

분산형 제어구조를 가진 Mobile Manipulator 제어 시스템의 설계

황보영*, 김용석*, 조영조*, 유범재*, 오상록*
*한국과학기술연구원 지능제어연구센터, °한양대학교 전자공학과

Design of Distributed Control Architecture for Mobile Manipulator

Myung Hwang-bo*, Yong-Seok Kim*, Cho Young Cho*, Bum-Jae You*, Sang-Rok Oh*

*Intelligent Control System Research Center, KIST °Dept. of Electronic Eng., Han-Yang Univ.

Abstract - 인간과 함께 협력하는 서어비스 산업에서 요구되는 로봇은 고정된 환경에서 단순 반복 작업을 하는 기존 산업용 로봇의 개념을 뛰어 넘어 변화하는 환경에서 이동하면서 물체 이송 및 조작 기능을 수행하는 작업 형태를 취한다. 일반적으로 서어비스 로봇은 이동 능력과 조작능력이 겸비된 mobile manipulator의 외양을 갖는 것이 가장 응용 분야가 넓다.

이러한 mobile manipulator는 구동 축만 10축 내외가 되므로 축별 서보 제어기와 종류별 센서 제어기들을 분산화 시키는 구조가 신뢰성, 연산 능력의 측면에서 바람직하다. 개방형 구조로써 서보 제어기와 센서 제어기의 결합에 따른 비용 절감을 위하여 저가의 고속 통신망을 사용하고, 제어기 구성 요소들의 수급 다양화와 보수 유지의 간편화를 위하여 개방화된 인터페이스 하드웨어를 채택한다. 또한 자체 동력원에 의해 구동되는 것이 작업범위에 제한을 가하지 않으므로 로봇 제어기가 로봇 내부에 실장되어 오랜 시간 운용 가능하도록 소형화된 구조를 지니도록 설계한다.

1. 서 론

최근 로봇의 응용 분야가 기존의 제조 산업으로부터 우주 및 해저 개발, 의료 및 복지 산업, 원자력 설비 관리, 건설 및 토목 등으로 다양해지면서, 로봇 제이기도 응용 목적 맞추어 여러 형태로 개발 적용되고 있다. 그러나, 응용에 특수하게 적용되는 제어기의 하드웨어 및 소프트웨어 구조를 취하면 제품 매각자(vendor)에 좌우되는 가격, 특수한 인터페이스 사용, 높은 시스템 통합 비용, 작동 및 수리를 위한 특별 훈련의 요구 등 많은 문제를 가지게 된다. 따라서 이런 문제들의 해결방안으로 개방형 시스템의 개념에 기초하여 로봇 제어기의 하드웨어 및 소프트웨어 구조를 설계할 필요가 있다.

IEEE의 정의에 따르면, 개방형 시스템이란 "복수의 매각자로부터 공급되는 다양한 플랫폼에 응용되고, 다른 시스템의 응용 프로그램과 내부적으로 상호 결합 운용되며, 사용자와 일관된 형식으로 상호 작용하는 능력을 갖춘 시스템"을 의미한다. 즉, 사용자의 입장에서 개방형 시스템의 두 가지 대표적인 특징으로, (1) 매각자 중립성(vendor-neutrality)과 (2) 구성요소의 결합능력(component-integrating-ability)을 들 수 있다[1]. 매각자 중립성에 따라 시스템은 단일 매각자에 종속되지 않도록 설정된 표준에 따라 설계되어야 하며, 시스템이 쉽게 결합되어 제 성능을 발휘할 수 있도록 모듈화 및 침식과 확장이 용이하여야 한다. 개방형 제어기에 관한 최신의 표준화 동향으로 유럽 공동체의 ESPRIT III 프로젝트 OSACA(Open System Architecture for Controls within Automation Systems)와 미국 GM Powertrain Group의 OMAC(Open Modular Architecture Controller) 프로젝트를 들 수 있다. OSACA에서는 일반 자동화 시스템에서 요구되는 개방형 제어기의 사양을 정의하고 시스템 플랫폼의 원형

(prototype)을 구현하였으며[2][3], OMAC에서는 자동차 산업에서 필요로 하는 제어기에 대하여 경제성, 보수성, 개방성, 모듈성, 확장성 등을 고려하여 하드웨어 및 소프트웨어의 표준을 정하고 있다[4]. 그러나 표준안들이 산업용 제어기를 위주로 정의되어 있어 로봇 제어기를 대상으로 할 때 좀 더 보완되어야 한다. 한편, 최근의 로봇 연구 동향이 고정된 환경에서 단순 반복 작업의 산업용 로봇으로부터 환경이 변화하고 이동 및 물체 조작 작업을 동시에 수행하는 서어비스 로봇으로 확장해 가는 추세에 있어, 로봇 제어기 및 전원의 자체 탑재를 위한 제어기의 소형화가 절실히 요구된다. 기존의 산업용 로봇 제어기에서도 유지 보수의 간편화와 연관된 문제로써 큰 의미를 갖는다.

다양한 응용을 갖는 로봇 제어기에 있어서 개방화 및 소형화 문제는 제품 자체의 원가 절감 뿐만 아니라 설치, 운전, 유지, 보수 등에 수반되는 막대한 엔지니어링 비용 절감을 가져오는 대책으로서 최근 많은 로봇 연구자에게 주목을 받고 있다. 최신의 산업용 로봇 제어기에 있어서 개방화의 경향은 PC를 기반으로 하드웨어 및 소프트웨어를 구축하여 표준화된 Windows GUI 및 ISA/PIC/Compact PCI등의 표준 인터페이스로 하드웨어 및 그 라이브러리를 구축하는 방향을 취하고 있다[5][6]. PC를 기반으로 제어기를 구성하는 것은 개방화에 접근하는 한가지 방법이지만 하나, 그에 인터페이스되는 서보 제어기 하드웨어에 종속적으로 제어기 소프트웨어가 구축된다면 개방형 제어기로 취급될 수 없다. 로봇 제어기의 개방화를 위해서는 오히려 하드웨어 종속성이 높은 서보 제어기 부분을 모듈화하고 표준화하는 방안이 더 바람직하게 생각된다[7]. 본 논문에서는 탑재형 로봇 제어기를 주 대상으로 개방화와 소형화를 추구하는 방안으로서, 로봇제어기의 동작 제어부와 서보 제어부간을 표준화된 실시간 통신망으로 결합하는 형태의 네트워킹 기반 개방형 제어 구조를 제안한다.

2. 네트워킹 기반 개방형 로봇 제어구조

일반적으로 로봇 제어기는 사용자 인터페이스부, 호스트 인터페이스부, 동작 제어부, 서보 제어부 및 센서 신호처리부로 구분될 수 있으며, 다음과 같은 기능의 하드웨어 및 소프트웨어를 갖는다.

- (1) 사용자 인터페이스부: 로봇 제어 언어의 편집 및 실행, 운전 모니터링 기능 등을 담당하며, 통상 그래픽 모니터, 터치스크린이나 마우스 등의 하드웨어 및 Windows GUI 소프트웨어를 갖는다. 탑재형 제어기의 경우 이 부분의 역할은 호스트 컴퓨터에서 이루어질 수 있다.
- (2) 호스트 인터페이스부: 하드웨어로 주로 Ethernet 드라이버 모듈이나 RS-232C/422 드라이버를 갖고 TCP/IP 프로토콜이나 Stream I/O function에 의해 상위 계층의 제어기에 의해 조종을 받을 수 있게 되어 있다.

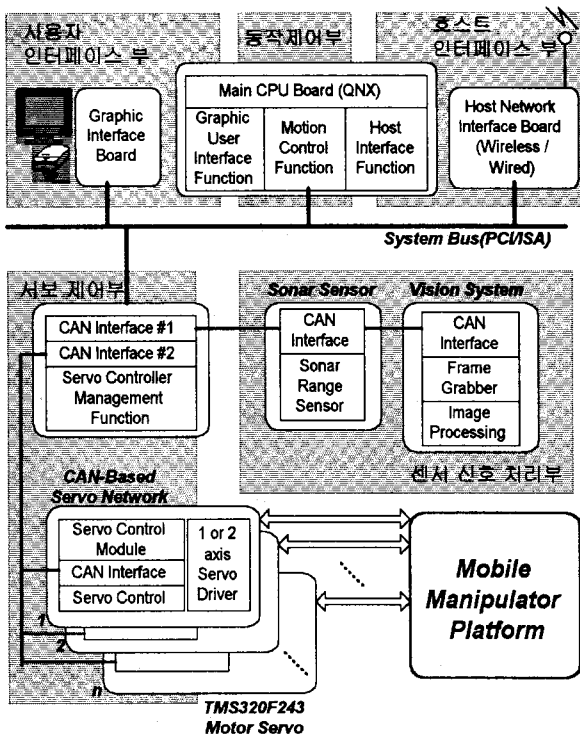


그림 1 네트워크 기반 개방형 로봇제어의 개념도

- (3) 동작 제어부: 로봇 언어의 해석과 궤적 계획 및 기구학적 동역학적 주 제어를 담당하는 핵심부이며, 컴퓨터 메인 보드에서 소프트웨어로 처리된다.
- (4) 서보 제어부: 로봇의 각 관절부의 위치나 토오크 등의 명령치를 동작 제어부로부터 전달 받아 모터의 위치제어나 토오크 제어 기능을 수행하며, 하드웨어로 서보 드라이브 회로 및 서보 제어용 마이크로프로세서를 갖고 동작 제어부 인터페이스와 서보 제어 소프트웨어를 firmware로 내장한다.
- (5) 센서 신호처리부: 로봇의 동작 제어를 위한 센서 신호를 처리하는 부분으로 응용에 따라 센서의 종류가 달라지므로 하드웨어 및 소프트웨어가 주로 시스템의 선택사항이 된다. 보편적으로 시각기능을 위해 image frame grabber 및 실시간 영상처리 모듈 등의 하드웨어를 내장하고 물체의 위치/자세 정보를 시스템 버스를 통해 동작 제어로 전달한다. 이동 로봇의 경우 장애물 회피나 로봇 자체의 위치 정보 추출을 위하여 초음파, Gyro 신호 처리 기능을 갖는 것이 일반적이다.

위의 5가지 부분 중에서 현재 로봇 제어기 개방화에 가장 큰 장애가 되는 부분은 서보 제어부로서, 로봇 응용에 따라 하드웨어 구성이 달라지고, 그에 맞춘 동작 제어기 인터페이스 하드웨어 및 소프트웨어의 특성화는 로봇 제어기의 특성화로 이어지게 된다. 특히, 최근의 로봇 응용이 공간 이동과 물체 조작 작업이 병용되는 분야로 확대되면서 로봇 자유도는 증가되고 탑재형 제어기의 필요성이 증가됨에 따라, 서보 제어부의 인터페이스 개방화 및 소형화는 바로 로봇 제어기의 개방화, 소형화 문제를 동일한 문제로 취급될 수 있다.

동작 제어기와 서보 제어기 간의 인터페이스는 VME bus나 PCI bus 등 시스템 버스를 사용하는 것이 보통인데, 이 경우 하나의 서보 제어기에서 다루는 서보 루프의 갯수와 데이터 교환 방식이 제품마다 다르므로 개방화에 큰 장애 요인이 된다. 또한, 응용에 따라 서보

루프의 종류 및 갯수가 달라질 수 있다는 것도 버스 방식의 인터페이스에서의 문제점이 될 수 있다. 한편, 서보 제어가 시스템 버스로 인터페이스되면, 로봇 제어가 하나의 큰 하드웨어로 통합되어 서보 제어를 위한 모든 전선 케이블들이 모든 로봇 관절들로 부터 통합 제어기로 집중되어야 하므로, 케이블 처리 문제가 제어가 소형화 및 보수 유지의 간편화를 가로막는 요인이 될 수 있다. 본 고에서 제안하는 네트워크 기반 로봇 제어기에서는 동작 제어기와 서보 제어기와의 인터페이스를 위하여 CAN(Controller Area Network)과 같은 표준화된 실시간 통신망을 사용하는 구조를 취함으로써, 개방화, 소형화, 유지 보수의 간편화를 지향하고자 한다[8]. 그림 1은 네트워크 기반 로봇 제어기의 개념도를 나타낸 것으로, 서보 네트워크의 한 node는 한 개의 서보 루프를 담당하게 된다. 특히, CAN은 자동차의 실시간 통신망을 대표하고 있으며 로봇과의 응용 유사성이 높고 서보 제어와 네트워크 겸용 마이크로 컨트롤러가 많이 상용화되어 있기 때문이다. CAN이 통신 1 Mbps의 통신속도로 한번에 8 byte의 데이터를 포함한 최대 14 byte의 프레임을 전송할 수 있기 때문에 8개 서보 루프를 갖는 서보 제어를 가상하면 로봇 제어의 샘플 주기를 1 ms 이내로 맞출 수 있어 서보 제어의 실시간 성능을 충분히 보장할 수 있는 강점도 있다.

3. CAN 기반 서보 모터 제어기

Mobile manipulator의 각 축 서보 제어에는 TI사의 DSP TMS320F243을 사용하여 설계하였다. 이는 분산형 제어에 적합한 CAN 모듈이 있어 개방형 구조로 전체 시스템을 구성하기가 용이하고 저전력으로 동작할 수 있고 DSP안에 각종 필요한 모듈이 있으므로 다른 외부 장치를 탑재할 필요 없이 사용이 가능하여 채택하였다. 본 논문에서 제안하는 DSP 제어기의 기본적인 구성은 그림 2과 같이 piggyback 구조로 되어 있다. 즉, TMS320F243으로 기본 DSP 코어(core) 모듈을 구성하고 모터 드라이버 모듈 위에 코어를 얹는 방식으로 구성된다. 또한 하나의 DSP 제어기 2축의 DC 모터를 구동하도록 설계하여 통신부하를 줄이고 하드웨어의 크기 및 내부에 부착되는 서보 제어기의 수를 줄일 수 있다.

3.1 DSP 코어 모듈

TMS320F243은 외부에서 5MHz 크리스탈 입력을 받아서 내부의 PLL을 사용하여 4배를 한 20MHz로 동작한다. DSP안의 내부 메모리는 16KB Flash EEPROM과 1088byte Data/Program Dual-Access RAM(DARAM)이 장착되어 최종 프로그램을 내부 메모리에 탑재하기가 용이하다. SCI 통신과 CAN 통신이 내장되어서 다른 시스템 또는 PC와의 통신을 할 수 있고, 프로그램 가능한 멀티플렉스 I/O와 8개의 A/DC(Analog Digital Conversion)이 있으므로 외부의 다른 인터페이스 회로를 구성할 필요가 없다. 그리고 watchdog(WDT) 타이머 모듈은 시스템의 에러가 발생 시 리셋신호를 발생하도록 되어있어서 시스템을 안정화시킬 수 있다.

DSP 코어 주변회로는 프로그램 RAM(128KB)과 데이터 RAM(128KB)을 확장 외부메모리에서 탑재하여 프로그램을 개발하는데 용이 하도록 하였고, RS232-C 드라이버(MAX233)를 탑재하여 프로그램 개발을 위한 모뎀링 역할을 하게 하였고, DIP 스위치와 CAN 드라이버(U5350)를 각 제어기에 탑재하여 고유 ID를 할당하여 여러 개의 노드를 연결하여 분산네트워크 제어를 하도록 하였다. DSP에서 모터제어는 PWM(Pulse Width Modulation)방식으로 2축의 모터를 제어하고, TMS320F243은 8개의 10bit A/DC가 있어 각 모터 드라이버에서 출력되는 전류 측정과 전

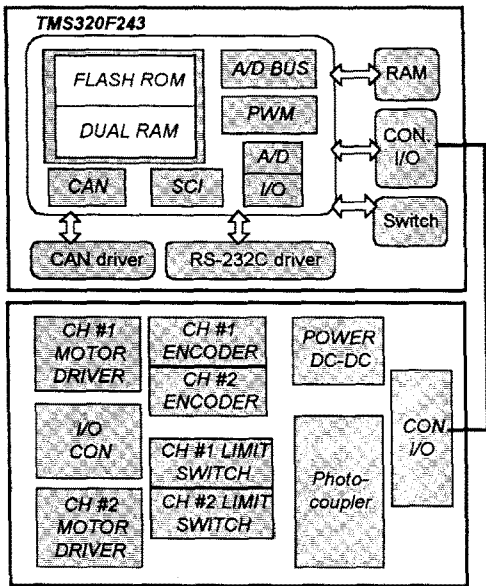


그림 2 TMS320F243을 이용한 네트워크 기반 서보 모터제어기의 구조

원 감시를 할 수 있게 설계되었다.

3.2 모터 드라이버 모듈

모터 드라이버 모듈은 크게 전원 및 모터 드라이버 두 가지로 부분으로 구성된다. 기본적인 전원공급은 배터리와 같은 탑재 전원에서부터 24V를 입력받아 DC-DC 컨버터를 통하여 5V로 DSP 코어 모듈과 모터 드라이버 모듈에 공급하고 모터 드라이버의 전력 전원은 배터리에서 24V를 직접 연결하였다. DC-DC 컨버터는 전원을 분리(isolation)시켜 줌으로써 DSP 제어기 시스템 전체의 전원을 안정화하고 외부로부터 단일 전원만을 요구하므로 배선이 단순해진다.

모터 드라이버는 National semiconductor사의 LMD18200을 사용하여 모터를 72W(24V-3A)의 전력으로 구동한다. 외부에 24bit 카운터(LS7166)를 탑재하여 모터에서 나오는 인코더 신호를 처리하며 리미트 스위치는 각 채널 당 3개씩 할당하였다. 모든 신호들은 노이즈 방지를 위해 외부회로와 분리(isolation)시키기 위해 photo-coupler를 사용하였다. 또한 OP-AMP와 photo-coupler를 이용하여 각 채널의 motor 드라이버에서 나오는 전류 측정 회로를 구성하였다.

3.3 CAN을 이용한 분산처리 네트워크

CAN 통신은 필드 버스의 하위 레벨에서 가장 폭 넓게 사용되고 있다. CAN 통신은 각 상호간의 노드(node)에 메시지를 전달을 통해 통신한다. 모든 메시지 처리는 다른 메시지에 비해 메시지 데이터가 짧은 주기인 실시간 처리, 발생 빈도가 낮고 메시지 데이터가 짧으나 긴급 상황에 해당하는 메시지로 반드시 제한된 시간 안에 처리해야 하는 실시간 처리 및 우선 순위가 가장 낮은 비실시간 메시지처리로 구성되어 있다. CAN 통신의 특징은 중앙 집중 방식보다는 노드를 추가하기 용이하고 각 노드간의 상호기능을 수행할 수 있으므로 네트워크 제어기 구축에 편리하다.

그러나, 노드 수가 증가하면 자연적으로 트래픽이 증가하므로 그림 1의 CAN 네트워크에서 한 개의 제어기에서 2개의 모터 제어를 담당하는 방법으로 트래픽을 줄일 수 있다. CAN 프로토콜을 이용하는 네트워크 기반 제어 시스템의 노드에서 발생하는 메시지의 발생 분

포와 전송 시간을 계산하여 모터 제어기뿐만 아니라 각종 추가되는 다른 장치의 메시지가 실시간을 보장하는 프로토콜을 이용하여 모든 노드가 실시간으로 응답하도록 하였다.

4. 결 론

다양한 응용을 갖는 로봇 제어기에 있어서 개방화 및 소형화 문제는 제품 자체의 원가 절감 뿐만 아니라 설치, 운전, 유지, 보수 등에 수반되는 막대한 엔지니어링 비용 절감을 가져오는 최선의 대책이라 할 수 있다. 특히, 서보 제어부의 개방화 및 소형화는 전체 로봇 제어기에 가장 큰 영향을 미치게 되므로, 본 논문에서는 이 문제에 초점을 맞추어 네트워크 기반 개방형 제어기 구조의 개념을 제안하고 mobile manipulator에 적용해 보았다. 현재의 기술 동향을 볼 때 향후 다양한 산업 환경에 적용될 로봇 제어기의 개방화/표준화/소형화 문제는 제어기 개발에 필수적인 개념이 되고 있어 점차 개방화/소형화의 추세에 맞추어 제어기를 구현해 가고 있다. 특히, 초소형의 네트워크 기반 관절 서보 제어기는 로봇 기구부 내부에 장착하여 로봇과 동작 제어용 컴퓨터 사이에 두 가닥의 통신선과 두 가닥의 전원선 만을 연결하는 로봇 제어기로 구축하게 되어 로봇 제어기의 설치, 운전, 보수, 유지 비용을 상당히 줄일 수 있을 것으로 기대된다.

[참고 문헌]

- [1] J. Park, Z.J. Pasek, Y. Shan, Y. Koren, K.G. Shin, G. Ulsoy, "An Open Architecture Controller for Machining Processes," *Manufacturing Systems*, pp. 23-27, Vol. 25, No.1, 1996
- [2] P. Lutz and W. Sperling, "OSACA - the vendor neutral Control Architecture," *Proc. of the European Conference on Integration in Manufacturing*, Dresden, Germany, 1997.
- [3] W. Sperling and P. Lutz, "Designing Applications for an OSACA Control," *Proc. of the Int'l Mech. Engineering Congress and Exposition*, Dallas, USA, Nov., 1997.
- [4] GM Powertrain Group, Open, Modular Architecture Controls at GM Powertrain - Technology and Implementation: Version 1.0, <http://www.arcweb.com/omac/>, May 14, 1996
- [5] Kazuhiko Onoue, "The Development of PC Based Robot Controller," pp. 20-23, *Robot(Japanese)*, No. 121, March 1998.
- [6] Tasato Tanaka, "PC Based Robot Controller," pp. 39-42, *Robot(Japanese)*, No. 121, March 1998.
- [7] Ken Oonishi, "The Open Controller for the MHI PA-10 Robot," pp. 34-38, *Robot(Japanese)*, No. 121, March 1998.
- [8] ISO/TC 184/SC 5/WG 2, Road vehicles - Interchange of Digital Information - Controller area network (CAN) for high-speed communication, *ISO 11898:1993(E) Working Document*, 1993
- [9] F.-Y. Wang and G.N. Saridis, "Task Translation and Integration Specification in Intelligent Machines", *IEEE Trans. on Robotics and Automation*, Vol. 9, No. 3, pp. 257-271, 1993.