'91 韓國自動制御學術會議論文集 1991. 10.22~24

32 비트 DSP를 사용한 로보트 제어기의 개발

김 성 권, 황 찬 영, °전 병 환, 이 규 철, 홍 용 준

삼성전자 생산기술본부

Robot Controller with 32-bit DSP Chip

Sung-Kwun Kim, Chan-Yung Hwang, Byeong-Hwan Jeon, Kyu-Cheol Lee, Yong-Juhn Hong

Production Engineering Division

Samsung Electronics

ABSTRACT

A new 6-axis robot controller with a high-speed 32-bit floating-point DSP TMS320C30 has been developed in Samsung Electronics. The controller composed of Intel 80386 microprocessor for the main controller, and TMS320C30 DSP chip for joint position controller. The characteristics of the controller are high sampling rate of 200us and fast reponsibilty. The main controller supports MS-DOS, kinematics, trajectory planning, and sensor fusion functions which are vision, PLC, and MAP. The one high speed DSP chip is used for controlling 6 axes of a robot in 200us simultaneously. The control law applied is PID controller including a velocity feedforward in joint position controller. The performance tests, such as command following, CP, were conducted for the controller integrated with a 6 axes robot developed in Samsung Electronics. The results showed a good performance. This controller can also perform the system control with other controllers, the communication with high priority controllers, and visual information processing.

1. 서론

오늘날 산업계에서는 생산력 증대와 제품의 품질 향상이라는 두 문제를 해결하기 위하여 생산 자동화에 관심이 집중되고 있고, 이를 실현하기 위하여 자동화의 핵심이 되는 로보트 개발 및 응용에 대한 노력이 지속적으로 진행되고 있다. 로보트를 이용한 자동화 시스템은 대량 생산 체제에서 뿐만 아니라 소량 다품종 생산에 절대적으로 필요한 유연 생산 체제(FMS, Flexible Manifacturing System)의 구현에 유리하며 사람의 노동력을 대체함으로써 단순 반복 작업이나 고난도 가공 및 조립에서 발생할 수 있는 제품 불량을 감소시킬 수 있다[1].

로보트를 사용한 생산 시스템의 보급이 확산됨에 따라 일반 민생용 제품의 경향과 같이 저가격, 고성능, 고기능, 조작의 편리성등이 상용 로보트에서도 요구되고 있다. 그러나, 지금까지 개발된 여러 종류의 로보트 제어기를 살펴보면 여전히 하드웨어와 소프트웨어의 구조가 복잡하고, 제어기를 개발함에 있어서 장비, 인력, 비용등이 많이 소요되고 있는데 비해 친숙한 사용자 환경을 갖추지 못하고 있다.

본 논문에서 개발한 6축 로보트 제어기는 32 비트 마이크로 프로세서와 32 비트 부동 소숫점 연산용 디지틀 신호 처리 프로세서(DSP)를 사용하여 로보트의 제어 성능을 향상시켰다. 또한, ISA 버스에 의거한 하부 제어기들을 개발함으로써 하드웨어 및 소프트웨어 개발이 용이하였으며 제어 시스템의 기능 확장 및 각 하부 제어기들의 통합이 간단하게 이루어지록 하였다. 그리고, 제어 시스템의 운영 체제로서 현재 널리 보급되어 있는 MS-DOS를 사용함으로써 편리한 사용자 환경이 이루어지도록 하였다.

본 논문은 삼성 전자에서 개발한 6축 제어기의 설계 개념, 제어기의 하드웨어 구성, 제어 구조, 궤적 발생 기법을 설명한 후 개발한 제어기와 로보트 몸체를 동작 시켜 얻은 실험 데이타에 대해 논한 후 결론을 맺는다.

2. 설계 개념

로보트 제어 시스템에서 컴퓨터 또는 계산용 소자의 사용은 지능이 있는 동물에게서 뇌의 존재와 같이 필수적이다. 제어기에서의 계산용 소자는 외부 세계와의 통신, 제어기에의 프로그램의 입력, 제어 시스템의 제어 및 감시,로보트 동작의 궤적 계획 등을 수행한다. 따라서, 로보트 제어기에서 핵심이 되는 컴퓨터 시스템의 구조는 전체 제어기 시스템이 보다 향상된 성능과 기능을 갖도록 잘 설계될 필요가 있다[2]. 본 장에서는 본 논문에서 개발한 6축 로보트 제어기에서 채택한 설계 개념과 계산 소자들 간의 역할 분담 등을 소개하고자 한다.

본 논문에서 개발한 6축 로보트 제어기의의 설계 목표는 인공 지능형 로보트 구현을 위한 고성능 제어기 개발에 있다. 이 목표를 구현하기 위한 방법으로서 다중 프로세서에 의한 분산 처리에 기반을 두어, 주 제어기, 관절 위치 제어기(JPC), 모터 제어기로 구성되는 3 단계 계층적 역할 분담에 의한 제어를 시도하였다.

주제어기의 컴퓨터 운영 체제는 파일, 활용 프로그램, 주변 장치, 외부와의 통신, 디버깅 도구 등의 운영에 대한 세부적인 일들을 수행하므로 소프트웨어 개발에 있어서 매우 중요한 역할을 담당한다. 최근 보편적으로 사용되고 있는 상용 컴퓨터 운영 체제는 MS-DOS와 UNIX인데 MS-DOS는 IBM PC(또는 그 호환기종)에 널리 사용되고 있는 OS의 표준으로서 다양한 종류의 컴퓨터 언어, 응용 프로그램을 사용할 수 있으며 UNIX에 비해 값이 싸고 운용에 소요되는 메모리 용량도 적은 장점이 있다. 한편 IBM PC는 일반 사무용, 각종 제어 시스템의 개발용으로 널리 보급되어 있을 뿐만 아니라 IBM PC/AT의 표준 버스인 ISA 버스와 인터페이스하는 다양한 종류의 내장 보드가 개발되어 있으므로 VME 버스, EISA 버스, Multibus 등에 비해 소프트웨어 및 하드웨어 개발이 용이하고, 널리 보급된 IBM PC/AT를 개발 장비로 사용할 수 있으므로 각종 제어 시스템에서 주 제어기로의 사용이 확산되고 있다. 본 논문에서는 주 제어기에 IBM PC/AT를 사용하고 DOS 커널과 내장된 ROM BIOS 커널을 사용하여 소프트웨어를 개발하였으며, ISA 버스에 인터페이스하는 하부 제어기를 개발하였다. 이미 개발된 관절 위치 제어기 뿐만 아니라 현재 개발되고 있는 시각 정보 처리기, 제어기 내장용 PLC(Programmable Logic Control), 공장 자동화용 통신 프로토콜(MAP)등을 모두 ISA 버스와 인터페이스 함으로써 로보트 지능의 향상, 로보트 주변 시스템과의 통합적인 제어, 궁극의 CIM(Computer Integrated Manufacturing) 구현을 시도하고 있다.

2축 또는 4축 로보트 제어기에 비해 6축 로보트 제어기의 정/역 기구학은 훨씬 복잡하므로 직선/원호 CP(Continuous Path) 동작시 많은 계산량을 필요로 하며, 로보트 제어 성능을 향상시키기 위하여 궤적 계획이나 제어 알고

리듬이 정교해짐에 따라 이에 소요되는 계산량도 큰 비중을 차지한다. 반면, 디지틀 제어기가 아나로그 제어기에 가까운 동특성을 유지하기 위해서는 실시간(real-time) 동작이 요구되므로 고속의 연산 속도가 필요하다. 본 논문에서는 고속의 계산을 위하여 8, 16 비트 프로세서 보다 수행 속도가 빠른 MC68020/30과 i80386 등과 같은 32 비트 프로세서 중 응용 소프트웨어가 풍부하고 개발 장비의 비용이 비교적 저렴한 i80386 마이크로 프로세서와 i80387의 부동 소숫점 연산 전용 보조 프로세서를 주 제어기의 CPU로 사용하였다. 또한, TMS320C30 32 비트 부동 소숫점 DSP(Digtal Signal Processor)를 관절 위치 제어기의 CPU로 사용하는 등 다중 프로세서를 사용한 분산 처리 기법을 로보트 제어기에 적용하였다. 한편, 종래의 다축 로보트 제어기의 관절 위치 제어기는 그림 1의 (a)와 같이 각 축에 1개씩의 프로세서가 존재하여 제어기 구조가 복잡하였으나 본 논문에서는 고속의 DSP인 TMS320C30 1개로 6축 제어를 시분할 방식에 의해 제어함으로써 그림 1의 (b)와 같이 하드웨어 및 소프트웨어 구조를 대폭 간소화하였다. 그런데, 관절 위치 제어기에 최신의 32 비트 고속 DSP인 TMS320C30을 사용함으로써 원가 상승의 측면이 있으나 향후 가격은 하향 조정될 예정이고, 제어 알고리듬의 향상을 꾀할 때 계산량이 다소 증가되더라도 하드웨어의 변경 없이 소프트웨어의 수정만으로 대처가 가능하므로 가격 대비 성능이 뛰어나다.

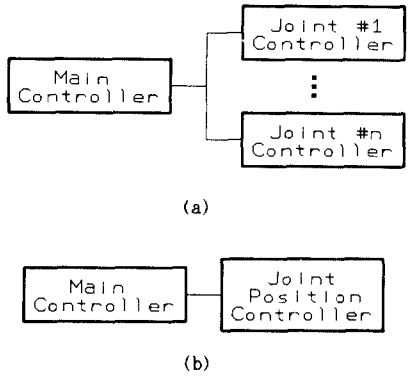

그림 1. (a) 종전의 제어기 구조
(b) 본 논문에서의 제어기 구조
Fig 1. (a) Traditional Controller Structure
(b) The controller structure in this paper

본 논문에서 개발한 로보트 제어기는 BLDC 모터를 구동 대상으로 하였다. BLDC(Brushless DC) 모터는 일반 DC 모터에 비해 출력이 높으며 EMI(Electo-Magnetic Inference)를 발생시키지 않는다. 또한 장기간 사용에 의한 브러쉬의 교환이 필요없으므로 유지 보수의 필요가 없으며 토크/속도 특성이 뛰어나기 때문에 최근 로보트 몸체의 구동 장치에의 사용이 확산되고 있다. 모터의 속도/위치 검출 센서로는 엔코더(encoder), 레졸바(resolver)가 널리 사용되고 있는데 레졸바 방식은 외부 노이즈에 강하고 RDC(Resolver-to-Digital Converter)에 의해 그 해상도를 가변시킬 수 있으므로 엔코더 방식에 의해 유연도가 뛰어나므로 본 논문에서 개발한 제어기의 위치/속도 검출 센서로서 레졸바를 채택하였다.

3. 시스템 하드웨어 구조

본 논문에서 개발한 6축 로보트 제어기는 그림 2의 개략도에 나타낸 바와 같이 IBM AT/386 컴퓨터를 주 제어기로 사용하고 ISA(Industry Standard Architecture) 버스에 각종 하부 제어기들을 인터페이스하여 통합된 제어 시스템을 이루었다. 주 제어기의 제어 환경으로는 키보드, 모니터 등의 기본적인 데이타 입출력 장치, 플로피 디스크 드라이브(FDD), 하드 디스크 드라이브(HDD)등의 보조 기억장치로 구성되며, 주 제어기에 접속되는 하부 제어기로서 I/O 제어기와 위치 제어기가 있다. I/O 제어기에는 원격 교시 장치(teach pendant)와 I/O 보드가 접속되며, 위치 제어기에는 모터 제어기가 접속된다. 그리고, PLC, MAP, Vision등은 현재 개발 중에 있다.

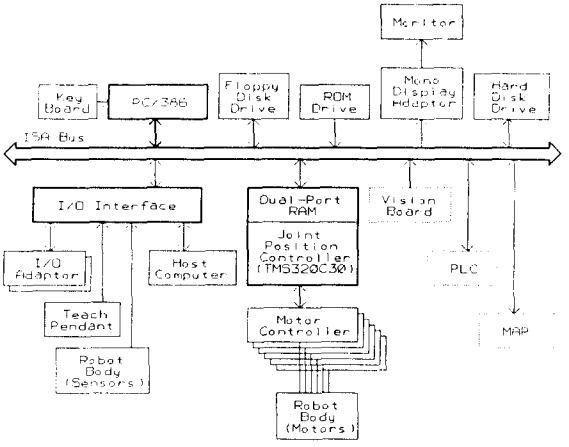

그림 2. 로보트 제어기의 개략도
Fig 2. Block diagram of the robot controller

주 제어기인 IBM PC/386 컴퓨터 보드는 INTEL사의 25MHz 32 비트 i80386 마이크로 프로세서와 i80387 수치연산 전용 보조 프로세서를 CPU로 이용하고 64K 바이트의 캐쉬 메모리를 사용하므로 고속 연산이 가능하며, 2개의 RS232-C 직렬 통신용 포트와 1개의 병렬 통신용 포트를 내장하고 있다. 대화식 프로그램 편집과 제어 명령어 입력을 위하여 주 제어기는 키보드와 모노크롬 그래픽 표시 장치(MDA)를 사용하였으며 3.5" 1.44MB FDD를 장착하여 프로그램 및 위치 데이타의 보관과 제어기간 상호 이동이 용이하게 하였다. 특히 주 제어기에 512K 바이트 용량의 ROM 디스크 드라이브를 장착하여 제어기의 운영 프로그램 및 각종 편의용 프로그램들을 ROM에 기입함으로써 프로그램 보관의 안정도를 높혔다. HDD는 로보트 제어기가 시각 검사 보드, 프로그래머블 로직 콘트롤러(PLC) 보드, 공장 자동화용 통신(MAP) 보드등의 운용 프로그램과 이에 필요한 각종 데이타를 보관하기 위함이다.

표 1. 주 제어기의 구성
Tab 1. Configuration of main controller

항 목	사 양
CPU	i80386 + i80387
FDD	3.5" 1.44MB
HDD	20MB
ROM drive	512K
Communication	2 Serial, 1 Parallel

I/O 제어기는 주 제어기의 제어를 받으면서 전체 시스템을 관장할 수 있는 몇 가지 기능을 갖고 있다. 첫째, I/O 제어기는 로보트 제어기에 부속된 각종 스위치와 램프를 구동하며, 입력 64점, 출력 64점 입출력점을 제어할 수 있는 I/O 보드들을 제어하여 최대 512개의 입력점, 512개의 출력점을 제어할 수 있다. 둘째, RS232-C 통신 포트보다 긴 전송 거리와 높은 전송률을 갖고 있는 RS422 직렬 포트를 내장하여 네트워크 관리용 호스트 컴퓨터와 통신할 수 있도록하고 있으며 RS232-C 직렬 포트를 사용하여 원격

교시 장치가 접속될 수 있도록 하였다. 셋째, 전기적으로 소거 가능하고 불휘발성 메모리인 EEPROM을 사용하여 시스템 파라메터, 사용자 가변 변수등을 보관하는 역할을 한다. 넷째, 로보트 몸체에 내장된 각종 센서를 구동할 뿐만 아니라 그 센서의 변화를 감지함으로써 이상 상태 발생시 주 제어기가 적절한 조치를 취할 수 있도록 하였다. 다섯째, 주 제어기의 궤적 발생기의 출력인 위치 명령이 16msec 마다 위치 제어기에 전달되도록, 그리고 위치 제어기의 출력인 속도 명령이 1msec 마다 모터 제어기에 전달되도록 16msec 및 1msec 타이밍 발생기를 내장하여 제어 시스템의 동기를 일치시켰다.

표 2. I/O 제어기의 구성
Tab 2. Configuration of I/O controller

항 목	사 양
입력점 수	128(최대 512)
출력점 수	128(최대 512)
직렬 포트	1 RS422, 1 RS232-C
EEPROM	2K
센서 입력	18점
타이머	3 채널

관절 위치 제어기(Joint Position Controller)는 주 제어기로 부터 위치 명령을 받아서 모터의 위치가 목표 위치를 추종하도록 제어한다. JPC의 CPU로서 Texas Instrument 사의 33MHz 32 비트 부동 소숫점 연산용 디지틀 신호 처리 프로세서인 TMS320C30을 사용하였다. 제어 알고리듬은 PID(Proportional Integral Differential) 제어에 속도 피드 포워드를 부가한 것으로서 디지틀 제어 방식이다. 제어 출력의 디지틀 값은 12 비트 DAC(Digital-to-Analog Converter)를 통하여 아나로그 신호로 변환되어 모터 드라이버의 속도 명령으로 입력된다. JPC로 피드백되는 위치 데이타는 2상(phase) 12 비트 해상도를 가진 엔코더 신호의 형태로서, 모터 제어기에서 출력되는 것을 사용하며 24 비트 카운터를 사용하여 현재 위치가 계산된다. 24 비트 위치 카운터를 사용함으로써 위치 제어 해상도는 0.0879° 이며 최대 4096 회전을 하드웨어적으로 계산할 수 있다. JPC의 프로그램 메모리는 8K X 32 비트의 ROM을 사용하며 10K X 32 비트의 프로그램 또는 데이타를 보관할 수 있는 RAM을 부가적으로 사용한다. 주 제어기와의 통신은 Dual-Port RAM을 사용하여 구현하며 JPC와 주 제어기간에 인터럽트를 주고 받음으로써 데이타 교환에서의 시간 손실을 최소한으로 줄였다. JPC는 24 비트 출력 포트와 24 비트 입력 포트를 사용하여 모터 제어기의 상태를 제어하거나 모터 제어기의 이상 상태를 감시한다. JPC는 1개의 TMS320C30으로 전체 6축을 동시에 제어하므로 JPC는 6개의 위치 카운터, DAC를 갖는다.

표 3. 관절 위치 제어기의 구성
Tab 3. Configuration of joint position controller

항 목	사 양
CPU	TMS320C30(33MHz)
제어 축 수	6 축
ROM	8K X 32 비트
RAM	10K X 32 비트
Dual-Port RAM	1K 바이트
DAC 해상도	12 비트
위치 해상도	0.0879°
최대 회전수	4096 rev
제어 방식	PID with Velocity Feed Forward
위치 데이타	12 비트 해상도 엔코더
I/O	24 비트 / 24 비트

모터 제어기는 JPC로 부터 입력되는 속도 명령에 의해 속도와 토크를 제어하고 모터를 구동하기 위한 파워 드라이브 역할을 수행한다. 본 논문에서의 모터 제어기는 동기형 AC 서보 모터를 대상으로 하며, 스위칭 파워 소자로서 MOSFET(Metal Oxide Silcon Field Effect Transistor)를 사용하며 삼각파 PWM(Pulse Width Modulation) 기법을 사용하여 최대 속도 4000rpm 까지 제어한다. 모터의 속도 검출 레졸바(resolver)를 사용하며, RDC(Resolver-to-Digital Converter)를 통하여 출력되는 12 비트 병렬 위치 데이타와 12 비트 해상도의 직렬 엔코더 신호는 상위 위치 제어기로 전달된다. 모터 제어기에서 속도 및 토크 제어는 OP 암프에 의한 아나로그 PI 제어 방식이 사용되나, 모터 제어기의 이상 상태 감지 및 보호 기능 처리, 위치 제어기와의 교신을 위하여 NEC사의 8 비트 싱글칩 (single-chip) 마이크로컴퓨터 78C10을 CPU로 사용하고 있다. 한편, 구동하고자 하는 모터의 용량에 따라 모터 제어기의 출력 전류 용량은 다르게 되는데 예를 들면 30W 급의 경우 연속 출력 전류는 1.0A이고 250W 급의 경우 7.0A이다.

표 4. 모터 제어기의 구성
Tab 4. Configuration of motor controller

항 목	사 양
CPU	NEC 78C10
제어 방식	PWM 3상 정현파
속도 검출	Resolver
최대 제어 속도	4000 rpm
최저 제어 속도	1 rpm
보호 기능	과전류, 과전압, 과부하 외 7종
위치 데이타 출력	12 비트 병렬 / 12 비트 엔코더

본 논문에서의 로보트 제어기 구조는 ISA 버스를 통하여 시스템 확장 및 통합이 용이하도록 설계가 되었다. 그림 2의 개략도와 같이 현재 본 논문에서 개발한 제어기의 기능 확장을 위하여 개발 중인 시각 검사용 보드, PLC 보드, 공장 자동화용 통신 보드등이 로보트 제어기에 통합될 수 있다. 시각 검사용 보드를 사용하여 로보트 제어기는 시각 센서 정보를 처리할 수 있으므로 더욱 고지능화를 이룰 수 있고, PLC 보드를 사용하면 로보트 제어기가 각종 주변 기기와 센서등을 주도적으로 통합 제어할 수 있다. 뿐만 아니라 공장 자동화용 고기능 통신 보드 (MAP)를 사용하면 네트워크 관리용 호스트 컴퓨터와 고속으로 통신이 가능해지므로 수십개의 로보트 워크셀(work cell)의 통합 운영이 가능해지므로 궁극의 CIM(Computer Integrated Manufacturing)을 달성할 수 있다.

4. 제어 구조

본 논문에서 개발한 6축 로보트 제어기의 제어 구조는 그림 3의 개략도에 나타낸 바와 같이 제어 대상과 내용에 따라 역할 분담이 되어 제어기 별로 계층적 구조를 이루고 있다. 주 제어기에서는 시스템 운영 체제(Operating System)를 비롯하여 각종 로보트 제어 시스템에 필요한 프로그

램 모듈이 수행되는데, 주요 내용은 로보트 언어의 해석 및 실행, 정/역 기구학 해석, 궤적 계획(trajectory planning), I/O 제어 및 통신, 시스템의 이상 유무 감시 등이다. 관절 위치 제어기는 주 역할인 모터의 위치 제어 외에 모터 제어기와 주 제어기간의 통신 통로 역할을 수행한다. 모터 제어기는 모터의 속도 및 토크 제어를 실시하며 동기형 AC 서보 모터에 3상 정현파 교류 전원을 공급하는 역할을 수행한다.

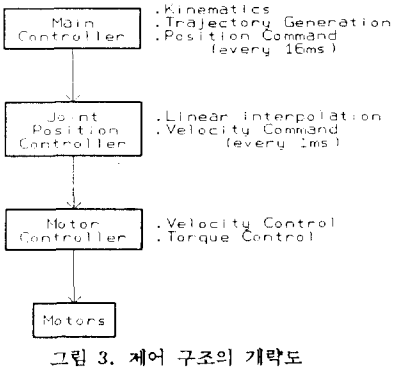

그림 3. 제어 구조의 개략도
Fig 3. Block diagram of the control structure

주 제어기는 IBM PC 호환 기종에서 널리 사용되고 있는 MS-DOS 커널(kernel)과 ROM BIOS(Basic Input Output System) 커널을 사용하여 독자적으로 개발된 로보트 운영 체제를 이용하고 있다. 이 운영 체제의 셸(shell)은 로보트 시스템 제어에 필요한 각종 명령어의 해석을 담당하며 입력된 명령은 그 명령의 속성에 따라 제어 커널, 도스 커널, BIOS 커널 등을 통하여 하드웨어를 조작하게 된다. 로보트 제어용 언어는 삼성 독자적으로 개발된 FARA 언어의 문법을 사용하고 있으며, 이 로보트 언어로써 제어기의 모든 하드웨어 상태 제어, 로보트의 동작 형태 제어, 외부 시스템 및 I/O와 연계된 동기 동작등을 수행된다.

주 제어기의 가장 중요한 역할 중의 하나는 로보트 몸체에 대한 정/역 기구학의 해석이다. 6축 로보트의 경우 역기구학 해석식은 2축 로보트나 4축 로보트의 그것과 비해 매우 복잡하며 계산량이 많이 소요되므로 서론에서 언급한 바와 같이 AT/386의 연산 능력에 의존하고 있다. 본 논문에서 개발한 로보트 몸체의 형태는 PUMA 로보트와 비슷하기 때문에 기구학 식은 PUMA의 식과 유사하다.

주 제어기는 각 하부 제어기들의 동작 상태를 제어하며 감시할 뿐만 아니라 외부 환경과의 인터페이스를 수행한다. 주 제어기는 I/O 제어기로 부터 입출력되는 스위치 및 센서의 상태와 JPC 및 모터 제어기의 동작 상태를 감시하여 갑작스런 오동작시나 에러 발생시 시스템을 보호할 수 있도록 적절한 조치를 취하게 된다.

주 제어기는 로보트 각 관절 동작의 궤적을 발생시켜 그 출력인 위치 명령은 16msec 마다 JPC에게 전달한다. 상용 로보트 제어기에서 보편적으로 사용되고 있는 관절 궤적은 사다리꼴 형태로서 생성 알고리듬이 간단하고 계산량이 적은 장점이 있으나 가속도 궤적에 불연속 점이 발생하므로 제어 시스템에 오버슈트(overshoot)가 발생되기 쉬운 단점이 있다. 또한 고차 다항식에 의한 S 곡선 형태, 큐빅 스플라인(cubic spline) 기법을 이용한 곡선 형태등이 사용되기도 하는데 가속도 형태가 연속이고 가감속이 부드러워 제어 시스템이 안정되는 등 그 성능은 사다리꼴 형태에 비해 우수하나 많은 계산량을 필요로 하므로 상용 제어 시스템에서 사용되는 경우가 드물다. 본 논문에서는 계산량이 적을 뿐만 아니라 가속도 형태의 연속성, 부드러운 S 형태의 가속 형태를 발생시킬 수 있는 컨벌루션 기법을 이용한 궤적 발생 알고리듬을 적용하였다. 본 논문에서 개발한 컨벌루션을 사용한 궤적 발생 기법에 대한 자세한 소개는 다음 장에서 소개하기로 한다.

관절 위치 제어기(Joint Position Controller)는 16 msec 마다 주 제어기에서 부터 전달되는 위치 명령을 받아 1msec 마다의 위치 명령을 발생시키는 보간(interpolation) 기능과 PID 기법을 사용한 위치 제어를 6축 전체를 대상으로 수행한다. 또한 JPC는 서보의 동작 상태를 제어하며 서보에서 발생되는 이상 상태를 주 제어기에 전달한다.

JPC에서 이루어지는 위치 보간은 선형 보간(linear interpolation)으로서 그림 4와 같은 모델에서 한 구간당 16 단계의 선형 보간을 하고자 할 때, X_i, X_f, S_i, V_i를 각각 아래와 같이 정의하고,

X_i : 구간의 초기 위치
X_f : 구간의 최종 위치
$S_i = X_i - X_f$: 구간의 변위
V_i : 구간의 초기 속도

보간점의 값 X[k]를 구하면 식 (1)와 같다.

$$X[k] = X_i + V_i + \frac{(S_i - V_i)k^2}{16^2}, \quad k=1,2,.....,16 \qquad (1)$$

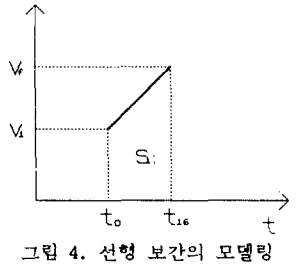

그림 4. 선형 보간의 모델링
Fig 4. The modeling of linear interpolation

위치 제어를 위하여 JPC는 매 1msec 마다 선형 보간으로 부터 출력되는 위치 명령을 추종하기 위하여 PID에 속도 피드포워드를 가한 제어 기법을 사용한다. 위치 제어기와 속도 제어기, 그리고 동기형 AC 서보 모터의 모델링은 그림 5와 같다.

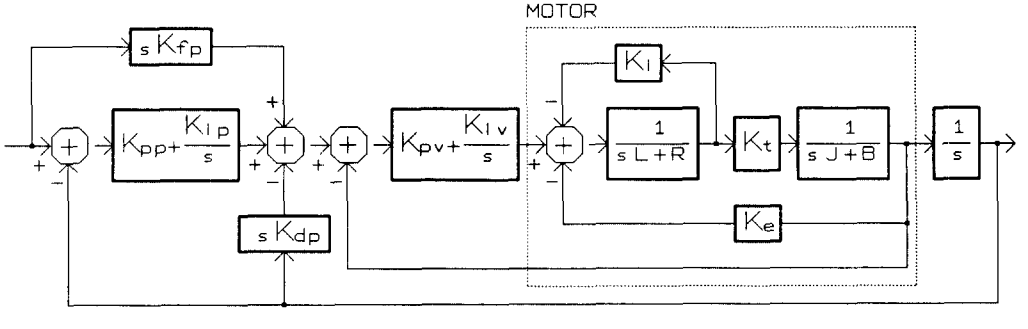

그림 5. 제어 시스템의 모델링
Fig 5. The modeling of the control system

5. 궤적 발생 기법

로보트 제어기의 개발에 있어서 궤적(trajectory)의 발생과 로보트 매니퓨레이터에 의한 궤도의 정확한 주행에 많은 연구가 이루어지고 있다. 궤적 발생 방법에는 크게 두 가지로 나눌 수 있는데, 첫번째 방법은 궤적에서의 특정점에서 매니퓨레이터의 위치, 속도, 가속도가 연속성, 부드러움등의 제약을 만족시키는 것이며, 두번째 방법은 매니퓨레이터가 직교 좌표계에서 직선 경로, 원호 경로와 같이 특정 함수에 의해 움직이도록 제약을 가하는 것이다 [3]. 궤적 발생 방법에서의 이들 두 가지에서 공통적으로 그 방법이 효과적일것, 부드러울 것, 정밀할 것, 실시간에 가깝도록 고속일 것등이 요구된다.

지금까지 발표된 궤적 발생 방법에는 고차 다항식을 이용하거나[4,5] cubic spline 방법[6]이 대부분이다. 그러나, 이들 방법은 계산이 복잡하고 부동 소숫점 연산을 하기 때문에 일반 범용 마이크로프로세서로 구현하기 어렵다. 또한 여러 축을 동시에 제어할 때 각 축의 동작을 동시시키는 방법이 복잡하다.

본 논문에서는 디지틀 필터 이론에서의 컨벌루션을 이용하여 위에서 열거한 문제점들을 해결할 수 있는 방법을 개발하였다. 본 논문의 이론 전개에 있어서 로보트의 작업 공간내에 장애물이 없다고 가정하였으며, 매니퓨레이트의 동역학적인 요소를 무시하였다.

주어진 두 디지틀 신호에 대한 컨벌루션은 다음과 같이 정의된다[7]. 즉, 시간 지속 (time duration)이 유한한 신호 x[n]과 h[n]에 대하여 컨벌루션으로 얻는 새로운 신호 y[n]은

$$\begin{aligned} y[n] &= x[0]h[n]+\ldots+x[k]h[n-k]+\ldots+x[n]h[0] \\ &= \sum_{k=0}^{n} x[k]h[n-k] \\ &= \sum_{k=0}^{n} x[n-k]h[k] \end{aligned} \tag{2}$$

이다. 식 (2)를 이진 연산자 '*'로 다시 표현하면,

$$y[n] = x[n]*h[n] \tag{3}$$

디지틀 필터 이론에 의하면, 임펄스 응답 h[n]을 가진 필터에 신호 x[n]을 입력시켰을 때의 출력 y[n]은 x[n]과 h[n]의 컨벌루션 즉, 식 (3)과 같다.

컨벌루션 이론을 이용한 궤적 발생 알고리듬을 설명하기 위하여 먼저 다음의 용어들을 정의한다.

$S_f[i]$: 로보트 팔의 i 번째 관절에서의 변위량
$V_{max}[i]$: i 번째 조인트에서의 최대 속도
Q(a,b) : 정수 a를 정수 b로 나누었을 때의 몫
R(a,b) : 정수 a를 정수 b로 나누었을 때의 나머지
K_{ap} : 가속 구간 수
T_s : 샘플링 시간

그리고, 샘플링 시간이 T_s이고, K_{ap}의 시간 길이를 가진 한 신호를 h[n]으로 정의한다. 그림 1은 h[n]의 한 예이며, 이 h[n]의 적분 Σh[n]의 값은 1이다.

i 번째 관절에서의 변위량 $S_f[i]$를 이용하여 진폭이 $V_{max}[i]$인 신호 $x_i[n]$을 만들면, 신호 $x_i[n]$은 진폭이 $V_{max}[i]$인 신호 $Q(S_f[i],V_{max}[i])$ 개와 진폭이 $R(S_f[i], V_{max}[i])$인 신호 1 개로 구성된다.

신호 $x_i[n]$에서 $R(S_f[i],V_{max}[i])$를 제외한 신호를 $x'_i[n]$이라고 하면, $x'_i[n]$은 단위 신호(unit step function) u[n]을 이용하여 아래와 같이 표현할 수 있다.

$$x'_i[n]=V_{max}[i](u[n]-u[n-Q(S_f[i],V_{max}[i])]) \tag{4}$$

신호 $x'_i[n]$를 h[n]과 컨벌루션하여 얻은 신호 $y_i[n]$은 다음과 같다.

$$\begin{aligned} y_i[n] &= x'_i[n]*h[n] \\ &= \sum_{k=0}^{n} x'_i[k]h[n-k] \\ &= \sum_{k=0}^{n} V_{max}[i](u[n]-u[n-Q(S_f[i],V_{max}[i])])h[n-k] \\ &= \sum_{k=0}^{n} V_{max}[i](h[n-k]-h[n-k-Q(S_f[i],V_{max}[i])]) \end{aligned} \tag{5}$$

식 (5)를 리컬시브 형태로 나타내면 아래와 같다.

$$y_i[n]=y_i[n-1]+V_{max}[i](h[n]-h[n-Q(S_f[i],V_{max}[i])]) \tag{6}$$

식 (6)의 물리적 의미를 살펴보면 h[n]이 곧 관절에서의 가속도(토크) 궤적이 되며 $y_i[n]$은 속도 궤적이 됨을 알 수 있다. 그리고 y[n]의 적분

$$p_i[n]=p_i[n-1]+y_i[n],\ p_i[0]=0 \tag{7}$$

은 위치 궤적이다. 따라서 컨벌루션을 이용하면 미리 정의된 가속도 궤적과 변위량을 이용하여 속도 궤적을 발생할 수 있으며 이 속도 궤적으로 부터 위치 궤적이 만들어진다. 식 (6)의 유도에서 제외된 값 $R(S_f[i],V_{max}[i])$은 위치 궤적의 구간 수 $Q(S_f[i],V_{max}[i])+K_{ap}$로 나눈 값 $Q(R(S_f[i],V_{max}[i]),Q(S_f[i],V_{max}[i])+K_{ap})$을 위치 궤적의 전 구간에서 산포하는 방법과 V_{max}를 다음과 같이 정의하고

$$V_{max}=Q(S_f[i],Q(S_f[i],V_{max}[i])+1) \tag{8}$$

$R(S_f[i],Q(S_f[i],V_{max}[i])+1)$를 위치 궤적의 전구간에서 산포하는 방법이 있다.

식 (6)에서 속도 궤적을 만들기 위한 계산량을 보면 샘플 구간당

덧셈 : 1회
뺄셈 : 1회
곱셈 : 1회

가 소요됨을 알 수 있다. 그리고, 수식에 사용된 변수가 모두 정수이므로 범용 마이크로 프로세서를 사용하여 실시간 연산이 가능하다.

컨벌루션을 이용한 궤적 발생 기법의 가장 큰 장점은 6축 로보트와 같은 다축 제어 시스템에서 전축의 동기 동작(synchronized motion)을 쉽게 만들 수 있다는 점이다. 6축 로보트를 예로 들어 살펴보자. 각 축의 변위 $S_f[i]$, i=1..6 중 가장 큰 값을 S_{max}라 한다. S_{max}를 V_{max}에 대하여 정규화하면 $Q(S_{max},V_{max})$를 얻을 수 있고, 나머지 다섯 축의 변위를 $Q(S_{max},V_{max})$로 나누면 진폭이 V_{max}보다 같거나 작고 시간 길이가 $Q(S_{max},V_{max})$인 각각의 신호를 얻는다. 이렇게 하여 얻은 6개의 신호 $x_i[n]$, i=1..6를 각각 필터 특성 h[n]에 컨벌루션하면 6축 전체가 동기되어 운동한다.

컨벌루션을 이용한 궤적 발생 방법은 디지틀 필터 이론에 기초한 것으로써, 고차 다항식을 이용하거나 spline 곡선 알고리듬을 이용하는 방법보다 계산량이 매우 적다. 따라서, 궤적 발생의 온라인 계산이 가능하며, 계산이 모두정수 연산이기 때문에 일반 범용 마이크로 프로세서를 사용하여 알고리듬 구현이 용이하다. 필터 특성 h[n]을 미리 규정하여 look-up table로 ROM이나 RAM에 저장할 수 있으므로 비선형 고차함수를 h[n]에 이용할 수 있으며, 시스템의 특성에 맞도록 h[n]을 가변할 수 있다. 또한, 컨벌루션을 이용하여 궤적을 발생시키면 관절 좌표계에서 뿐만 아니라 직교 좌표계에서 모든 축을 동기화시켜 운동하는 방법이 간단해지는 장점이 있다.

6. 실험 및 결과

본 논문에서 개발한 로보트 몸체 및 로보트 제어기의 제어 특성을 시험하기 위하여 제어기를 로보트 몸체와 연결하여 실험을 실시하였다. 표 5는 실험에 사용된 삼성 전자에서 개발한 6축 수직 다관절 로보트 몸체의 사양이며, 그림 6은 6축 로보트 몸체와 로보트 제어기의 실물 사진이고, 그림 7은 관절 위치 제어기의 실물 사진이다.

표 5. 6축 로보트의 제원
Tab 5. Specifications of 6 axes robot

항 목		사 양
형 식		수직다관절
자유도		6
작업영역	1축	320° (±160°)
	2축	320° (±160°)
	3축	270° (±135°)
작업영역	4축	360° (±180°)
	5축	210° (±105°)
	6축	560° (±280°)
동작속도	1축	150° / s
	2축	150° / s
	3축	150° / s
	4축	163° / s
	5축	163° / s
	6축	204° / s
	CP	1000 mm / s
가반중량		3 Kg
축간거리	BASE-2축 중심	625 mm
	2축 중심-3축 중심	300 mm
	3축 중심-5축 중심	300 mm
	5축 중심-6축 선단	91 mm
구동방식		AC SERVO MOTOR
반복정밀도		± 0.05 mm
중 량		90 Kg

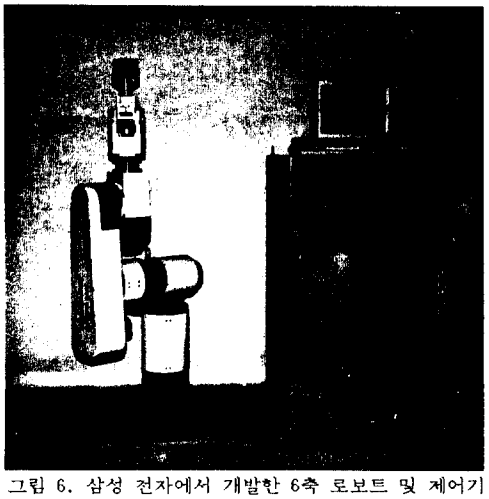

그림 6. 삼성 전자에서 개발한 6축 로보트 및 제어기
Fig 6. The 6-axes robot and controller developed in Samsung Electronics

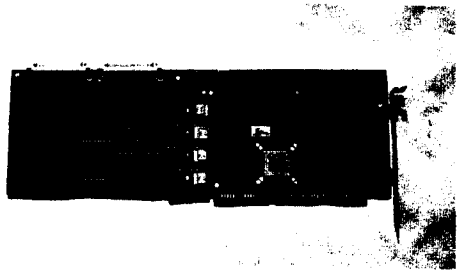

그림 7. 관절 위치 제어기
Fig 7. Joint position controller

디지틀 제어기는 아나로그 제어기에 비해 주위 온도 변화등의 환경의 영향이 적다는 점, 제어 변수를 프로그램으로 가변할 수 있으므로 유연성이 뛰어나다는 점등의 장점이 있으나 아나로그 제어기에 비해 실시간(real-time) 제어가 이루어지지 않는다는 단점이 있다. 그러나 모터 제어 시스템은 필연적으로 저역 통과 필터 (low-pass filter)의 특성을 지니므로 디지틀 제어 방식을 사용하더라도 나이키스트 원칙(Nyquist Theorem)을 만족시킬 수 있도록 충분한 제어 간격을 유지한다면 아나로그 제어 특성에 가까운 성능을 낼 수 있다. 표 6은 6축 로보트의 원호 CP(Continuous Path) 동작시 주제어기의 제어 주기와 JPC에서의 제어 주기를 측정한 결과이다. 표 6에서 주 제어기의 제어 주기 16msec에 실제 소요 시간은 8.7msec인데 나머지 7.2 msec 동안에는 각종 I/O 제어, 다음 구간의 계산 등에 소요된다. JPC에서의 제어 주기는 1msec 임에 비해 실제 소요 시간은 200usec로서 6축 전체의 제어 시간이 제어 주기의 1/5이다. 나머지 4/5의 시간은 비주기적인 주 제어기와의 통신을 하거나 아무런 일을 하지 않지만 향후 좀더 복잡한 제어 알고리듬으로의 개선을 위하여 이 시간은 현재로서는 유보되어 있다.

표 6. 제어기의 제어 주기
Tab 6. Control cycle of Controllers

제어기	제어 주기	실 제어 시간
주 제어기	16 msec	8.7 msec
JPC	1 msec	200 usec

그림 8은 본 논문에서 개발된 컨벌루션 기법을 적용하여 얻은 속도 궤적과 가속도 궤적을 측정한 결과이다. 가속도 궤적은 정현파를 사용하였으며 그림 8은 모터의 구동 용량에 해당하는 등가 부하를 모터에 인가한 후 모터 제어기의 속도 제어의 출력 신호를 측정한 것으로서 원래 설계했던 가속도 궤적인 정현파와 거의 일치하는 모습을 볼 수 있다.

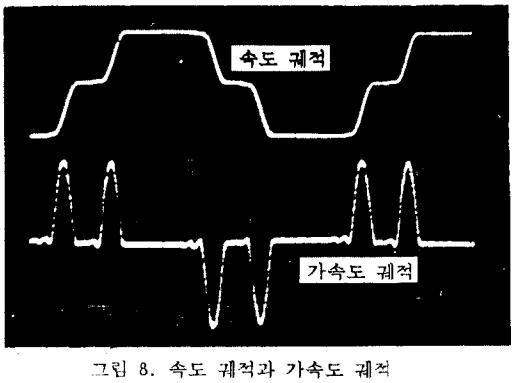

그림 8. 속도 궤적과 가속도 궤적
Fig 8. Trajectories of velocity and acceleration

그림 9는 컨벌루션에 의해 생성된 속도 궤적과 모터 제어기에서 피드백되는 속도 궤적을 측정한 결과이다. 이 그림에서 위의 파형은 속도 명령이고 아래의 파형은 궤환된속도 파형으로서, 모터 제어기는 JPC로 부터의 속도 명령을 잘 추종함을 알 수 있다.

그림 10은 본 논문에서 개발한 6축 로보트 제어기로 6축 수직 다관절형 로보트를 구동하여 직선 CP(Continuous Path), 원호 CP를 실행한 결과이다. 이 실험에서의 로보트의 동작 속도는 허용 최대 속도의 90%로서 900mm/s이다.

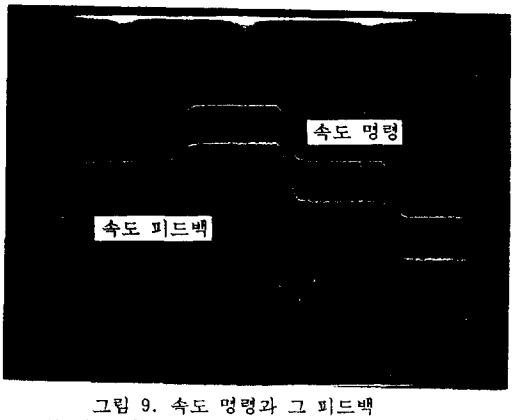

그림 9. 속도 명령과 그 피드백
Fig 9. Velocity command and the feedback

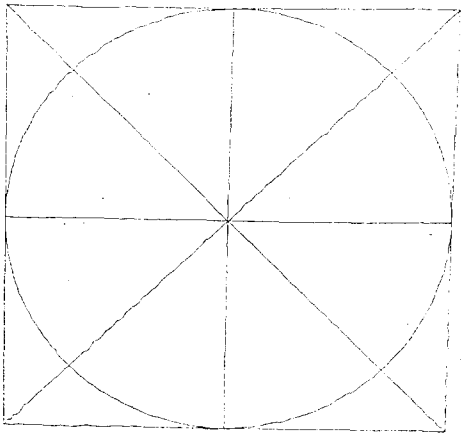

그림 10. 속도 900mm/s에서의 직선/원호 CP
Fig 10. Linear/circular CP at speed of 900mm/s

7. 결론

본 논문에서는 삼성 전자에서 개발한 6축 로보트 제어기의 설계 개념과 하드웨어 구조, 제어 구조, 궤적 생성 기법에 대하여 고찰하였다. 본 논문에서 개발한 제어기는 운영 체제로서 MS-DOS 커널을 사용하여 소프트웨어 개발이 용이하였고 로보트 제어기간의 프로그램 이동이 편리하다. 그리고, 주 제어기의 ISA 버스에 인터페이스 함으로써 JPC, I/O 제어기 등의 기능별 하부 제어기를 개발함으로써 간단하고도 명확한 제어 구조를 구현하였다. 특히, 32 비트 고속 DSP인 TMS320C30을 JPC의 CPU로 사용하여 시분할 방식에 의한 6축 전체의 위치 제어를 수행함으로써 종전의 각 관절마다 프로세서를 할당하는 방식에 비해 관절 위치 제어기의 하드웨어 및 소프트웨어 구조가 간단해졌다. 공통 버스 구조에 의한 시스템 기능 확장이 가능하므로 로보트 제어기를 중심으로 PLC, Vision System, MAP의 통합을 위한 연구가 현재 계속 진행 중이다. 또한, 아무리 고성능, 고기능의 로보트 제어기라 하더라도 신뢰성이 미흡하여 실제 생산 현장에서 고장이 잘 발생한다면 쓸모없는 존재에 불과하므로 신뢰성 향상을 위한 지속적인 현장 테스트가 현재 진행 중이다.

참고문헌

[1] 김 성권, "로보트 제어기 제조 기술", 대한기계학회지 게재 예정
[2] R. D. Klafter, T. A. Chmielewski, M. Negin, *Robotic Engineering*, Prentice-Hall, 1989.
[3] K. S. Fu, R. C. Gonzalez, and C. S. G. Lee, *Robotics Control, Sensing, Vision, and Intelligence*, McGraw-Hill, 1987.
[4] K. J. Kyriakopoulos and G. N. Saridis, "Minimum jerk path generation," Proceedings 1988 IEEE International Conference on Robotics and Automation, pp. 364-369.
[5] J. Y. S. Luh and C. S. Lin, "Approximate joint trajectories for control of industrial robots along cartesian paths," IEEE Trans. on Systems, Mans,and Cybernetics, vol. SMC-14, no. 3, pp. 444-450, 1984.
[6] B. K. Kim, "Joint space path generation of industrial robots using low-order spline functions," Journal of KIEE, vol. 3, no. 2, pp. 113-122, 1990.
[7] A. Papoulis, *Circuits and Systems*, Holt, Rinehart and Winston, 1980.