

신경회로망을 이용한 비선형 시스템 제어의 실험적 연구

Experimental Studies of Neural Network Control Technique for Nonlinear Systems

정 슬, 임 선 빈
(Seul Jung and Sun Bin Yim)

Abstract : In this paper, intelligent control method using neural network as a nonlinear controller is presented. Simulation studies for three link rotary robot are performed. Neural network controller is implemented on DSP board in PC to make real time computing possible. On-line training algorithms for neural network control are proposed. As a test-bed, a large x-y table was build and interface with PC has been implemented. Experiments such as inverted pendulum control and large x-y table position tracking control are performed. The results for different PD controller gains with neural network show excellent position tracking for circular trajectory compared with those for PD controller only. Neural control scheme also works better for controlling inverted pendulum on x-y table.

Keywords : neural network, back propagation algorithm, on-line control, nonlinear system, x-y table robot, inverted pendulum

I. 서론

신경회로망은 비선형 함수의 맵핑 능력 및 학습 능력이 있어 제어 분야뿐만 아니라 영상 및 음성신호처리 또는 통신 등 다양한 분야에서 사용되고 있다. 제어 분야에서의 신경회로망은 시스템 자체의 불확실성과 부정확성을 해결하기 위해 자체 알고리즘 안에서 오차를 줄여 가는 쪽으로 제어 출력을 변화시킴으로써 정확한 제어를 가능하게 한다. 또한 on-line 제어라는 틀 안에서 그 자체로서의 제어기 역할이 아닌 기존의 PID 또는 PD 제어기와 더불어 보조 제어기 역할로 사용되어 안정상태에서의 오차를 영으로 수렴하게 하는 역할을 한다[1][2]. 응용된 대상들은 비선형 공정으로서 로봇, 모터, 원격 조정 해저 탐사기, 그리고 공작 기계 등 다양하다[3]-[7]. 이러한 공정들을 제어할 수 있는 정확한 모델을 실제적으로 얻기 어려우므로 모델을 기반으로 하는 제어방식을 사용하여 정확한 제어를 하기란 어려운 일이다. 설령, 어떤 공정의 정확한 동적 방정식을 얻을 수 있다 하더라도 불확실성이 항상 존재하기 때문에 만족한 결과를 얻을 수 없다. 따라서, 이러한 불확실성을 내포한 대략적인 동적 방정식이 주어질 경우에 신경회로망을 이용하여 불확실성을 보상하는 방식이 제안되었다[8][9]. 이 경우에 신경회로망의 목적은 불확실성만을 보상하는 것으로 국한되어 신경회로망이 부분적으로 사용되었다.

하지만, 신경회로망을 이용하는 궁극적인 목적은 제어할 대상에 대하여 어떤 정보도 필요하지 않고 제어할 수 있게 하는 것이다. 제어할 대상의 모델에 대한 지식이 없이도 제어할 수 있다면, 신경회로망의 잠재적인 능력을 완전히 사용하게 되는 것이다. 이러한 기본 테두리 안에서 여러 신경

망제어 방식들이 제안되었는데, 그 중에서도 그림 1에 보여진 역델 인식을 모태로 한 직접 제어 방식이 기본을 이룬다 [10]. 신경망을 학습하기 위해 역전파 알고리즘을 사용할 경우에 학습신호의 설정이 매우 중요하다. 학습신호를 어떻게 설정하는가에 따라 시스템 자코비안 문제를 해결해야만 되기도 하고 할 필요가 없게도 된다[10][11]. 시스템 자코비안을 해결하는 방법에는 크게 두 가지 방법이 있다. 하나는 수치적으로 구하는 방법이 있고, 다른 하나는 또 다른 신경회로망을 사용하여 구하는 방법이 있다. 하지만 가장 좋은 방법은 자코비안이 필요 없는 경우인데 대표적인 방식으로 잘 알려진 귀환오차 학습방식(Feedback error learning)이 있다 [11]. 최근에는 귀환오차 학습방법의 일종으로 시스템의 안정성, 강건성, 그리고 passivity에 근거하여 로봇을 제어하는 방식을 제안하였다[2].

이 논문에서는 신경회로망 제어 방식의 하나인 입력 경로에 보상하는 입력보상방식을 사용한다. 입력보상방식은 자코비안 정보가 필요하지 않고 신경회로망의 출력을 입력에 보상하여 시스템의 비선형성을 제거하는 제어 방식의 일종이다. 첫째로, 제어할 대상이 PD(Proportional and derivative) 제어기로 먼저 안정된 후에 보상이 되기 때문에 안정성이 높다. 그림 1에 나타난 것처럼 역모델 제어의 가장 기본 구조인 직접 제어 방식은 안정성이 취약하다. 실시간 제어일 경우에 신경회로망에서 무슨 신호가 출력되는지 보장할 수

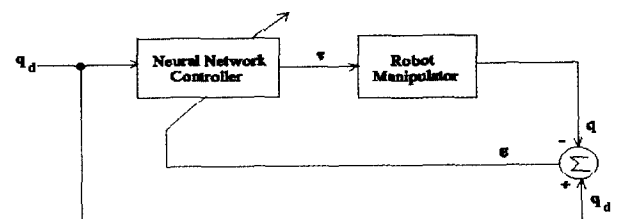


그림 1. 신경회로망 역제어 방식.
Fig. 1. Neural network inverse control scheme.

접수일자 : 2001. 3. 19., 수정완료 : 2001. 7. 10.
정 슬, 임선빈 : 충남대학교 메카트로닉스공학과(jungs@cnu.ac.kr/s_isbn@cnu.ac.kr)
※ 본 연구는 과학재단 핵심연구 981-0911-051-2의 지원 하에 이루어졌음을 알려드리며 이에 감사드립니다.

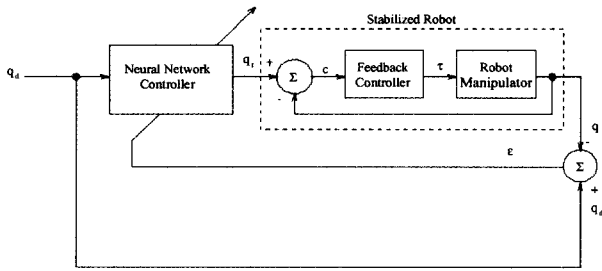


그림 2. 입력보상 신경회로망 역제어 방식.
Fig. 2. Reference compensation NN control scheme.

없으므로 로봇의 움직임에 대한 안정성을 보장할 수 없다. 그림 2에서는 신경 제어기가 PD 제어기로 미리 안정된 공정의 역모델을 인식하므로 제한한 제어 방식의 안정성을 나타낸다. 둘째로, 그림 2에서 보여진 것처럼 불확실성을 보상하는 위치가 제어하는 공정 밖에서 이루어지기 때문에 기존에 설치되어 있는 실제 시스템들의 제어기들을 수정할 필요 없이 제한한 방식을 적용하기가 쉬운 이점이 있다.

이 논문에서는 역전과 알고리즘을 실시간 제어 시스템에 적용하기 위해 학습 신호를 적당하게 설계하여 실시간 운용이 가능하도록 하였다. 또한 계산량이 많은 역전과 알고리즘 계산을 실시간으로 하기 위해 DSP 보드를 장착한 PC 제어 시스템을 구축하였다. 실험 장치로는 크기가 매우 큰 x-y 테이블 로봇 시스템을 제작하여 원경로를 실험하여 성능을 조사하였다. 또한 x-y 테이블 위에 역진자를 장착하여 역진자의 각과 카트의 위치를 제어하였다. 실험 결과 신경회로망을 사용할 경우 성능이 매우 우수함을 알 수 있었다.

II. 로봇의 동적 모델

일반적인 n관절(joint)의 로봇 동적 모델은 다음과 같다.

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + \tau_f(\dot{q}) = \tau \quad (1)$$

여기서 $D(q)$ 는 $n \times n$ 관성(inertia) 행렬, $C(q, \dot{q})\dot{q}$ 는 $n \times 1$ 벡터의 Coriolis와 원심력, $G(q)$ 는 $n \times 1$ 벡터의 중력, $\tau_f(\dot{q})$ 는 $n \times 1$ 벡터의 마찰력이고, τ 는 $n \times 1$ 벡터의 구동 토크이다. 또한 q 는 $n \times 1$ 벡터의 각(角)위치이고 \dot{q} 는 $n \times 1$ 벡터의 각(角)속도, \ddot{q} 는 $n \times 1$ 벡터의 각(角)가속도이다. 로봇의 안정성을 보장하는 가장 간단한 PD 제어 법칙은 다음과 같이 표현된다.

$$\tau = K_D(\dot{q}_r - \dot{q}) + K_P(q_r - q) \quad (2)$$

여기서 q_r 과 \dot{q}_r 는 제어기로 직접 입력되는 $n \times 1$ 벡터의 주어진 기준 경로(Reference trajectory)이고, K_D 와 K_P 는 $n \times n$ 제어기 이득 행렬이다.

오차벡터 e 와 \dot{e} 를 아래와 같이 정의하고, (1)과 (2)를 연립하면 다음과 같다.

$$K_D\dot{e} + K_P e = D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + \tau_f(\dot{q}) \quad (3)$$

여기서 $\dot{e} = \dot{q}_r - \dot{q}$, $e = q_r - q$ 이다. 그림 3에 PD제어기로 제어된 시스템이 부분적으로 잘 나타나있다.

(3)에서 양변에 \ddot{q}_d 를 더하고 정리하면 다음과 같이 이차 오차 방정식으로 유도될 수 있다.

$$\ddot{e} + D^{-1}K_D\dot{e} + D^{-1}K_P e = D^{-1}(D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + \tau_f(\dot{q})) \quad (4)$$

(4)에서 보면 로봇의 각 움직임에 따라 방정식이 달라지고 각 관절의 방정식이 서로 연계되어 있으므로 제어하기가 매우 어렵다. 일반적으로 PD제어기를 사용할 경우에 큰 제어기 이득값을 사용하면 로봇의 경로 수행 성능을 향상시킬 수는 있으나 비선형 불확실성을 보상하기 어려울 뿐만 아니라 안정성에 영향을 끼칠 수 있다. 따라서 낮은 이득을 가지고 역모델 인식과 비선형 불확실성을 보상할 수 있는 방식이 필요한데, 그 방식은 다음 장에서 소개한다.

III. 신경회로망 제어 방식

1. 입력보상 제어 방식 I

그림 3에는 입력보상 제어 방식의 구조가 잘 나타나 있는데, 이를 제어 방식 I이라 하자. 그림 3에서처럼 신경 제어기는 제어 회로 밖에서 선 필터를 사용하여 보상을 실행한다. 이 신경 제어기의 역할은 구조상 직접 역제어 방식처럼 역모델을 인식하는 것이지만 직접 역제어 방식처럼 공정 대상 그 자체의 역모델을 인식하는 것이 아니라, 제어기로 제어된 공정의 역모델을 인식한다는 데 현저한 차이가 있다.

그림 3의 제어구조를 살펴보면 제어기 입력 신호 \dot{q}_r, q_r 은 신경회로망의 출력으로 경로 오차와 신경회로망의 내부가중치에 따라 매번 변경됨을 알 수 있다. 보상 신호, 즉 신경회로망의 출력은 실제 출력 신호 q, \dot{q} 와 비교되어 오차 신호 e, \dot{e} 를 만든다. 이 오차 신호는 제어기의 이득을 통하게 되므로 증폭되어 직접 공정을 제어할 제어 입력 신호(예를 들면 토크)를 형성한다. 그림 3에서처럼 원하는 경로(Desired trajectory) \dot{q}_d 과 q_d 를 (3)의 양변에 더해서 새로운 오차 변수 ϵ 으로 구성된 새로운 오차 방정식을 구하면 다음과 같은 관계식을 얻는다.

$$K_D\dot{\epsilon} + K_P \epsilon = D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + \tau_f(\dot{q}) + K_D\dot{q}_d + K_P q_d - (K_D\dot{q}_r + K_P q_r) \quad (5)$$

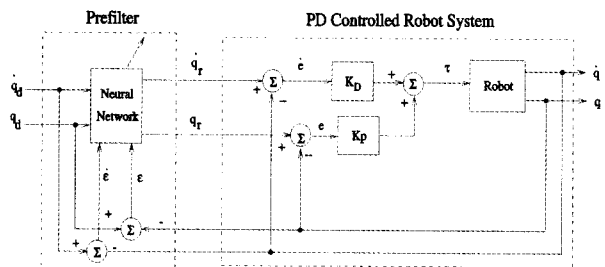


그림 3. 제어방식 I: 입력보상 신경회로망 역제어 방식.
Fig. 3. Scheme I: reference compensation NN control.

여기서 $\dot{\epsilon} = \dot{q}_d - \dot{q}$, $\epsilon = q_d - q$ 이고, 제어 입력 신호 q_r , \dot{q}_r 는 신경회로망의 출력과 같게 되어서 $\dot{q}_r = \phi_d$, $q_r = \phi_p$ 정의한다. 신경회로망의 출력 ϕ_d 와 ϕ_p 는 각각 $n \times 1$ 의 벡터이다.

(5)를 간단히 하기 위해서 신경망의 출력을 다음과 같이 벡터의 변수를 지정하자.

$$\Phi_1 = K_D \phi_d + K_P \phi_p, \tag{6}$$

그러면 (5)는 다음과 같이 간단하게 쓸 수 있다.

$$K_D \dot{\epsilon} + K_P \epsilon = \tau + K_D \dot{q}_d + K_P q_d - \Phi_1 \tag{7}$$

(7)에서 오차 방정식이 영(零)이 되려면 신경 제어기가 이상적으로 보상해야 할 신호 Φ_1 는 다음과 같다.

$$\Phi_1 = \tau + K_D \dot{q}_d + K_P q_d \tag{8}$$

이 경우에 있어서 수렴후의 신경 제어기는 로봇의 역모델을 인식하는 것이 아니라 PD 제어기로 제어된 폐쇄된 회로 시스템의 역모델을 인식하게 된다.

2. 입력보상 제어방식 II

앞에서 제안한 제어 방식 I를 더 간편하게 만들 수 있다. (3)에서 신경 제어기를 그림 4와 같이 다소 변경함으로써 $K_D \dot{q}_d + K_P q_d$ 를 간단하게 제거할 수 있는데, 이를 제어방식 II라 하자. 제어방식 II의 제어구조를 살펴보면 원하는 경로 \dot{q}_d, q_d 가 신경망제어기의 입력이 될 뿐만 아니라 보상하는 신호(ϕ_d, ϕ_p)에 더해지는 것을 볼 수 있다. 따라서 더해진 뒤의 기준 경로는 다음과 같다.

$$\dot{q}_r = \dot{q}_d + \phi_d, q_r = q_d + \phi_p \tag{9}$$

(9)를 (3)에 각각 대입하면 다음과 같다.

$$K_D(\dot{q}_d + \phi_d - \dot{q}) + K_P(q_d + \phi_p - q) = \tau \tag{10}$$

(10)을 오차 ϵ 의 식으로 정리하면 아래와 같다.

$$K_D \dot{\epsilon} + K_P \epsilon = \tau - \Phi_2 \tag{11}$$

여기서 신경망제어기의 출력 $\Phi_2 = K_D \phi_d + K_P \phi_p$ 이다.

(11)에서 제어방식 I의 (7)과 비교해 보면 $K_D \dot{q}_d + K_P q_d$ 가 없어진 것을 알 수 있다. 따라서 제어방식 II에서 보상해야 할 불확실성이 제어방식 I보다 작아지므로 신경망제어기의 부담이 적어지게 된다. 이상적으로 수렴이 되었을 때($\epsilon = 0$ 일 때) 신경회로망은 다음 식을 만족하는 로봇 동력학의 역 모델이 된다.

$$\Phi_2 = \tau \tag{12}$$

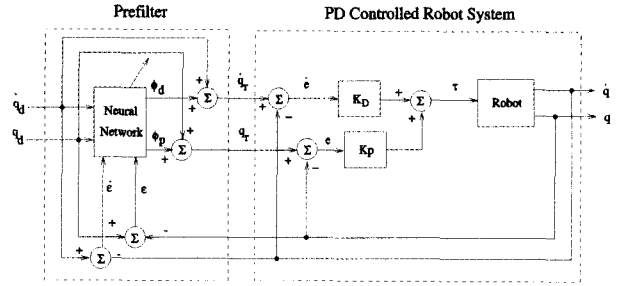


그림 4. 구조 II : 신경회로망 입력 보상 방식.
Fig. 4. Scheme II : reference compensation technique NN inverse control structure.

마찬가지로 대부분의 비선형 시스템은 (11)과 같은 형태로 만들 수 있게 되므로 제어방식 II와 같이 신경회로망을 사용하면 역동력학을 제어하게 된다.

이러한 특징 때문에 앞서 제어 방식 II는 역제어의 한 방식이라 할 수 있다. 제어방식 II의 경우에는 제어방식 I 경우보다 신경 제어기가 보상해야 할 불확실성이 작아서 신경 제어기가 일을 적게 하게 되므로 성능이 좋아지게 되는데, 이러한 사실은 V장의 모의 실험을 통해서 증명된다.

제어 방식 II를 구조적으로 보면, 보상 신호가 제어기 이득을 통해 증폭하기 때문에 작은 보상값, 즉 신경 제어기의 작은 출력을 가지고도 보상 신호를 증폭하는 큰 효과를 얻는다는 것이 본 제어 방식의 큰 이점이라 할 수 있다. 또한 숫자 정보 처리 과정(numerical information processing)시스템으로서의 신경회로망은 그 내부에서 진행되는 가중값과 비선형 영역(e.g. -1,+1 또는 0,1)에 따라서 그 성능이 좌우되는 특성이 있기 때문에 일반적으로 입력값의 정규화 과정과 비정규화 과정이 필요하지만, 앞서 제안한 제어 방식은 일반적으로 신경 제어기의 입력과 출력의 작동 영역이 작기 때문에 이러한 과정들이 필요하지 않다.

IV. 신경 제어기 구조와 학습 방법

1. 신경제어기의 구조

이 논문에서 사용된 신경망제어기는 입력층(input layer)-은닉층(hidden layer)-출력층(output layer)의 구조를 가진 두 겹 선형(two layer feedforward)식의 표준 신경회로망이다. 그림 5에서 보여진 것처럼 준비기(buffer)로 작용하는 입력층, 비선형인 은닉층, 그리고 선형인 출력층으로 구성된 신경망 제어기는 N_I 가 입력의 수, N_H 가 잠재 신경의 수, 그리고 N_o 가 출력의 수일 경우 전체의 가중치의 개수는 $W_T = (N_I + 1)N_H + (N_H + 1)N_o$ 가 된다. 입력 값은 입력층과 은닉층사이의 가중값(w_{ij})과 곱해지고, 은닉층에서 다시 더해진 뒤 활성화 함수(activation function)를 통하게 된다. 이 활성화 함수는 비선형으로 s자 함수라고도 불리는데, 그 값이 -1부터 1 사이에 존재하는 함수를 사용하였다.

$$f(x) = \frac{1 - \exp(-x)}{1 + \exp(-x)} \tag{13}$$

활성화 함수(activation function)를 통한 값들은 다시 은닉층

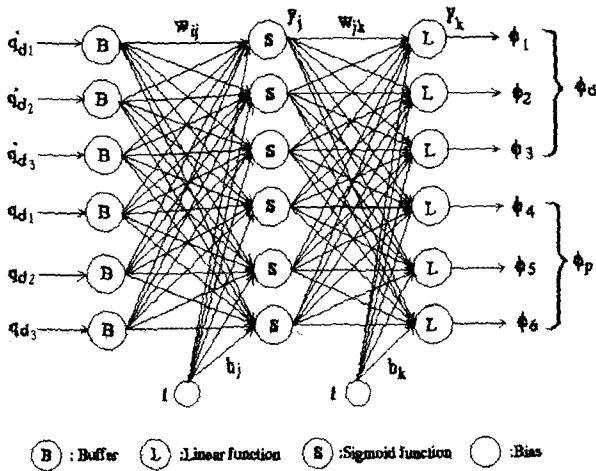


그림 5. 신경망제어기 구조.
Fig. 5. Neural network control structure.

과 출력층 사이의 가중값(w_{jk})과 곱해져서 각 출력층마다 (node)에서 더해져서 신경회로망의 출력(ϕ)이 된다. 신경망 제어기의 출력으로 선형이 사용되었으므로 출력층의 k 번째 출력 ϕ_k 는 다음과 같이 나타난다.

$$\phi_k = \left[\sum_{j=1}^{N_j} w_{jk} \frac{1 - \exp(-\sum_{i=1}^{N_j} x_i w_{ij} + b_j)}{1 + \exp(-\sum_{i=1}^{N_j} x_i w_{ij} + b_j)} \right] + b_k \quad (14)$$

b_j 는 j 번째 은닉층 신경의 편향 가중(biased weight)이고, b_k 는 출력층 k 번째 출력 신경(neuron)의 편향 가중인데 다른 가중들과 마찬가지로 매번 새롭게 바뀌게 된다.

2. 제어 방식 1의 학습 방법

이러한 가중값들은 오차 신호로 구성된 이차 함수인 목적함수 E 를 최소화시키면서 새로 얻어질 수 있다.

$$E = \frac{1}{2} \epsilon^T \epsilon \quad (15)$$

$$\epsilon = [\epsilon^T \quad \epsilon^T]^T$$

(15)를 신경회로망의 가중치 w 에 대해서 미분하면 E 의 그라디언트를 다음과 같이 얻을 수 있다.

$$\frac{\partial E}{\partial w} = \frac{\partial \epsilon^T}{\partial w} \epsilon \quad (16)$$

연쇄 법칙(chain rule)을 사용하여 (16)을 전개하면 다음과 같다.

$$\frac{\partial \epsilon^T}{\partial w} \epsilon = - \left[\frac{\partial q_r^T}{\partial w} \frac{\partial q_r^T}{\partial w} \right] J^T \epsilon \quad (17)$$

여기서 J 는 시스템 자코비안(Jacobian) 행렬로 다음과 같다.

$$J = \begin{bmatrix} \frac{\partial q}{\partial q_r} & \frac{\partial q}{\partial q_r} \\ \frac{\partial \dot{q}}{\partial q_r} & \frac{\partial \dot{q}}{\partial q_r} \end{bmatrix} \quad (18)$$

시스템 자코비안(Jacobian) 행렬은 다음의 이상적인 일차 식 으로부터 평가되어 얻어진다.

$$K_D(\dot{q}_r - \dot{q}) + K_P(q_r - q) = 0 \quad (19)$$

(19)로부터 J 는 다음과 같이 얻어진다.

$$J = \begin{bmatrix} \frac{\partial q}{\partial q_r} & \frac{\partial q}{\partial q_r} \\ \frac{\partial \dot{q}}{\partial q_r} & \frac{\partial \dot{q}}{\partial q_r} \end{bmatrix} = \begin{bmatrix} I & K_P^{-1} K_D \\ K_D^{-1} K_P & I \end{bmatrix} \quad (20)$$

3. 제어 방식 II의 학습 방법

먼저 학습 신호를 다음과 같이 정의한다.

$$v = K_D \epsilon + K_P \epsilon \quad (21)$$

목적함수는

$$E = \frac{1}{2} v^T v \quad (22)$$

(22)를 가중치에 대해 미분하면 목적함수 E 의 그라디언트를 다음과 같이 구할 수 있다.

$$\frac{\partial E}{\partial w} = \frac{\partial v^T}{\partial w} v \quad (23)$$

여기서 그라디언트 $\frac{\partial v^T}{\partial w}$ 는 (11)으로부터 구할 수 있다.

$$\frac{\partial v^T}{\partial w} = - \frac{\partial \Phi_2^T}{\partial w} \quad (24)$$

(16)과 (24)는 가중값들을 다음과 같이 새롭게 하기 위해 역전파(back-propagation) 알고리즘에 사용된다.

$$\Delta w(t) = -\eta \frac{\partial E}{\partial w} + \alpha \Delta w(t-1) \quad (25)$$

$$w(t+1) = w(t) + \Delta w(t) \quad (26)$$

여기서 η 값은 학습률이고, α 는 운동량(momentum) 상수이다.

V. 모의 실험 결과

이 장에서는 앞서 제안한 신경 제어 방식의 수행 능력을 모의실험을 통해 증명한다. 모의 실험을 위해 축이 셋인 수직 다관절 로봇을 이용하였다. 기본적인 동적 모델 방정식은 PUMA 560 로봇 팔의 처음 세 링크(Three-Link)로부터 구해졌다. 로봇의 동역학에 임의적인 불확실성을 주기 위해서 각 관절(joint)마다 다음의 마찰력을 더하였다.

$$\tau(\dot{q}) = 5.0 \text{sgn}(\dot{q}) + 8.0 \dot{q} \quad (27)$$

또한 10Kg의 물체가 로봇 팔 끝에 달려 있다고 가정하여

입의 불확실성을 부여하고자 한다. 이 장에서는 이 논문의 제어 방식이 이러한 임의적인 불확실성을 보상하는 지를 알아보기 위해 앞서 제안한 두 종류의 제어방식을 모의실험한다. 기존의 PD 제어기 이득은 작은 이득값으로는 $K_D = \text{diag}[40,40,40]$, $K_p = \text{diag}[100,100,100]$, 그리고 큰 이득값으로는 $K_D = \text{diag}[200,200,200]$, $K_p = \text{diag}[500,500,500]$ 들이 사용된다. 신경회로망 변수들로서 잠재 신경 수는 6, 모멘텀은 일반적인 값으로 $\alpha = 0.9$, 초기 가중값들은 -0.5와 0.5 사이의 범위에서 무작위로 선택하였고, 학습률 η 는 반복적인 실행에 의한 시행착오를 거쳐 최적값으로 정해졌다. 여기서 잠재 신경의 수는 뉴런의 개수를 바꾸어 가며 실행한 결과 오차의 최소값을 나타내는 최적값으로 설정하였다.

로봇의 작업은 주어진 경로, 즉 지름이 30cm이고 45도로 기울어진 원형 경로를 추종하는 것인데, 한 주기를 4초로 주었다. 표 1에서는 높은 이득을 사용했을 경우에 제어방식 I 과 II 두 제어기의 성능 및 신경 제어기로 보상하지 않을 경우의 성능을 비교한다. 제어기의 성능은 한 주기 동안에 생긴 경로 오차의 합으로 다음과 같이 정의되었다.

$$E_p = \sum_{i=1}^{PT} \|q_d - q\|^2, E_v = \sum_{i=1}^{PT} \|\dot{q}_d - \dot{q}\|^2$$

$$E_c = \sum_{i=1}^{PT} [(x_d - x)^2 + (y_d - y)^2 + (z_d - z)^2] \quad (28)$$

여기서 P 는 주기고, T 는 추출률(sampling rate)인데 4초와 0.005초가 각각 사용되었다. 그러므로 매 sampling 시간마다 신경회로망의 가중값들은 경로오차에 따라 새롭게 바뀐다. 표 1에 나타난 숫자들은 두 번째 주기, 즉 4초부터 8초까지의 오차의 합을 나타낸다. 학습률 η 는 각 구조마다 최적화되었다. 신경 제어기로 보상을 할 경우는 보상을 하지 않을 때(PD 제어기만 사용)보다 그 성능이 훨씬 우수함을 볼 수 있다. 또한 제어방식 II가 제어방식 I보다 우수한 성능을 보여준다.

작거나 큰 제어기 이득을 사용할 경우에 있어서 두 제어기의 수렴하는 시간과 정확도는 그림 6에서 잘 보여준다. 그림 6은 로봇이 주어진 원형 경로를 반복적으로 수행했을 때의 오차를 보여주는데 두 번째 반복할 때 이미 오차가 제로로 수렴하는 것을 볼 수 있다.

예상한 것처럼 큰 이득값의 제어기 사용시, 모든 제어기의 수행이 더 좋아짐을 볼 수 있는데, 어떤 크기의 이득값일 지라도 제어 방식 II의 성능이 제어 방식 I의 성능보다 우수

표 1. 두 번째 주기의 경로 오차(큰 이득을 사용할 경우).
Table 1. Second trajectory error(high gains).

	제어방식 I (Original)	제어방식 II (Modified)	보상되지 않은 경우
η	0.01	0.008	----
α	0.9	0.9	----
$E_p(\text{rad}^2)$	0.1292	0.0005	37.1022
$E_v(\text{rad/s})^2$	0.4678	0.1242	1.6018
$E_c(\text{m}^2)$	0.0265	0.0001	12.1274

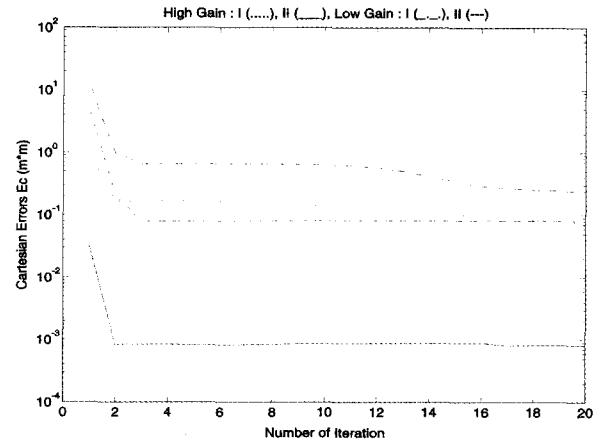


그림 6. 원경로 수행시 위치오차.
Fig. 6. Cartesian errors of circular trajectory.

함을 알 수 있다. 이 현상은 그림 7과 8을 통해서 더 분명하게 증명되는데, 그림 7과 8은 로봇의 첫 번째 주기의 경로 수행을 나타낸다. 두 경우에 있어서 제어 방식 II는 수렴이 빨리 된데 반해, 제어 방식 I은 수렴이 늦고 정확도가 떨어짐을 볼 수 있다. 또한 신경 제어기를 사용했을 경우의 성능이 사용하지 않았을 때보다 월등함을 보여 준다.

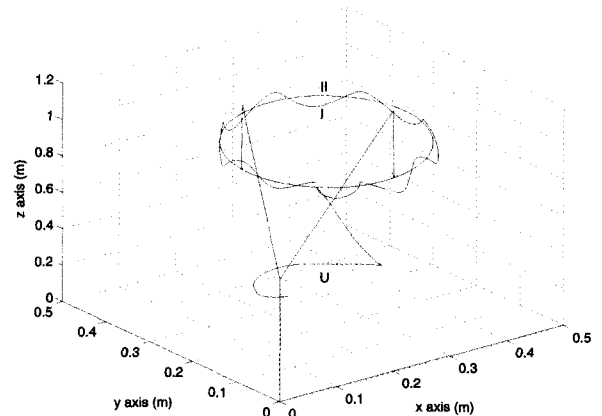


그림 7. 작은 제어기 이득의 원 경로 : I, II, PD 제어.
Fig. 7. Circular trajectory with low gains : I, II, U.

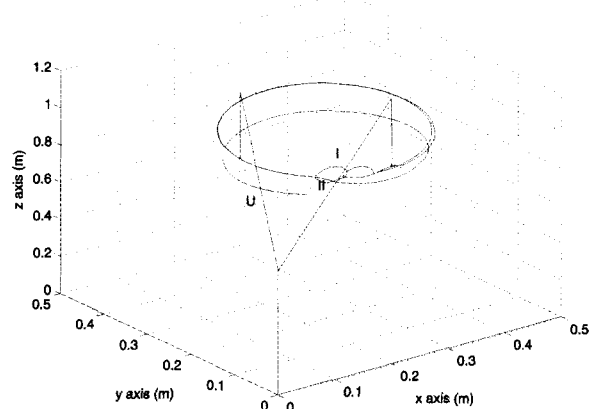


그림 8. 제어기 이득이 큰 경우의 원 경로 : I, II, PD 제어.
Fig. 8. Circular trajectory with high gains : I, II, U.

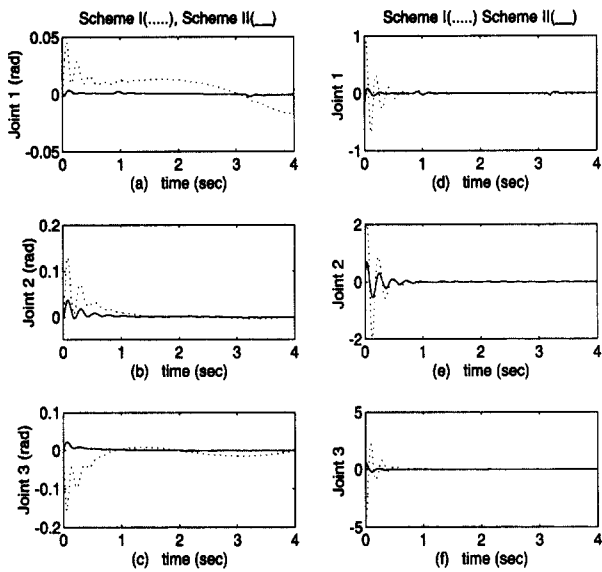


그림 9. 그림 8의 원경로 수행시 각 관절의 위치 오차.
Fig. 9. Joint error of circular trajectory shown in Fig. 8.

특히, 그림 8에서는 높은 이득의 제어기를 사용하는 이점을 단적으로 보여 주고 있는데, 제어 방식 I은 초기에 다소 경로 오차가 크지만 즉시 적응하여 제어 방식 II와 같이 주어진 경로를 잘 따라감을 볼 수 있다. 그림 9에서는 제어 방식 I 또는 II가 원형 경로를 얼마만큼 잘 따라 가는지를 경로 오차를 통해서 자세하게 보여지고 있다. 그림 9에서 보면 제어 방식 II 제어기의 성능이 위치 제어나 속도 제어에 있어서 제어 방식 I 제어기의 성능보다 훨씬 뛰어난 것을 알 수 있다. 또한 두 제어기 모두 1초 안에 수렴하되, 제어 방식 II 제어기가 더 빨리 수렴하는 것을 볼 수 있다. 제안한 신경망 제어기의 빠른 적응 능력은 실제 시간 제어(real time control)가 가능함을 보여준다. 보상하는 신호의 크기를 비교하면 제어 방식 II의 q_i 값이 방식 I보다 더 작는데, 이러한 특성은 신경 제어기의 일을 덜어 주게 되므로 (8)과 (12)의 차이에 의해 제어 방식 II의 성능을 향상시킨다.

VI. 전체 시스템 구성

전체 시스템은 크게 제어부, 구동부와 구조부로 나눌 수 있다. 제어부는 DSP 보드가 장착된 범용 PC 및 connector panel로 구성되고, 구동부는 DC 서보 모터와 드라이버로 구성되고, 구조부는 2단으로 형성되어진 X-Y table로 구성되어진다. 대략적인 전체 시스템 구성은 그림 10에 나타나 있다. 모터를 구동하기 위해서는 제어부의 DSP보드 상에 장착된 DAQ module에서 필요한 제어 신호를 내보내고 모터로부터의 엔코더 신호 피드백을 받아들인다. 이때 필요한 제어 알고리즘은 사용자가 PC를 이용하여 DSP에 전달한다. DSP 보드는 TMS320C30 칩을 사용하며 DAQ module로는 AD Converter 8채널과 DA Converter와 모터 구동을 위한 PWM 발생기 각각 4채널을 가지고 있어서 전체 시스템의 운영을 쉽게 하도록 되어 있다. 구조부인 x-y 테이블은 2.5m × 1.8 m 크기로 되어 있으며 1:10의 감속기를 사용하여 큰 토크로 구동된다. 구동 메커니즘은 한 쪽 모터가 쉬프트를

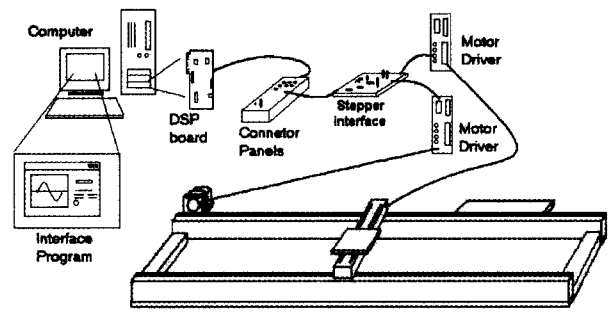


그림 10. 전체 시스템 구성.
Fig. 10. Overall structure.



그림 11. 실제 제작 x-y 테이블 로봇.
Fig. 11. X-Y table robot structure.

거쳐 타이밍 벨트에 의해 반대쪽에 동력을 전달하므로 불균형에서 오는 불확실성과 일정하지 않은 벨트의 장력에 의한 영향을 고려하여야 한다. 또한 한 축 위에서 다른 한 축이 움직이므로 그 축의 무게에 의한 관성도 고려해야 할 대상이다. 실제 제작된 x-y 테이블 시스템은 그림 11에 나타나 있다.

VII. 실험 방법 및 결과

1. x-y 테이블의 위치 제어

기본 제어 시스템이 되는 PD제어의 경우 몇 가지 선택적인 이득값을 추출하여 신경망을 학습시키에 따라 기준값과의 차이를 관찰한다. 사용된 x-y 테이블의 크기가 대략 2.5 (m) × 1.8(m) 정도의 큰 구조로 타이밍벨트로 구동되는 시스템이기 때문에 비선형 적인 요소가 크다.

실험에 사용되어진 PD이득값들의 크기로 비례이득값은 175, 미분 이득값은 3을 기준으로 하여 2배, 1/2배, 1/4배로 설정하여 각 이득값의 변화에 따른 위치오차를 비교하였다. 미분이득값의 경우는 구조상 유한 차분법에 의한 영향을 많이 받아 그 크기에 따라 진동이 심해지므로 작은 값을 사용하였다.

신경망의 경우는 온라인 학습을 위해서 일정한 패턴의 값들을 추종하도록 해 주어야 하는데 그러한 경우 속도나 위치의 급격한 변화가 있는 패턴보다는 일정한 비율을 가지고 증가 또는 감소하는 패턴을 이용하여 학습을 시켜야 한다. 이 논문에서는 X-Y 테이블이 직교 좌표계에서 0.5m 지름을 가지는 원운동을 하는 경로를 설정하여 학습을 시켰다.

표 2. 실험시 사용된 PD 이득값.

Table 2. PD gains for experiments.

	P Gain	D Gain
No 1	175	3
No 2	350	6
No 3	87	1.5
No 4	43.5	0.75

표 3. PD제어시 이득값에 따른 위치 오차.

Table 3. Position errors with different PD gains.

P Gain	D Gain	평균 오차 (m/cycle)
350	6	36.3
175	3	97.5
87	1.5	191.5
43.5	0.75	330.1

이는 각 축에서 볼 경우 Sinusoidal 형태의 Desired 위치를 가지게 되며 이를 PD이득값만을 이용하여 추종하도록 한 경우와 입력 보상 방식으로 신경망을 첨가하여 추종하도록 한 경우의 차이를 살펴보았다. 실험에 사용된 신경망의 구조는 입력층, 중간층, 출력층의 세 가지 층으로 이루어져 있으며 각각 입력층 6개 중간층 5개 출력층 2개의 뉴런의 형태로 이루어져 있다. 뉴런의 비선형 함수로는 (13)과 같은 Tangent hyperbaric 함수를 사용하였다. 샘플링 시간은 200Hz 정도로 신경회로망의 역전파 알고리즘을 계산하여 제어를 하였다. 또한 반복적인 원경로 추종을 통해 신경망을 학습하였다.

기본적인 PD 제어기만을 사용하였을 경우 이득값에 따라 오차의 정도가 크게 달라지게 된다. 위치 추종 성능은 이득값 설정상 주로 비례이득값의 영향을 상대적으로 많이 받게 된다. 표 3은 PD제어기만을 이용하여 원을 추종하도록 하였을 경우, 표 2에서 설정했던 각 이득값에 따른 오차의 평균치를 나타낸다. 표 3에서 사용되어지는 오차의 단위는 한번의 원을 그리는 사이클 동안 샘플링마다 X축, Y축 오차 절대치의 누적값이다. 이득값이 작아짐에 따라 추종오차가 커짐을 알 수 있다.

아래 그림들은 PD 이득값으로 $k_p = 43.5, k_d = 0.75$ 인 경우와 175, 3인 경우에서 PD만을 사용하였을 경우와 신경망 보상 방식을 사용하여 온라인 학습을 한 후 안정화되어진 상태에서의 직교 좌표상에서 그림을 비교하여 관찰한 것을 나타낸다. 먼저 제어기 이득값이 비교적 $K_p=175, K_d=3$ 인 경우 일반 PD 제어의 X-Y상의 위치 궤적은 그림 12와 같다. 약간의 위치 추종 오차를 유지하며 원경로를 추종하고 있다.

신경망을 사용하는 경우에는 신경망의 기본적인 파라미터인 모멘텀은 실험을 통하여 미리 적당한 값인 $\alpha=0.7$ 로 설정하여 사용하였다. 아래 그림 13은 신경회로망을 온라인으로 학습을 시켜야 하므로 초기에는 오차가 생기지만 대략 6-8번의 반복적인 회전 뒤에는 수렴하므로 맨 마지막 사이클의 경로를 출력하였다.

그림 13은 이득값이 $K_p=175, K_d=3$ 일 경우 신경망 보상

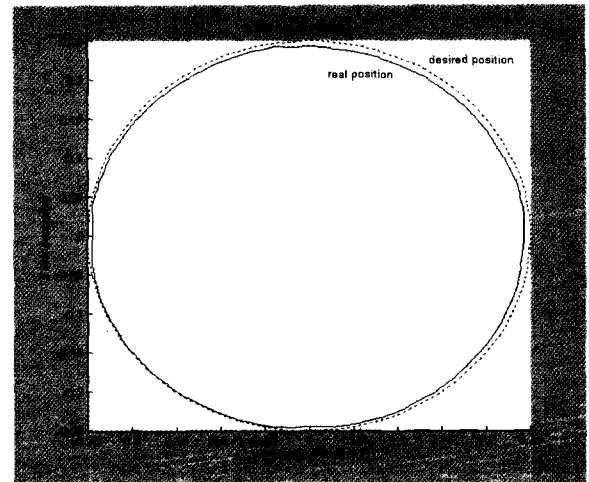


그림 12. 큰 PD 제어기의 이득값의 응답($K_p=175, K_d=3$).
Fig. 12. Large PD control gain($K_p=175, K_d=3$).

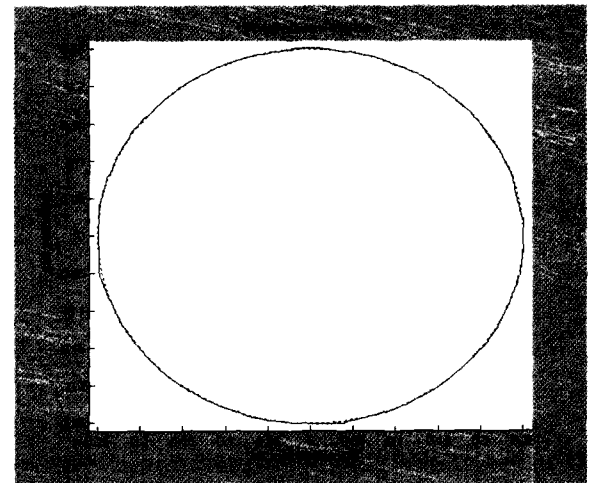


그림 13. 신경망 제어기의 응답($K_p=175, K_d=3$).
Fig. 13. Neural controller($K_p=175, K_d=3$).

방식을 사용하였을 경우이고 충분히 학습이 이루어진 후 12회전하였을 때의 직교 좌표계에서의 위치 궤적을 나타낸 것이다. 그림 12와 비교했을 경우 위치 추종이 매우 우수함을 알 수 있다.

또한 제어기의 이득값을 변화시켜 비교적 작은 값으로 설정할 경우에 위치 추종을 살펴보았다. PD 제어기 이득값 $K_p=43.5, K_d=0.75$ 에서의 원경로 위치 추종을 살펴보면 그림 14와 같다. 우선 PD 제어만을 사용한 경우는 궤적의 그래프에 나타난 것처럼 많은 오차를 가지고 있다.

같은 PD 제어기에 신경회로망으로 학습을 하면 그림 15에 나타나 것처럼 오차가 줄게 된다. 사용되어진 제어기 이득값들과 신경망 입력 보상 방식을 사용한 결과들을 살펴보면 앞에서 신경망의 보상 역할이 이루어지고 있음을 볼 수 있다. 특히, PD 제어기 자체만은 제어기 이득값에 큰 영향을 받지만 신경망의 경우에는 영향이 적어 이득값으로 인해 추종 오차를 크게 벗어나지는 않는다.

다음은 각기 다른 제어기 이득값에서 신경회로망의 학습으로 인해 초기 오차가 크고 일정한 회전이 이루어진 후에

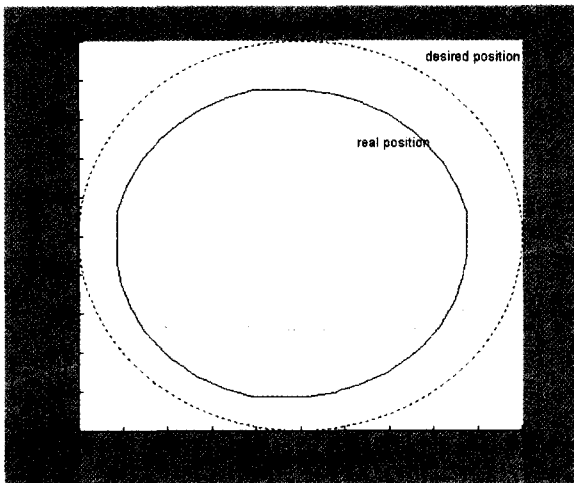


그림 14. 작은 이득값의 PD 제어기의 응답.
Fig. 14. Small PD control with($K_p=43.5, K_d=0.75$).

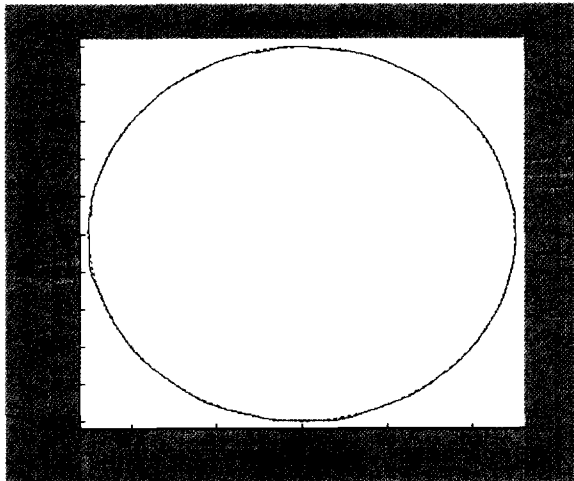


그림 15. 신경회로망 사용시 위치 추종($K_p=43.5, K_d=0.75$).
Fig. 15. Tracking performance using neural network($K_p=43.5, K_d=0.75$).

수렴하여 매 회전이 비슷한 오차를 가지게 되는 특성에 따라 각각의 신경회로망의 학습에 필요한 회전수와 초기 오차, 그리고 수렴 후 오차를 비교하여 나타내었다.

표 4에서 보이듯 PD 제어기 이득값과 학습률은 반비례적 성향을 보이며 같은 이득값을 가지고 실험을 할 경우 학습률이 높아질수록 초기 오차가 작아짐을 알 수 있다. 그러나 어느 정도 학습률이 커지면 발산으로 인한 시스템 불안정해

표 4. 신경망 학습후 학습률과 오차.
Table 4. Learning rates and errors after learning.

K_p	K_d	수렴 회전수	신경망 최적학습률	초기오차 (m/cycle)	수렴후 평균 오차(m/cycle)
350	6	11	0.00015	6.61	1.45
175	3	10	0.00045	10.64	2.4
87	1.5	2	0.002	11.80	2.05
43.5	0.75	2	0.0063	19.61	2.81

진다. 또한 이득값이 높아질수록 신경망의 수렴 시간이 길어지게 되나 어느 정도 회전이 이루어지면서 수렴한 후의 오차가 많이 줄어든다. 전체적으로 위의 학습후 결과를 살펴보면 PD 제어기만을 이용했을 때보다 많은 오차 감소로 인해 추종의 좋아졌다.

2. x-y 테이블 위의 역진자 제어

또 다른 비선형 시스템으로 역진자 제어 시스템을 실험하였다. 그림 11에 나타난 것처럼 X-Y 테이블 위에 역진자가 설치되어 있다. 역진자 제어는 일반적으로 많이 사용하는 비선형 제어 방식의 대표라 할 수 있다. 동력학 식이 비선형이고 입력으로 힘이 주어지면 출력으로 카트의 위치와 진자의 각이 제어되는 Single-input Multiple-output(SIMO) 시스템이기 때문이다. 먼저 기존의 PD나 PID 제어기를 사용했을 경우에 진자의 각은 쉽게 제어되었으나 동시에 카트의 위치는 제어하지 못하는 결과를 얻었다. 그림 16에서 보듯이 PD 제어기를 사용했을 경우 진자의 각은 0도로 급방 수렴하나 카트의 위치는 계속적으로 움직이는 것을 알 수 있다.

아래 그림 17은 신경망을 사용했을 경우 진자와 카트의

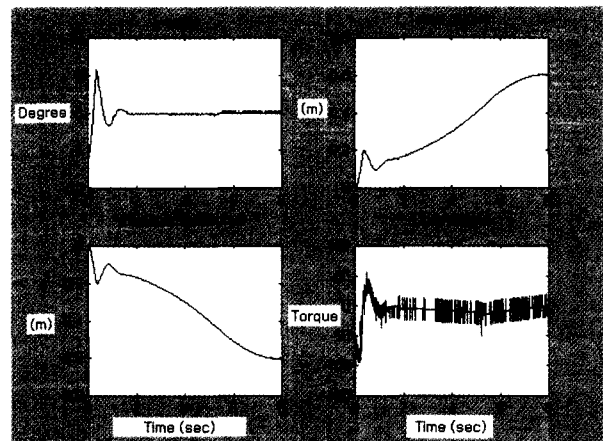


그림 16. $K_p=68, K_d=12.9, K_i=3.1$, 시작각도=5°.
Fig. 16. $K_p=68, K_d=12.9, K_i=3.1$, Initial angle=5°.

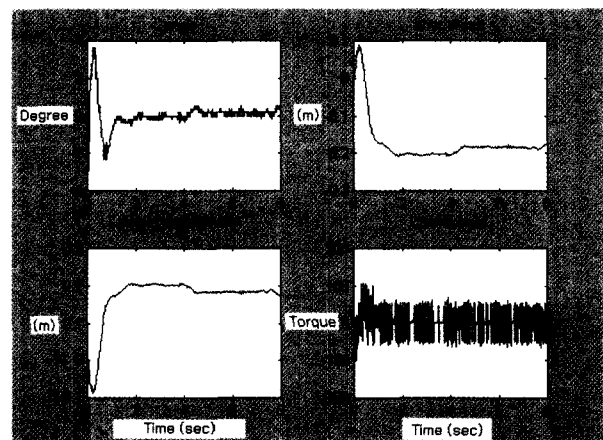


그림 17. $K_{pb}=59, K_{db}=9, K_{pc}=10, K_{dc}=7$, 초기각=3.4°, 학습률=0.0002, 뉴런 개수=6-6-4, $x_d=0.2m$.
Fig. 17. $K_{pb}=59, K_{db}=9, K_{pc}=10, K_{dc}=7$, initial angle=3.4°, $\eta=0.0002$. number of neuron=6-6-4. $x_d=0.2m$.

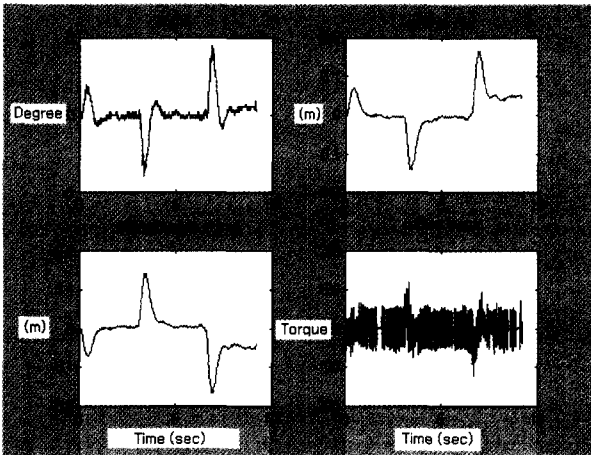


그림 18. $K_{\rho\theta}=59, K_{d\theta}=9, K_{\rho x}=10, K_{d x}=7$, 초기각=0°, 학습률=0.0008, 뉴런개수 6-6-4.

Fig. 18. $K_{\rho\theta}=59, K_{d\theta}=9, K_{\rho x}=10, K_{d x}=7$, initial angle=0°, $\eta=0.0008$, number of neuron 6-6-4.

움직임을 나타낸다. 진자와 카트 모두 1초 안에 원하는 위치로 수렴함을 알 수 있다. 아래 그림에서 진자의 각은 계속적으로 약간의 진동이 있으나 카트의 움직임은 진동이 없어 보이는 것은 카트의 단위가 meter로 표시되었기 때문이며 실제로 작은 진동이 있음을 알 수 있다.

그림 18에서는 외란의 형태로 손으로 진자를 툭 쳤을 때 나타나는 진자의 움직임을 나타낸다. 진자와 수레 모두 원하는 위치로 빠른 시간 안에 수렴함을 볼 수 있다.

VI. 결론

본 논문에서는 비선형시스템을 제어하기 위해 비선형제어기로 신경회로망을 사용하였다. 신경회로망 제어 방식의 하나인 입력 보상 방식을 실제로 DSP 보드에 구현하고 실험하여 온라인으로 제어가 가능함을 보였다. 신경회로망 제어 방식을 실제 시스템 x-y 테이블에 적용하여 원경로 위치 추종을 살펴보았다. 기존의 PD 제어기의 한계로 인한 오차를 보상하고 더 향상된 제어가 가능하도록 하였다. 제어기 이득값에 변화를 주어 실험한 결과, 신경회로망을 사용했을 경우의 위치 오차가 신경망을 사용하지 않았을 경우보다 현저하게 줄어들음을 알 수 있었다. 처음 사이클에는 추종 오차

가 다소 나타났지만 반복적인 작업을 통해 곧 줄어들었다. 역진자 제어의 경우에도 진자의 각과 카트의 움직임이 동시에 제어됨을 알 수 있었다.

참고문헌

[1] Sigeru Omatu, Marzuki Khalid, and Rubiyah Yusof, "Neuro-control and its applications," Springer, 1996.
 [2] F. L. Lewis, K. Liu, and A. Yesildirek, "Neural net robot controller with guaranteed tracking performance," *IEEE Symposium on Intelligent Control*, pp. 225-231, 1993.
 [3] 이보희, 이택승, 김진걸, "적용 학습률을 이용한 신경회로망의 학습 성능 개선 및 로봇 제어," 제어·자동화·시스템공학 논문지, pp. 363-372, 제3권, 제4호, 1997.
 [4] 최부귀, 이형기, 박양수, "신경회로망을 이용한 유연한 관절의 선단위치 tracking 제어기에 관한 실험적 평가," 제어 자동화 시스템 공학 논문지, pp. 738-746, 제4권, 제6호, 1998.
 [5] 주진화, 이장명, "신경망을 이용한 이동 로봇의 실시간 고속 정밀제어," 제어·자동화·시스템 공학회, pp. 95-104, 제5권, 제1호, 1999.
 [6] 김무진, 이영진, 박성준, 이만형, "신경망을 이용한 이동 로봇의 정밀제어," 제어 자동화 시스템 공학회, pp. 689-696, 제6권, 제8호, 2000.
 [7] 이민호, 최형식, "해저작업 로봇 매니퓰레이터를 위한 신경회로망을 이용한 슬라이딩모드 제어기," 제어·자동화·시스템 공학논문지, 제6권, 제4호, pp. 305-312, 2000.
 [8] T. Ozaky, T. Suzuki, T. Furuhashi, S. Okuma, and Y. Uchikawa, "Trajectory control of robotic manipulators using neural networks," *IEEE Trans. on Industrial Electronics*, vol. 38, pp. 195-202, 1991.
 [9] A. Ishiguro, T. Furuhashi, S. Okuma, and Y. Uchikawa, "A neural network compensator for uncertainties of robot manipulator," *IEEE Trans. on Industrial Electronics*, vol. 39, pp. 61-66, December, 1992.
 [10] K. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. on neural networks*, vol. 1, pp. 4-27, 1990.
 [11] M. Miyamoto, M. Kawato, T. Setoyama, and R. Suzuki, "Feedback error learning," *IEEE Trans. on Neural Networks*, vol. 1, pp. 251-265, 1988.

정 슬

1964년 9월 11일생. 1988년 미국 웨인주립대 전기 및 컴퓨터 공학과 졸업. 1991 미국 캘리포니아대 석사. 동대학 박사(1996). 1997년~현재 충남대학교 메카트로닉스공학과 조교수. 관심분야는 지능 제어 및 감성 공학, 로봇공학.

임 선 빈

1975년 10월 11일생. 1998년 충남대 메카트로닉스공학과 졸업. 2000년 충남대 메카트로닉스공학과 석사. 관심분야는 신경망 제어 및 DSP 응용.

