EE433 INTRODUCTİON TO DIGITAL SIGNAL PROCESSING

Computer Homework Part 2

Name: Korkut Emre Arslantürk **Number: 250206039**

Submission Date: 07/06/2021

Question 5

Starting from your solution in question 1 of Computer HW Part #1 and using the frequency transformations of low-pass filters described in ed. 3 of the textbook (pp.526-532, section 7.4), design a high-pass filter with cut-off frequency ωp = 0.7π:

(a) Give the system function, H(z), of the designed filter.

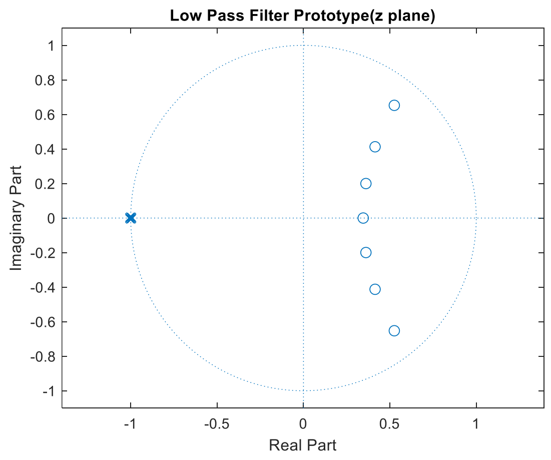
(b) Plot the magnitude of frequency response, indicating the corner frequencies corresponding to passband and stopband amplitude design targets given in (1).

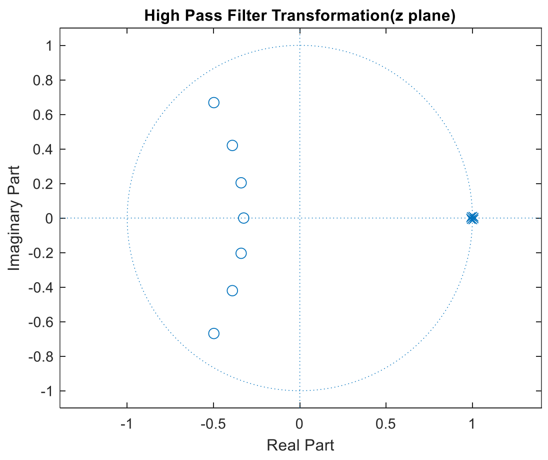
In this part, the code below generates low pass and high pass filter transfer functions. We use this formula writer instead of Z^{-1} while switch from low-pass filter to high-pass filter and we use this formula for alpha.

$$Z^{-1} = \frac{z^{-1} - \alpha}{1 - az^{-1}} \qquad \alpha = \frac{\sin\left(\frac{\theta_p - \omega_p}{2}\right)}{\sin\left(\frac{\theta_p + \omega_p}{2}\right)}$$

Where Wp equal to desired cutoff frequency.

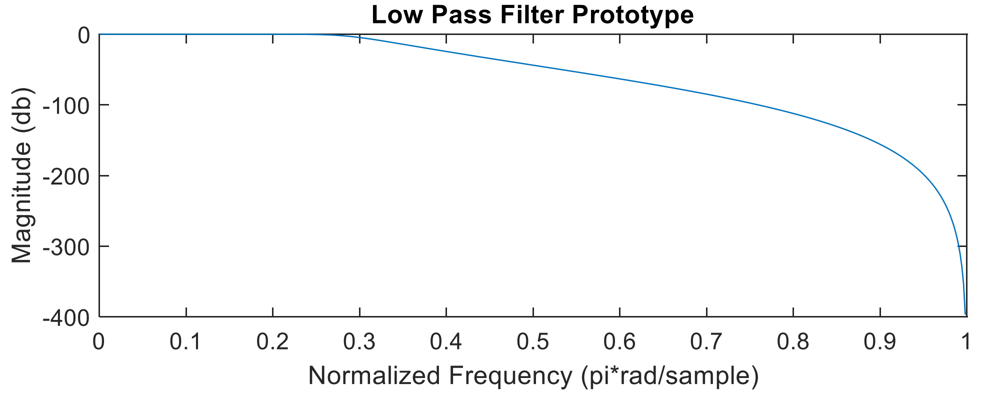
Simulation



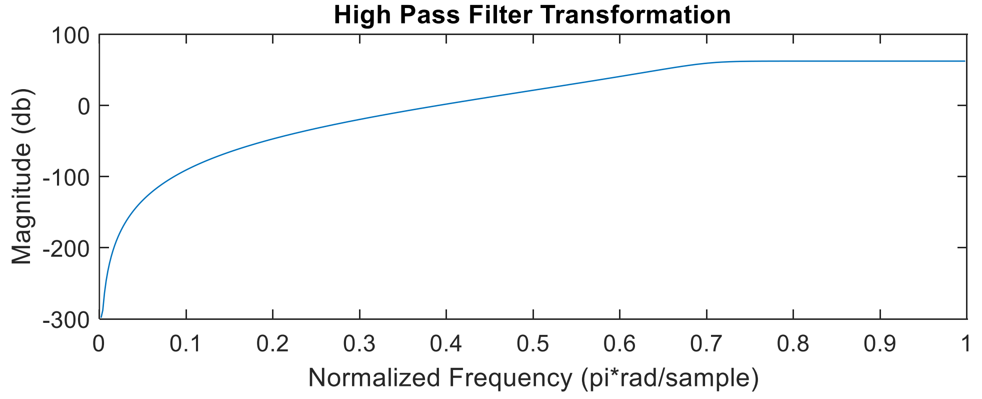


Poles:	Zeros:
-0.4979 + 0.6685i	1.0100 + 0.0049i
-0.4979 - 0.6685i	1.0100 - 0.0049i
-0.3907 + 0.4204i	1.0024 + 0.0109i
-0.3907 - 0.4204i	1.0024 - 0.0109i
-0.3399 + 0.2041i	0.9931 + 0.0086i
-0.3399 - 0.2041i	0.9931 - 0.0086i
-0.3252 + 0.0000i	0.9890 + 0.0000i

Low Pass Filter Prototype



High Pass Filter Transformation



Source Matlab Code

Design a high-pass filter

```
g_cut=0.905;
w_c=g_cut/pi;
order=7;
[y,x] = butter(order,w_c);
[z,p,k] = butter(order,w_c);
[h_1, w_1] = freqz(y,x);

w_d = 0.7*pi;
alpha = -cos((g_cut + w_d)/2) / cos((g_cut - w_d)/2);

syms Z(z)
Z(z)=-(z^-1 + alpha)/(1 + alpha*z^-1);
zero1 =1-(Z(z)*-1);
pole1 =1-(Z(z)*1.051) + (Z(z)*Z(z)*0.702);
pole2 =1-(Z(z)*0.829) + (Z(z)*Z(z)*0.342);
pole3 =1-(Z(z)*0.723) + (Z(z)*Z(z)*0.171);
pole4 =1-(Z(z)*0.346);

h = zero1^7 / (pole1*pole2*pole3*pole4);

[numerator_eq,denumerator_eq] = numden(h);
b_n = sym2poly(numerator_eq);
a_n = sym2poly(denumerator_eq);
[hhp, whp] = freqz(b_n, a_n);

figure
zplane(x,y);title('Low Pass Filter Prototype(z plane)')
figure
zplane(a_n,b_n);title('High Pass Filter Transformation(z plane)');

figure
subplot(211); plot(w_1/pi, 20*log10(abs(h_1)));
title('Low Pass Filter Prototype');
xlabel('Normalized Frequency (pi*rad/sample)'); ylabel('Magnitude (db)');
subplot(212); plot(whp/pi, 20*log10(abs(hhp)));
title('High Pass Filter Transformation');
xlabel('Normalized Frequency (pi*rad/sample)'); ylabel('Magnitude (db)');
```

Question 6

Starting from your solution in question 1 of Computer HW Part #1 and using the frequency transformations of low-pass filters described in ed. 3 of the textbook (pp.526-532, section 7.4), design a band-pass filter with cut-off frequencies ωp1 = 0.3π and ωp2 = 0.5π:

(a) Give the system function, H(z), of the designed filter.

(b) Plot the magnitude of frequency response, indicating the corner frequencies corresponding to passband and stopband amplitude design targets given in (1).

In this part, the code below generates low pass and high pass filter transfer functions. We use this formula writer instead of Z^{-1} while switch from low-pass filter to high-pass filter and we use this formula for alpha and k.

$$Z^{-1} = -\frac{z^{-2} - \frac{2\alpha k}{k+1} z^{-1} + \frac{k-1}{k+1}}{\frac{k-1}{k+1} z^{-2} - \frac{2\alpha k}{k+1} z^{-1} + 1} \qquad \alpha = \frac{\cos\left(\frac{\omega_{p2}+\omega_{p1}}{2}\right)}{\cos\left(\frac{\omega_{p2}-\omega_{p1}}{2}\right)}$$

$$k = \cot\left(\frac{\omega_{p2} - \omega_{p1}}{2}\right)\tan\left(\frac{\theta_p}{2}\right)$$ Where Wp1 equal to desired lower cutoff frequency and Wp2 desired upper cutoff frequency.

Poles	Zeros
-0.0442 + 0.9187i	-1.0042 + 0.0020i
-0.0442 - 0.9187i	-1.0042 + 0.0020i
0.5802 + 0.7350i	-1.0010 - 0.0046i
0.5802 - 0.7350i	-1.0010 - 0.0046i
0.0145 + 0.7835i	-0.9971 - 0.0037i
0.0145 - 0.7835i	-0.9971 - 0.0037i
0.4817 + 0.6640i	-0.9953 + 0.0000i
0.4817 - 0.6640i	1.0046 - 0.0022i
0.1100 + 0.6829i	1.0046 - 0.0022i
0.1100 - 0.6829i	1.0011 - 0.0050i
0.3664 + 0.6239i	1.0011 - 0.0050i

0.3664 - 0.6239i

0.2353 + 0.6267i

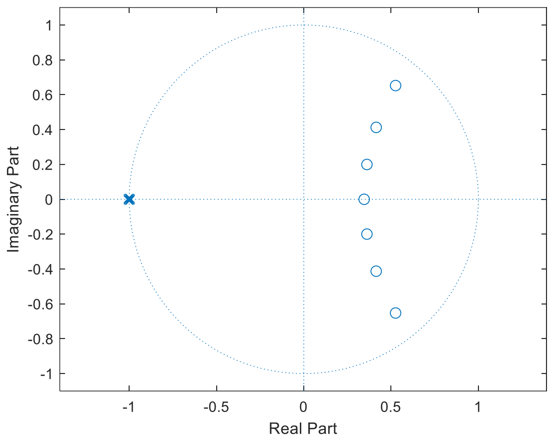
0.2353 - 0.6267i

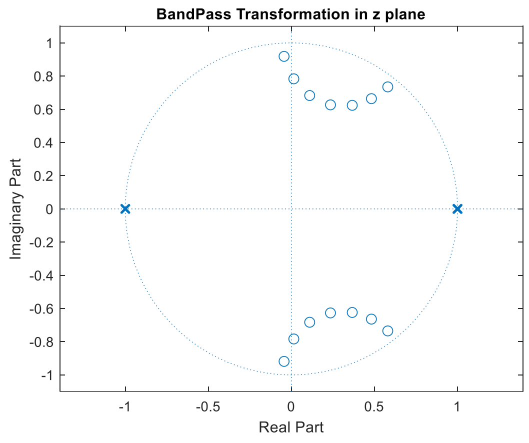
0.9968 + 0.0040i

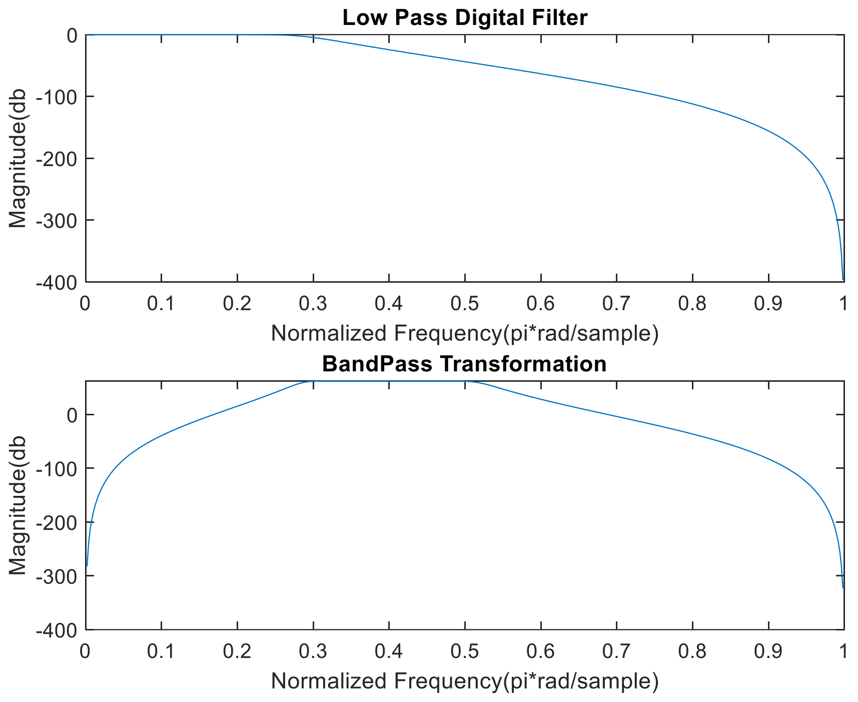
0.9968 - 0.0040i

0.9949 + 0.0000i

Simulation







Source Matlab Code

Design a band-pass filter

```
wp = 0.25*pi; ws = 0.40*pi;
gammacut=0.9047; w_c=gammacut/pi; order=7;
[y,x] = butter(order,w_c);
[z,p,k] = butter(order,w_c);
[h1,w1] = freqz(y,x);

wd_1=0.3*pi; wd_2=0.5*pi;

alpha=cos((wd_2 + wd_1)/2)/cos((wd_2 - wd_1)/2);
d_1=cot((wd_2-wd_1)/2)*tan(wp/2);

syms Z(z)
Z(z) = -(z^-2 - (2*alpha*d_1*z^-1)/(d_1+1) + (d_1-1)/(d_1+1))/ ((d_1-
1)/(d_1+1)*z^-2 - (2*alpha*d_1*z^-1)/(d_1+1) + 1);

% zeros and pole pairs
zero_1=1-(Z(z)*-1);
pole_1=1-(Z(z)*1.051) + (Z(z)*Z(z)*0.702);
pole_2=1-(Z(z)*0.829) + (Z(z)*Z(z)*0.342);
pole_3=1-(Z(z)*0.723) + (Z(z)*Z(z)*0.171);
pole_4=1-(Z(z)*0.346);
res=pole_1*pole_2*pole_3*pole_4;
h=zero_1^7/res;

[numeretor_equation, denumerator_equation] = numden(h);
y_new = sym2poly(numeretor_equation);x_new = sym2poly(denumerator_equation);
[hhp, whp]=freqz(y_new, x_new);

figure
zplane(x,y);
figure
zplane(x_new,y_new);title('BandPass Transformation in z plane');
figure
subplot(2,1,1); plot(w1/pi, 20*log10(abs(h1)));
title('Low Pass Digital Filter');
xlabel('Normalized Frequency(pi*rad/sample)'); ylabel('Magnitude(db');
subplot(2,1,2); plot(whp/pi, 20*log10(abs(hhp)));
title('BandPass Transformation');
xlabel('Normalized Frequency(pi*rad/sample)'); ylabel('Magnitude(db');
```

Question 7

Consider the DTFT shown below. That ideal filter will be approximated using a Kaiser window. The parameters of the design are A1/A2 ≥ 15 dB and A2/A3 ≥ 25 dB. Additionally, each transition band is not wider than 0.1π, that is the separation of successive critical frequencies obey the rule: |ωi − ωi+1| ≤ 0.1π

(a) Give the formula of h[n], the impulse response of the DT-filter.

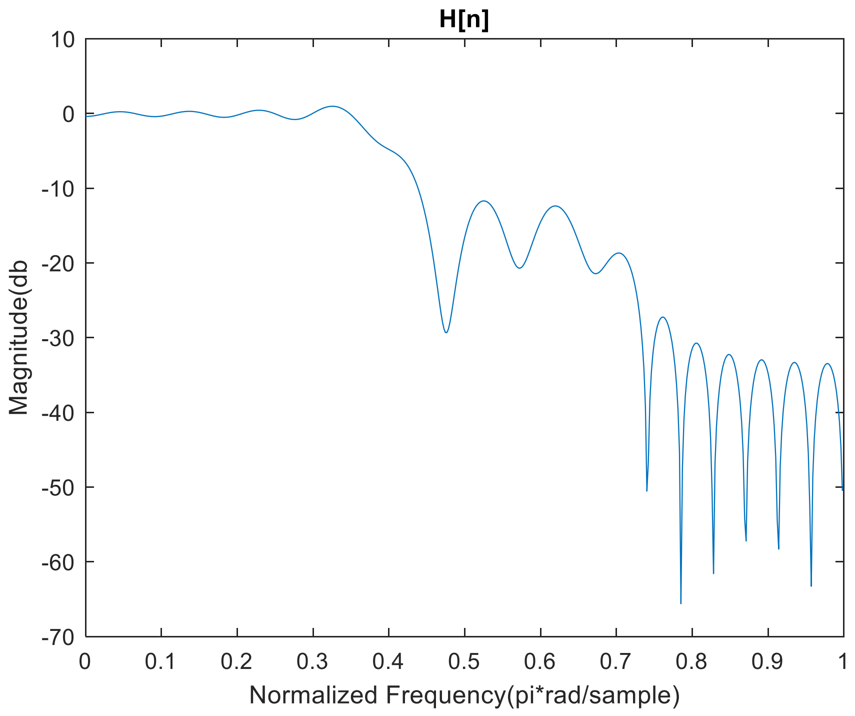
We have to multiply ideal low pass filter equation with Kaiser window.

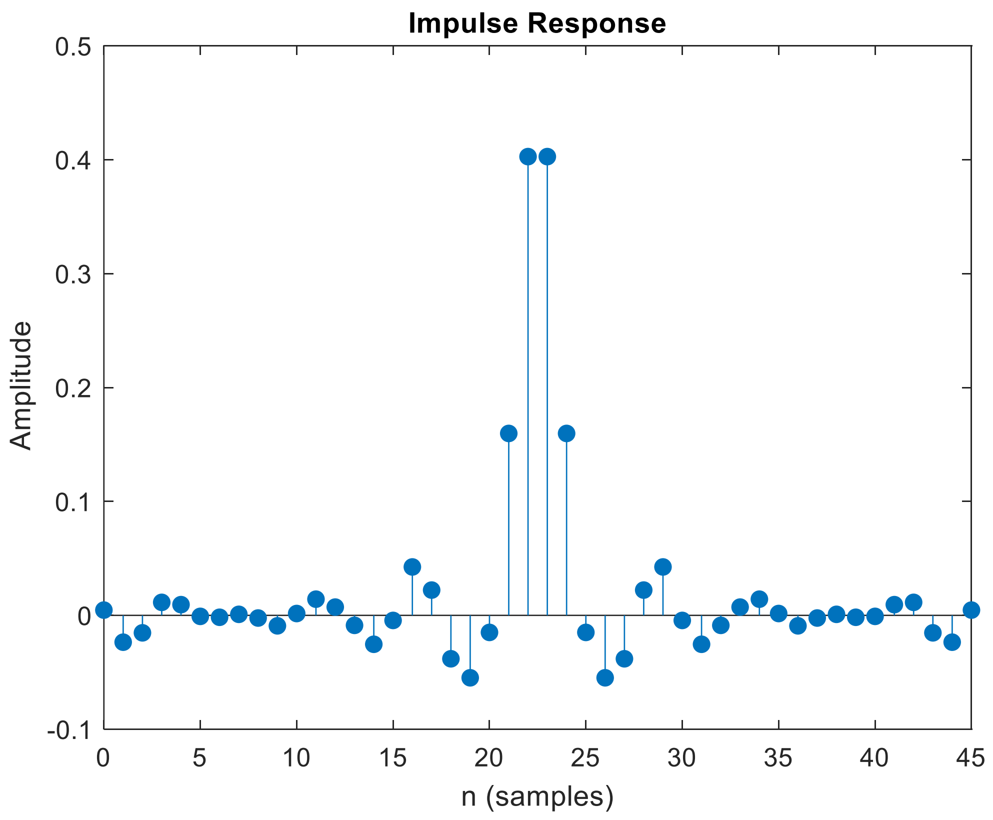
$$h[n] = \sum_{k=1}^{N_{mb}} = (A_k - A_{k+1}) \frac{\sin w_c (n-a)}{\pi (n-a)} \cdot \frac{I_0\left[\beta\left(1-\left[\frac{n-a}{a}\right]^2\right)^{1/2}\right]}{I_0(\beta)}$$

(b) Plot h[n].

(c) Plot the magnitude of the frequency response of the designed DT-filter. Check the gains at the critical frequencies.

Simulation





Source Matlab Code

```
parameter_1=1; parameter_2=0.178; parameter_3=(10^-2);
v_1=0.4*pi; v_2=0.7*pi;
theta=3.395; Nmb=2; order=45;
alpha=order/2;
G=[parameter_1 parameter_2];
w=[v_1 v_2];
h1=zeros(order+1,1);
i=1
while i<order+1;

    h1(i)=(G(1)-G(2))*(sin(w(1)*((i-1)-alpha))/(pi*((i-1)-
alpha)))*(besselj(0,theta*(1-((i-1-
alpha)/alpha)^2)^(1/2)))/besselj(0,theta);
    i=i+1;
end

h2=zeros(order,1);
j=1;
for j=1:order+1;

    h2(j)=(G(2)-parameter_3)*(sin(w(2)*((j-1)-alpha))/(pi*((j-1)-
alpha)))*(besselj(0,theta*(1-((j-1-
alpha)/alpha)^2)^(1/2)))/besselj(0,theta);
```

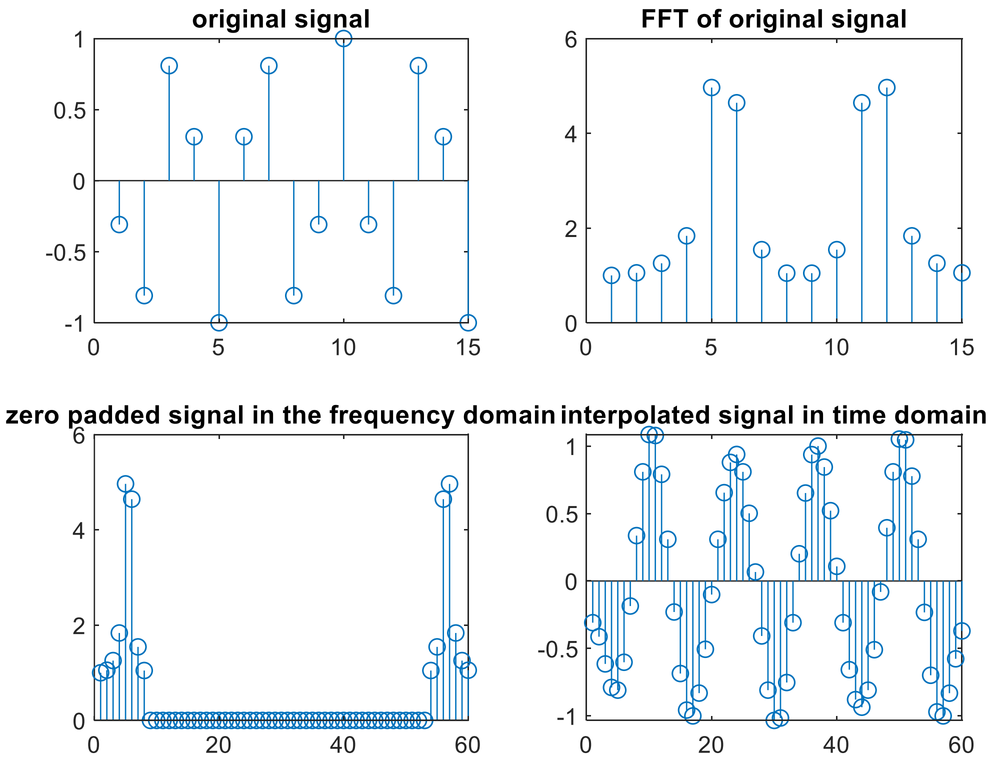
```
    j=j+1;
end

htotal=h1+h2;
[H W]=freqz(htotal);
figure
plot(W/pi,20*log10(abs(H)));
title('H[n]');
xlabel('Normalized Frequency(pi*rad/sample)'); ylabel('Magnitude(db)');

figure
impz(htotal)
```

Question 8

A signal is defined as x[n] = cos(2 ∗ π ∗ 0.3 ∗ n) for n = 1, 2, . . . , 15. Interpolate in time domain by a factor of 4 using zero padding in frequency domain. Plot all of the signals including every intermediate step in this procedure.



Source Matlab Code

```matlab
n=1:15;
y_t=cos(2*pi*0.3*n);
subplot(2,2,1);
stem(y_t);
title('original signal');
k=4;
pad_len = length(y_t)*(k - 1);
z_len = ceil((length(y_t)+1)/2);
z = (fft(y_t));
subplot(2,2,2);
stem(abs(z));
title('FFT of original signal');
z_p1= [z(1:z_len) zeros(1, pad_len) z(z_len+1:end)];
if ~mod(length(y_t),2)
z_p1(z_len) = z_p1(z_len)/2; z_p1(z_len+pad_len) = z_p1(z_len);
end
subplot(2,2,3);
stem(abs(z_p1));
title('zero padded signal in the frequency domain');
yp = real(ifft(z_p1))*k;
subplot(2,2,4);
stem(yp);
title('interpolated signal in time domain');
```