

시스템의 불확실성에 대한 신경망 모델을 통한 강인한 비선형 제어

論文
43~5~18

A Robust Nonlinear Control Using the Neural Network Model on System Uncertainty

李 壽 榮*·鄭 明 振**
(Soo-Yeong Yi·Myung-Jin Chung)

Abstract—Although there is an analytical proof of modeling capability of the neural network, the convergency error in nonlinearity modeling is inevitable, since the steepest descent based practical learning algorithms do not guarantee the convergency of modeling error. Therefore, it is difficult to apply the neural network to control system in critical environments under an on-line learning scheme. Although the convergency of modeling error of a neural network is not guaranteed in the practical learning algorithms, the convergency, or boundedness of tracking error of the control system can be achieved if a proper feedback control law is combined with the neural network model to solve the problem of modeling error. In this paper, the neural network is introduced for compensating a system uncertainty to control a nonlinear dynamic system. And for suppressing inevitable modeling error of the neural network, an iterative neural network learning control algorithm is proposed as a virtual on-line realization of the Adaptive Variable Structure Controller. The efficiency of the proposed control scheme is verified from computer simulation on dynamics control of a 2 link robot manipulator.

Key Words : Adaptive VSC, Neural Network, Ultimate Boundedness, Iterative

1. 서 론

시스템 제어분야에 신경망을 도입하려는 노력이 근래에 점차 증가되고 있다. 시스템 제어분야에서의 신경망의 응용은 주로 신경망이 가지는 비선형성 모델 능력을 이용하는 것이다. 최근에 Hornik[1]등에 의하여 단순 비선형 소자의 다층 병렬 구조를 가지는 신경 회로망이 임의의 비선형성을 원하는 정도까지 모사할 수 있음이 해석적으로 입증되고, Rumelhart[2]등에 의하여 현실적인 그의 학습방법이 제안되면서 신경망 공

학자들은 시스템 특성에 대한 엄밀한 수학적 모델을 필요로 하던 기존의 제어 분야에 신경망이라는 통일된 방법을 채택하려고 시도하고 있다. Gradient 방식에 기초한 신경망 학습 방식은 시스템의 입출력 데이터에 의한 일종의 적응성을 제공하기 때문에, 주변 환경의 변화나 구성 부품 등의 노화에 기인하는 시스템의 시변성에 대한 적응 제어에 적용될 수 있다.

그렇지만 신경망의 모델 능력에 대한 해석적인 입증에도 불구하고 gradient 방식에 근거한 현실적 학습 방식은 모델 오차의 수렴성에 대해 아직까지 확실한 근거를 제공하지 못하고 있으며, 이러한 모델 오차는 실제 제어 적용시에 큰 장애가 될 수 있다. 모델 오차의 수렴성 문제를 좀더 자세히 살펴보기 위해서 먼저 다음을 생각해 보자. 신경망 지도 학습에 일반적으로 사용되는 방식은 학습의 목적 함수에 따라 (i) batch

*正 會 員 : KAIST 電氣 및 電子工學科 博士課程

**正 會 員 : KAIST 電氣 및 電子工學科 教授·工博

接受日字 : 1993年 10月 25日

1 次修正 : 1994年 2月 18日

2 次修正 : 1994年 3月 31日

학습과 (ii) pattern 학습으로 나눌 수 있다. Batch 학습은 목적 함수

$$J_b = \frac{1}{2} \int_0^{\infty} (\Delta(x)) - \hat{\Delta}(x, w))^2 dt \quad (1)$$

가 신경망 연결자 w 에 대해서 최소가 되도록 w 를 조정해 나가는 방법이다. 여기서 $\hat{\Delta}(x, w)$ 는 비선형 함수 $\Delta(x)$ 를 근사화하기 위한 신경망 모델이다. 이 방식은 기본적으로 오프라인 개념이므로 미리 신경망을 학습시켜 놓은 후에 실제 제어에 투입하는 것이다. 이 경우 오프라인 학습을 위한 데이터 수집 문제와 함께 오프라인 학습 결과가 실제 제어시에도 완전히 학습한대로 움직일 지에 대한 보장은 할 수 없다. 더우기 오프라인 방식은 신경망의 적응 능력을 활용할 수 없는 방식이 된다. 한편 Pattern 학습의 목적 함수는

$$J_p = \frac{1}{2} (\Delta(x) - \hat{\Delta}(x, w))^2 \quad (2)$$

이다. (2)를 시간에 대해 미분하면

$$\begin{aligned} \frac{dJ_p}{dt} &= (\Delta(x) - \hat{\Delta}(x, w)) \left(\frac{d\hat{\Delta}(x)}{dt} - \frac{d\hat{\Delta}(x, w)}{dt} \right) \\ &= (\Delta(x) - \hat{\Delta}(x, w)) \left(\frac{d\hat{\Delta}(x)}{dt} - \frac{\partial \hat{\Delta}(x, w)}{\partial x} \frac{dx}{dt} \right. \\ &\quad \left. - \frac{\partial \hat{\Delta}(x, w)}{\partial w} \frac{dw}{dt} \right) \quad (3) \\ &= J_{px} + J_{pw}, \end{aligned}$$

이고, 여기서

$$\begin{aligned} J_{px} &= (\hat{\Delta}(x) - \hat{\Delta}(x, w)) \left(\frac{d\hat{\Delta}(x)}{dt} - \frac{\partial \hat{\Delta}(x, w)}{\partial x} \frac{dx}{dt} \right) \\ J_{pw} &= -(\hat{\Delta}(x) - \hat{\Delta}(x, w)) \frac{\partial \hat{\Delta}(x, w)}{\partial w} \frac{dw}{dt} \end{aligned}$$

이다. 그러므로 w 를

$$\frac{dw}{dt} = \alpha \frac{\partial \hat{\Delta}(x, w)}{\partial w} (\hat{\Delta}(x) - \hat{\Delta}(x, w)), \quad \alpha > 0 \quad (4)$$

에 따라 조정할 때 J_{pw} 는

$$J_{pw} = -\alpha \left(\frac{\partial \hat{\Delta}(x, w)}{\partial w} \right) (\hat{\Delta}(x) - \hat{\Delta}(x, w))^2$$

$$\leq 0, \quad \forall t$$

이 된다. 주목할 것은, 이것이 $\frac{dJ_p}{dt} \leq 0, t$ 임을 의미하지는 않는다는 것이다. 하지만 J_{px} 는 w 로 조절할 수 있는 항이 아니므로 어쩔 수 없고, 따라서 (4)는 잘 알려져 있듯이 흔히 δ -rule, 혹은 오차 역전달 학습이라는 이름으로 사용되는 알고리즘이다. 이 경우는 기존의 적응 제어 알고리즘 처럼 기본적으로 온라인 방식이며, batch 학습의 경우보다 local minimum에 빠질 확률이 적기 때문에 흔히 많이 채택된다. 그러나 보통 신경망은 학습의 과정에서 local minimum에 빠질 가능성도 있으며, (4)와 같은 pattern 학습 알고리즘에 의해서도 $\frac{dJ_p}{dt}$ 가 항상 음이 되지 않는다. 즉 J_p 가 시간에 대해서 단조 감소하지 않으므로 Lyapunov 함수 처럼 행동하지 못하고 이는 전체 폐루프 시스템의 안정성을 보장하지 못하는 이유가 된다. 따라서 많은 제어 공학자들이 중요한 환경에서 실제로 신경망을 이용하는 접근 방식을 채택함에 주저하고 있는 것이 사실이다. 비록 모델 오차의 수렴성은 보장할 수 없다 하더라도, 모델 오차 문제를 해결하기 위해 적절한 제한 제어 방식이 신경망에 결합된다면 제어 시스템의 추종 오차의 수렴성, 혹은 최소한 추종 오차가 어떤 한계값내에 항상 머무르도록 하는 유계성은 얻을 수 있다. 그러한 제한 제어 방식으로서 시스템의 불확실성을 극복하기 위한 강인 제어 방식으로 많이 연구되고 있는 가변 구조 제어(variable structure control)[3][4][5][6] 방식은 신경망의 불가피한 모델 오차에 대한 한가지 해결책이 될 수 있다. 즉, 시스템의 불확실성에 대한 모델 도구로서 신경망을 이용하고, 신경망의 모델 오차에 대한 극복 방법으로서 가변 구조 제어 방식을 결합하는 것이다. 신경망의 응용이라는 입장에서 보면 이것은 신경망의 불가피한 모델 오차와 그의 적용에 따르는 불안정성에 대한 해결책이 되고, 가변 구조 제어 방식의 입장에서 보면 불확실성에 대한 모델의 도입으로 기존 가변 구조 제어 방식에서의 제어 입력의 스위칭량을 줄여서 chattering 문제를 해결하며, 입력 에너지를 줄이는 장점을 얻을 수 있다.

본 논문에서는 신경망을 이용하여 시스템 불확실성의 모델을 구하고, 이때 나타나는 모델 오차를 가변 구조 제어 방식으로 처리하므로써, 정

확한 수학적 모델을 구하기 어려운 비선형 시스템의 제어 효율을 높이는 제어 알고리즘을 제시하는데 그 목적이 있다.

2. 적응적 가변 구조 제어기의 안정성 해석

먼저 다음과 같은 2차의 단일 입력 동적 시스템을 생각해 보자.

$$\ddot{x} = f_n(x) + \Delta(x) + v \tag{5}$$

여기서 x 는 $x = [x, \dot{x}]'$ 이고, $f_n(x)$ 는 시스템의 알려진 공칭 동력학이며, $\Delta(x)$ 는 구조적, 비구조적 불확실성을 망라하는 일괄적인 불확실성이다. 그리고 시스템 (5)에 대한 원하는 궤적을 x_d 라고 하고, $e = x - x_d$ 를 추종 오차로, 그리고 오차 위상 평면에 $s = e + \lambda_1 e$, $\lambda_1 > 0$ 을 슬라이딩 직선으로 정의한다. 또 불확실성 $\Delta(x)$ 에 대한 모델의 출력력을 $\hat{\Delta}(x)$ 라 하자. 예로써 $\Delta(x)$ 가 x 에 대해서 선형임이 알려져 있는 경우는 $\hat{\Delta}(x)$ 도 선형으로 잡는 것이 적합할 것이고, 혹은 $\Delta(x)$ 가 비선형적이라 하더라도 그의 구조와 비선형 요소가 완전히 알려져 있는 경우는 매개 변수의 재편성(reparameterization) 과정을 통하여 구조적으로는 같고 매개 변수만 모르는 모델 $\hat{\Delta}(x)$ 을 택할 수도 있다. 또 구조적으로도 모르는 불확실성의 경우에도 만일 그의 주파수 스펙트럼의 입장에서 최고조파의 주파수만 안다면 유한 푸리에 급수를 통한 모델 $\hat{\Delta}(x)$ 를 만들거나 혹은 어떤 경우든 직교 함수들의 집합(orthogonalized basis)을 통한 모델을 만들 수도 있다. 후자의 대표적인 것이 다항식 모델(polynomial model)이다. 매개 불확실성에 대한 아무런 정보도 없을 때는 퍼지 모델이나 혹은 신경망 모델이 적합할 것이다.

지금 유한 상수 $\hat{\Delta}$ 를 $|\hat{\Delta}(x) - \Delta(x)|$ 에 대한 예측값이라고 하자. 이때 다음과 같은 제어 법칙과 시스템 (5)에 의해서 이루어지는 페루프 시스템에 대해서 안정성에 관한 정리를 보일 수 있다:

$$u = -f_n(x) - \hat{\Delta}(x) - u_s \text{sgn}(s) + \ddot{x}_d - \lambda_1 \dot{e} - \eta s, \quad \eta > 0 \tag{6}$$

$$\alpha = -\lambda_2(u_s - \hat{\Delta}) + |s|, \quad \lambda_2 > 0 \tag{7}$$

정리 1 (안정성) 만일 유한 상수 $\hat{\Delta}$ 가 $\hat{\Delta} \geq \sup_x |(\Delta(x) - \hat{\Delta}(x))|$ 이면, 제어입력 (6)과 (7)과 시스템 (5)에 의해서 이루어지는 페루프 시스템은 점근적으로 안정하게 된다.

증명; 시스템 (5)와 제어 입력 (6), (7)식에 의한 페루프 시스템의 Lyapunov 함수 후보를

$$V(s, u_s) = \frac{1}{2}s^2 + \frac{1}{2}(u_s - \hat{\Delta})^2 \tag{8}$$

로 잡자. (8)식을 시간에 대해 미분하고 (5), (6), 그리고 (7)식을 대입하면

$$\begin{aligned} \dot{V} &= s\dot{s} + (u_s - \hat{\Delta})\dot{u}_s \\ &= s(\ddot{x} - \ddot{x}_d + \lambda_1 \dot{e}) + (u_s - \hat{\Delta})\{-\lambda_2(u_s - \hat{\Delta}) + |s|\} \\ &= -\eta s^2 + s(\Delta(x) - \hat{\Delta}(x) - u_s \text{sgn}(s) - \lambda_2(u_s - \hat{\Delta}))^2 + (u_s - \hat{\Delta})|s| \\ &\leq -\eta s^2 - \lambda_2(u_s - \hat{\Delta})^2 - |s|(\hat{\Delta} - |\Delta(x) - \hat{\Delta}(x)|) \end{aligned} \tag{9}$$

가 되어, $\hat{\Delta} \geq \sup_x |(\Delta(x) - \hat{\Delta}(x))|$ 이면 평형점

을 제외한 모든 s 와 u_s 에 대해서 $\dot{V} < 0 \forall t$ 이고, 따라서 페루프 시스템의 평형점 $s=0, u_s=\hat{\Delta}$ 은 점근적으로 안정하게 된다. □

여기서 신경망의 모델 오차가 항상 $\hat{\Delta} = 0 \geq |\Delta(x) - \hat{\Delta}(x)| = 0$ 이면 평형점에서 $u_s = 0$ 이므로 제어 입력 (6)식의 스윛칭항이 사라진다. 즉 $\Delta(x)$ 에 대한 모델 $\hat{\Delta}(x)$ 가 정확할 수록 제어 입력의 스윛칭항이 작아져도 된다는 것이다. 점근적 안정을 위한 정리 1의 조건, 즉 불확실성

$$\text{모델 오차의 예측값이 } \hat{\Delta} \geq \sup_x |\Delta(x) - \hat{\Delta}(x)|$$

이어야 한다는 것은 학습 과정에서 미리 예측값을 알아야한다는 것이 되므로 다소 강한 조건이 된다. 그러나 이 조건이 만족되지 않을 때에 대해서도 페루프 시스템의 추종 오차가 어떤 상수값으로 크기가 제한됨을 다음과 같은 보조 정리를 통해서 알 수 있다:

따름 정리 : (유계성) $\hat{\Delta} < \sup_x |(\Delta(x) - \hat{\Delta}(x))|$

$|\Delta(x) - \hat{\Delta}(x)| < \infty$ 인 경우, 시스템 (5)와 제어 입력 (6), (7)로 이루어지는 페루프 시스템은 유계(ultimate boundedness)된다.

증명; (8)과 같은 Lyapunov 함수 후보에 대해서 (9)식으로 부터

$$\dot{V} \leq -\eta s^2 - \lambda_2 (u_s - \hat{D})^2 + |s| |\Delta(x) - \hat{D}(x)| - \dot{\hat{D}} \quad (10)$$

이 된다. 따라서 모든 u_s 와 $|s| > \phi$ 에서 $\dot{V} < 0$, t 가 되는 유한 상수 ϕ 가 존재함을 알 수 있다. 이는 페루프 제어 시스템이 유계됨을 말해 준다 [8][9][10]. □

신경망을 이용하여 제어 입력 (6)과 (7)을 구현하기 위해 본 논문에서는 다음과 같은 반복 학습 제어의 개념을 제안한다. 먼저 다음의 용어를 정의한다. 시스템 (5)에 대한 제어 작업을 여러번(k) 반복 수행한다고 하고 그 주기를 T 라고 할 때,

- $W_{k=n}$; n 번째 작업 수행시 사용되는 신경망
- $\hat{D}(x, W_{k=n}) = \hat{D}_{k=n}$; 불확실성 $\Delta(x)$ 에 대한 신경망 $W_{k=n}$ 의 모델값
- $\tilde{\Delta}_{k=n} \geq |(\Delta(x) - \hat{D}(x, W_{k=n}))|$, 즉 신경망 모델 $\hat{D}(x, W_{k=n})$ 의 최대 모델 오차
- Δ_{max} ; $\Delta_{max} > |\Delta(x)|$, 즉 불확실성의 최대값
- $\Delta^-(x)$; 신경망 학습을 위한 기준 신호, 즉 $\Delta^-(x) = \ddot{x} - f_n(x) - u$ 이다.

• 반복 학습 제어 알고리즘 :

$k=0$ 에서; ($t=0^-$)

(i) 신경망 weight 변수들을 초기화한다. 그것을 $W_{k=0}$ 이라 하자.

$k=1$ 에서; ($0 \leq t \leq T^-$)

(i) 제어 입력

$$u = -fn(x) - u_s \text{sgn}(s) + \ddot{x}_d - \lambda_1 \dot{e} - \eta s, \eta > 0 \quad (11)$$

$$\dot{u}_s = -\lambda_2 (u_s - \Delta_{max}) + |s| \quad (12)$$

(ii) 이 주기 동안 신경망 $W_{k=1}$ 의 최대 모델 오차 $\tilde{\Delta}_{k=1}$ 을 찾는다. 즉,

$$\Delta_{k=1} = \max_{0 \leq t \leq T^-} |\Delta^-(x) - \hat{\Delta}_{k=1}| \quad (13)$$

이다.

(iii) $W_{k=1}$ 으로부터 pattern 학습을 통해 $W_{k=2}$ 를 구한다. 즉

$$\frac{dW_{k=1}}{dt} = \alpha \cdot \frac{\partial \hat{\Delta}_{k=1}}{2W_{k=1}} \cdot (\hat{\Delta}(x) - \hat{\Delta}_{k=1}), \alpha > 0 \quad (14)$$

이고, $t=T$ 에서 $W_{k=2} = W_{k=1}$ 로 만든다.

$k=n$ 에서; ($(n-1)T \leq t \leq n/T^-$)

(i) 제어 입력

$$u = -fn(x) - \tilde{\Delta}_{k=n} - u_s \text{sgn}(s) + \ddot{x}_d - \lambda_1 \dot{e} - \eta s \quad (15)$$

$$u_s = -\lambda_2 (u_s - \tilde{\Delta}_{k=n-1}) + |s| \quad (16)$$

(ii) 이 주기 동안 신경망 $W_{k=n}$ 의 최대 모델 오차 $\tilde{\Delta}_{k=n}$ 을 구한다. 즉,

$$\Delta_{k=n} = \max_{(n-1)T \leq t \leq nT^-} |\Delta^-(x) - \hat{\Delta}_{k=n}| \quad (17)$$

이다.

(iii) $W_{k=n}$ 으로부터 pattern 학습을 통해 $W_{k=n+1}$ 을 구한다. 즉,

$$\frac{dW_{k=n}}{dt} = \alpha \cdot \frac{\partial \hat{\Delta}_{k=n}}{2W_{k=n}} \cdot (\hat{\Delta}(x) - \hat{\Delta}_{k=n}), \alpha > 0 \quad (18)$$

이고, $t=nT$ 에서 $W_{k=n+1} = W_{k=n}$ 로 만든다.

$k=n+1$ 에서; ($nT \leq t \leq (n+1)T^-$)

위 $k=n$ 과정에서 n 을 $n+1$ 로 대체하고 반복한다. □

여기서 주의할 것은 (11)식에는 보상항 $\hat{\Delta}_{k=1}$ 이 없다는 것이다. 정의에 의해서 $\hat{\Delta}_{k=1}$ 은 $k=0$ 에서 초기화된 신경망 $W_{k=1}$ 의 출력이므로 보상기로서의 효율이 없다.

위 알고리즘은 그림 1에서 보듯 2개의 신경망 쌍을 가지고 작업하면서 하나는 작업 시행중 페루프 시스템 내에서 보상기로 사용하고, 또 하나는 페루프 시스템의 바깥에서 단지 시스템의 데이터만으로 불확실성을 학습하며, 한 주기의 작업이 끝나는 시점에서 학습한 신경망의 weight로 보상기의 weight를 대체하는 의사 온라인 방식으로 생각할 수 있다. $k=n$ 번째 작업중인 신

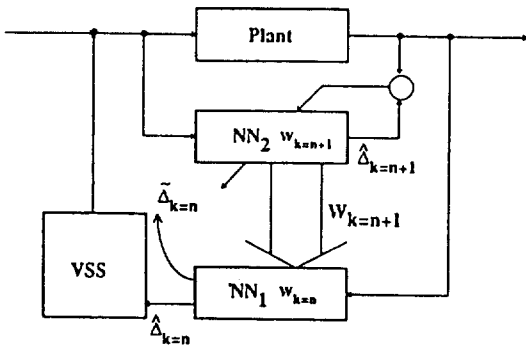


그림 1 주기 반복 과정을 통한 신경망의 학습
Fig. 1 The neural network learning through iterative scheme

경망 불확실성 모델의 최대 모델 오차로서 $k=n-1$ 번째에서 구한 신경망 모델 $\hat{\Delta}_{k=n-1}$ 의 최대 모델 오차 $\tilde{\Delta}_{k=n-1}$ 을 사용하는 것이다. $k=n$ 번째에 불확실성 모델로 사용되는 $W_{k=n}$ 은 제어 과정중 학습하지 않고, 똑같은 weight 값으로 시작하는 또다른 신경망이 페루프 시스템의 바깥에서 학습되므로 (8)식이 페루프 시스템의 Lyapunov 함수일 조건을 만족하고 정리 1을 적용할 수 있다. 만일 $k=n$ 에서 사용되는 $W_{k=n}$ 에 직접 온라인 학습이 가해진다면 학습 알고리즘 (18)이 페루프 시스템에 또다른 상태를 제공하므로 (8)식은 전술한 pattern 학습의 비단조 감소성 때문에 전체 페루프 시스템의 Lyapunov 함수 후보가 될 수 없고, 따라서 시스템의 안정성을 보이기가 어렵게 된다.

물론 이 경우에 정리 1의 안정성에 대한 증명 이 성립하기 위해서는 $k=n$ 번째 수행의 제어 작업 한 주기 동안 항상 $|\Delta(x) - \hat{\Delta}_{k=n}(x)| \leq \tilde{\Delta}_{k=n-1}$ 이 보장되어야 한다. 그러나 신경망의 모델 능력에 대한 오랜 시뮬레이션과 여러 학자들의 견해에 비추어 볼 때, gradient 방식에 기본을 둔 학습 방법은 비록 그 학습 오차 곡선이 매 시간 sample에 대해서 단조 감소하지는 않더라도, 주기 반복 수행의 경우 처럼 여러 sample 간격을 두고 학습 오차를 볼 때는 어느 정도 단조 감소성이 있다고 볼 수 있다. 즉 $T=nt - (n-1)t \gg 0$ 일때, 주기 학습 최대 오차 곡선이 $\tilde{\Delta}_{k=n-1} \geq \tilde{\Delta}_{k=n}$ 이다. 또 pattern 학습의 step size α 가 작을수록 단조 감소하는 경향이 커짐이 알려져 있다[11]. 그리고 비록 주기 최대 학습 오차가 단조 감소하지 않을 때도 앞의 따름 정리로

부터 상태 오차의 크기가 유한하게 제한됨을 보일 수 있다. 즉 시스템 (5)와 제어 입력식 (15), (16)으로 이루어지는 $k=n, n=2, 3, \dots$ 에 대한 페루프 시스템의 Lyapunov 함수 후보 (8)을 분석해보면

$$V(s, u_s) = \frac{1}{2}s^2 + \frac{1}{2}(u_s - \Delta_{k=n-1})^2 \quad (19)$$

에서

$$\begin{aligned} \dot{V} &= \dot{s}s + (u_s - \tilde{\Delta}_{k=n-1})u_s \\ &= s(\ddot{x} - \ddot{x}_d + \lambda_1 e) - (u_s - \tilde{\Delta}_{k=n-1})\{-\lambda_2(u_s - \tilde{\Delta}_{k=n-1}) + |s|\} \\ &= -\eta s^2 + s(\Delta(x) - \Delta_{k=n} - u_s \text{sgn}(s)) \\ &\quad - \lambda_2(u_s - \tilde{\Delta}_{k=n-1})^2 + (u_s - \tilde{\Delta}_{k=n-1})|s| \\ &\leq -\eta s^2 - \lambda_2(u_s - \tilde{\Delta}_{k=n-1})^2 + (\tilde{\Delta}_{k=n} - \tilde{\Delta}_{k=n-1})|s| \end{aligned} \quad (20)$$

가 된다. 따라서 이는

(a) $\tilde{\Delta}_{k=n-1} \geq \tilde{\Delta}_{k=n}$, 즉 주기 학습의 최대 오차 곡선이 감소한 주기에서는 (20)이 항상 음이되어 페루프 시스템은 점근적으로 안정하게 되고,

(b) 어떤 $k=n$ 의 경우에 $\tilde{\Delta}_{k=n-1} < \tilde{\Delta}_{k=n}$ 일 때, 즉 최대 오차가 오히려 증가한 주기에서는 따름 정리로 부터 어떤 유한 상수 ϕ 값으로 추종 오차가 유계됨을 알 수 있다. 더우기 정성적으로 말해서, 식 (16)으로 부터 정상 상태에서

$$u_s - \tilde{\Delta}_{k=n-1} \approx \frac{|s|}{\lambda_{k=2}}$$

시스템의 오차의 크기는 $\frac{\lambda_2}{1 + \lambda_2 \eta} (\tilde{\Delta}_{k=n-1} - \tilde{\Delta}_{k=n})$ 으로 제한될 것임을 짐작할 수 있다[9].

물론 $k=1$ 일 때도 페루프 시스템의 Lyapunov 함수 후보를 $V(s, u_s) = \frac{1}{2}s^2 + \frac{1}{2}(u_s - \Delta_{\max})^2$ 로 취하므로써 추종 오차의 점진적 안정성을 보일 수 있다.

3. 컴퓨터 시뮬레이션

위의 3. 1절에서 제안한 제어 규칙과 그 구현 알고리즘을 입증해 보기위해 간단한 2축 로봇 매니퓰레이터의 동력학 제어에 적용하여 보았다. 2축 로봇의 동력학은 다음식 (21)와 같이 표현되며[7] 불확실성 항은 (22)로서 크기가 제한된 임의의 비선형성으로 하였다. 이는 제한 보상에 의해서 각 축의 입력을 기준으로 (5)식과 같은

형태의 다변수 시스템으로 decoupling될 수 있다.

$$\begin{aligned}
 & \begin{bmatrix} m_2 l_2^2 + 2m_2 l_1 l_2 C_3 + (m_1 + m_2) l_1^2 + J_1 \\ m_2 l_1 l_2 C_2 + m_2 l_2^2 \end{bmatrix} \\
 & \begin{bmatrix} m_2 l_2^2 + m_2 l_1 l_2 C_2 \\ m_2 l_2^2 + J_2 \end{bmatrix} \begin{bmatrix} \theta'_1 \\ \theta'_2 \end{bmatrix} \\
 & + \begin{bmatrix} -m_2 l_1 l_2 S_2 \theta_2^2 - 2m_2 l_1 l_2 S_1 \theta_1 \theta_2 \\ m_2 l_1 l_2 S_2 \theta_2^2 \end{bmatrix} \\
 & + \begin{bmatrix} m_2 l_2 C_{12} G + (m_1 + m_2) l_1 G C_1 \\ m_2 l_2 G C_{12} \end{bmatrix} + \begin{bmatrix} \Delta_1 \\ \Delta_2 \end{bmatrix} \\
 & = \begin{bmatrix} T_1 \\ T_2 \end{bmatrix} \tag{21}
 \end{aligned}$$

$$\begin{bmatrix} \Delta_1 \\ \Delta_2 \end{bmatrix} = \begin{bmatrix} 100.0 \sin(\theta_1 + \theta_2 + \theta_1 + \theta_2) \\ 125.0 \cos(\frac{\theta_1}{2.5} + \frac{\theta_2}{2.0} + \frac{\theta_1}{1.5} + \theta_2) \end{bmatrix} \tag{22}$$

이고, 이는 보통

$$m(\theta)\theta + F(\theta, \dot{\theta}) + \Delta(\theta, \dot{\theta}) = T \tag{23}$$

의 형식으로 나타낸다. 여기서 $m_1=0.5$, $m_2=6.25$, $l_1=1$, $l_2=0.8$, $J_1=5.0$, $J_2=5.0$, $G=0.8$ 이고 $C_i = \cos\theta_i$, $S_i = \sin\theta_i$, $i=1, 2$ 이며 $C_{ij} = \cos(\theta_i + \theta_j)$, $S_{ij} = \sin(\theta_i + \theta_j)$, $i, j=1, 2$ 임을 의미한다. 그리고 $F(\theta, \dot{\theta})$ 는 (21)식 좌변의 2, 3번째항 즉, 코리올리항과 중력항을 포함한다. 원하는 추종 궤적은 직교 좌표계상의 원 궤적

$$x = 0.7 + 0.5 \sin s(t) \tag{24}$$

$$y = 0.7 + 0.5 \cos s(t), \quad -\pi \leq s(t) \leq \pi, \quad 0 \leq t \leq P$$

이고, 이는 역기구학과 역자코비안 과정을 통하여 θ_i 와 θ'_i 로 변환될 수 있다. 여기서

$$s(t) = -\pi \cos \frac{\pi}{P} t \text{이고, } P=5 \text{초로 하였다. 그림 2는}$$

신경망의 주기 학습 최대 모델 오차 곡선을 보여준다. 항상은 아니지만 몇번의 주기 학습 과정을 거치면 그후로는 대체로 주기 학습 최대 모델 오차 곡선이 단조적으로 감소한다고 볼 수 있다. 그림 3은 학습 초기 주기 ($k=10$)에서의 신

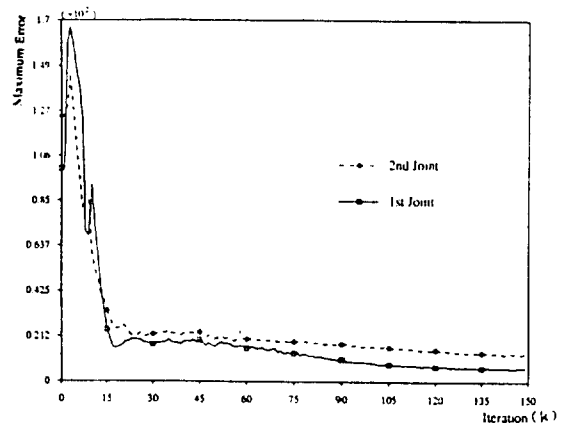
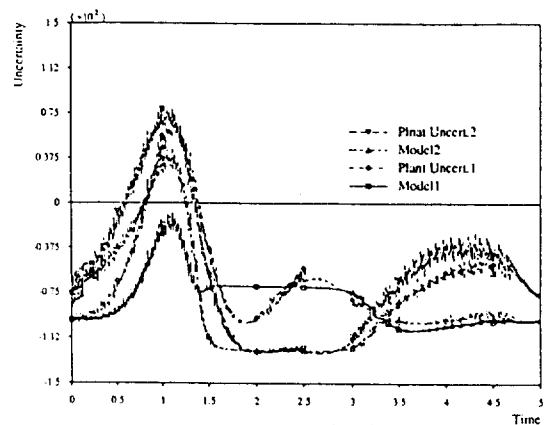
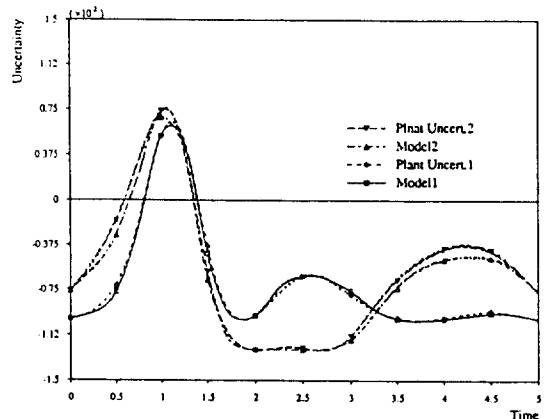


그림 2 신경망 주기 학습 알고리즘의 최대 모델 오차

Fig. 2 Maximum modeling error at each period in iterative learning algorithm of the neural network



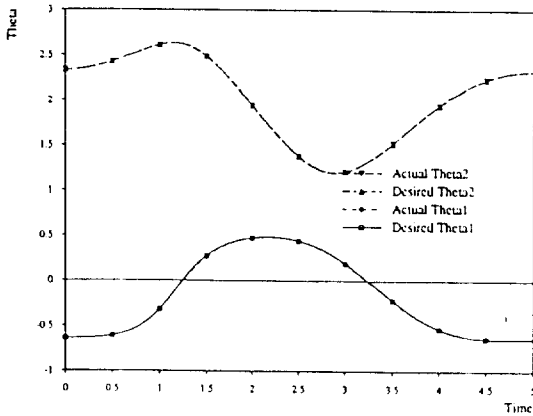
(a) $k=10$ 일 때



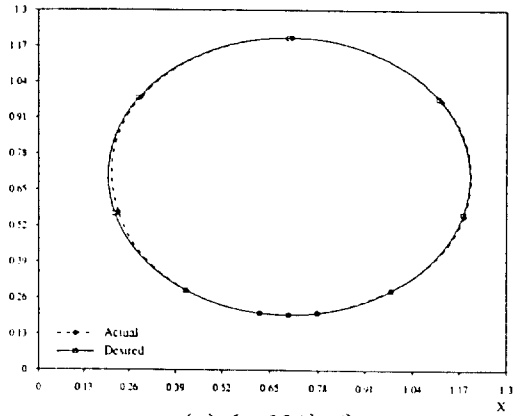
(b) $k=150$ 일 때

그림 3 시스템 불확실성에 대한 신경망 모델

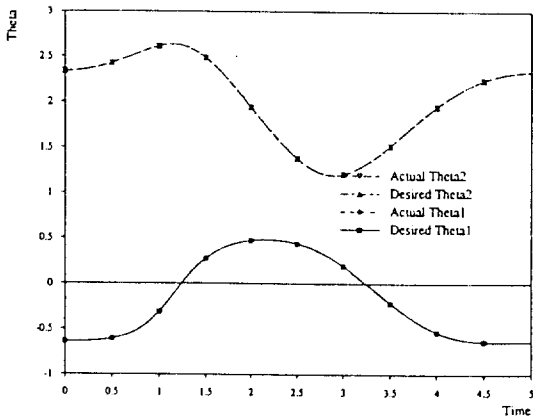
Fig. 3 The neural network model on system uncertainty



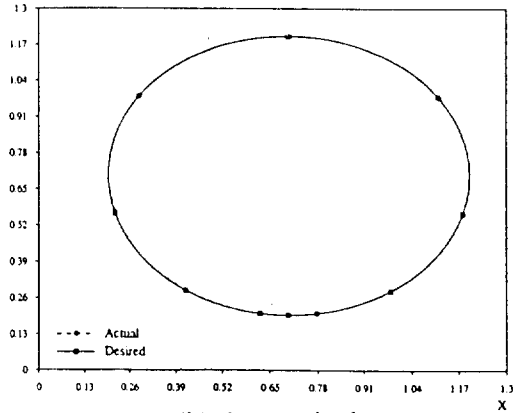
(a) $k=10$ 일 때



(a) $k=10$ 일 때



(b) $k=150$ 일 때



(b) $k=150$ 일 때

그림 4 각 축의 추종 결과

Fig. 4 Tracking of each joint

그림 5 직교 좌표 공간에서의 추종 결과

Fig. 5 Tracking in the cartesian space

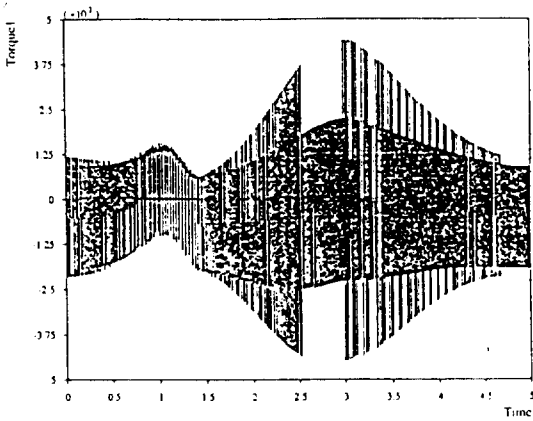
경망의 모델 곡선 (a)와 마지막 주기에서의 신경망 모델 곡선 (b)를 보여 준다. 여기서 식 (23) 형태의 불확실성 함수가 고주파 성분을 포함하고 있는 것은 신경망의 불확실성에 대한 학습을 위해서 식 (22)에서 보듯이 입력 신호에서 공칭 동력학을 빼 신호를 불확실성의 기준 신호로 사용하기 때문이다. 즉

$$\bar{d} - (\theta, \dot{\theta}) = T - M(\theta)\ddot{\theta} - F(\theta, \dot{\theta}) \quad (25)$$

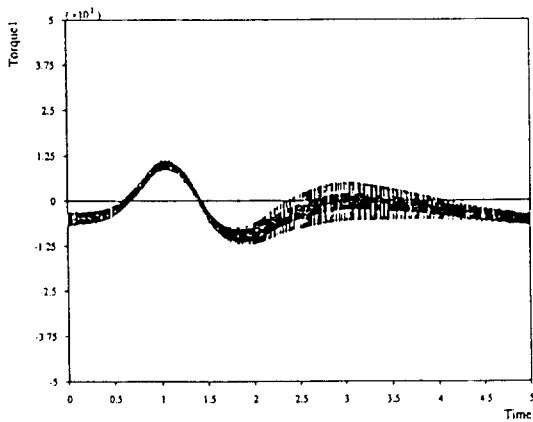
이다. 제어 입력 T 가 스위칭에 의한 고주파 성분을 포함하고 있기 때문에 기준 신호 $\bar{d}(\theta, \dot{\theta})$ 도 역시 고주파 성분을 포함하게 된다. 그림 4는 학습의 초기 주기($k=10$)와 마지막 주기($k=150$)에서 각 축의 추종 결과를 보여 준다. $k=10$ 번째 주기는 그림 2에서 보듯이 신경망의 주기 학습 최대 모델 오차가 증가하는 주기이므로

앞에서의 분석 처럼 추종 오차가 약간은 발생하게 된다. 그림 5는 직교 좌표 공간상에서의 추종 결과를 보여 준다. 앞서 언급했듯이 신경망 학습의 초기 주기($k=10$)에서 약간의 추종 오차가 있음을 볼 수 있다. 그리고 초기 주기($k=10$)와 마지막 주기($k=150$)에서의 제어 입력의 크기 비교를 그림 6에 나타내었다. 학습이 진행되어감에 따라 제어 입력의 크기가 크게 감소하고 따라서 제어 효율이 증가했음을 볼 수 있다.

본 모의 실험의 결과에서는 신경망 pattern 학습의 최대 모델 오차가 크게 증가한 주기가 없어서 (20)식에 대한 (b)와 같은 추종 오차의 분석이 어려웠다. 그래서 임의로 모델 오차의 최대값을 증가시켜 놓고 추종 오차를 실험한 결과를 다음에 제시한다. 즉 그림 2에서 보듯이 $k=10$ 번째 주기에서 신경망의 최대 오차는 원래는 대략 95정도 인데, 이 때의 신경망 $W_{k=10}$ 과 $\bar{d}_{k=9}$



(a) $k=10$ 일 때



(b) $k=150$ 일 때

그림 6 제어 입력(T_1)의 크기

Fig. 6 Magnitude of a control input(T_1)

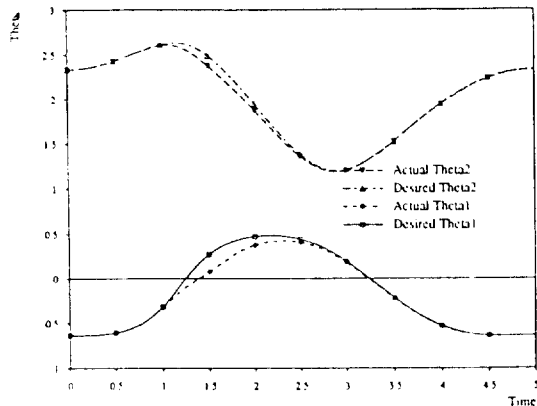


그림 7 최대 모델 오차가 이전 주기에 비해서 증가했을 때 각 축의 추종 결과

Fig. 7 Tracking of each joint when the maximum modeling error increases compared to the prior period

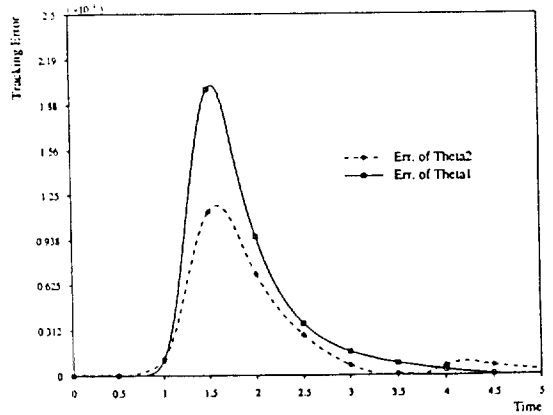


그림 8 최대 모델 오차가 이전 주기에 비해서 증가했을 때의 추종 오차의 크기

Fig. 8 Magnitude of tracking error of each joint when the maximum modeling error increases compared to the prior period

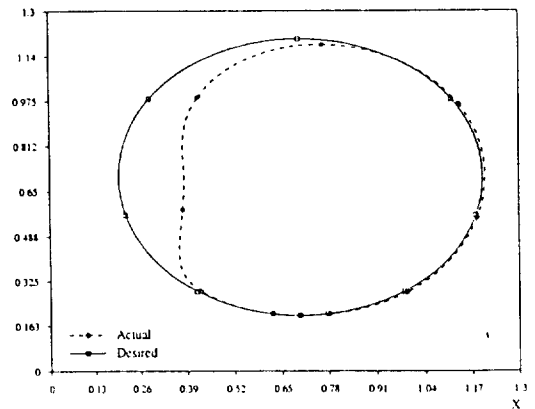


그림 9 최대 모델 오차가 이전 주기에 비해서 증가했을 때의 직교 좌표 공간에서의 추종 결과

Fig. 9 Tracking in the cartesian space when the maximum modeling error increases compared to the prior period

를 임의로 $\hat{J}_{k=9}=10$ 으로 정한 것을 가지고 식 (15), (16)으로 제어 결과를 실험한 것이다. 그림 7은 이 경우의 각 축의 추종 곡선을 보여 주며 그림 8은 추종 오차의 크기 곡선을 보여 준다. 그리고 그림 9는 직교 좌표상의 추종 곡선이다. 그림 8과 최대 모델 오차값이 감소한 경우, 즉 대표적으로 $k=150$ 번째의 추종 오차의 크기

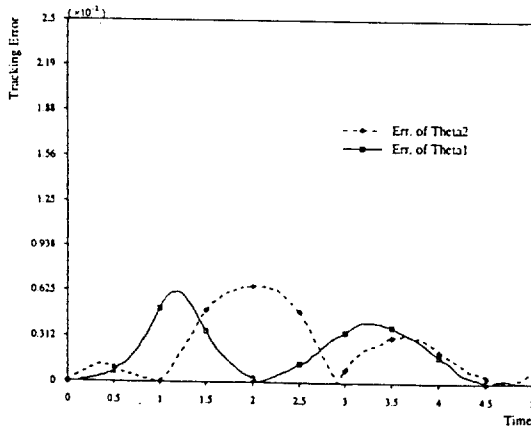


그림 10 최대 모델 오차가 이전 주기에 비해서 감소했을 때($k=150$)의 각 축의 추종 오차의 크기

Fig. 10 Magnitude of tracking error of each joint when the maximum modeling error decreases compared to the prior period

곡선 그림 10을 비교해 보면 최대 모델 오차가 증가할 때 식 (20)에서의 분석 처럼 어느 정도의 추종 오차가 발생할 수 있음을 볼 수 있다.

4. 결 론

본 논문에서는 시스템의 불확실성을 모델링하여 제어 효율을 높이기 위한 신경망의 응용과 그의 어쩔 수 없는 모델 오차를 극복하기 위한 적응적 가변 구조 제어 기법의 한가지 조합 방식을 제안하였다. 온라인 방식의 모델 학습 과정에서 매 순간 제어 입력 발생에 필요한 모델 오차를 미리 알 수는 없고, 더군다나 pattern 학습의 수렴성을 보장할 수 없기 때문에 그러한 난점을 해결하기 위하여 반복 학습 제어 방식을 도입하고 컴퓨터 시뮬레이션을 통해 그 효용성을 입증해 보였다. 결과적으로 불확실성에 대한 모델 보상이 없을 때의 제어 효율보다 신경망 모델 보상이 있을 때의 제어 효율이 훨씬 높음을 보였고, 가변 구조 제어 방식의 큰 문제점이던 chattering 현상에 대한 한가지 해결 방법이 될 수 있음을 보일 수 있었다. 본 제어 알고리즘은 다음과 같은 면에서 통상의 신경망 반복 학습 제어와 다르다. 즉 시스템의 역동력을 학

습하므로써 시스템을 보상 제어하려는 신경망 반복 학습 제어는 학습이 충분히 이루어지기 전까지의 반복 수행 과정은 단지 학습 자체에만 매달려 있을 뿐이지, 원하는 제어 작업을 수행하지는 못한다는 것이다. 그러나 본 논문의 알고리즘은 학습의 전 과정에 걸쳐서 항상 원하는 제어를 추종하면서 의사 온라인 방식으로 학습하게 되므로 학습이 진행될 수록 제어 입력이 달라질 뿐이고 필요한 작업은 항상 수행하고 있다는 것이다. 학습을 통하여 최대 모델 오차가 오히려 증가한 경우도 추종 오차의 크기가 그다지 크지 않음을 볼 수 있었다.

· 참 고 문 헌

- [1] K. Hornik, M. Stinchcombe, H. and White, "Multilayer feedforward networks are universal approximators", Neural Networks, vol. 2, 1989.
- [2] D. E. Rumelhart and J. L. McClelland, Parallel Distributed Processing, Explorations in the Microstructure of Cognition, Cambridge, MA:MIT press, 1986
- [3] John Y. Hung, Weibing Gao, and James C. Hung, "Variable Structure Control: A Survey", IEEE Tr. on Industrial Electronics, vol. 40, no. 1, pp. 2-22, 1993
- [4] Y. Itkis, Control Systems of Variable Structure, Newyork:Wiley, 1976
- [5] V. A. Utkin, Sliding Modes and Their Application in Variable Structure System, Moscow:Mir, 1978
- [6] Jean-Jacques E. Slotine and Weiping Li, Applied Nonlinear Control, Prentice-Hall International, 1991
- [7] Craig, J. J, Introduction to Robotics-Mechanics and Control : 2ed., Addison-Wesley, 1989
- [8] Martin J. Corless and George Leitmann, "Continuous State Feedback Guaranteeing Uniform Ultimate Boundedness for Uncertain Dynamic Systems", IEEE Tr. on Automatic Control, vol. 26, no. 5, 1981
- [9] Hassan K. Khalil, Nonlinear Systems, Macmillan, 1992
- [10] Shaul Gutman, "Uncertain Dynamical Systems-A Lyapunov Min-Max Ap-

proach", IEEE Tr. on Automatic Control, vol. 24, no. 3, 1979

[11] Si-Zhao Qin, Hong-Te Su, and Thomas J. McAvoy, "Comparison of Four Neural

Net Learning Methods for Dynamic System Identification", IEEE Tr. on Neural Networks, vol. 13, no. 1, 1992

저자소개



이수영(李壽榮)

1964년 10월 2일생. 1988년 연세대학교 전자 공학과 졸업(학사). 1990년 한국 과학 기술원(KAIST), 전기 및 전자 공학과 졸업(석사). 1990년~현재 한국 과학 기술원(KAIST), 전기 및 전자 공학과 박사 과정 재학중

정명진(鄭明振)

전기학회 논문지 제43권 1호 참조