

유전자 알고리즘을 이용한 유도전동기 Cascade PI 제어기 설계

이학주, 권성철, 양승권, 한승호
한전 전력연구원

Design of Cascade PI Controller for Induction Motor Drives using Genetic Algorithm

H.J. Lee, S.C. Kwon, S.K. Yang, and S.H. Han
Korea Electric Power Research Institute (KEPRI)

ABSTRACT

In this paper, we describe a design procedure for cascade controller for induction motor drives based on Genetic Algorithms(GAs). Most electric drives have two separate controllers for current and speed control, which are in general designed in two consecutive steps(firstly the current controller and then the speed controller). We search simultaneously for the couple of discrete anti-windup controllers achieving the optimal compromise of weighted cost and performance indices related to both current and speed responses.

1. 서 론

산업현장의 동력원으로 사용되고 있는 유도전동기 제어는 전력전자 소자 및 DSP의 발전과 더불어 벡터제어기법의 개발로 타여자 직류전동기에 필적할 수 있는 성능을 갖게 되었다. 유도전동기 벡터 제어 시스템에서 일반적으로 적용되는 PI제어기는 제어이득이 적절한 조절이 가능한 경우 좋은 운전 특성을 얻을 수 있고, 제어알고리즘이 간단하여 구현이 용이하다. 그러나 운전조건 변동 또는 전동기 파라미터의 변동 시 최적 운전상태를 유지하기 위하여 제어이득을 동조하여야 한다. 특히 유도전동기는 비선형 특성을 가지므로 기존의 선형 PI제어기로 만족할 만한 제어특성을 기대하기 어렵다. 이러한 문제를 해결하기 위하여 적응제어 및 지능제어에 대한 연구가 활발하게 진행되고 있지만 적응제어는 제어 환경변화에 대응은 가능하나 설계방법이 까다롭고 제어 알고리즘이 복잡하여 많은 계산이 요구됨에 따라 연산시간이 길고 적응시간이 느려 시정수가 작은 시스템에 적용의 어려움이 발생

한다. 본 논문에서는 유도전동기 제어시스템에 적용한 PI제어기를 유전자 알고리즘에 의한 제어이득 온라인 동조로 운전조건 또는 파라미터 변동에 만족할 만한 제어특성을 얻고자 한다. 참고문헌[1]은 제어 성능지수의 가중치 합을 적합도 함수로 이용하여 혼합 GA에 의한 직류전동기 제어시스템의 캐스케이드 제어기를 설계하였으며, 참고문헌[2]에서는 유전자 알고리즘에 의한 BLDC 전동기 PID 속도제어기의 온라인 동조와 관련하여 GA 파라미터 변화에 따른 최적해 수렴 영향을 분석하였으나 속도제어기의 최적 동조로 제한되어있다. 본 논문에서는 GA를 이용하여 유도전동기 제어시스템에서 직결로 구성된 속도 및 전류제어기의 제어이득을 동시에 최적화하기 위하여 Matlab/Simulink를 이용하여 시스템을 설계, 시뮬레이션하여 기존의 PI제어기와 제어성능 비교에 의해 우수성 및 유도전동기 구동시스템에서 GA 적용에 대한 가능성을 확인하고 있다.

2. 유도전동기 수학적 모델 및 벡터제어

2.1 유도전동기 수학적 모델

고정자 직교좌표계의 전압과 전류는 모두 교류이고 각주파수 ω_e 로 회전하는 교류량을 직류로 해석하기 위해서 좌표축이 ω_e 로 회전하는 동기좌표계를 도입하여 d-q축으로 표현된 등가회로의 고정자, 회전자 전압방정식은 식(1)과 같이 표현할 수 있다.

$$\begin{aligned} V_{ds}^e &= R_s i_{ds}^e + p\lambda_{ds}^e - \omega_e \lambda_{qs}^e \\ V_{qs}^e &= R_s i_{qs}^e + p\lambda_{qs}^e + \omega_e \lambda_{ds}^e \\ V_{dr}^e &= R_r i_{dr}^e + p\lambda_{dr}^e - (\omega_e - \omega_r) \lambda_{qr}^e \\ V_{qr}^e &= R_r i_{qr}^e + p\lambda_{qr}^e + (\omega_e - \omega_r) \lambda_{dr}^e \end{aligned} \tag{1}$$

또한, 동기 회전좌표계에서의 d-q축 회전자 쇠교

자속에 대한 상호인덕턴스(L_m), 고정자 및 회전자 자기인덕턴스(L_s, L_r)의 관계는 식 (2)와 같다.

$$\begin{aligned} \lambda_{ds}^c &= L_s \cdot i_{ds}^c + L_m \cdot i_{dr}^c \\ \lambda_{dr}^c &= L_r \cdot i_{dr}^c + L_m \cdot i_{ds}^c \\ \lambda_{qs}^c &= L_s \cdot i_{qs}^c + L_m \cdot i_{qr}^c \\ \lambda_{qr}^c &= L_r \cdot i_{qr}^c + L_m \cdot i_{qs}^c \end{aligned} \quad (2)$$

이를 행렬식으로 나타내면 다음 식 (3)과 같다.

$$\begin{bmatrix} V_{ds}^c \\ V_{qs}^c \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} R_s + pL_s & -\omega L_s & pL_m & -\omega L_m \\ \omega L_s & R_s + pL_s & \omega L_m & pL_m \\ pL_m & -(w-\omega_r)L_m & R_r + pL_r & -(w-\omega_r)L_r \\ (w-\omega_r)L_m & pL_m & (w-\omega_r)L_r & R_r + pL_r \end{bmatrix} \begin{bmatrix} i_{ds}^c \\ i_{qs}^c \\ i_{dr}^c \\ i_{qr}^c \end{bmatrix} \quad (3)$$

여기서 p 는 미분연산자이며, 유도전동기 회전자 전압은 단락회로로 구성되어 있어 항상 0으로 표현할 수 있다

2.2 유도전동기 간접 벡터제어

자속성분 전류(회전자계)와 토크성분 전류(회전자 전류)로 분리하여 이들 성분이 직교하도록 하여 최대 토크가 발생하도록 제어하여 직류기와 동일한 특성을 얻을 수 있는데 전동기의 고정자 전류를 회전자 자속각을 기준으로 제어하는데 이를 벡터제어라 한다. 이 때 자속성분 전류는 자속각과 일치하여야 하고 동기회전 좌표계의 d 축과 일치하도록 하고, 토크성분 전류는 자속각과 직교하여 동기회전 좌표계의 q 축과 일치하도록 하여 벡터제어를 수행하는데 회전자 자속의 위치를 결정하는 방법에 따라 직접 및 간접 벡터제어로 구분된다. 간접벡터제어 방식은 토크전류와 자속전류로부터 슬립 각속도를 계산하여 좌표변환에 필요한 단위벡터를 발생시키는 방식이다.

$$\theta_e = \int_0^t \omega_e dt = \int_0^t (\omega_{sl} + \omega_r) dt \quad (4)$$

식 (1)의 회전자 전압식에서 슬립속도 및 d 축 자속은 다음과 같이 나타낼 수 있다.

$$\omega_{sl} = L_m \frac{R_r}{L_r} \frac{i_{qs}^c}{\lambda_{dr}^c} \quad (5)$$

$$\lambda_{dr}^c = \frac{1}{1+p} \frac{R_r}{L_r} L_m i_{ds}^c \quad (6)$$

간접 벡터제어방식은 저속으로부터 고속영역까지 운전이 가능하고 시스템 구성이 간단하나 전동기의 시정수 등 파라미터 변동에 대한 영향이 큰 단점이 있다. 본 논문의 간접벡터제어 시스템은 공간전압 벡터PWM인버터, 고정자좌표를 기준의 유도전동기 모델, 전류 및 속도 Anti-windup PI제어기, 자속관측기로 구성되며 Matlab/Simulink를 이용하여 그림 [2]와 같이 모델링하고 시뮬레이션 하였다.

2.3 제어기 설계

유도전동기 구동시스템은 전류, 위치 및 속도 센서로부터 정보를 받아 외부 지령과 비교에 의해 전류용 반도체소자의 On/Off 시점을 결정, 원하는 제어목표를 달성하도록 한다. 그림[2]는 유도전동기 제어계의 직렬접속을 나타낸 것이다. 내부 루프의 교차 각주파수를 외부 루프보다 높여 외부 제어기 설계가 간편하고 내부 루프의 전류 및 속도를 제한할 수 있다. 본 논문에서는 직렬접속의 전류 및 속도 PI제어기에 GAOT (Genetic Algorithm Optimization Toolbox)를 적용, PI 제어이득의 최적화에 의한 시스템의 제어성능 향상을 목적으로 한다.

2.3.1 전류제어기 설계

전류제어기의 출력은 리미터에 의해 제한되며, 리미터의 입,출력 오차를 Anti-windup 이득(K_a)을 통하여 적분기에 연결함으로써 적분기의 windup 현상을 방지하는데 본 논문에서는 K_a 를 $1/K_p$ 로

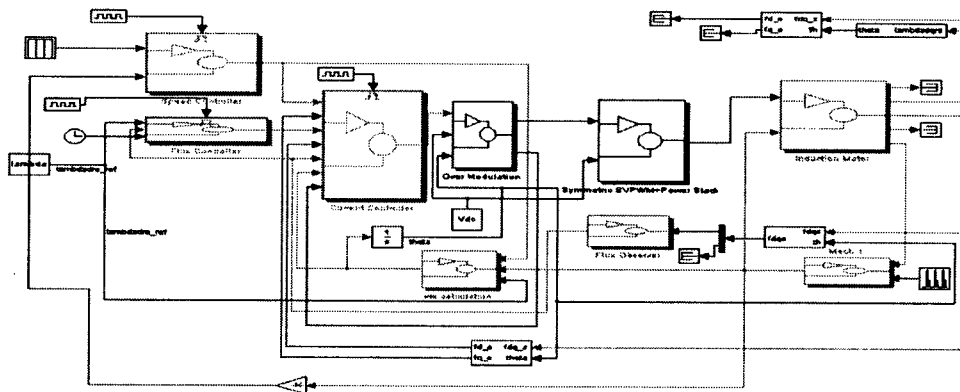


그림 1. 유도전동기 간접벡터제어의 Simulink 모델

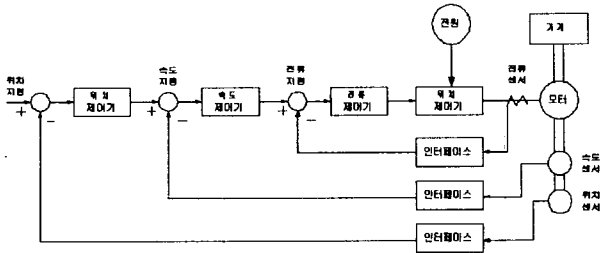


그림 2. 제어계의 직렬 접속

선정하였다. 비례, 적분이득을 식(7)과 같이 선정하였으며, 차단주파수 대역(w_c)은 $2\pi \cdot 150$ 로 설정하였다.

$$K_p = Lw_c, K_i = R w_c \quad (7)$$

2.3.2 속도 및 자속제어기

속도제어기는 q축 전류제어기와 직결된 구조 설계시 전류제어기에 미치는 영향을 검토하여야 한다. 일반적인 경우 속도제어기의 차단주파수를 전류제어기 차단주파수의 5~10분의 1 범위 내에서 선정하면 속도제어기가 전류제어기에 미치는 영향을 무시할 수 있다. 그림[3]은 속도제어기 모델로 제어이득은 다음과 같이 설정한다.

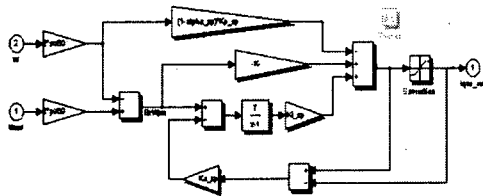


그림 3. PI 속도제어기의 Simulink 모델

여기서, 속도제어기의 출력은 I_{qse_ref} 의 최대, 최소값으로 제한하며, 비례, 적분이득은 시스템의 빠른 응답을 고려하여 식(8)과 같이 선정하였다.

$$K_p = \frac{\sqrt{2}J_m w_c}{K_t}, K_i = \frac{J_m w_c^2}{K_t}, K_a = \frac{2}{K_p} \quad (8)$$

여기서 K_t 는 $\frac{3}{4}p \frac{L_m}{L_r} \lambda_{dr}^*$ 이다. 또한, 속도제어기의 차단주파수(w_c)는 $2 \cdot \pi \cdot 10$ 으로 설정하였다. 마찬가지로 자속 PI제어기 이득은 다음과 같다.

$$K_p = \frac{L_r}{R_r L_m}, K_i = \frac{R_r}{L_r} K_p = \frac{w_c}{L_m}, K_a = \frac{1}{K_p} \quad (9)$$

3. GA를 이용한 제어기 동조 알고리즘

3.1 유전자 알고리즘 개요

유전 알고리즘은 1975년경 John Holland에 의해

처음 도입되어 최적화 탐색 알고리즘으로 전역적인 탐색을 기본으로 하며 이를 통해 최적화 문제를 해결하는데 이용한다. 이는 자연의 진화과정을 인공 시스템에 적용한 것으로 이진수의 문자열(string)로 표현되는 개체군(population)으로부터 출발한다. 이들 개체(individual)들은 모여서 하나의 세대(generation)를 이루며 복제(reproduction), 교배(crossover), 돌연변이(mutation)등과 같은 과정을 거쳐 다음 세대를 형성한다. 유전 알고리즘이 적용되는 목적에 따라 적합도(fitness value)를 판별하는 함수가 정의되고 각 세대들은 이 적합도가 큰 방향으로 적자생존의 이론을 기본으로 진화되어 나간다. 유전 알고리즘은 탐색 영역의 미분가능과 같은 수학적 조건이 필요하지 않고 국부 수렴을 피할 수 있다는 점에서 다른 최적화 기법에 비하여 우수하다. 전형적인 유전 알고리즘은 임의의 값으로 초기화된 개체들의 집합으로 시작하며 각각의 개체는 상대적인 문제해결 능력에 따라 적합도가 평가되고, 이에 따라 다음 세대에 부모의 유전자 복제(reproduction)되는 정도를 달리 함으로써 우성 형질을 지닌 개체들이 열성인 개체들에 비하여 더 많은 자식을 생성할 수 있도록 유도된다. 이렇게 선택, 복제된 개체들은 교배(crossover), 돌연변이(mutation)등의 유전연산자들에 의해 결합되어 다음 세대의 개체군을 형성한다. 이와 같은 방법으로 세대교체가 원하는 수준의 해가 개체군내에 존재하거나 다른 종료조건이 만족될 때까지 계속 반복된다.

3.2 GA를 이용한 PI제어기 최적화

PI제어기는 제어이득에 의존하며, 전동기의 운전 조건 및 파라미터의 변동에 따라 조정되어야 한다. 이에 PI제어기 이득 결정을 위하여 시스템 차수가 낮은 극배치기법부터 Nyquist 곡선의 한점의 정보에 의해 이득을 결정하는 Ziegler-Nichols의 계단 응답기법 및 수정Ziegler-Nichols 등이 있고 최적 이득 결정을 위한 이산형 공액경사법 등이 적용되고 있으나 보다 정확 이득을 구하기 위하여 본 논문에서는 적합도 함수 ITAE의 최소화에 의해 최적화를 수행한다.

목적함수를 최소화하는 것이 목적이나 GAOT에서는 최대화 문제를 해결하므로 목적함수에 -을 곱하여 최적화를 수행하므로 세대가 증가함에 따라 목적함수는 증가하고 있지만 크기가 작아지므로 실제는 최소화문제이다.

유도전동기 PI제어기는 다음 식 (9)로 표현된다.

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt \quad (10)$$

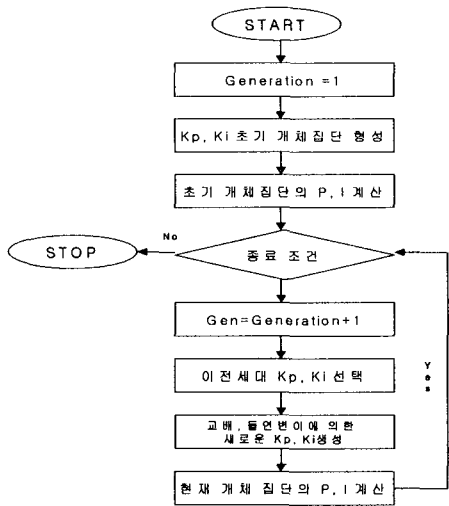


그림 4. GA에 의한 PI계수 조정 흐름도

여기서, $e(t) = y(t+1) - y(t)$ 이다. GA를 이용한 유도전동기 PI제어기의 최적화는 다음과 같이 정식화 할 수 있다.

$$\text{Minimize } K_p, K_i = \int_0^t |e(t)| dt \quad (11)$$

$$\text{bound to } K_{pl} \leq K_p \leq K_{pu}, K_{il} \leq K_i \leq K_{iu}$$

표 [1]은 본 논문에서 사용한 유전자 알고리즘 연산 파라미터, PI계수의 제약조건이다.

표 1. 유전자 알고리즘 파라미터

Parameter	Property	Value	
Selection Function	normGeomSelect/Roulette	0.08	
Termination Operator	maxGenTerm	20	
Mutation Operator	Binary	0.05	
Crossover Operator	simple	0.6	
Bounds	Current	Kp	0, 10
		Ki	1800, 3000
	Speed	Kp	0, 3
		Ki	80, 100

4. 시스템 시뮬레이션 결과

4.1 GA를 이용한 PI제어기 최적화

본 논문은 이진코딩의 GA를 적용하였으며 개체 선택함수로 "normGeomSelect" 및 "Roulette wheel"을 이용, 비교하였다. 전자는 6세대, 후자는 17세대 이후 수렴을 보여 "normGeomSelect"의 우수성을 확인하였다. 그림 [5]는 전류 및 속도제어기의 최적화 과정을 나타낸 것이다. 시뮬레이션에 의한 전류 및 자속제어기 최적해는 표 [2]와 같다.

표 2. PI제어기 최적해

	속도제어기		전류제어기	
	GA Tuning	일반선계	GA Tuning	일반선계
K_p	2.7913	1.9693	3.1903	6.4392
K_i	95.067	87.4927	2626.1	1991.5

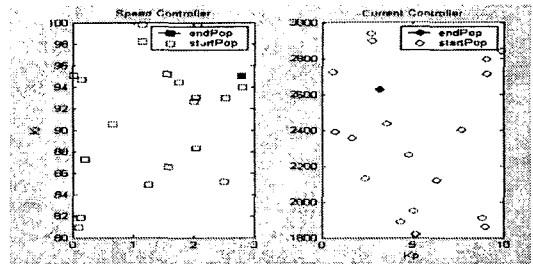


그림 5. Convergence of GA for PI Controller

4.2 시스템 시뮬레이션 결과

4.1에서 얻어진 전류 및 속도PI제어기의 최적해를 적용한 유도전동기 구동시스템 응답 특성을 그림 [6]에서 나타내고 있다.

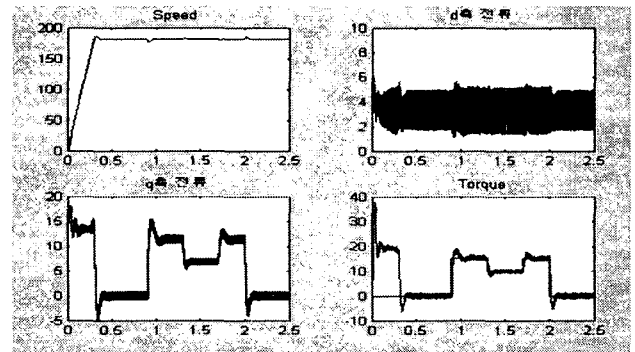


그림 6 시스템 응답 (GA Tuning based PI Controller)

GA로 PI계수를 최적화하면 일반PI제어기 보다 오버슈트가 0.28% 감소하고 정착시간이 0.3[s]로 감소한다.

5. 결론

본 논문은 유도전동기 벡터제어 시스템의 성능향상을 위하여 유전자 알고리즘을 이용하여 동시에 PI제어기의 온라인 동조를 제안한 논문이며, 본 논문으로 유도전동기 구동시스템에의 유전자 알고리즘의 적용 가능성을 보이지만, 동시 최적화를 위한 적합도 함수의 개발에 의한 연산시간 단축에 대한 연구가 필요할 것이다.

참고 문헌

- [1] Francesco Cupertino, "Design of Cascade Controllers for DC Drives using Evolutionary Algorithms" *Proceedings of the 2002 Congress on Evolutionary Computation*, Vol.2, pp. 1255-1260, 2002, May.
- [2] WG da Silva, "Tuning of a Brushless DC Drive Speed Controller with an On-Line Genetic Algorithm" *EPE'99 Lausanne*, 1999.
- [3] Wander G. da Silva, "Application of Genetic Algorithms to the Online Tuning of Electric Drive Speed Controllers" *IEEE, Trans. on Industrial Electronics*, Vol. 47, No. 1, pp. 217-219, Feb. 2000.