

개방형 로봇제어를 위한 표준기준모델에 관한 연구

A Standard Reference Model for Open Architecture Robot Control System

김호철 (부산대 대학원), 홍금식, 최경현, 이석희 (부산대 기계공학부)

H. C. Kim (Graduate College), K. S. Hong, K. H. Choi, S. H. Lee (Dept. of Mechanical Engr. PNU)

ABSTRACT

The strategy of open architecture control system intends to integrate manufacturing components on a single platform, so that a particular component can be easily added and/or replaced. Therefore, the control scheme is neither hardware dependent nor software dependent. In this paper a modular and object oriented approach for the open architecture structure of control systems is investigated. A standard reference model for genetic manufacturer system, which consists of three modules: hardware module, operating system module, and application software module, is first proposed. Then a standard reference model for open architecture robot control system is suggested.

Key Words: open architecture control(개방구조제어), robot control(로봇제어), object-oriented analysis(객체지향분석), virtual motion controller(가상모션제어기)

1. 서론

현대의 제조업은 제품의 짧은 수명주기(life cycle)와 소비자들의 다양한 요구에 신속하게 대응할 수 있는 유연성(flexibility)과 제조기술의 빠른 진보로 다양한 기능을 공급할 수 있는 제조장비들을 요구하고 있다. 산업용 로봇, NC공작기계등은 다양한 기능을 가진 제조장비이지만 각 제조회사마다 자사의 전용제어시스템을 사용함에 따른 폐쇄구조(closed architecture)로 되어있어 제어시스템의 구성시 상호호환성의 문제로 불필요한 주변장치로 인한 가격상승과 운영자가 각 제조회사의 로봇이나 NC공작기계등에 대하여 전문적인 지식을 알아야 하는 불필요한 노력을 요구하게 된다. 이러한 문제를 해결하기 위하여 표준화가 된 제어기의 개발이 시도되고 있지만, 제조회사들로부터 호응을 얻고 있지 못하는 실정이다. 최근에는 컴퓨터 하드웨어와 소프트웨어 기술의 급격한 발전에 힘입어 폐쇄구조의 제어기로부터 사용자 편의의 독자적인 기능을 구현/해석할 수 있는 새로운 구조의 개방구조(open architecture)의 제어기 개발이 이루어지고 있다. 이미 선진국에서는 개방형 제어시스템에 대한 구체적인 연구와 자체의 표준모델을 제시하고 있다. 미국에서는 GM등 상위 몇 개의 자동차회사가 상호 협력하여 OMAC(Open, Modular Architecture Controllers)이라 불리우는 개방형 모듈구조의 제어기 개발에 관한 연구를 하고 있고, 일본에서는 6개의 대표적인 제조회사들

이 OSEC(Open Systems Environment for Control) 컨소시움을 형성하여 개방형구조에 대한 연구를 하고 있다. 하지만, 이들 대부분의 연구는 NC공작기계에 한정되어 있어 보다 복잡한 제어구조를 가진 산업용 로봇의 개방형 구조에 관한 연구는 미흡한 실정이다. 본 논문에서는 먼저 모든 제조장비의 제조시스템에 범용으로 사용할 수 있는 개방제어시스템의 표준기준모델(Standard Reference Model: SRM)을 객체지향분석(object oriented analysis)을 통하여 제시하고 그리고 이렇게 제시한 표준기준모델을 산업용 로봇제어시스템에 적용하여 개방형 로봇 제어시스템의 구성요소와 구조에 대하여 제안하고자 한다.

2. 개방형제어시스템

(Open Architecture Control System)

개방형 제어시스템이란 자동화 공장에서 사용하고 있는 다양한 제조장비의 제어시스템을 하나의 제어기의 환경으로 통합하여 제어할 수 있는 개방된 제어시스템을 의미한다. 객체지향분석방식에 의한 개방형 제어시스템의 표준기준모델의 구축은 모든 제어시스템의 구성요소를 객체지향분석에 의해 모듈화하여 모듈제어시스템(modular control system)을 구성하는데 있다. 개방형 제어시스템의 구성요소는 객체지향 특징을 만족시키는 모듈들로 이루어 졌고, 아래와 같은 5가지의 특징을 갖고 있다[1].

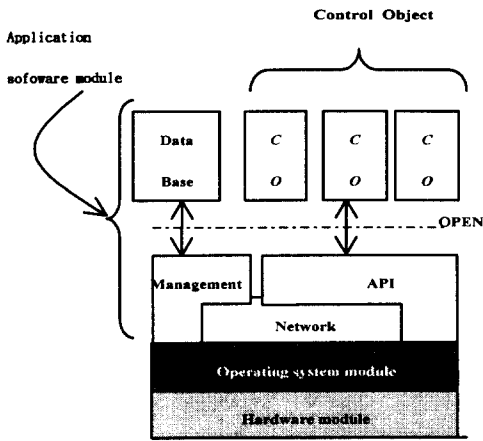


Fig. 1. A Standard Reference Model for Open Architecture Control

- (1) 이식성(portability): 제어시스템의 모듈들이 다른 제어시스템에서도 구동할 수 있는 성질.
- (2) 확장성(expandability): 모듈의 기능을 쉽게 확장할 수 있는 성질.
- (3) 상호 교환성(exchangeability): 모듈이 상호호환성의 기능을 가진 다른 모듈로 대처할 수 있는 성질.
- (4) 가변성(scalability): 성능향상을 만족시키기 위하여 쉽고 효과적인 방법으로 모듈간의 환경재설정이 가능한 성질.
- (5) 상호호환성(inter-operability): 모듈간에 서로 데이터의 교환이 가능한 성질.

3. 개방형시스템의 표준기준모델 (Standard Reference Model for OAC)

개방형 제어시스템의 표준기준모델은 모듈제어시스템의 구조로 형성되었고 하드웨어모듈(hardware module), 운영시스템모듈(operating system module), 응용소프트웨어모듈(application software module)로 구성되어 있다(Fig. 1.).

3.1 하드웨어 모듈

개방구조에서 하드웨어 모듈은 가장 표준화 된 사양을 따르는 구조로 구성되어야 한다. 오늘날 하드웨어가 빠른 속도로 발전하기 때문에 하드웨어 자체에 대한 사양을 표준화하기보다는 상황에 따라 가장 적합하고 가격이 저렴한 하드웨어를 어느 때고 쉽게 사용이 가능하게 하기 위해서는 하드웨어 드라이버의 표준화가 되어야 한다. 오늘날 개방형 제어시스템의 하드웨어 추세는 PC(Personal Computer)를 개

방형 제어시스템의 하드웨어로 대체하여 이용한다. 이러한 이유는 PC가 대량 생산으로 인한 저가격화와 이에 따라 정기적/고장시 교체가 용이하고 CPU의 추가적인 성능향상을 기대할 수 있다.

3.2 운영시스템 모듈

운영시스템 모듈은 하드웨어, 응용 소프트웨어, API (Application Programming Interface)등에 대한 유용성(availability)을 제공해 준다. 한 개의 코프로세서(coprocessor)가 탑재된 PC를 개방형 제어시스템에 포함하여 사용하려면 반드시 운영시스템 모듈이 실시간 운영시스템(Real Time Operating System: RTOS)형식으로 구성되어야 한다. 이러한 RTOS와 Windows/DOS의 중요한 차이는 코프로세서의 속도의 차이보다는 CPU시간을 어떻게 관리하고, 메모리를 어떻게 할당하고, 인터럽트(interrupt)에 어떻게 응답하고, 기계장치를 구동하는 처리과정에서 PC의 자원을 어떻게 공유하는가에 달려 있다.

3.3 응용소프트웨어 모듈

(1) 관리자(Management): 개방제어시스템의 가장 중요한 특징은 환경재설정(reconfiguration)의 기능이 가능한 점이다. 관리자는 이러한 개방형 제어기의 환경재설정의 역할을 한다. 일반적인 개방형 제어시스템의 환경설정과정은 먼저 off-line에서 사용자는 실제의 제어시스템의 기능을 분석한 다음 개방형 표준기준모델에 입력할 변수값을 환경설정편집기(configuration editor)를 이용하여 환경설정주문(configuration order)을 작성한다. 관리자는 시스템이 매번 부팅시 환경설정파일을 읽어 들이고, 환경설정주문에 포함된 변수값을 각종 제조장비의 구성요소에 대하여 체계적으로 분류하여 구성한 data base에서 갖고와서 개방형 제어시스템의 표준기준모델에 입력한다. 이러한 제조환경의 변화에 대한 관리자의 운영(management operation)은 개방형 제어시스템의 구조형성에 기본이 된다(Fig. 2.참조).

(2) API: 제조환경은 생산제품이나 생산공정의 변화에 따라 제조장비의 재구성이나 추가적인 설치를 요구하게 된다. 제조환경의 변화에 따라 제어시스템의 구성요소도 재조정을 요구하게 된다. 이로 인해 제조환경에 이미 설치한 제어시스템을 재사용하는 것이 불가능해진다. 제어시스템의 유연성이나 확장성에 대한 문제는 개방형 제어시스템을 구성하는데 많은 문제점을 주고 있다. 이러한 대처방안으로 하드웨어나 소프트웨어측면에서 기존의 하드웨어나 소프트웨어가 공통된 인터페이스를 형성한다면 모듈제조시스템이 형성된다. 이러한 방법으로 제조시스템의 구성요소에 대하여 범용 API의 모듈들을 시스템 기반(system platform)에 포함시키면, 제어시스템의 모

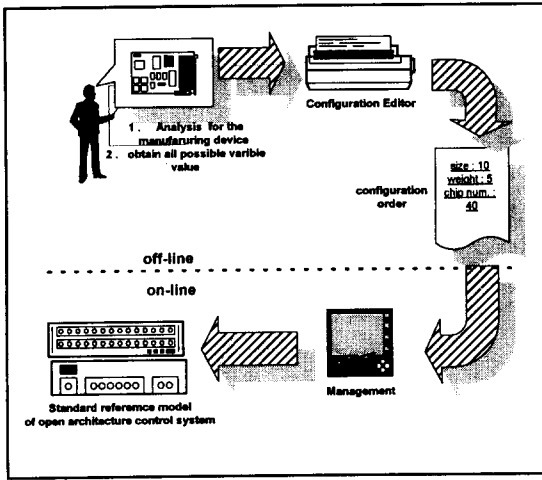


Fig. 2. Reconfiguration Operation

은 구성요소가 "plug and play" 기능으로 재사용이나 추가설치를 쉽게 할 수 있다.

(3) 네트워크 (Network): 네트워크는 객체들의 현재 상태의 값을 읽어 들이거나 필요한 동작에 대한 적절한 값을 내릴 때 data bus의 역할을 한다. 네트워크는 개념적으로 두개의 부분으로 나뉘어 데이터 교환을 한다. 어떠한 기능을 수행하라고 명령하는 클라이언트(client)부분과 이러한 클라이언트의 명령에 따라 요구된 기능을 수행하는 서버(server)부분이다 [2].

(4) 제어객체(Control Object): 제어객체는 제조장치를 실제로 제어하는데 사용되는 구성요소를 소프트웨어측면에서 모듈화하여 객체로 나타낸 것이다. 제어객체에 대한 사용자의 환경설정에 따라 다른 제조장비 제어시스템 환경으로 전환이 가능하며, 여러개의 제조장비 제어시스템을 동시에 구성이 가능하다. 제어객체는 개방구조를 가지고 있으므로 사용자 정의 개발들도 정의할 수 있다.

4. 개방형 로봇제어시스템

일반적인 로봇제어시스템의 구조는 Fig. 3에서 보여지는 바와 같이 매니플레이터, 연결케이블, 모션컨트롤러(motion controller), 작업제어(task controls), 외부센서(external sensor)로 이루어 졌다[3]. 이러한 5가지 구성요소 중에서 로봇 종류에 관계없이 일정한 구성요소로 모션컨트롤러, 태스크제어, 외부센서로 3가지 부분으로 나눌 수 있다. 이러한 일반적인 로봇제어시스템의 3가지의 구성요소를 앞에서 제시한 개방형 제어시스템의 표준기준모델의 제어객체로 정의하여 개방형 로봇제어시스템을 구축할

수 있다. 이러한 개방형 구조 로봇제어시스템의 제어객체를 각각 가상모션제어기객체(Virtual Motion Controller Object)와 태스크제어객체(task control object)와 센서장치객체(sensory device object)로 정의할 수 있다(Fig. 4참조).

4.1 가상모션제어기 객체

VMC객체는 실제로 존재하는 물리적 객체가 아니라 가상적인 객체이다. 가상적 객체는 명령을 내리

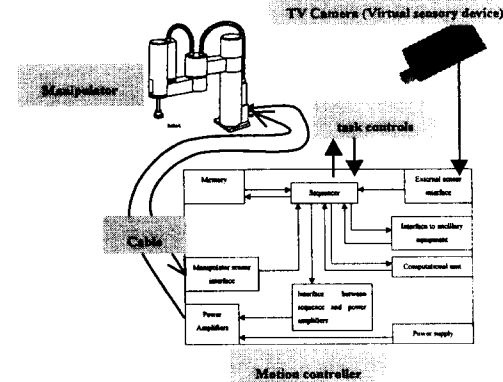


Fig.3. Diagram of conventional robot control system

는 클라이언트의 입장에서 봤을 때 어떤 명령을 내릴 때 이에 대응하여 어떤 행동을 나타내는 객체를 의미한다[2]. VMC는 정해진 형식으로 구성된 메시지에 응답하는 소프트웨어로 논리적 객체로 동일시되고, 다른 객체가 요구한 행동을 한다. 네트워크상에서 일반 제조장비에 포함된 물리적 로봇모션컨트롤러의 기능들을 VMD로 사상한 것이라고 볼 수 있다. 예를 들면, 전형적인 로봇제어시스템에서 지정된 두 위치사이를 로봇에게 움직이라고 하는 작업을 수행할 때 기존의 로봇제어시스템은 정해진 방법에 따라 로봇컨트롤러에 일련의 원시적인 신호로 보내지만, 개방형 로봇제어시스템에서는 네트워크상으로 계산된 두 위치사이의 좌표변환의 데이터를 VMC로 보내고 VMC는 일련의 신호로 서버드라이브(servo driver)로 보낸다. VMC는 개방형 제어시스템에서 기계장치를 제어하는 주된 객체이기 때문에 만약 제어시스템이 VMC를 가지고 있지 않다면, 실제 물리적인 매니플레이터는 네트워크상에서 서버로서 작용할 수 없다.

4.2 태스크제어 객체

태스크제어 객체는 매니플레이터의 완전한 태스크를 생성하는 태스크생성객체(task generation object)와 발생되는 오류(error)를 감지 및 처리하는 역할을 하는 오류조종객체(error handling object)로 구성되어 있다. 경로계획, 경로생성의 작업은 매니플레이

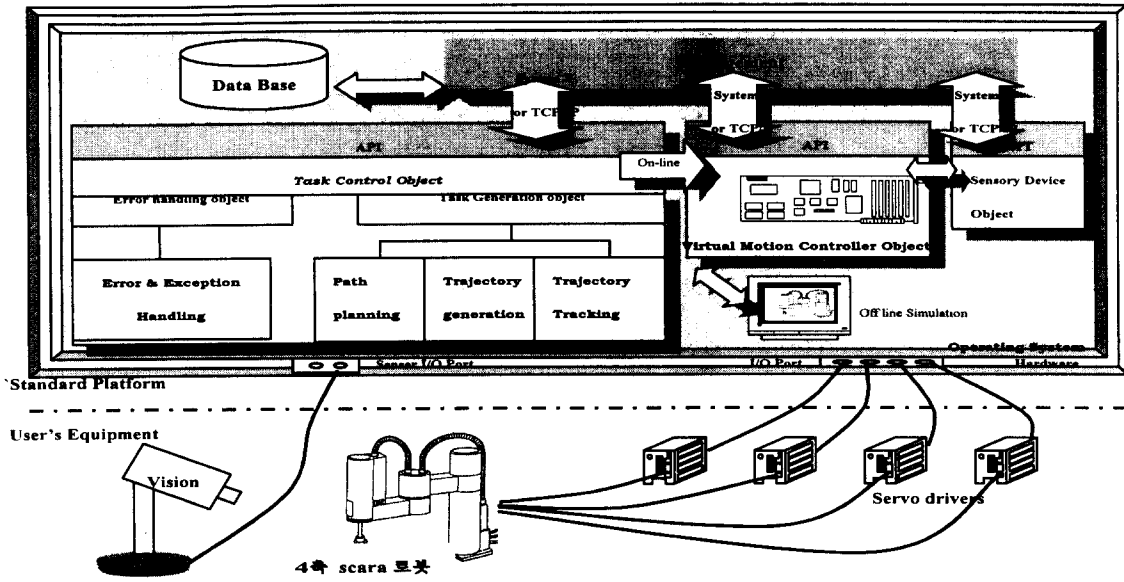


Fig. 4. Diagram for Open Architecture Robot Controller

터의 동역학적 사항을 고려하여 만들어진 제어 알고리즘을 바탕으로 운영자의 의도대로 매니플레이터의 경로를 계획 및 생성하는 작업이고, 궤적추적작업은 만들어진 궤적에 대하여 실제 매니플레이터의 동작을 명령하는 작업을 한다. 이러한 궤적추적작업은 서버루프(servo loop)에서 엔코더 출력 값이나 다른 여러 장치들의 입출력장치의 데이터가 필요하다. 이러한 데이터는 반드시 실시간에 근거해서 입출력해야 하므로 실시간으로 작동하며, 데이터를 다룰 수 있는 기능을 테스트생성 객체에 포함시켜야 한다. 오류조종객체는 전체 테스트제어 작업시 발생하는 오류를 감지하여 발생한 오류의 정확한 원인과 이에 대한 대비책을 제시하고 해결하는 작업을 한다.

4.3 센서장치객체(Sensory Device Object)

객체는 외부환경으로부터 다양한 센서를 통하여 정보를 모으는 수단을 제공한다. 그리고 매니플레이터가 다음 행동에 대한 결정 및 작동을 할 때 로봇 구성요소에 대한 상태에 대한 정보를 공급한다[4].

5. 결론

본 논문에서는 자동화공장에서 사용되고 있는 여러 제조장비들의 제어시스템을 하나의 제어기로 통합이 가능한 개방형 제어시스템의 표준기준모델을 객체지향방식을 이용하여 제시하였다. 객체지향방식으로 구성된 표준기준모델은 모듈제어시스템 구조로 구성되어 각각의 사용자가 원하는 제조장치 제어시스템의 환경에 따라 재환경이 가능하다. 그리고, 제시된 표준기준모델을 로봇제어시스템에 적용하여 개

방형 로봇제어시스템의 표준기준모델의 구조 및 구성요소를 제안하였다.

참고문헌

1. W. Sperling and P. Lulz, "Enabling Open Control Systems: An Introduction to the OSACA System Platform." ESPRIT III Project: Stuttgart: FISW GmbH, 1995.
2. J.P.T. Mo and Y. Wang and C.K. Tang, "The Use of the Virtual Manufacturing Device in the Manufacturing Message Specification Protocol for Robot Task Control", Computers in Industry vol.28, 123-136, 1996.
3. R.D. Klafter and T.A. Chmielewski and M. Negin, "Robotic Engineering an Itegrated Approach.", Prentice-Hall, Englewood Cliffs, NJ, 1989.
4. LP Elektronik GmbH, Real time though PC, Intenet document at [http:// www.ip -elektronik .com/omac.html](http://www.ip-elektronik.com/omac.html)
5. J. Rumbaugh and M. Blaha and W. Premerlani and F. Eddy and W. Lorensen, " Object-Orented Modeling and Design.", Prentice-Hall, Englewood Cliffs, NJ, 1991.