

신경회로망을 이용한 비선형 화학공정의 실시간 학습제어

유 준 · 한중훈[†] · 장근수*

(주)삼성엔지니어링 기계팀

*공정산업의 지능자동화 연구센터/포항공과대학교 화학공학과
(1997년 12월 11일 접수, 1999년 1월 23일 채택)

On-Line Learning Control of Nonlinear Chemical Processes Using Neural Networks

June Liu, Chonghun Han[†] and Kun Soo Chang

Mechanical Engineering Department, Samsung Engineering Co. Ltd.

*Automation Research Center/Department of Chemical Engineering, POSTECH
(Received 11 December 1997; accepted 23 January 1999)

요 약

본 논문은 신경회로망에 근거한 비선형 동적시스템의 실시간 학습제어를 위한 직접 역제어의 개선된 구조와 학습방법을 제시한다. 제안된 방법은 기존 신경망을 이용한 직접 역제어의 경우 플랜트의 Jacobian을 정확히 구하지 못해서 발생하는 악영향을, 한 단계 앞 예측모델과 보다 강건하고 정확한 신경회로망의 연결가중치를 갱신하는 방법인 batch updating을 도입하여 해결하였다. 이 방법은 공정의 예측모델을 사용하므로 시간지연이 크고 품질변수를 실시간으로 측정할 수 없는 공정에도 직접 역제어를 적용할 수 있는 장점을 가진다. 제시된 개선된 구조의 학습방법은 receding horizon control으로도 해석될 수 있다. 제시한 방법을 회분식 반응기 개시조업의 최적 되먹임 제어에 적용한 사례연구를 통하여 보였다. 사례연구 결과, 제안한 방법은 기존 방법에 비해 좋은 제어성능을 보였으며 측정잡음, 입력외란, 모델-플랜트 불일치 등에 대한 강건성도 뛰어남을 알 수 있었다.

Abstract—This paper presents a modified neural network structure and a learning algorithm for direct inverse control of nonlinear and dynamic chemical processes. The major drawback of the typical direct inverse control, the difficulty of estimating the Jacobian for the plant, has been solved by introducing one-step ahead prediction model and batch updating rules. Using process prediction model offers the direct inverse control the ability to handle the processes with large time delays and quality variables that are difficult to measure on-line. The learning algorithm of the proposed architecture can be viewed as a receding horizon control. The proposed methods are illustrated by application case studies for optimal feedback control of batch reactor startup operations. The case studies have shown the proposed methods are better in terms of control performance and more robust to measurement noises, input disturbances, and model-plant mismatch than the previous ones.

Key words: Direct Inverse Control, Neural Networks, One-Step Ahead Prediction Model, Batch Updating Rule, Optimal Feedback Control

1. 서 론

회분식 반응기(batch reactor)의 개시조업(start-up operation)이나 연속공정의 사양변경조업(grade change operation) 등 화학공장의 많은 문제는 비선형 시스템의 최적제어와 연관되어 있다[1-3]. 예를 들어 연속공정의 사양변경조업의 경우 새로운 정상상태(steady state)에 도달하기 전에 생산된 제품은 원하는 사양을 갖추지 못하게 되어 싼 값에 판매하거나 폐기 처분하고 있는 실정이므로 최적제어를 통해 정상상태에 도달하는데 필요한 전환시간(transition time)을 줄여 생산성을 향상시키는 것이 매우 중요하다[3]. 그러나 실제로 현장에서

는 조업경험에 근거하여 오프라인(off-line)으로 제어입력계획(control input trajectory)을 구하고 폐루프(closed-loop) 제어를 사용하여 미리 설정된 궤적을 따라가게 하거나[4] 조업 전문가가 공정의 출력을 살펴면서 그때마다 제어입력을 바꾸어 주고 있다[3]. 이러한 방법은 과도한 시간과 노력이 소요되며 얻은 결과 또한 최적이라는 보장이 없기 때문에 보다 체계적인 접근방법인 최적제어기법의 도입이 절실히 요구된다.

최근 수치해석 방법의 발달과 급속한 컴퓨터 처리능력의 향상으로 개루프(open loop) 최적계획(optimal trajectory)을 아주 효율적으로 구할 수가 있게 되었으나 화학공정의 특징인 강한 비선형성과 복잡한 동특성때문에 최적계획계산의 실시간 수행은 거의 불가능하며 구한 궤적도 개루프이므로 작은 외란에 대해서도 민감하여 공정의

[†]E-mail : chan@postech.ac.kr

실제 제적이 미리 구한 최적제적을 벗어나게 된다. 더구나 이렇게 복잡한 화학공정에 대해 조업영역이 넓은 개시조업 및 사양변경조업에 맞는 정밀한 수학적 모델을 얻기가 매우 힘들고 많은 노력을 통해 얻은 모델도 시간이 지남에 따라 정확도를 잃게 되므로 최근에 각광을 받고 있는 모델 기반 제어이론을 적용하기도 어렵다.

인간의 학습기능을 수학적인 모델로 표현한 신경회로망은 위에서 언급한 문제점들을 풀기에 적합한 특성을 갖고 있어 최근 여러 분야에 도입, 적용되고 있다. 신경회로망은 뉴런(neuron)이라 불리는 작은 요소들로 이루어져 있으며 그 기능은 학습을 하는 동안 연결가중치(connection weight)들을 변화시키면서 결정된다. 비선형 써스시스템의 정적(static) 및 동적(dynamic) 특성을 근사화할 수 있는 신경회로망은 병렬계산처리가 가능하여 실시간 학습이 가능하고 공정의 수학적 모델이 아닌 입출력 정보만이 필요하며 공정이나 주위환경이 변화하는 경우에도 학습에 의해 적응할 수 있다. 그리고 학습에 의해 그 성능이 점점 더 향상되고 정보의 분산저장으로 일부 소자가 고장난 경우에도 성능이 많이 나빠지지 않는다. 이런 장점들을 가지고 있기 때문에 공정분야에도 신경회로망이 도입되어 공정의 모델링뿐만 아니라 공정 역모델링을 통한 제어기 설계에도 이용되고 있다[4-7]. 현재까지 제시된 신경회로망 제어기 종류에는 직접 역제어(direct inverse control)[4, 7], backpropagation through time[8], 신경적응제어(neural adaptive control)[9] 등이 있으며 그 적용분야도 로봇팔 제어[10], 직류모터 속도제어[11] 등 기계, 전자 분야는 물론 최근에는 증류탑[12], pH 공정[13] 등 화학분야까지 매우 넓다.

본 논문에서는 비선형 동적써스시스템의 실시간 학습제어를 위한 개선된 직접 역제어의 구조를 제시하고 회분식 반응기 개시조업의 최적 피드백 제어(optimal feedback control)에 이를 적용하였다. 기존의 직접 역제어는 노이즈(noise) 등의 원인에 의하여 플랜트의 Jacobian을 정확히 구하지 못하면 제어성능의 저하는 물론 공정의 안정성에도 영향을 미치게 되어[9] 제어기로 사용할 수 없는데[14] 본 논문에서는 신경회로망의 연결가중치를 갱신하는 방법 중에서 강건하고 정확한 batch updating 방법[15-17]을 사용하여 이 문제를 해결할 수 있었으며 이를 위해 도입한 공정의 한단계 앞 예측(one-step ahead prediction)모델로 인해 화학공정과 같이 시간지연이 크고 품질변수를 실시간으로 측정할 수 없는 공정에도 직접 역제어를 적용할 수 있게 되었다. 그리고 개선된 구조의 학습방법을 receding horizon control으로도 해석할 수 있었다. 개선된 방법을 이용한 제어기는 앞의 예제에 대해 좋은 성능을 보였으며 강건성 역시 뛰어나 측정잡음과 입력의란, 그리고 모델-플랜트 불일치(mismatch)에 대해서도 좋은 제어성능을 보였다.

2. 직접 역제어

여기서는 공정의 역모델(inverse model)을 제어기로 이용하는 직접 역제어와 그 종류에 대해서 간단히 설명하기로 하였다.

일련의 비선형 상미분방정식

$$\dot{x} = f(x, u, t), y = h(x, t) \tag{1}$$

x: 상태변수, y: 써스시스템 출력, u: 제어입력

으로 표현되는 시변(time varying) 동적써스시스템이 있을 때 써스시스템의 제어입력 u를 x와 t의 함수 $u = \phi(x, t)$ 로 나타낸다면 $\dot{x} = f[x, \phi(x, t), t]$ 는 페루프 써스시스템의 동특성을 표현하게 된다. 신경회로망 제어기의 목적은 매 샘플링 주기마다 원하는 출력 $y_d(t)$ 와 실제 출력 $y(t)$ 의 차이 $e_c(t)$ 를 점근적으로 0이 되게 하는 제어입력 $u(t)$ 의 함수 $\phi(\cdot)$ 를 신경회로망으로 근사하는 것이다. Goh 등[4]은 이러한 함수가 존재함을 수학적으로 증명하였으며 또 이러한 함수의 성질에 대해서 언급

하기도 하였다. 이때 함수 $\phi(\cdot)$ 로서 일반적으로 널리 사용되는 것이 공정의 역모델이며 이를 제어기로 사용하는 방법을 직접 역제어라고 하는데 이는 가장 대표적인 신경회로망 근거 제어기의 하나이다.

일반적으로 공정의 역모델을 제어기로 사용하면 완벽한 제어를 할 수 있다는 장점이 있으나 만약 공정이 nonminimum phase일 경우 공정의 right half plane zero나 시간지연(time delay)때문에 제어기가 불안정(unstable)해지거나 noncausal 즉, 존재하지 않을 수도 있다는 것은 널리 알려져 있다. 그러나 이 경우에도, Wiener-Hopf 식의 causal한 답을 역모델 대신 사용하면 제어기가 불안정해지는 것을 막을 수 있고[18, 19] 미래의 공정출력 $y(t+1)$ 대신 원하는 출력 $y_d(t+1)$ 을 사용하면 noncausality를 극복할 수도 있다. 또한 제어기가 공정의 역모델이 아니라 다른 제어법칙을 학습하게 하면 이러한 문제를 피할 수도 있는데 이에 대해서는 후에 더 자세히 설명하기로 한다.

신경회로망 제어기는 그 구조와 학습방법에 따라 다음과 같이 분류된다.

2-1. 간접학습구조(Indirect Learning Architecture)[7]

Fig. 1과 같이 원하는 출력 r을 바꾸어 가며 그에 따른 공정의 제어 입력과 출력을 가지고 공정의 역모델을 만든다. 다시 말해서 공정 제어입력과 그에 따른 출력을 신경회로의 출력과 입력으로 각각 사용하여 제어기를 학습시킨다. 이 구조는 공정의 원하는 운전범위에서 학습시킬 수 있는 장점을 가지고 있으나 신경회로의 오차 $|u - t|$ 를 최소화한다고 해서 써스시스템의 출력오차 $|r - y|$ 가 최소가 되지 않는 문제점이 있다.

2-2. 일반학습구조(Generalized Learning Architecture)[4, 7]

이 방법에서는 넓은 범위에서 공정의 제어입력을 바꾸어 그에 해당하는 출력을 가지고 2-1에서와 같이 신경회로를 학습시킨다(Fig. 2). 이 방법은 앞에서 언급한 제어입력 u의 함수관계를 신경회로망이 완벽하게 학습해야만 가능한데 이를 위해서는 신경회로망의 은닉층(hidden layer)의 뉴런(neuron) 개수가 충분히 많아야 하며[20] 공정의 제어입력 공간 전역을 span하는 학습데이터가 있어야 한다. 예를 들어 Goh 등[4]은 2축 로봇의 최적제어에 이 구조를 사용하였는데 제어기로 사용한 신경회로망은 각 은닉층(hidden layer)이 20개의 뉴런(neuron)을 가진 5층의 매우 큰 신경회로망이었으며 시뮬레이션을 통하여 얻은 무수히 많은 학습데이터를 가지고 학습을 시킨

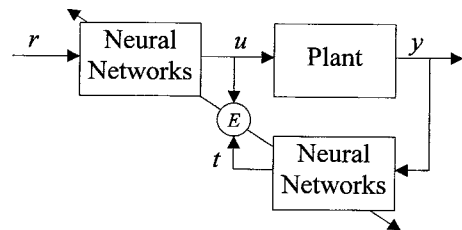


Fig. 1. Indirect learning architecture.

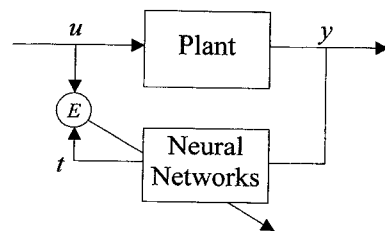


Fig. 2. Generalized learning architecture.

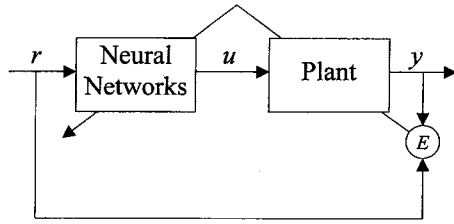


Fig. 3. Specialized learning architecture.

후 제어기로 사용하였다. 그러나 이러한 큰 신경회로망을 학습하는 데는 시간이 많이 걸리며 더구나 실제로 학습을 위하여 얻을 수 있는 현장의 조업데이터는 충분하지 않을 수도 있기 때문에 신경회로망의 완벽한 학습은 불가능하다. 그리고 이 방법은 원하는 운전범위에 국한되는 학습이나 실시간 학습을 시킬 수 없고 공정의 입출력이 일대일 함수관계가 있는 경우에만 적용된다[7, 21].

2-3. 특정학습구조(Specialized Learning Architecture)[7]

이 방법은 위에서 언급한 두 가지 방법의 단점을 극복하기 위해서 제안된 것으로 공정의 출력을 신경회로 제어기에 입력되는 원하는 출력에 근접하도록 학습시킨다(Fig. 3). 이 방법은 미리 설정된 제어를 tracking하는데 매우 적합할 뿐 아니라 임의의 목적함수를 최대(최소)화할 수도 있어서 최적제어에도 사용되며 원하는 운전범위에서의 학습이 가능하고 실시간 학습도 가능하다. 실제로는 앞의 일반 학습구조로 제어기를 학습시켜 초기화한 후 특정학습구조로 실시간 fine-tuning하여 사용한다.

3. 개선된 직접 역제어 구조

3-1. 특정학습구조와 그 단점

위에서 언급한 바와 같이 특정학습구조는 다른 두 구조와는 달리 직접 시스템의 출력오차 $|r-y|$ 를 최소화할 수 있는데 이는 공정의 Jacobian(differential gain)에 의해 공정출력의 차이를 공정의 제어입력 차이로 변환시키면 신경회로망의 역전파 학습방식(backpropagation learning rule)을 그대로 적용할 수 있기 때문이며 이를 오차의 공정 역전파(error backpropagation through the plant)라고 한다[7]. 이를 자세히 설명하면 아래와 같다.

제어기, 즉 신경회로망이 최소화할 목적함수 E_y 를 공정출력오차로 정의하면

$$E_y = \frac{1}{2} (r - y)^2 \tag{2}$$

가 된다. 출력오차 $(r - y)$ 를 e_y 라 하고 식 (2)의 양변을 신경회로망의 연결가중치 w 로 미분하면

$$\frac{\partial E_y}{\partial w} = -e_y \frac{\partial y}{\partial w} \tag{3}$$

$$= -e_y \frac{\partial y}{\partial u} \frac{\partial u}{\partial w} \tag{4}$$

가 되는데 $\partial y / \partial u$ 가 바로 공정의 Jacobian 또는 differential gain이다. 역전파 학습방법에서 w_{ji}^m 를 m번째 층의 i번째 뉴런과 m-1번째 층의 j번째 뉴런을 연결하는 가중치라 하면, 학습속도가 η 일 때 가중치의 변화 Δw_{ji}^m 는 delta rule에 따라서

$$\Delta w_{ji}^m \propto -\eta \frac{\partial E_y}{\partial w_{ji}^m} \tag{5}$$

$$= \eta \delta_j^m q_i^{m-1} \tag{6}$$

가 된다. 이때 δ_j^m 와 q_i^{m-1} 는 각각 역전파될 오차와 뉴런의 출력을 나타낸다. 그런데 플랜트를 신경회로망의 또 하나의 고정된 층이라고 가정하고 e_y 와 $\partial y / \partial u$ 를 이용하여 신경회로망의 마지막 층 n의 역전파될 오차 δ_j^n 를 아래와 같이 정의하면

$$\delta_j^n = g'(p_j^n) e_y \frac{\partial y}{\partial u} \tag{7}$$

시스템의 출력오차 e_y 를 플랜트를 통해 역전파시킬 수 있고 따라서 역전파 학습방식을 이용하여 출력오차를 최소화할 수 있다(여기서 g 는 신경회로망의 활성화함수(activation function)이다). Psaltis 등[7]은 Jacobian을 구하기 위하여 다음과 같은 근사식을 이용하였다. 여기서 k 는 샘플링 시간을 나타낸다.

$$\left. \frac{\partial y}{\partial u} \right|_k \cong \frac{y(u_k) - y(u_{k-1})}{u_k - u_{k-1}} \tag{8}$$

그러나 이 방법은 공정의 시간지연이나 측정지연(measurement delay)이 있으면 사용하기가 곤란하며 게다가 공정출력에 측정잡음이 섞이면 Jacobian값에 급격한 변화가 일어나고 결과적으로 연결가중치 변화량 Δw_{ji}^m 에 큰 변화가 생기게 되어 제어기의 성능은 물론 전체 시스템의 안정성에도 영향을 미치게 되므로[9, 14] 강건성을 높이기 위해 다른 신경회로망으로 플랜트 Jacobian을 근사하는 gain layer 구조[22] 등이 제시되었다. 그러나 이 방법을 사용하려면 실험을 통해서 제어입력을 조금 바꾸어 그에 따른 출력 변화를 측정할 후 gain layer를 학습시켜야 하는데 이러한 실험은 화학 시스템에서는 매우 힘들어서 사실상 불가능하기 때문에 화학 시스템에는 gain layer 구조를 사용할 수 없다는 단점이 있다.

3-2. 한 단계 앞 예측모델과 Batch Updating

화학공정은 다른 공정과는 달리 시간지연이 비교적 크며 또한 품질변수를 실시간으로 측정할 수 있는 경우가 드물기 때문에 제어를 하기가 힘들며 이 때문에 식 (8)을 이용하여 특정학습구조를 적용하기도 매우 어렵다. 이같은 문제에 대해 공정분야에서는 조업데이터를 기반으로 하여 공정의 모델을 만든 후에 이를 제어나 다른 용도로 사용하는 방법이 널리 사용되어 왔다[23, 24]. 이때 공정모델에는 예측모델뿐만 아니라 추론모델(inferential model)도 포함된다.

조업데이터로부터 공정의 한 단계 앞 예측모델을 만들면 위와 같은 경우에도 직접 역제어를 적용할 수 있을 뿐 아니라 batch updating의 이용도 가능한데 예측모델로부터 얻은 예측값 y 를 이용하여 식 (2)와 (4)를 다시 쓰면

$$E_y \cong \frac{1}{2} (r - y)^2 \tag{9}$$

$$\frac{\partial E_y}{\partial w} \cong -(r - y) \frac{\partial y}{\partial u} \frac{\partial u}{\partial w} \tag{10}$$

가 되고 k 부터 $k+H_p$ 까지 반복 예측한 후 평균한 결과를 이에 다시 적용하면

$$\bar{E}_y^k \cong \frac{1}{2H_p} \sum_{i=k}^{k+H_p} (r(i) - y(i))^2 \tag{11}$$

$$\left. \frac{\partial \bar{E}_y^k}{\partial w} \right|_k \cong -\frac{1}{H_p} \sum_{i=k}^{k+H_p} (r(i) - y(i)) \left. \frac{\partial y}{\partial u} \right|_i \frac{\partial u}{\partial w} \tag{12}$$

가 된다. $r(i) - y(i)$ 를 $e_y(i)$ 라 하고 여기에 신경회로망의 연결가중치를 갱신하는 방법 중 하나인 batch updating[15-17]을 적용하여 식 (7)을

아래와 같이 정의하면

$$\bar{\delta}_j^m = \frac{1}{H_p} g'(\mathbf{p}_j^n) \sum_{i=k}^{k+H_p} e_y(i) \left. \frac{\partial y}{\partial u} \right|_i \quad (13)$$

식 (5)와 (6)으로부터 k번째 샘플링때의 연결가중치 변화량은 아래와 같이 계산된다.

$$\Delta w_{\mu}^m(k) \propto -\eta \frac{\partial E_j^k}{\partial w_{\mu}^m} \quad (14)$$

$$= \eta \bar{\delta}_j^m q_{\mu}^{m-1} \quad (15)$$

Batch updating은 학습데이터 전체 혹은 일부를 반영한 후에 연결가중치를 갱신하는 것으로 각각의 학습데이터에 대한 gradient 값을 평균하여 연결가중치를 계산하게 된다. 이 batch updating은 역전파되는 오차에 대해 smoothing 효과[15]를 가지고 있으며 다른 학습방법 보다 gradient 값을 보다 정확하게 구할 수가 있어 학습데이터가 complete하다면, 즉 모든 가능한 입력과 출력을 포함한다면 gradient 값이 참값과 일치하게 된다[16, 17]. 따라서 기존의 특정학습구조에 한 단계 앞 예측모델을 도입하면 시간지연이 크고 품질변수를 측정하기 힘든 화학공정에도 특정학습구조를 적용할 수 있다. 그리고 한 단계 앞 예측모델을 도입함에 따라 batch updating을 이용하여 제어기의 연결가중치를 갱신할 수 있기 때문에 batch updating의 smoothing 효과를 이용하여 플랜트 Jacobian을 잘못 구하는데서 생기는 악영향을 최소화할 수 있으며 정확한 gradient 값을 구할 수 있기 때문에 제어기를 계속 사용함에 따라 제어기가 정확한 공정의 역모델에 근접하게 된다는 장점이 있다. 또한 제시된 방법은 k번째 샘플링때마다 k부터 k+H_p까지의 설정출력과 예측출력의 차이를 최소화하고 제어기의 연결가중치는 한 번만 갱신되며 공정에는 k번째 제어입력만이 적용되기 때문에 receding horizon control으로도 해석될 수가 있다. 이때 H_p는 신경회로망 제어기의 조율변수로 모델예측제어의 예측구간(prediction horizon)과 동일한 역할을 하며 여기서는 편의상 예측구간과 제어구간(control horizon)을 같게 하였다. 개선된 제어기의 구조는 Fig. 4에 나타내었다.

3-3. Nonminimum Phase 공정에의 적용

공정이 nonminimum phase일 경우 공정의 역모델을 제어기로 사용하면 문제가 생긴다는 것은 이미 앞에서 언급한 바 있다. 공정이 선형일 경우에는 Wiener-Hopf 식의 causal solution을 사용하면 이 문제를 해결할 수 있으나 이 방법은 비선형 공정의 경우에는 사용할 수 없고 더구나 실제로는 공정을 모르므로 신경회로망으로 구한 공정의 역모델을 제어기로 사용하는 직접 제어는 nonminimum phase 공정에는 곧바로 적용될 수 없다. 하지만 특정학습구조는 임의의 목적함수를 최소화할 수 있으므로 제어기의 목적함수로 식 (11)이 아닌 식 (16)을 사용하면 공정의 역모델을 사용하는에서 생기는 문제를 피할 수 있다.

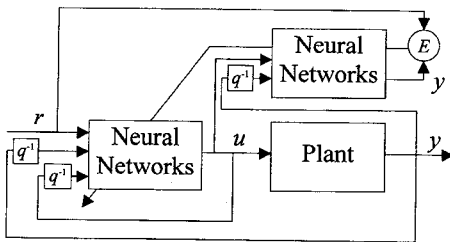


Fig. 4. Modified learning architecture.

$$J = \sum_k (r(k) - y(k))^2 + \rho(u(k))^2 \quad (16)$$

따라서 제어기를 설계할 때 일반학습구조로 공정의 역모델을 구하여 초기화하고 실시간 학습에서는 식 (16)을 목적함수로 사용하여 제어기가 공정의 역모델이 아니라 식 (16)을 최소화하는 최적 되먹임 제어법칙(optimal feedback control law)을 근사하도록 하면 nonminimum phase 공정의 역모델을 사용함으로써 생기는 문제를 피할 수 있게 된다.

4. CASE STUDY

4-1. 예제

여기서는 Luus[2]가 최적제어 문제로서 다루었던 회분식 반응기의 개시조업 문제에 대해서 제시된 방법을 적용해 보겠다. 반응은 A ↔ B의 가역 반응이며 반응기의 물질 및 에너지 수지는 다음과 같다. 여기서 x₁과 x₂는 각각 B의 농도와 반응기내의 온도이다.

$$\frac{dx_1}{dt} = (1 - x_1)k_1 - x_1k_2 \quad (17)$$

$$\frac{dx_2}{dt} = 300[(1 - x_1)k_1 - x_1k_2] - u(x_2 - 290) \quad (18)$$

초기 조건은 x^T=[0 380]이고 속도 상수는

$$k_1 = 1.7536 \times 10^5 \exp\left(\frac{-1.1374 \times 10^4}{1.9872x_2}\right) \quad (19)$$

$$k_2 = 2.4885 \times 10^{10} \exp\left(\frac{-2.2748 \times 10^4}{1.9872x_2}\right) \quad (20)$$

이며 제어입력은 0 ≤ u ≤ 0.36으로 제한되어 있다. 문제는 시간구간 0 ≤ t ≤ t_f에서 목적함수 J=x₁(t_f)를 극대화하는 제어입력궤적을 찾는 것이다. 본 논문에서는 x₁을 공정출력으로 가정하고 t_f=25분, 시간구간은 20개로 하여 최적 출력궤적과 그에 따른 제어입력궤적을 IDP (Iterative Dynamic Programming)[1, 2]를 사용하여 구하였다. 이때 출력의 샘플링 주기는 시간 구간의 크기와 같은 1.25분으로 하였다. 이를 reference 궤적, r(t)로 하고 부하변동(load disturbance)을 가정하여 이 reference 궤적과의 차이를 최소화하는 8개의 제어입력궤적을 다시 구하였다. 이때 외란으로는 x₂의 초기치에 ±1 범위 안의 변동을 무작위로 주었다. 현장 조업데이터가 부족한 상황을 가상하기 위해 reference 궤적을 포함한 9개의 공정입출력 궤적만을 조업데이터로 가정하여 신경회로망의 학습데이터로 사용하였다.

제어기와 한 단계 앞 예측모델의 입력은 각각 [u(t-1) u(t) y(t-1) y(t) r(t+1)]^T, [u(t) u(t+1) y(t)]^T으로 동적 모델링을 위해 moving window 개념[25]을 사용하였으며 출력은 각각 t+1에서의 제어입력과 출력이다. 제어기의 입력항에는 y(t-1)와 y(t)를 되먹임하여 모델링오차를 극복할 수 있도록 하였으며 r(t+1)항을 사용하여 noncausality를 극복함과 동시에 제어기가 feedforward 특성을 갖도록 하였다. 앞서도 언급했듯이 뉴런이 sigmoid 활성화수를 갖고 이러한 뉴런을 무한히 많이 가지고 있는 은닉층 1개로 구성된 다층 신경회로망은 모든 비선형 연속함수를 근사할 수 있다는 것이 수학적으로 증명되었으나, 실제로 특정 문제에 이를 적용할 때 최적 신경회로망의 구조, 즉 최적의 은닉층의 개수와 각 층의 뉴런 수를 결정하는 것은 매우 힘들다는 것이 이미 널리 알려져 있으며 이 때문에 신경회로망의 최적구조를 구하는 것 자체가 하나의 큰 연구분야를 이루고 있다. 이같은 이유로 신경회로망을 응용한 여러 논문에서는 시행착오법으로

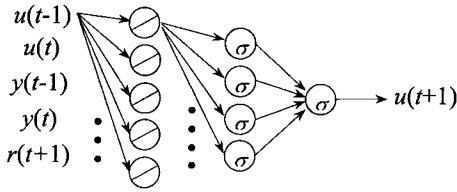


Fig. 5. Neural networks structure of inverse model.

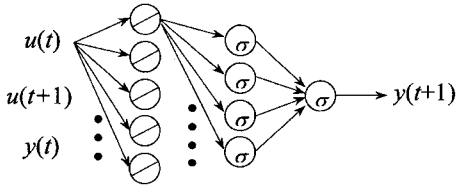


Fig. 6. Neural networks structure of plant model.

그 구조를 결정하는 것이 대부분이며 본 논문에서도 시행착오법을 이용하여 신경회로망의 구조를 결정하였다. 다만 공정의 역모델을 구할 때에는 공정의 역모델링은 일반적인 모델링과는 달리 불연속 함수를 근사하는 것이므로 Sontag[26]이 제시한 guideline을 따라 은닉층 2개를 갖는 신경회로망으로 그 범위를 제한하였고 출력층 뉴런의 활성화함수로 대칭 sigmoid 활성화함수를 사용하여 제어입력 제한을 다룰 수 있도록 하였다[4]. 역전파 학습방법을 사용하여 시행착오법으로 구한 두 신경회로망의 구조는 각각 (5-5-4-1), (3-5-4-1)으로 Fig. 5와 6에 나타내었다. 그림에서 대각선은 선형 활성화함수를, σ 는 대칭 sigmoid 활성화함수를 각각 나타낸다.

Moving window는 아래의 식과 같이 일련의 시계열 데이터를 각각 신경회로망의 입력과 출력으로 사용하는데 일반적으로 미래 공정의 n-step 앞의 예측을 위해서는 신경회로망의 입력 data로서 미래의 공정 출력값과 미래의 공정 입력값이 필요하다.

$$\begin{bmatrix} y(t+n) \\ \dots \\ y(t+1) \end{bmatrix} = f \left(\begin{bmatrix} y(t+m) \dots y(t) \dots y(t-m+1) \\ u(t+m) \dots u(t) \dots u(t-m+1) \end{bmatrix} \right) \quad (21)$$

그러나 현재시점에서 미래의 공정입출력값을 아는 것은 현실적으로 불가능하므로(noncausal) 본 논문에서는 다음과 같은 방법을 고안하여 이를 해결하였다. 공정의 다단계 앞 예측모델을 구성하는 대신 아래의 Fig. 7과 같이 공정의 역모델(제어기), f^{-1} 를 구성하고 공정의 한 단계 앞 예측모델, f 와 연결하여 예측값을 다시 입력값으로 사용하여 recursive하게 n번 반복 계산하면 causality를 어기지 않고 n-step 앞의 미래 공정출력값과 공정입력값이 구해지게 된다.

$$u(t+1) = f^{-1} \left([r(t+1) \dots y(t-m+1)]^T, [u(t) \dots u(t-m+1)]^T \right) \quad (22)$$

$$y(t+1) = f \left([y(t) \dots y(t-m+1)]^T, [u(t+1) \dots u(t-m+1)]^T \right) \quad (23)$$

4.2. 결과

IDP를 제외한 모든 프로그래밍과 시뮬레이션은 Pentium 133 MHZ

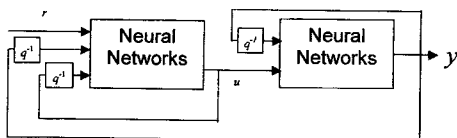


Fig. 7. Multi-step ahead prediction with one-step ahead prediction model.

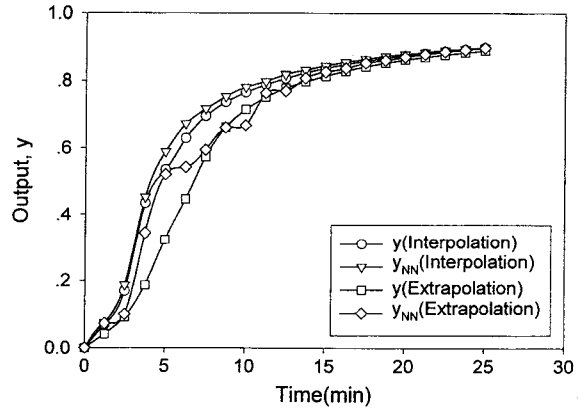


Fig. 8. Comparison of training data and identified result by NN plant model.

PC에서 MATLAB을 이용하여 수행하였고 IDP는 FORTRAN 77로 coding하여 IBM RS 6000/370 workstation에서 모사를 수행하였다. 개선된 제어기의 조율변수인 H_p 는 모두 4로 하였고 매 샘플링 시간마다 최대 학습횟수는 40번으로 제한하였다. 그리고 플랜트 Jacobian은 middle point formula를 이용하여 한단계 앞 예측모델의 예측값을 가지고 구하였다. 이때 개선된 제어기의 계산시간은 매 샘플링 때마다 최대 1.9초 미만으로 실시간 학습 및 실행에 적합하며 이는 학습시에 conjugate gradient 방법이나 Levenberg-Marquardt 방법 등의 최적화 방법을 사용한다면 더 줄일 수 있을 것으로 생각된다. 먼저 학습 데이터로부터 구한 한 단계 앞 예측모델의 성능을 Fig. 8에 나타내었다. 일반적으로 신경회로망이 학습데이터 범위 밖의 데이터에 대한 외삽성능이 좋지 않음을 고려하여 내삽의 경우와 내삽의 경우 모두에 대하여 예측해 보았는데 외삽의 경우 어느 정도 경향성은 잘 나타내고 있으나 SPE(Sum of squared Prediction Error)가 내삽의 경우(SPE=6.97e-3)보다 10배나 큰 7.74e-2이었다. 그리고 신경회로망으로 구한 제어기(공정의 역모델)의 경우에 예측결과와 예측오차를 Fig. 9와 10에 나타내었다. 공정 역모델의 경우 최대 예측오차가 약 0.15(Fig. 10)로 제어입력 범위의 크기가 0.36임을 고려할 때 매우 큰 값을 알 수 있고 이를 그대로 제어기로 사용하는 일반학습구조 제어기의 성능이 좋지 않음을 예견할 수 있다. 이 공정 역모델과 공정의 한 단계 앞 예측모델을 가지고 Fig. 2, 4와 같이 각각의 제어기를 구성하였다.

두 제어기의 성능을 비교해 보기 위하여 공정에 외란이 없는 상태에서의 페루프 제어결과를 Fig. 11에 나타내었다. Fig. 9와 10에서 예견된 바와 같이 일반학습구조의 제어성능은 매우 좋지 않은 반면, 개

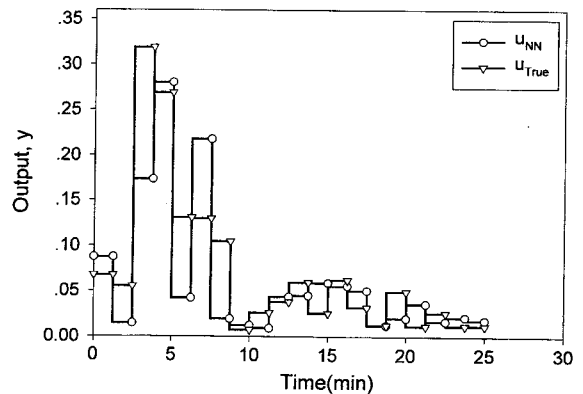


Fig. 9. Comparison of training data and identified result by NN inverse model.

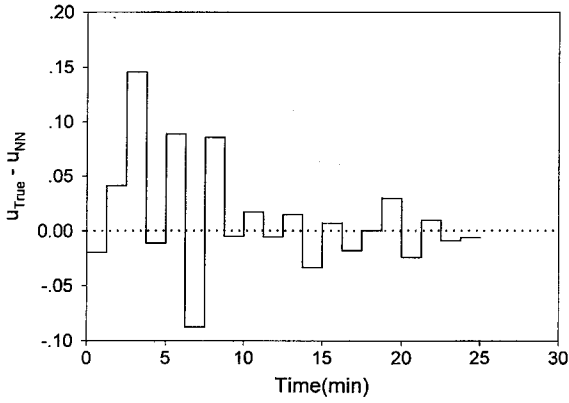


Fig. 10. Difference between true control values and predicted values.

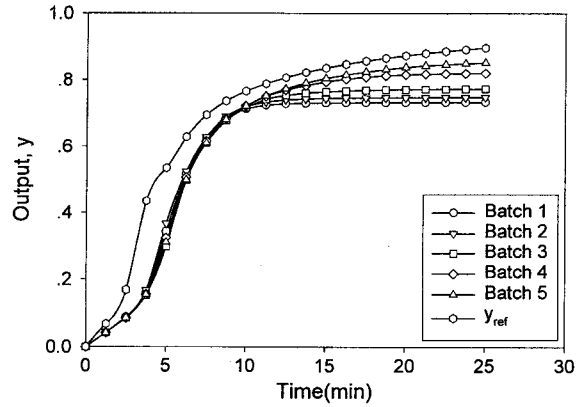


Fig. 13. Improvement of load disturbance rejection performance.

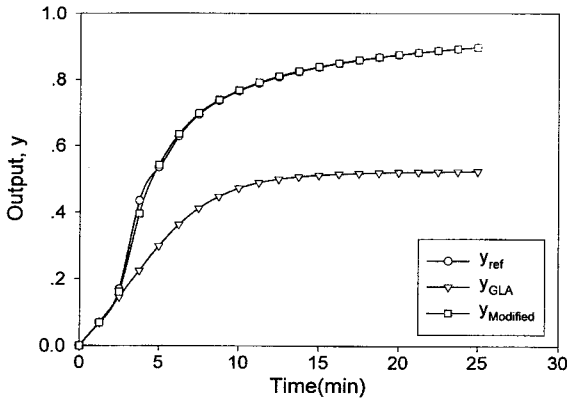


Fig. 11. Closed loop performance of modified architecture.

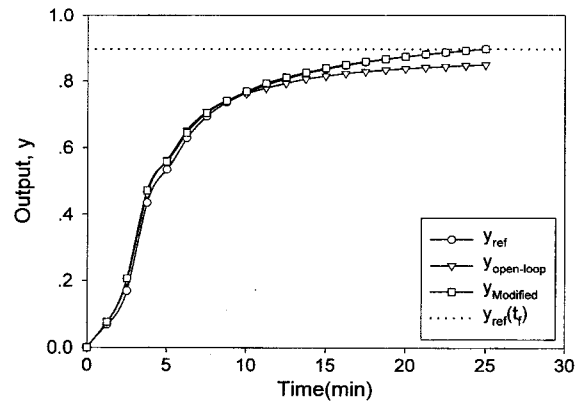


Fig. 14. Robustness to input disturbance.

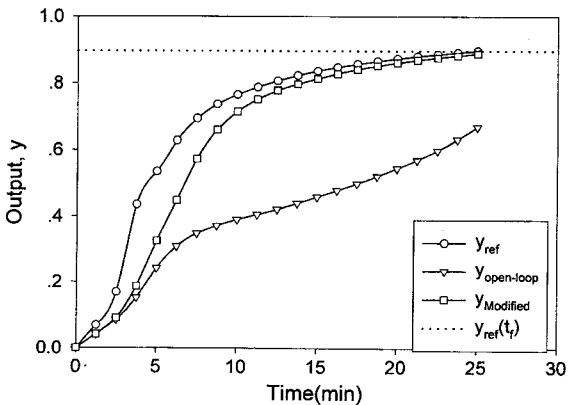


Fig. 12. Load disturbance rejection performance.

선된 구조는 학습데이터 즉, 현장 조업데이터가 부족한 상황에서도 실시간으로 공정을 학습하면서 연결가중치를 갱신하기 때문에 제어 성능이 일반학습구조에 비해 매우 우수함을 알 수 있다. 그리고 공정에 부하변동이 들어왔을 때의 제어기의 성능을 Fig. 12에 나타내었다. 부하변동은 학습데이터의 범위 밖인 x_2 (초기치)=375로 하였는데 학습데이터 범위 밖의 부하변동에 대해서도 일반학습구조의 제어기는 그 성능이 매우 나쁜 반면, 개선된 구조의 제어기는 제어결과 $y(t)$ 가 $r(t)$ 의 99.1%로 제어기가 부하변동을 거의 완벽하게 제거하는 것을 알 수 있는데 이는 제어기가 출력오차를 최소화하는 방향으로 학습을 하기 때문이며 제어성능은 여러 batch를 거듭할수록 더 향상될 것이다. 이를 알아보기 위해 부하변동이 $x_2=370$ 일 때의 제어

성능을 Fig. 13에 나타내었다. 처음에는 학습하지 못한 큰 부하변동을 제거하지 못하나 여러 batch가 거듭될수록 부하변동 제거 성능이 향상됨을 알 수 있고 성능향상 정도를 볼 때 1-2 batch후엔 $y(t)$ 와 $r(t)$ 의 차이가 거의 없을 것으로 생각되며 제어기의 조절변수라고 할 수 있는 H_p 와 최대학습횟수를 조절하면 제어기의 성능향상은 이보다 더 빠를 것으로 생각된다.

제어기의 강건성을 테스트하기 위해 입력외란, 출력 측정잡음, 모델링오차에 대해서도 페루프 모사를 하여 Fig. 14-16에 각각 나타내었다. 입력외란 Δu 는 제어입력 상한값(0.36)의 10%로 하여 매번 공정에 입력될 때마다 무작위로 주었는데 일반적으로 기존의 예측제어기법이 좋은 성능을 나타내지 못하는 입력외란에 대해서도 개선된 제어구조는 좋은 성능을 보임을 알 수 있다(Fig. 14). 이것은 외란들로 인한 모델의 예측값과 플랜트의 출력값의 차이가 되먹임될 때 한 단계 전에 측정된 출력값 하나만을 사용하는 기존의 비례-적분-미분 제어기나 DMC(Dynamic Matrix Control)와는 달리 ARX(Auto-Regressive with eXogeneous input) 모델과 같이 일련의 제어입력과 출력값들($u(t-1) u(t) y(t-1) y(t) r(t+1)$)이 제어기에 되먹임되므로 입력외란에 대해서 기존의 제어기들보다 더 좋은 강건성을 보장하기 때문이며 이는 측정잡음의 경우에도 해당될 것이다.

측정잡음은 샘플링 주기때마다 $r(t)$ (=0.8963)의 5%의 크기로 주었으며 모델링오차는 속도상수[식 (19), (20)]의 활성화 에너지(activation energy) 항에 10%의 오차를 주었다. 여기서 말한 모델링 오차란 공정이 변화하여 공정의 현 상태가 신경회로망으로 학습한 당시의 공정과 차이를 나타내게 되는 것을 의미한다. 이 모델링 오차, 즉 모델-플랜트 불일치에 대한 제어성능의 검증은 모델기반 제어에 있어서 매우 중요한데 공정의 역모델을 제어기로 사용하는 기존의 SLA에

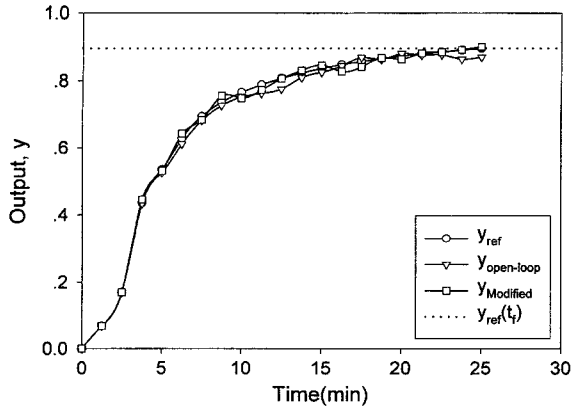


Fig. 15. Robustness to measurement noise.

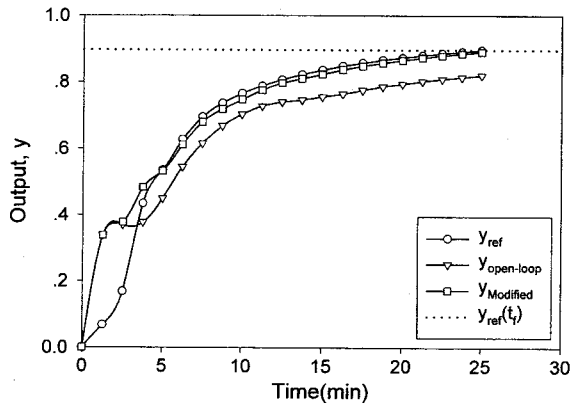


Fig. 16. Robustness to modeling error.

비해, 개선된 방법은 역모델뿐만 아니라 공정의 한 단계 앞 모델도 사용하기 때문에 모델링 오차에 대한 강건성 테스트는 더욱 중요하다 할 수 있다. 예제에서 반응속도 상수식의 활성화 에너지에 오차를 준 이유는 수식 (19), (20)에서 보듯이 강한 비선형성을 보이는 지수항이므로 공정에 가장 큰 영향을 끼치기 때문에 제어기의 강건성 테스트에 적합하다고 생각되기 때문이다. 제어기의 강건성을 테스트한 결과를 Fig. 15와 16에 나타내었는데 이로부터 개선된 제어구조가 측정잡음과 모델링오차에 대해서도 강건함을 알 수 있다. 측정잡음에 대한 강건성은 앞에서 언급한 일련의 제어입력과 출력값의 사용 때문은 물론, batch updating의 smoothing 효과때문에 연결가중치의 갱신 시에 측정잡음에 의한 플랜트 Jacobian의 급격한 변화가 감소되기 때문이며 모델-플랜트 불일치의 경우에도 제어기가 출력오차를 최소화하는 방향으로 실시간 학습을 하기 때문에 좋은 성능을 보이고 만약 공정 모델을 실시간으로 갱신하게 된다면 그 성능은 더 향상될 것이다.

5. 결 론

신경회로망을 이용한 기존의 직접 역제어 구조는 공정의 Jacobian을 잘못 구하면 제어성능이 크게 나빠져 시스템의 안정성에도 영향을 받는데 한 단계 앞 예측모델과 batch updating을 이용한 개선된 제어구조는 이를 극복함은 물론 시간지연 등이 큰 공정에도 쉽게 적용이 가능해졌으며 시뮬레이션을 통해 뛰어난 제어성능과 입력외란, 측정잡음, 그리고 모델링오차에 대한 강건성을 보였다. 그리고 이 신경회로망 제어기는 공정의 복잡한 수식이나 번거로운 시스템 식별 실험이 필요없고 과거의 조업데이터만을 가지고 구성할 수 있

으며 데이터가 불충분한 상황에서도 실시간으로 학습하기 때문에 제어성능이 향상됨은 물론, 공정에 변화가 생겨도 다시 조율할 필요가 없고 MIMO 시스템으로의 확장이 용이하다는 장점이 있어 앞으로 이에 대한 연구가 진행되어야 할 것이다.

감 사

본 연구는 한국과학재단 지정 우수 연구센터인 공정산업의 지능화 동화 연구센터의 연구비 지원에 의하여 이루어진 것으로 연구비를 지원해 주신 연구센터에 감사드립니다.

사용기호

- E : sum of squared error
- \bar{E} : averaged sum of squared error
- e : error
- f() : system dynamics
- g() : activation function
- H_p : prediction horizon
- h() : system output function
- J : performance index(object function)
- p : input of neuron
- q : output of neuron
- q^{-1} : backward shift operator
- r : command signal or desired value
- t : time
- u : control variable
- w : connection weight
- x : state variable
- y : system output predicted by plant model
- Δ : increment

그리스 문자

- $\phi()$: control law
- η : learning rate in error backpropagation
- δ : backpropagated error
- $\bar{\delta}$: averaged backpropagated error

상첨자

- k : sampling index
- m, n : m-th and n-th layer of neural networks

하첨자

- d : desired value
- f : final value
- i, j : i-th and j-th neuron
- k : sampling index
- ref : reference
- y : system output

참고문헌

1. Luss, R.: *Int. J. Control*, **52**(1), 239(1990).
2. Luus, R.: *J. Proc. Cont.*, **4**(4), 218(1994).
3. Lee, M. H., Han, C. and Chang, K. S.: *J. of Control, Automation*

- and Systems Engineering*, **3**(1), 46(1997).
4. Goh, C. J., Edwards, N. J. and Zoyama, A. Y.: *Opt. Cont. App. & Methods*, **14**, 1(1993).
 5. Morris, A. J., Montague, G. A. and Willis, M. J.: *Trans. IChemE*, **72**(Part A), 3(1994).
 6. Hunt, K. J., Sbarbaro, D., Zbikowski, R. and Gawthrop, P. J.: *Automatica*, **28**(6), 1083(1992).
 7. Psaltis, D., Sideris, A. and Yamamura, A. A.: *IEEE Cont. Syst. Mag.*, **4**(17), 17(1988).
 8. Nguyen, D. and Widrow, B.: *IEEE Contr. Syst. Mag.*, **10**(3), 18 (1990).
 9. Yabuta, T. and Yamada, T.: *IEEE Trans. Syst. Man & Cybernet.*, **22**(1), 170(1992).
 10. Wang, S. D. and Yeh, H. M. S.: *Int. Joint Conf. On Neural Networks*, June(1990).
 11. Yen, Y., Liu, T. Z. and Liu, D. Y.: *IEE Proc.-Control Theory Appl.*, **42**(5), 493(1995).
 12. Kim, S. J., Lee, M. and Park, S.: *Journal of Chem. Eng. of Japan*, **29**(5), 805(1996).
 13. Kim, S. J., Lee, M., Park, S., Lee, S. Y. and Park, C. H.: *Computers chem. Engng.*, **21**(2), 201(1996).
 14. Kim, S. J. and Park, S.: *ICASE MAGAZINE*, **2**(1), 48(1996).
 15. Fausett, L.: "Fundamentals of Neural Networks : Architectures, Algorithms and Applications", Prentice Hall, New Jersey, NJ(1994).
 16. Haykin, S.: "Neural Networks : A Comprehensive Foundation", Macmillan(1994).
 17. Hagan, M. T., Demuth, H. B. and Beale, M.: "Neural Network Design", PWS, Boston(1996).
 18. Widrow, B. and Walach, E.: "Adaptive Inverse Control", Prentice Hall, New Jersey, NJ(1996).
 19. Bode, H. W. and Shannon, C. E.: *Proc. IRE*, **38**, 417(1950).
 20. Hornik, K., Stinchcombe, M. and White, H.: *Neural Networks*, **2**, 359(1989).
 21. Jordan, M. I. and Rumelhart, D. E.: *Cognitive Science*, **16**, 307(1992).
 22. Kawato, M., Furukawa, L. and Suzuki, R.: *Biological Cybernetics*, **57**, 169(1987).
 23. Mejdell, T. and Skogestad, S.: *Ind. Eng. Chem. Res.*, **30**, 2543 (1991).
 24. Song, J. J. and Park, S.: *Journal of Chem. Eng. of Japan*, **26**(4), 347(1991).
 25. Bhat, N. and McAvoy, T. J.: *Computers chem. Engng.*, **14**(4/5), 573 (1990).
 26. Sontag, E. D.: Technical report, Rutgers Center for Systems and Control, Dept. of Mathematics, Rutgers University.