

논문 2007-44SC-5-3

# 다중 패킷을 전송하는 네트워크 제어시스템의 안정성 분석

(Stability Analysis of a Networked Control System  
with Multiple Packet Transmission)

정준홍\*, 박기현\*\*, 이재호\*\*

(Joonhong Jung, Kiheon Park, and Jaeho Lee)

## 요약

본 논문에서는 다중 패킷을 전송하는 네트워크 제어시스템을 대상으로, 전체 시스템의 안정성 변화를 분석할 수 있는 새로운 방법을 제안하였다. 먼저, 메시지의 최대 시간 지연을 보장할 수 있는 스케줄링 방식과 데이터 손실을 나타낼 수 있는 이산 스위치 상태방정식을 새롭게 제시하였으며, 이를 이용하여 네트워크 제어시스템과 등가인 비동기 시스템(asynchronous dynamical system)을 유도하였다. 이후, 제어기의 샘플링 주기, 시간 지연, 데이터 전송 성공률, 전송 패킷의 개수에 따라 변화하는 네트워크 제어시스템의 안정성을 판별할 수 있도록 새로운 정리들을 제안하였다. 마지막으로, Batch Reactor 시스템을 대상으로 제안한 정리들을 적용하여 시뮬레이션하고 그 결과를 분석함으로써 본 논문에서 제안한 안정성 분석 방법의 타당성을 입증하였다.

## Abstract

The main objective of this paper is to propose a new stability analysis method for a networked control system with multiple packet transmission. The new scheduling method that can guarantee the maximum time delay and discrete switch state equation model which represent a network data loss is proposed. The equivalent model of a MIMO(multi-input multi-output) networked control system is derived from a state space model of linear time invariant interconnected systems in the form of asynchronous dynamical system. Using this model, this paper presents new stability theorems that can determine stability of the networked control system with regard to time delay, data loss, and the number of transmission packets. Simulation results verify the effectiveness of proposed stability analysis method.

**Keywords:** networked control system, multiple packet transmission, switch model, stability analysis

## I. 서론

제어공학 분야에 네트워크를 도입하려는 연구는 지난 7~8년 전부터 매우 활발하게 진행되고 있으며, 그

결과 네트워크 제어시스템(networked control system)이라는 새로운 연구 분야에 관심이 집중되고 있다. 네트워크 제어시스템은 “폐환 경로(feedback loop)가 네트워크를 통해 구성된 제어시스템”으로 정의할 수 있으며, 기존 제어시스템 구축에 필요했던 다량의 배선을 공유 매체(common medium)로 통합, 시스템의 부피와 무게를 획기적으로 감소시키고 유지·보수에 필요한 시간 및 비용을 절감할 수 있을 뿐 아니라 유연한 확장성을 제공하는 등의 장점을 지닌다<sup>1-2)</sup>.

일반적인 제어시스템과 마찬가지로 네트워크 제어시스템의 설계 및 분석에서 중요한 고려 사항은 전체 시스템의 안정성 문제이다. 네트워크 제어시스템의 안정

\* 정희원, 한전KDN(주) 전력IT연구원

(Research and Development Department, Korea Electric Power Data Network Co., Ltd.)

\*\* 정희원, 성균관대학교 정보통신공학부

(School of Information and Communication Engineering, Sungkyunkwan University)

※ 본 연구는 산업자원부 전력IT 정부주도 기술개발사업(R-2005-1-402)의 지원으로 수행되었음.

접수일자: 2007년1월4일, 수정완료일: 2007년9월7일

성은 네트워크에서 발생하는 불확실성의 영향을 받으며, 이에 대한 연구 분야는 크게 시간 지연을 고려한 안정성 분석과<sup>[3~8, 10~11]</sup> 데이터 손실을 고려한 안정성 분석이<sup>[9~11]</sup> 있다. 전자는 시간 지연의 특성(시불변, 시변, 랜덤 등)이나 길이를 기준으로 하여 안정성 변화를 분석하고 후자는 데이터 손실의 모델링이나 처리 방식 등에 따라 안정성 변화를 분석한다. 또한, 네트워크 제어 시스템의 구성 방식, 메시지 전송 방식 등에 따른 다양한 연구 결과가 있으며, 시간 지연과 데이터 손실을 동시에 고려한 안정성 분석<sup>[10~11]</sup> 연구도 진행 중이다.

상기한 네트워크 제어 시스템의 안정성 분석 방법은 각각 장단점을 지니고 있다<sup>[2, 4, 11]</sup>. 먼저, 참고문헌 [3]~[8]에서는 시간 지연만을 고려했을 때, 이와 반대로 참고문헌 [9]는 데이터 손실만을 고려했을 때 네트워크 제어 시스템의 안정성 판별이 가능하다. 참고문헌 [10], [11]은 시간 지연과 데이터 손실을 동시에 고려할 수 있는 안정성 분석법이나 단일 패킷을 전송하는 네트워크 제어 시스템 구조에만 적용할 수 있기 때문에 일반적인 다입력 다출력(multi-input multi-output:MIMO) 시스템에 적용하는데 무리가 따른다. 즉, 현재까지의 연구 결과로는 다중 패킷을 전송하는 MIMO 네트워크 제어 시스템에 대하여 시간 지연과 데이터 손실을 모두 고려한 안정성 분석이 어렵다.

본 논문에서는 단일/다중 패킷을 전송하는 네트워크 제어 시스템을 새로운 스케줄링 방식과 이산 스위치 방정식을 이용하여 모델링하고 이를 통해 전체 시스템의 안정성 변화를 판별할 수 있는 방법을 제시한다. 먼저, MIMO 네트워크 제어 시스템의 구조, 시간 지연, 데이터 손실 등을 정의하고 전송 데이터의 최대 지연 시간을 보장할 수 있는 새로운 스케줄링 방식을 제안한다. 이후, 데이터 손실을 새로운 이산 스위치 상태방정식으로 표현하여 네트워크 제어 시스템에 대한 비동기 시스템(asynchronous dynamical system)<sup>[9~11]</sup> 등가 모델을 유도한다. 그리고 단일/다중 패킷 전송시 네트워크 제어 시스템의 안정성 변화를 판별할 수 있는 정리들을 제안한다. 마지막으로, 제안된 정리들을 입출력이 2차인 Batch Reactor 시스템에 적용하여 시물레이션하고 그 결과를 분석한다.

## II. 네트워크 제어 시스템의 스케줄링과 모델링

네트워크 제어 시스템의 성능이나 안정성을 정확히 판별하려면 네트워크의 영향을 추가적으로 반영할 수

있는 합리적인 제어 시스템 모델을 수립하는 과정이 선행되어야 한다. 특히, 네트워크에서 발생할 수 있는 가장 큰 불확실 요소인 시간 지연과 데이터 손실은 네트워크 제어 시스템의 안정성에 큰 영향을 미칠 수 있으므로 이를 효과적으로 반영할 수 있는 모델을 개발하는 것이 무엇보다 중요하다. 본 장에서는 네트워크 제어 시스템의 일반적인 구조, MIMO 시스템에서 단일/다중 패킷을 전송할 때의 시간 지연과 데이터 손실, 데이터의 최대 지연 시간을 보장할 수 있는 메시지 스케줄링 기법을 논한다. 그리고 네트워크 제어 시스템의 시간 지연을 플랜트와 제어기의 지연 입력으로 나타내고 데이터 손실을 새로운 형태의 이산 스위치 상태방정식으로 모델링한다.

### 1. MIMO 네트워크 제어 시스템의 스케줄링 기법

다음과 같은 네트워크 제어 시스템을 고려하자<sup>[2]</sup>.

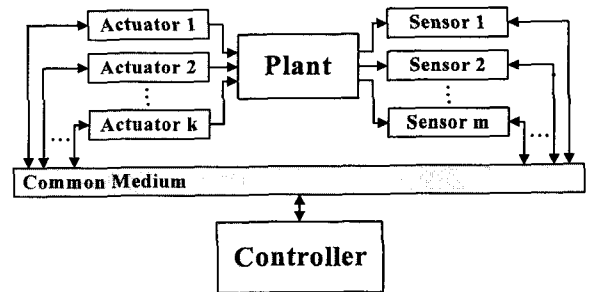


그림 1. 일반적인 MIMO 네트워크 제어 시스템  
Fig. 1. A typical networked control system with multi-input multi-output.

그림 1은 플랜트 입출력과 제어기가 공유 매체를 통해 연결된 MIMO 네트워크 제어 시스템의 구조를 나타낸다. 여기서, 제어 대상인 플랜트는 식(1)과 같은 선형 시불변 시스템이고

$$\begin{aligned} \dot{x}_p(t) &= A_p x_p(t) + B_p u(t) \\ y(t) &= C_p x_p(t) \end{aligned} \tag{1}$$

여기서,  $x_p(t) \in R^n$ ,  $u(t) \in R^k$ ,  $y(t) \in R^m$

이를 안정화시키는 제어기는 식(2)와 같은 동적 제어기 형태라 가정하자<sup>[11]</sup>.

$$\begin{aligned} \dot{x}_c(t) &= A_c x_c(t) + B_c u_c(t) \\ y_c(t) &= C_c x_c(t) + D_c u_c(t) \end{aligned} \tag{2}$$

여기서,  $x_c(t) \in \mathbb{R}^p$ ,  $y_c(t) = u(t)$ ,

$u_c(t) = w(t) - y(t)$ ,  $w(t)$  : 외부 입력

먼저,  $m$ 개의 센서에서 측정된 데이터가 단일 패킷을 통해 제어기로 전달되고, 제어기에서 계산된 입력 데이터가 단일 패킷을 통해  $k$ 개의 구동기에 전달되는 경우를 고려하자. 이러한 네트워크 제어시스템은 주로 time-driven 센서, event-driven 제어기, event-driven 구동기로 구성되며<sup>[3~4]</sup>, 한 샘플링 주기 내에서 발생하는 데이터의 흐름은 그림 2와 같이 나타낼 수 있다.

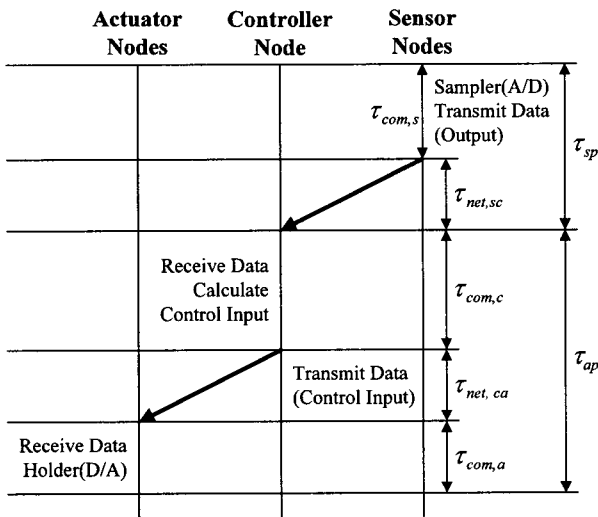


그림 2. 네트워크 제어시스템의 신호흐름선도(단일 패킷)

Fig. 2. Signal flow graph of networked control system (single packet).

그림 2에서 네트워크 제어시스템의 시간 지연은 측정된 출력이 센서, 제어기, 구동기를 거쳐 시스템의 제어 입력으로 인가될 때까지 소요되는 시간이며, 식(3)과 같이 계산된다. 여기서,  $\tau_{sp}$ 는 센서 경로의 시간 지연이고  $\tau_{ap}$ 는 구동기 경로의 시간 지연을 나타낸다<sup>[11]</sup>. 각 경로의 시간 지연은 네트워크 지연(network-induced delay,  $\tau_{net}$ )과 계산 지연(computation delay,  $\tau_{com}$ )의 합이며, 네트워크 지연은 전송 지연(transmission delay,  $\tau_{trans}$ ), 전파 지연(propagation delay,  $\tau_{prop}$ ), 큐 지연(queueing delay,  $\tau_{queue}$ )으로 세분할 수 있다<sup>[7]</sup>.

$$\tau_{sp} = \tau_{com,s} + \tau_{net,sc}$$

$$\tau_{ap} = \tau_{com,c} + \tau_{net,ca} + \tau_{com,a} \quad (3)$$

여기서,  $\tau_{net,sc} = \tau_{queue,s} + \tau_{trans,sc} + \tau_{prop,sc}$

$\tau_{net,ca} = \tau_{queue,c} + \tau_{trans,ca} + \tau_{prop,ca}$

이제,  $m$ 개의 센서에서 측정된 데이터가  $M$ 개의 패킷을 통해 제어기 노드로 전달되고, 제어기 노드에서 계산된 입력 데이터도  $K$ 개의 패킷을 통해  $k$ 개의 구동기 노드로 전달되는 경우를 고려하자. 다중 패킷을 전송하는 두 가지 큰 이유는 첫째, 네트워크 프로토콜의 종류에 따라 패킷 크기가 한정되어 있으므로 해당 크기를 초과하는 데이터는 여러 개의 패킷으로 나누어서 전송해야 하고 둘째, 일반적인 네트워크 제어시스템이 다수의 입출력을 가지므로 각각의 센서와 구동기 노드에서 개별적인 패킷을 전송해야 하기 때문이다.

단일 패킷을 전송할 때 센서 경로와 구동기 경로가 하나씩 존재한다면, 다중 패킷을 전송할 때는 각각  $M$ 개의 센서 경로와  $K$ 개의 구동기 경로가 존재한다. 본 논문에서는 이러한 다중 경로를 통해 데이터가 전송될 수 있도록 아래 (가)~(마)항에서 설명하는 스케줄링 기법을 제안한다.

가. 제어기 노드는 time-driven 방식으로, 매 샘플링 시점을 기준으로 센서 노드와 구동기 노드를 동기화하는 메시지를 방송(broadcasting)한다. 여기서, 동기화를 수행하는데 소요되는 시간 지연은 모든 노드에서  $\tau_b$ 라 가정한다. 이는 센서 노드와 구동기 노드의 시간 왜곡(time-skew)<sup>[7]</sup>이 일정한 상수값을 갖는다는 가정<sup>[1]</sup>과 같은 의미를 지니며, 동일 매체 상에 네트워크 제어시스템을 구축하고 센서 노드와 구동기 노드의 하드웨어 구성이 유사하다는 조건하에 그 의미를 가질 수 있다.

나. 센서 노드는 event-driven 방식으로 동작한다. 즉,  $i$ 번째 센서 노드는( $i=1, 2, \dots, m$ ) 제어기 노드에서 전송하는 동기화 메시지를 수신하는 즉시 해당 출력에 대한 샘플링을 시작하고 샘플링이 종료되면 이 값을 제어기 노드로 전송한다. 여기서,  $i$ 번째 센서 노드의 계산 지연은  $\tau_{com,s}^i$ , 네트워크 지연은  $\tau_{net,sc}^i$ 이다.

다. 제어기 노드는 미리 설정된  $\tau_{sp,max}$  동안 센서 노드에서 전송하는 측정 데이터가 도착할 때까지 대기한다. 만약,  $i$ 번째 센서 노드에서 전송하는 데이터를  $\tau_{sp,max}$  내에 수신하지 못하면 이를  $i$ 번째 센서 노드가

속한 경로의 데이터 손실로 처리하고 가장 최근에 저장된 데이터를 이용하여 제어 입력을 계산한다. TOD (try-once-discard) 스케줄링 방식이 패킷 재전송을 허용하지 않는다[5-6], 이와 같은 방식은 최대 시간 지연  $\tau_{sp, max}$  내에서 패킷 재전송이 가능하다.

라. 제어기 노드는 제어 입력의 계산이 완료되는 즉시 구동기 노드에 각 제어 입력을 전송한다.

마. 구동기 노드는 event-driven 방식으로 동작한다. 즉,  $j$ 번째 구동기 노드는( $j=1, 2, \dots, k$ ) 제어기 노드에서 전송하는 입력 데이터를 수신하는 즉시 플랜트에 인가할 실제 입력을 계산한다. 만약, 미리 설정된  $\tau_{ap, max}$  동안 입력 데이터를 수신하지 못했거나 수신한 입력 데이터로부터 플랜트에 인가할 제어 입력을 계산하지 못했을 경우에는 이를  $j$ 번째 구동기 노드가 속한 경로의 데이터 손실로 처리하고 이전 시점의 제어 입력을 계속해서 플랜트에 인가한다. 즉,  $\tau_{ap, max}$ 는 플랜트에 인가하는 제어 입력의 갱신 시점이 된다. 여기서,  $j$ 번째 구동기 노드의 계산 지연은  $\tau_{com, a}^j$ , 네트워크 지연은  $\tau_{net, ca}^j$  이다.

다중 패킷을 전송하는 네트워크 제어시스템은  $M$ 개의 센서 경로,  $K$ 개의 구동기 경로에 따라 서로 다른 시간 지연을 갖게 되므로 그 해석이 매우 복잡하다. 만약, 센서 경로의 최대 시간 지연  $\tau_{sp, max}$ 와 구동기 경로의 최대 시간 지연  $\tau_{ap, max}$ 를 상수로 정의하고 위의 스케줄링 방식을 적용한다면, 모든  $M$ 개의 경로에서 센서 경로의 시간 지연은  $\tau_{sp, max}$ 이고 모든  $K$ 개의 경로에서 구동기 경로의 시간 지연은  $\tau_{ap, max}$ 가 된다. 즉, 다중 패킷을 전송하는 MIMO 네트워크 제어시스템의 시간 지연을 단일 경로의 시불변 시간 지연처럼 다룰 수 있게 되므로 시간 지연에 따른 네트워크 제어시스템의 안정성 분석을 매우 용이하게 만들 수 있다.

그림 3은 입출력이 각각 2개이고 독립적인 센서와 구동기 노드를 가지는 MIMO 네트워크 제어시스템에 대하여, 각 노드에서 하나의 패킷을 전송할 때 본 논문에서 제시한 스케줄링 방식을 예시한 것이다. 여기서, 센서 경로의 최대 시간 지연  $\tau_{sp, max}$ 는 제어기의 지연 시간, 구동기 경로의 최대 시간 지연  $\tau_{ap, max}$ 는 플랜트의

지연 시간이 된다. 이 때, 각각의 시간 지연은 아래와 같이 계산될 수 있다.

$$\tau_{sp, max} = \tau_b + \max_i \{ \tau_{com, s}^i + \tau_{net, sc}^i \} + \tau_{sp, wait}$$

$$\tau_{ap, max} = \tau_{com, c} + \sum_j \tau_{net, ca}^j + \tau_{com, a}^q + \tau_{ap, wait} \quad (4)$$

식(4)에서  $i, j=1, 2$ 이며,  $q$ 는 구동기 경로의 네트워크 지연( $\tau_{net, ca}$ )이 최대값을 갖는 경로이다(그림 3의 경우,  $q=2$ ). 여기서,  $\tau_{sp, wait}$ 와  $\tau_{ap, wait}$ 는 각 경로에서 메시지 수신이 완료될 때까지 대기하는 여유 시간을 나타낸다. 이 값은 네트워크 제어시스템의 설계자가 결정하는 일종의 설계변수로 생각할 수 있으며, 이를 얼마만큼 주는가에 따라 각 경로의 최대 시간 지연이 증감하게 되어 전체 시스템의 성능에 영향을 미치게 된다.

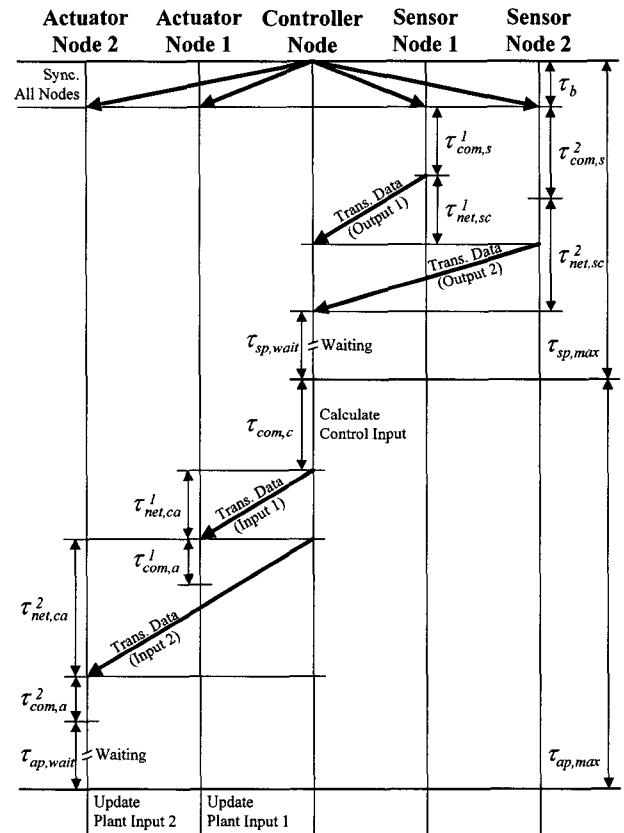


그림 3. 네트워크 제어시스템의 신호흐름선도(다중 패킷)  
Fig. 3. Signal flow graph of networked control system (multiple packet).

2. 이산 스위치 모델

다음과 같은 이산 스위치 모델을 고려하자.

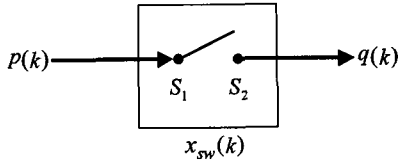


그림 4. 이산 스위치 모델  
Fig. 4. Discrete switch model.

그림 4에서 입력  $p(k)$ 가 출력  $q(k)$ 에 도달할 확률을  $r$ 이라 하면 스위치의 동적 특성은  $r$ 의 확률을 가지는 닫힘 상태와  $(1-r)$ 의 확률을 가지는 열림 상태로 나눌 수 있다. 데이터가 전송될 때는 스위치가 닫혀 있으므로 입력  $p(k)$ 와 출력  $q(k)$ 가 동일한 반면, 데이터가 손실될 때는 스위치가 열려 있으므로 스위치의 상태  $x_{sw}(k)$ 가 출력  $q(k)$ 가 된다. 이와 같은 개념으로부터, 새로운 형태의 이산 스위치 상태방정식을 식(5)와 같이 제안한다.

- 스위치 닫힘 : event  $E$ , rate  $r$

$$x_{sw}(k+1) = p(k), q(k) = p(k) \quad (5.a)$$

- 스위치 열림 : event  $\bar{E}$ , rate  $(1-r)$

$$x_{sw}(k+1) = x_{sw}(k), q(k) = x_{sw}(k) \quad (5.b)$$

여기서,  $x_{sw}(k)$  : 스위치의 상태

예를 들어, 크기 1, 주기 1[sec], 샘플링 주기 0.05[sec]인 정현파가 전송 성공률  $r=0.8$ 을 가지는 스위치를 통해 전달된다고 하자. 이때, 식(5)의 이산 스위치 상태방정식을 이용한 시뮬레이션 결과는 그림 5와 같다.

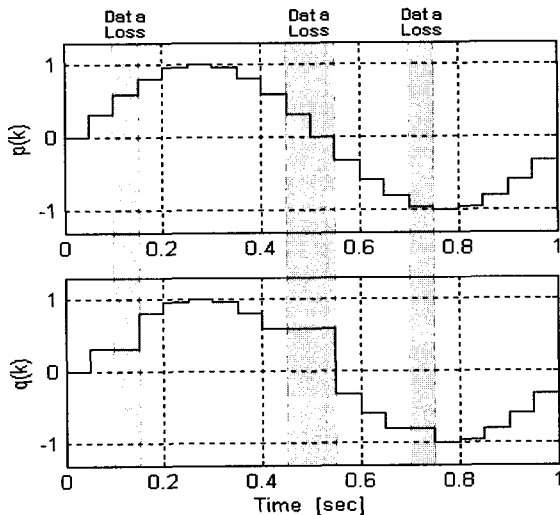


그림 5. 제안한 스위치 방정식의 시뮬레이션 결과  
Fig. 5. Simulation results of proposed switch equation.

위 시뮬레이션 결과는 20개의 샘플링 데이터 중에서 4개의 데이터가 손실되었음을 나타낸다. 데이터 손실이 발생하지 않았을 때는 입력 신호가 그대로 출력단에 전달되는 반면, 데이터 손실이 발생했을 때는 한 시점 이전의 입력 신호가 계속해서 출력단에 유지되고 있음을 알 수 있다(그림 5의 음영 부분). 즉, 본 논문에서 새롭게 제안한 식(5)의 이산 스위치 상태방정식이 데이터의 전송과 손실 사건을 정확하게 표현할 수 있는 모델임을 확인할 수 있다.

### 3. 네트워크 제어시스템의 등가 모델

1절에서 제안한 스케줄링 기법을 이용한다면,  $M$ 개 센서 경로의 최대 시간 지연  $\tau_{sp, max}$ 는 제어기 입력의 지연,  $K$ 개 구동기 경로의 최대 시간 지연  $\tau_{ap, max}$ 는 플랜트 입력의 지연이 된다. 또한,  $\tau_{sp, max}$ 와  $\tau_{ap, max}$ 의 합을 샘플링 주기  $h[sec]$ 보다 작게 설정한다면, 이보다 큰 시간 지연을 갖는 패킷은 모두 데이터 손실로 처리할 수 있다. 각 경로의 시간 지연을 고려한 플랜트와 제어기의 이산화 상태방정식은 다음과 같다<sup>[12]</sup>.

- 플랜트

$$\begin{aligned} \bar{x}_p(k+1) &= \bar{A}_p \bar{x}_p(k) + \bar{B}_p u(k) \\ y(k) &= \bar{C}_p \bar{x}_p(k) \end{aligned} \quad (6)$$

여기서,  $\bar{x}_p^T(k) = [x_p^T(k) \quad u^T(k-1)]$ ,

$$\bar{A}_p = \begin{bmatrix} e^{A_p h} & \int_{h-\tau_{ap, max}}^h e^{A_p s} B_p ds \\ 0 & 0 \end{bmatrix},$$

$$\bar{B}_p = \begin{bmatrix} \int_0^{h-\tau_{ap, max}} e^{A_p s} B_p ds \\ I \end{bmatrix},$$

$$\bar{C}_p = [C_p \quad 0]$$

- 제어기

$$\begin{aligned} \bar{x}_c(k+1) &= \bar{A}_c \bar{x}_c(k) + \bar{B}_c u_c(k) \\ y_c(k) &= \bar{C}_c \bar{x}_c(k) + \bar{D}_c u_c(k) \end{aligned} \quad (7)$$

여기서,  $\bar{x}_c^T(k) = [x_c^T(k) \quad u_c^T(k-1)]$ ,

$$\bar{A}_c = \begin{bmatrix} e^{A_c h} & \int_{h-\tau_{sp, max}}^h e^{A_c s} B_c ds \\ 0 & 0 \end{bmatrix},$$

$$\overline{B}_c = \begin{bmatrix} \int_0^{h^{-1}t_{sp, max}} e^{A_c s} B_c ds \\ I \end{bmatrix},$$

$$\overline{C}_c = [ C_c \quad 0 ], \quad \overline{D}_c = D_c$$

이제, 식(6), (7)의 변형된 플랜트와 제어가 이산 스위치로 연결된 그림 6을 살펴보자. 여기서, 각 경로별 이산 스위치의 입출력 신호와 상태는 다음과 같이 정의 하였다.

- 센서 경로

$$u_s(k) = \{ u_{s1}^T(k), u_{s2}^T(k), \dots, u_{sM}^T(k) \}^T,$$

$$x_s(k) = \{ x_{s1}^T(k), x_{s2}^T(k), \dots, x_{sM}^T(k) \}^T,$$

$$y_s(k) = \{ y_{s1}^T(k), y_{s2}^T(k), \dots, y_{sM}^T(k) \}^T \quad (8)$$

- 구동기 경로

$$u_a(k) = \{ u_{a1}^T(k), u_{a2}^T(k), \dots, u_{aK}^T(k) \}^T,$$

$$x_a(k) = \{ x_{a1}^T(k), x_{a2}^T(k), \dots, x_{aK}^T(k) \}^T,$$

$$y_a(k) = \{ y_{a1}^T(k), y_{a2}^T(k), \dots, y_{aK}^T(k) \}^T \quad (9)$$

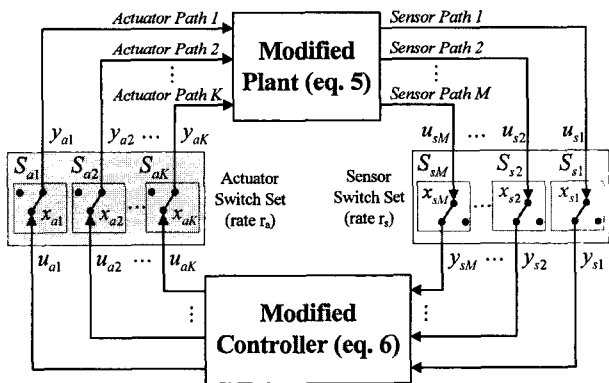


그림 6. 이산 스위치 모델로 나타낸 다중 경로  
Fig. 6. Multiple paths described by discrete switch models.

먼저, 센서 및 구동기 경로에서 단일 패킷이 전송될 때의 네트워크 제어시스템을 고려하자. 이는 그림 6의 센서 경로와 구동기 경로가 하나씩 존재하는 것이므로 스위치  $S_{si} (i=1, 2, \dots, M)$ 와  $S_{aj} (j=1, 2, \dots, K)$ 도 단일 스위치 집합으로 통합할 수 있다.

$$S_s = \{ S_{s1}, S_{s2}, \dots, S_{sM} \},$$

$$S_a = \{ S_{a1}, S_{a2}, \dots, S_{aK} \} \quad (10)$$

다음으로, 센서 경로에서 데이터가 전송되는 사건  $E_s$  는 스위치 집합  $S_s$ 가 닫힌 상태, 데이터가 손실되는 사건  $\overline{E}_s$ 는 스위치 집합  $S_s$ 가 열린 상태라 하자. 스위치  $S_s$ 가 닫혔을 경우에는 현재의 출력  $y_s(k)$ 는 현재의 입력  $u_s(k)$ 와 같고 스위치  $S_s$ 가 열렸을 경우에는 현재의 출력  $y_s(k)$ 는 이전의 입력  $u_s(k-1)$ 과 같으므로 센서 경로의 동적 방정식은 식(5)로부터 아래와 같이 결정된다.

- 센서 경로 데이터 전송 ( $E_s$  : rate  $r_s$ )

$$x_s(k+1) = u_s(k), y_s(k) = u_s(k) \quad (11.a)$$

- 센서 경로 데이터 손실 ( $\overline{E}_s$  : rate  $1-r_s$ )

$$x_s(k+1) = x_s(k), y_s(k) = x_s(k) \quad (11.b)$$

여기서,  $x_s(t) \in R^m$  : 센서 경로의 상태,

$r_s$  : 센서 경로의 데이터 전송 성공률 ( $0 < r_s \leq 1$ )

동일한 방법으로 구동기 경로의 데이터 전송 ( $E_a$ )과 손실 ( $\overline{E}_a$ ) 사건을 표현하면 식(12)와 같은 동적 방정식을 유도할 수 있다.

- 구동기 경로 데이터 전송 ( $E_a$  : rate  $r_a$ )

$$x_a(k+1) = u_a(k), y_a(k) = u_a(k) \quad (12.a)$$

- 구동기 경로 데이터 손실 ( $\overline{E}_a$  : rate  $1-r_a$ )

$$x_a(k+1) = x_a(k), y_a(k) = x_a(k) \quad (12.b)$$

여기서,  $x_a(t) \in R^k$  : 구동기 경로의 상태,

$r_a$  : 구동기 경로의 데이터 전송 성공률 ( $0 < r_a \leq 1$ )

데이터 손실이 발생하지 않을 경우 네트워크 제어 시스템은 식(6)→식(11.a)→식(7)→식(12.a)의 시스템으로 표현할 수 있다. 동일한 방법으로 센서 경로에서 데이터 손실이 발생할 경우에는 식(6)→식(11.b)→식(7)→식(12.a), 구동기 경로에서 데이터 손실이 발생할 경우

에는 식(6)→식(11.a)→식(7)→식(12.b) 그리고 모든 경로에서 데이터 손실이 발생할 경우에는 식(6)→식(11.b)→식(7)→식(12.b)로 네트워크 제어시스템을 표현할 수 있다. 즉, 단일 패킷을 전송하는 네트워크 제어시스템에서 시간 지연과 데이터 손실을 고려했을 때 전체 시스템의 등가 모델은 식(6), (7), (11), (12)의 시스템이 모두 선형적으로 연결된 시스템(linear interconnected system)이라 생각할 수 있다. 이러한 시스템은 참고문헌 [13]의 Lemma 1, 참고문헌 [11]의 정리 1을 적용할 때, 식(13)과 같은 등가 모델로 나타낼 수 있다.

정리 1 센서 경로와 구동기 경로의 최대 시간 지연이 각각  $\tau_{sp, max}$ ,  $\tau_{ap, max}$  이고 각 경로에서 단일 패킷이 전송될 때, 그림 6의 네트워크 제어시스템과 등가인 비동기 동적 시스템은 식(13)과 같다.

$$\begin{aligned} x_t(k+1) &= \Phi_s x_t(k) + \Gamma_s w(k) \\ y(k) &= \begin{bmatrix} 0 & 0 & \overline{C}_b & 0 \end{bmatrix} x_t(k) \end{aligned} \quad (13)$$

여기서,  $s = 1, 2, 3, 4$

$$\begin{aligned} x_t^T(k) &= \begin{bmatrix} \overline{x}_c^T(k) & x_a^T(k) & \overline{x}_b^T(k) & x_s^T(k) \end{bmatrix} \\ \Phi_1 &= \begin{bmatrix} \overline{A}_c & 0 & -\overline{B}_c \overline{C}_b & 0 \\ \overline{C}_c & 0 & -\overline{D}_c \overline{C}_b & 0 \\ \overline{B}_b \overline{C}_c & 0 & \overline{A}_b - \overline{B}_b \overline{D}_c \overline{C}_b & 0 \\ 0 & 0 & \overline{C}_b & 0_m \end{bmatrix}, \Gamma_1 = \begin{bmatrix} \overline{B}_c \\ \overline{D}_c \\ \overline{B}_b \overline{D}_c \\ 0 \end{bmatrix} \\ \Phi_2 &= \begin{bmatrix} \overline{A}_c & 0 & 0 & -\overline{B}_c \\ \overline{C}_c & 0 & 0 & -\overline{D}_c \\ \overline{B}_b \overline{C}_c & 0 & \overline{A}_b & -\overline{B}_b \overline{D}_c \\ 0 & 0 & \overline{C}_b & I_m \end{bmatrix}, \Gamma_2 = \begin{bmatrix} \overline{B}_c \\ \overline{D}_c \\ \overline{B}_b \overline{D}_c \\ 0 \end{bmatrix} \\ \Phi_3 &= \begin{bmatrix} \overline{A}_c & 0 & -\overline{B}_c \overline{C}_b & 0 \\ 0 & I_k & 0 & 0 \\ 0 & \overline{B}_b & \overline{A}_b & 0 \\ 0 & 0 & \overline{C}_b & 0_m \end{bmatrix}, \Gamma_3 = \begin{bmatrix} \overline{B}_c \\ 0 \\ 0 \\ 0 \end{bmatrix} \\ \Phi_4 &= \begin{bmatrix} \overline{A}_c & 0 & 0 & -\overline{B}_c \\ 0 & I_k & 0 & 0 \\ 0 & 0 & \overline{A}_b & 0 \\ 0 & 0 & 0 & I_m \end{bmatrix}, \Gamma_4 = \begin{bmatrix} \overline{B}_c \\ 0 \\ 0 \\ 0 \end{bmatrix} \end{aligned}$$

증명 : 참고문헌 [11],[13]의 선형 결합시스템 등가 모델 유도과정 참조. ■

참고로, 본 논문에서 유도한 식(13)의 비동기 시스템은  $\Pi$ 2장의 시뮬레이션을 통해 그 적합성을 입증한 식(5)의 새로운 스위치 모델을 적용한 것이다. 따라서 참고문헌 [11]의 등가 모델과는 다른 형태를 가진다.

정리 1의 네트워크 제어시스템에서 데이터 손실이 발생하지 않는다면 네트워크 제어시스템은 이산 상태  $s=1$  일 때의 시스템과 동일하다. 이때, 시스템의 전달함수 행렬  $T_{\Phi_1}(z)$ 는 전달함수 행렬의 등가 소거법칙을 통해 식(14)와 같이 나타낼 수 있다.

$$T_{\Phi_1}(z) = \left[ \begin{array}{cc|c} \overline{A}_c & -\overline{B}_c \overline{C}_b & \overline{B}_c \\ \overline{B}_b \overline{C}_c & \overline{A}_b - \overline{B}_b \overline{D}_c \overline{C}_b & \overline{B}_b \overline{D}_c \\ \hline 0 & \overline{C}_b & 0 \end{array} \right] \quad (14)$$

반대로, 네트워크 제어시스템의 모든 경로에서 데이터 손실이 발생했을 때를 고려하자. 이 경우, 네트워크 제어시스템은 이산 상태  $s=4$  일 때의 시스템과 같고 시스템의 전달함수 행렬  $T_{\Phi_4}(z)$ 는 식(15)와 같이 나타낼 수 있다.

$$T_{\Phi_4}(z) = \left[ \begin{array}{cc|c} \overline{A}_c & 0 & \overline{B}_c \\ 0 & \overline{A}_b & 0 \\ \hline 0 & \overline{C}_b & 0 \end{array} \right] \quad (15)$$

식(14)로부터, 네트워크 제어시스템에서 데이터 손실이 발생하지 않을 경우에는 스위치 모델의 동적 특성이 전체 시스템에 영향을 미치지 않음을 알 수 있다. 또한, 식(15)는 모든 네트워크 경로에서 데이터 손실이 발생, 궤환 경로가 단절됨으로써 제어기와 플랜트가 완전히 분리되었음을 나타내고 있다. 이상의 고찰로부터, 식(14), (15)는 본 논문에서 유도한 네트워크 제어시스템의 등가모델이 논리적으로 타당함을 입증하는 결과라 할 수 있다.

정리 2 정리 1의 네트워크 제어시스템에서 데이터 손실이 발생하지 않는다면( $r_s = r_a = 1$ ), 전체 시스템이

안정하기 위한 필요충분조건은 식(16)과 같다.

$$\left| \lambda_i \left( \begin{array}{cc} \overline{A}_c & -\overline{B}_c \overline{C}_p \\ \overline{B}_p \overline{C}_c & \overline{A}_p - \overline{B}_p \overline{D}_c \overline{C}_p \end{array} \right) \right| < 1 \quad (16)$$

여기서,  $i = 1, 2, \dots, n + p + m + k$

증명 : 식(14)로부터 자명하다. ■

### III. 다중 패킷을 전송하는 네트워크 제어시스템의 안정성 분석

외부 사건  $E_i (i = 1, 2, \dots, M)$ 에 의해 달라지는  $N$ 개의 동적 특성을 갖는 비동기 시스템은 식(17)과 같이 나타낼 수 있다<sup>[9]</sup>.

$$x(k+1) = f_j \{ x(k) \}, \quad j = 1, 2, \dots, N \quad (17)$$

여기서,  $j$  : 서로 다른 동적 특성을 가지는 이산 상태(discrete state)

$f_j \{ \cdot \}$  : 이산 상태  $j$ 의 동적 특성

본 논문에서 유도한 네트워크 제어시스템은 각 경로의 데이터 전송과 손실을 외부 사건으로 갖는 비동기 시스템이라 생각할 수 있다. 먼저, 식(13)으로 나타낸 네트워크 제어시스템에서 안정성에 영향을 미치는 시스템 행렬을 식(18)과 같이 표현하고

$$x_i(k+1) = f_j \{ x_i(k) \} = \Phi_j x_i(k) \quad (18)$$

여기서,  $j = 1, 2, 3, 4$

센서 경로와 구동기 경로의 데이터 전송과 손실을 집합  $E$ 에 속하는 사건이라 하자.

$$E \in \{ E_s, \overline{E}_s, E_a, \overline{E}_a \} \quad (19)$$

여기서,  $E_s / \overline{E}_s$  : 센서 경로 데이터 전송 / 손실

$E_a / \overline{E}_a$  : 구동기 경로 데이터 전송 / 손실

이때, 각 발생 사건에 따라 달라지는 네트워크 제어시스템의 이산 상태는 식(20)과 같은 네 가지 경우로 나눌 수 있다.

$$j \in \{ E_s \times E_a, \overline{E}_s \times E_a, E_s \times \overline{E}_a, \overline{E}_s \times \overline{E}_a \} \quad (20)$$

이와 같은 네트워크 제어시스템의 이산 상태 변화는 그림 7과 같은 상태 천이도로 나타낼 수 있다.

다음으로  $j$ 라는 이산 상태가 되기 위한 사건 집합을  $I(j) = \{ E_{j1}, E_{j2}, \dots, E_{jM_j} \}$ 라 하자. 식(20)과 그림 7로부터  $M_j = 2$ 임을 알 수 있으며, 사건 집합  $I(j)$ 는 식(21)과 같이 나타낼 수 있다.

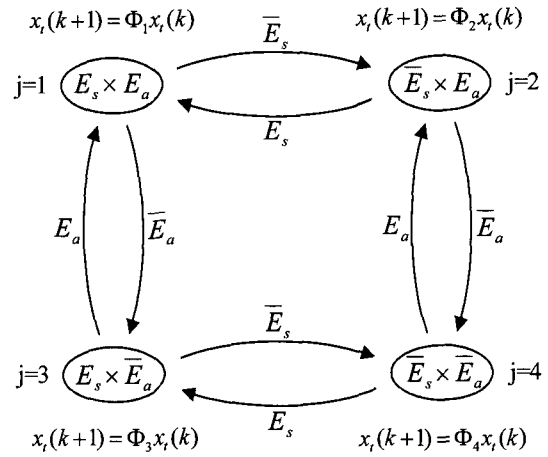


그림 7. 네트워크 제어시스템의 상태 천이도  
Fig. 7. State transition diagram of networked control system.

$$\begin{aligned} I(1) &= I(E_s \times E_a) = \{ E_{11}, E_{12} \} = \{ E_s, E_a \}, \\ I(2) &= I(\overline{E}_s \times E_a) = \{ E_{21}, E_{22} \} = \{ \overline{E}_s, E_a \}, \\ I(3) &= I(E_s \times \overline{E}_a) = \{ E_{31}, E_{32} \} = \{ E_s, \overline{E}_a \}, \\ I(4) &= I(\overline{E}_s \times \overline{E}_a) = \{ E_{41}, E_{42} \} = \{ \overline{E}_s, \overline{E}_a \} \quad (21) \end{aligned}$$

센서 경로의 전송 성공률을  $r_s$ , 구동기 경로의 전송 성공률을  $r_a$ 라 하면 각 사건의 발생 확률은 식(22)와 같다.

$$E_s: r_s, \overline{E}_s: (1 - r_s), E_a: r_a, \overline{E}_a: (1 - r_a) \quad (22)$$

따라서 참고문헌 [9]의 비동기 시스템 안정 조건은 식(19)~(22)로부터 아래와 같이 변환할 수 있다.

$$r_s \log a_1 + r_a \log a_2 + (1 - r_s) \log a_3 + (1 - r_a) \log a_4 > \log a > 0 \quad (23)$$

$$\begin{aligned} V\{x_i(k+1)\} - V\{x_i(k)\} \\ \leq (a_{j1}^{-2} a_{j2}^{-2} - 1) V\{x_i(k)\} \quad (24) \end{aligned}$$

여기서,  $j = 1, 2, 3, 4$ .

$a_1, a_2, a_3, a_4$  : 사건  $E_s, E_a, \overline{E}_s, \overline{E}_a$ 에 대응되는 양수



이상으로부터, 식(13)의 네트워크 제어시스템이 안정하기 위한 충분조건은 식(24)를 만족하는 Lyapunov 함수  $V\{x_i(k)\}$ 와 식(23)을 만족하는 양수  $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ 가 존재하는 것임을 알 수 있으며, 이는 쌍일차 행렬 부등식(bilinear matrix inequality: BMI)의 해를 찾는 문제로 귀결된다<sup>[9, 11]</sup>. 이 문제를 직접 풀기 위해서는 상당한 계산 시간이 소요되며 플랜트나 제어기의 차수가 높아질수록 이러한 계산 시간은 더욱 증가한다. 또한, 센서와 구동기 경로의 전송 성공률을 임의로 결정한 후 BMI 해의 존재 여부를 계산하기 때문에, 네트워크 제어시스템의 안정성을 보장하는 전송 성공률 범위를 한번에 파악할 수 없다는 단점이 있다. 만약, 센서 및 구동기 경로의 전송 성공률이 동일하다고 가정하면( $r_s = r_a = r$ )<sup>[11]</sup>, 정리 3과 같은 간단한 대수 부등식 형태의 안정 조건을 유도할 수 있다.

정리 3 정리 2를 만족하는 네트워크 제어시스템에서 센서 경로와 구동기 경로의 데이터 전송 성공률이 같다면, 다음과 같은 범위에 속한 모든 전송 성공률  $r$ 에서 전체 시스템은 안정하다.

$$\frac{\log \lambda_{\max}^2(\bar{\Phi}_4)}{\log \lambda_{\max}^2(\bar{\Phi}_4) - \log \lambda_{\max}^2(\bar{\Phi}_1)} < r \leq 1 \quad (25)$$

$$\text{여기서, } \bar{\Phi}_1 = \begin{bmatrix} \bar{A}_c & -\bar{B}_c \bar{C}_b \\ \bar{B}_b \bar{C}_c & \bar{A}_b - \bar{B}_b \bar{D}_c \bar{C}_b \end{bmatrix}$$

$$\bar{\Phi}_4 = \begin{bmatrix} \bar{A}_c & 0 \\ 0 & \bar{A}_b \end{bmatrix}$$

증명 : 참고문헌 [11]의 정리 4 증명과정 참조. 단, 증명과정 내에서  $\beta_4$ 의 범위는 식(16)을 이용하여 다음과 같이 나타낼 수 있고

$$\begin{aligned} \beta_4 &\geq \lambda_{\max}^2(\bar{\Phi}_4) \\ &= \max[\lambda_{\max}^2(\bar{A}_b), \lambda_{\max}^2(\bar{A}_c), 1] \\ &= \max[\lambda_{\max}^2(\bar{A}_b), \lambda_{\max}^2(\bar{A}_c)] \\ &= \lambda_{\max}^2(\bar{\Phi}_4) \end{aligned} \quad (26)$$

식(26)에서 결정한 새로운  $\beta_4$ 의 하한값을 통해 정리 3의 전송 성공률 범위를 유도할 수 있다. ■

정리 3의 결과로부터, 네트워크 제어시스템의 안정성을 보장하는 전송 성공률의 범위는 모든 경로에서 데이터가 손실된 경우를 나타내는 이산 상태  $\bar{\Phi}_4$ 와 데이터 손실이 발생하지 않은 경우의 이산 상태  $\bar{\Phi}_1$ 에 의해서 결정됨을 알 수 있다.

이제, 다중 패킷이 전송될 때 네트워크 제어시스템의 안정 조건을 살펴보자. 이는 그림 6의  $M$ 개의 센서 경로 스위치,  $K$ 개의 구동기 경로 스위치가 모두 독립적으로 동작하는 경우이다. 따라서, 네트워크 제어시스템의 이산 상태는 단일 패킷을 전송했을 때의 4가지에서 다중 패킷을 전송할 때는  $2^{M+K}$ 가지로 증가하게 된다. 이 경우, 정리 1과 유사한 방법을 통해  $2^{M+K}$ 개에 달하는 네트워크 제어시스템의 등가 모델을 유도하고 안정성을 분석하는 것은 매우 어려운 문제가 된다. 만약, 단일 패킷을 전송할 때 센서와 구동기 경로의 전송 성공률을  $r$ 이라 한다면,  $M$ 개의 출력 데이터가 제어기에 성공적으로 전송될 확률은  $r^M$ ,  $K$ 개의 입력 데이터가 구동기에 성공적으로 전송될 확률은  $r^K$ 와 같다. 이로부터, 다중 패킷을 전송할 때 네트워크 제어시스템의 안정성을 보장하는 전송 성공률의 범위를 정리 4와 같이 제안한다.

정리 4 센서 경로와 구동기 경로의 전송 성공률이  $r$ 이고 센서 경로에서는  $M$ 개의 패킷이, 구동기 경로에서는  $K$ 개의 패킷이 전송된다고 하자. 이때, 정리 2를 만족하는 네트워크 제어시스템은 식(27)의 전송 성공률 범위 내에서 안정하다.

$$\max[r^{\frac{1}{M}_{\min}}, r^{\frac{1}{K}_{\min}}] < r \leq 1 \quad (27)$$

$$\text{여기서, } r_{\min} = \frac{\log \lambda_{\max}^2(\bar{\Phi}_4)}{\log \lambda_{\max}^2(\bar{\Phi}_4) - \log \lambda_{\max}^2(\bar{\Phi}_1)}$$

증명 : 정리 3을 통해 구해진 네트워크 제어시스템의 안정성을 보장하는 전송 성공률 하한값을  $r_{\min}$ 이라 한다면, 단일 패킷을 전송하는 네트워크 제어시스템은 각 경로에서  $r_{\min}$ 보다 큰 전송 성공률을 가질 경우 전체 시스템의 안정성을 보장할 수 있다.  $M$ 개의 패킷을 전송하는 센서 경로에서 전체 데이터 패킷이 모두 전송될 확률은  $r^M$ 이고  $K$ 개의 패킷을 전송하는 구동기 경로에서 전체 데이터 패킷이 모두 전송될 확률은  $r^K$ 이

므로  $r^M > r_{\min}$  이고  $r^K > r_{\min}$  인 전송 성공률  $r$  에 대하여 전체 시스템은 안정하다. 따라서

$$\max [r_{\min}^{\frac{1}{M}}, r_{\min}^{\frac{1}{K}}] < r \quad (28)$$

인 모든 전송 성공률  $r$ 에 대하여 전체 네트워크 제어 시스템은 안정하다. ■

#### IV. 시뮬레이션

다음과 같은 Batch Reactor 모델을 고려하자<sup>[5~6]</sup>.

$$\begin{aligned} \dot{x}_p(t) &= A_p x_p(t) + B_p u(t) \\ y(t) &= C_p x_p(t) \end{aligned} \quad (29)$$

여기서,

$$A_p = \begin{bmatrix} 1.38 & -0.2077 & 6.715 & -5.676 \\ -0.5814 & -4.29 & 0 & 0.675 \\ 1.067 & 4.273 & -6.654 & 5.893 \\ 0.048 & 4.273 & 1.343 & -2.104 \end{bmatrix},$$

$$B_p = \begin{bmatrix} 0 & 0 \\ 5.679 & 0 \\ 1.136 & -3.146 \\ 1.136 & 0 \end{bmatrix}, \quad C_p = \begin{bmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

이를 안정화시키는 제어기는 식(30)과 같이 설계하였다.

$$\begin{aligned} \dot{x}_c(t) &= A_c x_c(t) + B_c u_c(t) \\ y_c(t) &= C_c x_c(t) \end{aligned} \quad (30)$$

여기서,

$$A_c = \begin{bmatrix} -5.6810 & -2.0343 & -0.3460 & 1.3850 \\ -0.6960 & -5.0138 & -1.0545 & 0.8919 \\ -4.8464 & -6.9103 & -8.1633 & 6.3130 \\ -3.0617 & -4.4990 & -1.9547 & 1.0261 \end{bmatrix},$$

$$B_c = \begin{bmatrix} 7.0610 & 1.8266 \\ 0.6063 & 1.3652 \\ 3.1138 & 10.1069 \\ 3.2080 & 8.9004 \end{bmatrix},$$

$$C_c = \begin{bmatrix} -0.0866 & -0.1129 & 0.0789 & 0.0686 \\ -0.9211 & -0.3830 & 0.5385 & -0.8315 \end{bmatrix}$$

식(29)의 Batch Reactor 모델과 식(30)의 제어기에 대하여 네트워크 제어시스템을 구축하고 그림 3의 스케줄링 기법을 적용했을 때, 안정성을 보장하는 시간 지연의 범위는 정리 2를 적용하여 구하였다(전송 성공률  $r = 1$ ). 그림 8은 이때의 시뮬레이션 결과를 보여주고 있으며, 샘플링 주기별로 안정성을 보장하는 최대 시간 지연의 값이 달라지고 있음을 확인할 수 있다.

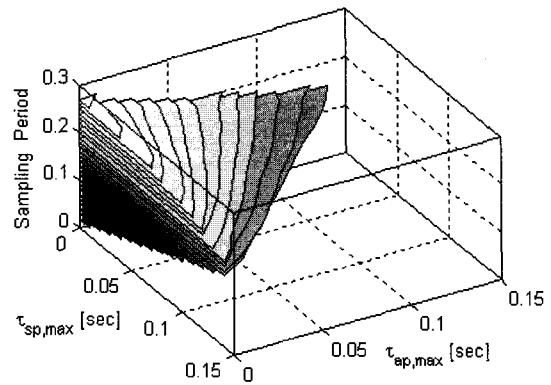


그림 8. 네트워크 제어시스템의 안정 영역(시간 지연과 샘플링 주기)

Fig. 8. Stability regions of networked control system(time delay vs. sampling period).

동일 조건 하에서 샘플링 주기  $h = 0.2[sec]$ , 모든 센서와 구동기 노드가 패킷을 전송할 때(각 경로당 2개의 패킷), 안정성을 보장하는 시간 지연과 데이터 전송 성공률의 범위는 정리 4를 적용하여 구하였으며, 그 결과는 그림 9에 표시된 영역의 내부에 해당한다.

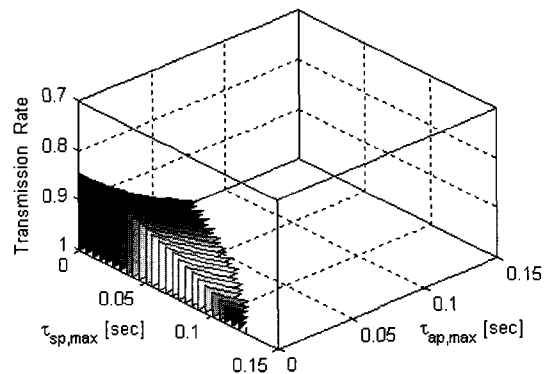


그림 9. 네트워크 제어시스템의 안정 영역(다중 패킷 전송, 시간 지연과 전송 성공률)

Fig. 9. Stability regions of networked control system(time delay vs. data loss with multiple packet transmission).

만약, 샘플링 주기  $h=0.2[sec]$ , 센서 경로의 최대 시간 지연  $\tau_{sp,max}=0.01[sec]$ , 구동기 경로의 최대 시간 지연  $\tau_{ap,max}=0.005[sec]$ 라면, 전송 패킷의 개수에 따라 안정성을 보장하는 전송 성공률의 범위는 표 1과 같다.

표 1. 데이터 패킷 수에 따른 전송 성공률  
Table 1. Transmission rates with respect to the number of data packets.

경로별 최대 패킷 수 \ 전송 성공률	1	2	3	4	5
안정성을 보장하는 전송 성공률의 하한값 (정리 4)	0.7528	0.8677	0.9097	0.9315	0.9448

표 1로부터 센서 또는 구동기 경로에서 최대 2개의 데이터 패킷이 전송될 때 안정성을 보장하는 전송 성공률의 하한값은  $r=0.8677$ 임을 알 수 있다. 그림 10은 이때의 시스템 출력과 전송 성공률  $r=1$ 일 때의 시스템 출력을 비교한 것이며, Batch Reactor 모델의 상태 초기값은  $x_p(t)=[1 \ -1 \ 1 \ -1]^T$ 라 가정하였다.

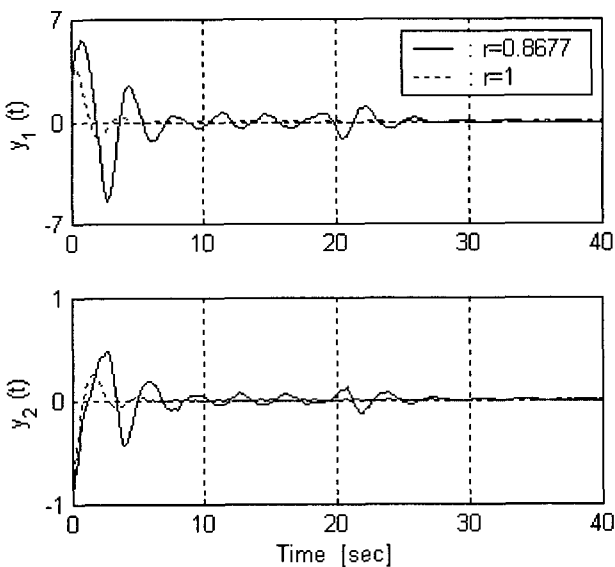


그림 10. Batch Reactor 시스템의 출력 비교  
Fig. 10. Output Comparison of batch reactor system.

이상의 시뮬레이션 결과로부터, 본 논문에서 제안한 정리들이 다중 패킷을 전송하는 네트워크 제어시스템의 안정성을 효과적으로 분석할 수 있음을 입증할 수 있다.

### V. 결 론

본 논문에서는 네트워크의 데이터 손실을 반영할 수 있는 새로운 이산 스위치 방정식을 제안하고 이를 이용하여 MIMO 네트워크 제어시스템의 등가 모델을 유도하였다. 또한, 다중 데이터 패킷을 전송할 때 사용할 수 있는 새로운 메시지 스케줄링 방식을 제안하였다. 이후, 제안된 스케줄링 방식을 적용하였을 때 네트워크 제어시스템의 안정성 변화를 분석할 수 있는 정리들을 제시, 증명하였다. 4장에서는 Batch Reactor 시스템을 대상으로 시간 지연, 데이터 손실, 샘플링 주기, 전송 데이터 패킷의 개수에 따른 안정성 분석을 시뮬레이션하고 그 결과를 분석함으로써 제안한 정리들의 타당성을 입증하였다. 본 논문의 연구 결과를 실제 시스템 구축에 적용한다면 네트워크 제어시스템 초기 설계과정에서 시간 지연, 데이터 손실, 전송 데이터 패킷의 개수에 따른 안정성 변화를 정확히 분석할 수 있으므로 전체 시스템의 신뢰성을 높이는데 일조할 것으로 기대된다. 또한, 제안된 네트워크 제어시스템 분석 기법은 대규모 시스템, 분산형 시스템, 공장자동화, 로봇을 포함한 메카트로닉스, 전력설비 감시진단 등 다양한 분야에 응용할 수 있을 것으로 판단된다. 향후, 본 논문에서 제안한 안정성 분석 정리를 적용하여 네트워크 제어시스템의 안정화 제어를 설계하는 연구가 이루어져야 할 것이다.

### 참 고 문 헌

- [1] Yodyium Tipsuwan and Mo-Yuen Chow, "Control Methodologies in Networked Control Systems," *Control Engineering Practice*, vol. 11, issue 5, pp. 483-492, May 2003.
- [2] T.C. Yang, "Networked Control System:A Brief Survey," *IEE Proceedings-Control Theory and Applications*, vol. 153, issue 4, pp. 403-412, July 2006.
- [3] Michael S. Branicky, Stephen M. Phillips, and Wei Zhang, "Stability of Networked Control Systems:Explicit Analysis of Delay," *Proceedings of America Control Conference*, Chicago, IL, pp. 2352-2357, June 2000.
- [4] Wei Zhang, Michael S. Branicky, and Stephen M. Phillips, "Stability of Networked Control Systems," *IEEE Control Systems Magazine*, vol. 21, no. 1, pp. 84-99, Feb. 2001.
- [5] Gregory C. Walsh, Hong Ye, and Linda G. Bushnell, "Stability Analysis of Networked

Control Systems," *IEEE Transactions on Control Systems Technology*, vol. 10, no. 3, pp. 438-446, May 2002.

[6] D. Nešić and A.R. Teel, "Input-Output Stability Properties of Networked Control Systems," *IEEE Transactions on Automatic Control*, vol. 49, no. 10, pp. 1650-1667, Oct. 2004.

[7] Feng-Li Lian, James Moyne, and Dawn Tilbury, "Modelling and Optimal Controller Design of Networked Control Systems with Multiple Delays," *International Journal of Control*, vol. 79, no. 6, pp. 591-606, June 2003.

[8] Luis A. Montestruque and Panos Antsaklis, "Stability of Model-Based Networked Control Systems with Time-Varying Transmission Times," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1562-1572, Sep. 2004.

[9] Arash Hassibi, Stephen P. Boyd, and Jonathan P. How, "Control of Asynchronous Dynamical Systems with Rate Constraints on Events," *Proceedings of 38th Conference on Decision & Control*, Phoenix, AZ, pp. 1345-1351, Dec. 1999.

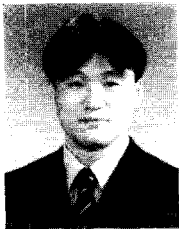
[10] Dong Yue, Qing-Long Han, and Chen Peng, "State Feedback Controller Design of Networked Control Systems," *IEEE Transactions on Circuits and Systems*, vol. 51, no. 11, pp. 640-644, Nov. 2004.

[11] 정준홍, 최수영, 박기현, "시간 지연과 데이터 손실을 고려한 네트워크 제어시스템의 안정도 분석," *대한전기학회논문지*, 제53D권, 제10호, 678- 689쪽, 2004년 10월

[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.

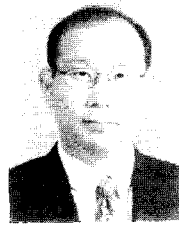
저 자 소 개



정 준 홍(정회원)  
1996년 성균관대학교 전기공학과  
학사 졸업.  
1998년 성균관대학교 전기공학과  
석사 졸업.  
2005년 성균관대학교 전기전자 및  
컴퓨터공학과 박사 졸업.

2005년~현재 한전KDN(주) 전력IT연구원 주임  
연구원.

<주관심분야 : 네트워크 제어시스템, 임베디드 시  
스템, 전력IT 응용>



박 기 현(정회원)  
1978년 서울대학교 전기공학과  
학사 졸업.  
1980년 서울대학교 전기공학과  
석사 졸업.  
1987년 Polytechnic University  
박사 졸업.

1988년~1990년 한국전자통신연구원 선임연구원

1990년~현재 성균관대학교 정보통신공학부

정교수.

<주관심분야 : 다변수제어, 강인제어시스템>



이 재 호(정회원)  
2003년 건양대학교 제어계측  
공학과 학사 졸업.  
2005년 성균관대학교 전자전기  
공학과 석사 졸업.  
2005년~현재 성균관대학교 전자  
전기공학과 박사 과정.

<주관심분야 : 진동제어, 최적제어, 강인제어, 다  
변수제어, 디지털 신호처리>