

## CONTROL DESIGN PROCEDURE BASED ON MODIFIED DELTA FORM FOR IMPLEMENTATION

Tatsu Aoki

*Department of Electrical Engineering  
Tokyo Metropolitan College of Technology  
1 - 10 - 40 Higashi-Ohi, Shinagawa-ku, Tokyo 140 - 0011, Japan*

**Abstract:** This paper considers the computer based design procedure that reflects the latest theoretical development of the delta form. A fixed-point implementation of a delta form achieves poor performance due to underflow and overflow in the delta inverse operations. To solve this problem, a new modified delta form is proposed based on the methodology of Pulse Width Modulation (PWM). The dynamic range of the calculation becomes twice or more. Then, the implementation procedure by using MATLAB/SIMULINK is described generally. Finally, the simulated results are presented to show the practicality and effectiveness of the proposed algorithm. Copyright©2000 IFAC

**Keywords:** Microprocessor control, Implementation, Operators, Pulse width modulation, Control system design, Nonlinear control systems, PID controllers

### 1. INTRODUCTION

In order to realize quicker and precise motion, high-speed sampling is required in modern mechatronics applications. It has been shown that a numerical sensitivity of a control algorithm at high-speed sampling can be greatly reduced by a delta form(Agarwal and Burrus, 1971), (Goodall, 1985), (Middleton and Goodwin, 1986), (Goodall, 1989), (Middleton and Goodwin, 1990), (Goodall, 1990), (Goodwin *et al.*, 1992), etc. On the other hand, a fixed-point implementation of a control algorithm is usually adopted for an industrial application, since it is faster, cheaper and lower power consumption than floating-point implementation. Thus, a control system design based on fixed-point arithmetic is necessary in industrial sectors. However, a delta form achieves poor performance due to underflow and overflow in the delta inverse operations with fixed-point arithmetic.

To solve this problem, a new modified delta form was proposed (Aoki and Furukawa, 1996)(Aoki *et al.*, 1997)(Aoki, 1999). First, a new integral calculation method named Variable Modulation Method (V.M.M.) is proposed to reduce occurrence of underflow. The

principle is based on the methodology of Pulse Width Modulation (PWM). Because the quantization step becomes 0.5, 0.25, etc. in the integral calculation step, the dynamic range of the calculation becomes twice or more. The input-output relation of the delta inverse operation is shown in Fig. 1 after a hundred times iterations. The calculation is executed at 8bits fixed-point arithmetic and the sampling period  $T$  is set to be the quantization step. Second, time scales  $T_1 - T_n$ , which is defined as design parameters, are introduced to prevent overflow in  $n$ th-order controllers. Because an usual delta operator is divided by a sampling period  $T$ , it is difficult to express the algorithm coefficients in narrow dynamic range of fixed-point arithmetic. Simulations showed that the numerical advantages of V.M.M. were verified for digital filters.

Thus, in this paper the new control design procedure based on modified delta form using MATLAB SIMULINK is described generally. Fixed-Point Blockset is available for fixed-point simulation within SIMULINK. In order to illustrate the effectiveness of the proposed methodology, a PID controller is considered. This procedure can be extended to  $n$ th-order controller easily.

## 2. CONVENTIONAL DESIGN PROCEDURE

### 2.1 Derivation of control algorithm

The PID controller transfer function is given by (1).

$$H(s) = k_p \left( 1 + \frac{T_d s}{1 + a T_d s} + \frac{1}{T_i s} \right) \quad (1)$$

where  $k_p$  is proportional gain,  $T_d$  is derivative gain,  $T_i$  is integral gain, and  $a$  is 0.1. The transfer function is discretized by using zero-order hold.

$$H(z) = k \frac{z^2 + b_1 z + b_2}{z^2 + a_1 z + a_2} \quad (2)$$

where

$$\begin{aligned} k &= k_p \left( 1 + \frac{1}{a} \right) \\ a_1 &= -(1 + e^{-\frac{T}{a T_d}}) \\ a_2 &= e^{-\frac{T}{a T_d}} \\ b_1 &= -\frac{a}{a+1} \left( \frac{2}{a} + \left( 1 - \frac{T}{T_i} \right) + e^{-\frac{T}{a T_d}} \right) \\ b_2 &= \frac{a}{a+1} \left( \frac{1}{a} + \left( 1 - \frac{T}{T_i} \right) e^{-\frac{T}{a T_d}} \right) \end{aligned}$$

A block diagram of a controller is the canonical direct form 2 because the canonical direct form 2 is usually adopted in a delta operation. From (2) a control algorithm based on shift operator is derived.  $m()$  represents rounding.

$$\begin{aligned} x_k &= e_k - m(a_1 x_{k-1}) - m(a_2 x_{k-2}) \\ \bar{u}_k &= x_k + m(b_1 x_{k-1}) + m(b_2 x_{k-2}) \\ u_k &= m(k \bar{u}_k) \\ x_{k-2} &= x_{k-1} \\ x_{k-1} &= x_k \end{aligned} \quad (3)$$

### 2.2 Design procedure

#### (Step 1) Modeling a plant using floating-blocks

As an illustration, the design of a computer hard-disk read/write head position controller (Grace *et al.*, 1990). The transfer function of the head is shown as follows.

$$H(s) = \frac{K_i}{Is^2 + Cs + K} \quad (4)$$

where  $I$  is the inertia of the head assembly,  $C$  is the viscous damping coefficient of the bearings,  $K$  is the return spring constant, and  $K_i$  is the motor torque constant. The values  $I = 0.01 \text{ kgm}^2$ ,  $C = 0.004 \text{ Nm/(rad/s)}$ ,  $K = 10 \text{ Nm/rad}$ , and  $K_i = 0.05 \text{ Nm/A}$  are used. Because a plant seems to be equivalent to connect a plant with the filter  $(1+z^{-1})/2$  in series at the proposed algorithm, a augmented plant is defined as shown in Fig. 2.

#### (Step 2) Adjusting the three gain factors

The three gain factors are determined by classical frequency response analysis by using the function *frdesign* (Saadat, 1993). The values  $k_p = 5000$ ,  $T_d = 0.01$ , and  $T_i = 0.05$  are obtained.

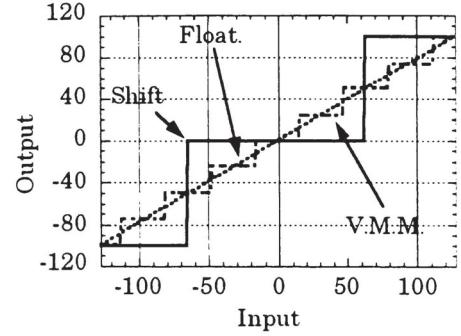


Fig. 1. Delta inverse operation

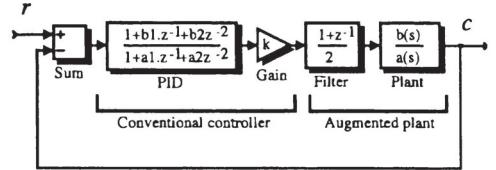


Fig. 2. PID controlling system (Shift form)

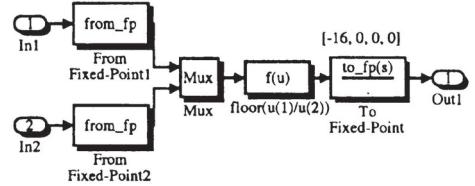


Fig. 3. Block diagram for exact truncation

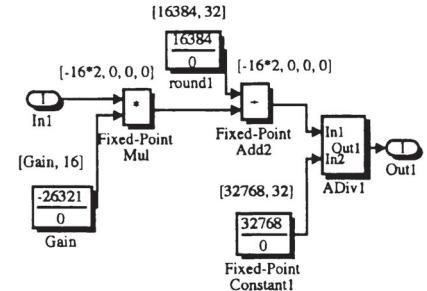


Fig. 4. Gain block with exact rounding

In addition, sampling period is set to be 0.5 ms from step response.

#### (Step 3) Scaling

From step response using usual floating-point blockset the maximum value of the internal value  $x_k$  can be obtained. Word length of input, output, internal variables, coefficients of the algorithm are supposed to be all signed 16bits. Also, binary point is located at LSB. The scaling factor for the controller input is obtained as 2048 (= 32768/ $x_{max}$ ).

#### (Step 4) Simulation of a closed-loop system

The step response and the sinusoidal response are shown in Fig. 5 and Fig. 6. As the magnitude of input is decreased, the response is far away from the reference.

### 3. PROPOSED DESIGN PROCEDURE

#### 3.1 Derivation of control algorithm

A controller based on V.M.M. is shown in Fig. 7 and Fig. 8(Aoki and Furukawa, 1996)(Aoki *et al.*, 1997). It is assumed that a transfer function of a controller based on the shift operator is obtained by any method.

(Step 1) Transformation a shift form

$$H(-z) = H(z) |_{z=-z} \quad (5)$$

(Step 2) Transformation a shift form to a delta form by symbolic computation or hand calculation

The delta operator is usually defined by  $(z - 1)/T$ , as  $T$  is sampling period. However, in this procedure the delta operator is defined by  $\delta' = -z - 1$ . This modified delta operator can reduce a numerical sensitivity of a control algorithm as usual delta operator.

$$\begin{aligned} H(\delta') &= H(-z) |_{z=-\delta'-1} \\ &= k \frac{(\delta'^2 + b_1(\delta' + 1) + b_2)}{(\delta'^2 + a_1(\delta' + 1) + a_2)} \\ &= k \frac{\delta'^2 - b''_1 \delta' + b''_2}{\delta'^2 + a''_1 \delta' + a''_2} \end{aligned} \quad (6)$$

where

$$\begin{aligned} a''_1 &= a_1 + 2 \\ a''_2 &= a_1 + a_2 + 1 \\ b''_1 &= b_1 + 2 \\ b''_2 &= b_1 + b_2 + 1 \end{aligned}$$

(Step 3) Introduction of two time scales  $T_1$  and  $T_2$   
Because the delta inverse operation is a discrete-time integrator, there are  $n$  integrators of gain  $T$  in a  $n$ th order controller. In order to adjust the magnitude of the integral value,  $n$  time scales  $T_1, T_2, \dots, T_n$ , which are design parameters, are introduced. By the help of equivalent transformation of block diagram, each coefficients of an control algorithms must be divided by  $T_1, T_1 T_2, T_1 T_2 T_3$ , respectively. Thus, each coefficient of a control algorithm can be adjusted independently by  $T_j$ . From (6) a control algorithm based on modified delta operator is derived.

$$\begin{aligned} e'_k &= (-1)^{-k} e_k \\ x'_k &= e'_k - m(a'_1 x'_k) - m(a'_2 x'_k) \\ u'_k &= x'_k + \bar{m}(b'_1 x'_k) + \bar{m}(b'_2 x'_k) \\ u''_k &= -\frac{u'_k - u'_{k-1}}{2} \\ \bar{u}_k &= (-1)^{-k} u''_k \\ u_k &= \bar{m}(k \bar{u}_k) \\ x'_{k+1} &= -x'_k - \bar{m}(T_2 x'_k) \\ x'_{k+1} &= -x'_k - \bar{m}(T_1 x'_k) \\ u'_{k-1} &= u'_k \end{aligned} \quad (7)$$

where

$$\begin{aligned} a'_1 &= \frac{a''_1}{T_1}, \quad b'_1 = \frac{b''_1}{T_1} \\ a'_2 &= \frac{a''_2}{T_1 T_2}, \quad b'_2 = \frac{b''_2}{T_1 T_2} \\ m(y) &= \frac{y - 2^{n-2}}{2^{n-1}}, \quad \bar{m}(y) = \frac{y + w + 2^{n-2}}{2^{n-1}} \end{aligned}$$

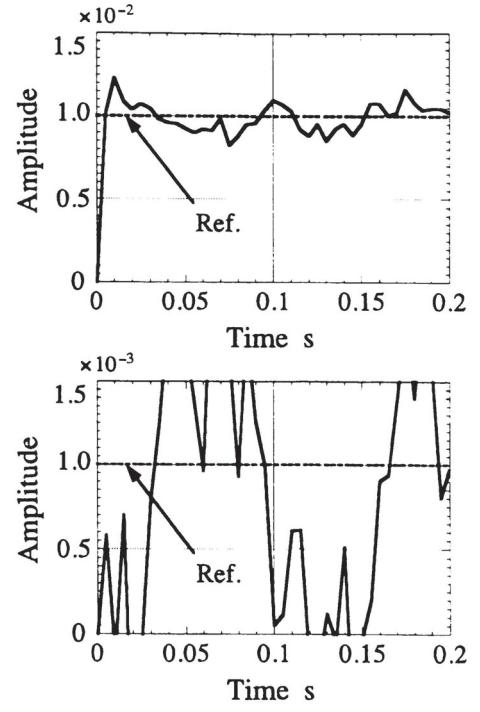


Fig. 5. Step response using fixed-point block(Shift form)

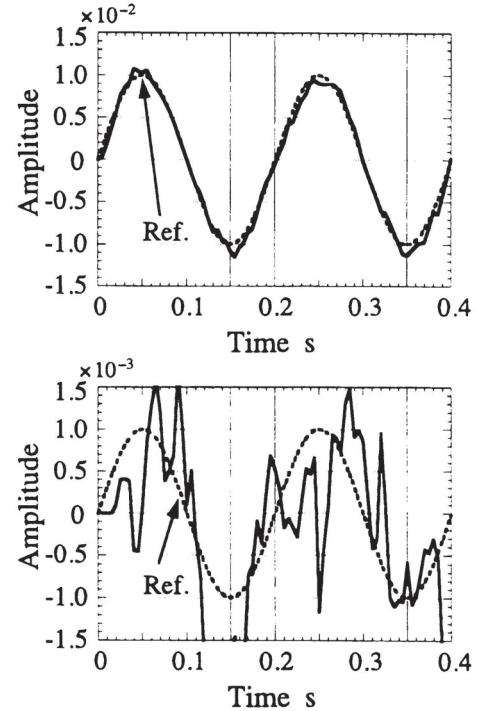


Fig. 6. Sinusoidal response using fixed-point block(Shift form)

1) One bias

$$w = 2^{16-3}$$

2) Two biases

$$w_1 = 3 \times 2^{16-4}, w_2 = 2^{16-4}$$

3) Four biases

$$w_1 = 7 \times 2^{16-5}, w_2 = 5 \times 2^{16-5}$$

$$w_3 = 3 \times 2^{16-5}, w_4 = 2^{16-5}$$

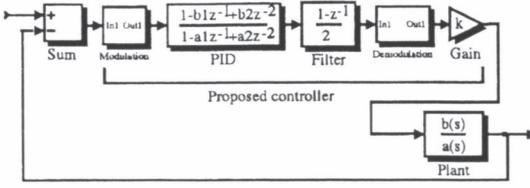


Fig. 7. PID controlling system(Modified delta form)

These biases can be regarded as inputs of the integral calculation. Thus, biases change such as  $w_1, w_1, w_4, w_4, w_2, w_2, w_3, w_3$  every  $T$  seconds in order to cancel the transient responses to these biases. Bias generator block is shown in Fig. 11. In addition, Eq. (7) can be divided into two groups of equations whether  $k$  is odd or even number. Thus, by using 'flag jump' the program becomes very simple and the calculation steps are almost same as those of the conventional delta operation.

if flag >= 0

$$\begin{aligned} x_k^0 &= e_k - m(a'_1 x_k^1) - m(a'_2 x_k^2) \\ u'_k &= x_k^0 + \bar{m}(b'_1 x_k^1) + \bar{m}(b'_2 x_k^2) \\ u''_k &= -\frac{u'_k - u'_{k-1}}{2} \\ u_k &= \bar{m}(k u''_k) \\ x_{k+1}^2 &= -x_k^2 - \bar{m}(T_2 x_k^1) \\ x_{k+1}^1 &= -x_k^1 - \bar{m}(T_1 x_k^0) \\ u'_{k-1} &= u'_k \\ flg &= -flg \end{aligned}$$

else

$$\begin{aligned} x_k^0 &= -e_k - m(a'_1 x_k^1) - m(a'_2 x_k^2) \\ u'_k &= x_k^0 + \bar{m}(b'_1 x_k^1) + \bar{m}(b'_2 x_k^2) \\ u''_k &= \frac{u'_k - u'_{k-1}}{2} \\ u_k &= \bar{m}(k u''_k) \\ x_{k+1}^2 &= -x_k^2 - \bar{m}(T_2 x_k^1) \\ x_{k+1}^1 &= -x_k^1 - \bar{m}(T_1 x_k^0) \\ u'_{k-1} &= u'_k \\ flg &= -flg \end{aligned} \quad (8)$$

(Step 4) Modeling a controller using floating-blocks  
A block diagram of controller, whose structure is the canonical direct form 2. The canonical direct form 2 is usually adopted in a delta operation because of minimal amount of calculations.

(Step 5) Developing fixed-point delta inverse block for the modified delta form

In order to realize a delta form, the shift inverse is replaced by the delta inverse, which is a discrete integrator. For V.M.M. the bias  $w$  is introduced as like the bias for rounding in fixed-point arithmetic. Fixed-point delta inverse block is realized by fixed-point unit delay block, add block, sub block, and gain block, which are provided in Fixed-Point Blockset library. In order to realize V.M.M., the unit delay output should feedback negatively, not positively in the delta inverse. This negative feedback makes the output switch zero or unity every  $T$  seconds like PWM. This bias and negative feedback are the key in V.M.M.. The number of biases is determined by a specification.

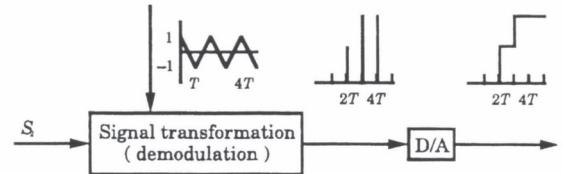
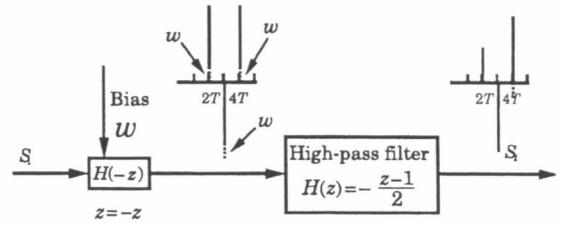
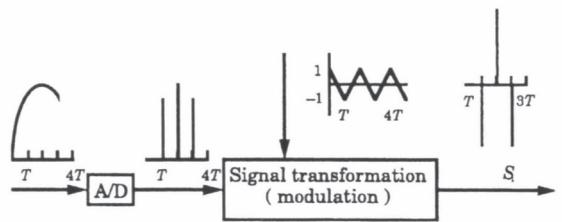


Fig. 8. Digital controller with V.M.M.

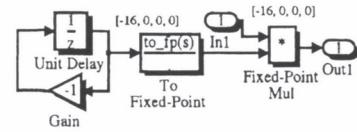


Fig. 9. Modulation and demodulation block

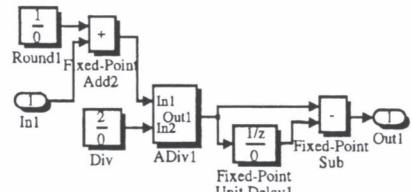


Fig. 10. Filter block for V.M.M.

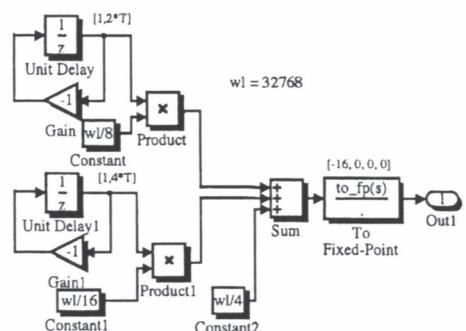


Fig. 11. Bias generator block

### 3.2 Proposed design procedure

(Step 1) Simulation of a closed-loop system using floating-blocks

Two coefficients  $a'_j$  and  $b'_j$  are determined by one  $T_j$  simultaneously, not independently. Therefor, the control algorithm is separated by two groups, namely, the  $a'_j$  part and the  $b'_j$  part.

#### [1] Adjustment of $T_j$ and $a'_j$

Design parameters  $T_j$  are adjusted sequentially from  $T_1$  so that the magnitude of the internal variable  $x_k^j$  becomes less than unity to prevent of overflow. On the other hand, the magnitude of  $T_j$  is set as large as possible to prevent underflow and quantization noise. Also, the value of  $T_j$  should be set to be quantization step multiple. Because the magnitude of  $a'_j$  is varied by  $T_j$ , it should be verified whether the magnitude of  $a'_j$  exceeds unity for fixed-point arithmetic. Design parameters  $T_j$  are adjusted sequentially from  $T_1$  to prevent overflow of the delta inverse. Conversely, the magnitude of  $T_j$  is set as large as possible to prevent underflow and quantization noise. Also, the value of  $T_j$  should be set to be quantization step multiple. It should be verified whether the magnitude exceeds unity for fixed-point calculations. Though this step is the iterative design cycle, auto-tuning can be possible using command procedure within MATLAB.

#### [2] Normalization of $b'_j$

The coefficients  $b'_j$  are divided by  $b'_{j\max}$  so that the magnitude of them is less than unity for fixed-point arithmetic. The controller gain  $k$  is multiplied by  $b'_{j\max}$ .

#### (Step 2) Scaling

From Step 1 time scales  $T_1, T_2$  and coefficients of an algorithm are determined as floating-point numbers. For a fixed-point implementation of a control algorithm on microprocessors these numbers must be converted to integer ones. Because the magnitude of these parameters is less than unity, usual tedious scaling is not necessary.

#### (Step 3) Simulation of a closed-loop system

A controller is simulated in floating-point arithmetic, and a plant is simulated in fixed-point arithmetic. Step and sinusoidal responses are shown in Fig. 12 and Fig. 13. Here, one bias is used in Fig. 12 and Fig. 13, four biases are used in Fig. 14 and Fig. 15. From these figures the numerical advantages of V.M.M. are verified for wide range of reference. As the number of biases increases, calculation errors decrease. In cases of a steady state input, this methodology achieves good performance. From the PWM point of view, as the number of biases increases, the total sampling period increases. In order to follow rapidly changing inputs, the number of biases must be limited, or the sampling period must be short. However, the decrease of the sampling period makes the magnitude of time scale small. This causes underflow at delta inverse. Therefore, the tradeoff between the numerical accuracy and the sampling period is required.

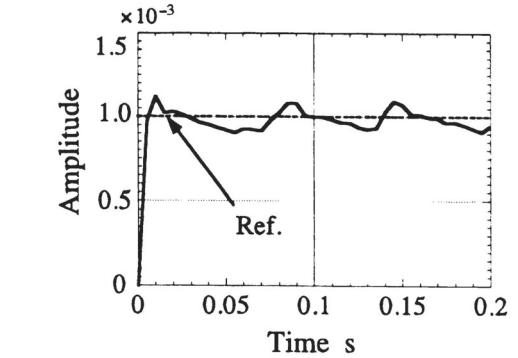
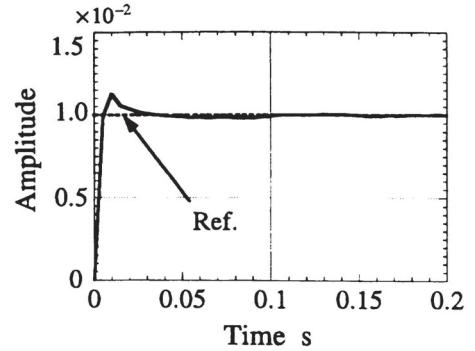


Fig. 12. Step response using fixed-point block(V.M.M. with one bias)

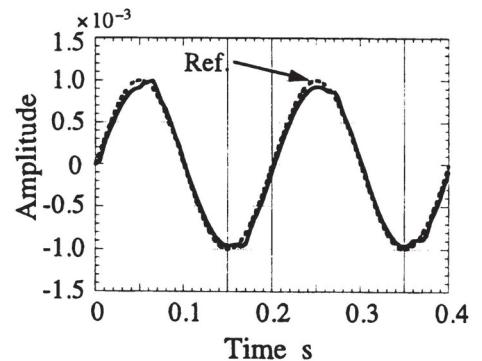
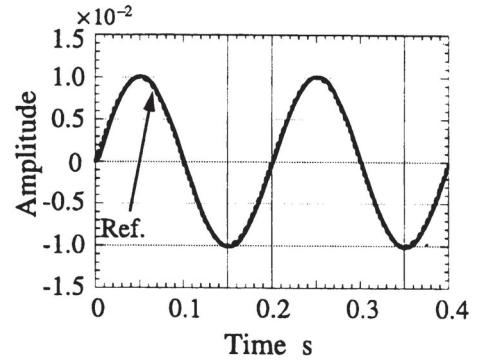


Fig. 13. Sinusoidal response using fixed-point block(V.M.M. with one bias)

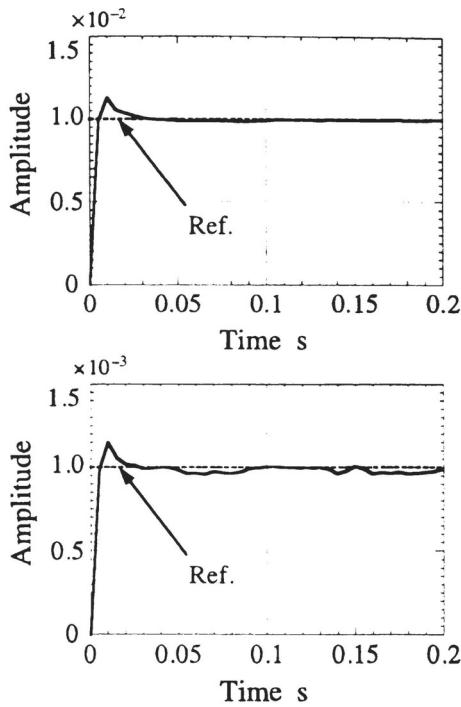


Fig. 14. Step response using fixed-point block(V.M.M. with four biases)

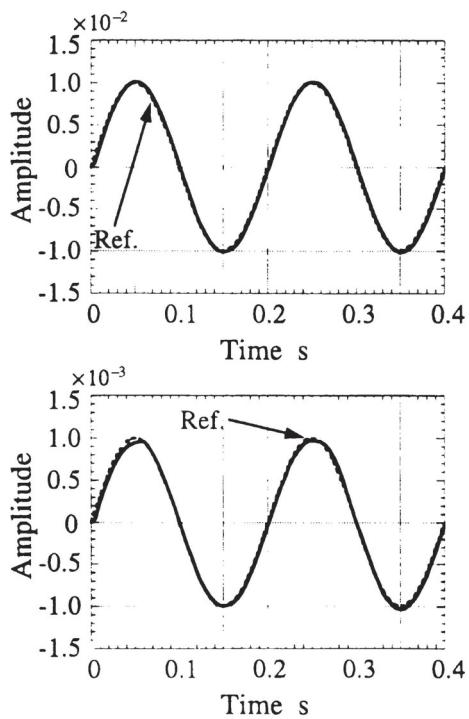


Fig. 15. Sinusoidal response using fixed-point block(V.M.M. with four biases)

#### 4. CONCLUSIONS

In order to establish a modified delta operation in fixed-point arithmetic, the following results have been obtained.

(1) The implementation procedure of V.M.M. has been proposed generally. Therefor, this methodology can be widely available. Simulation results show that its availability have been made clear.

(2) The calculation steps required by V.M.M. are almost same as usual delta operation. The extra calculation steps are the execution of high-pass filter and 'flag jump' for modulation in a control program. These calculation steps are negligible for usual microprocessors.

#### 5. REFERENCES

- Agarwal, R.C. and C.S. Burrus (1971). New recursive digital filter having very low sensitivity and round-off noise. *IEEE Trans. CAS* **22**(12), pp. 921–927.
- Aoki, T. (1999). Implementation of modified delta form for microprocessors using fixed-point arithmetic. *Proc. of American Control Conf.* **6**, pp. 4056–4060.
- Aoki, T. and Y. Furukawa (1996). Proposal of modified delta operation with v. m. m. and its application to controlling algorithm in fixed-point arithmetic. *Proc. of Fourth Int. Conf. Control, Automation, Robotics and Vision (ICARCV'96)* **3**, pp. 2356–2360.
- Aoki, T., N. Moronuki and Y. Furukawa (1997). A study on controlling algorithm to realize high-speed & high-accuracy control systems - proposal of modified delta operator -. *Journal of Robotics and Mechatronics* **9**(6), pp. 446–454.
- Goodall, R.M. (1985). High-speed digital controllers using an 8bit microprocessor. *Software & Microsystems* **4**(5/6), pp. 246–250.
- Goodall, R.M. (1989). Minimisation of computation for digital controllers. *Trans. Inst MC* **11**(5), pp. 218–224.
- Goodall, R.M. (1990). The delay operator  $z^{-1}$  - inappropriate for use in recursive digital filters?. *Trans. Inst MC* **12**(5), pp. 246–250.
- Goodwin, G.C., R.H. Middleton and H.V. Poor (1992). High-speed digital signal processing and control. *Proc. The IEEE* **80**(2), pp. 240–259.
- Grace, A., A.J. Laub, J.N. Little and C. Thompson (1990). *Control System Toolbox User's Guide*. The MathWorks. MA.
- Middleton, R.H. and G.C. Goodwin (1986). Improved finite word length characteristics in digital control using delta operators. *IEEE Transactions on Automatic Control* **31**(11), pp. 1015–1021.
- Middleton, R.H. and G.C. Goodwin (1990). *Digital Estimation and Control: A Unified Approach*. Prentice-Hall. New Jersey.
- Saadat, H. (1993). *Computational Aids in Control Systems Using MATLAB*. McGraw-Hill. New Jersey.