

퍼지 이론을 이용한 DC 모터의 위치 제어기 구성

⁰전영수* 김정도* 정동근** 이택중*
*성균관 대학교 전자공학과, **서울 보건 전문대학

The Construction of Fuzzy Controller
for DC Servo Motor Position Control

Young Su Jean, Jeong DO Kim, Dong Kuen Cheong, and Taek Chong Lee

* Dept. of Electronic Engr. S.K.K.Univ.

ABSTRACT

In this paper, the control of a DC servo motor position control is implemented with fuzzy controller using IBM PC AT. In addition to the fuzzy controller with the error and change of error inputs PI controller is applied as well to improve the steady state error response. It shows the above configuration results in a good control characteristics.

1. 서론

퍼지제어이론은 1965년 퍼지집합[1] 이론을 제안한 미국의 Zadeh교수가 1973년 이 이론을 시스템 자동 제어에 응용할 수 있다는 의견을 제안한 후, 1974년 영국의 Mamdani교수에 의해 스티븐진 제어에 처음으로 응용되어 실용가능성을 보여 주었고, 퍼지제어 알고리즘을 적용하여 종래의 제어이론의 결과보다 많은 경우 우수함이 증명되었다.[2]

기존의 제어 이론인 PID 제어는 주위 환경에 따라서 파라메타가 변동하므로 결정된 제어계수를 바꾸어 주어야 하고, 파라메타의 변동에 대처가능한 적응 제어 방식은 제어 알고리즘이 복잡하고 시스템의 정확한 파악이 불가능한 경우에는 적용 시키기가 어렵다. 그러므로 이러한 문제를 해결 하고자 인간의 사고 방식을 모방하는 인공지능을 제어 이론에 도입하게 되었고 이것이 퍼지 제어의 출현을 가져왔다.

퍼지제어의 특징[3]으로는 첫째 논리제어라는 점이다. 이것은 퍼지제어 규칙이 제어논리를 표현하기 쉬운 if...then-형식의 논리형이라는 것과, 규칙의 운용은 추론에 의해 행해지고, 다양한 전제 조건에 대응할 수 있다는 것을 의미한다. 둘째는 병렬(분산)형 제어인 점이다. 이것은 종래의 고전제어나 현대제어에서는 하나의 수식으로써 전체적인 제어 규칙을 나타내고 있지만, 퍼지제어에서는 부분적인 복수개의

제어 규칙을 사용한다. 하나의 규칙은 제어에 이용되는 정보 공간의 특정한 영역에서 유효한 것이고, 분할된 정보 공간의 부분 영역마다 제어 규칙을 만들면 된다. 또 제어량이 복수개 있을 때도 목적별로 제어 규칙을 만들면 된다. 종래 제어에서는 목적을 하나로 정리해서 함수로 표시하지 않으면 안되기 때문에 이론적으로 취급할 수 있는 목적의 종류는 매우 한정된 것이었다. 퍼지제어에서는 목적함수를 필요로 하지 않으므로 다양한 목적에 잘 대처할 수 있다는 것이다. 세째는 언어적 제어라는 점이다. 이것은 if-then 형식의 규칙이 언어를 사용하여 쓰여져 있어서 손쉽게 제어기의 성능을 개선시킬 수도 있다는 것이다. 네째는 정확한 수학적 모델을 요구하지 않고 인간의 의사결정 방식을 도입하는 보다 인공지능적인 제어기의 구성이 가능하다는 것이다.

퍼지제어는 퍼지논리(fuzzy logic)를 기초로 하고 있으며 퍼지논리는 기존의 논리체계보다는 훨씬 더 인간의 사고방식과 언어 체계에 유사한 논리이며 불확실성과 대략적인 사실을 표현하는 효과적인 수단을 제공한다. 그리고 대상 플랜트(plant)에 대한 정보가 언어적인 수단으로만 표현이 가능한 경우나, 수리적인 연산으로 처리하기 매우 힘들 경우에 퍼지 논리를 이용한 제어기를 유용하게 사용할 수 있다.

DC 전동기는 산업계에서 가장 널리 쓰이고 있으나 주위 환경에 따라 파라메타의 변화가 있다. 이런 경우 변화에 대처가능한 제어기의 설계가 요구된다. 본 논문에서는 퍼지 제어기의 이런 유용성을 DC 전동기의 위치제어 시스템에 적용하여 내부의 동적 모델을 고려하지 않고도 충분한 제어성능을 나타내는 것을 보인다. 디지털 컴퓨터에 적용 시키기 위해서 양자화를 하였고, 퍼지추론 방법으로는 직접법을 사용하였으며, 비퍼지화에는 무게중심법을 사용하였다. 그리고 설정치 근처에서의 정밀제어를 위하여 오차가 큰 경우와 작은 경우로 구분하여 제어하는 Dual mode 방법[4]을 사용하였다. 그리고 오차와 오차의 변화로부터 제어규칙을 얻는 PD 형태의 제어규칙을 사용하였다. 그러나 일반적으로 PD 형태의 제어기에서는 정상 상태 오차가 존재한다. 그러므로 본 논문에서는 PI 제어기를 퍼지 제어기와 병렬로 연결하여 정상상태

오차를 감소시켰다.

퍼지 위치제어 시스템을 실제 하드웨어로 DC 전동기의 위치 제어에 적용하였다. 제어기로 IBM PC AT를 사용하였고, PC와 전동기 사이의 인터페이스 카드, 전동기 구동을 위한 PWM 구동회로, 위치검출 회로 등을 제작하였다.

2. 퍼지제어기의 구성

퍼지논리 제어기는 모든 동작이 퍼지집합의 퍼지 연산으로 구성되게 된다. 그러므로 두가지의 서로 다른 시스템을 연결하여 주는 장치가 필요하게 되며 퍼지화기와 비퍼지화기가 이 역할을 담당한다. 그림 2.1은 퍼지제어기의 구성도이다. 여기서 퍼지화기는 퍼지하지 않은 시스템으로부터 얻은 출력 오차를 퍼지한 값(퍼지 집합)으로 변환시키는 장치를 말하며, 비퍼지화기는 퍼지 연산에 의해서 구해진 퍼지값으로부터 실제 플랜트를 제어하기 위해서 필요한 비퍼지값(crisp-value)을 만들어 내는 부분을 말한다.

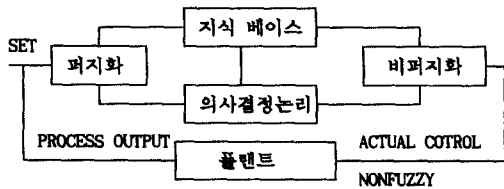


그림 2.1. 퍼지제어기의 구성

2-1. 퍼지화 (fuzzification)[5][6]

본 논문에서는 입력변수로 오차와 오차의 변화를 사용하였다. 그리고 퍼지화시에 사용한 소속 함수는 퍼지 싱글톤이다. 퍼지 싱글톤은 측정된 입력 x_0 에서 소속함수값이 1 이고, 그 이외의 점들에 대해서는 소속함수값이 0인 퍼지집합이다. 그리고 퍼지집합의 대집합은 디지털 컴퓨터를 이용한 정보의 처리, 제어기 구성상의 편리성을 위하여 양자화(quantization)된 값으로 결정하였다.

2-2. 양자화

본 논문에서는 대집합을 이산화 시킴으로써 양자화를 하였다. 이때 이산화된 전체집합의 각각의 원소에 소속값을 부여함으로써 퍼지집합을 정의하였다. 그리고 양자화 과정시, 측정된 입력 변수의 범위를 이산화한 전체집합에 scale mapping을 하였다.

2-3. 퍼지추론

퍼지 제어기는 복수개의 퍼지 제어 규칙과 퍼지추론을 통해서 제어 입력을 찾는다. 퍼지 제어기의 입력을 e (error), ce (change of error), 출력을 y 로 하면 제어 규칙은 if e is A and ce is B then y is C 와 같이 표현된다. 그리고 여기서 A, B, C 는 퍼지집합의 언어 변수이다. 언어변수란 변수의 값으로 숫자가 아니라 자연언어의 단어나 문장이 올때의 변수를 말한다. 추론방법은 퍼지관계의 합성법칙에 의거한 직접법을 사용한다. 퍼지 제어 규칙이 n 개이고, 다음과 같이

or 로 결합되어 있다고 하자.

$$R^1 : \text{if } e \text{ is } A_{11} \text{ and } ce \text{ is } A_{12} \text{ then } y \text{ is } B_1$$

or

$$R^2 : \text{if } e \text{ is } A_{21} \text{ and } ce \text{ is } A_{22} \text{ then } y \text{ is } B_2$$

: or

:

$$R^i : \text{if } e \text{ is } A_{i1} \text{ and } ce \text{ is } A_{i2} \text{ then } y \text{ is } B_i$$

:

:

$$R^n : \text{if } e \text{ is } A_{n1} \text{ and } ce \text{ is } A_{n2} \text{ then } y \text{ is } B_n$$

이때 i 번째의 제어규칙은 e, ce, y 의 공간을 각각 E, CE, Y 로 하면, $E \times CE \times Y$ 상의 퍼지 관계 R_i 에 의해

$$(e, ce, y) \text{ is } R_i \quad (2.1)$$

로써 나타낼 수 있다. R_i 는 $R_i = (A_{i1} \times A_{i2}) \times B_i$ 로 된다.

$R^1 R^2 \dots R^n$ 은 or 결합되어 있기 때문에 n 개의 제어규칙 전체의 퍼지관계는

$$R = R^1 \cup R^2 \cup \dots \cup R^n = \bigcup_{i=1}^n R_i \quad (2.2)$$

지금 입력이 A_1^0, A_2^0 라고 하면 출력 B^0 는

$$B^0 = R * (A_1^0 \times A_2^0) \quad (2.3)$$

$$B^0(y) = \max_{e, ce} [R(e, ce, y) \wedge A_1^0(e) \wedge A_2^0(ce)] \quad (2.4)$$

이지만 제어를 위한 정보를 주는 전건부 변수 e, ce 는 확정된 수치로서 관측되므로 그 값을 e^0, ce^0 라고 하자. 이 경우 식(3.4)의 A_1^0, A_2^0 를

$$A_1^0(e) = \begin{cases} 1 & e^0 = e \\ 0 & e^0 \neq e \end{cases} \quad (2.5)$$

$$A_2^0(ce) = \begin{cases} 1 & ce^0 = ce \\ 0 & ce^0 \neq ce \end{cases} \quad (2.6)$$

라 한다면 식(2.4)는 퍼지집합의 특수한 경우이므로 식(2.4)는

$$B^0(y) = R(e^0, ce^0, y) \quad (2.7)$$

와 같이 간단히 된다.

식(2.2), 식(2.3)을 사용하면 식(2.7)의 우변은

$$R(e^0, ce^0, y) = R^1(e^0, ce^0, y) \vee R^2(e^0, ce^0, y) \vee \dots \vee R^n(e^0, ce^0, y) \quad (2.8)$$

$$\text{여기서 } R_i(e^0, ce^0, y) = A_{i1}(e^0) \wedge A_{i2}(ce^0) \wedge B_i(y) \quad (2.9)$$

와 같이 쓸 수 있다.

이제,

$$\omega_i = A_{i1}(e^0) \wedge A_{i2}(ce^0) \quad (2.10)$$

라고 정의하면, 식(2.7)은

$$\begin{aligned} B^0(y) &= [\omega_1 \wedge B_1(y)] \vee [\omega_2 \wedge B_2(y)] \vee \dots \vee [\omega_n \wedge B_n(y)] \\ &= \bigvee_{i=1}^n [\omega_i \wedge B_i(y)] \end{aligned} \quad (2.11)$$

와 같이 된다.

2-3. 비퍼지화

추론의 결과로서 얻어지는 것은 퍼지집합이며 기본적으로

비퍼지화는 출력의 대집합에서 정의된 퍼지한 제어값을 명확한 제어값으로 대응시키는 것이다. 비퍼지화 방법으로서 가장 널리 쓰이는 방법은 무게중심법이다. 무게중심법은 전체 소속함수의 무게중심이 되는 곳을 제어값으로 결정하는 것이다. 출력 대집합이 이산화된 본 논문의 경우 다음과 같은 식을 비 퍼지화에 사용하였다.

$$y^o = \frac{\sum_{j=1}^n h(y_j) y_j}{\sum_{j=1}^n h(y_j)} \quad (2.12)$$

여기서 y_j 는 양자화 레벨값이고 $h(y_j)$ 는 거기에 해당하는 소속함수 값이다. n 은 양자화 레벨의 개수이다.

2-4. 제안된 제어시스템

일반적으로 퍼지제어기를 구성할 때 사용되는 입력 변수는 오차와 오차의 변화이다. 즉, 오차와 오차의 변화에서 제어 입력을 찾는 것이다.

R_i : if error(e) is A_i and change of error(ce) is B_i
then output is C_i

$$K_3 U[n] = F(K_1 e[n], K_2 ce[n]) \quad (2.13)$$

여기서 F 는 규칙 베이스에서 정의된 퍼지 관계이고, K_i ($i=1, 2, 3$)은 scale 계수이다.

만약에 F 가 선형 함수라면 기존의 PD 제어가 되는 것을 알 수가 있다. [5][7] 그러므로 오차와 오차의 변화에서 제어 입력을 찾는 퍼지 제어기는 일반적인 PD 제어기라고 할 수 있다. 그러나 일반적으로 PD 제어기에서는 정상상태 오차가 존재하므로 PD형태의 퍼지 제어기에서도 정상상태 오차가 존재하게 된다. 그러므로 본 논문에서는 정상상태 오차를 감소시키기 위하여 그림 2.2 처럼 병렬로 PI(비례 적분) 제어기를 부과하였다. PI제어기는 정상상태 오차를 감소 시킬수가 있으므로 정상상태에서의 오차를 줄여준다.

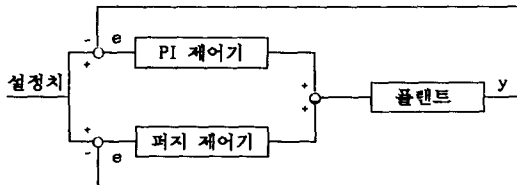


그림 2.2 제안한 퍼지제어기

3. 제어시스템의 구성

3.1 하드웨어의 구성

본 논문에서는 직류 전동기의 퍼지 위치 제어기를 구성하기 위하여 IBM PC AT를 이용하였다. 제어기로 IBM PC AT를 사용하였고, PC와 직류 전동기 시스템간을 연결하는 인터페이스 카드, 전동기를 구동하기 위한 PWM 구동회로, 전동기의 위치 측정을 위한 위치 검출회로 등을 제작하였다. 본 논문에서 사용한 제어시스템의 블록선도는 그림 3.1에 나타나 있다. 제어기인 IBM PC AT에서는 퍼지 알고리즘에 따른 제어

프로그램의 작성과 수정작업을 하고 실행하여 실행결과를 즉시 나타내도록 한다. 8253(Programable Timer: 프로그램 가능 타이머)에서는 샘플링 순간을 알 수 있도록 매 샘플링 시간마다 인터럽트 신호를 발생시키고, PWM 신호를 출력 포트를 통해서 발생 시키게 된다. PWM 신호는 PWM 구동회로를 거쳐 증폭되며 직류 전동기가 회전하면 엔코더에서 나오는 펄스를 방향 판별 회로를 거쳐 UP, DOWN 펄스로 바꾸어 UP/DOWN 카운터로 펄스의 갯수를 계수한다. 8255(Programable Peripheral Interface: 병렬 입출력 장치)에서는 카운터에서 계수된 값을 읽어 PC에 전달하고 카운터에 초기 클리어 신호를 주며, PWM 신호 발생기의 정방향 회전과 역방향 회전 신호를 준다.

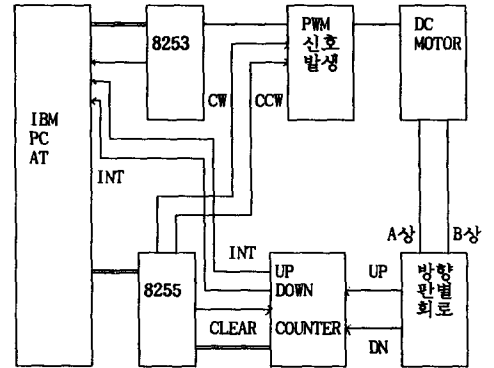


그림 3.1 위치제어 시스템의 블록선도

3.2 구동회로

본 논문에서는 직류 전동기를 구동하기 위한 증폭 회로로서 PWM(Pulse-Width Modulation)방식을 사용하였다[7]. 그림 3.2는 PWM 신호 발생회로와 PWM 증폭 회로이다. 8253 타이머의 출력 OUT1에서는 매 샘플링 주기마다 PC에서 계산된 제어 입력에 해당하는 펄스를 발생하고, 8255의 PC0와 PC1에서는 전동기의 정방향과 역방향에 대한 방향 펄스가 출력된다. PWM 증폭 회로에서 TR1가 High 이면 정방향 회전 TR2가 High 이면 역방향 회전을 한다.

3.3 위치 검출회로

본 실험에서 사용된 엔코더는 1 회전당 1000 펄스를 발생하며 회전 방향에 따라 90도의 위상 차이를 갖는 A상과 B상의 2상 펄스를 출력한다. 전동기 회전속의 위치를 알기 위해서 엔코더에서 나오는 펄스를 계수해야 한다. 그러나 회전 방향도 고려해야 하므로 그림 3.3 과 같이 A상과 B상을 조합한 방향 판별 회로를 구성하였다. 이 회로에서는 회전 방향에 따라 UP 펄스만 출력 되거나 DOWN 펄스만 출력된다. 출력된 UP/DOWN 펄스는 UP/DOWN 카운터(74LS193)에 의해서 계수된다. 그러나 카운터는 8bit로 제한되어 있으므로 계수의 범위를 벗어나게 된다. 그러므로 계수의 범위를 벗어나는 경우, 즉, Carry(CO)나 Borrow(BO)가 발생하는 경우까지 고려해서 계수를 해야한다. 그러므로 CO나 BO가 생기면 카운터에서 출

력되는 CO/BO 펄스를 인터럽트 신호로 PC에 알려줌으로써 정확한 위치 검출이 가능하도록 하였다. 계수된 결과는 매 샘플링 시각마다 8255의 포트B로 읽어져서 위치가 계산된다.

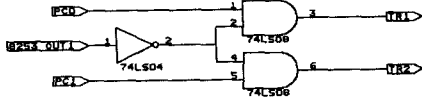


그림 3.2 (a) PWM 신호 발생회로

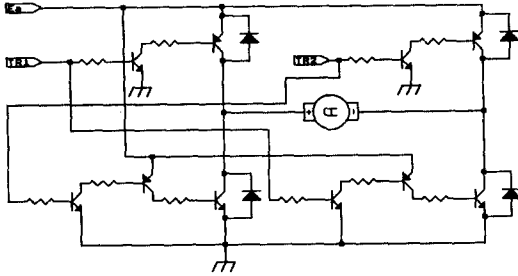


그림 3.2 (b) PWM 증폭회로

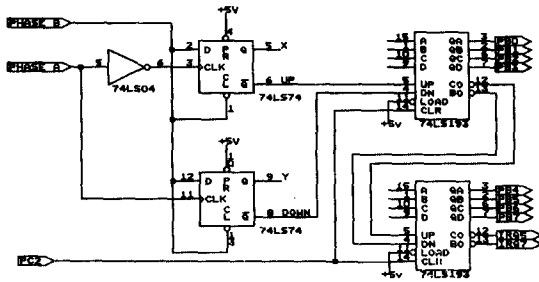


그림 3.3 방향판별회로

4. 실험결과

퍼지제어 이론을 실제 DC 전동기의 위치 제어 시스템에 적용하여 실험을 하였고, 제어 프로그램은 C-언어를 사용하였다. 본 논문에서 사용한 직류 전동기는 정격전압 75[V], 정격출력 80[W]이다. 입력변수는 오차와 오차의 변화 그리고 출력변수는 전동기에 인가되는 전압으로 하였다. 샘플링 시간은 6[msec]로 하였고, 설정치는 $\pi \sim 1.0$ [rad]으로 0.6[SEC]마다 변화하게 하여 설정치에 전동기의 회전속의 위치가 추종하도록 하였다. 그리고 오차가 큰 영역과 작은영역을 구분하여 제어규칙을 적용하였다. 비퍼지화를 거쳐 나오는 출력은 이산화 단계중의 하나의 값으로 된다. 이것을 제어입력으로 바꾸기 위해 적절한 scaling을 하였다.

오차 큰 경우 : $U = y_0 * k_1$

오차 작은 경우 : $U = y_0 * k_2$

실험에 사용한 입력 전압은 -12 ~ 12[V]이고, k_1 은 3.0, k_2 는 1.3 이다. 표1에 퍼지 제어를 위한 양자화 단계가 정의되어 있다. 양자화 단계는 -1에서 1로 정규화 하였다. 이것은 양자화 단계를 정할때의 편의성 때문이다. 사용한 언어 변수는 다음과 같다.

- PB : POSITIVE BIG
- PS : POSITIVE SMALL
- ZE : ZERO
- NS : NEGATIVE SMALL
- NB : NEGATIVE BIG

표2에 언어변수의 정의와 소속함수 테이블이 있다. 그림 4.1에 표2에 따른 소속함수를 나타내었다.

퍼지 제어기만으로 전동기를 제어하는 경우에 대해서 실험 결과를 보면, 그림 4.2에 전동기 회전속의 위치와 설정치가 나타나 있다. 과도 상태에서는 잘 추종하나 정상 상태에서는 오차가 존재하는 것을 볼수가 있다. 그림 4.3은 이에 대한 오차의 변화를 나타낸다. 역시 정상상태에서의 오차가 존재함을 볼수가 있다. 그림 4.4는 입력전압의 변화하는 모습을 나타내는데 정상상태에서 입력이 계속 인가되고 있다. 이것은, 정상상태에서 오차가 존재하므로 이것을 감소시키려고 입력전압이 2[v]정도가 인가 되지만 본 실험에서 사용한 전동기는 2[v]정도로는 회전하지 않으므로 오차가 감소하지 않게 되고 오차가 있으므로 입력전압은 다시 인가된다. 그래서 그림 4.4와 같이 정상상태에서 입력이 계속 인가된다. 표3에는 사용한 제어규칙이 나타나 있다.

다음은 퍼지 제어기와 PI 제어기를 함께 사용한 경우의 실험결과이다. 사용한 k_p 는 1.0, k_i 는 0.001이다. 그림 4.5는 설정치와 실제 전동기의 위치를 나타낸다. 전동기가 변화하는 설정치를 오차 없이 잘 따라감을 알 수 있다. 그림 4.6은 이에 대한 오차이다. 오차가 정상상태에서 충분히 작아지는 것을 나타낸다. 그림 4.7은 이 경우의 속도 변화이다. 속도 변화도 부드럽게 나타나는 것을 볼 수 있다. 그림 4.8은 입력전압의 변화이다. 이것은 오차가 없으므로 입력전압도 정상 상태에서 더 이상 인가되지 않는 것을 나타낸다.

이상 실험에서 PD 형태의 제어규칙을 갖는 퍼지제어기는 PI 제어기를 함께 사용하는 것이 정상상태 오차를 감소 시키는 데 효과적임을 확인할 수 있었다.

COARSE ERROR	FINE ERROR	CHANGE ERROR	LEVEL
-1.0 ~ -0.8	~ -0.1	~ -0.08	-4
-0.8 ~ -0.6	-0.1 ~ -0.05	-0.08 ~ -0.06	-3
-0.6 ~ -0.4	-0.05 ~ -0.03	-0.06 ~ -0.04	-2
-0.4 ~ -0.2	-0.03 ~ -0.01	-0.04 ~ -0.02	-1
-0.2 ~ 0.2	-0.01 ~ 0.01	-0.02 ~ 0.02	0
0.2 ~ 0.4	0.01 ~ 0.03	0.02 ~ 0.04	1
0.4 ~ 0.6	0.03 ~ 0.05	0.04 ~ 0.06	2
0.6 ~ 0.8	0.05 ~ 0.1	0.06 ~ 0.08	3
0.8 ~ 1.0	0.1 ~	0.08 ~	4

표1 양자화 단계

	-4	-3	-2	-1	0	1	2	3	4
NB	1	0.6	0.2	0	0	0	0	0	0
NS	0	0.6	1	0.6	0	0	0	0	0
ZE	0	0	0	0.1	1.0	0.1	0	0	0
PS	0	0	0	0	0	0.6	1	0.6	0
PB	0	0	0	0	0	0	0.2	0.6	1

표2 소속함수 테이블

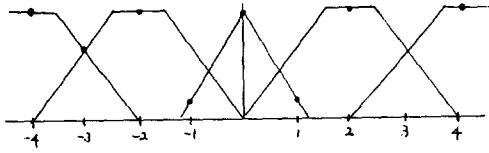


그림 4.1 소속함수

e/c	NB	NS	ZE	PS	PB
NB	NB	NB	NB	NB	NB
NS	NB	NS	NS	NS	ZE
ZE	NS	ZE	ZE	PS	PB
PS	ZE	PS	PS	PB	PB
PB	PB	PB	PB	PB	PB

(a) 오차 큰 경우 제어규칙

e/c	NB	NS	ZE	PS	PB
NB	NB	NB	NB	NS	ZE
NS	NB	NS	NS	ZE	PS
ZE	NB	NS	ZE	PS	PB
PS	NS	ZE	PS	PS	PB
PB	ZE	PS	PB	PB	PB

(b) 오차 작은 경우 제어규칙
표3 퍼지 제어기 제어규칙

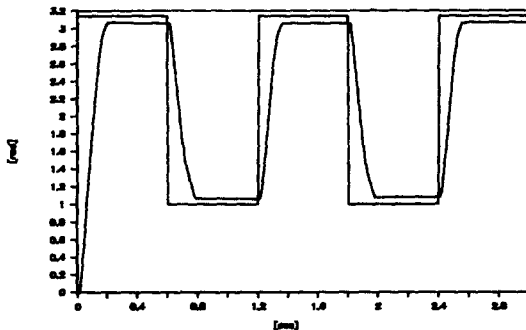


그림 4.2 퍼지 제어기 사용시 전동기 위치

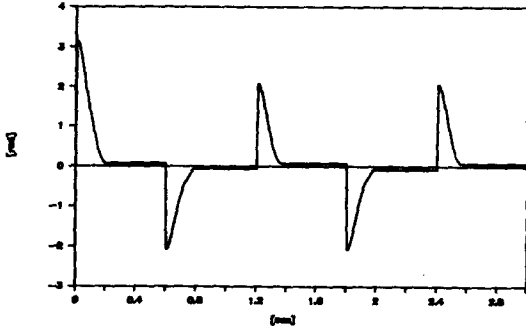


그림 4.3 퍼지 제어기 사용시 오차의 변화

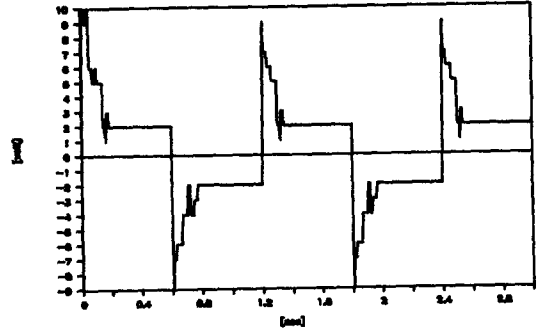


그림 4.4 퍼지 제어기 사용시 입력전압의 변화

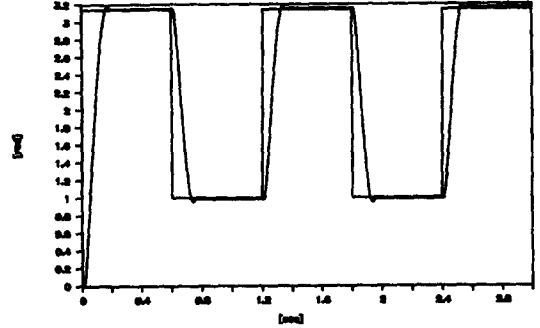


그림 4.5 퍼지와 PI 제어기 사용시 전동기 위치

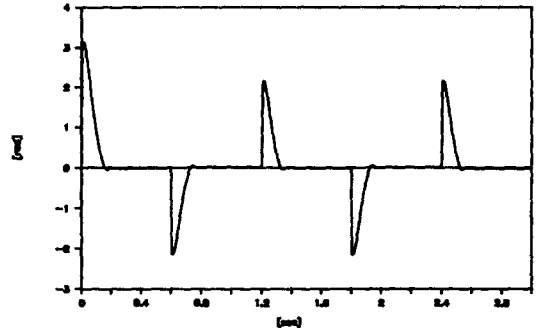


그림 4.6 퍼지와 PI 제어기 사용시 오차의 변화

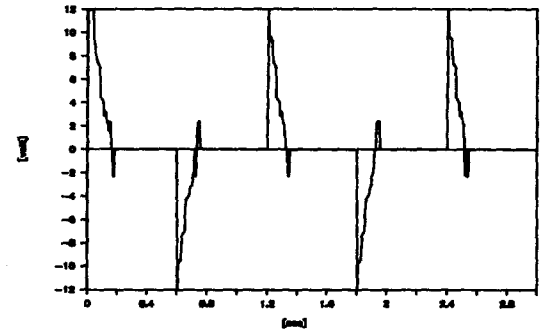


그림 4.7 퍼지와 PI 제어기 사용시 입력전압의 변화

5. 결론

본 논문에서는 퍼지이론을 DC 전동기의 위치제어 시스템에 적용하여 퍼지제어기를 실시간으로 처리하도록 구성하였다. 제어규칙이 PD 형태이고, 정상상태 오차가 존재하는 경

우 PI 제어를 퍼지제어기와 병렬로 연결하여 제어를 구성함으로써 정상상태 오차를 상당히 줄일 수 있음을 제안하고 실험으로 확인하였다. 제안된 방법은 퍼지 제어를 사용했을때 만약 정상상태 오차가 있다면 퍼지 제어기의 파라 메타를 바꾸지 않고도 부과된 PI 제어기 만으로 정상상태 오차를 줄일 수 있음을 보여준다.

참고문헌

- [1] L.A Zadadeh, "Fuzzy sets", Information & Control vol 8, pp. 338-353, 1965.
- [2] K.L.Tang, R.J.Mulholland " Comparing Fuzzy Logic with Classical controller Designs", IEEE Transactions on Systems, Man and Cybernetics, vol SMC-17, no. 6, pp1085-1087, November/December 1987.
- [3] 寺野, 菅野. "퍼지 시스템의 응용입문"
- [4] Y.F.Li, "Development of Fuzzy Algorithms for Servo Systems", IEEE Control Systems vol. 9, pp. 65-72, April 1989.
- [5] C. C. Lee, "Fuzzy Logic in control system : Fuzzy logic in controller part I", IEEE Transactions on Systems, Man and Cybernetics, vol 20, no. 2, pp404-418. 1990.
- [6] C. C. Lee, "Fuzzy Logic in control system : Fuzzy logic in controller part II", IEEE Transactions on Systems, Man and Cybernetics, vol 20, no. 2, pp404-418. 1990.
- [7] Takashi kenjo, Power Electronics for Microprocessor Age, Oxford Science Publicatins, 1990.