# [Matlab 강의] Ch.2~3 제어 설계 소프트웨어를 이용한 시스템 시뮬레이션

[배경]

- 제어시스템의 설계 및 해석을 위해서는 <u>수학적 모델</u>이 필수적이다.
- 대부분의 제어 설계 s/w는 "**전달함수 형태로 주어진 시스템**"을 이용하여 시스템을 해석 및 설계 할 수 있다.

일반적으로 여러 입력과 초기 조건이 변하는 폐루프 궤환 제어시스템을 시뮬레이션할 때는 해석적인 해를 구하기 어렵다. 이때, MATLAB을 이용하여 계산상으로 해를 구하고 해를 그림으로 그려낼 수 있다.

[목표]

- 전달함수와 블록선도 (신호흐름선도) 논의
- 다항식 다루기
- 전달함수의 극, 영점 계산
- 폐루프 전달함수 계산
- 단위 계단응답

[다루는 함수] roots, poly, conv, polyval, tf, pzmap, pole, zero, series, parallel, feedback, minreal, step

1. 외부에서 힘을 가하지 않은 스프링-질량-감쇠기 시스템의 응답 $y(t)$

$$y(t) = \frac{y(0)}{\sqrt{1-\zeta^2}} e^{-\zeta w_n t} \sin(w_n \sqrt{1-\zeta^2}\, t + \theta) \quad (\zeta < 1;\ \text{감쇠부족})$$

where, $\omega_n = \sqrt{k/M}$, $\zeta = b/(2\sqrt{kM})$, $\theta = \cos^{-1}\zeta$, $y(0)$ : 초기변위

- $y(0) = 0.15m$, $\omega_n = \sqrt{2}\, rad/\mathrm{sec.}$, $\zeta = \frac{1}{2\sqrt{2}}$ $\left(\frac{k}{M} = 2,\ \ \frac{b}{M} = 1\right)$일 경우,

외부에서 힘을 가하지 않은 응답을 그리는 명령어

```
y0=0.15;
wn=sqrt(2);
zeta=1/(2*sqrt(2));
t=[0:0.1:10];
unforced
```

```
unforced.m
%Compute Unforced Response to an Initial Condition
c=(y0/sqrt(1-zeta^2));
y=c*exp(-zeta*wn*t).*sin(wn*sqrt(1-zeta^2)*t+ acos(zeta));
bu=c*exp(-zeta*wn*t);bl=-bu;
plot(t,y, t,bu,'--', t,bl,'--'), grid
xlabel('Time (sec)'), ylabel('y(t) (meters)')
legend(['₩omega_n=',num2str(wn),' ₩zeta=',num2str(zeta)])
```

2. 다항식 다루기

1) MATLAB에서는 다항식의 계수를 차수가 높은 순서대로 행벡터로 표현.

예) $p(s)=s^3+3s^2+4$

```
p=[1 3 0 4] %p(s)의 계수를 차수가 높은 순서대로 배열한 행벡터
r=roots(p)    %다항식 p의 근들이 배열된 열벡터
q=poly(r)     %다항식의 계수를 차수가 높은 순서대로 배열한 행벡터
```

2) 다항식의 곱셈; conv 함수

예) $n(s)=(3s^2+2s+1)(s+4)=3s^3+14s^2+9s+4$

```
p=[3 2 1];q=[1 4];
n=conv(p,q)             % 다항식 p와 q의 곱셈
value=polyval(n,-5)  % s=-5일 때, 다항식 n의 값
```

3. 전달함수

1) tf 함수

```
num=[...];  % 전달함수의 분자 다항식
den=[...];   % 전달함수의 분모 다항식
sys=tf(num,den)
```

2) 전달함수 객체의 발생

$$G_1(s)=\frac{10}{s^2+2s+5},\quad G_2=\frac{1}{s+1}$$

=> $G(s)=G_1(s)+G_2(s)=\dfrac{s^2+12s+15}{s^3+3s^2+7s+5}$

```
num1=[10]; den1=[1 2 5]; sys1=tf(num1,den1)
num2=[1]; den2=[1 1]; sys2=tf(num2,den2)
sys=sys1+sys2
```

3) pole, zero 함수

```
p=pole(sys) % sys가 전달함수일 때, pole을 계산
z=zero(sys) % zero를 계산
[P,Z]=pzmap(sys) % pole, zero의 위치를 구함.
pzmap(sys) % 복소평면 위에 pole, zero의 위치를 그려줌.
```

예) 

```
sys=tf([1 10],[1 2 1])
p=pole(sys)
z=zero(sys)
[P,Z]=pzmap(sys)
pzmap(sys)
```

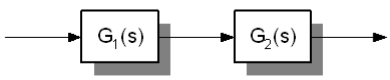
예제2.16) $G(s)=\dfrac{6s^2+1}{s^3+3s^2+3s+1}$, $H(s)=\dfrac{(s+1)(s+2)}{(s+2i)(s-2i)(s+3)}$

MATLAB 명령문을 이용하여 $G(s)$의 극점과 영점을 계산하고 $H(s)$의 특성방정식을 구하고, $G(s)$를 $H(s)$로 나눌 수 있다. 그리고 복소평면위에 $G(s)/H(s)$의 극-영점 선도를 그릴 수 있다.

```
numg=[6 0 1]; deng=[1 3 3 1]; sysg=tf(numg,deng)
[p,z]=pzmap(sysg)
n1=[1 1]; n2=[1 2]; numh=conv(n1,n2);
d1=[1 2*i]; d2=[1 -2*i]; d3=[1 3]; denh=conv(d1, conv(d2,d3));
sysh=tf(numh,denh)
sys=sysg/sysh
pzmap(sys)
```

4. 블록선도 모델 - 블록선도 간략화

1) 직렬연결



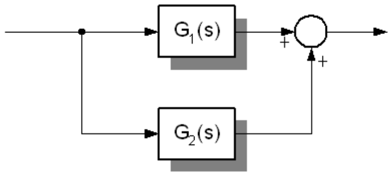

[sys]=series(sys1,sys2); % $T(s)=$sys, $G_1(s)=$sys1, $G_2(s)=$sys2

예1) $G_1(s)=\dfrac{s+1}{s+2}$, $G_2(s)=\dfrac{1}{500s^2}$,

=> $T(s)=G_1(s)G_2(s)=\dfrac{s+1}{500s^3+1000s^2}$

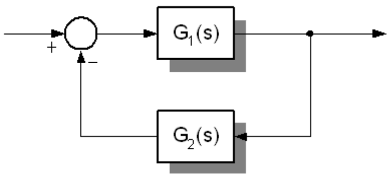
```
n1=[1 1];d1=[1 2];sys1=tf(n1,d1);
n2=1;d2=[500 0 0];sys2=tf(n2,d2);
sys=series(sys1,sys2)
```

2) 병렬연결




[sys]=parallel(sys1,sys2);
% $T(s)=$sys, $G_1(s)=$sys1, $G_2(s)=$sys2

3) feedback이 존재할 때




```
[sys]=feedback(sys1,sys2,sign); %T(s)=sys,  G1(s)=sys1,
                                %G2(s)=sys2, unit feedback => [1]
                                %sign +1 : positive feedback
                                %     -1 : negative feedback (default)
```

예2) 위의 예1)에 단위궤환(neg. feedback)이 있는 경우,

```
nc=[1 1];dc=[1 2];sys1=tf(nc,dc);
ng=1;dg=[500 0 0];sys2=tf(ng,dg);
sys3=series(sys1,sys2)
sys=feedback(sys3,[1])
```

cf.) series, parallel, feedback은 다중루프블록선도에서 블록선도를 조작하기 위해 사용됨.
- 그런데 이것이 복잡하다 하여서, Mason 공식을 배웠으므로 여기까지만...^^;

4) minreal 함수 : 전달함수의 공통 극, 영점 인수를 제거.

```
sys=minreal(sys1);
          % sys : no common factors, sys1 : possible common factors
```

5. 계단응답 : step 함수 - 선형시스템의 단위 계단응답을 계산. 그림을 그리는 것이 목적이라면, 좌변의 변수를 없앤다. $y(t)$가 필요한 경우, 아래와 같이 사용.

```
[y,T]=step(sys,t);
  % y : t에서 출력응답, T : simulation time
  % sys : 폐루프 전달함수, t : 사용자 시간벡터 or simulation final time
```

예) 전달함수가 다음과 같은 시스템에 대한 계단응답을 구하라.

$$T(s)=\frac{5400}{2s^2+2.5s+5402}$$

```
num=[5400];den=[2 2.5 5402];sys=tf(num,den);
t=[0:0.005:3];
[y,t]=step(sys,t);
plot(t,y),grid
xlabel('time(sec.)'),ylabel('y(t)')
```

3장 상태변수 모델

-시간영역 방법은 시스템 모델의 <u>상태공간 표현</u>을 이용한다. (여기서, 단일 입·출력시스템)

$$\dot{x} = Ax + Bu \qquad \text{식(3.1)}$$

$$y = Cx + Du$$

여기서 다룰 함수 - **ss 함수, lsim 함수, expm 함수**

1. ss 함수 - 전달함수 표현을 상태공간 표현으로 변환.

   cf.) tf 함수 - 상태공간표현을 전달함수 표현으로 변환.

```
sys_ss = ss(A,B,C,D)
                % sys_ss : 상태공간 모델,
                % A, B, C, D : 식(3.1)의 A, B, C, D
```

```
sys_ss = ss(sys_tf)  % sys_tf : 전달함수 모델
sys_tf = tf(sys_ss)
```

예) 다음의 3차 시스템을 고려하여, 상태공간 표현을 구하라.

$$T(s) = \frac{Y(s)}{R(s)} = \frac{2s^2+8s+6}{s^3+8s^2+16s+6}$$

```
n=[2 8 6]; d=[1 8 16 6]; sys_tf=tf(n,d);
sys_ss = ss(sys_tf)
tf(sys_ss)
```

※ 상태변수 표현은 유일하지 않으므로, 특정한 소프트웨어와 버전에 따라 달라질 수 있다.

2. expm 함수 - 행렬지수를 계산.

   cf.) exp(A) 함수는 행렬 A의 각요소 $a_{ij}$에 대해 $e^{a_{ij}}$를 계산.

3. lsim 함수 - 시스템의 시간응답을 구함.

예) 그림3.4의 RLC 회로에서, $A = \begin{pmatrix} 0 & -2 \\ 1 & -3 \end{pmatrix}$, $B = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$, $C = (1\ \ 0)$, $D = 0$이고, 초기조건은 $x_1(0) = x_2(0) = 1$이고, 입력은 $u(t) = 0$이다.

1) 상태천이행렬 : 주어진 시간 $\Delta t$(=dt)에서 상태천이행렬 계산

```
A=[0 -2; 1 -3]; dt=0.2; Phi=expm(A*dt)
```

2) 주어진 초기조건과 입력에 대한 시간응답

```
A=[0 -2; 1 -3]; B=[2;0]; C=[1 0]; D=[0]; sys=ss(A,B,C,D);
x0=[1 1]; t=[0:0.01:1]; u=0*t;
[y,T,x]=lsim(sys,u,t,x0);
subplot(121),plot(T,x(:,1)), subplot(122), plot(T,x(:,2))
```

※ lsim 함수의 결과와 상태천이행렬에 초기조건 상태벡터를 곱한 결과는 같다.