

論文93-30B-5-13

제어량의 크기를 조건부에 포함하는 PI형 퍼지제어기

(A PI-Type Fuzzy Controller Taking Control Input into Conditional Part of Rules)

李志弘*

(Ji Hong Lee)

要約

PI형 퍼지제어기는 정상상태오차를 없애는 목적에는 유용하지만 적분기의 작용에 의해 2차 이상의 시스템에 대해서 오버슈트와 라이즈 타임의 개선에 한계를 갖고 있다는 단점을 극복하기 위해 퍼지제어기의 조건부에 해당시각에 가해지고있는 제어입력의 크기를 포함하는 규칙을 추가한 형태의 퍼지 PI 제어기를 제안한다. 제안된 제어기는 PI 제어기만으로는 2차 이상의 제어 대상 시스템에 대해서는 필요한 만큼의 제어입력을 빨리 변화시키지 못해 좋은 응답을 주지 못하는 경우가 많다는 점에 착안하여 일단 측정된 오차와 오차의 변화율로부터 제어입력의 증분을 계산하며 측정된 오차와 해당시각에 가해지고있는 제어입력으로부터 쌓여있는 제어입력의 양에서 남겨놓을 부분을 결정하도록 하였다. 즉, 경우에 따라 과도하게 누적된 제어입력을 리셋시킴으로써 PI형 제어기에 의한 불필요한 진동을 없애도록 하였다. 제안된 방법은 시뮬레이션을 통하여 기존의 퍼지제어기로 제어한 경우와 비교하여 그 유용성을 보였다.

Abstract

To improve limitations of fuzzy PI controllers especially when applied to systems of order higher than one, we propose a fuzzy PI controller that takes out appropriate amounts of accumulated control input according to fuzzily described situations in addition to the calculation of incremental control input as in the case of conventional fuzzy PI controllers. The structure of the proposed controller was motivated by the characteristics of fuzzy PI controller that it generally gives inevitable overshoot when one tries to reduce rise time of the response especially when a system of order higher than one is under consideration. Since the undesirable characteristics of the fuzzy PI controller is caused by integrator of the controller, even though the integrator is introduced to overcome steady state error of response, we propose a controller that fuzzily clears out integrated quantities according to situation to give reduced rise time as well as small overshoots. To show the usefulness of the proposed controller, it is applied in simulations to such systems as are difficult to stabilize or difficult to get satisfactory responses by conventional fuzzy PI controllers.

1. 서론

* 正會員, 金烏工科大学, 電子制御工學科
(Dept. of Cont. Eng., Kumoh Nat'l Inst. of
Tech. Univ.)

接受日字: 1992年 10月 15日

Mamdani^[3,4] 은 Zadeh^[1,2] 에 의해 제안된 퍼지이론을 제어문제에 성공적으로 도입하였으며 그 후로도 퍼지제어에 관한 많은 이론적 진전이^[5,6,7] 있었

으며 또한 많은 성공적인 응용사례가 보고되고 있다. [5] 이러한 많은 응용을 살펴보면 크게 두가지 부류로 분류가 가능한데 하나는 입출력 변수의 수가 아주 많거나 시스템의 모델링이 아주 어려워 기존의 제어 방법으로 다루기 곤란한 경우이고 다른 하나는 시스템은 간단하지만 기존의 제어 방법에서의 성능향상에 한계가 있는 경우이다. 전자에 대한 응용의 이유로는 대상시스템에 대한 여러 정보가 전문가로부터 언어적 정보로 입수 가능하기만하면 제어기의 골격이 쉽게 퍼지제어기로 변환될 수 있다는 퍼지제어기가 갖고있는 고유한 특성때문이라고 볼 수있으며, 후자의 경우는 퍼지제어기의 성능에 관계하는 퍼지화 방법, 추론 규칙, 비퍼지화 방법, 그리고 언어적으로 묘사되는 각 변수의 모양등에 의해 쉽게 비선형적인 제어기를 설계할 수있기 때문이라고 볼 수있다. 후자의 경우는 대상 시스템이 간단하므로 퍼지제어기를 사용한다는 사실이 이치에 맞지 않는듯 생각되기 쉽지만 퍼지제어기는 그 구조상 룩업 테이블(look-up table) 방식으로 발생 가능한 여러 경우에 대해 적절한 제어동작을 저장할 수있고, 또 어떤 경우에 대한 제어 동작을 직관을 바탕으로 쉽게 제어기 내부에 구현할 수있다는 기존제어기가 줄 수없는 이점이있다. 즉 제어 결과가 만족스럽지 않을 경우 그 경우에 대한 대응을 쉽게 제어기 내부에 포함시킬 수가 있다는 점이다. 이러한 이유 때문에 직류 전동기, 온도제어등의 간단한 시스템에도 퍼지 이론이 적용되어 기존의 여러 방식에 비해 월등한 결과를 낸 경우가 많이 있다. 본 논문에서는 후자의 부류에 속하는 제어대상 시스템에 대한 퍼지제어기를 설계하는 문제를 다루려한다.

현재까지 해석적인 연구가 많이 이루어지고있는 퍼지제어기의 형태는 위치형 퍼지제어기(position type fuzzy controller : $(e, \dot{e}) \rightarrow u$)와 속도형 퍼지제어기(velocity type fuzzy controller : $(e, \dot{e}) \rightarrow \Delta u$)이다. 전자는 측정된 시스템 출력으로부터 제어 입력 자체를 구하는 형태이고 후자는 측정된 시스템 출력으로부터 제어입력의 증분을 구하는 형태인데 퍼지제어기가 사람이 일반적으로 사용하는 제어규칙을 근간으로한다는 측면에서보면 후자의 형태가 보통의 경우 더욱 실제적일 뿐아니라 부하나 시스템 특성등을 잘 모르는 상황에서 계단형 입력의 정상상태 오차를 없앨 수 있다는 측면에서 더 타당한 구조라고 볼 수있다.

그런데 속도형 퍼지제어기는 그 구조상의 특성으로 퍼지 PI 제어기라고도 불리우는데 정상 상태의 오차를 줄이는 능력은 탁월하지만 과도 응답을 개선하기가 2차 이상의 시스템인 경우에는 매우 어려울때가 많다. [13] 이러한 이유로 퍼지 제어기 설계를 재안한

많은 논문 [8-11] 들이 1차 시스템에 대해 시뮬레이션한 결과를 보고하고있다고 생각되며, 2차 이상의 경우에도 시스템의 시정수보다 천천히 변화하도록 증분의 최대값의 크기를 제한하고있다고 판단된다. [13] 이러한 현상의 이유는 보통의 PI 제어기에서 이득 결정이 쉽지않다는 사실에서 유추가능하며 이러한 문제의 해결을 위해서는 어떤 추가적인 정보의 사용이 불가피하다. 그 추가적인 정보로는, PID형을 만들기위해 가속도와 관계하는 \dot{e} 항을 생각할 수있는데 일반적인 사람의 경우 이 \dot{e} 의 값을 감지하여 제어에 이용하고 있다고 보기는 어렵다. 또 이 양은 직접 측정장비가 고가이거나 측정 자체가 불가능한경우가 있어 \dot{e} 이나 Δe 를 이용하여 간접적으로 구하는 경우가있는데 이때에도 해당제어시각 이전의 정보를 필요로한다는 단점이 있다.

본 논문에서는 힘과 가속도가 관계한다는 점에 착안하여 현 제어시각에 가해지고있는 입력을 보조정보로 이용하여 제어성능을 개선하는 제어기의 구조를 제안한다. 사람의 경우에도 자신이 현재 가하고있는 제어입력의 양은 정확히는 모르더라도 퍼지하게 알고 있다고하는 가정은 타당성이 있으므로 전문가의 전문지식을 수용하는데 무리가 없는 형태임을 강조한다. 이러한 내용을 구현하는 가장 기본적인 방법은 퍼지 제어규칙의 조건부에 오차(e), 오차의 변화율(\dot{e}), 제어 입력(u)가 모두 포함된 3차원 제어규칙을 만드는 방법이지만 이 세가지 양에 의해 만들어지는 경우의 수가 너무 많아 처리시간등의 관점에서 보면 바람직하지 못하다. 여기서 좀더 합리적인 방법을 찾아보기위해 기존의 퍼지제어기의 문제점에 대해 고찰해볼 필요가있다. 즉, 기존의 PI 퍼지제어기의 문제점이 누적된 적분기내의 제어량을 상쇄시키거나 필요한 제어량을 누적시켜나가는데 시간이 많이 걸리기 때문이라는 점에 착안하면 [14] 다음의 두가지 방향으로의 해결이 가능하다. 즉, 다음의 기본적인 PI 제어기의 계산에대해

$$u(k+1)=u(k) + \Delta u(k) \quad (1)$$

우변의 둘째항 ($\Delta u(k)$)에 어떤 조치를 가함으로써 퍼지 PI 제어기가 계산한 제어량의 증분(Δu)과 측정된 제어대상 시스템의 상태에 따라 구해진 가중치에 의해 실질적인 증분의 량을 달리하도록하는 방법이나 [15]와 첫째항($u(k)$)의 양을 불필요할때 어느정도 버려버리는, 즉 리셋(reset) 시키는 방법이 가능하다. 전자의 경우는 퍼지 PI 제어기의 역할을 중복적으로 관여하는 보조하는 소극적인 형태라고 볼 수있

고, 후자의 경우는 퍼지 PI 제어기가 할 수없는 역할을 함으로써 서로 보완하는 적극적인 형태라고 볼 수 있다. 본 논문에서는 후자의 방법에 의한 제어기의 구조를 제안한다.

본 논문에서의 핵심 부분을 퍼지하게 표현하면,

If response approaches given set value with large control input, reset the accumulated control input appropriately.

가 된다. 이러한 내용을 수행하게하기위해 기존의 퍼지 PI 제어기의 형태를 포함하고 오차가 "크다", "작다", 제어 입력이 "크다", "작다", 등으로 표현되는 (오차, 제어입력) 공간에서 정의된 퍼지규칙에 의해 현재 누적되어있는 제어입력에 대한 소거량을 "많이 하라", "작게하라" 등의 퍼지 변수로 연결하여 각 경우에 맞는 소거량에 의해 현재 누적되어있는 제어입력의 얼마만큼을 없애 버릴것인지를 정하여 경우에 따라 과도하게 누적된 제어입력이 소거되는 퍼지 제어를 고안하였다.

이어 2장에서는 새로운 형태의 제어를 소개하고 3장에서는 시뮬레이션 결과를 보이며, 4 장에서는 결론을 맺도록하겠다.

II. PI+U형 퍼지 제어기

앞서 1장에서 언급한 바와 같이 PI 형 퍼지제어기는 구조적으로 2차 이상의 시스템(물리적으로 관성을 갖는 시스템)에 대해 적용되기에는 오버슈트와 라이즈 타임의 개선에 문제점이 있는것으로 지적되므로 기존의 PI 형 퍼지제어기의 구조를 개선한 형태의 PI+U 형 퍼지제어기의 구조를 제안한다. PI 제어기의 단점을 보완한 PID 제어기의 도입 동기를 생각하면 퍼지 제어기의 경우에도 PID 형 퍼지제어기를 생각할 수 있다는 것이다. 정상상태오차를 없애기 위해서 Δu 를 계산하는 증분형 퍼지제어기의 형태를 유지한다면 오차의 2차 미분에 해당하는 \ddot{e} 과 관련된 양을 퍼지제어기의 규칙의 전제부에 포함시켜야 하는데 \ddot{e} 은 보통의 경우에 직접 측정이 쉽지 않아 \dot{e} 으로 부터 간접 계산해서 사용하는 방법이 일반적이다.^[12] 그러나 이렇게 하기 위해서는 해당 제어 시각 이전의 정보가 필요하게 되는 문제가 발생하며, 또한 퍼지제어기가 어떤 전문가의 전문지식을 기초로 이루어진다는 측면을 생각하면 과연 그 전문가가 \dot{e} 의 정보를 자신의 제어 행위에 이용하고있으며 그러한 형태의 제어규칙을 퍼지제어기에 이식할 수있는 형태로 제공받을 수

있는가하는 또 다른 문제점이 발생한다. 왜냐하면 \dot{e} 은 인간이 감지하기 쉽지않은 양이기 때문이다. 그러므로 합리적인 보조정보를 찾기 위해 일반적인 물리계에서 힘과 가속도가 관계있다는 사실에 주목하였다. 즉 PID 형 퍼지제어기를 구성하기위해 \dot{e} 을 직접 사용하기보다는 이와 관련이있고 전문가가 자신의 제어 행위에 이용하고있을 가능성이 크고 퍼지제어기로 이식이 용이한 양을 찾아보았다. 이러한 양으로써 제어량 자체를 생각했다. 자동차의 운전이라든가 레버를 당기거나 하는 일반적인 경우 전문가 자신이 해당 제어시각에 가지고있는 제어량은 "퍼지"하게 알고 있다고 볼 수있으므로 도입에 타당성을 부여할 수있겠다. 이러한 제어량을 이용하는 퍼지제어기의 형태는 전제부와 결론부에 제어량을 포함시키는 방법에 따라 여러가지가 있을 수있겠으나, 여기서는 앞서 설명한 PI 퍼지제어기의 단점이 발생하는 원인에 주목하여 다음과 같은 형태의 제어 규칙을 추가하는 방법을 제안한다. 즉 PI 퍼지제어기가 규칙 자체가 관성을 유발하여 누적된 제어양을 상쇄시키기 위한 시간이 오래 걸리고, 또한 필요한 양의 제어양에 도달하기위한 시간이 오래 걸린다는 점에 주목하여 누적된 제어량을 Δu 의 역할과 별도로 상황에 따라 적절히 제거한다는 직관적인 관찰에 기초한 형태의 제어를 제안한다. 이러한 형태의 퍼지제어기를 PI+U형 퍼지제어기라 부르기로하고 본 논문에서 제안한 형태는 그림 1과 같다.

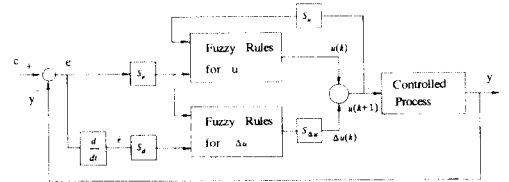


그림 1. PI+U형 퍼지제어기의 예
Fig. 1. PI+U type fuzzy controller.

A. 모델

먼저 제어기의 구조를 살펴보자.

그림 1의 S_e , S_d , $S_{\Delta u}$ 와 S_U 는 스케일링 계수 (scaling factor)로써 실질적으로 입력력 변수의 퍼지 분할에 관계하는 양으로 제어성과 밀접한 관계를 갖고있다. 이 계수를 구하는 것은 제어대상 시스템이 정해지면 그 시스템의 전문가의 지식에 의해 어느정도 범위의 값으로 정해진다고 볼 수있으며 추후 시스템의 운전을 통해 조정한다든지 또는 자체 조정 방법등에 의해 개선된 성능이 나오도록 하는 방법등

을 사용할 수있다. 이러한 스케일링 계수를 사용하면 임출력 변수의 퍼지 분할에 어떤 표준적인 형태를 사용할 수있다는 장점이 있다. 궁극적으로 PI 퍼지제어기의 적분역할이 문제점으로 제기되었으므로 상황에 따라 누적된 제어량을 적절히 상쇄시키는 구조의 제어기를 제안한다. 결국 다음의 제어규칙을 기본적인 PI 동작에 추가하는 것이 기본 착상이다.

If error is small but control input is too large, then take out appropriate amounts of accumulated control input according to situation which is fuzzily described by error and control input.

이렇게 언어적으로 묘사되는 추가적인 조치를 구현하는 방법에 있어 전체적으로 최종 제어입력은 그림 2와 같은 과정을 거쳐서 구해지는데, 크게 두개의 부분으로 구성되어있다.

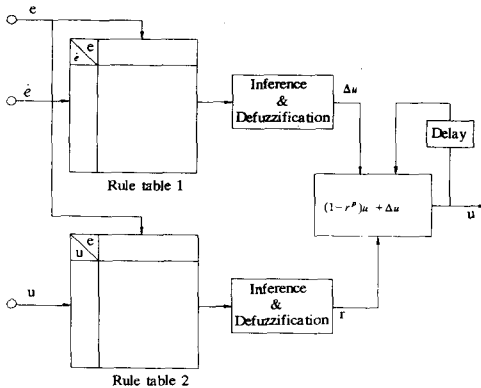


그림 2. PI+U형 퍼지제어기의 블럭도
Fig. 2. Block diagram of a PI+U type fuzzy controller.

하나는 오차와 오차의 변화율로부터 제어입력의 증분을 구하는 부분이고((e, e-dot) → Δu), 나머지 하나는 오차와 가해지고있는 제어입력의 크기로부터 리셋율을 구하는 부분이다((e, u) → r). 이렇게 구해진 두 양으로부터 다음식에 의해 최종제어입력을 계산한다.

$$u \leftarrow (1 - r^p) u + \Delta u \tag{2}$$

즉, 리셋율 r이 0과 1 사이의 값을 상황에 따라 적절히 가짐으로써, 과도하게 누적된 입력을 상쇄한다. 이 식의 의미를 살펴보면 r이 0이면 보통의 적분과정과

완전히 같아져 퍼지 PI 제어기의 형태가되며 r이 1이면 제어입력이 전혀 누적되지않아 그순간은 Δu가 바로 가해지는 입력 u 자체가 되므로 의미상 퍼지 PD 제어기와 같아진다. 여기서 상황에 따라 r의 값을 적절히 이용하므로써 두가지 제어기의 장점을 취해보자는 것이 본논문의 주된 의도이다. 식 (2)에는 p라는 유리수가있다. 이 값은 r의 영향을 반영하는 방법을 결정하게 되는데 이러한 영향을 그림 3에 나타내었다. p값이 크면 r=1근처를 제외한 나머지 r의 영역에서 리셋시키는 작용이 작으며 p값이 작으면 r=0 근처를 제외한 r의 영역에서 리셋시키는 작용이 많음을 알 수있다. 이러한 p값의 설정원칙은 리셋량의 비선형적 결정을 가능하게하며 다음에 설명할 규칙테이블과 상호관련되어 PI 퍼지제어기의 단점을 극복하는 역할을 한다. 각 부분의 계산과정을 자세히 설명하겠

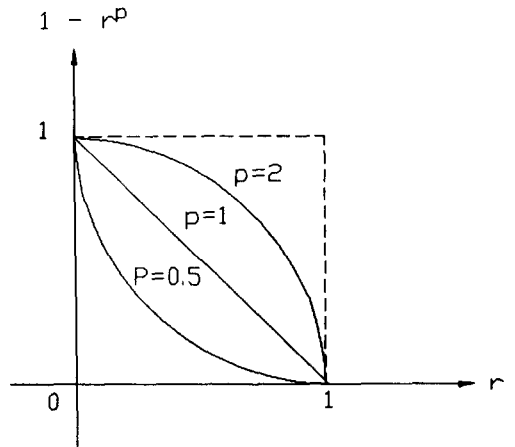


그림 3. p의 변화에 따른 1 - r^p의 그래프
Fig. 3. Effect of p on function 1 - r^p.

먼저 제어입력 증분 Δu는 다음의 과정에 의해 계산된다. 이 과정은 퍼지 추론부와 defuzzification 부분으로 구성되고 이를 수행하기위한 방법에는 여러가지가 제안되어있으나 여기서는 프로그래밍상의 구현이 간단한 [14]에서 사용된 바있는 단순화된 Mamdani의 방법을 채택한다. 즉, rule table 1의 각 규칙은 다음과 같은 형태로 기술되고

$$R_k : \text{If } e \text{ is } A_k \text{ and } e' \text{ is } B_k, \text{ then } \Delta u \text{ is } C_k \tag{3a}$$

이 각 규칙과 측정 상태의 적합도를 다음의 식에 의해 계산하며

$$w_k = \min (\mu_{A_k}(e), \mu_{B_k}(e)) \quad (3b)$$

이 적합도로부터 결론부의 퍼지집합을 다음식으로 구하고

$$\mu_{C_k}(u) = \min (w_k, \mu_{C_k}(u)) \quad (3c)$$

각 제어규칙으로부터 구해진 이 퍼지 집합을 defuzzification하여 각 규칙의 제어입력의 중분을 무게중심법(Center Of Gravity)에의해

$$\Delta u_k = \text{COG}(C'_k) \quad (3d)$$

로 구한다. 그리고 최종 제어 입력 증가분은

$$\Delta u = \frac{\sum_{k=1}^N w_k \Delta u_k}{\sum_{k=1}^N w_k} \quad (4)$$

에 의해 구해진다.

다음으로 리셋을 r을 구하는 방법에 대해 설명하겠다. r을 구하는데에도 궁극적으로 Δu를 구하는 (3) 식의 과정과 같은 방법을 사용한다. 즉, 다음의 규칙으로부터

$$R_k : \text{If } e \text{ is } A_k \text{ and } u \text{ is } U_k, \text{ then } r \text{ is } R_k \quad (5a)$$

로 규칙과 상황의 적합도를

$$w_k = \min(\mu_{A_k}(e), \mu_{U_k}(u)) \quad (5b)$$

로 구하고 결론부의 퍼지집합을 다음으로 구하며

$$\mu_{R_k}(u) = \min(w_k, \mu_{R_k}(u)) \quad (5c)$$

이 퍼지집합을 defuzzification 시켜

$$\Delta r_k = \text{COG}(R'_k) \quad (5d)$$

를 구한뒤 최종 리셋율은

$$r = \frac{\sum_{k=1}^N w_k \Delta r_k}{\sum_{k=1}^N w_k} \quad (5e)$$

로 구해진다.

이렇게 k번째 제어시각에 (3)의 과정으로 구해진 Δu(k)와 (5)의 과정을 통해 구해진 r에 의해 k+1번째 제어시각의 제어입력을 누적시키는데 있어서 다음의 식을 이용한다.

$$u(k+1) = (1 - (r(k))^p) u(k) + \Delta u(k) \quad (6)$$

B. 규칙 테이블의 해석

제어성능에 직접적인 영향을 미치는 가장 중요한 요소는 제어규칙이다. 일단 전제부와 결론부의 퍼지 집합이 정해지면 그 퍼지집합들 사이의 관계를 정해주는 제어규칙은 전제부의 변수가 2개인 경우는 테이블 형식의 표시가 가장 편리하다. 본 논문에서는 2개의 규칙 테이블이 사용되고있는데, 하나는 제어입력의 증분 Δu 를 구해내는 부분과 리셋을 r을 구하는 부분이다. 이중에서 Δu 를 구하는 테이블1 부분은 기존의 PI 퍼지제어기와 완전히 동일하므로 여기서는 그 해석을 생략한다. 그러나 r을 계산하는 (e, u) 공간에서 정의된 규칙 테이블 2는 그 의미를 해석하여 설계의 지침으로 삼고자한다.

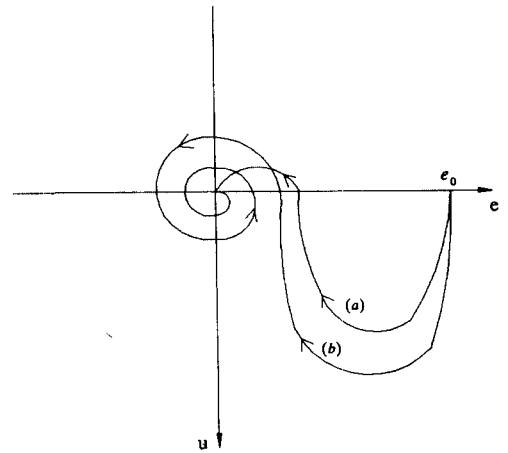


그림 4. (e, u) 공간에서의 궤적의 예

Fig. 4. Examples of trajectory on (e, u) space.

먼저 2차 시스템에 부하가 없는 경우 계단입력에 대한 응답을 (e, u) 공간에서의 궤적으로 표시하여 보자. 그림 4에 두가지 경우가 예로 나와있는데 (a)는 적절한 제어가 이루어져있는 경우이고 (b)의 경우는 오버슈트가 유발된 상태이다. (a)의 경우를 보면 초기에 오차가 양의 큰 값일때 양의 큰 제어입력이 가해지다가 오차가 작아지면서, 즉 응답이 설정치에 접근하면서 제어입력이 작아지다가 적절한 시각에 음의

값이 제어입력으로 가해져서 안정하게 설정치에 도착하고있다. 물론 부하가 존재하는 경우는 이 제적이 원점으로 수렴하지않고 u축 상의 어느점으로 수렴하게 된다. 그러나 (b)의 경우는 오차가 클때 양의 제어입력이 가해지다가 응답이 설정치에 접근하면서, 즉 오차가 작아지면서 누적된 양의 입력이 적절히 상쇄되지 못하여 설정치를 지나쳐버리고 이어 오차가 음의 값이 된후에도 과도한 음의 제어입력이 다시 상쇄되지 못하여 또 언더슈트를 유발하고있는 경우이다. 이러한 상황을 검토하여보면 (오차, 제어입력) 공간에 바람직하지 않은 영역이 어디인지를 판단할수있게 된다. 이를 언어적으로 표현하여보면 “오차는 양의 작은 값인데 제어입력은 양의 큰 값이 가해지고있는 경우” 또는 “오차는 음의 작은 값인데 제어입력은 지나치게 음의 큰값인 경우”등 여러가지로 기술될 수있다. 이러한 관찰을 바탕으로 그림 4의 시스템에 대한 비정상영역을 표시하면 그림 5처럼 나타내어진다. 다음에 기술할 퍼지 제어규칙과의 통일을 위해 제어 입력축의 양의 방향이 거꾸로 되어있음을 주의할 필요가있다. 그러므로 이러한 비정상적인 영역에 들어오게 되면 현재 가해지고있는 입력이 비정상적이라 볼수있으므로 그 제어입력을 없애 버린다는 것이 본 논문의 핵심 착안이다.

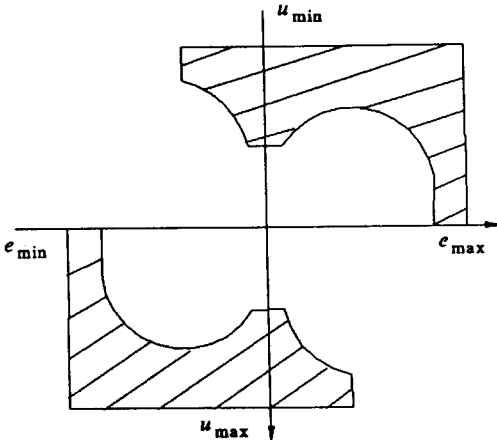


그림 5. (e, u) 공간에서 오버슈트를 유발하는 부분
Fig. 5. Parts causing overshoot on (e, u) space.

그런데 사실 어떠한 상황이 “정상이다.” “비정상이다”하고 판단하기는 그리 쉽지가 않다. 즉, 어느 정도는 정상이라고 볼 수있고 또한 어느정도는 비정상이라고 볼 수도있는 그러한 문제가 발생한다. 또한 비정상적이라 하여 현재 누적되어있는 입력을 전부

소거해 버리면 부하가 존재하는 시스템의 경우에는 순간적으로 부하를 지탱할 힘이 없어서 응답이 점프하는 현상등이 발생할 수도있다. 그러므로 “정상,” “비정상”을 퍼지하게 표현하는 것이 바람직하며, 소거량도 “완전소거”와 “무소거” 두개의 동작만 고려하는것보다는 소거량에 차등을 두는 퍼지 이론에 근거한 표시 방법이 합리적이다. 그러므로 본 논문에서는 “정상” “비정상”은

- NB : Negative Big
- NM : Negative Medium
- NS : Negative Small
- ZE : ZERo
- PS : Positive Small
- PM : Positive Medium
- PB : Positive Big

의 7등급으로 정의된 오차와 제어입력의 조합으로 표현하고 소거량도 다음의 퍼지변수에 의해 표시하였다.

- NR : No Reset
- VS : Very Small reset
- SR : Small Reset
- MR : Medium Reset
- BR : Big Reset
- VB Very Big reset
- CR : Complete Reset

그러므로 예를 들어 그림 5와 같은 형태의 비정상영역에 대한 상황과 리셋율을 연결시키는 테이블은 그림 6과 같은 형태가 되겠다. 물론 이러한 테이블은 시스템의 특징에 따라 달라지며 이러한 테이블을 만들기에 필요한 정보는 대상 시스템의 숙련 조작자로부터 전달받을 수있다는 모든 퍼지제어기의 설계과정과 마찬가지로의 가정을 한다. 그리고 이 리셋율을 계산하는 퍼지분할이 꼭 Δu 를 계산하는 퍼지분할과 같아야할 필요는 없겠으나 여기서는 공통의 퍼지 변수 값을 이용한다는 측면에서 같은 갯수의 퍼지분할을 택하였다.

결국 그림 6과 같은 퍼지규칙을 추가함으로써 (오차, 오차의 변화율, 제어입력) 공간에 직접 규칙을 설정한 경우의 조건부의 가짓수(이경우 $7 \times 7 \times 7 = 343$)에 비해 훨씬 작은 경우의 수($49 + 49 = 98$)를 갖게 되었지만 그림 6의 리셋율을 계산하는 규칙은 궁극적으로 오버슈트를 없애기 위한 장치이므로 이러한 제어기의 설계순서는 먼저 PI 부분으로 오버슈트를 고

려하지않고 만족할 만한 리즈타임을 갖도록 설계하고 이어 리셋율 계산 부분으로 유발된 오버슈트를 줄이는 방향으로 설계하는 방법이 바람직하다.^[13] 구체적인 적용예에 대해서는 다음장에서 설명하겠다.

μ^e	NB	NM	NS	ZE	PS	PM	PB
NB	NR	CR	CR	CR	CR	CR	CR
NM	NR	BR	VB	CR	VB	BR	CR
NS	NR	NR	NR	SR	SR	NR	CR
ZE	NR	NR	NR	NR	NR	NR	NR
PS	CR	NR	SR	SR	NR	NR	NR
PM	CR	BR	VB	CR	VB	BR	NR
PB	CR	CR	CR	CR	CR	CR	NR

그림 6. 리셋율을 계산하는 퍼지규칙의 예
Fig. 6. A fuzzy rule calculating reset rate.

Ⅲ. 모의 실험

먼저 다음의 플랜트에 대해 퍼지 PI 제어를 적용하여보자.

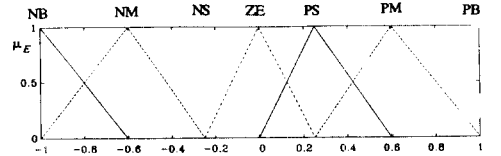
$$G(s) = \frac{1}{s(s+1)} \quad (7)$$

그림 7과 같은 비선형 퍼지 분할과 그림 8의 규칙이 Δu 를 구하는데 이용된 바, 응답은 그림 9에, (오차, 제어입력) 공간에서의 궤적은 그림 10처럼 나타났다. 그림 9와 10에서 각 경우의 설계계수는

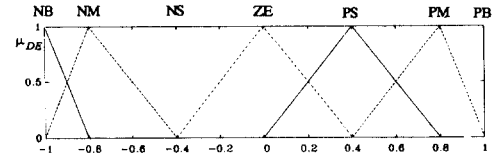
- (a) $S_e = 1.0$ $S_i = 1.5$ $S_u = 0.5$
- (b) $S_e = 1.0$ $S_i = 2.0$ $S_u = 0.5$
- (c) $S_e = 1.0$ $S_i = 3.0$ $S_u = 0.5$
- (d) $S_e = 0.8$ $S_i = 2.5$ $S_u = 0.5$

로 하였는데 이를 보면 오버슈트를 작게하기위해서 계수를 조절하면 리즈 타임이 현저하게 길어지고 리즈 타임을 개선하려면 오버슈트가 커짐을 알 수있다. 이러한 현상을 개선하기 위해서 가장 쉬운 노력은 일단 가속구간에서 누적되는 제어입력의 양을 적

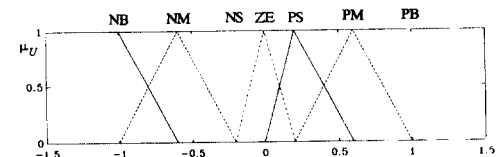
절히 조절해 주어야 하는데 퍼지 PI 제어기의 구조에서는 이러한 조절이 결코 쉽지 않다.



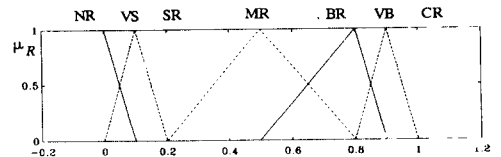
(a)



(b)



(c)



(d)

그림 7. 사용된 퍼지 변수들
Fig. 7. Fuzzy sets used in simulation.

이러한 결과는 PI형 제어기의 일반적인 현상으로 입출력 변수의 퍼지 분할이나 제어 규칙의 조정으로는 개선에 한계가있다고 볼 수있어 보조 정보를 이용하는 새로운 구조의 제어기 형태가 필요하다는 결론에 이르게된다. 이와 같은 플랜트에 본 논문에서 제안한 제어기를 적용하여 보았다. 이때의 계수는 $(S_e, S_i, S_{\Delta u}, S_u) = (1.0, 0.52, 300, 1/18)$ 로 리셋율을 계산하기 위한 규칙 테이블은 그림 11로 (4)식의 p는 1로 했을때 응답은 그림 12에 (오차, 제어입력) 공간에서의 궤적은 그림 13에 나타나있다. 여기서 비교에 사용된 두 개의 제어기에서 계수값이 달라진 것은 여기서 제안된 방법은 리셋율이라는 새로운 요소가 추

가되었기 때문에 그 역할이 달라지기 때문에 퍼지 PI 제어기에서의 계수들을 그대로 사용하게 되면 비교의 의미가 없게된다.

\dot{e} \ e		NB	NM	NS	ZE	PS	PM	PB
NB	NB	NB	NB	NM	NS	NS	NS	ZE
NM	NB	NM	NM	NM	NS	ZE	PS	PS
NS	NB	NM	NS	NS	ZE	PS	PM	PM
ZE	NB	NM	NS	ZE	PS	PM	PB	PB
PS	NM	NS	ZE	PS	PS	PM	PB	PB
PM	NS	ZE	PS	PM	PM	PM	PB	PB
PB	ZE	PS	PS	PM	PB	PB	PB	PB

그림 8. (7)의 시스템에 대한 Δu 를 계산하기위한 규칙

Fig. 8. Rule table for calculating of Δu for the system of (7).

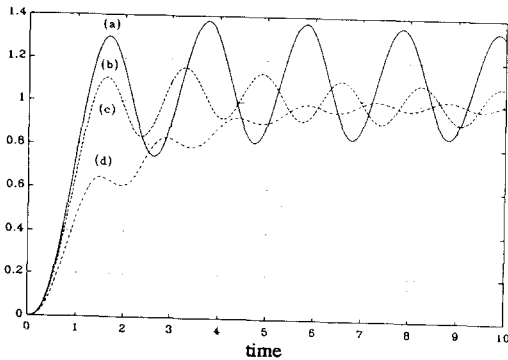


그림 9. 퍼지 PI 제어기에 의한 (7)의 응답

Fig. 9. Response of (7) with fuzzy PI controller.

그러므로 여기서는 시행착오 후에 찾아진 각 제어기의 출력중 상당히 양호한 결과를 주는 경우를 비교하는 방법을 사용하였다. 그림 12에서의 "r"의 역할을 보면, 오차가 클때는 0의 값을 가져 제어입력이 충분히 누적되는데 아무 영향을 미치지 않다가 오차가 작아지면서 누적된 제어입력을 감소시켜 (0.1 < t < 0.2) 응답의 기울기를 감소시키고 급기야 제어입력을 음의 값으로 만든다. 이때 제어입력이 과도한 음의 값을

가지면 설정치에 도달하지 못하는 경우가 발생할 수 있으므로 과도한 음의 값을 갖지 않도록 적당량을 소거하여 (0.3 < t < 1.3) 신호가 설정치에 안정되게 도달하도록 하고있음을 알 수있다. 이상의 결과에서 퍼지집합 또는 계수들의 설정에 따라 다른 결과를 내게되지만 여기서는 최적의 계수를 찾는 내용이 주된 핵심이 아니므로 몇 번의 시행 착오에 의해 구해내는 방법을 사용하였음을 밝힌다.

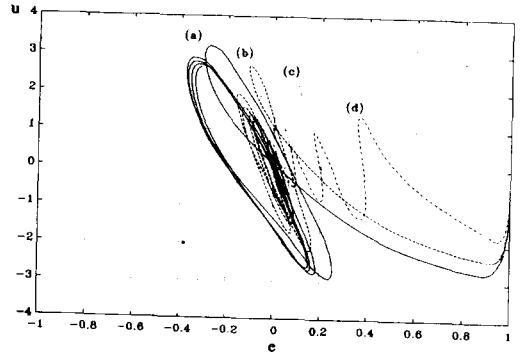


그림 10. (오차, 제어입력) 공간에서 그림 9의 궤적

Fig. 10. Trajectories of the responses of Fig. 9 on (e, u) space.

u \ e		NB	NM	NS	ZE	PS	PM	PB
NB	NR	CR	CR	CR	CR	CR	CR	CR
NM	NR	VB	CR	CR	CR	BR	CR	CR
NS	NR	NR	SR	BR	SR	NR	CR	CR
ZE	NR	NR	NR	NR	NR	NR	NR	NR
PS	CR	NR	SR	BR	SR	NR	NR	NR
PM	CR	BR	CR	CR	CR	VB	NR	NR
PB	CR	CR	CR	CR	CR	CR	CR	NR

그림 11. (7)의 시스템에 대한 reset 율을 계산하기위한 규칙

Fig. 11. Fuzzy rules calculating reset rate for (7).

또한 보다 복잡한 다음의 시스템에 2장의 퍼지 PI 제어기를 적용시킨 결과 그림 14의 응답과 그림 15의 결과를 얻었다.

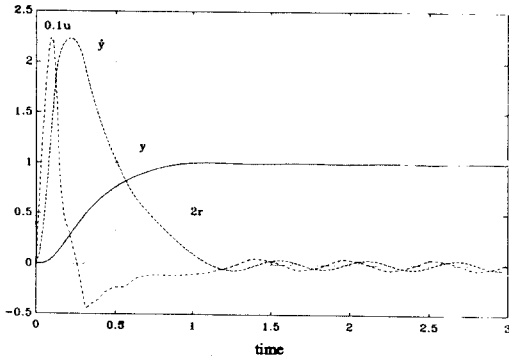


그림 12. 본 논문에서 제안된 방법이 (7)에 적용된 결과의 응답
 Fig. 12. Response of (7) under the proposed controller.

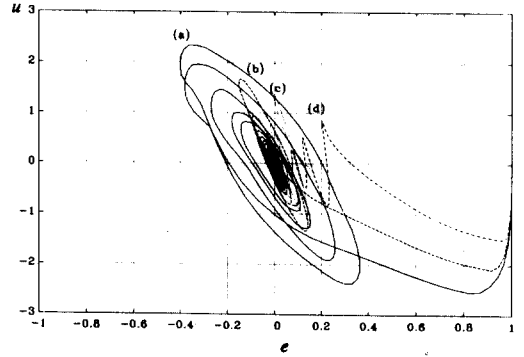


그림 15. (오차, 제어입력) 공간에서 그림 14의 궤적
 Fig. 15. Trajectories of the responses of Fig. 14 on (e, u) space.

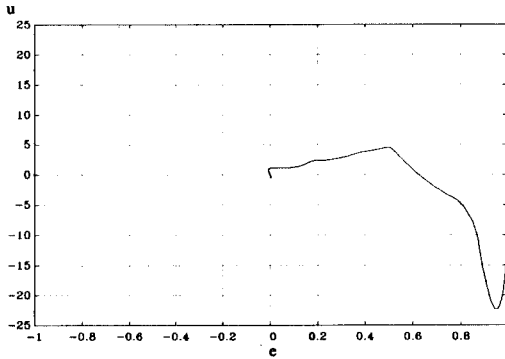


그림 13. (오차, 제어입력) 공간에서의 그림 12의 궤적
 Fig. 13. Trajectory of the responses of Fig. 12 on (e, u) space.

$e \backslash u$	NB	NM	NS	ZE	PS	PM	PB
NB	NR	VB	CR	CR	CR	CR	CR
NM	NR	NR	BR	CR	CR	CR	CR
NS	NR	NR	SR	SR	NR	CR	CR
ZE	NR	NR	NR	NR	NR	NR	NR
PS	CR	CR	NR	SR	SR	NR	NR
PM	CR	CR	CR	CR	BR	VS	NR
PB	CR	CR	CR	CR	CR	VB	NR

그림 16. (8)의 시스템에 대한 reset 율을 계산하기 위한 규칙
 Fig. 16. Fuzzy rules calculating reset rate for (8).

$$G_2(s) = \frac{2000}{s(s+2)(s+100)} \quad (8)$$

이때의 계수들은

- (a) $(S_e, S_d, S_u) = (1.0, 1.0, 10)$
- (b) $(S_e, S_d, S_u) = (1.0, 1.5, 10)$
- (c) $(S_e, S_d, S_u) = (1.0, 2.0, 10)$
- (d) $(S_e, S_d, S_u) = (1.0, 3.0, 10)$

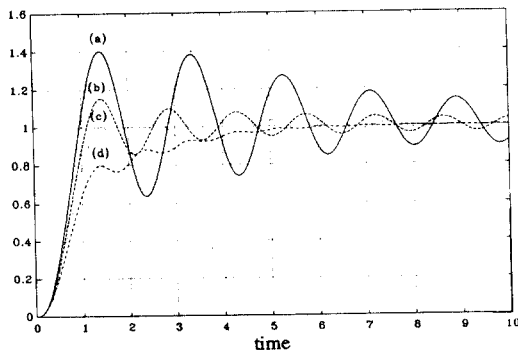


그림 14. 퍼지 PI 제어기에 의한 (8)의 응답
 Fig. 14. Response of (8) with fuzzy PI controller.

이고 (8)의 시스템에 대해 여기서 제안된 방법을 적용하여보았다.

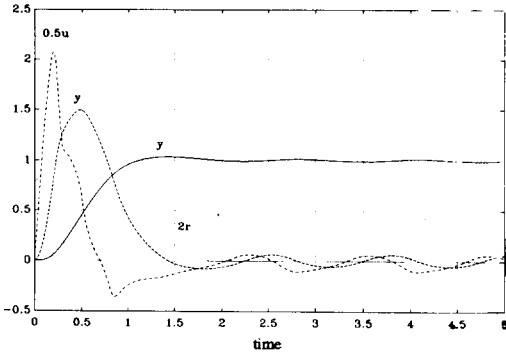


그림 17. 본 논문에서 제안된 방법이 (8)에 적용된 결과의 응답

Fig. 17. Response of (8) under the proposed controller.

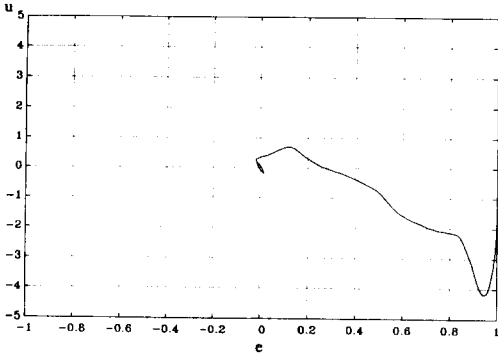


그림 18. (오차, 제어입력) 공간에서의 그림 17의 궤적

Fig. 18. Trajectory of the response of Fig. 17 on (e, u) space.

계수들은 $(S_e, S_d, S_{\Delta u}, S_v) = (1.0, 0.6, 25, 1/2.5)$ 이며 (6)식의 p 는 역시 1로 r 의 결정규칙은 그림 16로 하였을때 그림 17, 18의 응답을 얻어 그특성이 많이 개선되었음을 알 수있다. r 의 역할에 대해서는 역시 그림 12의 경우와 같은 해석을 할 수있음을 알 수있다.

IV. 결론

PI 형 퍼지제어기는 구조적으로 2차 이상의 시스템에 대해 적용되기에 문제점이 지적되므로 기존의 PI 형 퍼지제어기의 구조를 개선한 형태의 PI+U 형 퍼지제어기의 구조를 제안했다. PI 제어기의 단점을 보완한 PID 형 퍼지 제어기의 도입을 고려하여 오차의

2차 미분에 해당하는 \ddot{e} 과 관련된 양을 퍼지제어기의 규칙의 전제부에 포함시킨다는 동기로 보통의 경우에 직접 측정이 쉽지 않은 \ddot{e} 대신 해당 제어시각에 가하고있는 제어량은 “퍼지”하게 표현하여 퍼지제어규칙의 전제부에 포함시킨 PI+U 퍼지 제어기를 제안하였다. 기존의 퍼지제어규칙과 제어시각의 시스템 상태와 가해지고있는 제어량으로부터 퍼지 리셋율을 구하는 퍼지관계를 추가로 구성하여 궁극적으로

*If current u is large and error is small,
then reset the accumulated control input according to situation.*

제안된 퍼지제어기는 2차 이상의 시스템에 시뮬레이션 적용해 본 결과 기존의 PI 퍼지제어기보다 우수한 특성을 나타내었다. 그러나 제어량을 제어기의 입력정보로 이용하는 방법에는 여러가지 형태의 구조가 가능하므로 추후 보다 효과적인 구조의 개발과 추가된 제어량의 퍼지 표현, 퍼지 분할등에 대한 다른 각도의 연구도 병행되어 필요하다고 할 수있겠으며, 제안된 퍼지제어기 내의 설계요소, 스케일링 계수, 리셋율 계산 규칙등의 상호 작용을 면밀히 분석하여 그 실정을 보다 조직적으로 수행할 수있는 지침의 확립등도 추후 연구되어야 할 과제로 판단된다.

제안된 퍼지 제어기는 기존의 퍼지 PI 제어기보다 우수하다는 결론만이 지어진 상태로 어떻게 최상의 제어기를 구성하느냐하는 제어기 tuning에 관한 문제가 아직 남아있다고 볼 수있으므로 최적의 퍼지제어의 설계를 위한 이득이나 계수등의 조정 알고리즘에 대해서도 제안된 제어기의 성능을 개선하기위해 지속적인 연구가 필요하다고하겠다.

參考文獻

[1] L. A. Zadeh, “Fuzzy sets.”, *Imformt. Control*, vol. 8, pp. 338-353, 1965.
 [2] S. S. L. Chang and L. A. Zadeh, “On Fuzzy Mapping and Control,” *IEEE Trans. Syst. Man Cybern.*, vol. 15, no. 1, pp. 175-189, 1985.
 [3] E. H. Mamdani, “Application of Fuzzy Algorithms for the Control of Dynamic Plants,” *Proc. IEE*, vol. 12, no. 12, 1974
 [4] S. Assilian and E. H. Mamdani, “An Experiment in Liguistic Synthesis with

- a Fuzzy Logic Controller," *Int. J. Man-Machin Stud.*, vol. 7, 1974.
- [5] C. C. Lee, "Fuzzy Logic in Control Systems : Fuzzy Logic Controller - Part I, II," *IEEE Trans. Syst. Man Cybern.*, vol. 20, no. 2, pp. 404-435, 1990.
- [6] R. M. Tong, "A Control Engineering Review of Fuzzy Systems," *Automatica*, vol. 13, pp. 559-569, 1977.
- [7] R. M. Tong, "Some Properties of Fuzzy Feedback Systems," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-10, no. 6, pp. 327-330, 1980.
- [8] K. Tanaka and M. Sano, "A New Tuning Method of Fuzzy Controllers," *Proc. IFSA '91*, pp. 207-210, 1991.
- [9] M. Maeda, T. Sato, and S. Murakami, "Design of the Self-Tuning Fuzzy Controller," *Proc. Int'l Conf. Fuzzy Logic and Neural Networks*, pp. 393-396, 1990.
- [10] S. Yamada, H. Fujikawa, A. Fuji, and J. Yamakawa, "A Tuning Method for a Fuzzy Controller," *Proc. Int'l Conf. Fuzzy Logic and Neural Networks*, pp. 397-400, 1990.
- [11] M. Yoshida, Y. Tshutsumi, and T. Ishida, "Gain Tuning Method for Design of Fuzzy Control Systems," *Proc. Int'l Conf. Fuzzy Logic and Neural Networks*, pp. 405-408, 1990.
- [12] S. Hayashi, "Auto-Tuning Fuzzy PI Controller," *Proc. IFSA '91*, pp. 41-44, 1991.
- [13] B. Kuo, *Automatic Control Systems, 6th ed.*, pp. 463-479, Prentice-Hall International Ed., 1991.
- [14] 이 지 흥, "퍼지 리셋기능을 갖는 PI 형 퍼지 제어기," 대한 전자공학회 논문지에 제출, 1992.
- [15] 채 석, 이 지 흥, "제어량의 증분에 가중치를 가한 퍼지 PI 제어기," 금오공대 전자제어공학과, technical report 92-08-20.

 著 者 紹 介

李 志 弘 (正會員) 第 29卷 B編 第 11號 參照

현재 금오공과대학 전자제어공학과 교수.