

비선형 시스템의 동정을 위한 자기 구조화된 RBFN의 구현

The Implementation of Self-Structuring Radial-Basis Function Network for Identification of Uncertain Nonlinear Systems

김기범*, 전재춘**, 김동원*, 허성회*, 박귀태*

*고려대학교 전기공학과

**삼성전자 컴퓨터시스템사업부 Mobile개발그룹

Ki-bum Kim*, Jae-choon Jun**, Dong-won Kim*, Sung-hue Huh* and Gwi-Tae Park*

*Dep. of Electrical Engineering, Korea University

**Mobile R&D Group, Computer System Division, Samsung Electronics Co.

E-mail : kikim@elec.korea.ac.kr

ABSTRACT

본 논문에서는 새로이 제안된 자기 구조화하는(Self-structuring) 새로운 Radial-Basis Function Network(RBFN)에 대해서 실험적인 검증을 했다. 이 자기 구조화하는 새로운 RBFN은 기존의 RBFN과 비교해서 여러 장점이 있다. Lyapunov 이론에 기초해서 새로운 학습 규칙을 선정하였기 때문에 시스템의 안정도를 보장할 수 있다. 그리고, 자기 구조화의 과정 즉, 생성과 병합을 통해 은닉층에서 적정수의 뉴런을 결정할 수 있다. 기존의 RBFN과 성능을 비교하기 위하여, 실제 비선형 시스템인 2축 암로봇에 대해 실험한 결과를 보였다. 결과적으로, 우리는 실험결과를 통해 자기 구조화하는 RBFN의 효율적인 구조와 시스템에 대한 안정도를 보장함을 볼 수 있다.

Key words : RBFN, 자기 구조화, 안정도, 암 로봇, 리아프노프 이론

I. 서론

비선형 시스템을 동정(identification)하는데 있어서 여러 가지 기법들이 있다. 예를 들면, 퍼지 시스템, 신경망, 그리고 RBFN 등이 있다. 최근에는 단층구조로 인해 빠른 수렴속도를 가진 RBFN 에 많은 관심을 가지기 시작하였다. 그러나, 일반적인 RBFN 은 몇가지 기본적인 단점이 있다. 하나는 고정된 구조를 갖는다는 점이다. 경험적인 판단으로 은닉층에서 사용되는 뉴런 수를 결정해야만 한다. 이로 인해서 비효율적인 구조를 갖을 수 있다. 이는 자기구조화 알고리즘을 통해 적절한 구조를 찾아 내어질 수 있다. 다른 하나의 단점은 비선형 시스템을 동정 할 때, 시스템의 안정도라는 의미에서 해석적인 결과를 보장할 수 없다는 것이다. 이는 내부적으로 시스템이 발산 할 수도 있으므로, 기존과는 다른 새로운 학습규칙이 필요하게 된다. 이를 위해서, Lyapunov 이론에 기초를 둔

학습방법을 제안한다. 이 방법은 수학적으로, 시스템의 안정도를 보장할 수 있게 된다. 기존의 RBFN 의 단점들을 극복하기 위해 새로 제안된 자기구조화하는 새로운 RBFN 의 성능을 확인하기 위하여, ARM-7720C 라는 Manipulator 를 이용한 비선형 시스템을 만들어, 직접 적용시켜 본다.

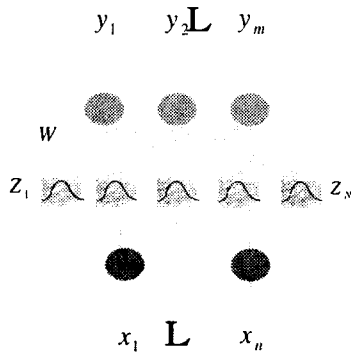
결과적으로, 수학적인 증명뿐만이 아니라, 실제적인 시스템인 Manipulator 에서도 시스템의 안정도를 보장하면서, 적절한 구조를 갖는 네트워크를 시각적으로 확인 할 수 있다. 또한, 기존의 RBFN 보다 더 뛰어난 성능을 확인 할 수 있다.

II. 본론

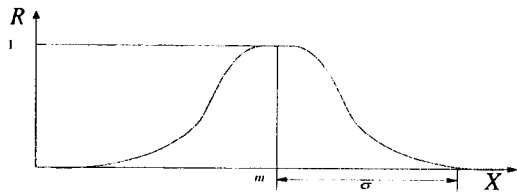
2.1 일반적인 RBFN의 알고리즘

일반적인 RBFN은 Kohonen 신경망의 확장된 형태로 널리 사용되고 있다. 그 구조는 그림1의 (a)

와 같이 입력층, 은닉층, 출력층으로 구성되어 있다.



(a) 일반적인 RBFN의 구조



(b) R.B.(Radial Basis)의 구조

그림1 일반적인 RBFN과 R.B.의 구조

입력 x_1, \dots, x_n 과 은닉층사이의 가중치는 1이고, 은닉층 각 뉴런치의 출력치는 다음과 같이 주어진다.

$$z_q = \frac{R_q(x)}{\sum_k R_k(x)}, \quad (q = 1, 2, \dots, N) \quad (1)$$

$$R_q = \exp\left[-\frac{|\mathbf{x} - \mathbf{m}_q|^2}{2\sigma_q^2}\right]$$

이다.

RBFN의 출력층 뉴런의 결과는 다음과 같다

$$y_i = a_i \left(\sum w_{iq} z_q + \theta_i \right), \quad (i = 1, 2, \dots, m)$$

여기에서, a_i 는 활성화 함수, θ_i 는 문턱치 단순화 하기 위해서 $a_i = 1, \theta_i = 0$ 을 선택했다.

일반적인 RBFN의 학습규칙은 다음과 같다.

$$\Delta w_{iq} = \eta (d_i - y_i) z_q \quad (2)$$

$$\Delta \mathbf{m}_q = \eta \sum_i (d_i - y_i) \frac{\partial y_i}{\partial \mathbf{m}_q} \quad (3)$$

$$\Delta \sigma_q = \eta \sum_i (d_i - y_i) \frac{\partial y_i}{\partial \sigma_q} \quad (4)$$

2.2 자기구조화 알고리즘

2.2.1 뉴런 생성

그림2의 (a)에서와 같이 입력변수인 x_1, \dots, x_n 에 대한 Gaussian 함수의 응답치중 최대값이 설계자가 설정한 기준치보다 작으면, 새로운 뉴런 C를 생성한다. 다시 말해서, 임의의 입력값에 반응하는 뉴런이 없으면, 새로운 뉴런을 생성하는 것이다. 생성시 새

로운 뉴런의 중심값은 그 입력값에 해당하고, 가중치는 0, 폭은 기존의 값으로 설정한다. 뉴런을 생성 후, 원하는 학습규칙으로 학습을 한다.

2.2.2 뉴런 병합

마지막으로 그림2의 (b)와 같이, 가장 유사한 뉴런들을 찾아내어 기준치보다 크면, 뉴런을 병합하게 만든다. 이 때, 유사한 뉴런들을 찾아내기 위해 사용된 개념인 유사도는 다음과 같이 정의 한다.

$$E(A, B) = \frac{M(A \cap B)}{M(A \cup B)} \quad (5)$$

여기에서,

$$M(A, B) = \sum_{i=1}^n |\mu_A(u_i) - \mu_B(u_i)| \quad (6)$$

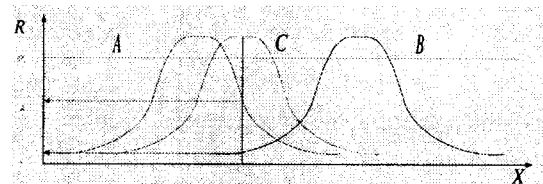
μ_A, μ_B 들은 각각 A, B의 Membership 함수를 의미한다. 즉, 유사도는 완전히 일치할 경우 1을, 전혀 다를 경우 0을 나타내게 된다.

계산 속도를 빠르게 하기 위하여 R.B.함수를 삼각형 모양의 함수를 이용하면, 다음과 같이 근사화한다.

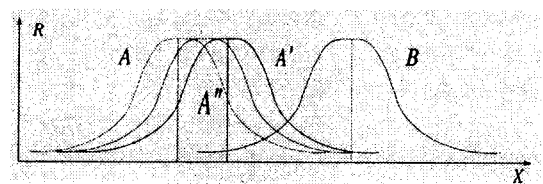
$$E(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{\sigma_1 \sqrt{\pi} + \sigma_2 \sqrt{\pi} - |A \cap B|} \quad (7)$$

그림2 (b)에서 병합된 뉴런 A'' 의 중심값과 폭은 각각 병합할 뉴런들의 중심값들의 평균과 폭들의 평균이고, 가중치는 다음과 같이 설정한다.

$$w_{annexation} = \frac{w_i z_i + w_j z_j}{z_{new}} \quad (8)$$



(a) 뉴런의 생성



(b) 뉴런의 병합

그림2 뉴런의 생성과 병합

2.3 자기구조화 RBFN의 안정도 해석

일반적인 비선형 계통은 다음과 같이 나타낸다.

$$\begin{aligned} \dot{\mathbf{x}} &= f(\mathbf{x}, \mathbf{u}) \\ &= -\mathbf{A}\mathbf{x} + \{ \mathbf{A}\mathbf{x} + f(\mathbf{x}, \mathbf{u}) \} \\ &= -\mathbf{A}\mathbf{x} + F(\mathbf{x}, \mathbf{u}) \end{aligned} \quad (3)$$

여기서, A는 정칙 행렬이고, $\mathbf{u} \in R^m$ 는 시스템의 입력이다.

자기 구조화 RBFN을 이용하면, (1)의 비선형 계통을 다음과 같이 나타낼 수 있다.

$$\dot{x} = f(x, u) = -Ax + \theta^T \xi(x, u; c^*, \delta^*) + \varepsilon \quad (4)$$

여기서 $A > 0$, $\xi \in [\xi_1 \xi_2 L \xi_{N(u)}]^T$ 는 은닉층 출력벡터, $c = [m_1^T m_2^T L m_{N(u)}^T]^T$, $\delta = [\sigma_1^T \sigma_2^T L \sigma_{N(u)}^T]^T$ 는 중심값과 폭을 나타내는 벡터이다. $\theta \in R^{N(u) \times n}$ 는 가중치이고, ε 는 근사화 오차를 나타내며, θ^* 는 최적 RBFN 파라미터이고, 다음과 같다.

$$\theta^* = \arg \min_{(\theta, u) \in \Omega} \left[\sup_{x \in \Omega} \hat{F}(x, u) - F(x, u) \right] \quad (5)$$

정리1: (3)의 비선형 계통에 대하여, 다음의 파라미터 갱신 규칙을 가지는 자기구조 RBFN은 모든 내부 계통 파라미터와 신호들의 유계를 보장한다.

$$\hat{\theta} = \hat{\theta} + r_\theta [\xi \hat{\phi} e^T P - \hat{\xi} e^T P] - \kappa r_\theta |e| \hat{\theta} \quad (6)$$

$$\hat{\phi} = \hat{\phi} + r_\phi [-e^T P \hat{\theta}^T \xi + \hat{\phi} e] - \kappa r_\phi |e| \hat{\phi} \quad (7)$$

여기에서, $e = \hat{x} - x$, $\hat{\theta} = \hat{\theta} - \theta^*$, $\hat{\phi} = \hat{\phi} - \phi^*$, $\hat{c} = \hat{c} - c^*$, $\hat{\delta} = \hat{\delta} - \delta^*$, $\phi = [c^T \delta^T]^T$

증명: 다음과 같은 리아프노프 후보함수를 정의한다.

$$V = \frac{1}{2} e^T P e + tr \left[\frac{1}{2r_\theta} \hat{\theta} \hat{\theta}^T \right] + \frac{1}{2r_\phi} \hat{\phi} \hat{\phi}^T \quad (8)$$

여기에서 $P > 0$, $r_\theta > 0$ 그리고, $r_\phi > 0$ 인 임의의 값이다.

(6), (7)의 적응규칙을 이용하면, V의 시간도함수가 음이 된다. 그러므로, 상태추정오차 e는 원하는 정도로 유계 할 수 있으며, θ, ϕ 는 유계임이 보장된다.

III. 실험 결과

3.1 시스템의 구성

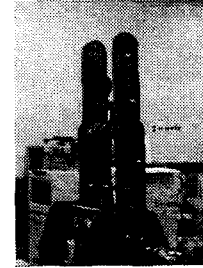
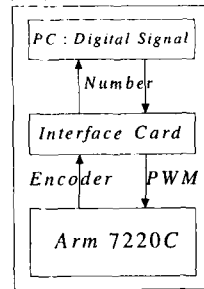
Arm 7220C에서 2축만을 사용하며, PC와 7220C와 통신하기 위해 Interface Card를 제작하였고, 사용자와 정보를 교환하기 위해 시스템의 GUI를 구성하였다. Arm 7200C 각축의 특징은 표1과 같다.

표1 Arm 7220C의 각축의 특징 설명

	1축	2축
Motor	24V/4.6W	24V/4.6W
Encoder	200pulse/회	200pulse/회
Gear	144:1	144:1
작동범위	130°	345°
기 타	Limit 스위치	

그림3 (a)에서 원하는 임의의 입력 수치를 입력하면, Interface Card를 통해 증폭된 PWM신호로 변환한다. 또한, Arm 7220C의 Encoder신호를

Interface Card통해 정확한 위치와 속도를 측정하며, 과전류를 차단한다.



(a) 전체 Block 선도 (b) Arm 7220C
그림3 전체 시스템 구성

그림4는 제안한 RBFN의 모든 정보를 시각적으로 보여주는 시스템의 GUI이다. 여기서, 각 뉴런의 가중치, 중심값, 폭과 그리고 오차를 확인 할 수 있다. Section-1은 하나의 입력에 대한 은닉층의 모든 뉴런을 보여주며, Section-2에서 가변하는 전체적인 구조를, Section-3에서는 암로봇의 각축의 위치와 속도, 그리고, RBFN의 추정치를 도시한다. 마지막으로, Section-4에서는 은닉층에 있는 하나의 뉴런에 대한 모든 입력층에서의 R.B. 함수를 볼 수 있다.

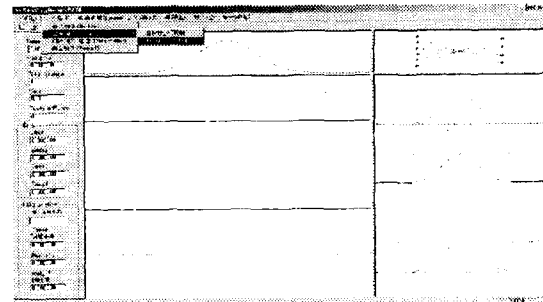
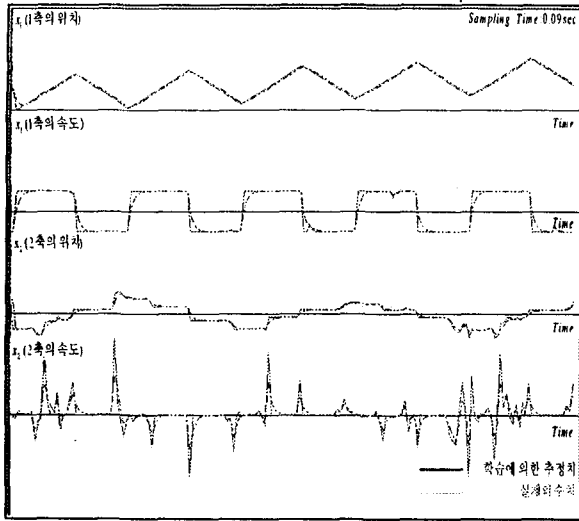


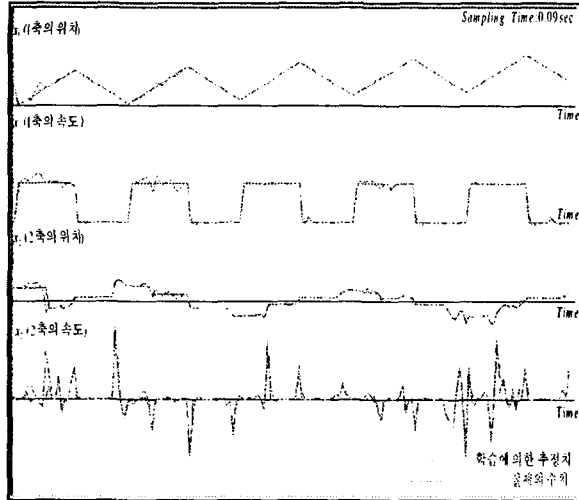
그림4 시스템의 GUI

3.2 실험 결과

그림6(a)와 같이 은닉층의 뉴런 개수를 임의로 20개로 고정시키고, 일반적인 RBFN의 학습규칙으로 2160회 학습을 하였을 경우, 전체 오차는 0.00481이고, 그림 5(a)에서 전체적인 추정성능을 볼 수 있다. 일반적인 RBFN은 적절한 파라미터를 설정할 경우 좋은 성능과 학습속도를 나타내지만, 안정도를 보장하지 않았기 때문에 가장 폭이 큰 R.B.를 선택 했을 때, 폭과 중심값이 표2에서와 같이 발산함을 알 수 있다. 자기구조화 알고리즘으로 같은 학습횟수로 학습하였을 때, 그림6(b)에서는 최종 뉴런 수가 11개임을, (c)에서는 변화과정을 보여 주고 있다. 전체 오차는 0.00281이다, 그림 5(b)에서 알 수 있듯이 일반적인 RBFN보다도 더욱 높은 성능을 나타냄을 알 수 있다. 또한, 안정도를 보장하는 새로운 알고리즘을 사용하였기 때문에 표2에서 보이듯이 가장 폭이 큰 R.B. 함수가 3.6에서 더 이상 증가하지 않음을 알 수 있다.

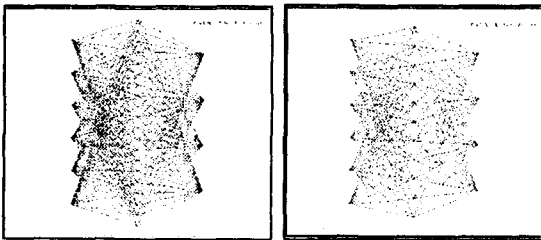


(a) 일반적인 RBFN



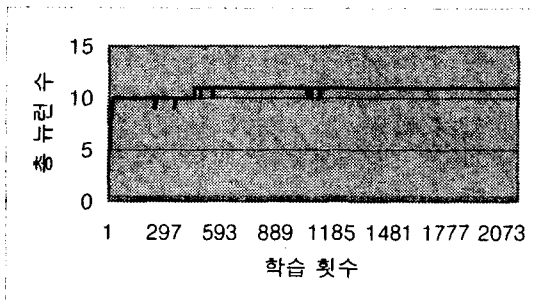
(b) 자기구조 RBFN

그림5 각축의 위치, 속도의 오차 비교(Section 3)



(a) 일반적인 RBFN

(b) 새로운 RBFN



(c) 자기구조 RBFN의 총 뉴런수 변화 과정

그림6 RBFN의 구조 비교(Section 2)

표2 일반적인 RBFN과 자기구조 RBFN의 파라미터

	일반적인 RBFN	자기구조 RBFN
순번	20	1
중심값	-369.7728	-1.3690
폭	7571940361	3.5056
가중치	3.8986	3.3425

IV. 결론

본 논문에서는 기존 RBFN의 가지고 장점인 빠른 계산 속도로 인하여, 실제의 시스템에서 온라인 학습이 가능함을 알 수 있었다. 그러나, 단점으로 비효율적인 구성과 내부 파라미터들의 발산을 들 수 있다. 기존의 RBFN의 단점들을 극복하기 위해 제시된 자기구조 RBFN 알고리즘을 통해 최적의 구조를 구성함을 보였고, 안정도까지 고려하였기 때문에 모든 뉴런의 가중치, 중심값, 폭들이 일정한 경계 내에서 수렴함을 실제 실험을 통해 확인하였다. 즉, 기존의 RBFN의 단점은 극복하면서, 장점을 그대로 수용한 자기구조 RBFN을 실제의 시스템에 적용하여 알고리즘의 성능을 증명하였다. 앞으로의 과제는 자기구조 RBFN을 이용한 제어기 설계가 필요하다.

V. 참고문헌

- [1] 전재춘 "자기 구조화 RBFN을 이용한 불확실한 비선형 계통의 동정", 고려대학교 학위논문
- [2] E.J. Hartman, J.D. Keeler, and J.M. Kowalski, "Layered neural networks with Gaussian hidden units as universal approximations," *Neural Comput*, vol.2, pp. 210-215, 1990.
- [3] J. Nie, and D.A. Linkens, "Learning control using fuzzified self-organizing radial-basis function network," *IEEE Trans. On Fuzzy syst.*, vol.1, No.4, pp. 280-287, 1993.
- [4] C.T. Lin, and C.S.G. Lee, "Real-time supervised structure /parameter learning for fuzzy neural network," *Proc. IEEE Int. Conf. Fuzzy syst.*, San Diego, pp. 1283-1291.
- [5] T.K. Kohonen. "Self-Organization and Associative Memory", 3rded. New York, Springer-Verlag, 1977.