

# 효율적인 뉴로-퍼지 시스템의 설계 방법론

## The Design Methodology of An Efficient Neuro-Fuzzy System

조 영 임\*, 황 중 선\*  
Young Im Cho\*, Chong Sun Hwang\*

### 요 약

퍼지 제어기(FLC)는 Max-Min CRI 방법을 이용하여 추론하는 시스템이다. 그러나 이 방법은 주관적인 멤버십 함수의 결정, 오류 발생 가중치 전략, 비합리적인 추론 규칙들의 조합이라는 세가지 문제점 때문에 원하는 추론 결과와 실제 추론 결과 사이에 상당한 오류 영역을 발생시킨다.

본 논문에서는 이를 해결하기 위해 퍼지 이론에 신경 회로망의 학습 기능이 융합되어 지능적으로 작동하는 뉴로-퍼지 시스템(INFS)을 제안한다. INFS는 이상의 문제 해결 방안이 지식 획득 단계, 적응 조절 단계를 통해 작동함으로써 임의의 입력에 대해서도 추론이 가능한 시스템이다.

제안된 INFS를 직류 계열 모터에 적용한 결과 신경 회로망을 사용하지 않았을 때 보다 오류 영역을 상당히 줄여주었다. 또한 학습 시간을 고려해 볼 때, INFS에서 사용하는 추론 방법(NCRI 방법)이 지금까지 다른 방법에 비해 훨씬 효율적이었다.

### ABSTRACT

Fuzzy Logic Controller(FLC) makes use of the Max-Min CRI(Compositional Rule of Inference) method. However this method has three problems: subjective formulation of membership functions, error-prone weighting strategy, and inefficient compositional rule of inference. Because of these problems, there are insurmountable error regions between desired output and inferred output.

To overcome these problems, we propose an Intelligent Neuro-Fuzzy System(INFS) based on fuzzy theory and learning functions of neural networks. In INFS, the problem solving techniques operate through knowledge acquisition phase and adaptation phase. Therefore this system can infer from any input facts.

Applying INFS to a D. C. series motor, we can conclude that the error regions are completely reduced. Also, considering learning time, NCRI method used in INFS, performs better than any other method.

### I. 서 론

퍼지 제어기(Fuzzy Logic Controller : FLC)는 퍼지 논리[15]를 이용한 제어기로, 기존의 모델 기반 제어 시스템이 다루기 어려웠던 수학적으로 정의하기 어려운 비선형, 동적 시스템 등에 적용되어 좋은 성능을 보여주었다. FLC에서는 시스템 출력을 결정짓는 추론 방법으로 Zadeh[15]가 제안한 Max-Min CRI(Compositional Rule

\*고려대학교 전산과학과

of Inference)를 이용하고 있다. Max-Min CRI 방법은 Mamdani가 제안한 최소 연산 법칙(mini-operation)을 이용하는 추론 방법으로, 직접법 또는 Zadeh의 추론 방법이라고도 하며 기존에 많이 사용되어온 퍼지 추론 방법이다.

Max-Min CRI의 추론 방법은 다음과 같다. 식 (1)과 같은 퍼지 조건문이 있어서 가능성 분포로 표현되었고,  $A'$ ,  $B'$ 가 각각 입력 사실과 추론 결과를 나타내며,  $R$ 은 퍼지 관계를 나타내고,  $\Pi_X$ ,  $\Pi_Y$ 는 가능성 분포값을 나타낸다.

$$IF X \text{ is } A \text{ THEN } Y \text{ is } B \Rightarrow IF \Pi_X = \mu_A \text{ THEN } \Pi_Y = \mu_B \quad (1)$$

식 (1)과 같은 퍼지 조건문을 CRI 방법으로 표현하면 식 (2)가 된다.

$$\begin{aligned} B' &= A' \circ (A \rightarrow B) \\ &= A' \circ R \end{aligned} \quad (2)$$

Max-Min CRI 방법은 퍼지 조건 연산자  $R_8$ 을 포함하고 있는데,  $R_8$ 은 max, min연산에 'ALSO' 연산으로 합연산을 이용하는 퍼지 조건 연산자이다.

Max-Min CRI 방법은 지금까지 연구된 R8의 성능 분석을 통해 타당성을 입증할 수 있다. Kiszka[3]는 여러 퍼지 조건 연산자를 직류 계열 모터에 적용한 결과,  $R_8$ 이 'ALSO'연산을 합연산으로 할 때 원하는 출력값과 실제 출력값과의 차가 0.02173으로 다른 퍼지 조건 연산자에 비해 가장 작기 때문에 매우 효율적인 연산자라고 하였다. 그러나 Kiszka는  $R_8$ 이 'ALSO' 연산을 곱연산으로 하면(이 경우  $R_8^*$ 로 표기함) 그 차이가 1.0이 되어 매우 나쁜 결과를 보인다고 하였다. 또한 Kandel[1]은 여러 규칙 베이스를 가진 FLC 모터 모델과 데이터를 이용하여  $R_8$ 의 성능을 실험하였는데, 결과 다른 퍼지 조건 연산자에 비해 성능이 우수함을 입증하였다. Mizumoto[9]는 퍼지 추론을 위한 여러 기준들을 제시하여 이를 이용하여 GMP와 GMT의 경우 추론 결과를 보여 주었다. 그 결과  $R_8$ 이 GMP경우 추론 기준에 대해 약 43%의 만족율을 보여 다른 것들에 비해 나쁘지 않았으나 GMT인 경우는 약 20%의 저조한 만족율을 보였다. 또한 Lee[5]는  $R_8$ 이 잘 정의된 논리 구조는 갖지 않지만 그 결과는 특히 GMP인 경우, 근사 추론에 매우 적당하다고 하였다.

그러므로 Mamdani의 퍼지 조건 연산자 R8이 다른 퍼지 조건 연산자들에 비해 max, min만을 이용하므로 간단하면서도 성능이 매우 우수하다. 따라서 많은 연구자들이 max-min CRI 방법을 이용하여 하드웨어를 설계한다든지 칩을 설계하여 공학 시스템 등에 응용하고 있다[10,12,14].

Max-Min CRI 방법은 간단하고 구현하기 쉬우나 규칙 베이스의 임의의 한 규칙의 조건부 퍼지 명제와 입력 사실이 일치해도 그 규칙의 결론부에 해당되는 결론을 추론해 내지 못한다는 문제점을 갖고 있다. 그러나 이 문제는 GMP의 합성 순서에 관계된 것이므로 Takefuji[12]가 제안한 GMP합성 순서를 이용하면 어느 정도 해결할 수 있다. 이것이 일치도를 고려한 GMP 방법이다. 일치도  $\alpha$ 는 다음 식 (3)과 같은 방법으로 측정한다.

$$\alpha = \max \min(\mu_{A'}(u), \mu_A(u)) \quad (3)$$

따라서 일치도  $\alpha$ 에 의한 Max-Min CRI 방법은 식 (4)와 같다.

$$\begin{aligned} \forall v \in V, u \in U, \mu_{B'}(v) &= \max \min(\mu_{A'}(u), \mu_R(u, v)) \\ &= \min(\alpha, \mu_B(v)) \end{aligned} \quad (4)$$

그런데 Max-Min CRI 방법에는 다음과 같은 문제점이 있다. 첫째, 멤버십 함수가 주관적으로 결정된다. 둘째, 오류를 발생시키는 가중치 전략을 사용한다. 셋째, 비합리적으로 추론 규칙들을 조합한다는 점이다.

이와 같은 문제점 때문에 “If <조건부 퍼지 명제> Then <결론부 퍼지 명제>” 형태의 규칙 베이스에서 임의의 규칙의 조건부와 동일한 입력 사실이 주어져도 그 규칙의 결론부와 동일한 결론을 생성해 내지 못한다.

따라서 본 논문에서는 Max-Min CRI 방법의 문제점을 해결하고 FLC의 성능을 개선하기 위해 신경 회로망[7]과 퍼지 이론의 장점을 융합한 지능적인 INFS(Intelligent Neuro-Fuzzy System model)을 제안하고자 한다. INFS는 신경 회로망의 학습 능력과 퍼지 이론의 추론 능력을 융합한 지능적인 시스템이다. 여기서 지능적이란 자기 조절 기법들이 융합된 진정한 의미에서의 적응적 FLC 시스템을 의미한다.

INFS는 지식 획득 단계(Knowledge Acquisition Phase), 적응 조절 단계(Adaptation Phase)를 통해 Max-Min CRI 방법의 세가지 문제점을 해결하여 추론 단계(Inference Phase)에서는 임의의 입력 사실에 대해 실제 추론이 가능한 시스템이다.

INFS의 성능 평가를 위해 간단한 FLC인 직류 계열 모터(Direct Current series motor)에 적용하여 실험 결과를 토의하고자 한다.

본 논문의 구성은 다음과 같다. II장에서는 신경 회로망과 퍼지 이론의 특성들을 비교하고 뉴로-퍼지 융합 시스템의 형태를 분류한다. III장에서는 서론에서 제시한 문제점을 해결하기 위한 지능적인 뉴로-퍼지 시스템(INFS)을 설계하고 시스템의 기능을 설명한다. IV장에서는 직류 계열 모터에 적용하여 성능 평가를 하고 V장에서 결론을 맺고자 한다.

## II. 뉴로-퍼지 시스템

### 1. 신경 회로망과 퍼지 이론의 비교

뉴로-퍼지 시스템의 융합에 관한 개념을 확립한 대표적인 학자로는 Kosko와 Takagi를 들 수 있다.

Kosko[4]는 지식의 형태를 구조적 지식과 비구조적 지식으로 나누고 지식 표현 형태도 심볼릭과 수치적으로 나누어 여러 시스템들의 지식 처리 형태를 분류하였다. Kosko는 AI 전문가 시스템은 구조적 지식을 심볼릭하게 처리하고, 퍼지 시스템은 구조적 지식을 수치적으로 표현하며, 신경 회로망 시스템은 비구조적 지식을 수치적으로 표현하는 시스템이라 하였다. 그러나 그는 인간의 지식 정보 처리 과정은 어느 시스템에도 국한된 것이 아니라 여러 지식 정보 처리의 통합체라고 볼 수 있기 때문에 신경 회로망과 퍼지 시스템의 융합의 중요성은 매우 크다고 하였다.

Takagi[11]는 뉴로-퍼지 시스템은 신경 회로망과 퍼지 이론의 장점과 유사점들을 융합하려는 노력에서 부터 시작되었다고 하였다.

위의 결과들을 종합해서 본 논문에서는 다음 표 1에 신경 회로망과 퍼지 이론의 장단점 및 유사점들을 비교 정리하였다.

표 1. 신경 회로망과 퍼지 이론

| 특성 분류 기준      | 신경 회로망   | 퍼지 이론  |
|---------------|--|--|
| 지식 표현         | 분산 표현  | 멤버십 함수에 의한 표현  |
| 결합내구성         | 있 음  | 있 음  |
| 불완전한 자료 처리 능력 | 있 음  | 있 음  |
| 추론 연산         | 학습함수에 의한 자기조정 능력 반복 학습<br>뉴런들의 다중합 연산              | 추론 규칙들의 조합에 의한 방법<br>멤버십 함수의 MAX-MIN연산                 |
| 기능            | 특성 추출 능력 최적화 기능<br>연상 기억 기능<br>다중센서 오토마타 비선형적 매핑   | 단순 계산 과정<br>언어적 또는 수치적 값의 처리 능력<br>비퍼지화 기능 비선형적 멤버십 함수 |
| 단점            | 학습 시간 소모   | 주관적인 멤버십 함수  |
| 응용 영역         | 패턴 인식(문자, 이미지, 음성 등), 제어, 비전, 의사 결정 시스템, 전문가 시스템 등 |  |

표 1에서 알 수 있듯이 신경 회로망은 학습 기능이 있어서 기대 이상의 훌륭한 문제 해결책을 제시해 줄 수 있으나 학습 시간이 오래 걸린다는 단점을 갖고 있다. 반면 퍼지 이론은 신경 회로망이 갖지 못한 독특한 추론 기능이 있어서 절차적 추론이 가능하나 멤버십 함수가 주관적으로 결정된다는 단점을 갖고 있다. 또한 이 두 분야의 응용 영역도 패턴 인식, 제어 시스템 등으로 서로 중복하고 있다. 따라서 서로의 장단점을 보완하면서 문제를 해결한다면 매우 강력한 도구가 될 수 있다.

## 2. 뉴로-퍼지 시스템의 형태 분류

뉴로-퍼지 시스템의 형태를 분류하기 전에 순수한 신경 회로망 모델(비퍼지 뉴런 모델)과 기본적인 퍼지 뉴런 모델(또는 뉴로-퍼지 모델)을 보면 그림 2와 같다.

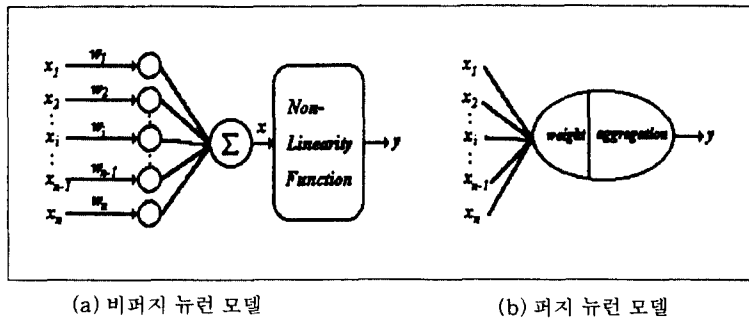


그림 2. 기본적인 뉴로-퍼지 모델

그림 2(a)는 McCulloch와 Pitts[8]가 약 40여년 전에 제안한 모델로, N개의 입력을 가진 뉴런들이 노드로 전달되어 가중치 합 연산 후 비선형 방정식을 통해 결과가 전달되는 모델을 나타낸 비퍼지 뉴런 모델이다. 각 뉴런들은 미리 정의된 내부 역치(threshold)를 갖는데 이 값을 넘어야 뉴런이 화이어 한다. 그림 2(b)는 (a)와 기능은 같으나 뉴런이 퍼지한 성질을 갖는다는 점과 가중치 합이 더해지는 것이 아니라 조합(aggregate)된다는 점이 차이점인 퍼지 뉴런 모델이다.

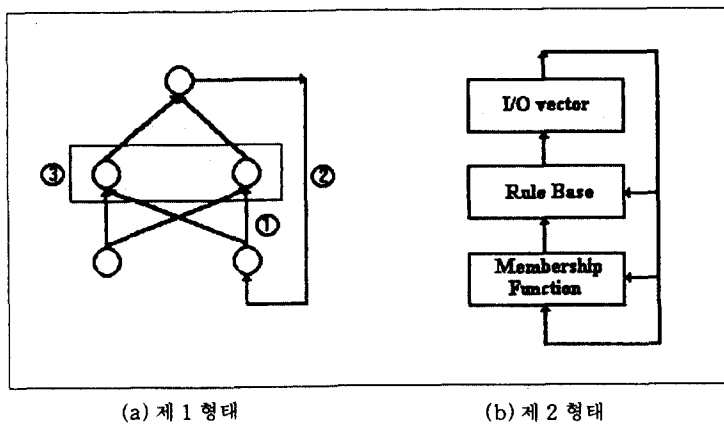


그림 3. 뉴로-퍼지 시스템의 형태 분류

이상의 특징들을 고려하여 본 논문에서는 뉴로-퍼지 시스템을 그림 3과 같이 크게 두가지 형태의 시스템으로 분류하였다.

그림 3(a)의 제 1형태는 신경 회로망 구조에 퍼지 이론을 도입한 시스템으로, 융합 방식에 따라 지식 표현시 연결 부분이나 가중치값에 멤버십 함수를 적용하는 연구(①)와 신경 회로망의 출력 결과를 퍼지 추론에 이용하는 연구(②), 그리고 신경 회로망의 일부 기능을 퍼지 논리 체계로 대체하려는 연구(③)로 구분된다.

그림 3(b)의 제 2형태는 퍼지 추론 구조에 부분적으로 신경 회로망을 도입한 시스템으로, 융합 방식에 따라 멤버십 함수의 생성과 적응적 조절에 관한 연구, 규칙 획득 및 규칙 베이스의 구성과 조절에 관한 연구, 바람직한 입출력 벡터쌍을 찾아내는 연구, 그리고 신경 회로망 구조로써 퍼지 추론 구조를 표현하는 연구 등으로 구분된다.

최근에는 제 1, 2형태의 융합 방식을 모두 통합된 구조로 융합하고 있는데, Lin[6]은 5층으로된 신경 회로망 구조를 갖는 퍼지 논리 시스템을 구축하여 제어 및 의사 결정 시스템에 응용하여 우수한 성능을 입증하였다. 또한 Gupta[2]는 입력과 출력은 퍼지 논리 시스템이며 내부 추론은 신경 회로망의 계층화된 구조에 의해 이루어지는 뉴로-퍼지 시스템을 구축하여 모델링과 제어 시스템에 응용하여 역시 좋은 결과를 보여 주었다.

그러나 어느 뉴로-퍼지 시스템도 Max-Min CRI 방법의 문제점을 모두 해결해 주지는 못하고 있다.

### III. 지능적인 뉴로-퍼지 시스템(INFS)의 설계

#### 1. 개요

INFS는 전문가에 의해 주관적으로 정의된 멤버십 함수로부터 초기 규칙 베이스를 구성한 후 원하는 입출력 쌍을 가장 잘 표현하는 최적의 규칙 베이스를 신경 회로망을 이용해 재구성하는 지능적인 시스템 모델이다. INFS는 신경 회로망의 학습 기능과 퍼지 이론의 추론 기능의 장점만을 융합하여 특성을 살림으로써 두 분야의 단점을 보완해 주는 뉴로-퍼지 시스템이다.

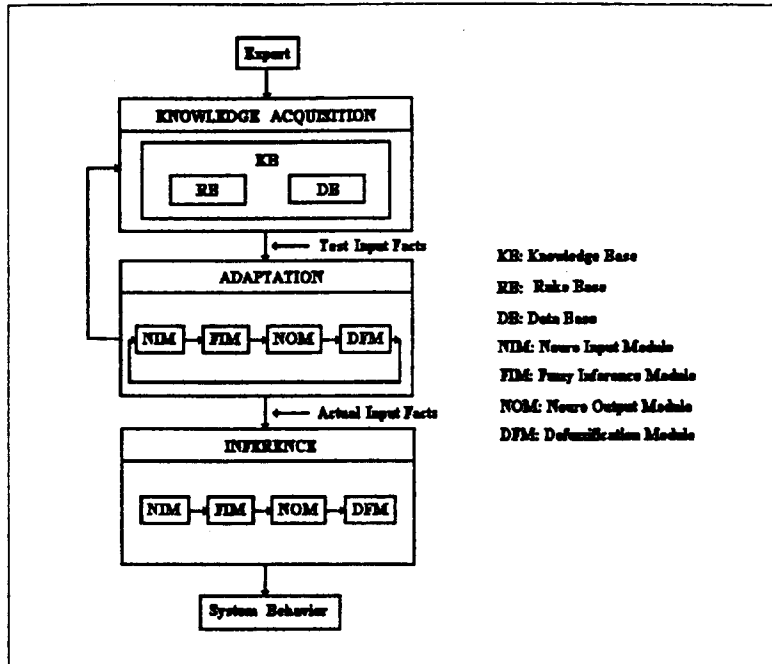


그림 4. INFS의 전체 구조

INFS는 문제 영역에 대한 최종적 시스템 행동을 결정짓기 위해 지식 획득 단계(Knowledge Acquisition Phase), 적응 조절 단계(Adaptation Phase), 추론 단계(Inference Phase)의 3단계의 수행 과정을 거친다.

다음 그림 4는 INFS의 전체 시스템 구조를 나타낸 그림이다.

지식 획득 단계는 전문가에 의해 주관적으로 정의된 멤버십 함수값을 데이터 베이스에 저장하고 이로부터 규칙 베이스를 구성하여 주어진 문제 영역에 대한 초기 지식 베이스를 구성하는 단계이다. 따라서 멤버십 함수의 조절로부터 데이터 베이스가 조절되며 이로부터 규칙 베이스가 조절되어 결국 지식 베이스가 조절될 수 있는 것이다. 그러므로 본 논문에서는 지식 베이스의 조절이란 용어와 규칙 베이스 및 멤버십 함수의 조절이란 용어는 같은 의미로 사용하겠다.

적응 조절 단계는 이러한 초기 규칙 베이스를 문제 영역에 적용된 최적의 규칙 베이스로 재구성하기 위한 단계이다. 이 단계에서는 규칙의 조건부 멤버십 함수와 일치하는 입력 사실에 대해 그 규칙의 결론부 멤버십 함수와 일치하는 출력 결과가 산출될 수 있도록 EBP 학습 알고리즘을 이용해 반복적으로 적응 조절한다. 조절 방법은 규칙의 조건부 멤버십 함수를 나타내는 벡터 표현을 학습하여 적응하고 올바른 결과가 나올 수 있도록 조절하여 결국 최적의 규칙 베이스를 재구성하는 과정을 반복한다. 여기서 적응은 학습이 끝난 상태를 말하고 조절은 변경된 값을 재학습하여 문제 영역에 가장 적합한 가중치를 생성한 상태를 의미하는데, 실제로는 이 둘이 분리된 것이 아니라 동시에 이루어진다.

추론 단계에서는 지식 획득 단계와 적응 조절 단계를 거쳐서 문제 영역에 적용된 최적의 지식 베이스가 형성되었을 때 필요시 실제로 추론을 행하는 단계이다. 그러므로 문제 영역에 대한 임의의 입력 사실에 대해 가장 적합한 시스템 행동을 결정짓게 된다.

INFS의 적응 조절 단계에는 뉴로 입력 모듈(Neuro Input Module : NIM), 퍼지 추론 모듈(Fuzzy Inference Module : FIM), 뉴로 출력 모듈(Neuro Output Module : NOM), 그리고 비퍼지화 모듈(Defuzzification Module : DFM)이 유기적으로 작동한다.

NIM에서는 원하는 출력값을 가장 잘 산출할 수 있는 규칙의 조건부 멤버십 함수를 학습하여 적용한 후 나중에 DFM의 출력을 입력으로 하여 주관적 멤버십 함수가 객관적이 되도록 반복 조절한다. 이때 EBP 학습 알고리즘을 이용한다. 따라서 기존의 Max-Min CRI 방법의 첫번째 문제점인 주관적인 멤버십 함수 결정 문제를 해결할 수 있다. NIM의 출력값은 조절된 조건부 멤버십 함수이며 이로부터 조절된 규칙 베이스를 갖고 FIM을 시작한다. FIM에서는 두번째 문제점인 오류 발생 가중치 전략 문제를 해결하기 위해 두 퍼지 집합간의 유사도를 정의하고 이를 이용한 새로운 추론 방법인 NCRI(New CRI) 방법을 이용한다. FIM의 출력값은 각 규칙마다 퍼지 추론한 결과인데 최종 시스템 행동 결정시 결론부 멤버십 함수의 반영 정도가 되는 값이다. 이 값은 NOM에 입력으로 사용된다. NOM에서는 원하는 출력값과 실제 출력값과의 차이를 최소화하기 위해 EBP 학습 알고리즘에 의해 규칙의 결론부 멤버십 함수의 반영 정도를 조절한다. DFM에서는 조절된 반영 정도로 부터 특정값(규칙 선택 기준) 이상인 규칙들만을 선택적으로 조합하는 비퍼지화 전략을 채택함으로써 세번째 문제점인 비합리적인 추론 규칙들의 조합 문제를 해결할 수 있다. 그러므로 INFS는 Max-Min CRI 방법의 세가지 문제점을 동적으로 해결할 수 있는 시스템 모델이다.

## 2. 각 단계별 특징

### 2.1 지식 획득 단계(Knowledge Acquisition Phase)

전문가로부터 얻은 지식을 바탕으로 다음 형태의  $n$ 개의 규칙을 갖는 초기 규칙 베이스를 구성한다. 각 규칙은 적당한 여러 멤버십 함수를 갖는다. 이때 전문가가 원하는 입출력 쌍에 대한 데이터도 함께 얻는다.

“IF  $X = F(i)$  and .... and  $Y = G(i)$  THEN  $Z = H(i)$ ”

여기서  $F(i)$ ,  $G(i)$ ,  $H(i)$  등은 멤버십 함수로 표현되어 데이터 베이스에 저장된다. 이때 멤버십 함수는 그림 6과 같은 삼각형 형태의 벡터로 표현하는데, 이 벡터값은 적응 조절 단계에서 원하는 출력값을 산출할 수 있도록 반복

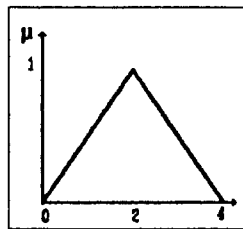
적으로 적용 조절된다.

2.2 적응 조절 단계(Adaptation Phase)

획득된 지식을 바탕으로 규칙 베이스와 멤버십 함수를 학습하여 적용하고 원하는 출력값을 산출할 수 있도록 조절하는 단계로, INFS에서 가장 중요한 핵심 단계이다. 이를 위해 규칙의 조건부를 입력 사실(test input facts)로 하여 적응 조절 과정을 반복한다.

(1) NIM(Neuro Input Module)

NIM은 EBP 학습 알고리즘을 이용해 규칙의 조건부 멤버십 함수를 학습하고 조절하는 모듈이다. 이때 각 멤버십 함수를  $F$ 라 하면, 벡터값  $V_F = (v_{F1}, v_{F2}, v_{F3})$ 로 표현한다. 여기서  $v_{F1}$ 은 멤버십 함수  $F$ 의 멤버십 값이 0( $\mu=0$ )인 왼쪽 점을,  $v_{F2}$ 는 멤버십 값이 1( $\mu=1$ )인 중심 점을,  $v_{F3}$ 는 멤버십 값이 0( $\mu=0$ )인 오른쪽 점을 각각 의미한다. 예를 들면, 언어적 변수 'ZERO'를 나타내는 멤버십 함수는 그림 5(a)와 같이 표현하며 그림 5(b)에서 처럼 벡터로 표현한다.  $v_{F1}, v_{F2}, v_{F3}$  값은 적응 조절 단계에서 원하는 출력값을 산출할 수 있도록 반복적으로 학습에 의해 적응되고 조절된다.



$V_{ZERO} = (0, 2, 4)$

(a) (b)

그림 5. 'ZERO'의 멤버십 함수와 벡터 표현

NIM은 그림 6과 같이 세층의 신경 회로망으로 구성된 학습 모듈(LEarning Module: LEM)과 두 층으로 구성된 조절 모듈(ADjust Module: ADM)로 구성되어 있다. LEM은 EBP 학습 알고리즘을 이용해 원하는 출력값과 그 출력값의 산출 근거가 되는 규칙의 조건부 멤버십 함수와의 관계를 학습하여 적용한다.

멤버십 함수를 조절하기 위해 학습 후 DFM의 비퍼지화된 출력값을 NIM의 LEM에 적용시켜 출력층의 출력 벡터를 생성한 다음 ADM에서 조절된다.

ADM의 알고리즘은 다음과 같다.

입력층: LEM의 정규화된 출력 벡터값을 입력으로 한다.

$$I = (i_1, i_2, \dots, i_k, \dots, i_n), (k = \text{규칙의 조건부 멤버십 함수}, n = \text{규칙의 수}) \tag{5}$$

출력층: 출력층은 조절된 멤버십 함수 벡터  $V_{kF}$ 을 생성하는 층이다. 따라서  $i_k$ 가 최대 벡터라 하면,  $k$ 번째 규칙의 조건부 멤버십 함수의 벡터값  $V_{kF}$ 은 다음과 같이 조절된다.

$$V_{kF}' = KV_{kF} \tag{6}$$

여기서,  $K = \begin{cases} i_k & (i_k \text{인 벡터}) \\ 1 + i_k & (i_k \text{의 오른쪽에 있는 벡터}) \\ 1 - i_k & (i_k \text{의 왼쪽에 있는 벡터}) \end{cases}$

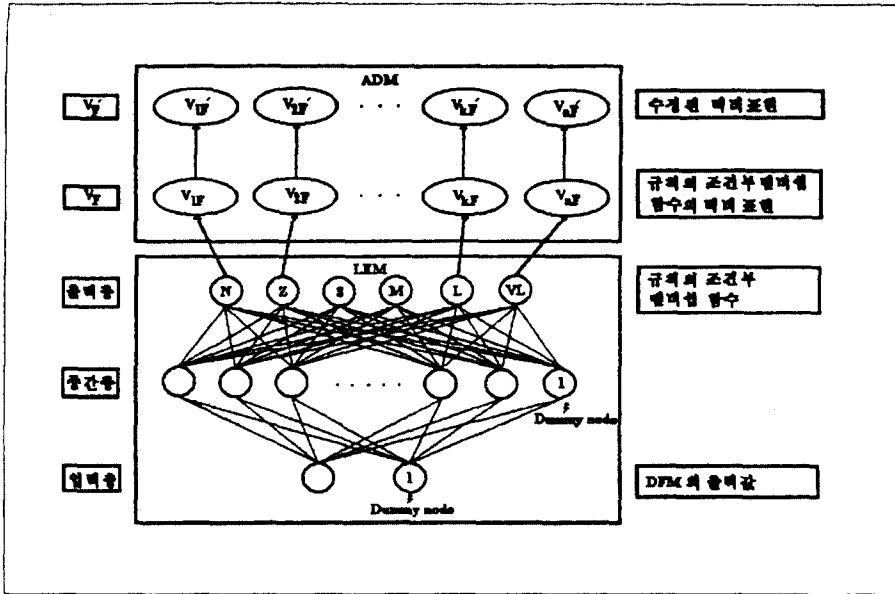


그림 6. NIM(Neuro Input Module)

이렇게 조절된 멤버십 함수를 FIM에 입력으로 하여 규칙 베이스를 재구성하고 유사도를 이용한 NCRI 방법을 적용한다.

(2) FIM(Fuzzy Inference Module)

FIM은 NIM에서 조절된 멤버십 함수로부터 재구성된 규칙 베이스를 가지고 추론을 수행하는 모듈이다. FIM에서는 기존의 Max-Min CRI 방법을 개선한 새로운 방법을 이용하여 추론하는데 이것을 NCRI(New max-min CRI)라고 명명한다.

NCRI 방법은 퍼지 집합간의 유사성을 측정하는 유사도를 정의하여 Max-Min CRI 방법에 적용함이 차이점이다.

본 논문에서 제안한 유사도를 정의하면 【정의 1】과 같다.

【정의 1】 유사도(Similarity Measure)

퍼지 집합 A, A'의 유사도는 모든  $u(u \in U)$ 에 대해 다음과 같이 측정한다.

이때  $\tau$ 는 측정된 유사도 값이다. ( $\wedge$ : min,  $\vee$ : max)

$$SM(A, A') = \tau = \frac{\int (\mu_A(u) \vee \mu_{A'}(u)) du}{\int (\mu_A(u) \wedge \mu_{A'}(u)) du}$$

유사도를 이용한 NCRI 방법에서의 추론 방법은 식 (7)과 같다. 주어진 규칙 베이스가 있고,  $\tau$ 를 측정된 유사도 값,  $\alpha$ 를 일치도라 하면 추론된 결과값은  $\mu_B$ 가 된다.

$$\begin{aligned} \forall v \in V, u \in U, \mu_B(v) &= \max \min(\mu_{A'}(u), \mu_R(u, v)) \\ &= \min(\alpha, \mu_B(v)) \times \tau \end{aligned} \tag{7}$$



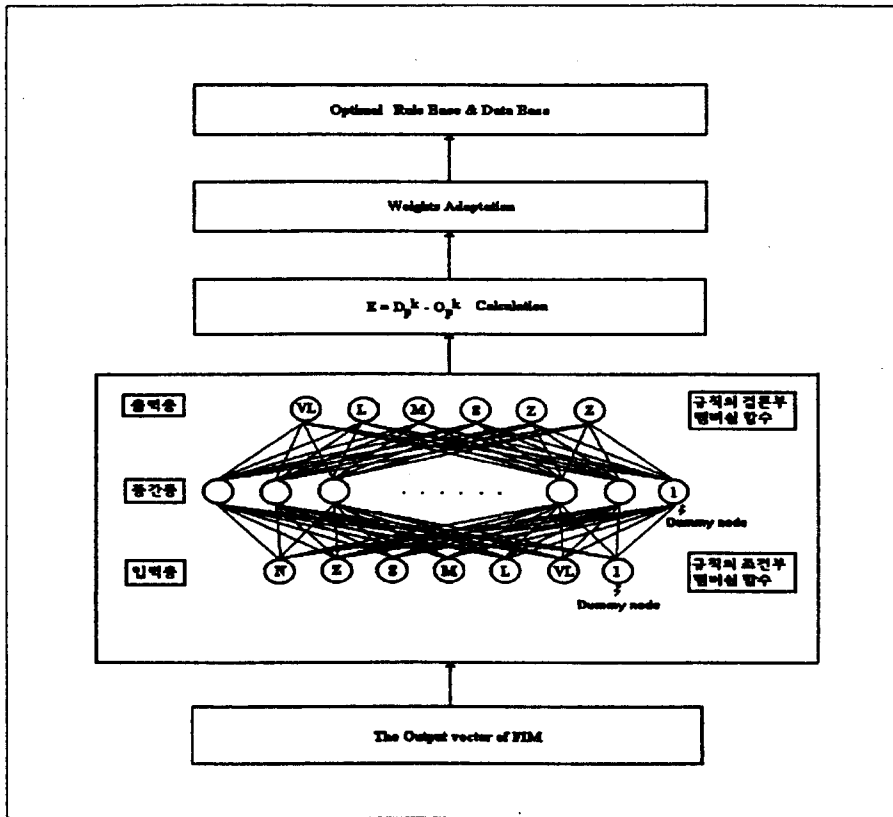


그림 8. NOM(Neuro Output Module)

FIM의 출력은 각 규칙의 결론부 멤버십 함수들의 반영 정도(또는 가중치)이며 이 값들은 NOM에 입력으로 이용된다.

(3) NOM(Neuro Output Module)

NOM은 EBP 학습 알고리즘을 이용해 규칙의 조건부 멤버십 함수와 대응되는 결론부 멤버십 함수와의 관계를 학습하는 모듈이다. 이때 출력층의 출력값과 원하는 출력값과의 차가 최소가 될 때까지 반복적으로 학습한다.

NOM은 그림 8과 같이 5단계 과정으로 이루어져 있다. FIM의 출력값은 입력층, 중간층, 출력층의 세층으로 구성된 신경 회로망에 입력된다. 그리고 EBP에 의해 학습하고, E를 계산하여 연결선의 가중치가 조절되어 결국 최적의 규칙 베이스와 멤버십 함수를 구성하게 된다.

NOM은 다음 식 (8)의 값을 최소로 하려는데 목적이 있다. 이 식에서 원하는 출력값인  $D_{pk}$ 는 패턴 P의 k번째 출력 벡터 값이고  $O_{pk}$ 는 실제의 출력값을 의미한다.

$$E = \sum_p E = \frac{1}{2} \sum_p \sum_k (D_{pk} - O_{pk})^2 \tag{8}$$

<Forward Path>

입력층: 입력 벡터는 퍼지 추론을 통한  $\theta \times \tau$  값으로 한다. 패턴 p에 대한 각 입력 노드의 입력 벡터  $I_p$ 는 다음과

같다. 각 노드의 출력값을  $O_{pi}^l$ 로 표시한다. 여기서 1은 노드 오프셋을 위한 dummy 노드이다.

$$I_p = (i_{p1}', i_{p2}', \dots, i_{p2}', \dots, i_{pn}', 1), (i = \text{규칙의 조건부 멤버쉽함수}) \quad (9)$$

**연결선 1:** 연결선은 노드 i와 노드 j간의 연결 정도를 나타내는 선으로 가중치로 표시한다.  $w_{ij}$ 는 학습에 의해 값이 수정되는데 초기값은  $[-0.1, 0.1]$ 사이의 값으로 한다.

**중간층:** 중간층의  $net_{pj}^h, f_j^h, O_{pj}^h$ 는 다음과 같다. 여기서 노드 오프셋( $\Theta$ )은 입력층에 항상 1 값을 갖는 dummy 노드를 뒀으로써 결정된다.

$$net_{pj}^h = \sum_{i=1}^n w_{ij} \times O_{pi}^l, (i = 1..n, j = 1..m, j = \text{중간층 노드}) \quad (10)$$

$$f(\alpha) = \frac{1}{1 + \exp^{-(\alpha - \Theta)}} \quad (11)$$

$$O_{pj}^h = f(net_{pj}^h) = \frac{1}{1 + \exp^{[-(\sum w_{ij} \times O_{pi}^l) + \Theta]}} \quad (12)$$

**연결선 2:** 연결선은 노드 j와 노드 k간의 연결 정도를 나타내는 선으로 가중치  $w_{ij}$ 로 표시한다.  $w_{ij}$ 는 학습에 의해 값이 수정되는데 초기값은  $[-0.1, 0.1]$ 사이의 값으로 한다.

**출력층:** 출력층의  $net_{pk}^o, f_k^o, O_{pk}^o$ 는 다음과 같다.

$$net_{pk}^o = \sum_{j=1}^m w_{jk} \times O_{pj}^h, (j = 1..m, k = 1..l, k = \text{규칙의 결론부 멤버쉽함수}) \quad (13)$$

$$f(\alpha) = \frac{1}{1 + \exp^{-(\alpha - \Theta)}} \quad (14)$$

$$O_{pk}^o = f(net_{pk}^o) = \frac{1}{1 + \exp^{[-(\sum w_{jk} \times O_{pj}^h) + \Theta]}} \quad (15)$$

<Backward Path>

**출력층:** 원하는 출력 벡터를 준 후 실제 출력 벡터와의 차로부터 출력층의 오류값을 계산한다.

$$D_p = (D_{p1}, \dots, D_{pk}) \quad (16)$$

$$\begin{aligned} \delta_{pk} &= -(D_{pk} - O_{pk}^o) f_k'(net_{pk}^o) \\ &= (D_{pk} - O_{pk}^o) O_{pk}^o (1 - O_{pk}^o) \end{aligned} \quad (17)$$

**연결선 2:** 다음 학습 규칙에 따라 연결 가중치를 수정한다.

$$\begin{aligned} w_{jk}(t+1) &= w_{jk}(t) + \eta (D_{pk} - O_{pk}^o) f_k'(net_{pk}^o) O_{pj}^h \\ &= w_{jk}(t) + \eta (D_{pk} - O_{pk}^o) O_{pk}^o (1 - O_{pk}^o) O_{pj}^h \\ &= w_{jk}(t) + \eta \delta_{pk} O_{pj}^h \end{aligned} \quad (18)$$

**중간층:** 중간층에 대한 오류값을 계산한다.

$$\delta_{pj} = O_{pj}^h(1 - O_{pj}^h) \sum_k \delta_{pk} W_{jk} \quad (19)$$

**연결층 1:** 다음 학습 규칙에 따라 연결 가중치를 수정한다.  $\beta$ 는 학습 속도와 관계된 모멘텀 상수로 임의의 실수 값을 갖는다.

$$\begin{aligned} w_{ij}(t+1) &= w_{ij}(t) + \eta O_{pj}^h(1 - O_{pj}^h) \sum_k \delta_{pk} w_{jk} \times O_{pi}^l + \beta(w_{ij}(t) - w_{ij}(t-1)) \\ &= w_{ij}(t) + \eta \delta_{pj} O_{pi}^l + \beta(w_{ij}(t) - w_{ij}(t-1)) \end{aligned} \quad (20)$$

여러 패턴에 대해  $E = \frac{1}{2} \sum_p \sum_k (D_{pk} - O_{pk}^o)^2$  이 허용치내에 들어오면 학습을 멈추고, 그렇지 않으면 이 과정을 반복한다. NOM의 출력값은 DFM의 입력으로 이용된다.

#### (4) DFM(Defuzzification Module)

DFM은 NOM에서의 출력값인 결론부 멤버십 함수들의 반영 정도를 입력으로 받아 최소 반영 정도 이상인 멤버십 함수들만을 조합하는 비퍼지화 과정을 수행하는 모듈이다.

이때 최소 반영 정도를 규칙 선택 기준(Rule Selection Level : RSL)이라 한다. 즉, RSL 이상인 출력 벡터 값을 이용해 다음 식 (21)의 방법에 따라 추론 결과를 산출한다. RSL은 신경 회로망을 이용해 정의할 수 있으나 본 논문에서는 출력 벡터들의 평균값을 이용한다.

따라서 합리적으로 관련된 규칙들만을 선별하여 조합할 수 있으므로 규칙 조합 시간이 줄어들 뿐 아니라 Max-Min CRI 방법의 세번째 문제를 해결할 수 있다.

$$\text{출력값} = \sum COA_k \times \frac{O_{pk}^o}{\sum O_{pk}^o} \quad (21)$$

여기서  $\sum COA_k$ 는 출력층의 각 출력 노드의 무게 중심값이고,  $O_{pk}^o$ 는 각 노드의 출력 벡터값이고,  $\sum O_{pk}^o$ 는 전체 출력 벡터값을 의미한다.

### 2.3 추론 단계(Inference Phase)

적용 조절 단계를 거쳐 초기 규칙 베이스와 멤버십 함수로부터 문제 영역에 적용된 최적의 규칙 베이스와 멤버십 함수를 구성했다면 이로 부터 임의의 입력 사실에 대해 추론을 수행할 수 있다.

추론 단계는 적용 조절 단계에서 학습을 통해 문제 영역에 가장 적합 규칙 베이스와 멤버십 함수가 재구성되었기 때문에 별도의 학습 과정 없이 NIM → FIM → NOM → DFM 순으로 차례대로 진행하면 되는 것이다.

추론 과정은 NIM으로 부터 NCRI 방법을 적용하는 FIM을 통해 출력값을 NOM에 입력하여 RSL을 정하고, RSL에 따라 규칙 베이스에서 적당한 규칙들만을 선별하여 DFM을 거쳐 문제 영역에 대한 시스템의 행동을 결정짓는다.

## IV. 실험 및 결과 분석

본 논문에서는 제안된 INFS의 성능을 평가하기 위해 간단한 퍼지 제어기인 직류 계열 모터(D. C. series motor)를 실험 대상으로 이용하였다.

### 1. 실험 대상

직류 계열 모터는 FLC의 성능 평가를 위한 매체(vehicle)로써 제공되는 간단한 퍼지 제어기로, 그림 9와 같은 구조를 갖는다. 이 모터의 일반적인 수행 범위는 다음과 같다[3].

Power rating :  $P_N = 0.7 \text{ Kw}$   
 Voltage rating :  $U_N = 110 \text{ V}$   
 Rated current :  $I_N = 8.84 \text{ A}$   
 Rated speed :  $n_N = 1500 \text{ r.p.m.}$

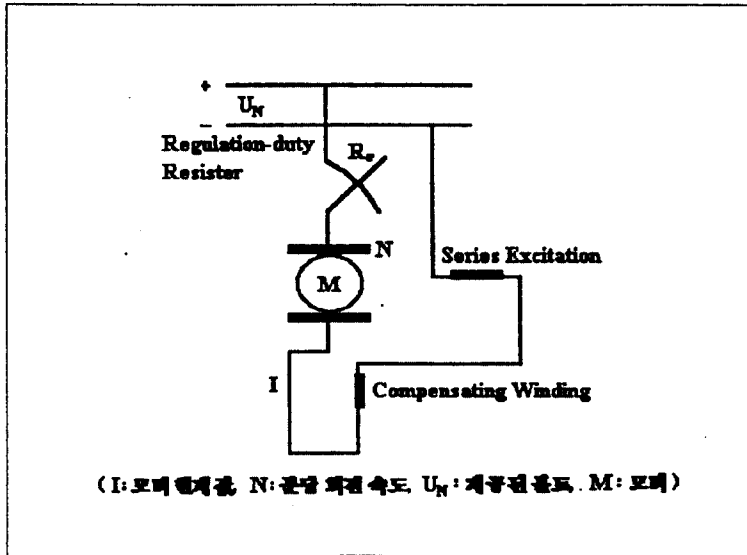


그림 9. 직류 계열 모터

이 모터의 전자적 유입량은 변화하는 현재값  $i$ 에 비례하며  $i$ 와 속도  $n$ 과는 비선형적 관계를 갖는다. 따라서 직류 계열 모터는 현재값  $i$ 와 정지 상태에서의 회전 속도값  $n$ 으로 부터 식  $n = f(i)$ 에 의해 모터의 실제 상태를 측정한다. 이때  $f$ 는 임의의 비선형 방정식이다.

직류 계열 모터가 작동하는 규칙 베이스 환경은 다음과 같다.

IF  $I = \text{null}$  then  $N = \text{very large}$   
 ALSO  
 IF  $I = \text{zero}$  then  $N = \text{large}$   
 ALSO  
 IF  $I = \text{small}$  then  $N = \text{medium}$   
 ALSO  
 IF  $I = \text{medium}$  then  $N = \text{small}$   
 ALSO  
 IF  $I = \text{large}$  then  $N = \text{zero}$   
 ALSO  
 IF  $I = \text{very large}$  then  $N = \text{zero}$

이 규칙 베이스에서는 전문가로 부터 얻은 규칙으로 모터의 현재값  $I$ 와 시간당 회전 속도  $N$ 을 나타내는 퍼지 집합을 얻어 데이터 베이스를 구성한다. 이 데이터는 kiszka의 데이터[3]를 이용한다.

이 규칙 베이스와 데이터 베이스로 부터 지식 베이스는 다음 그림 10과 같이 나타낼 수 있다.

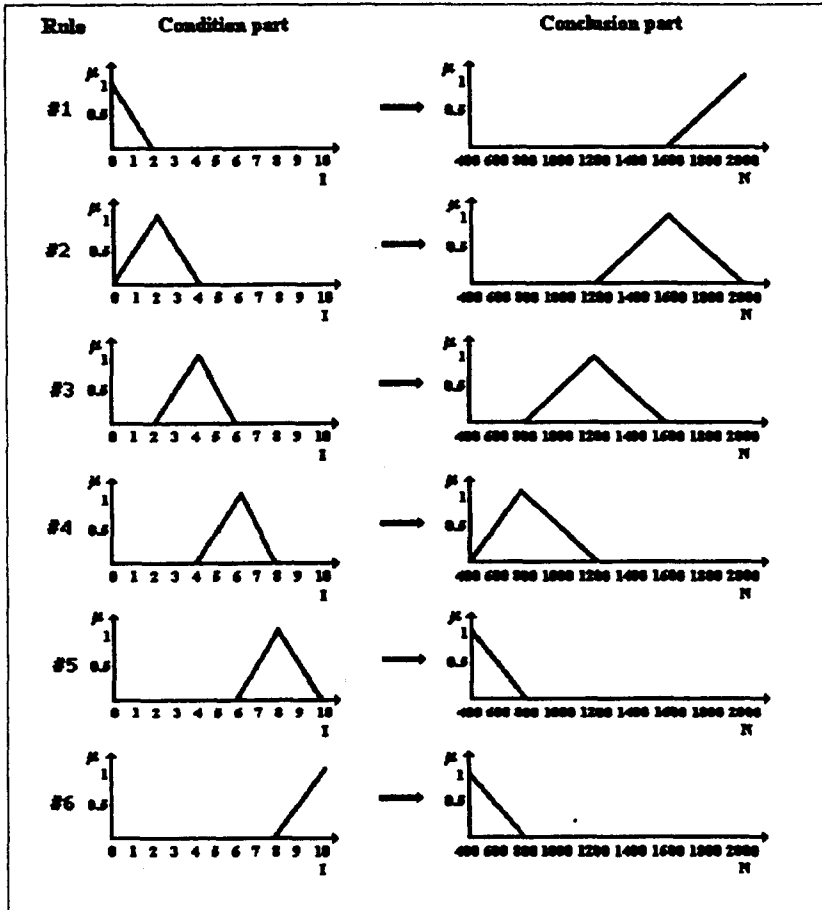


그림 10. 지식 베이스

2. 실험 결과 분석

본 절에서는 추론에 있어서 퍼지 추론 방법만을 사용하는 경우와 퍼지 추론 방법에 신경 회로망을 사용하는 경우의 추론 결과를 비교하고자 한다(그림 11과 그림 12 참조). 이때 퍼지 추론 방법은 Max-Min CRI 방법, Takefuji 방법, Turksen 방법 및 NCRI 방법을 각각 사용하였다. 여기서 Takefuji 방법 및 Turksen 방법은 Takefuji와 Turksen이 각각 제안한 유사도를 식 (7)의 대신 적용한 퍼지 추론 방법을 의미한다.

다음 식 (22)에서 Takefuji[12]가 제안한 유사도를, 식 (23)에서 Turksen[13]이 제안한 유사도를 소개한다.

$$- \text{Takefuji의 유사도} : \sigma = 1 - \left( \frac{d}{a+a'} \right) \tag{22}$$

$$\text{where, } d = \sum_{i=1}^m (\max(A_1(x_i), A'(x_i)) - \min(A_1(x_i), A'(x_i)))$$

$$a = \sum_{i=1}^m A_1(x_i), \quad b = \sum_{i=1}^m A'(x_i)$$

$A_1$ : 규칙의 조건부 멤버쉽 함수,  $A'$ : 입력 사실

$x_i$ : 기본 변수(base variable)

- *Turksen*의 유사도 :  $SM = 1 + \left( \frac{1}{DM} \right)$  (23)

where,  $DM = 1 - \max_{x \in X} \mu_A \cap_B(x)$

A, B: 퍼지 집합

x: 기본 변수(base variable)

본 실험에서는 퍼지 추론 방법 자체의 성능 비교도 수행되었다. 즉, 퍼지 추론 방법만을 사용하는 경우 NCRI 방법이 가장 우수했으며 그 다음이 Takefuji 방법이고 그 다음이 *Turksen* 방법이며 가장 성능이 떨어지는 방법은 유사도를 적용하지 않는 Max-Min CRI 방법으로 평가된다. 이 결과를 그림으로 나타내면 그림 11과 같다.

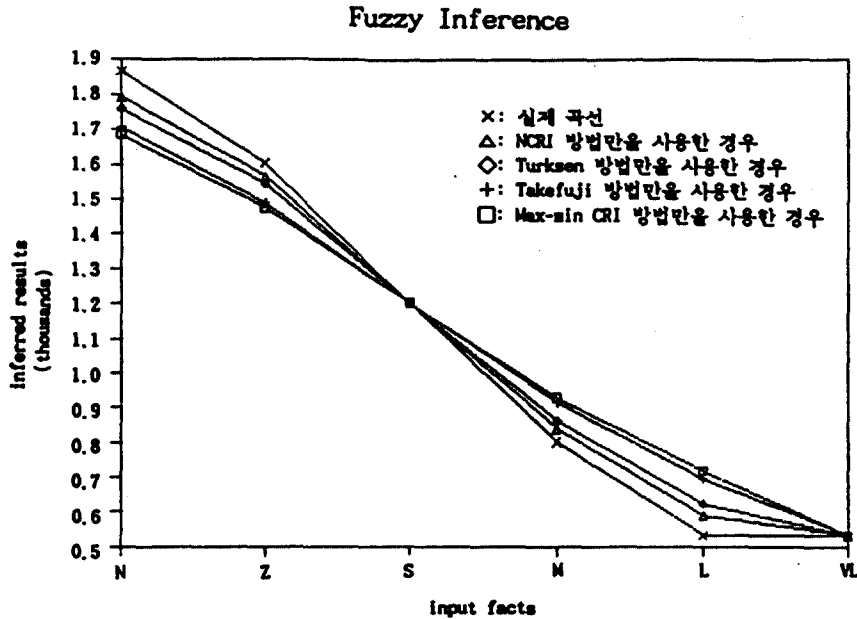


그림 11. 퍼지 추론 방법만을 사용하는 경우의 추론 결과

이때 실제 곡선과 각 그래프와의 차가 오류 영역을 나타내는 부분이다. 이 그림에서 NCRI 방법은 Max-Min CRI 방법에 비해 오류 영역을 최대 82.1%까지 줄여 주며, *Turksen* 방법에 비해서는 최대 74.6%까지, Takefuji 방법에 비해서는 최대 41.8%까지 줄여 준다.

한편, 퍼지 추론 방법에 신경 회로망을 사용하여 추론하는 경우의 추론 결과는 그림 12와 같다. 즉, 본 논문에서는 앞서 설명한 네가지 추론 방법을 각각 INFS에 적용하여 앞과 동일한 입력 사실들에 대해 추론하였다.

그림에서 보는 바와 같이 어떤 추론 방법을 사용하든지 INFS에서는 실제 곡선과 추론 결과값이 동일하다. 그 이유는 신경 회로망을 사용하여 실제 곡선과 같은 곡선이 나오도록 학습을 충분히 하였기 때문이다. 또 다른 이유는 실험한 INFS가 어떤 추론 방법을 사용하든지 출력 벡터값 중 평균값인 RSL이상인 벡터만을 조합하는 비퍼지화 전략을 채택하고 있기 때문이다. 따라서 학습된 입력 사실인 경우 당연히 가장 높은 벡터값으로 추론 결과가 산출될 수밖에 없다.

### Neuro-Fuzzy Inference

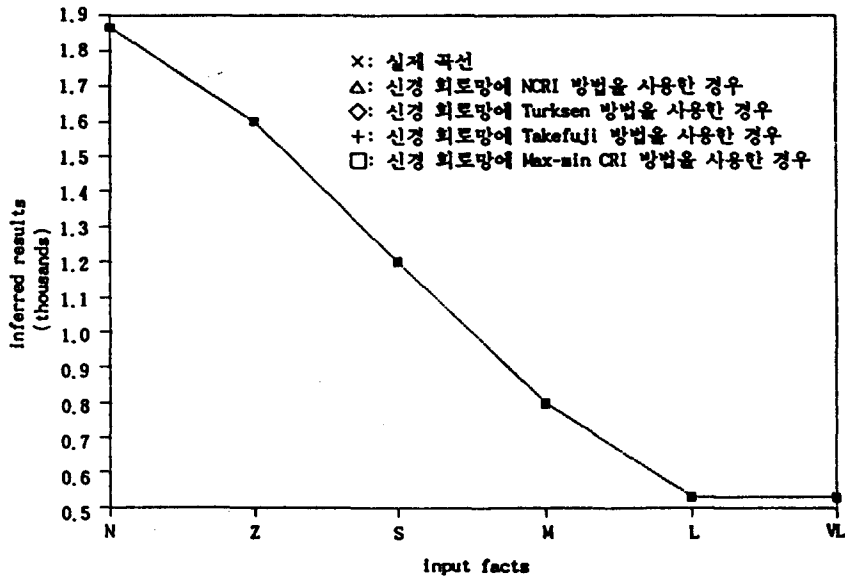


그림 12. 퍼지 추론 방법에 신경 회로망을 적용한 경우의 추론 결과

따라서 그림 11과 그림 12를 비교해 보면, 퍼지 추론 방법에 신경 회로망을 적용시킨 경우가 매우 정확한 추론 결과를 생성하는 효율적인 뉴로-퍼지 시스템이 될 수 있다. 물론 이 경우도 추론 방법으로 NCRI 방법을 적용하였을 때가 가장 효율적이다. 실험한 INFS에서는 NCRI 방법이나 나머지 세 방법을 적용해도 모두 학습된 입력 사실에 대해서는 매우 정확한 값을 추론해 준다. 그러나 평균 학습 시간을 고려하였을 때, NCRI 방법을 이용한 INFS의 경우가 Takefuji, Turksen, Max-Min CRI 방법들을 이용한 INFS의 경우 보다 평균 학습 시간이 각각 4.76%, 28.63%, 40.67%정도 감소되는 결과를 얻었다. 이것은 NCRI 방법을 채택하는 경우가 가장 효율적임을 의미한다.

지금까지 결과를 종합하면, 신경 회로망의 사용 유무에 관계없이 NCRI 방법은 매우 효율적이며, 특히 NCRI 방법에 신경 회로망이 융합된 INFS 모델은 문제 영역에 매우 지능적으로 작동하는 시스템임을 알 수 있다.

그러나 다음 표 2에서 처럼 학습되지 않은 입력 사실에 대해서는 어느 추론 방법도 좋은 결과를 보여주지 못한다.

이 표에서 ()안의 값은 전문가가 원하는 출력값을 의미하며, 이 값과 가장 가까운 추론 결과값은 밑줄로 표기하였다. 즉, NCRI 방법을 이용하면 "not N"에 대해서만 좋은 결과를 보여 주며, Turksen 방법을 이용하면 "not S"인 경우, Takefuji 방법을 이용하면 "more or less N, more or less Z, not Z, not N, not L"인 경우, Max-Min CRI 방법을 이용하면 "not VL"인 경우 좋은 결과를 보여 주고 있다. 이외에도 very N, very Z, very S, more or less S, very M, more or less M, very L, more or less L, very VL, more or less VL 등은 어떤 추론 방법을 사용하더라도 동일한 결과를 보여 준다. 따라서 학습되지 않은 입력 사실에 대해서는 어떤 추론 방법도 효율적이지 못함을 알 수 있다. 그러나 발생 가능한 경우를 모두 학습한다면 NCRI 방법을 이용한 INFS가 가장 효율적인 방법이 될 것이다.

표 2. 학습되지 않은 입력 사실에 대한 추론 결과

| 입력사실                        | 추론방법 | Max-Min CRI        | Takefuji           | Turksen            | NCRI               |
|-----------------------------|------|--------------------|--------------------|--------------------|--------------------|
| very N(1899.999146)         |      | <u>1866.661499</u> | <u>1866.661499</u> | <u>1866.661499</u> | <u>1866.661499</u> |
| more or less N(1839.996338) |      | <u>1866.661499</u> | <u>1866.661621</u> | <u>1866.661499</u> | <u>1866.661499</u> |
| not N(1104.749756)          |      | <u>906.482361</u>  | <u>1085.531494</u> | <u>1050.927734</u> | <u>1045.970703</u> |
| very Z(1599.996826)         |      | <u>1599.999146</u> | <u>1599.999146</u> | <u>1599.999146</u> | <u>1566.999146</u> |
| more or less Z(1600.005371) |      | <u>1599.999146</u> | <u>1599.999268</u> | <u>1599.999146</u> | <u>1599.999146</u> |
| not Z(1066.652222)          |      | <u>687.721619</u>  | <u>880.198486</u>  | <u>710.958746</u>  | <u>819.135193</u>  |
| very S(1199.996216)         |      | <u>1200.002808</u> | <u>1200.002808</u> | <u>1200.002808</u> | <u>1200.002808</u> |
| more or less S(1200.002197) |      | <u>1200.002808</u> | <u>1200.002808</u> | <u>1200.002808</u> | <u>1200.002808</u> |
| not S(1199.980225)          |      | <u>1450.153442</u> | <u>985.353821</u>  | <u>1382.685791</u> | <u>970.444397</u>  |
| very M(800.000854)          |      | <u>800.003174</u>  | <u>800.003174</u>  | <u>800.003174</u>  | <u>800.003174</u>  |
| more or less M(800.008972)  |      | <u>800.003174</u>  | <u>800.003174</u>  | <u>800.003174</u>  | <u>800.003174</u>  |
| not M(1333.315918)          |      | <u>533.311768</u>  | <u>1166.783691</u> | <u>820.792786</u>  | <u>1127.245972</u> |
| very L(499.973053)          |      | <u>533.311768</u>  | <u>533.311768</u>  | <u>533.311768</u>  | <u>533.311768</u>  |
| more or less L(559.981628)  |      | <u>533.311768</u>  | <u>533.311768</u>  | <u>533.311768</u>  | <u>533.311768</u>  |
| not L(1295.234619)          |      | <u>1186.144043</u> | <u>1226.027588</u> | <u>1372.283203</u> | <u>1210.126343</u> |
| very VL(499.973053)         |      | <u>533.311768</u>  | <u>533.311768</u>  | <u>533.311768</u>  | <u>533.311768</u>  |
| more or less VL(559.981628) |      | <u>533.311768</u>  | <u>533.311768</u>  | <u>533.311768</u>  | <u>533.311768</u>  |
| not V(1295.234619)          |      | <u>1194.447510</u> | <u>1060.990356</u> | <u>800.003174</u>  | <u>1014.961060</u> |

(N : Null, Z : Zero, S : Small, M : Medium, L : Large, VL : Very Large)

## V. 결 론

기존의 추론 방법인 Max-Min CRI 방법은 멤버십 함수가 주관적으로 결정되고 오류를 발생시키는 가중치 전략을 사용하며 비합리적으로 추론 규칙들을 조합하는 문제점 때문에 추론 결과에 상당한 오류 영역을 발생시킨다.

본 논문에서는 이 문제를 해결하기 위해 신경 회로망과 퍼지 이론의 장점만을 융합한 지능적인 뉴로-퍼지 시스템(INFS)을 제안하였다. INFS는 지식 획득 단계에서 전문가로부터 얻은 지식을 바탕으로 대략적인 규칙 베이스와 멤버십 함수를 구성한다. 적응 조절 단계에서는 오류 영역을 최소화시킬 수 있도록 규칙 베이스와 멤버십 함수를 학습하고 조절한다. 추론 단계에서는 임의의 입력 사실에 대해 실제의 추론을 수행할 수 있는 단계이다. 적응 조절 및 추론 단계에는 NIM(Neuro Input Module), FIM(Fuzzy Inference Module), NOM(Neuro Output Module), DFM(Defuzzification Module) 등이 유기적으로 작동한다. NIM에서는 원하는 출력값과 규칙의 조건부 멤버십 함수를 학습하고 또한 멤버십 함수를 조절함으로써 Max-Min CRI의 첫번째 문제점을 해결한다. FIM에서는 제안된 유사도를 이용한 NCRI 방법을 사용하여 추론함으로써 두번째 문제점을 해결한다. NOM에서는 원하는 출력값과 실제 출력값과의 차이를 최소화하기 위해 규칙의 결론부 멤버십 함수를 조절하고, DFM에서는 RSL에 따라 비퍼지화 과정을 수행함으로써 세번째 문제점을 해결한다.



INFS의 성능 평가를 위해 퍼지 추론 방법만을 사용하는 경우와 퍼지 추론 방법에 신경 회로망을 사용하는 경우의 성능을 비교하였다. 어느 경우이든지 Max-Min CRI 방법, Takefuji, Turksen 등이 제안한 유사도 및 NCRI 방법을 이용하였다.

실험 결과, 퍼지 추론 방법만을 사용하는 경우는 NCRI 방법이 나머지 세 방법들에 비해 오류 영역을 최대 82.1%까지 줄여줌으로써 상당히 우수한 성능을 나타내었다. 또한 퍼지 추론과 신경 회로망을 같이 사용하는 경우에는 추론 방법의 종류에 무관하게 학습된 입력 사실에 대해서 모두 원하는 출력값과 동일한 추론 결과를 산출해 줌으로써 신경 회로망을 사용하면 훨씬 더 정확한 결론을 얻을 수 있었다. 이는 RSL이상만을 조합하는 비퍼지화 방법을 사용하기 때문이며 충분히 학습한 결과이기도 하다. 그런데 이 경우에도 NCRI 방법을 적용한 INFS 방법이 Takefuji, Turksen, Max-Min CRI 방법들을 적용했을 때 보다 평균 학습 시간이 각각 4.76%, 28.63%, 40.67% 정도 적게 걸리므로 훨씬 효율적이었다. 따라서 NCRI 방법을 신경 회로망에 융합시킨 INFS 모델은 문제 영역에서 매우 지능적으로 작동하는 시스템임을 알 수 있었다. 그러나 학습되지 않은 입력 사실에 대해서는 어떤 추론 방법도 효율적이지 못하였다. 만약 발생 가능한 모든 경우를 학습 시킨다면 NCRI 방법을 이용한 INFS가 매우 효율적인 방법이 될 것이다.

앞으로 INFS를 다양한 문제 영역에 대해서도 적용시켜 성능을 평가하는 작업과 보다 효율적인 뉴로-퍼지 시스템을 구축하기 위해 추론 방법 및 신경 회로망을 개발해야 할 것이다.

## 참 고 문 헌

1. Z. Cao, A. Kandel, "Applicability of Some Fuzzy Implication Operators," *Fuzzy Sets and Systems*, North-Holland, vol. 31, pp. 151-186, 1989.
2. M. M. Gupta, M. B. Gorzalczany, "Fuzzy Neuro-Computational Technique and Its Application to Modeling and Control," *IEEE International Conference on Fuzzy Systems*, San Diego, pp. 1271-1274, March 8-12, 1992.
3. J. B. Kiszka, M. E. Kochanska, D. S. Sliwinska, "The Influence of Some Implication Operators on the Accuracy of a Fuzzy Model-Part I,II," *Fuzzy Sets and Systems*, North-Holland, vol. 15, pp. 112-128, 223-240, 1985.
4. B. Kosko, *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*, Prentice Hall, 1991.
5. C. C. Lee, "Fuzzy Logic in Control Systems : Fuzzy Logic Controller, Part I,II," *IEEE Transactions on SMC*, vol. SMC-20, no. 2, pp. 404-435, March/April, 1990.
6. C. T. Lin, C. S. George Lee, "Neural-Network-Based Fuzzy Logic Control and Decision System," *IEEE Transaction on Computers*, vol. 40, no. 12, pp. 1320-1336, December, 1991.
7. R. P. Lippmann, "An Introduction to Computing with Neural Nets," *IEEE ASSP Magazine*, pp. 4-20, April, 1987.
8. W. S. McCulloch, W. Pitts, *A Logical calculus of the ideas immanent in nervous activity*, *Bulletin of Mathematical Biophysics*, 5 : 115-133, 1943.
9. M. Mizumoto, H. J. Zimmermann, "Comparison of Fuzzy Reasoning Methods," *Fuzzy Sets and Systems*, North-Holland, vol. 8, pp. 253-283, 1982.
10. M. Sugeno, G. T. Kang, "Structure Identification of Fuzzy Model," *Fuzzy Sets and Systems*, North-Holland, vol. 28, no. 1, pp. 15-33, 1988.
11. H. Takagi, "Fusion Technology of Fuzzy Theory and Neural Networks -Survey and Future Directions-," *Proceedings of the International Conference on Fuzzy Logic & Neural Networks, IIZUKA, Japan*, pp. 13-26, July 20-24, 1990.

12. Y. Takefuji, Meng-hiot Lim, "Computation Scheme for the General Purpose VLSI Fuzzy Inference Engine as Expert System," *Butterworth & Co(publishers) Ltd.*, vol. 2, no. 2, pp. 109-116, June, 1989.
13. I. B. Turksen, Z. Zhong, "An Approximate Analytical Reasoning Scheme based on Similarity Measures and Interval Valued Fuzzy Sets," *Fuzzy Sets and Systems*, North-Holland, vol. 34, pp. 323-346, 1990.
14. T. Yamakawa, T. Miki, "The Current mode Fuzzy Logic Integrated Circuits/Fabricated by the Standard CMOS Process," *IEEE Transactions on Computer*, vol. C-35, no. 2, pp. 161-167, 1986.
15. L. A. Zadeh, "Outline of a new Approach to the analysis of complex Systems and Decision Process," *IEEE Transactions on SMC*, vol. SMC-3, no. 1, pp. 27-44, January, 1973.