'90 韓國自動制御學術會議論文集 1990. 10.26~27

제어 시스템의 신뢰도 향상을 위한 이중화 구조 연구

박 세 화　문 봉 채　김 병 국　변 증 남
한국과학기술원　전기 및 전자공학과

A Study on the Control System with Dual Structure to Enhance Its Reliability

Sehwa Park　Bong Chae Moon　Byung Kook Kim　Zeungnam Bien
Dept. of Electrical Engineering
Korea Advanced Institute of Science and Technologe

ABSTRACT

In this paper, a reliable control system structured with dual CPU modules and dual I/O modules is implemented as a means of achieving a highly reliable fault tolerant control system. For this, faults in the system modules are first examined, and a fault detection technique consisting of self diagnostic, comparison process, and exception processing is applied. Also reliability analysis is conducted for the discrete time Markov model with dual structure. It is shown quantitatively that the reliability is improved in the the control system with dual structure in comparison with a system with single module structure.

1. 서 론

제어시스템은 대상 플랜트를 적절히 제어하여 외란에도 불구하고, 원하는 출력을 얻도록 하는 데에 그 목적이 있다. 현대 산업사회가 복잡화되고 규모가 커짐에 따라, 대규모 제어시스템의 높은 신뢰도가 요구되고 있는 바, 지속적이고 안정되게 플랜트를 제어하기 위해 제어시스템 자체의 신뢰성을 높이려는 노력이 있어왔다[1]-[3].

발전소 역시 제어 대상이 많은 대규모 플랜트의 일종으로써, 만일 발전소 제어시스템의 고장사고로 발전소가 가동 중지하게 되면 경제적, 사회적인 큰 손실을 가져오게 되므로 제어시스템의 신뢰도 문제가 중대한 문제로 대두되고 있다. 제어시스템의 낮은 신뢰도로 인해, 고장사고가 생기기도 하는 데, 예로 최근의 국내 원자력 발전소의 가동 중지 46건중 7건이 제어시스템과 관련된 오동작에 의한 것으로 보고 되고 있다[4].

제어시스템에서의 오동작을 막는 방법으로, 고품질의 소자를 사용하여, 소자의 고장 자체를 되도록 줄게 하는 고장회피(fault avoidance) 방식과 소자에서 고장이 발생하더라도 시스템의 성능에는 영향을 주지 않도록 하는 내고장성제어(fault tolerant control) 방식이 있다[5]. 그런데, 고장회피 방식만으로는 운용 환경의 변화, 경년 변화 등으로 인한 고장에 대처할 수 없기 때문에 좀 복잡하지만 적극적인 방법인 내고장성 제어방식을 취할 필요가 있다. 내고장성 제어방식은 같은 기능을 하는 모듈을 여러 개 두는 중복구조(redundancy)를 갖도록 하는 것인데, 우주선, 항공기의 제어나 원자력 발전소 같은 대규모 공정 제어 분야[5] 등 높은 신뢰성을 요구하는 분야에 주로 이용되어 왔다.

Bailey 회사는 CPU 모듈을 이중화한 형태의 발전소 제어시스템을 선보이고 있으며[6], 국내의 연구로는 아날로그 모듈로 이루어진 고장이 잦은 한 발전소 제어기의 신뢰성을 높이기 위해 디지탈 백업 제어기를 구성한 경우도 있는 데[2], 이는 기존의 제어시스템에 추가로 덧붙여졌기 때문에, 하이브리드 형태의 중복구조를 취하고 있다. 높은 신뢰성을 요구하는 제어시스템은 중복구조를 가질 필요가 있으나, 지나치게 많은 중복구조를 갖도록 제어시스템을 다중화하면, 너무 많은 추가비용이 들어갈 수 있으며, 이중구조만으로도 충분히 경제적으로 시스템의 신뢰도를 높일 수 있다고 본다. 그러나, CPU 모듈만 이중화했을 경우에 I/O 모듈이 고장나면, 이중화된 CPU 모듈은 큰 의미가 없게 된다.

본 논문에서는, 동일한 형태와 기능을 갖는 CPU 모듈과 I/O 모듈 모두가 이중화된 제어시스템을 구현하였다. 그래서, 제어시스템 내의 어느 모듈 내의 고장에 대해서도 보다 적극적으로 대처하려 한다. 그리고, 구현된 이중구조를 갖는 제어시스템과 단일구조를 갖는 경우와의 신뢰도를 정량적으로 비교 분석하는 작업도 수행하였다.

2. 이중구조를 갖는 제어시스템

플랜트를 제어하기 위한 디지탈 제어기를 하드웨어적으로 구성함에 있어서 CPU 모듈, I/O 모듈, 신호 조절(signal conditioning) 기능이 있는 모듈등이 필요하다. I/O 모듈은 신호 변환(D/A 변환, A/D 변환 등) 등이 포함된 모듈을 말하며, 신호 조절 기능은 플랜트로 보내는 제어 신호를 전압 혹은 전류로 바꾸거나, 적절한 level로 신호를 바꾸는 기능을 말한다. 그림 1은 이중구조를 갖도록 구성한 시스템을 나타내고 있다. 점선 부분은 이중화시 추가로 포함된 부분이다. MLC(multi loop controller)는 제어 연산 및 정보 처리를 수행하는 CPU 모듈을 말하며, SIB(signal interface board)는 I/O 모듈로써 데이타 입출력을 위한 A/D 변환 또는 D/A 변환을 맡고 있으며, MLC와 SIB가 이중화되었다. STB(signal termination board)는 SIB와 플랜트 사이에서 신호 조절 기능을 담당함은 물론 이중구조를 갖는 SIB 출력 신호를 선택하기 위한 스윗칭 기능을 포함하고 있다.

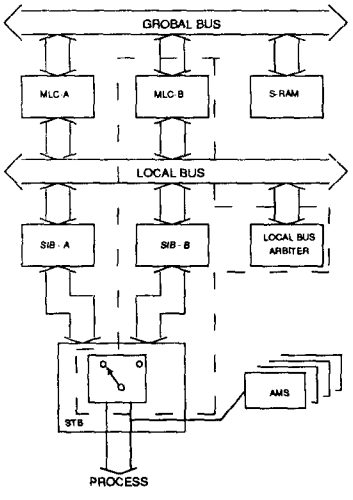

그림 1. 전체적인 하드웨어 구성도

Fig. 1. The Hardware Structure

MLC-A는 주제어기 모듈이고, MLC-B는 MLC-A가 고장시 백업 제어를 하게 되는 부제어기 모듈을 의미한다. 마찬가지로 SIB-A가 우선하는 모듈이고, SIB-B는 SIB-A의 고장 시에 그 기능을 대신하는 모듈이다. 그리고, S-RAM은 두 MLC 사이에서 글로발(global) 버스를 통한 통신과 데이타 저장 등을 위한 것이다.

MLC는 로컬(local) 버스를 통해 SIB와 데이타를 주고 받는데, 로컬 버스 아비터를 구현함으로써, MLC-A와 SIB-B 또는 MLC-B와 SIB-A 등 4가지 조합의 어느 구성도 가능하여 크로스 백업(cross back-up)을 할 수 있다. MLC는 모듈 내부 혹은 외부의 어떤 디바이스(예로 메모리)로부터든지 데이타를 입출력 받을 때 스트로브를 발생시키고 이에 대한 응답으로 데이타 전송 인식 신호(DTACK, data transfer acknowledge)를 받는 데, 로컬 버스를 통한 데이타 교환 시에는 입출력 버퍼를 통하여 데이타를 전송한다. 그러므로, SIB로부터 데이타를 입출력하려 할 때에도 스트로브를 보내고 DTACK을 받는 데, 로컬 버스 아비터는 두 MLC가 보내는 두 스트로브 신호를 이용하여 상태 머쉰(state machine)으로 구현하여 순차적으로 각 MLC의 입출력 버퍼의 여닫음을 제어하는 신호를 만들어 주는 역할을 한다.

STB에 SIB의 스윗칭을 위한 펄스 탐지 회로가 첨가되어 있는 데, 0 혹은 1의 레벨(level)로써 SIB의 스윗칭 신호를 보내게 되면, 고장으로 항상 0(stuck at zero fault)이거나 1일때 원하지 않는 스윗칭이 일어날 수 있다. 그래서 스윗칭 신호를 펄스 방식으로 함으로써 고장시에 잘못된 신호 레벨에 의한 스윗칭 가능성을 없앨 수 있다.

AMS(auto/manual station)는 수동 운전과 자동 운전의 절환을 용이하게 하기 위함으로 마련되었다. 만일 두 MLC나 두 SIB 모두 고장이 나서 더 이상 플랜트의 제어가 불가능 할 때는 제어기의 출력을 차단하고 수동 운전 상태로 절환을 시켜 사람이 직접 제어 출력값을 조작할 수 있도록 하는 것이 잘못된 제어 출력으로 인해 플랜트에 미치는 악영향을 막을 수 있어서 시스템의 안전성을 높이게 된다. 정상시에 MLC는 매 샘플링 시간마다 펄스를 보내면, AMS에서 주기적인 펄스를 탐지하게 되는 데, 만일 두 MLC 또는 두 SIB 모두의 고장으로 이 펄스가 오지 않으면 AMS는 스스로 수동절환을 하게 된다.

3. 제어시스템 내의 고장과 이중구조

앞절에서 언급한 바와 같이 MLC와 SIB를 이중화하였으며 이 두가지 모듈을 고장 진단의 대상으로 한다. MLC의 경우 고장 부위 결정 단위는 기능별 구성 요소로 설정하였으며, MC68000 CPU, Timer, 버스, 메모리 등으로 구분되고, SIB의 경우는 A/D 변환 회로, D/A 변환 회로, on-off의 디지탈 신호의 입출력 회로, MLC와 인터페이스(interface)하는 회로 등으로 구분된다.

MLC 내의 고장의 탐지는 크게 모듈 자체 테스트에 의한 방법과 다른 MLC 와의 계산된 데이타의 비교 과정, 그리고, 보완적인 방법으로 자체 진단과 비교 과정 만으로 모든 MLC 내의 고장을 찾지 못하는 경우를 위해, MC68000 CPU가 처리해 주는 예외처리(exception processing) 에러에 의해 고장을 탐지하게 된다.

모듈 자체 테스트는 각 소자별로 그 기능을 제대로 수행하는 가를 확인하는 방법으로 MC68000 CPU, Timer, ROM, RAM, 데이타 버스, 어드레스 버스 등을 MLC가 플랜트를 제어하고 있지 않는 동안에 계속 반복 테스트하는 것이다. 예를 들어 RAM의 경우 쓴 데이타와 읽은 데이타의 비교시에 다를 경우 RAM이 고장이라고 보는 것인데, [3]에서 사용한 방법을 보완 적용하였다. 아날로그 제어기의 디지탈 백업 제어기[2]에서의 고장탐지 방법은 주로 모듈 자체 테스트에 의해서만 했는 데, 여기서는, MLC의 고장탐지를 모듈 자체 테스트뿐만 아니라 비교 과정, 예외처리 과정을 둠으로써 보다 확장된 방법으로 고장에 대응한다. 비교 과정에 의한 고장 탐지는 계산된 출력값이 서로 차이가 날 때 한쪽을 고장이라고 판단하는 것이다. 일반적으로 그 비교값이 다를 경우 어느 모듈이 고장인지 그 결과만 보고 결정할 수 없기 때문에 다음과 같은 규칙에 의해 결정하였다.

$$| \, | y(hk)_{MLC-A} | - | y(hk)_{MLC-B} | \, | > \epsilon, \quad k=0,1,2...$$

일때

$$\max\left(\left| \frac{y(hk)_{MLC-A} - y(hk-h)}{y(hk-h) - y(hk-2h)} \right| , \left| \frac{y(hk)_{MLC-B} - y(hk-h)}{y(hk-h) - y(hk-2h)} \right| \right)$$

을 계산한 모듈을 고장난 MLC 모듈로 본다. 여기서, h는 샘플링 시간으로, hk는 총 동작 시간을 의미하며, y는 계산된 출력값이다. 즉, $y(hk)_{MLC-A}$는 hk시간에 MLC-A에서 계산된 출력값이다. 또한, ϵ은 이상 한계치(Threshold)를 의미하며 시스템에 맞게 정해준다. 위의 방법은 일반적으로 제어기의 경우 고장이 생기면, 그 출력 신호 값이 급격히 변한다는 사실을 고려한 것이다. 예외처리에 의한 고장 탐지는 MC68000 CPU가 처리해주는 고유한 기능을 이용한 것으로, 하드웨어나 소프트웨어에 이상이 생겼을 때, 정해진 예외 벡터(exception vector)에서 수행할 루틴의 시작 어드레스를 가져와 그 루틴을 수행하는 것이다. 예를 들어 하드웨어에 고장이 생겨 DTACK 신호를 발생시키지 못하면 예외처리 버스에러가 발생한다. 예외처리 에러시 즉시 예외처리 루틴을 수행하게 되며, 예외처리 에러 루틴에 고장정보를 담은 플래그를 세우게 함으로써 어떤 종류의 고장이 발생하였는지를 구분할 수 있다.

SIB에서 발생되는 고장은 MLC의 진단에 의해 탐지되며, 크게 다음의 3가지 방법을 적용하였다.

SIB에서 A/D 변환 시에 인터럽트(level 4)방식을 사용하여 인터럽트가 오면 변환된 데이타를 읽어 갈 수 있도록 하였기 때문에, 변환 시작후 정해진 시간 내에 인터럽트 신호가 오는 가를 확인하는 방법을 적용하였다. 즉, 일정한 A/D 변환 시간 내에 인터럽트 신호가 오지 않을 경우, A/D 변환 회로의 고장 임을 알 수 있다. 또, 범프 없는 전환(bumpless transfer)를 위해 MLC가 SIB를 통해 STB로 내보내는 신호를 다시 SIB를 거쳐 MLC가 되받게 되어 있는 데, 이 때, D/A 변환과 A/D 변환 과정을 거친다. 여기서 내보낸 값과 되받은 값이 다를 시에 이 부분 전체의 고장이라고 본다. 비슷한 방법이 디지탈 온오프(on-off) 신호에서도 적용된다. MLC와의 인터페이스부분은 MLC가 SIB에 데이타를 읽거나 쓸 때, DTACK 신호를 MLC로 보내주어야 한다. 만일, 이 신호가 오지 않으면 MLC에서 버스 에러가 발생하므로 이 부분의 고장임을 알 수 있다. 그러나, 버스 에러는 MLC 내부의 원인에 의해 생길 수도 있으므로, 이의 구별을 필요로 한다. 그 구별은 SIB 모듈을 읽거나 쓰기 전에 플래그를 세웠다가 후에 다시 플래그를 지우는 방법을 사용하여 만일 버스 에러가 생겼을 때 이 플래그가 세워져 있으면 SIB모듈에 의한 것이고 그렇지 않으면 MLC 내부의 원인에 의한 것으로 구별을 할 수 있다. MLC 모듈 내부의 원인에 의한 예외처리 에러인 경우에 MLC 백업등의 조처가 뒤따르고, MLC 모듈 외부의 원인 즉 SIB에 의한 경우이면 SIB 백업을 실행한다.

그리고, MLC에서의 고장 부위의 결정은 세부적인 모듈 자체 테스트를 통하여 보완한다.

이중구조 알고리즘을 실행함에 있어서 두 MLC 사이의 동작 상태를 나타내는 상태 플래그를 다음과 같이 설정하였다.

i) 각 MLC가 동작하고 있음을 나타내 주는 것
ii) SIB로부터 데이타를 입출력시에 고장이 있었는지를 나타내는 것
iii) SIB 고장시에 진단을 했음을 나타내 주는 것
iv) MLC-A가 제어 출력 값을 계산 했음을 나타내 주는 것
v) MLC-B가 비교 과정을 끝냈음을 나타내 주는 것
vi) 비교 과정의 결과를 나타내 주는 것
vii) MLC-A나 MLC-B의 동작을 강제로 멈추게 하는 것

주기적으로 반복되는 샘플링 시간 동안의 수행되는 일의 타이밍 다이아그램(timing diagram)은 그림 2와 같다.

첫번째 단계로 상태 플래그 ii), iii), iv), v), vi)를 초기화한다. 두번째 단계로 다른 MLC에서 모듈 자체 테스트 중에 고장이 있었는가를 플래그를 통해 점검한다. MLC-A의 고장시에는 MLC-B가 백업 제어를 하고, MLC-B의 고장시에는 플래그 vii)을 세워서 MLC-B의 동작을 멈추게 한다. 세번째 단계로 SIB를 통해 데이타를 입력받고, SIB 내에 고장이 탐지되면 SIB 백업을 한 후 ii), iii) 플래그를 세운다. SIB의 백업은 SIB-B를 통해 STB로 펄스를 보내어

SIB를 스윗칭하는 것이다. 네번째 단계에서 각 MLC는 제어 출력 값을 계산하고, MLC-A가 iv) 플래그를 세우면, MLC-B가 비교한 후 v), vi) 플래그를 세운다. 이 때 비교값에 차이가 있으면, 변화 폭이 큰 MLC를 고장으로 판단한다. 이 비교 과정에서 MLC-A가 고장일 경우 MLC-B가 백업 제어를 수행하며, 이 때 vii) 플래그를 세운다. 다지막 단계는 자체 테스트를 수행하는 과정으로 고장이 탐지되면 고장 정보를 담은 플래그를 세운다.

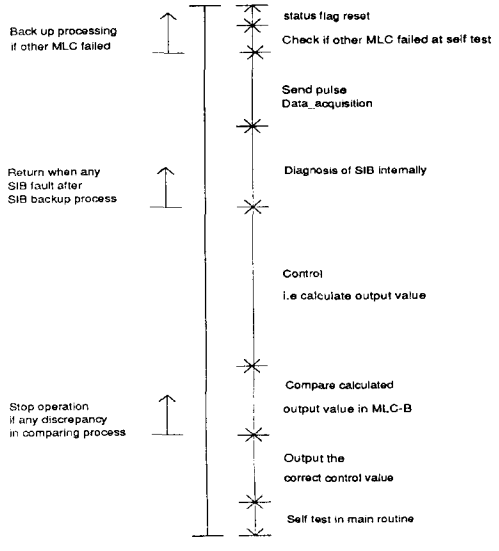

그림 2. 한 샘플링 시간 내의 일 수행도

Fig. 2. The Timing Diagram of Job in One Samling Time

위의 작업 수행을 위한 소프트웨어는 크게 주요 루틴(main routine)과 매 샘플링 타임마다 MLC 내부의 타이머 인터럽트(level 2)에 의한 타이머 인터럽트 처리 루틴, 그리고 A/D 변환이 끝난 후에 SIB가 발생시키는 인터럽트(level 4)에 의한 ADC 인터럽트 처리 루틴으로 이루어진다. 그림 3는 주요 루틴 알고리즘이고, 그림 4는 타이머 인터럽트 처리 루틴 알고리즘이다. 주요 루틴에서는 주로 MLC의 자체 테스트가 이루어지며, 타이머 인터럽트 루틴에서는 플랜트 제어와 관계되는 모든 작업이 수행되는 데, SIB를 통한 데이터 입력, SIB 진단, MLC가 정상임을 나타내는 펄스 발생, 제어 출력값 계산, 제어 출력값의 비교, 그리고 그 값들의 출력 등이 수행된다. 그리고 ADC 인터럽트 처리 루틴 에서는 A/D 변환이 끝났다는 플래그를 세워 변환된 데이타를 MLC가 읽어 갈 수 있도록 해준다.

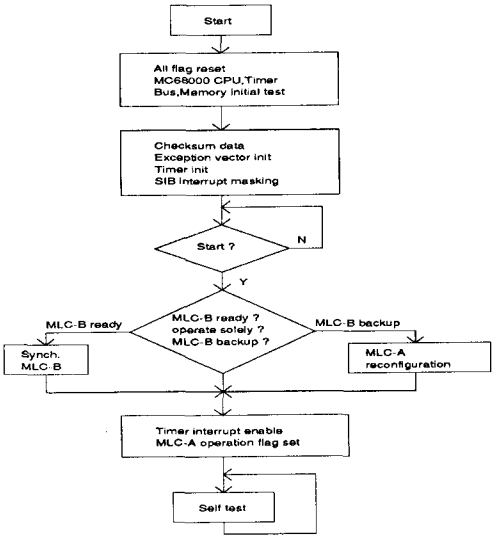

그림 3. 메인 루틴 알고리즘

Fig. 3. The Main Routine Algorithm

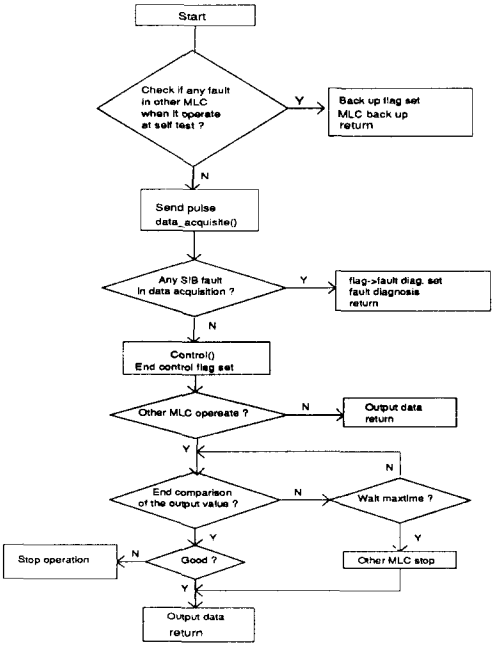

그림 4. 타이머 인터럽트 처리 루틴 알고리즘

Fig. 4. The Algorithm of Timer Interrupt Service Routine

4. 신뢰도 분석

전기전자 공학자 협회의 전기전자 용어 사전에 의하면 자동제어에서의 신뢰도는 한 장치가 특정한 운용 조건하에서 주어진 기간 동안에 목적한 바의 기능을 충분히 수행할

수학적 확률로 정의된다. 이에 대한 척도로써 평균수명, 평균고장간격, 신뢰도 함수(reliability fuction) 등이 있다. 본 논문에서는 정량적으로 신뢰도를 분석하기 위해서, 신뢰도 함수에 의한 방법을 사용하였으며, Markov 모델에 의하여 해석하는 것이 여러 요소를 고려함에 있어서, 융통성을 주기 때문에[7], 본 제어시스템을 Markov 모델로 표현하여 신뢰도 함수를 구했다. 신뢰도의 분석 방법은 Siljak이 중복구조를 갖는 다중 제어기 구조에 대해 해석을 했는 데[8], Siljak의 접근 방법은 상당히 단순화되어 실제와는 거리가 멀기 때문에, Johnson의 접근 방법[7]을 응용하였다.

우선, 본 제어시스템에서 STB와 AMS 부분은 고장의 모델에서 제외를 하고, 또 버스 아비터도 고장이 없다고 가정한다. 그래서, MLC-A와 MLC-B, SIB-A, 그리고, SIB-B에 국한시킨 이중구조의 제어시스템에 대하여 Markov 모델식을 세워 신뢰도 함수를 구했다[9].

일반적으로, λ_1은 MLC의 고장률이고, λ_2는 SIB의 고장률인데, 구조가 간단하고 사용되는 소자가 적을수록 좀 더 고장이 적게 일어난다. 그래서, SIB는 MLC에 비해 소자의 수가 상대적으로 적기 때문에, 고장률도 그만큼 적게 정했다.

그림 5는 어떠한 고장이더라도 완전한 조치가 일어나도록 MLC와 SIB 모두 이중화했을 때와 MLC만 이중화했을 때, SIB만 이중화했을 때, 그리고 모두 단일화했을 때와의 신뢰도 비교인데, 단일구조의 경우는 어느 한 소자라도 고장이 생기면 신뢰성이 없는 것이므로, 완전하게 고장이 처리되는 경우와 비교하였을 때 그만큼 신뢰도가 떨어진다. 그래서, 모두 이중화한 경우와 모두 단일화한 경우와는 신뢰도가 상당히 차이남을 볼 수 있다. 또한 MLC만 이중화했을시와 SIB만 이중화 했을시의 신뢰도는 SIB가 MLC보다 낮은 고장률에도 불구하고 MLC를 이중화했을 경우에 더 신뢰도가 높게 나타났는데 그 이유는 MLC는 핫-스탠드바이 구조로 SIB의 경우보다 많은 방법에 의해 고장을 탐지하여 조처를 취해주기 때문으로 SIB를 이중화한 콜드-스탠드바이보다 MLC를 이중화한 핫-스탠드바이의 신뢰도가 높게 분석되었다.

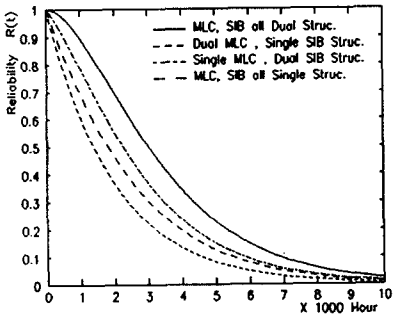

그림 5. 완전 고장 탐지시 MLC, SIB 모두 이중구조, MLC 만 이중 구조, SIB 만 이중 구조, MLC, SIB 모두 단일 구조와의 비교 (λ_1 = 0.00035 /hour, λ_2 = 0.00015 /hour)

Fig. 5. Reliability Comparison with Both MLC and SIB Dual, Only MLC Dual, Only SIB Dual, Both MLC and SIB Single at Exact Fault Detection (λ_1 = 0.00035 /hour, λ_2 = 0.00015 /hour)

5. 실 험

모듈 내의 소자에 고장이 났을 때, 백업 기능을 확인하기 위해 임의로 고장을 발생시키지만, 고장 발생을 위해 고가인 하드웨어의 소자에 손상이 가게 하거나 제거할 수는 없기 때문에 간단히 고장과 같은 상황을 인위적으로 발생시켜 처리하였다.

인위적 고장의 하나의 예로 SIB 모듈을 뽑는 고장을 실행하였으며, 이때 처리되는 순서는 다음과 같다.

i) MLC와의 인터페이스부의 고장 발생

ii) MLC가 SIB로부터 데이타를 입출력하려할 때 예외처리 버스 에러 발생. (fault detection)

iii) 어느 SIB에서 발생되었나를 세워진 플래그로부터 찾음. (fault location)

iv) SIB-B로 펄스를 보내주어 스윗칭. (fault isolation, reconfiguration)

v) MLC-B에 SIB 고장이 생겼음을 알리는 플래그를 세움으로써 그 샘플링 시간에서의 제어 출력 값의 비교는 하지 않도록 함. (fault broadcasting)

vi) 메인 루틴으로 되돌아와 계속 자체 테스트. (회복)

vii) 스윗칭된 SIB로 데이타 입출력.

또 다른 SIB의 고장도 위와 비슷한 방법으로 처리가 되며, 주로 고장의 발생은 점퍼선을 사용하여 오픈 또는 단락 등에 의해 실험하였다.

MLC에서의 고장은 고장 시에 세워질 플래그를 임의로 바꾸어 고장 상황을 만들었다. MLC의 메모리 고장시에 MLC-B는 다음과 같은 과정으로 MLC-A의 고장을 인지하며 백업 조치를 취한다.

i) MLC-A가 자체 테스트 중 메모리 고장이 있었다는 플래그 세움.

ii) MLC-B가 확인하여 MLC-A를 멈추게 하는 플래그 세움.

iii) MLC-B에 의한 제어 동작 실행.

실제로 로컬 버스에는 SIB 모듈이 8개까지 꽂힐 수 있으며, 모두 이중화하면 최대 16장이 꽂히게 된다. 그림 6은 실제 발전소의 보일러를 대신하는 시뮬레이터를 제어하는 도중에 고장이 발생하여 조처를 취한 것을 보인 것인데, 관리 제어시스템을 통해 공기 다이나믹스(air dynamics)를 모니터링한 것이다. 즉, SIB 0번 모듈을 뽑아서 고장을 발생시켰을 때 우상단의 0번에 불이 들어와 SIB 0번의 백업을 나타내 줌을 볼 수 있으며, 모듈을 뽑는 순간 피이크(peak)가 잠시 생겼다가 곧 정상 상태로 가는 것을 볼 수 있다. 이때 생긴 피이크는 SIB 모듈의 고장으로 이상 신호가 받아들여졌기 때문이다. 또한, 우상단 A에 불이 들어온 것은 이 상태에서 MLC-A가 시뮬레이터를 제어하는 것을 멈추게했을 때 MLC-B로의 백업을 나타낸 것이다. 이 때는 SIB 모듈을 뽑을 때처럼 피이크가 생기지 않았는데, 그 이유는 I/O 모듈 스윗칭이 아니기 때문에 이상 출력 또는 이상 입력이 잠시라도 나타나지 않게 되며, MLC의 스윗칭이 끝날 때까지 시뮬레이터와의 데이타 입출력을 중단하고 기존의 출력값을 유지하고 있기 때문이다.

두 MLC 모두 고장시에 수동절환이 됨을 확인하였으며, 이 경우에는 수동 조작에 의해 플랜트를 제어한다.

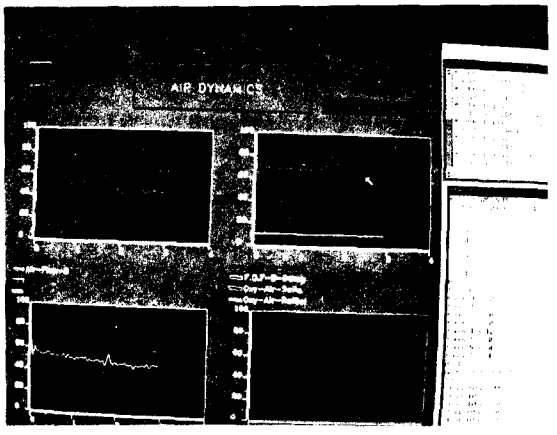

그림 6. 관리 제어시스템에서 본 고장 발생 상황

Fig. 6. The Generated Fault Status Monitored at Supervisory Control System

6. 결론

본 논문에서는 제어시스템의 신뢰도를 높이기 위해 CPU 모듈과 I/O 모듈을 각각 이중화한 제어 구조를 제안했으며, 신뢰도 함수를 이용하여 이중화된 시스템의 신뢰도를 정량적으로 분석했다. 또한, CPU 모듈만 이중화했을 때와 I/O 모듈만 이중화했을 때와의 정량적인 신뢰도 비교도 함께 포함시켰는데, CPU 모듈만 이중화했을 때가 I/O 모듈만 이중화했을 때보다 신뢰도가 높다는 결과를 얻었다.

고장의 탐지 방법에 있어서는 상호 보완적인 방법들을 결합함으로써 오판의 가능성을 줄이고 보다 신뢰성있는 이중화 구조가 될 수 있도록 했다. 또 플랜트의 안정을 위해 두 CPU 모듈 또는 I/O 모듈이 모두 고장이 나더라도 자동으로 수동절환토록 하는 기능도 포함시켰다. 그리고, 몇가지 고장을 인위적으로 발생시켰을 때, 백업 기능에 의해 정상 동작을 유지함을 실험적으로 보였다.

참 고 문 헌

[1] Theordore J. Williams, "The Developement of Reliabilty in Industrial Control Systems", IEEE MICRO, p66-80, 1984.

[2] "마이크로 컴퓨터를 이용한 전자제어 시스템의 고신뢰화에 관한 연구"(최종 보고서), 한전기술연구원, Feb. 1988.

[3] 신영달, "Boiler backup control을 위한 Multiprocessor 방식에서의 신뢰도 개선에 관한 연구", KAIST 석사논문, 1987.

[4] 허성광, "발전소 제어계통의 고장 진단", KAIST에서의 세미나 자료, May. 13. 1989.

[5] William F. McGill, Steven E. Smith, "Fault Tolerance in Continuous Process Control", IEEE MICRO, p22-33, 1984.

[6] "Bailey Network 90 (Multi-Function Controller Module - NMFC05)", Catalog, E93-906-13, 1988.

[7] Barry W. Johnson, "Reliability & Safety Analysis of a Fault-Tolerant Controller", IEEE Trans. on. Reliability, Vol.R-35, No.3, p515-524, 1983.

[8] D.D. Siljak, "Reliable Control using Multiple Control Systems", INT. J. Control, Vol.31, No.2, p303-329, 1980.

[9] 박세화, "이중구조를 갖는 제어시스템의 구현과 신뢰도 분석에 관한 연구", KAIST 석사논문, 1990.