

범용의 PC를 이용한 로봇 제어기 구성에 관한 연구  
(A Design of PC-based Robot Controller)

○ 정재문, 양윤모, 김선일  
(J.M. Chung, Y.M. Yang, S.I. Kim)

한국기계연구소 자동제어실  
( KIMM, Automatic Control Lab. )

Generally, Industrial robots are often controlled using joint processors and treating each joint as an independent servo loop. This paper presents a system architecture for robot control designed for real-time control of motion and sensory processing utilizing general-purpose Personal Computer. And for easily use and system expandability, robot language is implemented with C-language as base language. Through this system user can easily update robot language by design of his own language primitives. This system also don't require another development tool and can be used as advanced algorithm simulator in robotics laboratories.

1. 서론

일반적으로 로봇 제어기의 구성은 로봇 몸체의 각종 구동을 위한 모터 및 드라이브로 구성되는 구동부, 각종 구동을 제어하는 축제어기, 그리고 축들을 전체적으로 제어하는 주제어기로 이루어진다. 이외에도 로봇 프로그래밍 장치, 그리고 로봇의 지능화가 이루어져 감에 따라 화상 처리기능을 비롯한 고급 센서 처리장치가 추가된다.

본 연구에서는 로봇 제어기를 구성하는데 있어서 최근 광범위하게 사용되어지고 있는 범용의 개인용 컴퓨터(Personal Computer)의 기능을 최대한 이용함으로써 시스템 구성에 따르는 하드웨어 및 소프트웨어의 부담을 최소화하려고 하였다. 그리고 로봇 언어의 구성에 있어서도 로봇 언어를 위한 인터프리터(Interpreter)를 따로 구성할 필요가 없이 베이스 언어를 C-언어로 사용하였다.

로봇 언어를 컴퓨터의 언어와 비교할때 로봇에만 필요한 프리미티브(Primitive)가 전체의 20% 이하임을 고려할때 사용자 입장에서 특별히 새로운 학습이 필요 없는 일반 컴퓨터 언어인 C-언어의 사용은 타당성이 있다고 하겠다.

그렇게 함으로서 사용자는 자기가 특별히 필요한 새로운 기능의 프리미티브를 컴퓨터 라이브러리(Computer Library) 형태로 작성하여 추가함으로써 충분한 언어의 확장성을 추구할 수가 있다.

컴퓨터를 기반으로하는 제어기에 위의 언어프로그래밍 장치 외에 각축제어기, 주제어기의 기능을 구현하며 고급센서의 처리를 위한 하드웨어 및 소프트웨어를 쉽게 수용할 수 있어 시스템의 확장에 유리하며 경제성에서도 강점이 있다고 하겠다. 제어기와 구동부와의 인터페이스(Interfacig) 위한 한장의 보드를 설계하였다. 이 보드를 통하여 주제어기와 구동부를 포토 아이소레이션(Photo-Isolation)에 의해 접속 시킴으로서 시스템의 안정성을 추구하고있다. 각축 제어를 위한 축제어기가 주제어기 상에서 소프트웨어로 이루어지고 있는데 고속의 축구동제어가 필요하다면 인터페이스 보드(Interface Board)의 기능을 강화하므로써 축제어기의 기능을 고속화시킬 수 있다.

2. 시스템 구성

시스템의 구성은 (그림1)에 나타나있다.

시스템의 주제어기(Main Controller)는 IBM-PC/AT 호환 기종을 사용하고 로봇 제어를 위한 인터페이스 보드를 이 컴퓨터 버스(Computer Bus) 베이스로 설계하였다. 주제어기와 인터페이스 보드 사이의 통신은 I/O 포트(Port) 방식을 사용하였으며 인터페이스 보드와 각축 제어 모듈과는 포토 아이소레이션(Photo Isolation)을 통한 직렬 펄스트레인(Serial Pulse Train)방식을 채용함으로써 접속선의 가다수를 줄이면서 신뢰성을 높였다. 그리고 컴퓨터 버스를 통하

여 이미지 처리 보드 등의 센서 보드를 수용하여 시스템의 확장을 쉽게할수있게 하였다. (그림2)에는 시스템의 기능적 부분들과 이들간의 관계가 나타나 있다. 주제어 모듈은 시스템 내의 다른 모든 모듈들을 관장하며 시스템 구동이 시작되면 이 모듈이 동작하여 시스템의 각부분이 수행된다.

를 처리함으로써 로봇의 지능화를 가능하게 하여준다. 페스 플래너(Path Planner)는 랭귀지 프로세서와 센서 프로세서 등에 의해 생성된 동작에 관련된 코드를 가지고 이것들을 만족시키는 정확한 시간, 위치, 속도 그리고 가속도의 완전한 페스 사양을 생성해낸다. 여기서 생성된 직각좌표계에서

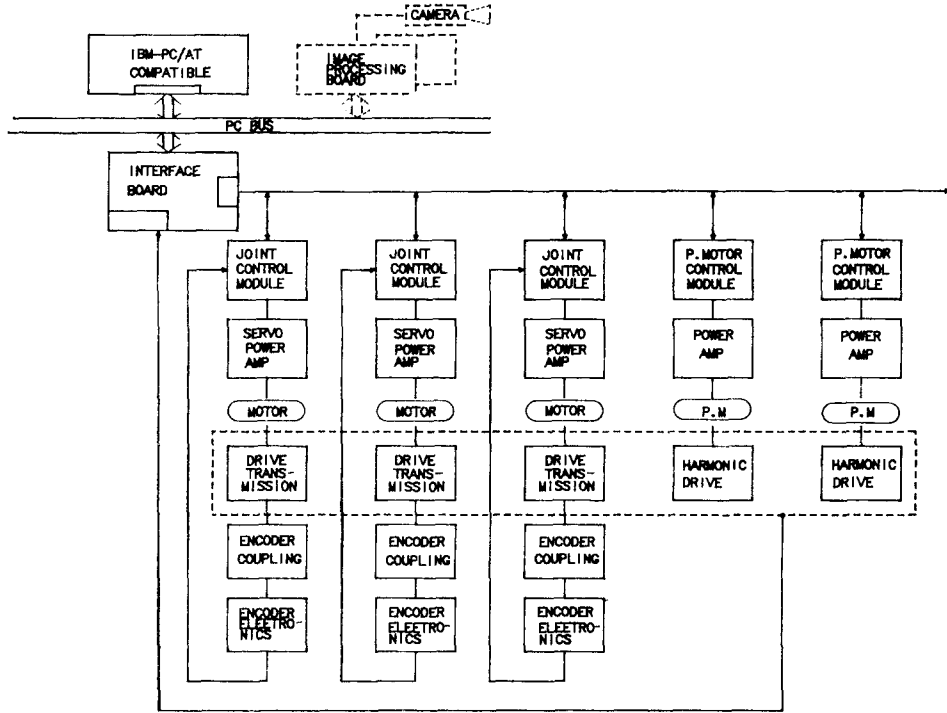


그림 1. 시스템 구성도

에디터(Editor)는 컴퓨터에 제공되는 각종의 에디터 관련 패키지를 사용하며 사용자가 키보드를 통한 파일의 편집을 가능하게 해준다. 파일핸들러(File Handler)는 MS DOS를 기반으로 파일의 처리를 수행하게 한다. 랭귀지 프로세서는 사용자 프로그램의 수행시 소스 프로그램(Source Program)을 내부명령어로 번역하여 수행하게 하는데 이중 랭귀지 인터프리트는 별도의 구성없이 PC에 내재하는 C 언어 인터프리트를 그대로 사용할수 있게 함으로서 사용및 구성이 용이하게 하였다. 조그 오퍼레이션(Jog Operation) 모듈은 키보드를 통한 각축의 매뉴얼 제어(Manual Control)를 통한 위치 학습(Position Teach)을 가능하게 해준다. 센서 프로세서(Sensor Processor)는 이미지(Image) 등의 고급의 센서정보

의 원하는 위치와 방향에 관한 벡터를 각축의 축위치에 관한 데이터로 변환시키는 기능은 키네매틱스(Kinematics) 모듈에서 이루어진다. 각축 제어 펄스 생성에는 각축에 할당된 위치, 속도, 가속도에 관한 명령이 정확한 시간에 이루어질 수 있도록 합당한 펄스 열(Pulse Train)을 생성시켜서 서보모듈에 보내주는 역할을한다. 각축의 서보 모듈은 (그림1)의 각축 제어모듈(Joint Control module)이하의 모듈들로 구성된다. (그림3)은 제어시스템의 소프트웨어의 블록 다이어그램이다.

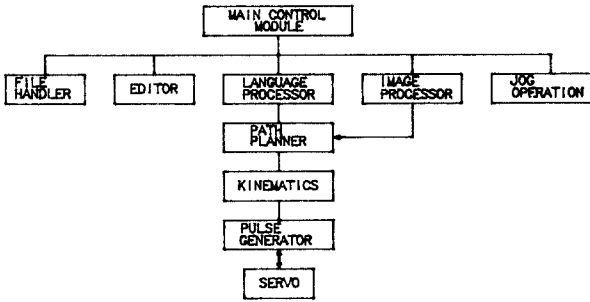


그림 2. 시스템 모듈 구성

시스템 소프트웨어는 기본적으로 컴퓨터 내의 MS DOS를 기반으로 하고 그의 필요한 여러 종류의 소프트웨어 패키지를 활용하는 방식으로 시스템 에디터(Editor), 디버거(Debugger), 컴파일러(Compiler), 어셈블러(Assembler), 링커(Linker) 및 각종의 시스템 라이브러리(System Library)등을 구성하였다.

### 3. 축 제어 (Axis Control)

이 절에서는 모든 로봇 운동의 기본이 되는 각 축의 포인트-포인트(Point-to-Point) 가감속 제어를 나타낸다.

본 시스템에서의 로봇 기계부의 사양을 살펴보면,

- 한축의 최고속도 50 mm/sec
- Ball Screw Pitch 5 mm
- Encoder Resolution 1000:1
- > 요구되는 최고 Pulse Rate 10000 Hz
- 본 컴퓨터 Clock Frequency 8 MHz
- > 1 Pulse 생성시 요구되는 최대허용 Clock 수 800 clocks

이상에서 보는바와 같이 800 clock 이내에서 1 pulse의 가감속 제어된 신호를 발생시킬 수 있다면 별도의 펄스발생을 위한 하드웨어의 구성없이 주제어기(Main Controller) 상에서 알고리즘의 수행에 의해 구동이 가능하다.

(그림4)은 축제어를 위한 전체 블록 다이어그램이다.

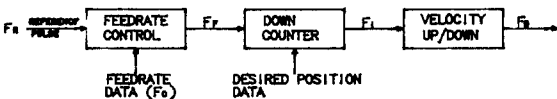


그림 4. 축제어 블록다이어그램

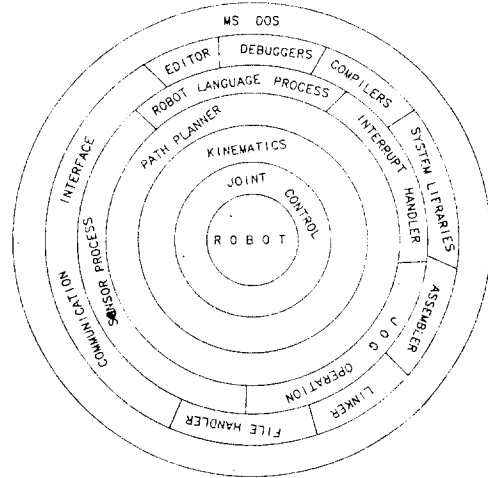


그림 3. 제어시스템 소프트웨어 구조

기준펄스(Fr)은 일정한 시차를 두고 발생되도록 딜레이(Delay)를 적절히 사용하여야 한다.

(그림5)은 속도제어부이다.

$$X(k+1)=X(k)+F_c, X(0)=0$$

$$X(k)=k \cdot F_c$$

\* Ff는 Fc에 비례하는 Pulse Rate를 가진다.

이때 Down Counter는 미리 전체이동거리에 해당하는 값으로 설정되어 있는데 Ff 펄스를 받을때 마다 값을 1씩 감소시켜 나가서 0이 되면 더이상 펄스를 통과시키지 않는다.

(그림6)은 가감속부이다.

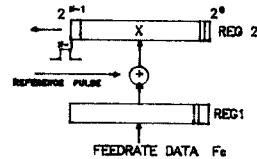


그림 5. 속도제어부

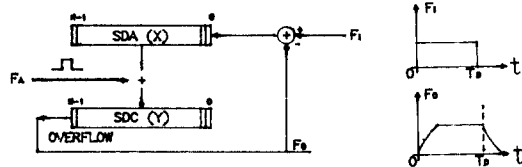


그림 6. 가감속부

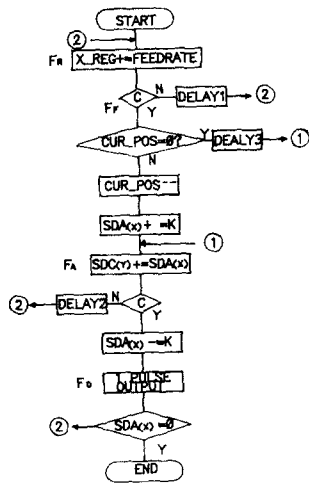


그림 7. 제어 Flow Chart

Fi가 도착할 때마다 SDA를 1증가시키고 Fa가 감지될 때마다 SDA 레지스터를 SDC에 합한다. 이때 SDC에서 Overflow가 발생하면 SDA를 1 감소시킨다.

$$dX = (F_i - F_o) \cdot dt \quad (1)$$

$$x(0) = y(0) = 0$$

$$dY = F_a \cdot X \cdot dt \quad (2)$$

$$F_o \cdot dt = dY / 2 \quad (3)$$

(1), (2), (3)에서

$$F_o = Fi [1 - e^{-(Fa \cdot t / 2)}], \quad (0 \leq t \leq td)$$

Down Counter가 0이 되는 순간( $t=td$ ) Fi는 0이되므로 감속시의 Fo는

$$F_o = F_o(td) \cdot \exp[-Fa \cdot (t - td) / 2], \quad (td < t)$$

이상의 알고리즘을 소프트웨어로 실행하기 위한 Flow Chart는 (그림7)와 같다.

#### 4. 로봇 언어

본 시스템의 주 제어기점 프로그래밍 장치로 범용의 개인용 컴퓨터가 사용되었으며, 주로 로봇 언어용 인터프리터 (Interpreter) 작성을 하지 않고 C-언어를 그대로 사용하는 RCCL (Robot Control 'C' Library) 방식을 채택하였으며, 그 구성은 아래와 같으며 언어처리 순서도는 (그림8)와 같다.

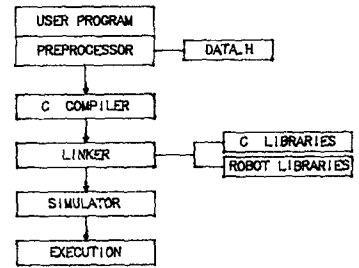


그림 8. 언어처리 순서도

- RCCL 'C' : C-언어
- RCCL 'Macro':C-library
- 추가된 Robot 관련 Primitives
- CRT 상의 Interactive 'menu'
- Robot 위치 data 관련하여 추가된 data type

#### 5. 결론

본 시스템은 X, Y, Z의 직각좌표형 3축을 베이스로 한 a, b 2축의 회전축을 가진 5축 로봇을 위한 제어기로 비교적 대형물체의 계측을 위해 설계되었다. 이 시스템에서는 특별히 높은 위치 정밀도를 요구하는 반면 속도는 빠를 필요가 없었으므로 모터의 Servo Loop이하 이외의 모든 부분이 개인용 컴퓨터 상에서 이루어 질수 있었다. 고속의 운동을 위한 인터페이스 보드의 보강과 필요한 소프트웨어의 구성으로 거의 모든 종류의 로봇을 위한 제어기로 이용될수 있을 것이다.

#### 참고문헌

- 여인택, 노태석, 이봉진, "마이크로 프로세서를 이용한 인터플레이터", 전기학회논문지 제33권 제2호 1984.2.
- T.L.Turner "A microprocessor architecture for advanced robot control", 14th ISIR, Oct. 1984
- Vincent Hayward, Richard P. Paul "Introduction to RCCL: A Robot Control 'C' library", IEEE Computer Society, International Conference on Robotics, 1984.