

## CAN 기반 휴머노이드 로봇에서의 데이터 프레임 최소화

권선구, 허옥렬, 김진걸

인하대학교 전기공학과, 전기전자공학부

### Minimizing Data Frame in CAN Controller Area Network for Humanoid Robot

Sun-Ku Kwon, Uk-Youl Huh, Jin-Geol Kim

Dep. of Electrical Engineering, Inha Univ.

**Abstract** - The Controller Area Network (CAN) is being widely used for real-time control application and small-scale distributed computer controller systems. When the stuff bits are generated by bit-stuffing mechanism in the CAN network, it causes jitter including variations in response time and delay. In order to eliminate this jitter, stuff bit must be controlled to minimize the response time and reduce the variation of data transmission time. At first, this paper shows that conventional CAN protocol causes the transmission time delay. Secondly, this paper proposes the method to reduce the stuff bits by restriction of available identifier. Finally, data manipulation method can be reduced the number of stuff-bits in the data field. The proposed restriction method of ID and manipulating data field are pretty useful to the real-time control strategy with respect to performance. These procedures are implemented in local controllers of the ISHURO (Inha Semyung Humanoid ROBOT).

## 1. 서 론

휴머노이드 로봇이 인간의 생활공간에서 자율적인 보행을 수행하기 위해서는 자체적으로 보행에 필요한 데이터를 생성하거나 외부로부터 데이터를 받아들일 수 있는 통신 시스템 구축이 필요하다. 더불어 각각의 지역 제어기(local controller)로 이러한 데이터를 빠르고 정확하게 전송하기 위한 네트워크 구축에 관한 연구는 필수적이다. 휴머노이드 로봇에 관한 네트워크 기반 실시간 시스템에 관한 연구가 활발히 진행되고 있다[1][2].

CAN 프로토콜은 차량 내부의 복잡한 배선과 전자 장치를 통합하기 위해 개발되었다. 이후 프로토콜의 유용성으로 인해 로봇, 공정제어, 자동제어 등의 다양한 산업 분야에 적용되어 신뢰성 높은 네트워크를 구성하고 있다. 특히 저가의 하드웨어 장치로 고속의 통신 인터페이스를 제공할 뿐만 아니라, 데이터 프레임의 오버헤드(overhead)가 작기 때문에 빠른 응답 특성을 가지며, 크기가 작은 주기적인 데이터 전송에 적합하다. 또한 일반적인 CSMA/CD 방식과 달리, 식별자(identifier)를 이용한 충돌 방지와 전송 중재(arbitration) 기능을 갖고 있어서 우선순위 기반의 실시간 제어 네트워크 프로토콜로 널리 사용되고 있다[3][4].

기존 연구에서는 통신상의 지연시간과 전송 데이터 손실 및 오류에 따른 시스템 성능 및 안정도 분석을 통해 실시간 제어 가능성을 판단하고 평가하는 방법에 대한 연구가 진행되었다[1][5]. 또 메시지 스케줄링 기법으로서는 RMS(Rate Monotonic Scheduling) 또는 DMS(Deadline Monotonic Scheduling), EDS(Earliest Deadline Scheduling), MTS(Mixed Traffic Scheduling)을 적용하였다[2][3][6]. 독

립적 구조의 네트워크 상에서 비정기적으로 통신을 하는 경우, 중재 실패에 따른 지연시간의 증가나 오류가 큰 문제가 될 수 있다. 그러나 지역 제어기와의 정기적인 통신을 하는 휴머노이드 로봇의 네트워크 구조에서는 이런 문제가 수용이 가능하다. 대역폭 할당 등의 태스크 기반 메시지 스케줄링 방식에서 RMS/DMS 방식은 데드라인(deadline) 정보만으로 데이터의 중요도를 판단하기 어렵다. EDS/MTS와 같은 동적 ID 인가 방식은 ID 갱신으로 인한 CPU의 연산량이 증가하고, 시스템의 local clock의 동기화가 필요하게 된다[7].

다축 구동 시스템으로써의 휴머노이드 로봇은 많은 지역 제어기들이 제어 및 샘플링 주기에 따라 유기적으로 동작해야 한다. 따라서 메시지 전송 중재가 식별자를 통해 시스템 수준에서 이루어지는 CAN에서는 시간 지연(delay)을 고려한 제어 방식이나 동적 ID 인가 방식보다는 지연 요소를 제거하고 시간 지연을 일정하게 함으로써 네트워크의 신뢰도를 향상시키는 것이 효과적이다.

본 논문에서는 CAN에서 비트 스테핑(bit-stuffing)에 의해 발생하는 응답 시간의 변화인 지터(jitter)를 제거해서 제어 모듈 간의 응답 시간을 최소화하는 방법을 제안한다. 기존 ID 필드(identifier field)와 데이터 필드(data field)를 적용했을 때의 응답 시간과 시간 특성에 대해 분석하고, 스테프 비트(stuff bit)를 발생하는 개별 bit를 제한하는 ID 인가 방법과 스테핑을 방지하는 데이터 전송 방법을 제안한다.

이러한 방법은 주기적인 데이터 전송이 빈번한 네트워크, 제어 주기의 실시간 제약 조건이 존재하는 제어 시스템 및 표준 CAN 프로토콜에서 발전된 응용 분야에도 동일하게 적용할 수 있다.

## 2. 본 론

### 2.1 CAN 네트워크의 지연시간

CAN 메시지의 최악의 응답시간 분석은 프로세서 스케줄링에 대한 고정된 우선순위에 대한 응답시간을 기반으로 하고 있다. 각각의 고정된 우선순위를 가지고 발생하는 메시지는 트래픽 스트림의 집합으로 가정한다. 프로세서 스케줄링으로 스트림 집합 S를 얻는다. 각각의  $S_m \in S$  안에는  $[P_m, T_m, C_m]$ 으로 구성되어진다. 여기서  $P_m$ 는 우선순위,  $T_m$ 는 주기,  $C_m$ 는 스트림  $S_m$ 에서 보내지는 메시지의 최악의 전송 지연시간이다. 스트림에서 보내지는 CAN 메시지의 최악의 지연시간  $R_m$ 은 다음과 같이 정의된다.

$$R_m = J_m + q_m + C_m \quad (1)$$

여기서  $J_m$ 은 프레임의 큐잉 지터이며,  $q_m$ 은 큐잉 지연시간으로 주어지고 다음과 같다.

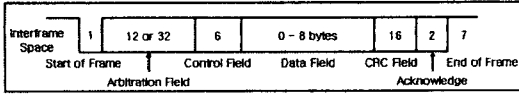


그림 1. CAN 데이터 프레임

$$q_m^{n+1} = B_m + \sum_{j \in hp(m)} \left[ \frac{q_m^n + J_j + \tau_{bit}}{T_j} \right] C_j + E(q_m + C_m) \quad (2)$$

$hp(m)$ 는  $m$ 보다 더 높은 우선순위의 시스템에서 발생하는 메시지 집합이며  $T_j$ 는 주어진 메시지  $j$ 의 주기,  $J_j$ 는 메시지  $j$ 의 큐잉 지터, 또  $\tau_{bit}$ 은 버스의 비트 시간으로 1Mbps의 전송 속도일 경우 1μs로 표현된다.  $E(q_m + C_m)$ 은 에러 처리에 소요되는 시간이며,  $B_m$ 은 식(3)과 같이 메시지  $m$ 보다 더 낮은 우선순위에 의해 지연되어질 수 있는 최대 시간으로 정의된다.

$$B_m = \max(C_k), \forall k \in lp(m) \quad (3)$$

여기서  $lp(m)$ 은 낮은 우선순위 메시지들의 집합이다.

### 2.2 Bit-Stuffing 방식

그림 1과 같이 스텝 비트는 CAN 프로토콜에서 전송 오류와 프로토콜 제어 목적으로 연속적인 6개의 동일한 비트(111111 또는 000000)를 허용하지 않기 위해 의도적으로 추가하는 6번째 비트를 말한다. 스텝핑은 SOF, arbitration field, control field, data field, CRC field 프레임 세그먼트에서 발생한다.

스텝핑 전과 후의 데이터 프레임의 비트 수는 식 (4), (5)와 같이 각각 표현되지만, 스텝핑 후에는 식 (5)와 같이 스텝 비트가  $\lfloor (g + 8s_m - 1) / 4 \rfloor$  비트 추가된다.

$$8s_m + g + 13 \quad (4)$$

$$8s_m + g + 13 + \left\lfloor \frac{g + 8s_m - 1}{4} \right\rfloor \quad (5)$$

여기에서  $g+13$ 은 제어 부분의 총 비트 수이고,  $g$ 는 표준 모드(standard format)에서 34이고, 확장 모드(extended format)에서 54이다.

따라서 스텝핑에 의한 전송 지연시간은 식 (6)과 같고, 결국 식 (1)에서의 응답 지연시간을 증가시키는 요인이 된다.  $C_m$ 은 버스 상에서 물리적으로 보내는 메시지  $m$ 이 버스 상에서 전송되는 시간으로 정의한다.

$$C_m = \left( 8s_m + g + 13 + \left\lfloor \frac{g + 8s_m - 1}{4} \right\rfloor \right) \tau_{bit} \quad (6)$$

여기서  $s_m$ 은 메시지의 바이트 단위 데이터 크기이며, 0-8[byte]의 값을 가진다.

전송 데이터가 크거나 저속 또는 확장 모드일 경우에 스텝핑에 의한 응답 지연의 영향은 더욱 크게 나타난다. 그림 2는 최악 조건 스텝핑에 의한 결과를 나타낸다.

### 2.3 제한적 ID 인가 및 데이터 필드 조작

스텝핑에 의한 응답 지연을 최소화하고 스텝 비트를 제거함으로써 시간 지연을 일정하게 하는 방법을 제안한

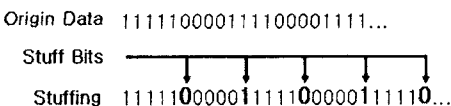
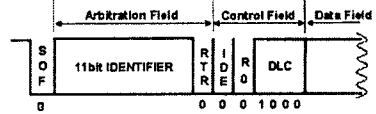
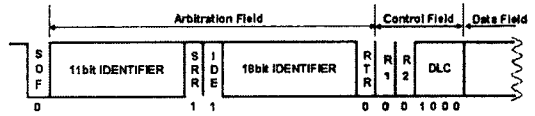


그림 2. 최악 조건에서의 bit-stuffing



(a) 표준 모드 (Standard Frame)



(b) 확장 모드 (Extended Frame)

그림 3. CAN 데이터 프레임(헤더)

다. 먼저 ID 영역에서 스텝핑을 방지하기 위해 제한된 ID를 사용한다. 그림 3과 같이 실제로는 arbitration field와 제어 필드(control field)에 '0'으로 고정된 비트가 존재함으로 인해 사용할 수 있는 ID의 제약 조건이 발생한다. 또한 지역 제어기와 주기적인 데이터 교환의 특성상 전송 데이터의 크기가 고정된다. 이로 인해 데이터 필드에서도 스텝핑 방지를 위한 제약 조건이 발생한다.

표 1은 스텝핑 제약 조건 하에서의 사용 가능 ID가 감소함을 나타낸다. 따라서 스텝핑 방지를 위해서는 스텝 비트를 발생하는 ID를 제한함으로써 적절한 ID를 정의하는 것이 필수적이다. 또한 데이터 필드의 8byte에 대해 스텝 비트를 최소화하도록 전송한다.

ISHURO는 총 21 자유도로서 다리부가 12 자유도를 가지고, 상체부가 9 자유도를 가지고 있다. 그림 4는 실험에 사용된 ISHURO의 지역 제어기 모듈을 나타낸다. 지역 제어기는 TMS320F2810 DSP를 탑재하고 LMD18200 모터 드라이버 및 eCAN 모듈이 장착되어 있다. 현재의 네트워크 시스템은 CAN 표준 모드인 CAN2.0A로 구성된다. 따라서 CAN을 이용하여 메시지를 송수신하기 위해서는 CAN 데이터 형식으로서 재정의해 주어야 한다. 메시지 재정의 과정에는 메시지의 우선순위를 지정하는 식별자의 정의와 하나의 메시지 프레임의 크기 정보가 포함된다.

표 2는 지역 제어기 1의 CAN메시지 식별자 및 데이터 크기를 나타내며, global sync. 메시지는 지역 제어기들이 동기화되어 움직일 때 사용되는 명령으로 우선순위가 가장 높게 설정되었다. Angle data는 보행 궤적 데이터 메시지를 뜻하며, encoder data와 sensor data는 각각 엔코더 데이터 메시지와 각종 센서로부터의 센서 데이터 메시지로 정의되었다. 또한 지역 제어기의 우선순위는 오름차순으로 1씩 증가하는 형태를 가진다. 즉 지역 제어기 2, 3, 4, 5의 angle data의 ID는 0x1404, 0x1408, 0x140C, 0x1410이 된다. 지역 제어기 1의 angle data의 ID는 0x1400이므로 F2810의 CAN 형식에서는 1 0100 0000 00으로 처리가 된다. 최상위 3 bit와 최하위 2bit는 무시하기 때문이다.

지역 제어기 1의 기존 angle data의 arbitration 필드는 101000000000이고, 제어 필드는 001000이지만, 실제로는 스텝핑에 의해서 1010000010000 0101000과 같이 전송되게 된다. 그림 2와 같이 연속되는 5개의 '0' 비트 때문에 스텝핑 룰에 의해 불필요한 비트가 전송된다. 제안하는 제한적 ID 인가 방법은 ID를 10100010001과 같이 인가하게 되면 스텝 비트에 의한 지터 발생을 방지할 수 있다.

8byte의 데이터 필드에 있어서도 스텝 비트는 발생하게 되고 최대 15개까지 발생하게 되며 1Mbps 속도로 전송할 경우에 15μs까지 데이터 전송시간이 지연되게 된다. 이런 현상은 0xF, 0x0 데이터를 전송할 때에 두드러지게 나타난다. 따라서 이들 데이터에 대해서는 0b1010과 0b0101 같은 형태로 각각 재정의 하고 수신부에서 다

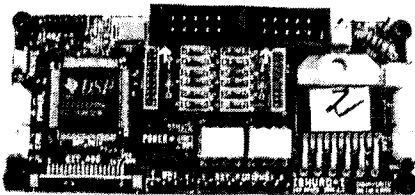


그림 4. 지역 제어기(Local Controller)

시 원래 데이터로 복구하는 방법을 사용한다. 0xA, 0x5와 같은 데이터와의 충돌을 방지하기 위해 보행 궤적 데이터에 중복되는지를 확인해야 한다. 현재는 4보를 위한 16초 보행 데이터를 전송하는 방식이지만, 주변의 환경을 인식해서 실시간 보행 궤적을 생성하고 수정하는 시스템에서는 문제가 발생할 여지가 있다.

다양한 메시지들이 제어 주기인 10ms 내에 21개의 지역 제어기가 데이터를 전송 및 수신을 완료하여야 하므로 제어 주기가 실시간 제약 조건이 된다. 예를 들어 angle data와 encoder data만 전송된다고 가정한다면 21개의 angle data가 지역 제어기로 전송되어야 하고, 동일한 방식으로 21개의 encoder data가 주제어기에 수신되어야 한다. 최소한 42개의 데이터가 제어 주기인 10ms 안에 처리되어야 한다. 따라서 데이터 전송마다 발생할 수 있는 지터를 줄임으로써 제어 주기 단축 및 응답 시간을 개선할 수 있다.

표 1. 스타프 비트 증가에 따른 ID 변화량

스타프 비트 수	0	1	2	3	4	5
사용 가능 ID 수 (표준/확장)	512 / 2,097, 152	64 / 65,536	4 / 8,192	./ / 4096	./ / 520	./ / 64

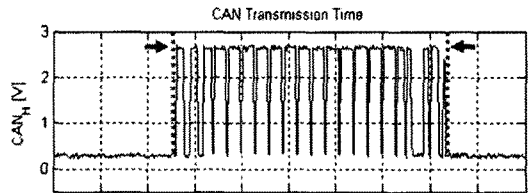
표 2. 지역 제어기 1의 메시지 정의

Message Name	ID	Description
Angle Data	0x1400	Receive
Encoder Data	0x1500	Transmission
Sensor Message 1	0x1800	Receive
Sync. Message 1	0x1600	Transmission
Sync. Message 2	0x1700	Receive
Global Sync.	0x12FC	Receive
Command Message 1	0x1A00	Receive
Command Message 2	0x1B00	Receive

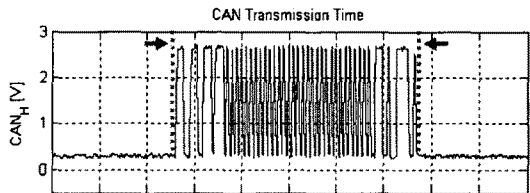
## 2.4 실험 및 결과 분석

그림 5에서 (a)는 기존에 정의된 ID와 데이터를 전송했을 때의 시간을 나타내고, (b)는 제안한 방법을 적용했을 때의 시간을 나타낸다. ID는 0x071A, 데이터는 0x0을 전송하는 기존의 형식에서의 전송시간은 115 $\mu$ s였지만, 제안한 방법을 적용했을 때에는 102 $\mu$ s가 소요되었다. 따라서 하나의 데이터 전송 시에 13 $\mu$ s 전송시간을 단축시킬 수 있고, 11.3% 응답시간 향상의 결과를 보여준다. 제어 주기의 제약 조건 속에서 많은 데이터를 전송해야 하는 CAN 적용 네트워크 기반의 제어 시스템에서 전송 시간 및 응답 시간을 단축시킬 수가 있다. 즉 샘플링 시간 및 제어 주기 단축을 통해 제어 성능을 향상시킬 수 있다.

스타프핑 기능은 비동기 방식인 CAN 프로토콜에서 수신 노드의 클럭의 동기화 정보로 활용된다. 그러나 연속 비트를 전송할 때에 유용하고 네트워크의 신뢰도를 향상시키지만, 다양한 패턴의 데이터 전송 시에는 오히려 응답 시간의 지연 요소로 작용한다. 실험을 통해 제안한



(a) 기존 데이터 전송 시간



(b) 제안한 방법에 의한 데이터 전송 시간

그림 5. 전송 시간 비교

방법이 스타프 비트를 제거하고 응답 시간이 향상됨을 확인할 수 있다.

## 3. 결 론

주 제어기가 전송하는 명령 값이 CAN 버스를 통해 지역 제어기에 수신되고, 지역 제어기가 전송하는 피드백 데이터가 제어 주기 내에 처리가 되어야만 실시간성이 보장된다. 네트워크의 안정성 보장 및 애러 검출을 위해 CAN 프로토콜에서 스타프핑 기능을 제공하고 있는데, 응답 시간의 변화를 유발시키는 요인이 된다. 이를 위해 스타프 비트를 제거함으로써 응답 시간의 가변 요인을 제거하여 시스템에 의한 지연 조건만 반영되도록 신뢰도를 향상시켰다. 휴머노이드 로봇의 네트워크 구성에 있어서 수많은 제어 모듈이 버스에 연결되고, 제어 주기 내에 많은 데이터가 전송 및 수신 되어야 한다. 따라서 제안하는 방법은 단일 전송보다는 주기적인 데이터 처리를 해야 하는 제어 시스템에 보다 효율적이고, 휴머노이드 로봇의 지역 제어기에 적용한 실험을 통해 제안하는 방법의 성능을 입증하였다.

## [참 고 문 헌]

- [1] 전충만, 김대원, "메시지 지연시간을 고려한 CAN 기반 피드백 제어시스템의 응답특성 분석", 대한전기학회 논문지, 제 51D권, 5호, pp. 190-196, 2002
- [2] 이병훈, 김홍렬, 김대원, "CAN기반 실시간 시스템을 위한 확장된 EDS 알고리즘 개발", 대한전기학회 논문지, 제51D권, 7호, pp. 294-301, 2002
- [3] G. C. Walsh and Y. Hong, "Scheduling of networked control systems", *IEEE Control Systems Magazine*, vol. 21, Issue 1, pp. 57-65, 2001
- [4] BOSCH, "CAN Specification version 2.0" pp. 42-44, sep, 1991
- [5] S. Punnekkat, H. Hansson and C. Norstrom, "Response time analysis under errors for CAN", *Real-Time Technology and Applications Symposium*, pp. 258-265, 2000
- [6] M. Natale, "Scheduling the CAN bus with earliest deadline techniques", *Real-Time Systems Symposium*, pp. 259-268, 2000
- [7] L.M. Pinho and F. Vasques, "Reliable real-time communication in CAN networks", *IEEE Transactions on Computers*, vol. 52, Issue 12, pp. 1594-1607, 2003