

서보 모터의 디지털 제어기 설계에 관한 연구

李斗馥*, 洪彦植**, 崔洪圭***, 蔡東奎****

A Study on the Design of a Digital Controller for DC Servo Motor

Doo-Bok Lee*, Eon-Sik Hong**, Hong-Kyu Choi***, Dong-Kyu Chae****

ABSTRACT

This paper deals with the design of the digital controller for DC servo motor, and it is implemented for the cartesian coordinate 4-axes manipulator.

A design method of the controller is adopted an algorithm using the digital position locked loop(DPLL) method and the linear PID control for the smooth motion.

To simplify the hardware configuration of control system, 8279 keyboard/display controller, Z-80 CTC counter and 8255 PPI are used. Therefore the design method to control each motor as real-time is presented.

To show effectiveness of the design, the PWM circuit and frequency/voltage converter are applied for the velocity control of robot system.

When the proposed controller is applied to the 4-axes manipulator, it reveals that the error probabilities of X,Y and Z axis as 0.033%, 0.023% and 0.028% respectively.

*,** 弘益工業專門大學 電子科 教授

*** 弘益工業專門大學 電氣科 副教授

**** 弘益工業專門大學 電氣科 謙任教授

1. 서 론

최근 마이크로 전자공학의 급속한 발달과 고성능 마이크로프로세서의 출현으로 NC 공작기계와 산업용 로봇을 중심으로한 공장자동화 기기와 각종 사무자동화 기기 등의 기능이 향상되고, 보다 고급화 되어 가고 있다. 따라서 이들 산업용 및 사무용 자동화 기기에 필수적으로 사용되는 스텝모터와 서보모터에 대한 다각적인 연구가 요구되는 것이다. 특히 지능로봇이나 반도체 제조 공정에 사용되는 모터는 초정밀 제어를 요하므로 종래의 스텝모터로서는 어려움이 있으므로 점차 서보모터로 전환되어 가고 있는 실정이다. 또한 이 서보모터의 속도제어 기술은 종래의 아날로그 방식을 탈피하여 디지털 제어방식을 채택하고 있다.^(1,2)

본 연구에서는 산업용 로봇에 많이 사용되고 있는 직류 서보모터를 제어함에 있어 Z-80 마이크로프로세서 시스템과 약간의 주변기기를 사용하여 설계하고, 이를 직교좌표형 4축 로봇에 적용시켜 실험하고자 한다.

제어기의 구성에서 디지털 서보시스템의 안정화를 위하여 속도 피드백으로 사용되는 타코미터(Tachometer) 대신에 F/V(Frequency/Voltage) 컨버터를 사용하고 엔코더 출력펄스를 CTC 카운터로 받아들여 위치검출을 한다.

직류 서보모터의 구동회로 구성에서는 펄스폭 변환(PWM) 방식을 채택하여 종래의 전기자 전압 공급방식에서의 발열현상을 제거하고, 전력소모를 감소시키며 대용량 부하에도 사용할 수 있도록 설계한다.^(3,4)

한편 매니플레이터 끝점의 위치가 정지시 오버슈트(overshoot)를 갖지 않도록 하기 위하여 PID(proportional integration derivative) 제어 알고리즘을 수행시키고⁽⁵⁾, 전류증폭기에 의하여 가속시의 모터 전류를 제한하도록 함으로써 급격한 가감속 때의 댐핑(damping)기능을 갖도록 설계한다.

2. 마이크로프로세서에 의한 직류모터 제어방식

마이크로프로세서에 의한 직류 서보모터 제어에 대한 디지털 서보시스템을 그림 1에 나타내었다.

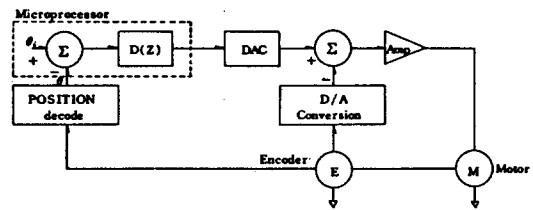


Fig.1 Digital servo system

이 서보 시스템의 프로세서에서는 출력의 피드백에 의하여 미리 계획한 궤적과 실제의 궤적을 비교하여 그 차를 감소시키기 위하여 오차에 일정한 이득을 곱하는 PID 제어 알고리즘을 수행시킨다. 그림 2는 이에대한 블록선도를 나타낸 것이다.

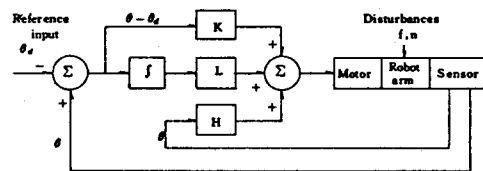


Fig.2 Block diagram of PID control system

여기서 K는 위치이득, L은 적분이득, θ_d 는 기준입력이다. 이때 비례 위치제어는 응답속도를 증가시키기 위한 것이고, 적분제어는 정상상태 오차가 0이 되게 하고, 미분제어는 안정도를 향상시키며 진동을 방지하기 위한 것이다.

그림 1에서 프로세서가 지시한 입력이 급격하게 변하면 정상 편차가 너무 크게 되어 보

간 연산한 값대로 작동하지 못하므로 그림 3의 실선과 같은 사다리꼴의 속도 profile을 설정한다. 즉 가감속시에 일정한 기울기로 전압을 가감속하여 가속시에 발생하는 overshoot나 감속시의 토크(torque), 관성(inertia), 마찰(friction) 등에 대처하도록 한다.

이때 점선은 실제로 모터의 회전운동에서 나타나는 곡선으로 디지털 위치 비교 검출(DPLL; digital position locked loop)방식을 적용하여 입력 지시값(실선)과 엔코더 출력에 의한 실제 이동거리(점선)가 일치되도록 조정한다.⁽¹⁰⁾

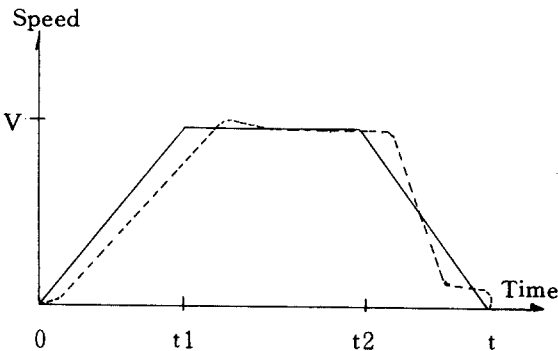


Fig.3 Velocity-time tracking curve in DPLL method

3. 제어시스템 하드웨어 구성

3.1 직류모터 제어시스템 구성

마이크로프로세서에 의한 직류모터의 제어시스템을 그림 4와 같이 구성하였다.

그림 4에서 보면 키보드를 통해서 입력된 명령은 표시장치에 나타나고, 요구하는 명령이나 데이터를 입력시킨 후 각 모드별 실행명령에 따라 마이크로프로세서에서 실시간 제어에 필요한 데이터를 정해진 기억장소에 저장시킨다.

Z-80 CPU는 매순간마다 각 축에 대한 이동속도와 현재 위치를 계산하여 지시하고, 동시에 각 축이 이동될 때의 거리를 계속 입력

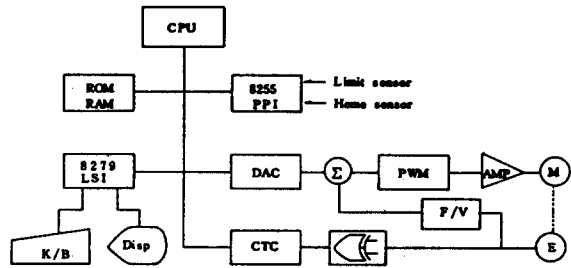


Fig.4 Block diagram of microprocessor control system

하여 실시간으로 처리해야 한다. 따라서 본 설계에서는 키보드의 표시장치를 8279 LSI로 제어하고, 각 모터축의 엔코더 출력 펄스는 CTC로 계수하도록 설계하였다.

또한 CPU에서 계산된 출력값이 D/A 변환기를 통과하여 F/V값과 비교되어 PWM 증폭기에 입력되며, 이 신호에 따라 증폭된 신호가 모터에 전달된다.

로봇트 팔의 각 축상에 있는 리미트 센서와 홈 센서로부터의 신호 입력을 받기 위하여 8255 PPI(programmable peripheral interface)를 사용하였다.⁽⁶⁾

한편 모터축의 엔코더(encoder) 출력인 위치 제환 신호는 XOR의 체배회로를 거쳐 CTC에 입력된다. 이 신호는 동시에 F/V 컨버터에 입력되어 속도 제어용 타코미터(tachometer)의 사용을 생략하였다.

3.2 엔코더와 F/V 컨버터

모터의 회전 속도 및 위치 검출용 엔코더의 출력에 XOR 게이트를 연결하여 펄스당 제어 거리를 1/2로 감소시켜 분해능을 두배로 하였다.

또한 엔코더 펄스를 이용한 속도제어를 위하여 그림 5와 같이 F/V 컨버터 회로를 설계하였다.

그림 5에서 LM331에 의하여 주파수의 높낮이에 따라 전압이 변하게 되고, 주파수 대 출력전압의 비는 R_1, R_f, R_i 및 C_i 에 따라 조

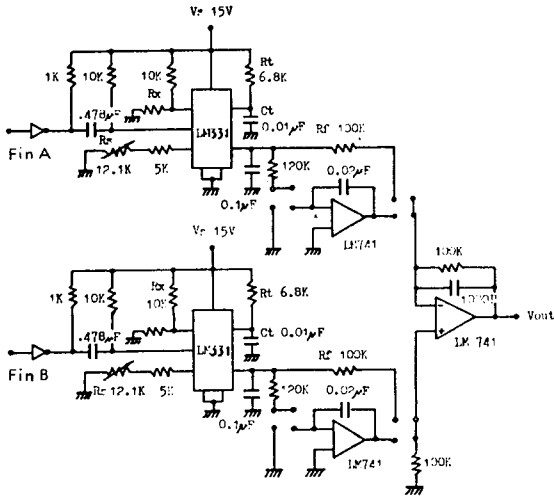


Fig.5 Frequency-Voltage converter circuit

정된다.

이때 F/V의 출력전압 Vout는 다음 식으로 계산된다.

$$V_{out} = -2.09 \times f_{in} \times \frac{R_f}{R_s} \times R_t C_t [V] \dots\dots\dots(1)$$

3.3 직류모터 구동이론

직류모터의 토오크는 전기자 코일 전류와 비례관계에 있으므로 식(2), 식(3)으로 나타낸다.

$$T = K_t \times I \dots\dots\dots(2)$$

$$V = L \times \frac{dI}{dt} + R \times I + E_q \dots\dots\dots(3)$$

여기서,

- T : 토오크 (N/m)
- L : 인덕턴스 (H)
- V : 단자전압 (V)
- K_t : 토오크 상수
- E_q : 내부유기전압 [V]
- I : 전기자 전류 [A]

이다.

또 각속도는 토오크와 반비례 관계가 있다. 즉,

$$\omega_{n1} = \frac{1}{K_e} \times V \dots\dots\dots(4)$$

$$T_{gs} = K_t \times \frac{V}{R} \dots\dots\dots(5)$$

여기서,

- ω_{n1} : 각속도 (rpm/sec)
- K_e : 모터전압 상수
- T_{gs} : 정지시 발생 토오크 (N/m)
- K_t : 토오크 상수

따라서 단자전압이 높을수록 무부하 속도와 정지 토오크가 크다. 식(4)에서와 같이 단자전압으로 속도를 제어하면 식(5)와 같이 토오크가 발생한다. 이때 문제점은 전력소비가 지속적인 점이다. 따라서 스위칭 파워 구동회로를 사용하여 전력소모와 발열현상을 감소시킬 수 있는 PWM(pulse width modulation)방식을 이용한다.^(3,4)

즉 PWM방식은 최대 펄스전압이 일정주기로 모터에 가해지도록 한 것이며, 이때 펄스폭을 가변시켜 모터의 속도를 제어하는 것이다.

3.4 PWM회로 및 증폭기

PWM회로는 신호발생회로, 모터의 정·역전을 원활히 하기 위한 블랭킹(blanking)회로 및 PWM 증폭회로로 구분되며 설계된 회로의 블럭선도는 그림 6과 같다. 이 블럭선도는 그림 4의 D/A 컨버터 후단으로 이어져 모터에 연결된다.

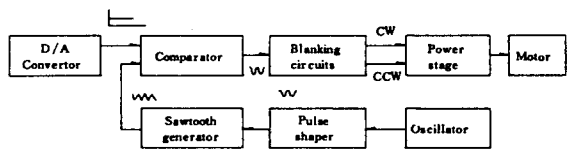
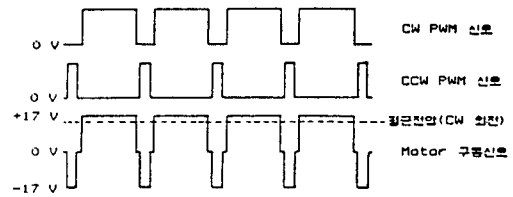


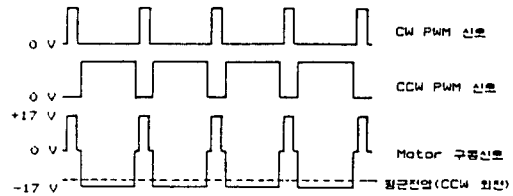
Fig.6 Block diagram of PWM circuit

PWM 신호발생 회로는 1KHz 의 발전기와 튜니파 발생기 및 비교기로 구성되고, 블랭킹 회로는 모터의 정회전과 역회전이 가능하도록 하고, 정·역회전 신호사이의 간격을 RC회로로 지연시켜 단락상태를 방지한다. 정회전(CW), 역회전(CCW) 출력신호와 그에따른 모터 구동신호는 그림 7 과 같다. 그림 7 (a) 는 정회전때의 출력이고, 그림 7 (b)는 역회전때의 출력을 나타내며 점선은 평균전압을 나타낸 것이다.

PWM회로의 동작파형을 오실로스코프(HP 1741A)로 촬영하여 그림 8 에 실었다. 그림 8 (a)는 모터가 무부하에서 정회전시의 신호파형이고, 사진 8 (b)는 역회전시의 신호파형으로 두

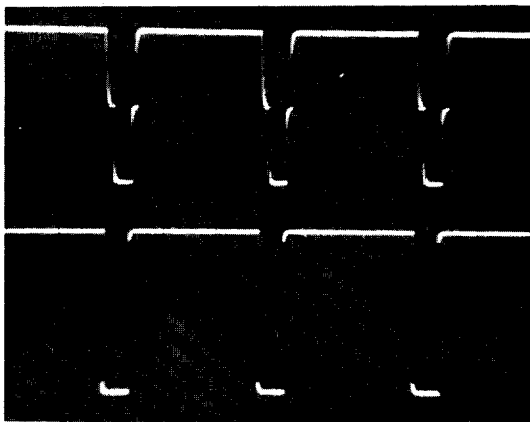


(a) CW 회전 PWM 파형

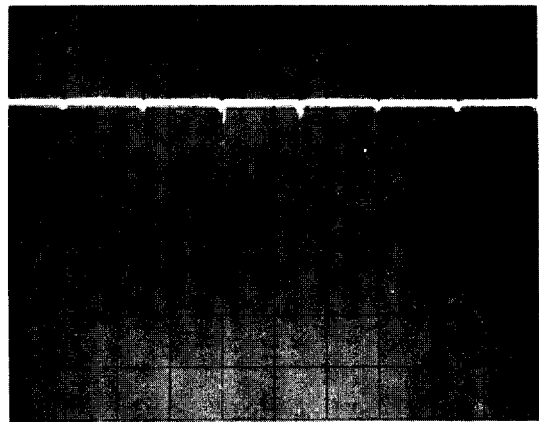


(b) CCW 회전 PWM 파형

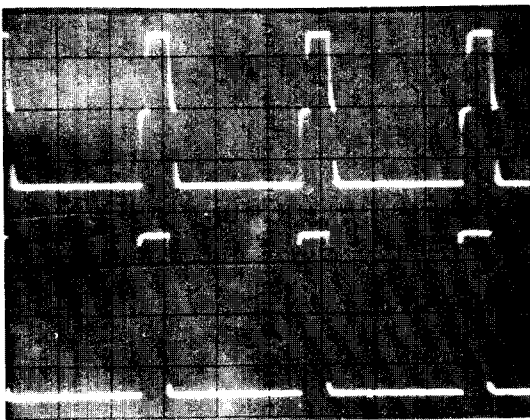
Fig.7 Output signal of blanking circuit



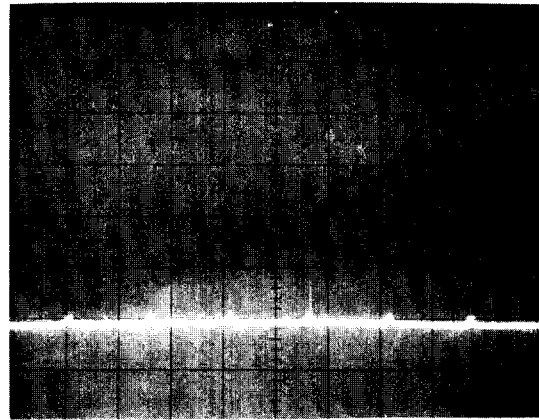
(a) CW파형 (1msec/Div, 1V/Div)



(c) 부하시 CW 파형 (0.5msec/Div, 1V/Div)



(b) CCW파형 (1msec/Div, 1V/Div)



(d) 부하시 CCW 파형 (0.5msec/Div, 1V/Div)

Fig.8 PWM output signal wave form

가 있다.

제어판넬의 arrow 키로 다시 이동시켜 ENT 키로 위치를 지정하도록 설계하였다. 그림 12 는 그에 대한 흐름도이다.

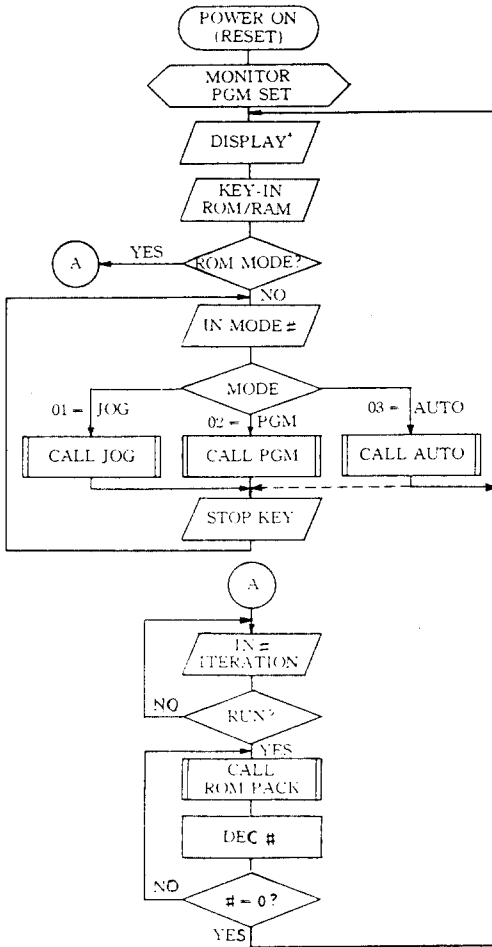


Fig. 11 Flowchart of the overall system

4.2 JOG 모드 서브프로그램

JOG 모드에서는 각 로봇트 축을 arrow 키에 의하여 직접 이동시키고, 그 위치에 해당하는 데이터를 지정하여 프로그램 작성에 활용할 수 있게 준비한다. 즉 BLOCK 키를 사용하여 임의의 위치를 표시하고 그 지점의 위치 데이터 값을 POS 키를 누른 다음 해당 메모리에 기억시킨다. POS 키를 누르지 않으면

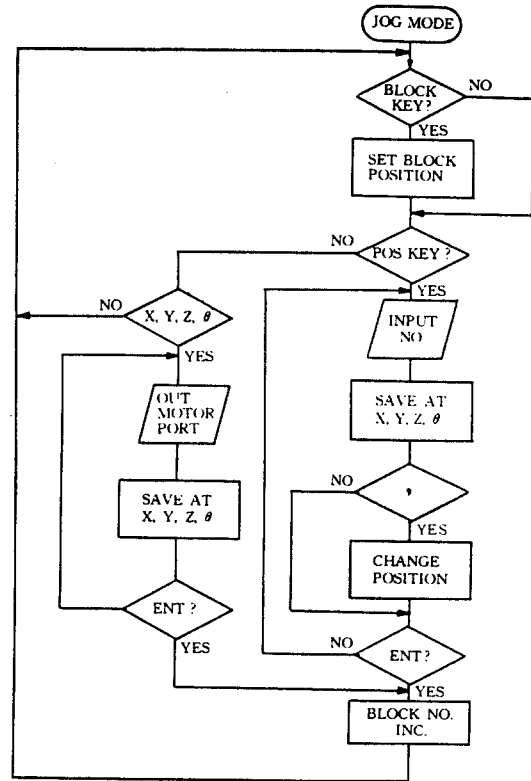


Fig. 12 Flowchart of the JOG mode subroutine

4.3 PGM 모드 서브프로그램

PGM 모드는 그림 13 과 같은 명령 형식에 따라 동작 프로그램을 작성하여 인터프리터로 해석하는 모드로서 그 흐름도를 그림 14 에 나타내었다. 프로그램의 작성은 키보드상의 명령

- (1) BLOCK * * CODE * * DATA * * * * * ENT
- (2) BLOCK * * CODE * * ENT
- (3) CODE * * DATA * * * * * ENT
- (4) CODE * * ENT

Fig. 13 Instruction format for programming in PGM mode

키인 BLOCK, CODE, DATA 및 ENT 키를 사용하여 작성하고 해당 어드레스의 block position 에 LC 카운터를 set 한다.

된 프로그램은 RUN 키로 수행시킬 수 있도록 하였다.

4.4 AUTO 모드 서브프로그램

그림 15는 AUTO 모드 처리 과정에 대한 흐름도이다. 이 모드는 RAM과 ROM 모드 구분되며 ROM 모드가 선택되면 준비된 ROM pack으로 부터의 프로그램을 실행횟수를 지정한 후 수행시킨다.

RAM 모드에서는 PGM 모드에서 미리 작성된 프로그램을 해당 블럭번호로 부터 NEXT

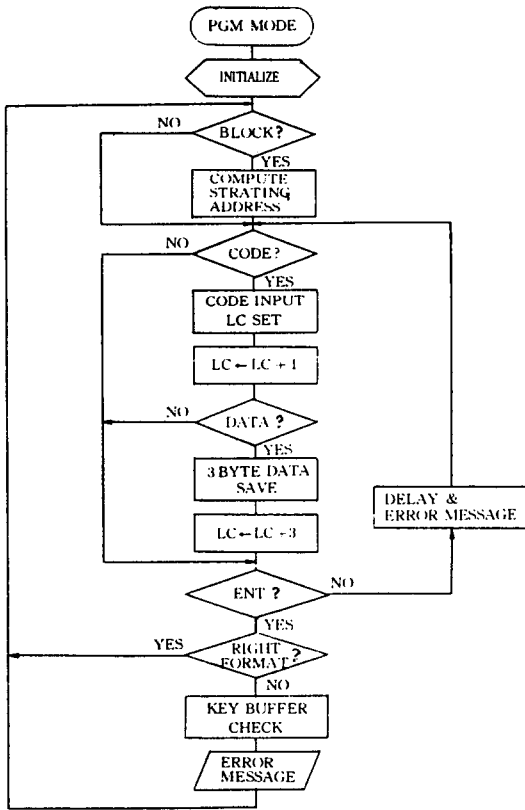


Fig.14 Flowchart of the PGM mode subroutine

이때 BLOCK은 명령어 시작 subroutine 이고, CODE는 각 명령마다 정해진 명령코드 이고, DATA는 각 명령코드에 따라 정해진 데이터 값이다.⁽¹¹⁾ 모든 명령어는 ENT 키가 입력되면 해석기의 기억장치에 저장된다.

작성된 프로그램의 해석이 끝나면 일시적으로 저장된 키버퍼의 내용은 사용자 메모리(user memory) 영역에 저장되고, 미리 설정한 working 메모리 상태를 수정한 후 다음 명령을 기다린다. 이렇게 하여 사용자 메모리에 지정

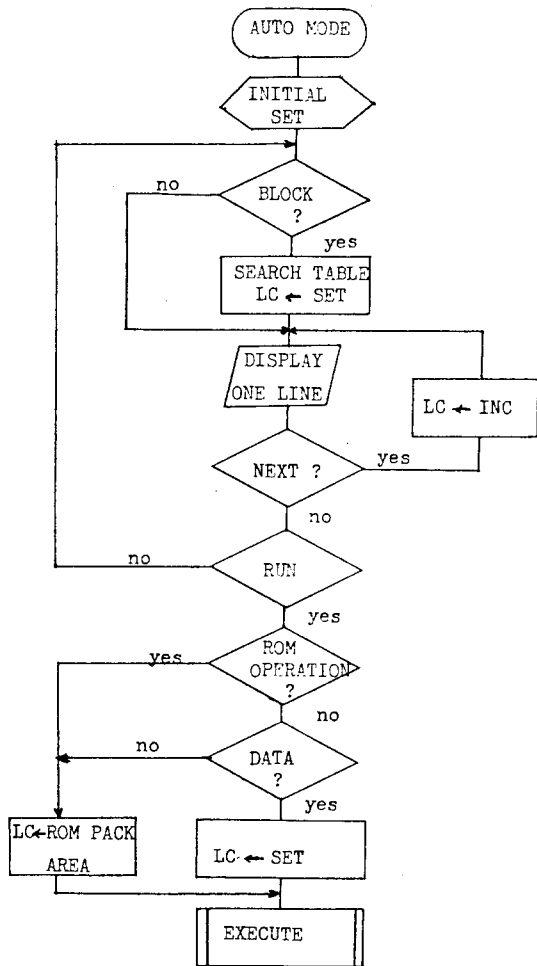


Fig.15 Flowchart of the AUTO mode subroutine

키에 의하여 표시장치로 확인한 후 DATA 키로 반복횟수의 지정과 함께 실행시키도록 하였다.

5. 실험결과 및 고찰

앞에서 설계한 제어기의 정밀성을 확인하기 위하여 표1과 같은 규격을 갖는 직교좌표형 로봇 팔에 적용실험 하였다.

Tab.1 Specifications of the robot arm

NO	ITEM	내용
1	DIMENSION	길이(L) : Max 1080mm 폭(W) : max 550mm 높이(H) : max 350mm
2	동작 범위	X축 : 450mm Y축 : 90mm Z축 : 60mm θ 축 : $-180 \sim +180$ 도
3	최대 속도	X축 : 200mm/sec (2400rpm) Y축 : 50mm/sec (1510rpm) Z축 : 75mm/sec (1510rpm) θ 축 : 720/sec (400pps)
4	동작시 분해능	X축 : max 0.01mm (500p/r) Y축 : max 0.040mm (500p/r) Z축 : max 0.006mm (500p/r) θ 축 : max 1.8도 (200p/r)
5	위치 반복 정도	X축 : max 0.01mm Y축 : max 0.004mm Z축 : max 0.006mm θ 축 : max 1.8도
6	중량	X축 : 12kg Y축 : 8kg Z축 : 3kg θ 축 : 0.3kg

X축 모터를 부하가 걸린 상태에서 최적속도인 1800rpm(150mm/sec)으로 이동시켰을 때의 속도 profile은 그림16과 같이 나타났다. 여기서 곡선1은 open loop로 제어했을 경우이고, 곡선2는 DPLL 방식을 적용했을 경우를 보여준 것이다. 이의 측정을 위하여는 로봇 팔의 이동시 F/V 컨버터 출력 데이터

를 A/D 컨버터로 변환시켜 RAM 메모리에 저장한 후 프린터를 사용하여 그래픽 하였다.

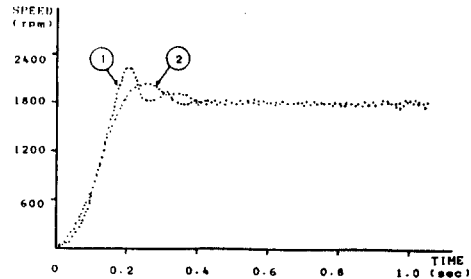


Fig.16 Velocity profile of X-axis

또한 설계된 제어기의 복합 제어를 위하여 그림17과 같은 운동을 수행시켰다. 이것은 3차원 운동 형태로서 직교좌표의 X, Y축 운동과 \otimes 와 \odot 로 Z축 운동이 수행되고 직선 보간에 의하여 원래의 위치로 복귀하도록 되어 있다.

여기서 X, Y, Z축 공간상의 펄스당 이동거리는 각각 $10 \mu m$, $4 \mu m$, $6 \mu m$ 이므로 (표1참조) X축의 2cm 이동에는 2000개의 펄스가 필요하고, Y축 3cm 이동에는 7500개의 펄스, Z축 3cm는 5000개의 펄스 공급이 필요하다.

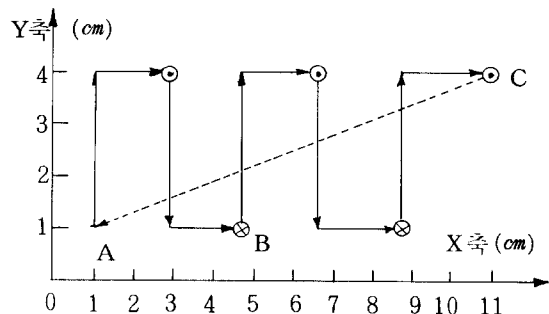


Fig.17 Example of the robot motion

표2에는 로봇 팔의 운동시의 각 축에 대한 이동시간, 이동편차거리 및 편차 발생 확률을 나타내었고, 사진1은 설계된 서보모터의 제

어장치 전경을, 사진2는 본 연구에서 수행한 실험장치의 전경을 나타낸 것이다.

Tab.2 Measured values of the errors for robot arm

로봇트 관 절	이 동 거 리 (mm)	최 적 속 도 (mm/sec)	이 동 시 간 (sec)	이 동 편 차 거 리 (mm)	편 차 발 생 확 률 (%)
X	400	150	3.4	0.13	0.033
Y	80	40	2.8	0.018	0.023
Z	50	45	1.8	0.014	0.028

Photo.1 Servo motor controller

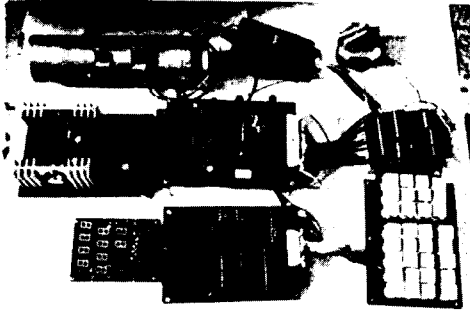
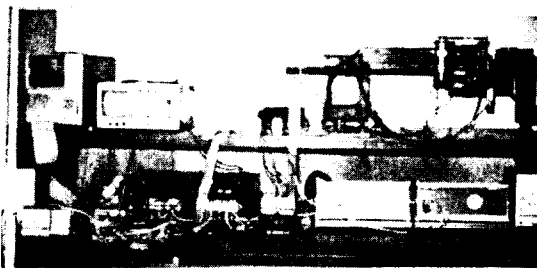


Photo.2 Implemented overall system



6. 결 론

본 논문에서는 산업용 로봇트나 정밀제어기에 활용될 직류 서보 모터의 제어를 위한 디지털 제어기를 설계하였으며 이를 직교좌표형

4 축 로봇트 팔에 적용 실험하였다.

서보모터의 정밀제어를 위하여 DPLL 방식을 채택함으로써 미리 계획된 경로에 대하여 추종제어를 가능하게 하였고, 서보계의 안정화를 위하여 속도 피드백을 위하여 F/V 컨버터를 사용하였다.

또한 PID 제어 알고리즘을 수행시킴으로써 정상상태 오차를 줄이고 진동을 감소시켜 안정도를 향상시켰으며, 서보모터의 효율적인 구동을 위하여 PWM 회로를 설계하여 사용하였다.

설계된 제어기를 직교좌표형 4 축 로봇트팔에 적용시켜 실험하였는 바 만족할만한 속도 profile을 얻었으며, 각 축상의 이동 오차확률은 0.032% 이하로 되어 제어기의 정밀성을 확인하였다.

< 참고 문헌 >

1. J.B. Plant, S.J. Jorna, and Y.T. chan: "Micro-processor control of Position or speed of an SCR D.C. Motor Drive.," IEEE Trans. Ind. Electron Contr. Instrum., Vol.IECI-27, pp.228-234, 1980.
2. Eric h. Stelzer, Randy h. Moss: "A micro-computer-based Control System for a Three- Joint Robot Arm.," IEEE Micro, pp.22-40, June, 1985.
3. Jacob. Tal, "Design and Analysis of PWM Amplifier for DC Servo System" IEEE Trans. on IA Vol.23, No.1, 1976.
4. Richard E. Wainwright: "A microcomputer-based Model Robot System with Pulse-Width Modulation Control." pp.7-21, Feb., 1985.
5. J.H. Aylor, R.L. Ramney and G. Cook, "Design and Application of a Microprocessor PID Predictor Controller," IEEE Trans. Ind. Elect. Contr. Instr., Vol.IECI-27, pp.133-137, Aug. 1980.
6. Goldsbrough, P., "Microcomputer Interfacing Using the 8255 PPI chip with Experiments," Howard W. Sams & Co., 1979.

7. K. Takase et al, "A Structured approach to Robot Programming Trans. on and Teaching," IEEE, SMC, SMC-11(4), pp. 274-289, 1981.
 8. Brady, Hollerbach, Johnson, Lozano-Perez, Mason, "Robot Motion Planning and Control", MIT, pp.19-23, 1982.
 9. Tomas Lozanno-Perez, "Robot Programming", Proceeding of the IEEE, Vol.71, No.7, pp.821-841, July 1983.
 10. 이두복 "A Study on the Design of a Programmable Controller for Industrial Robots", 연세대학교 대학원 박사학위논문, 1986.
 11. 김용득, 이두복 "A Design of the Robot Control System Using Microprocessor", 한국정밀공학회, 추계학술대회 논문집, pp.45-47, 1985.
-