

귀환 신경망의 안정적 학습 알고리즘 개발

On Developing the Stable Learning Algorithm for Recurrent Neural Network Control System

연 정 흠*, 원 경 재*, 정 일 훈*, 전 흥 태*
J. H. Yon*, K. J. Won*, I. H. Jung*, H. T. Jeon*

※이 연구는 1996년도 중앙대학교 교내학술 연구비 지원에 의해 이루어 졌음.

요 약

비선형 동적 시스템을 제어하기에 적합한 귀환 신경망에 대한 주요한 연구중의 하나는 안정적인 학습 알고리즘을 개발하는 것이다. 본 논문에서는 진화 연산 알고리즘을 이용한 안정적인 귀환 신경망의 학습 알고리즘을 개발한다. 또한 개발한 학습 알고리즘을 사용한 귀환 신경망을 전형적인 비선형 동적 시스템인 로봇트 매니퓰레이터의 제어 시스템에 적용하고 개발한 제어 알고리즘의 효용성을 입증한다.

ABSTRACT

One of major research areas in the recurrent neural network is to develop stable learning algorithm. In this paper, the stable learning algorithm is developed by utilizing the evolutionary programming. The effectiveness of the proposed learning algorithm will be verified by simulating two d.o.f. robot manipulator.

I. 서 론

제어 시스템의 설계에 있어서 고도의 비선형 동적 시스템(highly nonlinear dynamic system)을 제어하기 위해서는 시스템의 동적 특성을 수학적으로 모델링(modeling)해야 한다. 그러나 복잡한 동적 특성을 갖는 시스템의 전달함수, 상태방정식 형태의 정확한 수학적 모델링이 어렵거나 시스템 파라메타(parameter)의 변동, 또는 외부 외란이 존재하는 환경에서는 기

존의 제어 이론으로 강건한(robust) 제어기 구성이 용이하지 않다.

이러한 문제점을 극복하기 위하여 인간의 사고능력과 적응능력을 갖는 지능형 제어기의 연구가 활발히 진행되고 있다. 이중 대표적인 것이 신경망(neural networks)을 이용한 제어 구조이다. 신경망을 이용한 제어기는 일반적으로 시스템의 특성을 학습한 후 그 정보를 이용하여 제어기를 구성하는 데 입력과 출력의 정보로부터 입·출력 관계를 학습(learning)함으로써 자동적으로 지식 습득이 가능하다. 그러므로 정확한 수학적 모델링이 필요치 않으며 신경망의 병렬 처

*중앙대학교 전자공학과

리(parallel processing) 특성으로 인하여 실시간 처리(realtime processing) 및 시스템 변화에 강건한 제어가 가능하다.

신경망의 학습방법에는 정적 전방향 신경망을 이용하는 방법과 동적 귀환 신경망을 이용하는 방법이 있는데 정적 전방향 신경망은 구조상 정적 사상(static mapping)만을 학습할 수 있으므로 제어기에 시스템의 입력을 tapped delay line으로 구성하거나, 시스템의 출력을 신경망의 입력으로 피드백(feedback)하는 등의 변형이 필요한 반면 동적 귀환 신경망은 내부적으로 상태 피드백(state feedback)이 있기 때문에 입·출력 사이에 동적 사상(dynamical mapping)을 학습할 수 있다. 그러므로 제어기 구성에 정적 전방향 신경망과 같은 변형은 필요하지 않기 때문에 내부적으로 상태 피드백이 있어 동특성을 갖는 동적 귀환 신경망이 정적 전방향 신경망보다 비선형 동적 시스템을 제어하기에 적합하다.

Pineda와 Pearlmutter가 귀환 신경망에 역전파 학습을 적용하여 학습 규칙을 개발하였는데 이러한 학습 알고리즘은 뉴런의 상태가 이미 평형점(equilibrium point)으로 수렴했다는 제약 조건하에서 연구되었다. 하지만 이 제약 조건이 만족하지 않을 경우 수렴성을 보장하지 못하고 지역적 최소치(local minima) 문제 또한 해결하지 못한다. 또한 Sato와 Uchikawa 등은 수렴에 대한 전제 조건없이 최적 제어 이론의 한 방법인 Lagrange Multiplier를 이용하여 학습 규칙을 개발하였다. 하지만 이 방법은 전체 궤적을 저장하여야 하기 때문에 온라인(on-line)학습이 불가능하며 많은 저장공간을 필요로 한다는 단점이 있다.

본 논문에서는 귀환 신경망 뉴런의 상태가 이미 평형점으로 수렴했다는 전제 조건이 없는 동시에 동적 시스템 제어기 설계에 이용할 수 있는 온라인(on-line) 학습이 가능한 귀환 신경망의 학습 알고리즘을 개발을 위하여 최근 최적 문제 해결의 한 분야로 각광을 받고 있는 진화 프로그래밍(Evolutionary Programming: EP)을 이용한다. 이 진화 프로그래밍 알고리즘은 기존의 유전자 알고리즘과 더불어 자연진화론을 바탕으로한 집단에 기초한 최적화 과정이다. 이 알고리즘은 비교적 단순한 연산과정을 사용하며 단일 해가 아닌 해집단을 이용하므로 기존의 최적화 기법들이 안고 있는 국소 최적점에 수렴할 가능성을 배

제할 수 있다. 진화 프로그래밍을 사용하여 뉴런의 최적의 가중치를 선택하는데 이 방법은 수학적이고 확실적인 정보를 필요로 하는 기존의 방법보다 훨씬 효과적일 수 있다. 상기의 학습 알고리즘을 사용한 귀환 신경망을 전형적인 비선형 동적 시스템인 매니플레이터의 관절 변수(joint variable) 제어 시스템에 적용하고 기존의 학습 방법의 적용 결과와 비교하여 개발한 학습 알고리즘의 효용성을 입증한다.

II. 귀환 신경망의 구성

2.1 연속 귀환 신경망의 구성

정적 전방향 신경망(static forward neural networks)의 뉴런은 입력에 가중치를 곱한 다음 그것들의 합을 활성화 함수에 통과시켜 출력을 구한다. 반면 동적 귀환 신경망(dynamic recurrent neural networks)의 뉴런은 상태 변수의 상태방정식이 비선형 미분 방정식 형태로 정의하여 출력항을 계산한다.

귀환 신경망은 입력층과, 귀환층과 출력층으로 구성되며, 정적 전방향 다층 신경망과 비교할 때 은닉층(hidden layer)이 귀환층(recurrent layer)으로 대체된다. 이와 같은 귀환 신경망은 그림 1과 같은 구조를 갖는다.

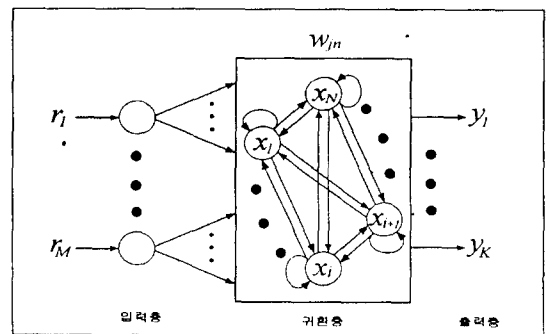


그림 1. 연속 귀환 신경망의 구성

Fig. 1 Constructure of continuous Recurrent Neural networks

귀환 신경망의 귀환층에 있는 i번째 뉴런의 상태 방정식과 출력 방정식은 다음과 같다.

$$\dot{x}_i(t) = -\gamma_i x_i(t) + \sum_{j=1}^N w_{ji} f(x_j(t)) + \sum_{j=1}^M r_j(t) \quad (1)$$

$$y_k(t) = x_k(t) \quad (k = 1, \dots, K; K \text{는 출력개수}) \quad (2)$$

$x_i(t)$ ($i = 1, \dots, N; N$ 은 뉴런 개수)는 뉴런의 현재 상태이며, γ_i 는 뉴런의 시상수(time constant)이고, $r_j(t)$ ($i = 1, \dots, M; M$ 은 입력 개수)는 뉴런으로 들어오는 외부의 입력이다. 또한 w_{ji} 는 뉴런들 사이에 가해지는 가중치이다. 비선형 함수 $f(\cdot)$ 는 뉴런의 활성화 함수로 미분 가능하고 시스템에 적당한 동적 성질을 주는 함수인 하이퍼볼릭 탄젠트(hyperbolic tangent; $\tanh(\cdot)$) 함수를 일반적으로 사용한다. 또한 귀환 신경망의 출력 방정식은 식 (2)와 같이 각 뉴런의 상태 변수로 구해진다.

귀환 신경망에서 제어목적과 제어대상에 따라 귀환 신경망의 파라메타 w_{ji} 는 학습 규칙에 의해 조정되어야 한다. 이러한 학습 규칙은 Pineda와 Pealmutter가 역전파 학습 규칙을 확장하여 개발하였다.

2.1.1 역전파 학습 규칙의 확장

기존의 다층 신경망에서 학습에서 사용된 오차 역전파 학습 규칙은 오차 함수를 최소화하도록 가중치를 조정하는 감독 학습 규칙(supervised learning rule)이다.

$$E(\mathbf{w}) = \frac{1}{2} \sum_{k=1}^K (\hat{x}_k - x_k)^2 = \frac{1}{2} \sum_{k=1}^K E_k^2 \quad (3)$$

여기서 \mathbf{w} 는 신경망의 모든 가중치 집합을 나타내며 \hat{x}_k 와 x_k 는 각각 신경망의 목표 출력치(target output value)와 실제 출력치(actual output value)를 나타내는데 최급 하강법(steepest descent method)인 델타 규칙(delta rule)을 이용하여 가중치 변화량을 구할 수 있다.

이 오차 역전파 학습 규칙을 귀환 신경망의 학습 규칙으로 확장하기 위하여 식 (1)에서 뉴런의 상태 x_i 가 점근적으로 안정한 평형점 x_i^* 가 적어도 하나 존재한다고 가정한다.

(1)식에서 주어진 귀환 신경망의 상태방정식에서 안정한 평형점이 존재한다는 가정에 의하여 $\frac{dx_i}{dt} = 0$ 으로 놓으면 다음의 (4)식과 같다.

$$x_i^* = \sum_{j=1}^N w_{ji} f(x_j^*) + \sum_{j=1}^M r_j(t) \quad (4)$$

여기서 가중치 w_{ji} 를 구하기 위해 최급 하강법인 델타 규칙을 적용한다.

$$\begin{aligned} \Delta w_{ji} &= -\rho \frac{\partial E}{\partial w_{ji}} = -\rho \frac{\partial E}{\partial x_k^*} \frac{\partial x_k^*}{\partial w_{ji}} \\ &= (-\rho) \left(\sum_{j=1}^N -(\hat{x}_k - x_k^*) \right) \frac{\partial x_k^*}{\partial w_{ji}} \\ &= \rho \sum_{j=1}^N E_k^* \frac{\partial x_k^*}{\partial w_{ji}} \end{aligned} \quad (5)$$

$$z_i^* = \sum_j f'(net_i^*) w_{ji} z_j^* + E_i^* \quad (6)$$

그래서 확장된 역전파 학습에 의한 귀환 신경망의 학습 규칙은 다음 식과 같다.

$$\Delta w_{ji} = \rho f'(net_i^*) z_i^* x_j^* \quad (7)$$

ρ 는 오차 역전파 학습 규칙의 학습률이며, z_i^* 는 귀환 역전파(recurrent backpropagation)로 정의되는 미분 방정식 (8)에 의하여 구해진다.

$$\frac{dz_i}{dt} = -z_i + \sum_j f'(net_i^*) w_{ji} z_j^* + E_i^* \quad (8)$$

2.2 이산 귀환 신경망의 구조와 학습 규칙

연속 귀환 신경망에서의 뉴런 상태 방정식은 미분 방정식 형태로 주어지는데 이때 뉴런의 상태 x_i 가 시간 t 에 의한 변화가 매우 적다고 가정한다면 이산 귀환 신경망의 뉴런 상태 방정식과 출력 방정식은 Euler

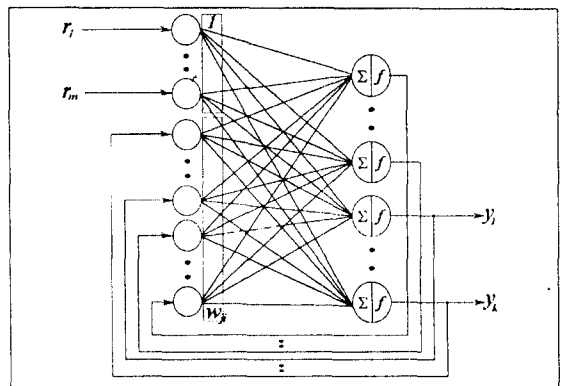


그림 2. 이산 귀환 신경망의 구조
Fig. 2 Structure of Discrete recurrent neural networks

근사화에 의하여 각각 (9), (10)식과 같이 구해진다. (시상수 γ_i 는 1로 고정)

$$x_i(t+1) = \sum_{j=1}^N w_{ji} f(x_j(t)) + \sum_{j=1}^M r_j(t) \quad (9)$$

$$y_i(t+1) = f(x_i(t+1)) \quad (10)$$

즉 연속 귀환 신경망에서의 미분방정식 형태의 상태방정식과 출력방정식이 이산 귀환 신경망에서 이산 방정식으로 표현할 수 있다는 것을 의미한다. 이산 귀환 신경망의 학습 알고리즘은 교사 학습(supervised learning)으로 특정 시간에 신경망의 출력치를 고정된 목표치와 일치하도록 것을 의미한다. 학습 알고리즘을 위한 오차 함수는

$$E(t) = \frac{1}{2} \sum_{k=1}^K \{(d_k(t) - y_k(t))\}^2 = \frac{1}{2} \sum_{k=1}^K e_k(t)^2 \quad (11)$$

로 정의된다. 여기서 목표치는 $d_k(t)$ 이고 신경망의 출력값은 $y_k(t)$ 이다.

$p_{ji}^k(t) \equiv \partial y_k(t) / \partial w_{ji}$ 로 정의하면 다음의 이산 상태방정식에 의해 $p_{ji}^k(t)$ 를 구할 수 있다.

$$p_{ji}^k(t+1) = f'(x_k(t)) \left\{ \sum_{l=1}^N w_{lk} p_{ji}^l(t) + \delta_{ki} x_j(t) \right\} \quad (12)$$

(12)식에서 상태의 초기치는 $p_{ji}^k(t_0) = 0$ 으로 놓는다.

이 역전과 규칙에 의해 오차 함수가 감소하도록 하는 가중치 수정 규칙은 식 (13)와 같다.

$$\Delta w = \eta \sum_{k=1}^K e_k(t) p_{ji}^k(t) \quad (13)$$

III. 진화 프로그래밍을 이용한 귀환 신경망 제어기의 설계

3.1 진화 프로그래밍

최근에 전역적 최적화(global optimization) 알고리즘으로 각광을 받고 있는 진화 프로그래밍(Evolutionary Programming)을 이용하여 귀환 신경망을 학습시킨다. 여기서 최적화 기법으로 적용하고자 하는 진화 연산 알고리즘은 기존의 유전자 알고리즘(Genetic Algorithm)과 더불어 자연 진화론을 바탕으로 한

집단에 기초한 최적화 과정으로서 공학적 목적에 유용하게 적용할 수 있는 알고리즘이다. 유전자 알고리즘이 이진코드(binary code)를 사용하기 때문에 많은 계산량, 복잡한 연산과정을 갖는 반면에 진화 프로그래밍은 실수값을 이용하므로 적은 계산량, 그리고 비교적 단순한 연산과정으로 동작한다는 장점을 가지고 있다. 그리고 진화 프로그래밍은 단일 해가 아닌 해집단을 이용하므로써 기존 최적화 기법들이 안고 있는 국소 최적점(local optimum)에 수렴할 가능성을 배제할 수 있다. 진화 연산 알고리즘의 전체 연산과정은 돌연변이, 재생산, 선택, 경쟁의 4가지 과정으로 요약할 수 있다.

개략적인 연산과정은, 먼저 초기 부모(parents) 집단을 무작위적으로 선정하고 각각의 해에 주어진 평가함수의 결과치(score)를 할당한다. 자손(offspring) 집단은 부모 집단을 돌연변이시켜 생성한다. 즉, 각 해의 구성요소는 평균(mean)이 0이고 부모 해의 결과치에 비례하는 분산(variation)을 가진 가우시안 랜덤 변수(Gaussian random variable)에 의해 변이된다. 최근에는 자기 적응적 표준편차를 사용하여 좋은 결과를 보여주고 있다. 부모 집단과 생성된 자손 집단을 확률적 토너먼트(probabilistic tournament)에 의한 경쟁을 통해 확률적으로 우수한 결과치를 가진 해들을 다음세대의 부모 집단으로 구성한다. 이 과정을 만족할 만한 결과치를 얻을 때까지 반복수행한다.

예를 들어 어떤 함수 $F(x): R^n \rightarrow R$ 를 최소화하는 실수해를 찾는 문제에 적용한 기본적인 진화 프로그래밍 알고리즘은 다음과 같다.

[단계 1] 각차원마다 일정한 범위 내에서 임의로 부모 벡터들($x_i, i=1, \dots, P$)의 집단을 초기화한다.

[단계 2] 각부모 벡터 x_i 에 Gaussian random variable과 미리 선택한 표준편차를 사용한 돌연변이 방법으로 자손 벡터들($x_i', i=1, \dots, P$)을 생성한다.

[단계 3] $F(x_i)$ 와 $F(x_i')$ 의 오차 비교를 통해 적은 오차를 가진 벡터들을 확률적으로 선택하여 다음 세대의 부모 벡터로 구성한다.

[단계 4] 만족할 만한 값에 도달하거나 일정한 값에 수렴할 때까지 위 [단계 2]와 [단계 3]을 반복 실행한다.

상기 [단계 2]에서의 돌연변이 과정은 다음과 같다.

부모 집단 벡터중 하나를 x_i 라 하고, 이 벡터가 n 차원 일 때, 돌연변이 연산자는 벡터를 구성하고 있는 각 구성원 즉, 각 차원에 모두 적용되며 다음의 식(14)으로 표현된다.

$$\begin{aligned} \sigma_i' &= \sigma_i \exp(\tau' \cdot N(0, 1) + \tau \cdot N_i(0, 1)) \\ x_i' &= x_i + N(0, \sigma_i') \quad (i=1, \dots, n) \end{aligned} \quad (14)$$

여기서, x_i : 최적화될 목적 변수에 대한 실수치 벡터
 σ_i : x 의 각 차원에 대한 표준편차 벡터
 (x_i', σ_i') : 다음 세대의 새로운 벡터
 $N(0, 1)$: 표준편차 1을 가진 0 평균 가우시안 random variable
 $N_i(0, 1)$: i 번째 독립적으로 분산된 표준 가우시안 random 변수
 τ, τ' : step-size를 결정하는 파라메타

이 돌연변이 방법을 사용함으로써 진화 알고리즘은 오차의 폭에 대해, 보다 효율적으로 자기 적응(self-adapt)을 할 수 있다. 위 돌연변이 방법은 부모 세대와 자손 세대간의 큰 변화보다는 작은 변화가 더 자주 일어난다는 생물학적 발견에 근거를 두고 있다.

그리고, [단계 3]에서의 다음 세대 구성방법은 자연계의 적자 생존을 모방하여 이루어지는데, 이는 경쟁(competition)과 선택(selection)의 두 가지 과정으로 구성된다. 진화 프로그래밍 알고리즘에서의 경쟁 방법은 평가 함수에 의해 결정되며 선택의 방법에서는 평가 함수에 의한 결과가 가장 좋은 것들을 순서대로 다음 세대로 구성하는 방법과 확률적으로 좋은 것들을 선택하는 방법 등 두가지 경우로 나눌 수 있다. 본 논문에서는 후자를 적용하며 경쟁 및 선택 알고리즘은 다음과 같이 요약할 수 있다.

- i) 부모 세대의 각 벡터 x_i ($i=1, \dots, P$)와 임의로 선택한 k 개의 x_j ($j \neq i, j < P, k < P$)와 평가 함수 결과치를 비교한다.
- ii) k 번의 비교중 $F(x_i) \leq F(x_j)$ 인 횟수를 판단한다.
- iii) ii)에서 결정한 횟수가 가장 많은 순서로 자손 세대를 선택 생성한다.

3.2 진화 프로그래밍과 귀환 신경망

본 논문에서 제안하는 귀환 신경망의 학습 알고리즘은 뉴런 상태가 안정한 평형점으로 수렴하였다는 전체 조건을 가지지 않으며 온라인 학습이 가능하도록 하는 학습 알고리즘이다. 이러한 학습 알고리즘을 개발하기 위하여 위에서 설명한 진화프로그래밍 이론을 도입한다.

귀환 신경망의 가중치가 시스템 제어에 막대한 영향을 주기 때문에 가중치를 최적화 인자로 삼고 귀환 신경망의 가중치를 실수값의 스트링으로 나타낸다. 이때 스트링의 내용이 진화에 의해 변화되는데 이것이 귀환 신경망의 가중치의 변화를 나타낸다. 또한 적합도(fitness) 값은 스트링으로 표현되어 있는 귀환 신경망의 가중치가 적절한가를 나타낸다.

평가함수(evaluation function)는 귀환 신경망 제어기의 성능을 평가하는 기준이 되는 함수로 스트링의 적합도를 측정하는 기준이 된다.

제어 목적을 최소 오차와 최소 오차 변화율로 삼고 식(15)과 같이 평가 함수를 설정한다.

$$F(s_i) = K_1 \times \sum e + K_2 \times \sum \dot{e} \quad (15)$$

여기서 s_i 는 i 번째 스트링이고 K_1, K_2 는 상수이다.

식(15)를 최소화하는 진화 알고리즘을 통해 최적화 과정이 이루어진다. 스트링과 적합도 함수를 결정하는 다음 남은 과정은 최상의 적합도 값을 가진 스트링을 구하는 일이다. 우선 귀환 신경망의 가중치를 나타내는 스트링 N 개를 임의로 난수 발생에 의해 만들어 초기 집단을 구성한다. 초기 집단을 구성한 다음 집단 내의 각 스트링은 평가함수에 의해 적합도 값을 결정한다. 이 스트링 집단은 돌연변이 방법을 통하여 새로운 스트링을 형성하게 되고 기존의 스트링 집단과 새 스트링 집단의 적합도를 비교하여 확률적으로 적합도가 높은 스트링들을 다음 세대의 부모 집단으로 구성한다. 즉 주어진 환경내에서 적합도가 높은 스트링이 다음 세대의 구성원으로 남을 확률을 높여 주는 것이다. 다시 말해서 보다 높은 적합도 값을 가진 스트링이 더 높은 선택 확률을 주어 재생에 많이 참여시켜 다음 세대를 구성한다. 이와 같은 과정을 반복 수행하여 조정 매개변수 즉, 가중치의 최적점 또는 최적영역을 찾는다. 진화가 진행되어 가는 동안 높은 적합도 값을 가진 스트링이 증가하며 정상상태에서

최상의 적합도를 가진 스트링이 얻어진다. 여기에서 구해진 최상의 적합도를 가진 스트링 내부의 가중치는 제어 목적에 가장 부합하는 스트링으로 귀환 신경망 제어기의 초기 파라미터가 된다.

3.3 진화 프로그래밍을 이용한 제어기 설계

이러한 제어 시스템의 구조는 피드백 학습(feedback learning) 방법과 목표 지향 학습(goal-oriented learning) 방법을 사용하는데, 또한 전방향(forward)형의 제어 구조로서 병렬로 연결되어 있는 RNN과 선형 제어기(PD controller), 그리고 제어 대상 시스템(plant)으로 구성되어 있다. 여기서 RNN은 적용 학습을 통해 제어 대상 시스템의 역동력학 특성을 모델링하여 기준 입력에 대한 제어 입력을 생성한다. 그리고 선형 제어기는 순간적인 제어환경의 변화 등에 대하여 시스템의 안정성을 보장한다. PD 제어기는 입력과 출력의 오차인 $|X_d - X_p|$ 를 최소화 시키기 위한 시스템의 입력을 대략적으로 결정한다.

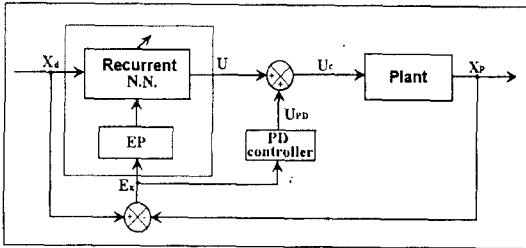


그림 3. 진화 프로그래밍에 의한 귀환 신경망 제어기
Fig. 3 Recurrent Neural networks controller constructed by evolutionary programming

상기의 그림 3과 같은 제어 시스템의 동작은 다음과 같은 단계로 이루어진다.

- [단계 1] 시스템의 초기 출력, 목표 출력, 귀환 신경망의 가중치를 초기화
- [단계 2] 시스템의 오차를 통해 PD 제어기의 출력을 산출
- [단계 3] PD 제어기의 출력으로 귀환 신경망을 학습
- [단계 4] 시스템의 오차를 이용하여 진화프로그래밍을 통한 귀환 신경망의 적합도 평가
- [단계 5] 귀환 신경망의 출력과 PD 제어기의 출력의

합을 시스템에 인가

$$U_c(t) = U(t) + U_{PD}(t) = U(t) + K_p e(t) + K_d \dot{e}(t) \quad (16)$$

[단계 6] 시스템의 출력이 원하는 상태로 도달하면 중지

IV. 컴퓨터 모의실험

본 논문에서 제안한 귀환 신경망 제어 시스템의 효용성을 입증하기 위하여 로봇트 매니플레이터의 관절 변수(joint variable)의 제어 시스템에 적용한다.

전형적인 비선형 동적 시스템인 로봇트 매니플레이터의 동적 방정식은 다음의 식(17)으로 나타내어진다.

$$\tau = \mathbf{M}(\theta) \ddot{\theta} + \mathbf{V}(\theta, \dot{\theta}) + \mathbf{G}(\theta) \quad (17)$$

- 여기서, $\theta \in \mathbb{R}^{n \times 1}$: 관절 벡터(joint vector)
- $\tau \in \mathbb{R}^{n \times 1}$: 토크 벡터(torque vector)
- $\mathbf{M}(\theta) \in \mathbb{R}^{n \times n}$: 관성행렬(inertia matrix)
- $\mathbf{V}(\theta, \dot{\theta}) \in \mathbb{R}^{n \times 1}$: 원심력(centrifugal force)과 코리올리 힘 벡터(coriolis force vector)
- $\mathbf{G}(\theta) \in \mathbb{R}^{n \times 1}$: 중력 벡터(gravitational vector)

그림 4와 같은 평면 2축 로봇트 매니플레이터의 위치 제어에 적용하여 귀환 신경망의 효율성을 입증한다. 이때 이 로봇트 매니플레이터의 파라메타를 표 1과 같이 설정하고 주행 경로는 표 2와 같은 타원형 궤적을 설정한다.

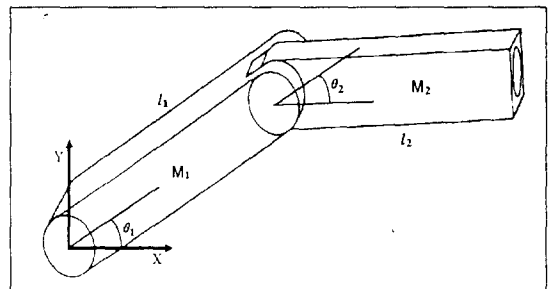


그림 4. 평면 2축 로봇트 매니플레이터
Fig. 4.2 Link Planar Robot Manipulator

표 1. 평면 2자유도 로봇 매니플레이터의 링크 파라메타
Table 1. The Link Parameter of the 2 Link Robot Manipulator

변 수	Link 1	Link 2
링크 길이	0.53 m	0.47 m
링크 무게	3.0 kg	2.5 kg

표 2. 로봇 매니플레이터의 타원형 궤적에 대한 주행 경로
Table 2. Desired Trajectory of Robot Manipulator

위 치	주행 경로
$P_x(t)$	$0.3 + 0.25 \cos(t)$ m
$P_y(t)$	$0.3 + 0.15 \sin(t)$ m
$P_z(t)$	0.0

그림 5은 정적 신경망을 이용한 제어기로 2000회 학습한 후 궤적을 나타낸 그림이고, 그림 8은 동적 귀환 신경망을 이용한 제어기로 2000회 학습한 후 궤적

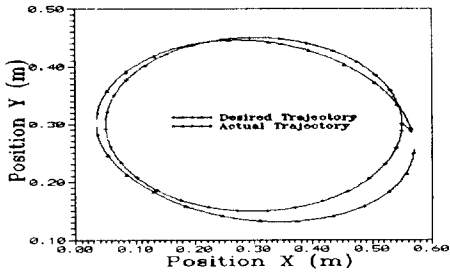


그림 5. 전방향 신경망을 이용한 궤적 학습 결과
Fig. 5 Learning Result of the Desired Trajectory Using the Forward Neural Network

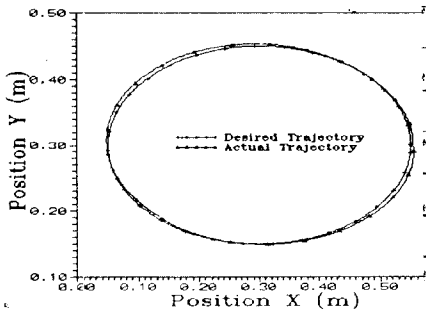


그림 6. 귀환 신경망을 이용한 궤적 학습 결과
Fig. 6 Learning Result of the Desired Trajectory Using the Recurrent Neural Network

을 나타낸 그림이다.

그림 (7)와 그림 (8)는 이러한 두 신경망의 학습 효율성을 보이기 위하여 2000회 학습하는 동안 관절 변수에 대한 오차의 변화를 나타낸 그림이다. 이 그림에서 보는 바와 같이 학습 속도와 학습 결과에서 귀환 신경망이 정적 전방향 신경망보다 로봇 매니플레이터를 효율적으로 제어함을 알 수 있다.

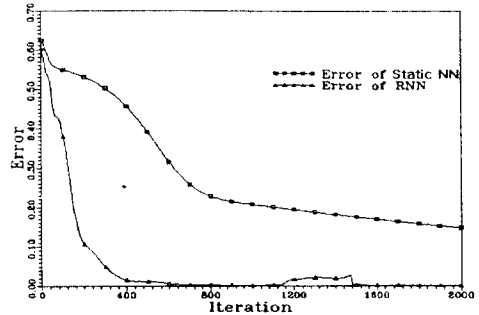


그림 7. 링크 1에 대한 관절 변수 오차의 변화
Fig. 7 Error Variation of the Joint Variable of Link 1

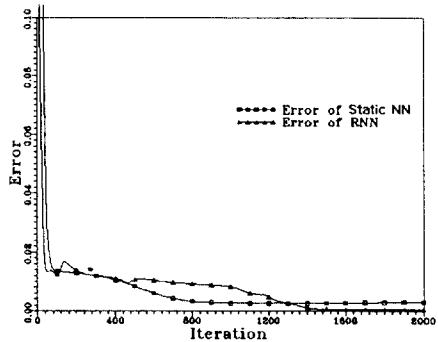


그림 8. 링크 2에 대한 관절 변수 오차의 변화
Fig. 8. Error Variation of the Joint Variable of Link 2

V. 결 론

비선형 동적 시스템을 제어하는데 적합한 귀환 신경망은 연구가 되어왔다. 하지만 학습 알고리즘 개발에 전제로한 가정, 즉 뉴런의 상태가 이미 안정한 평형점으로 수렴하였다는 것이 맞지 않을 경우 귀환 신경망의 뉴런 상태의 수렴성을 유지할 수 없는 문제가 발생한다. 따라서 본 논문은 뉴런이 이미 안정한 평

형점으로 수렴하였다는 전제 조건 없이 온라인 학습이 가능한 귀환 신경망의 학습 알고리즘을 개발하였다. 개발한 학습 알고리즘은 최근 최적화 알고리즘으로 각광을 받고 있는 진화 프로그래밍을 이용하였다. 또한 본 논문에서 평면 2축 로보트 매니플레이터의 위치 제어를 통해 귀환 신경망이 전방향 신경망보다 효율적이라는 증명함으로써 효용성을 입증하였다.

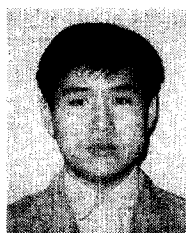
참 고 문 헌

1. M. Kawato, Y. Uno, M. Isobe, R. Suzuki, "Hierarchical Neural Network Model for Voluntary Movement with Application to Robotics," *IEEE Control Syst. Mag.*, pp. 8-16, Apr. 1988.
2. S. Nagata, M. sekiguchi, K. Asakawa, "Mobile Robot Control by a Structured Hierarchical Neural Network," *IEEE Control Syst. Mag.*, pp. 69-76, Apr. 1990.
3. Jeffrey L. Elman, *Neural Networks for Control*, The MIT Press, 1990.
4. K. S. Narendra, and K. Patharathy, "Identification and Control of Dynamical System Using Neural Networks," *IEEE Trans. on Neural Networks*, Vol. 1, No. 1, pp. 4-27, 1990.
5. T. Yabuta, and T. Yamada, "Neural Network Controller Characteristics with regard to Adaptive Control," *IEEE Trans. on SMC*, Vol. 22, pp. 170-177, 1992.
6. S. Haykin, *Neural Networks*, A Comprehensive Foundation, IEEE Press, 1994
7. K. S. Narendra and K. Parthasarathy, "Gradient Method for the Optimization of Dynamical Systems Containing Neural Networks," *IEEE Trans. on Neural Networks*, Vol. 3. No. 1, Jan, 1992.
8. A. Karakasoglu, S. I. Suharsanan, and M. K. Sundareshan, "Identification and Decentralized Adaptive Control Using Dynamical Neural Networks with Application to Robotic Manipulators," *IEEE Trans on Neural Networks*, vol. 4, no. 6, pp. 919-930, 1993.
9. George A. Rovithakis, and A. Christodoulou, "Adaptive Control of Unknown Plants Using Dynamical Neural Networks," *IEEE Trans. on SMC*, Vol 24, No. 3, pp. 400-411, March., 1994.
10. Quan N., and T. J. Sejnowski, "Learning to Solve Random-Dot Stereograms of Dense Transparent Surfaces with Recurrent Backpropagation," *Proc. of the 1988 Connectionist Models Summer School*, pp. 435-443, San Mateo, CA.
11. A. Guez, V. Protopopsecu, J. Barhen, "On the Stability, Storage Capacity, and Design of Non-linear Continuous Neural Networks," *IEEE Trans. on SMC*, Vol. 18, No. 1, pp. 80-87, Jan./Feb., 1988.
12. Liang Jin, Peter N. Nikiforuk, and Madan M. Gupta, "Dynamics and Stability of Mutilayered Recurrent Neural Networks," in *Proc. 1993 IEEE Int. Conf. Neural Net.*, 1993. Vol. II, pp. 1135-1140.
13. F. J. Pineda, "Recurrent Backpropagation and the Dynamical Approach to Adaptive Neural Computation," *Neural Computation 1*, pp. 161-172 1989.
14. B. A. Pearlmutter, "Learning State Space Trajectories in Recurrent Neural Networks," *Neural Computation 1*, pp. 263-269 1989.
15. L. B. Almeida, "A Learning Rule for Asynchronous Perception with Feedback in a Combinatorial Environment," *1987 Proc. of IEEE First Int. Conf. Neural Networks*, Vol. II, pp. 609-618, 1987.
16. R. J. Williams, and D. Zipser, "A Learning Algorithm for Continually Running Fully Recurrent Neural Networks," *Neural Computation 1*, pp. 270-280 1989.
17. M. Sato, "A Real Time Learning Algorithm for Recurrent Analog Neural Networks," *Biological Cybernetics*, Vol. 62, pp. 237-241, 1990.
18. K. Gouhara, T. Wadanabe, Y. Uchikawa, "Learning Process of Recurrent Neural Networks," *Proc. IJCNN '91 Singapore*, Vol. 1, pp. 746-751, 1991.
19. Y. Uchikawa, H. Takase, T. Watanabe, K. Gouhara, "Recurrent Neural Networks," *Fifth IFSA*

World Congress, vol. 2, pp. 1005-1008, 1993.

20. V. I. Utkin, *Sliding Modes and Their Application in Variable Structure System*, Mir Publishers, MOSCOW, 1978.
21. V. I. Utkin, *Sliding Modes in Control Optimization*, Springer-Verlag, Heidelberg, 1992.
22. U. Itkis, *Control Systems of Variable Structure*, John Wiley & Sons, INC., New York, 1976.
23. D. E. Rumelhart, G. E. Hinton and R. J. William, "Learning Internal Representations by Error Propagation," *Parallel Distributed Processing*, the MIT press, pp. 318-362, 1986.
24. K. Matsuoka, "Stability Conditions for Nonlinear Continuous Neural Networks with Asymmetric Connection Weights," *Neural Networks*, Vol. 5, pp. 495-500, 1992.
25. Kar-Keung D. Young, "Controller Design for a Manipulator Using Theory of Variable Structure System," *IEEE Trans on SMC-8*, pp. 101-109, Feb., 1978.
26. Dorato, P., and A. Kestenbaum, "Application of game theory to the sensitivity design of optimal systems," *IEEE Trans. Automatic Control AC-12*, pp. 85-87, Feb., 1967.
27. Utkin, V. I. "Equation of sliding mode in discontinuous systems I, II," *Automation and Remote Control*, pp. 211-219, 1972.
28. Subramania I. Sudharsanan and Malur K. Sundareshan, "Equilibrium Characterization of Dynamic Neural Networks and a Systematic Synthesis Procedure for Associative Memories," *IEEE Trans on Neural Networks*, pp. 509-521, vol. 2, no. 5, Sep., 1991.
29. A. Balestrino, G. De Maria and L. Sciavicco, "An Adaptive Model Following Control for Robotic Manipulators," *Journal of Dynamic Systems, Measurement and Control*, vol. 105, pp. 143-151, Sep., 1983.
30. Liang Jin, Peter N. Nikiforuk and Madan M. Gupta, "Approximation of Discrete-Time State Trajectories Using Dynamic Recurrent Neural

Networks," *IEEE Trans on Automatic Control*, vol. 40, no. 7, pp. 1266-1270, July, 1995.



延 貞 欽(J. H. Yon) 정희원
1970年 9月 23日生
1994年 3月: 중앙대학교 전자공학과 졸업(학사)
1996年 8월: 중앙대학교 대학원 전자공학과 졸업(석사)
1996年 9월~현재: 중앙대학교 대학원 전자공학과 박사과정 재학중

사과정 재학중

※주관심분야: 퍼지, 신경망, 퍼지 뉴럴 융합, 유전 알고리즘 등임



元 敬 勳(K. J. Won) 정희원
1973年 5月 27日生
1996年 2월: 중앙대학교 전자공학과 졸업(학사)
1996年 2월~현재: 중앙대학교 대학원 전자공학과 석사과정 재학중

※주관심분야: 퍼지, 신경망, 퍼지 뉴럴 융합, 유전 알고리즘 등임



鄭 日 薰(I. H. Jung) 정희원
1974年 6月 20日生
1997年 2월: 중앙대학교 전자공학과 졸업(학사)
1997年 3월~현재: 중앙대학교 대학원 전자공학과 석사과정 재학중

※주관심분야: 퍼지, 신경망, 퍼지 뉴럴 융합, 유전 알고리즘 등임

全 洪 兌(H. T. Jeon)

정희원

현재: 중앙대학교 전자공학과 교수