韓國精密工學會 '95年度 秋季學術大會論文集 pp.670~675

에이전트기반 실시간 고장진단 시뮬레이션기법
(Agent based real-time fault diagnosis simulation)

배용환(Bae,Y.H)**, 이석희(Lee,S.H)*, 배태용(Bae,T.Y)**, 이형국(Lee, H.K.)**

* 부산대학교 생산기계공학과

** 부산대학교 생산기계공학과 대학원

Abstract

This paper describes a fault diagnosis simulation of the Real-Time Multiple Fault Diagnosis System (RTMFDS) for forcasting faults in a system and deciding current machine state from signal information. Comparing with other diagnosis system for single fault, the system developed deals with multiple fault diagnosis, comprising two main parts. One is a remote signal generating and transimission terminal and the other is a host system for fault diagnosis. Signal generator generate the random fault signal and the image information, and send this information to host. Host consists of various modules and agents such as Signal Processing Module(SPM) for sinal preprocessing, Performence Monitoring Module(PMM) for subsystem performance monitoring, Trigger Module(TM) for multi-triggering subsystem fault diagnosis, Subsystem Fault Diagnosis Agent(SFDA) for receiving trigger signal, formulating subsystem fault D/B and initiating diagnosis, Fault Diagnosis Module (FDM) for simulating component fault with Hierarchical Artificial Neural Network(HANN), numerical models and Hofield network, Result Agent(RA) for receiving simulation result and sending to Treatment solver and Graphic Agent(GA). Each agent represents a separate process in UNIX operating system, information exchange and cooperation between agents was done by IPC(Inter Process Communication : message queue, semaphore, signal, pipe). Numerical models are used to describe structure , function and behavior of total system, subsystems and their components. Hierarchical data structure for diagnosing the fault system is implemented by HANN. Signal generation and transmittion was performed on PC. As a host, SUN workstation with X-Windows(Motif) is used for graphic representation.

1. 서 론

우리가 앞에서 살펴본 기존의 고장진단시스템은 대부분이 유연성이 결여되어 사고가 일어나야만 그때부터 진단을 시작한다. 향후 자동화된 진단시스템의 발전은 인간의 감각기관과 추론형태를 닮아가야만 한다. 인간의 추론 장점은 현재 자신의 기억영역속에 저장되어 있는 지식을 스위칭하므로써 얼마든지 유연성 있는 진단을 가능하게 한다. 인간의 감각시스템은 모든 사물대상에 대하여 항상 열려진 상태로 주위을 감시(monitoring)한다. 만약 인간의 어떤 부위에 물리적인 고통이 가해지면 인간은 감각적인 신호(전기적신호)에 의하여 해당부분을 감시하다가 그보다 더 큰 충격이나 다른 사건이 발생하면 이것을 멈추고 새로운 곳으로 사고를 급히 전이한다. 그러나 인간의 유연성은 뛰어나지만 한꺼번에 강도가 같은 자극이 들어오면 그것에 대하여는 대처할 방법이 없다. 그러나 컴퓨터시스템의 특징은 내부작동 알고리즘의 복잡함에 기인하여 비록 인간처럼 그렇게 뛰어난 유연성을 갖는 추론은 할 수 없지만 인간보다는 빠르게 계산하는 능력 때문에 또한 유용하다. 따라서 인간의 추론도 결국은 이전에 경험에 의하여 학습되어진 지식을 바탕으로 추론하므로 그 중에서 특정분야를 컴퓨터에 이식시키고 인간이 기억하기 어려운 많은 고장정보를 컴퓨터내부에 저장하고, 또한 인간의 능력으로 할 수 없는 복잡한 계산을 컴퓨터가 대신함으로써 좀더 지적인 고장진단시스템이 개발될 가능성이 있다. 일반적으로 인간진단전문가나 의사들은 인체의 질병이나 기계의 고장을 진단할 때 이용하는 정보는 주로 구조적, 기능적, 동적정보로써 자기의 경험을 바탕으로 계층화시키고, 진단시 이를 회수하여 계층적 특성에 따라 진단을 해간다. 여기서도 알 수 있듯이 계층화된 구조의 장점은 복잡하지만 시스템을 구성하는 각각의 요소들간에 상호연관성이 있을때 그 연관성을 회수하여 분석해 감으로써 최종적으로 고장요소를 찾아낼 수 있다. 복합시스템(complex system)은 각각 주어진 특정의 기능을 수행하고, 또한 그 메카니즘이 각각 다르므로 이들의 고장을 진단하기 위해서는 각각 분리된 지식영역(knowledge domain)을 가지고 있어야 한다. 따라서 또한 인간의 한 사람이 이 복잡한 시스템을 동시에 관찰할 수 없으므로 각 분야별 고장전문가를 설정하여 진단한다. 이것은 인체의 경우 의사들의 전공과목이 외과,내과,이비후과, 정신과로 분리되어 있는 것과 같은 이치이다. 기계와 인체의 각 부분은 비밀스럽게 특정의 자기고유의 어떤 방식(모델)을 가지고 운용되는데 이 모델은 인간의 경우는 어떤 계산, 감정, 관습, 도덕 등이고, 기계의 경우는 수식모델, 물리모델, 신경모델등이 될 수 있고 이러한 모델들은 매우 많은 정보를 포함하고 있으므로, 정보를 빠르게 처리하기 위해서는 프로세스의 다중처리(multitasking)가 필요하고, 또한 각각의 다른 형태의 정보를 모듈별로 처리함으로써 진단시스템의 효율성과 구조를 간략화시킬 수 있지만 이들 사이의 인관관계는 각 프로세스의 협조(cooperation)를 통하여 이루어져야만 한다. 본 연구에서는 시스템에서 들어오는 센서의 고장정보를 이용하여 성능체크 후 고장진단엔진을 트리거하고 특정의 신호패턴처리기를 이용하여 특정패턴을 추출한 다음 고장학습이 된 계층신경망에 입력하여 고장요소를 분리해내는 시스템으로 수십개의 독립된 프로세스로 운영되며, 특정의 업무를 수행하는 프로세스단위를 agent라 하며, 각 에이전트는 다른 에이전트와

상호 통신하며 정보를 주고 받는다. 여기서 각 프로세스는 필요에 따라 다른 프로세스와 동기화가 가능하다. 이와같은 기능들은 UNIX운영체제의 IPC설비를 이용하여 구현되었다. 기존의 고장진단시스템은 수정시 전체시스템을 수리해야 하고, 어떤 기능을 추가하고자 할 때 그 시스템을 초기에 만든 설계자만이 가능하지만, 모듈화된 고장진단시스템의 장점은 특정의 분리된 고장진단시스템과 각각 그 독립성을 유지하면서 필요시 협조할 수 있고, 또한 시스템은 다른 시스템이 어떻게 운행되어도 상관이 없고, 내부구조의 획일적인 통일성이 없어도 된다. 또한 기존의 시스템들은 항상 진단프로세스가 메인 메모리속에 존재하면서 수행되는데 비하여 본 시스템은 특정 진단스케쥴러가 필요시 특정 모듈을 로드하거나 제거함으로써 메모리를 동적으로 활용할 수 있어 진단의 효율성을 높일수 있는 프로토타입형의 실시간다중고장진단시스템 (Real-Time Multipule Fault Diagnosis System : RTMFDS) 이다.

2. 인체와 기계시스템의 계층화 표현

일반적으로 인체는 Fig. 1과 같이 아주 발달된 계층화구조를 이루고 있는데 이러한 계층화구조는 모든 생태계에서 일어나는 활동의 모습이며, 또한 조직화된 모습이다. 이러한 계층구조의 특징은 상위레벨제어권에서 하위레벨을 효율적으로 제어할 수 있는 체계이다. 즉 인간의 모든 신체활동정보는 신경을 통하여 뇌로 전달되고, 뇌에서는 이제까지의 경험과 자가모델에 의하여 상황을 판단하고, 진단 및 조치사항을 수립한다. 뇌의 고장인식 과정은 특이하다. 뇌는 주로 신체 각 부위를 특수한 뇌신경이 병렬처리를 하는 것으로 알려져 있다. 이러한 병렬분산처리를 컴퓨터에서 구현하기 위해서는 시스템은 다중처리방식을 제공해야 한다. 다음은 기계의 각 시스템의 구조적인, 기능적, 동적 특성에 대하여 살펴보자. 기계시스템도 인체와 마찬가지로 각 기관에 해당하는 부속시스템으로 이루어져 있고, 각 부속시스템은 작은 모듈별 유니트(unit), 유니트는 각 아이템(item)으로, 아이템은 다시 작은 요소(component)로 구성된다. (Fig. 2) 여기서 대부분의 경우 고장은 주로 최소단위인 요소에서 시작하여 그 상위레벨로 전파되고, 관찰되는 증상은 한층 높은 상위레벨에서 나타난다. 이와같은 계층구조를 이용하면, 고장시 의심가는 요소의 부속구조만을 탐색할 필요가 있고, 이중에서 고장을 탐지해 내거나 의심의 소지가 없는 요소들을 제거할수 있으므로 고장진단을 위하여 아주 효율적이다.

3. 에이전트와 블랙보드시스템

3.1 Agent System

본 연구에서는 에이전트와 블랙보드 기법을 이용하여 실시간 고장진단시스템을 구축하는데 에이전트는 블랙보드 구조를 기본으로 하고 있다. 에이전트 구조에서는 클라이언트 개념에 해당하는 에이전트들이 블랙보드에 해결할 작업을 기록하는 식으로 진행된다.[1](Fig. 3) 블랙보드는 parent agent에 의하여 제어된다. 다음에 parent agent와 child agent의 역할을 알아보자. Parent agent의 역할은 다음과 같다. 첫째, 일종의 글로벌 데이타를 저장관리하고, 어느 에이전트가 어떤 일을 할 수 있는지 알아낼 수 있다. 분산적으로 사용자 요구를 처리할때 child agent들간의 정보교류를 스케쥴링하고 관리한다. 둘째, parent agent는 black board를 포함하며 이를 관리하고, child agent간의 모든 교류는 블랙보드를 통하여 이루어 진다. Agent 들간의 모든 교류는 블랙보드를 통해서만 이루어진다. 에이전트들간의 교류를 위한 메세지(Inter Communication Language, ICL)는 Prolog형태와 유사한 모양의 표현방식을 선택문제를 해결할 때 쓰인다. Parent agent의 주된 임무는 ICL을 분석하여 이를 해결해 줄수 있는 child agent에게 전달해 주는 일종의 중개자 역할을 하며 메세지에 따라서 임의의 방향으로 진행한다. Child agent의 역할은 다음과 같다. 첫째, child agent는 블랙보드와의 교류를 위한 통신층과 지식베이스층으로 구성된다. 지식베이스는 C, Prolog, LISP등 언어로 구성된다. 둘째, 각각의 에이전트들은 사용자나 다른 에이전트의 요구를 만족시켜 주기 위하여 정보를 주거나 어떤 행위를 할 수있다. 또한 조건이 만족되는지를 감시하기 위하여 트리거를 설치할 수 있다. 트리거란 조건부와 행위부로 나누어지는 데이타의 형태이다. 트리거의 조건부는 일정시간을 주기로 만족여부를 결정하기 위하여 조회된다. 트리거 조건부는 블랙보드메세지(예: "X 에이전트가 Y 작업을 완료하면"), 블랙보드데이타 (예: "X가 블랙보드에 기록되면") 또는 특정에이전트의 조건(예: "X에 관한 메일이 도착하면")에 관련된 것들이다.

3.2 Blackboard System

지능제어시스템에서 다양한 지식베이스 시스템을 하나로 묶어서 실시간으로 그들을 작동하게 할 필요가 생길 것이다. 이와 같은 연결에서 고려되는 문제는 지식베이스의 실시간 능력이다. 지식베이스 시스템의 공통적인 통신수단으로써 소위 "black-board"가 개발되었다. 블랙보드기반시스템은 때때로 차세대 지식베이스 시스템이라 일컬어진다. 블랙보드시스템은 에이전트간에 정보를 전달해주는 가상메모리로 정의되며 3가지 요소들을 가진다.[2] Knowledge sources(KSs) 는 문제해결에 필요한 지식과 추론을 포함하는 독립모듈들이다. KSs는 표현과 추론기법에 따라 다양화 될 수 있다. 지식소스들은 절차(produces), 일련의 룰(set of rules)혹은 논리적 추론들이다. KSs는 전문가시스템에 의하여 사용된 각개 규칙들보다는 훨씬 큰 입도를 가진다. 전문가 시스템이 어떤 자극에 의하여 규칙을 점화시키므로써 작동하는 반면에 블랙보드시스템은 전체지식모듈이나 KS, 전문가시스템, neural net 혹은 fuzzy-logic routine, 혹은 procedure 에 의하여 점화함으로써 작동한다.

4. 실시간 다중프로세스생성 및 메세지전달기법

4.1 다중프로세스 생성 및 대체

UNIX 운영체제에서 사용자가 새로운 프로세스를 생성할 수 있는 유일한 방법은 fork()시스템 호출을 부르는 것이다. fork()를 부른 프로세스를 부모프로세스라하고, 새로이 생성된 프로세스를 자식프로세스라고 부른다. fork()시스템 호출구문은 다음과 같다. [3]

```
pid = fork();
```

fork()시스템 호출로 부터 복귀할 때, 두개의 프로세스는 복귀값인 pid를 제외하고는 사용자 수준의 문맥이 똑같은 상태이다. 한편 fork()된 자식프로세스는 자신의 영역을 다른 프로세스로 대체시키기 위해서 execl()시스템 호출을 수행한다.

4.2 다중프로세스간 메세지전달 및 동기화 설비

본 연구에서는 계층형신경망에서 각 계층간의 신경망이 입력

데이타와 결과를 특정의 처리기(OR 프로세스, Result agent)에 전달하기 위하여 메세지큐(message queue)를 사용하는데 메세지큐는 여러개의 프로세스들간의 통신을 위하여 제공되는 것이다. 메세지에는 세개의 시스템 호출을 통하여 이루어진다. [4] 메세지 id 를 얻기 위한 msgget()과 다른 프로세스에서 메세지를 받기 위한 msgrcv(), 다른 프로세스에 메세지를 주기 위한 msgsnd() 이다. 공유메모리(shared memory)는 특정의 관심있는 데이터를 이것을 필요로하는 여러 프로세스에게 동시에 제공하기 위하여 자신들의 가상주소공간의 일부분(공유메모리)을 공유하면서, 공유메모리에 저장된 데이타를 읽고, 또 공유메모리에 데이타를 기록함으로써 상호통신할 수 있다.[5] 본 연구에서는 시그날발생 터미날에서 전송된 이미지 데이타는 신호처리기(Siganal Processor)의 저장모듈과 Hopfield network에서 공유한다. 세마포아(semaphore)는 특정의 임계영역(공유자원, 화일) 속에서 한 순간에 존재하는 프로세스는 오직 하나로 제한하기 위하여 사용하는 것으로, 본 연구에서는 신호처리기와 Hopfield network사이에 공유메모리의 데이터 영역을 다른 프로세스가 조작하는 중에 또 다른 프로세스가 이 부분을 접근해 읽어오면 에러가 발생하는 것을 방지하기 위하여 어떤 하나의 프로세스가 공유메모리 영역을 조작중이면 다른 프로세스는 wait를 한후 시그날이 오기를 기다린다. 반면 조작을 완료한 프로세스는 시스날을 보내어 작업을 개시할 것을 명령한다.

5. 실시간 다중고장진단시스템 개발

일반적으로 시스템 내부에서 일어나는 현상은 다양한 수식모델과 다양한 정보패턴을 가지고 있으므로 이러한 정보를 빠르게 처리하기 위해서는 프로세스의 다중처리(multitasking)가 필요하고, 또한 각각의 다른 형태의 정보를 모듈별로 처리함으로써 진단시스템의 효율성 구조를 간략화 시킬 수 있지만 이들사이의 연관관계는 각 프로세스의 협조(cooperation)를 통하여 이루어져야만 한다. 하나의 요소의 고장진단중 다른 요소고장이 생기면 해당 부속고장진단모듈이 트리거 되도록되어 있고, 각각의 부속고장진단모듈은 실시간으로 자기의 맡은 부속시스템에 대한 고장진단을 실시간으로 수행한다.

5.1 실시간다중고장시스템의 적용범위

본 연구에서 고장진단 대상으로 삼은 시스템은 복합장치시스템으로 이 시스템을 구성하는 요소는 구체적으로 4개의 서브시스템, 7개의 아이템, 7개의 부속요소로 구성된다. 센서는 각 아이템의 고장특성을 대표하는 서브시스템 센서 4개와 부속요소의 고장을 대표하는 아니템센서가 7개로 가정한다. 각 센서는 그 하부요소에 대한 고장징후를 포함한다고 가정한다. 이 시스템은 특정기간 동안 쉬지 않고 작동되어야 하고 또한 예기치 않는 사고로 인하여 막대한 손실을 초래하여 생산성 저하를 가져올 뿐만이 아니라 수리에 어려움이 많다. 따라서 이와같은 시스템은 각 부분별로 나누어 진단을 수행해야 하고 고장종류도 단일고장(single fault), 다중고장(multiple fault)이 생길 수 있다.

5.2 실시간다중 고장진단시스템 구성

Fig. 4에 본 연구에서 사용된 실시간 다중고장진단시스템의 예를 표현하였다. 실시간고장진단시스템은 다음과 같이 여러개의 에이전트들로 구성되는데, 그리고 연구에서 에이전트는 주로 다른 에이전트와 독립적인 프로세스를 수행하고 특정의 통신설비를 통하여 다른 에이전트들과 통신하는 프로세스이다. 먼저 진단대상이 되는 시스템에서 서브시스템의 상태를 대표하는 기계의 각종 상황을 센서를 통하여 입력되면 에뮬레이터에서는 이러한 센서의 정보와 위치코드를 결합하여 호스트에 보낸다. 호스트에서는 각 센서의 정보를 입력받아 온라인 모니터링과 다른 기타 모듈들에 대한 특정의 입력조건에 맞도록 신호전처리를 실시하는 전처리 모듈들로 보내어진다. 이때 프로세스는 공유메모리를 이용하여 두개의 모듈에 동시에 센서정보를 보낸다. 보내진 정보는 이것을 참조하여 서브 D/B에 저장되어 있는 각종의 정상상태값과 비교하고, 정해진 기준치를 넘으면 고장부위(fault location)가 정해지는데 이 고장부위는 주로 서브시스템으로 분리된다. 서브시스템에 이상이 생기면 정밀진단에 들어간다. 메인진단모듈은 부속시스템의 정밀진단을 위하여 특정의 부속시스템 진단시뮬레이터는 fork()와 execl()함수를 이용하여 정밀진단에서는 먼저 서브모듈을 대표하는 센서의 입력을 근거로 현재의 고장이 어느 위치로부터 일어났는지를 계층형신경망(Hierarchical Artificial Neural Network)을 이용하여 진단한다. 하나의 계층형신경망은 주로 2단계로 나누어져 추론하는데, 초기단계는 단일고장을 분류해내는 역할을 하고 두번째 단계는 의심이 가는 고장부위를 테스트하여 확실한 고장진단을 수행한다. 하나의 상위레벨 계층형신경망은 부속시스템의 아이템(item)들의 이상여부를 가려내기 위하여 시뮬레이션되고, 또 다른 하위레벨 계층형신경망은 아이템내부의 요소(component)의 고장을 구별하기 위하여 시뮬레이션 되어진다. 계층신경망에서 고장요소가 확실히 정해지고 다른 성질을 시뮬레이션 하기 위해서는 하나 이상의 수식모듈을 시뮬레이션 할 수 있다. 진단된 결과는 메세지큐를 이용하여 결과처리 에이전트로 전달되고, 결과처리 에이전트는 각 고장에 맞는 수리, 분해, 조립, 셀업, 그래픽정보등을 출력한다. Fig. 5는 실시간 지적 다중고장 진단시스템의 제어구조를 나타낸 것이다. 여기서 블랙보오드에는 두개의 특정영역이 존재하는데 트리거와 결과처리에이전트로 이곳에서는 전달받은 정보를 이용하여 정해진 임무를 수행한다.

5.2.1 실시간고장진단시스템 모듈별 특성

1) 신호발생시뮬레이터 터미날 (Signal generator)

실제 대상시스템에서 각종 센서를 통하여 데이터를 입력받고, 이것의 상태를 진단하기 위하여 데이터를 실시간으로 호스트에 보내는 널모뎀 터미날시뮬레이터는 BC++ 4.5를 이용하여 구현되었고, 호스트에서는 터미날로부터 들어오는 데이터를 특정장소에 저장했다가 처리하는 신호처리기, 성능감시모듈, 경향체크 에이전트, 수명예측에이전트에 공유메모리를 통하여 제공한다. 터미날 에뮬레이터는 데이터 수집버튼을 누르면 난수데이타와 image processing 데이터(소수형데이타 32개 랜덤함수이용(sensor갯수:32개), 이미지 데이터(10x12 matrix)) 4개를 수집한 후 데이터 포맷에 맞추어 데이터를 상위 호스트로 전달할 수 있고, 또한 특정형식의 고장을 만들 수 있다.

2) 신호전처리모듈 (Signal preprocessing module(SPM))

에뮬레이터에서 호스트로 보내진 데이터를 받아서 특정의 버퍼에 저장하는데 이곳에서는 센서 데이터는 FFT 처리모듈에서는 특정의 수(64개 만큼 모아지면)데이터를 모은 다음 FFT 처리하여 그래픽 에이전트에게로 보내고, trend agent는 raw data를 받아서 화면에 표시한다. Performance Monitoring Module에 입력된 데이터는 특정한계치(limit value)를 계속 감시하다가 정해진 횟수를 넘어서면 트리거를 활성화시킨다. FFT 처리된 데이터는 7개의 특징패턴으로 나누어서 신경망에 입력된다. 이

때 FFT처리모듈과 trend agent, Hopfield network는 터미날로부터 들어오는 데이터를 공유메모리를 이용해서 서로 공유하므로써 프로세스의 속도를 증대시킨다.

3) 성능감시모듈(Performance monitoring module(PMM))

Performance monitoring module은 SPM에서 들어오는 4개의 데이터를 모니터링 한다. 이들은 미리 데이터가 들어오기 전부터 초기 learning data를 가지고 와서 학습을 완료해 놓은 상태이다. 그리고 입력데이타가 계속 들어오면 for loop를 이용하여 출력데이타를 출력하는데 D/B의 정상참조데이타와 어긋나는 경우에 trigger에 신호를 보낸다.

4) Trigger Module (TM)의 역할

Trigger는 PMM에서 신호가 들어오면 Subsystem Fault Daignosis Agent에 신호를 보내게 되는데 이때 문제가 되는 것은 한번 트리거된 신호가 계속 트리거될수 있으므로 (즉 고장이 시작되면 계속 고장이 전파되어 간다.) 이것을 처리할 필요가 있다. 이것을 해결하는 방법은 SFDA에 semaphore 설비를 장착하든지, 특정의 플래그(flag) table을 설치해 두고서 trigger가 작동되면 이곳에 1을 기록하고, 만약 모든 고장진단이 끝나면 다시 이것을 0으로 세팅하는 방법을 마련한다. 따라서 트리거는 일단 PMM에서 신호를 받고서 이상이 있다는 것을 감지하고서 플래그 테이블에 가서 현재의 플래그 상태를 조사하고 플래그가 세팅(1)이 되어 있으면 SFDA에 메세지를 보내지 않고 계속 공전한다.

5) 부속시스템 고장진단에이전트 (Subsystem Fault Daignosis Agent (SFDA))

Subsystem Fault Diagnosis Agent는 기존의 D/B module에서 데이터를 가져다가 시뮬레이터가 기동할수 있도록 D/B를 따로 설치하는데 이때 D/B 생성의 단위는 서브시스템을 구성하고 있는 Subsystem별로 설치된다. SFDA는 트리거 모듈에서 응답이 오면 D/B agent에게 특정의 토큰을 보낸다. 특정의 토큰을 보낸후 이상유니트에 관련된 데이터를 분리하여 가지고 와서 새로운 데이터베이스 (Sub D/B)를 생성하는데 이때 생성단위는 서브시스템에 속하는 모든 데이터를 가지고 온다. 그 다음 SFDA는 상부 계층형신경망을 이용하여 Sub D/B에서 학습데이타를 가지고 와서 학습후 추론을 시작한다. 이때 서브시스템(unit)가 4개 이면 동시에 4개의 프로세스(시뮬레이션모듈)를 형성하게 되는데 이중 어느 하나가 비정상이면 그 하부 조직의 아이템을 시뮬레이션한다.

6) 고장진단모듈 (Fault Diagnosis Module (FDM))

고장진단시뮬레이터는 계층형신경망으로 구성되어 있다. 본 연구에서 사용된 계층형신경망은 Fig. 6과 같은 구조를 가지고 있다. 각 신경망은 백프로파게이션 다층퍼셉트론을 이용하였고 각 셀 신경망은 입력층 노드는 7개이고, 은닉층은 30개 출력층 7개로 이루어져 있다. First level의 신경망은 단일고장과 다중고장 데이터를 학습해 두었다가 FFT모듈에서 들어오는 전처리된 7개의 특징패턴데이타를 입력으로 고장을 분리해 내는데 여기서 고장은 이 경우는 학습데이타수가 많으므로 주로 징후가 보이는 고장(즉 출력층이 0.5이상인 신경망)을 찾아내어 second level로 넘어간다. Second level에서는 7개의 신경망으로 이루어져 있는데 각각의 신경망은 고장1에서 고장7까지 이중고장 조합으로 학습되어 있다. First level에서 징후가 나타나는 7개의 고장요소에 대하여 정해진 second level 신경망을 선택하여 다중고장을 시뮬레이션한다. 여기서 나온 결과는 OR process에서 합쳐지는데 이때 first level과 second level 신경망은 각기 다른 프로세스이므로 이들 프로세스간에 통신이 필수적인데 이들간에 IPC설비인 메세지큐를 써서 선택된 신경망에 입력데이터를 입력한다. OR process는 first level 신경망의 결과에 따라 second level 신경망으로부터 시뮬레이션 결과를 받아들이는 정보를 받는다. 그리고 OR결과에 의하여 발생된 이상징후를 가진 item요소의 고장을 진단하기 위하여 third level 신경망을 선택하여 서브시스템 하부요소인 아이템요소의 센서정보를 입력받아서 고장을 시뮬레이션하고, 그 결과를 OR 한후 징후가 나타나는 부분에 대하여 하부부속요소 고장진단을 위하여 forth level 신경망을 활성화시켜 고장진단을 수행한다.

(1) IFDS (Item Fault Diagnosis Shell)의 역할

일반적으로 서브시스템은 Fig. 2 에서 보여진 것처럼 몇개의 모듈(item)로 분리되어 있는데 Item Fault Diagnosis Shell은 어떤 서브시스템에 이상이 있으면, 그 서브시스템의 해당 서브시스템의 동정을 시뮬레이션한다. 즉 Fig. 6에서 first, second level에 해당되는 것이다. 시스템 설계시 모듈별로 각각 고장이 절연되어 있고 모듈의 출력은 다른 출력에 영향을 받지 않는 것으로 한다. 여기서 어떤 특정의 서브시스템이 고장이 일어나면 그 하부 아이템중 고장징후가 나타나는 부분에 대하여 시뮬하게 되는데 다음은 item 동정시뮬레이터의 시뮬레이션 순서는 다음과 같다.

- 트리거된 unit simulator는 자기가 가진 모델에 맞는 초기 학습데이타를 가지고 학습을 완료한 후 유니트 센서데이터의 FFT 특징패턴을 입력으로 시뮬레이션을 시작한다. 시뮬레이션 결과 이상이 발생한 아이템 대하여 그 부속 아이템들의 다중고장을 진단하기 위하여 새로운 신경망을 선택적으로 형성한 후 아이템요소의 다중고장진단을 수행한 후 그 결과를 OR process로 보낸다. 각 계층의 신경망은 sub D/B에 입력된 정보(화일이름)를 입력하여 학습을 완료한다.

(2) CFDS (Component Fault Diagnosis Shell)의 역할

Component Fault Diagnosis Shell은 어떤 특정이상이 있는 아이템들에 대하여 그 하부요소인 부속요소(component) 이상을 진단하는 계층형신경망이다. 이것은 상부의 IFDS에서 발생된 결과를 OR프로세스가 종합하여 이상이 있는 부분에 대하여 이상 아이템 수 만큼 신경망을 활성화 시키는데 이때 신경망 입력데이타는 시그날 처리모듈에서 각 이상 아이템의 센서정보를 FFT 처리한 특징패턴을 입력으로 사용하여 이상징후가 있는 component 를 분리해낸 후 그 결과를 다시 상위 OR 프로세스로 메세지큐를 이용하여 보내게되는데 OR은 그 결과를 종합하여 기계요소(component)에 이상징후를 일으키는 물리적 변화를 진단할 수 있다.

7) Result Agent(RA)의 역할

Result Agent는 실시간 고장진단시스템의 OR결과와 trigger 를 입력받아 현재의 고장상황을 계층적으로 표시하는데 이때 계층적정보는 앞에서 언급된FTA정보에 근거하여 작성된 상황판

(result panel) 에 표시된다. 사용자는 현재의 기계상태를 이 상황판을 통하여 알 수 있고, RA로 넘겨진 결과는 상황판의 색을 반전시킨 후 조치사항 및 화상정보와 음성정보를 이용하여 출력한다. 여기서 조치사항 및 화상정보는 주로 기계의 테스트, 분해순서, 조립순서, 셀업, 각종 부속설치를 나타낸다. Fig. 7은 실시간고장진단시스템의 운용예를 나타낸 것이다. (a)는 PC측 signal generator의 화면을 나타낸 것이며, (b)는 host측(SUN)의 고장진단화면을 나타낸 것이다.

6. 결 론

기존의 고장진단기법은 정적인 전문가시스템에 의하여 이루어지고 있어서 수정시 전체시스템을 수리해야하고, 이를 위하여 시스템의 전체메카니즘을 알아야한다. 또한 시스템을 처음 설계한 사람만이 수정가능하고, 새로운 진단모듈 첨가시 기존의 진단모듈과 상호협조에 많은 문제점이 있다. 또한 기존의 시스템은 주로 순차적 처리에 의한 진단을 수행하지만 이렇게 될때 전파속도가 빠른 시스템은 진단이 진행되는 동안에 시스템 전체에 고장이 전파되어 커다란 손상을 초래한다. 따라서 고속, 복잡시스템은 다중고장을 진단할수 있어야 한다. 그리고 기존의 시스템은 프로그램을 한꺼번에 로딩시켜 사용하는 반면 본 연구에서 제시된 기법은 필요시 메모리 속으로 읽어들여 사용하고, 사용후 제거함으로써 메모리를 동적으로 사용할수 있다. 본 연구에서 주로 사용되는 진단제어모듈은 후향추론 전문가시스템셀모델, 각기 다른 신경망회로모델, 그리고 각 기계의 상태를 예측하기 위한 수식모델, 그리고 각 모델간의 정보교환과 상호협동을 위한 통신모델이 사용된다. 이러한 모델들은 각각 기계시스템에 하나의 부속시스템이나, 중요한 모듈, 혹은 복합요소(complex component)의 동정을 기술하기 위하여 각각 사용된다. FT에서 얻어진 구조적, 기능적 지식을 이용하여 고장원인을 추론한다. 각각의 모듈은 메인모듈의 필요에 따라 생성되거나 소멸되고, 또한 생성된 프로세스는 또한 필요한 각각의 모델을 생성하고, 소멸시킨다. 각각의 모듈간 제어는 메세지전달에 의하여 행하여 진다. 본 시스템에서 제시된 제어기법은 UNIX운영체제 IPC(Inter Process Communication) 설비를 이용하였다.

참고문헌

(1) Palajiappan, M., " The Envoy Framework : An Open Architecture for Agents ", ACM Transaction on Information Systems, Vol. 10, No.3, pp.233-264, 1992.

(2) Robert, E. and Toney, M., Blackboard System, Addision-Wesley, pp.10, 1988.

(3) Bach, M.J., " The design of the UNIX Operating System ", Prentice Hall, pp.105-237, 1986.

(4) Stevens, W.R., Advanced Programming in the UNIX Environment, Addison Wesely, pp.188-212, 1992.

(5) Stevens, W.R., " UNIX network programming ", Prentice Hall, pp.153-170, 1991.

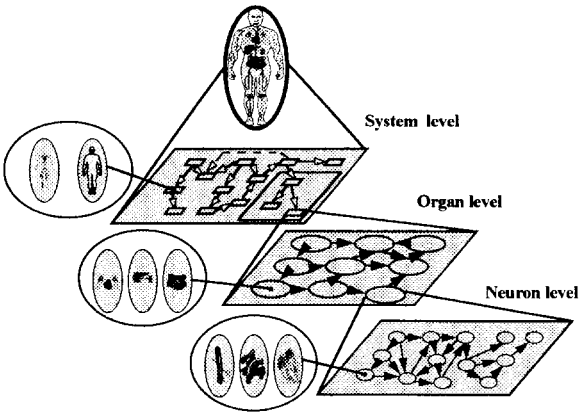


Fig. 1 Human body hierarchical structure and interaction

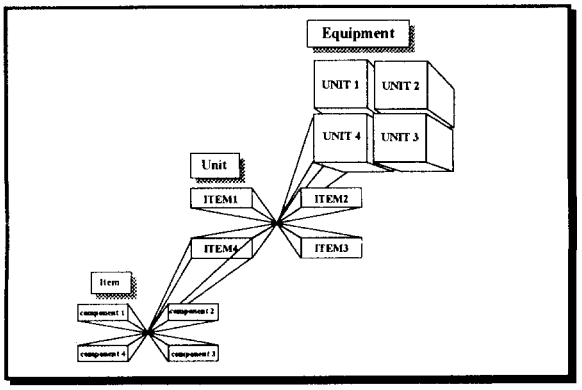


Fig. 2 Hierarchical decomposition of complex system

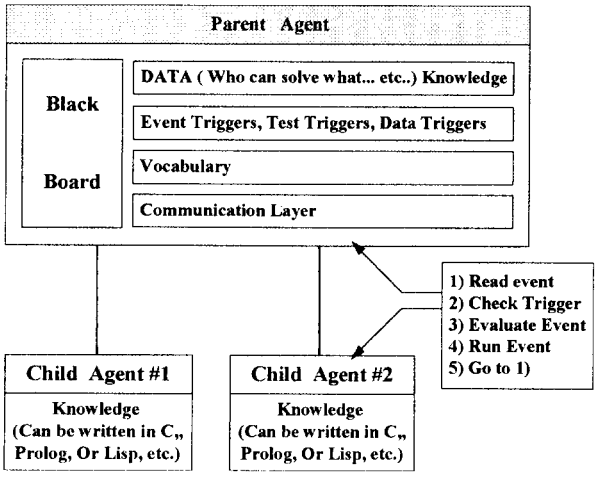


Fig. 3 The structure of agent [1]

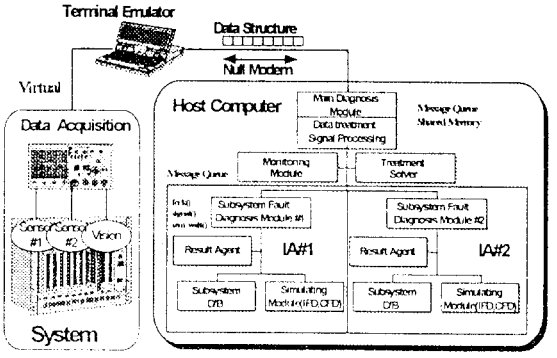


Fig. 4 Experimential of real time fault diagnosis system

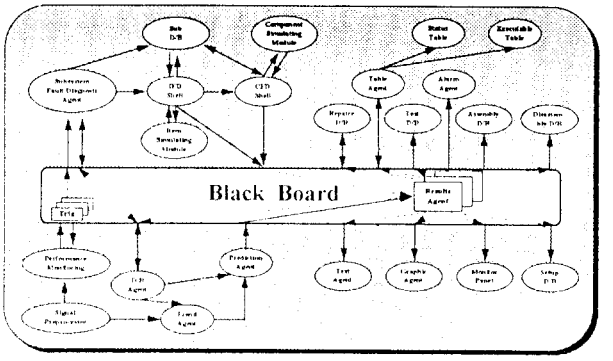


Fig. 5 Diagnosis control flow in RTMFDS

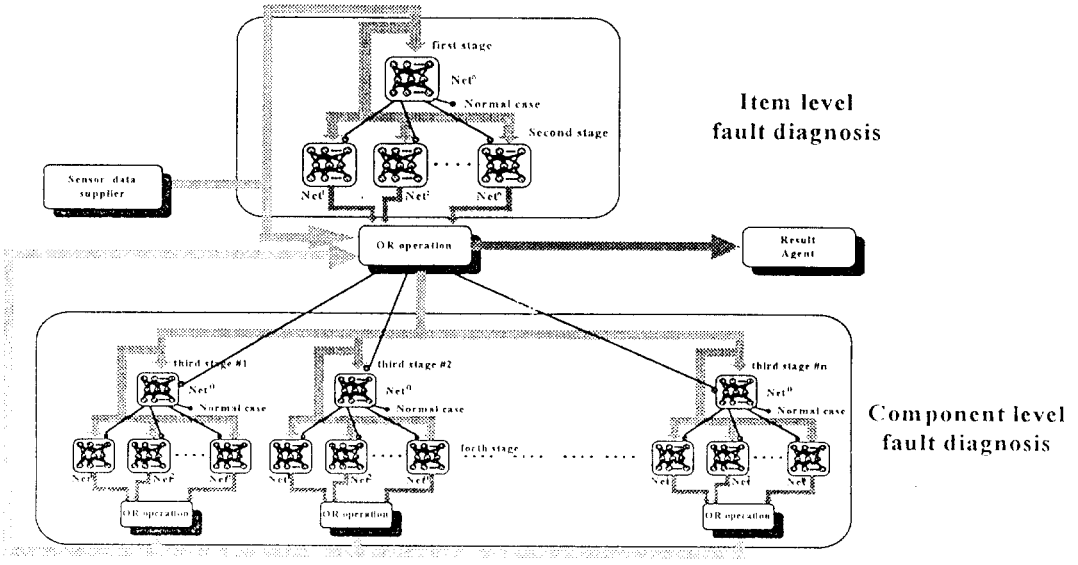
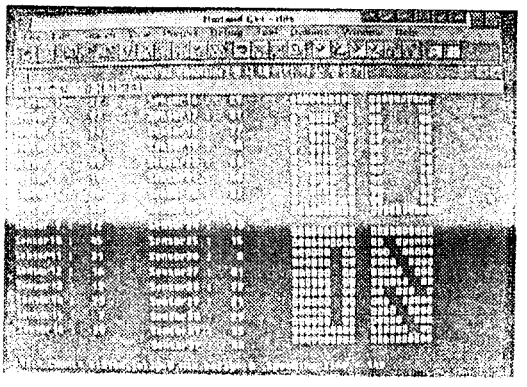
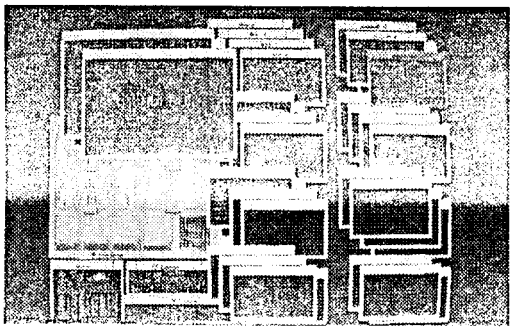


Fig. 6 Hierarchical Artificial Neural Network(HANN) in RTMFDS



(a) Picture of signal generating terminal in RTMFDS



(b) Picture of host simulation station in RTMFDS

Fig. 7 Picture of RTFDS