

센서리스 유도전동기의 속도제어를 위한 개선된 신경회로망 기반 자기동조 퍼지 PID 제어기 설계

\* 김상민\*, 한우용\*\*, 이창구\*, 한후석\*\*\*  
 전북대학교 전자·정보공학부\*, 전주공업대학 전기과\*\*, 기초전력공학공동연구소\*\*\*

Improved Neural Network-based Self-Tuning Fuzzy PID Controller for Sensorless Vector Controlled Induction Motor Drives

\* Sang-min Kim\*, Woo-yong Han\*\*, Chang-goo Lee\*, Hoo-suk Han\*\*\*  
 Chonbuk National University\*, Jeonju Technical College\*\*, EESRI\*\*\*

**Abstract** - This paper presents a neural network based self-tuning fuzzy PID control scheme with variable learning rate for sensorless vector controlled induction motor drives. MRAS(Model Reference Adaptive System) is used for rotor speed estimation. 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. The proposed scheme is simple in structure and computational burden is small. The simulation using Matlab/Simulink and the experiment using DS1102 board show the robustness of the proposed controller to parameter variations.

1. 서 론

AC 전동기 벡터 제어를 위해 사용하는 PID 속도제어기는 구조가 단순하고 견실하며 현장 운전자에게 익숙하다는 장점을 지니고 있어서 산업현장에 널리 사용되고 있다[1-2]. 그러나 AC 전동기 제어시스템은 비선형성, 전동기 파라미터들의 시변 특성에 의하여 제어성능이 저하되기 때문에 이를 보상하기 위하여 계수들의 재동조가 필요하다. 이러한 제어계수들의 지속적인 동조를 위하여 파라미터 동조기법이나 적응제어 알고리즘들이 많이 제안되어왔다. 최근 일부 논문에서는 퍼지 PID 제어나 뉴로-퍼지 제어 등 신경회로망과 퍼지를 이용한 제어기법들이 연구되고 있다[3-4]. 불행하게도 PID 제어기의 파라미터들을 계속적으로 동조하는 알고리즘들은 계산이 복잡하고 계산량이 많아서 고가의 장비를 이용하지 않는다면 실시간 제어가 어렵다는 단점을 가지고 있다. 또한, 역전과 알고리즘의 특성상 학습계수가 증가하면 수렴속도가 향상되는 반면에 외란에 대한 강인성이 저하되어 시스템이 쉽게 불안정해질 수 있다는 단점이 있다[5-7].

본 논문에서는 퍼지의 선형성을 이용하여 PI-PD형 퍼지 PID 제어기의 퍼지 연산부를 간략화 시키고 제어기 출력에 가장 큰 영향을 미치는 출력측 스켈링 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 변화시키

는 가변 학습계수 역전과 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 DS1102 보드를 이용한 실험시스템을 구성하여 유도전동기 정역 속도제어 실험을 수행하였다.

2. 유도전동기의 센서리스 벡터제어

2.1 유도전동기의 모델링

3상 유도전동기의 동적 상태방정식을 고정자 기준 dq 고정 좌표계에서 표현하면 식 (1)과 같다[8].

$$\begin{aligned} \frac{di_{ds}^s}{dt} &= -a_1 i_{ds}^s + a_2 \phi_{dr}^s + a_3 n_p \omega_r \phi_{dr}^s + a_0 V_{ds}^s \\ \frac{di_{qs}^s}{dt} &= -a_1 i_{qs}^s + a_2 \phi_{qr}^s - a_3 n_p \omega_r \phi_{qr}^s + a_0 V_{qs}^s \\ \frac{d\phi_{dr}^s}{dt} &= -a_4 \phi_{dr}^s - n_p \omega_r \phi_{qr}^s + a_5 i_{ds}^s \\ \frac{d\phi_{qr}^s}{dt} &= -a_4 \phi_{qr}^s + n_p \omega_r \phi_{dr}^s + a_5 i_{qs}^s \\ \frac{d\omega_r}{dt} &= -a_6 \omega_r + a_7 (T_e - T_L) \end{aligned} \quad (1)$$

여기서  $i_{ds}^s, i_{qs}^s$ : 고정자 전류,  $\phi_{dr}^s, \phi_{qr}^s$ : 회전자 자속,  $V_{ds}^s, V_{qs}^s$ : 고정자 전압,  $\omega_r$ : 회전자 속도,  $n_p$ : 폴 수

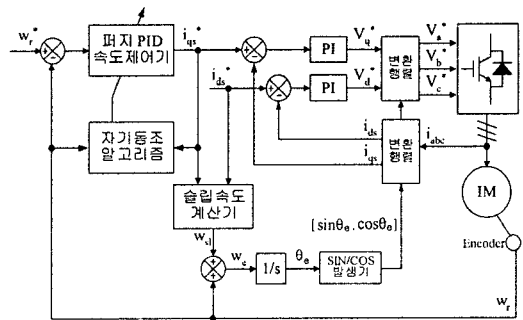


그림 1 간접벡터 유도전동기 구동 시스템

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  $\omega_r$ 와 dq 축 전류명령  $i_{qs}^s, i_{ds}^s$  은 변환행렬(고정좌표계 ↔ 동기좌표계)을 만드는데 사용된다. 본 논문에서는 속도정보를 얻기 위하여 MRAS 기반 속도 추정기를 사용하고, 엔코더는 모니터링을 위해 사용한다.

## 2.2 MRAS 기법을 이용한 속도 추정기 설계

식 (1)에서 고정자 전류와 회전자 자속의 방정식을 이용하여 식 (3)-(4)와 같은 자화 전류 추정기를 설계할 수 있다.

$$p \begin{bmatrix} i_{md}^s \\ i_{mq}^s \end{bmatrix} = \frac{L_r}{L_m^2} \begin{bmatrix} V_{dr}^s \\ V_{qr}^s \end{bmatrix} - \begin{bmatrix} R_s + \sigma L_s & p & 0 \\ 0 & R_s + \sigma L_s & p \end{bmatrix} \begin{bmatrix} i_{md}^s \\ i_{mq}^s \end{bmatrix} \quad (3)$$

$$p \begin{bmatrix} i_{md}^s \\ i_{mq}^s \end{bmatrix} = - \begin{bmatrix} -1/T_r & -\omega_r \\ \omega_r & -1/T_r \end{bmatrix} \begin{bmatrix} i_{md}^s \\ i_{mq}^s \end{bmatrix} + \frac{1}{T_r} \begin{bmatrix} i_{dr}^s \\ i_{qr}^s \end{bmatrix} \quad (4)$$

여기서  $p=d/dt$ ,  $T_r=L_r/R_r$ ,  $(i_{md}^s, i_{mq}^s)^T = (1/L_m) \times (\phi_{dr}^s, \phi_{qr}^s)^T$ 는 각각 미분연산자, 회전자 시정수, 자화전류이다.

본 논문에서는 MRAS 기법을 이용하여 그림 5와 같은 회전자 속도 추정기를 구성한다. 먼저 회전자 속도  $\omega_r$ 를 포함하지 않는 식 (3)을 기준모델로 선정하고  $\omega_r$ 를 포함하는 식 (4)는 적응모델로 선정한다. 그런 다음 각 추정기의 출력과 Popov의 안정도 이론을 이용하여 식 (5)와 같은 속도 동조 신호를 유도해낼 수 있다[9].

$$e_\omega = i_{mq} \hat{i}_{md} - i_{md} \hat{i}_{mq} \quad (5)$$

결과적으로 비례적분 제어기를 이용하여 회전자 속도 추정치는 식 (6)으로부터 얻을 수 있다.

$$\hat{\omega}_r = K_p e_\omega + K_i \int e_\omega dt \quad (6)$$

여기서  $K_p > 0$ ,  $K_i > 0$ 는 비례이득이다.

본 논문에서는 속도제어를 위하여 신경회로망 기반 SF-PID 제어기를 이용한다.

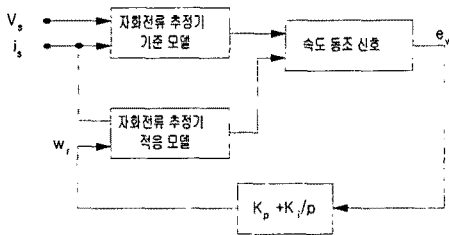


그림 2 MRAS에 기초한 회전자 속도 추정기

## 3. 신경회로망 기반 SF-PID 제어기

### 3.1 퍼지 PID 제어기 설계와 초기 동조

퍼지 제어기는 일반적으로 비선형 특성을 지닌다. 그러나 특별한 형태의 소속 함수와 추론방법, 그리고 제어규칙 등을 이용하여 선형적인 특성을 갖게 할 수 있다 [10-11]. J. H. Im가 제안한 방법을 이용하여 선형 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.

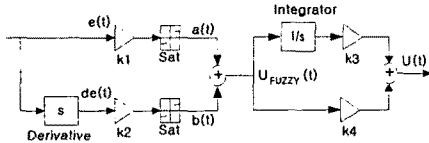


그림 3 간략화된 퍼지 PID 제어기

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

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

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

### 3.2 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 제어기를 동일하게 적용할 수 있다. Ishida는 일반 PID 제어기와 비슷한 구조를 갖는 신경회로망 기반 제어기를 구성하고 비선형 시스템에 적용하여 좋은 결과를 얻었다[5]. 이를 바탕으로 퍼지형 PID 제어기의 출력계수를 그림 4와 같이 단일 신경 회로망 구조로 변형하였다. 그림 4에 대한 이산화된 수식은 식 (8)과 같다.

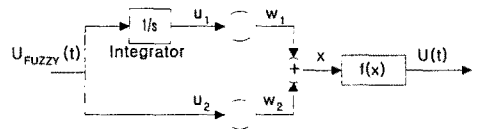


그림 4 신경회로망 구조의 퍼지 PID 제어기 출력부

$$u_1(kT) = u_1((k-1)T) + U_{FUZZY}(kT)T$$

$$u_2(kT) = U_{FUZZY}(kT)$$

$$x(kT) = u_1(kT)w_1(kT) + u_2(kT)w_2(kT)$$

$$U(kT) = f(x(kT), u_1(kT), u_2(kT), w_1(kT), w_2(kT)) \quad (8)$$

본 논문에서  $U(kT)$ 는  $q$ 속 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 활성 함수로서 본 논문에서는  $f(x) = x$ 의 선형 함수를 사용하였으며 성능지수  $E$ 를 식 (9)와 같이 정의하고 성능 지수  $E$ 가 최소화되도록 역전파 알고리즘을 적용하였다.

$$E = \frac{(r(kT) - y(kT))^2}{2}$$

$$= e(kT)^2/2, \quad e(kT) = r(kT) - y(kT) \quad (9)$$

$$w_1(kT) = w_1((k-1)T) - \eta \frac{\partial E}{\partial w_1} \Big|_{kT} \quad (10)$$

$$w_2(kT) = w_2((k-1)T) - \eta \frac{\partial E}{\partial w_2} \Big|_{kT}$$

본 논문에서  $r(kT)$ 와  $y(kT)$ 는 각각 기준 속도  $\omega_r^*$ 와 전동기의 회전자 속도  $\omega_r$ 이며  $\eta$ 는 학습계수이다. 연쇄법칙을 이용하여 식을 전개하고 다시 정리하면 식 (11)을 얻을 수 있다.

$$w_1(kT) = w_1((k-1)T) + \eta e(kT)u_1 \frac{\partial y}{\partial u} \Big|_{kT} \quad (11)$$

$$w_2(kT) = w_2((k-1)T) + \eta e(kT)u_2 \frac{\partial y}{\partial u} \Big|_{kT}$$

학습계수  $\eta$ 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.

$$\eta = \frac{\eta_{\max} - \eta_{\min}}{2} \left[ \frac{1 - e^{-(\tau - \epsilon/2)\delta}}{1 + e^{-(\tau - \epsilon/2)\delta}} + 1 \right] + \eta_{\min} \quad (12)$$

$$\delta = \frac{2R}{\epsilon}$$

$\tau$ 는 시스템 오차와 관련된 정보로서 초기값은 0이며 현

제의 오차부호와 한 스텝 이전 오차부호를 비교하여 부호가 같으면 1씩 증가하고 부호가 다를 경우 초기값으로 초기화된다.  $\varepsilon$  은 사용자가 설정하는 값이며  $\gamma = \varepsilon/2$ 일 때  $\eta = (\eta_{max} - \eta_{min})/2$ 가 된다.  $\delta$  는  $\gamma = \varepsilon/2$ 에서의 기울기를 결정한다.

#### 4. 시뮬레이션 결과

앞 절에서 제시한 신경회로망 기반 자기동조 퍼지 PID 제어기의 성능을 평가하기 위하여 Matlab/Simulink 상에서 벡터제어 유도전동기의 시뮬레이션과 실험을 수행하였다. 실험에 사용한 3상 유도전동기 모델은 표 1에 제시한 상수값을 갖는다.

표 1 유도전동기의 정수 및 정격

|      |                              |       |                   |
|------|------------------------------|-------|-------------------|
| 정격출력 | 400 [W]                      | $R_s$ | 3.68 [ $\Omega$ ] |
| 정격전압 | 120 [V]                      | $R_r$ | 2.4 [ $\Omega$ ]  |
| 정격전류 | 3.5 [A]                      | $L_s$ | 470.6 [mH]        |
| 극 수  | 1                            | $L_r$ | 470.6 [mH]        |
| J    | 0.007257[kg·m <sup>2</sup> ] | $L_m$ | 441.8 [mH]        |

그림 5는 실제값의 0.25배 크기의 관성모멘트를 사용하여 100[rad/s] → -100[rad/s]로 정역운전 할 때의 회전자 속도 및 dq축 전류 파형을 나타낸다. 그림 5(a)는 일반적인 PID 제어기를 사용한 경우로써 약 20%의 오버슈트가 발생하는 것을 알 수 있다. 그림 5(b)는 신경회로망기반 자기동조 퍼지 PID 제어기를 사용한 경우로써 초기에 오버슈트가 발생하지만 시간에 지남에 따라 학습효과에 의하여 오버슈트가 없어지고 정상상태오차도 0으로 뉘을 알 수 있다. dq축 전류 파형을 살펴보면 그림 5(a)의 경우에 토크 성분 전류  $i_{dq}$ 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  $i_{dq}$ 가 빠르게 수렴하게 되어  $i_{de}$ 가 대략 0.3(s)의 과도상태를 지나면 커플링 현상이 사라지는 것을 알 수 있다.

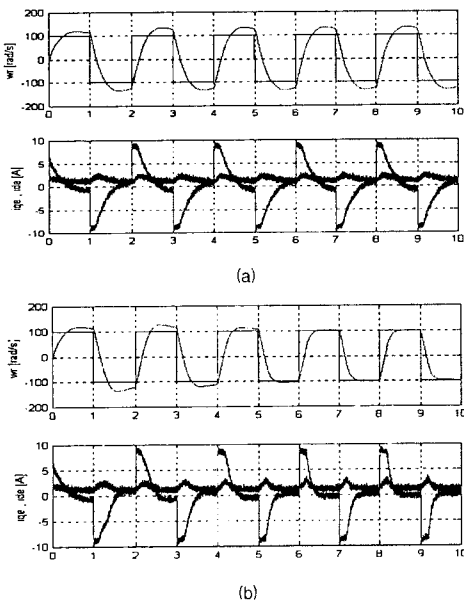


그림 5 기준속도  $\omega_r = \pm 100$  rad/s,  $J = 0.25 \times J_n$ 일 때 (a) 일반 PID 제어기를 이용한 시뮬레이션 결과 (b) 신경회로망 기반 SF-PID 제어기를 이용한 시뮬레이션 결과

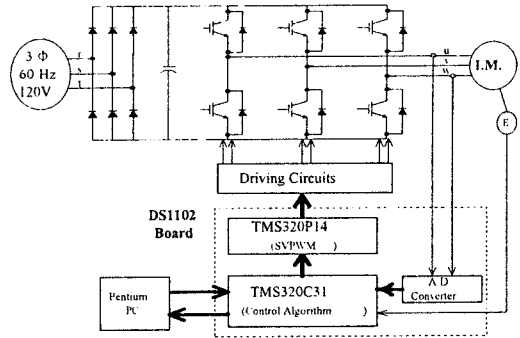


그림 6 실험장치 구성

#### 5. 실험 결과

제안한 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 7은 관성모멘트 값을 실제값의 0.25배로 설정하고 기준속도를  $\pm 100$ [rad/s]로 설정하여 정역운전 하였을 때의 실험결과를 나타낸다. 기존 PID 제어기를 이용하였을 경우에는 20%의 오버슈트가 발생하는 것을 알 수 있다. 제안한 방법을 이용한 경우에는 대략 4(s)의 시간이 지나면 오버슈트가 사라지고 정상상태 오차도 0으로 접근하는 것을 알 수 있다.

그림 8은 관성모멘트 값을 그림 7과 동일하게 설정하고 기준속도를  $\pm 10$ [rad/s]로 설정하여 저속으로 정역운전 하였을 때의 실험결과를 나타낸다. 기존 PID 제어기를 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 0.3(s)로 줄어들고 6(s)가 지난 후에 정상상태 오차가 0으로 접근하는 것을 알 수 있다.

#### 6. 결 론

본 논문에서는 일반 PID 제어기와 퍼지 제어기의 특성을 광범위로 갖는 신경회로망 기반 자기동조 퍼지 PID 제어기를 설계하고 유도전동기 속도제어에 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 역전과 알고리즘을 적용함으로써 적응성을 부여하였으며 학습계수를 고정시키지 않고 시스템 오차 정보에 따라 가변하도록 하였다.

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

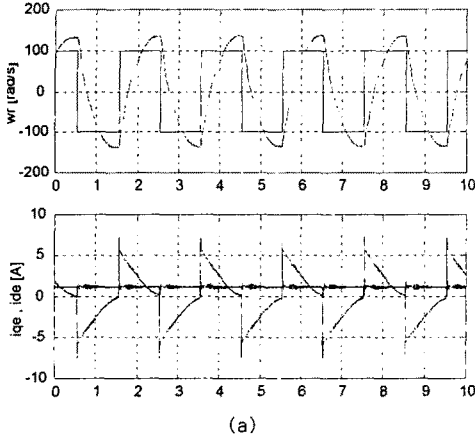


그림 7 기준속도  $\omega_r = \pm 100$  rad/s,  $J = 0.25 \times J_n$  일 때 (a) 일반 PID 제어기를 이용한 실험결과 (b) 신경회로망 기반 자기동조 퍼지 PID 제어기를 이용한 실험결과

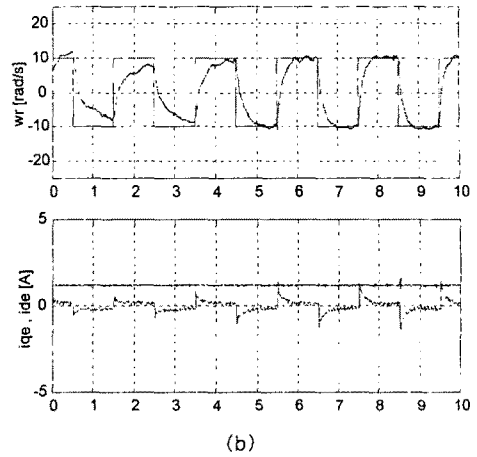
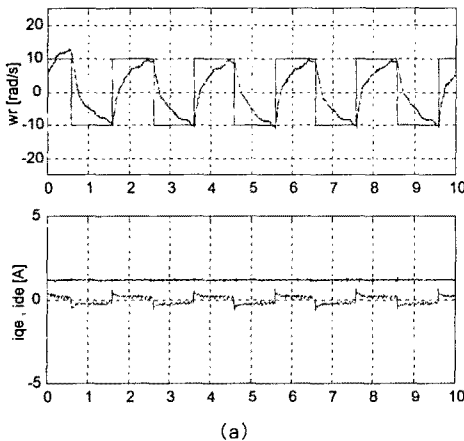


그림 8 기준속도  $\omega_r = \pm 10$  rad/s,  $J = 0.25 \times J_n$  일 때 (a) 일반 PID 제어기를 이용한 실험결과 (b) 신경회로망 기반 자기동조 퍼지 PID 제어기를 이용한 실험결과

#### [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 Power Tech., 1-6, 1995.
- [3] Li J. and Stronach A. F., "Artificial neural network based control of electromechanical system", 4th European Conference on Control, IEE, Conventry, 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] Y. Ishida, "Nonlinear PID controller using neural networks", IEEE Int. Conf. on Neural Networks, pp. 811-814, 1997.
- [6] Li Cs, Priemer R, "Self-learning general purpose PID controller", Journal of the Franklin Institute, vol. 334B, no. 2, 1997
- [7] Marian B. Gorzalczany, "On some idea of a neuro-fuzzy controller", Information Sciences, vol. 120, no. 1-4, 1999.
- [8] D. W. Novotny and T. A. Lipo, *Vector control and dynamics of AC drives*, Oxford University Press, 1996.
- [9] Y. D. Landau, *Adaptive Control : The Model Reference Approach*, New York: Marcel Dekker, 1982.
- [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, 이창규, 김배선, "Matlab/Simulink & dSPACE 보드를 이용한 유도 전동기 벡터 제어 시스템의 구현", 대한전기학회 논문지, vol. 49, no. 10, pp. 675-682, 2000.