'91 KACC 1991. 10. 22~24

SOFTWARE LINEAR AND EXPONENTIAL ACCELERATION/DECELERATION METHODS FOR INDUSTRIAL ROBOTS AND CNC MACHINE TOOLS

Dong-Il Kim, Jin-Il Song, Yong-Gyu Lim, and Sungkwun Kim

Control R/D Team, Production Engineering Division,
Consumer Electronics Business, Samsung Electronics.

Abstract

Software linear and exponential acceleration /deceleration algorithms for control of machine axes of motion in industrial robots and CNC machine tools are proposed. Typical hardware systems used to accelerate and decelerate axes of motion are mathematically analyzed. Discrete-time state equations are derived from the mathematical analyses for the development of software acceleration/deceleration algorithms.

Synchronous control method of multiple axes of motion in industrial robots and CNC machine tools is shown to be easily obtained on the basis of the proposed acceleration/deceleration algorithms.

The path error analyses are carried out for the case where the software linear and exponential acceleration/deceleration algorithms are applied to a circular interpolator.

A motion control system based on a floating point digital signal processor (DSP) TMS 320C30 is developed in order to implement the proposed algorithms. Experimental results demonstrate that the developed algorithms and the motion control system are available for control of multiple axes and nonlinear motion composed of a combination of lines and circles which industrial robots and CNC machine tools require.

1. Introduction

The new era of automation, which started with the introduction of industrial robots and CNC machine tools, was undoubtedly stimulated by the digital computer. Digital technology and high performance microprocessors enabled the control systems for industrial robots and CNC machine tools to be designed more flexibly. The advantages of flexible control systems are adaptability to change of products in a short time. In addition, such control systems can resolve stringent restrictions in performances required in industry.

The significant common requirement of industrial robots and CNC machine tools is to generate coordinated movement of the separately driven axes of motion to achieve the desired path tracking of the tool or arm relative to the workpiece. This involves both the generation of reference signals prescribing the shape of the produced part and the control of the machine to follow these signals. Generally, industrial robots and CNC machine tools require distinct interpolation routines and acceleration/deceleration routines to generate their corresponding signals.

The desired path in an industrial robot or a CNC machine tool is usually built up from a combination of linear and circular segments, and accordingly, both linear and circular interpolation routines are basically contained in the reference signal generation [1], [2]. However, a well-known 2 dimensional circular interpolator (so called reference-word interpolators) is only discussed briefly in this paper.

For control of movable elements with little rigidity without vibration at start and stop points in industrial robots and CNC machine tools, acceleration/deceleration techniques are required. In this paper, typical hardware systems used to accelerate and decelerate axes of motion linearly or exponentially are mathematically analyzed. Through the analyses, discrete-time state equations describing the characteristics of the hardware systems can obtained, which enable the equivalent software algorithms to be obtained.

In actual operations of industrial robots and CNC machine tools, the selection of acceleration/deceleration methods deeply affects the accuracy in machining. Therefore, the optimal acceleration/deceleration method which guarantees minimum error in machining should be selected. Accordingly, the mathematical analyses of path error for linear and exponential acceleration/deceleration is performed to compare the effect of each method on a circular interpolator.

In order for axes of motion to track the corresponding reference signals for a desired path, the optimal control technique and setting of control parameters are firstly required to be selected. Recently, it has become possible to apply modern control theories due to the great advances in microprocessor technology.

From a view point of the robot and CNC controller design, there is a greater need to simplify installation, programming, and to enhance the execution speed of the control algorithms. Especially, the performance of tracking of the desired path and the maximum attainable tracking speed are proportional to the computing speed of interpolators, acceleration/deceleration, and control algorithms. These practical requirements impose a heavier load on the microprocessor CPU, making the use of a DSP mandatory. In this paper, a motion control system based on a 32 bit floating point DSP has been developed. Software acceleration/deceleration, interpolations, and control algorithms are executed in this system.

Experimental results show that the proposed algorithms and the motion control system offer efficient solutions to the problems such as flexible control of coordinated multiple axes of motion and nonlinear motion composed of a combination of lines and circles which industrial robots and CNC machine tools require.

2. Software Acceleration/Deceleration

First, let f_i shown in Fig.1 be input to the hardware systems for acceleration/deceleration. One can see that the input f_i is a pulse train whose frequency is f_c. The physical meaning of each pulse in the pulse train is a unit position increment in the movement along a desired path.

2.1. Linear Acceleration/Deceleration

The schematic diagram of the hardware system for linear acceleration/deceleration is shown in Fig.2 [3], where buffer registers act as delay elements. Fig.2 shows that f_i, f_i delayed by m-steps, and the output f_o delayed by one step are summed and the result is divided by m in order to obtain the linear acceleration/deceleration characteristics. The final result is expressed as the following discrete-time state equation:

$$f_o(k) = [f_i(k)-f_i(k-m)]/m + f_o(k-1), \quad (1)$$

where m is the number of buffer registers. If the sampling time is given by T_s, then the acceleration/deceleration time $t_{acc/dec}$ is adjusted by the following relationship:

$$t_{acc/dec} = mT_s. \quad (2)$$

The software linear acceleration/deceleration algorithm for control of machine axes of motion in industrial robots and CNC machine tools can be obtained from (1) .

In order to analyze the linear acceleration /deceleration characteristic, it is required to derive the transfer function H(z) ($F_o(z)/F_i(z)$) in the z-domain which describes the characteristic of the discrete-time state equation (1). The transfer function H(z) is given by

$$H(z) = z(1-z^{-m})/m(z-1). \quad (3)$$

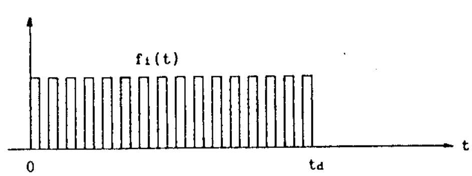


Fig.1 Input to the hardware systems for acceleration/deceleration.

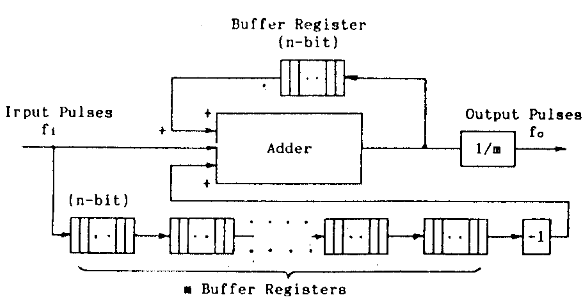


Fig.2 The schematic diagram of the hardware system for linear acceleration/deceleration.

2.2. Exponential Acceleration/Deceleration

The schematic diagram of the hardware system for exponential acceleration/deceleration is shown in Fig.3 [2], [4].

The mathematical expressions which describe the hardware system in Fig.3 are given by the following equations [5], [6]:

$$dx/dt = (f_i-f_o), \quad (4)$$

$$dy/dt = xf_a, \quad (5)$$

$$f_o = (dy/dt)/2^n. \quad (6)$$

Then, the transfer function H(s) ($F_o(s)/F_i(s)$) is obtained as

$$H(s) = \tau/(s+\tau), \quad (7)$$

where $\tau=f_a/2^n$. The transfer function $F_o(s)/F_i(s)$ represents a differential equation.

Now, we approximate the derivatives in (3)-(6) with the backward difference approximation in order to obtain the discrete-time state equations describing the characteristic of the hardware system in Fig.3. Using the backward difference approximation, the transfer function H(z) ($F_o(z)$ $/F_i(z)$) is obtained as follows:

$$H(z) = z(1-a)/(z-a), \quad (8)$$

where $a=1/(1+\tau T)$. Then, the approximated discrete-time state equations are obtained as

$$f_o(k) = f_o(k-1)+a[f_i(k)-f_o(k-1)]. \quad (9)$$

The software exponential acceleration/deceleration algorithm for control of machine axes of motion in industrial robots and CNC machine tools can be obtained from (9). If fine tuning of the acceleration/deceleration time is needed, 2^n may be replaced any arbitrary integers which give the desired performance. Such replacement is not possible in a hardware system but enabled by software through a microprocessor.

Hardware systems composed of the combination of digital or analog circuits, which provide other acceleration/deceleration methods, can be analyzed mathematically in the discrete-time do-

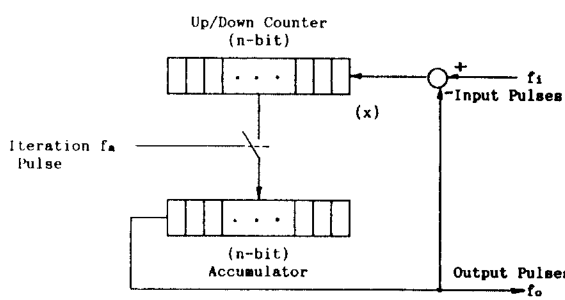


Fig.3 The schematic diagram of the hardware system for exponential acceleration/deceleration.

main by the similar way. The mathematical analyses will give discrete-time state equations, from which software acceleration/deceleration algorithms can be easily obtained.

2.3. Synchronous Control of Multiple axes of motion

Now, the method to synchronize the movement of four axes of motion in industrial robots and CNC machine tools is described [6], [7]. Let us define position deviations of x-, y-, z-, and w-axis as Px, Py, Pz, and Pw, respectively. The maximum value P_{max} among the position deviations is defined as MAX {Px, Py, Pz, Pw }. At every sampling instant k, the increments of input pulses to reduce the position deviations should be calculated from maximum speeds of servo motors and resolution of position sensors mounted on motor shafts such as optical encoders or resolvers. The number of iteration steps N, which is needed to eliminate the maximum possible position deviation, is obtained from

$$N = P_{max}/f_i(k), \tag{10}$$

where $f_i(k)$ has the same value at every sampling instant k. Hence, if we assume that $P_{max}= P_x$, the increment of input pulse for each axis is given by

$$f_{ix}(k) = f_i(k), \quad f_{iy}(k) = P_y/N,$$
$$f_{iz}(k) = P_z/N, \quad f_{iw}(k) = P_w/N. \tag{11}$$

The velocity profile for each axis of motion can be calculated from (1), (9), which is described as follows:

(1) Linear acceleration/deceleration

$$f_{ox}(k) = [f_{ix}(k)-f_{ix}(k-m)]/m + f_{ox}(k-1),$$
$$f_{oy}(k) = [f_{iy}(k)-f_{iy}(k-m)]/m + f_{oy}(k-1),$$
$$f_{oz}(k) = [f_{iz}(k)-f_{iz}(k-m)]/m + f_{oz}(k-1), \tag{12}$$
$$f_{ow}(k) = [f_{iw}(k)-f_{iw}(k-m)]/m + f_{ow}(k-1).$$

(2) Exponential acceleration/deceleration

$$f_{ox}(k) = f_{ox}(k-1) + a[f_{ix}(k)-f_{ox}(k-1)],$$
$$f_{oy}(k) = f_{oy}(k-1) + a[f_{iy}(k)-f_{oy}(k-1)],$$
$$f_{oz}(k) = f_{oz}(k-1) + a[f_{iz}(k)-f_{oz}(k-1)], \tag{13}$$
$$f_{ow}(k) = f_{ow}(k-1) + a[f_{iw}(k)-f_{ow}(k-1)].$$

where $f_{ox}(-1)=f_{oy}(-1)=f_{oz}(-1)=f_{ow}(-1)=0$. Σf_o for each axis from the initial point to an arbitrary sampling instant k yields the position path for elimination of the position deviation of each axis:

$$P_{cx} = \sum_{i=0}^{k} f_{ox}(i), \quad P_{cy} = \sum_{i=0}^{k} f_{oy}(i),$$
$$P_{cz} = \sum_{i=0}^{k} f_{oz}(i), \quad P_{cw} = \sum_{i=0}^{k} f_{ow}(i), \tag{14}$$

where P_{cx}, P_{cy}, P_{cz}, and P_{cw} are position path commands of x-, y-, z-, and w-axis, respectively.

3. Circular Interpolator

Linear and circular interpolators incorporated into the foregoing acceleration/deceleration algorithms efficiently provide the nonlinear motion composed of lines and circles which industrial robots and CNC machine tools require. In this section, a well-known 2 dimensional cir-cular interpolator, which is called reference-word interpolator, is only discussed briefly in order to analyze the path error attributed to exponential acceleration/deceleration. In case of the linear interpolation, there exists no path error.

In the circular interpolation, the simultaneous motion of axes generates a circular arc at a constant tangential velocity, or feedrate. The velocity components Vx and Vy are generated by the circular interpolator and supplied as inputs to the acceleration/deceleration algorithm. The circular arc generated in this case actually comprised straight line segments.

Each iteration of the interpolation algorithm corresponds to an angle α, as is illustrated in Fig.4. Then, the following difference equations are obtained [1], [2]:

$$X(k+1) = AX(k)-BY(k),$$
$$Y(k+1) = AY(k)+BX(k), \tag{15}$$
$$\theta(k+1) = \theta(k)+\alpha,$$

where

$$A = \cos\alpha \text{ and } B = \sin\alpha. \tag{16}$$

At each iteration point X(k), Y(k), the implemented circular interpolator calculates the coordinates of the successive point X(k+1), Y(k+1) according to (15). The segment lengths are expressed as

$$DX(k) = (A-1)X(k)-BY(k),$$
$$DY(k) = (A-1)Y(k)+BX(k), \tag{17}$$

and the corresponding velocities are given by

$$V_x(k) = VDX(k)/DS(k),$$
$$V_y(k) = VDY(k)/DS(k), \tag{18}$$

where

$$DS(k) = \sqrt{DX^2 + DY^2}, \tag{19}$$

and V is a constant tangential velocity, or feedrate on a circular arc. Since the angle α is rel-

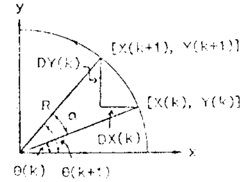


Fig.4 Two successive points on a circular arc.

atively small, the chord length DS can be approximated by its arc length $R\alpha$, and the calculation of the velocities $V_x(k)$, $V_y(k)$ can be simplified to

$$
\begin{aligned}
V_x(k) &= -[2V\sin(\alpha/2)/\alpha]\sin(\theta(k)+\alpha/2),\\
V_y(k) &= \ \ [2V\sin(\alpha/2)/\alpha]\sin(\theta(k)+\alpha/2).
\end{aligned}
\tag{20}
$$

If the interpolation is performed every T second, the following relationship is obtained:

$$
R\alpha/V = T. \tag{21}
$$

Let Ω be V/R, then

$$
\theta(k) = \Omega Tk. \tag{22}
$$

Substituting (21) and (22) into (20) yields

$$
\begin{aligned}
V_x(k) &= -[2R\sin(\alpha/2)/T]\sin(\Omega Tk+\Omega T/2),\\
V_y(k) &= \ \ [2R\sin(\alpha/2)/T]\cos(\Omega Tk+\Omega T/2).
\end{aligned}
\tag{23}
$$

The data $V_x(k)$, $V_y(k)$ computed based on the above circular interpolator are supplied as reference inputs to the acceleration/deceleration algorithm to generate a circular arc.

4. Path Error Analysis for the Circular Interpolation

We define the path error as the difference in the radius between a required circular arc and the one generated from the output of the acceleration/deceleration algorithm which has data $V_x(k)$, $V_y(k)$ as input. In this section, we carry out the path error analyses in the steady state for the case where the linear and exponential acceleration/deceleration algorithms are applied to the circular interpolator described in preceding section.

As a precondition, the sampling time T is assumed to be so small as to satisfy the validity of equations to be developed. Practically, this assumption is satisfied due to the high performance characteristic of a DSP TMS 32030

In case of the linear interpolation, there exists no path error attributed to linear and exponential acceleration/deceleration. This can be easily proved from the fact that the ratio between Zx and Zy , the increment values in the x- and y- axis which are input commands to acceleration/deceleration, is the same as that between output responses through acceleration /deceleration.

Now, the path errors for linear and exponential acceleration/deceleration are analyzed. The commands to acceleration/deceleration are the velocity data $V_x(k)$, $V_y(k)$ in (23).

(1) Linear acceleration/deceleration

The output responses $Y_x(k)$ and $Y_y(k)$ to $V_x(k)$ and $V_y(k)$ in (23) are given by

$$
\begin{aligned}
Y_x(k) &= -[2R\sin(\alpha/2)/mT][\cos(\Omega T/2)Y_s(k)+\sin(\Omega T/2)Y_c(k)],\\
Y_y(k) &= \ \ [2R\sin(\alpha/2)/mT][\cos(\Omega T/2)Y_c(k)+\sin(\Omega T/2)Y_s(k)],
\end{aligned}
\tag{24}
$$

where

$$
\begin{aligned}
Y_c(k) &= [\{\cos(\Omega Tk+\Phi_w)/(\sqrt{2-2c_0})+(1-c_0)/(2-2c_0)\}u(k)-\{\cos(\Omega Tk-\Omega Tm+\Phi_w)/(\sqrt{2-2c_0})+(1-c_0)/(2-2c_0)\}u(k-m)]/m,\\
Y_s(k) &= [\{\sin(\Omega Tk+\Phi_w)/(\sqrt{2-2c_0})+s_1/(2-2c_0)\}u(k)-\{\sin(\Omega Tk-\Omega Tm+\Phi_w)/(\sqrt{2-2c_0})+s_1/(2-2c_0)\}u(k-m)]/m,
\end{aligned}
\tag{25}
$$

and

$$
\begin{aligned}
c_0 &= \cos\Omega T, \quad s_1 = \sin\Omega T,\\
\Phi_w &= \mathrm{atan}\{-s_1 a/(1-c_0 a)\}.
\end{aligned}
\tag{26}
$$

If k tends to the infinity, (24) yields

$$
\begin{aligned}
Y_x(\infty) &= -[2R\sin(\alpha/2)/T][2\sin(\Omega Tm/2)\sin(\Omega Tk+\Omega T/2+\Phi_w+\pi/2-\Omega Tm/2)/(m\sqrt{2-2c_0})],\\
Y_y(\infty) &= \ \ [2R\sin(\alpha/2)/T][2\sin(\Omega Tm/2)\cos(\Omega Tk+\Omega T/2+\Phi_w+\pi/2-\Omega Tm/2)/(m\sqrt{2-2c_0})].
\end{aligned}
\tag{27}
$$

Now, let us define R' as the radius of the generated circular arc from the data $V_x(k)$, $V_y(k)$. Then, the following result which describes the path error for linear acceleration /deceleration is obtained from (27).

$$
\Delta R = R - R' = R[1-2/m\sqrt{2-2c_0}]. \tag{28}
$$

(2) Exponential acceleration/deceleration [6]

The output responses $Y_x(k)$ and $Y_y(k)$ to $V_x(k)$ and $V_y(k)$ in (23) are given by

$$
\begin{aligned}
Y_x(k) &= -[2R\sin(\alpha/2)/T][G_w\sin(\Omega Tk+\Omega T/2+\Phi_w)+\{M_{sw}\cos(\Omega T/2)+M_{cw}\sin(\Omega T/2)\}a^k],\\
Y_y(k) &= \ \ [2R\sin(\alpha/2)/T][G_w\cos(\Omega Tk+\Omega T/2+\Phi_w)+\{M_{cw}\cos(\Omega T/2)+M_{sw}\sin(\Omega T/2)\}a^k],
\end{aligned}
\tag{29}
$$

where

$$
\begin{aligned}
G_w &= (1-a)/\sqrt{a^2-2c_0 a+1},\\
M_{sw} &= (1-a)s_1 a/(a^2-2c_0 a+1),\\
M_{cw} &= (1-a)(a^2-c_0 a)/(a^2-2c_0 a+1).
\end{aligned}
\tag{30}
$$

If k tends to the infinity, (29) yields

$$
\begin{aligned}
Y_x(\infty) &= -[2R\sin(\alpha/2)/T][G_w\sin(\Omega Tk+\Omega T/2+\Phi_w),\\
Y_y(\infty) &= \ \ [2R\sin(\alpha/2)/T][G_w\cos(\Omega Tk+\Omega T/2+\Phi_w).
\end{aligned}
\tag{31}
$$

Finally, the following result which describes the path error for exponential acceleration/deceleration is derived from (31).

$$
\Delta R = R - R' = R[1-1/\sqrt{1+2a(1-c_0)/(a-1)^2}]. \tag{32}
$$

If some conditions are assumed, (28) and (32) can be approximated to much more simple forms which represent the practical physical meanings. However, they are not included in this paper due to the limited space.

In case of parabolic, S-curve, or other acceleration/deceleration, the analysis of the path error can be carried out by the same way.

5. Motion Control System Based on a DSP

In order to implement the proposed algorithms, a motion control system with a floating point DSP TMS320C30 as a CPU [8], which can control six

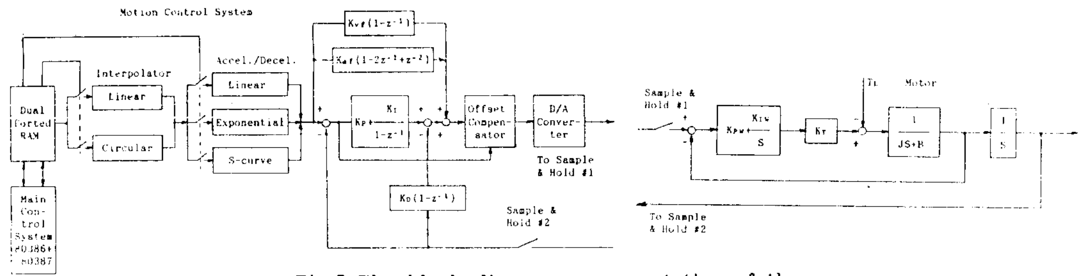


Fig.5 The block diagram representation of the motion control system for one machine axis of motion.

machine axes of motion synchronously, has been designed [9]. The main characteristics of the motion control system whose block diagram representation for one machine axis of motion is shown in Fig.5 are summarized as follows:

1) Signals between the motion controller and the main CPU(32bit) which handles main operating algorithms including CNC functions, graphics, communication and user interface are processed through dual ported RAMs.
2) The motion control system offers linear/circular interpolations, interchangeable acceleration/deceleration (linear, exponential, S-curve, etc.), and very high speed encoder capability with precise synchronization between motion and external events.
3) In position control loop, velocity and acceleration feedforward control terms are incorporated into the PID (Proportional-Integral-Derivative) controller with individually adjustable gains for reduction or elimination of following errors [10].
4) A tunable notch filter which is used to damp a critical machine vibration frequency is included, if needed.
5) The motion control systems may be daisy-chained together for extension of controlled axes in general automation, robots, and CNC machine tools.
6) The motion control system outputs either a DC signal for driving servo motor drives or two sinusoidal phase commutated signals for commands in current controllers.

6. Experimental Results

Performances of the proposed algorithms and the motion control system were investigated by experiments. In experiments, a cartesian type robot of Samsung Electronics (Fara Robot C2) whose actual photograph is shown in Fig.6 was used instead of a real machining center. Fara Robot C2 is driven by permanent magnet AC servo motors with position sensors whose resolutions are 4096 puses/rev.

First, the experimental results of linear and exponential acceleration/deceleration algorithms are presented. Considered was the case where 81920 pulses were needed in order to reach the goal position in the x-axis. Fig.7(a) shows the position path command for linear acceleration/deceleration and Fig.7(b) shows the position path command for exponential acceleration/deceleration.

Second, we show the experimental results for the circular interpolation under linear acceleration/deceleration. In case of exponential acceleration/deceleration, the results showed almost the same characteristics, so are not included in this paper. Fig.8(a) shows the locus for the linear interpolation under linear acceleration/deceleration while Fig.8(b) that for the circular interpolation. Actually, the offset in the input port of a servo motor controller deteriorates the performance of interpolations. In experiments, an adaptive offset compensation algorithm was incorporated into the motion control algorithm in order to solve such demerits.

In experiments, only 12 bit position data could be obtained due to resolution of the encoder mounted on a motor shaft. If the encoder with higher resolution is used, the DSP TMS320C 30 with fast execution time will enable more accurate and enhanced motion control performances to be obtained.

7. Conclusion

A motion control system based on a 32 bit floating point DSP has been developed. This system is very effective in implementing software acceleration/deceleration, interpolation, and control algorithms. In addition, it provides advantages such as the simplification of hardware structure and flexibility in programming.

It has been shown that typical hardware systems which provide linear and exponential acceleration/deceleration characteristics can be analyzed to obtain equivalent software algorithms. This fact indicates that if the typical hardware systems for acceleration/deceleration can be analyzed mathematically, it is possible to realize equivalent characteristics through software by microprocessors and to derive mathematical analyses of performance.

Further researches should be directed to the mathematical analysis of the entire closed loop system including path error, speed, and position control loops so as to attain the efficient design method of the motion control algorithm in industrial robots and CNC machine tools.

Experimental results show that the proposed motion control system offers efficient solutions to the classic problems such as flexible control of coordinated multiple axes of motion and nonlinear motion composed of a combination of lines and circles which CNC manufacturing systems require.

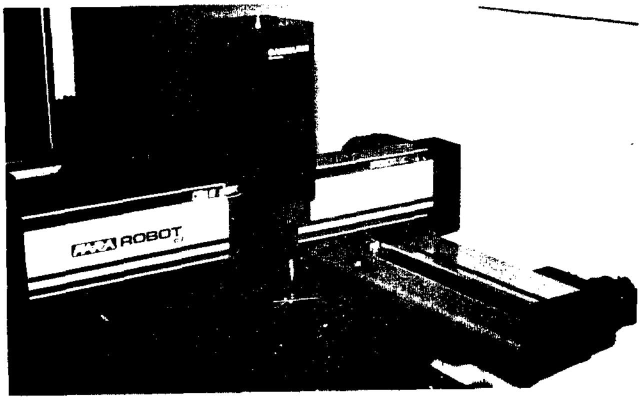
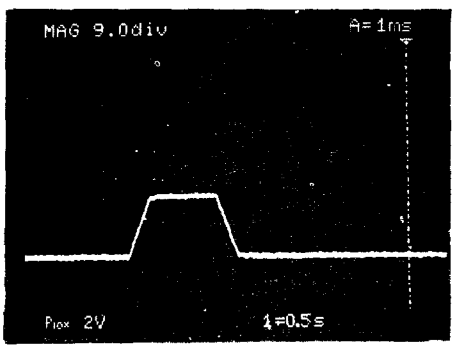
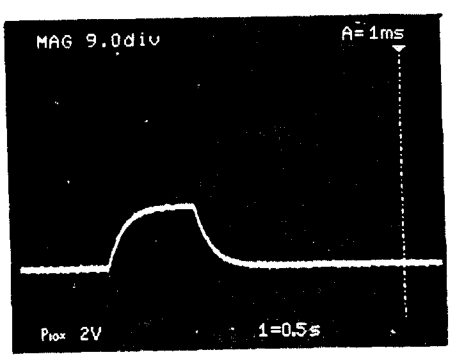


Fig.6 Fara Robot C2 of Samsung Electronics.

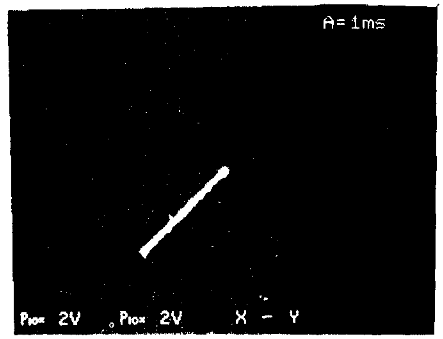


(a) Linear acceleration/deceleration.
(x-axis: 0.5sec)

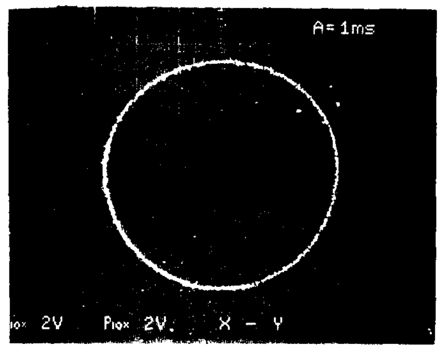


(b) Exponential acceleration/deceleration.
(x-axis: 0.5sec)

Fig.7 The position path command for acceleration /deceleration.



(a) Linear interpolation.



(b) Circular interpolation

Fig.8 The loci for the linear and circular interpolations under linear acceleration /deceleration.

References

[1] O. Masory and Y. Koren,"Reference-Word Circular Interpolators for CNC Systems," Trans. of ASME, vol.104, pp.400-405, 1982.

[2] Y. Koren, Computer Control of Manufacturing Systems, McGraw-Hill Inc, 1988.

[3] R. Nozawa et al., Acceleration/Deceleration Circuit, U.S. Patent 4,554,497.

[4] J. R. Armstrong, "Design of a Graphic Generator for Remote Terminal Application," IEEE Trans. Comput., vol.22, no.5, pp.464-469, 1973.

[5] J. I. Song, " Position Control of Servo Motors with Exponential Acceleration /Deceleration," pp.271-284, 2nd Proceeding of Technical Graduate School of Samsung Electronics, 1990.

[6] D. I. Kim, J. I. Song, and S. Kim, "Digital Signal Processor System for CNC Systems," will appear in IEEE IECON'91

[7] J. I. Song, "A Position Control Means and Method for Servo Motors," filed on Dec.29, 1989, Appl. No.07/456,788 (U.S. Patent)

[8] Texas Instruments, Third-Generation TMS 320 User's Guide, 1988.

[9] Samsung Electronics, Fara-AR1 Operator's Guide, 1991.

[10] D. I. Kim, J. W. Lee, and S. Kim, "Control of Permanent Magnet AC Servo Motors without Absolute Rotor Position Transducers," PESC' 91, pp.578-585, 1991.