

# HMI 소프트웨어 작업 특성을 고려한 분산제어시스템 설계

이건우<sup>o</sup>, 지정훈, 우균  
부산대학교 컴퓨터공학과

leoric99<sup>o</sup>@pusan.ac.kr, jhji@pusan.ac.kr, woogyun@pusan.ac.kr

## (Design a Distributed Control System Considering a HMI Software Task Characteristic)

Gunwoo Lee<sup>o</sup>, Junghoon Ji, Gyun Woo

Dept. of Computer Engineering, Pusan National University

### 요 약

분산제어시스템은 지리적으로 분산되어 있는 제어장치들을 제어시스템과 HMI(Human Machine Interface) 소프트웨어를 통하여 감시제어 한다. 사용자는 HMI 소프트웨어를 사용해 전체 시스템을 제어한다. HMI 소프트웨어는 통신, 저장, 화면 갱신, 데이터 분석 등 모든 작업을 수행한다. 분산제어시스템의 특성상 항상 동작 상태에 있어야 하기 때문에 HMI 소프트웨어의 부담은 고장 가능성을 증가시킨다. HMI 소프트웨어의 작업은 실시간 특성을 기준으로 나눌 수 있다. 작업은 실시간 데이터를 이용하는 감시와 레포트(report)와 데이터 분석을 위하여 저장된 데이터를 이용하는 작업으로 나눌 수 있다. 본 논문이 제안한 분산제어시스템에서는 제어시스템(PCS)과 제어감시 소프트웨어(HIS)가 작업의 특성에 따라 동적(dynamic)으로 연결된다. HIS는 동적연결을 통하여 디스크 쓰기 동작을 줄이고 제어시스템으로부터의 데이터 전송을 감소시킨다. 결과적으로 제어시스템과 HMI 소프트웨어의 작업량을 줄임으로서 분산제어시스템의 성능향상과 유효성을 증가시킨다. 본 논문에서는 HMI 소프트웨어의 작업(task) 특성을 고려한 분산제어시스템 아키텍처를 제안하고 실험을 통하여 아키텍처 성능을 평가하였다.

### 1. 서 론

분산제어시스템(DCS : Distributed Control System)[1]은 Honeywell[2]과 Yokogawa[3]에 의하여 1975년에 처음 소개되었다. Honeywell과 Yokogawa는 비슷한 시기에 TDC-2000과 CENTUM 시스템을 생산하였다. DCS 시스템은 제조, 석유 화학 플랜트, 철강 등 산업분야 전반에서 사용된다. 또한 수자원 관리, 교통신호 관리, 발전소 등 기간사업에도 이용되고 있다.

DCS 시스템은 크게 스위치, 계측기, 펌프와 같은 제어장치와 제어장치로부터 데이터를 수집하고 제어신호를 보내는 제어시스템(예를 들어: FCS-Field Control Station), 사용자와 제어시스템을 연결하는 HMI(Human Machine Interface)로 구성된다. HMI는 사용자가 시스템을 제어할 수 있도록 하는 제어감시 소프트웨어이다.

지리적으로 분산되어 있는 제어장치들은 측정된 값들을 물리적으로 연결된 제어시스템에 전달한다. 그리고 제어시스템들은 이더넷 또는 시리얼(RS-232)을 통하여 HMI 스테이션으로 데이터를 전달한다. HMI 스테이션은 HMI 소프트웨어가 설치되어 있는 컴퓨터이다. HMI 소프트웨어는 전달받은 데이터를 프로세싱, 분석, 기록을 수행한다. 최종적으로 사용자는 HMI 소프트웨어를 통하여 제어장치로부터 전해진 데이터를 감시한다. 제어 또한 HMI 소프트웨어를 통하여 명령을 내린다.

분산제어시스템에서는 HMI 스테이션을 여러 개 배치한다. HMI 스테이션은 작게는 2개에서 많게는 8개 이상 된다. HMI 스테이션을 많이 배치하는 이유는 분산제어시스템의 안정성을 높이기 위해서다. 분산제어시스템에서 사용자가 제어장치들을 제어할 수 있는 최선의 방법은 HMI 소프트웨어를 통해 명령을 내리는 것이다. 물론 하드웨어를 직접적으로 제어하는 방법이 있지만, 분산되어 있는 제어장치는 수백 미터에서 수십 킬로미터까지 떨어져 있을 수 있기 때문에 위급한 상황에서 하드웨어를 직접 제어하는 것은 힘들다. 이는 HMI 스테이션의 멈춤은 곧 분산제어시스템의 통제 불능 상태와 같다는 것을 의미한다. HMI 스테이션을 많이 두는 것이 장점만 있는 것은 아니다.

HMI 스테이션을 여러 개 두는 것은 분산제어시스템의 안정성을 높이지만 제어시스템과 HMI에 많은 부하를 가져 온다. 하나의 제어시스템은 모든 HMI 소프트웨어에 데이터를 전송해 주어야 한다. 그리고 하나의 HMI 소프트웨어는 모든 제어시스템으로부터 데이터를 받아야 한다. 분산제어시스템은 대부분 실시간으로 제어와 감시를 하기 때문에 제어시스템과 HMI 소프트웨어 사이에 주고받는 데이터의 양은 상당히 크다. 또한 HMI 소프트웨어는 데이터 수집뿐만 아니라 디스크에 데이터를 기록하고, 실시간 트렌드(real-time trend) 그래프를 화면에 출력하고, 제어로직에 따라 알람(alarm)을 발생시키고, 이를 외부 프린트나 휴대폰으로 SMS 문자를 전송을 해야 한다. 실시간 특성이 강한 분산제어시스템에서는 제어시스템과 HMI 소프트웨어에 걸리는 부하를 줄이는 것이 필요하다.

본 논문에서는 HMI 소프트웨어의 작업(task) 특성을 고려한 분산제어시스템 아키텍처를 제안한다. HMI 소프트웨어의 작업은 실시간 특성을 기준으로 나눌 수 있다. 데이터들의 변화 추이를 보여주는 실시간 트렌드나 아키텍처 그림을 이용한 실시간 감시, 알람 출력 등은 작업 특성상 실시간 조건을 만족하여야 한다. 실시간 특성을 만족하는 작업들은 항상 제어시스템으로부터 데이터를 받아서 업데이트를 하여야 한다. 반면에 레포트 출력이나 트렌드(historical trend) 분석과 같은 작업은 실시간 특성을 가질 필요가 없다. 그러므로 제어시스템과 직접적으로 연결될 필요가 없다. 본 논문에서는 분산제어시스템의 요구 사항인 실시간(real-time)적 특성과 신뢰성(reliability)을 만족하면서 유효성(availability)을 향상 시키는데 초점을 맞추었다.

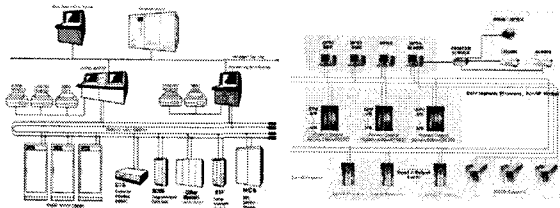
### 2. DCS(Distributed Control System)

고전적인 제어장치(control device)들은 독립적인(stand-alone) 구조로 자동 제어 방법을 제공하였다. 제어장치들 간의 연결 또한 동일한 회사의 장치들끼리만 가능하였다. 이후 마이크로프로세서 기술의 발달로 제어장치들은 점점 더 복잡해지고 다양한 기능을 제공하게 되었다. 이기종의 제어장치들은 통신을 통한 데이터 교환이 가능해졌다. 이러한 자동제

어장치들로 구성된 제어시스템을 PCS(Process Control System)이라고 한다[4].

PCS는 구조에 따라 중앙집중형과 분산형 시스템으로 구분할 수 있다. 초창기 PCS들은 중앙집중형 구조로 구성되었다. 중앙의 제어시스템에 모든 제어장치들이 물리적으로 연결되고, 데이터 수집과 프로세싱, 동작들을 모두 중앙제어시스템에서 제어하였다. 이후 마이크로프로세서 기술의 발달과 네트워크 기술의 발달로 PCS는 중앙집중형에서 분산형 구조로 바뀌었다. 분산형 구조의 자동제어시스템인 분산제어시스템은 지리적으로 분산되어 있는 제어시스템을 통하여 제어장치들을 제어하고 감시한다. 분산된 제어시스템들은 시리얼이나 이더넷을 통하여 연결된다. 또한 표준화된 프로토콜과 데이터 교환 기술을 사용한다.

2.1 분산제어시스템 구조



[그림 1] 분산제어시스템 구조도 (왼쪽 : 효성중공업의 HIPAC DCS, 오른쪽 : 삼성중공업의 SSAS-Pro DCS)

SCADA(데이터 수집 및 처리, 상태예측, 전력량처리, 통계분석 등의 기능), 수처리, 화학, 철강 등 여러 산업 분야에서 사용되고 있는 분산제어시스템들은 활용되는 산업분야에 따라 조금씩 다르지만 전체적인 시스템 구조는 비슷하다. 그림 1은 국내에서 수처리, 전력감시, 화학 분야에서 사용되고 있는 효성중공업과 삼성중공업의 분산제어시스템[5]의 구조도이다. 두 시스템이 사용하는 시스템의 명칭은 다르지만 전체적인 구조는 비슷한 것을 알 수 있다.

분산제어시스템은 크게 제어장치, 제어시스템, HMI 스테이션, 기타 외부장치들로 나눌 수 있다. 제어장치에는 펌프, 스위치, 수량 계측기와 같은 장치들이 있다. 제어장치는 제어시스템들과 하드웨어적으로 연결되어 일출력 데이터를 전달한다. 제어시스템은 제어장치들로부터 받은 데이터를 이더넷이나 시리얼 통신을 통하여 HMI 소프트웨어로 전달한다. HMI 스테이션은 사용자가 장치의 상태를 확인하고 동작을 시키는 컴퓨터이다. 기타 외부장치로는 레포트(report)나, 하드카피(hardcopy) 출력을 위한 프린트나 알람 수신을 위한 경보기 등이 있다. 그림 1에서 FCS, CIS, MCS, PCS등이 제어시스템에 속하고, HIPAC-6000P 또는 OPS-EWS등은 HMI 스테이션에 속한다.

2.2 분산제어시스템의 특징

분산제어시스템은 전력, 화학, 철강 등 산업분야 전반에 걸쳐 사용되고 있다. 분산제어시스템은 사람이 수동으로 제어하기 힘든 작업을 자동으로 제어하고 감시할 수 있도록 한다. 분산제어시스템은 다음과 같은 특징을 가진다.

- 실시간성(Real-Time): 분산제어시스템은 제어장치들을 실시간으로 감시하고 제어할 수 있어야 한다. 또한 사용자의 명령을 지연 없이 장치에 전달하여야 하고, 장치의 상

태를 취합하여 정해진 기준시간 안에 HMI 소프트웨어에 전달하여야 한다. 만약 그렇지 않다면 사용자는 잘못된 상태의 정보를 이용하여 장치들을 제어하게 될 것이다.

- 신뢰성(Reliability): 분산제어시스템은 항상 올바른 동작을 수행하여야 한다. 분산제어시스템은 대부분 사람이 시스템을 수동으로 제어하기 힘든 작업에 사용되기 때문에 오동작을 일으킬 경우, 사람들에게 피해를 가져올 수 있다. 이 때문에 분산제어시스템의 하드웨어와 소프트웨어는 설계단계에서부터 철저한 검증이 필요하다.
- 유효성(Availability): 분산제어시스템은 항상 동작 상태에 있어야 한다. 만약 하나의 시스템이 동작을 멈추더라도 장치들을 제어할 수 있도록 다수 개의 제어시스템을 배치한다. HMI 소프트웨어 또한 동일한 기능을 수행하는 소프트웨어를 2개에서 N개까지 여러 개 배치한다. DCS는 실시간성과 유효성을 동시에 만족하여야 한다. 이는 시스템이 빠르면서 오랫동안 고장 없이 동작하여야 한다는 것을 의미한다.
- 대용량 데이터 관리(Large Data Management): 분산제어시스템이 관리하는 데이터의 양은 제어장치들의 개수와 저장시간 간격에 영향을 받는다. 실시간 특성을 가지는 분산제어시스템에서 주고받는 데이터 크기는 제어장치들의 개수와 HMI 소프트웨어의 개수에 비례한다. 그리고 데이터를 수집하는 시간 간격에 반비례한다. 따라서 일반적으로 분산제어시스템은 대용량 데이터를 관리할 수 있어야 한다.
- 보안(Security): 분산제어시스템에서 보안은 매우 중요한 사항이다. 만약 외부로부터 침입을 받는다면 상당히 위험한 상황에 처하게 된다. 예를 들어, 수자원 제어시스템이 외부로부터 침입을 받아 염의 수분을 열게 된다면 강의 수위가 높아져 범람할 수도 있다. 또한 많은 분산제어시스템들이 인터넷을 통한 감시 및 제어를 지원하고 있기 때문에 보안은 더욱 중요하다.

2.3 관련연구

분산제어시스템에 관한 연구는 시스템 아키텍처 설계, 시스템 통합 및 신뢰도 향상 기법, 성능개선을 위한 데이터 전송지연 최소화에 대하여 활발히 이루어졌다.

Jena[6]는 시스템 유지 및 관리 비용 감소에 초점을 맞춘 분산제어 아키텍처인 Opto 22를 설계하여 상용화하였다. Bonsangue[7]는 Shared Data Space와 시스템 동작에 대한 의미(semantics)를 기반으로 한 분산제어시스템에 대한 소프트웨어 아키텍처를 정의하였다. Parunak[8]는 분산제어시스템의 긴급 상황관리와 처리를 위해 비선형 시스템 이론(nonlinear system theory)과 에이전트(agent)에 기반을 둔 아키텍처를 제안하였다. 그리고 Blecha[9]와 SZABÓ[10]는 산업용 로봇제어 시스템을 위한 분산제어시스템을 제안하였다.

시스템 통합 및 신뢰도 향상에 관한 연구로는 Craig[11]와 Diedrich[4]의 연구가 있다. Craig는 SCADA 시스템의 통합과 신뢰도 향상에 관한 방법을 제안하였고, Diedrich는 신뢰도 향상을 위해 ISO 10303 기반의 분산제어시스템 엔지니어링 기법을 제안하였다.

마지막으로 분산제어시스템의 성능 개선을 위한 데이터 전송 지연 최소화에 관한 연구는 분산제어시스템의 노드 증가에 따른 데이터 전송지연에 관하여 다룬다. J.K. YOON과

Tilbury[12]는 제한된 대역폭(bandwidth)을 사용하는 각 노드 간의 통신을 줄이는 알고리즘을 제안하였다. YOOK의 알고리즘에서는 노드의 실제 출력 값 대신 알고리즘에 의해 예측된 출력 값을 사용함으로써 네트워크를 통한 데이터 전송횟수를 줄인다.

이 밖에도 최근에는 Stanford 대학에서 리눅스를 기반으로 한 분산제어시스템 개발이 공개 소스 프로젝트로 진행하고 있다 [13]. 그리고 국내외에서 사용하는 대표적인 분산제어시스템은 표 1과 같다.

[표 1] 국내외의 분산제어시스템

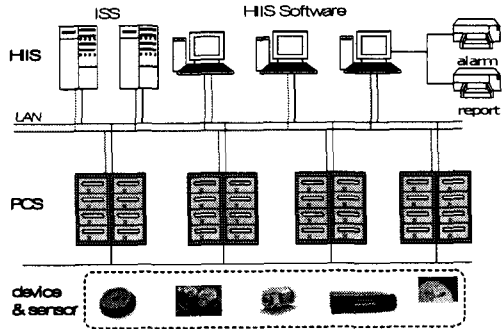
구분	DCS	사용 분야
국내	삼성중공업 SSAS-Pro, DNA	산업설비 및 공정 제어
	LG산전 MASTER P-3000, MASTER P-3000NT	수처리, 발전 화학 등
	효성 HIPAC-Series	수처리, 산업 설비 제어
국외	Honeywell Experion Platform	공장관리 및 제어
	HITACHI EX-5000 System	산업설비 및 계측 제어
	YOKOGAWA Centum CS 3000 R3, Centum CS	공정 제어
	SIXNET Truly DCS(SIXNET DCS) Open	공정 제어
	FOXBORO Total Distributed Control System (DCS) Solutions - The I/A Series	공정 제어, 산 업설비 제어

3. 분산제어 아키텍처 정의

본 장에서는 분산제어시스템이 수행해야 하는 작업(task)의 특성을 고려한 PNU-분산제어시스템을 설계한다. PNU-분산제어시스템은 실시간으로 장비들을 감시할 수 있어야 하고 지난 데이터들을 분석하여 향후 변동 상황에 대한 예측을 할 수 있어야 한다. 그리고 분산제어시스템이 갖추어야 하는 기본 조건들을 모두 만족하여야 한다. 또한 사용자들은 HMI 소프트웨어를 통하여 전체 시스템을 제어하기 때문에 HMI 소프트웨어의 안전성과 유효성을 향상시키는데 초점을 맞춘다.

3.1 하드웨어 아키텍처

그림 2는 PNU-DCS의 하드웨어 아키텍처이다. 하드웨어 아키텍처는 다른 일반적인 분산제어시스템과 비슷하다. PNU-DCS는 크게 PCS(Process Control Station), HIIIS(Human Intelligence Interaction Station)으로 나뉜다. PCS는 펌프, 스위치, 등 각종 센서들과 연결되어 실시간으로 장비의 상태 정보를 전달받는다.



[그림 2] 하드웨어 아키텍처

PCS의 주요 특징은 다음과 같다.

- 지리적으로 분산된 곳에 다수 개를 배치할 수 있다
- 펌프, 스위치, 온도, 압력, 유량 등의 장치 및 센서들과 물리적으로 연결된다.
- 이기종의 제어장치들과 호환되어 동작할 수 있다.
- IOCS 및 HIIIS와 LAN으로 연결된다.
- HIIIS로부터 제어로직을 다운로드 받는다. 제어로직을 다운로드 받는 작업은 IOCS와는 독립적으로 수행되며, 다운로드가 끝나면 즉시 반영된다.

HIIIS는 분산제어시스템에서 사용자 인터페이스를 담당한다. 사용자는 HIIIS 소프트웨어를 사용하여 전체 시스템을 제어한다. HIIIS는 제어로직 다운로드 및 감시를 위한 소프트웨어, 기타 외부 출력장치들을 포함한다. HIIIS는 다음의 특징을 가진다.

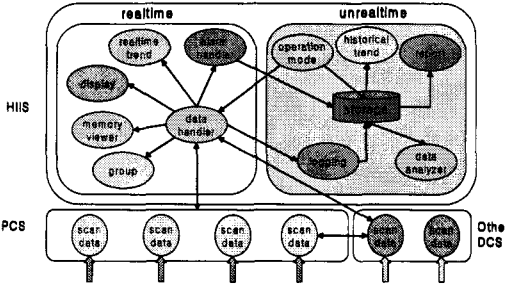
- PCS 및 IOCS와 LAN을 통하여 연결된다.
- 프린터, 휴대폰, 알람과 같은 외부 출력 장치와 연결된다.
- 전체 분산제어시스템을 제어감시 한다.
- 실시간으로 데이터를 저장한다.

본 절에서는 하드웨어 아키텍처를 정의하였다. 다음 절에는 HIIIS 소프트웨어가 수행하는 작업(task)과 HIIIS 소프트웨어에 대한 아키텍처를 정의할 것이다. 본 논문에서 제안하는 PNU-분산제어시스템은 수행하는 작업의 특성에 따라 제어시스템과 HIIIS 소프트웨어가 동적으로 연결된다. 이는 분산제어시스템의 실시간성과 신뢰성을 유지하면서 유효성을 증가시킨다.

3.2 작업(Task) 아키텍처

분산제어시스템의 작업은 실시간성을 지켜야 하는 작업과 그렇지 않은 작업으로 나눌 수 있다. 장치들의 감시는 실시간으로 이루어져야 한다. 그러나 저장장치에 기록된 데이터를 분석하는 작업은 실시간으로 수행될 필요는 없다. 그림 3은 분산제어시스템이 수행하는 작업(task)을 실시간성을 기준으로 구분하였다.

그림 3에서 PCS는 장치들로부터 데이터 얻는다. 그리고 이더넷을 통하여 HIIIS로 데이터를 전달한다. HIIIS가 수행하는 작업들은 실시간적인 것과 그렇지 않은 것으로 구분할 수 있다.



[그림 3] 분산제어시스템의 작업(Task) 아키텍처

실시간적인 작업에는 알람 핸들링(alarm handler), 실시간 트렌드 출력(realtime trend), 화면 갱신(display refresh), 실시간 데이터 검사(memory viewer), 제어장치 그룹 검사(group), 메모리 감시(memory viewer) 등이 있다. 데이터 핸들러는 제어 시스템으로부터 실시간으로 데이터를 전달받는다. 그리고 저장 장치에 기록(logging)하는 작업을 수행한다.

알람 핸들링은 위급한 상황에 대하여 경보를 울리고 저장장치에 기록을 한다. 실시간 트렌드는 데이터의 변화를 그래프 형태로 사용자에게 알려준다. 화면갱신은 그림을 통하여 검사할 때 매번 다시 데이터를 갱신하여야 한다. 그룹(group) 작업은 장치들을 그룹으로 묶어서 감시를 수행한다. 그리고 메모리 감시는 디버깅을 위해 메모리에 저장되는 데이터를 감시한다.

실시간 특성이 필요하지 않은 작업으로는 레포트(report) 출력, 히스토리컬 트렌드(historical trend), 데이터 분석(data analyzer), 데이터 로깅(logging) 등이 있다. 레포트는 프린터와 같은 외부 출력장치로 데이터를 보낸다. 히스토리컬 트렌드는 지난 데이터들의 변화를 그래프 형태로 출력한다. 데이터 분석은 상관관계(correlation)나 회귀분석(regression analysis) 등의 데이터 분석 기법을 이용하여 데이터를 분석하고 변화를 예측한다.

본 논문의 분산제어시스템은 작업의 특성에 따라 동적으로 PCS와 HHS가 연결된다. 다음 절에서는 HHS 소프트웨어의 작업 특성을 고려한 제어감시 방법을 제안한다.

4. 작업 특성을 고려한 감시

일반적인 분산제어시스템은 안전성을 높이기 위하여 HMI 컴퓨터를 N개 배치한다. 이 방법은 안전성을 높이지만 동일한 동작의 반복적 수행으로 인하여 분산제어시스템에 부담을 준다. 제어시스템과 HMI 소프트웨어 사이에 네트워크 데이터 전송량을 증가시키고, 빈번하게 디스크 동작(disk operation)을 수행한다. 이와 같은 구조에서는 하나의 HMI 소프트웨어가 통신, 데이터 저장, 화면갱신 등 모든 작업을 수행해야 하기 때문에 HMI 고장가능성을 높인다.

4.1 실시간 특성을 고려한 분산처리시스템

본 절의 감시 알고리즘은 작업(task)의 실시간적 특성을 이용하여 분산제어시스템의 부담을 줄인다. 감시 알고리즘은 작업 그래프(task graph)를 사용하여 정의된다.

정의1. 작업그래프 G와 G'는 다음과 같이 정의된다.

$$G = (PCS, HS, E)$$

$$G' = (ISS, HHS, E')$$

위 정의에서 PCS는 Process Control Station의 집합이다. HS는 ISS와 HHS 컴퓨터의 집합이다. E와 E'는 그래프의 두 노드를 연결한다. DCS 구성은 다음과 같이 정의한다.

$$PCS = \{p_1, p_2, \dots, p_m\}$$

$$ISS = \{s_1, s_2, \dots, s_k\}, HHS = \{h_1, h_2, \dots, h_l\}$$

$$HS = ISS \cup HHS$$

$$E = \{(e_1, e_2) | e_1 \in PCS, e_2 \in HS\}$$

$$E' = \{(e_3, e_4) | e_3 \in ISS, e_4 \in HHS\}$$

정의2. 분산제어시스템에서는 다음 함수들을 정의한다. Total<sub>PCS</sub>()는 |PCS|을 반환한다. Total<sub>HS</sub>()는 |HS|을 반환한다. ScanData<sub>i</sub>(PCS)는 PCS에 연결된 장치들의 데이터를 읽는다. Trans<sub>i</sub>(E, E')는 두 노드 사이에 데이터를 전송한다. 아래첨자 t는 현재시간을 가정하며 생략가능하다. 예를 들어, Trans({p<sub>1</sub>, h<sub>1</sub>})는 첫 번째 PCS에서 첫 번째 HHS로 데이터를 전송한다. Logging(ISS)는 저장장치(ISS)에 데이터를 기록한다.

TaskNo는 HS 소프트웨어에서 수행하는 작업에 대한 상수 값이다. CurrentTask()는 사용자가 현재 감시하고 있는 HHS 소프트웨어의 TaskNo를 반환한다. IsReal(TaskNo)은 true 또는 false를 반환한다. TaskNo의 작업이 실시간 특성을 가지는 작업이면 true를 반환하고 그렇지 않으면 false를 반환한다. ProcessingTask(TaskNo)는 TaskNo에 대한 작업을 수행한다.

본 절의 알고리즘은 DCS가 수행하는 작업을 실시간적 특성에 따라 구분한다. 그리고 실시간 특성을 이용하여 제어시스템과 HHS 소프트웨어는 동적으로 연결된다. 실시간 트렌드나 알람 핸들링 같은 실시간 특성을 가지는 작업의 수행은 PCS로부터 데이터를 전송받는다. 그리고 히스토리컬 트렌드나 레포트와 같은 작업의 수행은 ISS로부터 데이터를 받는다. 분산제어시스템의 수행 알고리즘은 다음과 같다

[표 2] 실시간 특성을 고려한 분산제어시스템 수행 알고리즘

```

Input : Task Graph G = (PCS, HS, E)
numberOfPCS = the number of Process Control Stations;
numberOfHS = the number of ISS and HHS stations;

Process :
1. numberOfPCS := TotalPCS();
2. numberOfHS := TotalHS();
3. ScanData(CS);
4. for i := 0 to i < numberOfPCS; do
5.   for j := 0 to j < |ISS|; do
6.     Trans({pi, sj}); Logging(sj);
7.   end for
8. for j := 0 to j < numberOfHS - |ISS|; do
9.   var TaskNo = CurrentTask();
10.  if IsReal(TaskNo) then
11.    Trans({pi, hj}); ProcessingTask(TaskNo);
12.  else
    
```

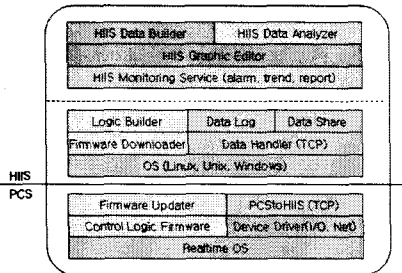
```

13. Trans( $\{s_{(n < ISS)}, h_j\}$ );
14. ProcessingTask(TaskNo);
15. end if
16. end for
17. end for
    
```

표 2의 알고리즘에서 4~6줄은 PCS에서 보내는 데이터를 ISS에 기록한다. 그리고 8~14줄에서 HHS 소프트웨어는 작업 특성에 따라 PCS와 ISS에 동적으로 연결된다. 실시간 감시는 PCS와 연결되고 실시간 감시가 아닌 작업은 ISS와 연결된다. 위의 알고리즘에서는 실시간으로 데이터를 디스크에 저장하는 작업을 ISS에서만 수행한다. HHS 소프트웨어는 디스크 동작을 하지 않는다. 데이터를 디스크에 쓰는 작업은 수행시간에서 많은 부분을 차지한다. 그리고 HHS 소프트웨어가 실시간 감시를 하지 않을 경우에 PCS로부터 데이터를 전달받지 않기 때문에 전체 시스템에서 전송되는 데이터의 양을 줄일 수 있다. 디스크 쓰기 동작과 데이터 전송의 감소는 분산제어시스템의 성능을 향상시킨다.

4.2 소프트웨어 아키텍처

분산제어시스템의 소프트웨어는 크게 제어시스템을 위한 소프트웨어와 HMI 소프트웨어 두 가지로 구분할 수 있다. 제어시스템 소프트웨어는 각종 I/O(Input/Output) 장치를 위한 디바이스 드라이버와 장치제어를 위한 펌웨어(firmware) 프로그램이 있다. 그리고 HMI 소프트웨어는 펌웨어 작성 프로그램, GUI를 위한 그래픽 에디터, 실시간 감시 프로그램 등이 필요하다. 그림 4는 분산제어시스템의 소프트웨어 아키텍처를 정의한 것이다. PCS에는 실시간 운영체제를 사용한다. 운영체제에는 각종 I/O 장치와 네트워크 장치를 위한 디바이스 드라이버가 설치된다. 그리고 장치제어를 위한 펌웨어 프로그램(Control Logic Firmware)이 탑재된다. PCstoHHS는 이더넷을 통하여 수집된 데이터를 HHS에 전송한다. PCstoHHS는 HHS 소프트웨어가 실시간 감시를 수행할 때 연결된다.



[그림 4] 소프트웨어 아키텍처

HHS 소프트웨어는 두 부분으로 나뉜다. 한 부분은 PCS 또는 ISS와 연결되는 프로그램들이고 다른 프로그램들은 감시를 위한 HHS 소프트웨어이다. PCS와 연결되는 프로그램들은 펌웨어 빌더와 다운로더, 그리고 작업에 따라 동적으로 PCS 또는 ISS와 연결되어 데이터를 얻어오는 데이터 핸들러(Data Handler)가 있다. 데이터 핸들러는 데이터를 기록하는 프로그램(Data Log)과 HHS 모니터링 프로그램에 데이터를 전달하는 프로그램(Data Share)과 연결된다.

장치들의 감시와 제어를 위한 소프트웨어에는 데이터 빌더

(HHS Data Builder), 데이터 분석기(HHS Data Analyzer), 그래픽 에디터(HHS Graphic Editor), 실시간 감시 프로그램(HHS Monitoring Service)이 속한다. 데이터 빌더와 그래픽 에디터는 감시를 위하여 화면을 만들고 감시를 위한 태그(tag)들을 설정한다. 그리고 데이터 분석기는 지난 데이터들을 통계적 기법을 이용하여 분석하고 데이터 변화를 예측한다. 실시간 감시 프로그램은 전체 시스템을 감시하고 제어한다. 최종적으로 사용자는 실시간 감시 프로그램인 HHS Monitoring Service 프로그램을 이용하여 시스템을 제어한다.

4.3 성능 평가

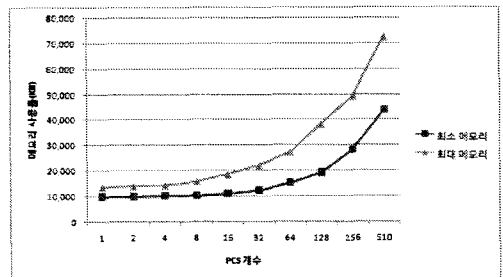
본 논문에서 제안한 실시간 특성을 고려한 분산제어시스템 수행 알고리즘의 성능 평가를 위해서 Java를 사용하여 실시간 특성을 반영한 경우와 반영하지 않은 경우에 HMI 메모리 사용량을 비교하였다. HMI는 제어-감시를 위해서 다수 개 설치할 수 있고, PCS의 경우는 수십에서 수백 개를 설치할 수 있다. 이를 위해 PCS는 스레드(thread)로 동작하게끔 하였고, 아래 그림 5와 같이 한 프로세스에서 생성할 수 있는 최대 스레드 수인 510개까지로 하였다.

실험에서 가장한 상황은 메모리 높이를 측정하여 일정량 이상이 되면 경고 메시지를 HMI로 보내고 HMI를 통하여 수위를 조정하여 물 높이를 조절하게 하였다. 그리고 측정값들은 실시간 특성을 반영한 경우 ISS에 1초 단위로 저장시키도록 하였고, 필요시 HMI는 ISS에 접속하여 로그정보를 확인할 수 있도록 하였다. 실시간 특성을 고려하지 않은 경우는 HMI에서 측정값 확인부터 로그정보를 처리하여야 한다. 이를 위해 HMI, PCS, ISS를 별도의 PC에서 수행시켰고 각 PC의 사양은 다음과 같다.

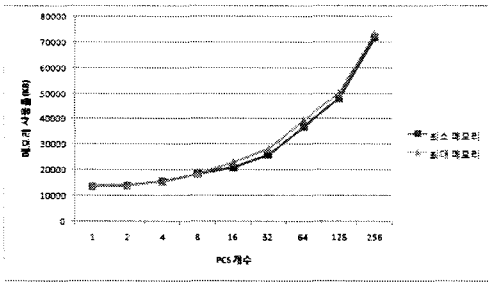
[표 3] 실험 환경

구성	사양
HMI	Pentium D-2.8GHz, 1G, JDK-1.5, WindowsXP
PCS	Pentium M-1.4GHz, 1G, JDK-1.5, Linux
ISS	Pentium4 3.0GHz, 2G, JDK-1.5, Mysql-4.1.11, Linux

실험결과는 실시간 특성을 반영하여 PCS로부터 받는 측정값을 감시하는 기능과 ISS에 동적으로 연결되는 경우 HMI의 메모리 사용량은 그림 5와 같고 실시간 특성을 반영하지 않은 경우는 그림 6과 같이 나왔다. 두 그림에서 보이는 것처럼 실시간 특성을 고려한 경우 실시간 특성을 고려하지 않은 경우에 비해서 동적 연결을 통해 최소 메모리 사용량이 적게는 28%에서 많게는 50% 정도 메모리를 적게 사용하였다.



[그림 5] PCS 개수에 따른 HMI의 메모리 사용량(실시간 특성 반영)



[그림 6] PCS 개수에 따른 HMI의 메모리 사용률(실시간 특성 미반영)

본 실험에서는 PCS가 전송해주는 측정값을 저장·감시하고 일정 값 범위를 벗어나면 이를 알려주는 가장 기본적인 수행만을 고려하였고, 스래드의 특성상 메모리 사용률 측정이 어려운 점 등은 향후 개선해야 할 점이다.

### 5. 결론 및 향후연구

분산제어시스템은 실시간으로 제어와 감시가 수행되어야 한다. 그리고 시스템의 신뢰성과 유효성이 높아야 한다. 또한 대용량 데이터를 효율적으로 처리해야 한다. 사용자는 HMI 소프트웨어를 통해 전체 분산제어시스템을 제어 및 감시하기 때문에 안전성을 위하여 다수 개의 동작을 HMI 소프트웨어를 배치한다. 하지만 동일한 작업의 수행은 분산제어시스템에 부담을 준다. 또한 HMI 소프트웨어는 항상 동작하면서 통신, 저장, 화면 갱신, 데이터 분석 등의 모든 작업을 수행하기 때문에 고장 가능성도 높다.

본 논문에서는 분산제어시스템의 하드웨어와 작업(task)에 대한 아키텍처 및 실시간 특성을 고려한 알고리즘을 제안하였다. 제안한 하드웨어 아키텍처가 일반 분산제어시스템과 비슷한 것은 장치들과 직접 연결되기 때문에 비슷하다. 본 논문의 작업 아키텍처는 실시간성을 기준으로 나뉜다. HIIIS 소프트웨어는 수행되는 작업의 실시간적 특성에 따라 동적으로 PCS와 ISS에 연결된다. 실시간 작업을 수행할 경우, PCS로부터 실시간 데이터를 받아온다. 반대로 실시간으로 수행되는 작업이 아닌 경우, HIIIS 소프트웨어는 ISS에서 데이터를 받아온다. 작업의 특성에 따른 동적연결은 HIIIS 소프트웨어의 디스크 쓰기 동작과 데이터 전송량을 줄인다. HIIIS 소프트웨어의 작업에 따른 동적연결은 분산제어시스템의 작업량을 줄임으로서 유효성을 증가시킨다. 그리고 실험을 통해 작업에 실시간 특성을 고려한 경우가 그렇지 않은 경우에 비해 메모리를 28~50% 더 적게 사용하는 것을 확인할 수 있었다.

향후 연구로는 본 논문에서 수행한 실험을 기반으로 보다 다양한 환경을 고려하여 개선된 실험을 지속적으로 수행하고 분산제어시스템에 분산 컴퓨팅 환경인 그리드 컴퓨팅(Grid Computing) 기술의 적용에 관하여 연구할 것이다. 그리드 컴퓨팅 기술은 작업을 분산시켜 시스템의 성능을 향상시킬 수 있다. 이는 분산제어시스템의 성능을 향상시킬 수 있을 것이다.

### 참고문헌

[1] Wikipedia, Distributed Control System, "[http://en.wikipedia.org/wiki/Distributed\\_Control\\_System](http://en.wikipedia.org/wiki/Distributed_Control_System)"  
 [2] Honeywell, "<http://www.honeywell.co.kr>"  
 [3] YOKOGAWA, "<http://www.yokogawa.com>"  
 [4] Christian Diedrich, Rent5 Simon, Matthias Redl,

"Engineering of Distributed Control Systems", Industrial Electronics, 2000. ISIE 2000. Proceedings of the 2000 IEEE International Symposium on, pp.661~665, vol.2, 2000.

[5] 정태경, "SSAS-Pro", 計裝技術 제12권 제3호 통권122호 (2004. 3) pp.130-135

[6] opto22, "<http://www.opto22.com>"

[7] Marcello M. Bonsangue, Joost N. Kok, "Software Architecture Distributed Control Systems Its Transition System Semantics", Proc. ACM/SIGAPP Symp. on Applied Computing (SAC 98), pp.159~168, 1998.

[8] H. Van Dyke Parunak, Raymond S. VanderBok, "Managing Emergent Behavior in Distributed Control Systems", H. V. D. Parunak and R. S. VanderBok. Managing Emergent Behavior in Distributed Control Systems. In Proceedings of ISA Tech '97, Instrument Society of America, 1997.

[9] Radim Blecha, Zdeněk Bradáč, "Distributed Control System for Robotic Manipulators", Industrial Technology, 2003 IEEE International Conference on, pp.169~72, Vol.1, 10-12 Dec. 2003.

[10] Štefan SZABÓ, Vladimír Opluštil, "Distributed CAN Based Control System for Robotic and Airborne Application", Control, Automation, Robotics and Vision, 2002. ICARCV 2002. 7th International Conference on Volume 3, 2-5 Dec. 2002 Page(s):1233 - 1238 vol.3

[11] Dean Craig, Craig Befus, "Implementation of a distributed control system for electric distribution circuit reconfiguration", Power Engineering Society General Meeting, 2005. IEEE, pp.2436~2441, Vol.3, 12~16, June, 2005.

[12] Yook, J.K., Tilbury, D.M., Soparkar, N.R., 2000, "Performance Evaluation of Distributed Control Systems with Reduced Communication," to appear in Proceedings of the Allerton Conference on Communication, Control, and Computing.

[13] BLU-ICE/DCS, Open Source Project, "<http://smb.slac.stanford.edu/research/developments/blu-ice/>"