

# 적응제어 기법을 이용한 IPMSM의 속도제어

이홍균 · 이정철 · 정택기 · 이영실 · 정동화

순천대학교 전기제어공학과

## 1. 서 론

PMSM 드라이브의 벡터제어를 위한 속도제어기는 일반적으로 PI 알고리즘을 사용한다.[1] PI 알고리즘에서는 플랜트의 동적 특성을 동정하기 위하여 연속적인 측정이 가능한 플랜트 파라미터의 자동동조 또는 적응 알고리즘을 추가하여 사용하기도 한다. 그러나 PI 제어기는 플랜트의 변동이 발생하여 연속적으로 적응추종을 해야할 경우, 계산이 매우 복잡하고 응답특성이 저하된다. 이러한 시스템에 양호한 성능을 얻기 위한 연구로 적응제어가 연구되었으며 적응제어는 종래의 PI 제어기에 비해 매우 양호한 성능을 얻을 수 있었다. 최근 적응제어는 STC(Self Tuning Control), MRAC(Model Reference Adaptive Control), SMC(Sliding Mode Control) 및 EKF(Extended Kalman Filter) 등의 다양한 기법들이 제시되었다.[2] 이러한 적응제어 기법은 수학적인 모델링에 기초를 두고 있으며 다량의 알고리즘 때문에 매우 복잡하다.

이러한 문제점을 해결하기 위하여 종래의 PI 제어기에 퍼지제어기를 적용하였으며[3], PID 제어기에 퍼지제어를 혼합하여 사용하였다.[4] 퍼지동작은 기준 근방의 동작점을 근사화시키고 PID 동작은 채터링 퍼지의 영향에 의한 오차를 제거하기 위하여 사용하였다. 또한, 부동 소속함수, 퍼지를 변화 및 초안정도 이론에 의한 시스템의 이득변화의 기법이 제시되었다. 이러한 적응기법은 MRAC에 기초하며 프로세서에 적응제어를 수행하기 위하여 단지 직접 퍼지제어기만을 사용하였다.

본 연구에서는 PMSM 드라이브의 고성능을 위하여 MRAC 기법을 응용한 적응 퍼지제어기를 제시한다. 적응 퍼지제어기는 적응제어와 종래의 직접 퍼지제어의 문제점을 보완하여 종래의 퍼지제어기에 기준 모델을 추가하여 MRAC 형태의 퍼지제어기를 구성한다. 적응 퍼지제어기는 2개의 퍼지제어의 블록으로 동작한다. 하나의 블록은 종래에 사용했던 직접 퍼지제어기이며 PMSM 드라이브의 속도제어를 위하여 고려한다. 다른 블록은 기준 모델을 사용하고 퍼지추론을 근거로 하여 퍼지적응 메카니즘을 구현한다. 이 두 블록의 출력인 제어신호의 변화를 합하여 전체 제어신호의 변화로 사용한다. 종래의 직접 퍼지제어기와 본 논문에서 제시한 적응 퍼지제어기의 응답특성의 결과를 비교한다. 본 논문에서 제시한 적응 퍼지제어기는 종래의 직접 퍼지제어기 보다 응답특성이 매우 양호하게 나타났으며 그 결과들을 제시한다.

## 2. 퍼지제어기

그림 1은 벡터제어 PMSM 드라이브의 퍼지 속도제어기의 블록도를 나타낸다.

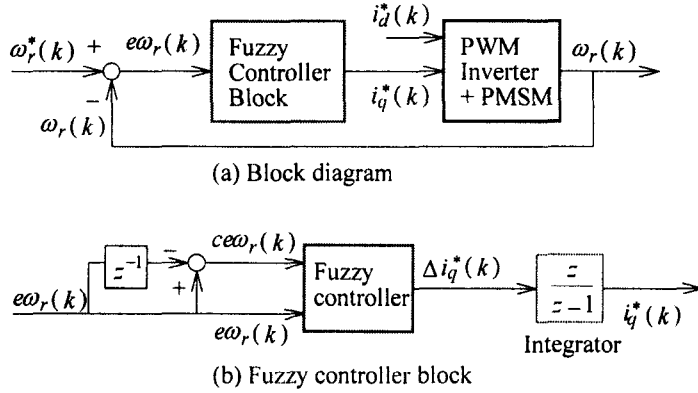


그림 1 벡터제어 PMSM의 퍼지 속도제어기

퍼지제어기로 선정되어 입력되는 가장 중요한 변수는 속도오차와 속도오차의 시간 변화분이다. 두 개의 입력변수  $e\omega_r(k)$ 와  $ce\omega_r(k)$ 이며 매 샘플링 순간에서 다음과 같이 계산된다.

$$e\omega_r(k) = \omega_r^*(k) - \omega_r(k) \quad (1)$$

$$ce\omega_r(k) = e\omega_r(k) - e\omega_r(k-1) \quad (2)$$

여기서  $\omega_r^*(k)$ 는 지령속도이며  $\omega_r(k)$ 는 실제 회전자 속도이다.

퍼지제어기는 퍼지화, 룰 평가 및 비퍼지화의 3단계로 나눈다. 퍼지화 단계에서 보통 변수  $e\omega_r(k)$ 와  $ce\omega_r(k)$ 는 삼각형 소속함수를 사용하여 퍼지변수  $e\omega_r$ 와  $ce\omega_r$ 로 변환한다. 간접 벡터제어 시스템에 적용하는 지령전류  $i_q^*(k)$ 는 다음과 같다.

$$i_q^*(k) = i_q^*(k-1) + \Delta i_q^*(k) \quad (3)$$

출력변수  $\Delta i_q^*(k)$ 의 보통 값은 중심 비퍼지화의 방법을 이용하여 구할 수 있다.

$$\Delta i_q^*(k) = \frac{\sum_{i=1}^n (\Delta i_q^*)_i \mu[(\Delta i_q^*)_i]}{\sum \mu[(\Delta i_q^*)_i]} \quad (4)$$

### 3. 적응 퍼지제어기

적용 퍼지제어기는 그림 2와 같고 퍼지제어 시스템에 적용되는 MRAC 형태를 나타내는 블록도이다. 기준모델과 퍼지적용 메카니즘의 2개의 중요한 블록은 종래의 퍼지제어기에 추가된다. 기준모델은 상승시간과 오버슈트와 같은 설계기준을 만족시키는 요구성능을 설정하는데 사용한다. 기준입력  $r$ 는 기준모델과 퍼지제어기에 동시에 공급된다. 입력  $r$ 는 지령속도  $\omega_r^*(k)$ 이며 플랜트는 PWM 인버터와 PMSM이다. 시스템의 실제출력은 기준모델의 출력과 비교한다. 결과적인 오차와 오차의 변화는 FLAM(Fuzzy Logic Adaptation Mechanism)에 적용되며 이는 기준모델과 같이 동작하는 시스템을 강제화하기 위하여 FLC(Fuzzy Logic Controller)의 특성을 수정한다.

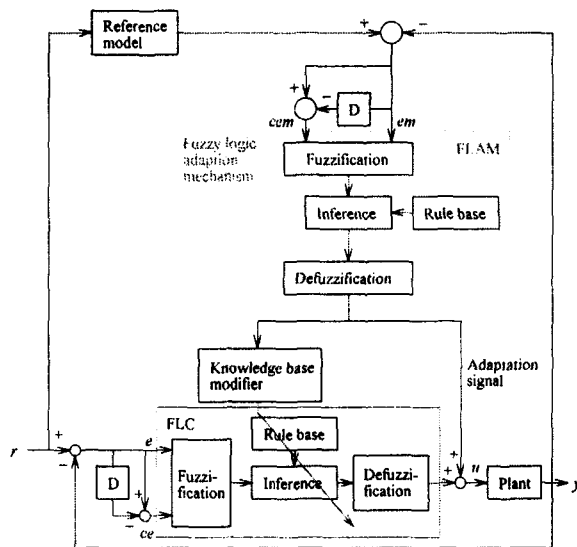


그림 2 MRAC 형태의 적응 퍼지제어기 시스템

적용제어는 파라미터 적응과 신호 적응의 두 방법에 의해 실현될 수 있다. 파라미터 적응방법에서는 FLC의 지식 베이스가 소속함수와 규칙의 특성 또는 퍼지집합의 토폴로지 등을 변화시켜 수정한다. FLAM의 출력신호는 지식 베이스 수정자를 위한 입력 신호로 사용되고 이 기능은 플랜트 입력  $u$ 에서 요구되는 변화를 발생하기 위하여 지식 베이스를 수정한다. 지식 베이스는 정규구간 동안 새롭게 수정하여 시스템의 메모리에 저장된다. 이 방법을 위하여 FLC는 자기진단을 하며 경험보다 더 우수한 내용을 학습한다. 그러나 지식 베이스가 필요한 정보의 양은 과대하기 때문에 구현하기는 복잡하다. 신호 적응방법은 적응 메카니즘은 보조 제어신호를 발생시킨다. 이 신호는 동작의 변화와 부하조건에 의해 발생하는 성능의 오차를 보상하기 위하여 FLC 출력에 첨가한다. 이 메카니즘은 학습능력을 제공하지 않으며 지식 베이스의 수정이 필요 없게 되어 구현은 매우 간단하다.

FLAM의 내부구조는 퍼지화, 규칙수행 및 비퍼지화 등 직접 FLC와 동일하다. FLC와 FLAM 제어 규칙에 대하여 드라이브 지식과 일반적인 개념에 기초하여 형성한다. 기준모델은 요구되는 제어성능을 위한 성능목표로 나타나고 적응 메카니즘이 실제출력과 기준모델 출력사이의 오차를 최소화로 강제화할 수 있다면 요구되는 제어성능을 얻을 수 있다.

그림 3은 기준모델에 의한 적응 퍼지 메카니즘을 나타낸다. 적응 퍼지 메카니즘의 루프는 종래의 퍼지제어기 루프와 병렬로 연결된다.

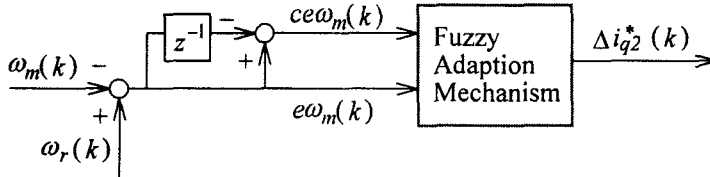


그림 3 기준모델에 의한 적응 퍼지 메카니즘

적응 메카니즘에 의한 출력은  $\Delta i_{q2}^*(k)$ 를 발생시킨다.  $\Delta i_{q1}^*(k)$ 과  $\Delta i_{q2}^*(k)$ 를 합하여  $\Delta i_q^*(k)$ 를 구하여 이를 적분기를 통해  $i_q^*(k)$ 를 발생시켜 플랜트에 인가한다.

적응 퍼지 메카니즘의 입력은 기준모델의 출력과 실제속도의 오차( $e\omega_m(k)$ )와 오차의 변화분( $ce\omega_m(k)$ )을 사용한다. 오차와 오차의 변화분은 다음과 같다.

$$e\omega_m(k) = \omega_m(k) - \omega_r(k) \quad (5)$$

$$ce\omega_m(k) = e\omega_m(k) - e\omega_m(k-1) \quad (6)$$

위의 두 입력은 적응 퍼지의 룰 베이스에 의해 지령 q축 전류에 보상되는  $\Delta i_{q2}^*(k)$ 가 발생되고  $\Delta i_{q1}^*(k)$ 에 추가된다. 그러므로 퍼지제어기의 지령 q축 전류는 다음과 같다.

$$i_q^*(k) = i_q^*(k-1) + [\Delta i_{q1}^*(k) + \Delta i_{q2}^*(k)] \quad (7)$$

$e\omega_m, ce\omega_m$  및  $\Delta i_{q2}^*(k)$ 의 전체집합은 각각 (-180, 180) [rad/s], (-7, 7) [rad/s] 및 (-1, 1) [A]로 설정한다.

#### 4. 결과 및 고찰

본 논문의 샘플링 주기를  $T_s = 1[msec]$ 로 하였으며 기준모델에서 1차 지연요소의 시정수는  $\tau = 120[msec]$ 로 설정하였다. 그림 4는 다양한 부하토크에 대한 지령속도의 스텝변화에 대한 응답특성을 나타낸다. 산업계의 제어시스템에서 빈번하게 요구되는 부하토크인  $T_L = K_0 + K_1\omega_r + K_2\omega_r^2$ 를 인가한다. 본 논문의 적응 퍼지제어기를 사용하였으며 부하토크 중 속도에 비례하는 상수  $K_1$ 을 변화시키며 그림 (a)  $K_1 = 0.001$ , (b)  $K_1 = 0.02$  및 (c)  $K_1 = 0.1$ 에 대한 응답특성이다. 부하가 커지면 오버슈트 및 채터링이 거의 나타나지 않지만 안정화 시간은 길어진다.

그림 5는 다양한 샘플링 시간에 대한 지령속도의 스텝변화에 대한 응답특성을 나타낸다. 적응 퍼지제어기를 사용하였으며 그림 (a)는  $T_s = 1[msec]$ , (b)  $T_s = 0.5[msec]$  및 (c)  $T_s = 100[\mu sec]$ 에 대한 응답특성이다. 샘플링 시간이 작아지면 속도는 오버슈트 및 채터링이 거의 나타나지 않으며  $q$ 축 전류는 증가하고 빠르게 정상상태에 도달한다. 그림 6은 (a)는 관성 모멘트  $3J_n$ , (b)는 부하토크의 정수  $K_1 = 0.02$  및 (c)는 샘플링 시간  $T_s = 0.4[msec]$ 일 경우의 PI, 직접 퍼지 및 적응 퍼지제어기의 응답특성을 비교한다. 그림 (a)에서 관성이 증가하면 PI와 직접 퍼지제어기에서는 오버슈트와 상승시간이 증가하며 적응 퍼지제어기는 상승시간은 약간 길어지지만 오버슈트가 감소되므로 빠르게 정상상태에 도달한다. 그림 (b)에서 부하가 증가하면 오버슈트는 감소되지만 정상상태에 느리게 정상상태에 도달한다. 그림(c)에서 샘플링 시간이 짧아지면 직접 퍼지 및 적응 퍼지제어기는 정상상태에 빠르게 추종하지만 PI 제어기는 채터링이 증가하여 안정화 시간이 길어진다. 따라서 적응 퍼지제어기가 종래의 PI 및 직접 퍼지제어기 보다 응답특성이 매우 양호하게 나타나고 있다.

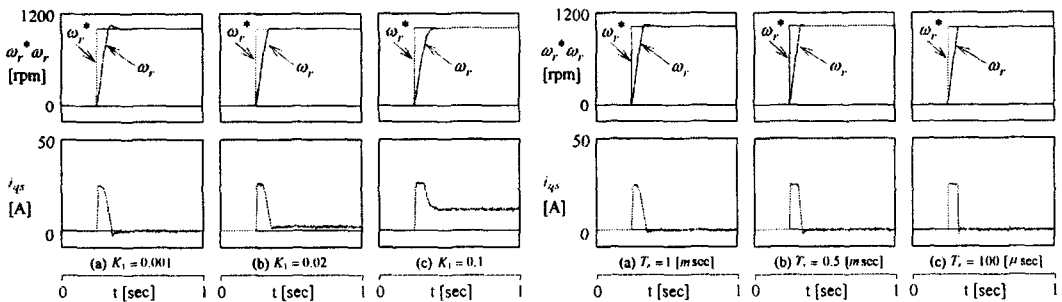


그림 4

그림 5

그림 4 다양한 부하토크에 대한 지령속도의 스텝변화에 대한 응답특성

그림 5 다양한 샘플링 시간에 대한 지령속도의 스텝변화에 대한 응답특성

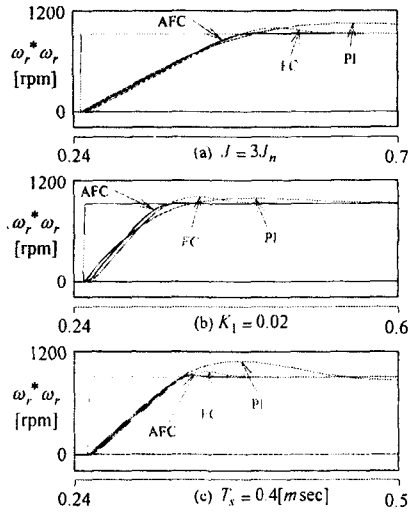


그림 6 다양한 파라미터에 대한 응답특성의 비교

## 6. 결론

본 논문에서는 PMSM 드라이브의 고성능 제어를 위하여 퍼지추론을 이용한 적응 퍼지제어기를 제시하였다. 제시한 퍼지제어기는 기준모델에 기초한 적응 퍼지제어기를 종래의 직접 퍼지제어기에 추가하고 직접 퍼지제어기에서 사용한 퍼지논리를 응용하였다.

적응 퍼지제어기는 PMSM의 지령속도의 변화와 부하 변화에 대하여 종래의 직접 퍼지제어기 보다 양호한 응답특성이 나타났다. 그리고 관성 증가에 따른 동작조건의 변화에 적응 퍼지제어기가 오버슈트 및 채터링이 매우 작게 나타났으며 안정화 시간도 양호하게 나타났다. 본 논문에서 제시한 적응 퍼지제어기를 종래의 PI, 직접 퍼지제어기 보다 응답특성이 매우 우수하게 나타났으며 제시한 적응 퍼지제어기의 타당성을 입증할 수 있었다.

## 참고문헌

- [1] B. K. Bose, "Power electronics and AC drives," *Englewood Cliffs, Ed. Prentice Hall*, 1986.
- [2] K. J. Astrom and B. Wittenmark, "Adaptive control," *Addison-Wesley*, 1989.
- [3] D. H. Chung, et al., "Efficiency optimization control of induction motor system using fuzzy control," *KIEE*, vol. 50-D, no. 7, pp. 318-324, 2001.
- [4] S. Tzafestas, N. P. Papanikolopoulos, "Incremental fuzzy expert PID control," *IEEE Trans, IA*, vol. 37, no. 5, pp. 365-371, 1990.