2020-2학기 임베디드신호처리실습

결과보고서

Lab6. TF

2016146041 정성균
2014146012 박동훈

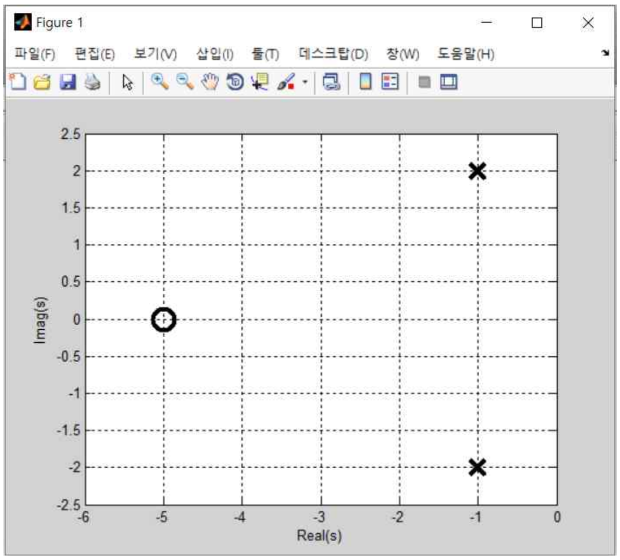
3.1 연속시스템의 전달함수

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

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

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

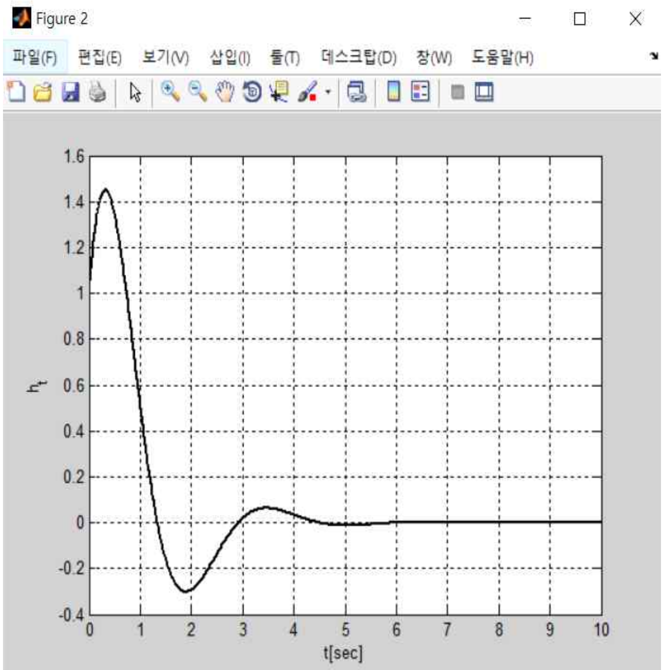
```
clear;
clc;
t= linspace(0,10,1000);

pole = [1 2 5];
zero = [1 5];

%==============3.1=============

poleroots = roots(pole);  %분모 다항식의 해
zeroroots = roots(zero);  %분자 다항식의 해

figure(1)
plot(real(poleroots),imag(poleroots),'xk',real(zeroroots),imag(zeroroots),'ok','MarkerSize',15,'LineWidth',3)
grid on;
axis([-6,0,-2.5,2.5])
xlabel('Real(s)');
ylabel('Imag(s)');
```



- 안정성 판단

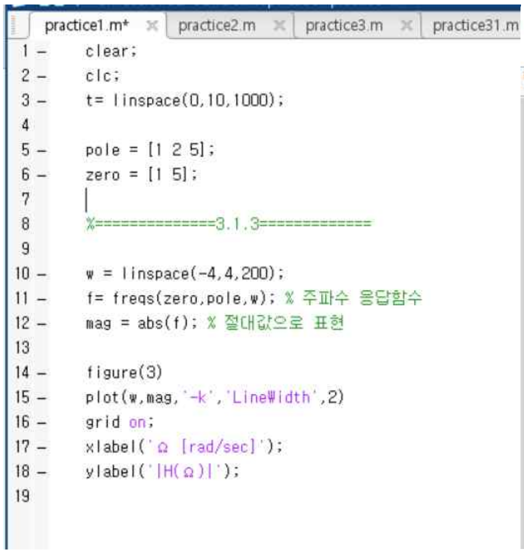
연속시스템에서 안정하기 위해서는 모든 pole이 0보다 작아야 한다.
즉 모든 pole이 좌평면에 위치해야 한다는 것이다.
현재 pole의 실수 값이 -1인 것을 보아 예비보고서에서 언급하였던 안정조건을 만족하고 있음을 알 수 있다.

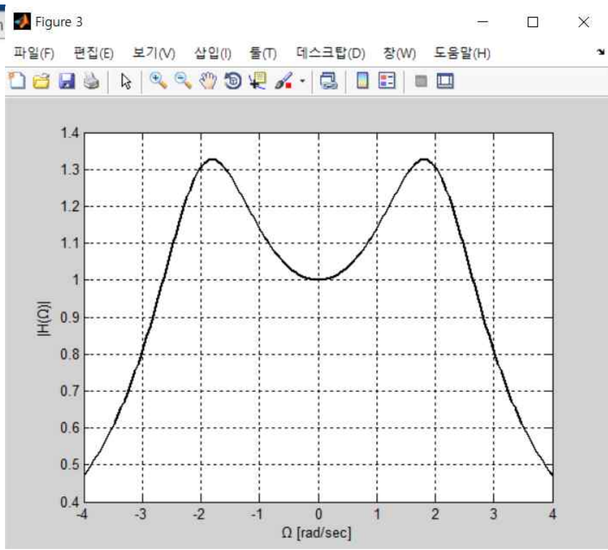
```matlab
clear;
clc;
t= linspace(0,10,1000);

pole = [1 2 5];
zero = [1 5];

%==============3.1.2=============
                               %r, p = 부분분수의 계수와 극점
[r,p,k] = residue(zero,pole) %pole 과 pole에 해당하는 계수
                               %입력 신호의 분모의 차수가 분자의 차수보다 높기 때문에 k에는 값이 안들어감
r_re = repmat(r,1,length(t));%충격응답을 구하기 위해 residue 함수로 구한 부분분수의 역변환을 구현
p_re = repmat(p,1,length(t));% 2행 1000열짜리 변수로 만들어 새로운 변수 r_re, p_re에 저장
t_re = repmat(t,length(p),1);%벡터 곱을 해주기 위해 2행 1000열로 확장하여 t_re에 저장
h_t = r_re.*exp(p_re.*t_re);%라플라스 역변환 공식
                              % 2행 1000열이 나오게 되고 각 행은 residue를 통해 얻은 두 개의 p에관한 역변환
h_t = sum(h_t);%sum함수로 한번 묶어주게 되면 행이 합쳐지게 되고 각 t마다 두 개의 p에 관한 역변환을 합해준 값


figure(2)
plot(t,h_t,'-k','LineWidth',2);
grid on;
xlabel('t[sec]');
ylabel('h_t');
```



- 안정성 판단

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





w의 구간을 정한 다음 pole과 zero에 계수를 넣어 주고 주파수 응답 함수(freqs)를 이용하여 주파수를 구한 다음, 절댓값으로 표현했다.
주파수 응답의 크기는 각 pole까지의 거리의 곱이 작을수록, 각 zero까지의 거리의 곱이 커질수록 커진다. 각 pole까지의 거리의 곱이 가장 작아지는 경우는 jw가 pole의 허수부와 같아질 때이다.
즉 이 그래프에서는 pole에서 w = -2, 2인 경우 거리가 가까워져 거리의 곱이 작아지고 주파수 응답이 커진다. 반대인 경우에는 주파수 응답이 작아진다. 결론적으로 나머지 영역에서는 응답이 작아지는 LPF의 특성을 갖는 주파수 응답을 얻을 수 있다.

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(0,10,1000);
zero = -5;
pole = [-1 + 2i, -1 - 2i];
[num, den] = zp2tf(zero,pole,1); %전달함수를 구한다.
[r(1,:),p(1,:),k] = residue(num,den);%residue함수를 이용하여 부분분수를 구한다.

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(1)
for x = 1:3 %충격응답 구하는 알고리즘
    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');
```

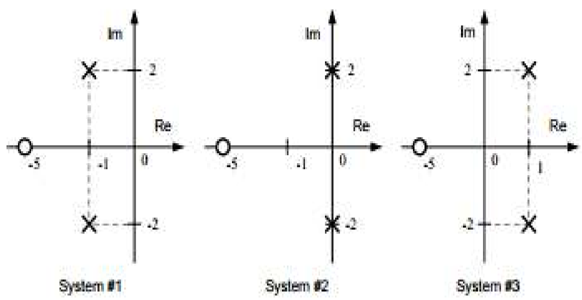
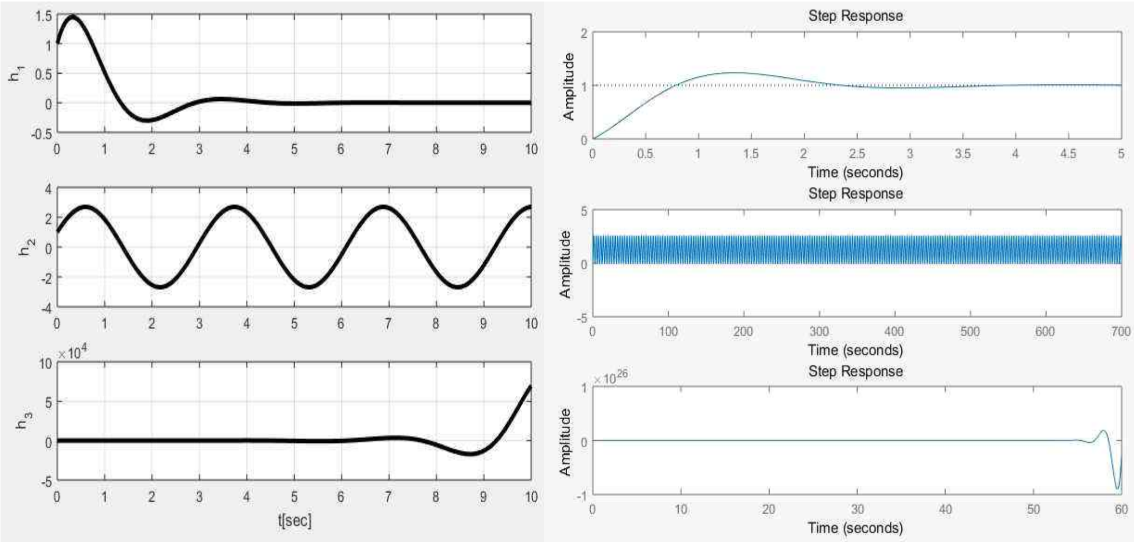


그림 17: 실습 3.2의 시스템.

- 코드설명

그림 17에 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.



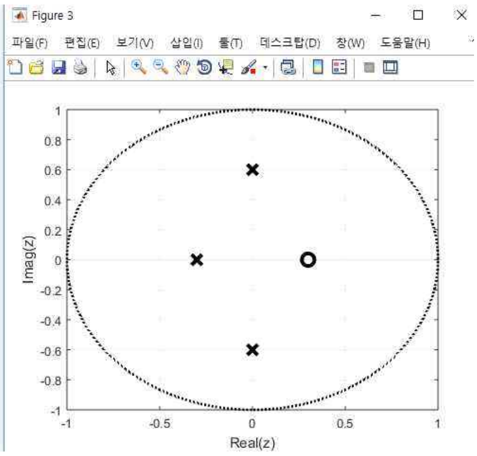
좌측 그래프는 충격응답의 결과이고 우측 그래프는 해당 시스템의 시간축 응답에 대해 안정성을 확인해 보았습니다. 좌측 그래프에서 첫 번째 충격응답(h_1)은 시스템의 모든 극점이 좌반평면에 있기 때문에 t→∞ 면 0으로 수렵하는 것을 볼 수 있습니다. 해당 시스템의 step 응답 결과인 우측 첫 번째 그래프를 보면 안정성을 확인할 수 있습니다. 두 번째 충격응답 (h_2)은 시스템의 극점이 허수축에 걸쳐있는데, 앞서 확인했던 결과처럼 실수성분이 없기 때문에 진동하게 되고 안정하지 않다고 그렇기 때문에 안정하지 않다고 할 수 있습니다. 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→∞ 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.)

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

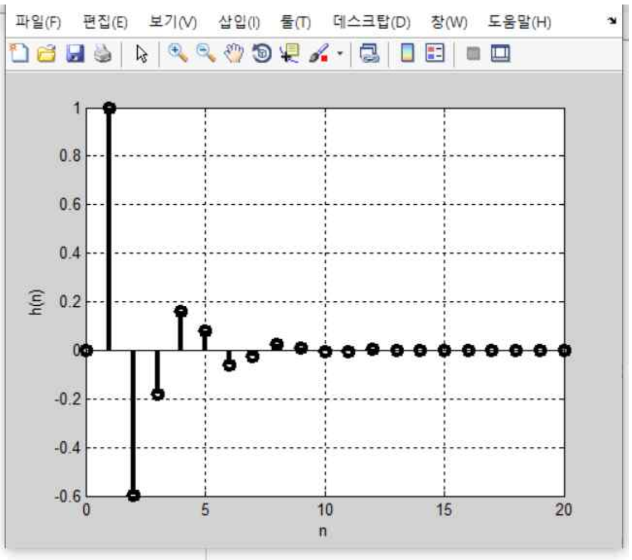
```matlab
clear;
clc;
n = linspace(0,20,21);
num = [1, - 0.3];
den = [1, 0.3, 0.36,0.108];

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)')
```



이산시스템에서 안정하기 위해서는 모든 pole이 1보다 작아야 한다. 즉 모든 pole이 단위원 안에 위치해야 한다는 것이다. 단위원안에 위치하는 영역을 수렴 영역(Region of convergence, ROC) 이라고 하는데, 여기서는 단위원이 ROC라고 할 수 있다. 현재 pole이 전부 단위원 안에 있는 것을 보아 예비보고서에서 언급하였던 안정조건을 만족하고 있음을 알 수 있다. pole과 zero에 계수를 넣고 roots 함수를 이용하여 각각 방정식의 해를 구했다. 실수부(zeroroots)를 O 표시로, y축에는 허수부(poleroots)를 X 표시로 나타냈다.

• 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);
pole = [1 0.3 0.36 0.108];
zero = [1 -0.3];
%========== 3.3.2 ==========
[r,p,k] = residue(zero,pole); %부분분수 전개
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);%z를 곱해준다.z-역변환
h_t = sum(h_t);

figure(2)
stem(n,h_t,'k','LineWidth',3);
grid on;
xlabel('n');
ylabel('h(n)');
```



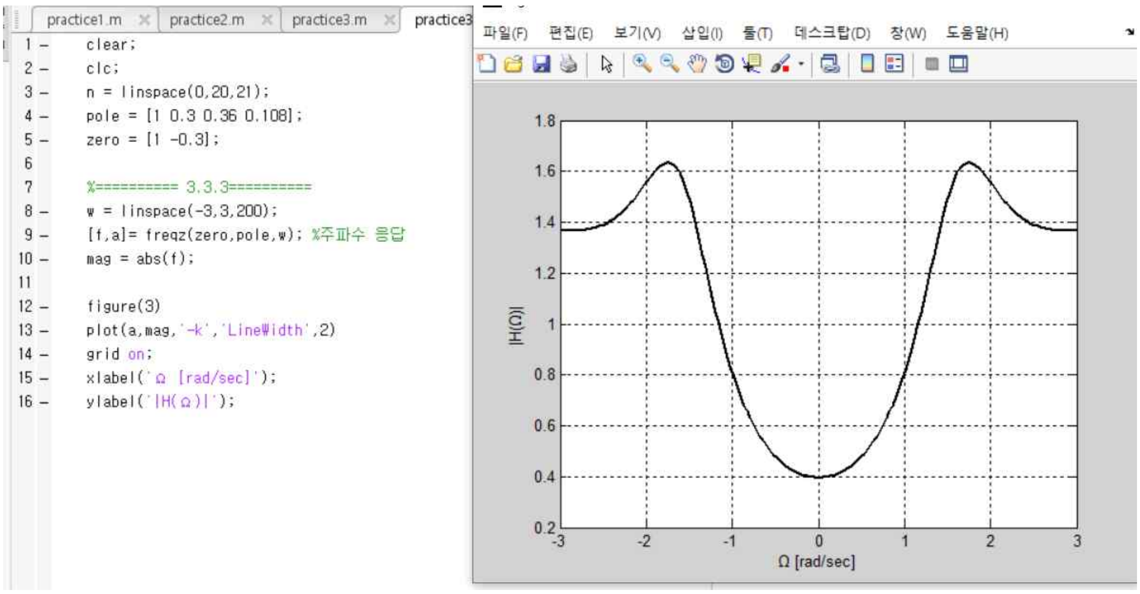
z-역변환 꼴을 맞춰주기 위해 전달함수에 z를 미리 나눠주고 부분분수를 전개한 뒤 나중에 z를 곱해 주는 방법을 사용하여 충격 응답을 구해주었다.

-안정성 판단

충격응답이 0으로 수렴하는 그래프를 볼 수 있는데, 이산 신호 시스템의 안정조건, 즉 n→∞일 때 h[n]이 0으로 수렴하는 조건을 만족시켜야 하는데 충격응답은 9초까지는 불안정한 상태를 보이나 그 이후로부터는 0에 거의 일치한 것을 확인할 수 있다. 즉 이 시스템은 안정하다고 볼 수 있다.

- **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)



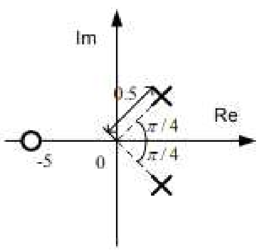
w의 구간을 정한 다음 pole과 zero에 계수를 넣어 주고 주파수 응답 함수(freqs)를 이용하여 주파수를 구한 다음, 절댓값으로 표현했다.
주파수 응답의 크기는 각 pole까지의 거리의 곱이 작을수록, 각 zero까지의 거리의 곱이 커질수록 커진다. 각 pole까지의 거리의 곱이 가장 작아지는 경우는 jw가 pole의 허수부와 같아질 때이다. 즉 이 그래프에서는 pole에서 w = -2, 2인 경우 거리가 가까워져 거리의 곱이 작아지고 주파수 응답이 커진다.
w = -2, 2인 지점을 약 1.7 rad로 볼 수 있는데, 각도로 계산하면 약 97.4˚ 이다. sin과 cos을 이용하여 단위원 상의 좌표를 구한 두 점 사이의 거리 공식을 사용하여 각각의 거리를 구할 수 있다.
w = 0일 때에는 zero와 거리가 가장 가까우므로 주파수 응답의 크기가 제일 작은 것을 알 수 있고, w = pi일 때에는 단위원 상의 좌표가 (-1,0) 이므로 거리를 계산하면 1.4에 가까운 주파수 응답의 크기를 확인할 수 있다.

3.4 이산시스템의 안정성

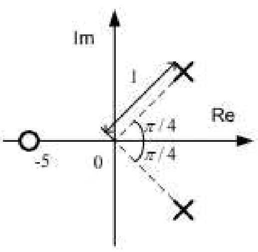
- 실습 DEMO 그림 22의 세 시스템에 대해 전달함수를 구하고, 충격응답을 그래프에 표시하라. (zp2tf 함수를 이용하라.) (그림 23 참고)

- 실습 위 세 시스템의 안정성을 판단하고 결과를 시간영역에서 확인하라.

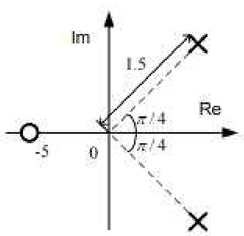
```matlab
clear;
clc;
n = linspace(0,20,21);

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

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

zero = -5;
pole = [1.5*exp(1j*pi/4),1.5*exp(-1j*pi/4) ];
[num,den] = zp2tf(zero,pole,1);%전달함수를 구한다.
[r(3,:),p(3,:),k] = residue(num,[den,0]);%residue함수를 이용하여 부분분수를 구한다.
h_n = [];
figure(1)

for i = 1:3 %충격응답 구하는 알고리즘
    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');
```



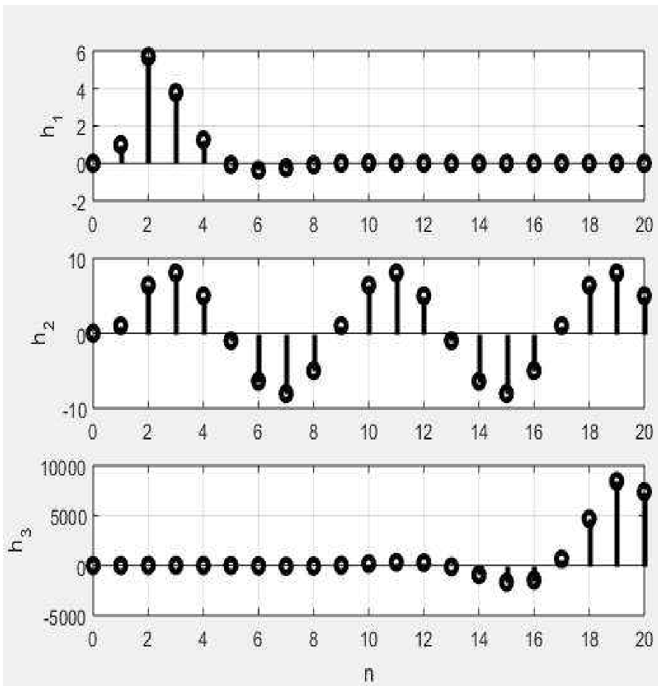
System #1



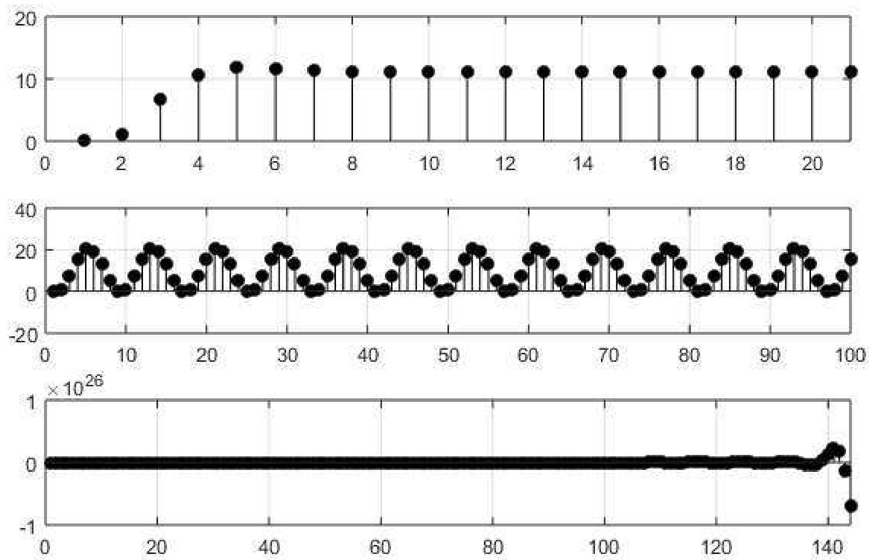
System #2



System #3



그래프의 결과는 각 극점에 해당하는 충격응답입니다.
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. $n \to \infty$을 직접 볼 순 없지만, h_1의 0으로 수렴하는 안정적인 결과와 비교해보니 h_2와 h_3 는 $n \to \infty$일 때 0으로 수렴하지 못할 것으로 예측했고, 불안정한 시스템이라고 판단했습니다.



위에서부터 순서대로 h_1, h_2, h_3의 시간 응답에 대하여 안정성을 확인한다.
첫 번째 시스템에서는 시간이 무한대로 갔을 때 충격응답이 수렴했었던 것처럼, 시간 응답에서도 곧장 수렴하여 안정하다고 판단할 수 있다.
두 번째 시스템에서는 일정하게 진동하는 것을 확인할 수 있는데, 조건적으로 안정함을 확인할 수 있다. 하지만 한 값에 수렴하지 않고 계속 일정한 수를 주기적으로 진동하는 것을 보면 안정하지 않다고 볼 수 있다.
세 번째 시스템에서는 두 번째 시스템과 달리 일정 시간이 지난 뒤 값이 진동하기 시작하는데 두 번째 시스템과 달리 더 크게 발산하는 모습을 보인다. 두 번째 시스템에서는 극점의 실수부가 0이기 때문에 응답이 감소하지도 증가하지도 않는 것을 보이는 것이고, 세 번째 시스템에서는 극점의 실수부가 양수이기 때문에 응답이 계속 증가하는 것이다. 결국 이 시스템은 불안정하다.