

## CAN(Controller Area Network) 프로토콜을 이용한 네트워크 제어시스템 설계

정준홍\*, 최수영\*, 조용석\*\*, 박기환\*  
\*성균관대학교 정보통신공학부, \*\*건양대학교 IT학부

### Design Methodology of Networked Control System using CAN(Controller Area Network) Protocol

Joonhong Jung\*, Sooyoung Choi\*, Yongseok Cho\*\*, Kiheon Park\*  
\*School of Information and Communication Eng., SungKyunKwan Univ.  
\*\*School of Information Tech., Konyang Univ.

**Abstract** - This paper presents a new design methodology of networked control system using CAN(Controller Area Network). Feedback control systems having control loops closed through a network are called networked control systems. We design CAN nodes which can transmit control and monitoring data through network bus and apply these to networked control system design. We analyze the variation of stability property according to network-induced delay and determine a proper sampling period of networked control system that preserves stability performance. The results of the experimental example validate effectiveness of our networked control system.

### 1. 서 론

네트워크 제어시스템(networked control system)은 센서, 제어기, 그리고 구동기에 이르는 제어경로가 공유매체(bus)를 통해 이루어진 시스템을 말하며, 다수의 제어장치를 통합하여 운영해야 하는 대규모 제어시스템이나 분산형 제어시스템에 효과적으로 적용되고 있다[1]~[6]. 네트워크를 이용한 제어방식은 기존의 제어장치간 연결에 주로 사용되었던 직접 연결방식이나 RS-232를 이용한 일대일 연결방식에 비해 시스템 구성에 필요한 배선을 최소화 할 수 있고, 플랜트 확장 시 기존 시스템과의 호환성을 향상시킬 뿐만 아니라, 시스템 유지, 보수를 용이하게 만드는 등의 장점을 가진다. 그러나, 네트워크 시간지연이나 패킷손실 등과 같은 불확실성을 제어시스템 설계문제에 추가적으로 반영해야 하는 단점을 지니고 있다[3]~[6]. 네트워크 제어시스템 설계 시 주로 사용되는 프로토콜로는 PROFIBUS, TCP/IP(Ethernet), CAN(Controller Area Network) 등이 있다[3],[7]. 이 중에서 CAN은 저가의 하드웨어 장치로 효율성이 높은 네트워크를 구축할 수 있으며, 프레임 오버헤드가 작아 크기가 작은 주기적인 데이터 전송에 적합하고, 통신잡음에 강인하며 강력한 오류검출 및 처리 기능을 내장하고 있어 전송 데이터의 신뢰성이 매우 높기 때문에 제어응용에 적합한 프로토콜로 평가받고 있다[3],[6],[8].

본 논문에서는 CAN 프로토콜을 이용한 네트워크 제어시스템 설계방법을 제안하였다. 이를 위해 제어에 관련된 각종 데이터를 CAN 버스를 통해 송수신 할 수 있는 센서, 구동, 그리고 제어노드를 직접 설계/제작하여 실험용 네트워크 제어시스템을 구축하였다. 이후, 네트워크 제어시스템

에서 발생하는 시간지연을 측정하고, 이러한 시간지연 하에서도 전체 시스템의 안정도를 보장할 수 있는 샘플링 주기를 결정하였다. 마지막으로, 모의 발전계통을 네트워크 제어시스템의 제어대상 플랜트로 선정하고 본 논문에서 구축한 네트워크 제어시스템을 적용, 제어 실험을 수행함으로써 제안된 설계방법의 효율성을 입증하였다.

## 2. 네트워크 제어시스템의 설계

### 2.1 CAN 노드

일반적인 네트워크 제어시스템은 그림 1과 같이 플랜트, 제어기, 센서, 구동기, 그리고 이들을 연결하는 공유매체인 네트워크 버스로 구성된다[1]~[4],[6].

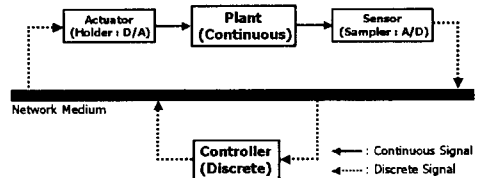


그림 1 일반적인 네트워크 제어시스템  
Fig. 1 Typical networked control system

위의 네트워크 제어시스템에서 한 샘플링 주기동안 발생하는 신호 흐름은 플랜트(출력)→센서(sampler)→네트워크→이산제어기→네트워크→구동장치(holder)→플랜트(입력)와 같은 순서를 가진다. 센서노드(sensor node)는 샘플링 주기를 기준으로 플랜트의 출력(연속신호)을 CAN 프로토콜로 변환하여 제어노드에 전송하고, 제어노드(control node)는 센서노드에서 전송하는 데이터를 이용하여 제어입력을 계산한 후 구동노드에 전송하며, 구동노드(actuator node)는 제어노드에서 전송하는 제어입력 데이터를 연속신호로 변환하여 플랜트에 인가하는 역할을 수행한다. 이러한 센서, 구동, 그리고 제어노드의 역할을 수행할 수 있도록 본 논문에서 설계한 CAN 노드는 그림 2의 구조를 갖는다. 여기서, CAN 전송기는 Philips PCA82C250, CAN 제어기는 Philips PCA82C200을 사용하였으며, 센서노드의 AD 변환과 구동노드의 DA 변환은 각각 Analog Device AD1674, AD667을 이용하여 구현하였다. 그리고, 이들을 제어하기 위해 Intel 80C196KC20 마이크로컨트롤러를 주 제어기로 사용하였으며, 제어노드와 PC사이에는 프로그램 관리, 모니터링 데이터 전송 등을 위한 직렬통신 기능을 추가하였다(RS-233 드라이버).

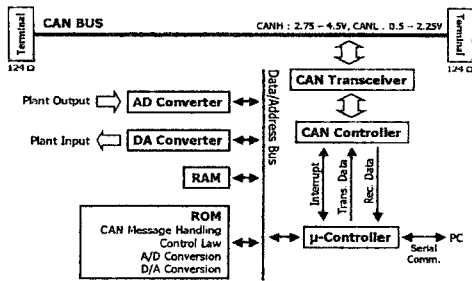


그림 2 CAN 노드의 구조  
Fig. 2 The configuration of CAN node

## 2.2 네트워크 제어시스템의 시간지연 분석

네트워크 제어시스템 설계에서 가장 먼저 고려해야 할 사항은 시간지연에 따른 전체 시스템의 안정도 변화이다 [4]~[6]. 네트워크 제어시스템에서 발생하는 시간지연은 노드를 구성하고 있는 하드웨어의 종류와 성능, 네트워크 프로토콜, 그리고 응용계층의 프로그램에 따라 크게 달라진다 [6],[8]. 본 논문에서는 네트워크 제어시스템의 시간지연을 측정하기 위해 센서, 제어, 그리고 구동 노드의 마이크로컨트롤러에 내장된 타이머 기능을 이용하였다. 즉, 전체 네트워크 제어시스템이 매 주기마다 그림 3과 같은 제어흐름을 갖도록 설계할 때, 한 주기동안 발생하는 시간지연  $r$ 는 각 노드에서 측정된 타이머 값인  $T_i$ 를 이용하여  $r = T_5 - T_0$ 와 같이 계산할 수 있다.

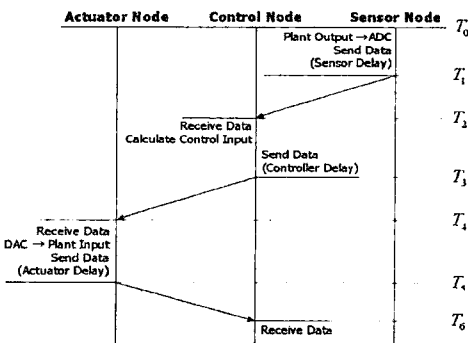


그림 3 네트워크 제어시스템의 제어흐름선도  
Fig. 3 Control flow graph of networked control system

## 2.3 전체 시스템 구성

본 논문에서는 식(1)과 같은 동적 방정식을 갖는 발전계통을 제어대상 플랜트로 선정하고, 이를 dSPACE 사의 제어시스템 개발/분석장치인 DS1102로 구현한 시스템을 대상으로 네트워크 제어시스템을 구축하였다[6],[9].

$$\dot{x}(t) = Ax(t) + Bu(t) + F\xi(t)$$

$$y(t) = Cx(t) \quad (1)$$

여기서,  $A = \begin{bmatrix} 0 & -0.139 & -0.128 & 0 \\ 376.990 & 0 & 0 & 0 \\ 0 & -0.201 & -0.476 & -27.418 \\ 0 & -5.310 & 19.670 & -50.000 \end{bmatrix}$

$$B = \begin{bmatrix} 0 \\ 0 \\ 27.418 \\ 0 \end{bmatrix}, F = \begin{bmatrix} 0.143 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

식(1)의 플랜트는 우반평면에 극점을 갖는 불안정한 시스템이므로, 이를 안정화하기 위한 제어기는  $H_2$  제어기법을 이용하여 식(2)와 같이 구하였다[6].

$$G_{DSS}(s) = \begin{bmatrix} -4.638 & -0.139 & -0.128 & 0 & 4.638 \\ 376.990 & 0 & 0 & 0 & -0.000 \\ 892.383 & -6.715 & -8.368 & -23.551 & -24.252 \\ 7.519 & -5.310 & 19.670 & -50.000 & -7.519 \\ \hline -31.663 & 0.238 & 0.288 & -0.141 & 0 \end{bmatrix} \quad (2)$$

플랜트의 입력, 출력, 상태, 그리고 제어 및 모니터링을 위한 모든 데이터는 각 노드를 통해 CAN 버스에 연결된다. 전체 시스템의 구성도는 그림 4와 같다.

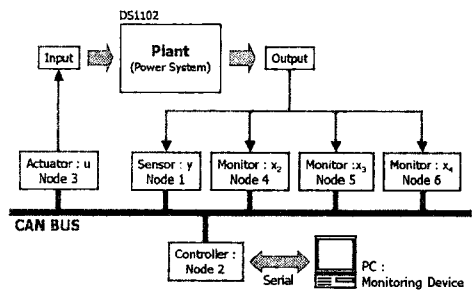


그림 4 CAN 프로토콜을 이용한 네트워크 제어시스템  
Fig. 4 Networked control system using CAN protocol

실험용 네트워크 제어시스템의 시간지연을 약 35분 동안 측정된 결과, 평균시간지연은  $15.99[msec]$ , 최대시간지연은  $16.14[msec]$  이었다. 따라서, 식(1)의 제어대상 플랜트와 식(2)의 제어기를 네트워크 제어시스템으로 구성할 경우, 최대시간지연인  $16.14[msec]$  하에서도 전체 시스템의 안정도가 유지되도록 설계해야 함을 알 수 있다. 이를 위해서는 시간지연과 샘플링 주기에 따른 전체 시스템의 안정도 변화를 분석하는 것이 필수적이며, 본 논문에서는 참고문헌 6의 [정리 2]를 이용하여 이를 수행하였다. 그 결과, 시간지연이  $r = 16.14[msec]$  일 때 위의 네트워크 제어시스템이 안정하기 위한 샘플링 주기  $h$ 의 범위를 식(3)과 같이 구하였다.

$$16.14[msec] \leq h \leq 85.46[msec] \quad (3)$$

## 3. 실험 및 결과분석

CAN 프로토콜을 이용한 네트워크 제어시스템을 실제 구현한 사진은 그림 5와 같다. 여기서, 전체 네트워크 제어시스템의 샘플링 주기는 식(3)을 만족하도록  $h = 60[msec]$ 로 결정하였다.

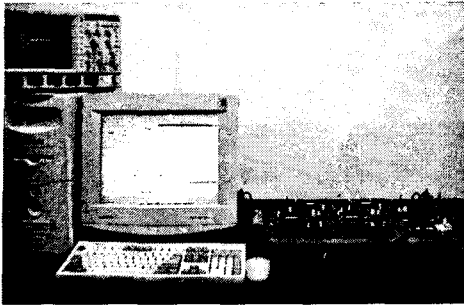


그림 5 실험장치 구성  
Fig. 5 Experimental setup

실험에서 사용한 CAN 통신속도는 0.5[Mbps]이며, 버스는 일반적인 TP(Twisted Pair) 선을 사용하였다. 센서노드와 구동노드는 0.5m 거리를 두고 CAN 버스를 중앙에 설치한 후 연결하였으며, 제어노드는 20m 떨어진 원격지에 설치하였다.

그림 6은 발전계통에 크기  $\xi \Delta(t) = 1$  [pu]의 외란을 0.1[sec]동안 인가하였을 때 측정된 출력 파형이다.

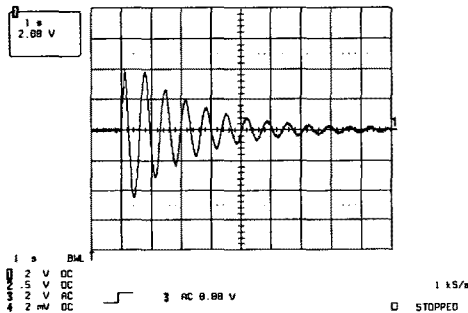


그림 6 측정 출력  
Fig. 6 Measurement output

그림 7은 이 때의 측정 출력과 시간지연이 16.14[msec]일 때의 시뮬레이션 출력을 비교한 것이며, 이로부터 실험에서 측정된 값이 이론적인 출력 값과 거의 일치함을 알 수 있다.

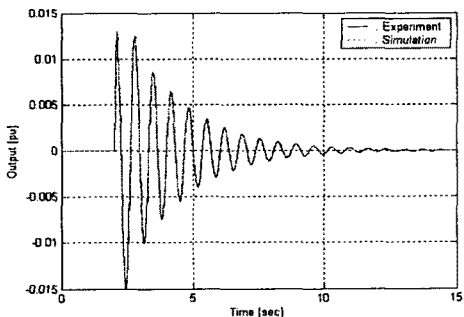


그림 7 측정 출력과 시뮬레이션 출력의 비교  
Fig. 7 The comparison of measurement output and simulation output

이상의 실험결과를 통해, 그림 5의 CAN 프로토콜을 이용한 네트워크 제어시스템은 제어대상 플랜트에 인가한 외란의 영향을 점점 감소시킬 수 있음을 확인할 수 있으며, 이로부터 본 논문에서 제안한 네트워크 제어시스템 설계방법이 효과적임을 입증할 수 있다.

#### 4. 결 론

본 논문에서는 CAN 프로토콜을 이용한 네트워크 제어시스템을 구축하였다. 이를 위해, 고가의 상용 CAN 통신장비나 제어보드를 이용하지 않고 직접 설계한 CAN 노드를 이용하여 센서, 구동기, 그리고 제어기 역할을 수행하는 네트워크 노드를 구성함으로써 경제성과 효율성을 높였다. 이후, 네트워크 제어시스템에서 발생하는 시간지연을 실험적으로 측정하였으며, 측정된 시간지연 하에서 전체 시스템의 안정도를 보장하는 샘플링 주기를 결정하였다. 마지막으로, 실험을 통해서 본 논문에서 구축한 CAN 프로토콜을 이용한 네트워크 제어시스템이 적용대상 플랜트인 발전계통에 대해서 효과적인 제어를 수행할 수 있음을 확인하였다. 향후, 본 논문에서 제안한 네트워크 제어시스템 설계방법은 다수의 입출력 장치를 갖는 대규모 제어계통이나 분산형 제어계통의 네트워크 제어시스템 설계에 적용할 수 있으리라 판단된다.

#### [참 고 문 헌]

- [1] Asok Ray, "Introduction to Networking for Integrated Control Systems," IEEE Control Sys. Mag., vol. 9, pp. 76-79, Jan. 1989.
- [2] R. S. Raji, "Smart Networks for Control," IEEE Spectrum, vol. 31, no. 6, pp. 49-55, June 1994.
- [3] Feng-Li Lian, James Moyne, and Dawn Tilbury, "Performance Evaluation of Control Networks: Ethernet, ControlNet, and DeviceNet," IEEE Control Sys. Mag., vol. 21, no. 1, pp. 66-83, Feb. 2001.
- [4] Wei Zhang, Michael S. Branicky, and Stephen M. Phillips, "Stability of Networked Control Systems," IEEE Control Sys. Mag., vol. 21, no. 1, pp. 84-99, Feb. 2001.
- [5] Feng-Li Lian, James Moyne, and Dawn Tilbury, "Network Design Consideration for Distributed Control Systems," IEEE Trans. Control Sys. Tech., vol. 10, no. 2, pp. 297-307, Mar. 2002.
- [6] 정준홍, 최수영, 박기현, "CAN을 이용한 발전계통의 제어 및 모니터링 기법 연구," 대한전기학회논문지, Vol. 52, No. 5, May, pp. 268-276, 2003.
- [7] BOSCH GmbH, CAN Specification Part A & Part B, 1991.
- [8] K. Tindell, A. Burns, and A. J. Wellings, "Calculating Controller Area Network(CAN) Message Response Times," Control Eng. Practice, vol. 3, no. 8, pp. 1163-1169, Aug. 1995.
- [9] RTI and RTI-MP Implementation Guide-For Release 3.4, dSPACE GmbH, May 2002.