

신경회로망을 이용한 시간최적 제어 Time optimal Control via Neural Networks

윤종선, 최원수*

부산대학교 정밀기계공학과 Intelligence Information Control Lab.

Phone : (051)510-2456 Fax : (051)514-0685 E-mail : jsoon@hyowon.cc.pusan.ac.kr

ABSTRACT

A time-optimal control law for quick, strongly nonlinear systems like revolute robots 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 known and unknown systems with constrained inputs and/or states. The nature of neural networks as a parallel processor would circumvent the problem of "curse of dimensionality".

1. 서론

회전관절 로봇처럼 비선형성이 강한 계의 시간최적 제어는 종료시간과 종료상태 오차의 동시 최소화라는 상충의 복잡성 때문에 제어기 개발이 매우 어렵다[1, 2]. 제한된 크기의 입력을 가진 시간최적 제어 문제의 경우 여러점에서의 값이 주어진 비선형미분방정식을 풀어야 하는 비선형 여러점 경계값 문제(multipoint boundary value problem)가 된다. 이 문제는 보통 해석적으로 풀 수 없으므로 수치해석적으로 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 성질지수와 특정의 2차나 3차 선형계의 최적을 다루는 문제들을 빼고는 거의 해석적으로 답을 찾을 수 없다. 따라서 최적제어 문제는 반복계산으로 풀어야 한다[3, 4]. 제한된 크기의 입력을 가지는 최적제어 문제가, 특히 로봇 머니플레이터의 제어기 설계 분야에서 많은 관심을 끌어 왔다[5]. 이 방법들의 성공 여부는 문제의 특성과 높은 초기값의 가정에 달려 있다. 따라서 비선형 높은 차수 문제를 풀, 구현하기 쉽고 체계적인 방법이 필요하다.

하나의 대안으로 대수 문제의 근을 찾는 알고리즘인 Davidenko 법[5, 6]에 따른 새로운 수치해석법이 제안되었다. 유한차분(finite difference)법과 같은 이산화(discretization)과정으로 미분방정식들을 대수방정식의 형태로 바꾸는 이러한 2점 경계값 문제의 풀이법은 여러 점 경계값 문제에도 바로 쓰일 수 있다.

또다른 대안으로 유연한 비선형 mapping인 신경회로망을 시간최적 제어를 학습할 상태 피드백 제어제로 사용한 방법을 제안한다. 제어대상에 대한 선형적 가정

이 없다는 점과 문제 고유의 제한 조건들을 포함한다는 점에서 매우 일반적(general)이다. 또한 제어대상의 정보가 부족할 때도 신경회로망 동정(identification) 기법을 거쳐 시간최적 제어를 할 수 있다. 제안된 방법을 모터 관성계에 적용하여 검증하였다.

2. 시간최적 제어문제

시간 최적 제어기의 목적은 제어계를 모든 제한조건을 만족시키며 가능한한 최소시간에 바람직한 정상상태로 가도록 하는 것이다.

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

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

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

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

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

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

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

이다.

최적해는 Pontryagin의 최대정리(Maximum Principle)기를 써서 끌어낼 수 있다.

hamiltonian

$$H(\mathbf{x}, \underline{\psi}, \mathbf{u}, t) = \sum_{i=0}^n \psi_i f_i(\mathbf{x}, \mathbf{u}) \quad (2.2)$$

에서 상호상태(covariant)방정식과 상태방정식의 표준계

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

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

를 만든다. 최적제어는 \mathbf{x}_0 을 \mathbf{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 \mathbf{x} 을 가지는 변수 \mathbf{x} 와 $\underline{\psi}$ 의 $2n$ 개의 비선형 1차 상미분방정식의 조합 즉 2점 경계값 문제가 된다.

3. 신경회로망에 의한 시간최적 제어

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 계산량은 매우 낮아 적은 메모리만 필요로 한다.

신경 제어계는 디지털로 구현되므로 다음과 같은 이산화 모델(discrete-time formulation)[8]을 써서 미리 오차를 없애도록 한다.

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

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

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

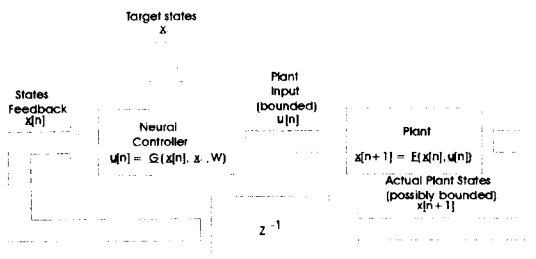


Fig. 1 Control System Structure

일반적으로 목표 상태에 도달할 최소 종료시간과 해당 입력을 모르기 때문에 적절한 제어 법칙은 다음 두 단계로 나누어 찾는다[9].

- (a) 임의의 초기 상태를 목표 상태로 보내는데 드는 최소 시간 간격 K 의 결정
- (b) 최소 시간에 바람직한 상태 변이를 수행할 제어 법칙의 결정

제어기는 sigmoid 비선형성을 가지는 두개의 은닉층과 하나의 선형 출력층으로 이루어진 다층 퍼셉트론 (MLP)으로 정해진다[10].

3.1 종료시간의 최적화

종료시간은 Fig. 2와 같은 반복적 알고리즘(iterative algorithm)[9]을 써서 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 Ψ 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. T 는 500에서 2000을 썼다.

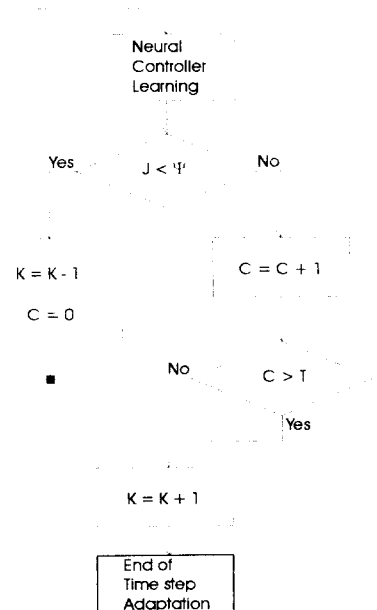


Fig. 2 Basic Structure of the Adaptation Algorithm

3.2 종료상태 오차의 최적화

종료시간 최적화의 과정이 끝나면 추정된 시간으로 평균적인 종료상태 오차 최적화를 위한 제어기 학습과정이 시작된다. 오차가 사용자에 의하거나 오차나 오차 미분항들이 기준값(threshold)에 의해 판단하여 수렴하면 학습 과정을 끝낸다.

종료상태 오차 최적화 알고리즘[9]은 다음과 같다.

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

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

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

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

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

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

반복적으로 계산된다. $\left. \frac{\partial J}{\partial \underline{u}[n]} \right|_{x=\text{const}}$ 의 값이 구해지면

신경회로망으로 역전파(backpropagation)되어 종료상태 오차 J 가 줄어들도록 학습된다. 여기서 BP는 역전파학습(backpropagation learning)[10] 구조를 가진다. 제한된 상태를 가지는 시간최적 제어의 오차 함수 J '는 상태를 벗어난 크기만큼 가중치(penalty)를 더하여 다음과 같이 정의된다.

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

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

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

이 때 ϵ 과 α 로 오차 기울기의 기여도를 조절할 수 있게 한다.

3.3 시스템 동정에 의한 최적화

지금까지와는 달리 제어대상의 정보가 부족할 때 Fig. 3과 같이 신경회로망 동정(identification)을 거쳐 제어를 할 수 있는 방법을 제안한다.

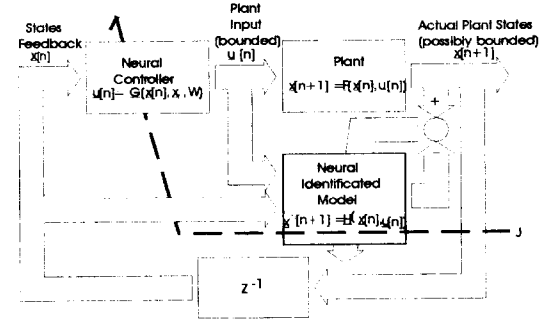


Fig. 3 Neural Network Control for Unknown Plant with Identification

동정에 사용되는 신경회로망은

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

의 입력과 출력의 mapping 관계를 가지며 식 (3.1)과 같이 되도록 학습된다. 여기서 $\underline{W}^{\text{ID}}$ 는 시스템 동정에 쓰인 신경회로망의 가중치이다. 동정에 사용되는 신경회로망의 학습은 실제 대상의 출력과 신경회로망 출력의 오차

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

를 신경회로망에 역전파하여 이루어진다.

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

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

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

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

로 대체하여 종료상태 최적화는 3.2절과 같이 이루어진다. 종료시간의 최적화는 Fig. 2의 과정을 거쳐 이루어진다.

4. 시뮬레이션

예로 쓰인 모터 관성계(double integrator)의 운동 방정식은

$$\tau = J\theta \quad (4.1)$$

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

$$\dot{\underline{x}} = A \underline{x} + Bu \quad (4.2a)$$

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

이다. 이산화 모델[8]은 $J = 0.525 \text{ kg} \cdot \text{m}^2$, $\Delta t = 0.05 \text{ sec}$ 이고 이산화 상태벡터 $\underline{x}[n] = (\theta[n] \ \dot{\theta}[n])^T$ 일 때

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

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

와 같다.

입력 제한이 있을 경우의 종료시간 최적화와 종료상태 오차 최적화 결과는 Fig. 4, Fig. 5, Fig. 6, Fig. 7과 같고 시뮬레이션 파라미터 값은 Table 1과 같다. Pontryagin의 최대원리에서 구한 결과와 비교하였다.

Table 1. Simulation Parameter Values for Optimization with Bounded Input

	Name	Symbol	Value
Plant	Initial position	$\theta[0]$	-0.4667 rad
	Initial velocity	$\dot{\theta}[0]$	0 rad/sec
	Target position	θ_T	0 rad
	Target velocity	$\dot{\theta}_T$	0 rad/sec
	Upper bound of control input	u^{\max}	2 N · m
	Lower bound of control input	u^{\min}	-2 N · m
	Neural Controller	Input nodes	
1st Hidden nodes			10
2nd Hidden nodes			2
Output nodes			1
Learning rate		η	1
Initial final-time step		K	20
Convergence threshold		ψ	0.001
Results	Counter-status variable	C	2000
	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

상태 제한도 있을 경우의 종료시간 최적화와 종료상태 오차 최적화 결과는 Fig. 8, Fig. 9, Fig. 10, Fig. 11과 같고 시뮬레이션 파라미터 값은 Table 2와 같다.

Table 2. Simulation Parameter Values for Optimization with Bounded States

	Name	Symbol	Value
Plant	Upper bound of velocity	θ^{\max}	0.75 rad/s
	Lower bound of velocity	θ^{\min}	-0.75 rad/s
Neural Controller	Convergence threshold	ψ	0.0004
	Counter-status variable	C	10000

나머지 파라미터는 Table 1과 같음

신경회로망 동정에 사용된 파라미터는 Table 3과 같다. 사용된 파라미터에 대한 모델은 종료상태에서 약 0.02 rad의 오차는 학습 간격을 작게하면 줄어든다.

Table 3. Simulation Parameter Values for System Identification

	Name	Symbol	Value
Neural Controller	Input nodes		3
	1st Hidden nodes		10
	2nd Hidden nodes		10
	Output nodes		2
	Learning rate	η	0.005
	Convergence threshold	ψ	0.001
Learning data	Position data	θ	-0.5 ~ 0.001 rad
	Velocity data	$\dot{\theta}$	0 ~ 1.2 rad
	Control input data	u	-2 ~ 2 kg · m

learning data는 0.1 간격으로 제공

입력 제한이 있을 경우 동정 모델에 대한 최적화 결과는 Fig. 12, Fig. 13, Fig. 14, Fig. 15와 같고 이 때 쓰인 파라미터 값은 Table 4와 같다.

Table 4. Simulation Parameter Values for Optimization with Identified Model

	Name	Symbol	Value
Neural Controller	Learning rate	η	0.005
	Initial final-time step	K	20
	Convergence threshold	ψ	0.001
	Counter-status variable	C	2000

신경망의 구조는 Table 1과 같다.

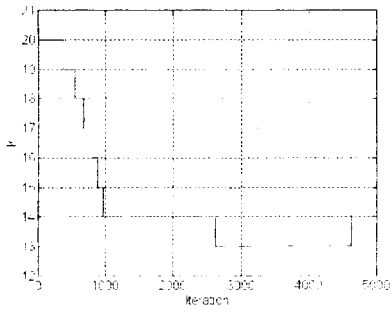


Fig. 4 Evolution of a Typical Estimate of the Minimum Time

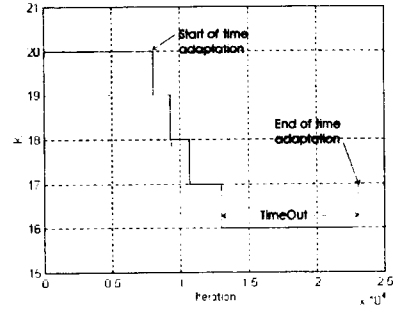


Fig. 8 Evolution of a Typical Estimate of the Minimum Time

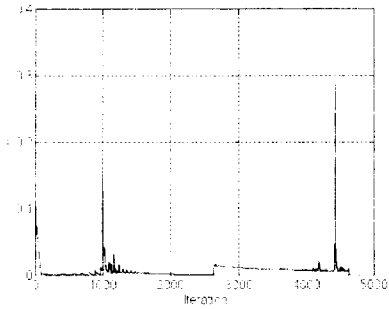


Fig. 5 Evolution of Final State Error

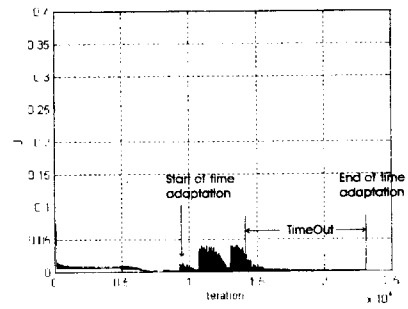


Fig. 9 Evolution of Final State Error

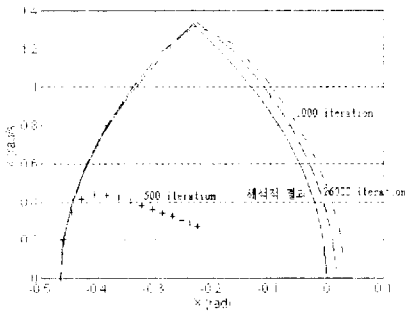


Fig. 6 Evolution of the Phase-plane Plot

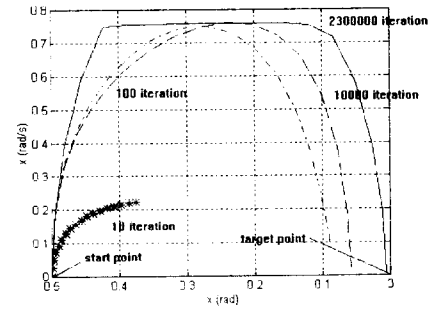


Fig. 10 Evolution of the Phase-plane Plot

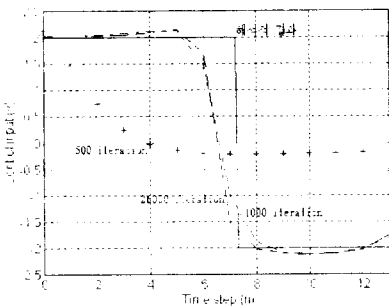


Fig. 7 Evolution of the Control Signal

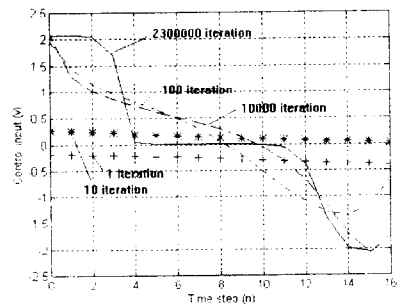


Fig. 11 Evolution of the Control Signal

5. 결론

회전관절 로봇의 운동처럼 비선형성이 강한 계의 시간 최적 제어 방법이 제안되고 검증되었다. 신경회로망에 의한 이 방법은 제한된 크기의 입력이나 상태를 가진 제어대상을 적절히 다루고 있다. 제어대상의 정보가 부족할 때도 신경회로망에 의한 동정(identification) 기법을 사용하여 바람직한 결과를 얻었다. 모터 제어계의 시뮬레이션은 긍정적인 성능을 보여주고 있고 실제 서보 모터 제어계가 구성되어 긍정적인 점검이 이루어지고 있다. 신경회로망의 병렬처리의 속성은 다자유도 로봇과 같은 고자유도계에도 확장 적용되어 "curse of dimensionality" 문제[11]를 피할 수 있을 것이다.

6. 참고문헌

1. Spong, M. W., and Vidyasagar, M., *Robot Dynamics and Control*, Wiley, New York, 1989.
2. Lewis, F. L., Abdallah, C. T., and Dawson, D. M. *Control of Robot Manipulators*, Macmillan, 1993.
3. Athans, M., "The Status of Optimal Control Theory and Applications for Deterministic Systems", *IEEE Transactions on Automatic Control*, Vol. AC-11, No. 5, pp. 580-596, June 1966.
4. Paiewonsky, B., "Optimal Control: A Review of Theory and Practice", *AIJA Journal*, Vol. 3, No. 1, pp. 1985-2006, 1965.
5. Yoon, J. S., "The Numerical Solution of Time-Optimal Control Problems by Davidenko Method", *한국정밀공학회지*, 제12권, 제5호, pp. 57-61, 1995, 5.
6. Davidenko, D. F., "On a New Method for Numerical Solution of Systems of Nonlinear Equations (Russian)", *Doklady Akad. Nauk. SSSR*, Vol. 88, pp. 601-602, 1953 (Journal translated in Soviet Math. Dokl.).
7. Pontryagin, L. S., Boltyanskii, V. G., Gamkrelidze, R. V., and Mishchenko, E. F., *The Mathematical Theory of Optimal Processes*, The Macmillan Press, New York, 1961.
8. Middleton, R. H., and Goodwin, G. C., *Digital Control and Estimation: A Unified Approach*, Prentice-Hall, London, pp. 33-36, 1990.
9. Niesler, T. R. and du Plessis, J. J., "Time-Optimal Control by Means of Neural Networks", *IEEE Control System Magazine*, Vol. 15, No. 5, pp. 23-31, October 1995.
10. Haykin, S., *Neural Networks*, Macmillan, 1994.
11. Bellman, R. E., and Kalaba, R. E., *Dynamic Programming and Modern Control Theory*, Academic Press, New York, 1965.

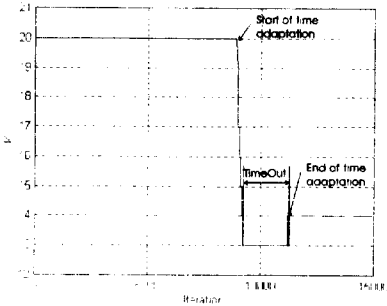


Fig. 12 Evolution of a Typical Estimate of the Minimum Time

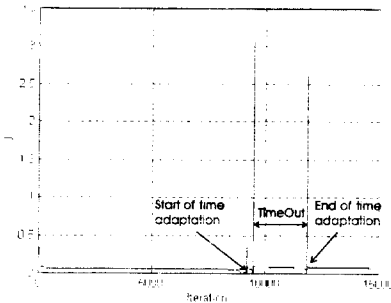


Fig. 13 Evolution of Final State Error

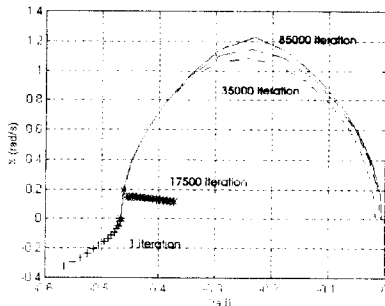


Fig. 14 Evolution of the Phase-plane Plot

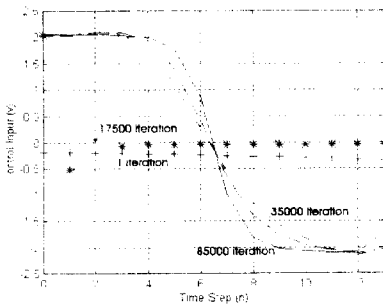


Fig. 15 Evolution of the Control Signal