

자율분산 신경회로망을 이용한 간접 적응제어

최종수* · 김형식** · 김성중** · 권오신***

- * 포스콘 기술연구소 분산제어팀
- ** 전북대학교 제어계측공학과
- *** 군산대학교 제어계측공학과

Indirect Adaptive Control Based on Self-Organized Distributed Network(SODN)

J. S. Choi* · H. S. Kim** · S. J. Kim** · O. S. Kwon***

* DCS Team, R&D Center, POSCON

** Dept. of Control & Instrumentation Eng., Chonbuk Nat'l University

*** Dept. of Control & Instrumentation Eng., Kunsan Nat'l University

Abstract - The objective of this paper is to control a nonlinear dynamical systems based on Self-Organized Distributed Networks (SODN). The learning with the SODN is fast and precise. Such properties are caused from the local learning mechanism. Each local network learns only data in a subregion. Methods for indirect adaptive control of nonlinear systems using the SODN is presented. Through extensive simulation, the SODN is shown to be effective for adaptive control of nonlinear dynamic systems.

1. 서론

비선형 시스템의 식별 및 제어에 이용되는 신경망의 대부분을 차지하는 모델은 오차 역전달 학습법을 이용한 다층 신경망(Multilayer Neural Networks, 이하 MNN)이다. Narendra등[1]은 비선형 시스템 식별을 미지의 비선형 시스템 식별과 제어 문제에 일반화된 MNN과 선형 필터가 직렬 및 피드백 구조로 구성되는 모델을 적용하였고, 식별 모델로써 직렬식별 및 병렬식별 모델구조를 도입하였다. Qin 등은 MNN과 순환신경망(recurrent neural networks)을 비선형 시스템 식별에 적용하였고 개별학습(pattern learning) 및 일괄학습(batch learning)의 특성을 해석하였다[2]. 또한 Sastry 등은 메모리 뉴턴 신경망(Memory Neuron Network)이라는 내부 메모리를 갖는 순환신경망 모델을 제안하여 비선형 시스템 식별 및 제어에 적용하였다[3]. 이와 같이 MNN과 같은 완전연결 신경망(fully connected neural networks)은 전체적 일반화 능력이 우수하기 때문에 비선형 시스템 식별 및 제어에 가장 많이 사용된다. 그러나, 완전연결 신경망은 전체적 근사화 특성이 좋은 반면에 지역적인 변동을 갖는 비선형성이 매우 심한 시스템을 학습하는 것이 매우 어렵다. 이런 경우 만족할만한 일반화 특성을 얻기 위해서 충분히 많은 훈련 데이터를 필요로 하게 되고, 특성이 현저하게 다른 영역에 한정된 연결강도로 학습하기 때문에 일시적인 혼란(temporal crosstalk)을 일으키게 되어 특성이 서로 다른 영역 모두를 만족시키는 연결강도 값을 찾아야 하는 어려움에 봉착한다. 다행히도 혼란 영역이 서로의 경계를 넘나든다면 전체적인 일반화 특성을 얻을 수 있지만 학습시간이 길어지고 지역적 일반화 특성이 떨어지게 된다. 일시적 혼란 문제는 복잡한 비선형 시스템 식별과 제어 문제에서 두드러진다. 시계열 예측과 같은 시스템 식별 문제는 인접한 영역간의 비선형성이 매우 심해서 학습이 어렵

다. 또한 제어 분야에서 비선형 동적 시스템은 상태공간을 통해서 상대적으로 천천히 이동하는 경향이 있기 때문에 훈련 데이터는 주로 구역화된 영역으로 표현되며, 플랜트의 동특성이 동작점에 따라서 변화한다면 일시적인 혼란현상은 피할 수가 없게 된다.

저자들은 전체연결 신경망 및 경쟁 신경망의 장점을 결합한 자율분산 신경망(Self-Organized Distributed Networks, 이하 SODN)을 제안하였다[5, 6]. SODN은 입력 공간을 학습 데이터의 분포와 학습의 어려운 정도에 따라 자율적으로 분할하고 각각 분할된 공간에 부분 신경망을 분산 배치하여 지역적 학습을 수행하게되므로 학습 속도가 빠르다. 또한 SODN은 학습 대상을 정확히 학습하는 지역연결 신경망의 구조를 갖고 있으며 일반화 능력이 우수한 완전연결 신경망의 장점을 갖춘 신경망이다. 이 연구에서는 이러한 SODN의 특성을 이용하여 비선형 동적 시스템의 간접 적응제어에 적용하고 시뮬레이션을 통해서 SODN이 비선형 시스템 제어기로서 유용함을 입증한다.

2. 자율분산 신경회로망(SODN)

2.1 SODN의 구조

SODN의 구조는 그림 1과 같으며 학습 신경망(Learning Networks)과 부분망의 분산 배치용인 분산 신경망(Distribution Networks)으로 구성된다. SODN은 학습 신경망내의 MNN 한 세트와 분산 신경망내의 유니트 한 개는 짝이 되어 한 세트의 부분 신경망을 구성하며 분할된 공간의 함수 학습을 수행한다. 입력 벡터는 학습 신경망과 분산 신경망에 공통으로 연결되고 학습 신경망과 분산 신경망의 출력들은 상호 짝이 되는 신경망의 출력과 곱하여 연결된다. SODN에 입력 벡터가 인가되면 분산 신경망 내의 유니트 중 입력 벡터와 가장 유사한 연결강도 배터를 갖는 유니트만 활성화되며, 이 유니트는 짝을 이루는 학습 신경망의 출력과 선형결합을 통해서 전체 신경망의 출력이 된다.

학습 신경망은 두세 개의 은닉층 유니트를 갖는 소규모의 MNN과 제한망(LN)이 결합된 집합으로 구성되어 있다. 시간 k 에서 MNN의 $l(1 \leq l \leq L)$ 층에 있는 j 번째 유니트의 출력을 $x_j^l(k)$ 라 하면 출력은

$$\begin{aligned} x_j^l(k) &= \Psi\left(\sum_i w_{ji}^l(k) x_i^{l-1}(k) + b_j^l(k)\right) \\ &= \Psi(\text{net}_j^l(k)) \end{aligned} \quad (1)$$

로 계산된다. 여기서 $w_{ij}^l(k)$ 는 $l-1$ 층의 i 번째 유니트와 l 층의 j 번째 유니트 간의 연결강도이고, $b_j(k)$ 는 j 번째 유니트의 바이어스이다. 또한 $net_j(k)$ 는 입력 값의 총합이고, $\Psi(\cdot)$ 는 시그모이드 함수로 표현되는 유니트의 활성화 함수이다.

$$\Psi(net) = \frac{1}{1 + e^{-net}} \quad (2)$$

각 부분 학습 신경망의 출력은 MNN의 출력에 LN이 직렬로 연결되어 있어서 목표 값의 동작범위로 제한을 받는다. LN은 MNN 출력단에 연결되어 있고 환산 연결강도(scaling weight)와 바이어스를 갖는 선형 퍼셉트론으로 구성되어 있다. 환산 연결강도는 목표 값의 동작범위이고 퍼셉트론의 활성화 함수는 램프함수이며, 바이어스는 목표 값의 최소치이다. $m(1 \leq m \leq M, \text{여기에서 } M \text{은 부분 신경망의 수})$ 번째 학습 신경망에서 MNN의 $n(1 \leq n \leq N, \text{여기에서 } N \text{은 출력 유니트의 수})$ 번째 출력을 $x_n^l(k)$ 이라 하면, 학습 신경망의 출력 $y_{nm}(k)$ 는 다음과 같다.

$$y_{nm}(k) = f(x_n^l(k) s_m(k) + b_m(k)) = x_n^l(k) s_m(k) + b_m(k) \quad (3)$$

$s_m(k)$ 는 환산 연결강도로서 동작범위의 최대 값과 최소 값의 차이이고, 바이어스 $b_m(k)$ 는 동작범위의 최소 값이므로 식 (3)은 다음과 같다.

$$y_{nm}(k) = x_n^l(k) (t_{\max}(k) - t_{\min}(k)) + t_{\min}(k) \quad (4)$$

여기에서, $x_n^l(k)$ 의 범위는 $0.0 \leq x_n^l(k) \leq 1.0$ 이다. 그러므로 $y_{nm}(k)$ 의 범위는 $t_{\min}(k) \leq y_{nm}(k) \leq t_{\max}(k)$ 로 제한된다. 환산 연결강도와 바이어스 값을 매 학습주기마다 선택된 부분 신경망 영역에서 학습데이터를 통해서 갱신된다.

한편, 분산 신경망은 0 또는 1을 출력하는 경쟁망으로서 각 유니트 내에 신경망의 생성/소멸에 관련된 에너지 저장소를 갖고 있다. 분산 신경망은 단층 신경망으로 구성되며 각 유니트는 입력단으로부터 연결강도를 통하여 연결된다. 즉, fan-out 유니트로부터 입력신호를 받아들이는 M 개의 자율신경 유니트로 구성된다. 입력 벡터에 대하여 m 번째 자율신경 유니트의 중심위치(연결강도)를 $c_m(k)$ 라 하면, 각 유니트는 다음 식과 같이 입력강도 $I_m(k)$ 를 계산한다.

$$I_m(k) = D(c_m(k), \mathbf{x}(k)) \quad (5)$$

여기에서 $D(\cdot)$ 는 유클리드 거리 함수이며, 잡음은 고려하지 않았다. 분산 신경망의 입력강도 $I_m(k)$ 가 계산되면 가장 작은 연결강도를 갖는 유니트를 찾기 위해 서로 경쟁을 한다. 경쟁이 끝나고 Winner 유니트가 결정되면 Winner 유니트의 출력 $u_m(k)$ 만 1이 되어 활성화한다.

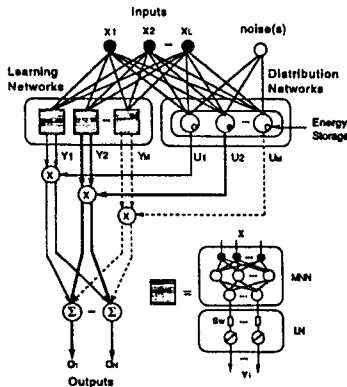


그림 1. SODN의 구조.

j 번째 출력 유니트에 대한 입력의 총합을 $net_j(k)$ 라 하면

$$net_j(k) = \sum_{m=1}^M y_{jm}(k) u_m(k) \quad (6)$$

이다. SODN에 대한 j 번째 출력 유니트의 최종 출력 $o_j(k)$ 는

$$o_j(k) = \Psi(net_j(k)) = \Psi\left(\sum_{m=1}^M y_{jm}(k) u_m(k)\right) \quad (7)$$

가 된다. 여기서 y 는 학습 신경망의 출력이고 Ψ 는 출력단의 활성화 함수이다.

2.2 SODN의 학습

신경망 학습의 목표는 비용함수를 최소화하는 연결강도 벡터를 찾는 것이다. 이 연구에서는 일괄학습을 사용하여 SODN을 학습시킨다. SODN은 학습 신경망의 감독학습과 분산 신경망의 비감독학습의 복합학습 구조로 이루어져 있다.

주어진 동적 시스템의 훈련 데이터 $\{I(k), \mathbf{x}(k)\}, k=1, 2, \dots, T$ 에 대하여 일괄학습을 적용할 경우 비용함수는 다음과 같은 제곱 오차함수로 나타낼 수 있다.

$$E = \frac{1}{2} \sum_{k=1}^T \sum_{j=1}^J (t_j(k) - o_j(k))^2 \quad (8)$$

여기에서 $t_j(k)$ 는 동적 시스템의 목표출력이고 $o_j(k)$ 는 SODN의 출력이다.

학습 신경망은 감독학습으로써 오차 역전파 학습 알고리즘을 사용하여 학습하는데, 역전파 학습은 합자승 오차 함수의 최대 경사면 방향으로 연결강도를 점차 변화시켜가는 학습 원리이므로 입력 패턴에 대해 연결강도의 변화는 다음과 같이 계산된다.

$$w_{ij}(k+1) = w_{ij}(k) - \eta \frac{\partial E(k)}{\partial w_{ij}(k)} = w_{ij}(k) + \eta \delta_j(k) x_i(k) \quad (9)$$

여기에서 η 는 학습률이다. 출력단 또는 은닉층에서 j 번째 유니트로부터 역전파될 오차 벡터 $\delta_j(k)$ 는

$$\delta_j(k) = \begin{cases} (t_j(k) - o_j(k)) \Psi'(net_j(k)), & \text{출력단에 대하여} \\ \Psi'(net_j(k)) \sum_k \delta_k(k) w_{jk}(k), & \text{은닉층에 대하여} \end{cases} \quad (10)$$

이다. j 번째 출력 유니트로부터 역전파될 오차벡터 $\delta_j(k)$ 를 계산하기 위해서 식 (7)과 (10)을 이용하면

$$\delta_j(k) = error_j(k) o_j(k) \Psi'(net_j(k)) = (t_j(k) - o_j(k)) \Psi'(net_j(k)) \quad (11)$$

이다. 또 모든 출력 유니트는 선형 결합기로서 그 활성화 함수는 램프함수이므로 미분값 Ψ' 는 1이다. 따라서 식 (11)은

$$\delta_j(k) = error_j(k) = t_j(k) - o_j(k) \quad (12)$$

가 된다. 또 각 학습 신경망에 역전파될 오차 성분은 등가적 연결강도 벡터 \mathbf{u} 를 따라 전파시켜 얻을 수 있으며, 이 때 i 번째 부분 신경망에 역전파될 오차 성분은 식 (10)과 (12)에 의해서

$$\delta_i(k) = \Psi' \sum_{j=1}^J \delta_j(k) u_j(k) = u_i(k) \Psi' \sum_{j=1}^J (t_j(k) - o_j(k)) \quad (13)$$

가 된다. 여기에서의 $u_i(k)$ 는 분산 신경망의 출력벡터 \mathbf{u} 중의 i 번째 요소로서 분산 신경망 중의 i 번째 유니트가 선택된 경우에만 값이 1이고 나머지 경우에는 0인 binary 값이다. 따라서 최종 출력에서 발생한 오차는 분산 신경망에 의해 선택된 부분 신경망에만 전파되어 연결강도 벡터가 학습한다.

다음, 분산 신경망은 비감독 학습으로써 자율신경 유니트 사이의 억제적 연결을 이용하여 입력 벡터가 인가되면 입력벡터와 가장 유사한 연결강도를 갖는 유니트만 활성화되는 "Winner-Take-All (WTA)" 규칙을 이용한다. 분산 신경망내의 i 번째 유니트 위치를 c_i 라고 하면, 입력벡터 \mathbf{x} 가 인가될 때의 유니트 위치이동은 다음과

같은 학습규칙을 따른다.

$$c_i(k+1) = \begin{cases} (1-\alpha)c_i(k) + \alpha x_i, & \text{for winning unit} \\ c_i(k), & \text{otherwise} \end{cases} \quad (14)$$

여기에서 $\alpha(0 < \alpha < 1)$ 는 학습률로서 큰 초기값으로 부터 점차 작은 최종값으로 학습 스텝(또는 반복횟수)에 따라 점차 줄어드는 함수로 선정한다면 초기에 시스템은 정확한 유니트의 위치를 빠르게 학습할 수 있다. 이 학습 규칙은 활성화된 유니트의 연결강도 벡터만 입력 벡터 방향으로 이동하게 되므로 학습한 후 유니트들은 현재의 위치에서 가장 가까운 데이터군의 중심으로 이동하게 된다.

3. SODN을 이용한 간접 적응제어

간접 적응제어에서는 두 개의 신경회로망 모델이 필요하다. 하나는 식별기(Identifier 또는 emulator)라 부르는 식별 모델로서 플랜트를 식별하기 위한 모델이고, 다른 하나는 제어기로 사용하기 위한 모델이다. 간접 적응제어는 식별을 통해서 제어 대상 플랜트를 모델링하고 제어기를 학습시키기 위해서 이 식별 모델의 파라미터를 이용한다. 따라서 간접 적응제어의 성능은 주어진 식별된 모델로부터 제어 법칙을 유도하기 때문에 식별기의 정확성에 달려있다. 신경회로망을 이용한 간접 적응제어에서의 어려움은 플랜트를 모사하는 식별기가 파라미터화되지 않은 식별기이기 때문이다. 따라서 식별된 모델 신경회로망의 연결강도와 미지의 플랜트 파라미터 사이의 관계가 간단하지가 않다.

제어기 신경회로망을 훈련시키는 일반적인 방법이 있는데 이 방법은 위에 언급한 식별 모델을 이용하는 방법이다. 즉, 모델을 미분하는 방법 또는 앞먹임 및 역 모델링이라 부르는 이 방법은 제어기 신경회로망을 훈련하기 위해서 식별된 모델의 파라미터를 이용하지 않고 식별된 모델을 이용하여 제어기 신경회로망을 훈련할 수 있는 신호를 얻는 것이다. 그림 2에 식별기와 제어기로 SODN을 사용한 경우의 간접 적응제어 시스템을 나타냈다. 플랜트의 앞먹임 모델 또는 식별기 모델(그림 2에서 'SODN Identifier')은 플랜트의 출력으로 부터 플랜트 입력까지 오차($e_c(k+1)$)를 역전파하는 통로로 사용된다. 이 역전파된 오차는 제어기 신경회로망(그림 2에서 'SODN Controller')를 훈련시키기 위해서 이용된다. 그러므로 제어기와 식별기 신경회로망은 직접로 연결되어 있는 것으로 생각할 수 있다.

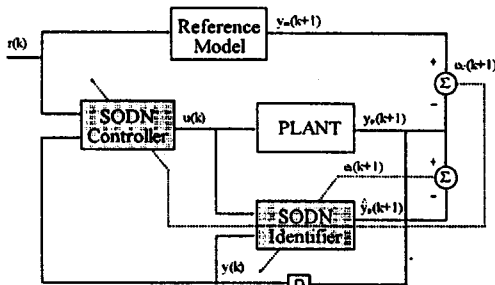


그림 2. SODN을 이용한 간접 적응제어 시스템.

그림 2는 SISO 플랜트를 고려하고 있으므로 두 개의 입력과 한 개의 출력을 갖는 SODN 모델을 구성한다. 시간 k에서 $r(k)$ 는 기준입력이고, $u(k)$ 는 제어기로 사용되는 SODN 제어기의 출력(또는 플랜트의 입력)이다. $y_p(k)$ 와 $\hat{y}_p(k)$ 는 각각 플랜트의 출력 및 SODN 식별기의 출력이다. 또한 $y_m(k)$ 는 기준모델의 출력이다.

제어기를 훈련시키는 과정은 다음과 같이 세 단계로 구분할 수 있다.

1단계 : 오프라인 플랜트 식별

플랜트를 모델링하기 위해 SODN 식별기를 오프라인으로 훈련시킨다.

2단계 : 오프라인 제어기 훈련

기준모델을 모델링하기 위해 SODN 제어기와 SODN 식별기를 직접로 연결하여 제어기를 오프라인으로 훈련시킨다.

3단계 : 온라인 훈련

플랜트 출력이 기준모델의 출력을 추종하도록 온라인으로 식별기 및 제어기를 훈련시킨다.

첫 번째 단계인 오프라인 플랜트 식별과정은 플랜트를 모사하기 위해 SODN 식별기를 학습하는 과정이다. 식별기를 훈련하기 위해서 기준신호 $r(k)$ 는 불규칙 입력으로 공급된다. 식별기가 플랜트를 근사하게 모사할 수 있도록 오프라인으로 훈련한다.

즉, 식별오차 $e_p(k)$ 가

$$e_p(k) = y_p(k) - \hat{y}_p(k) < \epsilon \quad (15)$$

이 될때까지 오프라인으로 훈련시킨다. 다음, 식 (15)가 만족되도록 충분히 학습이 진행되면 기준모델을 모사하기 위해 SODN 제어기와 SODN 식별기를 직접로 연결하여 훈련을 시작한다. 이때 플랜트는 훈련과정에서 제외된다. 기준모델의 출력과 식별기 출력의 오차가

$$y_m(k) - \hat{y}_p(k) < \epsilon \quad (16)$$

이 될때까지 식별기는 고정해 놓고 제어기만 계속 훈련된다. 이 과정에서 기준모델과 식별기 출력 사이의 오차가 만족할 정도로 줄어들긴 하지만 플랜트와 식별기 사이에는 어느정도의 오차가 존재하기 때문에

$$\lim_{k \rightarrow \infty} (y_m(k) - \hat{y}_p(k)) = 0 \quad (17)$$

을 만족할 수는 없다. 신경회로망 제어기는 오랜 시간동안 학습한 후에 제어기로 사용할 수 있기 때문에 훈련을 시작하면서 플랜트를 제어할 수는 없다. 따라서 더욱 정밀한 제어기를 구성하기 위해서 온라인으로 제어기를 훈련해야 하는데 이 과정에서 오프라인으로 훈련한 제어기는 초기의 제어과정에 유용하게 사용될 수 있다. 마지막으로 세 번째 단계는 더욱 정밀한 제어기를 구성하기 위해 온라인으로 식별기와 제어기를 훈련시키는 과정이다. 즉,

$$\lim_{k \rightarrow \infty} (e_f(k)) = 0, \quad (18)$$

$$\lim_{k \rightarrow \infty} (e_c(k)) = 0$$

가 되도록 제어기와 식별기를 충분히 온라인으로 훈련시킨다.

4. 시뮬레이션 및 검토

간접 적응제어를 위해서 고려한 플랜트는

$$y_p(k+1) = \frac{y_p(k)}{1 + y_p^2(k)} + u^3(k) \quad (19)$$

이고, 기준 모델은 다음 식과 같다.

$$y_m(k+1) = 0.6y_m(k) + r(k) \quad (20)$$

제어기 및 식별기로 사용한 SODN은 각각 20세트의 부분 신경회로망으로 구성된다. 학습을 위한 파라미터는 학습 신경회로망의 학습계수 및 모멘텀 계수는 각각 0.1 및 0.9이고 분산 신경회로망의 학습계수는 0.9이다. 학습 파라미터 또한 제어기 및 식별기에 대하여 동일하다. 제어기 및 식별기의 훈련은 위에 설명한 세 단계를 통해서 이루어진다. 그림 3은 제어하지 않은 상태의 응답으로 $u(k)=r(k)$ 일 경우의 기준모델의 출력, 플랜트의 출력 및 기준입력 $r(k)$ 를 나타냈다. 이 때의 기준입력은

$$r(k) = \sin(2\pi k/25) + \sin(2\pi k/10) \quad (21)$$

이다. 식별기 신경회로망을 오프라인으로 훈련한 결과는 그림 4에 나타났다. 결과는 SODN 식별기가 플랜트를 훌륭하게 모사하

고 있다. 오프라인으로 훈련한 식별기를 가지고 SODN 제어기를 오프라인으로 적당히 훈련한 다음 다시 온라인으로 훈련한 후 테스트한 결과는 그림 5에 제시하였다. 결과로부터 알 수 있듯이 SODN을 기반으로한 간접 적응제어 방식은 직접 제어에서의 마찬가지로 기준모델의 출력력을 잘 추종할수 있음이 입증되었다.

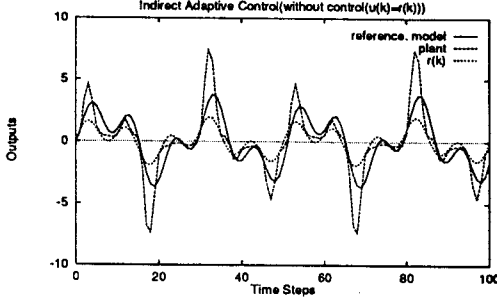


그림 3. 제어하지 않은 경우의 시스템 출력.

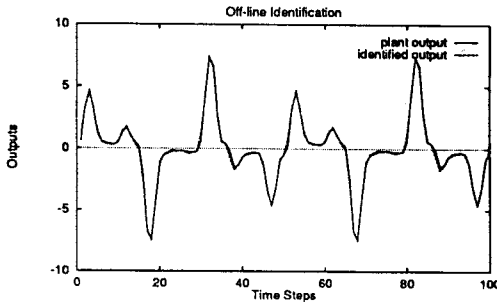


그림 4. 오프라인 식별 결과.

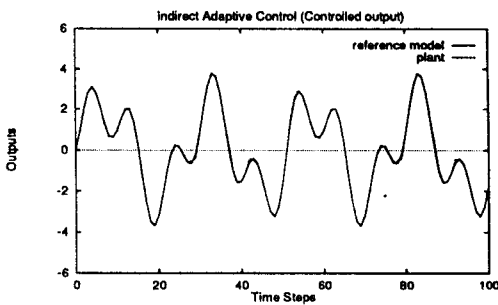


그림 5. SODN 간접 제어기로 제어되는 경우의 시스템 출력.

5. 결 론

신경망은 비선형 동적 시스템을 모델링 및 제어에 유용한 일반적인 기법이다. MNN과 같은 일반적인 모델은 가장 널리 사용하는 모델이지만 학습이 어렵고 오차 평면이 매우 복잡하다는 사실은 널리 알려져 있다. 이것은 MNN의 심각한 단점이며 복잡한 시스템의 식별과 같은 문제에서 잠재적인 뜻이기도 하다. 그러나, 복잡한 문제를 다수의 단순한 문제로 분할 학습하는 SODN은 입력공간을 분할하여

지역적 학습을 수행하므로 학습 속도가 빠르며, 다른 지역연결 신경망의 단점인 전체적 일반화 능력이 개선된 신경망으로 비선형 동적 시스템 식별에 적합한 모델이다. 이 연구에서는 SODN을 이용하여 비선형 동적 시스템 제어를 위한 간접 적응제어기를 구성하였다. 간접 적응제어기는 시스템 식별기와 제어기로 구성된다. 시뮬레이션을 통해서 제안한 SODN은 비선형 동적 시스템 식별 및 제어에 유용함을 입증하였다.

참 고 문 헌

- [1] K. S. Narendra and K. Parthasarathy, " Identification and control of dynamical systems using neural networks," *IEEE Transactions on Neural Networks*, vol. 1, no. 1, pp. 4-27, 1991.
- [2] S.-Z. Qin, H.-T. Su, and T. J. McAvoy, " Comparison of four neural net learning methods for dynamic system identification," *IEEE Transactions on Neural Networks*, vol. 3, no. 1, pp. 122-130, 1992.
- [3] P. S. Sastry, G. Santharam, and K. P. Unnikrishnan, " Memory neuron networks for identification and control of dynamical systems," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 306-319, 1994.
- [4] L.-C. Chen, W.-C. Chen, F.-Y. Chang, " Hybrid learning algorithm for Gaussian potential function networks," *IEE Proceedings-D*, vol. 140, no. 6, 1993.
- [5] 최중수, 김형석, 김성중, 권오신, 김종만, "자율분산 신경망을 이용한 비선형 동적 시스템 식별," *전기학회논문지*, 제45권, 4호, pp.574-581, 1996.
- [6] H. Kim, J. S. Choi, and C.-S. Lin, " Self-Organized Distributed Networks for learning highly nonlinear mapping," *Proceedings of artificial Neural Networks in Engineering(ANNIE'94)*, pp.109-114, Nov., St. Louis, U.S.A., 1994.