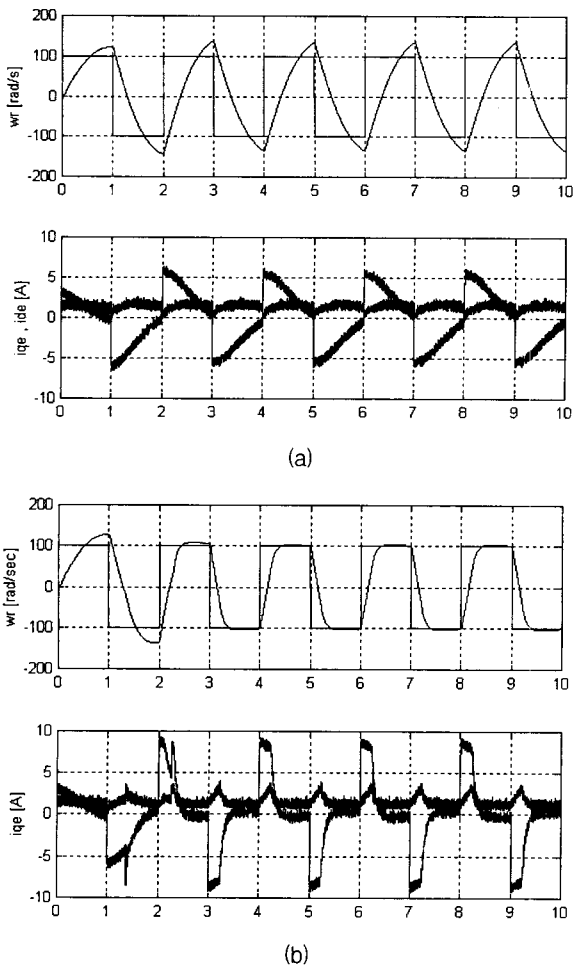


그림 5 기준속도 $\omega_r = \pm 100$ rad/s, 제어기 설계에 사용한 관성모멘트 $J = 0.25 \times J_n$ 일 때 (a) 일반 PID 제어를 이용한 시뮬레이션 결과 (b) SF-PID 제어를 이용한 시뮬레이션 결과

Fig. 5 When the reference speed $\omega_r = \pm 100$ rad/s, the moment of inertia used in controller design, $J = 0.25 \times J_n$, the speed and current responses of (a) general PID controller (b) SF-PID controller

5. 실험 결과

제한한 알고리즘의 타당성을 검토하기 위하여 그림 6과 같은 실험 장치를 구성하였다. 사용한 유도전동기는 Nikki Denso의 NA20-15F이고 사양은 표 1과 같다. 전압형 인버터에 공간벡터 변조기법(이하 SVPWM)을 적용하였다. 고정자 전류는 CT(Current Transducer)를 이용하여 측정되고 측정된 아날로그 신호는 12bit 해상도를 갖는 A/D 컨버터를 통하여 이산화된다. 회전자 속도를 측정하기 위하여 2000[ppr] 엔코더를 사용하였다. 알고리즘은 dSPACE사의 DS1102 보드를 이용하여 구현하였다. DS1102 보드는 주프로세서로 TMS320C31 (제어 및 추정 알고리즘의 연산 수행), 부프로세서로 TMS320P14 (PWM 신호 발생)를 사용한다. 전류제어와 SVPWM의 샘플링 주기는 0.2[ms]로 하였고 속도 제어의 샘플링 주기는 2[ms]로 하였다.

추정 및 제어 알고리즘은 Matlab/Simulink 상에서 작성하고, 참고문헌 [14]에서 제시한 과정을 통하여 실험시스템에 적용하였다. 실험결과는 Control Desk(인터페이스 프로그램) 상에서 모니터링하고 실시간으로 데이터를 저장하였다.

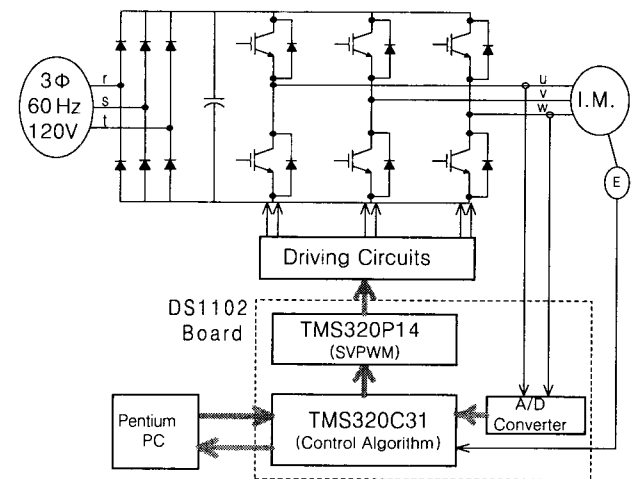


그림 6 실험장치 구성
Fig. 6 Experimental setup

그림 7은 운전자가 알고 있는 관성모멘트 값이 실제값의 0.25배이고 기준속도를 ± 100 [rad/s]로 설정하여 정역운전 하였을 때의 실험결과를 나타낸다. 기존 PID 제어를 이용하였을 경우에는 30%의 오버슈트가 발생하는 것을 알 수 있다. 제안한 방법을 이용한 경우에는 대략 5[s]의 시간이 지나면 오버슈트가 사라지고 정상상태 오차도 0으로 접근하는 것을 알 수 있다.

그림 8은 그림 7과 동일한 환경하에서 기준속도를 ± 10 [rad/s]로 설정하여 저속으로 정역운전 하였을 때의 실험 결과를 나타낸다. 기존 PID 제어를 이용하였을 경우에는 토크 성분 전류가 작아 오버슈트는 발생하지 않지만 상승시간이 0.8[s]인 것을 알 수 있다. 제안한 방법을 이용한 경우에는 대략 5[s]가 지난 후에 상승시간이 0.4[s]로 줄어들고 6[s]가 지난 후에 정상상태 오차가 0으로 접근하는 것을 알 수 있다.

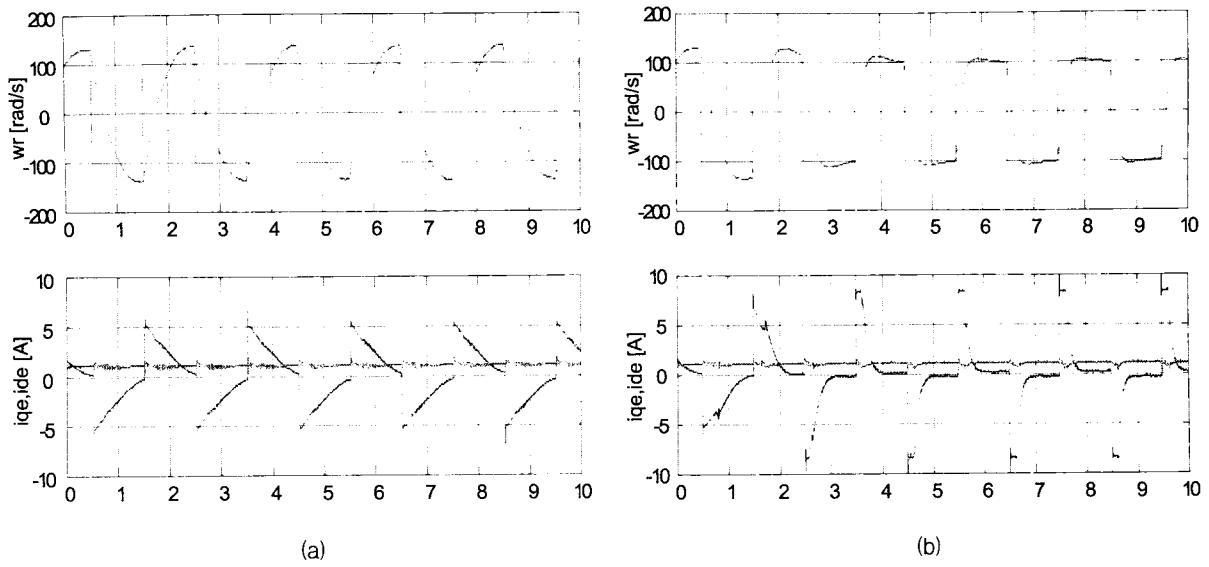


그림 7 기준속도 $\omega_r = \pm 100$ rad/s, 제어기 설계에 사용한 관성모멘트 $J = 0.25 \times J_n$ 일 때 (a) 일반 PID 제어기를 이용한 실험결과 (b) SF-PID 제어기를 이용한 실험결과

Fig. 7 When the reference speed $\omega_r = \pm 100$ rad/s, the moment of inertia used in controller design, $J = 0.25 \times J_n$, the speed and current responses of (a) general PID controller (b) SF-PID controller

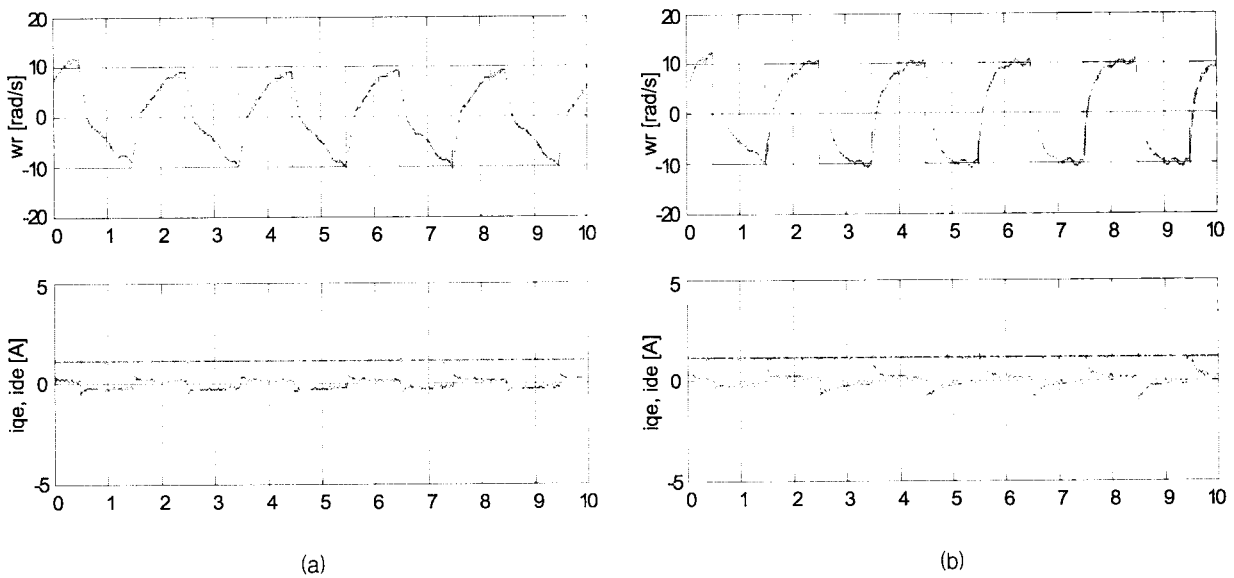


그림 8 기준속도 $\omega_r = \pm 10$ rad/s, 제어기 설계에 사용한 관성모멘트 $J = 0.25 \times J_n$ 일 때 (a) 일반 PID 제어기를 이용한 실험결과 (b) SF-PID 제어기를 이용한 실험결과

Fig. 8 When the reference speed $\omega_r = \pm 10$ rad/s, the moment of inertia used in controller design, $J = 0.25 \times J_n$, the speed and current responses of (a) general PID controller (b) SF-PID controller

6. 결 론

본 논문에서는 일반 PID 제어기와 퍼지 제어기의 특성을 공통으로 갖는 SF-PID 제어기를 설계하고 유도전동기 속도 제어에 적용하였다. 제안된 제어기는 일반 PID 제어기와 구조적인 유사성으로 인하여 PID 제어기에 사용하는 초기 동조 기법을 동일하게 사용할 수 있고 기본 구조가 단순하여 실시간 제어에 적합한 것을 알 수 있었다. 제어기의 출력측 계수를 신경회로망 구조로 변형하고 오차 역전파 알고리즘을 적용함으로써 적응성을 부여하였으며 학습계수를 고정시키지 않고 시스템 오차 정보에 따라 가변하도록 하였다.

속도제어계는 제어기 설계시에 기계적인 파라미터인 관성 모멘트를 사용하기 때문에 이 값의 변화에 민감한 특성을 보인다. 제안한 알고리즘을 유도전동기 정역 속도제어에 적용하여 시뮬레이션 및 실험을 수행하였고 파라미터 변화에 대한 강인성을 확인하였다.

참 고 문 헌

[1] T. Hagglund and K. J. Astrom, "Automatic tuning of PID controller", ISA, NC, 1995.
 [2] Vas. P., Stronach, A. F. and Neuroth M., "Fuzzy, pole-placement and PI controller design for high-performance drives", Stockholm Powr Tech., 1-6, 1995.
 [3] Li J. and Stronach A. F., "Aritificial neural network based control of electromechanical system", 4th European Conference on Control, IEE, Coventry, pp. 1065-1070, 1994.
 [4] Vas. P., Drury W. and Stronach A. F., "Recent developments in artificial intelligence-based drives: a review", PCIM, Nurnberg, pp. 59-71, 1996.
 [5] L. X. Wang, *Adaptive fuzzy systems and control : design and stability analysis*, Prentice Hall, 1994.
 [6] Y. Ishida, "Nonlinear PID controller using neural networks", IEEE Int. Conf. on Neural Networks, pp. 811-814, 1997.
 [7] Li Cs, Priemer R, "Self-learning general purpose PID controller", Journal of the Franklin Institute, vol. 334B, no. 2, 1997
 [8] Marian B. Gorzalczany, "On some idea of a neuro-fuzzy controller", Information Sciences, vol. 120, no. 1-4, 1999.
 [9] D. W. Novotny and T. A. Lipo, *Vector control and dynamics of AC drives*, Oxford University Press, 1996
 [10] W. Z. Qiao, M. Mizumoto, "PID type fuzzy controller and parameters adaptive method", Fuzzy Sets and Systems, vol. 78, pp. 23-25, 1996.
 [11] S. Gallchet and L. Foulloy, "Fuzzy controllers: synthesis and equivalences", IEEE Trans. Fuzzy Syst., vol. 3, pp. 140-148, 1995.

[12] J. H. Im, C. G. Lee, "Design of a neural network based self-tuning fuzzy PID controller", Trans. of KIEE, vol. 50, no. 1, pp. 22-30, 2001.
 [13] Derek P. Atherton, "PID controller tuning", Computing & Control Engineering Journal, vol. 10, pp. 44-50, 1999.
 [14] 한우용, 이창구, 김배선, "Matlab/Simulink & dSPACE 보드를 이용한 유도 전동기 벡터 제어 시스템의 구현", 대한전기학회 논문지, vol. 49, no. 10, pp. 675-682, 2000.

저 자 소 개



김 상 민 (金相敏)

1974년 2월 1일생. 1996년 전북대 제어계측공학과 졸업. 1998년 동 대학원 전기공학과 석사. 1998년~현재 동 대학원 전자공학과 박사과정. 관심분야는 전력전자 및 전동기 제어, 지능제어.
 Tel : 063-270-2476, Fax : 063-270-2451
 E-mail : smkim74@orgio.net



한 우 용 (韓宇勇)

1964년 5월 23일생. 1986년 전북대 전기공학과 졸업. 1994년 동대학원 전기공학과 졸업(박사). 1994년~현재 전주공업대학교 전기과 부교수. 관심분야는 전력전자 및 전동기 제어.
 Tel : 063-220-3834
 E-mail : wyhan@jtc.ac.kr



이 창 구 (李昌求)

1958년 12월 25일생. 1981년 전북대 전기공학과 졸업(학사). 1991년 동 대학원 전기공학과 졸업(박사). 1983년~1992년 한국전자통신연구소 선임연구원. 1996년 Alberta 대학 방문교수. 현재 전북대 공대 전자정보공학부 부교수.
 E-mail : changgoo@moak.chonbuk.ac.kr