

Neural Network를 이용한 제어기 설계

김대중, 구영모, 장석호, 우광방
연세대학교 전기공학과

Design of Controller Utilizing Neural-Network

Dae-jong Kim, Young-mo Koo, Seog-ho Chang, Kwang-bang Woo
Dept. of Electrical Engineering, Yonsei University

ABSTRACT

This study is to design a method of parameter estimation for a second order linear time invariant system of self-tuning controller utilizing the neural network theory proposed by Hopfield.

The result is compared with the other methods which are commonly used in controller theories.

1. 서론

뉴럴 기법을 이용한 방법은, 기존의 컴퓨터가 정확한 해답을 오면 시간에 걸쳐서 찾아내는 대신, 비교적 정확도가 높고 빠른 시간 내에 해답을 찾는 데에 장점이 있다. Altes는 basis가 $\{\phi_i; i = 1, 2, \dots, n\}$ 이고 data vector X 가 $X = \sum V_i \phi_i$ 로 표시될 때 Hopfield가 제안한 방법을 이용해서 임의의 실수 계수 $\{V_i; i = 1, 2, \dots, n\}$ 가 최소의 오차를 갖도록 추정하는 것이 가능하도록 했다.

본 논문에서는 이러한 결과를 이용해서 뉴럴 네트워크를 제어 이론에 적용시키는데 특별히 자기 동조 제어기 부분에서 매개 변수 추정 부분을 Hopfield의 뉴럴 네트워크로 구현하고, 그 결과를 기존의 제어 이론과 비교 검토하였다.

2. 매개 변수 추정

2-1. Hopfield 이론에 근거한 매개 변수 추정

뉴럴 네트워크에서 i 번째 뉴런의 출력은 아래와 같다
 $V_i(t) = g[U_i(t)]$

여기에서 $g(x)$ 는 단조 비감소 함수로 $g(-\infty) = 0, g(0) = 1/2, g(\infty) = 1$ 의 성질을 갖는 sigmoid 함수다. 또한 이 때 i 번째 뉴런의 입력은 다음과 같다.

$$C_i(d/dt)U_i(t) = \sum_{j=1}^n T_{ij}V_j(t) - U_i(t)/R_i + I_i \quad (1)$$

여기에서
 $1/R_i = 1/\rho_i + \sum_{j=1}^n (1/R_{ij})$

이고, C_i, R_i 는 상수로 뉴럴 네트워크의 시상수를 결정하게 된다. 계산의 편의를 위해서 그 값을 1로 두면,

$$(d/dt)U_i(t) = \sum_{j=1}^n T_{ij}V_j(t) - U_i(t) + I_i \quad (2)$$

가 된다. 여기에서 T_{ij} 는 뉴럴 네트워크의 연결 강도를 나타낸다. 또, 아헷식의 조건이 만족될 때

$$T_{ij} = \begin{cases} 0, & i = j \\ T_{ji}, & i \neq j \end{cases} \quad (3)$$

Hopfield 모델은 다음 식과 같은 에너지 함수의 local minima를 갖게 된다.

$$E = -(1/2) \sum_{i=1}^n \sum_{j=1}^n T_{ij}V_i(t)V_j(t) - \sum V_i(t)I_i(t) \quad (4)$$

벡터 X 가 basis $\{\phi_i; i = 1, 2, \dots, n\}$ 의 weighted sum $\sum U_i \phi_i$ 로 표시될 때, 식(5)에서 MSE(U)의 오른쪽 두 항을 최소화하는 계수 $\{U_i; i = 1, 2, \dots, n\}$ 를 찾을 때 MSE(U)는 최소화된다.

$$\begin{aligned} \text{MSE}(U) &= (X - \sum_{i=1}^n U_i \phi_i)^T (X - \sum_{j=1}^n U_j \phi_j) \\ &= X^T X + \sum_{i=1}^n \sum_{j=1}^n \phi_i^T \phi_j U_i U_j - 2 \sum_{i=1}^n U_i (\phi_i^T X) \end{aligned} \quad (5)$$

Gradient vector를 다음과 같이 정의하고,

$$\nabla U = [(d/du_1)\text{MSE}(U), (d/du_2)\text{MSE}(U), \dots, (d/du_n)\text{MSE}(U)]^T$$

$\nabla U(k)$ 를 MSE(U)의 gradient로 정의해서 steepest descent algorithm을 사용하면,

$$U(k+1) = U(k) - \mu Vu(k) \quad (6)$$

이 되고 여기에서,

$$Vu(k) = 2[CU(k) - r] \quad (7)$$

이다. 단,

$$C_{ij} = \phi_i \phi_j \quad (N \times N \text{ matrix})$$

$$r_i = X^T \phi_i \quad (N \times 1 \text{ matrix})$$

$\phi^T \phi_i = 1, i = 1, 2, \dots, n$ 인 조건을 만족한다고 하면 식(6)과 식(7)에서

$$U(k+1) = U(k) - 2\mu [CU(k) - r] \\ = (1-2\mu)U(k) + 2\mu [(I-C)U(k) + r] \quad (8)$$

처럼 되고 I는 $N \times N$ identity matrix다.

이제 식(2)와 식(8)을 계수 비교하면

$$T_{ij} = -C_{ij}/g'(0) = -\phi^T \phi_i \phi_j / g'(0), \quad i \neq j \\ = 0, \quad i = j$$

$$I_1 = r_1 - g(0) \sum_{j=1}^n T_{1j} \\ = X^T \phi_1 + [2g'(0)]^{-1} \sum_{j=1, j \neq 1}^n \phi_1 \phi_j$$

가 된다.

이에 따라 우리는 Hopfield의 뉴럴 네트워크를 이용해서 임의의 실수를 계수로 갖는 매개 변수를 추정할 수 있게 된다.

2-2. 문제 설정

아래 식(9)와 같은 방정식으로 표시된 플랜트에 100 번의 샘플링 스텝마다 1 또는 -1을 갖는 파형을 입력 $U(k)$ 로 사용한다.

$$x(k) = a_1 x(k-1) + a_2 x(k-2) + b_0 u(k) + b_1 u(k-1) \\ = \phi^T(k) \theta \quad (9)$$

여기에서,

$$\phi^T(k) = [\phi_1(k), \phi_2(k), \phi_3(k), \phi_4(k)] \\ = [x(k-1), x(k-2), u(k), u(k-1)] \\ \theta^T = [\theta_1, \theta_2, \theta_3, \theta_4] \\ = [a_1, a_2, b_0, b_1]$$

여기에서 우리가 해결해야 할 문제는 Hopfield Neural-network를 이용해서 식(9)에 있는 매개 변수 a_1, a_2, b_0, b_1 값을 추정해 내고, 추정된 매개 변수에 의한 값과 $x(k)$ 가 최소 제곱 오차(MMSE)를 가지도록 작업한다.

2-3. 매개 변수 추정

식(9)와 같은 플랜트의 매개 변수를 추정하려면, 표준화된 basis { $\phi_i; i = 1, 2, 3, 4$ }의 하중합 $\sum_{i=1}^n U_i \phi_i$ 형태로 표시되는 벡터 X 와의 차가 0이 되도록 매개변수

추정하면 된다.

$$\varepsilon(1) = x(1) - \hat{x}(1) = x(1) - \phi^T(1)\theta \text{ 이므로}$$

$$E(t) = [\varepsilon(t-3), \varepsilon(t-2), \varepsilon(t-1), \varepsilon(t)]^T \text{라 하면}$$

$$E(t) = 0 \text{ 에서}$$

$$X(t) = a_1 \phi^1 + a_2 \phi^2 + b_0 \phi^3 + b_1 \phi^4 \quad (10)$$

여기서,

$$X(t) = [x(t-3), x(t-2), x(t-1), x(t)]^T$$

$$\phi^1 = [x(t-4), x(t-3), x(t-2), x(t-1)]^T$$

$$\phi^2 = [x(t-5), x(t-4), x(t-3), x(t-2)]^T$$

$$\phi^3 = [u(t-3), u(t-2), u(t-1), u(t)]^T$$

$$\phi^4 = [u(t-4), u(t-3), u(t-2), u(t-1)]^T$$

식(10)에서 { $\phi_i; i=1, 2, 3, 4$ }는 선형독립이므로 basis가 될 수 있다. 따라서 이들 basis를 표준화하면 다음 식(11)과 같이 된다.

$$X(t) = A_1 \phi_1 + A_2 \phi_2 + B_0 \phi_3 + B_1 \phi_4 \quad (11)$$

$$= \sum_{i=1}^4 U_i \phi_i$$

여기서,

$$U_1 = A_1 = a_1 \|\phi^1\|, U_2 = A_2 = a_2 \|\phi^2\| \quad (12)$$

$$U_3 = B_0 = b_0 \|\phi^3\|, U_4 = B_1 = b_1 \|\phi^4\|$$

$$\phi_1 = \phi^1 / \|\phi^1\|, \quad \phi_2 = \phi^2 / \|\phi^2\|$$

$$\phi_3 = \phi^3 / \|\phi^3\|, \quad \phi_4 = \phi^4 / \|\phi^4\|$$

이때, 식(11)은 식(6)과 같이 표현할 수 있다. 한편 식(2)에서 V_j 는 입력 U_j 가 매우 작은 값이라고 가정하면 테일러 급수를 이용하여 아래 식(13)과 같이 근사화 할 수 있다.

$$V_j = g(U_j) \\ = g(0) + U_j g'(0) + (U_j^2/2)g''(0) + \dots \\ \approx g(0) + g'(0)U_j \quad (13)$$

따라서 $X(t)$ 대신 $X_c(t) = X(t)/K, (K = 10000 * (X^T X)^{1/2})$ 인 $X_c(t)$ 를 사용하면 식(8)은 다음 식(14)와 같이 된다.

$$U(k+1) = (1-2\mu)U(k) \\ + 2\mu [(I-C)(V(k)-g(0))/g'(0) + r] \quad (14) \\ = (1-2\mu)U(k) \\ + 2\mu [(I-C)V(k)/g'(0) + r - (I-C)g(0)/g'(0)]$$

여기에서 식(2)와 식(14)를 계수비교하면

$$T_{ij} = \begin{cases} -C_{ij}/g'(0) = -\phi^T \phi_i \phi_j / g'(0) & i \neq j \\ 0 & i = j \end{cases} \quad (15)$$

$$I_1 = r_1 - g(0) \sum_{j=1}^n T_{1j} \\ = X^T \phi_1 + [2g'(0)]^{-1} \sum_{j=1}^n \phi_1 \phi_j$$

식(15)와 같이되며 이때 Hopfield 뉴럴 네트워크의 뉴런 입력 ($U_i; i=1, 2, 3, 4$)이 $X_c(t)$ 의 매개변수 $A_1/K, A_2/K, B_0/K, B_1/K$ 를 나타내게 된다. 따라서 식(15)를 이용하여 { $U_i; i=1, 2, 3, 4$ } 즉 $X_c(t)$ 의 매개변수를 구한결

과를 K배하여 식(11)의 매개변수 A_1, A_2, B_0, B_1 을 구하면 식(12)에 의해 식(10)에서의 매개변수 a_1, a_2, b_0, b_1 을 구할 수 있다. 매개변수 추정 블록선도는 그림 1과 같으며 이때의 뉴럴 네트워크 연결강도 T_{ij} 와 외부입력 I_i 는 식(15)와 같다.

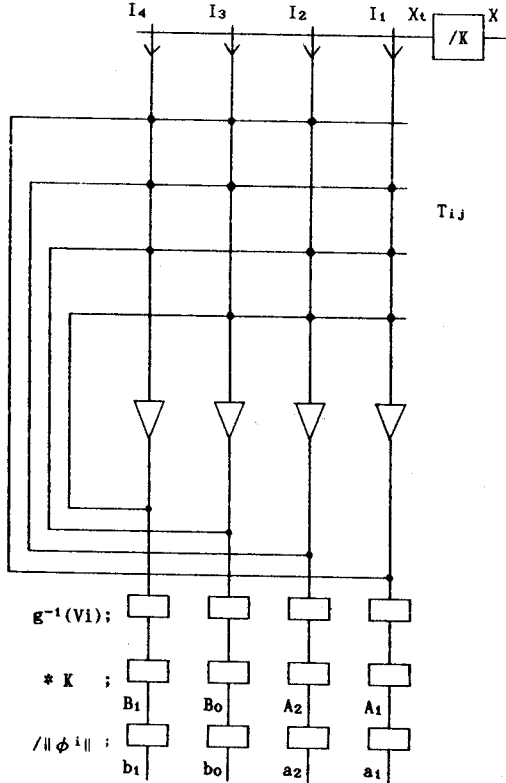


그림 1 매개변수 추정 뉴럴 네트워크

3. Simulation & Results

Hopfield의 뉴럴 네트워크를 이용해서 매개 변수를 추정하는 것을 알아보기 위해서 앞에 있는 식(9)와 같이 표현되는 단위 시간 지연을 갖는 2차 선형 시불변 플랜트 가운데 안정하고 최소 위상을 갖는 플랜트<1>과, 안정하고 비최소 위상을 갖는 플랜트<2>를 설정하고 각 플랜트에 대해서 기존에 이미 있는 E

RLS (Extended-RLS) 방법 및 제안된 방법을 써서 매개 변수를 추정했다.

시뮬레이션 대상 시스템인 플랜트<1>과 플랜트<2>는 아래와 같다.

Plant<1> :

$$x(k) = 0.857x(k-1) - 0.548x(k-2) + 0.381u(k)$$

$$+ 0.31u(k-1)$$

Plant<2> :

$$x(k) = 0.7x(k-1) - u(k) + 2u(k-1)$$

Simulation은 IBM-pc에서 처리되었으며 시뮬레이션 대상 시스템을 구성하여 100번의 샘플링 시스템마다 1 또는 -1 값을 갖는 구형파를 입력 $u(k)$ 로 하였을 때 플랜트 매개 변수의 추정 상태를 관찰하였다. 그림 2.는 각각의 플랜트에 대한 입력 $u(k)$ 이다. 그림 3.과 그림 4.는 각 플랜트에 대해서 다음의 ERLS 방법 및 뉴럴 네트워크에 대한 두가지 방법을 써서 실행한 시뮬레이션 결과를 나타낸다. 뉴럴 네트워크를 써서 실행한 첫번째 방법은 ϕ 를 4x4 matrix를 써서 실행한 것으로 ERLS와 비교해보면 ERLS 방법보다는 다소 떨어지지만 실제의 현장 작업에서는 적합하리라 여겨진다. 뉴럴 네트워크를 써서 실행한 두번째 방법은 ERLS 방법보다도 우수하다. 하지만 실지로 쓰일 때 saturation이 일어날 가능성도 있으므로 주의할 필요가 있다.

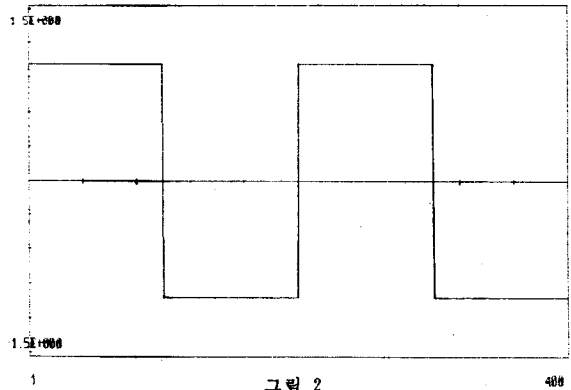


그림 2

4. 결 론

시뮬레이션의 결과를 토대로 하였을 때 Hopfield 뉴럴 네트워크를 이용해서 자기 동조 제어기를 설계하는 것의 가능성을 알 수 있었고, 매개 변수 추정 결과를 통해 볼 때 상당히 정확한 값을 알 수 있었다. 따라서 뉴럴 네트워크 기법을 자기 동조 제어기 구성에 적용될 수 있음을 제시하였다.

향후 이루어져야 할 과제는 매개 변수 추정 작업과정을 하드웨어로 구현하는 일과, 자기 동조 제어기에서 매개 변수 추정 이외의 부분을 뉴럴 네트워크로 구현하는 것이라 할 수 있다.

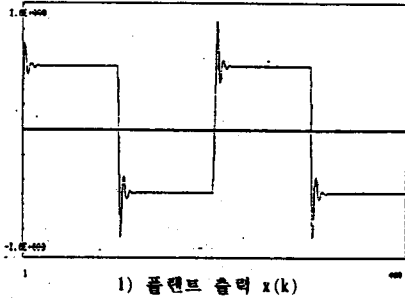
5. References

1) J.J.Hopfield, 'Neural networks and physical systems with emergent collective computational abilities' Proc. of the Nat' Aca. of Sci., 1982

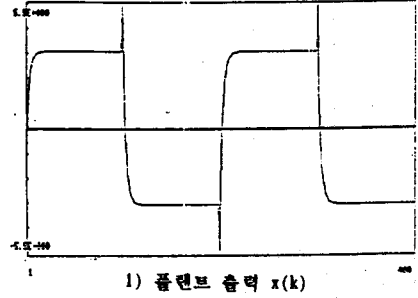
2) D.W.Tank & J.J.Hopfield, 'Simple Neural Optimisation Networks: An A/D Converters, Signal Decision Circuit and a Linear Programming Circuit', IEEE Tran. on CAS, Vol. CAS-33, No. 5, May, 1986
 3) 전기 학회지 '특집: 신경회로망', 1989.2
 4) D.W.Tank & J.J.Hopfield, 'Collective Computation in Neuron-like circuits', Scientific Amer. 1987
 5) R.A.Altes, 'Unconstrained Minimum Mean-square Error Parameter Estimation with Hopfield Networks',

IEEE ICNN pp.541-548, 1988

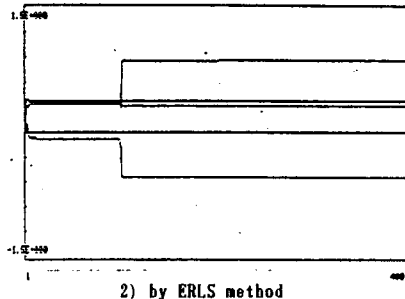
6) K.J.Åstrom & B.Wittenmark, 'Adaptive Control', Addison Wesley, 1989
 7) J.J.Hopfield, 'Neurons with graded response have collective computational properties like those of two state neurons', Proc. Nat'. Acad. Sci., USA, 1984
 8) 최창현, 이창훈, 임은빈, 우광방, 'Expert형 제어 기법에 의한 자기 동조 제어기에 관한 연구', 대한 전기 학회 논문지. 제38권, 8월. 1989



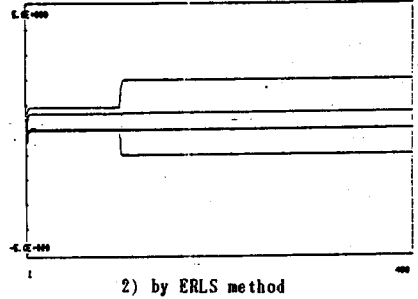
1) 플랜트 출력 $x(k)$



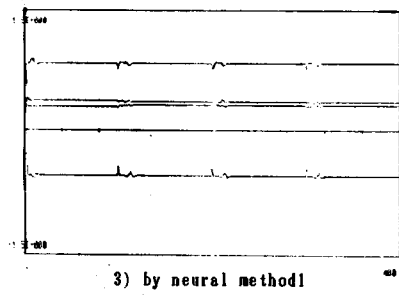
1) 플랜트 출력 $x(k)$



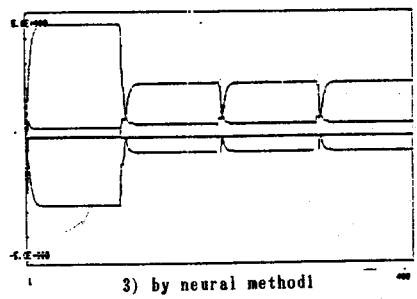
2) by ERLS method



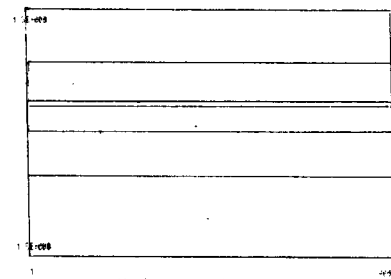
2) by ERLS method



3) by neural method1

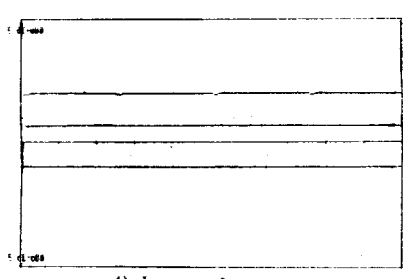


3) by neural method1



4) by neural method2

그림 3. 플랜트 1의 시뮬레이션결과



4) by neural method2

그림 4. 플랜트 2의 시뮬레이션결과