

퍼지 시스템과 진화연산을 이용한 DC 모터 속도제어

황기현, 문경준, 이화석, 김형수, 박준호
부산대학교 공과대학 전기공학과

A DC Motor Speed Control using Fuzzy System and Evolutionary Computation

K. H. Hwang, K. J. Mun, H. S. Lee, H. S. Kim, J. H. Park
Dept. of Electrical Engineering, Pusan National University

Abstract : This paper proposes a design of self-tuning fuzzy controller based on evolutionary computation. Optimal membership functions are found by using evolutionary computation.

Genetic algorithms and evolution strategy are used for tuning of fuzzy membership function. A arbitrarily speed trajectories is selected to show the performance of the proposed methods. Simulation results show the good performance in the DC motor control system with the self-tuning fuzzy controller based on evolutionary computation.

1. 서론

제어 시스템의 특성이 복잡하여 기존의 정량적인 방법으로는 해석할 수 없거나 언어지는 정보가 정성적이고, 부정확하고, 불확실한 경우에 퍼지 제어기는 우수한 제어 성능을 나타내고 있다. 그러나 퍼지 규칙들이 시스템에 제대로 반영되도록 하려면 규칙에 사용한 퍼지 변수의 소속함수 모양등을 조정하는데 있어서 전문가의 경험 혹은 시행착오적으로 경정해야하는 단점이 있다.

근래에 자연의 유전학과 진화이론에 바탕을 둔 확률적인 최적화 탐색방법인 유전알고리즘(genetic algorithms : GA)이 종래의 방법으로는 최적화하기 어려운 목적함수의 최적화, 신경회로망의 학습, 퍼지의 멤버십(membership) 함수의 튜닝, 기계학습, 시스템식별 및 제어등에 널리 응용되고 있다[1-10]. 또한, 유전알고리즘은 목적함수의 미분가능, 연속성이 관계없이 최적해를 구할 수 있는 장점이 있다.

본 연구에서는 퍼지 멤버십 함수의 동조에 유전알고리즘 및 evolution strategy(ES)를 결합[11]한 진화연산을 이용하여 DC 모터 제어기를 설계하였으며, 시뮬레이션을 통해 설계된 퍼지 제어기의 성능이 우수함을 보여주었다.

2. 퍼지 이론 및 진화 연산

2.1 퍼지 이론

퍼지 이론의 용어는 많은 분야에서 적용되고 있다. 퍼지 제어는 기존의 정확한 수학적 모델을 기초로 설계되는 제어기와는 달리 인간의 의사결정 방식을 도입하여 보다 인공지능적인 제어를 할 수 있다. 퍼지 제어기는 병렬 분산형제어, 논리형 제어, 언어적 제어등의 특징을 지니고 있으며, 퍼지 제어의 과정은 크게 세 가지로 분류된다.

- 1) 퍼지화 : 퍼지의 도입부에서 입력 변수값을 적절한 퍼지값으로 바꾸는 연산이다.
- 2) 퍼지 추론 : 제어규칙을 이용하여 출력을 계산하는 과정이며 MAX - MIN, MAX - PRODUCT 등이 있다.
- 3) 비퍼지화 : 출력부 전체 집합에서 정의된 퍼지 제어 조작량을 명확한 퍼지 제어 조작량으로 변화시켜 주는 작업을 말한다. 비퍼지화 방법에는 최대값 방법, 최대 평균법, 무게 중심법등이 있다.

2.2 진화 알고리즘

이 절에서는 멤버십 함수를 결정하기 위해 사용된 진화연산을 간략하게 기술한다.

2.2.1 유전 알고리즘

자연의 유전학 및 자연도태의 원리에 근거한 유전알고리즘은 확률적인 방법과 계계적인 계계적인 정보교환을 통해 탐색공간을 조사해나감으로써 주위환경에 알맞은 가장 적절한 해를 구하는 최적화 알고리즘이며 부호화 및 초기화, 평가, 복제, 교차 및 돌연변이 과정을 거쳐 최적해를 구하는 방법이다[7].

2.2.2 Evolution Strategy

GA가 유전자형에 기초한 최적화 알고리즘인 반면, ES는 걸 으로 표현되는 생물의 체질 즉, 표현형에 기초한 방법이다[6]. ES를 적용하여 최적해를 구하는 과정은 다음과 같다.

- a) 초기화 : 부모 X^0 (n차원 벡터)를 허용영역내에서 μ 개를 랜덤하게 발생시켜 초기해집단 $X_i, i = 1, \dots, \mu$ 를 생성한다.
- b) 돌연변이 : 부모 X^0 에 Gaussian random variable(평균은 0이고 표준편차는 임의로 설정)을 더하여 자손 $X_i, i = 1, \dots, \lambda$ 를 생성한다.
- c) 평가 : 주어진 목적함수에 따라 각 해($\mu + \lambda$ 혹은 λ)의 적합도를 평가한다.
- d) 선택 : 적합도가 높은 μ 개를 다음 세대의 새로운 부모로 선택 한다.
- e) b) ~ d) 과정을 반복하여 만족할만한 해를 얻으면 종료한다.

3. 진화 연산을 이용한 자기동조 퍼지 제어기 설계

진화 연산을 이용하여 퍼지 멤버십 함수를 동조하기 위해서 그림 1과 같은 제어기 모델을 제안하였다.

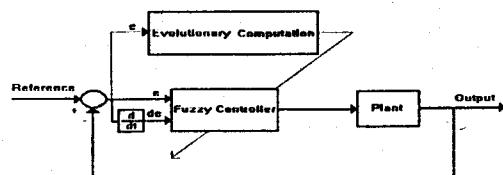


그림 1 진화연산과 퍼지 시스템을 이용한 제어기 모델

먼저 유전알고리즘으로 멤버십 함수를 동조하기 위해서는 삼각형 멤버십 함수의 꼭지점과 밑면 길이의 반음 하나의 스트링으로 구성한다. 즉 그림 2와 같은 형태로 초기 해집단을 구성한다.

또한 각 스트링을 평가하기 위한 적합도를 여러 방법으로 선정할 수 있으나, 본 연구에서는 T시간 동안에 취득한 실제출력과 목표출력간의 오차의 절대치의 합을 이용하여 다음과 같이 정하였다.

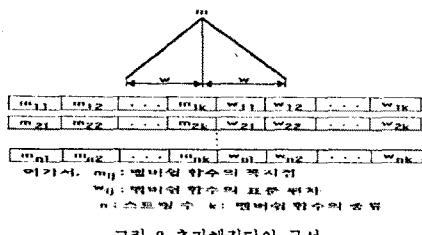


그림 2 초기해집단의 구성

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

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

$\omega_r(k)$: 원하는 입력

$\omega_p(k)$: 실제 출력

α, β : 상수

n : T 시간 동안에 취득한 데이터의 수

초기에 발생시킨 랜덤한 해집단을 이용하여 식(1)의 적합도를 평가한 후 복제, 교차 및 돌연변이를 수행하고 반복횟수를 설정 값 이하가 될 때까지 반복 수행한다.

본 연구에서 제시한 제어기 투닝방법은 ES를 이용하는 방법과 유전 알고리즘 및 ES를 결합하여 이용하는 방법이다. 전자는 해의 가능한 범위내에서 초기해집단을 생성한 후 평가, 돌연변이를 수행하고 반복횟수가 설정값 이하가 될 때까지 반복하는 방법이며, 후자는 초기에는 유전알고리즘을 이용한 경우와 동일하나 적합도의 변화가 일정값 이내 혹은 설정된 반복횟수에 도달하면 ES를 이용한다. 왜냐하면 유전알고리즘은 최적해 근처로 빨리 수렴하나 확률적인 방법의 특성상 최적해를 구하는 데 시간이 많이 걸리는 반면, ES는 초기 수렴성은 높으나 최적해를 찾는 능력이 뛰어나기 때문이다. 시뮬레이션을 통하여 제안한 두 방법에 대한 성능을 비교 검토하였다.

4. 사례 연구

4.1 적류 모터 방정식과 동작조건

사례 연구에 사용된 적류 모터의 동적 방정식은 다음과 같다.

$$K\omega_p(t) = -R_s i_a(t) - L_a \frac{di_a(t)}{dt} + V_i(t) \quad (2)$$

$$K i_a(t) = J \frac{d\omega_p(t)}{dt} + D\omega_p(t) + T_L(t) \quad (3)$$

이다.

또한, 부하 토오크 $T_L(t)$ 는 다음과 같은 비선형 형태로 표시되고, μ 는 상수이다.

$$T_L(t) = \mu \omega_p^2(t) [\text{sign}(\omega_p(t))] \quad (4)$$

본 연구에서는 전류제어 방식을 사용하며, 식(3)에서 $r(t) = K i_a(t)$ 로 두면,

$$r(t) = J \frac{d\omega_p(t)}{dt} + D\omega_p(t) + T_L(t) \quad (5)$$

이다.

시뮬레이션을 위해, DC 모터의 여러 파라메타는 참고문헌[12]의 데이터를 사용하였다.

4.2 시뮬레이션

그림 1에 제안한 피지 제어기의 구조는 입력으로 기준입력과 실제 출력과의 오차 및 오차의 변화율을 이용하였다. 투닝에 이용한 피지 규칙은 표 1과 같으며, 기준입력은 다음과 같은 비선형 함수를 택하였다.

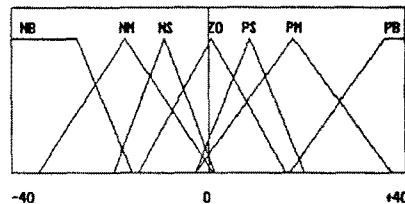
$$\omega_r(k) = 15 \sin(2\pi kT/4) + 15 \sin(2\pi kT/7), \quad (6)$$

제안한 방법으로 투닝된 오차의 멤버쉽 함수는 그림 3과 같다. 이 경우의 출력응답 특성은 그림 4에 나타나 있고 일반화 능력을 평가하기 위하여 투닝에 사용되지 않은 식 (7)과 같은 기준 입력에 대한 결과를 그림 5에 나타내었다.

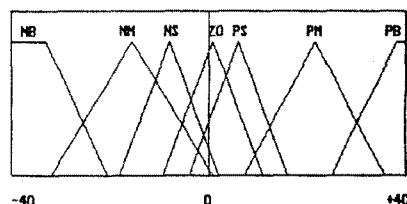
$$\omega_r(k) = 15 \sin(2\pi kT/3) + 15 \sin(2\pi kT/7), \quad (7)$$

표 1 투닝에 사용한 피지 규칙

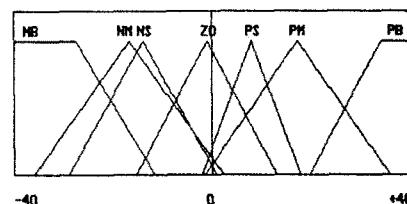
| ΔE | NB | NM | NS | ZO | PS | PM | PB |
|------------|----|----|----|----|----|----|----|
| NB | | | | NB | NM | | |
| NM | | | | NM | | | |
| NS | | | | NS | ZO | PM | |
| ZO | NB | NM | NS | ZO | PS | PM | PB |
| PS | NM | | | ZO | PS | | |
| PM | | | | | PM | | |
| PB | | | | | | | |



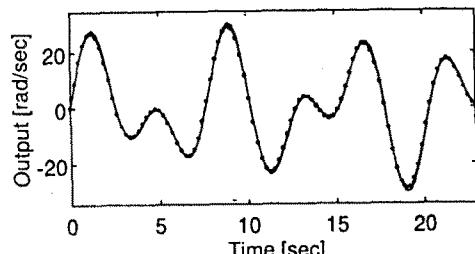
a) GA를 이용한 경우



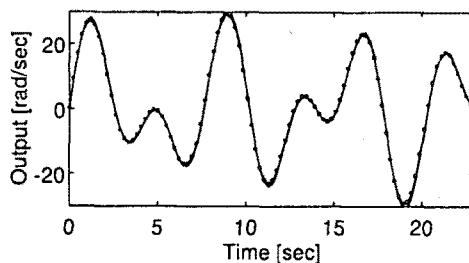
(b) ES를 이용한 경우



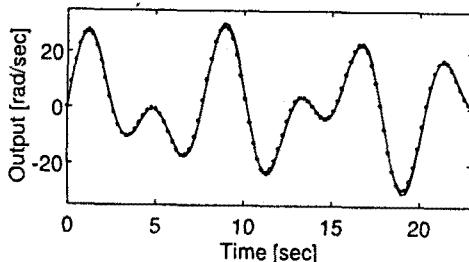
(c) GA 와 ES를 결합한 경우
그림 3 오차 멤버쉽 함수의 투닝 결과



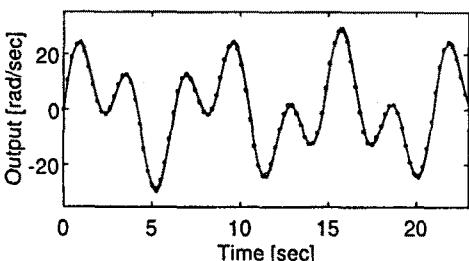
a) GA를 이용한 경우



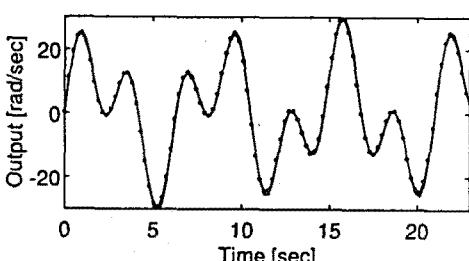
b) ES를 이용한 경우



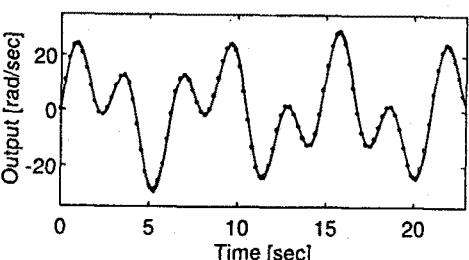
c) GA와 ES를 결합한 경우
그림 4 기준입력에 대한 출력응답



a) GA를 이용한 경우



b) ES를 이용한 경우



c) GA와 ES를 결합한 경우
그림 5 새로운 기준입력에 대한 출력응답

그림 5에서 보는 것처럼 제이기 투닝시에 사용된 입력과 다른 입력이 인가되었을 때에도 반죽할 만한 성능을 얻었으며 GA와 ES를 합성한 방법이 가장 우수함을 확인할 수 있었다.

5. 결 론

본 연구에서는 전화인산을 이용하여 퍼지 제이기의 멤버쉽 함수를 투닝하는 퍼지 제이기의 설계방법을 제시하였다. 종래의 퍼지 제이기의 단점은 각 멤버쉽 함수의 투닝을 시행착오법에 바탕을 두었으나 본 논문에서 제시한 전화인산을 이용하면 최적화 방법으로 자동으로 투닝할 수 있다. 또한 제안한 두 가지 방법을 시뮬레이션한 결과, 유전알고리즘에 ES를 결합한 방법이 유전알고리즘만 이용한 경우보다 더 나은 성능을 얻었다.

6. 참고 문헌

- [1] K. Krishnakumar, David E. Goldberg, "Control System Optimization Using Genetic Algorithms", Journal of Guidance, Control, and Dynamics Vol. 15, No. 3, pp. 735 - 740, May ~ June, 1992
- [2] 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
- [3] A. Varšek, T. Urbančič, and B. Filipič, "Genetic Algorithm in Controller Design and Tuning", IEEE Trans. on Systems, Man, and Cybernetics, Vol. 23, No. 5, pp. 1330 - 1339 Sept./Oct. 1993
- [4] N. Sarvanan and D. B. Fogel, "Evolving Neurocontrollers using Evolutionary Programming", Proc. of the First IEEE Conference on Evolutionary Computation, Vol. 1, pp. 217 ~ 222, 1994
- [5] J. J. Buckley and Y. Hayashi, "Fuzzy Genetic Algorithms for Optimization", Proc. of 1993 International Joint Conference on Neural Networks, pp. 725 - 728, 1993
- [6] David B. Fogels, "An Introduction to Simulated Evolutionary Optimization", IEEE Trans. on Neural Network, Vol. 5, No. 1, pp. 3 - 14, Jan., 1994
- [7] David E. Goldberg, "Genetic Algorithms in Search, Optimization, and Machine Learning", Addison-Wesley, 1989
- [8] V. Nissen, "Solving the Quadratic Assignment Problem with Clues from Nature", IEEE Trans. on Neural Network, Vol. 5, No. 1, pp. 66 - 72, Jan., 1994
- [9] X. Qi and F. Palmieri, "Theoretical Analysis of Evolutionary Algorithms with an Infinite Population Size in Continuous Space", IEEE Trans. on Neural Network, Vol. 5, No. 1, pp. 102 - 119, Jan., 1994
- [10] T. Bäck, "Selective Pressure in Evolutionary Algorithms: A Characterization of Selection Mechanisms", Proc. of the First IEEE Conference on Evolutionary Computation, Vol. 1, pp. 57 - 62, 1994
- [11] 황기현, 박준호, 이화석, "신경망과 전화알고리즘을 이용한 DC 모터 속도제어", 대한전기학회 추계종합학술대회논문집, pp. 359 ~ 361, 1994.
- [12] Siri Weerasriya and M. A. El-Sharkawi, "Identification and Control of a dc motor using back-propagation neural networks", IEEE Trans. on Energy Conversion, Vol. 6, No. 4, pp. 663 - 669, Dec., 1991