2017 DSP LAB01 convolution _ solution

```
function y=my_conv(x,h)

% h flip
fliph = fliplr(h);

% output 길이 지정.
outputLength = length(x)+length(fliph)-1;

% x data 에 zero padding 수행.
padX = [zeros(1,outputLength-length(x)), x];

% h data 에 zero padding 수행
padH = [fliph, zeros(1,outputLength-length(fliph))];

% output 변수 선언.
y = zeros(1,outputLength);
for i=1:outputLength % output 길이만큼 반복문 수행.

    % 배열의 element-wise multiplication 수행.
    y(1,i) = sum(padX.*padH);

    % 배열을 한칸씩 순환적으로 이동.
    padH = circshift(padH,1);
end
```

DSP LAB02 Solution

LAB 02

Text 3.22

$$y[n] = -\frac{1}{4}y[n-1] + \frac{1}{8}y[n-2] + x[n] + x[n-1]$$

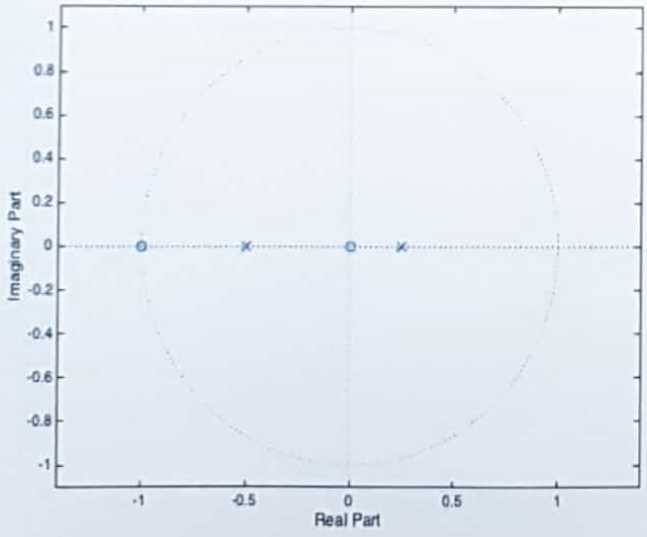
(a) Plot the pole-zeros pattern using the function zplane(a,b)

$$H(z) = \frac{1+z^{-1}}{1+\frac{1}{4}z^{-1}-\frac{1}{8}z^{-2}}$$

- Code :

```
>> n = 0:20;
>> b = [1, 1];
>> a = [1, 1/4, -1/8];
```

- Plot :



(b) Compute and plot the impulse response using the functions filter and stem. Compare with the plot obtained using the function impz

$$H(z) = \frac{\frac{-2}{3}}{1+\frac{1}{2}z^{-1}} + \frac{\frac{5}{3}}{1-\frac{1}{4}z^{-1}}$$

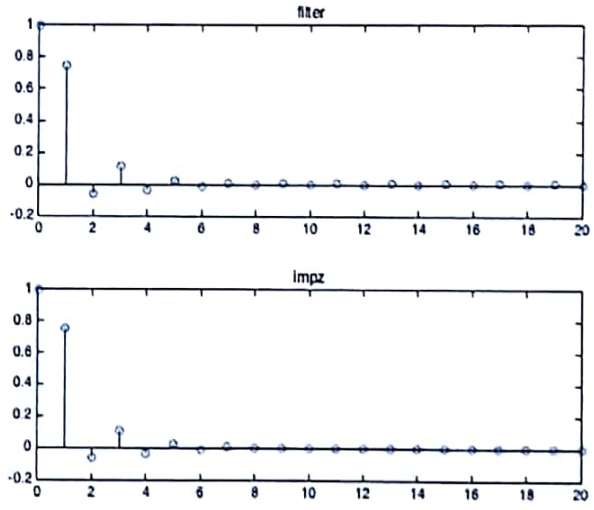
- Impulse response using "filter" :

```
hn1 = filter(b,a,[1,zeros(1,length(n)-1)]);
```

- Impulse response using "impz" :

```
hn2 = impz(b,a,length(n));
```

- Plot :



(c) Use the function residuez and the z-transform pairs in Table 3.1 to find and analytical expression for the impulse response h[n].

- **Analytical expression :**

$$h[n] = \frac{-2}{3}(\frac{-1}{2})^n u[n] + \frac{5}{3}(\frac{1}{4})^n u[n]$$

- **Use the function residuez :**

```
>> [r, p, k] = residuez(b,a)

r =

   -0.6667
    1.6667


p =

   -0.5000
    0.2500


k =

     []
```

(d) Compute the first ten samples of $h[n]$ using the formula obtained in Part (c) and compare with the values obtained from the difference equation.

- **Compute the first ten samples :**

```
>> n= 0:10;
>> hn = r(1)*p(1).^n + r(2)*p(2).^n

hn =

    1.0000    0.7500   -0.0625    0.1094   -0.0352    0.0225   -0.0100    0.0053   -0.0026    0.0013   -0.0006
```

- **Compare :**

```
>> hn = r(1)*p(1).^n + r(2)*p(2).^n

hn =

    1.0000    0.7500   -0.0625    0.1094   -0.0352    0.0225   -0.0100    0.0053   -0.0026    0.0013   -0.0006

>> hn_ref = filter(b,a,[1, zeros(1,length(n)-1)])

hn_ref =

    1.0000    0.7500   -0.0625    0.1094   -0.0352    0.0225   -0.0100    0.0053   -0.0026    0.0013   -0.0006
```

- **Plot :**