

휴머노이드 로봇에 대한 CAN적용

구자봉, 허욱열, 김진걸
 인하대학교 전기공학과

An Controller Area Network(CAN) Application for Humanoid Robot

Jabong Ku · Ukyoul Huh · Jingeol Kim
 Dept. of Electrical Engineering INHA University

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 algorithm, this paper also presents some performance figures obtained using a specially developed software simulator, while the behavior of the new algorithm is compared with the traditional CAN systems, in order to see how effective they are.

1. 서 론

현재 인간의 신체 메커니즘인 직립보행과 동작을 모방한 휴머노이드 로봇에 관한 연구개발이 급속도로 진전되고 있다.

이런 휴머노이드 로봇이 실제 인간과 같은 행동을 하기 위하여 제어 구조를 사용자 인터페이스, 행동 계획기, 비전 프로세싱, 양다리, 팔과 같이 모듈별로 분리하는 분산형 제어 구조를 가지며 안정화와 좌표틀을 제어하는 정보들을 네트워크를 통하여 보내어 주고 있다[1]. 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 로봇의 하위 통신을 CAN 프로토콜을 사용하여 설계하며 메시지 스케줄링과 실시간 분석을 하였다. 특히, CAN 프로토콜은 ID(Identifier)가 고정되어 있기 때문에 낮은 우선순위를 가진 메시지의 전송 문제가 큰 문제점이다. 이 점을 해결하기 위하여 MTS(Mixed Traffic Scheduler) 알고리즘과 같은 동적 스케줄링의 방법이 연구되었다[4].

본 논문의 구성은 먼저 2장에서 휴머노이드 로봇의 네트워크 시스템을 설계하며 3장에서는 ID 업데이트 알고리즘을 제안하고 기본 CAN과의 비교 분석을 모의실험을 통하여 실행한다. 4장에서는 결과와 앞으로 연구 되어야 할 방향을 조명한다.

2. 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.

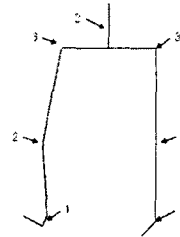


Fig.1 The location of the joints in the lower part of humanoid robot, indicating the degrees of freedom in each joint.

여러 개의 자유도로 구성되어있는 곳(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.

2.1 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 하이버스의 리눅스 보드를 중심으로 부동소수점 계산을 위한 TI사의 TMS320C6713 DSP, 하위 통신을 위한 Infineon 81C91 8비트 CAN 제어기와 트랜시버로 구성 되어진다.

2.2 지역제어 부

휴머노이드 로봇의 하위 모듈은 몸체 2개, support leg 과 swing leg 각각 6개씩 총 14개의 모듈로 이루어져있으며 각각의 모듈마다 모터 컨트롤을 위해 만들어진 TI사의 TMS320F2812 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.

Fig. 2에서와 같이 주제어기와 지역제어 기들은 서로 CAN 버스를 통하여 연결되어 있으며 주제어기에서 지역제어기로 연결부 사이의 좌표 정보들을 보내주고 출력측 엔코더 정보와 센서 정보들을 받는 형태의 시스템으로 구성되어진다.

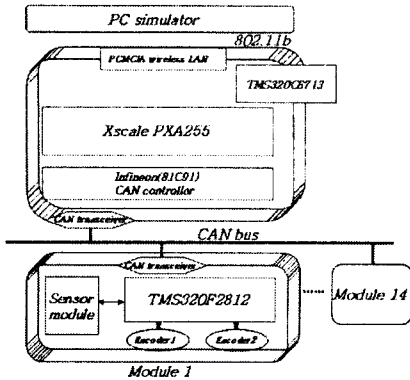


Fig.2 Network system architecture of humanoid robot

priority	Message Description	Size [byte]	Length [#ofmsg]	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 RTRdata	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

Table 1. description of the main communication message

주제어기에서 각각의 하위 스테이션으로 보내어지는 엔코더 정보는 10ms 주기와 2바이트 크기로 만들어지며 4개의 정보씩 8바이트의 크기로 하나의 메시지 안에 넣어 전송한다.

Table 1에서 보는 바와 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 프레임용 14개의 노드로 하나씩 보내며 이 리모트 프레임은 받은 노드는 RTR 데이터를 보내게 된다. 이를 15~28의 우선순위를 노드의 번호순으로 부여하였다. 그 다음 각각의 노드의 센서 정보들을 받는 것으로 스케줄링 하였으며 29~42의 우선순위를 정하였다. Table 1은 보행을 위한 40ms의 정보를 나타낸 것이며 20ms의 데드라인을 가지고 우선순위대로 전송하는 고정된 우선순위 방식으로 전송한다. PC 시뮬레이터상의 휴머노이드 로봇의 한 보 보행 시간은 3초이므로 이 40ms의 정보 주기가 75번 전송 되었을 경우 3초를 넘지 말아야 실시간성이 보장 된다고 할 수 있다. 이 시스템을 시뮬레이션을 통하여 구현하고 500Kbps, 1Mbps의 전송속도로 전송하여 실시간 성을 분석한다.

	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

Table 2. Response time of messages in the CAN (Unit: ms)

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

3. CAN의 우선순위

3.1 데이터 교환원칙

CAN에 의해 데이터가 교환될 때 어떠한 스테이션도 추소화되지 않고 메시지의 내용이 통신망에서 유일하게 존재하는 메시지 ID(identifier)에 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 액세스하려고 할 때 버스 할당을 위하여 중요하다. 이진법에 의해 표현되고 동적으로 변할 수 없으며 가장 낮은 이진수를 가진 ID가 가장 큰 우선순위를 갖는다[5].

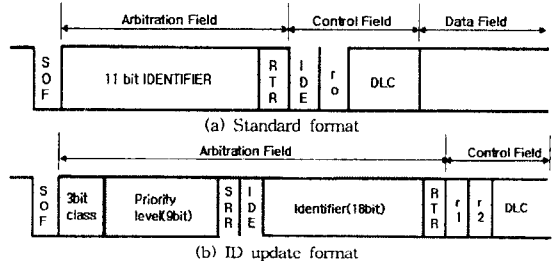


Fig. 4. CAN data frame

3.2 ID 업데이트 알고리즘

ID 업데이트 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) 놓아진다. Fig 4.의 (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 제기된다. 로봇의 주기적인 동작 메시지는 간헐적으로 발생하는 메시지보다 더 우선되어야 한다. 이 부분은 3비트의 우선순위 클래스를 통해 해결하며 메시지의 중요도별로 묶어서 클래스로 만든다.

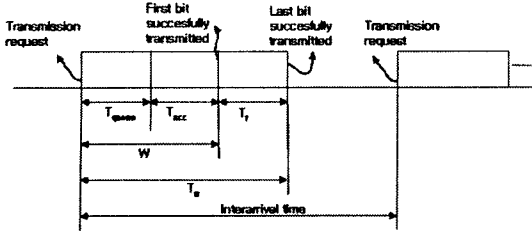


Fig 5. Transmission sequence of non-RT

위 Fig 5는 비실시간 트래픽의 전송 순서를 나타낸 그림이다. FIFO 큐 안에서 프레임의 대기기에 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 W 는 전송 지연시간으로 T_{queue} 와 T_{acc} 의 합으로 나타내어진다. 시스템의 대역폭은 T_f 의 역수로 나타내지며 네트워크 안에서 처리할 수 있는 최대의 처리량을 나타낸다.

3.3 모의실험

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

Table 2는 대역폭의 10%, 50%, 90%에 해당하는 부하를 생성하여 500Kbps와 1Mbps의 전송속도로 전송하였을 경우에 기본 CAN과 ID 업데이트된 CAN의 각 ID에 대한 전송 지연시간을 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.

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	0.1396	0.1584	0.1387	0.1593	0.1609	0.1306	0.1530	0.1407
	IU	0.1355	0.1423	0.1619	0.1479	0.1584	0.1642	0.1322	0.1372	0.1524
50%	CAN	0.7391	1.1806	2.1844	0.6641	1.1679	2.5508	0.6784	1.1966	2.6659
	IU	0.7391	1.1806	2.1844	0.6641	1.1679	2.5508	0.6784	1.1966	2.6659
90%	CAN	1.4654	4.0378	42.384	1.2535	4.0155	92.644	1.2014	4.1776	105.15
	IU	1.4654	4.0378	42.384	1.2535	4.0155	92.644	1.2014	4.1776	105.15

(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	0.276	0.2855	0.3239	0.2837	0.3290	0.2670	0.3128	0.2876
	IU	0.2771	0.2909	0.3311	0.3023	0.3239	0.3357	0.2703	0.3116
50%	CAN	1.5111	2.4137	4.4660	1.3577	2.3878	5.2150	1.3871	2.4465
	IU	1.5111	2.4137	4.4660	1.3577	2.3878	5.2150	1.3871	2.4465
90%	CAN	2.9961	8.2552	86.653	2.5627	8.2096	189.40	2.4563	8.5410
	IU	2.9961	8.2552	86.653	2.5627	8.2096	189.40	2.4563	8.5410

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

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

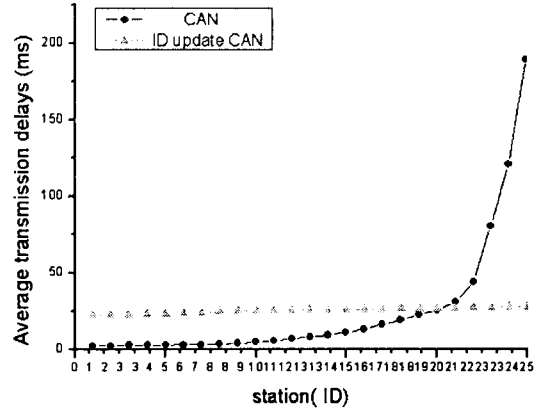


Fig 6. Average transmission delays for non RT exchange

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

Table 2에서 보듯이 부하가 적은 경우라면 ID 업데이트의 필요성이 없다. 하지만 네트워크상에 부하가 커질수록 공정한 메시지 전송을 위해서는 ID 업데이트 알고리즘은 꼭 필요하다고 하겠다.

4. 결론

본 논문에서는 휴머노이드 로봇의 네트워크의 구조를 제한하였으며 네트워크의 로드가 많아질수록 생기는 낮은 우선순위를 가진 메시지 전송문제를 해결하기 위한 방안을 제시하였다. 앞으로 실시간 트래픽에 관한 분석이 필요할 것이며 또 실제 로봇의 네트워크 구현과 쉽게 적용할 수 있는 스케줄링의 방법의 연구가 있어야 하겠다.

감사의 글

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

[참고 문헌]

- [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] BOSCH "CAN Specification version 2.0" page42-44,sep,1991.