'98 추계학술발표회 논문집
한국원자력학회

객체지향언어를 이용한 모델기반 이상상태 검출시스템 구현

Implementation of Early Fault Detection System with COAST

이동영, 김창회, 송순자, 박원만, 이장수

한국원자력연구소
대전광역시 유성구 덕진동 150

요 약

Early Fault Detection(EFD) 시스템은 모델기반 이상상태 검출시스템으로, 진단하려는 대상 플랜트를 수학적 모델링 수식으로 표현한다. 모델의 결과는 에러가 없는 정상적인 프로세스의 상태를 나타내며, 플랜트에서 취득한 변수는 실제 플랜트의 상태를 나타낸다. 모델링 결과와 플랜트에서 관찰한 변수사이에 편차가 발생하면, 플랜트에 이상상태가 발생하였음을 알려주는 EFD 경보를 발생한다. EFD 시스템은 1980년대 후반 할덴 프로젝트(Halden Reactor Project)에서 FORTRAN 언어로서 구현되어 폭넓은 실증시험을 거쳤으며, 최근 IDS(Integrated Diagnosis System) 구성의 일환으로 객체지향언어인 COAST(COmputerized Alarm System Toolbox)를 이용해 다시 구현되었다. 본 연구는 할덴 국제공동연구 프로젝트에 참여하여 직접 수행한 내용 중 IDS 및 EFD 시스템의 개요, EFD 시스템의 구현결과 등에 대해 서술한다.

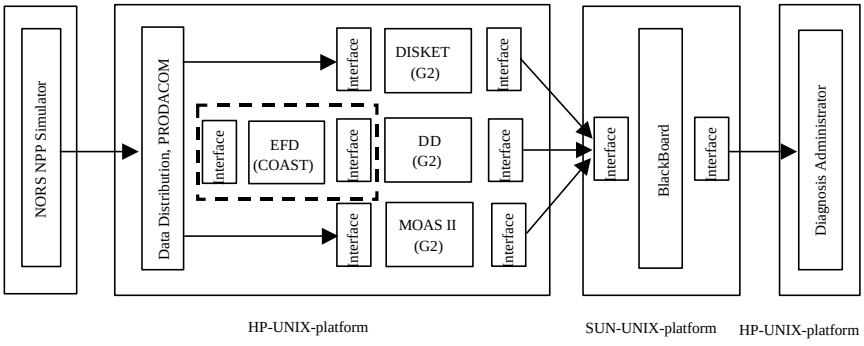
Abstract

EFD(Early Fault Detection) system is the model_based fault detection system, it divides target plant for diagnosis into several subprocess presented with mathematical model. Each subprocess receives process data from plant, and calculates error-free process variable from the mathematical model. Also EFD system retrieves real plant variable periodically, and compares it with the error-free process variable. When the deviation between the calculated error-free value and the real plant variable is detected, EFD alarm is generated. EFD system is implemented with FORTRAN in Halden Reactor Project in the late 1980s. Recently EFD system is reimplemented with COAST(COmputerized Alarm System Toolbox) to build IDS(Integrated Diagnosis System). In this paper, the concept of IDS and EFD, and the result of the implemented EFD with COAST is described.

1. 서 론

할덴 프로젝트는 경제협력개발기구(OECD: Organization for Economic Cooperation and Development) 산하 Nuclear Energy Agency의 국제 공동연구 프로그램으로, 노르웨이의 남쪽에 위치한 작은 도시 할덴에 설립되어 올해 창립 40주년을 맞고 있다. 현재 OECD에 가입한 20개 회원국으로 구성되어 있으며 한국은 1995년에 가입하였다. 할덴 프로젝트의 수행목적은 원자력발전소의 안전성과 신뢰성을 높이기 위함이며, 수행업무는 크게 핵연료 및 MMI(Man-Machine Interface)관련 연구로 나눌 수 있다. 핵연료 연구는 Halden Reactor를 이용해 재료 및 부식시험을 수행하고 있다. 또한 MMI관련 연구는 HAMMLAB[1](Halden Man-Machine Laboratory)을 이용하여, 최신의 컴퓨터기반 기술을 원전에 적용하기 위한 연구 및 이러한 기술이 원전에 적용되었을 때 운전원에게 미치는 영향을 분석하는 연구들을 폭넓게 수행하고 있다.

그 동안 할덴 프로젝트에서는 컴퓨터기술을 적용하여 플랜트 이상상태의 원인을 파악하기 위한 많은 연구를 수행하였으며, 그 결과 DISKET[2](DIagnosis System using Knowledge Engineering Technique), EFD/DD[3](Early Fault Detection/Detailed Diagnosis), 및 MOAS II[4] 등과 같은 여러 이상상태 진단시스템의 프로토타입을 구현하였다. DISKET는 룰기반 진단시스템이며, EFD/DD 및 MOAS II는 모델기반 진단시스템이다. 그러나 이들 각각의 진단시스템이 발전소의 운전모드, 플랜트 및 기기의 상태를 망라하여, 이상상태의 모든 원인을 진단할 수 있도록 완전한 지식기반 및 추론엔진을 구비할 수는 없다. 그러므로 할덴 프로젝트에서는 개별 진단시스템의 이상상태 진단능력을 높이기 위해 지식기반을 공유하여 통합된 상위레벨 진단결과를 만들어내는 IDS[5](Integrated Diagnosis System)를 개발하고 있다. 이상상태 발생 시 IDS의 각 진단시스템은 추론결과를 같은 모양의 지식기반으로 'Black Board[6]'에 저장하고, 'Diagnosis Administrator' 모듈에서 이들 각 진단시스템의 결과를 분석하여 통합된 진단결과를 발생한다. EFD는 객체지향언어인 COAST[7](COmputerized Alarm System Toolbox)로서 구현하였으며, 진단시스템인 DISKET, EFD/DD 및 MOAS II는 실시간 전문가시스템인 G2 expert shell[8]을 사용하여 구현하였다. 그리고 IDS는 HAMMLAB 시뮬레이터로부터 변수 값을 실시간으로 받아 처리하며, 구성은 그림 1과 같다.

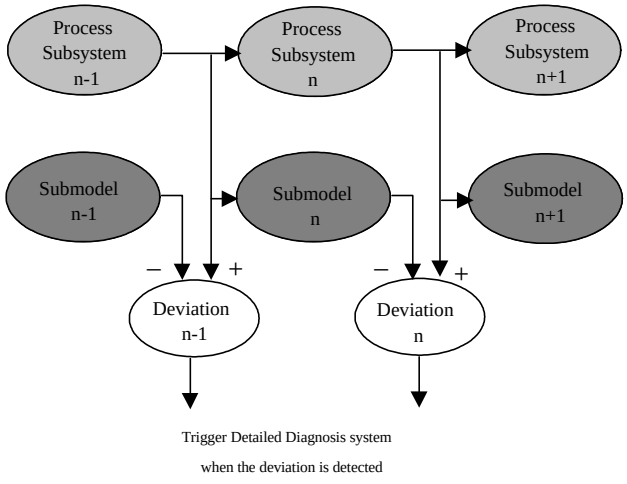


<그림 1> IDS 구성도

EFD/DD(Early Fault Detection/Detailed Diagnosis) 시스템은 IDS 시스템의 일부분으로, 원전에서 이상상태 발생 시 이를 조기에 검출하는 EFD 시스템과, EFD에서 주어지는 에러패턴을 바탕으로 이상상태의 원인을 진단하는 DD 시스템으로 구성되어 있다. EFD 시스템은 1980년대 말에 FORTRAN 언어의 일종인 EFFORT로 구현되어 여러 발전소에서 폭넓은 실증시험을 거친 진단시스템이다. 그러나 EFD를 IDS의 일부로 통합하기 위해 1998년 객체지향 언어인 COAST로서 EFD를 재 구현하였다.

2. EFD 개념

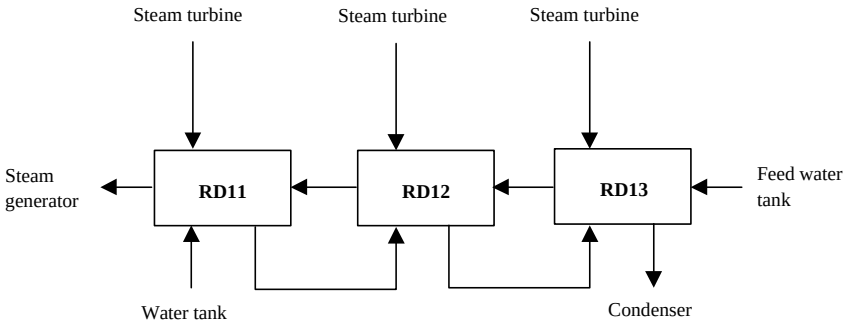
EFD는 모델기반 이상상태 진단시스템으로, 진단하려는 대상 플랜트를 여러 개의 서브프로세스로 구분하고 각 서브프로세스를 수학적 모델링 수식으로 표현한다. 수학적 모델로 표현된 각 서브모델은 실제 발전소에서 취득한 변수 값을 이용하여 감시하고자하는 프로세스의 정상운전상태 변수 값을 계산한다. 수학적 모델에서 계산된 정상상태 변수 값은 플랜트에서 취득한 실제 프로세스의 해당 변수 값과 주기적으로 비교된다. 정상상태 운전 시 두 값은 일정한 에러범위 내에서 존재하나, 플랜트에서 이상상태가 발생하면 두 값 사이의 편차가 증가하며, 이때 EFD 시스템은 대상 플랜트에 이상상태가 발생하였음을 나타내는 EFD 경보를 발생한다. 또한 이상상태시 EFD 시스템은 각 서브프로세스 단위로 에러패턴을 발생하여 DD(Detailed Diagnosis) 시스템으로 전송한다. DD 시스템은 EFD 경보에 의해 기동하며, DD의 추론엔진에서 EFD에서 전송된 에러패턴을 이용해서 플랜트 이상상태의 원인을 파악한다. 그림 2는 EFD 시스템의 개념을 나타낸다.



<그림 2> EFD 개념도

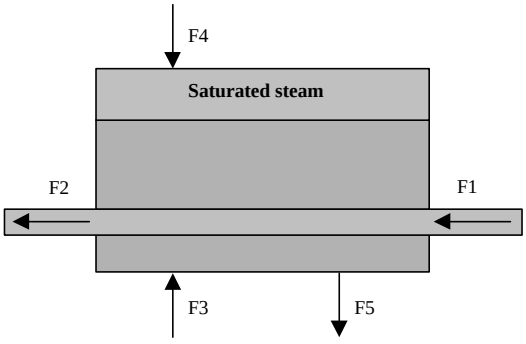
3. 구 현

IDS에 포함된 여러 이상상태 진단시스템의 방법을 실증하기 위해 HPP(High Pressure Preheater) 시스템이 선정되었다. HPP는 2차계통에서 증기발생기로 귀환하는 주급수를 가열하므로 원전의 효율을 증대시키기 위해 마련된 시스템으로, 직렬로 연결된 3개의 열교환기로 구성되어 있다. 주급수는 주급수 탱크에서 공급되어 첫 번째 열교환기로 들어간다. 주급수를 가열하기 위한 열원(heat source)은 터빈에서 추출된 포화증기를 사용하며, 열교환 후 응축된 뜨거운 물은 다음 단의 열교환기 탱크나 응축기(condenser)로 전달된다. 그림 3은 HPP 시스템의 배열 및 각 열교환기 사이의 상관관계를 나타낸다.



<그림 3> HPP 시스템 구성도

EFD 시스템은 해당 프로세스를 여러 개의 주요한 서브프로세스로 구분하고, 각 서브프로세스를 수학 모델로 표시한 후 이를 객체지향 언어인 COAST를 사용하여 구현하였다. 본 논문에서는 해당 HPP 시스템을 4개의 밸브궤도 및 3개의 유량변수를 계산할 수 있는 7개의 서브프로세스로 나누었다. HPP 열교환기는 그림 4와 같이 5개의 포트를 갖는 모델로 단순화 할 수 있다. 주급수는 튜브를 통해 포트1로부터 포트2로 흘러간다. 튜브측(tube side)의 주급수를 가열하기 위한 열원으로 터빈에서 추출된 포화증기(포트4)를 사용하며, 주급수를 가열한 후 응축된 뜨거운 물은 다음 단의 열교환기로 들어가고(포트5), 이전 단의 열교환기



<그림 4> HPP 모델

에서 발생하는 응축수는 포트3을 통해 탱크로 들어온다. 이들 변수 중 탱크 쉘측(shell side) 유량, *F5*는 열교환기의 건전성을 확인할 수 있는 중요한 변수이다. 정상상태의 유량 *F5*는 HPP 모델 및 에너지질량 보존방정식(energy and mass balance equation)에 의해 계산되고, 실제 플랜트에서 측정된 프로세스변수와 비교하여 플랜트 이상상태 발생을 검출한다.

HPP 모델은 객체지향언어인 COAST로서 속성(attribute)과 메소드(method)를 갖는 클래스(class)로 정의하고, 정의된 클래스의 객체(object)로서 각각의 특정한 열교환기를 나타낸다. 정의된 클래스는 같은 종류의 모든 열교환기를 나타내기 위해 사용할 수 있다. HPP 모델의 입력을 받아들이고, 모델로부터 정상상태 변수 값을 계산하여, 실제 플랜트의 변수 값과 비교한 후, 이상상태 발견시 EFD 경보를 발생하는 EFD 열교환기의 간단한 클래스 정의는 다음과 같이 표시된다.

classdef EFD_HeatX_5Ports

```
-- Models a heat exchanger with 5 ports, as the High Pressure Preheaters in NORS. A tank with saturated
-- water and steam, and a level measurement constitutes the shell side. It has one inflow of water (port3),
-- one of saturated steam (port4), and one outflow of water (port5). The tube side is being warmed up,
-- water flowing in through port1 and out of port2. Whenever any of the input variables are changed, the
-- attributes F5_Fault_Free, Deviation and AlarmStatus are calculated.

-- Input variables:
   attr F1:real                              -- Flow into port1
   attr T1:real                              -- Temperature at port1
   attr T2:real                              -- Temperature at port2
   attr F3:real                              -- Flow into port3
   attr T3:real                              -- Temperature at port3
   attr P4:real                              -- Pressure at port4
   attr T5:real                              -- Temperature at port5
   attr F5_Real:real                         -- Measured flow from real process
   attr Wfract:real := 0.0                   -- Water fraction in steam and water
   attr Area:real := 2.9                     -- Cross-sectional area
   attr Vv_Pos:real                          -- RL30S004 valve position
   attr Level:real with history of 1 updates -- Level in tank
   attr Levinc:real = Level -Level as in 1 updates ago -- Increase of tank level
   attr SampleTime:real = timestamp of Level -timestamp of Level as in 1 updates ago

-- Local variables: Check the status of bypass valve RL30S004
   attr Valve_Status:string = if (Vv_Pos < 0.1)
                              then "Operating" else "Bypassed"

-- Output variable:
   attr F5_Fault_Free:real = F3 + F4 + (-Area) * Levinc / SampleTime
                                        -- F4 is calculated from energy equation

-- Determine EFD alarm status:
   attr Compensation_Value:real          -- Tuning coefficient of model
   attr Tolerance:real := 4              -- EFD alarm limit
   attr Deviation:real = F5_Real - (F5_Fault_Free + Compensation_Value)
   attr AlarmStatus:atring = if((abs(Deviation) > Tolerance) and (Valve_Status == "Operating"))
```

```
        then "EFD_Alarm"
    else if((abs(Deviation) < Tolerance) and (Valve_Status == "Operating"))
        then "OffAlarm"
    else if(Valve_Status == "Bypassed")
        then "Bypassed"
```

endclassdef

객체지향 프로그래밍에서는 정의된 클래스를 이용하여, 필요한 각각의 객체를 정의할 수 있다. RD11 열교환기의 객체는 다음과 같이 표시되며, 입력변수의 값이 바뀔 때마다 RD11 열교환기의 상태를 계산한다.

```
objectdef HX_RD11 : EFD_HeatX_5Ports
    (F1 = RL30F001.Value)
    (T1 = RL30T003.Value)
    (T2 = RL30T004.Value)
    (F3 = ValveRN13S001.Real_Flow)
    (T3 = RN13T001.Value)
    (P4 = RD11P003.Value)
    (T5 = RN21T001.Value)
    (Vv_Pos = RL30S004.Value)
    (Level = RD11L001.Value)
    -- HX_RD11.F5_Fault_Free is calculated whenever one of the input is changed.
```

4. 결 론

1997년 여름부터 약 1년 동안 할덴 프로젝트에 참여하여 국제공동연구를 수행하였다. 많은 연구 항목 중 COAST를 이용한 EFD 시스템의 구현 및 네델란드 KEMA사의 경보시스템 프로토타입 구현 업무에 참여하였다. 원래 COAST는 경보시스템개발 도구로 만들어졌으나, 본 연구를 통해 진단시스템 개발 업무에 COAST의 적용 가능성을 타진해 보았다. 개발된 EFD 시스템은 DD 시스템 및 HAMMLAB과 on-line으로 연결하여 HPP의 이상상태 진단시험을 수행하였다. 개발된 EFD 시스템은 향후 구현될 IDS의 모듈로서 HAMMLAB 내에 존재할 것이며, 또한 EFD 시스템 개발 결과는 액체금속로 KALIMER의 supervisory 제어시스템 구현에 활용될 것이다.

* *본 연구는 과학기술부의 원자력연구개발 사업의 일환으로 수행되었습니다.* *

참고문헌

1. Halden Man-Machine Laboratory as of 1996, HWR-476, May 1996, OECD HRP.
2. Documentation of the Diagnosis System DISKET, HWR-188, Feb. 1987, OECD HRP.

3. Detailed Diagnosis based on Early Fault Detection, HWR-267, Jan. 1990, OECD HRP.
4. A New Surveillance and Diagnosis System for NORS Based on the MOAS II Methodology, Oct. 1994, OECD HRP.
5. Representation and Use of Knowledge in Automatic Fault Diagnosis, HWR-448, Apr. 1996, OECD HRP.
6. Harmon, Leroy, Spaggiari, Bremdal, Grini, Nilsen, "Tool for Capitalizing and Utilizing Knowledge", OECD HRP EHPG Meeting, Leon, Norway, 1996.
7. COAST Alarm System Toolbox, Reference Manual version2.2, OECD HRP.
8. Gensym Corporation, "G2 Reference Manual".