

휴머노이드 로봇 ISHURO-II의 실시간 분산 제어를 위한 미들웨어 구현

Implementation of Middleware for Real-Time Distributed Control System of a Humanoid Robot Using CAN and TCP/IP

최우창*, 김진결**

Woo-Chang Choi* and Jin-Geol Kim**

Abstract - This paper deals with implementation of middleware using CAN(Controller Area Network) network and TCP/IP for real-time distributed control system of a humanoid robot. Existent system using CAN network is available. But, there is problems in extensibility and flexibility. In this paper, the new system using TCP/IP for solution and improvement of problems is proposed. The new system is applied to ISHURO-II, real-humanoid robot. The performance is verified through experiment.

Key Words :middleware, TCP/IP, CAN, humanoid robot

1. 서론

최근 지능형 로봇에 대한 관심이 높아지고, 그 중에서도 인간과의 상호작용과 생활환경에 적응할 수 있는 휴머노이드 로봇에 대한 관심은 더욱 대단하며 다양한 분야에서 연구가 활발하게 진행되고 있다. 휴머노이드 로봇은 인간과 유사한 구조를 가지므로 인간과의 협동 작업이 가능하며 인간이 수행하기 어렵고 위험한 작업을 대신할 수 있다.[1-3]

이러한 지능형 휴머노이드 로봇이 자율적인 보행과 인공지능 서비스의 제공을 하기 위해서는 로봇의 내부에 외부 환경을 인식하는 센서와 환경정보로부터 동작을 결정하는 연산부와 많은 관절들을 구동시키는 제어부가 필요하며 사용자의 명령을 쉽게 받을 수 있는 사용자 환경이 제공되어야 한다.[4-5] 이러한 조건들을 충족시키기 위해서 휴머노이드 로봇은 센서입력부와 연산부와 여러 제어부등을 분산하고 다양한 방식의 네트워크를 이용하여 통합하여 관리하고 있으며 이에 따른 연구도 활발하게 진행하고 있다.[6] 위의 선행연구들을 기반으로 휴머노이드 로봇인 ISHURO-I를 위한 네트워크 기반의 분산제어 시스템이 개발되었지만 연구를 진행하면서 기존의 네트워크 기반의 분산제어 시스템의 한계점들이 나타나게 되었다.

본 논문에서는 기존의 네트워크 시스템을 분석하고 기존 시스템의 문제점을 제시한 후, 이를 해결하기 위한 구조를 제안한다. 또 제안된 시스템을 새로운 시스템인 ISHURO-II에 적용하는 것을 목적으로 한다.

2. ISHURO-II 시스템 구성

2.1 기구부 구성

그림 1은 현재 개발중인 휴머노이드 로봇 ISHURO-II이다. 전체 24자유도로 각 다리에 6자유도, 각 팔에 3자유도, 몸체에 2자유도, 머리에 4자유도를 가진다.

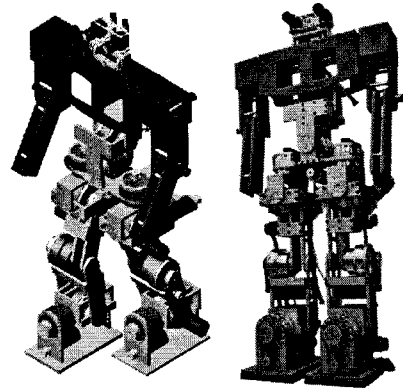


그림 1. ISHURO-II의 3D모델과 실제 모습
Fig. 1. 3D model and features of ISHURO-II

2.2 전장부 구성

휴머노이드 로봇은 다축 구동 시스템이며 구동기로 사용되는 모터를 제어하기 위해서는 고급의 서보 제어 기술과 계측 기술이 필수적이다. 그리고 구동축이 다수인 관계로 한 개의 고성능 CPU로 모든 구동축을 제어하는 방법보다 여러개의 CPU를 이용한 분산 제어 방법이 구성이 용이하고 확장성이 좋아지는 등의 장점을 가진다. 따라서 각 관절에는 모터의 서보제어와 메시지 컨트롤러와 CAN 통신이 가능한 로컬 컨트롤러

저자 소개

* 최우창: 인하대학교 전기공학부 석사과정

** 김진결: 인하대학교 전기공학부 정교수

트롤러 장착되어 있다. 메시지 컨트롤러는 로컬 컨트롤러와 메인 컨트롤러 사이의 정보의 이동을 관장한다. 메인 컨트롤러는 메시지 컨트롤러로부터 온 센서 정보를 수집하여 보행을 결정하거나 생성하여 다시 메시지 컨트롤러로 전송하는 역할을 수행한다.

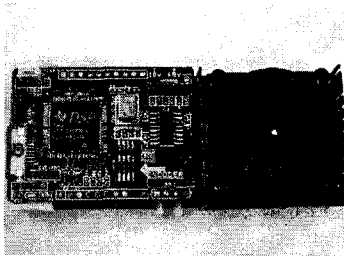


그림 2. ISHURO-II의 로컬 컨트롤러
Fig. 2. ISHURO-II local controller

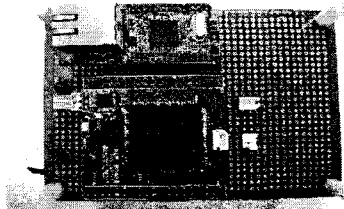


그림 3. ISHURO-II의 메시지 컨트롤러
Fig. 3. ISHURO-II message controller

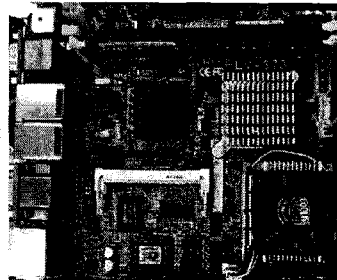


그림 4. ISHURO-II의 메인 컨트롤러
Fig. 4. ISHURO-II main controller

메인 컨트롤러는 고성능의 연산을 할 수 있고 편리한 개발 환경을 고려하여 PC와 OS는 윈도우XP를 사용하였다.

3. 분산처리 시스템구조

3.1.1 기존의 분산처리 시스템

그림 5는 기존의 분산처리 시스템의 구조를 보여준다. 이 구조는 크게 시뮬레이터, 유저 인터페이스 모듈, 메인컨트롤러, 메시지 컨트롤러, 로컬 컨트롤러로 구성되어 있다. 로봇 시뮬레이터에서 생성된 보행 정보가 유저 인터페이스 모듈에서 입력된 사용자의 명령과 함께 I2C 통신을 통해 메인컨트

롤러에 전송된다. 전송된 데이터는 메인컨트롤러에서 분석되어 사용자 명령, 시뮬레이션 데이터, 로봇의 센서 데이터를 함께 처리하여 로봇의 동작 정보는 모두 EMI 방식으로 메시지 컨트롤러에 전송된다. 메시지 컨트롤러에 전송된 동작 정보는 각 관절의 로컬 컨트롤러에 CAN 통신을 이용하여 전송되며 로컬컨트롤러의 센서정보와 제어정보는 CAN을 통하여 메시지 컨트롤러로 전송된다.

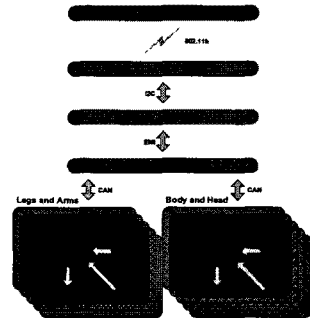


그림 5. ISHURO의 시스템 구조
Fig. 5. System architecture of ISHURO

3.1.2 기존 시스템 구조의 한계점

기존 시스템은 무선랜(802.11b)과 I2C, EMI, CAN, SCI 등의 통신 방법을 이용해 분산되어 있는 컨트롤러들을 관리 통합해 왔다. 하지만 너무 많은 통신 방식을 사용하고 네트워크 구조가 수직적이어서 시뮬레이터에서 들어온 정보가 유저 인터페이스 모듈을 거쳐서 메인컨트롤러까지 전송되어야 하는 불편함이 있어서 확장의 어려움뿐만 아니라 전송되는 데이터양이 단계를 거치면서 점점 커지는 구조였다. 또한 CAN통신으로 각각의 로컬 컨트롤러에 전송되는 모션 정보가 모터의 절대위치가 아닌 상대 위치로 전송되는 방식을 선택해 전송데이터는 적지만 하나의 동작이 완료되지 않은 상태에서 다른 동작을 시행하는데 문제가 있으며 시뮬레이션에도 어려움이 있었다.

3.2.1 새로운 분산처리 시스템

기존의 시스템에서 제기된 불만사항을 만족하기 위하여 새로운 구조의 분산처리 시스템을 설계하였다.

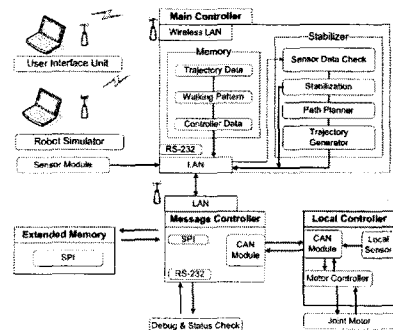


그림 6. ISHURO-II의 시스템 구조
Fig. 6. System architecture of ISHURO-II

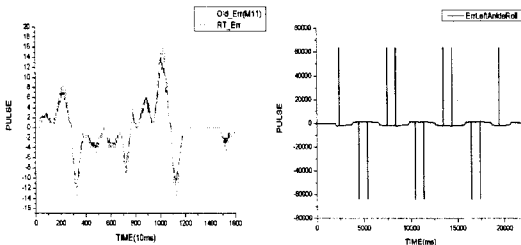
그림 6은 새롭게 설계된 분산 처리 구조를 보여준다. 실제 구동부인 로컬 컨트롤러들과 메시지 컨트롤러간의통신구조는 그대로 유지하고 메시지 컨트롤러에 새로운 센서입력을 받는 부분과 디버그와 상태체크를 위한 SCI 통신을 추가하고 구동과 관계된 통신은 모두 메시지 컨트롤러에 집중 시켰다. 시뮬레이터, 유저인터페이스 모듈, 지능 서비스를 위한 센서 등은 모두 메인 컨트롤러에 집중시키고 기존 시스템에서 사용하던 I2C나 EMI 통신은 배제하고 무선랜과 SCI 통신을 사용하였다. 메인컨트롤러에 사용되는 통신을 무선랜과 유선랜을 사용하게 됨으로 윈도우 프로그램인 유저인터페이스 모듈이나 로봇 시뮬레이터와 추가될 지능 서비스 프로그램이 보다 쉽게 휴머노이드 로봇을 컨트롤 할 수 있게 되었고 기존 시스템에서 다양한 통신 방식 때문에 어려웠던 디버깅 작업을 보다 쉽게 진행할 수 있게 되었다.

4. 실험 결과

4.1 실험환경

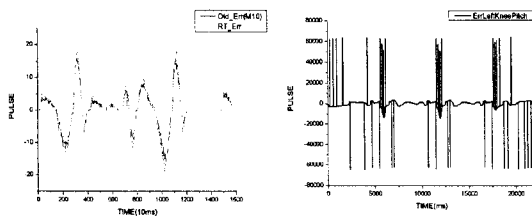
실험을 위하여 시뮬레이터를 이용하여 최적화된 보행 궤적을 생성하였다. 보행 궤적은 기존의 시스템은 2보 16초 궤적을 제안된 시스템은 6보 22초 궤적을 사용 하였다. 실험에서는 두 시스템 모두 로봇의 상체는 구동하지 않았다.

4.2 관절실험 결과



(a)기존 시스템의 오차 (b)새로운 시스템의 오차

그림 7. 좌측 발목 관절
Fig. 7. Left ankle joint



(a)기존 시스템의 오차 (b)새로운 시스템의 오차

그림 8. 좌측 무릎 관절
Fig. 8. Left knee joint

위의 실험결과에서기존의 시스템의 1펄스는 약 0.09도이며 제안된 시스템에서 1펄스는 약 0.0002도이다. 기존의 시스템 오차의 최대치는 20펄스로 약 1.8도이다. 하지만 제안된 시스

템에서 오차의 최대치는 6000펄스 정도로 실제 각도 1.3도 이다. 따라서 오차의 최대치가 더 작다. 또한 이 6000의 에러는 궤적의 급격한 변화에 따른 것이며 20ms의 시간에 사라짐을 알 수 있다. 기존의 시스템에서 오차가 보행 전반에 걸쳐서 발생하는 것에 비해 새로운 시스템은 오차의 절대적 크기도 작고 보상 시간도 빠르다. 따라서 새로이 설계된 시스템이 기존의 시스템에 비해 떨어지지 않는다는 것을 확인할 수 있다.

5. 결론 및 향후 과제

본 논문에서는 기존의 로봇의 분산제어 시스템의 한계점을 해결한 CAN과 TCP/IP 기반의 실시간 네트워크를 설계하고 구현 하였다. 또한 마지막으로 구현된 시스템을 이용하여 관절 모듈의 실험과 보행 실험을 실행하였다. 실험을 통하여 새로 설계된 시스템이 기존 시스템의 한계를 해결하면서 동시에 성능이 떨어지지 않음을 확인할 수 있었다.

향후에는 네트워크를 이용한 보다 다양한 지능형 서비스의 개발과 많은 불규칙한 네트워크 부하에도 안정적으로 동작하는 네트워크 관리에 관한 연구가 더 수행되어야 한다.

후 기

이 논문은 2005년 정부(교육인적자원부)의 재원으로 한국 학술진흥재단의 지원을 받아 수행된 연구임 (KRF-2005-041-D00320)

참 고 문 헌

- [1] N.Kanehira, T.U. Kawasaki, S. Ohta, T. Ismumi, T. Kawada, F. Kanehiro, S. Kajita and K. Kaneko, "Design and experiments of advanced leg module (HRP-2L) for humanoid robot (HRP-2) development," IEEE/RSJ International Conference on Intelligent Robots and System, vol.3, pp2455-2560, 2002.
- [2] K.Hirai, M. Hirose, Y. Haikawa and T. Takenaka, "The development of Honda humanoid robot," Robotics and Automation, Proceedings. IEEE International Conference on, vol.2, pp.1321-1326, 16-20 May 1998.
- [3] 이보희, "이족보행로봇 IWR의 걸음새 구현," 박사학위논문, 인하대학교, 1996.
- [4] 임선호, "이족보행로봇 IWR-III의 균형점 제어 구현에 관한 연구," 박사학위논문, 인하대학교, 1999.
- [5] 최상호, "이족보행로봇 IWR의 동적 안정도를 위한 균형점 해석에 관한 연구," 석사학위논문, 인하대학교, 1998.
- [6] 안상민, "휴머노이드 로봇의 분산제어를 위한 실시간 네트워크 구현" 석사학위논문, 인하대학교 2006