

신경회로 (Neural Network)의 로봇틱스 및 산업 자동화 응용

吳世泳

浦項工科大学 電子電氣工學科

제6세대 컴퓨터로 불리는 신경컴퓨터는 학습과 병렬처리에 의해 인간의 지능을 모방한다. 따라서 지능과 빠른 계산을 요하는 여러 분야에 응용되고 있으며 실제 로봇의 제어나 sensor에 의거한 제어에 응용하여 좋은 결과를 내고 있다. 신경회로의 로봇나 공정제어(process control) 응용은 학술적인 측면에서는 복잡한 비선형 시스템의 지능제어(intelligent control) 연구이며 산업적 측면에서 보면 산업 자동화라는 막대한 시장을 뒤로 하고 있어 우리나라도 활발한 연구를 절실히 필요로 하고 있다.

본 해설에서는 신경회로를 간단히 소개한 후 로봇 제어 응용을 다루기로 한다. 신경회로의 응용분야중 보고된 결과가 비교적 적은 제어분야를 소개함으로써 독자들에게 연구 자료들을 제공하고 또한 흩어져 있는 신경회로의 제어응용 논문들을 분류 통일함으로써 이 분야를 조감할 수 있게 한다. 또한 로봇을 하나의 복잡하고 비선형적 plant로 보았을 때 로봇의 신경제어는 곧 산업공정의 신경제어에도 그대로 응용되리라 믿는다. 신경제어는 plant의 모델없어도 학습에 의하여 고속 정확한 제어가 가능하고 또 plant 특성변화에 잘 적응하며 병렬성으로 인하여 실시간 제어도 가능하다는 점에서 무한한 잠재력이 있으나 전세계적인 연구는 아직도 크게 미흡한 편이다. 더욱 많은 연구가 절실히 필요하다고 본다.

I. 서론

최근 국내외적으로 신경회로에 대한 관심은 소규모 혁명이라 불릴 정도로 가히 폭발적이라고 할 수 있다. 이는 신경회로가 인간의 지능을 가장 자연스럽게 효과적으로 모방하고 또한 실제적 응용이 가능할

것이라는 기대 때문이다. 필자의 전망으로는 향후 20년간 신경회로는 인공지능(AI) 기술을 대체 또는 보강하여 패턴 인식과 신호처리, 최적화, 상업 및 공장 자동화등 실로 광범위하게 응용될 것이다.

신경회로는 병렬 분산처리(parallel distributed processing)원리에 의거하여 임의의 입출력 데이터 변환을 수행하므로 근본적으로 모든 계산을 대처할 수 있다. 단지 계산하는 방식이 종래의 컴퓨터와 정확한 알고리즘을 수행하는 것이 아니고 학습기록된 점들을 기준으로 보간(interpolation)하는 방식이 다를 뿐이다. 따라서 선형보다는 비선형 계산에서, 단순한 계산보다는 복잡한 계산에서 그 위력을 발휘한다.

신경회로의 응용분야중 제어분야는 음성이나 영상 인식 보다는 데이터량이 적고 접근하기가 용이하다. 그 이유는 대부분의 제어문제는 입출력이 명확히 정의되기 때문이다. 즉 제어목표와 plant 상태를 감지하는 센서값으로부터 actuator 명령을 발생하는 일종의 정보처리문제로 귀결되고 신경회로는 하나의 정보처리 모델이기 때문이다.

현재까지의 제어시스템은 그 주류가 지난 50년간에 걸쳐 연구한 자동제어 이론에 기초하고 있다. 간단한 시스템의 경우 수학적 모델에 의거하여 주로 선형제어를 이용한다. 그러나 복잡한 시스템의 경우 또는 기존 시스템을 미지의 다른 시스템과 연결하여 더 큰 시스템을 형성할 경우 종래의 제어 이론을 적용하기가 힘들게 된다. 산업 시스템에서는 여러 기기가 유기적으로 연결되어 아주 복잡하다. 무수한 센서들로부터 시스템의 현재상태를 파악하고 시시각각 적합한 제어신호를 계산하는 것은 종래의 제어이론으로는 매우 힘든 일이다. 이론적으로 제어장치를

실제하는 것도 어렵지만 할 수 있다라도 실제로 잘 동작할 지 의문이며 그 개발비 또한 엄청날 것이다. 신경컴퓨터는 이에 대해 신선한 해결책을 제시해 준다.

신경회로는 센서를 통해 들어오는 플랜트 상태나 환경에서 오는 feedback을 본 후 과거의 경험에 비추어 제어신호를 보내보는 지능제어를 한다. 즉, 종래의 제어가 단계적 프로그램 계산에 의거한다면 신경 회로방식은 훈련(training)을 거친 직관, 즉 패턴 인식에 의거하며 이는 생물계에서 대자연이 택한 방식이기도 하다. 따라서 이를 패턴인식 제어(pattern recognizing control)라 부를 수 있다. 이렇게 훈련된 지능제어 시스템은 실제 운용하며 그때 그때 불완전한 지식을 보강하기가 용이하고 프로그램을 거치지 않으므로 가격이 저렴하다. 더욱이 플랜트의 특성이 시간이 지남에 따라 바뀌어도 또 외부에서 교란이 들어와도 학습에 의해 적응하게 된다. 따라서 신경컴퓨터는 인간의 적응, 훈련, 학습기능을 모방하게 된다.

이 신경컴퓨터 기술을 산업계에 실제로 응용한 예로서는 미국의 AI WARE 회사가 개발한 신경컴퓨터 소프트웨어를 이용하여

- 1) 절삭공정 (grinding)에 이용하여 60%의 생산성 향상과 스크랩(버리는 재료)을 제거하여 연간 50만 불을 절약했고
- 2) 금형제조공정에서 불량품을 없애고 연간 30만 불 절약했으며
- 3) 화학제품 설계에서 최적의 원료배합물을 훈련 학습하여 설계비용을 절감했다.

이러한 지능제어의 효과는 경비절감뿐 아니라 더 중요한 품질향상을 하고, 재료의 낭비를 극소화하여 생산성과 직결된다.

본 해설에서는 신경 회로의 기초이론과 로봇트 제어 응용을 소개하고 이를 일반적인 공정제어에도 응용할 수 있음을 보이기도 한다. 마지막으로 신경제어 연구에서 풀어야 할 현안 문제를 다룬다.

II. 신경회로의 기초이론

그림1,2와 같이 인공 신경회로는 생물학적 신경회로를 간단한 수학을 이용하여 하드웨어를 구현한 것이다. 신경회로의 학습은 환경에서 오는 데이터로부터 뉴론과 뉴론을 연결하는 synapse의 연결강도(weight)를 조정하는 과정이다. 즉 학습전의 신경회

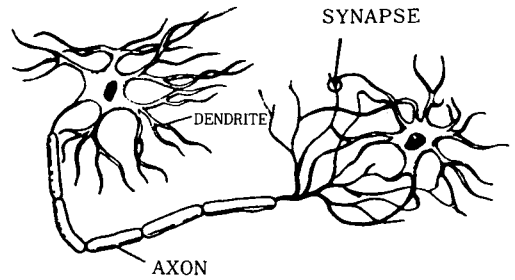


그림 1. 생물학적 신경회로

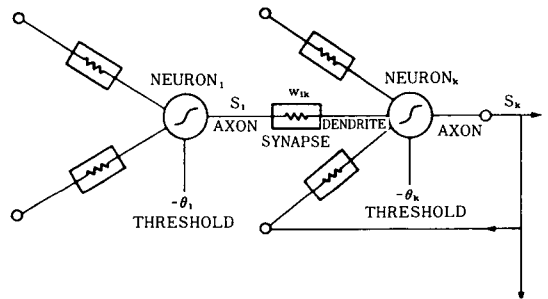


그림 2. 인공 신경회로

로는 아메바와 같은 무정형이며 학습은 원하는 입출력 데이터 관계를 만족시키도록 연결강도를 조정하여 자신을 변형시킨다고 볼 수 있다.

1. 신경회로의 분류

기능에 따라 크게 다음의 네가지로 분류된다.

- (1) 패턴인식형 (pattern recognition)
- (2) 비선형 변환형 (nonlinear mapping)
- (3) 연상형 (associative memory)
- (4) 최적화형 (optimization)

이중 패턴인식은 입력 데이터를 몇개의 class로 분류하는 것이고 문자인식, 영상인식, 음성인식 등 공장 자동화에 필요 불가결한 응용분야이다. 비선형 변환은 임의의 연속함수를 모델하는 것으로서 로봇트 제어, 공정제어, 신호처리 등에 사용된다. 연상형 회로는 군사용으로 물체 인식이나 잠음제거 등에 사용된다. 최적화형은 모든 공학분야에 응용된다.

또한 신경회로는 훈련방법에 따라 두 가지로 나눌 수 있다.

(1) Supervised 학습

하나의 입력과 원하는 출력이 훈련쌍을 이루어, 그 입력에 대한 원하는 출력과 실제 출력 값의 차이가 오차로 작용하여 다음 훈련에서는 오차가 줄어드는 방향으로 weight를 수정한다.

(2) Unsupervised 학습

훈련이 있기 전에는 출력을 예측할 수 없다. 어떠한 입력이 인가될 때 마다 그에 상응하는 출력 값을 나타내도록 'self organize' 한다. 생물학적인 관점에서는 타당성이 적으나, 실제적인 문제를 해결하는데 좋은 결과를 나타낸다.

2. Back Propagation(BP) 신경회로⁽¹⁾

Supervised 학습을 사용하는 신경회로중 로봇제어에 가장 많이 쓰이는 BP회로에 관하여 기술한다. BP 알고리즘은 그림3과 같은 다층구조의 신경회로를 훈련시키는데 사용한다. 2단 구조의 신경회로는 중간단 즉 hidden layer가 없는 경우로서 입력단에 들어온 데이터를 곧바로 출력단에 투영한다. 이러한 회로는 비슷한 입력은 비슷한 출력으로 투영(mapping)되어야 하는 성질을 갖고 있어서 임의의 입력 출력 관계를 만족시킬 수는 없다. 그러나 다층 신경회로를 사용하면 중간뉴론을 통하여 임의의 입력에 대해 원하는 출력값을 갖도록 할 수 있다. BP 회로의 학습과정은 다음과 같다.

어떤 입력이 입력단으로 부터 출력단으로 진행함에 따라 각단의 뉴론은 바로 앞단 뉴론의 출력에

weight를 곱하여 출력으로 내보내게 되는데 비선형 특성을 부여하기 위해서 활성화함수를 거치게 된다. 이 활성화함수는 단조증가, 미분가능이어야 하며 흔히 다음과 같은 형태의 것이 많이 사용된다.

$$F(\text{input}) = 1 / (1 + \exp(-\text{input}))$$

이렇게 만들어진 출력 값은 다시 weight가 곱하여져 다음단에 입력으로 인가된다. 이러한 과정이 최종 출력단에 도달할 때 까지 반복된다.

BP 알고리즘은 supervised 학습의 일종으로 최종단의 출력과 원하는 출력의 차이의 제곱의 합을 줄이는 방향으로 weight를 조절한다. 각단의 뉴론에서 나타난 오차는 회로를 역방향으로 진행하면서 중간단의 weight를 수정한다. 학습규칙(learning rule)은 다음과 같다.

1) Weight의 수정

원하는 출력과 현재의 출력과의 오차로부터 weight를 수정한다. 즉

$$\Delta W_{ij} = \eta \delta_j O_i$$

O_i : 앞단의 i번째 뉴론의 출력, δ_j : 역방향으로 전달되는 오차

W_{ij} : 앞단의 i번째 neuron과 출력단의 j번째 neuron 사이의 weight

η : 학습률(일반적으로 0.1임)

2) 최종 출력단의 오차계산

$$\delta_j = (T_j - O_j) F'(Net_j)$$

$$Net_j = \sum_i W_{ij} * O_i$$

T_j : 출력단의 j번째 뉴론에서의 목표 값

O_j : j번째 뉴론의 출력

3) 중간단의 오차계산

$$\delta_j = F'(Net_j) * \sum_k (\delta_k * W_{kj})$$

δ_k : 처리중인 단의 뒷단에서 전달되어온 오차

W_{kj} : 처리중인 단의 j번째 뉴론과 뒷단의 k번째 뉴론간의 weight

이러한 과정을 입력단에 도달할 때까지 반복한다.

III. 기존의 제어방식과 신경제어방식의 차이점

현재까지 산업용 로봇이나 실험용 로봇에서 약 30년간 연구한 제어방식은 크게 (1) Kinematic control (2) Dynamic control, (3) Adaptive control 등으로 분류할 수 있다. 그림4(a)는 가장 단순한 PID 제어방

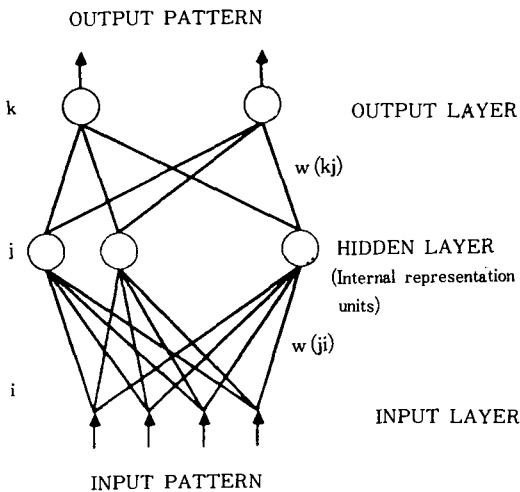
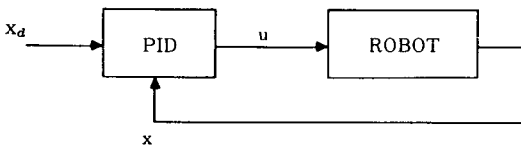
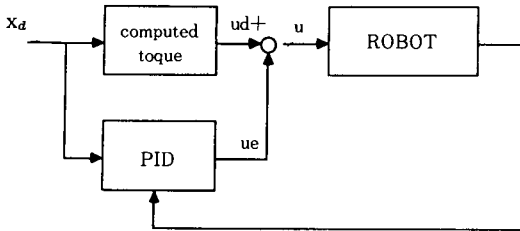


그림 3. 다층구조의 신경회로



(a) 간단한 PID-Kinematic 제어



(b) Dynamic 제어

그림 4. 고전적인 제어

식으로 로봇의 위치, 속도, 가속도등의 kinematic error에 의해 각 joint를 구동할 motor torque가 결정된다. 즉 dynamics를 고려 안하기 때문에 고속이나 정확한 운동에는 부적합하다. 그림4(b)는 dynamic control로써 질량, 관성등을 고려한 dynamic torque를 계산하여 모터에 가해준다. 빠르고 정확한 운동에 필수적이다. 이 두 방식은 모두 로봇의 kinematic, dynamic 파라미터가 필요하나 adaptive control은 파라미터가 필요없다. 그러나 공식유도가 복잡하고 광범위한 uncertainty에 robust 하지 않고 real-time 제어에 부적합하다.

이에 반해 신경제어는 종래의 제어이론과 전혀 다른 방향에서 참신한 접근을 한다. 신경회로제어의 장점을 요약하면

- (1) 제어될 플랜트나 그 환경 모델의 불필요-복잡한 시스템에 효과적임
- (2) Uncertainty나 플랜트-환경의 변화에 적응 가능-robustness and fault tolerance
- (3) 제어성능의 자체학습에 의한 항구적 개선
- (4) 신경회로의 병렬개산에 의한 고속 real-time 제어
- (5) 신경회로의 뉴런수에 무관한 steady-state로의 수렴속도 때문에 많은 수의 파라미터에 빨리 적응
- (6) 예시에 의해 배우므로 명확한 제어 공식이 불필요-heuristic rule 사용 가능
- (7) 신경회로내 정보의 분산 표현으로 인한 fault tolerance-일부 회로 소자가 고장나거나 불완전한

경우 또는 noisy sensor data에 대해서도 graceful degradation 가능

(8) 마찰, 비선형 torque 특성,샘플링 효과, 센서 잡음등 unmodeled effect의 자연적 해결

(9) Sensor data fusion이 자연스럽게 처리됨 등이다. 실제의 산업용 로봇은 실제적 이유로 인해 복잡한 dynamic 계산을 하지 않는다. 그러나 kinematic 계산과 경로보간 계산을 한다. Servo 제어로는 간단한 PID 제어가 주종을 이룬다.

IV. 신경제어 방식의 분류

1. 신경회로만으로 제어하는 경우

다음의 세 방식은 Psaltis⁽²⁾에 의한 것이다.

1) 간접 학습방법

그림 5와 같이 원하는 출력을 바꿔가며 플랜트를 움직이는 동안 신경회로를 역 플랜트(inverse plant)의 전달함수로서 훈련시킨다. 이와 같이 훈련되는 신경회로와 동일한 구조의 신경회로를 플랜트앞에 간접 제어기(controller)로 사용한다. 원하는 동작범위에서 훈련시킬 수 있는 장점이 있는 대신 |u-t|를 극소화한다고 해서 |d-y|가 극소가 되는 것이 아닌 단점이 있다.

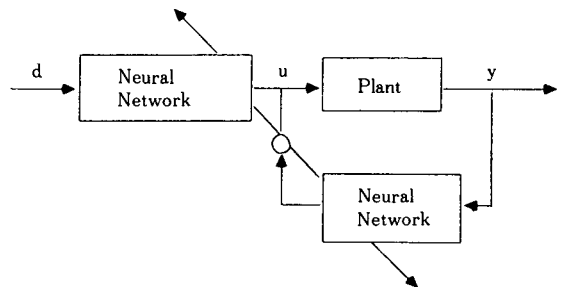


그림 5. 간접학습구조

2) 일반 학습방법

그림6에서는 플랜트의 제어 입력을 광범위하게 바꿔가며 그에 상응하는 출력을 신경회로의 입력으로 하여 훈련시킨후 이 신경회로를 제어기로 사용한다. 그러나 원하는 동작 범위에 국한되는 훈련이나 on-line 훈련을 시킬 수 없다는 단점이 있다.

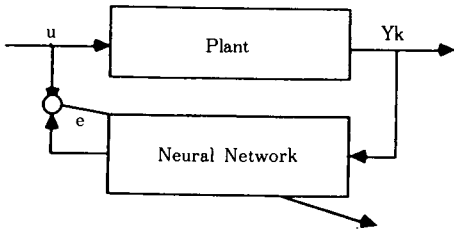


그림 6. 일반학습구조

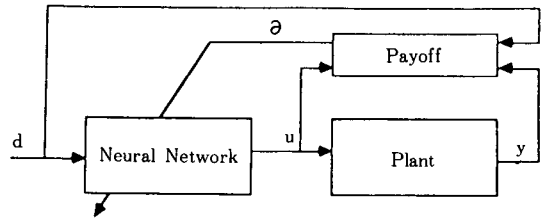


그림 8. 보상함수를 극대화하는 신경회로

3) 특정 학습 방법

그림7과 같이 원하는 입력을 바꿔가며, 신경회로 제어기와 플랜트를 거친 후 나오는 출력을 원하는 입력에 근접하도록 신경회로 제어기를 훈련시킨다. 원하는 동작 범위에서 훈련이 가능하고 on-line 훈련도 가능하다. 실제로는 전항의 일반 학습방법으로 광범위한 지역에서 훈련시킨후 특정 학습방법으로 fine-tuning하는 방법을 사용하는 것이 좋다.

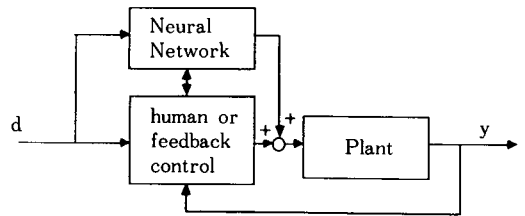


그림 9. Feedback-feedforward 제어 방식

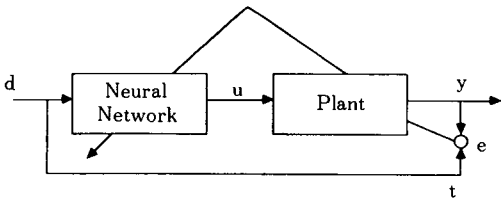


그림 7. 특정학습구조

4) Unsupervised 학습

Unsupervised learning을 사용한 것으로서는 그림8 처럼 신경회로가 플랜트를 제어하면서 그 성능을 보상 함수로 받아서 이를 극대화하는 방향으로 자신을 훈련시킨다. 이 방식은 Cart-Pole 시스템처럼 주어진 상태에 대하여 원하는 제어출력값을 모를때 매우 유용하다.

2. 종래의 간단한 선형제어와 병렬로 신경회로를 쓰는 경우 (Feedback-Feedforward 제어)

그림9와 같이 신경회로가 종래의 제어 또는 사람이 직접 제어하는 것을 보며 이를 학습한 후 이를 feedforward 제어기로 사용한다. 훈련 후의 실제 제어에 있어 feedback loop는 계속 작용할 수도 혹은 제거할 수도 있다.

3. 종래 제어구조의 일부를 신경회로로 훈련하는 방식

Guez⁽⁴⁾는 종래의 PD(proportional derivative) 제어의 gain을 grossberg 신경회로를 써서 동적으로 조정하며 제어하는 방법을 소개했다. 이 방법을 1축 manipulator의 제어에 응용했으나 선형제어의 틀을 벗어나지 못하는 단점이 있다. 한편 Elsley⁽⁵⁾는 종래의 inverse Jacobian 제어구조에서 신경회로로 inverse Jacobian을 계산하였다.

V. 신경회로의 로봇트 제어 응용들

필자가 신경회로의 제어응용에 관련하여 조사한 논문들은 거의 모두가 simulation이 용이한 간단한 선형 및 비선형 시스템과 Cart-Pole 문제, 로봇트의 kinematic control, dynamic control, 시각 제어, obstacle avoidance 등에의 응용이었다. 사용한 신경회로는 거의 대부분이 back propagation 회로이고 그 외에 Ritter가 Kohonen의 topology-conserving map을, Tsutsumi가 hopfield 회로, Guez가 grossberg 회로, Miller가 Albus의 CMAC을, Barto가 associative search network을 사용하였다.

1. Miller의 CMAC 제어 방식⁽⁶⁾

CMAC(cerebellar model articulation controller)은

Albus^[7]가 1975년에 발표한 뇌의 모델로서 다른 신경 회로구조와 판이한 구조를 가지고 있다. 그림10에서 보듯이 CMAC은 $S \rightarrow A' \rightarrow A \rightarrow f$ 의 mapping chain으로 되어 있다. 우선 입력S가 A' 테이블의 몇 개의 주소를 선택한 후, 테이블 크기를 줄이기 위해 random hash coding에 의해 축소된 테이블A의 몇 개의 weight를 선택한 후 이들을 합산하여 출력을 낸다. CMAC의 훈련은 로봇트를 움직이며 그때 그때 가해준 torque와 출력 response의 관계를 보며 A 테이블의 weight를 조정함으로써 이루어진다. 이 방식은 실제로 camera 데이터를 받아 general electric의 5축 P-5 로봇트로 컨베이어상의 물체를 추적하는데^[8] 응용되었다. 이는 신경 회로를 처음으로 현실적 문제에 적용한 좋은 예이다. 또 2축 로봇트의 dynamic control에도 사용되었다.

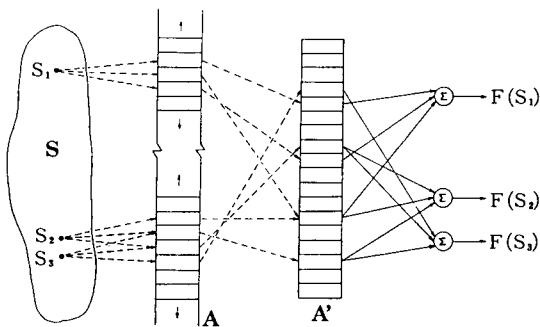


그림10. CMAC의 구조

2. Psaltis의 일반학습, 특수학습 제어방식

이미 2절에서 설명한 대표적 구조 유형으로서 2차원 극좌표에서 직교좌표로 변환시키는 mapping에 back propagation 회로를 사용하였다. 다른 많은 방법들이 이 유형에 해당된다. 실제로는 일반 학습과 특수학습을 병행하여 일반 학습으로는 global 특성을 배우고 특수 학습으로는 특정 지역에서 fine-tuning을 하도록 한다.

3. Widrow의 ADALINE 제어방식

Widrow는 60년에 발표한 ADALINE(adaptive linear element)과 그것의 학습법칙인 Widrow-Hoff 법칙으로 유명하다. 63년에는 ADALINE^[9]으로 inverted pendulum을 제어했으며 87년에는 pendulum의 상태를

sensor 대신 시각 입력으로 대체하여 같은 문제를 풀었다.^[10] 이와 비슷한 것으로 88년도 Boston에서 열린 INNS(International Neural Network Society) 학회에서 HNC 회사가 ANZA plus와 자사의 독특한 알고리즘으로 pole-balancing 문제를 푸는 실험을 보였다.

4. Guez, Pao, Josin의 Back Propagation Feed-forward Layer 제어방식

이들은 모두 inverse kinematic 함수의 샘플로써 back propagation 회로를 훈련시켜 그것을 테스트 입력에 대하여 일반화하는 것을 보였다. Guez^[11]와 Josin^[12]은 2축 로봇트 simulation으로, Sobajic은 실제로 INTELLEDEX 로봇트의 3,4축을 제어하였다.^[13]

5. Guez의 Human Teacher를 직접 모방하는 제어방식

신경 회로를 훈련시키는 방법으로 종래의 제어방식을 이용하는 것이 있고, 또 사람이 눈으로 보며 직관적으로 제어시키는 방법을 Guez가 설명하였다.^[3] Cart-Pole 문제를 사람이 직접 푸는 것을 신경회로가 옆에서 보고 배운 후에는 신경회로가 자발적으로 제어하는 방식을 보였다. 이는 산업 현장에서 사람이 하는 일을 신경회로로 대체하는 데도 쓰일 수 있리라 본다.

6. Ritter의 Topology-Conserving Map을 사용한 로봇트의 Kinematic과 Dynamic 제어

Ritter^[14]는 로봇트의 state space를 작은 구역들로 나눈 후 각 구역을 2차원의 뉴론 배치로 mapping한 후 이들 뉴론들이 로봇트를 제어하는 알고리즘을 소개했다. 각 뉴론은 담당 구역내의 함수의 Taylor 전개계수를 계산한다. 이 방법이 카메라를 이용한 inverse kinematic 제어와 dynamic 제어에 응용되었다.

7. Barto^[15]의 Reinforcement 제어

Barto는 cart-pole 시스템의 보상 함수가 pole이 쓰러지거나 cart가 벽에 닿을때만 negative reinforcement가 작용함으로써 그전의 순간 순간의 reinforcement를 예측하는 adaptive critic element와 associative search element 신경 회로를 동시에 썼다. 이 두 종류의 뉴론들은 Barto가 고안한 독특한 학습법칙을 가지고 있다.

8. Miyamoto의 Feedback Error Learning 제어^[16]

그림11의 방법은 세 가지면에서 다른 방법과 특이

하다. 우선 기존의 PD제어와 신경회로가 병렬로 연결되어 제어를 한다. 제어의 주체는 처음에는 PD제어이고 그 후 신경회로가 충분히 학습을 하면서 점차 신경 회로쪽으로 넘어간다. 둘째로 dynamics의 비선형 항들이 미리 계산된 후 뉴론에 의해 결합된다는 것이다. 이로써 뉴론은 선형 계산만 하게 되므로 부담이 줄어든다. 셋째로 feedback torque가 오차 신호로 작용하여 back propagation 회로를 훈련시킨다. Kawato는 3축 로봇트에 대해 simulation 하였고, Miyamoto는 실제 PUMA 260 로봇트에 적용하였다.

로봇트의 Newton-Euler dynamic 방정식을 전개하여 각 항의 계수를 로봇트를 움직이며 훈련시키는 dynamic 제어를 제안했다. 이로써 single layer 구조를 쓸 수 있어 훈련이 용이해진다. 3축 로봇트의 simulation 제어를 예시했다.

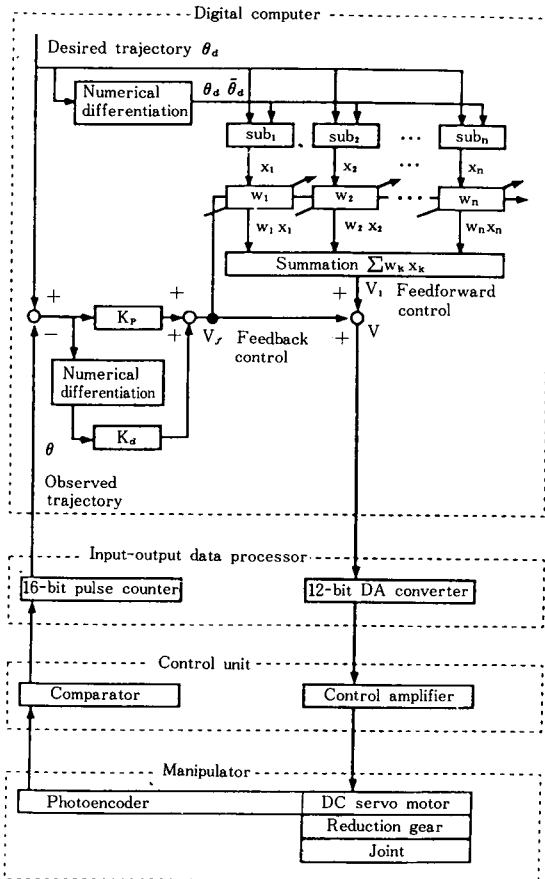


그림 11. Miyamoto의 feedback error 학습 제어 구조

9. Sanner의 Back Propagation 회로를 이용한 최적 제어^[17]

Sanner는 플랜트 앞의 신경회로 제어장치 (neuro-morphic controller)를 입력 에너지와 위치오차를 동시에 극소화하도록 훈련시키는 방법을 제안했다. 즉 신경회로를 써서 최적화 제어문제를 풀고자 한 것이다. 3층 회로구조를 몇개의 간단한 선형시스템들과 2축 로봇트의 조인트 토크제어에 사용했다.

10. Elsley의 Inverse Jacobian 제어 방식

Inverse Jacobian 제어는 Cartesian 위치 오차에 inverse Jacobian을 곱하여 제어명령을 내린다. Elsley는 실제로 로봇트를 움직이면서 back propagation 회로를 훈련시켜 이 inverse Jacobian mapping을 수행하도록 하였다.^[5] 이는 2축 로봇트의 시각 제어 simulation에 이용되었다.

11. Tsutsumi의 Hopfield 회로를 이용한 Obstacle Avoidance 제어^[18]

Hopfield 회로는 constrained optimization 문제에 많이 이용되고 있다. Tsutsumi는 이를 8축 평면 로봇트가 장애물을 피해서 목표에 도달하도록 제어하는데 이용하였다. 각 link 위치의 함수로써 에너지를 정의하고 이 에너지를 극소화함으로써 여러 constraint를 만족하면서 목표위치에 도달하게 된다.

VI. 신경회로 제어의 성능분석과 문제해결

로봇트의 성능은 다음과 같은 양들로 판정된다. 즉,

- (1) 위치정확도
- (2) 반복도
- (3) 최대속도
- (4) 최대부하/중량비
- (5) Controller 설계 및 구현에 드는 시간과 가격등

그것이다. 신경회로를 사용하면 이러한 성능 파라미터들이 얼마나 개선이 될 수 있는지 고찰해 보기로 한다. 신경회로는 실제의 시스템에서 데이터를 추출하여 훈련시키기 때문에 기타의 해석적 방법으로 모델하기 힘든 dynamics나 마찰, 잡음등의 효과까지 implicit하게 고려된다. 신경회로는 충분한 뉴론의 수와 layer를 쓰면 어떠한 mapping이라도 임의의 정확도를 가지고 나타낼 수 있으므로 정확도 및 반복도 면에서 종래의 제어보다 우수하다. 최대속도 역시 dynamics를 고려해 주므로 개선할 수 있다. 최대부하

/중량비는 요즘 많이 연구되는 flexible manipulator 로써 대폭 개선하는데 종래의 제어로는 flexible arm 을 효과적으로 다룰 수가 없다. 지금 수준이 간신히 1축 로봇을 제어하는데 머무르는 것으로 안다. 신경회로제어가 flexible arm 같은 복잡한 multi-parameter 시스템을 효과적으로 제어할 수 있다는 것은 이미 밝힌 바 있다.

마지막으로 neurocontroller를 써서 산업용 로봇을 설계할 때의 경제적인 측면을 보겠다. 신경회로는 이미 하드웨어나 소프트웨어로 구현이 되어 있으므로 실제 로봇에 연결하여 훈련을 시키면 되므로 훈련이 간단하다. 또한 로봇이 바뀌어도 같은 제어를 재훈련시키면 되므로 범용제어기로 사용할 수 있다.

Ⅶ. 결 론

신경 회로의 실로 광범위한 응용중에 시스템 제어가 차지하는 중요도는 산업 공정제어 시스템이라는 막대한 시장으로 보아 가장 크다고 본다. 그러나 아직까지 발표된 바에 의하면 상업적 응용은 하나도 없고 단지 실험실 단계에서 실제의 시스템에 적용하고 있는 것이 Miller의 CMAC 제어, Jennings⁽¹⁹⁾의 proximity sensor를 사용한 로봇 Fork lift, Hogan⁽²⁰⁾의 potential energy 신경회로를 사용한 obstacle avoidance, Jorgensen⁽²¹⁾의 이동로봇 제어 등이 있을 뿐이다. 그럼에도 불구하고 비교적 단일일내에 상업적 응용시스템이 나올 전망이며 이는 기존의 컴퓨터 시스템 simulation으로도 가능하다. 장기적으로 볼 때 신경회로제어가 실제시스템에서 보편화 되기 위해서는 다음과 같은 연구가 선행되어야 한다.

- (1) 신경회로의 경제적 구현 방식 - analog, digital, optical 회로
- (2) 빠르고 효율적인 학습 알고리즘
- (3) 사용하기 쉬운 software 개발
- (4) 신경제어의 새로운 응용분야 개발


신경회로의 제어 응용이 세계적인 큰 관심을 가지고 연구된 것은 약 2년으로 그나마 제어분야의 응용은 패턴 인식의 응용보다 발표된 결과가 적은 편이다. 그나마 지극히 간단한 시스템에 대해 주로 simulation을 통한 실험이 대부분이다. 따라서 이 정도의 미미한 결과를 가지고 신경회로가 미래의 제어 문제를 다 해결하리라는 기대는 성급하다고 볼지도 모른다. 그러나 현재까지 얻은 결과들은 한결같이 서론에서 나열한 바와 같이 종래의 제어 방식에서 볼 수

없는 너무도 중요한 장점들을 가지고 있다. 종래의 방식으로 풀 수 있는 문제들은 더 효과적이며, 효율적으로, 종래의 방식으로 풀 수 없는 문제의 해를 가능하게 할지도 모른다.

신경 회로는 이런 시스템을 가장 효과적으로 제어하는 방식일 것이다. 이로 볼때 앞으로 20년간 현존하는 산업 제어 시스템은 차차 신경 회로제어로 부분적 또는 전체적으로 대체되어야 한다. 물론 그에 앞서 기존의 시스템과 신경 회로방식을 어떻게 효율적, 경제적으로 결합하느냐 하는 engineering 연구가 선행되어야 함은 물론이다.

參 考 文 獻

- [1] Y.H. Pao, *Adaptive Pattern Recognition and Neural Networks*, Addison-Wesley, 1989.
- [2] D. Psaltis, A. Sideris, and A.A. Yamamura, "A Multilayered Neural Network Controller," *IEEE Control Systems Magazine*, pp. 17-21, April 1988.
- [3] A. Guez and J. Selinsky, "A Neuromorphic Controller with a Human Teacher," *Proc. of the IEEE International Conference on Neural Networks*, pp. 595-602, July 1988.
- [4] A. Guez, J.L. Eilbert, and M. Kam, "Neural Network Architecture for Control," *IEEE Control Systems Magazine*, pp. 22-24, April 1988.
- [5] R.K. Elsley, "A Learning Architecture for Control Based on Back-Propagation Neural Networks" *Proc. of the IEEE Conference on Neural Networks*, pp. 587-594, 1988.
- [6] W.T. Miller, E. Glanz, and L.G. Kraft, "Application of General Learning Algorithm to the Control of Robotic Manipulators," *The International Journal of Robotics Research*, pp. 84-98, Summer 1987.
- [7] J.S. Albus, "Mechanisms of planning and problem solving in the brain," *Mathematical Biosciences*, 45, pp. 247-293, 1979.
- [8] W.T. Miller, "Sensor based control of robotic manipulators using a general learning algorithm," *IEEE Journal of Robotics and Automation*, vol. RA-3, no. 2, pp. 157-165, 1987.
- [9] B. Widrow, "The Original Adaptive Neural Net Broom-Balancer," *The IEEE Interna-*

- tional Symposium on Circuits and Systems, IEEE, pp. 351-357, May 1987.
- [10] V.V. Tolat and B. Widrow, "An Adaptive Broom Balancer with Visual Inputs," Proc. of the IEEE Conference on Neural Networks, pp. 641-647, 1988.
- [11] A. Guez and Z. Ahmad, "Solution to the Inverse Kinematic Problem in Robotics by Neural Networks," Proc. of the IEEE Conference on Neural Networks, pp. 617-624, 1988.
- [12] G. Josin, D. Charney, and D. White, "Robot Control Using Neural Networks," Proc. of the IEEE International Conference on Neural Networks, pp. 625-631, July 1988.
- [13] D.S. Sobajic, J.J. Lu, and Y.H. Pao, "Intelligent Control of the INTELLEDEX 605T Robot Manipulator," Proc. of the IEEE Conference on Neural Networks, pp. 633-640, 1988.
- [14] H.J. Ritter, T.M. Martinez, and K.J. Schulten, "Topology-Conserving Maps for Learning Visuomotor-Coordination," The First INNS Meeting, Boston, 1988.
- [15] A.G. Barto, R.S. Sutton, and C.W. Anderson, "Neuronlike adaptive elements that can solve difficult learning control problems," *IEEE Trans. Systems, Man, and Cybernetics*, vol. SMC-13, no. 5, September 1983.
- [16] H. Miyamoto, M. Kawato, T. Setoyama, and R. Suzuki, "Feedback-error-learning neural network for trajectory control of a robotic manipulator," *Neural Networks*, vol. 1, no. 3, 1988.
- [17] R.M. Sanner and D.L. Akin, "Neuromorphic Regulation of Dynamic Systems Using Back Propagation Networks, Poster Session, First Annual INNS Meeting, Boston, 1988.
- [18] K. Tsutsumi and H. Matsumoto, "Neural Computation and Learning Strategy for Manipulator Position Control," Proc. of the IEEE International Conference on Neural Networks pp. 525-534, 1987.
- [19] V.A. Jennings, "Fork Lift Robot," DARPA Neural Network Study, pp. 445-450, 1988.
- [20] N. Hogan, "Tool-Use in Biological and Artificial Systems," DARPA Neural Network Study, pp. 538, 1988.
- [21] C.C. Jorgensen, "Neural network representation of sensor graphs for autonomous robot navigation," International Conference on Neural Networks, June 1987. 

筆者紹介



吳世泳

1952年 9月 18日生

1974年 서울대 공대 전자공학과 졸업

1981年 Case Western Reserve Univ. (공학박사)

1981年~1984年 University of Illinois at Chicago 조교수

1985年~1988年 University of Florida 조교수

1988年~현재 포항공대 전자전기공학과 부교수