

퍼지제어 시스템에서의 파라미터 동조방법

최종수 김성중* 권오신**

* 전북대학교 제어계측공학과

** 군산대학교 제어계측공학과

A Parameter Tuning Method in Fuzzy Control Systems

J. S. Choi S. J. Kim* O. S. Kwon**

* Dept. of Control & Instrumentation Eng.
Chunbuk National University** Dept. of Control & Instrumentation Eng.
Kunsan National University**Abstract**

This paper defines the relationship between PI type fuzzy control system and conventional PI control system, and discusses the relationship of parameters and control action in fuzzy controller. The tuning algorithm that updates output variable scaling factor of fuzzy controller is proposed. The proposed scheme is applied to the simulations of 2 selected dynamical plants. The simulation results show that the controller is effective in controlling dynamical plants.

I. 서 론

최근 몇년 사이에 퍼지제어는 퍼지 이론의 응용분야로서 가장 활발히 연구응용되어 왔다. 특히, 입출력 관계의 정량적인 분석이 어려운 여러가지 공정의 제어 시스템에서 모델을 기준으로한 기존 제어기의 부적합한 점을 보완 대체하여 성공적으로 도입되고 있다.

퍼지제어는 Zadeh에 의해서 제창된 퍼지집합 이론을 도입하여 1974년에 Mamdani가 퍼지제어를 적용한 이래로 퍼지제어는 공정제어, 열차의 자동운전 시스템, 지동 컨테이너 크레인 시스템, 엘리베이터 제어, 자동차 운송제어, 핵원자로 제어 등과 같은 영역에서 기존 시스템보다 우수한 것으로 밝혀졌다. 최근에 퍼지제어를 지원해주는 컴퓨터 하드웨어 기술(예를들면, 퍼지 VLSI Chips)의 진보는 세탁기, 진공청소기, 에어컨, 캠코더 등과 같은 많은 상업적인 퍼지제어 응용을 가져왔다.

기존의 제어기법에 비교하여, 퍼지제어는 다음과 같은 세가지의 중요한 장점을 갖는데, 첫째는 동등한 성능을 갖는 시스템을 개발하는 비용이 저렴하다는 것이고, 둘째는 퍼지제어가 PID제어가 극복할수 있는 동작조건 보다 더욱 넓은 영역의 동작조건을 극복하기 때문에 PID제어 보다 더 건설하다는 것

이다. 마지막으로 퍼지제어는 운전자의 전략을 모방할 수 있을 뿐만아니라, 자연어에서 이용되는 언어적 용어로 표현할 수 있기 때문에 주문에 따라 만들 수 있는(customizable) 장점을 갖는다.

이러한 점에도 불구하고, 퍼지제어의 한계점은 퍼지 제어기를 개발하기 위한 체계적인 방법론의 결핍에 있다고 할 수 있다. 이는 퍼지 제어기의 구성내용은 제어대상에 따라 달라질 수가 있고, 또한 안정된 제어기로서의 동작조건과 제어성능을 해석할 수 있는 이론적인 바탕이 완전하지 않기 때문이다. 또한, 퍼지 제어기의 동조가 기존의 제어기를 동조하는 것 보다 더 어렵다는 사실은 이미 잘 알려진 사실이다. 이 때문에 많은 연구자들이 퍼지 제어기를 동조하는 여러가지 방법을 연구해왔다. 그러나, 엔지니어에 대하여 이러한 동조방법을 사용하기 위한 가이드나 방법론이 아직까지 빈약한 상황이다. 어떤 연구자들은 퍼지 제어기의 최적값들을 찾기 위해 노력해왔다. 어떤 최적값은 공정과 제어목적의 특정한 모델에 의존하기 때문에 퍼지 제어기에 대한 최적값들을 얻을 수 있는 일반적인 방법은 없다. 그래서, 퍼지 제어기의 동조는 계산에 의해서가 아니라 제어하려는 플랜트에 대한 전문가의 지식(expert's knowledge)에 기반을 두고 행해져야 한다.

이 논문에서는 PI형 퍼지제어 시스템과 기존의 PI 제어 시스템과의 연관성을 유도하고, 파라미터와 제어동작 사이의 관계를 정의한다. 또한 제안한 파라미터 동조방법의 유용성을 보이기 위하여 1차 및 2차 플랜트에 적용하여 성능이 우수함을 보였다.

II. PI형 퍼지제어 시스템의 동조

2.1 파라미터와 제어성능 사이의 관계

PI형 퍼지 제어기는 입력 및 출력이 기존의 PI제어기와 동일하기 때문에 붙여진 이름이며, 대부분의 퍼지제어 시스템은

PI형 퍼지 제어기 형태를 하고 있다. PI퍼지 제어기의 출력 $u(t)$ (출력변수)와 입력 $e(t)$ (제어오차) 사이의 관계는 다음과 같이 간략하게 표현할 수 있다.

$$\Delta u(t)/Sdu = \Delta e(t)/Sde + e(t)/Se$$

$$u(t) = (Sdu/Sde)[e(t) + (Sde/Se)e(t)] \quad (2-1)$$

여기에서 Se , Sde , Sdu 는 각각 제어기 입력 및 출력변수에 대한 scaling factor이다.

반면에, 기존의 PI제어기는 다음과 같다.

$$u(t) = Kp[e(t) + (1/Ti)fe(t)] \quad (2-2)$$

여기에서 Kp 는 비례상수이고, Ti 는 PI제어기에 대한 적분시간이다. 식(2-1)과 (2-2)을 비교하면, 다음의 관계를 얻을 수 있다.

$$Kp = Sdu/Sde \quad (2-3)$$

$$Ti = (Se/Sde)T \quad (2-4)$$

여기에서 T 는 퍼지 제어기의 샘플링 시간이다. 식 (2-3)과 (2-4)를 기초로 PI형 퍼지 제어기의 scaling factor와 제어동작 사이의 관계를 표 2.1에 보였다.

표 2.1 Scaling factor와 제어동작

Scaling Factors	Error의 범위	ΔError의 범위	적분 시간	이득
Se ↑ Sde ↑ Sdu ↑	↑	↑	↑ ↓	↓ ↑

↑: increase ↓: decrease

PI형 퍼지 제어기를 이용하여 제어 시스템을 구성하면 그림 2.1과 같고, 시스템의 개루우프 전달함수 $G(s)$ 는 다음과 같다.

$$G(s) = Gc(s) \cdot Gp(s)$$

$$= K [(s + z)/s] Gp(s) \quad (2-5)$$

여기에서 $Gp(s)$ 는 제어하려는 플랜트의 전달함수이고, K 와 z 는 다음식과 같다.

$$K = Sdu / Sde \quad (2-6)$$

$$z = Sde / Se \quad (2-7)$$

그러므로, PI형 퍼지 제어기는 $(-Sde/Se, 0)$ 인 영점과 원점의 극점을 추가한 것이다.

일반적으로 z 는 근궤적의 형태에 영향을 끼치고, K 는 근궤적에서 특성근의 위치를 결정한다. 잘 알려진바와 같이 기존의 제어이론에서 특성근의 위치와 제어시스템의 성능 사이의 관계는 다음과 같다.

① 근의 실수부분이 작으면, 감쇄가 더욱 빨라지고 오버슈트가 작다.

② 근의 허수부분이 크면, 오실레이션 주파수가 높고 상승 시간이 짧다.

사실, PI형 퍼지 제어기를 동조할때 실질적으로 z 와 K 의 두 파라미터만이 동조되며, 이것은 scaling factor를 동조하는 것이다.

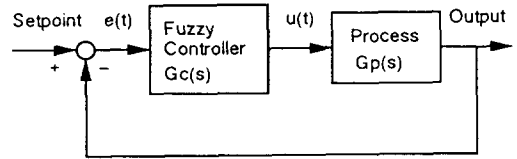


그림 2.2 PI형 퍼지제어 시스템의 블록선도

2.2 파라미터의 동조

단일입력 단일출력(SISO) 시스템을 고려해보자. PI형 퍼지 제어기는 두 개의 제어입력(control inputs)을 갖는다. 즉, 현재의 오차($e(k)$)와 그 변화량($de(k)$)이다. 제어기에 의해서 생성된 제어동작(control action)은 조절되는 변수 $u(k)$ 의 변화량 $du(k)$ 이며 아래와 같이 정의한다.

$$e(k) = sp(k) - y(k) \quad (2.8)$$

$$de(k) = e(k) - e(k-1) \quad (2.9)$$

$$du(k) = u(k) - u(k-1) \quad (2.10)$$

여기에서 $sp(k)$ 와 $y(k)$ 는 각각 k 번째 샘플링 시간의 목표치와 플랜트 출력이다.

소속함수와 퍼지제어 규칙은 각각 그림 2.3과 표 2.2에 나타났다. 입출력 변수에 대한 scaling factor는 각각 Se , Sde , Sdu 이며, 이 scaling factor는 소속함수에서 각 퍼지변수의 전체집합을 정의하는 요소이다. 따라서 입출력 변수값의 실제 범위를 아래의 식과 같이 결정해 주며, 소속함수의 scaling factor가 변화할 경우 각 퍼지변수의 Support값은 동일한 비율로 변화하게 된다.

$$-Se \leq e(t) \leq Se \quad (2.11)$$

$$-Sde \leq \Delta e(t) \leq Sde \quad (2.12)$$

$$-Sdu \leq \Delta u(t) \leq Sdu \quad (2.13)$$

퍼지 제어기의 입력과 출력에 적용된 scaling factors는 시스템의 전체적인 성능을 기반으로 원하는 응답을 얻기 위해서 동조된다. Yen[6] 등은 동조의 성능평가 기준으로써 오버슈트(OV), 도달시간(RT) 및 공정의 과도응답에 대한 진폭의 크기(OSC)를 이용하여 단지 입력변수에 대한 scaling factor만 동조하였다. 동조에 이용된 규칙의 형태는 다음과 같다.

IF Performance-measure is X1, THEN ΔSe is Y1
 IF Performance-measure is X2, THEN ΔSde is Y2

여기에서 performance-measure는 위에 언급한 세가지 평가 기준 중의 하나이다. X는 평가 기준에 대한 퍼지집합이고, Y는 scaling factor를 조절하는 퍼지집합이다.

이 방법은 규칙들이 서로 상충되는 문제점이 발생하는데, 이것을 해결하기 위해서 퍼지추론을 거쳐 적당한 값을 찾아야 하는 어려움이 있다. 예를들어, ΔOV와 ΔRT가 모두 negative 라면, 규칙에서 ΔSe가 positive 그리고 negative인 경우.

또한, 입출력 scaling factor 중 일반적으로 제어 시스템의 성능에 가장 큰 영향을 미치는 파라미터는 제어기의 이득을 결정하는 출력에 대한 scaling factor인 Sdu이다. 따라서 이 논문에서는 입력 scaling factor는 고정시키고, 단지 출력 scaling factor만 동조과정에서 동조한다.

이 동조 기법에서 성능평가의 기준은 오버슈트이다. 시스템에서 오버슈트가 발생하게 되면 제어기의 출력을 줄여야 할 것

이다. 즉, 출력변수에 대한 scaling factor인 Sdu를 감소시켜야 한다. 이 논문에서는 복잡한 추론과정을 거치지 않고 Sdu를 다음과 같이 매 샘플링 시간에 조절된다.

$$Sdu(k) = Sdu(k-1) + \Delta Sdu(k) \quad (2.14)$$

이 때 ΔSdu(k)는 ΔSdu(k) = G·E(k) 이다. 여기에서 G는 임의의 이득이고, E(k)는 오버슈트가 발생할 때의 오차이다. 위의 동조과정은 그림 2.4에 나타났다.

III. 모의실험 결과

모의실험은 다음과 같은 1차 및 2차 플랜트에 각각 적용하였다.

$$\text{모델 1 : } G(s) = \frac{1}{s + 20}$$

$$\text{모델 2 : } G(s) = \frac{e^{-ds}}{(s + 1)(s + 0.5)}$$

이 때의 샘플링 시간은 1(sec)로 하였다. 퍼지 제어기는 그림 2.3과 표 2.2의 소속함수와 퍼지제어 규칙을 기반으로 구성되고, 추론결과는 실시간 퍼지추론을 통해서 구해진다. 비퍼지화 방법은 아래의 식과 같이 정의되는 COA(Center-of-Area) 비퍼지화 알고리즘을 적용하였다.

$$du = \frac{\sum \mu(ui) \cdot ui}{\sum \mu(ui)}$$

입력변수의 scaling factor는 Se, Sde 모두 1.0으로 고정하고, 매 샘플링 마다 출력변수의 scaling factor Sdu만이 조절된다. 그림 3.1은 1차 플랜트(모델 1)에 대한 모의실험 결과이고, 그림 3.2부터 3.5는 2차 플랜트(모델 2)에 대한 결과이다. 그림 3.1은 출력변수 scaling factor를 변화시킨 경우로써 우수한 결과를 보였다. 그림 3.2는 출력변수 scaling factor를 적당히 정해줌으로써 우수한 결과를 보였지만, 그림 3.3(a)와 3.4(a)는 scaling factor가 너무 커짐에 따라 오실레이션이 발생하여 불안정한 시스템 응답을 보이고 있다. 그러나 이 논문에서 제안한 알고리즘으로 출력변수를 조정하여 오

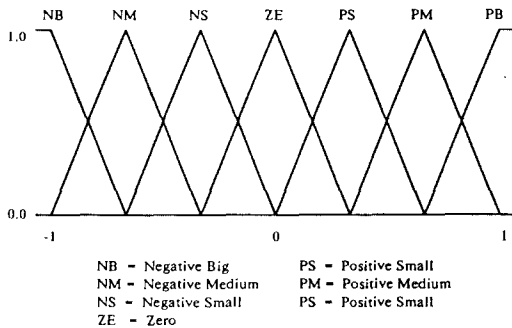


그림 2.3 입출력 변수에 대한 소속함수

표 2.2 퍼지제어 규칙

	Δ Error						
	NB	NM	NS	ZO	PS	PM	PB
NB	NB	NB	NB	NB	NM	NS	ZO
NM	NB	NB	NM	NM	NS	ZO	PS
NS	NB	NM	NS	NS	ZO	PS	PM
ZO	NM	NM	NS	ZO	PS	PM	PM
PS	NM	NS	ZO	PS	PS	PM	PB
PM	NS	ZO	PS	PM	PM	PB	PB
PB	ZO	PS	PM	PB	PB	PB	PB

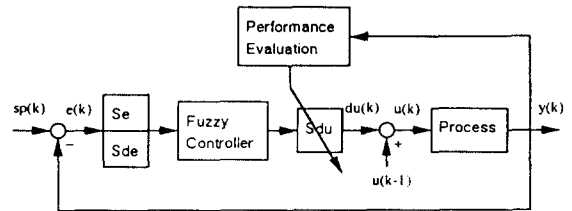


그림 2.4 동조과정의 블럭선도

실레이션을 제거한 시스템 응답을 얻을 수 있었다. 이 때의 이득(G)은 0.5로 정하였다. 그림 3.5는 시스템에 지연시간이 존재하는 경우로써 발생하는 오버슈트와 오실레이션을 효과적으로 제거하여 안정한 응답을 얻었다.

IV. 결 론

이 논문에서는 PI형 퍼지제어 시스템과 PI제어 시스템 사이의 연관성을 정의하였고, 파라미터와 제어동작과의 관계를 알아보았다. 또한 scaling factors 중 입력변수에 대한 scaling factor는 고정시키고, 시스템의 성능에 가장 많은 영향을 미치는 출력변수 scaling factor만을 동조하는 알고리즘을 제안하여 1차 및 2차 플랜트 모델에 적용한 결과 우수한 시스템 응답을 얻을 수 있었다.

참 고 문 헌

- [1] K.L. Tang and R.J. Mulholland, "Comparing Fuzzy Logic with Classical Controller Designs," *IEEE Trans. on Systems, Man and Cybernetics*, Vol. SMC-17, No.6, pp.1085-1087, 1987.
- [2] C.C. Lee, "Fuzzy Logic in Control Systems : Fuzzy Logic Controller - Part I, II" *IEEE Trans. on Systems, Man and Cybernetics*, Vol.SMC-20, No.2, pp.404-432, 1990.
- [3] W.H. Bare, R.J. Mulholland, and S.S. Sofer, "Design of a Self-Tuning Rule Based Controller for a Gasoline Refinery Catalytic Reformer," *IEEE Trans. on Automatic Control*, Vol.35, No.2, pp.156-164, 1990.
- [4] P.J. MacVicar-Whelan, "Fuzzy Sets for Man-Machine Interaction," *Int. J. Man-Machine Studies*, Vol.8, pp.687-697, 1976.
- [5] L. Zheng, "A Practical Guide to Tune of Proportional and Integral(PI) like Fuzzy Controllers," *IEEE International Conference on Fuzzy Systems*, Vol.1, pp.633-640, March, 1992.
- [6] W.C. Daugherty, B. Rathakrishnan and J. Yen, "Performance Evaluation of a self-Tuning Fuzzy Controller," *IEEE International Conference on Fuzzy Systems*, Vol.1, pp.389-397, March, 1992.
- [7] M. Maeda, T. Sato, and S. Murakami, "Design of the self-tuning controller," In *proceedings of the International conference on Fuzzy Logic & Neural Networks*, pp. 187-191, July 1990, Iizuka Japan.

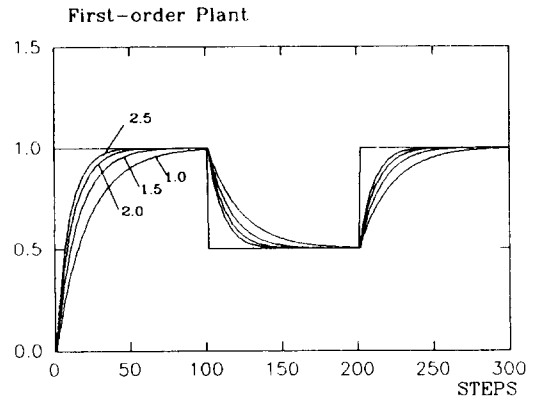


그림 3.1 Sdu를 변화시킨 경우(Model 1)

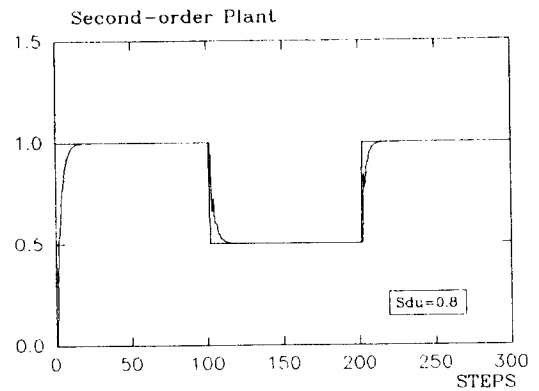
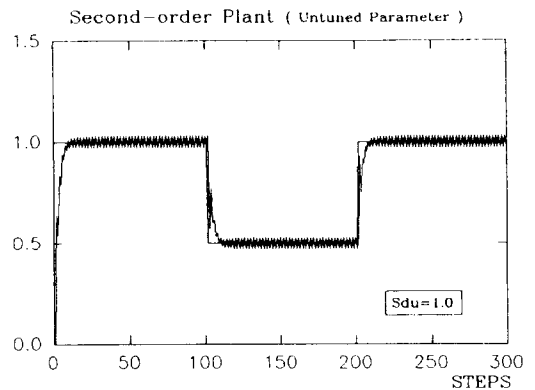
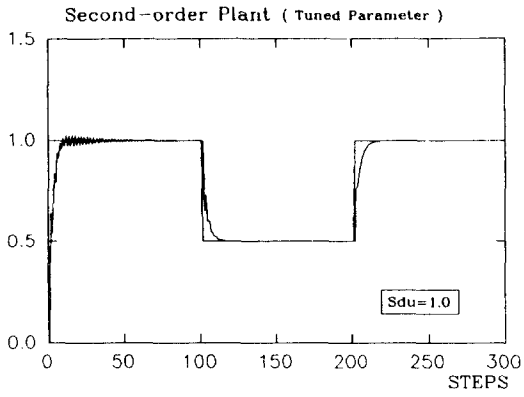


그림 3.2 Sdu=0.8인 경우의 결과(Model 2)



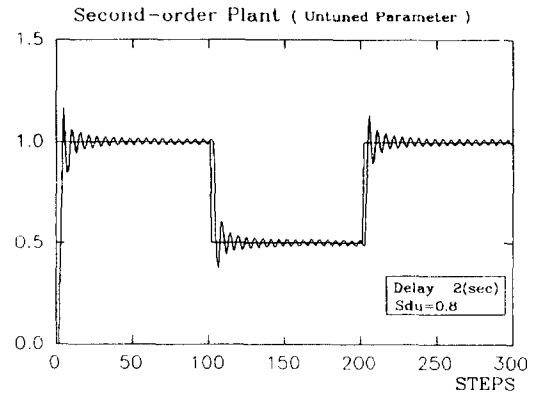
(a)

그림 3.3 Sdu=1.0인 경우의 결과(Model 2)

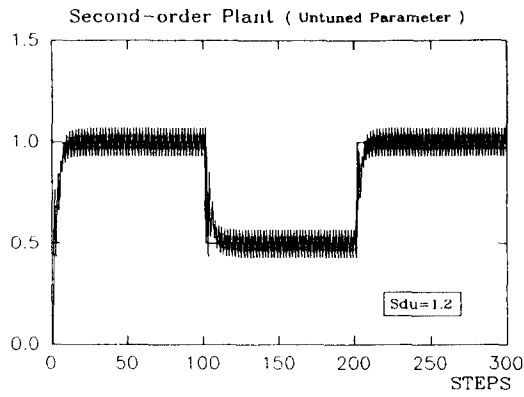


(b)

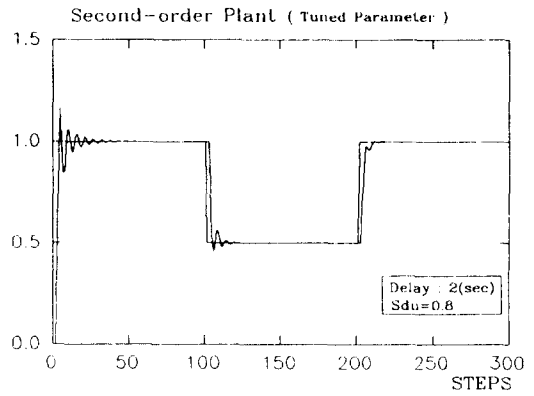
그림 3.3 Sdu=1.0인 경우의 결과(Model 2)



(a)

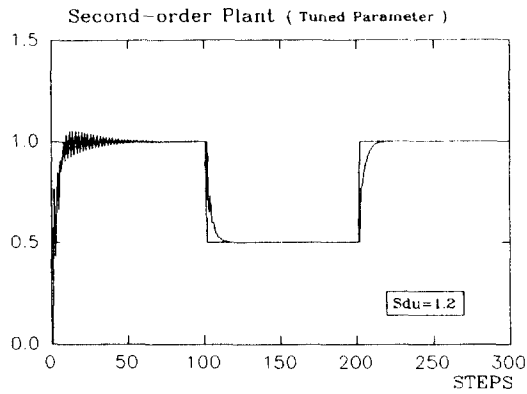


(a)



(b)

그림 3.5 지연시간이 존재하는 경우의 결과(Model 2)



(b)

그림 3.4 Sdu=1.2인 경우의 결과(Model 2)