

유전알고리즘을 이용한 직류직권모터 시스템의 퍼지제어에 관한 연구

A Study on the Fuzzy Control of Series Wound Motor Drive Systems Using Genetic Algorithms

김종건*, 배종일, 이동철, 이만형

부산대학교 대학원(Tel: 051-510-1456),

부경대학교 전기공학과(Tel: 051-620-1437, E-mail: jibae@pine.pknu.ac.kr),

부경대학교 전기공학과(Tel: 051-620-1430, E-mail: dclee@pine.pknu.ac.kr),

부산대학교 기계공학부(Tel: 051-510-2331, E-mail: mahlee@hyowon.cc.pusan.ac.kr)

Abstract : Designing fuzzy controller, there are difficulties that we have to determine fuzzy rules and shapes of membership functions which are usually obtained by the amount of trial-and-error or experiences from the experts.

In this paper, to overcome these defects, genetic algorithms which is probabilistic search method based on genetics and evolution theory are used to determine fuzzy rules and fuzzy membership functions. We design a series compensation fuzzy controller, then determine basic structures, input-output variables, fuzzy inference methods and defuzzification methods for fuzzy controllers. We develop genetic algorithms which may search more accurate optimal solutions. For evaluating the fuzzy controller performances through experiments upon an actual system, we design the fuzzy controllers for the speed control of a DC series motor with nonlinear characteristics and show good output responses to reference inputs.

Keywords : Genetic algorithms, Fuzzy rule, Fuzzy membership function

1. 서론

퍼지제어기는 기존의 정확한 모델을 기초로 설계되는 제어기와는 달리 플랜트의 입출력 관계만으로도 좋은 제어 성능을 나타낼 수 있으며, 또한 인간의 의사결정방식을 도입함으로써 지능적인 제어를 할 수 있다. 그러나 퍼지규칙들이 시스템에 제대로 반영되도록 하려면 규칙에 사용된 퍼지변수의 소속함수 모양 등을 조정해야 하며, 종래에는 전문가의 경험과 지식 혹은 시행착오법으로 소속함수의 모양을 결정해야하는 단점이 있었다.

본 논문에서는 이러한 문제점을 해결하기 위해서 자연의 진화를 모의한 확률적인 최적화 탐색알고리즘인 유전알고리즘(genetic algorithms : GA)을 이용하여 퍼지규칙 및 퍼지변수의 소속함수를 동시에 동조하는데 유전알고리즘을 이용하는 방법을 제안하였다. 실시스템에 대한 전체하드웨어는 C언어로 프로그래밍한 퍼지제어기와 디지털 PWM 전류제어기, 대전력 증폭회로 등으로 구성하였다. 제안한 방법을 이용하여 실험을 해본 결과, 동조에 사용된 기준입력에 대한 속도응답과 동조에 사용되지 않은 임의의 기준입력에 대한 속도응답이 잘 추종함을 보여 주었다.

2. 퍼지제어이론 및 유전알고리즘

2.1 퍼지제어이론

퍼지제어기의 성능은 퍼지규칙의 적절한 선택과 멤버십 함수의 형태에 의해서 결정된다. 멤버십 함수의 형

태는 삼각형, 사다리꼴 등 여러 가지가 있으나, 본 연구에서는 삼각형 멤버십 함수를 사용하였으며, 동시에 퍼지규칙도 최적화 방법인 유전알고리즘을 사용하여 튜닝하였다.

2.2 유전자 알고리즘

유전자 알고리즘이 광범하게 적용되는 이유는 그 알고리즘이 0과 1로서 표현되는 비트로 이루어지기 때문에 컴퓨터 언어로 표현하기에 아주 간단하고, 발견하고자 하는 최적점이 존재하는 탐색공간에 특별한 가정을 필요로 하지 않기 때문이다.

유전자 알고리즘이 일반적인 최적화방법과 다른점은 다음의 네가지로 말할 수 있다.

- 1) 목적함수의 설계변수를 직접 조작하지 않고 Bit String으로 변환하여 최적화를 수행한다.
- 2) 최적점의 탐색이 하나의 점에 국한되지 않고, 여러 개의 점의 집합을 탐색한다.
- 3) Fitness값만을 사용하며, 미분계수나 기타의 정보를 필요로 하지 않는다.
- 4) 결정론적인 규칙을 사용하는 것이 아니라 확률론적인 지식을 최적화 과정에 이용한다.

2.3 유전자 알고리즘의 연산자

2.3.1 복제(reproduction)

고유의 string이 그들의 목적함수값 f에 의해 복제되

는 과정이다. string으로 표현되는 탐색점의 적합도에 의한 string copy는 더욱 좋은 적합도를 가진 것 또는 다음 세대의 자손에 기여할 가능성이 높은 값을 가진 string을 선택하여 자신의 string을 다음 세대에 복제해주는 과정이다. 이때 적합도가 낮은 것은 도태되어 진다.

2. 3. 2. 교차(crossover)

교차는 다음의 두 단계로 진행된다.

첫째, 유전자 pool에서 새롭게 reproduction된 string이 random하게 짝지어진다.

둘째, 각 string의 짝은 1과 [1, l-1]의 string 길이 사이에서 random하게 선택되어진 string의 정수위치 k에서 잘려지고, 두 개의 string이 위치 k+1과 l 사이의 모든 비트를 swapping(맞바꿈)함으로써 새로운 string을 만들어 낸다.

교차의 과정이 그림 1에 나타나 있다.

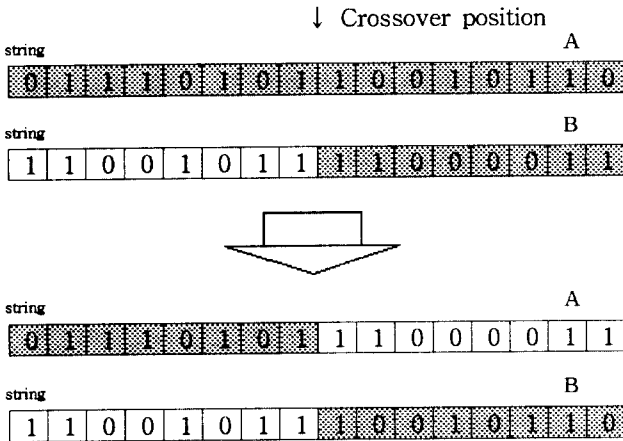


그림 1. 교차의 과정

2. 3. 3. 돌연변이(mutation)

돌연변이 연산자는 string이 고유한 유전인자를 잃어버리는 회복할 수 없는 손실을 막는 것으로, 유전자 알고리즘에서 돌연변이는 string 특정비트의 값을 random하게 변경한다. 돌연변이의 과정이 그림 2에 나타나 있다.

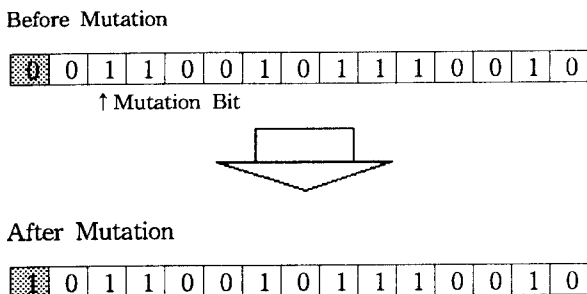


그림 2. 돌연변이의 과정

2. 4. 알고리즘의 단계

유전자 알고리즘은 설계변수와 목적함수를 컴퓨터에서 1 또는 0의 값을 가지는 여러 개의 Bit로 구성되는 string으로 표현하고, string을 조작하므로써 최적화를 수행하며, 그림 3과 같은 7단계로 이루어져 있다. 각 단계의 주요 내용은 다음과 같다.

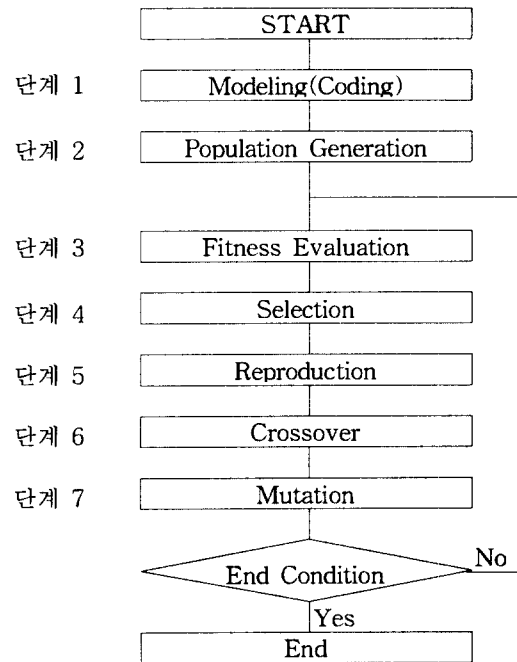


그림 3. 유전알고리즘의 흐름도

단계 1) 유전자 형의 설정

최적화 문제를 컴퓨터가 인식할 수 있도록 기호열로서 Coding한다. 각 기호열의 요소에 어떠한 숫자, 문자 등을 분배할까 하는 것은 설계자가 결정한다.

단계 2) 유전자 집단의 생성

단계 1)에서 결정한 기호열과 동일한 형태를 갖는, 그러나 각 기호열의 요소가 다른 일정 규모의 기호열의 집단을 랜덤하게 생성한다.

단계 3) 적합도(Fitness) 평가

단계 2)에서 생성된 기호열의 집단은 주어진 환경에 잘 적응할 수 있는 것부터 그렇지 못한 것까지 포함되어 있다. 얼마나 잘 적응하는가 하는 것을 판단하기 위한 것이 적합도(목적함수)이다. 즉, 각 기호열의 적합도를 계산하는 것으로, 단계 1)의 Coding에 대한 역변환인 Decoding 과정이라 볼 수 있다.

단계 4) 도태

계산된 적합도에 따라 다음 세대에서 살아남을 개체를 선정하고 선정되지 못한 개체는 도태시킨다. 도태비율은 설계자에 의하여 선정되어 진다.

단계 5) 복제(Reproduction)

적자생존 개념을 유전자 알고리즘에서 도입한 것으로 단계 4)에서 도태시킨 개체수 만큼 우수한 종의 개체를 복사하여 전체 개체수를 초기의 개체수와 동일하게 만든다. 이렇게 하여 적합도가 높은 것들 즉, 목적함수의 값이 좋은 것은 점점 숫자가 늘어나게 되어 전반적으로 최적으로 이동하게 된다.

단계 6) 교차(Crossover)

복제과정에서 적합도가 높아진 개체들을 무작위로 두 개씩 선정하고, 이들 개체들의 기호열 중간부분을 잘라 기호열을 서로 교환하여 새로운 개체를 생성한다. 새롭게 생성된 개체 가운데서 우수한 유전인자를 많이 받은 개체는 적합도가 높아지게 되어 복제과정에서 자신의 세력을 확장할 수 있게 된다. 이 때 교차시킬 개체의 선정과 교차시키는 위치의 선정은 확률론적 규칙에 따라 랜덤하게 선택한다.

단계 7) 돌연변이(Mutation)

적합도를 높이기 위하여 개체를 구성하고 있는 문자열의 임의의 Bit요소를 무작위로 하나 택하여 그 비트값을 바꾼다. 돌발적으로 적합도가 우수한 개체가 생성될 가능성이 있고 다목적함수(Multimodal Objective Function)에서 국부적인 최적해에 빠질 가능성을 줄일 수 있게 하는데 그 목적이 있다.

전체적으로 유전자 알고리즘은 단계 3)에서 단계 7)까지의 과정을 반복하면서 최적점을 찾아가는 방법이며, 3가지 과정 모두에 Random Number Generation이 깊이 관계된다. 유전자 알고리즘에서는 Coding된 기호열과 실세계 설계변수와의 관계를 그림 4.처럼 Mapping개념을 이용하여 연결한다. 이를 위하여 입력자료로서 설계변수들의 하한값과 상한값을 입력하고, 이들 값의 범위와 Coding된 기호열을 Mapping함으로서 설계변수의 형태가 정수, 실수, 정수와 실수들에 관계없이 사용할 수 있다.

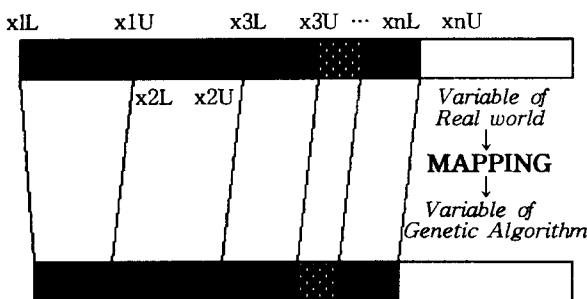


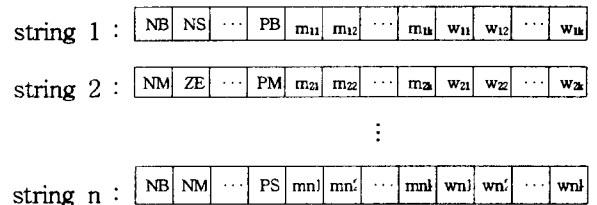
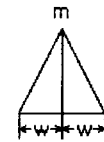
그림 4. 설계변수와 디코딩과의 관계

2.5 유전알고리즘을 이용한 퍼지제어기의 설계

유전알고리즘을 이용하여 퍼지규칙 및 퍼지변수의 소속함수를 동조하기 위해서 퍼지제어기의 입력으로 기준 입력과 실제출력 사이의 오차 및 오차의 변화율을 이용하였고, 퍼지제어기의 소속함수를 동조하기 위하여 유전 알고리즘을 사용하였다.

유전알고리즘으로 퍼지규칙 및 퍼지변수의 소속함수를 동조하기 위해서 퍼지규칙과 삼각형 소속함수의 꼭지점 및 밑변 길이의 1/2을 하나의 스트링, 즉 그림 5.와 같은 형태로 초기해집단을 구성하였다.

또한, 각 스트링을 평가하기 위한 적합도를 여러방법으로 선정할 수 있으나, 본 논문에서는 T시간 동안에 취득한 기준입력과 실제출력간 오차의 절대치의 합을 이용하여 식(1)과 같이 정하였다.



여기서, NB ... PB : 언어적인 변수

m_{ij} : 소속함수의 꼭지점

w_{ij} : 소속함수의 밑변의 길이의 1/2

n : 스트링의 수

k : 소속함수의 종류

그림 5. 초기 해집단의 구성

$$\text{적합도} = \frac{\beta}{\alpha + \sum_{k=0}^N |\text{error}_k|} \quad (1)$$

여기서, $\text{error}_k = \omega_r(k) - \omega_p(k)$

$\omega_r(k)$ = 기준입력

$\omega_p(k)$ = 실제출력

$N = T$ 시간 동안에 취득한 데이터의 수

α, β = 상수

본 논문에서 제안한 제어기 동조방법은 초기에 발생시킨 랜덤한 해집단에 대하여 식(1)로 표현된 적합도를 평가한 후 복제, 교차 및 돌연변이를 수행하고 반복 횟수

가 설정값 이하가 될 때까지 반복 수행한다. 반복 수행 과정에서 유전알고리즘의 개선된 방법 중의 하나인, 즉 적합도가 가장 높은 스트링 하나를 교차, 돌연변이가 과정을 수행하지 않고 다음세대로 그대로 복제하는 엘리티즘(elitism)을 사용하였다. 실시시스템에 대한 실험을 통해서 제안한 방법에 대한 성능을 검토하였다.

그림 6.에서 보는 바와 같이 총 49개의 퍼지규칙 (NB=1, NM=2, NS=3 등)중에서 점선의 윗부분에 해당하는 퍼지규칙 21개, 삼각형 멤버쉽 함수의 꼭지점과 밀변 길이의 1/2을 하나의 스트링으로 구성하였다. 이 때 오차, 오차의 변화율 및 출력부의 멤버쉽 함수의 꼭지점과 밀변의 길이가 각각 7개이므로 총 42개가 하나의 스트링이 된다. 또한, 멤버쉽 함수의 꼭지점과 꼭지점 사이를 15등분하여 탐색을 하였다.

본 연구에서 제시한 제어기 튜닝방법은 처음에는 멤버쉽 함수의 꼭지점과 밀변의 길이를 일정하게 두고 퍼지규칙을 유전알고리즘으로 튜닝하다가 적합도의 변화가 일정값 이내 혹은 설정된 반복횟수에 도달하면 퍼지규칙과 멤버쉽 함수를 동시에 튜닝하는 방법을 사용하였다.

Δe							
e	NB	NM	NS	ZO	PS	PM	PB
NB	1	2	3	3	2	1	4
NM	3	2	1	1	2	4	7
NS	2	3	3	1	4	6	6
ZO	2	2	1	4	7	7	5
PS	2	2	4	7	5	7	5
PM	2	4	6	6	5	6	6
PB	4	6	6	6	6	5	7

함수	Fuzzy rule sets	Fuzzy Membership Functions	
		꼭지점	밀변 길이의 1/2
string	123321321122331221222	457826...316543	854319...5465431

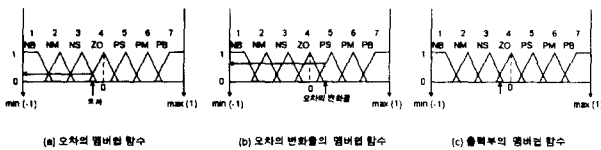


그림 6. 스트링의 구조

3. 실시시스템에 대한 실험

3.1 전체 하드웨어 구성

직류직권모터의 속도제어를 위한 전체하드웨어 구성도는 그림 7.과 같다. 전체하드웨어는 퍼지제어기, 디지털 PWM 전류제어기, 전류센서, 타코메타 및 대전력 증폭회로 등으로 이루어져 있다. 퍼지제어기와 디지털 PWM 전류제어기는 C와 어셈블러 언어를 이용하여 프로그래밍하였다.

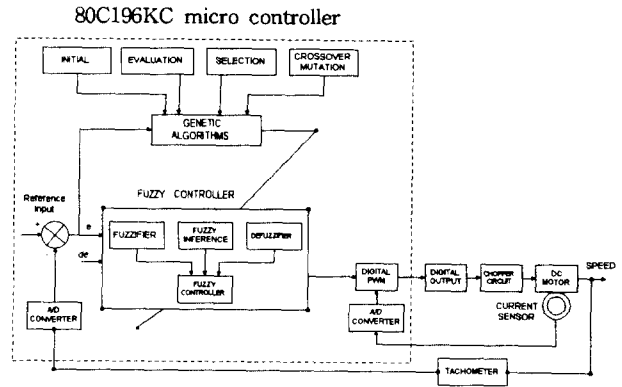


그림 7. 실시시스템의 속도제어를 위한 전체하드웨어 구성도

3.1.1 속도루프에 대한 성능 평가

제안한 방법으로 퍼지규칙 및 퍼지변수의 소속함수 모양을 동조하는데 사용된 실험계수는 표 1.와 같다. 그림 8.과 표 2.에서 얻어진 소속함수와 퍼지규칙은 정격부하 (160 [A])에서 속도가 1400 [rpm]일 때 얻어진 결과를 나타내고, 여기에서 1는 NB, 2는 NM, 3는 NS 등을 나타낸다. 그림 9.에서는 최대적합도 추이를 나타내었는데, x축은 세대수, y축은 최대적합도를 나타낸다. 그림 10.은 저속, 중속, 고속부분을 모두 포함하는 기준입력에 대한 속도응답을 나타낸다. 그림 8.과 표 2.에서 얻어진 퍼지규칙 및 퍼지변수의 소속함수를 이용하여 그림 10.에서 사용된 기준속도명령과 다른 임의의 기준속도명령에 대한 속도응답특성을 그림 11.에 나타내었다. 그림 11.에서 보는 바와같이 동조에 사용되지 않은 기준속도명령에 대해서도 좋은 속도응답특성을 확인할 수 있었다.

표 1. 유전알고리즘의 실험계수

계수	GA
해집단 갯수	20
스트링의 길이	12 비트
교배확률	0.95
돌연변이 확률	0.05
세대수	20

표 2. 동조된 퍼지규칙

Δe							
e	NB	NM	NS	ZO	PS	PM	PB
NB	1	1	3	2	1	1	4
NM	1	3	2	3	1	4	7
NS	2	1	1	1	4	7	7
ZO	2	1	1	4	7	5	6
PS	1	1	4	7	7	6	5
PM	2	4	7	7	7	5	6
PB	4	6	7	6	6	7	7

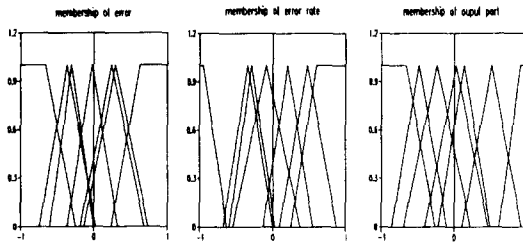


그림 8. 동조된 퍼지변수의 소속함수

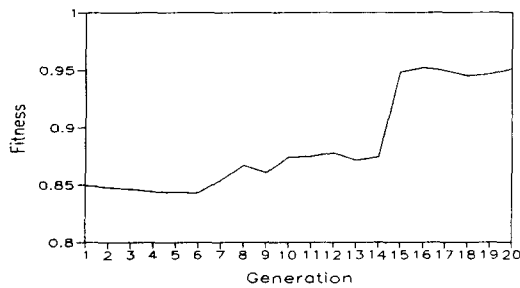


그림 9. 최대적합도 추이

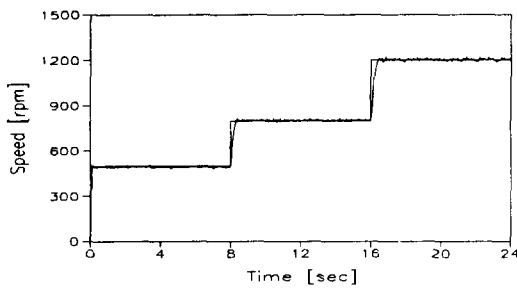


그림 10. 기준속도명령에 대한 속도응답특성

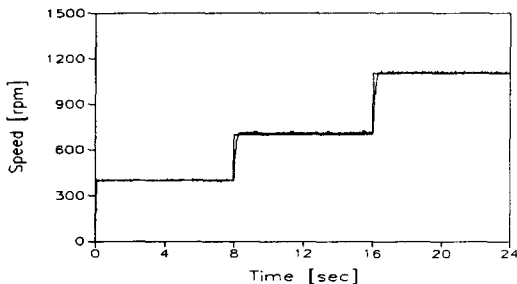


그림 11. 임의의 기준속도명령에 대한 속도응답특성

4. 결론

본 논문에서는 유전알고리즘을 이용하여 퍼지제어기의 소속함수의 자기동조 및 퍼지규칙을 튜닝하는 퍼지

제어기를 설계하는 방법을 제안하였다.

제안한 방법을 실시스템에 적용해 본 결과 튜닝한 기준속도명령에 대한 속도응답과 튜닝된 퍼지제어기의 일반화 능력을 평가하기 위하여 임의의 기준속도명령에 대한 속도응답이 잘 추종함을 보여 주었다.

향후 계획으로는 전용제어기를 DSP를 이용하여 부하 변동, 파라메타 변화, 비선형 부하 등에 대한 퍼지제어기의 성능 평가가 행해져야 할 것이다.

참고 문헌

- [1] Abraham Kandel, Gideon Langholz, *Fuzzy Control Systems*, CRC Press, 1994
- [2] Daihee Park, Abram, "Genetic-Based New Fuzzy Reasoning Models with Application to Fuzzy Control," *IEEE Trans. on Fuzzy Systems, Man, and Cybernetics*, Vol. 24, No. 1, pp. 39-47, Jan., 1994
- [3] David B. Fogels, "An Introduction to Simulated Evolutionary Optimization," *IEEE Trans. on Neural Network*, Vol. 5, No. 1, pp. 3-14, Jan., 1994
- [4] V. Nissou, "Solving the Quadratic Assignment Problem with Clues from Nature," *IEEE Trans. on Neural Network*, Vol. 5, No. 1, pp. 66-72, Jan., 1994
- [5] Vittorio Maniezzo, "Genetic Evaluation of the Topology and Weight Distribution of Neural Networks", *IEEE Trans. on Neural Networks*, Vol. 5, No. 1, pp. 39 - 53, Jan., 1994
- [6] D. A. Linkens, H. O. Nyongesa, "Genetic algorithms for fuzzy control Part 1 : Offline system development and application", *IEEE Proc. Control Theory Appl.*, Vol. 142, No. 3, pp. 161 - 176, May, 1995
- [7] D. A. Linkens, H. O. Nyongesa, "Genetic algorithms for fuzzy control Part 2 : Online system development and application", *IEEE Proc. Control Theory Appl.*, Vol. 142, No. 3, pp. 177 - 185, May, 1995