Chen Ji
SUNetID:jjjjjc

Homework 5b

Problem 1

(a) The first order All-pass filter can be written as:

$$A(z) = \frac{\rho + z^{-1}}{1 + \rho z^{-1}}$$

The derivation of group delay is shown as below:

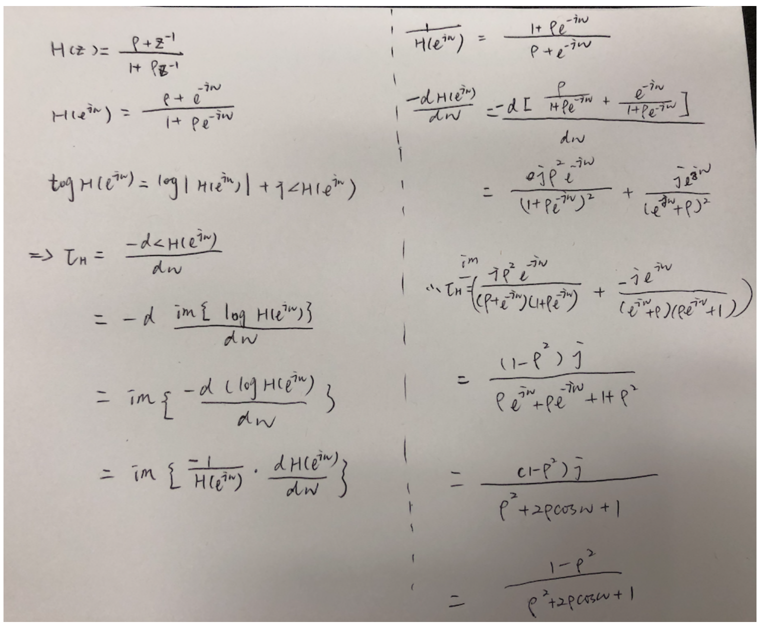


Figure 1: Derivation of all-pass group delay

Thus the Group Delay of all-pass filter can be calculated as:

$$\tau(w) = \frac{1 - \rho^2}{\rho^2 + 2\rho\cos(\omega) + 1}$$

(b) Plot the group delay for ρ values of [-0.75, -0.2, 0.5] in the frequency range of [0, fs/2]. The MATLAB script for doing this task is shown as below:

```
fs = 44100;
f = [0:fs]/(fs);
w = pi*f;
rho = -0.75;
groupDelay = (1-rho*rho)./(rho*rho + 2*rho*cos(w)+1);
plot(w,groupDelay)
hold on ;
rho = -0.2;
```

```
groupDelay = (1-rho*rho)./(rho*rho + 2*rho*cos(w)+1);
plot(w,groupDelay);
hold on;
rho = 0.5;
groupDelay = (1-rho*rho)./(rho*rho + 2*rho*cos(w)+1);
plot(w,groupDelay);
legend('rho = -0.75','rho = -0.2','rho = 0.5');
xlabel('radian frequency(radian/s)');
ylabel('Group Delay');
title('Group Delay of all-pass system when rho = -0.75, -0.2 and 0.5');
```

Then the plot can be obtained:

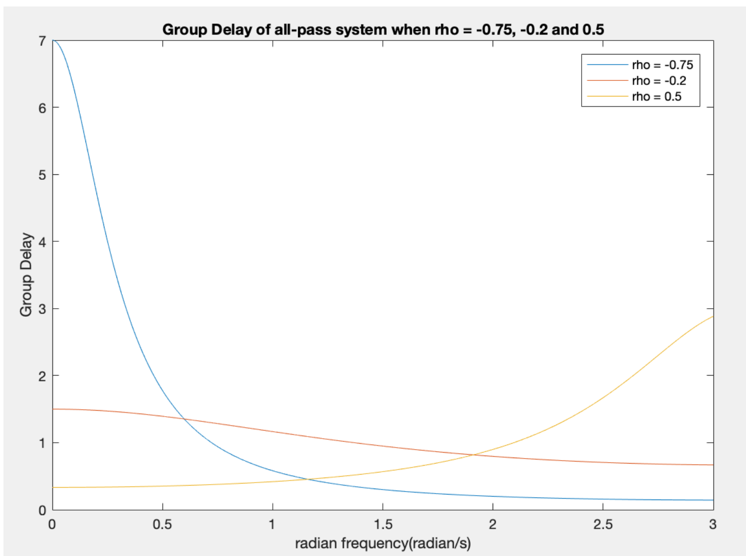


Figure 2: Impulse response of all-pass system when given different rho values

(c) The following impulse response shows the N cascaded first-order allpass filters:

```
N = 1000;
fs = 8000;
rho = -0.75;
t = 0:fs-1;
w = 2*pi*[0:fs-1]/fs;
A = (rho + exp(-1j*w))./(1 + rho*exp(-1j*w));
G = power(A,N);
g = real(ifft(G));
plot(t,g);
hold on;
rho = -0.2;
A = (rho + exp(-1j*w))./(1 + rho*exp(-1j*w));
G = power(A,N);
```

```
g = real(ifft(G));
plot(t,g);
hold on;
rho = 0.5;
A = (rho + exp(-1j*w))./(1 + rho*exp(-1j*w));
G = power(A,N);
g = real(ifft(G));
plot(t,g);
xlabel('time(samples)');
ylabel('amplitude');
legend('rho = -0.75','rho = -0.2','rho = 0.5');
title('impulse response of N cascaded all-pass filters')
```

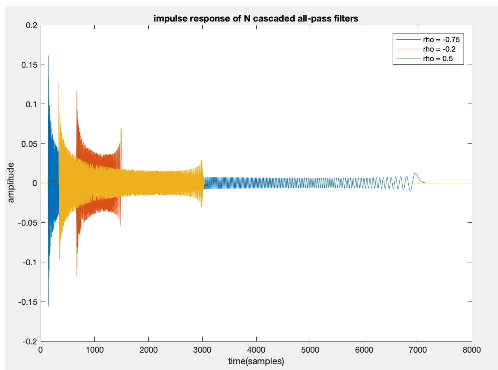


Figure 3: Group delay of all-pass system when given different rho values

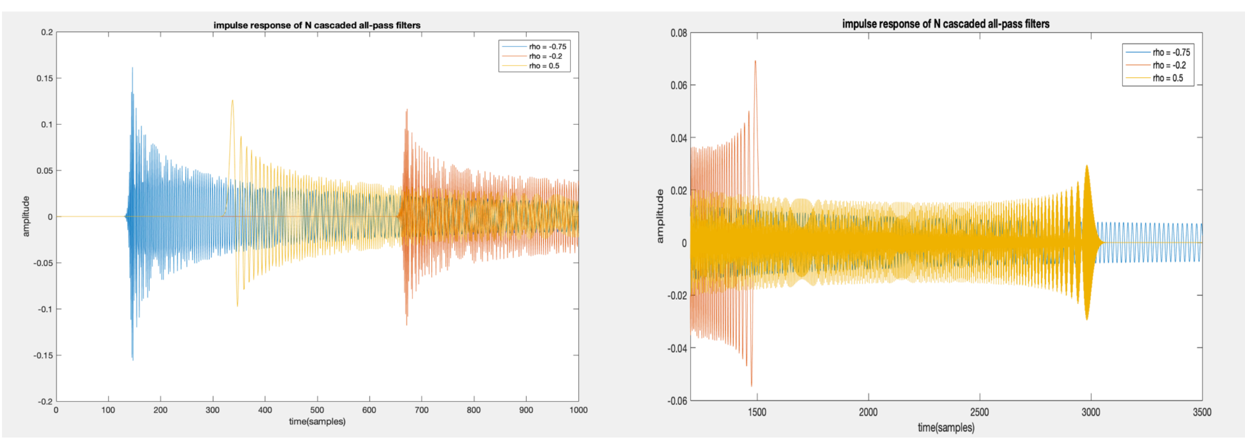
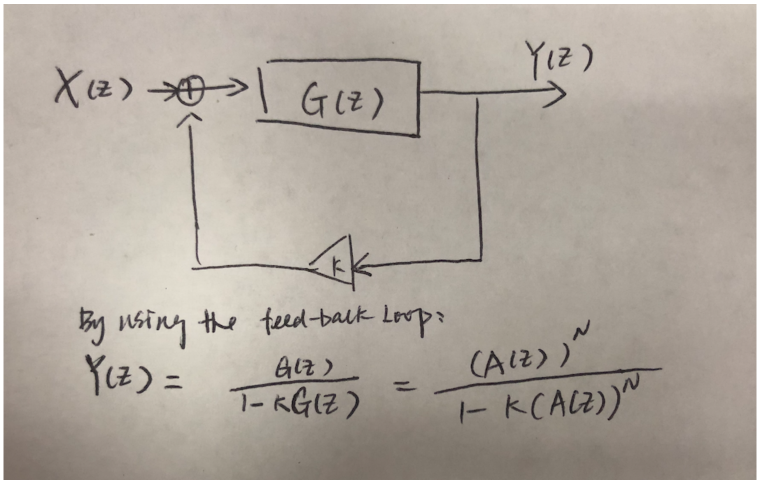


Figure 4: Group delay of all-pass system when given different rho values

It can be seen from the plot that when rho is negative, the impulse response concentrate the high-frequency is close to the zero origin while when the rho is positive, the low-frequency is close to the zero origin. When the rho has the same sign, the rho value is smaller, the impulse response will be more intense along the time axis.

(d) By applying the feedback control theory, the following equation can be obtained:

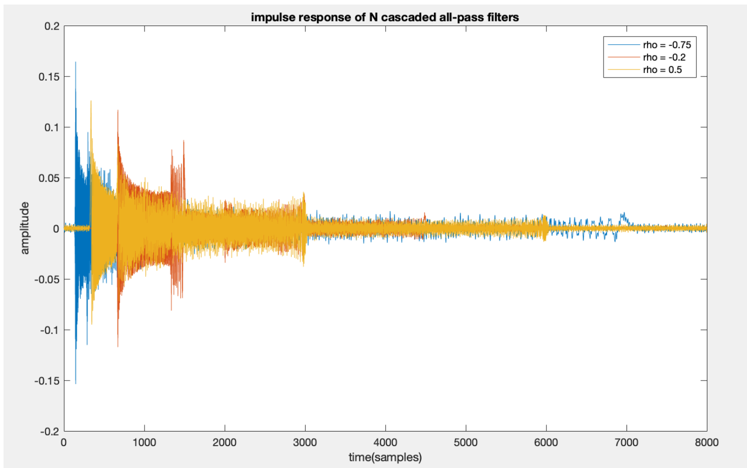


I calculate the Y(z) in the frequency domain, and then apply ifft in the MATLAB to get the impulse response, the following MATLAB script can be shown:

```
N = 1000;
k = 0.5;
fs = 8000;
rho = -0.75;
t = 0:fs-1;
w = 2*pi*[0:fs-1]/fs;
A = (rho + exp(-1j*w))./(1 + rho*exp(-1j*w));
G = power(A,N);
Y = (G)./(1-k*G);
y = real(ifft(Y));
plot(t,y);
hold on;
rho = -0.2;
A = (rho + exp(-1j*w))./(1 + rho*exp(-1j*w));
G = power(A,N);
Y = (G)./(1-k*G);
y = real(ifft(Y));
plot(t,y);
hold on;
rho = 0.5;
A = (rho + exp(-1j*w))./(1 + rho*exp(-1j*w));
G = power(A,N);
Y = (G)./(1-k*G);
y = real(ifft(Y));
plot(t,y);
xlabel('time(samples)');
ylabel('amplitude');
legend('rho = -0.75','rho = -0.2','rho = 0.5');
```

```
title('impulse response of N cascaded all-pass filters')
```

Then the plot can be shown:



From the polt, we can see that the impulse response is more noisy, and the noisy level is depend on the k (feedback coefficient).

Problem 2

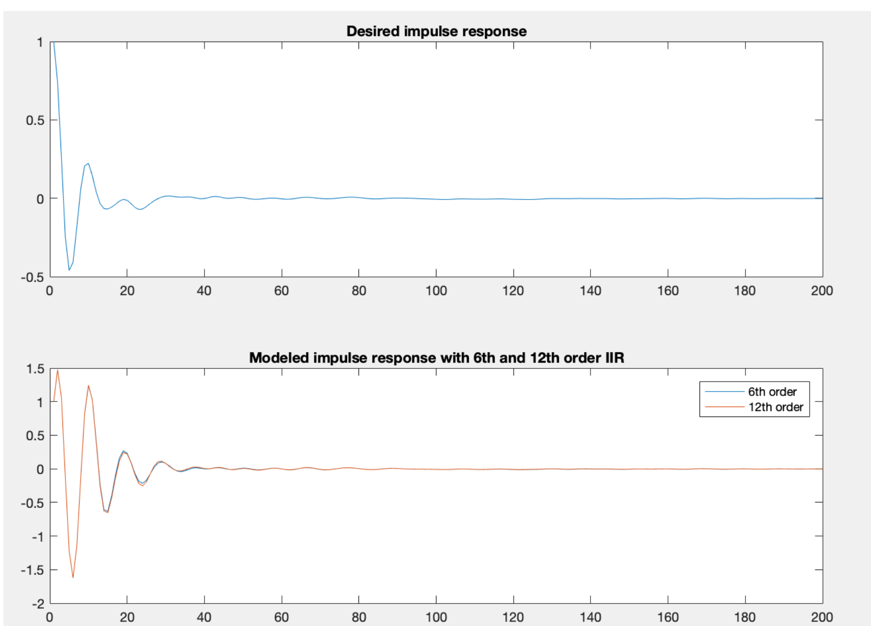
In this problem, you will fit an IIR filter to the FIR filter manipulated in the previous problems. Use a windowed, minimum-phase impulse response, based on a 2.0 ERB- smoothing of the measured impulse response.

(a) The Matlab code can bde written as:

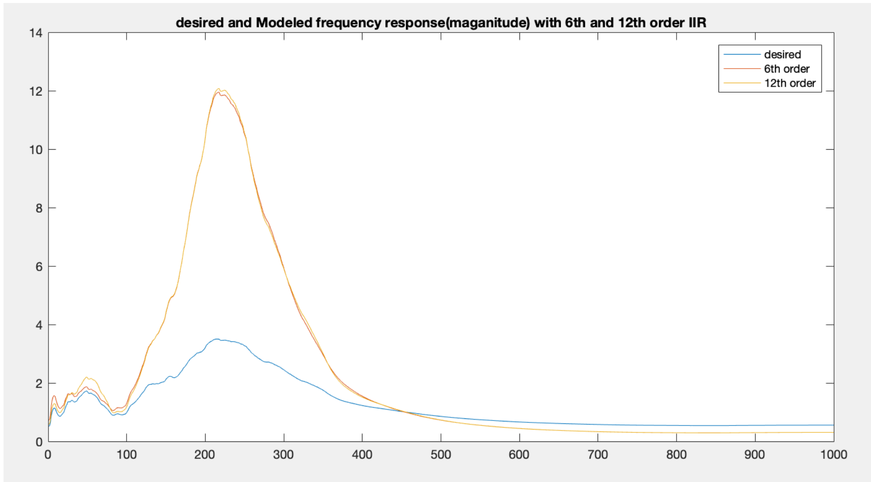
```
% Constants:
beta = 2.0;
N = 8192;
% Read file:
[h, fs] = audioread('Jen57On.wav');
% Get length:
L = length(h);
% Convert to mag:
h = h/max(abs(h));
H = fft(h,N);
smoothedH = cbsmooth(H,beta,N);
subplot(3,1,1);
smoothedh = real(ifft(smoothedH));
smoothedh = smoothedh/max(smoothedh);
smoothedh = smoothedh(1:2000);
```

```
plot(smoothedh);
xlim([0,200]);
title('Desired impulse response');
% Using prony's method
% 6th order
denom_order = 6;
num_order = 6;
[Num,Den] = prony(smoothedh,num_order,denom_order);
subplot(3,1,2);
model = filter(Num,Den,smoothedh);
plot(model);
xlim([0,200]);
title('Modeled impulse response with 6th order IIR');
% 12th order
denom_order = 12;
num_order = 12;
[Num,Den] = prony(smoothedh,num_order,denom_order);
subplot(3,1,3);
model = filter(Num,Den,smoothedh);
plot(model);
xlim([0,200]);
title('Modeled impulse response with 12th order IIR');
```

Then the plot can be obtained as below:



Also the spectrum diagram can be obtained by using FFT, and we can see from the plot , they are not matching with each other:



(b) Let's try warping before modeling the impulse using Prony's method. From the course reading, for choosing the ρ value, the following formula can be used:

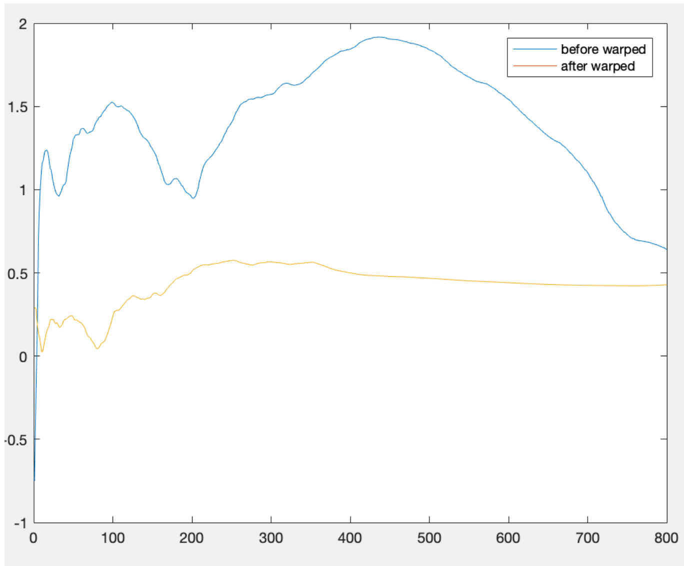
$$\rho^*_{\mathrm{ERB}}(f_s) \;=\; 1.05 \cdot \left[\frac{2}{\pi} \cdot \arctan(0.072 f_s)\right]^{\frac{1}{2}} - 0.196$$

Then we calculate the ρ is around 0.8. Then the MATLAB script is written as below:

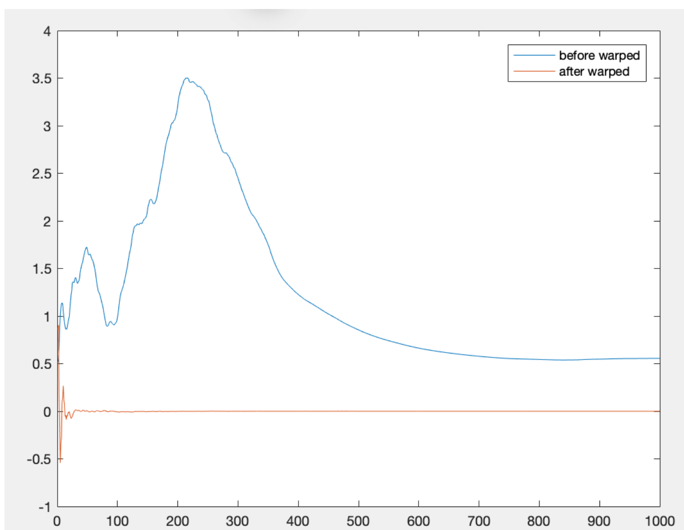
```
beta = 2.0;
N = 8192;
% Read file:
[h, fs] = audioread('Jen57On.wav');
% Get length:
L = length(h);
% Convert to mag:
h = h/max(abs(h));
H = fft(h,N);
smoothedH = cbsmooth(H,beta,N);
db = 20*log10(smoothedH);
plot(db/20);
xlim([0, 800]);
hold on;

smoothedh = real(ifft(smoothedH));
smoothedh = smoothedh/max(smoothedh);
smoothedh = smoothedh(1:2000);
rho = 0.85;
b = [rho 1];
a = [1 rho];
warped = filter (b,a,smoothedh);
warpedH = fft(warped);
```

```
plot(abs(log10(warpedH)));
xlim([0, 800]);
legend('before warped','after warped');
```



The impulse response are shown as below:



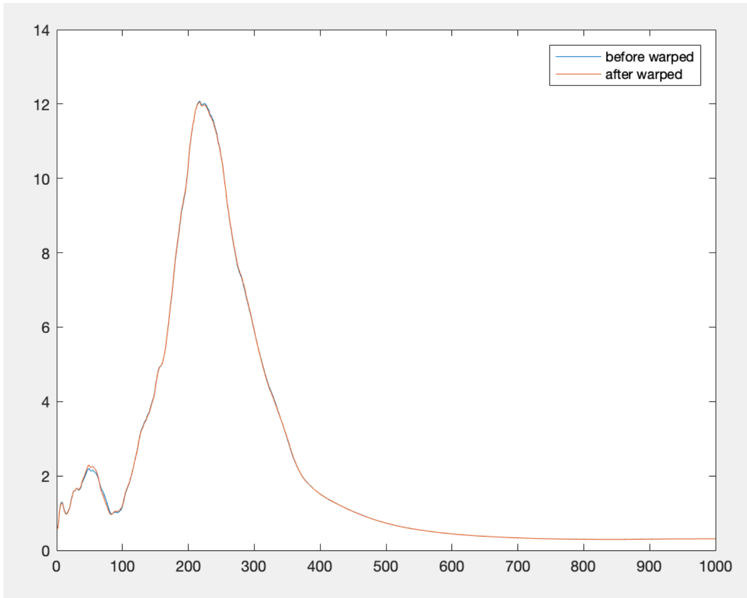
(c) Then using the prony's method with warped filter. The MATLAB code is:

```
beta = 2.0;
N = 8192;
% Read file:
```

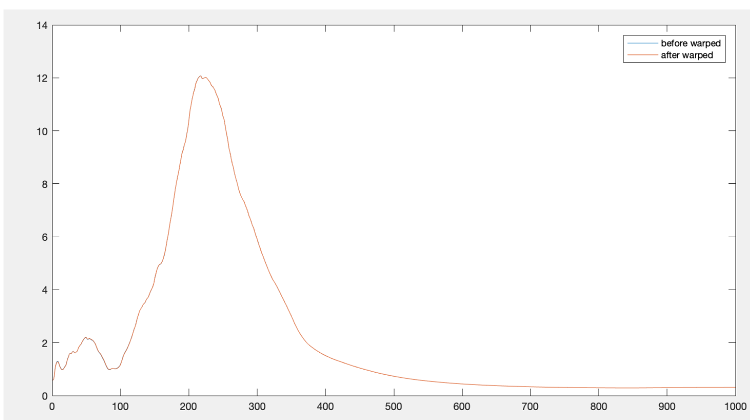
```
[h, fs] = audioread('Jen57On.wav');
% Get length:
L = length(h);
% Convert to mag:
h = h/max(abs(h));
H = fft(h,N);
smoothedH = cbsmooth(H,beta,N);
smoothedh = real(ifft(smoothedH));
smoothedh = smoothedh/max(smoothedh);
smoothedh = smoothedh(1:2000);
% 12th order
denom_order = 12;
num_order = 12;
[Num,Den] = prony(smoothedh,num_order,denom_order);
model = filter(Num,Den,smoothedh);
plot(abs(fft(model)));
xlim([0,1000]);
legend('before warped','after warped');

hold on;
rho = 0.8;
b = [rho 1];
a = [1 rho];
warped = filter (b,a,smoothedh);
% 12th order
denom_order = 12;
num_order = 12;
[Num,Den] = prony(warped,num_order,denom_order);
model = filter(Num,Den,warped);
plot(abs(fft(model)));
xlim([0,1000]);
legend('before warped','after warped');
```

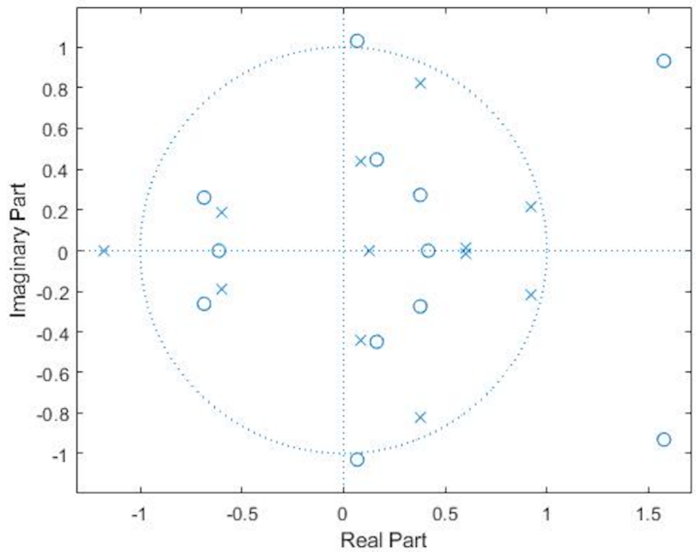
Then we can see that the plot matches with each other:



(d) By using the ρ as -0.8, the plot can be shown as:



(e) By using zplane, we can obtain the zero-pole plot as below:



(f) By using the guitar audio, we can convolve the audio with the FIR and warped FIR.

```
[audio, fs] = audioread('ElecGtr-Preflex-Dirty-dry.wav');
Audio = abs(fft(audio));
% Prony's method
denom_order = 12;
num_order = 12;
[Num,Den] = prony(smoothedh,num_order,denom_order);
model = filter(Num,Den,smoothedh);
modelH = abs(fft(smoothedh));

filtered = Audio*modelH;
% Warped Prony's method
rho = 0.85;
b = [rho 1];
a = [1 rho];
warped = filter (b,a,smoothedh);
denom_order = 12;
num_order = 12;
[Num,Den] = prony(smoothedh,num_order,denom_order);
model = filter(Num,Den,warped);
modelH = abs(fft(smoothedh));

filtered = Audio*modelH;
```

(g) Convolve the provided guitar clip with the given FIR filter. We compare uisng prony's filter with not using prony's filter.Warp the input impulse response using an allpass transformation with allpass parameter to set so as to assign a good amount of bandwidth to spectral features of importance. The audio before and after prony's method are in the files.