



전송라인으로 대체하는 것 만으로 간단히 CNC 화가 이루어진다. 따라서 기존의 NC 공작기계의 MCU 의 내부구조에는 아무런 수정도 필요없게 된다. 다만 전송라인에 의해 MCU 에 연결된 컴퓨터가 종이 테이프 reader 의 역할을 대신하게 되므로, NC 프로그램은 컴퓨터의 주 메모리 혹은 보조 기억장치에 기억되어, MCU 가 필요로 할 때 전송라인을 통해 전달되게 된다.

이러한 프로그램의 전송을 위해서는 MCU 와 종이 테이프 reader 와의 신호전송 방식을 그대로 모방하는 하드웨어가 필요하게 된다. 즉 MCU 와의 적절한 신호교환을 위해 MCU 가 종이 테이프 reader 에 보내는 신호를 확인하고, 그 신호에 따라 알맞은 정보를 MCU 에 보내주는 기능을 갖는 하드웨어를 두어 컴퓨터와 MCU 간의 신호 전송이 이루어지도록 하여야 한다.

컴퓨터에 기억되어 있는 NC 프로그램은 널리 퍼져있는 소프트웨어들을 이용하여 쉽게 작성, 편집, 수정이 가능하고, 또한 보조 기억 장치를 이용하면 자유곡선 혹은 자유곡면의 가공의 경우와 같은 긴 프로그램도 쉽게 저장할 수가 있다. 가령 1 키로 바이트의 내부 메모리에 저장되어 있는 NC 프로그램의 경우, 이 프로그램을 종이 테이프에 천공하면 2.58 m 의 길이가 되므로, 프로그램의 보관이 훨씬 간편하고, 특히 보조 기억장치를 이용하는 경우에는, 360 키로 바이트의 플로피 디스크 한 장에 저장되는 프로그램은 종이 테이프 931 m 에 해당한다. 또한 텍스트 에디터 ( text editor ) 등의 구입이 쉬운 소프트웨어를 이용하면 프로그램의 편집, 복사, 부분 수정 등이 자유롭게 되므로, NC 프로그램의 작성이 용이해진다. 그리고 여러 CAD 시스템으로부터 생성되는 공구 경로 정보 ( cutter location data ) 로부터 NC 프로그램을 하게 되면, 설계에서 제품가공 까지의 일련의 작업의 자동화가 가능해지므로, 생산 자동화를 위한 구체적인 수단이 이러한 BTR 형식의 CNC 에 의해 이루어지게 된다.

### 3. IBM PC 를 이용한 BTR 형식의 CNC 화

본 연구에서는 일본 FANUC 사 에서 제작한 SYSTEM 5M NC 컨트롤러와 IBM PC 를 연결하는 BTR 방식의 CNC 시스템을 구성하여, 2 1/2 차원의 컨트롤을 통해 NC 밀링 머시인을 작동시키고자 한다. 전체 시스템의 구성도는 그림 1. 에 있으며, 그림에서 보는 바와 같이 NC 컨트롤러와 종이 테이프 reader 와의 신호 전송라인 중에서 필요한 부분을 IBM PC 와 연결시킴으로써 NC 프로그램을 컴퓨터에서 컨트롤러로 직접 전송할 수 있게 하여 BTR 방식의 CNC 화를 구현한다.

NC 컨트롤러에 부착되어 있는 종이 테이프 reader 는 LED 를 이용하여 photoelectric 방식에 의해 초당 300 개의 character 를 읽어 들일 수 있게 되어 있다. 따라서 컴퓨터가 대신하여 신호를 전송하려면 초당 300 개의 신호를 보내주는 것이 필요하다. 또한 NC 컨트롤러에 전달되는 신호는 TTL 레벨의 digital 신호이므로, 컴퓨터와의 접속 회로 역시 TTL 레벨의 신호를 만들어

주도록 구성해야 한다.

그리고 NC 컨트롤러는 ISO 코드와 EIA 코드의 두가지 character 를 사용하는데, 본 연구에서는 EIA 코드를 이용하여 NC 프로그램을 전송한다. NC 컨트롤러의 작동에는 기존의 종이 테이프를 사용하는 경우와 BTR 방식의 CNC 시스템의 경우가 차이가 없으므로, MCU 가 프로그램을 필요로 할 때 종이 테이프 reader 대신에 컴퓨터가 작동하게 하여줌으로써 공작기계의 운전이 이루어지게 한다.

NC 컨트롤러와 종이 테이프 reader 사이에 주고 받는 신호들 중에서 BTR 형식의 CNC 시스템 개발을 위해 이용하게 되는 신호는, 컨트롤러로부터 테이프 reader 로 전송되는 FDT 신호와  $\overline{ALM}$  신호, 그리고 테이프 reader 에서 컨트롤러로 전송되는 RDT 신호와 D0 에서 D7 까지의 8 bit 의 데이터 신호가 있다. FDT 신호는 테이프 reader 에서 종이 테이프를 전진시키도록 하는 신호로서, 이 신호는 NC 컨트롤러가 프로그램 데이터를 받아들일 준비가 되어 있다는 ready 신호가 된다. 따라서 이 신호를 모니터 하는 회로를 두어 데이터 전송의 시작 신호로 사용하면 된다. 이 FDT 신호가 주어진 후, 50  $\mu$ sec 후에 데이터들이 안정되게 하고, 테이프 reader 에서 feed hole 의 도착을 알리는 신호인 RDT 신호를 300 에서 350  $\mu$ sec 후에 컨트롤러에 보내면 데이터의 전송이 이루어 지게 된다. 그리고  $\overline{ALM}$  신호는 컨트롤러에서 에러가 발생한 경우 주어지는 신호로서 이 신호를 이용하여 데이터 전송을 중지하게 하도록 한다. 그림 2. 는 이 신호들의 timing chart 이다.

이러한 신호들의 timing 을 이용하여, 데이터 전송을 위해 그림 3. 에 나타난 바와 같은 회로를 둔다. 즉, 데이터 전송은 300 Hz 의 속도로 이루어져야 하므로, 300 Hz 의 pulse 발생 회로를 두어 전체 데이터 전송 속도를 조절하고, 이 회로에서 발생하는 pulse 와 FDT 신호를 AND 시킨 결과를 컴퓨터의 내부 데이터 버스의 D0 bit 에 연결시켜, 컴퓨터에서 데이터 전송의 timing 을 맞출 수 있게 하고, 또한 자동적으로 350  $\mu$ sec 후에 TSP 신호를 컨트롤러에 보낼 수 있는 회로를 두어 FDT 와 TSP 각각의 신호의 timing 은 하드웨어 적으로 이루어 지게 한다.

IBM PC 에는 접속 회로들의 연결을 위해 접속 slot 이 마련 되어 있는데, 이 접속 slot 에 주어져 있는 신호들 중, D0 에서 D7 까지의 데이터 버스 와 A0 에서 A9 까지의 어드레스 버스, 그리고 어드레스 enable 신호인 AEN 과 I/O read 와 I/O write 신호인  $\overline{IOR}$ ,  $\overline{IOW}$  의 신호를 이용하여 컨트롤러에 데이터를 보내는 접속 회로를 만든다. I/O 어드레스는 0280 번지가 데이터 output port 가 되게 하며, 0281 번지는 FDT 와  $\overline{ALM}$  신호를 읽어들이기 위한 input port 가 되게 하고, 0282 번지와 0283 번지는 각각  $\overline{ALM}$  신호와 FDT 신호의 flip-flop 들을 clear 하도록 decode 한다. 그림 4. 는 이러한 접속 회로를 나타낸다.

위에서 설명한 하드웨어의 작동을 위해 IBM PC 에서 사용가능한 언어들 중에서 TURBO PASCAL (Ver.3.0) 을 사용한다. 이 언어는 프로그램의 수행에 편리하도록 화면처리 및 그래픽, 그리고 기계어와의 접속 등의 다양한 기능을 갖고 있으며, 기준 PASCAL 언어가 갖는 모든 기능을 갖추고 있으므로 사용하기가 편리하다. 또한 기본 DOS 에서 수행시킬 수 있는 COM 형태의 화일로 컴파일 가능하므로, 프로그램의 사용자가 언어자체에 대한 지식 없이도 프로그램을 실행할 수 있게 소프트웨어를 개발할 수가 있다. 위에서 설명한 접속 회로를 통한 컨트롤러로의 데이터 전송을 위해 필요한 procedure 들은 IBM PC 의 CPU 인 8088 의 기계어를 직접 PASCAL 언어에 연결시키는 명령어를 이용한다.

전체 프로그램의 구조는, 첫째, 보조 기억 장치 ( 하드 디스크 혹은 플로피 디스켓 ) 에 저장되어 있는 텍스트 화일을 내부 메모리로 읽어 들이는 부분과, 둘째, 내부 메모리에 저장되어 있는 프로그램을 모니터에 디스플레이하여 확인하는 부분, 그리고 셋째로 저장된 프로그램을 NC 컨트롤러로 전송함으로써 NC 밀링 머시인을 작동시키는 부분으로 구성된다. 특히, 컴퓨터의 기억장치에 기억되는 텍스트 화일은 ASCII 코드로 이루어져있고, NC 컨트롤러는 EIA 코드를 인식하므로, 이 두 가지 코드들 사이의 변환이 필요하며, 이를 위한 procedure 가 작성된다.

#### 4. 자유곡선의 자동 프로그래밍

NC 공작기계에 의한 공작물의 가공을 위해서는 우선 NC 컨트롤러에 입력되는 NC 프로그램이 필요하다. 이러한 프로그램을 용이하게 하기 위해 비교적 high level 에 해당하는 언어인 APT 언어가 개발되어 있고, 이러한 언어로 구성된 프로그램으로부터 종이테이프를 천공하기 위해 NC 프로그래머가 있다. 즉 설계된 제품을 가공하기 위해서는 APT 와 같은 high level 언어로 쉽게 프로그램을 하고, 이로부터 공구경로를 얻어내어, NC 컨트롤러가 인식 하여 공작기계를 운전시킬 수 있는 NC 프로그램으로 컴파일하여 그내용을 종이 테이프에 천공하게 된다. 따라서 APT 등의 high level 언어로 자유곡선을 가공하는 프로그램을 작성하려면, 자유곡선의 좌표들 모두 알아내어 이들을 이용하여 프로그램하여야 한다.

본 연구에서는 이러한 자유곡선의 가공을 위해 2 차원 digitizer 를 이용하여 자유곡선에 대한 좌표값을 자동적으로 컴퓨터에 입력시키고, 이 입력된 좌표 정보에 의해 NC 컨트롤러가 NC 밀링 머시인을 작동시킬 수 있도록 NC 프로그램이 자동으로 이루어지게 한다. 이를 위해서는 digitize 된 자유곡선의 정보로부터 NC 프로그램을 작성하기 위한 알고리즘의 개발이 우선 필요하게 된다.

자유곡선의 좌표들 컴퓨터에 입력시키기

위해, 평면 digitizer 를 IBM PC 에 연결시켜 자유곡선을 digitize 하면, 좌표 정보가 자동적으로 컴퓨터에 입력되도록 한다. PC 의 serial communication port 에 연결된 digitizer 에 의해 초당 95 개의 좌표 정보를 컴퓨터에 입력시키게 된다. 이와 같이 좌표 정보가 입력되면, 이들을 이용하여 가공을 위한 NC 프로그램을 작성하게 된다. 본 연구에서 사용된 NC 컨트롤러는 FANUC 의 SYSTEM 5M 으로서, X,Y,Z 축들 중에서 2 축의 동시 제어가 가능한 2 1/2 차원의 컨트롤러이다. NC 프로그램은 원점 설정, 스핀들 스피드 설정, feed rate 결정 등의 기본상태를 setting 하는 부분과, 자유곡선의 정보들 중에서 각 선들의 시작점과 끝점에서 공구를 Z 축 방향으로 움직이는 부분을 제외하고는 point-to-point 방식으로 공구가 좌표를 따라 움직이도록 한다. 여기에서 작성된 NC 프로그램은 텍스트 화일로서 보조 기억 장치에 기억되어 있다가 앞절에서 설명된 CNC 프로그램에 의해 사용되어 공작물을 가공하게 된다.

자유곡선을 digitize 한 좌표 정보를 살펴보면, 결정된 좌표 정보의 수가 너무 많게 되어 실제 가공에는 불필요한 좌표점들이 생기게 된다. 이렇게 좌표 정보의 수가 많아지면 NC 프로그램의 길이가 길어지게 되고, 따라서 메모리의 낭비가 된다. NC 프로그램의 길이를 줄이기 위해, 직선 위에 놓이는 점들은 그 직선의 시작점과 끝점을 제외한 점들을 제외한 나머지 점들을 생략함으로써, 좌표 정보의 갯수를 줄이게 된다. 즉 그림 5. 에서 보는 바와 같이 digitize 된 각 점들을 잇는 선분을 생각하여 각 선분들의 arctangent 값을 구하고, 인접한 선분들의 이 arctangent 값들의 차이가 일정한 범위 안에 들면 이 선분들을 하나의 직선으로 보아, 가운데의 점은 생략하기로 한다. 그림 5. 의 경우, 점  $P_6$  의 경우에  $\theta_5$  와

$\theta_6$  과 의 차이가 일정한 범위 안에 놓이므로  $P_6$  를 생략하고  $P_5$  에서  $P_7$  으로 잇는 직선의 공구 경로를 얻게 된다.

digitizer 로부터 좌표값들을 읽어 들이고, 이 값들을 이용하여 같은 모양의 자유곡선을 가공하기 위한 NC 밀링 머시인 프로그램을 자동으로 만들어 내는 프로그램을 앞 절의 CNC 프로그램의 경우와 마찬가지로 TURBO PASCAL 언어를 이용하여 작성한다. 프로그램의 기능은 위에서 설명한 바와 같이, digitizer 로부터 좌표값을 읽어 들이는 기능과, 이 좌표값들을 직선 선분만을 남기고 좌표값의 수를 줄이는 기능, 줄어든 갯수의 좌표값들을 그래픽 화면에 점과 점을 잇는 선분으로 나타내어 확인하는 기능, 메인 메모리에 기억되어 있는 좌표값 정보를 텍스트 화일로 저장하는 기능, 보조 기억 장치에 저장된 좌표값을 읽어 들이는 기능, 그리고 좌표값 정보로부터 NC 프로그램을 작성하여 CNC 프로그램을 통해 가공 할 수 있도록 보조 기억 장치에 저장하는 기능 등이다. 각각의 기능들은 모두 사용자가 사용하기 편리하도록 메인 메뉴에서 선택할 수 있는 모듈로 구성된다.

5. CNC 시스템의 실제 가공의 한 예

지금 까지 설명한 BTR 방식의 CNC 시스템과 자유곡선의 자동 프로그래밍 소프트웨어를 이용하여 실제로 NC 밀링 머시인을 사용, 자유곡선을 가공한다. 그림 6. 은 가공하려는 자유곡선의 digitize 되는 모습을 나타낸다. 이 자유곡선은 digitize 결과에 의하면 8 개의 곡선으로 구성되며, 각각의 곡선들은 2, 254, 16,9,4, 211, 24, 3 개의 점들로 이루어진다. 그림 7. 에 이 곡선들이 그래픽 화면을 통해 디스플레이 된 모습을 나타내며, 그림 8. 에 이 자유 곡선이 가공되어 있는 모습을 나타낸다.

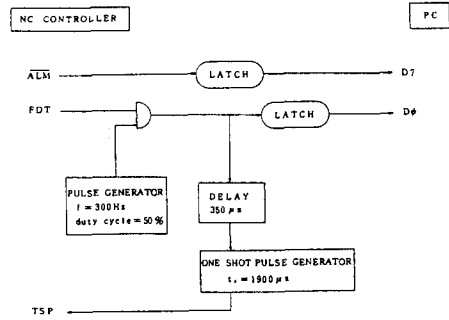


그림 3. 신호 발생 회로

6. 참고 문헌

- "Microprocessor-Based Direct Numerical Control Systems", Crossley, T.R., D. McCartney, Jan. 1979, vol.28, Annals of the CIRP.
- "DNC experience leads to CNC", Apr. 1976, American Machinist.
- Fanuc System P-Model G Operator's Manual
- Fanuc Tape Drill- Model E 보수 설명서
- IBM Technical Reference
- 8088 Assembly Language Programming, Willen, D.D., J.I.Krantz, 1983, Howard W. Sams
- TURBO PASCAL - REFERENCE MANUAL, Borland International.
- TURBO GRAPHIX TOOL BOX, Borland International.

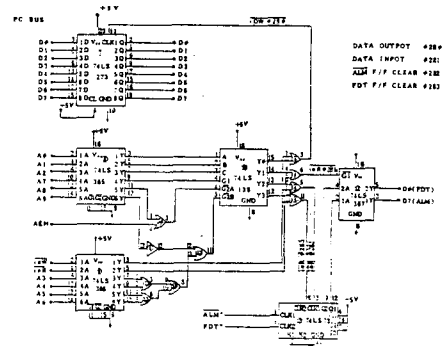


그림 4. 접속 회로도

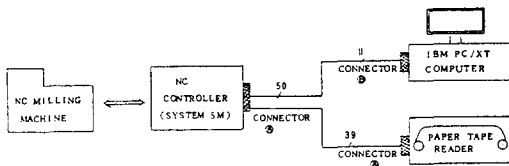


그림 1. CNC 시스템의 구성도

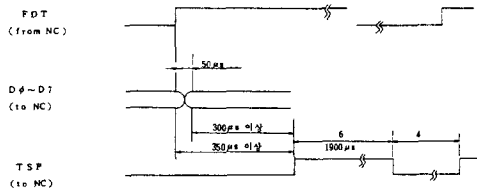


그림 2. 각 신호들의 Timing Chart

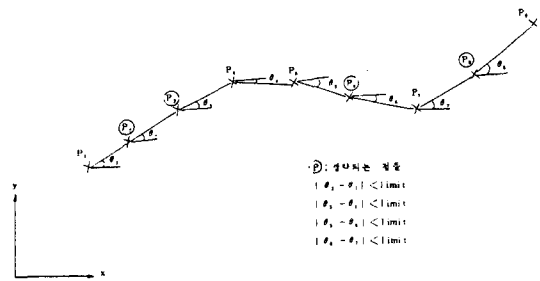


그림 5. digitize 된 곡선의 한 예



그림 6. 자유 곡선을 digitize 하는 모습

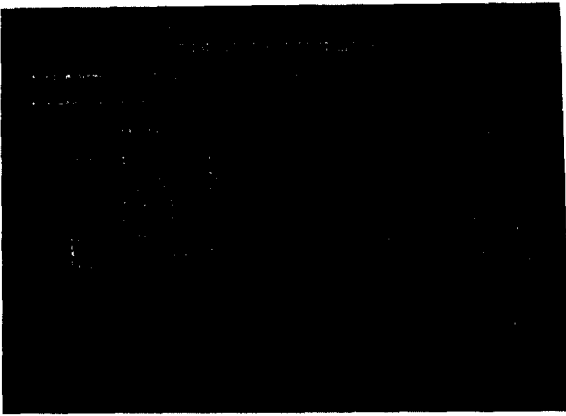


그림 7. 화면에 나타난 곡선의 모습



그림 8. 가공된 공작물의 모습