

자동제어 교육에서의 컴퓨터 소프트웨어 이용

吳 俊 鎬

한국과학기술원 생산공학과 교수



●1954년 10월 3일생
●자동제어를 전공하고 있으며 그 중 특히 적응 제어 시스템, 로봇 및 로봇 제어, 공작기계 제어, 디지털시스템 등을 연구하고 있다.

1. 머리말

자동제어에서 다루어지는 분야는 시스템의 공학적 분석, 시스템의 수학적 모델링, 시스템의 동특성 및 안정성해석, 제어기 설계 및 성능분석, 기타 고등자동제어이론 등으로 대별될 수 있다. 이들을 다루기 위하여는 고전적으로 시간응답법, 근계적법, 주파수 특성법 등 여러 가지 해석적 방법이 동원되어 왔다. 그러나 이러한 접근방법들은 고차복소수함수 및 경우에 따라 비선형 문제를 포함하고 있어 일반해를 얻는 것은 거의 불가능한 경우가 많다. 이에 주로 근사해법이 사용되고 있으며, 과거에는 애널로그 컴퓨터를 이용한 시뮬레이션도 이용되었다. 그러나 60년대 이후 급격히 발달한 디지털컴퓨터 및 최근 소형컴퓨터의 광범위한 보급에 따라 과거에는 상상하기도 어려웠던 전산 시뮬레이션이 매우 간편하게 수행될 수 있게되었다. 뿐만 아니라 상업용, 연구용 혹은 교육용으로 개발되어 사용되는 자동제어시스템해석 전용 패키지는 그 효율을 배가시키고 있다⁽¹⁾. 이에 필자는 자동제어교육에서의 컴퓨터를 이용한 관점에서 몇가지를 살펴보고자 한다.

2. 자동제어교육의 문제점 및 컴퓨터 이용

60년대말 자동제어가 기계공학의 교과목으로

국내 일부대학에 설강되었을 때에 그 강의는 주로 전기공학 혹은 전자공학의 전공자들에 의하여 이루어졌다. 따라서 그 당시의 교과서 선택 및 교과내용은 그들에게 의존할 수 밖에 없었으므로 기계공학도들에게 적절한 흥미를 불러 일으키기에는 미흡했을 뿐 아니라 자동제어가 기계공학에 구체적으로 어떻게 기여할 수 있는지 이해하기에는 어려움이 많았다. 그러나 그후 로봇, NC공작기계등 기계-전자시스템이 복합된 대상물의 개념이 일반화 됨에 따라 모든 기계시스템의 자동화 혹은 개선에 적절한 형태의 제어기법의 도입이 필요하다는 인식이 보편화 되기 시작하였다. 이에 따라 근간에는 기계공학계열을 지원하는 학생의 상당수가 자동제어 및 관련분야를 선호하는 경향이 두드러지게 나타나고 있다. 대다수의 일반대학에서는 자동제어 혹은 관련 교과목을 개설하고 있으며, 그 수강생 역시 점차 증가하고 있는 추세이다. 그러나 이렇게 자동제어분야의 양적·질적 수요에도 불구하고 자동제어교육은 아직 구태의연한 과거의 틀에서 벗어나지 못하고 있다. 즉 자동제어 관련 교과서는 아직도 전기 및 전자공학자에 의하여 쓰여진것이 대부분일 뿐 아니라 교과내용 역시 그 범위를 벗어나지 못하고 있어 수강자의 흥미 및 관심도에 부응하지 못하고 있는 실정이다. 일례로 필자가 본원 기계공학계열 신입생 중 자동제어 과목을 수강하고 있는 학생을 상대로 면담을 실시하여 그 의견을 종합해 본 결과 자동제어에 대하여

「자동제어란 학문분야에 큰 호기심과 관심을 가지고 수강을 시작 하였으나 수강완료후 자동제어란 학문을 이해하는데 미흡 하였다. 결과적으로 수강중 배운 지식을 실제로 활용하는데에는 큰 어려움을 겪고있다」라고 하는 의견이 지배적 이었다. 면담결과 교과내용이 너무 이론에 치우쳐있어 개념파악이 힘들뿐 아니라 강의가 각 교과주제별로 내용이 분산되어 있어 전체를 이해할 수 없었던 데에서 그 이유를 찾을 수 있었다. 다시 말하여 지나치게 교과서에 따른 교과과정에 집착하여 정작 자동제어의 목적이라 할수있는 제어기 설계 및 시스템 구성에 대한 충분한 연습이 부족하게 된 것이다.

이에 자동제어를 강의함에 있어서 수강자들로 하여금 충분히 흥미를 느끼고, 나아가 확실한 개념을 정립시키기 위하여는 단순한 이론으로서의 자동제어가 아니라 사례연구(case study)를 포함한 실제 적용될 수 있는 자동제어로서의 강의가 이루어져야 하겠다. 이를 위하여는 교과주제별 강의시간 재 배정을 비롯한 교과내용 재검토와 실험, 컴퓨터의 활용등을 통한 자동제어 교육의 활성화가 이루어져야 할 것이다. 즉 시간응답성능시험, 안정성시험, 근계적작도, Nyquist선도, Bode선도 등 근사해법을 습득하는데 소요되는 시간을 컴퓨터에 이용함으로써 대폭 줄이고 이 시간을 컴퓨터에 의하여 구해진 각종 선도를 해독하고 이를 이용한 제어기 설계, 시스템 분석, 디지털 제어 등에 할애 함이 바람직 할 것이다.

3. 자동제어용 소프트웨어

자동제어용 소프트웨어는 일반 동역학 및 진동관련 소프트웨어와 상당부분 공통점을 가지고 있으나, 제어이론에 입각한 제어시스템 혹은 동적 시스템등을 해석, 시뮬레이션 할수있는 여러가지 기능이 더욱 강조되고 있다. 이를 소프트웨어는 사용자의 입장에서 볼때 다음과 같이 크게 세가지 형태로 구분될 수 있다⁽²⁾.

(1) 고급언어(FORTRAN, PASCAL 등)와

연결되어 사용되는 소프트웨어

주로 라이브러리 화일(library file) 형태로 되어있어 고급언어에서 서브루틴(subroutine) 혹은 함수(function)의 형식으로 사용된다.(예 : SIMON⁽³⁾ 등)

(2) 메뉴 타입 소프트웨어

갖가지 유용한 기능들을 메뉴에 따라 제한적으로 사용할 수 있는 소프트웨어 이다.(예 : CC 등)

(3) 시뮬레이션 전용 고급언어

BASIC, PASCAL 등과 유사한 언어로서, 미분방정식, 차분방정식 등을 해독할 수 있는 언어이다.(예 : CSMP,⁽⁴⁾ MS⁽⁵⁾ 등)

이 세가지 형태의 소프트웨어는 사용자의 목적이나 사용편의성, 사용자의 자동제어관련 지식수준 등에 따라 선택 되어 사용될 수 있으며 각각 일반적인 장·단점을 가지고 있다. (1)의 경우 라이브러리 루틴(library routine)을 이용함에 있어 사용되는 고급 언어의 기존 프로그램을 그대로 사용함으로써 프로그램의 다양성을 최대한 살릴수 있다는 장점이 있다. 그러나 이러한 방식의 단점은 사용자가 고급언어에 익숙해 있어야하며, 프로그램을 작성한 후 컴파일링(compiling), 링킹(linking)의 과정을 수행 하여야만 한다. (2)의 경우의 최대 장점은 사용자가 이용하기 매우 용이하다는 점이다. 즉 소프트웨어의 주어진 메뉴에 따라 대화식으로 자료를 입력, 출력 시킴으로써 원하는 결과를 쉽고 신속하게 출력할 수가 있다. 뿐만 아니라 특정결과를 얻기위한 해석방법에 대한 깊은 지식이 없이도 이러한 소프트웨어의 사용이 가능하며, 자료의 입력만 정당하다면 결과에 대한 일반적인 신뢰도도 매우 높은 편이다. 그러나 이러한 방식은 메뉴에 나타나지 않는 기능을 수행할 수 없다는 큰 단점이 있다. 그러므로 이러한 소프트웨어를 사용함에 있어, 주어진 기능(메뉴)을 충분히 파악하여 그 효용성을 극대화시켜야 한다. 또 다른 단점으로는 출력결과가 어떠한 과정을 거쳐 수행되었는지 사용자의 입장에서 알 수 없으므로 일반적인 결과

의 고신뢰도에도 불구하고, 경우에 따라 전혀 해독할 수 없는 결과나, 틀린 결과를 출력하는 경우가 있다. (3)의 경우는 (1),(2)방식을 복합한 형태로서 고급언어의 높은 programmability와 제어에 관련된 고유 기능을 결합시킨 새로운 형태의 프로그램 언어이다. 국내에서 널리 사용되고 있는 CSMP, GPSS가 그 예이다. 이러한 소프트웨어 사용상의 단점은 사용자가 그 언어에 익숙하여야 하며 원하는 결과를 얻기 위한 과정에 익숙하여야만 한다. 그러나 이러한 확장성 때문에 그 사용영역이 거의 무한정하다고 하겠다.

5. 프로그래밍 실례

필자가 보유하고있는 CSMP, PARASOL III, MS, CC에 대하여 간단한 예제를 보이하고자 한다.

예제 1: 그림 1과 같은 DC 서보 모터의 위치제어 시스템의 시간응답을 0부터 10초 까지 구하라. 단 적분증분은 0.1초, 결과는 매 0.5초 간격으로 출력하라.

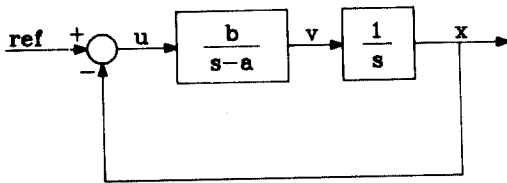


그림 1 DC서보 모터의 위치제어

• CSMP의 프로그램 코딩

표 1 CSMP 프로그램 예

```
INCON IC1=0.0, IC2=0.0, X=0.0, V=0.0
PARAMETER GAIN=1.0, A=-1.5, B=1.0

REF = STEP(0.0)
U = GAIN*(REF - X)
P = -1.0/A
XX = -B*U/A

V = REALPL (IC2,P,XX)
X = INTGRL (IC1,V)

TIMER FINTIN=10.0, DELT=0.1, PREDEL=0.5
PRINT X,V
END
STOP
```

• PARASOL⁽⁶⁾II의 프로그램 코딩

표 2 PARASOL II의 프로그래밍 예

```
"run.p" Sirtl ;Load start-up file for simulation.
Sdrsb sys ;Define simulation block.
u = ref - x - gain * SS ; u = gain*(ref - x)
#x = v SS ; x = v
#v = a v + b u + SS ; v = a*v + b*u
endsb ; End of simulation block.
-1.5 @a 1 @b 1 @gain 1 @ref ; a= 1.5, b=1, gain=1, ref=1
0 @x 0 @v Sic ; x=0, v=0 for initial conditions.
setpr ; Set printing parameters.
.t x v SS ; Print time, x, and v.
0.5 ; Set printing interval to 0.5.
```

• MS의 프로그램 코딩

표 3 MS의 프로그램 예

```
gain=1; a=-1.5; b=1
ref = 1

x=0; v=0

intgr1 time = 0,10,0.1 (
    u = gain*(ref - x )
    #x = v
    #v = a*v + b*u
    every 5 ( print time,x,v )
)
stop
```

• CC⁽⁶⁾의 입력자료 및 출력 결과

앞의 세경우가 일종의 전산언어인 반면 CC는 대화형 메뉴방식(interactive menu type)패키지이므로 메뉴에 따라 대화식으로 자료를 순

표 4 CC의 입력자료

```
CC>enter,gl
Enter each polynomial:a(n)*s^n + a(n-1)*s^(n-1) + ... + a(1)*s + a(0)
as follows:n, a(n), a(n-1), ..., a(1), a(0)

Enter # of polynomials in numerator > 1
Enter poly #1 > 0,1
Enter # of polynomials in denominator > 2
Enter poly #1 > 1,1,1,1,1
Enter poly #2 > 1,1,1,0

1
G(1) = -----
s ( s + 1.5)

CC>time,gl
Enter TIME command option >: 1
1=Closed loop step 2=Closed loop impulse
3=Open loop step 4=Open loop impulse
5=Open loop non-causal impulse

Automatic entry of remaining parameters ? (y/n) > y
```

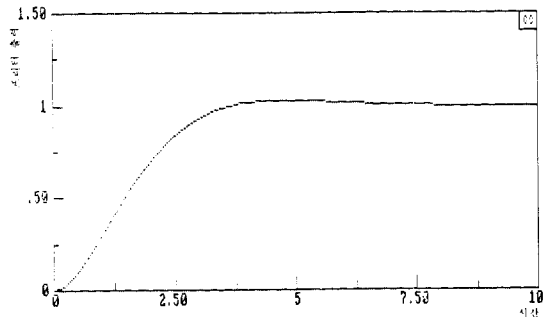


그림 2 CC결과의 프린터 출력

차적으로 입력 시켜야 한다. 표 4에서는 입력 방식을 나타내고 있으며 도표상의 굵은글자는 사용자가 입력시켜야 할 자료이다. CC의 계산 결과는 그래프로 출력되며 본 예제에 대한 결과는 그림 2에 보이는 바와 같다.

예제 2 : 그림 3과 같은 시스템의 최적 제어를 위한 상태변수 피드백 이득 K 를 구하라. 단,

$$A = \begin{bmatrix} 1 & 0 \\ 2 & -1 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

이고 코스트 기능(cost functional)은

$$J = \int_0^{\infty} (X^T Q X + U^T R U) dt$$

여기서 $Q = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$, $R = \text{임의의 양수}$

이다.

이 문제는 행렬 Ricatti방정식의 정상 상태를 구하는 문제로서 MS의 코딩으로 표 5에 나타내었다.

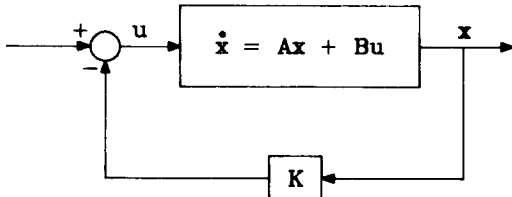


그림 3 2차 시스템의 상태변수 제환

표 5 Ricatti방정식의 MS코딩

```
matrix H[2,2],A[2,2],B[2,1],Q[2,2]
A=[0, 1, \Define matrix A. \
  2,-1] \equivalent to A=[0,1,2, 1] \
B=[0, \Define vector B. \
  1] \equivalent to b=[0,1] \
Q=[2,0, \Define matrix Q. \
  0,1]

print *,(ix,'input R')
input R \Read R, if negative then repeat. \
again: if R <= 0 (print *,(ix,'R must be positive number')) \
goto again

H=[0,0,0,0] \Set initial condition for H. \
intgr t=10,0,-0.1] \Backward integration \
##=-H*A-A'*H-Q+H*B/R*B'*H \Ricatti equation \
every 5 (print t,H) \Print the result \
] \every 5 iterations \

K=R~*B'*H
print *,(ix,'The optimum gain=',2f10.5) K
stop
```

5. 맺음말

컴퓨터는 그 자체만으로도 일반인의 흥미와 관심을 불러 일으키기에 충분한 기자제 이다. 이러한 컴퓨터를 자동제어 교육에 심분 활용함으로써 교육효과를 극대화 시킬 수 있을 것이다. 이를 위하여 교육에 대한 과감한 투자로서 컴퓨터등 기자재를 과감히 확보하고 이를 활용함과 아울러 이에 따른 적절한 교과내용의 재검토가 이루어져야 할 것이다. 본 글에서는 주로 컴퓨터 시뮬레이션 등 소프트웨어 활용에 대하여 주로 언급하였으나 이를 데이터수집 및 해석, 디지털 제어 등 하드웨어로서의 컴퓨터 활용도 더욱 교육에 도입되어야 할 것이다.

참 고 문 헌

- (1) M. Donath, Editor, 1984, "Dynamics : Newsletters of the Dynamic Systems and Control Division", Vol. 32, ASME.
- (2) M.G. Singh, Editor-in-Chief, 1987, "Systems and Control Encyclopedia", Vol. 6, pp. 4299 ~4418, Pergamon Press.
- (3) A.T. Clementson, 1976, "Extended Control and Simulation Language", University of Birmingham, Birmingham.
- (4) 1975, "Continuous System Modeling Program III(CSMP/III) Program Reference Manual", IBM.
- (5) J.H. Oh, 1988, "ms Tutorial," Internal Circulation, Dept. of Production Engineering, KAIST.
- (6) D.M. Auslander, 1986, "A Tutorial Introduction to Parasol II", Dept. of Mechanical Engineering, U. C., Berkeley.