

신경망을 이용한 입력제한 플랜트의 제어

°김동희*, 이시일*, 김성식*, 유동완*, 서보혁*
*경북대학교 대학원 전기공학과

Control Of A Bounded-Input Plant Using Neural Network

°Dong-Hee Kim*, Si-Il Lee*, Sung-Sik Kim*, Dong-Wan Ryo*, Bo-Hyeok Seo*
Dept. of Electrical Eng. Kyungpook Nat. Univ.*

Abstract - Generally, neural networks can be used efficiently for the identification and control of nonlinear dynamical system, then it always needs to learn in order that the output values is closed to desired values. But, if plant input to control is limited to certain bounded values, former learning rules has the another problem. This paper demonstrates algorithm to control the bounded-input plant using neural network controller.

1. 서론

일반적인 제어 기법은 시스템을 수학적으로 모형화하고 특정 동작점에서 선형화한 선형 모형으로부터 제어를 설계한다. 그러나 플랜트가 고차의 미분방정식으로 표현되는데 따른 복잡성과 플랜트의 비선형성, 그리고 플랜트 모형화의 부정확성으로 인해 만족할 만한 제어 성능을 나타내기에는 어려움이 있다. 이러한 어려움을 극복하고자 신경망을 이용한 제어기[1]가 도입되었다. 이러한 신경망 제어기는 플랜트의 출력이 원하는 값에 가까워지도록 학습하는 과정이 필요하고 일반적으로 플랜트 출력오차의 제곱합을 목적함수로 하여 기울기 강하법을 이용하여 이 값을 줄이는 방향으로 학습이 이루어진다. 그러나, 그림 1과 같이 제어하고자 하는 플랜트의 입력이 어떤 범위 안의 값으로 제한될 때, 이러한 학습은 문제점을 가지게 된다. 즉, 제어기의 출력 u_c 가 플랜트 입력 범위 u_b 를 벗어날 경우 실제 플랜트 입력으로는 u_c 가 아니라 $\pm u_b$ 가 들어가게 되어 일반적인 학습 알고리즘을 적용하기 곤란하다.

본 논문에서는 신경망 제어기를 이용하여 입력제한 플랜트를 제어하기 위하여 목적함수를 설정하는 방법을 제시하고 사례연구를 통하여 그 타당성을 검증하고자 한다.

2. 문제 설정

본 논문에서는 그림 1과 같이 입력 제한이 있는 플랜트를 신경망 제어기를 이용하여 제어하고자 한다.

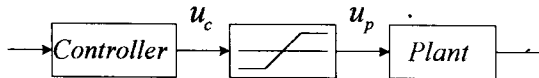


그림 1. 입력제한이 있는 플랜트
Fig. 1. Bounded-input plant

$$u_p = \begin{cases} -u_b, & u_c \leq -u_b \\ u_c, & |u_c| < u_b \\ u_b, & u_c \geq u_b \end{cases} \quad (1)$$

여기서 u_c 는 제어기의 출력, u_p 는 플랜트의 입력, 그리고 u_b 는 플랜트의 입력 제한의 절대값이다.

3. 시스템의 구성과 학습

제어 시스템은 그림 2와 같이 기준 모델이 없는 형태이며 직교 신경망(Orthogonal Neural Network)을 이용한 식별기와 제어기로 구성된다. 그리고 식별기와 제어기는 모두 오프-라인으로 학습한다.

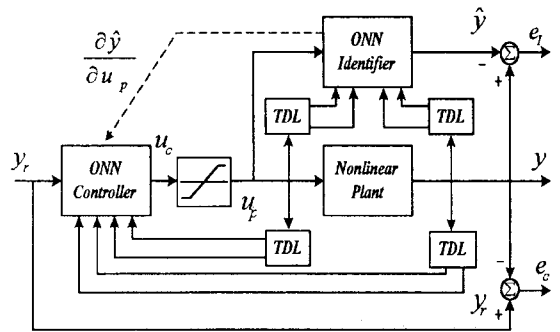


그림 2. 제어 시스템의 구조
Fig. 2. Block diagram of control system

3.1 직교 신경망

그림 3은 Yang과 Tseng이 제안한 m 개의 입력층, N 개의 중간층, 그리고 1개의 출력층으로 구성된 직교함수를 이용한 직교 신경망[2]을 보인 것이다.

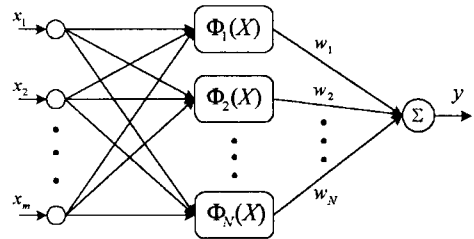


그림 3. 직교 신경망
Fig. 3. Orthogonal neural network

이는 비선형 시스템의 수학적 모형을 모르는 경우더라도 신경망의 학습 알고리즘을 이용하여 시스템의 과거 입출력 자료로부터 직교함수 계수의 계수를 구하여 시스템을 모형화 할 수 있다. 이러한 형태의 신경망을 수식으로 나타내면 식 (2)와 같다

$$y = \sum_{i=1}^N w_i \Phi_i(X) = W^T \Psi(X) \quad (2)$$

여기서,

$$\Phi(X) = [\Phi_1(X) \ \Phi_2(X) \ \cdots \ \Phi_M(X)]^T$$

$$X = [x_1 \ x_2 \ \cdots \ x_m]^T$$

$$W = [w_1 \ w_2 \ \cdots \ w_M]^T$$

이다.

각 입력 변수에 대해 n 차의 직교 함수로 신경망을 구성한 경우 중간층의 수 N 은 식 (3)과 같다.

$$N = n^m \quad (3)$$

m 은 입력 변수의 수, n 은 각 변수에 대한 단일변수 직교 함수의 차수를 각각 나타낸다.

본 논문에서는 식 (4)과 같은 단일 변수 레전드레 다항 함수(Legendre polynomial function)[2]의 곱으로 된 다변수 함수를 기본적인 직교 함수 $\Phi_M(X)$ 로 사용하며, 레전드레 다항함수의 도함수는 식 (5)과 같다.

$$L_1(x) = 1, \quad L_2(x) = x$$

$$L_{i+1}(x) = a_i x L_i(x) - b_i L_{i-1}(x) \quad (4)$$

$$a_i = 2 - 1/i, \quad b_i = 1 - 1/i$$

$$l_1(x) = 0, \quad l_2(x) = 1$$

$$l_{i+1}(x) = a_i(L_i(x) + x l_i(x)) - b_i l_{i-1}(x) \quad (5)$$

$$-1 \leq x \leq 1$$

여기서, L_i 는 레전드레 다항함수, l_i 는 그 도함수를 나타낸다.

직교 신경망은 요구되는 정확성을 기준으로 직교 함수의 급수를 증가시키므로 직교 신경망의 구조를 결정할 수 있다. 또한, 중간층과 출력층 사이에만 가중치가 존재하기 때문에 목적 함수를 적절히 설정하면, 학습 알고리즘이 오차 역전파법 학습 알고리즘에 비해 간단하고 수렴 속도도 빠르다. 그리고 가중치의 초기치는 모두 0으로 설정하므로 다층 신경망에서처럼 난수를 발생시켜야 하는 부담감이 없다.[2]

3.2 식별기의 학습

플랜트의 동특성을 모형화하기 위한 목적 함수는 식 (6)과 같으며, 식별기의 학습은 목적 함수를 최소화하도록 하는 것이다.

$$e_f(k+1) = y(k+1) - \hat{y}(k+1)$$

$$J_f = \frac{1}{2} \sum_{k=0}^T e_f(k+1)^2 \quad (6)$$

여기서 $y(k)$ 는 플랜트의 출력, $\hat{y}(k)$ 은 식별기의 출력, $e_f(k)$ 는 플랜트와 식별기의 출력 오차를 각각 나타낸다. 또, 식별기를 최적화하기 위한 가중치 $W_f = [w_{f1} \ w_{f2} \ \cdots \ w_{fN_f}]^T$ 의 증분량 ΔW_f 는 식 (7),

(8)과 같이 구한다.

$$\frac{\partial J_f}{\partial W_f} = \frac{\partial J_f}{\partial e_f(k+1)} \frac{\partial e_f(k+1)}{\partial W_f}$$

$$= -e_f(k+1) \Psi_f(X_T(k)) \quad (7)$$

$$\Delta W_f = \eta_{w_f} \left(-\frac{\partial J_f}{\partial W_f} \right)$$

$$= \eta_{w_f} e_f(k+1) \Psi_f(X_T(k)) \quad (8)$$

3.3 제어기의 학습

그림 4는 원하는 출력 u_p^* 와 학습중인 제어기의 출력 u_c 의 대표적인 3가지 경우 u_c^i , $i=1,2,3$ 을 나타낸 것이다. 플랜트의 입력이 u_p^* 와 같이 되려면 제어기의 출력 u_c 는 식(9)와 같아야 한다.

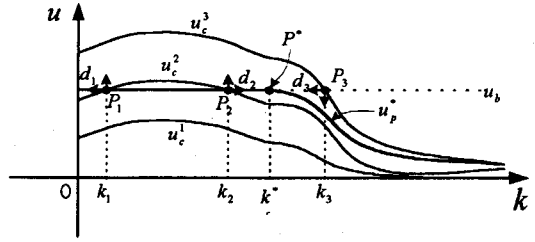


그림 4. 학습중인 제어기의 출력 형태

Fig. 4. Output of controller on learning

$$u_c \geq u_p^*, \quad 0 \leq k \leq k^* \quad (9)$$

$$u_c = u_p^*, \quad k^* < k \leq T$$

따라서 제어기의 학습은 $|u_c| < u_p$ 인 구간에서는 출력 오차 제곱을 줄이는 방향으로 이동시키도록 학습하여야 한다. 즉, u_c^1 의 경우는 모든 값에 대해서 출력오차 제곱을 줄이도록 학습을 해 주어야 한다. 그리고 u_c^2 의 경우는, $0 \leq k < k_1$ 과 $k_2 < k \leq T$ 에서는 출력오차 제곱을 줄이도록 학습해 주어야 하고, $k_1 \leq k \leq k_2$ 에서는 점 P_1 을 d_1 의 방향으로, 점 P_2 를 d_2 의 방향으로 이동시켜야 하므로 점 P_1 과 P_2 에서의 u_c 값을 증가시켜야 한다. 또한 u_c^3 의 경우는 $k_3 < k \leq T$ 에서는 출력오차 제곱을 줄이도록 학습해 주어야 하고, $0 \leq k < k_3$ 에서는 점 P_3 을 d_3 방향으로 이동시켜야 하므로 점 P_3 에서의 u_c 의 값을 감소시켜야 한다. 각 경우에서 알 수 있듯이 점 P 를 원하는 방향으로 이동시키는 것 역시 그 점에서의 출력오차 제곱을 줄이도록 하는 방향이다. 따라서 목적함수를 식 (10)과 같이 설정하여 $|u_c| \leq u_p$ 일 때의 출력오차 제곱을 줄이는 방향으로 제어기를 학습하면 원하는 플랜트 제어 입력 $u_p \approx u_p^*$ 를 구할 수 있다.

$$e_c(k+1) = y_r(k+1) - y(k+1)$$

$$J_c = \frac{1}{2} \sum_{|u_c(k)| \leq u_p} e_c(k+1)^2 \quad (10)$$

식(9)에서 $y_r(k+1)$ 은 기준 입력이고 $e_c(k+1)$ 는 기준 입력과 플랜트 출력과의 오차이다. 제어기 학습에 필요한 가중치 $W_c = [w_{c1} \ w_{c2} \ \cdots \ w_{cN_c}]^T$ 의 증분량 ΔW_c 는 식(10)으로부터 식 (11)과 같이 구할 수 있다.

$$\frac{\partial J_c}{\partial W_c} = \frac{\partial J_c}{\partial e_c(k+1)} \frac{\partial e_c(k+1)}{\partial W_c}$$

$$= -e_f(k+1) \frac{\partial y(k+1)}{\partial u_p(k)} \Psi_c(X_T(k)) \quad (11)$$

$$\Delta W_c = \eta_{w_c} \left(-\frac{\partial J_c}{\partial W_c} \right)$$

$$= \eta_{w_c} e_c(k+1) \frac{\partial y(k+1)}{\partial u_p(k)} \Psi_c(X_T(k)) \quad (12)$$

여기서,

$$\Psi_c(X_T(k)) = [\Phi_{c1}(X_T(k)) \ \cdots \ \Phi_{cN_c}(X_T(k))]^T$$

이 때, 학습과정에서 식(12)에서 보이는 것처럼 플랜트의 입력에 대한 출력의 변화를 $\frac{\partial y(k+1)}{\partial u_p(k)}$ 을 알아야 한다. 그러나 가정에서 플랜트를 알지못하므로 식별기로부터 그 추정치 $\frac{\partial \hat{y}(k+1)}{\partial u_p(k)}$ 을 받아서 식(13)과 같이 구

한 근사값 \hat{W}_c 를 이용한다.

$$\Delta \hat{W}_c = \eta_{w_c} e_c(k+1) \frac{\partial \hat{y}(k+1)}{\partial u_p(k)} \Psi_c(X_T(k)) \quad (13)$$

여기서, η_{w_c} 는 제어기 가중치의 학습률로서 양의 작은

값이고, 학습에 필요한 $\frac{\partial \hat{y}(k+1)}{\partial u_p(k)}$ 는 식별기로부터,

$x_1(k) = u_p(k)$ 라 하면 식 (14)과 같이 구할 수 있다.

$$\frac{\partial \hat{y}(k+1)}{\partial x_1(k)} = \frac{\partial}{\partial x_1(k)} (W_f^T \Psi_f(X_T(k))) \quad (14)$$

4. 사례연구

제한한 제어 알고리즘을 비선형 시스템 제어의 대표적인 예 중 하나인 식 (15)로 표현되는 도립 진자(Inverted Pendulum) 균형문제[4]에 적용하여 제어 성능을 조사하였다.

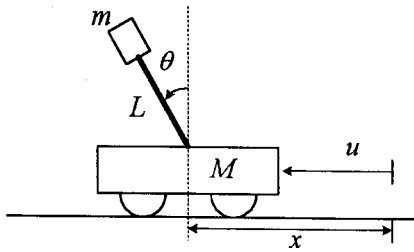


그림 5 도립진자 시스템

Fig. 5. Inverted Pendulum System

$$\begin{cases} (M+m)\ddot{x} + mL \cos \theta \ddot{\theta} - mL \sin \theta \dot{\theta}^2 - u = 0 \\ m \cos \theta \ddot{x} + mL \ddot{\theta} - mg \sin \theta = 0 \end{cases} \quad (15)$$

그림 5에서 M 은 수레의 질량으로 50g, m 은 진자의 질량으로 30g, l 은 진자의 길이이며 2.0m, x 는 수레의 위치, θ 는 진자의 기울어진 각도, u 는 플랜트의 입력으로서 수레를 미는 힘을 각각 나타낸다.

식별기는 $y(k)$, $y(k-1)$, $y(k-2)$, $u(k)$, $u(k-1)$ 을 입력으로 하고, 각 입력에 대하여 직교 함수는 3차까지로 구성하였다. 플랜트가 동작하는 각도 θ 의 범위는 $-1rad \leq \theta \leq 1rad$, 그리고 입력 u 의 범위는 $-1.5N \leq u \leq 1.5N$ 로 하였다. 식별기 학습에 필요한 플랜트의 출력은 룬게쿠타(Runge-Kutta) 방법을 사용하여 구하였고 샘플링 시간은 0.1초로 하였다.

제어기는 $y(k)$, $y(k-1)$, $y(k-2)$, $u(k-1)$, $r(k)$ 를 입력으로 하여 각 입력에 대하여 직교 함수를 4차까지로 구성하고 시뮬레이션을 행하였다.

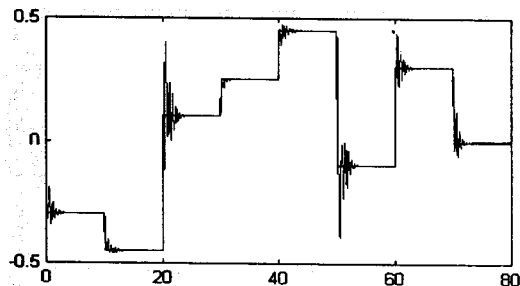


그림 6. 기준 입력과 플랜트 출력

Fig. 6. Reference input and output of plant

그림 6은 제어기 학습을 마친 후 학습을 끊고 플랜트를 제어했을 때 플랜트의 출력과 기준 입력 r 을 나타낸 것이다. 그림 7과 그림 8은 이 때의 제어기 출력 u_c 와 플랜트 입력 u_p 를 각각 나타낸 것이다.

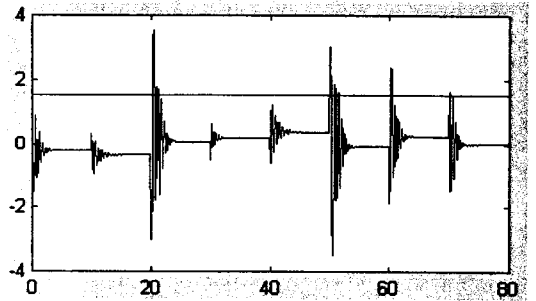


그림 7. 제어기의 출력(u_c)

Fig. 7. Output of controller

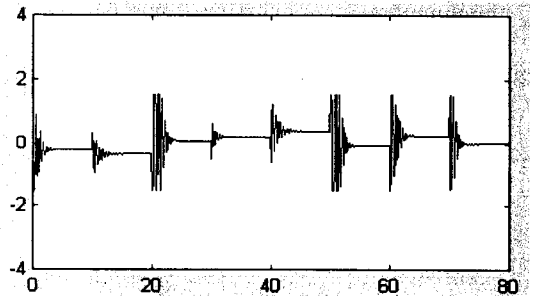


그림 8. 플랜트의 입력(u_p)

Fig. 8. Input of plant

4. 결 론

본 논문에서는 신경망 제어기를 이용하여 입력제한이 있는 플랜트를 제어하기 위하여 목적함수를 식 (10)과 같이 설정하는 학습 방법을 제시하였다. 그리고 사례연구를 통하여 제한한 학습 방법의 타당성을 보였다.

[참고 문헌]

- [1] K. S. Narendra and K. Parthasarathy, "Identification and Control of Dynamical Systems Using Neural Networks," IEEE Trans. Neural Networks, vol. 1, no. 1, pp. 4-27, 1990.
- [2] Shio-Shung Yang and Ching-Shio Tseng, "An Orthogonal Neural Network for Function Approximation", IEEE Trans. Systems, Man, and Cybernetics, vol. 26, no. 5, pp. 779-785, 1996.
- [3] 김성식, 이영석, 서보혁, "수정된 직교 신경망을 이용한 비선형 시스템의 제어", 대한전기학회 논문지, vol 47 no 6, pp.840-849, 1998, 6)
- [4] Youn-Moon Park, Myeon-Song Choi, "An Optimal Tracking Neuro-Controller for Nonlinear Dynamic Systems", IEEE Trans. Neural Networks, vol. 7, no. 5, pp. 1099-1110, 1996.