

안정성을 고려한 동적 신경망의 최적화와 비선형 시스템 제어기 설계

Optimization of Dynamic Neural Networks Considering Stability and Design of Controller for Nonlinear Systems

유동완, 전순용, 서보혁

(Dong-Wan Ryoo, Seong-Wook Park, Soon-Yong Chun, and Bo-Hyeok Seo)

Abstract : This paper presents an optimization algorithm for a stable Self Dynamic Neural Network(SDNN) using genetic algorithm. Optimized SDNN is applied to a problem of controlling nonlinear dynamical systems. SDNN is dynamic mapping and is better suited for dynamical systems than static forward neural network. The real-time implementation is very important, and thus the neuro controller also needs to be designed such that it converges with a relatively small number of training cycles. SDNN has considerably fewer weights than DNN. Since there is no interlink among the hidden layer. The object of proposed algorithm is that the number of self dynamic neuron node and the gradient of activation functions are simultaneously optimized by genetic algorithms. To guarantee convergence, an analytic method based on the Lyapunov function is used to find a stable learning for the SDNN. The ability and effectiveness of identifying and controlling a nonlinear dynamic system using the proposed optimized SDNN considering stability is demonstrated by case studies.

Keywords : self dynamic neural network, learning algorithm, stability, adaptive control, system identification, nonlinear system, Lyapunov function, genetic algorithm

I. 서론

비선형 동적 시스템을 제어하기 위해 연구가 진행되고 있는 신경망은 입출력의 비선형 함수관계를 학습에 의해 표현할 수 있고 본질적으로 병렬처리가 가능한 구조와 일반화 특성을 가지므로 비선형 시스템을 제어하는데 사용할 경우 기존의 제어방법에서의 한계를 극복할 수 있다. 신경망을 이용한 적응 제어기법은 기존의 제어기법에 비해 시스템 파라미터를 추정하는 데 복잡한 수식이 필요 없으며 비선형성이 강한 시스템에 대해서도 좋은 제어성능을 보인다. 신경망을 이용한 비선형 동적 시스템의 식별과 제어가 폭넓게 연구되고 있다[1-5]. Narendra[5]등은 기존의 모델 기준 적응 제어와 유사한 구조를 갖는 정적 신경망을 이용하여 비선형 시스템에 대해 제어방법을 제시하였다. Chen[6]은 비선형 시스템의 자기동조 제어를 위한 정적 전방향 신경망의 학습방법을 제안하였다.

이러한 비선형 시스템의 제어기 설계에 이용되어 온 신경망은 두 가지로 분류될 수 있다. 하나는 정적 전방향 신경망(static forward neural network)구조이며, 다른 하나는 동적 신경망 구조(dynamic neural network)이다. 정적 전방향 신경망은 동적 사상을 표현하지 않는 정적사상인 반면에 동적 신경망은 동적 사상이다. 이 중에 동적 시스템을 제어하기에 적합한 신경망은 내부적으로 동적인 특성을 갖는 동적 신경망이다[7][8].

이러한 동적 신경망은 안정성(stability)과 학습 알고리듬(learning algorithm)개발의 두 가지 방향으로 연구가 되고 있다[7-10]. 시스템을 식별할 때 식별기의 파라미터가 필요 이상으로 많아서는 좋지 않으며 계산시간이 오래 걸린다[11]. 그러므로 식별기를 위한 신경망의 연결강도수는 적절하게 할 필요가 있다. 그러나 동적 신경망은 연결강도의 수가 많기 때문에 실시간 적응 제어를 수행하기 위해서는 무엇보다도 학습속도가 빠른 신경망 구조가 필요하고 안정성이 있어야 한다.

신경망제어기의 노드 수와 활성화 함수의 기울기는 시스템의 출력에 직접적인 영향을 미치므로 신중을 기해야 한다. 그러나 기존의 신경망으로 구성된 제어기는 체계적인 방법이 제시되어 있지 않아 노드수와 활성화 함수의 기울기를 결정하는데 어려움이 있다. 이러한 신경망제어기에 유전 알고리듬을 이용하여 최적화 하려는 연구가 되고 있다. 그러나 Eric[15]은 정적신경망에 적용하였으며, Anna[16]는 안정성보다는 신경망 구조에 역점을 두었다.

따라서 본 논문에서는 안정성을 고려한 연결강도수가 적은 자기 동적 신경망을 사용하여 유전알고리듬으로 노드수와 활성화함수를 최적화 시키므로 비선형 시스템을 안정하게 제어할 수 있는 알고리듬을 제안한다. 유전알고리듬은 진화 원리에서 발견된 몇몇 특징들을 컴퓨터 알고리듬과 결합시켜 복잡한 최적화 문제를 해결하려는 도구로서 목적함수 외 탐색공간에 대한 사전 지식이 필요하지 않고 매우 크고 복잡한 공간일지라도 전역해쪽으로 수렴해 가는 특성을 가지고 있다[12-14]. 이러한 특성 때문에 유전알고리듬은 실제 환경에서 여러 복잡한 최적

접수일자 : 1998. 4. 30., 수정완료 : 1998. 11. 30.

유동완 : 경북대학교 전기공학과

전순용 : 동양대학교 제어계측공학과

서보혁 : 경북대학교 전자·전기공학부

화 문제를 해결하는 도구로서 이용되고 있다. 그리고 자기 동적 신경망은 동적 신경망(Dynamic Neural Network : DNN)[9][10]에 비해 연결강도수가 적어서 학습시간을 단축할 수 있으므로 실시간 연산이 가능하다. 학습하는 동안 제어 알고리듬의 안정성과 수렴성을 보장하기 위해 Lyapunov 함수를 유도하여 학습률을 안정화 시켰다. 제안된 기법을 사용한 최적화 된 자기 동적 신경망으로 비선형 동적 시스템의 식별과 제어에 관한 사례연구를 통하여 제어성능과 오차의 수렴특성을 비교 검토하므로써 제어 알고리듬의 안정성과 효용성을 확인하였다.

II. 정적 신경망과 동적 신경망의 구조

정적 전방향 신경망은 단순히 연결강도에 가해진 입력의 합을 활성화 함수를 통해 출력을 나타낸다. 반면에 동적 신경망은 상태 변수의 방정식으로 출력을 표현할 수 있다. 정적 전방향 신경망의 표현식은 (1)과 같다.

$$O^F(k) = \sum_{j=1}^n W_j^0 F(S_j(k)) = \sum_{j=1}^n W_j^0 F\left(\sum_{i=1}^m W_{ij}^I I_i(k)\right) \quad (1)$$

$$\text{단 } S_j(k) = \sum_{i=1}^m W_{ij}^I I_i(k)$$

여기서 $O^F(k)$ 는 정적 전방향 신경망의 출력, $I_i(k)$ 는 신경으로 들어오는 외부입력을 나타내고, W_{ij}^I, W_j^0 는 입력층, 출력층의 연결강도를 나타낸다. 그리고 m, n 은 각각 입력층, 중간층의 노드를 나타내고, k 는 이산시간을 나타낸다. 동적 신경망의 수학적인 표현은 (2)로 표현된다.

$$\begin{aligned} O^R(k) &= \sum_{j=1}^n W_j^0 F(S_j(k)) = \sum_{j=1}^n W_j^0 F\left(\sum_{i=1}^m W_{ij}^I I_i(k) + \sum_{q=1}^n W_{qj}^H X_q(k-1)\right) \\ &+ \sum_{q=1}^n W_{qj}^H X_q(k-1) = \sum_{j=1}^n W_j^0 F\left(\sum_{i=1}^m W_{ij}^I I_i(k) + \sum_{q=1}^n W_{qj}^H X_q(k-1) + F\left(\sum_{i=1}^m W_{ij}^I I_i(k-1) + \sum_{q=1}^n W_{qj}^H X_q(k-2)\right)\right) \quad (2) \end{aligned}$$

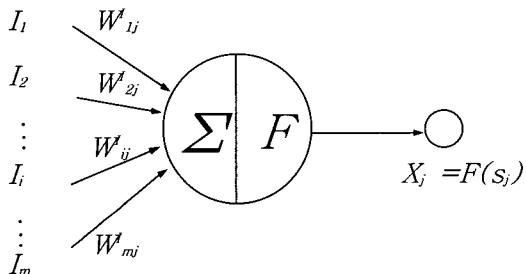


그림 1. 정적 전방향 신경기(FNU)의 구조.

Fig. 1. Structures of feedforward neural unit.

여기서 $X_q(k-1)$ 은 $k-1$ 시각에서 q 번째 신경의 상태이며, $I_i(k)$ 는 신경으로 들어오는 외부입력, $S_j(k)$ 는 외부입력의 합을 나타낸다. $O^R(k)$ 는 동적 신경망의 출력, W^I, W^H, W^0 는 입력층, 중간층, 출력층의 연결강도행렬을 나타낸다. $F(\cdot)$ 는 신경의 활성화 함수로 하이퍼볼릭탄젠트($\tanh(\cdot)$) 함수를 일반적으로 사용한다. (1)은 정적 전방향 신경망으로 정적 사상이고 (2)는 동적 신경

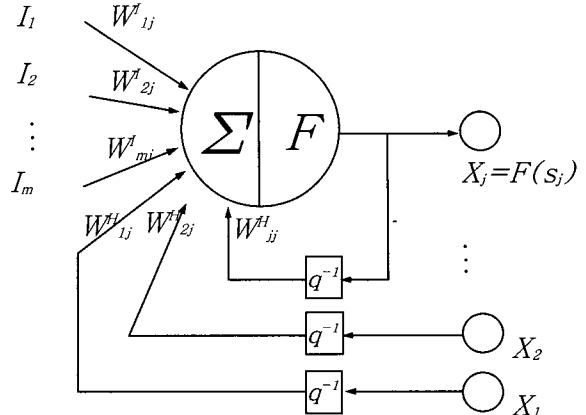


그림 2. 동적 신경기(DNU)의 구조.

Fig. 2. Structures of dynamic neural unit.

망으로 동적 사상을 나타내고 있다. 이러한 신경기의 기본구조를 그림 1과 그림 2에 나타내었다.

III. 자기 동적 신경망의 최적화와 학습 제어 알고리듬

1. 유전 알고리듬을 이용한 동적 신경망의 최적화

유전 알고리듬은 자연 세계의 진화 과정을 컴퓨터상에서 시뮬레이션 하므로써 복잡한 실세계의 문제를 해결하고자 하는 계산 모델이다. 특히 적응적 탐색 학습과 최적화를 통한 공학적인 문제의 해결에 많이 응용된 바 있다. 이러한 유전 알고리듬을 이용하여 본 논문에서 동적신경망의 중간층 유니트의 개수와 활성화 함수의 기울기를 최적화 한다.

유전 알고리듬은 자연 선택과 유전학에 기반한 탐색 방법으로 풀고자 하는 문제에 대한 가능한 해들을 염색체(chromosome)로 표현한 다음 이를 점차적으로 변형함으로써 점점 더 좋은 해들을 생성한다. 각각의 가능한 해를 하나의 개체(individual)로 보고 이들의 집합을 개체군(population)이라 한다. 하나의 개체는 보통 한 개 또는 여러 개의 염색체로 구성되며 염색체를 변형하는 연산자들을 유전연산자(genetic operation)라 한다. 전제적인 유전 알고리듬 흐름은 다음과 같이 요약 할 수 있다.

1. N개의 개체를 가진 최초 개체군 $P(0)$ 를 형성한다.
2. 개체군 $P(i)$ 의 적합도를 평가한다.
3. 적합도를 기반으로 부모 개체를 선택한다.
4. 교차(crossover)나 돌연변이(mutation)와 같은 유전연산자에 의해 변형 하므로써 새로운 N개의 개체 군을 형성한다.
5. 정지 조건을 만족하면 적합한 해를 구한 것으로 보고 끝낸다.
6. 새로운 개체군은 다음 세대가 되고 앞의 2단계로 가서 위의 과정을 반복한다.

1) 개체 선택법

유전 알고리듬에서 적자 생존의 원리를 구현한 것으로 새로운 세대의 개체군이 이 전 개체군 보다 나은 적합도를 갖도록 선택하는 방법이다. 그 기본 원리는 더 좋은 개체(우성)에 선택될 수 있는 특권을 부여하는 것이다.

2) 교차 연산자

다른 구조를 갖는 개체들을 탐구하기 위한 연산자로서 먼저 한 세대의 염색체 수만큼의 난수(random number)를 발생하여 교차 확률(crossover probability)보다 작은 난수에 해당되는 염색체를 교차시킨다. 선택된 염색체들 중에서 교차 쌍을 정하는 것도 무작위로 선택하며 만약 선택된 염색체의 수가 홀수이면 한 개의 염색체를 더 선택하던지 아니면 선택에서 제거시킨다. 두 개의 부모 개체가 선택되면 교차 위치가 결정되고 결정된 교차 위치를 기준으로 두 염색체간의 유전자 값이 교환된다. 예를 들면 선택된 두 염색체가 $v_1=(0\ 1\ 1\ 0\ 1\ 1)$, $v_2=(1\ 1\ 0\ 1\ 0\ 0)$ 이고 교차 위치가 3이면 이때 생성되는 염색체는 $v'_1=(0\ 1\ 1\ 1\ 0\ 0)$, $v'_2=(1\ 1\ 0\ 0\ 1\ 1)$ 가 된다.

3) 돌연변이 연산자

개체군의 다양성을 유지하여 새로운 유전 형질을 부여해 주는 연산자로서 개체군 전체의 비트 수만큼의 난수를 발생시켜 돌연변이 확률(mutation probability)보다 작은 난수에 해당되는 비트를 0은 1로 1은 0으로 변형한다. 예를 들면 염색체가 $v=(1\ 1\ 0\ 1\ 1\ 0)$ 이고 선택된 비트가 세번째라면 발생되는 염색체는 $v'=(1\ 1\ 1\ 1\ 1\ 0)$ 가 된다.

2. 자기 동적 신경망의 구조

자기 동적 신경망은 동적 신경망에서 변화된 구조를 가진다. 즉 자기 동적으로 구성된 신경으로 구성되며 연결강도의 수가 적다. 자기 동적 신경망을 수학적인 표현식으로 나타내면 (3)과 같다.

$$O(k) = \sum_j W_j^0 X_j(k) \quad (3)$$

여기서, $X_j(k) = F(S_j(k))$, $S_j(k) = W_j^D X_j(k-1) + \sum_i W_{ij}^I I_i(k)$ 이다. (3)을 다시 전개하여 표현하면 (4)와 같이 나타낼 수 있다.

$$\begin{aligned} O(k) &= \sum_{j=1}^n W_j^0 F \left(\sum_{i=1}^m W_{ij}^I I_i(k) + W_j^D \cdot \right. \\ &\quad \left. F \left(\sum_{i=1}^m W_{ij}^I I_i(k-1) + W_j^D X_j(k-2) \right) \right) \\ &= \phi_N^D(I_l(l)), l=0,1,2,\dots,k \end{aligned} \quad (4)$$

여기서 $X_j(k)$ 는 j 번째 동적 신경의 출력이며, $I_l(k)$ 는 신경으로 들어오는 i 번째 외부입력이며, $S_j(k)$ 는 j 번째 동적 신경으로 들어오는 외부입력의 합이며, $O(k)$ 는 신경망의 출력이며, W^I , W^D , W^0 는 입력층에서 중간층, 중간층에서 중간층, 중간층에서 출력층의 연결강도행렬이다. $F(\cdot)$ 는 시그모이드 함수이며, $\phi_N^D(\cdot)$ 은 비선형 함수이고 l 과 k 는 0 또는 양의 실수이다. (4)는 이전의 입력을 포함하고 있는 비선형 동적 사상을 나타낸다. 자기 동적 신경기의 구조를 그림 3에 나타내었다.

3. 자기 동적 신경망으로 구성된 식별기

자기 동적 신경망으로 구성된 식별기를 수식으로 표현하면 (5)~(9)와 같이 나타낼 수 있다.

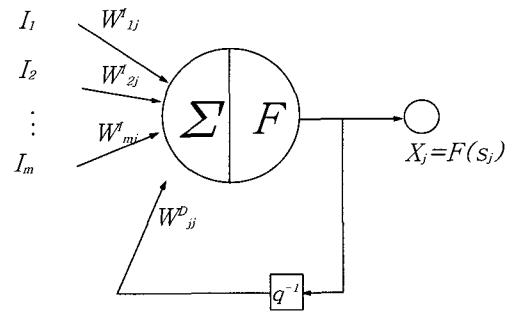


그림 3. 자기 동적 신경기(SDNU)의 구조.

Fig. 3. Structure of self dynamic neural unit.

$$\hat{y}(k) = \sum_j W_j^{f0}(k) X_j^f(k) + \sum_j W_j^{go}(k) X_j^g(k) U(k) \quad (5)$$

$$X_j^f(k) = F(S_j^f(k)) \quad (6)$$

$$S_j^f(k) = W_j^{fD} X_j^f(k-1) + \sum_i W_{ij}^{fI} I_i(k) \quad (7)$$

$$X_j^g(k) = F(S_j^g(k)) \quad (8)$$

$$S_j^g(k) = W_j^{gD} X_j^g(k-1) + \sum_i W_{ij}^{gI} I_i(k) \quad (9)$$

여기서 $X_j^f(k)$, $X_j^g(k)$ 는 추정함수 $\hat{f}(\cdot)$ 과 $\hat{g}(\cdot)$ 의 j 번째 동적 신경의 출력이며, $I_i(k)$ 는 신경으로 들어오는 i 번째 외부입력이며, $S_j^f(k)$ 와 $S_j^g(k)$ 는 각각의 추정함수 $\hat{f}(\cdot)$ 와 $\hat{g}(\cdot)$ 의 j 번째 동적 신경으로 들어오는 외부입력의 합이며, $\hat{y}(k)$ 는 식별기의 출력이며, $U(k)$ 는 시스템 입력이며, W^f , W^D , W^{f0} 와 W^g , W^{gD} , W^{go} 는 추정함수 $\hat{f}(\cdot)$ 와 $\hat{g}(\cdot)$ 입력층에서 중간층, 중간층에서 중간층, 중간층에서 출력층의 연결강도행렬이다. 비선형 함수의 자기 동적 신경망의 모형과 유전알고리듬의 적용을 그림 4에 나타내었다. 그림 4(a)에서와 같이 연결강도수를 줄인 자기동적신경망을 이용하여 시스템을 f와 g로 분리 추정하며 그림 4(b)에서 유전알고리듬으로 동적신경망의 유니트의 개수와 활성화 함수의 기울기를 최적화 한다.

4. 자기 동적 신경망으로 구성된 비선형 동적 시스템의 제어기와 최적화 알고리듬

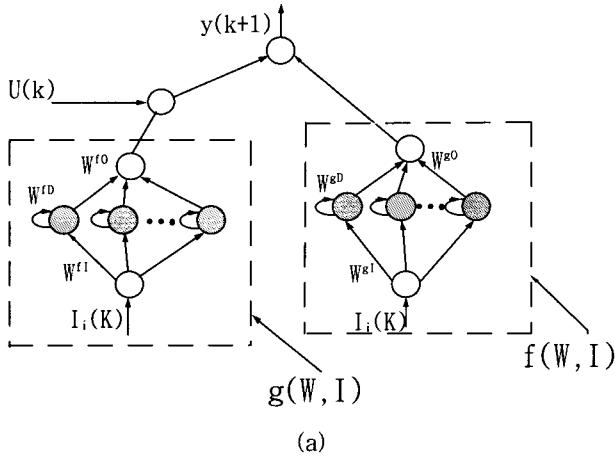
자기 동적 신경망을 사용하여 비선형 시스템을 식별하고 제어할 수 있다. 자기 동적 신경망으로 구성된 식별기는 원하는 시스템의 출력을 추정하여 신경망 제어기의 입력으로 사용되어지고, 신경망으로 구성된 제어기는 시스템을 제어한다. 신경망으로 구성된 제어기는 식별기로부터 제공된 정보를 가지고 시스템의 제어입력을 구해낸다. 입의의 비선형 동적 시스템을 이산 시스템으로 나타내면 (10)과 같이 표현할 수 있다[6][10][17].

$$\begin{aligned} y(k+1) &= f[y(k), y(k-1), \dots, y(k-n), u(k-1), \\ &\quad u(k-2), \dots, u(k-m)] + g[y(k), y(k-1), \dots, \\ &\quad y(k-n), u(k-1), u(k-2), \dots, u(k-m)] \cdot u(k) \end{aligned} \quad (10)$$

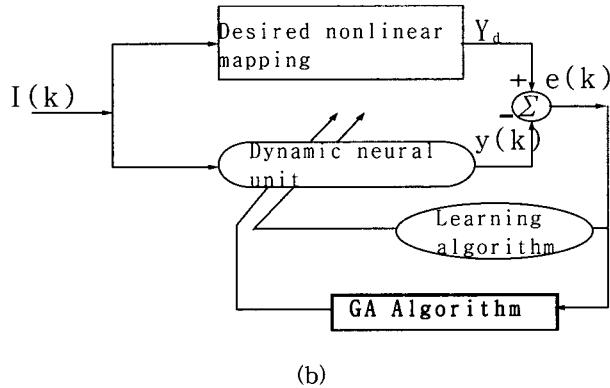
여기서 $y(k)$ 는 시스템 출력이며 $u(k)$ 는 시스템 입력이고 $f(\cdot)$ 와 $g(\cdot)$ 는 비선형 함수이다. (10)의 f 와 g

함수를 자기 동적 신경망을 사용하여 식별된 출력은 (11)와 같이 나타낼 수 있다.

$$\hat{y}(k+1) = \hat{f}(I, W^f) + \hat{g}(I, W^g) U(k) \quad (11)$$



(a)



(b)

그림 4. (a) 비선형 함수의 자기 동적 신경망 모형, (b) 유전 알고리듬을 이용한 동적 신경망의 최적화.

Fig. 4. (a) SDNN model of the nonlinear system, (b) Optimization of dynamic neural unit using GA algorithm.

오차 함수는 실제 시스템의 출력과 식별기 출력의 합으로 표현된다.

$$E = \frac{1}{2} (y(k) - \hat{y}(k))^2 \quad (12)$$

추정되는 함수 $\hat{f}(I, W)$ 의 각 연결강도 오차 기울기는 (13)과 같이 나타낼 수 있다.

$$\frac{\partial E}{\partial W^f} = -e(k) \frac{\partial \hat{y}(k)}{\partial W^f} \quad (13)$$

여기서 $e(k) = y(k) - \hat{y}(k)$ 이다. 신경망의 연결강도의 학습 법칙은 (14)과 같다.

$$W^f(k+1) = W^f(k) + [\eta_f] \left(-\frac{\partial E}{\partial W^f} \right) \quad (14)$$

여기서 $[\eta_f]$ 는 추정되는 함수 $\hat{f}(I, W)$ 의 각 연결강도 학습률이다.

본 논문에서 사용된 자기 동적 신경망 $\hat{f}(I, W)$ 의 입력층, 중간층, 출력층의 연결강도에 대한 출력의 기울기는 (15)~(17)과 같이 나타낼 수 있다.

$$\frac{\partial \hat{y}(k)}{\partial W_j^{f0}} = X_j^f(k) \quad (15)$$

$$\frac{\partial \hat{y}(k)}{\partial W_j^{gD}} = W_j^{g0} L_j^f(k) \quad (16)$$

$$\frac{\partial \hat{y}(k)}{\partial W_{ij}^{gI}} = W_j^{g0} M_{ij}^f(k) \quad (17)$$

여기서 $L_j^f(k) = \partial X_j^f(k) / \partial W_j^{fD}$, $M_{ij}^f = \partial X_j^f(k) / \partial W_{ij}^{fI}$ 이며 (18)과 (19)가 성립한다.

$$L_j^f(k) = F'(S_j^f) (X_j^f(k-1) + W_j^{fD} L_j^f(k-1)), \quad L_j^f(0) = 0 \quad (18)$$

$$M_{ij}^f(k) = F'(S_j^f) (I_i(k) + W_j^{fD} M_{ij}^f(k-1)), \quad M_{ij}^f(0) = 0 \quad (19)$$

(15)에서 (19)에 표현된 자기 동적 신경망 $\hat{f}(I, W)$ 의 입력층에서 중간층, 중간층에서 중간층, 중간층에서 출력층의 연결강도에 대한 출력의 기울기를 사용하여 역전파법으로 학습시킬 수 있다.

추정되는 $\hat{g}(I, W)$ 의 각 연결강도 오차 기울기는 (20)과 같이 표현된다.

$$\frac{\partial E}{\partial W^g} = -e(k) \frac{\partial \hat{y}(k)}{\partial W^g} \quad (20)$$

신경망의 연결강도의 학습법칙은 (21)와 같다.

$$W^g(n+1) = W^g(n) + [\eta_g] \left(-\frac{\partial E}{\partial W^g} \right) \quad (21)$$

여기서 $[\eta_g]$ 는 추정되는 함수 $\hat{g}(I, W)$ 의 각 연결강도 학습률이다.

본 논문에서 사용된 자기 동적 신경망 $\hat{g}(I, W)$ 의 입력층, 중간층, 출력층의 연결강도에 대한 출력의 기울기는 (22)에서 (24)과 같다.

$$\frac{\partial \hat{y}(k)}{\partial W_j^{g0}} = X_j^g(k) U(k) \quad (22)$$

$$\frac{\partial \hat{y}(k)}{\partial W_j^{gD}} = W_j^{g0} U(k) L_j^g(k) \quad (23)$$

$$\frac{\partial \hat{y}(k)}{\partial W_{ij}^{gI}} = W_j^{g0} U(k) M_{ij}^g(k) \quad (24)$$

여기서 $L_j^g(k) = \partial X_j^g(k) / \partial W_j^{gD}$, $M_{ij}^g = \partial X_j^g(k) / \partial W_{ij}^{gI}$ 이며 (25)과 (26)가 성립한다.

$$L_j^g(k) = F'(S_j^g) (X_j^g(k-1) + W_j^{gD} L_j^g(k-1)), \quad L_j^g(0) = 0 \quad (25)$$

$$M_{ij}^g(k) = F'(S_j^g) (I_i(k) + W_j^{gD} M_{ij}^g(k-1)), \quad M_{ij}^g(0) = 0 \quad (26)$$

(13)에서와 같이 비선형 시스템의 출력과 식별기의 출력에 의해 얻어진 오차함수를 이용하여 (15)~(19)를 이용하여 제어기 신경망 $\hat{f}(I, W)$ 의 입력층, 중간층, 출력층

의 연결강도를 학습시키며, (20)에서와 같이 비선형 시스템의 출력과 식별기의 출력에 의해 얻어진 오차함수를 이용하여 (22)–(26)을 이용하여 제어기 신경망 $\hat{g}(I, W)$ 의 입력층, 중간층, 출력층의 연결강도를 학습 시킬 수 있다.

그림 5는 자기 동적 신경망으로 구성된 제어기의 블럭선도를 나타낸 것이다. 여기서 TDL은 Tapped Delay Line을 나타내고 입력은 시간지연을 갖는 제어기 입력과 시스템 출력이 된다. 유전알고리듬은 식별기와 제어기를 최적화 시킨다. 자기 동적 신경망으로 구성된 식별기의 입력은 이전 시스템의 출력, 그리고 이전의 제어신호로 구성되며, 식별기 출력과 시스템 출력사이의 오차를 최소화하므로써 식별된 정보를 제어기에 제공한다. 자기 동적 신경망으로 구성된 제어기의 입력은 기준입력, 이전 시스템의 출력, 그리고 이전의 제어신호로 구성되며, 신경망으로 구성된 제어기의 출력이 시스템의 제어신호가 된다.

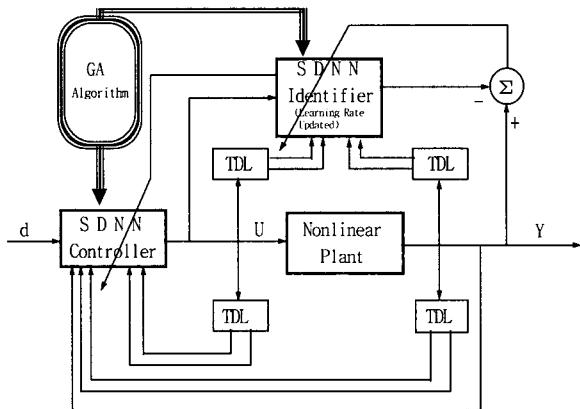


그림 5. 자기 동적 신경망으로 구성된 제어기의 블럭선도.

Fig. 5. A block diagram of SDNN based control system.

하나의 유전자 개체에 의해 표현되는 구조를 갖는 동적신경망을 정해진 횟수만큼 학습시키는 동안에 측정한 mean square error의 역수를 적합도 함수로 결정하였고 정수형 염색체를 사용하였다.

$$Fitness = \frac{1}{m} \sum_{k=0}^m (error(k))^2 = \frac{1}{MSE} \quad (27)$$

IV. 안정성을 고려한 학습률

학습 제어 알고리듬에서 안정한 범위의 학습률을 결정하는 것은 대단히 중요하다. 작은 값의 학습률은 수렴성은 보장할지라도 학습속도가 매우 느리게 된다. 반면에 학습률이 너무 크면 제어 알고리듬이 불안정하게 된다. 즉 안정성을 보장하지 못한다. 그러므로 이 장에서는 안정성을 가진 적응 학습률을 Lyapunov 함수를 사용하여 유도한다. Lyapunov 후보 함수를 (28)과 같이 나타낸다.

$$V(k) = \frac{1}{2} e^2(k) \quad (28)$$

Lyapunov 함수의 변화량은 (29)와 같이 표현될 수 있다.

$$\Delta V(k) = V(k+1) - V(k) = \frac{1}{2} [e^2(k+1) - e^2(k)] \quad (29)$$

(28)에서 학습에 의한 오차는 오차의 변화분으로 나타낼 수 있다.

$$e(k+1) = e(k) + \Delta e(k) = e(k) + [\frac{\partial e(k)}{\partial W}]^T \Delta W \quad (30)$$

여기서 ΔW 는 임의의 연결강도의 변화를 나타낸다. (13)와 (14)로부터 ΔW 를 (34)와 같이 나타낼 수 있다.

$$\Delta W = -\eta e(k) \frac{\partial e(k)}{\partial W} = \eta e(k) \frac{\partial \hat{y}(k)}{\partial W} \quad (31)$$

여기서 W 와 η 는 신경망의 임의의 연결강도와 학습률을 나타낸다.

[보조 정리 1] : η 를 연결강도의 학습률이라 하고 P_{MAX} 를 연결강도에 대한 출력의 변화율의 최대값으로 (32)와 같이 정의한다.

$$P_{MAX} = \max_k \| P(k) \| \quad (32)$$

여기서 $P(k) = \frac{\partial \hat{y}(k)}{\partial W}$ 이고 $\| \cdot \|$ 는 Euclidean norm이다. η 가 (33)을 만족하면 학습 알고리듬의 수렴성을 보장한다.

$$0 < \eta < \frac{2}{P_{MAX}^2} \quad (33)$$

증명 : (30)의 $e(k+1)$ 를 (29)에 대입하면 $\Delta V(k)$ 는 (34)과 같이 표현할 수 있다.

$$\begin{aligned} \Delta V(k) &= \Delta e(k) [e(k) + \frac{1}{2} \Delta e(k)] = [\frac{\partial e(k)}{\partial W}]^T \eta e(k) \frac{\partial \hat{y}(k)}{\partial W} \\ &\quad \cdot \{ e(k) + \frac{1}{2} [\frac{\partial e(k)}{\partial W}]^T \eta e(k) \frac{\partial \hat{y}(k)}{\partial W} \} \end{aligned} \quad (34)$$

$$\frac{\partial e(k)}{\partial W} = -\frac{\partial \hat{y}(k)}{\partial W} \text{이므로,}$$

$$\begin{aligned} \Delta V(k) &= -\eta e^2(k) \|\frac{\partial \hat{y}(k)}{\partial W}\|^2 + \frac{1}{2} \eta^2 e^2(k) \|\frac{\partial \hat{y}(k)}{\partial W}\|^4 \\ &\equiv -\lambda e^2(k) \end{aligned} \quad (35)$$

(35)을 $P(k)$ 로 나타내고 $\eta_V = \eta P_{MAX}^2$ 라고[18] 두면 (36)와 같이 표현할 수 있다.

$$\begin{aligned} \lambda &= \frac{1}{2} \|P(k)\|^2 \eta (2 - \eta \|P(k)\|^2) \\ &= \frac{1}{2} \|P(k)\|^2 \eta (2 - \frac{\eta_V \|P(k)\|^2}{P_{MAX}^2}) \\ &\geq \frac{1}{2} \|P(k)\|^2 \eta (2 - \eta_V) > 0 \end{aligned} \quad (36)$$

(36)로부터 학습률의 수렴조건은 $0 < \eta_V < 2$ 이 된다. 즉, (36)를 만족한다면 수렴성을 보장한다.

$$\eta_V (2 - \eta_V) / P_{MAX}^2 > 0 \quad (37)$$

여기서 최적의 안정성을 고려한 학습률의 값은 $\eta_V=1$ 일 때이다.

$$\eta^* = \frac{1}{P_{MAX}^2} \quad (38)$$

[보조 정리 2] : η_f^0 , η_f^D 와 η_f^I 를 각각의 신경망 연결강도 W^{gO} , W^{gD} 그리고 W^{gI} 의 학습률이라고 할 때 η_f^0 , η_f^D 그리고 η_f^I 가 (39)~(41)를 만족하면 학습 알고리듬의 수렴성을 보장한다.

$$(a) 0 < \eta_f^0 < \frac{2}{h_f} \quad (39)$$

$$(b) 0 < \eta_f^D < \frac{2}{h_f} \left[\frac{1}{W_{MAX}^{gO}} \right]^2 \quad (40)$$

$$(c) 0 < \eta_f^I < \frac{2}{(n_f + h_f)} \left[\frac{1}{W_{MAX}^{gO}} I_{MAX}^f \right]^2 \quad (41)$$

여기서 $0 < |W_j^{gD}| < 1$ ($j=1, 2, \dots, h_f$)이고 h_f 는 중간층 노드수를 나타내고 n_f 는 입력층의 노드수를 나타낸다. $W_{MAX}^{gO} \equiv \max_k \|W^{gO}(k)\|$, $I_{MAX}^f \equiv \max_k \|I^f(k)\|$ 그리고 $\|\cdot\|$ 는 sup-norm을 나타낸다.

[보조 정리 3] : η_g^0 , η_g^D 그리고 η_g^I 를 각각의 신경망 연결강도 W^{gO} , W^{gD} 그리고 W^{gI} 의 학습률이라고 할 때 η_g^0 , η_g^D 그리고 η_g^I 가 (42)~(44)를 만족하면 학습 알고리듬의 수렴성을 보장한다. 여기서 $0 < |W_j^{gD}| < 1$, $j=1, 2, \dots, h_g$ 이다

$$(a) 0 < \eta_g^0 < \frac{2}{U_{MAX}^2 h_g} \quad (42)$$

$$(b) 0 < \eta_g^D < \frac{2}{U_{MAX}^2 h_g} \left[\frac{1}{W_{MAX}^{gO}} \right]^2 \quad (43)$$

$$(c) 0 < \eta_g^I < \frac{2}{U_{MAX}^2 (n_g + h_g)} \left[\frac{1}{W_{MAX}^{gO}} I_{MAX}^g \right] \quad (44)$$

여기서 h_g 는 $\hat{g}(\cdot)$ 중간층의 신경 수이며, n_g 는 신경망의 입력 개수이다. $W_{MAX}^{gO} \equiv \max_k \|W^{gO}(k)\|$, $I_{MAX}^g \equiv \max_k \|I^g(k)\|$, $U_{MAX} \equiv \max_k |U(k)|$ 를 나타낸다.

증명 :

[A] :

(22)에서 $P_g(k) = \frac{\partial \hat{y}(k)}{\partial W^{gO}} = X^g(k) U(k)$ 이며 $X^g = [X_1^g, X_2^g, \dots, X_{h_g}^g]^T$ 이다. $0 < X_j^g < 1$, $j=1, 2, \dots, h_g$ 이기 때문에 $\|P_g(k)\| < U_{MAX} \sqrt{h_g}$ 이며, $P_{g, MAX}^2 = U_{MAX}^2 h_g$ 이다.

[B] :

(25)에서 $\frac{\partial X_i(k)}{\partial W_j^{gD}} = L_j^g(k)$ 이며 (45-1)과 같이 나타낼 수 있다.

$$L_j^g(k) = F'(k)(X_j(k-1) + W_j^{gD} L_j^g(k-1)) \quad (45-1)$$

여기서 $F'(k) \equiv F'(S_i(k))$ 이다. (45-1)의 해를 (45-2)

와 같이 나타낼 수 있다.

$$L_j^g(l) = \sum_{m=0}^{l-1} \prod_{n=0}^{l-m-1} F'(l-n) X_j(m) (W_j^{gD})^{l-1-m} + \prod_{n=1}^l F'(n) L_j^g(0) (W_j^{gD})^l \quad (45-2)$$

초기치를 $L_j^g(0)=0$ 로 두면 (45-3)과 같이 나타낼 수 있다.

$$L_j^g(l) = \sum_{m=0}^{l-1} \prod_{n=0}^{l-m-1} F'(l-n) X_j(m) (W_j^{gD})^{l-1-m}, \quad j=1, 2, \dots, h_g \quad (45-3)$$

여기서 $0 < F'(S_i) < 0.5$, $0 < X_j(m) < 1$, 그리고 $0 < W_j^{gD} < 1$, $j=1, 2, \dots, h_g$ 이기 때문에 (45-4)와 같이 표현된다.

$$|L_j^g(l)| \leq \sum_{m=0}^{l-1} \prod_{n=0}^{l-m-1} |F'(l-n)| |X_j(m)| |W_j^{gD}|^{l-1-m} \quad (45-4)$$

$$|L_j^g(l)| \leq \sum_{m=0}^{l-1} (0.5)^{l-m} \quad (45-5)$$

$r=l-m-1$ 라 두면 (45-6)과 같이 표현된다

$$|L_j^g(l)| \leq \sum_{r=0}^{l-1} (0.5)^{r+1} \leq 0.5 \sum_{r=0}^{\infty} (0.5)^r = 1, \quad j=1, 2, \dots, h_g \quad (45-6)$$

그러므로 벡터 $L^g(l) = [L_1^g, L_2^g, \dots, L_{h_g}^g]^T$ 에 대해서 (45-7) 또는 (45-8)로 나타낼 수 있다.

$$\|L^g(l)\| \leq \sqrt{h_g} \quad (45-7)$$

$$\left\| \frac{X(k)}{W^{gD}} \right\| \leq \sqrt{h_g} \quad (45-8)$$

(23)으로부터 (45-9)와 같이 표현된다

$$\left| \frac{\partial \hat{y}(k)}{\partial W_j^{gD}} \right| \leq |W_j^{gO}| |U(k)| |L_j^g(k)| \leq |W_j^{gO}| |U(k)| \leq \|W^{gO}(k)\| |U(k)| \quad (45-9)$$

여기서 $\|W^{gO}(k)\| \equiv \max_j |W_j^{gO}(k)|$ 이다.

$$\left\| \frac{\partial \hat{y}(k)}{\partial W^{gD}} \right\| \leq \sqrt{h_g} \|W^{gO}(k)\| |U(k)| \quad (45-10)$$

$W_{MAX}^{gO} \equiv \max_k \|W^{gO}(k)\|$, (45-10)은 (45-11)과 같이 나타낼 수 있다.

$$\left\| \frac{\partial \hat{y}(k)}{\partial W^{gD}} \right\| \leq \sqrt{h_g} U_{MAX} W_{MAX}^{gO} \quad (45-11)$$

[C] :

(26)에서 $\frac{\partial X_i(k)}{\partial W_{ij}^{gI}} = M_{ij}^g(k)$ 이며 (45-12)와 같이 표현할 수 있다.

$$M_{ij}^g(k) = F'(k)(I_i^g(k) + W_j^{gD} M_{ij}^g(k-1)) \quad (45-12)$$

그리고 (45-12)의 해를 (45-13)과 같이 표현할 수 있다.

$$M_{ij}^g(l) = \sum_{m=0}^{l-1} \prod_{n=0}^{l-m-1} F'(l-n) I_i^g(m) (W_j^{gD})^{l-1-m} + \prod_{n=1}^l f(n) M_{ij}^g(0) (W_j^{gD})^l \quad (45-13)$$

이때 초기치를 $M_{ij}^g(0)=0$ 라 두면 (45-14)와 같이 나타낼 수 있다.

$$M_{ij}^g(l) = \sum_{m=0}^{l-1} \prod_{n=0}^{l-m-1} F'(l-n) I_i^g(m) (W_j^{gD})^{l-1-m}, \\ j=1, 2, \dots, h_i \quad (45-14)$$

$$|M_{ij}^g(l)| \leq \sum_{m=0}^{l-1} (0.5)^{l-m} I_i^g(m) \quad (45-15)$$

여기서 I_{MAX}^g 를 입력층의 최대값으로 두면 (45-16)과 같이 나타낼 수 있다.

$$|M_{ij}^g(l)| \leq \sum_{m=0}^{l-1} (0.5)^{l-m} I_{MAX}^g \leq I_{MAX}^g \quad (45-16)$$

$$\left\| \frac{X(k)}{W^{gl}} \right\| \leq \sqrt{n_g + h_g} I_{MAX}^g \quad (45-17)$$

그러므로 (24)으로부터 (45-18)과 같이 표현할 수 있다.

$$\left\| \frac{\partial \hat{y}(k)}{\partial W^{gl}} \right\| \leq \sqrt{n_g + h_g} I_{MAX}^g \|W^{go}(k)\| |U(k)| \\ \leq \sqrt{n_g + h_g} U_{MAX} I_{MAX}^g W_{MAX}^{go} \quad (45-18)$$

여기서 $W_{MAX}^{go}(k) \equiv \max_k \|W^{go}(k)\|$ 이다.

보조정리 2도 보조정리 3과 유사하게 유도할 수 있다. 보조정리 2와 3에서와 같이 스텝(k)에 대해서 신경망의 연결강도와 신경망의 입력에 의해서 다음 스텝의 학습률을 (39)-(44)에서의 유계중에서 (38)과 같이 조절해 나간다.

V. 모의실험

사례연구를 통하여 유전알고리듬으로 최적화된 안정한 학습을 고려한 자기동적신경망을 사용하여 비선형 시스템의 식별, 제어능력의 유용성을 알아보기 위해서 모의실험을 실시하였다. 첫째로, 다음과 같은 비선형 시스템 모형[6][9][10]에 적용하여 시스템을 식별하는 경우를 다루어 본다. 비선형 시스템 모형이 (46)-(47)으로 주어진 경우, 오차가 작게 수렴하는 가를 알아보는 실험이다.

$$y_p(k+1) = a \sin[2y_p(k)] + b u(k) \\ a=0.8, b=1.2 \quad (46)$$

$$u(k) = 0.5 [\sin(0.5r_k) + \cos(2r_k) + \sin^2(3r_k) + \cos^3(4r_k) \\ \cos(5r_k) + \sin(20r_k)], r_k = \pi k/40 \quad (47)$$

여기서 $y_p(k+1)$, $u(k)$ 는 플랜트의 출력과 입력을 나타내고 $r(k)$ 는 샘플링 변수이다. (46)-(47)으로부터 입·출력 데이터를 얻어 시스템을 식별하고 15회 반복 학습한 후 학습된 초기치를 가지고 플랜트 파라미터가 변화할 때, 시스템의 출력을 식별한다. 반복 학습한 후 원하는 출력을 추정하여 학습능력을 조사한다. 또한 초기값

의 학습 없이 온라인으로 학습하여 온라인 식별이 가능함을 조사해 본다. 둘째로, 비선형 시스템을 자기 동적 신경망 제어기를 사용하여 비선형 함수가 시간적으로 변화하는 경우 적응 추적하며, 플랜트의 동적 특성이 변화하는 경우에 대해 적응하여 추적하는 동적 특성을 알아보기로 한다. 제안된 제어 알고리듬을 (48)-(49)와 같은 비선형 시스템 모형[20]에 적용하여 제어 성능을 조사한다. (48)-(49)로부터 입·출력 데이터를 얻어 시스템을 식별하고 15회 반복 학습한 후 학습된 초기치를 가지고 (50)과 같은 플랜트 변화를 식별하여 제어한다. 반복 학습한 후 원하는 출력을 추정하여 제어성능을 조사한다.

$$y_p(k+1) = 0.35 \left\{ \frac{y_p(k) y_p(k-1) + [y_p(k) + 2.5]}{1 + y_p(k)^2 + y_p(k-1)^2} + u(k) \right\} \quad (48)$$

$$u(k) = [\sin(0.5r_k) + \cos(2r_k) + \sin^2(3r_k) + \cos^3(4r_k) \\ \cos(5r_k) + \sin(20r_k)] - 0.75, r_k = \pi k/40 \quad (49)$$

$$y_p(k+1) = 0.35 \left\{ \frac{y_p(k) + [y_p(k-1) + 2.5]}{1 + y_p(k)^2} + 1.2 u(k) \right\} \quad (50)$$

VI. 결과 및 검토

첫 번째 사례연구에서는 유전알고리듬으로 최적화된 안정한 학습을 고려한 자기동적신경망을 사용하여 비선형 시스템의 식별, 제어능력을 알아보기 위해서 모의실험을 실시하였다. 그림 6은 자기동적신경망을 유전알고리듬으로 유니트 개수와 활성화 함수의 기울기를 최적화 시킬 때 최대 적합도의 변화를 나타내었다. 교배확률(P_c)은 0.9, 돌연변이 확률 P_m 은 0.3으로 하였다. 최적화 된 신경망을 가지고 (46)로부터 200개의 입·출력상의 데이터를 15회 반복 학습한 후 결과를 그림 7에 나타내었다. 그림에서 나타난 바와 같이 유전알고리듬과 안정성을 고려한 학습률을 사용한 자기 신경망은 원하는 출력을 잘 추적함을 알 수 있었다. 참고문헌[9]와 [10]에서 정적 정방향신경망과 동적신경망을 비교하여 동적신경망이 더 좋은 결과를 나타냄을 보였다. 그러므로 본 논문은 일반적인 동적 신경망과의 학습능력을 비교 검토하기 위해 반복 학습하는 동안 오차의 실효치를 그림 8에 나타내었다. 유전알고리듬으로 찾은 f 함수의 중간층 신경망유니트가 7개, g 함수의 중간층 신경망유니트가 1개, 활성화함수의 기울기 $\tanh(3.5s)$ 를 사용한 자기동적신경망과, f 와 g 함수의 각 중간층 유니트를 더 증가시켜 7개, 활성화함수의 기울기 $\tanh(1s)$ 를 사용한 자기동적신경망을 사용하여 시스템을 식별하고 비교하였다. 여기서 입력층과 출력층의 유니트수는 고정되어 있고, 층수는 3층 신경망으로 하였다. 유전알고리듬으로 최적화 시킨 동적신경망오차의 수렴특성이 우수함을 그림 8로부터 알 수 있었다. 15회 학습한 후 동적 신경망으로 학습한 결과와 비교한 결과를 표 1에 나타내었다. 학습된 자기동적신경망을 사용하여 기준 입력과 시스템 출력 결과를 그림 9에 도시하였고 플랜트의 파라미터가 70번 째부터 변화하는 경우에 제어 성능을 그림 10과 11에 나

표 1. 사례연구 1에 대한 15회 학습한 후 적합도와 오차의 실효치.

Table 1. Fitness and mean square error for case study 1 after 15 training cycles.

Fitness, MSE (before Optimize)	Fitness	270.27
	MSE	0.0037
Fitness, MSE (after Optimize)	Fitness	2618.126
	MSE	0.00038

타내었다. 실험결과로부터 제안한 학습 알고리듬을 사용한 경우에 회상능력과 파라미터의 변화에 대해서도 우수한 특성이 있음을 알 수 있었다. 제안한 학습률과 자기동적 신경망을 사용하여 온라인 제어시 결과를 그림 12에 나타내었다. 일반적으로 온라인 사용할 경우에 출력응답이 초기치에 따라 발산하므로 사전 학습후 어느 정도 시스템을 식별하여 원하는 출력을 추정하는 데, 본 연구에서는 학습되지 않은 상태에서 결과를 얻었다. 즉 신경망으로 구성된 제어기는 학습하는데 시간이 많이 걸리고 학습을 하고 나서 학습된 정보로부터 제어를 하지만 신경망의 학습없이 온라인으로 제어하기는 어렵다. 그림 12나 13은 제어성능은 초기에는 응답이 진동하지만, 신경망의 학습없이 온라인으로 제어를 행한 것이다.

두 번째, 플랜트가 이산시간에 따라 변화는 경우에 대해 사례연구하므로써 연구결과의 일반성과 강인성을 조사하였다. 그림 14는 자기동적신경망을 유전알고리듬으로 유니트 개수와 활성화 함수의 기울기를 최적화시킬 때 최대 적합도의 변화를 나타내었다. 교배확률(P_c)은 0.9, 돌연변이 확률 P_m 은 0.3으로 하였다. (48)과 (49)로부터 200개의 입·출력쌍의 데이터를 가지고 15회 반복 학습한 후 결과를 그림 15에 나타내었다. 최적화된 자기신경망은 원하는 출력을 잘 추적함을 알 수 있었다. 최적화된 동적 신경망과의 학습능력을 비교 검토하기 위해 반복 학습하는 동안 오차의 실효치를 그림 16에 나타내었다. 유전알고리듬으로 찾은 f 함수의 중간층 신경망유니트가 7개, g 함수의 중간층 신경망유니트가 2개, 활성화함수의 기울기 $\tanh(2.5s)$ 를 사용한 자기동적신경망과, f 와 g 함수의 각 중간층 유니트가 7개, 활성화함수의 기울기 $\tanh(1s)$ 를 사용한 자기동적 신경망을 사용하여 시스템을 식별하고 비교하였다. 여기서 입력층과 출력층의 유니트수는 고정되어 있고, 층수는 3층 신경망으로 하였다. 그림 16에서 나타난 바와 같이 오차의 수렴특성이 향상됨을 알 수 있었다. 본 논문에서 제안한 방법이 모의 실험에서 그림 8과 16에서 기준의 방법과 비교검토하였다. 오차실효치의 수렴속도가 향상되었으며, 최종 오차실효치도 더 줄어들었음을 알 수 있었다. 15회 학습한 후 동적 신경망으로 학습한 결과와 비교한 결과를 표 2에 나타내었다. 플랜트가 100번째부터 (50)과 같이 변화하는 경우에 대해 학습한 결과를 그림 17에 나타내었다. 플랜트의 변화 후에 바로 원하는 출력을 추정함을 결과그림을 통해 알 수 있었다. 실험결과로부터 제안된 학습률과 제어 학습 알고리듬은 비선형 시스템의 출력을 잘

표 2. 사례연구 2에 대한 15회 학습한 후 적합도와 오차의 실효치

Table 2. Fitness and mean square error for case study 2 after 15 training cycles.

Fitness, MSE (before Optimize)	Fitness	75.7576
	MSE	0.0132
Fitness, MSE (after Optimize)	Fitness	82.6446
	MSE	0.0121

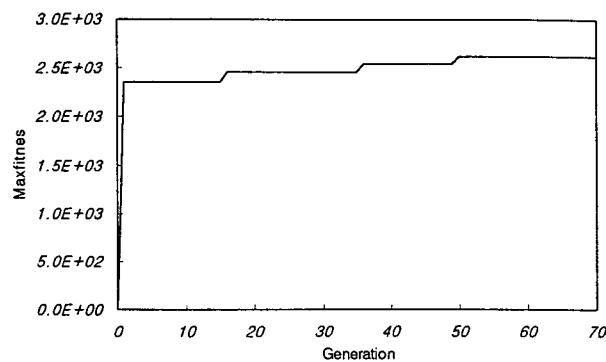


그림 6. SDNN의 유전알고리듬 적합도.

Fig. 6. Fitness of genetic algorithm for SDNN.

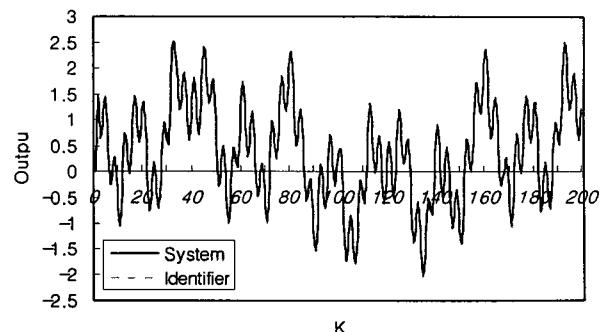


그림 7. 200쌍을 15회 학습 후 시스템 출력(y_p)과 식별기 출력(\hat{y}).

Fig. 7. Output of the system and identifier after 15 training cycles using 200 pairs.

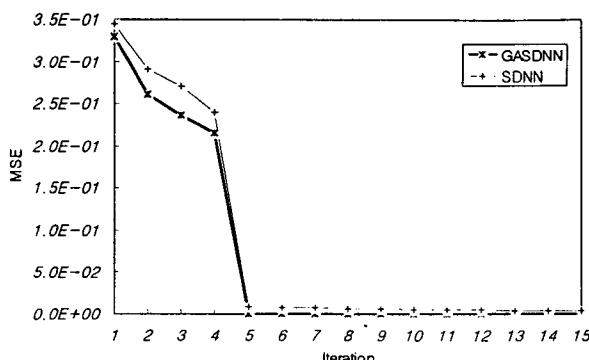


그림 8. 최적화된 SDNN 오차의 실효치 비교.

Fig. 8. Mean squared error comparison with optimized SDNN.

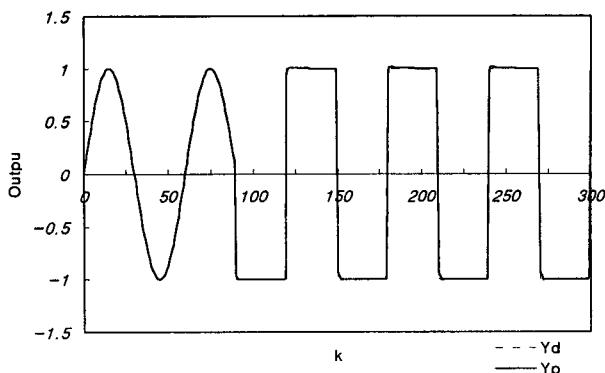


그림 9. 15회 학습한 후 시스템의 출력(y_p)과 기준 모형출력(y_d).

Fig. 9. Output of the system and reference input after 15 training cycles.

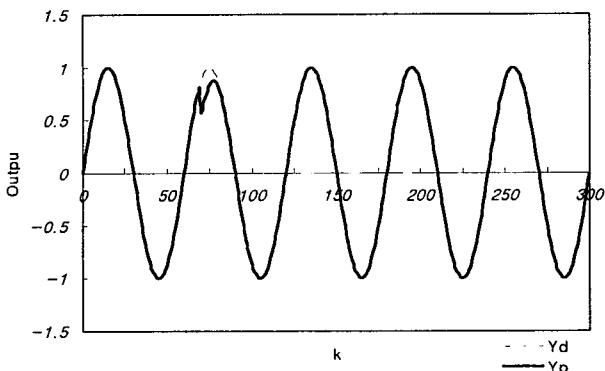


그림 10. 시스템 파라미터 a와 b가 70번째 샘플링순간에 a=0.8에서 a=0.5, b=1.2에서 b=0.8로 변했을 때의 제어성능(15회 학습 후).

Fig. 10. Control performance when system parameter a and b are varied 0.8 to 0.6 and 1.2 to 0.8 at 70-th sampling time (after 15 cycles).

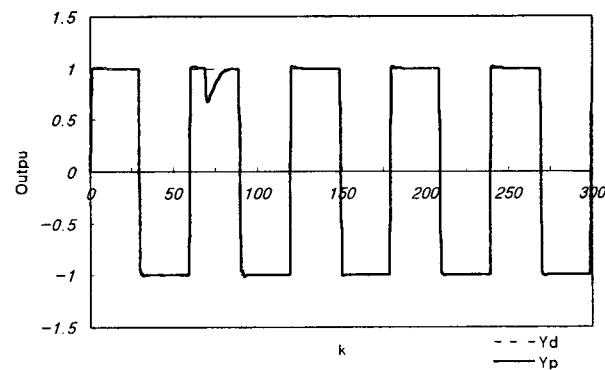


그림 11. 시스템 파라미터 a와 b가 70번째 샘플링순간에 a=0.8에서 a=0.5, b=1.2에서 b=0.8로 변했을 때의 제어성능(15회 학습 후).

Fig. 11. Control performance when system parameter a and b are varied 0.8 to 0.6 and 1.2 to 0.8 at 70th sampling time (after 15 cycles).

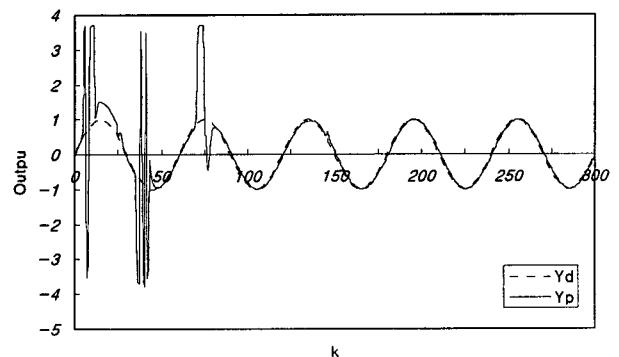


그림 12. 연결강도의 학습없이 초기화 후 온라인 제어시 출력 응답.

Fig. 12. Output response when on-line control without training of weight.

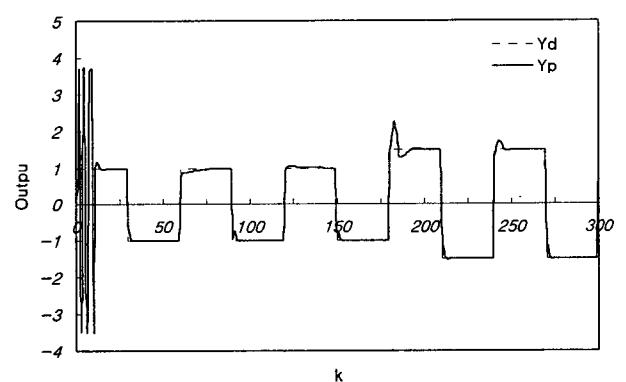


그림 13. 연결강도의 학습없이 초기화 후 온라인 제어시 출력 응답.

Fig. 13. Output response when on-line control without training of weight.

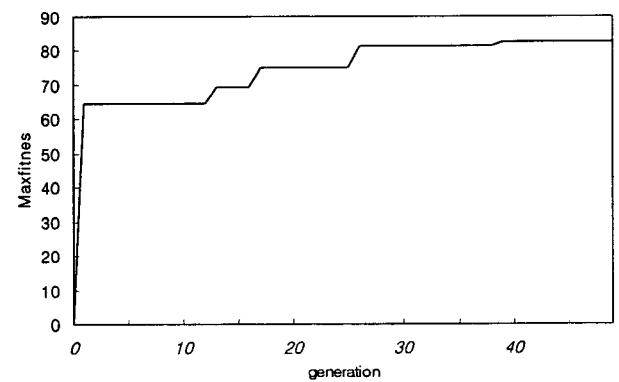


그림 14. SDNN의 유전알고리듬 적합도.

Fig. 14. Fitness of genetic algorithm for SDNN.

추정할 수 있으며 온라인으로 입의의 동적 시스템을 식별하고 제어하여 원하는 출력을 추정할 수 있음을 알 수 있었다. 그림 18은 (48)의 플랜트가 70번째 샘플링 순간에 (50)으로 변하고 210번째 샘플링순간에 다시 (48)로 변할 때의 제어성을 보여주고 있다.

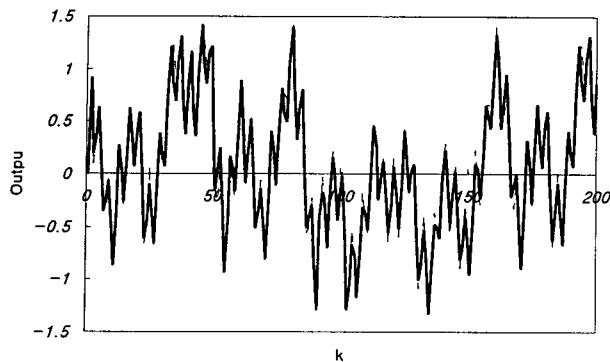


그림 15. 200쌍을 15회 학습 후 시스템 출력(y_p)과 식별 기(\hat{y}) 출력. Identifier : solid line, System : dashed line

Fig. 15. Output of the system and identifier after 15 training cycles using 200 pairs.

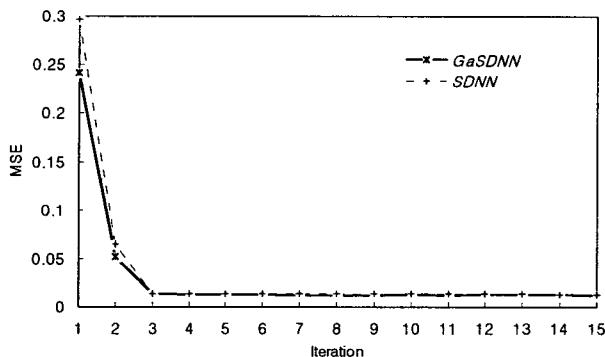


그림 16. 최적화된 SDNN 오차의 실효치 비교.
Fig. 16. Mean squared error comparison with optimized SDNN.

VII. 결론

동적 시스템을 제어하기에 적합한 신경망은 내부적으로 동적인 특성을 갖는 동적 신경망이다. 동적신경망은 연결강도의 수가 많으므로 적응 제어를 성공적으로 수행하기 위해서는 무엇보다도 학습속도가 빠른 신경망 구조가 필요하며 또한 안정성이 고려되어야 한다. 또한 신경망 제어기의 노드 수와 활성화 함수의 기울기는 시스템의 출력에 직접적인 영향을 미치므로 신중을 기해야 한다. 그러나 기존의 신경망으로 구성된 제어기는 체계적인 방법이 제시되어 있지 않아 노드수와 활성화함수의 기울기를 결정하는데 어려움이 있다. 이러한 신경망제어기에 유전 알고리듬을 이용하여 최적화 하려는 연구가 되고 있다. 그러나 정적신경망에 적용하였으며[15], 안정성보다는 신경망 구조에 역점을 두었다[16]. 따라서 본 논문에서는 안정성을 고려한 연결강도수가 적은 자기 동적 신경망을 사용하여 유전 알고리듬으로 노드수와 활성화함수를 최적화 시키므로 비선형 시스템을 안정하게 제어할 수 있는 알고리듬을 제안한다. 유전 알고리듬은 세대를 거듭함에 따라 어떤 최적화 하는 해에 도달하는 탐색 알고리듬이다. 탐색 공간이 크거나 분석적으로 해를 찾을 수 없는 문제에 대해 유용한 알고리듬이다. 자기

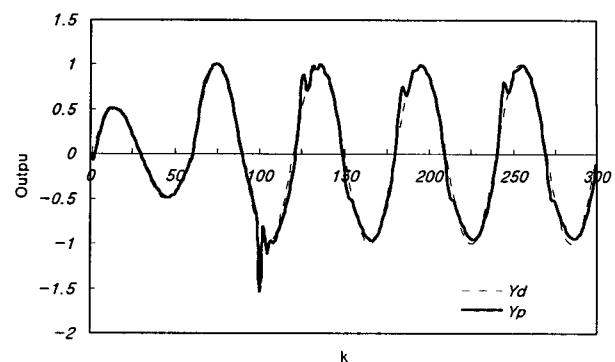


그림 17. 100번째 샘플링 순간에 시스템이 변화했을 때의 제어 성능(15회 학습 후).

Fig. 17. Control performance when system are varied at 100th sampling time (after 15 cycles).

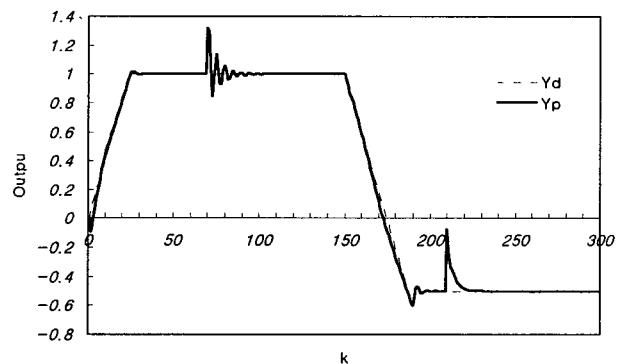


그림 18. 70번째와 210번째 샘플링 순간에 시스템이 변화했을 때의 제어 성능(15회 학습 후).

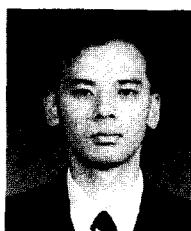
Fig. 18. Control performance when system are varied at 70th and 210th sampling time (after 15 cycles).

동적 신경망을 사용하므로써 학습속도가 빠르며, 또한 동적신경망의 연결강도의 수를 줄임에도 불구하고, 리아 푸노프 함수를 도입해 안정성이 고려된 학습률을 사용하여 학습시키므로써 오차의 수렴도 안정적임을 보였다. 제안된 알고리듬을 사용하여 최적화된 자기 귀환신경망으로 비선형 시스템의 식별과 제어를 위한 모의실험을 통해 학습 알고리듬의 안정성과 효용성을 확인하였다.

참고문헌

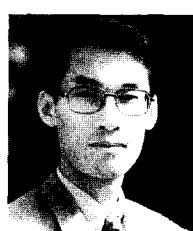
- [1] L. J. Elman, *Neural Networks for Control*, The MIT Press, 1990.
- [2] C. Unbehauen, *Neural Networks for Optimization and Signal Processing*, Wiley, 1992.
- [3] K. S. Narendra and K. Pathasarathy, "Gradient methods for the optimization of dynamical systems containing neural networks," *IEEE Trans. Neural Networks*, vol 2, pp. 252-262, Mar., 1991.
- [4] B. Widrow and M. A. Lehr, "30 years of adaptive neural networks: perceptron, madaline, and back-

- propagation," *Proc. IEEE*, vol. 78, no 9, pp. 1415-1441, 1990.
- [5] K. S. Narendra and K. Pathasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. Neural Networks*, vol. 1, pp. 4-27, 1990.
- [6] C. F. Chen, "Back-propagation neural networks for nonlinear self-tuning adaptive control," *IEEE Control System Magazine*, vol. 10, no. 3, pp. 44-48, Apr., 1990.
- [7] L. Jin, P. N. Nikiforuk and M. M. Gupta, "Dynamics and stability of multilayered recurrent neural networks," *Proc. 1993 IEEE Int. Conf. Neural Net*, Vol II, pp. 1135-1140, 1993.
- [8] George A. Rovithakis and A. Christodoulou, "Adaptive control of unknown plants using dynamical neural networks," *IEEE Trans. Systems, Man and Cybernetics*, vol. 24, no. 3, pp. 400-411, March, 1994.
- [9] S. I. Sudharsanan and M. K. Sundareshan, "Training of a three layer Dynamical recurrent neural network for nonlinear input-output mapping," *Proc. Int. Joint Conf. Neural Networks (IJCNN-91)*, vol. II, pp. 111-116, Seattle, WA 1991.
- [10] A. Karakasoglu, S. I. Sudharsanan and M. K. Sundareshan, "Identification and decentralized adaptive control using dynamical neural networks with application to robotic manipulators," *IEEE Trans. Neural Networks*, vol. 4, no 6, pp. 919-930, 1993.
- [11] L. Ljung, "Issue in system identification," *IEEE Contr. Syst. Mag.*, vol. 11, pp. 270-280, 1989.
- [12] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, 1989.
- [13] Z. Michalewicz., *Genetic Algorithms+Data Structures =Evolution Programs*, Springer-Verlag, 1994.
- [14] M. R. Cheng., *Genetic Algorithms and Engineering Design*, A Wiley-Interscience Publication, 1996.
- [15] E. P. Maillard G and Didier G "Designing an optimal RBF classifier using genetic algorithm," *Int. Neural Networks Society Annual Meeting*, vol. 2 pp. 1054-1058 San Diego 1996.
- [16] I. E. Ana and C. S. Kenneth "Genetic evolution of recurrent neural network architectures," *Int. Neural Networks Society Annual Meeting*, vol. 2 pp. 1059-1062 San Diego 1996.
- [17] K. S. Narendra, J. Balakrishnan and M. K. Ciliz, "Adaptation and learning multiple models, switching, and tuning," *IEEE Control. System Magazine*, vol. 15, no. 3, pp. 37-51, June, 1995.
- [18] M. M. Polycarpou and P. A. Ioannou, "Learning convergence analysis of neural-type structured networks," *IEEE Trans. Neural Networks*, vol. 3, no. 1, pp. 39-50, Jan., 1992.
- [19] C.-C. Ku and Y. L. Kwang, "System identification and control using diagonal recurrent neural networks," *Proc. ACC*, vol. 1, pp. 545-549, July, 1992.
- [20] P. S. Sastry, G. Santharam and K. P. Unnikrishnan, "Memory neuron networks for identification and control of dynamical system," *IEEE Trans. Neural Networks*, vol. 5, no. 2, pp. 306-319, Mar., 1994.
- [21] 이영석, 유동완, 서보혁 "기준 모델 추종 기능을 이용한 뉴로-퍼지 적응제어기 설계." 제어·자동화·시스템공학회 논문지, vol. 4 no. 2 pp. 203-208, 1997.



유 동 완

1995년 영남대 전기공학과 졸업. 경북대학교 대학원 석사(1997), 1997년-현재 경북대학교 전기과 박사과정. 관심분야는 시스템이론, 적응제어, 지능제어 등.



전 순 용

1986년 경북대 전기공학과 졸업. 동대학원 석사(1991), 동대학 박사(1995). 1990-1995년 한라중공업(주) 연구소, 1995-1996년 김천전문대 방사건과, 1996년-현재 동양대학교 제어계측공학과 전임강사/산업기술연구소장. 관심분야는 제어응용, 산업 시스템모델링, 뉴럴퍼지 응용, system identification.

서 보 혁

제어·자동화·시스템공학회 논문지 제4권, 제2호, 참조.