마이크로프로세서를 이용한 인터폴레이터

論 文
33~2~3

A Microprocessor-Based Interpolator

呂 寅 澤*·盧 台 錫*·李 奉 珍**
(In-Tack Yea · Tae-Seok Nho · Bong-Jin Lee)

Abstract

In this paper we present a microprocessor-based interpolator using algebraic arithmetic method. The interpolator consists of 2900 "bit-slice" microprocessor chips and 0.5K ROMs of 36-bit microprogram memory. The system design is an instuction-data-based architecture with 250ns cycle time. A significant feature of the interpolator is that it has flexibility, very fast interpolation speed of 250K pulses/sec, and performs additional functions simultaneously. Throughout the paper detailed explanations are given as to how one can design the hardware and software, and experimental results are presented.

1. 서론

보간기는 연속경로제어를 하는 수치제어장치와 로봇 등에 널리 쓰여지고 있다. 연속경로제어를 위한 보간방식 중에 동경대학에서 개발된 대수연산방식(Algebraic Arithmetic Method)이 있다.[1] 그런데 실제로 연속경로제어를 할 때에는 보간의 기능뿐만이 아니라 정확한 이송속도제어(Feedrate Control), 가감속기능(Speed Up/Down), 현재위치정보 발생기능 등을 필요로 한다. 지금까지는 이런 부수적인 기능을 보간의 기능과는 별도로 생각하여 실현했으므로 회로가 복잡해졌고, 또한 이를 논리회로 소자(Logic IC)를 이용해서 실현한 경우 신뢰성이 낮았다. 그런데 이 보간기를 Intel 8085 microprocessor를 사용해서 실현해 보았으나 처리속도가 3K pulses/sec 밖에 되지 않아 실제로 사용하는 데에는 문제가 있었다.[2] 본 논문에서는 위의 두 문제를 bipolar bit-slice microprocessors를 사용함으로써 해결하였다.

본논문에서 우리는 2,900계열 bipolar bit-slice microprocessors 를 이용한 효과적인 보간기의 실현에 대해서 서술하고자 한다. 이 보간기의 특징은 다음과 같다.

- 대수연산 방식을 사용
- 부수적인 기능을 보간의 수행과 동시에 수행
- 250K pulses/sec 처리 속도(최소지령 단위가 1 μ/pulse 인 경우 15m/min 의 이송속도에 해당)
- 직선 및 원호보간 만이 아니라 조송, 수동펄스발생기에 의한 이송, Dwell의 기능이 가능

보간기에 대한 상세한 설명은 다음과 같다. 특히 중점을 두어서 서술하려고 하는 바는 하드웨어와 소프트웨어에 대한 실현과정이다. 서론에 이어 2장에서는 전체시스템과 대수연산방식에 대하여 설명하고, 3장에서는 하드웨어 구성에 대한 설명을 하였다. 제4장에서는 소프트웨어 구성에 대한 것과 간단한 2축 동시 제어장치의 구성을 통해 얻은 실험결과를 제시하였고 마지막으로 5장에서 결론을 맺었다.

2. 전체 시스템구성과 대수연산방식

가. 전체 시스템

우리가 원하는 경로제어를 해야 하는 N/C Ma-

*正 會 員 : 韓國機械研究所 研究員
**正 會 員 : 江原大 工大 機械工學科 教授·工博
接受日字 : 1983年 9月23日

nufacturing 시스템에서 이를 위한 명령을 내는 부분은 바로 보간기이다. 그렇기 때문에 보간기는 연속경로제어 장치에 있어서 상당히 중요한 부분이 아닐 수 없다. 전체 Computerized 수치제어 장치에 대한 것은 참고문헌(3)에 있으므로 본 논문에서는 보간기에 대한 설명만을 하고자 한다.

실현한 보간기는 250 K pulses /sec 를 처리하며 부수적인 기능들을 동시에 수행함으로써 하드웨어를 간단화했으며 유연성(flexibility)을 갖게 하였다. 이를 위해서 실시간 수행(real-time execution)을 고려한 효과적인 소프트웨어의 구성이 필요하다. 전체 보간기의 구성은 그림 1과 같다.

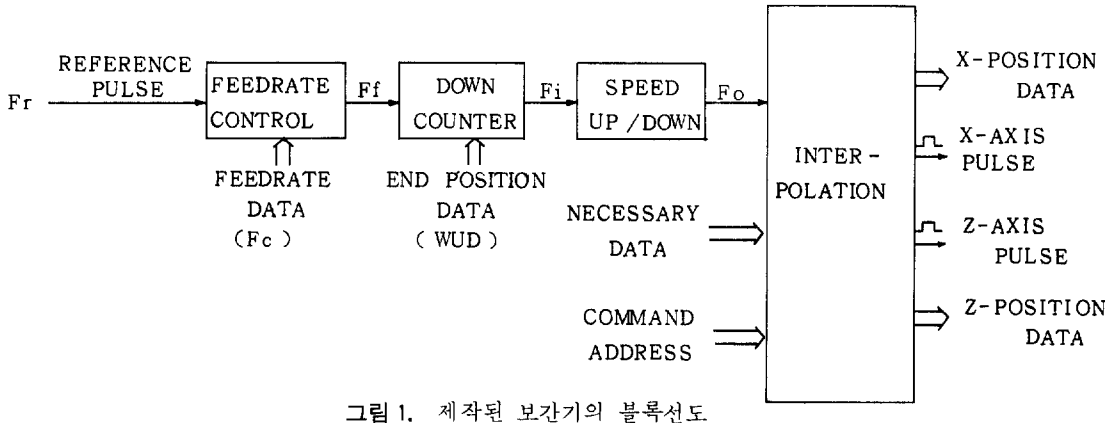

그림 1. 제작된 보간기의 블록선도
Fig. 1. Block diagam of interpolator

그림 1에서 기준펄스(Reference pulse ; Fr)는 먼저 이송속도제어 부분으로 들어가며 이 기준펄스는 N/C 선반의 경우 스핀들 펄스코더로 부터 공급된다. 이송속도 제어 부분은 이 기준펄스를 받아 이송속도를 제어하는데 그 과정은 아래 그림 2와 같다.

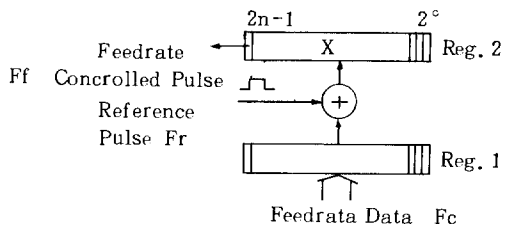

그림 2. 이송속도제어 과정에 대한 블록 선도
Fig. 2. Block diagram of feedrate control process

기준 펄스가 발생할 때마다 reg. 1의 내용을 reg. 2에 더하고 만약 이때 캐리(carry)가 발생했을 경우에는 펄스를 한개 발생시킨다. 이렇게 하면 reg. 1에 설정된 값 (F_c)에 따라서 출력펄스(F_f)의 속도가 제어되며 이는 다음과 같이 수식화된다.

$$X(k+1)=X(k)+F_c,\quad X(0)=0 \qquad (1)$$

(1)을 풀면 $X(k)=k\cdot F_c$ 가 된다. 그러므로 F_f는 Fc에 비례하는 펄스 속도를 갖는다. Down counter 는 속도제어된 펄스(F_f)를 공급받게 된다. Down counter 는 미리 전체 이동거리에 해당하는 데이터(WUD)로 설정되어 있는데, F_f 펄스를 받을 때마다 값을 하나씩 감소시켜, 0이 될 때에는 더 이상의 펄스는 통과시키지 않지만 그전까지는 통과시킨다. 이 통과된 펄스는 가감속 부분으로 공급된다. 이 부분에서는 motor 의 매끄러운 운동을 위해서 F_f 펄스의 속도를 가감속시킨다. 말하자면 motor 기동시에는 속도를 0으로부터 우리가 바라는 속도까지 가속시키고 한 동안 그 속도를 유지한다. 그러나 Down counter 의 내용이 0으로 되는 순간($t=T_d$) 부터 속도를 감속시켜 0으로 만든다. 이 과정은 아래 그림 3으로 도식화 된다.

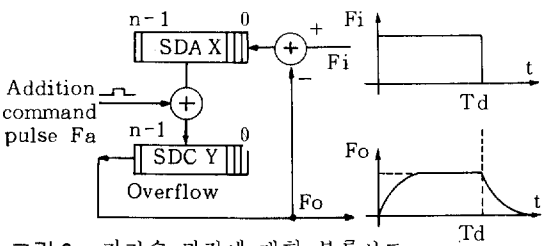

그림 3. 가감속 과정에 대한 블록선도
Fig. 3. Block diagram of speed up /down process

그림 3에서 F_i 펄스가 도달할 때마다, SDA reg. 를 1증가시키고 Addition command pulse 가 감지될 때마다 SDA reg. (X)를 SDC reg. (Y)에 더한다. 이때 SDC에서 overflow 가 발생하면 SDA

reg. 를 하나 감소시킨다. 이를 수식화해 보면,

$dX=(F_i-F_o)\cdot dt$ (1), $X(0)=Y(0)=0$

$dY=F_a\cdot X\cdot dt$ (2)

$F_o\cdot dt=dY/2^n$ (3)

식 (1), (2), (3)으로 부터

$F_o=F_i\cdot\{1-\exp(-F_a\cdot t/2^n)\}$,

$(0\leq t\leq T_d)$ (4)

를 얻는다. Down counter 가 0으로 되는 순간 ($t=T_d$) F_i 는 0이 되므로 감속시의 F_o 는

$F_o=F_o(T_d)\cdot\exp[-F_a\cdot(t-T_d)/2^n]$,

$(T_d<t)$ (5)

로 된다. 마침내 F_o 펄스는 보간기 부분으로 공급된다. 보간부분이 F_o 펄스를 감지하면 이 펄스를 Z축 또는 X축으로 분배할 것인가를 결정함으로써 주어진 일을 수행하게 된다.

나. 대수연산 보간 방식[2)]

1) 직선보간

직선보간을 위해서 필요한 정보는 출발점(Z_s, X_s)과 종점(Z_e, X_e)이다. 그림 4와 같은 경로를 따라 가는 데 있어서, 현재 위치가 실선의 아래에 있는지 위에 있는지를 판단하여 현재 위치가 선의 아래에 있으면 $+X$방향으로 한 step 움직여서 윗방향으로 가고, 위에 있는 경우에는 $+Z$방향으로 가서 밑으로 향하게 된다. 이를 수식화하기 위하여 현재 위치 좌표를 (Z_k, X_i)라 하고 판별식 $D_{k,i}$ 를

$D_{k,i}=Z_e\cdot X_i-X_e\cdot Z_k$ 라 하면

a) $D_{k,i}\geq 0$일때($+Z$방향)

$Z_{k+1}=Z_k+1$

$D_{k+1,i}=D_{k,i}-X_e$

b) $D_{k,i}<0$일때 ($+X$방향)

$X_{i+1}=X_i+1$

$D_{k,i+1}=D_{k,i}+Z_e$ 이다.

2) 원호보간

원호보간을 위해서 필요한 정보도 역시 직선보간의 경우와 마찬가지이다. 그림 5와 같은 경로제어인 경우 현재 위치가 원호의 안쪽에 있는가 바깥쪽에 있는 가를 결정한다. 만약 현재 위치가 원호의 안쪽에 있을 경우에는 다음 step 은 바깥쪽으로 가기 위해 $+X$방향이고 바깥쪽에 있을 때는 안으로 들어 가기 위해서 $-Z$ 방향으로 향한다. 마찬가지로 이를 수식화해 보면

$D_{k,i}=(Z_k^2+X_i^2)-(Z_e^2+X_e^2)$이라 하면

a) $D_{k,i}\geq 0$인 경우 ($-Z$방향)

$Z_{k+1}=Z_k-1$

$D_{k+1,i}=D_{k,i}-(2Z_{k+1}+1)$

b) $D_{k,i}<0$인 경우 ($+X$방향)

$X_{i+1}=X_i+1$

$D_{k,i+1}=D_{k,i}+(2X_i+1)$

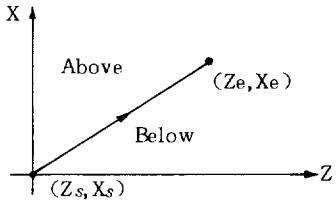

그림 4. 직선보간

Fig. 4. Linear interpolation

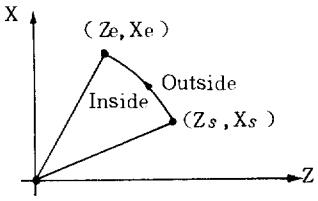

그림 5. 원호보간

Fig. 5. Circular interpolation

3. 하드웨어 설계

가. 하드웨어 구성

보간기의 하드웨어 구성에 대한 간단한 블록 선도가 그림 6에 제시되어 있다. 이 블록선도는 4부분으로 나누어 진다. 첫번째 부분은 마이크로프로그램 제어부분(MCU)이고, 둘째는 중앙프로세싱부분(CPU), 세째는 마이크로프로그램 기억부분(MM) 이고, 네째는 입출력부분(I/O)이다. 모든 마이크로 인스트럭션들은 250n sec 내에 수행된다. 이 시스템의 핵은 Advanced Micro Devices (AMD 2901 bipolar microprocessors 로 구성된 CPU이다. 6개의 2901과 2902 (Look Ahead Carry Generater) 들을 사용하여 24 - bit 연산장치(ALU)를 만들었다. 또한 현재 Z축 및 X축의 위치에 대한 정보는 2901의 Y를 통해 출력 레지스터로 보내 진다. 또한 입력정보는 2901의 D를 통해 받아 들여 진다. 그리고 연산 결과에 대한 상태를 알려주는 carry (BC), Zero (IZ)와 Negative (IS)는 스테이터스 레지스터 (Status Register)에 보내진다. 이 스테이

터스에 대한 서비스를 해주기 위해 본 시스템은 Instruction - Data - Based Architecture 를 갖고 있다. 그림 6에 있어서 I1과 I2가 24 - bit CPU에 연결되어 있다. 여기서 I1과 I2는 tri-state data bus 로 각각 I/O port 를 통해서 main 의 data bus에 연결되어 있다. I1을 통해서는 보간을 하는데 필요한 data (시점좌표, 종점좌표, 이송속도 등)이 공급된다. 들어온 이 data 들은 2901 소자의 16개 레지스터 중에 한 군데로 설정된다. 여기서 CPU는 M4를 통해서 제어된다. 마이크로프로그램 제어부분 (MCU)은 AMD 2910 Sequence Controller, 스테이터스 레지스터, Condition MUX 등으로 구성되어 있다. MCU의 중심인 것은 2910 소자이다. 이 소자는 external condition input ($\overline{CC}$), 12 - bit branch address input (D_{0-11}), 외부의 Jump 주소를 여러 군데 중에서 선별하여 받아 들이기 위한 3개의 제어출력 ($\overline{PL}$, $\overline{Vect}$, $\overline{Map}$)과 4K의 마이크로프로그램 메모리를 관장하기 위한 12 - bit 출력 (Y_{0-11})이 있고 CSC는 2910 Sequencer 를 제어하기 위해 있으며 INIT는 Y_{0-11}을 모두 0으로 만들기 위해서 쓰여졌다.

본 시스템에서는 branch input 의 소스(Source)는 두 군데이며 그 하나는 main에서 공급해 주는것으로 각 명령을 수행하기 위한 마이크로프로그램의 첫번째 주소를 지정해 주기 위한 것이고 나머지는 이 수행에 있어서 자체적으로 Jump를 하는 데에 필요한 마이크로프로그램 내의 branch address 이다. 이들은 각각 $\overline{Map}$과 $\overline{PL}$에 의해서 선택된다. PMSI는 main에서 모든 정보를 보간기에 준 다음 실지로 명령을 수행 하라는 Strobe 신호이고 Ipulse 는 reference pulse (Fr)이다. 또한 $\overline{CCEN} = 0$ 일 때에는 무조건 Jump ($\overline{PL} = 0$)가 행해진다.

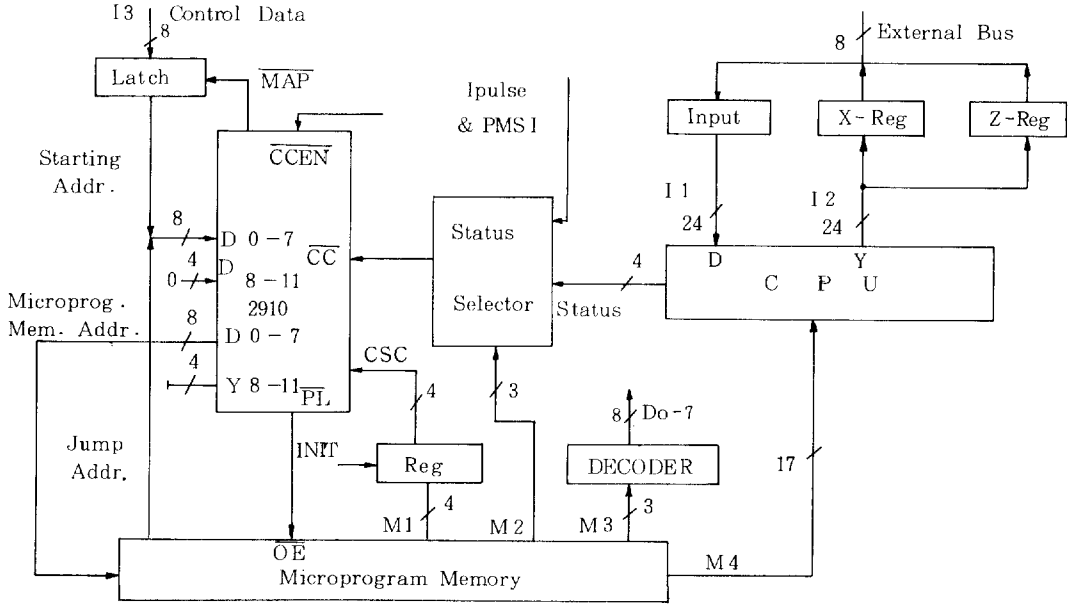

그림 6. 보간기 하드웨어의 구성 블록선도
Fig. 6. Interpolator hardware block diagram

본 시스템은 빠른 시간이 요구되므로 마이크로프로그램 메모리로서 AMD 2727과 AMD 2713 bipolar memory 사용했다. AMD 2727은 출력 레지스터를 가지고 있기 때문에 속도를 빠르게 하기 위한 pipeline technique 을 사용하기에 편리한 메모리 소자이다. 마이크로프로그램 메모리는 36 bits로 구성되어 있으며 이 출력은 clock에 의해 동기되어 전체 시스템을 제어하게 된다.[4] I/O 부분은 두 부분으로 나누어져 있는데, 하나는 펄스발생기 부분이고 다른 하나는 I/O port 이다. 펄스발생기는 Z축 또는 X축 motor 를 움직이기 위한 펄스를 내기도 하고 main 과 교신하는데 필요한 Strobe 신호를 내는 역할을 한다. Decoder 는 필요한 마이크로프로그램 메모리의 bit수를 줄이기 위해서 쓰여진

것으로 이 Decoder 의 출력 신호는 다음과 같은 역할을 한다.

FBSTB : 정보를 읽어 들이기 위한 Strobe 신호

Sample : main 이 현재 위치를 읽기 위한 Strobe 신호

$D0$: 2901 Y출력을 X-reg 에 싣기 위한 신호

$D1$: 2901 Y출력을 Z-reg 에 싣기 위한 신호

$D2$: X축 펄스 출력

$D3$: Z축 펄스 출력　　$D5$: Clear Ipulse

$D4$: 명령완료 신호　　$D6$: Clear PMSI

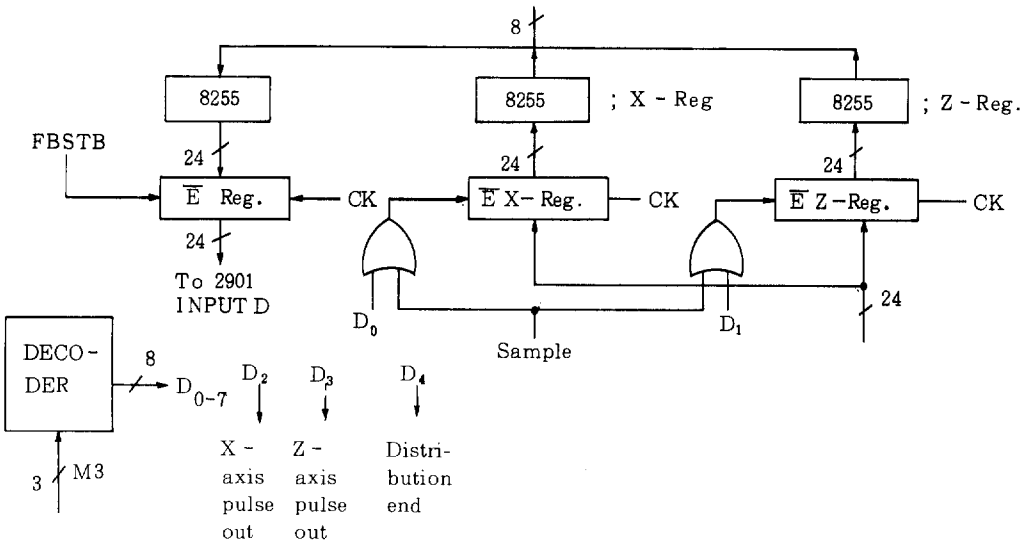

그림 7. I/O unit 의 블록선도.

Fig. 7. I/O unit block diagram

나. Instruction format

마이크로인스트럭션은 아래와 같이 36 빗으로 이루어져 있으며 이들은 CPU, MCU, I/O 부분을 각각 제어한다.

CPU control (17 bits) : M4 ;

1. ALU source (3 bits)
2. ALU function (3 bits)
3. ALU destination (2 bits)
4. ALU address (8 bits)
5. Carry generator (1 bit)

MPU control (16 bits) :

1. Sequencer control (4 bits)
2. Branch address (8 bits)
3. Status selector (3 bits)
4. Unconditional jump control (1 bit)

I/O control (3 bits) : M3

4. 소프트웨어 설계와 실험결과

가. 소프트웨어 설계

보간기의 마이크로프로그램은 3부분으로 나누어져 있다 : 일을 수행하기 위해 필요한 데이터를 받아 들이기 위한 data accept routine , 실지로 보간 명령을 수행하는 부분, 초기상태 결정및 return routine . Power ON 시와 아무런 일도 하지 않을 때에는 시스템은 초기상태의 부분에 있으며 이 곳에서 명령수행 개시신호(PMSI)에 의해서 main 에서 지령된 명령을 수행하기 위한 Jump 를 한다. 그리하여 그곳에서 명령을 수행한다. 모든 명령은 수행이 끝나면 return routine 을 거쳐 초기상태로 되돌아 가서 다음 명령을 기다린다. 그런데 관심이 있는 직선 및 원호보간은 16 Step 이내로 끝난다. 그러므로 Addition Command Frequency (Fa)는 자동적으로 250 KHz 로 결정된다. 다시 말하면 직선및 원호보간에 대한 명령이 하나의 Macro - instruction

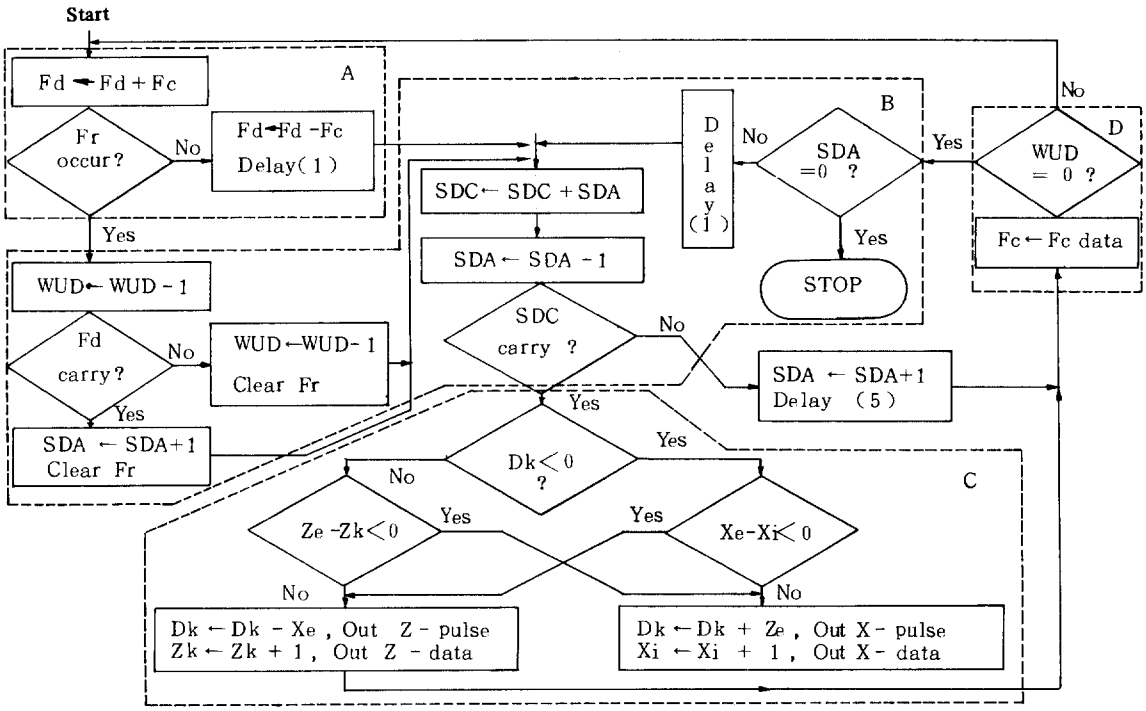

그림 8. 직선보간에 대한 flow-chart

Fig. 8. Linear interpolation flow-chart

이 되고 이를 수행하기 위해서 16 Step 이내의 마이크로인스트럭션이 계속 반복하여 서비스가 끝날 때까지 수행되는 것이다. 그리고 이때에 필요한 데이터는 아래와 같다.

Z_k : Z-axis start position
X_i : X-axis start position
Z_e : Z-axis end position
X_e : X-axis end position
F_c : Feedrate control data
WUD : Whole traverse distance
D_k : Discriminant Value
SDA : } Initial zero for speed Up/Down
SDC : }
F_d : Initial zero for feedrate control

그림 8은 직선보간에 대한 Flow-chart 를 나타내고 있다. A부분에서는 속도제어 (Feedrate Control)에 대한 algorithm 을 수행하고, B부분에서는 가감속을 위한 algorithm 을 행하며, C부분에서는 직선보간에 대한 algorithm 을 수행하여 Z축 및 Y축으로의 펄스분배와 현재 위치정보를 출력하며, D부분은 가감속을 위한 Speed Down 개시시간(Td)를 결정한다. 또한 원호보간의 경우에는 C부분을

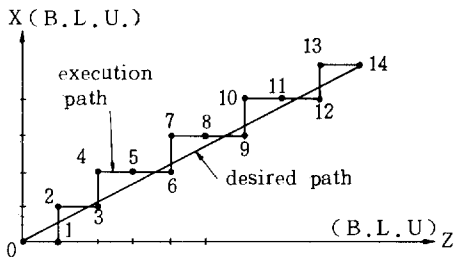

그림 9. 직선보간 결과

Fig. 9. Linear interpolation result

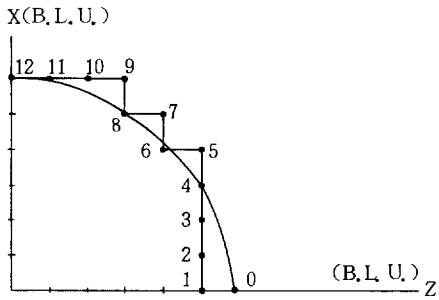

그림10. 원호보간 결과

Fig. 10. Circular interpolation result

표 1. 이송 속도제어 결과

Table 1. Result of feedrate control

Feed data	Speed [rpm]	Feed data	Speed [rpm]
10	41	60	246
20	82	70	287
30	123	80	327
40	164	90	368
50	205	100	408

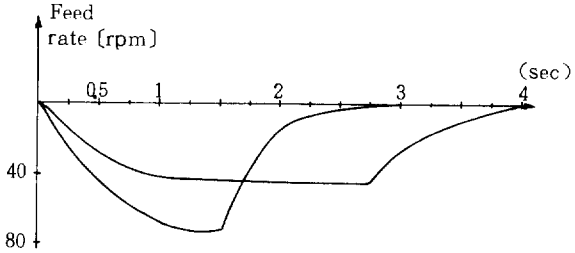

그림 11. 가감속수행 결과

Fig. 11. Result of speed up/down

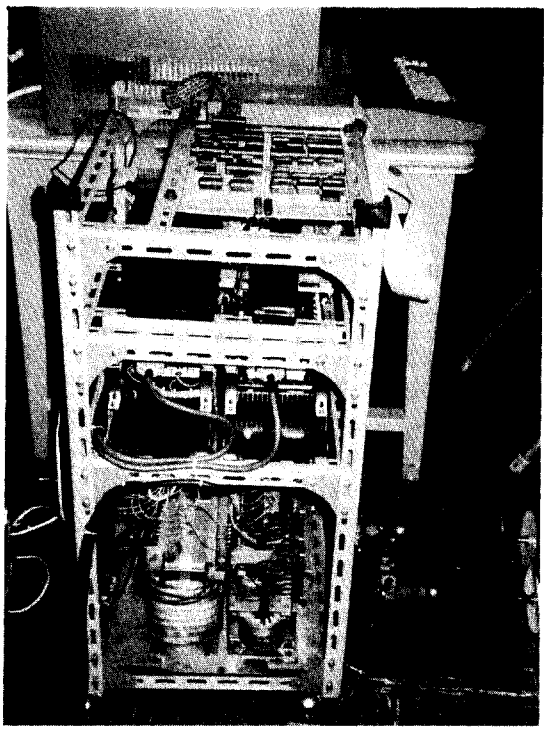

그림 12. Proto-type 2축연속경로제어 장치

Fig. 12. Proto-type 2-dimensional contouring control system

제외한 나머지 부분은 직선보간의 경우와 마찬가지이고 C부분을 원호보간에 대한 algorithm으로 대치함으로써 수행된다.

나. 실험결과

실험을 위해서 Intel Microcomputer ICE-85 Emulator, Position Error Control Unit, FANUC Velocity Control Unit, FANUC Model-5 DC Servo Motor를 사용해 만든 Proto-type 2축 연속경로제어장치를 만들었다. 이를 사용한 실험 결과는 그림 9～11과 표1에 나타나 있다. 그림 9와 10은 각각 직선과 원호보간에 대한 결과를 나타낸 것으로 기대한 대로 주어진 경로에 대한 오차는 한개의 B.L.U.(Basic Length Unit) 내에 있음을 보여주고 있다. 표1은 속도제어에 대한 결과를 측정한 것으로 각 속도 명령에 대한 선영특성이 좋은 것을 얻을 수 있었다. 그런데 여기서 나타난 속도 오차는 측정오차로 생각된다. 마지막으로 그림 11은 가감속 기능에 대한 결과로 속도가 영에서부터 설정된 속도에 도달하기까지 가속을 하고Td 순간부터는 감속을 하는 것을 알 수 있다.

5. 결과

2900 bipolar bit-slice microprocessors를 사용한 보간기를 실현하였다. 하드웨어와 소프트웨어의 효과적인 설계로 250 Kpulses/sec의 보간속도를 얻었는데 이는 8085 microprocessor에 의한 보간속도 (3 Kpulsec/sec)의 80배가 넘는 속도로서 최소지령단위가 1μ/pulse인 System에서 15 m/min의 이송속도에 해당한다. 또한 Feedrate Control 기능, 가감속 기능, 현재위치정보 발생 기능 등의 부수적인 기능들을 보간과 동시에 처리하게 함으로써 종래의 logic IC에 의한 실현에 비해 회로가 간단하면서도 보다 높은 신뢰성과 유연성을 갖게 하였다. 이러한 결과는 실험을 위하여 구성한 prototype 2축 연속 경로제어장치 (그림 13)를 통하여 확인되었다.

참 고 문 헌

1) 山岸正議 ; "NC 工作機械" pp 15～17 日刊工業新聞社

2) B. J. Lee and T. S. Nho ; "Linear and Circular Interpolation for 2-dimensional

Contouring Control, "Trans.of KSME Vol. 6, No. 4 (No. 1982)
3) YORAM KOREN : " Interpolator for Computer Numerical Control System, " IEEE Trans. on computer Vol. C- 25, No. 1 (Jan. 1976)
4) Mick & Brick ; "BIT-SLICE MICROPROCESSOR DESIGN" published by McGraw-Hill Book Company