

# 자기학습형 퍼지제어기에 의한 유도전동기 고성능 속도제어에 관한 연구

\*박영민\*, 김연중\*, 김재문\*, 원충연\*, 김영렬\*\*, 김학성\*\*\*  
 \*성균관대학교, \*\*안양대학교, \*\*\*동양공업전문대학

## A Study on the High Performance Speed Control of Induction Motor Using Self-Learning Fuzzy Controller

Y.M. Park\*, Y.C. Kim\*, J.M. Kim\*, C.Y. Won\*, Y.R. Kim\*\*, H.S. Kim\*\*\*  
 \*Sung Kyun Kwan Univ., \*\*An Yang Univ., \*\*\*Dong Yang Tech. College

### Abstract

In this paper, an auto-tuning method for fuzzy controller based on the neural network is presented. The backpropagated error of neural emulator offers the path which reforms the fuzzy controller's membership functions and fuzzy rule, and used for speed control of induction motor. For the torque control method, an indirect vector control scheme with slip calculation is used because of its stable characteristics regardless of speed. Motor input current is regulated by a current controlled voltage source PWM inverter using space voltage vector technique. Also, the scheme of current control fuzzy controller is synchronous reference frame with decoupling term.

DSP(TMS320C31) is used to achieve the high speed calculation of the space voltage vector PWM and to build the self-learning fuzzy control algorithm. An IPM is used to simplify hardware design.

### 1. 서론

최근 그 응용 범위가 넓어지고 있는 유도전동기는 유지 보수면과 경제성 등에서 직류전동기보다 우수하고, 백터제어 이론과 전력전자 기술의 발달로 어느 정도 높은 정밀도의 제어를 요구하는 시스템에도 적용이 가능하게 되었다. 하지만, 유도전동기는 심한 비선형성을 나타내므로 PID제어기와 같은 기존의 선형제어기로는 만족할 만한 제어특성을 기대하기 힘들다. 따라서 유도전동기를 제어하기 위한 새로운 제어방식에 관한 연구가 진행되고 있는데, 이 중 퍼지제어기는 시스템의 모델링없이 설계자의 경험적 지식만으로도 설계가 가능하다는 것 등의 많은 장점을 지니고 있어 퍼지제어기를 이용한 유도전동기 제어 논문이 많이 발표되고 있다.[2]

퍼지제어기는 불확실하고 모호한 상태를 퍼지변수들로 표현하며 이러한 퍼지변수 입력에 대해 설계자의 직관이나 경험에 의한 법칙을 적용하여 역시 퍼지변수로 나타내어지는 출력을 발생하는 제어기이다.

퍼지추론과 신경회로망의 유사성은 퍼지추론의 최소-최대 연산(Min-Max operation)이 신경회로망의 곱셈-덧셈연산(Product-Sum)에 해당되며, 또한 부분적인 특성함수인 멤버십 함수, 시그모이드 함수에 의해 시스템 전체의 복잡한

비선형성을 표현한다는 점이며, 퍼지 추론이 논리구조를 취급하는 반면에, 신경회로망은 학습기능을 가지는 상호 보완 관계가 있다.[1]

본 논문에서는 속도제어를 위하여 신경망 에뮬레이터를 이용하여 퍼지제어기의 소속함수와 제어규칙을 실시간으로 학습하는 자기학습형 퍼지제어기를 사용하였다. 전류제어기법은 동기좌표계에서의 퍼지제어를 하였고 이에 해당하는 정자좌표계에서의 3상 전압을 얻기 위하여 공간 전압벡터 변조 방법을 이용하였다.

### 2. 시스템 구성

유도전동기 속도제어와 전류제어를 위한 전체 시스템은 그림1과 같다.

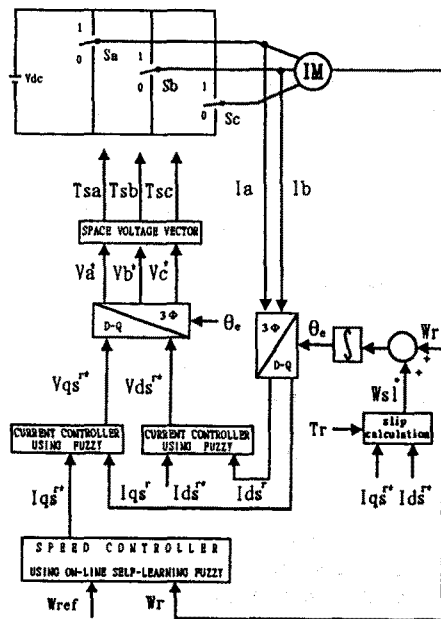


그림 1. 제어 알고리즘 블록도

파라미터의 변화나 외란과 같은 제어환경의 변화에 대응할 수 있도록 학습기능을 가지고 있는 자기학습형 퍼지제어기를 비선형 시변계인 유도전동기 속도 제어 시스템에 적용하여 실시간으로 auto-tuning할 수 있도록 하였다. 또한, 연산시간을 최소화하기 위하여 퍼지 전류제어기는 look-up table을 이용하고, 공간전압벡터는 dq축이 아닌 abc축에서 구현하였으며[5], 속도에 관계없이 안정된 특성을 지닌 간접벡터제어를 이용하였다.[4]

### 2-1. 속도 제어기

퍼지 제어기와 신경망 에뮬레이터를 사용한 유도전동기 속도 제어 시스템은 그림2와 같다.

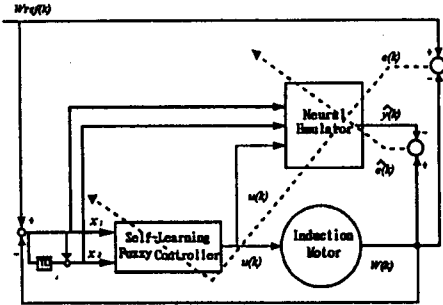


그림 2. 유도전동기 속도제어 시스템

구성된 유도전동기의 속도 제어 시스템 동작 순서를 살펴 보면 다음과 같다. 먼저 전체 시스템을 초기화한다. 그리고 퍼지제어기의 출력으로서 임의의 값을 인가하여 매 샘플링 시간마다 신경망 에뮬레이터의 출력과 플랜트의 출력과의 오차를 최소화하는 방향으로 에뮬레이터를 학습시킨다. 예비학습 단계를 거치면 기준 속도 \$Wref(k)\$와 전동기 실제 속도 \$W(k)\$와의 오차 \$x\_1\$과 오차의 변화율 \$x\_2\$가 자기 학습형 퍼지 제어기 입력으로 들어가고 출력으로 토크성분 전류 기준값인 \$u(k)\$를 내보낸다. 그리고 제어기 입력값인 \$x\_1\$과 \$x\_2\$가 \$u(k)\$와 함께 다시 에뮬레이터 입력으로 사용되고 에뮬레이터에서 계산된 출력값과 실제 전동기 속도와의 오차를 이용하여 신경망 에뮬레이터를 학습시킨다. 에뮬레이터 학습 단계가 완료되면 자기 학습형 퍼지 제어기의 학습을 위하여 기준 속도와 전동기의 실제 속도와의 오차를 신경망 에뮬레이터를 통하여 역전파시켜 자기 학습형 퍼지 제어기 출력단에서의 오차량을 계산한다. 그리고 마지막으로 이 오차량을 이용하여 신경회로망의 형태로 구성된 자기 학습형 퍼지 제어기의 제어규칙과 소속함수를 학습시키게 된다.

본 논문에 쓰인 에뮬레이터는 온라인으로 몇 개의 임의의 제어 입력 샘플을 임의의 시간 동안 제안하는 온라인 예비 학습 단계를 거친다. 그리고 난 후 제어기의 학습을 한다.[2]

### 2-2. 신경망 에뮬레이터

그림3은 유도전동기의 동특성을 모의하기 위하여 사용된 신경회로망 에뮬레이터를 보여준다.

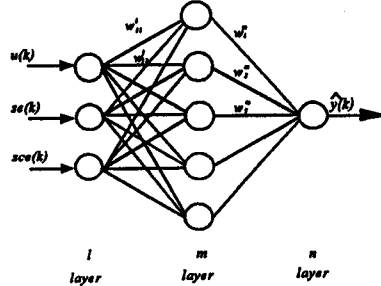


그림 3. 신경회로망 에뮬레이터의 구조

신경망 에뮬레이터의 입력과 출력의 오차함수는 다음과 같이 정의된다.

$$\hat{e}(k) = \frac{1}{2}(y(k) - \hat{y}(k))^2 \quad (1)$$

여기서 \$y(k)\$는 플랜트의 출력, \$\hat{y}(k)\$는 에뮬레이터 출력이다. 학습은 오차를 최소화하도록 일반화된 델타 규칙에 의해 이루어지며, 빠른 학습 속도와 오차진동을 적게하기 위해 관성항(momentum)이 추가되는데 이 경우 가장치 조건은 다음과 같이 표현된다.

$$\Delta W_j^m(k+1) = \eta \delta^m out_j^m + \alpha \Delta W_j^m(k) \quad (2)$$

$$W_j^m(k+1) = W_j^m(k) + \Delta W_j^m(k+1) \quad (3)$$

여기서 \$k\$는 학습 반복 횟수, \$\eta(0 < \eta < 1)\$는 학습률, \$\alpha(0 < \alpha < 1)\$는 관성항이다.

### 2-3. 퍼지제어기의 구성과 학습

그림4는 퍼지제어기를 나타낸 것으로 퍼지 제어기의 퍼지화에 해당하는 입력층, 제어규칙을 구성하는 은닉층, 비퍼지화에 해당하는 출력층 등의 3층 구조로 구성되며, 입력은 범종형, 출력은 싱글턴형으로 구성하였다.

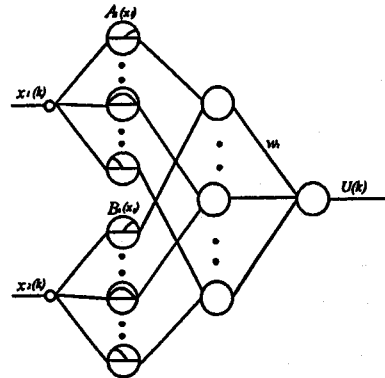


그림 4. 퍼지제어기

입력에 대한 퍼지변수의 소속함수값은

$$A_i(x_1) = \frac{1}{1 + \left(\frac{x_1 - a_i}{a_i}\right)^2} \quad (4)$$

$$B_i(x_2) = \frac{1}{1 + \left(\frac{x_2 - b_2}{b_a}\right)^2} \quad (5)$$

이 값들은 은닉층으로 전달되며 은닉층에서는 이 두 값을 곱한 값

$$\mu_i = A_i(x_1) \times B_i(x_2) \quad (6)$$

를 출력하며 가중치  $W_i$ 와 곱해서 출력층으로 전달한다.

출력층은 비퍼지화 단계에 해당하므로 다음 식과 같이 무게중심법에 의해 비퍼지화를 실행한다.

$$y = \frac{\sum \mu_i W_i}{\sum \mu_i} \quad (7)$$

퍼지제어기의 학습과정은 신경망 에뮬레이터를 통하여 역전파법 오차항을 이용하여 역전파 알고리즘을 통하여 이루어진다.

퍼지제어기 출력층에서의 가중치 변화량은

$$\Delta W_i \propto -\frac{\partial e(k)}{\partial W_i(k)} \quad (8)$$

체인 룰(chain rule)에 의해

$$\begin{aligned} \Delta W_i &= -\eta \frac{\partial e(k)}{\partial W_i(k)} \\ &= -\eta \frac{\partial e(k)}{\partial u(k)} \times \frac{\partial u(k)}{\partial W_i(k)} \\ &= -\eta \delta^{*+1} \frac{\partial}{\partial W_i(k)} \left( \frac{\sum \mu_i(k) W_i(k)}{\sum \mu_i(k)} \right) \\ &= -\eta \delta^{*+1} \frac{\mu_i(k)}{\sum \mu_i(k)} \end{aligned} \quad (9)$$

입력층의 소속함수 파라미터의 변화량  $\Delta a_n$ 은

$$\begin{aligned} \Delta a_n &\propto -\frac{\partial e(k)}{\partial a_n(k)} \quad \text{이므로} \\ \Delta a_n &= -\eta \frac{\partial e(k)}{\partial u(k)} \frac{\partial u(k)}{\partial \mu_i(k)} \frac{\partial \mu_i(k)}{\partial A_n(k)} \frac{\partial A_n(k)}{\partial a_n(k)} \\ &= -\eta \delta^{*+1} \frac{1}{\sum \mu_i(k)} (W_i(k) - u(k)) \times \end{aligned} \quad (10)$$

$$B_i(x_2(k)) \frac{1}{a_n(k)} [A_i(k) - A_i^2(k)]$$

유사하게

$$\begin{aligned} \Delta a_e &= -\eta \frac{\partial e(k)}{\partial u(k)} \frac{\partial u(k)}{\partial \mu_i(k)} \frac{\partial \mu_i(k)}{\partial A_i(k)} \frac{\partial A_i(k)}{\partial a_e(k)} \\ &= -\eta \delta^{*+1} \frac{1}{\sum \mu_i(k)} (W_i(k) - u(k)) \times \end{aligned} \quad (11)$$

$$B_i(x_2(k)) \frac{1}{a_e(k)} [A_i(k) - A_i^2(k)]$$

$$\begin{aligned} \Delta b_n &= -\eta \frac{\partial e(k)}{\partial u(k)} \frac{\partial u(k)}{\partial \mu_i(k)} \frac{\partial \mu_i(k)}{\partial B_i(k)} \frac{\partial B_i(k)}{\partial b_n(k)} \\ &= -\eta \delta^{*+1} \frac{1}{\sum \mu_i(k)} (W_i(k) - u(k)) \times \end{aligned} \quad (12)$$

$$A_i(x_1(k)) \frac{1}{b_n(k)} [B_i(k) - B_i^2(k)]$$

$$\begin{aligned} \Delta b_e &= -\eta \frac{\partial e(k)}{\partial u(k)} \frac{\partial u(k)}{\partial \mu_i(k)} \frac{\partial \mu_i(k)}{\partial B_i(k)} \frac{\partial B_i(k)}{\partial b_e(k)} \\ &= -\eta \delta^{*+1} \frac{1}{\sum \mu_i(k)} (W_i(k) - u(k)) \times \end{aligned} \quad (13)$$

$$A_i(x_1(k)) \frac{1}{b_e(k)} [B_i(k) - B_i^2(k)]$$

### 3. 전류제어기법과 전압벡터의 변조

본 연구에서는 전류제어기법으로 동기좌표계에서의 퍼지제어기를 이용한다.[3] 일반적인 정지좌표계에서의 전류제어 기법은 위상지연이 발생하여 제대로 전류 지령치를 추종 못하는데 반해 이 방법은 모든 제어변수들을 동기좌표 상에서 제어하는 것으로서 벡터제어와 함께 사용될 수 있다

각 상의 스위칭 상태에 따라 인버터의 출력전압을 복소평면상에 벡터로 표시하면 6개의 유효벡터와 2개의 영벡터로 표현 가능하다.

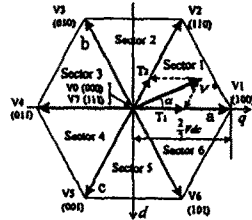


그림 5. 공간 전압 벡터 다이어그램

3상 인버터의 최대 출력 전압은 그림5에 보인 바와 같이 벡터적으로 정육각형을 이루고 있고 PWM을 행하면 평균적인 의미에서 정육각형 내부의 임의의 점에 놓일 수 있다. 출력하고자 하는 전압 벡터가 위치한 삼각형의 두 변에 해당하는 전압 벡터로부터 지령 전압을 합성할 수 있고 두 전압 벡터의 인가 시간의 합이 전류 샘플링 시간  $T_s$ 보다 작으면 나머지 시간만큼 영벡터를 인가한다.[5]

### 4. 실험 결과

그림6은 유도전동기 제어 시스템의 하드웨어 구성도를 나타낸다.

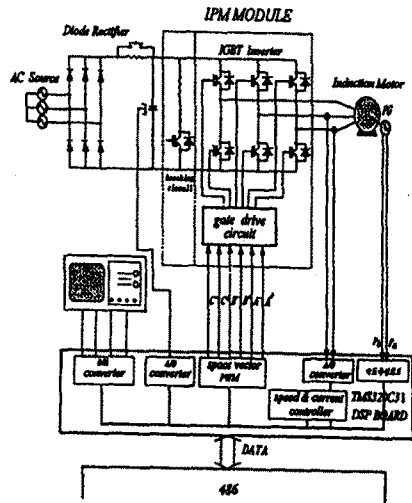
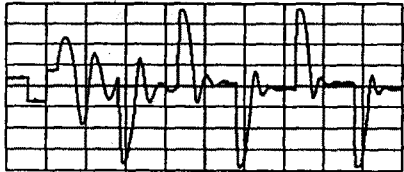


그림 6. 실험장치 구성도

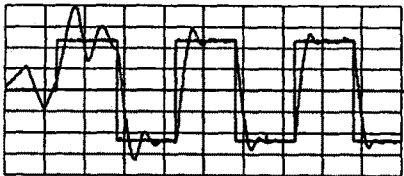
**표 1. 실험에 사용된 전동기 계격 및 상수**

전압 : 220[V]    출력 : 2.2[Kw]    극수 : 4[극]  
 회전수 : 1720[rpm]    주파수 : 60[Hz]  
 $R_s$  : 0.687[Ω]     $R_r$  : 0.842[Ω]  
 $L_s$  : 0.08397[H]     $L_r$  : 0.08525[H]  
 $M$  : 0.08136[H]     $J$  : 0.03[Kg · m<sup>2</sup>]     $B$  : 0.01[N · m/sec]

속도제어기 샘플링 주기는 3[ms]로 하였으며 5회 학습만으로도 off-line 퍼지제어기 보다 우수하게 학습하며, 전류제어기 샘플링 주기는 100[μs]로 PI제어기보다 퍼지제어기가 부하변동에 우수함을 확인할 수 있었다.

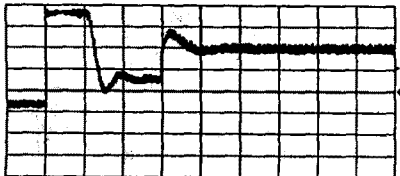


(X축: 2 sec/div, Y축: 2 A/div)  
 (a)토크 성분 전류 지령치

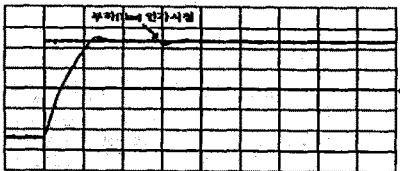


(X축: 2 sec/div, Y축: 400 rpm/div)  
 (b)지령속도와 전동기 속도

그림 7. 자기학습형 퍼지제어기의 학습과정



(X축: 0.5 sec/div, Y축: 2 A/div)  
 (a)토크 성분 전류

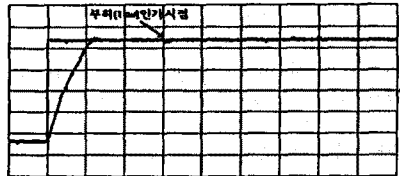


(X축: 0.5 sec/div, Y축: 400 rpm/div)  
 (b)지령속도와 전동기 속도

그림 8. 퍼지 속도제어기와 PI 전류제어기



(X축: 0.5 sec/div, Y축: 2 A/div)  
 (a)토크 성분 전류



(X축: 0.5 sec/div, Y축: 400 rpm/div)  
 (b)지령속도와 전동기 속도

그림 9. 5회 학습 후의 자기학습형 퍼지 속도제어기와 퍼지 전류제어기

**5. 결론**

본 논문에서는 유도전동기 속도제어를 위하여 간접형 신경피로망의 신경망 에뮬레이터를 이용하여 퍼지제어기의 소속함수와 제어규칙을 실시간으로 학습하는 자기학습형 퍼지제어기를 사용하였다. 전류제어기에는 일반적인 퍼지제어기와 공간벡터변조기법을 이용하여 부하변화에 강인성을 얻었다. 이러한 복잡한 연산은 종래의 범용 마이크로프로세서를 이용할 경우 연산에 많은 시간을 필요로 하여 실제 구현에 어려움이 있다. 그래서 본 논문에서는 32bit 부동소수점 연산이 가능한 Texas Instrument사의 고성능 DSP TMS320C31과 DSP 에뮬레이터 White-Mountain사의 Mountain-510/30을 이용하여 IBM- PC486 시스템에서 개발하여 본 제어기의 우수성을 입증하였다.

**참고 문헌**

[1] Jyh-Shing R. Jang, "Self-Learning Fuzzy Controllers Based on Temporal Back Propagation", IEEE Trans. on Neural Networks, Vol.3, No.5, pp.714-723, Sep. 1992.  
 [2] 박영주, "유도전동기 속도제어를 위한 퍼지-뉴럴제어기 설계에 관한 연구", 성균관 대학교 석사학위 논문, 1996.  
 [3] Gilberto C. D. Sousa and Bimal K. Bose, "A Fuzzy Set Theory Based Control of a Phase-Controlled Converter DC Machine Drive", IEEE Trans. on Industry Applications, Vol.30, No.1, pp.34-44, Jan. 1994.  
 [4] B. K. Bose, Power Electronics and AC Drives, 1986.  
 [5] 김준석, "공간 전압 벡터 PWM의 새로운 기법", 전기학회 논문지44권 7호, pp.865-874, 1995. 7  
 [6] Texas Instrument, TMS320C3x User's Guide, 1994.