

신뢰성 있는 실시간 분산제어 시스템의 성능분석에 관한 연구

A Study on the Analysis of Performance for a Real-time Distributed Control System with Reliability

金來鎭*, 朴仁甲**
(Nae-Jin Kim* and In-Kap Park**)

요 약

네트워크 발전으로 인하여 제어시스템은 집중화에서 분산화 된 아키텍처를 지향하고 있다. 그러나 이러한 제어시스템은 범용 운용 시스템을 기반으로 설계되어 공정처리에서 요구되는 실시간 성능을 보장하기는 미흡한 점이 많아 플랜트의 제어설비로서 실시간 운영체제를 탑재한 제어시스템이 점차로 증가하고 있다.

본 논문에서는 분산제어 시스템의 기능을 위해 운영체제는 실시간 성능을 보장하는 조건을 만족하도록 실시간 처리 능력, 실시간 응답의 신뢰성 및 멀티 프로세스의 기능구현에 중점을 두었으며, 이 운영체제의 바탕 위에 주요 공정처리의 기능인 루프, 로직의 가능한 처리 수를 분석하고, 이의 타당성을 검토하였다. 그리고 온라인 시 처리중인 데이터에 영향을 미치지 않는 시스템을 설계하고, 절체에 소비되는 시간을 측정하였다.

Abstract

As the network technologies advance, the control systems progress from a centralized architecture to a distributed one. However, these control systems were designed mostly based on the general-purpose operating systems(OS) and have many problems for assurance of a real-time property required for plant processing fields. Therefore, the control systems for a plant process upon real-time OS have been increased gradually.

In this paper, the real-time OS emphasizes on the realization of real-time processing capability, reliability of real-time response, and multi-processing functionality which are prerequisites for a distributed control system. And on the basis of this OS, the number of executable loop and logic, the functions of main plant processing, was analyzed and its validity was also evaluated. The system in this paper was designed not to effect on processing data while online, and the time spent on switching was measured.

I. 서론

네트워크 발전으로 인하여 제어시스템은 집중화에서 분산화 된 아키텍처를 지향하고 있다. 그러나 이러한 제어시스템은 범용 운용시스템을 기반으로 설계되어 공정처리에서 요구되는 실시간 성능을 보장하기에

* (株)光明制御
(Kwang Myung Industrial Controls Co., Ltd.)

** 建國大學校 電子工學科
(Dept. of Electronic Eng., Konkuk Univ.)

接受日: 1998年8月27日, 修正完了日: 1998年12月22日

는 미흡한 점이 많아 플랜트의 제어설비로서 실시간 운영체제를 탑재한 제어시스템이 점차로 증가하고 있다[1].

본 논문에서는 이러한 개방형 시스템을 위한 실시간 분산제어 시스템의 설계 및 분석을 하였고, 신뢰성 있는 이중화 시스템 구현을 목표로 하였다. 이러한 분산제어 시스템의 기능을 위해 운영체제는 실시간성을 보장하는 조건을 만족하도록 실시간 처리 능력, 실시간 응답의 신뢰성 및 멀티 프로세스의 기능구현에 중점을 두었다. 이러한 운영체제의 바탕 위에 프로세스 처리에 필요한 시간을 측정하여, 분산제어 시스템의 성능 척도인 루프·로직의 가능한 처리 수를 분석하고, 이의 타당성을 검토하였다. 또한 공정중에 예기치 못한 결함 발생 시 공정에 영향을 미치지 않는 결함내성을 연구하였고, 하드웨어 및 소프트웨어의 데이터 처리시간을 분석하고 이중화 절체에 필요한 처리시간도 측정하였다.

그리고, 본 논문은 온라인 시 I/O보드의 탈착이 가능하며, 각 포인트의 채널단위 진단기능이 가능한 소프트웨어 및 하드웨어를 구현하였다.

II. 실시간 분산제어 시스템

본 연구의 대상이 되는 분산제어 시스템은 온라인 시 에러복구 기능은 물론 루프·로직 처리 알고리즘을 수행 중에 분석 가능하도록 하여 최적의 시스템 베이스로 플랜트 공정 서비스를 제공하도록 하는 시스템이다.

그림 1은 이러한 구조를 나타내는 분산제어 시스템의 전체구성도를 보여주고 있으며, 크게 POS(process operating station)와 PRS(process remote station)로 나누어진다. 중앙의 상위 스테이션인 POS는 그래픽 처리를 위한 XPU(x-window process unit)와 데이터를 처리하는 NPU(network process unit)로 나뉘고, PRS는 플랜트 포인트 제어를 실제 담당하는 프로세스 장치이다.

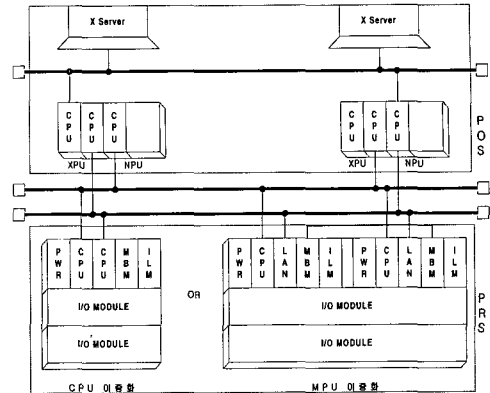


그림 1. 분산제어 시스템의 전체구성도

Fig. 1. configuration of DCS.

그림 1의 분산제어 시스템은 여러 PRS가 플랜트 공정을 처리하면서 POS와 통신이 두절되더라도 독립적으로 서비스를 계속할 수 있는 구조로 되어 있으며, 시스템의 확장성과 신뢰성을 위해 표준화된 구조를 지향하며, 진단기능과 제어명령 및 감시데이터의 처리를 위해 고속 LAN으로 연결된다.

PRS 시스템의 신뢰성을 위하여 하드웨어의 완전 이중화 구조인 MPU(main process unit) 이중화를 설계하였다. MPU 이중화라 함은 개별 버스에 각각의 CPU를 구성하고 LAN 이중화 보드를 이용하여 구성한 것으로 각 PRS의 CPU이중화 및 데이터 통신을 이중화할 수 있는 구조를 의미한다.

2.1 하드웨어 구현

플랜트 처리를 담당하는 PRS는 그 중요도에 따라 결함내성 서비스를 여러 형태로 변형시키는데, 이중 가장 중요한 공정처리를 하기 위한 결함내성 구현 방법은 별도의 버스를 사용하고 하드웨어와 소프트웨어를 완전 이중화하여 설계하는 것이다. 이런 설계 구성 방법은 빠른 절체와 이중화된 시스템간의 데이터 일치 문제가 대두되지만 신뢰성 측면에서 사용자에게는 안전한 시스템을 제공할 수 있다.

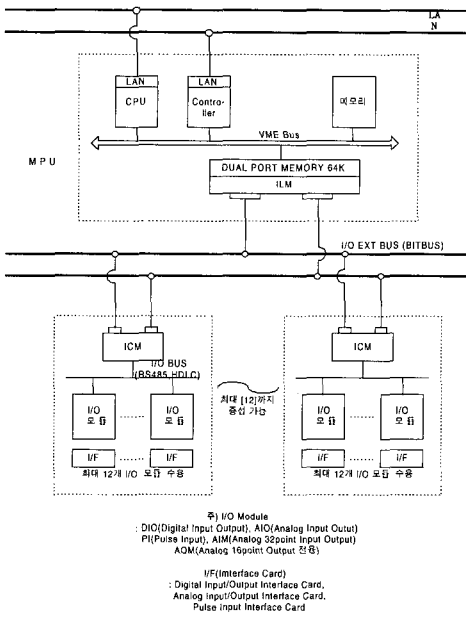


그림 2. PRS 시스템 구성도
Fig. 2. system block diagram of PRS.

그림 2는 PRS의 전체 구성도로서 각 PRS의 CPU는 연산을 위해 ILM을 통해서 다른 버스의 모듈정보를 읽어오고 이 ILM은 입·출력 모듈을 관장하는 ICM으로부터 데이터를 수신한다. 또한 각 ICM은 입·출력 모듈 12장까지 포인트 제어를 할 수 있도록 되어 있다.

본 논문에서는 기존 분산제어 시스템이 채널단위 진단기능과 온라인 시 I/O보드의 탈착이 불가능한 구조로 되어있는 것을 개선하기 위해 PRS를 구성하는 CPU와 LAN 콘트롤러, 메모리, ILM, ICM과 I/O보드의 설계 및 구현을 할 것이다. 그래서 핫스왑 기능의 제공과 채널 온라인 기능을 부여함으로써 공정 중 특정 하드웨어의 결함 시 현재 처리중인 채널에 영향을 미치지 않고 이상이 생긴 하드웨어의 교체 또는 수리를 할 수 있으며 각 채널의 이상유무를 온라인시 볼 수 있도록 하였다.

2.1.1 ILM 보드의 구현

ILM(input/output link module)보드는 VMEbus 시스템과 2개 포트의 동기 또는 비동기 시리얼 통신 장치

와 인터페이스 할 수 있는 시리얼 포트 및 2개의 bitbus 채널을 갖는 시스템 콘트롤러이다. ILM은 기동과 동시에 각 ICM과 I/O보드의 상태를 조사하여 동작 가능 상태와 동작여부를 저장한다. 상위 POS로부터 다운로드 된 구성정보와 ICM으로부터 업로드된 구성정보를 비교하여 저장하고, 매 30ms로 ICM과 통신한다. ILM은 최대 12장의 ICM과 통신할 수 있다.

2.1.2 ICM 보드의 구현

ICM(input/output control module)은 상부로부터 내려오는 제어명령을 이중화된 bitbus라인을 통하여 받아들이며 각 I/O 모듈들을 제어하는 기능을 한다. 이 ICM 동작은 PRS CPU와 ILM을 통하여 통신하며 해당 rack에 설치된 12장의 I/O 모듈들을 제어한다.

2.1.3 I/O 보드의 구현

I/O보드는 DIO(digital input/output module)와 AIO(analog input/output module)로 분류된다. DIO는 32개의 디지털 입·출력 포인트를 갖는 디지털 입·출력 보드로서 외부와는 필요시 별도의 인터페이스 모듈을 사용하여 필드와 접속한다. 또한 AIO는 16개의 아날로그 입·출력 포인트를 갖는 보드이다.

그림 3은 I/O 보드의 블록도를 보여주며, I/O 채널 진단을 위한 CPU가 탑재되어 있다. CPU는 채널의 이상유무를 확인하여 ICM으로 진단 데이터를 전달하며 ICM은 ILM의 폴링 요청 시 자신이 담당하는 I/O 데이터를 응답한다.

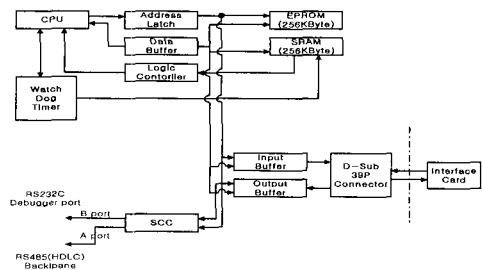


그림 3. I/O 보드의 하드웨어 블록도
Fig. 3. Hardware block diagram of a I/O board.

2.1.4 메모리 보드의 구현

메모리 보드는 MPU에 이상이 생길 경우를 대비하여 부팅 시 상위 POS로부터 데이터의 재 다운로드를 받지 않기 위한 기능 및 이중화 처리를 위해 현재 주로 동작하는 MPU 데이터를 중 MPU로 일치시키기 위한 기능을 제공한다. 메모리 간 데이터 일치를 위해 96PIN SCSI (small computer system interface) 케이블로 연결하고, 2M 바이트 SRAM으로 전원 오프 시 데이터가 삭제되지 않도록 한다.

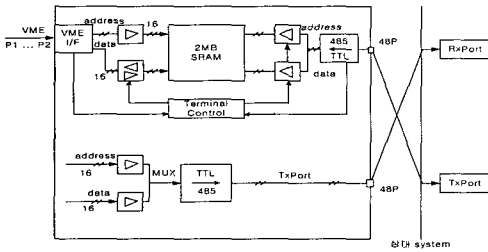


그림 4. 메모리 보드의 하드웨어 블록도
Fig. 4. Hardware block diagram of a memory board.

2.2 시스템 소프트웨어 구현

시스템 소프트웨어는 플랜트 공정을 처리하는 시스템 운영 소프트웨어와 신뢰성을 위한 결함내성 소프트웨어로 나뉘어진다. 시스템 운영 소프트웨어는 분산제어 시스템의 상위 POS 및 하위 PRS에서 실행되며, 분산제어 시스템의 고유기능인 루프·로직 및 플랜트 운영을 위한 소프트웨어이다. 결함내성 소프트웨어는 PRS의 이중화 처리를 위한 소프트웨어로서 PRS의 치명적인 결함에 대비하여 PRS의 다운이 생겨도 현재 처리중인 공정에 이상 없이 처리할 수 있는 소프트웨어이다.

상위 POS의 운영자화면 MMI(man machine interface)에서 사용자가 제어나 데이터의 전달을 위해 전송을 시작하면, 이 내용을 NPU 버퍼에 저장한다. NPU는 이 버퍼에 저장된 데이터를 네트워크 통신 프로그램을 통해 PRS로 보낸다.

상위 POS 스테이션은 누산 데이터를 수집, 보관, 관리하고, 레포트 데이터의 수집 및 인쇄요구 시 생성된 포맷 데이터를 인쇄한다.

또한 PRS의 독립적인 운영을 위하여 다운/업로드 기능을 제공하며, 아날로그 추이를 보여주는 트랜드 기능을 제공한다.

NPU는 실시간으로 XPU의 제어명령을 PRS로 보내주기 위한 기능 수행이나 PRS의 정주기 데이터를 XPU로 전달하기 위한 기능으로서 그림 2는 NPU의 대략적인 타스크 구조이다.

그림 5에서 posras (process operating station reliability-availability-serviceability task)는 각 PRS의 네트워크가 이상이 있는지 감시하고, bcsrv (broadcast server task)는 PRS의 정주기 데이터를 전달받아 상위 POS로 전달하기 위해 공유 메모리에 저장하는 기능을 하며, cmdsnd (command send task), cmdrcv (command receive task)는 상위 POS의 비주기 명령을 전달받아 PRS로 전달하거나, PRS 응답을 POS로 전송하는 기능을 담당한다.

cmdmgr (command manager task)은 cmdsnd의 제어 명령에 대한 시간을 측정하여 무 응답 시 재전송을 수행하는 기능을 한다.

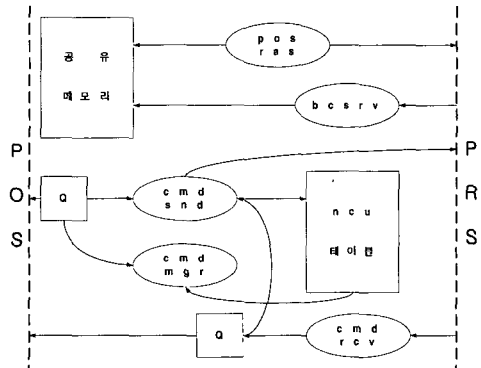


그림 5 NPU 타스크 구조도
Fig. 5. task configuration diagram of NPU.

PRS는 상위 POS로부터 다운로드 데이터를 전달받아 플랜트공정의 감시,제어를 수행하는 역할을 하며 그림 6에 소프트웨어 구성을 나타내었다.

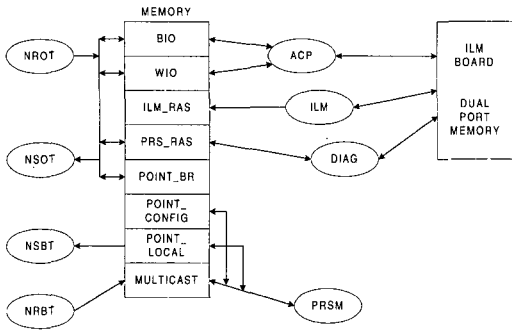


그림 6 PRS 태스크 구조도

Fig. 6. task configuration diagram for a PRS.

그림 6에서 nrot (network receive operating task) 및 nsot (network send operating task)는 비 주기적인 명령 처리를 위한 태스크로 PRS의 포인트 제어 및 속성을 변경시킬 때 이용된다. 또한 nsbt (network send broadcast task) 및 nrbt (network receive broadcast task)는 주기적인 명령처리를 위한 태스크로 상위 POS의 MMI화면 디스플레이 요구나 정주기 PRS 데이터를 전달할 때 이용된다. 그리고 PRS의 주 태스크로서 prsm(process remote station main)이 존재하며, 각 이벤트 발생마다 루프처리를 위한 acp(advanced control point) 태스크를 호출하며, 루프처리의 결과가 1초 미만이면, 남아있는 시간동안 로직처리를 수행한다. 또한 I/O보드로 데이터를 송·수신하기 위한 ilm(I/O link module), PRS내 모든 보드 상태를 진단하는 diag(diagnostic) 태스크가 존재한다.

이중화를 통한 PRS의 공정 처리 중 신뢰성을 높이기 위한 방법으로 본 논문에서는 시스템을 주·중 MPU로 분류하여 주 MPU에서 실제 제어에 관한 동작과 공정 데이터를 처리하도록 하였으며, 이 처리된 데이터는 자신의 메모리 카드 내 메모리에 저장하고, 데이터 일치를 위해 주 MPU의 저장된 정보를 메모리 보드의 PLD(programmable logic controller) 로직을 이용하여 상대 메모리에 곧바로 전송한다. 전송은 메모리 보드가 동작상태인 경우에만 전송하고 대기상태에서는 데이터의 수신만이 가능하다. 그리하여 주 MPU와 중 MPU는 항상 같은 플랜트 정보를 유지시킨다. 따라서 주 MPU의 치명적 오류시 중 MPU는 절

체되어 주요 공정인 루프처리에 관한 일을 제일 먼저 수행하게 되고 시간이 허용된다면 계속해서 로직이나 다운로드 정보 참조 및 포인트 입출력에 관계된 작업을 수행해 나갈 수 있도록 하였다.

결함내성을 갖는 실시간 시스템은 그림 7과 같이 각 노드별로 주요 제어부분인 MPU를 이중화하여 어떤 부분에 예기치 못한 이상이 발생하더라도 시스템이 정상적으로 동작하도록 설계하였다.

그림 7에서 주 MPU와 보조 MPU간에는 데이터 일치를 위한 SCSI 케이블이 존재하며, 플랜트 정보에 대한 I/O보드 데이터를 수집하는 ICM 보드와 ILM 보드가 주·중 MPU에 각각 존재한다.

또한 ILM은 ICM을 이중화 할 수 있도록 ICM과 통신할 수 있는 포트를 두 개 가지고 있으며, 각 ICM은 보드의 상태진단 및 온라인 시 탈착이 가능하도록 전송 방법을 bitbus에서 RS422방식으로 바꾸어 처리하였으며, 각 I/O보드는 CPU를 탑재하여 보드의 이상유무를 진단할 수 있도록 하였다.

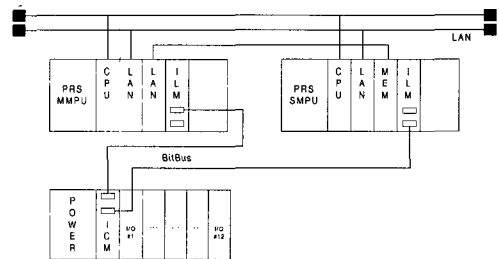


그림 7 이중화된 PRS 시스템의 하드웨어 구성도

Fig. 7. hardware configuration diagram of duplicated PRS system.

III. 실험 및 고찰

시스템의 신뢰성 향상을 위한 구현 방법은 여러 가지가 있겠지만 오류 발생을 사전에 완전히 배제하기란 실제로 거의 불가능하므로 오류 발생 시 대응하는 결함내성(fault tolerance) 방법으로 분산 시스템의 신뢰성을 향상시키는 연구가 진행되고 있다.

본 연구는 시스템의 신뢰성 향상을 위해 루프처리의 공정을 담당하는 PRS의 제어 부분을 이중화 하

여 예기치 못한 결함이 발생하더라도 공정처리를 위한 PRS의 다운을 유발시키지 않도록 하여 궁극적으로 시스템 이상 발생 시 처리 중인 루프 처리는 진행시킬 수 있도록 하였으며, 성능 평가를 통하여 시스템에서 처리 가능한 loop 수와 logic 처리 수를 알아보았다.

3.1 시스템 포인트의 성능평가

우선 본 논문에서는 운영체제의 성능 평가를 위해 작업 실행 시간의 한계를 계산하고 이 작업 시간의 한계는 전체 프로세서 사이클의 몇 퍼센트 이하를 차지한다는 것만 증명하면 작업이 언제 수행될지 알지 못하더라도 이 작업의 양은 상한선 이내에 달성됨을 보장할 수 있도록 하는 것이다.

이러한 방법은 비율 단조 스케줄링(rate monotonic scheduling theory: RMT)을 이용하여 처리한다. 이 RMT 이론은 프로세스 이용도가 어떤 경계값 이하이면 그 집합에 있는 모든 작업들은 주어진 작업이 언제 실행될지 모르더라도 그들의 상한 조건을 만족시킬 것이라는 것을 보장하는 이론이다[2]. 따라서 본 논문에서는 비율단조 알고리즘에 의해 스케줄링된 n개의 독립적이고 주기적인 작업들이 다음조건을 만족하도록 한다.

$$\frac{C_1}{T_1} + \dots + \frac{C_n}{T_n} \leq U(n) \quad (1)$$

식 (1)에서 U는 프로세서 사용률, C와 T는 작업 실행 시간과 주기이다. U(n)는 n개의 작업에 대해 프로세서가 사용 가능한 최대 비율, 즉 스케줄링 한도(scheduling bound)이다.

식 (1)은 모든 작업들이 전체 프로세서 사이클의 몇 퍼센트 이하를 차지한다면 작업이 언제 수행되었는지 정확히 알지 않고서도 상한선(deadline)이내에 달성됨을 보장할 수 있는 방법이다.

식 (1)은 다음과 같은 식으로 나타낼 수 있다 [3][4].

$$U(n) = n(2^{1/n} - 1) \quad (2)$$

위의 수식 (2)는 RMT적용의 예로서 작업수가 5가 지라면 U(5)=0.744 가 된다. 이는 프로세서 사이클의 74.4%까지 사용할 수 있다는 의미이다. 그러나 시스템의 설계가 복잡하여 어떤 시간에 얼마나 많은 작업이 수행될 지 모를 때가 있다. 다행히 RMT의 수식을 이용하면 U(n)의 n이 무한대로 갈 때 log₂(약0.693)에 수렴함을 알 수 있다. 이는 얼마나 많은 작업들이 수행되는가에 상관없이 작업을 위한 프로세서 스케줄링의 상한선을 보여준다.

그러나, 위의 수식은 플랫폼에서 처리되는 모든 공정마다 처리되는 작업시간의 한계가 다르기 때문에 대신 본 논문에서는 시스템 타스크의 메모리, 커널 메모리, 라이브러리 사용량 및 각 PRS의 동작을 위한 데이터 메모리, PRS간 데이터를 주고 받는 멀티 캐스트 메모리와 PRS 데이터를 상위에서 참조하기 위한 IJF 테이블을 조사하였다.

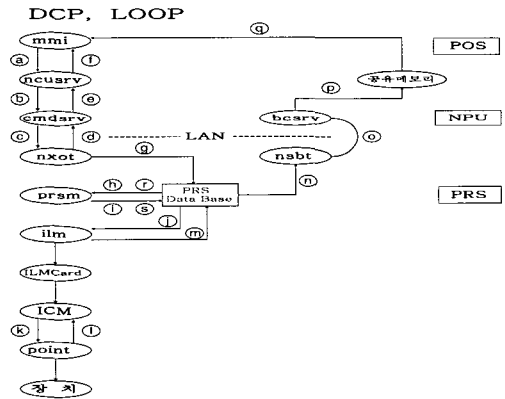


그림 8. 로직, 루프 처리과정
Fig. 8. logic and loop processing flow.

그러므로 본 논문의 시스템에서는 온라인 시스템 부하율을 볼 수 있도록 하여 PRS에서 실행되는 포인트 데이터 처리가 실패될지, 성공할지 분석할 수 있도록 하였다.

그림 8은 본 논문의 시스템이 로직과 루프를 처리하는 과정으로서 DCP(discrete control point)는 로직 처리를 한다. 이중 DCP나 루프 포인트를 PRS가 처리하는 과정은 (h)+(i)+(j)+(k)+(l)+(m)+(r)+(s) 이고 처리시간은 20+30+10+10+10+20+30+10 인 160ms가 걸린다.

이것은 본 논문에서 ㉔의 과정은 비동기로 처리되며, 지연이 가장 많을 때를 기준으로 정한 시간이다. 또한 루프포인트 처리 주기가 160ms이고 각 루프포인트 1개의 처리시간이 0.2ms라고 가정하면 앞에서 설명한 RMT이론을 이용하여 실제 처리할 수 있는 루프의 개수를 알 수 있다. 즉 한 개의 루프 작업주기는 160ms가 되고 처리시간은 0.2ms가 된다. 다음 루프동작을 위한 작업시간은 처음 루프처리 시간에다 다음 루프처리 시간인 0.2ms를 더한 0.4ms가 2번째 루프 처리의 작업시간이 된다. 이렇게 처리된 수식은 식 3과 같다.

$$\left(\frac{0.2}{160} + \frac{0.4}{160} + \frac{0.6}{160} + \dots + \frac{6.4}{160}\right)ms \quad (3)$$

식 (3)은 다시 식 (4)와 같이 표시할 수 있다.

$$0.2\left(\frac{1}{160} + \frac{2}{160} + \frac{3}{160} + \dots + \frac{32}{160}\right)ms \quad (4)$$

식 (4)을 계산하면 0.66임을 알 수 있다. 식 (3)과 식 (4)는 RMT이론의 비율 단조 증가함을 알 수 있으며, 본 논문의 루프처리가 32개일 경우 최대 작업 부하율은 66%임을 알 수 있다. 만약 루프포인트가 33개일 경우 시스템 부하율은 70% 이상이 되므로 인하여 RMT이론의 최대부하율 69.3%를 넘게된다. 이것은 본 논문에서 가장 낮은 우선 순위 태스크를 이용한 시뮬레이션 결과와 일치한다. 따라서 시스템의 최대 루프포인트 수는 32개임을 알 수 있다.

또한 루프포인트 처리시간이 160ms이므로 1초 이내의 로직포인트 처리를 위한 시간은 840ms이다. 즉 구현된 시스템은 1초 이내에 루프와 로직을 처리하므로 로직포인트의 주기시간 결정은 루프포인트 처리시간을 제외한 시간이 된다는 것이다. 따라서 로직포인트의 처리시간이 840ms이고 1ms의 작업시간을 가지는 로직포인트는 1080스텝처리가 대략 69.5%의 부하율을 보인다. 따라서 구현된 시스템의 최대 로직포인트 개수는 1080포인트 이하를 처리함을 알 수 있다.

3.2 결합내성의 성능평가

MPU의 절체시간은 자동과 수동으로 나누어 처리하며 절체과정은 그림 5와 같다.

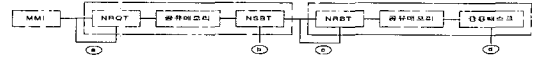


그림 9. 시스템 절체과정

Fig. 9. switching over method of DCS.

그림 6에서 ㉑ 부분은 MMI와 주 MPU간 네트워크 통신으로서 처리시간은 2ms이다. ㉒는 주 MPU의 NBST 처리시간 500ms를 가리킨다. ㉓는 주 MPU와 종 MPU간 네트워크 통신속도이며 7ms에 처리된다. ㉔는 종 MPU의 응용 처리시간으로서 200ms이다.

절체시간에는 데이터베이스 갱신 시간 21ms^[5]가 더해지는데 운영자에 의한 수동절체일 경우 최단 절체는 ㉑ + ㉓에 의한 절체로 30ms가 소요된다. 또한 최장절체는 ㉑ + ㉒ + ㉓ + ㉔에 의한 절체로 730ms가 소요된다. 또한 자기진단에 의한 자동절체로서 최단절체는 ㉓로서 2ms가 소요되고 데이터베이스 갱신 시간을 더한 23ms가 걸린다. 그리고 자기진단에 의한 절체의 최장절체는 ㉒ + ㉓ + ㉔로서 728ms가 절체에 소비되는 시간이다.

IV. 결론

본 논문에서는 플랜트 공정처리에 중요한 역할을 차지하고 있는 분산제어 시스템의 설계와 이에 대한 성능분석을 하였다.

하드웨어적으로 신뢰성을 위해 PRS내 각 보드의 진단 및 채널단위 진단까지도 가능하도록 하였고, 특히 기존의 분산제어 시스템이 채널 단위 진단 기능이 미비한 약점을 보완하였다. 또한 온라인 시 I/O 보드의 활착이 가능하도록 하여 보드의 이상 시 전원을 오프시키지 않고도 이상이 생긴 I/O 보드를 교체, 수리 가능하도록 하였다. 이렇게 하기 위해 각 I/O 보드가 채널 단위 진단 정보를 ICM에 전달하기 위하여 CPU를 두었으며, 진단방법은 피드백 회로를 통하여 출력 값에 대한 입력을 되돌려 받는 방법을 이용하였다. 또한 I/O 보드의 활착을 위하여 기존 bitbus 체제의 I/O 보드를 RS-485방식으로 전환하여 처리하였다.

또한 신뢰성을 위한 PRS 이중화를 구현하기 위하여 전체 PRS 이중화가 아닌 MPU 부분만 이중화하

여 성능과 비용면에서 효과적인 시스템 설계를 하였다. MPU의 이중화 처리 시 LAN을 통한 데이터의 일치는 LAN 부하가 많아질 때, 즉 플랜트 I/O가 많아 PRS 수가 많아지면 주기적인 핫 백업이 불가능하므로 메모리 보드를 이용하여 데이터의 일치를 이루었다.

소프트웨어적으로는 분산제어 시스템이 가져야할 응용 소프트웨어 및 결합내성 소프트웨어를 구현하였다.

구현된 시스템의 분석을 위하여 실시간 시스템의 성능분석 척도인 부하율을 측정하여 구현된 시스템이 얼마나 많은 로직 및 루프 포인트 처리를 할 수 있는지 RMT 이론을 통하여 알아보았고, 이 이론이 맞는지의 타당성 검토를 위하여 가장 낮은 우선순위 타스크를 본 논문에서는 실행시켜 보았으며, 결과는 거의 유사함을 알 수 있었고, 이러한 루프처리의 최대 수행 가능한 포인트 수는 32개이며, 로직 포인트의 최대 수행 가능한 수는 1080 스텝이 걸림을 보았다.

그리고 결합내성에 대한 성능평가는 진행 중인 공정상에서 운영자요구에 의한 절체 및 하드웨어 결합에 의한 MPU 절체 시 계열변경에 대한 처리 속도를 측정하였다. 각 단계별 수행 시간을 합산하여 소요되는 총 수행시간을 분석한 결과 운영자 요구에 의한 절체 시 최단 시간은 30ms가 소요되고 최장 시간은

730ms이며, 하드웨어 결합 시 자기진단에 의한 최단 절체는 23ms가 걸리며 최장시간은 728ms가 절체에 소비되는 시간이었다.

참고문헌

- [1] George Coulouris, *Distributed Systems Concepts and Design*, Addison-Wesley Publishing Company, Second Edition, 1994.
- [2] John T. Baldwin, *Realtime Performance 예측과 추정*, 월간전자기술, 1995. 10월호.
- [3] C.Liu and J. Layland, *Scheduling Algorithms for multi-programming in a hard real-time environment* Journal of the ACM, 20(1), pp 46-61, Jan 1973.
- [4] J.P.Lehoczky, L.Sha and Y.Ding *The rate monotonic scheduling algorithm; Exact characterization and average case behavior* in proc of IEEE real-time systems Symp. pp 166-171, 1989.
- [5] 김내진, 원거리 통신을 제공하는 실시간 분산제어 시스템의 성능평가 및 분석, 건국대학교 박사학위 청구논문, 1998. 8.

저 자 소 개



金來鎭 (學生會員)

1966년 3월 21일생. 1989년 2월 건국대학교 전자공학과 졸업(공학사). 1991년 8월 건국대학교 전자공학과 대학원 졸업(공학석사). 1998년 8월 건국대학교 전자공학과 대학원 졸업(공학박사). 1991년 7월 ~ 현재 (주)광명제어 연구소 선임연구원. 주

관심분야는 DCS, SCADA등임.



朴仁甲 (正會員)

1950년 2월 1일생. 1973년 고려대학교 전자공학과 졸업(공학사). 1979년 3월 고려대학교 대학원 전자공학과 졸업(공학석사). 1986년 8월 고려대학교 대학원 전자공학과 졸업(공학박사). 1989년 7월 ~ 1990년 6월 Oregon State Univ. Post

doc. 1980년 9월 ~ 현재 건국대학교 전기전자 공학부 교수. 1995년 1월 ~ 현재 대한전자공학회 이사. 주 관심 분야는 computer network, parallel processing, distributed control system 등임.