

2축 서보시스템을 위한 위치제어장치 설계

장 석 호* · 김 기 택** · 김 형 중***

Design of a Position Controller for Two Axis Servo System

Seok-Ho Chang, Gi-Taek Kim, Hyoung-Joong Kim

ABSTRACT

In this study, we design a position controller for two axis servo system. The position controller performs numerical control(NC) to DC or AC servo motor or step motor, and also has a digital input/output sequence capability. The control program composed of position and sequence command, which is called channel, is programmed easily and user-interactively. And it is interpreted and the straight line and arc position command is interpolated.

We develop the Z80 microprocessor based system and the software with assembly and C language, and also PC based graphic simulator for the debugging and educational purposes.

1. 序 論

서보모터를 이용한 위치제어의 예로는 NC 공작기계, 로봇 등이 있으며 2축용으로 XY 테이블이 많이 사용된다. 회전장치로는 AC 및 DC 서보모터를 사용하고 구동장치로서 서보드라이버가 있다. 서보드라이버는 전력증폭기와 속도제어 기능을 수행한다. 이러한 서보장치를 이용하여 NC와 같은 위치제어를 수행하기 위해서는 위치제어장치를 부가하

여야 한다. 즉 그림 1과 같이 기능별로 모듈화되어 위치제어 혹은 위치결정기능을 수행한다. 상위계통은 이동할 위치와 시간 등을 결정한다.

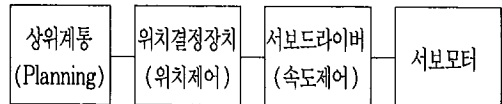


그림 1. 위치제어 시스템

종래에 그림 1의 전체 시스템이 하나로 집중되어 있었으나 마이크로프로세서 기술의 발달로 각 기능별로 마이크로프로세서를 기본으로 하는 디지털 방식으로 모듈화되어 고장시 해당 부분만 교체하여 지속적인 운전이 가능

* 강원대학교 전자공학과 석사과정

** 강원대학교 제어계측공학과 전임강사

*** 강원대학교 제어계측공학과 조교수

하게 되어 신뢰성이 많이 개선되었다. 또한 디지털 방식으로 프로그래밍을 용이하게 하여 서보시스템에 대한 전문지식이 없는 사용자도 쉽게 사용할 수 있게 되었다. 즉 사용자는 어떤 시간에 어떤 조건에서 어느 위치로 이동하게 할 것인가를 결정하고 여러가지 파라미터를 설정하는 것으로 충분하게 되며 쉽게 변경할 수 있다. 상위계통과의 통신기능을 갖추고 있어 상위계통에서 프로그램을 다운로드하여 다양한 동작을 가능하게 하여 FMS(Flexible Manufacturing System) 등의 생산자동화의 중요한 요소로 동작하게 된다.

위치제어장치의 하드웨어는 크게 CPU부분과 입출력부분 및 프로그램 로더로 나누어진다. 사용자가 작성한 제어 프로그램(채널이라고 함) 및 파라미터는 환전 되더라도 보존되어야 하기 때문에 RAM을 사용해야 하는 경우에는 백업 전지를 내장하여 내용을 보존할 수 있도록 한다. 사용자 인터페이스를 위해 간단한 키보드 장치와 40~50자 정도의 디스플레이 장치가 필요하게 된다. 사용자는 키보드와 디스플레이를 통해 프로그램을 작성하고 파라미터를 변경하게 된다. 입출력 부분은 프로그램 수행을 위한 디지털 접점정보를 읽어들이고 접점정보를 출력하며 XY축 서보모터의 위치이동을 위한 펄스를 발생한다. 이와 함께 운전 상태 및 정보를 표시하기 위한 보조 접점이 준비되어 있다. CPU와 입출력부가 서로 영향을 미치는 것을 막기 위하여 각 입출력 접점은 포토 커플러를 사용하여 CPU부와 전기적으로 절연된다.

소프트웨어는 위치제어장치에서 가장 핵심적인 부분으로 사용자가 정한 프로그램과 파라미터로부터 출력할 속도 패턴을 연산하는 부분이다. 각 소프트웨어는 기능별로 모듈화되어야 한다. 사용자는 각 소프트웨어 모듈을 호출하거나 연결하는 것만을 채널에서 지정하며 각 소프트웨어 모듈에서 필요한 상수를 파라미터로 지정하게 된다. 그 밖에 프로그램과 파라미터를 효율적으로 입력시키고 용이하게

변경시키기 위한 사용자 인터페이스 소프트웨어가 필요하다.

2. 위치제어장치

위치제어장치에는 채널이라고 하는 제어 프로그램이 20개 있으며 각 채널은 50개의 스텝을 가지고 있다. 채널은 프로그램에 해당하고 스텝은 프로그램의 명령어에 해당한다. 각 스텝에서 위치이동명령, 원호이동명령, 디지털 입출력 접점에 관한 명령 등이 있으며 다른 채널을 호출할 수도 있다. 표 1과 표 2에 채널과 스텝의 내용을 보여주고 있다.

표 1. 채널의 내용

내 용	동 작
PEPEAT	반복 횟수 0~99설정 0이면 무한 횟수를 반복
POSITION MODE	ABS : 절대 위치 INC : 상대 위치
STEP	1~50 STEP MODE와 SEQ MODE가 있다.

표 2. 스텝의 내용

명령어	기 능
POSITION	위치 명령 X POS=000000 Y POS=000000
CIRCLE	원호 명령 X POS=000000 Y POS=000000 RADIUS=00000 ROTATION=0
CHANNEL	다른 채널을 서브루틴으로 호출한다. 25중 루프 가능

END	서브루틴일 경우 복귀한다. 메인루틴일 경우 채널수행을 종료한다.
RTN	XY축 원점 복귀
DIMENSION	현재 위치를 수정한다. X DIM=000000 Y DIM=000000
IN	번호의 입력 접점이 SET될 때까지 대기 IN=00(1~12)
NOT IN	번호의 입력 접점이 Clear될 때까지 대기 NOT IN=00(1~12)
OUT	번호의 입력 접점이 Set될 때까지 대기 OUT=00(1~12)
NOT OUT	번호의 입력 접점이 Clear될 때까지 대기 NOT OUT=00(1~12)
IN JMP	번호의 입력 접점이 SET일 때 정한 STEP으로 분기
NOT IN JMP	번호의 입력 접점이 Clear이면 정한 STEP으로 분기
JMP	무조건 분기 STEP=00
TIMER	설정된 시간만큼 대기 TIMER=000.0 sec

위치제어장치의 각 동작 모드는 그림 2와 같다.

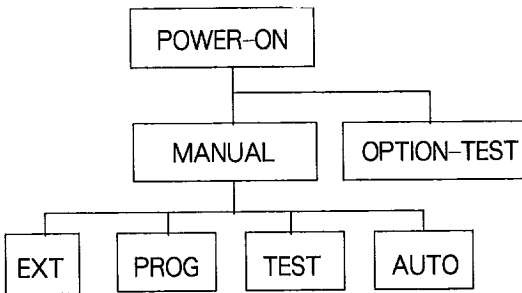


그림 2. 동작 모드

POWER-ON시 DIP SW 설정에 따라 수동운전 모드(MANUAL)와 OPTION-TEST 모드를 선택할 수 있다. OPTION-TEST 모드는 각종 파라미터의 변경과 자기진단 기능

을 수행한다. 다시 수동운전 모드 즉 일반적 운전 모드로 복귀하기 위해서는 DIP SW를 다시 설정하고 전원을 재투입하여야 한다.

수동 모드에서 프로그램 로더의 키보드를 이용하여 수동으로 운전할 수 있으며 MODE 키에 의해 4가지 모드를 실행할 수 있다. 수동운전 모드에서 MODE키를 누름에 따라 로더 LED에 표시된 동작 모드가 순차적으로 변한다. 즉 MODE키가 눌러짐에 따라 EXT → PROG → TEST → AUTO → EXT... 등으로 변하게 되며 각 모드에서 SET키를 누르면 각 모드를 수행시킬 수 있다. 각 모드에서 탈출하여 다시 수동 모드로 복귀하기 위해서는 다시 MODE키를 누르면 된다. 즉, MODE키는 각 모드의 선택과 각 모드로부터의 탈출 기능을 갖는다. 수동 모드에서는 XCW, XCCW, YCW, YCCW 키에 의해 수동으로 X축, Y축을 각각 독립적으로 정회전, 역회전 시킬 수 있다. RTN/ORG 키로 원점으로 이동시킬 수 있으며 OUT 키로 출력접점 신호를 Set시키거나 Clear시킬 수 있다. 이 모드에서는 입력접점 신호에 의해서도 운전시킬 수 있다. PROG 모드에서는 채널과 보조 데이터의 내용을 대화 형식으로 수정할 수 있다.

자동운전 모드에서 운전하기 위해서는 RTN/ORG 키를 눌러 먼저 원점으로 복귀해야 하며 원점복귀가 안된 상태에서 자동운전 모드를 선택하면 경고를 출력한다. 자동운전 모드에는 운전할 채널을 로더에서 선택하는 AUTO 모드와 외부 입력접점에서 선택하는 EXT 모드가 있다. TEST 모드로서 로더로부터 채널을 선택하는 점에서는 AUTO 모드와 같으나, 각 스텝 실행 후 XCW, XCCW, YCW, YCCW 키에 의해 위치를 수정하고 채널내용을 변경하는 Teaching 모드로 동작하게 되는 점이 다르다.

3. 하드웨어

위치제어장치의 하드웨어는 크게 CPU부분

과 입출력부분 및 프로그램 로더로 나누어진 다. CPU부분은 CPU를 중심으로 한 기억장치와 각종 주변장치로 구성된다. CPU부분은 위치결정 기능 즉, 속도 패턴을 연산하고 이동할 위치와 이동할 조건과 시간을 대화 형식으로 입력하고 각종 파라미터를 변경하게 된다. 시스템 프로그램은 EPROM에 저장되며 사용자가 작성한 제어 프로그램 및 파라미터는 단전 되더라도 보존되어야 하기 때문에 RAM을 사용해야 하는 경우에는 백업 전지를 내장하여 내용을 보존할 수 있도록 한다. 메모리는 EPROM, RAM 모두 32K byte용량을 가지고 있다.

사용자 인터페이스로는 간단한 키보드 장치와 40~50자 정도의 디스플레이 장치가 필요하게 된다. 사용자는 키보드와 디스플레이를 통해 프로그램을 작성하고 파라미터를 변경하게 된다. 상위계통과의 통신을 위한 장치가 필요하며 또한 내부 시간을 동기화시키기 위한 타이머의 사용도 필수적이다. 입출력 부분은 프로그램 수행을 위한 디지털 점점 정보를 읽어 들이고 점점정보를 출력한다. CPU와 입출력부가 서로 영향을 미치는 것을 막기 위하여 각 입출력 점점은 포토 커플러를 사용하여 CPU부와 전기적으로 절연된다. 이와 함께 운전 상태 및 정보를 표시하기 위한 보조 점점이 준비되어 있다. 그리고 XY축 서보모터의 위치이동을 위한 펄스를 발생하는 회로가 있다.

3-1. 펄스발생 회로

모터에 정확한 주파수를 갖는 펄스를 정확한 갯수로 발생하기 위해서 8253 카운터를 사용하였다. 펄스는 곧 모터의 속도 및 위치 명령으로서 가감속 패턴을 갖게 된다. 즉 펄스의 주파수 및 갯수가 변하게 되며 이에 대한 연산은 직선보간 및 원호보간 알고리즘에 의하여 계산된다. 주파수가 변하는 펄스를 실시간(Real Time)으로 발생하기 위해서는 2개의 카운터를 사용하여 하나의 카운터가 미

리 계산된 주파수의 펄스를 정해진 갯수만큼 발생하면 자동적으로 다른 카운터로 옮겨 펄스 발생을 계속하게 된다. 하나의 카운터가 펄스를 발생하는 사이에 CPU는 다음 펄스의 주파수와 갯수를 계산하여 동작하고 있지 않은 카운터에 입력시키고 대기하게 된다. 즉 그림 3과 같이 8253 두개가 한 쌍이 되어 모터 하나의 펄스를 만들어 낸다. 두 8253이 서로 맞물려 교대로 펄스를 만들어 내고 있다. 8253의 채널 3개 중 채널 0,1의 2개만이 쓰이고 채널 3은 쓰이지 않는다. 채널 0은 펄스발생을 담당하고, 채널 1은 카운터 역할을 한다. 펄스발생의 기준 클럭(8253의 채널 0의 clk0으로 입력됨)은 2MHz이다. 8253 채널 0의 출력은 채널 1의 clk1으로 입력되어 채널 0의 펄스 갯수를 채널 1가 셀 수 있는 것이다. 펄스발생회로 부분은 8253과 몇개의 게이트들과 RS 플립플롭(74LS279)이 사용되었다. 게이트들과 플립플롭이 상호 작용을 함으로써 한 구간 동안 두개의 8253 중 하나만이 동작하게 해준다.

처음 전원을 넣었을 때는 8253이 어떤 상태에 있을지 예측할 수 없다. 예측 못한 펄스가 발생하고 있을 수도 있으므로 위치제어장치는 되도록 빨리 이 부분을 초기화시켜 펄스 발생을 금지시켜야 한다.

펄스발생 동작이 시작되기 위해 미리 계산된 펄스의 주파수를 위 8253의 채널 0에 펄스 갯수정보를 채널 1에 입력한 후 출력포트 A1을 Low로 하면 위의 RS 플립플롭이 Set되어 채널 0의 gate가 Active되고 펄스가 발생된다. 이 때 아래 8253의 gate는 Deactive되어 펄스가 발생하지 않는다. 다음구간의 펄스 주파수와 갯수를 계산하여 아래 8253에 입력시킨다. 위의 8253 카운터의 펄스가 정해진 갯수만큼 발생하면 채널 1의 out이 High로 되어 RS 플립플롭에 의하여 위의 8253 gate는 Deactive되고 아래 8253 gate는 Active되어 아래 8253에 의해 펄스가 발생한다. 다시 다음 펄스의 주파수와 갯수를

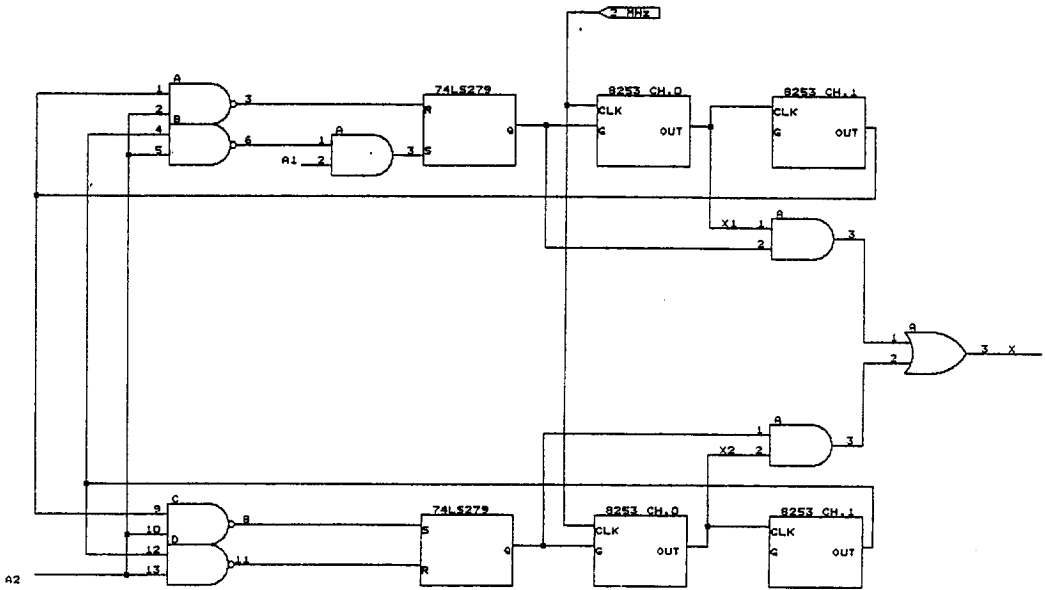


그림 3. 펄스발생 회로

계산하여 위 8253에 입력하고 아래 8253 펄스발생이 끝나기를 대기한다. 이러한 과정으로 2개의 8253으로 교대로 펄스발생을 계속한 후 펄스발생이 종료되면 A2, A3 포트를 이용하여 8253의 gate가 Active되는 것을 금지시킨다. 위 아래의 8253 펄스 X1, X2는 gate 신호와 AND, OR 로직으로 X축 펄스 X를 발생시킨다. Y축에도 X축과 동일한 회

로가 있으며 회로의 동작을 그림 4에 보여주고 있다. 그림 4에서 보면 X축과 Y축이 서로 다른 주파수로 가감속이 되면서 펄스를 발생하는 것을 볼 수 있다. 그림 3의 회로에 의해 처음에는 X1, Y1만 펄스가 발생하고 X2, Y2로 교대되고 있으며 이펄스들이 합쳐져서 완전한 펄스 X와 Y가 되는 것을 볼 수 있다.

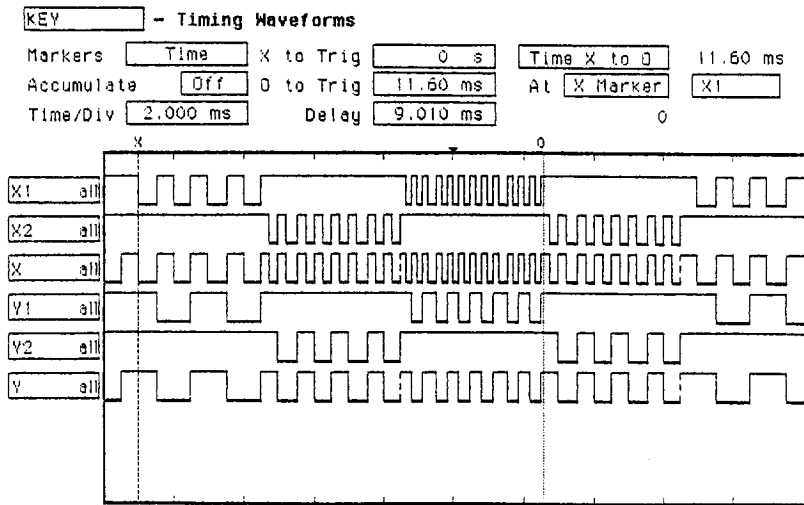


그림 4. 펄스발생 타이밍도

3-2. 프로그램 로더 회로

프로그램 로더의 목적은 대화형식(user-interactive)으로 채널 및 파라미터 내용을 입력하고 수정하기 위함이다. 사용자는 로더의 키를 통하여 위치제어장치를 프로그램할 수 있다. 또 채널이 수행되는 동안 위치제어장치의 상태를 액정 디스플레이(LCD)의 정보로써 알 수도 있다. 로더와 연결되는 모든 신호는 포토커플러에 의해 절연되고 있다. 회로를 간소화하기 위해 8개의 입출력 포트 핀을 사용하였으며 이중 4개는 양방향성의 데이터 버스처럼 사용하였고 4개는 키, 액정 디스플레이, 부저 및 LED 구동용 어드레스 및 제어 기능을 사용하였다. 액정 디스플레이는 4개의 데이터 버스를 사용하는 모드로 사용되었다.

4. 소프트웨어

위치제어장치를 동작시키기 위하여 소프트웨어는 어셈블리 언어와 C 언어를 사용하였다. 어셈블리 언어는 하드웨어를 핸들링하는 기본적인 루틴을 제공하고 거의 대부분의 알고리즘은 C 언어를 이용하여 구현하였다. C 언어로 구성된 프로그램을 디버깅하기 위하여 IBM PC에서 운용될 수 있는 그래픽 시뮬레이터 프로그램을 Turbo C로 작성하였다. 이는 Turbo C의 디버깅 기능을 이용하여 디버깅을 용이하게 하고 특히 직선보간 및 원호보간 기능을 눈으로 확인하기 위해서이다. 이 시뮬레이터 프로그램에서 IBM PC와 위치제어장치와의 하드웨어상의 차이를 흡수하면 거의 그대로 위치제어장치에 그대로 이용할 수 있다. 즉 직접 하드웨어에 관여하지 않는 소프트웨어는 공통으로 사용할 수 있다. 하드웨어의 차이는 키 입력과 디스플레이 및 실제 펄스발생부 등이 이에 해당된다. 예를 들어 IBM PC에서는 직선보간 및 원호보간에서의 펄스발생명령을 화면상에 그래픽으로 처리하지만 위치결정장치에서는 8253타이머를 사용하여 펄스를 발생하게 된다.

어셈블리 프로그램은 하드웨어를 핸들링하는 루틴(이 루틴을 C 프로그램에서 기본 함수로 호출된다.) 외에 프로그램의 실행 환경은 만들고 C의 main 함수를 호출하는 start-up루틴이 있다. 어셈블리 프로그램의 기본 루틴은 디지털 집점 입출력 루틴, 부저 제어 루틴, 키보드 입력, 액정 디스플레이 루틴 및 XY축 서보모터에 분배할 펄스를 발생하는 루틴 등으로 구성되어 있다.

C 프로그램은 어셈블리 기본 루틴을 이용하여 각 채널과 스텝의 내용을 입력하고 입력된 내용을 해석하여 수행하는 루틴으로 되어 있다.

4-1. Start-up 루틴

컴퓨터 시스템에서 C 프로그램을 수행시키기 전에 start-up 루틴을 수행시켜야 한다. 이 루틴은 어셈블리 프로그램으로 되어 있으며 C 프로그램을 실행시키기 위한 환경을 만들어 준다. 즉, 스택 영역을 결정하고 C 프로그램의 시작인 main 함수를 호출한다. 이 외에 기본적인 입출력(주로 키보드 입력과 디스플레이 출력) 루틴을 만들어 주고 전역 변수 영역을 초기화시켜준다.

C 프로그램에서 스택영역은 매우 중요하게 사용된다. 일반적으로 스택 영역은 어셈블리 프로그램에서 서브루틴을 호출할 때 현재의 프로그램 카운터를 스택에 일시 저장하여 서브루틴에서 리턴할 때 스택에서 꺼내 원래 프로그램을 수행하는 용도로 많이 사용된다. 즉, 리턴 어드레스를 기억하게 된다. C 프로그램에서는 함수간의 전달값(argument)을 패싱하거나 자동 변수(auto variable 혹은 local variable)를 생성하는 영역으로 스택이 사용된다. Z80에서는 스택이 낮은 번지로 증가하기 때문에 RAM의 가장 큰 번지를 스택 포인터로 정하게 된다. C 프로그램에서 사용되는 전역 변수로 초기화되는 변수는 C 프로그램이 시작하기 전에 초기화되어야 한다. 변수이기 때문에 읽기, 쓰기 할 수 있는 메모리

즉, RAM에 설정되며 초기화 루틴에서 C 프로그램이 시작하기 전에 블럭 전송 명령을 통해 ROM에서 RAM으로 복사된다. 초기화되지 않은 전역변수는 0으로 초기화 된다.

준비가 다되면 C 프로그램을 호출하게 된다. C 프로그램에서 끝나고 리턴되면 일반적으로 프로그램을 종료 시킨다. 어셈블리와 C 프로그램에서 함수나 변수는 밑줄을 붙여 사용하기 때문에 C 프로그램에서 main 함수는 어셈블리 루틴에서 _main이 된다.

```
call _main
halt
```

4-2. C 프로그램

시뮬레이터 프로그램은 IBM PC의 Turbo

C로 작성된 것이며 동작상태를 화면상태를 화면상에 그래픽으로 표시한다. 시뮬레이터 프로그램에서 하드웨어와 직접 관련된 부분이 어셈블리 프로그램으로 바뀌어 Z80용 프로그램이 된다.

시뮬레이터가 기동하면 그림 5와 같은 화면을 보여준다. 왼쪽 위의 박스 내부는 위치 결정장치에서 로더의 액정 디스플레이와 각 모드에 해당되는 정보를 디스플레이한다. 오른쪽의 큰 박스를 위치결정장치에서 출력한 XY 펄스에 의해 위치 이동을 그래픽으로 보여주고 있다. 점선은 각각 X축 Y축의 원점을 나타내고 있으며 그림 5에서 실선으로 위치를 이동한 예를 보여주고 있다.

C 프로그램의 시작함수로서 main 함수는 그래픽 화면을 위한 준비작업을 하고 그림 5의 화면을 그리는 함수를 호출한다. 이후 수

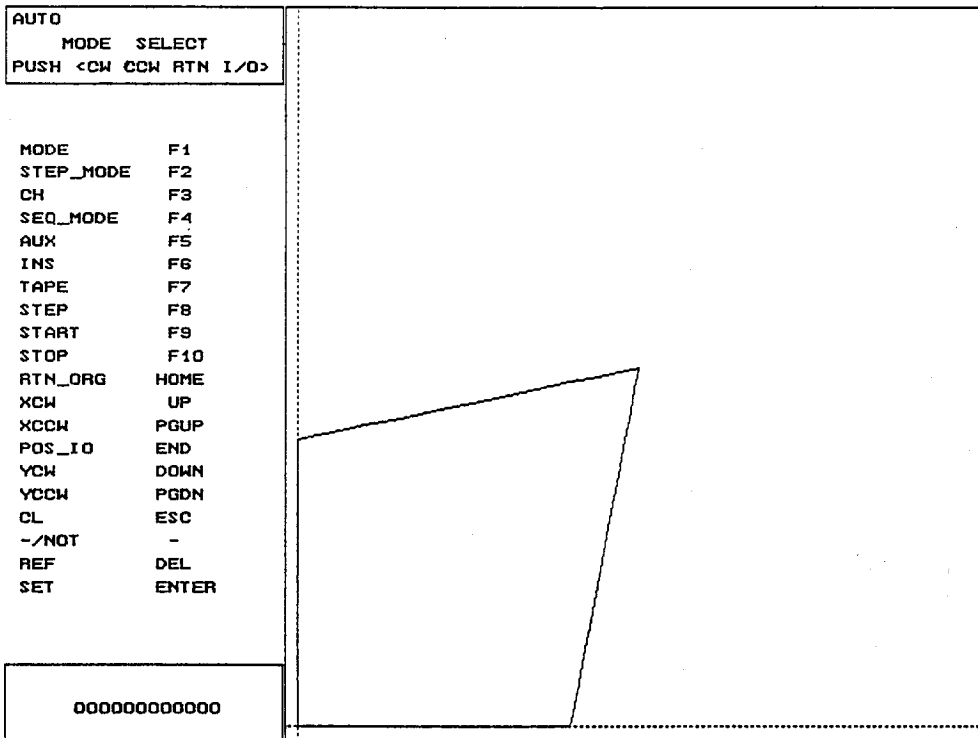


그림 5. 시뮬레이터의 화면

동 모드로 진입하여 MODE키가 눌러짐에 따라 ext, test, auto의 4가지 모드를 디스플레이한다.

수동 모드에서 XCW, XCCW, YCW, YCCW 키를 누르면 수동운전인 manual 함수가 수행된다. SET키를 누르게 되면 MODE키에 의해 정해진 ext, prog, test, auto함수를 수행하고 다시 수동 모드로 복귀하여 프로그램의 출구가 없는 무한 루프가 된다. auto나 test함수가 수행되면 수행할 채널과 스텝을 결정하고 각 스텝의 내용을 interpret 함수로 해석하여 각 기능을 수행한다. 위치제어장치에서 가장 중요한 기능인 직선보간을 위한 position 함수와 원호보간을 위한 circle 함수에 대해 알아보기로 한다.

4-3. position 함수

2축 위치결정 시스템의 경우 각 축의 속도 패턴은 직선보간이나 원호보간에 의하여 발생한다. 예를 들어 직선보간의 경우 위치명령이 그림 6(a)와 같이 P1에서 P2로 주어지는 경우 많이 움직이는 쪽(그림 6(a)에서 X축)을 정해 속도패턴을 정한 후 다른 쪽(그림 6(a)에서 Y축)도 같은 가감속, 정속 구간을 갖도록 그림 6(b)와 같이 속도패턴을 발생한다. dx가 dy보다 큰 경우를 예를 들어 설명한다. dy가 큰 경우는 Y축에 기준을 맞추면 된다. 실제로는 dx와 dy의 단순비교가 아니고 속도 정격에 대한 비를 조사해야 한다. 즉 dx가 dy보다 작아도 x축의 속도 정격이 낮아 최대 속도 정격으로 x축이 더 오랜 시간이 소요되면 x축을 기준으로 삼아야 한다. 이와 같이 할 경우 X축과 Y축 모두 정격을 넘지 않는 속도패턴이 가해지기 때문에 선형영역에서 동작해 직선적인 이동이 가능하다. 그림 6(b)에서 X축이 많이 움직이는 방향이므로 X축 이동에 대한 대형 속도패턴을 결정한 후 Y축으로도 같은 가감속 구간을 정하게 되며 X, Y축 이동거리의 비 $T(=\tan\theta)$ 로 Y축의 최대속도와 가감속율이 보정된다.

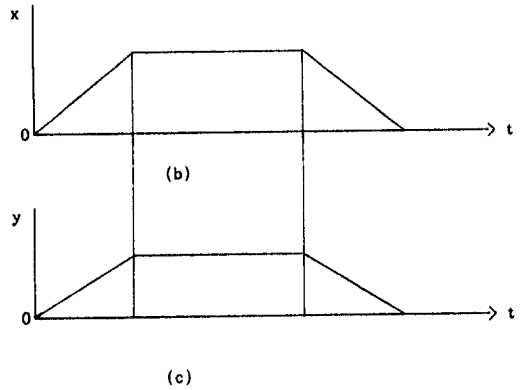
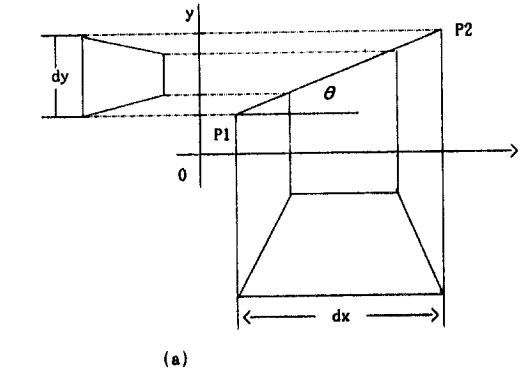


그림 6. 직선보간

위치결정장치는 위치명령으로 부터 속도명령을 발생하는 것으로 PLC나 One Loop controller에서의 위치제어와 유사한 기능을 수행한다. 그러나 위치결정장치는 피드백 제어를 수행하지 않고 단지 시간에 따라 미리 정해진 속도명령패턴을 발생하는 개루우프 방식이다. 따라서 위치결정장치에서 본 서보드라이버와 서보모터는 스텝모터와 같이 동작하는 것처럼 된다. 즉 속도 명령을 발생한 후 실제 움직인 거리를 확인하지 않고 다음 위치로 이동하게 되어 위치오차가 누적될 수도 있다. 따라서 정밀한 위치제어를 위해서는 속도패턴을 출력한 후 실제 움직인 위치에 대한 정보를 피드백 받아 위치보상을 수행하는 것이 정확도를 개선시키는 방법이 된다.

X축과 Y축의 속도 정격으로 Jog 속도와

Max 속도가 있다. Jog 속도는 정지 후 운전 되는 속도이며 Max 속도는 정격최대 속도로 각각 파라미터에서 정해진다. 위치제어장치에서 각 속도는 펄스의 주파수로 표현되며 8253 카운터에 의해 발생되기 때문에 그림 6 처럼 연속적인 변화를 할 수 없다. 따라서 일정시간 동안 일정한 주파수로 펄스를 발생하고 주파수를 높여 발생하는 것을 반복하게 된다.

움직여야 되는 거리 즉 총 펄스 수의 크기에 따라 그림 7처럼 3가지 직선보간 모드가 있다. 즉 거리가 짧을 경우 그림 7(a)처럼 가

감속 없이 Jog 속도로 계속 움직이게 되며 거리가 어느 이상 길게 되면 그림 7(b)와 같이 가감속을 하지만 최대속도에는 도달하지 못하는 경우가 있다. 또 움직이는 거리가 매우 길면 그림 7(c)와 같이 최대 속도 즉 최대 주파수의 펄스를 발생하는 구간이 있게 되며 그림 7의 (a), (b), (c) 모두 중심에서 대칭 형태를 이루고 있다. 그림 7 (b), (c)에서 정지하기 전의 Jog 속도 구간이 다른 구간에 비해 긴 것은 위치 에러를 보정할 시간적 여유를 주기 위해서이다.

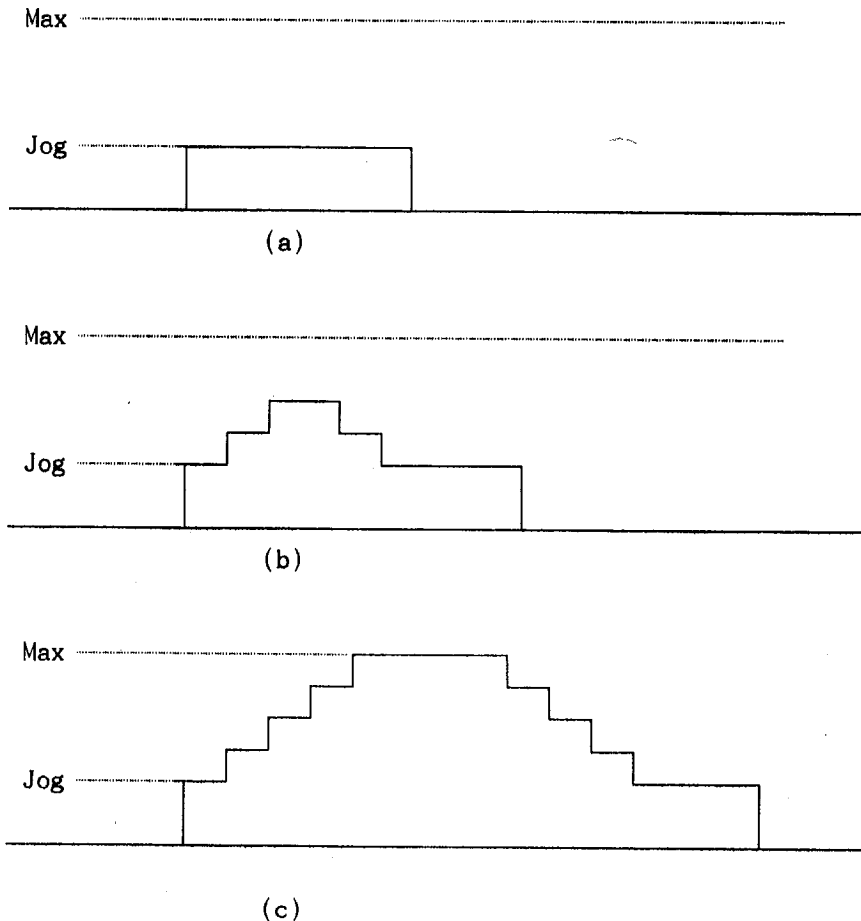


그림 7. 직선보간 모드

4-4. circle 함수

원호를 그리기 위해서는 원의 중심과 원호의 양 끝점 및 반경이 주어져야 한다. 그러나 실제로는 원호의 두 끝점을 주고 반경을 주는 것이 일반적이며 이 정보로부터 원의 중심을 계산해 내게 된다. 직선보간과 마찬가지로 원호보간에서도 연속적으로 속도를 변화시킬 수 없기 때문에 원호를 몇 개의 구간으로 나누어 직선으로 움직이게 된다. 몇 개의 구간으로 나누느냐는 원호 알고리즘에서 요구하는 중요한 데이터로서 사용자가 정하게 하였다. 같은 양끝점과 반경을 갖는 원은 그림 8과 같이 A, B, C, D의 4가지 경우가 생긴다. 이 4가지 경우의 선택도 사용자가 결정하게 된다. 만약 시작점과 끝점이 일치하게 되면 완전한 원을 그리게 된다.

본 원호보간 알고리즘에서는 다음의 방법을 사용하였다.⁴⁾

$$x = x - \frac{\phi}{2}y$$

$$y = y + \frac{\phi}{2}x$$

$$x = x - \frac{\phi}{2}y$$

여기서 ϕ 는 한번에 이동하는 각도이다.

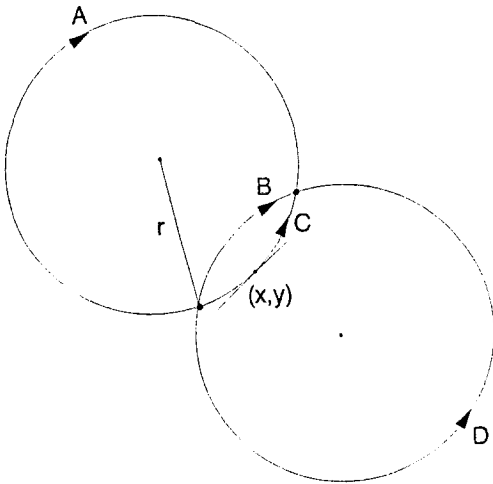


그림 8. 원호의 4가지 경우

5. 결 론

2축 서보시스템을 위한 위치제어장치를 설계, 제작하였다. 하드웨어는 Z80마이크로 프로세서를 기본으로 하여 전체 시스템을 설계하였다. 하드웨어의 특징으로는 산업용 설비이기 때문에 외부와의 모든 신호 연결은 포토 커플러를 사용하여 전기적으로 절연시켰다. 8253 카운터를 이용하여 정밀한 펄스 발생을 위한 회로를 제안하였으며 프로그램과 파라미터를 입력하기 위한 프로그램 로더는 키보드와 액정 디스플레이(LCD)를 이용하였다.

소프트웨어는 어셈블리와 C언어를 사용하였다. 거의 모든 기능 및 알고리즘은 C 프로그램으로 구현되었으며 어셈블리 프로그램은 start-up 루틴과 하드웨어를 핸들링 하는 기본적인 디바이스 드라이버 루틴과 펄스 발생과 같이 엄밀한 타이밍을 요구하는 루틴에 사용되었다. 위치제어 장치의 프로그램 로더의 매우 복잡한 과정을 C 프로그램을 이용하여 간결하게 기술할 수 있었으며 디버깅을 위하여 IBM PC용 그래픽 시뮬레이터를 개발하였다. 이 그래픽 시뮬레이터는 직선보간 및 원호보간 알고리즘을 확인하는데 사용될 수 있으며 위치제어 장치의 교육용으로도 사용할 수 있다.

참 고 문 헌

1. PIMCO-IIA 사용설명서, 일본 OHM사, 1990
2. Z80 C Compiler, 2500AD Software Inc., 1990
3. Z80 Assembler, 2500AD Software Inc., 1990
4. L. R. Neal and M. L. V. Pitteway, "Yet MORE Circle Generators", The Computer Journal, Vol. 33, No. 5, 1990