Proceedings of the 11th
KACC, October 1996

신경회로망을 이용한 모터의 시간최적 제어
Time-optimal Control for Motors via Neural Networks

°최원수*, 윤중선**
부산대학교 정밀기계공학과 Intelligence Information Control Lab.
*(Tel: (051)510-3084; Fax: (051)514-0685; E-mail: wschoi1@hyowon.pusan.ac.kr)
**(Tel: (051)510-2456; Fax: (051)514-0685; E-mail: jsyoon@hyowon.pusan.ac.kr)

Abstracts 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 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". The control law has been demonstrated for a velocity input type motor identified by a genetitic algorithm called GENOCOP.

Keywords Time-optimal Control, Input/States Contraints, Neural Networks, GA based Identification

1. 서론

회전관절 로봇처럼 비선형성이 강한 계의 시간최적 제어는 종료시간과 종료상태 오차의 동시 최소화라는 설계의 복잡성 때문에 제어기 개발이 매우 어렵다[1].

대안으로 유연한 비선형 mapping인 신경회로망을 시간최적 제어를 학습할 상태 피드백 제어계로 사용한 방법이 제안되었다[2, 3]. 제어대상에 대한 선형적 가정이 없다는 점과 문제 고유의 제한 조건들을 포함한다는 점에서 매우 일반적(general)이다. 또한 속도 입력형 모터와 같이 제어대상의 정보가 부족할 때도 유전자 알고리즘에 의한 동정(identification) 기법을 거쳐 시간최적 제어를 할 수 있다. 제안된 방법을 모터 관성계에 적용하여 검증하였다.

2. 신경회로망에 의한 시간최적 제어

신경회로망 제어계[2, 3]

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

$$\mathbf{u}[n] = \mathbf{G}(\mathbf{x}[n], \mathbf{x}_T, W) \quad (2)$$

는 Fig. 1과 같고 $\mathbf{x}[n]$은 시간간격 n일 때의 실제 제어계의 상태, $\mathbf{u}[n]$은 시간 간격 n일 때의 제어 입력, $\mathbf{x}_T$는 제어계의 바람직한 종료상태, W는 제어기의 가중치 행렬을 나타낸다.

일반적으로 목표 상태에 도달할 최소 종료시간과 해당 입력을 모르기 때문에 적절한 제어 법칙은 종료시간 최적화와 종료상태 오차 최적화의 두 단계로 나누어 찾는다[2, 3].

2.1 종료시간의 최적화

종료시간은 반복적 알고리즘(iterative algorithm)[2, 3]을 써서

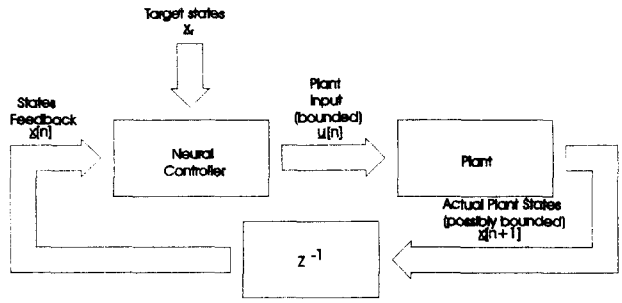

그림 1. 제어계 구조
Fig. 1. Control System Structure

찾는다. 종료시간의 간격 K_i를 실제 값보다 크게 두고 시작한다. 첫 단계는 이 K_i로 계산한 종료상태 오차 함수 J가 수렴 기준값 Ψ보다 작으면 적은 시간 안에 최적화를 시도할 수 있으므로 K_i값을 하나씩 줄인다. 반복 시간 T안에 수렴 조건을 맞추지 못하면 그 시간 간격으로는 최적화가 될 수 없으므로 K_i를 하나 늘여 수정을 마무리 짓는다. 각 K_i의 초기 상태에서 반복 시간을 나타내는 counter/status 변수 C_i를 가진다. 이 변수는 각 최적과정 시작에서 0으로 놓고 반복 시간 T와 비교된다. T는 10000에서 50000을 썼다.

2.2 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[2, 3]은 다음과 같다.

$$J = \frac{1}{2}\| \underline{x}_T - \underline{x}[K]\|^2 = \frac{1}{2}(x_T - x[K])^T(x_T - x[K]) \quad (3)$$

$$\frac{\partial J}{\partial \underline{x}[K]} = -(\underline{x}_T - \underline{x}[K])^T \quad (4)$$

$$\left.\frac{\partial J}{\partial \underline{x}[n]}\right|_{\underline{u}=\text{const}} = \left.\frac{\partial J}{\partial \underline{x}[n+1]}\frac{\partial \underline{x}[n+1]}{\partial \underline{x}[n]}\right|_{\underline{u}=\text{const}} \quad (5)$$

$$\left.\frac{\partial J}{\partial \underline{u}[n]}\right|_{\underline{x}=\text{const}} = \left.\frac{\partial J}{\partial \underline{x}[n+1]}\frac{\partial \underline{x}[n+1]}{\partial \underline{u}[n]}\right|_{\underline{x}=\text{const}} \quad (6)$$

$$\frac{\partial J}{\partial \underline{x}[n]} = BP\left(\left.\frac{\partial J}{\partial \underline{u}[n]}\right|_{\underline{x}=\text{const}}\right) + \left.\frac{\partial J}{\partial \underline{x}[n]}\right|_{\underline{u}=\text{const}} \quad (7)$$

계산은 $\frac{\partial J}{\partial \underline{x}[K]}$로부터 식 (5), (6), (7)의 순으로 반복적으로 계산된다. $\left.\frac{\partial J}{\partial \underline{u}[n]}\right|_{\underline{x}=\text{const}}$의 값이 구해지면 신경회로망으로 역전파(backpropagation)되어 종료상태 오차 J가 줄어들도록 학습된다. 여기서 BP는 역전파학습(backpropagation learning)[4] 구조를 가진다. 제한된 상태를 가지는 시간최적 제어의 오차 함수 J'는 상태를 벗어난 크기만큼 가중치(penalty)를 더하여 다음과 같이 정의된다.

$$J' = \begin{cases} J + \frac{1}{2}\varepsilon(x_i[n] - x_i^{min})^2 & \text{for } x_i[n] < x_i^{min} \\ J & \text{for } x_i^{min} \le x_i[n] \le x_i^{max} \\ J + \frac{1}{2}\varepsilon(x_i[n] - x_i^{max})^2 & \text{for } x_i[n] > x_i^{max} \end{cases} \quad (8)$$

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

$$J'' = \begin{cases} J' + \frac{1}{2}\alpha(u_i[n] - u_i^{min})^2 & \text{for } u_i[n] < u_i^{min} \\ J' & \text{for } u_i^{min} \le u_i[n] \le u_i^{max} \\ J' + \frac{1}{2}\alpha(u_i[n] - u_i^{max})^2 & \text{for } u_i[n] > u_i^{max} \end{cases} \quad (9)$$

이 때 ε과 α로 오차 기울기의 기여도를 조절할 수 있게 한다.

3. 유전자 알고리즘에 의한 계의 모델링

검증에 사용된 모터제어계는 속도 입력형 Yaskawa AC servo motor/servopack USAREM-07CE2K/CACR-SR07AC1ER[5], interface board DR_DAS D/A[6], PC로 구성된다.

속도 입력형 모터의 경우, 직접 토크 입력형 모터의 운동방정식 (1)은

$$\underline{x}[n+1] = \underline{F}^*(\underline{x}[n], \underline{v}[n]) \quad (10a)$$

$$\underline{v}[n] = \underline{V}(\underline{v}[n-1], \underline{u}[n]) \quad (10b)$$

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

속도 입력형 모터는 덧붙여져 필요한 서보 제어기의 변수(parameter) 정보를 얻기 힘드므로, 식 (10)의 선형 이산화 모델

$$\underline{x}[n+1] = A\,\underline{x}[n] + B\,v[n] \quad (11a)$$

$$\underline{v}[n] = \underline{v}[n-1] + \underline{u}[n]\cdot \Delta T \quad (11b)$$

의 A, B 행렬 값을 동정하여 검증에 사용한다. A, B 행렬의 동정은 유전자 알고리즘의 하나인 GENOCOP(GEnetic algorithm for Numerical Optimization for COnstrained Problems)[7]을 써서 구할 수 있다. GENOCOP은 이진화 스트링을 쓰는 SGA(Simple Genetic Algorithms)[7]와 달리 실수값으로 이루어진 스트링을 가진다. 적합도는 각 학습 데이타에 대한 모터 측정값과 스트링 계산값의 오차 제곱 합의 역수를 사용한다.

검증에 쓰인 모터의 GENOCOP 기법에 의한 동정 결과는

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

이다. 동정에 사용된 변수는 Table 1과 같고 알고리즘 반복 횟수에 대한 적합도의 최대 값의 변화는 Fig. 2와 같다.

표 1. 제어계 동정을 위한 모의실험 변수
Table 1. Simulation Parameter for Plant Identification

	Name	Symbol	Value
Learning Data	Position	θ	-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^2(Δu=2)
GENOCOP Parameters	Population		50
	Mutation	*Pm*	0.3
	Crossover	*Pc*	0.9
	Constraints	A	-2~2
		B	-1~1

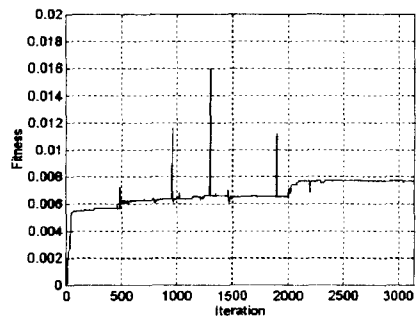

그림 2. 적합도 최대 값의 변화
Fig. 2. Evolution of a Maximum Fitness Value

4. 검증

입력 제한이 있을 경우의 종료시간 최적화와 종료상태 오차 최적화 시뮬레이션 결과는 Fig. 3, Fig. 4, Fig. 5, Fig. 6과 같고 시뮬레이션 파라미터 값은 Table 2와 같다.

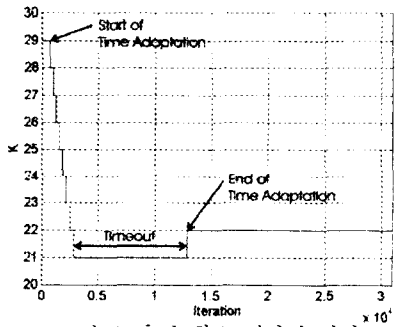

그림 3. 추정 최소 시간의 변화

Fig. 3. Evolution of a Typical Estimate of the Minimum Time

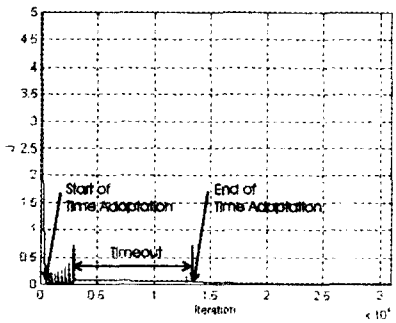

그림 4. 종료상태 오차의 변화

Fig. 4. Evolution of Final State Error

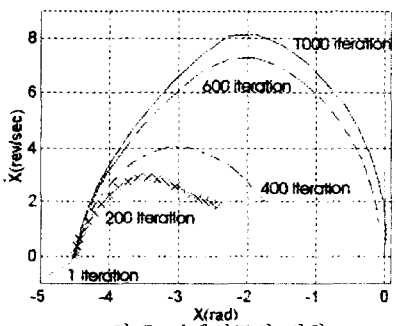

그림 5. 상태선도의 변화

Fig. 5. Evolution of the Phase-plane Plot

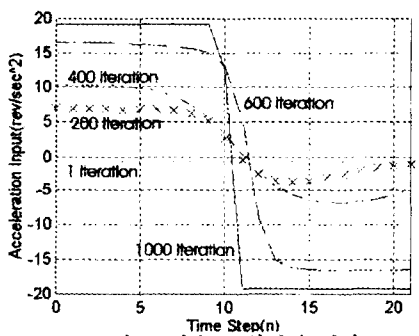

그림 6. 가속도 입력의 변화

Fig. 6. Evolution of the Acceleration Input

표 2. 입력 제한이 있는 시간최적 제어의 시뮬레이션 파라미터

Table 2. Simulation Parameter Values for Optimization with Bounded Input

	Name	Symbol	Value
Plant	Initial Position	$\theta[0]$	-4.4995 rad
	Initial Velocity	$\dot{\theta}[0]$	0 rev/sec
	Target Position	θ_T	0 rad
	Target Velocity	$\dot{\theta}_T$	0 rev/sec
	Bound of Acceleration Input	u^{max}	20 rev/sec^2
		u^{min}	-20 rev/sec^2
Neural Controller	Input Nodes		2
	1st Hidden Nodes		10
	2nd Hidden Nodes		2
	Output Nodes		1
	Learning Rate	η	0.5
	Initial Final-time Step	K	29
	Convergence Threshold	Ψ	0.001
	Counter/status Variable	C	10000

입력 제한이 있을 때의 시뮬레이션 결과와 비교된 실제 모터 제어 시스템에 적용한 결과는 Fig. 7, Fig. 8과 같다.

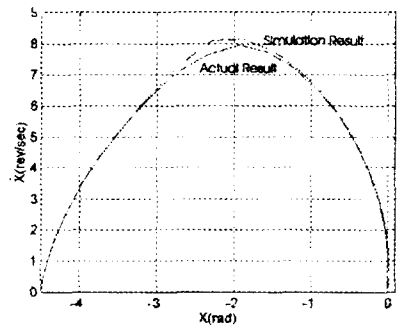

그림 7. 상태선도의 변화

Fig. 7. Evolution of the Phase-plane Plot

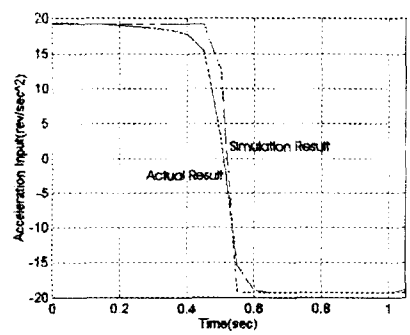

그림 8. 가속도 입력의 변화

Fig. 8. Evolution of the Accelleration Input

상태 제한도 있을 경우의 종료시간 최적화와 종료상태 오차 최적화 결과는 Fig. 9, Fig. 10, Fig. 11, Fig. 12와 같고 시뮬레이션 파라미터 값은 Table 3과 같다.

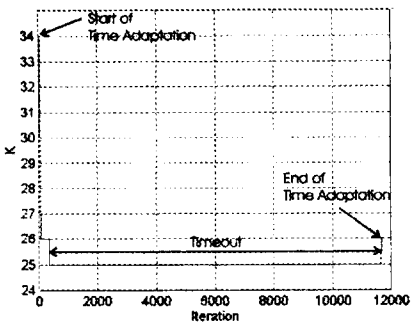

그림 9. 추정 최소 시간의 변화

Fig. 9. Evolution of a Typical Estimate of the Minimum Time

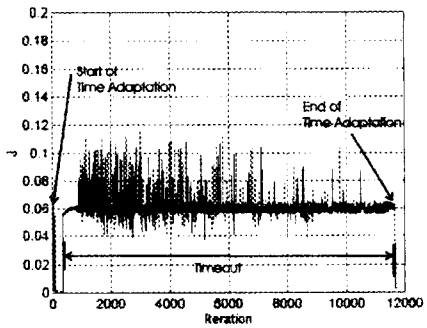

그림 10. 종료상태 오차의 변화

Fig. 10. Evolution of Final State Error

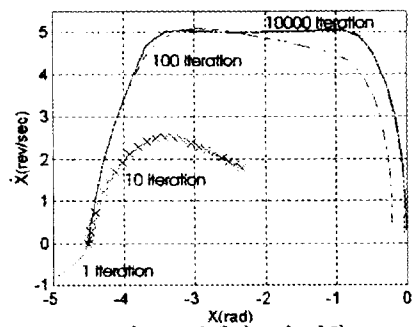

그림 11. 상태선도의 변화

Fig. 11. Evolution of the Phase-plane Plot

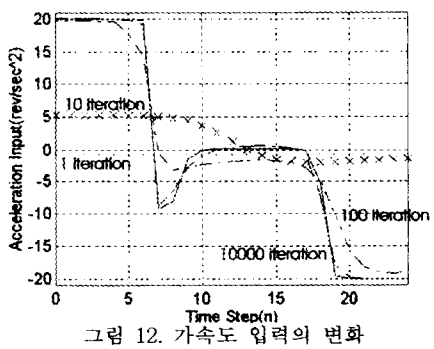

그림 12. 가속도 입력의 변화

Fig. 12. Evolution of the Acceleration Input

표 3. 입력/상태 제한이 있는 시간최적 제어 시뮬레이션 파라미터

Table 3. Simulation Parameter Values for Optimization with Bounded Input and States

	Name	Symbol	Value
Plant	Upper Bound of Velocity	θ^{max}	5 rev/sec
	Lower Bound of Velocity	θ_{min}	-5 rev/sec
Neural Controller	Convergence Threshold	Ψ	0.001
	Counter/status Variable	C	10000

나머지 파라미터는 Table 2와 같음

5. 결론

회전관절 로봇의 운동처럼 비선형성이 강한 계의 시간최적 제어 방법이 제안되고 검증되었다. 신경회로망에의한 이 방법은 제한된 크기의 입력이나 상태를 가지는 제어대상을 적절히 다루고 있다. 제어대상의 정보가 부족한 속도입력형 모터의 경우 GENOCOP과 같은 유전자 알고리즘에 의한 모델링 기법을 써서 바람직한 결과를 얻었다. 모터 제어계의 시뮬레이션과 실험은 긍정적인 성능을 보여주고 있다. 신경회로망의 병렬처리의 속성은 다자유도 로봇과 같은 고자유도계에도 확장 적용되어 "curse of dimensionality" 문제[8]를 피할 수 있을 것이다.

6. 참고문헌

1. Spong, M. W., and Vidyasagar, M., *Robot Dynamics and Control*, Wiley, NewYork, 1989.

2. 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. 23-33, October 1995.

3. 윤중선, 최원수, "신경회로망을 이용한 시간최적 제어", 대한정밀공학회 춘계 학술대회 논문집, pp. 372-377, 1996년 5월.

4. Haykin, S., *Neural Networks*, Macmillan, 1994.

5. *AC Servo Drive Manual*, Yaskawa, 1993.

6. 다림 시스템, [DR_DAS 12] 사용자 설명서, 1992.

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

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