

CAN 기반 휴머노이드 로봇의 실시간 데이터 통신 구현

論 文

55D-1-5

Implementation of Real-Time Communication in CAN for a Humanoid Robot

權 宣 九[†] · 金 丙 倫^{*} · 金 眞 煥^{**} · 許 旭 烈^{***}

(Sun-Ku Kwon · Byung-Yoon Kim · Jin-Hwan Kim · Uk-Youl Huh)

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 bits must be controlled to minimize the response time and to reduce the variation of data transmission time. This paper proposes the method to reduce the stuff bits by restriction of available identifier and bit mask using exclusive OR operation. This data manipulation method are pretty useful to the real-time control strategy with respect to performance. However, the CAN may exhibit unfair behavior under heavy traffic conditions. When there are both high and low priority messages ready for transmission, the proposed precedence priority filtering method allows one low priority message to be exchanged between any two adjacent higher priority messages. In this way, the length of each transmission delays is upper bounded. These procedures are implemented as local controllers for the ISHURO(Inha Semyung Humanoid ROBOT).

Key Words : CAN, Distributed Control, Message Scheduling, Transmission Delay, Bit Mask, Round-Robin Acces

1. 서 론

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

휴머노이드 로봇은 구동축이 다수이기 때문에 하나의 고성능 CPU로 모든 구동축을 제어하는 방법 보다는 분산 제어 방법이 제어기의 구조를 간단하게 하고, 가격과 확장성에서 장점을 가진다[3]. 또 로봇의 이동성 때문에 범용 컴퓨터의 사용은 부피, 무게, 전력 사용량 등의 문제를 발생시킨다. 따라서 자유도마다 독립적인 제어기를 구성하고, 분산된 모듈간의 네트워크를 구축해야 한다. 통신 방법에 있어서도 병렬 방식은 전송 속도의 이점은 있으나 접속 회로는 복잡해진다. 결국 분산 제어 구조로 제어기를 모듈화하고, 개별 제어 모듈간의 데이터 통신에 직렬 방식인 CAN 프로토콜이 적합하다.

CAN 프로토콜은 차량 내부의 복잡한 배선과 전자 장치를 통합하기 위해 개발되었으나 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 데이터의 주기적인 전송에 적합하다. 또한 일반적인 CSMA/CD 방식과 달리 ID(identifier)에 의한 충돌 방지와 전송 중재(arbitration) 기능을 갖고 있어서 우선순위 기반의 실시간 제어 네트워크 프로토콜로 널리 사용되고 있다[4][5]. CAN 프로토콜이 사용하는 MAC 방식은 충돌이 발생했을 때 가장 우선순위가 높은 데이터 프레임을 전송하고, 나머지는 전송을 중지시키는 방식이다. 즉 고유한 ID로 메시지에 우선순위를 부여하여 메시지 간의 충돌을 방지하고 중재하는 역할을 한다[1][2].

ID를 할당하는 방법은 CAN의 응용 계층에서 중요한 부분이고 그 방법에 따라 다른 특성을 가진다. 그러나 네트워크에서 발생하는 메시지의 우선순위를 정확하게 ID로 표현하기 어렵다. CAN 프로토콜은 ID가 고정되어 있기 때문에 비주기적인 메시지나 우선순위가 낮은 메시지의 전송 지연 현상이 가장 큰 문제점이 될 뿐만 아니라, 네트워크 기반 휴머노이드 로봇의 안정성을 결정하는 중요한 요소가 된다. 이를 위한 기존 연구에서는 메시지 스케줄링에 의한 ID 할당 방법으로써 RMS(Rate Monotonic Scheduling), DMS(Deadline Monotonic), EDS(Earliest Deadline 혹은 EDF : Earliest Deadline First), MTS(Mixed Traffic)와 같은 방법들이 제안되었다[6][7][8][9]. 또한 통신상의 지연시간과 전송 데이터 손실 및 오류에 따른 시스템 성능과 안정도 분석을 통해 실시간 제어 가능성을 판단

[†] 교신저자, 學生會員 : 仁荷大學校 電氣工學科 碩士課程

E-mail : genuine@inhain.net

^{*} 學生會員 : 仁荷大學校 電氣工學科 博士課程

^{**} 正 會 員 : 仁荷工業專門大學 電氣情報科學科 助教授 · 工博

^{***} 終身會員 : 仁荷大學校 電子電氣工學部 教授 · 工博

接受日字 : 2005年 11月 1日

最終完了 : 2005年 12月 12日

하고 평가하는 방법에 대한 연구가 진행되었다[1][10].

그러나 다축 구동 시스템으로써 휴머노이드 로봇은 많은 지역제어기들이 제어 및 샘플링 주기에 따라 유기적으로 동작해야 한다. 그리고 메시지 전송 중재가 ID에 의해 시스템 수준에서 이루어지는 CAN에서는 전송 지연(transmission delay)을 고려한 제어 방식이나 동적 ID 할당 방식보다는 지연의 변동을 일정하게 하고 지연시간을 최소화함으로써 네트워크의 신뢰도를 향상시키는 것이 효과적이다. 즉 CAN에서 비트 스템핑(bit-stuffing)에 의해 발생하는 전송시간의 변화인 지터(jitter)를 제거해서 전송시간의 변동을 최소화할 뿐만 아니라 제어 모듈 간의 전송시간을 최소화하고자 한다. 스템프 비트(stuff bit) 발생을 제한하는 ID 제한 방법 및 비트 마스크(bit mask) 방법을 결합하여 데이터 조작(DM : Data Manipulation)을 이용한 전송 방법을 제안하고, ID 필드(field)와 데이터 필드(data field)에 적용했을 때의 전송시간과 시간 특성에 대해 분석한다.

또한 휴머노이드 로봇은 보행 동작을 하면서 swing leg과 support leg이 교차되고 이에 따라 개별 제어기와 메시지의 우선순위가 바뀌게 된다. 다시 말해 로봇의 보행에 관련된 메시지는 절대적으로 높은 우선순위를 가질 수 없고 우선순위가 유사한 경향이 있다. 전송 요청 순서도 고정할 수 없고, 무작위적인 전송 요청 순서에 따라 메시지의 전송 지연은 항상 변하게 된다. 따라서 유사한 우선순위를 가진 메시지의 공정한 전송 기회 부여와 전송 지연을 일정하게 하는 것이 네트워크 기반 휴머노이드 로봇 제어의 핵심이라고 할 수 있다.

대역폭 할당 등의 태스크 기반 메시지 스케줄링 방식에서 RMS/DMS 방식은 데드라인(deadline) 정보만으로 데이터의 중요도와 우선순위를 판단하기 어렵다. EDS, MTS와 같은 동적 ID 할당 방식은 ID 갱신으로 인한 연산량이 증가하고, 시스템의 local clock의 동기화가 필요하게 된다[11]. 무엇보다도 이들 방법은 우선순위가 높은 메시지의 빠른 전송을 보장하는 방법이므로 우선순위가 낮은 메시지는 계속적으로 전송이 저지되어 전송 지연시간이 길어지게 된다[3][12]. 본 논문에서는 동적 ID 할당 방식에 의한 불필요한 연산 시간과 전송 지연 문제를 위해 선행순위 필터링(PPF : Precedence Priority Filtering) 방법을 제안한다. 이를 통해 메시지 간의 전송 지연시간을 일정하게 하는 round-robin 방식을 구현하고 낮은 우선순위 메시지에 대해서도 공정한 전송기회를 부

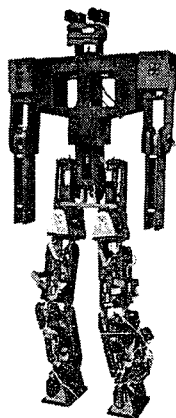


그림 1 휴머노이드 로봇 ISHURO
Fig. 1 The Humanoid robot "ISHURO"

여하여 제어의 안정성을 보장할 수 있다. 순차적으로 높은 우선순위에 낮은 우선순위의 메시지를 전송함으로써 전송 지연이 우선순위에 의해 편중되는 현상을 개선한다.

휴머노이드 로봇의 네트워크 구조에 적합한 CAN 프로토콜 개발을 위해 데이터 조작 방법과 선행순위 필터링 방법을 적용하고, 이러한 방법은 주기적인 데이터 전송이 빈번한 네트워크와 제어 주기의 실시간 제약 조건이 존재하는 제어 시스템 및 표준 CAN 프로토콜에서 발전된 응용 분야에도 동일하게 적용할 수 있다. 그림 1은 실험에 사용된 ISHURO (Inha Semyung Humanoid RObot)를 나타낸다.

2. CAN 프로토콜 분석

2.1 CAN 프로토콜의 지연시간

로봇 보행 데이터의 안정적인 전송을 보장하기 위해서는 CAN 프로토콜에 따른 데이터의 전송시간 분석이 필요하다 [12][13]. CAN 메시지의 최악 전송시간 분석은 메시지 스케줄링에 대한 고정된 우선순위에 대한 전송을 기반으로 하고 있다. 각각의 고정된 우선순위를 가지고 발생하는 메시지는 스트림(stream)의 집합으로 가정한다. CAN에서 스트림 s_m 을 전송하기 위한 최악 전송시간 R_m 과 큐잉 시간 q_m 은 다음과 같이 정의된다[3][10][13][14].

$$R_m = A_m + q_m + C_m \tag{1}$$

$$q_m^n = B_m + \sum_{j \in hp(m)} \left\lceil \frac{q_m^{n-1} + A_j + \tau_{bit}}{T_j} \right\rceil (C_j + 3\tau_{bit}) \tag{2}$$

여기서 A_m 은 메시지 m 의 큐잉 지터이고 메시지의 액세스 시간으로 정의되며, C_m 은 최악 프레임 시간이다. $hp(m)$ 는 메시지 m 보다 높은 우선순위를 갖는 메시지 j 의 집합이고, T_j 는 메시지 j 의 주기, A_j 는 메시지 j 의 액세스 시간, τ_{bit} 은 버스의 비트 시간이며 1비트의 전송 시간으로써 1Mbps의 전송 속도 일 경우 1 μ s이다. 또 $3\tau_{bit}$ 은 프레임 간의 오버헤드(overhead) 3비트이고, B_m 은 식 (3)과 같이 메시지 m 보다 낮은 우선순위의 메시지가 지연될 수 있는 최대 시간으로 정의된다.

$$B_m = \max_{\forall k \in lp(m)} (C_k) + 3\tau_{bit} \tag{3}$$

여기서 $lp(m)$ 은 낮은 우선순위를 갖는 메시지의 집합이다.

2.2 비트 스템핑 메커니즘

CAN 프로토콜에서 비트 스템핑은 연속되는 동일한 6개의 비트(111111 또는 000000)를 이용하여 오류 검출과 프로토콜 제어 목적으로 사용된다. 스템프 비트는 이런 특별한 비트의 패턴을 피하기 위해 의도적으로 추가하는 역전된 6번째 비트

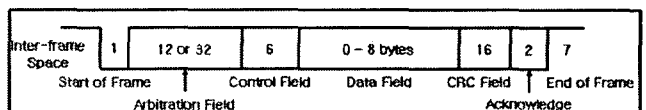


그림 2 CAN의 데이터 프레임
Fig. 2 The data frame in CAN

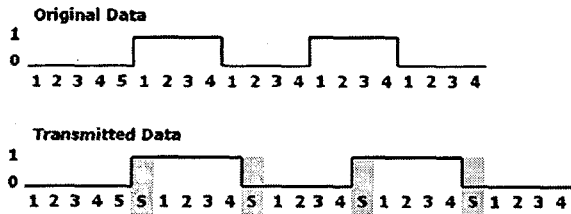


그림 3 최악 조건에서의 비트 스템핑
Fig. 3 The worst-case scenario by bit-stuffing mechanism

를 말하고 수신 측에서는 발생된 스템프 비트를 제거한다. 비트 스템핑은 그림 2의 데이터 프레임 중에서 SOF, 중재 필드(arbitration field), 제어 필드(control field), 데이터 필드, CRC 세그먼트에서 발생한다[14]. 그림 3은 비트 스템핑에 의한 최악 전송 지연의 경우를 나타낸다. 전송 데이터가 크거나 저속 또는 확장 모드(extended format)일 경우에 스템핑에 의한 전송 지연의 영향은 더욱 크게 나타난다. 식 (4)는 비트 스템핑 전의 프레임 시간 C_f 를 나타낸다. 이런 메커니즘에 의해 전송 데이터 프레임은 실제 프레임보다 길어지게 되고, 스템핑 후에는 프레임 시간이 식 (5)와 같이 길어진다.

$$C_f = (8s_m + g + 10)r_{bit} \quad (4)$$

여기에서 $g+10$ 은 제어 부분의 총 비트 수이고, g 는 SOF(1), 중재(arbitration : 12/32), 제어 필드(control : 6), CRC 세그먼트(15)를 합친 비트 수이고, 표준 모드(standard format)에서 34, 확장 모드에서 54이다. 10은 비트 스템핑 메커니즘이 적용되지 않는 CRC delimiter(1), ACK 필드(2), EOF(7)을 합친 비트 수이다. 또한 s_m 은 메시지의 바이트 단위 데이터 크기이고 0~8byte의 값을 가진다.

따라서 비트 스템핑 후의 최악 프레임 시간 C_m 은 식 (5)와 같고, 결국 식 (1)에서의 최악 전송시간을 변동시키는 요인이 된다. 따라서 최악 프레임 시간은 전송 메시지 m 이 버스 상에서 실제 물리적으로 전송되는 시간으로써 다음과 같이 정의한다.

$$C_m = \left(8s_m + g + 10 + \left\lfloor \frac{8s_m + g - 1}{4} \right\rfloor \right) r_{bit} \quad (5)$$

2.3 CAN 프로토콜의 전송시간

CAN 프로토콜에서 전송시간은 식 (1)과 그림 4에서 표현된 것처럼 큐잉 시간(queuing time) q_m , 액세스 시간(access time) A_m , 최악 프레임 시간(frame time) C_m 으로 구성된다. 큐잉 시간은 전송 요청 시점으로부터 미디어 접근 시간까지의 경과 시간을 뜻하고, 시간 분석과 제어가 어렵다. 액세스 시간은 MAC에 의해 미디어를 점유하기까지의 시간으로 전송 중재 메커니즘에 따라 가변적이다. 일반적으로 큐잉 시간과 액세스 시간의 합을 전송 지연 W 로 정의한다. 반면에 프레임 시간은 데이터를 버스(bus) 상에 전송할 때에 소요되는 시간을 말하고 데이터 프레임의 길이와 대역폭에 의해 변동되므로 일정하지 않다.

비트 스템핑 메커니즘에 의해 식 (5)에서 스템프 비트가 $\lfloor (8s_m + g - 1)/4 \rfloor$ 비트만큼 추가되고, 이로 인해 실제 전송시간은 식 (4)의 C_f 보다 [스템프 비트 수 $\times r_{bit}$] 만큼 지연이 되

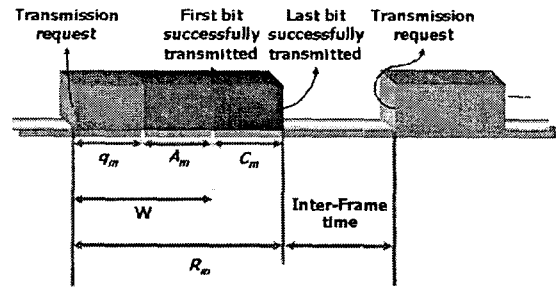


그림 4 비실시간 전송 순서
Fig. 4 The transmission sequence of non-RT

어 식 (5)의 C_m 만큼 소요된다. 즉 비트 스템핑은 최악 전송시간 R_m 의 변동과 증가를 유발하므로 데이터 조작 방법으로 최악 프레임 시간 C_m 을 줄이고 일정하게 하여 최악 전송시간 R_m 을 최소화하도록 한다.

더불어 우선순위 기반의 전송 중재 메커니즘에 의해 낮은 우선순위를 가진 메시지는 높은 우선순위의 메시지에 비해 상대적으로 액세스 시간 A_m 이 증가하고, 전송 지연 W 가 증가하게 된다. 제안하는 선행 순위 필터링 방법을 통해 낮은 우선순위 메시지의 전송 기회를 보장하여 액세스 시간 A_m 을 균등하게 배분하고 개별 메시지들의 평균 전송 지연 W 가 일정하게 한다. 즉 모든 메시지들의 액세스 시간이 균일하게 함으로써 메시지별 최악 전송시간 R_m 의 편차를 줄인다.

3. 데이터 조작 방법

본 논문은 앞서 언급한 비트 스템핑에 의한 전송시간의 변동을 최소화하고, 스템프 비트를 제거함으로써 전송시간을 최소화하는 데이터 조작 방법을 제안한다. 우선 프레임 헤더 영역인 ID 필드에서 비트 스템핑을 방지하기 위해 ID 제한 방법을 적용한다. 그림 5와 같이 실제로는 중재 필드와 제어 필드에 '0'으로 고정된 비트가 존재함으로 인해 사용할 수 있는 ID의 제약 조건이 발생한다. 또한 지역 제어기의 주기적인 데이터 교환의 특성상 전송 데이터의 크기가 정해져 있으므로 제어 필드의 하위 4비트도 고정이 된다. 이로 인해 데이터 필드의 최상위 비트에서도 스템핑 방지를 위한 제약 조건이 발생한다. 그러나 인위적으로 데이터 필드를 수정할 수 없으므로 전송 데이터에 비트 마스크를 하고, 수신 후에 다시 비트 마스크를 하여 원래의 데이터를 복구하는 비트 마스크 방법을 제안한다. 제안된 데이터 조작 방법은 ID 제한 방법과 비트 마스크 방법을 결합한 것을 뜻한다.

3.1 ID 제한 방법

먼저 ID 제한 방법은 ID 필드에서의 스템프 비트 발생을 방지한다. 그림 5와 같은 CAN 프레임 헤더 부분에서 특정

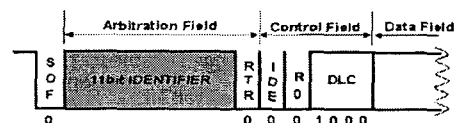


그림 5 CAN 데이터 프레임(헤더)
Fig. 5 The header of data frame in CAN

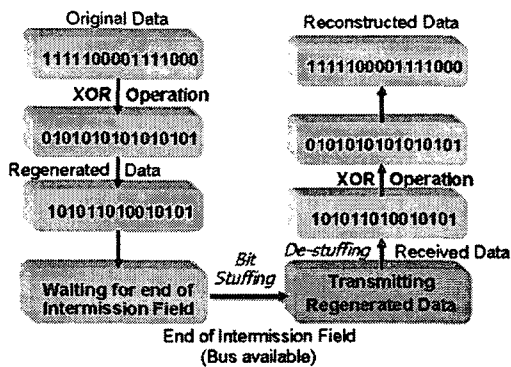


그림 6 비트 마스크 방법
Fig. 6 The procedures of bit mask method

ID를 제한함으로써 비트 스테핑을 회피하는 방법이다. SOF, RTR, IDE, R0 비트는 '0'으로 고정되어 있으므로 스테핑 비트를 방지하여 전송시간을 줄이기 위해서는 가능한 ID를 주의하여 선택해야 한다. 간단한 구조이지만 네트워크 이용률이 20% 향상되는 표준 모드에서 적용하였다[9]. 즉 중재 필드는 12비트로 설정되었다. 예를 들어 0000xxxxxx으로 메시지의 ID를 선택한다면, SOF를 포함하여 최상위 5개의 비트가 동일 비트가 되어 스테핑 비트가 발생하므로 사용하지 않는다. 또한 RTR, IDE, R0 비트가 '0'이므로 ID가 xxxxxxxx00인 경우도 제한된다.

ID 제한 방법을 적용할 경우의 최악 프레임 시간을 식 (5)를 이용하여 계산하면 SOF, ID 필드에서 스테핑 비트의 수는 $\lfloor (12-1)/4 \rfloor$ 로써 최대 3개까지 제거할 수 있고, 1Mbps 환경에서 3μs의 이득을 얻을 수 있다. 표준 모드에서 2048(2^{11})개의 ID를 할당할 수 있지만 ID 제한 방법을 이용할 경우 사용할 수 있는 ID는 1131개가 된다[14]. 최악의 경우 표 1에서와 같이 주제어기는 23개의 지역제어기와 6종류의 데이터를 전송하고, 23개의 지역제어기는 2종류의 데이터를 전송하게 된다. 이와 같이 8종류의 메시지가 폭주하더라도 184개의 ID가 필요하므로 표준 모드에 적용 가능하다. 또한 8byte 데이터 전송으로 인해 DLC는 0b1000으로 고정되기 때문에 데이터 필드의 최상위 2비트가 '0'이 되면 스테핑 비트가 발생하므로 이런 점까지 고려해야 한다. 그러나 스테핑 비트를 방지하기 위해 실제 전송되는 데이터를 인위적으로 바꿀 수 없으므로 데이터 필드에서는 비트 마스크 방법을 제한한다.

3.2 비트 마스크(bit mask) 방법

CAN의 데이터 필드에서는 연속되는 '1'이나 '0'을 제거하

표 1 지역 제어기 1의 메시지 정의

Table 1 The definition of message in the local controller 1

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

기 위해 비트 연산자를 이용하여 스테핑 비트의 수를 줄인다. 이를 위해 원래의 데이터 필드에 8byte XOR(exclusive OR) 연산을 통한 비트 마스크 방법으로 데이터 필드를 변환한다. 그림 6은 일정한 비트 패턴을 원래의 데이터 필드와 XOR 시키는 비트 마스크의 과정을 나타낸다. 전송을 할 데이터 필드와 8byte의 데이터(01010101...)를 XOR 연산하여 재구성된 데이터를 전송하고, 수신부에서는 수신된 데이터와 다시 XOR 연산을 하여 원래의 데이터 필드를 복구하는 방법이다. 일반적인 경우에 XOR 연산은 대부분의 프로세서에서 CPU 1클럭(clock) 내에 처리할 수 있는 명령어이다. 이 방법은 빠른 연산이 가능하고, 연산량에 비해서 전송시간의 이득이 큰 장점이 있다. 또 다양한 개발 환경에서도 XOR 연산을 통한 비트 마스크 방법을 쉽게 구현할 수 있다.

표준 모드에서 8byte 데이터를 전송할 때 데이터 필드에 비트 마스크 방법을 적용할 경우에는 식 (5)에서와 같이 스테핑 비트가 $\lfloor (8 \times 8 - 1) / 4 \rfloor$ 비트만큼 줄어들게 되므로 최대 16 μs까지 전송시간을 단축시킬 수 있다. 그러나 전송 데이터의 비트 패턴에 따라서 무작위로 결정된다.

4. 선행순위 필터링 방법

메시지 스케줄링에 의한 기존의 ID 할당 방법은 정적 ID 할당 방법과 동적 ID 할당 방법으로 구분된다. 정적 ID 할당 방법으로는 DMS, RMS 방식이 있는데, 구현이 간단하고 가능한 대역폭의 효율이 높다는 장점이 있다[6][7][15]. 그러나 정적 ID 할당 방법은 동적 ID 할당 방법에 비해 네트워크 이용률이 낮은 단점이 있다. 동적 ID 할당 방법은 EDS, MTS 방식이 있고 주기적 데이터와 비주기적 데이터가 혼재되어 있는 실시간 시스템을 위해 개발되었다[3][7][8][9][16]. EDS는 기존 방법에 비해 네트워크 이용률이 20% 향상되는 장점이 있으나, 시간이 경과함에 따라 ID 필드의 한계를 초과하고 우선순위 역전 현상이 발생하는 문제점이 있다. 또한 매번 중재에서 지속적으로 ID를 갱신(update)해야 하기 때문에 제어기의 연산량을 과중하게 만든다[8][9]. 반면에 MTS는 DMS와 EDS 방식을 결합한 형태이고, 네트워크 이용률은 다소 떨어지지만 연산량은 줄어드는 특성이 있다. 이처럼 네트워크 특성을 분석하고 장·단점을 비교하여 시스템에 적합한 ID 할당 방법을 결정하는 것은 어려운 일이다.

앞서 살펴본 바와 같이 메시지 스케줄링에 의한 ID 할당 방법은 메시지의 중요도와 우선순위 결정의 어려움, ID 필드의 제한, 프로세서 연산량 증가의 문제가 발생한다. 메시지 스케줄링은 배정능력과 실시간성 보장을 위해 연구되고 있지만 네트워크 시스템에 있어서 발생하는 총 전송 지연시간을 줄이거나 개선하기는 어렵다. 무엇보다도 ID가 갱신되면 ID의 고유성을 침해하므로 CAN 프로토콜에서는 사용이 제한된다[1][2][17]. 또한 CAN은 네트워크 상에서 개별 메시지의 우선순위를 ID로 할당하는데, 여러 메시지가 동시에 전송을 시작할 때 일어나는 충돌을 해결할 수는 있으나 개별 메시지의 최대 허용 지연시간 범위는 보장할 수 없다. 만약 네트워크 상에 많은 메시지가 전송을 대기하고 있다면 우선순위가 낮은 메시지는 전송 기회를 계속적으로 잃어버릴 수도 있다[18]. 우선순위가 낮은 메시지는 버스를 점유할 때까지의 액세스 시간이 길어져서 전송 지연시간이 증가하기 때문이다.

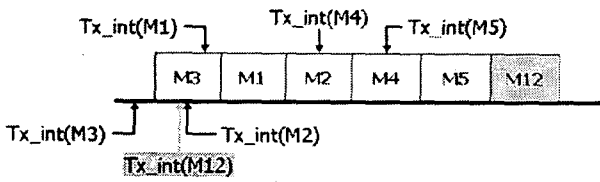


그림 7 최악 전송 지연 현상
Fig. 7 The worst-case scenario of transmission delay

우선순위 기반 네트워크 제어에 있어서 메시지의 우선순위에 따라 그림 7과 같이 낮은 우선순위의 메시지에 전송 지연이 편중되는 현상이 발생한다. 주제어기와 하부의 12개의 지역제어기의 데이터 통신을 묘사하기 위해 메시지의 우선순위는 $M1 > M2 > M3 > M4 > M5 > \dots > M11 > M12$ 순으로 높다고 가정한다. 그림 7에서, 메시지 12는 빠른 전송 요청에도 불구하고 우선순위가 높은 메시지에 의해 전송이 지연되면서 다른 메시지에 비해 전송 지연시간이 훨씬 커진다. 총 전송 지연 시간은 동일하지만 낮은 우선순위 메시지의 불공정한 전송 기회로 인한 전송 지연의 편중 현상을 개선하고자 한다.

앞서 보행 궤적 데이터는 유사한 우선순위를 가진다고 말했지만, 휴머노이드 로봇 시스템에는 보행 궤적 데이터 뿐 아니라 비주기적인 센서 정보와 제어 명령도 있으므로 최소한의 우선순위를 할당할 필요가 있다. EDS나 MTS 방식, CANopen에서는 상위 2~6비트를 우선순위 클래스(priority class)로 할당하고 있다[2][8][9][19]. 현재 묘사된 모든 메시지들을 하나의 클래스로 묶는다고 가정하면 ID는 최소한 8비트($2^8=256$)를 할당해야 한다. 따라서 그림 8과 같이 CAN 데이터 프레임의 헤더를 정의한다.

그림 9는 제안하는 선행순위 필터링 방법의 과정을 나타낸다. 우선순위는 메시지를 전송할 때에 각 전송 개체(메시지)에 할당되고, 선행순위는 대기 열이 생성되는 경우 대기 열 내에서의 전송순위로 구분되어 정의된다[18]. 즉, 우선순위는 메시지의 중요도 및 긴급성에 따라 고정적으로 할당하여 사용하는 것이며 선행순위는 대기 열에서의 전송순서를 나타낸다. 그림 7에서와 같이 낮은 우선순위와 높은 우선순위의 메시지가 대기 열에서 전송을 대기하는 과정에서 선행순위인 높은 우선순위의 메시지가 중재에 성공하여 전송이 된다. 선행순위 필터링 방법은 대기 열에서 선행순위를 가지는 높은 우선순위의 메시지를 필터링하여 순차적으로 낮은 우선순위를 가진 메시지가 전송되도록 하는 방법이다. 제안하는 방법은 버스가 휴면(idle)인 상태에 대해서는 적용하지 않고, 메시지들이 대기 열에 진입한 경우에만 적용이 된다. 선행순위 필터링 방법을 적용한 후에 비트 마스크 방법이 실행된다.

먼저 선행순위 필터링 방법은 전송 요청이 들어오면 우선순위 클래스 3비트와 ID 8비트를 분리하여 저장해둔 후, 클래스 값을 참조하여 해당 클래스를 결정한다. 클래스가 결정

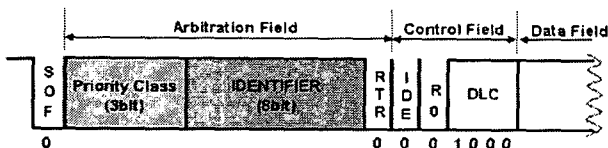


그림 8 CAN 데이터 프레임의 재정의
Fig. 8 The re-definition of data frame in CAN

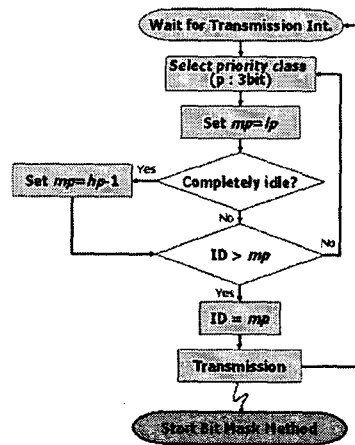


그림 9 선행순위 필터링 방법
Fig. 9 The procedures of precedence priority filtering method

이 되면 과거 전송된 메시지의 ID가 저장되는 변수인 mp를 가장 낮은 우선순위로 lp(11111111)로 초기화를 시킨다. 버스가 휴면 상태이면 mp를 '0(hp-1)으로 초기화하고, 이 때는 $ID > mp$ 를 만족하므로 무조건 전송을 시작하며 전송 요청을 받은 클래스와 ID를 다음 전송 요청을 받을 때까지 비교 값으로 저장한다. 만약 메시지 전송으로 인해 버스가 휴면 상태가 아니라면 전송된 메시지의 ID인 mp와 전송 요청을 받은 메시지의 ID 값을 비교하여 우선순위가 낮은 경우에 전송 요청을 수락하여 대기 열에 해당 메시지를 보낸다. 이런 과정으로 순차적으로 우선순위가 낮은 메시지들을 전송하고, 도중에 우선순위가 높은 메시지가 전송 요청을 보내면 버스가 휴면 상태가 될 때까지 대기시켜서 대기 열에 있는 낮은 우선순위의 메시지의 전송이 완료된 후에 전송한다. 그림 10은 선행순위 필터링 방법에 의해 그림 7의 문제점이 개선된 예를 보여준다. 낮은 우선순위의 메시지에 대해서도 전송 기회를 보장하고, 전송 지연의 편중 현상을 개선할 수 있다. 따라서 제안한 방법을 통해 round-robin access를 구현하게 된다.

4. 성능 평가 및 결과 분석

그림 11은 데이터 조작 방법을 적용하여 단일 메시지를 전송할 때의 CANH-CANL의 버스 신호 변화를 측정된 실험 결과이고, 메시지의 SOF부터 ACK 비트까지의 신호를 측정된 파형이다. 결과적으로 데이터 조작 방법은 121μs에서 102μs로 19μs만큼 전송시간을 단축시킬 수 있음을 보여준다. 그림 11 (a)와 (c)는 최악 조건에서 ID 제한 방법만을 적용했을 때의 결과를 보여주고, 스테프 비트를 3개 제거하여 121μs에서 118μs까지 3μs의 이득을 얻을 수 있다. 동일하게 그림 11 (b)와 (d)에서도 3μs만큼 단축되었다. 또한 그림 11 (a)와 (b),

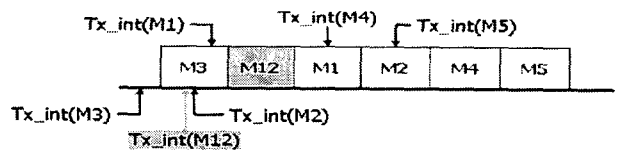


그림 10 선행순위 필터링에 의한 전송의 예
Fig. 10 The case study of precedence priority message filtering

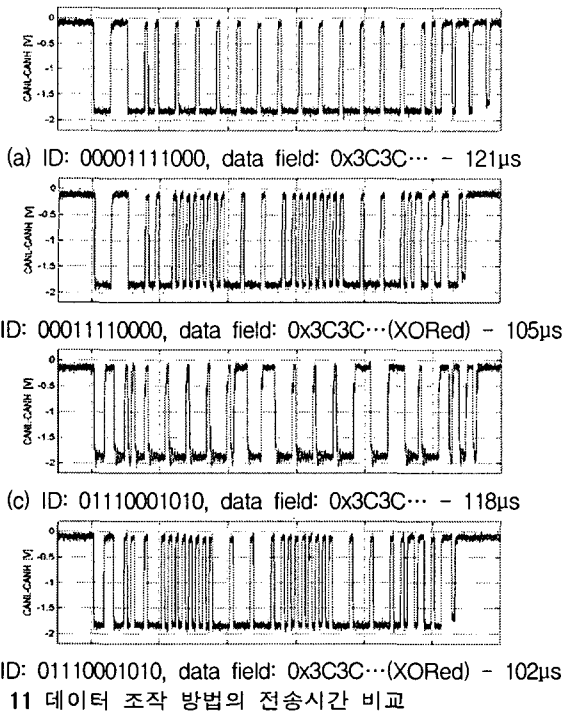


그림 11 데이터 조작 방법의 전송시간 비교
 Fig. 11 The message transmission time of data manipulation

(c)와 (d) 각각은 비트 마스크 방법에 의해 전송했을 때의 16 μ s의 전송시간이 단축되는 결과를 나타낸다. 따라서 ID 제한 방법과 비트 마스크 방법을 결합하여 그림 11 (d)처럼 최대 19 μ s(15.7%)만큼 전송시간을 단축할 수 있다.

추가적으로 휴머노이드 로봇의 주제어기와 하체 12개 지역 제어기의 네트워크 환경을 구현하기 위해 12개 메시지에 대해 실험하였다. 그림 12에 나타난 바와 같이 데이터 조작 방법은 메시지 전송이 많은 네트워크에서 유용하다. 하나의 메시지 전송에서 얻어지는 전송시간의 단축 효과가 누적되어 다량의 메시지를 전송할 때 효율적이다. 짧은 연산 시간에 비해 메시지의 전송시간을 단축시키는 효과가 크고, 전송시간의 지터를 최소화하여 네트워크의 신뢰성을 보장할 수 있다. 더불어 전송시간의 단축을 통해 제어 주기의 제약 조건 속에서 많은 데이터를 전송해야 하는 제어 시스템에서 샘플링 시간 및 제어 주기를 단축하여 제어 성능을 향상시킬 수 있다.

또한 그림 12, 13의 기존 CAN과 제안된 데이터 조작과 선

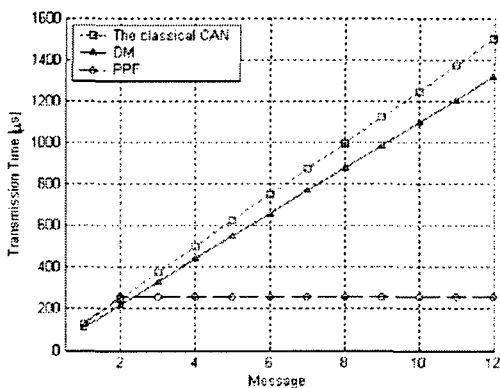


그림 12 가장 낮은 우선순위 메시지의 전송시간
 Fig. 12 The transmission time of the lowest priority message

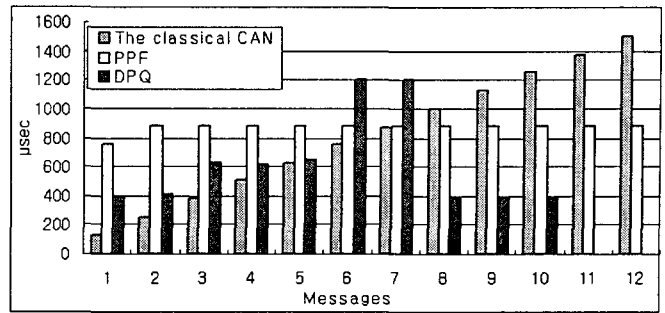


그림 13 PPF 방법 적용시 평균 전송 지연시간
 Fig. 13 The average transmission delay time of PPF method

행순위 필터링 방법을 적용한 결과에서 볼 수 있듯이 네트워크 부하가 커지거나 메시지가 많아질수록 전송 지연 시간은 점점 증가하고, 각각 다른 ID를 가지는 메시지의 전송 지연은 부하가 커지면 불공평하게 배분된다. 기존 CAN의 경우 전송 지연이 메시지마다 불공평하게 작용하는 반면에 선행순위 필터링 방법을 적용한 CAN은 전송 지연시간이 부하의 양만큼 공평하게 발생한다. 그림 12에서는 네트워크의 부하가 커질수록 공정한 메시지 전송 기회 부여와 전송 지연시간의 균등화를 위해서 선행순위 필터링 방법이 효율적임을 보여준다. 그림 13은 기존 CAN 방식과 선행순위 필터링, DPQ 방법을 적용했을 때 평균 전송 지연시간을 비교한 결과이다 [18]. 제안한 방법의 표준편차는 35인 것과 대조적으로 기존 CAN은 432이고, 뿐만 아니라 DPQ 방법은 310임을 비교해 볼 때에 보다 균일한 평균 전송 지연시간을 얻을 수 있으며, 낮은 우선순위 메시지의 공정한 전송기회를 보장할 수 있다.

5. 결 론

본 논문에서는 실시간 제어 시스템에 유용한 CAN 프로토콜을 휴머노이드 로봇의 네트워크로 활용 및 최적화하기 위해 주요한 특징을 살펴보았다. 네트워크의 안정성 보장 및 여러 검출을 위해 CAN 프로토콜에서 비트 스타핑 기능을 제공하고 있는데, 전송시간의 변동을 유발시키는 요인이 된다. 먼저 스타프 비트를 제거하기 위해 ID 제한 방법과 비트 마스크 방법을 결합한 데이터 조작 방법을 제안하였다. 이를 통해 전송시간의 가변 요인을 제거하고 시스템에 의한 지연 조건만 반영되도록 신뢰도를 향상시켰다. 또한 기존에 제안된 메시지 스케줄링 방식에 의한 ID 할당 방법들의 장·단점을 보였고, 기존 방법의 한계를 극복하고 휴머노이드 로봇에 적합한 선행순위 필터링 방법을 제안하였다. 대기 열에 있는 낮은 우선순위 메시지의 전송을 보장하기 위해 우선순위가 순차적으로 낮아지도록 선행순위의 메시지 전송을 대기시키는 방법이다. 이를 통해 전송 지연시간을 일정하게 만들어주며 휴머노이드 로봇의 네트워크와 같이 안전성을 중시하는 데이터 특성을 반영할 수 있도록 하였다. 제안하는 방법은 단일 전송보다는 주기적인 데이터 처리를 해야 하는 제어 시스템에 보다 효율적이고, 휴머노이드 로봇과 비슷한 데이터 특성을 가진 시스템에 유용하게 적용할 수 있을 것이다.

감사의 글

본 연구는 한국 과학재단 목적기초연구 (R01-2003-000-10364-0) 지원으로 수행되었음.

참 고 문 헌

[1] 전종만, 김대원, "메시지 지연시간을 고려한 CAN 기반 피드백 제어시스템의 응답특성 분석", 대한전기학회 논문지, 제51D권, 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] K.W. Tindell, H. Hansson and A.J. Wellings, "Analysing real-time communications : controller area network (CAN)", in *Proceeding of Real-Time Systems Symposium*, pp. 259-263, 1994.

[4] G.C. Walsh and Y. Hong, "Scheduling of networked control systems", *IEEE Control Systems Magazine*, vol. 21, No. 1, pp. 57-65, 2001.

[5] *CAN Specification version 2.0*, BOSCH, pp. 42-44, sep, 1991.

[6] N.C. Audsley, A. Burns and A.J. Wellings, "Deadline monotone scheduling theory and application", *IFAC J. Contr. Eng. Practice*, vol. 1, No. 1, pp. 71-78, 1993.

[7] M.D. Natale, "Scheduling the CAN bus with earliest deadline techniques", in *Proceeding of Real-Time Systems Symposium*, pp. 259-268, 2000.

[8] K.M. Zuberi and K.G. Shin, "Non-preemptive scheduling of messages on controller area network for real-time control applications", in *Proceeding of Real-Time Technology and Applications Symposium*, pp. 240-249, 1995.

[9] K.M. Zuberi and K.G. Shin, "Design and implementation of efficient message scheduling for controller area network", *IEEE Transactions on Computers*, vol. 49, No. 2, pp. 182-188, 2000.

[10] S. Punnekkat, H. Hansson and C. Norstrom, "Response time analysis under errors for CAN", in *Proceeding of Real-Time Technology and Applications Symposium*, pp. 258-265, 2000.

[11] L.M. Pinho and F. Vasques, "Reliable real-time communication in CAN networks", *IEEE Transactions on Computers*, vol. 52, No. 12, pp. 1594-1607, 2003.

[12] K.W. Tindell, A. Burns and A. Welling, "Calculating Controller Area Network(CAN) Message Response Time", *Control Engineering Practice*, vol. 3, No. 8, pp. 1163-1169, 1995.

[13] S.H. Hong and W.H. Kim, "Bandwidth allocation scheme in CAN protocol", in *IEE Proceeding of Control Theory and Applications*, vol. 147, No. 1, pp. 37-44, 2000.

[14] T. Nolte, H. Hansson and C. Norstrom, "Probabilistic worst-case response-time analysis for the controller area network", in *Proceeding of Real-Time and Embedded Technology and Applications Symposium*, pp. 200-207, 2003.

[15] B. Kao and H. Garcia-Molina, "Deadline Assignment in a Distributed Soft Real-Time System," *IEEE Transactions on Parallel and Distributed Systems*, vol. 8, No. 12, pp. 1268-1274, 1997.

[16] K.G. Shin, "Real-time communications in a computer-controlled workcell", *IEEE Transactions on Robotics and Automation*, vol. 7, No. 1, pp. 105-113, 1991.

[17] 구자봉, 허욱렬, 김진걸, 김병륜, "CAN내장 휴머노이드 로봇에 대한 진보된 우선순위 적용", 대한전기학회 논문지, 제 53D권, 10호, pp. 714-719, 2004.

[18] 최호식, 이장명, "CAN 네트워크상의 효율적인 대역 할당을 위한 분산 선행대기 열 기법", 제어 자동화 시스템 공학 논문지, 제 10권, 11호, pp. 1058-1064, 2004.

[19] *CANopen Application Layer and Communication Profile*, CiA Draft Standard DS301, Rev. 4.01, June 2000.

저 자 소 개



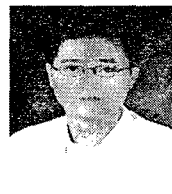
권 선 구 (權 宣 九)

2000년 인하대학교 전기공학과 졸업.
현재 동 대학원 전기공학과 석사과정.
Tel : 032-860-7394
Fax : 032-864-6442
E-mail : genuine@inhain.net



허 욱 렬 (許 旭 烈)

1978년 서울대학교 전기공학(공학석사).
1982년 서울대학교 전기공학(공학박사). 1985~1986 미국 일리노이대 방문교수. 1997~현재 대한 전기학회 이사 및 부회장. 2004~현재 한국과학재단 전기정보분과 전문위원. 현재 인하대학교 전자전기공학부 교수
E-mail : uyhuh@inha.ac.kr



김 병 륜 (金 丙 倫)

1993년 인하대학교 대학원 전기공학과(공학석사). 2002년 Univ. of Pittsburgh, Department of information Science for telecommunication(통신석사). 현재 인하대학교 대학원 전기공학과 박사과정.
E-mail : tycoon482@naver.com



김 진 환 (金 眞 煥)

1996년 인하대학교 대학원 전기공학과(공학박사). 1997년~2001년 두원 공과대학 디지털 정보제어과 교수. 2001~현재 인하공업전문대학 전기정보과학과 조교수
Tel : (032)-870-2193
E-mail : jhkim@inhac.ac.kr