CAN내장 휴머노이드 로봇에 대한 진보된 우선순위 적용

論 文
53D-10-6

An Improved Priority Application for Humanoid Robot in the Controller Area Network(CAN)

具 滋 俸[†] · 許 旭 烈[*] · 金 振 傑[**] · 金 丙 倫[***]
(Ja-bong Ku · Uk-youl Huh · Jin-geol Kim · Byung-yoon Kim)

Abstract – The Controller Area Network (CAN) is being widely used in real-time control applications such as automobiles, aircraft, and automated factories. Unfortunately, CAN, in its current form, is not able to either share out the system bandwidth among the different devices fairly or to grant an upper bound on the transmission times experienced by the nodes connected to the communication medium as it happens, for instance, in the token-based networks. In this paper, we present the message scheduling for CAN, based on the distributed control scheme to integrate actuators and sensors in a humanoid robot. Besides introducing the new algorism, this paper also presents some performance figures obtained using a specially developed software simulator, while the behavior of the new algorism is compared with the traditional CAN systems, in order to see how effective they are.

Key Words : CAN, Bandwidth, Distributed Control, Message Scheduling

1. 서 론

인간과 상호작용을 통하여 인간의 명령 및 감정을 이해하고, 반응하며 정보 통신 기술을 바탕으로 인간에게 다양한 서비스를 제공하는 지능형 로봇인 휴머노이드 로봇의 연구가 활발히 이루어지고 있다.[1]

휴머노이드 로봇은 다수의 관절을 가져야 하며 관절 구동용 엑츄에이터로는 주로 전기 모터를 사용하고 있다. 따라서 휴머노이드 로봇은 필연적으로 모터가 많이 사용되며, 이러한 모터를 제어하기 위한 제어기와 주변 인터페이스 부분도 복잡하게 되고, 개발 비용도 많이 들게 된다. 이런 문제점을 해결하기 위해서 또한 휴머노이드 로봇이 자율적인 보행을 수행하기 위해서는 자체적으로 보행에 필요한 데이터를 생성하거나 외부로부터 데이터를 받아들일 수 있는 통신 시스템 구축이 필요하다. 휴머노이드 로봇은 제어 시 많은 양의 정보를 필요로 하나 몸체의 크기가 제한되어 있으므로 PC를 몸체 외부에 두고 데이터를 무선으로 통신하는 구조를 가진다. 특히 PC에서 받은 정보를 각각의 하위 모듈로 전달하여 주는 네트워크 구조상 실시간성 분석은 로봇이 인간과 같은 유동적인 행동을 하기 위해서 가장 먼저 수행 되어야 할 작업일 것이다[2],[3].

본 논문에서는 휴머노이드 로봇에 적용하기 위한 CAN 네트워크 시스템의 구조를 제안하며 휴머노이드 로봇의 특성에 맞는 CAN 응용층 설계로서 시스템이 동작하면서 우선순위를 가지게 되는 ID(Identifier) 인가 알고리즘을 제안한다. 또한 응용층 설계에 따른 네트워크 성능을 분석한다. 특히, CAN 프로토콜은 ID가 고정되어 있기 때문에 낮은 우선순위를 가진 메시지의 전송 문제가 큰 문제점 이다. 이 점은 네트워크 기반 휴머노이드 로봇의 안전성을 결정하는 매우 중요한 요소이며 큰 영향을 미치는 부분이다. CAN의 응용계층에서 가장 골자가 되는 것은 CAN의 가장 중요한 특징이자 장점인 ID를 배정하는 방법이다. ID를 실제 시스템에서 발생하는 데이터들과 연결하고 인가하는 방법에 따라 같은 CAN을 사용한다 하더라도 매우 다른 특성을 가지게 된다. 그리하여 CAN의 장점을 최대한 살릴 수 있는 ID 인가방법을 찾기 위하여 지금까지 DM(Deadline Monotonic), ED(Earliest Deadline), MTS(Mixed Traffic Scheduling)와 같은 방법들이 제안되었다[4]. ID 인가 알고리즘은 스테이션이 프레임을 전송하기 위해 처음 시도한 순간부터 경과된 시간에 대하여 비례적으로 점점 증가되어가는 각 스테이션의 우선순위를 표시해주는 것이 기본적인 생각이다. 고정된 ID로의 전송과 제안된 ID 인가 알고리즘으로의 전송을 모의실험을 통하여 비교한다.

본 논문의 구성은 먼저 2장에서는 CAN 통신 지연에 대한 수학적 모델링을 보이고 3장에서 휴머노이드 로봇의 네트워크 시스템을 설계하며 4장에서는 기존의 응용계층의 설계 방법들과 그에 분석을 보이며 5장, 6장에서는 ID 인가 알고리즘을 제안하고 기본 CAN과의 비교 분석을 모의실험을 통하

† 교신저자, 正會員 : 仁荷大學 電氣工學科 碩士課程
E-mail : snoopyace@hanmail.net
* 正 會 員 : 仁荷大學 電氣工學科 敎授 · 工博
** 正 會 員 : 仁荷大學 電氣工學科 敎授 · 工博
*** 正 會 員 : 仁荷大學 電氣工學科 博士課程
接受日字 : 2004年 6月 16日
最終完了 : 2004年 8月 23日

여 실행한다. 7장에서는 결과와 앞으로 연구 되어야 할 방향을 조명한다.

2. CAN통신 지연에 대한 수학적 모델링

이 장에서는 CAN 메시지의 최악의 응답시간 계산을 수학적 모델링을 통해 분석한다. 이 분석은 프로세서 스케줄링[5]에 대한 고정된 우선순위[6] 응답시간을 기반으로 하고 있다. 응답시간의 계산은 최악의 경우 프레임의 큐잉 패턴(queueing pattern)의 범위 안에서 이루어진다. 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_m는 우선순위, T_m는 주기, C_m는 스트림 S_m에서 보내지는 메시지의 최악의 경우 전송 지연시간이다[7],[8]. 스트림에서 보내지는 CAN 메시지의 최악의 경우 응답 지연시간 R_m은 다음과 같이 정의되어진다.

$$R_m = J_m + w_m + C_m \tag{1}$$

여기서 J_m은 프레임의 큐잉 지터이며 w_m은 큐잉 지연시간으로 주어지고 다음과 같다.

$$w_m = B_m + \sum_{j \in hp(m)} \left[\frac{w_m + J_j + \tau_{bit}}{T_j} \right] C_j \tag{2}$$

hp(m)는 m보다 더 높은 우선순위의 시스템에서 발생하는 메시지 집합이며 T_j는 주어진 메시지 j의 주기, J_j는 메시지 j의 큐잉 지터, 또 τ_{bit}은 버스의 비트 시간으로 1Mbps의 전송속도일 경우 1μs로 표현 된다. B_m은 메시지 m보다 더 낮은 우선순위에 의해 지연 되어질 수 있는 최대 시간으로 정의되며 다음과 같다.

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

여기서 lp(m)은 낮은 우선순위 메시지들의 집합이다.

CAN은 메시지당 47비트의 오버헤드를 가지고 있고 5비트의 넓이로 스터핑 비트(stuffing bit)가 들어가며 단지 오버헤드의 47비트 중 34비트만 스터핑 조건으로 한다. 식(1)의 C_m은 버스 상에서 물리적으로 보내는 메시지 m이 버스 상에서 전송되어지는 시간으로 다음과 같이 정의 할 수 있다.

$$C_m = \left(\left[\frac{34 + 8s_m}{5} \right] + 47 + 8s_m \right) \tau_{bit} \tag{4}$$

여기서 s_m은 메시지의 데이터 크기이며 바이트 단위이다.

3. 휴머노이드의 네트워크 설계

인간의 관절 기능에서 움직일 수 있는 범위는 로봇 디자인에서 매우 중요한 부분이다[1]. Fig. 1은 로봇의 하체 부분의 각각의 관절 부분에 구성되어있는 자유도를 나타낸 그림이다.

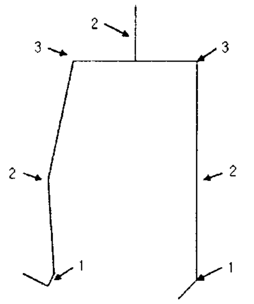


그림 1 휴머노이드 로봇의 하위 부분의 관절 자유도

Fig. 1 The location of the joints in the lower part of humanoid robot, indicating the degrees of freedom in each joint

여러 개의 자유도로 구성되어있는 곳은(예를 들어 hip 부분) 구형의 관절을 사용하는 것보다 짧은 링크들로 연결하여 순차적으로 구현한다. 전체 14개의 자유도를 가지고 있으며 하나의 자유도마다 모듈화하고 분산형 구조로 만들며 각각의 모듈들은 CAN 프로토콜을 사용하여 통신한다.

3.1 주제어 부

휴머노이드 로봇의 네트워크 구조는 Fig. 2와 같이 구성되어진다. 로봇의 몸체 밖에 있는 PC 시뮬레이터에서 보내지는 정보는 802.11b 무선 LAN 프로토콜을 사용하여 통신하며 받는 쪽을 주제어기라 정의한다. 이 주제어기는 인텔사의 Xscale PXA255를 기반으로 한 하이버스사의 리눅스 보드를 중심으로 부동소수점 계산을 위한 TI사의 TMS320C6713 DSP, 하위 통신을 위한 Infineon 81C91 8비트 CAN 제어기와 트랜시버로 구성 되어진다.

3.2 지역제어 부

휴머노이드 로봇의 하위 모듈은 몸체 2개, support leg과 swing leg 각각 6개씩 총 14개의 모듈로 이루어져있으며 각각의 모듈마다 모터 컨트롤을 위해 만들어진 TI사의 TMS-320F2812 DSP칩을 중심으로 구성되어 있다. 또한 이 칩에는 CAN 통신을 위한 eCAN이 포함되어져 있고 32개의 메일 박스를 포함하고 있다. 이런 구성의 모듈이 동일하게 14개로 하위 로봇의 구조를 형성하고 있으며 이들을 지역제어기라고 정의 한다.

표 1 주요 메시지 묘사

Table 1 Description of the main communication message

priority	Message Description	Size [byte]	Length [# of msg]	T[ms]	D[ms]	J[ms]	From	To
1 ~ 14	Motor 1~14 data	8	1	40	20	1	Main	Module 1 ~ 14
15~28	Motor RTR data	8	1	40	20	1	Main	Module 1 ~ 14
29~42	Motor 1~14 Sensor data	8	1	40	20	1	Main	Module 1 ~ 14

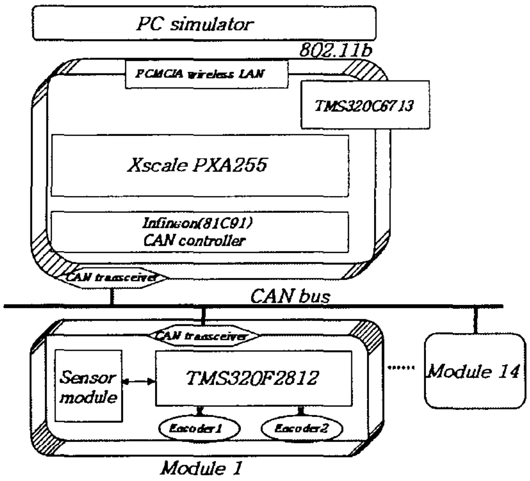


그림 2 휴머노이드 로봇의 네트워크 시스템 구조
Fig. 2 Network system architecture of humanoid robot

Fig. 2에서와 같이 주제어기와 지역제어 기들은 서로 CAN 버스를 통하여 연결되어 있으며 주제어 기에서 지역제어기로에 연결부 사이의 좌표 정보들을 보내주고 출력 측 엔코더 정보와 센서정보들을 받는 형태의 시스템으로 구성되어진다.

주제어 기에서 각각의 하위 스테이션으로 보내어지는 엔코더정보는 10ms 주기와 2바이트 크기로 만들어지며 4개의 정보씩 8바이트의 크기로 하나의 메시지 안에 넣어 전송한다.

Table 1에서 보는바와 같이 1~14번 노드까지 차례로 우선순위를 부여 하였다. 주제어기 안에 있는 Infineon 81C91 8비트 CAN 제어기는 16개의 메일 박스를 가지고 있으므로 우선 메시지를 송신한 다음 수신 메일 박스로 변화하면서 출력 엔코더 값을 요청하는 리모트 프레임을 14개의 노드로 하나씩 보내며 이 리모트 프레임을 받은 노드는 RTR 데이터를 보내게 된다. 이를 15~28의 우선순위를 노드의 번호순으로 부여하였다. 그 다음 각각의 노드의 센서정보들을 받는 것으로 스케줄링 하였으며 29~42의 우선순위를 정하였다. Table 1은 보행을 위한 40ms의 정보를 나타낸 것이며 20ms의 데드라인을 가지고 우선순위대로 전송하는 고정된 우선순위방식으로 전송한다. PC 시뮬레이터상의 휴머노이드 로봇의 한 보 보행 시간은 3초이므로 이 40ms의 정보 주기가 75번 전송되었을 경우 3초를 넘지 말아야 실시간성이 보장 된다고 할 수 있다. 이 시스템을 시뮬레이션을 통하여 구현하고

표 2 CAN 메시지의 응답시간
Table 2 Response time of messages in the CAN (Unit: ms)

	500kbps			1Mbps		
	Ave	Min	Max	Ave	Min	Max
Motor data 1~14	2.08	0.28	3.86	1.05	0.14	1.96
Motor RTR 15~28	6.76	4.25	9.26	3.42	2.15	4.70
Sensor data 29~42	11.3249	9.5306	13.1191	5.7365	4.8297	6.6433

500Kbps, 1Mbps의 전송속도로 전송하여 실시간 성을 분석한다. Table 2는 Table 1에 맞추어 각각의 전송 시간에 따른 최악의 응답속도를 구한 결과이다. 500Kbps이상의 전송속도로 전송했을 경우 데드라인을 넘지 않았다. 또한 500 Kbps의 전송속도에서 응답시간은 13.1191ms이며 75번이 반복되었을 경우, 로봇이 한 보 보행하는데 필요한 모든 데이터를 3초 내인 984.94ms의 시간 안에 전송 할 수 있었다.

4 기존의 CAN 우선순위 운영방법

4.1 Deadline Monotonic Scheduling

DM(Deadline Monotonic)은 각 메시지들의 데드라인을 비교하여 이 데드라인이 짧을수록 높은 우선순위를 인가받는 방식이다[9]. 여기서 데드라인은 한 메시지에 해당하는 데이터가 발생하거나 혹은 갱신된 시점부터 이 데이터가 전송을 마칠 때까지의 시간제한으로 주기적인 데이터의 경우 일반적으로 주기에서 큐잉 지연시간을 제외한 나머지가 된다. DM은 주기가 짧을수록 높은 우선순위를 갖도록 한 RM(Rate Monotonic)의 방법을 개선하여 주기적인 데이터뿐만 아니라 비주기적인 데이터들도 처리할 수 있도록 하였다. 이 DM은 고정 우선순위 인가방법으로 쉽게 실제 시스템에 적용할 수 있는 장점을 가지고 있다. 그러나 단순히 데드라인의 길고 짧음만으로 실제 시스템에서 발생하는 데이터들의 중요도를 반영하기는 어렵다. 또한 동적 우선순위 인가방법에 비하여 배정능력이 떨어지는 단점을 드러낸다. 여기서 배정능력이 좋다는 것은 각 데이터의 데드라인 내에 전송을 보장 할 수 있는 메시지의 개수가 많음을 뜻하는데 DM에서 메시지 개수는 이론상 2048개의 메시지를 모두 사용 할 수 있는 것으로 되어 있으나 실제 시스템에서 메시지의 시간조건을 맞추어 줄 수 있는 메시지의 개수는 이보다 훨씬 적게 된다[10].

4.2 Earliest Deadline Scheduling

ED(Earliest Deadline)는 데드라인까지 남은 시간인 슬랙시간이 경과함에 따라 줄어들어 이것이 작을수록 우선순위가 높은 메시지가 되는 방식이다[11]. ED는 DM과 달리 동적 우선순위 인가 방법이므로 배정능력이 매우 높다. 그러나 ED 역시 메시지의 우선순위와 실제 데이터의 중요도가 일치하지 않으며 실제 시스템에 적용하려 할 경우 크게 두 가지 문제점이 발생한다. 첫째로는 실시간 시스템에서 발생하는 데드라인은 작게는 수ms에서 수초에 이르기까지 광범위하게 정해지기 때문에 이를 모두 만족시키려면 슬랙시간의 비트수가 늘어나야한다. 이것은 사용 가능한 메시지의 감소를 뜻하는데 실례로써 표준형태의 CAN에서 슬랙시간에 5비트를 사용하면 메시지를 32개밖에 쓸 수가 없게 된다. 둘째는 슬랙시간을 갱신하기 위하여 CPU는 매시간 ID를 다시 써야하므로 과다한 부하를 갖게 된다. 또한 ID를 갱신하는 동안 그 메시지는 사용할 수 없는 상태가 된다. 뿐만 아니라 전체시스템의 각 local clock들이 전부 동기화 되어야만 하므로 이를 위한 부가적인 다른 수단을 마련해야 하는 문제점을 지닌다.

4.3 Mixed Traffic Scheduling

DM과 ED의 장단점을 상호 보상하여 제안된 MTS(Mixed Traffic Scheduling)는 고속메시지의 경우에는 ED를 저속의 메시지의 경우에는 DM을 이용하는 방법이다. 비실시간 데이터에 대해서는 별도의 고정 우선순위를 인가한다.

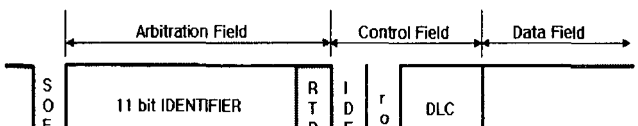
MTS는 실시간 제어 시스템에서 발생하는 데이터의 실제 중요도를 곧바로 식별자의 우선순위에 반영하도록 되어 있으며 사용 가능한 메시지의 개수도 고속 메시지 32개, 저속 메시지 512개, 비실시간 메시지 512개로 충분하며 이론상 배정 능력도 ED에 필적한다. 그러나 고속의 메시지에 여전히 동적 우선순위 인가방법을 사용하므로 CPU의 과다한 부하와 모든 local clock을 동기화해야 되는 문제, 메시지가 식별자 갱신중 사용불능의 메시지가 되는 문제가 그대로 남아있다[9],[11].

5. 휴머노이드 로봇을 위한 CAN 네트워크의 응용층 설계

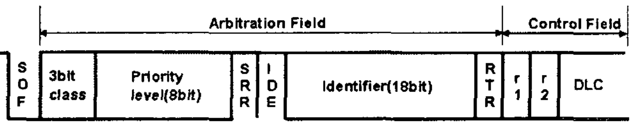
5.1 CAN 기반 휴머노이드 로봇의 데이터 특징

휴머노이드 로봇의 네트워크를 구성함에 있어 고려해야 하는 점으로 여러 가지가 있겠지만 가장 중요한 것은 로봇의 안전성이 최우선적으로 고려되어야 하는 실시간 제어 시스템이다. 다시 말해서 평상시에 정상적인 조건에서의 동작은 물론이고 이상 사태가 발생하였을 시에도 네트워크가 제 기능을 발휘한다는 보장이 있어야 한다는 점이다. 휴머노이드 로봇에 각 관절의 모터들의 이상, 긴급 상황의 발생 등의 경우에는 이러한 이벤트를 알리는 메시지를 발생시키게 된다. 이러한 이벤트들이 언제나 데드라인 내에 전송될 수 있을 때, 전체 시스템의 진단, 모니터링 시스템이 제기능을 발휘하게 된다. 즉 각각의 스테이션 상에서 발생하는 메시지의 우선순위에 따라 대역폭을 할당하는 것이 아니라 공평하게 대역폭을 나누어야 한다. 기존의 방법들 중 MTS같은 방법에서는 이러한 메시지들을 처리하도록 제안되어 있으나 이것이 동적 우선순위 인가방법으로 되어 있어 사실상 적용하기가 어렵다.

5.2 데이터 교환원칙



(a) 표준 형태
(a) Standard format



(b) 제안된 형태
(b) ID update format

그림 4 CAN 데이터 프레임
Fig. 4 CAN data frame

CAN에 의해 데이터가 교환될 때 어떠한 스테이션도 주소화되지 않고 메시지의 내용이 통신망에서 유일하게 존재하는 메시지 ID에 의해서 정해진다. 실시간 처리에 있어서 통신망에서 교환되는 메시지의 우선순위는 메시지의 내용에 따라 다르며 해당 메시지의 ID에 의해 결정 되는데 시스템을 디자인하는 동안 결정되고 이것은 여러 개의 스테이션이 동시에 버스를 엑세스하려고 할 때 버스 할당을 위하여 중요하다. 이진법에 의해 표현되고 동적으로 변할 수 없으며 가장 낮은 이진수를 가진 ID가 가장 큰 우선순위를 갖는다[12].

5.3 제안된 ID 인가 알고리즘

제안된 ID 인가 알고리즘은 스테이션이 프레임을 전송하기 위해 처음 시도한 순간부터 경과된 시간에 대하여 비례적으로 점점 증가되어가는 각 스테이션의 우선순위를 표시해주는 것이 기본적인 생각이다. 우선 시스템이 동작하면 우선순위 레벨의 비트는 전부 1로 놓아진다. 프레임의 우선순위 레벨은 각 스테이션의 지역 변수 안에 기록되어지며 동작 중 전송 실패 시(프레임간의 충돌이나 전송 에러에 발생한 경우)에 1의 값이 감소되어지는 형태로 업데이트 되어진다. 다음 전송의 경우에는 우선순위 레벨이 낮아지므로 더 낮은 ID를 가진 메시지와 arbitration이 일어날 경우 우선순위 레벨이 낮은 메시지의 전송이 이루어진다. 만약 두 개 또는 더 많은 노드들이 같은 우선순위 레벨을 가지고 전송을 동시에 시도한다면 그 충돌은 ID 필드(18비트)를 고려하면 해결되어진다. 이 경우에는 각 스테이션의 우선순위가 완전하게 공정한 전송을 하지는 않는다. 왜냐하면 더 낮은 ID를 가진 프레임을 전송하기 때문이다. 또한 전송이 성공한다면 스테이션의 지역 변수는 다시 전부 1의 값으로(8비트이므로 255) 놓아 진다. Fig. 4의 (b)는 우선순위 레벨이 있는 프레임의 구조이다. (a)의 표준 프레임과 비교해 보았을 때 오버 헤드 부분이 늘어나는 단점이 있지만 고정 되어있는 ID를 사용했을 때 발생하는 낮은 우선순위를 가진 메시지의 전송 시간의 불공정성을 해결할 수 있다.

휴머노이드 로봇의 경우 특별히 중요하고 중요하지 않은 모터가 없으므로 메시지의 공정한 전송은 중요한 문제로 제기 된다. 로봇의 주기적인 동작 메시지는 간헐적으로 발생하는 메시지보다 더 우선되어야 한다. 이 부분은 3비트의 우선순위 클래스를 통해 해결하며 메시지의 중요도별로 묶어서 클래스로 만든다.

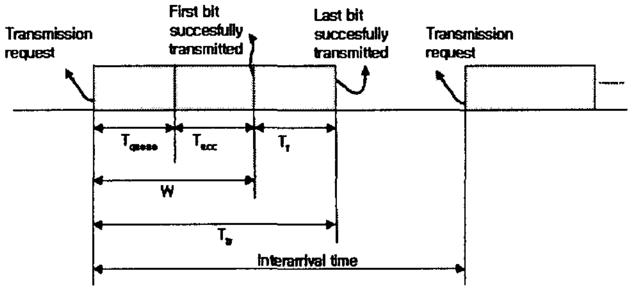


그림 5 비실시간 전송 순서
Fig. 5 Transmission sequence of non-RT

위의 Fig. 5는 비실시간 트래픽의 전송 순서를 나타낸 그림이다. FIFO 큐 안에서 프레임의 대기에 의해 발생하는 시간(T_{queue})과 공유된 버스의 제어에 의해 얻어지는 시간(T_{acc}), 그리고 하나의 메시지 프레임이 전송되어 첫 비트의 전송이 성공한 시간부터 마지막 비트의 전송이 끝나는 시간(T_f)으로 구성 되어진다.

또한 W는 전송 지연시간으로 T_{queue}와 T_{acc}의 합으로 나타내어진다. 시스템의 대역폭은 T_f의 역수로 나타내지며 네트워크안에서 처리할 수 있는 최대의 처리량을 나타낸다.

6 모의실험

먼저 네트워크의 비실시간 트래픽에 관한 MAC 메카니즘을 분석한다. 실험은 이산신호 분석을 위한 ARENA를 사용하며 각 노드의 메시지의 전송과정은 POISSON분포에 따라 전송 한다.

표 2 네트워크 부하와 스테이션 수를 고려한 평균 전송 지연 시간

Table 2 Average transmission delays varying network load and number of station

(a) 전송속도 1Mbps인 경우 [단위:ms]

(a) The case of transmission speed with 1Mbps [Unit:ms]

Load		5 nodes			25 nodes			100 nodes		
		ID 1	ID 3	ID 5	ID 1	ID13	ID 25	ID 1	ID 50	ID 100
10%	CAN	0.135	0.1396653	0.1584597	0.1387855	0.15933949	0.1609389	0.130628	0.1530213	0.14070498
	IU	0.1355598	0.1423045	0.1619787	0.1479028	0.15843303	0.1642447	0.1322541	0.1372127	0.15243483
50%	CAN	0.739141	1.1806368	2.1844994	0.6641232	1.16797383	2.5508174	0.6784923	1.1966588	2.66595675
	IU	1.0391321	1.3318454	1.6075769	1.0050889	1.37111374	1.6344224	0.9928525	1.3408027	1.63394254
90%	CAN	1.4654858	4.0378703	42.384775	1.2535218	4.01558353	92.644816	1.201484	4.1776688	105.15311
	IU	10.537758	11.931985	13.292621	11.095006	12.6244461	13.670749	10.901143	12.107106	12.9733294

(b) 전송속도 500Kbps인 경우 [단위:ms]

(b) The case of transmission speed with 500Kbps [Unit:ms]

Load		5 nodes			25 nodes			100 nodes		
		ID 1	ID 3	ID 5	ID 1	ID13	ID 25	ID 1	ID 50	ID 100
10%	CAN	0.276	0.2855379	0.3239621	0.2837393	0.32576073	0.3290306	0.2670616	0.3128436	0.28766351
	IU	0.2771445	0.2909336	0.3311564	0.3023791	0.32390753	0.3357891	0.2703862	0.2805237	0.31164454
50%	CAN	1.5111327	2.4137464	4.4660877	1.357763	2.38785761	5.2150045	1.3871398	2.4465024	5.45040047
	IU	2.1244479	2.7228839	3.2866017	2.0548483	2.80316588	3.3414858	2.0298317	2.7411967	3.34050474
90%	CAN	2.9961043	8.2552014	86.653318	2.5627558	8.20963744	189.40718	2.4563673	8.5410118	214.979692
	IU	21.54386	24.39428	27.176026	22.683123	25.8099788	27.949088	22.286782	24.752306	26.5232512

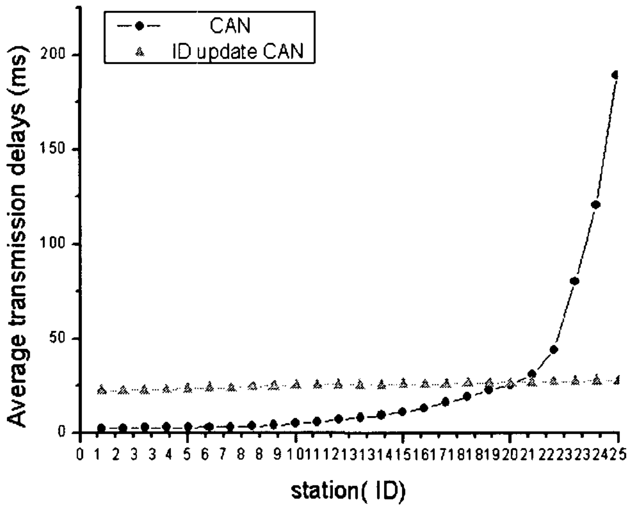


그림 6 비실시간 교환에 대한 평균 전송시간

Fig. 6 Average transmission delays for non-RT exchange

Table 2는 대역폭의 10%, 50%, 90%에 해당하는 부하를 생성하여 500Kbps와 1Mbps의 전송속도로 전송하였을 경우에 기본 CAN과 제안된 ID 인가 CAN의 각 ID에 대한 전송속도를 나타낸 결과이다. 결과에서 볼 수 있듯이 부하가 커질수록 또 스테이션의 수가 더 많아질수록 각 스테이션에 걸리는 전송 지연 시간은 점점 증가함을 볼 수 있다. 각각 다른 ID를 가지는 스테이션들의 전송지연은 부하가 커지면 불공평하게 걸린다.

Fig 6은 부하 90%, 스테이션 25개의 모델로 비실시간 트래픽에 대한 기본 CAN과 제안된 ID 인가 CAN의 평균 전송 지연 시간에 대한 그래프이다. 그림에서 볼 수 있듯이 기본 CAN의 경우 전송지연의 각 ID마다 불공평하게 작용하는 반면에 제안된 ID 인가 CAN은 전송지연 시간이 부하의 양만큼 공평하게 발생한다는 것을 볼 수 있다.

Table 2에서 보듯이 부하가 적은 경우라면 제안된 ID 인가 방법의 필요성이 없다. 하지만 네트워크상에 부하가 커질수록 공정한 메시지 전송을 위해서는 제안된 ID 인가 방법은 꼭 필요하다고 하겠다.

7. 결 론

본 논문에서는 실시간 제어 시스템에 적용할 경우 많은 이점을 가진 것으로 알려진 CAN을 휴머노이드 로봇의 네트워크에 적용하기 위한 구조를 설계하였고 또한 응용계층을 제안하고 성능을 분석하였다. CAN은 가장 중요한 특징인 식별자를 인가하는 방법으로 기존에 제안된 방법들의 특징을 보였으며 본 논문의 요지인 휴머노이드 로봇을 위한 CAN의 식별자 인가 방법을 제안 하였다. 이것은 시간이 변함에 따라 계속해서 식별자를 새로이 인가하는 방법과 달리 고정 우선순위 인가방법으로 전송 실패 시 우선순위를 높이는 방법이다. 또한 가능한 대역폭을 각각의 스테이션마다 균등하게 할당함으로서 전송지연시간을 일정하게 만들어주며 휴머노이

드 로봇의 네트워크와 같이 안전성을 가장 중시하는 데이터 특성을 잘 반영할 수 있도록 하였다. 앞으로 휴머노이드 로봇의 다양한 분석을 통해 더욱 안전성을 꽤하며 메시지 전송의 지연시간을 더 감소시킬 수 있는 방안이 연구 되어야 할 것이며 본 논문에서 제안된 알고리즘은 휴머노이드 로봇과 비슷한 데이터 특성을 가진 시스템에도 유용하게 적용 되어질 것이라고 생각된다.

감사의 글

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

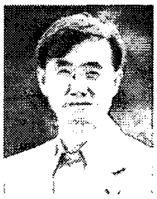
참 고 문 헌

[1] Miwa, H., Itoh, K., Ito, D., Takanobu, H., Takanishi, A., "Introduction of the need model for humanoid robots to generate active behavior," Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on, Volume: 2, Pages: 1400-1406 Vol. 2, 27-31 Oct. 2003.
[2] K. Hirai, "The Honda Humanoid Robot: Development and Future Perspective," Industrial Robot: An International Journal, Vol. 26, No. 4, 1999, pp. 260-266.
[3] Y. Sakagami and R. Watanabe, "The intelligent ASIMO: System overview and integration," in Proceedings of the IEEE Intl. Confer-ence, 2002 Intelligent Robots and Systems.
[4] Khawar M. Zuberi, "Design and Implementation of Efficient Message Scheduling for Controller Area Network," IEEE TRANSACTIONS ON COMPUTERS, Vol. 49, No. 2, FEBRUARY 2000
[5] Tindell, K., Burns, A., "Guaranteed Message Latencies for Distributed Safety Critical Hard Real-Time Networks," YCS 229, Dept. Computer Science, Univ. of York(1994).
[6] M.A. Livani, J. Kaiser, W.J. Jia: "Scheduling Hard and Soft Real-Time Communication in the Controller Area Network (CAN)," 23rd IFAC/IFIP Workshop on Real Time Programming, Shantou, China, June 1998.
[7] Miwa, H., Itoh, K., Ito, D., Takanobu, H., Takanishi, A., "Introduction of the need model for humanoid robots to generate active behavior," Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on, Volume: 2, Pages: 1400-1406 Vol. 2, 27-31 Oct. 2003.
[8] K. Tindell, A. Burns, and a Wellings, "Calculating Controller Area Network(CAN) Message Response Times," IFAC workshop on Distributed Computer Control Systems(DCCS), Toledo, Spain, September, 1994.
[9] Ben Kao, Hector Garcia-Molina, "Deadline Assignment in a Distributed Soft Real-Time System," IEEE Transactions on Parallel and Distributed Systems, Vol. 8, No. 12, December 1997.
[10] Khawar M. Zuberi, Kang G. Shin, "Scheduling Messages on Controller Area Network for Real-Time CIM Applications," IEEE Transactions on Robots and Automation, Vol. 13, No. 2, April 1997.
[11] Di Natale, M., "Scheduling the CAN bus with earliest deadline techniques," Real-Time Systems Symposium, 2000. Proceedings. The 21st IEEE, Pages:259-268, 27-30 Nov. 2000.
[12] BOSCH "CAN Specification version 2.0," Page:42-44, Sep, 1991.
[13] D, Kelton, R. Sadowski & D. Sadowki "Simulation with Arena, 2/e," ISBN 0-07-239270-3, McGraw-Hill.

저 자 소 개

구 자 봉(具 滋 俸)
1978년 경북 의성 태생
현재 인하대학교 전기공학과 석사과정

허 욱 열(許 旭 烈)
1978년 서울대학교 전기공학(공학석사), 1982년 서울대학교 전기공학과(공학박사), 1985~1986 미국 Univ. of illinoi, Visiting Professor, 1997.04~1997.07 독일 RWTH Aachen Univ., Visiting researcher, 현재 인하대학교 전기공학과 교수



김 진 걸(金 振 傑)
1983년 미국 Univ. of Iowa 전기공학(공학석사), 1983~1987년 미국 Univ. of Iowa, Teaching Assistant, 1988년 미국 Univ. of Iowa 전기공학(공학박사), 1996~1997년 미국 Univ. of California Irvine, Visiting Professor, 현재 인하대학교 전기공학과 교수

김 병 륜(金 丙 倫)
1967년 서울 태생, 1993년 인하대학교 전기공학과(공학석사), 2002년 Univ. of Pittsburgh, Department of information Science for telecommunication(통신석사). 현재 인하대학교 전기공학과 박사과정