

유전자 알고리즘을 이용한 퍼지 스케일링 게인 제어기의 설계

신현석, 고재원[†], 권 철, 박민용
 연세대학교 전자공학과, [†]유한대학 제어계측과

Design of Fuzzy Scaling Gain Controller using Genetic Algorithm

Hyunseok Shin, Jaewon Kho[†], Cheol Kwon, Mignon Park
 Dept. of Electronic Engineering, Yonsei University,
[†]Dept. of Control and Instrumentation Engineering, Yuhan College

Abstract - This paper proposes a method which can resolve the problem of existing fuzzy PI controller using optimal scaling gains obtained by genetic algorithm. The new method adapt a fuzzy logic controller as a high level controller to perform scaling gain algorithm between two pre-determined sets.

1. 서 론

퍼지 논리 제어(fuzzy logic control, FLC)는 고전적인 제어 방식으로 분석이 어려운 복잡한 플랜트에 적용했을 때 그 유용성이 드러난다. 그러나, 실제적으로 고전적인 제어방식으로는 과도 상태와 정상 상태 모두에서 좋은 성능을 얻기가 매우 어렵기 때문에 퍼지 PI 제어기의 구조를 가지는 FLC에 스케일링 게인(scaling gain)을 적용시켜 과도 상태와 정상 상태에 대해 서로 다른 2개의 게인집합을 얻은 후 각각의 상태에 적절한 값으로 스위칭하여 두 상태에서의 응답을 향상시키는 연구가 발표된 바 있다[2]. 이 방법의 장점은 게인에 따라 멤버 함수의 파라미터들이 간접적으로 튜닝되는 효과를 얻을 수 있어서 적은 수의 파라미터 조정만으로도 성능의 향상을 꾀할 수가 있다는 것이다. 또한, 병렬 전역 탐색 알고리즘인 유전자 알고리즘(genetic algorithm, GA)을 이용해 각각의 집합들을 찾아낸 사례가 있다 [1]. 그러나, 이러한 경우에는 집합 간의 스위칭에 의한 채터링의 문제가 발생하여 플랜트나 제어기에 손상을 줄 수도 있고, 심한 비선형성을 가지는 플랜트를 제어하려면 두 집합 간을 계속적으로 스위칭하게 되는 경우가 있을 수 있다. 또한, 유전자 알고리즘을 통해 얻은 게인 집합들은 유전자 알고리즘의 실행시 설정하였던 환경에서만 최적화 된 것이라고 볼 수 있으므로, 그 이외의 에러의 범위 내에선 최적의 게인값들이 적용되고 있다고 볼 수 없는 단점이 있다. 그러므로, 본 논문에서는 스케일링 게인을 퍼지화하여 이를 상위 퍼지 제어기로 두고, 하위 PI 퍼지 제어기를 최적의 상태로 유지하여 위와 같은 문제를 해결하는 방법을 제시한다.

2. 퍼지 PI 제어기와 스케일링 게인

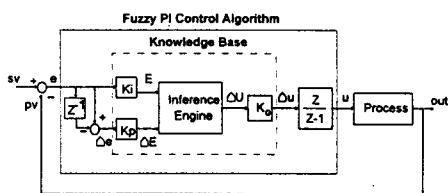


Fig.2-1 퍼지 PI 제어기의 블록도

일반적인 퍼지 PI 제어기는 그림 2-1과 같다[2]. 만일 K_i , K_p , 그리고 K_o 를 스케일링 게인으로 이용한다면 많은 장점이 생긴다. 이는 스케일링 게인 집합을 변화시킨다는 것이 입출력 변수의 퍼지도와 제어 정밀도를 연관시키는 작용을 하기 때문이다. 이러한 사실을 이용하면 간단히 스케일링 게인을 조정함으로써 제어기의 정밀도를 조절할 수 있다(그림 2-2)[2]. 일반적으로 시스템의 출력은 과도상태를 지나 정상상태에 도달한다. 이때 각 상태에서의 특성이 다르기 때문에 서로 다른 정밀도를 필요로 하게 된다. 예를 들어, 과도상태에서는 에러가 크기 때문에 대략적이면서도 빠른 제어 입력이 필요하며, 정상상태에서는 정상상태 에러를 줄이기 위해 세밀한 제어 입력이 필요하다. 그러므로, 이러한 제어 입력들을 위해서는 멀티-룰 베이스를 구성해야만 한다. 그러나, 위에서 말한 스케일링 게인 집합을 적용하여 조정하면 멀티-룰 베이스의 효과를 기대할 수 있다.

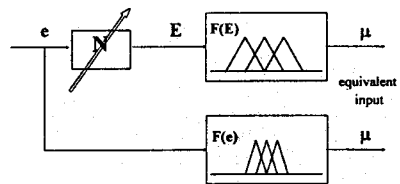


Fig.2-2 Effect of Scaling Gain

3. 퍼지 스케일링 게인 알고리즘

기존에 제안된 제어기는[1], 몇가지 문제가 발생할 수 있다. 즉, 집합 간의 스위칭에 의한 채터링의 문제가 발생하여 플랜트나 제어기에 손상을 줄 수도 있고, 심한 비선형성을 가지는 플랜트를 제어하려면 두 집합 간을 계속적으로 스위칭하게 되는 경우가 있을 수 있다. 또한, 유전자 알고리즘을 통해 얻은 게인 집합들은 유전자 알고리즘의 실행시 설정하였던 환경에서만 최적화 된 것이라고 볼 수 있으므로 그 이외의 에러의 범위 내에선 최적의 게인값들이 적용되고 있다고 볼 수 없다. 그러므로, 본 논문에서는 게인 집합을 스위칭하는 대신 퍼지추론에 의해 두 개의 스케일링 게인 집합을 보간하는 방법을 제안한다(그림 3-1). 제안된 방법은 위에 기술한 문제들을 해결하고, 전문가의 경험을 보다 쉽게 반영할 수 있는 장점이 있다. 제안된 방법의 성능은 선형 및 비선형 플랜트에 대한 컴퓨터 시뮬레이션으로 입증될 것이다.

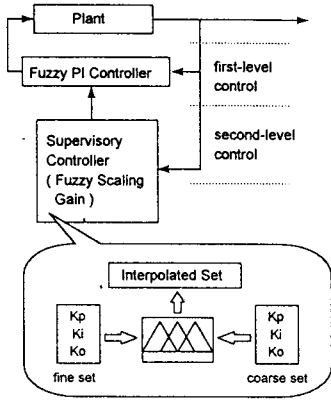


Fig.3-1 퍼지 스케일링 제어기의 블록도

4. 시뮬레이션 및 결과

제안된 방법의 성능을 입증하기 위해 간단한 선형 플랜트와 복잡한 비선형 플랜트가 이용되었다. 각 플랜트에 대해 유전자 알고리즘은 최적의 스케일링 게인 집합을 얻어낼 수 있다. 또한, 각 스케일링 게인 집합의 특성을 비교하기 위해 두 플랜트에 대한 시뮬레이션은 아래와 같은 네 개의 경우로 나누어 실행하였다.

- case 1. 과도상태의 스케일링 게인 집합만을 이용한 시뮬레이션
- case 2. 정상상태의 스케일링 게인 집합만을 이용한 시뮬레이션
- case 3. 기존의 퍼지 PI 제어기를 이용한 시뮬레이션
- case 4. 제안된 퍼지 스케일링 제어기를 이용한 시뮬레이션

4.1 선형 플랜트에 대한 시뮬레이션

첫 번째 시뮬레이션은 다음과 같은 간단한 선형 플랜트에 적용되었다.

$$G(s) = \frac{s+1}{s^2+4s+5} \quad (4-1)$$

그리고, 최적의 게인 집합을 얻어내기 위한 유전자 알고리즘의 설정은 표 4-1과 같다. ω_1 과 ω_2 는 에러와 에러의 미분값에 대한 가중치이다.

표 4-1 유전자 알고리즘의 파라미터

| | |
|-------|-------------------------------------------------|
| 개체 수 | 50 |
| 세대 수 | 50 |
| 돌연변이율 | 0.1 |
| 교배율 | 0.7 |
| 적합도 | $\frac{1}{1+\omega_1 E^2 + \omega_2 \dot{E}^2}$ |

유전자 알고리즘의 실행 결과 얻어진 게인 집합은 다음과 같다(표 4-2).

표 4-2 유전자 알고리즘에 의해 얻어진 게인 집합

| | | 게인 집합 |
|------|----|------------------|
| 과도상태 | Kp | 0.06451612903226 |
| | Ki | 1.85043988269795 |
| | Ko | 0.55816226783969 |
| 정상상태 | Kp | 0.14271749755621 |
| | Ki | 1.21603128054741 |
| | Ko | 0.97067448680352 |

다음의 시뮬레이션 결과에 의해 제안된 방법의 우수성을 확인할 수 있다.

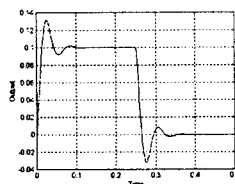


Fig.4-1 (1) Output (using only transient state gains)

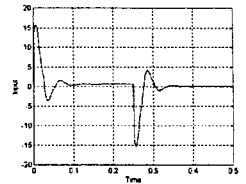


Fig.4-1 (2) Input (using only transient state gains)

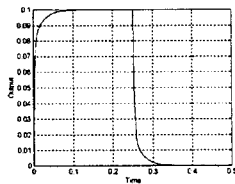


Fig.4-2 (1) Output (using only steady state gains)

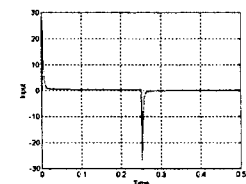


Fig.4-2 (2) Input (using only steady state gains)

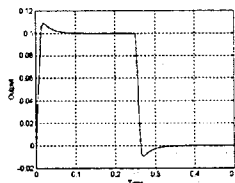


Fig.4-3 (1) Output (using scaling gain algorithm)

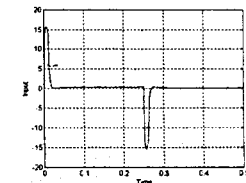


Fig.4-3 (2) Input (using scaling gain algorithm)

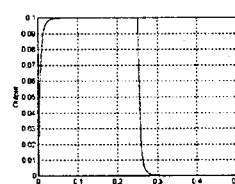


Fig.4-4 (1) Output (using proposed method)

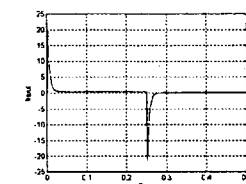


Fig.4-4 (2) Input (using proposed method)

시뮬레이션 결과의 수치적 비교를 위해서, 표 4-3에 몇 가지 척도를 보였다.

표 4-3 각 경우에 대한 비교척도

| | Rising Time (sec) | Percent Overshoot (%) | Settling Time (Sec) |
|--------|-------------------|-----------------------|---------------------|
| case 1 | 0.010 | 32.0 | 0.064 |
| case 2 | 0.015 | 0 | 0.032 |
| case 3 | 0.009 | 9.2 | 0.036 |
| case 4 | 0.011 | 0 | 0.018 |

case 1에서 플랜트의 출력은 큰 오버슈트와 진동을 보이고 있고(그림 4-1), case 2에서는 수렴속도가 너무 느리다는 것을 알 수 있다(그림 4-2). 그러므로, 각 스케일링 게인 집합의 장점을 혼합하는 스케일링 게인 알고리즘을 이용하면 더 나은 결과를 얻을 수 있고(그림 4-3), 제안된 방법을 이용하면 가장 좋은 성능의 제어기를 구성할 수 있음을 확인할 수 있다(그림 4-4, 표 4-3).

4.2 비선형 플랜트에 대한 시뮬레이션

회전역진자(그림 4-5)가 두 번째 시뮬레이션에 이용되었다(8). 널리 알려진 것처럼 회전역진자는 심한 비선형성을 보이며, 분석하기가 매우 복잡하여 원하는 제어 입력을 쉽게 만들어 내기가 어렵다.

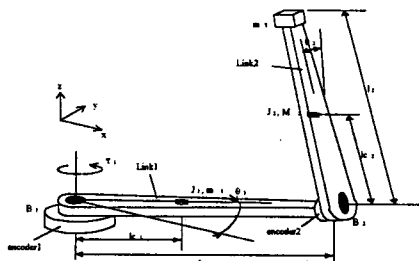


Fig.4-5 회전역진자

회전역진자의 동역학 방정식은 다음의 식 4-2와 같다(8).

$$\begin{aligned}
 & (J_1 + M_e l_1^2 + M L_e^2 \sin^2 \theta_2) \ddot{\theta}_1 - l_1 M L_e \cos \theta_2 \ddot{\theta}_2 \\
 & + (B_1 + M L_e^2 \sin 2\theta_2 \dot{\theta}_2) \dot{\theta}_1 + l_1 M L_e \sin \theta_2 \dot{\theta}_2^2 = \tau \\
 & l_1 M L_e \cos \theta_2 \ddot{\theta}_1 - (J_2 + M L_e^2) \ddot{\theta}_2 + M L_e g \sin \theta_2 \\
 & + M L_e^2 \sin 2\theta_2 \cos \theta_2 \dot{\theta}_1^2 - B_2 \dot{\theta}_2 = 0
 \end{aligned} \quad (4-2)$$

$$(M_e = m_2 + m_{\tau}, M L_e = m_2 l_{c2} + m_1 l_2, M L_e^2 = m_2 l_{c2}^2 + m_1 l_2^2)$$

시뮬레이션을 위하여, 회전역진자의 상수를 표 4-4와 같이 정했다.

표 4-4 회전역진자의 상수

| | | | |
|-------|-------------------------------------|----------|-------------------------------------------------------------|
| l_1 | 0.22m | J_2 | $1.98 \times 10^{-4} \text{kg} \cdot \text{m}^2$ |
| l_2 | 0.12m | B_1 | $0.118 \text{N} \cdot \text{m} \cdot \text{s}$ |
| m_2 | 0.055kg | B_2 | $8.3 \times 10^{-5} \text{N} \cdot \text{m} \cdot \text{s}$ |
| J_1 | $0.0175 \text{kg} \cdot \text{m}^2$ | l_{c2} | 0.06m |

유전자 알고리즘의 계산 결과, 최적의 스케일링 게인 집합은 표 4-5와 같이 얻어졌다.

표 4-5 유전자 알고리즘을 통해 얻은 최적의 게인 집합

| | | Gain Value |
|-----------------|----|------------------|
| Transient State | Kp | 0.70869990224829 |
| | Ki | 0.41231671554252 |
| | Ko | 0.3264907135875 |
| Steady State | Kp | 0.78592375366569 |
| | Ki | 0.22228739002933 |
| | Ko | 0.5591397849462 |

다음의 시뮬레이션 결과들을 보면 각각의 게인 집합이 초기값, 즉 에러상태에 따라 어떤 출력을 내는지를 알 수 있다. 우선 그림 4-6과 그림 4-7에서는 정상상태의 게인이 초기값이 0.01° 일 때, 즉 에러의 크기가 작을 때 더 빠른 수렴을 보임을 알 수 있다. 그러나, 다음의 그림 4-8과 그림 4-9에서 알 수 있듯이 정상상태의 게인 집합은 초기값이 20° 일 때 즉, 에러의 크기가 큰 경우에는 제어를 하지 못하여 진자가 쓰러지게 되며(그림 4-9 (1), 180° 를 중심으로 진동), 대신 과도상태의 게인 집합은 진자를 세울 수 있다(그림 4-8). 위의 두 게인 집합을 이용하는 스케일링 게인을 통한 제어를 하게 되면 그림 4-10과 같이 과도상태의 게인 집합을 이용한 경우의 출력보다 더 빠른 수렴을 보이지만 에러가 작아져서 게인 집합간의 스위칭이 일어나는 순간에 입력에 채터링이 발생하고 이에 따라 정상상태의 응답이 계속적으로 불안함을 알 수 있다. 이에 반하여, 본 논문에서 제안한 방법을 이용하면 두 게인 집합 사이의 스위칭이 없이 부드러운 게인의 스케줄링으로 빠른 수렴속도와 안정성을 얻을 수 있음을 그림 4-11을 통해 알 수 있다. θ_1 에 대한 출력이 지속적으로 발산하는 것(그림 4-11 (3))은 하위 제어기인 퍼지 PI 제어기 자체가 단지 θ_2 에 대한 에러와 에러의 미분값만을 입력으로 받아들이기 때문이다. 즉, θ_1 에 대한 정보가 없기 때문에 θ_2 가 0으로 수렴을 한 후에도 θ_1 은 수렴을 하지 않고 일정속도로 등속운동을 하여 θ_2 를 0으로 유지시키는 것이다.

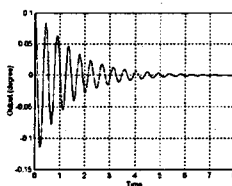


Fig.4-6 (1) Output
(using only transient state gains when error is small)

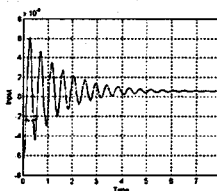


Fig.4-6 (2) Input
(using only transient state gains when error is small)

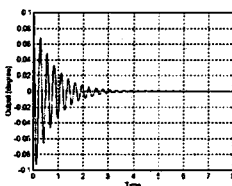


Fig.4-7 (1) Output
(using only steady state gains when error is small)

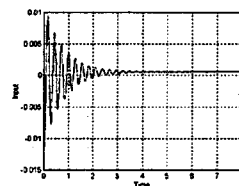


Fig.4-7 (2) Input
(using only steady state gains when error is small)

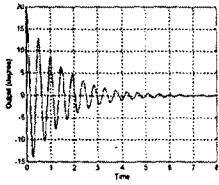


Fig.4-8 (1) Output
(using only transient state gains
when error is big)

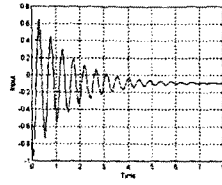


Fig.4-8 (2) Input
(using only transient state gains
when error is big)

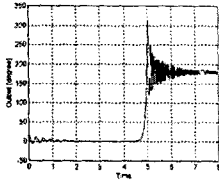


Fig.4-9 (1) Output
(using only steady state gains when
error is big)

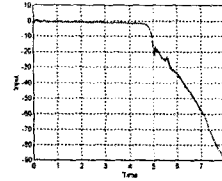


Fig.4-9 (2) Input
(using only steady state gains when
error is big)

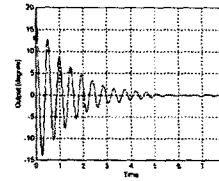


Fig.4-10 (1) Output
(using scaling gain algorithm)

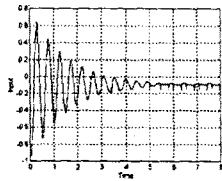


Fig.4-10 (2) Input
(using scaling gain algorithm)

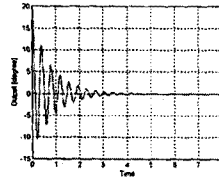


Fig.4-11 (1) Output
(using proposed method)

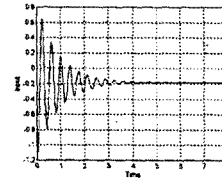


Fig.4-11 (2) Input
(using proposed method)

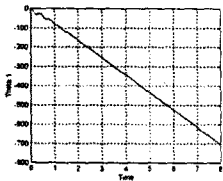


Fig.4-11 (3) θ ,
(using proposed method)

다음 표 4-6은 선형플랜트에 대한 시뮬레이션과 마찬가지로 제어기의 성능을 수치적으로 나타내어 본 것이다.

표 4-6 각 경우에 대한 비교척도

| | | Settling Time (sec) | Sum of Error |
|----------|--------|---------------------|----------------------|
| 에러가 작을 때 | case 1 | 4.32 | 0.0783 |
| | case 2 | 2.41 | 0.0716 |
| 에러가 클 때 | case 1 | 4.02 | 2.1769×10^3 |
| | case 2 | 발산 | 8.1982×10^9 |
| case 3 | | 4.23 | 2.1765×10^3 |
| case 4 | | 2.35 | 1.8481×10^3 |

$$* \text{Sum of Error} = 100 \times e^2 + e^2$$

5. 결 론

본 논문에서 제안된 퍼지 스케일링 게인 제어기는 기존의 퍼지 PI 제어기의 문제점을 해결할 수 있음을, 선형 및 비선형 플랜트에 적용한 시뮬레이션을 통해 확인하였다.

그러나, 일반적 의미에서 스케일링 게인 제어기는 두 개의 게인 집합보다 더 많은 게인집합을 이용하는 것이 보통이므로, 적절한 수의 최적의 게인들을 찾아내어 제안된 방법을 적용한다면 제어기의 성능이 더 향상될 것이다. 또한 실제 플랜트에 제안된 방법을 적용하고자 할 경우에 최적의 게인 집합을 얻어낼 유전자 알고리즘을 어떻게 실행할 것인지에 대한 문제점이 있다. 온라인 튜닝의 경우 잘못된 게인 집합은 시스템의 파손을 가져올 수 있으며, 오프라인 튜닝의 경우에는 플랜트 모델링의 부정확성, 불확실성 등으로 인한 상수의 오차가 있을 수 있으므로 이를 보완하여야 한다.

(참 고 문 헌)

- [1] Hyunseok Shin, Hansoo Shim, Cheol Kwon, Hyungjin Kang, Mignon park, "Designing a Fuzzy Logic Controller Using GA Optimal scaling Gain Tuning Method," Proceedings of JCEANF '96 pp 23-26, 1996
- [2] Han-Xiong Li and H. B. Gatland, "A New Methodology for Designing a Fuzzy Logic Controller," IEEE Trans. Syst. Man Cyber., vol 25, no. 3, pp 505-512, March 1995
- [3] Jianzhou WANG, Jim R. JORDAN, "Auto-Tuning Algorithm For Fuzzy Control Systems," 7th IFSA World Congress, Prague, pp 517-525, 1997
- [4] G. M. Abdelnour, "Designing of a fuzzy controller using input and output mapping factors," IEEE Trans. Syst. Man Cyber., vol. 21, no.5, pp 952-960, Sept 1991
- [5] D. E. Goldberg, "Genetic Algorithms in Search, Optimization, and Machine Learning," Addison-Wesley, Reading, MA, 1989
- [6] M. A. Lee, H. Takagi, "Integrating Design Stages of Fuzzy Systems using Genetic Algorithms," IEEE Fuzz, pp 612-617, 1993
- [7] Li-Xin Wang, A Course in Fuzzy Systems and Control, Prentice Hall, pp 249-264, 277-288, 1997
- [8] Y. K. Ha and A. Tomizuka, "Fuzzy Global and Local Motion Control of an Inverted Pendulum Using Multiple Rule Bases," Proc. of Asian Control Conf., July, pp 27-30, 1994