휴머노이드 로봇에 대한 CAN(Controller Area Network) 적용

Application of Controller Area Network to Humanoid Robot

구 자 봉*, 허 욱 열**, 김 진 걸***
(Ja-bong Ku · Uk-youl Huh · Jin-geol Kim)

Abstract - Because robot hardware architecture generally is consisted of a few sensors and motors connected to the central processing unit, this type of structure is led to time consuming and unreliable system. For analysis, one of the fundamental difficulties in real-time system is how to be bounded the time behavior of the system. When a distributed control network controls the robot, with a central computing hub that sets the goals for the robot, processes the sensor information and provides coordination targets for the joints. If the distributed system supposed to be connected to a control network, the joints have their own control processors that act in groups to maintain global stability, while also operating individually to provide local motor control. We try to analyze the architecture of network-based humanoid robot's leg part and deal with its application using the CAN(Controller Area Network) protocol.

Key Words : real-time; humanoid robot; CAN

1. 서 론

최근 인간의 신체 메커니즘인 직립보행과 동작을 모방한 휴머노이드 로봇에 관한 연구개발이 급속도로 진전되고 있다. 이족 직립에 의한 보행식 이동은 crawler[1] 방식으로 다른 이동식 로봇에 비하여 불안정한 자세이나 고르지 못한 지면이나 장애물 등 작업경로 상에 요철이 있는 보행면, 계단, 사다리, 승강과 같은 불연속적인 보행 면에 대응할 수 있는 등 유연한 이동 작업을 실현할 수 있다는 점에서 그 우수성이 인정되고 있는 경향이다.

이런 휴머노이드 로봇이 실제 인간과 같은 행동을 하기 위하여 제어 구조를 user interface, motion planner, vision processing, 양다리와 팔과 같이 모듈별로 분리하는 분산형 제어 구조를 가지며 안정화와 좌표를 제어하는 정보들을 네트워크를 통하여 보내어 주고 있다[2]. 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.

본 논문에서는 휴머노이드 로봇의 하위 통신을 CAN 프로토콜을 사용하여 설계하며 메시지 스케줄링과 실시간 분석을 통하여 시스템의 안정성과 신뢰도를 높일 수 있는 방안을 제시하였다. 휴머노이드 로봇의 보행 메시지에 관한 스케줄링을 하고 CAN 프로토콜을 적용한 시스템 노드에서 발생되는 메시지의 수학적 모델링을 통해 최악의 응답시간을 찾음으로 시스템의 실시간성 보장을 분석하였다.

본 논문의 구성은 먼저 2장에서 통신 지연 시간의 수학적 모델링을 보이고 3장에서는 휴머노이드 로봇의 네트워크 시스템을 설계하며 4장에서는 수학적 모델링을 기반으로 시뮬레이션 함으로써 최악의 응답시간을 계산하고 분석 결과를 보인다. 5장에서는 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[3]에 대한 고정된 우선순위[4] 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[2, 6]. 스트림에서 보내지는 CAN 메시지의 최악의 경우 응답 지연시간 R_m은 다음과 같이 정의되어진다.

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

여기서 J_m은 프레임의 큐잉 지터이며 w_m은 큐잉 지연시간

저자 소개
* 具 滋 俸 : 仁荷大學 電氣工學科 碩士課程
** 許 旭 烈 : 仁荷大學 電氣工學科 敎授 · 工博
***金 振 傑 : 仁荷大學 電氣工學科 敎授 · 工博

으로 주어지고 다음과 같다.

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

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 전송 속도일 경우 1μs로 표현 된다. B_m은 메시지 m보다 더 낮은 우선순위에 의해 지연 되어질 수 있는 최대 시간으로 정의되며 다음과 같다.

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

여기서 lp(m)은 낮은 우선순위 메시지들의 집합이다.

CAN은 메시지당 47비트의 오버헤드를 가지고 있고 5비트의 넓이로 스터핑비트(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.

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

여기서 s_m은 메시지의 데이터 크기 바이트 단위이다.

메시지의 스케줄링은 고정된 우선순위 선점방식을 사용하며 프로세서 스케줄링 방식과 메시지 스케줄링 방식은 DMS(deadline monotonic scheduling)를 사용한다[5].

3. 휴머노이드 로봇에 대한 네트워크 디자인

인간의 관절 기능에서 움직일 수 있는 범위는 로봇 디자인에서 매우 중요한 부분이다[2]. Fig.1은 로봇의 하체 부분의 각각의 관절 부분에 구성되어있는 자유도를 나타낸 그림이다.

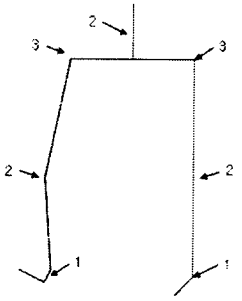


Fig.1 The location of the joints in the lower part of humanoid robot, indicating the degrees of freedom in each joint.

여러 개의 자유도로 구성되어있는 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 Main Controller

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

3.2 Local Controller

휴머노이드 로봇의 하위 모듈은 몸체 2개, support leg과 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 Local Controller라고 정의 한다.

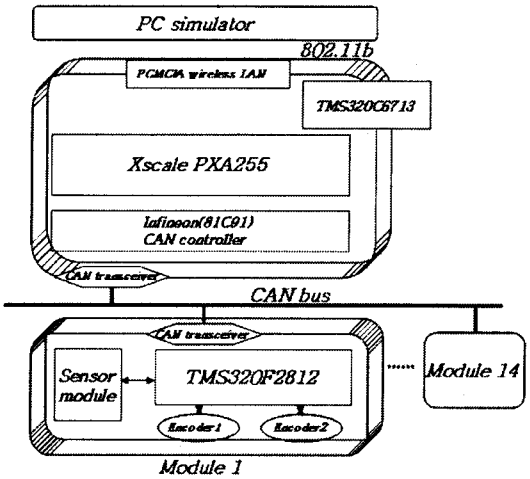


Fig.2 Network system architecture of humanoid robot

4. 제안된 디자인의 모의실험

Fig. 2에서와 같이 main controller와 local controller들은 서로 CAN 버스를 통하여 연결되어 있으며 main controller에서 Local Controller로 연결부 사이의 좌표 정보들을 보내주고 출력 측 엔코더 정보와 센서정보들을 받는 형태의 시스템으로 구성되어진다.

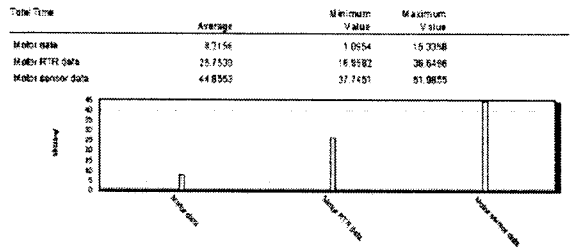
priority	Message Description	Size [byte]	Length [# of msg]	T[ms]	D[ms]	J[ms]	From	To
1 ~ 14	Motor 1~14 data	8	1	40	20	1	Main	Module 1 ~ 14
15~28	Motor RTR data	8	1	40	20	1	Main	Module 1 ~ 14
29~42	Motor 1~14 Sensor data	8	1	40	20	1	Main	Module 1 ~ 14

Table 1. description of the main communication message

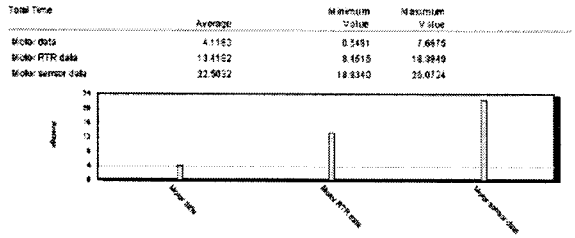
메인 컨트롤러에서 각각의 하위 노드로 보내어지는 엔코더 정보는 10ms 주기와 2바이트 크기로 만들어지며 4개의 정보

씩 8바이트의 크기로 하나의 메시지 안에 넣어 전송한다.

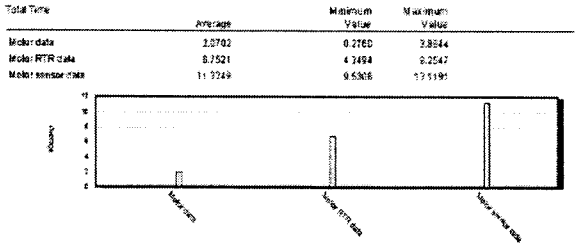
Table 1에서 보는바와 같이 1 ~ 14번 노드까지 차례로 우선순위를 부여 하였다. 메인 컨트롤러 안에 있는 Infineon 81C91 8비트 CAN 컨트롤러는 16개의 메일 박스를 가지고 있으므로 우선 메시지를 송신한 다음 수신 메일 박스로 변화하면서 출력 엔코더 값을 요청하는 remote frame을 14개의 노드로 하나씩 보내며 이 remote frame을 받은 노드는 RTR 데이터를 보내게 된다. 이를 15~28의 우선순위를 노드의 번호순으로 부여하였다. 그 다음 각각의 노드의 센서정보들을 받는 것으로 스케줄링 하였으며 29~42의 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 전송 되었을 경우 3초를 넘지 말아야 실시간성이 보장 된다고 할 수 있다. 이 시스템을 시뮬레이션을 통하여 구현하고 125Kbps, 250Kbps, 500Kbps, 1Mbps의 전송속도로 전송하여 실시간 성을 분석한다.



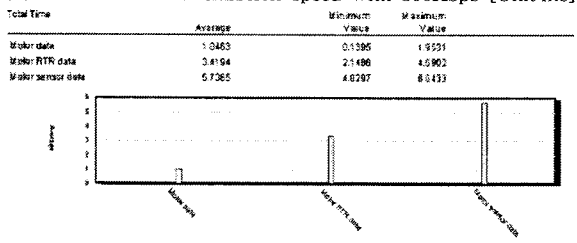
(a) The case of transmission speed with 125Kbps [Unit:ms]



(b) The case of transmission speed with 250Kbps [Unit:ms]



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



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

Fig.3 Network simulation result for humanoid robot's leg part using CAN protocol

위의 (a)~(d) 그림은 table 1에 맞추어 각각의 전송 시간에 따른 최악의 응답속도를 구한 결과이다. 125Kbps와 250Kbps의 전송 속도의 경우는 데드라인을 넘기므로 500Kbps이상의 전송 속도로 전송해야만 한다. 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.

5. 결 론

네트워크 시스템에서는 다양한 특성을 갖는 데이터들이 하나의 네트워크 미디엄을 공유하기 때문에 만일 네트워크 시스템이 부적절하게 설계된다면 실시간 데이터의 전송지연시간이 미리 지정된 한계치를 초과하여 네트워크에 접속된 응용 시스템의 성능을 저하 시키거나, 최악의 경우에는 네트워크 시스템과 네트워크에 접속된 응용 시스템이 불안정한 상태에 도달 할 수 있다.

본 논문에서는 휴머노이드 로봇의 네트워크 시스템 구조를 설계하고 제안하였으며 각 노드에서 발생되는 메시지를 스케줄링하고 수학적 모델링을 통하여 트래픽을 분석함으로써 신뢰성을 보장할 수 있는 전송속도를 결정하였다. CAN 프로토콜은 하나의 사건이 끝나면 다음 사건이 발생 하는 이벤트 트리거 방식으로 사용되어졌고, 오류가 발생하지 않는다는 가정 하에 실험하였다. 앞으로 가정을 줄이며 확실하고 더 안정된 실시간성을 가진 로봇을 위해 타임 트리거 방식의 CAN인 TTCAN(Time Triggered CAN)[7] 프로토콜의 적용이 필요할 것으로 생각된다.

참 고 문 헌

[1] Kaneko, K., Kajita, S., Yokoi, K., Hugel, V., Blazevic, P., Coiffet, P., "Design of LRP humanoid robot and its control method," Robot and Human Interactive Communication, 2001. Proceedings. 10th IEEE International Workshop on ,Pages:556 - 561, 18-21 Sept. 2001

[2] 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

[3] Tindell, K., Burns, A., "Guaranteed Message Latencies for Distributed Safety Critical Hard Real-Time Networks," YCS 229, Dept. Computer Science, Univ. of York (1994).

[4] 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.

[5] 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

[6] 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

[7] Amos Albert, Robert Bosch GmbH, "Comparison of Event-Triggered and Time-Triggered Concepts with Regard to Distributed Control Systems," Embedded World 2004, pages 235 - 252, Nurnberg, 17.-19. 02. 2004