

論文96-33B-2-16

신경회로망의 자코비안을 이용한 Feedforward/Feedback 병합제어기 설계

(The Combined Feedforward/Feedback Controller Design Using Jacobians of Neural Network)

趙奎翔*, 林濟鐸*

(Gyu-Sang Cho and Chae-Tak Lim)

요 약

본 논문에서는 미지의 비선형 플랜트를 학습한 신경회로망 식별기의 자코비안 관계식을 이용한 Feedforward/Feedback 병합제어기 구조를 제안한다. 신경회로망 식별기에서 구한 자코비안 관계식은 자코비안 피드백 제어를 설계하는데 사용하며, 신경회로망 피드포워드 제어기의 역전과 학습에 이용한다. 신경회로망 제어기의 학습시에 발생하는 초기의 큰 오차값으로 인해서 학습-동시-제어(learning-while-functioning)기능을 할 수 없는 단점을 해결하기 위해서, 신경회로망 제어기와 자코비안 제어를 병합한 구조를 사용하였다. 시뮬레이션 과정에서, 제안한 제어기는 학습-동시-제어 기능을 갖는 제어기로 사용될 수 있음을 보였고, 신경회로망 제어기, 자코비안 제어기만으로 구성된 제어 시스템과의 성능을 비교하여 그 우수성과 유용성을 보였다.

Abstract

This paper proposes a combined feedforward/feedback controller which uses Jacobians of neural network. The Jacobians are calculated from the neural network that identifies the nonlinear plant, which are used for designing a Jacobian controller and for training a neural network controller. Normally, it takes much time to train the neural network controller. Combining the neural and the Jacobian controller, it can be a stable controller from the beginning of training phase of neural network, and it can be implemented as a learning-while-functioning controller. Simulated results for the proposed controller show its effectiveness and better performances..

I. 서 론

최근에, 신경회로망의 일반화 특성과, 근사화 능력을 이용하여 식별기로 사용하거나, 제어기로 사용한 구조가 많이 제시되므로써, 신경회로망을 이용한 제어시스템이 기존의 이론들과 더불어 새로운 영역으로 개척되

고 있다¹¹⁾.

Psaltis¹²⁾ 등은 일반 학습 구조와 특별 학습 구조를 제안하였는데, 일반 학습 구조는 오프라인 방식으로 플랜트의 역다이나믹스를 학습하는 것이고, 특별 학습 구조는 플랜트출력과 기준 신호와의 오차를, 플랜트를 경유하여 역전파시키는 방식이다. 이 방식에서는 Jacobian을 구하기 어려운 단점이 있다. 이런 단점을 극복하기 위해서 여러가지 방법들이 제안되었다. Sarens¹³⁾ 등은 플랜트의 근사화한 자코비안을 이용하여 역전파하는 방식을 사용하였고, Cui¹⁴⁾ 등은 플랜트 출력의

* 正會員, 漢陽大學校 電子工學科

(Dept. of Electronic Eng., Hanyang Univ.)

接受日字: 1995年5月31日, 수정완료일: 1996年1月18日

방향정보를 이용하여 제어기를 학습하는 방식을 사용하였다.

Nguyen^[5] 등은 플랜트를 학습한 신경회로망 예물레이터를 경유하여 출력에서 발생한 오차를 역전파시켜 신경회로망 제어기를 학습하는 구조를 제안하였다. 이 방법에서는 신경회로망 제어기를 학습하는데 오랜 시간이 소요되고, 학습 초기에는 정확한 제어를 할 수 없는 단점을 갖고 있다.

Kawato^[6] 등은 플랜트의 역다이나믹스를 신경회로망이 학습하는 피드백 오차 학습 구조를 사용하였다. 이 제어 구조는 역다이나믹스 모델의 목표 출력이 필요치 않고, 학습과 제어가 온라인으로 수행되며 플랜트를 경유한 오차 역전파를 하지 않는 장점이 있다.

본 논문에서는 학습을 하면서 제어를 할 수 있는 기능을 갖는 Feedforward/Feedback 병합 제어기 구조를 제안한다. 본 논문의 II장에서는 신경회로망 제어기의 일반적인 형태와 문제점에 대한 제기를 하고, III장에서는 신경회로망 제어기의 학습알고리즘을 소개하며, IV장에서는 신경회로망 제어기의 역전파 학습에 사용되고, 자코비안 제어기를 설계하기 위해 필요한 자코비안 관계식을 신경회로망 식별기로부터 유도한다. V장에서는 자코비안 제어기 설계방법에 대하여 검토하고^[7], VI장에서는 신경회로망과 자코비안 제어기로 구성된 Feedforward/Feedback 병합제어기의 온라인학습과 오프라인 학습 구조에 대해서 기술한다. VII장의 시뮬레이션에서는 신경회로망 제어기, 자코비안 제어기, 병합 제어기에 대한 각각의 제어 성능에 대한 비교 검토를 하며, VIII장에서 결론을 맺는다.

II. 문제설정

비선형 이산시간 동적 시스템을 다음과 같이 나타낸다.

$$y(k+1) = f(y(k), \dots, y(k-n), u(k), \dots, u(k-m)) \quad (n \geq m) \quad (1)$$

여기서 $y(k)$, $u(k)$ 는 각각 시스템 출력과 입력을 나타낸다. n 과 m 은 각각 출력과 입력에서의 최대 시간지연을 의미한다. 이 시스템에서는 $u(k)$ 를 인가하여 시스템의 출력 $y(k+1)$ 와 원하는 출력 $y_d(k+1)$ 의 오차 E_c 가 0이 되도록 하여야 한다.

$$E_c = \lim_{k \rightarrow \infty} (y_d(k+1) - y(k+1)) \cong 0 \quad (2)$$

$u(k)$ 를 생성하는 제어기를 다층신경회로망을 사용하는

경우에 이 값은 신경회로망의 출력이 된다.

$$u(k) = NN_c(y(k), \dots, y(k-n), u(k), \dots, u(k-m)) \quad (3)$$

다층 신경회로망은 원하는 출력을 얻기 위해서 목표신호와 신경회로망의 출력의 오차를 줄이는 방향으로 경사하강법 알고리즘을 이용하여 학습한다. 제어기로 사용되는 신경회로망은 다음식의 오차값이 0으로 근접하도록 학습을 해야한다.

$$E_u = u_d(k) - u(k) \quad (4)$$

그러나, 일반적으로 신경회로망 제어기는 플랜트와 직렬로 연결되어 있기 때문에, 식(2)의 조건을 만족하는 제어 입력 $u_d(k)$ 를 확정적으로 정하기 어렵다. 신경회로망을 이용한 제어 시스템에서는 $u_d(k)$ 를 결정하고 학습하는 방법에 따라서 제어기의 종류를 다음과 같이 구분할 수 있다^[1,8,9].

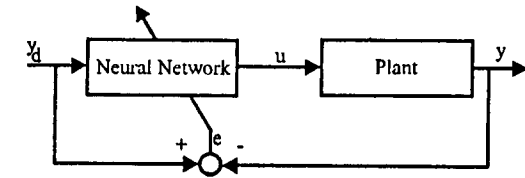
1) 직접 역제어기는 플랜트가 역변환 가능한 경우에 신경회로망으로 이 역함수를 학습하여 제어기로 사용하는 구조이다. 신경회로망의 입력을 플랜트의 출력으로 정하고, 플랜트의 입력을 신경회로망의 목표신호로 설정하여 학습한다. 이 방식은 가장 간단하게 제어기를 구성할 수 있어 효율적이지만, 플랜트가 역변환이 되지 않는 경우에 적용하기 힘들고, 플랜트의 변동에 따른 제어를 할 수 없다는 단점이 있다.

2) 적응형 제어기는 오랜시간에 걸쳐 플랜트의 변동이 있거나, 동작중의 특성변화에 따른 영향을 최소화할 수 있는 구조이다. 이 제어기는 직접(direct)방식과 간접(indirect)방식으로 구분할 수 있는데, 비선형 시스템을 제어하기 적합한 구조는 간접 방식이다. 간접 방식에서는 목표출력과 시스템 출력간의 오차가 최소가 되도록 플랜트의 입출력 특성을 학습한 식별모델을 경유하여 제어기를 학습한다. 이 방식에서는 학습초기의 제어오차가 크기 때문에 초기 제어 상태가 불안정하다.

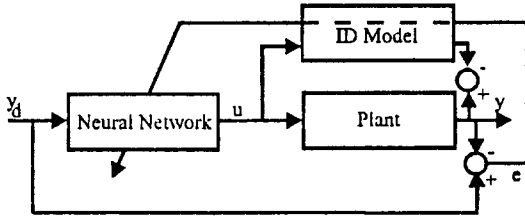
3) 병합형 제어기 구조는 Kawato^[6] 등이 제안한 구조로써, 신경회로망 제어기의 학습초기에 안정성을 유지하기 위해서 피드백 제어기법을 사용하며, 학습을 하고 난 신경회로망이 독자적인 제어기로 구성된다.

앞에서 언급한 세가지 제어기들은 각기 장단점을 갖고 있다. 시스템의 변동에 대처하기 위해서는 적응형 구조를 채택하고, 초기의 안정한 제어를 보장하기 위해서는 피드백 제어기를 사용하는 것이 안정한 제어를

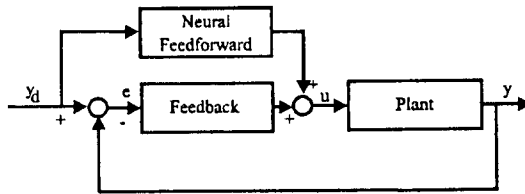
위한 수단이다. 그러므로 본 논문에서는 이 두가지 조건을 만족하는 새로운 제어기 구조를 제안한다.



(1)



(2)



(3)

그림 1. 신경회로망 제어기 학습 방식
 (1) 직접 역 제어기 (2) 적응형 제어기
 (3) 병합 제어기
 Fig. 1. Learning architectures for a neural network controller.
 (1) direct inverse controller
 (2) adaptive controller
 (3) combined controller

III. 제어기의 학습

다층 신경회로망의 입력층을 I, 은닉층을 H, 출력층을 O로 표기하고, 출력층의 뉴런의 수는 N_o , 은닉층의 뉴런의 수는 N_H , 입력의 수는 N_I 로 나타낸다. 출력층의 k번째 뉴런에서의 출력 o_k^O 은 다음과 같이 된다.

$$o_k^O = \Gamma \left(\sum_{j=1}^{N_H} W_{jk}^{HO} o_j^H + W_b^O \right) = \Gamma (U_k^O) \quad (5)$$

여기서,

$$U_k^O = \sum_{j=1}^{N_H} W_{jk}^{HO} o_j^H + W_b^O \quad (6)$$

이다. 은닉층에서의 출력 o_j^H 는 다음과 같다.

$$o_j^H = \Gamma \left(\sum_{i=1}^{N_I} W_{ij}^{HI} x_i^I + W_b^H \right) = \Gamma (U_j^H) \quad (7)$$

여기서,

$$U_j^H = \sum_{i=1}^{N_I} W_{ij}^{HI} x_i^I + W_b^H \quad (8)$$

이다. W_{jk}^{HO} 는 출력층의 k번째와 은닉층의 j번째 뉴런을 연결하는 웨이트를 나타내며, W_{ij}^{HI} 는 은닉층의 j번째와 입력층의 i번째 입력을 연결하는 웨이트를 나타낸다. W_b^O 와 W_b^H 는 각각 은닉층과 출력층 뉴런의 바이어스를 나타낸다. $\Gamma(\cdot)$ 는 일반적으로 다음과 같은 함수를 사용한다.

$$\text{sig}(U) = \frac{1}{1 + e^{-U}} \quad (9)$$

$$\text{tanh}(U) = \frac{1 - e^{-U}}{1 + e^{-U}} \quad (10)$$

$$\text{idn}(U) = U \quad (11)$$

신경회로망 제어기의 평가함수는 다음과 같이 정의 된다.

$$E_u(k) = \frac{1}{2} \sum_{k=1}^N (e_{uk}(k))^2 \quad (12)$$

여기서, $e_{uk}(k) = u_{dk}(k) - u_k(k)$ 이다. 이것은 출력층의 k번째 노드에서의 목표신호 $u_{dk}(k)$ 와 신경회로망의 출력 $u_k(k)$ 와의 오차를 의미한다. 본 논문에서 적용하는 시스템에서 입력 $u(k)$ 는 한개인 경우이므로 신경회로망의 출력 노드 수는 $N_o = 1$ 이다. 그러므로 k첨자는 1인 경우이다. 일반성을 유지하기 위해서 이 첨자를 계속 사용하기로 한다.

식(12)의 평가함수에서, $u_{dk}(k)$ 를 확정적으로 결정할 수 없기 때문에 $E_u(k)$ 가 유효하지 않다. 이 값을 결정하기 위해서는, 플랜트 출력과 목표 출력간의 오차를 이용해야 한다.

시스템의 평가함수는 다음과 같이 정의된다.

$$E_s(k+1) = \frac{1}{2} (e_s(k+1))^2 = \frac{1}{2} (y_d(k+1) - y(k+1))^2 \quad (13)$$

식(13)에서 $y_d(k+1)$ 는 결정할 수 있는 값이다. 그러므로 식(13)의 평가함수에 대한 계산을 할 수 있고, 오차값을 역전파시켜 식(13)가 최소가 되도록 신경회로망 제어기를 학습할 수 있다.

경사하강법 알고리즘을 이용하여 신경회로망 제어기의 은닉층에서 출력층으로 연결된 웨이트를 다음과 같이 갱신(update)할 수 있다.

$$W_{jk}^{HO}(k+1) = W_{jk}^{HO}(k) + \Delta W_{jk}^{HO} \quad (14)$$

$$\Delta W_{jk}^{HO} \propto - \frac{\partial E_y(k)}{\partial W_{jk}^{HO}} \quad (15)$$

여기서, $u_k(k) = o_k^O(k)$ 이므로

$$\frac{\partial E_y(k+1)}{\partial W_{jk}^{HO}} = -(y_d(k+1) - y(k+1)) \frac{\partial y(k+1)}{\partial u_k(k)} \frac{\partial o_k^O(k)}{\partial U_k^O(k)} \frac{\partial U_k^O(k)}{\partial W_{jk}^{HO}} \quad (16)$$

가 된다. 여기서, $\frac{\partial o_k^O(k)}{\partial U_k^O(k)} = \Gamma(U_k^O)$ 이고 $\frac{\partial U_k^O(k)}{\partial W_{jk}^{HO}} = o_j^H$ 이므로 잊식은

$$\frac{\partial E_y(k+1)}{\partial W_{jk}^{HO}} = -(y_d(k+1) - y(k+1)) \frac{\partial y(k+1)}{\partial u_k(k)} \Gamma(U_k^O) o_j^H \quad (17)$$

가 된다. 이것을 식(15)에 대입하면

$$\Delta W_{jk}^{HO} = \eta \delta_k^O o_j^H \quad (18)$$

이다. 여기서,

$$\delta_k^O = (y_d(k+1) - y(k+1)) \frac{\partial y(k+1)}{\partial u_k(k)} \Gamma(U_k^O) \quad (19)$$

이다. $\Gamma(\cdot)$ 는 사용하는 함수에 따라 각각 다음과 같은 형태가 된다.

$$\text{sig}(U) \rightarrow \dot{\Gamma}(U) = \alpha(1-\alpha) \quad (20)$$

$$\text{tanh}(U) \rightarrow \dot{\Gamma}(U) = (1+\alpha)(1-\alpha) \quad (21)$$

$$\text{idn}(U) \rightarrow \dot{\Gamma}(U) = 1 \quad (22)$$

입력층에서 출력층으로 연결된 웨이트의 갱신은 다음과 같다.

$$W_{ij}^{HI}(k+1) = W_{ij}^{HI}(k) + \Delta W_{ij}^{HI} \quad (23)$$

$$\Delta W_{ij}^{HI} = \eta \delta_j^H x_i^I \quad (24)$$

$$\delta_j^H = \sum_{k=1}^{N_o} \delta_k^O W_{jk}^{HO} \Gamma(U_j^H) \quad (25)$$

식(17)에서 $y_d(k+1) - y(k+1)$ 값은 측정가능한 값이고, $\Gamma(U_k^O)$ 는 신경회로망에서 구할 수 있는 값이다. 이 식 중에서 미지의 값은 $\partial y(k+1) / \partial u_k(k)$ 이다. 이 값은 다음장에서 플랜트를 학습한 신경회로망에서 유도하기로 한다.

IV. 자코비안의 계산

식(1)로 표현되는 시스템을 다층신경회로망으로 학습할 때, 신경회로망의 입력에는 $y(k), y(k-1), \dots, y(k-n), u(k), u(k-1), \dots, u(k-m)$ 을 지정하고, 목표 출력으로 $y(k+1)$ 을 지정한다. 학습을 하고 난 후에는

$$y(k+1) \cong NN(y(k), y(k-1), \dots, y(k-n), u(k), u(k-1), \dots, u(k-m)) \quad (26)$$

으로 된다. 플랜트의 동특성을 학습한 신경회로망은 식별기로 사용될 수 있다^[1,9]. 이것을 통해서 플랜트에 대한 자코비안정보를 간접적으로 구할 수 있다.

식별 신경회로망의 입력층은 i, 은닉층은 h, 출력층은 o로 표시한다. 입력층의 노드수를 N_i , 은닉층의 노드수를 N_h , 출력층의 노드수를 N_o 로 나타낸다. 수식 표현을 간단히 하기 위해서 한개의 은닉층을 갖는 신경회로망에 대해서 고려하기로 한다. 출력층의 k번째 뉴런과 은닉층의 j번째 뉴런에서는 다음과 같은 관계가 성립한다.

$$\frac{\partial o_k^o}{\partial o_j^o} = \frac{\partial o_k^o}{\partial U_k^o} \cdot \frac{\partial U_k^o}{\partial o_j^o} \quad (27)$$

여기서,

$$\frac{\partial o_k^o}{\partial U_k^o} = \dot{\Gamma}_k^o(U_k^o) \quad (28)$$

$$\frac{\partial U_k^o}{\partial o_j^o} = \frac{\partial}{\partial o_j^o} \left(\sum_{i=1}^{N_i} W_{ki}^{o} o_i^o + W_b^{ho} \right) = W_{kj}^{o} \quad (29)$$

이다. 식(28)와 식(29)로부터 식(27)을 다시 다음과 같이 나타낼 수 있다.

$$\frac{\partial o_k^o}{\partial o_j^o} = \dot{\Gamma}_k^o(U_k^o) W_{kj}^{o} \quad (30)$$

또한, 은닉층의 j번째 뉴런과 입력층의 i번째 입력에서는 다음 관계가 성립한다.

$$\frac{\partial o_j^h}{\partial x_i^i} = \frac{\partial o_j^h}{\partial U_j^h} \cdot \frac{\partial U_j^h}{\partial x_i^i} \quad (31)$$

여기서,

$$\frac{\partial o_j^h}{\partial U_j^h} = \dot{\Gamma}_j^h(U_j^h) \quad (32)$$

$$\frac{\partial U_j^h}{\partial x_i^i} = \frac{\partial}{\partial x_i^i} \left(\sum_{l=1}^{N_i} W_{jl}^h x_l^i \right) = W_{ji}^h \quad (33)$$

이다. 식(30)은 식(32),식(33)로부터 다음과 같이 나타낼 수 있다.

$$\frac{\partial o_i^h}{\partial x_j^h} = \Gamma_j^h(U_j^h)W_{ij}^h \quad (34)$$

출력층의 k번째 뉴런과 입력층의 i번째 입력에서는 다음 관계가 성립한다.

$$\frac{\partial o_k^a}{\partial x_i^h} = \sum_{j=1}^{N_h} \left(\frac{\partial o_k^a}{\partial o_j^h} \cdot \frac{\partial o_j^h}{\partial x_i^h} \right) \quad (35)$$

식(35)은 식 (30),(34)로부터 다음과 같이 된다.

$$\frac{\partial o_k^a}{\partial x_i^h} = \sum_{j=1}^{N_h} \left(\Gamma_k^a(U_k^a)W_{jk}^a \right) \left(\Gamma_j^h(U_j^h)W_{ij}^h \right) \quad (36)$$

출력이 한개인 시스템에서, 신경회로망의 출력층의 노드수 $N_o=1$ 이 되므로 첨자 k는 1인 경우가 된다. 식 (1)로 표현되는 시스템의 자코비안은 식(36)으로부터 각각 다음과 같이 나타낼 수 있다.

$$\frac{\partial y(k+1)}{\partial y(k)} = \sum_{j=1}^{N_i} \left(\Gamma_1^a(U_1^a)W_{j1}^a \right) \left(\Gamma_j^h(U_j^h)W_{ij}^h \right) \quad (37)$$

$$\vdots$$

$$\frac{\partial y(k+1)}{\partial y(k-n)} = \sum_{j=1}^{N_i} \left(\Gamma_1^a(U_1^a)W_{j1}^a \right) \left(\Gamma_j^h(U_j^h)W_{ij}^h \right) \quad (38)$$

$$\frac{\partial y(k+1)}{\partial u(k)} = \sum_{j=1}^{N_i} \left(\Gamma_1^a(U_1^a)W_{j1}^a \right) \left(\Gamma_j^h(U_j^h)W_{ij}^h \right) \quad (39)$$

$$\vdots$$

$$\frac{\partial y(k+1)}{\partial u(k-m)} = \sum_{j=1}^{N_i} \left(\Gamma_1^a(U_1^a)W_{j1}^a \right) \left(\Gamma_j^h(U_j^h)W_{ij}^h \right) \quad (40)$$

V. 자코비안 제어기의 설계^[7]

식(1)로 표현되는 이산시간 비선형 시스템에서, $f(\cdot)$ 는 미분가능한 함수이다. $\partial f(\cdot)/\partial u(k) \neq 0$ 이면, 이 시스템은 시간 k에서 국부제어가 가능(locally controllable)하다. 이 시스템에 대한 Lyapunov함수를 다음과 같이 정의한다.

$$V(k) = e^2(k) + K_1 \Delta e^2(k-1) \quad (K_1 > 0) \quad (41)$$

여기서, $e(k) = y_d(k) - y(k)$ 이고, $\Delta e(k) = e(k+1) - e(k)$ 이다.

이 시스템이 안정하기 위한 조건으로

$$\Delta V(k) = V(k+1) - V(k) \leq 0 \quad (42)$$

가 성립해야 한다. 이 조건을 계산하면 다음 관계를 얻을 수 있다.

$$\Delta y(k) = K(y_d(k) - y(k)) + \Delta y_d(k) \quad (0 < K < 2) \quad (43)$$

여기서, $\Delta y(k) = y(k+1) - y(k)$ 이고, $\Delta y_d(k) = y_d(k+1) - y_d(k)$ 이다. 식(1)의 시스템을 자코비안 선형모델로 근사화 하면 다음과 같은 식을 얻을 수 있다.

$$\Delta y(k) = \frac{\partial f}{\partial y(k)} \Delta y(k-1) + \dots + \frac{\partial f}{\partial y(k-n)} \Delta y(k-n) \quad (44)$$

$$+ \frac{\partial f}{\partial u(k)} \Delta u(k) + \dots + \frac{\partial f}{\partial u(k-m)} \Delta u(k-m)$$

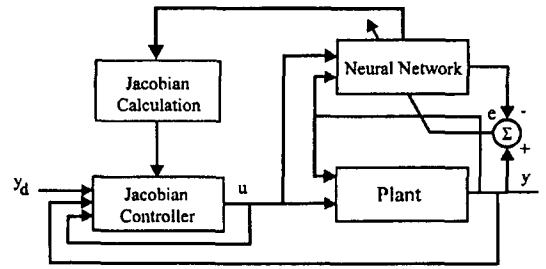


그림 2. 자코비안 제어기의 시스템

Fig. 2. Architecture of Jacobian controller.

각 자코비안은 플랜트를 학습한 신경회로망으로 부터 식(37)~식(40)에 의하여 계산할 수 있다. 식(43)에 식(44)을 대입하면 다음을 구할 수 있다.

$$\Delta u(k) = \left(\frac{\partial f}{\partial u(k)} \right)^{-1} \left(K(y_d(k) - y(k)) + \Delta y_d(k) \right.$$

$$- \left(\frac{\partial f}{\partial y(k)} \Delta y(k-1) + \dots + \frac{\partial f}{\partial y(k-n)} \Delta y(k-n) \right.$$

$$\left. \left. + \frac{\partial f}{\partial u(k-1)} \Delta u(k-1) + \dots + \frac{\partial f}{\partial u(k-m)} \Delta u(k-m) \right) \right) \quad (45)$$

여기서, $\Delta u(k) = u(k) - u(k-1)$ 이고, $\Delta u(k-m) = u(k-m) - u(k-m-1)$ 이다. 위에서 $\Delta u(k)$ 는 식(1)로 표현되는 비선형 시스템의 자코비안 계수를 이용한 선형화 기법으로 얻어진 제어측이다.

VI. 병합 제어기의 설계

신경회로망 제어기는 오랜시간을 학습한 후에 제어기로 사용할 수 있기때문에 학습을 시작하면서 제어를 수행하는 것은 불가능하다. 자코비안정보를 이용하여 설계한 제어기로 비선형 시스템을 제어하는데 있어서의 문제점은 자코비안 제어기는 플랜트의 입출력을 학

습한 신경회로망 식별기에서 자코비안을 계산하여 선형으로 제어기를 구성하기 때문에 비선형성이 심한 플랜트를 완전하게 선형화할 수 없다. 또한, 식별오차가 큰 경우에는 정확한 자코비안을 구하기 힘들다. 이 두 가지 방식상의 단점을 보완하기 위해, 자코비안 제어기와 신경회로망 제어기를 병합한 구조를 사용한다. 이 병합제어기의 신경회로망 제어기를 학습하는 방식에 따라, 오프라인 학습방식과 온라인 학습방식으로 병합 제어기를 설계하기로 한다.

1. 병합제어기의 오프라인 학습

신경회로망 제어기의 입력은 기준신호, 플랜트 출력과 과거값, 입력의 과거값이고, 목표신호는 자코비안 제어기의 출력값이다. 신경회로망의 초기 웨이트의 값은 $[-1, +1]$ 사이의 값으로 무작위로 정한다. 은닉층의 뉴런은 tanh함수를 사용하며, 출력층의 뉴런은 선형뉴런을 사용한다. 제어의 초기에는 자코비안 제어기로써 안정한 상태를 유지하고 이것을 기준으로 오프라인방식으로 신경회로망 제어기를 학습한다. 자코비안 제어기에서 생성된 $u_j(k)$ 를 목표출력으로 설정하여 신경회로망의 출력과 $u_j(k)$ 와의 오차의 허용범위 ϵ 이하에 이를 때까지 학습을 한다.

$$e = u_j(k) - NN, < \epsilon \tag{46}$$

신경회로망 제어기를 학습하는 목표신호는 자코비안 제어기에서 생성된 $u_j(k)$ 이다(식 45).

신경회로망의 학습을 마친 후에는 스위칭 동작에 의해서 자코비안 제어기에서 신경회로망 제어기로 전환한다. 이 때, 신경회로망 제어기는 온라인 방식으로 원하는 출력과 플랜트의 출력간의 오차를 식별 신경회로망을 경유하여 역전파시켜 제어기를 학습한다. 이때 적용되는 출력층에서의 델타는 다음과 같다.

$$\delta_k^o = (y_d(k+1) - y(k+1)) \left(\sum_{j=1}^{N_i} \Gamma_j^o(U_j^o) W_{jk}^{o} \right) \left(\Gamma_j^h(U_j^h) W_{(u+1)j}^h \right) \Gamma(U_j^o) \tag{47}$$

은닉층에서는

$$\delta_j^h = \sum_{k=1}^{N_i} \delta_k^o W_{jk}^{ho} \Gamma(U_j^h) \tag{48}$$

이다.

자코비안 제어기에서의 오차로 인해서 학습한 신경회로망은 오차가 포함된 출력을 하게 된다. 이 오차를 줄이기 위해서는 신경회로망제어기를 온라인으로 학습

해야 한다. 온라인 학습을 하지않고, 오프라인 학습만을 한 신경회로망 제어기는 적응형 시스템으로 동작하지 못하기 때문에 시스템 파라미터의 변동이나, 외란에 의한 영향에 대처할 수 없게 된다.

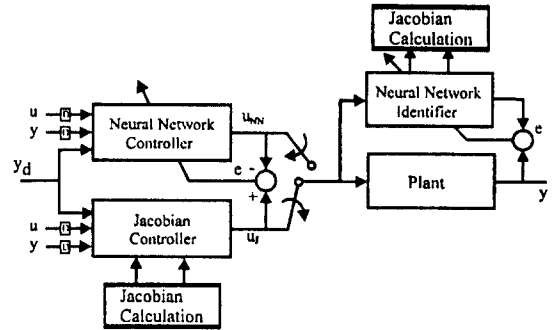


그림 3. 병합제어기의 오프라인 학습 구조

Fig. 3. Off-line learning architecture for combined controller.

2. 병합제어기의 온라인 학습

신경망 제어기의 출력을 $u_{NN}(k)$ 라 하고, 자코비안 제어기 출력을 $u_j(k)$ 라고 할 때, 전체 제어기는 다음과 같은 형태가 된다.

$$u(k) = u_j(k) + u_{NN}(k) \tag{49}$$

여기서, $u_{NN}(k)$ 는 신경회로망 제어기 출력 $o^o(k)$ 이다. 이것은 식(5),(7)로부터 다음과 같이 나타낼 수 있다.

$$u_{NN}(k) = \Gamma \left(\sum_{j=1}^{N_i} W_{jk}^{ho} \Gamma \left(\sum_{i=1}^{N_i} W_{ji}^h X_i^h + W_b^h \right) + W_b^o \right) \tag{50}$$

제어의 초기에는 신경회로망 제어기가 학습을 하지 않은 상태이기 때문에, 식(50)의 제어입력은 부정확한 값이 된다. 그러므로, 학습초기의 신경회로망 제어기의 역할을 최소화하도록 다음과 같이 웨이트를 설정한다. 은닉층과 출력층사이의 웨이트는 다음과 같이 정한다.

$$W_{jk}^{ho} = 0 \quad (j=1, \dots, N_{ih}, \quad k=1, \dots, N_o) \tag{51}$$

전체 제어입력은 자코비안 제어기에서 생성되는 제어입력과 신경회로망에서 생성되는 제어입력이다. 자코비안 제어입력만으로도 비선형 시스템의 근사적인 제어가 가능하다. 그러므로 온라인 학습구조의 병합제어기에서 생성된 제어입력은 자코비안 제어입력이 추가되고 신경회로망의 제어입력은 보조적인 역할을 한다.

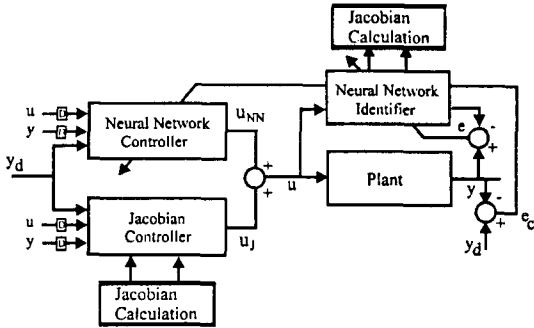


그림 4. 병합제어기의 온라인 학습구조
Fig. 4. On-line learning architecture for combined controller.

신경회로망 제어기의 온라인 학습에 적용되는 델타규칙은 오프라인 학습에 사용된 식(47), (48)의 경우와 같다.

온라인 학습에서의 오차값은 원하는 출력 $y_d(k+1)$ 과 플랜트의 출력 $y(k+1)$ 간의 오차값이다. 만일, 이 오차의 기준을 원하는 출력 $y_d(k+1)$ 과 식별신경회로망의 출력 $o^*(k)$ 으로 정하게 되면, 플랜트를 학습한 식별 신경회로망이 갖고 있는 식별 오차의 영향으로 인해서 기준신호를 제대로 추종하지 못하는 결과가 생긴다.

VII. 시뮬레이션

이산시간 비선형 플랜트는 다음과 같다.

$$y(k+1) = \frac{-0.9y(k) + u(k)}{1 + y^2(k)} \quad (52)$$

식(52)로 표현된 비선형 플랜트의 임펄스 패턴상으로 신경회로망을 학습시킨 후, 이 신경회로망은 플랜트의 특성을 모사하게 되어, 식별기로 사용될 수 있다. 위의 시스템을 학습하기 위한 신경회로망의 은닉층의 뉴런의 수는 $N_H=20$ 으로 정하고, 출력층의 뉴런의 수는 $N_O=1$ 로 정한다. 입력의 수는 $N_I=2$ 로 정한다. 입력의 첫번째 노드에는 $y(k)$ 를 인가하고 두번째 노드에는 $u(k)$ 를 인가한다. 은닉층의 뉴런에는 tanh함수를 사용하고 출력층에는 선형뉴런(idn)을 사용한다. 신경회로망의 초기 웨이트는 $[-1, +1]$ 사이의 값으로 무작위로 정한다. 식별신호는 $u(k) = \sin(2\pi k/25)$ 을 인가한다. 학습비율 $\eta=0.05$ 로 선택하여 10,000회 학습을 한다. 학습을 마친후의 평균 오차는 0.15이하이다.

신경회로망 제어기의 입력층의 수는 $N_I=3$, 은닉층

의 수는 $N_H=20$, 출력층의 수는 $N_O=1$ 이다. 은닉층에는 tanh함수를 사용하며, 출력층에는 선형 뉴런을 사용한다. 각 뉴런을 연결하는 웨이트는 $[-1, +1]$ 사이의 임의의 값으로 설정한다. 플랜트의 출력과 기준신호와의 오차를 신경회로망 식별기를 통해 역전파시켜 신경망 제어기를 학습시킨다. 학습비는 $\eta=0.05$ 이다. 그림 5은 신경회로망 제어기를 학습하는 과정을 나타낸 것이다. 학습을 시작한 직후에는 기준신호와 플랜트 출력간의 오차가 매우 크다. 대체로 300회 이상 학습한 후에 기준신호를 근사하게 추종하게 된다. 이 때의 기준신호와 플랜트 출력간의 오차는 0.028이하이고, 1000회 학습 후, 기준신호와 플랜트출력간의 평균 오차는 0.0064이하이다.

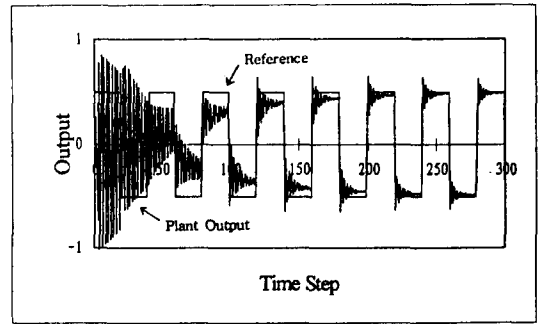


그림 5. 신경회로망 제어시스템의 출력과 기준신호
Fig. 5. Reference and output of neural network system.

그림 6에서는, 기준신호와 플랜트의 출력간의 오차가 아닌, 기준신호와 신경회로망 식별기의 출력의 오차를 역전파시켜 제어기를 학습하였을 때, 플랜트의 출력값이 기준신호를 추종하지 않고 오차가 누적되어 점차 발산하는 모습을 나타낸다. 초기의 제어는 안정한 상태에서 1000스텝 이후부터 기준신호와 플랜트간의 오차값이 점차 커진다. 1000스텝에서의 평균오차는 0.024이고, 1300스텝에서의 평균오차는 0.095이다.

그림 6은 자코비안 제어기를 사용한 결과이다. 자코비안은 플랜트를 학습한 신경회로망 식별기에서 구하게 되므로(식37~40) 신경회로망 식별기 오차의 영향이 자코비안 값에 나타난다. 식별오차가 적을수록 정확한 자코비안을 구할 수 있다. 현재의 자코비안 값은 k-1시간에서의 신경회로망에서 구하게 되므로 현재의 k시간에서의 제어입력에는 오차가 생기게 된다. 이 제어기는 온라인 방식이 아니므로 플랜트 출력과 기준신

호와의 오차가 계속 남게 된다. 기준신호와 플랜트간의 평균오차는 0.018이다.

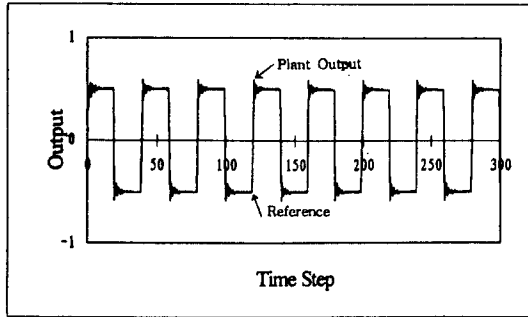


그림 6. 자코비안 제어 시스템의 출력과 기준신호
Fig. 6. Reference and output of Jacobian control system.

병합제어기의 오프라인 학습 구조는 자코비안 제어기의 $u_i(k)$ 를 목표신호로 정하여 신경회로망을 학습시킨 후에, 제어기로 사용하는 구조이다.

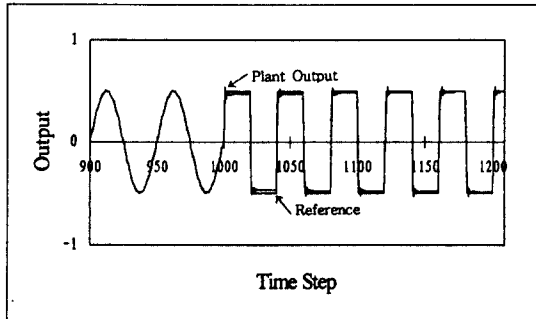


그림 7. 병합제어기의 오프라인 학습시의 플랜트 출력과 기준신호
Fig. 7. Reference and plant output of combined control system.(off-line learning)

그림 7에서, 신경회로망을 1000회 학습시킨 후에 제어기로 사용한 경우이다($\eta=0.05$). 신경회로망을 학습시키는 동안에는 자코비안 제어기만으로 시스템을 제어한다. 1000회의 오프라인 학습을 마친 후에 신경회로망 제어기는 기준신호와 플랜트 출력간의 오차를 온라인 방식으로 학습하게 된다. 오프라인 방식에서 온라인 방식으로 스위칭 동작이 있는 직후의 기준신호와 플랜트 출력간의 오차는 평균 0.035에서 200회의 온라인 학습한 후에 평균 0.017으로 감소한다. 신경회로망 제어기만을 사용하는 경우에 발생하는 초기의 큰

오차값 문제를 자코비안 제어기를 도입함으로써 해결할 수 있다.

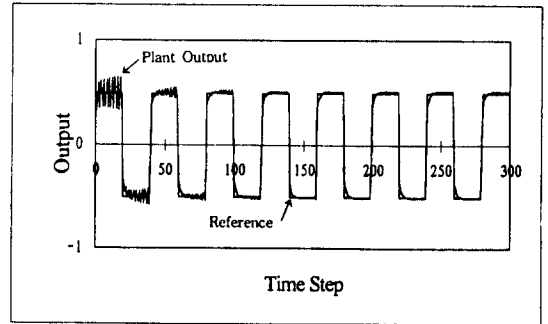


그림 8. 병합제어기의 온라인 학습시의 플랜트 출력과 기준신호
Fig. 8. Reference and plant output of combined control system.(on-line learning)

병합 제어기의 온라인 학습구조에서는 신경회로망 제어기의 출력층과 은닉층간의 초기 웨이트를 0으로 정하고, 입력층과 은닉층간의 웨이트는 $[-1, +1]$ 사이의 값으로 정한다. 학습비는 $\eta=0.05$ 으로 정한다. 그림 9에서, 50스텝이전의 출력은 학습이 제대로 되지 않은 신경회로망 제어기의 영향으로 인하여 오차값이 크게 나타난다. 이때 오차의 평균은 0.12이다. 150스텝 근처부터 안정한 출력값을 나타낸다(평균오차 0.02).

병합제어기의 오프라인 학습구조와 온라인 학습구조에서의 신경회로망의 역할은 다르다. 오프라인 학습구조에서는 자코비안 제어기의 기능을 학습하고 난 후에, 스위칭 동작 후에 제어기로 사용을 하지만, 온라인 학습구조에서는 자코비안 제어기의 오차부분에 대한 보상기로서 역할을 한다. 이 때의 신경회로망 제어기가 학습하는 값들은 자코비안 제어기의 오차부분에 해당하는 값들이므로 상당히 작은 범위의 값이다. 그림 7과 그림 8에서, 온라인 학습 방식과 오프라인 방식의 결과를 비교하여 보면, 오프라인 학습 방식의 결과가 우수하다는 것을 알 수 있다. 이것은 신경회로망이 안정한 자코비안 제어기를 학습한 후에, 스위칭 동작후 온라인으로 다시 오차에 대해서 학습을 하기 때문이다. 이에 비해, 온라인 학습방식에서는 초기의 제어오차가 다소 발생하며, 신경회로망이 학습하는 범위가 대체로 작은 영역이기 때문에 적절치 못한 학습비를 선정할

경우에는 상당히 오랜 학습시간을 소요한 후에 기준신호와 플랜트 출력간의 오차를 줄이게 되는 경우도 있다. 그러나, 오프라인 학습구조의 성능이 우수하지만 학습-동시-제어의 기능을 갖기 위해서는 온라인 학습구조를 사용하는 것이 바람직하다.

VIII. 결 론

본 논문에서는, 비선형 시스템의 제어를 위해 신경회로망 제어기와 자코비안 제어기를 병합한 제어기 구조를 제안하였다. 또한, 플랜트를 학습한 다층신경회로망에서 웨이트와 뉴런의 출력으로 표현되는 자코비안 관계식을 구하였다.

제안한 병합제어기는 신경회로망 식별기로부터 자코비안 관계식을 구하여, 이를 토대로 자코비안 제어기를 설계하였고, 또한 이 관계식을 신경회로망 제어기를 학습시키는데 사용하였다. 제안한 제어기는 식별기의 정보만을 이용하여 제어기를 설계하였으므로, 사전 정보가 많이 필요한 기존의 다른 제어기에 비해서 간단히 설계할 수 있었다.

병합제어기의 오프라인 학습구조에서는 신경회로망 제어기만을 사용하는 경우에 발생하는 학습 초기 오차 문제를 해결할 수 있었고, 온라인 방식으로 전환 한 후에 기준신호를 정확하게 추종하는 특성을 나타냈다. 온라인 학습 구조에서는 신경회로망 제어기의 학습초기부터 제어가 가능하여 학습-동시-제어(learning-while-functioning)기능¹¹⁰⁾을 하는 효율적인 제어기로 사용할 수 있음을 입증하였다.

참 고 문 헌

[1] K.J. Hunt, D. Sbarbaro, R. Zbikowski and P.J. Gawthrop, "Neural networks for control systems - A survey", *Automatica*, Vol. 28, No. 6, pp 1083-1112, 1992.

[2] D.Psaltis, A.Sideris and A.A. Yama

mura, "A multilayered neural network controller", *IEEE Control Systems Magazine*, Vol. 8, No. 2, pp 17-21, 1988.

- [3] M. Saerens and A. Soquet, "Neural controller based on back-propagation algorithm", *IEE Proceedings F*, Vol. 138, No. 1, pp 55-62, Feb. 1991.
- [4] X.Cui and K.G. Shin, "Direct control and coordination using neural networks", *IEEE Trans. System, Man, and Cybernetics*, Vol. 23, No.3, pp 686-697, May 1993.
- [5] D.H. Nguyen, B.Widrow, "Neural networks for self-learning control systems", *IEEE Control Systems Magazine*, Vol. 10, No. 2, pp 18-23, 1990.
- [6] M. Kawato, Y. Uno, M. Isobe, and R. Suzuki, "Hierarchical neural network controller", *IEEE Control System Magazine*, Vol. 8, pp17-21, Apr. 1988.
- [7] M. Iwata and S. Kitamura, "An input tracking control system using a forward model on neural network", *World Congress on Neural Network*, pp II-793~797, 1992.
- [8] T. Yabuta and T. Yamada, "Neural network controller characteristics with regard to adaptive control", *IEEE Trans. System, Man, and Cybernetics*, Vol. 22, No.1, Jan. 1992.
- [9] T. Yamada and T. Yabuta, "Dynamic system identification using neural networks", *IEEE Trans. System, Man, and Cybernetics*, Vol. 23, No.1, Jan. 1993.
- [10] M.M. Gupta and D.H. Rao, "Dynamic neural units in the control of linear and nonlinear systems", *World Congress on Neural Networks*, pp II-100~105, 1992.

저 자 소 개

趙奎翔(正會員) 第32卷 B編 第3號 參照

林濟鐸(正會員) 第30卷 B編 第5號 參照

현재 한양대학교 전자공학과 교수