

# 비선형 함수의 분해를 이용한 퍼지시스템의 재구성과 퍼지규칙수 줄임 알고리즘

## Fuzzy Rule Reduction Algorithms and the Reconstruction of Fuzzy System using Decomposition of Nonlinear Functions

유 병 국

Byung Kook Yoo

### 요 약

일반적으로 퍼지시스템은 compact한 공간에 대한 어떠한 비선형 함수도 일정오차 이내에서 근사할 수 있다. 그러나 퍼지시스템의 응용은 퍼지규칙의 수가 많아지는 경우, 특히 고차의 비선형 시스템에 대하여는 사용되기 어렵다는 단점을 가지고 있다. 본 논문에서는 근사하고자 하는 비선형 함수의 분해를 이용한, 병렬형과 종속형의 두 가지 형태의 퍼지시스템 재구성 방식을 제안한다. 이 두 가지 형태의 재구성을 적절히 이용하여 퍼지규칙의 수를 기하급수적으로 줄일 수 있다. 제안된 알고리즘은 적응구조를 가진 퍼지시스템에 대하여 응용 가능하며 두 가지 적응 퍼지 슬라이딩모드제어 예를 통하여 그 타당성을 보인다.

### ABSTRACT

Fuzzy system is capable of uniformly approximating any nonlinear function over compact input space. The applications of fuzzy system, however, have been primarily limited by the need for large number of fuzzy rules, in particular, for the high-order nonlinear system. In this paper, we propose the reconstruction methods of fuzzy systems, parallel type and cascade, based on the decomposition of some classes of high-order nonlinear functions. Using the both types appropriately, we can reduce the number of fuzzy rules geometrically. It can be applied to the fuzzy system that has an online adaptive structure. Two examples of adaptive fuzzy sliding mode control are shown in the computer simulations to verify the validity of the proposed algorithm.

Keywords : Fuzzy system, Fuzzy rule reduction, Decomposition, Adaptive fuzzy sliding mode control.

### I. 서 론

퍼지시스템이 compact 입력공간에 대한 어떠한 비선형 함수도 근사할 수 있다는 Universal Approximation Theorem[3]을 이용한 많은 연구들이 있었다[3-13]. 이러한 연구는 복잡하고 비선형인 시스템에 대하여 제어를 설계하기 어렵다는 기존 제어이론의 단점과 퍼지제어이론이 가진 수학적 해석의 단점을 보완하기 위한 연구였다. Wang[3]은 오차방정식을 이용한 적응퍼지제어방식을 제안하였으며 [6-11]에서는 여러 가지 기존의 제어이론과의 접목을 시도하였다. 그러나 이러한 방식들은 수학적으로 그 안정도를 증명하는데 완벽하지 못하였으며 특히 Wang은 이러한 수학적 해석상의 단점을 극복하기 위해 Supervisory 방식을 도입하기도 하였다. Yoo는 [12]에서 퍼지 슬라이딩모드제어방식을 제안하였으며 여기서 기존의 제어방식인 슬라이딩모드 제어방식과 퍼지시스템의 이용에 대한 수학적 안정도 해석의 문제를 어느 정도 해

결할 수 있었다. 또한 [13]에서 로봇 매니퓰레이터의 불확실성을 보상하기 위한 근사기로서 퍼지시스템이 이용되었으나 퍼지규칙의 수가 많아 실제로 제어를 구현하는 관점에서는 실현하기 어렵다는 단점을 가지고 있다. 이것은 그 시스템의 차수가 증가하면 퍼지시스템의 규칙베이스를 구성하는 퍼지규칙의 수가 기하급수적으로 증가하기 때문이다.

일반적으로 퍼지시스템의 응용은 그 퍼지시스템 내의 지식기반을 이루는 퍼지규칙의 수에 의해 제한된다. 즉, 이론적으로는 퍼지시스템의 이용이 가능하더라도 퍼지규칙의 수가 기하급수적으로 증가하는 문제로 인하여 실제 시스템에의 적용이 어려워진다는 것이다. 이러한 문제를 해결하고자 수치해석적 방법인 보간법을 이용하는 방식 [14]가 있었으며 여기에 단일값 분해법(SVD:singular value decomposition)을 이용한 방식[15]과 퍼지규칙의 잉여(redundancy) 부분을 제거하려는 방식[16]이 있었다. 또한 뉴로-퍼지에서 방식으로서 iteration algorithm을 이용한 방식[17]이 제안되기도 하였다. 그러나 이들 방식은

미리 설계해놓은 퍼지규칙의 전체부분을 분석하여 규칙 베이스를 재 설계하는 방식으로서 이는 필요로 하는 규칙을 미리 찾아야 한다는 단점을 가지고 있다. 따라서 퍼지시스템의 파라미터를 실시간으로 변경하는 적응퍼지방식에는 사용될 수 없는 알고리즘이다. 더구나 이러한 알고리즘은 복잡하고 고차의 시스템에 적용하기 위해서는 많은 분석시간을 요하는 것이다.

본 논문에서는 이러한 어려움을 해결하고자 근사하고자 하는 비선형 방정식을 적절한 형태로 분해하여, 병렬형과 종속형의 두 가지 형태의 퍼지시스템 재구성방식을 제안한다. 제안된 알고리즘을 통하여 퍼지규칙의 수를 줄일 수 있음을 간단히 증명하며 두 예제 시스템에 대하여 [12]에서의 적응 퍼지 슬라이딩모드제어기를 설계하고 시뮬레이션을 통하여 제안된 알고리즘의 타당성을 보인다. 제안된 방식은 퍼지시스템의 파라미터를 실시간으로 적용시키는 응용에서 사용될 수 있으며 또한 미리 알고있는 지식을 이용할 수 있다는 퍼지의 근본 이용목적에 만족하는 방식이다.

본 논문의 구성은 II장에서는 기본적인 퍼지시스템에 대하여 살펴보고 III장에서는 두 가지의 퍼지시스템 구성 방식, 병렬형과 종속형을 제안하며 또한 퍼지규칙 수가 지수적으로 감소될 수 있음을 보인다. IV장에서는 두 가지의 예제를 들어 퍼지시스템의 재구성을 보이며 [12]의 적응 퍼지 슬라이딩모드제어기를 설계하고 이에 대한 컴퓨터시뮬레이션 결과를 살펴본다. 마지막으로 V장에서는 결론과 향후 연구에 대하여 논한다.

## II. 퍼지 시스템

퍼지 시스템은 지식베이스, 퍼지화기, 퍼지추론엔진, 비퍼지화기의 4가지 부분으로 구분된다. 여기서 지식베이스부분은 퍼지 if-then 규칙으로 이루어지며 퍼지 추론엔진에서 이 퍼지규칙을 기반으로 추론을 행하게 된다. 본 논문에서는 MISO 퍼지규칙을 사용하게 되는데 이러한 퍼지규칙은 다음과 같은 형태를 가진다.

$$R = \bigcup_{l=1}^M R_l \quad (1)$$

여기서  $\bigcup$ 은 퍼지규칙들의 union을 나타내는 것이며 각각의  $R_l$ 는 다음과 같은 MISO의 형태를 가진다.

$$R_l: \text{If } x_1 \text{ is } A_1^l, x_2 \text{ is } A_2^l, \dots, x_n \text{ is } A_n^l, \text{ then } y \text{ is } B^l \quad (2)$$

여기서  $\mathbf{x} = (x_1, x_2, \dots, x_n)^T \in U \subset R^n$ ,  $U = U_1 \times U_2 \times \dots \times U_n$ 이며  $y \in V \subset R$ 이며 각각 퍼지시스템의 입력과 출력변수를 나타낸다.  $A_i^l$ 과  $B^l$ 은 각각  $U_i$ 와  $V$  공간상에 정의된 퍼지집합의 언어변수이며 이 집합의 소속함수는 각각  $\mu_{A_i^l}(x_i)$ 과  $\mu_{B^l}(y)$ 로 나타내지며 여기서

$l = 1, 2, \dots, M$ 이다.

퍼지화 부분에서는  $\mathbf{x}$ 의 상태 값을 퍼지시스템의 입력 공간  $U \in R^n$ 상의 하나의 퍼지집합  $A^l$ 로의 맵핑을 수행한다. 일반적으로 이러한 맵핑에는 두 가지의 방법이 있으나 본 논문에서는 단일 값 퍼지화방법을 사용한다.

퍼지 추론엔진에서는  $U$ 내의 퍼지집합을 출력공간  $V$  내의 퍼지집합으로의 맵핑을 수행한다. 이 때 추론엔진은 퍼지 규칙베이스내의 if-then 퍼지규칙을 기반으로 조합 연산을 이용하여 추론을 행한다. 본 논문에서는 각 퍼지 규칙에 대하여 곱추론(product inference)을 사용하며  $M$ 개의 결과 퍼지집합에 대하여 max연산을 이용한다.

비퍼지화에서는 퍼지시스템의 출력공간  $V$ 내의 퍼지 집합을 실수공간내의 하나의 보통값(crisp value)으로의 변환을 수행한다. 일반적으로 사용하는 방법으로는 세가지 방법이 있으나 본 논문에서는 가장 일반적인 무게중심법을 사용한다.

**정의 1 :** 무게중심법 비퍼지화(center-average defuzzifier)방법, 곱 추론(product inference), 단일값 퍼지화(singleton fuzzifier)방법을 사용하는 MISO-퍼지시스템의 출력은 다음과 같다.

$$y(x) = \frac{\sum_{l=1}^M (\prod_{i=1}^n \mu_{A_i^l}(x_i)) \bar{y}^l}{\sum_{l=1}^M (\prod_{i=1}^n \mu_{A_i^l}(x_i))} \quad (3)$$

여기서  $\bar{y}^l$ 은  $V$ 내의  $\mu_{B^l}(y)$ 의 소속정도가 최고치 또는 정규화된 퍼지집합인 경우는 1에 해당되는  $y$ 상의 보통값이다. ■

기본적인 퍼지시스템은 고정되어있으나 이 퍼지시스템을 어떠한 비선형 함수를 실시간으로 근사하는 근사기로 사용하기 위해서는 적응력이 있는 구조로 만들어야 한다. 따라서 퍼지시스템을 적응법칙을 이용하여 적응시키거나 학습법칙을 이용하여 학습능력을 가지도록 하기 위해  $A_i^l$ 은 고정으로 하고  $\bar{y}^l$ 를 적응 가능한 파라미터로 한다. 이러한 구조는 Universal Approximation Theorem[3]에서 그 근사능력이 증명되어있다. 이 구조를 나타내기 위하여 (3)을 퍼지 베이스스함수벡터와 파라미터벡터로 구분하여 다음과 같이 나타낼 수 있다.

$$y(x) = \theta^T \xi(x) \quad (4)$$

$$\xi(x) = \frac{\prod_{i=1}^n \mu_{A_i^l}(x_i)}{\sum_{l=1}^M (\prod_{i=1}^n \mu_{A_i^l}(x_i))} \quad (5)$$

여기서  $\xi(x) = (\xi_1(x), \dots, \xi_M(x))^T \in R^M$ 는 퍼지베이스 함수벡터 또는 전제함수벡터라 하며  $\theta = (\bar{y}^1, \dots,$

$\bar{y}^M)^T \in R^M$  을 파라미터벡터라 한다.

다음 정리는 compact한 입력공간에 대한 어떠한 비선형 함수도 퍼지시스템이 허용 오차 내로 근사할 수 있다는 정리이다.

**정리 1 :** (Universal Approximation Theorem)

Compact 공간상에 연속 함수  $f(x)$ 가 주어졌을 때 임의의 상수  $\epsilon > 0$ 에 대하여 다음 식을 만족시키는 식 (4)와 같은 퍼지시스템  $\hat{f}(x|\theta)$ 가 존재한다.

$$\sup_{x \in U} |\hat{f}(x|\theta) - f(x)| < \epsilon. \quad (6)$$

**증명 :** 참고문헌 [3]의 부록 참조. Q.E.D.

퍼지 시스템의 어떠한 비선형 함수도 근사할 수 있다는 성질을 이용하여 대부분의 퍼지시스템의 응용에서 비선형 함수를 근사하는데 사용한다. 즉, 퍼지시스템은 근사기로 사용되는 것이다. 그러나 일반적으로 비선형 함수를 근사하기 위해서는 입력변수  $n$  개가 모두 퍼지시스템의 입력변수가 되어야 하며 따라서 각 입력변수에 일괄적으로  $k$ 개의 퍼지라벨을 설정한다면 전체적으로 필요한 퍼지규칙의 수는  $k^n$ 개가 된다. 따라서 고차의 SISO-비선형 시스템의 경우 퍼지시스템의 퍼지규칙의 수는 그 차수가 하나 증가할 때마다 기하급수적으로 증가하게 된다. (좀더 일반적으로 말하면  $n$ 개의 각 입력변수에 대하여 각각  $k_1, k_2, \dots, k_n$ 개의 퍼지라벨을 설정한다면 전체적으로 필요한 퍼지규칙의 수는  $k_1 \times k_2 \times \dots \times k_n$ 개의 퍼지규칙이 필요하다.) 결국  $M = k^n$ 이 되며  $\xi$ 와  $\theta$ 는 각각  $k^n \times 1$  벡터가 되고 퍼지시스템의 출력계산에 필요한 계산양도 기하급수적으로 증가하게 된다.

이러한 문제를 해결하기 위하여 다음 장에서는 제어하고자 하는 플랜트의 모델링 된 시스템 함수에 대하여 미리 알고있는 내용, 즉 함수구조를 이용하여 함수를 분해하고 이에 맞도록 퍼지시스템의 형태를 다시 재구성하여 각 퍼지시스템에 대한 입력변수의 수를 줄임으로서 전체적으로 필요한 퍼지규칙의 수를 줄이는 두 가지 형태의 퍼지시스템 구조를 제안한다.

**III. 퍼지시스템의 재구성**

다음과 같은  $n$ 차 비선형 시스템을 고려하자.

$$\begin{aligned} \dot{x}^{(n)} &= f(x, \dot{x}, \dots, x^{(n-1)}) + b(x, \dot{x}, \dots, x^{(n-1)})u(t) \\ y &= x \end{aligned} \quad (7)$$

여기서  $u \in R, y \in R, x = (x_1, x_2, \dots, x_n) = (x, \dot{x}, \dots, x^{(n-1)})^T \in R^n$ 이며 모든 상태가 측정가능하다고 가정한다.

$f$ 는 그 함수의 형태는 알고있으나 파라미터의 값은 잘 모른다고 가정한다. 이것은 대부분의 동력학 시스템에 대한 수학적 모델은 Lagrange 방정식을 통하여 유도할 수 있으나 그 동력학에 포함된 물리적 파라미터의 값은 알수 없는 경우가 많기 때문이다. 예를 들면 로봇 매니플레이터의 경우 동조해야하는 파라미터의 개수가 링크 하나당 10개에 달한다. 또한 본 논문에서는 재구성된 퍼지시스템을 이용한 근사기를 설계하는 것이 목적이므로 입력이득  $b$ 는 알고있는 함수라 가정한다. (7)이 제어가능하기 위해서 가제어 영역내의 모든  $x$ 에 대하여  $b > 0$  이라 가정한다.

[12]에서와 같이 근사하고자 하는 비선형 함수  $f(x, t)$ 를 단지 상태벡터  $x$ 의 함수로 보아 모든 상태변수들을 퍼지시스템의 입력변수로 한다면, 그리고 각 입력변수에 대하여  $k$ 개의 퍼지라벨을 정의한다면 비선형 함수를 근사하기 위한 퍼지시스템은  $k^n$ 개의 퍼지 규칙이 필요하게 된다. 그러나 주어진 비선형 함수  $f(x, t)$ 의 모델은 알고 있다고 가정하였으므로 그 함수의 형태를 적절히 이용하여 분해하여 퍼지시스템의 구조를 변경한다면 그 퍼지규칙의 수를 줄일 수 있다. 먼저 이러한 분해의 형태를 다음과 같이 병렬형과 종속형의 두 가지로 구분하여 재구성된 퍼지시스템의 퍼지규칙의 수가 줄어들 수 있음을 살펴본다.

**1. 병렬형 퍼지시스템**

이러한 형태의 구조를 갖도록 하기 위해서는 다음과 같이 근사하고자 하는 비선형 함수가 두개 이상의 비선형 함수로 나뉘어질 수 있어야 한다.

$$f(x) = f^1(x^1) + f^2(x^2) + \dots + f^z(x^z) \quad (8)$$

여기서  $x^1 \in R^{p_1}, x^2 \in R^{p_2}, \dots, x^z \in R^{p_z}$ 이다. 이러한 비선형 함수에 대한 근사기로서 퍼지시스템은 다음과 같이 재구성될 수 있으며 이를 그림으로 나타내면 그림 1과 같다.

$$\hat{f}(x|\theta) = \hat{f}^1(x^1|\theta^1) + \hat{f}^2(x^2|\theta^2) + \dots + \hat{f}^z(x^z|\theta^z) \quad (9)$$

**보조정리 1 :** 각 입력이러한 병렬형 퍼지시스템은 그림 1과 같다. 변수  $(x_1, x_2, \dots, x_n)$ 에 일괄적으로  $k$ 개의 퍼지라벨을 정의할 때 (9)와 같이 재구성된 퍼지시스템이 다음의 조건을 만족한다면 퍼지규칙의 수는 (10)을 만족한다.

$$k^{p_1} + k^{p_2} + \dots + k^{p_z} \leq k^n \quad (10)$$

(조건 P1)  $p_j < n, j=1, 2, \dots, z$

(조건 P2)  $z \leq k, k \geq 1.$

**증명 :** 재구성된 퍼지시스템의 전체 퍼지규칙의 수는

$k^{p_1} + k^{p_2} + \dots + k^{p_z}$  개이며 이는 조건 P2로부터  $z$  가 가질 수 있는 최대값이  $k$  이므로 다음이 성립된다.

$$k^{p_1} + k^{p_2} + \dots + k^{p_z} \leq k^{p_1} + k^{p_2} + \dots + k^{p_k} \quad (11)$$

또한 조건 P1로부터  $p_j$  가 가질 수 있는 최대값은  $n-1$  이므로 다음 식이 성립된다.

$$(12)$$

따라서 재구성된 퍼지시스템의 퍼지규칙의 수는  $k^n$  개보다 작거나 같으며 참여지는 퍼지규칙의 수는  $k^n - (k^{p_1} + k^{p_2} + \dots + k^{p_z})$  개이다. Q.E.D.

2. 종속형 퍼지시스템

종속형 구조를 갖도록 하기 위해서는 근사하고자 하는 비선형 함수가 종속하는 형태로 표현되는 경우, 즉 다음과 같이 표현될 수 있는 경우이다.

$$f(x) = f^c(x^f, g^1(x^1, g^2(x^2, \dots, g^z(x^z, \dots)))) \quad (13)$$

여기서  $x^f \in R^{c_0-1}, x^1 \in R^{c_1-1}, \dots, x^z \in R^{c_z}$  이다. 이러한 비선형 함수에 대한 근사기로서 퍼지시스템은 다음과 같이 재구성될 수 있다.

$$\hat{f}(x|\theta) = \hat{f}^c(x^f, \hat{g}^1(x^1, \hat{g}^2(x^2, \dots, \hat{g}^z(x^z|\theta^z)|\dots|\theta^1)|\theta^1)) \quad (14)$$

이러한 종속형 퍼지시스템은 그림 2와 같다. 여기서 각 퍼지시스템  $\hat{f}^c, \hat{g}^1, \dots, \hat{g}^z$  은 각각 단일출력을 가지는 퍼지시스템이다.

**보조정리 2 :** 각 입력변수  $(x_1, x_2, \dots, x_n)$  에 일괄적으로  $k$  개의 퍼지라벨을 정의할 때 (14)와 같이 재구성된 퍼지시스템이 다음의 조건을 만족한다면 퍼지규칙의 수는 (15)를 만족한다.

$$k^{c_0} + k^{c_1} + \dots + k^{c_z} \leq k^n \quad (15)$$

(조건 C1)  $c_j < n, j=0, 1, \dots, z$

(조건 C2)  $z \leq k-1, k \geq 1.$

**증명 :** 재구성된 퍼지시스템의 전체 퍼지규칙의 수는  $k^{c_0} + k^{c_1} + \dots + k^{c_z}$  개이며 이는 조건 C2로부터  $z$  가 가질 수 있는 최대값이  $k-1$  이므로 다음이 성립된다.

$$k^{c_0} + k^{c_1} + \dots + k^{c_z} \leq k^{c_0} + k^{c_1} + \dots + k^{c_{k-1}} \quad (16)$$

또한 조건 C1로부터  $c_j$  가 가질 수 있는 최대값은  $n-1$  이므로 다음 식이 성립된다.

$$k^{c_0} + k^{c_1} + \dots + k^{c_{k-1}} \leq k^{(n-1)} + k^{(n-1)} + \dots + k^{(n-1)} = k \cdot k^{(n-1)} \quad (17)$$

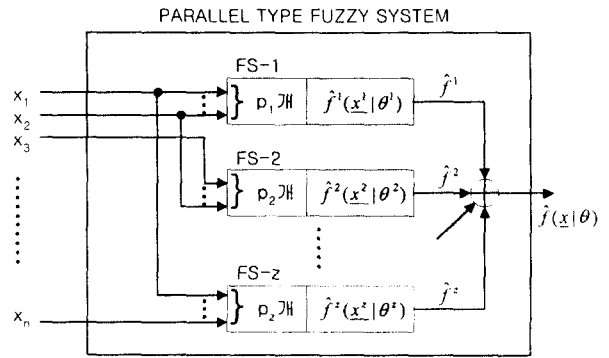


그림 1. 병렬형 퍼지시스템의 구조

Fig. 1. The structure of parallel type fuzzy system

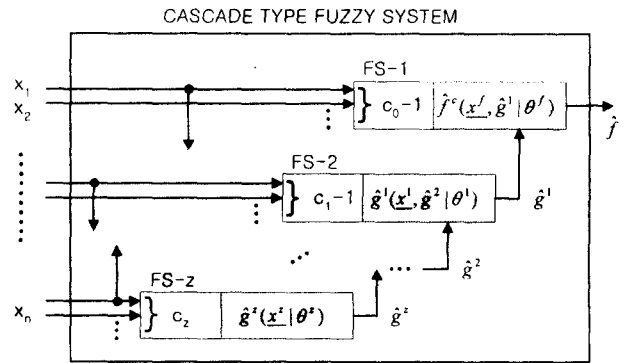


그림 2. 종속형 퍼지시스템의 구조

Fig. 2. The structure of cascade type of fuzzy system

따라서 재구성된 퍼지시스템의 퍼지규칙의 수는  $k^n$  개보다 작거나 같으며 참여지는 퍼지규칙의 수는  $k^n - (k^{c_0} + k^{c_1} + \dots + k^{c_z})$  개이다. Q.E.D.

IV. 재구성된 퍼지시스템을 이용한 적응퍼지 슬라이딩모드제어

본 장에서는 [12]에서의 적응퍼지 슬라이딩모드제어방식 중에서 가장 간단한 첫 번째 알고리즘에 근거한 추적제어기(tracking controller)를 설계한다. 이는 본 논문이 퍼지시스템의 재구성을 통한 퍼지규칙 수의 줄임에 그 주요안점을 두었기 때문에 [12]에서의 견실한 제어방식이나 두번째 방식에 대하여는 생략하기로 한다. 또한 병렬형 예제와 병렬/종속 혼합형 시스템에 대하여 제안된 알고리즘의 타당성을 시뮬레이션을 통하여 보인다. 이 때 사용되는 퍼지시스템은 본 논문에서 제안된 알고리즘을 통해 재구성된 퍼지시스템이다.

먼저 적응퍼지 슬라이딩모드제어방식에 대하여 간단

히 살펴본다. (7)과 같은 시스템에서 앞에서의 가정과 시스템 함수  $f$ 를 완전히 알고 있다면 일반적인 슬라이딩 모드 제어기는 다음과 같이 설계할 수 있다.

$$u = b^{-1} \left( - \sum_{i=1}^{n-1} a_i e^{(i)} - f(x, t) + \dot{x}_d^{(n)} - h \cdot \text{sgn}(s) \eta \right),$$

$$h = \begin{cases} 1, & \text{if } s \neq 0 \\ 0, & \text{if } s = 0 \end{cases} \quad (18)$$

여기서 추적오차  $e = x(t) - x_d(t) = (e, \dot{e}, \dots, e^{(n-1)})^T$ , 슬라이딩평면  $s(e) = a \cdot e = 0$  이며  $a = [a_1, a_2, \dots, a_{n-1}, 1]$ 이며 각  $a_i$ 는 슬라이딩평면  $s(e) = 0$ 이 안정하도록 설계한다. 그러나 앞의 가정에서와 같이  $f$ 는 완전히 알고 있는 함수가 아니므로 (18)과 같은 제어기는 실현 불가능하다. 따라서 퍼지근사기를 이용하여 제어기를 설계해야 하며 이렇게 설계된 제어법칙은 다음과 같다.

$$u = \frac{1}{b} \left( - \sum_{i=1}^{n-1} a_i e^{(i)} - \hat{f}(x|\theta) + \dot{x}_d^{(n)} - h \cdot \text{sgn}(s) \cdot \eta \right) \quad (19)$$

여기에서  $\hat{f}$ 는 (4)와 같은 퍼지시스템이며  $\eta > 0$ 인 실수이다. 이 제어기는 [12]에서의 제안된 적응퍼지 슬라이딩 모드 제어기의 첫번째 방식이며 이에 대한 자세한 안정도 분석은 생략한다. (19)의 퍼지시스템  $\hat{f}$ 를 모든 상태 변수에 대하여 하나의 퍼지시스템으로 설계한다면 그 규칙의 수는  $k^n$ 가 되어  $n$ 이 커짐에 따라서 기하급수적으로 증가하게 된다. 따라서 본 논문에서 제안한 병렬형과 종속형의 두 가지 재구성방식을 적용하여 퍼지규칙의 수를 줄이는 방식을 두 가지 예로 살펴봄으로써 컴퓨터시뮬레이션 결과를 분석한다. 각 시뮬레이션에서 각각의 입력변수에 대한 퍼지라벨은 NB, NS, ZE, PS, PB의 다섯개를 설정하였으며 퍼지집합의 형태는 다음과 같은 가우시안 퍼지집합을 사용하였다.

$$\mu_{A_i^l}(x_i) = \exp \left[ - \left( \frac{x_i - \bar{x}_i^l}{\delta_i^l} \right)^2 \right] \quad (20)$$

여기서  $\bar{x}_i^l$ 은  $l$ 번째 퍼지규칙의  $i$ 번째 입력변수, 즉  $x_i$ 에 대한 퍼지집합  $A_i^l$ 의 소속정도가 1.0인 부분에 대응되는  $U_i$ 내의 한 점이다. 즉,  $R_l$  규칙에서  $U_i$ 에 설정한 퍼지집합  $A_i^l$ 의 위치를 나타내는 값이다. 또한  $\delta_i^l = 24$ 로 하였으며 퍼지시스템의 각 입력변수  $x_1, x_2, x_3, x_4$ 와 예제 2의  $\hat{g}^1$ 에 대한 범위, 즉  $U_i$ 의 범위는  $\pm \pi/6$ 로 설정하였다. 퍼지시스템의 각 파라미터  $\theta$ 의 초기값은 모두  $\theta(0) = 0$ 로 놓았다. 그리고 (18)과 같은 제어법칙에 대한 퍼지시스템의 적용법칙은 다음과 같다.

$$\theta = \Gamma s \xi(x) \quad (21)$$

여기서  $\Gamma$ 는 양의 실수이며 각 시뮬레이션에서 적절히 선택하였다. 이 적용법칙은 두 예제에 대한 모든 시뮬레이션에 적용되는 것이다. 그리고  $x_{1d} = (\pi/10)\sin(t)$ 로 하였으며  $a = [1, 3, 3, 1]$ 로 설정하였다. 각 예제에 대하여 먼저 하나의 퍼지시스템, 즉 퍼지시스템을 재구성하지 않은 경우와 본 논문에서 재구성한 경우에 대하여 각각 시뮬레이션을 행하였으며 그 결과를 비교하였다.

**예제 1.** (병렬형) 다음과 같은 4차 비선형 시스템을 고려하자.

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= -6 \cos x_1 - x_2^2 + x_3 - 4x_4 + 4u \end{aligned} \quad (22)$$

이 때 퍼지시스템으로 근사 시키고자 하는 비선형 함수  $f(x, t) = -6 \cos x_1 - x_2^2 + x_3 - 4x_4$ 를 기존의 방식인 하나의 퍼지시스템으로 구성하자. 각 퍼지시스템의 입력변수  $x_1, x_2, x_3, x_4$ 에 대한 퍼지라벨을 5개로 구분한다면 모두 625개의 퍼지규칙을 필요로 하게된다. 그러나 근사 시키고자 하는 비선형 함수가 (8)과 같이 분해될 수 있는 것을 이용한다면  $f(x) = f^1(x_1) + f^2(x_2) + f^3(x_3) + f^4(x_4)$ 와 같이 분해될 수 있으며 여기서  $f^1(x_1) = -6 \cos(x_1)$ ,  $f^2(x_2) = -x_2^2$ ,  $f^3(x_3) = x_3$ ,  $f^4(x_4) = -x_4$ 가 된다. 따라서 (9)와 같은 병렬 형태로 퍼지시스템을 재구성 할 수 있다.

$$\hat{f}(x|\theta) = \hat{f}^1(x_1|\theta^1) + \hat{f}^2(x_2|\theta^2) + \hat{f}^3(x_3|\theta^3) + \hat{f}^4(x_4|\theta^4) \quad (23)$$

따라서 전체 퍼지시스템은  $x_1$ 을 입력으로 하는  $\hat{f}^1$ ,  $x_2$ 을 입력으로 하는  $\hat{f}^2$ ,  $x_3$ 을 입력으로 하는  $\hat{f}^3$ , 그리고  $x_4$ 을 입력으로 하는  $\hat{f}^4$ 의 4개의 sub-blocks으로 재구성될 수 있다. 이와같이 병렬형으로 구분하여 퍼지시스템을 구성한다면 20개의 퍼지규칙만을 필요로 하게된다.

**예제 2.** (종속형과 병렬형의 혼합형) 다음과 같은 4차 비선형 시스템을 고려하자.

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= 2 \cos(x_1 + x_2)x_3 - 5x_4 + 3u \end{aligned} \quad (24)$$

이 경우에도 퍼지시스템으로 근사 시키고자 하는 비선형 함수  $f(x) = 2 \cos(x_1 + x_2)x_3 - 5x_4$ 를 기존의 방식인 하나의 퍼지시스템으로 구성한다면 입력변수가 4개이기 때문에 퍼지라벨을 각각 5개로 구분한다면 모두 625개의

퍼지규칙을 필요로 하게된다. 그러나 근사 시키고자 하는 비선형 함수를 (8)과 (13)의 혼합된 형태로 적절히 분해하면  $f(x) = f^3(g^1(x_1, x_2), x_3) + f^4(x_4)$  로 표현 가능하다. 여기서

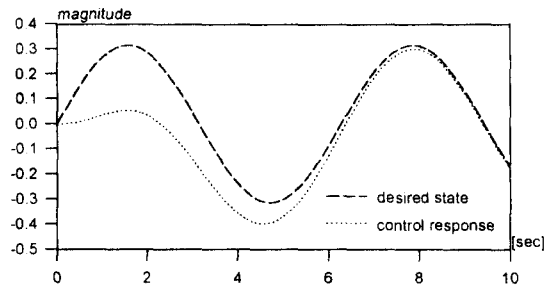
$$g^1(x_1, x_2) = f^1(x_1) + f^2(x_2), \quad f^1(x_1) = x_1,$$

$$f^2(x_2) = x_2, \quad f^3(g^1(x_1, x_2), x_3), \quad f^4(x_4) = -5x_4$$

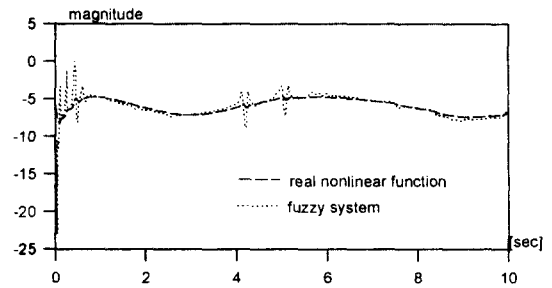
가 되며 퍼지시스템을 다음과 같이 재구성 할 수 있다.  $\hat{K}_x|\theta = \hat{F}^3(\hat{g}^1(x_1, x_2|\theta^k), x_3|\theta^3) + \hat{F}^4(x_4|\theta^4)$  (25) 따라서 이 경우의 퍼지시스템은  $x_1$ 을 입력으로 하는  $\hat{F}^1$ ,  $x_2$ 을 입력으로 하는  $\hat{F}^2$ ,  $x_4$ 을 입력으로 하는  $\hat{F}^4$  와  $\hat{F}^1$ 과  $\hat{F}^2$ 의 출력 합  $\hat{g}^1$ 과  $x_3$ 를 입력으로 하는  $\hat{F}^3$ 의 4개의 sub-blocks로 재구성될 수 있다. 여기서  $\hat{g}^1(x_1, x_2|\theta^k) = \hat{F}^1(x_1|\theta^1) + \hat{F}^2(x_2|\theta^2)$ 이며  $\theta^1, \theta^2, \theta^4$ 는 각각  $5 \times 1$  벡터이며  $\theta^k$ 는  $25 \times 1$  벡터이다. 따라서 이 경우에는 40개의 퍼지규칙을 사용하게 된다.

그림 3은 예제 1의 병렬형에 대한 것으로서, 하나의 퍼지시스템, 즉  $\hat{K}_x|\theta$ 를 사용한 것이며 (a)는 시스템의 원하는 상태궤적과  $\hat{F}$ 를 사용한 제어응답을, (b)는 실제 비선형 함수와 퍼지시스템의 값을 나타낸 것이다. 이 경우에는 퍼지규칙이 모두 625개가 사용된다. 그림 4 또한 예제 1의 병렬형에 대한 것으로서 주어진 비선형 함수를 (23)과 같이 분해하여 퍼지시스템을 재구성한 경우이며 이 때 사용된 퍼지규칙은 25개에 불과하다. 그림 3과 4에서 주어진 비선형 함수를 본 논문에서 제안한 알고리즘을 이용하여 적절히 퍼지시스템을 재구성하면 그 규칙의 수를 대단히 많이 줄일 수 있으며 그 성능에서도 큰 차이가 없음을 알 수 있다.

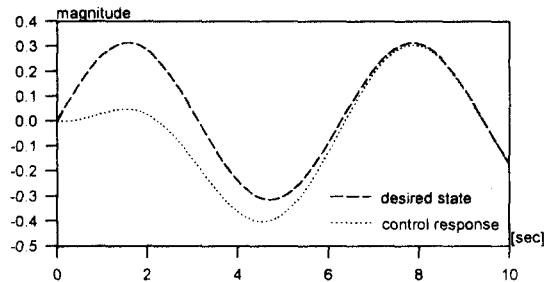
그림 5와 6은 예제 2의 병렬형과 종속형의 혼합형태를 시뮬레이션 한 결과이다. 먼저 그림 5는 하나의 퍼지시스템을 사용한 경우이며 이 때에도 모두 625개의 퍼지규칙이 사용된 결과이다. 그림 6은 (25)와 같이 병렬형과 종속형을 적절히 이용하여 퍼지시스템을 재구성한 경우이며 이 때 사용된 퍼지규칙의 수는 40개이다. 이 모든 결과는 슬라이딩 평면  $s(e)$ 의 설정과  $\Gamma$ 의 값에 매우 의존적이기 때문에 정확히 비교할 수는 없으나 이 두 가지 예를 통하여 본 논문에서 제안된 퍼지규칙 수 줄임 알고리즘이 타당함을 알 수 있다. 즉, 근사하고자 하는 비선형 함수를 파라미터의 값을 정확히 알 수 없으나 그 형태에 따라 병렬형이나 종속형, 또는 그 혼합형으로 적절히 분해하여 퍼지시스템을 재구성하여도 그 비선형 함수의 근사에는 큰 차이가 없다는 것을 알 수 있다.



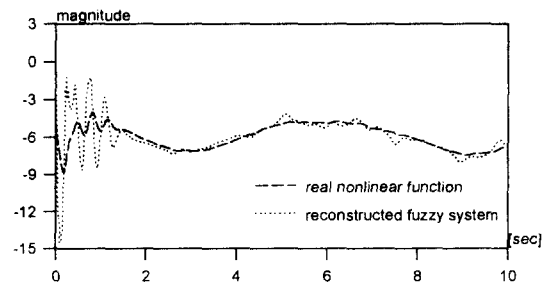
(a) 원하는 상태궤적과  $\hat{K}_x|\theta$ 를 이용한 제어응답  
(a) Desired state and control response with  $\hat{K}_x|\theta$



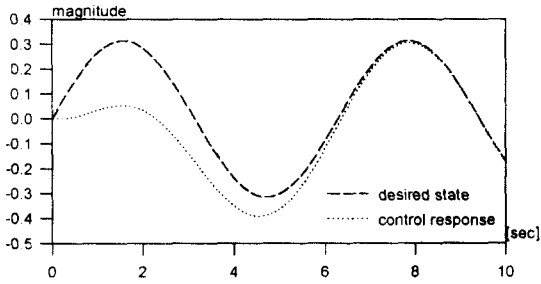
(b) 실제 함수  $f$  와 퍼지시스템  $\hat{K}_x|\theta$   
(b) Real function  $f$  and fuzzy system  $\hat{K}_x|\theta$   
그림 3. 예제 1 시뮬레이션 (퍼지규칙 : 625개),  $\Gamma = 5000$   
Fig. 3. Simulation for example 1 (625 rules),  $\Gamma = 5000$



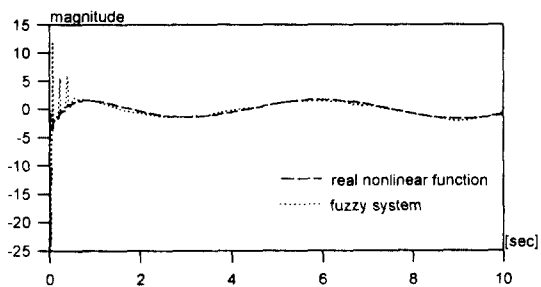
(a) 원하는 상태궤적과 (23)의  $\hat{F}$ 를 이용한 제어응답  
(a) Desired state and control response with (23),  $\hat{F}$



(b) 실제 함수  $f$  와 퍼지시스템 (23)의  $\hat{F}$   
(b) Real function  $f$  and fuzzy system (23),  $\hat{F}$   
그림 4. 예제 1 시뮬레이션 (퍼지규칙 : 25개),  $\Gamma = 100$



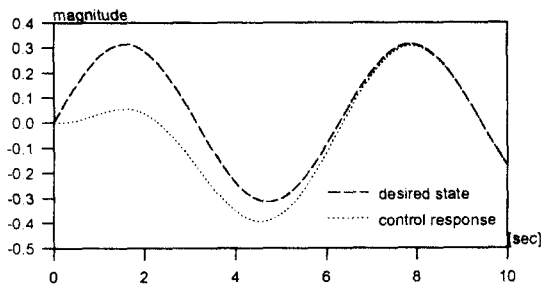
(a) 원하는 상태궤적과  $\hat{\chi}(x|\theta)$ 를 이용한 제어응답  
(a) Desired state and control response with  $\hat{\chi}(x|\theta)$



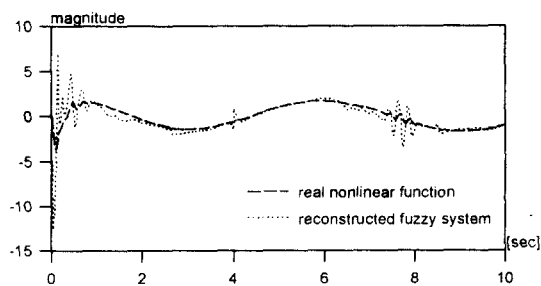
(b) 실제 함수  $f$  와 퍼지시스템  $\hat{\chi}(x|\theta)$

(b) Real function  $f$  and fuzzy system  $\hat{\chi}(x|\theta)$

그림 5. 예제 2 시뮬레이션 (퍼지규칙: 625개),  $\Gamma=10000$   
Fig. 5. Simulation for example 2 (625 rules),  $\Gamma=10000$



(a) 원하는 상태궤적과 (25)의  $\hat{f}$ 를 이용한 제어응답  
(a) Desired state and control response with (25),  $\hat{f}$



(b) 실제 함수  $f$  와 퍼지시스템 (25)의  $\hat{f}$

(b) Real function  $f$  and fuzzy system (25),  $\hat{f}$

그림 6. 예제 2 시뮬레이션 (퍼지규칙: 40개),  $\Gamma=500$   
Fig. 6. Simulation for example 2 (40 rules),  $\Gamma=500$

## V. 결론

본 논문에서는 compact 입력공간상에 정의된 어떠한 비선형 함수도 임의의 오차이내에서 근사시킬 수 있다는 퍼지시스템에 대하여 살펴보았으며 이러한 퍼지시스템의 응용에서 입력변수의 수가 증가함에 따라 기하급수적으로 증가하는 퍼지규칙의 수를 줄이기 위해 비선형 함수의 형태를 적절히 이용하여 함수를 분해하고 이 분해 결과를 토대로 병렬형과 종속형으로 퍼지시스템을 재구성하는 방법을 제안하였다. 이 방법은 퍼지규칙을 미리 설계하지 않고 실시간으로 비선형 함수를 근사해 가는 적응퍼지시스템의 응용에서 사용 가능한 방식이다. 또한 이 알고리즘은 파라미터를 완전히 알 수 없는 플랜트에 대한 퍼지모델링에서도 규칙의 수를 줄이는 방법으로 사용될 수 있다. 두 예제시스템에 대한 적응퍼지 슬라이딩모드제어에 적용하여 그 타당성을 보였으며 재구성을 통하여 퍼지규칙의 수를 줄일 수 있음을 간단히 증명해 보았다. 제안된 알고리즘을 적절히 이용한다면 퍼지규칙의 수가 증가함에 따라 그 응용에 있어 발생하는 어려움을 어느 정도 해결할 수 있을 것으로 생각된다. 향후 연구계획은 이러한 방식을 통한 퍼지제어시스템의 응용에 대하여 실제시스템에 적용시키는 연구와 이와 더불어 본 논문에서 가정한 모든 상태의 측정가능성이 실제 시스템에서는 구현이 어려우므로 관측기의 설계에 대한 연구가 계속되어야 하겠다.

접수일자 : 2001. 2. 23      수정완료 : 2001. 4. 3

## 참고문헌

- [1] C. C. Lee, Fuzzy logic in control system : Fuzzy logic controller-Part I, II, IEEE Trans. on Systems, Man, and Cybernetics, vol. SMC-20, no. 2, pp.404-435, 1990.
- [2] H. J. Zimmermann, Fuzzy set theory and its applications, Kluwer Academic Publishers, 1991.
- [3] L. X. Wang, Adaptive fuzzy systems and control, Prentice-Hall International, Inc., 1994.
- [4] L. X. Wang and J. M. Mendel, Fuzzy basis functions, universal approximation, and orthogonal least squares learning, IEEE Trans. Neural Networks, vol. 3, no. 5, pp. 807-814, 1992.
- [5] L. X. Wang, Design and analysis of fuzzy identifiers of nonlinear dynamic systems, IEEE Trans. on Control Systems Technology, vol. 4, no.2. 1996.

no.2, 1996.

[6] J. C. Wu and T. S. Liu, A sliding-mode approach to fuzzy control systems, IEEE Trans. on Fuzzy Systems, vol. 1, no. 2, 1993.

[7] M. B. Ghalia and A. T. Alouani, Sliding mode control synthesis using fuzzy logic, Proc. of the American Control Conference, Seattle, Washington, pp. 1528-1532, 1995.

[8] G. C. Whang and S. C. Lin, A stability approach to fuzzy control design for nonlinear systems, Fuzzy sets and Systems, vol. 48, pp. 279-287, 1992.

[9] R. Palm, Sliding mode fuzzy control, IEEE Int. Conf. on Fuzzy Systems, San Diego, California, USA, pp. 519-526, 1992.

[10] S. C. Lin and Y. Y. Chen, Design of adaptive fuzzy sliding mode for nonlinear system control, Third IEEE Int. Conf. on Fuzzy Systems(FUZZ-IEEE), Orlando, Florida, USA, pp. 35-39, 1994.

[11] F. C. Sun, Z. Q. Sun and G. Feng, Design of adaptive sliding mode controller for robot manipulators, Fifth IEEE Int. Conf. on Fuzzy Systems(FUZZ-IEEE), New Orleans, USA, vol. 1, pp. 62-67, 1996.

[12] B. K. Yoo and W. C. Ham, Adaptive fuzzy sliding mode control of nonlinear system, IEEE Trans. on Fuzzy Systems, vol. 6, no. 2, pp.315-321, 1998.

[13] B. K. Yoo and W. C. Ham, Adaptive control of robot manipulator using fuzzy compensator, IEEE Trans. on Fuzzy Systems, vol. 8, no. 2, pp. 186-199, 2000.

[14] L. T. Koczy and K. Hirota, "Size reduction by interpolation in fuzzy rule base," IEEE Trans. System, Man, and Cybernetics-Part B: Cybernetics, vol. 27, no. 1, 1997.

[15] Y. Yam, P. Baranyi and C. T. Yang, "Reduction of fuzzy rule base via singular value decomposition," IEEE Trans. Fuzzy Systems, vol. 7, no. 2, 1999.

[16] S. Galichet and L. Foulloy, "Size reduction in fuzzy rulebases," Proc. of IEEE International Conference on Systems, Man, and Cybernetics, California, USA, vol 3, pp. 2107-2112, 1998.

[17] G. Castellano and A. M. Fanelli, "A neuro-fuzzy model reduction strategy," Proc. of International Conf. on Neural Networks(ICNN'97), Texas, USA, vol. 1, 1997.

[18] J. Yen and L. Wang, "Simplifying fuzzy

rule-based models using orthogonal transformation methods," IEEE Trans. Systems, Man, and Cybernetics-Part B: Cybernetics, vol. 29, no. 1, 1999.



유병국(Bung-Kook Yoo)

正會員

1992년 전북대학교 전자공학과 공학사

1995년 전북대학교 전자공학과

공학석사

1999년 전북대학교 전자공학과

공학박사

1992년~1993년 포항종합제철주식회사 제어시스템부

1996년~1999년 한려대학교 의공학과 전임강사

2000년~현재 한려대학교 멀티미디어정보통신공학과

전임강사

관심분야: 퍼지시스템, 적응제어, 가변구조시스템,

로봇제어, 자바 등