

A Study on the Power System Control and Monitoring Technique Using CAN

丁俊弘* · 崔壽榮** · 朴基憲***
(Joonhong Jung, Sooyoung Choi, Kiheon Park)

Abstract - In this paper, we present a new control and monitoring technique for a power system using CAN(Controller Area Network). Feedback control systems having control loops closed through a network(i.e. Ethernet, ControlNet, CAN) are called NCSs(Networked Control Systems). The major problem of NCSs is the variation of stability property according to time delay including network-induced delay and computation delay in nodes. We present a new stability analysis method of NCSs with time delay exploiting a state-space model of LTI(Linear Time Invariant) interconnected systems. The proposed method can determine a proper sampling period of NCSs that preserves stability performance even in NCSs with a dynamic controller. We design CAN nodes which can transmit control and monitoring data through CAN bus and apply these to NCSs for a power system. The results of the experiment validate effectiveness of our control and monitoring technique for a power system.

Key Words : CAN(Controller Area Network), NCSs(Networked Control Systems), time delay, stability analysis

1. 서 론

발전계통과 같은 대규모 시스템을 효율적으로 운용하기 위해서는 분산형 제어시스템(Distributed Control System) 구축이 필수적이며, 여기에는 각각의 분산장치를 상호 연결, 제어 및 모니터링에 필요한 데이터를 실시간 전송하는데 필요한 통신수단이 요구된다. 기존 제어시스템에서 가장 널리 사용되었던 통신방식은 RS-232 기반의 일대일(Point to Point) 접속구조로 소규모 중앙집중형 시스템 구축에는 성공적으로 적용할 수 있었으나, 이를 분산형 제어시스템에 적용할 경우에는 각 장치들을 연결하기 위한 통신선로가 크게 증가하여 설치비용이 늘어날 뿐 아니라 전체 시스템을 복잡하게 만들게 되어, 결과적으로 시스템 유지정비 및 확장을 어렵게 만들 수 있다. 최근, 이 같은 문제점을 해결하기 위해 Ethernet, ControlNet, CAN(Controller Area Network) 등의 네트워크를 분산형 제어시스템 구축에 이용하는 연구가 활발히 진행 중에 있다[1]~[6].

네트워크 제어시스템(Networked Control Systems)은 피드백 제어경로가 공유매체(Common Bus)를 통해 구성된 시스템으로[3],[6]~[9], 센서, 제어기, 구동기, 그리고 모니터링 장치 등의 시스템 구성요소를 네트워크에 연결된 하나의 노

드(Node)로 간주하고, 이를 공유매체에 연결한 버스 또는 링 방식의 데이터 통신구조를 가지고 있다. 이는 기존의 일대일 접속방식에 비해 시스템 구성에 필요한 배선을 최소화할 수 있고, 플랜트 확장 시 기존 시스템과의 호환성을 향상시킬 뿐만 아니라, 시스템 유지, 보수를 용이하게 만드는 등의 장점을 가진다. 그러나, 공유매체를 이용하여 데이터를 전송하는 네트워크를 제어계에 도입하면서 네트워크 시간지연이나 데이터 전송오류 등의 네트워크 불확실성을 제어시스템 설계문제에 추가적으로 반영해야 하는 단점을 가지고 있다. 특히, 시간지연은 전체 시스템의 안정도를 변화시키므로, 네트워크 제어시스템 설계 시 가장 우선적으로 고려해야 할 대상이다[6]~[9].

CAN은 80년대 중반 차량 내부의 전자장치(Electronic Control Units)를 통합, 연결하기 위한 프로토콜로 개발된 이후, 로봇, 공정제어, 자동제어 등의 다양한 응용분야에 그 적용범위를 넓혀가고 있다. 특히, CAN은 저가의 하드웨어 장치로 효율성이 높은 네트워크를 구축할 수 있으며, 프레임 오버헤드가 작아 크기가 작은 주기적인 데이터 전송에 적합하고, 강력한 오류처리 기능을 내장하고 있어 전송 데이터의 신뢰성이 매우 높기 때문에 제어응용에 적합한 프로토콜로 평가받고 있다[2],[3].

본 논문에서는 CAN을 이용하여 발전계통을 제어하고 모니터링 할 수 있는 네트워크 제어시스템을 구축하였다. 먼저, 동기발전기 1기 무한대 모션 모델을 제어대상 플랜트로 선정하고 이를 안정화시키는 H_2 제어기를 설계하였다. 이후, 선형 시불변 결합시스템(LTI Interconnected System)의 상태방정식 모델에서 유도한 새로운 네트워크 제어시스템의 안정도 판별 조건을 제시하고, 이를 발전계통 네트워크 제어

* 正 會 員 : 成均館大 情報通信工學部 博士課程修了

** 正 會 員 : 成均館大 情報通信工學部 時間講師 · 工博

*** 正 會 員 : 成均館大 情報通信工學部 教授 · 工博

接受日字 : 2002年 12月 18日

最終完了 : 2003年 3月 19日

시스템의 안정도 분석에 적용하였으며, 이로부터 측정된 시간지연에서도 전체 시스템의 안정도를 유지할 수 있는 샘플링 주기를 결정하였다. 또한, 네트워크 제어시스템에서 제어기, 센서, 그리고 구동기 역할을 수행하는 CAN 노드를 직접 설계/제작하여 발전계통에 대한 실험용 네트워크 제어시스템을 구축하였다. 동시에, 각 노드에서 전송중인 데이터를 PC를 통해 확인할 수 있는 모니터 프로그램을 작성하여 현재 네트워크 제어시스템의 동작상태를 실시간 모니터링 할 수 있도록 하였다. 마지막으로, 본 논문에서 구축한 CAN을 이용한 네트워크 제어시스템이 발전계통에 대해 효율적인 제어 및 모니터링을 수행할 수 있음을 실험을 통해 입증하였다.

2. 발전계통 모델링 및 H_2 제어기 설계

본 논문에서 고려한 발전계통 모델은 동기발전기로서 전송선을 통해 무한대 모선에 연결된 1기 시스템이다. 이와 같은 모델은 그림 1과 같이 소신호 안정도 해석에 주로 사용되는 무한대 모선 블럭선도로 표현할 수 있다[10],[11].

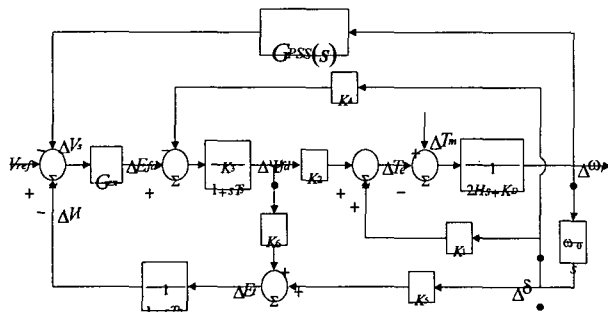


그림 1 동기발전기 1기 무한대 모선의 블럭선도
 Fig. 1 The block diagram of a synchronous generator connected to a infinite bus

그림 1에서 $\Delta\delta$ 는 로터의 각 변동, $\Delta\omega_r$ 은 동기 각속도의 변동, ΔT_m 은 기계적 토오크 입력의 변동, 그리고 ΔT_e 는 전기적 토오크 입력의 변동을 나타낸다. 또한 G_{ex} 와 $1/(1+sT_R)$ 로 표시된 블럭은 다이리스터 여자기 부분이며 (IEEE type ST1A)[10], $G_{PSS}(s)$ 는 전력계통 안정기(Power System Stabilizer)를 나타낸다.

위의 블럭선도에 대한 상태방정식을 유도하기 위해, 각각 상태변수 $x(t) = [\Delta\omega_r, \Delta\delta, \Delta\Psi_{fd}, \Delta V_1]^T$, 입력 $u(t) = \Delta V_s$, 외란 $\xi_d(t) = \Delta T_m$, 그리고 출력 $y(t) = \Delta\omega_r$ 을 설정한 후 정리하면,

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + F\xi_d(t) \\ y(t) &= Cx(t) \end{aligned} \tag{1}$$

$$\begin{aligned} \text{여기서, } A &= \begin{bmatrix} -\frac{K_D}{2H} & -\frac{K_1}{2H} & -\frac{K_2}{2H} & 0 \\ \omega_o & 0 & 0 & 0 \\ 0 & -\frac{K_3K_4}{T_3} & -\frac{1}{T_3} & -\frac{K_3G_{ex}}{T_3} \\ 0 & \frac{K_5}{T_R} & \frac{K_6}{T_R} & -\frac{1}{T_R} \end{bmatrix}, \\ B &= \begin{bmatrix} 0 \\ 0 \\ \frac{K_3G_{ex}}{T_3} \\ 0 \end{bmatrix}, F = \begin{bmatrix} \frac{1}{2H} \\ 0 \\ 0 \\ 0 \end{bmatrix}, C = [1 \ 0 \ 0 \ 0] \end{aligned}$$

본 논문에서 사용한 발전계통 모델의 동작점은 참고문헌 [10, pp843]의 선로상태와 동작조건을 이용하여 결정하였으며, 이 때 식(1)의 파라미터들은 아래와 같이 계산된다.

$$\begin{aligned} A &= \begin{bmatrix} 0 & -0.139 & -0.128 & 0 \\ 376.990 & 0 & 0 & 0 \\ 0 & -0.201 & -0.476 & -27.418 \\ 0 & -5.310 & 19.670 & -50.000 \end{bmatrix}, \\ B &= \begin{bmatrix} 0 \\ 0 \\ 27.418 \\ 0 \end{bmatrix}, F = \begin{bmatrix} 0.143 \\ 0 \\ 0 \\ 0 \end{bmatrix} \end{aligned} \tag{2}$$

위와 같이 표현된 발전계통 모델의 극점은 $0.39 \pm j7.77$, -17.78 , -33.48 이므로, 불안정한 시스템을 알 수 있다. 이러한 불안정한 발전계통을 안정화시키기 위해 본 논문에서는 H_2 제어기법[11]을 이용한 전력계통 안정기를 설계하였다. 설계된 H_2 제어기의 상태공간 모델은 식(3)과 같다.

$$G_{PSS}(s) = \left[\begin{array}{cccc|c} -4.638 & -0.139 & -0.128 & 0 & 4.638 \\ 376.990 & 0 & 0 & 0 & -0.000 \\ 892.383 & -6.715 & -8.368 & -23.551 & -24.252 \\ 7.519 & -5.310 & 19.670 & -50.000 & -7.519 \\ \hline -31.663 & 0.238 & 0.288 & -0.141 & 0 \end{array} \right] \tag{3}$$

3. 네트워크 제어시스템(Networked Control Systems)

3.1 시간지연을 갖는 네트워크 제어시스템

다음과 같은 선형 시불변 시스템을 고려하자.

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A} \mathbf{x}(t) + \mathbf{B} \mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C} \mathbf{x}(t) \end{aligned} \quad (4)$$

여기서, $\mathbf{x}(t) \in R^{n \times 1}$, $\mathbf{u}(t) \in R^{r \times 1}$, $\mathbf{y}(t) \in R^{m \times 1}$

식(4)의 시스템을 안정화시키는 이산제어기는 식(5)와 같은 동적 제어기의 형태를 갖는다.

$$\begin{aligned} \mathbf{x}_c((k+1)h) &= \mathbf{A}_c \mathbf{x}_c(kh) + \mathbf{B}_c \mathbf{u}_c(kh) \\ \mathbf{y}_c(kh) &= \mathbf{C}_c \mathbf{x}_c(kh) + \mathbf{D}_c \mathbf{u}_c(kh) \end{aligned} \quad (5)$$

여기서, $\mathbf{x}_c(kh) \in R^{p \times 1}$, $\mathbf{u}_c(kh) = \mathbf{y}(kh)$, $\mathbf{y}_c(kh) = -\mathbf{u}(kh)$

식(4), (5)로 표현된 플랜트와 이산제어기를 이용하여 네트워크 제어시스템을 구성할 경우, 플랜트, 센서, 구동기, 제어기, 그리고 네트워크를 포함하는 전체 시스템은 그림 2와 같이 나타낼 수 있다.

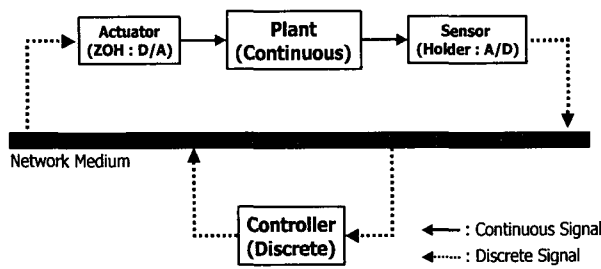


그림 2 일반적인 네트워크 제어시스템
Fig. 2 Typical networked control systems

위의 네트워크 제어시스템에서 한 샘플링 주기동안 발생하는 신호 흐름은 플랜트(출력)→센서(A/D)→네트워크→이산제어기→네트워크→구동장치(D/A)→플랜트(입력)와 같은 순서를 가진다. 플랜트의 출력을 미리 결정된 주기적인 시점에서 샘플링 한다면(Clock Driven System[8]), k 번째 샘플링 시점에서 측정된 출력이 네트워크와 제어기를 통해 플랜트의 제어입력으로 전달되는 시점은 그림 3과 같다.

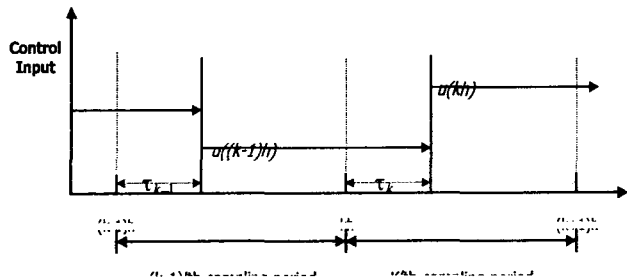


그림 3 시간지연을 고려한 제어입력의 타이밍 선도
Fig. 3 Timing diagram of control inputs with time delay

여기서, τ_k 는 k 번째 출력이 네트워크와 이산제어기를 통해 입력으로 전달될 때까지의 시간지연을 나타내며, 그 크기는 한 샘플링 구간보다 작다고 가정한다(즉, $0 \leq \tau_k \leq h$). k 번째 샘플링 구간에서 플랜트에 영향을 미치는 제어입력은 k 번째 제어 입력 $\mathbf{u}(kh)$ 와 시간지연으로 인해 추가적으로 나타난 $k-1$ 번째 제어입력 $\mathbf{u}((k-1)h)$ 이다. 따라서, 시간지연을 고려한 네트워크 제어시스템은 식(6)과 같이 입력지연을 갖는 이산시스템[12]으로 모델링 할 수 있다.

$$\begin{aligned} \mathbf{x}((k+1)h) &= \bar{\mathbf{A}} \mathbf{x}(kh) + \bar{\mathbf{B}}_0(\tau_k) \mathbf{u}(kh) + \bar{\mathbf{B}}_1(\tau_k) \mathbf{u}((k-1)h) \\ \mathbf{y}(kh) &= \mathbf{C} \mathbf{x}(kh) \end{aligned} \quad (6)$$

$$\begin{aligned} \text{여기서, } \bar{\mathbf{A}} &= e^{\mathbf{A}h}, \\ \bar{\mathbf{B}}_0(\tau_k) &= \int_0^{h-\tau_k} e^{\mathbf{A}s} \mathbf{B} ds, \\ \bar{\mathbf{B}}_1(\tau_k) &= \int_{h-\tau_k}^h e^{\mathbf{A}s} \mathbf{B} ds \end{aligned}$$

새로운 상태벡터를 $\bar{\mathbf{x}}(kh) \equiv [\mathbf{x}^T(kh) : \mathbf{u}^T((k-1)h)]^T$ 로 정의하면,

$$\begin{aligned} \bar{\mathbf{x}}((k+1)h) &= \begin{bmatrix} \bar{\mathbf{A}} & \bar{\mathbf{B}}_1(\tau_k) \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \bar{\mathbf{x}}(kh) + \begin{bmatrix} \bar{\mathbf{B}}_0(\tau_k) \\ \mathbf{I} \end{bmatrix} \mathbf{u}(kh) \\ \mathbf{y}(kh) &= [\mathbf{C} \quad \mathbf{0}] \bar{\mathbf{x}}(kh) \end{aligned} \quad (7)$$

식(7)로 변환된 네트워크 제어시스템은 시간지연에 따라 전체 시스템의 안정도가 변한다. 이를 해석하기 위해 Walsh 등은 센서노드에서 제어노드로 전송하는 데이터의 최대 전송주기를 MATI(Maximum Allowable Transfer Interval)라 정의하고, 네트워크 제어시스템의 안정도를 보장하는 MATI의 허용범위를 유도하였다[7]. Branicky 등은 네트워크 제어 시스템을 연속플랜트와 이산제어기가 혼합된(Hybrid) 시스템으로 간주하고, 전체 시스템의 안정도를 보장하는 시간지연과 샘플링 주기를 구할 수 있는 방법을 제시하였다[8],[9]. 그러나, 전자는 구해진 MATI가 네트워크 제어시스템에 적용할 수 있는 샘플링 주기보다 너무 작아서 실제 구현이 어렵다는 문제점을 가지고 있으며, 후자는 상태피드백 제어기를 사용하는 네트워크 제어시스템에만 적용할 수 있다는 제약조건을 지니고 있다. 따라서, 본 논문에서는 H_2 제어기와 같은 동적 제어기를 사용하는 네트워크 제어시스템에서도 시간지연에 따른 전체 시스템의 안정도 변화를 분석할 수 있는 새로운 방법을 제안한다.

3.2 선형 시불변 결합시스템

일반적인 선형 시불변 결합시스템은 그림 4와 같이 나타낼 수 있다[13].

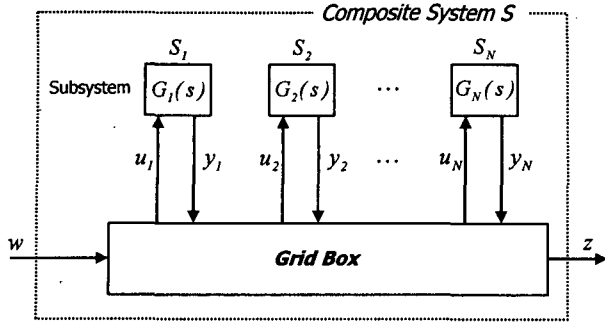


그림 4 선형 시불변 결합시스템
Fig. 4 LTI(Linear Time Invariant) interconnected systems

전체 시스템 S 는 $p \times 1$ 차의 입력 $w(t)$ 와 $q \times 1$ 차의 출력 $z(t)$ 를 가지며, N 개의 부 시스템 ($S_i, i=1, 2, \dots, N$)과 이들의 입출력 관계를 표현하는 Grid Box로 구성된다. 먼저, i 번째 부 시스템 S_i 를 식(8)과 같은 상태방정식으로 표현한다.

$$\begin{aligned} \dot{x}_i(t) &= A_i x_i(t) + B_i u_i(t) \\ y_i(t) &= C_i x_i(t) + D_i u_i(t) \end{aligned} \quad (8)$$

여기서, $x_i(t) \in R^{n_i \times 1}$, $u_i(t) \in R^{r_i \times 1}$, $y_i(t) \in R^{m_i \times 1}$

다음으로, 식(9)와 같이 모든 부 시스템의 상태, 제어, 그리고 출력벡터를 포함하는 새로운 벡터를 정의한다.

$$x_i(t) = \begin{bmatrix} x_{i1}(t) \\ x_{i2}(t) \\ \vdots \\ x_{in_i}(t) \end{bmatrix}, u_i(t) = \begin{bmatrix} u_{i1}(t) \\ u_{i2}(t) \\ \vdots \\ u_{in_i}(t) \end{bmatrix}, y_i(t) = \begin{bmatrix} y_{i1}(t) \\ y_{i2}(t) \\ \vdots \\ y_{im_i}(t) \end{bmatrix} \quad (9)$$

이제, 식(10)과 같은 행렬을 정의한다.

$$\left. \begin{aligned} A_i &= \text{diag}(A_i), & B_i &= \text{diag}(B_i) \\ C_i &= \text{diag}(C_i), & D_i &= \text{diag}(D_i) \end{aligned} \right\} i=1 \rightarrow N \quad (10)$$

식(8), (9), 그리고 식(10)으로부터,

$$\begin{aligned} \dot{x}_i(t) &= A_i x_i(t) + B_i u_i(t) \\ y_i(t) &= C_i x_i(t) + D_i u_i(t) \end{aligned} \quad (11)$$

여기서, $x_i(t) \in R^{n_i \times 1}$, $u_i(t) \in R^{r_i \times 1}$, $y_i(t) \in R^{m_i \times 1}$

$$n = \sum_{i=1}^N n_i, r = \sum_{i=1}^N r_i, m = \sum_{i=1}^N m_i$$

Grid Box는 전체 시스템 S 의 입력 $w(t)$ 와 출력 $z(t)$, 그리고 각 부 시스템의 입력 $u_i(t)$ 와 출력 $y_i(t)$ 를 상호 연결하는 선형조합이라 생각할 수 있다. 따라서,

$$\begin{bmatrix} u_i(t) \\ z(t) \end{bmatrix} = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} y_i(t) \\ w(t) \end{bmatrix} \quad (12)$$

식(11), (12)로부터,

$$u_i(t) = (I - M_{11} D_i)^{-1} (M_{11} C_i x_i(t) + M_{12} w(t)) \quad (13)$$

위에서 구한 $u_i(t)$ 를 식(11), (12)에 대입하고 정리하면, 입력 $w(t)$ 와 출력 $z(t)$ 에 대한 전체 시스템 S 의 동적 방정식을 아래와 같이 구할 수 있다.

$$\begin{aligned} \dot{x}_i(t) &= (A_i + B_i E_1^{-1} M_{11} C_i) x_i(t) + B_i E_1^{-1} M_{12} w(t) \\ z_i(t) &= M_{21} E_2^{-1} C_i x_i(t) + (M_{21} E_2^{-1} D_i M_{12} + M_{22}) w(t) \end{aligned} \quad (14)$$

여기서, $E_1 = I - M_{11} D_i$, $E_2 = I - D_i M_{11}$

식(14)에서 E_1^{-1} , E_2^{-1} 의 존재조건은 전체 시스템 S 의 well-posedness 조건이다.

정리 1 [13, Lemma 1] 그림 4의 전체 시스템 S 에 대한 상태방정식 파라미터는 식(15)와 같이 나타낼 수 있다.

$$G(s) = \left[\begin{array}{c|c} A_i + B_i E_1^{-1} M_{11} C_i & B_i E_1^{-1} M_{12} \\ \hline M_{21} E_2^{-1} C_i & M_{21} E_2^{-1} D_i M_{12} + M_{22} \end{array} \right] \quad (15)$$

증명 : 식(14)로부터 자명하다. ■

따라서, 그림 4의 전체 시스템 S 가 내부 안정하기 위한 필요충분조건은 시스템 행렬 $A_i + B_i E_1^{-1} M_{11} C_i$ 의 고유값이 모두 음의 실수부를 가져야 한다(만약, S_i 가 이산시스템이라면, 전체 시스템이 내부안정하기 위한 필요충분조건은 시스템 행렬의 모든 고유값이 단위원 내에 존재해야 한다).

3.3 네트워크 제어시스템의 안정도 분석

동적 제어를 포함하는 네트워크 제어시스템의 안정도를 분석하기 위해 그림 2의 네트워크 제어시스템을 3.2절에서 고려한 선형 시불변 결합시스템으로 변형한다.

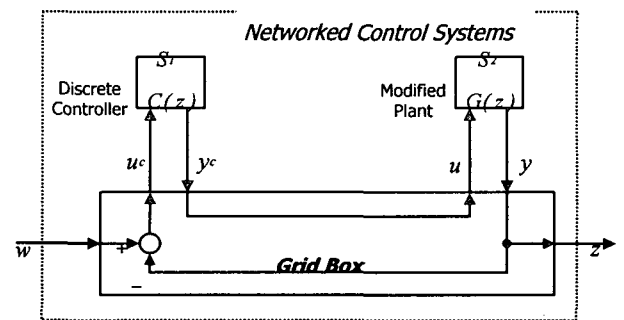


그림 5 변형된 네트워크 제어시스템
Fig. 5 Modified networked control systems

식(5), (7)로부터, 그림 5의 부 시스템 S_1, S_2 는

$$C(z) = \begin{bmatrix} A_c & B_c \\ C_c & D_c \end{bmatrix}, G(z) = \begin{bmatrix} \overline{A} & \overline{B}_1(\tau_k) & \overline{B}_0(\tau_k) \\ 0 & 0 & I \\ C & 0 & 0 \end{bmatrix} \quad (16)$$

다음으로, 각 부 시스템의 입력과 출력을 모두 포함하는 전체 시스템의 상태, 제어, 그리고 출력벡터를 정의한다.

$$\mathbf{x}_t(kh) = \begin{bmatrix} \mathbf{x}_c(kh) \\ \mathbf{x}(kh) \end{bmatrix}, \mathbf{u}_t(kh) = \begin{bmatrix} \mathbf{u}_c(kh) \\ \mathbf{u}(kh) \end{bmatrix}, \mathbf{y}_t(kh) = \begin{bmatrix} \mathbf{y}_c(kh) \\ \mathbf{y}(kh) \end{bmatrix} \quad (17)$$

여기서, $\mathbf{x}_t(kh) \in R^{(n+r+p) \times 1}$, $\mathbf{u}_t(kh) \in R^{(m+r) \times 1}$,
 $\mathbf{y}_t(kh) \in R^{(m+r) \times 1}$

식(11)로부터,

$$\begin{aligned} \mathbf{x}_t((k+1)h) &= A_t \mathbf{x}_t(kh) + B_t \mathbf{u}_t(kh) \\ \mathbf{y}_t(kh) &= C_t \mathbf{x}_t(kh) + D_t \mathbf{u}_t(kh) \end{aligned} \quad (18)$$

여기서,

$$A_t = \begin{bmatrix} A_c & 0 & 0 \\ 0 & \overline{A} & \overline{B}_1(\tau_k) \\ 0 & 0 & 0 \end{bmatrix}, B_t = \begin{bmatrix} B_c & 0 \\ 0 & \overline{B}_0(\tau_k) \\ 0 & I \end{bmatrix},$$

$$C_t = \begin{bmatrix} C_c & 0 & 0 \\ 0 & C & 0 \end{bmatrix}, D_t = \begin{bmatrix} D_c & 0 \\ 0 & 0 \end{bmatrix}$$

Grid Box는 식(12)와 그림 5의 연결관계를 이용하면,

$$\begin{bmatrix} \mathbf{u}_c(kh) \\ \mathbf{u}(kh) \\ \mathbf{z}(kh) \end{bmatrix} = \begin{bmatrix} 0 & -I & I \\ I & 0 & 0 \\ 0 & I & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y}_c(kh) \\ \mathbf{y}(kh) \\ \mathbf{w}(kh) \end{bmatrix} \quad (19)$$

네트워크 제어시스템의 well-posedness 조건은 식(14)로부터 아래와 같이 결정할 수 있다.

$$E_1 = \begin{bmatrix} I & 0 \\ -D_c & I \end{bmatrix}, E_2 = \begin{bmatrix} I & D_c \\ 0 & I \end{bmatrix} \quad (20)$$

여기서, E_1, E_2 의 행렬식은 모두 1이므로 역행렬이 존재하고, 이로부터 그림 5의 네트워크 제어시스템은 3.2절의 정리 1을 통해서 상태방정식 파라미터를 구할 수 있음을 확인할 수 있다.

정리 2 모든 k 에 대해서 τ_k 가 상수 ($\tau = \tau_k, \forall k$)라면, 그림 5의 네트워크 제어시스템이 내부 안정하기 위한 필요충

분조건은,

$$|\lambda_i(\tilde{A})| < 1, \forall i, i=1,2,\dots,n+r+p \quad (21)$$

$$\tilde{A} = \begin{bmatrix} A_c & -B_c C & 0 \\ \overline{B}_0(\tau) C_c & \overline{A} - \overline{B}_0(\tau) D_c C & \overline{B}_1(\tau) \\ C_c & -D_c C & 0 \end{bmatrix}$$

증명 : 만약 시간지연이 상수라면, 식(16)의 플랜트는 선형 시불변 이산 시스템이 된다. 이 때, 네트워크 제어시스템의 시스템 행렬은 정리 1로부터,

$$\begin{aligned} \tilde{A} &= A_t + B_t E_1^{-1} M_{11} C_t \\ &= \begin{bmatrix} A_c & 0 & 0 \\ 0 & \overline{A} & \overline{B}_1(\tau) \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} B_c & 0 \\ 0 & \overline{B}_0(\tau) \\ 0 & I \end{bmatrix} \\ &\quad \cdot \begin{bmatrix} I & 0 \\ -D_c & I \end{bmatrix}^{-1} \begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix} \begin{bmatrix} C_c & 0 & 0 \\ 0 & C & 0 \end{bmatrix} \\ &= \begin{bmatrix} A_c & 0 & 0 \\ 0 & \overline{A} & \overline{B}_1(\tau) \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & -B_c C & 0 \\ \overline{B}_0(\tau) C_c & \overline{A} - \overline{B}_0(\tau) D_c C & 0 \\ C_c & -D_c C & 0 \end{bmatrix} \\ &= \begin{bmatrix} A_c & -B_c C & 0 \\ \overline{B}_0(\tau) C_c & \overline{A} - \overline{B}_0(\tau) D_c C & \overline{B}_1(\tau) \\ C_c & -D_c C & 0 \end{bmatrix} \end{aligned} \quad (22)$$

선형 시불변 이산 시스템이 내부 안정하기 위한 필요충분 조건은 시스템 행렬의 모든 고유값이 단위원 내에 존재해야 한다. 따라서, 그림 5로 나타낸 네트워크 제어시스템이 내부 안정하기 위한 필요충분조건은 식(22)의 모든 고유값이 단위원 내에 있어야 하므로, 식(21)이 성립해야 한다. ■

4. CAN을 이용한 발전계통의 네트워크 제어시스템

본 논문에서는 DSP 장비[14]로 모의 구현한 식(1)의 발전계통을 대상으로, CAN을 이용한 네트워크 제어시스템을 구축하였다. 여기서, 발전계통의 입력, 출력, 상태, 그리고 제어 및 모니터링을 위한 모든 데이터는 각 노드를 통해 CAN 버스에 연결된다. 전체 시스템의 구성도는 그림 6과 같다.

그림 6에서 센서노드(Sensor Node)는 샘플링 주기를 기준으로 DSP 출력단자(발전계통의 상태 및 출력)의 연속신호를 CAN 프로토콜로 변환하여(AD변환) 제어노드에 전송한다. 제어노드(Controller Node)는 센서노드에서 전송하는 데이터를 이용하여 제어입력을 계산한 후 구동노드에 전송하며, 동시에 한 주기동안 각 노드에서 송수신한 모든 데이터를 PC에 전송한다. 구동노드(Actuator Node)는 제어노드에서 전송하는 입력데이터를 연속신호로 변환하여(DA변환)

DSP의 입력단자(발전계통의 입력)에 인가하는 역할을 수행한다.

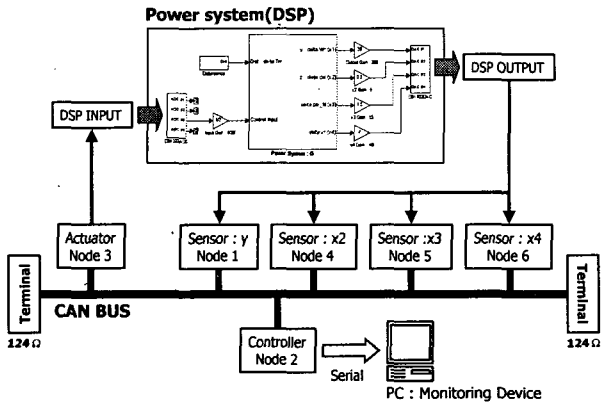


그림 6 CAN을 이용한 발전계통의 네트워크 제어시스템
Fig. 6 Networked control systems of power system using CAN

이러한 센서, 구동, 그리고 제어노드의 역할을 수행할 수 있도록 본 논문에서 설계한 CAN 노드는 그림 7의 구조를 갖는다. 여기서, CAN 전송기(CAN Transceiver)는 Philips PCA82C250, CAN 제어기(CAN Controller)는 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, 노드와 PC 사이의 직렬통신에는 Maxim MAX233 RS-232 드라이버를 사용하였다.

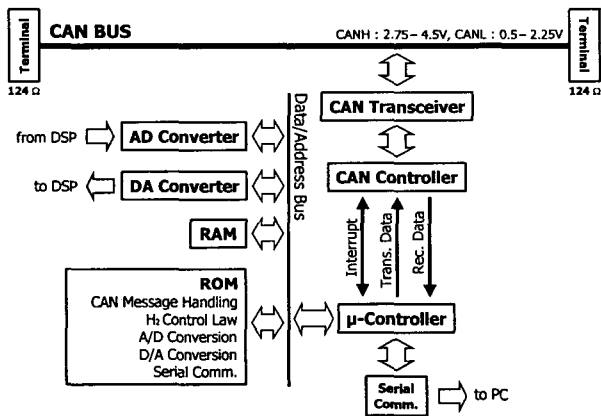


그림 7 CAN 노드의 구조
Fig. 7 The configuration of CAN node

그림 8은 실제 제작된 CAN 노드의 사진이다.

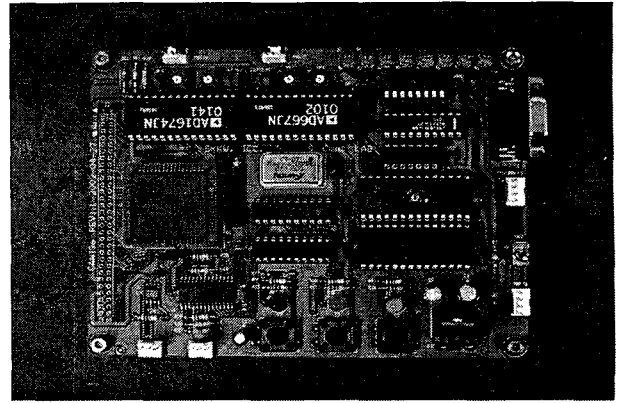


그림 8 CAN 노드의 사진
Fig. 8 The picture of CAN node

3장에서 살펴본 바와 같이, 네트워크 제어시스템에서 가장 먼저 고려해야 할 사항은 시간지연에 따른 전체 시스템의 안정도 변화이다. 네트워크 제어시스템에서 발생하는 시간지연은 노드를 구성하고 있는 하드웨어의 종류와 성능, 네트워크 프로토콜, 그리고 응용계통의 프로그램에 따라 크게 달라진다[2],[6]. 따라서, 본 논문에서는 그림 6과 같이 구성된 네트워크 제어시스템의 시간지연을 측정하기 위해 각 노드의 마이크로컨트롤러에 내장된 타이머 기능을 이용하였다. 즉, 전체 시스템이 매 주기마다 그림 9의 흐름선도와 같이 발전계통에 대한 제어를 수행하도록 설계할 때, 한 주기 동안 발생하는 시간지연은 각 노드에서 측정된 타이머 값인 T_i 를 이용하여 식(23)과 같이 계산할 수 있다.

$$\tau = T_{cp, delay} + T_{trans, delay} = T_5 - T_0 \quad (23)$$

여기서, $T_{cp, delay} = T_{cp, sensor} + T_{cp, controller} + T_{cp, actuator}$

$$T_{trans, delay} = T_{trans, sc} + T_{trans, ca}$$

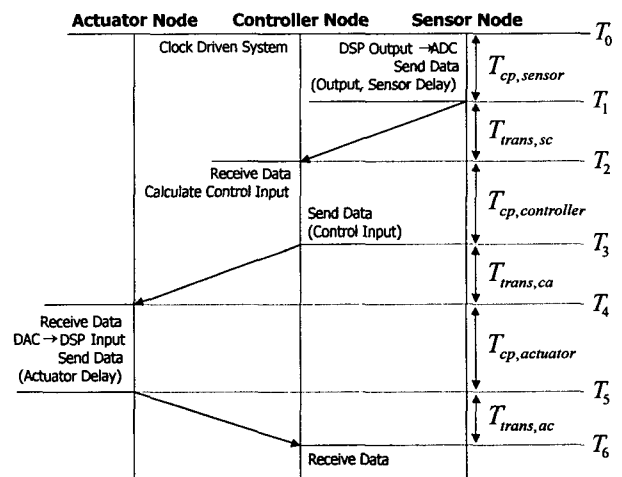


그림 9 설계된 네트워크 제어시스템의 제어흐름선도
Fig. 9 Control flow chart of designed networked control systems

실험용 발전계통 네트워크 제어시스템의 시간지연을 약 35분 동안 측정한 결과, 평균시간지연은 15.99[msec], 최대 시간지연은 16.14[msec] 이었다. 즉, 그림 6의 발전계통 네트워크 제어시스템은 최대시간지연인 16.14[msec] 하에서도 전체 시스템의 안정도를 유지하도록 설계해야 함을 알 수 있다.

5. 시뮬레이션

2장에서 구한 식(1)의 발전계통 모델과 식(3)의 발전계통 안정기를 네트워크 제어시스템으로 구성할 경우, 시간지연과 샘플링 주기에 따른 전체 시스템의 안정도 변화를 분석하기 위해 정리 2를 적용하여 시뮬레이션 하였다.

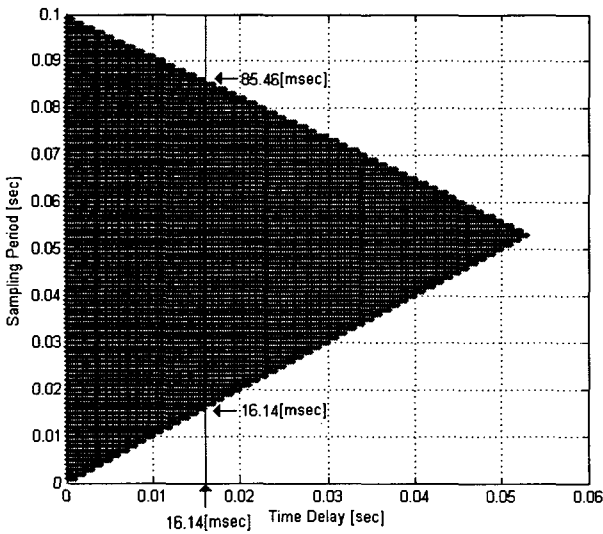


그림 10 발전계통의 안정영역
Fig. 10 Stability region of power system

그림 10에서 x축은 시간지연을 나타내고, y축은 샘플링 주기를 나타내며, *로 표시된 영역은 전체 시스템의 안정도가 보장되는 영역이다. 위의 시뮬레이션 결과로부터, 시간지연이 $\tau=16.14[msec]$ 일 때(4장에서 측정한 최대시간지연), 네트워크 제어시스템의 안정도를 보장하는 샘플링 주기는,

$$16.14[msec] \leq h \leq 85.46[msec] \quad (24)$$

식(24)에서 샘플링 주기의 하한은 최대시간지연과 같은데, 이는 3.1절의 시간지연에 대한 가정 즉, 샘플링 주기가 시간지연보다 크다는 가정에서 비롯된 조건이다. 본 논문에서는 식(24)를 만족하도록 전체 네트워크 제어시스템의 샘플링 주기를 $h=60[msec]$ 로 정하였으며, 이 경우 안정도를 보장하는 최대시간지연은 $\tau \leq 45.73[msec]$ 이 된다.

그림 11은 샘플링 주기가 $h=60[msec]$ 일 때, 발전계통에 외란 $\xi(t)=1[pu]$ ($t \in [2, 2.1][sec]$)을 인가한 경우, 시간지연을 변화시키면서 발전계통의 출력을 비교한 결과이다.

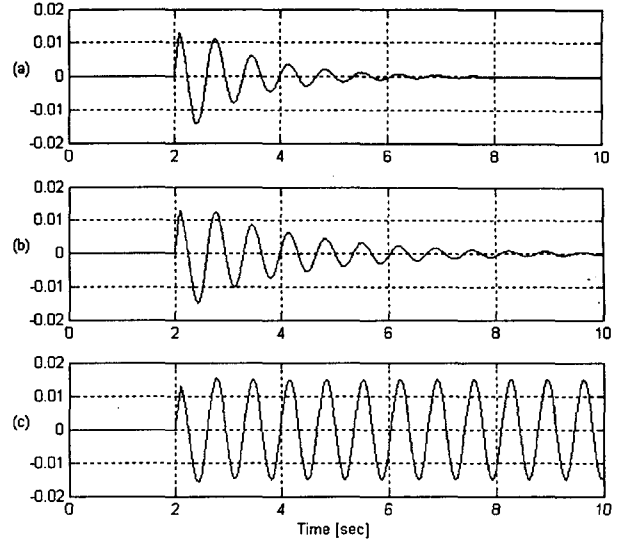


그림 11 시간지연에 따른 발전계통의 출력비교
(a) $\tau=0$, (b) $\tau=16.14$, (c) $\tau=45.73[msec]$
Fig. 11 Comparison of different power system outputs with (a) $\tau=0$, (b) $\tau=16.14$, (c) $\tau=45.73[msec]$

그림 11에서 시간지연이 증가함에 따라 발전계통 네트워크 제어시스템은 외란에 대한 영향을 감쇠 시키지 못해 점점 불안정해 지는 것을 확인할 수 있으며, 시간지연이 안정도를 보장하는 최대 한계에 이를 때는($\tau=45.73[msec]$) 출력 또한 임계 진동함을 알 수 있다.

6. 실험 및 결과분석

CAN을 이용한 발전계통의 네트워크 제어시스템을 실제 구현한 사진은 그림 12와 같다. 여기서, 식(1)의 발전계통은 dSPACE DS1102[14]를 이용하여 구현하였다.

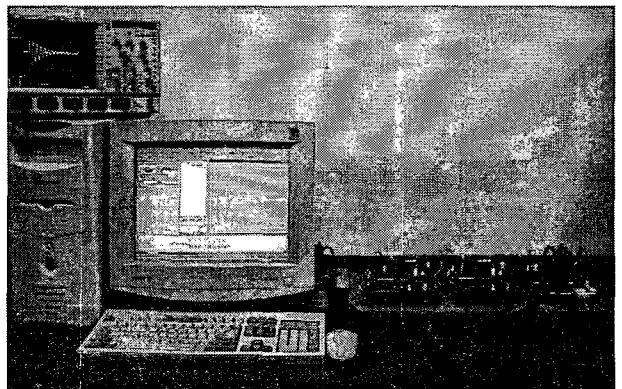


그림 12 실험장치 구성
Fig. 12 Experimental setup

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 13은 발전계통에 크기 $\xi_d(t)=1$ [pu]의 외란을 0.1[sec]동안 인가하였을 때 측정된 출력 파형이다.

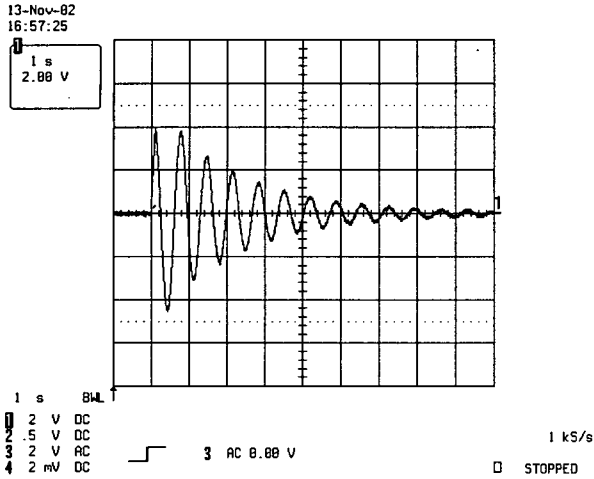


그림 13 측정 출력(크기×300)
Fig. 13 Measurement output(scale×300)

그림 14는 이 때의 측정 출력과 시간지연이 16.14[msec] 일 때의 시뮬레이션 출력을 비교한 것이며, 이들의 평균자승 오차는 식(25)와 같이 계산할 수 있다. 이로부터, 실험에서 측정된 값이 이론적인 출력 값과 거의 일치함을 알 수 있다.

$$MSE = \frac{1}{m+1} \sqrt{\sum_{i=0}^m \left\{ y_{exp}\left(i \frac{t_f}{m}\right) - y_{sim}\left(i \frac{t_f}{m}\right) \right\}^2}$$

$$\approx 4.754 \times 10^{-6} \quad (25)$$

여기서, $m=3000$, $t_f=15$

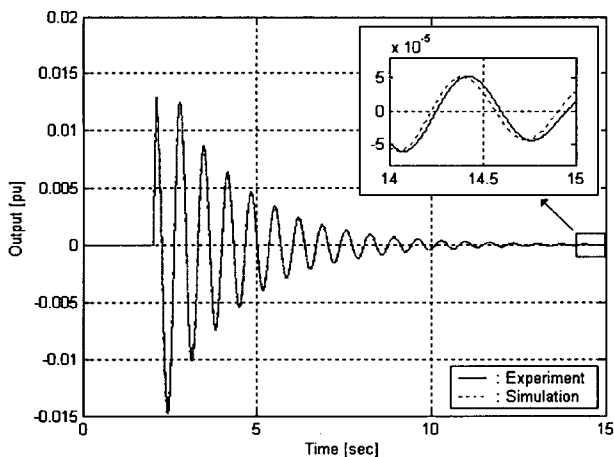


그림 14 측정 출력과 시뮬레이션 출력의 비교
Fig. 14 The comparison of measurement output and simulation output

그림 15는 각 노드에서 전송중인 데이터를 실시간 확인하기 위해 작성한 모니터링 프로그램의 실행화면이다. 여기서, 좌측 그래프는 시스템 출력, 제어입력, 그리고 상태 값을 실시간 출력하는 부분이고, 우측상단의 텍스트패널은 제어노드의 동작 상태 및 네트워크 제어시스템의 시간지연을 출력하는 부분이다. 또한, 우측하단은 직렬통신 설정 및 실시간 출력 옵션을 조절할 수 있는 제어패널이다.

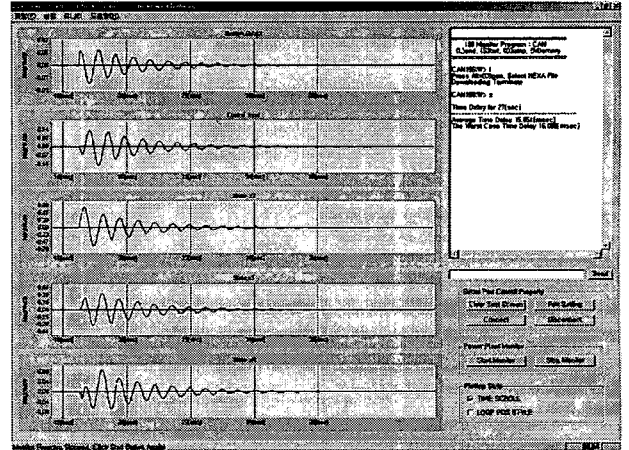


그림 15 모니터링 화면
Fig. 15 Monitoring screen

이상의 실험결과를 통해, 본 논문에서 구축한 CAN을 이용한 발전계통의 제어 및 모니터링 시스템이 만족할 만한 성능을 가지고 있음을 확인할 수 있다.

7. 결 론

본 논문에서는 CAN을 이용하여 발전계통을 제어하고 모니터링 할 수 있는 네트워크 제어시스템을 구축하였다. 네트워크 제어시스템을 설계하기 위해서는 시간지연에 따른 전체 시스템의 안정도 변화를 분석하는 것이 필수적이다. 기존[8],[9]에 제안된 네트워크 제어 시스템의 안정도 분석방법은 상태피드백 제어를 사용하는 네트워크 제어시스템에만 적용할 수 있다는 제약조건을 지니고 있으며, 발전계통을 안정화하기 위해 사용한 H_2 제어기는 동적 구조를 가지고 있으므로 위의 방법을 적용할 수 없다. 따라서, 본 논문에서는 동적 제어를 포함하는 네트워크 제어시스템에서도 시간지연에 따른 전체 시스템의 안정도 변화를 분석할 수 있는 새로운 방법을 제안하였다. 제안한 안정도 판별방법은 선형 시불변 결합시스템의 내부 안정도 등가조건을 이용하여 증명할 수 있으며, 측정된 시간지연 하에서 전체 시스템의 안정도를 보장하는 샘플링 주기를 결정할 수 있음을 입증하였다. 또한, 센서, 구동기, 그리고 제어기 역할을 수행하는 네트워크 노드를 구성하기 위해 고가의 상용 CAN 통신 장비나 제어보드를 이용하지 않고 직접 설계한 CAN 노드를 이용함으로써 경제성과 효율성을 높였으며, 각 노드에서 전송하는 제어 데이터를 PC를 통해 확인할 수 있는 모니터링 프로그램을 작성하여 네트워크 제어시스템의 동작 상태

를 실시간 감시할 수 있도록 하였다. 마지막으로, 구축한 CAN을 이용한 네트워크 제어시스템이 발전계통에 대한 효과적인 제어 및 모니터링을 수행할 수 있음을 실험을 통하여 확인하였다. 향후, 본 연구에서 개발된 방법을 실제 발전계통의 제어 및 감시 시스템에 적용한다면 기존의 일대일 접속 방식에 비해 필요한 배선을 대폭 줄일 수 있으므로, 시스템을 유지, 보수하는 시간 및 비용을 절감할 수 있고 고장이 발생했을 때 이를 신속히 검출함으로써 발전계통 운용의 신뢰성을 높이는 데 기여할 수 있으리라 기대된다. 더불어 본 논문에서 제안한 CAN을 이용한 발전계통의 제어 및 모니터링 기법은 일반적인 네트워크 제어시스템 설계 시 참조할 수 있는 실제적인 응용사례가 될 수 있을 것이다.

감사의 글

본 연구는 산업자원부의 지원에 의하여 기초전력공학 공동연구소 주관으로 수행된 과제임.

참고 문헌

[1] BOSCH GmbH, CAN Specification Part A & Part B, 1991.
 [2] 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.
 [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] Asok Ray, "Introduction to Networking for Integrated Control Systems," IEEE Control Sys. Mag., vol. 9, pp. 76-79, Jan. 1989.
 [5] R. S. Raji, "Smart Networks for Control," IEEE Spectrum, vol. 31, no. 6, pp. 49-55, June 1994.
 [6] 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.
 [7] Gregory C. Walsh, Hong Ye, and Linda G. Bushnell, "Stability Analysis of Networked Control Systems," IEEE Trans. Control Sys. Tech., vol. 10, no. 3, pp. 438-446, May 2002.
 [8] Michael S. Branicky, Stephen M. Phillips, and Wei Zhang, "Stability of Networked Control Systems : Explicit Analysis of Delay," Proc. Amer. Control Conf., Chicago, IL, pp. 2352-2357, June 2000.
 [9] 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.

[10] P. Kunder, Power System Stability and Control, NY: McGraw-Hill, 1993.
 [11] 민덕기, 조용석, 박기현, " H_2 와 H_∞ 제어기법을 이용한 전력계통 안정기의 설계," 전기학회논문집, Vol. 45, No. 5, pp 710-718, 1996.
 [12] Karl J. Åström and B. Wittenmark, Computer-Controlled Systems:Theory and Design, 3rd ed. Upper Saddle River, NJ: Prentice-Hall, 1997.
 [13] Kiheon Park, "On the Equivalence Condition of Internal and External Stability of Linear Time-invariant Interconnected Systems," KIEE, vol. 9, no. 3, pp. 166-170, 1996.
 [14] RTI and RTI-MP Implementation Guide-For Release 3.4, dSPACE GmbH, May 2002.

저 자 소 개



정 준 홍(丁 俊 弘)

1975년 1월 14일생. 1996년 성균관대학교 공대 전기공학과 졸업. 1998년 동 대학원 전기공학과 졸업(석사). 1998년~현재 동 대학원 정보통신공학부 박사과정수료.
 Tel : (031)290-7175, Fax : (031)290-7165
 E-mail : nelcast@mail.skku.ac.kr



최 수 영(崔 壽 榮)

1969년 9월 21일생. 1997년 성균관대학교 공대 전기공학과 졸업. 1999년 동 대학원 전기전자 및 컴퓨터공학과 졸업(석사). 2003년 동 대학원 정보통신공학부 졸업(공학박). 현재 동 대학교 정보통신공학부 시간강사.
 Tel : (031)290-7175, Fax : (031)290-7165
 E-mail : sychoi@sorak.skku.ac.kr



박 기 현(朴 基 憲)

1955년 7월 5일 생. 1978년 서울대학교 공대 전기공학과 졸업. 1980년 동 대학원 전기공학과 졸업(석사). 1987년 Polytechnic Univ. 졸업(공학박). 1990년~현재 성균관대학교 정보통신공학부 교수
 Tel : (031)290-7116, Fax : (031)290-7165
 E-mail : khpark@yurim.skku.ac.kr