

PWM디지털 서어보 제어기를 이용한 산업용로봇의 연속경로 제어에 관한 연구

(A Study on Continous Path Control of Industrial Robot Using PWM Digital Servo Controller)

金 志 泓*, 吳 英 碩*, 崔 榮 奎*, 卞 增 男*

(Jee Hong Kim, Young Seok Oh, Young Kiu Choi and Zeungnam Bien)

要 約

본 논문은 산업용 로봇의 PWM 디지털 서어보 제어기와 실시간 궤적계획 알고리즘 및 궤적추적알고리즘의 설계에 관한 것이다. 구체적으로 산업용 로봇의 제어가 계급적 구조의 분산 제어 시스템으로 실현되었으며, 부속 Hardware 및 Software가 설계 제작되었고, 사용자의 편의를 도모하기 위한 모니터 프로그램이 개발되었다. 실제로 이 서어보 제어기와 제어 알고리즘을 이용하여 R- ϕ 평면상에서 연속경로 제어가 가능함을 실험적으로 알 수 있다.

Abstract

This paper presents a design of the PWM digital servo controller and a real-time trajectory planning algorithm and a trajectory tracking algorithm for industrial robot.

To be specific, a decentralized control system with hierarchical structure for industrial robot, related hardware and software, and monitor program for convinence of user are implemented.

Actually, it was recognized by experiments that continuous path control on the R- ϕ plane was possible using the above servo controller and control algorithms.

I. 序 論

60년대에 들어 등장한 산업용 로봇은 오늘날 생산 현장에서 용접, 도장, 운반, 가공 등에 광범위하게 이용되고 있다. 산업용 로봇은 FA(Factory Automation)의 주역으로 생산성을 향상시키고 간단한 프로그램의 조작에 의해 작업수행의 다양화를 기할 수 있다. 특히

최근 급속히 발전하는 전자 공업과 컴퓨터, 정밀 기계 공업의 결합으로 고도의 지능을 갖춘 로봇과 무인화 공장이 출현하기 시작했고 일본 및 구미 선진국에서는 이러한 것들이 핵심 연구과제로 부각되고 있다. 이러한 세계적 추세 및 요구에 부응하여 한국과학기술원 제어 시스템 연구실에서는 수년전부터 NC를 비롯하여 산업용 로봇의 점이동(point-to-point: PTP) 제어 및 연속 경로(continuous path: CP) 제어에 관해 연구하고 있다.^{1,2)} 본 연구에서는 디지털 PWM(Pulse-Width-Modulation) 모터 구동장치와 8비트 마이크로 프로세서를 이용한 디지털 서어보 제어기 및 16비트 마이크로프로세서인 MC 68000 based Mono board

*正會員, 韓國科學技術院 電氣 및 電子工學科
(Dept. of Electrical Engineering, KAIST)

(※ 본 논문은 한국과학재단의 지원에 의하여 이루어진 것임.)

接受日字: 1985年 1月 28日

computer를 계급적으로 결합하여 산업용 로봇 KAI-SEM 1의 실시간(real-time) 연속경로(CP) 제어에 대해 연구하였다. 본 연구에서 사용한 KAISEM ROBOT 1은 1979년에 한국과학원에서 제작한 자유도 3의 원통 좌표형 로봇이다.

이 로봇의 R, ϕ , z축은 직류 서보모터에 의해 구동되고, 손과 손목은 공압 제어에 의해 ON/OFF 형식으로 구동된다.

직류 모터의 속도 및 위치 제어를 위한 구동 방식으로는 듀얼 컨버터 방식과 PWM 방식을 들 수 있다.

듀얼 컨버터의 방식은 SCR을 사용한 전력 변환기를 이용하여 SCR의 도통각을 조절하며, PWM 방식은 전력 변환기로서 스위칭이 빠른 전력용 트랜지스터를 사용하고 트랜지스터에 인가되는 펄스의 폭을 조정하나, 이제까지 사용한 아날로그 하드웨어에 의한 제어는 수준높은 제어 알고리즘의 실현이 복잡하며 IC의 offset, drift 등에 의해 성능이 악화될 수 있다. 본 연구에서는 기존의 PWM 신호발생 회로 및 비례 적분 속도 제어를 간단한 디지털 타이밍 회로와 8비트 마이크로 컴퓨터의 소프트웨어로 대체함으로써 로봇 제어 알고리즘의 유연성과 제작이 손쉽고 정확한 장점을 갖도록 하였다. 또한 성능 향상을 위하여 계급적으로 구성된 컴퓨터 시스템의 주 컴퓨터는 연속 경로 및 각 관절의 제어를 계획하며 실시간으로 제어가 가능하도록 계산 속도가 빠른 MC68000마이크로 프로세서를 사용하고 부동 소숫점 연산이 가능하도록 소프트웨어를 구성하였다.

중속 컴퓨터는 각 관절별로 주어진 궤적을 추적하기 위한 추적 알고리즘을 수행하여 PWM 구동 장치를 동작시킨다.

본 논문의 구성은 전체 시스템의 구성, 하드웨어의 구성, 제어 알고리즘 및 소프트웨어의 구성, 실험 결과 및 검토 그리고 결론의 순서로 PWM 디지털 서보 제어기의 실현 및 궤적 계획 및 추적 알고리즘에 대해 기술하였다.

II. 전체 시스템의 구성

본 논문에서 서술할 로봇 시스템은 크게 로봇 팔과 구동부, 서보 시스템, 컴퓨터 시스템 및 센서로 나눌 수 있다.^[5](그림 1)

컴퓨터 시스템은 인간의 두뇌에 해당하며, 사용자의 명령과 각 센서를 통해 들어온 로봇의 상황을 받아들이고, 각 서보 시스템은 주어진 제어 방식에 따라 제어신호(PWM 출력)을 발생하고 이를 증폭시켜 로봇 팔의 구동부를 구동시킨다. 본 시스템에서는, 각

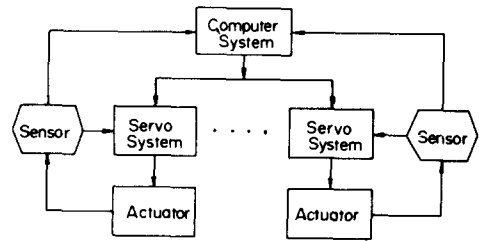


그림 1. 로봇 제어 시스템의 일반 블록 선도
Fig. 1. General block diagram of robot control system.

서보 시스템을 중속 컴퓨터인 Z-80이 제어하고, 2대 이상의 서보 시스템을 MC 68000 주컴퓨터가 통괄하여 제어하는 계급적 구조를 갖고 있다. 각 서보 시스템은 독립적으로 움직이므로 주컴퓨터는 각 축의 모터의 위치와 속도를 조화시켜 제어함으로써, 로봇 팔이 원하는 경로를 따라 움직이게 한다. 센서는 모터의 현재 위치를 측정하여 컴퓨터와 서보 시스템에 전해주고, 모니터의 전기가 전류가 주어질 최대치를 초과했거나, 로봇 팔이 기계적 구조상 움직일 수 있는 범위를 넘어가려는 순간에 그 상황을 알려 준다. 본 연구에서는 컴퓨터 시스템 인터페이스 및 소프트웨어와 PWM 디지털 서보 제어기가 제작되었고 그 구조는 그림(2)와 같다.

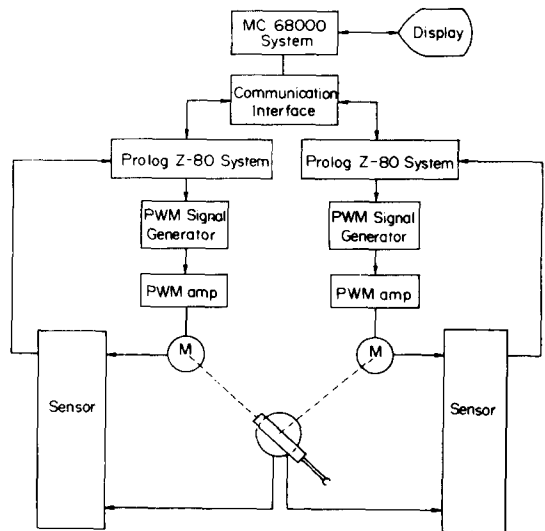


그림 2. 전체 시스템 구조
Fig. 2. Structure of the overall system.

III. 하드웨어의 구성

본 연구에서는 제작한 제어기의 하드웨어는 마이크로

도통시키기 위해서는 1A 이상의 베이스 전류가 필요하므로 각 트랜지스터에 구동회로가 첨가되었다. PWM 증폭기는(그림5, 6)과 같다.

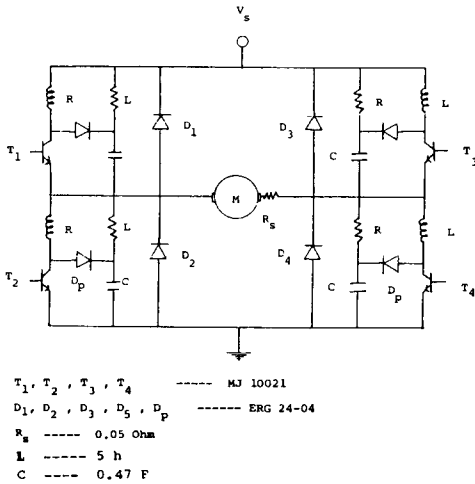


그림 6. PWM 증폭기
Fig. 6. PWM amplifier.

4) 위치 측정 회로 및 전류 측정 회로

직류 전동기의 광학 엔코더에서는 회전방향에 따라 90°의 위상차를 갖는 2 개의 펄스가 1 회전당 2000개씩 나오며 이를 up-down 카운트하여 16비트 위치 정보로 사용한다. 한편 모터의 전기자 전류는, 전기자 회로에 전기자 저항보다 훨씬 적은 값의 외부 저항을 직렬 연결하고 외부 저항 양단의 전압차로 측정되며, 이 값이 지정된 한계를 넘게되면 인터럽트 신호가 발생되게 된다.

IV. 제어 알고리즘 및 소프트 웨어의 구성

1) 궤적 계획 알고리즘

로봇트 팔의 end effector가 움직여야 할 공간상의 궤적을 경로(path)라 하며 이 경로가 시간을 매개 변수로 하여 표시 될 때 이를 궤적(trajecory)이라 할 수 있다. 본 연구에서는 연속 경로증 기본이 되는 직선 경로 운동에 역점을 두어 이를 조인트 좌표상의 궤적으로 바꾼 뒤, 서어보 시스템에 명령하는 방법을 사용하였다. 로봇트 팔이 직각 좌표계의 시점 P_i과 종점 P_f 사이에 주어진 경로를 가기 위하여는, 시점과 종점 사이의 경로를 여러개의 구간으로 분할하여 각 구획점에 대한 위치 정보를 제어 기간마다 서어보 시스템에 전해주어야 하며, 부드러운 운동을 위하여 경로를 가속구간과 정속도 구간 및 감속 구간으로 구분

하여 직각 좌표계의 구획점의 위치 정보가 inverse kinematics에 의해 조인트 좌표로 나타낸다.

KAISEM1 robot의 좌표계는(그림7)과 같으며, R-φ 평면상의 직각좌표(x, y)와 조인트 좌표(r, φ)의 관계는 식(1)에서 식(4)와 같이 나타난다.

$$x = (RF+r) \cos \phi + LF \sin \phi \quad (1)$$

$$y = (RF+r) \sin \phi - LF \sin \phi \quad (2)$$

$$r = \sqrt{x^2 + y^2} - LF^2 - RF \quad (3)$$

$$\phi = \tan^{-1} \frac{LF \cdot x + (RF+r) \cdot y}{-LF \cdot y + (RF+r) \cdot x} \quad (4)$$

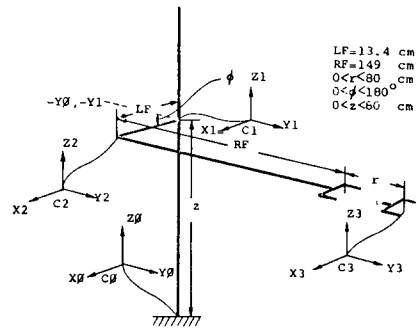


그림 7. 로봇트의 기하학적 구조
Fig. 7. Geometric diagram of robot.

이때의 직각 좌표계에서의 속도 및 위치 profile은(그림8)과 같으며 조인트 좌표계에 대응하는 각 점의 정보를 매 제어기간 마다 각 서어보 시스템에 명령한다. 따라서 컴퓨터에서는 점 P_{k-1}에서 P_k까지 로봇트 팔이 움직일동안 다음 위치의 정보를 미리 계산하고 있어야 하며 계산량이 많아 주어진 시간내 위치 정보를 제공할 수 없다면 off-line에 의해 계산하거나 조인트 좌표상에서 보간(interpolation)에 의해 해결하여야 한다. 그리고 분할점의 수에 의해 로봇트 팔의 속도가 결정되며 실험에서는 16msec 마다 위치 정보를 서어보 시스템에 명령하고 서어보 시스템에서는 2msec 간격으로 보간에 의해 전체적으로 실시간 궤적 계획이 가능하도록 하였다.

특히 자료 처리를 위한 각종 계산은 부동 소숫점 연산을 이용하여서 유효 숫자의 증가에 노력하였고, 초월 함수나 기타 수학적인 함수 값은 table look-up 혹은 소프트 웨어 알고리즘에 의해 실현되어 68000 어셈블리 언어에 의해 프로그램 되었다. 실제로 32bit word에 유효 숫자를 23bit로 잡고 8kbyte look-up table과 Taylor series를 사용하였다.

한 예로 평방근 계산시 최대 오차가 0.002%인 연산이 가능함을 알 수 있다.⁽⁶⁾

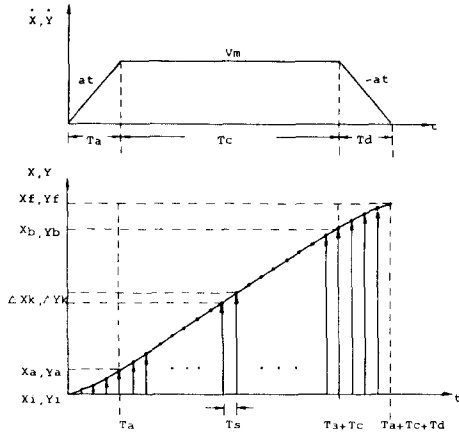


그림 8. 직각 좌표계에서의 속도 및 위치 profile
Fig. 8. Profile of speed and position in cartesian space.

2) 경로 추적 알고리즘

서어보 제어기는 매 제어기간마다 계단 위치 명령이 입력되고, 가능한 한 빠른 시간안에 도달하도록 설계하였다. 이러한 경우 각 구획점에서 정지하려는 경향이 나타나고, 이것은 각 구획에서 영이 아닌 속도를 명령함으로써 제거할 수 있다. 실제로 서어보 제어기는 명령으로 주어진 위치 정보를 이용하여 제어기간 내에 주어진 위치에 도달하도록 ramp 입력을 만든 후 그 값을 추적하도록 구성하여 부드러운 운동을 실현하였다. 따라서 각 제어구간에서 ramp 위치 입력을 가진 위치 제어기를 (그림 9)와 같이 PID 제어기를 사용하여 구현하였다.

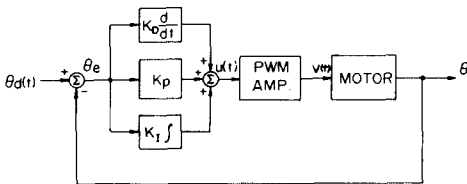


그림 9. PID 제어기
Fig. 9. PID controller.

이는 미분기가 오차를 예견하여 댐핑을 줌으로써 안정성을 개선시키며 디지털 시스템에서는 D-제어의 실현이 용이하기 때문이다. 그리고 적분 제어는 정상상태 오차를 제거하는 역할을 한다. PID 이득 K_p , K_i , K_d 의 선택은 시스템 성능에 중요한 영향을 미치며, 본 제어기는 성능지수를 식(5)과 같이 잡고 최대 속도 운전시 J 를 최소화하는 K_p , K_i , K_d 를 최적화 프로그램에 의해 구하였다.^[11]

$$J = \int_0^{t_f} |t(\theta_d - \theta) + 0.1u^2(t)| dt \quad (5)$$

θ_d = 위치 명령, θ = 위치 출력
 u = 제어 입력, t_f = 최종 시간(16msec)

(그림 9)에서 PWM Amp.의 gain은 1로 잡고, 제어대상인 모터로서 직류서어보 FANUC사 제품 MODEL 5L을 사용하였다.

모터의 사양은 (표 1)과 같으며 입력전압 V 와 모터의 각속도 ω 와의 관계는 (식 6)과 같은 전달함수로 모델링 할 수 있다.

$$\omega(s) = \frac{K_t}{(R_a + L_a s)(B_{eff} + J_{eff} s) + K_v K_t} V(s) \quad (6)$$

$$= \frac{K_t / R_a B_{eff}}{(1 + \tau_m s)(1 + \tau_e s) + K_v K_t / R_a B_{eff}} V(s)$$

단, $B_{eff} = B_L + B_m \approx 2B_m$, $J_{eff} = J_L + J_m \approx 2J_m$

(표 1)에서 기계적 시정수 τ_m 이 전기적 시정수 τ_e 보다 충분히 크다고 볼 수 있으므로 식(6)을 식(7)과 같이 간략화한 후 표 1의 값을 대입한 결과 모터에 인가한 전압 $v(t)$ 와 모터의 위치 $\theta(t)$ 사이에는 식(8)과 같은 전달함수가 성립되고 이 관계를 시뮬레이션에 이용하였다.

$$\omega(s) = \frac{K_t / R_a B_{eff}}{(1 + \tau_m s) + K_v K_t / R_a B_{eff}} V(s) \quad (7)$$

$$\frac{B(s)}{V(s)} = \frac{2.08}{s(14.5 \times 10^{-2} s + 1)} \quad (8)$$

표 1. 직류 서어보 모터 사양

Table 1. DC servo motor specification.

PARAMETER	SYMBOL	VALUE	UNITS
Rotor Inertia	J	0.050	kg, cm, sec ²
Mechanical Time constant	τ_m	0.015	sec
Back EMF constant	k_v	0.477	V, sec/rad
Torque Constant	k_t	4.87	kg, cm/Amp
Armature D. C. Resistance	R_a	0.68	Ohm
Armature Induc.	L_a	0.0027	Henry
Electrical Time Constant	τ_e	0.0038	sec
Viscous Friction Coefficient	B	0.033	kg, cm, sec/red

시스템의 성능으로서의 속응성을 나타내는 정착시간과 정상상태 오차를 고려하여 성능 지수를 식(5)와 같이 선택하였고, 입력을 제한하는 가중인자는 여러 값에 대한 시스템의 응답을 조사하여 식(5)와 같이 선택하였다.

이러한 이득의 제어 시스템의 최대 속도 입력에 대한 응답은 (그림 10)과 같으며, 실제의 제어 알고리즘은 Z-80 어셈블리 언어로 구현되었다.

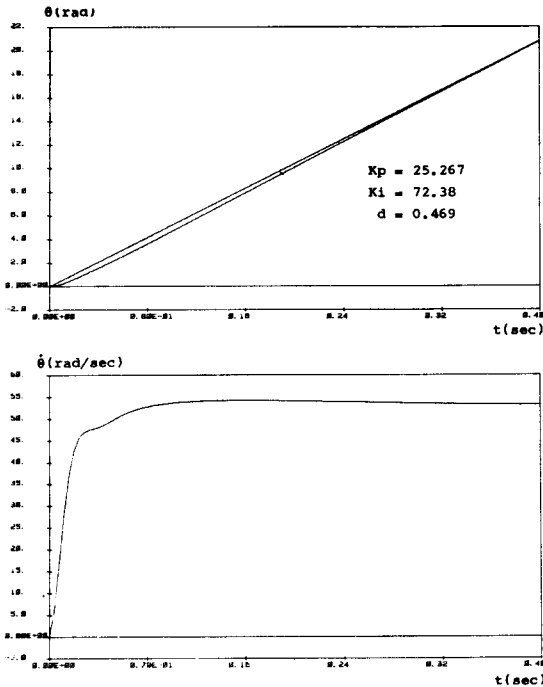


그림10. PID 제어 시스템의 시뮬레이션 결과
Fig.10. Simulation result of PID control system.

프로그래밍시 모든 위치 data는 16bit로 처리하였으므로 위치 정확도를 0.05mm까지 정교한 제어가 가능하였다.

3) 모니터 프로그램

모니터 프로그램은 사용자가 CRT 터미널에서 주는 명령 및 그에 따른 데이터를 컴퓨터에 적합하도록 변경하고 이를 디코딩하여 원하는 명령을 수행하도록 한다. 명령에 따라 각 서어브 루틴을 수행하고 결과적으로 각 관절의 서어브 시스템에 명령을 내리게 된다. 또한 입력상의 오차나 현재의 상황등을 표시하여 사용자의 편의를 제공한다. 수행하는 명령의 종류는 표 2와 같으며 입력시 Jog-mode는 사용자가 로봇트 팔을 원하는 위치로 이동시킨 후 현재의 위치값을 저장하고 필요한 경우 그 위치로 로봇트를 움직이도록 하며 5개의 위치값을 저장 하였다가 연속적으로 로봇트 팔을 이동시킬 수 있다.

모니터 프로그램의 계통도는 (그림11, 12)와 같으며, 전체적으로 인터럽트 기능을 충분히 활용하였다.

특히 slave-computer 시스템의 UIC(Universal Interrupt Controller)는 모든 입출력 및 로봇트의 상황 판단 기능을 순위가 정해진 인터럽트 서어비스 루틴으로 수행하여 빠른 응답과 정확한 샘플링 타임을 가능하게 하며 컴퓨터간의 데이터 통신을 원활히 수행

하였다.

표 2. 모니터 프로그램에서 사용하는 명령들
Table 2. Command in monitor program.

명 령	의 미
RESet	Reset to Zero-Position,
SET #1 #2	(#1, #2) 위치로 가라.
XEQ	연속경로로 움직여라.
Zero	정지하라.
Emergency	위험하다./ 무조건 정지하라.
Run	Jogmode Data를 수행하라(연속적)
Jog	Jogmode를 수행하라(단계적)
+R	R-축을 증가시켜라.
-R	R-축을 감소시켜라.
+P	ϕ -축을 증가시켜라.
-P	ϕ -축을 감소시켜라.

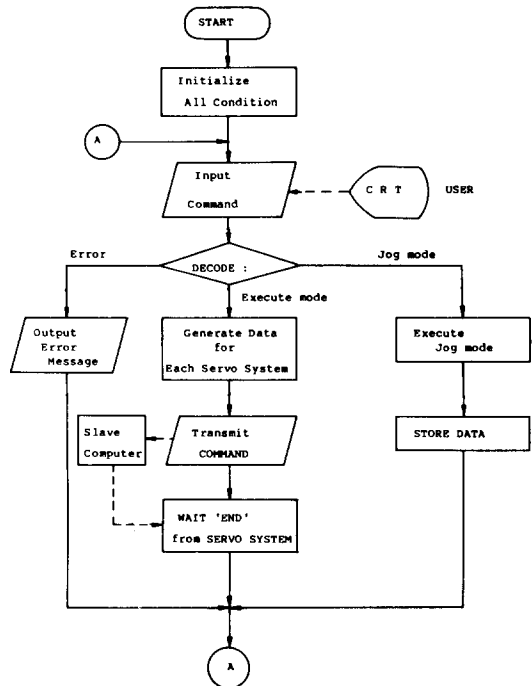


그림11. 모니터 프로그램 계통도(host-computer)
Fig.11. Monitor program flow-chart.

V. 실험 결과 및 검토

III, IV절에서 설명한 PWM 디지털 서어브 제어기를 제작하여 두 점의 좌표를 교시한 후 로봇트 팔의 직선 운동을 실험하였다.

로봇트 팔에 펜을 부착한 후 평면상의 직선 운동을

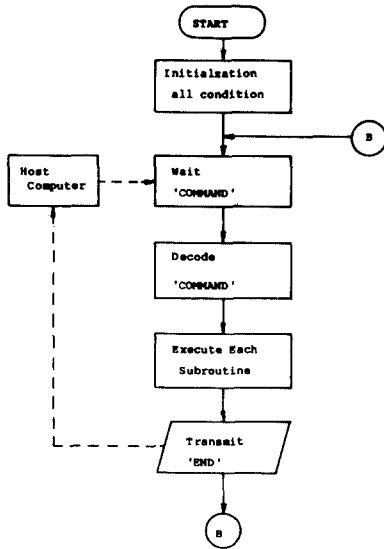


그림12. 모니터 프로그램 계통도(slave-computer)
Fig.12. Monitor program flow-chart.

실험한 결과를 (그림13)에 도시하였다. 궤적 계획 알고리즘에서 직선 구간을 80(hex)의 구간으로 나누었을 때 A점에서 B점으로 이동시 12초가 걸렸다. 40(hex)개로 나누면 8초가 걸리고 평균 속도는 각각 9.6(cm/sec), 14.4(cm/sec) 나타났다. A점에서 B점으로 이동시 로봇트 팔의 경로는, A, B의 직선 경로에서 최대 이탈 오차가 0.3cm로 거의 직선에 가까웠고

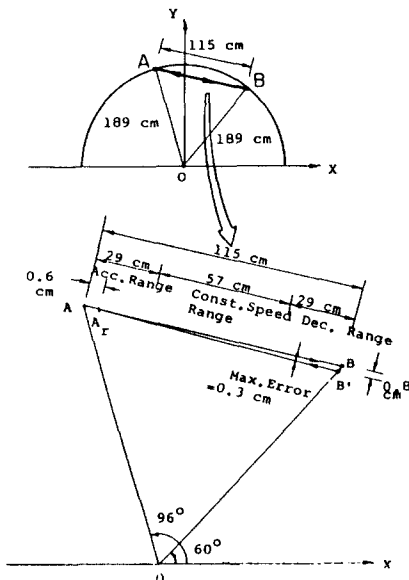


그림13. 실험결과
Fig.13. Experiment result.

B점에서 A점으로의 이동시의 경로와 일치하지 않는 것은 로봇트 축의 기계적 backlash에 기인된 것이다.

IV. 結 論

PWM 디지털 제어기에 의한 로봇트의 위치 및 속도 제어가 연구 되었고, 이와 더불어 연속 경로 운동에 대한 연구도 수행되었다.

구체적으로 컴퓨터 시스템은 MC 68000과 Z80 시스템을 계급적으로 구성하였으며, 부동 소숫점 연산 및 인터럽트 기능을 효율적으로 사용하였으며, 주변장치를 이용한 제반 하드웨어가 설계 제작되었다. 또한 궤적 계획 및 추적 알고리즘이 제시되고, 사용자의 편의를 위한 여러 명령어의 사용 및 교시 반복 동작을 위한 Jog-mode 운전이 가능한 모니터 프로그램이 소프트웨어로 구현되었다.

추후 연구 과제로서 원호 보간 방법과 로봇트 다이내믹스를 고려한 적용 제어기법의 적용에 의한 정밀 서어보 제어의 방법이 진행될 것이다.

참 考 文 獻

- [1] M. Brady, *Robot Motion: Planning and Control*. MIT Press, 1982.
- [2] R. P. Paul. *Robot Manipulator*. MIT Press, 1982.
- [3] 최영규, "산업용 로봇트 KAISEM 1의 연속 경로 제어에 관한 연구" KAIST 석사 논문, 1982.
- [4] 김형일, *A Study on a μ -Processor Based PWM DC Motor Velocity Control* 1983. A KAIST 석사 논문, 1983.
- [5] 김지홍, "연속 경로 추적 가능한 PWM 디지털 로봇트 제어기에 대한 연구" KAIST 석사 논문, 1984.
- [6] 오영석, "산업용 로봇트의 실시간 궤적 계획에 관한 연구" KAIST 석사 논문, 1984.
- [7] Jacob. Tal, "Design and Analysis of PWM Amplifier for DC Servo System" *IEEE Trans. on IA* vol. 23, no. 1, 1976.
- [8] W. Mcmurry, *Selection of Snubber and Clamps to Optimize of Design of Transistor Switching Converter* *IEEE Trans. IA*, IA-16, July, 1980.
- [9] Motorola manual *MC 68000 Course Note*, 1980.
- [10] Prolog manual "STD product Z-80 single board computer"
- [11] James L. Kuester, *Optimization Techniques with Fortran*. McGraw Hill, 1973.*