14th International Symposium on Industrial Robots — October 2-4, 1984
Gothenburg, Sweden

A MICROPROCESSOR-BASED INTERPOLATOR

B.J. Lee*, T.S. Nho**, I.T. Yea***

Korea Advanced Institute of Science and Technology

P.O. Box 131 Dongdaemun Seoul, Korea

ABSTRACT

In this paper we present a microprocessor-based interpolator using algebraic arithmetic method. The-interpolator consists of 2901 "bit-slice" microprocessor chips and 0.5K ROMs of microprogram memory. The system design is an instruction-data-based architecture with 250ns cycle time. A significant feature of the interpolator is that it has flexibility, very fast interpolation speed of (max) 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 of the interpolator efficently. In addition to hardware and software design, experimental results are presented.

INTRODUCTION

Interpolator is being used extensively in N/C machine and robot for continuous path control. For continuous path control there are several methods of interpolation. One of these is algebraic arithmetic method which executes linear and circular interpolation, and was developed at Tokyo Univ.. (1) For practical continuous path control, in addition to interpolation, there requires several functions such as feedrate control, speed up/down and distribution-end acknowledge. Until this time these additional functions are realized separately from interpolation function, so that the whole hardware is relatively complex and unreliable because it is built by hard-wired logic circuits having complex timming consideration. And someone implements it using 8085-microprocessor. (2) But its execution speed is too low for practical application. However, as a result of development of high speed microprocessors, particularly "bit-slice" bipolar processors, the above problems can be solved nicely. And additional functions can be implemented simultaneously.

In this work we are concerned with practical hardware realization of powerful interpolator using the 2900-series bipolar microprocessors. The features of this interpolator are as follows.

- It is based on the algebraic arithmetic interpolation method
- Simultaneous processing of additional functions
- 250K pulses/sec speed (refers to 15m/min machine movement)
- In addition to linear and circular interpolation, other commands such as rapid traverse, handle pulse movement, and dwell are implemented.

Detailed of the interpolation system follow. Emphasis will be placed on the discussion of hardware architecture and software development. Following the introduction, we discuss the overall system and algebraic arithmetic method in section I. In Section II we consider in detail the design of hardware and in section III, we discuss software implementation of the system and also present the experimental results, Finally, in section IV we make conclusions.

I. OVERALL SYSTEM CONFIGURATION AND ALGEBRAIC INTERPOLATION METHOD

A. Overall system

In the computer control of N/C manufacturing system for the desired path movement, the command pulse generation part is the interpolator so that interpolator is important for continuous path control system. Since overall computerized numerical control concepts have appeared in the literature, (3) we describe here just the interpolation part. A significant feature of the present interpolation system is that it can handle 250K pulses/sec , simultaneously executes additional functions, thereby reduces the hardware complexity and

* Present KANGWON NATIONAL UNIVERSITY
** 〃 KIMM
*** 〃 KIMM

has flexibility. However, very efficient programming of software is essential to accomodate these functions in real time execution.

The overall interpolator system is shown in Fig. 1.

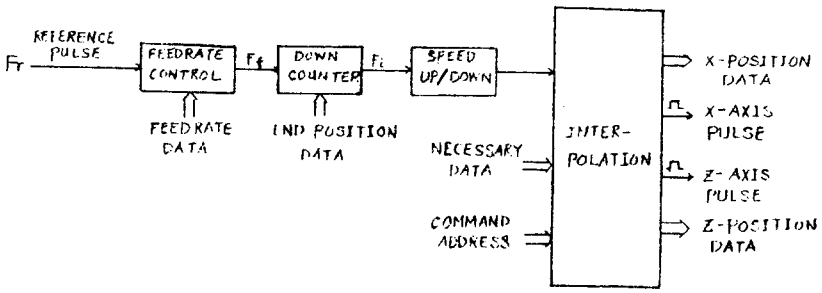
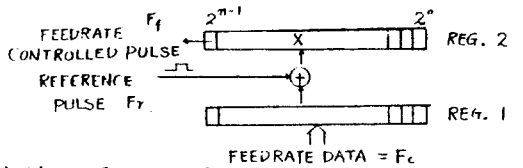


Fig. 1. Block Diagram of Overall Interpolator System

Reference pulse is fed to feedrate control part, this pulse is generated from spindle pulse coder in case of N/C lathe. Feedrate control part receives reference pulse and processes the feedrate control. Feedrate control algorithm is as follows.

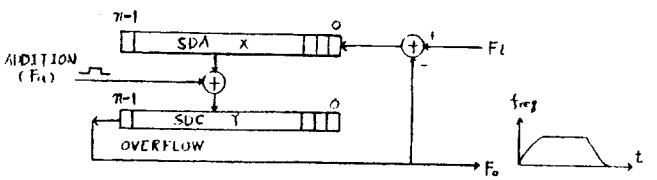


At each time reference pulse occurres, and Reg. 1 to Reg. 2 and if carry generated it is applied to Down Counter. Like this, according to the setting data in Reg. 1, feedrate can be controlled. This can be formulated as follows.

$X(k+1) = X(k) + Fc,\ X(0) = 0$ (1)

Solving Eq.(1), $X(k) = K \cdot Fc$, so that Ff is proportional to Fc.

Down counter receives feedrate controlled pulse, at each time it is detected, decrement Down Counter by one. In case it becomes zero no further pulse is not transmitted. This transmitted pulse is fed to speed up/down part. At this portion pulse rate is varied for the purpose of smooth motor movement. At start time the pulse rate is accelerate from zero, and after some setting time passed it reaches static value. Just after Down Counter becomes zero, speed down action begins. The following algorithm executes above speed up/down action.



At each time Fi pulse arrives, increment SDA reg. by one, and at each time Addition Command pulse is detected, add SDA reg. to SDC reg., and if overflow occurs at SDC, decrement SDA reg. by one.

Above can be formulated as follows.

$dx = (Fi - Fo) \cdot dt$ (1) $\quad x(0) = y(0) = 0$

$dy = Fa \cdot x \cdot dt$ (2)

$Fo \cdot dt = dy/2^{-n}$ (3)

From (1),(2),(3) we get $Fo = fi \cdot \{1 - \exp(-Fa \cdot t/2^{-n})\}$ -----(4).

Just after Down Counter becomes zero, Fi becomes zero, then eq(4) becomes $Fo = F(Td) \cdot \exp(-Fa \cdot t/2^{n})$ ----(5).

Finally Fo pulse is applied to interpolation part. At time Fo pulse detected interpolator process determines whether this pulse must be distributed to X-axis or Z-axis.

B. Algebraic Arithmetic Interpolation Method

1) Linear Interpolation

For linear interpolation required data are start position(Zs,Xs) and end position(Ze, Xe). For the path of Fig. 2, determine the present position is located at above the line or below the line. If present position is below the line the next step must go to +X direction, in order to go upward, and if otherwise next step must be +Z direction, in order to go downward.

This is formulated as follows.

Let $D_{k,i} = Ze\ Xi - Xe \cdot Zk$

a) In case $D_{k,i} \geq 0$

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

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

b) In case $D_{k\ i} < 0$

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

$D_{k,i+1} = D_{k,i} + Ze$

2) Circular Interpolation

For circular interpolation, required data are the same as linear case. For the path of Fig. 3, determine whether the present position is located at the inside of the circle or outside of the circle. If the present position is inside of the circle, next step must go to + X direction in order to go outward, and if otherwise next is -Z direction to go inward.

This can be formulated as follows.

Let $D_{k,i} = (Z_k \cdot Z_k + X_i \cdot X_i) - (Z_e \cdot Z_e + X_e \cdot X_e)$

a) In case $D_{k,i} \geq 0$

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

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

b) In case $D_{k,i} < 0$

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

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

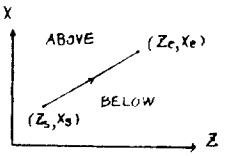


Fig. 2. Linear Interpolation

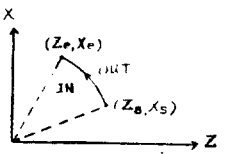


Fig. 3. CCW Circular Interpolation

II. Hardware design and Interpolation

A. Hardware Architecture

The simplified block diagram for the interpolator hardware system is shown in Fig. 4. It can be divided into four units. The four are microprogram control unit(MCU), central processing unit(CPU), microprogram memory, and input/output(I/O) unit. All micro-instructions are executed in a cycle time of 250ns. The nucleus of this system is the CPU which uses Advanced Micro Devices(AMD) 2901 bipolar microprocessor chips. Six such chips are used along with a lookahead-carry generator chip to form a 24-bit arithmetic logic unit (ALU). Resulting present X-axis or Z-axis position data is transmitted to output register. And resulting status such as Carry(BC), Zero(IZ), and Negative(IS) is latched at status register, since this system employs instruction-data-based architecture.

Refering again to Fig. 4, it is seen that the 24-bit CPU is connected to I1 and I2. Both I1,I2 are unidirectional tri-state data bus, and externally connected to the data bus of Main Computer. Through I1 necessary data for interpolation are supplied such as initial position, end position, feedrate and etc, and these are stored into one of the 16 registers 2901 chip has. M4 controls the operation of CPU.

Microprogram control unit consists of an AMD 2910 sequence controller, status register, condition multiplexer. The 2910 chip that is the core of MCU has an external condition input($\overline{CC}$), a 12-bit branch address input(D_{0-11}), three control outputs($\overline{PL}$, $\overline{VECT}$,and $\overline{MAP}$) to enable external address onto the branch input, and a 12-bit output(Y_{0-11})to address up to 4K of microprogram memory. CSC controls the operation of the sequencer, and INIT makes Y_{0-11} become zero.

In the present application, the branch input of the sequencer is derived from one of the two address sources; the 8-bit field in the microprogram register and the control register which contains the starting address of each command. These are enabled by $\overline{MAP}$,$\overline{PL}$ respectively. PMSI initialize processing of the command whose start address is fetched from control register. The condition MUX. provides jump condition to the 2910 from one of six condition inputs; four flags in the status register, PMSI, and Ipulse which refers to reference pulse.

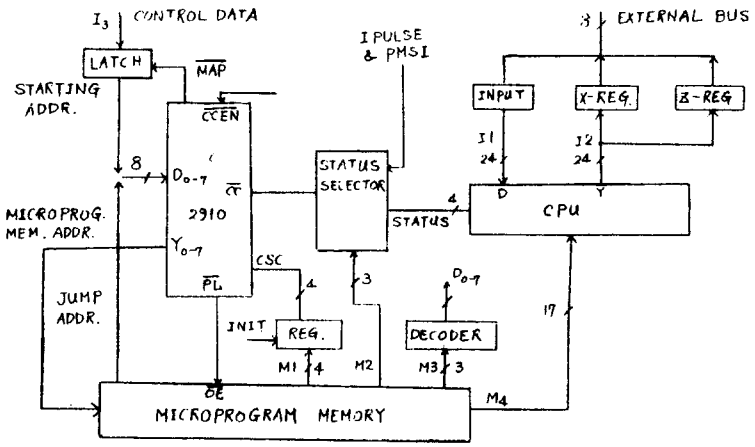


Fig. 4. Block Diagram of Interpolator Hardware

When $\overline{CCEN}$ is logic 0, then unconditional jump(PL=0) is executed. We use AMD 2727 and 2713 bipolar memory whose access time is 70ns for fast execution, and AMD 2727 has output reg. so that it is good for pipeline technique. The microprogram memory consists of 0.5K of 36-bit words. The output of this memory is clocked into this output microprogram memory register and then executed. In these scheme we can realize the interpolator with instruction-data-base architecture. (4)

The I/O unit consists of two parts. One is pulse generation part for output control, the other is I/O port. Fig. 5 shows the block diagram of the I/O unit. For the purpose of reducing the microprogram memory bit number we use DECODER. Now we explain the output control signals.

FBSTB : strobe signal to match the 8-bit data bus and 24-bit data bus
Sample: strobe signal to make the content of the X and Z-axis register data invariant during the sampling period
D0: enable signal to make 2901 output Y be on the X-reg.
D1: " Z-reg.
D2: X-axis pulse out
D3: Z-axis pulse out
D4: indication signal of execution end
D5: clear Ipulse
D6: clear PMSI

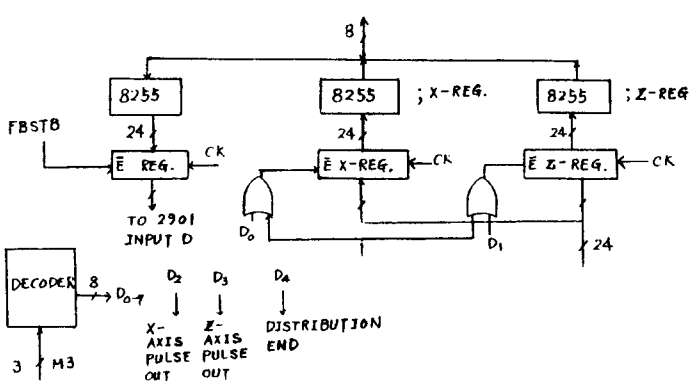


Fig. 5. Block Diagram of I/O Unit

B. Instruction Format

The micro-instruction is made up of 36 bits whose format is shown below, The word is

divided into CPU, MCU, and I/O control field.

CPU control (17-bit) : M4 ; 1. ALU source (3-bit)
2. ALU function (3-bit)
3. ALU destination (2-bit)
4. ALU address (8-bit)
5. Carry generator (1-bit)

MPU control (16-bit) :
1. Sequencer control (4-bit)
2. Branch address (8-bit)
3. Status selector (3-bit)
4. Unconditional jump control (1-bit)

I/O control (3-bit) : M3

III. SOFTWARE DESIGN AND IMPLEMENTATION

Interpolator firmware can be divided into three parts; data accept routine that accepts all required data for the execution of function, command execution routine which executes linear and circular interpolation, and etc, and initialization and return-routine. The system is in initialization routine just after POWER ON or when no function is executing. In initialization routine interpolator accepts the command. If command start signal (PMSI) detected, unconditional jump to the start address of command is executed, and then executes that command. Within 16 steps-cycle both linear and circular-interpolation can be executed, so that the Addition Frequency (Fa) is automatically determined to be 250K Hz.

Required data for interpolation are as follows.

Zk ; Z-axis start position — Xe : X-axis end position
Xi : X-axis start position — Dk : Discriminant Value
Ze : Z-axis end position — WUD: Whole traverse distance
Fc : Feedrate control data — Fd : Initial zero for feedrate
SDA,SDC : Initial zero for speed UP/DOWN

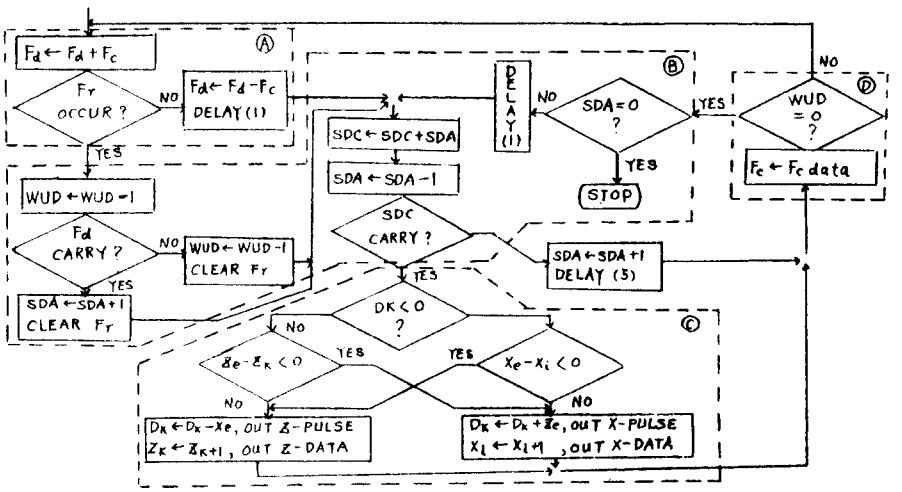


Fig. 6. Flow chart of Linear Interpolation

Fig. 6 shows the flow chart of linear interpolation. At portion A feedrate control is executed, portion B executes the speed UP/DOWN, portion C executes linear interpolation, and portion D determines speed DOWN time (Td).
And circular interpolation is the same as the linear interpolation except interpolation portion C.

The experiment is performed with Intel Microcomputer ICE-85 Emulator, position error control unit, FANUC velocity control unit, FANUC Model-5 DC servo motor. Results are shown as Fig. 7,8,9,10. Fig. 7 and 8 show the linear and circular interpolation result respectively. From these we know that error is within one B.L.U. (Basic Length Unit). Fig. 9 shows the speed of motor according to feedrate data. This shows good linear characteristic. Fig. 10 shows the performance of speed UP/DOWN action. At low feedrate we can see clearly speed up/DOWN action, but at high speed it begins speed-down action before it reaches static feedrate. These are the same with expected results.

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

Fig. 7. Result of Linear Interpolation

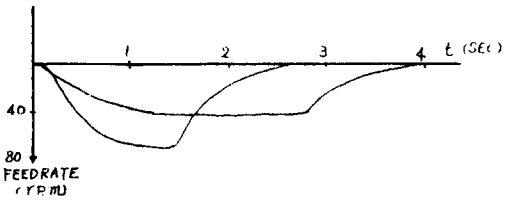


Fig. 8. Result of Circular Interpolation

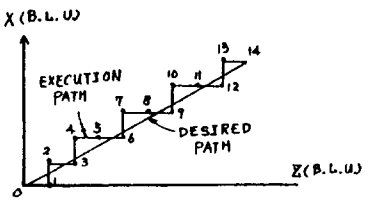


Fig. 9. Result of Feedrate Control

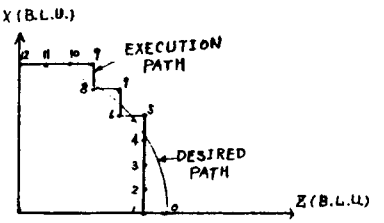


Fig. 10. Result of Speed UP/DOWN

IV. CONCLUSION

We have presented implementation of a microprocessor-based interpolator. With efficient design of hardware and software, it was possible to realize the system that is capable of processing the 250K pulses/sec speed interpolation and additional functions simultaneously. Thus it makes possible to enhance the reliability and to reduce the hardware complexity. And we have good experimental results.

REFERENCES

1. 山岸正曠, "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

※編輯者註 : 이 논문은 제 14 차 ISIR에서 채택된 것을 발췌한 것임.