REPORT

TF 결과 보고서

전자 - 임베디드 전공

4조

주성민

목차

1. 개요

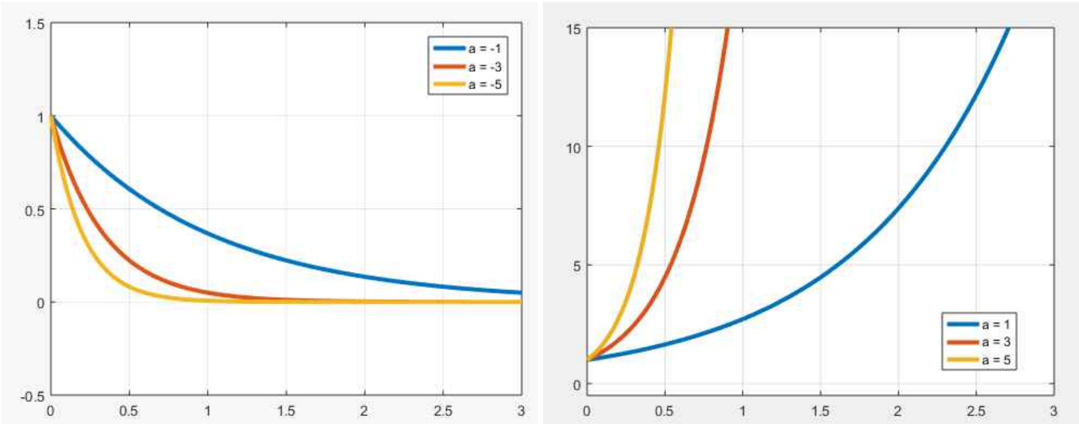
예비보고서를 통해 시스템 안정조건의 근거를 공부했었는데, 안정한 조건 이외에 극점과 영점이 시스템에 미치는 영향이 궁금하여 좀 더 상세하게 알아보고, 실습을 진행하였습니다.

2. 본문

- 극점이 시스템에 미치는 영향

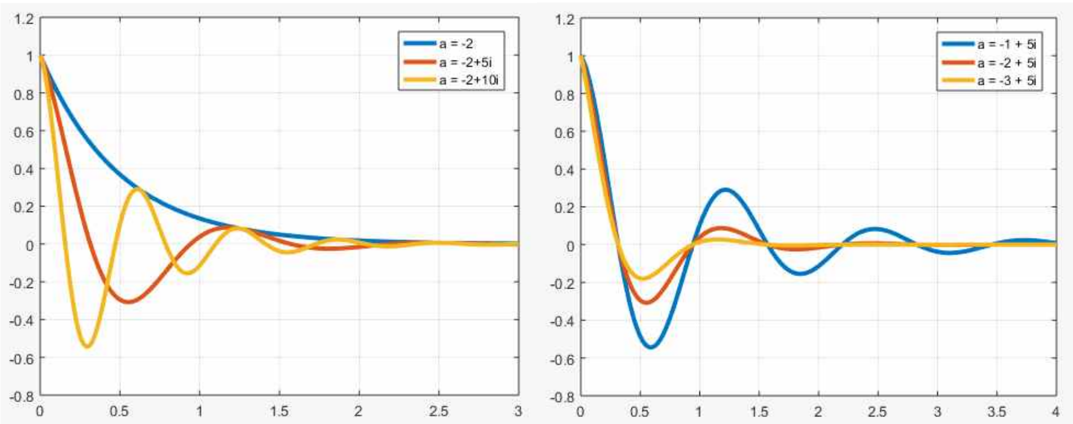
극점은 시스템의 특성에서 중요하다고 한다. 안정성을 판단할 때 영점의 위치와는 상관없이 극점의 위치를 고려했던 것을 보아도 알 수 있다. 연속 신호 시스템에서는 극점 a가 $a > 0$ 일 때 $\lim_{t\to\infty} h(t)$=0 이고, $a < 0$ 일 때 $\lim_{t\to\infty} h(t)=\infty$ 임은 이미 예비보고서에서 증명이 된 사실이다. 극점의 위치에 따라 h(t)가 어떻게 변화하는지 좀더 자세하게 알아보기위해 매트랩을 사용해서 구현해 보았다.

먼저 극점의 위치가 a<<0, a>>0 인 경우이다.



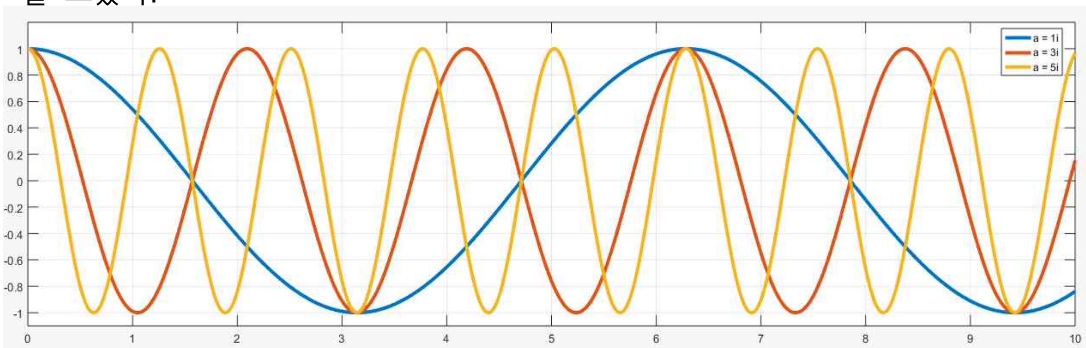
위 그래프는 실수축 a의 위치를 바꿔가면서 그려보았다. 극점 a가 0에서 좌측으로 멀어질수록 수렴 속도가 빨라지고, 우측으로 멀어질수록 발산 속도가 빨라지는 것을 볼 수 있다. 시스템이 빠르게 수렴한다는 것은 정상상태에 빠르게 도달하고, 빠르게 발산한다는 것은 많이 불안정하다고 볼 수 있다고 생각하여 0에서 좌측으로 멀어질수록 더 좋은 시스템이라는 결론을 내렸다.

다음은 극점의 허수 성분이 시스템에 미치는 영향을 확인해 보았다.



좌측은 동일한 실수성분에 허수성분을 변화시킨 결과이고, 우측은 동일한 허수 성분에 실수성분을 변화시킨 결과이다. 좌측 그래프에서 허수성분이 커질수록 진동이 심해지는 결과를 볼 수 있고, 우측 그래프에서 허수성분이 동일하다면 h(t)가 동일한 진동 주파수를 갖게 되는 결과를 볼 수 있다. 이러한 결과를 통해 극점의 허수성분은 시스템의 진동에 영향을 미친다는 결론을 내렸다.

마지막으로 강의자료에서 제한적 안정이라고 정의된 순 허수 극점일 때 결과를 보았다.



시간이 지나도 h(t)가 수렴하지 않고 1과 -1을 끝점으로 진동하는 결과를 볼 수 있다. 직전의 결과에서 허수성분이 커지면 진동이 심했는데 위 그래프에서도 허수성분의 크기가 커질수록 더 심하게 진동하는 결과를 볼 수 있다. 제한적 안정이라고 정의된 순 허수 극점이 있는 시스템은 $t \rightarrow \infty$ 일 때 발산하지는 않지만 충분한 시간이 지나도 안정된 출력이 나오지 않아서 안정이라고 볼 수 없는 것 같다.

- 영점이 시스템에 미치는 영향

예비보고서에서는 대부분 극점의 위치만 고려하고 영점의 영향을 제대로 확인하지 못했다. 영점은 시스템에 어떤 영향을 미치는지 확인하기 위해 다음과 같은 전달함수를 예시로 들었다.

$$H_a = \frac{2}{(s+1)(s+2)}\ , H_b = \frac{2(s+1.1)}{2.2(s+1)(s+2)}$$

두 번째 전달함수에 영점 2.2를 추가해 주고 분모에 -2.2를 추가로 나눠주었다. (두 전달함수를 비교하기 위해 DC성분을 맞춰주어야 한다고 한다.) 두 전달함수의 부분분수를 구해보면 다음과 같다.

$$H_a = \frac{2}{s+1} - \frac{2}{s+2}\ , H_b = \frac{0.18}{s+1} + \frac{1.64}{s+2}$$

좌측 전달함수에 s = -1.1 영점을 추가하니 s = -1 극점의 부분분수 계수가 줄어들었다. 역변환 시 e^{-t}항의 계수가 줄어들게 되고 결과적으로는 e^{-t}의 영향을 약화시키는 역할을 한다. e^{-t}항은 시간이 지나면 결국 0으로 수렴하는 항이기 때문에 $\lim_{t \rightarrow \infty} h(t)$=0, 즉 안정성에 형향을 주지는 않지만, t = 0부터 일정 시간 동안은(과도상태에서는) 영점을 추가하기 전과 다른 결과를 보여 줄 것이다. 이러한 결과로 영점은 과도상태에 영향을 줄 수 있는 성분이라는 결론을 내렸다.

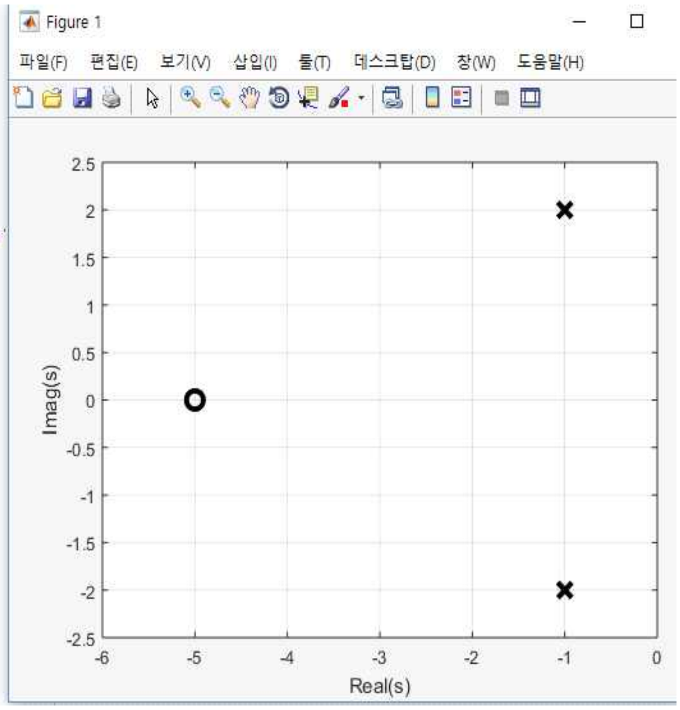
3. 실습결과

3.1 연속시스템의 전달함수

- 실습 다음 시스템의 pole-zero plot을 그리고 안정성을 판단하라. (roots 함수를 이용하라.)

$$H(s) = \frac{s+5}{s^2+2s+5}$$

```
clear;
clc;
t = linspace(0,15,1000);
temp1 = [1,5];
temp2 = [1, 2, 5];
zero = roots(temp1);
pole = roots(temp2);
figure(1)
plot(real(zero),imag(zero),'ko',real(pole),imag(pole),'kx','LineWidth'
axis([-6,0,-2.5,2.5])
grid on
xlabel('Real(s)')
ylabel('Imag(s)')
```



- 안정성 판단

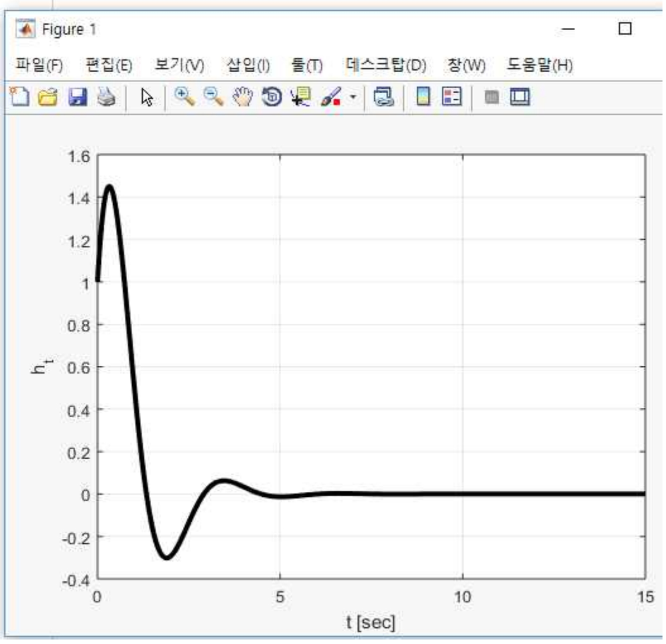
예비보고서에서 증명한 바와 같이 연속 신호 시스템에서 안정하기 위해서는 모든 극점이 0보다 작아야 합니다. (좌반평면에 있어야 함.) 현재 두 극점의 실수부분의 값은 -1 로 0보다 안정조건을 만족하고, 이 시스템은 안정하다고 말할 수 있습니다.

- 코드 설명

변수 temp1에 전달함수의 분자, temp2에 분모의 계수를 넣어 각각의 해를 roots 함수를 사용하여 구하면 zero 와 pole 이 됩니다. pole-zero plot 은 x축이 실수, y축이 허수이기 때문에 x축에는 real()을 사용하여 실수부를, y축에는 imag를 사용하여 허수부분을 정했습니다..

- **실습** 위 시스템 $H(s)$로부터 충격응답 $h(t)$를 그래프에 표시하고 시스템의 안정성을 판단하라. (residue 함수를 이용하라.)

```
clear;
clc;
t = linspace(0,15,1000);
[r,p,k] = residue([1,5],[1, 2, 5]);
r_re = repmat(r,1,length(t));
p_re = repmat(p,1,length(t));
t_re = repmat(t,length(p),1);
h_t = r_re.*exp(p_re.*t_re);
h_t = sum(h_t);

figure(1)
plot(t,h_t,'k','LineWidth', 3);
grid on;
xlabel('t [sec]')
ylabel('h_t')
```



- 안정성 판단

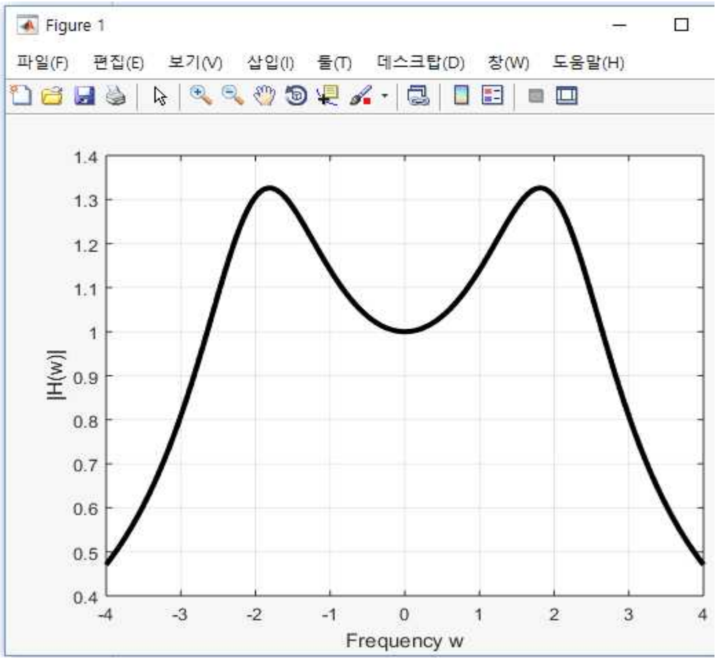
시스템이 안정하기 위해서 가장 기본적인 조건은 h(t) (충격응답)이 $t \longrightarrow \infty$ 일 때 0으로 수렴해야 합니다. 위 충격응답 그래프를 보면 5초 이후부터는 0에 거의 근접한 값이 나오고, 수식으로 풀어서 보면 $t \longrightarrow \infty$ 일 때 0으로 수렴한다고 할 수 있습니다. 따라서 이 연속신호 시스템은 안전하다고 볼 수 있습니다.

- 코드설명

먼저 residue 함수를 통해 pole 과 pole에 해당하는 계수를 구합니다. 입력 신호의 분모의 차수가 분자의 차수보다 높기 때문에 k에는 값이 들어가지 않게 됩니다. 변수 r, p는 각각 부분분수의 계수와 극점을 나타내며 두 변수의 길이는 동일 합니다. 충격응답을 구하기 위해 residue 함수로 구한 부분분수의 역변환을 구현해야 합니다. 변수 r, p는 위 시스템에서 2행 1열로 나오게 되는데, 2행 1열짜리 r, p를 2행 1000열짜리 변수로 만들어 새로운 변수 r_re, p_re에 저장합니다. 변수 t는 시간축을 나타내는 변수인데 벡터 곱을 해주기 위해 2행 1000열로 확장하여 t_re에 저장합니다. 변수 h_t에 라플라스 역변환 공식에 근거하여 변수를 대입해주면 2행 1000열이 나오게 되고 각 행은 residue를 통해 얻은 두 개의 p에관한 역변환이 됩니다. sum함수로 한번 묶어주게 되면 행이 합쳐지게 되고 각 t마다 두 개의 p에 관한 역변환을 합해준 값이 나옵니다. 최종적으로 위의 그래프와같은 결과가 나오게 됩니다.

- **실습** 위 시스템 $H(s)$로부터 주파수 응답 $H(\omega)$를 구하고 크기 $|H(\omega)|$를 그래프에 표시하라.
- **DEMO** 위 세 결과를 화면에 표시하라. (그림 14, 15, 16 참고)

```
clear;
clc;
w = linspace(-4,4,1000);
w_j = 1i*w;
temp1 = [1,5];
temp2 = [1, 2, 5];
zero = roots(temp1);
pole = roots(temp2);
H_w = (w_j-zero)./((w_j-pole(1)).*(w_j-pole(2)));
%크기를 구할 경우 abs사용
%H_w = abs(w_j-zero)./abs((w_j-pole(1)).*abs(w_j-pole(2)));
figure(1)
plot(w,abs(H_w),'k','LineWidth', 3)
grid on
xlabel('Frequency w')
ylabel('|H(w)|')
```



-코드설명

예시와 똑같은 그래프를 그려주기 위해 w는 -4~4구간을 설정해주었습니다. 먼저 pole과 zero를 구해준 뒤 전달함수 형태를 만들어줍니다. 주파수 응답을 구현하기 위해, 실제 jw를 변수 w_j로 저장한 뒤, s가 있는 자리에 변수 w_j를 넣어주었습니다. 강의자료에 있는 허수축 위의 한 점과 극점, 영점과의 거리를 통해 주파수 응답의 크기를 구하는 방법도 있지만, 문제에서 주파수 응답을 먼저 구하고 크기를 출력하라고 했기 때문에 주석으로만 남겼습니다. 위의 그래프와 같은 주파수 응답이 나오게 되며, |w| > 4인 구간에서는 점점 크기가 0으로 수렴할 것입니다.

3.2 연속시스템의 안정성

- 실습 DEMO 그림 17의 세 시스템에 대해 전달함수를 구하고, 충격응답을 그래프에 표시하라. (zp2tf 함수를 이용하라.) (그림 18 참고)
- 실습 위 세 시스템의 안정성을 판단하고 결과를 시간영역에서 확인하라.

```
clear;
clc;
t = linspace(0,15,1000);
zero = -5;
pole = [-1 + 2i,-1 - 2i ];
[num, den] = zp2tf(zero,pole,1);
[r(1,:),p(1,:),k] = residue(num,den);

zero = -5;
pole = [2i, -2i];
[num, den] = zp2tf(zero,pole,1);
[r(2,:),p(2,:),k] = residue(num,den);

zero = -5;
pole = [1 + 2i,1 - 2i];
[num, den] = zp2tf(zero,pole,1);
[r(3,:),p(3,:),k] = residue(num,den);

h_t = [];
figure(2)
for x = 1:3
    r_re = repmat(r(x,:)',1,length(t));
    p_re = repmat(p(x,:)',1,length(t));
    t_re = repmat(t,length(p(x,:)),1);
    h_t = r_re.*exp(p_re.*t_re);
    h_t = sum(h_t);
    subplot(3,1,x)
    plot(t,h_t,'k','LineWidth',3)
end
subplot(3,1,1)
axis([0,10,-0.5,1.5])
grid on;
ylabel('h_1');

subplot(3,1,2)
axis([0,10,-4,4])
grid on;
ylabel('h_2');

subplot(3,1,3)
axis([0,10,-50000,100000])
grid on;
xlabel('t[sec]');
ylabel('h_3');
```

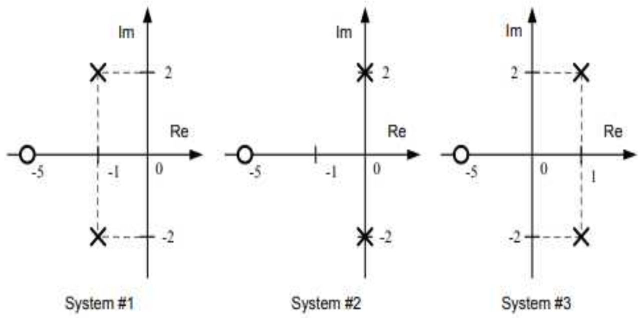
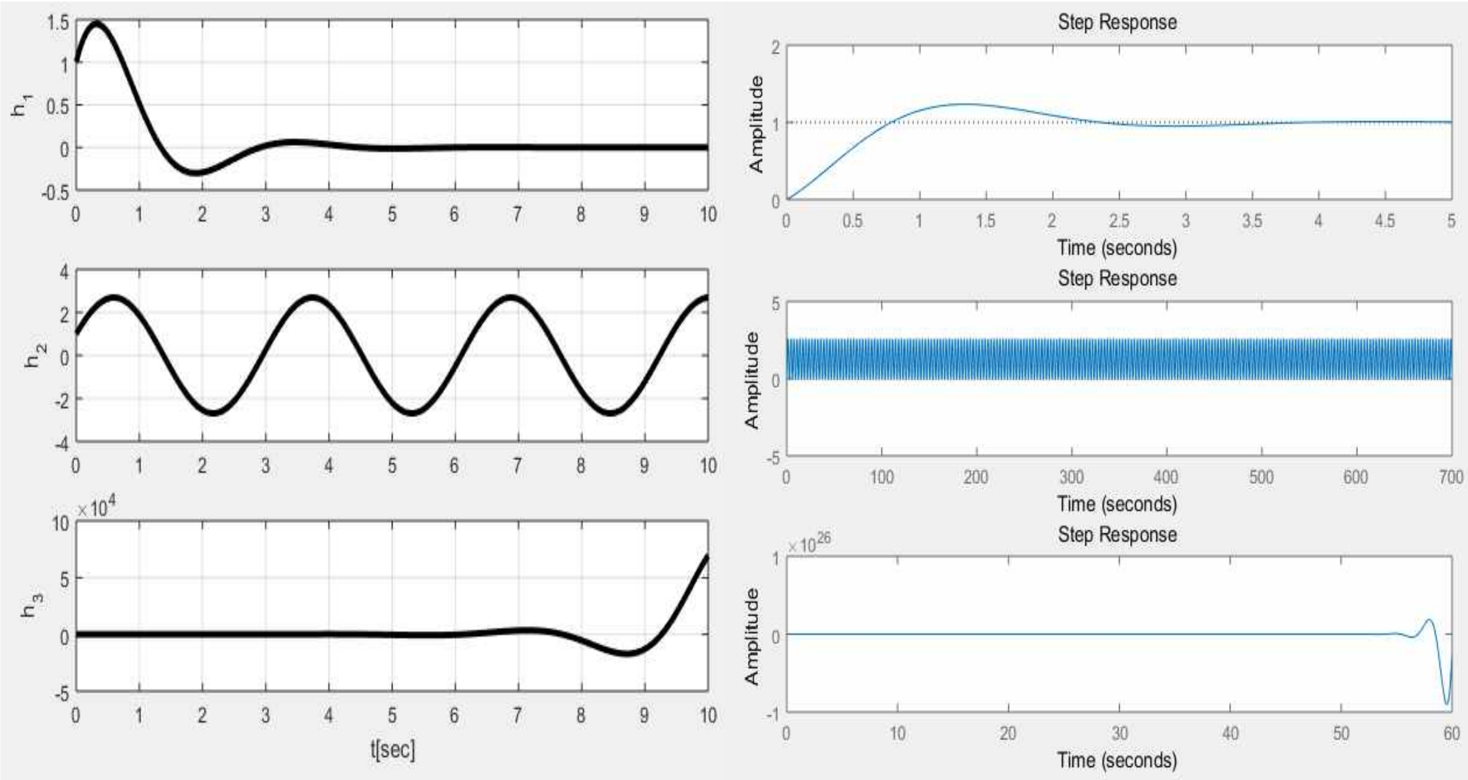


그림 17: 실습 3.2의 시스템.

-코드설명

그림 18에 있는 세가지 극점을 변수 pole, zero에 저장한 후 zp2tf함수를 이용전달함수를 구하게 됩니다. 전달함수를 구한 뒤 다시 residue함수를 이용하여 부분분수를 구하고 1번 문제에서 사용했던 알고리즘을 이용하여 각각의 충격응답을 구현하였습니다. 세 개의 pole, zero에 대해 한 번에 처리하기 위해 변수 r과 p의 차원을 확장 시켰습니다.

- 뒷장에 결과확인이 있습니다.



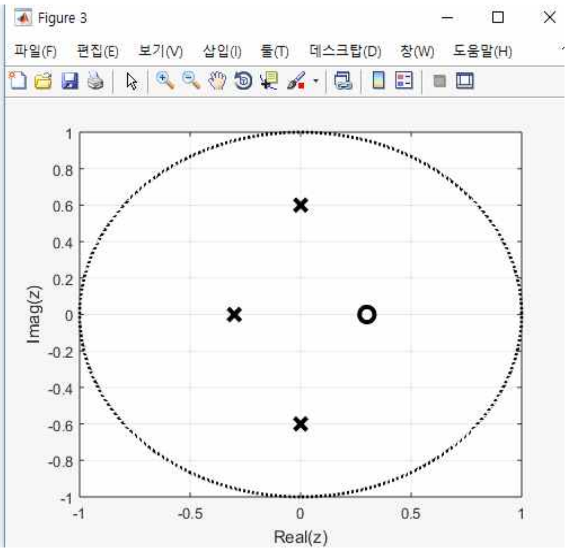
좌측 그래프는은 충격응답의 결과이고 우측 그래프는 해당 시스템의 시간축 응답에 대해 안정성을 확인해 보았습니다. 좌측 그래프에서 첫 번째 충격응답(h_1)은 시스템의 모든 극점이 좌반평면에 있기 때문에 $t \longrightarrow \infty$ 면 0으로 수렵하는 것을 볼 수 있습니다. 해당 시스템의 step응답 결과인 우측 첫 번째 그래프를 보면 안정성을 확인할 수 있습니다. 두 번째 충격응답(h_2)은 시스템의 극점이 허수축에 걸쳐있는데, 앞서 확인했던 결과처럼 실수성분이 없기 때문에 진동하게 되고 안정하지 않다고 그렇기 때문에 안정하지 않다고 할 수 있습니다. 해당 시스템의 step응답 결과인 우측 두 번째 그래프를 보면 700초가 지날 때 까지도 진동하고 있는 것을 볼 수 있습니다. 세 번째 충격응답(h_3)은 시스템의 극점이 우반평면에 있는데, $t \longrightarrow \infty$ 면 ∞로 발산하는 결과를 볼 수 있습니다. 우측 세 번째 그래프의 결과를 보면 50초 이후 step응답에 대해 값이 진동하기 시작하는데, 안정하지 못하다고 판단할 수 있습니다.

3.3 이산시스템의 전달함수

- **실습** 다음 시스템의 pole-zero plot을 그리고 안정성을 판단하라. 반지름 1인 원과 함께 표시하라. (roots 함수를 이용하라.)

$$H(z) = \frac{z - 0.3}{z^3 + 0.3z^2 + 0.36z + 0.108}$$

```
clear;
clc;
n = linspace(0,20,21);
num = [1, - 0.3];
den = [1, 0.3, 0.36,0.108];

zero = roots(num);
pole = roots(den);
theta = 0:0.01:2*pi;
x = cos(theta);
y = sin(theta);
figure(3)
plot(x,y,'k:','LineWidth', 2)
hold on
plot(real(zero),imag(zero),'ko',real(pole),imag(pole),'kx','LineWidth', 3,'MarkerSize',10)
axis([-1,1,-1,1])
grid on
xlabel('Real(z)')
ylabel('Imag(z)')
```



-안정성 판단

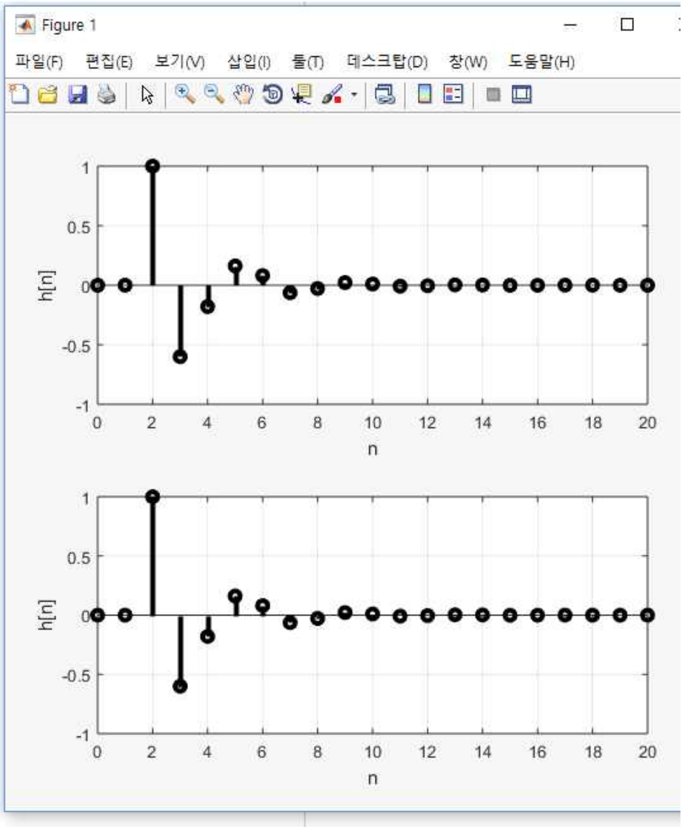
예비보고서에서 증명한 바와 같이 이산 신호 시스템에서 안정하기 위해서는 모든 극점의 크기가 1보다 작아야 합니다. (단위 원 안에 있어야 함.) 현재 세 극점 모두 단위원 기준으로 내부에 위치하여 안정조건을 만족하기 때문에 이 시스템은 안정하다고 말할 수 있습니다.

-코드설명

roots함수를 사용하여 문제의 이산 신호 시스템의 영점과 극점을 찾아낸 뒤 1번 문제에서 했던 방법과 동일하게 pole과 zero를 나타냈습니다. 이산 신호 시스템에서는 단위원을 기준으로 안정성을 판단하기 때문에 문제에서 그려주려고 한 것 같은데, 예전에 매트랩 과제에서 사용했던 cos과 sin을 이용하여 원을 그려주는 방법을 사용했습니다.

• **실습** 위 시스템 $H(z)$로부터 충격응답 $h[n]$을 그래프에 표시하고 시스템의 안정성을 판단하라. (residue 함수를 이용하라.)

```
clear;
clc;
n = linspace(0,20,21);
%%%%%%%%%%   (1)    %%%%%%%%%%
num = [1, - 0.3];
den = [1, 0.3, 0.36,0.108, 0];
[r,p,k] = residue(num,den);
r_re = repmat(r,1,length(n));
p_re = repmat(p,1,length(n));
n_re = repmat(n,length(p),1);
h_t = r_re.*(p_re).^(n_re);
h_t = sum(h_t);
%%%%%%%%%%   (2)    %%%%%%%%%%
num = [1, - 0.3];
den = [1, 0.3, 0.36,0.108];
[r,p,k] = residue(num,den);
Temp_A = r(1)*(p(1).^(n));
Temp_B = r(2)*(p(2).^(n));
Temp_C = r(3)*(p(3).^(n));
h_t_2 = Temp_A + Temp_B + Temp_C;
h_t_2 = [0,h_t_2(1:end-1)];

subplot(2,1,1)
stem(n,h_t,'k','LineWidth', 3);
grid on;
xlabel('n')
ylabel('h[n]')

subplot(2,1,2)
stem(n,h_t_2,'k','LineWidth', 3);
grid on;
xlabel('n')
ylabel('h[n]')
```



-안정성 판단

충격응답이 0으로 수렴하는 그래프를 볼 수 있는데, 이산 신호 시스템의 안정조건, 즉 $n \rightarrow \infty$ 일 때 h[n]이 0으로 수렴하는 조건을 만족시키기 때문에 안정하다고 볼 수 있습니다.

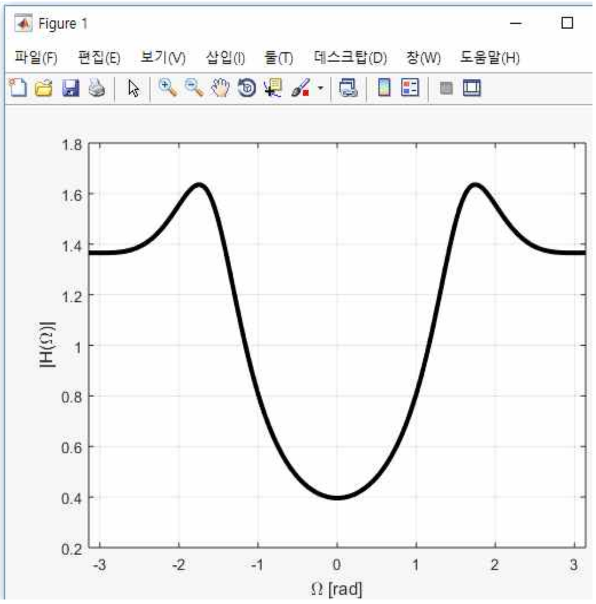
-코드설명

두 가지 방법을 사용해서 구현해 보았습니다. 첫 번째 방법은 z-역변환 꼴을 맞춰주기 위해 전달함수에 z를 미리 나눠주고 부분분수를 전개한 뒤 나중에 z를 곱해 주는 방법을 사용하였습니다. 두 번째 방법은 전달함수를 부분분수로 나눠준 뒤 전체 항에 $\frac{z}{z}$를 곱해줍니다. 분자의 z는 역변환에 사용되고 분모의 z는 시간지연의 특성을 가져오게 됩니다. 두 방법의 충격응답 수식을 손으로 구해보면 약간 다른 형태로 결과가 나오는데 매트랩으로 구현하여 결과 그래프를 확인해보니 동일한 결과가 나왔습니다. 시간 지연을 간단하게 구현하기 위해서 첫 번째 배열에 0을 넣어주고 마지막 배열을 제거하는 방법을 사용 하였습니다.(n < 0 => h[n] = 0)

- 실습 위 시스템 $H(z)$로부터 주파수 응답 $H(\Omega)$를 구하고 크기 $|H(\Omega)|$를 그래프에 표시하라.

- DEMO 위 세 결과를 화면에 표시하라. (그림 19, 20, 21 참고)

```
clear;
clc;
num = [1, - 0.3];
den = [1, 0.3, 0.36,0.108];
zero = roots(num);
pole = roots(den);
Om = linspace(-pi,pi,1000);
H_Om = (exp(Om*1j)-zero)./((exp(Om*1j)-pole(1)).*(exp(Om*1j)-pole(2)).*(exp(Om*1j)-pole(3)));
figure(1)
plot(Om,abs(H_Om),'k','LineWidth', 3)
axis([-pi, pi, 0.2,1.8]);
grid on
xlabel('\Omega [rad]')
ylabel('|H(\Omega)|')
```



- 코드설명

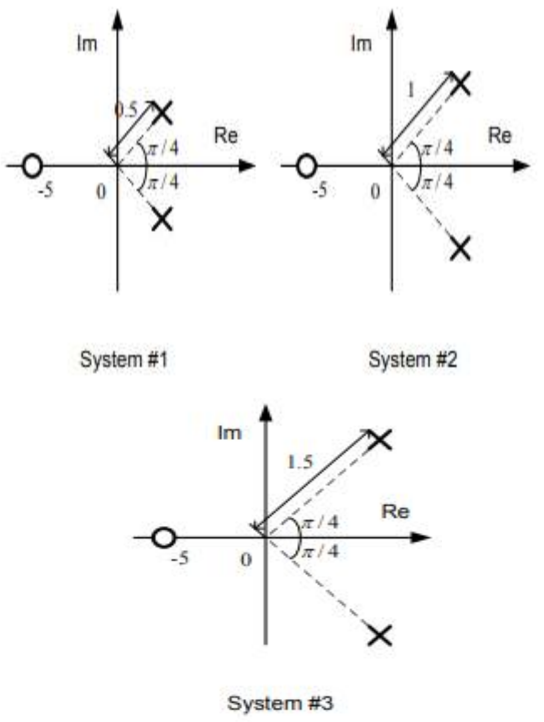
주어진 전달함수의 영점과 극점을 roots()함수를 사용하여 구한 뒤 z가 들어갈 자리에 단위 원 위의 한 점씩 넣어주면서 주파수 응답을 표현하였습니다. 범위는 $-\pi$~$+\pi$까지 봤는데 단위 원 위에 한 점은 2π를 기준으로 똑같은 위치로 돌아오기 때문에 주파수 응답은 2π를 주기로 반복 스펙트럼이 되고, 유효값만 보기 위해 한번의 주기만 표현했습니다.

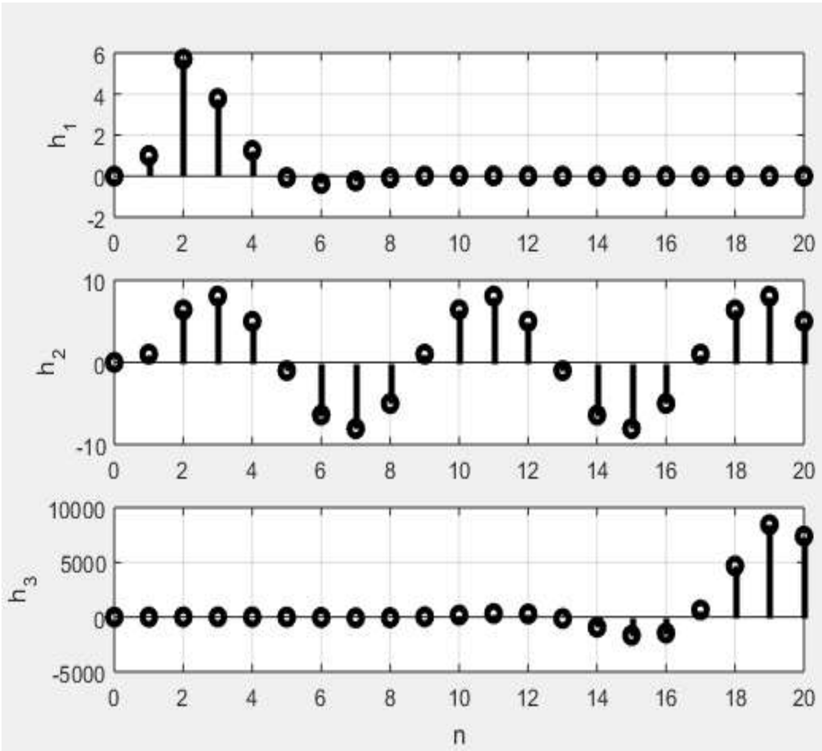
3.4 이산시스템의 안정성

- 실습 DEMO 그림 22의 세 시스템에 대해 전달함수를 구하고, 충격응답을 그래프에 표시하라. (zp2tf 함수를 이용하라.) (그림 23 참고)

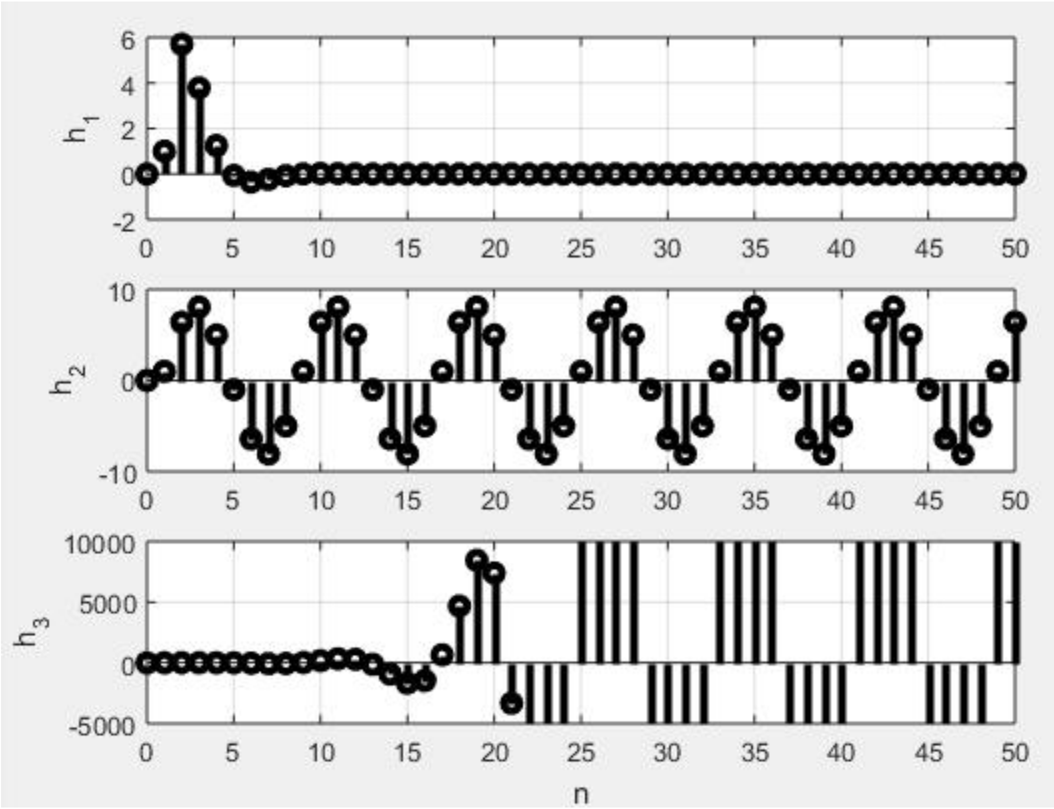
- 실습 위 세 시스템의 안정성을 판단하고 결과를 시간영역에서 확인하라.

```
clear;
clc;
n = linspace(0,20,21);

zero = -5;
pole = [0.5*exp(1j*pi/4),0.5*exp(-1j*pi/4) ];
[num, den] = zp2tf(zero,pole,1);
[r(1,:),p(1,:),k] = residue(num,[den, 0]);

zero = -5;
pole = [1*exp(1j*pi/4),1*exp(-1j*pi/4) ];
[num, den] = zp2tf(zero,pole,1);
[r(2,:),p(2,:),k] = residue(num,[den, 0]);

zero = -5;
pole = [1.5*exp(1j*pi/4),1.5*exp(-1j*pi/4) ];
[num, den] = zp2tf(zero,pole,1);
[r(3,:),p(3,:),k] = residue(num,[den, 0]);
h_n = [];
figure(4)
for i = 1:3
    r_re = repmat(r(i,:)',1,length(n));
    p_re = repmat(p(i,:)',1,length(n));
    n_re = repmat(n,length(p),1);
    temp_h_n = r_re.*(p_re).^(n_re);
    h_n = sum(temp_h_n);
    subplot(3,1,i)
    stem(n,h_n,'k','LineWidth', 3);
end
subplot(3,1,1)
axis([0,20,-2,6])
grid on;
ylabel('h_1');
subplot(3,1,2)
axis([0,20,-10,10])
grid on;
ylabel('h_2');
subplot(3,1,3)
axis([0,20,-5000,10000])
grid on;
xlabel('n');
ylabel('h_3');
```





그래프의 결과는 각 극점에 해당하는 충격응답 입니다. 이산 신호 시스템에서 step과 같은 함수를 어떻게 구현해야 할지 몰라서 이전의 3.2번과 같은 비교는 하지 못하였습니다. 그래프를 보면 첫 번째 시스템은 모든 극점이 단위 원 안에 존재하고 있습니다. 앞서 배운대로 pole-zero plot으로 판단한 결과는 안정한 시스템으로 예측되며, 충격응답 h_1의 결과가 시간이 지날수록 0으로 수렴하기 때문에 예측대로 안정한 시스템이라고 볼 수 있습니다. 나머지 두 시스템의 극점들은 단위 원 위에 있거나 밖에 존재하는데, pole-zero plot으로 판단한 결과는 불안정한 시스템으로 예측되었고, 충격응답 h_2는 진동하고 h_3도 오히려 시간이 지났을 때 값이 증가하는 결과를 볼 수 있습니다. $n \rightarrow \infty$ 을 직접 볼 순 없지만, h_1의 0으로 수렴하는 안정적인 결과와 비교해보니 h_2와 h_3 는 $n \rightarrow \infty$ 일 때 0으로 수렴하지 못할 것으로 예측했고, 불안정한 시스템이라고 판단했습니다.



n값을 늘려 관찰해본 결과 세 번째 충격응답 h_3는 엄청나게 큰 값으로 발산하는 것을 볼 수 있고, 불안정한 시스템이라는 것이 확인되었습니다. 또한 두 번째 충격응답 h_2도 똑같은 파형으로 계속 진동하기 때문에 불안정한 것을 확인했습니다.