Oğuz Can Duran 21803175 Lab-6

$M_1 = 1$ $\Rightarrow$ $M_1 = 3$ $\Rightarrow$ Cut-off frequencies: $\frac{\pi}{3}$, $\frac{\pi}{5}$
$M_2 = 3$ $\quad\;\;$ $M_2 = 5$

$\hookrightarrow$ Order of the filter = 6

The filter must be causel and steble. In ordr to make the system "cousel" the poles must be right sided. Also to make the system steble, unit circle must be insid the ROC. By considering these specificatiers poles and zeros were chosen.

1) The impulse response h[n] of a causel system can be found by applying an impulse to system:

$$h[n] = -\sum_{k=1}^{6} a_k\, h[n-k] + \sum_{l=0}^{6} b_l\, \delta[l]$$

this formula comes from (A) which is shown below.
$y[n] \to h[n]$
$x[n] \to \delta[n]$

noting that $h[n] = 0$ for $n < 0$ since the system is cousel.

Since the order of the filter is 6, I have 6 poles and to have a proper filter 6 zeros.

Poles: $\{ e^{\pm j \frac{40\pi}{180}}, e^{\pm j \frac{50\pi}{18}}, e^{\pm j \frac{60\pi}{180}} \}$

Zeros: $\{ 0{,}9 \cdot e^{\pm j \frac{\pi}{59}}, 0{,}9 \cdot e^{\pm j \frac{2\pi}{3}}, 0{,}9 \cdot e^{\pm j \frac{4\pi}{5}} \}$

$\Rightarrow$

$$h[n] \xrightarrow{z} H(z) = \frac{Y(z)}{X(z)} = \frac{(z-e^{j\frac{4\pi}{18}})(z-e^{-j\frac{4\pi}{18}})(z-e^{j\frac{5\pi}{18}})(z-e^{-j\frac{5\pi}{18}})(z-e^{j\frac{6\pi}{18}})(z\; e^{-j\frac{6\pi}{18}})}{(z-0{,}9e^{j\frac{\pi}{59}})(z-0{,}9e^{-j\frac{\pi}{59}})(z-0{,}9e^{j\frac{2\pi}{3}})(z-0{,}9e^{-j\frac{2\pi}{3}})(z-0{,}9e^{j\frac{4\pi}{5}})(z-0{,}9e^{-j\frac{4\pi}{5}})}$$

$$\Rightarrow H(z) = \frac{z^6 + 0{,}619z^5 - 0{,}6154z^4 - 1{,}9964z^3 - 0{,}6154z^2 + 0{,}619z + 1}{z^6 - 3{,}436z^5 + 6{,}31z^4 - 7{,}002z^3 + 5{,}11z^2 - 2{,}254z + 0{,}531} \quad (A)$$

$\Rightarrow$ if $x[n]$ is input to the system, and $y[n]$ is output, by inverse z transform:

$$y[n] - 3{,}436\, y[n-1] + 6{,}31 \cdot y[n-2] - 7{,}002 \cdot y[n-3] + 5{,}11\, y[n-4] - 2{,}254\, y[n-5] + 0{,}531\, y[n-6]$$
$$= x[n] + 0{,}619\, x[n-1] - 0{,}6154[n-2] - 1{,}9964\, x[n-3] - 0{,}6154\, x[n-4] + 0{,}619\, x[n-5] + x[n-6]$$

Corresponding difference equation to the system

2) $x_a(t) = \cos(\alpha t^2)$

$T_1 = \sqrt{\frac{\pi}{a\,512}} \Rightarrow x_1[n] = \cos\left(\frac{\pi t^2}{512}\right)$

$T_2 = \sqrt{\frac{\pi}{a \cdot 8192}} \Rightarrow x_2[n] = \cos\left(\frac{\pi t^2}{8192}\right)$

3)
$$y[n] = -\sum_{k=1}^{K} a_k \cdot y[n-k] + \sum_{\ell=0}^{L} b_\ell \cdot x[n-\ell]$$

where a_i's comes from zeros, and b_i's comes from poles.
All a_i's and b_i's can be seen in (*).

- The output of the system to $x_1[n]$ and $x_2[n]$ are calculated using the equation above: $x_1 \rightarrow \boxed{T} \rightarrow y_1$, $x_2 \rightarrow \boxed{T} \rightarrow y_2$
- If the given input to the system is an impulse function then the output of the system will be impulse response of the system. In Question 1 $h[n]$ is calculated using this logic. No need to worry about past data of the $h[n]$ since $h[n] = 0$ for $n < 0$ since the system is causal.
- The chirpsignal has increasing frequency so when x_1 and x_2 are applied to the filter, this behavior of chirp signal can be observed. Therefore, we can observe in which intervals the frequencies can pass through the filter. However, since we have finite amount of memory and time we could test the filter for only limited interval.

4) By sampling the signal we get a periodic signal which is proven in previous lab. Therefore $x_r(t)$ is periodic, while $x_a(t)$ is not. Therefore these functions are not same, $x_r(t)$ is only $x_a(t)$'s representation for a limited frequency spectrum.

5) As mentioned before the system outputs only a finite interval of frequencies while the input has increasing frequency. Therefore as seen in the plot of $y_2[n]$ the sound can only be listened in 2 passbands only.

6) original cut-off frequencies are $\frac{\pi}{3}, \frac{\pi}{5}$

$\Rightarrow$ cutoff frequencies of the wanted system : $\left\{\frac{\pi}{3T_S}, \frac{\pi}{5T_S}\right\}$

$$= \left\{\frac{\pi}{3\sqrt{\frac{\pi}{8192000}}}, \frac{\pi}{5\sqrt{\frac{\pi}{8192000}}}\right\} = \{1691, 1015\} \text{ rad/s}$$

7) Since the music is originally an analog signal when it is digitalized some of the information lost when storing which brings noise. Also when this signal pass through the filter, noises will pass as well. Therefore, the noises can be heard better when the signal passed through the filter.

8) Similar to part 7, some of the original data lost and noise occuring. when the data is passed through the filter noises can be heard better.

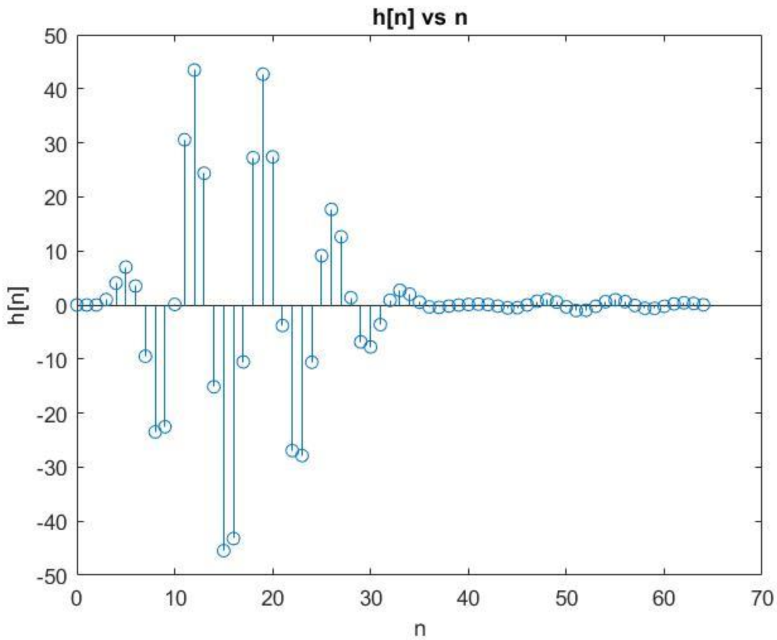

Figure 1) h[n] vs n

h[1:80]: [0.0,0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 4.0549, 7.0093, 3.5114, -9.4793, -23.4844, -22.5127,0.1122, 30.5737, 43.4585,24.3745, -15.1404, -45.4623, -43.2097, -10.5297,27.2547, 42.7015, 27.4034, -3.8027, -26.9469, -27.8571, -10.6015, 9.1178, 17.6848, 12.6231, 1.3517, -6.8283, -7.7683, -3.6287, 0.8719, 2.7172, 2.0064, 0.5168, -0.3611, -0.4404, -0.2061, -0.0199, 0.0914, 0.1579, 0.0791, -0.2136, -0.5291, -0.5072, 0.0025, 0.6888, 0.9791, 0.5491, -0.3411, -1.0242, -0.9734, -0.2372,0.614, 0.962, 0.6174, -0.0857, -0.6071, -0.6276, -0.2388, 0.2054, 0.3984, 0.2844, 0.0305, -0.1538, -0.175, -0.0817, 0.0196, 0.0612, 0.0452, 0.0116, -0.0081, -0.0099, -0.0046]

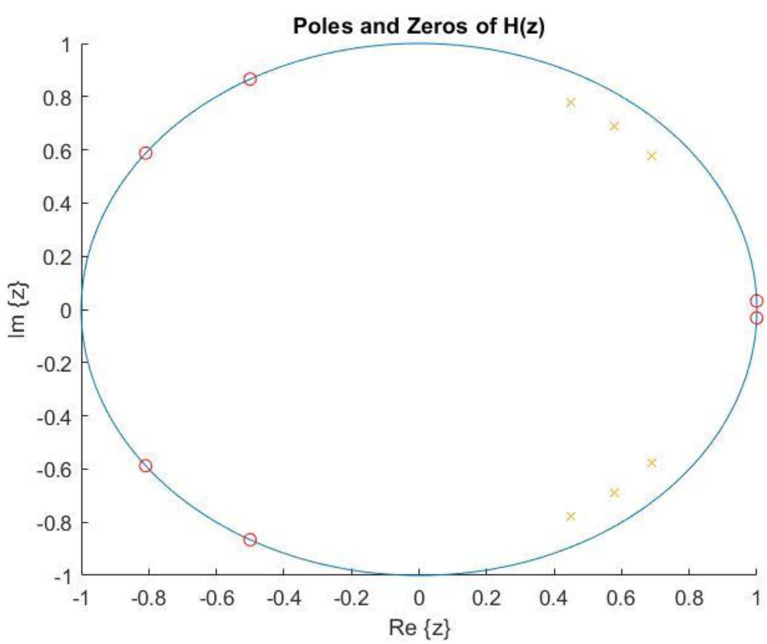

Figure 2) Poles and Zeros of H(z)

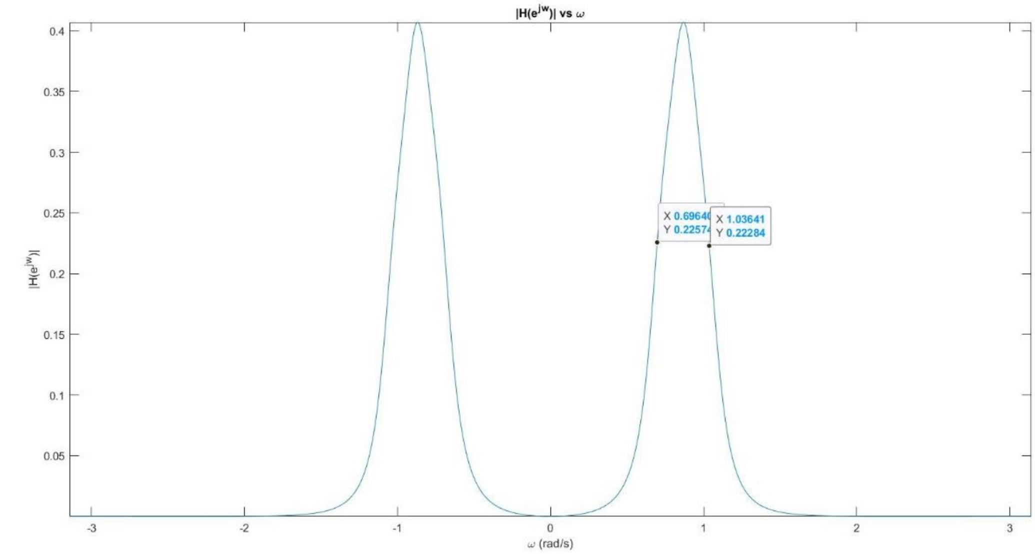

Figure 3) |H(z)| vs w

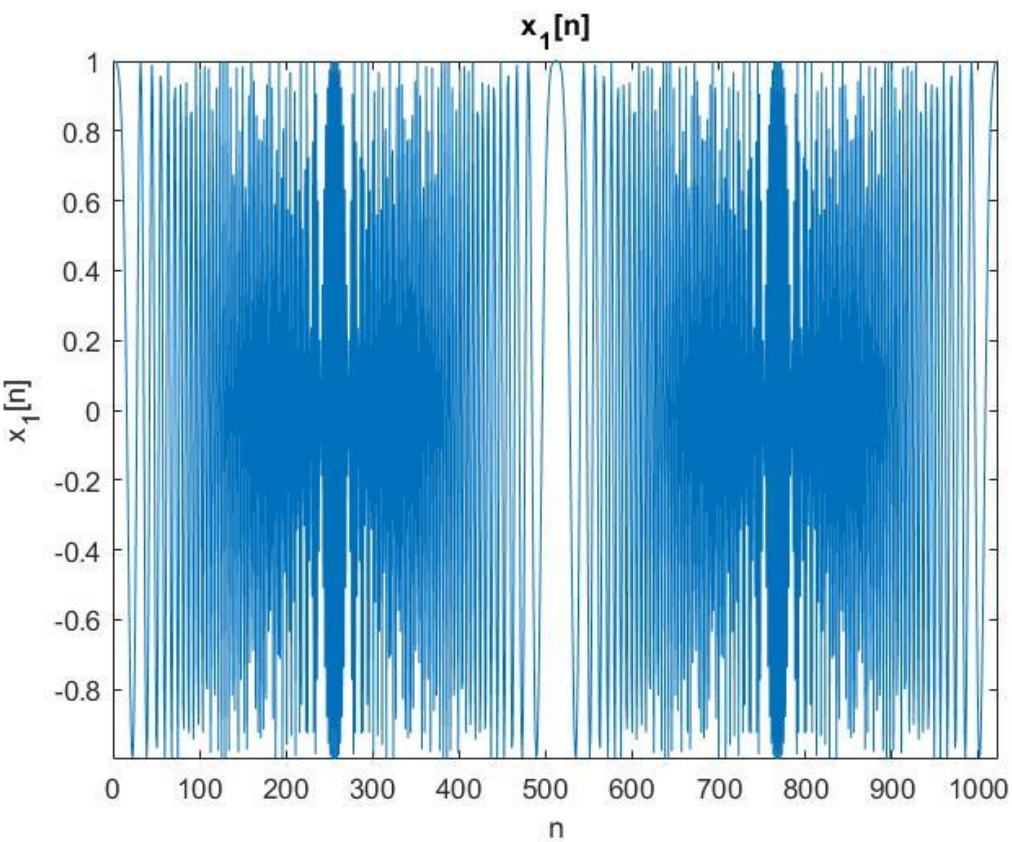

Figure 4) x_1[n] vs n

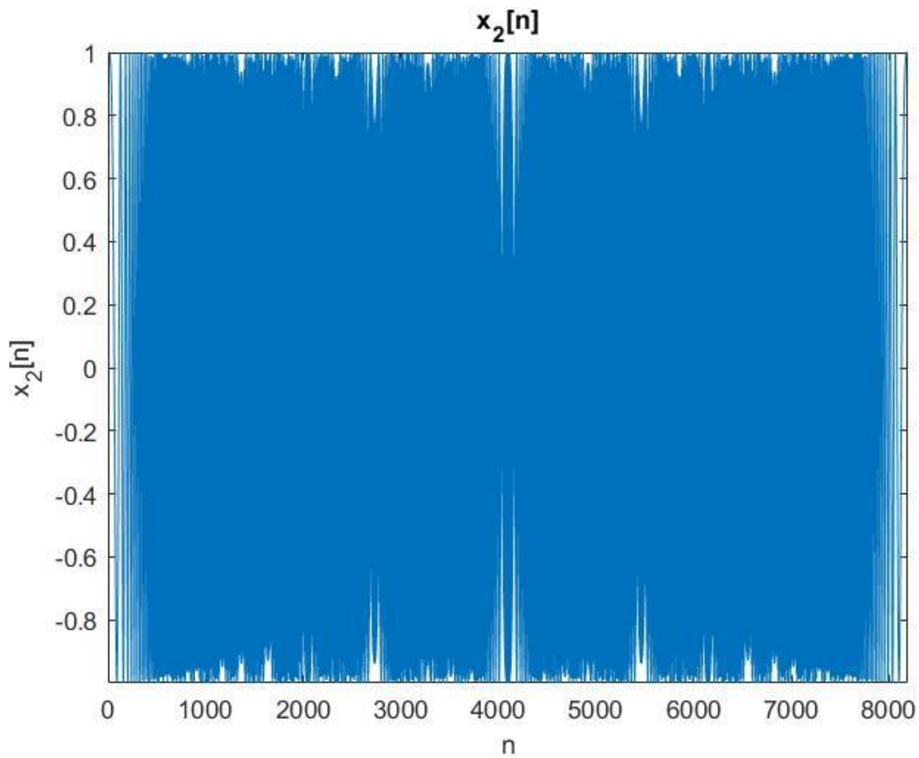

Figure 5) x_2[n] vs n

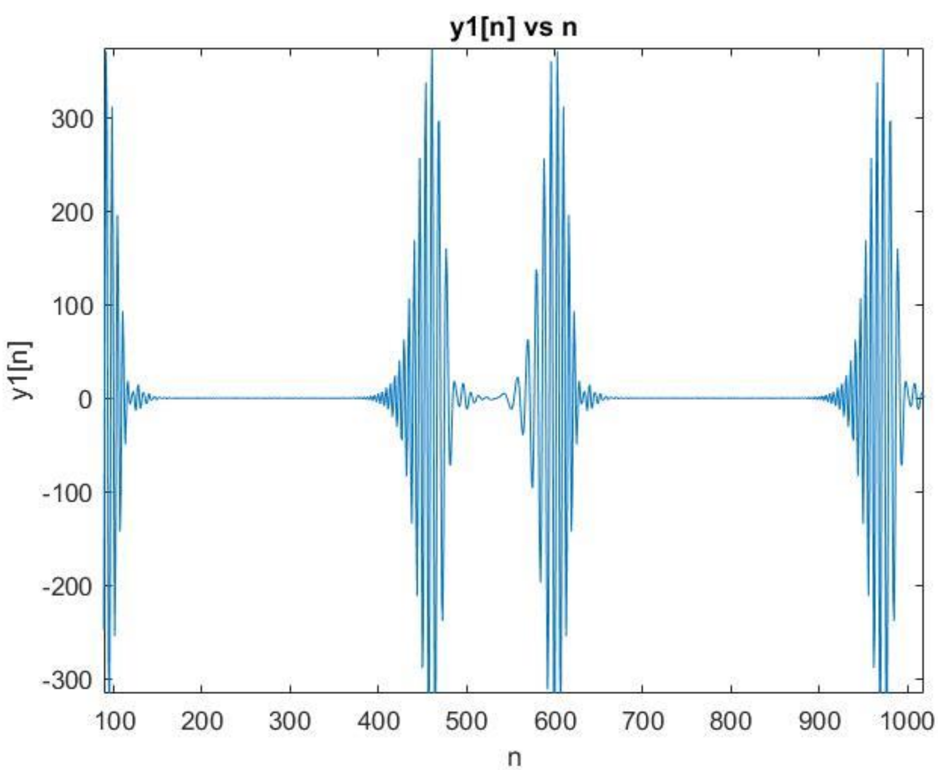

Figure 6) y_1[n] vs n

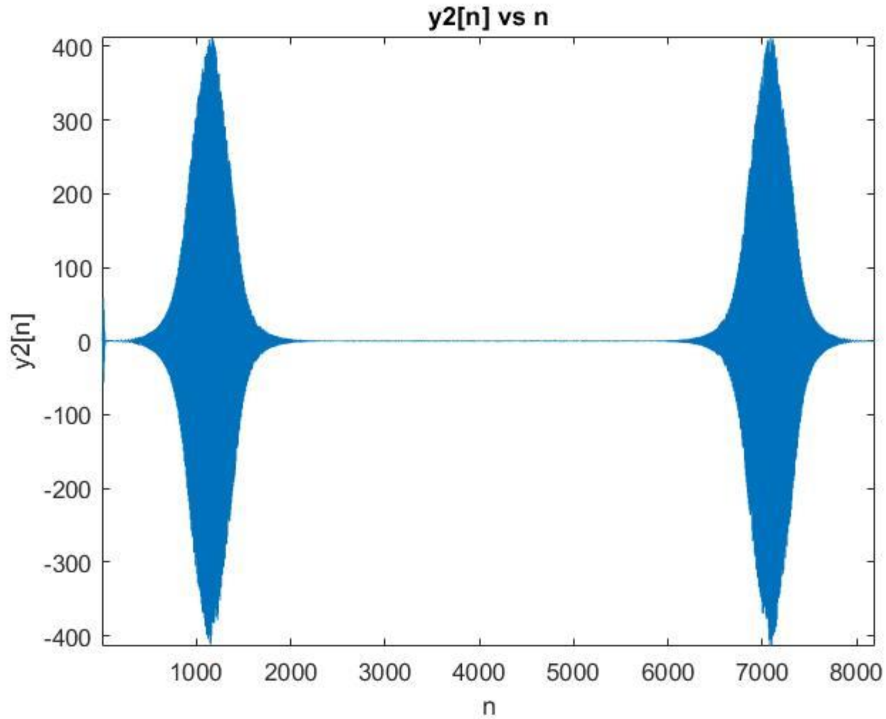

Figure 7) y_2[n] vs n

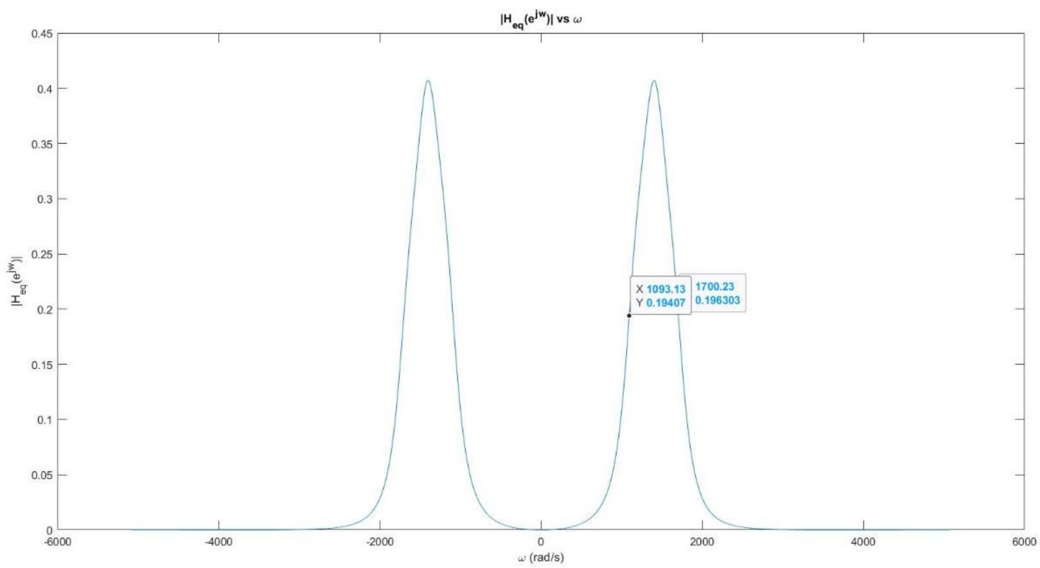

Figure 8) |H_eq(z)|vs w

```
close all;
%% Q1

pol = [40*pi/180, -40*pi/180, 50*pi/180, -50*pi/180, 60*pi/180, -60*pi/180];
zer = [ 1*pi/99, -1*pi/99, 2*pi/3, -2*pi/3, 4*pi/5, -4*pi/5 ];

c_p = real(poly(9*(exp(1i.*pol))/10));
c_z = real(poly(exp(1i.*zer)));

disp(c_z);
disp(c_p);

hn = zeros(2000,1);
x = zeros(2000,1);
x(1:7) = coef_z(1:7);
for i = 9:2000
    hn(i)=x(i-8);
    for k=2:7
    hn(i)=hn(i)-coef_p(k)*hn(i-k+1);
    end
    hn(n)=hn(i)/coef_p(1);
end
save('hn.mat', 'hn');
disp(hn);
figure(1)
indis = 0:1:64;
hn = hn(6:70);
stem(indis,hn);
title('h[n] vs n')
ylabel('h[n]');
xlabel('n');



k = -1:0.001:1;
ome = 0:pi/500:2*pi;
figure(2);
hold on
plot(cos(ome), sin(ome));
hold on
scatter(real((exp(1i.*z))),imag((exp(1i.*z))),'r');
axis tight;
hold on
scatter(real(9*(exp(1i*p))/10),imag(9*(exp(1i*p))/10),'x');
axis tight
title('Poles and Zeros of H(z)')
ylabel('Im \{z\}');
xlabel('Re \{z\}');
w = -pi:0.001:pi-0.001;
s=size(w,2);
for i = 1:s
x = 1;
for k = 1:6
x = x*(exp(1j*w(i))-9*(exp(1j.*p(k)))/10);
end
```

```
p1(i)=1/x;
end


for i = 1:s
x = 1;
for k = 1:6
x = x*(exp(1j*w(i))-(exp(1j.*z(k))));
end
z1(i)= x;
end


figure(3);
hz = z1 .* p1/1021;
plot(w,abs(hz));
title('|H(e^j^w)| vs \omega');
xlabel('\omega (rad/s)');
ylabel("|H(e^j^w)|")
axis tight


%% Q2
n_2 = 0:1:1023;
xf = cos((n_2.^2).*(pi/512));
figure(4);
plot(n_2,xf)
save('x_f.mat', 'xf');
xlabel('n');
ylabel('x_1[n]');
title('x_1[n]');
axis tight
n_3 = 0:1:8191;
xg = cos((n3.^2).*(pi/8192));
save('x_g.mat', 'xg');
figure(5);
plot(n3,xg);
title('x_2[n]');
ylabel('x_2[n]');
xlabel('n');
axis tight
%% Q3

y1= zeros(1, 1024);
for n=9:1024
    for i= 1:7
        y1(n)=y1(n)+coef_z(i)*xf(n-i+1);
    end
end

figure(6);
stem(1:1024,y1)
axis tight
title('y_1[n] vs n');
xlabel('n');
```

```
ylabel('y_1[n]');




y2= zeros(8192, 1);
for n=9:8192
    for i= 1:7
        y2(n)=y2(n)+coef_z(i)*xg(n-i+1)-coef_p(i)*y2(n-i+1);
    end
end


figure(7);
stem(1:8192,y2)
title('y_2[n] vs n');
xlabel('n');
ylabel('y_2[n]');
axis tight

%% Q4 5
y_1 =conv(hn, xf);
figure(10);
plot(1:1024,y_1(1:1024));
title('y1[n] vs n');
ylabel('y1[n]');
xlabel('n');
axis tight
sound(y1);
save('q4.mat', 'y_1');
audiowrite('q4.wav',y_1,48000);

y_2 = conv(hn,xg);
figure(11);
plot(1:8192,y_2(1:8192));
title('y2[n] vs n');
ylabel('y2[n]');
xlabel('n');
axis tight
sound(xg);
sound(y_2);
save('q5.mat', 'y_2');
audiowrite('q5.wav',y_2,48000);


%% Q6

Ts = sqrt(pi/8192000);
weq = -pi/Ts:0.001/(fix(1/Ts+1)*Ts):pi/Ts-0.001/Ts;
Heq=zeros(length(weq),1);
for i=1:length(weq)-4000
    Heq(i)= hz(fix(i*Ts+1)) + 1j*(hz(fix(i*Ts+1))-hz(fix(i*Ts)+2))/fix(1/Ts);
end
figure(12);
plot(weq,abs(Heq));
```

```
title("|H_{eq}(e^j^w)| vs \omega")
ylabel("|H_{eq}(e^j^w)|")
xlabel("\omega (rad/s)")


%% Q7
load("hn.mat");
[yy_1,Fs] = audioread('mozart_vl2_6.mp3');
music_new = yy_1(250000:500000,1);
music_new = conv(music_new,hn);
sound(music_new,Fs);
audiowrite('out_song.wav',music_new,Fs);


%% Q8

load("hn.mat");
music_new_1 = yy_2(:,1);
music_new_2=yy_2(:,2);
music_neww = [conv(music_new_1,hn),conv(music_new_1,hn)];
audiowrite('record_out.wav',music_neww,Fs);
```