

# 실시간 폐루프 제어 시스템을 위한 Profibus-FMS 네트워크의 구현

## Implementation of Profibus-FMS Network for Real-Time Closed-Loop Control System

이 경 창, 김 기 웅, 김 희 현, 이 석  
(Kyung Chang Lee, Keewoong Kim, Hee Hyun Kim, and Suk Lee)

**Abstract** : As many sensors and actuators are used in various automated systems, the application of network to real-time distributed control system is gaining acceptance in many industries. In order to take advantages of networking, however, the network should be carefully designed to satisfy real-time distributed control. This paper presents an implementation method of closed-loop control using Profibus-FMS. In order to implement a closed-loop control system, we used industrial computers with Profibus-FMS network cards and a DC servo motor. Through various experiments, the step response of the control system with network was compared with the reference response without network.

**Keywords**: fieldbus, distributed control system, Profibus-FMS, performance parameter, closed-loop control

### I. 서론

최근 고성능의 마이크로 프로세서가 발달함에 따라 대부분의 제어 시스템은 디지털 제어 시스템으로 구현되고 있다. 컴퓨터를 이용한 디지털 제어 시스템은 뛰어난 유연성(flexibility)과 적응성을 가지고 있기 때문에, 발전소와 같이 복잡한 시스템을 효과적으로 제어할 수 있다. 그러나, 이러한 시스템에서 수집할 데이터의 양과 출력할 양이 많아짐에 따라 데이터 처리량이 증가하게 됨으로써, 단일 시스템으로는 실시간 처리, 엄격한 시간 제약 조건의 만족 및 신뢰성의 확보에 한계가 발생하게 되었다. 이러한 문제의 해결을 위하여, 분산 구조가 도입되었으며, 이러한 시스템은 실시간 분산 제어 시스템(real-time Distributed Control System, DCS)으로 분류된다[1][2].

보다 좋은 응답 특성과 신뢰성을 확보하기 위하여, DCS에 더 많은 수의 센서, 구동기 및 제어기가 사용됨에 따라, 종래의 중앙집중식 연결방식은 배선의 복잡함을 가져와 시스템의 유지 및 보수에 많은 어려움을 주었다. 따라서 많은 배선을 사용하여 기기를 연결하는 것보다 유지 및 보수를 편하게 하는 네트워크의 사용에 관심을 가지게 되었다. 즉, DCS에 필드기기를 단일 전송 매체를 사용하는 단순한 구조의 네트워크에 접속시킴으로써, 시스템을 단순화시키고, 성능을 향상시킬 수 있게 되었으며, 전송 매체의 안정성 증대로 인하여, 시스템의 신뢰성을 획득할 수 있게 되었다[3][4][5].

이러한 목적으로, 1980년대 초반 GM에서 MAP (Manufacturing Automation Protocol)이 개발되었으나, MAP은 ISO에서 규정하는 OSI 참조 모델(Open System Interface reference model)의 7개 계층을 모두 사용하는

이유로, 생산 및 제어현장에서의 엄격한 실시간 제약 조건을 만족시키기에는 한계를 보여주었다. 이에 따라, 1980년대 후반부터 필드기기들 사이의 실시간 통신을 지원할 수 있는 네트워크 즉, 필드버스(fieldbus)가 개발되었다. 현재에는 Profibus, WorldFIP, CAN, Fieldbus Foundation, Device-Net, Control-Net 등과 같은 다양한 필드버스 프로토콜이 사용되고 있으며, 발전소, 제철소 등과 같은 대규모 분산제어시스템을 중심으로 활발히 적용되고 있다.

그러나, 매우 엄격한 시간 제약 조건을 가진 실시간 DCS에 네트워크를 도입하는 경우, 실시간 제약 조건이 만족되지 못하면, 시스템의 성능에 치명적인 영향을 미칠 뿐만 아니라, 인적 및 물적 피해를 입히게 된다. 따라서, DCS에 네트워크를 도입할 때, 실시간 제약 조건이 만족되도록 네트워크가 설계되어야 한다[1].

본 논문에서는 DCS용 네트워크로 Profibus-FMS를 사용하였다. Profibus-FMS는 다양한 서비스와 관리 기능을 가지고 있고, 사건신호와 같이 엄격한 시간 제약 조건을 가지고 비주기적으로 발생하는 동기 비주기 메시지(synchronous aperiodic message)가 주로 전송되는 시스템에서 매우 효율적이다. 그러나, 엄격한 시간 제약 조건을 가지고 주기적으로 발생하는 동기 주기 메시지(synchronous periodic message)가 주로 전송되는 시스템에서,

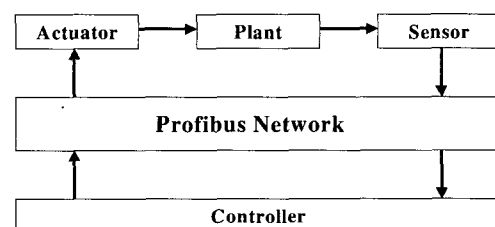


그림 1. 네트워크에서의 폐루프 제어 시스템.  
Fig. 1. Closed-loop control system on a network.

접수일자 : 2000. 1. 22., 수정완료 : 2000. 5. 27.

이경창, 김기웅 : 부산대학교 대학원 지능기계공학과

김희현 : LG전자 DA 사업부

이 석 : 부산대학교 기계공학부 및 기계기술연구소

Profibus-FMS가 시간 제약 조건을 만족시킬 수 있는지에 대해서는 정확히 알려져 있지 않다. 따라서, DCS에서 Profibus-FMS를 사용하는 경우, 시간 제약 조건을 만족하는가에 대한 연구가 필요하다.

이를 위하여, 본 논문에서는 Profibus-FMS 네트워크를 이용하여 그림 1과 같은 실시간 폐루프 제어 시스템(closed-loop control system)을 구축하였으며, 실시간 성능을 보장하기 위한 Profibus-FMS의 구축 기술에 대해 연구하였다. 2장에서는 Profibus-FMS 프로토콜의 특성에 대하여 설명하였으며, 3장에서는 실시간 폐루프 제어 시스템을 위한 플랜트와 Profibus-FMS 네트워크의 구축 및 제어 시스템의 구성에 대해 설명하였다. 4장에서는 플랜트의 제어 성능을 분석하고, 실시간 제어 시스템을 위한 Profibus-FMS 네트워크 구축 기술에 대하여 설명하였다. 마지막으로 5장에서는 결론 및 향후 과제를 서술하였다.

### II. Profibus 네트워크

Profibus 프로토콜은 OSI 참조 모델에 따라, 물리적 전송 특성을 정의한 1계층(physical layer), 버스 접속 방식을 정의한 2계층(data link layer) 및 응용 기능을 정의한 7계층(application layer)을 가지고 있으며, 추가로 사용자와의 인터페이스를 정의한 사용자 계층을 가지고 있다.

그림 2에 Profibus의 구조를 나타내었다. Profibus는 7계층 및 사용자 계층의 특징에 따라 FMS, DP, PA의 3가지로 구분한다. Profibus-DP(Decentralized Peripherals)는 빠른 데이터 전송이 보장되므로, 주로 분산된 필드기기들 간의 실시간 통신이 중요한 시스템에 주로 사용된다. Profibus-FMS(Fieldbus Message Specification)는 다양한 실시간 통신 서비스를 선택할 수 있으므로, 일반 산업 응용에 주로 사용된다. Profibus-PA(Process Automation)는 전자기적 간섭에 대한 내성(intrinsic safety)을 가지므로, 특수한 환경의 공정 제어에 주로 사용된다[6][7][8].

Profibus의 물리계층[7]에는 RS-485가 사용되며, 버스 토폴로지를 사용한다. 전송속도는 FMS의 경우 9.6Kbps에서 500Kbps까지, DP의 경우 12Mbps까지 지원한다.

데이터링크계층[7]에는 매체접속제어(Medium Access Control, MAC) 기능과 논리링크제어(Logical Link

Control, LLC) 기능이 정의되어 있다. Profibus의 경우, 토큰 전달(token passing) 방식과 폴링(polling) 방식을 함께 사용하며, 버스에 접속된 스테이션들을 마스터(master)와 슬레이브(slave)로 구분한다. 마스터는 논리적 링을 따라 회전하는 토큰을 소유한 경우 전송권한을 가지며, 다른 마스터와의 데이터 교환이나 슬레이브로의 폴링을 수행한다. 그러나, 슬레이브는 토큰을 소유할 수 없으며, 마스터의 요청에 의해서만 통신에 참여할 수 있다.

특히, 마스터는 전송권한의 설정을 위해 TTR(Target Rotation Time), TRR(Real Rotation Time), TTH (Token Holding Time)로 정의된 3개의 타이머를 갖는다. 여기서, TTR은 토큰이 버스를 1회전하는데 예상되는 시간 값이며, TRR은 실제 토큰이 버스를 1회전하는 데 걸린 시간이다. 또한, TTH는 TTR-TRR로 계산되며, 마스터의 전송권한을 판단하기 위해 사용된다. 즉, 마스터는 토큰을 소유하였을 때, TRR<TTR, 즉 TTH>0인 경우 통신을 수행한다.

Profibus-FMS의 응용계층[7]에는 FMS(Fieldbus Message Specification), LLI(Lower Layer Interface) 및 FMA7(Fieldbus Management Application 7)이 포함된다. FMS는 통신객체와 응용서비스를 정의하며, LLI는 응용서비스와 FDL(Fieldbus Data link Layer)을 접속하는 기능을 정의하고 있다. 마지막으로 FMA7은 네트워크 관리를 정의하고 있다.

Profibus-FMS에서 통신 서비스가 수행되기 위해서는 서비스를 요구하는 클라이언트와 요구에 응답하는 서버와의 관계가 설정되어야 한다. 여기에서 클라이언트와 서버 관계는 통신객체(communication entity), 통신관계(communication relationship) 및 통신서비스(communication service)들로 구성된다. 통신객체는 각 노드 사이에서 교환되는 데이터를 의미하며, 센서에서 측정되는 데이터나 통신 관련 파라미터 등이 여기에 속한다. 한 노드는 여러 개의 통신객체를 가질 수 있으며, 이러한 통신객체의 목록을 OD(Object Dictionary)라 한다. 통신관계는 각 노드간 서비스의 논리적 연결 기능을 의미하며, 통신관계의 목록은 CRL(Communication Relationship List)에서 관리된다. 이 목록은 CR (Communication Reference)순으로 기입되며, 각 CR은 해당 노드에서 응용 프로세스들 간의 데이터 교환을 위한 관계 설정이 정의되어 있다. 통신서비스는 이들 통신객체들에 대한 해당 노드의 응용 서비스를 의미하며, Profibus-FMS에서는 FMS, LLI, FMA7 등으로 분류된 서비스들이 사용된다.

Profibus-FMS의 응용계층인 FMS는 필드버스의 사용자와 연결되는 부분으로, 필드기기의 응용 프로그램과 연결된다. FMS의 서비스에는 VFD관리(Virtual Field Device support), OD관리(Object Directory management), 연결관리(context management), 변수관리(variable access), 영역관리(domain management), 프로그램기동관리(program invocation management), 사건관리(event management) 등의 7개 서비스군이 있다. LLI는 7계층과 2계층의 상호연결(mapping)을 다루며, 흐름 제어(flow control)와 연결 관리를 수행한다. 마지막으로 FMA7은 버

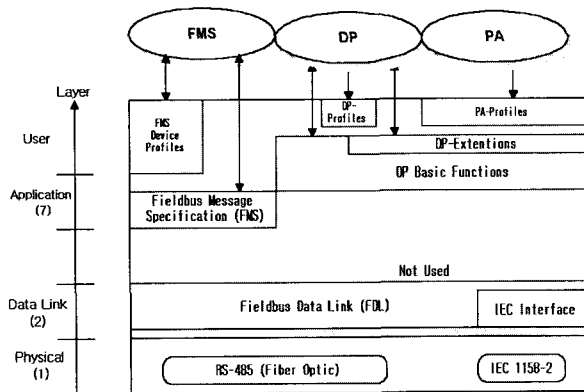


그림 2. Profibus 프로토콜의 구조.

Fig. 2. Architecture of Profibus protocol.

스 시스템 전체의 구성과 유지, 보수 등의 Profibus의 관리 역할을 수행하며, 연결관리, 오류관리(fault management), 시스템 구성 관리(configuration management)의 기능을 제공한다.

III. 네트워크를 이용한 페루프 제어 시스템 구성

1. 플랜트 선정 및 분석

본 연구에서는 네트워크에서의 페루프 제어 성능을 파악하기 위하여, 먼저 플랜트를 선정하고, 이를 제어하기 위한 제어기를 선정하였다.

먼저, 플랜트로는 그림 3의 블록선도와 같이 DC 서보 모터를 선정하였으며, 시스템의 동특성 및 미지 파라미터를 추정하는 방법중의 하나인 신호압축법[9]을 이용하여 DC 서보 모터의 동특성을 그림 4의 보드선도와 같이 구하였다. 또한, 모터의 페루프 전달함수를 2차로 가정하고, 그림 4의 보드선도를 이용하여  $\omega = 400\text{rad/s}$ 까지 주파수 특성을 만족시킬 수 있는 페루프 전달함수  $C(s)$ 를 (1)과 같이 추정하고, (2)와 같이 플랜트 전달함수  $G(s)$ 를 계산하였다.

$$C(s) = \frac{KG(s)}{1 + KG(s)} = \frac{0.82 \times 260^2}{s^2 + 2 \times 0.78 \times 260s + 260^2} \quad (1)$$

$$G(s) = \frac{55432}{s^2 + 405.6s + 12168} \quad (2)$$

마지막으로, 설계 사양을  $M_p \leq 25\%$ ,  $\xi = 0.5$ ,  $T_S$ (정착시간)  $\leq 0.6\text{sec}$ 로 설정하여, PID 제어기를 (3)과 같이 설계하였다. 여기에서  $K_p$ 는 비례이득,  $K_i$ 는 적분이득,  $K_d$ 는 미분이득이며, 각각  $K_p=1.909$ ,  $K_i=99.6$ ,  $K_d=0.0083$ 이다[10].

$$G_{PID}(s) = K_p + K_i \frac{1}{s} + K_d s = \frac{0.0083s^2 + 1.909s + 99.6}{s} \quad (3)$$

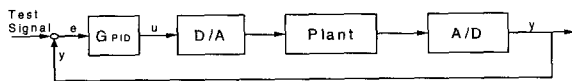


그림 3. DC 서보 모터 시스템의 블록선도.  
Fig. 3. Block diagram of DC servo motor system.

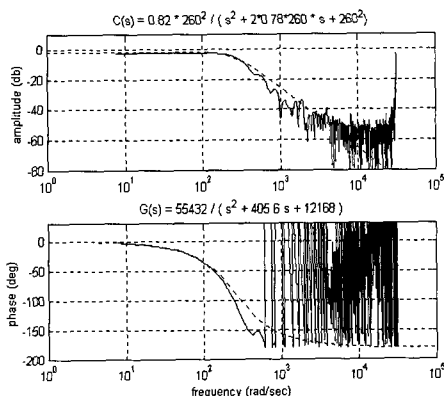


그림 4. DC 서보 모터의 보드선도.  
Fig. 4. Bode diagram for DC servo motor.

설계된 제어기를 사용하여 플랜트의 응답을 구한 결과는 그림 5와 같다. 그림에서 PID 제어기를 이용한 결과,  $M_p=22\%$ ,  $T_s=0.56\text{sec}$ 로 나타났으며, 설계된 제어기가 설계 사양을 만족하고 있음을 알 수 있다.

2. Profibus-FMS에서의 페루프 제어 시스템 구성

일반적으로 발전소 등과 같은 대규모 DCS의 경우, 입출력 데이터의 송수신이 여러 기기들에 의해 이루어진다. 따라서, 본 연구에서는 DCS의 입출력 데이터의 전송을 모사하기 위하여, 그림 6과 같이 플랜트의 입력과 출력이 각각 하나의 스테이션에서 이루어지는 실험 모델을 구성하였으며, 이를 이용하여 Profibus-FMS에서의 페루프 제어 성능을 평가하였다.

실험을 위한 네트워크에는 4개의 스테이션을 접속하였다. 먼저, Profibus-FMS 서비스를 수행할 수 있는 Softing사의 Profi-IF 보드를 PC1 및 PC2에 장착하고, 모터 제어 신호를 입출력하기 위하여 DR\_DAS12 보드를 추가하였다. 또한, 네트워크에 부가적인 부하를 가하기 위하여 디지털 입출력 및 PLC 기능을 할 수 있는 SMART I/O 2대를 마스터로 접속하였으며, 네트워크에 부하를 주지 않고, 전송되는 프레임의 상태를 관찰할 수 있는 버스 모니터(bus monitor)를 접속하였다[11].

실험 모델에서 PC1은 Interrupt 함수를 사용하여 10msec 간격으로 모터의 출력 전압을 DR\_DAS12 보드의

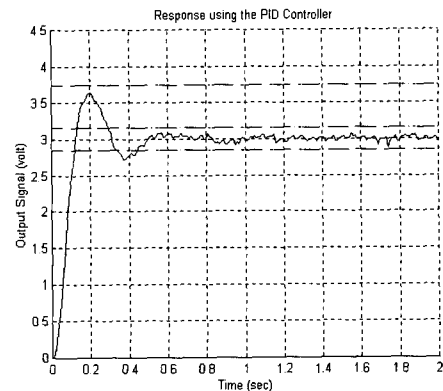


그림 5. PID 제어기를 이용한 단위응답.  
Fig. 5. Step response using the PID controller.

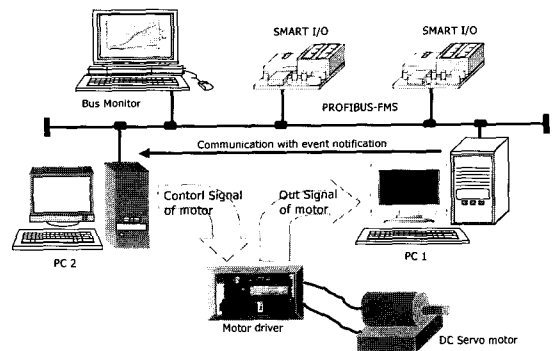


그림 6. DC 서보 모터 제어 실험 모델.  
Fig. 6. Experiment model for DC servo motor control.

A/D 변환기를 통해 읽어들이며, 이를 Profibus-FMS의 서비스 중 하나인 Event Notification 서비스를 사용하여 PC1의 CR에 정의되어 있는 PC2로 모터의 출력 신호를 전송한다. 이때, PC2는 모터 출력 신호를 전송받은 후, PID 제어 프로그램을 실행하여 모터 제어 신호를 만들며, 이를 DR\_DAS12 보드의 D/A 변환기를 사용하여 모터 구동기로 전송하여, 모터를 제어하게 된다.

Profibus 네트워크에서 출력 및 제어 신호의 전송에 영향을 주는 변수로는 연결관계(connection type)와 TTR, max\_scc(maximum send confirmed counter), max\_rcc(maximum receive confirmed counter), max\_sac(maximum send acknowledge counter), max\_rac(maximum receive acknowledge counter) 등이 있으며, 이들 변수에 대한 자세한 설명은 다음과 같다.

· 연결관계는 서비스의 종류에 따른 두 스테이션간의 통신 연결관계로, 본 실험에서는 MMAC(master-master connection for acyclic data transfer)와 MSAC(master-slave connection for acyclic data transfer)의 두 가지 경우를 고려하였다. 여기에서 MMAC는 마스터와 마스터 스테이션이 비순환 메시지를 전송하는 연결관계이며, MSAC는 마스터와 슬레이브 스테이션이 비순환 메시지를 전송하는 관계이다.

· TTR은 한 스테이션이 최대 전송할 수 있는 시간을 결정하는 변수로서, 본 실험에서는 5msec, 20msec, 100msec의 세 가지 경우를 고려하였다.

· max\_scc와 max\_rcc는 비동기 데이터 전송에서 마스터 측이 동시에 요구 및 응답할 수 있는 confirmed LLI 서비스의 최대 개수이며, max\_sac와 max\_rac는 동시에 요구 및 응답할 수 있는 unconfirmed LLI 서비스의 최대 개수로서, 본 실험에서는 각각 1, 3, 20의 세 가지 경우를 고려하였다.

**IV. 페루프 제어 시스템의 제어 성능 평가**

본 실험에서는 페루프 제어 시스템의 응답 특성을 평가하기 위하여, 10ms 주기로 생성되는 모터 제어 신호를 PC1에서 제어기 프로그램을 가진 PC2로 전송하는 실험을 4초간 수행하였으며, 이때 생성되는 제어 신호의 처리율(throughput, 전송된 메시지 수를 생성된 메시지 수로 나눈 값)과 오버슈트(% overshoot, 스텝응답에서 최대값과 정상상태값과의 차이를 정상상태값으로 나눈 값의 백분율), 상승시간(rise time, 스텝응답에서 최종값의 10%에서 90%에 도달하는 데 필요한 시간) 및 정착시간(settling time, 스텝응답이 최종값의 5% 이내에 들어가는 데 필요한 시간) 등과 같은 모터의 스텝 응답 특성을 관찰하였다.

**1. 처리율에 대한 연결관계 및 TTR의 영향 평가**

먼저 연결관계, max\_scc, max\_rcc, max\_sac, max\_rac와 같은 카운터, TTR과 처리율간의 관계에 대하여 실험을 하였다. 본 실험에서는 표 1과 같이 통신관계와 max\_scc, max\_sac들을 6가지 조합(MMAC1, MSAC1, MMAC3, MSAC3, MMAC20, MSAC20)으로 구성하였으며, 각 조합에 따라 TTR을 5msec~1000msec까지 변화시켜 가면서

표 1. 실험조건.

Table 1. Experiment conditions.

Type	Connection Type	max_scc	max_rcc	max_sac	mac_rac
MMAC1	MMAC	1	1	1	1
MSAC1	MSAC	1	1	1	1
MMAC3	MMAC	3	3	3	3
MSAC3	MSAC	3	3	3	3
MMAC20	MMAC	20	20	20	20
MSAC20	MSAC	20	20	20	20

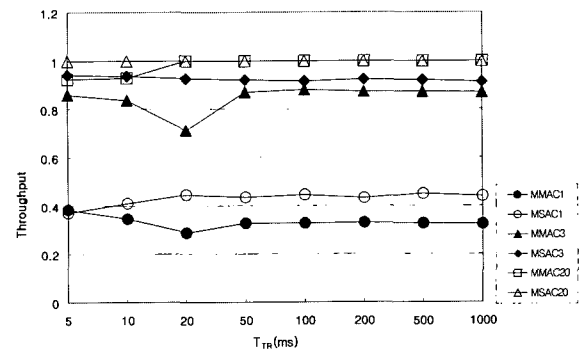


그림 7. 연결관계, TTR 및 처리율과의 관계.

Fig. 7. Relationship of connection type, TTR and throughput.

처리율을 측정하였다.

그림 7에 연결관계와 max\_scc, max\_sac들의 조합 및 TTR과 처리율간의 관계를 나타내었다. 먼저, max\_scc, max\_sac들이 1인 경우 처리율이 0.5 이하로 나타났다. 이 이유는 max\_scc, max\_sac들을 1로 설정할 때, PC1의 전송큐에 하나의 제어신호만이 저장될 수 있기 때문이다. 즉, 실험에서 토큰이 1회전하는데 10msec 이상이 소요되기 때문에, 제어신호가 생성된 후 10msec 이내에 전송되지 못하면, 전송큐에 계속 남아 있게 되며, 이에 따라, 다음 주기에 생성되는 제어신호가 저장되지 못하고 소멸되어 버린다. 이 이유로 처리율이 50% 이하로 나타난다.

max\_scc, max\_sac들이 3일 때, MSAC의 경우 처리율은 전 영역에서 0.9 이상이며, MMAC의 경우 TTR이 50msec 이상일 때 처리율은 거의 0.9가 됨을 관찰할 수 있었다. 이 이유는 MMAC의 경우, 두 스테이션이 토큰을 주고 받으면서 통신을 하므로, TTR의 영향을 많이 받게 되나, MSAC의 경우는 슬레이브 스테이션으로 토큰을 전송하지 않으므로, 이의 영향을 고려하지 않아도 되기 때문이다.

max\_scc, max\_sac들이 20일 때, MSAC의 경우 처리율은 전 영역에서 1로 나타났으며, MMAC의 경우 20msec 이상에서 1을 나타내고 있다.

**2. 페루프 제어 시스템의 응답 특성 평가**

본 실험에서는 연결관계와 max\_scc, max\_sac들의 6개 조합에 대하여, TTR이 5, 20, 100msec일 때의 스텝응답을 측정하고, 이를 네트워크가 없는 경우의 스텝응답과 비교

하였다.

그림 8, 9는 max\_scc, max\_sac들이 1일 때 MMAC 및 MSAC의 스텝응답이다. 네트워크에서 모터의 스텝응답에는 오버슈트가 없지만, 상승시간이 기준응답(reference response without network)에 비해 3배 이상 크게 나타났다. 이는 max\_scc, max\_sac들의 영향에서 살펴본 바와 같이 제어신호의 처리율이 0.5이기 때문에 나타나는 결과이다. 특히, MMAC에서는 TTR에 관계없이 응답 특성이 유사하며, MSAC에서는 TTR이 5msec일 때 매우 느린 응답을 보였고, 20msec와 100msec일 때는 응답이 비교적 빨라진 것을 관찰할 수 있었다. 또한, 20msec 이상에서는 MSAC가 MMAC보다 상승시간이 빠른 것으로 나타났다.

그림 10, 11은 max\_scc, max\_sac들이 3일 때 MMAC 및 MSAC의 스텝응답이다. 두 경우 모두 응답 특성이 기준응답을 따라가고 있으며, MSAC에서 TTR이 5msec일 때, 기준응답과 거의 유사하다. 그러나, 두 경우 모두 오버슈트가 약간 크고, 정착시간이 조금 더 길게 나타났는데, 이는 네트워크에서의 전송지연에 의한 영향 때문인 것으로 평가된다. 여기에서도 MSAC가 MMAC보다 응답 특성이 더 좋음을 관찰하였다.

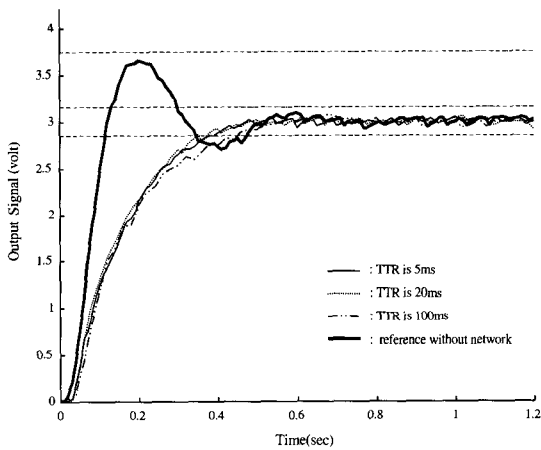


그림 8. MMAC1 연결관계에서의 제어 신호의 응답.  
Fig. 8. Response of the control signal in MMAC1.

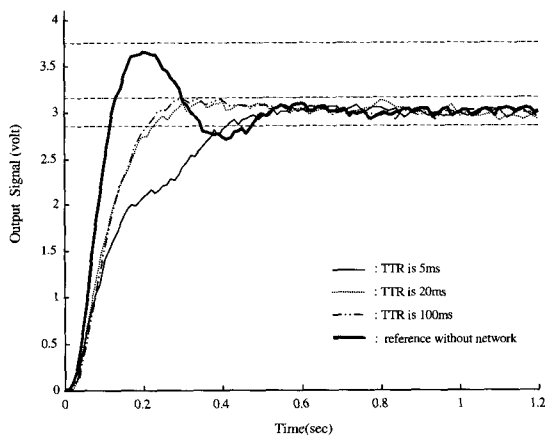


그림 9. MSAC1 연결관계에서의 제어 신호의 응답.  
Fig. 9. Response of the control signal in MSAC1.

그림 12, 13은 max\_scc, max\_sac들이 20일 때 MMAC 및 MSAC의 스텝응답이다. 두 경우 모두 오버슈트가 플랜트의 설계사양보다 크게 나타났으며, 정착시간도 설계사양보다 크게 나타났다. 이는 전송큐의 크기가 크므로, 제어신호가 큐에 오랫동안 누적됨으로써, 전송지연이 증가되기 때문이다.

이상의 실험을 통하여 연결관계, max\_scc, max\_sac들의 카운터 및 TTR의 영향에 대해 다음과 같은 결과를 얻을 수 있었다.

첫째, 샘플링 신호와 같은 주기 신호의 전송에는 MMAC 연결관계보다 MSAC 연결관계가 우수함을 알 수 있었다.

둘째, max\_scc, max\_sac들이 1로 설정되어 있는 경우, 메시지가 빈번하게 발생할 때, 발생주기 내에 전송되지 못하면, 처리율이 낮아지게 되며, 시스템의 성능을 악화시키게 됨을 관찰하였다. 또한 max\_scc, max\_sac들이 20인 경우, 메시지들이 전송큐에 누적됨으로써, 전송지연이 증가하였으며, 이에 따라 시스템의 성능이 악화됨을 관찰하였다. 따라서, max\_scc, max\_sac들의 설계시 시스템의 특성을 고려하여, 적절한 값을 선정할 필요가 있음을 알 수 있었다.

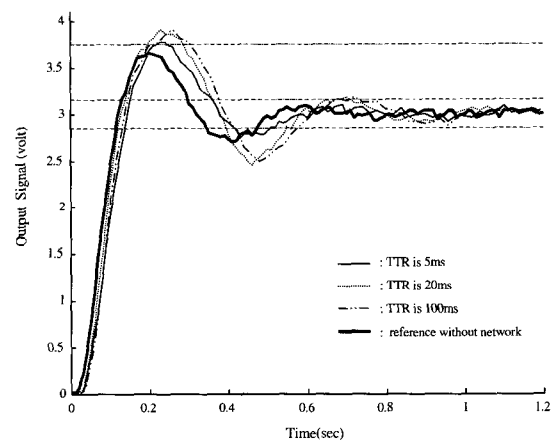


그림 10. MMAC3 연결관계에서의 제어 신호의 응답.  
Fig. 10. Response of the control signal in MMAC3.

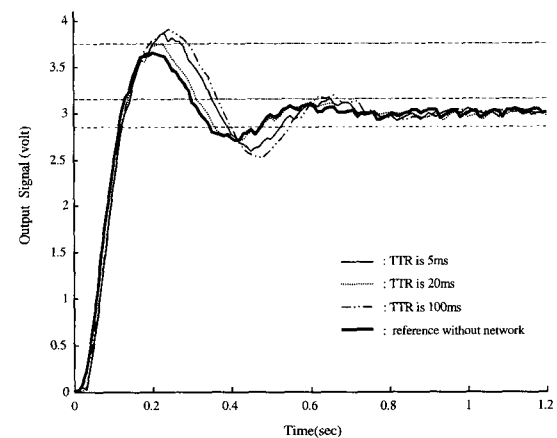


그림 11. MSAC3 연결관계에서의 제어 신호의 응답.  
Fig. 11. Response of the control signal in MSAC3.

셋째, TTR의 경우, 메시지 발생주기인 10msec보다 작은 경우에는 성능이 좋지 못함을 관찰하였으며, TTR이 20msec 이상에서는 TTR의 변화에는 관계없이 유사한 성능을 보임을 관찰하였다. 따라서, TTR을 메시지 발생주기보다 크게 설정하면, 성능에는 큰 차이가 없음을 알 수 있었다.

넷째, 네트워크를 이용하여 폐루프 제어 시스템을 구축하는 경우, 네트워크에서의 전송지연으로 인하여, 동일한 PID 제어를 사용한 경우보다 응답 특성이 느려짐을 관찰하였다. 따라서, 네트워크를 이용한 폐루프 제어 시스템을 구축하는 경우, 네트워크에서의 전송시간을 고려하여 제어를 설계하여야 함을 알 수 있었다.

**V. 결론**

본 연구에서는 다양한 형태의 서비스 구현이 필요한 일반 산업 응용에 주로 사용되는 Profibus-FMS 네트워크에서 실시간 폐루프 제어 시스템을 구현하였으며, Visual

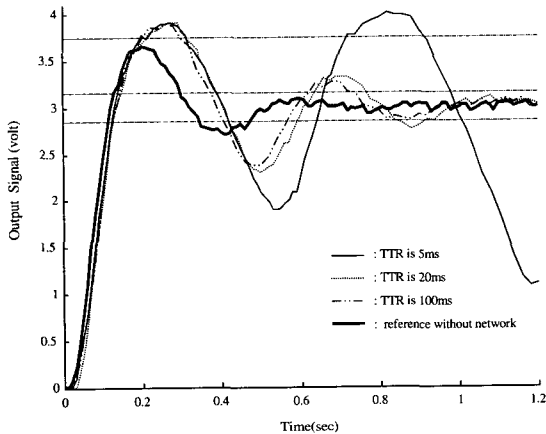


그림 12. MMAC20 연결관계에서의 제어 신호의 응답.

Fig. 12. Response of the control signal in MMAC20.

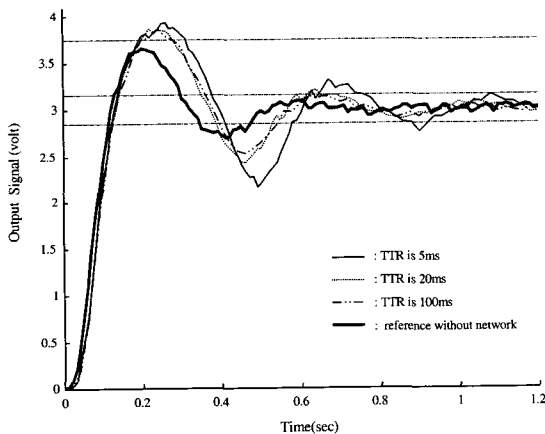


그림 13. MSAC20 연결관계에서의 제어 신호의 응답.

Fig. 13. Response of the control signal in MSAC20.

C++을 이용하여 응용 소프트웨어를 개발하였다. 또한, 플랜트로 DC 서보 모터를 선정하고, 제어 신호가 네트워크를 통하여 전송될 때의 스텝응답을 구하였으며, 이를 네트워크를 사용하지 않았을 때의 스텝응답과 비교하여 다음과 같은 결론을 얻었다.

첫째, 네트워크를 이용한 제어기에서 제어신호는 전송지연을 가지기 때문에, 네트워크를 사용하지 않았을 경우의 PID 제어기보다 응답 특성이 좋지 못함을 알 수 있었다. 따라서, 네트워크에서 제어 시스템을 사용하는 경우, 제어신호의 전송지연을 고려하여 제어를 설계하여야 한다.

둘째, 네트워크를 이용하여 엄격한 시간 제약 조건을 가진 실시간 폐루프 제어 시스템을 구축하는 경우, 프로토콜의 성능 변수에 따라 응답 특성이 달라지므로, 성능 변수의 적절한 선정이 필요함을 알 수 있었다. 특히, Profibus-FMS에서는 스테이션간의 통신 관계를 MSAC로 설정하고, max\_scc, max\_sac 등과 같은 큐 사이즈를 메시지의 처리율이 0.9~1이 되도록 설정하면, 원하는 제어 사양을 만족시킬 수 있는 제어를 설계할 수 있다.

본 연구의 후속 연구로는 네트워크에서의 제어기가 가진 전송지연을 고려하여 제어를 설계하는 방법에 대한 연구가 필요하며, 하나의 제어기가 다수의 필드기기를 제어하는 경우의 네트워크 설계 기법에 대한 연구가 필요하다.

**참고문헌**

- [1] 권옥현, 김영신, “분산 실시간 시스템에서의 네트워크 프로토콜”, 제어·자동화·시스템공학회지, 4권, 6호, pp. 27~34, 1998. 11.
- [2] 홍성수, “분산 실시간 제어 시스템의 개발”, 제어·자동화·시스템공학회지, 4권, 1호, pp. 33~38, 1998. 1.
- [3] A. Ray, “Networking for computer-integrated manufacturing”, *IEEE Network*, vol. 2, no. 3, pp. 40~47, 1988.
- [4] 홍승호, “필드버스 기술 동향”, 제어·자동화·시스템공학회지, 4권, 6호, pp. 13~18, 1998. 11.
- [5] 이경창, 김인준, 이석, “공장자동화를 위한 필드버스 프로토콜의 평가”, 한국정밀공학회, 16권, 3호, pp. 116~127, 1999. 3.
- [6] PNO(Profibus Network Organization), Technical Overview/hints and Tips, <http://www.profibus.com>, 1999.
- [7] EN 50170, Profibus Specification - Normative Parts of Profibus-FMS, -DP, -PA according to the European Standard, vol. 2, March, 1998.
- [8] Klaus Bender, Profibus-The Fieldbus for Industrial Automation, Prentice-Hall, 1993.
- [9] 홍영호, 이민철, “신호압축법을 이용한 위치제어용 유압실린더의 성능평가”, '95 KACC, pp. 693~696, 1995.
- [10] Katsuhiko Ogata, *Modern Control Engineering*, Prentice-Hall, pp. 257~270, 1990.
- [11] [DR\_DAS 12] 사용자 설명서, (주)다림시스템, 1992.



**이 경 창**

1971년 5월 1일. 1996년 부산대 생산기계공학과 졸업. 동대학원 석사(1998), 1998년~현재 동대학원 박사과정, 1997년~현재 기계공학연구정보센터 근무. 관심분야는 자동화네트워크

(Profibus, Fieldbus Foundation, Lonworks), IEEE 802.11 WLAN, Fuzzy Control, Web, 데이터베이스.



**김 기 응**

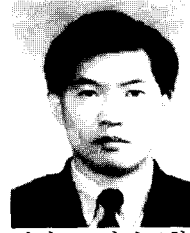
1972년 11월 29일. 1996년 부산대 생산기계공학과 졸업. 동대학원 석사(1998), 1998년~현재 동대학원 박사과정, 관심분야는 자동화 및 제어용 프로토콜(CAN, LIN, KWP2000), 자동차용

운영체제, 실시간제어, 이산사건시스템(DES).



**김 희 현**

1974년 7월 22일. 부산대 제어기계공학과 졸업. 동대학원 석사(1999), 1999년~현재 LG전자 DA사업부 근무. 관심분야는 자동화네트워크(Profibus).



**이 석**

1961년 12월 11일. 1984년 서울대 기계공학과 졸업. 1985년 Pennsylvania State Univ. 석사, 동대학원 박사(1990년), 1990년~1993년 신시내티 대학교 기계공학과 조교수 1993년~현재 부산

대학교 기계공학부 부교수. 관심분야는 자동화네트워크(Profibus, Fieldbus Foundation), 차량용 네트워크(CAN, LIN, KWP2000), DES, 자율주행.