한국퍼지 및 지능 시스템학회 논문지 1998, Vol. 8, No. 3. (1998) 56~66

신경망의 보상학습기능을 이용한 퍼지규칙의 자동생성기법

Automatic Fuzzy Rule Generation Using Neural Networks Based Reinforcement Larning

조재형* · 윤소정** · 오경환***

Jae Hyung Cho, So Jeong Youn and Kyung Whan Oh

*삼성전자 근무, **서강대학교 컴퓨터학과 박사과정
***서강대학교 컴퓨터학과 교수

요 약

본 논문에서는 보상 신호를 이용하는 근사 추론에 기반한 개선된 퍼지 논리 제어기를 제안한다. 제안된 방법은 근사 추론을 위한 인위적인 퍼지 규칙의 생성이나 소속 함수의 정의 없이 자동적으로 퍼지 논리 제어기를 구성할 수 있다. 제안된 퍼지 논리 제어기를 cart-pole 제어에 적용하여 기존의 방법들과의 비교를 통해 제시한 방법의 유용성을 검증한다.

ABSTRACT

In this paper, we propose an improved fuzzy logic controller based on an approximate reasoning method using reinforcements. Our fuzzy controller is constituted automatically without the artificial generation of the fuzzy rules for approximate reasoning and the determination of membership functions. To verify the soundness of our proposed fuzzy logic controller, it is applied to a cart pole balancing system and its performance is compared with other traditional methods'.

1. 서 론

퍼지 시스템의 적용 범위가 점차 확대되어가고 복잡해지는 현재의 상황에서는 사용되는 입출력 변수의 수가 많아지게 되어 전문가마다의 의견이 서로 상충되는 부분이 많아지게 되고, 이에 따라 규칙의 결정이 전문가의 주관적인 관점에 따라 크게 달라질 수가 있다. 또한 현재 대다수의 시스템이 비선형적인 입출력 관계의 특성을 갖고 있어 분석적인 모델링(analytical modeling)을 어렵게 만들어 이러한 문제가 더욱 두드러지게 되었다[6]. 또한 퍼지 규칙의 조건부와 결론부에 포함되는 퍼지 변수의 소속 함수를 어떻게 정의하는가에 따라 그 성능이 크게 달라진다는 것은 퍼지 시스템의 제작에 큰 부담으로 작용할 수밖에 없다. 그러므로 최적의 규칙을 결정한 후, 테스트를 통해 구성된 제어기의 제어 능력이 만족스러운 결과를 보여 줄 수 있도록 소속 함수와 규칙을 조정해야만 한다[2]. 최근의 경향은 스스로의 경험을 통해 학습할 수 있는 능력을 가진 퍼지 제어기를 설계하는 것이다[3,4]. 이를 위해 신경망(neural network)의 학습 능력을 퍼지 시스템에 도입하는 시도가 이루어지게 되었다[1,5].

퍼지 시스템에 신경망의 학습 능력을 도입하여 퍼지 제어기가 스스로의 경험을 통해 학습할 수 있는 능력을 갖게 하기 위해서 퍼지 제어기는 학습 데이터로부터 정보를 학습하여 스스로 규칙을 생성하고, 또 이 규칙들로부터 바람직한 제어를 추론해낼 수 있어야 한다. 그러나 비선형적인 입출력 관계로 인하여 수학적인 방법으로는 분석이 불가능할 뿐만 아니라 학습에 사용될 학습 데이터를 모집할 수 없는 경우에는 이와 같은 단순한 방법만으로는 퍼지 제어기의 구성할 수 없다. 이 경우 보상 신호(reinforcement)를 이용하여 학습을 가능하게 하는 것이 하나의 해결책이다. 보상 신호를 이용한 학습은 Sutton이 시간차 방법(temporal difference method)[12,14]에 의해 예측을 통한 학습을 제안함에 따라 본격적인 논의가 되었고, 이를 퍼지 규칙으로 표현하여 제어기에 적용하려는 연구[1,4,6,11,15]가 수행되었다. 본 논문에서는 보상

※본 논문은 1996년도 한국학술진흥재단 자유공모과제 연구지원사업에 의하여 연구되었습니다.

신호를 이용하여 학습을 수행하고, 이로부터 자동적인 퍼지 제어 규칙의 생성과 근사 추론(approximate-reasoning)을 수행하는 개선된 퍼지 논리 제어기를 제안한다.

2. 보상 신호를 이용한 퍼지 논리 제어기의 구축

본 논문에서는 퍼지 제어 규칙의 생성과 퍼지 추론의 수행 및 관련 소속 함수를 학습을 통해 조정하도록 구성된 신경망을 이용하여 보상 신호를 통해 학습하고 동적 시스템을 제어할 수 있는 개선된 형태의 퍼지 논리 제어기를 제안한다. 본 논문에서 제안하는 퍼지 논리 제어기는 그림 1과 같은 구조를 갖는다.

그림 1에서 보여주는 퍼지 논리 제어기는 크게 세 가지 부분으로 구성되어 있다. 제어 행동 선정자(action selector)와 제어 행동 평가자(action evaluator)는 현재의 시스템 상태를 입력으로 받아 이를 병렬적으로 처리한다. 제어 행동 선정자는 제어를 결정하기 위해 내부적으로 구성되어 있는 퍼지 제어 규칙에 근거한 퍼지 추론을 통해 현재의 시스템 상태에 대하여 적절하다고 판단이 되는 제어 행동(recommended action) F를 출력하게 된다. 제어 행동 평가자는 현재의 시스템 상태와 오류 신호를 이용하여 시스템의 현재 상태에 대한 안전성과 관련된 값, 즉 그림 1에서 현재 상태로부터 얻게 될 보상신호 r에 대한 예측값에 해당하는 내부 보상 신호(internal reinforcement) $\hat{r}$을 출력하며, 이와 관련된 퍼지 제어 규칙을 내부에 포함한다. 확률 제어 행동 수정자(stochastic action modifier)는 제어 행동 선정자와 제어 행동 평가자의 출력을 확률적인 방법으로 보완하여 실제로 시스템에 적용하게 될 제어 행동을 결정하는 역할을 담당한다.

2.1 퍼지 제어 규칙 생성을 위한 연결 구조 구축 및 학습

퍼지 제어 규칙을 생성하고 소속 함수를 자동적으로 조정하는 신경망의 구성을 위해서는 소속 함수와 퍼지 제어 규칙, 퍼지화와 비퍼지화의 수행, 그리고 퍼지 함의 등을 연결 구조(connectionist structure)로 집약시키고, 여기에 소속 함수와 퍼지 제어 규칙을 학습할 수 있는 능력을 갖추도록 해야한다. 이와 같은 연결 구조를 갖는 신경망은 그림 2와 같이 구성한다.

1) 신경망을 이용한 퍼지 제어 규칙 생성

입력층의 노드는 단순히 입력을 받아 다음 계층으로 전달하는 역할만을 한다. O^1은 입력층 노드의 출력 값이다.

$$O^1 = x \tag{1}$$

• 입력 소속 함수층(input membership layer)

각 노드는 고유의 소속 함수를 갖고 있으며, 이 소속 함수는 입력의 퍼지화에 사용된다. 소속 함수가 삼각형의 형태를 취하도록 하기 위해서는 소속 함수를 다음과 같이 정의할 수 있다.

$$\mu(x) = \begin{cases} \dfrac{s - |x - c|}{s}, & x \in \left[c - \dfrac{s}{2},\ c + \dfrac{s}{2}\right] \\ 0, & otherwise \end{cases} \tag{2}$$

만일 소속 함수가 종 모양의 함수라면 다음과 같이 정의될 수 있다.

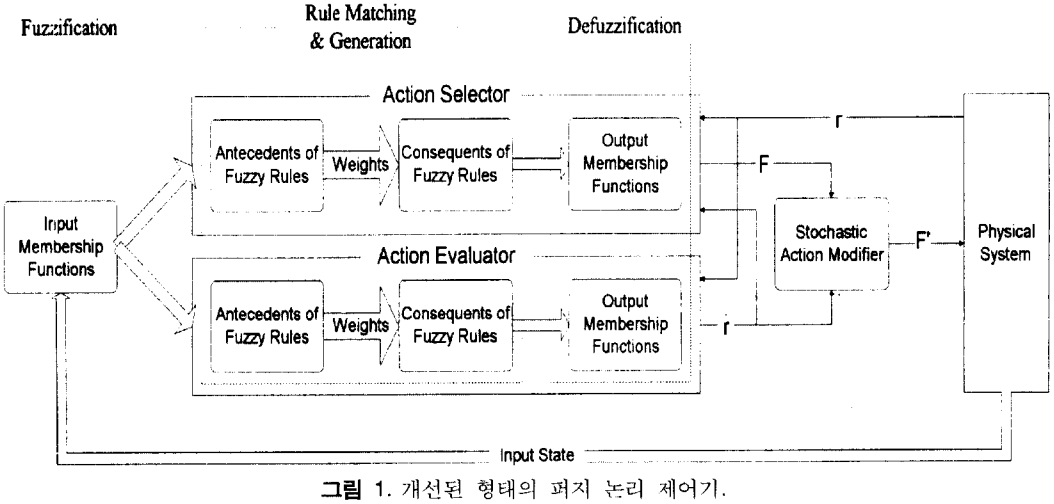


그림 1. 개선된 형태의 퍼지 논리 제어기.

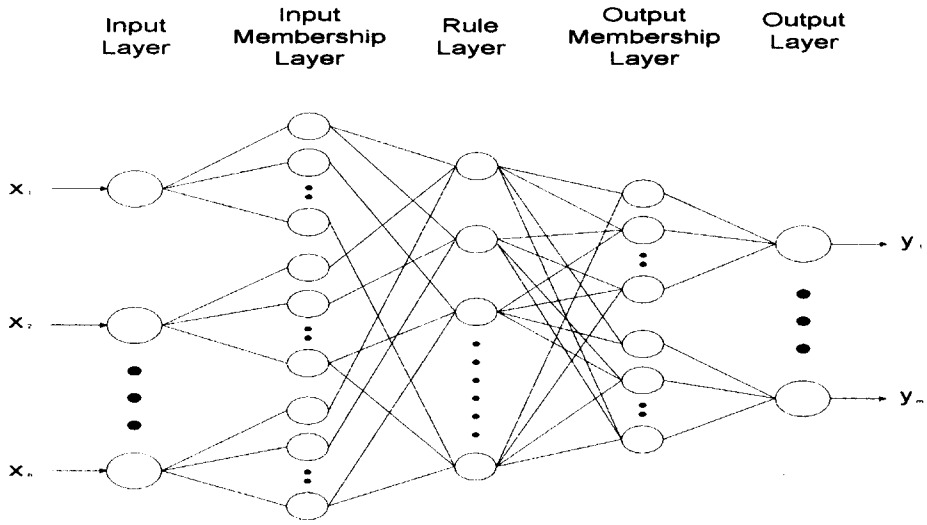


그림 2. 퍼지 제어 규칙 신경망(Fuzzy Control Rule Network).

$$\mu(x) = e^{\frac{-2e(x-c)^2}{S_2}} \tag{3}$$

식 (2)에서 c와 s는 각각 소속 함수의 중심과 폭을 나타낸다. O^2는 입력 소속 함수층 노드의 출력 값이다.

$$O^2 = \mu(x) \tag{4}$$

또한 소속 함수의 중심과 폭은 입력층과의 연결 상에서 연결 강도의 형태로도 표현이 가능하다

• 규칙층(rule layer)

규칙층은 퍼지 제어 규칙에서 존재 가능한 모든 조건을 결합하는 역할을 한다. 즉, 제어에 관한 규칙들을 총괄한 규칙 베이스(rule-base)가 된다.

$$r_i = \frac{\sum_j \mu_j e^{-k\mu_i}}{\sum_j e^{-k\mu_i}} \tag{5}$$

식 (5)에서 *softmin* 연산의 계산 결과로 산출되는 r_i은 규칙 i의 적용 가능 정도를 나타내게 된다. μ_i는 규칙 i의 조건에 해당하는 퍼지 레이블이 입력 변수와 얼마만큼 일치하는가를 나타내는 소속 정도를 나타낸다. 파라미터 k는 *softmin* 연산의 강도를 조절하는데, $k \rightarrow \infty$인 경우 *min* 연산과 같은 작용을 하게 된다. O^3는 규칙층 노드의 출력 값이다.

$$O^3 = r_i \tag{6}$$

• 출력 소속 함수층(output membership layer)

각 노드는 고유의 소속 함수를 갖고, 이 소속 함수는 결과를 산출해내기 위한 비퍼지화에 사용된다. 각 노드의 소속 함수는 입력 소속 함수층의 경우와 마찬가지로 중심과 폭으로 정의되고 삼각형이나 종 모양의 형태를 취하는 것이 가능하다. 규칙층의 노드에서 산출된 r_i이 입력으로 들어오면 출력 소속 함수층의 노드는 자신의 레이블을 결론으로 하는 점화된 모든 규칙들을 *OR* 연산을 이용하여 규합한다. 본 논문에서는 더하기 연산을 점화된 규칙들의 *OR* 연산에 사용하며, 연산의 결과는 1 이하의 값을 갖도록 한정시키기 위해 *min* 연산을 추가로 사용한다. O^4는 출력 소속 함수층 노드의 출력 값이다.

$$O^4 = min\{1, \sum_i w_{ji} r_i\} \tag{7}$$

• 출력층(output layer)

출력층의 노드는 $\mu^{-1}(r_i)$와 같이 나타낼 수 있다. 소속 함수 μ의 역함수에 해당하는 것으로 적절한 비퍼지화를 수행한다. 이것은 출력층과 출력 소속 함수층 간의 연결을 이용한다. 이 연결은 해당 출력 소속 함수층 노드가 갖고 있는 소속 함수의 중심과 폭을 곱한 수치를 연결 강도로 하며, 출력층의 노드가 제어 행동을 산출해낼 수 있도록 출력 소속 함수층의 노드에서 계산된 출력을 비퍼지화 하는 역할을 한다. 출력층의 노드는 비퍼지화된 입력들을 이용하여 제어 행동을 출력해야 하는데, 보통 이를 위하여 무게 중심법(center of area)을 이용한다. O^5는 출력층 노드의 출력 값이다.

$$O^5 = \frac{\sum_i c_{ji} s_{ji} O_i^4}{\sum_i s_{ji} O_i^4} \tag{8}$$

2) 퍼지 제어 규칙 신경망의 학습

신경망 학습의 목적은 문제의 특성에 맞게 정의된 오류 함수(error function)의 값을 최소화하는 것이다. 전체적으로 퍼지 제어 규칙 신경망의 학습은 어떤 입력에 대하여 퍼지 제어 규칙 신경망이 출력한 결과값으로부터 특정 메커니즘을 통해 오류값을 계산하고, 이 오류값을 상위 계층에서 하위 계층으로 역전파하여 내부의 연결 강도와 소속 함수를 조정하는 과정을 반복하게 된다.

퍼지 제어 규칙 신경망내의 소속 함수의 중심이나 폭을 나타내는 벡터(vector)를 p라 하면, p는 현재 퍼지 제어 규칙 신경망의 상태를 나타내게 된다. 또 E를 오류 함수로 정의하면 신경망에서의 학습 목표는 오류 함수의 값을 최소화하는 데에 있게 된다. 오류 함수 E는 신경망의 상태 p에 따라 다른 값을 갖게 되므로, E가 최소화되는 방향으로 p를 조정해야 한다.

$$p(t+1)=p(t)+\Delta p \tag{9}$$

식 (9)에서 p의 변화량 Δp는 E의 변화량과 다음과 같은 비례관계를 갖는다.

$$\Delta p \propto -\frac{\partial E}{\partial p} \tag{10}$$

$$p(t+1)=p(t)+\eta\left(-\frac{\partial E}{\partial p}\right) \tag{11}$$

그러므로 퍼지 제어 규칙에 사용되는 일반적인 학습 규칙은 다음과 같다.

$$\frac{\partial E}{\partial p}=\frac{\partial E}{\partial y}\frac{\partial y}{\partial p} \tag{12}$$

여기에서 η는 학습률(learning rate)이다. 식 (10)에서 $\frac{\partial E}{\partial p}$는 직접 구할 수 없는 것이 일반적이다. 그러므로 이 값을 학습에 이용하기 위해서는 다음과 같이 최급하강법을 적용시켜야 한다. 이러한 접근을 통해 퍼지 제어 규칙 신경망의 소속 함수 및 연결 강도는 다음과 같은 방법으로 조정된다.

• 출력층

출력층에서는 외부에서 계산된 오류값을 퍼지 논리 규칙 신경망으로 역전파 하는 역할만을 한다. δ^5는 출력층의 오류값으로 하위 계층으로 전파된다.

$$\delta^5=E \tag{13}$$

• 출력 소속 함수층

출력 소속 함수층에서는 소속 함수의 중심 c와 폭 s에 대한 조정이 이루어진다. 출력층과 출력 소속 함수층 사이의 연결 강도는 소속 함수에 의해 결정되므로 연결 강도는 소속 함수의 조정에 의해 함께 조정된다. 소속 함수의 중심은 식 (8)~(13)에 의해 다음과 같이 조정된다.

$$\frac{\partial E}{\partial c_i}=\frac{\partial E}{\partial y}\frac{\partial y}{\partial c_i}=-\delta^5\frac{s_i O_i^4}{\sum_i s_i O_i^4} \tag{14}$$

$$c_i(t+1)=c_i(t)+\eta\,\delta^5\frac{s_i O_j^4}{\sum_i s_i O_i^4} \tag{15}$$

한편 소속 함수의 폭은 유사한 방법으로 식 (8)~(13)에 의해 다음과 같이 조정된다.

$$\frac{\partial E}{\partial s_i}=\frac{\partial E}{\partial y}\frac{\partial y}{\partial s_i}=\delta^5\frac{c_i O_i^4(\sum_i s_i O_i^4)-(\sum_i c_i s_i O_i^4)O_i^4}{(\sum_i s_i O_i^4)^2} \tag{16}$$

$$s_i(t+1)=s_i(t)+\eta\delta^5\frac{c_i O_i^4(\sum_i s_i O_i^4)-(\sum_i c_i s_i O_i^4)O_i^4}{(\sum_i s_i O_i^4)^2} \tag{17}$$

출력 소속 함수층의 오류값 δ^4는 유사한 방법으로 다음과 같이 계산된다.

$$\delta^4=\frac{\partial E}{\partial \delta_i}=\frac{\partial E}{\partial y}=\frac{\partial E}{\partial (net-input)^5}\frac{\partial (net-input)^5}{\partial y}=\delta^5\frac{c_i O_i^4(\sum_i s_i O_i^4)-(\sum_i c_i s_i O_i^4)O_i^4}{(\sum_i s_i O_i^4)^2} \tag{18}$$

• 규칙층

규칙층에서는 규칙층과 출력 소속 함수층간의 연결 강도의 조정이 이루어진다. 이 연결 강도는 퍼지 제어 규칙의 신뢰도를 반영한다. 또한 출력 소속 함수층의 노드에서 출력되는 값은 해당 규칙이 제어에 기여한 정도를 나타내는 척도가 된다. 그러므로 연결 강도에 대한 수정은 다음과 같은 방법으로 이루어진다.

$$w_{ji}(t+1)=\eta O_i^4(t)(-w_{ji}(t)+O_j^3(t)) \tag{19}$$

규칙층의 오류값 δ^3는 다음과 같이 계산된다.

$$\delta^3=\frac{\partial E}{\partial \delta_i}=\frac{\partial E}{\partial y}=\frac{\partial E}{\partial (net-input)^4}\frac{\partial (net-input)^4}{\partial y}=\delta^4 \tag{20}$$

• 입력 소속 함수층

입력 소속 함수층에서는 출력 소속 함수층과 유사한 방법으로 소속 함수의 중심 c와 폭 s에 대한 조정이 이루어진다. 소속 함수의 중심은 다음과 같이 조정된다.

$$\frac{\partial E}{\partial c_{ji}} = \frac{\partial E}{\partial y_i}\frac{\partial y_i}{\partial c_{ji}} = -\delta^3 e^y \frac{2e(O_j^1 - c_{ji})}{s_{ji}^2} \quad (21)$$

$$c_{ji}(t+1) = c_{ji}(t) + \eta\,\delta^3 e^y \frac{2e(O_j^1 - c_{ji})}{s_{ji}^2} \quad (22)$$

유사한 방법에 의해 소속 함수의 폭은 다음과 같이 조정된다.

$$\frac{\partial E}{\partial s_{ji}} = \frac{\partial E}{\partial y_i}\frac{\partial y_i}{\partial s_{ji}} = -\delta^3 e^y \frac{2e(O_j^1 - c_{ji})}{s_{ji}^3} \quad (23)$$

$$s_{ji}(t+1) = s_{ji}(t) + \eta\,\delta^3 e^y \frac{2e(O_j^1 - c_{ji})}{s_{ji}^3} \quad (24)$$

입력층에서는 수정이 일어나지 않으므로 오류값의 계산은 필요하지 않다.

2.2 확률 제어 행동 수정자

확률 제어 행동 수정자(Stochastic Action Modifier)는 시간 t에서 제어 행동 선정자가 현재 상태에 적절하다고 판단하여 출력한 제어 행동 $F(t)$와 제어 행동 평가자가 시간 t에서 얻게 될 보상신호를 시간 t-1에서 예측한 값 $\hat{r}(t\text{-}1)$을 확률적인 방법을 이용하여 수정함으로써 실제로 동적 시스템에 적용할 제어 행동 $F'(t)$을 생성해낸다. 이를 위해 $F(t)$를 중심이 $F(t)$이고 표준 편차 $\sigma(\hat{r}(t\text{-}1))$인 가우시안 임의 변수(Gaussian random variable)로 설정한다. 이러한 과정이 필요한 이유는 신경망의 학습에 사용할 (*input, desired output*)형태의 학습 데이터의 모집이 불가능하기 때문에, 학습에 필요한 *desired output*을 확률적인 방법을 이용하여 결정하고 그에 따른 오류를 학습에 반영하기 위해서이다. 만일 제어 행동 평가자가 제어 행동 선정자가 산출해낸 제어 행동이 안정적인 제어를 수행했다는 판단을 내릴 경우에는 내부 보상 신호 $\hat{r}$은 큰 값이 될 것이고, 그 반대의 경우, 즉 성공적인 제어를 수행하지 못했다는 판단을 내릴 경우에는 내부 보상 신호 $\hat{r}$은 작은 값을 갖게 될 것이다. 그러므로 내부 보상 신호 $\hat{r}$의 값이 큰 경우에는 F에 적용해야할 수정의 범위가 작아야 하고, 그 반대의 경우, 즉 내부 보상 신호 $\hat{r}$의 값이 작은 경우에는 F에 적용해야할 수정의 범위가 커야한다. 그러므로 표준 편차 $\sigma(\hat{r}(t\text{-}1))$는 이와 같은 성질을 반영하도록 다음과 같이 정의된다.

$$\sigma(\hat{r}(t-1)) = \frac{1}{1+e^{\hat{r}(t-1)}} \quad (25)$$

그러므로 동적 시스템에 직접 적용될 제어 행동 F'의 값은 식 (25)에 의해 구해진 표준 편차 $\sigma(\hat{r}(t\text{-}1))$를 이용하여 구해진 범위 $[F(t)\text{-}\sigma(\hat{r}(t\text{-}1)),\ F(t)+\sigma(\hat{r}(t\text{-}1))]$내에서 임의로 결정된다. 여기에서 각 시간 단위마다의 섭동(perturbation)은 다음과 같이 나타낼 수 있다.

$$s(t) = \frac{F'(t) - F(t)}{\sigma(\hat{r}(t-1))} \quad (26)$$

2.3 제어 행동 선정자와 제어 행동 평가자의 구성 및 학습

제어 행동 선정자와 제어 행동 평가자는 각각 독립된 퍼지 제어 규칙 신경망이다. 제어 행동 선정자는 주어진 상태에 적용할 제어 행동을 결정하는데 필요한 규칙들을 생성하여 이를 기반으로 근사 추론을 수행하고 제어 행동 평가자는 현재의 상태에 대한 안정성을 나타내기 위한 규칙들을 생성하고 이를 기반으로 근사 추론을 수행한다. 제어 행동 선정자와 제어 행동 평가자의 학습은 앞에서 살펴본 방법에 따라 수행된다. 다만 제어 행동 선정자와 제어 행동 평가자는 학습 상에서의 오류값 E를 서로의 특성에 따라 서로 다른 값을 갖는다는 차이점을 갖는다.

이렇게 독립된 두개의 신경망이 필요한 이유는 시간 t에서 제어 행동 선정자와 제어 행동 평가자를 학습하기 위해서는 시간 t의 상태에 대한 오류값이 필요하기 때문이다. 시간 t에서의 오류값을 얻기 위해서는 보상 신호 $r(t)$이 필요한데, 이 값은 시간 t+1에서 얻을 수 있다. 그러므로 시간 t의 상태에서 얻을 수 있을 것으로 기대되는 보상 신호 $r(t)$의 예측값인 내부 보상 신호 $\hat{r}(t\text{-}1)$을 이용하여 오류값을 계산하는 방법을 이용한다.

1) 제어 행동 선정자의 학습

제어 행동 선정자에 적용되어야 할 오류값은 바람직한 제어 행동과 제어 행동 선정자가 결정한 제어 행동과의 차이가 되어야 한다. 그러나 바람직한 제어 행동은 구할 방법이 존재하지 않기 때문에, 2.2에서 살펴본 것과 같이 확률 적인 방법을 이용하여 본래의 오류값을 대체한다. 즉 바람직한 제어 행동을 F'의 값으로 사용하여 제어 행동 선정자가 결정한 제어 행동 F와의 차이를 오류값으로 이용한다. 그러나 여기에는 식 (26) 만큼의 섭동이 생기게 되고, 이는 보상 신호 r과 보상 신호의 예측값인 내부 보상 신호 $\hat{r}$의 차이에 비례하게 된다. 그러므로 이를 반영하여 다음과 같이 오류값을 결정한다.

$$E = (r(t-1) - \hat{r}(t-1))\frac{(F'(t) - F(t))}{\sigma(\hat{r}(t-1))} \quad (27)$$

식 (27)의 오류값을 2.1의 학습 방법에 적용하여 제어 행동 선정자를 학습한다.

2) 제어 행동 평가자의 학습

제어 행동 평가자에 적용되어야 할 오류값은 내부 보상 신호 $\hat{r}$의 값이 실제 보상 신호 r에 얼마나 근접하게 예측되어 왔는가를 반영해야 한다. 그러므로 이전의 내부 보상 신호에 대하여 현재의 내부 보상 신호가 얼마만큼의 진전을 보였는가를 보상 신호에 반영한 값을 오류값으로 이용해야한다. 그러므로 이러한 측면을 학습에 적용하기 위해 시간차 방법을 이용한다.

$$E = r(t) + \gamma\hat{r}(t) - \hat{r}(t-1) \tag{28}$$

식 (28)의 오류값을 3.1의 학습 방법에 적용하여 제어 행동 평가자를 학습한다.

2.4 퍼지 논리 제어기 구축

이상에서 살펴 본 각각의 기능들을 그림 1과 같이 결합하여 언어화된 퍼지 논리 제어 규칙을 생성할 수 있는 퍼지 논리 제어기를 구축한다. 여기에서 제어 행동 선정자와 제어 행동 평가자가 생성하는 규칙들은 동일한 입력 도메인 상에서 이루어져야 할 것이다. 만일 서로 다른 입력 도메인에서 이루어진다면 생성되는 규칙들은 서로 어긋나는 관점을 갖게 될 것이다. 그러므로 제어 행동 선정자와 제어 행동 평가자의 입력 도메인을 하나로 합하기 위해 입력층과 입력 소속 함수층을 공유하도록 구성한다. 학습의 관점에서는 제어 행동 선정자의 결정에 대하여 제어 행동 평가자가 평가를 한다는 것에 근거하여, 제어 행동 선정자가 입력 소속 함수층을 학습하도록 하고 제어 행동 평가자는 그것을 이용하도록 한다.

본 논문에서 제안하는 퍼지 논리 제어기는 학습의 종료 시점을 결정하는 데에 지정된 횟수를 이용한다. 일반적으로 역전파 신경망과 같은 기존의 신경망이 학습 데이터에 대한 오류 함수의 값을 기준으로 학습의 종료 시점을 결정한다. 그러나 제안된 방법에서는 학습 데이터의 모집이 불가능한 경우를 가정한 것이므로 이와 같은 오류 함수의 이용이 불가능하다. 그러므로 오류 함수를 이용하기 위해서는 계속적으로 바뀌는 입력 상태에 대한 제어기 내의 모든 오류값을 매 시간 간격마다 계속적으로 더하여 이를 경과된 시간 단위로 나누는 방법을 사용해야 한다. 그러나 이러한 방법은 기존의 방법처럼 학습 데이터들을 모두 학습하는 것을 단위로 오류 함수의 값을 결정하여 적절한 시기에 학습을 종료하는 것과는 달리, 누적된 오류값으로 인해 적절한 시기 이후에야 학습을 종료하게 되므로 초과 학습에 의해 제어기의 성능에 악영향을 미치게 된다. 그러므로 연속적인 제어의 결과를 관찰하면서 적절하다고 판단이 되는 시점에서 학습을 종료하거나 정해진 시간 동안만 학습을 한 후 종료하는 방법을 사용해야 한다.

3. 실험 및 결과

본 논문에서 제안된 개선된 퍼지 논리 제어기를 보상 신호를 이용한 제어기의 일반적인 성능 평가 방법으로 사용되는 cart-pole 문제에 적용시켜 보고, 그 결과를 Sutton[14]과 Beremji[15]에 의한 기존의 방법들과 비교하여 제안된 퍼지 논리 제어기의 타당성을 검증한다. 이 실험은 IBM PC 호환 기종인 Pentium 시스템에서 수행하였다.

3.1 Cart-pole 문제에의 적용

Cart-pole 문제에서 pole은 모터로 움직이는 cart 위에 놓여 있고, cart는 선로 위를 왼쪽이나 오른쪽으로만 이동할 수 있다. 이 문제의 제어 목적은 pole을 수직으로 균형을 맞추어 쓰러지지 않도록 하고, 또한 cart를 길이가 한정된 선로의 범위를 벗어나지 않으며 움직이도록 하는 것이다. 그림 3은 cart-pole 문제의

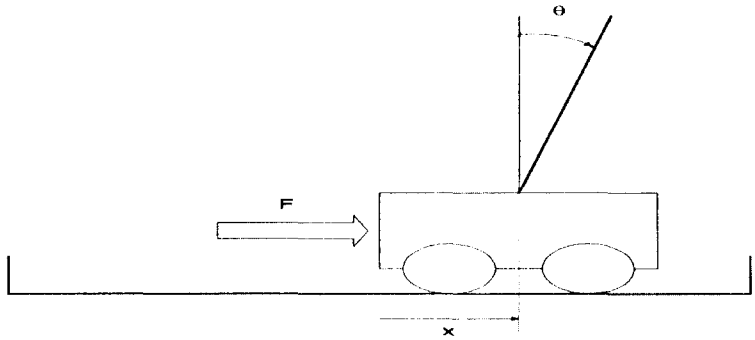


그림 3. Cart-pole 문제의 환경.

환경을 도식화한 것이다.

Cart-pole 문제에서는 네 가지의 상태 변수(state variable)가 시스템의 상태를 결정하고, 힘 F만이 cart의 제어에 이용된다. 이들 상태 변수는 표 1과 같다[23,24].

Cart-pole 시스템의 동적 특성은 다음과 같은 미분 가능한 비선형적인 수식으로 모델링할 수 있다[17].

$$\ddot{\theta} = \frac{g\sin\theta + \cos\theta\left[\dfrac{-F - ml\dot{\theta}^2\sin\theta + \mu_c sgn(\dot{x})}{m_c + m}\right] - \dfrac{\mu_p\dot{\theta}}{ml}}{l\left[\dfrac{4}{3} - \dfrac{m\cos^2\theta}{m_c + m}\right]} \tag{29}$$

$$\ddot{x} = \frac{ml[\dot{\theta}^2\sin\theta - \ddot{\theta}\cos\theta] - \mu_c sgn(\dot{x})}{m_c + m} \tag{30}$$

식 (29)와 (30)에서 사용된 상수들은 표 2와 같이 정의된다.

본 실험에서는 $|\theta| > 12^o$가 되거나 $|x| > 2.4$ m가 되면 실패로 간주한다. 이상의 수식과 상수값 등의 정보들은 퍼지 논리 제어기에 전혀 알려지지 않는다. 즉 퍼지 논리 제어기의 입장에서 cart-pole 모델은 블랙박스와 같다. 다만 cart-pole 모델에서 제공되는 보상 신호만이 퍼지 논리 제어기의 학습에 반영된다. 본 실험에서 정해진 보상 신호는 다음과 같다.

$$r(t) = \begin{cases} 0, & |\theta| \le 12^o \text{ and } |x| \le 2.4\text{ m} \\ -1, & otherwise \end{cases} \tag{31}$$

이와 같이 주어진 보상 신호를 이용해 III장에서 살펴본 것과 같이 학습이 이루어진다. Cart-pole에 대한 제어는 20 ms마다 이루어지며 이에 따라 20 ms마다 보상 신호를 얻게 된다.

본 실험에서는 입력과 출력 변수의 언어항은 표 3과 같이 정의하였다.

표 1. Cart-pole에서의 상태 변수 정의

상태변수	정 의
x	cart의 수평 위치(선로의 중심으로부터 cart의 중심 까지의 거리)
$\dot{x}$	cart의 속도
θ	pole이 수직선상으로부터 기울어진 각도
$\dot{\theta}$	pole θ의 각속도
F	cart에 적용되는 힘

표 2. Cart-pole에서의 상수 정의

상수	정 의
g	중력 가속도(-9.8 m/s^2)
m_c	cart의 중량(1 kg)
m	pole의 중량(0.1 kg)
l	pole의 길이(0.5 m)
μ_c	선로상에서 cart의 마찰 계수(0.0005)
μ_p	cart 위에서 pole의 마찰 계수(0.000002)

3.2 실험 방법 및 결과

1) 실험 방법

제안된 퍼지 논리 제어기는 cart가 선로의 중심에 위치해 있고 pole은 수직으로 되어 있는 상태로부터 학습을 시작한다. 퍼지 논리 제어기는 자신이 생성한 규칙에 의거하여 제어를 수행하고, 이를 동적 시스템 모델, 즉 cart-pole 모델에 적용시킨다. 이에 따라 cart의 위치와 pole의 기울기가 변화하게 되고, 다시 제어기는 변화된 상황에 대한 제어를 결정하게 된다. 이러한 과정에서 퍼지 논리 제어기는 자신의 소속 함수와 연결 강도를 조정하면서 제어 환경에 점차 적응하게 된다.

이와 같은 제어가 주어진 조건, 즉 $|\theta| \le 12^o$과 $|x| \le 2.4$ m을 만족하지 못하는 상황에 도달할 때까지를 한번의 시도로 간주한다. 전체적으로는 제어기의 학습 시간을 100번의 시도로 제한하고, 이 동안에 100,000번 이상의 제어가 일어난다면 학습이 만족할 만한 수준에 이르렀다고 간주하여 학습을 종료한다[14,15]. 학습이 완료된 후에는 더 이상의 소속 함수와 연결 강도에 대한 조정이 일어나지 않도록 하여 실제 시스템이나 플랜트 모델에 적용하게 된다. 실제 적용의 성공 여부 결정도 학습의 종료 결정과 마찬가지로 100번 이내의 시도 동안에 100,000번의 제어가 일어나는 경우 제어가 성공으로 수행된 것으로 간주한다.

2) 실험 결과 및 분석

학습을 통해 초기에 주어진 소속 함수와 연결 강도를 좋은 방향으로 조정하므로 써, 보통 40번 이하의 시도 내에 100,000번의 제어가 이루어지고, 100번의 시도 내에는 적어도 1,000,000번 이상의 제어가 이루어진다. 학습이나 실험 방법은 기존의 방법의 성능에 준하여 정해진 것인 만큼, 기존의 방법들이 70~90번 정도의 시도 내에 100,000번 정도의 제어를 수행하는

표 3. 입출력 변수의 언어항 정의

변수	언어항 정의
x	{*NF, NN, NC, AC, PC, PN, PF*}
$\dot{x}$	{*NF, NN, PN, PF*}
θ	{*NF, NN, AC, PN, PF*}
$\dot{\theta}$	{*NF, AC, PF*}
F	{*NL, NM, NS, PS, PM, PL*}
$\hat{r}$	{*NL, NM, NS, PS, PM, PL*}

것에 비해 비교적 좋은 제어 결과를 보여준다.

퍼지 논리 제어기 내의 퍼지 변수들 중 하나인 내부 보상 신호 $\hat{r}$의 소속 함수 변화를 살펴보면 학습 이후에 내부 보상 신호의 값이 큰 쪽으로 소속 함수들이 밀집해 있는 현상을 보여준다. 이는 제어 행동 선정자가 안정된 제어를 유지함에 따라 내부 보상 신호가 주로 큰 값을 출력하도록 조정되었음을 나타낸다. 제안된 퍼지 논리 제어기는 일반적으로 매 학습마다 조정된 소속 함수와 연결 강도에서 조금씩의 차이를 보인다. 이것은 제안된 퍼지 논리 제어기가 일정한 방향으로만 학습되는 것이 아니라 시스템의 상황에 따라 유동적으로 학습해 나가는 능력을 보유하고 있다는 사실을 보여준다. 이와 같은 소속 함수의 조정은 연결 강도의 조정과 연계되어 제어기가 최적의 제어를 수행할 수 있는 방향으로 수렴하게 한다.

제안된 퍼지 논리 제어기는 소속 함수의 조정과 연결 강도의 조정을 통해 제어 규칙을 생성하게 된다. 제어기 내부에 노드와 연결의 형태로 구성되는 제어 규칙들에 대한 신뢰도는 규칙층과 출력 소속 함수층 사이의 연결 강도로 표현이 된다. 따라서 추론망 내부에 구성되는 모든 규칙들을 고유의 신뢰도를 갖게 된다. 그러나 너무 낮은 신뢰도를 갖는 규칙을 제어에 이용하는 것은 바람직하지 않으므로 일정 기준 미만의 신뢰도를 갖는 규칙은 강제적으로 연결 강도를 0으로 주어 연결을 끊는 방법으로 규칙 베이스에서 제거하고, 일정 기준 이상의 신뢰도를 갖는 규칙만을 실제 제어에 이용한다.

표 4와 표 5는 각각 제어 행동 선정자와 제어 행동 평가자에 의해 생성된 규칙이다.

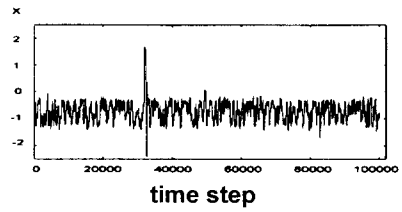
본 논문에서 제안된 방법에 의해 구성한 제어기를

표 4. 제어 행동 선정자에서 생선된 제어 규칙

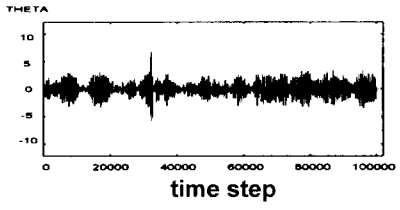
No.	x	$\dot{x}$	θ	$\dot{\theta}$	F	No.	x	$\dot{x}$	θ	$\dot{\theta}$	F
1	NF	NF	NN	AC	NL	17	AC	PF	PF	NF	NL
2	NF	NF	AC	PF	PL	18	PC	NF	AC	AC	NL
3	NF	NF	PF	AC	AC	19	PC	NN	NF	AC	NM
4	NF	NN	NF	PF	PF	20	PC	NN	NF	PF	NS
5	NF	PN	PN	PF	PF	21	PC	PN	PN	PF	PS
6	NN	NN	NN	PF	PF	22	PN	NF	PN	AC	NM
7	NN	PN	PN	NF	NF	23	PN	PN	PN	NF	PM
8	NN	PF	AC	AC	AC	24	PN	PN	PF	NF	NM
9	NN	PF	PN	PF	PF	25	PN	PF	NF	NF	NL
10	NC	NF	AC	AC	AC	26	PN	PF	NF	AC	NL
11	NC	NN	PN	AC	AC	27	PN	PF	PF	NF	NM
12	NC	PF	NF	PC	PF	28	PN	PF	PF	AC	PL
13	AC	NF	PN	PC	PF	29	PF	NF	NN	PF	NM
14	AC	NN	NF	AC	AC	30	PF	NF	PN	PF	NL
15	AC	PN	NN	PF	PF	31	PF	PN	NN	NF	PL
16	AC	PN	PF	PF	PF	32	PF	PF	AC	AC	NM

표 5. 제어 행동 평가자에서 생성된 평가 규칙

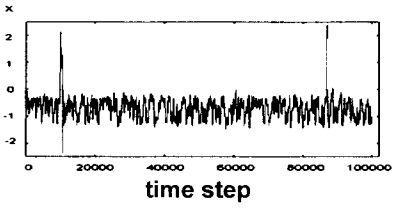
No.	x	$\dot{x}$	θ	$\dot{\theta}$	$\hat{r}$	No.	x	$\dot{x}$	θ	$\dot{\theta}$	$\hat{r}$
1	NF	NN	NF	PF	PL	16	PC	NN	PN	NF	NL
2	NF	PN	NF	NF	NM	17	PC	PN	AC	AC	NM
3	NF	PF	PN	NF	NL	18	PC	PF	NN	NF	NL
4	NN	NF	NF	NF	NS	19	PC	PF	PN	NF	NM
5	NN	NN	NF	NF	NS	20	PC	PF	PF	NF	NS
6	NN	NN	PN	PF	PM	21	PN	NF	NN	AC	NS
7	NN	PN	NN	NF	NS	22	PN	NF	PN	PF	NS
8	NN	PN	NN	PF	PL	23	PN	NN	NF	PF	NS
9	NN	PF	NF	PF	NM	24	PN	NN	AC	PF	NS
10	NC	NF	NF	AC	NS	25	PF	NF	NF	NF	NL
11	NC	NN	PF	NF	PM	26	PF	NF	PN	PF	NM
12	AC	NF	NF	AC	NS	27	PF	NN	NF	NF	PM
13	AC	NF	NF	PF	NM	28	PF	NN	PN	AC	NS
14	AC	PN	NF	NF	PM	29	PF	PF	NF	AC	NS
15	AC	PN	AC	AC	NL	30	PF	PF	AC	NF	PM



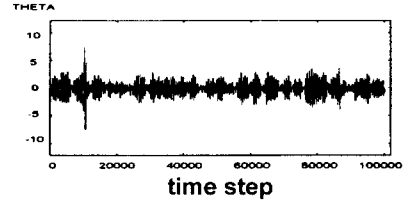
(a) 실험 1에서의 cart의 위치



(b) 실험 1에서의 pole의 기울기



(c) 실험 2에서의 cart의 위치



(d) 실험 2에서의 pole의 기울기

그림 4. 제안된 퍼지 논리 제어기의 제어 결과.

여러 차례 적용해 보고, 그중 두 가지의 실험 결과를 기존의 방법들과 비교하기 위해 시간 축에 대한 cart의 위치와 pole의 기울기의 값을 그림 4와 같이 도식화하였다.

그림 4에서 보여지는 두 번의 cart-pole 제어 결과는 안정된 흐름을 보여준다. 실험 1은 2번의 시도 내에, 실험 2는 3번의 시도 내에 100,000번 이상의 제어가 이루어진 것으로써, 기존의 방법에 비해 좋은 제어 결과를 보여준다. 이를 통해 시스템의 변화하는 환경에 능동적으로 적응하는 학습을 수행하고 있음을 알 수 있다. 이러한 평가는 기존의 방법들을 cart-pole 제어에 적용하여 얻은 다음의 실험 결과와의 비교를 통해 타당성을 검증할 수 있다.

Sutton[14]에 의해 제안된 방법을 cart-pole의 제어에 적용하여 얻은 실험의 경우 83번의 시도에 이르러 100,000번의 제어를 수행하였다. 변화의 폭이 크고 자주 제어에 실패하는 경향을 보이며, 매 시도마다 비슷한 제어 행동을 수행하는 것을 알 수 있다. 이를 통해 Sutton의 방법은 시스템의 변화하는 환경에 효과적으로 대처하지 못하는 약점을 가지고 있는 것으로 간주할 수 있다. 또한 Beremji에 의해 제안된 제어기인 GARIC(Generalized Approximate Reasoning-based Intelligent Control)을 cart-pole 의 제어에 적용한 결과에서는 100,000번의 제어까지는 67번의 시도가 소요되었다.

이상의 결과를 토대로 제어 수행에 있어서 본 논문에서 제안된 개선된 퍼지 논리 제어기가 기존의 방법에 비하여 만족스러운 성능을 보임으로써, 제안된 방법이 타당한 규칙을 생성하고, 이를 근거로 하여 타당한 추론을 수행함을 확인할 수 있다.

4. 결 론

본 논문에서는 분석적 모델링을 바탕으로 하는 기존의 제어기 구성 방법으로는 해결이 어려운 다수의 입출력 변수와 비선형적 특성을 가진 동적 시스템을 효과적으로 제어할 수 있는 개선된 퍼지 논리 제어기를 제안하였다. 본 논문에서 제안된 방법은 전문가를 통한 지식 및 규칙 획득이 어려운 경우에 적용이 가능하도록 스스로의 규칙 생성을 포함한다. 또한 제어기의 성능을 크게 좌우하는 민감한 부분인 퍼지 제어 규칙의 소속 함수를 전문가가 자신의 경험과 지식에 따라 주관적으로 결정하는 인위적인 방법을 통하지 않고 학습에 의해 소속 함수를 스스로 조정하여 입력 변수와 출력 변수의 퍼지화 없이 퍼지 관계에 기반을 둔 추론을 가능하게 한다. 또한 본 논문에서 제안된 퍼지 논리 제어기는 비선형적인 입출력 관계로 인해 분석적인 방법으로는 모델링이 불가능하여 학습 데이터의 모집이 불가능한 경우에도, 신경망 학습 방법의 근간이 되는 최급 하강법에 의해 유도된 개선된 학습 방법에 보상 신호를 도입하므로 써, 제어의 결과에 대한 극단적인 평가 외에는 시스템에 대한 어떤 정보도 알려져 있지 않은 보상 신호 문제에 적용이 가능하다. 본 논문에서 제안된 개선된 형태의 퍼지 논리 제어기는 기존의 방법이 적용될 수 없는 문제에 관한 언어화된 규칙을 생성할 수 있고, 이를 기반으로 근사 추론을 수행하여 비선형적인 동적 시스템에 대한 효율적인 제어를 가능하게 한다.

개선된 퍼지 논리 제어기는 규칙 생성이나 학습 데이터모집이 불가능한 경우에 기존의 방법에 비해 우수한 제어 성능을 보여준다. 제안된 퍼지 논리 제어기를 Cart-pole 제어에 적용한 III장에서의 실험 결과는, 일정 시도 내의 제어 시간을 비교하여 볼 때, Sutton이 제안한 방법에 비해 2.7배, GARIC에 비해 1.7배의 제어 성능 향상을 보여준다. 그러나 제안된 방법을 통해 추론망을 구축하고 학습시키는 과정에서 가장 어려운 점은 학습의 종료 시점을 파악하기가 어렵다는 점이다. 학습 종료 시점이 중요한 이유는 학습이 부족한 경우 좋은 결과를 얻을 수 없을 뿐만 아니라 지나친 학습이 오히려 제어기의 성능에 악영향을 미칠 수 있기 때문이다. 그러나 보상 신호를 이용한 학습의 경우에는 다른 경우에서처럼 추론망 내의 전체적인 오류를 계산해 내는 정립된 방안은 없다. 그러므로 본 논문의 실험에서는 경험적인 방법에 의해 가장 좋은 성능을 보여주는 학습 시간을 추정하여 학습 종료 시점을 결정하는 방법을 사용하였다. 그러나 이 시점이 언제나 좋은 성능을 보여줄 수 있는 학습량을 보장해 주지는 못한다. 그러므로 학문적인 접근을 통해 보상 신호를 이용한 학습의 경우에 제어기 내의 전체적인 오류를 추정해낼 수 있는 새로운 메커니즘의 개발이 필요할 것이다.

참고문헌

[1] V. Gullapalli, J. A. Franklin and H. Benbrahim, "Acquiring robot skills via reinforcement learning," *IEEE Control Systems*, pp. 13-24, Feb. 1994.

[2] C. T. Lin and C. S. George Lee, "Neural-network-based fuzzy logic control and decision system," *IEEE Trans. on Computers*, **40**(12), pp. 1320-1336, Dec. 1991.

[3] A. G. Barto, R. S. Sutton and C. W. Anderson, "Neuronlike adaptive elements that can solve difficult learning control problems," *IEEE Trans. Syst. Man, Cynern.*, **SMC-13**(5), pp. 834-846, 1983.

[4] C. C. Lee, "A self-learning rule-based controller employing approximate reasoning and neural net concepts," *Int. J. Intelligent Systems*, **5**(3), pp. 71-93, 1991.

[5] P. Werbos, "Beyond regression : New tools for prediction and analysis in the behavioral science," Ph.D. thesis, Harvard University, 1974.

[6] J. R. Jang, "Self-learning fuzzy controllers based on temporal back propagation," *IEEE Trans. on Neural Networks*, **3**(5), pp. 714-723, Sep. 1992.

[7] H. R. Berenji, "Fuzzy logic controllers," in *An Introduction to Fuzzy Logic Applications in Intelligent Systems*, R. R. Yager and L. A. Zadeh, Eds. Boston: Kluwer Academic, pp. 69-96, 1991.

[8] J. A. Bernard, "Use of rule-based system for process control," *IEEE Control Syst. Mag.*, **8**(5), pp. 3-13, 1988.

[9] Y. Kasai and Y. Morimoto, "Electronically controlled continuously variable transmission," in *Proc. Int. Congress Transportation Electronics*, pp. 45-53, 1988.

[10] T. J. Procyk and E, H. Mamdani, "A linguistic self-organizing process controller," *Automatica*, **15**(1), pp. 15-30, 1979.

[11] C. C. Lee and H. R. Berenji, "An intelligent controller based on approximate reasoning and reinforcement learning," in *Proc. IEEE int. Symp. Intellegent Control*, pp. 42-51, 1989.

[12] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Machine Learning*, **3**, pp. 9-44, 1988.

[13] J. Moody and C. J. Darken, "Fast learning in neural networks of locally-tuned processing units," *Neural Computation*, **1**, pp. 282-294, 1989.

[14] R. S. Sutton, Temporal credit assignment in reinforcement learning, Ph.D. thesis, University of Massachusetts, 1984.

[15] H. R. Beremji, "Learning and tuning fuzzy logic controllers through reinforcements," *IEEE Trans. on Neural Networks*, **3**(5), pp. 724-740, Sep. 1992.

[16] A. G. Barto, R. S. Sutton and C. W. Anderson, "Neurolike adaptive elements that can solve difficult learning control problems," *IEEE Trans. Syst., Man, Cybern.*, **SMC-13**, pp. 834-846, 1983.

조 재 형(Jae-Hyung Cho)

1994년 : 서강대학교 전자계산학과(학사)
1996년 : 서강대학교 전자계산학과(석사)
1996년~현재 : 삼성전자근무
주관심분야 : 전산선박설계, 형상모델링, CALS

오 경 환(Kyung-Whan Oh)

1978년 : 서강대학교 수학과(학사)
1985년 : Flolida State University 전산학과(석사)
1988년 : Flolida State University 전산학과(박사)
1989년~현재 : 서강대학교 컴퓨터학과 교수
주관심분야 : Intelligent Agent System, Fuzzy Neural Network, Computer Vision, Patter

윤 소 정(So-Jeong Youn)

1991년 : 서강대학교 전자계산학과(학사)
1993년 : 서강대학교 전자계산학과(석사)
1997년 : 한국전자통신연구원
1997년~현재 : 서강대학교 컴퓨터학과 박사과정
주관심분야 : Intelligent Agent System, Fuzzy Logic, Neural Network