

모듈신경망을 이용한 다중고장 진단기법

배 옹 환*, 이 석 희**

Multiple Fault Diagnosis Method by Modular Artificial Neural Network

Yong-Hwan Bae*, Seok-Hee Lee**

ABSTRACT

This paper describes multiple fault diagnosis method in complex system with hierarchical structure. Complex system is divided into subsystem, item and component. For diagnosing this hierarchical complex system, it is necessary to implement special neural network. We introduced Modular Artificial Neural Network(MANN) for this purpose. MANN consists of four level neural network, first level for symptom classification, second level for item fault diagnosis, third level for component symptom classification, fourth level for component fault diagnosis. Each network is multi layer perceptron with 7 inputs, 30 hidden node and 7 outputs trained by backpropagation. UNIX IPC(Inter Process Communication) is used for implementing MANN with multitasking and message transfer between processes in SUN workstation. We tested MANN in reactor system.

Key Words : Diagnosis Method(진단기법), Fault Detection(고장감지), Monitoring System(감시시스템), Modular Artificial Neural Network(모듈신경망), Multiple Fault Diagnosis(다중고장진단), IPC(Inter Process Communication)

1. 서 론

프로세스 유니트와 제어시스템의 물리적인 조건변화나 외란 조건들에 의하여 일반적으로 고장이 야기된다. 정상적으로 프로세스가 작동할 때, 프로세스 파라미터는 그들의 정상 상태하에 있어야 한다. 만약 어떤 시스템의 물리적인 변화가 정상상태로부터 편차를 나타내면 프로세스의 모델 파라미터들은 그들의 정상 상태값으로부터 벗어

날 것이다. 고장을 진단하기 위하여 측정값을 직접 감지하기 보다는 차라리 고장을 분리해 내기 위하여 모델 파라미터를 이용하는 모델기반 고장진단 기법이 자주 사용된다. 일반적으로 프로세스의 파라미터와 측정값 사이에는 특수한 관계가 존재한다. 하나의 고장은 프로세스 응답의 연속측정과 그들을 파라미터값들로 변환함으로써 진단할 수 있다. 고장진단의 문제는 패턴인식의 하나로 취급될 수 있는데, 문제는 프로세스 변수들의 측정에 의

* 안동대학교 기계공학교육과 (정회원)

** 부산대학교 생산기계공학과, 기계기술연구소 연구원 (중신회원)

하여 고장들을 어떻게 올바르게 측정할 수 있는가 하는 것이다. 다른 말로 하면 Fig.1 처럼 측정공간(measurement space)으로부터 고장공간(fault space)으로 매핑할 수 있는가 하는 것이다. Watanabe등은⁽¹⁾ 인공신경망을 이용하여 플랜트의 단일고장을 성공적으로 진단하였다. 여기서 Watanabe와 Himmelblau에 의하여 제시된 모델베이스 고장진단과 같은 다른 기법들보다 훨씬 빠르고 효과적인 초기의 다중 고장진단의 일반적인 문제해결에 관심을 두었다. 본 연구에서는 제충화된 시스템의 다중고장 진단기법으로써 먼저 시스템에 고장이 발생되면 시스템 전체에 나타나는 징후를 통하여 부속시스템을 진단하고, 고장후보 부속셀중 하나를 취하여 그 하부요소인 아이템(item)과 부속품(component)을 진단하는 것을 목적으로 한다. 이러한 목적을 위하여 본 연구에서는 다중고장 진단을 위한 4레벨의 모듈신경망(Modular Artificial Neural Network: MANN)을 이용한다. 상위 2레벨은 서브시스템 고장진단을 통한 고장난 아이템을 구별하는 것이고, 하부 2레벨은 그 아이템속의 고장난 부속품의 고장을 구별해 낼수 있는 유닉스 운영체제기반의 실시간 제충형 다중고장진단을 위한 신경망을 개발하였는데, 본 신경망의 특징은 하나의 신경망 셀이 독립적인 프로세스로 필요시 상위 프로세스에 요구에 따라 자동적으로 생성되며, 이렇게 생성된 프로세스들은 필요시 메시지큐라는 IPC(Inter Process Communication)설비를 이용하여 고장정보를 교환한다. 본 연구에서 개발된 신경망을 이용하여 반응로 설비의 다중고장진단에 이용하였는데 반응로의 고장은 프로세스를 지배하는 파라메타의 다중변화에 기인하며, 이러한 다중변화를 입력으로 한 다중고장을 진단하였다.

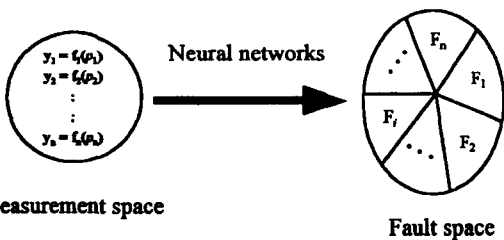


Fig. 1 Mapping from a measurement space to a fault space

2. 고장진단의 정의

진단은 고장의 위치를 찾아가는 과정이며⁽²⁾, 또한 탐지

된 고장을 위한 대응조치를 제시하는 것이다. 진단프로세스(diagnosis process)는 고장탐지(fault detection), 고장위치추정(fault localization), 고장분리(fault isolation)로 요약될 수 있다. 고장탐지는 시스템이 적절하게 작동하지 않을때 나타나는 징후를 관찰하는 것이다. 센서(sensors)는 시스템의 각종 징후를 표시해 주는 역할을 한다. 고장위치추정은 가장 적절한 원인을 찾아낼 때까지 가능한 원인들을 계속 추출하는 과정이다. 고장분리는 징후소스(symptom source)를 결정하기 위한 탐색 과정이다.⁽³⁾

고장(fault)은 시스템 설계자가 의도한 어떤 목적의 수행을 불가능하게 만드는 시스템 파라메타의 허용범위를 넘어선 편차로 이해될 수 있다. 프로세스 고장이 나타난다면 이것을 가능한 빨리 탐지해야 하는데, 이것은 특정 측정 및 미측정 추정변수가 정상값의 어떤 범위를 벗어나는지를 체크함으로써 행해진다. 이러한 체크과정을 통과하지 못하면 고장 메시지를 발생시킨다. 이러한 기능을 통상 감시(monitoring) 혹은 고장탐지(fault detection)라 부르며, 필요에 따라 고장진단(fault diagnosis)이 뒤따라진다. 다음 단계는 고장평가(fault evaluation)이다. 이것은 고장이 프로세스에 어떻게 영향을 미치는지에 대한 평가를 의미한다. 고장들은 사건, 순차 분석이나 고장나무목(fault tree analysis)에 따라 각기 다른 위험등급으로 나누어진다. 고장의 영향이 알려진 다음에는 취해져야할 행동의 결정(decision)이 형성되고, 고장이 허용 가능하다면 작동은 계속 이루어지고, 조건적으로 가능하다면 작동오차를 보정해야 한다. 그러나 고장이 허용되어질 수 없다면 작동(operation)은 즉각 정지되어야 하고, 고장은 제거되어야 한다.

현재의 고장진단 장치는 주로 시스템의 센서정보를 컴퓨터 자동고장진단 시스템으로 입력해서 추론하는 장치들이다. 이러한 장치들은 주로 정해진 기능내에서 실시간 고장 대처능력을 가지고 있다. Padalkar등은⁽⁴⁾ 실시간 고장감지와 진단에 대하여 다음과 같이 정의하고 있다. "고장감지(fault detection)는 고장난 시스템의 동정을 감지하는 프로세스로 정의한다. 고장난 시스템 동정은 예측된 시스템 동정보로부터 어떤 편차로 정의된다. 고장진단(fault diagnosis)은 고장난 시스템 동정의 소스(source)와 원인(cause)을 찾는 과정으로 정의된다. 소스와 원인은 무엇이 다른가? 특정의 밸브가 시스템 고장에 책임이 있다고 얘기되어질 수 있다. 이 경우 밸브는 고장난 시스템동정의 원인이고 소스이다. 사실 좀더 깊이

관찰해 보면 밸브에서 문제의 소스와 원인은 설계에러, 부식, 오동작 등으로 나누어질 수 있다. 이와 같은 분석 즉, 원자의 깊이, 분자의 깊이, 플랜트 조작자의 심리상태 까지 임의의 깊이로 행하여진다는 것은 명백하다. 고장시스템동정의 소스(source)는 고장난 시스템 동정에 전적으로 대응하는 시스템의 일부분인 구별가능한 물리적 부품요소로 정의한다. 그리고 고장난 시스템 동정의 원인(cause)은 부식, 오동작, 새가 엔진에 부딪히는 것과 같은 외부재해, 소스 물리적 요소를 고장으로 전개시키도록 기인한 것들로 정의한다.”⁽⁴⁾ 시스템 감시의 목표는 정상작동을 통하여 프로세스 변화와 고장들의 탐지와 다른 사건과 프로세스에 손상을 주는 것을 피하는 행동을 취하는 것이다. 이와 같은 임무를 수행하기 위하여 감시루프는 고장탐지와 진단, 고장평가, 작동여부결정, 고장제거등의 기능들을 포함하여 구현된다. 일반적으로 고장진단기법에서 가장 많이 사용되는 프로세스 파라미터에 기초한 고장진단을 위한 체계적인 수순이 Fig. 2에 보여지듯이 세 가지 단계로 구분된다.⁽⁵⁾

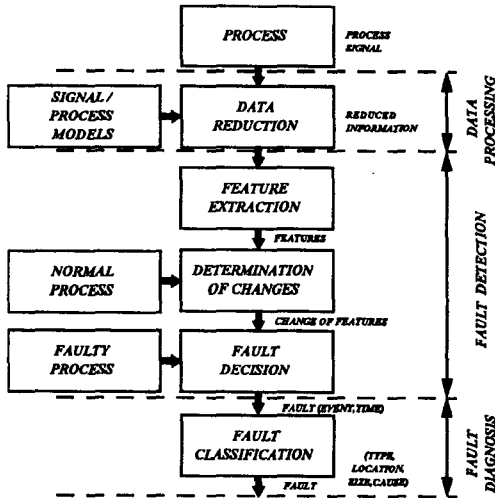


Fig. 2 Fault diagnosis procedure⁽⁵⁾

1. 데이터 처리 (data processing)

측정된 신호는 진단과 고장탐지를 위해 적합한 정보로 축소하기 위하여 필터링(filtering), 추정(estimation)등의 여러가지 신호처리 기법들에 의하여 처리된다. 예를들면, 축소된 정보는 프로세스 모델들이 적용된다면 필터링된 신호요소, 상관관계함수, 혹은 매개변수나 상태변수속에 포함되어 있다.

2. 고장탐지 (fault detection)

축소된 공정정보에 기초하여 공정속에서 고장탐지를 할 수 있게 하는 특징들(features)이 추출된다. 다음에 이러한 특징의 변화가 정상상태에 대한 참조값으로 결정된다. 이러한 변화들은 고장의 발생과 그 발생 시간을 인식하기 위하여 사용된다. 이러한 임무를 위하여 통계적 결정방법론이 사용된다.

3. fault diagnosis (고장진단)

고장사건이 감지된 후 특징들과 그들의 변화들이 고장 형태, 고장위치, 고장크기, 고장원인을 결정하기 위하여 분류과정에 입력되고, 이러한 데이터를 기반으로 고장진단 데이터 베이스속에 들어있는 데이터와 패턴매칭과 정해진 추론에 따라 구체적인 고장상태가 결정된다.

3. 복합시스템의 계층화 표현

3.1 복합시스템 구조적, 기능적 계층화

일반적으로 산업전반에 사용되는 시스템은 주로 공정프로세스와 기계 및 전기 시스템들로 구분될 수 있으며, 이러한 시스템은 인체의 구조적인 측면 혹은 생체적 측면과 비슷한 성격을 지니고 있다. 기계적 계층측면에서 보면 인체는 각 기관에 해당하는 부속시스템으로 이루어져 있고, 각 부속시스템은 작은 모듈별 유니트(unit), 유니트는 각 아이템(item)으로, 아이템은 다시 작은 요소(component)로 구성된다. 여기서 대부분의 경우 고장은 주로 최소단위인 요소 즉 인체의 일부분에서 시작하여 그 상위 레벨로 전파되고, 관찰되는 증상은 한층 높은 상위레벨에서 나타난다. 또한 공정프로세스의 경우에도 인체와 마찬가지로 어떤 하나의 질병은 그 질병을 유발시키는 다른 인체의 생체반응에 기인한다. 한 예를들면 고혈압의 경우는 식생활중 소금의 과다 섭취에 기인한 혈압의 증가, 심리적 요인에 기인한 혈압의 증가, 유전적 요인에 기인한 혈압의 증가등은 모두 생체내의 생체프로세스에 기인한 것이며, 또한 이러한 인자들은 혈압의 증가 뿐만 아니라 다른 인체시스템에 영향을 주어 질병을 유발시키며, 이러한 것들은 곧 산업시스템에서는 다중고장에 해당한다. 이러한 다중고장들은 여러개의 고장이 독립적으로 일어나는 것이 아니라 각각의 요소들간의 상호작용에 의하여 일어나는 것이다. 즉 이것은 어떤 인체내의 일부 기관에서 생긴 병이 서서히 전파되어 합병증으로 진전하는 경우이다. 그러나 실제 시스템에서는 여러개의 독립된 부속

시스템에서 동시에 고장이 일어날 수 있다.

또한 앞에서 언급한 인체시스템은 여러 가지 물리적인 현상에 의하여 운용되는 몇몇 부속시스템들로 이루어져 있는 복합시스템이며, 산업계의 대부분의 정밀시스템들은 이와같은 복합시스템들이다. 따라서 이러한 복합시스템이 제 기능을 발휘하기 위해서는 모든 부속시스템들이 그들의 기능을 올바르게 수행해야 하고, 전체가 조화있게 작동될 수 있도록 각자 서로간에 협의된 동작을 수행한다. 같은 방법으로 부속시스템들의 모든 부품과 요소들은 그들 자신의 기능을 올바르게 수행해야 하고, 부속시스템이 조화있게 작동할 수 있도록 상호간에 단계적으로 수행되어야 한다. 예를 들면, 엔진은 Fig.3에 보인 것처럼 계층의 첫 번째 레벨에서는 엔진 그 자체이다.⁽⁶⁾ 두 번째 레벨은 엔진을 구성하는 모든 부속시스템, 엔진블록, 크랭크 샤프트, 컨넥팅로드, 피스톤 어셈블리, 윤활부속시스템, 냉각부속시스템, 연료, 유도부속시스템, 점화부속시스템, 스타트부속시스템 등이다. 세 번째 레벨은 이러한 부속시스템을 구성하는 부품들 즉 점화부속시스템의 경우 배분기, 스파크 플러그 등이다. 사실 시스템의 신뢰성은 그들의 주요 부품요소들에 의존한다. 그래서 이와 같은 부품들은 엔진의 구조적 계층을 구축하는데 고려되어야 한다. Fig.4에 보이는 것처럼 엔진기능 계층에서 첫 번째 레벨은 엔진의 모든 부속시스템이 엔진이 올바르게 작동될 수 있도록 그들의 기능들을 어떤 방식으로 수행해야 하는지를 나타낸다. 두 번째 레벨에서는 몇가지 그림들로 이루어져 있는데, 부속시스템의 모든 부품들이 올바르게 작동하기 위하여 그들의 기능을 어떻게 수행해야 하는지를 나타낸다. 이와같은 복합시스템에서 일어나는 고장은 동시 다중고장을 유발하고, 고장전파가 매우 빠르다. 따라서 이러한 복합시스템의 다중고장을 진단하기 위해서는 각 시스템의 요소별 고장에 따른 고장특성을 학습한 다중신경망이 절대 필수적이다. 따라서 본 연구에서 개발된 모듈형 신경망은 이와같이 계층화되고 모듈화된 복합시스템의 고장진단에 적합할 것으로 생각된다.

본 연구의 적용대상 모델인 반응로의 프로세스들도 계층적인 관계를 가지고 있는데, 즉 시스템에 하나의 고장이 발생하면 그 고장과 관련된 파라메타의 변화가 발생한다. 즉 예를들면 명령신호의 감소나, 활성에너지 감소, 전체열전달계수 감소등의 고장이 동시에 일어날 수 있는데, 이러한 다중고장들은 그 하부계층 파라메타, 즉 열교환기 출구의 온도변화, 반응기 입출구의 온도변화, 제어기출력 변화, 반응기 입출구의 농도변화등을 유발시킨다. 따라

서 시스템의 정상상태로 부터의 이탈은 결과적으로 그 시스템내 다른 요소에 영향을 준다. 본 연구에서 사용된 반응기는 이와같은 형태로 보면 시스템 자체를 프로세스적인 측면에서는 계층적이지만 기계적인 측면에서 계층적이라기 보다 동일 레벨로 볼 수 있고, 하나의 고장이 다른 레벨에 영향을 주므로 계층적 성격이 강한 시스템이다. 본 연구에서 개발된 신경망은 이와같은 계층적 프로세스로 이루어진 화학공정 뿐만이 아니라 계층적 기계시스템에도 적용가능하리라 생각되며, 본 연구에서는 가장 기본적인 적용 형태인 반응기에 적용코자한다.

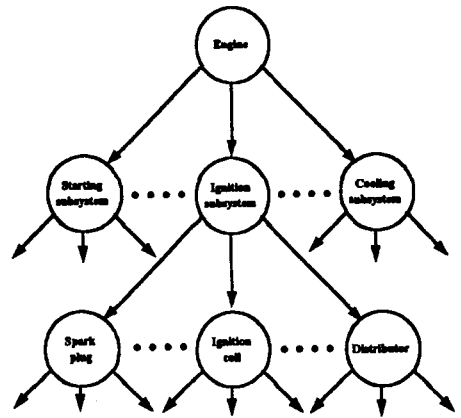
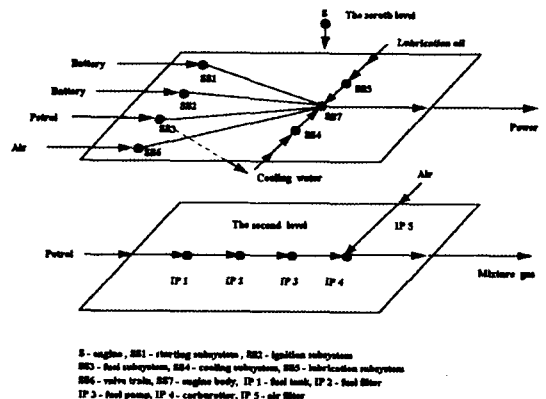


Fig. 3 Structural hierarchy of engine ⁽⁶⁾



S - engine, S1 - starting subsystem, S2 - ignition subsystem
 S3 - fuel subsystem, S4 - cooling subsystem, S5 - lubrication subsystem
 S6 - valve train, S7 - engine body, IP 1 - fuel tank, IP 2 - fuel filter
 IP 3 - fuel pump, IP 4 - carburetor, IP 5 - air filter

Fig. 4 Functional hierarchy of an engine ⁽⁶⁾

3.2 계층고장모델 (Hierarchical Fault Model)

진단추론 알고리즘에 적절한 정보는 계층구조모델과 계층기능모델, 그리고 이 두가지 사이의 상호작용에서 발견되어진다. 어떤 사실을 단순화하기 위하여 모든 적절한

고장진단정보를 포함하는 요약된 통합모델이 정의된다. 이 새로운 모델이 계층고장모델(HFM)이다.⁽⁷⁾ 모든 정보는 이 모델 속에서 자체 규정되어지거나 혹은 다른 모델에서 획득되어진다. 이와 같은 모델을 만드는 이유는 역시 진단추론기법의 이해를 높이는 것과 고장모델링 기법 표현을 단순화하기 위한 것이다. 고장모델로써 불리워지는 내용들은 계층기능모델에서 각 프로세스를 위하여 정의되어진다. 계층기능모델에서 어떤 프로세스의 고장모델은 프로세스의 고장전파 viewpoint정보, 고장정보셀 즉, 그들의 모든 부속시스템의 모든 고장모드의 알람 연결들, 요소들의 고장난 상태들과 모든 고장모드들 사이의 고장전파 링크스과 모든 부속사의 고장모드들에 대한 정보를 포함한다. 프로세스의 고장전파모델이 Fig. 5에 보여진다. 계층기능모델에서 모든 프로세스들의 고장모델들의 셀은 계층고장모델(HFM)이다.

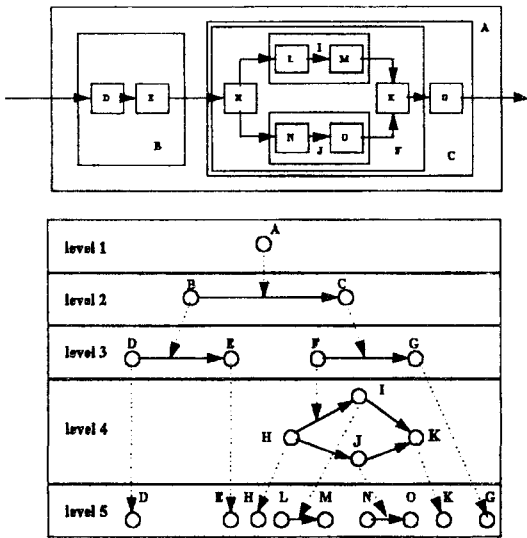


Fig. 5 Hierarchical fault propagation model

4. 고장진단기법

본 연구에서 고장진단을 위하여 제안된 과정은 다음의 두 가지 가정하에서 논의된다.

1. 고장의 결과는 그들의 정상값으로부터 시스템 파라미터들의 작은 편차로써 나타난다.
2. 모든 고장들은 고장변수들 속에서 대략 같은 상대적

인 편차로 발생한다.

Fig.1 에서 고장공간에 AND set에 대응하는 다중고장을 고려하자. 예를 들면, 모든 이중, 삼중고장에 대하여 AND set은

$$\{\{F_1F_2\}, \{F_1F_3\}, \{F_{n-1}F_n\}, \{F_1F_2F_3\}, \dots, \{F_{n-2}F_{n-1}F_n\}\}$$

고장을 진단하기 위하여 적절한 신경망이 주어진다면 진단에서의 성공도는 망을 학습하는데 사용된 데이터의 특성에 의존한다. 망이 단일고장을 진단하기 위하여 학습되어졌다면 망이 다중고장에 대한 결정을 내리기 위하여 어떻게 사용되는지를 알아보자. 앞에서 언급된 것처럼 측정값 y 는 프로세스에서 진단되는 파라미터 p 의 함수로써 표현될 수 있다고 가정한다.

$$y = f(p) \tag{1}$$

가정으로 부터

$$p = p^o + \Delta p \tag{2}$$

여기서 p^o 는 정상작동조건을 위한 파라미터들의 벡터이다. 측정값 y 는 식(2)를 식(1)에 대입함으로써 선형방정식의 셀으로 서술된다.

$$y = y_o - \Delta y = f(p^o) - A\Delta p \tag{3}$$

여기서

$$A = \left. \frac{\partial f(p)}{\partial p^T} \right|_{p=p^o} = [a_1, a_2, a_3, \dots, a_n]$$

은 상수계수를 가진 Jacobian matrix이다. $y_o = f(p^o)$ 이기 때문에 다음과 같은 선형방정식으로 표현가능하다.

$$\Delta y = a_1\Delta p_1 + \dots + a_i\Delta p_i + \dots + a_n\Delta p_n \tag{4}$$

우리는 위 식을 단일고장을 위한 학습데이터에 기초한 다중고장을 위한 분류전략을 위한 기초로 사용한다. 진단 전략의 구체적인 예시를 Fig.1에서 보인 고장공간에 속하는 AND set으로, 각기 다른 고장진단을 위한 측정

서 예상되는 출력편차의 세가지 예는 다음과 같다.

Case 1: single faults

$$\text{AND set} = \{F_1, F_2, \dots, F_n\}$$

Fault	Deviation
F_1	$\Delta y_1 = a_1 \Delta p_1$
F_2	$\Delta y_2 = a_2 \Delta p_2$
⋮	⋮
F_n	$\Delta y_n = a_n \Delta p_n$

Case 2: all of the double faults

$$\text{AND set} = \{\{F_1 F_2\}, \{F_1 F_3\}, \dots, \{F_{n-1} F_n\}\}$$

Fault	Deviation
$F_1 F_2$	$\Delta y_1 + \Delta y_2$
$F_1 F_3$	$\Delta y_1 + \Delta y_3$
⋮	⋮
$F_{n-1} F_n$	$\Delta y_{n-1} + \Delta y_n$

Case 3: double and triple faults which include F_1

$$\text{AND set} = \{\{F_1 F_2\}, \dots, \{F_1 F_n\}, \{F_1 F_2 F_3\}, \dots, \{F_1 F_{n-1} F_n\}\}$$

Fault	Deviation
F_1, F_2	$\Delta y_1 + \Delta y_2$
⋮	⋮
$F_1 F_n$	$\Delta y_1 + \Delta y_n$
F_1, F_2, F_3	$\Delta y_1 + \Delta y_2 + \Delta y_3$
⋮	⋮
F_1, F_{n-1}, F_n	$\Delta y_1 + \Delta y_{n-1} + \Delta y_n$

5. Modular Artificial Neural Networks

본 연구에서 모듈신경망(MANN)을 이용한 고장진단을 다중처리화하기 위하여 SUN workstation상에서 4개 레벨의 분산된 신경망이 동기화를 이루어 돌아가는데 여기서 각 신경망의 weight값은 off-line으로 학습되어, 신경망의 특성값과 함께 파일에 저장되어 있다. Fig.6에서 일단 runfst에 의한 진단이 시작되면(①), 프로세스는 first프로그램이 시작되면서 자기와 다른 프로그램의 동시시작을 위해서 fork()명령을 수행하면(②), first와 똑같은 이미지 프로세스가 하나가 생성되면서 하나는 first 본래의 기능을 수행하고, 새롭게 생성된 다른 하나의 프로세스는 OR프로세스로 대체된다.(③). 여기서 최상위

레벨인 first에서 최하위 레벨인 forth레벨의 각각의 요소 신경망은 이미 각각의 특성에 맞는 고장데이터들을 off-line학습한 후 최종상태의 weight값과 신경회로망 특성치를 파일 형식으로 저장하고 있다가 first와 각 second 신경망에 입력되어 고장분류 준비를 완료한 상태에서 first는 현재의 프로세스의 고장이 포함된 파라메타값을 입력받아(④) 고장을 분류하는 동안에 OR프로세스는 하부 요소부분의 학습데이터와 신경망 특성값을 입력받아 third와 forth의 신경망을 고장진단 준비를 완료한다.(⑤) first신경망이 고장진단 결과를 출력하면(⑥), OR프로세스로 메시지큐를 통하여 그 결과가 전달되고(⑦), OR프로세스는 고장이 존재하는 신경망이 포함된, 예를 들면 출력이 0010000이면 3번째 신경망을 시작시키고, 0011111이면, 3-7번째의 신경망을 시작시킨다.(⑧). 여기서 second레벨의 신경망은 Net1의 경우 F1에 관련된 이중고장만, Net7의 경우는 F7에 관련된 이중고장을 학습완료한 상태이다. 고장데이터가 입력되고(⑨), second 레벨 신경망이 수행되고(⑩), 그 결과가 다시 OR프로세스로 보내지고(⑪) OR프로세스는 second레벨의 각 신경망에서 결과를 받아 최종적으로 OR operation을 수행하여 결과를 출력한다. 본연구의 연구대상인 반응기는 구조상 second 레벨까지의 신경망의 운용으로 진단가능하다. 만일 진단된 각 고장들이 그 하부의 또 다른 프로세스 파라메타나 기타 부속요소에 기인된다면, OR프로세스는 위의 이제까지 수행된 순서와 똑같이 (①~⑪) 수행하기 위하여 세부진단에 들어간다. 여기서 third레벨은 각 고장F1~F7각각 고장을 일으키는 그 하부 요소가 7개 있다면 그 각각의 하부요소(component)를 진단하기 위하여 위와같은 수행을 반복한다. (⑫~⑳) 그리고 OR프로세스는 이제 더 이상 진단이 필요없으므로 third 레벨 신경망을 죽이고(㉑), 또한 forth 레벨 신경망도 죽인다.(㉒) 그리고 OR은 더 이상 필요하지 않는 first 신경망을 삭제시키고(㉓), 본 프로그램의 가장 부모프로세스인 runfst는 다시 OR프로세스를 마지막으로 종료시킨다(㉔). Fig.1의 고장공간속에서 단일, 이중 고장진단의 모든 경우를 고려하면

$$\{F_1, F_2, F_3, \dots, F_n, \{F_1 F_2\}, \dots, \{F_{n-1} F_n\}\}$$

이중고장의 수 n_d 그리고 전체 고장수 n_t 로 주어진다.

$$n_t = 1 + [n + (n-1) + \dots + 2 + 1] = 1 + [n(n+1)/2]$$

$$n_d = (n-1) + \dots + 2 + 1 = n(n-1)/2$$

첫 번째 레벨의 단일신경망은 전체 고장 셀으로 정상상태의 경우를 포함한 고장의 수 $1 + [n(n+1)/2]$ 개의 학습 패턴을 학습해야 한다. 첫 번째 레벨의 출력 노드에서 단일 고장의 경우는 출력값이 1인 것이 한 개가 있고, 이중 고장의 경우는 두 개가 있다. Fig.6 를 살펴보면, Net^0 으로 붙여진 첫 번째 레벨에서 모든 측정값이 입력으로 사용된다. 다시 말해서 i 번째 출력 노드가 첫 번째 레벨에서 활성화되면 두 번째 레벨에서 i 번째 Network인 Net^i 의 출력이 OR operation에서 사용되는 셀으로 형성된다. OR 프로세스는 두 번째 레벨에서 결과를 받아 세 번째 레벨의 신경망이 고장 결과에 해당하는 신경망을 활성화한다. 예를 들면, 고장 $\{F_5, F_6, F_7\}$ 이 일어날 때 첫 번째 레벨의 출력은 $\{F_5, F_7\}$ 이 되고, 두 번째 레벨에 있는 네트워크 Net^5 와 Net^7 이 활성화된다. Net^5 은 $\{F_5, F_6, F_7\}$ 을, Net^7 은 $\{F_5, F_7\}$ 을 출력한다. $\{F_5, F_6, F_7\}$ 과 $\{F_5, F_7\}$ 은 OR 프로세스로 보내지고, OR연산되어, $\{F_5, F_6, F_7\}$ 이라는 결과가 산출된다. 이 결과가 MANN의 출력이 된다.

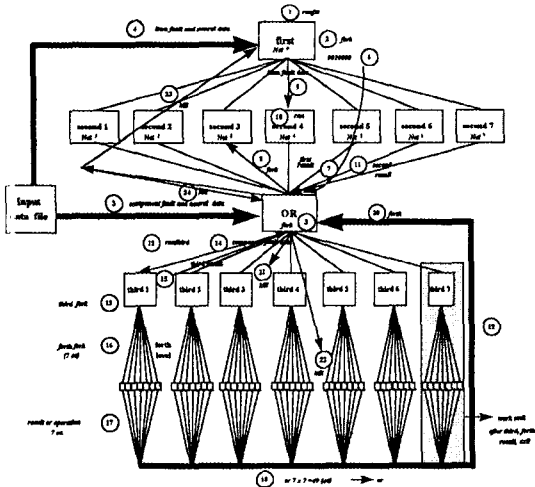


Fig. 6 MANN architecture

6. 모듈신경망의 적용예

Fig. 7은 헵탄을 톨루엔으로 반응시켜 새로운 물질을 합성하는 PI feedback controlled reactor을 나타낸 것으로 거의 정상상태 조건하에서 작동한다. 아주 극단적인 돌발적인 사고는 일어나지 않는다고 가정한다. 또한, 초기 단계에서 포함된 프로세스 고장은 강하게 억제되고 PI 제어기에 의하여 완화되도록 한다. 탱크 속에 저장된 헵

탄은 프로세스 펌프 1을 통하여 반응로로 보내지고, 반응로 속에 일어나는 작용으로서 반응비율은 반응로 속의 온도에 의하여 조절된다. 증기는 재생펌프 2를 통하여 열교환기로 공급된다. 열교환기의 출구값은 PI 제어기에 의하여 제어되는데, 제어되는 변수는 반응온도이다.

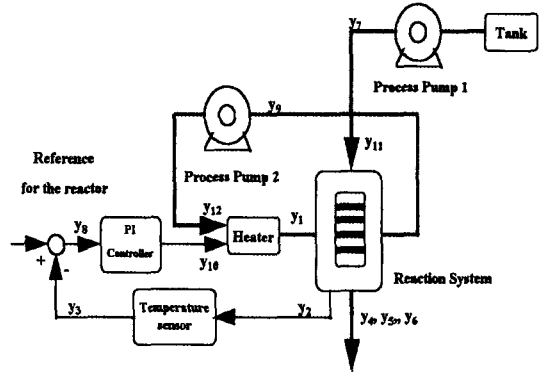


Fig. 7 Reactor system for MANN application

여기서 열교환기의 제어는 전자제어회로나 컴퓨터를 통하여 이루어 진다고 가정한다. 그리고, 여기서는 작동유체의 상태에만 관심이 있으므로, 제어기에서의 고장은 포함시키지 않는다. MANN의 적용으로써 7가지 가능한 고장예를 고려하면 다음과 같다.

- $F_1 = 0.1\%$ 감소 명령 신호 u_0
- $F_2 = 2.0\%$ 감소 주파수 요소 k_0
- $F_3 = 0.1\%$ 감소 활성화에너지 E_a
- $F_4 = 2.0\%$ 감소 열교환기 계인 k
- $F_5 = 2.0\%$ 감소 전체 열전달계수 h
- $F_6 = 1.0\%$ 감소 입구에서의 헵탄
- $F_7 = 1.0\%$ 감소 출구에서의 증기 반응온도 T_i

파라미터속에서 이와같은 변화들은 같은 정도(2.0%)의 프로세스 변수속에 오차를 유발한다. 고장공간은 7차원($n=7$)이고, 상관관계 분석에 의해서 측정 공간속에서 고장진단에 필요한 변수는 다음과 같이 7개만으로 가능해진다.

- $y_1 =$ 열교환기 출구 온도 T_1
- $y_2 =$ 반응기의 출구 온도 T
- $y_3 =$ PI 제어기의 출력 s_i

- y_4 = 반응기 출구 물질 농도 1
- y_5 = 반응기 출구 물질 농도 2
- y_6 = 반응기 입구 물질 농도
- y_7 = 반응기 입구 온도 T_i

본 연구에서는 최소의 측정으로 최소의 오분류를 가진 고장을 진단하는 것이다. 처음에는 문제속에 포함된 모든 변수를 전부 선택했다. 그러나 측정의 전처리에 의하여 네트워크의 입력을 최소로 할 필요가 있다. 측정을 적당하게 조절하기 위하여 프로세스 측정값의 모든 오차 Δy_i 는 측정값의 2%로 하여 정규화된다. 변화계수는 2%이기 때문에 몇몇 노이즈가 더해진다면 분류는 올바르게 수행된다. Table 1에서 학습데이터는 프로세스 측정의 정규화와 데이터 축소(reduction)에 의하여 얻어진다. 이와 같은 데이터는 Net^0 를 학습하는데 사용된다. 고장 F_i 이 존재할 때, 출력은 {1 0 0 0 0 0}이다. 7개의 고장에 대응한 7개의 네트워크 Net^i ($i=1,2,3,\dots,7$)가 두 번째 레벨에 존재한다. Net^i 를 위한 학습데이터는 앞에서 설명했듯이 두개의 고장을 위하여 Table 1에 더함으로써 얻을 수 있다. Table 2에서 F_1 을 포함하는 모든 이중고장의 경우를 조사하자. 예를들면 $\{F_1, F_2\}$ 를 위한 학습데이터는 Table 1에서 2행(F_1 에 대응하는 고장)데이터를 3행(F_2 에 대응고장)에 더함으로써 발생되고 출력은 {1 1 0 0 0 0}이 된다. 본 연구에서 사용된 신경망의 특성은 각 계층별 신경망의 입력은 7개, 은닉층은 30개, 출력층은 7개이고, $\eta = 0.1$, $\alpha = 0.9$ 로 세팅하고, 반복은 최소에러가 0.008보다 적을때 까지 계속된다.

Table 1 Training Patterns for Single Faults

Fault	Input Patterns							Output Patterns						
	y_1	y_2	y_3	y_4	y_5	y_6	y_7	1	2	3	4	5	6	7
Normal	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0	0	0	0	0	0	0
F_1	-1.85	-0.74	-0.23	2.78	-2.78	0.00	0.00	1	0	0	0	0	0	0
F_2	-0.83	0.00	-0.17	2.52	-2.52	0.00	0.00	0	1	0	0	0	0	0
F_3	0.66	0.00	0.13	-2.03	2.03	0.00	0.00	0	0	1	0	0	0	0
F_4	3.04	0.00	0.74	-2.03	2.03	0.00	0.00	0	0	0	1	0	0	0
F_5	-0.86	0.00	0.14	-5.11	-5.00	0.00	0.00	0	0	0	0	1	0	0
F_6	-1.42	-0.74	-0.14	2.78	-2.78	0.00	-3.00	0	0	0	0	0	1	0
F_7	0.43	0.00	0.09	0.00	0.00	0.00	-3.00	0	0	0	0	0	0	1

요구되는 반복수 평균은 6800이고, 학습패턴의 수는 정상상태 1, 단일고장 7, 이중고장 21개로 총 29개였다. 비록 학습은 단일, 이중고장에 기초하고 있지만, 테스트는 단일, 이중, 삼중고장들에 의하여 수행된다. Output layer 속에 first level의 노드출력이 0.5보다 클 때 정후가 존재하는 것으로 고려하고, second level에서 출

Table 2 Training Patterns for Double Fault

Fault	Input Patterns							Output Patterns						
	y_1	y_2	y_3	y_4	y_5	y_6	y_7	1	2	3	4	5	6	7
F_1	-1.85	-0.74	-0.23	2.78	-2.78	0.00	0.00	0	0	0	0	0	0	0
F_1F_2	-2.68	-0.74	0.403	5.30	-5.30	0.00	0.00	1	1	0	0	0	0	0
F_1F_3	-1.19	-0.74	-0.10	0.75	-0.75	0.00	0.00	0	1	0	1	0	0	0
F_1F_4	-1.85	-0.74	0.68	2.78	-2.78	0.00	0.00	1	0	0	0	1	0	0
F_1F_5	1.19	-0.74	0.38	2.78	-2.78	0.00	0.00	0	1	0	0	0	1	0
F_1F_6	-2.71	-0.74	-0.40	0.40	-5.37	-5.00	0.00	1	0	0	0	0	0	1
F_1F_7	-1.42	-0.74	-0.14	2.78	-2.78	0.00	-3.00	1	0	0	0	0	0	1
F_2	-0.83	0.00	-0.17	2.52	-2.52	0.00	0.00	0	1	0	0	0	0	0
F_2F_3	-2.68	-0.74	0.403	5.30	-5.30	0.00	0.00	1	1	0	0	0	0	0
F_2F_4	-0.17	0.00	-0.04	0.49	-0.49	0.00	0.00	0	1	1	0	0	0	0
F_2F_5	-0.83	0.00	0.74	2.52	-2.52	0.00	0.00	0	1	0	1	0	0	0
F_2F_6	2.21	0.00	0.44	2.52	-2.52	0.00	0.00	0	1	0	0	1	0	0
F_2F_7	-1.69	0.00	-0.34	0.14	-5.11	-5.00	0.00	0	1	0	0	0	0	1
F_3	-0.66	0.00	0.13	-2.03	2.03	0.00	0.00	0	0	1	0	0	0	0
F_3F_4	-1.19	-0.74	-0.10	0.75	-0.75	0.00	0.00	1	0	1	0	0	0	0
F_3F_5	-0.17	0.00	-0.04	0.49	-0.49	0.00	0.00	0	1	1	0	0	0	0
F_3F_6	0.66	0.00	1.04	-2.03	2.03	0.00	0.00	0	0	1	1	0	0	0
F_3F_7	3.70	0.00	0.74	-2.03	2.03	0.00	0.00	0	0	1	0	1	0	0
F_4	-1.69	0.00	-0.04	-4.41	-4.56	-5.00	0.00	0	0	1	0	0	1	0
F_4F_5	1.09	0.00	0.22	-2.03	2.03	0.00	-5.00	0	0	1	0	0	0	1
F_5	0.00	0.00	0.91	0.00	0.00	0.00	0.00	0	0	0	1	0	0	0
F_5F_6	-1.85	-0.74	0.68	2.78	-2.78	0.00	0.00	1	0	0	1	0	0	0
F_5F_7	-0.83	0.00	0.74	2.52	-2.52	0.00	0.00	0	1	0	1	0	0	0
F_6	0.66	0.00	1.04	-2.03	2.03	0.00	0.00	0	0	1	1	0	0	0
F_6F_7	3.04	0.00	1.52	0.00	0.00	0.00	0.00	0	0	0	1	1	0	0
F_7	-0.86	0.00	0.74	-2.38	-2.59	-5.00	0.00	0	0	0	1	0	1	0
F_7F_8	0.43	0.00	1.00	0.00	0.00	0.00	-3.00	0	0	0	1	0	0	1
F_8	3.04	0.00	0.61	0.00	0.00	0.00	0.00	0	0	0	0	1	0	0
F_8F_9	1.19	-0.74	0.58	2.78	-2.78	0.00	0.00	1	0	0	0	1	0	0
F_8F_{10}	2.21	0.00	0.44	2.52	-2.52	0.00	0.00	0	1	0	0	1	0	0
F_8F_{11}	3.70	0.00	0.74	-2.03	2.03	0.00	0.00	0	0	1	0	1	0	0
F_8F_{12}	3.04	0.00	1.52	0.00	0.00	0.00	0.00	0	0	0	1	1	0	0
F_8F_{13}	-0.86	0.00	0.44	-2.38	-2.59	-5.00	0.00	0	0	0	0	1	1	0
F_8F_{14}	3.47	0.00	0.70	0.00	0.00	0.00	-3.00	0	0	0	0	1	0	1
F_9	-0.86	0.00	-0.17	-2.38	-2.59	-5.00	0.00	0	0	0	0	0	0	1
F_9F_{15}	-2.71	-0.74	-0.40	0.40	-5.37	-5.00	0.00	1	0	0	0	0	0	1
F_9F_{16}	-1.19	0.00	-0.34	0.14	-5.11	-5.00	0.00	0	1	0	0	0	0	1
F_9F_{17}	-1.69	0.00	-0.04	-4.41	-4.56	-5.00	0.00	0	0	1	0	0	0	1
F_9F_{18}	-0.86	0.00	0.74	-2.38	-2.59	-5.00	0.00	0	0	0	0	1	0	1
F_9F_{19}	2.18	0.00	0.44	-2.38	-2.59	-5.00	0.00	0	0	0	0	0	1	1
F_9F_{20}	-0.43	0.00	-0.08	-2.38	-2.59	-5.00	-3.00	0	0	0	0	0	1	1
F_{10}	-0.43	0.00	0.09	0.00	0.00	0.00	-3.00	0	0	0	0	0	0	1
$F_{10F_{21}}$	-1.42	-0.74	-0.14	2.78	-2.78	0.00	-3.00	1	0	0	0	0	0	1
$F_{10F_{22}}$	-0.40	0.00	-0.08	2.52	-2.52	0.00	-3.00	0	1	0	0	0	0	1
$F_{10F_{23}}$	1.09	0.00	0.22	-2.03	2.03	0.00	-3.00	0	0	1	0	0	0	1
$F_{10F_{24}}$	0.43	0.00	1.00	0.00	0.00	0.00	-3.00	0	0	0	1	0	0	1
$F_{10F_{25}}$	3.47	0.00	0.70	0.00	0.00	0.00	-3.00	0	0	0	0	0	1	1
$F_{10F_{26}}$	-0.43	0.00	-0.08	-2.38	-2.59	-5.00	-3.00	0	0	0	0	0	0	1

력이 0.9보다 큰 경우에 아이템 고장이 존재하는 것으로 하고, third level에서는 노드출력이 0.5이상이면 component속에 고장징후가 존재하고, fourth level에서는 출력노드가 0.9 이상이면 component에 고장이 존재하는 것으로 한다.

7. 고장진단 결과 및 고찰

본 연구에서는 단일고장과 이중고장에 대해서만 학습을 수행하고, 단일고장과 이중고장, 그리고 삼중고장에 대해서 진단을 수행하였다. 테스트할 패턴수는 정상 1개, 단일 7개, 이중 21개, 삼중 35개로 총 64개이다. 정상상태는 한가지 뿐이고, 나머지는 7개의 고장중에 감지될 고장 요소의 갯수를 조합으로 계산한 것이다. 위의 단일, 이중, 삼중고장에 대하여 진단을 수행한 결과 단일고장에서는

7가지의 고장요소($F_1, F_2, F_3, F_4, F_5, F_6, F_7$)를 포함하고 있는 입력데이터에 대하여 모두(100%) 감지 가능할 수 있었고,

Table 3의 이중고장 진단에서는 21가지의 이중고장 요소를 각각 포함하고 있는 입력데이터에 대해 4개를 제외한 고장을 감지(91%) 할 수 있었으며, 이중 가장 복잡한

Table 3 Diagnosing results of double fault

Fault	After First Level	After Second Level	After OR Operation	Classification*
F_1F_2	F_1F_2	$F_1F_2 \cup F_1F_3$	F_1F_2	2/2
F_1F_3	F_1	F_1F_3	F_1F_3	2/2
F_1F_4	F_1	F_1F_4	F_1F_4	2/2
F_1F_5	F_1	F_1F_5	F_1F_5	1/2
F_1F_6	F_1F_6	$F_1F_6 \cup F_1F_7$	F_1F_6	2/2
F_1F_7	F_1	F_1F_7	F_1F_7	2/2
F_2F_3	F_2	F_2F_3	Impossible	0/2
F_2F_4	F_2F_4	$F_2F_4 \cup F_2F_5$	F_2F_4	2/2
F_2F_5	F_2	F_2F_5	F_2F_5	2/2
F_2F_6	F_2F_6	$F_2F_6 \cup F_2F_7$	F_2F_6	2/2
F_2F_7	F_2	F_2F_7	F_2F_7	2/2
F_3F_4	F_3	F_3F_4	F_3F_4	2/2
F_3F_5	F_3	F_3F_5	F_3F_5	2/2
F_3F_6	F_3	F_3F_6	F_3F_6	2/2
F_3F_7	F_3	F_3F_7	F_3F_7	2/2
F_4F_5	F_4	F_4F_5	F_4F_5	2/2
F_4F_6	F_4F_6	$F_4F_6 \cup F_4F_7$	F_4F_6	2/2
F_4F_7	F_4	F_4F_7	F_4F_7	1/2
F_5F_6	F_5	F_5F_6	F_5F_6	2/2
F_5F_7	F_5F_7	$F_5F_7 \cup F_6F_7$	F_5F_7	2/2

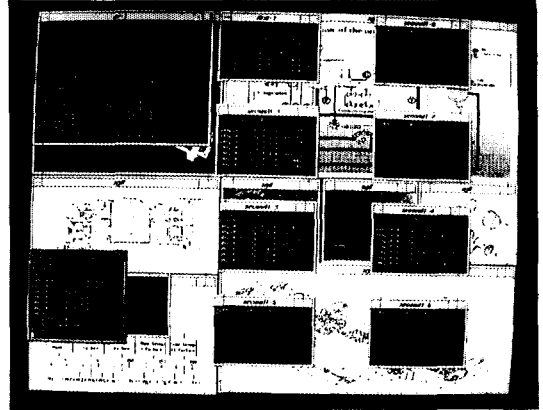


Fig. 8 MANN 운용예

Table 4 Diagnosing results of triple fault

Fault	After Item Level	After Component Level	After OR Operation	Classification*
$F_1F_2F_3$	F_1	F_1	F_1	1/3
$F_1F_2F_4$	F_1	F_1F_4	F_1F_4	2/3
$F_1F_2F_5$		Impossible	Impossible	0/3
$F_1F_2F_6$	F_1	F_1F_2	F_1F_2	2/3
$F_1F_2F_7$	F_1F_7	$F_1F_2 \cup F_1F_7$	F_1F_7	2/3
$F_1F_3F_4$	F_4	F_4	F_4	1/3
$F_1F_3F_5$	F_5	F_5	F_5	1/3
$F_1F_3F_6$	F_6	F_6	F_6	1/3
$F_1F_3F_7$	F_1	F_1F_3	F_1F_3	2/3
$F_1F_4F_5$	F_4	F_2F_4	F_2F_4	1/3
$F_1F_4F_6$	F_6	F_1F_6	F_1F_6	2/3
$F_1F_4F_7$	F_1	$F_1F_4F_7$	$F_1F_4F_7$	3/3
$F_1F_5F_6$		Impossible	Impossible	0/3
$F_1F_5F_7$	F_2F_7	$F_2F_5 \cup F_2F_7$	$F_2F_5F_7$	2/3
$F_1F_6F_7$	$F_1F_6F_7$	$F_1F_6F_7 \cup F_1F_6F_5$	$F_1F_6F_7$	3/3
$F_2F_3F_4$	F_4	F_4	F_4	1/3
$F_2F_3F_5$	F_5	F_5	F_5	1/3
$F_2F_3F_6$	F_6	F_6	F_6	1/3
$F_2F_3F_7$	F_7	F_7	F_7	1/3
$F_2F_4F_5$	$F_2F_4F_5$	$F_2F_4F_5 \cup F_2F_4F_7$	$F_2F_4F_5$	3/3
$F_2F_4F_6$	F_4F_6	$F_4F_6 \cup F_4F_5$	F_4F_6	2/3
$F_2F_4F_7$	F_2F_4	$F_2F_4F_7 \cup F_2F_4F_5$	$F_2F_4F_7$	3/3
$F_2F_5F_6$	F_5	F_5F_6	F_5F_6	2/3
$F_2F_5F_7$	F_2F_5	$F_2F_5F_7 \cup F_2F_5F_6$	$F_2F_5F_7$	3/3
$F_2F_6F_7$	F_6F_7	$F_2F_6F_7 \cup F_6F_7$	$F_2F_6F_7$	3/3
$F_2F_7F_3$	$F_2F_7F_3$	$F_2F_7F_3 \cup F_2F_7F_5$	$F_2F_7F_3$	3/3
$F_2F_7F_4$	F_4F_7	$F_4F_7 \cup F_4F_6$	F_4F_7	3/3
$F_2F_7F_5$	F_2F_7	$F_2F_7 \cup F_2F_6$	F_2F_7	2/3
$F_2F_7F_6$	$F_2F_7F_6$	$F_2F_7F_6 \cup F_2F_7F_5$	$F_2F_7F_6$	3/3
$F_2F_7F_7$	F_7	F_7	F_7	2/3
$F_3F_4F_5$	F_4F_5	$F_4F_5 \cup F_4F_6F_7$	$F_4F_5F_6$	3/3
$F_4F_5F_6$	F_4F_5	$F_4F_5F_6 \cup F_4F_5F_7$	$F_4F_5F_6$	3/3
$F_4F_6F_7$	F_4F_6	$F_4F_6F_7 \cup F_4F_6F_5$	$F_4F_6F_7$	3/3
$F_5F_6F_7$	F_5F_6	$F_5F_6F_7 \cup F_5F_6F_5$	$F_5F_6F_7$	3/3

* Number of faults diagnosed correctly / number of faults occurring.
 F_2 = misclassification

다중고장의 형태인 삼중고장의 결과를 Table 4에 나타내었다. 삼중고장에서는 35가지 입력 데이터중에 76% ($\frac{39}{51}$)의 고장인식율을 가지고 있다. Table 3.4에서 이탤릭체로 진하게 표시된 F_2 는 모델링 에러나 상관관계함수의 오용 등으로 기인한 오분류이고, Impossible로 표시된 것은 존재하는 고장요소를 감지하지 못한 경우이다. Fig. 8은 본 연구에서 사용된 하드웨어장비인 워크스테이션에서 MANN의 구현예를 나타낸 것이다. 여기서 각각의 작은 창(window)은 하나이 신경망이 독립적인 프로세스로 돌아가는 가정을 나타낸 것이다.

8. 결론

본 연구에서는 다중고장 진단을 하기 위하여 사용될 수 있는 UNIX 운영체제하에서 멀티태스킹으로 IPC(Inter Process Communication)설비를 이용한 모듈화신경망(MANN)이라 불리는 neural network가 제시되었다. MANN을 반응로의 고장진단에 적용한 결과 다음과 같은 결론을 내린다.

첫째, MANN은 기존의 신경망이 단일, 이중, 삼중고장에 대한 데이터를 전부 학습하는 반면에 본 연구에서 제시된 신경망은 각각에 요소고장이 포함된 독립적인 신경망을 선별하여 학습하므로써 다중고장 인식율을 높일 수 있다. 또한, MANN은 인체의 신경망을 계층화하여 표현함으로써 각 프로세스는 독립적으로 수행될 수 있고, 다음에 수정시 필요한 부분만 수정이 가능하며 또한 계층화되고 다중고장을 유발하는 다른 복합시스템에 다중고장 가능할 것으로 생각되며, 본 MANN을 전문가시스템이나 다른 모델베이스 고장진단과 연결 운용이 가능할 것으로 사료된다.

둘째, 개발된 MANN을 반응로의 고장진단 예로부터 다음과 같은 결론을 얻었다. MANN은 7개의 단일고장은 100%, 21개의 이중고장의 경우는 91%, 35개의 삼중고장의 경우는 73%로 매우 높은 다중고장 진단을 할 수 있었다. 또한 다중 동시고장과 관계된 추론은 단일고장에 대한 기본적인 데이터만으로 부터 발생될 수 있다.

참고 문헌

1. Watanabe, K. et al., "Incipient Fault Diag-

nosis of Nonlinear Processes with Multiple Causes of Faults", Chem, Eng., Sci., vol.39, pp.491, 1983.

2. Khaksari, M., "Expert diagnostics system", First Int. Conf. Industrial and Engineering Applications of AI and Expert Systems, Tulsa, Oklahoma, pp.43-53, 1988.

3. Iserman, R., "Process Fault Detection based on Modeling and Estimation Method - A Survey", Automatica, Vol.20, pp.387-394, 1984.

4. Padalkar, S., "Graph Based Real-Time Fault Diagnostics", Proceeding of the 4th Conference on AI for Space Applications, pp.115-123, 1988.

5. Iserman, R., "Process Fault Diagnosis based on Dynamic Models And Parameter Estimation Methods", Fault Diagnosis in Dynamic Systems : Theory and Applications, Prentice Hall, pp.253-291, 1989.

6. Zheng, X., Yang, Z. and Shi, H., "A Knowledge-Based Diagnosis System Automobile Engines", International Journal of Advanced Manufacturing Technology, Vol.3, No.3, 1988.

7. Narayanan, N., "A methodology for knowledge acquisition and reasoning in failure analysis of systems", IEEE Transaction on Systems, Man and Cybernetics, Vol.17, No.2, pp.274-pp.288, 1987. 8.

8. Bach, M.J., The Design of the UNIX Operating System, Prentice Hall, pp.105-237, 1986.

9. Stevens, W.R., Advanced Programming in the Unix Environment, Addison Wesley, pp.188-212, 1992.

10. 홍릉과학출판사 편집부, UNIX 시스템 프로그래밍, 홍릉과학출판사, 1991.

11. 원유현, 백정현, UNIX 시스템 프로그래밍, 정익사, 1995.