

Fara Robot에서의 RCCL(Robot Control C Library) 구현

선경일 김병국
한국과학기술원 전기및 전자과

Implementation of RCCL on Fara Robot

K.I. Sun B.K. Kim
Dept. of Electrical Engineering, KAIST

Abstract

An intelligent robot control system is developed, which is based on extensible hardwares and softwares. The system could be used to test advanced and complex real time application programs to avoid constraints on present robot control system in executing a complex or precise algorithms, due to the limitation of hardware and software.

In this paper we used the RCCL(Robot Control C Library) on SUN4 as a supervisory system that plays the path planning and man-machine interface. And we used VxWORKS as a real time OS on a VME bus CPU equipped with some interface boards. Two systems were connected through the Ethernet network.

We used the 4 axis manipulator, FARA, developed by Samsung Electronics Co.

1 서론

이제까지의 로봇 매니플레이터의 제어는 로봇 프로그래밍 언어와 밀접하게 관련되어 왔다. 로봇을 프로그래밍 할 수 있다는 것은 전통적인 자동화 측면에서 볼때 큰 장점이었으므로 이런 로봇제어 언어는 로봇이 하는 일에 관한 적절한 규격을 제공하는 것이었다.

그러나 최근에 오면서 로봇이 혼자 독립적으로 작업한다는 것에서 탈피되어 전체 제조시스템(manufacturing system)에서 한 부품으로서의 역할을 하게 되었다. 그 결과 로봇언어는 로봇사이의 인터페이스, 작업환경(working environment) 등을 다루기 위해 좀 더 강력하게 만들어져야만 했다. 또한 센서

나 비전(vision), 혹은 제어 알고리즘 분야의 발전에서 이루어진 것들을 쉽게 수용하기 위한 유연하고 확장성이 있는 시스템 개발도 요구 되어졌다.

현재 로봇 프로그래밍 시스템의 대부분은 특정한 언어를 가진 로봇 콘트롤러를 기초로 하고 있다. 이런 로봇제어 언어는 로봇의 실시간 제어, 공간 좌표(world modelling), 센서로 부터의 제한, 입출력의 복잡성들을 극복해야만 했다. 또한 사용자와 로봇 혹은 로봇과 로봇 사이의 통신도 다룰 수 있어야만 했다. 조립작업등에서는 로봇간의 협력작업은 절실히 요구되었으며 이는 다중 로봇트 제어의 필요성을 제기 하였다.

이러한 요구에 부응하여, 다중 로봇트 제어를 위한 하드웨어로서 워크스테이션을 관리 제어기로 사용 하였으며 다중 CPU에 의한 다중 로봇트 제어가 가능하도록 확장성이 가능한 VME 턱을 사용하였다. 이때 관리제어기와 로봇트 제어기는 network을 통해 서로 연결이 되어있다. 소프트웨어 측면에서도 유연성을 높이기 위해 RCCL을 전체 제어 언어로 채택하였다. RCCL은 로봇 종속적인 데이터들을 라이브러리화 하여 제어 대상인 로봇이 바뀌어도 단지 새로운 로봇에 대한 데이터만을 입력 시켜줌으로서 큰 변화없이 빠른 시일안에 시스템을 재 구성할 수 있고, 사용자가 새로운 제어 방식을 사용하고자 원할때에는 단지 C 언어로 새로운 제어 루틴을 추가함으로써 적용시킬수 있는 로봇제어 환경을 제시하고 있다. 원래 VAX상에서 PUMA를 제어하기 위해 쓰여졌지만, 본 논문에서 제시하는 것처럼 이를 SUN4상에서 스카라(SCARA)형의 로봇을 제어하도록 하는 환경으로 변경시켰다. 이것으로부터 전체적인 프로그래밍이 융통성이 있음을 알 수 있다.

또한 여러대의 로봇이 협력하여 작업할 때 문제가 되는 로봇사이의 통신과 실시간 제어 등을 위하여 실시간 운영체제 VxWORKS를 사용하게 되었다. 실시간 운영체제 VxWORKS는 멀티 태스킹을 기초로 하기때문에 태스크간의 통신을 위해 메시

지 큐(message queue), 동기화를 위한 세마포어(semaphore), 태스크간의 스케줄링을 위한 태스크 우선순위 (task priority), 선점(preemption) 등을 제공한다.

센서나 비전등의 분야에서 발전된 것들을 즉각 수용하 위해서는 유연한(flexible) 구조의 하드웨어 구성도 필수적이다. 본 논문에서는 세계적으로 널리 쓰이는 VME 버스플 기초로 하여 만들어진 VME 랙(rack)을 사용하였다. 최대 16개의 보드를 장착할 수 있어서 사용자가 다른 기능들을 추가하고 싶은 경우, 부가시키고 싶은 기능을 보드 단위로 만들어서 랙에 장착함으로써 그 기능을 추가할 수 있다.

본 논문은 다음과 같이 구성되어 있다.

2.1에서는 전체 시스템의 하드웨어 측면을 소개하고 2.2에서는 전체 소프트웨어 환경을 설명한다. 2.3에서는 강력한 실시간 운영 체제 VxWORKS를 바탕으로 제어되는 실시간 제어기에 대해, 2.4에서는 로봇 인터페이스 보드에 대해 설명한다.

2 본론

2.1 범용 로봇 제어 시스템 구성

본 논문에서 제시하는 시스템 구조는 그림 1과 같다. 전체 로

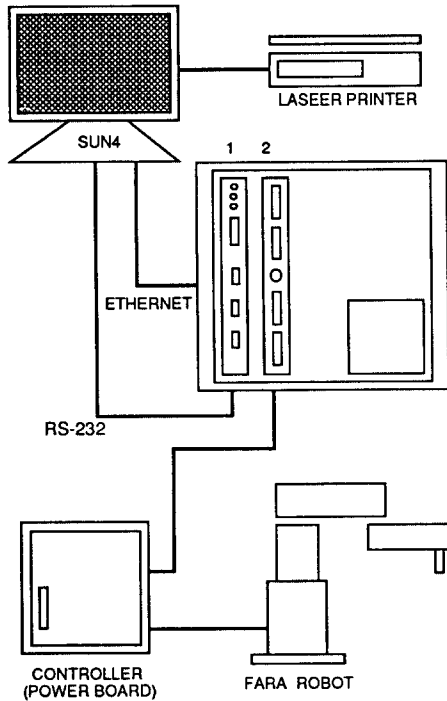


그림 1. 하드웨어 구성

봇 제어는 SUN4상의 RCCL 로서 디지털 서보를 통해 로봇에 명령을 전달할 수 있다. 일반적인 상위 로봇제어 프로그램이 SUN4상에서 돌아간다. SUN4와 디지털 서보는 이더넷(ethernet)으로 물려 있으며 소프트웨어 개발시의 서보 프로그램들을 다운로드(download)하게 된다. 디지털 서보는 슬롯(slot) 1에 위치해 있다. 슬롯1은 디지털 제어기 보드로서 VxWORKS 실시간 운영체제가 돌아간다. 64Kbyte 데이터 버퍼(data buffer)를 채용한 LAN 컨트롤러가 있으며 16Mbyte의 공유메모리(shared memory)를 가진다. 이 보드에서는 하드웨어(hardware)에 의존하는 제어루틴들이 돌아간다.

슬롯 2는 로봇 인터페이스 보드로서 슬롯 1의 디지털 서보로부터의 명령을 컨트롤러의 아날로그(analog) 보드에 전달하게 된다. 슬롯 2는 FARA에 대한 4축 인터페이스 보드이다. 각 슬롯들은 VME 후면(back plane) 버스로 연결되어서 데이터를 교환한다. 원래 FARA 컨트롤러에 있던 디지털 서보는 사용하지 않고 cpu-30x 보드 하나로서 이를 대체하며 아날로그 보드는 컨트롤러에 있는 것을 그대로 사용한다.

FARA는 삼성에서 개발한 스카라(SCARA)형의 로봇이다

2.2 범용 로봇 제어 소프트웨어 환경

위와 같은 하드웨어 구조를 채택한 것은 시스템 개발 측면에서의 용이성 때문이다. 이제까지는 로봇 시스템을 개조하고자 할 때, 혹은 진보된 로봇 알고리즘을 적용하려 할 때, 하드웨어에 크게 의존하여 소프트웨어 개발에 많은 어려움이 따랐었다. 학문을 배우고 이를 적용시켜 보는 측의 입장에서는 좀 더 빠르게 새로운 알고리즘들을 적용시켜 볼 수 있고, 쉽게 소프트웨어를 바꿀수 있는 환경이 필요하게 되었다. 본 시스템에서는 이러한 맥락하에서 디지털 서보와 SUN4를 ethernet으로 연결하여 완성된 프로그램을 만들기 까지 여러 프로그램을 다운로드 해서 실험해 볼 수 있도록 하였다. 그림 2에서 이를 설명하고 있다.

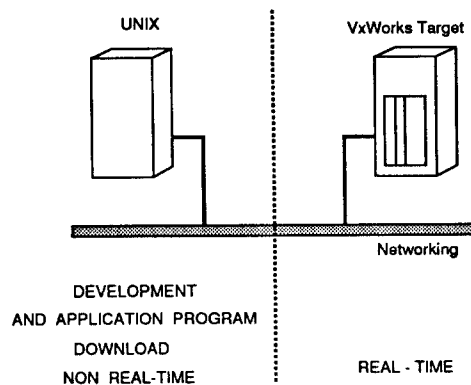


그림 2. 소프트웨어 환경

다른 이용로서는 확장성을 들 수 있겠다. 타깃을 VME 렉에 장착했기 때문에 다른 기능을 추가하고 싶은 경우 사용자가 원하는 각종 CPU 보드나 I/O 보드를 장착함으로써 해결될 수 있다. 이런 환경은 다중 로봇으로의 확장에 필수적이다.

소프트 환경은 크게 SUN4에서 돌아가는 RCCL 상위레벨 프로그램과 Target 보드 1에서 실행되는 하위 레벨 프로그램이다.

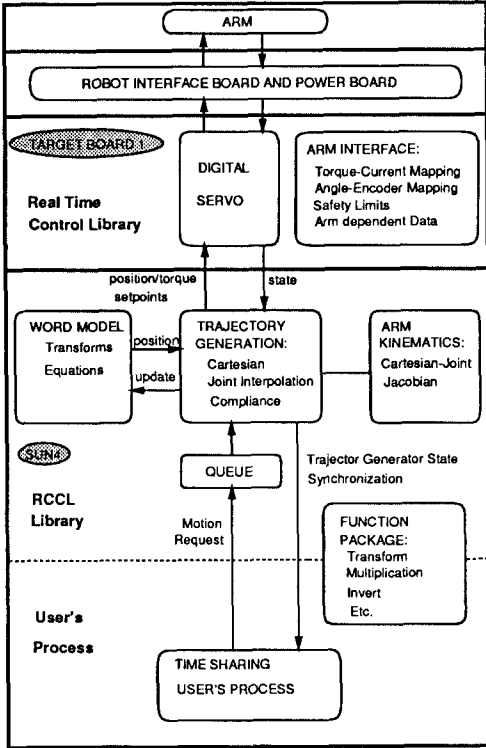


그림 3. RCCL 프로그램 구성

상위 레벨에서는 RCCL이 제공하는 라이브러리를 이용, 사용자가 프로그램을 작성하며 이를 실행하므로써 궤적생성기 (Trajectory Generator)에 의해 하위 레벨에 로봇의 목표치를 전달해 주게 된다. 이때 궤적 생성기는 하나의 독립된 태스크로서 존재하며 매 샘플타임마다 디지털 서보에게 그 값을 전달해 주게 된다. 하위 레벨은 실시간 운영체제를 이용한 여러 태스크 들로서 구성되며 로봇 인터페이스 보드와의 직접적인 통신을 담당하게 된다. 이곳에서는 로봇 인터페이스와 관련한 라이브러리 (Library)들이 존재한다. 원래 RCCL은 PUMA 로봇을 대상으로 만들어져 있어서 PUMA 컨트롤러에 관한 라이브러리들만이 존재하므로 FARA의 컨트롤러에 관한 자료들을 라이브러리화 하여 입력 시켰다. 토오크와 전류간의 매핑(mapping), 조인트 각도와 엔코더의 매핑, FARA 각 조인트의 좌우 제한 (Left and right limit), 각 포트(port)의 어드레스, 각 조인트

의 디지털 서보에 대한 PID 이득 등이 입력 자료이다.

2.3 실시간 제어기(Real Time Controller)

디지털 서보는 실시간 운영체제 VxWORKS에 의해 운영되는 보드로서 VME 렉의 시스템 컨트롤러 위치인 가장 좌측에 위치해 있다. VME 후면(back plane)과 SUN4의 통신을 가능하게 하는 게이트 어레이(gate array)로서의 역할도 수행하게 된다. FARA 로봇의 4축을 제어할 만큼 빠른 계산속도를 지니며 멀티태스킹(multitasking)이 가능하다.

내부 알고리즘을 간단히 설명하면 그림 4와 같다.

SUN4상에서의 RCCL과 디지털 서보는 UNIX BSD 4.2 ver. 이상의 운영체제에서 제공하는 인터넷 통신(internet communication) 수단인 소켓(socket)을 통하여서만이 가능하다. 소켓을 기초로한 통신에서는 서버(server)와 클라이언트(client) 개념을 사용한다. SUN4상의 하위 디바이스 드라이버인 vemDrive를 클라이언트로 하고, 디지털 서보에서는 일정시간 마다 로봇의 목표치인 각 조인트값을 받거나 스테이트(state)보내기 위해 새로운 태스크인 메시지 태스크(message task)를 두어서 이를 서버로 정하였다. 메시지 태스크는 항상 SUN4에서 메시지가 올때까지 기다리는 상태(blocking)이며, 비로소 vemDrive로부터 메시지를 받으면 자신의 동작 루틴을 수행하게 된다. 컨트롤 태스크(control task)는 SUN4에서 받은 메시지를 기초로 하여 각 조인트의 목표치를 읽어내며, 이를 여러개로 보간(interpolation)한 뒤, 각 조인트 별로 제어를 수행하게 된다. 이때 메시지 태스크와 컨트롤 태스크는 서로 독립적으로 수행되므로 서로간의 통신을 필요로 하게 되는데 이는 실시간 운영체제 VxWORKS에서 제공하는 공유 메모리(shared memory)와 메시지 큐(message queue)를 사용하게 된다.

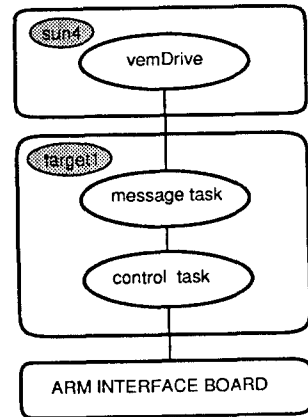


그림 4. 디지털 서보 제어 태스크 구성

2.4 로봇 인터페이스 보드 (Robot Interface Board)

원래 FARA 콘트롤러에 있던 4개의 디지털 서보를 버리고 하나의 CPU-30x 보드로 대체하면서, VME 랙에 장착된 보드와 FARA 콘트롤러에 있는 네개의 아날로그 보드가 서로 데이터를 주고 받을 수 있는 인터페이스 보드가 필요로 하게 되었다. 이 보드는 단순히 디지털 서보 보드에서 나오는 디지털 값을 아날로그 서보에게 디지털 혹은 아날로그로 보내주며, 4개의 아날로그 보드로 부터 케환(feedback)되는 아날로그 값이나 디지털 값을 디지털 서보에게 전해주는 용도로써 VME 버스 규격에 따라 제작된 보드이다.

속도 명령(velocity command)를 출력하기 위한 D/A 변환기, 로봇에 PWM 전원을 넣는 다거나 에러(error)가 발생하였을 때 에러를 클리어하기 위한 디지털 출력, FARA로부터 위치를 케환 받는 디지털 입력 등의 신호를 사용한다.

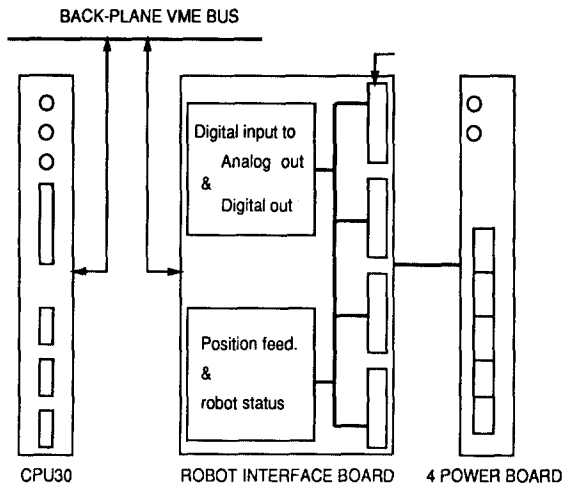


그림 5. 로봇 인터페이스 계략도

3 결론 및 추후과제

본 연구에서는 워크스테이션(workstation), 다중 CPU를 사용한 다중 로봇 제어 시스템 하드웨어를 구현 하였고, 범용 로봇 제어 언어인 RCCL을 SCARA 형의 로봇 FARA에 적용하였으며 멀티 유연한(flexible) 시스템 구성에 중점을 두어 구성하였다. 이런 시스템 구성은 이제까지 하드웨어와 소프트웨어의 제약으로 말미암아 복잡하고도 정교한 알고리즘 수행에 문제가 많았던 현 컴퓨터 환경을 극복할 수 있고 미래 지향적인 로봇 시스템임을 확신한다.

앞으로의 과제로서 다중로봇에의 확장과 진보된 알고리즘의

적용을 들 수 있겠다. 사용자와의 인터페이스를 위하여 그래픽(graphic) 기능도 추가할 수 있다.

Reference

1. FARA SCARA ROBOT SM2/SS2 OPERATOR GUIDE
2. 이 덕만, 오 중한, 이 진수, "실시간 운영체제를 이용한 범용 로봇 제어 언어의 개발", KACC, 1991
3. 배 본호, 이 진수, "PUMA Robot에서의 RCCL(Robot Control C Library) 의 구현", KACC, 1991
4. Hayward V., Paul R. P., "Introduction to RCCL: A Robot Control C Library", IEEE First International Conference on Robotics, Atlanta, June 1984
5. Vincent Hayward, Richard P. Paul, "Robot Manipulator Control under Unix RCCL: A Robot Control "C" Library", Int. Journal of Robotics Research, 1986
6. 삼성전자 가전부문 생산기술본부, "삼성 파라 스카라 로봇 시스템", KACC, 1991
7. 서울대학교 공과대학 생산기술 연구소, "수평다관절형 조립용 로봇 제어부 설계", 1987