

퍼지로직을 이용한 권선형 유도기의 벡터제어

진 상 훈* · 김 일 환**

Vector Control of a Wound Rotor Induction Machine Using Fuzzy Logic

Sang-Hun Jin* and Eel-Hwan Kim**

ABSTRACT

This paper presents the concept and implementation of fuzzy logic control system for variable speed drive of wound rotor induction machine. Comparison of performance obtained with a conventional PI controller and fuzzy logic PI controller in speed control loop are presented. To verify the theoretical analysis, PWM inverter system with a DSP controller is built. Computer simulation and experimental results are shown in excellent agreement.

Key words : Fuzzy logic control, PI controller, PWM inverter, DSP

1. 서론

현대 산업의 발전에서 가장 주목되는 현상의 하나는 공장의 자동화라고 할 수 있는데, 산업체의 생산 공정 자동화는 가변속 운전이 가능한 전동기를 필수적으로 요구하게 된다. 이러한 가변속 운전이 요구되는 분야중에서도 특히 10[kW] 이하의 서보 응용 분야에는 종래부터 비교적 제어가 용이한 직류 전동기가 주로 사용되었다.¹⁾ 그러나, 직류기는 제어 성능과 제어 시스템이 간단하다는 장점이 있지만 교류기에 비해 유지 보수 및 브러시 마모의 단점 때문에 현재

는 서보 제어용으로 교류기가 주종을 이루고 있다. 이 때 3상 교류를 2상으로 좌표 변환하여 교류기를 제어하는 벡터 제어가 주를 이루고 있다. 이를 이용한 속도 제어에 있어서는 우수한 제어 이론들이 많이 개발되었음에도 불구하고 일반적으로 비례·적분(Proportional Integral, PI) 제어가 많이 사용되고 있다. 그 이유는 제어 구조가 간단하며 제어 대상이 시간 지연이나 비선형 특성을 갖지 않는다면 만족스러운 제어 목적을 이룰 수 있기 때문이다.

그러나 일반적인 PI 제어는 현장 적용에 있어서 예기치 않은 외란이나 부하 변동에 따라 응답 특성이 나빠진다.²⁾ 따라서 이러한 문제점들을 해결하기 위해 1965년 Zadeh 교수에 의해 처음으로 발표된 퍼지 이론을 현장에 적용하는 연구가 활발하게 이루어지고 있다. 퍼지 제어 기법은 수학적인 모델링이 없어도 전문가적인 경험적 정보를 이용하여 정량적 명제로

* 제주대학교 산업대학원

Graduate School of Industry, Cheju Nat'l Univ.

** 제주대학교 전기공학과, 산업기술연구소

Faculty of Electrical & Electronic Eng., Res. Inst. Ind. Tech., Cheju Nat'l Univ.

구성된 간단한 제어 규칙을 소프트웨어적으로 실현하여 제어하는 기법이다.³⁻⁵⁾ 이를 이용하면 복잡한 비선형이나 외란이 부가된 시스템에서도 우수한 응답 특성을 얻을 수 있고, 또한 퍼지 룩업 테이블을 적절히 이용하면 실시간 제어가 가능하다. 일반적으로 슬립 전력제어에는 권선형 유도기를 많이 사용한다. 권선형 유도기는 회전자가 권선으로 되어 있고 슬립링과 브러쉬를 통하여 외부에 접속하여 제어할 수 있지만 구조가 복잡하고 관성이 크며 회전자 온도 상승에 따른 파라메타 변동 등이 존재하여 제어 성능을 떨어뜨릴 수 있다.

따라서, 본 연구에서는 복잡한 비선형이나 외란이 부가된 시스템에서도 우수한 응답특성을 얻을 수 있는 퍼지로직 알고리즘을 도입하여 관성이 큰 권선형 유도기의 가변속 제어를 구현한다. 구현한 알고리즘의 출력특성을 검증하기 위해 일반적인 PI 제어 출력특성들을 컴퓨터 시뮬레이션과 실험을 통하여 서로 비교 분석하고자 한다.

II. 퍼지제어 시스템

2.1. 퍼지변수와 제어규칙

퍼지제어 시스템에서 퍼지제어기는 퍼지화기(fuzzifier), 제어규칙 베이스(control rule base), 추론엔진(inference engine), 비퍼지화기(defuzzifier)로 구성된다. 이에 대한 시스템 블록선도를 나타내면 Fig. 1과 같다. 여기서 G1, G2, G3는 게인값이고, e 는 기준 설정치와 출력사이의 오차를 나타내고, Δe 는 e 의 한 샘플링 구간에서의 변화분을 나타낸다.

퍼지제어기의 설계는 퍼지제어기의 입출력 변수와 그들의 언어값, 퍼지제어규칙, 수치적 입력변수의 퍼지화방법, 추론법, 비퍼지화 방법을 결정하는 것이다. 이때 먼저 입출력 변수들을 결정하고 퍼지제어 규칙과 연관된 입출력변수들의 대집합, 언어값과 소속함수를 결정하여야 한다. 입출력 변수들의 언어값의 종류는 제어규칙의 수와 밀접한 관계가 있으며 제어기의 성능에 큰 영향을 준다.

언어값의 수가 많으면 제어정책의 기술능력이 우수

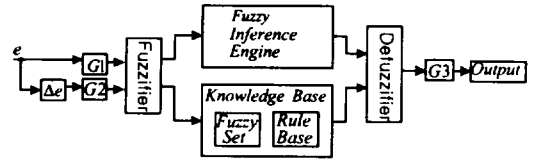


Fig. 1 Block diagram of fuzzy control system

하고 결과적으로 우수한 제어기를 구현할 수 있으나, 제어규칙의 수가 증가하여 제어기의 계산부담이 커지므로 적당하게 설정해야 한다. 본 연구에서는 7개의 언어값을 택하고 언어값의 소속함수는 삼각형의 것을 택하여 제어목적에 적합하게 사용한다. 7개의 언어값은 다음과 같다.

- NB : Negative Big
- PB : Positive Big
- NM : Negative Medium
- PM : Positive Medium
- NS : Negative Small
- PS : Positive Small
- ZE : Zero

퍼지제어기 설계에서 제일 중요한 문제는 제어규칙을 작성하는 것이다. 제어규칙을 작성하기 위해 합성에 의한 추론법을 사용할 때 제어규칙은 'If x is A then B'의 형식으로 기술된다. 제어규칙의 전제조건 변수로는 e 와 Δe 를 사용하고, 출력변수로는 조작량 u 의 변화분 Δu 를 사용한다. 설정치를 ref 라 하고, 샘플링 시각 $t = kT, k = 0, 1, 2, 3, \dots$ 에서의 출력을 $y(kT)$, 오차를 $e(kT)$, 조작량을 $u(kT)$, 조작량 변화분을 $\Delta u(kT)$ 라고 하면,

$$e(kT) = ref - y(kT) \tag{1}$$

$$\begin{aligned} \Delta e(kT) &= e(kT) - e((k-1)T) \\ &= y((k-1)T) - y(kT) \end{aligned} \tag{2}$$

$$u(kT) = u((k-1)T) + \Delta u(kT) \tag{3}$$

이다.

식(1), 식(2), 식(3)에 있는 $\Delta u(kT)$ 는 $e(kT)$ 와 $\Delta e(kT)$ 로부터 결정되므로

$$\Delta u(kT) = f(e(kT), \Delta e(kT)) \quad (4)$$

이다. $\Delta u(kT)$, $e(kT)$ 와 $\Delta e(kT)$ 의 언어값을 각각 $\Delta U(kT)$, $E(kT)$, $\Delta E(kT)$ 라 하면 식(4)는 다음식과 같이 쓸 수 있으며, 식(1)에서부터 식(5)까지의 관계를 이용하면 R 은 퍼지관계를 의미한다.

$$\Delta U(kT) = R(E(kT), \Delta E(kT)) \quad (5)$$

식(1)에서 부터 식(5)까지의 퍼지관계 R 은 식(6)과 같이 나타낼 수 있으며 이런 제어알고리즘을 속도형이라 한다.

$$R : E(kT), \Delta E(kT) \rightarrow \Delta U(kT) \quad (6)$$

식(1)에서 부터 식(6)과 같은 제어기는 E 와 ΔE 로부터 ΔU 를 추론하는 구조이기 때문에 퍼지 PI 제어기라고 한다. 이는 E 와 ΔU 의 관계가 $u(t) = K_I \int e(t)dt$ 의 적분관계에 대응되고, ΔE 와 ΔU 의 관계가 $u(t) = K_P e(t)$ 의 비례관계에 대응되기 때문이다.

2.2. $e, \Delta e, \Delta u$ 의 소속함수

측정된 입력값이 외란의 영향을 받아서 불확실하다고 간주될 때 퍼지화 작업을 통해서 입력값의 확실적인 특성을 적절한 퍼지숫자로 변환시켜 주어야 한다. 본 논문에서는 이등변 삼각형 형태의 소속함수를 이용하고 $e, \Delta e, \Delta u$ 의 언어값과 그들의 소속함수를 각각 Fig. 2와 같이 정의한다.

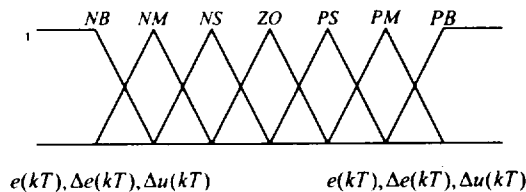


Fig. 2 Membership function of $e, \Delta e, \Delta u$

2.3. 제어규칙의 작성

전동기의 스텝속도 명령에 대한 응답특성을 관찰하여 위상공간에 표현하면 Fig. 3과 같다.

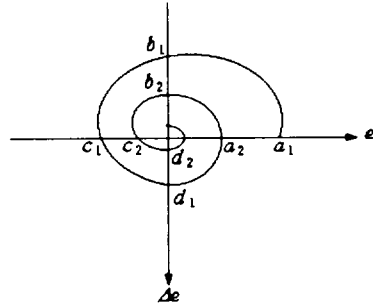


Fig. 3 Characteristics of step response in locus

제어규칙을 작성하기 위해 Fig. 3의 e 와 Δe 의 위상공간에서의 응답특성을 관찰하여 스텝응답의 특징점들 $a_i, b_i, c_i, d_i, i=1, 2, \dots$ 에서 필요한 제어규칙을 얻을 수가 있다. 한 예로 Fig. 3의 a_i 점 부근에서는 $e = positive$ 이고 $\Delta e = 0$ 이어서 오차를 줄이기 위해 가속이 요구되므로 Δu 를 $positive$ 로 하여야 한다. 따라서 a_i 점 부근에서 필요한 제어규칙 e 값은 다음 4개의 제어규칙중 하나 또는 두 개이다.

- If $e = PB$ and $\Delta e = ZE$ then $\Delta u = PB$.
- If $e = PM$ and $\Delta e = ZE$ then $\Delta u = PM$.
- If $e = PS$ and $\Delta e = ZE$ then $\Delta u = PS$.
- If $e = ZE$ and $\Delta e = ZE$ then $\Delta u = ZE$.

나머지 b_i, c_i, d_i 점에 대해서도 마찬가지로 하면 제어규칙을 작성할 수가 있다. 본 연구에서는 Table 1과 같이 49개의 제어규칙을 작성하였고, 가로는 Δe , 세로는 e 이고 표안의 값은 Δu 의 언어값이다.

2.4. 추론법

입력변수 e 와 Δe 의 퍼지화 방법으로 Mamdani의 추론법을 채택할 때 추론 알고리즘은 다음과 같다.

Table 1. Rule base

$\Delta e \backslash e$	NB	NM	NS	ZE	PS	PM	PB
NB	NB	NB	NB	NB	NM	NS	ZE
NM	NB	NB	NB	NM	NS	ZE	PS
NS	NB	NB	NM	NS	ZE	PS	PM
ZE	NB	NM	NS	ZE	PS	PM	PB
PS	NM	NS	ZE	PS	PM	PB	PB
PM	NS	ZE	PS	PM	PB	PB	PB
PB	ZE	PS	PM	PB	PB	PB	PB

Table 1에 있는 i 번째 제어규칙 R_i 는 다음식으로 나타낼 수 있다.

$$R_i : \text{If } e \text{ is } A_i \text{ and } \Delta e \text{ is } B_i \text{ then } \Delta u \text{ is } C_i \quad (7)$$

$$\mu_{A_i}(e) = \begin{cases} 1, & \text{if } e = e^0 \\ 0, & \text{otherwise} \end{cases}$$

$$\mu_{B_i}(\Delta e) = \begin{cases} 1, & \text{if } \Delta e = \Delta e^0 \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

i 번째 제어규칙 R_i 에 의한 근사추론 결과를 C_i . C_i 를 삼각형의 무게중심법(Center of Gravity, COG)으로 비퍼지화한 값을 Δu_i . 전체 제어규칙에 의한 비퍼지화 추론결과를 Δu 라 하면 Δu_i 와 Δu

는 다음과 같이 구한다. 입력이 $(e, \Delta e) = (e^0, \Delta e^0)$ 일 때 e^0 와 Δe^0 의 퍼지값 A 와 B 를 각각 퍼지싱글턴으로 정의하면, i 번째 제어규칙 R_i 의 적합도 ω_i 는 다음식으로 계산되며,

$$\omega_i = \mu_{A_i}(e^0) \wedge \mu_{B_i}(\Delta e^0) \quad (9)$$

i 번째 제어규칙 R_i 에 의한 퍼지추론결과 C_i 는 다음식으로 구하고,

$$\mu_{C_i}(\Delta u) = \omega_i \wedge \mu_{C_i}(\Delta u) \quad (10)$$

C_i 를 삼각형의 무게중심법으로 비퍼지화한 추론결과 Δu_i 는 다음과 같이 계산된다.

$$\Delta u_i = COG(C_i) \quad (11)$$

2.5. 비퍼지화

퍼지추론한 결과를 제어기에 입력하기 위해서는 크리스프한 값으로 변환하여야 한다. 삼각형의 무게중심법을 사용하여 Table 1에 있는 49개의 제어규칙에 의한 비퍼지화된 추론결과는 다음식으로 구한다.

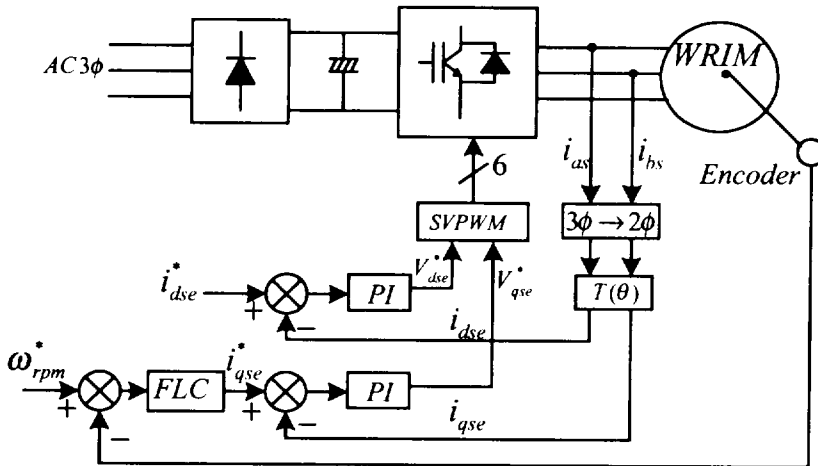


Fig. 4 Block diagram of fuzzy PI control system

$$\Delta u = \frac{\sum_{i=1}^4 \omega_i \Delta u_i}{\sum_{i=1}^4 \omega_i} \quad (12)$$

2.6. 퍼지 PI 제어 구동시스템

Fig. 4는 권선형 유도기를 가변속 제어하기 위한 퍼지 PI 제어기의 제어시스템 블록도이다.

이 시스템은 이중 제어루프를 가지고 있다. 하나는 전류제어를 위한 일반적인 PI제어 루프이고 다른 하나는 속도오차에 해당하는 토크분 전류 지령치 i_{qse} 를 만들어 내는 퍼지 PI제어 루프이다. 속도제어를 위한 퍼지 PI 제어 루프에서 속도오차와 속도오차 변화분에 대해 Table 1에 따른 제어 규칙값을 선택하게 된다. 이 때 실제 구동에 사용한 제어규칙은 Table 2와 같다. Table 2에 나타난 값들은 Table 1을 기초로 하여 얻어진 것이며 또한 Fig. 1의 G1, G2, G3의 게인값들과 전류제어 루프의 게인들도 튜닝에 의해 알맞은 값을 택한다.

Table 2. Rule base

$e \backslash \Delta e$	-3	-2	-1	0	1	2	3
-3	-3	-3	-3	-2	-2	-1	0
-2	-3	-3	-2	-2	-2	0	1
-1	-3	-2	-2	-1	0	1	3
0	-2	-2	-1	0	1	2	3
1	-2	-1	0	1	2	2	3
2	-1	0	1	2	2	3	3
3	0	1	2	2	3	3	3

III. 컴퓨터 시뮬레이션

시스템을 설계하고 구동하기 앞서 설계한 시스템의 출력특성을 분석하기 위하여 일반적으로 컴퓨터 시뮬레이션을 행한다. 전력전자 시스템 해석을 위해 많이 사용되는 Psim프로그램으로 권선형 유도기의 벡터제어를 위한 시스템 제어 블록선도를 Fig. 5와 같이 만

들어 Table 3과 같은 파라메타를 갖는 모델 전동기로 퍼지로직에 의한 속도제어 응답특성을 해석하였다.

Table 3. Parameters of model motor

Rated Power	5 hp	R_s	0.36 Ω
Rated Voltage	220 V	L_s	0.04633 H
Rated Current	13.8 A	R_r	0.04423 Ω
Rated Speed	1789 rpm	L_r	0.04633 H
J	0.0365 kg-m ²	L_m	0.04423 H
Pole Number	4		

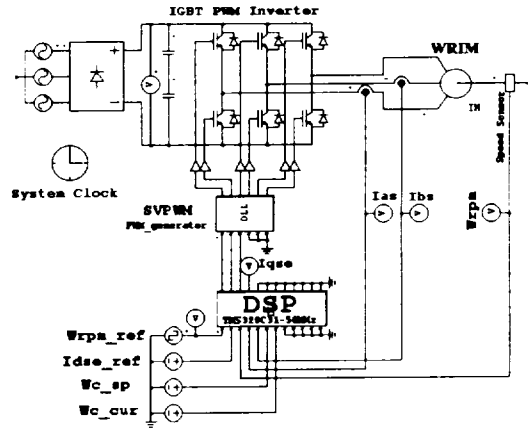


Fig. 5 Schematic diagram of Psim simulation program

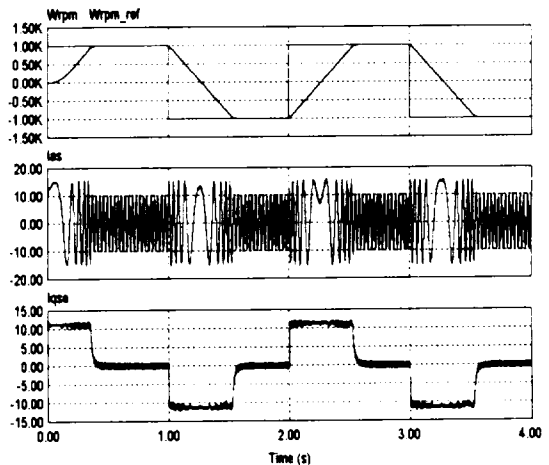


Fig. 6 Simulation results of the conventional PI speed control($\omega_{rpm_ref} = \pm 1000(\text{rpm})$)

Fig. 6은 일반적인 PI 속도제어시 속도명령으로 $\pm 1000[\text{rpm}]$ 을 인가하였을 때의 속도 명령값, 속도 응답, 전동기 a상전류, 그리고 동기좌표계의 q축 전류를 나타내고 있다. 여기서 샘플링 시간은 속도제어루프에서는 $1[\text{ms}]$, 전류제어 루프에서는 $100[\mu\text{s}]$ 로 하고 과전류 방지를 위해 전류제한기를 최대 $15[\text{A}]$ 로 제한 하였다. 그리고 자속분 전류 i_{dse} 는 $10[\text{A}]$ 로 유지시켰다. 이 결과에서 보면 $0[\text{rpm}]$ 에서 $1000[\text{rpm}]$ 까지 도달하는 시간은 대략 $0.4[\text{sec}]$ 정도 걸리고 있음을 알 수 있다. 또한 정상상태 도달시 오버슈트가 없

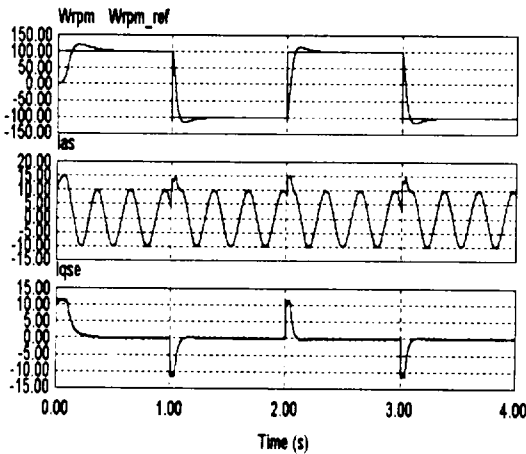


Fig. 7 Simulation results of the conventional PI speed control(\bullet rpm_ref = $\pm 100[\text{rpm}]$)

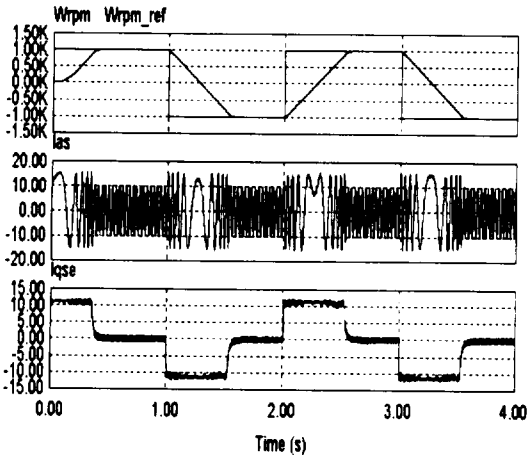


Fig. 8 Simulation results of fuzzy PI speed control(\bullet rpm_ref = $\pm 1000[\text{rpm}]$)

고 역방향 회전시도 속도 명령값에 잘 추종하고 있음을 알 수 있다.

Fig. 7은 Fig. 6에서 구동할 때 사용한 PI 제인을 가지고 속도명령을 $\pm 100[\text{rpm}]$ 으로 운전하였을 때의 출력특성을 나타내고 있다. 이 파형을 관찰해 보면 정상상태에 도달할 때 오버슈트가 크게 나타나고 있음을 알 수 있다.

Fig. 8은 퍼지로직을 이용한 $\pm 1000[\text{rpm}]$ 속도명령 제어시 출력을 나타내고 있다. 이 파형에서 보면 $0[\text{rpm}]$ 에서 $1000[\text{rpm}]$ 까지 도달하는 시간이 약 $0.4[\text{sec}]$ 이고 정상상태에 도달할 때 오버슈트 없이 제대로 잘 동작하고 있음을 알 수 있다. Fig. 9는 Fig. 8과 같은 조건에서 $\pm 100[\text{rpm}]$ 명령을 주었을 때 출력특성을 나타내고 있다. 이 파형에서 보면 오버슈트 없이 속도명령값에 잘 추종하고 있음을 알 수 있다. 컴퓨터 시뮬레이션 결과들을 분석해 보면 $\pm 1000[\text{rpm}]$ 속도명령시는 일반적인 PI제어나 퍼지로직 제어에서 나타나는 출력특성은 큰 차이가 없지만 저속운전 $\pm 100[\text{rpm}]$ 에서는 퍼지로직을 이용한 제어특성이 우수하게 나타나고 있음을 알 수 있다.

IV. 실험결과

제안한 해석방법을 검증하기 위해 3상 IGBT 전압

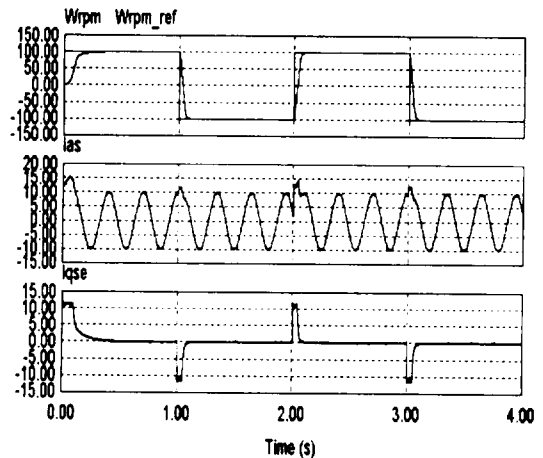


Fig. 9 Simulation results of fuzzy PI speed control(\bullet rpm_ref = $\pm 100[\text{rpm}]$)

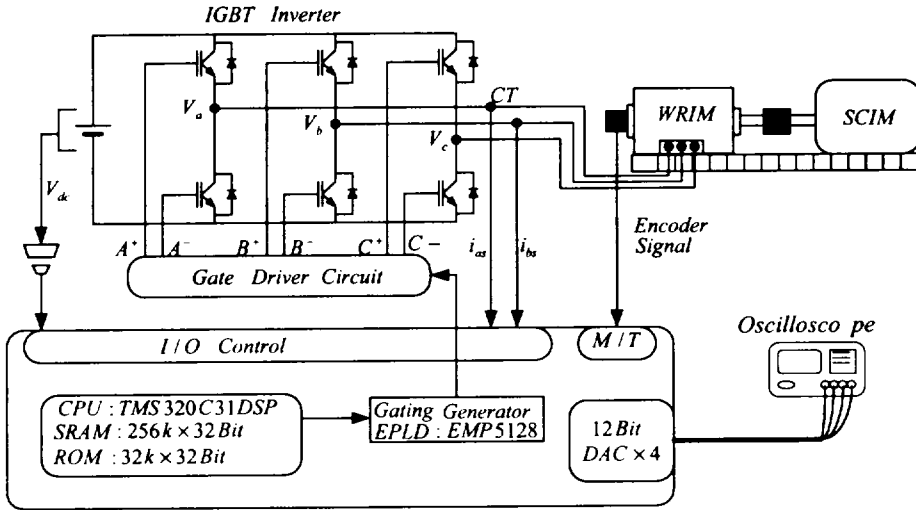


Fig. 10 Schematic diagram of experimental system

원 인버터와 TMS320C31 DSP를 이용하여 Fig. 10과 같은 형태의 제어시스템으로 실험을 행하였다. 컴퓨터 시뮬레이션에서 이용된 것과 같은 정수를 갖는 권선형 유도기를 택하여 구동시 속도제어루프는 1[ms], 전류제어루프는 100[us]의 샘플링 시간으로 제어를 행하였다. Fig. 11은 일반적인 PI속도제어시 정역운전에 대한 응답특성을 나타내고 있다. 이 파형을 관찰해 보면 $\pm 1000[\text{rpm}]$ 의 운전시는 오버슈트 없이 제어가 잘 되고 있지만 $\pm 100[\text{rpm}]$ 에서는 오버슈트가 나

타고 있음을 알 수 있다. Fig. 13은 Fig. 11과 같은 조건에서 퍼지로직을 이용하여 제어를 행한 결과이다. 이 파형에서 보면 Fig. 11과 거의 같은 응답특성을 나타내고 있다. Fig. 14는 Fig. 12와 같은 조건하에서 운전된 결과를 나타내는 것으로 속도응답은 오버슈트 없이 기준 명령값에 잘 추종되고 있음을 알 수 있다.

이상과 같은 실험결과들을 종합해보면 일반적인 PI제어시보다 퍼지로직을 이용한 PI제어시가 저속에서 더 우수한 응답특성을 나타내고 있다. 이것은 일

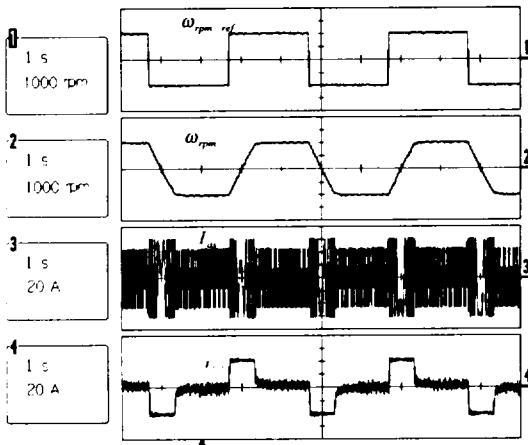


Fig. 11 Experimental results of the conventional PI speed control ($\omega_{rpm_ref} = \pm 1000[\text{rpm}]$)

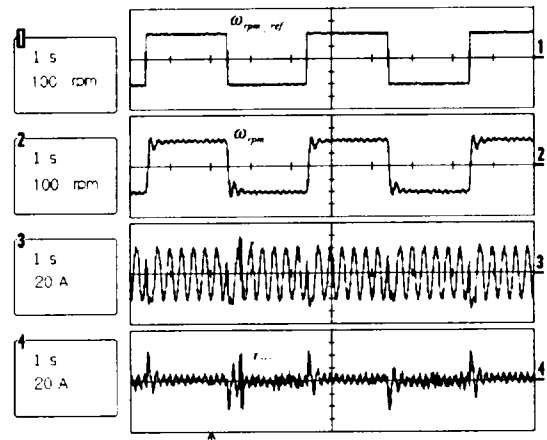


Fig. 12 Experimental results of the conventional PI speed control ($\omega_{rpm_ref} = \pm 100[\text{rpm}]$)

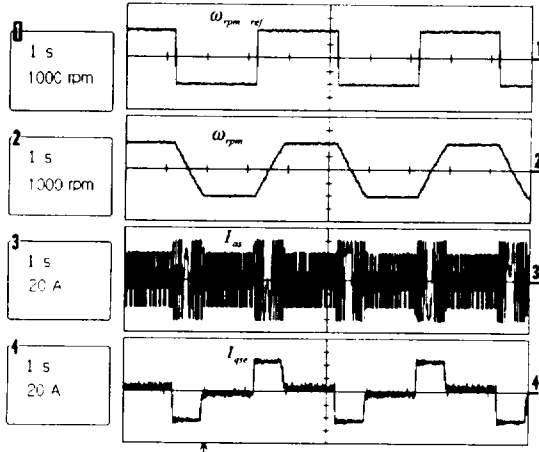


Fig. 13 Experimental results of fuzzy PI speed control system ($\omega_{rpm_ref} = \pm 1000[\text{rpm}]$)

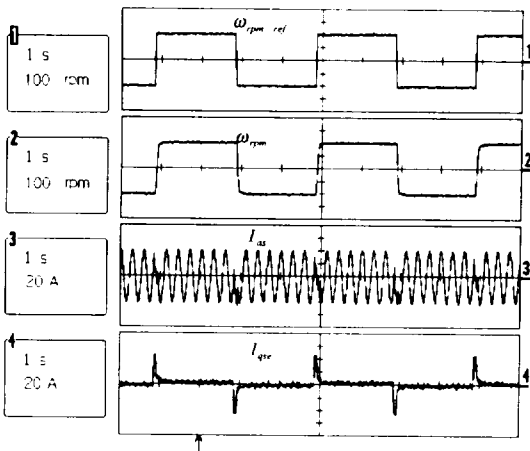


Fig. 14 Experimental results of fuzzy PI speed control system ($\omega_{rpm_ref} = \pm 100[\text{rpm}]$)

반적인 PI제어시에는 속도명령값이 변하면 응답특성도 나빠지는 것을 나타내고 있다.

V. 결론

제어시스템이 비선형성이 강할 경우 적용제어 같은 현대제어이론을 도입하여 제어하여야 한다. 이 때 시스템의 정확한 수학적 모델링이 전개되어야 하며 또한 제어가 복잡하여 진다. 반면에 인간의 전문가적

경험을 바탕으로 한 퍼지제어방식은 제어이론이 간단하고, 정확한 시스템의 수학적 모델링이 필요하지 않아 외란과 비선형에 강한 제어성능을 가지고 있다.

따라서 본 논문에서는 일반적인 PI제어 방법과 퍼지로직을 이용한 PI제어방법을 Psim프로그램을 이용하여 컴퓨터 시뮬레이션을 수행하였고, 실험적 검증을 위하여 IGBT PWM인버터 및 TMS320C31 DSP를 이용하여 컴퓨터 시뮬레이션과 같은 조건으로 실험을 행하였다. 그 결과 $\pm 1000[\text{rpm}]$ 의 속도 명령값에 대한 출력특성들에서는 일반적인 PI제어때나 퍼지 PI제어시에도 큰 차이가 없지만 저속영역 $\pm 100[\text{rpm}]$ 운전 영역에서는 일반적인 PI제어시 오버슈트가 나타나고 있지만 퍼지로직을 이용한 PI제어시는 응답특성이 아주 우수하게 나타나고 있음을 알 수 있었다. 이상의 결과들을 종합해 보면 기존의 PI제어보다 퍼지 PI제어가 속도 명령변화에도 잘 추종하고 있어, 비선형성이 강한 시스템에 적용하면 우수한 운전특성을 얻을 수 있을 것으로 사려된다.

참고 문헌

- 1) 윤덕용, "공간전압벡터 PWM기법을 이용한 영구자석형 동기 전동기의 속도 제어", 단국대학교 학위 논문, 1995.
- 2) Karl J. Astrom Bjorn Wittenmark, 1984, "Computer Controlled Systems Theory and Design", Prentice-Hall.
- 3) L. A. Zadeh, 1968, "Fuzzy algorithm", Information and Control, vol. 12, pp.94-102.
- 4) E. H. Mamdani, 1974, "Applications of fuzzy algorithms for control of simple dynamic plant", Proc. IEEE, vol. 121, No. 12, pp.1585-1588.
- 5) T. Yamazaki and M. Sugeno, 1984, "Fuzzy control", SYSTEMS AND CONTROL, vpl. 28, No 7, pp.42-446.
- 6) V. Miranda and J. T. Saraiva, 1992, "Fuzzy Modeling of Power Systems Optimal Load Flow", IEEE Trans. on Power Systems, vol. 7, No. 2, pp. 843-849.
- 7) B. K. Bose, 1991, "Power Electronics and AC Drives", Prentice-Hall.