

## CAN(Controller Area Network)을 이용한 역진자 시스템 제어

최성섭, 유래성, 홍석교  
아주대학교 전자공학부

### Inverted Cart Pendulum Control Using CAN(Controller Area Network)

Seong-Seop Choi, Lae-Sung Yu, Suk-Kyo Hong  
School of Electronics Engineering, Ajou University

**Abstract** - This paper considers a networked control system (NCS) that consists of an inverted cart pendulum, a digital controller, and a controller area network (CAN) in which the actuator and sensors of the pendulum are connected to form a closed-loop system. The worst-case message response time (WCMRT) in the CAN is analyzed and the analysis results are applied to the target control system. For the case where the control system cannot satisfy the WCMRT condition and therefore time delays are inevitable, the Luck and Ray method is used to compensate the network-induced time delays. Simulations are carried out to show the feasibility of the proposed scheme.

#### 1. 서 론

제어시스템이 점차로 복잡해짐에 따라, 배선에 드는 비용이 증가하거나 시스템을 설치하고 유지, 보수, 진단하기가 더욱 어렵게 되었다. 특히, 복잡한 시스템에서는 센서 및 구동기 등이 주제어기와 공간적으로 많이 떨어져 있는 경우가 많아 이들을 배선할 때 여러 가지 어려움이 따른다.

네트워크 기반 제어시스템 (NCS : Networked Control System)은 제어기 신호 및 센서 신호 등의 전달경로를 네트워크로 구성한 시스템을 말한다. NCS는 특히 이렇게 센서 및 구동기 등이 주제어기와 공간적으로 많이 떨어져 있는 시스템에 유용한데, 이는 각 신호들의 전달경로를 네트워크로 구성함으로써 배선에 드는 비용을 절감하고 시스템의 설치, 유지, 보수, 진단을 보다 쉽게 할 수 있기 때문이다. 이러한 시스템의 예로는 대표적으로 항공기나 자동차를 들 수 있다. 특히, 자동차의 경우, 요즘 CAN (Control Area Network)이 제어시스템 신호전달의 새로운 매개체로 부각되고 있어 앞으로 자동차 제어시스템의 네트워크화가 가속될 전망이다.

NCS는 배선에 드는 비용을 절감하고 시스템 유지 및 보수를 용이하게 하며 시스템의 모듈화를 가능하게 하는 이점이 있으나, 네트워크에 기인하는 시간지연이라는 큰 문제점을 안고 있다. 특히, 대상 제어시스템이 빠른 동적 특성을 갖게 되면 이 문제는 더 심각해진다. 이 시간지연은 시간에 따라 변하기 때문에, 일정한 시간지연을 갖는 시스템을 위하여 고안된 기존의 방법으로는 이를 다루기가 쉽지 않다. 따라서 네트워크에 기인하는 시간지연에 대한 새로운 접근법이 필요하다.

이 논문에서는 최소 3개의 노드로 구성되어 있는 기존의 역진자 시스템을 네트워크 기반 제어시스템으로 환경을 바꾸어 그에 따라 발생하는 시간 지연문제를 해결해 봄으로써 앞으로 산업설비 전반에 걸친 네트워크화를 기대해 볼 수 있다. 본론에서는 크게 4가지 부분으로 나누어 CAN에 대한 소개, 메시지 최대 응답 시간 계산, 실현환경에 대한 소개, 모의실험 결과를 다루었다.

CAN은 독일의 Bosch사에서 자동차 내의 통신용 프로토콜로 개발하여 1993년에 ISO 11898로 국제 표준화된 비동기 직렬통신 방식이다[6]. 데이터 전송속도는 버스 길이에 따라 다르긴 하지만 40m기준으로 최대 1Mbps로 데이터를 전송할 수 있다. 신호의 전달은 두 버스라인(CAN\_H, CAN\_L)의 차이에 의해 이루어지기 때문에 외부 전자기파의 영향이 있어도 두 라인의 전압차는 일정하게 된다. 따라서 외부 전자기파의 영향에 상대적으로 덜 민감한 편이다. 지금부터 역진자 시스템을 제어하는데 고려해야할 CAN의 특징에 대하여 좀 더 세부적으로 살펴보겠다.

#### 2.1 통신 방식

##### 2.1.1 Broadcast 통신 방식

Broadcast 통신 방식은 네트워크를 구성하는 모든 노드들이 전송 노드의 프레임에 감지하는 방식을 말한다. 메시지 프레임을 받은 모든 노드에서는 그 프레임의 식별자를 통해, 이 프레임이 자신들이 받아들여야할 프레임인지 아닌지를 판단한다.

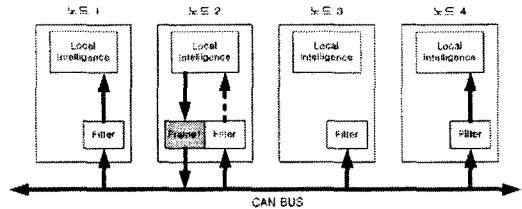


그림 1. Broadcast 통신 방식

그림 1은 Broadcast 통신 방식을 보여주고 있다. 이 경우에는 노드 2에서 메시지를 생성하고 다른 노드들은 이를 감지한다. 이때 생성된 메시지를 받아들일 수 있는 필터가 있는 노드 1과 노드 4만 이 메시지를 받아들인다.

##### 2.1.2 Remote request 방식

한 노드가 특정 데이터를 필요로 하는 경우 원격 프레임을 보내게 된다. 송신 노드가 원하는 데이터에 해당하는 메시지 식별자를 원격 프레임에 실어 버스에 보내면 그 메시지 식별자를 보내는 노드가 응답하게 된다. 그림 2에서 보면 노드 1이 RTR(Remote Transmission Requests) 메시지를 전송하면 노드 2에서 이에 대한 응답 메시지를 보내준다. 이때 모든 메시지는 Broadcast 통신 방식에서처럼 네트워크 상의 모든 노드로 전송된다.

## 2. CAN(Controller Area Network)

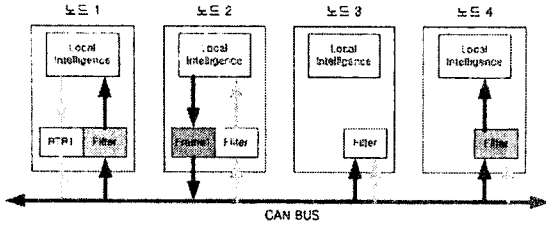


그림 2. Remote request 통신 방식

## 2.2 매체 접근 방식

CAN 프로토콜에서는 master와 slave가 따로 없기 때문에 만약 두 개 이상의 노드에서 동시에 메시지 전송이 이루어진다면 버스 중재가 필요하게 된다. 이러한 경우가 생기게 되면 각 메시지는 서로 식별자를 1비트씩 비교하여 우선순위가 높은 메시지는 전송되고 우선순위가 낮은 메시지들은 전송이 중단된다. 전송을 중단한 노드는 버스의 상태를 계속 감지하여, 버스가 다시 비활성 상태가 되면 자동적으로 다시 메시지의 전송을 시작한다.

이처럼 상대적으로 우선순위가 낮은 메시지는 자신의 차례가 돌아올 때까지 기다려야 하기에 CAN을 이용하는 본 시스템에서는 메시지들의 시간 지연 문제를 고려해야 한다. 이제 CAN을 이용하는 시스템에서 생기는 최악의 메시지 응답시간을 살펴보겠다.

## 3. 메시지 응답 시간

1995년에 K.Tindell등에 의해 제안된 방법으로 CAN 메시지 응답 시간을 구할 수 있다[4][9].

$$R_m = t_m + C_m \quad (1)$$

$$t_m = B_m + \sum_{\forall i: hp(m)} \left\lceil \frac{t_m + J_i + \tau_{bit}}{T_i} \right\rceil C_i \quad (2)$$

$$C_m = \left( \left\lceil \frac{54 + 8b_m}{4} \right\rceil + 67 + 8b_m \right) \tau_{bit} \quad (3)$$

$$B_m = \max_{\forall k: Nnode} (C_k) \quad (4)$$

식 (1), (2), (3), (4)에 의해 최악의 상황에서의 메시지 응답 시간을 구할 수 있다.

하나의 메시지가 전달되기 까지 걸리는 최대시간을 찾아내기 위해 몇몇 용어와 변수를 정의한다. 최악의 상황에서의 메시지 응답시간( $R_m$ )은 가장 높은 우선순위 메시지를 대기열에 넣는 것에서부터 그 메시지의 수락에까지 걸리는 시간을 말한다. 메시지의 차단시간( $B_m$ )은 한 스테이션이 버스가 비어있는 상태(idle)가 될 때까지 기다리는 시간을 말하고, 실제로 이 시간이 CAN 메시지 전송에서 가장 긴 시간을 차지한다. 한계시간( $D_m$ )은 같은 식별자를 갖는 다음번 메시지가 대기열에 놓이기 전에 앞의 메시지가 버스로 전송되어야 하는 시간을 말한다. 이는 대기열에서 이전의 메시지가 전송되기 전에 다음 번 메시지가 대기열에 놓여서 메시지가 겹쳐짐에 따라 이전의 정보가 손실되기 때문에 정해진 시간 개념이다. 주기( $T_m$ )은 메시지 m의 주기이다. 지터( $J_i$ )는 메시지를 대기열에 놓는데 있어서 상황에 따라 변화되는 시간을 말한다.  $b_m$ 은 메시지의 비트수를 의미한다.  $C_m$ 은

버스에서 물리적으로 메시지를 전송하는데 걸리는 시간을 의미한다.  $hp(m)$ 은 메시지 m보다 더 높은 우선순위를 가진 메시지들로 구성된 집합을 의미한다.  $Nnode$ 는 네트워크 상에 있는 모든 노드들의 집합을 의미한다.

$\tau_{bit}$ 는 1bit를 전송하는데 걸리는 시간을 의미한다. 위의 식에 있는  $\lceil x \rceil$ 는 x보다 더 큰 숫자 중에서 가장 작은 정수를,  $\lfloor x \rfloor$ 는 x보다 더 작은 숫자 중에서 가장 큰 정수를 취하는 수학적 기호이다. 최악의 상황에서의 메시지 응답시간은 대기열에서 지연되는 시간( $t_m$ )과 실제로 메시지가 전송되는 시간( $C_m$ )으로 구성된다.

그림 3과 같은 시스템의 구현 가능성을 살펴보기 위해 식(1)-(4)를 사용하여 시스템의 특성에 맞게 메시지의 최대응답시간을 구해보았다. 이 때 다음과 같은 조건을 적용하였다.

### 조건

- (1) CAN 버스에는 노드1, 노드2, 노드3 이 연결되어 있다.
- (2) 우선 순위는 노드1 > 노드2 > 노드3 순서이다.
- (3) 데이터의 크기는 각각 노드1 = 12bit, 노드2 = 16bit, 노드3 = 16bit 이다.
- (4) CAN 버스 길이는 40m 이하이고, 데이터 전송속도는 1Mbit/sec 이다.

표 1. 역진자 시스템에서의 최대 응답 시간 이론값

	노드1(us)	노드2(us)	노드3(us)
$R_m$	190.6	285.9	381.2
$t_m$	95.3	190.6	285.9
$C_m$	95.3	95.3	95.3
$B_m$	95.3	95.3	95.3

\* m은 노드번호,  $T_1, T_2 = 0.01$  sec (100Hz)

K.Tindell등에 의해 제안된 방법에 따라 최대 응답시간을 역진자시스템의 상황에 맞게끔 계산한 결과가 표 1과 같다. 시스템을 구성하는 노드들 중에서 최대 응답시간을 갖고 있는 노드1의 경우 시스템의 표본화 주기 0.01sec보다 매우 작은 최대 응답 시간을 갖고 있기에 이론상으로는 네트워크로 구성된 시스템이라도 별무리 없이 동작할 것임을 예상할 수 있다.

## 4. 실험 환경 소개

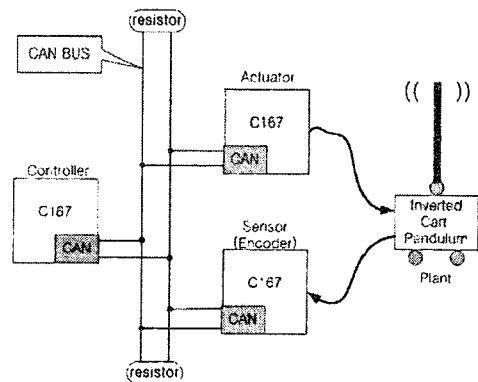


그림 3. 역진자 시스템의 실험 환경

CAN BUS는 물리적으로 비차폐된 두 가닥의 꼬임선을 이용하여 구성한다. 역진자 시스템의 엔코더 값을 버스

에 전송하고 이를 제어기가 받아 정해진 알고리즘대로 계산을 한 후 다시 버스에 전송한다. 그리고 제어기가 계산한 값을 버스로부터 받아 구동기가 역진자 시스템에 제어입력을 가해 주는 순서로 진행된다. 이 실험 환경에서 각 노드는 지멘스사의 C167 u-C를 사용했는데 자체적으로 CAN 모듈을 내장하고 있기 때문이다. 역진자 시스템의 차체의 위치 값 그리고 역진자의 각을 읽은 엔코더 값과 제어기에서 계산된 제어값이 CAN BUS를 통해 전달된다.

앞에서 계산된 메시지 최대응답시간을 바탕으로 역진자 시스템의 모의실험 결과를 살펴보고, 노드수가 증가할 경우를 대비하여 표본화 주기 시간 보다 더 느린 메시지 응답 시간이 걸릴 경우에 일어날 수 있는 시간 지연을 보상하는 제어 기법을 통하여 대비책도 마련해 보겠다.

### 5. 모의실험 결과

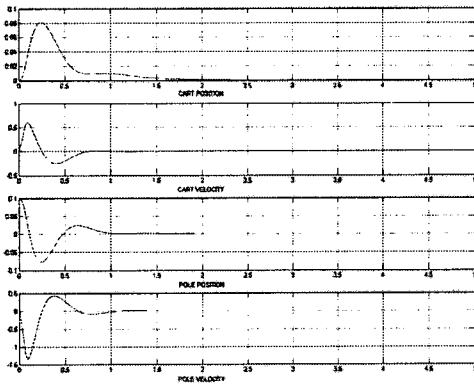


그림 4. 최대응답시간이 표본화주기보다 작은 경우

그림 4는 노드 3개로 구성된 역진자 시스템에 K.Tindell등이 제시한 최대응답 시간을 적용해 모의실험해본 결과이다. 동작 주파수가 100Hz이므로 센서 신호와 제어신호가 네트워크 버스를 통해 전달될 때 걸리는 시간을 충분히 감당 할 수 있음을 확인할 수 있다. 그러나 노드수가 증가하게 될 경우는 시스템 제어에 문제가 생기게 될텐데 가상적으로 표본화 주기의 2배만큼 메시지 전달이 지연 되는 경우를 살펴보고 이를 보상하는 방법을 고찰해 보겠다.

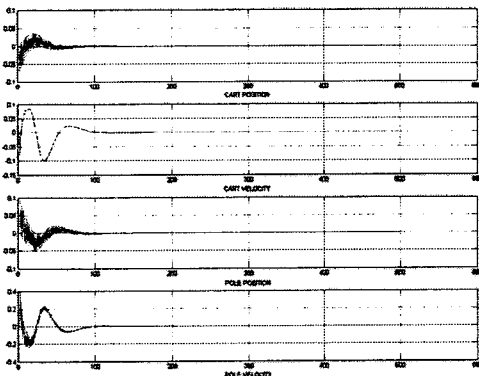


그림 5. Luck and Ray의 시간지연 보상방법적용

그림 5는 노드 수가 증가하는 경우를 가상적으로 설정

하여 메시지의 최대 응답시간이 커져서 메시지가 표본화 주기보다 2배 늦게 도달 되는 경우를 모의 실험한 것이다. 그림 4에서와 같은 조건으로 했을 때는 메시지가 표본시간의 2배만큼 지연 되었기에 발산 하였으나, Luck 과 Ray가 제안한 observer와 predictor를 사용하여 시간 지연을 보상하였을 때 성능은 저하되지만 시스템의 안정화를 이룰 수 있음을 확인 하였다[8].

### 6. 결 론

이 논문에서는 점차로 복잡해지는 제어 시스템에서 발생하는 배선문제, 시스템 유지, 보수 문제에 대한 대안책으로 제시되고 있는 NCS의 간단한 형태로 경성 실시간(Hard real-time) 시스템의 예인 역진자 시스템을 채택하여 CAN에서의 메시지 응답시간을 수식으로 계산하여 제어 가능성을 살펴보고 모의실험으로 검증해 보았다. 이후에는 시뮬레이션의 결과를 실험환경에 직접 적용해 보고 예상치 못했던 문제점들이 발생할 경우 부가적인 해결책을 모색하여 경성 실시간(Hard real-time) NCS에서의 안정화를 구현 할 것이다.

#### (참 고 문 헌)

- [1] W. Zhang, M. S. Branicky, and S. M. Philips, "Stability of networked control systems," IEEE Control Systems Magazine, pp.84-99, February 2001.
- [2] G. Walsh, H. Ye, and L. Bushnell, "Stability analysis of networked control systems," in Proc. Amer. Control Conf., San Diego, CA, pp.2876-2880, June 1999.
- [3] R. Raji, "Smart network for control," IEEE Spectrum, pp.49-55, June 1994.
- [4] K. Tindell, A. Burns, and A.J. Wellings, "Calculating controller area network (CAN) message response times," Contr. Eng. Practice, vol.3, no. 8, pp.1163-1169, Aug.1995.
- [5] J. Nilsson, "Real-time control systems with delays," Ph.D. dissertation, Dept. Automatic Control, Lund Institute of Technology, Lund, Sweden, January 1998.
- [6] Bosch, "CAN specification version 2.0", 1991.
- [7] Wolfhard Lawrenz, CAN System Engineering From Theory to Practical Application, Springer, 1997.
- [8] Luck, R. and A. Ray, "An observer-based compensator for distributed delays," Automatica, 26(5), pp.903-908, 1990.
- [9] M. Ellims, S. Parker, and J. Zurlo, "Design and analysis of a robust real-time engine control network," IEEE Micro, pp.20-27, July-August 2002.