

규칙 제거 기능이 있는 자기구성 퍼지 시스템 Self-Organizing Fuzzy Systems with Rule Pruning

이창욱*, 이평기**
Chang-Wook Lee*, Pyeong-Gi Lee**

<Abstract>

In this paper a self-organizing fuzzy system with rule pruning is proposed. A conventional self-organizing fuzzy system having only rule generation has a drawback in generating many slightly different rules from the existing rules which results in increased computation time and slowly learning. The proposed self-organizing fuzzy system generates fuzzy rules based on input-output data and prunes redundant rules which are caused by parameter training. The proposed system has a simple structure but performs almost equivalent function to the conventional self-organizing fuzzy system. Also, this system has better learning speed than the conventional system. Simulation results on several numerical examples demonstrate the performance of the proposed system.

Key words : *Self-organizing Fuzzy System, Rule Pruning*

1. 서론

고전적 제어이론에서는 공정의 모델을 기초로 해석적으로 접근하여 적절한 제어입력량을 산출하므로 제어대상의 수학적 모델링이 시스템의 성능을 결정하는 중요한 역할을 한다. 그러나 시스템이 복잡하거나 불확실성이 존재하면 수학적으로 시스템을 정확히 모델링할 수 없을 뿐만 아니라 원하는 제어 목적을 달성하기

가 어렵거나 불가능하다. 따라서 제어이론의 괄목할만한 발전에도 불구하고 실제 산업현장에서는 여전히 숙련된 전문가의 경험에 의존하는 부분이 많은 실정이다. 이와 같이 시스템의 주변환경에 익숙하면서 경험이 많은 전문가의 지식이나 정보를 이용하여 제어규칙을 만들고 이러한 규칙을 토대로 현재 상황에 가장 적절한 제어입력을 추론하는 퍼지제어 이론이 그 동안 활발히 연구가 진행되어 여러 산업분야에 성공

* 정회원 : 경북대학교 전기전자공학부, 박사과정
701-811 대구시 동구 신암5동 131-13 낙원빌라 302호
** 정회원 : 위덕대학교 정보전자공학부, 공학박사
경북대학교 대학원 졸업

* School of Electronic and Electrical Engineering
Kyungpook National University
** School of Informatin and Electronic Engineering
Uiduk University
E-mail : pglee@mail.uiduk.ac.kr

적으로 적용되고 있다. 퍼지제어이론은 제어대상 시스템의 정확한 모델링이 없어도 제어가 가능한 것이 가장 큰 장점이나 복잡한 화학공정제어와 같은 비선형 시스템에 대해 믿을 수 있는 제어규칙을 구성하는 것이 어렵고 시스템 주변의 예기치 못한 외란이나 잡음에 대해서도 견실한 성능을 얻을 수 없는 것이 문제점이다.

이러한 문제점을 해결하기 위해서 모사 또는 제어대상에 대한 규칙을 상황에 따라 적절하게 스스로 구성해 가면서 모사 또는 제어하는 자기구성 퍼지시스템(self-organizing fuzzy system)에 대한 이론이 제시되었다. 주변환경의 변화에 스스로 적응하여 원하는 목표값을 추적하도록 하는 자기구성 퍼지시스템은 숙련된 전문가의 경험에 의해서도 규칙을 생성하기 어려운 복잡한 비선형 시스템 등의 제어에 매우 유용하다. 자기구성 퍼지시스템은 1979년 Procyk과 Mamdani¹⁾에 의해 처음으로 소개된 이래로 많은 연구가 계속되고 있다^{2,3,4,5)}.

그러나 지금까지의 일반적인 자기구성 퍼지제어기는 퍼지규칙을 계속 생성만 하므로 규칙수가 많아져 복잡한 규칙 베이스를 갖는 문제점이 있었다. 많은 규칙수는 계산량을 증가시킬 뿐만 아니라 학습능력을 저하시키는 원인이 된다. 이와 같은 문제점은 신경회로망에서도 나타나는데 여기에서는 Network-pruning이라는 방법으로 이를 보완하고 있다. 이 방법은 많은 수의 다층신경회로망에서 시작하여 출력에 영향이 미미한 가지를 잘라내는 것인데, 신경회로망의 크기를 적정수준으로 만들고 과도학습의 문제를 방지하는 장점이 있다. 퍼지분야에서도 최근 Tang 등의 논문⁶⁾에서는 유전알고리즘을 이용하여 최적의 규칙 베이스를 구성하는 과정에 대한 연구결과를 발표하면서 규칙수가 적은 것을 장점 중 하나로 언급하고 있다. 이렇게 퍼지시스템의 규칙수를 줄이는 것이 필요함에도 불구하고, 규칙 베이스를 자기구성하는 과정에서 규칙을 제거하는 연구결과는 아직 발표된 바가 없다.

이 논문에서는 규칙수가 불필요하게 많이 늘어나는 문제점을 보완하기 위해서 퍼지규칙의 생성뿐만 아니라 학습을 통한 후에 불필요하다고 판단되는 규칙들을 제거하는 자기구성 퍼지

시스템을 제안한다. 즉, 매 샘플링 단계마다의 입출력 정보가 새로운 것인지를 판별하여 그 정보가 새로운 것이라면 새로운 규칙을 만들고, 오차역전파방법으로 퍼지집합의 변수들을 학습한 후 유사한 규칙들을 하나로 합쳐 규칙수를 줄여나간다. 그리고 규칙의 생성과 제거가 동시에 이루어지는 자기구성 알고리즘으로 퍼지시스템의 규칙 베이스를 구성하여 지나친 규칙수의 증가를 막을 수 있음을 모의실험을 통해 확인한다.

2. 자기구성 퍼지시스템

본 논문에서 사용하는 퍼지추론시스템은 일반적으로 많이 사용하는 Mamdani구조로, 퍼지시스템의 i 번째 규칙은 다음과 같다.

If x_1 is μ_1^i and x_2 is $\mu_2^i \dots$ and x_N is μ_N^i ,
then y^i is r^i .

(1)

위 규칙은 R^i 로 정의할 수 있고, 퍼지시스템의 i 번째 규칙을 의미한다. x_j 는 입력, μ_j^i 는 x_j 에 대한 퍼지집합, y^i 는 규칙 R^i 로부터의 출력을 의미한다. 또 퍼지집합 μ_j^i 는 다음과 같은 가우스함수를 사용한다.

$$\mu_j^i = \exp\left[-\left(\frac{x - c_j^i}{\sqrt{2}\sigma_j^i}\right)^2\right] \quad (2)$$

각 퍼지규칙의 조건부의 적합도 $w^i(x)$ 는 T-노움연산을 통해 계산된다. 이렇게 구해진 한 규칙의 적합도는 모든 규칙의 적합도에 대한 합의 비를 계산하여 정규화시킨다.

$$\bar{w}^i = \frac{w^i(x)}{\sum_{i=1}^I w^i(x)} \quad (3)$$

여기서 I 는 최종 규칙수이며, 최종적인 퍼지추론시스템의 출력 y 는 다음 식으로 구할 수 있다.

$$y = \sum_{i=1}^I \bar{w}^i(x) y^i = \frac{\sum_{i=1}^I w^i(x) y^i}{\sum_{i=1}^I w^i(x)} \quad (4)$$

한편, 자기구성 퍼지시스템의 규칙 베이스의 자기구성과정을 살펴보면 다음과 같다. 먼저 매

샘플링 단계마다의 입출력 정보로부터 규칙의 생성 여부를 결정한다. 규칙의 생성 여부는 현재 입출력 정보가 주는 영향을 기존의 규칙으로 표현할 수 있는가로 판단한다. 즉, 기존의 규칙으로 적합도를 구하여 사용자가 정하는 수치, $G_{threshold}$ 보다 작으면 규칙을 생성하는 것이다. 그리고 나서 규칙을 생성해야 한다고 판단되면 정해진 방법으로 규칙을 만들어 규칙 베이스에 저장하게 된다. 마지막으로 규칙 베이스의 변수들을 정해진 학습방법으로 학습한다.

3. 제안한 자기구성 퍼지시스템

퍼지 제어 규칙들이 모든 입출력 공간에 적당히 분포할 수 있도록 α -cut개념을 도입한다. 시행착오방법으로 구해진 G_α 와 P_α 를 이용하여 새로운 규칙의 생성과 이미 존재하는 두 규칙의 결합을 반복하므로써 규칙 제거 기능이 있는 퍼지시스템을 구현할 수 있다.

시스템의 입출력 데이터를 이용하여 퍼지시스템의 제어규칙을 생성하고 제거하는 과정은 Fig. 1과 같고, 각각의 과정은 다음과 같다.

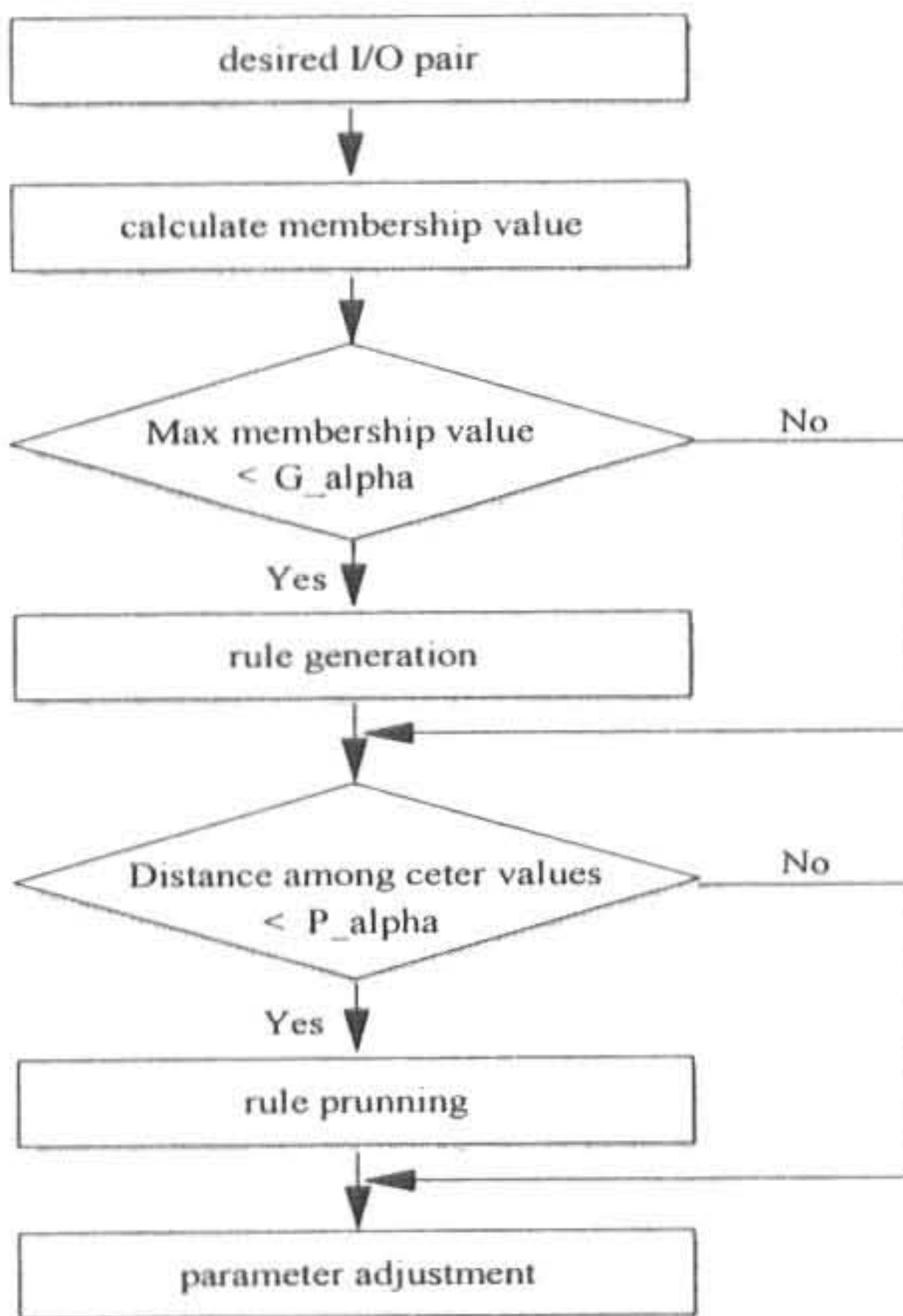


Fig. 1. Self-organizing procedure of rule base.

step 1 : 새로운 입출력 데이터가 퍼지 추론 시스템의 입력이 된다.

step 2 : 기존의 퍼지규칙의 소속값 $\mu_i(x_i)$ 를 구한다.

step 3 : step 2에서 구한 소속값들 중 최대값이 미리 정해진 임의의 상수 G_α 보다 작은 경우 step 4로 가고, 작지 않은 경우 step 5로 간다.

즉 새로운 입출력 데이터가 기존의 제어규칙의 소속정도가 작아서 기존의 규칙으로 이 입출력 데이터를 영향을 잘 나타낼 수 없다면 이 입출력 데이터로부터 새로운 규칙을 만드는 것이다.

step 4 : 입력이 x_i 이고 원하는 출력 y_d 인 데이터로부터 새로운 제어 규칙을 식(1)처럼 만들어 낸다. 이때 x_i 가 μ_i 의 c_i 값이 되고, 원하는 출력 y_d 가 결론부의 싱글톤 값, r^i 가 된다. 그리고 μ_i 의 σ_i 값은 적절히 정해진 값을 준다.

step 5 : 현재 존재하는 퍼지규칙들 중에서 어떤 두 규칙의 모든 센터값들의 차이가 P_α 보다 모두 작은 경우는 step 6으로 가고, 그렇지 않은 경우에는 step 7로 간다. 즉 기존의 퍼지규칙들이 학습의 과정을 거치면서 서로 비슷한 규칙으로 되면 그 두 규칙을 합쳐서 하나의 퍼지규칙으로 만들어 하나의 퍼지규칙을 제거하는 효과를 낸다.

step 6 : 비슷해진 퍼지규칙을 하나로 합친다. 만일 유사한 규칙이 i 번째 규칙과 j 번째 규칙이라면, 이 과정은 다음과 같이 각 변수에 대한 산술적인 평균을 이용한다.

$$\mu_{new}^i = \exp \left[- \left(\frac{x - (c_i + c_j)/2}{\sqrt{2}(\sigma_i + \sigma_j)/2} \right)^2 \right] \quad (5)$$

$$r_{new}^j = \frac{r^i + r^j}{2} \quad (6)$$

step 7 : 각각의 변수에 대해서 경사법(Gradient Descent Method)에 기초한 학습방법인 오차역전파방법(Error Back Propagation Method)을 사용한다[5]. 퍼지 시스템의 전반부 소속함수의 변수 c_i , σ_i 와 결론부 변수인 r^i 를 학습하여 퍼지 시스템의 출력이 목표치를 따라가도록 최소화하는 비용함수로 다음을 사용한다.

$$E = \frac{1}{2} [y - y_d]^2 \quad (7)$$

여기서 y 는 퍼지 시스템의 출력이고 y_d 는 원하는 출력의 목표치이다. 전반부와 결론부 변수들

의 학습식은 다음과 같다.

$$c_j^i(k+1) = c_j^i(k) - \eta_c \left(\frac{w^i}{\sum_{i=1}^I (w_i)} \right) \left(\frac{x_j - c_j^i}{\sigma_j^i} \right) (y^j - y)(y - y_d) \quad (8)$$

$$\sigma_j^i(k+1) = \sigma_j^i(k) - \eta_\sigma \left(\frac{w^i}{\sum_{i=1}^I (w_i)} \right) \left(\frac{(x_j - c_j^i)^2}{\sigma_j^i} \right) (y^j - y)(y - y_d) \quad (9)$$

$$r^i(k+1) = r^i(k) - \eta_r \left(\frac{w^i}{\sum_{i=1}^I (w_i)} \right) (y - y_d) \quad (10)$$

하나의 입출력 데이터마다 위 step들을 반복함으로써 규칙 생성 및 제거가 가능한 자기구성 퍼지시스템을 구성할 수 있다.

4. 모의 실험 결과

예제 1. 입력 x 와 원하는 목표치 y_d 는 다음과 같다[6].

$$x(k) = \begin{cases} \sin(2\pi k/250) & (1 \leq k \leq 250) \\ 0.8\sin(2\pi k/250) + 0.2\sin(2\pi k/25) & (250 < k \leq 500) \end{cases}$$

$$y_d(k) = \frac{y(k-1)y(k-2)y(k-3)u(k-1)(y(k-3)-1) + u(k)}{1 + y^2(k-3) + y^2(k-2)}$$

예제 1에 대한 모의 실험 결과는 Fig. 2와 같고, 여기서 실선은 목표치이고 점선은 규칙생성만을 하는 기존의 자기구성 퍼지시스템이고, 일점쇄선은 제안한 퍼지시스템이다. 이 예제에서는 G_a 는 0.2, P_a 는 0.7, σ_i 는 1, η_c 와 η_σ 는 0.3이고, η_r 는 1의 값을 사용하였다. 그리고 두 시스템 모두 오차자승의 합은 0.3335로 같았고, 규칙 수도 동일하였다. 이 예제에서는 제안한 방법의 step 5를 만족하는 입출력 데이터쌍이 없었음을 의미한다.

예제 2. 입력 x 와 원하는 목표치 y_d 는 다음과 같다[5].

$$x(k) = \begin{cases} \sin(2\pi k/250) & (1 \leq k \leq 250) \\ 0.5\sin(2\pi k/250) + 0.5\sin(2\pi k/25) & (250 < k \leq 500) \end{cases}$$

$$y_d(k) = 0.6\sin(\pi x(k)) + 0.3\sin(3\pi x(k)) + 0.1\sin(5\pi x(k))$$

예제 2에 대해서 기존의 퍼지시스템의 모사 결과는 그림 3과 같고, 제안한 퍼지 시스템의 모사 결과는 Fig. 4에 나타나 있다. 여기서 실선은 목표치이고, 점선은 각각의 퍼지시스템의 모사결과이다. 그리고 예제 1과 같은 변수 값을 사용하였

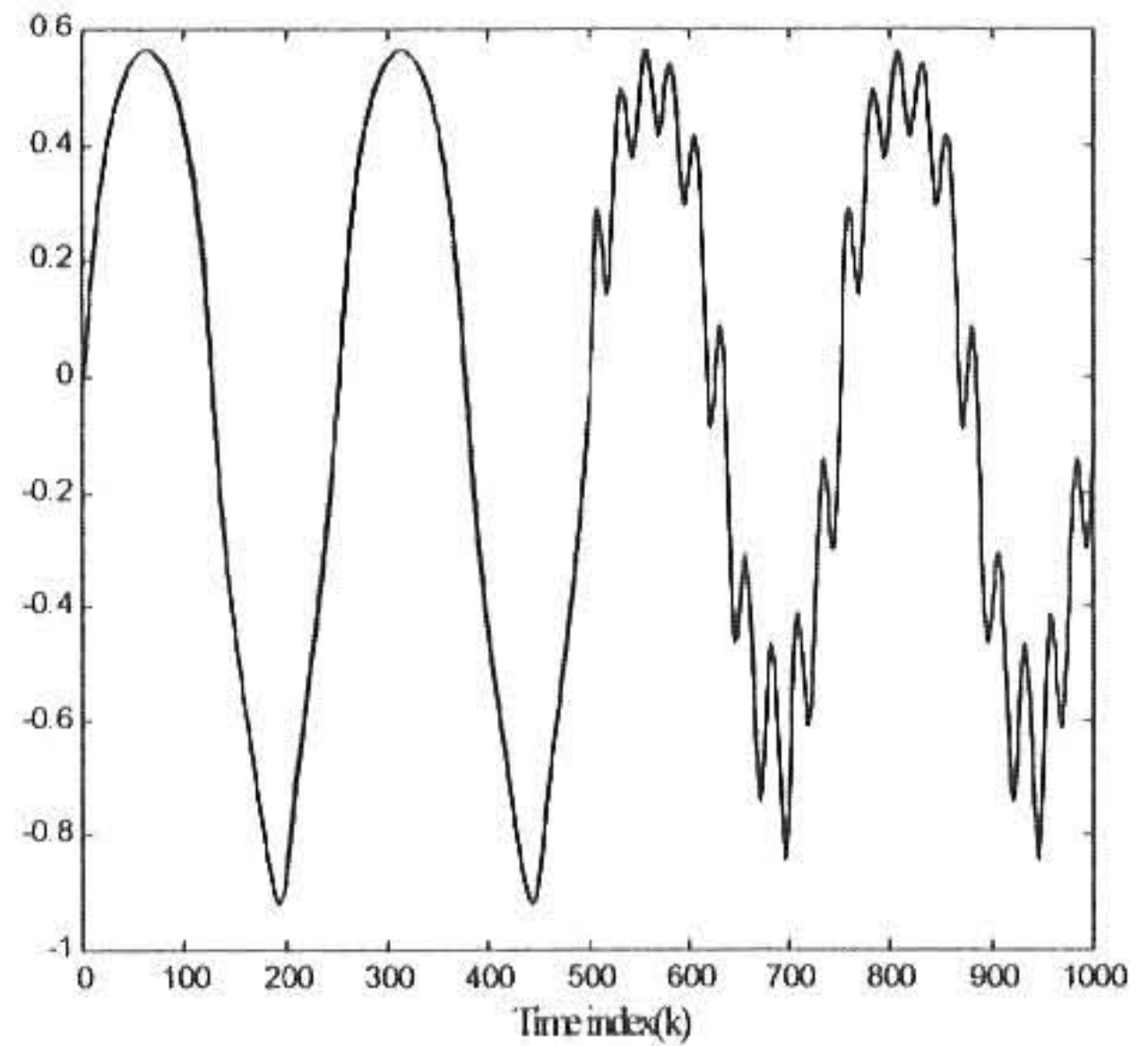


Fig. 2. Identification result(Example1).

다. Fig. 3, 4에서 기존의 퍼지시스템으로 모사한 결과의 오차자승합은 10.26이고, 제안한 퍼지시스템은 오차자승합은 10.39이다. 그리고 그림 5는 세 번의스위프동안에 나타난 제어규칙수의 변화를 나타낸 것이다. Fig. 5에서 실선은 규칙생성만을 하는 퍼지시스템의 규칙수이고, 점선은 생성 및 제거를 하는 퍼지시스템의 규칙수이다. 여기서 나타난 것처럼 규칙의 생성과 제거가 동시에 이루어짐을 알 수 있다. 이 예제에서는 제안한 방법을 적용하여 시스템의 오차자승합은 비슷하면서도 규칙수가 적은 간단한 퍼지시스템이 설계됨을 알 수 있었다.

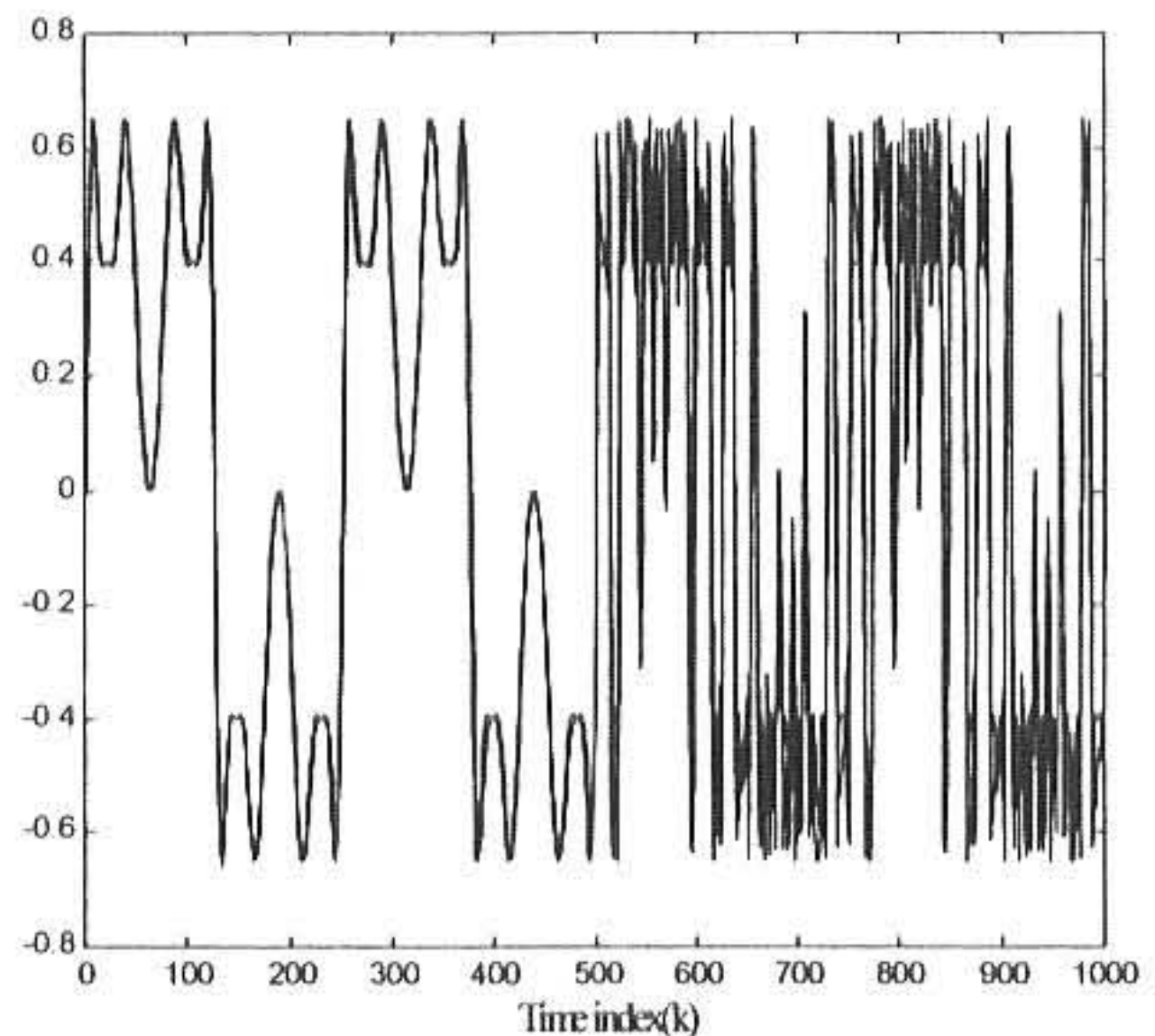


Fig. 3. Identification result of a conventional self-organizing fuzzy system(Example 2).

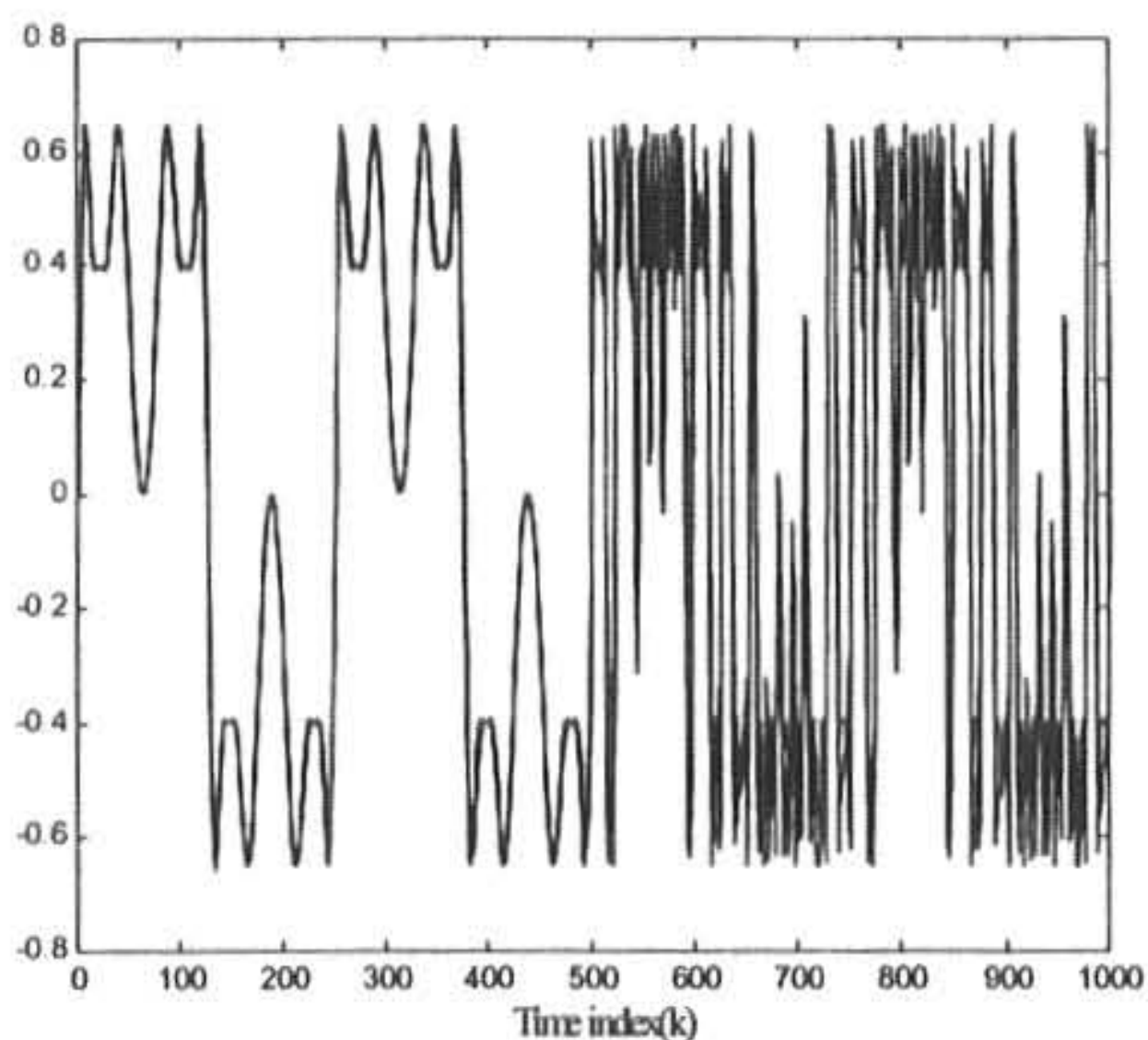


Fig. 4. Identification result of the proposed self-organizing fuzzy system(Example 2).

의 데이터를 사용하였고, 학습과정을 두 번 시행하였다. 그리고 예제 1과 같은 변수값을 사용하였다.

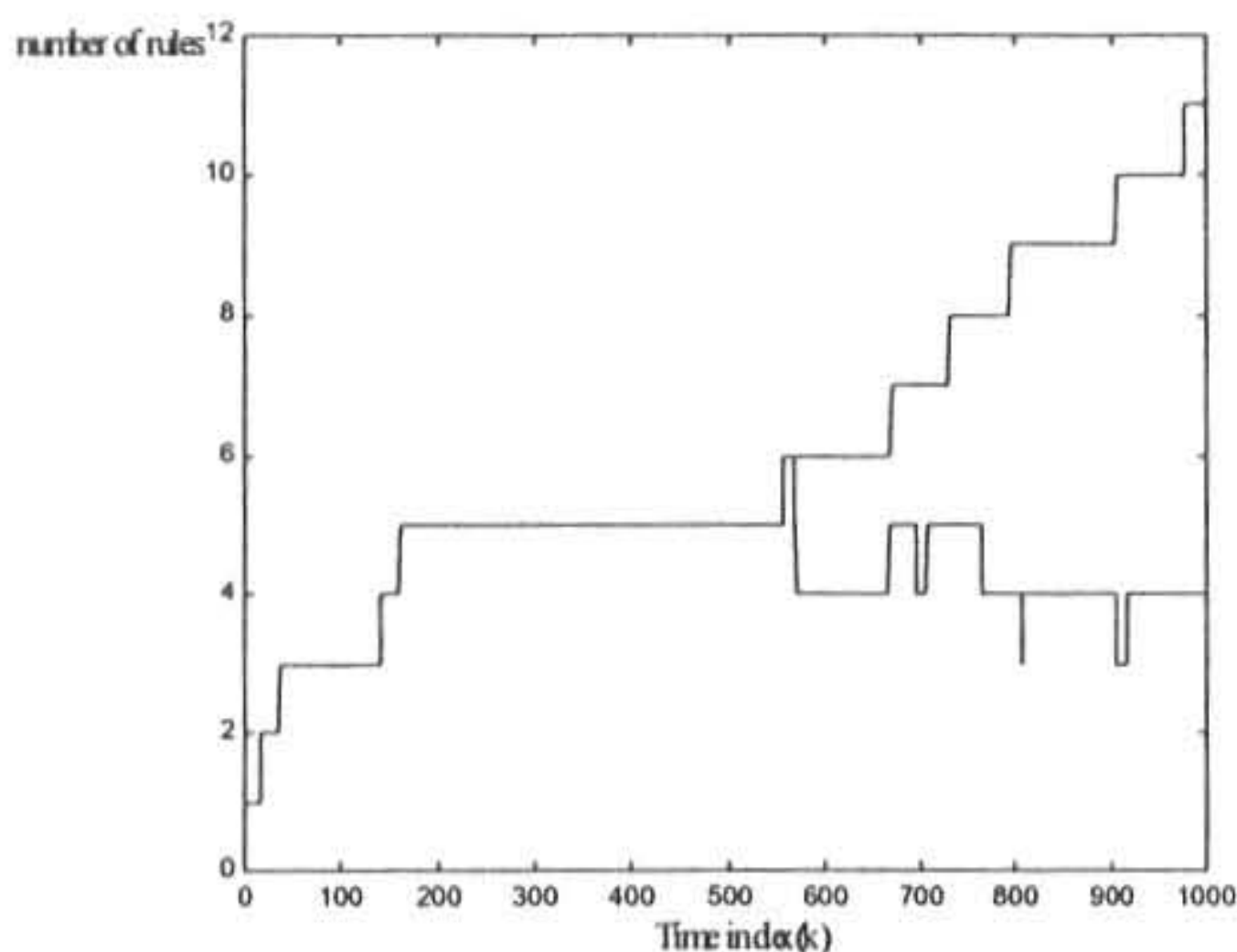


Fig. 6. Variations of number of rules(step 7 is performed twice).

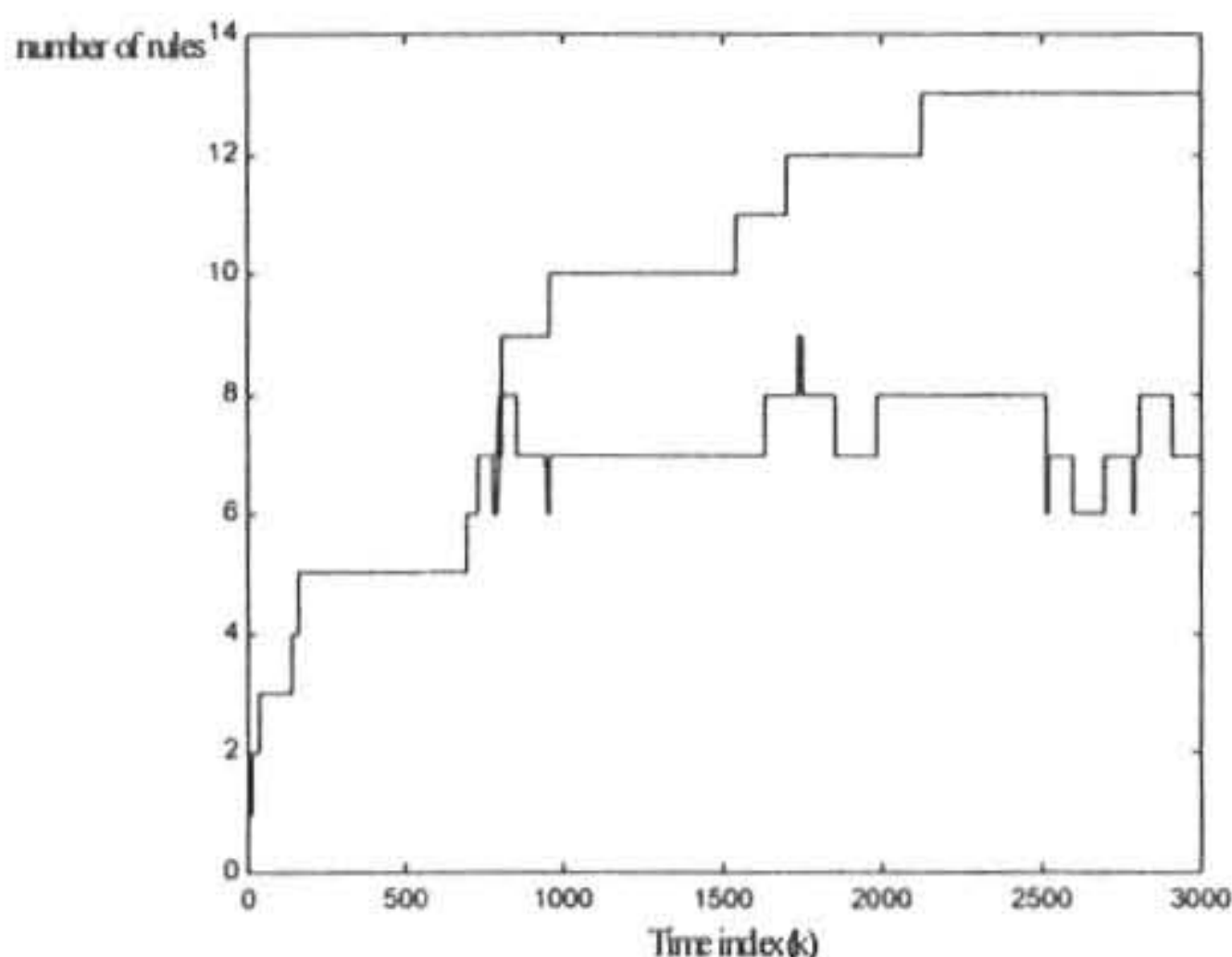


Fig. 5. Variation of number of rules.

보다 나은 모사결과를 위해 한 타임스텝에서 두 번의 학습을 한 경우에는 기존의 퍼지시스템의 오차자승합이 9.11이고, 제안한 방법의 퍼지시스템에서는 9.17로 줄어들었다. 그리고 규칙 수의 변화는 Fig. 6과 같다. 한 번의 학습만 했을 때보다 더 적은 수를 규칙으로도 비슷한 성능을 보인다.

예제 3. 이번 예제는 두 개의 입력(x_1, x_2)과 하나의 출력(y_d)을 가지는 시스템에 대한 모사이다⁵⁾.

$$y_d = \frac{\sin(x_1) \sin(x_2)}{x_1 x_2}$$

여기서 입력의 범위는 $[-10, 10] \times [-10, 10]$ 이고, 입력은 전 입력범위에 걸쳐 균일하게 250개

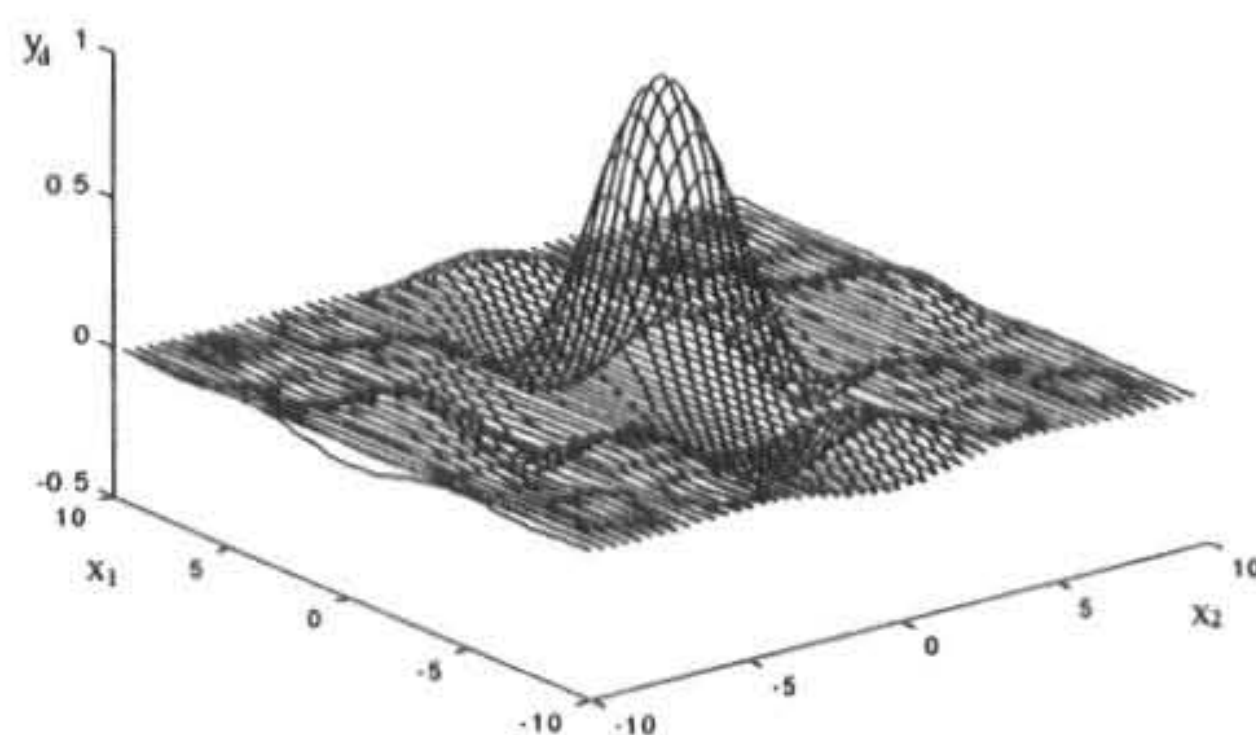


Fig. 7. Desired output(Example 3).

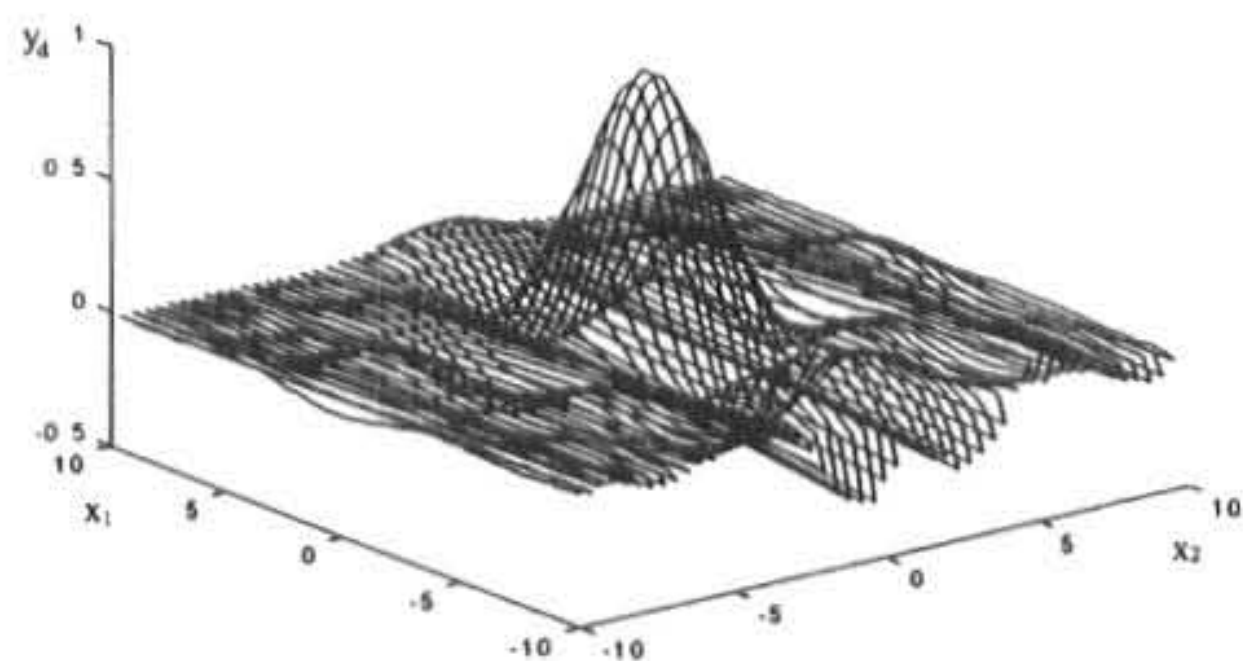


Fig. 8. Identification result of a conventional self-organizing fuzzy system(Example 3).

참고문헌

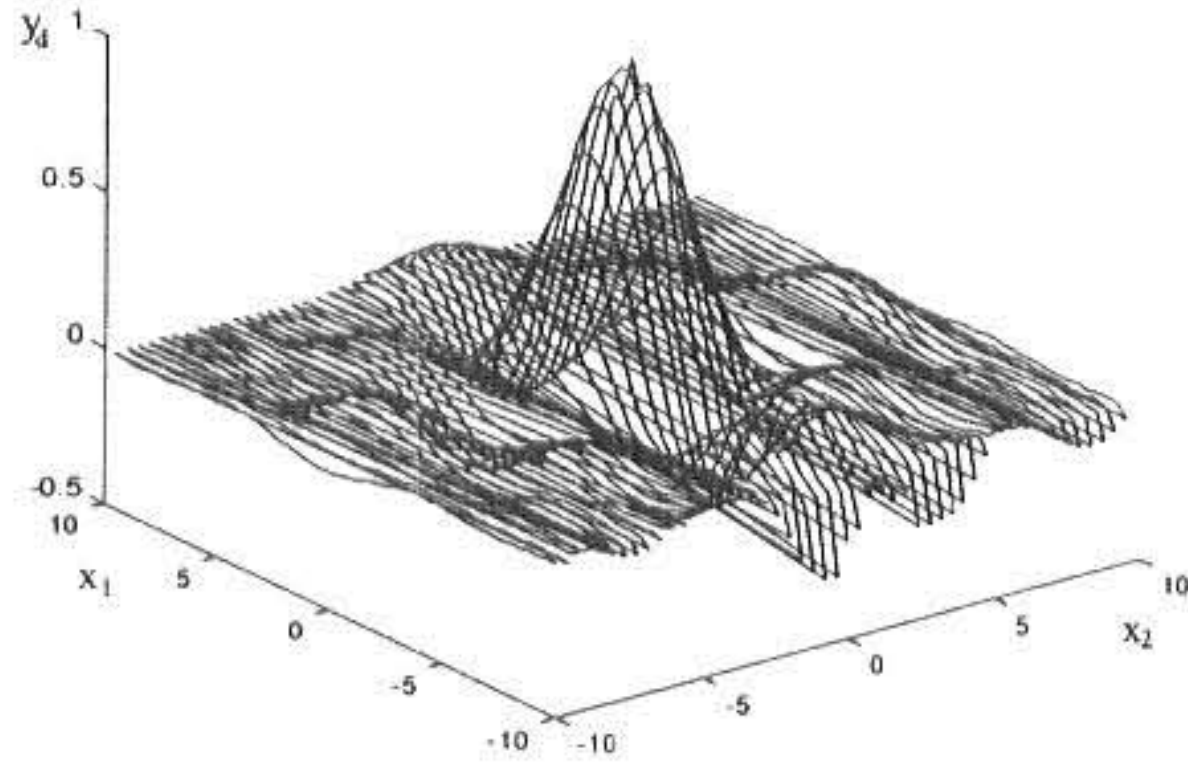


Fig. 9. Identification result of the proposed self-organizing fuzzy system(Example 3).

Fig. 7은 원하는 출력의 형태이고, Fig. 8은 규칙생성만 행한 시스템의 모사결과이고, Fig. 9는 규칙제거를 동시에 한 시스템의 모사결과이다. Table 1은 두 시스템의 최종규칙수와 오차자승합을 비교하였다. 규칙수가 더 작으면서도 작은 오차자승합을 나타내고 있다. 이것은 퍼지 규칙의 수가 단순히 많다고해서 모사가 잘 이루어지는 것이 아니고, 불필요한 규칙을 제거함으로써 더 좋은 결과를 얻을 수도 있음을 보여준다.

Table 1. Comparison of Results(Example 3).

	기존의 자기구성 퍼지시스템	제안한 자기구성 퍼지시스템
최종 규칙수	29	26
오차자승합	1.6978	1.4029

4. 결론

본 논문에서는 규칙을 생성만하는 기존의 일반적인 자기구성 퍼지시스템의 단점을 보완하여 규칙의 생성과 제거를 동시에 행하는 자기구성 퍼지시스템을 제안하였다. 생성과 제거에 있어서 α -cut을 이용하여 전체의 입출력 공간에 적당한 규칙이 존재할 수 있도록 하였다. 제안한 퍼지시스템은 성능면에서는 기존의 자기구성 퍼지시스템과 비슷하면서도 적은 수의 규칙만으로 구성되어 있음을 알 수 있었다.

- 1) T. J. Procyk and E. H. Mamdani : A linguistic Self-Organizing Process Controller. *Automatica*, vol. 15, pp. 15-30, (1979)
- 2) Shihuang Shao : Fuzzy Self-Organizing Controller and Its Application for Dynamic Processes. *Fuzzy Sets and Systems* 26, pp. 151-164, (1988)
- 3) Y. M. Park, U. C. Moon, and K. Y. Lee : A Self-Organizing Fuzzy Logic Controller for Dynamic Systems Using a Fuzzy Auto-Regressive Moving Average(FARMA) Model. *IEEE Trans. Fuzzy Systems*, vol. 3, no. 1, pp. 75-82, (1995)
- 4) Kit-sang Tang, Kim-fung Man, Zhi-feng Lie, and Sam Kwong : Minimal Fuzzy Memberships and Rules Using Hierarchical Genetic Algorithms. *IEEE Trans. Industrial Electronics*, vol. 45, no. 1, pp. 162-169, (1998)
- 5) J. S. R. Jang : ANFIS Adaptive Network Based Fuzzy Inference Systems," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, no. 3, pp. 665-684, (1993)
- 6) K. S. Narendra, and K. Parthasarathy : Identification and control of Dynamical Systems Using Neural Networks. *IEEE Trans. Neural Networks*. vol. 1, no. 1, pp. 4-27, (1990)

(2002년 10월 17일 접수, 2003년 1월 20일 채택)