

# 신경회로망을 이용한 시간최적 제어

최원수\*, 윤종선\*\*

## Time-optimal Control Utilizing Neural Networks

Wonsu Choi\*, Joongsun Yoon\*\*

### ABSTRACT

A time-optimal control law for quick, strongly nonlinear systems has been developed and demonstrated. This procedure involves the utilization of neural networks as state feedback controllers that learn the time-optimal control actions by means of an iterative minimization of both the final time and the final state error for the systems with constrained inputs and/or states. A neural identifier or a genetic algorithm identifier could be utilized for modeling the partially known systems and the unknown systems. The nature of neural networks as a parallel processor would circumvent the problem of "curse of dimensionality". The control law has been demonstrated for both a torque input motor and a velocity input motor identified by a genetic algorithm called GENOCOP.

**Key Words:** Time-optimal Control(시간최적제어), Input/States Constraints(입력/상태제한)  
Neuro Controller and Identifier(신경회로망 제어기 및 식별기)  
Genetic Algorithm Identifier(유전자알고리즘 식별기)

### 1. 서론

비선형성이 강한 계의 시간최적 제어는 종료시간과 종료상태 오차의 동시 최소화라는 설계의 복잡성 때문에 제어기 개발이 매우 어렵다<sup>(1)</sup>.

제한된 크기의 입력을 가진 시간최적 제어 문제의 경우 여러 점에서의 값이 주어진 비선형미분방정식을 풀어야 하는 비선형 여러점 경계값 문제(multipoint boundary-value problem)가 된다. 이 문제는 보통 해석적으로 풀 수 없으므로 수치 해석적으로 풀어야만 할 때가 많다. 지금까지의 방법들은 linear, low-order, scalar

input, fixed-terminal-time, and/or moving-end-point problems와 같은 제한된 문제들을 다루고 있다. 2차 성능지수와 특정의 2차나 3차 선형계의 최적을 다루는 문제들을 빼고는 거의 해석적으로 답을 찾을 수 없다. 따라서 최적제어 문제는 반복계산으로 풀어야 한다. 제한된 크기의 입력을 가지는 최적제어 문제가, 특히 로봇 머니플레이터의 제어기 설계 분야에서 많은 관심을 끌고 있다. 이 방법들의 성공 여부는 문제의 특성과 좋은 초기 값의 가정에 달려 있다. 따라서 비선형 높은 차수 문제를 풀, 구현하기 쉽고 체계적인 방법이 필요하다<sup>(2)</sup>.

하나의 대안으로 대수 문제의 근을 찾는 알고리즘인

\* LG 전자 생산기술센터

\*\* 부산대학교 기계공학부 지능제어정보 연구실, 기계기술연구소

Davidenko 법에 따른 새로운 수치해석법이 제안되었다. 유한차분(finite difference)법과 같은 이산화(discretization)과정으로 미분방정식들을 대수방정식의 계로 바꾸는 이러한 2점 경계값 문제의 풀이법은 여러점 경계값 문제에도 바로 쓰일 수 있다<sup>(2)</sup>.

또다른 대안으로 유연한 비선형 매핑인 신경회로망을 시간최적 제어를 학습할 상태 피드백 제어계로 사용한 방법을 제안한다. 제어대상에 대한 선형적 가정이 없다는 점과 문제 고유의 제한 조건들을 포함한다는 점에서 매우 일반적(general)이다. 또한 제어대상의 정보가 부족할 때도 신경회로망 및 유전자 알고리즘 식별(identification) 기법을 거쳐 시간최적 제어를 할 수 있다. 제안된 방법을 모터 관성계에 적용하여 검증한다.

2. 시간최적 제어문제

시간최적 제어기의 목적은 제어계를 모든 제한조건을 만족시키며 가능한 한 최소시간에 바람직한 정상상태로 가도록 하는 것이다.

$\underline{x}$ 가 상태벡터,  $\underline{x} = (x_1, x_2, \dots, x_n)^T$  이고  $\underline{u}$ 는 제어 입력벡터,  $\underline{u} = (u_1, u_2, \dots, u_r)^T$  일 때 초기상태  $\underline{x}_0$  종료상태  $\underline{x}_1$  과 제한조건을 만족하는 제어 조합  $U$ 를 가지는 연속 동적계

$$\dot{\underline{x}}(t) = \underline{f}(\underline{x}(t), \underline{u}(t), t) \tag{2.1a}$$

$$\underline{x}(t_0) = \underline{x}_0 \tag{2.1b}$$

$$\underline{x}(t_1) = \underline{x}_1 \tag{2.1c}$$

$$\underline{u}(t) \in U \text{ 모든 } t \text{ 에서} \tag{2.1d}$$

를  $\underline{x}_0$ 에서  $\underline{x}_1$ 로 최단시간에 옮기는  $U$ 에 속하는 제어기를 설계하는 문제가 된다. 여기서 평가함수  $J$ 는

$$J = \int_{t_0}^{t_1} f_0(\underline{x}(t), \underline{u}(t))dt \tag{2.1e}$$

이다. 최적 해는 Pontryagin의 최대정리(Maximum Principle)를 써서 끌어낼 수 있다<sup>(3)</sup>. Hamiltonian

$$H(\underline{x}, \underline{\varphi}, \underline{u}, t) = \sum_{i=0}^n \varphi_i f_i(\underline{x}, \underline{u}) \tag{2.2}$$

에서 상호상태(covariant)방정식과 상태방정식의 표준계

$$\dot{\varphi}_i = -\frac{\partial H}{\partial x_i}, \quad i = 1, 2, \dots, n, \tag{2.3}$$

$$\dot{x}_i = \frac{\partial H}{\partial \varphi_i}, \quad i = 1, 2, \dots, n. \tag{2.4}$$

를 만든다. 최적제어는  $\underline{x}_0$ 을  $\underline{x}_1$ 로 움직이는데 드는 전 시간에 걸친 Hamiltonian의 최적화로 서로 얽힌(coupled) 상태방정식과 상호상태방정식을 푸는 문제가 된다. 이는 시간  $t_0$ 와  $t_1$ 에서의 경계값  $\underline{x}$ 을 가지는 변수  $\underline{x}$ 와  $\underline{\varphi}$ 의  $2n$ 개의 비선형 1차 상미분방정식의 조합 즉 2점 경계값 문제가 된다.

3. 신경회로망에 의한 시간최적 제어

시간최적 제어문제를 풀 신경회로망 제어계의 구조는 Fig. 1과 같다. 제어계의 학습(training)은 회로망을 피드백 구조 안에서 구현하기 전에 오프라인으로 수행된다. 제어기의 입력과 출력 계산은 주로 행렬 곱으로 이루어지고, 제어기의 가중치(weight)들만 필요하므로 온라인 계산 양은 매우 낮아 적은 메모리만 필요로 한다.

신경 제어계의 이산화 모델(discrete-time formulation)은

$$\underline{x}[n+1] = \underline{F}(\underline{x}[n], \underline{u}[n]) \tag{3.1}$$

$$\underline{u}[n] = \underline{G}(\underline{x}[n], \underline{u}_T[W]) \tag{3.2}$$

이고 이 때  $\underline{x}[n]$ 은 시간간격  $n$ 일 때의 실제 제어계의 상태,  $\underline{u}[n]$ 은 시간 간격  $n$ 일 때의 제어 입력,  $\underline{x}_T$ 는 제어계의 바람직한 종료상태,  $W$ 는 제어기의 가중치 행렬을 나타낸다.

일반적으로 목표 상태에 도달할 최소 종료시간과 해당 입력을 모르기 때문에 적절한 제어 범칙은 종료시간 최적

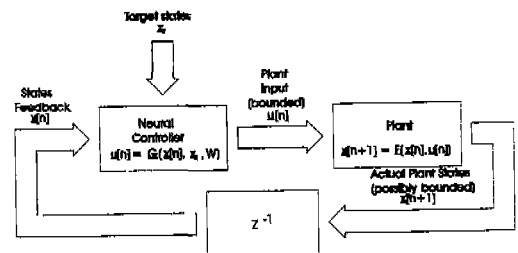


Fig. 1 Control System Structure

화와 종료상태 오차 최적화의 두 단계로 나누어 찾는다<sup>(4, 5)</sup>. 제어기는 sigmoid 비선형성을 가지는 두개의 은닉 층과 하나의 선형 출력 층으로 이루어진 다층 퍼셉트론(multilayer perceptron)으로 정해진다<sup>(6)</sup>.

### 3.1 종료시간의 최적화

종료시간은 반복적 계산(iterative algorithm)으로 찾는다. 종료시간의 간격  $K_i$ 를 실제 값보다 크게 두고 시작한다. 첫 단계는 이  $K_i$ 로 계산한 종료상태 오차 함수  $J$ 가 수렴 기준 값  $\mathcal{W}$ 보다 작으면 적은 시간 안에 최적화를 시도할 수 있으므로  $K_i$ 값을 하나씩 줄인다. 반복 시간  $T$ 안에 수렴 조건을 맞추지 못하면 그 시간 간격으로는 최적화가 될 수 없으므로  $K_i$ 를 하나 늘려 수정을 마무리 짓는다. 각  $K_i$ 의 초기 상태에서 반복 시간을 나타내는 counter/status 변수  $C_i$ 를 가진다. 이 변수는 각 최적 과정 시작에서 0으로 놓고 반복 시간  $T$ 와 비교된다<sup>(4)</sup>.

### 3.2 종료상태 오차의 최적화

종료상태 오차 최적화 알고리즘<sup>(4, 5)</sup>은 다음과 같다.

$$J = \frac{1}{2} \|x_T - x[K]\|^2 = \frac{1}{2} (x_T - x[K])^T (x_T - x[K]) \quad (3.3)$$

$$\frac{\partial J}{\partial x[K]} = -(x_T - x[K])^T \quad (3.4)$$

$$\left. \frac{\partial J}{\partial x[n]} \right|_{u=\text{const}} = \frac{\partial J}{\partial x[n+1]} \left. \frac{\partial x[n+1]}{\partial x[n]} \right|_{u=\text{const}} \quad (3.5)$$

$$\left. \frac{\partial J}{\partial u[n]} \right|_{x=\text{const}} = \frac{\partial J}{\partial x[n+1]} \left. \frac{\partial x[n+1]}{\partial u[n]} \right|_{x=\text{const}} \quad (3.6)$$

$$\frac{\partial J}{\partial x[n]} = \text{BP} \left( \left. \frac{\partial J}{\partial u[n]} \right|_{x=\text{const}} \right) + \left. \frac{\partial J}{\partial x[n]} \right|_{u=\text{const}} \quad (3.7)$$

계산은  $\frac{\partial J}{\partial x[K]}$ 로부터 식 (3.5), (3.6), (3.7)의 순으로

반복적으로 이루어진다.  $\frac{\partial J}{\partial x[K]}$ 의 값이 구해지면 신

경회로망으로 역전파(backpropagation)되어 종료상태 오차  $J$ 가 줄어들도록 학습된다. 여기서 BP는 역전파학습(backpropagation learning)<sup>(6)</sup> 구조를 가진다. 제한된 상태를 가지는 시간최적 제어의 오차 함수  $J'$ 는 상태를 벗어난 크기만큼 가중치(penalty)를 더하여 다음과 같이 정의된다.

$$J' = \begin{cases} J + \frac{1}{2} \varepsilon (x_i[n] - x_i^{\min})^2 & \text{for } x_i[n] < x_i^{\min} \\ J & \text{for } x_i^{\min} \leq x_i[n] \leq x_i^{\max} \\ J + \frac{1}{2} \varepsilon (x_i[n] - x_i^{\max})^2 & \text{for } x_i[n] > x_i^{\max} \end{cases} \quad (3.8)$$

또 제한된 입력을 가지는 시간최적 제어의 오차함수는 입력 제한을 벗어난 크기만큼 가중치(penalty)를 주어 다음과 같이 정의된다.

$$J'' = \begin{cases} J' + \frac{1}{2} \alpha (u_i[n] - u_i^{\min})^2 & \text{for } u_i[n] < u_i^{\min} \\ J' & \text{for } u_i^{\min} \leq u_i[n] \leq u_i^{\max} \\ J' + \frac{1}{2} \alpha (u_i[n] - u_i^{\max})^2 & \text{for } u_i[n] > u_i^{\max} \end{cases} \quad (3.9)$$

이 때  $\varepsilon$  과  $\alpha$  로 오차 기율기의 기여도를 조절할 수 있게 한다.

### 3.3 신경회로망에 의한 최적화

제어 대상의 정보가 부족할 때 Fig. 2와 같이 신경회로망 식별을 거쳐 제어를 할 수 있다<sup>(5)</sup>.

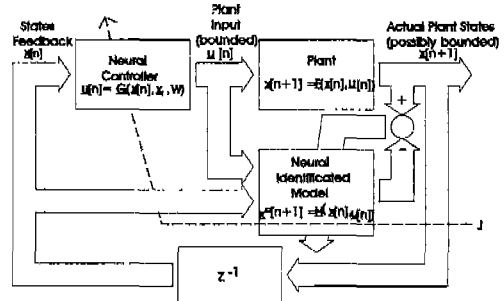


Fig. 2 Neural Network Control for Unknown Plant with Identification

모델링에 사용되는 신경회로망은

$$\underline{x}^{ID}[n+1] = \underline{H}(\underline{x}[n], \underline{u}[n], W^{ID}) \quad (3.10)$$

의 입력과 출력의 매핑 관계를 가지며 식 (3.1)과 같이 되도록 학습된다. 여기서  $\alpha$  는 시스템 식별에 쓰인 신경회로망의 가중치이다. 식별에 사용되는 신경회로망의 학

습은 실제 대상의 출력과 신경회로망 출력의 오차

$$J^{ID} = \frac{1}{2} \|\underline{x}[n] - \underline{x}^{ID}[n]\|^2 \quad (3.11)$$

를 신경회로망에 역전파하여 이루어진다.

제어 대상의 수학적 모델인 식 (3.1)의 정보가 모자라거나 비선형성이 강하여 계산이 힘들면 식별된 모델인 식 (3.10)을 써서 식 (3.5)와 식 (3.6)을

$$\underline{x}[n] = \underline{x}^{ID}[n] \quad (3.12)$$

$$\left. \frac{\partial \underline{x}[n+1]}{\partial \underline{x}[n]} \right|_{\underline{u}=\text{const}} = \left. \frac{\partial \underline{x}^{ID}[n+1]}{\partial \underline{x}[n]} \right|_{\underline{u}=\text{const}} = \frac{\partial \underline{H}(\underline{x}[n], \underline{u}[n], \mathbf{W}^{ID})}{\partial \underline{x}[n]} \quad (3.13)$$

$$\left. \frac{\partial \underline{x}[n+1]}{\partial \underline{u}[n]} \right|_{\underline{x}=\text{const}} = \left. \frac{\partial \underline{x}^{ID}[n+1]}{\partial \underline{u}[n]} \right|_{\underline{x}=\text{const}} = \frac{\partial \underline{H}(\underline{x}[n], \underline{u}[n], \mathbf{W}^{ID})}{\partial \underline{u}[n]} \quad (3.14)$$

로 바꾸고 종료상태 최적화는 3.2결과 같이 이루어지고 종료시간의 최적화는 3.1결과 같이 이루어진다.

### 3.4 유전자 알고리즘에 의한 최적화

속도 입력형 모터의 경우, 힘성분의 제어 입력을 사용하지 못하므로 제어 입력을 속도성분으로 바꾸어주기 위해 직접 토크 입력형 모터의 운동방정식 (3.1)은

$$\underline{x}[n+1] = \mathbf{F}^*(\underline{x}[n], \underline{v}[n]) \quad (3.15a)$$

$$\underline{v}[n] = \mathbf{V}(\underline{v}[n-1], \underline{u}[n]) \quad (3.15b)$$

와 같은 제어대상과 속도 발생기(적분기)의 이산화 모델로 나타내어진다. 이 때  $\underline{v}[n]$ 은 시간 간격  $n$ 일 때의 제어 입력으로 가속도이며  $\underline{v}[n]$ 는 시간 간격  $n$ 일 때  $\underline{v}[n]$ 에 의한 제어계의 속도 입력을 나타낸다.<sup>(5)</sup>

신경회로망의 학습은 식 (3.15)를 써서 식 (3.6)을

$$\left. \frac{\partial J}{\partial \underline{u}[n]} \right|_{\underline{x}=\text{const}} = \frac{\partial J}{\partial \underline{x}[n+1]} \cdot \frac{\partial \underline{x}[n+1]}{\partial \underline{v}[n]} \cdot \frac{\partial \underline{v}[n]}{\partial \underline{u}[n]} \Big|_{\underline{x}=\text{const}} \quad (3.16)$$

로 바꾸고 종료상태 오차 최소화와 종료시간의 최적화를

행한다.<sup>(6)</sup>

속도 입력형 모터는 덧붙여져 필요한 서보 제어기의 변수 정보를 얻기 힘들므로, 식 (3.15)의 선형 이산화 모델

$$\underline{x}[n+1] = \mathbf{A} \underline{x}[n] + \mathbf{B} \underline{v}[n] \quad (3.17a)$$

$$\underline{v}[n] = \underline{v}[n-1] + \underline{u}[n]\Delta T \quad (3.17b)$$

의 A, B 행렬 값을 식별하여 검증에 사용한다. 실제 모터의 여러 위치와 속도 단계에 여러 단계의 속도 입력을 주어 샘플링 시간 후의 위치와 속도를 식별하기 위한 데이터로 두고 유전자 알고리즘의 하나인 GENOCOP(GENetic algorithm for Numerical Optimization for CONstrained Problems)을 써서 A, B 행렬의 값을 구한다.<sup>(6, 7)</sup>

GENOCOP은 Michalewicz가 제안한 유전자 알고리즘으로 해를 찾는 과정은 Holland의 SGA(Simple Genetic Algorithms)와 같으나 부호화된 유전자 대신 실제의 파라미터인 실수 형태의 유전자를 사용하여 부호화 계산 과정이 필요 없다. 또한 최적화 파라미터의 크기에 제한이 있는 문제(constrained problem)에 적합하여 해에 관한 정보를 계속하여 사용할 수 있으며 교배 및 돌연변이 연산이 유전자 사이의 염색체 교환이 아닌 염색체(gene) 사이의 linear combination으로 이루어진다. GENOCOP은 이진화 스트링을 쓰는 SGA와 달리 실수 값으로 이루어진 스트링을 가진다.<sup>(7)</sup>

식별될 파라미터는

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \quad (3.18a)$$

$a_{11}$	$a_{12}$	$a_{21}$	$a_{22}$	$b_1$	$b_2$
----------	----------	----------	----------	-------	-------

(3.18b)

와 같이 유전자로 만들어 실제 위치와 속도 값과 유전자를 식 (3.17)에 대입하여 구한 위치 속도 값과의 차이가 최소가 되도록 유전자 알고리즘을 사용한다. 적합도(fitness)는

$$\text{Fit}_i = \frac{1}{1 + \sqrt{\sum_k E_v^2 + \sum_k E_p^2}} \quad (3.19)$$

이며 각 학습 데이터에 대한 모터 측정값과 스트링 계산값의 오차 제곱 합의 역수를 사용한다. 여기서  $\text{Fit}_i$ 는  $i$ 번째 유전자의 적합도,  $E_v$ 와  $E_p$ 는 각각 속도 오차와 위치

오차를 뜻하고  $k$ 는 식별을 위한  $k$ 번째 데이터를 뜻한다.

#### 4. 검증

##### 4.1 토크 입력형 모터의 최적화

모터 관성계(double integrator)의 운동 방정식은

$$\tau = J\ddot{\theta} \quad (4.1)$$

이고 상태벡터  $\underline{x} = (\theta, \dot{\theta})^T$  입력  $u = \tau$  일 때 상태방정식은

$$\dot{\underline{x}} = \mathbf{A}^* \underline{x} + \mathbf{B}^* u \quad (4.2a)$$

$$\mathbf{A}^* = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{B}^* = \begin{bmatrix} 0 & 1 \\ 1 & J \end{bmatrix} \quad (4.2b)$$

이다. 이산화 모델은 이산화 상태벡터가  $\underline{x} = (\theta[n], \dot{\theta}[n])^T$  이고  $J = 0.525 \text{ kg} \cdot \text{m}^2$ ,  $\Delta t = 0.05 \text{ sec}$  일 때

$$\underline{x}[n+1] = \mathbf{A} \underline{x}[n] + \mathbf{B} u \quad (4.3a)$$

$$\mathbf{A} = \begin{bmatrix} 1 & 0.05 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0.002381 \\ 0.09524 \end{bmatrix} \quad (4.3b)$$

와 같다.

입력 제한이 있을 경우의 최적화 결과는 Fig. 3과 Fig. 4와 같다. 시뮬레이션 파라미터 값은 Table 1과 같고 Pontryagin의 최대정리에서 구한 결과와 비교하였다.

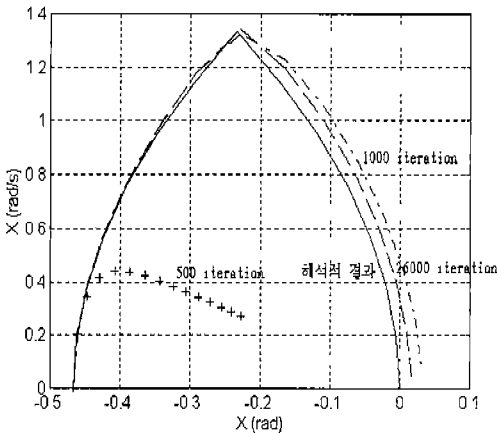


Fig. 3 Evolution of the Phase-plane Plot

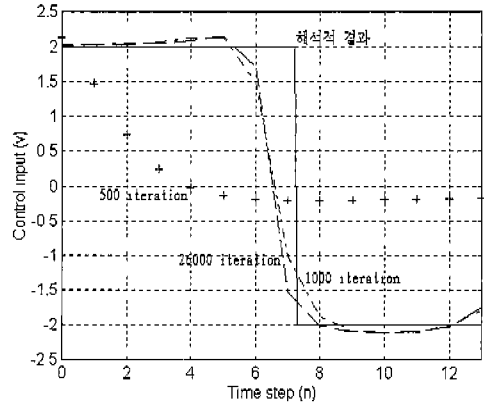


Fig. 4 Evolution of the Control Signal

Table 1. Simulation Parameter Values for Optimization with Bounded Input

	Parameter Name	Symbol	Parameter Value
Plant	Initial position	$\theta [0]$	-0.4667 rad
	Initial velocity	$\dot{\theta} [0]$	0 rad/sec
	Target position	$\theta_T$	0 rad
	Target velocity	$\dot{\theta}_T$	0 rad/sec
	Bounds of control input	$u^{\max}$ $u^{\min}$	2 N · m -2 N · m
Neural Controller	Input nodes		2
	1st Hidden nodes		10
	2nd Hidden nodes		2
	Output nodes		1
	Learning rate	$\eta$	1
	Initial final-time step	$K$	20
	Convergence threshold	$\Psi$	0.001
	Counter/status variable	$C$	2,000
Results	Switching time	$t_s^*$	0.3625 sec
		$\bar{t}_s$	0.35~0.4 sec
	Minimum time	$t_f^*$	0.725 sec
		$\bar{t}_f$	0.7 sec

- true optimal solution
- adapted neural optimal solution
- + switching occurs between 0.35 sec and 0.4 sec

입력 제한과 상태 제한이 같이 있을 경우의 종료시간 최적화와 종료상태 오차 최적화 결과는 Fig. 5와 Fig. 6과 같고 시뮬레이션 파라미터 값은 Table 2와 같다.

Table 3의 파라미터 값으로 식별한 모델은 종료상태에서 약 0.02 rad의 오차를 보여준다. 이 오차는 학습 간격을 작게 하면 줄어든다. 입력 제한이 있을 경우 식별 모델에 대한 최적화 결과는 Fig. 7과 Fig. 8과 같고 시뮬레이션 파라미터 값은 Table 4와 같다.

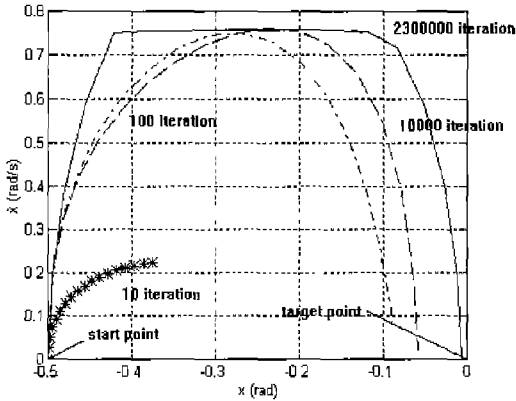


Fig. 5 Evolution of the Phase-plane Plot

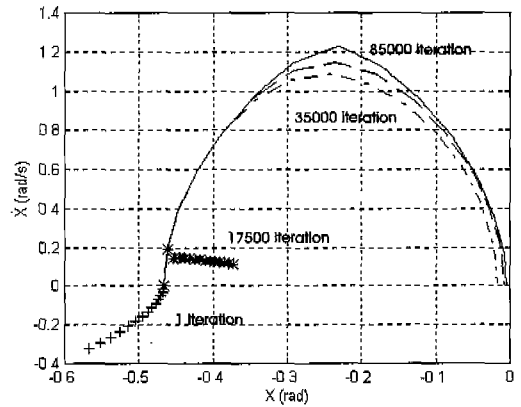


Fig. 7 Evolution of the Phase-plane Plot

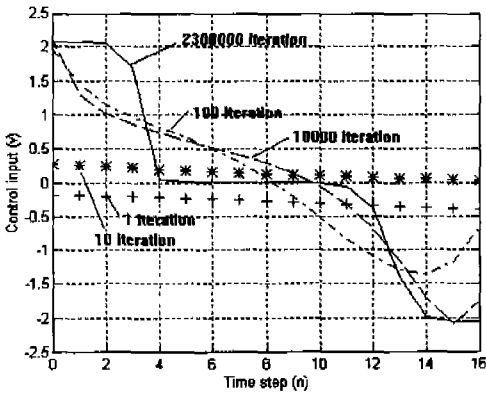


Fig. 6 Evolution of the Control Signal

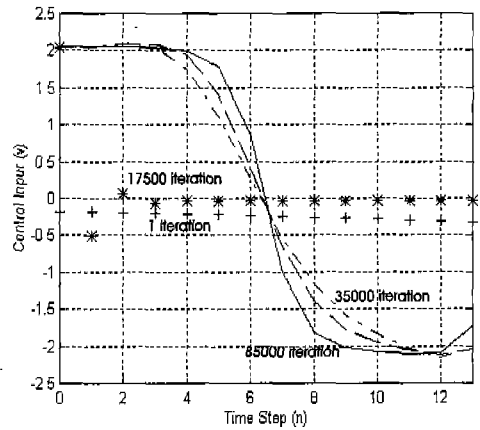


Fig. 8 Evolution of the Control Signal

Table 2. Simulation Parameter Values for Optimization with Bounded States

	Parameter Name	Symbol	Parameter Value
Plant	Bounds of velocity	$\theta_{max}$	0.75 rad/s
		$\theta_{min}$	-0.75 rad/s
Neural Controller	Convergence threshold	$\Psi$	0.0004
	Counter/status variable	C	10,000

나머지 파라미터는 Table 1과 같음

Table 3. Simulation Parameter Values for System Identification

	Parameter Name	Symbol	Parameter Value
Neural Identifier	Input nodes		3
	1st Hidden nodes		10
	2nd Hidden nodes		10
	Output nodes		2
	Learning rate	$\eta$	0.005
Learning Data	Convergence threshold	$\Psi$	0.001
	Position data	$\theta$	-0.5~0.001rad
	Velocity data	$\dot{\theta}$	0~1.2 rad
	Control input data	$u$	-2~2 kg · m <sup>2</sup>

learning data는 0.1 간격으로 제공

Table 4. Simulation Parameter Values for Optimization with Identified Model

	Parameter Name	Symbol	Parameter Value
Neural Controller	Learning rate	$\eta$	0.005
	Initial final-time step	K	20
	Convergence value	$\Psi$	0.001
	Counter/status variable	C	2,000

신경망의 형태는 Table 1과 같음

#### 4.2 속도 입력형 모터의 최적화

검증에 쓰인 모터 제어기는 Yaskawa AC servo motor, interface board DR\_DAS D/A, PC로 구성된 다<sup>6)</sup>. 검증에 쓰인 모터의 GENOCOP 기법에 의한 식 (3.17)의 A, B 행렬의 식별 결과는  $\Delta t = 0.05$  sec.일 때

$$A = \begin{bmatrix} 1 & 0.02 \\ 0 & 0.57 \end{bmatrix}, \quad B = \begin{bmatrix} 0.029519 \\ 0.424567 \end{bmatrix} \quad (4.4)$$

이다. 식별에 사용된 변수는 Table 5와 같고 알고리즘 반복 횟수에 대한 적합도의 최대 값의 변화는 Fig. 9와 같다.

Table 5. Simulation Parameter Values for Plant Identification

	Parameter Name	Symbol	Parameter Value
Identification Data	Position	$\theta$	-0.5~0.1 rad ( $\Delta\theta=0.05$ )
	Velocity	$\dot{\theta}$	0~1.2 rev/sec ( $\Delta\dot{\theta}=0.1$ )
	Acceleration	$u$	-20~20 rev/sec <sup>2</sup> ( $\Delta u=2$ )
GENOCOP Parameters	Population		50
	Mutation	$P_m$	0.3
	Crossover	$P_c$	0.9
	Constraints	A B	-2~-2 -1~-1

\*Max. Iteration 3100 Threshold 0.001~0.0001

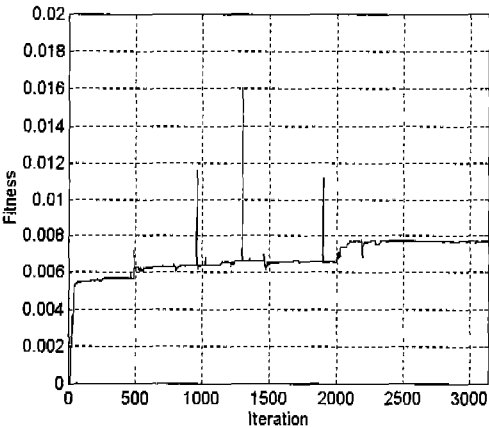


Fig. 9 Evolution of the Maximum Fitness Value

입력 제한이 있을 경우의 최적화 시뮬레이션 결과는 Fig. 10, Fig. 11과 같고 시뮬레이션 파라미터 값은 Table 6과 같다.

입력 제한과 상태 제한이 같이 있을 경우의 최적화 결과는 Fig. 12와 Fig. 13과 같고 시뮬레이션 파라미터 값은 Table 7과 같다.

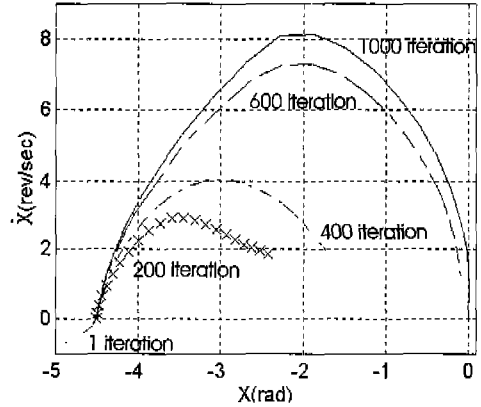


Fig. 10 Evolution of the Phase-plane Plot

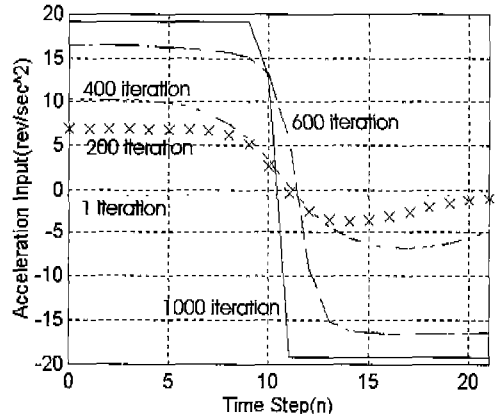


Fig. 11 Evolution of the Acceleration Input

Table 6. Simulation Parameter Values for Optimization with Bounded Input

	Parameter Name	Symbol	Parameter Value	
Plant	Initial position	$\theta [0]$	-4.4995 rad	
	Initial velocity	$\dot{\theta} [0]$	0 rev/sec	
	Target position	$\theta_T$	0 rad	
	Target velocity	$\dot{\theta}_T$	0 rev/sec	
	Bounds of acceleration input	$u^{\max}$		20 rev/sec <sup>2</sup>
		$u^{\min}$		-20 rev/sec <sup>2</sup>
Neural Controller	Input nodes		2	
	1st Hidden nodes		10	
	2nd Hidden nodes		2	
	Output nodes		1	
	Learning rate	$\eta$	0.5	
	Initial final-time step	$K$	29	
Convergence threshold	$\Psi$	0.001		
Counter/status variable	$C$		10000	

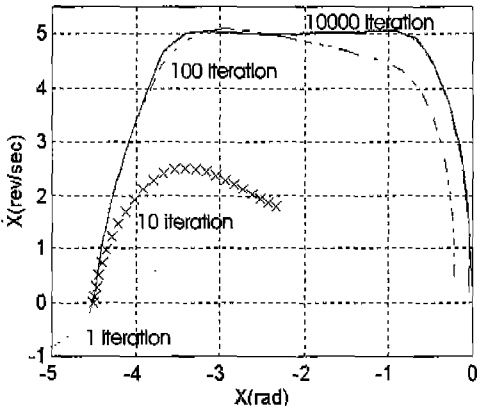


Fig. 12 Evolution of the Phase-plane Plot

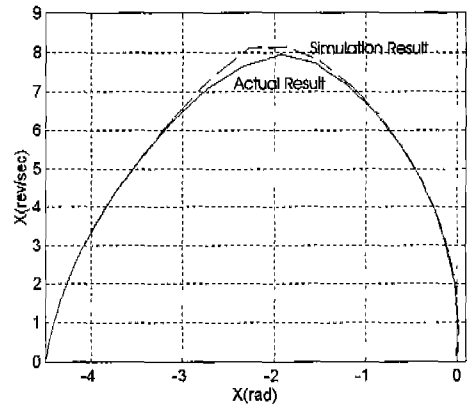


Fig. 14 Evolution of the Phase-plane Plot

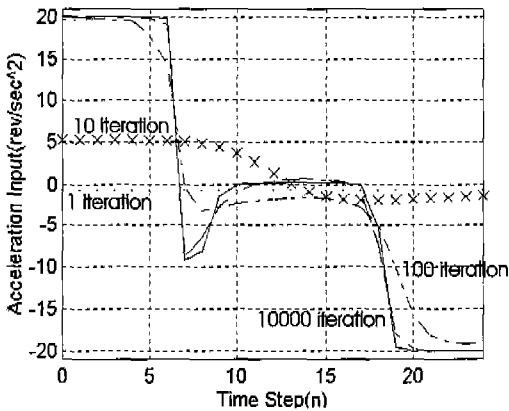


Fig. 13 Evolution of the Acceleration Input

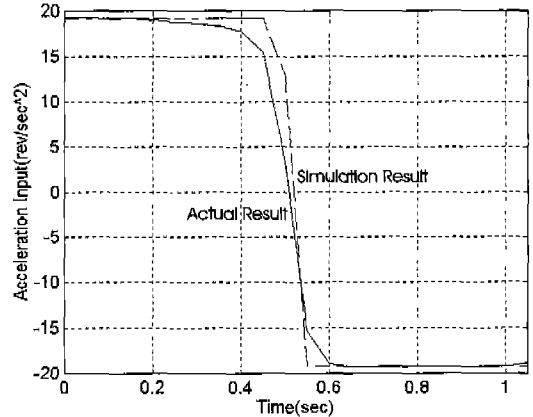


Fig. 15 Evolution of the Acceleration Input

Table 7. Simulation Parameter Values for Optimization with Bounded Input and States

	Parameter Name	Symbol	Parameter Value
Plant	Bounds of velocity	$\theta_{max}$	5 rev/sec
		$\theta_{min}$	-5 rev/sec
Neural Controller	Convergence threshold	$\psi$	0.001
	Counter/status variable	C	10000

나머지 파라미터는 Table 6과 같음

입력 제한이 있을 때의 속도 입력형 모터를 사용한 시뮬레이션 결과와 비교된 실제 모터 제어 시스템에 적용한 결과는 Fig. 14, Fig. 15와 같다.

시뮬레이션과 실험 결과는 switching 부분에서 어긋남을 보여주는데 이러한 어긋남은 Table 3의 거친(rough) 학습 간격 때문인 것 같다. 학습 간격을 작게 하면 오차는 줄일 수 있을 것이고 본 검증의 예에서는 모델의 정확도를 기하지 않고도 최적화라는 소기의 목적을 달성할 수 있기에 이러한 불일치를 줄이려는 노력을 하지 않았다. 비선형 다자유도 문제에서는 학습 간격의 조정에 따른 더욱 정확한 모델링이 필요하고 효율적인 제어를 보장해줄 것으로 생각된다.

## 5. 결론

비선형성이 강한 계의 시간최적 제어 방법이 제안되고



검증되었다. 신경회로망에 의한 이 방법은 제한된 크기의 입력이나 상태를 가지는 제어 대상을 적절히 다루고 있다. 모터 제어계의 정보가 부족한 경우 신경회로망이나 GENOCOP과 같은 유전자 알고리즘에 의한 모델링 기법을 써서 사용하여 바람직한 결과를 얻었다. 모터 제어계의 검증은 바람직한 결과를 보여주었다. 신경회로망의 병렬처리의 속성은 다자유도 로봇과 같은 고자유도계에도 확장 적용되어 "curse of dimensionality" 문제(8)를 피할 수 있을 것이다.

널리 쓰이는 역전파 다중 퍼셉트론(multilayer perceptron with backpropagation learning)은 정적인 패턴 분류(static pattern classification) 문제를 푸는데 효과적이다. 이러한 신경회로망 제어기로 모터의 시간 최적 제어와 같은 동적 문제의 제어를 풀 제어기의 역전파 학습에는 엄청난 양의 학습 데이터가 필요하다. 퍼드백 형의 동적 신경회로망이나 강화학습(reinforcement learning)을 쓰면 더욱 효과적인 제어를 피할 수 있을 것이다.<sup>(6)</sup>

참 고 문 헌

1. Lewis, F., et al., Control of Robot Manipulators, Macmillan, New York, 1993.

2. 윤중선, "Davidenko법에 의한 시간최적 제어문제의 수치해석해", 한국정밀공학회지, 제12권 제5호, pp. 57-68, 1995.

3. Pontryagin, L., et al., The Mathematical Theory of Optimal Processes, The Macmillan Press, New York, 1964.

4. Niesler, T. and du Plessis, J., "Time-Optimal Control by Means of Neural Networks", IEEE Control System Magazine, Vol. 15 No. 5, pp. 23-33, October 1995.

5. 최원수, 신경회로망을 이용한 시간최적 제어, 석사논문, 부산대학교, 1997.

6. Jang, J.-S., et. al., Neuro-Fuzzy and Soft Computing, Prentice-Hall, Upper Saddle River, 1997.

7. Michalewicz, Z., Genetic Algorithms + Data Structure = Evolution Programs, Springer-Verlag, New York, 1994.

8. Bellman, R. and Kalaba, R., Dynamic Programming and Modern Control Theory, Academic Press, New York, 1965.