



신경회로망을 이용한 시스템 모델링 및 제어

최진영, 박현주

서울대학교 제어계측공학과, ERC, ACRC

1. 머릿말

생물체의 뇌 신경의 정보처리 메카니즘을 모방한 인공 신경회로망(artificial neural network)에 대한 관심이 한동안의 침체기를 거친후 근래들어 다시 고조되었다. 현재 생물체의 뇌 신경계 메카니즘은 완전히 규명되지는 못했지만 지금까지 밝혀진 뇌 관련 지식과 물리적/공학적인 모든 기법을 동원하여, 생물체의 정보처리능력을 모방하는 인공적인 신경회로망을 구현하고자하는 노력이 활발히 진행되고 있다. 과학적 관점에서 보면 인공 신경회로망이 얼마나 실제 생물체에 근접하는가에 관심이 많겠지만, 공학적 입장에서는 비록 실제 생물체 신경계와는 다를지라도 얼마나 유용하게 활용될 수 있는가에 더 관심이 있다.

이러한 관점에서 신경회로망(neural network)을 여러 공학적인 문제를 해결하는데 활용하고자하는 연구가 진행되어 왔으며, 시스템 모델링 및 제어분야도 중요한 응용대상 분야로 많은 관심을 끌었다. 현재까지는 기존의 제어기법으로 잘 안풀리는 문제에 적용하여 좋은 결과를 보여 줌으로써 적용가능성을 타진하는 연구와 실제 문제에 응용하는 연구가 주로 이루어졌으며, 이론적인 연구는 큰 진전을 보지 못했다.

최초의 뉴로제어 즉 신경회로망을 이용한 제어는 1963년에 Widrow와 Smith에 의해 시도되었다[1]. 이들은 최초의 신경회로망들중의 하나인 적응선형요소(ADaptive LINear Element : ADALINE)를 사용하였는데, ADALINE은 현재까지도 다른 신경회로망들에서 널리 사용되는 간단한 구조를 가지고 있다. ADALINE의 출력은 이산적으로, 단순히 두가지의 상태만을 표현할 수 있으며, 역진자 학습 제어에 이용되었다. Miche, Chamvers[2]와 Barto [3]가 역진

자 제어문제를 그들의 강화 학습제어(reinforcement learning control system)의 성능을 평가하는데 사용한 이후로, 역진자 제어문제는 다양한 제어방법을 평가하는 수준기표(bench mark) 문제로 자리잡았다. 1970년대에 Albus는 소뇌의 기능을 표형대로 모방한 CMAC라는 새로운 신경회로망을 만들었고 [4, 5], 이것을 로봇의 제어에 이용하였다. 1980년대초 이후로 CMAC은 비선형성이 심한 화학공정을 모델링하고 제어하는데 사용되었고[6], 1980년대 중반기에는 다시 로봇 응용분야에 이용되었다 [7]. CMAC은 원래 인식자(perceptron) 구조에서 나왔으나, 입력공간을의 이진 코딩(binary coding)하여 사용하는것을 보면, 의사 연속 출력(pseudo-continuous output)을 만드는 개선된 ADALINE이라고 볼 수 있다. 1980년대 동안 다양한 신경회로망 구조와 이것을 직접 하드웨어로 구현한 IC구조가 발표되었다. 한편, 강화 학습과 적응 구조가 광범위하게 연구되어 왔고 [8], 다층 신경회로망 [9], 방사형 기저함수망 [10], 동적 신경회로망 [11], 모듈형 신경회로망 [12]과 같은 새로운 신경회로망이 개발되었다. 한편 순환 신경망(recurrent networks) [37]은 플랜트 모델링과 추정 등에 이용되었다.

최근들어 응용뿐 아니라 기존의 제어 이론과 접목하여 이론적인 정립을 위한 연구가 시도되고 있고, 뉴로제어 시스템의 안정성등에 대한 연구 결과들이 발표되고 있다 [13]. 앞으로 신경회로망 자체의 연구 및 응용 연구와 더불어 이론 연구에서도 진전이 있을 것으로 기대된다. 본 원고에서는 뉴로제어의 배경 및 필요성, 기본적인 뉴로제어 시스템 구조 소개, 뉴로제어용 신경회로망 모델, 응용 예 등을 기술하고 앞으로 해결해야할 문제들을 고찰한다.

2. 뉴로제어의 배경 및 필요성

실세계 시스템은 비선형 요소를 많이 포함하고 있으나 현재까지의 제어기법은 선형 시스템에 기반한 제어 이론을 중심으로 발전되어왔다. 선형 제어 기법은 이론적으로 잘 정형화되어 있고 해석이 용이하다는 측면에서 매우 유용한 접근 방법이나, 실제 문제에 적용시는 비선형성의 정도에 따라 적용의 한계가 존재한다. 실제 시스템이 비선형 시스템이라 하더라도 레귤레이션(regulation) 문제라면 특정 동작점에서 동작하므로, 그 동작점 근처에서 선형화된 모델을 사용하면 어느 정도 제어 성능을 발휘할 수 있다. 그러나 만약 외부환경의 변화, 동작점의 변동 등으로 선형화된 시스템에 모델화되지 않은 동역학이 발생하게 되면, 이러한 비모델 동역학(unmodeled dynamics)이 제어 시스템 성능에 미치는 영향이 커져 시스템이 불안정해질 수 있다. 이러한 비모델 동역학은 선형시스템 관점에서 보면 불확실성(uncertainty)로 간주되며, 이는 구조적 불확실성(structured uncertainty)과 비구조적 불확실성(unstructured uncertainty)로 나눌 수 있다. 구조적 불확실성은 시스템의 구조를 고정한 상태에서 매개변수만 변동하는 경우이고, 비구조적 불확실성은 시스템의 차수등 구조자체의 불확실성을 의미한다. 이에 대처하는 제어 기법으로 자기동조 제어기(self-tuning regulator), 기준모델 적응제어(model reference adaptive control), 적응 예측제어, 자동동조 PID등의 적응 제어 기법과 H_∞ 등의 강인제어 기법이 있다. 그러나 이러한 기법도 제한된 불확실성 내에서 적용가능하며 비선형성이 강하거나 동작점의 범위가 넓은 경우는 또 다른 대책을 강구해야 한다. 이에 대한 대책으로는 동작점 별로 따로따로 제어기를 설계하고 동작점 변동에 따라 제어기를 교체하는 개념인 게인 스케줄링(gain scheduling)이 있다.

한편 비선형제어 접근 방법으로는 궤환선형화, 가변구조 제어등의 기법이 있으며 특정조건하의 비선형 동적모델을 알고 있는 경우 제어가 가능하다. 그러나 비선형 동적 모델링이 정확하지 않을 때 시스템의 안정성, 즉 강인성은 보장되지 않는다. 실제적으로 복잡한 시스템의 정확한 비선형 모델을 구한다는 것은 어려우며, 모델을 수학적으로 구하는 것 자체가 설게 비용이 많이 드는 작업이다. 시변 특성, 부정확한 모델링등에 대한 대처방법은 실시간 적응보상(adaptive compensation)을 필요로 한다. 또한 모르는 시스템을 모델링하기위해 비선형 시스템의 식별(system identification)기법을 사용할 수가 있다. 이때 임의의 비선형 시스템을 식별하기 위해서는 범용 함수근사자(universal function approximator)가 사용되며 신경회로망이 이 범용 근사자로서의 역할을 할 수 있음이 밝혀져 있다 [14]. 우선 먼저 인공 신경회로망이 모방하고자 하는 생물체의 뇌 신경

모델의 특성을 살펴보자.

- 뉴런간의 시냅스라는 연결가중치들에 의해 정보가 분산 저장된다. 이 분산 저장을 동역학 시스템에서의 예로 살펴보자. 임의의 선형시스템의 동역학은 상태변수의 선형 조합(linear combination)으로 표현된다. 즉 선형조합의 계수(coefficient)들은 선형 시스템에 대한 동역학 특성을 저장한다고 볼 수있다. 이 계수들의 값에 따라 입출력 패턴이 결정되며, 이 입출력 패턴들은 이 계수들에 분산저장된다고 볼 수있다. 생물체의 각 뉴런은 입력들의 선형조합 (또는 가중합)을 출력한다고 볼 수 있으며 선형조합의 계수는 연결가중치에 해당한다. 이 뉴런의 출력은 활성화 함수를 거친 다음에 다음 뉴런으로 전달된다. 즉 같은 형태의 선형 요소들이 무수히 많이 결합되어 임의의 비선형 사상 또는 비선형 시스템에 대한 정보를 저장할 수 있다.
- 대규모 병렬처리 형태로 정보를 처리한다. 간단한 기능을하는 뉴런들이 대규모 병렬처리함으로써 고속 응답특성을 갖는다.
- 적응능력이 있다. 외부의 입력에 따라 그에 적응하여 연결가중치를 갱신하는 메카니즘을 가지고 있다. 연결가중치는 정보를 저장하는 기본 요소이므로 연결가중치를 갱신하는 것은 새로운 정보를 받아들여 적응하는 과정으로 이해할 수 있다. 이미 제어분야에서 많이 적용되고 있는 선형 적응제어에서 제어기의 매개변수 갱신도 같은 맥락으로 생각할 수 있다.
- 학습능력이 있다. 학습은 적응된 정보가 저장된다는 것을 의미한다. 외부의 입력에 대해 계속 모든 연결가중치가 갱신된다면 최신 정보를 제외한 과거의 모든 정보는 잊어 버리게 된다. 다행히 생물체에서는 수많은 연결가중치 중에서 정보가 기억이 안된 연결가중치가 주로 적응함으로써 정보의 누적 즉 학습기능을 가지고 있다고 볼 수 있다.

이와 같은 특성을 인공 신경회로망으로 실현할 수있다면, 앞에서 언급한 선형 제어기법, 또는 비선형 제어기법에서의 문제점을 보상하기위해 다음과 같이 신경회로망을 활용할 수 있다.

- 선형제어기법 또는 비선형제어기법에서 부적확한 모델링에 의해 발생하는 비선형특성을 갖는 불확실성을 실시간 적응하여 보상하는데 신경회로망을 활용할 수 있다.
- 수학적인 방법으로 모델링이 어렵거나 불가능한 복잡한 시스템인 경우, 대상 시스템을 모델링하는데 활용할 수 있다. 즉, 모르는 임의의 비선형 시스템의 입출력 데이터로부터 원 시스템의 동역학특성을 근사적으로 나타내는 신경회로망을 적응 및 학습메카니즘을 통

해 찾을 수 있다. 물론 적절한 데이터의 선정, 적정 신경회로망의 구조 및 규모 선정, 타당성 확인(model validation) 등에 대한 연구를 요한다.

- 임의의 모르는 시스템을 제어하는 제어기를 신경회로망으로 구성하고 학습에 의해 원하는 제어 정보를 저장하여 제어기를 설계할 수 있다. 이 경우 오프-라인에 의해 학습이 이루어지며 온라인 상태에서는 신경망의 가중치는 갱신되지 않고 고정된다. 이 경우는 신경회로망의 학습기능을 이용하여 제어기를 설계하는 문제로 볼 수 있다.
- 온라인 상태에서 새로운 상황에 적응하기 위해 필요한 제어정보를 실시간으로 습득하여 적응하고, 적응된 정보는 저장되어 같은 상황이 반복될 시 적응 과도기간 없이 곧바로 응답하는 적응학습제어에 활용될 수 있다. 이 방법은 생물체의 동작특성을 가장 잘 모방하는 경우이나, 실제 구현에 있어서는 적응 및 학습시 안정도 및 수렴성 등이 엄격히 보장되어야 한다. 현재까지의 뉴로제어 기법은 이를 보장하지 못하며 이 문제를 해결을 위한 연구가 요망된다.

결국 뉴로제어는 알고 있는 시스템의 모델에 불확실성이 존재하거나, 시변특성이 있거나, 또는 시스템을 모르는 경우에 활용될 수 있는 것으로 요약할 수 있다. 모르는 플랜트를 오프라인 학습에 의해 모델링하고 제어기를 설계할 시는, 복잡한 시스템을 수학적으로 모델링할 필요가 없어서 설계 비용이 절약된다는 장점이 있다. 또한 시스템의 불확실성을 보완하고자 하는 경우에도, 뉴로제어는 뛰어난 적응능력을 가지고 있으면서 적응된 정보가 저장되는 특성으로 인하여 한번 경험된 (적용된) 상황에 대해서는 기존의 선형모델 기반 적응제어에 비해 빠른 적응능력을 갖는다. 선형모델 적응제어는 시스템의 구조를 선형으로 보고 제어기의 파라미터를 변화시켜 제어하는 방식이기 때문에 과거에 적용된 정보는 잊어 버리게 되어 상황이 반복되어도 처음부터 새로이 적응을 해야한다. 그러나 온라인 상태에서 적응 및 학습기능을 갖는 뉴로제어를 채택하기 위해서는 안정도, 수렴성 등에 대한 연구를 더 필요로 하며, 지금까지는 주로 시스템 식별 및 오프라인 학습제어에 신경회로망이 응용 되어 왔다. 다음 장에서는 지금까지 발표된 뉴로제어 구조 중에서 대표적인 몇가지를 소개한다.

3. 뉴로제어 시스템 구조

3.1 직접 역모델 뉴로제어

직접 역모델 뉴로제어(direct inverse model neuro-control)는 가장 보편적인 뉴로제어의 기본 개념으로, 제어대상 플랜트의 역모델을 학습하여 제어기로 사용하는 것이다. 플

랜트의 역모델을 구해서 제어하고자 하는 플랜트의 바로 앞에 붙이면, 시스템 출력은 기준입력과 같게 된다. 그림 1은 신경회로망을 이용한 직접 역모델제어 구조를 보여준다. 학습은 플랜트 입력 u 와, 플랜트의 출력 y 가 다시 신경회로망(A)를 통과한 후 나온 신호의 오차 e 를 줄이는 방향으로 이루어진다. 신경회로망(A)가 충분히 학습된 후, 구한 역모델(A)와 똑같은 신경회로망(B)을 플랜트 바로 앞에 연결하여 제어기로 사용한다.

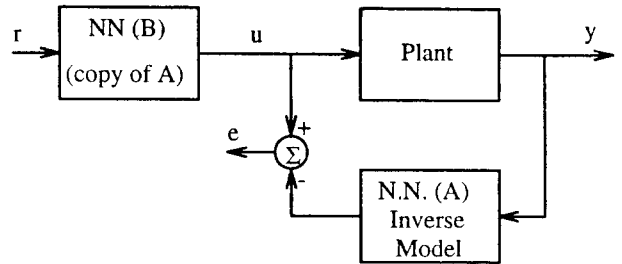


그림 1. 직접 역모델 뉴로제어 구조.

이 직접 역모델 제어는 뉴로제어의 기본 개념이므로 좀더 자세히 설명해보자. 설명의 편의상 상대차수가 d 인 단일 입출력 이산시간 비선형 시스템인 경우를 고려하자.

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

이때 $u, y \in R$ 는 각각 입력과 출력이고, $f: R^{m+n+1} \rightarrow R$ 는 임의의 연속 함수이라고 가정한다. 이 시스템에서는 시간 k 에서의 입력 $u(k)$ 가 시간 d 가 지난 후의 출력 $y(k+d)$ 에 영향을 준다. 즉, 시스템의 상대차수가 d 이다. 상기 시스템이 기준 궤적 $y_c(k)$ 를 따라가도록 제어하는 문제를 생각해 보자. 이때 제어 입력 $u(k)$ 는 다음과 같이 $u(k)$ 에 대한 f 의 역함수로 부터 구할 수 있다.

$$u(k) = f^{-1}(y_c(k+d), y_c(k+d-1), \dots, y_c(k+d-n), u(k-1), \dots, u(k-m)), \quad (2)$$

이때 대상 시스템이 제어가 가능하기 위해서는 다음의 가정을 필요로 한다.

가정 1: 함수 $f(\cdot)$ 는 $u(k)$ 에 대하여 역함수가 존재해야 한다.

그러나, 상기의 가정을 만족한다 하더라도 $u(k)$ 는 미래의 출력값인 $y(k+d-1), \dots, y(k+1)$ 의 함수가 되므로, 실제 시스템에서는 원하는 입력 $u(k)$ 를 구할 수 없다. 따라서, (1)에 있는 미래의 출력값들을 현재와 과거의 입출력값들로 바꾸어 한다. (1)을 $k+d-i$ 일 때의 출력값을 나타내는 식으로 바꾸면 다음과 같다.

$$y(k+d-i) = f(y(k+d-i-1), \dots, y(k+d-i-n), u(k-i), \dots, u(k-i-m)), i=1, \dots, d. \quad (3)$$

(3)을 반복하여 (1)에 대입하면 (1)은 다음과 같이 표현된다.

$$\begin{aligned} y(k+d) &= g(y(k-1), \dots, y(k-n), \\ &u(k), u(k-1), \dots, u(k-m-d)) \\ &= g(x(k), u(k)), \end{aligned} \quad (4)$$

여기서 상태변수 $x(k)$ 를 $x(k)^T := [y(k-1), \dots, y(k-n), u(k-1), \dots, u(k-m-d)]$ 로 정의한다. 위의 가정 1을 만족하면 함수 $g: R^{m+n+d+1} \rightarrow R$ 도 $u(k)$ 에 대한 역함수가 존재하므로 제어입력 $u(k)$ 를 다음과 같이 구할 수 있다.

$$u(k) = g^{-1}(x(k), y(k+d)), \quad (5)$$

여기서 시스템의 안정성을 위해서는 다음의 가정을 필요로 한다.

가정 2[13]: 주어진 시스템은 최소위상 (minimum phase) 시스템이다. 즉 주어진 시스템의 영점 동역학 (zero dynamics) 이 점근적으로 안정해야 한다.

이는 시스템의 역동역학이 안정해야 한다는 것을 의미한다. 그렇지 않은 경우 내부적으로 불안정하거나 제어입력에 스위칭현상이 발생할 수 있다. 결론적으로 대상 시스템은 현재 입력에 대한 역함수가 존재해야 하고, 역동역학 (inverse dynamics or zero dynamics)이 안정해야 뉴로제어의 대상이 될 수 있다.

상기 두 가정을 만족하는 임의의 모르는 시스템에 대한 뉴로제어를 설계하는 것은 (5)의 제어입력을 충분히 근사하여 출력하는 신경회로망 $N(W, x(k), y(k+d))$ 를 오프라인 또는 온라인 상황에서 찾는 것으로 볼 수 있다. 여기서 W 는 신경회로망의 가중치인 조정매개변수 (adjustable parameter)를 나타낸다. 즉, 뉴로제어기 N 의 출력 $\hat{u}(k)$ 가

$$\hat{u}(k) = N(W, x(k), y(k+d)), \quad (6)$$

일 때, 임의의 ϵ 및 모든 제어가능 입력 영역 및 제어대상 출력 영역에 대해 $|u(k) - \hat{u}(k)| \leq \epsilon_1$ 을 만족하도록 학습 메카니즘에 의해 뉴로제어기를 적응시키는 것이 뉴로제어의 기본 개념이다. 이것은 결국 $|y_r(k) - y(k)| \leq \epsilon_2$ 를 만족시키는 문제와 일치하게 된다.

이 제어구조는 역모델 신경회로망이 바로 제어기로 사용되므로, 제어성능은 역모델의 정확도에 크게 의존한다. 특히 플랜트가 가정 1을 만족하지 않아 역모델이 유일하지 않거나 가정 2를 만족하지 않는 경우는 이 제어구조로는 제어가 불가능하다. 또한 궤환이 없으므로, 시스템의 견실성이 나쁘다. 그러나, 이 견실성 문제는 역모델 신경회로망의 온라인 학습을 통해 파라미터를 온라인으로 적응시킴으로서 어느정도 보완할 수 있다.

이러한 기본적인 뉴로제어 개념인 역모델제어를 근간으로

다양한 형태의 제어 구조가 제시되었으며, 기존 제어기법과의 결합을 위해 변형된 형태도 연구되었다. 지금부터, 현재까지 제안된 많은 뉴로제어구조 중에서 대표적인 몇 가지를 좀더 살펴보자

3.2 모델기준 적응 뉴로제어

모델기준 적응 뉴로제어 (model reference adaptive neuro-control) 구조는 선형제어 분야에 널리 사용되어온 모델기준 적응제어를 비선형인 경우로 확장된 개념이다. [15] 이 구조에서는 기준 모델을 정하고, 플랜트의 출력 $y(t)$ 가 기준 모델의 출력 $y_r(t)$ 를 점근적으로 따라가도록, 즉,

$$\lim_{t \rightarrow \infty} \|y_r(t) - y(t)\| \leq \epsilon, \quad \epsilon > 0 \quad (7)$$

이 되도록 제어입력을 조절한다[16]. 그림 2은 모델기준 적응 뉴로제어의 전체적인 구조를 보여준다. 먼저 플랜트와 식별 신경회로망(NN2)의 입출력 특성이 같아지도록 NN2를 학습시킨 뒤, NN2를 고정하고 전체 시스템의 출력과 기준모델의 출력이 같게되도록 오차 역전달 방법에 의해서 신경망 NN1을 학습시킨다. 이 제어구조의 성능은 적절한 기준모델의 선택에 영향을 받으며, 기준모델의 전달함수가 1인 경우에는 제어로 사용되는 신경망 NN1은 플랜트의 역모델이 된다.

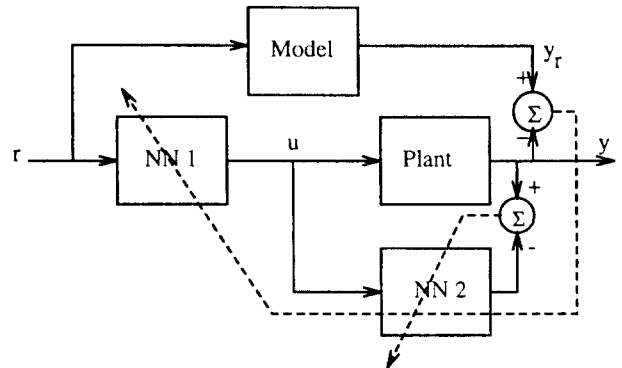


그림 2. 모델기준 적응 뉴로제어 구조.

3.3 예측 뉴로제어 predictive neuro-control

예측제어(predictive control)는 미래의 출력값들이 원하는 값들이 나오도록 제어입력을 계산하는 방식으로, 예측 뉴로제어는 선형시스템에 적용되어 온 예측제어 기법을 뉴로제어에 적용한 것이다. 예측제어 방법 중 이동 구간 (receding horizon) 방식은 미래의 어느 구간 동안 최적의 출력이 나오도록 제어입력을 계산한다. 이것의 구조가 그림 3에 나와 있다. 먼저 신경회로망을 이용해서 제어대상 시스템과 동일한 입출력 특성을 갖는 모델(NN2)을 구하고, 이 신경망 모델(NN2)로부터 특정 구간에서의 미래의 출력값들을 얻는다. 이 미래 출력값들이 최적의 출력값이 되도록 최적

용되는 신경망(NN1)의 출력이, 최적화 루틴의 출력으로 나오는 최적 제어 입력값 u 와 같도록 신경망(NN1)을 학습시킨다. NN1의 학습이 끝나면, 최적화 루틴의 역할을 NN1이 하므로 앞서 사용되었던 최적화 루틴의 필요 없이 신경망(NN1)의 출력이 바로 제어입력으로 사용된다. 이 방법은 Montague 등에 의해 연구되었다[17,18,6].

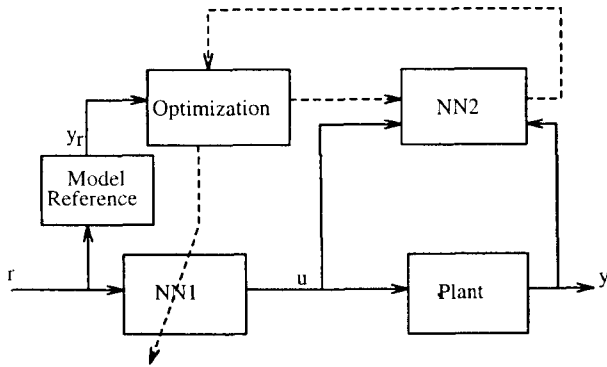


그림 3. 예측 뉴로제어 구조.

3.4 내부 모델 뉴로제어

내부 모델 뉴로제어 (internal model neuro-control) [19]은 그림 4에서 보는 바와 같이 예측 뉴로제어(predictive neuro-control)와 비슷한 구조를 사용한다. 먼저 신경회로망(NN2)이 플랜트와 같은 입출력 특성을 갖도록 학습된 후, 신경망 모델(NN2)과 플랜트 출력 사이의 오차가 되먹임되어서 제어기로 사용되는 신경망(NN1)의 입력으로 들어온다. 결국 이 내부모델 뉴로제어기로 사용되는 신경망(NN1)은 플랜트의 역모델이 되도록 학습된다. NN2가 플랜트의 특성을 완벽히 학습하면 되먹임 오차는 0이 되고 앞의 직접 역모델 제어기와 같게 된다. 이 제어구조의 구현은 개루우프(open-loop) 상태에서 안정한 플랜트에 대해서만 가능하다. 이런 제약조건에도 불구하고, 안정성에 대한 이론을 바탕으로 비선형 시스템에의 적용 결과가 소개되었다 [20].

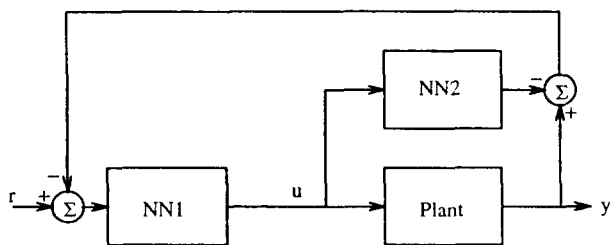


그림 4. 내부 모델 뉴로제어 구조.

3.5 되먹임 오차 학습제어

되먹임 오차 학습제어 (feedback error learning control) 는 그림 5에서와 같이 되먹임 제어와 앞먹임 제어(feedfor-

ward control)를 혼용한 경우로 앞먹임 제어에 신경회로망을 사용한다. 그림에서 고정된 선형 되먹임 제어기는 초기 상태에 시스템을 안정하게 하는 효과를 가져오며, 앞먹임 제어기 (NN)를 학습시키는데 사용된다[21,22,23]. 여기서 사용된 되먹임 제어기는 폐루우프(closed-loop) 시스템이 넓은 동작점에서 안정하도록 설계되는 것이 바람직하며, 선형 제어기에 의한 폐루우프 시스템의 성능은 현재의 동작점에 영향을 받게 된다. 앞먹임 제어기로 신경회로망을 이용한 학습제어기(NN)를 사용함으로써 동작점이 바뀔 때마다 신경회로망 제어기는 각 동작점에서의 적절한 비선형 모델을 학습하게 되고, 동작점이 다시 바뀔 때에도 전에 학습한 결과를 저장하게 된다. 이 방법에서 앞먹임 제어기는 결국 시스템의 역모델을 학습하게 되어 정확한 학습이 이루어졌을 때 되먹임 제어기는 역할을 하지 않게 된다. 이 제어 알고리즘의 단순함에도 불구하고 안정한 되먹임 제어기를 미리 설계해야만 하는 것이 이 제어구조의 단점이다. 또한, 되먹임 제어기의 동작상태가 나쁠 때에는 앞먹임 제어기의 학습속도가 느려지게 된다. 하지만, 안정한 되먹임 제어기를 사용한 경우 전체적으로 이 제어구조는 견실하다는 것이 알려졌다.

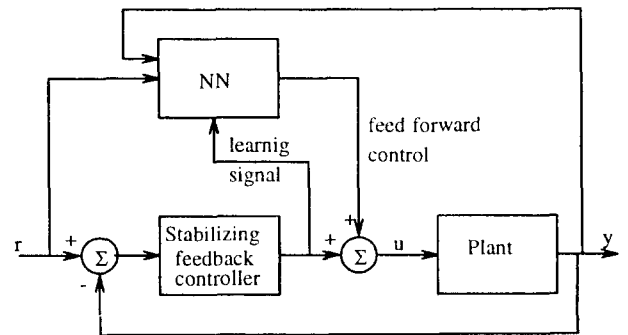


그림 5. 되먹임 오차 학습 제어 구조.

3.6 강화 학습제어

강화 학습제어 (reinforcement learning control) 기법은 최소한의 교사 정보를 이용하여 학습하는 일종의 지도학습 알고리즘이다. 학습에 사용하는 유일한 정보는 특정한 제어 신호가 성공적이었는가 그렇지 않았는가 하는 것이다. 이 기법을 처음 발표한 Barto [3]의 논문이후로 신경회로망과 밀접한 관계를 갖으며 연구되었다. 처음 강화 학습제어의 응용은 역진자의 균형유지 문제에 적용되었다. 이때, 지지대(platform)를 초기 지점에서 어떤 거리 이상을 움직이지 못하도록 제한하고, 역진자는 수직상태를 계속 유지하도록 했다. 만약 이 두 제약조건 중 하나라도 만족시키지 못하면, 실패 신호가 학습 알고리즘에 전달되어 학습이 진행된다. 일단 제어기가 역진자의 균형을 잘 유지하도록 학습이 이루어 지면 실패가 드물게 일어나게 되고, 따라서 학습은 거의 이루어지지 않는다.

Barto에 의해 제안된 방법은 그림 6에서와 같이 두 개의 적응 요소로 구성된 제어 구조로 구성되어 있다. 이 두 개의 적응 요소는 연상 탐색 요소(ASE : Associative Search Element)와 적응 평가 요소(ACE : Adaptive Critic Element)이다. ASE는 주어진 목적을 만족하는 최적의 제어입력을 생성하는 기능을 수행한다. 한편 ACE는 내부적으로 제어기의 성능을 살펴보고, ASE를 학습하는데 필요한 내부 강화 교사 신호를 ASE에 제공한다. ACE의 학습은 외부 실패/성공 신호를 통해서 이루어진다. 이 두 적응 요소의 학습이 연속적으로 이루어짐에 따라 전체 시스템의 성능은 점진적으로 향상되도록 되어 있다. Barto 이후 이 기법의 변형된 구조에 대한 시뮬레이션과 응용 연구가 있었고[24, 25, 26, 27], 전체 시스템에 대한 더 많은 이론적인 연구가 진행되었다[28, 29].

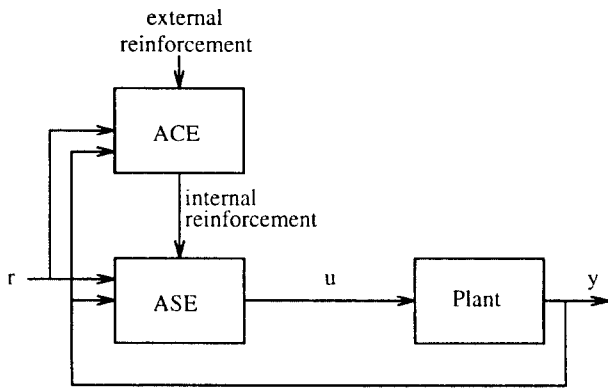


그림 6. 강화 학습 제어 구조.

3.7 자동조정 되먹임 뉴로제어

PID 제어 기법을 비롯한 선형 되먹임 제어기는 산업적 유용성, 시스템의 견실성, 폐 루우프 안정성, 값싼 구현비용 등의 이유로 산업 현장에서 많이 이용되고 있다. 이 경우 동작점의 변동에 따른 비선형성의 영향을 보상하기 위해 게인 스케줄링(gain-scheduling) 형태의 접근방법을 많이 사용한다. 이 게인 스케줄링 방법은 동작점에 따라 각각 선형되먹임 제어기를 설계하고, 설계된 제어기의 동작점에 따른 매개변수는 테이블 형태로 저장하며, 동작점 조건에 따라 테이블 탐색방법으로 제어기의 매개변수를 실시간 바꾸어 주는 방법이다. 자동조정 되먹임 뉴로제어(auto-tuning feedback neuro-control)는 이 게인 스케줄링 방법을 신경회로망으로 대체하여 기존 선형 제어기의 파라미터를 온라인으로 바꾸어주는 제어기법이다. 이 방법은 게인 스케줄링에서 각 동작점별 제어기 설계과정을 학습방법으로 대체 할 수 있다는 장점이 있으며, 또한 테이블 탐색없이 되먹임 제어기의 매개변수를 실시간 자동조정할 수 있다.

초기에 Kumar와 Guez[30]에 의해 제안된 방법은 학습보다는 플랜트의 특성을 고려하여 설계하는 간접 제어 설계

구조(indirect control design structure)를 채택하였다. 이 구조는 지연시간(delay time), 최대 오버슈트(peak overshoot), 정착시간(settling time) 등 플랜트의 폐 루우프 응답을 특징짓는 특징(feature)들을 뽑아내고, 이 특징들을 입력으로하는 적응 공명 이론 분류기(Adaptive Resonance Theory classifier)에 보내서 플랜트의 파라미터를 알아내도록 한다. 플랜트의 파라미터가 추정되면, 원하는 폐 루우프의 응답특성과 함께 고전적인 극배치(pole placement) 기법을 이용하여 선형제어기의 이득을 구한다. 이렇게 설계된 제어 정보를 그림 7에서 보듯이 신경회로망의 일종인 연상 기억 모델에 저장하여 제어 시스템을 구성한다. 즉, 원하는 폐 루우프 응답을 나타내는 특징들과 시스템의 폐 루우프 응답을 나타내는 특징들이 비교되어 그 오차가 연상 기억 모델에 입력된다. 그러면, 신경회로망(associative memory)은 오차 입력에 대해서, 전체 시스템의 출력이 원하는 출력과 같아지도록 선형 되먹임 제어기의 파라미터 변화량 Δk 를 출력한다. 비슷한 접근법이 Ruano[31]에 의해서도 제안되었다. 이외에도 입출력 데이터만을 가지고 PID 파라미터를 온라인으로 찾아내는 방법이 연구되었으며[32], 폐 루우프 시스템의 응답을 나타내는 특징을 추출하는 방법도 Lawrence와 Harris[33]에 의해 독립적으로 제안되었다. 이러한 제어기법을 사용함에 따라 PID 제어의 견실성이 증가하였으며, 이 분야에 대한 연구가 계속되리라 전망한다.

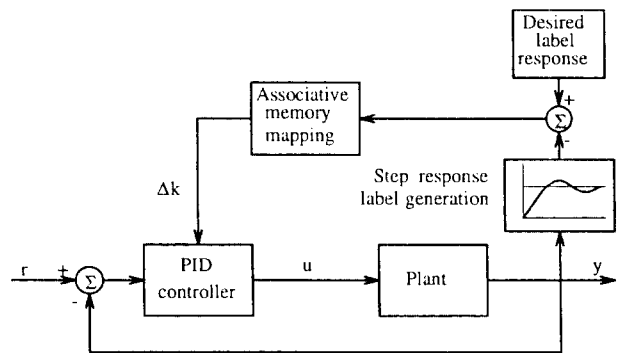


그림 7. 자동조정 되먹임 뉴로제어 구조.

4. 뉴로제어용 신경회로망 모델

McCulloch-Pitts 신경모델[34] 이후 신경회로망 모델에 관해서 많은 연구가 이루어져 왔고, 그에 따라 많은 신경회로망 모델이 제안되었다. 대표적인 신경회로망으로는 시그모이드 활성화 함수를 갖는 다층 신경회로망(multilayer neural network)[35, 16, 36]과 그의 변형 형태인 동적신경회로망(dynamic neural network)[11], 순환신경회로망(recurrent neural network)[37]등이 있으며, 국부 기저함

수의 선형조합에 기반한 방사형기저함수망(radial basis function network)[38,39,40], 확률신경회로망(probabilistic neural network)[41]등이 있다. 이 외에도 부분학습형 모듈 신경회로망[12,42], CMAC(cerebellar model articulation controller)[4]등 다양한 형태의 신경회로망이 제안되었다. 이 장에서는 지금까지 발표된 신경회로망 모델 중 비교적 널리 알려져 있고, 앞장에서 설명한 제어구조에 보편적으로 사용되는 것들을 중심으로 간략히 소개한다.

4.1 다층 신경회로망

다층 신경회로망(multilayered neural network)은 실제 생물체의 신경계를 근사 모델링한 것으로 대표적인 신경회로망이며 일명 다층인식자 (multilayer perceptron)라 불린다. 하나의 뉴런은 선형 함수를 나타낼 수 있으며, 많은 뉴런들을 (그림 8)에서와 같이 3층 이상의 망구조로 연결함으로써 임의의 비선형 함수를 나타낼 수 있음이 밝혀졌다[14,43]. 다층 신경회로망은 시스템 식별 및 제어, 패턴 인식 등에 많이 사용되나 학습속도가 느리고, 국부 최소점에 잘 빠지며, 적절한 망의 규모 결정이 용이하지 않는 등의 단점을 가지고 있다. 이를 해결하고자 하는 연구가 많이 진행되어 왔으나 실시간 적용 측면에서는 아직도 많은 문제를 가지고 있다. 그럼에도 실제 생물체 신경계를 모델링하였다는 점과 성능이 망규모에 민감하지 않고 적은 규모로도 구현할 수 있기 때문에 여전히 관심을 끌고 있다. 특히, 각 뉴런에서 활성화 함수를 제외하면 선형모델이라는 측면도 시스템 모델링등에는 흥미를 끄는 부분이다. 또한 뉴런 출력이 다시 뉴런입력으로 순환되는 순환신경회로망 (recurrent neural network)[37]도 동역학 시스템 모델링에 많이 활용되고 있다.

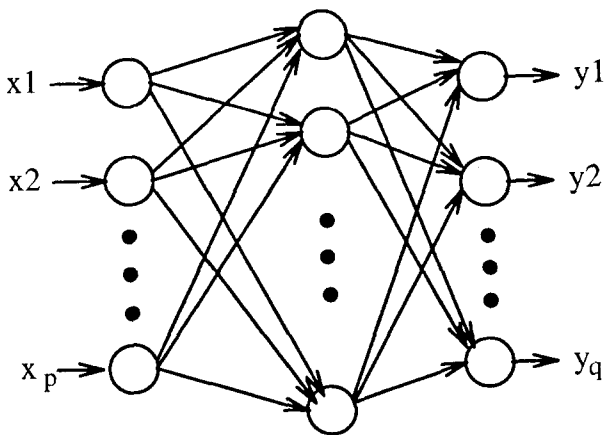


그림 8. 다층 신경회로망.

4.2 동적 신경회로망

앞먹임 다층 신경회로망이 임의의 비선형 함수를 주어진 오차안에서 근사화할 수 있어서, 이것으로 동역학 시스템을

모델링하는데 많은 연구가 있었다. 그러나, 출력의 되먹임이 없는 순수 앞먹임 신경망은 임의의 동역학을 모델링하는 데는 제약이 있음이 알려졌다[44]. 예를 들어 다음과 같은 IIR 시스템을 FIR 시스템으로 표현하려면,

$$\frac{1}{1-z^{-1}} = 1 + z^{-1} + z^{-2} + \dots + z^{-n} \quad n \rightarrow \infty, \quad |z| < 1 \quad (8)$$

와 같이 된다. 이 경우 순수 입력 시퀀스만 가지고 표현할 수 있지만, 차수가 많이 길어지게 된다. 따라서 되먹임이 있는 구조에 비해서 많은 계산량을 필요로 할 뿐더러, 불안정한 시스템의 경우 ($|z| > 1$)는 구현이 불가능하다. 그러나, 앞먹임 신경회로망에 시스템의 과거 출력을 입력으로 가하면 IIR 시스템 구조를 그대로 구현할 수 있다. 이 경우 순환신경회로망의 일종으로 분류되기도 한다. 실제로 앞먹임 다층신경회로망을 이용하여 동역학 시스템을 구현하는 경우는 이러한 형태로 실현된다.

동적신경회로망 (dynamic neural networks)은 그림 9에서와 같이 각 뉴런의 가중치를 동적 필터로 대체시킴으로써 순수 시스템 입력만으로 임의의 동역학 모델을 구현할 수 있도록 한 모델이다[11]. 이는 되먹임을 구조적으로 뉴런들 사이에 걸어주어서 신경회로망이 내부적으로 동역학적 특성을 갖도록 하는 것이다. 그림 9에서 가중치를 대신하는 부분의 전달함수는

$$\frac{C + Dz^{-1} + Ez^{-2}}{1 - Az^{-1} - Bz^{-2}} \quad (9)$$

이 된다. 만일 $C=w$, $A=B=D=E=0$, $g=1$ 이면 앞먹임 신경회로망과 같은 구조가 됨을 알 수 있다. 동적신경회로망은 복잡성 때문에 일반적인 이론이 제안되지 못했으나 동역학 시스템을 더 다양하게 모델링할 수 있다는 점에서 관심의 대상이 되고 있다.

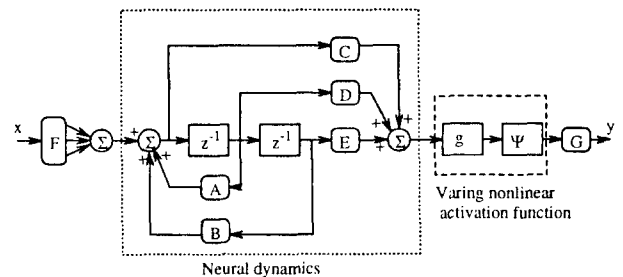


그림 9. 동적 뉴런의 구조.

4.3 CMAC : Cerebellar Model Articulation Controller

CMAC은 운동근육의 제어에 관계된 뇌의 한 부분인 소뇌를 모방한 모델이다. 소뇌에 전기적인 충격을 주어도 운

동근육의 동작에 거의 영향을 미치지 않기 때문에, 소뇌는 뇌의 조용한 부분이라 불린다. 그러나, 소뇌를 없애면 운동근육이 아주 비정상적으로 움직인다. 소뇌는 신체의 모든 부분으로부터 적절한 동작을 위해, 필요한 현재의 육체상태에 관한 정보를 다양한 경로를 통해서 받아들인다. 소뇌는 운동의 속도와 목적점에 도달하기 위한 반응 시간을 자동적으로 계산한 다음, 적절한 신호를 근육에 전달하여 근육이 움직이도록 한다. 이런 소뇌의 능력에 착안점을 두고 Albus [4]는 CMAC라 불리는 새로운 구조를 제안했다. CMAC은 각 뉴런들의 특성들과 각각의 뉴런들 사이의 구조적 특성을 모델링하지 않고서도, 뇌의 기능을 모방할 수 있다는 가정에서 출발하였다.

이 신경회로망의 구조가 그림 10에 나와 있다. CMAC은 두 개의 분리된 사상(mapping)으로 나눌 수 있다. 하나는 입력공간에서 기저함수(basis function)로 가는 사상이며, 이 기저함수공간은 일반적으로 입력공간보다 더 큰 차수를 갖는다. 비슷한 입력은 기저함수공간의 비슷한 영역으로 연결되고, 차이가 많이 나는 입력은 완전히 다른 영역으로 연결된다. 설계자는 기저함수(basis functions)를 생성하는 은닉층에서 주어진 입력에 대해 0이 아닌 출력값을 갖는 노드 수, 즉 일반화 매개변수(generalization parameter)를 결정해 주어야 한다. 이 일반화 매개변수가 출력에 영향을 미치는 내부 영역의 크기를 결정하게 된다. 초기의 기저함수는 이진 출력을 발생하는 함수를 사용하였으나 나중에는 방사형 기저함수(radial basis function) 형태도 사용하였다. 소뇌의 기능과 연관지어 설명하면, 감각세포에서 입력 벡터 $X = [x_1, x_2, \dots, x_n]^T$ 가 감지되면, 입력이 연상세포로 전달되어 기저함수를 생성한다. 기저함수 A중 활성화된 기저함수 값 A^* 이 응답세포에 의해 가중합(weighted sum)되어 출력된다. A^* 에 관계한 가중치를 갱신함으로써 주어진 입력 벡터 X에 대해 임의의 원하는 출력벡터 Y가 나오도록 학습시킬 수 있다. CMAC에서는 입력공간에서 한 점의 함수값은 다수의 기저함수 A^* 의 가중합으로 얻어지므로 함수값들은 분산방법에 의해 저장된다. 또한 전체적으로 많은 수의 가중치가 있을 때에도, 활성화된 기저함수와 관련된 가중치만 학습하므로 계산량이 적다. 따라서 다른 신경망에 비해 훨씬 적은 시간에 학습이 되고 이것은 실시간 제어에 많은 이점을 준다. 그러나 CMAC은 뉴런들 사이에 보다 많은 연결을 필요로 하는 단점도 있다. Ersü와 Tolle에 의해 CMAC은 비선형 제어문제에 적용되었고[6], 이후 실시간 로봇제어, 패턴인식, 신호처리등 응용분야에 많이 활용되고 있다.

4.4 방사형 기저함수망

방사형 기저함수망(radial basis function network)은 가우시안(Gaussian) 함수 형태의 방사형 기저 함수들의 유한개 선형결합 형태로 표현된 신경회로망이다[38]. 위의

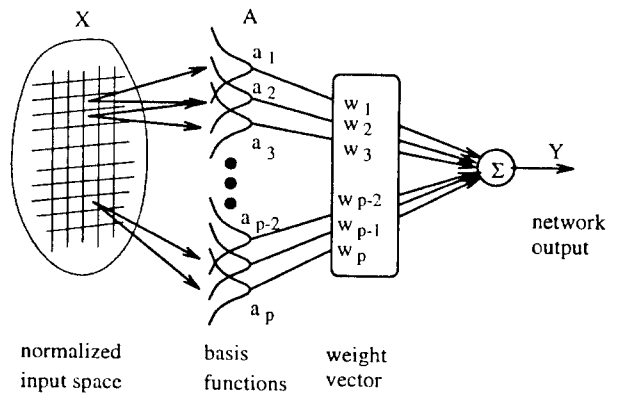


그림 10. Cerebellar model articulation controller(CMAC) 구조.

CMAC에서 연상세포층의 기저함수를 방사형 기저함수를 사용하면 개념상으로는 비슷하게 된다. 그러나 방사형 기저함수망은 함수 근사이론(function approximation theory)으로부터 유래되어 설계방법, 학습방법 등이 CMAC과 상이하다. 방사형 기저함수망을 수식으로 표현하면 다음과 같다.

$$f(x) = \sum_{i=1}^N w_i \phi_i(\|x - t_i\|), \quad (10)$$

$$\phi_i(r) = \exp\left(-\frac{r^2}{2\sigma^2}\right), \quad \sigma > 0 \quad (11)$$

이때, $\{\phi_i(x) \mid i = 1, 2, \dots, M\}$ 는 선형 독립인 방사형 기저함수들의 집합이고, t_i 는 각 기저함수의 중심점이다. 이 신경회로망의 구조가 그림 11에 나와 있다.

이 신경회로망을 학습하는 방법은 다양하게 제시되어 있으나, 크게 방사형 기저함수를 생성하는 은닉층 구성 방법과 이들의 가중합을 위한 가중치를 학습하는 방법으로 나누어진다. 은닉층 생성방법은 주로 자기 구성(self-organizing)방법을 사용하는데 ART(Adaptive Resonance Theory)[45], SOFM(Self-organizing Feature Map)[46] 등을 사용한다. 이들은 입력공간을 패턴의 거리를 척도로 군집화(clustering)한다는 점에서는 같지만, 알고리즘이 조금씩 다르다. 이러한 군집화 알고리즘은 주로 패턴인식 문제에 응용되었으나 함수 근사 문제도 입력의 패턴을 인식하여 그에 대응하는 함수값을 생성한다는 점에서 같은 문제로 이해할 수 있다. 한편, 출력층 노드가 원하는 함수값을 출력하도록 가중치를 학습하는 데는 지도학습 알고리즘, 즉 경사감 소법(gradient descent), 순환 최소자승법(recursive least squares) 및 이들의 변형 알고리즘 등이 사용된다. 또한 출력오차를 고려하여 은닉층 노드를 자동생성하면서 원하는 오차를 갖도록 학습하는 알고리즘을 제시한 연구도 있다[40]. 또한 입력 패턴마다 가우시안 기저함수를 생성시키면서 패턴 분류에 응용된 확률 신경회로망(probabilistic neural network)[41]도 같은 종류의 신경회로망으로 분류

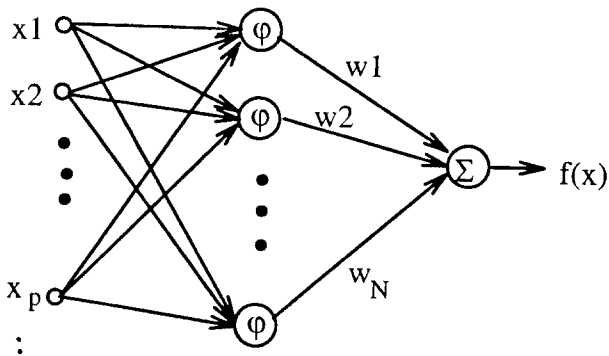


그림 11. 방사형 기저함수망 구조.

된다.

4.5 모듈형 신경회로망

시그모이드 형의 다층신경망은 학습속도가 느리고, 국부 최소점 문제로 학습에 실패하는 등의 단점 때문에 동역학 시스템에 실시간으로 응용하는데 어려움이 있었다. 학습속도를 높이고 추가학습을 용이하게 하기 위해 모듈형 신경회로망이 연구되었다. 뉴런이 선형 사상을 나타낸다는 점을 감안하면 입력공간을 국부영역으로 분할하고 이 국부영역에서 선형 사상을 나타내도록 하는 것이 기본 개념이다. 이것은 Jacobs와 Jordan[12]에 의해 처음으로 제시되었다. 이 모듈형 신경회로망은 두개의 망으로 구성되는데 하나는 입력공간을 계층적으로 분할하는 역할을 하고, 다른 하나는 분할된 입력공간에 대해 선형 함수를 나타낸다. 분할망의 뉴런은 입력공간을 선형 평면으로 양분하는 기능을 하고, 선형 근사가 불충분한 영역은 계속하여 양분해 나가는 형태이다. 한편 입력공간을 겹치지 않는 초삼각형(hypertriangular)으로 분할하고, 연속된 구분선형근사(piecewise linear approximation) 함수를 형성하는 구성적 신경회로망(constructive neural network)도 제시되었다[47]. 또한 입력공간을 분할하는 기능을 방사형 기저함수가 수행하는 구조로서 ART, SOFM 등의 기법을 써서 빠르게 분할하고, 각 국부영역에 대해 선형 학습을 수행함으로써 학습속도를 빠르게 할 수 있는 부분선형학습신경회로망(piecewise linearly trained network : PLTN)도 제시되었다[42]. 이외에도 함수의 형태에 따라 시그모이드형 신경회로망과 방사형 기저함수 형태의 신경회로망을 단순히 조합하여 학습하는 알고리즘도 발표된 바 있다.

5. 뉴로제어의 예

본 장에서는 온라인 상황에서 적응학습기능을 갖는 뉴로제어의 한 예를 시뮬레이션을 통하여 보여 주고자 한다. 대상 시스템은 다음의 상대차수가 $d=3$ 인 비선형 시스템이다.

$$y(k+3) = \frac{y(k+2)+y(k+1)}{1+y^2(k+2)+y^2(k+1)} + d(k) + (1+y^2(k+2))u(k), \quad (12)$$

여기서 구조적 변동 $d(k)$ 는 다음과 같다.

$$d(k) = \begin{cases} 0 & 1 \leq k < 2100 \\ \frac{0.5(y(k+2)y(k+1))}{1+y^2(k+2)+y^2(k+1)} & k \geq 2100 \end{cases}$$

이 시스템은 3.1절의 가정 1, 2를 모두 만족하는 시스템이다. 이 시스템의 동역학 모델을 모른다고 가정하면, 이 시스템을 2장에서 설명한대로 변형하면 다음의 일반적인 형태로 표시할 수 있다.

$$\begin{aligned} y(k+3) &= g(y(k-1), \dots, y(k-2), \\ &\quad u(k), u(k-1), \dots, u(k-3)) \\ &= g(x(k), u(k)), \end{aligned} \quad (13)$$

여기서 상태변수 $x(k)$ 는 $x(k)^T := [y(k-1), y(k-2), u(k-1), u(k-2), u(k-3)]$ 이다.

대상 시스템 (13)을 식별하기 위해서, 앞서 언급된 PLTN(piecewise linearly trained network)[42]을 사용한다. PLTN은 범용 근사자이므로 (4)에서 함수 $g(\cdot)$ 를 임의로 근사하는 함수 $\hat{g}(\cdot)$ 를 나타내는 PLTN을 학습할 수 있다. PLTN의 특성상 함수 $\hat{g}(\cdot)$ 로부터 역함수 $\hat{g}^{-1}(\cdot)$ 는 직접 계산이 가능하며[48], 따라서 제어 입력 $\hat{u}(k)$ 는

$$\hat{u}(k) = \hat{g}^{-1}(y(k+3), x(k)) \quad (14)$$

로 계산된다. 적응학습 제어기의 전체적인 구조가 그림 12에 나와 있으며, 시스템 식별 회로망(system identification network)의 학습과정과 이로 부터 제어계산 회로망(control computation network)의 매개변수를 갱신하는 과정이 온라인 상황에서 매 샘플링 구간 마다 반복된다.

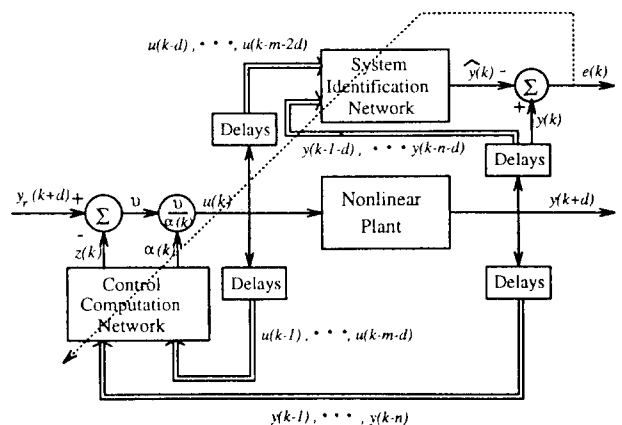


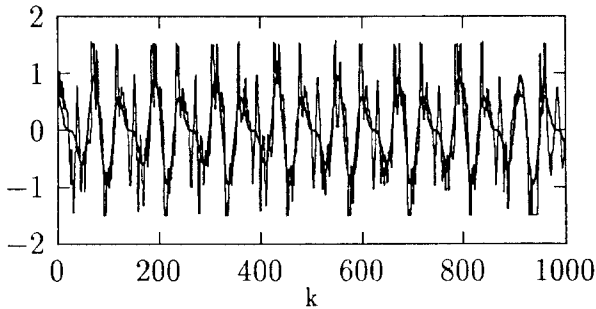
그림 12. 온라인 적응학습 제어 시스템의 전체적 구조.

시뮬레이션 시 동작점의 변동을 유발하기 위해 기준 출력

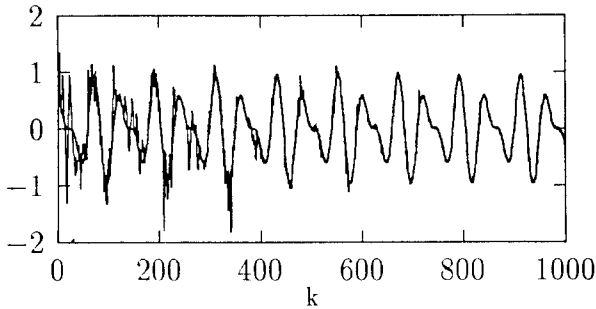
을 다음과 같이 주었다.

$$y_r(k) = 0.6 \sin(\pi k/30) + 0.4 \cos(\pi k/20) \quad (15)$$

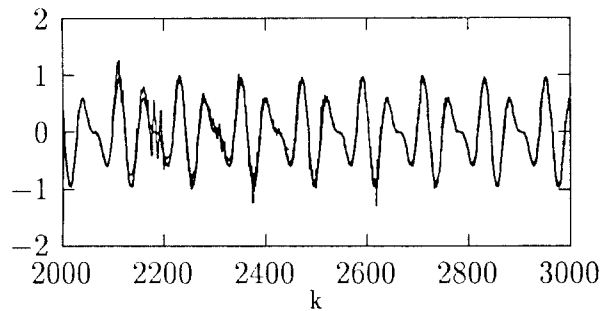
식별 초기상태에서 안정성을 위해 제어 입력은 $|u(k) - u(k-1)| \leq 0.2$ 로 제한하였다. 학습 알고리즘인 재귀 최소자승 알고리즘[49]에서 망각요소(forgetting factor)는 0.9로 놓았다. 위 시스템에 대한 온라인 적응 학습제어의 모의실험 결과가 그림 13에 나와 있다. 그림 13(a)는 하나의 은닉 유닛이 있을 때, 즉 선형모델 기반 적응제어 (dead beat adaptive control) 방법의 결과를 보여 준다. 선형모델 기반 적응제어는, 그림에서 보듯이 동작점이 변할 때마다 적응과도기를 거쳐야 한다. 즉, 과거에 나온 상황에 대해서도 똑



(a)



(b)



(c)

그림 13. 실시간 적응 학습제어의 결과; (a) 선형 모델 기반 적응제어(유닛수가 1개일 때), (b) $\gamma = 0.02$ (90개 유닛이 생성)일 때의 적응 결과, (c) disturbance $d(k)$ 가 $k = 2100$ 에서 입력되었을 때 결과(4개 유닛 추가 생성).

같이 처음부터 적응과정을 거쳐야 하므로 성능이 나쁘다. 그림 13(b)에는 90개의 유닛이 생긴 적응 학습제어의 결과이다. (a)와는 달리 초기 적응기가 지난 뒤에 기준신호를 잘 따라감을 알 수 있다. 즉, 과거에 학습된 상황에 대한 정보가 새로운 상황에 대한 적응과정에서 파괴되지 않는다. 시스템의 구조적 변동에 대한 성능을 알아보기 위해 $k = 2100$ 부터 (13)과 같은 변동을 주었다. 이에 대한 결과가 그림 13(c)에 나와 있다. 그림에서 보듯이 새로운 상황에 대한 적응과정을 거친 다음에 기준신호를 잘 따라감을 알 수 있다.

본 예제를 통하여 살펴 보았듯이 온라인 적응학습제어 기법은 선형 적응제어 기법과 달리 과거의 학습된 상황에 대한 정보를 저장할 수 있는 학습능력을 갖고 있어, 과거 학습한 상황의 재발생시 바로 적응할 수 있음을 알 수 있다. 그러나 적응 과도기간에서의 안정성등은 일반적으로 보장되지 않으며 이를 해결하기 위해서는 오프라인 상황에서의 초기 학습을 시켜야 할 필요가 있다. 또한 온라인 상황에서 안정도를 보장하는 초기 조건을 찾고 이를 만족하도록 설계하는 체계적인 방법이 요구된다.

6. 맺음말

지금까지 뉴로제어의 배경 및 필요성, 일반적인 뉴로제어의 구조, 뉴로제어용 신경회로망 모델 등을 개괄적으로 살펴보고 이를 이용한 적응학습제어의 한 예를 시뮬레이션을 통하여 예시했다. 뉴로제어와 관련된 연구는 크게 두가지로 생각할 수 있다. 하나는 제어기로 사용되는 신경회로망 자체에 대한 연구이고, 다른 하나는 전체 뉴로제어 시스템에 관한 연구이다. 전자의 경우는 제어에 적합한 신경회로망의 구조, 학습방법 등에 대한 연구로서 동역학 시스템을 얼마나 잘 모델링할 수 있는가, 실시간 적응을 위해 얼마나 빠른 수렴성을 갖는가 등에 관심이 주어진다. 후자의 경우는 전체 제어시스템의 관점에서 가제어성, 수렴성, 안정성 등을 고려해야 하며, 이러한 제어 시스템의 기본 조건을 만족하는 제어시스템의 설계 기법 연구에 역점이 주어져야 할 것이다. 신경회로망 연구는 최적화 기법등의 공학적인 기법뿐 아니라, 신경생리학적인 현상도 연계되어 이루어져야 할 것이고, 뉴로제어의 시스템적인 연구는 기존의 제어 이론과 접목하여 체계적으로 정립할 필요가 있다.

현재 까지 제시된 대부분의 뉴로제어기법을 적용하기 위해서는 첫째, 대상 플랜트의 역 동역학(inverse dynamics)이 존재해야 하고, 둘째, 대상 플랜트가 비최소위상(inimum phase)이어야 한다. 이 조건을 만족하지 못하는 경우는 오프라인 학습제어인 경우에도 제어 목적을 달성할 수 없거나 내부적으로 불안정해 질 수 있음을 지적한 바 있다. 따라서 비 최소위상 플랜트에 대한 뉴로제어는 새로운

각도에서 생각해야 된다. 또한 적응학습제어인 경우는 온라인 상황에서 제어정보의 실시간 학습이 보장되어야 하고, 온라인 적응학습시 안정성이 보장되도록 설계되어야 한다. 지금까지는 이에대한 연구가 미미하고, 일반적으로 학습제어기는 전역 안정성은 보장되지 않는다. 따라서 온라인 상황에서 안정도를 보장하는 초기 조건을 제시하는 것이 현실적이다. 이 조건을 만족하는 초기 제어 시스템을 오프라인으로 학습하고, 온라인 상황에서는 안정도를 보장하는 범위 내에서 적응학습제어를 하도록 하는 것이 바람직할 것이다.

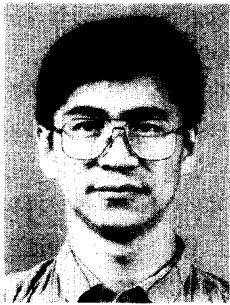
끝으로, 첫번째 저자가 신경회로망 및 제어관련 지식을 습득하고, 연구하는 과정에서 방향제시와 세심한 조언을 아끼지 않으신 서울대학교 최종호 교수님께 감사드립니다.

참 고 문 헌

1. B. Widrow. IEEE Int. Symp. Circuits and Systems., pp. 351-357, 1987.
2. R.A. Chambers. D. Michie. Machine Intelligence 2, pp. 137-152, 1968.
3. R.S. Sutton a.G. Barto. IEEE Trans. Sys. Man and Cyb., 13(5) : 834-846, 1983.
4. J.S. Ablus. Trans. of the ASME, J. of Dynamic Systems, Measurement, and Control, 97 : 220-227, 1975.
5. J.S. Albus. Trans. of the ASME, J of Dynamic Systems, Measurement, and Control, 97 : 228-233, 1975
6. E. Ersü H. Tolle. Neurocontrol : Learning Control Systems Inspired by Neuronal Architectures and Human Problem Solving. Springer-Verlag, 1992.
7. L.G. Kraft W.T. Miller, F.H. Glanz. IEEE trans. on Robot. and Auto., 6(1) : 1-9, 1990.
8. P.J. Werbos W.T. Miller, R.S. Sutton. Neural Networks for Control. MIT Press, 1990.
9. D.E. Rumelhart and J.L. McClelland. Parallel Distributed Processing. MIT Press, Cambridge, 1986.
10. S.A. Billings S. Chen. In C.J. Harris, editor, Advances in Intelligent Control. Taylor and Francis, London, 1994.
11. D.H. Rao M.M. Gupta. Journal of Intelligent and Fuzzy Systems, 1(1) : 73-92, 1993.
12. M.I. Jordan R.A. Jacobs. In D.J. Touretzky R.P.Lippmann, J.E.Moody, editor, Advances in Neural Information Processing Systems 3, pp. 767-773. San Mateo, 1991.
13. F.-C. Chen and H.K.Khalil. IEEE trans. on Automatic Control., 40(5) : 791-801, 1995.
14. K. Hornik. Neural Networks, 4 : 251-257, 1991.
15. B. Wittenmark K.J. Astrom. Adaptive Control. Addison Wesley, 1989.
16. K.S. Narendra and K.P. parthasarathy. IEEE Trans. on Neural Networks, 1(1) : 4-27, Mar. 1990.
17. M.T. Tham A.J.Morris G.A. Montague, M.J. Willis. IEE 2nd Int. Conf. Artificial Neural Networks, pp. 119-123, 1991.
18. T.J. McAvoy J. Saint-Donat, N. Bhat. In Taylor C.J. Harris and Francis, editors, Advances in Intelligent Control. London, 1994.
19. D. Sbarbaro-Hofer K.J. Hunt. IEE Proc. D, 138 (5) : 431-438, 1991.
20. D. Sbarbaro-Hofer K.J. Hunt. In Warwick K. et al., editor, Neural Networks for Control and Systems, pp. 94-122. IEE Cont. Eng. Series 46, 1992.
21. Kraft and Campagna. IEEE Control Sys. Mag., pp. 36-43, April 1990.
22. W.T. Miller. IEEE J. Robot. and Auto., 3(2) : 157-165, 1987.
23. T. Setoyama R. Suzuki H. Miyamoto, M. Kawato. Neural Networks, 1 : 251-265m 1988.
24. P. Khedkar H.R. Berenji, IEEE Trans. Neural Networks, 3(5) : 724-740, 1992.
25. W.L. Baker P.J. Millington. AIAA Guid. Nav. and Cont. Conf., 2 : 1120-1128, 1990.
26. J.S. Collins D.P. Precino. IEEE International Joint Conference on Neural Networks, 2 : 417-420, 1990.
27. J.K. Peterson R.O. Shelton. Simulation, 58(5) : 319-326, 1992.
28. S.P. Singh A.G. Barto, S.J. Bradtke. Learning to Act using Real-Time Dynamic Programming. Dept. of Computer Science, University of Massachusetts, 1993.
29. R.J. Williams R.S. Sutton, A.G. Barto. IEEE Cont. Sys. May., pp. 19-22, April 1992.
30. A.Guez S.S. Kumar. Neural Networks, 4(3) : 319-335, 1991.
31. D.I. Jones A.E.B. Ruano, P.J. Flemming. IEE Proc. D, 139(3) : 279-285, 1992.
32. T. Hagglund K.J. Astrom. Automatic Tuning of PID Controllers. Instrument society of america, 1988.
33. C.J. Harris A. Lawrence. In Page G.F et al., editor, Application of Neural Networks to Modelling and Control. Chapman and Hall, 1992.

34. W.S. McCulloch and W. Pitts. Bull. Math. Biophys., 21 : 11-133, 1943.
35. A.U. Levin and K.S. Narendra. IEEE Trans. on Neural Networks, 4(2) : 192-206, Mar. 1993.
36. T. Yamada and T. Yabuta. IEEE Trans. on Neural Networks, 3(4) : 595-601, July 1992.
37. Amir F. Atiya Alexander G. Parlos, Kil t. Chong. IEEE trans. on Neural Networks, 5(2), 1994.
38. T. Poggio and F. Girosi. Proceedings of the IEEE, 78 (9) : 1481-1497, Sep. 1990.
39. J. Moody and C.J. Darken. Neural Computation 1, pp. 281-283, 1989.
40. S. Lee and R.M. Kil. Neural Networks, 4(2) : 207-224, 1991.
41. D.F. Specht. Neural Networks, 3 : 109-118, 1990.
42. J. Y. Choi and R. M. Kill. IEEE International Conference on Neural Networks(ICNN), pp. 1359-1363, June 1994.
43. P. Koiran. Int'l Joint Conf. on Neural Networks, pp. 1.883-I.886, June 1992.
44. J.J. Hopfield. IEEE Expert, pp. 3-6, April 1990.
45. Y-H. Pao. Adaptive pattern Recognition and Neural Networks. Addison-Wesley Publishing Company, Inc., 1989.
46. T. Kohonen. Self Organization and Associative Memory. Springer-Verlag, Berlin, 1984.
47. C.-H. Choi and J.Y. Choi. IEEE Trans. On Neural Networks, 5(6) : 936-944, Nov. 1994.
48. J.Y. Choi H.J. Park. KACC Domestic session, II : 808-811, Oct. 1995.
49. G.C. Goodwin and R.L. payne. Dynamic System Identification : Experiment Design and Data Analysis. Academic Press, 1977.

저 자 소 개



최 진 영

1982년, 1984년, 1993년 서울대학교 공과대학 제어계측공학과에서 학사, 석사 및 박사학위 취득.
 1984년부터 1994년까지 한국전자통신연구소에서 연구원 및 선임연구원으로 TDX 개발 및 정보기술기초연구 참여. 현재 서울대학교 제어계측공학과 전임강사.
 관심분야는 Neural Networks, Adaptive and Learning System, Adaptive Control of Autopilot, Detection and Recognition System.
 (151-742) 서울 관악구 신림동 산 56-1
 TEL)(02)880-8372 / FAX)(02)885-6620

박 현 주

1995년 2월 서울대학교 공과대학 제어계측공학과 졸업 (학사).
 현재 서울대학교 공과대학 제어계측공학과 석사과정.
 관심분야는 Neural Networks, Adaptive and Learning Control.
 (151-742) 서울 관악구 신림동 산 56-1