

로봇 매니플레이터의 퍼지논리 제어를 위한 신경회로망을 사용한 규칙 베이스 유도방법

* 이석원*, 경계현*, 김대원**, 이범희*, 고명삼*
*서울대학교 제어계측공학과, 자동화시스템공동연구소
**영지대학교 제어계측공학과

A Rule Base Derivation Method using Neural Networks for the Fuzzy Logic Control of Robot Manipulators

* Seok-Won Lee*, Kye-Hyun Kyung*, Dae-Won Kim**, Bum-Hee Lee*, Myoung-Sam Ko*
*Dept. of Control & Instrumentation Eng. Seoul National University,
Automation and Systems Research Institute
**Dept. of Control & Instrumentation Eng. Myong Ji University

ABSTRACT

We propose a control architecture for the fuzzy logic control of robot manipulators and a rule base derivation method for a fuzzy logic controller(FLC) using a neural network. The control architecture is composed of FLC and PD(Positional Derivative) controller. And a neural network is designed in consideration of the FLC's structure. After the training is finished by BP(Back Propagation) and FEL(Feedback Error Learning) method, the rule base is derived from the neural network and is reduced through two stages - smoothing, logical reduction. Also, we show the performance of the control architecture through the simulation to verify the effectiveness of our proposed method.

1. 서론

로봇은 여러개의 입력과 출력을 가진 복잡한 비선형 동적 특성(nonlinear dynamic characteristics)을 갖고 있으므로 정확한 동력학 모델(model)을 찾아내기 어렵고, 비록 찾아냈다 할지라도 주위의 환경이나 로봇 자체의 계수(parameter) 변화를 어떻게 모델화 시킬 것인가 하는 문제에 부딪힐 수 밖에 없다.

Computed torque 제어 방법[1]-[2] 등은 로봇의 완전한 동력학 모델을 필요로 하고 동력학 모델의 계수에 작은 오차가 발생해도 좋은 제어 결과를 얻을 수 없는 단점을 가지고 있으므로 이러한 단점을 보완할 수 있는 로봇 제어 방법에 관한 연구가 많이 이루어져 왔다.

적용제어(adaptive control) 방법[3]-[6]은 로봇의 정확한 동력학 모델을 필요로 하지 않으나 로봇의 계수 추정을 위한 계산량과 잡음(noise)에 대한 민감도가 로봇의 변수가 늘어남에 따라 증가하는 단점을 가지고 있다. 또한, 반복 학습 제어 방법[7]-[8]도 정확한 동력학 모델을 모르더라도 사용할 수 있으나 단지 반복 동작에만 적용할 수 있다는 단점을 가지고 있다.

한편, 정확한 수식으로 표현하기 힘든, 숙련된 제어 경험자의 정량적이고 부정확한 언어적 제어 규칙을 퍼지 집합 이론(fuzzy set theory)과 퍼지 논리(fuzzy logic)로써 구현하는 퍼지 논리 제어기(Fuzzy Logic Controller : FLC)는 지금까지 많은 분야에서 전통적인 제어 방법으로는 제어 불가능했던 시스템을 제어할 수 있고 그 성능 또한 우수함이 밝혀져 왔다.

그러나, 퍼지 논리 제어를 로봇 제어에 이용하는 데에는 많은 문제점이 따른다. 로봇은 서로 연관된 여러개의 입력과 출력을 갖는 시스템이므로 제어 경험자의 언어적 제어 규칙을 추출하는 것이 상당히 어렵고, 따라서 제어 성능에 핵심적인 영향을 미치는 규칙 베이스(rule base)의 미세 조정(fine-tuning)에 많은 시간과 노력이 필요하다.

그러므로, 로봇을 퍼지 논리 제어기로 제어하여 좋은 결과를 얻기 위해서는 첫째, 제어 대상체의 동력학을 반영하고 제어 목표에 충실한 퍼지 논리 제어기의 규칙 베이스를 유도할 수 있는 방법이 제시되어야 하고, 둘째, 제어 성능을 높이기 위한 전체적인 제어 구조가 마련되어야 한다.

한편, 신경회로망(neural network)은 간단한 구조를 갖고 있는 여러개의 단위 처리기(processing element)들과 그들을 연결시켜 주는 연결 가중치(connection weights)로 구성되며, 시스템 자체나 외부 환경의 변화에 적응할 수 있도록 학습되어 진다. 이러한 신경회로망을 퍼지 논리 제어기에 응용한다면 제어대상체의 동력학과 제어 목표에 대한 최적화를 만족시키는 규칙 베이스의 작성이 가능할 것이다.

지금까지 퍼지논리 제어기에 신경회로망을 이용하는 많은 연구가 이루어져 왔다. Iwata 등[9]은 오차 역전달 알고리즘(back propagation algorithm)을 사용하여 퍼지제어 규칙을 다층 신경회로망(multilayered neural network)에 학습시키고 이를 퍼지 제어에 이용하였다. 이 방법은 다층 신경회로망의 보간 능력(interpolative ability)을 퍼지논리 제어기에 이용함으로써 퍼지논리 제어기만을 사용했을 때보다 좋은 결과를 얻을 수 있고 퍼지제어 규칙의 수가 많아져도 출력 계산에 걸리는 시간은 항상 일정하다는 장점이 있으나, 퍼지제어 규칙을 작성하기 어려운 경우에는 효과를 볼 수 없다는 단점이 있다.

Horikawa 등[10]은 전문가(expert)의 제어 데이터(control data)를 이용하여 자동적으로 퍼지제어 규칙을 찾아내고 소속 함수의 미세 조절을 할 수 있는, 신경회로망을 이용한 새로운 퍼지논리 제어기를 제안하였다. 이 방법은 제어 규칙이 전혀 없는 상태에서 학습을 시작할 수 있다는 장점이 있으나 전문가의 제어 데이터를 얻을 수 없거나 얻기 힘든 경우에는 사용하기 어렵다는 문제점을 가지고 있다.

Enbutsu 등[11]은 퍼지 뉴런 구조(fuzzy - neuron network)를 사용하여 전문가의 경험과 지식이 포함된 훈련 데이터(training data)로부터 입력 뉴런과 출력 뉴런 사이의 연결 가중치들의 곱에 의해 계산되는 상관 지수(causal index)를 계산하고 이로부터 퍼지제어 규칙을 찾아내는 방법을 제안하였다. 그러나, 이 방법은 상관 관계가 알려져 있는 훈련 데이터 집합(training data cluster)을 만들지

어렵고 만들더라도 주관적이고 경험적인 자료일 수 밖에 없으므로 신뢰성에 문제가 있다. 그리고, 상관 지수의 상대적 크기만으로 제어규칙을 결정하기 때문에 그릇된 훈련 데이터로 학습시킨 경우 전혀 쓸모없는 제어규칙이 나올 수 있다.

Baba 등[12]은 은닉 뉴런(hidden neuron)을 통한 모든 경로(path)를 고려한 상관 지수를 이용하여 입/출력 관계에 관한 지식을 추출하는 방법을 제안하였다. 이 방법 역시 상관 지수를 구하기 위한 훈련 데이터를 구하는 방법과 신뢰성에 대한 문제가 발생한다.

Patrikar 등[13]은 자동 형성 제어(self-organizing control)를 위한 다층 신경회로망을 제안하였다. 이 방법은 입력층의 뉴런들이 입력 변수의 퍼지 부분집합으로 구성되어 있는 부분을 제외하면 일반적인 신경회로망 제어기와 비슷하고, 제어 규칙의 미세 조절에 필요한 성능 평가 결정표(performance measure decision table)를 만들기 어렵다는 단점이 있다.

본 논문에서는 신경회로망 구조를 이용하여 퍼지논리 제어기의 규칙 베이스를 유도하는 방법을 제시하고, 이 규칙 베이스를 사용한 퍼지논리 제어기로 로봇을 제어할 수 있는 제어 구조를 제안한다. 또한, 모의 실험을 통하여 그 성능을 비교, 검토한다.

본 논문은 다음과 같이 구성된다. 2 장에서는 본 논문에서 다룬 문제에 관하여 서술하며, 3 장에서는 퍼지논리 제어기의 기본적인 구조에 대해서 설명하고, 로봇을 제어할 수 있는 퍼지논리 제어 구조를 제안한다. 4 장에서는 퍼지논리 제어기의 규칙 베이스를 추출하기 위한 신경회로망 구조와 그 학습 방법 및 규칙 베이스의 유도방법을 제시한다. 5 장에서는 모의 실험 방법 및 그 결과를 검토하고, 6 장에서 결론을 맺는다.

2. 문제의 정의

일반적으로 퍼지 논리 제어기의 성능은 퍼지 논리 제어기의 규칙 베이스가 제어 대상체의 특성과 제어 전문가의 경험을 얼마나 적절히 반영하고 있는가에 달려있다. 그러나, 로봇은 서로 연관된 여러개의 입력과 출력을 갖기 때문에 제어 전문가의 경험을 언어적 제어 규칙으로 전환시키는 것이 매우 힘들고 퍼지 제어 규칙 베이스의 미세 조절에 많은 시간과 노력이 필요하다.

따라서, 로봇과 같은 제어대상체를 퍼지 논리 제어기로 제어하여 좋은 결과를 얻기 위해서는 제어 대상체의 동역학을 반영하면서 제어 목표를 달성할 수 있는 퍼지 논리 제어기의 규칙 베이스를 유도할 수 있는 체계적인 방법이 필요하다. 본 논문에서는 신경회로망을 규칙 베이스를 필요로 하는 퍼지 논리 제어기를 고려하여 구성하고 그 신경회로망을 학습시켜 퍼지 제어 규칙을 유도하는 방법을 제안한다. 이 제안한 방법에서 가장 중요한 문제는 신경회로망을 어떻게 학습시킬 것인가 하는 학습 방법과 학습이 완료된 신경회로망으로부터 규칙 베이스를 어떻게 유도할 것인가 하는 규칙 베이스 유도방법에 관한 문제이다. 본 논문에서는 각 학습 단계에서의 오차와 전 학습 단계에서의 오차와의 차이를 오차항으로 하여 신경회로망을 학습시키고 학습 완료 후 실제 제어 단계에서 그 신경회로망의 입출력 관계로부터 퍼지 제어 규칙들을 추출하는 방법을 사용한다.

이러한 방법으로 유도된 규칙 베이스를 사용하더라도 퍼지 논리 제어기만으로는 제어 성능에 한계가 있을 수 밖에 없다. 그러므로, 제어 성능을 높이기 위한 전체적인 제어 구조로서 제어기의 안정성을 보장받을 수 있는, 비례 미분 제어기(positional derivative controller)를 포함한 로

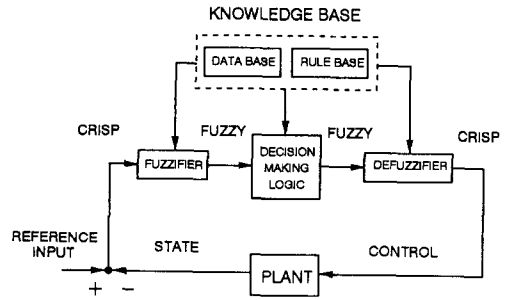


그림 1. 퍼지논리 제어기의 기본적 개념도

봇의 퍼지 제어 구조를 제시한다. 이 때, 발생할 수 있는 문제는 전체적인 제어구조가 달라지면 규칙 베이스를 유도하기 위한 신경회로망의 학습방법도 달라져야 한다는 점이다. 이 문제는 비례 미분 제어기를 포함시킨 상태에서 신경회로망을 학습시킴으로써 해결한다.

본 논문에서는 2 자유도 로봇을 이용한 모의 실험을 통하여 제안한 방법의 성능을 비교, 검토한다. 이를 통하여, 로봇의 목표 궤적이 주어졌을 때 로봇이 목표 궤적을 가장 안정적이면서도 효율적으로 추적하게 하는 퍼지 제어 방법 및 그 제어 규칙의 유도방법을 제시한다.

3. 로봇 제어를 위한 퍼지논리 제어기

3.1 퍼지 논리 제어기의 기초[14]-[17]

퍼지 논리 제어기에서 제어 규칙은 다음과 같이 IF~THEN~ 형태의 퍼지 프로덕션 규칙(fuzzy production rule)으로써 표현된다.

IF(전제 조건이 만족) THEN(특정 결론 추정)

퍼지 논리 제어기의 기본적인 개념도는 그림 1과 같으며, 각 블록(block)의 기능을 설명하면 다음과 같다:

- 1) 퍼지화 단계(fuzzification stage) : 크리스프(crisp)한 입력값을 퍼지 집합으로 바꾸어 준다.
- 2) 의사결정 단계(decision making stage) : 퍼지 규칙 베이스에 의거하여 퍼지 추론을 실행한다.
- 3) 역퍼지화 단계(defuzzification stage) : 의사결정 단계에서 결정된 퍼지 제어 입력을 역퍼지화 전략에 의하여 크리스프 값으로 바꾸어 준다.
- 4) 규칙 베이스(rule base) : 의사결정 단계에 퍼지 규칙을 제공해 준다.
- 5) 데이터 집합(data base) : 퍼지화 단계와 역퍼지화 단계에서 사용되는 퍼지 집합의 소속 함수를 제공한다.

3.2 로봇 제어를 위한 퍼지논리 제어 구조

본 논문에서는 목표 궤적을 추적하는데 필요한 토크(torque)를 퍼지 논리 제어기로부터 입력받아서 움직이는 로봇을 제어대상체로 한다.

목표 궤적(desired trajectory)과 실제 궤적(actual trajectory)사이의 오차(error)를 입력으로 받아들이는 퍼지 논리 제어기에서는 계산된 토크 출력이 직접 로봇에 전달되는 구조만으로는 제어 목표로의 최적화된 수렴성은 좋아 지지만 안정성(stability)을 보장받을 수 없다[18]. 따라

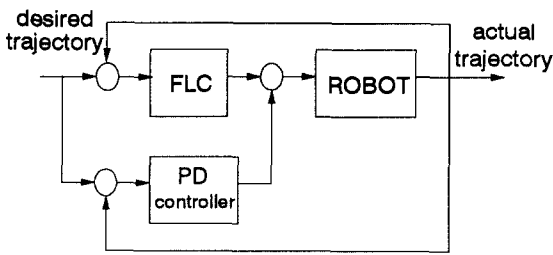


그림 2. 전체적인 제어 구조도

서, 그림 2와 같이 비례 미분 제어를 포함시켜 퍼지 논리 제어기만을 사용할 때의 단점을 극복한 한가지의 제어기 구조를 제시한다.

4. 규칙 베이스의 유도 방법

4.1 규칙 베이스를 추출하기 위한 신경회로망 구조

본 논문에서는 규칙 베이스를 추출하고자 하는 퍼지논리 제어기의 구조를 고려하여 신경회로망을 구성하는 방법을 사용한다. 즉, 퍼지논리 제어기의 각 입력 변수의 언어 변수치의 개수만큼 입력층의 뉴런의 개수를 설정하고, 출력층의 뉴런의 개수는 퍼지 논리 제어기의 출력의 개수와 같이 잡는다.

이렇게 신경회로망을 구성하면 다음 절에서 설명할 학습이 완료된 후 신경회로망의 입출력을 관측함으로써 퍼지논리 제어기의 규칙을 추출할 수 있다.

4.2 신경회로망의 학습 방법

4.1 절에서 설명한 퍼지논리 제어기의 규칙 베이스를 추출하기 위한 신경회로망을 학습시키기 위해서는 퍼지논리 제어기의 특성을 고려한 학습 방법이 필요하다. 실제 제어의 과정을 통해서 학습이 이루어지게 되는데 우선 신경회로망의 입력인 로봇의 각 관절의 위치 오차와 속도 오차는 입력층의 뉴런의 개수를 양자화 단계의 수로 한 양자화 과정이 필요하다. 이 때, 각 오차를 일제치로 나누어 줌으로써 오차의 한계를 설정한다. 양자화 된 각 입력에 해당하는 입력층의 뉴런에는 1.0의 값이, 다른 모든 뉴런에는 0.1의 값을 주게 되는데 이것은 퍼지 논리 제어기에서 입력이 퍼지 단자(fuzzy singleton)의 형태를 갖는 것을 가정하기 때문이다.

이러한 방식으로 입력을 넣어준 신경회로망의 학습시 관측되는 신경회로망의 출력도 역시 특정 값으로 나누어 줌으로써 출력 변수의 언어변수치의 개수만큼을 양자화 단계의 수로 한 양자화 과정을 거친다. 이 양자화 된 출력 값은 전 학습 단계에서의 값과 합해지고 역 정규화 상수 Ka 에 곱해진 후 비례 미분 되먹임 제어기의 출력과 함께 로봇에 목표 궤적을 추적하기 위한 토오코 값으로 전달된다.

토오코 값을 전달받은 로봇이 이동한 후, 관측된 로봇의 실제 궤적과 목표 궤적 사이의 오차가 계산되면 전 학습 단계에서의 오차와의 차를 오차항으로 하여 되먹임 오차 학습(FEL)방법[19]에 의하여 신경회로망이 학습되어진다. 이때, 시스템의 되먹임 오차(feedback - error)가 신경회로망을 통하여 역전달되어 되먹임 오차가 최소화되도록 신경회로망을 학습시킨다. 그리고, 양자화된 입력과 출력을 토대로 실제 학습이 이루어지므로 학습이 완료된 후 퍼지논리 제어기를 위한 규칙 베이스를 추출할 수 있는 기초가 마련

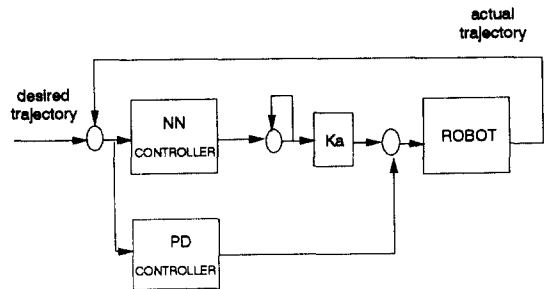


그림 3. 신경회로망의 학습 방법

된다. 이 과정을 그림으로 나타내면 그림 3과 같다.

앞에서 설명한대로 신경회로망은 제어 대상체의 동역학을 만족하도록 학습되어지므로 제어 대상체의 동역학에 대한 정보가 필요없고, 되먹임 오차가 최소화되도록 학습되어지므로 시스템 목표에 최적화된 퍼지제어 규칙을 추출할 수 있다.

4.3 신경회로망 구조로부터 규칙 베이스를 유도하는 방법

퍼지 논리 제어기를 고려한 신경회로망의 학습이 모두 끝나면 이 학습 완료된 신경회로망으로부터 퍼지 논리 제어기를 위한 규칙 베이스를 어떻게 추출하고, 추출된 규칙 베이스중에서 제어 성능을 높이는데 필요한 규칙들을 어떤 방법으로 선별할 것인가 하는 문제가 남는다.

신경회로망은 성능 기준(performance criteria)이 만족될 때까지 실제 제어 단계에서 학습된다. 학습 완료후 퍼지 논리 제어기를 위한 퍼지 제어 규칙은 신경회로망의 입출력 관계로부터 다음과 같이 추출된다.

제한한 신경회로망 구조에서 신경회로망의 출력은 양자화(quantization)된 크리스프한 제어 입력(control action)이고, 각 양자화 단계(quantization level)는 제어 입력변수의 언어변수치에 대응된다. 그러므로, IF~ THEN~ 구조로 이루어진 퍼지 논리 제어기의 규칙 베이스는 실제 제어 단계에서 궤적 오차가 충분히 수렴한 후 입력상태(input state)의 언어변수(linguistic variable)들에 해당하는 양자화 단계를 기록하고 그 때마다 신경회로망의 양자화된 출력을 관측하여 그 출력 단계에 해당하는 언어 변수치를 출력상태(output state)의 언어변수에 대응시킴으로써 추출할 수 있다.

본 논문에서 제안한 방법으로 추출된 모든 퍼지논리 제어 규칙이 제어 성능의 향상에 도움을 주는 것은 아니다. 즉 이 규칙들 중에는 오히려 제어 성능을 떨어뜨리는 잡음(noise)의 성격을 가지고 있는 것이 있다. 그러므로, 더 많은 수의 퍼지 논리 제어 규칙이 더 좋은 성능을 보장하지 않을 뿐만 아니라, 더 많은 수의 퍼지 논리 제어 규칙은 더 많은 계산량과 기억 용량을 요구하기 때문에 퍼지 논리 제어 규칙 베이스의 축소(reduction)가 필요한데 본 논문에서는 평탄화와 논리적 축소의 두 단계를 통하여 입력 변수의 언어 변수치가 같고 출력 변수의 언어 변수치는 다른 제어 규칙들을 정리한다. 이 과정들은 다음과 같이 두 개의 입력과 한 개의 출력을 가진 퍼지 논리 제어기의 경우에 대하여 설명한다.

입력 변수 : x_1, x_2

출력 변수 : y

각 변수의 언어변수치 집합 : { PB(Positive Big),

PM(Positive Medium), PS(Positive Small),
 ZO(Zero),NS(Negative Small), NM(Negative Medium),
 NB(Negative Big) }.

이때 입력 변수의 언어변수치들은 같고 출력 변수의 언어변수치는 다른 규칙들의 조합에 의해 다음과 같이 구성되는 퍼지제어 규칙들을 퍼지 규칙 집단(fuzzy rule group)으로 정의한다. 여기서 X, Y는 입력 변수의 언어 변수치를 ?는 출력 변수의 임의의 언어변수치를 각각 의미한다.

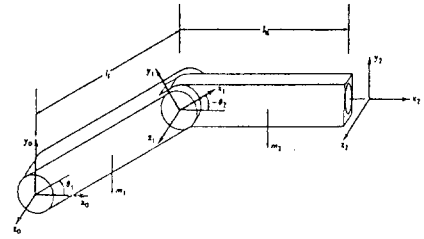


그림 4. 모의 실험을 위한 2 자유도 로봇

퍼지 규칙 집단 :

If x_{1or2} is X and x_{2or1} is Y, Then y is ?.
 If x_{1or2} is X and x_{2or1} is Y, Then y is ?.
 If x_{1or2} is X and x_{2or1} is Y, Then y is ?.
 If x_{1or2} is X and x_{2or1} is Y, Then y is ?.
 If x_{1or2} is X and x_{2or1} is Y, Then y is ?.
 If x_{1or2} is X and x_{2or1} is Y, Then y is ?.
 If x_{1or2} is X and x_{2or1} is Y, Then y is ?.

If x_{1or2} is X and x_{2or1} is Y, Then y is NB.
 If x_{1or2} is X and x_{2or1} is Y, Then y is NB.
 If x_{1or2} is X and x_{2or1} is Y, Then y is NB.
 If x_{1or2} is X and x_{2or1} is Y, Then y is NB.

→ If x_{1or2} is X and x_{2or1} is Y, Then y is NB.

1) 과정 1 : 평탄화(smoothing)

① 출력 변수의 모든 언어변수치를 숫자화 한다

(PB≡1, PM≡2, PS≡3, ZO≡4, NS≡5, NM≡6, NB≡7).

② 퍼지 규칙 집단으로부터 출력 변수의 언어변수치의 기대값 m_j 를 계산한다. m_j 를 계산하는 식은 다음과 같다:

$$m_j = \frac{1}{k} \sum_{i=1}^k n_i$$

여기서 k는 퍼지 규칙 집단에 속해 있는 규칙의 개수를, n_i 는 i번째 규칙의 출력 변수의 언어변수치의 값을, m_j 는 j번째 퍼지 규칙 집단의 n_i 의 기대값을 각각 의미한다.

③ j 번째 퍼지 규칙 집단의 모든 규칙의 출력 변수에 기대값 m_j 에 가장 가까운 언어 변수치를 대응시킨다.

If x_1 is X and x_2 is Y, Then y is NB.
 If x_1 is X and x_2 is Y, Then y is NB.
 If x_1 is X and x_2 is Y, Then y is NM.
 If x_1 is X and x_2 is Y, Then y is NB.
 If x_1 is X and x_2 is Y, Then y is NB.
 If x_1 is X and x_2 is Y, Then y is NB.
 If x_1 is X and x_2 is Y, Then y is NB.

→ If x_1 is X and x_2 is Y, Then y is NB.
 If x_1 is X and x_2 is Y, Then y is NB.
 If x_1 is X and x_2 is Y, Then y is NB.
 If x_1 is X and x_2 is Y, Then y is NB.
 If x_1 is X and x_2 is Y, Then y is NB.
 If x_1 is X and x_2 is Y, Then y is NB.
 If x_1 is X and x_2 is Y, Then y is NB.

2) 과정 2 : 논리적 축소(logical reduction)

신경회로망 구조로부터 추출된 퍼지제어 규칙은 논리적으로 비독립적인 규칙들을 포함할 수 있다. 이러한 규칙들은 논리적으로 독립적인 더 적은 수의 규칙들로 결합될 수 있다. 비독립적인 규칙들의 논리적 결합은 과정 1의 평탄화 과정을 거친 퍼지 규칙 집단을 하나의 규칙으로 정리함으로써 수행할 수 있다.

If x_{1or2} is X and x_{2or1} is Y, Then y is NB.
 If x_{1or2} is X and x_{2or1} is Y, Then y is NB.
 If x_{1or2} is X and x_{2or1} is Y, Then y is NB.

5. 모의실험 결과 및 검토

본 논문에서 제안한 퍼지 논리 제어기와 제어 구조의 성능을 평가하기 위해 그림 4의 2 자유도 로봇에 대하여 모의 실험을 실행한다.

이 로봇의 동력학은 다음 식과 같이 표현된다[20].

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{3}m_1 l_1^2 + \frac{4}{3}m_2 l_2^2 + m_2 C_2 l_1 l_2 & \frac{1}{3}m_2 l_2^2 + \frac{1}{2}m_2 l_1 l_2 C_2 \\ \frac{1}{3}m_2 l_2^2 + \frac{1}{2}m_2 C_2 l_1 l_2 & \frac{1}{3}m_2 l_2^2 \end{bmatrix} \times \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} + \begin{bmatrix} -\frac{1}{2}m_2 S_2 l_1 l_2 \theta_2^2 - m_2 S_2 l_1 l_2 \theta_1 \theta_2 \\ \frac{1}{2}m_2 S_2 l_1 l_2 \theta_1^2 \end{bmatrix} + \begin{bmatrix} \frac{1}{2}m_1 g l_1 C_1 + \frac{1}{2}m_2 g l_2 C_1 + m_2 g l_1 C_1 \\ \frac{1}{2}m_2 g l_2 C_2 \end{bmatrix}$$

여기서, $C_1 \equiv \cos(\theta_1), C_2 \equiv \cos(\theta_2), S_2 \equiv \sin(\theta_2), C_{12} \equiv \cos(\theta_1 + \theta_2)$ 로 정의되며, m_i 와 $l_i (i = 1, 2)$ 는 i번째 링크의 무게와 길이를 나타낸다. g는 중력 상수이다. 또한, 이 논문에서 사용한 조건들은 다음과 같다.

sampling time : T = 5 [ms]

링크의 무게 : $m_1 = 10$ [Kg], $m_2 = 5$ [Kg]

링크의 길이 : $l_1 = 1.0$ [m], $l_2 = 1.0$ [m]

중력 상수 : $g = 9.8062$ [m/sec²]

feedback gain : $K_P = \text{diag}(150, 150)$ [N/rad]

$K_V = \text{diag}(100, 100)$ [N·s/rad]

목표 궤적 : $\theta_1 = 5\pi/12 \cos(\pi t/2)$ [rad]

$\theta_2 = -\pi/3 \cos(\pi t/2) + 2\pi/5$ [rad]

초기 조건 : $\theta_1(0) = 5\pi/12$ [rad]

$\theta_2(0) = \pi/15$ [rad]

$\dot{\theta}_1(0) = \dot{\theta}_2(0) = 0$ [rad/sec]

퍼지 논리 제어기의 규칙 베이스를 추출하기 위한 신경회로망은 2개의 은닉층을 갖는 4층 신경회로망으로 구성하였고, 학습은 오차 역전달(Back Propagation) 학습 방법을 사용하였다. 퍼지 논리 제어기의 입력 변수의 언어변수치의 갯수가 21개이므로 입력층의 뉴런의 갯수는 각 입력 변수 당 21개가 필요하고 모두 4개의 입력을 가지므로 총 84개의 입력 뉴런을 사용하였고, 출력의 갯수가 2개이므로 출력 뉴런의 갯수는 역시 2개이다. 신경회로망의 학습시 사용한 학습률(learning rate)은 0.1, 모멘텀 계수(momentum)는 0.1이다. 신경회로망의 구조를 정리하면 표 1과 같고, 학습시 각 입력 변수와 출력 변수를 양자화하는데 사용한 계수값은 표 2와 같다.

그림 5에서는 신경회로망을 사용한 실제 제어 단계에서 다섯 번째 제어 단계에서의 목표 궤적과 실제 궤적 사이의 오차를 나타내었고, 그림 6에서는 양자화된 신경회로망의 출력의 변화를 나타내었다. 각 관절의 위치 오차와 속도 오차 모두 충분히 작은 오차 범위 안에 들었으나, 속도 오차는 위치 오차보다 변화의 폭이 매우 커서 제어하기에 어려움을 보여주고 있다.

6. 결론

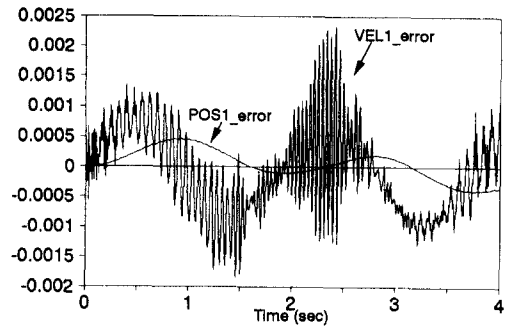
본 논문에서는 첫째, 제어 대상 시스템에 대한 정보와 전문가의 지식이나 경험 없이도 제어 대상 시스템의 동역학과 최적화를 만족시키는 제어 입력을 출력하는 퍼지논리 제어기의 규칙 베이스를 구할 수 있는 체계적인 방법을 제시하였다. 둘째, 로봇은 여러개의 입력과 출력을 가진 복잡한 비선형 동적 특성(nonlinear dynamic characteristics)을 갖고 있으므로 정확한 동역학 모델(model)을 찾아내기 어렵고, 비록 찾아냈다 할지라도 주위의 환경이나 시스템 자체의 계수(parameter) 변화를 어떻게 모델화 시킬 것인가 하는 문제를 가지고 있으므로 지금까지 많은 퍼지 제어 방법들이 제안되어 왔으나, 이들 방법보다 제어 성능이 우수한 새로운 방법을 제시하였다.

표 1. 모의실험에 사용한 신경회로망 구조

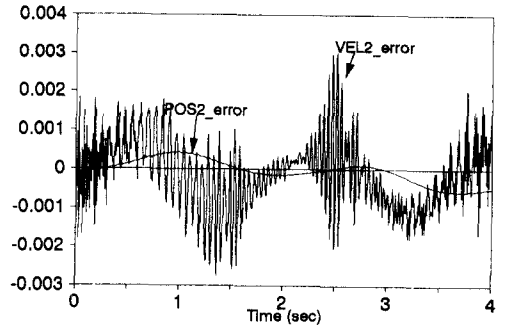
| | 입력층 | 제 1은닉층 | 제 2은닉층 | 출력층 |
|--------|-----|--------|--------|-----|
| 뉴런의 개수 | 84 | 20 | 20 | 2 |
| 학습률 | 0.1 | | | |
| 모멘텀 | 0.1 | | | |

표 2. 입출력 언어 변수의 양자화 계수

| 언어 변수 | 양자화 계수 |
|-------------------|--------|
| $e\theta_1$ | 0.011 |
| $e\dot{\theta}_1$ | 0.017 |
| $e\theta_2$ | 0.009 |
| $e\dot{\theta}_2$ | 0.0135 |
| τ_1 | 0.0072 |
| τ_2 | 0.008 |

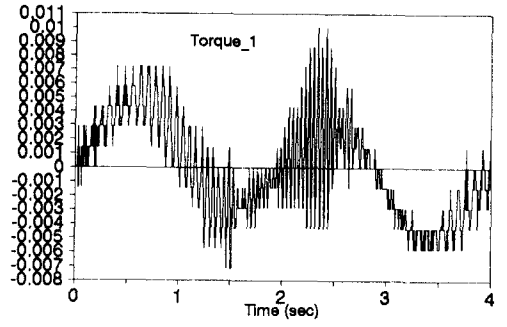


(a) 관절 1

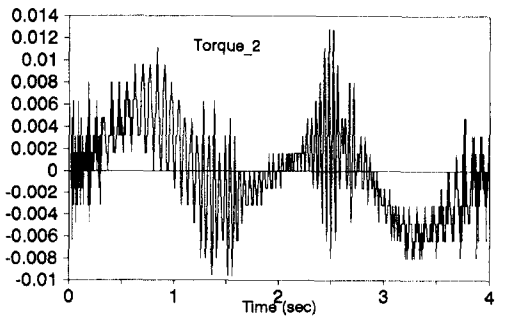


(b) 관절 2

그림 5. 신경회로망을 사용한 실제 제어시 각 관절의 위치 오차와 속도 오차



(a) 관절 1



(b) 관절 2

그림 6. 신경회로망을 사용한 실제 제어시 각 관절의 신경회로망의 출력

7. 참고 문헌

- [1] R.P.Paul, "Modelling, Trajectory Calculation and Servicing of a Computer Controlled Arm", Stanford Artificial Intelligence Lab. Memo AM-177, 1972.
- [2] J.Y.S.Luh, M.W.Walker, and R.P.Paul, "Resolved Acceleration Control of Mechanical Manipulator", IEEE Trans. Automat. Contr., Vol.AC-25, pp.468-474, 1980.
- [3] S.Dubowsky and D.T.DesForges, "The Application of Model Referenced Adaptive Control to Robot Manipulators", Trans. ASME, J. Dyn. Syst., Meas. Contr., Vol.101, pp.193-200, 1979.
- [4] A.J.Koivo and T.H.Guo, "Adaptive Linear Controller for Robotic Manipulators", IEEE Trans. Automat. Contr., Vol.AC-28, pp.162-171, 1983.
- [5] C.S.G.Lee and M.J.Chung, "An Adaptive Control Strategy for Mechanical Manipulators", IEEE Trans. Automat. Contr., Vol.AC-29, pp.837-840, 1984.
- [6] C.S.G.Lee and B.H.Lee, "Resolved Motion Adaptive Control for Mechanical Manipulators", Trans. ASME, J. Dyn. Syst., Meas. Contr., Vol.106, No.2, pp.134-142, 1984.
- [7] S.Arimoto, S.Kawamura, and F.Miyazaki, "Bettering Operation of Robots by Learning", J. Robotics Syst., Vol.1, pp.123-140, 1984.
- [8] S.Kawamura, F.Miyazaki, and S.Arimoto, "Realization of Robot Motion Based on a Learning Method", IEEE Trans. Syst., Man, Cybern., Vol.SMC-18, No.1, pp.126-134, Jan./Feb. 1988.
- [9] T.Iwata, K.Machida, Y.Toda, "Fuzzy Control using Neural Network Techniques", Proceeding of the International Joint Conference on Neural Network, pp. III.461-III.465, 1990.
- [10] S.Horikawa, T.Furuhashi, S.Okuma, and Y.Uchikawa, "A Fuzzy Controller using a Neural Network and its Capability to Learn Expert's Control Rules", Proceeding of the International Conference on Fuzzy Logic & Neural Networks, pp.103-106, 1990.
- [11] I.Enbutsu, K.Baba, and N.Hara, "Fuzzy Rule Extraction from a Multilayered Neural Network", Proceeding of the International Joint Conference on Neural Network, pp. II.461-II.465, 1991.
- [12] K.Baba, I.Enbutsu, and M.Yoda, "Explicit Representation of Knowledge Acquired from Plant Historical Data using Neural Network", Proceeding of the International Joint Conference on Neural Network, pp. III.155-III.159, 1990.
- [13] A.Patrikar and J.Provence, "A Self - Organizing Controller for Dynamic Processes using Neural Networks", Proceeding of the International Joint Conference on Neural Network, pp. III.359-III.364, 1990.
- [14] H.J.Zimmermann, Fuzzy set theory and its application, Kluwer Academic Publisher Group, Netherland, 1986.
- [15] C.C.Lee, "Fuzzy Logic in Control Systems : Fuzzy Logic Controller - Part I", IEEE Trans. Syst., Man, Cybern., Vol.20, No.2, pp.404-418, 1990.
- [16] C.C.Lee, "Fuzzy Logic in Control Systems : Fuzzy Logic Controller - Part II", IEEE Trans. Syst., Man, Cybern., Vol.20, No.2, pp.404-418, 1990.
- [17] 이계성, "퍼지 이론을 사용한 시스템 식별", 서울대학교 공학석사학위논문, 1992년.
- [18] 경계현, 이범희, 고명삼, "매니플레이터의 신경제어를 위한 새로운 학습 방법", 한국 자동제어 학술회의 논문집, pp.1022-1027, 1991.
- [19] 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, pp.251-265, 1988.
- [20] K.S.Fu, R.C.Gonzalez, and C.S.G.Lee, ROBOTICS : Control, Sensing, Vision, and Intelligence, McGraw-Hill Book Company, 1987.