

고급 분산 제어 시스템을 위한 고장 진단 퍼지 전문가 시스템의 개발

변승현, 박세화, 허윤기, 서창준, 이재혁, 김병국, 박동조, 변중남
한국과학기술원 전기 및 전자공학과

Development of Fault Diagnosis Fuzzy Expert System for Advanced Control System

S.H. Byun, S.-H. Park, Y.K. Heo, C.J. Seo, J.H. Lee, B.K. Kim, D.J. Park and Z. Bien
Dept. of EE, KAIST

ABSTRACTS

We developed fault diagnosis fuzzy expert system for ACS(Advanced Control System). ACS is a DCS(Distributed Control System) with advanced control algorithm, fault tolerance capabilities, fault diagnosis functions, and so on. Fuzzy expert system developed for an ACS in this paper gives an operator alarm signal depending on the state of process value and manipulated value, and the cause of alarm in real time. Simple experiment result with several rules for the fault diagnosis of drum level loop in Seoul Power Plant

따라서 본 논문에서는 운용자로 하여금 OIS를 통해 플랜트 및 PCS의 운전상황을 모니터링해서 알람(alarm) 발생의 가능성 여부를 예측하고 알람 발생의 여지가 있을 경우 원인이 될 수 있는 고장부위를 운용자에게 미리 알려줄 수 있는 전문가 시스템을 개발하고자 한다. 지식베이스에 구축된 전문가의 지식들이 일반적으로 정량적 표현이 아닌 언어적 표현에 의존하므로 그 지식의 불확실성과 표현 용어 자체에 애매함이 포함될 수 있는 퍼지 집합을 이용해 지식을 표현하고 퍼지 집합으로 표현된 지식을 퍼지 논리로 처리할 수 있는 퍼지 전문가 시스템을 개발하고자 한다.

II절에서는 개발한 퍼지 전문가 시스템을 설명하고, III절에서는 개발한 전문가 시스템의 적용가능성에 대한 실험을 통한 결과를 보이고, IV절에서는 결론 및 추후과제를 기술하고자 한다.

II. 퍼지 전문가 시스템

1. 개요

전문가시스템은 적용하는 영역의 문제를 해결할 수 있는 지식을 저장하는 지식 베이스와 지식 베이스에 저장된 지식을 이용하여 새로운 사실을 추론하는 추론 엔진, 사용자와 지식 베이스, 규칙 베이스를 연결시켜 주는 사용자 접합부로 이루어진다. 본 논문의 퍼지 전문가 시스템의 지식은 실제 퍼지 응용에서 가장 널리 이용되는 퍼지 if-then 규칙에 의해 표현한다.

2. 시스템 구성

본 논문에서의 퍼지 전문가 시스템의 구성도는 그림 2-1과 같고, 각 부분에 대한 내용은 다음과 같다. 퍼지 전문가 시스템은 [2]를 참조해서 구현하였다.

(1) 객체 관리 루틴

각 객체가 어느 규칙에서 이용되었나를 관리하며 주어진 사실과 추론되거나 받아들인 사실의 객체를 관리한다. 객체베이스의 표현 형식은 그림 2-2와 같다.

(2) 퍼지 용어 관리 루틴

객체가 언어변수로써 가질수 있는 퍼지집합으로 표현되어지는 용어, 전체집합(universe of discourse)을 관리한다. 객체가 언어변수로써 가질수 있는 퍼지 집합과 라벨들의 퍼지 베이스의 형식은 그림 2-3과 같다.

(3) 사실 관리 루틴

추론을 하는 데 반드시 필요한 사실들(추론에 의해 얻을 수 없는 사실들)을 받아들여 객체 관리 루틴에 넘겨준다.

(4) 규칙 관리 루틴

퍼지 if-then 규칙의 전건부와 후건부에서 이용되어진 객체와 객체가 언어 변수로써 가지는 언어 값을 나타내는 퍼지 집합, 규칙의 확신도, 그리고 전건부에서 이용되는 논리 연산자를 관리한다. 규칙베이스의 표현 형식은 그림 2-4와 같다.

(5) 추론 엔진

알고자하는 사실에 관련된 객체 목표(goal)이 주어지면 목표로 주어진 객체에 대한 사실을 추론하기 위해 필요한 규칙들을 역방향 체인(backward chaining)으로 찾아 트리를 형성하고, 전방향 계산(forward evaluation)으로 알고자 하는 사실을 추론한다. 그리고 지식 베이스의 규칙과사실로부터 새로운 사실을 추론하기 위해서 CRI(Compositional Rule of Inference)방법을 이

I. 서론

공정 제어를 위해 디지털 컴퓨터가 사용되기 시작한 이래, 컴퓨터 기술의 발달은 산업 공정 제어 분야에 매우 큰 영향을 미치게 되었다. 특히 마이크로 프로세스의 발달은 제어 장치의 혁신적인 변화를 초래해서 대규모 공정 제어 분야에도 크게 영향을 미쳐 대부분의 대규모 제어 시스템에 마이크로 프로세서들이 이용한 디지털 제어 방식이 사용되어 전자 제어 시스템은 모듈화 계층화 되었다. 전자 제어 시스템이 모듈화 계층화됨에 따라 효과적인 제어를 위해서 분산 제어 시스템이 도입되었다. 국내에서도 KAIST등 몇몇 학교와 기업, 연구기관에서 분산제어 시스템의 국산화에 박차를 가하고 있다. [7][8][9]

일반적으로 분산 제어 시스템은 PCS (Process Control System), EWS (Engineering Work Station), 그리고 OIS (Operator Interface Station)로 구성되어 있다. 여기서 PCS는 실제적으로 플랜트의 제어를 담당하고 EWS는 PCS를 위한 여러 엔지니어링작업을 하며, OIS는 PCS가 보낸 플랜트의 공정 정보 및 PCS의 제어값을 운용자에게 보여준다. 이런 기본 기능을 갖춘 일반적인 분산 제어 시스템 보다 진보된 고급 분산 제어 시스템은 기존의 분산 제어 시스템에 추가적으로 예측제어, 신경회로망제어등의 고급제어 알고리즘을 갖추고 있고, 또한 운용자에게 플랜트 및 제어 시스템의 고장에 대한 대처 기능 및 정보를 제공해 주는 기능을 갖고 있다. 이러한 고급 분산 제어 시스템은 기능적인 면이나 성능 면에서 공정 제어에서 지향해 나가야될 방향임에도 불구하고 아직 뚜렷한 개발 사례가 알려지지 않고 있다.

본 논문에서는 고급 분산 제어 시스템의 구성 요소가 될 수 있는 고장 진단 시스템의 개발을 목표로 한다. 대규모 제어 시스템의 경우 전자 제어 시스템이 모듈화, 계층화됨에 따라 고장 발생 요인도 많아지고 고장에 따른 손실이 크므로 고장 진단의 자동화가 매우 중요하다고 볼 수 있다. 그런데 대규모 시스템은 그 구성요소가 많고, 구조가 복잡하므로 숙련된 운전자의 경험과 지식을 지식베이스로 가짐으로써 전체 시스템의 고장 진단 기능을 하는 전문가 시스템에 의한 고장 진단이 바람직하다.

용하였다

앞으로의 내용에서 이용되는 모든 퍼지 집합을 다음과 같이 미리 정의하고자 한다.

$$A = \int_U \mu_A(u)/u, A' = \int_U \mu_{A'}(u)/u$$

$$B = \int_V \mu_B(v)/v, B' = \int_V \mu_{B'}(v)/v$$

$$C = \int_W \mu_C(w)/w, C' = \int_W \mu_{C'}(w)/w$$

가) CRI

규칙 : If x is A then y is B

사실 : x is A'

결론 : y is B'

위와 같이 규칙과 사실이 주어지고, 그로부터 y에 대한 새로운 결론 퍼지 집합 B'는 다음과 같이 추론된다.

$$B' = R(A) \circ R(A, B) = A \circ R(A, B)$$

즉 이 추론방법은 어떤 규칙과 사실이 주어졌을 때, 규칙의 전건부와 후건부의 퍼지 집합들로부터 하나의 관계를 형성하고, 사실로 주어진 퍼지 집합을 또 하나의 관계로 보아서 두 관계를 합성 연산해서 새로운 결론을 추론하는 방법이다.[4] 따라서 CRI 추론방법은 전건부와 후건부의 퍼지 집합을 관계시키는 퍼지 관계와 사실로 주어진 퍼지 집합과 규칙으로부터의 관계를 합성시키는 합성 연산자의 선택에 따라서 추론 능력이 좌우된다고 볼 수 있다. 이 논문에서는 [3]에서 제시된 직관적인 판단 기준에 가장 적합하다고 생각되는 퍼지 관계 R_s 를 이용하였고 [1]에 나와있는 합성 연산자를 여러 퍼지 관계에 대해서 시뮬레이션 해 본 결과 가장 잘 추론을 해 낸다고 생각되는 sup-drastic-product 연산을 합성 연산으로 이용하였다. 위에서 기술한 규칙과 사실이 주어졌다고 하면 퍼지관계 R_s 와 합성 연산 sup drastic product 연산은 다음과 같이 규칙의 전건부와 후건부의 퍼지집합으로부터 관계를 형성하고 사실로 주어진 퍼지 집합과의 합성에 의해 결론의 퍼지 집합을 추론한다.

$$R_s = \int_{U \times V} [\mu_A(u) \rightarrow_s \mu_B(v)] / (u, v)$$

$$\text{where } \mu_A(u) \rightarrow_s \mu_B(v) = \begin{cases} 1 & \mu_A(u) \leq \mu_B(v) \\ 0 & \mu_A(u) > \mu_B(v) \end{cases}$$

$$B' = \int_V \bigvee_{u \in U} [\mu_{A'}(u) \wedge \mu_{R_s}(u, v)] / v$$

$$\text{where } x \wedge y = \begin{cases} x & y=1 \\ y & x=1 \\ 0 & x, y < 1 \end{cases}$$

나) 규칙 추론(rule evaluation)

전건부에 다중 명제를 가지는 규칙들에 대해서 경우별로 나누어서 설명하면 다음과 같다.

경우 1)

규칙 : IF x is A AND y is B THEN z is C (CF_R)

사실 : x is A' (CF_1), y is B' (CF_2)

위와 같이 규칙과 사실이 주어졌을 때 일단 다음과 같이 전건부에 단일 명제를 갖는 규칙으로 논리적 동가(logical equivalence)에 의해서 나뉘어 생각한다.

규칙1 : IF x is A THEN z is C (CF_R)

규칙2 : IF y is B THEN z is C (CF_R)

규칙1과 규칙2는 OR로 묶이면 논리적으로 동가인 규칙이 된다. 규칙 1과 사실로부터 결론1, 규칙 2와 사실로부터 결론2를 추론하고 최종적으로 규칙과 사실로부터의 결론을 다음과 같이 추론한다.

결론1 : z is C_1 , where $C_1 = A' \circ R(A, C)$

결론2 : z is C_2 , where $C_2 = B' \circ R(B, C)$

결론 : z is C' (CF_c), where $C' = C_1 \cup C_2$

$$CF_c = \min(CF_1, CF_2) * CF_R$$

경우 2)

규칙 : IF x is A OR y is B THEN z is C (CF_R)

사실 : x is A' (CF_1), y is B' (CF_2)

위와 같이 규칙과 사실이 주어졌을 때 경우 1)에서와 같이 퍼지 집합들이 정의되었다고 하면, 결론은 다음과 같이 구할 수

있다.

결론 : z is C' (CF_c), where $C' = C_1 \cap C_2$

$$CF_c = \max(CF_1, CF_2) * CF_R$$

경우 3)

규칙의 전건부에 3개이상의 명제가 2개 이상의 AND, OR 연산자를 가질 때 규칙속에 내재된 경우 1)과 경우 2)를 결론에 도달할 때까지 반복적으로 적용하면 된다.

다) 중후 결합(evidence combination)

규칙 1 : IF x is A THEN z is C

규칙 2 : IF y is B THEN z is C

사실 : x is A', y is B'

위와 같이 후건부에 같은 객체(언어 변수)를 갖는 규칙이 두 개 존재해서 각 규칙으로부터 근사추론에 의해서 결론 1과 결론 2를 구하고 그 결론들로부터 다음과 같이 최종의 결론을 추론할 수 있다.

결론 1 : z is C_1 (CF_1), where $C_1 = A' \circ R(A, C)$

결론 2 : z is C_2 (CF_2), where $C_2 = B' \circ R(B, C)$

결론 : z is C_R (CF_R), where $C_R = C_1 \cap C_2$

$$CF_R = CF_1 * CF_2 * (1 - CF_1)$$

3개이상의 규칙들이 후건부에 같은 언어변수를 가지고 있다면 위에 기술된 방법을 반복 적용해서 결론에 도달한다.

(6) 전처리 (preprocessing)

공정 정보들을 받아들여 미리 정의된 퍼지집합에 대응시키는 퍼지화들을 포함하는 루틴이다.

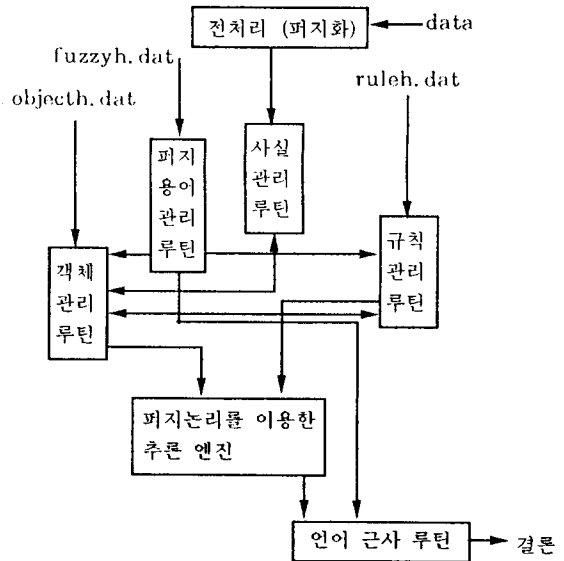


그림 2-1. 시스템 구성도

Fig 2-1. system configuration

```
{ Objectname : objectname
Attributes : (attributename1 ; type1 )
              (attributename2 ; type2 )
              .....
              (attributenameN ; typeN )
Description : this is format of rule }
```

그림 2-2 객체의 형식

Fig 2-2 format of object

```
{ Variablename : object.attribute
universe of discourse : num1 num2
Unit : unit
Number of value : N
fuzzylabell1 : num10, num11, ..., num1a
fuzzylabell2 : num20, num21, ..., num2a
.....
fuzzylabelln : numn0, numn1, ..., numna
Description : this is ling var format }
```

그림 2-3 퍼지 언어 변수의 형식
Fig 2-3 format of linguistic variable

```
{ Rulename : N
( antecedent propositions )
->
( consequent propositions )
Certainty : num
Description : if ant then con(num). }
```

그림 2-4 규칙의 형식
Fig 2-4 format of rule

III. 실험

1. 실험 환경

사용된 실험 환경은 OIS/EWS의 기능을 갖는 호스트 컴퓨터로서 SUN 워크스테이션을 이용하였고, 공정 제어 위해 핵심이 되는 프로세서 모듈로는 모듈자체에 네트워크를 위한 이더넷(ethernet) 포트를 내장하고 있으며 고속의 연산 기능을 가진 CPU30을 사용하였다. 그리고 서울화력 4호기 발전소 보일러 시스템은 디지털로 구현한 시뮬레이터로서도 CPU30이 이용된다. CPU30은 VME BUS에 인터페이스 되어서 전체적으로 SUN 시스템이 네트워크를 통해 정보를 주고 받게 된다. 또 실시간 운영 체제인 VxWorks를 OIS/EWS 기능을 하는 SUN 시스템에 구축해서, 타겟이 되는 프로세서 모듈인 CPU30이 부팅시에 Vxworks 관련 파일을 LAN(Local Area Network)를 통해 가져다가 RAM에 올려놓고, 프로세서 모듈에서 실시간 목적을 위해 이용한다. 실험 시스템 구성도는 그림 3-1과 같다.[9]

2. 실험 내용

II절에서 기술한 퍼지 전문가 시스템을 이용해서 서울화력 4호기 보일러의 시뮬레이터를 대상으로 위에서 기술한 실험 환경하에서 모의 고장 실험을 통해 개발된 퍼지 전문가 시스템의 적용 가능성을 보이고자 한다. 실험 대상인 보일러는 드럼형 보일러로서 주증기 압력 제어루프, 노내압 제어루프, 주증기 온도 제어루프, 드럼 수위 제어루프등 많은 제어 루프를 갖는다. 본 연구에서는 PCS로부터 MV(Manipulated Value), PV(Process Value)를 읽어들이어 드럼 수위 제어루프에 관한 고장여부를 운용자에게 알려주고자 한다. 제어기 출력인 MV와 시뮬레이터 출력인 PV에 대한 단위는 그것들이 가질 수 있는 가장 큰 값을 100, 가장 작은 값을 0으로 변환한 %단위를 이용하였다. 실험에서 이용하는 시뮬레이터는 신호흐름도 모델을 이용해 발전소 보일러를 모사한 시뮬레이터이다.[6] 드럼 수위 동특성에 해당하는 신호흐름도 모델은 그림 3-2와 같다. 드럼 수위에 대한 상태는 그림 3-3처럼 드럼수위의 값에 따라 알람 상태와 정상상태로 나뉜다. 드럼 수위에 대한 신호흐름을 체크하고 미리 알람상태를 예측하며 알람의 원인이 될 수 있는 고장 부위를 체크해서 결과를 운용자에게 보여준다. 실험 환경을 고려한 전체 시스템 구성도는 그림 3-4와 같다.

실험은 다음과 같은 방법에 의해서 진행되었다. 드럼 수위 동특성 신호흐름도인 그림 3-2에 나오는 급수량의 제어는 2대의 급수펌프(Boiler Feedwater Pump), BFPa, BFPb에 의해서 행해진다. 급수량을 제어하는 2대의 액츄에이터중 하나인 BFPb의 출력이 250초이후로는 항상 MV에 관계 없이 0으로 나타나는 고장을 발생시켜서 MV와 PV의 변화를 주시하고 관찰한 내용을 규칙으로 기술해서 전문가 시스템의 지식베이스로 활용하였다. 그리고 나서 똑같은 고장을 일으켰을 때 전문가 시스템이 어떤 정보를 주는지 실험해보았다. 정상상태에서 250초이후로 1000초까지 급수량을 제어하는 actuator인 BFPb를 0으로 출력했을때의 주요데이터들의 경향이 그림 3-5와 같이 나타난다. 이 데이터들의 경향의 관찰에 의해서, 그림 3-6과 같은 객체들을 정의하고, 객체가 언어변수로서 가질 수 있는 그림 3-7과 같은 퍼지 용어들의 정의를 이용하여 그림 3-8과 같은 규칙을 세웠다. 전처리과정으로써 퍼지화는 그림 3-5에 나온 MV와 PV를 고려하여 각 변수마다 구간을 나누고 그 구간을 하나의 퍼지 집합으로 대응시켜서 했다.

3. 실험 결과

추론되어서 언어근사과정을 거친 드럼 수위 상태에 대한 결과가 정상(normal)이면 75%를 출력하게 하고, 지금은 알람 상태는 아니지만 알람상태로 가고 있는 것이면, 즉 알람 상태를 미리 예측하면(bad) 35%를 출력하게 하며, 알람상태(alert)이면

25%를 출력하게 했다. 그리고 실험 상에 이용된 규칙만으로는 평가가 어려울 때는 unknown이 나오는데, 그 때는 65%를 출력하게 했다. 또 하나의 추론 결과로써 BFPb가 고장일 것 같으면(possible) 20%를 출력하게 하고, 정상일 것 같으면(unlikely) 80%를 출력하게 하였으며 unknown이면 70%를 출력하게 했다. 그림 3-9의 실험 결과에서 드럼 수위에 대한 상태의 추론 결과는 드럼 수위 L과 함께 나타내었고, BFPb고장에 대한 추론 결과는 BFPb와 함께 실험결과를 나타내었다. 실험 결과가 알람 상태 전에 알람을 예측하고 고장을 탐지함을 볼 수 있다.

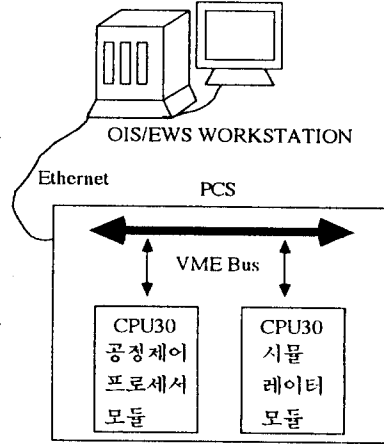
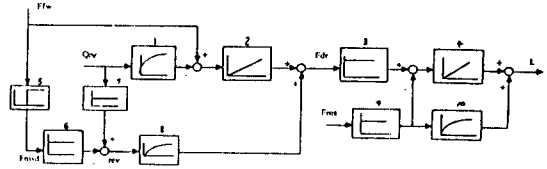


그림 3-1 실험 시스템 구성도
Fig 3-1 experiment system



Ffw(feedwater flow) : 급수량
L(Level) : 드럼 수위
Fms(mainsteam flow) : 주증기량
Qev : 증발기에서의 열량

그림 3-2 드럼 수위 동특성 신호 흐름도
Fig 3-2 drum level dynamic characteristic

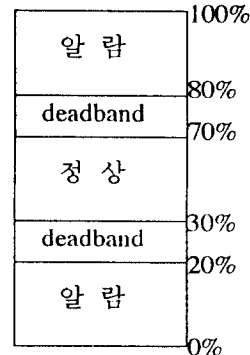


그림 3-3 드럼 수위의 상태
Fig 3-3 state of drum level

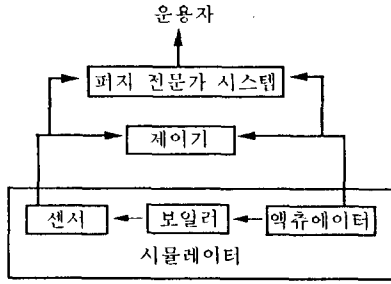


그림 3-4 전체 시스템 구성도
Fig 3-4 integrated system configuration

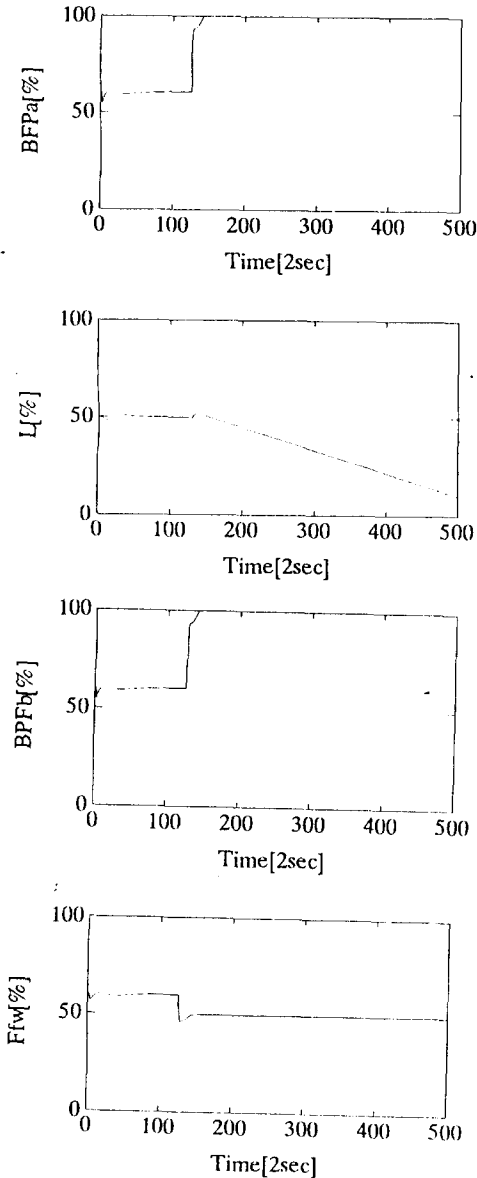


그림 3-5 모의 고장시의 데이터 흐름
Fig 3-5 data flow

- (Objectname : drum
Attributes : (level ; fuzzy)
Description : This is)
- (Objectname : state
Attributes : (drumlevel ; fuzzy)
Description : This is)
- (Objectname : output
Attributes : (BFPb ; fuzzy)
Description : This is)
- (Objectname : change
Attributes : (drumlevel ; fuzzy)
(feedwaterflow ; fuzzy)
(BFPboutput ; fuzzy)
Description : This is)
- (Objectname : fault
Attributes : (BFPb ; fuzzy)
Description : This is)

그림 3-6 진단을 위해 정의되어진 객체
Fig 3-6 defined object for diagnosis

- (variablename : drum.level
universe of discourse : 0 100
unit : percent
number of value : 7
highest : .0 .0 .0 .0 .0 .0 .0 .5 1.0
higher : .0 .0 .0 .0 .0 .0 .0 1.0 1.0 .0
high : .0 .0 .0 .0 .0 .5 1.0 1.0 1.0 .0
middle : .0 .0 .0 .5 1.0 1.0 1.0 .0 .0 .0
low : .0 .0 .5 1.0 1.0 .0 .0 .0 .0 .0
lower : .0 1.0 1.0 .0 .0 .0 .0 .0 .0 .0
lowest : 1.0 1.0 .0 .0 .0 .0 .0 .0 .0 .0
description : -15 - 15 inch -> %)
- (variablename : state.drumlevel
universe of discourse : -1 1
unit : dob
number of value : 4
worst : 1.0 1.0 .0 .0 .0 .0 .0 .0 .0 .0
alert : .0 1.0 1.0 .0 .0 .0 .0 .0 .0 .0
bad : .0 .0 .0 .5 1.0 1.0 1.0 .0 .0 .0
normal : .0 .0 .0 .0 .0 .0 .0 .0 .5 1.0
description :)
- (variablename : change.drumlevel
universe of discourse : -1 1
unit : ratio
number of value : 5
poitiveabrupt : .0 .0 .0 .0 .0 .0 .0 .0 .5 1.0
positive : .0 .0 .0 .0 .0 .0 .3 1.0 1.0 .3 .0
zero : .0 .0 .0 .5 1.0 1.0 .0 .0 .0 .0
negative : .0 .3 1.0 1.0 .3 .0 .0 .0 .0 .0
negativeabrupt : 1.0 1.0 .0 .0 .0 .0 .0 .0 .0 .0
description :)
- (variablename : change.feedwaterflow
universe of discourse : -1 1
unit : ratio
number of value : 5
poitiveabrupt : .0 .0 .0 .0 .0 .0 .0 .0 .5 1.0
positive : .0 .0 .0 .0 .0 .0 .3 1.0 1.0 .3 .0
description :)

```

zero : .0 .0 .0 .0 .5 1.0 .5 .0 .0 .0 .0
negative : .0 .3 1.0 1.0 .3 .0 .0 .0 .0 .0 .0
negativeabrupt : 1.0 .5 .0 .0 .0 .0 .0 .0 .0 .0 .0
description : )

```

```

( variablename : change.BFFboutput
  universe of discourse : -1 1
  unit : ratio
  number of value : 5
  poitiveabrupt : .0 .0 .0 .0 .0 .0 .0 .0 .5 1.0
  positive : .0 .0 .0 .0 .0 .0 .3 1.0 1.0 .3 0.0
  zero : .0 .0 .0 .0 .5 1.0 .5 .0 .0 .0 .0
  negative : .0 .3 1.0 1.0 .3 .0 .0 .0 .0 .0 .0
  negativeabrupt : 1.0 .5 .0 .0 .0 .0 .0 .0 .0 .0 .0
  description : )

```

```

( variablename : output.BFFpb
  universe of discourse : 0 100
  unit : percent
  number of value : 5
  full : .0 .0 .0 .0 .0 .0 .0 .0 .5 1.0
  high : .0 .0 .0 .0 .0 .0 .5 1.0 .5 .0 .0
  middle : .0 .0 .0 .0 .5 1.0 .5 .0 .0 .0 .0
  low : .0 .0 .5 1.0 .5 .0 .0 .0 .0 .0 .0
  zero : 1.0 .5 .0 .0 .0 .0 .0 .0 .0 .0 .0
  description : )

```

```

( variablename : fault.BFFpb
  universe of discourse : 0 1
  unit : possibility
  number of value : 2
  possible : .0 .0 .0 .0 .0 .0 .5 .5 1.0 1.0
  unlikely : 1.0 1.0 .5 .5 .5 .0 .0 .0 .0 .0 .0
  description : )

```

그림 3-7 진단을 위해 정의되어진 퍼지 용어
Fig 3-7 defined fuzzy terms for diagnosis

```

(
  RuleName : 1
  ( Is drum.level higher OR Is drum.level lower )
  -->
  ( Is state.drumlevel alert )
  Certainty : 1.0
  Description : IF drum.level is higher or drumlevel is
                lower THEN state.drumlevel is alert
)

(
  RuleName : 2
  ( Is drum.level highest OR Is drum.level lowest )
  -->
  ( Is state.drumlevel worst )
  Certainty : 1.0
  Description : This is the rule named by
)

(
  RuleName : 3
  ( Is drum.level middle AND ( Is change.feedwaterflow
    zero OR Is change.drumlevel zero ) )
  -->
  ( Is state.drumlevel normal )
  Certainty : 1.0
  Description : This is the rule named by
)

(
  RuleName : 4

```

```

( Is drum.level low AND Is change.drumlevel zero
  AND Is change.feedwaterflow zero )
-->
( Is state.drumlevel normal )
Certainty : 1.0
Description : This is the rule named by
)

(
  RuleName : 5
  ( Is drum.level low AND Is change.drumlevel negative
    AND Is change.feedwaterflow zero )
  -->
  ( Is state.drumlevel bad )
  Certainty : 1.0
  Description :
)

(
  RuleName : 6
  (Is change.drumlevel negative AND Is change.feedwaterflow
    negative AND Is change.BFFboutput positiveabrupt )
  -->
  ( Is fault.BFFpb possible )
  Certainty : 1.0
  Description :
)

(
  RuleName : 7
  (Is output.BFFpb full AND Is change.drumlevel negative
    AND Is change.feedwaterflow zero )
  -->
  ( Is fault.BFFpb possible )
  Certainty : 1.0
  Description :
)

(
  RuleName : 8
  (( Is drum.level middle OR Is drum.level low )
    AND Is fault.BFFpb possible )
  -->
  ( Is state.drumlevel bad )
  Certainty : 1.0
  Description :
)

(
  RuleName : 9
  (Is change.drumlevel zero AND Is change.feedwaterflow
    zero )
  -->
  ( Is state.drumlevel normal )
  Certainty : 1.0
  Description :
)

(
  RuleName : 10
  ( Is change.BFFboutput zero AND Is output.BFFpb middle )
  -->
  ( Is fault.BFFpb unlikely )
  Certainty : 1.0
  Description :
)

```

그림 3-8 진단에 이용되어진 규칙
Fig 3-8 used rule for diagnosis

IV. 결론 및 추후과제

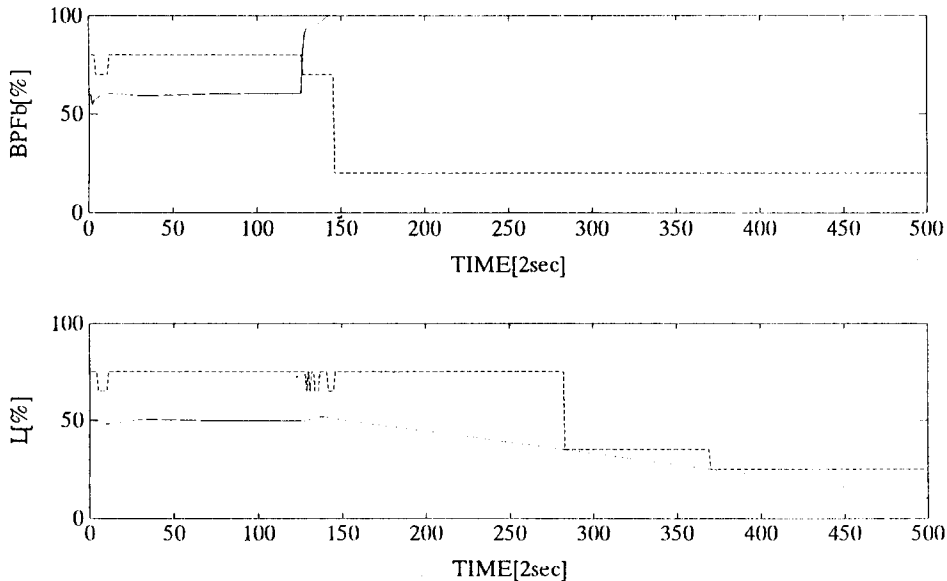
본 연구에서는 퍼지 전문가 시스템을 이용해서 서울화력 4호기 보일러의 시뮬레이터를 대상으로 모의 고장 실험을 통해 본 연구의 적용 가능성을 보였다. 앞으로 사용자 인터페이스 기능을 강화해서 전문가 시스템과의 통합이 요구된다. 본 연구에서는 입력 데이터를 퍼지화 할 때 공정 제어 데이터들을 구간으로 나눠 구간 하나를 하나의 퍼지 집합에 대응시켰는데 효과적인 퍼지화가 요구된다. 본 논문의 모의 고장 실험에서는 적용 규칙이 적어 실시간 고장 진단이 가능하지만 많은 규칙이 지식 베이스를 이루어 대규모 제어 시스템의 고장 진단을 행할 경우에는 실시간 제어가 어려워 이의 해결도 요구된다. 본 연구에서는 아날로그 신호만 다루어 알람 처리를 했는데, 디지털 신호도 고려한 알람 처리도 포함되는 연구도 요구된다.

V. 참고 문헌

- [1] C.C.Lee, "Fuzzy Logic in Control Systems: Fuzzy Logic controller, Part II.", IEEE Trans. SMC vol 20, NO.2, pp419-435, 1990
- [2] K.S.Leung and W.Lam, "Fuzzy Concepts in Expert Systems", IEEE computer september pp43-56, 1988
- [3] Mizumoto and Zimmermann, "comparison of fuzzy reasoning methods", Fuzzy sets and systems 8 pp253-283, 1982
- [4] L.A.Zadeh, "Outline of a New Approach to the analysis of complex systems and decision processes", IEEE Trans.SMC vol 3, NO.1, pp28 - 44, 1973
- [5] 이광형, 오길복 "퍼지 이론 및 응용" I 이론, II 응용, 흥통과 학 출판사, 1991
- [6] 김 재선 "신호 흐름도 모델을 이용한 화력발전소 드럼형 보일러 시뮬레이터에 관한 연구", 한국과학기술원, 전기 및 전자공

학과, 석사학위논문, 1989

- [7] 한전 기술 연구원, "발전소 제어용 디지털 계장 제어 시스템 개발", 최종보고서, 1990.8
- [8] 한전 기술 연구원 "발전소 제어 시스템용 고급 제어 알고리즘 및 고장 진단 퍼지 전문가 시스템 개발", 중간 보고서, 1992. 11
- [9] 한전 기술 연구원 "분산 제어 시스템의 고장 대처 기능 및 제어 언어의 구현", 최종보고서, 1993.3



-- : 추론 결과

드럼수위상태 : normal(75%), bad(35%), alert(25%), unknown(65%)

BPFb의 고장 여부 : possible(80%), unlikely(20%), unknown(70%)

— : 모의 고장시의 PV, MV

그림 3-9 실험 결과

Fig 3-9 experiment result