

## 퍼지 PID 제어기 설계 및 자동 동조

°임 정훈\*, 임 형순\*, 이 창구\*  
\*전북대학교 전자 정보 공학부

### Fuzzy PID Controller Design and Auto Tuning

°Jeong-Heum Im\*, Hyung-Soon Lim\*, Chang-Goo Lee\*  
\*School of Electronics & Information, Chonbuk Nat. Univ.

**Abstract** - In this paper, we re-analyzed the fuzzy controller as conventional PID controller structure, and proposed a self tuning fuzzy PID controller whose input output scaling factors were tuned automatically. At first stage, the tuning parameters of fuzzy controller were determined by Ziegler-Nichols tuning method and then they were adjusted as the delay time and process environment were changed. Proposed controller was simple in its structure and computational burden was small so that on line adaptation was easy to apply to. The result of computer simulation and practical experiment showed the proposed controller's excellent performance

#### 1. 서 론

PID 제어기는 구조가 단순하면서도 목표치의 추적, 외란 효과의 감쇄, 프로세스와 모델계수의 불일치에 대한 강인성 등 우수한 성능으로 산업 현장에서 널리 사용되고 있다. 그러나 PID 제어기는 운전중 지연시간이 변하거나, 공정환경이 변화 할 경우 성능이 급격히 저하되며 비선형 시스템이나 큰 지연시간을 갖는 시스템에는 적합하지 못하다. 이러한 문제의 해결을 위해 PID 이득을 동조시키는 연구가 다양하게 이루어져 왔다. PID 이득 동조법으로 주로 사용되는 Ziegler Nichols 동조법이나 릴레이 궤환법<sup>[1, 2])</sup>은 시스템에 대한 사전정보를 요구하지 않으며 건실성이 우수하다는 좋은 장점을 가져 초기 동조에는 적합하나 온라인 동조에는 적합하지 못하며 신경 회로망이나 퍼지 등을 이용한 방법<sup>[3-6])</sup>들은 방대한 계산량과 복잡한 구조에도 불구하고 초기 값과 학습계수 설정에의 민감성, 오프라인 학습 필요성 등 여전히 많은 문제를 안고 있다.

본 논문에서는 기존 PID제어기의 동조법과 퍼지제어기법을 동시에 사용할수 있도록 제어기 구조를 변경하고 입출력 조정변수를 동조시켜 적용성을 갖게 함으로써 운전중 지연시간의 변화나 공정계수의 변화에 보다 강인한 적응 퍼지 PID 제어기를 설계하였다. 제시한 제어기는 구조가 간단하고 계산량이 적어 실시간 제어가 가능하며 모의실험 및 실제 시스템에 적용하여 매우 우수한 성능을 확인하였다.

#### 2. 퍼지 PID 제어기 설계 및 동조

##### 2.1 퍼지 PID 제어기의 설계

선형 퍼지 제어기의 설계 단계는 다음과 같다.

**Step 1 :** 퍼지 로직의 두 입력  $a$ 와  $b$ 를 구간  $[-1, 1]$  사이에서 정규화시킨다.

**Step 2 :** 퍼지 집합  $A$ 와  $B$ 를  $2N+1$ 개의 언어 변수로 분할한다. ( $N$  : 양의 정수 )

$$A_i \ (i \in I = [-N, \dots, -2, -1, 0, 1, 2, \dots, N])$$

$$B_j \ (j \in J = [-N, \dots, -2, -1, 0, 1, 2, \dots, N])$$

**Step 3 :** 인접 멤버십 함수와 50% 중첩이 되는 대칭 삼각 멤버십 함수를 구성한다.

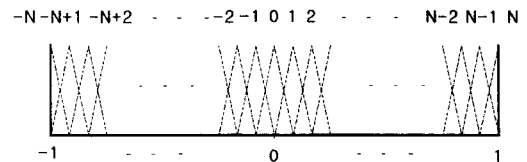


그림 1. 멤버십 함수 (A & B)

**Step 4 :** 퍼지 제어 규칙은 다음과 같이 주어진다.

$$\text{if } a \text{ is } A_i \text{ and } b \text{ is } B_j, \text{ then } u \text{ is } u_{i,j}$$

여기서  $u_{i,j}$  ( $i \in I, j \in J$ )는 퍼지 부분집합이 아닌 실제 값이다.

표 1. 퍼지 규칙표

A \ B	-N	-N+1	..	N-2	N-1	N
-N	-2	$-2 + \frac{1}{N}$	..	$-\frac{2}{N}$	$-\frac{1}{N}$	0
-N+1	$-2 + \frac{1}{N}$	$-2 + \frac{2}{N}$	..	$-\frac{1}{N}$	0	$\frac{1}{N}$
-N+2	$-2 + \frac{2}{N}$	$-2 + \frac{3}{N}$	..	0	$\frac{1}{N}$	$\frac{2}{N}$
:	:	:	:	:	:	:
N-2	$-\frac{2}{N}$	$-\frac{1}{N}$	..	$2 - \frac{4}{N}$	$2 - \frac{3}{N}$	$2 - \frac{2}{N}$
N-1	$-\frac{1}{N}$	0	..	$2 - \frac{3}{N}$	$2 - \frac{2}{N}$	$2 - \frac{1}{N}$
N	0	$\frac{1}{N}$	..	$2 - \frac{2}{N}$	$2 - \frac{1}{N}$	2

**Step 5 :** Product-sum 추론법과 Center of gravity 비퍼지화 기법을 사용한다.

Product-sum 추론법을 이용한 퍼지 제어기 전전부의 실제 값은 다음과 같다.

$$f_{i,j} = A_i(a)B_j(b) \quad (i \in I, j \in J) \quad (1)$$

Center of gravity를 적용하면 제어기의 실제 출력  $u$ 는 다음과 같이 주어진다.

$$u = \frac{\sum_{i,j} f_{i,j} u_{i,j}}{\sum_{i,j} f_{i,j}} \quad (2)$$

$$A_i(a) = \frac{a_{i+1}-a}{a_{i+1}-a_i}, \quad A_{i+1}(a) = \frac{a-a_i}{a_{i+1}-a_i},$$

$$A_k(a) = 0 \quad (k \neq (i, i+1) \in I), \quad \text{for } a \in [a_i, a_{i+1}]$$

$$B_j(b) = \frac{b_{j+1}-b}{b_{j+1}-b_j}, \quad B_{j+1}(b) = \frac{b-b_j}{b_{j+1}-b_j},$$

$$B_t(b) = 0 \quad (t \neq (j, j+1) \in J), \quad \text{for } b \in [b_j, b_{j+1}]$$

임의의 순간 최대 4개의 rule이 발생하며 방정식 (1) 과 (2)를 결합하여 다음과 같이 쓸 수 있다.

$$u = \frac{\sum_{k=(i, i+1), t=(j, j+1)} (A_k(a)B_t(b))u_{k,t}}{\sum_{k=(i, i+1), t=(j, j+1)} A_k(a)B_t(b)} \quad (3)$$

$$\begin{aligned} & \sum_{k=(i, i+1), t=(j, j+1)} A_k(a)B_t(b) \\ &= (A_i(a) + A_{i+1}(a))(B_j(b) + B_{j+1}(b)) \\ &= 1 \end{aligned}$$

그러므로 퍼지 제어기의 출력은 다음과 같다.

$$\begin{aligned} & \sum_{k=(i, i+1), t=(j, j+1)} (A_k(a)B_t(b))u_{k,t} \\ &= \left(\frac{a_{i+1}-a}{a_{i+1}-a_i}\right)\left(\frac{b_{j+1}-b}{b_{j+1}-b_j}\right)u_{i,j} + \\ & \left(\frac{a-a_i}{a_{i+1}-a_i}\right)\left(\frac{b_{j+1}-b}{b_{j+1}-b_j}\right)u_{(i+1),j} + \\ & \left(\frac{a_{i+1}-a}{a_{i+1}-a_i}\right)\left(\frac{b-b_j}{b_{j+1}-b_j}\right)u_{i,(j+1)} + \\ & \left(\frac{a-a_i}{a_{i+1}-a_i}\right)\left(\frac{b-b_j}{b_{j+1}-b_j}\right)u_{(i+1),(j+1)} \end{aligned}$$

$$a_{i+1} = a_i + \frac{1}{N}, \quad b_{j+1} = b_j + \frac{1}{N},$$

$$u_{i+1,j} = u_{i,j+1} = u_{i,j} + \frac{1}{N}, \quad u_{i+1,j+1} = u_{i,j} + \frac{2}{N}$$

$$a = a_i, \quad b = b_j \quad \text{일 때 } u = u_{i,j} = a_i + b_j = a + b \quad \text{이며}$$

$$a = a_i + \delta a, \quad 0 < \delta a < \frac{1}{N}, \quad b = b_j + \delta b, \quad 0 < \delta b < \frac{1}{N}$$

일 때 출력은  $u = u_{i,j} + \delta a + \delta b = a + b$  이다.

Step 1~5와 같이 퍼지 로직을 구성하면 정규화된 두 입력  $a(t)$ 와  $b(t)$ 에 대하여 일반적인 퍼지 제어기는  $u(t) = a(t) + b(t)$ 로 나타낼 수 있다.

일반적으로 사용되는 PID 제어기의 구조는 그림 2와 같으며, 앞서 기술한 퍼지 로직을 이용한 퍼지 PID 제어기는 구조는 그림 3과 같다.

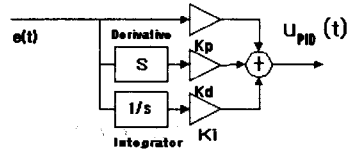


그림 2. PID 제어기 구조

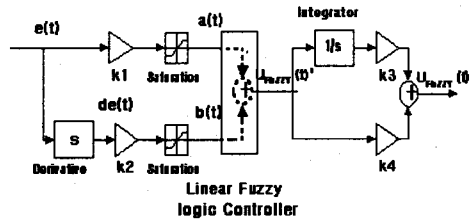


그림 3. Fuzzy PID 제어기 구조

이산시간 제어에서 제어기의 출력은 다음과 같다.

$$u_{PID}(n) = [K_p + K_d \frac{z-1}{Tz} + K_i \frac{Tz}{z-1}]e(n)$$

$$u_{FUZZY}(n)' = (k_1 + \frac{k_2}{T} \frac{z-1}{z})e(n)$$

$$u_{FUZZY}(n) = [k_3 \frac{Tz}{z-1} + k_4]u_{FUZZY}(n)'$$

$$= (k_1 k_4 + k_2 k_3)e(n) + k_2 k_4 \frac{z-1}{Tz} e(n) + k_1 k_3 \frac{Tz}{z-1} e(n)$$

$T$ : 샘플링 주기

PID제어기의 출력과 퍼지 제어기의 출력이 같기 위해서는 주어진 PID계수에 대하여 다음식을 만족하여야 한다.

$$\begin{aligned} K_p &= k_1 k_4 + k_2 k_3, \\ K_d &= k_2 k_4, \quad K_i = k_1 k_3. \end{aligned} \quad (4)$$

또한  $k_1, k_2, k_3$  그리고  $k_4$ 는 양의 실수 값이 되어야 하므로 주어지는 PID계수들은 다음의 식을 만족하여야 한다.

$$K_p^2 \geq 4K_i K_d \quad (5)$$

식 (5)를 만족시키는 임의의 PID 제어기에 대하여 동일한 출력을 만들어 내는 퍼지 제어기를 구성할 수 있으며 이때  $[k_1, k_2, k_3, k_4]$ 의 순서쌍은 무한개 존재한다. 앞 절에서 퍼지 로직의 입력  $a(t)$ 와  $b(t)$ 는  $[-1, 1]$ 사이에서 정규화 시켰다. 그러므로  $a(t)$ 와  $b(t)$ 를  $[-1, 1]$ 사이에서 제한되게 하는 모든  $[k_1, k_2, k_3, k_4]$ 의 순서쌍에 대하여 PID 제어기와 동일한 성능을 가지는 퍼지 PID 제어기를 임의의 양의 정수  $N$ 에 대하여 그림 3과 같은 구조로 구현할 수 있다.

이러한 형태의 제어기 구조를 이용함으로써 기존 PID 제어기의 동조법을 동일하게 사용하여 초기 값 설정문제를 해결할 수 있으며 퍼지 제어 기법을 쉽게 접목시킬 수 있도록 하였다.

## 2.2 퍼지 PID 제어기의 동조

지연시간이나 동특성 변화에 대하여 적응성을 갖도록 입력력 조정 이득을 동조 시키는 방법을 적용하였다.

Ishida는 신경 회로망을 이용하여 기존의 PID제어기와 비슷한 구조를 갖는 직접 적응 PID제어기 구조를 제시하여 초기값 설정의 문제를 해결하고 적응성을 부여하였다.<sup>[3]</sup> Chen은 시스템 오차의 변화에 따라 퍼지 규칙 평면을 조정하게 하여 제어 성능을 향상 시켰다.<sup>[5]</sup> 이와 동일한 효과를 내도록 입력력 조정 변수가 시스템 오차의 변화에 따라 자동 동조되도록 하였다.

출력 조정 이득은 Gradient descent방법에 의하여 제어 성능지수 ( $J$ )를 최소화 하도록 자동 동조 된다.

$$J = \frac{1}{2} [r(n) - y(n)]^2$$

$$k_3(n+1) = k_3(n) - \eta \frac{\partial J}{\partial k_3} \Big|_n$$

$$k_4(n+1) = k_4(n) - \eta \frac{\partial J}{\partial k_4} \Big|_n$$

$r(n)$  : 기준입력,  $y(n)$  : 시스템 출력,  $\eta$  : 학습계수

입력 조정 이득은 정규화된 오차  $a(n)$ 에 의하여 결정 된다.

$$k_1(n) = 2\alpha k_1(0), \quad k_2(n) = 2(1-\alpha)k_2(0)$$

$$\alpha = |a(n)|(a_2 - a_1) + a_1$$

$$0 \leq \alpha_1 \leq \alpha_2 \leq 1, \quad \alpha \in [0, 1]$$

$k_1(0)$  과  $k_2(0)$ 은 각각 입력 조정 이득의 초기값 이며  $\alpha_1 = \alpha_2 = 0.5$  일 때  $k_1(n) = k_1(0)$ ,  $k_2(n) = k_2(0)$  이다.

## 3. 모의실험 및 Magnetic Levitation 제어 실험

모의 실험은 MATLAB SIMULINK를 사용하였다. 비선형 성분 보상 실험에 있어서는 Feedback instruments Ltd.의 Magnetic Levitation System 33-320, DSP6031 보드와  $\mu$ -Pass/C31를 사용하여 실험하였다. 자기 부상 시스템은 전자기 코일에 전류(전압)을 인가하여 Steel ball의 수직 위치를 제어하며 ball의 위치를 측정하기 위해 적외선 광센서가 사용된다.

$$\text{Plant 1 : } G_p(s) = \frac{e^{-Ls}}{s(s+1)}$$

$$K_p = 2.07, \quad T_i = 1.80, \quad T_d = 0.45, \quad L = 0.1$$

$$\text{Plant 2 : } G_p(s) = \frac{1 - \rho s}{(s+1)^3}$$

$$K_p = 0.92, \quad T_i = 2.33, \quad T_d = 0.5825, \quad \rho = 1.4$$

그림 4는 2.2절에서 기술한 Ishida의 직접 적응 신경망 제어기의 공정중 지연시간 변화에 따른 응답이다. 지연시간의 변화가 적을 경우에는 어느 정도 적응성을 보이지만 지연시간이 큰 경우 적응성이 없는 PID 제어기와 마찬가지로 발산하였다. 적응 퍼지 PID (Adaptive Fuzzy PID, AFPID) 제어기는 운전중 지연시간의 변화에 대하여 훨씬 강인한 성능을 보였다.(그림 5)

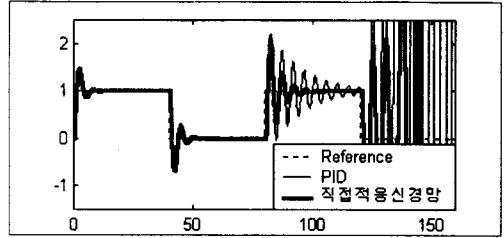


그림 4. 지연시간 변화시 적응신경망 제어기 응답  
L=0.1→0.3 (40s), L=0.3→0.5 (80s),  
L=0.5→0.7 (120s) : Plant1

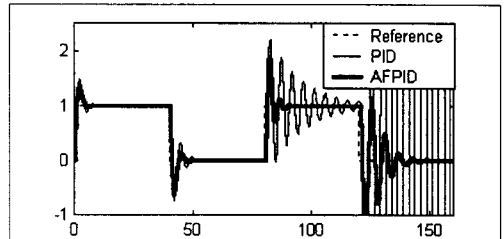


그림 5. 지연시간 변화시 AFPID 제어기 응답  
L=0.1→0.3 (40s), L=0.3→0.5 (80s),  
L=0.5→0.7 (120s) : Plant1

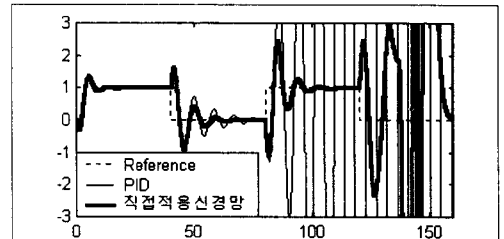


그림 6. 공정계수 변화시 적응 신경망 제어기 응답  
 $\rho = 1.4 \rightarrow 2.0$  (40s),  $\rho = 2.0 \rightarrow 2.6$  (80s),  
 $\rho = 2.6 \rightarrow 3.2$  (120s) : Plant2

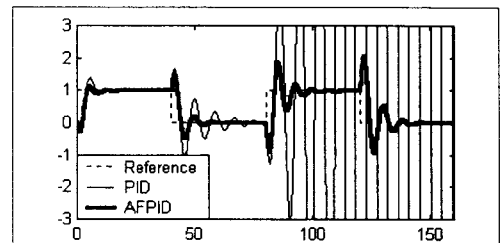


그림 7. 공정계수 변화시 AFPID 제어기 응답  
 $\rho = 1.4 \rightarrow 2.0$  (40s),  $\rho = 2.0 \rightarrow 2.6$  (80s),  
 $\rho = 2.6 \rightarrow 3.2$  (120s) : Plant2

그림 6, 7은 운전중 공정계수의 변화에 따른 직접 적응 신경망 제어기와 AFPID 제어기의 응답이다. 직접 적응 신경망 제어기의 경우 지연시간의 변화에 대한 실험과 마찬가지로 공정계수의 적은 변화에는 적응성을 보

이지만 변화가 클 경우 불안정해졌으나 AFPID 제어기는 운전중 공정계수의 변화에 보다 강인하였다.

그림 8, 9는 실제 자기부양 시스템에 기존 PID 제어기와 본 논문에서 제안한 AFPID 제어기를 적용하였을 때의 응답을 보여준다. 자기 부양 시스템은 개 루프 불안정 시스템이며 비선형 특성이 강하다. PID제어기는 동작점에서 선형화한 모델식에 기반하여 설계된다.<sup>(8)</sup> PID 제어기는 선형화한 모델식에 대하여서는 상당히 좋은 성능을 보이지만 실제 시스템에 적용시 모델화 되지 않은 비선형 성분 때문에 그림 8과 같은 응답을 보였다. 반면에 AFPID 제어기는 비 선형 성분 보상 성능이 우수함을 볼 수 있다.

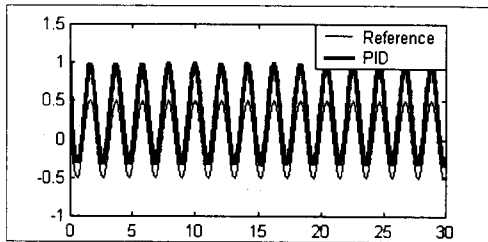


그림 8. PID 제어기 응답 : 자기부양 시스템  
 $K_p = 3, K_i = 0.01, K_d = 0.1$

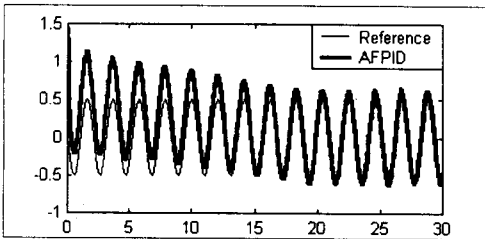


그림 9. AFPID 제어기 응답 : 자기부양 시스템

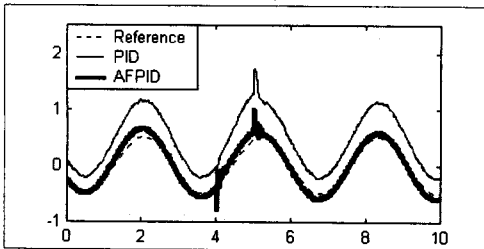


그림 10. 외란 인가시(0.4 : 4-5s) PID 제어기와 AFPID 제어기 응답 : 자기부양 시스템

그림 10과 11은 외란에 대한 PID와 AFPID제어기의 응답이다. 4~5초 구간동안 각각 0.4(그림 10), 0.5(그림 11)크기의 외란이 인가 되었다. 외란에 대한 성능은 PID 제어기와 AFPID 제어기가 비슷한 성능을 보였다. 그러나 AFPID 제어기의 경우 비선형 성분을 보상하여 기준 입력과의 오차를 줄임으로써 상대적으로 허용 가능한 외란의 크기가 커졌다.(그림 11)

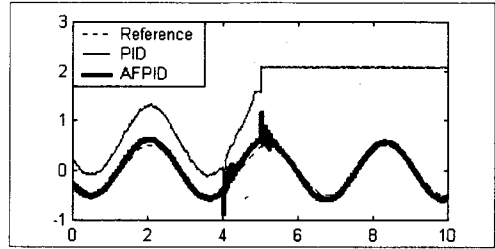


그림 11. 외란 인가시(0.5 : 4-5s) PID 제어기와 AFPID제어기 응답 : 자기부양 시스템

#### 4. 결 론

본 논문에서는 퍼지 로직의 선형성과 Gradient Decent 기법 그리고 퍼지 제어 기법 등을 이용한 적응 퍼지 PID 제어기를 설계하였다. 제안된 제어기는 PID 제어기와 수식적으로 동일한 퍼지 제어기를 구현하므로써 PID 제어기의 동조법을 사용하여 초기 이득을 결정하고 퍼지 제어 기법을 쉽게 적용할 수 있으며 간단한 구조와 적은 계산량으로 실시간 제어가 가능하였다.

모의 실험 결과 제안된 제어기는 운전중 지연시간이나 공정계수의 변화에 PID 제어기를 비롯한 기존의 제어기에 비하여 훨씬 우수한 성능을 보였으며 개루프 불안정 비선형 특성을 지닌 자기 부양 시스템에 실제 적용하여 비선형 성분의 보상 성능과 외란에 대한 강인성을 확인하였다.

#### (참 고 문 헌)

- [1] Derek P. Atherton, "PID Controller Tuning", Computing & Control Engineering Journal, vol. 10, pp. 44-50, 1999
- [2] Wang QG, Hang CC, Bi Q, "A Frequency domain Controller Design Method", Chemical Engineering Research & Design, vol. 75, pp. 64-72, 1997
- [3] Y. Ishida "Nonlinear PID Controller Using Neural Networks", IEEE International Conf. on Neural Networks, pp. 811-814, 1997
- [4] M. Mizumoto, "Realization of PID Controls by Fuzzy Control Methods", Fuzzy Sets and Systems, vol. 70, pp. 171 -182, 1995
- [5] Minyou Chen, D.A. Linkens, A hybrid neuro-fuzzy PID controller, Fuzzy Sets And Systems, vol. 99, pp. 27-36, 1998
- [6] Wu Zhi Qiao, Masaharu Mizumoto, PID type Fuzzy Controller and Parameters Adaptive method, Fuzzy Sets And Systems, vol. 78, pp. 23-35, 1996
- [7] S. Gallchet and L Foulloy, "Fuzzy Controllers: Synthesis and equivalences", IEEE Trans. Fuzzy Syst., vol. 3, pp. 140-148, 1995
- [8] Haley, Pamela J., Soloway, Don, Gold, Brian, "Real-time Adaptive Control Using Neural Generalized Predictive Control", Proceedings of the 1999 American Control Conference, vol. 6, pp. 4278-4282, 1999