

화력 및 원자력 발전소 시뮬레이터 I/O 인터페이스 시스템 구축에 관한 연구

변승현, 장태인
(한전 전력연구원)

An Implementation of I/O Interface System for T/P and N/P Plant Simulator

S.H.Byun, T.I.Jang
(KEPCO, KEPRI)

요 약

발전소의 안정적 운영을 위한 발전소 운전원의 훈련에는 기존 발전소 주제어실의 각종 판넬과 계기들을 기계적으로 동일하게 만들고 정확한 수학적 모델로 발전소 시스템을 모사한 전 범위 시뮬레이터가 주로 이용된다. 전범위 시뮬레이터는 실제 현장을 모사하는 시뮬레이션 컴퓨터와 주제어실의 판넬 계기들을 연결시켜 주는 I/O 인터페이스 시스템을 필요로 한다. 기 설치 운용되고 있는 선행 호기 시뮬레이터를 보면 주 공급업체는 모델링 기술을 가지고 있는 업체가 많고, I/O 인터페이스 시스템은 I/O 모듈과 네트워크 프로토콜까지 협력업체나 자사의 독자 모델을 취하고 있다. 그러나 I/O 모듈의 제작사가 대부분 외국의 특정 업체여서 예비품의 입수가 용이하지 않아 유지보수가 어렵고, 많은 비용을 지불해야 하는 단점을 가지고 있다. 따라서, 유지보수 용이성과 경제적 측면에서 특정 벤더 규격에 종속되지 않고 일반적인 표준을 따르는 off-the-shelf 제품을 이용하는 발전소 시뮬레이터 I/O 인터페이스 시스템의 구축이 요구된다. 본 논문에서는 화력 발전소와 원자력 발전소를 대상으로 시뮬레이터 I/O 인터페이스 시스템의 요구사항을 소개하고, 네트워크에서 시스템 아키텍처까지 개방형 인터페이스 방식의 상용 제품을 채용하는 PC기반 시뮬레이터 I/O 인터페이스 시스템 구축방안을 제시하고, 성능 예측을 통해 제안한 시스템 구축방안의 효용성을 보인다.

1. 서 론

한 나라의 기간산업으로서 중요한 전력산업은 사회 발전과 더불어 증가되는 전력 소비량과 양질의 전력 요구에 의해 지속적으로 발전되고 있다. 양질의 전력을 안정적으로 공급하기 위해서는 발전소를 고장없이 효율적으로 운용하는 것이 중요하며, 이에 대해 제어실 요원의 유사시 상황 대처 능력 배양과 제어 시스템의 최적관리 중요성이 부각되고 있다[1].

발전소 사고 사례를 보면 많은 경우에 운전원의 부적절한 사고 대처에 기인하고 있다. 또한, 발전소 시스템의 복잡성으로 인해 시스템의 특성을 충분히 숙지하고 운전할 때만이 만족스러운 운전 결과를 얻을 수 있다. 따라서 양질의 전력의 안정적 공급을 위해서는 발전소 운전원의 훈련이 필요하다[2].

발전소 운전원의 훈련에는 기존 발전소 주제어실의

각종 판넬과 계기들을 기계적으로 동일하게 만들고 정확한 수학적 모델로 발전소 시스템을 모사한 전 범위 시뮬레이터가 주로 이용된다. 발전소 운전원 훈련용 시뮬레이터는 표준 화력 발전소와 모든 원자력 발전소를 대상으로 이미 설치되어 운용 중이거나 설치 계획 중이다. 전 범위 시뮬레이터는 실제 현장을 모사하는 시뮬레이션 컴퓨터와 주제어실의 판넬 계기들을 연결시켜 주는 I/O 인터페이스 시스템을 필요로 한다. 기 설치 운용되고 있는 선행 호기 시뮬레이터를 살펴보면 주 공급업체는 대부분 모델링 기술을 가지고 있는 업체가 많고, I/O 인터페이스 시스템은 I/O 모듈과 네트워크 프로토콜까지 협력업체나 자사의 독자 모델을 취하고 있다. 그러나 I/O 모듈의 제작사가 대부분 외국의 특정 업체여서 예비품의 입수가 용이하지 않아 유지보수가 어렵고, 많은 비용을 지불해야 하는 단점을 가지고 있다. 따라서, 유지보수 용이성과 경제적 측면에서 특정 벤더 규격에 종

속되지 않고 일반적인 표준을 따르는 상용 제품을 이용하는 발전소 시뮬레이터 I/O 인터페이스 시스템의 구축이 요구된다.

본 논문에서는 화력 발전소와 원자력 발전소를 대상으로 시뮬레이터 I/O 인터페이스 시스템의 요구사항을 소개하고, 네트워크에서 시스템 아키텍처까지 개방형 인터페이스 방식의 상용 제품을 채용하는 PC기반 시뮬레이터 I/O 인터페이스 시스템 구축방안을 제시하며, 예측성능으로 제시한 시스템 구축방안의 효용성을 보이고자 한다.

2. 발전소 시뮬레이터의 개요 및 요구사항

2.1 발전소 시뮬레이터의 개요

보통의 발전소 시뮬레이터는 발전소 시스템을 모사하는 현장 모델과 제어를 내장하는 시뮬레이션 컴퓨터, 제어 프로그램 다운로드, 각종 파라미터 수정 등 엔지니어링 작업을 수행할 수 있는 엔지니어링 워크스테이션, I/O 인터페이스 시스템, 강사가 훈련을 위해 시뮬레이터의 상태나 조건들을 조작하여 시연할 수 있는 컴퓨터 시스템인 강사 조작반, 발전소 중앙 제어실을 그대로 모사하기 위한 판넬로서 실제 발전소 중앙 제어실에 있는 기구물들을 그대로 모사하기 위해 실제 발전소 중앙 제어실에서 채용한 I&C 계기와 콘솔의 외양을 그대로 채용하여 실장하고 있는 하드판넬로 구성된다.

발전소 시뮬레이터에서 I/O 인터페이스 시스템은 현장의 상태들을 중앙 제어실에 알려주는 각종 판넬의 I&C 계기, 콘솔 데이터를 시뮬레이션 컴퓨터와 인터페이스 시켜주는 시스템으로 하드 판넬의 계기와 콘솔을 직접 처리하기 위한 I/O 카드와 시뮬레이션 컴퓨터와의 인터페이스를 위한 네트워크 카드, I/O 데이터 처리와 네트워크 처리를 위한 CPU 카드 등으로 구성되어진다.

2.2 화력 및 원자력 시뮬레이터에서의 I/O 포인트 용량

I/O 인터페이스 시스템은 하드 판넬의 모든 계기와 콘솔 데이터를 수용해야만 한다. 본 논문에서 대상으로 삼은 화력 발전소 시뮬레이터의 중앙 제어실의 판넬 위치도는 그림 1과 같고, 각 판넬에서 채용하는 판넬 계기와 콘솔 포인트 수를 I/O 신호유형별로 나열하면 표 1과 같다[1].

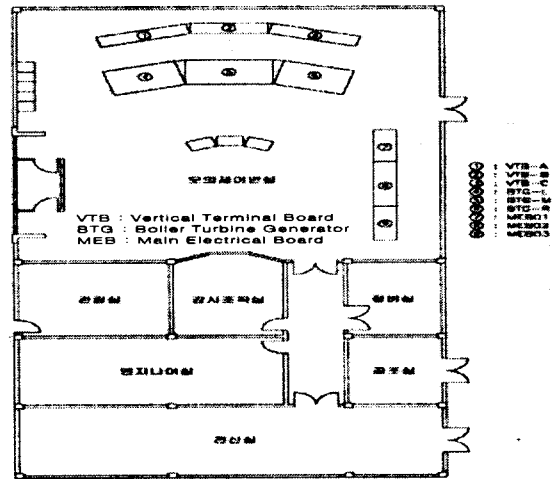


그림 1. 화력 발전소 시뮬레이터 제어실 배치도

원자력 발전소 시뮬레이터의 중앙 제어실의 판넬 위치도는 그림 2와 같고, 각 판넬에서 수용하는 I/O 포인트수는 표 2과 같다[3]. 표 2의 데이터는 참고문헌 [3]에 나와 있는 데이터를 가공하여 약간 변형한 데이터이다.

표 1, 표 2의 I/O 신호유형에서 DI는 디지털 입력을 의미하며, DO는 디지털 출력을 의미한다. AI는 아날로그 입력을 나타내고, AO는 아날로그 출력을 나타내며, RO는 릴레이 출력을 나타낸다.

표 1. 화력 발전소 시뮬레이터에서의 I/O 포인트 수

판넬	신호유형				합계
	DI	DO	RO	AO	
BTG-L	50	154	0	31	235
BTG-M	190	399	0	108	697
BTG-R	141	239	0	44	424
MEB01	98	0	290	31	419
MEB02	183	0	321	102	606
MEB03	46	0	168	53	267
VTB-A	230	283	2	93	608
VTB-B	0	4	0	34	38
VTB-C	35	49	44	64	192
	973	1128	825	560	3486

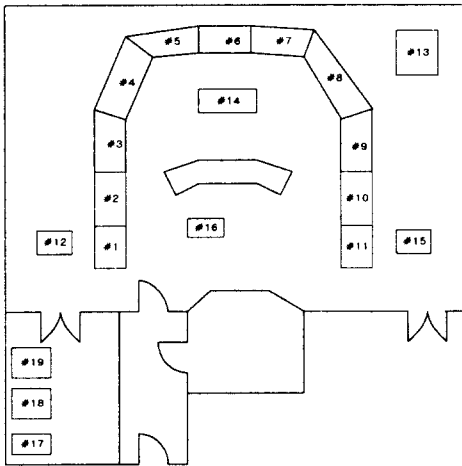


그림 2. 원자력 발전소 시물레이터 제어실 배치도

표 2. 원자력 발전소 시물레이터에서의 I/O 포인트 수

신호유형 판넬	AI	AO	DI	DO	RO	합계
#1	7	63	567	544	503	1684
#2	7	102	312	362	326	1109
#3	7	156	500	477	441	1581
#4	24	120	389	361	334	1228
#5	7	113	118	203	161	602
#6	14	136	180	707	600	1637
#7	7	69	344	342	316	1078
#8	7	101	363	456	369	1296
#9	7	5	397	500	408	1317
#10	7	128	274	568	270	1247
#11	7	52	132	295	210	696
#12	5	18	9	122	113	267
#13	28	19	700	763	600	2110
#14	0	12	0	14	14	40
#15	3	13	69	118	72	275
#16	0	13	20	10	10	53
#17	0	13	28	27	26	94
#18	7	25	150	121	112	415
#19	7	25	147	118	110	407
계	151	1183	4699	6108	4995	17136

2.3 시물레이터에서의 I/O 데이터 전송 주기 사양

보통의 대용량 플랜트의 제어 시스템의 1초에 4~5번의 스캐닝을 요구하는 반면에 시물레이터의 경우에는 1초에 8~12번의 스캐닝을 요구한다. 본 논문에서 구축하고자 하는 발전소 시물레이터 I/O 인터페이스 시스템의 요구사항은 표 1, 표 2에 나와 있는 I/O 점수를 수용하면서, 시물레이션 컴퓨터가 모든 I/O 포인트를 1초에 12번 스캐닝 할 수 있어야 한다.

2.4 I/O 인터페이스 시스템 구성상의 요구사항

I/O 인터페이스 시스템의 모든 구성 요소는 네트워크에서 시스템 아키텍처까지 표준 인터페이스 방식이나 개방형 방식을 채택하는 상용 제품을 사용하여 유지 보수 및 시스템 확장이 용이하도록 해야한다.

3. I/O 인터페이스 시스템의 구현

3.1 분산 처리를 위한 판넬별 I/O 포인트의 분산

I/O 인터페이스 시스템이 표 1, 표 2의 하드 판넬의 I/O 포인트 수용하기 위해서는 I/O 신호 종류별로 I/O 카드를 채용해야 하고, 시물레이션 컴퓨터와의 인터페이스를 위해서는 네트워크 기능을 가지고 있어야 한다. 표 1, 표 2를 보면, 시물레이터에서 요구하는 I/O 포인트 수가 3,000 포인트 이상의 대용량이므로, 이 I/O 포인트 들을 I/O 인터페이스 시스템에서 수용하기 위해서는 각 판넬에서 수용하는 I/O 포인트들을 위치와 수량을 고려해 분산한 후 네트워크 관점에서의 노드의 갯수를 결정하고, 각 노드를 네트워크로 시물레이션 컴퓨터와 연결함으로써 I/O 인터페이스 시스템을 구현할 수 있다. 본 논문에서는 분산 처리를 위해 나눈 노드를 입출력 처리 노드라고 정의하겠다.

3.1.1 화력 발전소 시물레이터의 I/O 포인트의 분산

그림 1의 대상 시스템의 제어실 배치도에서 각 판넬의 위치와 각 판넬별 I/O 포인트 수를 고려하여 분산하면 표3과 같이 노드를 나눌 수 있다.

표 3. 화력 발전소 시물레이터 입출력처리 노드별 I/O 포인트 수

신호유형 노드	DI	DO	RO	AO	합계	판넬
노드 1	240	553	0	139	932	BTG-I, BTG-M
노드 2	239	239	290	75	843	BTG-R, MEB01
노드 3	229	0	489	155	873	MER02, MEB03
노드 4	265	336	46	191	838	VTB
계	973	1128	825	560	3486	

3.1.2 원자력 발전소 시물레이터의 I/O 포인트의 분산

그림 2의 대상 시스템의 제어실 배치도에서 각 판넬의 위치와 각 판넬별 I/O 포인트 수를 고려하여 분산하면 표4와 같이 노드를 나눌 수 있다.

표 4. 원자력 발전소 시뮬레이터 입출력처리 노드별 I/O 포인트 수

노드	AI	AO	DI	DO	RO	합계	판넬
노드 1	14	63	325	266	248	916	#17,18,19
노드 2	12	81	576	666	616	1951	#1,12
노드 3	7	115	332	372	336	1162	#2,16
노드 4	7	156	500	477	441	1581	#3
노드 5	31	233	507	564	495	1830	#4,5
노드 6	14	148	180	721	614	1677	#6,14
노드 7	28	19	700	763	600	2110	#13
노드 8	17	193	475	981	552	2218	#10,11,15
노드 9	14	170	707	798	685	2374	#7,8
노드 10	7	5	397	500	408	1317	#9
계	151	1183	4699	6108	4995	17136	

3.2 I/O 인터페이스 시스템의 구성

본 논문에서 구성하고자 하는 I/O 인터페이스 시스템의 구성도를 도시하면 그림 3과 같다. 화력 발전소의 경우에 입출력 처리 노드의 갯수 N은 4가 되며, 원자력 발전소의 경우 N은 10이 된다.

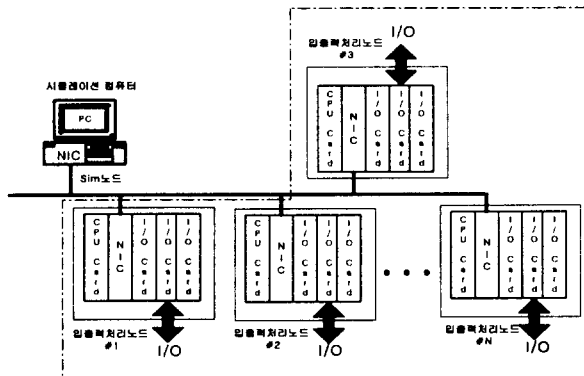


그림 3. I/O 인터페이스 시스템 구성도

3.2.1 입출력 처리 노드

그림 1, 그림 2의 각 판넬의 I&C 계기와 콘솔의 상태를 읽어서, 시뮬레이션 컴퓨터에 전달하고, 시뮬레이션 컴퓨터에서의 모델링 결과로 나오는 각 프로세스 값들의 지시치나 로직에 의한 알람창의 알람 여부 등 상태치 등을 출력 지시하게 하는 노드로, CPU 카드와 네트워크 카드, I/O 카드 등으로 구성되어진다.

3.2.2 Sim 노드

입출력 처리 노드와 시뮬레이션 컴퓨터 간의 통신을 중개하는 노드로 시뮬레이션 컴퓨터안에 탑재된 네트워크 카드로 정의한다.

3.3 입출력 처리 노드의 구성

3.3.1 입출력 처리 노드의 시스템 아키텍처 선정

표 3, 표 4의 각 입출력 처리 노드에서 수용해야 하는 I/O 포인트 수를 보면 대략 1,000 ~ 2,400 포인트 정도 됨을 볼 수 있다. 각 노드에서 1,000에서 2,400 포인트 정도의 대용량 입출력 포인트를 처리하기 위해서는 각 노드의 시스템 아키텍처의 선택이 중요하다. 즉 어떠한 버스를 채용하느냐 하는 것이다. 본 논문에서는 I/O 카드의 표준 인터페이스로 데이터 수집과 제어 분야에서 가장 널리 이용되어지고 있으며, 모든 PC에서 PCI(Peripheral Component Interconnect) 버스와 함께 채용되고 있는 ISA(Industry Standard Architecture) 버스를 채용한다[4]. 원자력 시뮬레이터의 경우인 표 4에서 노드당 처리 데이터 양을 계산하면 노드 5와 노드 8이 744바이트로 가장 크다. 이 데이터를 8비트 표준 ISA 버스 I/O 디바이스를 이용하여 ISA 버스 상에서 데이터를 처리하는데 걸리는 시간을 계산하면, 0.58[ms] 정도의 시간이 소요된다[4].

3.3.2 입출력 처리 노드 구성방안

입출력 처리 노드는 판넬 상의 계기들을 직접 처리해야 하므로 I/O 카드를 채용해야만 한다. 각 노드에서 처리해야 하는 많은 I/O 포인트로 인해 각 노드에서 채용해야 하는 I/O 카드를 수용하기 위해서는 많은 ISA 버스 슬롯을 필요로 한다. 본 논문에서는 많은 ISA 슬롯을 채용하기 위해서 ISA 버스 확장 카드를 이용하여 ISA 버스를 확장하는 방법을 채택하였다[4]. ISA 버스의 확장에도 한계가 있기 때문에 너무나 많은 슬롯을 채용해야 하는 경우의 처리방법으로 본 논문에서는 부분적으로 특정 I/O 포인트 종류에 대해서는 필드버스 카드를 채용하고, 그 카드를 필드버스를 채용하는 active I/O 리모트 모듈과 멀티드롭으로 연결함으로써 많은 I/O 포인트를 수용하고자 한다[4]. 필드버스를 채용하는 구성방안은 선택적으로 노드에 따라서 채용할 수도 있고 채용하지 않을 수도 있다. 본 논문에서 제시하는 입출력 처리 노드의 시스템 구성방안은 그림 4와 같으며, 화력 발전소 시뮬레이터의 I/O 인터페이스 시스템의 경우에는 리모트 I/O 모듈을 채용하지 않으며, 원자력 시뮬레이터의 경우에도, 노드에 따라서 리모트 I/O 모듈의 채용은 선택적이다.

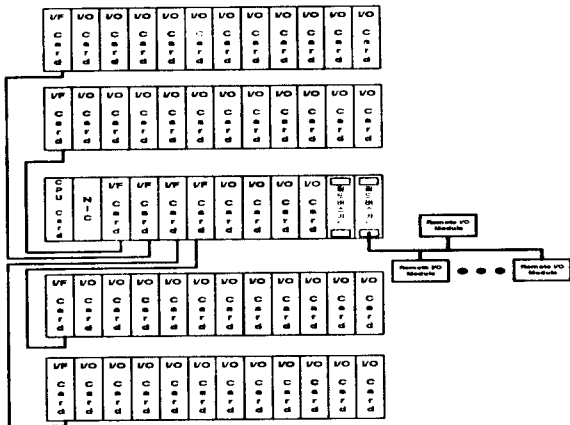


그림 4. 입출력 처리노드의 시스템 구성 방안

그림 4의 입출력 처리 노드의 시스템 구성 방안에서 I/F Card는 ISA 버스 확장 카드를 의미하며, NIC(Network Interface Controller)는 각 노드를 연결시켜 주기 위한 네트워크 카드를 의미하며, I/O 카드는 아날로그 입력/출력 카드, 디지털 입력/출력 카드, 릴레이 출력 카드를 나타낸다.

3.3.3 노드간 및 리모트 I/O 모듈간 네트워크 프로토콜

노드간 네트워크 카드로는 저가이면서 쉽게 구할 수 있고, 표준으로 널리 쓰이고 있는 이더넷 카드를 채용한다[4]. 필드버스 카드와 리모트 I/O 모듈간 프로토콜은 모듈과 카드와의 인터페이스가 용이하면서, 프로토콜이 공개되어서 쉽게 프로그램 가능한 제품을 선택한다. 본 논문에서는 1Mbps의 CAN(Controller Area Network)을 리모트 I/O 모듈간 네트워크 프로토콜로 채용한다.

3.4 I/O 인터페이스 시스템의 전체 성능 예측

1초에 12번 판넬의 I/O 포인트를 스캐닝해야 하는 조건을 만족하기 위해서, 전체 I/O 인터페이스 시스템 관점에서 소요되는 시간을 예측하도록 하겠다[6]. 표 3과 표 4의 화력과 원자력 시뮬레이터 I/O 인터페이스 시스템의 각 입출력 노드의 데이터 양을 비교해보면 원자력 시뮬레이터의 경우가 화력 시뮬레이터의 경우보다 훨씬 많은 데이터 양을 다루고 있음을 볼 수 있다. 따라서 본 절에서는 원자력 시뮬레이터의 데이터 양을 근거로 성능 예측을 하고, 그 결과를 토대로 제안한 시스템 구성방안의 효율성을 보임으로써 화력 발전소 시뮬레이터의 I/O 인터페이스 시스템의 구성방안의 효율성도 보이고자 한다.

3.4.1 NIC를 통한 데이터 교환 시간 예측

표 4의 각 노드별 처리 데이터로부터 시뮬레이션 컴퓨터 관점에서의 최대 입력 데이터와 출력 데이터를

산출하면 각각 152바이트와 616바이트가 나온다. 본 논문에서 채용하는 이더넷 프레임의 구조는 표 5와 같다.

표 5. 이더넷 프레임 구조

	preamble	D/S	Type	데이터 영역			FCS
				송신 ID	수신 ID	데이터	
Byte 수	8	6	6	1	1	44~149	4
				46~1500			

네트워크의 효율성을 높이기 위하여 Sim 노드를 버스 중재 노드로 정하고, 통신 방식은 폴링 방식을 채용하며, 응용 계층에서 TCP/IP를 거치지 않고 바로 이더넷 카드를 액세스하여 실시간성을 보장하도록 하는 방식을 채용한다. 버스 중재 노드는 통신을 위한 초기화를 수행하고, 입출력 처리 노드 1로 출력해야 할 데이터를 포함한 요청 프레임을 전송하면, 입출력 처리노드 1은 요청 프레임에서 출력 데이터를 추출하고, 자신이 생성하는 시뮬레이션 컴퓨터 관점에서의 입력데이터를 포함한 응답 프레임을 시뮬레이션 컴퓨터로 전송하게 된다. 버스 중재 노드는 노드 1로부터의 응답프레임으로부터 입력 데이터를 추출하고, 노드 2 ~ 노드 10에 차례대로 출력 데이터를 포함한 요청 프레임을 전송하고, 각 노드로부터 입력 데이터를 포함한 응답 프레임을 수신한다. 버스 중재 노드는 타이머를 2개 갖는데, 하나는 각 노드와의 데이터 교환시에 적용하고, 하나는 전체 노드 스캐닝 시간을 관장하는데 적용한다. 입출력 처리 노드 1 ~ 노드 10에서 요청 프레임을 받고, 요청 프레임에 대한 응답 프레임을 전송하는데 걸리는 시간을 TR_1 이라 하고, Sim 노드에서 응답 프레임을 받고, 요청 프레임을 전송하는데 걸리는 시간을 TR_2 라 하자. 이더넷 inter-frame gap 시간 지연인 $9.6[\mu s]$ 과 end-to-end propagation 시간 지연인 $22.5[\mu s]$ 을 반영하고, 10Mbps 이더넷 카드에 기반해 네트워크를 통한 데이터 전송시간을 예측해보면 식 (1)과 같다.

$$T_{\text{네트워크}} = 10 * (TR_1 + TR_2) + \frac{10 * 8 * (180 + 644)}{10M} + 20 * (9.6\mu s + 22.5\mu s) = 10 * (TR_1 + TR_2) + 7234\mu s \quad (1)$$

펜티엄 100 CPU 채용 PC에서의 실험결과 TR_1 은 $1.0[ms]$, TR_2 는 $1.2[ms]$ 정도 나왔다[4]. 이 결과를 식 (1)에 적용하면 $T_{\text{네트워크}}$ 은 $29.234[ms]$ 정도를 예측할 수 있다. 이 시간은 I/O 인터페이스 시스템 요구사항인 1초에 12번의 I/O 판넬 데이터 스캐닝 시간인 $83[ms]$ 을 충분히 만족함을 알 수 있다.

3.4.2 각 노드에서의 데이터 처리 시간 예측

각 노드별로 83[ms]의 샘플링 주기를 만족할 수 있는지의 여부를 판별하기 위해, 각 노드별로 데이터 처리를 위해 필요한 시간을 예측하도록 하겠다. 표 4의 최대 I/O 포인트를 수용하는 노드 9와 Sim 노드에 대해서 데이터 처리 시간을 예측하도록 한다. 먼저 노드 9의 데이터 처리 시간을 식으로 표현하면 식 (2)와 같다.

$$T_{\text{노드9데이터처리}} = T_{\text{네트워크처리}} + T_{\text{I/O카드스캔}} + T_{\text{리모트스캔}} \quad (2)$$

$T_{\text{네트워크처리}}$ 는 노드 9가 NIC를 통해 데이터를 처리하는데 시간을 의미하며, $T_{\text{I/O카드스캔}}$ 은 노드 9의 I/O 카드가 데이터를 스캔하는데 걸리는 시간을 의미한다. $T_{\text{리모트스캔}}$ 은 리모트 I/O 모듈을 통해서 데이터를 스캔하는데 걸리는 시간을 의미한다. 노드 9에서 채용하는 각 I/O 카드의 수와 리모트 모듈과 필드버스 카드의 갯수를 나열하면 표 6과 같다. 리모트 RO 모듈은 24개의 릴레이 출력 채널을 가지며, 필드버스 카드는 각각 2개의 포트를 갖는다.

표 6. 각 노드에서 필요로 하는 I/O 카드 수

카드 종류	AI	AC	DI	DC	RO	필드버스 스	리모트 RO 모듈
노드 9	1	11	12	13	10	4	16

실험 결과에 의해 $T_{\text{I/O카드스캔}}$ 은 1.71[ms]정도의 시간 소요를 예측할 수 있다[5]. CAN의 SDO(Service Data Object) 커맨드를 이용하여 폴링 방식으로 리모트 I/O 모듈의 데이터를 스캔한다. SDO 메시지 포맷은 표 7과 같고, 완전한 CAN 메시지는 표 8과 같다[7].

표 7. SDO 메시지 포맷

Byte 0	Byte 1,2	Byte 3	Byte 4,5,6,7
Read/Write	Index	Subindex	Data

표 8. 완전한 CAN 메시지 포맷

메시지 ID	RTR	DLEN	SDO 메시지
11 bit	4 bit	4 bit	8 byte

리모트 I/O 모듈에서 요청 프레임을 받고, 요청 프레임에 대한 응답 프레임을 전송하는데 걸리는 시간을 $TR_{\text{리모트}}$ 라 하고, 필드버스 카드에서 응답 프레임을 받고, 요청 프레임을 전송하는데 걸리는 시간을 $TR_{\text{카드}}$ 라 하자. 표 6의 필드버스 카드와 리모트 모듈 사이의 통신은 8채널씩 이루어진다고 하면, 2개의 리모트 RO 모듈을 스캐닝하기 위해서는 한 포트당 6번의 리모트 스캔이 이루어져야만 한다. 본 논문에서는 리모트 스캔 시간을 줄이기 위해 8개의 포트에 순차적으로 요청 프레임을 다 보낸 후, 다시 순차적으로 응답프레임을 받는 통신 방식을 취한다.

$$T_{\text{리모트스캔}} = 48 * T_{\text{요청프레임}} + 6 * TR_{\text{리모트}} + 48 * T_{\text{응답프레임}} + 6 * TR_{\text{카드}} \quad (3)$$

편의상 $TR_{\text{리모트}}$ 와 $TR_{\text{카드}}$ 는 같다고 하고, 1[ms]이라고 가정하면, $T_{\text{리모트스캔}}$ 은 19.968[ms]의 시간을 예측할 수 있다.

노드 9에서의 $T_{\text{네트워크처리}}$ 는 식 (1)을 참조하면, 7.602[ms]의 시간이 예측가능하다. 식 (2)의 $T_{\text{노드9데이터처리}}$ 은 29.28[ms]의 시간이 소요됨을 예측할 수 있다.

Sim 노드의 I/O 인터페이스 시스템과 관련되어진 데이터 처리 시간 $T_{\text{sim노드데이터처리}}$ 는 $T_{\text{네트워크처리}}$ 만 존재하므로 식 (1)을 참조하면, 18.688[ms]의 시간 소요가 예측되어진다.

표 4의 노드 9와 Sim 노드에서의 네트워크 카드를 통한 교환 시간 예측 및 각 노드에서의 데이터 처리 시간 예측을 통해 원자력 및 화력의 시뮬레이터 I/O 인터페이스 시스템이 1초에 12번의 판별 데이터 스캐닝 요구조건을 모두 만족함을 알 수 있다.

4. 결 론

본 논문에서는 발전소 운전원 훈련에 활용되고 있고, 아직도 많은 수요를 가지고 있는 전 범위 발전소 시뮬레이터를 대상으로, 구축과, 활용, 유지보수에 많은 장점을 제공하는 PC 기반 I/O 인터페이스 시스템 구축방안을 제시하고, 시스템 성능 예측을 통해 그 효용성을 보였다. 표준과 개방형 기술을 채용함으로써, 어려움없이 향후 발전소 시뮬레이터 I/O 인터페이스 시스템 구축에 활용할 수 있으리라 기대된다.

참고문헌

- [1] 한전 전력연구원, “하동화력 발전소 시뮬레이터 개발”, 중간보고서, 1998
- [2] 한전 전력연구원, “발전소 운전원 훈련용 모의제어반 국산화 개발 적용”, 최종보고서, 1998
- [3] 한전 전력연구원, “발전소 운전원 훈련용 모의제어반 국산화 개발 적용”, 외부위탁분 최종보고서, 1998
- [4] 변승현, 장태인, 광귀일, 조지용, “PC 기반 제어용 I/O 시스템 구축에 관한 연구”, 대한전자공학회 하계종합학술대회, 1998
- [5] 변승현, 장태인, 조지용, 광귀일, “발전소 시뮬레이터 I/O 인터페이스 시스템 구축에 관한 연구” 대한전자공학회 하계종합학술대회, 1999
- [6] 장태인, 광귀일, 변승현, 조지용, “PC/Ethernet 기반의 제어용 통신 프로토콜 설계·구현” 대한전자공학회 하계종합학술대회, 1999
- [7] ADVANTECH, “ADAM 5000 Series CAN” User’s Manual, 1997