

컴퓨터의 간헐적 차단시 디지털 제어시스템의 알고리즘 연구와 시뮬레이션

A Study on the Algorithm for Digital Control System with Computer Interruption and its Simulation

張 成 煥* · 杜 坪 洙** · 趙 賢 燮***
(Seong-Hwan Jang · Pyeng-Soo Doo · Hyun-Seop Cho)

Abstract - The digital computer control systems which are used in time sharing control or remote control may experience intermittent interruptions. Further it can be made unstable by the intermittent interruptions of a control computer. This paper presents a mathematical model with intermittent computer interruptions and an unstable output response through simulation. Also it presents to improve the control quality using extrapolation method in interrupt synchronizing feedback path.

Key Words : Interrupt, 디지털컴퓨터 제어시스템, 역보간법.

1. 서 론

시스템의 제어를 위하여 과거에는 아날로그 제어기를 사용하였으나 마이크로 프로세서가 개발되어 마이크로 컴퓨터가 급속하게 발전되고 또한 신뢰성이 향상 되므로써 우주선 및 비행기와 로봇, 선박, 각종 공정 제어 등과 같이 산업 전반에 걸쳐서 디지털 컴퓨터를 제어기로 사용하고 있는

추세에 있으며 특히 디지털 컴퓨터를 제어시스템에 사용하므로 실시간 처리가 엄격하게 요구되는 공정을 제어할 수 있을 뿐 아니라 그에 수반되는 부수적인 작업들 까지도 처리가 가능하게 되었다. [1~4]

일반적으로 컴퓨터 제어 시스템은 공정과 공정의 데이터 입출력을 위한 interface장치, 물리량을 전기적 신호로 변환하여 주는 검출장치, 보조기억장치나 프린터와 같은 주변장치들과 연동하여 대상공정을 실시간 on-line 제어가 되도록 DDC (direct digital control), hierachy, supervisory 와 같은 형태로 사용되고 있으나 소형 제어 시스템에서는 Uniprocessor형의 DDC를 주로 사용하며 방

*正 會 員 : 圓光大 工大 電氣工學科 教授 · 工博
 **正 會 員 : 全北産業大 電子工學科 助教授
 ***正 會 員 : 圓光大 大學院 電氣工學科 碩士課程
 接受日字 : 1991年 7月 23日
 1次修正 : 1992年 4月 3日
 2次修正 : 1992年 5月 8日

대한 정보의 계산이 필요한 곳에서는 pipe-line이나 multi-processor를 이용한 병렬 처리 시스템을 사용하기도 한다. [5~7]

컴퓨터 제어시스템의 구성에 필수적인 도구인 마이크로 컴퓨터 사용의 증가와 함께 cpu의 제어 신호인 interrupt의 사용 방법도 다양해져 cpu의 공전시간(idle time) 감소를 위한 입출력 장치와의 제어 신호 및 응급 사태 처리를 위한 개시 신호, 특정 서브루틴 처리를 위한 신호, 직접 기억장치 구동(direct memory access) 등과 같이 다양한 용도로 사용이 되고 있는 실정에 있다. [5~8]

그러나 컴퓨터 제어시스템이 다른 작업과 병용되는 방식에서 좀더 중요한 작업에 비해 처리 우선 순위가 낮거나, 원격 제어 시스템에서 데이터 전송도중 전송 매체에서의 방해 현상으로 데이터 전송이 중단되는 경우, 시분할 제어 시스템에서 어떤 공정에서의 interrupt요구에 의해 할당시간(time slice)이 커진 경우 등에서는 새로운 정보의 입력과 제어 법칙의 연산이 지연되므로 공정은 정상적인 경우와는 달리 불안정한 현상을 유발할 것으로 예상된다. [9~12]

이에 대하여 Lin은 상태궤환 제어시스템에서 상태는 계속 유지되면서 데이터만 간헐적으로 차단이 될 때 Jury의 안정도 판별법에 의한 주궤환 경로의 이득 범위를 제시하였고[9], Rekasius는 Lin과 같은 상황에서 Lyapunov안정도 판별법을 근거로 시스템이 안정하기 위한 충분 조건을 제시하였고, [10] 장성환은 Lin과 Rekasius와 같은 상황을 근거로 하여 deadbeat제어기에서 발생할 수 있는 현상에 대하여 고찰하였으나[11] 이들이 고찰한 현상은 실증되지 못했으며 또한 하나의 샘플링 동안만 간헐적으로 차단이 되는 경우에만 적용하였다.

그렇지만 DMA에 의한 다량의 정보전송등과 같이 interrupt를 요구한 작업에 따라서는 다수개의 샘플링 동안 계속하여 차단이 되어야 하므로 하나의 샘플링시간만 차단이 되는 경우와는 다르며, 컴퓨터에서는 interrupt에 의해 상태가 변하고 데이터의 입출력이 차단되기 때문에 적용이 곤란하게 되었다.

본 논문에서는 interrupt가 있는 컴퓨터제어시스템의 해석 및 분석이 가능하도록 interrupt이 하나 이상의 샘플링 시간에서 계속되는 경우에 대한 제반조건들을 제시하고 정상적 동작을 할 경우와 interrupt가 걸린 경우를 시뮬레이션하여 비교한 후 interrupt에 동기되는 궤환 경로를 두고 여기에 역보간법(extra polation)을 도입하여 interrupt의

영향을 감소시킬 수 있도록 한후 시뮬레이션을 통해 이를 입증하였다.

2. 컴퓨터 제어 시스템의 수학적 모델

그림1과 같은 구조를 갖는 제어 시스템의 수학적 모델을 구하기 위해 공정은 가관측 가제어 되는 선형 시불변 시스템으로서 다음과 같은 동적 특성을 가지고 있다면

$$\begin{aligned} \dot{X} &= FX + Gm \\ y &= HX \end{aligned} \tag{1}$$

이것은

$$\begin{aligned} A_p &= e^{FT} = L^{-1}[(SI - F)^{-1}] \\ B_p &= \int_0^T e^{F(T-t)} G dt \\ C_p &= H \end{aligned} \tag{2}$$

단 L^{-1} 은 Laplace역변환을 뜻하고, T 는 샘플링 시간을 뜻함.

에 의하여 이산 상태 방정식으로 변경할 수 있다.

$$\begin{aligned} X_{p(k+1)} &= A_p X_{p(k)} + B_p m(k) \\ y(k) &= C_p X_{p(k)} \end{aligned} \tag{3}$$

그리고 제어기의 동적 특성이

$$\begin{aligned} X_{c(k+1)} &= A_c X_{c(k)} + B_c e(k) \\ u(k) &= C_c X_{c(k)} + D_c e(k) \end{aligned} \tag{4}$$

이면

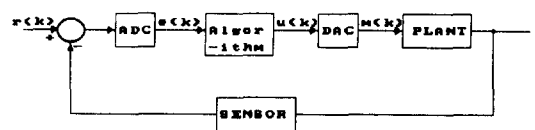


그림 1 컴퓨터 제어 시스템의 블록도
Fig. 1 Block diagram of computer controlled system

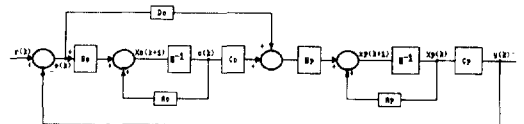


그림 2 제어시스템의 블록도
Fig. 2 Block diagram of cotrol system

$$\begin{aligned} e_{(K)} &= r_{(K)} - y_{(K)} \\ u_{(K)} &= m_{(K)} \end{aligned} \quad (5)$$

이므로 이것들을 정리하면 다음과 같은 차수가 증가한(augment) 행렬을 구할 수 있다.

$$\begin{aligned} X_{(K+1)} &= AX_{(K)} + Br_{(K)} \\ y_{(K)} &= CX_{(K)} \\ X_{(K)} &= \text{col}\{X_{P(K)}, X_{C(K)}\} \\ A &= \begin{bmatrix} A_P - B_P D_C C_P & B_P C_C \\ -B_C C_P & A_C \end{bmatrix} \\ B &= \begin{bmatrix} B_P D_C \\ B_C \end{bmatrix} \\ C &= [C_P \quad 0] \end{aligned} \quad (6)$$

식 (3)~(5)를 블록도로 표시하면 그림2)와 같다.

2.1 정상적인 동작시의 컴퓨터 제어시스템의 수학적 모델

초기상태가 $X_{(0)}$ 이면 임의시간 K 에서의 상태와 출력은 반복적 연산에 의하여 다음과 같이 표현된다.

$$\begin{aligned} X_{(1)} &= AX_{(0)} + Br_{(0)} \\ X_{(2)} &= AX_{(1)} + Br_{(1)} \\ &= A^2 X_{(0)} + AB r_{(0)} + Br_{(1)} \\ &\vdots \\ X_{(K+1)} &= A^{K+1} X_{(0)} + \sum_{i=0}^K A^i Br_{(K-i)} \end{aligned} \quad (7)$$

$$\begin{aligned} y_{(0)} &= CX_{(0)} \\ y_{(1)} &= CX_{(1)} \\ &= CA X_{(0)} + CB r_{(0)} \\ &\vdots \\ y_{(K)} &= CA^K X_{(0)} + \sum_{i=0}^{K-1} CA^i Br_{(K-i-1)} \end{aligned} \quad (8)$$

2.2 interrupt시의 수학적 모델

컴퓨터가 정상동작중 interrupt이 되면 cpu는 상태 레지스터를 비롯한 모든 레지스터들의 내용들을 stack에 push한 후 하드웨어적으로 고정되거나 interrupt vector에 의해 지정된 서브루틴으로 분기를 하여 interrupt을 요구한 장치가 필요로 하는 작업을 수행하게 되므로 어느 샘플링 시간 동안에는 새로운 데이터를 입력하지 못하고 제어법칙 연산도 불가능 할뿐 아니라 공정을 안정하게 제어할 수 있는 조작량이 출력되지 못하여 공정에서는 정상적인 동작이 되지 못하게 될 것이다.

어떤 임의의 샘플링 시간 N 에서 interrupt이 발생하면 $(N-1)$ 까지는 정상적인 동작을 하므로

$$X_{in(N)} = X_{(N)} = A^N X_{(0)} + \sum_{i=0}^{N-1} A^i Br_{(N-1)} \quad (9)$$

$$\begin{aligned} y_{in(N-1)} &= y_{(N-1)} = CA^{N-1} X_{(0)} \\ &\quad + \sum_{i=0}^{N-2} CA^i Br_{(N-1-i)} \end{aligned}$$

가 되며 interrupt이 N 에서 $N+i$ 까지 계속된다면 $D-A$ (ZOH)의 영향으로 조작량 $m_{(N)}$ 만이 한 샘플링 시간동안 데이터를 유지할 뿐 나머지 변수들은 활용되지 못하게 되므로

$$\begin{aligned} X_{C(N+i)} &= 0 \quad (i=0, 1, 2, \dots) \\ r_{(N+i)} &= 0 \quad (i=0, 1, 2, \dots) \\ m_{(N+i)} &= m_{(N-1)} \quad (i=0) \\ m_{(N+i)} &= 0 \quad (i=1, 2, 3) \end{aligned} \quad (10)$$

이 된다.

이 조건을 식 7) 8)에 대입하면

$$\begin{aligned} X_{in(N+1)} &= AX_{in(N)} + Br_{(N)} \\ &= AX_{in(N)} \\ X_{in(N+2)} &= AX_{in(N+1)} + Br_{(N+1)} \\ &= AX_{in(N+1)} = A^2 X_{in(N)} \\ &\vdots \\ X_{in(N+i)} &= AX_{in(N+i-1)} + Br_{(N+i-1)} \\ &= A^i X_{in(N)} \end{aligned} \quad (11)$$

여기에서 $X_{in(N)} = \text{col}\{X_{P(N)}, 0\}$ 이다.

$$\begin{aligned} y_{in(N)} &= CX_{in(N)} = CX_{(N)} = y_{(N)} \\ y_{in(N+1)} &= CX_{in(N+1)} = CA X_{in(N)} \\ y_{in(N+2)} &= CX_{in(N+2)} = CA^2 X_{in(N)} \\ &\vdots \\ y_{in(N+i-1)} &= CA^i X_{in(N)} \end{aligned} \quad (12)$$

가 되며 interrupt이 종료되면 cpu는 stack에 저장했던 데이터를 pop하여 제어법칙으로 복귀되어 interrupt을 발생했던 다음 번지에 있는 명령에서부터 정상적으로 수행하므로 A/D 및 D/A를 통한 입출력 및 제어법칙 연산이 가능하게 된다.

따라서

$$\begin{aligned} X_{C(N+i+1)} &= X_{C(N)} \\ r_{(N+i+1)} &= r_{(N+i+1)} \\ m_{(N+i+1)} &= m_{(N+i+1)} \end{aligned} \quad (13)$$

이 되어 상태는

$$\begin{aligned} X_{in(N+i+1)} &= AX_{(N)} + Br_{(N+i+1)} \\ &= A^{N+1} X_{(0)} + \sum_{i=0}^N A^i Br_{(N-1)} \\ X_{in(N+i+2)} &= AX_{in(N+i+1)} + Br_{(N+i+2)} \\ &= A^{N+2} X_{(0)} + \sum_{i=0}^{N+1} A^i Br_{(N-1)} \\ &\vdots \\ X_{in(K+1)} &= AX_{i(K)} + Br_{(K)} \\ &= A^{K+1-i} X_{(0)} + \sum_{i=0}^{K-i} A^i Br_{(K-1)} \end{aligned} \quad (14)$$

이 되며 출력은

$$\begin{aligned}
 y_{in(N+i)} &= CX_{(N)} \\
 &= CA^N X_{(0)} + \sum_{i=0}^{N-1} CA^i B r_{(N-i-1)} \\
 y_{in(N+i+1)} &= CX_{(N+i+1)} \\
 &= CA^{N+1} X_{(0)} + \sum_{i=0}^N CA^i B r_{(N-i-1)} \\
 y_{in(k)} &= CA^{k-i} X_{(0)} + \sum_{i=0}^{k-1} CA^i B r_{(k-i-1)} \quad (15)
 \end{aligned}$$

가 된다.

따라서 연속적으로 연산을 수행하는 컴퓨터 제어시스템에서 interrupt이 존재할 때의 수학적 모델은 식(9)~(15)를 정리하면 다음과 같이 표현할 수 있다.

$$\begin{aligned}
 X_{in(k-1)} &= A^{k+1-i} X_{(0)} + \sum_{i=0}^n A^i B r_{(k-1)} \\
 &\quad + \sum_{i=1}^i A^i X_{in(N)} \quad (16) \\
 y_{in(k)} &= CA^{k-i} X_{(0)} + \sum_{i=0}^{k-1} CA^i B r_{(k-i-1)} \\
 &\quad + \sum_{i=1}^i CA^i X_{in(N)}
 \end{aligned}$$

식(7), (8)과 (16)을 비교하여 보면 정상적으로 제어법칙을 수행한 경우와 도중에 interrupt이 요구되어 처리된 결과는 상이하며 interrupt이 걸린 경우에는 공정상태와 연속된 interrupt의 갯수에 따르는 항이 존재하며 이것이 제어의 질을 나쁘게 한다.

3. interrupt감소 알고리즘

interrupt에 의해 발생하는 진동을 줄이기 위해

서 interrupt과 동기되는 제환회로를 두어 보상하는 방법에는 여러가지 방법이 있을 것으로 예상되지만, 이 회로는 주제어기 보다는 간단한 구조를 갖어야 하므로 알고리즘이 간단 하여야 한다. 여기에서는 간단한 알고리즘으로 안정한 출력을 얻을 수 있는 역보간법(extra polation)을 사용하여 출력을 예측한다. 예측된 출력은

$$\begin{aligned}
 \hat{y}_{in(N+1)} &= 2y_{in(N)} - y_{in(N-1)} \\
 &= 2y_{(N)} - y_{(N-1)} \\
 &\quad \vdots
 \end{aligned}$$

$$\hat{y}_{in(N+i-1)} = 2\hat{y}_{in(N+i-1)} + \hat{y}_{in(N+i-2)} \quad (17)$$

이 되며 정상인 경우와 일치되어야 하므로

$$y_{(N+i-1)} = \hat{y}_{in(N+i-1)} \quad (18)$$

이 되어야 한다.

따라서

$$\begin{aligned}
 \hat{y}_{in(N+i-1)} &= y_{(N+i-1)} = C_p X_p(N+i-1) \\
 &= C_p A_p X_p(N+i-1) + C_p B_p m(N+i-1) \quad (19)
 \end{aligned}$$

이 되므로

$$\begin{aligned}
 m(N+i-1) &= (C_p B_p)^{-1} \hat{y}_{in(N+i-1)} \\
 &\quad - (C_p B_p)^{-1} C_p A_p X_p(N+i-1) \quad (20)
 \end{aligned}$$

의 관계를 만족하면 된다.

상기한 결과에 의해 interrupt시 오차를 줄일 수 있는 제어시스템의 블록도를 그리면 그림(3)과 같다.

4. 시뮬레이션 및 결과검토

시뮬레이션 및 결과검토는 Salama에 의해 기설계된 제어법칙과 공정모델을[13] 근거로하여 정상

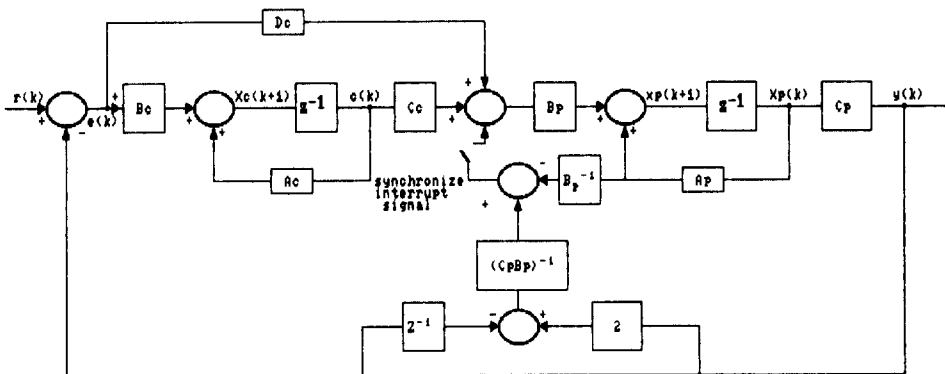


그림 3 interrupt영향 감소 제어시스템의 블록도
 Fig. 3 Block diagram of control system which reduce interrupt effect

적인 동작을 시킨 결과와 2장에서 살핀 제반조건들을 대입하여 동작시킨 interrupt시의 결과를 비교한 후 interrupt효과를 감소시키도록 3장에서 유도한 수학적 모델을 적용시켜 그 결과를 관찰하였다. 이를 위한 흐름도를 그림(4)에 보였으며 사용된 계수는 다음과 같다.

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

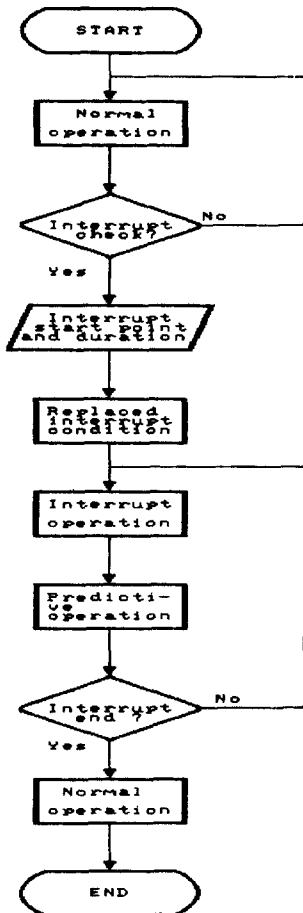


그림 4 모의를 위한 흐름도
Fig. 4 flow chart for simulation

$$B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -1 \\ -1 \\ -1 \end{bmatrix} \quad C = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \\ 1 & 0 \\ 1 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}^T$$

Salama가 설계한 PID제어기의 출력을 그림(5)

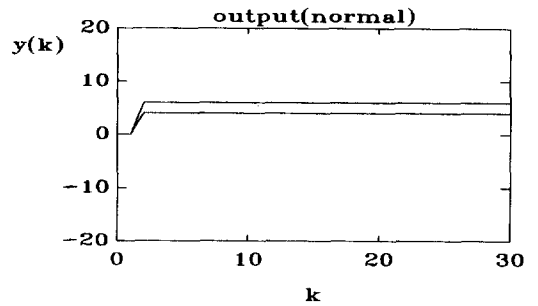


그림 5 Salama가 설계한 제어기의 출력
Fig. 5 Output designed by Salama

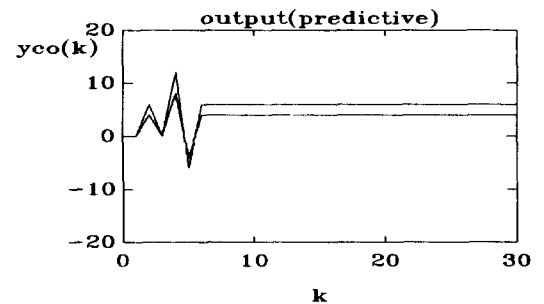
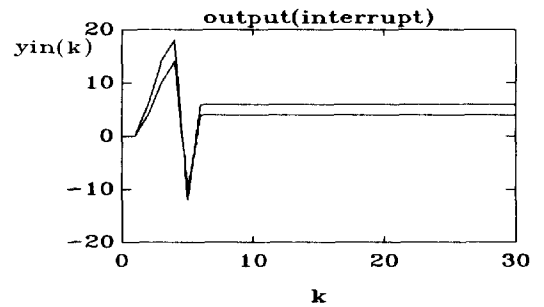


그림 6 k=2, 3에서 interrupt이 발생했을 때의 출력
Fig. 6 Output occurred interrupt at k=2,3

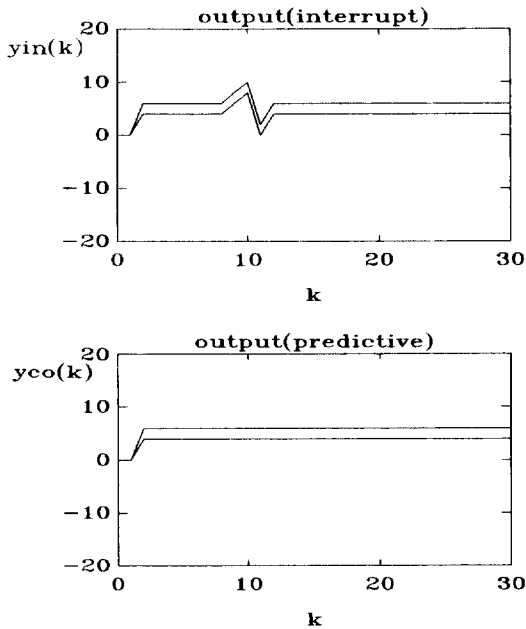


그림 7 $k=7, 8, 9$ 에서 interrupt이 발생했을 때의 출력

Fig. 7 Output occurred interrupt at $k=7, 8, 9$

에 보였으며 $k=2, 3$ 가 $K=7, 8, 9$ 에서 interrupt이 발생했을 때의 출력을 그림(6)과 (7)에 보였다. 그림(6)과 (7)에서 interrupt에 의해 출력이 동요되어 시스템이 불안정 하여지며, interrupt와 동기되는 케환경로에 역보간법을 사용하였을 경우의 출력은 interrupt의 시점에 따라 과도기에서 시작이 된 경우에는 역보간법을 사용하더라도 안정되지 않았고 그림(6), 정상기에서 시작이 된 경우에는 interrupt와는 무관하게 안정함을 보인다 그림(7). 그리고 interrupt가 제거되더라도 바로 안정되지 못하고 시간이 경과 된후에 비로소 안정이 됨을 알 수 있다. 이상의 결과들을 요약하면 다음과 같다.

- 1) interrupt는 공정출력에 동요를 발생시켜 불안정 하게 한다.
- 2) 불안정한 정도는 과도기와 정상기에 따라 다르게 나타난다.
- 3) interrupt와 동기되는 케환경로와 역보간법을 사용하므로써 정상기에서의 interrupt의 영향을 줄일 수 있으나 과도기에서는 문제점을 가지고 있다.

5. 결 론

본 연구에서는 디지털컴퓨터를 제어기로서 사용하였을시 interrupt가 공정에 미치는 영향을 고찰하고저 제어시스템의 수학적모델을 도출하고 시뮬레이션 하였으며 또한 제어의 질을 향상시킬 수 있도록 interrupt에 동기되는 케환 경로를 두고 여기에 역보간법을 도입하여 시뮬레이션을 행하였다.

시뮬레이션한 결과를 통해 컴퓨터의 interrupt는 공정에 동요를 유발케 하는 불안정의 요소가 되며, interrupt와 동기되는 케환경로에 역보간법을 사용하므로써 정상기에서는 안정시킬 수 있음을 확인할 수 있었다.

앞으로 interrupt에 강한 제어알고리즘의 개발과 과도기에서도 안정시킬 수 있는 간단한 케환알고리즘에 대한 연구가 계속되어야 할 것으로 사료된다.

참 고 문 헌

- [1] Karl J. Astrom and Bjorn Wittenmark, "Computer controlled systems," Prentice-hall Inc., 1984
- [2] B.C. Kuo, "Digital control system," Holt-reinehard and Winston Inc., 1980
- [3] Paul. Katz, "Digital control using micro-processors," Prentice-hall Inc., 1981
- [4] Ralstonn, "Encyclopedia of computer science and engineering," Van norstrand reinhold, 1976
- [5] 최동진, "계산기제어시스템," 도서출판 세화, 1989
- [6] H. Hanselmann, "Implementation of digital controllers-A survey," automatica, Vol. 23, No. 1, pp. 7~32, 1987
- [7] Bimal K, Bose, "Microcomputer control of power electronics and drivers," IEEE press, pp. 3~22, 1987
- [8] M. Morris, Mano, "Computer architecture," Prentice-hall Inc, 1982
- [9] T.S. Lin, "Computer control with interruptions," MS theis of technological institute North Weston university, June, 1983
- [10] Z.V. Rekasius, "Stability of digital control with interruptions," IEEE trans on aut. con, Vol. ac-31, No. 4, pp. 356~359, April, 1986
- [11] 장성환, "제어용 컴퓨터 차단시 이산치 제어 시스템의 안정도," 대한 전기학회 논문집,

Vol. 36, No. 7, pp. 30~37, 1987, 7

- [12] 두평수의 2인, "컴퓨터의 간헐적 차단시 디지털 제어시스템의 알고리즘 연구와 시뮬레이션," 1991년 하계학술대회 논문집, p. 885~887, 대한 전기학회
- [13] Ahamed, I.A. Salama, "Design techniques of digital PI and PID controller for linear discrete time multivariable systems," North

-Holland, Applied digital control, pp. 149~169, 1985

- [14] Graham C, Goodwin and Kwai Sang Sin, "Adaptive filtering prediction and control," Prentice-hall Inc, 1984
- [15] Ioan, Dore landau, "System identification and control design," Prentice-hall Inc, 1990

저 자 소 개



장성환(張成煥)

1950년 6월 5일생. 1974년 홍익대 공대 전기공학과 졸업. 1976~83년 연세대 대학원 전기공학과 졸업(석·공박). 1985~86년 미국 노스웨스턴 대학 전기 및 전산학과 객원교수. 현재 원광대 공대 전기공학과 교수.



두평수(杜坪洙)

1952년 8월 19일생. 1984년 원광대 공대 전기공학과 졸업. 1986년 전북대 대학원 전기공학과 졸업(석사). 현재 원광대 대학원 전기공학과 박사과정 및 전북산업대 전자공학과 조교수.



조현섭(趙賢燮)

1965년 3월 15일생. 1990년 원광대 공대 전기공학과 졸업. 현재 원광대 대학원 전기공학과 석사과정.